

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

An Information Theoretic Approach to Language Relatedness

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Information Systems
at Massey University.

Anand Venkt Raman
1997

This documentation was prepared from a camera ready copy made by the author using \LaTeX . The International Phonetic Alphabet characters were produced using the \TeX font (wsuipa) created by Washington State University. The graphs were produced using `gnuplot`. PFSA diagrams were produced using a modified version of Jos Van Einjdhoven's `graphplace` program. Other figures were produced using the `xfig` program. The final Postscript file was generated using `dvips`, printed on a Hewlett-Packard Laserjet printer and photoreproduced and bound by Wills Bookbinding and Printing, 6 Dahlia Street, Palmerston North.

The software used to produce the results for this work are available through anonymous ftp from the URL <ftp://fims-ftp.massey.ac.nz/pub/ARaman>. Bug reports may be sent to A.Raman@massey.ac.nz. Work done using this software must cite either Raman and Patrick (1997d) or this dissertation.

Abstract

This dissertation examines the prospect of applying information theoretic principles to help solve problems in historical linguistics. The Minimum Message Length principle attributed to Chris Wallace (similar to the Minimum Description Length principle of Jorma Rissanen) is used to judge the goodness of hypotheses in the field of historical linguistics. The idea is that theories that require a shorter message to describe with their data are better than those that require long messages.

Work in collecting the linguistic data tracing the derivation of some 2714 words in Modern Cantonese and Modern Beijing from their forms in a reconstruction of Middle Chinese is described as also is the work in transforming this data into a format suitable for use with software developed for this project.

Heuristics for inferring Probabilistic Finite State Automata (PFSA¹) from such data are reviewed and some new heuristics are introduced. These are then applied to training data and benchmark results presented.

Finally, the inference process is applied to the actual linguistic data which allows a conjecture regarding a relative closeness of the Chinese dialects to their reconstructed ancestor to be formed.

¹In this dissertation, the abbreviation PFSA has been used to denote both the singular and plural of these machines, the "A" in PFSA being understood to represent both *Automaton* and *Automata*.

Contents

1	Introduction	1
1.1	Statement	1
1.2	Linguistic Preliminaries	2
1.2.1	Phones, Phonemes, Allophones and Morphemes	2
1.2.2	Contrast, Minimal Pairs and Distributions	3
1.2.3	Language Change	5
1.2.4	Bleeding and Feeding	6
1.3	Language Reconstruction	7
1.3.1	The Comparative Method	7
1.3.2	Internal Reconstruction	9
1.4	Language Similarities and Relatedness	10
2	Quantitative Methods in Historical Linguistics	12
2.1	Early Work	12
2.2	Glottochronology	14
2.2.1	Description	15
2.2.2	Example	16
2.2.3	Criticisms of glottochronology	17
2.3	The Direction of This Thesis	18
3	Similarity Measures – An Example	19
3.1	Overview	19
3.2	Modelling *LNWG to OF	20
3.3	Modelling *LNWG to OHG	23
3.4	Discussion	24
4	Data Collection	32
4.1	Introduction	32
4.2	Original Sources and Reliability	32
4.2.1	The <i>Qie yun</i>	33
4.2.2	Reliability	34
4.3	<i>Zihui</i> and DOC	35
4.4	Motivation	37
4.5	Chen76 and CN84	39
4.6	Relative Chronology (RC)	41
4.7	Phonotactic Conditions and Allophonic rules	44
4.8	Automatic Derivations	47
4.9	Dealing with Exceptions	47
4.9.1	Exceptions to the RC	49
4.9.2	Undocumented Changes	49
4.9.3	Unapplied Rules	49

4.10	Status of Allophonic Changes	51
4.11	Consistency of Presented Data	52
5	The MML Principle	55
5.1	Modelling Inductive Hypotheses	55
5.2	Information Measures and MML codes	59
5.3	MML, Bayesianism and MDL	61
5.4	Encoding Deterministic PFSA	63
5.5	Encoding Non-Deterministic PFSA	66
6	The Sk-strings Method of Inferring PFSA	73
6.1	Introduction	73
6.2	Inducing the Gaines Machine	74
6.3	Minimum Message Length	76
6.4	The K-tails Approach	81
6.5	The Sk-strings Approach	83
6.6	Tractability Considerations	86
6.6.1	Method A - Build strings in decreasing order of probability . . .	87
6.6.2	Method B - Reject strings less than a certain probability	88
6.6.3	Method C - Reject relatively improbable strings	88
6.7	Results	89
6.8	Conclusion	96
7	A Heuristic Using Cross-Entropy for Sk-Strings	101
7.1	Introduction	101
7.2	A Cross-Entropic Measure	102
7.3	Implementation	105
7.4	Results	107
7.5	Conclusion	110
8	Beam Search and Simulated Beam Annealing	115
8.1	Introduction	115
8.2	Beam Search	115
8.2.1	The Beam Search Algorithm for PFSA Inference	116
8.2.2	Algorithm Complexity	117
8.3	Simulated Annealing	118
8.3.1	Simulated Annealing with a Beam	120
8.4	Results and Discussion	123
8.5	Discussion and Conclusion	126
9	Results and Conclusion	131
9.1	Program outputs	131
9.2	Discussion	133
9.2.1	Status of exceptions	134
9.2.2	Chen76 and CN84 on exceptions	137
9.2.3	Relation between the PFSA and RC diagrams	139
9.2.4	Significance of diachronic regularities in MB and MC	140
9.2.5	Glottochronology revisited	141
9.3	Conclusion and Prospects	142

A	Guide to Using the Programs	147
A.1	Introduction	147
A.2	Canon	149
A.3	Genstr	150
A.4	Randpfsa	150
A.5	Skstr, Ktail, Beams and Simba	150
A.6	Dfa	151
A.7	Grafit and Groupnodes	151
A.8	DISCLAIMER	152
B	Reduced PFSA	154
B.1	PFSA for SMC>MC (Diachronic only)	154
B.2	PFSA for SMC>MC (Diachronic and Allophonic)	157
B.3	PFSA for SMC>MB (Diachronic only)	160
B.4	PFSA for SMC>MB (Diachronic and Allophonic)	163
C	From Simplified Middle Chinese to Modern Beijing	166
D	From Simplified Middle Chinese to Modern Cantonese	204

List of Tables

1.1	Example illustrating bleeding and feeding relationships among phonological rules.	7
2.1	An example illustrating common cognate scores for glottochronology. . .	16
3.1	Diachronic rules for converting *LNWG → OF.	21
3.2	The sequences of diachronic rules to derive 20 words from *LNWG to OF.	23
3.3	Diachronic rules for converting *LNWG → OHG.	24
3.4	The sequences of diachronic rules to derive 20 words from *LNWG to OHG.	25
4.1	An example of the Relative Chronology in operation.	44
4.2	Allophones of ə; Operation of the *CHAMEL PC.	45
4.3	An example of an allophonic rule in feeding relationship with a diachronic rule.	51
4.4	An extract from Appendix D to illustrate the actual presentation of the data.	53
5.1	A comparison of MMLs for 1 to 5 state PFSA explaining Andreae's rigged Casino.	70
6.1	MMLs of 1000 inferred automata using the AND heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).	90
6.2	MMLs of 1000 inferred automata using the OR heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).	90
6.3	MMLs of 1000 inferred automata using the LAX heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).	92
6.4	MMLs of 1000 inferred automata using the STRICT heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1). . . .	92
6.5	MMLs of 1000 inferred automata using the AND heuristic for various tail sizes and agreement percentages (Training set 2).	92
6.6	MMLs of 1000 inferred automata using the OR heuristic for various tail sizes and agreement percentages (Training set 2).	93
6.7	MMLs of 1000 inferred automata using the LAX heuristic for various tail sizes and agreement percentages (Training set 2).	93
6.8	MMLs of 1000 inferred automata using the STRICT heuristic for various tail sizes and agreement percentages (Training set 2).	93
6.9	The number of times (out of 100 training sets) that each tail size succeeded in inferring the minimum automaton using K-tails and the various sk-strings heuristics.	94
6.10	Success rates of the k-tails and 4 sk-strings heuristics in inferring the best PFSA out of 100 small test cases.	95

6.11	Success rates of the k-tails and 4 sk-strings heuristics in inferring the best PFSA out of 100 large test cases.	97
7.1	Success rates of the k-tails and 6 sk-strings heuristics (including XENTROPIC and VARDIST) in inferring the best PFSA out of 100 small test cases	111
7.2	Success rates of the k-tails and 6 sk-strings heuristics (including XENTROPIC and VARDIST) in inferring the best PFSA out of 100 large test cases	112
8.1	The number times (out of 100 training sets) that each beam size (1-10) succeeded in inferring the minimum automaton using the beam search and simba search methods.	125
9.1	MMLs for the canonical PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively.	131
9.2	MMLs for the canonical PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively using Formula (5.4).	132
9.3	MMLs for the reduced PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively.	132
9.4	MMLs for the PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively further reduced with beam search.	133
9.5	20 most frequent exceptions in the phonology of Modern Beijing.	136
9.6	20 most frequent exceptions in the phonology of Modern Cantonese.	137
9.7	20 most frequent rules in the phonology of Modern Beijing and Modern Cantonese.	138

List of Figures

3.1	Canonical PFSA representing the derivation of 20 *LNWG words into OF.	28
3.2	Canonical PFSA representing the derivation of 20 *LNWG words into OHG.	29
3.3	A generalisation of the PFSA in Figure 3.1 (*LNWG>OF).	30
3.4	A generalisation of the PFSA in Figure 3.2 (*LNWG>OHG).	31
4.1	Chen76 Relative Chronology for *SMC>MB.	41
4.2	Chen76 Modified Relative Chronology for *SMC>MB: RC25 (SHARP > PROCOPE) orders SHARP before PROCOPE.	42
4.3	CN84 Relative Chronology for *SMC>MC.	43
5.1	An encoding of 4 events (ABCD) ⁿ	57
5.2	An coding of the 4 (ABCD) ⁿ with an exception B'.	58
5.3	The events (ABCD) ⁿ encoded in the form of a PFSA.	58
5.4	The events (ABCD) ⁿ with exception B' encoded in the form of a PFSA.	59
5.5	An Illegal Automaton.	65
5.6	1-state PFSA explaining Andrae's rigged Casino.	70
5.7	2-state PFSA explaining Andrae's rigged Casino.	71
5.8	3-state PFSA explaining Andrae's rigged Casino.	71
5.9	4-state PFSA explaining Andrae's rigged Casino. Note that the output is completely predictable in states 1 and 2.	72
5.10	5-state PFSA explaining Andrae's rigged Casino.	72
6.1	Gaines' 4 state machine.	75
6.2	Two automata generating the same language: a*.	76
6.3	4 theories for the event sequence "abc!" seen once.	77
6.4	4 theories for the event sequence "abc!" seen 100 times.	78
6.5	2 candidate states for merging in a PFSA.	82
6.6	K-tails and sk-strings reductions of a canonical machine.	83
6.7	Test machine used to generate strings for training set 2.	91
6.8	MMLs of 1000 inferred automata using the AND/OR heuristics for tail sizes of 1-10 and agreement percentages of 1-100 (Training set 1). See also Tables 6.1 and 6.2.	94
6.9	MMLs of 1000 inferred automata using the LAX/STRICT heuristics for tail sizes of 1-10 and agreement percentages of 1-100 (Training set 1). See also Tables 6.3 and 6.4.	95
6.10	MMLs of 1000 inferred automata using the AND/OR heuristics for tail sizes of 1-10 and agreement percentages of 1-100 (Training set 2). See also Tables 6.5 and 6.6.	96

6.11	MMLs of 1000 inferred automata using the LAX/STRICT heuristics for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 2). See also Tables 6.7 and 6.8.	97
6.12	The k-tail method compared with sk-strings (s=50%) using 100 random training sets generated by small random automata (Best machines inferred using any heuristic and tail sizes from 1 to 10).	98
6.13	The k-tail method compared with sk-strings (s=50%) using 100 random training sets generated by large random automata (Best machines inferred using any heuristic and tail sizes from 1 to 5).	99
7.1	MMLs of 1000 inferred automata using the XENTROPIC heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 1).	108
7.2	MMLs of 1000 inferred automata using the VARDIST heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 1).	109
7.3	MMLs of 1000 inferred automata using the XENTROPIC heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 2).	110
7.4	MMLs of 1000 inferred automata using the VARDIST heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 2).	111
7.5	The sk-strings (s=50%) XENTROPIC heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by small random automata.	112
7.6	The sk-strings (s=50%) XENTROPIC heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by large random automata.	113
7.7	The sk-strings (s=50%) VARDIST heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by small random automata.	113
7.8	The sk-strings (s=50%) VARDIST heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by large random automata.	114
8.1	Performance of the beam search and simba search procedures on Training set 1 from Chapter 6. The plot shows the MMLs of 1000 machines induced against various beam sizes or tail sizes (for the 6 sk-strings heuristics).	123
8.2	Performance of the beam search and simba search procedures on Training set 2 from Chapter 6. The plot shows the MMLs of 1000 machines induced against various beam sizes or tail sizes (for the 6 sk-strings heuristics).	124
8.3	Beam search compared with sk-strings (s=50%) for 100 training sets generated from small random automata.	126
8.4	Simba search compared with sk-strings (s=50%) for 100 training sets generated from small random automata.	127
8.5	Beam search compared with sk-strings (s=50%) for 100 training sets generated from larger random automata.	128
8.6	Simba search compared with sk-strings (s=50%) for 100 training sets generated from larger random automata.	129
8.7	Seeded beam search compared with sk-strings (s=50%) and unseeded beam search for 100 training sets generated from small random automata (Best of tail sizes 1 to 10 and beam sizes 1 to 10).	130
9.1	Reduced PFSA for the diachronic phonology from Middle Chinese to Modern Beijing (Allophonic detail excluded).	145

9.2	Reduced PFSA for the diachronic phonology from Middle Chinese to Modern Cantonese (Allophonic detail excluded).	146
A.1	Andreae's 5 state machine without vertical node grouping	152
A.2	Andreae's 5 state machine with nodes 0, 2 and 4 grouped vertically . . .	153

Preface

In late 1993, Professor Jon Patrick, who was then just about to leave Deakin University in Melbourne for Massey gave a seminar here on some computational work he was doing in Basque linguistics. I had at that time just joined the Computer Science department as an assistant lecturer and was avidly looking for a PhD project to begin. Like almost everybody else I have met, I also had a compelling interest in historical linguistics and had my own opinion of how it should be done. I sent a brief CV to Jon asking if he would like to take on a doctoral student in his Basque project after his arrival and the answer soon came back in the affirmative. In the beginning, though, our intention was quite different, and much more grandiose and ambitious — we were hoping to assess whether or not the linguistic isolate Basque was related to the Dravidian language Tamil, which by the way, I happen to be very fluent in.

As time went on, however, the focus of the project changed rather dramatically and eventually became much more realistic. It narrowed down into developing a technique for finding distance measures between related natural languages to aid linguists in their task of subgrouping. We decided, wisely perhaps, that bold proposals in the linguistic domain are best made by linguists. We also decided, with the advice of my second supervisor, Dr John Newman in the Department of Linguistics and Second Language Teaching, to use Chinese data for the project. This data had been provided to John courtesy of Professor William S-Y. Wang at the University of California, Berkeley.

The field of computational linguistics is such that almost all Computer Scientists seem to have their own unique notions of how one should apply computational methods to historical data. While there may be merits and demerits to each such approach, this dissertation is not an evaluation of their relative goodness. It is a detailed account of one particular approach — the one described here — and how it can be used to provide linguists with a tool for effective subgrouping of languages.

Overview of contents

Chapter 1 gives a brief introduction to the problem and describes some linguistic terms and concepts for the benefit of the reader unacquainted with them. Chapter 2 is a brief survey of previous work in the field of linguistics which has been pursued along similar lines. An example of the methodology applied to a toy problem is then provided in Chapter 3. Chapter 4 describes the procedure by which the data for this project was collected and the motivation for using it. The MML criterion, which is central to this project, is introduced and described in Chapter 5, which is followed by Chapters 6–8 where methods to infer structure from the data using the MML criterion are looked at. Finally, Chapter 9 gives the results obtained, discusses them and describes the prospects for future work in this area.

Publications

Parts of this dissertation have been published in various places during the course of working on this project. The following is a list of them, with brief notes on which chapters they refer to.

Raman, A. V. and J. D. Patrick (1997). Linguistic similarity measures using the Minimum Message Length principle. In R. D. Blench and M. Spriggs (Eds.), *Archaeology and language I: Theoretical and methodological orientations*, pp.260–277, London: Routledge. This paper was also read at the WAC-3 Conference, New Delhi, December 1994. Material from this paper can be found in Chapters 2 and 3.

Raman, A. V. and J. D. Patrick (1997). The sk-strings method for inferring PFSA. In *Proceedings of the workshop on automata induction, grammatical inference and language acquisition at the 14th international conference on machine learning — ICML-97*, Nashville, Tennessee, (in press). Material from this paper can be found in Chapter 6.

Raman, A. V. and J. D. Patrick (1997). A heuristic using cross-entropy for sk-strings In *Technical Report TR 1/97*, Information Systems Department, Massey University. Material from this report can be found in Chapter 7.

Raman, A. V. and J. D. Patrick (1997). Beam search and simba search for PFSA inference, In *Technical Report TR 2/97*, Information Systems Department, Massey University. Material from this report can be found in Chapter 8.

Raman, A. V., J. Newman and J. D. Patrick (1997). A complexity measure for diachronic Chinese phonology. In *Proceedings of the SIGPHON97 workshop on computational phonology*, Madrid (in press). Material from this paper can be found in Chapter 9.

An explanation for some typesetting decisions

I feel compelled to say a few words about the typesetting process this dissertation has been through. I initially started writing the thesis using Microsoft Word on my Apple Macintosh, but found its performance unsatisfactory. Moreover, all my programs to manipulate the data were being run under Unix. What I wanted was a single command that could automatically take the ASCII outputs from my various experiments, turn them into tables, graphs and pictures and insert them in the right places in the thesis and also generate my bibliography neatly.

I thus deemed it best to migrate the work to \LaTeX and embarked on this bold venture. Not one bit of this effort was wasted as \LaTeX , and more particularly the Unix environment have paid it back several times and over. The amazing variety of utilities and high-quality programs available freely under Unix enabled me to get precisely what I wanted. Unix scripts using `awk`, `sed` and `grep` did most of the initial formatting of my program's ASCII outputs. All the formatted tables were generated automatically by scripts that ran the experiments themselves. So were the various graphs which were produced using the excellent `gnuplot` program. The various files which made up the chapters were edited under \TeX mode in GNU `emacs`. `Emacs` macros also helped immensely in migrating my bibliography database from `EndNote` on the Macintosh into `BibTeX`. Finally, I was able to put the various commands into a `Makefile` and my dream of generating the entire thesis with a single command was realised. "`Make dvi`" generated the dvi version of my thesis, which I then printed off using `dvips`. The dvi and postscript versions of the thesis are available from me on request. My thanks to Donald Knuth for \TeX , Leslie Lamport for \LaTeX , Oren Patashnik for `BibTeX` and to the authors of the various free software packages, especially GNU `emacs` and family, which I have found to be of utmost use in typesetting this work.

BibTeX explicitly recommends against the use of the Chicago style A for bibliographies.² Since I did use that style in the end, I feel a need to justify my decision. When I read a paper, I like to get an idea of when a certain result being cited was achieved. The plain style doesn't allow for this easily, as it is annoying having to flip to the reference section each time you have passed a citation. Also, I am told that although real scientists don't admit to it, most of us do unofficially build up knowledge about the reputation of authors and would like to know if a result being cited is attributed to a reputable author or not. Again, the plain style doesn't allow for this easily. Perhaps it is alright for a short paper as there aren't too many references to wade through, but in a dissertation such as this, I felt Chicago A would be most appropriate.

Acknowledgments

From the time this project began in 1994, several people have had signal input to it. It would be futile task to list them all, but I try to do a modest job of it here.

First and foremost among those I should thank are Jon Patrick and John Newman, my two supervisors. Jon's infectious enthusiasm for MML and comparative linguistics from which this project originated has been the source of much inspiration for many of the results achieved. In many ways, Jon was my ideal supervisor. His constantly supportive role played a more than important part in this project and his eagerness to discuss and try out new ideas was extremely refreshing during times of burn-out. Jon was always willing to listen, not only to my research problems but also to all my ideas. I also thank him for financial assistance he has provided in travelling to conferences so as to present this work.

I would, of course, have found it impossible to do much at all in this project if not for John Newman, under whose watchful eye I learnt all the linguistics necessary³ and finally how to achieve consistency in the presentation of this dissertation. I owe him tremendously for his countless pieces of invaluable advice and for introducing me to the scientific method in linguistics. Thanks also for his timely injections of encouragement which supplied me with much needed fuel to march on with this work.

In mid-1996, I had the opportunity to spend one week at Monash University, Melbourne, discussing various aspects of this project with researchers in the Computer Science Department there. Most importantly, it gave me a chance to meet and discuss with Chris Wallace, who in 1968 had originated the MML principle that is fundamental to this work. However, things didn't go as planned — I was down with a severe cold just that week and Chris was already semi-retired from the University at that stage. I only got to meet him twice, and that too only briefly. Nevertheless, these two brief meetings made all the difference. Chris had given me several important insights⁴ into the use of MML and I had given him my cold. Chris retired from Monash as this dissertation was being written up. I hope this is fitting as a tribute to a man who laid the foundations for what is and will continue be a prolific research area in future. The kindness of the Monash CS Department who made available to me resources during that week is also gratefully acknowledged.

Other people who have also had significant input to various parts of this work by way of stimulating discussions are John Hudson and Peter Kay, Computer Science Department Massey University, Peter Andreae, Computer Science Department, Victoria University, Ian Witten and Craig Neville-Manning, Computer Science Department,

²See BibTeXing, by Oren Patashnik, p.11

³This included auditing his phonology lectures in 1994.

⁴The decision to use a prior and directly specify the number of outgoing arcs from a state of the PFSA in Chapter 5 is due to a discussion I had with him on the 26th August.

Waikato University and Lloyd Allison, Rohan Baxter, Jon Oliver and Matthew Collins, Computer Science Department, Monash University. Peter Kay also proof-read pre-final drafts of this thesis and made many useful suggestions to improve its content. I am certain that any errors you may discover in this thesis belong to the few paragraphs that have been added since his careful eyes had scoured the thesis.

The work has also benefited greatly from several reviewers, some anonymous, who read interim reports and portions of the work presented at conferences. Among them, special thanks go to Dr Roger Blench, Cambridge University, Dr Peter Christian, Goldsmith's college, London, and Dr Sheila Embleton, York University, Toronto.

Dr Rosemary Haddon, Department of East Asian Languages, Massey University gave me a primer course in Mandarin and I am grateful to her for letting me audit her lectures in 1995.

Dr Siva Ganesh, Statistics Department, Massey University spent much effort in introducing me to SAS for graphing some of my results. I thank him for it, although I decided in the end to use gnuplot. Likewise, thanks also to the software support staff of the Faculty, who persevered in helping me with several Microsoft programs, although I decided in the end to not use any of them. I did, however, find much use for Philip Etheridge's timely and prompt assistance with T_EX and L^AT_EX and the many new packages he had to install on the system for me.

Ben Patrick and Kirsty How undertook the painstaking jobs of entering into a spreadsheet the derivations for each of the 2714 words in Modern Beijing and Modern Cantonese. I doubt if I could have done a better job in less time. My thanks go to both of them.

The Computer Science Department at Massey, especially Dr Chris Phillips, deserve a big note of thanks for arranging things so that I would have a reduced teaching load during the time I wrote up. Also, thanks to Professor Mark Apperley, now at Waikato University, who encouraged me to apply for an academic position in 1993 and start work on my PhD.

Finally I must record here my heartfelt thanks to my lovely wife Mathangi and beautiful son Panini⁵ who made it possible to persevere in the face of apparently blank walls in my research at times.

⁵<http://fims-www.massey.ac.nz/~ARaman/images/pdgr.jpg>

A Koan about Prior Knowledge⁶

In the days when Sussman was a novice, Minsky once came to him as he sat hacking at the PDP-6.

“What are you doing?”, asked Minsky.

“I am training a randomly wired neural net to play Tic-Tac-Toe” Sussman replied.

“Why is the net wired randomly?”, asked Minsky.

“I do not want it to have any preconceptions of how to play”, Sussman said.

Minsky then shut his eyes.

“Why do you close your eyes?”, Sussman asked his teacher.

“So that the room will be empty.”

At that moment, Sussman was enlightened.

⁶Taken from The New Hackers Dictionary, 2nd ed., Compiled by Eric S Raymond, MIT Press, Cambridge MA, 1994, p.475

Chapter 1

Introduction

1.1 Statement

This dissertation examines the problem of determining distance measures between related natural languages and the possibility of solving it through the application of information theoretic methods. The problem that is considered is as follows. Suppose we have a set of languages which we can group into a single family based on linguistic evidence. Suppose also, that we are given a linguistic hypothesis that postulates an ancestral language, now possibly non-existent, from which the given languages are said to have descended. Is there any objective method by which we can tell how closely a selected language from this family is related to its linguistic ancestor? A related problem is that of choosing one of many postulated ancestral languages. Ideally, we would also like to study the complexity of the process by which a daughter language evolved from its parent by determining the amount of regularity or irregularity in this evolution. What methods exist to make such facilities available to the interested linguist? In the following discussion, we look at the problem description in more detail.

1.2 Linguistic Preliminaries

In this section we review some linguistic details that are relevant to this work. Because of the interdisciplinary nature of this work, a brief description of linguistic terms and concepts is provided here for the benefit of the reader not acquainted with linguistic terminology. For more comprehensive explanations, see Fromkin and Rodman (1983), Bright (1992), (Crowley 1987), Crowley et al. (1995) or other such introductions to the field of linguistics.

1.2.1 Phones, Phonemes, Allophones and Morphemes

A phone is the unit of sound in a language. When we work with the individual sounds in a language, we are said to be working with phones, or at the phonetic level. The phonetic level is regarded as the most concrete level of analysis in a language (Crowley et al. 1995, p.1). Although the human vocal apparatus is capable of making a large number of phones, no language uses all possible vocal sounds distinctively.

Phonemes are an abstraction over phones in a language. In contrast to phones, phonemes are defined to be significant units of analysis which function as units of sound in a language, i.e. phonemes are just the significant phones in a particular language. Thus while phones can be defined independent of any language, phonemes are specific to a given language. For example, the phones [d] and [ð] are distinct phonemes in the English language because they are significant in differentiating between, say, the words *day* and *they*. However, in Palauan, the language of Belau in Micronesia, this difference is insignificant, since speakers may change from a [d]-sound to a [ð]-sound without changing the meaning of what they say (Crowley et al. 1995, p.2). So [d] and [ð] belong to the same phoneme in Palauan. We represent this phoneme simply as /d/ for convenience. The level of language dealing with phonemes is called the phonemic or phonological level.

Phones are usually denoted with square brackets whereas phonemes are denoted with slashes. So for example, we could say the phoneme /d/ in Palauan consists of the two phones [d] and [ð]. Phones which are thus associated with a

single phoneme are called allophones of this phoneme.

Sometimes allophones can be in free variation, meaning that the speakers can use either of the allophones without altering the meaning of what they wish to say. At other times, there are rules that govern which allophone of a particular phone is to be used in which context. Rules that describe the distribution of allophones for a given phoneme are called allophonic rules.

A morpheme is a larger unit of sound that is composed of one or more phonemes. It can be said to be the smallest meaning-bearing unit of sound in a language. In the English word *non-magistrates*, for example, there are three morphemes: *non*, *magistrate* and *s*. The study of morphemes and the rules which govern how they can be put together to form words in the language is known as morphology. In this dissertation, we will, in general, not be concerned with the morphological aspects of language. The Chinese forms which constitute the linguistic data in this thesis are all mono-morphemic, i.e. each form represents a single morpheme.

1.2.2 Contrast, Minimal Pairs and Distributions

Which phones in a particular language belong to the same phoneme? The answer to this is found by looking for contrast within the distribution of phones in the language. Two phones are said to contrast and therefore belong to distinct phonemes if switching one for the other in a word changes the meaning of that word. For example, in English, [v] and [f] contrast because *van* and *fan* are two words meaning different things. A minimal pair is a pair of words in a language which have different meanings but are phonetically identical except for the phones in question. It is strong evidence of contrast. If even one minimal pair can be found in a language, the phones that are different between them can be classified as different phonemes.

A slightly weaker condition for classification into separate phonemes is the existence of what are called sub-minimal pairs. These are pairs of words which are different in meaning and are phonetically identical up to the immediate environment in which the distinguishing phone is present. For example, the pair

consisting of *that* and *thatch* in English is a sub-minimal pair when attempting to contrast the phonemes /ð/ and /θ/ because the immediate environment in which the phones [ð] and [θ] occur is identical in both cases: /#_____æ (word initial positions followed by the vowel /æ/). A minimal pair for contrasting the same phones is [i:θə] and [i:ðə] (*ether* and *either*).¹ An additional, or alternative, argument for identifying phonemes, then, can be constructed on the basis of several sub-minimal pairs in as many structurally different word positions as possible.

Two phones in a language are said to be in contrastive distribution if there exists a minimal pair or sub-minimal pairs that contrasts them. On the other hand, they are in complementary distribution if one phone is never found in the environment of the other and vice-versa. In Spanish, for example, the phoneme /d/ is always pronounced as the fricative² [ð] when it occurs between vowels, but as the stop [d] everywhere else.

To sum up, we see that in English, there are two phonemes /d/ and /ð/ as evidenced by the minimal pair *ether* and *either*. In Palauan, we have one phoneme /d/ which is realised as either [d] or [ð] in free variation (eg. [di:l] or [ði:l], both meaning “nail”). Finally in Spanish there is again a single phoneme /d/, which is, however, realised in complementary distribution — as [ð] between vowels, but as [d] elsewhere (eg. [dinero] “money” and [naða] “nothing”). The existence of complementary distribution thus for two phones is strong evidence that the two phones are actually allophones of a single phoneme whose pronunciation is determined by its phonetic environment. More relevantly, the existence of complementary distribution for two phones could be an indication of their possible common ancestry as we shall see in the section 1.3

¹Crowley et al. (1995, p.77), in fact, considers *ether* and *either* to constitute the only minimal pair in contemporary English contrasting the phones [ð] and [θ]. Since *thy* is now considered an archaic form, the minimal pair consisting of the words *thy* and *thigh* is not recognised.

²Fricatives are sounds that are produced by almost, but not quite, stopping the air flow through the oral cavity. As the air escapes during articulation, audible friction is heard, as for example during the pronunciation of [v], [f], [s] or [h] in English.

1.2.3 Language Change

Since the publication of Ferdinand de Saussure's book *Course in General Linguistics*, linguists have generally considered that two kinds of changes — diachronic and synchronic — affect phonemes in a language. Diachronic phonological changes are those that happen to a phoneme over a period of time as the language evolves. Synchronic phonological changes happen to a phoneme as it is instanced in the spoken vocabulary and grammar of the language at any one time. The distinction between diachronic and synchronic changes is not just concerned with the greater or lesser extent of range of time within which a language is being studied, but with the linguist's attitude towards the time dimension itself. In synchronic study the linguist does not consider time as a factor at all and effectively considers the current state of the language being studied to be ideally steady and unchanging. In diachronic linguistics a sequence of successive synchronic states is studied and the changes that happen to a language over time are inferred, analogous to the way a circle is inferred by studying a vast number of points on its circumference (Bynon 1983, p.1-2).

Allophonic rules are synchronic in nature. The cumulative effect of diachronic changes over time and space to a language causes it to diversify, initially into dialects and eventually into one or more independent languages.³ It is known that diachronic changes are regular in nature and typically affect whole families of phonemes.⁴ Consequently it is possible to group together several related diachronic changes resulting from the same kind of transformation as instances of operation of a single rule. Such rules are called diachronic rules. Diachronic rules that deal with phonemes in the language are called diachronic phonological rules and their study is called diachronic phonology.

In contrast to synchronic rules, diachronic rules are always retrospective in nature. It is possible to predict the outcome of an application of a synchronic

³This raises a question: When does a dialect cease to be part of the language it developed from? In this the definition in Dyen et al. (1992, p.9) is followed — Two dialects (or idiolects) belong to the same language if they are connected by a chain of pairs of mutually intelligible dialects, without geographical constraints.

⁴Allophonic rules also often apply to families of phonemes.

rule. But with diachronic rules, the aim is really to derive the rule from an observation of the initial and resultant phonological forms of words (hereafter referred to as simply *forms*). Sometimes, it is possible to postulate and reconstruct a hypothetical ancestor for a language or a group of languages by studying similarities between them. This hypothetical language is called a protolanguage. A protolanguage is related to its descendants through a body of diachronic rules. Forms in the offspring language should ideally be derivable from corresponding forms in the protolanguage through the application of one or more diachronic rules, except where borrowing has occurred.⁵ One of the main tasks of the historical linguist is to attempt to reconstruct this hypothetical ancestor and explain how the descendant languages have reached their present form by postulating diachronic rules (Bynon 1983, p.45).

In subsequent sections, we call the sequence of steps in obtaining the offspring form from the parent form a diachronic phonological derivation or, where the context serves to disambiguate, simply a derivation. Linguistic forms in descendant languages that have been thus derived from a common original form in the protolanguage are called reflexes of this original form. These reflexes are then said to be cognate with each other.

1.2.4 Bleeding and Feeding

Many sound changes are conditioned by the phonetic environment in which they occur. Because of this, it is often possible not only to reconstruct the sound changes themselves, but also the order in which they happened.

When the result of operation of a rule results in a phonetic environment which triggers another rule to apply, the first rule is said to feed the second rule. On the other hand, if a rule destroyed the phonetic environment for the operation of another rule, then it is said to bleed the other. In the example in Table 1.1 from Chen and Newman (1985, p.134) that derives the Modern Cantonese word for “mouth” from its ancestral form, the rule X-LABIALISATION

⁵It is, in fact, possible in the majority of cases to distinguish a native word in a language from a borrowed or loan word. See for example Thieme (1964, p.590–591)

would labialise the /x/ in the environment of a labial vowel. But the DIPH-THONGISATION rule bleeds it by operating earlier and removing the labial environment. The “-” in this case indicates that the X-LABIALISATION rule was prevented from applying.

mouth	
k ^h u	Ancestral form
xu	SPIRANT
xəu	DIPH
-	X-LABIALISATION
həu	X-WEAK
hau	IO-FLIP
/hau/	Modern Cantonese form

Table 1.1: Example illustrating bleeding and feeding relationships among phonological rules.

Likewise, the SPIRANT rule which converts an aspirated /k/ into an /x/ feeds the X-WEAK rule which converts an /x/ into an /h/.

1.3 Language Reconstruction

In the previous section, we discussed the term *protolanguage* and how it is related to its descendants through a body of diachronic rules. Here we look at how the protolanguage itself could be reconstructed in the absence of any written records of it whatsoever. Two methods are widely used by linguists for this purpose. They are known as the comparative method and the method of internal reconstruction.

1.3.1 The Comparative Method

The idea of *regular sound correspondences* is central to the comparative method. When two languages exhibit a degree of similarity such that a certain phone in one is always instanced as a certain (possibly different) phone in the other, the correspondence between the two phones is called a regular correspondence. Early historical linguists (belonging to the so called neogrammarian school)

claimed that language change is governed by the principle of *the regularity of sound change* (Bynon 1983, p.25). That is,

- Sound change is independent of morphological, syntactic or semantic properties of words and depends only upon their phonetic properties.
- Sound change is exceptionless, and affects all phones falling within its scope.

Exceptions were in fact, initially found to this proposed regularity of sound correspondences, but many of these exceptional phones were quickly discovered to be due to extraneous factors such as analogical change, spelling, pronunciation or borrowing (Crowley 1987, pp.129–135). The rest were found to be in complementary distribution. Thus, the neogrammarians argued, a single rule could still be held to apply *exceptionlessly*, but its output would now be considered conditioned by its operating environment. A single sound in the protolanguage could have changed into more than one sound in a descendant language depending upon its conditioning environment. Such a change is called a conditioned sound change. The result, invariably, is the production of sound sets in which the sounds are in complementary distribution.

The comparative method, which originated in this neogrammarian school, works by treating regular sound correspondences as reflexes of forms in the protolanguage. It then postulates the reconstruction of these forms in the protolanguage by working out which original form these reflexes could have *reasonably* descended from. It seems that there is a great deal of leeway in the process because of the use of the non-specific word “reasonably”. But in fact the process is much more precise than the previous statement seems to imply. To be sure, there is a component of guesswork involved in this reconstruction, but Crowley (1987, p.89) calls it *intelligent guesswork* as opposed to *blind guesswork*. Various factors are considered in the process of reconstructing proto-forms. Important among these are descriptive economy (Bynon 1983, p.47), plausibility and elegance (Harms 1990, p.314), knowledge of what kinds of linguistic changes are likely and what are unlikely (Crowley 1987, p.90), and

in the case of Chinese, insights of the “Chinese philological tradition” (Newman 1987). These heuristics help to address some of the criticisms levelled at the neogrammarian model. Further, the reconstruction is a scientific hypothesis much like any other scientific hypothesis and is amenable to empirical testing where it is possible to find such data as would corroborate the claim (Thieme 1964, p.585).

1.3.2 Internal Reconstruction

The method of internal reconstruction is similar to the comparative method, but reconstruction is now based on evidence within a single language, rather than from several separate languages (Crowley 1987, p.149). Since it operates with a much smaller inventory of data, it does not take us as far back in time as the comparative method. Nevertheless, there are situations in which it is more suitable, as for instance, in the case of linguistic isolates like Basque, which have no known relatives.

It makes the assumption that language complexity increases with time due to the accretion of features which place cognitive loads upon the speaker. It also assumes that changes are more likely to happen at unprotected ends like word boundaries rather than within words. Thus, for example, in the case of Samoan, where present day transitive verbs have a variety of suffixes (-*mia*, -*sia*, -*tia*, etc.), we can postulate a single suffix -*ia* in pre-Samoan and assume the preceding consonant to actually be part of the reconstructed verb root (Ibid. p.152).

As another example illustrating internal reconstruction, we could consider the German singular forms (Ibid. p.154): /laut/, /bo:t/, /ta:k/ and /hunt/ (“sound”, “boat”, “day” and “dog”), which have as plurals /laut-e/, /bo:t-e/, /ta:g-e and /hund-e/ respectively. We note that some plurals have voiced consonants where the corresponding roots in the singular forms had unvoiced consonants. Since there is additional complexity involved in explaining this unexpected correspondence, we assume that the plural roots represent the original ones and that the singular roots had undergone change. We thus postulate the

rule:

$$C \rightarrow [-\text{voiced}] / ___ \#$$

which devoices word-final voiced consonants.⁶ The consonants in the plural would have been protected from this rule by the presence of the plural suffix.)

1.4 Language Similarities and Relatedness

The idea that there exist common ancestries for today's different languages is at least as old as Trombetti (1905)'s *L'unità d' origine del linguaggio*. However, the idea of long range comparison between widely separated languages remains contentious although this contention is challenged from time to time, for example by Greenberg (1987) and more recently by Greenberg (1991), Shevoroshkin and Ramer (1991) and Ruhlen (1991).

Some of the issues relating to this topic are covered in Austerlitz (1991) as are some of the methodologies used. An idea common to many of them is the derivation of a degree of relatedness between languages. Commonly, a similarity or dissimilarity measure between languages is used as a rough guide to their relatedness. An implicit assumption here is that more similar languages are more strongly related to each other. This is by no means a proven fact, but it serves as a reasonable heuristic in drawing up genetic relationships among languages and subgrouping languages and language families. It is, of course, possible that language similarities exist due to chance. But it is claimed that one can estimate the degree of such chance similarities (Ringe 1992) and therefore incorporate this into similarity measures derived otherwise.

Thus this dissertation is primarily concerned with finding such a dissimilarity measure between languages, which for the first time, incorporates the complexity of the protolanguage hypothesis into the distance measure. The method was initially proposed in Raman and Patrick (1997c). Subsequent work which

⁶In other words, this removes the "voiced" feature from consonants which are in word final positions. The notation followed in describing the rule is a somewhat "relaxed" form of the notation in Chomsky and Halle (1968).

we will discuss in the chapters to come involves modifications and refinements to this basic methodology, an illustration of which can be found in Chapter 3. Chapter 2 reviews some similar work in linguistics that has been pursued along these lines for some decades now.

Chapter 2

Quantitative Methods in Historical Linguistics

2.1 Early Work

Mathematical methods seeking to quantify one or more aspects of languages are not particularly new. Embleton (1991) describes many earlier approaches along these lines. For example, in Kroeber and Chretien (1937) we find a quantitative classification of Indo-European languages based on a technique used in ethnography and physical anthropology. The degree of association, Q , between two languages L_1 and L_2 is given by

$$Q = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

where a is the number of features exhibited by both, d is the number of features exhibited by neither, b is the number of features exhibited by L_1 and not L_2 and c is the number of features exhibited by L_2 and not L_1 . The above formula is called the tetrachoric correlation coefficient or Karl Pearson's tetrachoric R . The values of Q thus computed allow us to rank the closeness of relationship between L_1 and L_2 . Obviously, the presence or absence of the features under consideration must be unambiguously decidable and it is of importance for statistical significance that the features also be independent of each other. This, coupled

with the additional problem of reconstructing features in the protolanguage in order to determine the value of d , beset the method with practical and conceptual difficulties and other methods were suggested to overcome these problems. Ross (1950), for example, describes a related technique, which doesn't require the parameter d in the tetrachoric correlation coefficient.

Hierarchical cluster analysis, a method popular in numerical taxonomy, has also been applied, for example, in the classification of African Bantu languages (Henrici 1973). This reconstructs a family tree for a set of languages based upon the number of cognate and non-cognate items each has in a given word list. The use of multidimensional scaling in this area has also been proposed (Dobson and Black 1979; Black 1976). A common thread that links several of these methods is the use of cognate counts to derive a degree of relatedness. Cognates are words in two or more languages which have directly descended from a single lexical item in the protolanguage (Bynon 1983, p.47). Since these techniques employ the vocabulary of the language to derive similarity measures and employ the lexicon of the language in one form or another, these are collectively called lexicostatistical methods.

Lexicostatistics is the statistical study of vocabulary for historical inference. Although lexicostatistics tends to be often confused with a related field called glottochronology, the two are essentially distinct fields. Glottochronology, which is described in more detail in Section 2.2, is aimed at coming to conclusions about the temporal relationships between languages. It may or may not be lexicostatistical in nature. Since glottochronology based on rate of change in sectors of language other than vocabulary (eg. syntax or phonology) is conceivable, lexicostatistics and glottochronology are best conceived as intersecting fields (Hymes 1960, p.4). However, the most common applications of glottochronology in the past, as well as its initial exposition have used language vocabulary, and this is the source of confusion between the two. Thus, it could be said that lexicostatistics and glottochronology are in principle intersecting fields, but in practice, glottochronology tends to be a subset of lexicostatistics. It should be remembered, also, that there are many lexicostatistical methods

which are strictly atemporal, for example those of Ellegård (1959) and Gleason (1959) which were not designed for coming to conclusions regarding the chronologies of linguistic separations.

2.2 Glottochronology

Glottochronology is the study of rate of change in language, and the use of the rate for historical inference, especially for estimation of time depths and the use of such time depths to provide a pattern of internal relationships within a language family. The technique itself was popularised by Morris Swadesh in the early 1950s. Stimulated by the achievements of the radiocarbon dating technique in archaeology, Swadesh undertook studies of rate of vocabulary change. In Swadesh (1952), he suggested that it was possible to date the separations of languages by incorporating the idea of radioactive decay into the process of vocabulary turnover. This initial exposition of the technique made it so popular that it became perhaps the best known lexicostatistical method.

However, in spite of the occasional arguments in its favour, some as recent as Rea (1990), glottochronology seems to have largely fallen into disrepute now, but as expounded originally by Swadesh, it still remains one of the earliest mathematical methods that attempted to date language separations with a reasonable degree of success. It is noteworthy, however, that its origin goes back much earlier than 1951. It is likely that Swadesh himself developed it from a suggestion originally made by his teacher Edward Sapir:

The greater the degree of linguistic differentiation within a stock, the greater is the time that must be assumed for the development of such differentiation. (Sapir 1916)

Hymes (1971) traces anticipations of this technique even further back to Captain J Dumont d'Urville (1790-1842), a French explorer who chanced to come upon the work of a certain self-styled professor called Constantin Rafinesque (1783-1840). d'Urville was part of a committee to decide the best paper for a competition on the topic of "The origin of the Asiatic Negroes". Rafinesque

had mailed d'Urville a paper titled "Languages do not lie" and a subsection of it outlined

a procedure by which to appreciate at a glance the diverse degrees of identity and affinity that can be established between two languages, of which a certain number of words common to both are known.

Here then, we have one of the earliest foretastes of the glottochronological technique. Its subsequent refinement by Swadesh had led to some degree of success in dating language separations, which caught the attention of a good many contemporary linguists. Sarah Gudchinsky, who has given one of the clearest expositions of glottochronology, hailed it as novel and useful (Gudchinsky 1956).

2.2.1 Description

Essentially, the glottochronology of Swadesh, as far as it concerns vocabulary, makes the following basic assumptions (Swadesh 1951):

- Some parts of a vocabulary (core-words) change less rapidly than others
- The rate of replacement of words in this core-word list is constant through time.
- The rate of loss of items from this list is the same for all languages.

Given these assumptions, it is reasonably easy to derive the relation in Lees (1953):

$$t = \frac{\log c}{2 \log r}$$

where t is the time depth of separation in millenia, r is the rate of retention of words in any core-word list (assumed constant at 85% per millenia) and c is the percentage of common cognates in two given core-word lists. Such core-word lists, which are necessary for the technique, have been variously compiled for a number of languages. As proposed by Swadesh, the original word list was 200

items long.¹ But its length and contents have undergone significant revisions. The idea behind using a core-word list is that items in this list are chosen to include a so called *basic vocabulary* of the language, consisting of items that are most likely to be native to it and least likely to be replaced by borrowing and other haphazard events. We will not discuss this point further, but try to clarify the basic technique by means of an example in Section 2.2.2.

2.2.2 Example

Let us assume that the five-word lists in Table 2.1 are used for English and German: The English words *four*, *I* and *sun* in the table have cognates in

English	German	Score
animal	Tier	×
four	vier	✓
head	Kopf	×
I	ich	✓
sun	Sonne	✓

Table 2.1: An example illustrating common cognate scores for glottochronology.

German, but *animal* and *head* haven't. To be sure, there might exist cognates for those words also in German, but their meanings may have diverged since the two languages separated — The list can only have words with a common origin whose meanings also are the same. The score is based on the number of such diverged words in a basic vocabulary for the two languages. In the example being considered, we can see a 60% agreement in the two word lists — 3 out of 5 words have cognates whose meanings haven't changed. This allows us to calculate

$$t = \frac{\log 0.6}{2 \log 0.85} = 1.57159$$

¹It has been claimed that a list of 200 words is optimal for glottochronology since shorter lists are clearly less accurate while longer lists tend to offer little increased accuracy in return for much greater labour. See Ringe (1992, footnote, p.7) and Embleton (1986, pp.43–45,66–67,89–93)

or 1.572 millenia, i.e. German and English started diverging from each other approximately 1572 years ago, around 425 AD! The example presented above is contrived and small, but it sufficiently illustrates the basic principles behind glottochronology and the procedure involved in coming to conclusions about the time of separation of languages. Needless to say, glottochronology was considered a remarkable technique for a time, but its premises are not indisputable and it has come under heavy criticism from many quarters (Chrétien 1962; Hoijer 1956).

2.2.3 Criticisms of glottochronology

A devastating critique of its fundamental postulate of the universality of replacement rates, i.e. that the rate of change of items in the basic vocabulary is the same for all languages, was published in Bergsland and Vogt (1962) and Chrétien (1962) strongly criticised the mathematics originally examined by Lees (1953). But several other approaches have been more forgiving, including Dyen (1971) which sought to defend glottochronology against Chrétien (1962) in particular. More relevant to this work are criticisms levelled at the technique which seek to modify one or more aspects of it to make it more acceptable. Among these may be included Ellegård (1959), Gleason (1959) and Sankoff (1972) which sought to modify the mathematics involved. Ultimately, this resulted in a revised formula which incorporated stochasticity into the model and allowed for chance cognation and borrowing into the test list. Embleton (1981) is also concerned with incorporating borrowing of items into the original core-word list. But glottochronology was now deviating more and more from its original simple formulation and it is noteworthy that Sankoff observed that more complex models offered little or no predictive advantage over the simpler ones. Thus a fallback to earlier proposals or the introduction of fundamental revisions was inevitable.

2.3 The Direction of This Thesis

The technique described in this dissertation, which is illustrated in Chapter 3, overcomes the most significant objection levelled at glottochronology — that of the non-universality of replacement rates — since a chronological thesis, if one is made, will be based on a dissimilarity measure rather than on the total loss from a core vocabulary. The other important contribution made by this method is that of pioneering the use of phonological data to establish dissimilarity measures, as opposed to lexical (which has been used most frequently until now), semantic or syntactic data.² Cheng (1982), for example, attempts to address exactly this topic, that of obtaining distance measures between related Chinese dialects. However, he does this at the lexical level by using Karl Pearson's tetrachoric R on 905 words from a lexical dictionary called the Cihui (1964) rather than at the phonological level.

An argument can also be made that phonetic or phonological dissimilarity measures, being the least abstract of all, could give the most realistic results. Unfortunately, studies in this direction have been relatively rare. Two such works which should be mentioned are Grimes and Agard (1959) and Hsieh (1973), both of which are, however, constrained by the use of lexicostatistical methodology. In fairness to existing methods, it must be noted that many other existing methods for obtaining dissimilarity measures are in fact applicable to non-lexical data for deriving non-lexical measures. In practice, though, they have been constrained by a preoccupation with the lexicon as well as by the unavailability of phonological data.³ Hopefully, the phonological data developed in this project should provide fresh input to those methods and revive their application to the problem area in future research.

²Indeed, semantic similarity, which is usually necessary for the identification of cognates in Indo-European languages, is not even relevant in the case of the Chinese languages we are concerned with in this project because cognates can be visually identified in Chinese languages due to a common ideographic writing system stretching back over 3 millenia (Streeter 1977, p.103).

³This was also pointed out by Dr Sheila Embleton, York University, Toronto in a personal communication: Comment on using a phonological dissimilarity measure. In email correspondence dt. 9 Oct 1994.

Chapter 3

Similarity Measures – An Example

3.1 Overview

Suppose we want to encode the degree of relatedness between a parent language L_1 and a daughter language L_2 , one method will be as follows: we consider a message encoding two sets of words - W_1 and W_2 , one from each language. Let the words in each set be $\{x_1 \cdots x_n\}$ and $\{y_1 \cdots y_n\}$. Where a derivation d_i exists between x_i and y_i , it will be possible to code y_i simply as d_i with a flag indicating the fact that it is coded as a derivation. If the derivation d_i occurs sufficiently frequently, as is the case with regular correspondences, then it can be coded concisely and the resultant encoding of the two word sets will be smaller than if the actual words were specified. Thus, the conciseness of coding W_1 and W_2 will be indicative of the degree of relatedness between L_1 and L_2 . Alternately, as is done in this chapter, a Probabilistic Finite State Automaton (PFSA) could be constructed that encodes these sequences of derivations. These are state determined machines which have stochastic transition functions. If this is the case, then the size of the minimal description of this PFSA would be indicative of the distance between the parent and daughter languages. Chapter 5 describes PFSA in more detail. For now it will suffice to note that there is a method by which we can, from the data, construct what are called canonical PFSA

which are direct encodings of the data, make generalisations to the structure as represented by them, and compute the size of describing such PFSA minimally. For consistency with the terminology from later chapters, we will call the length of their description the MML or Minimum Message Length of PFSA.

In comparing closeness between the languages in two or more language pairs, the length of the message encoding the words in the parents will cancel out provided the same set of words is used. Thus the significant factor in comparison will only be the derivations where they exist and the actual words in the daughter language where derivations could not be found.

The following two examples demonstrate the application of the method to the two language pairs, Later North-West Germanic and Old Frisian and Later North-West Germanic and Old High German. In the following sections, Later North-West Germanic is referred to as *LNWG, Old Frisian as OF and Old High German as OHG for brevity. The “*” in “*LNWG” indicates its special (protolanguage) status. The *LNWG, OHG and OF words and the rules for deriving them are all taken from Voyles (1992) which is hereafter referred to as Voyles92. The reason that Voyles92 in particular was chosen for the protoforms and diachronic rules was because it contained a reasonably coherent and self contained set of derivations. It was originally intended to use a hypothetical language to illustrate the methodology, but later on, it was decided to pick a random sample of words from Voyles92 so the example would at the very least look realistic. However, it must be stressed that no general conclusion can be drawn from any results derived in this chapter because the sample is inadequate and unrepresentative of the actual proportion of phonological categories in the languages. The data is thus best treated as fictional, intended purely to illustrate the methodology at this stage, although with modifications, corrections and expansion, it may produce more significant results.

3.2 Modelling *LNWG to OF

Table 3.1 lists a number of diachronic phonological rules that apply to Later North-West Germanic words of about 400 A.D. to derive corresponding Old

Frisian words of about 1200 A.D. The rule numbering used by Voyles92 is retained. The numbering in the table may not be sequential as only the rules that are actually used in the example have been given. Each rule label is followed by the corresponding rule's number, its name as given by Voyles92, and then the statement of the actual rule. In the occasional event when no rule in Voyles92 was found to account for a change that was found necessary, a tentative one was created as a stop-gap measure. Such rules are designated with lower-case labels (a, b, c, etc.). The table is only intended as a rough guide. See Volyes92 for a more comprehensive discussion of each.

Label	No.	Name	Rule
A	7.1.1	ð to d	ð → d
D	7.1.4	ã to ò	[ã +str] → ò
E	7.1.5	z-dele	z → ∅ / __##
G	7.1.7	a to e	[a +str] → e /closed syllable
H1	7.1.8	ai/au change	ai,au → æ,ô /open syllable
H2	7.1.8	ai/au change	æ,ô → ā; ā → ē / __Ci(:), j, closed syllable
H3	7.1.8	ai/au change	ai → æ/closed syllable
H4	7.1.8	ai/au change	æ → ā / __x,p,b,m or w
H5	7.1.8	ai/au change	ē → ē
I1	7.1.9	i/j umlaut	[V +str+low] → [-bk,-low] / __Ci(:),j
I2	7.1.9	i/j umlaut	[V +str] → [-bk, -low] / __Ci(:), j
I3	7.1.9	i/j umlaut	a(:),o(:),u(:) → e (ē), e (ē), i(:) / __Ci(:), j
I4	7.1.9	i/j umlaut	a(:), o(:),u(:) → e, ē / __Ci(:),j
J	7.1.10	Palatalization	k → c, g → j /## __[V,-bk]
L	7.1.12	Unstr.short.V dele	[V -long -str -nasal] → ∅ /C [__-high]##
N	7.1.14	Unstr.V short	[V +long -str -nasal] → [-long] / __##
R	7.1.18	Obstr.Voi/Devoi	[C + obs +cont] → [-voi] / __[-voi or ##]
Y	7.1.25	n-dele	opt. n → ∅ /([V -str] __##
Z	7.1.26	r-metathesis	r [V +str] → [V +str] r /C [__[C +coronal or xC]
a	-	(ad hoc - 1)	[a +str] → e /open syllable
b	-	(ad hoc - 2)	ò → ā / __##
c	-	(ad hoc - 3)	b → f
d	-	(ad hoc - 4)	j → i

Table 3.1: Diachronic rules for converting *LNWG → OF.

The rules in Table 3.1 were used to derive OF forms for about 20 *LNWG words. The *LNWG and OF words, with their English translation, and the

sequence of rules deriving each OF word from the corresponding *LNWG word is given in Table 3.2. Initially, it was assumed that the derivations were more likely to be correct if the words chosen were from the Swadesh basic vocabulary list for the languages. However, it was pointed out by Dr Peter Christian of Goldsmith's College, London, who reviewed this work before its publication in Raman and Patrick (1997c) that whether or not the words come from the Swadesh list is irrelevant. What the Swadesh list guarantees is only that the words in it are less likely to be semantically different from their cognates. The method described here only needs a list of words in one language and their cognates in another, regardless of their semantic differentiation. For instance, if the word "dog" was part of Swadesh's basic vocabulary, the pair of words *dog* and *hund* which are not cognates, would be listed for English and German. However, since semantic agreement is irrelevant to this work, we could use the pair *hound* and *hund* instead which are indeed cognates. Thus they would have a common ancestor in a protolanguage from which we can trace their development.

The sequences of diachronic rules that are applied to the *LNWG words to transform them into OF words are given in the column labeled *Path* in Table 3.2. Colons separate the individual rules in each sequence. A colon by itself indicates that no diachronic rule was applied to derive the final form from the initial. Obviously, the order of application of rules is significant in the determination of the paths. The order set out in Voyles92 has been followed wherever possible. Where such an order was not specified, the rules are applied in the order they appeared in the work.

A Probabilistic Finite State Automaton (PFSA) derived from the above data is given in Figure 3.1. It can be interpreted as a graphical representation of the evolution of words in OF from *LNWG. The labels on the arcs indicate the rules that were applied in transiting from one state to another. Their frequencies are indicated in superscript font above the symbol, except when there is more than one symbol on an arc, in which case the frequencies are denoted by the superscript marker "^". Exclamation marks ("!") indicate arcs

No.	English	*LNWG	OF	Path
1	father	faðder	feder	A:a:
2	moon	menō	mōna	D:b:Y:N:
3	stone	stainaz	stēn	E:H1:L:H2:
4	staff	stab	stef	G:c:
5	field	akker	ekker	G:
6	lamb	lamb	lamb	:
7	night	naxt	naxt	:
8	apple	appel	appel	:
9	bone	bain	bēn	H3:H2:
10	tree	baum	bām	H1:H2:
11	bed	badja	bed	I1:d:L:L:
12	gap	klufti	kleft	I2:I3:I4:L:
13	skin	hūdi	hēd	I2:I3:I4:L:
14	power	maxti	maxt	L:
15	day	daga	dei	J:L:G:d:
16	guest	jesti	jest	L:
17	word	wordu	word	L:
18	brother	brōther	brōder	R:
19	man	man	man	:
20	grass	gres	gers	Z:

Table 3.2: The sequences of diachronic rules to derive 20 words from *LNWG to OF.

on delimiter symbols to state 0 from the state they superscript.

3.3 Modelling *LNWG to OHG

As in the previous section, Table 3.3 lists the diachronic phonological rules that were used in deriving words in the Eastern Franconian dialect of Old High German dating from about 830 A.D. from the Later North-West Germanic of about 400 A.D.

Table 3.4 shows the derivation of the 20 OHG words from the corresponding reconstructed *LNWG words. As before, the sequences of diachronic rules that are applied to each *LNWG word to transform it into OHG is given in the column labelled *Path* and colons separate individual rules within each sequence.

The PFSA derived from the data in Table 3.4 is shown in Figure 3.2. It can

Label	No.	Name	Rule
A	9.1.1	đ to d	đ → d
C	9.1.3	ai/au changes	ai, au → ē, ö
D	9.1.4	Str. ē/ō-to-ie/uo	[ē, ö +str] → ie, uo
E	9.1.5	z-dele	z → ∅ / [V, -str] ___ ## or [V -bk +str] ___ [+obs]
F	9.1.6	Unstr. short V dele	[V -long -str -nas] → ∅ / ___ [-high] ##
I	9.1.9	Unstr V short	[V +long -str -nas] → [-long] / ___ ##
J	9.1.10	WG Gem	C → CC (C ≠ r, θ, z) / [V -long or -str] ___ j p, t → pp, tt / [V -long] ___ l or r k → kk / [V -long] ___ r or w
K	9.1.11	2nd SS	[C +obs -cont -voi +asp] → pf, tz / ## /or C [___ -bk] [C +obs -cont -voi +asp] → ff, zz, xx / V ___
L	9.1.12	i/j umlaut	[V +low +bk -long +str] → [-low -bk] / ___ Ci(:), j
O	9.1.15	h, g to b, g	h, g → b, g
P	9.1.16	d to t	d → t / ___ ##, in gemination, [+son -voc] ___ V, opt. V ___ [+son] or opt. ## ___
T	9.1.20	j to i	j → i / C ___ V
W	9.1.23	q to d	θ → d / opt. ## ___ or obligatory
a	-	(ad hoc - 1)	[ō-str] → ö
b	-	(ad hoc - 2)	e → a
c	-	(ad hoc - 3)	e → u
d	-	(ad hoc - 4)	j → g

Table 3.3: Diachronic rules for converting *LNWG → OHG.

be interpreted as a graphical representation of the evolution of words in OHG from *LNWG. See the discussion regarding Figure 3.1 at the end of Section 3.2 for a fuller description of the symbols used.

3.4 Discussion

In the examples using Old Frisian, Old High German, and reconstructed Later North-West Germanic, we see that the length of a message to describe hypothesis to explain the derivation of 20 words from *LNWG to OF is 314.67 bits, whereas that required to explain a similar derivation from *LNWG to OHG is only 301.84 bits. The message lengths thus computed represent the complexity of the derivation process. There are intuitive grounds for believing that it

No.	English	*OHG	OF	Path
1	father	fader	fater	A:P:
2	moon	menō	māno	a:I:b:
3	stone	stainaz	stein	E:C:F:D:
4	staff	staḅ	stab	O:
5	field	akker	akkar	b:
6	lamb	lamb	lamb	:
7	night	naxt	naxt	:
8	apple	appel	apful	K:c:
9	bone	bain	bein	C:D:
10	tree	baum	boum	C:D:
11	bed	badja	beti	L:T:F:P:J:
12	gap	klufti	kluft	F:
13	skin	hūdi	hūt	F:P:
14	power	maxti	maxt	F:
15	day	daḡa	tage	F:P:O:
16	guest	jesti	gast	F:b:d:
17	word	wordu	wort	F:
18	brother	brōer	bruoder	W:
19	man	man	man	:
20	grass	gres	gras	O:b:

Table 3.4: The sequences of diachronic rules to derive 20 words from *LNWG to OHG.

will be less complex to derive words between languages that are close to each other than between those that are distant. Thus a greater message length could indicate a greater dissimilarity between the two languages.

In the example given in this chapter, it happens that the computed message lengths are also consistent with the chronological order of separation of these languages from *LNWG as the OF words we used date from around 1200 A.D. and the OHG words from around 830 A.D. But we can draw no specific conclusion based on this small test sample because the words used were purely for the purposes of illustrating our method. Also, we cannot make a claim about the extent of chronological separation between the two languages. Whether there is a functional dependence between an MML of a PFSA and the amount of time the daughter language has been separated from its parent is questionable and needs to be addressed separately. This chapter has, how-

ever, described a framework within which sensible numeric figures indicative of language (dis)similarities can be derived.

The message lengths given are for what we could call canonical PFSA which directly encode a representation of the data in the *Path* columns of Tables 3.2 and 3.4. It is possible to make generalisations of these PFSA (for example, by merging states which have mostly similar outputs) and infer more general structures. The details of this are left till later chapters as is a discussion of the linguistic significance of such reduced PFSA. But generalisations of the two canonical PFSA are nevertheless presented here in Figures 3.3 and 3.4.

In each case, we see savings of about 50 bits in describing the generalised PFSA as compared to the canonical ones. In a following chapter, we discuss the possible significance of such savings and its implication for the linguistic hypothesis.

In this example, as in the rest of the dissertation, The Minimum Message Length (MML) method due to Wallace and Georgeff (1984) has been used to compute the size of PFSA. As explained in Chapter 5, MML is a quantized version of Bayesian inference. So everything that uses MML has a corresponding probabilistic interpretation. It is also important from the MML point of view to incorporate prior knowledge into the hypothesis as much as possible. In the case of the PFSA in this example, much caution has been exercised in the use of priors. In particular, no priors on the distribution of diachronic rules have been used that were not explicitly given by Voyles92. Doing so would be taking liberties with the linguistic data as made available to this project. The data that was used did not contain information that specified the probabilities of the various rules applying at each state. Thus, the message length computation assumes a uniform prior (equiprobable rules to start with), which may not be true, but is at least linguistically plausible. However, should such probabilities become available in the future, it would be a simple matter of incorporating them into the present framework.

A final point needs to be made regarding the motivation for the additional sophistication embodied in this method as compared to, say, a more simplistic

phonological approach like a distance measure based on a simple summation of the number of proposed rules. Our method not only gives a measure dependent on the number of rules, but also on the inter-relationship between them, or the regularity present in the whole phonology. A lower value indicates the presence of greater regularity in the derivation process. As a case in point, we may look at two closely related dialects, which have the same number of rules in their phonology from a common parent. It may be the case that one has diverged more by losing more of its original structure. As in the method of internal reconstruction, if we assume that the complexity of a language increases with time due to the presence of residual forms (Crowley 1987, p.150–153), the PFSA derived for the more distant language will have a greater MML than the other, thereby measuring dissimilarity or *distance*.

In later chapters, we will discuss the application of this method, albeit with some modifications, to construct PFSA for measuring the development of some 2700 forms in Modern Cantonese and Modern Beijing¹ from their reconstructed forms in the protolanguage – Simplified Middle Chinese.

¹Modern Beijing was formerly called the Peking dialect of Mandarin.

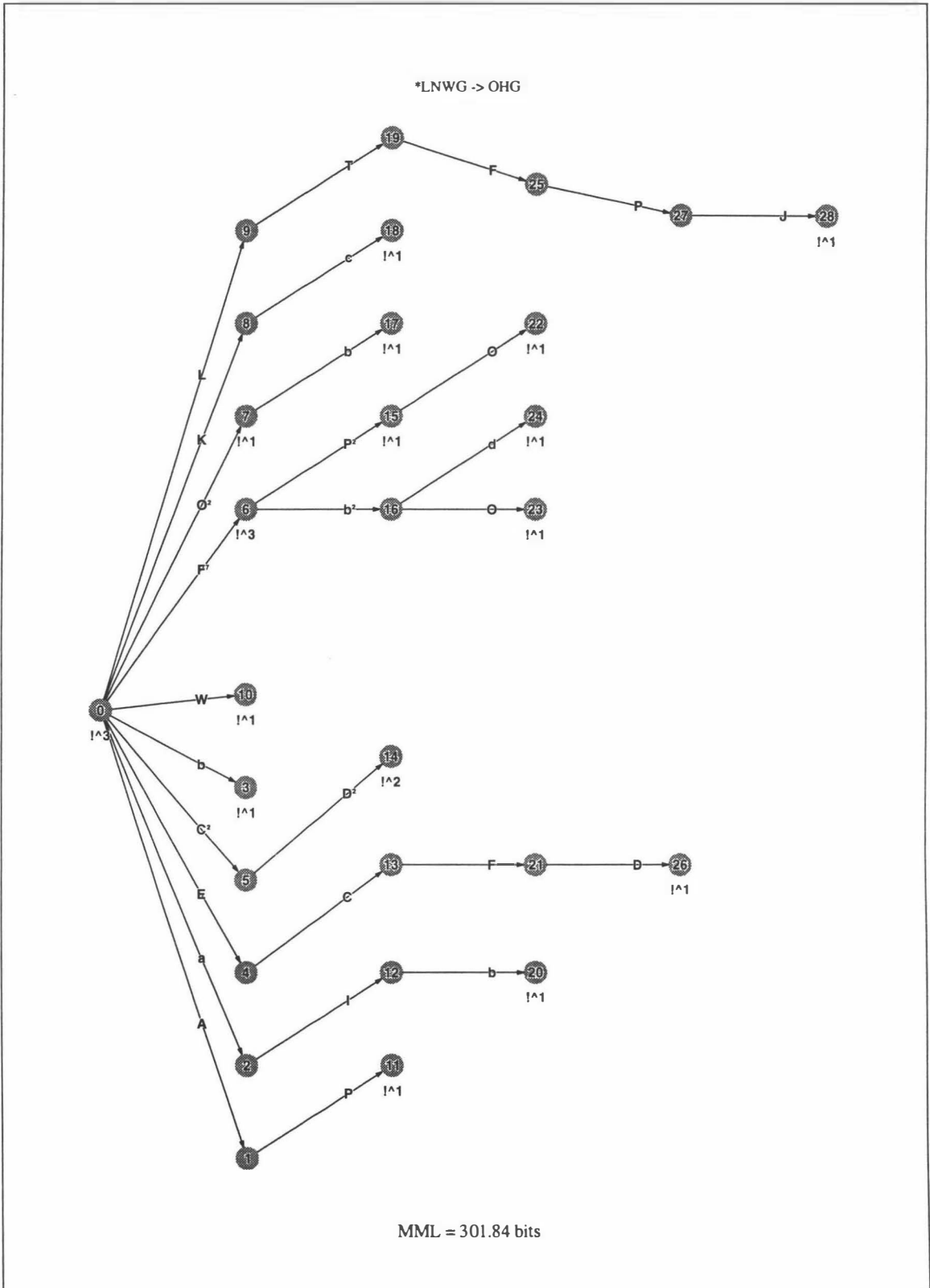


Figure 3.2: Canonical PFSA representing the derivation of 20 *LNWG words into OHG.

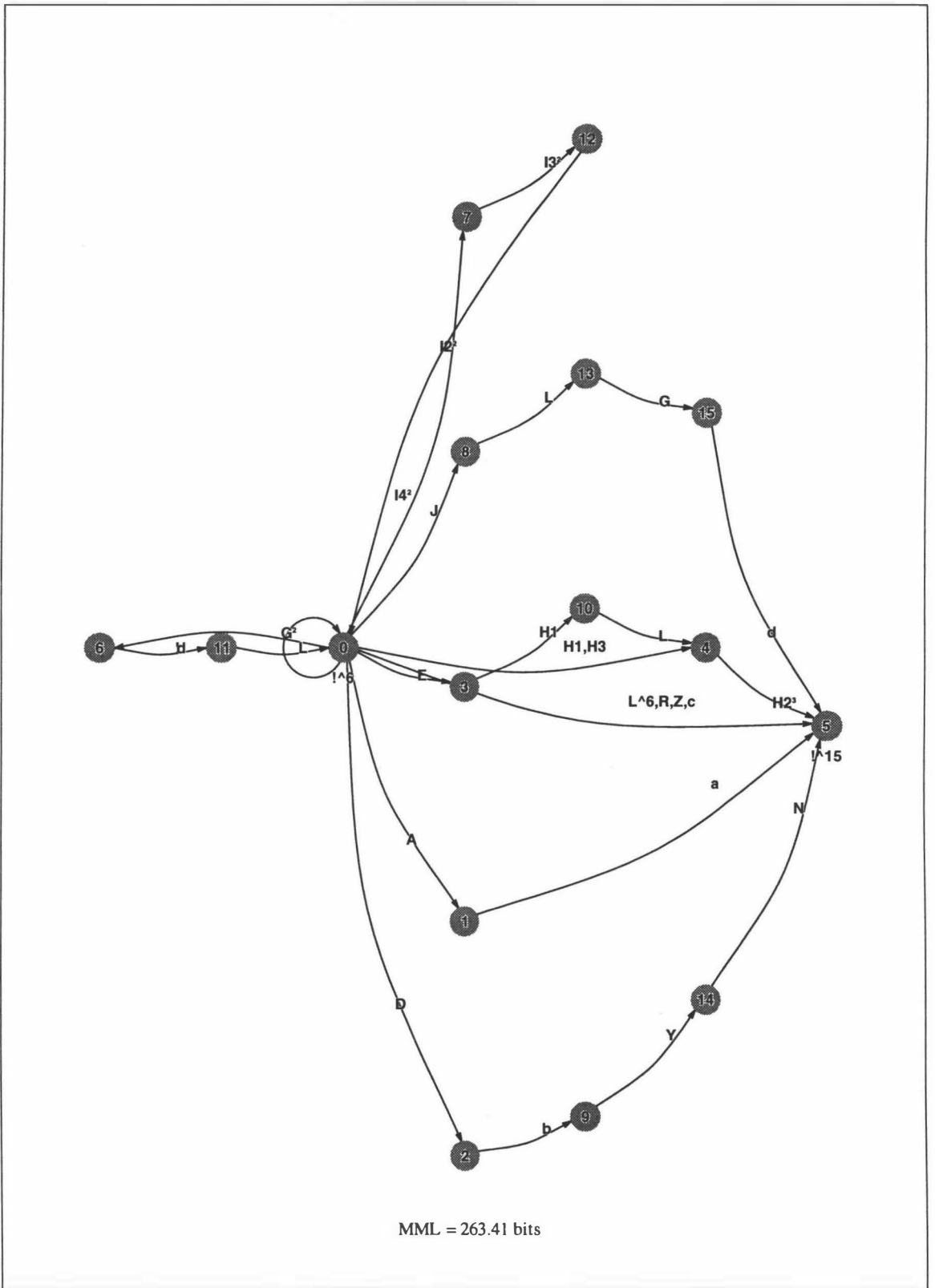


Figure 3.3: A generalisation of the PFSA in Figure 3.1 (*LNWG>OF).

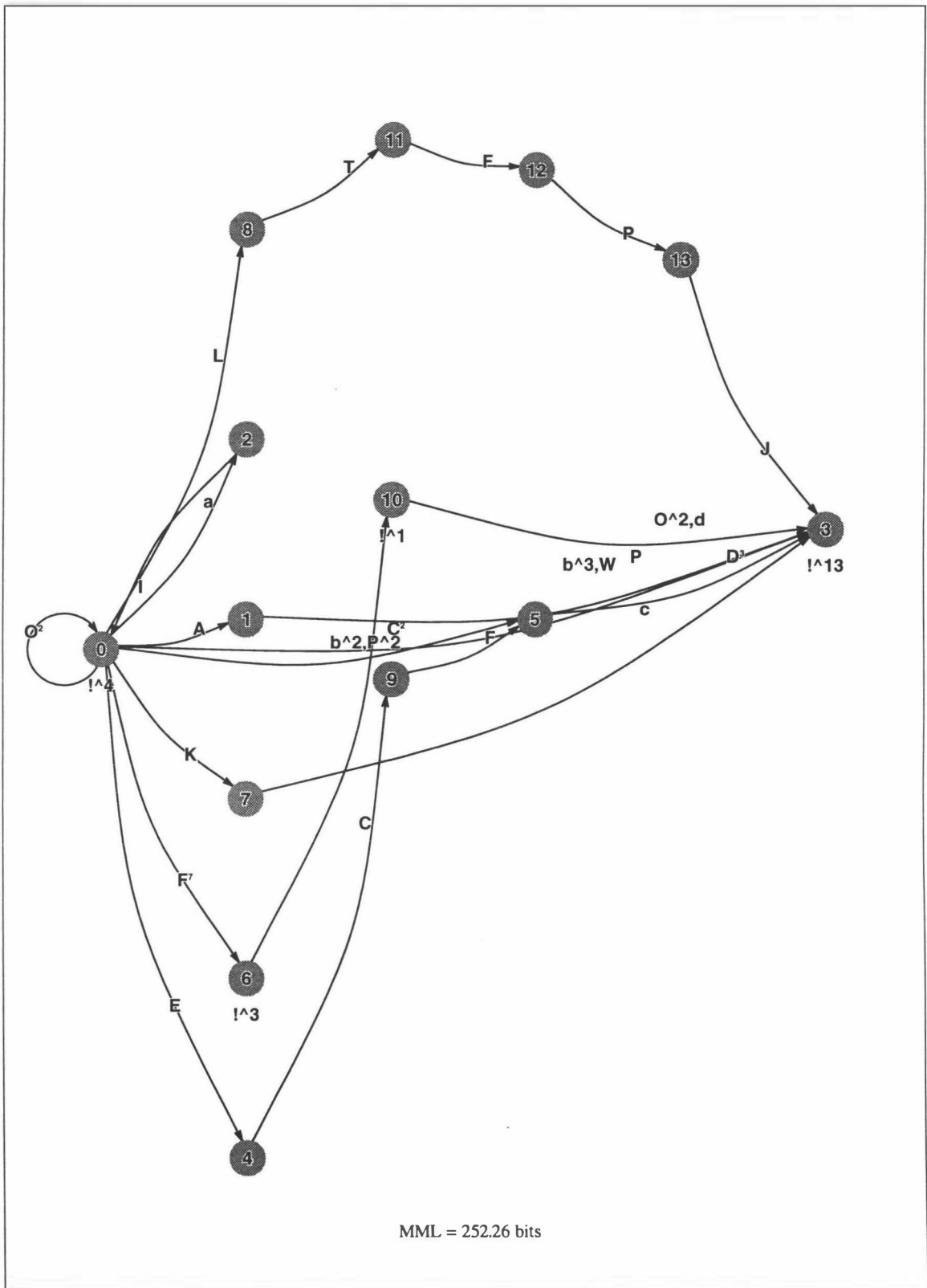


Figure 3.4: A generalisation of the PFSA in Figure 3.2 (*LNWG>OHG).

Chapter 4

Data Collection

4.1 Introduction

This chapter discusses the procedure undertaken to collect the linguistic data for analysis. A brief and informal discussion about the original source of data, its reliability and the motivation for using it is first presented. An overview of Matthew Chen's hypothesis concerning Simplified Middle Chinese (SMC) reconstruction is then given, followed by an outline of Matthew Chen's and John Newman's proposals for the diachronic phonology between SMC and Modern Beijing (MB) and Modern Cantonese (MC). In the following sections, Chen76 is used to refer to the Chen (1976) hypothesis for the diachronic phonology — *SMC>MB. CN84 is used to refer to the Chen and Newman (1984a, 1984b, 1985) hypothesis — *SMC>MC. A brief overview of the Relative Chronologies (RC) in Chen76 and CN84 is then given. Finally the procedure by which Chen76 and CN84 were used to produce and modify the raw data into a form directly usable in this project is described.

4.2 Original Sources and Reliability

In his book *Chinese*, Norman (1988, p.42), states

Chinese historical phonology hitherto has been the study of the development of the various stages of the literary language as codified

in traditional dictionaries.

Indeed, all scholarly work in historical Chinese phonology ultimately derives from the rhyming and pronouncing dictionary *Qie yun*, a single work compiled in AD 601 by a small group of scholars aiming at providing a guide to the proper recitation of literary texts (Arendrup 1994, p.523). The *Qie yun* mentions various earlier rhyming dictionaries in its preface, but they all seem to have been eclipsed by *Qie yun*'s status and prominence. The person principally responsible for its compilation seems to have been the poet *Lu Fayan* (Norman 1988, p.24).

4.2.1 The *Qie yun*

The *Qie yun*, which is considered for all practical purposes identical to and thus the primary source for the Middle Chinese language (Norman 1988, p.24,40ff) records the pronunciation of Chinese characters arranged by tone and rhyme. It is said to have made the innovation called *fan qie* or 'reverse cut', by means of which pronunciation of words without homophones could be specified. The *fan qie* is a method by which a syllable is broken down into its component parts — the initial consonant and the syllable final elements with the tone — and the pronunciation of each is given by citing a known syllable in which it occurs (Wang and Asher 1994, p.526). An example in English would be that of saying that the initial of *life* is pronounced the same way as in *light* and the final the same way as in *wife*. The scheme itself and its interesting history is discussed in more detail in three recent publications — Yiming (1990), Jianing (1990) and Wen (1995).

The exact nature of the language the *Qie yun* represents is the subject of some controversy. Bernhard Karlgren, one of the earliest and most prominent Western scholars of Chinese linguistics, held that it represented the language of the Sui dynasty capital, *Changan* (Karlgren 1954, p.212). However, most Chinese linguists of the present time accept Zhou Zumo's view (Zumo 1963) that it represents a compromise between the the literary pronunciation of north and south of that time (Norman 1988, p.24).

4.2.2 Reliability

One thing certain, though, is that the *Qie yun* represents the literary components of the language rather than popular ones. Popular forms are those which go back in an unbroken line to the protolanguage, while literary forms are those that at some point had ceased to be living words in the language (Norman 1988, p.41). In this, popular forms must be distinguished from colloquial forms, which refers to their contemporary status. The popular/literary distinction is of a historical nature, whereas colloquial refers more to synchronic usage. It is also incorrect to say that popular forms are those literary forms that have survived to the present day because literary forms may have ceased to exist at some point in the past, yet survived to the present day because of revival from texts, perhaps more than once. Thus a form in current usage might still be considered literary, just as Latin forms in contemporary English are considered literary.

The *Qie yun*, unfortunately however, makes no distinction between popular and literary forms. In fact, it does not even reflect the spoken language at the time it was written, but was an attempt to codify the *proper* pronunciation of words that were fast becoming literary. The express intention of Yan Zhitui, one of the principal advisers to Lu Fayen in the compilation of the *Qie yun*, in fact, was that his “children and grandchildren should speak correctly” (Pulleyblank 1984, p.60), presumably using literary forms and the *Qie yun* was a step towards ensuring this.

But just how reliable would such a base be for work in historical phonology? Opinion on this matter is clearly divided. Norman (1988, p.42) seems to advocate that a “reconstruction of the ancestor of spoken (popular) forms of Chinese, working backwards from present spoken dialects...would avoid much of the artificiality which plagues the traditional approach,” a proposal that is also welcomed in Newman (1996, p.110, note 2). It must be remembered, however, that reconstructed forms in Chinese are a result of both combining insights of the Chinese philological tradition and the comparative method in modern linguistics (Newman 1987). Indeed, Newman (1996, p.93) states:

The weight of the tradition is such that it would take an extremely brave (or arrogant) researcher to dismiss the philological evidence.

Miller (1975), on the other hand, had mounted a vicious attack on all traditional methods in Chinese linguistics. He describes them as “highly eclectic” and even tainted by “mystical speculation” (Norman 1988, p.40) although these are not the only grounds on which he condemns them. Chang (1974) maintains the extreme possibility that the phonology embodied in the *Qie yun* represents the *sum* of all distinctions in one dialect known to the compilers rather than a dialect that can be considered *standard* in some sense.

Pulleyblank (1970) himself, a prominent sinologist, had criticised the traditional approach in 1970, but in 1984, he seemed more forgiving. Reporting from the work of Zumo (1963), he says:

...there is no basis for such exaggerated scepticism [*towards Qie yun*].
...[*It*] represents a standard of correct speech common to the educated classes of north and south China in the sixth century. In other words, it was a cultivated “Mandarin”, not narrowly based on a single regional dialect but representing a norm which those in both parts of the country could aim at (Pulleyblank 1984, p.129).

Under these circumstances, an exclusive reliance on the *Qie yun* or its derivatives as a base for ongoing research in historical Chinese phonology seems justified, if only by the absence of any other equally authoritative source. A significant component of the data used in this study is thus ultimately based on the *Qie yun* — the Middle Chinese reconstructed forms as interpreted, phonetically, by Chen76. The remaining data is derived using diachronic rules that give forms in two modern Chinese dialects from these reconstructions. The next section discusses this in more detail.

4.3 *Zihui* and DOC

The raw data used in this project comes from the Pronouncing Dictionary of Chinese Dialects, compiled by the Beijing University, referred to here as *Zihui*

(1962). This is a comparative dialectological dictionary which records, in modified IPA (called DOCIPA), the pronunciation of 2714 characters in seventeen Chinese dialects covering all the major dialect groups in mainland China. In addition to the pronunciation in contemporary Chinese dialects, the *Zihui* also gives information on how each morpheme was pronounced in Middle Chinese. The data for this comes from the *Qie yun* and the data for the dialects from the 1956-58 dialect survey undertaken by the Government of China.

The *Zihui* itself was published in 1962. All the data in *Zihui* has since been computerised under the direction of William S-Y. Wang at the University of California, Berkeley. This resulted in the DOC, 'Dictionary On Computer', from which we have easy access to a large amount of comparative information and statistics.¹ Streeter (1977) gives a comprehensive description of the DOC and its contents. An informal and fascinating account of the development of the DOC between 1966 and 1993 at Berkeley is given by Cheng (1994a).

The DOC seeded prolific research in historical and comparative Chinese linguistics, some even before its completion. Reports of many of these were published as interim progress reports of the phonology laboratory at Berkeley, titled MIMs or Monthly Internal Memorandums; see (Cheng 1994a, p.73) and Streeter (1977, p.102ff) for an account of these. But more importantly, the data in DOC has continued to be useful and viable up till today. Recent studies in Chinese phonology that have used the DOC, other than Chen76, CN84 or the present work include Newman (1994) which attempts to explain the unexpected occurrence of the high front rounded vocalic reflex — *y* in Wenzhou and Newman (1996) which is an account of exceptions where the Chinese philological tradition took precedence over the comparative method of modern linguistics in Middle Chinese reconstruction.

In 1989, a second edition of the *Zihui* was published. There have been some corrections to errors in the first edition and also some significant updates

¹The DOC also indicates for each entry whether it is a literary or non-literary pronunciation in the dialect concerned. In the case of the Beijing and Cantonese data, there are very few instances of a literary/non-literary distinction (less than 10 in 2714). In such cases non-literary versions have been chosen for this project.

— information about 3 more dialects has been included and the number of Chinese characters used as the basis for the transcription has increased from 2722 to 2961.² However, its electronic mirror in the DOC has yet to catch up. Meanwhile, studies that have required a computerised database such as the DOC have used mainly the first edition, and those that didn't have used the second. This study uses DOC93 which is the 1993 edition of DOC based on the first edition of *Zihui*.

4.4 Motivation

Why was the method of this thesis applied to the Chinese language family as opposed to, say, the Indo-European or Austronesian families? This question is begging to be asked, especially since the example in Chapter 3 illustrated the method by its application to the Germanic family of languages. In fact, the original intention was to continue work within the Germanic family. To this end, the possibility of collaborative work was investigated with Professor Joseph Voyles of the department of Germanics, University of Washington, Seattle. Voyles himself welcomed the idea after having read a preliminary version of Raman and Patrick (1997c).³ However, practical difficulties in this collaboration soon surfaced since he did not have access to the Internet and all correspondence had to be routed via the postal system. Besides, the data for the Germanic family was rather limited in quantity. Although the Voyles hypothesis of diachronic phonology between early and later Germanic languages was available, an authoritative corpus of data, namely a significant number of reconstructed *LNWG forms and their corresponding attestations in later Germanic, to which the hypothesis could be applied was lacking. These could no doubt have been constructed through collaboration with Voyles, but inevitable delays in correspondence through the postal system introduced difficulties. Thus this

²Only 2714 of the 2722 forms stated to be represented were actually available for study in this project because of the absence of the corresponding reconstructed Middle Chinese forms in the *Zihui*.

³Personal communication with Professor Joseph Voyles: Correspondence regarding possible collaboration, in Letter dt. 12 Sep 1994.

option was given low priority.

One other option investigated was the use of a hypothetical language especially designed to test the proposed methodology. While this has the advantage of being easy and uncontroversial, the results would have been uninteresting and of little use, nor would they be verifiable against present linguistic knowledge. The data would be purely illustrative and serve no other purpose. This had already been done using the artificially small Germanic dataset in Chapter 3. At this point, it was decided to use the Chinese data which was available through John Newman at Massey University's Department of Linguistics and Second Language Teaching.

Why wasn't the Chinese data used to start with? Initially, before the methodology had taken shape, it had not yet been decided whether the method was to be purely illustrative or was to be applied to real data. Besides, at that point, the amount of data that would be needed had not been estimated, and it was assumed that a small sample of reconstructed words would be sufficient. While an awareness at that time of the availability of the Chinese data and possibility of using it was present, the investment in time and effort in learning the diachronic rules and their relative chronology coupled with the then unknown agreeability of this project among contemporary linguists elsewhere seemed to outweigh the advantages of using a real dataset. It was only after some initial work had been done and a preliminary publication had generated favorable suggestions from reviewers that it became apparent the method required a large corpus of data in order to produce significant results.

Besides the ready availability of a large corpus of this data at a local site in electronic format⁴, its use for this project had the following other advantages:

- Availability of a large and authoritative corpus of data (the *Zihui*) covering a wide variety of sounds in various phonological contexts.

⁴The SMC reconstructions of the DOC forms, the IPA transcriptions of the SMC, Beijing and Cantonese forms and the English meanings of the characters had all been entered into a FilemakerPro database on an Apple Macintosh by Dr John Newman. It was initially exported using a Macintosh scripting language into Microsoft Excel. But since it was tedious to edit it using Excel, it was migrated into an ASCII L^AT_EX table, where it currently resides.

- Availability of a set of diachronic laws from a single reconstruction to two descendants presented in comparable format (Chen76 and CN84).
- Easy accessibility for discussion of one of the authors of the CN84 hypothesis, namely John Newman.

Although the *Zihui* has these advantages, it is not free of problems. However, it was decided to use it for this study as its shortcomings were far outweighed by the advantages of using it. Newman (1994, p.334) states:

Admittedly the *Zihui* has its limitations: it is not as comprehensive as one would like it to be nor does it include age and gender differences. Nevertheless, it is a solid database which allows some quantification of historical changes and enables easy comparison with other Chinese dialects.

In the following sections, a further aspect of the motivation for using this data is presented — namely, the availability of a simplified reconstruction of Middle Chinese based on the *Zihui* with a comparable relative chronology for Modern Beijing and Modern Cantonese.

4.5 Chen76 and CN84

Chen76 and CN84 both give diachronic phonologies, the former for the Beijing dialect (on which standard Mandarin is based) and the latter for Modern Cantonese. Obviously, the diachrony presupposes the existence of a protolanguage and in both cases, the protolanguage is Middle Chinese. More precisely, it is the language of the *Qie yun* — (QMC). Although the phonetic representation of Middle Chinese categories is available through the *Qie yun* and described in Li (1952), the representation used in this dissertation is a simplification of it by Matthew Chen in Chen76 called Simplified Middle Chinese (SMC). The reason for this is that SMC is claimed to take the most conservative position amongst the many one could take with regard to the Middle Chinese of *Qie yun* “consisting of a subset of phonological categories that constitute the non-controversial

core of present knowledge about it” (Chen76 p.129). It is in fact identical to the QMC of Li as far as the initials and tones are concerned. SMC and QMC are only different in the finals, and that too only in some categories which are distinct in QMC, but have been collapsed in SMC. Furthermore, the distinction is tentative, as Chen76 is open to reintroducing contrast as and when a need arises.

In addition to the SMC reconstruction in Chen76, which CN84 also uses as a starting point for diachrony, the following enhancements to traditional Chinese phonology is made by them. At the time of publication of Chen76, Hashimoto (1965) was perhaps the only serious study in the phonological evolution between Middle Chinese and Mandarin. Although Pulleyblank also addressed the issue in 1984, he seems to have devoted just one short chapter to it (Pulleyblank 1984, ch.2, pp.41–59) and that too does not address the issues raised by Chen76 (p.115). Chen76 seems to be the only definitive attempt in this direction. The specific inadequacies of the earlier phonologies that Chen76 states are:

- No attempt was made to generalise what could be classed as general sound changes.
- No relative chronology was supplied and thus rules were unordered with respect to each other.
- There was a failure to capture the greater picture from the smaller changes that were happening to individual categories. For example, seemingly disparate groups of vowel changes could actually be pieced together as in a jigsaw puzzle into a movement that Chen76 calls ‘The Great Vowel Shift’
- There was excessive preoccupation with descriptive formalism, especially in Hashimoto (1965).

A similar criticism is made by CN84 against Hashimoto (1972) in connection with Modern Cantonese. Thus, with these re-engineering tasks to accomplish, Chen76 and CN84 propose new diachronic phonologies for modern Beijing and

Cantonese. This includes a Relative Chronology (RC) for each which is discussed in the following section.

4.6 Relative Chronology (RC)

The RCs depend on bleeding and feeding relationships between diachronic rules. If a rule consistently feeds another, for example, then the feeding rule is chronologically ordered before the fed rule. Since sound correspondences in complementary distribution undergo non-intersecting development except for the origin, there are obviously sets of rules that bear neither bleeding nor feeding relationships to each other. These rules are grouped together. Based on such considerations, Chen76 and CN84 proposed a tier structured RC where rules are grouped horizontally into tiers and the tiers themselves are arranged vertically. In the RC for Mandarin represented diagrammatically in Figure 4.1, there are 6 tiers. A rule belonging to a higher tier (lower numbered) may not apply after a rule in any one of the lower tiers has applied. The diagram comes

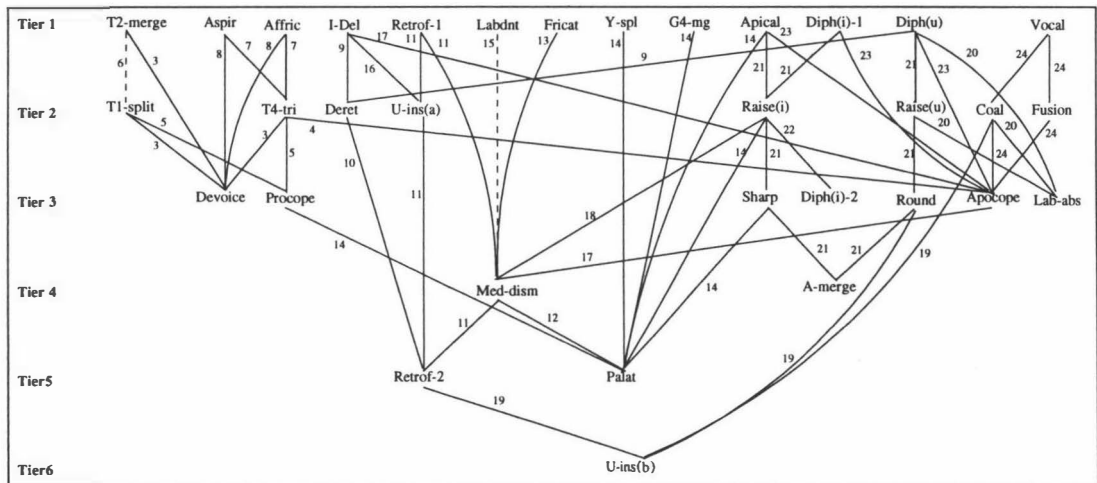


Figure 4.1: Chen76 Relative Chronology for *SMC>MB.

from Chen76 (p.247) and that paper should be consulted for an explanation of all the labels, but some minor changes to the rule names have been made to get the diagram to fit within the A4 paper in a reasonable sized font. The

original diagram was in landscape mode and thus much bigger. However, the names that have been used are similar enough to the ones used by Chen to avoid any confusion. Lines between the rule names indicate bleeding/feeding relationships. The numbers labelling the lines are the numbers assigned to the particular RCs and are used to index into the Chen76 RC (pp.220–246) to locate further discussion about specific bleeding or feeding relationships and justifications for particular orderings.

A minor correction needs to be made to the original diagram with regard to the ordering of the rule PROCOPE. It has a definite bleeding relationship with the rule SHARP, since SHARP requires a velar or glottal initial environment to operate which PROCOPE would delete if it applied first. Since there were a significant number of forms in which SHARP seemed to have applied before PROCOPE, PROCOPE was shifted down to tier 4 from tier 3 and the appropriate modification was made to the phonology.⁵ This change is reflected in the diagram in Figure 4.2.

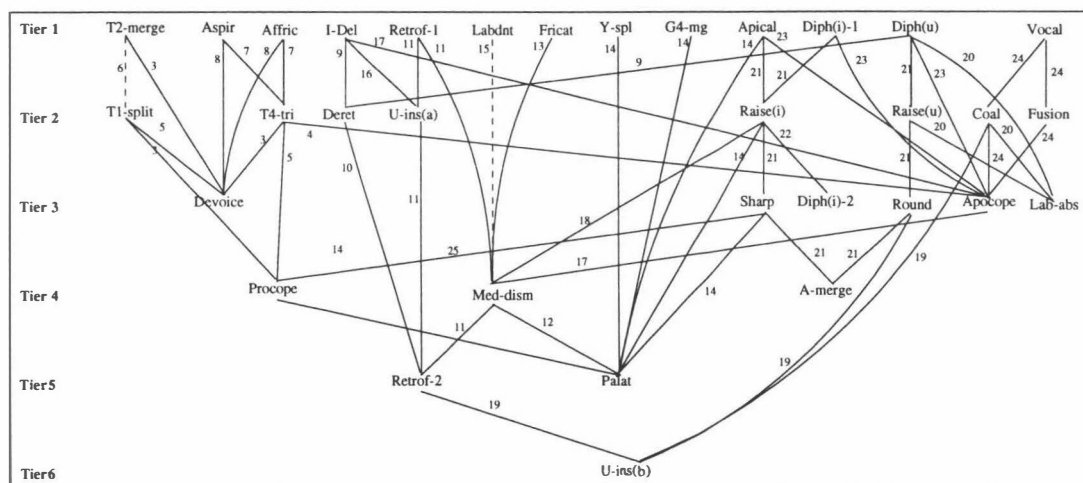


Figure 4.2: Chen76 Modified Relative Chronology for *SMC>MB: RC25 (SHARP > PROCOPE) orders SHARP before PROCOPE.

For Cantonese, CN84 proposes 8 tiers. The RC for Cantonese phonology is

⁵Personal communication with John Newman: Comment on diachronic bleeding and feeding relationships, in Discussion dt. Aug 96.

represented diagrammatically in Figure 4.3. See CN84 for an explanation of all the terms used. Again, some rule names have been slightly changed, but not enough to confuse the reader. The other details concerning the lines and their labels are similar as for Chen76, only they apply to CN84 instead.

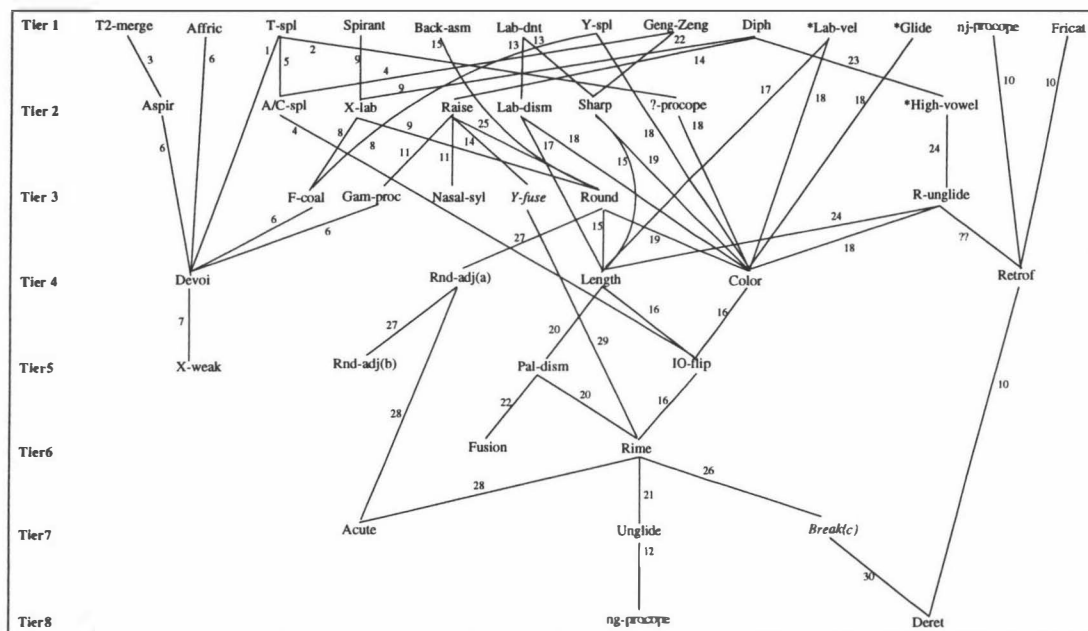


Figure 4.3: CN84 Relative Chronology for *SMC > MC.

For a description of the actual rules and the RCs themselves, see Chen76 and CN84. In Table 4.1 a typical example from Chen76 (p.222) is given. Two words in Modern Beijing — *soldier* and *example*, are derived from their corresponding SMC forms alongside each other. Column 3 lists the rules that apply to the initial and intermediate forms. Each form starts off in SMC and undergoes a series of rule applications, and finally emerges as the Modern Beijing form. A dash in one of the first two columns indicates that the corresponding rule in column 3 didn't apply at that point. This could be significant as the rule, if it applied, could have changed the final output by bleeding a later rule.

The ordering of rules is critical to the final output. This specific example was chosen for a purpose. It allows for the explanation of an unusual phenomenon in the RC with respect to what are called Phonotactic Conditions (PCs) which

soldier	example	
piań	*liai	*SMC
-	liei	RAISE(I-UMLAUT(b))
-	liii	RAISE(I-UMLAUT(c))
-	lii	*DEGEMIN
-	li	*DEGEMIN
piań	-	VOCAL
piaŋ	-	FUSION
pieŋ	-	COALESCE
piəŋ	-	*CHAMEL
piŋ	-	*RIMEMIC
/piŋ/	/li/	Modern Beijing

Table 4.1: An example of the Relative Chronology in operation.

is discussed in the next section. In the example, following the convention in Chen76, PCs are marked off with an asterisk to set them apart from the regular phonological rules, which Chen76 calls P-rules. This is to be distinguished from the usual asterisk marker for SMC words which indicates their protoform status.

4.7 Phonotactic Conditions and Allophonic rules

The constitution of a syllable in Chinese is governed by a set of sequential and distributional rules. Chen76 calls these rules Phonotactic Conditions or PCs. These are brought about by the constraints imposed on forms by the nature of the language itself at that point in time.

In contrast to Phonological rules (P-rules), which represent specific innovations at a given point in time, Phonotactic Conditions (PCs) characterise one or another aspect of a more or less stable *intermediate* state (Chen76, p.220, emphasis mine). They differ significantly from the P-rules with regard to their mode of operation. Also, a PC must enter a language and fade out of the language at a given moment in its history. They are brought about by the way that the language is implicitly realised. In this they are more like phonotactic constraints than conditions, because if a context arises for the application of a PC and the PC is currently in force, it must apply.

In the example used in the previous section, *DEGEMIN is a typical example of a PC. Neither SMC nor Modern Beijing permit geminate (repeated) vowels or consonants. Thus a PC inhibits their viability. As soon as a situation is realised where such a repetition occurs, the PC *DEGEMIN will apply to whittle the sequence down. In the example, this is seen in the instance of “liii” where the geminate “i” was generated by the I-UMLAUT P-rules. One further obvious fact from the previous example is that PCs can apply iteratively.

In general, PCs have no feeding or bleeding relationships with P-rules. Thus they are unordered with respect to P-rules and are, strictly speaking, not part of the RC. However, Chen76 names just two PCs which are the exception to this rule for Mandarin — *CHAMEL (Chameleon vowel) and *PAL-DISM (Palatal Dissimilation), which both, according to him are late rules which must have arisen in post-Middle Chinese times. *CHAMEL says in effect, that there is only one mid-vowel phoneme /ə/ with contextually predictable allophones. Its mode of operation is summarised in Table 4.2.

S_2	i	u	c	#	Key
S_1					S_1 = preceding segment
i, y	-	o	∅	e	S_2 = following segment
u	e	-	∅	o	P = labial consonant
P	e	o	ə	o	C = Any other consonant
C, #	e	o	ə	ɤ	# = word/syllable boundary

Table 4.2: Allophones of ə; Operation of the *CHAMEL PC.

Since this is only true of a 5-vowel system like that of Mandarin, it obviously could not have been a PC applicable to SMC which was an 8-vowel system, with contrast between the three mid-vowels /e, ə, o/. *PAL-DISM causes the depalatalisation of a palatal onglide in the environment of a palatal final. Thus in the previous example, *PAL-DISM could not have applied before either RAISE or VOCAL because it would have bled them, both of which operate only in the environment of a palatal onglide. This is the justification for Chen76 RC2 which incidentally is unrepresented in Figure 4.1. There are no corresponding exceptions with regard to PCs in CN84 although Modern Cantonese does have

tighter constraints due to it being strictly trisegmental.⁶

In view of these circumstances, PC's have been treated as follows: They have been considered applicable at arbitrary places (except, of course, *CHAMEL and *PAL-DISM) and are treated as part of the diachronic derivation process. Thus, the sequence of rules deriving Beijing *li* from SMC **liai* would be “raise(I-umlaut(b)):raise-(I-umlaut(c)):*degemin:*degemin:” where lower-case labels have been used for readability. For deriving Beijing *piŋ* from SMC **pian* the sequence would be “vocal:fusion:coalesce:*chamel:*rimemic:” where again, lower case labels are used.⁷

As regards the special status of *PAL-DISM and *CHAMEL, the following decision was made. They are ordered as late as possible in the derivation process within the constraints of feeding and bleeding relationships. Also, wherever it is possible to successfully apply *CHAMEL as the last rule, it is treated effectively as an allophonic rule because that is presumably its main function in Modern Beijing. What exactly constitutes an allophonic rule and when an allophonic rule ceases to be considered synchronic and becomes part of the diachronic phonology seems rather unclear in historical linguistics. Since it is beyond the scope of the present dissertation to examine this issue in any detail, a procedural decision was made to order all allophonic rules in the descendant forms (be they Cantonese or Beijing) as far as possible at the end of the derivation process. The only exceptions to this are when bleeding and feeding relationships between them and P-rules prevent this. In such a case, the embedded application of allophonic rules is considered exceptional. Both these decisions bear significance because two levels of dissimilarity measures are provided, one including allophonic detail and one not. This issue will be

⁶Syllables in other dialects and the reconstructed Middle Chinese can have up to four segments. This is discussed in greater detail in Newman (1987).

⁷These sequences are now in a form suitable for direct processing by the PFSA construction and manipulation programs made available with this dissertation. These need rule labels separated by colons. A colon at the end of a line denotes the end of the derivation process and indicates that the final form had been reached. The programs accept input in either upper or lower case, but downcased rule names have been used to make the sequences more readable. Also, to keep lengths of lines representing rule sequences manageable, the actual rule names have been abbreviated, sometimes slightly more than in the Figures 4.1 and 4.3, but still not short enough to cause confusion.

revisited in a following section and in a later chapter.

4.8 Automatic Derivations

Deriving Cantonese and Beijing forms for each of the 2714 SMC forms in the *Zihui* was one of the most time consuming operations in the project. The derivations were prepared by hand for each of the 2714 forms and entered into a file for further processing. The obvious question that springs to mind at this point, however, is regarding why the process wasn't automated. Presumably, since there are clear-cut notions of contexts where each rule can and can't apply, and there is a body of Relative Chronology to aid the process of deciding which rule to use, it should have been easy to automate the whole process.

The answer to this is two-fold. Firstly, the original data, as available to this project was not given in the form of distinctive features per segment. The rules themselves, however, depend largely on the features exhibited by the individual segments. For example, the *PAL-DISM PC is stated as:

$$\begin{array}{c} [+high + palatal] \quad (V) \quad [+high + palatal] \\ \downarrow \\ [-palatal] \end{array}$$

meaning that the medial onglide, which is both high and palatal is to be depalatalised if it is followed by a vowel and a similarly high and palatal segment. To change the *Zihui* database so that it lists each form in terms of its segmental features would have been a major undertaking and in many ways, more cumbersome than deriving each form separately.

The more important reason as to why the process wasn't automated, however, is the likelihood of exceptional derivations which is discussed in the following section.

4.9 Dealing with Exceptions

It should be noted that neither the CN84 nor the Chen76 RC is entirely binding. There are exceptions to them in the derivations of some forms (eg. CN84,

p.349 — reversal of usual order of LABIAL DISSIMILATION and LENGTH). One reason for the occasional conflicts between the philological and linguistic approaches in reconstructing protoforms is the “dominance of a long and venerable philological tradition within the Chinese culture” (Newman 1996, p.92) that was discussed in Section 4.2.1. When linguistic reconstruction of a form using the comparative method conflicts with Chinese philological evidence, the latter form is usually adopted and an exception introduced into the diachronic derivation process.

Such exceptions also bear significance in the results obtained. A large number of unpredictable exceptions to the linguistic hypothesis will contribute to a greater complexity in the diachronic phonology. If we assume that these exceptions are a result of residual forms being left behind in a language, then one could say that the complexity of a language increases with time due to the presence of such residual forms and exceptions (Crowley 1987, p.150–153). The PFSA derived for the more distant language will have a greater MML than the other. If on the other hand, one subscribes to the *wave theory of linguistic change* and assumes that exceptions are caused by a change that has swept over part of a geographic region, and is yet to affect phonemes completely in other parts, then a greater MML would indicate quite the opposite since a completed linguistic change is considered to achieve uniformity of application.

The problem surfaces for any work that needs to utilise a reconstruction comprehensively because the exceptions have not been accounted for by any of the popular reconstructions, including the two that this dissertation is concerned with — Chen76 and CN84. Whatever the reasons for these exceptions, and whatever their significance may be, it remains that they pose some problems to this project. They must be dealt with and incorporated somehow into the framework of deriving distance measures. Besides the obvious problem of deciding how to deal with them, they constitute a reason for being unable to automate the process of generating derivations, since the entire output of an automatic derivation process will have to be checked for exceptions anyway. The following subsections discuss the most common types of exceptions and

the decisions made towards addressing them.

4.9.1 Exceptions to the RC

Consistency with the RC proposed in Chen76 and CN84 has been maintained as far as possible. For the most part, violations to them are viewed as serious exceptions. Thus if Rule A is ordered before Rule B in the RC, but is required to apply after Rule B in a specific instance under consideration, it is made an exceptional application of Rule A, denoted by “[A]”. Such exceptional rules are considered distinct from their normal forms. The sequence of rules deriving Beijing *tou* from SMC **to* (“all”), for example, is given as “t1-split:raise-u:diphthong-u:chamel:”. However, “diphthong-u” is ordered before “raise-u” in the RC. It is thus made an exceptional application and the rule sequence is given instead as “t1-split:raise-u:[diphthong-u]:chamel:”.

4.9.2 Undocumented Changes

There are also exceptional phonological changes not accounted for by CN84 or Chen76. In these cases, a new rule is formed in consultation with John Newman representing the change that took place, and it is denoted in square brackets to show its exceptional status. Related exceptions are grouped together as a single exceptional rule. For example, Tone-4 in SMC only changes to Tone-1a or Tone-2 in Beijing when the form has a voiceless initial. However, for the SMC form **niat* (“pinch with fingers”) in Tone-4, the corresponding Beijing form is *nie* in Tone-1a. Since the n-initial is voiced, the t4-tripart rule is considered to apply exceptionally. The complete rule sequence is thus denoted by “raise-i:apocope:chamel:[t4:]” where the “[t4]” exceptional rule covers cases when Tone-4 in SMC unexpectedly changed into Tone-1a or Tone-2 in Beijing in the absence of a voiceless initial.

4.9.3 Unapplied Rules

It also needs to be mentioned that there are a few cases where an environment for the application of a rule existed, but the rule itself did not apply although

it was required to by the CN84 or Chen76 hypothesis. This would constitute an exception again. The details of how to handle this situation more accurately are left as a topic for future work, but we try to account for it here by applying a special rule [!A] where the ‘!’ is meant to indicate that the rule A didn’t apply when it ought to have. As an example, we may consider the derivation of Modern Cantonese *hap*(Tone 4a) from SMC **k^hap*(Tone 4) (“exactly”). The sequence of rules deriving the MC form is “t4-split:spirant:x-weak:”. However, since the environment is appropriate (voiceless initial) for the application of a further rule, AC-split, which should apply after t4-split had applied, the non-application of this additional rule is specified as an exception. Thus, “t4-split:spirant:x-weak:[!AC-split:]” is the actual rule sequence used.

In general, the following conventions in representing and treating exceptions have been followed as far as possible: Exceptional rules are always denoted in square brackets. They are considered excluded from the RC and thus are consistently ordered at the end of the rest of the derivation process wherever possible.

The most frequent exception thus encountered in the data was the unexpected occurrence of Tone A in a Modern Cantonese form, where the corresponding SMC form had a voiced initial. CN84’s TONE-SPLIT rule (CN84-I, p.165) predicts Tone B in such Cantonese forms. This exception occurs in 90 of the 2714 forms, approximately 3.28% of the total sample. It is noteworthy that it is sometimes possible to avoid exceptions such as this by violating the RC — in this case, by ordering the DEVOICE rule before the TONE-SPLIT rule, but from the preceding discussion, even this would constitute an exception. So either way there is a penalty to pay for describing this change.

In the few cases that exceptions have a feeding or bleeding relationship to other rules, they are ordered in-line with the other P-rules. In some cases, a derivation may be composed of two or more rule sequences, within each of which rules may be ordered, but the sequences themselves neither bleed nor feed one another within the context of the word being derived. Here again, an attempt to be consistent is made by ordering the earlier rules before later ones.

4.10 Status of Allophonic Changes

A final detail concerns the status of allophonic changes in the phonology. The derivation process is actually two-stage, comprising a diachronic phase during which phonological changes take place in the vocabulary and a synchronic phase during which phonological changes are automatically realised in it. Changes caused by Cantonese or Beijing Phonotactic Constraints (PCs) are treated as allophonic rules and fall into the synchronic category, whereas PCs applying to earlier forms are treated inline with the P-rules.

Allophonic rules present a minor problem. There is some confusion concerning their exact classification. In Chen76 and CN84, they are not really considered part of the historical derivation process. Yet, the environment for the application of a diachronic rule is sometimes produced by an allophonic rule. Such feeding relationships between allophonic and diachronic rules make the classification of those allophonic rules difficult. Table 4.3 gives an example from Chen76 (p.227) of the CHAMEL PC feeding the DERET rule which de-retroflexes an initial followed by a schwa.

duty	
tʂac	*SMC
tʂaic	VOCAL, FUSION
tʂek	COALESCE
tʂək	*CHAMEL
tsək	DERET
tsə	APOCOPE
tsɤ	*CHAMEL
/tsɤ/	Modern Beijing

Table 4.3: An example of an allophonic rule in feeding relationship with a diachronic rule.

The only rule considered allophonic in Beijing is the *CHAMEL PC. Its treatment has been discussed already. For Cantonese, CN84 has included two allophonic rules within its RC under bleeding and feeding relationships with P-rules. These are the BREAK-C and Y-FUSE rules. In these cases, every instance of their application within the diachronic phonology has been treated as

an exception, effectively elevating these exceptions to the status of diachronic rules. In other cases, as with other allophonic rules, they are always ordered after all the diachronic rules. Since the problem regarding the status of allophonic rules in general is properly in the domain of historical linguists, it is beyond the scope of this work. It was thus decided to provide two complexity measures — one including allophonic detail and one excluding all allophonic detail not required for the derivation process.⁸ The Cantonese and Beijing forms in the data corpus, however, are shown including the effect of various allophonic rules. The rule-sequences themselves are segregated into two separate columns, the first for diachronic and the second for allophonic (see appendices D and C).

4.11 Consistency of Presented Data

If results from the two sets of data dealing with the phonologies of Modern Beijing and Cantonese respectively are to be commensurable, they must reflect the development of the language up to the same period in time. Besides, they must also include the same level of linguistic detail. Furthermore, the presented data must be consistent with any adjustments made to achieve commensurability in order to be easily followed in conjunction with the discussions in this dissertation.

The data in Appendices C and D are given in tabular form with 9 columns each, listing for a given SMC form: its page number (col 1) and column (2) in the original *Zihui*, the corresponding Chinese character (3), its reconstructed SMC form (sans asterisk marker) in IPA (4) and tone (5), its Modern Beijing or Cantonese form in IPA (6) and its tone (7), the colon separated sequence of P-rules (including exceptions) that derive the stated modern form from its protoform (8), and finally a colon separated sequence of allophonic rules (A-rules) that are used for the phonetic realisation of the form in present day usage (9). It is the rule sequences in columns 8 and 9 which constitute the final data for this project. They are used as input to the programs that generate PFSA

⁸See a further discussion of this point in the Section 4.11.

and it is the MMLs of these PFSA which signify distance measures. Table 4.4 is a short extract from the Appendix D to illustrate the way in which the data has been presented.

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
24	9	夺	duat	4	tyt	4b	t-split:devoi:round	round-adj:degem:acute
25	1	朵	tua	2	ta	2a	t-split:round:unghide	

Table 4.4: An extract from Appendix D to illustrate the actual presentation of the data.

For representing tones (columns 4 and 7), the notation in Chen76 is used (which uses the symbols *T1-*T4 for SMC tones and uses splits A, B, and C within these) rather than the notation of Chao (1930). Besides, we can not know just what the tones were, phonetically, in SMC. Hence *T1-*T4 would be a more cautious way of representing the tones.⁹ If the data is to be processed with allophonic detail, then the derivation sequences in columns 8 and 9 are concatenated for each row. If allophonic detail is not required, then column 9 is simply ignored.

The decision to present two levels of distance measures (pre and post allophonic forms) was made relatively late in the project. Earlier, it was decided to represent the data in column 5 sans allophonic detail since the realised forms could be contextually predicted. This posed a problem because of the somewhat volatile status of allophonic rules in each of the two phonologies. In particular, some allophonic rules have been treated as diachronic rules by some authors. As a case in point we could consider the GLIDE-INSERT rule from CN84 (I, p.163). There is evidence to classify it as allophonic in the common Hanyu-pinyin representations of /u/ and /i/ in Beijing which are written as *wu* and *yi*. CN84, however, treats it as significant in the RC when deriving modern Cantonese forms from SMC. Pulleyblank (1984, p.43) and Krachtovil (1968, p.32),

⁹The representations of tonal categories used in the historical phonology of Chinese must be distinguished from the representations used in modern Chinese languages, especially in books aiming to teach a person to speak one of the modern languages. For example, many books for learning Mandarin frequently talk about tones 1, 2, 3 and 4. But these are not to be confused with tones 1, 2, 3 and 4 of SMC. They correspond to the tones 1a, 1b, 2 and 3 in the historical tradition.

on the other hand, consider comparable facts in Standard Beijing as part of allophonic detail. Similarly, Chen76 does not include any discussion of glides in [wu] and [ji] in his account of the history of Beijing. Thus there is a discrepancy in the scale of change considered for the two sets of derivations in this project — where the historical development of standard Beijing forms stops at [u] and [i], we find corresponding Cantonese forms go on to become [wu] and [ji].

If we took the two datasets as they stood, they would have been incomparable since allophonic detail was considered to be part of the rule-sets in Cantonese but not in Beijing. The solution was then to either exclude the allophonic rule details from the former or include it in the latter. Since it was found that there were several Cantonese reflexes in the data which had already accounted for other allophonic rules (CN84 III p.128), and reversing their effects would have been an awkward process, the latter option was adopted. Besides, if the former option had been chosen, the derivation sequence will seem not to account for the exact reflex represented in the data as the derivation process would have gone all the way, (in column 9) including allophonic detail, but the representation would have stopped short. Thus the data presented in the appendices include allophonic detail in both MC and MB forms.

Chapter 5

The MML Principle

5.1 Modelling Inductive Hypotheses

The example in Chapter 3 shows that the problem posed in this thesis involves the modelling of an inductive hypothesis derived from a finite sample of word transformations. This chapter discusses just how such inductive hypotheses may be modelled and how the cost of describing them may be computed. To do this, we first consider the nature of hypotheses in general. An inductive hypothesis can be said to be an abstraction over a set of observations. The problem of abstracting such a structure (a pattern of occurrence) from a given set of observations is “ancient and fundamental” (Gaines 1976), “but the process itself has undergone little conceptual change in centuries.” The abstraction, in general, consists of two parts, (Georgeff and Wallace 1984):

- A statement of the hypothesis itself encoded in a suitable language

and

- A specification of the data given this hypothesis

Under normal circumstances, the data specification part is dominated by exceptions to the hypothesis because confirmations, which are already implicit in the hypothesis, can typically be stated succinctly. It may be that the hypothesis is accurate and that there are no exceptions to it in which case the

data specification part will be very concise. In general, a good hypothesis will minimise the second component of the abstraction.

One of the ways in which the hypothesis can be stated is as a list of the probabilities of occurrence of each data item. The data items can then be specified using these probabilities, assigning the shortest code to the most probable data item and the longest code to the least probable one. Assuming that the symbols representing the data item are encoded using an alphabet with n unique symbols, information theory guidelines suggest that the optimal codelength for a symbol representing data item i is $-\log_n p_i$ bits where p_i is the probability of occurrence of the i th symbol (Shannon and Weaver 1963). Huffman encoding, (Huffman 1952; Gallager 1968), achieves this in practice with a prefix code over a binary alphabet. A code has the prefix property if the encoded symbols are self-delimiting. This is important so that the codes can be unambiguously discerned from a stream of concatenated encodings. The fact that $-\log_n p_i$ may not always be a whole number means that there will always be some redundancy as the length of the code will have to be rounded away from zero. However, this is not a major concern in coding as other coding schemes, notably arithmetic coding (Witten et al. 1987) allow us to achieve this theoretical optimum per-symbol length.

A variation of this approach encodes the abstraction in the form of a Probabilistic Finite State Automaton (PFSA). A PFSA is a state determined machine with a stochastic transition function. That is, moves from one state to another are governed by probabilities. Gaines (1971) adopted this approach with regard to the problem of developing a behavioral account of human behaviour. This was further developed by Wallace and Georgeff (1984) who looked at the general problem of inferring a structure for a PFSA from a given string with zero or more embedded delimiters.

PFSA, in general, can code any hypothesis that can be coded using our earlier scheme, but somewhat more intuitively. To see how, consider the following scenario. Four events A, B, C and D are observed to occur in that order a large number of times, say $n \gg 1$. Eventually, a biased observer — it is important

that some kind of bias exist (Hanson 1958) for an inductive hypothesis to take shape — will begin to form a generalisation about the environment in which these events occur and the generalisation is then formally stated. Let us see how the hypothesis may be coded using each scheme. Since temporal information allows us to capture the sequence of occurrence of the four events, the sequence itself can be viewed as a single event, say X . This is a result of Shannon's noiseless coding theorem (Abramson 1963) which allows us to improve coding efficiency by grouping source symbols where such grouping information is available. X being the only event known to occur, it will now be assigned the shortest code, say C_1 , in accordance with Huffman (1952). Thus the hypothesis specification will code the probability of occurrence of X and the data specification will be n repetitions of this code, which will be concise as C_1 is small. The entire abstraction will look as in Figure 5.1

Hypothesis
$C_1 = ABCD$
Data
C_1
C_1
\vdots
C_1 (n times)

Figure 5.1: An encoding of 4 events $(ABCD)^n$.

Suppose now, that the sequence was disturbed abruptly by the sudden occurrence of an unexpected event B' after the m th A ($m < n$), which is followed by C and D .¹ Assuming that our coding algorithm dictates that the sequence A, B', C, D be viewed as a single event Y and assigned a code C_2 longer than C_1 , the abstraction will now look as in Figure 5.2 which will be longer than the size of the first explanation as $|C_2| > |C_1|$. The sizes of C_1 and C_2 in this hypothesis are determined by the probabilities of occurrences of the event sequences

¹Note that this is not the only kind of exception that could exist for a theory. Quinlan (1993, p.52), for instance, notes two classes of exceptions in classifying rules — positive examples not predicted by the hypothesis (commissions) and negative examples predicted by the hypothesis (omissions).

A,B,C,D and A,B',C,D respectively. Accordingly, C_1 is $\log n - \log(n - 1)$ symbols long and C_2 is $\log n$ symbols long where the base of the logarithm is the number of unique symbols in the alphabet.

Hypothesis
$C_1 = ABCD$
$C_2 = AB'CD$
Data
C_1
\vdots
C_2
C_1
\vdots
C_1

Figure 5.2: An coding of the $4(ABCD)^n$ with an exception B'

If we decided to code the hypothesis explaining the first data set as a PFSA, we would construct a state determined machine (see Figure 5.3) which simulates the sequence of occurrence of the events A, B, C and D, in that order. We would consider a five state machine, in which event A takes it from state 0 to 1, event B takes it from 1 to 2 and so on. Also, we would specify the transition probabilities for each symbol. We could do this indirectly by specifying the frequencies of occurrences of the symbols, as the probabilities are directly computable given the frequencies of all outgoing transitions from a state.

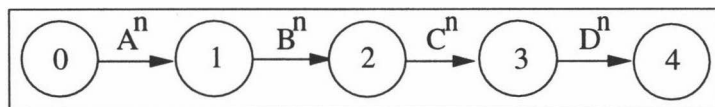


Figure 5.3: The events $(ABCD)^n$ encoded in the form of a PFSA.

A PFSA can also be represented as a set of 4-tuples (S_1, S_2, M, P) where $S_1, S_2 \in S$, a finite set of states, $M \in \Sigma$, a finite set of output symbols, and $0 \leq P \leq 1$ is the probability of transiting on M from S_1 to S_2 . Alternatively, $P \in W$, in which case it stands for the frequency of transitions on M

in that 4-tuple. $\{(0, 1, A, n), (1, 2, B, n), (2, 3, C, n), (3, 4, D, n)\}$ would be an ASCII representation of the PFSA just considered. With the anomalous event B' , a PFSA encoding would be as in Figure 5.4.

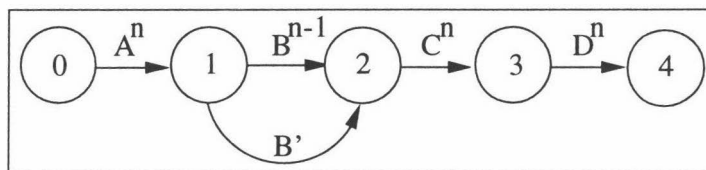


Figure 5.4: The events $(ABCD)^n$ with exception B' encoded in the form of a PFSA.

The number of states of the automaton has not increased in this case, but this may not always be so. The extra transition from state 1 to 2 accounts for the anomalous event B' . The ASCII representation of this PFSA will be $\{(0, 1, A, n), (1, 2, B', 1), (1, 2, B, n - 1), (2, 3, C, n), (3, 4, D, n)\}$ which is slightly longer than the previous one.

5.2 Information Measures and MML codes

At this stage, it is worth pointing out that every set of observations has at least one hypothesis purporting to explain it. As proof of this statement, we may consider the trivial null theory which claims to explain any given data set, which however has as exceptions to it all the observed data and no confirmations whatsoever. Thus the problem of modelling an inductive hypothesis from a body of data really becomes one of choosing between competing models. Georgeff and Wallace (1984) proposes the Minimum Message Length (MML) principle to help make such a decision. It states that the most preferred theory to form is that with the highest posterior probability or, equivalently, that theory which maximises the product of the prior probability of the theory with the probability of the data occurring in light of that theory (Oliver and Hand 1994; Cheeseman 1993). Quantitatively, this is the same as saying that the best theory for explaining a set of data is one which minimises the sum of

- the description length of the hypothesis *and*
- the description length of the data when encoded using this hypothesis.

This is justified thus: Let T be a theory with prior probability $p(T)$ purporting to explain data D . The posterior probability of the theory T in light of the data, $p(T|D)$, is given by

$$p(T|D) = \frac{p(TD)}{p(D)} = \frac{p(T)p(D|T)}{p(D)}$$

by application of Bayes' theorem. Since we wish to choose T , we prefer that choice which maximises $p(T|D)$ and therefore $p(T) \cdot p(D|T)$, D and $p(D)$ being given. Also, since $\log x$ is a monotonically increasing convex function in x , maximising the posterior probability $p(T|D)$ is equivalent to minimising $-\log p(T) - \log p(D|T)$. Information theory tells us that the optimum code length for a symbol of probability p is $-\log p$. Consequently, maximising the posterior probability of T can be interpreted as minimising the length of the message encoding it. Hence the name *minimum message length* principle for thus choosing a theory. The technique has its origins in Wallace and Boulton (1969) where it is fairly obvious that the MML criterion is being considered as a general purpose inductive tool. But Wallace and Freeman (1987) and Rissanen (1987) are more explicit expositions of the principle. Even earlier, Solomonoff (1964a), Kolmogorov (1965) and Chaitin (1966) had already looked at the problem of obtaining complexity measures for descriptions and Solomonoff (1964b, p.20) had suggested its use in choosing between rival inductive hypotheses.

The MML principle has been applied with some success in Patrick (1978) to choose between proposed shapes of the Megalithic stone geometries in the British Isles. Muggleton et al. (1991) uses the similar Minimum Description Length (MDL) principle of Rissanen (1978) within the framework of Inductive Logic programming and Quinlan and Rivest (1989) apply it to infer decision trees. Wallace and Patrick (1993) correct a derivation error in the latter paper and apply the MML criterion to the decision tree inference problem. A program for classification (SNOB) has also been developed that implements an MML

technique for discovering class structure (Patrick 1991; Wallace 1990; Wallace and Boulton 1968; Wallace 1986; Wallace and Dowe 1994).

Other applications of the MML principle in the past include the analysis of behaviour of players in an Australian rules football game (Patrick and Chong 1987), pattern matching in DNA strings (Allison et al. 1990b; Allison and Yee 1990; Allison et al. 1990a; Allison et al. 1992; Milosavljevic et al. 1990), segmenting of DNA into structural and functional elements (Babcock et al. 1990), evolutionary tree reconstruction (Cheeseman and Kanefsky 1990), detection of features in data that are useless for theory-formation (Sheinvald et al. 1990), parameter estimation in a multivariate Gaussian model (Wallace and Freeman 1992) and estimation of the concentration parameter in the von Mises distribution (Wallace and Dowe 1993).

More recently, the principle of minimal length encoding for inference has found renewed application in inferring Decision Graphs (Oliver 1993), Reduced Ordered Decision Graphs (Oliveira and Vincentelli 1995; Oliveira and Vincentelli 1996), evaluating rulesets (Cleary et al. 1996), causal discovery (Wallace et al. 1996), evaluation of alternative inductive theories (Conklin and Witten 1994; Pfahringer 1995), in conjunction with genetic algorithms in evolving classification graphs (Neil and Korb 1996) and inducing finite automata (Clelland and Newlands 1994; Clelland 1995), substructure discovery (Cook and Holder 1994; Djoko 1994) and deriving computer program complexity measures (Patrick 1996).

5.3 MML, Bayesianism and MDL

The MML methodology has its roots in classical Bayesianism. There is enough commonality between the two that people often tend to confuse them. However, it is worth bearing in mind that there are differences between them too. While the posterior probability of any one particular model is not the focus of the Bayesian approach, it is of the MML approach. The Bayesian framework is geared towards constructing a posterior density over all possible models, whereas the MML framework is designed to select one model over all competing

ones. This is discussed in greater detail in Oliver and Baxter (1994).

There is also some confusion regarding MML and MDL. Indeed, there is what can be called a whole family of names — Minimum Message Length, Minimum Encoding Inference, Minimum Description Length, Minimum Length Inference, Minimum Information Estimation — all of which have been used in the literature and more. It seems that an author generally picks one of these that sounds most fashionable to him or her. While there may be much (perhaps all) in common between most of these methods, it is important that some degree of consistency and conformity is enforced in the literature, if only for the benefit of the uninitiated researcher making his or her debut in the field. However, where a difference does exist, such as between MML and MDL, it is also important that nomenclature differ accordingly. Unfortunately, the liberty authors seem to take in picking a name apparently willy-nilly confuses the issue surrounding some important distinctions between MML and MDL (Baxter and Oliver 1995, p.3).

The MDL method is due to Jorma Rissanen (1978, 1987) and its express objective is to “find the best model class and the best model in it as judged by shortest code length”. The MML method, on the other hand is due to Chris Wallace and David Boulton (1968) and is concerned with finding *the* model that is the best explanation of the data. An even more important distinction between MML and MDL is in the use of subjective priors. For MML, “there can be no substitute for the careful specification of whatever prior knowledge is available” (Wallace and Freeman 1987, p.241), but for MDL, the incorporation of subjective priors “spoils the beauty of the inference procedure”.² Thus the majority of work in MDL uses a uniform or universal prior to represent complete prior ignorance (Rissanen 1983). Baxter and Oliver (1995) gives a more comprehensive account of the similarities and differences between the two approaches.

Since the focus of the present dissertation is not to do with either the jus-

²This was a comment made in response to a suggestion by Peter Cheeseman about the inclusion of priors into MDL at the ISIS conference in Melbourne, Wednesday 20 Aug 1996.

tification of the MML methodology nor its detailed exposition but rather with its application, we will not discuss this any further here, but carry on to see how we can apply the MML methodology to help compute the size of linguistic hypotheses.

5.4 Encoding Deterministic PFSA

If the hypothesis and the data are stated in the form of a PFSA as discussed in Section 5.1, we derive the number of bits to encode it as follows: Let the PFSA have N states and be deterministic.³ Furthermore, let t_j be the number of times the j th state is visited, V be the cardinality of the alphabet including the delimiter symbol, n_{ij} the number of times the i th symbol is produced from the j th state and m_j be the number of different transitions on non-delimiter symbols from the j th state.

The distinction between transitions and arcs which has not been explained so far must now be clarified. A transition is a single change from a source state to a target state on a given symbol. An arc is composed of one or more transitions from a given state to a given target on the same symbol. The total number of incoming transitions must be equal to the total number outgoing, but the same is not true of arcs. A state can be entered on an arc with a frequency (transition) count of, say, 10. But it may be exited with 10 arcs each with a frequency of 1. The number of arcs, however, cannot exceed the number of incoming transitions; we assume that an arc cannot have a frequency of zero.

Since the total number of incoming transitions into a state must equal the total number of outgoing ones for consistent PFSA, $t_j = \sum_{i=1}^V n_{ij}$. We use this prior knowledge about the structural constraint in the PFSA to find a partition of t_j transitions among V symbols. This is a simple problem in combinatorial analysis. The formula for this is given, for example, in Riordan (1958). We can imagine $V - 1$ identical boundaries and compute the number of ways in which

³A deterministic PFSA is defined to be one in which there is a unique transition on each symbol of the alphabet. That is, the transition function $Q \times \Sigma$ maps into Q itself, rather than its power set 2^Q , where Q is the set of states of the PFSA and Σ is its alphabet.

length of minimally encoding the hypothesis and the data in this form. As before, the base of the logarithm is the cardinality of the coding alphabet. If a binary code was used, the logs will be to the base 2 and the MML will be expressed as a number of bits. On the other hand, if the base of the logarithm is assumed to be e , as Wallace does, then the MML is computed in nits (a contraction of natural bits), a term used by Wallace and Boulton to indicate that natural logs are used instead. Abramson (1963, p.12) uses the synonymous term nats for a similar measure.

5.5 Encoding Non-Deterministic PFSA

The encoding used in the Section 5.4 will only work for deterministic PFSA.⁴ Such a PFSA is formally defined by the six-tuple $(\Sigma, Q, \delta, P, q_0, F)$ where Σ is a finite alphabet, Q is a finite set of states, q_0 is the start state, δ is the transition function $Q \times \Sigma \rightarrow Q$, P is the transition probability function $P \rightarrow N$ for arc frequencies and $P \rightarrow \{p : 0 \leq p \leq 1; p \in \mathfrak{R}\}$ for arc probabilities and F is a set of accepting states. For the kind of deterministic PFSA considered here, F is a set of states which have transitions on the delimiter symbol, i.e. $F = \{q : \delta(q, \backslash n) \rightarrow q_0\}$ where “ $\backslash n$ ” is the delimiter symbol. For such automata, there is an upper limit on the total number of arcs emanating from any given state: It cannot exceed $V = |\Sigma|$, the alphabet cardinality. Thus (5.1) which codes the number of different ways in which outgoing arcs could share the incoming transition count is valid.

Non-deterministic PFSA are formally defined by the six-tuple $(\Sigma, Q, \delta, P, q_0, F)$ where the symbols mean the same as above, except δ which is now $Q \times \Sigma \rightarrow 2^Q$, where 2^Q is the power-set of Q . Thus, the total number of outgoing arcs is limited by the number of incoming transitions only in contrast

⁴Some authors consider PFSA to be intrinsically non-deterministic. However, the sense in which they consider it so is different. In fact, the distinction between non-deterministic and deterministic PFSA used here is more consistent with the convention in Computer Science generally and agrees well with a corresponding distinction between deterministic and non-deterministic models of non-probabilistic automata. For this, see for example Aho et al. (1986, pp.113–127) or Aho and Ullman (1995, pp.537–539).

to deterministic PFSA where the number is determined by the minimum of the alphabet cardinality and the number of incoming transitions.

With this in mind, here is a first attempt at deriving the MML for describing a non-deterministic PFSA: We derive the formula in the same way as we did for the deterministic case, but we now try to compute the number of ways in which $t_j - 1$ boundaries can be inserted into a pool of t_j transitions. As before, we also code the data by naming a distribution of m_j outgoing arcs over t_j transitions. The number of bits required to code these is:

$$\log_2 \frac{(2t_j - 1)!}{m_j (t_j - 1)! \prod_{i=1}^{m_j} n_{ij}!}$$

where the terms mean the same as before except for m_j which represents the number of arcs from state j (including arcs labelled with the delimiter symbol) and n_{ij} which stands for the frequency of the i th arc from state j .

Having specified the distribution, we now need to specify the symbols labelling each of the m_j arcs. As in Section 5.4 let $M = \sum_{j=1}^N m_j$. Since it takes $\log_2 V$ bits to specify one of V symbols, it will cost us $M \log_2 V$ bits to specify the symbols labelling every arc in the PFSA.

To specify the target nodes of each of these arcs, we remember that we can use prior knowledge that the target of an arc transiting on a delimiter symbol is always the start state. Since this can be known to both the receiver and sender by prior arrangement, let $M' =$ total number of arcs in the PFSA not labelled with a delimiter symbol. The cost of specifying target states for each of these is $M' \log_2 N$ bits. Finally, as before, we subtract out the redundancy incurred by having to specify the state numbers. The total length of the message specifying the entire structure of the PFSA would then be:

$$\sum_{j=1}^N \left\{ \log_2 \frac{(2t_j - 1)!}{m_j (t_j - 1)! \prod_{i=1}^{m_j} n_{ij}!} \right\} + M \log_2 V + M' \log_2 N - \log_2(N - 1)! \quad (5.5)$$

However, this formula for computing the MML of a non-deterministic PFSA

is unlikely to be very efficient in detecting structure since there is little bias against selecting PFSA with bad fragmentation (i.e. states with a high number of outgoing transitions). Chris Wallace suggested the use of a prior for the direct specification of the m_j value for each state and the use of this m_j value in subsequent terms of the formula.⁵ To discover structure, he suggested that an assumption be made about the distribution of transition counts among the various outgoing arcs⁶ and that a prior of 2^{-n} for this purpose would be suitable, i.e. the probability that a given state has n outgoing arcs would be given by 2^{-n} . Thus the cost of specifying that a state has m_j outgoing arcs is $-\log_2 2^{-m_j}$ which is just m_j bits.

Having directly specified m_j , the number of different arcs emanating from state j , we must now specify how t_j , the incoming transition count, is partitioned among these m_j arcs and the data given this partition. To do this, we make use of another important piece of prior knowledge on a constraint upon the structure of the PFSA — no arc may have a frequency count of 0. Thus each arc must have at least one of the incoming transitions. The number of ways that the transition count can be split among m_j arcs, with each arc having at least one transition, is again a simple combinatorial problem. There are many ways of solving this; the approach taken here is to allocate one transition to each of the m_j arcs, and then partition the remaining $t_j - m_j$ transitions as before among the m_j arcs. The latter is done by the insertion of $m_j - 1$ boundaries into a pool of $t_j - m_j$ transitions and computing the number of possible permutations of this. Again, we have used another important prior that $t_j \geq m_j$ which is assumed known to both the sender and the receiver of a message encoding the PFSA structure. The length of code required to specify such a partition and the data given this partition is now:

⁵Personal communication with Chris Wallace: Comment on using priors for the number of outgoing arcs. In Discussion at Monash University, dt. 29 Aug 1996.

⁶A similar suggestion about special codes for states with a single outgoing arc was also made earlier by Peter Andreae, Computer Science department, Victoria University, during a post-seminar discussion with the author on 30 Nov 1995.

$$\log_2 \frac{(t_j - 1)!}{(m_j - 1)! \prod_{i=1}^{m_j} (n_{ij} - 1)!}$$

The remaining part of the cost is the same as in the previous case. Hence the total cost of specifying the structure of the non-deterministic PFSA is:

$$\sum_{j=1}^N \left\{ m_j + \log_2 \frac{(t_j - 1)!}{(m_j - 1)! \prod_{i=1}^{m_j} (n_{ij} - 1)!} \right\} + M \log_2 V + M' \log_2 N - \log_2 (N - 1)! \quad (5.6)$$

This formula, in fact, performs empirically better than Formula (5.5). When used to choose between alternative PFSA explaining the same dataset, it gives results that are consistent with those given by the formula derived by Wallace and Georgeff (1984) for deterministic machines. Just one interesting example from Andreae (1972, 4, p.122) that shows this is presented here. This example problem is also looked at by Gaines (1976, pp.352-354):

The following sequence of 100 coin tosses is observed from a gaming machine suspected of being rigged:

```

h:h:t:h:h:t:h:h:h:t:t:h:h:h:h:t:t:h:t:t:t:h:h:h:h:
h:t:h:h:h:t:t:t:h:h:h:t:t:t:h:h:h:h:h:h:t:t:h:t:t:
t:t:t:h:t:t:h:t:t:t:h:h:h:t:t:h:h:h:t:h:h:h:h:h:h:
h:h:h:t:t:h:h:h:t:t:t:t:h:h:h:h:h:t:t:t:t:t:t:t:h:

```

Of the various PFSA given in Figures 5.6-5.10 that explain the sequence of tosses, the four and five state machines alone capture predictable outcomes and hence discover the rigging. Readers are asked to detect this themselves before looking at the automata models which are presented at the end of this chapter. In spite of the long time since the publication of Andreae's original example, the problem still remains an intriguing one to solve.

The five state machine, however, makes no more claims than the four state one, but is more complex. One would expect the four state machine then to be the ideal compromise between theory complexity and explanatory power. This result is indeed borne out by the MML calculations of those machines, using either of the formulas (5.6) or (5.4) as shown in Table 5.1

States	MML using (5.6)	MML using (5.4)
1	113.34 bits	114.33 bits
2	114.48 bits	116.39 bits
3	115.62 bits	118.24 bits
4	<i>98.16 bits</i>	<i>107.38 bits</i>
5	104.43 bits	113.29 bits

Table 5.1: A comparison of MMLs for 1 to 5 state PFSA explaining Andreae's rigged Casino.

One further note about Formula (5.6) is that the MML derived using this is slightly smaller than the MML derived using Formula (5.4) indicating that it is able to exclude more redundancy in its measure. This is possibly due to the incorporation of the prior that every outgoing arc must have a frequency count of at least one. This is the formula that is used in subsequent sections for estimating MMLs of non-deterministic PFSA.

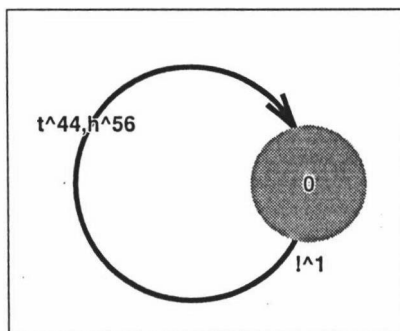


Figure 5.6: 1-state PFSA explaining Andreae's rigged Casino.

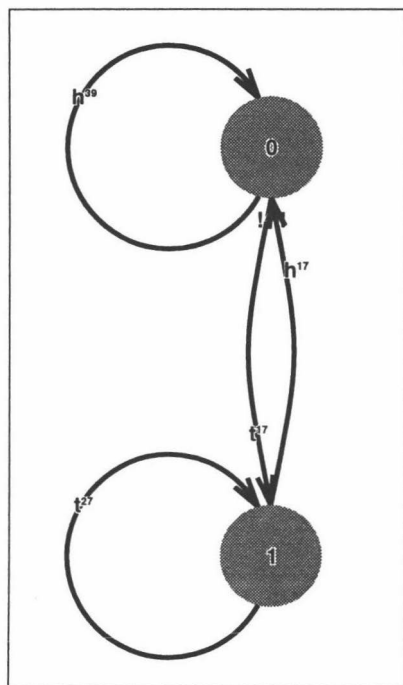


Figure 5.7: 2-state PFSA explaining Andreae's rigged Casino.

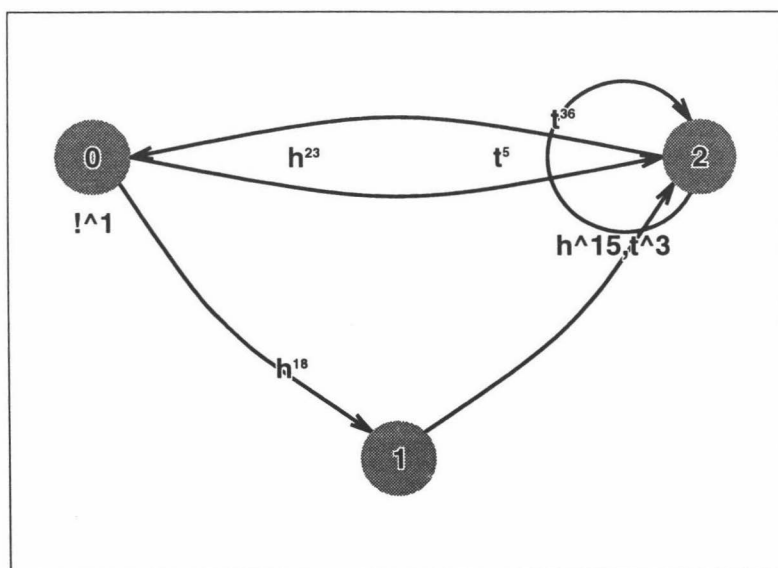


Figure 5.8: 3-state PFSA explaining Andreae's rigged Casino.

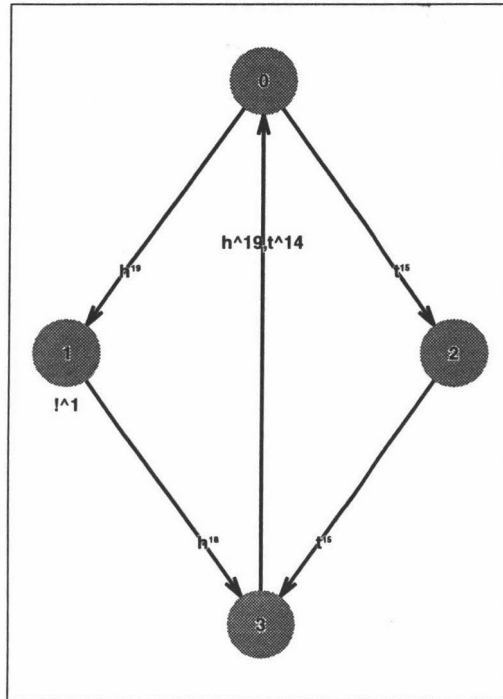


Figure 5.9: 4-state PFSA explaining Andreae's rigged Casino. Note that the output is completely predictable in states 1 and 2.

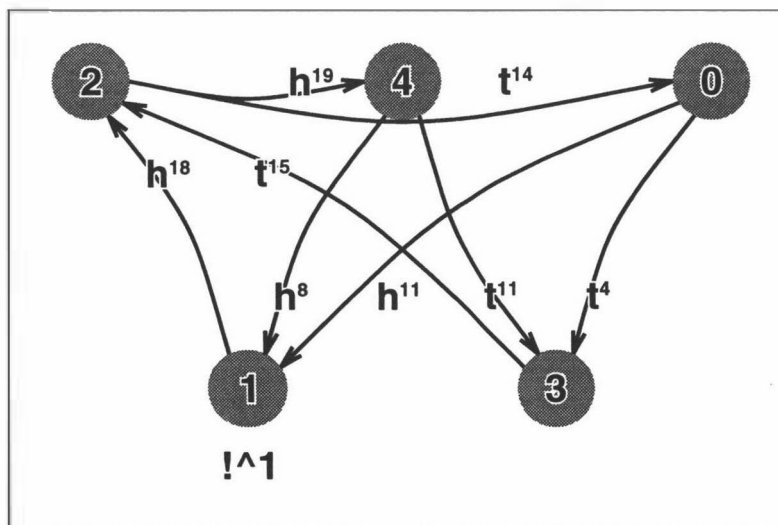


Figure 5.10: 5-state PFSA explaining Andreae's rigged Casino.

Chapter 6

The Sk-strings Method of Inferring PFSA

6.1 Introduction

When given a finite number of examples of the behaviour of a probabilistic state determined machine, it is possible to imagine methods by which we can infer its structure. When the examples are actual outputs of the machine, or sequences of symbols which will be accepted by the machine without error, they are called positive examples. If on the other hand, they consist of symbol sequences which we know could not have been generated by the machine, or if they are rejected by the machine, then they are called negative examples. Negative examples are often used to prune over-generalisations in an inferred structure, but they may not always be available. Unfortunately, also, the process of inferring the generating structure from positive examples is demonstrably intractable.

We discuss this problem in greater detail in this chapter and examine a simple, fast and easy-to-implement recursive algorithm that uses an intuitive heuristic for inferring such automata. The algorithm is a variation of the k-tails method due to Biermann and Feldman (1972). In Chapter 7, we discuss an information theoretic variation of this method and finally in Chapter 8 methods based on current mainstream search strategies, namely beam search and simulated annealing. It will be shown that the methodology outlined here which

works along intuitive guidelines performs as well as the other approaches, and is significantly faster. All the methods to be discussed will use the MML criterion described in Chapter 5 to judge the goodness of the inferred structure.

A specific example of inducing the Gaines machine is demonstrated first. This is a stochastic automaton proposed by Gaines (1976) to have generated the !-delimited string “cab!caaab!bbaab!caab!bbab!bbb!cb!”. A second machine constructed by hand is then used to generate training sequences from which we try to infer its structure using both the k-tails and the sk-strings algorithms. Finally, we benchmark the performance of these algorithms using a number of random training sequences generated by machines which have themselves been randomly generated.

6.2 Inducing the Gaines Machine

The Gaines machine is a stochastic finite automaton which purportedly is the most likely machine to have generated the strings “cab!”, “caaab!”, “bbaab!”, “caab!”, “bbab!”, “bbb!” and “cb!” where ! is the delimiter symbol. It is called so here because Gaines (1976) was the first to try to construct stochastic automata that could have generated the strings. The problem itself is a variant of one that attempts to infer a grammar given these strings of terminal symbols as positive examples. It has its origins in Feldman et al. (1969). Evans (1971) also analysed the possibility of deriving a finite-state grammar from these strings.

Gaines constructed several alternative automata, from one to seven states which could have generated the strings, but he only had intuitive grounds for choosing the 4-state model (Figure 6.1) because the improvement in entropy of the 7-state model over the 4-state model did not appear significant to him. He identifies the inference from the finite sample of an infinite number of a’s (the loop on a^9 from state 1 to itself in Figure 6.1) as the inductive leap which seems to be responsible for his choice. Thus, the 7-state model is rejected although it has the least entropy of all.

Ideally, we should like to identify the exact automaton which generated the strings. But it is impossible to do this from the behaviour of the machine be-

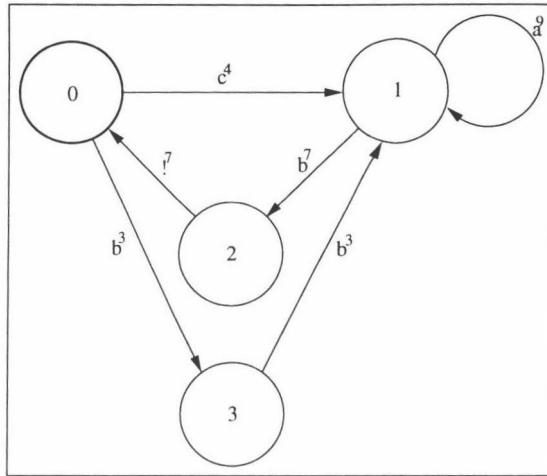


Figure 6.1: Gaines' 4 state machine.

cause more than one non-minimal machine may generate the same language. Fortunately, Occam's Razor dictates that it is enough to identify the best possible automaton we could infer from our sample strings. We assume Einstein's oft stated reassurance about Nature being intrinsically simple (Einstein 1956, p.298), which is conceptually just Occam's Razor. As to why this should really be the case is, strictly speaking, ungrounded, because there is no evidence for the non-existence of redundancy in Nature.¹ The reasons for our choice are purely pragmatic, as it lets us deal with simpler models. Thus, for example, if automata A and B are possible alternatives in Figure 6.2, we bias our selection towards the automaton A although automaton B generates exactly the same language. Our bias is justified because A is simpler and behaviorally indistinguishable from B. Such a decision is accepted scientific practice after Popper

¹If at all, there is only evidence and ample reason *for* the existence of redundancy in Nature. By Darwinian evolution or its modifications, redundancy alone is not sufficient grounds for the extinction of a feature. It has to confer a fatal disadvantage to its possessor. Here then, is an *argument from efficiency* to counter the theological *argument from design* — If indeed the universe was designed by an intelligent being, it should be much smaller than it is :-). If a biological organism, for instance, is re-engineered from scratch (using MML as a guiding principle!), its genetic material would be pruned by well over 90%, causing much smaller DNA molecules, and possibly smaller cell sizes as a consequence, contributing eventually to smaller and more compact organisms. This is, of course, just fascinating speculation on the part of the author in a field that is well outside the domain of his expertise. It belongs properly in a science fiction book which he hopes to write someday :-)

established that the most we can ever hope for is the identification of tentative theories which are consistent with current observations and choose the best among them (Popper 1969).

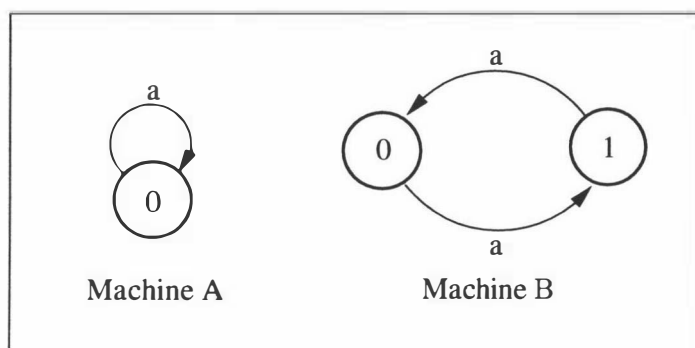


Figure 6.2: Two automata generating the same language: a^*

6.3 Minimum Message Length

In view of the above considerations, our problem then is not the identification of *the* Gaines machine, but our best determination of a machine that is most likely to have generated the sample strings. But how can we decide which among several alternative machines is the most likely one to have generated the positive examples? The best theory about the machine is not always the smallest or simplest theory as we can quickly discover. For example, Figure 6.3 shows four theories formed by observing an event sequence “abc!” exactly once.

The four state theory is the worst we can form at this stage as there is no inductive evidence to presuppose an event ordering of any kind. The single state theory which does not assign a specific order to any of the events is the most likely one and our initial guess that the smallest theory is the best seems plausible indeed. But what if the same sequence of events “abc!” is observed several times (Figure 6.4), say 100 (or even 2) times? The single state theory quickly loses plausibility, as an ordering among the events now seems justified.

An objective solution to this problem is in the estimate of the length of description of the theory together with the data it tries to explain. A good

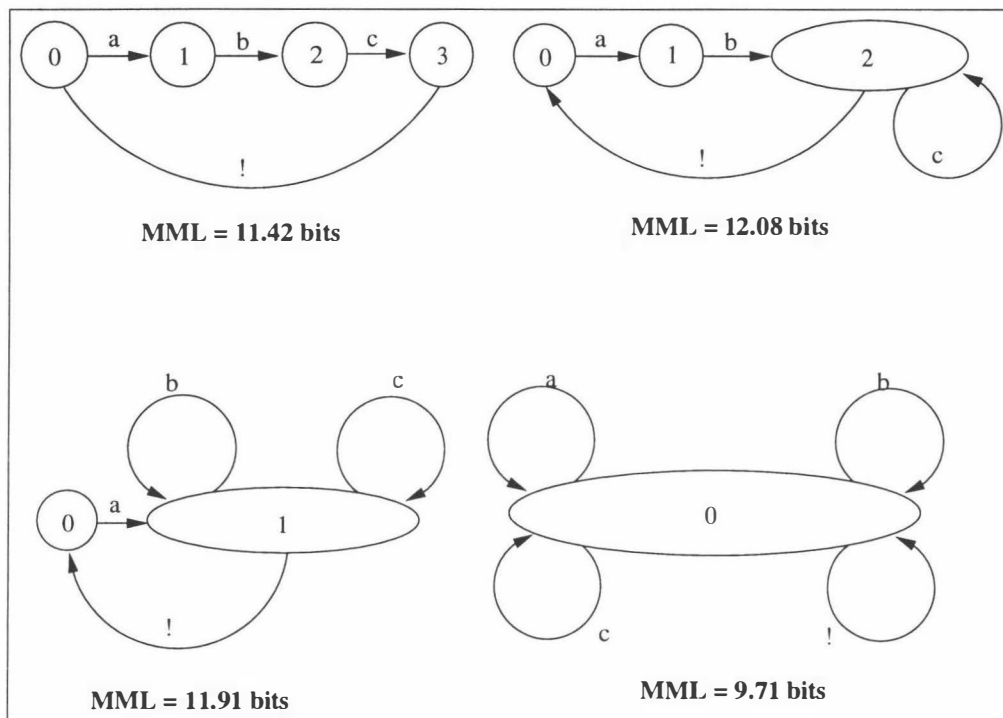


Figure 6.3: 4 theories for the event sequence “abc!” seen once.

theory would minimise such a measure. The Minimum Message Length (MML) principle of Georgeff and Wallace (1984) discussed in Chapter 5 is proposed as one such estimate. In the context of PFSA, the MML is a sum of:

- the length of encoding a description of the proposed machine
- the length of encoding the dataset assuming it was emitted by the proposed machine

Our intuitive choice of the 1-state theory in Figure 6.3 and of the 4-state theory in Figure 6.4 now seem to be objectively justified in light of this principle. We now conjecture that the 4-state Gaines machine should have a lower MML than the 7-state one. This is in fact the case. Wallace and Georgeff (1984) derive an MML of 54.1 bits for the 4-state machine and 70.0 bits for the 7-state machine.² Indeed, it so happens that the 4-state Gaines machine has got the

²Formula (5.6) gives MMLs of 46.78 bits and 65.98 bits respectively for the 4 and 7-state models.

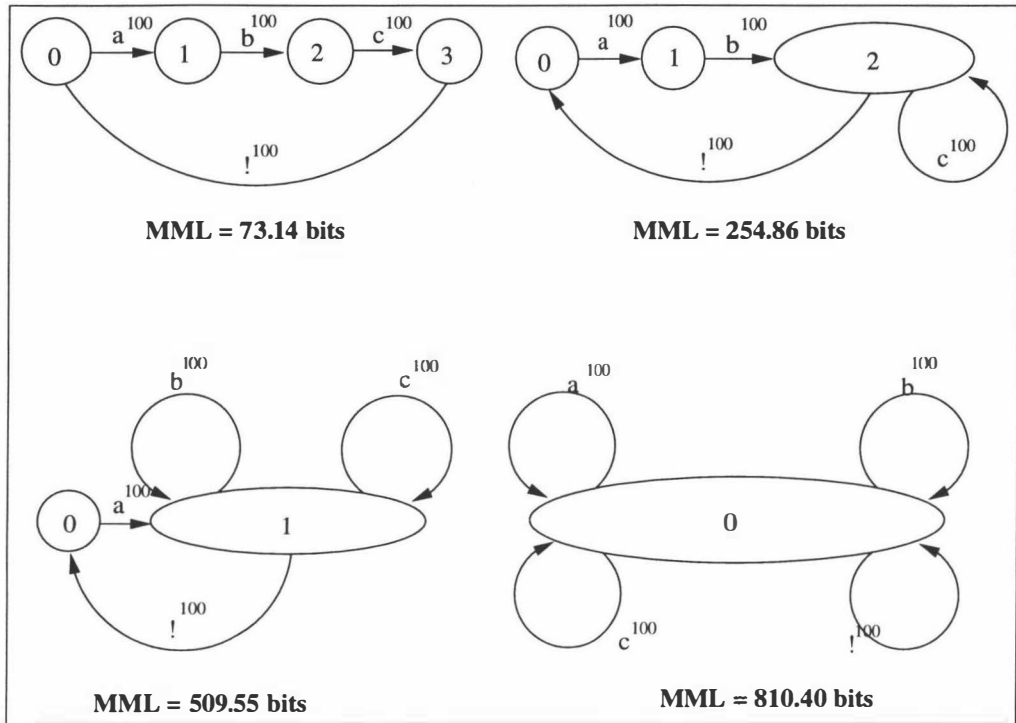


Figure 6.4: 4 theories for the event sequence “abc!” seen 100 times.

least MML of all possible machines accepting the same data. Thus Wallace and Georgeff provide objective justification for Gaines’ intuitive choice of the 4-state machine. The MMLs in this example are computed using Formula (5.4) which is explained in Chapter 5.

From this argument, we could say that in searching for the best automaton to account for a given dataset, we would like to infer a stochastic automaton with arc frequencies reflecting the absorption of the given dataset, whose description length is the minimum of all possible ones. However, the problem of inferring the best such automaton by enumeration is computationally intractable since for a given alphabet the number of possible automata with n states is exponential in n . Gold (1978), Angluin (1978) and Feldman et al. (1969) prove that the equivalent problem of finding the minimum deterministic finite automaton consistent with a set of positive and negative examples is NP-complete in the number of states of the automaton. But it is known that imposing constraints upon the grammar of the language accepted by the

automaton reduces the search space to a considerable extent. For example, Angluin (1982) defines a class of regular grammars called k -reversible, identification of which is possible in time $O(n^3)$ in the number of sentences examined. Also, Angluin (1987) shows that it is possible to exactly identify the automaton in polynomial time if the identification procedure involves a combination of passive observation and active experimentation. But what if neither is possible?

Again, it seems possible to compromise on something else in order to make the problem tractable. This time, we trade in the guarantee of a correct solution and accuracy for tractability which is achieved using heuristics. At first sight, this might seem to be rather misguided. But a little thought reveals that this is in fact very similar to the way in which we humans ourselves form theories about Nature. That is, it is accepted scientific practice to work with tentative theories which mostly over or under generalise to some extent. Methods that work along these lines usually start by building a minimal or maximal canonical automaton that is an exact representation of the input strings, and then generalise it progressively by merging states that satisfy some equivalence relation, until no more merges are possible. Attempts in this direction include the k -tails method of Biermann and Feldman (1972), the tail-clustering method of Miclet (1980), the partial similarities method of Kudo and Shimbo (1988) and the recursive merging methods of Itoga (1981) and Oncina and Garcia (1992).

The successor method of Richetin and Vernadet (1984) and the predecessor-successor method of Vernadet and Richetin (1984) are also based on equivalence relations, but they form equivalence classes of symbols and construct automata states from them. The successor method works by assigning one state to each symbol in the input alphabet from which there are transitions on every symbol that is observed to be a successor of this symbol in the set of positive examples. The target of each such transition is to the state that is assigned to the symbol on which the transition is being made. The predecessor-successor method is a modification of the former to prevent over-generalisations in the case of large sets of positive examples with low cardinality alphabets where there is little symbol repetition. It does this by assigning different numbers to multiple oc-

currences of the same symbol in the input alphabet, except within runs. This effectively expands the input alphabet and hence the number of states of the inferred automaton. Also, the method initially forms equivalence classes by comparing the predecessors of every symbol in the new alphabet instead of its successor. When the formation of equivalence classes thus stabilises, the successor method is applied after collapsing the expanded alphabet back into the original one.

The k -tails method works by merging states that are indistinguishable in the set of accepted output strings up to a given length. This is discussed in greater detail in the next section. In Miclet's tail-clustering method, two states in the maximal canonical machine are merged if a distance measure between their sets of tails is below a certain limit. This distance measure is a function of the the number of common and different tails between them. The partial similarities method starts by constructing sets of predecessor and successor substrings of length k for each state. A number of different equivalence relations are then defined based on these. For example, two states may be considered mergeable if they have the same set of predecessor k -substrings, or if they have the same set of successor k -substrings, or either, or both, etc. The recursive merging method due to Oncina and Garcia merges two states in a lexicographically ordered prefix tree acceptor of the positive examples if there exists transitions from a common intermediate state to both of them and if the merged automaton does not accept any negative examples which must also be given. Furthermore, a state is only considered for merging with the lowest numbered state possible. This is not, strictly speaking, a heuristic method since it has been shown to guarantee automaton identification in the limit.

It might appear from the previous discussion that generalisations such as those described above could also be guided by some criterion to reject undesirable merges. Maryanski and Booth (1977), for example, propose the chi-square metric as a measure of the goodness of fit of the inferred automaton to the given data.³ MML has also been proposed as such a criterion, i.e. merges which result

³See Gaines (1978) for a correction.

in an increase of the automaton's MML should be rejected. Patrick and Chong (1987) employ this method, by coalescing chains of states separated by common transitions and accepting the resultant automaton as an intermediate in the derivation process only if it resulted in a decrease in MML. Such a method has also been described as one of the most straightforward ones (Bell et al. 1990). Indeed, the procedure could even consist of blindly merging every pair of states possible and checking if the resultant automaton has a lower MML. Oliver (1993) uses a similar greedy algorithm to infer decision graphs. But the method is not likely to always succeed, for it is known that in several cases a good overall merge might consist of at least some bad component merges. This is a pity, because otherwise, we could correctly identify the best model in $O(n^3)$ time.

Thus we may defer the MML test until a final model is reached and guide the intermediate merges with heuristics such as those discussed earlier. The MML is now used as a measure of error in the inference process. If the final model doesn't give a better MML we could try applying a different set of heuristics to the canonical machine or the same heuristic again with different values for its tunable parameters if applicable.

6.4 The K-tails Approach

The k-tails approach ascribed to Biermann and Feldman (1972) is a modification of the Nerode (1958) realisation technique for the synthesis of finite state machines. It is one of the earliest algorithms for inducing finite state machines from positive examples of their behaviour. Several other techniques mentioned above, as is the one proposed in this chapter are all modifications of this in one form or other.

If the minimal canonical automaton is defined by $A_c = (\Sigma, Q_c, \delta_c, q_{0c}, F_c)$ where Σ is a finite number of input symbols (the alphabet), Q_c is a finite set of states, δ_c is the transition function $Q_c \times \Sigma \rightarrow Q_c$, q_{0c} is the start state and F_c is the set of accepting states, we define $k\text{-tails}(q)$ to be the set $\{ z \mid z \in \Sigma^* \wedge |z| \leq k, \delta(q, z) \cap F_c \neq \emptyset \}$ where $|z|$ stands for the length (num-

ber of symbols) of z .

The k -tails algorithm attempts to construct a non-deterministic finite state automaton by successively merging states of the canonical automaton that are indistinguishable for k -tails. The formal equivalence relation R_k is: for $(q, q') \in Q_c$, $(q, q') \in R_k \iff \forall t : t \in k\text{-tails}(q) \iff t \in k\text{-tails}(q')$.

The algorithm is easy to implement and the value of k may be adjusted to suit the particular inference strategy. For low k , the algorithm overgeneralises while at higher values of k , the inferred automaton has a better entropy. It can also be shown (Aho and Ullman 1972) that at $k \geq N - 2$, where $N =$ size of Q , R_k reduces to the Nerode equivalence relation, i.e. $\forall m \geq N - 2 : R_m = R_\infty$.

Some motivation for modification to this method comes from the observation that stochasticity is intrinsic to the natural inductive process. We often form generalisations from specific instances, and in doing so, we may even dismiss certain anomalies as noise. The k -tails method will not attempt to merge two states which have even a minor disagreement between them. This is because it was originally intended for non-stochastic automata where each transition out of a state is equally likely. For instance, if there are k -tails beginning with 'a' and 'b' from state 1 and with 'a' from state 2 with transition probabilities as shown in Figure 6.5, 1 and 2 will not be merged by the k -tails method because at least some k -tails (those that begin with 'b') of state 1 are not k -tails for state 2, although it seems intuitive that the two states have a lot in common and should be candidates for a possible merge.

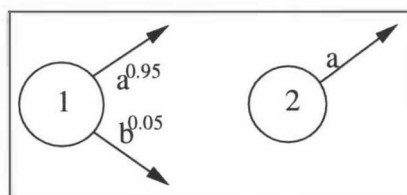


Figure 6.5: 2 candidate states for merging in a PFSA.

That the two states should be merged is also suggested from the MML

criterion as we can see in Figure 6.6. A canonical machine and two reductions are shown. The MML of the automaton where states 1 and 2 were merged is only 80.08 bits, whereas if they are not merged, the MML is 88.05 bits.

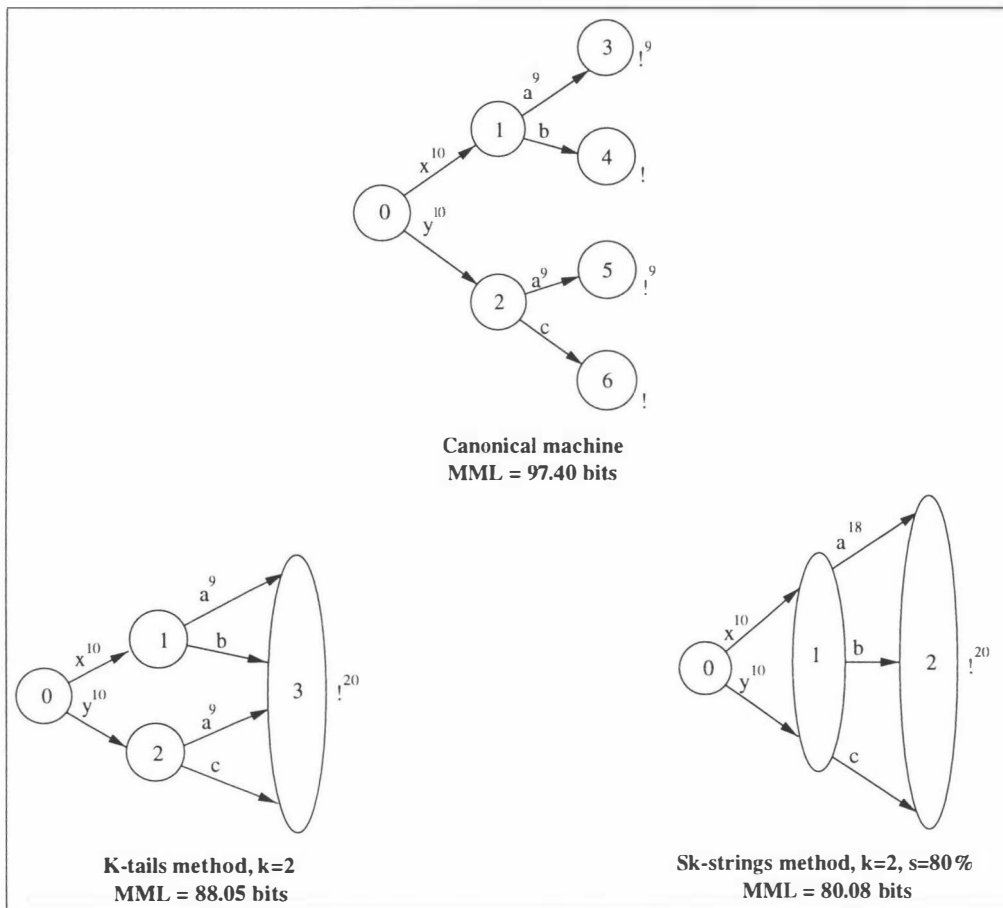


Figure 6.6: K-tails and sk-strings reductions of a canonical machine.

6.5 The Sk-strings Approach

The sk-strings method is a modification of the k-tails method for stochastic automata. It constructs a non-deterministic finite state automaton by successively merging states of the canonical machine that are indistinguishable for their top s percent of the most probable k -strings.

We define a k -string to be the same as a k -tail, except that it does

not have to end at a finish state unless it is shorter than the specified string size. That is, the set of k -strings of state q k -strings(q) is defined to be the set $\{z \mid z \in \Sigma^*, |z| = k \wedge \delta(q, z) \subset Q \vee |z| < k \wedge \delta(q, z) \cap F_c \neq \emptyset\}$. Each such string will have a probability associated with it which is given by the product of the probabilities of each arc traversed in generating that string.

We then define sk -equivalence such that two states are considered mergeable if the sets consisting of the top s percent of their distribution of k -strings are the same. This is computed as follows: The k -strings of a state are arranged in decreasing order of probability. The top n of these strings, whose probabilities add up to s percent or more with n being as small as possible are retained, and the remaining strings are ignored. At least one string from each state will be considered with this approach; thus s is never exactly 0 in practice.

If $s \leq 95\%$ and $k = 1$, for example, then the two states 1 and 2 in Figure 6.6 will be sk -mergeable, but they will not be if $s > 95\%$. In general, when s is low, the method overgeneralises, and with $s = 100\%$, the method requires that all the k -strings of the two states must agree. With $s = 100\%$ and a sufficiently high value for k , the method reduces to the Nerode equivalence relation. Obviously, the heuristic can operate with various degrees of strictness as below:

- The top $s\%$ of the k -strings of state 1 are k -strings at state 2 OR vice-versa. (The OR heuristic)
- The top $s\%$ of the k -strings of state 1 are k -strings at state 2 AND vice-versa. (The AND heuristic)
- The top $s\%$ of of the k -strings of state 1 is the same set as the top $s\%$ of the k -strings of state 2, in the same order, but perhaps with a different probability distribution. (The LAX heuristic)
- The top $s\%$ of the k -strings of state 1 is the same set as the top $s\%$ of the k -strings of state 2 in the same order, with the same probability distribution. (The STRICT heuristic)

Algorithm 6.5.1 gives the procedure for minimising an N -state canonical machine using an sk -strings equivalence relation.

Algorithm 6.5.1 (sk-strings)

```
1 begin
2   for  $p := state_0$  to  $state_N$  do
3     for  $q := p$  to  $state_N$  do
4       if  $sk\_equivalent(p, q)$ 
5         then
6            $merge(p, q);$ 
7            $p := state_0;$ 
8            $q := state_1;$ 
9         fi
10      od
11   od
12 end
```

Algorithm 6.5.2 gives the equivalency decision procedure using the AND heuristic. The procedure to determine equivalency using the other heuristics are similar.

Algorithm 6.5.2 (Sk-equivalence using AND Heuristic)

```
1 comment: Determines if two states  $p$  and  $q$  are sk-equivalent
2 comment: Returns TRUE if they are, FALSE otherwise
3 comment: Global: Agreement% (Equals parameter  $s$  to sk-strings)
4 begin
5    $counter := 0;$ 
6    $i := 0;$ 
7    $S_p :=$  List of (string,probability) pairs output from state  $p$ ;
8    $S_q :=$  List of (string,probability) pairs output from state  $q$ ;
9   Sort  $S_p$  in descending order of  $S_p.probability$ ;
10  Sort  $S_q$  in descending order of  $S_q.probability$ ;
11  for  $i := 1$  to  $num\_strings\_in(S_p)$  do
12     $counter := counter + S_p[i].probability;$ 
13    if  $notacceptable\_at(q, S_p[i].string)$ 
14      then  $return(FALSE);$ 
15    fi
16    if  $counter \geq Agreement\%$ 
17      then
18         $counter := 0;$ 
19        for  $i := 1$  to  $num\_strings\_in(S_q)$  do
20           $counter := counter + S_q[i].probability;$ 
21          if  $notacceptable\_at(p, S_q[i].string)$ 
22            then  $return(FALSE);$ 
23          fi
24        if  $counter > Agreement\%$ 
```

```

25         then return(TRUE);
26         fi
27         od
28         return(TRUE);
29     fi
30     od
31     return(FALSE);
32 end

```

The efficiency of this process can be improved significantly by cacheing the sorted list of k-strings of the argument states. The cache only needs to be flushed when a merge affects the output strings and/or probabilities of upstream states. The function *num_strings_in()* returns the number of strings in the list. The algorithms for generating the list of strings and for sorting them are relatively standard and straightforward and hence not given here. However, the acceptability criterion in Algorithm 6.5.2 is given in Algorithm 6.5.3 where δ is the usual transition function for non-deterministic automata $Q \times \Sigma \rightarrow 2^Q$. A string is considered acceptable if there is a transition on it to at least one state of the automaton. The algorithm returns TRUE if string z is acceptable at state p and FALSE otherwise.

Algorithm 6.5.3 (Acceptability Criterion)

```

1 comment: Returns TRUE if string  $z$  is acceptable at state  $p$ , FALSE otherwise
2 begin
3   if  $\delta(p, z) = \emptyset$ 
4     then return(FALSE)
5     else return(TRUE)
6   fi
7 end

```

6.6 Tractability Considerations

The space and time requirements of the equivalency decision procedure depends on the number of strings that need to be considered and compared between the two candidate states. Since the latter is exponential in tail size, sk-strings is essentially an exponential algorithm. For canonical automata this doesn't matter

as much as automata in which some generalisations have been performed since canonical automata typically represent a finite number of strings. Efficiency of the algorithm suffers only when we are dealing with large tail sizes and automata which have cycles and thus represent infinite languages, but then these are inevitable during the process of making generalisations.

Thus for tail sizes of 2 and above, the algorithm complexity increases rapidly and makes the exact implementation infeasible. Ideally, we should build strings in decreasing order of probability and stop when the sum of probabilities of strings built up to that point exceeds s . But this is not possible unless all strings have been examined. Therefore we need to ignore some strings at least. The following procedural alternatives were considered to limit the number of strings considered to some reasonable value.

6.6.1 Method A - Build strings in decreasing order of probability

An upper limit is put on the number of strings we are willing to consider, say N . At each state, we traverse transitions in decreasing order of probability and thus build up the string list recursively. We stop when the N strings have been accumulated in our list. This is clearly infeasible for the following reason: Suppose that state 0 has a transition of probability 0.9 to state 1 and of probability 0.1 to state 2. Suppose also that state 1 has a transition to a finish state with a prob of 0.99 and 10 other transitions, also to finish states, of probability 0.001 each. Now suppose that state 2 has one transition with a probability of 0.9 and one with a probability of 0.1 both to finish states. Then the string list should ideally include one string with a probability of 0.9×0.99 , one string with probability 0.1×0.9 and one with 0.1×0.1 . However this will not be the case since all the 10 strings of probability 0.9×0.001 strings will be added to the string list before the strings of probability 0.1×0.1 are. If the string list should fill up before considering state 2, then some undeserving candidates would have been included while some deserving candidates were left out.

6.6.2 Method B - Reject strings less than a certain probability

Strings less than a certain threshold of probability are rejected. Consider a state with one or more low probability transitions ($< 1\%$, for example). If the selected threshold is 1% we ignore these transitions in building strings to be considered from this state. Now either this state has other higher probability transitions, or not. If it has other high probability transitions, then there will either be many moderately high probability transitions or a few high probability transitions. In either of these cases, these high probability transition(s) determine the character of this state and not the low probability ones. Thus it is safe to ignore the low probability transitions. However, if there are no high probability transitions from this state, then there are many (at least 100 in this case) low probability transitions. These strings will all have to be considered in merging this state with another similar state. However, this is an unlikely situation, thus this heuristic can be said to perform satisfactorily. Furthermore, it is reasonably safe to conjecture that no worthwhile induction can be performed on the output of a state with so much information as there is little reason to favour any one of many approximately equiprobable output strings.

6.6.3 Method C - Reject relatively improbable strings

Strings that are less than an n th (say $n = 10$) as probable than the most probable k -string from that state are rejected. This is a modification of the previous heuristic in which low probability strings were rejected, but the rejection is now based on the relative probability of a string compared with the most probable string from that state as opposed to its absolute probability. The complexity of finding the most probable string from a state, however, is exponential in the length of the string, since all possible strings need to be considered. It won't do just to follow the maximum frequency transition at each state because of the possibility of bad fragmentation down the line. For example, we might choose to follow a transition of probability 0.6 as opposed to an alternative one of probability 0.4 from a state, but the latter transition may lead directly to a finish state, whereas the former may be badly fragmented into 10 transitions of

probability 0.1 each at the very next state. Our initial guess would only have yielded a string of probability 0.06 at best, whereas if we had chosen otherwise, we would have obtained a string of probability 0.4.

Based on these considerations, method B was implemented in the actual program to limit the number of strings considered. A further advantage with this approach is that it guarantees that no more strings than the reciprocal of the threshold will be considered and we can build our internal buffer sizes in the program accordingly. The maximum number of strings that can be produced from a state with probabilities greater than or equal to 0.001 is 1000. In practice, this proves to be a satisfactory value large enough to make generalisations and decisions about merging two states. For more extreme cases, the program user is given the option to change the limit if necessary. Thus the threshold was given a default value of 0.1%. Since no string less than 0.1% in probability was now being considered, significant savings in running time and space were observed compared to earlier runs when the string generation was unlimited. In particular, further exploration of any path could now be abandoned if the probability of being at the current state on it was found to be less than this threshold.

6.7 Results

The program reported here implements all four of the *sk*-strings heuristics. The AND, OR, LAX and STRICT heuristics were first compared with each other using two sample training sets using values of *s* (Agreement%) from 1% to 100% and values of *t* (Tail size) from 1 to 10. The procedure involved first building the minimal canonical automaton from the training set and then performing the *k*-tails and *sk*-strings generalisations on it. The MML recorded is the MML of the resultant PFSA.

The first training set is the same as the one used by Gaines (1976). The second one is a set of 100 randomly generated strings from the machine in Figure 6.7 which was constructed by hand to resemble on a smaller scale the actual linguistic data to be processed. The results are presented in tabular

form (Tables 6.1 – 6.8 where the cells list the MML of the automaton inferred) and graphical form (Figures 6.8 – 6.11) where the vertical axis lists the MML inferred for each combination of tail size and agreement percentage which are represented in the two horizontal axes.

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	531	531	531	531	531	531	827	827	827	827
...	Previous line repeats 28 times									
30	531	555	531	531	531	531	827	827	827	827
...	Previous line repeats 23 times									
54	531	555	555	531	827	827	827	827	827	827
...	Previous line repeats 2 times									
57	555	555	555	531	827	827	827	827	827	827
...	Previous line repeats 11 times									
69	555	583	531	531	531	827	827	827	827	827
...	Previous line repeats 20 times									
90	587	586	573	573	586	586	586	586	586	586
...	Previous line repeats 4 times									
95	683	650	610	610	623	623	623	623	623	623
...	Previous line repeats 5 times									

Table 6.1: MMLs of 1000 inferred automata using the AND heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	531	531	531	531	531	688	3151	3151	3151	3151
...	Previous line repeats 93 times									
95	583	583	583	583	583	583	583	583	583	583
...	Previous line repeats 5 times									

Table 6.2: MMLs of 1000 inferred automata using the OR heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).

Since sk-strings is a reasonably fast algorithm, it is economical to run several of its variations and choose the best among them. In the following set of experiments, the best machine inferred by the sk-strings algorithm using any of its four heuristics and various tail sizes using a fixed value of $s = 50\%$ is

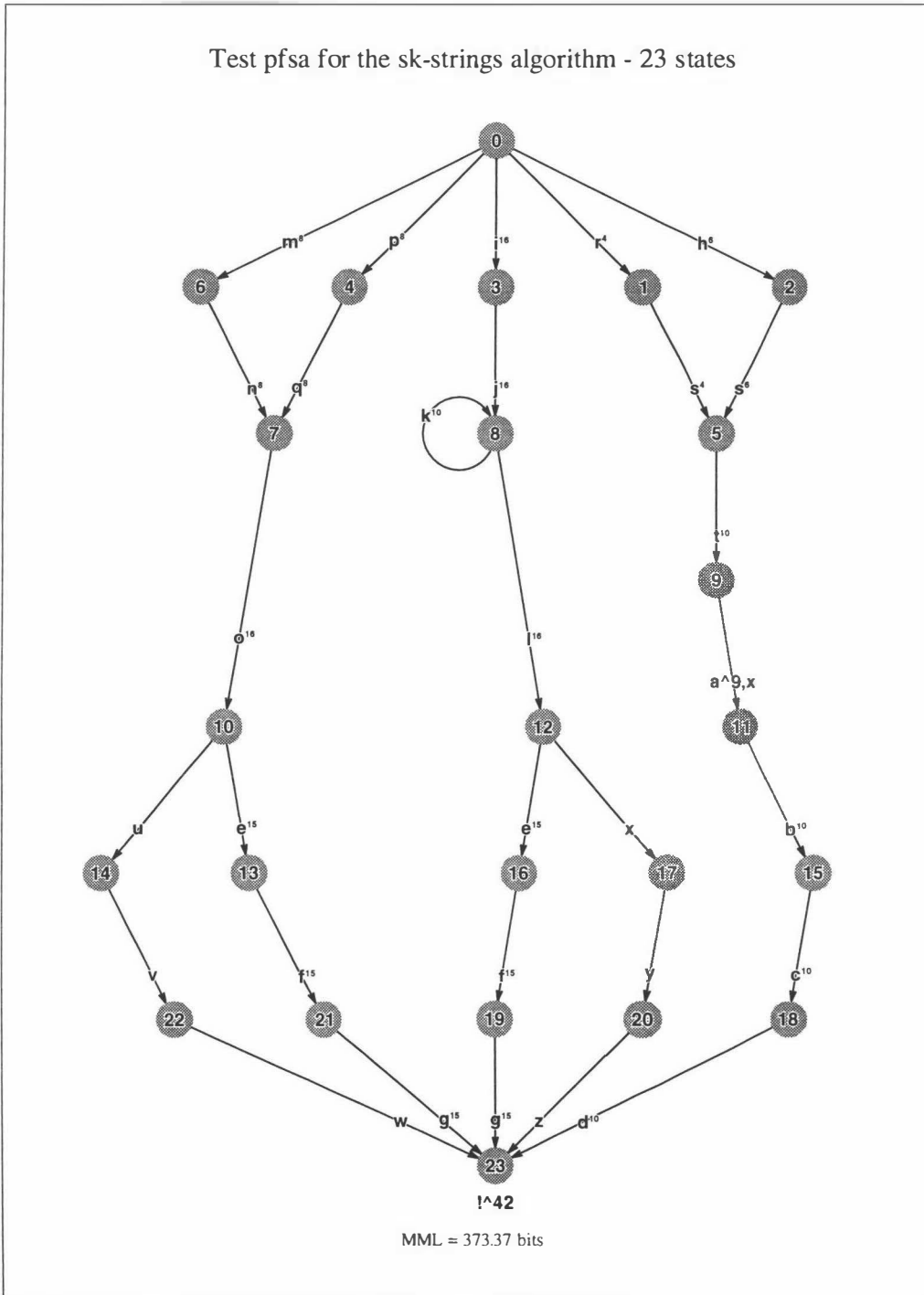


Figure 6.7: Test machine used to generate strings for training set 2.

compared with the best machine inferred by the k-tails method. One hundred small automata were randomly generated, each having from 1 to 6 states, up

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	563	569	569	569	569	569	569	569	569	569
...	Previous line repeats 99 times									

Table 6.3: MMLs of 1000 inferred automata using the LAX heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	700	668	641	641	641	641	641	641	641	641
...	Previous line repeats 99 times									

Table 6.4: MMLs of 1000 inferred automata using the STRICT heuristic for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1).

to 10 arcs and with an alphabet size of 5. A training set of 10 strings was randomly generated from each automaton and this was used to infer back the generating automaton. The results of this experiment are shown in Figure 6.12. The horizontal axis represents the training sets and the vertical axis gives the

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	62.37	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78
...	Previous line repeats 34 times									
36	62.37	46.78	59.03	59.03	59.03	59.03	59.03	59.03	59.03	59.03
...	Previous line repeats 13 times									
50	62.37	67.73	67.73	67.73	67.73	67.73	67.73	67.73	67.73	67.73
...	Previous line repeats 7 times									
58	62.37	67.73	59.03	67.73	67.73	67.73	67.73	67.73	67.73	67.73
...	Previous line repeats 8 times									
67	62.37	67.73	74.37	67.73	67.73	67.73	67.73	67.73	67.73	67.73
...	Previous line repeats 7 times									
75	62.37	67.73	65.98	65.98	65.98	65.98	65.98	65.98	65.98	65.98
...	Previous line repeats 25 times									

Table 6.5: MMLs of 1000 inferred automata using the AND heuristic for various tail sizes and agreement percentages (Training set 2).

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	65.81	53.27	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78
...	Previous line repeats 41 times									
43	65.81	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78
...	Previous line repeats 23 times									
67	69.63	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78
...	Previous line repeats 33 times									

Table 6.6: MMLs of 1000 inferred automata using the OR heuristic for various tail sizes and agreement percentages (Training set 2).

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	69.92	54.60	46.78	46.78	46.78	46.78	46.78	46.78	46.78	46.78
...	Previous line repeats 99 times									

Table 6.7: MMLs of 1000 inferred automata using the LAX heuristic for various tail sizes and agreement percentages (Training set 2).

error in the inferred automaton computed using $1 - i/a$, where i is the MML of the inferred machine and a is the MML of the generating machine after being adjusted for exactly the emitted data. Negative errors indicate the inference of more compact PFSA than the generating ones which are consistent with the emitted data. This is due to the existence of some redundancy in the randomly generated structures.

Table 6.9 shows the success rate of the various tail sizes — each cell shows how many times out of 100 that particular tail size was the one that inferred the automaton with the least MML. We see that low tail sizes seem to be preferable

Tail Agr%	1	2	3	4	5	6	7	8	9	10
1	69.93	65.98	65.98	65.98	65.98	65.98	65.98	65.98	65.98	65.98
...	Previous line repeats 99 times									

Table 6.8: MMLs of 1000 inferred automata using the STRICT heuristic for various tail sizes and agreement percentages (Training set 2).

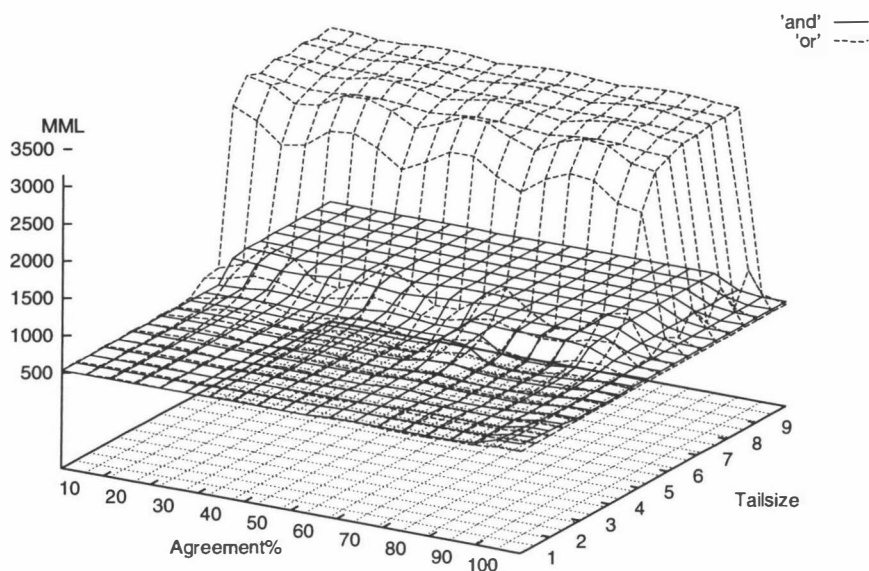


Figure 6.8: MMLs of 1000 inferred automata using the AND/OR heuristics for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1). See also Tables 6.1 and 6.2.

with both the k-tails and the sk-strings algorithms.

Tailsizes	K-tail	AND	OR	LAX	STRICT
1	75	75	77	78	64
2	19	15	11	12	22
3	3	6	4	5	6
4	1	3	6	4	6
5	2	1	2	1	2
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0

Table 6.9: The number of times (out of 100 training sets) that each tail size succeeded in inferring the minimum automaton using K-tails and the various sk-strings heuristics.

Table 6.10 shows the success rates of each of the 4 sk-strings heuristics and

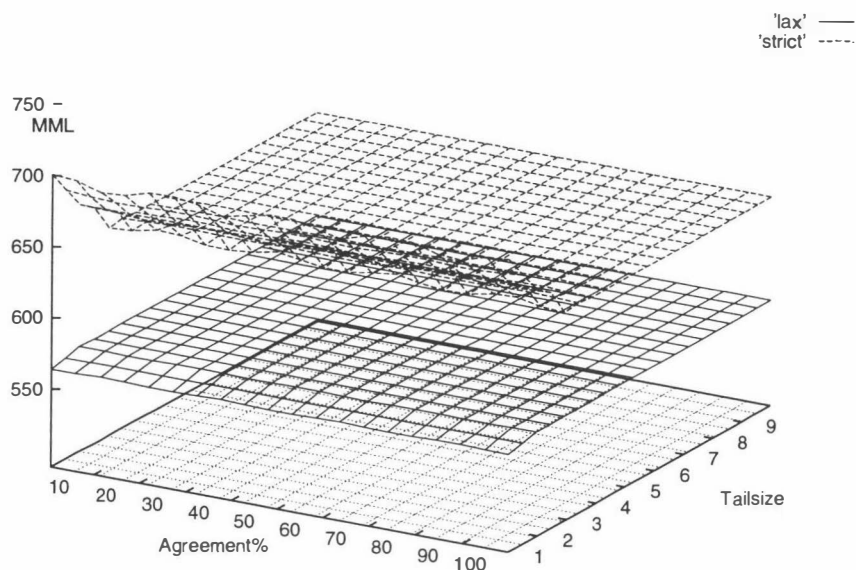


Figure 6.9: MMLs of 1000 inferred automata using the LAX/STRICT heuristics for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 1). See also Tables 6.3 and 6.4.

the k-tails algorithm in inferring the best PFSA from the training sets generated by 100 small automata. Column 1 shows the number of times out of 100 the algorithm in column 2 was the one that inferred the best PFSA. When more than one algorithm is listed, they had both inferred automata with the same MML.

Success Rate	Algorithm(s)
73	OR
12	KTAIL,OR
7	KTAIL
6	LAX
1	KTAIL,AND,LAX
1	AND

Table 6.10: Success rates of the k-tails and 4 sk-strings heuristics in inferring the best PFSA out of 100 small test cases.

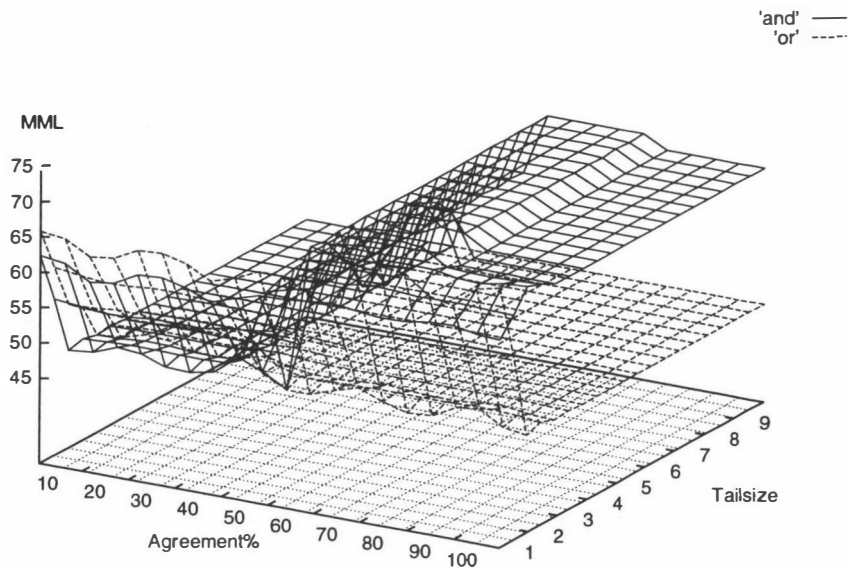


Figure 6.10: MMLs of 1000 inferred automata using the AND/OR heuristics for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 2). See also Tables 6.5 and 6.6.

The experiment was repeated with larger random automata with up to 100 states, 500 arcs and an alphabet size of 25, using 25 randomly generated strings in each training set. Each process given a maximum time of three minutes to run after which the inference procedure on that training set was killed. Only tail sizes of 1–5 were used as compared to tail sizes of 1–10 in the case of smaller automata. The results of this experiment are shown in Figure 6.13.

Table 6.11 shows the success rates of each of the 4 sk-strings heuristics and the k-tails algorithm in inferring the best PFSA from the training sets generated by 100 small automata.

6.8 Conclusion

The sk-strings approach to inducing PFSA has been described here. Its performance has been compared with the earlier k-tails method. The particular

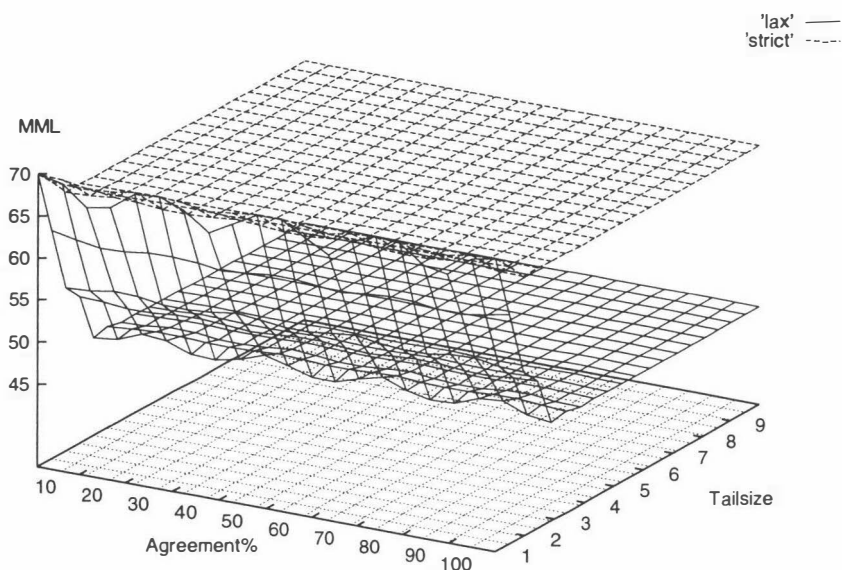


Figure 6.11: MMLs of 1000 inferred automata using the LAX/STRICT heuristics for tail sizes of 1–10 and agreement percentages of 1–100 (Training set 2). See also Tables 6.7 and 6.8.

Success Rate	Algorithm(s)
74	KTAIL
26	OR
0	Others

Table 6.11: Success rates of the k-tails and 4 sk-strings heuristics in inferring the best PFSA out of 100 large test cases.

innovations are in including the stochasticity of the machine as a determinant in the equivalence relation and using the MML criterion to judge the results. The decision to ignore low probability strings emitted from a state has also markedly improved the speed of the algorithm.

Experimental results bear out the interesting fact that of the four sk-strings heuristics described in this chapter, the OR heuristic, which is the weakest condition for merging, seems to out-perform the other three in both the small

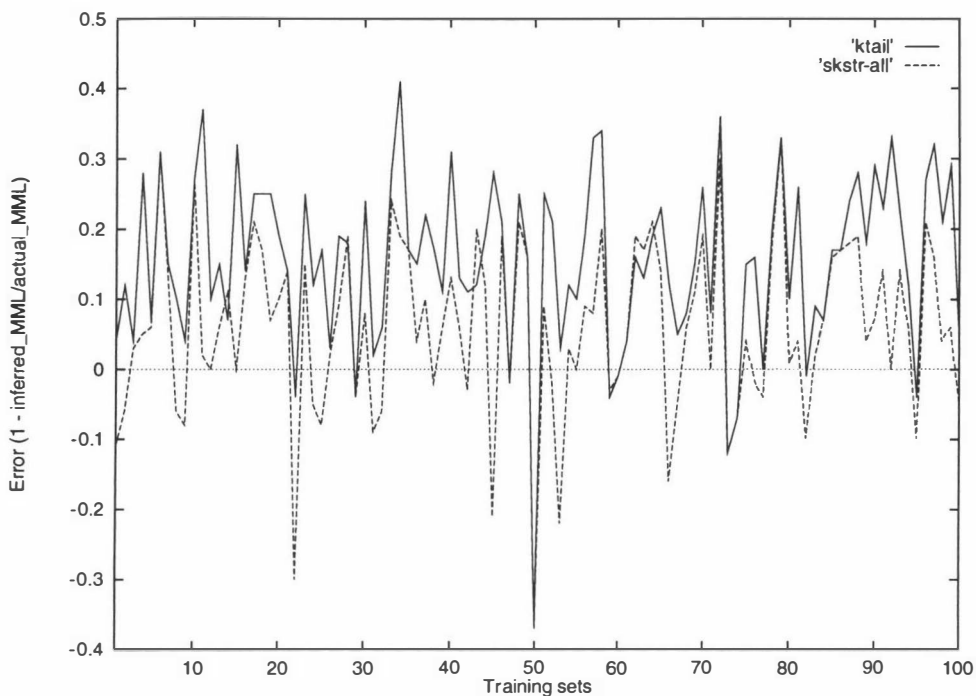


Figure 6.12: The k-tail method compared with sk-strings ($s=50\%$) using 100 random training sets generated by small random automata (Best machines inferred using any heuristic and tail sizes from 1 to 10).

and large test cases. They also show that for small random automata, sk-strings performs as well as k-tails, and collectively with its four heuristics for merging, it is able to infer better automata on average than k-tails for various values of tail size. For large automata, k-tails seems to perform better, but it requires much more time to run than sk-strings. On a Sun Sparcserver 1000 with 256 Megabytes of main memory, each run of the sk-strings program on small and large random automata took no more than about 0.3 seconds regardless of tail size, but the running time of k-tails increased exponentially with tail size for most automata. It is possible that these automata elicited the worst-case behaviour of k-tails by having states with several self transitions. This worst-case scenario, however, did not present a problem for sk-strings since it puts an artificial limit on the least probability of a generated string it is able to consider. A detailed investigation of the actual reasons for the disparity in the relative performance of sk-strings with respect to k-tails in the case of small and

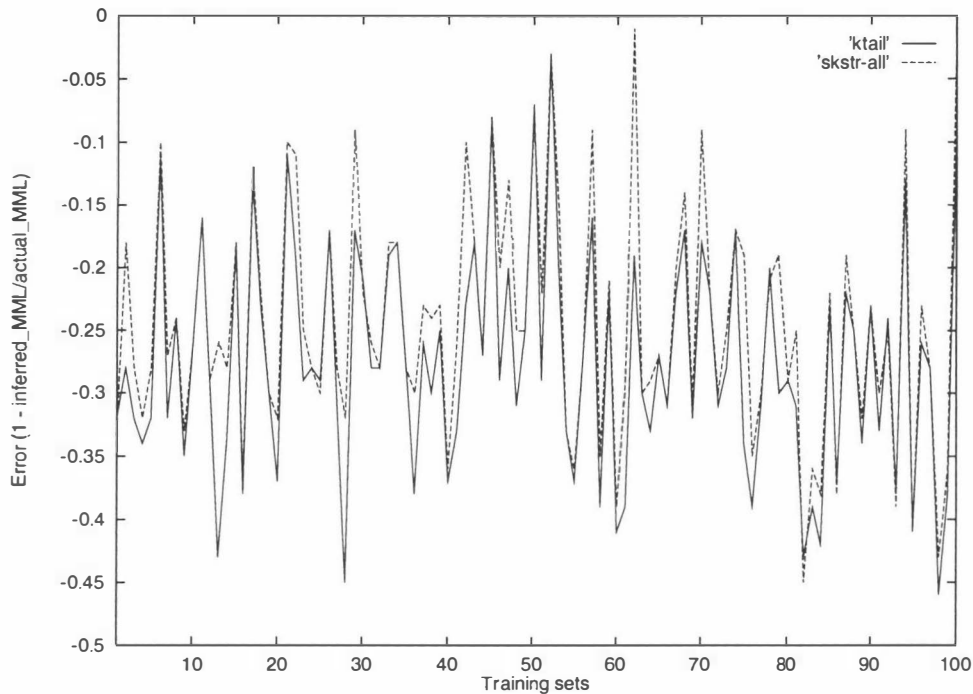


Figure 6.13: The k-tail method compared with sk-strings ($s=50\%$) using 100 random training sets generated by large random automata (Best machines inferred using any heuristic and tail sizes from 1 to 5).

large automata is left for future work, but it is suspected that it may be due in part to the limit on the number of strings actually considered by sk-strings in the process of comparing two states. There are also reasons to believe that an information theoretic heuristic for sk-strings, such as one based on cross-entropy may enhance its performance significantly in the case of large automata. This is to be addressed separately in the next chapter.

In summary, it is seen that sk-strings is a fast and efficient algorithm for inferring PFSA, and compares well with the related k-tails method for non-stochastic automata. While it is not obvious from the experiments which of the four sk-strings heuristics is the best, it seems likely that there are situations when one particular heuristic may be preferable to the others. In any case, the sk-strings algorithm is polynomial in space and time and it will still be economical to run all possible heuristics on the input data and choose the best automaton inferred from them. This is the approach that has been followed in

inferring generalised automata from the linguistic data in this dissertation.

Chapter 7

A Heuristic Using Cross-Entropy for Sk-Strings

7.1 Introduction

Chapter 6 discussed the sk-strings algorithm to infer PFSA from positive examples of its language. Four different intuitive heuristics to determine equivalency were presented in the discussion of sk-strings. Since the algorithm essentially only looked at strings output from a given state (not strings leading to a given state), the degree of similarity depended upon the difference between two probability distributions over a single set of strings S , where $S = S_1 \cup S_2$, S_i being the set of strings output from state i . It is easy to see how this problem is one of determining a distance measure between two probability distributions over a single set of points. Here a fifth possible heuristic is presented based on information theoretic principles (as opposed to the previous intuitive ones) to decide the equivalency of two candidate states. This is compared with the earlier intuitive heuristics and also with a method that uses the variational distance between the distribution of strings generated from two states to determine equivalency.

7.2 A Cross-Entropic Measure

In sk-strings, the intuitive decision to merge two candidate states can itself be viewed as an inductive inference process because it is tantamount to a general claim that these two states are identical based on their expected behaviour (the distribution of strings output from them). It has been demonstrated by Shore and Johnson (1980) that the use of any information-theoretic criterion other than cross-entropy for inductive inference when new information is in the form of expected values will lead to inconsistency in the result. Accordingly, cross-entropy (Good 1963) is used here to determine the similarity of two probability distributions p and q , over the set of strings S .

First proposed by Kullback (Kullback 1959; Kullback and Leibler 1951), cross-entropy, which is also called the mean information for discrimination or I directed divergence has been applied in various fields ranging from statistical mechanics to spectral analysis (Shore and Johnson 1980). In its original form, it provides a numerical estimate for discriminating in favour of hypothesis H_1 , against hypothesis H_2 , when H_1 and H_2 imply probability distributions p and q respectively over a space ω of points $\omega \in \Omega$. This information is defined by

$$I(p : q) = \sum_{\Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)}$$

when the space Ω is discrete, and

$$I(p : q) = \int_{\Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)} d\omega$$

when Ω is continuous.

$I(p:q)$ measures the divergence between the two distributions. If the two distributions are identical then $I(p:q) = 0$ since $\forall i : p_i = q_i$. However, we must note that cross-entropy measures only the difference between two given hypotheses, not necessarily the validity of one as compared to the other, since wrong hypotheses can be equally distant from a null hypothesis as correct ones. The best use of cross-entropic measure, then is not in picking out correct hy-

potheses, but in getting a difference measure between given hypotheses. It is easy to see that $I(p:q)$ is always non-negative as shown by the following lemma. This is a standard result in information theory and has been attributed to the physicist Gibbs by Hamming (1980, p.117) who called it the *fundamental Gibbs inequality*, but Goldie and Pinch (1991, p.15) suspects that the connection with the physicist Gibbs is *remote at best*.

Lemma 7.2.1 *For two discrete probability distributions p and q over a space ω of points $\omega \in \Omega$, $\sum_{\Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)} \geq 0$*

Proof

Consider $\sum_{\Omega} p(\omega) \log \frac{q(\omega)}{p(\omega)}$

Since $\forall x > 0 : \ln x \leq x - 1$

$$\begin{aligned} \sum_{\Omega} p(\omega) \log \frac{q(\omega)}{p(\omega)} &\leq \sum_{\Omega} p(\omega) \left(\frac{q(\omega)}{p(\omega)} - 1 \right) \\ &\leq \sum_{\Omega} q(\omega) - p(\omega) \\ &\leq 0 \quad \text{since } \sum_{\Omega} p(\omega) = \sum_{\Omega} q(\omega) = 1 \end{aligned}$$

$$\text{Thus } \sum_{\Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)} \geq 0$$

Furthermore, there is equality to zero only when $p = q$. When applied in the context of the sk-strings algorithm, if S_1 is the set of strings generated from state P and S_2 is the set of strings generated from state Q, and p and q are the distributions over $S = S_1 \cup S_2$, then the cross-entropy of states P and Q is given by

$$I(p : q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{7.1}$$

where p_i and q_i are the probabilities in p and q respectively of the i th string in S . There is an asymmetry in the cross-entropy above (Johnson 1979b; Kullback 1959) which is undesirable from the sk-strings perspective. If state Q produces a string that state P doesn't, then the corresponding p_i is zero, and the term $p_i \ln \frac{p_i}{q_i} = 0$ since $\lim_{x \rightarrow 0} x \ln x = 0$. However, if state P produces a string that state Q doesn't, then the corresponding q_i is zero and $I(p : q)$ becomes very

large. To make the measure symmetric, we consider the difference between the entropy of coding P's strings at Q and vice versa and coding P's and Q's strings at themselves, i.e, we consider a new information measure called the J divergence (Jeffreys 1946):

$$\begin{aligned}
 J(p : q) &= - \sum_i p_i \ln q_i - \sum_i q_i \ln p_i - \left(- \sum_i p_i \ln p_i - \sum_i q_i \ln q_i \right) \\
 &= \sum_i \left(p_i \ln \frac{p_i}{q_i} + q_i \ln \frac{q_i}{p_i} \right) \\
 &= \sum_i (p_i - q_i) \ln \frac{p_i}{q_i} \tag{7.2}
 \end{aligned}$$

The non-negativity of $J(p : q)$ follows directly from the non-negativity of $I(p : q)$ already proven above. Regardless, it is also easy to see that the sum is always positive because $p_i - q_i$ and $\log \frac{p_i}{q_i}$ are either both positive or both negative. The value of $J(p : q)$ is small and close to zero when p_i and q_i are close to each other and is large when either p_i or q_i is disproportionately large with respect to the other. Thus $J(p : q)$ is a symmetric cross-entropic measure that can be used as a potential metric in determining whether to merge two states or not.

While the use of the J divergence instead of the I divergence solves the problem of asymmetry, we are still left with the problem of unboundedness for the distance measure. Work has been done along the lines of finding lower bounds for $I(p : q)$ and hence $J(p : q)$ (Vajda 1970; Toussaint 1975; Kullback 1967), but no general upper bound exists for either (Vajda 1970). This is because p must be absolutely continuous with respect to q for $I(p : q)$ to be defined since when $p(\omega) \neq 0$, but $q(\omega) = 0$, $I(p : q)$ becomes undefined (Kullback 1959). Also, it is obvious that for $p(\omega) \neq 0$ and $q(\omega) \rightarrow 0$, $I(p : q)$ becomes very large. There are other distance measures that one may choose to use, notably the K-divergence (Lin 1991) $K(p:q)$ given by

$$K(p : q) = \sum_i p_i \log \frac{2p_i}{p_i + q_i} \tag{7.3}$$

with guaranteed lower and upper bounds. However the problem of unbounded-

ness does not arise for us because of specific details of implementation which is discussed in the next section.

7.3 Implementation

A program that uses the cross-entropic criterion for merging was developed for comparative testing with the intuitive heuristics outlined in Chapter 6. Now, it can be shown by the following lemmas that an upper bound exists on the value of $J(p : q)$ if a limit ε is imposed on the minimum value of p_i and q_i . This maximum value is elicited in theory when p_i and q_i have values at the opposite ends of their range, i.e, $p_i = \varepsilon$ and $q_i = 1$ or vice-versa.

Lemma 7.3.1 *If $\varepsilon \leq p_i \leq 1$, $\varepsilon \leq q_i \leq 1$, and $\sum_i p_i = \sum_i q_i = 1$, then the variational distance V between p and q has an upper bound at $2(1 - \varepsilon)$*

$$V = \sum_i |p_i - q_i| \leq 2(1 - \varepsilon)$$

Proof

$$\begin{aligned} V = \sum_i |p_i - q_i| &= \sum_{p_i > q_i} (p_i - q_i) + \sum_{q_i > p_i} (q_i - p_i) \\ &= \sum_{p_i > q_i} p_i + \sum_{q_i > p_i} q_i - \sum_{p_i < q_i} p_i - \sum_{q_i < p_i} q_i \\ &\leq \max \left(\sum_{p_i > q_i} p_i \right) + \max \left(\sum_{q_i > p_i} q_i \right) - \\ &\quad \min \left(\sum_{p_i < q_i} p_i \right) - \min \left(\sum_{q_i < p_i} q_i \right) \\ &\leq (1 + 1 - \varepsilon - \varepsilon) \\ &\leq 2(1 - \varepsilon) \end{aligned}$$

Lemma 7.3.2 *If $\varepsilon \leq p_i \leq 1$ and $\varepsilon \leq q_i \leq 1$ and $\sum_i p_i = \sum_i q_i = 1$, then*

$$J(p : q) = \sum_i (p_i - q_i) \ln \frac{p_i}{q_i} \leq 2(\varepsilon - 1) \ln \varepsilon$$

Proof

$$\begin{aligned} J(p : q) &= \sum_i (p_i - q_i) \ln \frac{p_i}{q_i} \\ &= \sum_i |p_i - q_i| \times \left| \ln \frac{p_i}{q_i} \right| \end{aligned}$$

Since for $\varepsilon \leq p_i \leq 1$ and $\varepsilon \leq q_i \leq 1$, $\ln \varepsilon \leq \ln \frac{p_i}{q_i} \leq -\ln \varepsilon$

$$\begin{aligned} J(p : q) &\leq -\ln \varepsilon \sum_i |p_i - q_i| \\ &\leq 2(\varepsilon - 1) \ln \varepsilon \quad (\text{from Lemma 7.3.1}) \end{aligned}$$

Thus, $J(p : q)$ is computed by accumulating the value of $(p_i - q_i) \ln \frac{p_i}{q_i}$ for each $S_i \in S_1 \cup S_2$. The resulting value is then normalised by dividing by $2(\varepsilon - 1) \ln \varepsilon$ to get a value between 0 and 1. 0 indicates total similarity between states P and Q and 1 indicates total dissimilarity. The user could now indicate a value between 0 and 1, which can be compared with the computed $J(p : q)$ to decide whether to merge states P and Q or not. A value of 1 for this threshold would merge any two states and a value of 0 would merge only states with identical distributions on output strings. By varying the value of this parameter, the user would have control over the degree of generalisation performed by this algorithm.

One more issue remains, however. How do we choose a value for ε ? Obviously, the choice influences the maximum value of $J(p : q)$ and thus the divisor for normalisation. While the choice of an arbitrarily low value for ε would be satisfactory for the majority of cases, it would preclude consideration of strings in S with probabilities less than the chosen value, which may be required by the user in some instances. Ideally, the value of ε should be chosen so that it is less than the least probability of any string in S, but how do we know this value beforehand?

It turns out that the procedural decision for tractability that was made in the implementation of the sk-strings algorithm solves the present problem.

The sk-strings algorithm was originally computationally intractable because the number of possible strings generated from a state is exponential in tail size. Thus, considering all possible strings from two given states would have placed enormous requirements on the memory and processing power of the machine, especially for large tail sizes. This problem was solved by putting an artificial limit on the minimum probability that strings generated from a given state can have. The limit defaulted to 0.001 (0.1%) and could be reset by the user if necessary. It was decided that this lower limit be used as the value for ϵ since that would be the least probability a string can have. Thus, by default, the symmetric cross-entropy $J(p : q)$ computed is in the range $0 \leq J(p : q) \leq 2 * (1 - 0.001) * \ln 0.001 = 13.80$ and the value was normalised by dividing by 13.8 to reduce it to the range 0–1.

For the purposes of comparison and verifying if this additional computational complexity had any marked advantages, another similar heuristic was also implemented along very simplistic lines. This operated similarly by giving a distance measure between the two sets of strings, but this second distance measure was simply the variational distance V between the two probability distributions p and q :

$$V = \sum_i |p_i - q_i|$$

It is easy to show $V(p : q)$ has an upper bound at 2 as a special case of lemma 7.3.1 where $\epsilon = 0$. In this implementation, however, where $\epsilon = 0.001$, value of $V(p : q)$ was normalised by dividing by its upper bound 1.998 to reduce it to the range 0–1.

7.4 Results

Programs implementing sk-strings with these two heuristics were run on the two training sets in Chapter 6. The results are too voluminous to be shown in tabular form and are hence presented only graphically in Figures 7.1 – 7.4. The plots also show the performance of one of the sk-strings heuristics, namely LAX, for purposes of comparison. As before, the horizontal axes represent values of

tail sizes from 1 to 10 and the agreement percentage from 1 to 100 while the vertical axes represent the MML of the inferred machines. In the legends and captions below, the heuristic using cross-entropy is called the XENTROPIC heuristic and the heuristic using the variational distance is called the VARDIST heuristic.

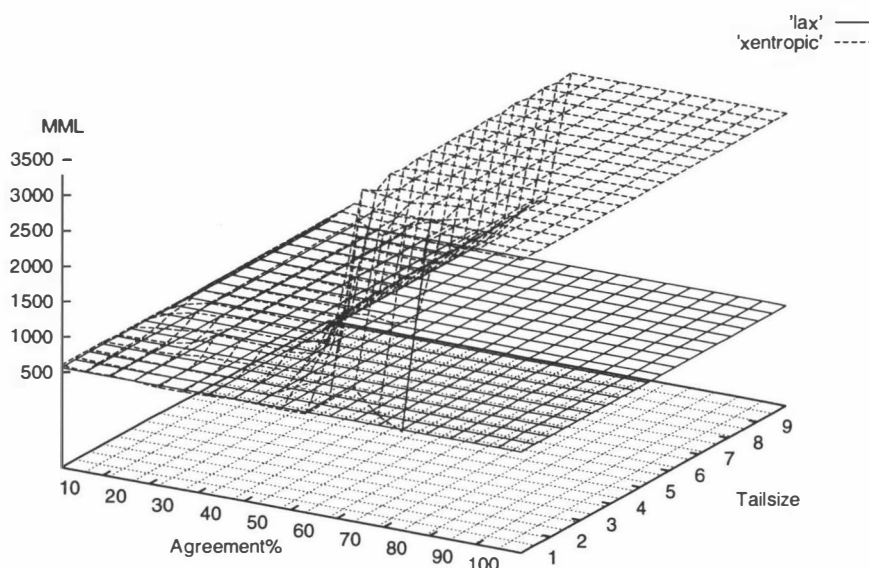


Figure 7.1: MMLs of 1000 inferred automata using the XENTROPIC heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 1).

Experiments were also run on the 100 randomly generated training cases of 10 strings each derived from small random automata (< 6 states, 10 arcs, 6 symbols) and the 100 training cases of 25 strings each, derived from larger automata (< 100 states, 500 arcs, 25 symbols). In both cases, the training sets were the same as the ones used in Chapter 6. The best machines inferred by each of the XENTROPIC and VARDIST heuristics for tail sizes 1–10 were compared for each training set against the corresponding best machine inferred by all of the four sk-strings heuristics combined. In the case of large automata only tail sizes of 1–5 were used.

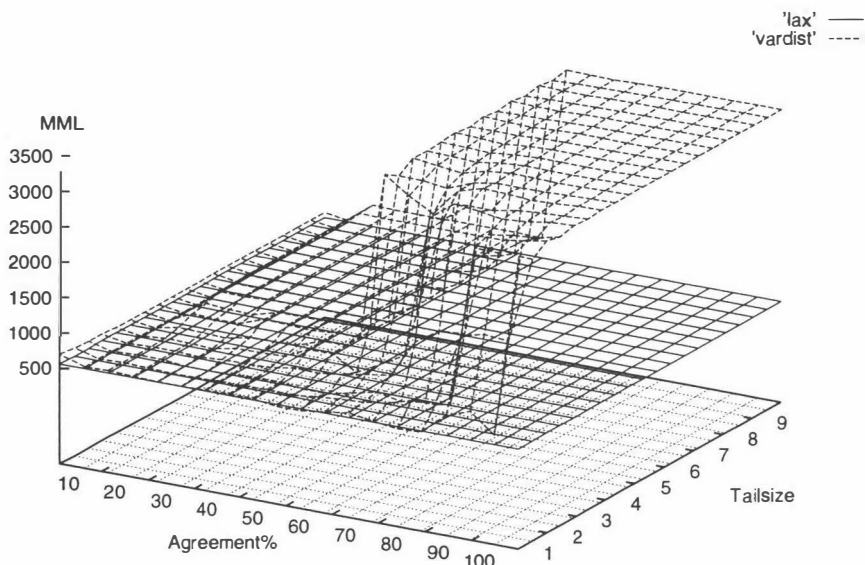


Figure 7.2: MMLs of 1000 inferred automata using the VARDIST heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 1).

Figure 7.5 shows the performance of the XENTROPIC heuristic contrasted with the performance of the 4 sk-strings heuristics in each case and the k-tails algorithm for training sets derived from small automata and Figure 7.6 shows the corresponding graph for large automata. Figures 7.7 and 7.8 show the corresponding graphs for the VARDIST heuristic. The horizontal axes represent the training sets and the vertical axes give the error in the inferred automaton computed using $1 - i/a$, where i is the MML of the inferred machine and a is the MML of the generating machine after being adjusted for exactly the emitted data. In each case, the inferred automaton chosen was the best one inferred of all tail sizes from 1 to 10 and values of the agreement percentage parameter (s) for sk-strings from 1 to 100. Negative errors indicate the inference of more compact PFSA than the generating ones which are consistent with the emitted data. This is due to the existence of some redundancy in the randomly generated structures.

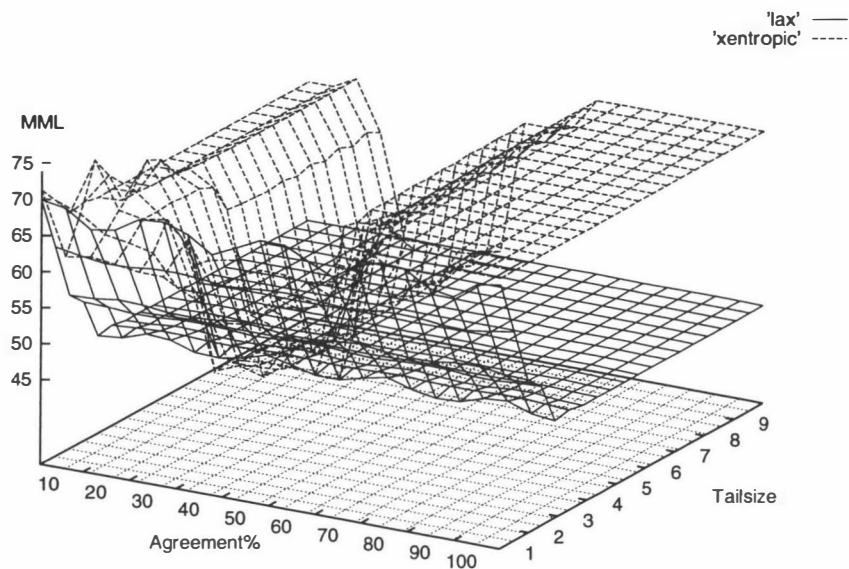


Figure 7.3: MMLs of 1000 inferred automata using the XENTROPIC heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 2).

Table 7.1 shows the success rates of each of the 6 sk-strings heuristics including XENTROPIC and VARDIST and the k-tails algorithm in inferring the best PFSA from the training sets generated by 100 small automata. Column 1 shows the number of times out of 100 the algorithm in column 2 was the one that inferred the best PFSA. When more than one algorithm is listed, they had both inferred the same automaton.

Table 7.2 shows the corresponding success rates for training sets generated by the 100 large automata.

7.5 Conclusion

The results show that for small automata the XENTROPIC heuristic performs as well as the OR heuristic, but no better. In the case of large automata, it clearly out-performs all other heuristics and the k-tails algorithm. Even more importantly, with the XENTROPIC heuristic, the collective ability of sk-strings

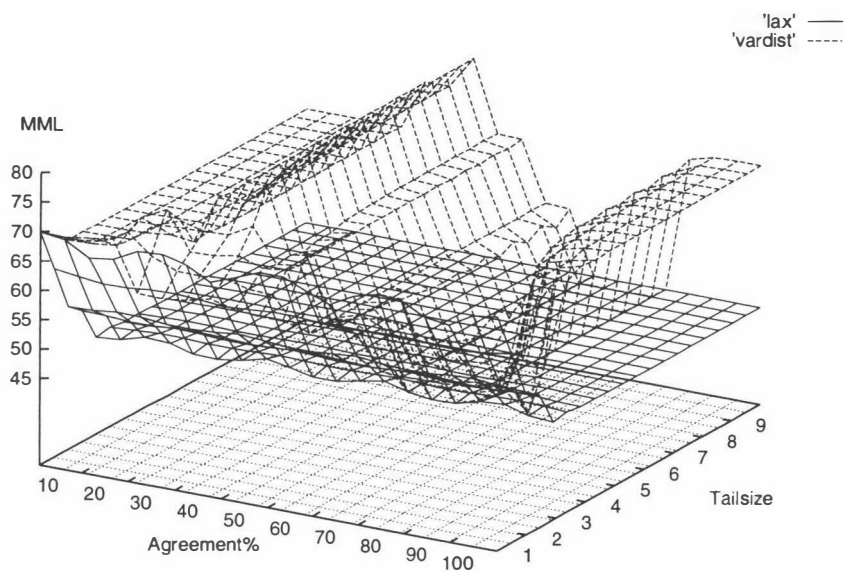


Figure 7.4: MMLs of 1000 inferred automata using the VARDIST heuristic for tail sizes 1–10 and agreement percentages 1–100 (Training set 2).

Success Rate	Algorithm(s)
38	OR,XENTROPIC
23	OR
14	XENTROPIC
7	KTAIL
6	KTAIL,OR
3	VARDIST
3	LAX
3	KTAIL,OR,XENTROPIC
1	KTAIL,OR,VARDIST
1	KTAIL,AND,LAX,VARDIST
1	AND

Table 7.1: Success rates of the k-tails and 6 sk-strings heuristics (including XENTROPIC and VARDIST) in inferring the best PFSA out of 100 small test cases

to infer the best automaton improves significantly. The VARDIST heuristic,

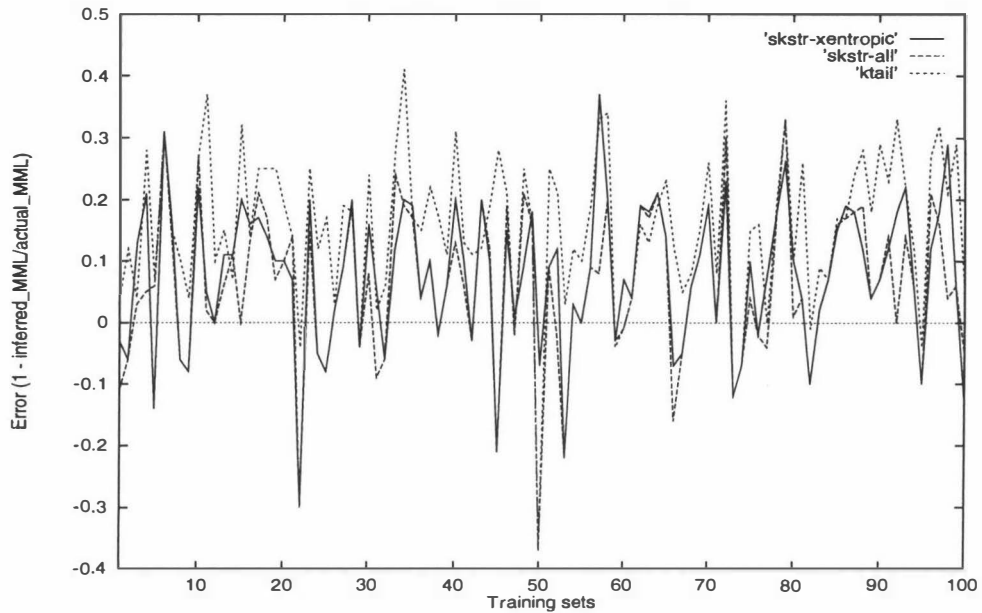


Figure 7.5: The sk-strings ($s=50\%$) XENTROPIC heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by small random automata.

Success Rate	Algorithm(s)
79	XENTROPIC
15	OR,XENTROPIC
6	KTAIL
0	Others

Table 7.2: Success rates of the k-tails and 6 sk-strings heuristics (including XENTROPIC and VARDIST) in inferring the best PFSA out of 100 large test cases

however, is not seen to perform as well. Nevertheless, since the XENTROPIC heuristic is still not able to consistently infer the best PFSA every time, we conclude that it is best to continue to choose the automaton with the least MML inferred by any of the 6 heuristics discussed so far and the k-tails algorithm for various values of their tunable parameters.

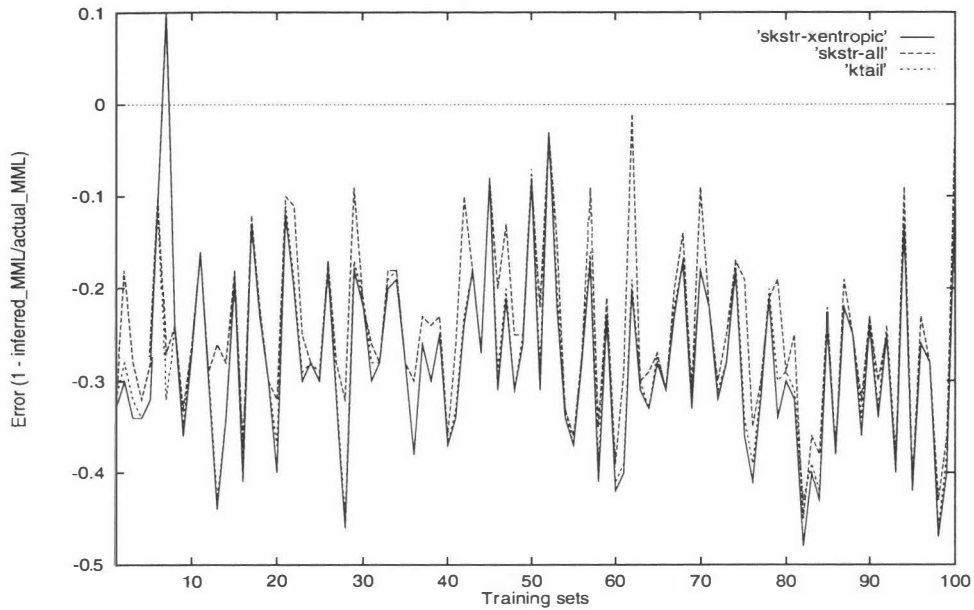


Figure 7.6: The sk-strings ($s=50\%$) XENTROPIC heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by large random automata.

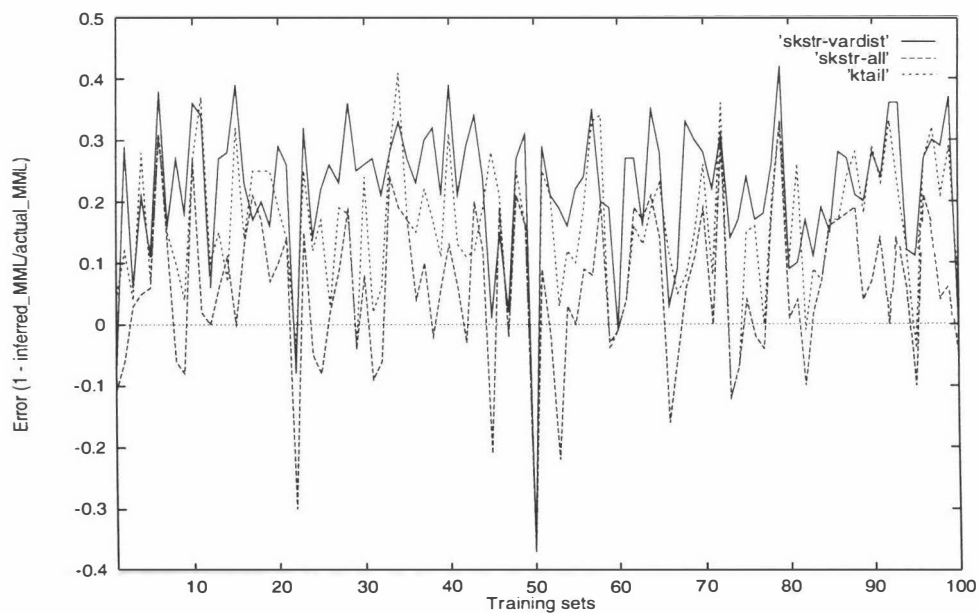


Figure 7.7: The sk-strings ($s=50\%$) VARDIST heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by small random automata.

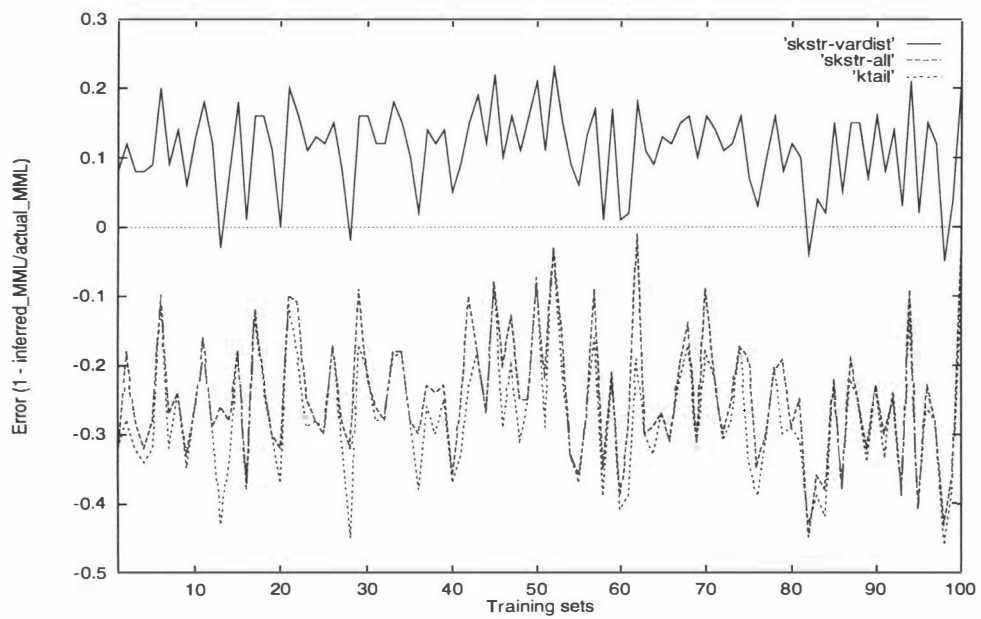


Figure 7.8: The sk-strings ($s=50\%$) VARDIST heuristic compared with the best of 4 sk-strings heuristics and the k-tails method using 100 random training sets generated by large random automata.

Chapter 8

Beam Search and Simulated Beam Annealing

8.1 Introduction

Chapters 6 and 7 have discussed intuitive and information theoretic heuristics for finding the best PFSA accepting a given set of positive examples. This chapter is concerned with solving the same problem using some search techniques from mainstream Artificial Intelligence. The two specific methods that are described here are the beam search method and a variation of simulated annealing.

8.2 Beam Search

The beam search algorithm is a variation of the best-first search strategy. The best-first search in turn, is itself a combination of the breadth-first and depth-first search techniques. In best-first search, a heuristic is used to select one of many possible descendants of the current state as the most promising one. This descendant state is expanded recursively, until either a solution is found or all the expansions of a current state look less promising than one of the top-level branches that were ignored previously. At this time, these expansions are added to a queue to be processed later and attention is returned to the next most

promising descendant of the top-level state. When all the descendants of the top-level state have been examined thus, and no solution still found, attention shifts to the expansions that were added to the queue before abandoning a particular path previously.

While the best-first search technique is more likely to find a solution than simple hill-climbing, it is very memory intensive and typically tends to take a long time to run for large search spaces. In such situations, the beam search technique may be more appropriate as it is more space efficient. Beam search works by not remembering all of the most promising descendants, but only the best m . This m is known as the beam size which is the parameter of its operation. The disadvantage in this is that the solution may be completely missed by pruning a search tree too early. But chances of this happening are low with a judicious choice of the beam size parameter and so it seems a good compromise for achieving tractability.

8.2.1 The Beam Search Algorithm for PFSA Inference

The beam search algorithm to infer the best set of merges to an input PFSA is presented in Algorithm 8.2.1. Basically, we start off with a given (possibly canonical) PFSA and compute the MMLs of all its immediate descendants (which we define to be every PFSA resulting from all possible two-state merges). There are ${}^N C_2$ or $(N^2 - N)/2$ such descendants for an N -state PFSA. The actual descendant PFSA don't have to be constructed, however, since it is reasonably easy to predict by how much the MML would change upon a given merge. The PFSA with the least m of these are chosen and these constitute our beam. The process is now repeated for each of the m PFSA in the beam giving $m \times {}^{N-1} C_2$ possible descendants. The best m of these and the current beam are chosen and the beam modified accordingly. The process repeated until there are no more changes to the beam. At this point, the PFSA with the least MML in the beam is put forward as our best guess of the original structure.

Algorithm 8.2.1 (Beam Search)

```
1 comment: Beam size = m
2 begin
3   Beam := Initialise a list of size m to hold m PFSA;
4   Add the input PFSA to Beam unmarked;
5   while There exist unmarked PFSA in Beam; do
6     Buffer := Initialise a list of size m to hold m PFSA;
7     for Each such unmarked PFSA in Beam; do
8       Mark it;
9       temp := best m of all node-pair merges in it, unmarked;
10    Buffer := best m PFSA in Buffer and temp;
11    od
12    Beam := best m PFSA in Buffer and Beam;
13  od
14  Return the best PFSA in Beam;
15 end
```

If the beam size is set to 1, the algorithm reduces to a simple hill-climbing search where only the best descendant at each node is explored further for a solution. An important thing to remember regarding beam search is that the solution space searched with a large beamsize will not necessarily be a superset of the space searched with a smaller beamsize. While larger beam sizes increase the space actually searched, an intuitive judgment as in the previous statement is incorrect. This is because one or more members of the beam can generate apparently better PFSA in the short term which fill up the beam by ousting some of the currently worse PFSA which might be the ones which actually lead to the best solution. However, the chances are good that the best solution will be found with larger beam sizes.

8.2.2 Algorithm Complexity

For a simple hill-climbing search starting from an N -state PFSA, at each stage in the choice process there are iC_2 alternatives to pick from, where i is the number of states of the currently held PFSA. In the worst case, we can expect the search process to continue right down to a single state machine. Thus the total number of intermediate PFSA to be examined will be: $\sum_{i=1}^N {}^iC_2$ which is given by $\sum_{i=1}^N (i^2 - i)/2$. This gives a time complexity of $O(n^3)$ for the algorithm

in the number of states of the PFSA. We only need to hold on to the one best PFSA at any time, so the space complexity is given by $O(n)$.

With beam search, we hold on to the best m PFSA at any time, thus the space complexity will be $O(mn)$. Regarding the number of intermediate PFSA that need to be examined, we note that each slot in the beam can only be replaced by a PFSA with a lower number of states since no intermediate PFSA is ever examined more than once (When a PFSA is marked, it is marked for ever). Thus in the worst case, each pass over the beam will replace its contents entirely with unmarked PFSA, but then each such PFSA will have at least one less state than the corresponding cell contents in the previous beam. Again in the worst case, each such PFSA will have exactly one fewer state than the previous one and thus the number of PFSA to be examined will be: $m \times \sum_{i=1}^N {}^i C_2$ giving a time complexity of $O(mn^3)$ in beam size (m) and the number of states of the original PFSA (n). This is an attractive option since not only is the algorithm just cubic, it is linear in beamsize as well.

8.3 Simulated Annealing

In spite of some of the advantages that beam search possesses over simple hill-climbing, it still retains one of its major shortcomings where better intermediate solutions are preferred over worse ones. This is a problem since *the best* solution can sometimes only be obtained by following an apparently unpromising path. What we need then is some mechanism by which we could also sometimes reject better intermediate solutions in favour of worse ones early in the search process.

Simulated annealing (Kirkpatrick et al. 1983) is one such technique with a well defined procedure for doing this. It is based on the physical process of annealing in which metals are heated to high temperatures and gradually cooled. During the process of annealing, electrons in the metal will emit photons and gradually descend from higher energy states to lower ones, but there is always some chance that an emitted photon may bump into another electron and knock it into a higher energy state. This chance, however, decreases with the temperature of the substance, i.e. there is a lower probability of a transi-

tion to a high energy state at a low temperature than at a higher one. This probability is, in fact, given by the function $p = e^{-\Delta E/kT}$ where ΔE is the positive change in the energy level of the electron, T is the temperature in kelvin and k is Boltzmann's constant. Two things are obvious from this expression — (1) At a given temperature, the probability of a large uphill move (transition to a relatively high energy level) is low as compared to the probability of a small uphill move and (2) The probability of an uphill move being made at all decreases with temperature T .

The rate of cooling is called the *annealing schedule*. This has significance for the end product because rapid cooling causes the formation of regions of stable high energy and very slow cooling causes wastage of time in the asymptotic region of the cooling curve. The optimum choice of the annealing schedule is often determined empirically.

In the simulated annealing procedure, Boltzmann's constant is irrelevant since it is specific to the physical process. Thus a modified probability formula

$$p' = e^{-\Delta E/T} \quad (8.1)$$

is used where the k can now be assumed incorporated into the artificial parameter T . ΔE also now stands not for the change in energy level, but for the change in the value of an objective function which determines the poorness of a current solution as compared with its predecessor. We use the MML gain ratio for our objective function — $(\text{new MML} - \text{old MML})/(\text{old MML})$. This is guaranteed to be a positive quantity because stochastic acceptance is only tried when the new MML is greater than the old MML (new PFSA is worse than the old one). Algorithm 8.3.1 gives the procedure for attempting to find the best solution using simulated annealing:

Algorithm 8.3.1 (Simulated Annealing)

```

1 begin
2   Best_so_far := Current_PFSA;
3   Temperature = Initial Temperature;
4   while No more changes to Current_PFSA do
```

```

5      for Node-pair merges in the Current PFSA do
6          Perform merge;
7          if Resultant PFSA is better than current one
8              then
9                   $Current\_PFSA :=$  Resultant PFSA;
10                  $Best\_so\_far :=$  Resultant PFSA;
11             else
12                 Compute  $p'$  using Formula (8.1);
13                  $x :=$  Random number in the range [0,1];
14                 if  $x > p'$ 
15                     then
16                         Reject this PFSA;
17                     else
18                          $Current\_PFSA :=$  This PFSA;
19                 fi
20             fi
21         od
22          $Temperature :=$  Revised value (possibly  $Temperature-1$ );
23     od
24     Return best PFSA in {Current_PFSA, Best_so_far};
25 end

```

8.3.1 Simulated Annealing with a Beam

As Algorithm 8.3.1 shows, there is usually one current guess about the problem solution in the standard version of simulated annealing. It seems reasonable to believe that the performance of this algorithm can be markedly improved by incorporating a beam into the search process resulting in what we can call a *simulated beam annealing* strategy. This, in fact, more accurately models the physical annealing process as there are now several potential candidates in the beam akin to several electrons in a piece of metal being annealed. In this dissertation, the simulated beam annealing search procedure is called *simba* search for brevity.

The procedure borrows much from the standard beam search technique. At each pass through the beam, we generate the possible descendants of the beam occupants. If one of the solutions is better than any of the ones already in the beam, it is added to the beam as before. However, if it is worse than any currently in the beam, we compute the objective function and accept it stochastically. The process of accepting such a *worse* candidate will result in

the currently best solution in the beam being shifted out, as the beam is always maintained in strictly sorted order of decreasing goodness of its occupants. This should not cause too much concern as we still hold on to the best intermediate solution as in the standard simulated annealing procedure. On the average, at the end of each pass, a certain proportion of the beam occupants would have migrated upwards (towards worse intermediate solutions) and the remaining either horizontally or downwards (towards better intermediate solutions). The proportion of upward migration decreases with temperature, until at a temperature of 0° all migrations are towards better intermediate solutions. Thus, simba search starts off seemingly stochastically, but gradually settles down into the standard beam search process.

Algorithm 8.3.2 gives the procedure for attempting to find the best solution using simba search.

Algorithm 8.3.2 (Simba search)

```

1 comment: Beam size = m
2 begin
3   Beam := Initialise a list of size m to hold m PFSA;
4   Best_so_far := Current_PFSA;
5   Add the Current PFSA to Beam unmarked;
6   while There exist unmarked PFSA in Beam; do
7     Buffer := Initialise a list of size m to hold m PFSA;
8     for Each such unmarked PFSA in Beam; do
9       Mark it;
10      for Each of the node-pair merges of the PFSA; do
11        if Resultant PFSA is better than any in Buffer;
12          then
13            Add it to Buffer scrolling out the worst one in it;
14            Best_so_far := Resultant PFSA;
15          else
16            Compute  $p'$  using Formula (8.1);
17             $x$  := Random number in the range [0,1];
18            if  $x > p'$ 
19              then
20                Reject this PFSA;
21            else
22              Add it to Buffer scrolling out the best one in it;
23          fi
24      fi

```

```

25         od
26     od
27         Merge PFSA in Buffer into Beam as in steps 11–23;
28         Temperature := Revised value (possibly Temperature-1);
29     od
30     Return best PFSA in {Best PFSA in Beam, Best_so_far};
31 end

```

The annealing schedule used in the implementation reported here consists of a fixed countdown of T from the initial temperature to 0, by one degree per pass. The initial temperature is fixed by default to be half the number of states in the input PFSA. This guarantees that stochastic moves uphill will be made at least during the upper half of the search process. If T is initialised to 0 for a particular run, however, the simulated annealing process becomes a beam search process. Obviously, with $T=0$ and beam size=1, the algorithm reduces to the simple hill-climbing search. It is easy to see that no PFSA being added into a revised beam can have more states than any that are already in it. Thus the time and space complexities of the simba search algorithm are the same as that for the beam search algorithm. Only the process of selecting PFSA for revising the beam is different.

On the Gaines (1976) data, the beam search technique converges to the best solution (4 state machine with MML = 46.78 bits) in fifteen passes over a beam of size 10. However, the simba search technique is not seen to do as well. It converges prematurely in 11 passes over the beam to an incorrect solution (11 state machine with MML = 89.72 bits). This is probably due to the rejection of potentially good paths early in the search process. With the Andrae data of Section 5.5, neither beam search nor simba search seems to be able to infer the correct 4 state machine from the 101-state input canonical automaton in reasonable time and space. However, given the 5 state machine as the initial guess, both converge rapidly to the correct four state machine.

8.4 Results and Discussion

Programs implementing beam search and simba search were run on the two training sets in Chapters 6 and 7. Figures 8.1 and 8.2 show their performance against the six sk-strings heuristics discussed earlier on each of the 2 fixed training sets from Chapter 6. The horizontal axis shows the tail size for the sk-strings algorithms ($s=50\%$) and beam size for the simba search and beam search algorithms and the vertical axis shows the MML of the machine inferred with that tail size.

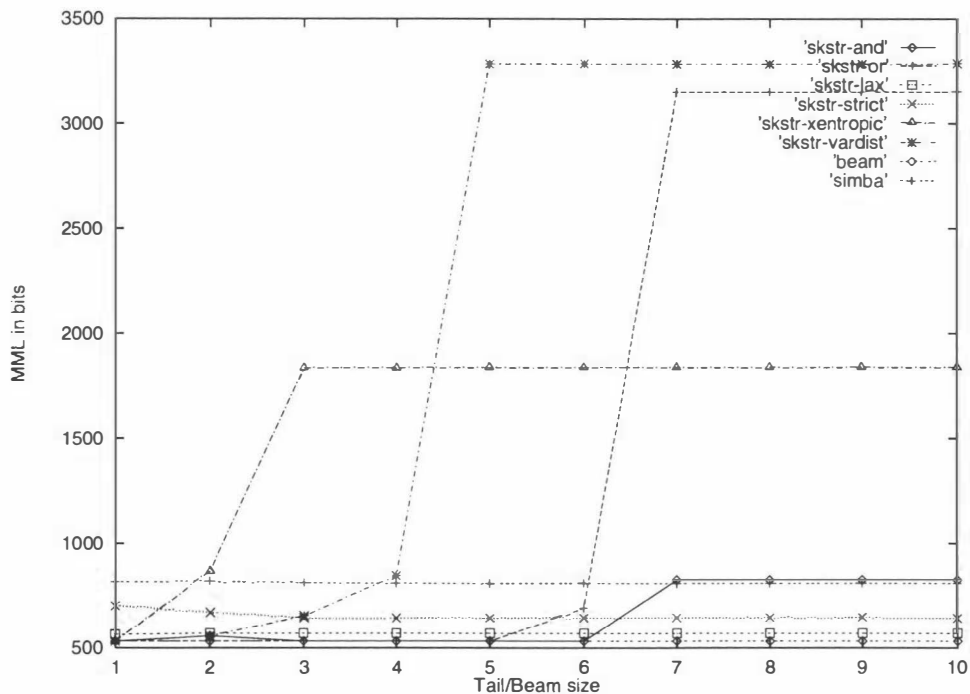


Figure 8.1: Performance of the beam search and simba search procedures on Training set 1 from Chapter 6. The plot shows the MMLs of 1000 machines induced against various beam sizes or tail sizes (for the 6 sk-strings heuristics).

Experiments were also run on the 100 randomly generated training cases of 10 strings each derived from small random automata (< 6 states, 10 arcs, 6 symbols) and the 100 training cases of 25 strings each, derived from larger automata (< 100 states, 500 arcs, 25 symbols). In both cases, the training sets were the same as the ones used in Chapters 6 and 7. The best machine inferred

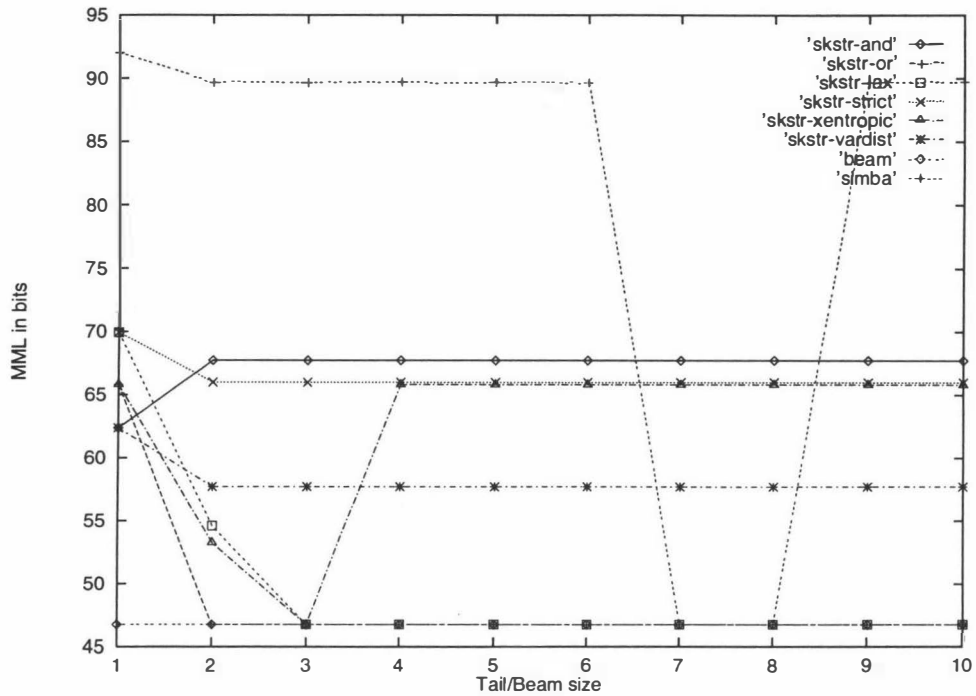


Figure 8.2: Performance of the beam search and simba search procedures on Training set 2 from Chapter 6. The plot shows the MMLs of 1000 machines induced against various beam sizes or tail sizes (for the 6 sk-strings heuristics).

by any of the 6 sk-strings heuristics (AND, OR, LAX, STRICT, XENTROPIC, VARDIST) for tail sizes 1–10 was the one compared for each training set against the corresponding best machine inferred by the beam search methods for beam sizes 1–10. In the case of large automata, only tail and beam sizes of 1–5 were used.

Table 8.1 lists the success rates of various beam sizes in inferring the best automaton using the beam and simba search procedures for small random PFSA. It shows that in over 70% of the test cases, a beam size of 1 was the best choice for simba search, indicating that there is much to be said for the standard simulated annealing procedure in comparison to simba search. Beam search, on the other hand, does not seem to show any particular preference for small beam sizes. The fact that small beam sizes have performed better than large beam sizes in at least some cases is empirical proof that large beam sizes do not search a super set of the space searched with small ones.

Beam size	Beam Search	Simba Search
1	13	70
2	14	5
3	5	6
4	7	2
5	10	1
6	6	3
7	5	1
8	10	5
9	19	4
10	11	3

Table 8.1: The number times (out of 100 training sets) that each beam size (1-10) succeeded in inferring the minimum automaton using the beam search and simba search methods.

Figures 8.3 and 8.4 show the performance of beam search and simba search against the six sk-strings algorithms on the 100 training sets from small random PFSA used in Chapter 6. As before, the horizontal axis represents the training sets and the vertical axis gives the error in the inferred automaton computed using $1 - i/a$, where i is the MML of the machine inferred by one of the algorithms and a is the MML of the machine that actually generated the training sets after being adjusted for exactly the emitted data. The graphs “skstr-all”, represents the combined performance of all the six sk-strings heuristics discussed earlier and the graphs “beams” and “simba” represent the performances of the beam search and simba search methods.

Figures 8.5 and 8.6 show the performance of beam search and simba search against the six sk-strings algorithms on the 100 training sets from the larger random PFSA used in the earlier chapters.

Figure 8.7 shows the performance of the algorithms where the output of the six sk-strings heuristics and the k-tails method on the 100 random training sets from small PFSA are used to seed the beam search method operating with a beam size of 10. The various line graphs trace the error in the MML of the best machines inferred by any of the 6 sk-strings methods with and without post-

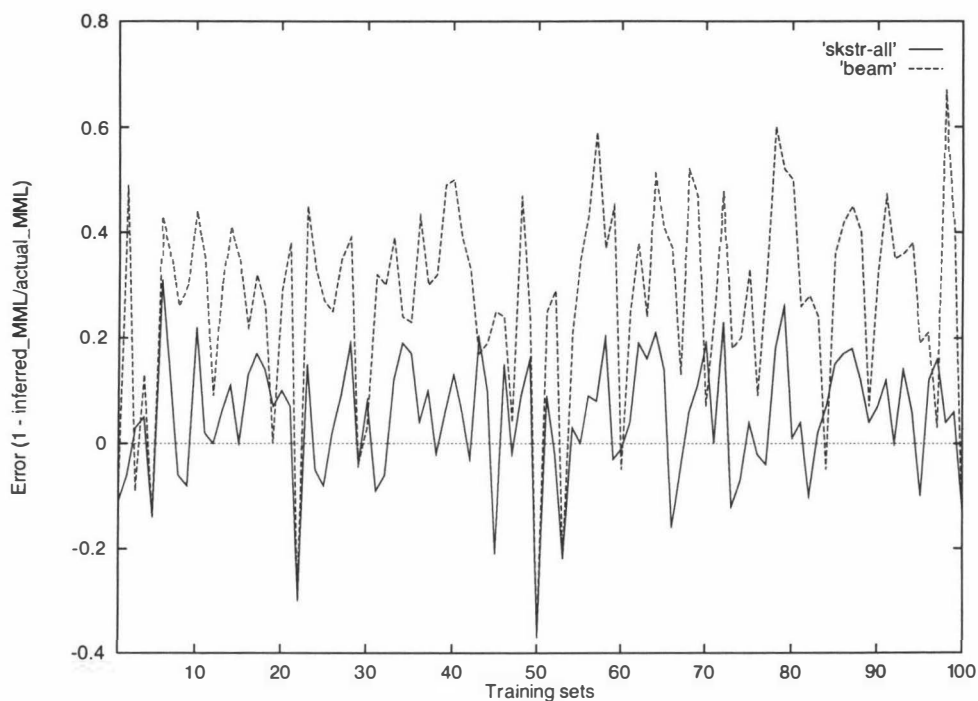


Figure 8.3: Beam search compared with sk-strings ($s=50\%$) for 100 training sets generated from small random automata.

processing by beam search (skstr-all and skstr-all+beam), the best machines inferred by the k-tails method with and without post-processing by the beam search method (ktail and ktail+beam), and also plain beam search method used directly on the canonical machine. It is seen that the finely dotted line representing machines inferred by the beam search seeded by sk-strings performs best of all.

8.5 Discussion and Conclusion

From the results in the previous section, it appears that neither beam search nor simba search seem to perform as well as the earlier more intuitive sk-strings heuristics in inferring PFSA. One reason for this could be that the methods described in this chapter are only effective when the solution spaces to be searched are small, i.e. the input machine has a small number of states. Even though the first experiment was an attempt to try and infer small automata, the canonical

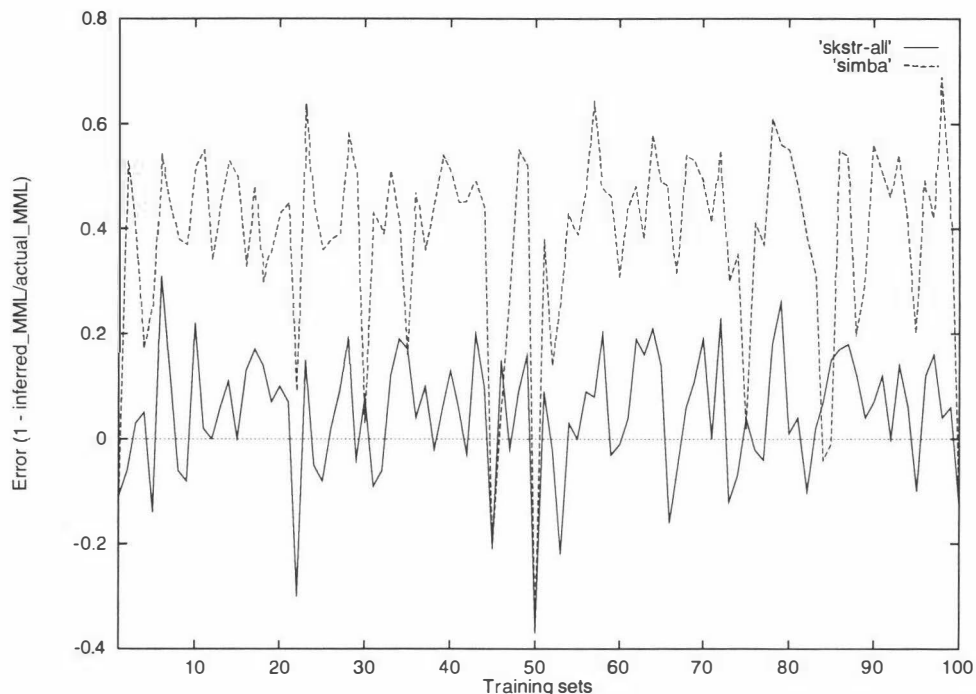


Figure 8.4: Simba search compared with sk-strings ($s=50\%$) for 100 training sets generated from small random automata.

machines constructed from the outputs of the small automata tended to be relatively large and thus the performance of these methods suffered.

Another reason for the poor performances of these techniques could be due to their effectiveness being determined only by objective functions causing them to explore the solution space blindly. This suggests that some important characteristic of the solution space which is captured in the intuitive process is being ignored by the information theoretic objective functions. While further analysis of the various algorithms and changes to them are possible, what is deemed best at this stage is a judicious choice of one or more of these algorithms operating together to search the solution space in more ways than one to give the best possible answer.

It also appears from the results that the nature of the solution space is such that beam search performs significantly better than simba search. This indicates the possibility that the path to the best solution might lie along suc-

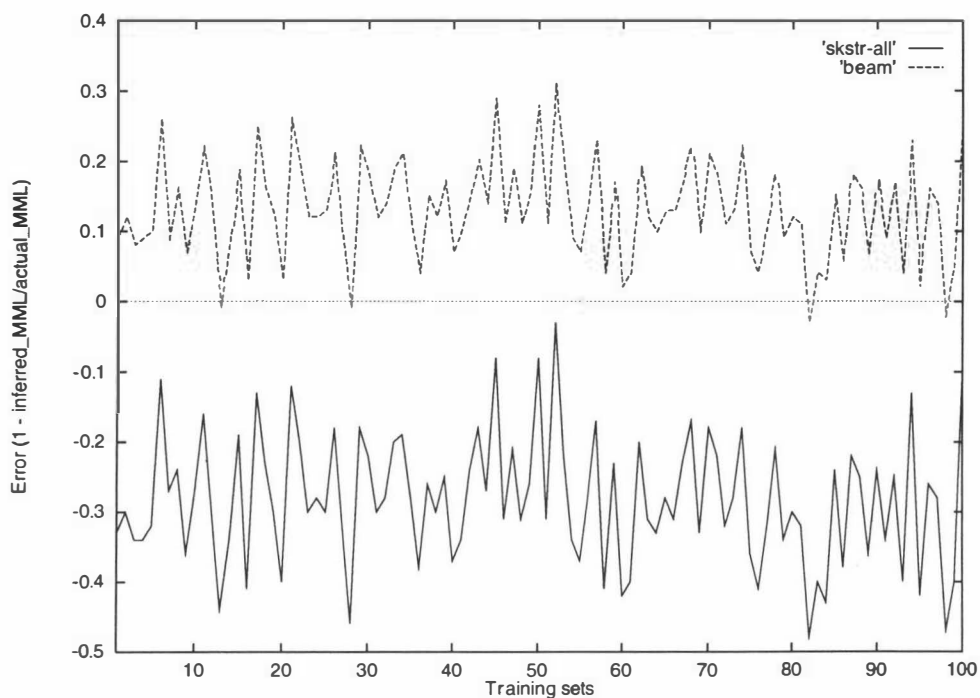


Figure 8.5: Beam search compared with sk-strings ($s=50\%$) for 100 training sets generated from larger random automata.

cessively better intermediate solutions. However, this is obviously not always true since simple hill climbing does not consistently give us the best solution. What looks reasonably certain though, is that rejection of better intermediate solutions in favour of worse ones is not likely to give the best final solution. It seems that of several equivalent PFSA, some had to get shifted out of the beam due to overcrowding and these were the ones that would have led to the best solution eventually. It might also be that better intermediate solutions should perhaps be accepted early and late in the simba search process, but rejected somewhere in the middle. This could be achieved with a bell shaped annealing schedule. As to the exact nature of the solution space and the fine tuning of the simba search annealing schedule, further research is necessary and this is left as a topic for future work.

The results have implications for hill climbing searches such as the one in Oliver (1993) which blindly attempt to find the decision graph with the smallest

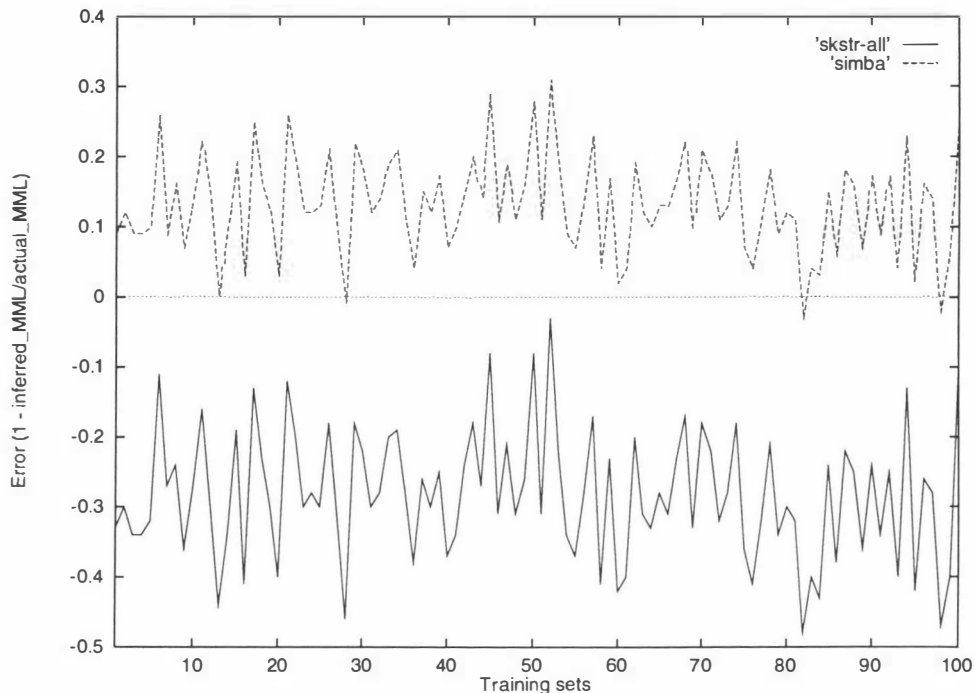


Figure 8.6: Simba search compared with sk-strings ($s=50\%$) for 100 training sets generated from larger random automata.

MML by following the path through the best immediate modifications to the current decision graph until no more modifications can be made. As mentioned earlier, the performance of such information theoretic searches can be improved significantly by incorporating some intuitive element into them or by decreasing the total space that needs to be searched since the experimental results show that a beam search tends to perform well on small input automata. This means that we can hope for good results by applying the beam search method to automata that have already been reduced by one of the other algorithms, for example sk-strings, which tend to perform better on large automata. That this is the best line of approach is evidenced by the results in Figure 8.7.

Thus, the procedure finally adopted in this dissertation is to initially apply the k-tails method and each of the six sk-strings heuristics to infer PFSA. The best PFSA inferred by them is selected as the seed for a beam search with a beam as large as computational resources permit. The output of the beam

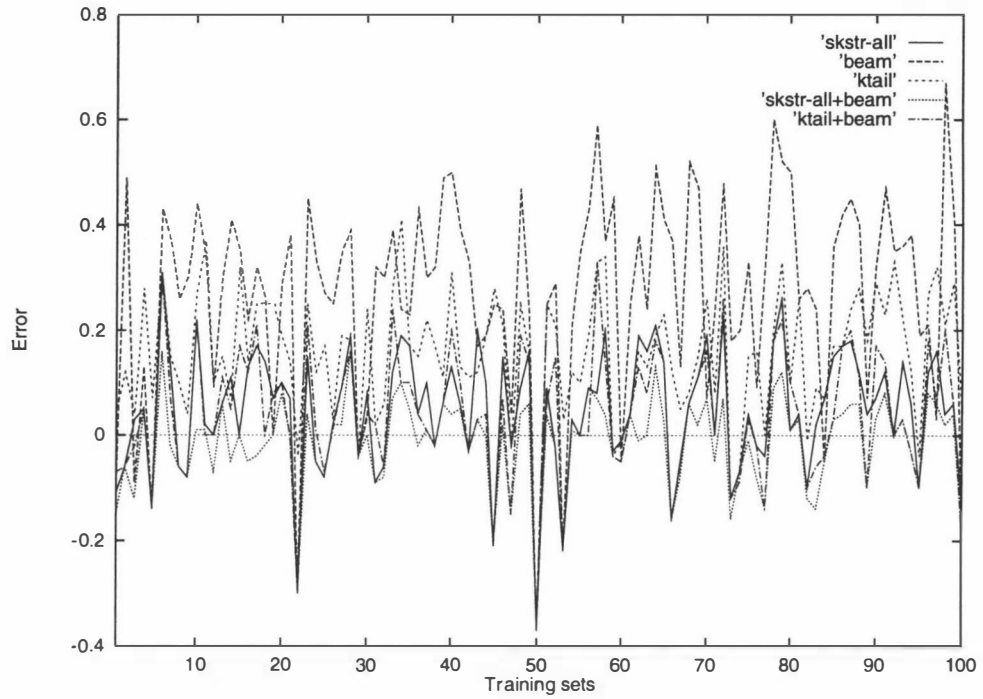


Figure 8.7: Seeded beam search compared with sk-strings ($s=50\%$) and unseeded beam search for 100 training sets generated from small random automata (Best of tail sizes 1 to 10 and beam sizes 1 to 10).

search, which is guaranteed to be at least as good as its input is then proposed as our best guess of the original structure.

Chapter 9

Results and Conclusion

9.1 Program outputs

Canonical machines were generated using the data for each of the Modern Cantonese and Modern Beijing dialects firstly using the diachronic rule sequences only, and then by including allophonic detail in them. The MMLs of the canonical machines are given in Table 9.1. Column 1 represents the MML of the PFSA derived for Modern Cantonese and column 2 represents the MML of PFSA for Modern Beijing. As mentioned in Section 3.4, smaller values of MML reflect a greater regularity in the structure. For comparison, Table 9.2 gives the MML for the canonical machines computed using Wallace’s original Formula (5.4) for deterministic PFSA.

	Cantonese	Beijing
Diachronic only	35243.58 bits (1168 states, 1167 arcs)	36790.93 bits (1212 states, 1211 arcs)
Diachronic + Allophonic	37782.43 bits (1321 states, 1320 arcs)	39535.43 bits (1468 states, 1467 arcs)

Table 9.1: MMLs for the canonical PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively.

Each of the four canonical machines was then reduced in ten different ways using the k -tails algorithm using values of k from 1 to 10. They were also reduced in 6000 more ways using each of the six sk -strings algorithms for each

	Cantonese	Beijing
Diachronic only	48942.11 bits	52327.07 bits
Diachronic + Allophonic	53142.11 bits	56486.48 bits

Table 9.2: MMLs for the canonical PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively using Formula (5.4).

of the 1000 combinations of s from 1 to 100 (or in the case of the XENTROPIC and VARDIST heuristics, from 0.01 to 1.00) and k from 1 to 10 (A total of 24000 machines were produced for the four datasets). The MMLs of the best machines inferred for each of the four cases are presented in Table 9.3, along with the specific heuristics and parameters that produced them. It is seen that the STRICT heuristic of sk-strings had the most success in inferring the machine with the minimum MML (three of the four minimum PFSA inferred by the STRICT heuristic and the remaining one by XENTROPIC).

	Cantonese	Beijing
Diachronic only	32439.32 bits (344 states, 882 arcs) Heuristic = STRICT, Tailsize = 4, Agr% = 1	33386.17 bits (365 states, 896 arcs) Heuristic = XENTROPIC Tailsize = 6, Err = 0.01
Diachronic + Allophonic	34345.71 bits (381 states, 969 arcs) Heuristic = STRICT, Tailsize = 4, Agr% = 1	34813.64 bits (373 states, 986 arcs) Heuristic = STRICT Tailsize = 6, Agr% = 1

Table 9.3: MMLs for the reduced PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively.

The reduced PFSA were then used as the starting point for a further reduction using the beam search algorithm with a beam size of 200.¹ The MMLs of the PFSA thus inferred are given in Table 9.4. The PFSA themselves are presented in tabular form in Appendix B as a list of Source, Target, Symbol,

¹It was also found that a beam size of 1 and other intermediate values between 1 and 200 inferred the same PFSA.

Frequency tuples.

	Cantonese	Beijing
Diachronic only	30379.01 bits (174 states, 640 arcs)	30366.55 bits (142 states, 595 arcs)
Diachronic + Allophonic	32711.49 bits (195 states, 707 arcs)	31585.79 bits (153 states, 634 arcs)

Table 9.4: MMLs for the PFSA for Middle Chinese to Modern Cantonese and Modern Beijing respectively further reduced with beam search.

The canonical PFSA are too large and complex to be printed on A4 paper using viewable type, but the reduced PFSA can be presented graphically if some of their low frequency arcs are trimmed off. Thus the reduced PFSA for Modern Beijing and Modern Cantonese are presented in Figures 9.1 and 9.2, but arcs with a frequency less than 10 have been pruned. Since several arcs have been pruned, the PFSA may not make complete sense as some nodes may have outgoing transitions without incoming ones and vice-versa. There is furthermore a small amount of overprinting. The figures are given solely for the purpose of a rough visualisation of the end-results and any detailed analysis should be based on the complete PFSA in Appendix B. Some interesting facts, however, seem to be borne out by them.

9.2 Discussion

The results obtained from the MMLs of canonical machines show that there is a greater complexity in the diachronic phonology of Modern Beijing than there is in Modern Cantonese. These complexity measures may be construed as measures of distances between the languages and their ancestor. Nevertheless we exercise caution in interpreting the results as such. The measures were obtained using just one of many reconstructions of Middle Chinese and one of many proposed diachronic phonologies. It is, of course, hypothetically possible that a simplistic reconstruction and an overly generalised phonology could give smaller complexity measures by resulting in less complex PFSA. One might

argue that this wrongly indicates that the method of obtaining distances as described here points to the simplistic reconstruction as the better one. This problem arises partly because of the fact that the methodology outlined here assumes all linguistic hypotheses to be equally likely *a-priori*. We note, however, that simplicity and descriptive economy are not the only grounds for preferring one linguistic hypothesis to another (Bynon 1983, p.47). Many other factors are usually taken into consideration to ensure whether a reconstruction is linguistically viable. As mentioned earlier, plausibility and elegance (Harms 1990, p.314), knowledge of what kinds of linguistic changes are likely and what are unlikely (Crowley 1987, p.90), and in the case of Chinese, insights of the “Chinese philological tradition” (Newman 1987) are all used when deciding the viability of a linguistic reconstruction. Even though Chen himself chooses one alternative derivation over another because, “if nothing else,” the former is “shorter than the latter” (Chen76, p.184), it is to be noted that such a choice was also implicitly guided by the other linguistic factors mentioned. Thus, a final conclusion about the linguistic problem of subgrouping is still properly within the domain of historical linguists. This dissertation just provides a valuable tool to help quantify one of the important parameters that is used in their decision procedure.

Besides just obtaining distance measures, it is also possible to draw other interesting conclusions from the results of the experiments conducted. In the discussion that follows, we allow ourselves the liberty of mild speculation to envisage the prospects and scope for future work in this project.

9.2.1 Status of exceptions

We note that the reduced PFSA are mostly consistent with the RCs given by Chen76 and CN84 showing that the reduction process has not introduced any inconsistency which would be precluded by the original linguistic hypotheses. But although they do not contradict the phonologies given in Chen76 and CN84 (reproduced in Figures 4.2 and 4.3) in the main, the rules enclosed in square brackets in Figures 9.1 and 9.2 and Tables 9.5 and 9.6 do constitute exceptions

to them.²

Table 9.5 and Table 9.6 list the 20 most frequent exceptions encountered in Modern Beijing and Modern Cantonese respectively and their frequencies in the whole data corpus. A discussion of the names of the various exceptions is given in Appendix C. It is, however, worth restating here that the interesting exceptions are those that are exceptional applications of the regular phonological rules. Such exceptions indicate that the regular rule had applied, but in violation of either its conditioning environment or the RC. In general, no distinction has been drawn between these two different kinds of exceptions because they are interchangeable from a computational point of view for reasons discussed earlier — there is an equal penalty to pay for describing either kind of violation (see Section 4.9.3). However, for purposes of linguistic interest, the tables list where applicable the number of times an exception happened because of an RC violation. In the case of Modern Beijing, for example, all 83 occurrences of the [chamel] exception were violations of the RC, whereas only 11 of the 17 [round] exceptions were. The other 6 were violations of the rule's conditioning environment. The rule [asipr-r], the aspiration redundancy condition, causes an aspirated fricative to lose its aspirated status (Chen76, p.156). It is not part of the RC although it is discussed in Chen76 and hence the RC violation column is not applicable to it. For comparison, Table 9.7 list the 20 most frequent regular phonological rules in Modern Beijing and Modern Cantonese.

From the Tables 9.5 and 9.6, it is seen that the most frequent exception by far in both the Modern Beijing and Modern Cantonese linguistic hypotheses as presented in Chen76 and CN84 is the unexpected tone split rule, denoted in the PFSA by the symbol “[t-split]”, in the phonology of Modern Cantonese. This occurs when a word with a voiced initial in SMC split to Tone A (and thence possibly to Tone C via the AC-SPLIT rule) or when a word with an unvoiced initial in SMC split to Tone B in Modern Cantonese. Despite being the most frequent exception, this unexpected tone split rule only occurs 125 times (only

²Few such exceptional rules are evident in the Figures 9.1 and 9.2 because most of the exceptional rules are low in frequency and are thus unrepresented.

Exception	Description	Frequency	RC Violations
[chamel]	P-rule fed by Chameleon vowel PC	83	83
[!t2]	t2-merge non-application	18	n/a
[round]	Exceptional vowel rounding	17	11
[i-del]	Exceptional i-deletion	17	7
[aspir]	Exceptional aspiration	16	7
[aspir-r]	Aspiration redundancy condition	19	n/a
[t1-split]	Exceptional split of tone 1	12	4
[u-del]	Exceptional u-deletion	11	n/a
[k-u]	k changes to u	11	n/a
[u-o]	u changes to o	9	n/a
[t3-t2]	Tone 3 changes to Tone 2	8	n/a
[ŋ-n]	Velar nasal dentalisation	8	n/a
[t4-tripart]	Exceptional t4-tripartition	6	5
[t1-t2]	Tone 1 changes to Tone 2	6	n/a
[procope-j]	Initial j deletes	6	n/a
[i-y]	i changes to y	6	n/a
[fric]	Exceptional frication	6	1
[t3-t1a]	Tone 3 changes to Tone 1a	5	n/a
[affric]	Exceptional affrication	5	4
[!affric]	Exceptional non-affrication	5	n/a

Table 9.5: 20 most frequent exceptions in the phonology of Modern Beijing.

105 shown in the diagram), i.e. in less than 3.9 percent of the total number of words in the database. The relative frequency of this exception is to some extent already obvious from Appendix A of CN84 (III, p.153) which lists the Modern Cantonese Tone correspondences. But an important aspect in which the present work *goes beyond* CN84 is by giving the relative frequency of the tone split exception among all other exceptions; the Appendix in CN84 only gives the relative frequency of the tone split exception among other tonal rules and exceptions.

Another important aspect in which the present study extends the previous work is in bringing to light some important exceptions that were not represented in Chen76 or CN84. For example, the second-most frequent exception encountered was the unexpected feeding relationship of the chameleon vowel rule to other diachronic rules in Modern Beijing. This occurred 83 times, approximately 3 percent of the total sample. However, this is not obvious in the Chen76 RC, except for an actual instance of this feeding relationship in

Exception	Description	Frequency	RC Violations
[t-split]	Exceptional tone-split	125	0
[aspir]	Exceptional aspiration	33	12
[!ac-split]	AC-split non-application	30	n/a
[procopie-j]	Initial j deletion	21	n/a
[s-ts]	s changes to ts	15	n/a
[t2-t3]	Tone 2 changes to Tone 3	14	n/a
[t3-t2]	Tone 3 changes to Tone 2	13	n/a
[de-aspir]	Initial de-aspiration	13	n/a
[ts-s]	ts changes to s	11	n/a
[procopie-x]	x deletion	11	n/a
[a-front]	Back a is fronted	10	n/a
[t2-t1]	Tone 2 changes to Tone 1	9	n/a
[t1-t2]	Tone 1 changes to Tone 2	9	n/a
[round]	Exceptional vowel rounding	6	6
[x-k ^h]	x changes to k ^h	5	n/a
[t3-t1]	Tone 3 changes to Tone 1	5	n/a
[t1-t3]	Tone 1 changes to Tone 3	5	n/a
[i-del]	Exceptional i-deletion	5	n/a
[unglide]	Glide removal	4	4
[j-ins]	j insertion	3	n/a

Table 9.6: 20 most frequent exceptions in the phonology of Modern Cantonese.

tion in the derivation of MB *tsɿ* (to test) from SMC **tʂac* (Chen76, p.227).

These two exceptions were followed in frequency by the unexpected aspiration of an unvoiced initial and the unexpected non-application of the AC-SPLIT rule, both of which occurred in Modern Cantonese 31 and 30 times respectively, approximately 1.1 percent of the total sample. Other exceptions were much fewer in number. Discoveries such as this demonstrate the value of a systematic computational analysis of the original phonological data.

9.2.2 Chen76 and CN84 on exceptions

The position of Chen76 and CN84 with respect to such exceptions is that they are to be classed as diachronic rules if they satisfy the following two criteria:³

³Newman (1996) also discusses many interesting reasons for the presence of exceptions in Chinese phonology and their status. Several reasons given, such as imitation of prestigious dialects, borrowing, etc. would seem to argue for exceptions being treated as footnotes only and not being elevated to the status of diachronic rules.

P-rule (MB)	Frequency	P-rule (MC)	Frequency
t1-split	1102	t-split	2578
devoice	639	io-flip	884
palat	484	devoice	587
t4-tripart	383	round	557
apocope	368	rime	522
retrof-2	366	deret	462
degemination	365	length	372
med-dism(a)	361	retrof	344
fusion	347	unglide	335
a-merge	340	aspir	221
rimemic	263	raise	213
raise-il	237	fusion	193
raise-u	218	back-asm	171
vocal	216	glide-ins	164
aspir	201	color	157
procope	171	glide	153
raise-i2	154	x-weak	147
fricat	153	diphthong	139
coalesce	152	fricat	117
syllab	147	lab-dism	112

Table 9.7: 20 most frequent rules in the phonology of Modern Beijing and Modern Cantonese.

- They occur sufficiently frequently
- Some general condition exists which can predict their application or non-application

The first condition seems readily satisfied at least by the most frequent exceptions encountered, for it is noteworthy that the delabialisation of a vowel following velars or glottals in Modern Beijing phonology warranted the formation of the U-DELETE rule (Chen76, p.189–190) which occurred only 25 times in the total sample of words. But it is not clear from the PFSA if the second condition is also satisfied since linguistic context is not preserved in the diagrams.

These frequent exceptions can thus best be viewed as problems to be tackled by linguists who, now armed with their actual frequencies in the full data corpus, can attempt to find general conditions to elevate them to the status of diachronic

rules. For instance, it is obvious from the PFSA that there are also a number of exceptional instances of the chameleon vowel rule feeding the phonotactic condition RIMEMIC. These are not readily obvious from a reading of Chen76 and CN84.

What is even more significant about the PFSA is that they also clearly present the probabilities of the actual rules and rule sequences as they applied in the lexicon of the language. It is easy to see, for instance, that in Modern Cantonese the T-SPLIT rule was the most frequent initial rule occurring with a probability of 2538/2714 or 0.93 and that an exceptional application of it has a probability of 0.039. Besides, the PFSA also indicate probabilities of sequences of rules and which rule is most likely to occur after which other one in some circumstances. Such relative probabilities are hard to discern from the tables provided in the appendices of Chen76 and CN84. Various interesting facts like this come to light when studying the PFSA which embody the original linguistic hypothesis with matter from the language's real vocabulary.

9.2.3 Relation between the PFSA and RC diagrams

It is also worth mentioning that the PFSA may seem to resemble the graphical representation of the RCs in Chen76 and CN84 (reproduced here in Figures 4.2 and 4.3) superficially, but in fact they are more significant. They represent the actual sequences of rules used in deriving the forms rather than just the ordering relation among them.⁴ The frequencies on the arcs also give an idea of the total number of times a particular rule was applied at a certain stage of the derivation process in the corpus of words.

As mentioned earlier, the arcs in the RC diagrams trace ordering relationships between the rules whereas the arcs in the PFSA show the actual rule applications as well as their ordering. It is, however, possible to note some similarity between them as the PFSA structures are implicitly determined by the ordering relationships given in the RCs. In the RC of Modern Cantonese

⁴Certain rules that rarely apply may not show up in the PFSA diagrams, but that is because arcs representing them have been deleted from the diagram. The MML computation process, however, accounted for those as well.

(Figure 4.3), for example, the ordering of the T-SPLIT rule before the AC-SPLIT rule, and the AC-SPLIT rule before the IO-FLIP rule can be seen in the transition from state 0 to state 1 on “t-split”, state 1 to state 18 on “ac-split” and state 18 to state 36 on “io-flip”. However, this cannot be carried too far. The RC consists of sequences of rules which are ordered internally within their respective sequences, but the sequences themselves are not ordered with respect to other sequences. In the case of more than one sequence of rules having to apply to a given reconstructed form, a procedural decision was made to concatenate the sequences in the order in which they are introduced in Chen76 and CN84. For example, we see a transition sequence from state 0 to state 1 on “t-split”, 1 to 22 on “fricat”, 22 to 101 on “retrof” and 101 to 153 on “deret”. Note that the T-SPLIT rule is not ordered with respect to the sequence of rules FRICAT, RETROF and DERET.

The embodiment of an RC in a canonical PFSA is readily apparent if the exceptions are removed and predecessor and successor symbols are determined for every symbol in it. The information so determined would then indicate the bleeding and feeding relationships described in Chen76 or CN84 giving a list of which rules can apply before and after which other rules. They would also include, however, the extra ordering criteria imposed for the purpose of obtaining complete derivations as discussed in the previous paragraph, namely the order in which the rules have been discussed in Chen76 and CN84, which might need to be filtered out. A similar attempt can also be made with the reduced PFSA and this would result in a modified RC. This is likely to be more generalised than the original RCs. The actual significance of such a generalisation of an RC would need to be subjected to the critical analysis of historical linguists.

9.2.4 Significance of diachronic regularities in MB and MC

We make a further observation about the results that the complexity measures for the phonologies of Modern Beijing and Modern Cantonese are not immensely different from each other. Interestingly also, while the MML of the canonical PFSA for Modern Beijing is greater than that for Modern Cantonese, the MML

of the reduced PFSA for Modern Beijing is less than that for Modern Cantonese. While the differences might be within the margin of error in constructing the derivations and the PFSA, it is possible to speculate that the generalisation process has been able to discern more structure in the diachronic phonology of Modern Beijing than in Modern Cantonese. From a computational point of view, one could say that the scope for further generalisation of the diachronic rules is greater for Modern Cantonese than for Modern Beijing or that there is greater structure in the evolution of Modern Beijing from Middle Chinese than in the evolution of Cantonese. One could perhaps claim that this is due to the extra liberty taken historically by current Modern Cantonese speakers to introduce changes into their language as compared to their Mandarin speaking neighbours. But it would be naïve to conclude so here. The study of the actual socio-cultural factors which would have resulted in this situation is beyond the scope of this dissertation.

It is also no surprise that the MMLs obtained for the two languages are not very different from each other although the difference is large enough to be statistically significant. Indeed, this is to be expected as they are both contemporary and have descended from a common ancestor. We can expect more interesting results when deriving complexity measures for the phonologies of languages that are more widely separated in time and space. It is here that the method described in this dissertation can provide an effective tool for subgrouping.

9.2.5 Glottochronology revisited

In this context, it is tempting to postulate a chronological thesis for the time separations of the two languages from Middle Chinese. Here, at least, the results are consistent, since the MMLs are close together for the two contemporary languages, but we realise that much further work is necessary before any such thesis is made. We need to study several such distance measures in other language families and draw up a thesis based on a number of MML and chronological separation pairs. Besides there is the problem of which theory

of linguistic change one subscribes to. One of the common traditional views is that language complexity increases with time. In this view, exceptions to the diachronic phonology are caused by further changes to sounds which result from phenomena such as paradigmatic levelling, analogy, borrowing of pronunciation from prestigious dialects, etc. If this were the case, then the measure we derive would be a true distance measure since it indicates the presence of more irregular forms and thus greater complexity. However, there is also an alternative view in the theory of lexical diffusion, first proposed by Wang (1969), according to which a language change starts off at one point in the vocabulary and gradually spreads to all the phonemes in it over time.⁵ In this view then, a completed change leaves fewer exceptions than an incomplete change or a change in the process of happening. Thus a larger number of exceptions and a greater complexity in the diachronic phonology will point to changes in progress rather than completed changes and hence to a language that is actually nearer to the parent language. If this were the case, the distance measures proposed in this dissertation would be considered *closeness* measures! Here however, we make no attempt to argue for any one particular model of linguistic change. We only provide a complexity measure that is open to interpretation by linguists.

9.3 Conclusion and Prospects

The work described in this chapter was originally presented at the World Archaeological Congress III conference, New Delhi, in 1994 and has undergone significant revision since then, mostly due to the suggestions of paper reviewers. One of the points raised by a reviewer was that the methodology described could in fact be put to better use than in just deriving distance measures. The suggestion was that it should be possible, in principle, to use the method to choose between competing reconstructions of protolanguages as this tends to be a relatively more contentious area than subgrouping.

⁵Further and more recent descriptions of lexical diffusion can be found, for example, in Robinson (1977), Cheng and Wang (1977), Wang and Cheng (1977), Chen (1977, pp. 211–221), Hock (1986, pp.649–652) or Cheng (1994b).

It is indeed possible to use the method to do this — we could retain the basic procedure, but shift the focus from studying two descendants of a common parent to studying two proposed parents of a common set of descendants. A protolanguage is usually postulated in conjunction with a set of diachronic rules that derive forms in the descendant languages. We could thus use the methodology described in this chapter to derive a large number of forms in the descendant languages from each of the two competing protolanguages. Since descriptive economy is one of the deciding factors in selecting historical linguistic hypotheses, the size of each body of derivations, suitably encoded in the form of automata, in conjunction with other linguistic considerations can then be used to decide the plausibility of that reconstruction. However, this is not the purpose to which it has been put in the present dissertation. The reason for this was not only the unavailability of data for the latter task, but also the relatively easy availability of data for the former. Further study of that line of approach is left as a topic for future research.

In addition to possible work in applying the method as outlined in this dissertation to two daughter languages in a new language family or to two proposed reconstructions of a different proto-language, there is also scope for future work in improving PFSA inferencing strategies. As mentioned earlier, the simba search procedure could perhaps be improved by incorporating a different annealing schedule, or variations of the sk-strings approach could be tried. For example, work could be done along the lines of incorporating predecessor substrings as parameters into the merging criterion in addition to successor substrings in the algorithm.

Finally, the inferencing procedures could also be applied to various other domains. One obvious candidate that readily springs to mind is in the measure of dissimilarity between DNA strands. Indeed, one can see the application of this method of dissimilarity quantification even in such fields as psychology for modelling human behaviour or history in trying to discern general patterns in global events. While such interesting work is definitely possible and worthwhile, it is deemed best to conclude the present dissertation at this point and embark

on these as new projects.

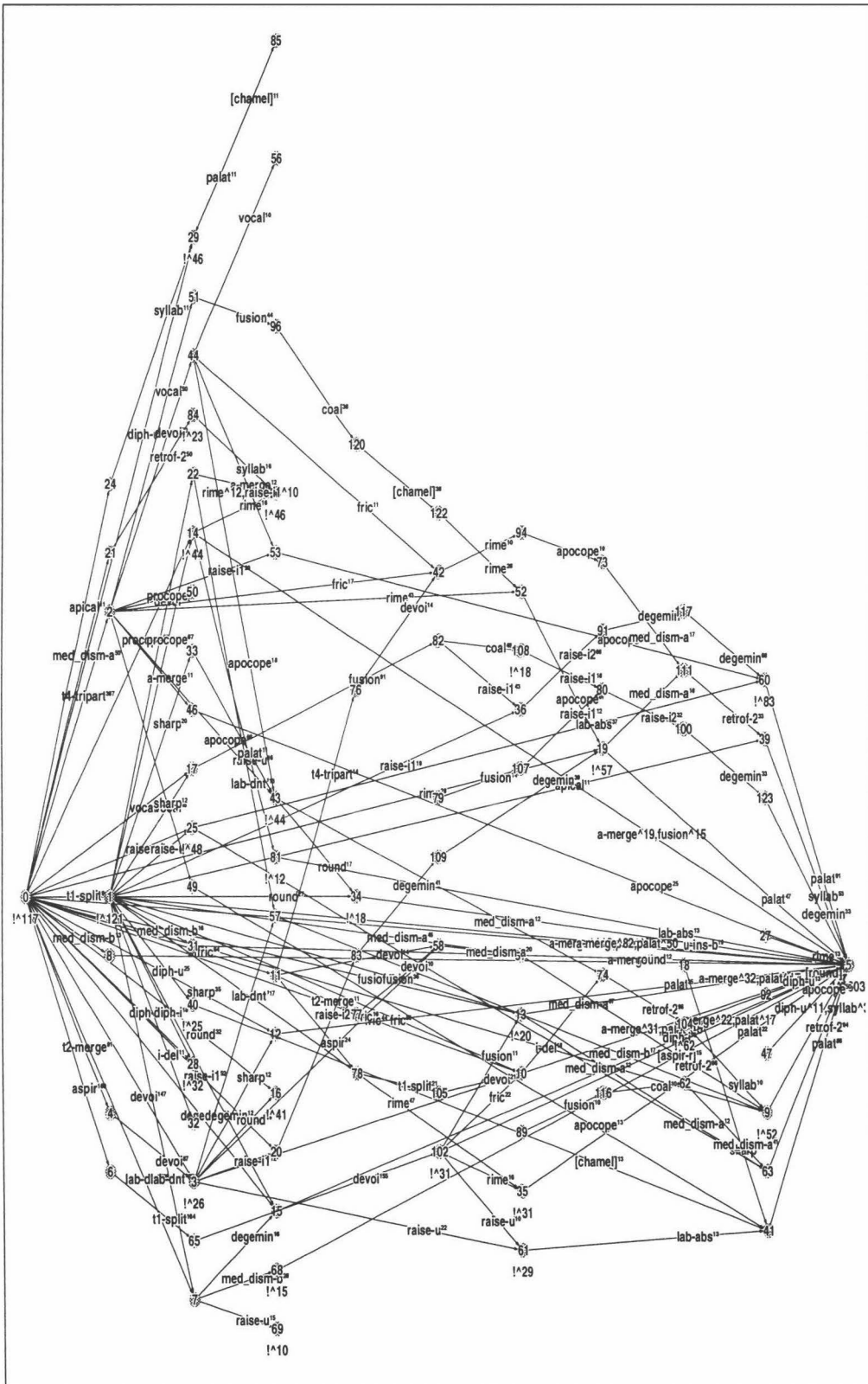


Figure 9.1: Reduced PFSA for the diachronic phonology from Middle Chinese to Modern Beijing (Allophonic detail excluded).

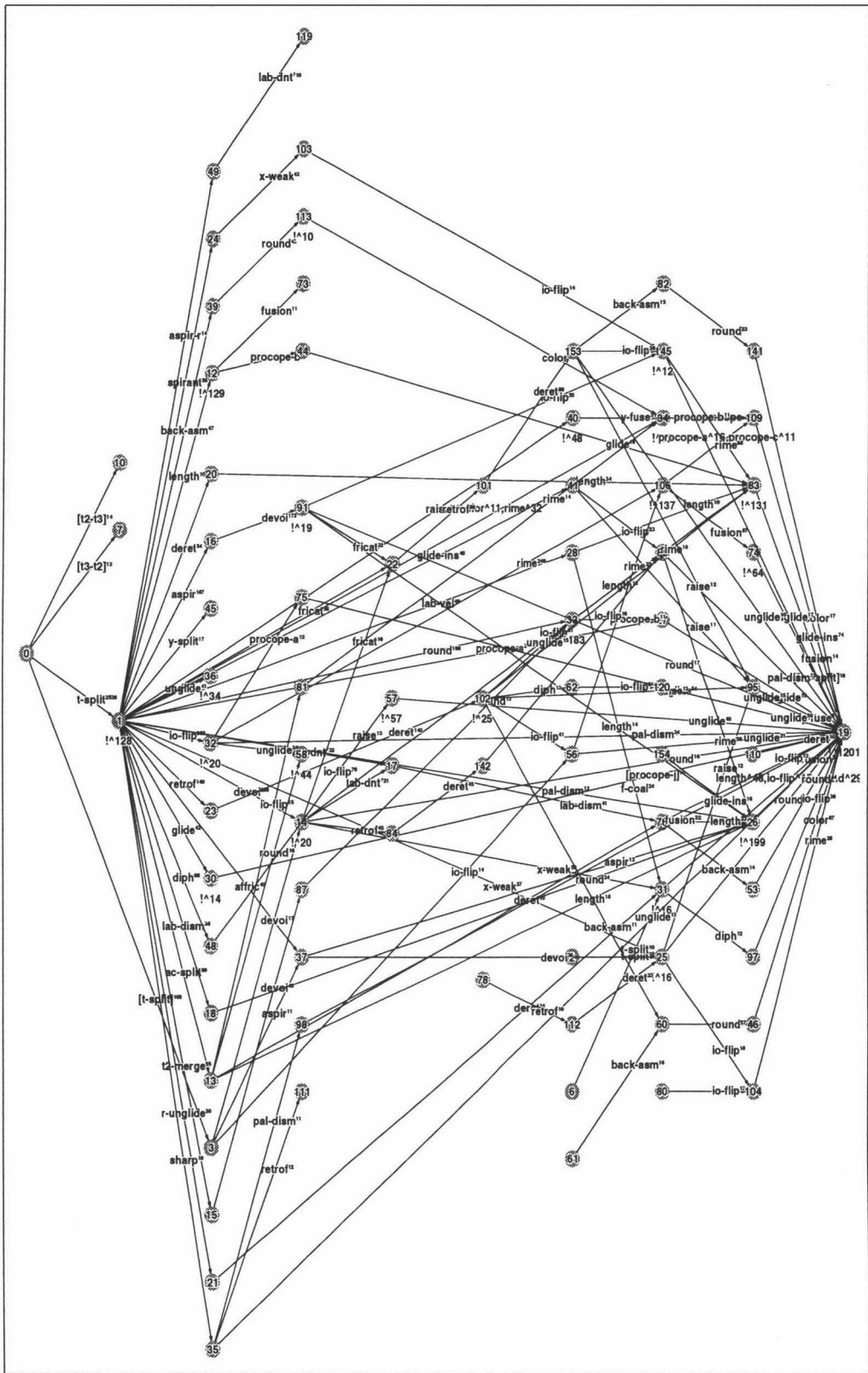


Figure 9.2: Reduced PFSA for the diachronic phonology from Middle Chinese to Modern Cantonese (Allophonic detail excluded).

Appendix A

Guide to Using the Programs

A.1 Introduction

A number of programs were developed to test the various algorithms discussed in this dissertation. They are: *randpfsa*, which generates a random PFSA within given limits, *genstr*, which generates random strings using a given PFSA, *canon*, which constructs a minimal canonical PFSA from a given set of strings, *dfa*, which determinises a given non-deterministic automaton using the subset construction algorithm (Rabin and Scott 1959) and minimises it (Aho et al. 1986, pp.141–144), *ktail*, *skstr*, *beams* and *simba* which all perform generalisations of input PFSA, *grfit* and *groupnodes* which postprocesses the output of a PFSA printing program to make it suitable for graphical presentation using Jos van Eindhoven’s *graphplace* program.¹ For best results, it is recommended that the user also have a compiled version of *graphplace* and *ghostview* available.

These program binaries are available with a brief description of how to use them from the URL “[ftp://fims-ftp.massey.ac.nz/ARaman/pfsa.\[arch\].tar.gz](ftp://fims-ftp.massey.ac.nz/ARaman/pfsa.[arch].tar.gz)” where [arch] stands for “sparc”, “alpha”, “i486”, or “indy”. The source code is available on request from the author.

The programs have all been implemented with the Unix command line interface in mind. Thus they conform to the general Unix way of doing things through pipes and redirections. The output of one program may frequently be piped into another or into other Unix tools. The programs have been designed to facilitate this.

In general, the following conventions have been followed. PFSA are specified as a list of “source target symbol frequency” tuples, delimited with either a semicolon or a newline. White space in PFSA specification is ignored, as is any text on a line following a hash “#” character up to the end of that line. Source

¹ *Graphplace* reads in an ASCII graph specification and outputs placement information on visual media or optionally a postscript file that can be printed or viewed using a postscript viewer such as *ghostview*. In order to prettyprint the graph, a small change was made to the Postscript prologue used by *graphplace*. This allows the printing of arc frequencies in superscript font above the symbols. To use this feature version of the file *graphpro.ps* supplied with the PFSA distribution must be installed in the lib directory (\$HOME/lib) of the user instead of the original version in the *graphplace* distribution.

specification is optional and defaults to either state 0 or to the most recent source specified. The frequency of transition is also optional and defaults to 1. The following is an example PFSA specification:

```
# Gaines' (1976) 4 state machine
# source target symbol frequency
0 1 c 4
  3 b 3
1 1 a 9
  2 b 7
2 0 \n 7
3 1 b 3
```

State 0 is a special state and every PFSA must have a state numbered 0. Transitions on delimiter symbols are assumed to return to state 0. It is also assumed that the specified PFSA is consistent, i.e. the total number of incoming transitions is equal to the total number outgoing and that every state has at least one transition leaving it.

A string to be processed as input to a PFSA manipulation program is a series of colon (:) separated labels ending with a delimiter symbol, which is a newline by default. Labels which make up the string may not start with a numeric character. Specifically, they may only start with one of the characters matching the regular expression “[A-Za-z!~()]” although they may contain any character other than whitespace, a hash sign, a colon or a semicolon in them. The following are valid strings:

```
a:b:c:d:e:
hello:world:
```

A file (test.strings) is supplied with the distribution which contains a list of sample strings that can be used to test the various programs.

Default input to all the programs comes from standard input and the default output is to standard output. Input may optionally be specified on the command line, but the extension of the filename must conform to what is expected. A program that expects a PFSA as input will expect a filename ending with “.pfsa” if one is specified. A program that expects strings for input will expect a filename ending with “.strings”.

The default output of all programs is to standard output if no input filename is specified. If input is explicitly specified, then the output file name is created by substituting the extension part of the input filename appropriately. *Genstr* will substitute the “.pfsa” part of its input filename with “.strings” and *canon* will substitute the other way. The various optimisation programs will substitute the “.pfsa” part of the filename by “.opfsa”. This behaviour can, however, be overridden by specifying an explicit output filename with the “-o” option.

Most of the programs in the distribution have some common options. These are described here: The “-v” flag causes the program to enter verbose mode. This prints extra information in the final output. In particular, PFSA generation programs print the MML of the PFSA and other miscellaneous information

such as the number of states and the total number of arcs. Programs that output PFSA support the “-g” flag which causes them to output the PFSA in a format suitable for processing by the *graphplace* program. All programs support the “-D” flag which allows the user to change the delimiter² symbol from “\n”.

In addition, the “-d” flag may be used one or more times on the command line to make a program enter various levels of debugging. This causes copious amounts of debugging information to be printed while the program is executing.

All the PFSA optimisation programs (*ktail*, *skstr*, *simba* and *beams*) support the “-a” option which will terminate the program after a specified number of seconds if given. This helps to break out of extremely time consuming executions automatically. By default, a program will continue until it either completes or is stopped by a user generated signal. Most programs can have debugging dynamically toggled by sending them a SIGUSR1 signal during execution. This is especially useful in checking the status of long running optimisations. Sending SIGUSR2 will print an interim progress report.

Although the programs have all been designed with the Unix philosophy in mind, Unix style manual pages have not been provided for any of them. Installing these pages on a multi-user system where it is probably irrelevant to other users of the system tends to be wasteful and separate manual pages could sometimes get lost! Instead, each program comes with built in help which can be elicited with the “-h” option.

A.2 Canon

Canon reads in a list of strings and generates a minimal canonical automaton that represents the input strings. The canonical PFSA is just a trie of the input with frequencies on each arc reflecting how many times the arc was traversed in creating that structure. The following are some examples of usage of *canon* (“%” below is the shell prompt. Command outputs are not given.)

```
% canon test.strings
% canon -v test.strings
% canon -v <test.strings
```

It can also be used with the “-a” option to read in a preliminary PFSA specification and create frequency counts on its arcs by reading in a list of strings. In its second version, it will reject strings that it finds unacceptable with the given PFSA specification. Here are some examples:

```
% canon -va test.pfsa <test.strings
% canon -a test.pfsa
m:n:o:e:f:g:
m:n:o:
```

²The author himself has rarely ever used a different delimiter and so this option has not been extensively tested.

```
h:s:r:
^D
```

A.3 Genstr

Genstr reads in a PFSA specification and prints a number of strings it generates randomly using it. The “-n” option controls the number of strings output. The default is 100. The “-r” option is given to enable *genstr* to output different random strings by seeding the random number library with the current value of the system clock. A “-R” option was added late to *genstr* to cause it to print out the input PFSA renumbered breadth-first.

Genstr is also useful in just printing out the MML of a given PFSA. The “-b” and “-e” options cause *genstr* to print the MML of the PFSA computed using formulas 5.6 and 5.4 respectively and exit. Here are some examples:

```
% genstr test.pfsa
% genstr -b <test.pfsa
% genstr -e gaines.pfsa
% canon <test.strings | genstr | more
```

A.4 Randpfsa

Randpfsa generates a random PFSA within the specified limits of size. Like *genstr*, *randpfsa* supports a “-r” option which seeds the random number library. Otherwise, it generates the same random PFSA each time! The user has control over the maximum number of states, maximum number of arcs and cardinality of the generated automaton. These are specified using the “-n”, “-a” and “-c” options. Also, a “-l” option is available which controls a bias towards tight loops (self-transitions). The “-l” option takes an argument from 1 to 10 which determines how likely (or unlikely) a self-transition is. If this value is 0, there will be no self-transitions (however there may still be loose loops). If, on the other hand, this value is 10, the PFSA will have only state 0 and all transitions will be to and from this state. Values in-between select intermediate values of the bias. The default value is 5, which means that there is no bias towards tight loops.

A.5 Skstr, Ktail, Beams and Simba

These programs are all either links to a program called *opt* or are copies of it. The algorithm to execute is determined by looking at the program name. *Ktail* executes the k-tails algorithm, *skstr* executes the sk-strings algorithm, *beams* and *simba* execute the beam search and simba search algorithms on the input PFSA. The “-H” option can be used to select one of the 6 heuristics — AND, OR, LAX, STRICT, XENTROPIC and VARDIST — for sk-strings; the default is OR.

Needless to say, the output of one of these programs can directly be piped into the input of another for further minimisation attempts. The following examples illustrate the use of these programs:

```
% skstr <test.pfsa
% skstr -d test.pfsa
% skstr -H XENTROPIC -e0.1 <test.pfsa
% canon <test.strings | skstr -HLAX -t3 | beams -vb100
% skstr -v <test.pfsa | beams -vb100
% skstr -dv test.pfsa
% beams -dv test.pfsa
% beams test.pfsa
% simba -dv -t10 <test.pfsa
```

A.6 Dfa

Dfa determinises the input PFSA using the subset construction algorithm. If the “-m” flag is given, the determinised automaton is minimised in its number of states as well. Since the frequencies of transitions on the arcs of the PFSA will be irrelevant to the determinised machine, these are ignored in the process and will not be printed. However, the machine output by *dfa* is still readable by programs that require PFSA input, notably *canon* which can read it in and reconstruct arc frequencies via its “-a” option. The following is an example of its usage:

```
% canon <test.strings | skstr -H LAX -p50 -t3 | \
> dfa -m >/tmp/test.pfsa
% canon -va /tmp/test.pfsa <test.strings | more
```

A.7 Grafit and Groupnodes

Grafit reads in a PFSA and prints it out in a format suitable for van Eindhoven’s *graphplace* program. It is different from the “-g” option of the other programs in that it can be used on existing PFSA. In addition, it supports the “-t” option which can be used to give the printed PFSA a title.

Groupnodes is a simple shell script which enables *graphplace* to constrain the vertical distribution of nodes. *Groupnodes* takes as argument node numbers, which it will arrange for *graphplace* to assign the same y-coordinate in the figure. Optionally, *groupnodes* takes a first non-numeric argument which will label each group of nodes it arranges for identical y-coordinates. When nodes are grouped together thus, *graphplace* must be informed of this grouping by using its “-g:” option.

Figure A.1 shows the 5-state Andreae PFSA without nodes grouped vertically, generated using the command:

```
% grafit <a5.pfsa | graphplace -pc | ghostview -
```

The same PFSA is shown in Figure A.2 with nodes 0, 2 and 4 grouped along the same y-coordinate, generated using the command:

```
% cat a5.pfsa | grafit | groupnodes 0 2 4 | \
> graphplace -pc -g: | ghostview -
```

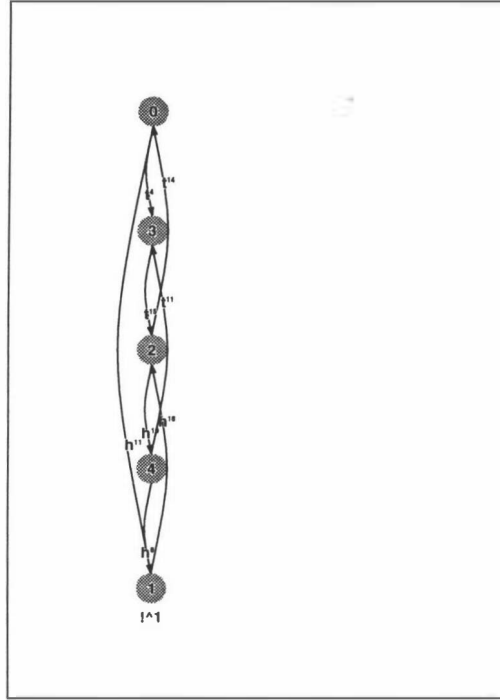


Figure A.1: Andreae's 5 state machine without vertical node grouping

The following examples further illustrate the use of *grfit* and *groupnodes*.

```
% canon <test.strings | skstr | grafit | \
> groupnodes 0 1 2 3 10 12 | graphplace -pg: | lpr -Plaser
% skstr <test.pfsa | grafit -t "date" | groupnodes x 0 1 | \
> groupnodes y 2 3 | graphplace -pg: | ghostview -
```

A.8 DISCLAIMER

The programs are provided "as is" without any express or implied warranties whatsoever. Neither the author, nor the distributor is liable for any damages that may be caused by their use. Work done using these programs must cite either Raman and Patrick (1997d) or this dissertation.

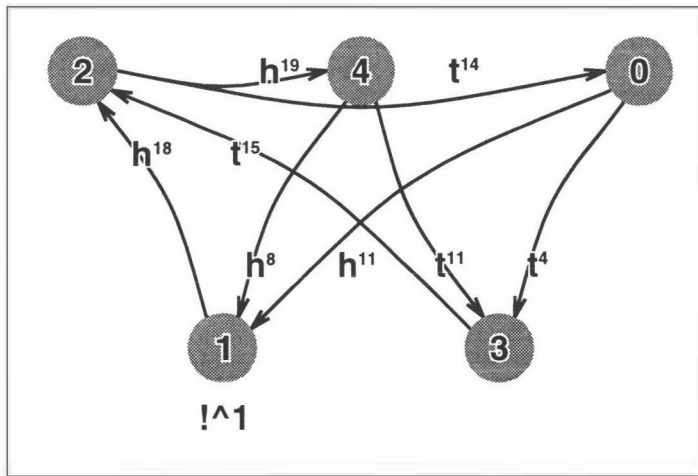


Figure A.2: Andreae's 5 state machine with nodes 0, 2 and 4 grouped vertically

Appendix B

Reduced PFSA

B.1 PFSA for SMC>MC (Diachronic only)

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
0	1	t-split	2538	1	51	geng-zeng	3	13	80	x-weak	3
0	2	[t3-t1]	4	2	25	t-split	10	13	19	fusion	1
0	3	[t-split]	105	2	65	[t-split]	3	13	20	procope-a	1
0	4	[t4-t1]	1	3	26	length	3	13	31	lab-vel	3
0	5	[t4-t3]	2	3	25	ac-split	7	13	81	io-flip	15
0	2	[t2-t1]	9	3	14	devoi	17	13	58	round	16
0	6	[t1-t2]	9	3	37	aspir	11	13	35	sharp	3
0	7	[t3-t2]	13	3	52	lab-dnt'	2	13	19	[a-front]	1
0	6	[t1-t3]	5	3	53	lab-dism	4	13	82	back-asm	8
0	8	[t4-t2]	1	3	54	fricat	2	14	0	\n	20
0	9	[t1-t4]	1	3	55	retrof	2	14	83	length	13
0	10	[t2-t3]	14	3	26	x-weak	2	14	19	aspir	2
0	11	[t2-t2]	1	3	1	lab-vel	1	14	17	lab-dnt'	21
1	0	\n	128	3	26	glide	2	14	26	lab-dism	2
1	12	length	169	3	56	io-flip	14	14	31	deret	2
1	13	ac-split	99	3	57	round	8	14	55	r-unglide	3
1	14	devoi	269	3	58	pal-dism	1	14	22	fricat	16
1	15	t2-merge	55	3	19	color	2	14	55	retrof	2
1	16	aspir	187	3	36	unglide	1	14	84	retrof	45
1	17	lab-dnt'	32	3	19	rime	1	14	31	x-weak	50
1	18	lab-dism	38	3	59	affric	1	14	125	lab-vel	1
1	19	[de-aspir]	2	3	38	procope-d	2	14	19	lab-vel	1
1	20	deret	54	3	19	[a-front]	1	14	85	io-flip	51
1	21	r-unglide	30	3	60	back-asm	2	14	19	round	16
1	22	fricat	58	3	30	raise	7	14	19	color	2
1	23	retrof	140	3	61	glide-ins	2	14	35	sharp	3
1	24	spirant	58	3	26	[aspir]	2	14	19	unglide	6
1	25	x-weak	37	3	45	y-split	2	14	19	rime	3
1	26	fusion	9	3	46	pre-color	2	14	86	affric	1
1	27	procope-a	21	3	62	diph	3	14	18	back-asm	4
1	28	lab-vel	49	3	14	aspir-r	1	14	57	raise	13
1	29	procope-b	2	4	63	[t-split]	1	14	42	y-split	1
1	30	glide	42	5	62	t-split	1	14	48	diph	7
1	31	[unglide]	4	5	64	[t-split]	1	14	38	[procope-x]	2
1	32	io-flip	362	6	31	t-split	10	14	124	[x-kh]	1
1	33	round	198	6	66	[t-split]	4	15	87	devoi	45
1	26	pal-dism	12	7	66	t-split	9	15	88	aspir	1
1	34	color	11	7	67	[t-split]	4	15	89	glide	1
1	35	sharp	19	8	68	[t-split]	1	15	73	io-flip	1
1	36	unglide	41	9	69	t-split	1	15	73	round	3
1	34	rime	32	10	70	t-split	8	15	90	affric	3
1	37	affric	40	10	71	[t-split]	6	15	30	raise	2
1	38	procope-d	9	11	72	t-split	1	16	91	devoi	173
1	19	[a-front]	1	12	0	\n	129	16	92	t2-merge	2
1	39	back-asm	47	12	19	[t3-t1]	1	16	93	lab-dnt'	3
1	40	raise	104	12	19	[de-aspir]	1	16	31	deret	1
1	41	glide-ins	46	12	73	fusion	11	16	94	fricat	4
1	42	[aspir]	4	12	74	procope-a	8	16	41	x-weak	1
1	43	x-lab	2	12	44	procope-b	14	16	19	round	1
1	44	procope-c	9	12	19	[aspir]	2	16	94	affric	1
1	45	y-split	17	12	75	procope-c	2	16	95	raise	2
1	46	pre-color	8	12	19	[n'-del]	1	17	0	\n	9
1	47	[ac-split]	2	12	75	[procope-x]	1	17	76	lab-dism	41
1	48	diph	68	13	26	length	18	17	96	unglide	1
1	19	[u-o]	1	13	17	lab-dnt'	3	17	19	raise	5
1	49	aspir-r	14	13	76	deret	10	17	97	diph	1
1	50	[procope-j]	1	13	77	r-unglide	1	18	0	\n	7
1	19	[schwa-i]	2	13	78	fricat	6	18	36	io-flip	4
1	13	[back-asm]	1	13	79	retrof	6	18	26	round	34
1	25	lab-dnt	4	13	70	spirant	6	19	0	\n	1201

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
19	19	[l-t]	1	27	61	glide-ins	8	38	19	glide-ins	6
20	0	\n	9	27	97	diph	3	38	19	procope-m	4
20	83	length	34	27	19	[!ac-split]	1	39	113	round	48
20	98	pal-dism	1	27	19	[j-ins]	3	40	0	\n	48
20	19	color	2	28	0	\n	9	40	109	procope-a	1
20	19	unglide	1	28	83	length	10	40	109	procope-b	11
20	60	back-asm	4	28	19	io-flip	5	40	19	unglide	8
20	61	glide-ins	3	28	26	round	7	40	114	diph	1
20	19	[a-ins]	1	28	19	[aspir]	1	40	34	y-fuse	31
21	31	deret	34	28	31	f-coal	24	40	19	nasal-syl	7
21	32	round	2	29	70	glide	2	41	0	\n	4
22	99	devoi	8	30	0	\n	14	41	85	io-flip	23
22	100	aspir	2	30	19	length	1	41	60	back-asm	9
22	101	retrof	86	30	109	procope-b	2	41	95	raise	11
23	102	deret	140	30	105	io-flip	8	42	19	lab-vel	2
24	103	x-weak	43	30	19	round	1	42	19	raise	1
24	28	lab-vel	7	30	19	sharp	2	42	19	[s-ts]	9
24	43	x-lab	7	30	19	unglide	2	43	115	f-coal	9
24	48	diph	1	30	19	[aspir]	1	44	83	glide	19
25	0	\n	16	30	97	diph	7	44	61	glide-ins	4
25	19	length	4	30	19	[procope-j]	10	45	0	\n	1
25	66	devoi	2	30	19	[procope-x]	2	45	91	x-weak	1
25	21	aspir	1	30	19	[round]	1	45	116	lab-vel	8
25	17	lab-dnt'	1	30	19	y-fuse	2	45	97	glide	2
25	53	lab-dism	4	31	0	\n	16	45	57	io-flip	1
25	31	deret	2	31	19	length	48	45	117	io-flip	5
25	112	retrof	4	31	116	lab-dism	2	45	116	x-lab	1
25	128	spirant	2	31	79	retrof	1	46	19	color	67
25	19	fusion	1	31	19	fusion	3	46	19	unglide	3
25	40	fusion	3	31	75	procope-a	1	47	118	procope-a	1
25	31	lab-vel	1	31	26	glide	2	47	19	[u-o]	1
25	30	glide	1	31	19	io-flip	15	48	57	io-flip	76
25	35	io-flip	3	31	76	io-flip	1	49	119	lab-dnt'	10
25	104	io-flip	18	31	19	round	29	49	19	lab-dism	1
25	19	round	11	31	167	pal-dism	1	49	120	raise	3
25	19	pal-dism	1	31	19	[a-front]	2	50	1	glide	1
25	35	sharp	4	31	53	back-asm	7	51	121	io-flip	1
25	46	sharp	1	31	60	back-asm	2	51	122	rime	1
25	19	unglide	9	31	19	raise	2	51	123	x-lab	1
25	19	rime	5	31	97	diph	12	52	31	lab-dism	2
25	39	back-asm	1	32	0	\n	20	53	120	round	3
25	60	back-asm	6	32	75	procope-a	12	53	19	round	44
25	95	raise	12	32	75	procope-b	5	54	66	retrof	2
25	147	y-split	2	32	19	pal-dism	34	55	26	deret	7
25	62	diph	2	32	58	unglide	38	56	19	pal-dism	3
25	116	diph	2	32	106	rime	249	56	19	unglide	21
25	19	[t-n]	1	32	38	procope-d	2	56	106	rime	23
25	19	[k-l]	1	32	19	[aspir]	2	56	25	rime	8
25	19	[k-ins]	1	32	75	[procope-x]	1	57	0	\n	57
25	19	[iu-o]	1	32	19	[s-ts]	1	57	75	procope-a	5
25	19	[a-oe]	2	33	0	\n	183	57	75	procope-b	1
26	0	\n	199	33	19	[de-aspir]	1	57	19	unglide	40
26	26	length	2	33	83	procope-a	9	57	75	procope-d	2
26	19	length	2	33	27	procope-b	11	57	75	[procope-x]	5
26	153	devoi	8	33	19	unglide	15	57	19	[s-ts]	2
26	164	lab-dism	2	33	107	raise	1	57	19	y-fuse	5
26	19	[de-aspir]	3	33	42	[aspir]	3	58	0	\n	44
26	19	fusion	31	33	74	x-lab	1	58	19	procope-a	1
26	120	fusion	1	33	19	[t-f]	2	58	19	[aspir]	8
26	36	fusion	1	34	0	\n	92	58	19	[x-kh]	3
26	38	procope-a	3	34	109	procope-a	5	59	90	aspir	1
26	75	procope-b	2	34	109	procope-d	12	60	46	round	57
26	120	io-flip	1	34	19	glide-ins	1	61	0	\n	6
26	19	io-flip	3	34	19	[aspir]	1	61	60	back-asm	16
26	19	round	1	34	109	procope-c	9	62	120	io-flip	31
26	98	pal-dism	1	34	19	[n-ng]	1	63	127	length	1
26	19	color	3	35	110	r-unglide	1	64	1	devoi	1
26	19	unglide	3	35	35	fricat	1	65	129	length	1
26	108	unglide	3	35	111	retrof	12	65	80	devoi	1
26	53	back-asm	6	35	98	pal-dism	11	65	130	r-unglide	1
26	15	back-asm	1	35	19	unglide	13	66	0	\n	1
26	97	diph	1	35	98	rime	1	66	12	length	1
26	75	[procope-j]	1	35	75	procope-d	1	66	133	devoi	1
26	74	[a-i]	1	36	0	\n	34	66	131	aspir	2
26	19	[!ac-split]	7	36	19	[de-aspir]	3	66	31	deret	5
26	19	[s-ts]	3	36	19	deret	1	66	134	r-unglide	1
26	19	final-dism	3	36	19	round	1	66	135	fricat	1
26	19	[ts-s]	4	36	19	raise	3	66	79	retrof	1
26	19	[p-t]	1	36	19	[round]	1	66	19	round	1
26	19	[kw-w]	2	36	19	[x-tsh]	2	66	132	affric	1
26	19	[m-n]	2	36	19	[j-h]	2	66	109	procope-c	1
26	19	degemin	1	36	19	[f-p]	1	66	14	aspir-r	1
26	19	[m-w]	1	37	25	devoi	36	67	21	devoi	3
26	19	[n-m]	1	37	26	aspir	12	67	30	glide	1
27	0	\n	1	37	126	fricat	1	68	136	round	1
27	19	glide	15	37	112	retrof	4	69	137	retrof	1
27	19	round	1	38	19	glide	8	70	128	length	1

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
70	19	length	1	91	148	diph	5
70	26	lab-dnt'	1	91	149	geng-zeng	2
70	13	r-unglide	1	92	110	devoi	2
70	78	fricat	4	93	97	lab-dism	3
70	138	retrof	1	94	150	devoi	5
70	80	x-weak	4	95	19	y-fuse	51
70	125	lab-vel	2	96	19	[fh-ph]	1
70	104	io-flip	1	97	19	io-flip	39
70	19	[a-front]	4	98	19	fusion	22
71	70	devoi	4	99	151	retrof	8
71	37	aspir	2	100	152	devoi	2
72	16	devoi	1	101	153	deret	86
73	0	\n	8	102	0	\n	25
73	75	procopce-b	7	102	19	length	2
74	0	\n	64	102	56	io-flip	41
74	154	procopce-a	7	102	19	round	1
74	154	procopce-b	2	102	26	pal-dism	2
74	19	unglide	1	102	19	color	4
74	26	glide-ins	3	102	33	unglide	15
74	19	[aspir]	1	102	83	rime	10
74	109	procopce-c	1	102	60	back-asm	11
74	19	[!ac-split]	19	102	95	raise	13
74	139	procopce-m	1	102	42	[aspir]	3
75	19	glide	54	102	62	diph	13
76	0	\n	2	103	0	\n	8
76	26	length	23	103	26	length	7
76	19	io-flip	12	103	145	io-flip	14
76	53	back-asm	14	103	26	round	7
76	19	final-dism	3	103	19	unglide	2
77	140	deret	1	103	18	back-asm	3
78	112	retrof	10	103	97	diph	1
79	80	deret	8	103	19	[!ac-split]	1
80	98	length	2	104	0	\n	1
80	104	io-flip	12	104	19	unglide	1
80	35	sharp	1	104	19	rime	28
80	53	back-asm	1	104	26	rime	6
81	19	round	1	105	19	procopce-d	6
81	34	rime	14	105	19	[n-del]	1
82	141	round	23	105	19	[ng-n]	1
83	0	\n	131	106	0	\n	137
83	19	[de-aspir]	2	106	74	fusion	87
83	26	fusion	9	106	109	procopce-a	16
83	19	fusion	14	106	109	procopce-b	6
83	19	glide	2	106	154	procopce-d	8
83	19	[a-front]	1	106	26	[aspir]	3
83	19	[i-del]	5	106	109	procopce-c	11
84	142	deret	45	106	19	[!ac-split]	1
85	0	\n	6	106	19	final-dism	1
85	19	pal-dism	12	106	19	[t-p]	1
85	26	unglide	7	106	19	[n-l]	1
85	19	unglide	6	107	58	degemin	1
85	26	rime	59	108	19	[round]	4
86	143	retrof	2	109	0	\n	1
87	0	\n	3	109	19	glide-ins	74
87	83	length	1	110	19	deret	10
87	26	lab-dnt'	2	111	35	deret	4
87	97	lab-dism	1	111	120	deret	8
87	144	fricat	2	112	25	deret	23
87	112	retrof	5	113	0	\n	10
87	26	x-weak	3	113	34	color	33
87	19	lab-vel	1	113	117	unglide	4
87	145	io-flip	7	113	19	[!ac-split]	1
87	33	round	13	114	155	io-flip	1
87	19	unglide	1	115	19	round	3
87	57	raise	5	115	19	raise	7
87	46	pre-color	1	116	26	io-flip	3
88	53	devoi	1	116	19	io-flip	5
89	146	round	1	116	19	round	1
90	61	devoi	4	116	97	f-coal	4
91	0	\n	19	117	0	\n	2
91	26	length	14	117	75	[procopce-j]	8
91	53	lab-dism	6	118	156	glide-ins	1
91	31	deret	4	119	0	\n	4
91	110	r-unglide	1	119	26	lab-dism	6
91	22	fricat	22	120	162	unglide	1
91	110	retrof	2	120	19	unglide	50
91	26	lab-vel	2	121	117	rime	1
91	145	io-flip	55	122	157	[procopce-j]	1
91	19	round	17	123	158	io-flip	1
91	19	color	1	124	74	[de-aspir]	1
91	35	sharp	2	125	53	f-coal	3
91	26	unglide	4	126	19	devoi	1
91	19	rime	4	127	19	[p-i]	1
91	60	back-asm	2	128	19	x-weak	3
91	57	raise	9	129	19	[ng-del]	1
91	147	y-split	2	130	108	deret	1
91	46	pre-color	1	131	31	devoi	2

Src	Dst	Sym	Freq
132	86	devoi	1
133	19	lab-dnt'	1
134	115	deret	1
135	159	retrof	1
136	19	[k-del]	1
137	160	deret	1
138	161	retrof	1
139	74	[p-t]	1
140	42	length	1
141	0	\n	6
141	19	color	17
142	0	\n	6
142	85	io-flip	16
142	162	color	5
142	26	unglide	3
142	19	rime	3
142	163	back-asm	7
142	95	raise	1
142	62	diph	4
143	104	deret	2
144	165	retrof	2
145	0	\n	12
145	19	length	2
145	19	pal-dism	3
145	19	unglide	17
145	83	rime	68
146	19	procopce-b	1
147	97	lab-vel	4
148	166	io-flip	5
149	120	rime	2
150	0	\n	1
150	110	retrof	4
151	167	deret	8
152	21	retrof	2
153	0	\n	7
153	26	length	3
153	98	length	1
153	104	io-flip	3
153	120	io-flip	1
153	145	io-flip	26
153	19	color	3
153	19	unglide	10
153	82	back-asm	15
153	60	back-asm	5
153	95	raise	12
153	62	diph	8
154	19	glide	1
154	26	glide-ins	16
155	19	[b-f]	1
156	104	pre-color	1
157	19	[glide]	1
158	168	unglide	1
159	53	deret	1
160	169	io-flip	1
161	61	deret	1
162	0	\n	1
162	19	[ts-s]	7
163	170	round	7
164	76	length	2
165	171	deret	2
166	57	unglide	5
167	0	\n	1
167	98	length	5
167	120	io-flip	1
167	19	rime	2
168	36	[x-kh]	1
169	172	unglide	1
170	26	color	7
171	173	io-flip	2
172	19	[!fricat]	1
173	162	rime	2

MML = 30379.01 bits

B.2 PFSA for SMC>MC (Diachronic and Allophonic)

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
0	1	t-split	2538	3	64	glide-ins	2	17	98	lab-dism	24
0	2	[t3-t1]	4	3	65	[aspir]	2	17	99	unglide	1
0	3	[t-split]	105	3	45	y-split	2	17	100	raise	5
0	4	[t4-t1]	1	3	47	pre-color	2	17	101	diph	1
0	5	[t4-t3]	2	3	66	diph	4	18	34	io-flip	4
0	6	[t2-t1]	9	3	53	aspir-r	1	18	102	round	41
0	7	[t1-t2]	9	4	67	[t-split]	1	18	19	brk-a	7
0	6	[t3-t2]	13	5	66	t-split	1	19	0	\n	1337
0	7	[t1-t3]	5	5	68	[t-split]	1	20	0	\n	5
0	8	[t4-t2]	1	6	69	t-split	15	20	103	length	34
0	9	[t1-t4]	1	6	70	[t-split]	7	20	104	pal-dism	1
0	10	[t2-t3]	14	7	31	t-split	10	20	19	color	2
0	11	[t2-t2]	1	7	21	[t-split]	4	20	19	unglide	1
1	0	\n	78	8	71	[t-split]	1	20	63	back-asm	4
1	12	length	169	9	72	t-split	1	20	34	acute	4
1	13	ac-split	99	10	73	t-split	8	20	64	glide-ins	3
1	14	devoi	269	10	74	[t-split]	6	20	19	[a-ins]	1
1	15	t2-merge	55	11	75	t-split	1	21	147	aspir	2
1	16	aspir	187	12	0	\n	129	21	31	deret	38
1	17	lab-dnt'	32	12	19	[t3-t1]	1	21	81	retrof	1
1	18	lab-dism	38	12	19	[de-aspir]	1	21	180	round	2
1	19	[de-aspir]	2	12	53	fusion	11	21	148	affric	1
1	20	deret	54	12	26	procope-a	8	22	105	devoi	8
1	21	r-unglide	30	12	44	procope-b	13	22	106	aspir	2
1	22	fricat	58	12	19	[aspir]	2	22	107	retrof	87
1	23	retrof	140	12	76	procope-c	2	23	108	deret	140
1	24	spirant	58	12	19	[n'-del]	1	24	109	x-weak	43
1	25	x-weak	37	12	76	[procope-x]	1	24	110	lab-vel	7
1	26	fusion	9	13	34	length	18	24	43	x-lab	7
1	27	procope-a	21	13	77	lab-dnt'	3	24	49	diph	1
1	28	lab-vel	49	13	78	deret	9	25	0	\n	6
1	29	procope-b	2	13	79	r-unglide	1	25	104	length	2
1	30	glide	42	13	80	fricat	6	25	19	length	4
1	31	[unglide]	4	13	81	retrof	6	25	21	devoi	2
1	32	io-flip	362	13	73	spirant	6	25	18	lab-dism	4
1	33	round	198	13	25	x-weak	3	25	31	deret	2
1	34	pal-dism	12	13	19	fusion	1	25	174	retrof	4
1	34	color	11	13	20	procope-a	1	25	111	io-flip	32
1	35	sharp	19	13	31	lab-vel	3	25	19	round	10
1	36	unglide	41	13	82	io-flip	15	25	35	sharp	5
1	34	rime	32	13	26	round	15	25	47	sharp	1
1	37	affric	40	13	35	sharp	3	25	19	unglide	8
1	38	procope-d	9	13	19	[a-front]	1	25	19	rime	4
1	19	[a-front]	1	13	63	back-asm	8	25	159	back-asm	1
1	39	back-asm	47	14	0	\n	11	25	39	back-asm	1
1	40	raise	104	14	26	length	13	25	63	back-asm	6
1	19	brk-b	1	14	19	aspir	2	25	40	raise	6
1	19	acute	8	14	77	lab-dnt'	19	25	129	raise	1
1	41	glide-ins	46	14	65	lab-dism	2	25	166	y-split	2
1	42	[aspir]	4	14	31	deret	2	25	19	brk-a	5
1	43	x-lab	2	14	83	r-unglide	3	25	66	diph	2
1	44	procope-c	9	14	22	fricat	16	25	19	[iu-o]	1
1	45	y-split	17	14	84	retrof	47	25	19	[a-oe]	2
1	46	brk-a	41	14	31	x-weak	50	26	0	\n	258
1	47	pre-color	8	14	85	lab-vel	2	26	19	[de-aspir]	3
1	48	[ac-split]	2	14	86	io-flip	51	26	19	fusion	34
1	49	diph	68	14	33	round	16	26	34	fusion	3
1	19	[u-o]	1	14	87	color	2	26	19	procope-a	1
1	50	aspir-r	14	14	35	sharp	3	26	76	procope-b	2
1	51	[procope-j]	1	14	19	unglide	6	26	34	glide	2
1	19	[schwa-i]	2	14	19	rime	3	26	19	unglide	1
1	52	[back-asm]	1	14	88	affric	1	26	19	[a-front]	1
1	53	lab-dnt	4	14	56	back-asm	4	26	90	back-asm	1
1	54	geng-zeng	3	14	40	raise	13	26	19	acute	2
2	53	t-split	4	14	19	acute	1	26	117	[aspir]	1
3	26	length	3	14	89	y-split	1	26	112	[a-i]	1
3	25	ac-split	7	14	19	brk-a	8	26	19	[!ac-split]	6
3	14	devoi	17	14	49	diph	7	26	19	[s-ts]	2
3	37	aspir	11	14	62	[procope-x]	2	26	19	final-dism	3
3	55	lab-dnt'	2	14	139	[x-kh]	1	26	19	[ts-s]	3
3	56	lab-dism	4	15	90	devoi	45	26	118	round-adj	1
3	57	fricat	2	15	91	aspir	1	26	155	procope-m	1
3	58	retrof	2	15	92	glide	1	26	19	[i-del]	3
3	34	x-weak	2	15	53	io-flip	1	26	19	[kw-w]	2
3	1	lab-vel	1	15	93	round	2	26	19	[m-n]	1
3	34	glide	2	15	94	affric	3	26	19	degemin	1
3	59	io-flip	15	15	40	raise	2	26	19	[n-m]	1
3	53	round	8	16	69	devoi	173	27	65	glide	7
3	60	pal-dism	1	16	95	t2-merge	2	27	19	round	1
3	19	color	5	16	96	lab-dnt'	3	27	64	glide-ins	8
3	36	unglide	1	16	31	deret	1	27	101	diph	3
3	19	rime	1	16	97	fricat	4	27	19	[!ac-split]	1
3	61	affric	1	16	69	x-weak	1	27	19	[j-ins]	1
3	62	procope-d	2	16	19	round	1	28	0	\n	2
3	19	[a-front]	1	16	97	affric	1	28	26	length	19
3	63	back-asm	4	16	40	raise	2	28	19	io-flip	9
3	40	raise	8	17	0	\n	4	28	26	round	7

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
28	159	back-asm	2	35	19	unglide	11	56	19	io-flip	3
28	85	[aspir]	1	35	76	procope-d	1	56	34	round	16
28	19	brk-a	8	36	0	\n	28	56	163	back-asm	2
28	19	final-dism	3	36	19	[de-aspir]	1	56	19	brk-a	4
28	56	f-coal	17	36	19	deret	1	57	21	retrof	4
29	73	glide	2	36	19	round	1	58	141	deret	10
30	0	\n	9	36	19	raise	3	59	115	pal-dism	1
30	19	length	1	36	113	[round]	1	59	19	unglide	55
30	114	io-flip	8	36	19	[x-tsh]	2	59	125	unglide	1
30	19	round	1	36	19	[brk-a]	4	59	53	rime	8
30	19	sharp	2	36	19	[j-h]	2	60	0	\n	31
30	19	brk-a	3	37	25	devoi	36	60	19	[aspir]	7
30	101	diph	7	37	125	aspir	12	60	19	[x-kh]	1
30	85	[procope-j]	10	37	140	fricat	1	61	94	aspir	1
30	85	[procope-x]	2	37	107	retrof	4	62	19	glide	3
30	113	[round]	1	38	19	glide	5	62	34	glide	2
31	0	\n	12	38	34	glide-ins	6	62	19	glide-ins	2
31	19	length	46	39	126	round	48	62	19	procope-m	4
31	131	lab-dism	2	40	0	\n	9	63	47	round	80
31	81	retrof	1	40	122	procope-a	1	64	0	\n	11
31	19	fusion	3	40	122	procope-b	13	64	63	back-asm	16
31	76	procope-a	1	40	19	unglide	16	64	19	round-adj	3
31	34	glide	2	40	19	brk-b	57	65	0	\n	6
31	19	io-flip	12	40	100	[aspir]	1	65	102	round	1
31	98	io-flip	1	40	127	diph	1	65	19	brk-a	6
31	19	round	21	40	128	y-fuse	49	65	19	[s-ts]	3
31	141	pal-dism	1	40	19	nasal-syl	7	66	59	io-flip	31
31	19	[a-front]	2	41	0	\n	10	67	143	length	1
31	159	back-asm	5	41	104	length	1	68	1	devoi	1
31	63	back-asm	2	41	34	length	3	69	0	\n	10
31	18	back-asm	1	41	86	io-flip	49	69	26	length	15
31	100	raise	2	41	111	io-flip	3	69	149	devoi	1
31	19	brk-b	1	41	19	unglide	11	69	18	lab-dism	6
31	101	diph	12	41	63	back-asm	27	69	31	deret	5
32	0	\n	20	41	129	raise	24	69	123	r-unglide	2
32	76	procope-a	12	41	19	acute	1	69	22	fricat	23
32	76	procope-b	5	41	66	diph	7	69	165	retrof	2
32	115	pal-dism	34	42	85	lab-vel	1	69	144	spirant	2
32	60	unglide	36	42	100	raise	2	69	56	lab-vel	2
32	116	rime	249	42	19	[s-ts]	9	69	31	lab-vel	1
32	62	procope-d	2	43	130	f-coal	9	69	86	io-flip	55
32	19	[aspir]	2	44	26	glide	18	69	33	round	20
32	76	[procope-x]	1	44	64	glide-ins	4	69	19	pal-dism	1
32	19	[s-ts]	1	45	0	\n	1	69	87	color	1
33	0	\n	169	45	69	x-weak	1	69	35	sharp	2
33	19	[de-aspir]	1	45	131	lab-vel	8	69	34	unglide	4
33	26	procope-a	9	45	101	glide	2	69	34	rime	4
33	117	procope-b	11	45	134	io-flip	1	69	63	back-asm	2
33	103	unglide	13	45	132	io-flip	5	69	40	raise	11
33	119	raise	1	45	98	x-lab	1	69	122	procope-c	1
33	42	[aspir]	3	46	0	\n	45	69	166	y-split	2
33	112	x-lab	1	46	19	[e-u]	1	69	19	brk-a	10
33	118	round-adj	41	46	19	[m-w]	1	69	47	pre-color	1
33	19	[t-f]	2	47	0	\n	5	69	167	diph	5
34	0	\n	121	47	19	color	84	69	53	aspir-r	1
34	19	length	2	47	19	unglide	3	69	168	geng-zeng	2
34	34	length	2	47	19	round-adj	1	70	145	length	1
34	185	lab-dism	2	48	133	procope-a	1	70	21	devoi	3
34	19	[de-aspir]	3	48	19	[u-o]	1	70	25	devoi	1
34	19	fusion	11	49	134	io-flip	76	70	146	r-unglide	1
34	59	fusion	1	50	135	lab-dnt'	10	70	30	glide	1
34	122	procope-a	5	50	36	lab-dism	1	71	150	round	1
34	62	procope-a	3	50	59	raise	3	72	151	retrof	1
34	76	procope-b	1	51	1	glide	1	73	157	length	1
34	19	io-flip	3	52	60	round	1	73	19	length	1
34	104	pal-dism	1	53	0	\n	24	73	34	lab-dnt'	1
34	19	color	3	53	21	aspir	1	73	152	r-unglide	1
34	19	unglide	1	53	17	lab-dnt'	3	73	80	fricat	4
34	120	unglide	3	53	40	fusion	3	73	153	retrof	1
34	122	procope-d	10	53	19	fusion	1	73	25	x-weak	4
34	159	back-asm	4	53	76	procope-b	6	73	154	lab-vel	2
34	19	acute	10	53	30	glide	1	73	111	io-flip	1
34	19	glide-ins	1	53	111	io-flip	1	73	19	[a-front]	4
34	87	[aspir]	1	53	19	unglide	4	74	73	devoi	4
34	122	procope-c	2	53	19	rime	1	74	37	aspir	2
34	101	diph	1	53	129	raise	1	75	16	devoi	1
34	121	[procope-j]	1	53	131	diph	2	76	161	glide	1
34	19	round-adj	6	53	142	[procope-x]	2	76	19	glide	49
34	19	[n-ng]	1	53	118	round-adj	3	77	0	\n	5
34	19	[p-t]	1	53	19	[t-n]	1	77	28	lab-dism	17
34	19	[f-p]	1	53	19	[k-l]	1	78	34	length	8
34	169	[y-t]	1	53	19	[k-ins]	1	78	18	back-asm	1
34	87	[m-n]	1	54	136	io-flip	1	79	156	deret	1
35	123	r-unglide	1	54	137	rime	1	80	81	retrof	10
35	35	fricat	1	54	138	x-lab	1	81	25	deret	23
35	124	retrof	12	55	31	lab-dism	2	82	19	round	1
35	104	pal-dism	11	56	19	length	2	82	34	rime	14

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
83	125	deret	3	112	26	glide-ins	3
84	34	deret	2	112	19	[aspir]	1
84	158	deret	45	112	122	procopce-c	1
85	19	brk-a	19	112	19	[!ac-split]	18
85	159	f-coal	1	113	87	round-adj	5
86	0	\n	18	114	19	procopce-d	6
86	19	length	2	114	19	[n-del]	1
86	115	pal-dism	14	114	19	[ng-n]	1
86	19	unglide	23	115	19	brk-d	52
86	26	rime	119	115	115	[!-t]	1
87	0	\n	10	116	0	\n	122
87	19	acute	48	116	112	fusion	79
88	160	retrof	2	116	122	procopce-a	16
89	19	lab-vel	1	116	122	procopce-b	6
90	0	\n	2	116	172	procopce-d	8
90	161	length	1	116	26	[aspir]	3
90	34	lab-dnt'	2	116	122	procopce-c	11
90	101	lab-dism	1	116	19	[!ac-split]	1
90	162	fricat	2	116	19	final-dism	1
90	81	retrof	5	116	19	[t-p]	1
90	34	x-weak	3	116	19	[n-l]	1
90	85	lab-vel	1	117	64	glide	8
90	86	io-flip	7	117	33	[j-ins]	2
90	53	round	14	117	173	round-adj	6
90	19	unglide	1	118	0	\n	2
90	40	raise	5	118	87	degemin	43
90	19	brk-a	1	119	60	degemin	1
90	47	pre-color	1	120	113	[round]	3
91	163	devoi	1	121	85	glide	1
92	164	round	1	122	0	\n	1
93	62	procopce-b	2	122	19	glide-ins	74
94	64	devoi	4	123	42	deret	1
95	123	devoi	2	123	19	deret	8
96	101	lab-dism	3	124	35	deret	4
97	141	devoi	5	124	59	deret	8
98	0	\n	1	125	0	\n	7
98	46	length	6	125	3	devoi	8
98	26	io-flip	9	125	19	unglide	1
98	159	back-asm	11	125	159	back-asm	1
99	19	[fh-ph]	1	125	19	[!ac-split]	1
100	19	brk-b	10	125	19	[ts-s]	8
101	19	io-flip	39	126	0	\n	9
102	0	\n	9	126	34	color	33
102	19	[de-aspir]	1	126	132	unglide	4
102	102	[!ac-split]	1	126	19	[!ac-split]	1
102	19	round-adj	37	126	19	round-adj	1
103	0	\n	51	127	175	io-flip	1
103	26	fusion	15	128	0	\n	3
103	19	acute	2	128	122	procopce-d	2
103	87	round-adj	2	128	122	procopce-c	7
104	19	fusion	22	128	19	brk-c	37
105	58	retrof	8	129	19	y-fuse	40
106	57	devoi	2	130	117	round	3
107	69	deret	1	130	19	raise	6
107	41	deret	90	131	19	io-flip	5
108	0	\n	22	131	34	io-flip	2
108	19	length	2	131	19	round	1
108	170	io-flip	41	131	101	f-coal	4
108	117	round	1	132	0	\n	2
108	34	pal-dism	2	132	76	[procopce-j]	8
108	19	color	4	133	176	glide-ins	1
108	33	unglide	15	134	0	\n	35
108	26	rime	10	134	76	procopce-a	5
108	63	back-asm	11	134	19	unglide	32
108	129	raise	13	134	76	procopce-d	2
108	19	acute	3	134	76	[procopce-x]	3
108	42	[aspir]	3	135	0	\n	4
108	66	diph	13	135	34	lab-dism	6
109	0	\n	4	136	132	rime	1
109	26	length	7	137	177	[procopce-j]	1
109	86	io-flip	14	138	178	io-flip	1
109	26	round	7	139	112	[de-aspir]	1
109	19	unglide	2	140	87	devoi	1
109	171	back-asm	3	141	0	\n	1
109	19	brk-a	4	141	104	length	5
109	101	diph	1	141	123	retrof	4
109	19	[!ac-split]	1	141	59	io-flip	2
110	31	f-coal	7	141	19	rime	2
111	0	\n	1	141	19	acute	2
111	19	unglide	4	142	102	glide	2
111	104	rime	1	143	19	[p-i]	1
111	19	rime	28	144	85	x-weak	2
111	26	rime	6	145	19	[ng-del]	1
112	0	\n	50	146	179	deret	1
112	172	procopce-a	7	147	31	devoi	2
112	38	procopce-b	2	148	88	devoi	1
112	19	unglide	1	149	19	lab-dnt'	1

Src	Dst	Sym	Freq
150	19	[k-del]	1
151	181	deret	1
152	98	deret	1
153	182	retrof	1
154	18	f-coal	2
155	112	[p-t]	1
156	42	length	1
157	19	x-weak	1
158	0	\n	3
158	183	io-flip	16
158	125	color	5
158	26	unglide	3
158	19	rime	3
158	184	back-asm	7
158	129	raise	1
158	19	acute	3
158	66	diph	4
159	19	round	25
160	111	deret	2
161	19	[i-del]	2
162	186	retrof	2
163	59	round	3
164	102	procopce-b	1
165	87	deret	2
166	101	lab-vel	4
167	187	io-flip	5
168	59	rime	2
169	19	[t3-t4]	1
170	115	pal-dism	2
170	19	unglide	16
170	103	rime	23
171	125	round	3
172	19	glide	1
172	26	glide-ins	14
173	19	degemin	6
174	53	deret	4
175	19	[b-f]	1
176	111	pre-color	1
177	19	[glide]	1
178	188	unglide	1
179	19	[round]	1
180	189	unglide	2
181	190	io-flip	1
182	64	deret	1
183	115	pal-dism	1
183	26	unglide	7
183	26	rime	8
184	191	round	7
185	28	length	2
186	192	deret	2
187	65	unglide	5
188	34	[x-kh]	1
189	102	[x-kh]	2
190	193	unglide	1
191	26	color	7
192	194	io-flip	2
193	19	[!fricat]	1
194	125	rime	2

MML = 32117.49 bits

B.3 PFSA for SMC>MB (Diachronic only)

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
0	0	\n	117	1	28	diph-i	18	5	5	[de-aspir]	2
0	1	t1-split	912	1	5	[u-o]	2	6	65	t1-split	164
0	2	t4-tripart	367	1	40	diph-u	25	6	66	t4-tripart	2
0	3	devoi	147	1	5	[procope-j]	4	6	29	devoi	2
0	4	t2-merge	91	1	5	y-spl-a	1	6	67	affric	1
0	5	[i-del]	3	1	5	y-spl-b	9	7	0	\n	1
0	6	aspir	169	1	5	[i-a]	1	7	68	med_dism-b	28
0	5	[t1-split]	2	1	5	[u-ao]	1	7	15	degemin	16
0	5	[t2]	1	1	41	[i-ins]	4	7	88	vocal	1
0	7	lab-dnt'	39	1	41	[k-ins]	1	7	69	raise-u	15
0	8	med_dism-b	13	1	5	[schwa-u]	1	7	5	diph-u	2
0	5	a-merge	113	1	5	[i-r]	4	7	32	aspir-r	3
0	2	[t4-tripart]	1	1	42	palat-d	1	8	0	\n	1
0	5	[n'-del]	1	2	0	\n	2	8	5	a-merge	10
0	5	deret	1	2	44	devoi	73	8	15	palat	1
0	9	i-del	18	2	45	lab-dnt'	9	8	16	round	5
0	10	fric	44	2	44	med_dism-b	5	8	70	syllab	1
0	5	retrof-l	1	2	46	a-merge	11	8	5	[!rime]	2
0	11	affric	64	2	47	deret	1	9	0	\n	52
0	5	[t3-tla]	2	2	48	i-del	6	9	32	deret	1
0	5	[u-del]	2	2	42	fric	17	9	5	rime	2
0	12	sharp	35	2	49	sharp	12	9	72	u-ins-a	5
0	5	palat	50	2	5	palat	1	9	5	raise-u	8
0	13	fusion	35	2	50	procope	16	9	5	diph-u	11
0	14	procope	63	2	50	degemin	1	9	110	[aspir]	1
0	5	[au-ua]	1	2	46	round	6	9	5	syllab	31
0	15	degemin	20	2	51	vocal	50	10	0	\n	1
0	16	round	32	2	52	rime	43	10	63	med_dism-b	1
0	17	vocal	41	2	53	raise-il	29	10	5	retrof-l	4
0	18	coal	6	2	54	med_dism-a	2	10	32	retrof-l	1
0	19	rime	28	2	55	y-spl-d	1	10	62	fusion	3
0	20	raise-il	52	2	42	[u-n]	1	10	13	fusion	3
0	21	med_dism-a	50	2	50	y-spl-c	1	10	89	vocal	8
0	22	u-del	3	2	43	apocope	66	10	62	raise-il	1
0	22	[t2-tla]	1	2	46	[round]	4	10	74	med_dism-a	87
0	5	u-ins-a	2	2	50	[y-spl-d]	1	10	62	raise-u	2
0	23	y-spl-d	1	2	56	[vocal]	1	10	75	diph-i	7
0	5	lab-abs	2	2	5	lab-dnt	1	10	73	apocope	4
0	5	[u-n]	1	2	5	[ek-ai]	1	11	62	t1-split	1
0	24	apical	11	2	5	[k-i]	3	11	76	t4-tripart	14
0	5	[t3-tlb]	1	2	5	[k-u]	4	11	58	devoi	14
0	5	[p-m]	1	3	0	\n	26	11	77	t2-merge	11
0	25	raise-u	71	3	5	[i-del]	3	11	78	aspir	24
0	26	[t3-t2]	2	3	57	lab-dnt'	17	12	5	palat	35
0	27	y-spl-c	2	3	8	med_dism-b	3	12	41	fusion	7
0	5	[a-ins]	2	3	5	a-merge	31	12	14	procope	8
0	28	diph-i	17	3	32	i-del	3	12	79	vocal	3
0	5	[u-au]	2	3	58	fric	16	13	0	\n	20
0	5	[u-o]	4	3	12	sharp	12	13	5	[t1-split]	1
0	5	[t-n]	1	3	5	palat	15	13	5	a-merge	32
0	29	diph-u	40	3	41	fusion	1	13	72	i-del	3
0	5	[l-n]	2	3	8	fusion	3	13	5	retrof-l	2
0	5	[u-i]	1	3	16	round	16	13	5	palat	17
0	5	[procope-j]	2	3	79	vocal	2	13	63	med_dism-a	12
0	5	chamel	2	3	59	vocal	7	13	5	[k-l]	1
0	5	y-spl-a	1	3	18	coal	2	14	0	\n	44
0	15	[x-k']	1	3	60	rime	9	14	5	[!t2]	1
0	5	y-spl-b	8	3	80	raise-il	1	14	5	a-merge	19
1	0	\n	121	3	20	raise-il	12	14	5	fusion	15
1	22	devoi	102	3	63	med_dism-a	3	14	34	round	5
1	30	aspir	8	3	36	med_dism-a	1	14	5	round	2
1	7	lab-dnt'	27	3	41	y-spl-d	1	14	79	vocal	9
1	31	med_dism-b	16	3	39	apical	6	14	26	rime	16
1	5	a-merge	82	3	5	[t3-tlb]	1	14	80	raise-il	1
1	32	i-del	11	3	61	raise-u	22	14	5	lab-abs	1
1	10	fric	35	3	5	diph-i	5	14	81	raise-u	16
1	5	[u-del]	1	3	5	[u-o]	1	14	5	diph-i	6
1	33	sharp	20	3	40	diph-u	9	14	5	diph-u	9
1	5	palat	50	3	5	y-spl-b	4	14	5	[i-y]	3
1	13	fusion	36	3	5	[!affric]	2	14	5	[schwa-ins]	2
1	14	procope	67	4	3	devoi	87	14	5	[x-del]	1
1	5	[au-ua]	1	4	62	fric	1	15	0	\n	8
1	15	degemin	12	4	63	med_dism-a	1	15	5	[!t2]	4
1	34	round	27	4	64	diph-i	2	15	90	fusion	1
1	17	vocal	45	5	0	\n	1603	15	59	vocal	1
1	18	coal	3	5	5	[t1-split]	6	15	5	rime	8
1	35	rime	47	5	5	[!t2]	5	15	20	raise-il	1
1	36	raise-il	19	5	5	a-merge	1	15	81	raise-u	3
1	37	med_dism-a	46	5	5	[t3-tla]	1	15	5	diph-i	15
1	14	u-del	5	5	5	[t3-t2]	2	15	5	diph-u	7
1	5	u-ins-a	2	5	71	apocope	1	15	5	y-spl-a	4
1	27	y-spl-d	1	5	5	[aspir]	1	16	0	\n	41
1	38	lab-abs	5	5	5	[t1-t2]	1	16	28	med_dism-b	1
1	39	apical	11	5	5	[t2-merge]	1	16	5	med_dism-b	7
1	25	raise-u	36	5	5	[i-y]	3	16	5	[u-del]	1
1	27	y-spl-c	3	5	5	[procope-v]	2	16	5	procope	1
1	5	[a-ins]	1	5	132	[fric]	1	16	71	apocope	2

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
16	5	[t2-merge]	1	33	14	procope	5	50	47	rime	9
16	29	u-ins-b	2	33	5	apocope	2	50	47	raise-i1	4
17	0	\n	1	33	5	[procope-γ]	2	50	47	raise-u	1
17	82	fusion	91	34	0	\n	18	50	81	apocope	6
18	41	sharp	12	34	26	med_dism-b	5	50	5	[k-u]	1
19	0	\n	57	34	5	[u-del]	2	50	5	[u-ins-b]	2
19	5	[t2]	1	34	5	procope	4	51	96	fusion	44
19	5	palat	47	34	5	[aspir]	1	51	97	coal	7
19	60	fusion	6	34	5	u-ins-b	19	52	98	i-del	3
19	111	med_dism-a	18	34	5	[u-ins-b]	2	52	47	y-spl-d	1
19	14	[ɣ-n]	1	35	0	\n	31	52	19	apocope	67
19	43	syllab	1	35	5	palat	22	53	99	med_dism-a	3
19	5	[f-p]	2	35	41	fusion	9	53	60	apocope	47
19	5	[diph-i]	1	35	5	[ɣ-n]	3	53	100	raise-i2	1
19	73	[palat]	1	35	5	[aspir]	2	54	93	retrof-2	7
20	0	\n	3	36	5	palat	3	55	49	raise-i1	1
20	5	palat	6	36	86	med_dism-a	1	56	118	fusion	9
20	63	med_dism-a	5	36	63	med_dism-a	2	56	49	coal	2
20	5	y-spl-d	2	36	91	raise-i2	68	57	0	\n	5
20	28	y-spl-d	4	36	5	[t'-t]	2	57	9	med_dism-b	17
20	71	apocope	3	37	9	retrof-2	66	57	64	fusion	4
20	83	raise-i2	41	38	5	palat	4	57	5	raise-u	4
20	5	[y-ru]	1	38	35	lab-abs	4	58	39	i-del	3
21	84	retrof-2	50	38	5	[j-š]	2	58	32	retrof-1	1
22	0	\n	8	39	39	syllab	1	58	62	fusion	2
22	57	lab-dnt'	13	39	5	syllab	53	58	89	vocal	2
22	26	a-merge	12	40	0	\n	25	58	62	raise-i1	1
22	73	fric	5	40	5	[ɣ-n]	1	58	37	med_dism-a	20
22	12	sharp	6	40	41	[chamel]	13	58	63	med_dism-a	6
22	5	palat	7	41	5	palat	86	58	62	raise-u	5
22	13	fusion	3	41	5	[procope-v]	1	59	42	fusion	4
22	8	procope	1	42	93	retrof-1	5	59	101	fusion	21
22	5	round	5	42	59	vocal	7	59	50	coal	4
22	59	vocal	9	42	94	rime	10	60	0	\n	83
22	18	coal	1	42	95	raise-i1	6	60	5	[u-del]	1
22	60	rime	7	42	54	med_dism-a	4	60	5	palat	91
22	36	raise-i1	8	42	62	apocope	7	60	61	palat	2
22	86	med_dism-a	2	42	73	apocope	1	60	112	fusion	1
22	31	u-del	3	42	63	apocope	2	60	86	med_dism-a	1
22	61	raise-u	8	42	43	apocope	6	60	41	u-del	1
22	16	diph-i	3	42	50	retrof-2	1	60	5	[n-l]	1
22	5	diph-u	3	43	0	\n	44	60	5	[kh-s']	1
22	5	[k-u]	1	43	5	[i-del]	8	61	0	\n	29
22	41	[u-y]	1	43	5	palat	2	61	63	med_dism-a	1
23	15	rime	1	43	34	round	17	61	41	lab-abs	13
24	29	syllab	11	43	74	med_dism-a	12	61	5	[t3-t2]	3
25	0	\n	48	43	60	lab-abs	8	62	124	retrof-1	1
25	5	[i-del]	3	43	5	diph-u	1	62	63	med_dism-a	41
25	5	[t2]	4	43	26	[round]	3	62	5	retrof-2	2
25	63	med_dism-a	12	43	5	[!affric]	3	63	29	retrof-2	2
25	5	[t2-t1a]	1	43	5	[aspir]	1	63	5	retrof-2	94
25	60	lab-abs	37	43	92	u-ins-b	2	64	5	med_dism-b	9
25	5	[aspir]	1	43	28	[y-k]	1	65	102	devoi	155
25	5	[diph-u]	1	44	46	lab-dnt'	6	65	5	med_dism-b	1
26	0	\n	46	44	47	a-merge	2	65	62	fric	2
26	5	fusion	3	44	44	i-del	1	65	17	vocal	6
26	15	procope	1	44	42	fric	11	66	46	devoi	2
26	5	[k-u]	1	44	49	sharp	3	67	103	t1-split	1
26	113	[aspir]	1	44	5	palat	1	68	0	\n	15
26	5	[a-del]	1	44	47	round	7	68	68	[t3-t2]	1
26	5	[raise-u]	1	44	56	vocal	10	68	68	syllab	7
27	49	rime	2	44	53	rime	12	68	61	syllab	2
27	5	rime	13	44	53	raise-i1	10	68	5	[aspir-r]	15
28	0	\n	32	44	54	med_dism-a	1	69	0	\n	10
28	5	[t2]	1	44	5	apocope	2	69	5	degemin	5
28	68	med_dism-b	2	44	43	apocope	18	70	19	[aspir-r]	1
28	5	retrof-1	5	44	5	[k-i]	1	71	0	\n	1
28	5	[u-del]	4	45	46	med_dism-b	4	71	5	[t4-tripart]	5
28	5	[aspir]	7	45	50	degemin	2	72	0	\n	5
28	28	[chamel]	1	45	5	apocope	3	72	5	deret	2
29	0	\n	46	46	0	\n	3	72	5	[deret]	3
29	5	[t1-split]	2	46	47	med_dism-b	3	73	111	med_dism-a	17
29	5	[t3-t1a]	2	46	47	procope	1	73	62	raise-u	5
29	5	[t2-t1a]	1	46	47	degemin	1	74	63	rime	3
29	110	[aspir]	1	46	47	rime	1	74	104	retrof-2	96
29	5	[t1-t2]	5	46	5	apocope	25	75	5	retrof-1	7
29	85	[chamel]	11	46	5	[k-u]	2	76	42	devoi	14
30	87	devoi	8	46	47	[chamel]	1	77	58	devoi	10
31	0	\n	3	46	5	[o-ins]	1	77	62	raise-i1	1
31	5	a-merge	5	47	5	apocope	50	78	105	t1-split	21
31	5	round	12	48	46	deret	5	78	106	devoi	3
31	5	syllab	3	48	51	vocal	1	79	107	fusion	14
32	0	\n	2	49	5	palat	4	80	100	raise-i2	32
32	5	u-ins-a	8	49	49	fusion	1	81	0	\n	12
32	5	raise-u	9	49	47	procope	3	81	5	lab-abs	13
32	5	diph-u	6	49	41	apocope	13	82	108	coal	48
33	43	palat	11	50	59	vocal	5	82	36	raise-i1	43

Src	Dst	Sym	Freq
83	109	degemin	41
84	0	\n	23
84	5	rime	2
84	5	[ŋ-n]	1
84	5	diph-u	8
84	26	syllab	16
85	29	palat	11
86	113	retrof-2	4
87	34	round	2
87	114	apical	6
88	115	fusion	1
89	116	fusion	10
90	5	[ya-ru]	1
91	117	degemin	68
92	5	[round]	10
93	129	vocal	2
93	119	apocope	8
93	5	[k-u]	2
94	73	apocope	10
95	99	med_dism-a	6
96	120	coal	38
96	5	apocope	6
97	121	sharp	2
97	38	apocope	5
98	122	syllab	3
99	47	retrof-2	9
100	123	degemin	33
101	125	coal	15
101	128	raise-il	3
101	36	raise-il	3
102	0	\n	31
102	31	med_dism-b	4
102	5	a-merge	22
102	32	i-del	3
102	10	fric	22
102	5	palat	17
102	13	fusion	11
102	64	round	3
102	126	vocal	7
102	35	rime	16
102	41	lab-abs	1
102	61	raise-u	10
102	5	diph-i	1
102	5	[u-o]	2
102	40	diph-u	5
102	5	y-spl-a	1
102	5	y-spl-b	1
103	73	devoi	1
104	0	\n	62
104	5	[t1-split]	1
104	5	rime	5
104	5	raise-u	3
104	5	diph-u	13
104	9	syllab	10
104	5	[n'-n]	2
105	10	devoi	21
106	62	t1-split	3
107	80	raise-il	12
107	128	raise-il	4
107	36	raise-il	2
107	141	[chamel]	1
108	0	\n	18
108	5	[!t2]	1
108	5	i-del	1
108	80	raise-il	18
108	63	med_dism-a	7
108	27	[chamel]	3
109	127	med_dism-b	2
109	19	degemin	39
110	0	\n	3
110	5	[fric]	5
111	32	retrof-2	2
111	39	retrof-2	33
112	14	palat	1
113	113	syllab	2
113	5	[affric]	5
114	110	syllab	6
115	5	coal	1
116	62	coal	10
117	60	degemin	68
118	130	coal	8
118	5	apocope	1
119	92	u-ins-b	8
120	122	[chamel]	38
121	61	apocope	2
122	47	deret	5
122	47	procope	1

Src	Dst	Sym	Freq
122	52	rime	28
122	28	apocope	7
123	5	degemin	33
124	5	[u-ins-a]	1
125	0	\n	1
125	131	raise-il	3
125	5	u-del	2
125	42	[chamel]	1
125	50	[chamel]	2
125	27	[chamel]	6
126	133	fusion	7
127	134	degemin	2
128	135	raise-i2	9
129	136	fusion	2
130	102	apocope	2
130	44	[chamel]	6
131	137	raise-i2	3
132	5	[raise-il]	1
133	107	coal	5
133	128	raise-il	2
134	68	[diph-i]	2
135	138	degemin	9
136	139	coal	2
137	140	degemin	3
138	41	degemin	9
139	72	apocope	2
140	28	degemin	3
141	41	rime	1

MML = 30366.55 bits

B.4 PFSA for SMC>MB (Diachronic and Allophonic)

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
0	0	\n	117	1	5	[a-ins]	1	5	5	[t2-merge]	1
0	1	t1-split	912	1	25	diph-i	18	5	5	[i-y]	5
0	2	t4-tripart	367	1	5	[u-o]	2	6	59	t1-split	164
0	3	devoi	147	1	25	diph-u	25	6	60	t4-tripart	2
0	4	t2-merge	91	1	5	[procopce-j]	4	6	61	devoi	2
0	5	[i-del]	3	1	5	y-spl-a	1	6	62	affric	1
0	6	aspir	169	1	5	y-spl-b	9	7	0	\n	1
0	5	[t1-split]	2	1	5	[i-a]	1	7	79	med_dism-b	15
0	5	[t2]	1	1	5	[u-ao]	1	7	10	med_dism-b	13
0	7	lab-dnt'	39	1	36	[i-ins]	4	7	17	degemin	16
0	8	med_dism-b	13	1	36	[k-ins]	1	7	80	vocal	1
0	9	a-merge	113	1	5	[schwa-u]	1	7	13	raise-u	15
0	2	[t4-tripart]	1	1	5	[i-r]	4	7	25	diph-u	2
0	5	[n'-del]	1	1	3	palat-d	1	7	53	aspir-r	3
0	5	deret	1	2	0	\n	2	8	5	a-merge	10
0	10	i-del	18	2	38	devoi	73	8	5	chamel	4
0	11	fric	44	2	39	lab-dnt'	9	8	25	round	20
0	5	retrof-l	1	2	40	med_dism-b	5	8	85	vocal	1
0	12	affric	64	2	41	a-merge	11	8	25	syllab	4
0	5	[t3-t1a]	2	2	41	deret	1	8	25	[rime]	1
0	13	[u-del]	2	2	42	i-del	6	8	5	[x-del]	1
0	14	sharp	35	2	43	fric	17	9	0	\n	165
0	5	palat	50	2	44	sharp	12	9	5	[i-del]	1
0	15	fusion	35	2	5	palat	1	9	5	[t1-split]	2
0	16	procopce	63	2	45	procopce	16	9	5	[t3-t1a]	2
0	5	[au-ua]	1	2	45	degemin	1	9	5	[t2-t1a]	1
0	17	degemin	20	2	46	round	6	9	5	[t3-t2]	1
0	5	chamel	2	2	47	vocal	50	9	63	apocope	1
0	18	round	32	2	48	rime	43	9	112	[aspir]	2
0	19	vocal	41	2	49	raise-il	29	9	5	[aspir]	1
0	20	coal	6	2	50	med_dism-a	2	9	5	[t1-t2]	1
0	21	rime	28	2	51	y-spl-d	1	9	142	[fric]	1
0	22	raise-il	52	2	45	[g-n]	1	9	21	[y-k]	1
0	23	med_dism-a	50	2	45	y-spl-c	1	10	0	\n	70
0	3	u-del	3	2	37	apocope	66	10	25	[t1-split]	1
0	3	[t2-t1a]	1	2	13	[round]	4	10	53	deret	1
0	5	u-ins-a	2	2	45	[y-spl-d]	1	10	5	chamel	27
0	24	y-spl-d	1	2	52	[vocal]	1	10	5	rime	7
0	5	lab-abs	2	2	5	lab-dnt	1	10	5	u-ins-a	6
0	25	[g-n]	1	2	5	[ek-ai]	1	10	13	u-ins-a	5
0	26	apical	11	2	25	[k-i]	3	10	5	raise-u	16
0	5	[t3-t1b]	1	2	5	[k-u]	4	10	10	[t3-t2]	1
0	5	[p-m]	1	3	0	\n	31	10	25	diph-u	22
0	27	raise-u	71	3	77	t4-tripart	1	10	25	syllab	8
0	13	[t3-t2]	2	3	5	[i-del]	3	10	13	syllab	16
0	28	y-spl-c	2	3	53	lab-dnt'	30	10	5	syllab	19
0	5	[a-ins]	2	3	15	med_dism-b	4	10	112	[aspir]	1
0	25	diph-i	17	3	5	a-merge	22	10	5	[aspir-r]	3
0	5	[u-au]	2	3	13	a-merge	21	10	5	[raise-u]	1
0	5	[u-o]	4	3	10	i-del	3	10	25	[n'-n]	2
0	5	[t-n]	1	3	54	fric	21	11	0	\n	1
0	29	diph-u	40	3	14	sharp	18	11	58	med_dism-b	1
0	5	[l-n]	2	3	5	palat	22	11	35	i-del	3
0	5	[u-i]	1	3	15	fusion	6	11	5	retrof-l	3
0	5	[procopce-j]	2	3	36	fusion	1	11	54	fusion	5
0	5	y-spl-a	1	3	15	procopce	1	11	64	fusion	1
0	17	[x-k']	1	3	5	chamel	2	11	82	vocal	6
0	5	y-spl-b	8	3	25	round	21	11	64	raise-il	2
1	0	\n	118	3	55	vocal	19	11	34	med_dism-a	89
1	3	devoi	102	3	20	coal	3	11	54	raise-u	4
1	30	aspir	8	3	21	rime	12	11	65	diph-i	5
1	7	lab-dnt'	27	3	13	rime	4	11	81	apocope	1
1	8	med_dism-b	16	3	22	raise-il	20	11	54	apocope	3
1	31	a-merge	82	3	70	raise-il	1	12	64	t1-split	1
1	10	i-del	11	3	56	med_dism-a	3	12	66	t4-tripart	14
1	11	fric	35	3	58	med_dism-a	3	12	11	devoi	14
1	5	[u-del]	1	3	8	u-del	3	12	67	t2-merge	11
1	32	sharp	20	3	36	y-spl-d	1	12	68	aspir	24
1	5	palat	50	3	35	apical	6	13	0	\n	135
1	15	fusion	36	3	5	[t3-t1b]	1	13	5	[t2]	3
1	16	procopce	67	3	57	raise-u	30	13	5	palat	27
1	5	[au-ua]	1	3	25	diph-i	8	13	13	palat	2
1	17	degemin	12	3	5	[u-o]	1	13	5	fusion	5
1	5	chamel	3	3	25	diph-u	12	13	36	fusion	3
1	18	round	27	3	5	y-spl-b	4	13	17	procopce	1
1	19	vocal	45	3	5	[k-u]	1	13	5	degemin	5
1	20	coal	3	3	5	[laffric]	2	13	5	chamel	6
1	21	rime	47	3	36	[u-y]	1	13	69	vocal	1
1	33	raise-il	19	4	3	devoi	87	13	115	med_dism-a	4
1	34	med_dism-a	46	4	54	fric	1	13	5	lab-abs	12
1	8	u-del	5	4	58	med_dism-a	1	13	5	[t3-t2]	2
1	5	u-ins-a	2	4	53	diph-i	2	13	25	diph-u	2
1	28	y-spl-d	1	5	0	\n	1642	13	25	apocope	2
1	21	lab-abs	5	5	5	[t1-split]	1	13	5	syllab	5
1	35	apical	11	5	5	chamel	1	13	5	[t1-t2]	1
1	27	raise-u	36	5	5	u-ins-a	2	13	5	[fric]	3
1	28	y-spl-c	3	5	5	[k-u]	1	13	5	[n-l]	1

Src	Dst	Sym	Freq	Src	Dst	Sym	Freq	Src	Dst	Sym	Freq
13	5	[deret]	1	25	57	raise-u	1	43	50	med_dism-a	4
14	0	\n	1	25	25	[t3-t2]	1	43	54	apocope	13
14	5	palat	41	25	71	apocope	1	43	58	apocope	2
14	36	fusion	7	25	25	syllab	1	43	81	apocope	1
14	13	procope	8	25	25	[aspir]	3	43	46	retrof-2	1
14	116	vocal	3	25	25	[t1-t2]	1	44	5	palat	3
14	56	med_dism-a	1	25	36	[chamel]	13	44	44	fusion	1
14	36	apocope	1	25	25	[u-ins-b]	1	44	41	procope	3
15	0	\n	20	25	25	[aspir-r]	3	44	36	apocope	9
15	5	[t1-split]	3	25	5	[a-del]	1	45	95	vocal	6
15	5	a-merge	37	25	25	[de-aspir]	2	45	41	rime	9
15	71	i-del	3	25	25	[f-p]	1	45	88	raise-il	4
15	5	retrof-1	1	26	9	syllab	11	45	41	raise-u	1
15	5	palat	17	27	0	\n	48	45	13	apocope	6
15	17	palat	1	27	5	[i-del]	3	45	5	[k-u]	1
15	74	med_dism-a	3	27	5	[t2]	4	46	88	deret	2
15	23	med_dism-a	7	27	58	med_dism-a	12	46	88	procope	2
15	25	[!rime]	1	27	5	[t2-t1a]	1	46	5	chamel	1
15	5	[k-l]	1	27	21	lab-abs	37	46	48	rime	28
16	0	\n	39	27	5	[aspir]	1	46	25	apocope	12
16	5	a-merge	19	27	25	[diph-u]	1	46	41	[chamel]	1
16	5	fusion	5	28	14	rime	2	46	5	[u-ins-b]	2
16	57	fusion	8	28	5	rime	13	47	96	fusion	43
16	5	chamel	1	29	25	[t1-split]	1	47	97	coal	7
16	25	round	4	29	25	[t3-t1a]	1	48	99	i-del	3
16	69	vocal	7	29	5	chamel	26	48	41	y-spl-d	1
16	13	rime	16	29	25	[t1-t2]	1	48	98	apocope	67
16	70	raise-il	1	29	32	[chamel]	11	49	101	med_dism-a	3
16	5	lab-abs	1	30	78	devoi	8	49	100	apocope	35
16	13	raise-u	16	31	0	\n	80	49	102	raise-i2	1
16	25	diph-i	6	31	5	[procope-v]	3	50	91	retrof-2	7
16	25	diph-u	9	32	9	palat	22	51	97	raise-il	1
16	5	[i-y]	1	32	16	procope	5	52	122	fusion	9
16	25	[schwa-ins]	2	32	5	apocope	2	52	97	coal	2
17	0	\n	8	32	5	[procope-r]	2	53	0	\n	5
17	5	[!t2]	4	33	25	palat	2	53	25	med_dism-b	9
17	83	fusion	1	33	74	med_dism-a	2	53	10	med_dism-b	13
17	55	vocal	1	33	84	raise-i2	63	53	114	fusion	3
17	5	rime	8	33	25	[t'-t]	1	53	3	fusion	1
17	22	raise-il	1	34	10	retrof-2	135	53	5	raise-u	8
17	13	raise-u	3	35	35	syllab	1	54	5	retrof-1	2
17	25	diph-i	15	35	5	syllab	53	54	103	fusion	1
17	25	diph-u	7	36	5	palat	69	54	82	vocal	1
17	5	y-spl-a	4	37	0	\n	25	54	58	med_dism-a	27
18	0	\n	1	37	5	[i-del]	7	54	115	med_dism-a	3
18	25	med_dism-b	10	37	5	palat	2	54	86	med_dism-a	13
18	25	[u-del]	3	37	5	chamel	10	54	54	raise-u	8
18	25	procope	4	37	18	round	17	54	87	u-ins-b	1
18	5	chamel	35	37	86	med_dism-a	7	54	5	retrof-2	1
18	71	apocope	1	37	13	lab-abs	8	55	104	fusion	20
18	25	[aspir]	1	37	25	diph-u	1	56	117	retrof-2	4
18	25	[t2-merge]	1	37	10	[round]	3	56	5	[t'-t]	1
18	25	u-ins-b	21	37	5	[!affric]	3	57	0	\n	34
18	25	[u-ins-b]	1	37	25	[aspir]	1	57	5	a-merge	1
19	0	\n	1	37	87	u-ins-b	1	57	105	med_dism-a	1
19	72	fusion	91	38	39	lab-dnt'	6	57	36	lab-abs	11
20	73	sharp	12	38	41	a-merge	2	57	5	[t3-t2]	1
21	0	\n	81	38	38	i-del	1	58	5	retrof-2	49
21	5	[u-del]	1	38	43	fric	11	59	106	devoi	155
21	5	palat	92	38	44	sharp	3	59	25	med_dism-b	1
21	36	fusion	6	38	5	palat	1	59	54	fric	2
21	13	fusion	7	38	88	round	3	59	19	vocal	6
21	36	u-del	1	38	52	vocal	10	60	39	devoi	2
21	5	[ŋ-n]	2	38	89	rime	7	61	5	[t1-t2]	2
21	5	[aspir]	6	38	49	raise-il	10	62	107	t1-split	1
21	5	[j-š]	2	38	50	med_dism-a	1	63	5	[t4-tripart]	1
21	5	[kh-s']	1	38	37	apocope	18	64	74	med_dism-a	18
22	25	palat	7	38	25	[k-i]	1	64	25	retrof-2	1
22	5	chamel	3	39	45	med_dism-b	3	65	25	retrof-1	7
22	56	med_dism-a	1	39	41	med_dism-b	4	66	43	devoi	14
22	74	med_dism-a	5	39	45	degemin	2	67	11	devoi	31
22	25	y-spl-d	6	39	41	rime	1	67	64	raise-il	1
22	71	apocope	3	39	5	apocope	7	68	67	t1-split	21
22	75	raise-i2	49	40	41	round	4	68	108	devoi	3
22	25	[y-ru]	1	40	150	rime	5	69	109	fusion	8
23	76	retrof-2	73	40	25	apocope	2	70	102	raise-i2	32
24	17	rime	1	41	41	degemin	1	71	25	[t4-tripart]	4
25	0	\n	1	41	5	apocope	37	71	25	deret	2
25	25	[t1-split]	2	41	5	[k-u]	2	71	5	chamel	5
25	25	[!t2]	5	42	88	deret	8	71	25	[aspir-r]	4
25	25	med_dism-b	5	42	90	vocal	1	72	110	coal	48
25	25	retrof-1	5	42	141	apocope	1	72	33	raise-il	43
25	25	[u-del]	4	43	91	retrof-1	5	73	25	palat	18
25	25	procope	1	43	92	vocal	5	73	100	lab-abs	4
25	5	chamel	449	43	146	coal	1	74	25	retrof-2	35
25	25	lab-abs	1	43	93	rime	10	75	111	degemin	49
25	25	[ŋ-n]	1	43	94	raise-il	6	76	0	\n	25

Src	Dst	Sym	Freq
76	5	chamel	15
76	5	rime	2
76	25	[ŋ-n]	1
76	25	diph-u	14
76	9	syllab	16
77	113	devoi	1
78	18	round	2
78	10	apical	6
79	0	\n	1
79	5	chamel	1
79	71	syllab	7
79	13	[aspir-r]	6
80	118	fusion	1
81	5	retrof-1	2
81	64	fusion	1
81	82	vocal	3
81	23	med_dism-a	16
81	65	diph-i	2
82	119	fusion	10
83	5	[ya-ru]	1
84	120	degemin	63
85	121	fusion	1
86	58	rime	3
86	13	retrof-2	17
87	25	[round]	10
88	25	apocope	27
88	25	[o-ins]	1
89	14	apocope	7
90	123	fusion	1
91	136	vocal	2
91	124	apocope	8
91	5	[k-u]	2
92	43	fusion	5
93	125	apocope	10
94	101	med_dism-a	6
95	126	fusion	2
95	46	coal	4
96	127	coal	37
96	5	apocope	6
97	128	sharp	2
97	73	apocope	8
98	0	\n	26
98	5	palat	23
98	115	med_dism-a	14
98	37	syllab	1
98	5	[f-p]	1
98	25	[diph-i]	3
98	125	[palat]	1
99	42	syllab	3
100	25	palat	17
100	5	chamel	20
100	25	[ŋ-n]	2
101	88	retrof-2	9
102	129	degemin	33
103	130	retrof-1	1
104	131	coal	12
104	134	raise-il	3
104	22	raise-il	2
104	33	raise-il	3
105	15	retrof-2	1
106	0	\n	24
106	8	med_dism-b	4
106	5	a-merge	22
106	10	i-del	3
106	81	fric	22
106	9	palat	17
106	15	fusion	11
106	5	chamel	6
106	53	round	3
106	116	vocal	7
106	13	rime	16
106	36	lab-abs	1
106	57	raise-u	9
106	25	diph-i	1
106	5	[u-o]	2
106	25	diph-u	5
106	5	y-spl-a	1
106	5	y-spl-b	1
107	125	devoi	1
108	64	t1-split	3
109	70	raise-il	12
109	151	[chamel]	1
110	25	[t2]	1
110	25	i-del	1
110	5	chamel	18
110	70	raise-il	18
110	74	med_dism-a	7

Src	Dst	Sym	Freq
110	28	[chamel]	3
111	132	med_dism-b	2
111	13	degemin	47
112	5	[fric]	2
112	5	[affric]	3
113	41	fusion	1
114	5	med_dism-b	3
115	35	retrof-2	33
116	133	fusion	10
117	112	syllab	2
117	25	[affric]	2
118	25	coal	1
119	64	coal	10
120	21	degemin	63
121	135	raise-il	1
122	137	coal	8
122	5	apocope	1
123	138	coal	1
124	87	u-ins-b	8
125	115	med_dism-a	12
126	139	coal	2
127	46	[chamel]	37
128	140	apocope	2
129	5	degemin	33
130	5	[u-ins-a]	1
131	5	chamel	1
131	33	raise-il	3
131	25	u-del	2
131	28	[chamel]	6
132	98	degemin	2
133	109	coal	5
133	134	raise-il	5
134	143	raise-i2	8
135	144	raise-i2	1
136	145	fusion	2
137	25	apocope	2
137	40	[chamel]	6
138	42	[chamel]	1
139	45	[chamel]	2
140	73	lab-abs	2
141	21	[chamel]	1
142	25	[raise-il]	1
143	147	degemin	8
144	148	degemin	1
145	149	coal	2
146	43	[chamel]	1
147	36	degemin	8
148	31	degemin	1
149	152	apocope	2
150	21	apocope	5
151	36	rime	1
152	25	[deret]	2

MML = 31585.79 bits

Appendix C

From Simplified Middle Chinese to Modern Beijing

The data in this appendix and the next are given in tabular form with 9 columns each, listing for a given SMC form: its page number (col 1) and column (2) in the original *Zihui*, the corresponding Chinese character (3), its reconstructed SMC form (sans asterisk marker) in IPA (4) and tone number in historical Chinese phonology (5), its Modern Beijing or Cantonese form in IPA (6) and its tone (7), the colon separated sequence of P-rules (including exceptions) that derive the stated modern form from its protoform (8), and finally a colon separated sequence of allophonic rules (A-rules) that are used for the phonetic realisation of the form in present day usage (9). The Chinese character in column 3 is a simplified form of the more traditional character originally given in the *Zihui*.

The diachronic and allophonic rule names have been abbreviated somewhat from their original names given in Chen76 and CN84, but effort has been made to ensure that the original rules can still be readily identified from the abbreviated forms given here. Chen76 and CN84 should be consulted for an explanation of the rules themselves. A note regarding the lab-dnt', pro-m and pre-color rules is, however, warranted.

Chen76 and CN84 both describe the lab-dnt rule which labio-dentalises bilabial initials that are followed by a palatal glide and removes the palatal glide in the process. Many reconstructed forms in Chen76 and CN84, however, are already in the form of a labio-dental followed by a palatal glide. It was suggested that such forms are in an intermediate stage of operation of the labio-dentalisation rule since the modern forms are consistent with the operation of the complete rule had the initial been bilabial.¹ It was thus decided to use the name lab-dnt' to indicate the application of the later phase of the complete labio-dentalisation rule.

In the case of Modern Cantonese alone, pro-m is used to represent a rare minor rule (CN84, I, p.188) which inserts an initial [ŋ] after the application of one of the other procope rules. Also, CN84 (III, p.366) discusses an "in-

¹Personal communication with John Newman: Comment regarding the environment for the lab-dnt rule, in Discussion dt. Nov 1995

explicable rounding of **iæn/t* to Cantonese *æ̃n/tʰ* and suggests that there are reasons to think that there was a partial merger of the Middle Chinese form in the direction of **yæn/t* which subsequently fed the color rule. We thus call this process the pre-color rule here.

Some discussion of the names used for the various exceptions is also required since they are not described in either Chen76 or CN84. We now try to do so. All exceptions are enclosed within square brackets. Many of them are of the form [X-Y] representing an unexpected change of the phoneme X into phoneme Y. An exception of the form [X-del] or [X-ins] represents a change wherein the phoneme X was unexpectedly deleted or inserted respectively. When a regular rule in Chen76 or CN84 is enclosed in square brackets, as for example [aspir], this indicates that the rule had applied, but in violation of either its conditioning environment or the Relative Chronology.² An exception of the form [!A] where A is a regular rule indicates that rule A failed to apply in a context where it was expected to.

Exceptions of the form [pro-X] indicate the deletion of the initial phoneme X as if the procope rule had applied. These have been classified separately from phoneme deletion exceptions subsequent to a discussion regarding their exact categorisation with John Newman where it was pointed out that the environment in which this exception operated was mostly identical to that of the procope rule. Other than this, no attempt has been made to categorise or generalise any exceptions. For example, [X-Y] in initial positions are not grouped separately from [X-Y] in medial or final positions.

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
1	1	疤	pa	1	pa	1a	t1:	
1	1	巴	pa	1	pa	1a	t1:	
1	2	八	pat	4	pa	1a	t4:apo:	
1	3	拔	bat	4	pa	1b	t4:devoi:apo:	
1	4	把	pa	2	pa	2	:	
1	5	坝	pa	3	pa	3	:	
1	5	霸	pa	3	pa	3	:	
1	6	色	pa	3	pa	3	:	
1	7	罢	bai	2	pa	3	t2:devoi:[i-del]:	
1	8	爬	ba	1	p ^h a	1b	asp:t1:devoi:	
1	9	怕	p ^h a	3	p ^h a	3	:	
2	1	妈	ma	1	ma	1a	[t1]:	
2	2	麻	ma	1	ma	1b	t1:	
2	3	马	ma	2	ma	2	[!t2]:	
2	4	骂	ma	3	ma	3	:	
2	5	发	fyat	4	fa	1a	t4:lab-dnt':apo:med-b:	
2	6	乏	vyap	4	fa	1b	t4: devoi:lab-dnt':apo:med-b:	
2	7	罚	vyat	4	fa	1b	t4:devoi:lab-dnt':apo:med-b:	
2	8	法	fyap	4	fa	2	t4:lab-dnt':apo:med-b:	
2	9	发	fyat	4	fa	2	t4:lab-dnt':apo:med-b:	
3	1	搭	tap	4	ta	1a	t4:A-mg:apo:	
3	2	答	tap	4	ta	1b	A-mg:apo:[t4]:	
3	3	达	dat	4	ta	1b	t4:devoi:A-mg:apo:	
3	4	打	tañ	2	ta	2	[n'-del]:	
3	5	大	da	3	ta	3	devoi:A-mg:	
3	6	他	t ^h a	1	t ^h a	1a	t1:A-mg:	
3	7	塌	t ^h ap	4	t ^h a	1a	t4:A-mg:apo:	
3	8	塔	t ^h ap	4	t ^h a	2	t4:A-mg:apo:	

²No distinction was made between these two kinds of exceptions because it would be irrelevant from a computational point of view. It is sometimes possible to avoid exceptions to the conditioning environment by violating the RC and vice-versa. But then, by our naming scheme, a rule occurring in violation of either would constitute an exception. So either way there is a penalty to pay for describing this change.

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
3	9	踏	t ^h ap	4	t ^h a	3	t4:A-mg:apo:	
4	1	拿	na	1	na	1b	t4:	
4	2	纳	nap	4	na	3	t4:A-mg:apo:	
4	3	拉	lap	4	la	1a	t4:A-mg:apo:	
4	4	腊	lap	4	la	3	t4:A-mg:apo:	
4	5	辣	lat	4	la	3	t4:A-mg:apo:	
4	6	扎	tʂat	4	tʂa	1a	t4:deret:apo:	
4	7	杂	dʒap	4	tʂa	1b	t4:devoi:A-mg:apo:	
4	8	擦	ts ^h at	4	ts ^h a	1a	t4:A-mg:apo:	
4	9	洒	ʂa	2	sa	2	deret:	
5	1	渣	tʂa	1	tʂa	1a	t1:	
5	2	札	tʂap	4	tʂa	1a	t4:apo:	
5	3	闸	dʒap	4	tʂa	1b	t4:devoi:apo:[!affric]:	
5	4	炸	dʒiap	4	tʂa	3	t4:devoi:i-del:apo:[!affric]:	
5	5	札	tʂat	4	tʂa	1b	t4:apo:	
5	6	𠵹	dʒat	4	tʂa	1b	t4:devoi:apo:[!affric]:	
5	7	炸	tʂa	3	tʂa	3	:	
5	8	榨			tʂa	3	:	
5	9	榨	tʂa	3	tʂa	3	:	
6	1	乍	dʒa	3	tʂa	3	devoi:[!affric]:	
6	2	叉	tʂ ^h a	1	tʂ ^h a	1a	t1:	
6	3	差	tʂ ^h a	1	tʂ ^h a	1a	t1:	
6	4	插	tʂ ^h ap	4	tʂ ^h a	1a	t4:apo:	
6	5	茶	ʈa	1	tʂ ^h a	1b	asp:t1:devoi:fric:retrof-1:	
6	6	揀	ʈa	1	tʂ ^h a	1b	asp:t1:devoi:fric:retrof-1:	
6	7	查	dʒa	1	tʂ ^h a	1b	affric:asp:t1:devoi:	
6	8	察	tʂ ^h at	4	tʂ ^h a	1b	t4:apo:	
6	9	岔	tʂ ^h a	3	tʂ ^h a	3	:	
7	1	差	tʂ ^h a	3	tʂ ^h a	1a	[t3-t1a]:	
7	2	沙	ʂa	1	ʂa	1a	t1:	
7	3	炒	ʂa	1	ʂa	1a	t1:	
7	4	杀	ʂat	4	ʂa	1a	t4:apo:	
7	5	傻	ʂua	2	ʂa	2	[u-del]:	
7	6	加	ka	1	tʂia	1a	t1:sharp:pal:	
7	7	嘉	ka	1	tʂia	1a	t1:sharp:pal:	
7	8	家	ka	1	tʂia	1a	t1:sharp:pal:	
7	9	佳	kai	1	tʂia	1a	t1:sharp:pal:[i-del]:	
8	1	夹	kap	4	tʂia	1a	t4:sharp:apo:pal:	
8	2	夹	kap	4	tʂia	1b	t4:sharp:apo:pal:	
8	3	假	ka	2	tʂia	2	sharp:pal:	
8	4	贾	ka	2	tʂia	2	sharp:pal:	
8	5	甲	kap	4	tʂia	2	t4:sharp:apo:pal:	
8	6	假	ka	3	tʂia	3	sharp:pal:	
8	7	稼	ka	3	tʂia	3	sharp:pal:	
8	8	嫁	ka	3	tʂia	3	sharp:pal:	
8	9	架	ka	3	tʂia	3	sharp:pal:	
9	1	驾	ka	3	tʂia	3	sharp:pal:	
9	2	价	ka	3	tʂia	3	sharp:pal:	
9	3	价	k ^h ap	4	tʂ ^h ia	1a	t4:sharp:apo:pal:	
9	4	卡			tʂ ^h ia	2	:	
9	5	恰	k ^h ap	4	tʂ ^h ia	3	t4:sharp:apo:pal:	
9	6	虾	xa	1	ʂia	1a	t1:sharp:pal:	
9	7	瞎	xat	4	ʂia	1a	t4:sharp:apo:pal:	
9	8	霞	ya	1	ʂia	1b	t4:devoi:sharp:pal:	
9	9	匣	yap	4	ʂia	1b	t4:devoi:sharp:apo:pal:	
10	1	精	yat	4	ʂia	1b	t4:devoi:sharp:apo:pal:	
10	2	吓	xac	4	ʂia	3	t4:sharp:fuse:apo:pal:	
10	3	下	ya	2	ʂia	3	t2:devoi:sharp:pal:	
10	4	下	ya	3	ʂia	3	devoi:sharp:pal:	
10	5	夏	ya	3	ʂia	3	devoi:sharp:pal:	
10	6	鸦	?a	1	ia	1a	t1:sharp:pro:	
10	7	押	?ap	4	ia	1a	t4:sharp:pro:apo:	
10	7	鸭	?ap	4	ia	1a	t4:sharp:pro:apo:	
10	8	压	?ap	4	ia	1a	t4:sharp:pro:apo:	
10	9	牙	ɲa	1	ia	1b	t1:sharp:apo:	
11	1	芽	ɲa	1	ia	1b	t1:sharp:apo:	
11	2	雅	ɲa	2	ia	2	sharp:pro:[!t2]:	
11	3	哑	?a	2	ia	2	sharp:pro:	
11	4	亚	?a	3	ia	3	sharp:pro:	
11	5	抓	tʂau	1	tʂua	1a	t1:[au-ua]:	
11	6	爪	tʂau	2	tʂua	2	[au-ua]:	
11	7	刷	ʂuat	4	ʂua	1a	t4:apo:	
11	8	耍	ʂua	2	ʂua	2	:	
11	9	瓜	kua	1	kua	1a	t1:	
12	1	刚	kuat	4	kua	1a	t4:apo:	
12	2	寡	kua	2	kua	2	:	
12	3	卦	kuai	3	kua	3	[i-del]:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allomorphic
12	3	挂	kuai	3	kua	3	[i-del]:	
12	4	夸	k ^h ua	1	k ^h ua	1a	t1:	
12	5	垮	k ^h ua	2	k ^h ua	2	:	
12	6	跨	k ^h ua	3	k ^h ua	3	:	
12	7	花	xua	1	xua	1a	t1:	
12	8	华	yua	1	xua	1b	t1:devoi:	
12	9	滑	yuat	4	xua	1b	t4:devoi:apo:	
13	1	化	xua	3	xua	3	:	
13	2	画	yuai	3	xua	3	devoi:[i-del]:	
13	3	话	yuai	3	xua	3	devoi:[i-del]:	
13	4	划	yua	4	xua	3	t4:devoi:fuse:apo:	
13	5	挖	ʔuat	4	ua	1a	t4:pro:apo:	
13	6	瓦	ɣua	2	ua	2	pro:	
13	7	袜	ɣjyat	4	ua	3	t4:lab-dnt':degem:apo:med-b:	
13	8	得	tək	4	te	1b	t4:apo:	cha:
13	9	德	tək	4	te	1b	t4:apo:	cha:
14	1	特	dək	4	t ^h e	3	t4:devoi:Apo:[asp]:	cha:
14	2	乐	lak	4	le	3	t4:apo:rnd:	cha:
14	3	勒	lək	4	le	3	t4:apo:	cha:
14	4	则	tsək	4	tse	1b	t4:apo:	cha:
14	5	择	ɖac	4	tse	1b	t4:devoi:fric:retrof- 1:vocal:fuse:coal:apo:[deret]:	cha:
14	6	泽	ɖac	4	tse	1b	t4:devoi:fric:retrof- 1:vocal:fuse:coal:apo:[deret]:	cha:
14	7	责	tɕac	4	tse	1b	t4:vocal:fuse:coal:[cha]:deret:apo:	cha:
14	8	测	tɕiək	4	ts ^h e	3	t4:rime:i-del:syllab:apo:[cha]:[asp]:	
14	9	涑	tɕ ^h iək	4	ts ^h e	3	t4:rime:i-del:syllab:deret:apo:	cha:
15	1	厕	tɕ ^h iək	4	ts ^h e	3	t4:rime:i-del:syllab:deret:apo:	cha:
15	2	策	tɕ ^h iək	4	ts ^h e	3	t4:i-del:vocal:fuse:coal:[cha]:deret:apo:	cha:
15	3	册	tɕ ^h iək	4	ts ^h e	3	t4:vocal:fuse:coal:[cha]:deret:apo:	cha:
15	4	涩	ɕiap	4	se	3	t4:i-del:deret:apo:	cha:
15	5	瑟	ɕiat	4	se	3	t4:i-del:deret:apo:	cha:
15	6	亩	ɕiək	4	se	3	t4:i-del:deret:apo:	cha:
15	7	色	ɕiək	4	se	3	t4:i-del:deret:apo:	cha:
15	8	遮	tɕia	1	tɕe	1a	t1:raise-il:med-a:retrof-2:	cha:
15	9	哲	ɕiat	4	tɕe	1b	t4:fric:raise-il:apo:med-a:retrof-2:	cha:
16	1	折	tɕiat	4	tɕe	1b	t4:raise-il:apo:med-a:retrof-2:	cha:
16	2	者	tɕia	2	tɕe	2	raise-il:med-a:retrof-2:	cha:
16	3	浙	tɕiat	4	tɕe	3	t4:raise-il:apo:med-a:retrof-2:	cha:
16	4	丰	ts ^h ia	1	ts ^h e	1a	t1:raise-il:med-a:retrof-2:	cha:
16	5	扯	tɕ ^h ia	2	ts ^h e	2	raise-il:med-a:retrof-2:	cha:
16	6	御	ɕ ^h iat	4	ts ^h e	3	t4:fric:raise-il:apo:med-a:retrof-2:	cha:
16	6	撤	ɕ ^h iat	4	ts ^h e	3	t4:fric:raise-il:apo:med-a:retrof-2:	cha:
16	7	蛇	dɕia	1	ɕe	1b	t1:devoi:raise-il:med-a:retrof-2:[affric]:	cha:
16	8	舌	dɕiat	4	ɕe	1b	affric:t4:devoi:raise-il:apo:med-a:retrof-2:	cha:
16	9	舍	ɕia	2	ɕe	2	raise-il:med-a:retrof-2:	cha:
17	1	赦	ɕia	3	ɕe	3	raise-il:med-a:retrof-2:	cha:
17	2	舍	ɕia	3	ɕe	3	raise-il:med-a:retrof-2:	cha:
17	3	社	ɕia	2	ɕe	3	affric:t2:raise-il:med-a:retrof-2:	cha:
17	4	射	dɕia	3	ɕe	3	affric:devoi:raise-il:med-a:retrof-2:	cha:
17	5	涉	ɕiap	4	ɕe	3	affric:t4:devoi:raise-il:apo:med-a:retrof-2:	cha:
17	6	设	ɕiat	4	ɕe	3	t4:raise-il:apo:med-a:retrof-2:	cha:
17	7	惹	ɕia	2	re	2	fric:raise-il:med-a:retrof-2:	cha:
17	8	热	ɕiat	4	re	3	t4:fric:raise-il:apo:med-a:retrof-2:	cha:
17	9	歌	ka	1	ke	1a	t1:rnd:	cha:
18	1	鸽	kap	4	ke	1a	t4:rnd:apo:	cha:
18	2	割	kat	4	ke	1a	t4:rnd:apo:	cha:
18	3	阁	kak	4	ke	1b	t4:[rnd]:apo:	cha:
18	4	格	kac	4	ke	1b	t4:vocal:fuse:coal:[cha]:apo:	cha:
18	5	隔	kac	4	ke	1b	t4:vocal:fuse:coal:[cha]:apo:	cha:
18	6	革	kac	4	ke	1b	t4:vocal:fuse:coal:[cha]:apo:	cha:
18	7	个	ka	3	ke	3	t4:rnd:	cha:
18	8	各	kak	4	ke	3	t4:apo:[rnd]:	cha:
18	9	科	k ^h ua	1	k ^h e	1a	t1:u-del:round:	cha:
19	1	棵	k ^h ua	1	k ^h e	1a	t1:u-del:round:	cha:
19	2	颗	k ^h ua	2	k ^h e	1a	[t2-t1A]:u-del:round:	cha:
19	3	壳	k ^h ak	4	k ^h e	1b	t4:[rnd]:apo:	cha:
19	4	可	k ^h a	2	k ^h e	2	rnd:	cha:
19	5	渴	k ^h at	4	k ^h e	2	t4:apo:rnd:	cha:
19	6	课	k ^h ua	3	k ^h e	3	u-del:round:	cha:
19	7	克	k ^h ək	4	k ^h e	3	t4:apo:	cha:
19	8	刻	k ^h ək	4	k ^h e	3	t4:apo:	cha:
19	9	客	k ^h ac	4	k ^h e	3	t4:vocal:fuse:coal:[cha]:apo:	cha:
20	1	何	ɣa	1	xe	1b	t1:devoi:round:	cha:
20	2	河	ɣa	1	xe	1b	t1:devoi:round:	cha:
20	3	荷	ɣa	1	xe	1b	t1:devoi:round:	cha:
20	4	禾	ɣua	1	xe	1b	t1:devoi:u-del:round:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
20	5	和	yua	1	xe	1b	t1:devoi:u-del:round:	cha:
20	6	合	yap	4	xe	1b	t4:devoi:rnd:apo:	cha:
20	7	盒	yap	4	xe	1b	t4:devoi:rnd:apo:	cha:
20	8	核	yac	4	xe	1b	t4:devoi:vocal:fuse:coal:[cha]:apo:	cha:
20	9	贺	ya	3	xe	3	devoi:rnd:	cha:
21	1	俄	ga	1	e	1b	t1:rnd:pro:	cha:
21	1	煨	ga	1	e	1b	t1:rnd:pro:	cha:
21	1	鹅	ga	1	e	1b	t1:rnd:pro:	cha:
21	2	流	gua	1	e	1b	t1:u-del:rnd:pro:	cha:
21	3	颞	gac	4	e	1b	t4:vocal:fuse:coal:[cha]:pro:apo:	cha:
21	4	饿	ga	3	e	3	rnd:pro:	cha:
21	5	悉	?ak	4	e	3	t4:rnd:pro:apo:	cha:
21	6	波	pua	1	po	1a	t1:med-b:rnd:	cha:
21	7	播	pua	1	po	1a	t1:med-b:rnd:	cha:
21	8	破	p ^h ua	1	po	1a	t1:med-b:rnd:[de-asp]:	cha:
21	9	梭	puat	4	po	1a	t4:rnd:apo:med-b:	cha:
22	1	剥	pak	4	po	1a	t4:[rnd]:	
22	2	博	pak	4	po	1b	t4:rnd:apo:	cha:
22	3	驳	pak	4	po	1b	t4:[rnd]:	
22	4	伯	pac	4	po	1b	t4:vocal:fuse:coal:[cha]:apo:	cha:
22	5	簸	pua	2	po	2	med-b:rnd:	cha:
22	6	簸	pua	3	po	3	med-b:rnd:	cha:
22	7	坡	p ^h ua	1	p ^h o	1a	t1:med-b:rnd:	cha:
22	8	颇	p ^h ua	1	p ^h o	1a	t1:med-b:rnd:	cha:
22	9	泼	p ^h uat	4	p ^h o	1a	t4:rnd:apo:med-b:	
23	1	婆	bua	1	p ^h o	1b	asp:t1:devoi:med-b:rnd:	cha:
23	2	颇	p ^h ua	3	p ^h o	3	med-b:rnd:	cha:
23	3	迫	pac	4	p ^h o	3	t4:vocal:fuse:coal:[cha]:apo:[asp]:	cha:
23	4	摸	mak	4	mo	1a	rnd:apo:[t4]:	cha:
23	5	摩	mua	1	mo	1b	t1:med-b:rnd:	cha:
23	5	魔	mua	1	mo	1b	t1:med-b:rnd:	cha:
23	6	磨	mua	1	mo	1b	t1:med-b:rnd:	cha:
23	7	模	mo	1	mo	1b	t1:	
23	8	抹	muat	4	mo	2	rnd:apo:med-b:[t4]:	cha:
23	9	磨	mua	3	mo	3	med-b:rnd:	cha:
24	1	末	muat	4	mo	3	t4:rnd:apo:med-b:	
24	2	沫	muat	4	mo	3	t4:rnd:apo:med-b:	
24	3	没	muət	4	mo	3	t4:apo:med-b:	cha:
24	4	美	mak	4	mo	3	t4:apo:[rnd]:	
24	5	墨	mək	4	mo	3	t4:apo:	cha:
24	6	默	mak	4	mo	3	t4:apo:	cha:
24	7	佛	vyat	4	fo	1b	t4:devoi:lab-dnt':apo:med-b:	cha:
24	8	多	ta	1	tuo	1a	t1:rnd:u-ins-b:	
24	9	夺	duat	4	tuo	1b	t4:devoi:rnd:apo:	cha:
25	1	朵	tua	2	tuo	2	rnd:	cha:
25	2	馥	tua	2	tuo	2	rnd:	cha:
25	3	情	dua	2	tuo	3	t2:devoi:rnd:	cha:
25	4	拖	t ^h a	1	t ^h uo	1a	t1:rnd:u-ins-b:	cha:
25	5	它	t ^h a	1	t ^h uo	1a	t1:rnd:u-ins-b:	cha:
25	6	脱	t ^h uat	4	t ^h uo	1a	t4:rnd:[cha]:apo:	
25	7	托	t ^h ak	4	t ^h uo	1a	t4:apo:rnd:u-ins-b:	cha:
25	7	托	t ^h ak	4	t ^h uo	1a	t4:apo:rnd:u-ins-b:	cha:
25	8	驮	da	1	t ^h uo	1b	asp:t1:devoi:rnd:u-ins-b:	cha:
25	8	驮	da	1	t ^h uo	1b	asp:t1:devoi:rnd:u-ins-b:	cha:
25	9	妥	t ^h ua	2	t ^h uo	2	rnd:	cha:
26	1	挪	na	1	nuo	2	rnd:u-ins-b:[t1-t2]:	cha:
26	2	糯	nua	3	nuo	3	rnd:	
26	3	罗	la	1	luo	1b	t1:rnd:u-ins-b:	cha:
26	3	罗	la	1	luo	1b	t1:rnd:u-ins-b:	cha:
26	4	萝	la	1	luo	1b	t1:rnd:u-ins-b:	cha:
26	5	骡	lua	1	luo	1b	t1:rnd:	cha:
26	6	洛	lak	4	luo	3	t4:apo:rnd:u-ins-b:	cha:
26	6	络	lak	4	luo	3	t4:apo:rnd:u-ins-b:	cha:
26	6	络	lak	4	luo	3	t4:apo:rnd:u-ins-b:	cha:
26	7	落	lak	4	luo	3	t4:apo:rnd:u-ins-b:	cha:
26	8	昨	dzak	4	tsuo	1b	t4:devoi:apo:rnd:u-ins-b:	cha:
26	9	凿	dzak	4	tsuo	1b	t4:devoi:apo:rnd:u-ins-b:	cha:
27	1	左	tza	2	tsuo	2	rnd:u-ins-b:	cha:
27	2	坐	dzua	2	tsuo	3	t2:devoi:rnd:	cha:
27	3	座	dzua	2	tsuo	3	t2:devoi:rnd:	cha:
27	4	做	tso	3	tsuo	3	u-ins-a:	
27	5	作	tsak	4	tsuo	3	t4:apo:rnd:u-ins-b:	cha:
27	6	搓	ts ^h a	1	ts ^h uo	1a	t1:rnd:u-ins-b:	cha:
27	7	错	ts ^h o	3	ts ^h uo	3	u-ins-a:	
27	8	糞	sua	1	suo	1a	t1:rnd:	cha:
27	9	梭	sua	1	suo	1a	t1:rnd:	cha:
28	1	缩	giuk	4	suo	1a	t4:i-del:deret:[o-ins]:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
28	2	锁	sua	2	suo	2	rnd:	cha:
28	3	琐	sua	2	suo	2	rnd:	cha:
28	4	所	gio	2	suo	2	i-del:u-ins-a:[deret]:	
28	5	索	sak	4	suo	2	t4:apo:rnd:u-ins-b:	cha:
28	6	拙	tšyat	4	tšuo	1a	t4:apo:med-a:retrof-2:u-ins-b:[rnd]:	cha:
28	7	卓	ʃak	4	tšuo	1a	t4:fric:retrof-1:apo:u-ins-b:[rnd]:	cha:
28	8	捉	tšak	4	tšuo	1a	t4:apo:u-ins-b:[rnd]:	cha:
28	9	着	diak	4	tšuo	1b	t4:devoi:fric:apo:med-a:retrof-2:u-ins-b:[rnd]:	cha:
29	1	浊	ʃak	4	tšuo	1b	t4:devoi:fric:retrof-1:apo:u-ins-b:[rnd]:	cha:
29	2	鐏	džak	4	tšuo	1b	affric:t4:devoi:apo:u-ins-b:[rnd]:	cha:
29	3	戮	ʃ ^h ak	4	tš ^h uo	1a	t4:fric:retrof-1:apo:u-ins-b:[rnd]:	cha:
29	4	说	šyat	4	šuo	1a	t4:apo:med-a:retrof-2:u-ins-b:[rnd]:	cha:
29	5	若	riiak	4	ruo	3	t4:fric:apo:med-a:retrof-2:u-ins-b:[rnd]:	cha:
29	6	弱	riak	4	ruo	3	t4:fric:apo:med-a:retrof-2:u-ins-b:[rnd]:	cha:
29	7	锅	kua	1	kuo	1a	t1:rnd:	cha:
29	8	郭	kuak	4	kuo	1a	t4:apo:rnd:	cha:
29	9	国	kuək	4	kuo	1b	t4:apo:	cha:
30	1	果	kua	2	kuo	2	rnd:	cha:
30	1	裹	kua	2	kuo	2	rnd:	cha:
30	2	过	kua	3	kuo	3	rnd:	cha:
30	3	阔	k ^h uat	4	k ^h uo	3	t4:apo:rnd:	cha:
30	4	括	kuat	4	k ^h uo	3	t4:apo:rnd:[asp]:	cha:
30	5	扩	k ^h ak	4	k ^h uo	3	t4:apo:rnd:[u-ins-b]:	cha:
30	6	活	yuat	4	xuo	1b	t4:devoi:apo:rnd:	cha:
30	7	火	xua	2	xuo	2	rnd:	cha:
30	8	伙	xua	2	xuo	2	rnd:	cha:
30	9	货	xua	3	xuo	3	rnd:	cha:
31	1	祸	yua	2	xuo	3	t2:devoi:rnd:	cha:
31	2	和	yua	3	xuo	3	devoi:rnd:	cha:
31	3	或	yuək	4	xuo	3	t4:devoi:apo:	cha:
31	4	获	yuac	4	xuo	3	t4:devoi:vocal:fuse:coal:apo:	cha:
31	5	窝	?ua	1	uo	1a	t1:pro:rnd:	cha:
31	6	我	ɲa	2	uo	2	pro:rnd:[u-ins-b]:	cha:
31	7	卧	ɲua	3	uo	3	pro:rnd:	cha:
31	8	蟹	?ak	4	uo	3	t4:pro:vocal:coal:[u-ins-b]:	
31	9	蟹	piat	4	pie	1a	t4:raise-il:apo:	cha:
32	1	别	biat	4	pie	1b	t4:devoi:raise-il:apo:	cha:
32	2	撇	p ^h iat	4	p ^h ie	2	t4:raise-il:apo:	cha:
32	3	灭	miat	4	mie	3	t4:raise-il:apo:	cha:
32	4	爹	ʃia	1	tie	1a	t1:raise-il:[t'-t]:	cha:
32	5	蝶	diat	4	tie	1a	devoi:raise-il:apo:[t4]:	cha:
32	6	蝶	diap	4	tie	1b	t4:devoi:raise-il:apo:	cha:
32	6	蝶	diap	4	tie	1b	t4:devoi:raise-il:apo:	cha:
32	7	蝶	diap	4	tie	1b	t4:devoi:raise-il:apo:	cha:
32	8	蝶	diap	4	tie	1b	t4:devoi:raise-il:apo:	cha:
32	9	贴	t ^h iap	4	t ^h ie	1a	t4:raise-il:apo:	cha:
33	1	帖	t ^h iap	4	t ^h ie	1a	t4:raise-il:apo:	cha:
33	2	铁	t ^h iat	4	t ^h ie	2	t4:raise-il:apo:	cha:
33	3	捏	niat	4	nie	1a	raise-il:apo:[t4]:	cha:
33	4	猎	liap	4	lie	3	t4:raise-il:apo:	cha:
33	5	列	liat	4	lie	3	t4:raise-il:apo:	cha:
33	5	烈	liat	4	lie	3	t4:raise-il:apo:	cha:
33	6	裂	liat	4	lie	3	t4:raise-il:apo:	cha:
33	7	劣	lyat	4	lie	3	t4:[y-spl-d]:raise-il:apo:	cha:
33	8	皆	kai	1	tšie	1a	t1:coal:sharp:pal:	cha:
33	9	阶	kai	1	tšie	1a	t1:coal:sharp:pal:	cha:
34	1	街	kai	1	tšie	1a	t1:coal:sharp:pal:	cha:
34	2	接	tšiap	4	tšie	1a	t4:raise-il:apo:pal:	cha:
34	3	结	kiat	4	tšie	1a	t4:raise-il:apo:pal:	cha:
34	4	杰	giat	4	tšie	1b	t4:devoi:raise-il:apo:pal:	cha:
34	5	节	tšiat	4	tšie	1b	t4:raise-il:apo:pal:	cha:
34	6	洁	kiat	4	tšie	1b	t4:raise-il:apo:pal:	cha:
34	7	截	dziat	4	tšie	1b	t4:devoi:raise-il:apo:pal:	cha:
34	8	姐	tšia	2	tšie	2	raise-il:pal:	cha:
34	9	解	kai	2	tšie	2	coal:sharp:pal:	cha:
35	1	借	tšia	3	tšie	3	raise-il:pal:	cha:
35	2	介	kai	3	tšie	3	coal:sharp:pal:	cha:
35	3	戒	kai	3	tšie	3	coal:sharp:pal:	cha:
35	3	界	kai	3	tšie	3	coal:sharp:pal:	cha:
35	3	芥	kai	3	tšie	3	coal:sharp:pal:	cha:
35	4	疥	kaj	3	tšie	3	coal:sharp:pal:	cha:
35	5	切	tš ^h iat	4	tš ^h ie	1a	t4:raise-il:apo:pal:	cha:
35	6	茄	gia	1	tš ^h ie	1b	asp:t1:devoi:pal:[fric]:[raise-il]:	cha:
35	7	且	tš ^h ia	2	tš ^h ie	2	raise-il:apo:	cha:
35	8	怯	k ^h iap	4	tš ^h ie	3	t4:raise-il:apo:pal:	cha:
35	9	些	sia	1	šie	1a	t1:raise-il:pal:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
36	1	歌	xiat	4	šie	1a	t4:raise-il:apo:pal:	cha:
36	2	邪	zia	1	šie	1b	t1:devoi:raise-il:pal:	cha:
36	3	鞋	yai	1	šie	1b	t1:devoi:coal:sharp:pal:	cha:
36	4	协	yiap	4	šie	1b	t4:devoi:raise-il:apo:pal:	cha:
36	5	写	sia	2	šie	2	raise-il:pal:	cha:
36	6	血	xyat	4	šie	2	t4:y-spl-d:raise-il:apo:pal:	cha:
36	7	泻	sia	3	šie	3	raise-il:pal:	cha:
36	8	卸	sia	3	šie	3	raise-il:pal:	cha:
36	9	谢	zia	3	šie	3	devoi:raise-il:pal:	cha:
37	1	械	yai	3	šie	3	devoi:coal:sharp:pal:	cha:
37	2	蚕	yai	2	šie	3	t2:devoi:coal:sharp:pal:	cha:
37	3	漉	siat	4	šie	3	t4:raise-il:apo:pal:	cha:
37	4	噎	?iat	4	ie	1a	t4:pro:raise-il:apo:	cha:
37	5	也	ia	2	ie	2	raise-il:	cha:
37	6	野	ia	2	ie	2	raise-il:	cha:
37	7	夜	ia	3	ie	3	raise-il:	cha:
37	8	叶	iap	4	ie	3	t4:raise-il:apo:	cha:
37	9	业	niap	4	ie	3	t4:pro:raise-il:apo:	cha:
38	1	疔	niak	4	nye	3	t4:vocal:coal:apo:lab-abs:[ŋ-n]:	cha:
38	1	虑	niak	4	nye	3	t4:vocal:coal:apo:lab-abs:[ŋ-n]:	cha:
38	2	略	liak	4	lye	3	t4:vocal:coal:apo:lab-abs:	cha:
38	3	绝	dzyat	4	tšye	1b	t4:devoi:raise-il:apo:pal:	cha:
38	4	掘	gyat	4	tšye	1b	t4:devoi:raise-il:apo:pal:	cha:
38	5	决	kyat	4	tšye	1b	t4:raise-il:apo:pal:	cha:
38	6	觉	kak	4	tšye	1b	t4:vocal:coal:sharp:apo:lab-abs:pal:	cha:
38	7	缺	k ^h yat	4	tš ^h ye	1a	t4:raise-il:apo:pal:	cha:
38	8	却	k ^h iak	4	tš ^h ye	3	t4:vocal:coal:apo:lab-abs:pal:	cha:
38	9	确	k ^h ak	4	tš ^h ye	3	t4:vocal:coal:sharp:apo:lab-abs:pal:	cha:
39	1	靴	xya	1	šye	1a	t1:raise-il:pal:	cha:
39	2	靴	siat	4	šye	1a	t4:[vocal]:coal:apo:pal:	cha:
39	3	削	siak	4	šye	1a	t4:vocal:coal:apo:pal:	cha:
39	4	学	yak	4	šye	1b	t4:devoi:vocal:coal:apo:pal:	cha:
39	5	雪	syat	4	šye	2	t4:raise-il:apo:pal:	cha:
39	6	约	?iak	4	ye	1a	t4:pro:vocal:coal:apo:lab-abs:	cha:
39	7	悦	yat	4	ye	3	t4:raise-il:apo:	cha:
39	8	阅	yat	4	ye	3	t4:raise-il:apo:	cha:
39	9	月	nyat	4	ye	3	t4:pro:raise-il:apo:	cha:
40	1	越	yat	4	ye	3	t4:raise-il:apo:	cha:
40	2	岳	ŋak	4	ye	3	t4:pro:vocal:coal:apo:	cha:
40	3	乐	ŋak	4	ye	3	t4:pro:vocal:coal:apo:	cha:
40	4	资	tsi	1	tsī	1a	t1:Apic:syllab:	cha:
40	5	姿	tsi	1	tsī	1a	t1:Apic:syllab:	cha:
40	6	兹	tsi	1	tsī	1a	t1:Apic:syllab:	cha:
40	7	滋	tsi	1	tsī	1a	t1:Apic:syllab:	cha:
40	8	紫	tsi	2	tsī	2	Apic:syllab:	cha:
40	9	姊	tsi	2	tsī	2	Apic:syllab:	cha:
41	1	子	tsi	2	tsī	2	Apic:syllab:	cha:
41	2	自	dzi	3	tsī	3	devoi:Apic:syllab:	cha:
41	3	字	dzi	3	tsī	3	devoi:Apic:syllab:	cha:
41	4	雌	ts ^h i	1	ts ^h ī	1b	Apic:syllab:[t1]:	cha:
41	5	差	ts ^h i	1	ts ^h ī	1a	t1:Apic:syllab:	cha:
41	6	瓷	dzi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:	cha:
41	7	慈	dzi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:	cha:
41	7	磁	dzi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:	cha:
41	8	辞	zi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:[fric]:	cha:
41	9	祠	zi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:[fric]:	cha:
41	9	词	zi	1	ts ^h ī	1b	asp:t1:devoi:Apic:syllab:[fric]:	cha:
42	1	此	ts ^h i	2	ts ^h ī	2	Apic:syllab:	cha:
42	2	刺	ts ^h i	3	ts ^h ī	3	Apic:syllab:	cha:
42	3	次	ts ^h i	3	ts ^h ī	3	Apic:syllab:	cha:
42	4	司	si	1	sī	1a	t1:Apic:syllab:	cha:
42	5	斯	si	1	sī	1a	t1:Apic:syllab:	cha:
42	5	斯	si	1	sī	1a	t1:Apic:syllab:	cha:
42	6	私	si	1	sī	1a	t1:Apic:syllab:	cha:
42	7	丝	si	1	sī	1a	t1:Apic:syllab:	cha:
42	8	思	si	1	sī	1a	t1:Apic:syllab:	cha:
42	9	死	si	2	sī	2	Apic:syllab:	cha:
43	1	赐	si	3	ts ^h i	3	Apic:syllab:[asp]:[fric]:	cha:
43	2	四	si	3	sī	3	Apic:syllab:	cha:
43	3	肆	si	3	sī	3	Apic:syllab:	cha:
43	4	似	zi	2	sī	3	t2:devoi:Apic:syllab:	cha:
43	5	祀	zi	2	sī	3	t2:devoi:Apic:syllab:	cha:
43	6	寺	zi	3	sī	3	devoi:Apic:syllab:	cha:
43	7	伺	zi	3	sī	3	devoi:Apic:syllab:	cha:
43	8	知	fi	1	tšī	1a	t1:fric:med-a:retrof-2:syllab:	cha:
43	9	蜘	fi	1	tšī	1a	t1:fric:med-a:retrof-2:syllab:	cha:
44	1	文	tši	1	tšī	1a	t1:med-a:retrof-2:syllab:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
44	2	枝	tʂi	1	tʂi	1a	t1:med-a:retrof-2:syllab:	
44	2	肢	tʂi	1	tʂi	1a	t1:med-a:retrof-2:syllab:	
44	3	脂	tʂi	1	tʂi	1a	t1:med-a:retrof-2:syllab:	
44	4	之	tʂi	1	tʂi	1a	t1:med-a:retrof-2:syllab:	
44	5	芝	tʂi	1	tʂi	1a	t1:med-a:retrof-2:syllab:	
44	6	汁	tʂiəp	4	tʂi	1a	t4:rime:Apo:med-a:retrof-2:syllab:	
44	7	织	tʂiək	4	tʂi	1a	t4:rime:Apo:med-a:retrof-2:syllab:	
44	8	只	tʂiac	4	tʂi	1a	t4:vocal:fuse:coal:[cha]:rime:Apo:med-a:retrof-2:syllab:	
44	9	执	tʂiəp	4	tʂi	1b	t4:rime:Apo:med-a:retrof-2:syllab:	
45	1	任	ɕiət	4	tʂi	1b	t4:devoi:fric:rime:Apo:med-a:retrof-2:syllab:	
45	2	直	ɕiək	4	tʂi	1b	t4:devoi:fric:rime:Apo:med-a:retrof-2:syllab:	
45	3	值	ɕiək	4	tʂi	1b	t4:devoi:fric:rime:Apo:med-a:retrof-2:syllab:	
45	4	植	ɕiək	4	tʂi	1b	t4:devoi:rime:apo:med-a:retrof-2:syllab:[affric]:	
45	5	职	tʂiək	4	tʂi	2	t4:rime:Apo:med-a:retrof-2:syllab:	
45	6	纸	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
45	7	只	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
45	8	旨	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
45	9	指	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
46	1	址	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
46	1	止	tʂi	2	tʂi	2	med-a:retrof-2:syllab:	
46	2	制	tʂiai	3	tʂi	3	raise-i1:raise-i2:degem:degem:med-a:retrof-2:syllab:	
46	3	智	ʈi	3	tʂi	3	fric:med-a:retrof-2:syllab:	
46	4	致	ʈi	3	tʂi	3	fric:med-a:retrof-2:syllab:	
46	5	至	tʂi	3	tʂi	3	med-a:retrof-2:syllab:	
46	6	置	ʈi	3	tʂi	3	fric:med-a:retrof-2:syllab:	
46	7	志	tʂi	3	tʂi	3	med-a:retrof-2:syllab:	
46	7	志	tʂi	3	tʂi	3	med-a:retrof-2:syllab:	
46	8	穗	tʂi	3	tʂi	3	med-a:retrof-2:syllab:	
46	9	治	ɕi	3	tʂi	3	devoi:fric:med-a:retrof-2:syllab:	
47	1	质	tʂiat	4	tʂi	3	t4:rime:Apo:med-a:retrof-2:syllab:	
47	2	吃	k ^h iac	4	tʂ ^h i	1a	t4:vocal:fuse:coal:[cha]:rime:Apo:[pal]:med-a:retrof-2:syllab:	
47	3	池	ɕi	1	tʂ ^h i	1b	t1:devoi:fric:med-a:retrof-2:syllab:	
47	4	驰	ɕi	1	tʂ ^h i	1b	t1:devoi:fric:med-a:retrof-2:syllab:	
47	5	迟	ɕi	1	tʂ ^h i	1b	t1:devoi:fric:med-a:retrof-2:syllab:	
47	6	匙	ʂi	1	tʂ ^h i	1b	asp:affric:t1:devoi:med-a:retrof-2:syllab:	
47	7	耻	ʈ ^h i	2	tʂ ^h i	2	fric:med-a:retrof-2:syllab:	
47	8	齿	tʂ ^h i	2	tʂ ^h i	2	med-a:retrof-2:syllab:	
47	9	尺	tʂ ^h iac	4	tʂ ^h i	2	t4:vocal:fuse:coal:[cha]:rime:Apo:med-a:retrof-2:syllab:	
48	1	翅	ʂi	3	tʂ ^h i	3	med-a:retrof-2:syllab:[asp]:[affric]:	
48	2	赤	tʂ ^h iac	4	tʂ ^h i	3	t4:vocal:fuse:coal:[cha]:rime:Apo:med-a:retrof-2:syllab:	
48	3	施	ʂi	1	ʂi	1a	t1:med-a:retrof-2:syllab:	
48	4	师	ʂi	1	ʂi	1a	t1:med-a:retrof-2:syllab:	
48	5	狮	ʂi	1	ʂi	1a	t1:med-a:retrof-2:syllab:	
48	6	尸	ʂi	1	ʂi	1a	t1:med-a:retrof-2:syllab:	
48	7	诗	ʂi	1	ʂi	1a	t1:med-a:retrof-2:syllab:	
48	8	虱	ʂiap	4	ʂi	1a	t4:rime:Apo:med-a:retrof-2:syllab:	
48	9	虱	ʂiat	4	ʂi	1a	t4:rime:Apo:syllab:[i-del]:	
49	1	失	ʂiat	4	ʂi	1a	t4:rime:Apo:med-a:retrof-2:syllab:	
49	2	时	ʂi	1	ʂi	1b	t1:devoi:med-a:retrof-2:syllab:[affric]:	
49	3	十	ʂiap	4	ʂi	1b	affric:t4:devoi:rime:Apo:med-a:retrof-2:syllab:	
49	4	拾	ʂiap	4	ʂi	1b	affric:t4:devoi:rime:Apo:med-a:retrof-2:syllab:	
49	5	实	dʂiət	4	ʂi	1b	affric:t4:devoi:rime:Apo:med-a:retrof-2:syllab:	
49	6	食	dʂiək	4	ʂi	1b	affric:t4:devoi:rime:Apo:med-a:retrof-2:syllab:	
49	7	蚀	dʂiək	4	ʂi	1b	affric:t4:devoi:rime:Apo:med-a:retrof-2:syllab:	
49	8	识	ʂiək	4	ʂi	1b	t4:rime:Apo:med-a:retrof-2:syllab:	
49	9	石	ʂiac	4	ʂi	1b	affric:t4:devoi:vocal:fuse:coal:[cha]:rime:Apo:med-a:retrof-2:syllab:	
50	1	矢	ʂi	2	ʂi	2	med-a:retrof-2:syllab:	
50	2	屎	ʂi	2	ʂi	2	med-a:retrof-2:syllab:	
50	3	史	ʂi	2	ʂi	2	i-del:syllab:	
50	4	使	ʂi	2	ʂi	2	i-del:syllab:	
50	5	驶	ʂi	2	ʂi	2	i-del:syllab:	
50	6	始	ʂi	2	ʂi	2	med-a:retrof-2:syllab:	
50	7	世	ʂiai	3	ʂi	3	raise-i1:raise-i2:degem:degem:med-a:retrof-2:syllab:	
50	7	势	ʂiai	3	ʂi	3	raise-i1:raise-i2:degem:degem:med-a:retrof-2:syllab:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
50	8	誓	ziai	3	gi	3	devoi:raise-i1:raise-i2:degem:degem:med-a:retrof-2:syllab:	
50	9	是	zi	2	gi	3	affric:t2:devoi:med-a:retrof-2:syllab:	
51	1	示	dzi	3	gi	3	affric:devoi:med-a:retrof-2:syllab:	
51	2	视	zi	3	gi	3	affric:devoi:med-a:retrof-2:syllab:	
51	3	士	dzi	2	gi	3	affric:t2:devoi:i-del:syllab:	
51	4	柿	dzi	2	gi	3	affric:t2:devoi:i-del:syllab:	
51	5	事	dzi	3	gi	3	affric:devoi:i-del:syllab:	
51	6	试	si	3	gi	3	med-a:retrof-2:syllab:	
51	7	市	zi	2	gi	3	affric:t2:devoi:med-a:retrof-2:syllab:	
51	8	侍	zi	2	gi	3	affric:t2:devoi:med-a:retrof-2:syllab:	
51	9	室	siət	4	gi	3	t4:rime:Apo:med-a:retrof-2:syllab:	
52	1	式	siak	4	gi	3	t4:rime:Apo:med-a:retrof-2:syllab:	
52	2	适	siac	4	gi	3	t4:vocal:fuse:coal:[cha]:rime:Apo:med-a:retrof-2:syllab:	
52	3	日	niət	4	ri	3	t4:fric:rime:Apo:med-a:retrof-2:syllab:syllab:	
52	4	儿	ni	1	ər	1b	t1:fric:med-a:retrof-2:syllab:syllab:	
52	5	而	ni	1	ər	1b	t1:fric:med-a:retrof-2:syllab:syllab:	
52	6	耳	ni	2	ər	2	fric:med-a:retrof-2:syllab:syllab:	
52	7	二	ni	3	ər	3	fric:med-a:retrof-2:syllab:syllab:	
52	8	逼	piək	4	pi	1a	t4:rime:Apo:	
52	9	鼻	bi	3	pi	1b	devoi:[t3-t1b]:	
53	1	彼	pi	2	pi	2	:	
53	2	比	pi	2	pi	2	:	
53	3	廊	pi	2	pi	2	:	
53	4	笔	piət	4	pi	2	t4:rime:Apo:	
53	5	蔽	piai	3	pi	3	raise-i1:raise-i2:degem:degem:	
53	6	币	biai	3	pi	3	devoi:raise-i1:raise-i2:degem:degem:	
53	6	弊	biai	3	pi	3	devoi:raise-i1:raise-i2:degem:degem:	
53	6	蔽	biai	3	pi	3	devoi:raise-i1:raise-i2:degem:degem:	
53	7	苑	biai	3	pi	3	devoi:raise-i1:raise-i2:degem:degem:	
53	8	闭	piai	3	pi	3	raise-i1:raise-i2:degem:degem:	
53	9	臂	pi	3	pi	3	:	
54	1	避	bi	3	pi	3	devoi:	
54	2	鹿	bi	3	pi	3	devoi:	
54	3	毕	piət	4	pi	3	t4:rime:apo:	
54	4	必	piət	4	pi	3	t4:rime:apo:	
54	5	壁	piac	4	pi	3	t4:vocal:fuse:coal:[cha]:rime:Apo:	
54	6	壁	piac	4	pi	3	t4:vocal:fuse:coal:[cha]:rime:Apo:	
54	7	批	p ^h iai	1	p ^h i	1a	t1:raise-i1:raise-i2:degem:degem:	
54	8	披	p ^h i	1	p ^h i	1a	t1:	
54	9	劈	p ^h iac	4	p ^h i	1a	t4:raise-i1:raise-i2:degem:degem:	
55	1	皮	bi	1	p ^h i	1b	asp:t1:devoi:	
55	2	疲	bi	1	p ^h i	1b	asp:t1:devoi:	
55	3	脾	bi	1	p ^h i	1b	asp:t1:devoi:	
55	4	匹	p ^h iat	4	p ^h i	2	t4:rime:Apo:	
55	5	譬	p ^h i	3	p ^h i	3	:	
55	6	屁	p ^h i	3	p ^h i	3	:	
55	7	迷	miai	1	mi	1b	t1:raise-i1:raise-i2:degem:degem:	
55	8	谜	miai	1	mi	1b	t1:raise-i1:raise-i2:degem:degem:	
55	9	米	miai	2	mi	2	raise-i1:raise-i2:degem:degem:	
56	1	秘	pi	3	mi	3	[p-m]:	
56	2	密	miət	4	mi	3	t4:rime:apo:	
56	3	蜜	miət	4	mi	3	t4:rime:apo:	
56	4	觅	miac	4	mi	3	t4:vocal:fuse:coal:[cha]:rime:Apo:	
56	5	低	tiai	1	ti	1a	t1:raise-i1:raise-i2:degem:degem:	
56	6	堤	tiai	1	ti	1a	t1:raise-i1:raise-i2:degem:degem:	
56	7	滴	tiac	4	ti	1a	t4:vocal:fuse:coal:[cha]:rime:apo:	
56	8	的	tiac	4	ti	1b	t4:vocal:fuse:coal:[cha]:rime:apo:	
56	9	敌	diac	4	ti	1b	t4:devoi:vocal:fuse:coal:[cha]:rime:apo:	
57	1	狄	diac	4	ti	1b	t4:devoi:vocal:fuse:coal:[cha]:rime:apo:	
57	1	笛	diac	4	ti	1b	t4:devoi:vocal:fuse:coal:[cha]:rime:apo:	
57	2	底	tiai	2	ti	2	raise-i1:raise-i2:degem:degem:	
57	3	抵	tiai	2	ti	2	raise-i1:raise-i2:degem:degem:	
57	3	帝	tiai	2	ti	2	raise-i1:raise-i2:degem:degem:	
57	4	帝	tiai	3	ti	3	raise-i1:raise-i2:degem:degem:	
57	5	弟	diai	2	ti	3	t2:devoi:raise-i1:raise-i2:degem:degem:	
57	6	第	diai	3	ti	3	devoi:raise-i1:raise-i2:degem:degem:	
57	7	递	diai	3	ti	3	devoi:raise-i1:raise-i2:degem:degem:	
57	8	地	di	3	ti	3	devoi:	
57	9	梯	t ^h iai	1	t ^h i	1a	t1:raise-i1:raise-i2:degem:degem:	
58	1	踮	t ^h iac	4	t ^h i	1a	t4:vocal:fuse:coal:[cha]:rime:apo:	
58	2	剔	t ^h iac	4	t ^h i	1a	t4:vocal:fuse:coal:[cha]:rime:apo:	
58	3	题	diai	1	t ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:	
58	4	提	diai	1	t ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:	
58	5	啼	diai	1	t ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
58	6	踴	diai	1	t ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:	
58	7	体	t ^h iai	2	t ^h i	2	raise-i1:raise-i2:degem:degem:	
58	8	替	t ^h iai	3	t ^h i	3	raise-i1:raise-i2:degem:degem:	
58	9	涕	t ^h iai	3	t ^h i	3	raise-i1:raise-i2:degem:degem:	
59	1	制	t ^h iai	3	t ^h i	3	raise-i1:raise-i2:degem:degem:	
59	2	泥	ni	1	ni	1b	t1:raise-i1:raise-i2:degem:degem:	
59	3	尼	ni	1	ni	1b	t1:	
59	4	你	ni	2	ni	2	:	
59	5	赋	ni	3	ni	3	:	
59	6	逆	ɲiac	4	ni	3	t4:[ŋ-n]:vocal:fuse:coal:[cha]:rime:apo:	
59	7	翠	liai	1	li	1b	t1:raise-i1:raise-i2:degem:degem:	
59	8	黎	liai	1	li	1b	t1:raise-i1:raise-i2:degem:degem:	
59	9	离	li	1	li	1b	t1:	
60	1	篱	li	1	li	1b	t1:	
60	2	梨	li	1	li	1b	t1:	
60	3	厘	li	1	li	1b	t1:	
60	4	理	li	1	li	1b	t1:	
60	5	礼	liai	2	li	2	raise-i1:raise-i2:degem:degem:[t2]:	
60	6	理	li	2	li	2	:	
60	6	里	li	2	li	2	:	
60	6	鲤	li	2	li	2	:	
60	7	李	li	2	li	2	:	
60	8	里	li	2	li	2	:	
60	9	例	liai	3	li	3	raise-i1:raise-i2:degem:degem:	
61	1	厉	liai	3	li	3	raise-i1:raise-i2:degem:degem:	
61	2	丽	liai	3	li	3	raise-i1:raise-i2:degem:degem:	
61	3	求	liai	3	li	3	raise-i1:raise-i2:degem:degem:	
61	4	利	li	3	li	3	:	
61	5	痢	li	3	li	3	:	
61	6	吏	li	3	li	3	:	
61	7	立	liəp	4	li	3	t4:rime:apo:	
61	8	粒	liəp	4	li	3	t4:rime:apo:	
61	9	乘	liət	4	li	3	t4:rime:apo:	
62	1	力	liək	4	li	3	t4:rime:apo:	
62	2	历	liac	4	li	3	t4:vocal:fuse:coal:[cha]:rime:apo:	
62	2	历	liac	4	li	3	t4:vocal:fuse:coal:[cha]:rime:apo:	
62	3	鸡	kiai	1	t ^h si	1a	t1:vocal:fuse:coal:[cha]:rime:apo:pal:	
62	4	基	ki	1	t ^h si	1a	t1:pal:	
62	5	饥	ki	1	t ^h si	1a	t1:pal:	
62	6	几	ki	1	t ^h si	1a	t1:pal:	
62	7	机	ki	1	t ^h si	1a	t1:pal:	
62	7	机	ki	1	t ^h si	1a	t1:pal:	
62	7	机	ki	1	t ^h si	1a	t1:pal:	
62	8	积	tsiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:apo:pal:	
62	9	激	kiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:apo:pal:	
63	1	击	kiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:apo:pal:	
63	2	集	dziəp	4	t ^h si	1b	t4:devoi:rime:apo:pal:	
63	3	急	kiəp	4	t ^h si	1b	t4:rime:apo:pal:	
63	4	级	kiəp	4	t ^h si	1b	t4:rime:apo:pal:	
63	5	及	giəp	4	t ^h si	1b	t4:devoi:rime:apo:pal:	
63	6	疾	dziət	4	t ^h si	1b	t4:devoi:rime:apo:pal:	
63	7	吉	kiət	4	t ^h si	1b	t4:rime:apo:pal:	
63	8	即	tsiək	4	t ^h si	1b	t4:rime:apo:pal:	
63	9	极	giək	4	t ^h si	1b	t4:devoi:rime:apo:pal:	
64	1	脊	tsiac	4	t ^h si	1b	t4:vocal:fuse:coal:[cha]:rime:apo:pal:	
64	2	籍	dziac	4	t ^h si	1b	t4:devoi:vocal:fuse:coal:[cha]:rime:apo:pal:	
64	3	挤	tsiai	2	t ^h si	2	raise-i1:raise-i2:degem:degem:pal:	
64	4	己	ki	2	t ^h si	2	pal:	
64	5	几	ki	2	t ^h si	2	pal:	
64	6	给	kiəp	4	t ^h si	2	t4:rime:apo:pal:	
64	7	祭	tsiai	3	t ^h si	3	raise-i1:raise-i2:degem:degem:pal:	
64	7	际	tsiai	3	t ^h si	3	raise-i1:raise-i2:degem:degem:pal:	
64	8	济	tsiai	3	t ^h si	3	raise-i1:raise-i2:degem:degem:pal:	
64	9	剂	dzi	3	t ^h si	3	devoi:raise-i1:raise-i2:degem:degem:pal:	
65	1	计	kiai	3	t ^h si	3	raise-i1:raise-i2:degem:degem:pal:	
65	2	继	kiai	3	t ^h si	3	raise-i1:raise-i2:degem:degem:pal:	
65	3	寄	ki	3	t ^h si	3	pal:	
65	4	技	gi	2	t ^h si	3	t2:devoi:pal:	
65	5	枝	gi	2	t ^h si	3	t2:devoi:pal:	
65	6	纪	ki	2	t ^h si	3	pal:[t2]:	
65	7	记	ki	3	t ^h si	3	pal:	
65	8	忌	gi	3	t ^h si	3	devoi:pal:	
65	9	既	ki	3	t ^h si	3	pal:	
66	1	季	kui	3	t ^h si	3	u-del:pal:	
66	2	迹	tsiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:Apo:pal:	
66	3	迹	tsiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:Apo:pal:	
66	4	迹	tsiac	4	t ^h si	1a	t4:vocal:fuse:coal:[cha]:rime:Apo:pal:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
66	5	妻	tsh ^h iai	1	tsh ^h i	1a	t1:raise-i1:raise-i2:degem:degem:pal:	
66	6	凄	tsh ^h iai	1	tsh ^h i	1a	t1:raise-i1:raise-i2:degem:degem:pal:	
66	7	欺	k ^h i	1	tsh ^h i	1a	t1:pal:	
66	8	七	tsh ^h iat	4	tsh ^h i	1a	t4:rime:Apo:pal:	
66	9	漆	tsh ^h iat	4	tsh ^h i	1a	t4:rime:Apo:pal:	
67	1	芥	dziai	1	tsh ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:pal:	
67	2	脐	dziai	1	tsh ^h i	1b	t1:devoi:raise-i1:raise-i2:degem:degem:pal:	
67	3	奇	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	4	畸	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	5	其	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	5	旗	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	5	棋	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	5	麒	gi	1	tsh ^h i	1b	asp:t1:devoi:pal:	
67	6	期	gi	1	tsh ^h i	1a	asp:t1:devoi:pal:	
67	7	启	k ^h iai	2	tsh ^h i	2	raise-i1:raise-i2:degem:degem:pal:	
67	8	起	k ^h i	2	tsh ^h i	2	pal:	
67	9	岂	k ^h i	2	tsh ^h i	2	pal:	
68	1	契	k ^h iai	3	tsh ^h i	3	raise-i1:raise-i2:degem:degem:pal:	
68	2	器	k ^h i	3	tsh ^h i	3	pal:	
68	3	弃	k ^h i	3	tsh ^h i	3	pal:	
68	4	气	k ^h i	3	tsh ^h i	3	pal:	
68	4	汽	k ^h i	3	tsh ^h i	3	pal:	
68	5	西	siai	1	si	1a	t1:raise-i1:raise-i2:degem:degem:pal:	
68	6	犀	siai	1	si	1a	t1:raise-i1:raise-i2:degem:degem:pal:	
68	7	溪	k ^h iai	1	si	1a	t1:raise-i1:raise-i2:degem:degem:[kh-s']:	
68	8	牺	xi	1	si	1a	t1:pal:	
68	9	稀	xi	1	si	1a	t1:pal:	
68	9	希	xi	1	si	1a	t1:pal:	
69	1	吸	xiəp	4	si	1a	t4:rime:Apo:pal:	
69	2	息	siək	4	si	1a	t4:rime:Apo:pal:	
69	3	熄	siək	4	si	1a	t4:rime:Apo:pal:	
69	4	情	siaç	4	si	1a	t4:vocal:fusion:coal:[cha]:rime:Apo:pal:	
69	5	析	siaç	4	si	1a	t4:vocal:fusion:coal:[cha]:rime:Apo:pal:	
69	6	习	ziəp	4	si	1b	t4:devoi:rime:apo:pal:	
69	7	席	ziəç	4	si	1b	t4:devoi:vocal:fusion:coal:[cha]:rime:Apo:pal:	
69	8	锡	siaç	4	si	1a	t4:vocal:fusion:coal:[cha]:rime:apo:pal:	
69	9	洗	siai	2	si	2	raise-i1:raise-i2:degem:degem:pal:	
70	1	喜	xi	2	si	2	pal:	
70	2	细	siai	3	si	3	raise-i1:raise-i2:degem:degem:pal:	
70	3	系	ɣiai	3	si	3	devoi:raise-i1:raise-i2:degem:degem:pal:	
70	4	戏	xi	3	si	3	pal:	
70	5	医	?i	1	i	1a	t1:pro:	
70	6	依	?i	1	i	1a	t1:pro:	
70	6	衣	?i	1	i	1a	t1:pro:	
70	7	一	?iat	4	i	1a	t4:pro:rime:apo:	
70	8	宜	ŋi	1	i	1b	t1:pro:	
70	9	移	i	1	i	1b	t1:	
71	1	爽	i	1	i	1b	t1:	
71	1	姨	i	1	i	1b	t1:	
71	2	疑	ŋi	1	i	1b	t1:pro:	
71	3	遭	ui	1	i	1b	t1:[u-del]:	
71	4	倚	?i	2	i	2	pro:	
71	5	椅	?i	2	i	2	pro:	
71	6	已	i	2	i	2	:	
71	7	以	i	2	i	2	:	
71	8	乙	?iat	4	i	2	t4:pro:rime:apo:	
71	9	艺	ŋiai	3	i	3	pro:raise-i1:raise-i2:degem:degem:	
72	1	义	ŋi	3	i	3	pro:	
72	1	议	ŋi	3	i	3	pro:	
72	2	易	i	3	i	3	:	
72	3	意	?i	3	i	3	pro:	
72	4	异	i	3	i	3	:	
72	5	亿	?iək	4	i	3	t4:pro:rime:apo:	
72	6	益	?iac	4	i	3	t4:pro:vocal:fuse:coal:[cha]:rime:apo:	
72	7	亦	iac	4	i	3	t4:vocal:fuse:coal:[cha]:rime:apo:	
72	8	译	iac	4	i	3	t4:vocal:fuse:coal:[cha]:rime:apo:	
72	9	易	iac	4	i	3	t4:vocal:fuse:coal:[cha]:rime:apo:	
73	1	役	yac	4	i	3	t4:	
73	1	疫	yac	4	i	3	t4:vocal:fuse:coal:[cha]:rime:Y-spl-d:apo:	
73	2	补	po	2	pu	2	raise-u:	
73	3	捕	bo	3	pu	2	devoi:raise-u:[t3-t2]:	
73	4	布	po	3	pu	3	raise-u:	
73	5	布	po	3	pu	3	raise-u:	
73	6	部	bo	2	pu	3	t2:devoi:raise-u:	
73	7	簿	bo	2	pu	3	t2:devoi:raise-u:	
73	8	步	bo	3	pu	3	devoi:raise-u:	
73	9	埠	bo	3	pu	3	devoi:raise-u:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
74	1	不	fuət	4	pu	3	t4:rime:apo:[f-p]:	
74	2	铺	p ^h o	1	p ^h u	1a	t1:raise-u:	
74	3	扑	p ^h uk	4	p ^h u	1a	t4:apo:	
74	4	蒲	bo	1	p ^h u	1b	asp:t1:devoi:raise-u:	
74	5	脯	bo	1	p ^h u	1b	asp:t1:devoi:raise-u:	
74	6	葡	bo	1	p ^h u	1b	asp:t1:devoi:raise-u:	
74	7	仆	buk	4	p ^h u	1b	asp:t4:devoi:apo:	
74	8	谱	po	2	p ^h u	2	raise-u:[asp]:	
74	9	普	p ^h o	2	p ^h u	2	raise-u:	
75	1	铺	p ^h o	3	p ^h u	3	raise-u:	
75	2	模	mo	1	mu	1b	t1:raise-u:	
75	3	母	mu	2	mu	2	:	
75	4	亩	mu	2	mu	2	:	
75	5	暮	mo	3	mu	3	raise-u:	
75	6	墓	mo	3	mu	3	raise-u:	
75	7	墓	mo	3	mu	3	raise-u:	
75	8	幕	muk	4	mu	3	t4:apo:[round]:[raise-u]:	
75	9	木	muk	4	mu	3	t4:apo:	
76	1	目	muk	4	mu	3	t4:apo:	
76	2	牡	miuk	4	mu	3	t4:lab-dnt:	
76	3	肤	fiu	1	fu	1a	t1:lab-dnt':raise-u:	
76	4	夫	fiu	1	fu	1a	t1:lab-dnt':raise-u:	
76	5	敷	f ^h io	1	fu	1a	t1:lab-dnt':[asp-r]:raise-u:	
76	6	敷	f ^h io	1	fu	1a	t1:lab-dnt':[asp-r]:raise-u:	
76	7	孵	f ^h io	1	fu	1a	t1:lab-dnt':[asp-r]:raise-u:	
76	8	浮	viu	1	fu	1b	t1:devoi:lab-dnt':	
76	9	符	vio	1	fu	1b	t1:devoi:lab-dnt':	
77	1	扶	vio	1	fu	1b	t1:devoi:lab-dnt':	
77	2	福	fiuk	4	fu	1b	t4:lab-dnt':apo:	
77	3	服	viuk	4	fu	1b	t4:devoi:lab-dnt':apo:	
77	4	伏	viuk	4	fu	1b	t4:devoi:lab-dnt':apo:	
77	5	俯	fiu	2	fu	2	lab-dnt':raise-u:	
77	5	府	fiu	2	fu	2	lab-dnt':raise-u:	
77	5	腑	fiu	2	fu	2	lab-dnt':raise-u:	
77	6	斧	fiu	2	fu	2	lab-dnt':raise-u:	
77	7	脯	fiu	2	fu	2	lab-dnt':raise-u:	
77	8	甫	fiu	2	fu	2	lab-dnt':raise-u:	
77	9	腐	vio	2	fu	2	devoi:lab-dnt':raise-u:	
78	1	辅	vio	2	fu	2	devoi:lab-dnt':raise-u:	
78	2	付	fiu	3	fu	3	lab-dnt':raise-u:	
78	3	傅	fiu	3	fu	3	lab-dnt':raise-u:	
78	4	父	vio	2	fu	3	devoi:lab-dnt':raise-u:	
78	5	附	vio	3	fu	3	devoi:lab-dnt':raise-u:	
78	6	富	fiu	3	fu	3	lab-dnt':	
78	7	妇	viu	2	fu	3	t2:devoi:lab-dnt':	
78	8	负	viu	2	fu	3	t2:devoi:lab-dnt':	
78	9	复	fiuk	4	fu	3	t4:lab-dnt':apo:	
79	1	腹	fiuk	4	fu	3	t4:lab-dnt':apo:	
79	2	复	viuk	4	fu	3	t4:devoi:lab-dnt':apo:	
79	3	都	to	1	tu	1a	t1:raise-u:	
79	4	独	duk	4	tu	1b	t4:devoi:apo:	
79	5	读	duk	4	tu	1b	t4:devoi:apo:	
79	6	牍	duk	4	tu	1b	t4:devoi:apo:	
79	7	毒	duk	4	tu	1b	t4:devoi:apo:	
79	8	堵	to	2	tu	2	raise-u:	
79	8	赌	to	2	tu	2	raise-u:	
79	9	肚	to	2	tu	2	raise-u:	
80	1	杜	do	2	tu	3	t2:devoi:raise-u:	
80	2	肚	do	2	tu	3	t2:devoi:raise-u:	
80	3	度	do	3	tu	3	devoi:raise-u:	
80	4	渡	do	3	tu	3	devoi:raise-u:	
80	5	镀	do	3	tu	3	devoi:raise-u:	
80	6	突	duət	4	t ^h u	1a	asp:t4:devoi:rime:apo:	
80	7	秃	t ^h uk	4	t ^h u	1a	t4:apo:	
80	8	徒	do	1	t ^h u	1b	asp:t1:devoi:raise-u:	
80	9	涂	do	1	t ^h u	1b	asp:t1:devoi:raise-u:	
80	9	途	do	1	t ^h u	1b	asp:t1:devoi:raise-u:	
81	1	图	do	1	t ^h u	1b	asp:t1:devoi:raise-u:	
81	2	屠	do	1	t ^h u	1b	asp:t1:devoi:raise-u:	
81	3	土	t ^h o	2	t ^h u	2	raise-u:	
81	4	吐	t ^h o	2	t ^h u	2	raise-u:	
81	5	兔	t ^h o	3	t ^h u	3	raise-u:	
81	6	奴	no	1	nu	1b	t1:raise-u:	
81	7	努	no	2	nu	2	raise-u:[t2]:	
81	8	怒	no	3	nu	3	raise-u:	
81	9	炉	lo	1	lu	1b	t1:raise-u:	
81	9	卢	lo	1	lu	1b	t1:raise-u:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
81	9	芦	lo	1	lu	1b	t1:raise-u:	
82	1	槽	lo	2	lu	2	raise-u:[t2]:	
82	1	鲁	lo	2	lu	2	raise-u:[t2]:	
82	2	卤	lo	2	lu	2	raise-u:[t2]:	
82	3	露	lo	3	lu	3	raise-u:	
82	4	路	lo	3	lu	3	raise-u:	
82	5	鹿	luk	4	lu	3	t4:apo:	
82	6	禄	luk	4	lu	3	t4:apo:	
82	7	陆	liuk	4	lu	3	t4:apo:[i-del]:	
82	8	录	liuk	4	lu	3	t4:apo:[i-del]:	
82	9	租	tso	1	tsu	1a	t1:raise-u:	
83	1	卒	tsyət	4	tsu	1b	t4:Y-spl-c:ime:apo:	
83	2	族	dzuk	4	tsu	1b	t4:devoi:apo:	
83	3	足	tsiuk	4	tsu	1b	t4:apo:[i-del]:	
83	4	祖	tso	2	tsu	2	raise-u:	
83	5	组	tso	2	tsu	2	raise-u:	
83	6	阻	tgio	2	tsu	2	i-del:deret:raise-u:	
83	7	租	ts ^h o	1	ts ^h u	1a	t1:raise-u:	
83	8	醋	ts ^h o	3	ts ^h u	3	raise-u:	
83	9	苏	so	1	su	1a	t1:raise-u:	
83	9	酥	so	1	su	1a	t1:raise-u:	
84	1	俗	ziuk	4	su	1b	t4:devoi:apo:[i-del]:	
84	2	嗦	so	3	su	3	raise-u:	
84	2	素	so	3	su	3	raise-u:	
84	2	诉	so	3	su	3	raise-u:	
84	3	塑	so	3	su	3	raise-u:	
84	4	速	suk	4	su	3	t4:apo:	
84	5	宿	siuk	4	su	3	t4:apo:[i-del]:	
84	6	粟	siuk	4	su	3	t4:apo:[i-del]:	
84	7	猪	ŋio	1	tŋu	1a	t1:fric:med-a:retrof-2:raise-u:	
84	8	诸	tsio	1	tŋu	1a	t1:med-a:retrof-2:raise-u:	
84	9	朱	tsio	1	tŋu	1a	t1:med-a:retrof-2:raise-u:	
84	9	珠	tsio	1	tŋu	1a	t1:med-a:retrof-2:raise-u:	
85	1	蛛	ŋio	1	tŋu	1a	t1:fric:med-a:retrof-2:raise-u:	
85	2	株	ŋio	1	tŋu	1a	t1:fric:med-a:retrof-2:raise-u:	
85	3	竹	ŋiuk	4	tŋu	1b	t4:fric:apo:med-a:retrof-2:	
85	4	筑	ŋiuk	4	tŋu	1b	t4:fric:apo:med-a:retrof-2:	
85	5	烛	tsiuk	4	tŋu	1b	t4:apo:med-a:retrof-2:	
85	6	煮	tsio	2	tŋu	2	raise-u:med-a:retrof-2:	
85	7	主	tsio	2	tŋu	2	raise-u:med-a:retrof-2:	
85	8	嘱	tsiuk	4	tŋu	2	t4:apo:med-a:retrof-2:	
85	9	著	ŋio	3	tŋu	3	fric:apo:raise-u:med-a:retrof-2:	
86	1	驻	ŋio	3	tŋu	3	fric:apo:raise-u:med-a:retrof-2:	
86	2	注	ŋio	3	tŋu	3	fric:apo:raise-u:med-a:retrof-2:	
86	3	注	tsio	3	tŋu	3	raise-u:med-a:retrof-2:	
86	4	蛙	tsio	3	tŋu	3	raise-u:med-a:retrof-2:	
86	5	铸	tsio	3	tŋu	3	raise-u:med-a:retrof-2:	
86	6	柱	ŋio	2	tŋu	3	t2:devoi:fric:raise-u:med-a:retrof-2:	
86	7	住	ŋio	3	tŋu	3	devoi:fric:raise-u:med-a:retrof-2:	
86	8	助	dŋio	3	tŋu	3	devoi:i-del:raise-u:	
86	9	祝	tsiuk	4	tŋu	3	t4:apo:med-a:retrof-2:	
87	1	初	tŋ ^h io	1	tŋ ^h u	1a	t1:i-del:raise-u:	
87	2	出	ts ^h yət	4	tŋ ^h u	1a	t4:apo:med-a:ime:retrof-2:	
87	3	除	ŋio	1	tŋ ^h u	1b	t1:devoi:fric:raise-u:med-a:retrof-2:	
87	4	锄	dŋio	1	tŋ ^h u	1b	asp:t1:devoi:i-del:raise-u:	
87	5	厨	ŋio	1	tŋ ^h u	1b	t1:devoi:fric:raise-u:med-a:retrof-2:	
87	6	储	ŋio	1	tŋ ^h u	2	devoi:fric:raise-u:med-a:retrof-2:[t1-t2]:	
87	7	楚	tŋ ^h io	2	tŋ ^h u	2	i-del:raise-u:	
87	7	础	tŋ ^h io	2	tŋ ^h u	2	i-del:raise-u:	
87	8	处	ts ^h io	2	tŋ ^h u	2	i-del:raise-u:	
87	9	处	ts ^h io	3	tŋ ^h u	3	i-del:raise-u:	
88	1	畜	ŋ ^h iuk	4	tŋ ^h u	3	t4:fric:apo:med-a:retrof-2:	
88	2	触	ts ^h iuk	4	tŋ ^h u	3	t4:apo:med-a:retrof-2:	
88	3	梳	gio	1	ŋu	1a	t1:i-del:raise-u:	
88	4	疏	gio	1	ŋu	1a	t1:i-del:raise-u:	
88	5	书	sio	1	ŋu	1a	t1:raise-u:med-a:retrof-2:	
88	6	舒	sio	1	ŋu	1a	t1:raise-u:med-a:retrof-2:	
88	7	输	sio	1	ŋu	1a	t1:raise-u:med-a:retrof-2:	
88	8	殊	zio	1	ŋu	1a	devoi:raise-u:med-a:retrof-2:[t1]:	
88	9	叔	siuk	4	ŋu	1a	t4:apo:med-a:retrof-2:	
89	1	嘘	dziuk	4	ŋu	1b	affric:t4:devoi:apo:med-a:retrof-2:	
89	2	暑	sio	2	ŋu	2	raise-u:med-a:retrof-2:	
89	2	黍	sio	2	ŋu	2	raise-u:med-a:retrof-2:	
89	3	鼠	sio	2	ŋu	2	raise-u:med-a:retrof-2:	
89	4	数	gio	2	ŋu	2	raise-u:med-a:retrof-2:	
89	5	属	ziuk	4	ŋu	2	affric:t4:devoi:apo:med-a:retrof-2:	
89	6	数	gio	3	ŋu	3	i-del:raise-u:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
89	7	竖	zio	2	gu	3	affric:t2:devoi:raise-u:med-a:retrof-2:	
89	8	树	zio	3	gu	3	affric:devoi:raise-u:med-a:retrof-2:	
89	9	漱	giu	3	gu	3	i-del:	
90	1	水	dzyat	4	gu	3	affric:t4:devoi:apo:med-a:rime:retrof-2:	
90	1	述	dzyat	4	gu	3	affric:t4:devoi:apo:med-a:rime:retrof-2:	
90	2	如	nio	1	ru	1b	t1:fric:raise-u:med-a:retrof-2:	
90	3	乳	nio	2	ru	2	fric:raise-u:med-a:retrof-2:	
90	4	入	n	4	ru	3	t4:fric:retrof-2:[u-ins-b]:	
90	5	辱	niuk	4	ru	2	[t4]:fric:apo:med-a:retrof-2:	
90	6	褥	niuk	4	ru	3	t4:fric:apo:med-a:retrof-2:	
90	7	孤	kio	1	ku	1a	t1:raise-u:[i-del]:	
90	7	孤	kio	1	ku	1a	t1:raise-u:[i-del]:	
90	8	辜	kio	1	ku	1a	t1:raise-u:[i-del]:	
90	9	估	ko	2	ku	1a	raise-u:[t2-t1a]:	
91	1	古	ko	2	ku	2	raise-u:	
91	1	股	ko	2	ku	2	raise-u:	
91	1	鼓	ko	2	ku	2	raise-u:	
91	2	骨	kuat	4	ku	2	t4:rime:apo:	
91	3	谷	kuk	4	ku	2	t4:apo:	
91	3	谷	kuk	4	ku	2	t4:apo:	
91	4	故	ko	3	ku	3	raise-u:	
91	5	固	ko	3	ku	3	raise-u:	
91	5	雇	ko	3	ku	3	raise-u:	
91	5	顾	ko	3	ku	3	raise-u:	
91	6	枯	k ^h o	1	k ^h u	1a	t1:raise-u:	
91	7	哭	k ^h uk	4	k ^h u	1a	t4:apo:	
91	8	苦	k ^h o	2	k ^h u	2	raise-u:	
91	9	库	k ^h o	3	k ^h u	3	raise-u:	
91	9	裤	k ^h o	3	k ^h u	3	raise-u:	
92	1	呼	xo	1	xu	1a	t1:raise-u:	
92	2	忽	xuat	4	xu	1a	t4:rime:apo:	
92	3	胡	yo	1	xu	1b	t1:devoi:raise-u:	
92	4	湖	yo	1	xu	1b	t1:devoi:raise-u:	
92	5	糊	yo	1	xu	1b	t1:devoi:raise-u:	
92	6	狐	yo	1	xu	1b	t1:devoi:raise-u:	
92	6	胡	yo	1	xu	1b	t1:devoi:raise-u:	
92	7	胡	yo	1	xu	1b	t1:devoi:raise-u:	
92	8	壶	yo	1	xu	1b	t1:devoi:raise-u:	
92	9	核	yuat	4	xe	2	t4:devoi:rime:apo:	
93	1	虎	xo	2	xu	2	raise-u:	
93	2	户	yo	2	xu	3	t2:devoi:raise-u:	
93	3	互	yo	3	xu	3	devoi:raise-u:	
93	4	互	yo	3	xu	3	devoi:raise-u:	
93	5	乌	?o	1	wu	1a	t1:pro:raise-u:	
93	6	屋	?uk	4	u	1a	t4:pro:raise-u:apo:	
93	7	吴	no	1	u	1b	t1:pro:raise-u:	
93	8	梧	no	1	u	1b	t1:pro:raise-u:	
93	9	无	njo	1	u	1b	t1:lab-dnt':raise-u:degem:	
94	1	伍	no	2	u	2	pro:raise-u:	
94	1	午	no	2	u	2	pro:raise-u:	
94	2	五	no	2	u	2	pro:raise-u:	
94	3	武	njo	2	u	2	lab-dnt':raise-u:degem:	
94	3	舞	njo	2	u	2	lab-dnt':raise-u:degem:	
94	4	悉	?o	3	u	3	pro:raise-u:	
94	5	误	no	3	u	3	pro:raise-u:	
94	6	悟	no	3	u	3	pro:raise-u:	
94	7	务	njo	3	u	3	lab-dnt':raise-u:degem:	
94	8	旁	njo	3	u	3	lab-dnt':raise-u:degem:	
94	9	物	njyat	4	u	3	t4:lab-dnt':degem:rime:apo:	
95	1	勿	njyat	4	u	3	t4:lab-dnt':degem:rime:apo:	
95	2	女	nio	2	ny	2	raise-u:lab-abs:	
95	3	驴	lio	1	ly	1b	t1:raise-u:lab-abs:	
95	4	吕	lio	2	ly	2	raise-u:lab-abs:	
95	5	旅	lio	2	ly	2	raise-u:lab-abs:	
95	6	虑	lio	3	ly	3	raise-u:lab-abs:	
95	7	滤	lio	3	ly	3	raise-u:lab-abs:	
95	8	律	lyat	4	ly	3	t4:rime:apo:	
95	9	绿	liuk	4	ly	3	t4:apo:lab-abs:	
96	1	居	kio	1	tšy	1a	t1:raise-u:lab-abs:pal:	
96	2	拘	kio	1	tšy	1a	t1:raise-u:lab-abs:pal:	
96	3	驹	kio	1	tšy	1a	t1:raise-u:lab-abs:pal:	
96	4	橘	kyat	4	tšy	1b	t4:rime:Apo:pal:	
96	5	菊	kiuk	4	tšy	1b	t4:Apo:lab-abs:pal:	
96	6	局	giuk	4	tšy	1b	t4:devoi:Apo:lab-abs:pal:	
96	7	举	kio	2	tšy	2	raise-u:lab-abs:pal:	
96	8	据	kio	3	tšy	3	raise-u:lab-abs:pal:	
96	9	锯	kio	3	tšy	3	raise-u:lab-abs:pal:	

pg	co	ch	sme	tn	mb	tn	Diachronic	Allophonic
97	1	巨	gio	2	tśy	3	t2:devoi:raise-u:lab-abs:pal:	
97	1	拒	gio	2	tśy	3	t2:devoi:raise-u:lab-abs:pal:	
97	1	距	gio	2	tśy	3	t2:devoi:raise-u:lab-abs:pal:	
97	2	聚	dzio	2	tśy	3	t2:devoi:raise-u:lab-abs:pal:	
97	3	句	kio	3	tśy	3	raise-u:lab-abs:pal:	
97	4	具	gio	3	tśy	3	devoi:raise-u:lab-abs:pal:	
97	5	剧	giac	4	tśy	3	t4:devoi:vocal:fuse:coal:apo:raise-u:lab-abs:pal:	
97	6	蛆	ts ^h io	1	tś ^h y	1a	t1:raise-u:lab-abs:pal:	
97	7	区	kh ^h io	1	tś ^h y	1a	t1:raise-u:lab-abs:pal:	
97	7	驱	kh ^h io	1	tś ^h y	1a	t1:raise-u:lab-abs:pal:	
97	8	屈	kh ^h yət	4	tś ^h y	1a	t4:rime:apo:pal:	
97	9	曲	kh ^h iuk	4	tś ^h y	1a	t4:apo:pal:	
98	1	曲	kh ^h iuk	4	tś ^h y	1a	t4:apo:pal:	
98	2	渠	gio	1	tś ^h y	1a	asp:t1:devoi:raise-u:lab-abs:pal:	
98	3	取	ts ^h io	2	tś ^h y	2	raise-u:lab-abs:pal:	
98	4	娶	ts ^h io	2	tś ^h y	2	raise-u:lab-abs:pal:	
98	5	去	kh ^h io	3	tś ^h y	3	raise-u:lab-abs:pal:	
98	6	趣	ts ^h io	3	tś ^h y	3	raise-u:lab-abs:pal:	
98	7	嘘	xio	1	śy	1a	t1:raise-u:lab-abs:pal:	
98	7	虚	xio	1	śy	1a	t1:raise-u:lab-abs:pal:	
98	8	需	sio	1	śy	1a	t1:raise-u:lab-abs:pal:	
98	8	须	sio	1	śy	1a	t1:raise-u:lab-abs:pal:	
98	9	须	sio	1	śy	1a	t1:raise-u:lab-abs:pal:	
99	1	徐	zio	1	śy	1b	t1:devoi:raise-u:lab-abs:pal:	
99	2	许	xio	2	śy	2	raise-u:lab-abs:pal:	
99	3	絮	sio	3	śy	3	raise-u:lab-abs:pal:	
99	4	叙	zio	2	śy	3	t2:devoi:raise-u:lab-abs:pal:	
99	4	序	zio	2	śy	3	t2:devoi:raise-u:lab-abs:pal:	
99	4	绪	zio	2	śy	3	t2:devoi:raise-u:lab-abs:pal:	
99	5	续	ziuk	4	śy	3	t4:devoi:apo:lab-abs:	
99	6	婿	siai	3	śy	3	raise-i1:raise-i2:degem:degem:pal:[i-y]:	
99	7	恤	syət	4	śy	3	t4:rime:apo:pal:	
99	8	畜	xiuk	4	śy	3	t4:apo:lab-abs:pal:	
99	8	蓄	xiuk	4	śy	3	t4:apo:lab-abs:pal:	
99	9	鱼	gio	1	y	1b	t1:pro:raise-u:lab-abs:	
100	1	余	io	1	y	1b	t1:raise-u:lab-abs:	
100	1	馀	io	1	y	1b	t1:raise-u:lab-abs:	
100	2	愚	gio	1	y	1b	t1:pro:raise-u:lab-abs:	
100	3	虞	gio	1	y	1b	t1:pro:raise-u:lab-abs:	
100	4	娱	gio	1	y	1b	t1:pro:raise-u:lab-abs:	
100	5	于	jio	1	y	1b	t1:degem:raise-u:lab-abs:	
100	6	孟	jio	1	y	1b	t1:degem:raise-u:lab-abs:	
100	7	榆	io	1	y	1b	t1:raise-u:lab-abs:	
100	8	愉	io	1	y	1b	t1:raise-u:lab-abs:	
100	9	语	gio	2	y	2	pro:raise-u:lab-abs:	
101	1	与	io	2	y	2	raise-u:lab-abs:	
101	2	雨	jio	2	y	2	degem:raise-u:lab-abs:	
101	3	与	io	3	y	3	raise-u:lab-abs:	
101	4	誉	io	3	y	3	raise-u:lab-abs:	
101	4	预	io	3	y	3	raise-u:lab-abs:	
101	5	遇	gio	3	y	3	pro:raise-u:lab-abs:	
101	6	寓	gio	3	y	3	pro:raise-u:lab-abs:	
101	7	裕	io	3	y	3	raise-u:lab-abs:	
101	8	喻	io	3	y	3	raise-u:lab-abs:	
101	9	域	jyək	4	y	3	t4:degem:rime:apo:	
102	1	育	iuk	4	y	3	t4:apo:lab-abs:	
102	2	玉	giuk	4	y	3	t4:pro:apo:lab-abs:	
102	3	玉	giuk	4	y	3	t4:pro:apo:lab-abs:	
102	4	欲	iuk	4	y	3	t4:apo:lab-abs:	
102	5	白	bac	4	pai	1b	t4:devoi:vocal:fuse:apo:	
102	6	摆	pai	2	pai	2	:	
102	7	百	pac	4	pai	2	t4:vocal:fuse:apo:	
102	8	柏	pac	4	pai	2	t4:vocal:fuse:apo:	
102	9	拜	pai	3	pai	3	:	
103	1	裨	bai	3	pai	3	devoi:	
103	2	败	bai	3	pai	3	devoi:	
103	3	拍	p ^h ac	4	p ^h ai	1a	t4:vocal:fuse:apo:	
103	4	排	bai	1	p ^h ai	1b	asp:t1:devoi:	
103	5	牌	bai	1	p ^h ai	1b	asp:t1:devoi:	
103	6	派	p ^h ai	3	p ^h ai	3	:	
103	7	埋	mai	1	mai	1b	t1:	
103	8	买	mai	2	mai	2	:	
103	9	迈	mai	3	mai	3	:	
104	1	迈	mai	3	mai	3	:	
104	2	麦	mac	4	mai	3	t4:vocal:fuse:apo:	
104	3	脉	mac	4	mai	3	t4:vocal:fuse:apo:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
104	4	歹	tai	1	tai	2	A-mg:[t1-t2]:	
104	5	戴	tai	3	tai	3	A-mg:	
104	6	貸	t ^h ai	3	tai	3	A-mg:	
104	7	待	dai	2	tai	3	t2:devoi:a-mg:	
104	8	怠	dai	2	tai	3	t2:devoi:a-mg:	
104	9	代	dai	3	tai	3	devoi:a-mg:	
105	1	袋	dai	3	tai	3	devoi:a-mg:	
105	2	帶	tai	3	tai	3	a-mg:	
105	3	胎	t ^h ai	1	t ^h ai	1a	t1:a-mg:	
105	4	台	dai	1	t ^h ai	1b	asp:t1:devoi:a-mg:	
105	5	抬	dai	1	t ^h ai	1b	asp:t1:devoi:a-mg:	
105	6	态	t ^h ai	3	t ^h ai	3	a-mg:	
105	7	太	t ^h ai	3	t ^h ai	3	a-mg:	
105	8	泰	t ^h ai	3	t ^h ai	3	a-mg:	
105	9	乃	nai	2	nai	2	a-mg:	
106	1	奶	niai	2	nai	2	[i-del]:	
106	2	耐	nai	3	nai	3	a-mg:	
106	3	奈	nai	3	nai	3	a-mg:	
106	4	來	lai	1	lai	1b	t1:a-mg:	
106	5	賴	lai	3	lai	3	a-mg:	
106	6	裁	tsai	1	tsai	1a	t1:a-mg:	
106	6	灾	tsai	1	tsai	1a	t1:a-mg:	
106	7	宰	tsai	2	tsai	2	a-mg:	
106	8	載	tsai	2	tsai	2	a-mg:	
106	9	在	tsai	3	tsai	3	a-mg:	
107	1	載	tsai	3	tsai	3	a-mg:	
107	2	在	dzai	2	tsai	3	t2:devoi:a-mg:	
107	3	猜	ts ^h ai	1	ts ^h ai	1a	t1:a-mg:	
107	4	才	dzai	1	ts ^h ai	1b	asp:t1:devoi:A-mg:	
107	4	財	dzai	1	ts ^h ai	1b	asp:t1:devoi:A-mg:	
107	5	材	dzai	1	ts ^h ai	1b	asp:t1:devoi:A-mg:	
107	6	裁	dzai	1	ts ^h ai	1b	asp:t1:devoi:A-mg:	
107	7	彩	ts ^h ai	2	ts ^h ai	2	a-mg:	
107	7	采	ts ^h ai	2	ts ^h ai	2	a-mg:	
107	7	睬	ts ^h ai	2	ts ^h ai	2	a-mg:	
107	8	菜	ts ^h ai	3	ts ^h ai	3	a-mg:	
107	9	蔡	ts ^h ai	3	ts ^h ai	3	a-mg:	
108	1	腮	sai	1	sai	1a	t1:a-mg:	
108	2	塞	sək	4	sai	1a	t4:[ek-ai]:	
108	3	賽	sai	3	sai	3	a-mg:	
108	4	塞	sai	3	sai	3	a-mg:	
108	5	耑	tʂai	1	tʂai	1a	t1:	
108	6	摘	tʂai	4	tʂai	1a	t4:fric:vocal:fuse:apo:med-a:retrof-2:	
108	7	擇	ɖac	4	tʂai	1b	t4:devoi:fric:vocal:fuse:apo:retrof-2:	
108	8	宅	ɖac	4	tʂai	1b	t4:devoi:fric:vocal:fuse:apo:retrof-2:	
108	9	窄	tʂac	4	tʂai	2	t4:vocal:fuse:apo:	
109	1	債	tʂai	3	tʂai	3	:	
109	2	差	tʂ ^h ai	1	tʂ ^h ai	1a	t1:	
109	3	拆	tʂ ^h ac	4	tʂ ^h ai	1a	t4:fric:vocal:fuse:apo:retrof-2:	
109	4	柴	dz ^h ai	1	tʂ ^h ai	1b	asp:t1:devoi:	
109	5	筛	ʂai	1	ʂai	1a	t1:	
109	6	筛	ʂai	3	ʂai	3	:	
109	7	该	kai	1	kai	1a	t1:a-mg:	
109	8	改	kai	2	kai	2	a-mg:	
109	9	概	kai	3	kai	3	:	
110	1	量	kai	3	kai	3	a-mg:	
110	2	开	k ^h ai	1	k ^h ai	1a	t1:a-mg:	
110	3	揩	k ^h ai	1	k ^h ai	1a	t1:	
110	4	慨	k ^h ai	3	k ^h ai	2	a-mg:[t3-t2]:	
110	5	孩	xai	1	xai	1b	t1:devoi:a-mg:	
110	6	海	xai	2	xai	2	a-mg:	
110	7	害	xai	3	xai	3	devoi:a-mg:	
110	8	哀	?ai	1	ai	1a	t1:pro:a-mg:	
110	9	挨	?ai	1	ai	1b	pro:a-mg:[t1]:	
111	1	矮	?ai	2	ai	2	pro:a-mg:	
111	2	碍	ŋai	3	ai	3	pro:a-mg:	
111	3	艾	ŋai	3	ai	3	pro:a-mg:	
111	4	爱	?ai	3	ai	3	pro:a-mg:	
111	5	衰	ʂui	1	ʂuai	1a	t1:vocal:	
111	6	摔	ʂui	1	ʂuai	1a	t1:[a-ins]:	
111	7	帅	ʂui	3	ʂuai	3	[a-ins]:	
111	8	率	ʂui	3	ʂuai	3	[a-ins]:	
111	9	乖	kuai	1	kuai	1a	t1:	
112	1	拐	kuai	2	kuai	2	:	
112	2	怪	kuai	3	kuai	3	:	
112	3	会	kuai	3	k ^h uai	3	a-mg:[asp]:	
112	4	块	k ^h uai	3	k ^h uai	3	a-mg:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
112	5	快	k ^h uai	3	k ^h uai	3	:	
112	5	快	k ^h uai	3	k ^h uai	3	:	
112	6	怀	yuai	1	xuai	1b	t1:devoi:	
112	7	愧	yuai	1	xuai	1b	t1:devoi:	
112	7	淮	yuai	1	xuai	1b	t1:devoi:	
112	8	坏	yuai	3	xuai	3	devoi:	
112	9	歪	xuai	1	ai	1a	t1:u-del:[x-del]:	
113	1	外	ɣuai	3	ai	3	u-del:pro:a-mg:	
113	2	杯	puai	1	pei	1a	t1:rnd:med-b:	cha:
113	3	碑	pi	1	pei	1a	t1:diph-i:	cha:
113	4	卑	pi	1	pei	1a	t1:diph-i:	cha:
113	5	悲	pi	1	pei	1a	t1:diph-i:	cha:
113	6	北	pək	4	pei	3	t4:[k-i]:	cha:
113	7	贝	pai	3	pei	3	rnd:	cha:
113	8	辈	puai	3	pei	3	rnd:med-b:	cha:
113	9	背	pai	3	pei	3	rnd:	cha:
114	1	悻	buai	2	pei	3	t2:devoi:rnd:med-b:	cha:
114	2	被	bi	2	pei	3	t2:devoi:diph-i:	cha:
114	3	备	bi	3	pei	3	devoi:diph-i:	cha:
114	4	坯	p ^h uai	1	p ^h i	1a	t1:rnd:med-b:[a-del]:	
114	5	培	buai	1	p ^h ei	1b	asp:t1:devoi:rnd:med-b:	cha:
114	6	陪	buai	1	p ^h ei	1b	asp:t1:devoi:rnd:med-b:	cha:
114	6	陪	buai	1	p ^h ei	1b	asp:t1:devoi:rnd:med-b:	cha:
114	7	配	p ^h uai	3	p ^h ei	3	rnd:med-b:	cha:
114	8	佩	buai	3	p ^h ei	3	devoi:rnd:med-b:[asp]:	cha:
114	9	梅	muai	1	mei	1b	t1:rnd:med-b:	cha:
115	1	枚	muai	1	mei	1b	t1:rnd:med-b:	cha:
115	2	媒	muai	1	mei	1b	t1:rnd:med-b:	cha:
115	3	眉	mi	1	mei	1b	t1:diph-i:	cha:
115	4	霉	mi	1	mei	1b	t1:diph-i:	cha:
115	5	每	muai	2	mei	2	rnd:med-b:[!t2]:	cha:
115	6	美	mi	2	mei	2	diph-i:[!t2]:	cha:
115	7	妹	muai	3	mei	3	rnd:med-b:	cha:
115	8	昧	muai	3	mei	3	rnd:med-b:	cha:
115	9	飞	fui	1	fei	1a	t2:diph-i:med-b:	cha:
116	1	非	fui	1	fei	1a	t2:diph-i:med-b:	cha:
116	2	肥	vui	1	fei	1b	t1:devoi:diph-i:med-b:	cha:
116	3	匪	fui	2	fei	2	diph-i:med-b:	cha:
116	4	废	fyai	3	fei	3	raise-i1:raise-i2:degem:med-b:degem:[diph-i]:	cha:
116	5	肺	f ^h yai	3	fei	3	raise-i1:raise-i2:degem:med-b:degem:[diph-i]:[asp-r]:	cha:
116	6	费	f ^h ui	3	fei	3	diph-i:med-b:[asp-r]:	cha:
116	7	内	nuai	3	nei	3	rnd:[u-del]:	cha:
116	8	雷	luai	1	lei	1b	t1:rnd:[u-del]:	cha:
116	9	擂	luai	1	lei	1b	t1:rnd:[u-del]:	cha:
117	1	累	lui	2	lei	2	diph-i:[u-del]:[!t2]:	cha:
117	2	累	lui	3	lei	3	diph-i:[u-del]:	cha:
117	3	类	lui	3	lei	3	diph-i:[u-del]:	cha:
117	4	泪	lui	3	lei	3	diph-i:[u-del]:	cha:
117	5	肋	lək	4	lei	3	t4:[k-i]:	cha:
117	6	贼	dzək	4	tsei	1b	t4:devoi:[k-i]:	cha:
117	7	给	kiəp	4	kei	2	t4:rime:apo:[diph-i]:	cha:
117	8	黑	xək	4	xei	1a	t4:[k-i]:	cha:
117	9	堆	tuai	1	tuei	1a	t1:rnd:	cha:
118	1	对	tuai	3	tuei	3	rnd:	cha:
118	2	队	duai	3	tuei	3	devoi:rnd:	cha:
118	3	兑	duai	3	tuei	3	devoi:rnd:	cha:
118	4	推	t ^h uai	1	t ^h uei	1a	t1:rnd:	cha:
118	5	腿	t ^h uai	2	t ^h uei	2	rnd:	cha:
118	6	退	t ^h uai	3	t ^h uei	3	rnd:	cha:
118	7	咀	tsui	2	tsuei	2	diph-i:	cha:
118	8	罪	dzuai	2	tsuei	3	t2:devoi:rnd:	cha:
118	9	最	tsuai	3	tsuei	3	rnd:	cha:
119	1	醉	tsui	3	tsuei	3	diph-i:	cha:
119	2	催	ts ^h uai	1	ts ^h uei	1a	t1:rnd:	cha:
119	2	崔	ts ^h uai	1	ts ^h uei	1a	t1:rnd:	cha:
119	3	脆	ts ^h yai	3	ts ^h uei	3	raise-i1:Y-spl-d:	cha:
119	4	翠	ts ^h ui	3	ts ^h uei	3	diph-i:	cha:
119	5	虽	sui	1	suei	1a	t1:diph-i:	cha:
119	6	隋	zui	1	suei	1b	t1:devoi:diph-i:	cha:
119	6	随	zui	1	suei	1b	t1:devoi:diph-i:	cha:
119	7	碎	suai	3	suei	3	rnd:	cha:
119	8	岁	syai	3	suei	3	raise-i1:Y-spl-d:	cha:
119	9	穗	zui	3	suei	3	devoi:diph-i:	cha:
120	1	追	tui	1	t ^h suei	1a	t1:fric:diph-i:retrof-l:	cha:
120	2	锥	t ^h sui	1	t ^h suei	1a	t1:diph-i:retrof-l:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
120	3	吹	tsh ^h ui	1	tsh ^h uei	1a	t1:diph-i:retrof-1:	cha:
120	4	炊	tsh ^h ui	1	tsh ^h uei	1a	t1:diph-i:retrof-1:	cha:
120	5	垂	zui	1	tsh ^h uei	1b	affric:asp:t1:devoi:diph-i:retrof-1:	cha:
120	6	捶	dui	1	tsh ^h uei	1b	asp:t1:devoi:fric:diph-i:retrof-1:	cha:
120	6	槌	dui	1	tsh ^h uei	1b	asp:t1:devoi:fric:diph-i:retrof-1:	cha:
120	7	淮	zui	1	suei	1b	affric:asp:t1:devoi:diph-i:retrof-1:	cha:
120	8	水	shui	2	suei	2	diph-i:retrof-1:	cha:
120	9	税	shyai	3	suei	3	raise-il:Y-spl-d:retrof-1:	cha:
121	1	睡	zui	3	suei	3	affric:asp:t1:devoi:diph-i:retrof-1:	cha:
121	2	蕊	riui	2	ruei	2	fric:diph-i:retrof-1:	cha:
121	3	悦	yai	3	ruei	3	raise-il:[y-ru]:	cha:
121	4	规	kui	1	kuei	1a	t1:diph-i:	cha:
121	5	龟	kui	1	kuei	1a	t1:diph-i:	cha:
121	6	归	kui	1	kuei	1a	t1:diph-i:	cha:
121	7	轨	kui	2	kuei	2	diph-i:	cha:
121	8	鬼	kui	2	kuei	2	diph-i:	cha:
121	9	桂	kyai	3	kuei	3	raise-il:y-spl-d:	cha:
122	1	跪	gui	2	kuei	3	t2:devoi:diph-i:	cha:
122	2	柜	gui	3	kuei	3	devoi:diph-i:	cha:
122	3	贵	kui	3	kuei	3	diph-i:	cha:
122	4	亏	k ^h ui	1	k ^h uei	1a	t1:diph-i:	cha:
122	5	葵	gui	1	k ^h uei	1b	asp:t1:devoi:diph-i:	cha:
122	6	愧	kui	3	k ^h uei	3	diph-i:[asp]:	cha:
122	7	灰	xuai	1	xuei	1a	t1:rnd:	cha:
122	8	挥	xui	1	xuei	1a	t1:diph-i:	cha:
122	8	辉	xui	1	xuei	1a	t1:diph-i:	cha:
122	9	徽	xui	1	xuei	1a	t1:diph-i:	cha:
123	1	回	yuai	1	xuei	1b	t1:devoi:rnd:	cha:
123	2	悔	xuai	2	xuei	2	rnd:	cha:
123	3	毁	xui	2	xuei	2	diph-i:	cha:
123	4	贿	xuai	2	xuei	3	rnd:[t2]:	cha:
123	5	汇	yuai	2	xuei	3	t2:devoi:rnd:	cha:
123	6	溃	yuai	3	xuei	3	devoi:rnd:	cha:
123	7	慧	yuai	3	xuei	3	devoi:rnd:	cha:
123	7	会	yuai	3	xuei	3	devoi:rnd:	cha:
123	8	绘	yuai	3	xuei	3	devoi:rnd:	cha:
123	9	惠	yyai	3	xuei	3	devoi:raise-il:y-spl-d:	cha:
124	1	沛	xui	3	xuei	3	diph-i:	cha:
124	2	微	mei	1	uei	1a	lab-dnt':degem:diph-i:[t1]:	cha:
124	3	威	?ui	1	uei	1a	t1:pro:diph-i:	cha:
124	4	槐	mei	1	uei	1b	t1:pro:rnd:	cha:
124	5	危	mei	1	uei	1a	pro:diph-i:[t1]:	cha:
124	6	为	jui	1	uei	1b	t1:degem:diph-i:	cha:
124	7	惟	ui	1	uei	1b	t1:diph-i:	cha:
124	7	维	ui	1	uei	1b	t1:diph-i:	cha:
124	8	违	jui	1	uei	1b	t1:degem:diph-i:	cha:
124	9	围	jui	1	uei	1b	t1:degem:diph-i:	cha:
125	1	委	?ui	2	uei	2	pro:diph-i:	cha:
125	2	伪	mei	3	uei	2	[t3-t2]:pro:diph-i:	cha:
125	3	尾	mei	2	uei	2	lab-dnt':degem:diph-i:[!t2]:	cha:
125	4	伟	jui	2	uei	2	degem:diph-i:[!t2]:	cha:
125	5	苇	jui	2	uei	2	degem:diph-i:	cha:
125	6	卫	jyai	3	uei	3	degem:raise-il:Y-spl-d:	cha:
125	7	喂	?ui	3	uei	3	pro:diph-i:	cha:
125	8	为	jui	3	uei	3	degem:diph-i:	cha:
125	9	位	jui	3	uei	3	degem:diph-i:	cha:
126	1	未	mei	3	uei	3	lab-dnt':degem:diph-i:	cha:
126	2	味	mei	3	uei	3	lab-dnt':degem:diph-i:	cha:
126	3	巍	mei	3	uei	3	pro:diph-i:	cha:
126	4	畏	?ui	3	uei	3	pro:diph-i:	cha:
126	5	胃	jui	3	uei	3	degem:diph-i:	cha:
126	6	谓	jui	3	uei	3	degem:diph-i:	cha:
126	7	谓	jui	3	uei	3	degem:diph-i:	cha:
126	8	裹	pau	1	pau	1a	t1:a-mg:	cha:
126	9	包	pau	1	pau	1a	t1:a-mg:	cha:
127	1	薄	bak	4	pau	1b	t1:devoi:a-mg:[k-u]:	cha:
127	2	雹	bak	4	pau	1b	t1:devoi:[k-u]:	cha:
127	3	宝	pau	2	pau	2	a-mg:	cha:
127	4	堡	pau	2	pau	2	a-mg:	cha:
127	5	堡	pau	2	pau	2	a-mg:	cha:
127	6	抱	pau	2	pau	2	:	cha:
127	7	抱	pau	3	pau	3	a-mg:	cha:
127	8	抱	bau	2	pau	3	t2:devoi:a-mg:	cha:
127	9	暴	bau	3	pau	3	devoi:a-mg:	cha:
128	1	豹	pau	3	pau	3	:	cha:
128	2	爆	pau	3	pau	3	:	cha:
128	3	鲍	bau	2	pau	3	t2:devoi:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
128	4	抛	p ^h au	1	p ^h au	1a	t1:	
128	5	袍	bau	1	p ^h au	1b	asp:t1:devoia-mg:	
128	6	刨	bau	1	p ^h au	1b	asp:t1:devoi:	
128	7	跑	bau	1	p ^h au	2	asp:devoi:[t1-t2]:	
128	8	炮	p ^h au	3	p ^h au	3	:	
128	8	炮	p ^h au	3	p ^h au	3	:	
128	9	猫	mau	1	mau	1a	[t1]:	
129	1	毛	mau	1	mau	1b	t1:a-mg:	
129	2	茅	mau	1	mau	1b	t1:	
129	3	锚	mau	1	mau	1b	t1:	
129	4	牙	miu	1	mau	1b	t1:[i-a]:	
129	5	卯	mau	2	mau	2	:	
129	6	胃	mau	3	mau	3	a-mg:	
129	7	帽	mau	3	mau	3	a-mg:	
129	8	貌	mau	3	mau	3	:	
129	9	茂	mu	3	mau	3	[u-au]:	
130	1	贾	mu	3	mau	3	[u-au]:	
130	2	刀	tau	1	tau	1a	t1:a-mg:	
130	3	岛	tau	2	tau	2	t1:a-mg:	
130	3	捣	tau	2	tau	2	t1:a-mg:	
130	4	神	tau	2	tau	2	t1:a-mg:	
130	5	倒	tau	2	tau	2	t1:a-mg:	
130	6	导	dau	3	tau	2	devoia-mg:[t3-t2]:	
130	7	倒	tau	3	tau	3	a-mg:	
130	8	到	tau	3	tau	3	a-mg:	
130	9	道	dau	2	tau	3	t2:devoia-mg:	
131	1	稻	dau	2	tau	3	t2:devoia-mg:	
131	2	盗	dau	3	tau	3	devoia-mg:	
131	3	滔	t ^h au	1	t ^h au	1a	t1:	
131	4	掏	t ^h au	1	t ^h au	1a	t1:	
131	5	桃	dau	1	t ^h au	1b	asp:t1:devoia-mg:	
131	6	透	dau	1	t ^h au	1b	asp:t1:devoia-mg:	
131	7	胸	dau	1	t ^h au	1b	asp:t1:devoia-mg:	
131	8	讨	t ^h au	2	t ^h au	2	a-mg:	
131	9	套	t ^h au	3	t ^h au	3	a-mg:	
132	1	挠	nau	1	nau	1b	t1:	
132	2	恼	nau	2	nau	2	a-mg:	
132	2	脑	nau	2	nau	2	a-mg:	
132	3	闹	nau	3	nau	3	:	
132	4	捞	lau	1	lau	1a	a-mg:[t1]:	
132	5	劳	lau	1	lau	1b	t1:a-mg:	
132	6	牢	lau	1	lau	1b	t1:a-mg:	
132	7	老	lau	2	lau	2	a-mg:	
132	8	涝	lau	3	lau	3	a-mg:	
132	9	烙	lak	4	lau	3	t4:a-mg:[k-u]:	
132	9	酪	lak	4	lau	3	t4:a-mg:[k-u]:	
133	1	遭	tsau	1	tsau	1a	t1:a-mg:	
133	2	糟	tsau	1	tsau	1a	t1:a-mg:	
133	3	早	tsau	2	tsau	2	a-mg:	
133	4	枣	tsau	2	tsau	2	a-mg:	
133	5	躁	tsau	3	tsau	3	a-mg:	
133	6	灶	tsau	3	tsau	3	a-mg:	
133	7	皂	dzau	2	tsau	3	t2:devoia-mg:	
133	8	造	dzau	2	tsau	3	t2:devoia-mg:	
133	9	操	ts ^h au	1	ts ^h au	1a	t1:a-mg:	
134	1	捷	ts ^h au	3	ts ^h au	1a	a-mg:[t3-t1a]:	
134	2	曹	dzau	1	ts ^h au	1b	asp:t1:devoia-mg:	
134	3	槽	dzau	1	ts ^h au	1b	asp:t1:devoia-mg:	
134	4	草	ts ^h au	2	ts ^h au	2	a-mg:	
134	5	骚	sau	1	sau	1a	t1:a-mg:	
134	6	靡	sau	1	sau	1a	t1:a-mg:	
134	7	扫	sau	2	sau	2	a-mg:	
134	8	嫂	sau	2	sau	2	a-mg:	
134	9	扫	sau	3	sau	3	a-mg:	
135	1	朝	t ^h iau	1	t ^h gau	1a	t1:fric:med-b:retrof-2:	
135	2	昭	t ^h iau	1	t ^h gau	1a	t1:med-a:retrof-2:	
135	3	招	t ^h iau	1	t ^h gau	1a	t1:med-a:retrof-2:	
135	4	肴	diau	4	t ^h gau	1b	t4:devoi:fric:med-a:retrof-2:[k-u]:	
135	5	爪	t ^h gau	2	t ^h gau	2	:	
135	6	找	t ^h gau	2	t ^h gau	2	:	
135	7	罩	t ^h au	3	t ^h gau	3	fric:retrof-1:	
135	8	赵	t ^h iau	2	t ^h gau	3	t2:devoi:fric:med-a:retrof-2:	
135	9	兆	t ^h iau	2	t ^h gau	3	t2:devoi:fric:med-a:retrof-2:	
136	1	照	t ^h iau	3	t ^h gau	3	med-a:retrof-2:	
136	2	抄	t ^h au	1	t ^h au	1a	t1:	
136	3	抄	t ^h au	1	t ^h au	1a	t1:	
136	4	超	t ^h iau	1	t ^h au	1a	t1:fric:med-a:retrof-2:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
136	5	巢	dz,au	1	tʂ ^h au	1b	asp:t1:devoi:	
136	6	潮	ɕiau	1	tʂ ^h au	1b	asp:t1:devoi:fric:med-a:retrof-2:	
136	7	炒	ɕiau	1	tʂ ^h au	1b	asp:t1:devoi:fric:med-a:retrof-2:	
136	8	炒	tʂ ^h au	2	tʂ ^h au	2	:	
136	9	炒	tʂ ^h au	2	tʂ ^h au	2	:	
137	1	梢	ʂau	1	ʂau	1a	t1:	
137	2	梢	ʂau	1	ʂau	1a	t1:	
137	3	稍	ʂau	3	ʂau	1a	[t3-t1a]:	
137	4	烧	ʂiau	1	ʂau	1a	t1:med-a:retrof-2:	
137	5	勺	ziak	4	ʂau	1b	t4:devoi:med-a:retrof-2:[k-u]:	
137	6	少	ʂiau	2	ʂau	2	med-a:retrof-2:	
137	7	少	ʂiau	3	ʂau	3	med-a:retrof-2:	
137	8	绍	ziiau	2	ʂau	3	t2:devoi:med-a:retrof-2:	
137	9	绕	riiau	1	rau	1b	t1:fric:med-a:retrof-2:	
138	1	扰	riiau	2	rau	2	fric:med-a:retrof-2:	
138	2	绕	riiau	2	rau	3	t2:fric:med-a:retrof-2:	
138	3	高	kau	1	kau	1a	t1:a-mg:	
138	4	糕	kau	1	kau	1a	t1:a-mg:	
138	4	羔	kau	1	kau	1a	t1:a-mg:	
138	4	膏	kau	1	kau	1a	t1:a-mg:	
138	5	稿	kau	2	kau	2	a-mg:	
138	6	稿	kau	2	kau	2	:	
138	7	告	kau	3	kau	3	a-mg:	
138	8	考	k ^h au	2	k ^h au	2	a-mg:	
138	9	烤	k ^h au	2	k ^h au	2	a-mg:	
139	1	靠	k ^h au	3	k ^h au	3	a-mg:	
139	2	蒿	xau	1	xau	1a	t1:a-mg:	
139	3	蒿	xau	1	xau	1a	t1:a-mg:	
139	4	毫	ɣau	1	xau	1b	t1:devoi:a-mg:	
139	4	豪	ɣau	1	xau	1b	t1:devoi:a-mg:	
139	5	好	xau	2	xau	2	a-mg:	
139	6	好	xau	3	xau	3	a-mg:	
139	7	耗	xau	3	xau	3	a-mg:	
139	8	号	ɣau	3	xau	3	devoi:a-mg:	
139	9	熬	ŋau	1	au	1b	t1:pro:a-mg:	
140	1	熬	ŋau	1	au	1b	t1:pro:a-mg:	
140	2	袄	?au	2	au	2	pro:a-mg:	
140	3	傲	ŋau	3	au	3	pro:a-mg:	
140	4	标	piau	1	piao	1a	t1:[u-o]:	
140	5	彪	piu	1	piao	1a	t1:[u-a-o]:	
140	6	表	piau	2	piao	2	[u-o]:	
140	7	彪	biau	2	piao	3	t2:devoi:[u-o]:	
140	8	飙	p ^h iau	1	p ^h iao	1a	t1:[u-o]:	
140	9	飙	biau	1	p ^h iao	1b	asp:t1:devoi:[u-o]:	
141	1	缥	biau	1	p ^h iao	1b	asp:t1:devoi:[u-o]:	
141	2	漂	p ^h iau	2	p ^h iao	2	[u-o]:	
141	3	票	p ^h iau	3	p ^h iao	3	[u-o]:	
141	4	漂	p ^h iau	3	p ^h iao	3	[u-o]:	
141	5	苗	miau	1	miau	1b	t1:	
141	6	描	miau	1	miau	1b	t1:	
141	7	秒	miau	2	miau	2	:	
141	8	庙	miau	3	miau	3	:	
141	9	妙	miau	3	miau	3	:	
142	1	刁	tiau	1	tiau	1a	t1:	
142	1	雕	tiau	1	tiau	1a	t1:	
142	2	貂	tiau	1	tiau	1a	t1:	
142	3	吊	tiau	3	tiau	3	:	
142	4	吊	tiau	3	tiau	3	:	
142	5	掉	diau	3	tiau	3	devoi:	
142	6	凋	diau	3	tiau	3	devoi:	
142	7	挑	t ^h iau	1	t ^h iao	1a	t1:	
142	8	条	diau	1	t ^h iao	1b	asp:t1:devoi:	
142	9	调	diau	1	t ^h iao	1b	asp:t1:devoi:	
143	1	眺	t ^h iau	3	t ^h iao	3	:	
143	2	鸟	tiau	2	niau	2	[t-n]:	
143	3	尿	niau	3	niau	3	:	
143	4	聊	liau	1	liau	1b	t1:	
143	4	辽	liau	1	liau	1b	t1:	
143	5	疗	liau	3	liau	1b	[t3-t1b]:	
143	6	了	liau	2	liau	2	:	
143	7	料	liau	3	liau	3	:	
143	8	交	kau	1	tɕiau	1a	t1:sharp:pal:	
143	8	郊	kau	1	tɕiau	1a	t1:sharp:pal:	
143	9	胶	kau	1	tɕiau	1a	t1:sharp:pal:	
144	1	教	kau	1	tɕiau	1a	t1:sharp:pal:	
144	2	焦	tsiau	1	tɕiau	1a	t1:pal:	
144	3	椒	tsiau	1	tɕiau	1a	t1:pal:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
144	3	蕉	tsiau	1	t ^s iau	1a	t1:pal:	
144	4	娇	kiau	1	t ^s iau	1a	t1:pal:	
144	4	桥	kiau	1	t ^s iau	1a	t1:pal:	
144	5	浇	kiau	1	t ^s iau	1a	t1:pal:	
144	6	嚼	dziak	4	t ^s iau	1b	t4:devoi:pal:	
144	7	绞	kau	2	t ^s iau	2	sharp:pal:	
144	8	狡	kau	2	t ^s iau	2	sharp:pal:	
144	9	攮	kau	2	t ^s iau	2	sharp:pal:	
145	1	脚	kiak	4	t ^s iau	2	t4:pal:	
145	2	角	kak	4	t ^s iau	2	t4:sharp:pal:	
145	3	教	kau	3	t ^s iau	3	sharp:pal:	
145	4	酵	kau	3	t ^s iau	3	sharp:pal:	
145	5	校	kau	3	t ^s iau	3	sharp:pal:	
145	6	较	kau	3	t ^s iau	3	sharp:pal:	
145	7	觉	kau	3	t ^s iau	3	sharp:pal:	
145	8	窖	kau	3	t ^s iau	3	sharp:pal:	
145	9	轿	giau	3	t ^s iau	3	devoi:pal:	
146	1	叫	kiau	3	t ^s iau	3	pal:	
146	2	敲	k ^h au	1	t ^s ^h iau	1a	t1:pal:	
146	3	嫩	ts ^h iau	1	t ^s ^h iau	1a	t1:	
146	4	惟	dziak	1	t ^s ^h iau	1b	asp:t1:devoi:	
146	5	瞧	dziak	1	t ^s ^h iau	1b	asp:t1:devoi:	
146	6	侨	giau	1	t ^s ^h iau	1b	asp:t1:devoi:pal:	
146	6	乔	giau	1	t ^s ^h iau	1b	asp:t1:devoi:pal:	
146	7	桥	giau	1	t ^s ^h iau	1b	asp:t1:devoi:pal:	
146	8	巧	k ^h iau	2	t ^s ^h iau	2	pal:	
146	9	鹊	ts ^h iak	4	t ^s ^h iau	2	t4:[k-u]:	
147	1	巧	k ^h iau	3	t ^s ^h iau	3	pal:	
147	2	壳	k ^h ak	4	t ^s ^h iau	3	t4:sharp:pal:	
147	3	宵	siau	1	siau	1a	t1:pal:	
147	3	消	siau	1	siau	1a	t1:pal:	
147	3	稍	siau	1	siau	1a	t1:pal:	
147	3	霄	siau	1	siau	1a	t1:pal:	
147	4	箫	siau	1	siau	1a	t1:pal:	
147	4	霄	siau	1	siau	1a	t1:pal:	
147	5	削	siak	4	siau	1a	t4:[k-u]:	
147	6	小	siau	2	siau	2	pal:	
147	7	晓	xiau	2	siau	2	pal:	
147	8	孝	xau	3	siau	3	sharp:pal:	
147	9	效	yau	3	siau	3	devoi:sharp:pal:	
148	1	校	yau	3	siau	3	devoi:sharp:pal:	
148	2	笑	siau	3	siau	3	pal:	
148	3	妖	?iau	1	iau	1a	t1:pro:	
148	3	要	?iau	1	iau	1a	t1:pro:	
148	3	邀	?iau	1	iau	1a	t1:pro:	
148	4	腰	?iau	1	iau	1a	t1:pro:	
148	5	么	?iau	1	iau	1a	t1:pro:	
148	6	淆	yau	1	iau	1b	t1:sharp:[pro-ɣ]:	
148	6	肴	yau	1	iau	1b	t1:sharp:[pro-ɣ]:	
148	7	摇	iau	1	iau	1b	t1:	
148	7	窑	iau	1	iau	1b	t1:	
148	7	窑	iau	1	iau	1b	t1:	
148	8	姚	iau	1	iau	1b	t1:	
148	8	遥	iau	1	iau	1b	t1:	
148	9	咬	ɲau	2	iau	2	sharp:pro:	
149	1	舀	iau	2	iau	2	:	
149	2	要	?iau	3	iau	3	pro:	
149	3	疍	ɲiak	4	iau	3	t4:pro:[k-u]:	
149	4	药	iak	4	iau	3	t4:[k-u]:	
149	5	朝	iak	4	iau	3	t4:[k-u]:	
149	6	谋	miu	1	mou	1b	t1:lab-dnt':diph-u:	cha:
149	7	某	mu	2	mou	2	diph-u:	cha:
149	8	否	fiu	2	fou	2	lab-dnt':diph-u:	cha:
149	9	都	to	1	tou	1a	t1:raise-u:[diph-u]:	cha:
150	1	兜	tu	1	tou	1a	t1:diph-u:	cha:
150	2	斗	tu	2	tou	2	diph-u:	cha:
150	3	抖	tu	2	tou	2	diph-u:	cha:
150	3	陡	tu	2	tou	2	diph-u:	cha:
150	4	斗	tu	3	tou	3	diph-u:	cha:
150	5	痘	du	3	tou	3	devoi:diph-u:	cha:
150	5	豆	du	3	tou	3	devoi:diph-u:	cha:
150	6	偷	t ^h u	1	t ^h ou	1a	t1:diph-u:	cha:
150	7	头	du	1	t ^h ou	1b	asp:t1:devoi:diph-u:	cha:
150	8	投	du	1	t ^h ou	1b	asp:t1:devoi:diph-u:	cha:
150	9	透	t ^h u	3	t ^h ou	3	diph-u:	cha:
151	1	楼	lu	1	lou	1b	t1:diph-u:	cha:
151	2	楼	lu	1	lou	2	diph-u:[t1-t2]:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
151	3	漏	lu	3	lou	3	diph-u:	cha:
151	4	邹	tgiu	1	tsou	1a	t1:i-del:diph-u:	cha:
151	5	走	tsu	2	tsou	2	diph-u:	cha:
151	6	奏	tsu	3	tsou	3	diph-u:	cha:
151	7	凑	ts ^h u	3	ts ^h ou	3	diph-u:	cha:
151	8	搜	giu	1	sou	1a	t1:i-del:diph-u:	cha:
151	9	搜	giu	1	sou	1a	t1:i-del:diph-u:	cha:
152	1	周	tgiu	1	tgou	1a	t1:med-a:retrof-2:diph-u:	cha:
152	1	州	tgiu	1	tgou	1a	t1:med-a:retrof-2:diph-u:	cha:
152	1	洲	tgiu	1	tgou	1a	t1:med-a:retrof-2:diph-u:	cha:
152	1	舟	tgiu	1	tgou	1a	t1:med-a:retrof-2:diph-u:	cha:
152	2	粥	tgiuk	4	tgou	1a	t4:diph-u:apo:med-a:retrof-2:	cha:
152	3	轴	diuk	4	tgou	1b	t4:devoi:fric:diph-u:apo:med-a:retrof-2:	cha:
152	4	肘	tiu	2	tgou	2	fric:med-a:retrof-2:diph-u:	cha:
152	5	帚	tgiu	2	tgou	2	med-a:retrof-2:diph-u:	cha:
152	6	昼	tiu	3	tgou	3	fric:med-a:retrof-2:diph-u:	cha:
152	7	皱	tgiu	3	tgou	3	i-del:diph-u:	cha:
152	7	约	tgiu	3	tgou	3	i-del:diph-u:	cha:
152	8	骤	dgiu	3	tgou	3	devoi:i-del:diph-u:	cha:
152	9	抽	tgiu	1	ts ^h ou	1a	t1:fric:med-a:retrof-2:diph-u:	cha:
153	1	绸	giu	1	ts ^h ou	1b	aspt1:devoi:fric:med-a:retrof-2:diph-u:	cha:
153	2	稠	giu	1	ts ^h ou	1b	aspt1:devoi:fric:med-a:retrof-2:diph-u:	cha:
153	3	筹	giu	1	ts ^h ou	1b	aspt1:devoi:fric:med-a:retrof-2:diph-u:	cha:
153	4	愁	giu	1	ts ^h ou	1b	aspt1:devoi:fric:med-a:retrof-2:diph-u:	cha:
153	5	仇	giu	1	ts ^h ou	1b	affric:aspt1:devoi:med-a:retrof-2:diph-u:	cha:
153	6	酎	giu	1	ts ^h ou	1b	affric:aspt1:devoi:med-a:retrof-2:diph-u:	cha:
153	7	丑	ts ^h iu	2	ts ^h ou	2	med-a:retrof-2:diph-u:	cha:
153	8	臭	ts ^h iu	3	ts ^h ou	3	med-a:retrof-2:diph-u:	cha:
153	9	收	giu	1	gou	1a	t1:med-a:retrof-2:diph-u:	cha:
154	1	熟	giuk	4	gou	1b	affric:t4:devoi:diph-u:Apo:med-a:retrof-2:	cha:
154	2	手	giu	2	gou	2	med-a:retrof-2:diph-u:	cha:
154	3	守	giu	2	gou	2	med-a:retrof-2:diph-u:	cha:
154	3	首	giu	2	gou	2	med-a:retrof-2:diph-u:	cha:
154	4	瘦	giu	3	gou	3	med-a:retrof-2:diph-u:	cha:
154	5	兽	giu	3	gou	3	med-a:retrof-2:diph-u:	cha:
154	6	受	giu	2	gou	3	affric:t2:devoi:med-a:retrof-2:diph-u:	cha:
154	7	寿	giu	3	gou	3	affric:devoi:med-a:retrof-2:diph-u:	cha:
154	8	授	giu	3	gou	3	affric:devoi:med-a:retrof-2:diph-u:	cha:
154	9	售	giu	3	gou	3	affric:devoi:med-a:retrof-2:diph-u:	cha:
155	1	揉	riu	1	rou	1b	t1:fric:med-a:retrof-2:diph-u:	cha:
155	1	柔	riu	1	rou	1b	t1:fric:med-a:retrof-2:diph-u:	cha:
155	2	肉	riuik	4	rou	3	t1:fric:diph-u:apo:med-a:retrof-2:	cha:
155	3	钩	ku	1	kou	1a	t1:diph-u:	cha:
155	3	勾	ku	1	kou	1a	t1:diph-u:	cha:
155	4	沟	ku	1	kou	1a	t1:diph-u:	cha:
155	5	狗	ku	2	kou	2	diph-u:	cha:
155	6	苟	ku	2	kou	2	diph-u:	cha:
155	7	够	ku	3	kou	3	diph-u:	cha:
155	7	媾	ku	3	kou	3	diph-u:	cha:
155	7	榘	ku	3	kou	3	diph-u:	cha:
155	7	构	ku	3	kou	3	diph-u:	cha:
155	7	购	ku	3	kou	3	diph-u:	cha:
155	8	口	k ^h u	2	k ^h ou	2	diph-u:	cha:
155	9	叩	k ^h u	3	k ^h ou	3	diph-u:	cha:
156	1	扣	k ^h u	3	k ^h ou	3	diph-u:	cha:
156	2	寇	k ^h u	3	k ^h ou	3	diph-u:	cha:
156	3	侯	yu	1	xou	1b	t1:devoi:diph-u:	cha:
156	4	喉	yu	1	xou	1b	t1:devoi:diph-u:	cha:
156	5	猴	yu	1	xou	1b	t1:devoi:diph-u:	cha:
156	6	厚	yu	2	xou	3	t2:devoi:diph-u:	cha:
156	7	后	yu	3	xou	3	devoi:diph-u:	cha:
156	8	候	yu	3	xou	3	devoi:diph-u:	cha:
156	9	欧	yu	1	ou	1a	t1:pro:diph-u:	cha:
157	1	藕	yu	2	ou	2	pro:diph-u:	cha:
157	2	偶	yu	2	ou	2	pro:diph-u:	cha:
157	3	呕	yu	2	ou	2	pro:diph-u:	cha:
157	4	沤	yu	3	ou	3	pro:diph-u:	cha:
157	5	丢	tiu	1	tiou	1a	t1:diph-u:	cha:
157	6	牛	giu	1	niou	1b	t1:diph-u:[ŋ-n]:	cha:
157	7	扭	niu	2	niou	2	diph-u:	cha:
157	7	纽	niu	2	niou	2	diph-u:	cha:
157	8	溜	liu	3	liou	1a	diph-u:[t3-t1a]:	cha:
157	9	流	liu	1	liou	1b	t1:diph-u:	cha:
158	1	琉	liu	1	liou	1b	t1:diph-u:	cha:
158	1	琉	liu	1	liou	1b	t1:diph-u:	cha:
158	2	刘	liu	1	liou	1b	t1:diph-u:	cha:
158	3	留	liu	1	liou	1b	t1:diph-u:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
158	4	榴	liu	1	liou	1b	t1:diph-u:	cha:
158	5	柳	liu	2	liou	2	diph-u:	cha:
158	6	六	liuk	4	liou	3	t4:diph-u:apo:	cha:
158	7	揪	tsiu	1	tšiou	1a	t1:diph-u:[cha]:pal:	
158	8	究	kiu	3	tšiou	1a	diph-u:[cha]:pal:[t3-t1a]:	
158	9	糾	kiu	2	tšiou	1a	diph-u:[cha]:pal:[t2-t1a]:	
159	1	酒	tsiu	2	tšiou	2	diph-u:[cha]:pal:	
159	2	九	kiu	2	tšiou	2	diph-u:[cha]:pal:	
159	3	久	kiu	2	tšiou	2	diph-u:[cha]:pal:	
159	4	灸	kiu	2	tšiou	2	diph-u:[cha]:pal:	
159	5	就	dziu	3	tšiou	3	devoi:diph-u:[cha]:pal:	
159	6	數	kiu	3	tšiou	3	diph-u:[cha]:pal:	
159	7	舅	giu	2	tšiou	3	t2:devoi:diph-u:[cha]:pal:	
159	8	旧	giu	3	tšiou	3	devoi:diph-u:[cha]:pal:	
159	9	秋	ts ^h iu	1	tš ^h iou	1a	t1:diph-u:[cha]:pal:	
160	1	丘	k ^h iu	1	tš ^h iou	1a	t1:diph-u:[cha]:pal:	
160	2	囚	ziu	1	tš ^h iou	1b	asp:t1:devoi:diph-u:[cha]:pal:	
160	3	求	giu	1	tš ^h iou	1b	asp:t1:devoi:diph-u:[cha]:pal:	
160	4	球	giu	1	tš ^h iou	1b	asp:t1:devoi:diph-u:[cha]:pal:	
160	5	修	siu	1	šiou	1a	t1:diph-u:[cha]:pal:	
160	5	羞	siu	1	šiou	1a	t1:diph-u:[cha]:pal:	
160	6	休	xiu	1	šiou	1a	t1:diph-u:[cha]:pal:	
160	7	朽	xiu	2	šiou	2	diph-u:[cha]:pal:	
160	8	秀	siu	3	šiou	3	diph-u:[cha]:pal:	
160	8	绣	siu	3	šiou	3	diph-u:[cha]:pal:	
160	8	锈	siu	3	šiou	3	diph-u:[cha]:pal:	
160	9	袖	ziu	3	šiou	3	devoi:diph-u:[cha]:pal:	
161	1	优	?iu	1	iou	1a	t1:pro:diph-u:	cha:
161	1	忧	?iu	1	iou	1a	t1:pro:diph-u:	cha:
161	2	悠	iu	1	iou	1a	diph-u:[t1]:	cha:
161	3	幽	?iu	1	iou	1a	t1:pro:diph-u:	cha:
161	4	尤	jiu	1	iou	1b	t1:degem:diph-u:	cha:
161	5	邮	jiu	1	iou	1b	t1:degem:diph-u:	cha:
161	6	油	iu	1	iou	1b	t1:diph-u:	cha:
161	6	游	iu	1	iou	1b	t1:diph-u:	cha:
161	6	犹	iu	1	iou	1b	t1:diph-u:	cha:
161	6	由	iu	1	iou	1b	t1:diph-u:	cha:
161	6	游	iu	1	iou	1b	t1:diph-u:	cha:
161	7	有	jiu	2	iou	2	degem:diph-u:	cha:
161	8	友	jiu	2	iou	2	degem:diph-u:	cha:
161	9	又	jiu	3	iou	3	degem:diph-u:	cha:
162	1	右	jiu	3	iou	3	degem:diph-u:	cha:
162	2	佑	jiu	3	iou	3	degem:diph-u:	cha:
162	3	柚	iu	3	iou	3	diph-u:	cha:
162	3	柚	iu	3	iou	3	diph-u:	cha:
162	4	幼	?iu	3	iou	3	pro:diph-u:	cha:
162	5	斑	pan	1	pan	1a	t1:	cha:
162	5	班	pan	1	pan	1a	t1:	cha:
162	5	颁	pan	1	pan	1a	t1:	cha:
162	6	扳	pan	1	pan	1a	t1:	cha:
162	7	般	puan	1	pan	1a	t1:med-b:a-mg:	cha:
162	8	搬	puan	1	pan	1a	t1:med-b:a-mg:	cha:
162	9	板	pan	2	pan	2	:	cha:
162	9	版	pan	2	pan	2	:	cha:
163	1	扮	pan	3	pan	3	:	cha:
163	2	办	ban	3	pan	3	devoi:	cha:
163	3	半	puan	3	pan	3	med-b:a-mg:	cha:
163	4	伴	buan	2	pan	3	t2:devoi:med-b:a-mg:	cha:
163	5	拌	buan	2	pan	3	t2:devoi:med-b:a-mg:	cha:
163	6	攀	p ^h an	1	p ^h an	1a	t1:	cha:
163	7	潘	p ^h uan	1	p ^h an	1a	t1:med-b:a-mg:	cha:
163	8	盘	buan	1	p ^h an	1b	asp:t1:devoi:med-b:a-mg:	cha:
163	9	盼	p ^h an	3	p ^h an	3	:	cha:
164	1	判	p ^h uan	3	p ^h an	3	med-b:a-mg:	cha:
164	2	蛮	man	1	man	1b	t1:	cha:
164	3	瞞	muán	1	man	1b	t1:med-b:a-mg:	cha:
164	4	满	muán	2	man	2	med-b:a-mg:	cha:
164	5	慢	man	3	man	3	:	cha:
164	6	漫	muán	3	man	3	med-b:a-mg:	cha:
164	7	幌	muán	3	man	3	med-b:a-mg:	cha:
164	8	帆	vyam	1	fan	1a	devoi:lab-dnt':fuse:med-b:[t1]:	cha:
164	9	番	f ^h yan	1	fan	1a	t1:lab-dnt':med-b:[asp-r]:	cha:
164	9	翻	f ^h yan	1	fan	1a	t1:lab-dnt':med-b:[asp-r]:	cha:
165	1	凡	vyam	1	fan	1b	t1:devoi:lab-dnt':fuse:med-b:	cha:
165	2	烦	vyan	1	fan	1b	t1:devoi:lab-dnt':med-b:	cha:
165	2	矾	vyan	1	fan	1b	t1:devoi:lab-dnt':med-b:	cha:
165	2	繁	vyan	1	fan	1b	t1:devoi:lab-dnt':med-b:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
165	3	反	fyan	2	fan	2	lab-dnt':med-b:	
165	4	犯	vyam	2	fan	3	t2:devoi:lab-dnt':fuse:med-b:	
165	4	范	vyam	2	fan	3	t2:devoi:lab-dnt':fuse:med-b:	
165	5	贩	fyan	3	fan	3	lab-dnt':med-b:	
165	6	饭	vyam	3	fan	3	devoi:lab-dnt':med-b:	
165	7	担	tam	1	tan	1a	t1:fuse:a-mg:	
165	8	丹	tan	1	tan	1a	t1:a-mg:	
165	9	单	tan	1	tan	1a	t1:a-mg:	
166	1	胆	tam	2	tan	2	fuse:a-mg:	
166	2	担	tam	3	tan	3	fuse:a-mg:	
166	3		dam	3	tan	3	devoi:fuse:a-mg:	
166	4	旦	tan	3	tan	3	a-mg:	
166	5	但	dan	3	tan	3	devoi:a-mg:	
166	5	蛋	dan	3	tan	3	devoi:a-mg:	
166	6	弹	dan	3	tan	3	devoi:a-mg:	
166	7	贪	t ^h am	1	t ^h an	1a	t1:fuse:a-mg:	
166	8	男	t ^h am	1	t ^h an	1a	t1:fuse:a-mg:	
166	9	摊	t ^h an	1	t ^h an	1a	t1:a-mg:	
166	9	滩	t ^h an	1	t ^h an	1a	t1:a-mg:	
167	1	潭	dam	1	t ^h an	1b	asp:t1:devoi:fuse:a-mg:	
167	2	谈	dam	1	t ^h an	1b	asp:t1:devoi:fuse:a-mg:	
167	3	痰	dam	1	t ^h an	1b	asp:t1:devoi:fuse:a-mg:	
167	4	谈	dam	1	t ^h an	1b	asp:t1:devoi:fuse:a-mg:	
167	4	懂	dan	1	t ^h an	1b	asp:t1:devoi:a-mg:	
167	5	弹	dan	1	t ^h an	1b	asp:t1:devoi:a-mg:	
167	6	毯	t ^h am	2	t ^h an	2	fuse:a-mg:	
167	7	坦	t ^h an	2	t ^h an	2	a-mg:	
167	8	探	t ^h am	3	t ^h an	3	fuse:a-mg:	
167	9	炭	t ^h an	3	t ^h an	3	a-mg:	
168	1	叹	t ^h an	3	t ^h an	3	a-mg:	
168	2	男	nam	1	nan	1b	t1:fuse:a-mg:	
168	3	南	nam	1	nan	1b	t1:fuse:a-mg:	
168	4	难	nan	1	nan	1b	t1:a-mg:	
168	5	难	nan	3	nan	3	a-mg:	
168	6	蓝	lam	1	lan	1b	t1:fuse:a-mg:	
168	7	篮	lam	1	lan	1b	t1:fuse:a-mg:	
168	8	栏	lan	1	lan	1b	t1:a-mg:	
168	8	兰	lan	1	lan	1b	t1:a-mg:	
168	9	拦	lan	1	lan	1b	t1:a-mg:	
169	1	览	lam	2	lan	2	fuse:a-mg:	
169	2	揽	lam	2	lan	2	fuse:a-mg:	
169	3	懒	lan	2	lan	2	a-mg:	
169	4	滥	lam	3	lan	3	fuse:a-mg:	
169	5	烂	lan	3	lan	3	a-mg:	
169	6	暂	dzam	3	tsan	3	devoi:fuse:a-mg:	
169	7	赞	tsan	3	tsan	3	a-mg:	
169	8	参	ts ^h am	1	ts ^h an	1a	t1:fuse:a-mg:	
169	9	餐	ts ^h an	1	ts ^h an	1a	t1:a-mg:	
170	1	蚕	dzam	1	ts ^h an	1b	asp:t1:devoi:fuse:a-mg:	
170	2	残	dzan	1	ts ^h an	1b	asp:t1:devoi:a-mg:	
170	3	惨	ts ^h am	2	ts ^h an	2	fuse:a-mg:	
170	4	三	sam	1	san	1a	t1:fuse:a-mg:	
170	5	伞	san	2	san	2	a-mg:	
170	5	散	san	2	san	2	a-mg:	
170	6	散	san	3	san	3	a-mg:	
170	7	沾	t ^h iam	1	t ^h gan	1a	t1:fric:fuse:med-a:retrof-2:	
170	8	粘	t ^h iam	1	t ^h gan	1a	t1:fric:fuse:med-a:retrof-2:	
170	9	毡	ts ^h ian	1	t ^h gan	1a	t1:med-a:retrof-2:	
171	1	斩	ts ^h am	2	t ^h gan	2	fuse:retrof-1:	
171	2	盏	ts ^h an	2	t ^h gan	2	retrof-1:	
171	3	展	t ^h ian	2	t ^h gan	2	fric:med-a:retrof-2:	
171	4	站	t ^h am	3	t ^h gan	3	fric:fuse:retrof-1:	
171	5	蘸	t ^h gam	3	t ^h gan	3	fuse:	
171	6	占	ts ^h iam	3	t ^h gan	3	fuse:med-a:retrof-2:	
171	7	栈	d ^h an	3	t ^h gan	3	devoi:[laffric]:	
171	8	战	ts ^h ian	3	t ^h gan	3	med-a:retrof-2:	
171	9	撵	t ^h iam	1	t ^h gan	1a	t1:fuse:	
172	1	撵	d ^h am	1	t ^h gan	1b	asp:t1:devoi:fuse:	
172	2	撵	t ^h ian	1	t ^h gan	1b	asp:t1:devoi:fric:med-a:retrof-2:	
172	3	撵	t ^h ian	1	t ^h gan	1b	affric:asp:t1:devoi:med-a:retrof-2:	
172	4	铲	t ^h an	2	t ^h gan	2	:	
172	5	产	t ^h an	2	t ^h gan	2	:	
172	6	杉	gam	1	gan	1a	t1:fuse:	
172	7	山	gan	1	gan	1a	t1:	
172	8	扇	t ^h ian	1	gan	1a	t1:med-a:retrof-2:	
172	9	陕	t ^h iam	2	gan	2	fuse:med-a:retrof-2:	
173	1	扇	t ^h ian	3	gan	3	med-a:retrof-2:	
173	2	善	t ^h ian	2	gan	3	t2:devoi:med-a:retrof-2:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
173	3	然	riian	1	ran	1b	t1:fric:med-a:retrof-2:	
173	4	燃	riian	1	ran	1b	t1:fric:med-a:retrof-2:	
173	5	染	riiam	2	ran	2	fric:fuse:med-a:retrof-2:	
173	6	甘	kam	1	kan	1a	t1:fuse:a-mg:	
173	7	柑	kam	1	kan	1a	t1:fuse:a-mg:	
173	8	干	kan	1	kan	1a	t1:a-mg:	
173	9	干	kan	1	kan	1a	t1:a-mg:	
173	9	竿	kan	1	kan	1a	t1:a-mg:	
173	9	肝	kan	1	kan	1a	t1:a-mg:	
174	1	感	kam	2	kan	2	fuse:a-mg:	
174	2	敢	kam	2	kan	2	fuse:a-mg:	
174	3	杆	kan	2	kan	2	a-mg:	
174	4	秆	kan	2	kan	2	a-mg:	
174	5	赶	kan	2	kan	2	a-mg:	
174	6	干	kan	3	kan	3	a-mg:	
174	7	堪	k ^h am	1	k ^h an	1a	t1:fuse:a-mg:	
174	8	刊	k ^h an	1	k ^h an	1a	t1:a-mg:	
174	9	坎	k ^h am	2	k ^h an	2	fuse:a-mg:	
175	1	坎	k ^h am	2	k ^h an	2	fuse:a-mg:	
175	2	看	k ^h an	3	k ^h an	3	fuse:a-mg:	
175	3	含	yam	1	xan	1b	t1:devoi:fuse:a-mg:	
175	4	函	yam	1	xan	1b	t1:devoi:fuse:a-mg:	
175	5	韩	yan	1	xan	1b	t1:devoi:a-mg:	
175	6	韩	yan	1	xan	1b	t1:devoi:a-mg:	
175	7	喊	xam	2	xan	2	fuse:a-mg:	
175	8	汉	xan	3	xan	3	a-mg:	
175	9	旱	yan	2	xan	3	t2:devoi:a-mg:	
176	1	汗	yan	3	xan	3	devoi:a-mg:	
176	2	安	?an	1	an	1a	t1:pro:a-mg:	
176	3	鞍	?an	1	an	1a	t1:pro:a-mg:	
176	4	暗	?am	3	an	3	pro:fuse:a-mg:	
176	5	岸	gan	3	an	3	pro:a-mg:	
176	6	按	?an	3	an	3	pro:a-mg:	
176	7	案	?an	3	an	3	pro:a-mg:	
176	8	鞭	pian	1	pian	1a	t1:	
176	9	编	pian	1	pian	1a	t1:	
177	1	边	pian	1	pian	1a	t1:	
177	2	贬	piam	2	pian	2	fuse:	
177	3	匾	pian	2	pian	2	:	
177	3	扁	pian	2	pian	2	:	
177	4	变	pian	3	pian	3	:	
177	5	辨	bian	2	pian	3	t2:devoi:	
177	5	辨	bian	2	pian	3	t2:devoi:	
177	6	便	bian	3	pian	3	devoi:	
177	7	辨	bian	2	pian	3	t2:devoi:	
177	8	遍	pian	3	pian	3	devoi:	
177	9	遍	pian	3	pian	3	devoi:	
178	1	偏	p ^h ian	1	p ^h ian	1a	t1:	
178	1	篇	p ^h ian	1	p ^h ian	1a	t1:	
178	2	便	bian	1	p ^h ian	1b	asp:t1:devoi:	
178	3	骗	p ^h ian	3	p ^h ian	3	:	
178	4	片	p ^h ian	3	p ^h ian	3	:	
178	5	棉	mian	1	mian	1b	t1:	
178	5	绵	mian	1	mian	1b	t1:	
178	6	眠	mian	1	mian	1b	t1:	
178	7	免	mian	2	mian	2	:	
178	7	勉	mian	2	mian	2	:	
178	8	面	mian	3	mian	3	:	
178	9	顺	tian	1	tian	1a	t1:	
179	1	点	tiam	2	tian	2	fuse:	
179	2	典	tian	2	tian	2	:	
179	3	店	tiam	3	tian	3	fuse:	
179	4	电	dian	3	tian	3	devoi:	
179	5	殿	dian	3	tian	3	devoi:	
179	6	佃	dian	3	tian	3	devoi:	
179	7	添	t ^h iam	1	t ^h ian	1a	t1:fuse:	
179	8	天	t ^h ian	1	t ^h ian	1a	t1:	
179	9	甜	diam	1	t ^h ian	1b	asp:t1:devoi:fuse:	
180	1	田	dian	1	t ^h ian	1b	asp:t1:devoi:	
180	2	填	dian	1	t ^h ian	1b	asp:t1:devoi:	
180	3	舔	t ^h iam	2	t ^h ian	2	fuse:	
180	4	黏	niam	1	nian	1b	t1:fuse:	
180	5	拈	niam	1	nian	1a	fuse:[t1]:	
180	6	年	nian	1	nian	1b	t1:	
180	7	碾	nian	2	nian	2	:	
180	8	辇	lian	2	nian	2	[l-n]:	
180	9	撵	nian	2	nian	2	:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
181	1	念	niam	3	nian	3	fuse:	
181	2	帘	liam	1	lian	1b	tl:fuse:	
181	2	廉	liam	1	lian	1b	tl:fuse:	
181	2	帘	liam	1	lian	1b	tl:fuse:	
181	2	簾	liam	1	lian	1b	tl:fuse:	
181	3	连	lian	1	lian	1b	tl:	
181	4	联	lian	1	lian	1b	tl:	
181	5	怜	lian	1	lian	1b	tl:	
181	6	莲	lian	1	lian	1b	tl:	
181	7	脸	kiam	2	lian	2	fuse:[k-l]:	
181	8	敛	liam	3	lian	3	fuse:	
181	9	炼	lian	3	lian	3	:	
181	9	练	lian	3	lian	3	:	
181	9	炼	lian	3	lian	3	:	
182	1	恋	luan	3	lian	3	[u-i]:	
182	2	监	kiam	1	tsian	1a	tl:fuse:pal:	
182	3	尖	tsiam	1	tsian	1a	tl:fuse:pal:	
182	4	歼	tsiam	1	tsian	1a	tl:fuse:pal:	
182	5	兼	kiam	1	tsian	1a	tl:fuse:pal:	
182	6	间	kan	1	tsian	1a	tl:[i-ins]:pal:	
182	7	艰	kan	1	tsian	1a	tl:[i-ins]:pal:	
182	8	奸	kan	1	tsian	1a	tl:[i-ins]:pal:	
182	8	奸	kan	1	tsian	1a	tl:[i-ins]:pal:	
182	9	煎	tsian	1	tsian	1a	tl:pal:	
183	1	筵	tsian	1	tsian	1a	tl:pal:	
183	2	肩	kian	1	tsian	1a	tl:pal:	
183	3	坚	kian	1	tsian	1a	tl:pal:	
183	4	减	kam	2	tsian	2	sharp:fuse:pal:	
183	5	碱	kam	2	tsian	2	sharp:fuse:pal:	
183	6	俭	kiam	2	tsian	2	fuse:pal:	
183	7	俭	giam	2	tsian	2	devoi:fuse:pal:[t2]:	
183	8	柬	kan	2	tsian	2	sharp:pal:	
183	8	简	kan	2	tsian	2	sharp:pal:	
183	9	捻	kan	2	tsian	2	sharp:pal:	
184	1	剪	tsian	2	tsian	2	pal:	
184	2	茧	kian	2	tsian	2	pal:	
184	3	渐	dziam	2	tsian	3	t2:devoi:fuse:pal:	
184	4	剑	kiam	3	tsian	3	fuse:pal:	
184	5	间	kan	3	tsian	3	sharp:pal:	
184	6	涧	kan	3	tsian	3	sharp:pal:	
184	7	箭	tsian	3	tsian	3	pal:	
184	8	贱	tsian	3	tsian	3	pal:	
184	9	贱	dzian	2	tsian	3	t2:devoi:pal:	
185	1	件	gian	2	tsian	3	t2:devoi:pal:	
185	2	建	kian	3	tsian	3	pal:	
185	3	健	gian	3	tsian	3	devoi:pal:	
185	4	腱	gian	3	tsian	3	devoi:pal:	
185	5	荐	tsian	3	tsian	3	pal:	
185	6	见	kian	3	tsian	3	pal:	
185	7	筵	ts ^h iam	1	ts ^h ian	1a	tl:fuse:pal:	
185	7	筵	ts ^h iam	1	ts ^h ian	1a	tl:fuse:pal:	
185	8	谦	k ^h iam	1	ts ^h ian	1a	tl:fuse:pal:	
185	9	迁	ts ^h ian	1	ts ^h ian	1a	tl:pal:	
186	1	千	ts ^h ian	1	ts ^h ian	1a	tl:pal:	
186	2	牵	k ^h ian	1	ts ^h ian	1a	tl:pal:	
186	3	铅	yan	1	ts ^h ian	1a	tl:sharp:pal:[y-k]:[asp]:	
186	4	钳	giam	1	ts ^h ian	1b	asp:tl:devoi:fuse:pal:	
186	5	钱	dzian	1	ts ^h ian	1b	asp:tl:devoi:fuse:pal:	
186	6	前	dzian	1	ts ^h ian	1b	asp:tl:devoi:fuse:pal:	
186	7	乾	gian	1	ts ^h ian	1b	asp:tl:devoi:fuse:pal:	
186	8	虔	gian	1	ts ^h ian	1b	asp:tl:devoi:fuse:pal:	
186	9	浅	ts ^h ian	2	ts ^h ian	2	pal:	
187	1	欠	k ^h iam	3	ts ^h ian	3	fuse:pal:	
187	2	仙	sian	1	sian	1a	tl:pal:	
187	3	鲜	sian	1	sian	1a	tl:pal:	
187	4	掀	xian	1	sian	1a	tl:pal:	
187	5	先	sian	1	sian	1a	tl:pal:	
187	6	咸	yam	1	sian	1b	tl:devoi:sharp:fuse:pal:	
187	7	衍	yam	1	sian	1b	tl:devoi:sharp:fuse:pal:	
187	8	嫌	yiam	1	sian	1b	tl:devoi:fuse:pal:	
187	9	闲	yan	1	sian	1b	tl:devoi:sharp:pal:	
187	9	闲	yan	1	sian	1b	tl:devoi:sharp:pal:	
188	1	贤	yian	1	sian	1b	tl:devoi:pal:	
188	2	险	xiam	2	sian	2	fuse:pal:	
188	3	显	xian	2	sian	2	pal:	
188	4	陷	yam	3	sian	3	devoi:sharp:fuse:pal:	
188	5	馅	yam	3	sian	3	devoi:sharp:fuse:pal:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
188	6	限	yan	2	śian	3	t2:devoi:sharp:fuse:pal:	
188	7	线	sian	3	śian	3	pal:	
188	8	宪	xian	3	śian	3	pal:	
188	9	献	xian	3	śian	3	pal:	
189	1	现	ȳian	3	śian	3	devoi:pal:	
189	2	县	ȳyan	3	śian	3	devoi:y-spl-d:pal:	
189	3	淹	?iam	1	ian	1a	t1:pro:fuse:	
189	4	腌	?iam	1	ian	1a	t1:pro:fuse:	
189	5	烟	?ian	1	ian	1a	t1:pro:	
189	6	岩	ɣam	1	ian	1b	t1:sharp:pro:fuse:	
189	7	盐	iam	1	ian	1b	t1:fuse:	
189	8	檐	iam	1	ian	1b	t1:fuse:	
189	9	阎	iam	1	ian	1b	t1:fuse:	
190	1	严	ɣiam	1	ian	1b	t1:pro:fuse:	
190	2	颜	ɣan	1	ian	1b	t1:sharp:pro:fuse:	
190	3	延	ian	1	ian	1b	t1:	
190	4	言	ɣian	1	ian	1b	t1:pro:	
190	5	研	ɣian	1	ian	1b	t1:pro:	
190	6	掩	?iam	2	ian	2	pro:fuse:	
190	7	眼	ɣan	2	ian	2	sharp:pro:	
190	8	演	ian	2	ian	2	:	
190	9	验	ɣiam	3	ian	3	pro:fuse:	
191	1	厌	?iam	3	ian	3	pro:fuse:	
191	2	雁	ɣan	3	ian	3	sharp:pro:fuse:	
191	3	晏	?an	3	ian	3	sharp:pro:fuse:	
191	4	观	ɣian	3	ian	3	pro:fuse:	
191	5	燕	?ian	3	ian	3	pro:fuse:	
191	6	咽	?ian	3	ian	3	pro:fuse:	
191	7	宴	?ian	3	ian	3	pro:fuse:	
191	8	端	tuan	1	tuan	1a	t1:a-mg:	
191	9	短	tuan	2	tuan	2	a-mg:	
192	1	断	tuan	3	tuan	3	a-mg:	
192	2	断	tuan	3	tuan	3	a-mg:	
192	3	断	duan	2	tuan	3	t2:devoi:a-mg:	
192	4	段	duan	3	tuan	3	devoi:a-mg:	
192	5	段	duan	3	tuan	3	devoi:a-mg:	
192	6	团	duan	1	t ^h uan	1b	asp:t1:devoi:a-mg:	
192	7	暖	nuan	2	nuan	2	a-mg:	
192	8	卵	luan	2	luan	2	a-mg:	
192	9	乱	luan	3	luan	3	a-mg:	
193	1	钻	tsuan	1	tsuan	1a	t1:a-mg:	
193	2	钻	tsuan	3	tsuan	3	a-mg:	
193	3	酸	suan	1	suan	1a	t1:a-mg:	
193	4	算	suan	3	suan	3	a-mg:	
193	4	蒜	suan	3	suan	3	a-mg:	
193	5	专	tśyan	1	tśuan	1a	t1:med-a:retrof-2:	
193	5	砖	tśyan	1	tśuan	1a	t1:med-a:retrof-2:	
193	6	转	tʃyan	2	tśuan	2	fric:med-a:retrof-2:	
193	7	赚	dam	3	tśuan	3	devoi:fric:fuse:retrof-1:[u-ins-a]:	
193	8	转	tuan	3	tśuan	3	fric:retrof-1:	
193	9	篆	dyan	2	tśuan	3	t2:devoi:fric:med-a:retrof-2:	
194	1	传	dyan	3	tśuan	3	devoi:fric:med-a:retrof-2:	
194	2	川	tś ^h yan	1	tś ^h uan	1a	t1:med-a:retrof-2:	
194	3	穿	tś ^h yan	1	tś ^h uan	1a	t1:med-a:retrof-2:	
194	4	传	dyan	1	tś ^h uan	1b	asp:t1:fric:med-a:retrof-2:	
194	5	椽	dyan	1	tś ^h uan	1b	asp:t1:fric:med-a:retrof-2:	
194	6	船	dzyan	1	tś ^h uan	1b	affric:asp:t1:devoi:med-a:retrof-2:	
194	7	喘	tś ^h yan	2	tś ^h uan	2	med-a:retrof-2:	
194	8	串	tś ^h uan	3	tś ^h uan	3	A-mg:	
194	9	铜	tś ^h yan	3	tś ^h uan	3	med-a:retrof-2:	
195	1	闫	ɣuan	1	ɣuan	1a	t1:	
195	2	拴	gyan	1	ɣuan	1a	t1:i-del:	
195	3	软	nyan	2	ruan	2	fric:med-a:retrof-2:	
195	4	官	kuan	1	kuan	1a	t1:a-mg:	
195	5	观	kuan	1	kuan	1a	t1:a-mg:	
195	6	冠	kuan	1	kuan	1a	t1:a-mg:	
195	7	关	kuan	1	kuan	1a	t1:a-mg:	
195	8	管	kuan	2	kuan	2	a-mg:	
195	8	馆	kuan	2	kuan	2	a-mg:	
195	9	罐	kuan	3	kuan	3	a-mg:	
195	9	贯	kuan	3	kuan	3	a-mg:	
196	1	灌	kuan	3	kuan	3	a-mg:	
196	2	观	kuan	3	kuan	3	a-mg:	
196	3	冠	kuan	3	kuan	3	a-mg:	
196	4	惯	kuan	3	kuan	3	:	
196	5	宽	k ^h uan	1	k ^h uan	1a	t1:a-mg:	
196	6	款	k ^h uan	2	k ^h uan	2	a-mg:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
196	7	欢	xuan	1	xuan	1a	t1:a-mg:	
196	8	还	yuan	1	xuan	1b	t1:devoi:	
196	9	环	yuan	1	xuan	1b	t1:devoi:	
197	1	缓	yuan	2	xuan	2	devoi:a-mg:[!t2]:	
197	2	唤	xuan	3	xuan	3	a-mg:	
197	2	换	xuan	3	xuan	3	a-mg:	
197	3	换	yuan	3	xuan	3	devoi:a-mg:	
197	4	患	yuan	3	xuan	3	devoi:a-mg:	
197	5	宦	yuan	3	xuan	3	devoi:a-mg:	
197	6	豌	?uan	1	uan	1a	t1:pro:	
197	7	弯	?uan	1	uan	1a	t1:pro:	
197	7	湾	?uan	1	uan	1a	t1:pro:	
197	8	完	yuan	1	uan	1b	t1:a-mg:[pro-y]:	
197	9	丸	yuan	1	uan	1b	t1:a-mg:[pro-y]:	
198	1	玩	guan	1	uan	1b	t1:pro:	
198	2	顽	guan	1	uan	1b	t1:pro:	
198	3	碗	?uan	2	uan	2	pro:a-mg:	
198	4	晚	nyan	2	uan	2	lab-dnt':degem:[!t2]:	
198	5	挽	nyan	2	uan	2	lab-dnt':degem:[!t2]:	
198	6	万	nyan	3	uan	3	lab-dnt':degem:	
198	7	蔓	nyan	3	uan	3	lab-dnt':degem:	
198	8	捐	yan	1	tsyan	1a	t1:[k-ins]:pal:	
198	9	卷	kyan	2	tsyan	2	pal:	
199	1	眷	kyan	3	tsyan	3	pal:	
199	2	卷	kyan	3	tsyan	3	pal:	
199	3	绢	kyan	3	tsyan	3	pal:	
199	4	倦	gyan	3	tsyan	3	devoi:pal:	
199	5	圈	gyan	2	tsyan	3	t2:devoi:pal:	
199	6	圈	k ^h yan	1	ts ^h yan	1a	t1:pal:	
199	7	泉	dzyan	1	ts ^h yan	1b	asp:t1:devoi:pal:	
199	8	全	dzyan	1	ts ^h yan	1b	asp:t1:devoi:pal:	
199	9	权	gyan	1	ts ^h yan	1b	asp:t1:devoi:pal:	
200	1	拳	gyan	1	ts ^h yan	1b	asp:t1:devoi:pal:	
200	2	犬	k ^h yan	2	ts ^h yan	2	pal:	
200	3	劝	k ^h yan	3	ts ^h yan	3	pal:	
200	4	宣	syau	1	syau	1a	t1:pal:	
200	5	旋	zuan	1	syau	1b	t1:devoi:[u-y]:pal:	
200	6	玄	yuan	1	syau	1b	t1:devoi:pal:	
200	7	悬	yuan	1	syau	1b	t1:devoi:pal:	
200	8	癣	sian	2	syau	2	pal:[i-y]:	
200	9	选	syau	2	syau	2	pal:	
201	1	揅	xyan	3	syau	3	pal:	
201	2	冤	?yan	1	yan	1a	t1:pro:	
201	3	员	jyan	1	yan	1b	t1:degem:	
201	4	圆	jyan	1	yan	1b	t1:degem:	
201	5	缘	yan	1	yan	1b	t1:	
201	6	原	nyan	1	yan	1b	t1:pro:	
201	6	源	nyan	1	yan	1b	t1:pro:	
201	7	元	nyan	1	yan	1b	t1:pro:	
201	8	袁	jyan	1	yan	1b	t1:[pro-j]:	
201	8	辕	jyan	1	yan	1b	t1:[pro-j]:	
201	9	园	jyan	1	yan	1b	t1:[pro-j]:	
202	1	援	jyan	1	yan	1b	t1:[pro-j]:	
202	2	远	jyan	2	yan	2	[pro-j]:	
202	3	院	jyan	3	yan	3	[pro-j]:	
202	4	怨	nyan	3	yan	3	pro:	
202	5	怨	?yan	3	yan	3	pro:	
202	6	奔	puən	1	pən	1a	t1:med-b:	cha:
202	7	本	puən	2	pən	2	med-b:[!rime]:	cha:
202	8	笨	buən	2	pən	3	t2:devoi:med-b:[!rime]:	cha:
202	9	喷	p ^h uən	1	p ^h ən	1a	t1:med-b:	cha:
203	1	盆	buən	1	p ^h ən	1b	asp:t1:med-b:	cha:
203	2	门	muən	1	mən	1b	t1:med-b:	cha:
203	3	网	muən	3	mən	3	med-b:	cha:
203	4	分	fyən	1	fən	1a	t1:lab-dnt':med-b:	cha:
203	5	芬	fh ^h yən	1	fən	1a	t1:lab-dnt':med-b:[asp-r]:	cha:
203	5	芬	fh ^h yən	1	fən	1a	t1:lab-dnt':med-b:[asp-r]:	cha:
203	6	粉	vyən	1	fən	1b	t1:devoi:lab-dnt':med-b:	cha:
203	7	粪	fyən	2	fən	2	lab-dnt':med-b:	cha:
203	8	粪	fyən	3	fən	3	lab-dnt':med-b:	cha:
203	9	奋	fyən	3	fən	3	lab-dnt':med-b:	cha:
204	1	愤	vyən	2	fən	3	t2:devoi:lab-dnt':med-b:	cha:
204	2	忿	vyən	2	fən	3	t2:devoi:lab-dnt':med-b:	cha:
204	3	份	vyən	3	fən	3	t2:devoi:lab-dnt':med-b:	cha:
204	4	嫩	nuən	3	nən	3	[u-del]:	cha:
204	5	参	t ^h iəm	1	ts ^h ən	1a	t1:fuse:i-del:deret:	cha:
204	6	森	giəm	1	sən	1a	t1:fuse:i-del:deret:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
204	7	針	tɕiəm	1	tɕən	1a	t1:fuse:med-a:retrof-2:	cha:
204	8	針	tɕiəm	1	tɕən	1a	t1:fuse:med-a:retrof-2:	cha:
204	9	珍	ʃiən	1	tɕən	1a	t1:fric:med-a:retrof-2:	cha:
205	1	貞	ʃiən	1	tɕən	1a	t1:fric:med-a:retrof-2:[n'-n]:	cha:
205	2	偵	tʰiən	1	tɕən	1a	t1:fric:med-a:retrof-2:[n'-n]:[de-asp]:	cha:
205	3	枕	tɕiəm	2	tɕən	2	fuse:med-a:retrof-2:	cha:
205	4	診	tɕiən	2	tɕən	2	med-a:retrof-2:	cha:
205	5	疹	tɕiən	2	tɕən	2	med-a:retrof-2:	cha:
205	6	鎮	ʃiən	3	tɕən	3	fric:med-a:retrof-2:	cha:
205	7	陣	ʃiən	3	tɕən	3	devoi:fric:med-a:retrof-2:	cha:
205	8	震	tɕiən	3	tɕən	3	med-a:retrof-2:	cha:
205	8	震	tɕiən	3	tɕən	3	med-a:retrof-2:	cha:
205	9	沉	ʃiəm	1	tɕʰən	1b	asp:t1:devoi:fric:fuse:med-a:retrof-2:	cha:
206	1	陳	ʃiən	1	tɕʰən	1b	asp:t1:devoi:fric:med-a:retrof-2:	cha:
206	2	塵	ʃiən	1	tɕʰən	1b	asp:t1:devoi:fric:med-a:retrof-2:	cha:
206	3	屨	ʃiən	1	tɕʰən	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
206	3	屨	ʃiən	1	tɕʰən	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
206	4	臣	ʃiən	1	tɕʰən	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
206	5	陛	tʰiən	3	tɕʰən	3	fric:med-a:retrof-2:	cha:
206	6	村	tɕʰiən	3	tɕʰən	3	i-del:	cha:
206	7	称	tɕʰiəŋ	3	tɕʰən	3	med-a:retrof-2:[ŋ-n]:	cha:
206	8	參	ʃiəm	1	ʃən	1a	t1:fuse:i-del:	cha:
206	9	深	ʃiəm	1	ʃən	1a	t1:fuse:med-a:retrof-2:	cha:
207	1	申	ʃiən	1	ʃən	1a	t1:med-a:retrof-2:	cha:
207	1	身	ʃiən	1	ʃən	1a	t1:med-a:retrof-2:	cha:
207	2	伸	ʃiən	1	ʃən	1a	t1:med-a:retrof-2:	cha:
207	3	紳	dʒiən	1	ʃən	1b	affric:t1:med-a:retrof-2:	cha:
207	4	紳	ʃiəm	2	ʃən	2	fuse:med-a:retrof-2:	cha:
207	5	审	ʃiəm	2	ʃən	2	fuse:med-a:retrof-2:	cha:
207	6	沈	ʃiəm	2	ʃən	2	fuse:med-a:retrof-2:	cha:
207	7	涉	ʃiəm	3	ʃən	3	fuse:med-a:retrof-2:	cha:
207	8	甚	ʃiəm	2	ʃən	3	affric:t2:devoi:fuse:med-a:retrof-2:	cha:
207	9	胃	ʃiən	2	ʃən	3	affric:t2:devoi:med-a:retrof-2:	cha:
208	1	慎	ʃiən	3	ʃən	3	affric:devoi:med-a:retrof-2:	cha:
208	2	人	ʃiən	1	rən	1b	t1:fric:med-a:retrof-2:	cha:
208	3	仁	ʃiən	1	rən	1b	t1:fric:med-a:retrof-2:	cha:
208	4	忍	ʃiən	2	rən	2	fric:med-a:retrof-2:	cha:
208	5	性	ʃiəm	3	rən	3	fric:fuse:med-a:retrof-2:	cha:
208	6	刃	ʃiən	3	rən	3	fric:med-a:retrof-2:	cha:
208	7	认	ʃiən	3	rən	3	fric:med-a:retrof-2:	cha:
208	8	跟	kən	1	kən	1a	t1:	cha:
208	9	根	kən	1	kən	1a	t1:	cha:
209	1	根	kʰən	2	kʰən	2	cha:	cha:
209	1	恩	kʰən	2	kʰən	2	cha:	cha:
209	2	肯	kʰəŋ	2	kʰən	2	[ŋ-n]:	cha:
209	3	很	ɣən	2	xən	2	devoi:	cha:
209	4	恨	ɣən	3	xən	3	devoi:	cha:
209	5	恩	ʔən	1	ən	1a	t1:pro:	cha:
209	6	寔	piən	1	pin	1a	t1:rime:	cha:
209	7	寔	piən	3	pin	3	rime:	cha:
209	8	寔	piən	3	pin	1a	:	cha:
209	9	寔	biən	1	pʰin	1b	asp:t1:devoi:rime:	cha:
210	1	頻	biən	1	pʰin	1b	asp:t1:devoi:rime:	cha:
210	2	品	pʰiəm	2	pʰin	2	rime:fuse:	cha:
210	3	聘	pʰiən	3	pʰiən	3	vocal:fuse:coal:[cha]:rime:	cha:
210	4	民	miən	1	min	1b	t1:rime:	cha:
210	5	敏	miən	2	min	2	rime:	cha:
210	6	林	liəm	1	lin	1b	rime:fuse:	cha:
210	6	淋	liəm	1	lin	1b	rime:fuse:	cha:
210	6	臨	liəm	1	lin	1b	rime:fuse:	cha:
210	7	邻	liən	1	lin	1b	rime:	cha:
210	7	邻	liən	1	lin	1b	t1:rime:	cha:
210	8	鱗	liən	1	lin	1b	t1:rime:	cha:
210	9	赁	niəm	3	lin	3	rime:fuse:[n-l]:	cha:
211	1	吝	liən	3	lin	3	rime:	cha:
211	2	津	tɕiən	1	tɕin	1a	t1:rime:pal:	cha:
211	3	今	kiəm	1	tɕin	1a	t1:rime:fuse:pal:	cha:
211	4	金	kiəm	1	tɕin	1a	t1:rime:fuse:pal:	cha:
211	5	襟	kiəm	1	tɕin	1a	t1:rime:fuse:pal:	cha:
211	6	巾	kiəm	1	tɕin	1a	t1:rime:fuse:pal:	cha:
211	7	巾	kiən	1	tɕin	1a	t1:rime:pal:	cha:
211	8	斤	kiən	1	tɕin	1a	t1:rime:pal:	cha:
211	8	筋	kiən	1	tɕin	1a	t1:rime:pal:	cha:
211	9	尽	tɕiən	2	tɕin	2	rime:pal:	cha:
212	1	察	kiən	2	tɕin	2	rime:pal:	cha:
212	2	仅	giən	3	tɕin	2	devoi:rime:pal:[t3-t2]:	cha:
212	3	察	kiəm	3	tɕin	3	rime:fuse:pal:	cha:

pg	co	ch	sme	tn	mb	tn	Diachronic	Allophonic
212	4	进	tsiən	3	tʂin	3	rime:pal:	
212	5	尽	dziən	2	tʂin	3	t2:devoi:rime:pal:	
212	6	近	giən	2	tʂin	3	t2:devoi:rime:pal:	
212	7	劲	kiən	3	tʂin	3	rime:pal:	
212	8	侵	ts ^h iəm	1	tʂ ^h in	1a	t1:rime:fuse:pal:	
212	9	亲	ts ^h iən	1	tʂ ^h in	1a	t1:rime:pal:	
213	1	擒	giəm	1	tʂ ^h in	1b	asp:t1:devoi:rime:fuse:pal:	
213	1	琴	giəm	1	tʂ ^h in	1b	asp:t1:devoi:rime:fuse:pal:	
213	1	禽	giəm	1	tʂ ^h in	1b	asp:t1:devoi:rime:fuse:pal:	
213	2	秦	dziən	1	tʂ ^h in	1b	asp:t1:devoi:rime:pal:	
213	3	勤	giən	1	tʂ ^h in	1b	asp:t1:devoi:rime:pal:	
213	3	芹	giən	1	tʂ ^h in	1b	asp:t1:devoi:rime:pal:	
213	4	寝	ts ^h iəm	2	tʂ ^h in	2	rime:pal:	
213	5	心	siəm	1	ʂin	1a	t1:rime:fuse:pal:	
213	6	新	siən	1	ʂin	1a	t1:rime:pal:	
213	6	薪	siən	1	ʂin	1a	t1:rime:pal:	
213	6	辛	siən	1	ʂin	1a	t1:rime:pal:	
213	7	欣	xiən	1	ʂin	1a	t1:rime:pal:	
213	8	信	siən	3	ʂin	3	rime:pal:	
213	9	音	ʔiəm	1	in	1a	t1:pro:rime:fuse:	
214	1	阴	ʔiəm	1	in	1a	t1:pro:rime:fuse:	
214	2	因	ʔiən	1	in	1a	t1:pro:rime:	
214	3	姻	ʔiən	1	in	1a	t1:pro:rime:	
214	4	殷	ʔiən	1	in	1a	t1:pro:rime:	
214	5	银	ŋiən	1	in	1b	t1:pro:rime:	
214	6	寅	iən	1	in	1b	t1:rime:	
214	7	饮	ʔiəm	2	in	2	pro:rime:fuse:	
214	8	隐	ʔiən	2	in	2	pro:rime:	
214	9	引	iən	2	in	2	rime:	
215	1	尹	yən	2	in	2	y-spl-d:rime:[t2]:	
215	2	印	ʔiən	3	in	3	pro:rime:	
215	3	墩	tuən	1	tun	1a	t1:rime:	
215	3	顿	tuən	1	tun	1a	t1:rime:	
215	4	顿	tyən	3	tun	3	y-spl-c:rime:	
215	5	盾	duən	2	tun	3	t2:devoi:rime:	
215	6	钝	duən	3	tun	3	devoi:rime:	
215	7	遁	duən	3	tun	3	devoi:rime:	
215	8	吞	t ^h ən	1	t ^h un	1a	t1:[schwa-u]:	
215	9	屯	duən	1	t ^h un	1b	asp:t1:devoi:rime:	
216	1	豚	duən	1	t ^h un	1b	asp:t1:devoi:rime:	
216	2	仑	luən	1	lun	1b	t1:rime:	
216	3	伦	lyən	1	lun	1b	t1:y-spl-c:rime:	
216	3	论	lyən	1	lun	1b	t1:y-spl-c:rime:	
216	4	论	luən	3	lun	3	rime:	
216	5	尊	tsuən	1	tsun	1a	t1:rime:	
216	6	遵	tsyən	1	tsun	1a	t1:y-spl-c:rime:	
216	7	村	ts ^h uən	1	ts ^h un	1a	t1:rime:	
216	8	存	dzuən	1	ts ^h un	1b	asp:t1:devoi:rime:	
216	9	寸	ts ^h uən	3	ts ^h un	3	rime:	
217	1	孙	suən	1	sun	1a	t1:rime:	
217	2	损	suən	2	sun	2	rime:	
217	3	笋	syən	2	sun	2	y-spl-c:rime:	
217	4	准	tʂyən	2	tʂun	2	med-a:retrof-2:rime:	
217	5	春	ts ^h yən	1	tʂ ^h un	1a	t1:med-a:retrof-2:rime:	
217	6	唇	dʒyən	1	tʂ ^h un	1b	affric:asp:t1:devoi:med-a:retrof-2:rime:	
217	7	纯	zyən	1	tʂ ^h un	1b	affric:asp:t1:devoi:med-a:retrof-2:rime:	
217	7	莼	zyən	1	tʂ ^h un	1b	affric:asp:t1:devoi:med-a:retrof-2:rime:	
217	8	蠢	ts ^h yən	2	tʂ ^h un	2	med-a:retrof-2:rime:	
217	9	顺	dʒyən	3	ʂun	3	affric:devoi:med-a:retrof-2:rime:	
218	1	润	riyən	3	run	3	fric:med-a:retrof-2:rime:	
218	2	闰	niyən	3	run	3	fric:med-a:retrof-2:rime:	
218	3	滚	kuən	2	kun	2	rime:	
218	4	棍			kun	3	:	
218	5	昆	kuən	1	k ^h un	1a	t1:rime:[asp]:	
218	5	昆	kuən	1	k ^h un	1a	t1:rime:[asp]:	
218	6	坤	k ^h uən	1	k ^h un	1a	t1:rime:	
218	7	捆	k ^h uən	2	k ^h un	2	t1:rime:	
218	8	困	k ^h uən	3	k ^h un	3	t1:rime:	
218	9	婚	xuən	1	xun	1a	t1:rime:	
218	9	昏	xuən	1	xun	1a	t1:rime:	
219	1	晕	xyən	1	xun	1a	t1:y-spl-d:rime:	
219	2	浑	yuən	1	xun	1b	t1:devoi:rime:	
219	3	魂	yuən	1	xun	1b	t1:devoi:rime:	
219	4	混	yuən	2	xun	3	t2:devoi:rime:	
219	5	温	ʔuən	1	un	1a	t1:pro:rime:	
219	5	瘟	ʔuən	1	un	1a	t1:pro:rime:	
219	6	文	ŋuən	1	un	1b	t1:lab-dnt ^h :degem:rime:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
219	6	纹	muan	1	un	1b	t1:lab-dnt':degem:rime:	
219	7	罔	mjan	1	un	1b	t1:lab-dnt':degem:rime:	
219	8	襖	?uan	2	un	2	pro:rime:	
219	9	问	mjan	3	un	3	lab-dnt':degem:rime:	
220	1	均	kyan	1	tšyn	1a	t1:rime:pal:	
220	1	钧	kyan	1	tšyn	1a	t1:rime:pal:	
220	2	君	kyan	1	tšyn	1a	t1:rime:pal:	
220	2	军	kyan	1	tšyn	1a	t1:rime:pal:	
220	3	俊	tsyan	3	tšyn	3	rime:pal:	
220	4	菌	gyan	2	tšyn	3	t2:devoi:rime:pal:	
220	5	群	gyan	1	tš ^h yn	1b	asp:t1:devoi:rime:pal:	
220	6	裙	gyan	1	tš ^h yn	1b	asp:t1:devoi:rime:pal:	
220	7	助	xyan	1	šyn	1a	t1:rime:pal:	
220	7	薰	xyan	1	šyn	1a	t1:rime:pal:	
220	8	寻	ziam	1	šyn	1b	t1:devoi:rime:fuse:pal:[i-y]:	
220	9	巡	zyan	1	šyn	1b	t1:devoi:rime:pal:	
220	9	循	zyan	1	šyn	1b	t1:devoi:rime:pal:	
220	9	旬	zyan	1	šyn	1b	t1:devoi:rime:pal:	
221	1	讯	sian	3	šyn	3	rime:pal:[i-y]:	
221	2	迅	syen	3	šyn	3	rime:pal:	
221	3	训	xyan	3	šyn	3	rime:pal:	
221	4	匀	yan	1	yn	1b	t1:rime:	
221	5	云	jyan	1	yn	1b	t1:degem:rime:	
221	6	允	yan	2	yn	2	rime:	
221	7	爨	?yan	3	yn	3	pro:rime:	
221	8	酌	jyan	3	yn	3	degem:rime:	
221	9	运	jyan	3	yn	3	degem:rime:	
222	1	晕	jyan	3	yn	3	degem:rime:	
222	2	孕	iaŋ	3	yn	3	rime:[ŋ-n]:[i-y]:	
222	3	帮	paŋ	1	paŋ	1a	t1:a-mg:	
222	4	邦	paŋ	1	paŋ	1a	t1:	
222	5	榜	paŋ	2	paŋ	2	a-mg:	
222	6	绑	paŋ	2	paŋ	2	:	
222	7	棒	baŋ	2	paŋ	3	t2:devoi:	
222	8	旁	baŋ	1	p ^h aŋ	1b	asp:t1:devoi:a-mg:	
222	9	畔	p ^h aŋ	3	p ^h aŋ	3	:	
223	1	忙	maŋ	1	maŋ	1b	t1:a-mg:	
223	2	方	fyaj	1	faŋ	1a	t1:lab-dnt':med-b:	
223	3	芳	f ^h yaŋ	1	faŋ	1a	t1:lab-dnt':med-b:[asp-r]:	
223	4	妨	f ^h yaŋ	1	faŋ	1a	t1:lab-dnt':med-b:[asp-r]:	
223	5	防	vyaŋ	1	faŋ	1b	t1:devoi:lab-dnt':med-b:	
223	6	房	vyaŋ	1	faŋ	1b	t1:devoi:lab-dnt':med-b:	
223	7	仿	fyaj	2	faŋ	2	lab-dnt':med-b:	
223	8	纺	fyaj	2	faŋ	2	lab-dnt':med-b:	
223	9	仿	f ^h yaŋ	2	faŋ	2	lab-dnt':med-b:[asp-r]:	
223	9	访	f ^h yaŋ	2	faŋ	2	lab-dnt':med-b:[asp-r]:	
224	1	访	f ^h yaŋ	3	faŋ	2	lab-dnt':med-b:[t3-t2]:[asp-r]:	
224	2	方	fyaj	3	faŋ	3	lab-dnt':med-b:	
224	3	当	taŋ	1	taŋ	1a	t1:a-mg:	
224	4	党	taŋ	2	taŋ	2	a-mg:	
224	5	挡	taŋ	2	taŋ	2	:	
224	6	当	taŋ	3	taŋ	3	a-mg:	
224	7	汤	t ^h aŋ	1	t ^h aŋ	1a	t1:a-mg:	
224	8	唐	daŋ	1	t ^h aŋ	1b	asp:t1:devoi:a-mg:	
224	8	堂	daŋ	1	t ^h aŋ	1b	asp:t1:devoi:a-mg:	
224	8	塘	daŋ	1	t ^h aŋ	1b	asp:t1:devoi:a-mg:	
224	9	糖	daŋ	1	t ^h aŋ	1b	asp:t1:devoi:a-mg:	
225	1	糖	t ^h aŋ	2	t ^h aŋ	2	a-mg:	
225	2	糖	t ^h aŋ	3	t ^h aŋ	3	a-mg:	
225	3	糖	t ^h aŋ	3	t ^h aŋ	3	a-mg:	
225	4	囊	naŋ	1	naŋ	1b	t1:a-mg:	
225	5	囊	laŋ	1	laŋ	1b	t1:a-mg:	
225	5	象	laŋ	1	laŋ	1b	t1:a-mg:	
225	5	郎	laŋ	1	laŋ	1b	t1:a-mg:	
225	6	浪	laŋ	3	laŋ	3	a-mg:	
225	7	贲	tsaŋ	1	tsaŋ	1a	t1:a-mg:	
225	7	贲	tsaŋ	1	tsaŋ	1a	t1:a-mg:	
225	7	贲	tsaŋ	1	tsaŋ	1a	t1:a-mg:	
225	8	贲	tsaŋ	3	tsaŋ	3	a-mg:	
225	9	贲	dzaŋ	3	tsaŋ	3	devoi:a-mg:	
225	9	贲	dzaŋ	3	tsaŋ	3	devoi:a-mg:	
226	1	仓	ts ^h aŋ	1	ts ^h aŋ	1a	t1:a-mg:	
226	2	苍	ts ^h aŋ	1	ts ^h aŋ	1a	t1:a-mg:	
226	3	藏	dzaŋ	1	ts ^h aŋ	1b	asp:t1:devoi:a-mg:	
226	4	桑	saŋ	1	saŋ	1a	t1:a-mg:	
226	5	丧	saŋ	1	saŋ	1a	t1:a-mg:	
226	6	嗓	saŋ	2	saŋ	2	a-mg:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
226	7	茂	saŋ	3	saŋ	3	a-mg:	
226	8	张	fiəŋ	1	tʂaŋ	1a	t1:fric:med-a:retrof-2:	
226	9	樟	tʂiaŋ	1	tʂaŋ	1a	t1:med-a:retrof-2:	
226	9	章	tʂiaŋ	1	tʂaŋ	1a	t1:med-a:retrof-2:	
227	1	长	fiəŋ	2	tʂaŋ	2	fric:med-a:retrof-2:	
227	2	涨	fiəŋ	2	tʂaŋ	2	fric:med-a:retrof-2:	
227	3	掌	tʂiaŋ	2	tʂaŋ	2	med-a:retrof-2:	
227	4	帐	fiəŋ	3	tʂaŋ	3	fric:med-a:retrof-2:	
227	4	涨	fiəŋ	3	tʂaŋ	3	fric:med-a:retrof-2:	
227	4	胀	fiəŋ	3	tʂaŋ	3	fric:med-a:retrof-2:	
227	4	账	fiəŋ	3	tʂaŋ	3	fric:med-a:retrof-2:	
227	5	丈	ɕiaŋ	2	tʂaŋ	3	t2:devoi:fric:med-a:retrof-2:	
227	6	杖	ɕiaŋ	2	tʂaŋ	3	t2:devoi:fric:med-a:retrof-2:	
227	7	仗	ɕiaŋ	2	tʂaŋ	3	t2:devoi:fric:med-a:retrof-2:	
227	8	障	tʂiaŋ	2	tʂaŋ	3	t2:med-a:retrof-2:	
227	9	昌	tʂ ^h iaŋ	1	tʂ ^h aŋ	1a	t1:med-a:retrof-2:	
228	1	肠	ɕiaŋ	1	tʂ ^h aŋ	1b	asp:t1:devoi:fric:med-a:retrof-2:	
228	1	长	ɕiaŋ	1	tʂ ^h aŋ	1b	asp:t1:devoi:fric:med-a:retrof-2:	
228	2	场	ɕiaŋ	1	tʂ ^h aŋ	1b	asp:t1:devoi:fric:med-a:retrof-2:	
228	3	常	ziaŋ	1	tʂ ^h aŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	
228	4	尝	ziaŋ	1	tʂ ^h aŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	
228	5	尝	ziaŋ	1	tʂ ^h aŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	
228	6	厂	tʂ ^h iaŋ	2	tʂ ^h aŋ	2	med-a:retrof-2:	
228	7	畅	ʃ ^h iaŋ	3	tʂ ^h aŋ	3	fric:med-a:retrof-2:	
228	8	唱	tʂ ^h iaŋ	3	tʂ ^h aŋ	3	med-a:retrof-2:	
228	9	倡	tʂ ^h iaŋ	3	tʂ ^h aŋ	3	med-a:retrof-2:	
229	1	商	ʂiaŋ	1	ʂaŋ	1a	t1:med-a:retrof-2:	
229	2	伤	ʂiaŋ	1	ʂaŋ	1a	t1:med-a:retrof-2:	
229	3	赏	ʂiaŋ	2	ʂaŋ	2	med-a:retrof-2:	
229	4	上	ziaŋ	2	ʂaŋ	3	affric:t2:devoi:med-a:retrof-2:	
229	5	尚	ziaŋ	3	ʂaŋ	3	affric:devoi:med-a:retrof-2:	
229	6	上	ziaŋ	3	ʂaŋ	3	affric:devoi:med-a:retrof-2:	
229	7	瓤	riəŋ	1	raŋ	1b	t1:fric:med-a:retrof-2:	
229	8	嚷	riəŋ	2	raŋ	2	fric:med-a:retrof-2:	
229	8	壤	riəŋ	2	raŋ	2	fric:med-a:retrof-2:	
229	8	让	riəŋ	2	raŋ	2	fric:med-a:retrof-2:	
229	9	让	riəŋ	3	raŋ	3	fric:med-a:retrof-2:	
230	1	刚	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	1	冈	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	1	钢	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	2	钢	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	3	缸	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	4	缸	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	5	扛	kaŋ	1	kaŋ	1a	t1:a-mg:	
230	6	港	kaŋ	2	kaŋ	2	:	
230	7	杠	kaŋ		kaŋ	3	:	
230	8	康	k ^h aŋ	1	k ^h aŋ	1a	t1:a-mg:	
230	8	糠	k ^h aŋ	1	k ^h aŋ	1a	t1:a-mg:	
230	9	糠	k ^h aŋ	2	k ^h aŋ	2	a-mg:	
231	1	抗	k ^h aŋ	3	k ^h aŋ	3	a-mg:	
231	2	抗	k ^h aŋ	3	k ^h aŋ	3	a-mg:	
231	3	杭	ɣaŋ	1	xaŋ	1b	t1:devoi:a-mg:	
231	3	航	ɣaŋ	1	xaŋ	1b	t1:devoi:a-mg:	
231	3	行	ɣaŋ	1	xaŋ	1b	t1:devoi:a-mg:	
231	4	航	?aŋ	1	aŋ	1a	t1:pro:a-mg:	
231	5	昂	ŋaŋ	1	aŋ	1b	t1:pro:a-mg:	
231	6	娘	niaŋ	1	niaŋ	1b	t1:	
231	7	凉	liaŋ	1	liaŋ	1b	t1:	
231	7	梁	liaŋ	1	liaŋ	1b	t1:	
231	7	良	liaŋ	1	liaŋ	1b	t1:	
231	8	粮	liaŋ	1	liaŋ	1b	t1:	
231	8	量	liaŋ	1	liaŋ	1b	t1:	
231	9	两	liaŋ	2	liaŋ	2	:	
232	1	亮	liaŋ	3	liaŋ	3	:	
232	1	量	liaŋ	3	liaŋ	3	:	
232	2	凉	liaŋ	3	liaŋ	3	:	
232	3	辆	liaŋ	3	liaŋ	3	:	
232	4	待	tʂiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	5	浆	tʂiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	6	僵	kiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	6	疆	kiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	7	姜	kiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	7	姜	kiaŋ	1	tʂiaŋ	1a	t1:pal:	
232	8	江	kaŋ	1	tʂiaŋ	1a	t1:sharp:pal:	
232	9	浆	tʂiaŋ	2	tʂiaŋ	2	pal:	
233	1	讲	kaŋ	2	tʂiaŋ	2	sharp:pal:	
233	2	酱	tʂiaŋ	3	tʂiaŋ	3	pal:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
233	3	将	tɕiaŋ	3	tɕiaŋ	3	pal:	
233	4	匠	dziaŋ	3	tɕiaŋ	3	devoi:pal:	
233	5	降	kaŋ	3	tɕiaŋ	3	sharp:pal:	
233	6	枪	tɕ ^h iaŋ	1	tɕ ^h iaŋ	1a	t1:pal:	
233	7	腔	k ^h aŋ	1	tɕ ^h iaŋ	1a	t1:pal:	
233	8	墙	dziaŋ	1	tɕ ^h iaŋ	1b	asp:t1:devoi:pal:	
233	9	强	giaŋ	1	tɕ ^h iaŋ	1b	asp:t1:devoi:pal:	
234	1	枪	tɕ ^h iaŋ	2	tɕ ^h iaŋ	2	pal:	
234	2	厢	siaŋ	1	ɕiaŋ	1a	t1:pal:	
234	2	相	siaŋ	1	ɕiaŋ	1a	t1:pal:	
234	2	箱	siaŋ	1	ɕiaŋ	1a	t1:pal:	
234	2	箱	siaŋ	1	ɕiaŋ	1a	t1:pal:	
234	3	香	xiaŋ	1	ɕiaŋ	1a	t1:pal:	
234	4	乡	xiaŋ	1	ɕiaŋ	1a	t1:pal:	
234	5	翔	ziaŋ	1	ɕiaŋ	1b	t1:devoi:pal:	
234	5	祥	ziaŋ	1	ɕiaŋ	1b	t1:devoi:pal:	
234	6	祥	ziaŋ	1	ɕiaŋ	1b	t1:devoi:pal:	
234	7	痒	ɣaŋ	1	ɕiaŋ	1b	t1:devoi:sharp:pal:	
234	8	想	siaŋ	2	ɕiaŋ	2	pal:	
234	9	享	xiaŋ	2	ɕiaŋ	2	pal:	
234	9	响	xiaŋ	2	ɕiaŋ	2	pal:	
235	1	相	siaŋ	3	ɕiaŋ	3	pal:	
235	2	像	ziaŋ	2	ɕiaŋ	3	t2:devoi:pal:	
235	3	像	ziaŋ	2	ɕiaŋ	3	t2:devoi:pal:	
235	3	象	ziaŋ	2	ɕiaŋ	3	t2:devoi:pal:	
235	4	向	xiaŋ	3	ɕiaŋ	3	pal:	
235	5	项	ɣaŋ	2	ɕiaŋ	3	t2:devoi:sharp:pal:	
235	6	巷	ɣaŋ	3	ɕiaŋ	3	devoi:sharp:pal:	
235	7	央	?iaŋ	1	yaŋ	1a	t1:pro:[i-y]:	
235	7	映	?iaŋ	1	iaŋ	1a	t1:pro:	
235	7	映	?iaŋ	1	iaŋ	1a	t1:pro:	
235	8	羊	iaŋ	1	iaŋ	1b	t1:	
235	9	杨	iaŋ	1	iaŋ	1b	t1:	
235	9	洋	iaŋ	1	iaŋ	1b	t1:	
236	1	杨	iaŋ	1	iaŋ	1b	t1:	
236	1	阳	iaŋ	1	iaŋ	1b	t1:	
236	2	仰	ŋiaŋ	2	iaŋ	2	pro:	
236	3	痒	iaŋ	2	iaŋ	2	pro:	
236	4	痒	iaŋ	2	iaŋ	2	pro:	
236	5	恙	iaŋ	3	iaŋ	3	pro:	
236	6	样	iaŋ	3	iaŋ	3	pro:	
236	7	庄	tɕiaŋ	1	tɕuaŋ	1a	t1:i-del:u-ins-a:	
236	8	装	tɕiaŋ	1	tɕuaŋ	1a	t1:i-del:u-ins-a:	
236	9	粧	ʔaŋ	1	tɕuaŋ	1a	t1:fric:retrof-l:u-ins-a:	
237	1	壮	tɕiaŋ	3	tɕuaŋ	3	i-del:u-ins-a:	
237	2	状	dɕiaŋ	3	tɕuaŋ	3	devoi:i-del:u-ins-a:	
237	3	撞	ʔaŋ	3	tɕuaŋ	3	devoi:fric:retrof-l:u-ins-a:	
237	4	疮	tɕ ^h iaŋ	1	tɕ ^h uaŋ	1a	t1:i-del:u-ins-a:	
237	5		tɕ ^h aŋ	1	tɕ ^h uaŋ	1a	t1:u-ins-a:	
237	6	床	dɕiaŋ	1	tɕ ^h uaŋ	1b	asp:t1:devoi:i-del:u-ins-a:	
237	7	闯	tɕ ^h iaŋ	2	tɕ ^h uaŋ	2	i-del:u-ins-a:	
237	8	创	tɕ ^h iaŋ	3	tɕ ^h uaŋ	3	i-del:u-ins-a:	
237	9	霜	ɕiaŋ	1	ɕuaŋ	1a	t1:i-del:u-ins-a:	
238	1	双	ɕaŋ	1	ɕuaŋ	1a	t1:u-ins-a:	
238	2	爽	ɕiaŋ	2	ɕuaŋ	2	i-del:u-ins-a:	
238	3	光	kuɑŋ	1	kuɑŋ	1a	t1:A-mg:	
238	4	广	kuɑŋ	2	kuɑŋ	2	a-mg:	
238	5	逛	kyɑŋ	3	kuɑŋ	3	y-spl-a:	
238	6	筐	k ^h yaŋ	1	k ^h uaŋ	1a	t1:y-spl-a:	
238	7	狂	gyɑŋ	1	k ^h uaŋ	1b	asp:t1:devoi:y-spl-a:	
238	8	矿	k ^h uaŋ	3	k ^h uaŋ	3	a-mg:	
238	9	况	xyaŋ	3	k ^h uaŋ	3	[x-k ^h]:y-spl-a:	
239	1	慌	xuaŋ	1	xuaŋ	1a	t1:a-mg:	
239	1	荒	xuaŋ	1	xuaŋ	1a	t1:a-mg:	
239	2	黄	ɣuaŋ	1	xuaŋ	1b	t1:devoi:a-mg:	
239	3	皇	ɣuaŋ	1	xuaŋ	1b	t1:devoi:a-mg:	
239	3	煌	ɣuaŋ	1	xuaŋ	1b	t1:devoi:a-mg:	
239	4	谎	xuaŋ	2	xuaŋ	2	a-mg:	
239	5	晃	ɣuaŋ	2	xuaŋ	2	devoi:a-mg:	
239	6	晃	xuaŋ	2	xuaŋ	3	:	
239	7	汪	?uaŋ	1	uaŋ	1a	t1:pro:a-mg:	
239	8	亡	ŋyaŋ	1	uaŋ	1b	t1:lab-dnt':degem:	
239	9	王	jyaŋ	1	uaŋ	1b	t1:degem:y-spl-a:	
240	1	网	ŋyaŋ	2	uaŋ	2	lab-dnt':degem:	
240	2	往	jyaŋ	2	uaŋ	2	degem:y-spl-a:	
240	3	忘	ŋyaŋ	3	uaŋ	3	lab-dnt':degem:	
240	4	望	ŋyaŋ	3	uaŋ	3	lab-dnt':degem:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
240	5	旺	jyaŋ	3	uaŋ	3	degem:y-spl-a:	
240	6	崩	pəŋ	1	pəŋ	1a	t1:	cha:
240	7	烹	p ^h aŋ	1	p ^h əŋ	1a	t1:vocal:fuse:coal:	cha:
240	8	棚	bəŋ	1	p ^h əŋ	1b	asp:t1:devoi:	cha:
240	9	棚	bəŋ	1	p ^h əŋ	1b	asp:t1:devoi:	cha:
241	1	彭	baŋ	1	p ^h əŋ	1b	asp:t1:vocal:fuse:coal:	cha:
241	2	棚	baŋ	1	p ^h əŋ	1b	asp:t1:vocal:fuse:coal:	cha:
241	3	蓬	buŋ	1	p ^h əŋ	1b	asp:t1:devoi:med-b:syllab:	cha:
241	3	蓬	buŋ	1	p ^h əŋ	1b	asp:t1:devoi:med-b:syllab:	cha:
241	4	捧	f ^h iuŋ	2	p ^h əŋ	2	med-b:syllab:[asp-r]:[f-p]:	cha:
241	5	碰			p ^h əŋ	3	:	
241	6	盟	miaŋ	1	məŋ	1b	t1:lab-dnt':vocal:fuse:coal:	cha:
241	7	蒙	muŋ	1	məŋ	1b	t1:med-b:syllab:	cha:
241	8	猛	maŋ	2	məŋ	2	vocal:fuse:coal:	cha:
241	9	孟	maŋ	3	məŋ	3	vocal:fuse:coal:	cha:
242	1	梦	miuŋ	3	məŋ	3	lab-dnt':med-b:syllab:	cha:
242	2	疯	fiuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:	cha:
242	2	风	fiuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:	cha:
242	3	丰	f ^h iuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:[asp-r]:	cha:
242	4	封	fiuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:	cha:
242	5	峰	f ^h iuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:[asp-r]:	cha:
242	5	峰	f ^h iuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:[asp-r]:	cha:
242	6	蜂	f ^h iuŋ	1	fəŋ	1a	t1:lab-dnt':med-b:syllab:[asp-r]:	cha:
242	7	冯	viuŋ	1	fəŋ	1b	t1:devoi:lab-dnt':med-b:syllab:	cha:
242	8	逢	viuŋ	1	fəŋ	1b	t1:devoi:lab-dnt':med-b:syllab:	cha:
242	9	缝	viuŋ	1	fəŋ	1b	t1:devoi:lab-dnt':med-b:syllab:	cha:
243	1	讽	fiuŋ	3	fəŋ	2	lab-dnt':med-b:syllab:[t3-t2]:	cha:
243	2	风	viuŋ	3	fəŋ	3	devoi:lab-dnt':med-b:syllab:	cha:
243	3	奉	viuŋ	2	fəŋ	3	t2:devoi:lab-dnt':med-b:syllab:	cha:
243	4	俸	viuŋ	3	fəŋ	3	devoi:lab-dnt':med-b:syllab:	cha:
243	5	缝	viuŋ	3	fəŋ	3	devoi:lab-dnt':med-b:syllab:	cha:
243	6	灯	təŋ	1	təŋ	1a	t1:	
243	6	登	təŋ	1	təŋ	1a	t1:	
243	7	等	təŋ	2	təŋ	2	:	
243	8	凳	təŋ	3	təŋ	3	:	
243	9	邓	dəŋ	3	təŋ	3	devoi:	
244	1	瞪	djaŋ	3	təŋ	3	Devoi:med-a:[t't]:	
244	2	腾	dəŋ	1	t ^h əŋ	1b	asp:t1:devoi:	cha:
244	3	腾	dəŋ	1	t ^h əŋ	1b	asp:t1:devoi:	cha:
244	4	滕	dəŋ	1	t ^h əŋ	1b	asp:t1:devoi:	cha:
244	5	疼	dəŋ	1	t ^h əŋ	1b	asp:t1:devoi:	cha:
244	6	能	nəŋ	1	nəŋ	1b	t1:	
244	7	棱	ləŋ	1	ləŋ	1b	t1:	
244	8	冷	laŋ	2	ləŋ	2	vocal:fuse:coal:[!t2]:	cha:
244	9	增	tsəŋ	1	tsəŋ	1a	t1:	
245	1	曾	tsəŋ	1	tsəŋ	1a	t1:	
245	2	赠	dzəŋ	3	tsəŋ	3	devoi:	
245	3	曾	dzəŋ	1	ts ^h əŋ	1b	asp:t1:devoi:	
245	4	层	dzəŋ	1	ts ^h əŋ	1b	asp:t1:devoi:	
245	5	僧	səŋ	1	səŋ	1a	t1:	
245	6	蒸	tšiaŋ	1	tʂəŋ	1a	t1:med-a:retrof-2:	cha:
245	7	争	tʂaŋ	1	tʂəŋ	1a	t1:vocal:fuse:coal:	cha:
245	8	争	tʂaŋ	1	tʂəŋ	1a	t1:vocal:fuse:coal:	cha:
245	9	净	tʂaŋ	1	tʂəŋ	1a	t1:vocal:fuse:coal:	cha:
246	1	正	tšiaŋ	1	tʂəŋ	1a	t1:vocal:fuse:coal:med-a:retrof-2:	cha:
246	2	征	tšiaŋ	1	tʂəŋ	1a	t1:vocal:fuse:coal:med-a:retrof-2:	cha:
246	3	整	tšiaŋ	2	tʂəŋ	2	vocal:fuse:coal:med-a:retrof-2:	cha:
246	4	症	tšiaŋ	3	tʂəŋ	3	med-a:retrof-2:	cha:
246	4	证	tšiaŋ	3	tʂəŋ	3	med-a:retrof-2:	cha:
246	5	郑	diaŋ	3	tʂəŋ	3	devoi:fric:vocal:fuse:coal:med-a:retrof-2:	cha:
246	6	正	tšiaŋ	3	tʂəŋ	3	vocal:fuse:coal:med-a:retrof-2:	cha:
246	7	政	tšiaŋ	3	tʂəŋ	3	vocal:fuse:coal:med-a:retrof-2:	cha:
246	8	称	tʂ ^h iaŋ	1	tʂ ^h əŋ	1a	t1:med-a:retrof-2:	cha:
246	9	鼎	tʂ ^h iaŋ	1	tʂ ^h əŋ	1a	t1:fric:vocal:fuse:coal:med-a:retrof-2:	cha:
247	1	乘	dziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
247	2	丞	ziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
247	2	承	ziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:med-a:retrof-2:	cha:
247	3	橙	daŋ	1	tʂ ^h əŋ	1b	asp:t1:devoi:fric:vocal:fuse:coal:retrof-2:	cha:
247	4	呈	diaŋ	1	tʂ ^h əŋ	1b	asp:t1:devoi:fric:vocal:fuse:coal:med-a:retrof-2:	cha:
247	5	程	diaŋ	1	tʂ ^h əŋ	1b	asp:t1:devoi:fric:vocal:fuse:coal:med-a:retrof-2:	cha:
247	6	成	ziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:vocal:fuse:coal:med-a:retrof-2:	cha:
247	7	城	ziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:vocal:fuse:coal:med-a:retrof-2:	cha:
247	8	减	ziaŋ	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:vocal:fuse:coal:med-a:retrof-2:	cha:

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
247	9	盛	ziań	1	tʂ ^h əŋ	1b	affric:asp:t1:devoi:vocal:fuse:coal:med-a:retrof-2:	cha:
248	1	秤	tʂ ^h iaŋ	3	tʂ ^h əŋ	3	med-a:retrof-2:	cha:
248	2	升	ʂiaŋ	1	ʂəŋ	1a	t1:med-a:retrof-2:	cha:
248	3	生	ʂań	1	ʂəŋ	1a	t1:vocal:fuse:coal:	cha:
248	4	甥	ʂań	1	ʂəŋ	1a	t1:vocal:fuse:coal:	cha:
248	5	牲	ʂań	1	ʂəŋ	1a	t1:vocal:fuse:coal:	cha:
248	6	声	ʂiań	1	ʂəŋ	1a	t1:vocal:fuse:coal:med-a:retrof-2:	cha:
248	7	绳	dziaŋ	1	ʂəŋ	1b	t1:devoi:med-a:retrof-2:[affric]:	cha:
248	8	省	ʂiań	2	ʂəŋ	2	vocal:fuse:coal:i-del:	cha:
248	9	剩	dziaŋ	3	ʂəŋ	3	affric:devoi:med-a:retrof-2:	cha:
249	1	胜	ʂiaŋ	3	ʂəŋ	3	med-a:retrof-2:	cha:
249	2	圣	ʂiań	3	ʂəŋ	3	vocal:fuse:coal:med-a:retrof-2:	cha:
249	3	盛	ziań	3	ʂəŋ	3	affric:devoi:vocal:fuse:coal:med-a:retrof-2:	cha:
249	4	奶	ńiaŋ	1	rəŋ	1a	fric:med-a:retrof-2:[t1]:	cha:
249	5	仍	ńiaŋ	1	rəŋ	1b	t1:fric:med-a:retrof-2:	cha:
249	6	更	kań	1	kəŋ	1a	t1:vocal:fuse:coal:	cha:
249	7	更	kań	1	kəŋ	1a	t1:vocal:fuse:coal:	cha:
249	8	羹	kań	1	kəŋ	1a	t1:vocal:fuse:coal:	cha:
249	9	庚	kań	1	kəŋ	1a	t1:vocal:fuse:coal:	cha:
250	1	耕	kań	1	kəŋ	1a	t1:vocal:fuse:coal:	cha:
250	2	更	kań	3	kəŋ	3	vocal:fuse:coal:	cha:
250	3	坑	k ^h ań	1	k ^h əŋ	1a	t1:vocal:fuse:coal:	cha:
250	4	恒	ɣəŋ	1	xəŋ	1b	t1:devoi:	cha:
250	5	衡	ɣań	1	xəŋ	1b	t1:devoi:vocal:fuse:coal:	cha:
250	6	横	ɣuań	1	xəŋ	1b	t1:devoi:vocal:fuse:coal:u-del:	cha:
250	7	横	ɣuań	3	xəŋ	3	devoi:vocal:fuse:coal:u-del:	cha:
250	8	冰	piaŋ	1	piŋ	1a	t1:rime:	
250	9	兵	piań	1	piŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
251	1	秉	piań	2	piŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:	
251	2	丙	piań	2	piŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:	
251	3	饼	piań	2	piŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:	
251	4	病	biań	3	piŋ	3	devoi:vocal:fuse:raise-il:raise-i2:degem:degem:	
251	5	井	piań	3	piŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:	
251	6	井	biań	2	piŋ	3	t2:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:	
251	7	凭	biaŋ	1	p ^h iŋ	1b	asp:t1:devoi:rime:	
251	8	平	biaŋ	1	p ^h iŋ	1b	asp:t1:devoi:rime:	
251	9	评	biaŋ	1	p ^h iŋ	1b	asp:t1:devoi:rime:	
252	1	坪	biań	1	p ^h iŋ	1b	asp:t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	2	瓶	biań	1	p ^h iŋ	1b	asp:t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	3	屏	biań	1	p ^h iŋ	1b	asp:t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	4	萍	biań	1	p ^h iŋ	1b	asp:t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	5	明	miań	1	miŋ	1b	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	6	鸣	miań	1	miŋ	1b	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	7	名	miań	1	miŋ	1b	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	8	铭	miań	1	miŋ	1b	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
252	9	命	miań	3	miŋ	3	vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	1	丁	tiań	1	tiŋ	1a	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	2	钉	tiań	1	tiŋ	1a	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	3	顶	tiań	2	tiŋ	2	vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	4	鼎	tiań	2	tiŋ	2	vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	5	钉	tiań	3	tiŋ	3	vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	6	订	tiań	3	tiŋ	3	vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	7	定	diań	3	tiŋ	3	devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	8	听	t ^h iań	1	t ^h iŋ	1a	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
253	9	厅	t ^h iań	1	t ^h iŋ	1a	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
254	1	亭	diań	1	t ^h iŋ	1b	asp:t1:devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
254	2	停	diań	1	t ^h iŋ	1b	asp:t1:devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
254	3	廷	diań	1	t ^h ij	1b	asp:t1:devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
254	4	庭	diań	1	t ^h ij	1b	asp:t1:devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
254	5	艇	diań	2	t ^h ij	2	devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:[asp]:	
254	6	挺	diań	2	t ^h ij	2	devoi:vocal:fuse:coal:raise-il:raise-i2:degem:degem:[asp]:	
254	7	凝	ŋiəŋ	1	niŋ	1b	t1:rime:[ŋ-n]:	
254	8	宁	niań	1	niŋ	1b	t1:vocal:fuse:coal:raise-il:raise-i2:degem:degem:	
254	9	凌	liəŋ	1	liŋ	1b	t1:rime:	
254	9	菱	liəŋ	1	liŋ	1b	t1:rime:	
254	9	陵	liəŋ	1	liŋ	1b	t1:rime:	
255	1	灵	liań	1	liŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
255	2	铃	liań	1	liŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
255	2	零	liań	1	liŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
255	3	伶	liań	1	liŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
255	4	领	liań	2	liŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:	
255	5	岭	liań	2	liŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:	
255	6	令	liań	3	liŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:	
255	7	另	liań	3	liŋ	3	:	
255	8	京	kiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
255	9	荆	kiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	1	惊	kiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	2	精	tsiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	3	晶	tsiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	4	睛	tsiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	5	经	kiań	1	t ^s iŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	6	景	kiań	2	t ^s iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	6	警	kiań	2	t ^s iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	7	井	tsiań	2	t ^s iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	8	颈	kiań	2	t ^s iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
256	9	境	kiań	2	t ^s iŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	1	竟	kiań	3	t ^s iŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	2	敬	kiań	3	t ^s iŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	3	镜	kiań	3	t ^s iŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	4	竞	giań	3	t ^s iŋ	3	devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	5	静	dziań	2	t ^s iŋ	3	t2:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	6	净	dziań	3	t ^s iŋ	3	devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	7	轻	k ^h iań	1	t ^s ^h iŋ	1a	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	8	清	ts ^h iań	1	t ^s ^h iŋ	1a	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
257	9	青	ts ^h iań	1	t ^s ^h iŋ	1a	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	1	倾	k ^h iań	1	t ^s ^h iŋ	1a	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	2	情	dziań	1	t ^s ^h iŋ	1b	asp:t1:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	3	晴	dziań	1	t ^s ^h iŋ	1b	asp:t1:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	4	请	ts ^h iań	2	t ^s ^h iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	5	顷	k ^h yań	2	t ^s ^h iŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:u-del:pal:	
258	6	庆	k ^h iań	3	t ^s ^h iŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	7	兴	xiəŋ	1	siŋ	1a	t1:rime:pal:	
258	8	星	siań	1	siŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
258	9	醒	siań	1	siŋ	1a	t1:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	1	行	yań	1	siŋ	1b	t1:devoi:[sharp]:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	2	刑	ɣiań	1	siŋ	1b	t1:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	2	型	ɣiań	1	siŋ	1b	t1:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	2	形	ɣiań	1	siŋ	1b	t1:devoi:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	3	省	siań	2	siŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	4	醒	siań	2	siŋ	2	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	5	兴	xiəŋ	3	siŋ	3	rime:pal:	
259	6	杏	ɣań	2	siŋ	3	t2:devoi:[sharp]:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
259	7	率	ɣań	2	śiŋ	3	t2:devoi:[sharp]:vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	8	性	siań	3	śiŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
259	9	姓	siań	3	śiŋ	3	vocal:fuse:raise-il:raise-i2:degem:degem:pal:	
260	1	应	ʔiəŋ	1	iŋ	1a	t1:pro:rime:	
260	1	鹿	ʔiəŋ	1	iŋ	1a	t1:pro:rime:	
260	2	樱	ʔań	1	iŋ	1a	t1:[sharp]:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	2	鸚	ʔań	1	iŋ	1a	t1:[sharp]:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	3	英	ʔiań	1	iŋ	1a	t1:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	4	婴	ʔiań	1	iŋ	1a	t1:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	5	纓	ʔiań	1	iŋ	1a	t1:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	6	蝇	iəŋ	1	iŋ	1b	t1:rime:	
260	7	迎	ŋiań	1	iŋ	1b	t1:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	8	盈	iań	1	iŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
260	9	赢	iań	1	iŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:	
261	1	营	yań	1	iŋ	1b	t1:vocal:fuse:raise-il:raise-i2:degem:degem:[u-del]:	
261	2	萤	ɣyań	1	iŋ	1b	t1:u-del:vocal:fuse:raise-il:raise-i2:degem:degem:[pro-ɣ]:	
261	3	影	ʔiań	2	iŋ	2	pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
261	4	应	ʔiəŋ	3	iŋ	3	pro:rime:	
261	5	硬	ŋań	3	iŋ	3	[sharp]:pro:vocal:fuse:raise-il:raise-i2:degem:degem:	
261	6	东	tuŋ	1	tuŋ	1a	t1:	
261	7	冬	tuŋ	1	tuŋ	1a	t1:	
261	8	懂	tuŋ	2	tuŋ	2	:	
261	8	董	tuŋ	2	tuŋ	2	:	
261	9	冻	tuŋ	3	tuŋ	3	:	
262	1	动	duŋ	2	tuŋ	3	t2:devoi:	
262	2	洞	duŋ	3	tuŋ	3	devoi:	
262	3	通	t ^h uŋ	1	t ^h uŋ	1a	t1:	
262	4	同	duŋ	1	t ^h uŋ	1b	asp:t1:devoi:	
262	5	筒	duŋ	1	t ^h uŋ	2	asp:devoi:[t1-t2]:	
262	5	铜	duŋ	1	t ^h uŋ	1b	asp:t1:devoi:	
262	6	桐	duŋ	1	t ^h uŋ	1b	asp:t1:devoi:	
262	7	瞳	duŋ	1	t ^h uŋ	1b	asp:t1:devoi:	
262	7	童	duŋ	1	t ^h uŋ	1b	asp:t1:devoi:	
262	8	桶	t ^h uŋ	2	t ^h uŋ	2	:	
262	9	统	t ^h uŋ	3	t ^h uŋ	2	[t3-t2]:	
263	1	痛	t ^h uŋ	3	t ^h uŋ	3	:	
263	2	农	nuŋ	1	nuŋ	1b	t1:	
263	3	脓	nuŋ	1	nuŋ	1b	t1:	
263	4	浓	nuŋ	1	nuŋ	1b	t1:	
263	5	弄	luŋ	3	nuŋ	3	[l-n]:	
263	6	笼	luŋ	1	luŋ	1b	t1:	
263	7	聋	luŋ	1	luŋ	1b	t1:	
263	8	隆	liuŋ	1	luŋ	1b	t1:y-spl-b:	
263	9	龙	liuŋ	1	luŋ	1b	t1:y-spl-b:	
264	1	拢	luŋ	2	luŋ	2	:	
264	2	陇	liuŋ	2	luŋ	2	y-spl-b:	
264	2	珑	liuŋ	2	luŋ	2	y-spl-b:	
264	3	棕	tsuŋ	1	tsuŋ	1a	t1:	
264	4	攀	tsuŋ	1	tsuŋ	1a	t1:	
264	5	宗	tsuŋ	1	tsuŋ	1a	t1:	
264	6	踪	tsiuŋ	1	tsuŋ	1a	t1:y-spl-b:	
264	7	总	tsuŋ	2	tsuŋ	2	:	
264	8	棕	tsuŋ	3	tsuŋ	3	:	
264	9	纵	tsiuŋ	3	tsuŋ	3	y-spl-b:	
265	1	囱	ts ^h uŋ	1	ts ^h uŋ	1a	t1:	
265	1	匆	ts ^h uŋ	1	ts ^h uŋ	1a	t1:	
265	1	聪	ts ^h uŋ	1	ts ^h uŋ	1a	t1:	
265	1	葱	ts ^h uŋ	1	ts ^h uŋ	1a	t1:	
265	2	丛	dzuŋ	1	ts ^h uŋ	1b	asp:t1:devoi:	
265	3	从	dziuŋ	1	ts ^h uŋ	1b	asp:t1:devoi:y-spl-b:	
265	4	松	suŋ	1	suŋ	1a	t1:	
265	5	松	suŋ	1	suŋ	1a	t1:	
265	6	送	suŋ	3	suŋ	3	:	
265	7	宋	suŋ	3	suŋ	3	:	
265	8	讼	ziuŋ	3	suŋ	3	devoi:y-spl-b:	
265	8	诵	ziuŋ	3	suŋ	3	devoi:y-spl-b:	
265	8	颂	ziuŋ	3	suŋ	3	devoi:y-spl-b:	
265	9	中	ʔiuŋ	1	tɕuŋ	1a	t1:fric:med-a:retrof-2:	

pg	co	ch	smc	tn	mb	tn	Diachronic	Allophonic
265	9	忠	ʈiuŋ	1	tʂuŋ	1a	t1:fric:med-a:retrof-2:	
266	1	终	tʂiuŋ	1	tʂuŋ	1a	t1:med-a:retrof-2:	
266	2	钟	tʂiuŋ	1	tʂuŋ	1a	t1:med-a:retrof-2:	
266	2	钟	tʂiuŋ	1	tʂuŋ	1a	t1:med-a:retrof-2:	
266	3	种	tʂiuŋ	2	tʂuŋ	2	med-a:retrof-2:	
266	4	中	ʈiuŋ	3	tʂuŋ	3	fric:med-a:retrof-2:	
266	5	仲	ʈiuŋ	3	tʂuŋ	3	devoi:fric:med-a:retrof-2:	
266	6	众	tʂiuŋ	3	tʂuŋ	3	med-a:retrof-2:	
266	7	重	ʈiuŋ	2	tʂuŋ	3	t2:devoi:med-a:retrof-2:	
266	8	种	tʂiuŋ	3	tʂuŋ	3	med-a:retrof-2:	
266	9	充	tʂ ^h iuŋ	1	tʂ ^h uŋ	1a	t1:med-a:retrof-2:	
267	1	冲	tʂ ^h iuŋ	1	tʂ ^h uŋ	1a	t1:med-a:retrof-2:	
267	2	春	ʂiuŋ	1	tʂ ^h uŋ	1a	t1:med-a:retrof-2:[asp]:[fric]:	
267	3	虫	ʈiuŋ	1	tʂ ^h uŋ	1b	asp:t1:devoi:fric:med-a:retrof-2:	
267	4	崇	dʒiuŋ	1	tʂ ^h uŋ	1b	asp:t1:devoi:i-del:	
267	5	重	ʈiuŋ	1	tʂ ^h uŋ	1b	asp:t1:devoi:fric:med-a:retrof-2:	
267	6	充	ʈ ^h iuŋ	2	tʂ ^h uŋ	2	fric:med-a:retrof-2:	
267	7	荣	jyaŋ	1	ruŋ	1b	t1:degem:fuse:[ya-ru]:	
267	8	戎	riiŋ	1	ruŋ	1b	t1:fric:med-a:retrof-2:	
267	9	绒	riiŋ	1	ruŋ	1b	t1:fric:med-a:retrof-2:	
268	1	茸	riiŋ	1	ruŋ	1b	t1:fric:med-a:retrof-2:	
268	2	融	iuŋ	1	ruŋ	1b	t1:[i-r]:	
268	3	容	iuŋ	1	ruŋ	1b	t1:[i-r]:	
268	3	蓉	iuŋ	1	ruŋ	1b	t1:[i-r]:	
268	3	熔	iuŋ	1	ruŋ	1b	t1:[i-r]:	
268	4	公	kuŋ	1	kuŋ	1a	t1:	
268	5	功	kuŋ	1	kuŋ	1a	t1:	
268	5	工	kuŋ	1	kuŋ	1a	t1:	
268	5	攻	kuŋ	1	kuŋ	1a	t1:	
268	6	弓	kiuŋ	1	kuŋ	1a	t1:y-spl-b:	
268	7	宫	kiuŋ	1	kuŋ	1a	t1:y-spl-b:	
268	7	躬	kiuŋ	1	kuŋ	1a	t1:y-spl-b:	
268	8	恭	kiuŋ	1	kuŋ	1a	t1:y-spl-b:	
268	9	供	kiuŋ	1	kuŋ	1a	t1:y-spl-b:	
269	1	拱	kiuŋ	2	kuŋ	2	y-spl-b:	
269	1	巩	kiuŋ	2	kuŋ	2	y-spl-b:	
269	2	贡	kuŋ	3	kuŋ	3	y-spl-b:	
269	3	供	kiuŋ	3	kuŋ	3	y-spl-b:	
269	4	共	giuŋ	3	kuŋ	3	devoi:y-spl-b:	
269	5	空	k ^h iuŋ	1	k ^h uŋ	1a	t1:y-spl-b:	
269	6	孔	k ^h uŋ	2	k ^h uŋ	2	:	
269	7	恐	k ^h iuŋ	2	k ^h uŋ	2	y-spl-b:	
269	8	控	k ^h uŋ	3	k ^h uŋ	3	:	
269	9	空	k ^h uŋ	3	k ^h uŋ	3	:	
270	1	轰	xyaŋ	1	xuŋ	1a	t1:pal-d:vocal:fuse:coal:[cha]:rime:	
270	2	烘	xuŋ	1	xuŋ	1a	t1:	
270	3	弘	ɣuaŋ	1	xuŋ	1b	t1:devoi:rime:	
270	4	宏	ɣuaŋ	1	xuŋ	1b	t1:devoi:	
270	5	红	ɣuŋ	1	xuŋ	1b	t1:devoi:vocal:fuse:coal:[cha]:rime:	
270	6	洪	ɣuŋ	1	xuŋ	1b	t1:devoi:vocal:fuse:coal:[cha]:rime:	
270	7	鸿	ɣuŋ	1	xuŋ	1b	t1:devoi:vocal:fuse:coal:[cha]:rime:	
270	8	虹	ɣuŋ	1	xuŋ	1b	t1:devoi:vocal:fuse:coal:[cha]:rime:	
270	9	哄	xuŋ		xuŋ	2	:	
271	1	翁	?uŋ	1	uəŋ	1a	t1:pro:[schwa-ins]:	cha:
271	2	瓮	?uŋ	3	uəŋ	3	pro:[schwa-ins]:	cha:
271	3	瓮	gyəŋ	2	tʂyŋ	2	devoi:rime:pal:	
271	4	琼	gyaŋ	1	tʂ ^h yŋ	1b	asp:t1:devoi:vocal:fuse:coal:[cha]:rime:pal:	
271	5	穷	giuŋ	1	tʂ ^h yŋ	1b	asp:t1:devoi:lab-abs:pal:	
271	6	兄	xyaŋ	1	ʂyŋ	1a	t1:vocal:fuse:coal:[cha]:rime:pal:	
271	7	凶	xiuŋ	1	ʂyŋ	1a	t1:lab-abs:pal:	
271	7	凶	xiuŋ	1	ʂyŋ	1a	t1:lab-abs:pal:	
271	8	胸	xiuŋ	1	ʂyŋ	1a	t1:lab-abs:pal:	
271	9	熊	jiuŋ	1	ʂyŋ	1b	t1:lab-abs:[j-ʂ]:	
272	1	雄	jiuŋ	1	ʂyŋ	1b	t1:lab-abs:[j-ʂ]:	
272	2	拥	?iuŋ	1	yŋ	1a	t1:pro:lab-abs:	
272	3	永	jyaŋ	2	yŋ	2	degem:vocal:fuse:coal:[cha]:rime:	
272	4	勇	iuŋ	2	yŋ	2	lab-abs:	
272	5	用	iuŋ	3	yŋ	3	lab-abs:	

Appendix D

From Simplified Middle Chinese to Modern Cantonese

For an discussion of the rule names and the exceptions, see the first section of Appendix C.

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
1	1	疤	pa	1	pa:	1a	t-split:length:	
1	1	巴	pa	1	pa:	1a	t-split:length:	
1	2	八	pat	4	pat	4c	t-split:ac-split:length:	
1	3	拔	bat	4	pat	4b	t-split:devoi:	
1	4	把	pa	2	pa:	2a	t-split:length:	
1	5	坝	pa	3	pa:	3a	t-split:length:	
1	5	霸	pa	3	pa:	3a	t-split:length:	
1	6	色	pa	3	pa:	1a	t-split:length:[t3-t1]:	
1	7	罢	bai	2	pa:	3b	t-split:t2-merge:devoi:length:[i-del]:	
1	8	爬	ba	1	p ^h a:	1b	t-split:asp:devoi:length:	
1	9	怕	p ^h a	3	p ^h a:	3a	t-split:length:	
2	1	妈	ma	1	ma:	1a	[t-split]:length:	
2	2	麻	ma	1	ma:	1b	t-split:length:	
2	3	马	ma	2	ma:	2b	t-split:length:	
2	4	骂	ma	3	ma:	3b	t-split:length:	
2	5	发	fyat	4	fat	4c	t-split:ac-split:lab-dnt':lab-dism:length:	
2	6	乏	vyap	4	fat	4b	t-split:devoi:lab-dnt':lab-dism:final-dism:	
2	7	罚	vyat	4	fat	4b	t-split:devoi:lab-dnt':lab-dism:	
2	8	法	fyap	4	fat	4c	t-split:ac-split:lab-dnt':lab-dism:length:final-dism:	
2	9	发	fyat	4	fat	4c	t-split:ac-split:lab-dnt':lab-dism:length:	
3	1	搭	tap	4	ta:p	4c	t-split:ac-split:length:	
3	2	答	tap	4	ta:p	4c	t-split:ac-split:length:	
3	3	达	dat	4	ta:t	4b	t-split:devoi:length:	
3	4	打	tañ	2	ta:	2a	t-split:length:[n'-del]:	
3	5	大	da	3	tai	3b	t-split:devoi:	
3	6	他	t ^h a	1	t ^h a:	1a	t-split:length:	
3	7	塌	t ^h ap	4	t ^h a:p	4c	t-split:ac-split:length:	
3	8	塔	t ^h ap	4	t ^h a:p	4c	t-split:ac-split:length:	
3	9	踏	t ^h ap	4	ta:p	4b	[t-split]:length:[de-asp]:	
4	1	拿	na	1	na:	1b	t-split:length:	
4	2	纳	nep	4	na:p	4b	t-split:length:	
4	3	拉	lap	4	lai	1a	[t4-t1]:[t-split]:length:[p-i]:	
4	4	腊	lap	4	la:p	4b	t-split:length:	
4	5	辣	lat	4	la:t	4b	t-split:length:	
4	6	扎	tʂat	4	tsa:t	4c	t-split:ac-split:deret:length:	
4	7	杂	dzap	4	tsa:p	4b	t-split:devoi:length:	
4	8	擦	ts ^h at	4	ts ^h a:t	4c	t-split:ac-split:length:	
4	9	洒	ga	2	sa:	2a	t-split:deret:length:	
5	1	渣	tʂa	1	tsa:	1a	t-split:deret:length:	
5	2	札	tʂap	4	tsa:t	4c	t-split:ac-split:deret:length:[p-t]:	
5	3	乍	dʒap	4	tsa:p	4b	t-split:devoi:length:	
5	4	炸	dʒiap	4	tsa:	3a	[t4-t3]:[t-split]:devoi:r-unglide:deret:length:	
5	5	札	tʂat	4	tsa:t	4c	t-split:ac-split:deret:length:	
5	6	侧	dʒat	4	tsa:t	4c	[t-split]:ac-split:devoi:deret:length:	
5	7	炸	tʂa	3	tsa:	3a	t-split:deret:length:	
5	8	炸	tʂa	3	tsa:	3a	:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
5	9	榨	tʂa	3	tʂa:	3a	t-split:deret:length:	
6	1	乍	dʒa	3	tʂa:	3b	t-split:devoi:deret:length:	
6	2	叉	tʂ ^h a	1	ts ^h a:	1a	t-split:deret:length:	
6	3	差	tʂ ^h a	1	ts ^h a:	1a	t-split:deret:length:	
6	4	插	tʂ ^h ap	4	ts ^h a:p	4c	t-split:ac-split:deret:length:	
6	5	茶	ʈa	1	ts ^h a:	1b	t-split:fricat:asp:devoi:retrof:deret:length:	
6	6	搽	ʈa	1	ts ^h a:	1b	t-split:fricat:asp:devoi:retrof:deret:length:	
6	7	查	dʒa	1	ts ^h a:	1b	t-split:asp:devoi:deret:length:	
6	8	察	tʂ ^h at	4	ts ^h a:t	4c	t-split:ac-split:deret:length:	
6	9	岔	tʂ ^h a	3	ts ^h a:	3a	t-split:asp:deret:length:	
7	1	差	tʂ ^h a	3	ts ^h a:	1a	[t3-t1]:t-split:asp:deret:length:	
7	2	纱	ʂa	1	sa:	1a	t-split:deret:length:	
7	3	杀	ʂa	1	sa:	1a	t-split:deret:length:	
7	4	杀	ʂat	4	sa:t	4c	t-split:deret:length:	
7	5	傻	ʂua	2	sɔ	1b	[t2-t1]:[t-split]:r-unglide:deret:[rnd]:	
7	6	加	ka	1	ka:	1a	t-split:length:	
7	7	嘉	ka	1	ka:	1a	t-split:length:	
7	8	家	ka	1	ka:	1a	t-split:length:	
7	9	佳	kai	1	ka:i	1a	t-split:length:	
8	1	夹	kap	4	ka:p	4c	t-split:ac-split:length:	
8	2	夹	kap	4	ka:p	4c	t-split:ac-split:length:	
8	3	假	ka	2	ka:	2a	t-split:length:	
8	4	贾	ka	2	ka:	2a	t-split:length:	
8	5	甲	kap	4	ka:p	4c	t-split:ac-split:length:	
8	6	假	ka	3	ka:	3a	t-split:length:	
8	7	稼	ka	3	ka:	3a	t-split:length:	
8	8	嫁	ka	3	ka:	3a	t-split:length:	
8	9	架	ka	3	ka:	3a	t-split:length:	
9	1	骂	ka	3	ka:	3a	t-split:length:	
9	2	价	ka	3	ka:	3a	t-split:length:	
9	3	拍	k ^h ap	4	ha:p	4c	t-split:ac-split:spirant:length:x-weak:	
9	4	卡			k ^h a	1a	:	
9	5	恰	k ^h ap	4	hap	4a	t-split:spirant:x-weak:[!ac-split]:	
9	6	虾	xa	1	ha:	1b	[t-split]:x-weak:length:	
9	7	瞎	xat	4	hat	4b	[t-split]:x-weak:	
9	8	霞	ya	1	ha:	1b	t-split:devoi:x-weak:length:	
9	9	匣	yap	4	kap	4a	[t-split]:devoi:[x-kh]:[de-asp]:[!ac-split]:	
10	1	辖	yat	4	hat	4b	t-split:devoi:x-weak:	
10	2	吓	xac	4	ha:k	4c	t-split:ac-split:x-weak:length:fusion:	
10	3	下	ya	2	ha:	2b	t-split:devoi:x-weak:length:	
10	4	下	ya	3	ha:	3b	t-split:devoi:x-weak:length:	
10	5	夏	ya	3	ha:	3b	t-split:devoi:x-weak:length:	
10	6	鸦	?a	1	a:	1a	t-split:length:pro-a:	
10	7	押	?ap	4	ɲa:p	4c	t-split:ac-split:length:pro-a:pro-m:	
10	7	鸭	?ap	4	ɲa:p	4c	t-split:ac-split:length:pro-a:pro-m:	
10	8	压	?ap	4	ɲa:t	4a	t-split:length:pro-a:pro-m:[p-t]:[!ac-split]:	
10	9	牙	ɲa	1	ɲa:	1b	t-split:length:	
11	1	芽	ɲa	1	ɲa:	1b	t-split:length:	
11	2	雅	ɲa	2	a:	1a	[t2-t1]:[t-split]:length:[ng-del]:	
11	3	哑	?a	2	a:	2a	t-split:length:pro-a:	
11	4	亚	?a	3	a:	3a	t-split:length:pro-a:	
11	5	抓	tʂau	1	tsa:u	2a	[t1-t2]:[t-split]:deret:length:	
11	6	爪	tʂau	2	tsa:u	2a	t-split:deret:length:	
11	7	刷	ɲuat	4	tsa:t	4c	t-split:ac-split:r-unglide:deret:length:[s-ts]:	
11	8	耍	ɲua	2	sa:	2a	t-split:r-unglide:deret:length:	
11	9	瓜	kua	1	kwa:	1a	t-split:lab-vel:length:	
12	1	刚	kuat	4	kwa:t	4c	t-split:ac-split:lab-vel:length:	
12	2	寡	kua	2	kwa:	2a	t-split:lab-vel:length:	
12	3	卦	kuai	3	kwa:	3a	t-split:lab-vel:length:[i-del]:	
12	3	挂	kuai	3	kwa:	3a	t-split:lab-vel:length:[i-del]:	
12	4	夸	k ^h ua	1	k ^h wa:	1a	t-split:lab-vel:length:	
12	5	垮	k ^h ua	2	k ^h wa:	1a	[t2-t1]:t-split:lab-vel:length:	
12	6	跨	k ^h ua	3	k ^h wa:	3a	t-split:lab-vel:length:	
12	7	花	xua	1	fa:	1a	t-split:lab-vel:f-coal:length:	
12	8	华	yua	1	wa:	1b	t-split:length:pro-b:glide:	
12	9	滑	yuat	4	wa:t	4b	t-split:length:pro-b:glide:	
13	1	化	xua	3	fa:	3a	t-split:lab-vel:f-coal:length:	
13	2	画	yuai	3	wa:	2b	[t3-t2]:t-split:length:pro-b:glide:[i-del]:	
13	3	话	yuai	3	wa:	3b	t-split:length:pro-b:glide:[i-del]:	
13	4	划	yuaç	4	wa:k	4b	t-split:length:fusion:pro-b:glide:	
13	5	挖	?uat	4	wa:t	4c	t-split:ac-split:length:pro-a:glide:	
13	6	瓦	ɲua	2	ɲa:	2b	t-split:[unglide]:length:	
13	7	辣	ɲyat	4	ma:t	4b	t-split:lab-dnt':lab-dism:length:	
13	8	得	tak	4	tak	4a	t-split:io-flip:	
13	9	德	tak	4	tak	4a	t-split:io-flip:	
14	1	特	dak	4	tak	4b	t-split:devoi:io-flip:	
14	2	乐	lak	4	lok	4b	t-split:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
14	3	勒	lək	4	lak	4b	t-split:io-flip:	
14	4	则	tsək	4	tsak	4a	t-split:io-flip:	
14	5	择	ɖac	4	tsa:k	4b	t-split:fricat:devoi:retrof:deret:length:fusion:	
14	6	泽	ɖac	4	tsa:k	4b	t-split:fricat:devoi:retrof:deret:length:fusion:	
14	7	责	tɕac	4	tsa:k	4a	t-split:deret:length:fusion:[!ac-split]:	
14	8	侧	tɕiək	4	tsak	4a	t-split:r-unglide:deret:io-flip:	
14	9	测	tɕ ^h iək	4	ts ^h ak	4a	t-split:r-unglide:deret:io-flip:	
15	1	厕	tɕ ^h iək	4	ts ^h ak	4a	t-split:r-unglide:deret:io-flip:	
15	2	策	tɕ ^h iək	4	ts ^h a:k	4c	t-split:ac-split:deret:length:pal-dism:fusion:	
15	3	册	tɕ ^h ac	4	ts ^h a:k	4c	t-split:ac-split:deret:length:fusion:	
15	4	涩	ɕiap	4	sap	4a	t-split:r-unglide:deret:io-flip:	
15	5	瑟	ɕiat	4	sat	4a	t-split:r-unglide:deret:io-flip:	
15	6	普	ɕiək	4	sik	4a	t-split:deret:color:	
15	7	色	ɕiək	4	sik	4a	t-split:deret:color:	
15	8	遮	tɕia	1	tɕe	1a	t-split:sharp:retrof:deret:unglide:	
15	9	哲	ɕiat	4	tsit	4c	t-split:ac-split:fricat:retrof:deret:io-flip:rime:	
16	1	折	tɕiat	4	tsit	4c	t-split:ac-split:retrof:deret:io-flip:rime:	
16	2	者	tɕia	2	tɕe	2a	t-split:sharp:retrof:deret:unglide:	
16	3	浙	tɕiat	4	tsit	4c	t-split:ac-split:retrof:deret:io-flip:rime:	
16	4	丰	tɕ ^h ia	1	ts ^h ɛ	1a	t-split:sharp:retrof:deret:unglide:	
16	5	扯	tɕ ^h ia	2	ts ^h ɛ	2a	t-split:sharp:r-unglide:deret:	
16	6	彻	tɕ ^h iat	4	ts ^h it	4c	t-split:ac-split:fricat:retrof:deret:io-flip:rime:	
16	6	撤	tɕ ^h iat	4	ts ^h it	4c	t-split:ac-split:fricat:retrof:deret:io-flip:rime:	
16	7	蛇	dɕia	1	sɛ	1b	t-split:affric:devoi:sharp:retrof:deret:unglide:	
16	8	舌	dɕiat	4	sit	4b	t-split:affric:devoi:retrof:deret:io-flip:rime:	
16	9	舍	ɕia	2	sɛ	2a	t-split:sharp:retrof:deret:unglide:	
17	1	敫	ɕia	3	sɛ	3a	t-split:sharp:retrof:deret:unglide:	
17	2	舍	ɕia	3	sɛ	3a	t-split:sharp:retrof:deret:unglide:	
17	3	社	ɕia	2	sɛ	2b	t-split:devoi:sharp:retrof:deret:unglide:	
17	4	射	dɕia	3	sɛ	3b	t-split:affric:devoi:sharp:retrof:deret:unglide:	
17	5	涉	ɕiap	4	sip	4c	[t-split]:ac-split:devoi:retrof:deret:io-flip:rime:	
17	6	设	ɕiat	4	tsit	4c	t-split:ac-split:retrof:deret:io-flip:rime:[s-ts]:	
17	7	惹	riia	2	je	2b	t-split:sharp:pro-d:glide:	
17	8	热	riiat	4	jit	4b	t-split:io-flip:rime:pro-d:glide:	
17	9	歌	ka	1	kɔ	1a	t-split:rnd:	
18	1	鸽	kap	4	kap	4c	t-split:ac-split:[back.a-fronted]:	
18	2	割	kat	4	kɔt	4c	t-split:ac-split:rnd:	
18	3	阁	kak	4	kɔk	4c	t-split:ac-split:rnd:	
18	4	格	kac	4	ka:k	4c	t-split:ac-split:length:fusion:	
18	5	隔	kac	4	ka:k	4c	t-split:ac-split:length:fusion:	
18	6	革	kac	4	ka:k	4c	t-split:ac-split:length:fusion:	
18	7	个	ka	3	kɔ	3a	t-split:rnd:	
18	8	各	kak	4	kɔk	4c	t-split:ac-split:rnd:	
18	9	科	k ^h ua	1	fɔ	1a	t-split:spirant:lab-vel:f-coal:rnd:	
19	1	棵	k ^h ua	1	fɔ	1a	t-split:spirant:lab-vel:f-coal:rnd:	
19	2	颗	k ^h ua	2	fɔ	2a	t-split:spirant:lab-vel:f-coal:rnd:	
19	3	壳	k ^h ak	4	hɔk	4c	t-split:ac-split:spirant:x-weak:back-asm:rnd:	
19	4	可	k ^h a	2	hɔ	2a	t-split:spirant:x-weak:rnd:	
19	5	渴	k ^h at	4	hot	4a	t-split:spirant:x-weak:rnd:[!ac-split]:	
19	6	课	k ^h ua	3	fɔ	3a	t-split:spirant:lab-vel:f-coal:rnd:	
19	7	克	k ^h ək	4	hak	4a	t-split:spirant:x-weak:io-flip:	
19	8	刻	k ^h ək	4	hak	4a	t-split:spirant:x-weak:io-flip:	
19	9	客	k ^h ac	4	ha:k	4c	t-split:ac-split:spirant:x-weak:length:fusion:	
20	1	何	ɣa	1	hɔ	1b	t-split:devoi:x-weak:rnd:	
20	2	河	ɣa	1	hɔ	1b	t-split:devoi:x-weak:rnd:	
20	3	荷	ɣa	1	hɔ	1b	t-split:devoi:x-weak:rnd:	
20	4	禾	ɣua	1	wɔ	1b	t-split:rnd:pro-b:glide:	
20	5	和	ɣua	1	wɔ	1b	t-split:rnd:pro-b:glide:	
20	6	合	ɣap	4	hap	4b	t-split:devoi:x-weak:[back.a-fronted]:	
20	7	盒	ɣap	4	hap	4b	t-split:devoi:x-weak:[back.a-fronted]:	
20	8	核	ɣac	4	hak	4b	t-split:devoi:x-weak:fusion:	
20	9	贺	ɣa	3	hɔ	3b	t-split:devoi:x-weak:rnd:	
21	1	俄	ɣa	1	ɣɔ	1b	t-split:rnd:	
21	1	俄	ɣa	1	ɣɔ	1b	t-split:rnd:	
21	1	鹈	ɣa	1	ɣɔ	1b	t-split:rnd:	
21	2	讹	ɣua	1	ɣɔ	1b	t-split:unglide:rnd:	
21	3	饿	ɣac	4	ɣa:k	4b	t-split:ler:gh:fusion:	
21	4	饿	ɣa	3	ɣɔ	3b	t-split:rnd:	
21	5	悉	ʔak	4	ɔk	4c	t-split:ac-split:rnd:pro-a:	
21	6	波	pua	1	pɔ	1a	t-split:lab-dism:rnd:	
21	7	播	pua	1	pɔ	3a	[t1-t3]:t-split:lab-dism:rnd:	
21	8	玻	p ^h ua	1	pɔ	1a	t-split:lab-dism:rnd:[de-asp]:	
21	9	拨	puat	4	put	4b	[t-split]:lab-dism:rnd:	rnd-adj:
22	1	剥	pak	4	pɔk	4a	t-split:back-asm:rnd:[!ac-split]:	
22	2	博	pak	4	pɔk	4c	t-split:ac-split:rnd:	
22	3	驳	pak	4	pɔk	4c	t-split:ac-split:back-asm:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
22	4	伯	pac	4	pak	4c	t-split:ac-split:fusion:	
22	5	簸	pua	2	pɔ	2a	t-split:lab-dism:rnd:	
22	6	簸	pua	3	pɔ	3a	t-split:lab-dism:rnd:	
22	7	坡	p ^h ua	1	p ^h ɔ	1a	t-split:lab-dism:rnd:	
22	8	颇	p ^h ua	1	p ^h ɔ	1a	t-split:lab-dism:rnd:	
22	9	浚	p ^h uat	4	p ^h ut	4a	t-split:lab-dism:rnd:[lac-split]:	rnd-adj:
23	1	婆	bua	1	p ^h ɔ	1b	t-split:asp:devoi:lab-dism:rnd:	
23	2	颇	p ^h ua	3	p ^h ɔ	3a	t-split:lab-dism:rnd:	
23	3	迫	pac	4	pik	4a	t-split:fusion:[a-i]:[lac-split]:	
23	4	摸	mak	4	mɔ	2a	[t4-t2]:[t-split]:rnd:[k-del]:	
23	5	摩	mua	1	mɔ	1a	[t-split]:lab-dism:rnd:	
23	5	魔	mua	1	mɔ	1a	[t-split]:lab-dism:rnd:	
23	6	磨	mua	1	mɔ	1b	t-split:lab-dism:rnd:	
23	7	模	mo	1	mou	1b	t-split:raise:	brk-b:
23	8	抹	muat	4	mut	4c	[t-split]:ac-split:lab-dism:rnd:	rnd-adj:
23	9	磨	mua	3	mɔ	3b	t-split:lab-dism:rnd:	
24	1	末	muat	4	mut	4c	[t-split]:ac-split:lab-dism:rnd:	rnd-adj:
24	2	沫	muat	4	mut	4b	t-split:lab-dism:rnd:	rnd-adj:
24	3	没	muat	4	mut	4b	t-split:rime:	
24	4	奥	mak	4	mək	4b	t-split:rnd:	
24	5	墨	mək	4	mək	4b	t-split:io-flip:	
24	6	默	mək	4	mək	4b	t-split:io-flip:	
24	7	佛	vyət	4	fat	4b	t-split:devoi:lab-dnt':lab-dism:io-flip:	
24	8	多	ta	1	tɔ	1a	t-split:rnd:	
24	9	夺	duat	4	tyt	4b	t-split:devoi:rnd:	rnd-adj:degem:acute:
25	1	朵	tua	2	tɔ	2a	t-split:rnd:unlide:	
25	2	躲	tua	2	tɔ	2a	t-split:rnd:unlide:	
25	3	惰	dua	2	tɔ	3b	t-split:t2-merge:devoi:rnd:unlide:	
25	4	拖	t ^h a	1	t ^h ɔ	1a	t-split:rnd:	
25	5	它	t ^h a	1	t ^h a:	1a	t-split:length:	
25	6	脱	t ^h uat	4	t ^h yt	4c	t-split:ac-split:rnd:	rnd-adj:degem:acute:
25	7	托	t ^h ak	4	t ^h ɔk	4c	t-split:ac-split:rnd:	
25	7	托	t ^h ak	4	t ^h ɔk	4c	t-split:ac-split:rnd:	
25	8	驮	da	1	t ^h ɔ	1b	t-split:asp:devoi:rnd:	
25	8	驮	da	1	t ^h ɔ	1b	t-split:asp:devoi:rnd:	
25	9	妥	t ^h ua	2	t ^h ɔ	2b	[t-split]:rnd:unlide:	
26	1	挪	na	1	nɔ	1b	t-split:rnd:	
26	2	糯	nua	3	nɔ	3a	[t-split]:rnd:unlide:	
26	3	罗	la	1	lɔ	1b	t-split:rnd:	
26	3	罗	la	1	lɔ	1b	t-split:rnd:	
26	4	萝	la	1	lo	1b	t-split:rnd:	
26	5	骡	lua	1	lɔ	1b	t-split:rnd:unlide:	
26	6	洛	lak	4	lɔk	4c	t-split:ac-split:rnd:	
26	6	洛	lak	4	lɔk	4c	t-split:ac-split:rnd:	
26	6	洛	lak	4	lɔk	4c	t-split:ac-split:rnd:	
26	7	落	lak	4	lɔk	4b	t-split:rnd:	
26	8	昨	dzak	4	tsɔk	4b	t-split:devoi:rnd:	
26	9	凿	dzak	4	tsɔk	4b	t-split:devoi:rnd:	
27	1	左	tɕa	2	tsɔ	2a	t-split:rnd:	
27	2	坐	dzua	2	ts ^h ɔ	3b	t-split:t2-merge:asp:devoi:rnd:unlide:	
27	3	座	dzua	2	tsɔ	3b	t-split:t2-merge:devoi:rnd:unlide:	
27	4	做	tso	3	tsou	3b	[t-split]:raise:	brk-b:
27	5	作	tsak	4	tsɔk	4c	t-split:ac-split:rnd:	
27	6	搓	ts ^h a	1	ts ^h ɔ	1a	t-split:rnd:	
27	7	错	ts ^h o	3	ts ^h ɔ	3a	t-split:	
27	8	梭	sua	1	sɔ	1a	t-split:rnd:unlide:	
27	9	梭	sua	1	sɔ	1a	t-split:rnd:unlide:	
28	1	缩	giuk	4	suk	4a	t-split:unlide:deret:	
28	2	锁	sua	2	sɔ	2a	t-split:rnd:unlide:	
28	3	琐	sua	2	sɔ	2a	t-split:rnd:unlide:	
28	4	所	gio	2	sɔ	2a	t-split:r-unlide:deret:	
28	5	索	sak	4	sɔk	4c	t-split:ac-split:rnd:	
28	6	拙	tɕyat	4	tsyt	4c	t-split:ac-split:retrof:deret:io-flip:rime:	
28	7	卓	ʔak	4	tsɔk	4c	t-split:ac-split:fricat:retrof:deret:[a-oe]:	
28	8	捉	tɕak	4	tsuk	4c	t-split:ac-split:deret:back-asm:rnd:	rnd-adj:
28	9	着	diak	4	tsɔk	4b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:color:	
29	1	浊	ɖak	4	tsuk	4b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:	rnd-adj:
29	2	镪	dzak	4	tsuk	4b	t-split:devoi:deret:back-asm:rnd:	rnd-adj:
29	3	戮	t ^h ak	4	ts ^h ɔk	4c	t-split:ac-split:fricat:retrof:deret:[a-oe]:	
29	4	说	ɕyat	4	syt	4c	t-split:ac-split:retrof:deret:io-flip:rime:	
29	5	若	niak	4	jɔk	4c	[t-split]:ac-split:back-asm:rnd:color:pro-d:glide-ins:	
29	6	弱	niak	4	jɔk	4b	t-split:back-asm:rnd:color:pro-d:glide-ins:	
29	7	钢	kua	1	wɔ	1a	t-split:lab-vel:rnd:[kw-w]:	
29	8	郭	kuak	4	kwɔk	4c	t-split:ac-split:lab-vel:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
29	9	国	kuək	4	kwək	4c	t-split:ac-split:lab-vel:io-flip:back-asm:rnd:	
30	1	果	kua	2	kwə	2a	t-split:lab-vel:rnd:	
30	1	囊	kua	2	kwə	2a	t-split:lab-vel:rnd:	
30	2	过	kua	3	kwə	3a	t-split:lab-vel:rnd:	
30	3	阔	k ^h uat	4	fut	4c	t-split:ac-split:spirant:lab-vel:f-coal:rnd:	rnd-adj:
30	4	括	kuat	4	k ^h ut	4c	t-split:ac-split:rnd:[asp]:	rnd-adj:degem:
30	5	扩	k ^h ak	4	k ^h wək	4a	t-split:rnd:[x-lab]:[!ac-split]:	
30	6	活	yuat	4	ut	4b	t-split:rnd:pro-b:	rnd-adj:degem:
30	7	火	xua	2	fə	2a	t-split:lab-vel:f-coal:rnd:	
30	8	伙	xua	2	fə	2a	t-split:lab-vel:f-coal:rnd:	
30	9	货	xua	3	fə	3a	t-split:lab-vel:f-coal:rnd:	
31	1	祸	yua	2	wə	3b	t-split:t2-merge:rnd:pro-b:glide:	
31	2	和	yua	3	wə	3b	t-split:rnd:pro-b:glide:	
31	3	或	yuək	4	wək	4b	t-split:io-flip:pro-b:glide:	
31	4	获	yua	4	wək	4b	t-split:fusion:back-asm:rnd:pro-b:glide:	
31	5	窝	?ua	1	wə	1a	t-split:glide:rnd:	
31	6	我	ŋa	2	ŋə	2b	t-split:rnd:	
31	7	我	ŋua	3	ŋə	3a	t-split:rnd:unnglide:	
31	8	握	?ak	4	ak	4a	t-split:pro-a:[!ac-split]:	
31	9	别	piat	4	pit	4c	t-split:ac-split:io-flip:rime:	
32	1	撇	biat	4	pit	4b	t-split:devoi:io-flip:rime:	
32	2	撇	p ^h iat	4	p ^h it	4a	t-split:io-flip:rime:[!ac-split]:	
32	3	灭	mia	4	mit	4b	t-split:io-flip:rime:	
32	4	爹	fia	1	tɛ	4a	[t1-t4]:t-split:retrof:deret:io-flip:unnglide:[!fricat]:	
32	5	跌	diat	4	tit	4a	[t-split]:devoi:io-flip:rime:[!ac-split]:	
32	6	蝶	diap	4	tip	4b	t-split:devoi:io-flip:rime:	
32	6	蝶	diap	4	tip	4b	t-split:devoi:io-flip:rime:	
32	7	叠	diap	4	tip	4b	t-split:devoi:io-flip:rime:	
32	8	碟	diap	4	tip	4b	t-split:devoi:io-flip:rime:	
32	9	贴	t ^h iap	4	t ^h ip	4c	t-split:ac-split:io-flip:rime:	
33	1	帖	t ^h iap	4	t ^h ip	4c	t-split:ac-split:io-flip:rime:	
33	2	铁	t ^h iat	4	t ^h it	4c	t-split:ac-split:io-flip:rime:	
33	3	捏	niat	4	nip	4b	t-split:io-flip:rime:[t-p]:	
33	4	猎	liap	4	lip	4b	t-split:io-flip:rime:	
33	5	列	liat	4	lit	4b	t-split:io-flip:rime:	
33	5	烈	liat	4	lit	4b	t-split:io-flip:rime:	
33	6	裂	liat	4	lit	4b	t-split:io-flip:rime:	
33	7	劣	lyat	4	lyt	4c	[t-split]:ac-split:io-flip:rime:	
33	8	管	kai	1	ka:i	1a	t-split:length:	
33	9	阶	kai	1	ka:i	1a	t-split:length:	
34	1	街	kai	1	ka:i	1a	t-split:length:	
34	2	接	tsiap	4	tsip	4c	t-split:ac-split:io-flip:rime:	
34	3	结	kiat	4	kit	4c	t-split:ac-split:io-flip:rime:	
34	4	杰	giat	4	kit	4b	t-split:devoi:io-flip:rime:	
34	5	节	tsiat	4	tsit	4c	t-split:ac-split:io-flip:rime:	
34	6	洁	kiat	4	kit	4c	t-split:ac-split:io-flip:rime:	
34	7	截	dziat	4	tsit	4b	t-split:devoi:io-flip:rime:	
34	8	姐	tsia	2	tse	2a	t-split:sharp:unnglide:	
34	9	解	kai	2	ka:i	2a	t-split:length:	
35	1	借	tsia	3	tse	3a	t-split:sharp:unnglide:	
35	2	介	kai	3	ka:i	3a	t-split:length:	
35	3	戒	kai	3	kai	3a	t-split:	
35	3	界	kai	3	kai	3a	t-split:	
35	3	芥	kai	3	kai	3a	t-split:	
35	4	疥	kai	3	kai	3a	t-split:	
35	5	切	ts ^h iat	4	ts ^h it	4c	t-split:ac-split:io-flip:rime:	
35	6	茄	gia	1	k ^h ɛ	1b	t-split:asp:devoi:sharp:unnglide:	
35	7	且	ts ^h ia	2	ts ^h ɛ	2a	t-split:sharp:unnglide:	
35	8	怯	k ^h iap	4	hip	4c	t-split:ac-split:spirant:x-weak:io-flip:rime:	
35	9	些	sia	1	se	1a	t-split:sharp:unnglide:	
36	1	歌	xiat	4	hit	4c	t-split:ac-split:x-weak:io-flip:rime:	
36	2	邪	zia	1	tse	1b	t-split:affric:devoi:sharp:unnglide:	
36	3	鞋	ɣai	1	ha:i	1b	t-split:devoi:length:	
36	4	协	ɣiap	4	hip	4b	t-split:devoi:io-flip:rime:	
36	5	写	sia	2	se	2a	t-split:sharp:unnglide:	
36	6	血	xyat	4	hyt	4c	t-split:ac-split:x-weak:io-flip:rime:	
36	7	泻	sia	3	se	3a	t-split:sharp:unnglide:	
36	8	卸	sia	3	se	3a	t-split:sharp:unnglide:	
36	9	谢	zia	3	tse	3b	t-split:affric:devoi:sharp:unnglide:	
37	1	械	ɣai	3	ha:i	3b	t-split:devoi:length:	
37	2	蟹	ɣai	2	ha:i	2b	t-split:devoi:length:	
37	3	泄	sia	4	sit	4c	t-split:ac-split:io-flip:rime:	
37	4	噎	?iat	4	jit	4c	t-split:ac-split:io-flip:rime:pro-a:glide-ins:	
37	5	也	ia	2	ja:	2b	t-split:glide:length:	
37	6	野	ia	2	je	2b	t-split:glide:sharp:	
37	7	夜	ia	3	je	1b	[t3-t1]:t-split:glide:sharp:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
37	8	叶	iap	4	jip	4b	t-split:glide-ins:io-flip:rime:	
37	9	业	ɲiap	4	jip	4b	t-split:io-flip:rime:pro-c:glide-ins:	
38	1	疔	ɲiak	4	jœk	4b	t-split:pro-c:glide-ins:back-asm:rnd:color:	
38	1	虐	ɲiak	4	jœk	4b	t-split:pro-c:glide-ins:back-asm:rnd:color:	
38	2	略	liak	4	lœk	4b	t-split:back-asm:rnd:color:	
38	3	绝	dzyat	4	tsyt	4b	t-split:devoi:io-flip:rime:	
38	4	獬	gyat	4	kwat	4b	t-split:devoi:y-split:lab-vel:	
38	5	决	kyat	4	k ^h yt	4a	t-split:io-flip:rime:[asp]:[!ac-split]:	
38	6	觉	kak	4	kœk	4c	t-split:ac-split:back-asm:rnd:	
38	7	缺	k ^h iat	4	k ^h yt	4c	t-split:ac-split:io-flip:rnd:	
38	8	却	k ^h iak	4	k ^h œk	4c	t-split:ac-split:back-asm:rnd:color:	
38	9	确	k ^h ak	4	k ^h œk	4c	t-split:ac-split:rnd:	
39	1	鞞	xya	1	hœ	1a	t-split:x-weak:sharp:color:	
39	2	鞞	sia	4	sit	4c	t-split:ac-split:io-flip:rime:	
39	3	削	sia	4	sœk	4c	t-split:ac-split:back-asm:rnd:color:	
39	4	学	yak	4	hœk	4b	t-split:devoi:back-asm:rnd:	
39	5	雪	syat	4	syt	4c	t-split:ac-split:io-flip:rime:	
39	6	约	ɲiak	4	jœk	4c	t-split:ac-split:pro-a:glide-ins:back-asm:rnd:color:	
39	7	悦	yat	4	jyt	4b	t-split:glide-ins:io-flip:rime:	
39	8	阅	yat	4	jyt	4b	t-split:glide-ins:io-flip:rime:	
39	9	月	ɲyat	4	jyt	4b	t-split:io-flip:rime:pro-c:glide-ins:	
40	1	越	yat	4	jyt	4b	t-split:glide-ins:io-flip:rime:	
40	2	岳	ɲak	4	ɲœk	4b	t-split:back-asm:rnd:	
40	3	乐	ɲak	4	ɲœk	4b	t-split:back-asm:rnd:	
40	4	资	tsi	1	tsi	1a	t-split:	
40	5	姿	tsi	1	tsi	1a	t-split:	
40	6	兹	tsi	1	tsi	1a	t-split:	
40	7	滋	tsi	1	tsi	1a	t-split:	
40	8	紫	tsi	2	tsi	2a	t-split:	
40	9	姊	tsi	2	tsi	2a	t-split:	
41	1	子	tsi	2	tsi	2a	t-split:	
41	2	自	dzi	3	tsi	3b	t-split:devoi:	
41	3	字	dzi	3	tsi	3b	t-split:devoi:	
41	4	峰	ts ^h i	1	ts ^h i	1a	t-split:	
41	5	差	ts ^h i	1	ts ^h i	1a	t-split:	
41	6	瓷	dzi	1	ts ^h i	1b	t-split:asp:devoi:	
41	7	慈	dzi	1	ts ^h i	1b	t-split:asp:devoi:	
41	7	磁	dzi	1	ts ^h i	1b	t-split:asp:devoi:	
41	8	辞	zi	1	ts ^h i	1b	t-split:affric:asp:	
41	9	祠	zi	1	ts ^h i	1b	t-split:affric:asp:	
41	9	河	zi	1	ts ^h i	1b	t-split:affric:asp:	
42	1	此	ts ^h i	2	ts ^h i	2a	t-split:	
42	2	刺	ts ^h i	3	ts ^h i	3a	t-split:	
42	3	次	ts ^h i	3	ts ^h i	3a	t-split:	
42	4	司	si	1	si	1a	t-split:	
42	5	斯	si	1	si	1a	t-split:	
42	5	斯	si	1	si	1a	t-split:	
42	6	私	si	1	si	1a	t-split:	
42	7	丝	si	1	si	1a	t-split:	
42	8	思	si	1	si	1a	t-split:	
42	9	死	si	2	si	2a	t-split:	
43	1	赐	si	3	ts ^h i	3a	t-split:[asp]:[s-ts]:	
43	2	四	si	3	si	3a	t-split:	
43	3	肆	si	3	si	3a	t-split:	
43	4	似	zi	2	ts ^h i	2b	t-split:affric:asp:	
43	5	祀	zi	2	si	3b	t-split:t2-merge:devoi:	
43	6	寺	zi	3	tsi	3b	t-split:affric:devoi:	
43	7	伺	zi	3	tsi	3b	t-split:affric:devoi:	
43	8	知	ti	1	tsi	1a	t-split:fricat:retrof:deret:	
43	9	痴	ti	1	tsi	1a	t-split:fricat:retrof:deret:	
44	1	支	tái	1	tsi	1a	t-split:retrof:deret:	
44	2	枝	tái	1	tsi	1a	t-split:retrof:deret:	
44	2	肢	tái	1	tsi	1a	t-split:retrof:deret:	
44	3	脂	tái	1	tsi	1a	t-split:retrof:deret:	
44	4	之	tái	1	tsi	1a	t-split:retrof:deret:	
44	5	芝	tái	1	tsi	1a	t-split:retrof:deret:	
44	6	汁	táiap	4	tsap	4a	t-split:retrof:deret:io-flip:unglide:	
44	7	织	táiak	4	tsik	4a	t-split:retrof:deret:rime:	
44	8	只	táiak	4	tsœk	4c	t-split:ac-split:sharp:pal-dism:fusion:	
44	9	执	táiap	4	tsap	4a	t-split:retrof:deret:io-flip:unglide:	
45	1	侄	áiœt	4	tsat	4b	t-split:fricat:devoi:retrof:deret:io-flip:unglide:	
45	2	直	áiœk	4	tsik	4b	t-split:fricat:devoi:retrof:deret:rime:	
45	3	值	áiœk	4	tsik	4b	t-split:fricat:devoi:retrof:deret:rime:	
45	4	植	ziœk	4	tsik	4b	t-split:affric:devoi:retrof:deret:rime:	
45	5	职	táiak	4	tsik	4a	t-split:retrof:deret:rime:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
45	6	纸	tʃi	2	tʃi	2a	t-split:retrof:deret:	
45	7	只	tʃi	2	tʃi	2a	t-split:retrof:deret:	
45	8	旨	tʃi	2	tʃi	2a	t-split:retrof:deret:	
45	9	指	tʃi	2	tʃi	2a	t-split:retrof:deret:	
46	1	址	tʃi	2	tʃi	2a	t-split:retrof:deret:	
46	1	止	tʃi	2	tʃi	2a	t-split:retrof:deret:	
46	2	制	tʃiai	3	tsai	3a	t-split:retrof:deret:pal-dism:	
46	3	智	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	4	致	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	5	至	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	6	置	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	7	志	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	7	志	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	8	埭	tʃi	3	tʃi	3a	t-split:retrof:deret:	
46	9	治	tʃi	3	tʃi	3b	t-split:retrof:deret:	
47	1	质	tʃiət	4	tsat	4a	t-split:retrof:deret:io-flip:unslide:	
47	2	吃	k ^h iac	4	hek	4c	t-split:ac-split:spirant:x-weak:sharp:pal-dism:fusion:	
47	3	池	tʃi	1	ts ^h i	1b	t-split:asp:fricat:devoi:retrof:deret:	
47	4	驰	tʃi	1	ts ^h i	1b	t-split:asp:fricat:devoi:retrof:deret:	
47	5	迟	tʃi	1	ts ^h i	1b	t-split:asp:fricat:devoi:retrof:deret:	
47	6	匙	tʃi	1	ts ^h i	1b	t-split:asp:fricat:devoi:retrof:deret:	
47	7	耻	tʃi	2	ts ^h i	2a	t-split:asp:fricat:devoi:retrof:deret:	
47	8	齿	ts ^h i	2	ts ^h i	2a	t-split:retrof:deret:	
47	9	尺	ts ^h iac	4	ts ^h ek	4c	t-split:ac-split:sharp:retrof:deret:pal-dism:fusion:	
48	1	翅	tʃi	3	ts ^h i	3a	t-split:retrof:deret:[asp]:[s-ts]:	
48	2	赤	ts ^h iac	4	tsik	4c	t-split:ac-split:retrof:deret:io-flip:rime:fusion:[de-asp]:	
48	3	施	tʃi	1	si	1a	t-split:retrof:deret:	
48	4	师	tʃi	1	si	1a	t-split:retrof:deret:	
48	5	狮	tʃi	1	si	1a	t-split:retrof:deret:	
48	6	尸	tʃi	1	si	1a	t-split:retrof:deret:	
48	7	诗	tʃi	1	si	1a	t-split:retrof:deret:	
48	8	湿	tʃiəp	4	sap	4a	t-split:retrof:deret:io-flip:unslide:	
48	9	虱	tʃiət	4	sat	4a	t-split:retrof:deret:io-flip:unslide:	
49	1	失	tʃiət	4	sat	4a	t-split:retrof:deret:io-flip:unslide:	
49	2	时	tʃi	1	si	1b	t-split:devoi:retrof:deret:	
49	3	十	tʃiəp	4	sap	4b	t-split:devoi:io-flip:unslide:	
49	4	拾	tʃiəp	4	sap	4b	t-split:devoi:io-flip:unslide:	
49	5	实	dʒiət	4	sat	4b	t-split:affric:devoi:io-flip:unslide:	
49	6	食	dʒiək	4	sik	4b	t-split:affric:devoi:rime:	
49	7	蚀	dʒiək	4	sik	4b	t-split:affric:devoi:rime:	
49	8	识	tʃiək	4	sik	4a	t-split:retrof:deret:rime:	
49	9	石	tʃiac	4	sək	4b	t-split:devoi:sharp:retrof:deret:pal-dism:fusion:	
50	1	矢	tʃi	2	ts ^h i	2a	t-split:retrof:deret:[asp]:[s-ts]:	
50	2	屎	tʃi	2	si	2a	t-split:retrof:deret:	
50	3	史	tʃi	2	si	2a	t-split:retrof:deret:	
50	4	使	tʃi	2	si	2a	t-split:retrof:deret:	
50	5	驶	tʃi	2	sai	2a	t-split:retrof:deret:[a-ins]:	
50	6	始	tʃi	2	ts ^h i	2a	t-split:retrof:deret:[asp]:[s-ts]:	
50	7	世	tʃiai	3	sai	3a	t-split:retrof:deret:io-flip:pal-dism:	brk-d:
50	7	势	tʃiai	3	sai	3a	t-split:retrof:deret:io-flip:pal-dism:	brk-d:
50	8	誓	tʃiai	3	sai	3b	t-split:devoi:retrof:deret:io-flip:pal-dism:	brk-d:
50	9	是	tʃi	2	si	3b	t-split:t2-merge:devoi:retrof:deret:	
51	1	示	dʒi	3	si	3b	t-split:affric:devoi:retrof:deret:	
51	2	视	tʃi	3	si	3b	t-split:devoi:retrof:deret:	
51	3	士	dʒi	2	si	3b	t-split:t2-merge:devoi:deret:[asp]:	
51	4	柿	dʒi	2	si	3b	t-split:t2-merge:devoi:deret:[asp]:	
51	5	事	dʒi	3	si	3b	t-split:asp:devoi:deret:	
51	6	试	tʃi	3	si	3a	t-split:retrof:deret:	
51	7	市	tʃi	2	si	2b	t-split:devoi:retrof:deret:	
51	8	恃	tʃi	2	si	2a	[t-split]:devoi:retrof:deret:	
51	9	室	tʃiət	4	sat	4a	t-split:retrof:deret:io-flip:unslide:	
52	1	式	tʃiək	4	sik	4a	t-split:retrof:deret:rime:	
52	2	适	tʃiac	4	sik	4a	t-split:retrof:deret:io-flip:rime:fusion:[lac-split]:	
52	3	日	njət	4	jat	4b	t-split:io-flip:pro-d:glide:	
52	4	儿	nj	1	ji	1a	[t-split]:pro-d:glide-ins:	
52	5	而	nj	1	ji	1b	t-split:pro-d:glide-ins:	
52	6	耳	nj	2	ji	2b	t-split:pro-d:glide-ins:	
52	7	二	nj	3	ji	3b	t-split:pro-d:glide-ins:	
52	8	遍	piək	4	pik	4a	t-split:rime:	
52	9	鼻	bi	3	pei	3b	t-split:devoi:	brk-a:
53	1	彼	pi	2	pei	2a	t-split:	brk-a:
53	2	比	pi	2	pei	2a	t-split:	brk-a:
53	3	鄙	pi	2	p ^h ei	2b	[t-split]:[asp]:	brk-a:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
53	4	笔	piət	4	pat	4a	t-split:io-flip:unglide:	
53	5	蔽	piai	3	pai	3a	t-split:io-flip:pal-dism:	brk-d:
53	6	币	biai	3	pai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
53	6	弊	biai	3	pai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
53	6	蔽	biai	3	pai	3a	[t-split]:devoi:io-flip:pal-dism:	brk-d:
53	7	毙	biai	3	pai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
53	8	闭	piai	3	pai	3a	t-split:io-flip:pal-dism:	brk-d:
53	9	臂	pi	3	pei	3a	t-split:	brk-a:
54	1	避	bi	3	pei	3b	t-split:devoi:	brk-a:
54	2	蓖	bi	3	pei	3a	[t-split]:devoi:	brk-a:
54	3	毕	piət	4	pat	4a	t-split:io-flip:unglide:	
54	4	必	piət	4	pit	4a	t-split:rime:	
54	5	壁	piac	4	pik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
54	6	壁	piac	4	pik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
54	7	批	p ^h iai	1	p ^h ai	1a	t-split:io-flip:pal-dism:	brk-d:
54	8	披	p ^h i	1	p ^h ei	1a	t-split:	brk-a:
54	9	野	p ^h iac	4	p ^h ik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
55	1	皮	bi	1	p ^h ei	1b	t-split:asp:devoi:	brk-a:
55	2	疲	bi	1	p ^h ei	1b	t-split:asp:devoi:	brk-a:
55	3	脾	bi	1	p ^h ei	1b	t-split:asp:devoi:	brk-a:
55	4	匹	p ^h iət	4	p ^h at	4a	t-split:io-flip:unglide:	
55	5	譬	p ^h i	3	p ^h ei	3a	t-split:	brk-a:
55	6	屁	p ^h i	3	p ^h ei	3a	t-split:	brk-a:
55	7	迷	miai	1	mai	1b	t-split:io-flip:pal-dism:	brk-d:
55	8	谜	miai	1	mai	1b	t-split:io-flip:pal-dism:	brk-d:
55	9	米	miai	2	mai	2b	t-split:io-flip:pal-dism:	brk-d:
56	1	秘	pi	3	pei	3a	t-split:	brk-a:
56	2	密	miət	4	mat	4b	t-split:io-flip:unglide:	
56	3	蜜	miət	4	mat	4b	t-split:io-flip:unglide:	
56	4	觅	miac	4	mik	4b	t-split:io-flip:rime:fusion:	
56	5	低	tiai	1	tai	1a	t-split:pal-dism:	
56	6	堤	tiai	1	t ^h ai	1b	[t-split]:pal-dism:[asp]:	
56	7	滴	tiaç	4	tik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
56	8	的	tiaç	4	tik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
56	9	敌	diac	4	tik	4b	t-split:devoi:io-flip:rime:fusion:	
57	1	狄	diac	4	tik	4b	t-split:devoi:io-flip:rime:fusion:	
57	1	笛	diac	4	tik	4b	t-split:devoi:io-flip:rime:fusion:	
57	2	底	tiai	2	tai	2a	t-split:io-flip:pal-dism:	brk-d:
57	3	抵	tiai	2	tai	2a	t-split:io-flip:pal-dism:	brk-d:
57	3	抵	tiai	2	tai	2a	t-split:io-flip:pal-dism:	brk-d:
57	4	帝	tiai	3	tai	3a	t-split:io-flip:pal-dism:	brk-d:
57	5	弟	diai	2	tai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
57	6	第	diai	3	tai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
57	7	递	diai	3	tai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
57	8	地	di	3	tei	3b	t-split:devoi:	brk-a:
57	9	梯	t ^h iai	1	t ^h ai	1a	t-split:pal-dism:	
58	1	踢	t ^h iac	4	t ^h ek	4c	t-split:ac-split:sharp:pal-dism:fusion:	
58	2	剔	t ^h iac	4	t ^h ik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
58	3	题	diai	1	t ^h ai	1b	t-split:devoi:io-flip:pal-dism:	brk-d:
58	4	提	diai	1	t ^h ai	1b	t-split:asp:devoi:io-flip:pal-dism:	brk-d:
58	5	啼	diai	1	t ^h ai	1b	t-split:asp:devoi:io-flip:pal-dism:	brk-d:
58	6	蹄	diai	1	t ^h ai	1b	t-split:asp:devoi:io-flip:pal-dism:	brk-d:
58	7	体	t ^h iai	2	t ^h ai	2a	t-split:io-flip:pal-dism:	brk-d:
58	8	替	t ^h iai	3	t ^h ai	3a	t-split:io-flip:pal-dism:	brk-d:
58	9	涕	t ^h iai	3	t ^h ai	3a	t-split:io-flip:pal-dism:	brk-d:
59	1	剃	t ^h iai	3	t ^h ai	3a	t-split:io-flip:pal-dism:	brk-d:
59	2	泥	niai	1	nai	1b	t-split:io-flip:pal-dism:	brk-d:
59	3	尼	ni	1	nei	1b	t-split:	brk-a:
59	4	你	ni	2	nei	2b	t-split:	brk-a:
59	5	腻	ni	3	nei	3b	t-split:	brk-a:
59	6	逆	ɲiac	4	ɲa:k	4b	t-split:length:pal-dism:fusion:	
59	7	翠	liai	1	lai	1b	t-split:io-flip:pal-dism:	brk-d:
59	8	黎	liai	1	lai	1b	t-split:io-flip:pal-dism:	brk-d:
59	9	离	li	1	lei	1b	t-split:	brk-a:
60	1	离	li	1	lei	1b	t-split:	brk-a:
60	2	梨	li	1	lei	1b	t-split:	brk-a:
60	3	厘	li	1	lei	1b	t-split:	brk-a:
60	4	理	li	1	lei	1b	t-split:	brk-a:
60	5	礼	liai	2	lai	2b	t-split:io-flip:pal-dism:	brk-d:
60	6	理	li	2	lei	2b	t-split:	brk-a:
60	6	里	li	2	lei	2b	t-split:	brk-a:
60	6	鲤	li	2	lei	2b	t-split:	brk-a:
60	7	李	li	2	lei	2b	t-split:	brk-a:
60	8	里	li	2	lei	2b	t-split:	brk-a:
60	9	例	liai	3	lai	3b	t-split:io-flip:pal-dism:	brk-d:
61	1	厉	liai	3	lai	3b	t-split:io-flip:pal-dism:	brk-d:
61	2	丽	liai	3	lai	3b	t-split:io-flip:pal-dism:	brk-d:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
61	3	求	liai	3	tai	3b	t-split:io-flip:pal-dism:[l-t]:	brk-d:
61	4	利	li	3	lei	3b	t-split:	brk-a:
61	5	痢	li	3	lei	3b	t-split:	brk-a:
61	6	吏	li	3	lei	3b	t-split:	brk-a:
61	7	立	liəp	4	lap	3b	[t4-t3]:t-split:io-flip:unglide:	
61	8	粒	liəp	4	lap	4a	[t-split]:io-flip:unglide:	
61	9	栗	liət	4	ləet	4b	t-split:pre-color:color:	
62	1	力	liək	4	lik	4b	t-split:rime:	
62	2	历	liac	4	lik	4b	t-split:io-flip:rime:fusion:	
62	2	历	liac	4	lik	4b	t-split:io-flip:rime:fusion:	
62	3	鸡	kiai	1	kai	1a	t-split:pal-dism:	
62	4	基	ki	1	kei	1a	t-split:	brk-a:
62	5	饥	ki	1	kei	1a	t-split:	brk-a:
62	6	几	ki	1	kei	1a	t-split:	brk-a:
62	7	机	ki	1	kei	1a	t-split:	brk-a:
62	7	机	ki	1	kei	1a	t-split:	brk-a:
62	7	机	ki	1	kei	1a	t-split:	brk-a:
62	8	积	tsiac	4	tsik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
62	9	激	kiac	4	kik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
63	1	击	kiac	4	kik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
63	2	集	dziəp	4	tsap	4b	t-split:devoi:io-flip:unglide:	
63	3	急	kiəp	4	kap	4a	t-split:io-flip:unglide:	
63	4	级	kiəp	4	k ^h ap	4a	t-split:io-flip:unglide:[asp]:	
63	5	及	giəp	4	k ^h ap	4b	t-split:asp:devoi:io-flip:unglide:	
63	6	疾	dziət	4	tsat	4b	t-split:devoi:io-flip:unglide:	
63	7	吉	kiət	4	kat	4a	t-split:io-flip:unglide:	
63	8	即	tsiək	4	tsik	4a	t-split:rime:	
63	9	极	giək	4	kik	4b	t-split:devoi:rime:	
64	1	育	tsiac	4	tsik	4c	t-split:io-flip:rime:fusion:	
64	2	籍	dziac	4	tsik	4b	t-split:devoi:io-flip:rime:fusion:	
64	3	挤	tsiai	2	tsai	1a	[t2-t1]:t-split:pal-dism:	
64	4	己	ki	2	kei	2a	t-split:	brk-a:
64	5	几	ki	2	kei	2a	t-split:	brk-a:
64	6	给	kiəp	4	k ^h ap	4a	t-split:io-flip:unglide:[asp]:	
64	7	祭	tsiai	3	tsai	3a	t-split:io-flip:pal-dism:	brk-d:
64	7	际	tsiai	3	tsai	3a	t-split:io-flip:pal-dism:	brk-d:
64	8	济	tsiai	3	tsai	3a	t-split:io-flip:pal-dism:	brk-d:
64	9	剂	dziiai	3	tsai	3a	[t-split]:devoi:io-flip:pal-dism:	brk-d:
65	1	计	kiai	3	kai	3a	t-split:io-flip:pal-dism:	brk-d:
65	2	继	kiai	3	kai	3a	t-split:io-flip:pal-dism:	brk-d:
65	3	寄	ki	3	kei	3a	t-split:	brk-a:
65	4	技	gi	2	kei	3b	t-split:devoi:	brk-a:
65	5	妓	gi	2	kei	3b	t-split:devoi:	brk-a:
65	6	纪	ki	2	kei	3a	t-split:	brk-a:
65	7	记	ki	3	kei	3a	t-split:	brk-a:
65	8	忌	gi	3	kei	3b	t-split:devoi:	brk-a:
65	9	既	ki	3	kei	3a	t-split:	brk-a:
66	1	季	kui	3	kwai	3a	t-split:lab-vel:	brk-a:
66	2	迹	tsiac	4	tsik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
66	3	迹	tsiac	4	tsik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
66	4	绩	tsiac	4	tsik	4a	t-split:io-flip:rime:fusion:[!ac-split]:	
66	5	妻	ts ^h iai	1	ts ^h ai	1a	t-split:pal-dism:	
66	6	凄	ts ^h iai	1	ts ^h ai	1a	t-split:pal-dism:	
66	7	欺	k ^h i	1	hei	1a	t-split:spirant:x-weak:	
66	8	七	ts ^h iət	4	ts ^h at	4a	t-split:io-flip:unglide:	
66	9	漆	ts ^h iət	4	ts ^h at	4a	t-split:io-flip:unglide:	
67	1	齐	dziiai	1	ts ^h ai	1b	t-split:devoi:io-flip:pal-dism:	brk-d:
67	2	脐	dziiai	1	ts ^h i	1b	t-split:devoi:io-flip:rime:degem:	
67	3	奇	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	4	骑	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	5	其	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	5	旗	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	5	棋	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	5	麒	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	6	期	gi	1	k ^h ei	1b	t-split:asp:devoi:	brk-a:
67	7	启	k ^h iai	2	k ^h ai	2a	t-split:io-flip:pal-dism:	brk-d:
67	8	起	k ^h i	2	hei	1a	[t2-t1]:t-split:spirant:x-weak:	brk-a:
67	9	岂	k ^h i	2	hei	1a	[t2-t1]:t-split:spirant:x-weak:	brk-a:
68	1	哭	k ^h iai	3	k ^h ai	3a	t-split:io-flip:pal-dism:	brk-d:
68	2	器	k ^h i	3	hei	3a	t-split:spirant:x-weak:	brk-a:
68	3	弃	k ^h i	3	hei	3a	t-split:spirant:x-weak:	brk-a:
68	4	气	k ^h i	3	hei	3a	t-split:spirant:x-weak:	brk-a:
68	4	汽	k ^h i	3	hei	3a	t-split:spirant:x-weak:	brk-a:
68	5	西	siai	1	sai	1a	t-split:io-flip:pal-dism:	brk-d:
68	6	暉	siai	1	sai	1a	t-split:io-flip:pal-dism:	brk-d:
68	7	溪	k ^h iai	1	k ^h ai	1a	t-split:io-flip:pal-dism:	brk-d:
68	8	栖	xi	1	hei	1a	t-split:x-weak:	brk-a:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
68	9	稀	xi	1	hei	1a	t-split:x-weak:	brk-a:
68	9	希	xi	1	hei	1a	t-split:x-weak:	brk-a:
69	1	吸	xiəp	4	k ^h ap	4a	t-split:io-flip:unglide:[x-kh]:	
69	2	息	siak	4	sik	4a	t-split:rime:	
69	3	熄	siak	4	sik	4a	t-split:rime:	
69	4	惜	siac	4	sik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
69	5	析	siac	4	sik	4a	t-split:io-flip:rime:fusion:[lac-split]:	
69	6	习	ziəp	4	tsap	4b	t-split:affric:devoi:io-flip:unglide:	
69	7	席	ziac	4	tsik	4b	t-split:affric:devoi:io-flip:rime:fusion:	
69	8	錫	siac	4	sik	4c	t-split:ac-split:io-flip:rime:	
69	9	洗	siai	2	sai	2a	t-split:io-flip:pal-dism:	brk-d:
70	1	喜	xi	2	hei	2a	t-split:x-weak:	brk-a:
70	2	細	siai	3	sai	3a	t-split:io-flip:pal-dism:	brk-d:
70	3	系	ɣiai	3	hai	3b	t-split:devoi:io-flip:pal-dism:	brk-d:
70	4	戏	xi	3	hei	3a	t-split:x-weak:	brk-a:
70	5	医	ʔi	1	ji	1a	t-split:pro-a:glide:	
70	6	依	ʔi	1	ji	1a	t-split:pro-a:glide:	
70	6	衣	ʔi	1	ji	1a	t-split:pro-a:glide:	
70	7	一	ʔiət	4	jat	4a	t-split:io-flip:pro-a:glide:	
70	8	宜	ŋi	1	ji	1b	t-split:pro-c:glide:	
70	9	移	i	1	ji	2b	[t1-t2]:t-split:glide:	
71	1	爽	i	1	ji	1b	t-split:glide:	
71	1	爽	i	1	ji	1b	t-split:glide:	
71	2	疑	ŋi	1	ji	1b	t-split:pro-c:glide:	
71	3	遺	ui	1	wai	1b	t-split:glide:	brk-a:
71	4	情	ʔi	2	ji	2a	t-split:pro-a:glide-ins:	
71	5	椅	ʔi	2	ji	2a	t-split:pro-a:glide-ins:	
71	6	已	i	2	ji	2b	t-split:glide-ins:	
71	7	以	i	2	ji	2b	t-split:glide-ins:	
71	8	乙	ʔiət	4	jyt	4c	t-split:[ac-split]:pro-a:glide-ins:pre-color:rime:	
71	9	艺	ŋiai	3	ŋai	3b	t-split:pal-dism:	
72	1	义	ŋi	3	ji	3b	t-split:pro-c:glide-ins:	
72	1	议	ŋi	3	ji	2b	[t3-t2]:t-split:pro-c:glide-ins:	
72	2	易	i	3	ji	3b	t-split:glide-ins:	
72	3	意	ʔi	3	ji	3a	t-split:pro-a:glide-ins:	
72	4	异	i	3	ji	3b	t-split:glide-ins:	
72	5	忆	ʔiak	4	jik	4a	t-split:rime:pro-a:glide-ins:	
72	6	益	ʔiac	4	jik	4a	t-split:io-flip:rime:fusion:pro-a:glide-ins:[ac-split]:	
72	7	亦	iac	4	jik	4b	t-split:glide-ins:io-flip:rime:fusion:	
72	8	译	iac	4	jik	4b	t-split:glide-ins:io-flip:rime:fusion:	
72	9	易	iac	4	jik	4b	t-split:glide-ins:io-flip:rime:fusion:	
73	1	役	yac	4	jik	4b	t-split:glide-ins:io-flip:rime:fusion:	
73	1	疫	yac	4	jik	4b	t-split:glide-ins:io-flip:rime:fusion:	
73	2	补	po	2	pou	2a	t-split:raise:	brk-b:
73	3	捕	bo	3	pou	3b	t-split:devoi:raise:	brk-b:
73	4	布	po	3	pou	3a	t-split:raise:	brk-b:
73	5	布	po	3	pou	3a	t-split:raise:	brk-b:
73	6	部	bo	2	pou	3b	t-split:t2-merge:devoi:raise:	brk-b:
73	7	簿	bo	2	pou	3b	t-split:t2-merge:devoi:raise:	brk-b:
73	8	步	bo	3	pou	3b	t-split:devoi:raise:	brk-b:
73	9	埠	bo	3	fau	3b	t-split:[raise]:diph:io-flip:[b-f]:	
74	1	不	fueɬ	4	pat	4a	t-split:lab-dism:io-flip:[f-p]:	
74	2	铺	p ^h o	1	p ^h ou	1a	t-split:raise:	brk-b:
74	3	扑	p ^h uk	4	p ^h ək	4c	t-split:[ac-split]:[u-o]:	
74	4	蒲	bo	1	p ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
74	5	脯	bo	1	p ^h ou	2a	[t1-t2]:[t-split]:asp:devoi:raise:	brk-b:
74	6	葡	bo	1	p ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
74	7	仆	buk	4	p ^h uk	4a	[t-split]:asp:devoi:	
74	8	谱	po	2	p ^h ou	2a	t-split:[asp]:raise:	brk-b:
74	9	普	p ^h o	2	p ^h ou	2a	t-split:raise:	brk-b:
75	1	辅	p ^h o	3	p ^h ou	3a	t-split:raise:	brk-b:
75	2	模	mo	1	mou	1a	[t-split]:raise:	brk-b:
75	3	母	mu	2	mou	2b	t-split:	brk-b:
75	4	亩	mu	2	mau	2b	t-split:diph:io-flip:	
75	5	暮	mo	3	mou	3b	t-split:raise:	brk-b:
75	6	慕	mo	3	mou	3b	t-split:raise:	brk-b:
75	7	墓	mo	3	mou	3b	t-split:raise:	brk-b:
75	8	幕	mak	4	mək	4b	t-split:rnd:	
75	9	木	muk	4	muk	4b	t-split:	
76	1	目	muk	4	muk	4b	t-split:	
76	2	牡	miuk	4	muk	4b	t-split:unglide:	
76	3	肤	fio	1	fu	1a	t-split:raise:unglide:	
76	4	夫	fio	1	fu	1a	t-split:raise:unglide:	
76	5	敷	f ^h io	1	fu	1a	t-split:[asp-r]:raise:unglide:	
76	6	肤	f ^h io	1	fu	1a	t-split:[asp-r]:raise:unglide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
76	7	解	f ^h io	1	fu	1a	t-split:[asp-r]:raise:unlide:	
76	8	浮	viu	1	fau	1b	t-split:devoi:diph:io-flip:unlide:	
76	9	符	vio	1	fu	1b	t-split:devoi:raise:unlide:	
77	1	扶	vio	1	fu	1b	t-split:devoi:raise:unlide:	
77	2	福	fiuk	4	fuk	4a	t-split:unlide:	
77	3	服	viuk	4	fuk	4b	t-split:devoi:unlide:	
77	4	伏	viuk	4	fuk	4b	t-split:devoi:unlide:	
77	5	俯	fio	2	fu	2a	t-split:raise:unlide:	
77	5	府	fio	2	fu	2a	t-split:raise:unlide:	
77	5	腑	fio	2	fu	2a	t-split:raise:unlide:	
77	6	斧	fio	2	fu	2a	t-split:raise:unlide:	
77	7	脯	fio	2	fu	2a	t-split:raise:unlide:	
77	8	甫	fio	2	fu	2a	t-split:raise:unlide:	
77	9	腐	fio	2	fu	3b	t-split:devoi:raise:unlide:	
78	1	辅	vio	2	fu	3b	t-split:devoi:raise:unlide:	
78	2	付	fio	3	fu	3b	[t-split]:raise:unlide:	
78	3	傅	fio	3	fu	3b	[t-split]:raise:unlide:	
78	4	父	vio	2	fu	3b	t-split:t2-merge:devoi:raise:unlide:	
78	5	附	vio	3	fu	3b	t-split:devoi:raise:unlide:	
78	6	富	fiu	3	fu	3a	t-split:unlide:	
78	7	妇	viu	2	fu	2b	t-split:devoi:unlide:	
78	8	负	viu	2	fu	3b	t-split:t2-merge:devoi:unlide:	
78	9	复	fiuk	4	fuk	4a	t-split:unlide:	
79	1	腹	fiuk	4	fuk	4a	t-split:unlide:	
79	2	复	viuk	4	fuk	4b	t-split:devoi:unlide:	
79	3	都	to	1	tou	1a	t-split:raise:	brk-b:
79	4	独	duk	4	tuk	4b	t-split:devoi:	
79	5	读	duk	4	tuk	4b	t-split:devoi:	
79	6	牍	duk	4	tuk	4b	t-split:devoi:	
79	7	毒	duk	4	tuk	4b	t-split:devoi:	
79	8	堵	to	2	tou	2a	t-split:raise:	brk-b:
79	8	赌	to	2	tou	2a	t-split:raise:	brk-b:
79	9	肚	to	2	t ^h ou	2b	[t-split]:raise:[asp]:	brk-b:
80	1	杜	do	2	tou	3b	t-split:t2-merge:devoi:raise:	brk-b:
80	2	肚	do	2	t ^h ou	2b	t-split:asp:devoi:raise:	brk-b:
80	3	度	do	3	tou	3b	t-split:devoi:raise:	brk-b:
80	4	渡	do	3	tou	3b	t-split:devoi:raise:	brk-b:
80	5	镀	do	3	tou	3b	t-split:devoi:raise:	brk-b:
80	6	突	duət	4	tat	4b	t-split:devoi:io-flip:unlide:	
80	7	秃	t ^h uk	4	t ^h uk	4a	t-split:	
80	8	徒	do	1	t ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
80	9	涂	do	1	t ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
80	9	途	do	1	t ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
81	1	图	do	1	t ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
81	2	屠	do	1	t ^h ou	1b	t-split:asp:devoi:raise:	brk-b:
81	3	土	t ^h o	2	t ^h ou	2a	t-split:raise:	brk-b:
81	4	吐	t ^h o	2	t ^h ou	3a	t-split:raise:	brk-b:
81	5	兔	t ^h o	3	t ^h ou	3a	t-split:raise:	brk-b:
81	6	奴	no	1	nou	1b	t-split:raise:	brk-b:
81	7	努	no	2	nou	2b	t-split:raise:	brk-b:
81	8	怒	no	3	nou	3b	t-split:raise:	brk-b:
81	9	炉	lo	1	lou	1b	t-split:raise:	brk-b:
81	9	卢	lo	1	lou	1b	t-split:raise:	brk-b:
81	9	芦	lo	1	lou	1b	t-split:raise:	brk-b:
82	1	檣	lo	2	lou	2b	t-split:raise:	brk-b:
82	1	鲁	lo	2	lou	2b	t-split:raise:	brk-b:
82	2	卤	lo	2	lou	2b	t-split:raise:	brk-b:
82	3	露	lo	3	lou	3b	t-split:raise:	brk-b:
82	4	路	lo	3	lou	3b	t-split:raise:	brk-b:
82	5	鹿	luk	4	luk	4b	t-split:	
82	6	禄	luk	4	luk	4b	t-split:	
82	7	陆	liuk	4	luk	4b	t-split:unlide:	
82	8	录	liuk	4	luk	4b	t-split:unlide:	
82	9	租	tso	1	tsou	1a	t-split:raise:	brk-b:
83	1	卒	tsyət	4	tsət	4a	t-split:color:	
83	2	族	dzuk	4	tsuk	4b	t-split:devoi:	
83	3	足	tsiuk	4	tsuk	4a	t-split:unlide:	
83	4	祖	tso	2	tsou	2a	t-split:raise:	brk-b:
83	5	组	tso	2	tsou	2a	t-split:raise:	brk-b:
83	6	阻	tgio	2	tsə	2a	t-split:deret:unlide:	
83	7	粗	ts ^h o	1	ts ^h ou	1a	t-split:raise:	brk-b:
83	8	醋	ts ^h o	3	ts ^h ou	3a	t-split:raise:	brk-b:
83	9	苏	so	1	sou	1a	t-split:raise:	brk-b:
83	9	酥	so	1	sou	1a	t-split:raise:	brk-b:
84	1	俗	ziuk	4	tsok	4b	t-split:affric:devoi:[ju-o]:	
84	2	嗦	so	3	sou	3a	t-split:raise:	brk-b:
84	2	素	so	3	sou	3a	t-split:raise:	brk-b:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
84	2	诉	so	3	sou	3a	t-split:raise:	brk-b:
84	3	塑	so	3	sou	3a	t-split:raise:	brk-b:
84	4	速	suk	4	ts ^h uk	4a	t-split:[asp]:[s-ts]:	
84	5	宿	siuk	4	suk	4a	t-split:unlide:	
84	6	粟	siuk	4	suk	4a	t-split:unlide:	
84	7	猪	ʃio	1	tsy	1a	t-split:fricat:retrof:deret:raise:y-fuse:	
84	8	诸	tʃio	1	tsy	1a	t-split:retrof:deret:raise:y-fuse:	
84	9	朱	tʃio	1	tsy	1a	t-split:retrof:deret:raise:y-fuse:	
84	9	珠	tʃio	1	tsy	1a	t-split:retrof:deret:raise:y-fuse:	
85	1	蛛	ʃio	1	tsy	1a	t-split:fricat:retrof:deret:raise:y-fuse:	
85	2	株	ʃio	1	tsy	1a	t-split:fricat:retrof:deret:raise:y-fuse:	
85	3	竹	ʃiuk	4	tsuk	4a	t-split:fricat:retrof:deret:unlide:	
85	4	筑	ʃiuk	4	tsuk	4a	t-split:fricat:retrof:deret:unlide:	
85	5	烛	tʃiuk	4	tsuk	4a	t-split:retrof:deret:unlide:	
85	6	煮	tʃio	2	tsy	2a	t-split:retrof:deret:raise:y-fuse:	
85	7	主	tʃio	2	tsy	2a	t-split:retrof:deret:raise:y-fuse:	
85	8	嘴	tʃiuk	4	tsuk	4a	t-split:retrof:deret:unlide:	
85	9	著	ʃio	3	tsy	3a	t-split:fricat:retrof:deret:raise:y-fuse:	
86	1	驻	ʃio	3	tsy	3a	t-split:fricat:retrof:deret:raise:y-fuse:	
86	2	注	ʃio	3	tsy	3a	t-split:fricat:retrof:deret:raise:y-fuse:	
86	3	注	tʃio	3	tsy	3a	t-split:retrof:deret:raise:y-fuse:	
86	4	蛙	tʃio	3	tsy	3a	t-split:retrof:deret:raise:y-fuse:	
86	5	铸	tʃio	3	tsy	3a	t-split:retrof:deret:raise:y-fuse:	
86	6	柱	ʃio	2	ts ^h y	2b	t-split:devoi:fricat:retrof:deret:raise:y-fuse:	
86	7	住	ʃio	3	tsy	3b	t-split:devoi:fricat:retrof:deret:raise:y-fuse:	
86	8	助	dʒio	3	tsə	3b	t-split:devoi:r-unlide:deret:	
86	9	祝	tʃiuk	4	tsək	4a	t-split:retrof:deret:unlide:	
87	1	初	ts ^h io	1	ts ^h ɔ	1a	t-split:r-unlide:deret:	
87	2	出	ts ^h yat	4	ts ^h æ̃t	4a	t-split:retrof:deret:color:	
87	3	除	ʃio	1	ts ^h y	1b	t-split:asp:devoi:fricat:retrof:deret:raise:y-fuse:	
87	4	锄	dʒio	1	ts ^h ɔ	1b	t-split:asp:devoi:r-unlide:deret:	
87	5	厨	ʃio	1	ts ^h y	1b	t-split:asp:devoi:fricat:retrof:deret:raise:y-fuse:	
87	6	储	ʃio	1	ts ^h y	1b	t-split:asp:devoi:fricat:retrof:deret:raise:y-fuse:	
87	7	楚	ts ^h io	2	ts ^h ɔ	2a	t-split:r-unlide:deret:	
87	7	础	ts ^h io	2	ts ^h ɔ	2a	t-split:r-unlide:deret:	
87	8	处	tʃio	2	ts ^h y	2a	t-split:retrof:deret:raise:y-fuse:	
87	9	处	tʃio	3	ts ^h y	3a	t-split:retrof:deret:raise:y-fuse:	
88	1	畜	ʃiuk	4	ts ^h uk	4a	t-split:affric:retrof:deret:unlide:	
88	2	触	tʃiuk	4	ts ^h uk	4a	t-split:retrof:deret:unlide:	
88	3	梳	gio	1	sɔ	1a	t-split:r-unlide:deret:	
88	4	疏	gio	1	sɔ	1a	t-split:r-unlide:deret:	
88	5	书	ʃio	1	sy	1a	t-split:affric:retrof:deret:raise:y-fuse:	
88	6	舒	ʃio	1	sy	1a	t-split:affric:retrof:deret:raise:y-fuse:	
88	7	输	ʃio	1	sy	1a	t-split:affric:retrof:deret:raise:y-fuse:	
88	8	殊	ʃio	1	sy	1b	t-split:affric:devoi:retrof:deret:raise:y-fuse:	
88	9	叔	ʃiuk	4	suk	4a	t-split:retrof:deret:unlide:	
89	1	糜	dʒiuk	4	suk	4b	t-split:devoi:retrof:deret:unlide:[ts-s]:	
89	2	糜	ʃio	2	sy	2a	t-split:retrof:deret:raise:y-fuse:	
89	2	季	ʃio	2	sy	2a	t-split:retrof:deret:raise:y-fuse:	
89	3	鼠	ʃio	2	sy	2a	t-split:retrof:deret:raise:y-fuse:	
89	4	数	gio	2	sou	2a	t-split:r-unlide:deret:raise:	brk-b:
89	5	属	ʒiuk	4	suk	4b	t-split:devoi:retrof:deret:unlide:	
89	6	数	gio	3	sou	2a	[t3-t2]:t-split:r-unlide:deret:raise:	brk-b:
89	7	竖	ʒio	2	sy	3b	t-split:t2-merge:devoi:retrof:deret:raise:y-fuse:	
89	8	树	ʒio	3	sy	3b	t-split:devoi:retrof:deret:raise:y-fuse:	
89	9	漱	giu	3	sou	3a	t-split:r-unlide:deret:	brk-b:
90	1	术	dʒyat	4	sæ̃t	4b	t-split:devoi:retrof:deret:color:[ts-s]:	
90	1	述	dʒyat	4	sæ̃t	4b	t-split:devoi:retrof:deret:color:[ts-s]:	
90	2	如	nio	1	jy	1b	t-split:raise:y-fuse:pro-d:glide-ins:	
90	3	乳	nio	2	jy	2b	t-split:raise:y-fuse:pro-d:glide-ins:	
90	4	入	ń	4	jap	4b	t-split:io-flip:pro-d:glide-ins:	
90	5	辱	niuk	4	juk	4b	t-split:pro-d:glide:	
90	6	褥	niuk	4	juk	4b	t-split:pro-d:glide:	
90	7	孤	kio	1	ku	1a	t-split:unlide:raise:	
90	7	孤	kio	1	ku	1a	t-split:unlide:raise:	
90	8	辜	kio	1	ku	1a	t-split:unlide:raise:	
90	9	估	ko	2	ku	2a	t-split:raise:	
91	1	古	ko	2	ku	2a	t-split:raise:	
91	1	股	ko	2	ku	2a	t-split:raise:	
91	1	鼓	ko	2	ku	2a	t-split:raise:	
91	2	骨	kuət	4	kwat	4a	t-split:lab-vel:io-flip:	
91	3	谷	kuk	4	kuk	4a	t-split:	
91	3	谷	kuk	4	kuk	4a	t-split:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
91	4	故	ko	3	ku	3a	t-split:raise:	
91	5	固	ko	3	ku	3a	t-split:raise:	
91	5	雇	ko	3	ku	3a	t-split:raise:	
91	5	顾	ko	3	ku	3a	t-split:raise:	
91	6	枯	k ^h o	1	fu	1a	t-split:spirant:x-lab:f-coal:raise:	
91	7	哭	k ^h uk	4	huk	4a	t-split:spirant:x-weak:	
91	8	苦	k ^h o	2	fu	2a	t-split:spirant:x-lab:f-coal:raise:	
91	9	库	k ^h o	3	fu	3a	t-split:spirant:x-lab:f-coal:raise:	
91	9	裤	k ^h o	3	fu	3a	t-split:spirant:x-lab:f-coal:raise:	
92	1	呼	xo	1	fu	1a	t-split:x-lab:f-coal:raise:	
92	2	忽	xuət	4	fat	4a	t-split:lab-vel:f-coal:io-flip:	
92	3	胡	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	4	湖	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	5	糊	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	6	狐	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	6	葫	yo	1	wu	1a	[t-split]:raise:pro-b:glide-ins:	
92	7	胡	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	8	壶	yo	1	wu	1b	t-split:raise:pro-b:glide-ins:	
92	9	核	yuət	4	wat	4b	t-split:io-flip:pro-b:glide:	
93	1	虎	xo	2	fu	2a	t-split:x-lab:f-coal:raise:	
93	2	户	yo	2	wu	3b	t-split:t2-merge:raise:pro-b:glide-ins:	
93	3	互	yo	3	wu	3b	t-split:raise:pro-b:glide-ins:	
93	4	互	yo	3	wu	3b	t-split:raise:pro-b:glide-ins:	
93	5	乌	ʔo	1	wu	1a	t-split:raise:pro-b:glide-ins:	
93	6	屋	ʔuk	4	wuk	4a	t-split:pro-a:glide-ins:	
93	7	吴	ŋo	1	ŋ	1b	t-split:raise:nasal-syl:	
93	8	梧	ŋo	1	ŋ	1b	t-split:raise:nasal-syl:	
93	9	无	ŋio	1	mou	1b	t-split:lab-dnt':raise:	brk-b:
94	1	伍	ŋo	2	ŋ	2b	t-split:raise:nasal-syl:	
94	1	午	ŋo	2	ŋ	2b	t-split:raise:nasal-syl:	
94	2	五	ŋo	2	ŋ	2b	t-split:raise:nasal-syl:	
94	3	武	ŋio	2	mou	2b	t-split:lab-dnt':raise:	brk-b:
94	3	舞	ŋio	2	mou	2b	t-split:lab-dnt':raise:	brk-b:
94	4	悉	ʔo	3	wu	3a	t-split:raise:pro-a:glide-ins:	
94	5	误	ŋo	3	ŋ	3b	t-split:raise:nasal-syl:	
94	6	悟	ŋo	3	ŋ	3b	t-split:raise:nasal-syl:	
94	7	务	ŋio	3	mou	3b	t-split:lab-dnt':raise:	brk-b:
94	8	雾	ŋio	3	mou	3b	t-split:lab-dnt':raise:	brk-b:
94	9	物	ŋyət	4	mat	4b	t-split:lab-dnt':lab-dism:io-flip:	
95	1	勿	ŋyət	4	mat	4b	t-split:lab-dnt':lab-dism:io-flip:	
95	2	女	nio	2	nœy	2b	t-split:raise:y-fuse:	brk-c:
95	3	驴	lio	1	lœy	1b	t-split:raise:y-fuse:	brk-c:
95	4	吕	lio	2	lœy	2b	t-split:raise:y-fuse:	brk-c:
95	5	旅	lio	2	lœy	2b	t-split:raise:y-fuse:	brk-c:
95	6	虚	lio	3	lœy	3b	t-split:raise:y-fuse:	brk-c:
95	7	滤	lio	3	lœy	3b	t-split:raise:y-fuse:	brk-c:
95	8	律	lyət	4	lœt	4b	t-split:color:	
95	9	绿	liuk	4	luk	4b	t-split:unlide:	
96	1	居	kio	1	kœy	1a	t-split:raise:y-fuse:	brk-c:
96	2	拘	kio	1	k ^h œy	1a	t-split:asp:raise:y-fuse:	brk-c:
96	3	狗	kio	1	k ^h œy	1a	t-split:asp:raise:y-fuse:	brk-c:
96	4	橘	kyət	4	kuat	4a	t-split:y-split:io-flip:	
96	5	菊	kiuk	4	kuk	4a	t-split:unlide:	
96	6	局	giuk	4	kuk	4b	t-split:devoi:unlide:	
96	7	举	kio	2	kœy	1a	[t2-t1]:t-split:raise:y-fuse:	brk-c:
96	8	据	kio	3	kœy	3a	t-split:raise:y-fuse:	brk-c:
96	9	锯	kio	3	kœy	3a	t-split:raise:y-fuse:	brk-c:
97	1	巨	gio	2	kœy	3b	t-split:t2-merge:devoi:raise:y-fuse:	brk-c:
97	1	拒	gio	2	kœy	2b	t-split:devoi:raise:y-fuse:	brk-c:
97	1	距	gio	2	kœy	2b	t-split:devoi:raise:y-fuse:	brk-c:
97	2	聚	dzio	2	tsœy	3b	t-split:t2-merge:raise:y-fuse:	brk-c:
97	3	句	kio	3	kœy	3a	t-split:raise:y-fuse:	brk-c:
97	4	具	gio	3	kœy	3b	t-split:devoi:raise:y-fuse:	brk-c:
97	5	剧	giac	4	k ^h ek	4b	t-split:asp:devoi:sharp:pal-dism:fusion:	
97	6	蛆	ts ^h io	1	ts ^h œy	1a	t-split:raise:y-fuse:	brk-c:
97	7	区	k ^h io	1	k ^h œy	1a	t-split:raise:y-fuse:	brk-c:
97	7	驱	k ^h io	1	k ^h œy	1a	t-split:raise:y-fuse:	brk-c:
97	8	屈	k ^h yət	4	wat	4a	t-split:y-split:x-lab:io-flip:[kw-w]:	
97	9	曲	k ^h iuk	4	kuk	4a	t-split:unlide:[de-asp]:	
98	1	曲	k ^h iuk	4	k ^h uk	4a	t-split:unlide:	
98	2	渠	gio	1	k ^h œy	1b	t-split:asp:devoi:raise:y-fuse:	brk-c:
98	3	取	ts ^h io	2	ts ^h œy	2a	t-split:raise:y-fuse:	brk-c:
98	4	娶	ts ^h io	2	ts ^h œy	2a	t-split:raise:y-fuse:	brk-c:
98	5	去	k ^h io	3	hœy	3a	t-split:asp:x-weak:raise:y-fuse:	brk-c:
98	6	趣	ts ^h io	3	ts ^h œy	3a	t-split:raise:y-fuse:	brk-c:
98	7	嘘	xio	1	hœy	1a	t-split:x-weak:raise:y-fuse:	brk-c:
98	7	虚	xio	1	hœy	1a	t-split:x-weak:raise:y-fuse:	brk-c:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
98	8	需	sio	1	scey	1a	t-split:raise:y-fuse:	brk-c:
98	8	须	sio	1	scey	1a	t-split:raise:y-fuse:	brk-c:
98	9	须	sio	1	scey	1a	t-split:raise:y-fuse:	brk-c:
99	1	徐	zio	1	ts ^h cey	1b	t-split:affric:asp:devoi:raise:y-fuse:	brk-c:
99	2	许	xio	2	hoe y	2a	t-split:x-weak:raise:y-fuse:	brk-c:
99	3	絮	sio	3	scey	3b	[t-split]:raise:y-fuse:	brk-c:
99	4	叙	zio	2	tscey	3b	t-split:affric:devoi:raise:y-fuse:	brk-c:
99	4	序	zio	2	tscey	3b	t-split:affric:devoi:raise:y-fuse:	brk-c:
99	4	绪	zio	2	tscey	2b	t-split:affric:devoi:raise:y-fuse:	brk-c:
99	5	续	ziuk	4	tsuk	4b	t-split:affric:devoi:unglide:	
99	6	媪	siai	3	sai	3b	[t-split]:io-flip:pal-dism:	brk-d:
99	7	恤	syət	4	sət	4a	t-split:color:	
99	8	蓄	xiuk	4	ts ^h uk	4a	t-split:unglide:[x-tsh]:	
99	8	蓄	xiuk	4	ts ^h uk	4b	[t-split]:unglide:[x-tsh]:	
99	9	鱼	gio	1	jy	1b	t-split:raise:y-fuse:pro-c:glide-ins:	
100	1	余	io	1	jy	1b	t-split:glide-ins:raise:y-fuse:	
100	1	余	io	1	jy	1b	t-split:glide-ins:raise:y-fuse:	
100	2	愚	gio	1	jy	1b	t-split:raise:y-fuse:pro-c:glide-ins:	
100	3	虞	gio	1	jy	1b	t-split:raise:y-fuse:pro-c:glide-ins:	
100	4	娱	gio	1	jy	1b	t-split:raise:y-fuse:pro-c:glide-ins:	
100	5	于	jio	1	jy	1b	t-split:raise:y-fuse:	
100	6	于	jio	1	jy	1b	t-split:raise:y-fuse:	
100	7	愉	io	1	jy	1b	t-split:glide-ins:raise:y-fuse:	
100	8	愉	io	1	jy	1b	t-split:glide-ins:raise:y-fuse:	
100	9	语	gio	2	jy	2b	t-split:raise:y-fuse:pro-c:glide-ins:	
101	1	与	io	2	jy	2b	t-split:glide-ins:raise:y-fuse:	
101	2	雨	jio	2	jy	2b	t-split:raise:y-fuse:	
101	3	与	io	3	jy	3b	t-split:glide-ins:raise:y-fuse:	
101	4	普	io	3	jy	3b	t-split:glide-ins:raise:y-fuse:	
101	4	预	io	3	jy	3b	t-split:glide-ins:raise:y-fuse:	
101	5	遇	gio	3	jy	3b	t-split:raise:y-fuse:pro-c:glide-ins:	
101	6	寓	gio	3	jy	3b	t-split:raise:y-fuse:pro-c:glide-ins:	
101	7	裕	io	3	jy	3b	t-split:glide-ins:raise:y-fuse:	
101	8	喻	io	3	jy	3b	t-split:glide-ins:raise:y-fuse:	
101	9	域	jyək	4	wik	4b	t-split:[pro-j]:glide:[schwa-i]:	
102	1	育	iuk	4	juk	4b	t-split:glide:	
102	2	玉	giuk	4	juk	4b	t-split:pro-c:glide:	
102	3	狱	giuk	4	juk	4b	t-split:pro-c:glide:	
102	4	欲	iuk	4	juk	4b	t-split:glide:	
102	5	白	bac	4	pa:k	4b	t-split:devoi:length:fusion:	
102	6	摆	pai	2	pai:	2a	t-split:length:	
102	7	百	pac	4	pa:k	4c	t-split:ac-split:length:fusion:	
102	8	柏	pac	4	pa:k	4c	t-split:ac-split:length:fusion:	
102	9	拜	pai	3	pai:	3a	t-split:length:	
103	1	裨	bai	3	pai:	3b	t-split:devoi:	
103	2	败	bai	3	pai:	3b	t-split:devoi:length:	
103	3	拍	p ^h ac	4	p ^h ak	4c	t-split:ac-split:length:fusion:	
103	4	排	bai	1	p ^h ai:	1b	t-split:asp:devoi:length:	
103	5	牌	bai	1	p ^h ai:	1b	t-split:asp:devoi:length:	
103	6	派	p ^h ai	3	p ^h ai:	3a	t-split:length:	
103	7	埋	mai	1	mai:	1b	t-split:length:	
103	8	买	mai	2	mai:	2b	t-split:length:	
103	9	迈	mai	3	mai:	3b	t-split:length:	
104	1	迈	mai	3	mai:	3b	t-split:	
104	2	麦	mac	4	mak	4b	t-split:fusion:	
104	3	脉	mac	4	mak	4b	t-split:fusion:	
104	4	太	tai	1	tai:	2a	[t1-t2]:t-split:length:	
104	5	戴	tai	3	tai:	3a	t-split:length:	
104	6	贷	t ^h ai	3	t ^h ai:	3a	t-split:length:	
104	7	待	dai	2	tai:	3b	t-split:t2-merge:devoi:rnd:	
104	8	怠	dai	2	tai:	3b	t-split:t2-merge:devoi:rnd:	
104	9	代	dai	3	tai:	3b	t-split:t2-merge:devoi:rnd:	
105	1	袋	dai	3	tai:	3b	t-split:t2-merge:devoi:rnd:	
105	2	带	tai	3	tai:	3a	t-split:length:	
105	3	胎	t ^h ai	1	t ^h ai:	1a	t-split:rnd:	
105	4	台	dai	1	t ^h ai:	1b	t-split:asp:devoi:rnd:	
105	5	抬	dai	1	t ^h ai:	1b	t-split:asp:devoi:rnd:	
105	6	态	t ^h ai	3	t ^h ai:	3a	t-split:length:	
105	7	太	t ^h ai	3	tai:	3a	t-split:length:[de-asp]:	
105	8	泰	t ^h ai	3	t ^h ai:	3a	t-split:length:	
105	9	乃	nai	2	nai:	2b	t-split:length:	
106	1	奶	niai	2	nai:	2b	t-split:length:pal-dism:	
106	2	耐	nai	3	nai:	3b	t-split:rnd:	
106	3	奈	nai	3	nai:	3b	t-split:rnd:	
106	4	来	lai	1	lai:	1b	t-split:rnd:	
106	5	赖	lai	3	lai:	3b	t-split:length:	
106	6	裁	tsai	1	tsai:	1a	t-split:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
106	6	次	tsai	1	tsɔi	1a	t-split:rnd:	
106	7	宰	tsai	2	tsɔi	2a	t-split:rnd:	
106	8	截	tsai	2	tsɔi	2a	t-split:rnd:	
106	9	在	tsai	3	tsɔi	3a	t-split:rnd:	
107	1	截	tsai	3	tsɔi	3a	t-split:rnd:	
107	2	在	dzai	2	tsɔi	3b	t-split:t2-merge:devoi:rnd:	
107	3	猜	ts ^h ai	1	ts ^h ai	1a	t-split:length:	
107	4	才	dzai	1	ts ^h ai	1b	t-split:asp:devoi:rnd:	
107	4	耐	dzai	1	ts ^h ai	1b	t-split:asp:devoi:rnd:	
107	5	材	dzai	1	ts ^h ai	1b	t-split:asp:devoi:rnd:	
107	6	裁	dzai	1	ts ^h ai	1b	t-split:asp:devoi:rnd:	
107	7	彩	ts ^h ai	2	ts ^h ai	2a	t-split:rnd:	
107	7	采	ts ^h ai	2	ts ^h ai	2a	t-split:rnd:	
107	7	睬	ts ^h ai	2	ts ^h ai	2a	t-split:rnd:	
107	8	菜	ts ^h ai	3	ts ^h ai	3a	t-split:rnd:	
107	9	蔡	ts ^h ai	3	ts ^h ai	3a	t-split:rnd:	
108	1	腮	sai	1	sɔi	1a	t-split:rnd:	
108	2	塞	sək	4	sək	4a	t-split:io-flip:	
108	3	赛	sai	3	ts ^h ai	3a	t-split:rnd:[asp]:[s-ts]:	
108	4	赛	sai	3	ts ^h ai	3a	t-split:rnd:[asp]:[s-ts]:	
108	5	耑	tʂai	1	tsa:i	1a	t-split:deret:length:	
108	6	滴	tiac	4	tsa:k	4b	[t-split]:fricat:retrof:deret:length:pal-dism:fusion:	
108	7	择	ɖac	4	tsa:k	4b	t-split:fricat:devoi:retrof:deret:length:fusion:	
108	8	宅	ɖac	4	tsa:k	4b	t-split:fricat:devoi:retrof:deret:length:fusion:	
108	9	窄	tʂac	4	tsa:k	4c	t-split:ac-split:deret:length:fusion:	
109	1	债	tʂai	3	tsa:i	3a	t-split:deret:length:	
109	2	拆	ts ^h ai	1	ts ^h ai	1a	t-split:length:	
109	3	拆	t ^h ac	4	ts ^h ai	4c	t-split:ac-split:fricat:retrof:deret:length:	
109	4	柴	dzai	1	ts ^h ai	1b	t-split:asp:devoi:deret:length:	
109	5	筛	ʂai	1	sai	1a	t-split:deret:	
109	6	晒	ʂai	3	sai	3a	t-split:deret:length:	
109	7	该	kai	1	kɔi	1a	t-split:rnd:	
109	8	改	kai	2	kɔi	2a	t-split:rnd:	
109	9	概	kai	3	k ^h ɔi	3a	t-split:[back-asm]:rnd:[asp]:	
110	1	益	kai	3	kɔi	3a	t-split:rnd:	
110	2	开	k ^h ai	1	hɔi	1a	t-split:spirant:x-weak:rnd:	
110	3	楷	k ^h ai	1	hɔi	1a	t-split:spirant:x-weak:length:	
110	4	慨	k ^h ai	3	k ^h ɔi	3a	t-split:rnd:	
110	5	孩	ʂai	1	hɔi	1b	t-split:devoi:x-weak:rnd:	
110	6	海	xai	2	hɔi	2a	t-split:x-weak:rnd:	
110	7	害	ʂai	3	hɔi	3b	t-split:devoi:x-weak:rnd:	
110	8	哀	?ai	1	ɔi	1a	t-split:rnd:pro-a:	
110	9	挨	?ai	1	ai	1a	t-split:length:pro-a:	
111	1	矮	?ai	2	ai	2a	t-split:length:pro-a:	
111	2	碍	ŋai	3	ŋɔi	3b	t-split:rnd:	
111	3	艾	ŋai	3	ŋai	3b	t-split:length:	
111	4	爱	?ai	3	ɔi	3a	t-split:rnd:pro-a:	
111	5	衰	ʂui	1	scey	1a	t-split:deret:	acute:
111	6	摔	ʂui	1	scey	1a	t-split:deret:	acute:
111	7	帅	ʂui	3	scey	3a	t-split:deret:	acute:
111	8	率	ʂui	3	scey	4a	t-split:deret:[y-t]:[t3-t4]:	acute:
111	9	乖	kuai	1	kwai	1a	t-split:lab-vel:length:	
112	1	拐	kuai	2	kwai	2a	t-split:lab-vel:length:	
112	2	怪	kuai	3	kwai	3a	t-split:lab-vel:	
112	3	会	kuai	3	k ^h ui	3a	t-split:[rnd]:raise:degem:[asp]:	
112	4	快	k ^h uai	3	fai	3a	t-split:spirant:lab-vel:f-coal:length:	
112	5	快	k ^h uai	3	fai	3a	t-split:spirant:lab-vel:f-coal:length:	
112	5	筷	k ^h uai	3	fai	3a	t-split:spirant:lab-vel:f-coal:length:	
112	6	怀	ʂuai	1	wai	1b	t-split:length:pro-b:glide:	
112	7	槐	ʂuai	1	wai	1b	t-split:length:pro-b:glide:	
112	7	淮	ʂuai	1	wai	1b	t-split:length:pro-b:glide:	
112	8	坏	ʂuai	3	wai	3b	t-split:length:pro-b:glide:	
112	9	歪	xuai	1	wai	1a	t-split:length:[pro-x]:glide:	
113	1	外	ŋuai	3	ŋɔi	3b	t-split:rnd:unglide:	
113	2	杯	puai	1	pui	1a	t-split:lab-dism:rnd:	rnd-adj:
113	3	碑	pi	1	pei	1a	t-split:	brk-a:
113	4	卑	pi	1	pei	1a	t-split:	brk-a:
113	5	悲	pi	1	pei	1a	t-split:	brk-a:
113	6	北	pək	4	pək	4a	t-split:io-flip:	
113	7	贝	pai	3	pui	3a	t-split:rnd:	rnd-adj:
113	8	辈	puai	3	pui	3a	t-split:lab-dism:rnd:	rnd-adj:
113	9	背	pai	3	pui	3a	t-split:rnd:	rnd-adj:
114	1	倍	buai	2	p ^h ui	2b	t-split:asp:devoi:lab-dism:rnd:	rnd-adj:
114	2	被	bi	2	pei	3b	t-split:t2-merge:devoi:	brk-a:
114	3	备	bi	3	pei	3b	t-split:devoi:	brk-a:
114	4	坯	p ^h uai	1	p ^h ui	1b	[t-split]:lab-dism:rnd:	rnd-adj:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
114	5	培	buai	1	p ^h ui	1b	t-split:asp:devoi:lab-dism:rnd:	rnd-adj:
114	6	贍	buai	1	p ^h ui	1b	t-split:asp:devoi:lab-dism:rnd:	rnd-adj:
114	6	陪	buai	1	p ^h ui	1b	t-split:asp:devoi:lab-dism:rnd:	rnd-adj:
114	7	配	p ^h uai	3	p ^h ui	3a	t-split:lab-dism:rnd:	rnd-adj:
114	8	佩	buai	3	p ^h ui	3a	[t-split]:asp:devoi:lab-dism:rnd:	rnd-adj:
114	9	梅	muai	1	mui	1b	t-split:lab-dism:rnd:	rnd-adj:
115	1	枚	muai	1	mui	1b	t-split:lab-dism:rnd:	rnd-adj:
115	2	媒	muai	1	mui	1b	t-split:lab-dism:rnd:	rnd-adj:
115	3	眉	mi	1	mei	1b	t-split:	brk-a:
115	4	霉	mi	1	mui	1b	t-split:	brk-a:[e-u]:
115	5	每	muai	2	mui	2b	t-split:lab-dism:rnd:	rnd-adj:
115	6	美	mi	2	mei	2b	t-split:	brk-a:
115	7	妹	muai	3	mui	3b	t-split:lab-dism:rnd:	rnd-adj:
115	8	味	muai	3	mui	3b	t-split:lab-dism:rnd:	rnd-adj:
115	9	飞	fui	1	fei	1a	t-split:lab-dism:	brk-a:
116	1	非	fui	1	fei	1a	t-split:lab-dism:	brk-a:
116	2	肥	vui	1	fei	1b	t-split:devoi:lab-dism:	brk-a:
116	3	匪	fui	2	fei	2a	t-split:lab-dism:	brk-a:
116	4	废	fyai	3	fai	3a	t-split:lab-dnt':lab-dism:	
116	5	肺	f ^h yai	3	fai	3a	t-split:[asp-r]:lab-dnt':lab-dism:	
116	6	费	f ^h ui	3	fai	3a	t-split:[asp-r]:lab-dism:	[brk-a]:
116	7	内	nuai	3	nɔi	3b	t-split:rnd:unlide:	
116	8	雷	luai	1	lɔey	1b	t-split:rnd:unlide:	rnd-adj:acute:
116	9	播	luai	1	lɔey	1b	t-split:rnd:unlide:	rnd-adj:acute:
117	1	累	lui	2	lɔey	2b	t-split:	acute:
117	2	累	lui	3	lɔey	3b	t-split:	acute:
117	3	类	lui	3	lɔey	3b	t-split:	acute:
117	4	泪	lui	3	lɔey	3b	t-split:	acute:
117	5	肋	lək	4	lak	4b	t-split:io-flip:	
117	6	贼	dzək	4	ts ^h a:k	4b	t-split:asp:devoi:io-flip:[length]:	
117	7	给	kiəp	4	k ^h ap	4a	t-split:io-flip:unlide:[asp]:	
117	8	黑	xək	4	hak	4a	t-split:x-weak:io-flip:	
117	9	堆	tuai	1	tɔey	1a	t-split:rnd:	rnd-adj:degem:acute:
118	1	对	tuai	3	tɔey	3a	t-split:rnd:	rnd-adj:degem:acute:
118	2	队	duai	3	tɔey	3b	t-split:devoi:rnd:	rnd-adj:degem:acute:
118	3	兑	duai	3	tɔey	3b	t-split:devoi:rnd:	rnd-adj:degem:acute:
118	4	推	t ^h uai	1	t ^h ɔey	1a	t-split:rnd:	rnd-adj:degem:acute:
118	5	腿	t ^h uai	2	t ^h ɔey	2a	t-split:rnd:	rnd-adj:degem:acute:
118	6	退	t ^h uai	3	t ^h ɔey	3a	t-split:rnd:	rnd-adj:degem:acute:
118	7	咀	tsui	2	tsɔey	2a	t-split:	acute:
118	8	罪	dzuai	2	tsɔey	3b	t-split:t2-merge:devoi:rnd:	rnd-adj:degem:acute:
118	9	最	tsuai	3	tsɔey	3a	t-split:rnd:	rnd-adj:degem:acute:
119	1	醉	tsui	3	tsɔey	3a	t-split:	acute:
119	2	催	ts ^h uai	1	ts ^h ɔey	1a	t-split:rnd:	rnd-adj:degem:acute:
119	2	催	ts ^h uai	1	ts ^h ɔey	1a	t-split:rnd:	rnd-adj:degem:acute:
119	3	脆	ts ^h yai	3	ts ^h ɔey	3a	t-split:pal-dism:unlide:[rnd]:	rnd-adj:acute:
119	4	翠	ts ^h ui	3	ts ^h ɔey	3a	t-split:	acute:
119	5	虽	sui	1	sɔey	1a	t-split:	acute:
119	6	隋	zui	1	ts ^h ɔey	1a	[t-split]:asp:fricat:devoi:	acute:
119	6	随	zui	1	ts ^h ɔey	1b	t-split:asp:fricat:devoi:	acute:
119	7	碎	suai	3	sɔey	3a	t-split:rnd:	rnd-adj:degem:acute:
119	8	岁	syai	3	sɔey	3a	t-split:pal-dism:unlide:[rnd]:	rnd-adj:acute:
119	9	穗	zui	3	sɔey	3b	t-split:devoi:	acute:
120	1	追	tui	1	tsɔey	1a	t-split:fricat:retrof:deret:	acute:
120	2	锥	tsui	1	tsɔey	1a	t-split:retrof:deret:	acute:
120	3	吹	ts ^h ui	1	ts ^h ɔey	1a	t-split:retrof:deret:	acute:
120	4	炊	ts ^h ui	1	ts ^h ɔey	1b	[t-split]:retrof:deret:	acute:
120	5	垂	zui	1	sɔey	1b	t-split:devoi:retrof:deret:	acute:
120	6	锤	dui	1	ts ^h ɔey	1b	t-split:asp:devoi:retrof:deret:	acute:
120	6	槌	dui	1	ts ^h ɔey	1b	t-split:asp:devoi:retrof:deret:	acute:
120	7	淮	zui	1	sɔey	1b	t-split:devoi:retrof:deret:	acute:
120	8	水	suai	2	sɔey	2a	t-split:retrof:deret:	acute:
120	9	税	syai	3	sɔey	3a	t-split:retrof:deret:pal-dism:unlide:[rnd]:	rnd-adj:acute:
121	1	睡	zui	3	sɔey	3b	t-split:devoi:retrof:deret:	acute:
121	2	蕊	nui	2	jɔey	2b	t-split:pro-d:glide-ins:	acute:
121	3	悦	yai	3	jɔey	3b	t-split:glide:[rnd]:	rnd-adj:acute:
121	4	规	kui	1	k ^h wai	1a	t-split:[asp]:lab-vel:	brk-a:
121	5	龟	kui	1	kwai	1a	t-split:lab-vel:	brk-a:
121	6	归	kui	1	kwai	1a	t-split:lab-vel:	brk-a:
121	7	轨	kui	2	kwai	2a	t-split:lab-vel:	brk-a:
121	8	鬼	kui	2	kwai	2a	t-split:lab-vel:	brk-a:
121	9	桂	kyai	3	kwai	3a	t-split:lab-vel:	brk-a:
122	1	跪	gui	2	kwai	3b	t-split:t2-merge:devoi:lab-vel:	brk-a:
122	2	柜	gui	3	kwai	3b	t-split:devoi:lab-vel:	brk-a:
122	3	亏	kui	3	kwai	3a	t-split:lab-vel:	brk-a:
122	4	亏	k ^h ui	1	k ^h wai	1a	t-split:lab-vel:	brk-a:
122	5	葵	gui	1	k ^h wai	1b	t-split:asp:devoi:lab-vel:	brk-a:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
122	6	愧	kui	3	k ^h wai	3a	t-split:lab-vel:[asp]:	brk-a:
122	7	灰	xuai	1	fui	1a	t-split:lab-vel:f-coal:rnd:	rnd-adj:
122	8	挥	xui	1	fai	1a	t-split:lab-vel:f-coal:	brk-a:
122	8	辉	xui	1	fai	1a	t-split:lab-vel:f-coal:	brk-a:
122	9	徽	xui	1	fai	1a	t-split:lab-vel:f-coal:	brk-a:
123	1	回	ɣuai	1	wui	1b	t-split:rnd:pro-b:glide:	rnd-adj:
123	2	悔	xuai	2	fui	3a	[t2-t3]:t-split:lab-vel:f-coal:rnd:	rnd-adj:
123	3	毁	xui	2	wai	2a	t-split:glide:[pro-x]:	brk-a:
123	4	贿	xuai	2	fui	2a	t-split:lab-vel:f-coal:rnd:	rnd-adj:
123	5	汇	ɣuai	2	wui	3b	t-split:t2-merge:rnd:pro-b:glide:	rnd-adj:
123	6	溃	ɣuai	3	k ^h ui	2a	[t3-t2]:[t-split]:devoi:rnd:unlide:[x-kh]:	rnd-adj:
123	7	慧	ɣuai	3	wai	3b	t-split:pro-b:glide:[back.a-fronted]:	
123	7	会	ɣuai	3	wai	3b	t-split:pro-b:glide:[back.a-fronted]:	
123	8	绘	ɣuai	3	k ^h ui	2a	[t3-t2]:[t-split]:devoi:rnd:unlide:[x-kh]:	rnd-adj:
123	9	惠	ɣyai	3	wai	3b	t-split:pal-dism:pro-b:glide:	
124	1	讳	xui	3	wai	2b	[t3-t2]:[t-split]:glide:[pro-x]:	brk-a:
124	2	微	ɲui	1	mei	1b	t-split:lab-dism:	brk-a:
124	3	威	?ui	1	wai	1a	t-split:pro-a:glide:	brk-a:
124	4	槐	ɣuai	1	wai	1b	t-split:pro-c:glide:[back.a-fronted]:	
124	5	危	ɲui	1	ɲai	1b	t-split:unlide:	[brk-a]:
124	6	为	jui	1	wai	1b	t-split:glide:[pro-j]:	brk-a:
124	7	惟	ui	1	wai	1b	t-split:glide:	brk-a:
124	7	堆	ui	1	wai	1b	t-split:glide:	brk-a:
124	8	违	jui	1	wai	1b	t-split:glide:[pro-j]:	brk-a:
124	9	围	jui	1	wai	1b	t-split:glide:[pro-j]:	brk-a:
125	1	委	?ui	2	wai	2a	t-split:pro-a:glide:	brk-a:
125	2	伪	ɲui	3	ɲai	3b	t-split:unlide:	[brk-a]:
125	3	尾	ɲui	2	mei	2b	t-split:lab-dism:	brk-a:
125	4	伟	jui	2	wai	2b	t-split:glide:[pro-j]:	brk-a:
125	5	苇	jui	2	wai	2b	t-split:glide:[pro-j]:	brk-a:
125	6	卫	jyai	3	wai	3b	t-split:pal-dism:[pro-j]:glide:	brk-a:
125	7	喂	?ui	3	wai	3a	t-split:pro-a:glide:	brk-a:
125	8	为	jui	3	wai	3b	t-split:glide:[pro-j]:	brk-a:
125	9	位	jui	3	wai	3b	t-split:glide:[pro-j]:	brk-a:
126	1	未	ɲui	3	mei	3b	t-split:lab-dism:	brk-a:
126	2	味	ɲui	3	mei	3b	t-split:lab-dism:	brk-a:
126	3	魏	ɲui	3	ɲai	3b	t-split:unlide:	[brk-a]:
126	4	畏	?ui	3	wai	3a	t-split:pro-a:glide:	brk-a:
126	5	胃	jui	3	wai	3b	t-split:glide:[pro-j]:	brk-a:
126	6	谓	jui	3	wai	3b	t-split:glide:[pro-j]:	brk-a:
126	7	谓	jui	3	wai	3b	t-split:glide:[pro-j]:	brk-a:
126	8	衰	pau	1	pou	1a	t-split:rnd:	
126	9	包	pau	1	pa:u	1a	t-split:rnd:	
127	1	薄	bak	4	pək	4b	t-split:devoi:rnd:	
127	2	雹	bak	4	pək	4b	t-split:devoi:back-asm:rnd:	
127	3	宝	pau	2	pou	2a	t-split:rnd:	
127	4	保	pau	2	pou	2a	t-split:rnd:	
127	5	堡	pau	2	pou	2a	t-split:rnd:	
127	6	炮	pau	2	pa:u	2a	t-split:length:	
127	7	报	pau	3	pou	3a	t-split:rnd:	
127	8	抱	bau	2	p ^h ou	2b	t-split:asp:devoi:rnd:	
127	9	暴	bau	3	pou	3b	t-split:devoi:rnd:	
128	1	豹	pau	3	p ^h au	3a	t-split:length:[asp]:	
128	2	爆	pau	3	pa:u	3a	t-split:length:	
128	3	鲍	bau	2	pa:u	3a	[t2-t3]:[t-split]:devoi:length:	
128	4	靴	p ^h au	1	p ^h au	1a	t-split:length:	
128	5	袍	bau	1	p ^h ou	1b	t-split:asp:devoi:rnd:	
128	6	刨	bau	1	p ^h au	1b	t-split:asp:devoi:length:	
128	7	跑	bau	1	p ^h au	2a	[t1-t2]:[t-split]:asp:devoi:length:	
128	8	泡	p ^h au	3	p ^h au	1a	[t3-t1]:t-split:	
128	8	炮	p ^h au	3	p ^h au	3a	t-split:	
128	9	猫	mau	1	ma:u	1a	[t-split]:length:	
129	1	毛	mau	1	mou	1b	t-split:rnd:	
129	2	茅	mau	1	ma:u	1b	t-split:length:	
129	3	锚	mau	1	ma:u	1b	t-split:length:	
129	4	矛	miu	1	ma:u	1b	t-split:lab-dnt:diph:io-flip[length]:	
129	5	卯	mau	2	ma:u	2b	t-split:length:	
129	6	胃	mau	3	mou	3b	t-split:rnd:	
129	7	帽	mau	3	mou	3b	t-split:rnd:	
129	8	貌	mau	3	ma:u	3b	t-split:length:	
129	9	茂	mu	3	mau	3b	t-split:diph:io-flip:	
130	1	贾	mu	3	mau	3b	t-split:diph:io-flip:	
130	2	刀	tau	1	tou	1a	t-split:rnd:	
130	3	岛	tau	2	tou	2a	t-split:rnd:	
130	3	捣	tau	2	tou	2a	t-split:rnd:	
130	4	搏	tau	2	tou	2a	t-split:rnd:	
130	5	倒	tau	2	tou	2a	t-split:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
130	6	导	dau	3	tou	3b	t-split:devoi:rnd:	
130	7	倒	tau	3	tou	3a	t-split:rnd:	
130	8	到	tau	3	tou	3a	t-split:rnd:	
130	9	道	dau	2	tou	3b	t-split:t2-merge:devoi:rnd:	
131	1	稻	dau	2	tou	3b	t-split:t2-merge:devoi:rnd:	
131	2	盗	dau	3	tou	3b	t-split:devoi:rnd:	
131	3	滔	t ^h au	1	t ^h ou	1a	t-split:rnd:	
131	4	掏	t ^h au	1	t ^h ou	1b	[t-split]:rnd:	
131	5	桃	dau	1	t ^h ou	1b	t-split:devoi:rnd:	
131	6	逃	dau	1	t ^h ou	1b	t-split:devoi:rnd:	
131	7	陶	dau	1	t ^h ou	1b	t-split:asp:devoi:rnd:	
131	8	讨	t ^h au	2	t ^h ou	2a	t-split:rnd:	
131	9	套	t ^h au	3	t ^h ou	3a	t-split:rnd:	
132	1	挠	nau	1	nau	1b	t-split:	
132	2	恼	nau	2	nou	2b	t-split:rnd:	
132	2	脑	nau	2	nou	2b	t-split:rnd:	
132	3	闹	nau	3	na:u	3b	t-split:length:	
132	4	捞	lau	1	lou	1b	t-split:rnd:	
132	5	劳	lau	1	lou	1b	t-split:rnd:	
132	6	牢	lau	1	lou	1b	t-split:rnd:	
132	7	老	lau	2	lou	2b	t-split:rnd:	
132	8	涝	lau	3	lou	3b	t-split:rnd:	
132	9	烙	lak	4	lɔk	4b	t-split:rnd:	
132	9	酪	lak	4	lɔk	4c	[t-split]:ac-split:rnd:	
133	1	遭	tsau	1	tsou	1a	t-split:rnd:	
133	2	糟	tsau	1	tsou	1a	t-split:rnd:	
133	3	早	tsau	2	tsou	2a	t-split:rnd:	
133	4	枣	tsau	2	tsou	2a	t-split:rnd:	
133	5	躁	tsau	3	ts ^h ou	3a	t-split:asp:rnd:	
133	6	灶	tsau	3	tsou	3a	t-split:rnd:	
133	7	皂	dzau	2	tsou	3b	t-split:t2-merge:devoi:rnd:	
133	8	造	dzau	2	tsou	3b	t-split:t2-merge:devoi:rnd:	
133	9	操	ts ^h au	1	tsou	1a	t-split:rnd:[de-asp]:	
134	1	髓	ts ^h au	3	ts ^h ou	3a	t-split:rnd:	
134	2	曹	dzau	1	ts ^h ou	1b	t-split:asp:devoi:rnd:	
134	3	槽	dzau	1	ts ^h ou	1b	t-split:asp:devoi:rnd:	
134	4	草	ts ^h au	2	ts ^h ou	2a	t-split:rnd:	
134	5	骚	sau	1	sou	1a	t-split:rnd:	
134	6	臊	sau	1	sou	1a	t-split:rnd:	
134	7	扫	sau	2	sou	2a	t-split:rnd:	
134	8	嫂	sau	2	sou	2a	t-split:rnd:	
134	9	扫	sau	3	sou	3a	t-split:rnd:	
135	1	朝	ʃiau	1	tsiu	1a	t-split:fricat:retrof:deret:io-flip:rime:	
135	2	昭	tʃiau	1	tsiu	1a	t-split:fricat:retrof:deret:io-flip:rime:	
135	3	招	tʃiau	1	tsiu	1a	t-split:fricat:retrof:deret:io-flip:rime:	
135	4	着	ʒiak	4	tsɔk	4b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:color:	
135	5	爪	tʃau	2	tsa:u	2a	t-split:deret:length:	
135	6	找	tʃau	2	tsa:u	2a	t-split:deret:length:	
135	7	罩	ʃau	3	tsa:u	2a	t-split:fricat:retrof:deret:length:	
135	8	赵	ʒiau	2	tsiu	3a	[t2-t3]:[t-split]:devoi:fricat:retrof:deret:io-flip:rime:	
135	9	兆	ʒiau	2	siu	3b	t-split:t2-merge:devoi:fricat:retrof:deret:io-flip:rime:[ts-s]:	
136	1	照	tʃiau	3	tsiu	3a	t-split:devoi:fricat:retrof:deret:io-flip:rime:	
136	2	抄	tʃ ^h au	1	ts ^h a:u	1a	t-split:deret:length:	
136	3	钞	tʃ ^h au	1	ts ^h a:u	1a	t-split:deret:length:	
136	4	超	t ^h iau	1	ts ^h iu	1a	t-split:fricat:retrof:deret:io-flip:rime:	
136	5	巢	dʒau	1	ts ^h a:u	1a	[t-split]:asp:devoi:deret:length:	
136	6	朝	ʒiau	1	ts ^h iu	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:	
136	7	潮	ʒiau	1	ts ^h iu	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:	
136	8	炒	tʃ ^h au	2	ts ^h a:u	2a	t-split:deret:length:	
136	9	吵	tʃ ^h au	2	ts ^h a:u	2a	t-split:deret:length:	
137	1	梢	gau	1	sa:u	1a	t-split:deret:length:	
137	2	稍	gau	1	sa:u	1a	t-split:deret:length:	
137	3	稍	gau	3	sa:u	2a	[t3-t2]:t-split:deret:length:	
137	4	烧	ʃiau	1	siu	1a	t-split:retrof:deret:io-flip:rime:	
137	5	勺	ʒiak	4	tsɔk	4b	t-split:devoi:retrof:deret:back-asm:rnd:color:[s-ts]:	
137	6	少	ʃiau	2	siu	2a	t-split:retrof:deret:io-flip:rime:	
137	7	少	ʃiau	3	siu	3a	t-split:retrof:deret:io-flip:rime:	
137	8	绍	ʒiau	2	siu	3b	t-split:t2-merge:devoi:retrof:deret:io-flip:rime:	
137	9	饶	ʃiau	1	jiu	1b	t-split:io-flip:rime:pro-d:glide-ins:	
138	1	扰	ʃiau	2	jiu	2b	t-split:io-flip:rime:pro-d:glide-ins:	
138	2	绕	ʃiau	2	jiu	2b	t-split:io-flip:rime:pro-d:glide-ins:	

Pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
138	3	糕	kau	1	kou	1a	t-split:rnd:	
138	4	糕	kau	1	kou	1a	t-split:rnd:	
138	4	燕	kau	1	kou	1a	t-split:rnd:	
138	4	膏	kau	1	kou	1a	t-split:rnd:	
138	5	稿	kau	2	kou	2a	t-split:rnd:	
138	6	搞	kau	2	katu	2a	t-split:length:	
138	7	告	kau	3	kou	3a	t-split:rnd:	
138	8	考	k ^h au	2	hatu	2a	t-split:spirant:x-weak:length:	
138	9	烤	k ^h au	2	hatu	2a	t-split:spirant:x-weak:length:	
139	1	靠	k ^h au	3	k ^h atu	3a	t-split:length:	
139	2	蒿	xau	1	hou	1a	t-split:x-weak:rnd:	
139	3	篙	xau	1	hou	1a	t-split:x-weak:rnd:	
139	4	毫	yau	1	hou	1b	t-split:devoi:x-weak:rnd:	
139	4	豪	yau	1	hou	1b	t-split:devoi:x-weak:rnd:	
139	5	好	xau	2	hou	2a	t-split:x-weak:rnd:	
139	6	好	xau	3	hou	3a	t-split:x-weak:rnd:	
139	7	耗	xau	3	hou	3a	t-split:x-weak:rnd:	
139	8	号	yau	3	hou	3b	t-split:devoi:x-weak:rnd:	
139	9	熬	ɣau	1	gou	1b	t-split:rnd:	
140	1	熬	ɣau	1	gou	1b	t-split:rnd:	
140	2	袄	?au	2	ou	2a	t-split:rnd:pro-a:	
140	3	傲	ɣau	3	gou	3b	t-split:rnd:	
140	4	标	piu	1	piu	1a	t-split:io-flip:rime:	
140	5	彪	piu	1	piu	1a	t-split:	
140	6	表	piu	2	piu	2a	t-split:io-flip:rime:	
140	7	缥	biau	2	p ^h iu	2b	t-split:asp:devoi:io-flip:rime:	
140	8	飘	p ^h iau	1	p ^h iu	1a	t-split:io-flip:rime:	
140	9	瓢	biau	1	p ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
141	1	漂	biau	1	p ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
141	2	漂	p ^h iau	2	p ^h iu	3a	[t2-t3]:t-split:io-flip:rime:	
141	3	票	p ^h iau	3	p ^h iu	3a	t-split:io-flip:rime:	
141	4	漂	p ^h iau	3	p ^h iu	3a	t-split:io-flip:rime:	
141	5	苗	miau	1	miu	1b	t-split:io-flip:rime:	
141	6	描	miau	1	miu	1b	t-split:io-flip:rime:	
141	7	秒	miau	2	miu	2b	t-split:io-flip:rime:	
141	8	庙	miau	3	miu	3b	t-split:io-flip:rime:	
141	9	妙	miau	3	miu	3b	t-split:io-flip:rime:	
142	1	刁	tiau	1	tiu	1a	t-split:io-flip:rime:	
142	1	雕	tiau	1	tiu	1a	t-split:io-flip:rime:	
142	2	刁	tiau	1	tiu	1a	t-split:io-flip:rime:	
142	3	钓	tiau	3	tiu	3a	t-split:io-flip:rime:	
142	4	吊	tiau	3	tiu	3a	t-split:io-flip:rime:	
142	5	掉	diau	3	tiu	3b	t-split:devoi:rime:	
142	6	调	diau	3	tiu	3b	t-split:devoi:rime:	
142	7	挑	t ^h iau	1	t ^h iu	1a	t-split:io-flip:rime:	
142	8	条	diau	1	t ^h iu	1a	[t-split]:asp:devoi:io-flip:rime:	
142	9	调	diau	1	t ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
143	1	跳	t ^h iau	3	t ^h iu	3a	t-split:io-flip:rime:	
143	2	鸟	tiau	2	niu	2b	[t-split]:io-flip:rime:[t-n]:	
143	3	尿	niau	3	niu	3b	t-split:io-flip:rime:	
143	4	聊	liau	1	liu	1b	t-split:io-flip:rime:	
143	4	辽	liau	1	liu	1b	t-split:io-flip:rime:	
143	5	疗	liau	3	liu	1b	t-split:io-flip:rime:	
143	6	了	liau	2	liu	2b	t-split:io-flip:rime:	
143	7	料	liau	3	liu	3b	t-split:io-flip:rime:	
143	8	交	kau	1	katu	1a	t-split:length:	
143	8	郊	kau	1	katu	1a	t-split:length:	
143	9	胶	kau	1	katu	1a	t-split:length:	
144	1	教	kau	1	katu	3a	[t1-t3]:t-split:length:	
144	2	焦	tsiau	1	tsiu	1a	t-split:io-flip:rime:	
144	3	椒	tsiau	1	tsiu	1a	t-split:io-flip:rime:	
144	3	蕉	tsiau	1	tsiu	1a	t-split:io-flip:rime:	
144	4	娇	kiau	1	kiu	1a	t-split:io-flip:rime:	
144	4	骄	kiau	1	kiu	1a	t-split:io-flip:rime:	
144	5	浇	kiau	1	kiu	1a	t-split:io-flip:rime:	
144	6	嚼	dziak	4	tsœk	4b	t-split:devoi:back-asm:rnd:color:	
144	7	蛟	kau	2	katu	2a	t-split:length:	
144	8	狡	kau	2	katu	2a	t-split:length:	
144	9	搅	kau	2	katu	2a	t-split:length:	
145	1	脚	kiak	4	kœk	4c	t-split:ac-split:back-asm:rnd:color:	
145	2	角	kak	4	kœk	4c	t-split:ac-split:back-asm:rnd:	
145	3	教	kau	3	katu	3a	t-split:length:	
145	4	酵	kau	3	katu	3a	t-split:length:	
145	5	校	kau	3	ka:u	3a	t-split:length:	
145	6	较	kau	3	ka:u	3a	t-split:length:	
145	7	觉	kau	3	ka:u	3a	t-split:length:	
145	8	窖	kau	3	ka:u	3a	t-split:length:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
145	9	轿	giau	3	kiu	3b	t-split:devoi:io-flip:rime:	
146	1	叫	kiau	3	kiu	3a	t-split:io-flip:rime:	
146	2	敲	k ^h au	1	hau	1a	t-split:spirant:x-weak:	
146	3	嫩	ts ^h iau	1	ts ^h iu	1a	t-split:io-flip:rime:	
146	4	嫩	dziau	1	ts ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
146	5	嫩	dziau	1	ts ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
146	6	侨	giau	1	k ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
146	6	侨	giau	1	k ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
146	7	桥	giau	1	k ^h iu	1b	t-split:asp:devoi:io-flip:rime:	
146	8	巧	k ^h iau	2	hau	2a	t-split:spirant:x-weak:io-flip:rime:	
146	9	鹊	ts ^h iak	4	ts ^h cek	4c	t-split:ac-split:back-asm:rnd:color:	
147	1	窍	k ^h iau	3	k ^h iu	3a	t-split:io-flip:rime:	
147	2	壳	k ^h ak	4	hok	4a	t-split:spirant:x-weak:back-asm:rnd:[!ac-split]:	
147	3	宵	siau	1	siu	1a	t-split:io-flip:rime:	
147	3	消	siau	1	siu	1a	t-split:io-flip:rime:	
147	3	稍	siau	1	siu	1a	t-split:io-flip:rime:	
147	3	霄	siau	1	siu	1a	t-split:io-flip:rime:	
147	4	霄	siau	1	siu	1a	t-split:io-flip:rime:	
147	4	霄	siau	1	siu	1a	t-split:io-flip:rime:	
147	5	削	siax	4	scek	4c	t-split:ac-split:back-asm:rnd:color:	
147	6	小	siau	2	siu	2a	t-split:io-flip:rime:	
147	7	晓	xiau	2	hiu	2a	t-split:x-weak:io-flip:rime:	
147	8	孝	xau	3	ha:u	3a	t-split:x-weak:length:	
147	9	效	yau	3	ha:u	3b	t-split:devoi:x-weak:length:	
148	1	校	yau	3	ha:u	3b	t-split:devoi:x-weak:length:	
148	2	笑	siau	3	siu	3a	t-split:io-flip:rime:	
148	3	妖	?iau	1	jiu	1a	t-split:io-flip:rime:pro-a:glide-ins:	
148	3	要	?iau	1	jiu	1a	t-split:io-flip:rime:pro-a:glide-ins:	
148	3	邀	?iau	1	jiu	1a	t-split:io-flip:rime:pro-a:glide-ins:	
148	4	腰	?iau	1	jiu	1a	t-split:io-flip:rime:pro-a:glide-ins:	
148	5	么	?iau	1	jiu	1a	t-split:io-flip:rime:pro-a:glide-ins:	
148	6	滑	yau	1	gau	1b	t-split:devoi:[pro-x]:pro-m:	
148	6	肴	yau	1	gau	1b	t-split:devoi:[pro-x]:pro-m:	
148	7	摇	iau	1	jiu	1b	t-split:glide-ins:io-flip:rime:	
148	7	窑	iau	1	jiu	1b	t-split:glide-ins:io-flip:rime:	
148	7	姚	iau	1	jiu	1b	t-split:glide-ins:io-flip:rime:	
148	8	遥	iau	1	jiu	1b	t-split:glide-ins:io-flip:rime:	
148	8	咬	iau	1	jiu	1b	t-split:glide-ins:io-flip:rime:	
148	9	咬	gau	2	gau	2b	t-split:	
149	1	召	iau	2	jiu	2b	t-split:glide-ins:io-flip:rime:	
149	2	要	?iau	3	jiu	3a	t-split:io-flip:rime:pro-a:glide-ins:	
149	3	疟	giak	4	joek	4b	t-split:pro-c:glide-ins:back-asm:rnd:color:	
149	4	药	iak	4	joek	4b	t-split:glide-ins:back-asm:rnd:color:	
149	5	朝	iak	4	joek	4b	t-split:glide-ins:back-asm:rnd:color:	
149	6	谋	miu	1	mau	1b	t-split:lab-dnt:diph:io-flip:	
149	7	菜	mu	2	mau	2b	t-split:diph:io-flip:	
149	8	否	fiu	2	fau	2a	t-split:lab-dnt':diph:io-flip:	
149	9	都	to	1	tou	2a	t-split:raise:	brk-b:
150	1	兜	tu	1	tau	1a	t-split:diph:io-flip:	
150	2	斗	tu	2	tau	2a	t-split:diph:io-flip:	
150	3	抖	tu	2	tau	2a	t-split:diph:io-flip:	
150	3	陡	tu	2	tau	2a	t-split:diph:io-flip:	
150	4	斗	tu	3	tau	3a	t-split:diph:io-flip:	
150	5	痘	du	3	tau	3b	t-split:devoi:diph:io-flip:	
150	5	豆	du	3	tau	3b	t-split:devoi:diph:io-flip:	
150	6	偷	t ^h u	1	t ^h au	1a	t-split:diph:io-flip:	
150	7	头	du	1	t ^h au	1b	t-split:devoi:diph:io-flip:	
150	8	投	du	1	t ^h au	1b	t-split:devoi:diph:io-flip:	
150	9	透	t ^h u	3	t ^h au	3a	t-split:diph:io-flip:	
151	1	楼	lu	1	lau	1b	t-split:diph:io-flip:	
151	2	楼	lu	1	lau	2b	[t1-t2]:t-split:diph:io-flip:	
151	3	漏	lu	3	lau	3b	t-split:diph:io-flip:	
151	4	邹	tgiu	1	tsau	1a	t-split:r-unglide:deret:diph:io-flip:	
151	5	走	tsu	2	tsau	2a	t-split:diph:io-flip:	
151	6	奏	tsu	3	tsau	3a	t-split:diph:io-flip:	
151	7	奏	ts ^h u	3	ts ^h au	3a	t-split:diph:io-flip:	
151	8	搜	giu	1	sau	1a	t-split:r-unglide:deret:diph:io-flip:	
151	9	搜	giu	1	sau	1a	t-split:r-unglide:deret:diph:io-flip:	
152	1	周	t ^h iu	1	tsau	1a	t-split:retrof:deret:diph:io-flip:unglide:	
152	1	洲	t ^h iu	1	tsau	1a	t-split:retrof:deret:diph:io-flip:unglide:	
152	1	洲	t ^h iu	1	tsau	1a	t-split:retrof:deret:diph:io-flip:unglide:	
152	2	舟	t ^h iu	1	tsau	1a	t-split:retrof:deret:diph:io-flip:unglide:	
152	2	粥	t ^h iuk	4	tsuk	4a	t-split:retrof:deret:unglide:	
152	3	轴	t ^h iuk	4	tsuk	4b	t-split:devoi:fricat:retrof:deret:unglide:	
152	4	肘	t ^h iu	2	tsau	2a	t-split:fricat:retrof:deret:diph:io-flip:unglide:	
152	5	帚	t ^h iu	2	tsau	2a	t-split:retrof:deret:diph:io-flip:unglide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
152	6	昼	t̥iu	3	tsau	3a	t-split:fricat:retrof:deret:diph:io-flip:un glide:	
152	7	皱	t̥giu	3	tsau	3a	t-split:r-un glide:deret:diph:io-flip:	
152	7	绉	t̥giu	3	tsau	3a	t-split:r-un glide:deret:diph:io-flip:	
152	8	绞	d̥ziu	3	tsau	3b	t-split:devoi:retrof:deret:diph:io-flip:un glide:	
152	9	蹙	t̥h̥iu	1	ts ^h au	1a	t-split:fricat:retrof:deret:diph:io-flip:un glide:	
152	9	抽	t̥h̥iu	1	ts ^h au	1a	t-split:fricat:retrof:deret:diph:io-flip:un glide:	
153	1	绸	diu	1	ts ^h au	1b	t-split:asp:devoi:fricat:retrof:deret:diph:io-flip:un glide:	
153	2	稠	diu	1	ts ^h au	1b	t-split:asp:devoi:fricat:retrof:deret:diph:io-flip:un glide:	
153	3	筹	diu	1	ts ^h au	1b	t-split:asp:devoi:fricat:retrof:deret:diph:io-flip:un glide:	
153	4	愁	diu	1	sau	1b	t-split:devoi:fricat:retrof:deret:diph:io-flip:un glide:[ts-s]:	
153	5	仇	ziu	1	ts ^h au	1b	t-split:asp:devoi:diph:io-flip:un glide:[s-ts]:	
153	6	酬	ziu	1	ts ^h au	1b	t-split:asp:devoi:diph:io-flip:un glide:[s-ts]:	
153	7	丑	ts ^h iu	2	ts ^h au	2a	t-split:retrof:deret:diph:io-flip:un glide:	
153	8	臭	ts ^h iu	3	ts ^h au	3a	t-split:retrof:deret:diph:io-flip:un glide:	
153	9	收	siu	1	sau	1a	t-split:retrof:deret:diph:io-flip:un glide:	
154	1	熟	ziuk	4	suk	4b	t-split:devoi:retrof:deret:un glide:	
154	2	手	siu	2	sau	2a	t-split:retrof:deret:diph:io-flip:un glide:	
154	3	守	siu	2	sau	2a	t-split:retrof:deret:diph:io-flip:un glide:	
154	3	首	siu	2	sau	2a	t-split:retrof:deret:diph:io-flip:un glide:	
154	4	瘦	giu	3	sau	3a	t-split:retrof:deret:diph:io-flip:un glide:	
154	5	兽	siu	3	sau	3a	t-split:retrof:deret:diph:io-flip:un glide:	
154	6	受	ziu	2	sau	3b	t-split:t2-merge:devoi:retrof:deret:diph:io-flip:un glide:	
154	7	寿	ziu	3	sau	3b	t-split:devoi:retrof:deret:diph:io-flip:un glide:	
154	8	授	ziu	3	sau	3b	t-split:devoi:retrof:deret:diph:io-flip:un glide:	
154	9	售	ziu	3	sau	3b	t-split:devoi:retrof:deret:diph:io-flip:un glide:	
155	1	揉	niu	1	jau	1b	t-split:diph:io-flip:pro-d:glide:	
155	1	柔	niu	1	jau	1b	t-split:diph:io-flip:pro-d:glide:	
155	2	肉	niuuk	4	juk	4b	t-split:pro-d:glide:	
155	3	钩	ku	1	kau	1a	t-split:diph:io-flip:	
155	3	勾	ku	1	kau	1a	t-split:diph:io-flip:	
155	4	沟	ku	1	kau	1a	t-split:diph:io-flip:	
155	5	狗	ku	2	kau	2a	t-split:diph:io-flip:	
155	6	苟	ku	2	kau	2a	t-split:diph:io-flip:	
155	7	够	ku	3	kau	3a	t-split:diph:io-flip:	
155	7	媾	ku	3	kau	3a	t-split:diph:io-flip:	
155	7	毅	ku	3	kau	3a	t-split:diph:io-flip:	
155	7	构	ku	3	kau	3a	t-split:diph:io-flip:	
155	7	购	ku	3	kau	3a	t-split:diph:io-flip:	
155	8	口	k ^h u	2	hau	2a	t-split:spirant:x-weak:diph:io-flip:	
155	9	叩	k ^h u	3	k ^h au	3a	t-split:diph:io-flip:	
156	1	扣	k ^h u	3	k ^h au	3a	t-split:diph:io-flip:	
156	2	寇	k ^h u	3	k ^h au	3a	t-split:diph:io-flip:	
156	3	侯	yu	1	hau	1b	t-split:devoi:x-weak:diph:io-flip:	
156	4	喉	yu	1	hau	1b	t-split:devoi:x-weak:diph:io-flip:	
156	5	猴	yu	1	hau	1b	t-split:devoi:x-weak:diph:io-flip:	
156	6	厚	yu	2	hau	2b	t-split:devoi:x-weak:diph:io-flip:	
156	7	后	yu	3	hau	3b	t-split:devoi:x-weak:diph:io-flip:	
156	8	候	yu	3	hau	3b	t-split:devoi:x-weak:diph:io-flip:	
156	9	欧	au	1	au	1a	t-split:pro-a:diph:io-flip:	
157	1	藕	gu	2	gau	2b	t-split:diph:io-flip:	
157	2	偶	gu	2	gau	2b	t-split:diph:io-flip:	
157	3	呕	ʔu	2	au	2a	t-split:pro-a:diph:io-flip:	
157	4	沔	ʔu	3	au	3a	t-split:pro-a:diph:io-flip:	
157	5	丢	tiu	1	tiu	1a	t-split:	
157	6	牛	giu	1	gau	1b	t-split:diph:io-flip:un glide:	
157	7	扭	niu	2	nau	2a	[t-split]:diph:io-flip:un glide:	
157	7	纽	niu	2	nau	2a	[t-split]:diph:io-flip:un glide:	
157	8	溜	liu	3	lau	3b	t-split:diph:io-flip:un glide:	
157	9	流	liu	1	lau	1b	t-split:diph:io-flip:un glide:	
158	1	琉	liu	1	lau	1b	t-split:diph:io-flip:un glide:	
158	1	琉	liu	1	lau	1b	t-split:diph:io-flip:un glide:	
158	2	刘	liu	1	lau	1b	t-split:diph:io-flip:un glide:	
158	3	留	liu	1	lau	1b	t-split:diph:io-flip:un glide:	
158	4	榴	liu	1	lau	1a	[t-split]:diph:io-flip:un glide:	
158	5	柳	liu	2	lau	2b	t-split:diph:io-flip:un glide:	
158	6	六	liuk	4	luk	4b	t-split:un glide:	
158	7	掀	tsiu	1	tsau	1a	t-split:diph:io-flip:un glide:	
158	8	究	kiu	3	kau	3a	t-split:diph:io-flip:un glide:	
158	9	纠	kiu	2	kau	2a	t-split:diph:io-flip:un glide:	
159	1	酒	tsiu	2	tsau	2a	t-split:diph:io-flip:un glide:	
159	2	九	kiu	2	kau	2a	t-split:diph:io-flip:un glide:	
159	3	久	kiu	2	kau	2a	t-split:diph:io-flip:un glide:	
159	4	灸	kiu	2	kau	3a	t-split:diph:io-flip:un glide:	
159	5	就	dziu	3	tsau	3b	t-split:devoi:diph:io-flip:un glide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
159	6	救	kiu	3	kau	3a	t-split:diph:io-flip:unlide:	
159	7	舅	giu	2	k ^h au	2b	t-split:asp:devoi:diph:io-flip:unlide:	
159	8	旧	giu	3	kau	3b	t-split:devoi:diph:io-flip:unlide:	
159	9	秋	ts ^h iu	1	ts ^h au	1a	t-split:diph:io-flip:unlide:	
160	1	丘	k ^h iu	1	jau	1a	t-split:spirant:diph:io-flip:[pro-x]:glide:	
160	2	囚	ziu	1	ts ^h au	1b	t-split:affric:asp:devoi:diph:io-flip:unlide:	
160	3	求	giu	1	k ^h au	1b	t-split:asp:devoi:diph:io-flip:unlide:	
160	4	球	giu	1	k ^h au	1b	t-split:asp:devoi:diph:io-flip:unlide:	
160	5	修	siu	1	sau	1a	t-split:diph:io-flip:unlide:	
160	5	羞	siu	1	sau	1a	t-split:diph:io-flip:unlide:	
160	6	休	xiu	1	jau	1a	t-split:diph:io-flip:[pro-x]:glide:	
160	7	朽	xiu	2	jau	2a	t-split:diph:io-flip:[pro-x]:glide:	
160	8	秀	siu	3	sau	3a	t-split:diph:io-flip:unlide:	
160	8	绣	siu	3	sau	3a	t-split:diph:io-flip:unlide:	
160	8	锈	siu	3	sau	3a	t-split:diph:io-flip:unlide:	
160	9	袖	ziu	3	tsau	3b	t-split:affric:devoi:diph:io-flip:unlide:	
161	1	优	?iu	1	jau	1a	t-split:diph:io-flip:pro-a:glide:	
161	1	忧	?iu	1	jau	1a	t-split:diph:io-flip:pro-a:glide:	
161	2	悠	iu	1	jau	1b	t-split:glide:diph:io-flip:	
161	3	幽	?iu	1	jau	1a	t-split:diph:io-flip:pro-a:glide:	
161	4	尤	jiu	1	jau	1b	t-split:diph:io-flip:unlide:	
161	5	邮	jiu	1	jau	1b	t-split:diph:io-flip:unlide:	
161	6	油	iu	1	jau	1b	t-split:glide:diph:io-flip:	
161	6	游	iu	1	jau	1b	t-split:glide:diph:io-flip:	
161	6	犹	iu	1	jau	1b	t-split:glide:diph:io-flip:	
161	6	由	iu	1	jau	2b	[t1-t2]:t-split:glide:diph:io-flip:	
161	6	游	iu	1	jau	1b	t-split:glide:diph:io-flip:	
161	7	有	jiu	2	jau	2b	t-split:diph:io-flip:unlide:	
161	8	友	jiu	2	jau	2b	t-split:diph:io-flip:unlide:	
161	9	又	jiu	3	jau	3b	t-split:diph:io-flip:unlide:	
162	1	右	jiu	3	jau	3b	t-split:diph:io-flip:unlide:	
162	2	佑	jiu	3	jau	3b	t-split:diph:io-flip:unlide:	
162	3	袖	iu	3	jau	3b	t-split:glide:diph:io-flip:	
162	3	釉	iu	3	jau	3b	t-split:glide:diph:io-flip:	
162	4	幼	?iu	3	jau	3a	t-split:diph:io-flip:pro-a:glide:	
162	5	斑	pan	1	pan	1a	t-split:	
162	5	班	pan	1	pan	1a	t-split:	
162	5	颁	pan	1	pan	1a	t-split:	
162	6	扳	pan	1	p ^h an	1a	t-split:length:[asp]:	
162	7	般	puan	1	pun	1a	t-split:lab-dism:rnd:	rnd-adj:
162	8	搬	puan	1	pun	1a	t-split:lab-dism:rnd:	rnd-adj:
162	9	版	pan	2	pa:n	2a	t-split:length:	
162	9	版	pan	2	pa:n	2a	t-split:length:	
163	1	扮	pan	3	pa:n	3a	t-split:length:	
163	2	办	ban	3	pa:n	3b	t-split:devoi:length:	
163	3	半	puan	3	pun	3a	t-split:lab-dism:rnd:	rnd-adj:
163	4	伴	buan	2	pun	3b	t-split:devoi:lab-dism:rnd:	rnd-adj:
163	5	拌	buan	2	p ^h un	3a	[t2-t3]:[t-split]:asp:devoi:lab-dism:rnd:	rnd-adj:
163	6	攀	p ^h an	1	p ^h an	1a	t-split:length:	
163	7	潘	p ^h uan	1	p ^h un	1a	t-split:lab-dism:rnd:	rnd-adj:
163	8	盘	buan	1	p ^h un	1b	t-split:asp:devoi:lab-dism:rnd:	rnd-adj:
163	9	盼	p ^h an	3	p ^h an	3a	t-split:length:	
164	1	判	p ^h uan	3	p ^h un	3a	t-split:lab-dism:rnd:	rnd-adj:
164	2	奎	man	1	ma:n	1b	t-split:length:	
164	3	瞞	muan	1	mun	1b	t-split:lab-dism:rnd:	rnd-adj:
164	4	满	muan	2	mun	2b	t-split:lab-dism:rnd:	rnd-adj:
164	5	慢	man	3	ma:n	3b	t-split:length:	
164	6	漫	muan	3	ma:n	3b	t-split:length:	
164	7	幔	muan	3	ma:n	3b	t-split:length:	
164	8	帆	vyam	1	fa:n	1b	t-split:devoi:lab-dnt':lab-dism:length:final-dism:	
164	9	番	f ^h yan	1	fan	1a	t-split:[asp-r]:lab-dnt':lab-dism:	
164	9	翻	f ^h yan	1	fan	1a	t-split:[asp-r]:lab-dnt':lab-dism:	
165	1	凡	vyam	1	fa:n	1b	t-split:devoi:lab-dnt':lab-dism:length:final-dism:	
165	2	烦	vyan	1	fa:n	1b	t-split:devoi:lab-dnt':lab-dism:length:	
165	2	矾	vyan	1	fa:n	1b	t-split:devoi:lab-dnt':lab-dism:length:	
165	2	繁	vyan	1	fa:n	1b	t-split:devoi:lab-dnt':lab-dism:length:	
165	3	反	fyam	2	fa:n	2a	t-split:lab-dnt':lab-dism:length:	
165	4	犯	vyam	2	fa:n	3a	[t2-t3]:[t-split]:devoi:lab-dnt':lab-dism:length:final-dism:	
165	4	范	vyam	2	fa:n	3b	t-split:t2-merge:devoi:lab-dnt':lab-dism:length:final-dism:	
165	5	贩	fyam	3	fa:n	3a	t-split:lab-dnt':lab-dism:length:	
165	6	饭	vyam	3	fa:n	3b	t-split:devoi:lab-dnt':lab-dism:length:	
165	7	担	tam	1	ta:m	1a	t-split:length:	
165	8	丹	tan	1	ta:n	1a	t-split:length:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
165	9	单	tan	1	ta:n	1a	t-split:length:	
166	1	胆	tam	2	ta:m	2a	t-split:length:	
166	2	担	tam	3	ta:m	3a	t-split:length:	
166	3		dam	3	ta:m	3b	t-split:devoi:length:	
166	4	旦	tan	3	ta:n	3a	t-split:length:	
166	5	但	dan	3	ta:n	3b	t-split:devoi:length:	
166	5	蛋	dan	3	ta:n	3b	t-split:devoi:length:	
166	6	弹	dan	3	t ^h a:n	3b	t-split:asp:devoi:length:	
166	7	贪	t ^h am	1	t ^h a:m	1a	t-split:length:	
166	8	坍	t ^h am	1	t ^h a:n	1a	t-split:length:	
166	9	摊	t ^h an	1	t ^h a:n	1a	t-split:length:	
166	9	滩	t ^h an	1	t ^h a:n	1a	t-split:length:	
167	1	潭	dam	1	t ^h a:m	1b	t-split:asp:devoi:length:	
167	2	谈	dam	1	t ^h a:m	1b	t-split:asp:devoi:length:	
167	3	痰	dam	1	t ^h a:m	1b	t-split:asp:devoi:length:	
167	4	檀	dan	1	t ^h a:n	1b	t-split:asp:devoi:length:	
167	5	弹	dan	1	t ^h a:n	1b	t-split:asp:devoi:length:	
167	6	毯	t ^h am	2	t ^h a:m	2a	t-split:length:	
167	7	坦	t ^h an	2	t ^h a:n	2a	t-split:length:	
167	8	探	t ^h am	3	t ^h a:m	3a	t-split:length:	
167	9	炭	t ^h an	3	t ^h a:n	3a	t-split:length:	
168	1	叹	t ^h an	3	t ^h a:n	3a	t-split:length:	
168	2	男	nam	1	na:m	1b	t-split:length:	
168	3	南	nam	1	na:m	1b	t-split:length:	
168	4	难	nan	1	na:n	1b	t-split:length:	
168	5	难	nan	3	na:n	3b	t-split:length:	
168	6	蓝	lam	1	la:m	1b	t-split:length:	
168	7	篮	lam	1	la:m	1b	t-split:length:	
168	8	栏	lan	1	lan	1a	[t-split];[back,a-fronted]:	
168	8	兰	lan	1	lan	1b	t-split:[back,a-fronted]:	
168	9	拦	lan	1	la:n	1b	t-split:length:	
169	1	览	lam	2	la:m	2b	t-split:length:	
169	2	揽	lam	2	la:m	2b	t-split:length:	
169	3	懒	lan	2	la:n	2b	t-split:length:	
169	4	滥	lam	3	la:m	3b	t-split:length:	
169	5	烂	lan	3	la:n	3b	t-split:length:	
169	6	暂	dzam	3	t ^s a:m	3b	t-split:devoi:length:	
169	7	赞	tsan	3	t ^s a:n	3a	t-split:length:	
169	8	参	t ^s h ^h am	1	t ^s h ^h a:m	1a	t-split:length:	
169	9	餐	t ^s h ^h an	1	t ^s h ^h a:n	1a	t-split:length:	
170	1	蚕	dzam	1	t ^s h ^h a:m	1b	t-split:asp:devoi:length:	
170	2	残	dzan	1	t ^s h ^h a:n	1b	t-split:asp:devoi:length:	
170	3	惨	t ^s h ^h am	2	t ^s h ^h a:m	2a	t-split:length:	
170	4	伞	sam	1	sa:m	1a	t-split:length:	
170	5	散	san	2	san	3a	[t2-t3]:t-split:[back,a-fronted]:	
170	5	散	san	2	san	3a	[t2-t3]:t-split:[back,a-fronted]:	
170	6	散	sam	3	sa:n	3a	t-split:length:	
170	7	沾	ŋiam	1	tsim	1a	t-split:fricat:retrof:deret:io-flip:rime:	
170	8	粘	ŋiam	1	tsim	1a	t-split:fricat:retrof:deret:io-flip:rime:	
170	9	毡	t ^s ian	1	tsin	1a	t-split:fricat:retrof:deret:io-flip:rime:	
171	1	斩	t ^s am	2	t ^s a:m	2a	t-split:retrof:deret:length:	
171	2	盏	t ^s an	2	t ^s a:n	2a	t-split:retrof:deret:length:	
171	3	展	ŋian	2	tsin	2a	t-split:fricat:retrof:deret:io-flip:rime:	
171	4	站	ŋam	3	t ^s a:m	3b	[t-split];fricat:retrof:deret:length:	
171	5	蘸	t ^s am	3	t ^s a:m	3a	t-split:deret:length:	
171	6	占	t ^s iam	3	tsim	3a	t-split:retrof:deret:io-flip:rime:	
171	7	栈	dz ^h an	3	t ^s a:n	2a	[t3-t2]:[t-split];devoi:deret:length:	
171	8	战	t ^s ian	3	tsin	3a	t-split:retrof:deret:io-flip:rime:	
171	9	撵	t ^s h ^h am	1	t ^s h ^h a:m	1a	t-split:deret:length:	
172	1	撵	dz ^h am	1	t ^s h ^h a:m	1b	t-split:asp:devoi:deret:length:	
172	2	撵	dian	1	t ^s h ^h in	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:	
172	3	蝉	ŋian	1	sim	1b	t-split:devoi:retrof:deret:io-flip:rime:[n-m]:	
172	4	铲	t ^s h ^h an	2	t ^s h ^h a:n	2a	t-split:deret:length:	
172	5	产	t ^s h ^h an	2	t ^s h ^h a:n	2a	t-split:deret:length:	
172	6	杉	gam	1	sa:m	1a	t-split:deret:length:	
172	7	山	gan	1	sa:n	1a	t-split:deret:length:	
172	8	扇	ŋian	1	sin	3a	[t1-t3]:t-split:retrof:deret:io-flip:rime:	
172	9	陕	ŋiam	2	sim	2a	t-split:retrof:deret:io-flip:rime:	
173	1	扇	ŋian	3	sin	3a	t-split:retrof:deret:io-flip:rime:	
173	2	善	ŋian	2	sin	3b	t-split:devoi:retrof:deret:io-flip:rime:	
173	3	然	ŋian	1	jin	1b	t-split:io-flip:rime:pro-d:glide-ins:	
173	4	燃	ŋian	1	jin	1b	t-split:io-flip:rime:pro-d:glide-ins:	
173	5	染	ŋiam	2	jin	2b	t-split:io-flip:rime:pro-d:glide-ins:[m-n]:	
173	6	甘	kam	1	ka:m	1a	t-split:length:	
173	7	柑	kam	1	ka:m	1a	t-split:length:	
173	8	干	kan	1	kɔ:n	1a	t-split:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
173	9	干	kan	1	kɔn	1a	t-split:rnd:	
173	9	竿	kan	1	kɔn	1a	t-split:rnd:	
173	9	肝	kan	1	kɔn	1a	t-split:rnd:	
174	1	感	kam	2	kaɪm	2a	t-split:length:	
174	2	敢	kam	2	kaɪm	2a	t-split:length:	
174	3	杆	kan	2	kɔn	2a	t-split:rnd:	
174	4	杆	kan	2	kɔn	2a	t-split:rnd:	
174	5	赶	kan	2	kɔn	2a	t-split:rnd:	
174	6	干	kan	3	kɔn	3a	t-split:rnd:	
174	7	堪	k ^h am	1	haɪm	1a	t-split:spirant:x-weak:length:	
174	8	刊	k ^h an	1	hɔn	1a	t-split:spirant:x-weak:rnd:	
174	9	坎	k ^h am	2	haɪm	2a	t-split:spirant:x-weak:length:	
175	1	砍	k ^h am	2	haɪm	2a	t-split:spirant:x-weak:length:	
175	2	看	k ^h an	3	hɔn	3a	t-split:spirant:x-weak:rnd:	
175	3	含	ɣam	1	haɪm	1a	[t-split]:devoi:x-weak:length:	
175	4	函	ɣam	1	haɪm	1a	[t-split]:devoi:x-weak:length:	
175	5	韩	ɣan	1	hɔn	1a	[t-split]:devoi:x-weak:rnd:	
175	6	韩	ɣan	1	hɔn	1a	[t-split]:devoi:x-weak:rnd:	
175	7	喊	xam	2	haɪm	3a	t-split:x-weak:length:	
175	8	汉	xan	3	hɔn	3a	t-split:x-weak:rnd:	
175	9	旱	ɣan	2	hɔn	2b	t-split:devoi:x-weak:rnd:	
176	1	汗	ɣan	3	hɔn	3a	[t-split]:devoi:x-weak:rnd:	
176	2	安	?an	1	ɔn	1a	t-split:rnd:pro-a:	
176	3	鞍	?an	1	ɔn	1a	t-split:rnd:pro-a:	
176	4	暗	?am	3	aɪm	3a	t-split:length:pro-a:	
176	5	岸	ɣan	3	ɣɔn	3b	t-split:rnd:	
176	6	按	?an	3	ɔn	3a	t-split:rnd:pro-a:	
176	7	案	?an	3	ɔn	3a	t-split:rnd:pro-a:	
176	8	鞭	pian	1	pin	1a	t-split:io-flip:rime:	
176	9	编	pian	1	p ^h in	1a	t-split:io-flip:rime:[asp]:	
177	1	边	pian	1	pin	1a	t-split:io-flip:rime:	
177	2	贬	pian	2	pin	2a	t-split:io-flip:rime:final-dism:	
177	3	匾	pian	2	pin	2a	t-split:io-flip:rime:	
177	3	扁	pian	2	pin	2a	t-split:io-flip:rime:	
177	4	变	pian	3	pin	3a	t-split:io-flip:rime:	
177	5	辨	bian	2	pin	3b	t-split:t2-merge:devoi:io-flip:rime:	
177	5	辨	bian	2	pin	3b	t-split:t2-merge:devoi:io-flip:rime:	
177	6	便	bian	3	pin	3b	t-split:devoi:io-flip:rime:	
177	7	辨	bian	2	pin	1a	[t2-t1]:[t-split]:devoi:io-flip:rime:	
177	8	遍	pian	3	pin	3a	t-split:io-flip:rime:	
177	9	遍	pian	3	pin	3a	t-split:io-flip:rime:	
178	1	偏	p ^h ian	1	p ^h in	1a	t-split:io-flip:rime:	
178	1	篇	p ^h ian	1	p ^h in	1a	t-split:io-flip:rime:	
178	2	便	bian	1	p ^h in	1b	t-split:asp:devoi:io-flip:rime:	
178	3	骗	p ^h ian	3	p ^h in	3a	t-split:io-flip:rime:	
178	4	片	p ^h ian	3	p ^h in	3a	t-split:io-flip:rime:	
178	5	棉	mian	1	min	1b	t-split:io-flip:rime:	
178	5	绵	mian	1	min	1b	t-split:io-flip:rime:	
178	6	眠	mian	1	min	1b	t-split:io-flip:rime:	
178	7	免	mian	2	min	2b	t-split:io-flip:rime:	
178	7	勉	mian	2	min	2b	t-split:io-flip:rime:	
178	8	面	mian	3	min	3b	t-split:io-flip:rime:	
178	9	颠	tian	1	tin	1a	t-split:io-flip:rime:	
179	1	点	tiam	2	tim	2a	t-split:io-flip:rime:	
179	2	典	tian	2	tin	2a	t-split:io-flip:rime:	
179	3	店	tiam	3	tim	3a	t-split:io-flip:rime:	
179	4	电	dian	3	tin	3b	t-split:devoi:io-flip:rime:	
179	5	殿	dian	3	tin	3b	t-split:devoi:io-flip:rime:	
179	6	佃	dian	3	tin	3b	t-split:devoi:io-flip:rime:	
179	7	添	t ^h iam	1	t ^h im	1a	t-split:io-flip:rime:	
179	8	天	t ^h ian	1	t ^h in	1a	t-split:io-flip:rime:	
179	9	甜	diam	1	t ^h im	1b	t-split:asp:devoi:io-flip:rime:	
180	1	田	dian	1	t ^h in	1b	t-split:asp:devoi:io-flip:rime:	
180	2	填	dian	1	t ^h in	1b	t-split:asp:devoi:io-flip:rime:	
180	3	舔	t ^h iam	2	t ^h im	2a	t-split:io-flip:rime:	
180	4	黏	niam	1	nim	1b	t-split:io-flip:rime:	
180	5	拈	niam	1	nim	1a	[t-split]:io-flip:rime:	
180	6	年	nian	1	nin	1b	t-split:io-flip:rime:	
180	7	碾	nian	2	nin	2b	t-split:io-flip:rime:	
180	8	辇	lian	2	lin	2b	t-split:io-flip:rime:	
180	9	撵	nian	2	lin	2b	t-split:io-flip:rime:[n-l]:	
181	1	念	niam	3	nim	3b	t-split:io-flip:rime:	
181	2	帘	liam	1	lim	1b	t-split:io-flip:rime:	
181	2	廉	liam	1	lim	1b	t-split:io-flip:rime:	
181	2	帘	liam	1	lim	1b	t-split:io-flip:rime:	
181	2	帘	liam	1	lim	1a	[t-split]:io-flip:rime:	
181	3	连	lian	1	lin	1b	t-split:io-flip:rime:	

pg	co	ch	sme	tn	mc	tn	Diachronic	Allophonic
181	4	联	lian	1	lyn	1b	t-split:unglide:[rnd]:	rnd-adj:acute:
181	5	怜	lian	1	lin	1b	t-split:io-flip:rime:	
181	6	莲	lian	1	lin	1b	t-split:io-flip:rime:	
181	7	脸	kiam	2	lim	2b	[t-split]:io-flip:rime:[k-l]:	
181	8	敛	liam	3	lim	3b	t-split:io-flip:rime:	
181	9	炼	lian	3	lin	3b	t-split:io-flip:rime:	
181	9	练	lian	3	lin	3b	t-split:io-flip:rime:	
181	9	炼	lian	3	lin	3b	t-split:io-flip:rime:	
182	1	恋	luan	3	lyn	2b	[t3-t2]:t-split:rnd:	rnd-adj:degem:acute:
182	2	监	kiam	1	ka:m	1a	t-split:[unglide]:length:	
182	3	尖	tsiam	1	tsim	1a	t-split:io-flip:rime:	
182	4	歼	tsiam	1	ts ^h im	1a	t-split:io-flip:rime:[asp]:	
182	5	兼	kiam	1	kim	1a	t-split:io-flip:rime:	
182	6	间	kan	1	ka:n	1a	t-split:length:	
182	7	艰	kan	1	ka:n	1a	t-split:length:	
182	8	奸	kan	1	kan	1a	t-split:	
182	8	奸	kan	1	kan	1a	t-split:	
182	9	煎	tsian	1	tsin	1a	t-split:io-flip:rime:	
183	1	筵	tsian	1	tsin	1a	t-split:io-flip:rime:	
183	2	肩	kian	1	kin	1a	t-split:io-flip:rime:	
183	3	坚	kian	1	kin	1a	t-split:io-flip:rime:	
183	4	减	kam	2	ka:m	2a	t-split:length:	
183	5	碱	kam	2	ka:m	2a	t-split:length:	
183	6	检	kiam	2	kim	2a	t-split:io-flip:rime:	
183	7	俭	giam	2	kim	3b	t-split:devoi:io-flip:rime:	
183	8	柬	kan	2	kan	2a	t-split:	
183	8	简	kan	2	kan	2a	t-split:	
183	9	捡	kan	2	ka:n	2a	t-split:length:	
184	1	剪	tsian	2	tsin	2a	t-split:io-flip:rime:	
184	2	茧	kian	2	ka:n	2a	t-split:[unglide]:length:	
184	3	渐	dziam	2	tsim	3a	[t-split]:devoi:io-flip:rime:	
184	4	剑	kiam	3	kim	3a	t-split:io-flip:rime:	
184	5	间	kan	3	ka:n	3a	t-split:length:	
184	6	河	kan	3	ka:n	3a	t-split:length:	
184	7	箭	tsian	3	tsin	3a	t-split:io-flip:rime:	
184	8	溅	tsian	3	tsin	3a	t-split:io-flip:rime:	
184	9	贱	dzian	2	tsin	3b	t-split:t2-merge:devoi:io-flip:rime:	
185	1	件	gian	2	kin	3b	t-split:t2-merge:devoi:io-flip:rime:	
185	2	建	kian	3	kin	3a	t-split:io-flip:rime:	
185	3	健	gian	3	kin	3b	t-split:devoi:io-flip:rime:	
185	4	腱	gian	3	kin	3a	[t-split]:devoi:io-flip:rime:	
185	5	荐	tsian	3	tsin	3a	t-split:io-flip:rime:	
185	6	见	kian	3	kin	3a	t-split:io-flip:rime:	
185	7	签	ts ^h iam	1	ts ^h im	1a	t-split:io-flip:rime:	
185	7	签	ts ^h iam	1	ts ^h im	1a	t-split:io-flip:rime:	
185	8	谦	k ^h iam	1	him	1a	t-split:spirant:x-weak:io-flip:rime:	
185	9	迁	ts ^h ian	1	ts ^h in	1a	t-split:io-flip:rime:	
186	1	千	ts ^h ian	1	ts ^h in	1a	t-split:io-flip:rime:	
186	2	辛	k ^h ian	1	hin	1a	t-split:spirant:x-weak:io-flip:rime:	
186	3	钳	yan	1	jyn	1b	t-split:glide-ins:io-flip:rime:	
186	4	钳	giam	1	k ^h im	1b	t-split:asp:devoi:io-flip:rime:	
186	5	钱	dzian	1	ts ^h in	1b	t-split:asp:devoi:io-flip:rime:	
186	6	前	dzian	1	ts ^h in	1b	t-split:asp:devoi:io-flip:rime:	
186	7	乾	gian	1	k ^h in	1b	t-split:asp:devoi:io-flip:rime:	
186	8	虔	gian	1	k ^h in	1b	t-split:asp:devoi:io-flip:rime:	
186	9	浅	ts ^h ian	2	ts ^h in	2a	t-split:io-flip:rime:	
187	1	欠	k ^h iam	3	him	3a	t-split:spirant:x-weak:io-flip:rime:	
187	2	仙	sian	1	sin	1a	t-split:io-flip:rime:	
187	3	鲜	sian	1	sin	1a	t-split:io-flip:rime:	
187	4	掀	xian	1	hin	1a	t-split:x-weak:io-flip:rime:	
187	5	先	sian	1	sin	1a	t-split:io-flip:rime:	
187	6	咸	yam	1	ha:m	1b	t-split:devoi:x-weak:length:	
187	7	衔	yam	1	ha:m	1b	t-split:devoi:x-weak:length:	
187	8	嫌	yiam	1	jim	1b	t-split:io-flip:rime:pro-b:glide-ins:	
187	9	闲	yan	1	ha:n	1b	t-split:devoi:x-weak:length:	
187	9	闲	yan	1	ha:n	1b	t-split:devoi:x-weak:length:	
188	1	贤	yian	1	jin	1b	t-split:io-flip:rime:pro-b:glide-ins:	
188	2	险	xiam	2	him	2a	t-split:x-weak:io-flip:rime:	
188	3	显	xian	2	hin	2a	t-split:x-weak:io-flip:rime:	
188	4	陷	yam	3	ha:m	3b	t-split:devoi:x-weak:length:	
188	5	馅	yam	3	ha:m	3b	t-split:devoi:x-weak:length:	
188	6	限	yan	2	ha:n	3b	t-split:devoi:x-weak:length:	
188	7	线	sian	3	sin	3a	t-split:io-flip:rime:	
188	8	宪	xian	3	hin	3a	t-split:x-weak:io-flip:rime:	
188	9	献	xian	3	hin	3a	t-split:x-weak:io-flip:rime:	
189	1	现	yian	3	jim	3b	t-split:io-flip:rime:pro-b:glide-ins:	
189	2	县	xyan	3	yn	3b	t-split:io-flip:rime:pro-b:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
189	3	淹	?iam	1	jim	1a	t-split:io-flip:rime:pro-a:glide-ins:	
189	4	腌	?iam	1	jim	1a	t-split:io-flip:rime:pro-a:glide-ins:	
189	5	烟	?iam	1	jin	1a	t-split:io-flip:rime:pro-a:glide-ins:	
189	6	岩	gam	1	ga:m	1b	t-split:length:	
189	7	盐	iam	1	jim	1b	t-split:glide-ins:io-flip:rime:	
189	8	檐	iam	1	jim	1b	t-split:glide-ins:io-flip:rime:	
189	9	间	iam	1	jim	1b	t-split:glide-ins:io-flip:rime:	
190	1	严	giam	1	jim	1b	t-split:io-flip:rime:pro-c:glide-ins:	
190	2	颜	gan	1	ga:n	1b	t-split:length:	
190	3	延	ian	1	jin	1b	t-split:glide-ins:io-flip:rime:	
190	4	言	gian	1	jin	1b	t-split:io-flip:rime:pro-c:glide-ins:	
190	5	研	gian	1	jin	1b	t-split:io-flip:rime:pro-c:glide-ins:	
190	6	掩	?iam	2	jim	2a	t-split:io-flip:rime:pro-a:glide-ins:	
190	7	眼	gan	2	ga:tn	2b	t-split:length:	
190	8	演	ian	2	jin	2a	t-split:glide-ins:io-flip:rime:	
190	9	验	giam	3	jim	3b	t-split:io-flip:rime:pro-c:glide-ins:	
191	1	厌	?iam	3	jim	3a	t-split:io-flip:rime:pro-a:glide-ins:	
191	2	雁	gan	3	ga:n	3b	t-split:length:	
191	3	晏	?an	3	a:n	3a	t-split:length:pro-a:	
191	4	砚	gian	3	jin	3b	t-split:io-flip:rime:pro-c:glide-ins:	
191	5	燕	?ian	3	jin	3a	t-split:io-flip:rime:pro-a:glide-ins:	
191	6	咽	?ian	3	jin	3a	t-split:io-flip:rime:pro-a:glide-ins:	
191	7	宴	?ian	3	jin	3a	t-split:io-flip:rime:pro-a:glide-ins:	
191	8	端	tuán	1	tyn	1a	t-split:rnd:	rnd-adj:degem:acute:
191	9	短	tuán	2	tyn	1a	[t2-t1]:t-split:rnd:	rnd-adj:degem:acute:
192	1	断	tuán	3	tyn	3a	t-split:rnd:	rnd-adj:degem:acute:
192	2	锻	tuán	3	tyn	3a	t-split:rnd:	rnd-adj:degem:acute:
192	3	断	duán	2	tyn	3b	t-split:t2-merge:devoi:rnd:	rnd-adj:degem:acute:
192	4	段	duán	3	tyn	3b	t-split:devoi:rnd:	rnd-adj:degem:acute:
192	5	锻	duán	3	tyn	3b	t-split:devoi:rnd:	rnd-adj:degem:acute:
192	6	团	duán	1	thyn	1b	t-split:asp:devoi:rnd:	rnd-adj:degem:acute:
192	7	暖	nuán	2	nyn	2b	t-split:rnd:	rnd-adj:degem:acute:
192	8	卵	luán	2	læn	2b	t-split:rnd:unglide:	acute:
192	9	乱	luán	3	lyn	3a	[t-split]:rnd:	rnd-adj:degem:acute:
193	1	钻	tsuan	1	tsyn	1a	t-split:rnd:	rnd-adj:degem:acute:
193	2	钻	tsuan	3	tsyn	3a	t-split:rnd:	rnd-adj:degem:acute:
193	3	酸	suan	1	syn	1a	t-split:rnd:	rnd-adj:degem:acute:
193	4	算	suan	3	syn	3a	t-split:rnd:	rnd-adj:degem:acute:
193	4	蒜	suan	3	syn	3a	t-split:rnd:	rnd-adj:degem:acute:
193	5	专	tśyan	1	tsyn	1a	t-split:retrof:deret:io-flip:rime:	
193	5	砖	tśyan	1	tsyn	1a	t-split:retrof:deret:io-flip:rime:	
193	6	转	fyán	2	tsyn	3a	[t2-t3]:t-split:fricat:retrof:deret:io-flip:rime:	
193	7	赚	đám	3	tsa:n	3b	t-split:devoi:fricat:retrof:deret:length:[m-n]:	acute:
193	8	转	fuán	3	tsyn	2a	[t3-t2]:t-split:fricat:retrof:deret:rnd:	rnd-adj:degem:acute:
193	9	篆	đyan	2	syn	3b	t-split:t2-merge:devoi:fricat:retrof:deret:io-flip:rime:[ts-s]:	
194	1	传	đyan	3	tsyn	3b	t-split:devoi:fricat:retrof:deret:io-flip:rime:	
194	2	川	ts ^h yan	1	ts ^h yn	1a	t-split:retrof:deret:io-flip:rime:	
194	3	穿	ts ^h yan	1	ts ^h yn	1a	t-split:retrof:deret:io-flip:rime:	
194	4	传	đyan	1	ts ^h yn	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:	
194	5	椽	đyan	1	ts ^h yn	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:	
194	6	船	dzyán	1	syn	1b	t-split:devoi:retrof:deret:io-flip:rime:[ts-s]:	
194	7	喘	ts ^h yan	2	ts ^h yn	2a	t-split:retrof:deret:io-flip:rime:	acute:
194	8	串	ts ^h uan	3	ts ^h yn	3a	t-split:retrof:deret:rnd:	rnd-adj:degem:
194	9	串	ts ^h yan	3	ts ^h yn	3a	t-split:retrof:deret:io-flip:rime:	
195	1	门	guan	1	sa:n	1a	t-split:r-unglide:deret:length:	
195	2	挂	gyán	1	sa:n	1a	t-split:r-unglide:deret:length:	
195	3	软	nyán	2	jyn	2b	t-split:io-flip:rime:pro-d:glide-ins:	
195	4	官	kuan	1	kun	1a	t-split:rnd:	rnd-adj:degem:
195	5	观	kuan	1	kun	1a	t-split:rnd:	rnd-adj:degem:
195	6	冠	kuan	1	kun	1a	t-split:rnd:	rnd-adj:degem:
195	7	关	kuan	1	kwatn	1a	t-split:lab-vel:length:	
195	8	管	kuan	2	kun	2a	t-split:rnd:	rnd-adj:degem:
195	8	馆	kuan	2	kun	2a	t-split:rnd:	rnd-adj:degem:
195	9	罐	kuan	3	kun	3a	t-split:rnd:	rnd-adj:degem:
195	9	贯	kuan	3	kun	3a	t-split:rnd:	rnd-adj:degem:
196	1	灌	kuan	3	kun	3a	t-split:rnd:	rnd-adj:degem:
196	2	观	kuán	3	kun	3a	t-split:rnd:	rnd-adj:degem:
196	3	冠	kuan	3	kun	3a	t-split:rnd:	rnd-adj:degem:
196	4	惯	kuan	3	kwatn	3a	t-split:lab-vel:length:	
196	5	宽	k ^h uan	1	fun	1a	t-split:spirant:x-lab:f-coal:rnd:	rnd-adj:degem:
196	6	款	k ^h uan	2	fun	2a	t-split:spirant:x-lab:f-coal:rnd:	rnd-adj:degem:
196	7	欢	xuan	1	fun	1a	t-split:spirant:x-lab:f-coal:rnd:	rnd-adj:degem:
196	8	还	yuan	1	wa:n	1b	t-split:length:pro-b:glide:	
196	9	环	yuan	1	wa:n	1b	t-split:length:pro-b:glide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
197	1	缓	yuan	2	wun	3b	t-split:t2-merge:glide:rnd:pro-b:	rnd-adj:
197	2	唤	xuan	3	wun	3b	[t-split]:rnd:[pro-x]:glide:	rnd-adj:
197	2	换	xuan	3	wun	3b	[t-split]:rnd:[pro-x]:glide:	rnd-adj:
197	3	换	yuan	3	wun	3b	t-split:rnd:pro-b:glide:	rnd-adj:
197	4	患	yuan	3	wa:n	3b	t-split:length:pro-b:glide:	
197	5	宦	yuan	3	wa:n	3b	t-split:length:pro-b:glide:	
197	6	跪	?uan	1	wun	1a	t-split:rnd:pro-b:glide:	rnd-adj:
197	7	穹	?uan	1	wa:n	1a	t-split:length:pro-b:glide:	
197	7	湾	?uan	1	wa:n	1a	t-split:length:pro-b:glide:	
197	8	完	yuan	1	jyn	1b	t-split:rnd:pro-b:[j-ins]:	rnd-adj:degem:acute:
197	9	丸	yuan	1	jyn	1b	t-split:rnd:pro-b:[j-ins]:	rnd-adj:degem:acute:
198	1	玩	ɲuan	1	wa:n	1b	t-split:length:pro-c:glide:	
198	2	顽	ɲuan	1	wa:n	1b	t-split:length:pro-c:glide:	
198	3	碗	?uan	2	wun	2a	t-split:rnd:pro-a:glide:	rnd-adj:
198	4	晚	ɲyan	2	ma:n	2a	[t-split]:lab-dnt':lab-dism:length:	
198	5	烧	ɲyan	2	wa:n	2b	t-split:lab-dnt':lab-dism:length:[m-w]:	
198	6	万	ɲyan	3	ma:n	3b	t-split:lab-dnt':lab-dism:length:	
198	7	蔓	ɲyan	3	ma:n	3b	t-split:lab-dnt':lab-dism:length:	
198	8	捐	yan	1	kyn	1a	[t-split]:io-flip:rime:[k-ins]:	
198	9	卷	kyan	2	kyn	2a	t-split:io-flip:rime:	
199	1	眷	kyan	3	kyn	3a	t-split:io-flip:rime:	
199	2	眷	kyan	3	kyn	3a	t-split:io-flip:rime:	
199	3	绢	kyan	3	kyn	3a	t-split:io-flip:rime:	
199	4	倦	gyan	3	kyn	3b	t-split:devoi:io-flip:rime:	
199	5	圈	gyan	2	kyn	3b	t-split:t2-merge:devoi:io-flip:rime:	
199	6	圈	k ^h yan	1	hyn	1a	t-split:spirant:x-weak:io-flip:rime:	
199	7	泉	dzyan	1	ts ^h yn	1b	t-split:asp:devoi:io-flip:rime:	
199	8	全	dzyan	1	ts ^h yn	1b	t-split:asp:devoi:io-flip:rime:	
199	9	权	gyan	1	k ^h yn	1b	t-split:asp:devoi:io-flip:rime:	
200	1	拳	gyan	1	k ^h yn	1b	t-split:asp:devoi:io-flip:rime:	
200	2	犬	k ^h yan	2	hyn	2a	t-split:spirant:x-weak:io-flip:rime:	
200	3	犬	k ^h yan	3	hyn	3a	t-split:spirant:x-weak:io-flip:rime:	
200	4	旋	syau	1	syn	1a	t-split:io-flip:rime:	
200	5	旋	zuan	1	syn	1b	t-split:devoi:io-flip:rime:	acute:
200	6	玄	xyan	1	jyn	1b	t-split:io-flip:rime:pro-b:glide-ins:	
200	7	悬	xyan	1	jyn	1b	t-split:io-flip:rime:pro-b:glide-ins:	
200	8	癣	sian	2	sin	2a	t-split:io-flip:rime:	
200	9	选	syau	2	syn	2a	t-split:io-flip:rime:	
201	1	揆	xyan	3	hyn	3a	t-split:x-weak:io-flip:rime:	
201	2	冤	?yan	1	jyn	1a	t-split:io-flip:rime:pro-a:glide-ins:	
201	3	员	jyan	1	jyn	1b	t-split:io-flip:rime:	
201	4	圆	jyan	1	jyn	1b	t-split:io-flip:rime:	
201	5	缘	yan	1	jyn	1b	t-split:glide-ins:io-flip:rime:	
201	6	原	ɲyan	1	jyn	1b	t-split:io-flip:rime:pro-c:glide-ins:	
201	6	源	ɲyan	1	jyn	1b	t-split:io-flip:rime:pro-c:glide-ins:	
201	7	元	ɲyan	1	jyn	1b	t-split:io-flip:rime:pro-c:glide-ins:	
201	8	袁	jyan	1	jyn	1b	t-split:io-flip:rime:	
201	8	猿	jyan	1	jyn	1b	t-split:io-flip:rime:	
201	9	园	jyan	1	jyn	1b	t-split:io-flip:rime:	
202	1	媛	jyan	1	jyn	1b	t-split:io-flip:rime:	
202	2	远	jyan	2	jyn	2b	t-split:io-flip:rime:	
202	3	院	jyan	3	jyn	3b	t-split:io-flip:rime:	
202	4	怨	ɲyan	3	jyn	3b	t-split:io-flip:rime:pro-c:glide-ins:	
202	5	怨	?yan	3	jyn	3a	t-split:io-flip:rime:pro-a:glide-ins:	
202	6	奔	puan	1	pan	1a	t-split:lab-dism:io-flip:	
202	7	本	puan	2	pun	2a	t-split:rime:	
202	8	笨	buən	2	pan	3b	t-split:t2-merge:devoi:lab-dism:io-flip:	
202	9	喷	p ^h uən	1	p ^h an	3a	[t1-t3]:t-split:lab-dism:io-flip:	
203	1	盆	buən	1	p ^h un	1b	t-split:asp:devoi:rime:	
203	2	门	muən	1	mun	1b	t-split:rime:	
203	3	闷	muən	3	mun	3a	[t-split]:rime:	
203	4	分	fyən	1	fan	1a	t-split:lab-dnt':lab-dism:io-flip:	
203	5	芬	f ^h yən	1	fan	1a	t-split:asp:lab-dnt':lab-dism:io-flip:	
203	5	芬	f ^h yən	1	fan	1a	t-split:asp:lab-dnt':lab-dism:io-flip:	
203	6	粉	vyən	1	fan	1b	t-split:devoi:lab-dnt':lab-dism:io-flip:	
203	7	粪	fyən	2	fan	2a	t-split:lab-dnt':lab-dism:io-flip:	
203	8	粪	fyən	3	fan	3a	t-split:lab-dnt':lab-dism:io-flip:	
203	9	奋	fyən	3	fan	3a	t-split:lab-dnt':lab-dism:io-flip:	acute:
204	1	愤	vyən	2	fan	2b	t-split:devoi:lab-dnt':lab-dism:io-flip:	
204	2	份	vyən	2	fan	1b	[t2-t1]:t-split:devoi:lab-dnt':lab-dism:io-flip:	
204	3	份	vyən	3	fan	3b	t-split:devoi:lab-dnt':lab-dism:io-flip:	
204	4	嫩	nuən	3	nyən	3b	t-split:rime:	
204	5	参	ts ^h iəm	1	ts ^h am	1a	t-split:r-unglide:deret:io-flip:	
204	6	森	giəm	1	sam	1a	t-split:r-unglide:deret:io-flip:	
204	7	针	tsiəm	1	tsam	1a	t-split:retrof:deret:io-flip:unglide:	
204	8	针	tsiəm	1	tsam	1a	t-split:retrof:deret:io-flip:unglide:	
204	9	珍	tiən	1	tsan	1a	t-split:fricat:retrof:deret:io-flip:unglide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
205	1	贞	ʃiaŋ	1	tsiŋ	1a	t-split:fricat:retrof:deret:io-flip:rime:fusion:	
205	2	侦	tʰiaŋ	1	tsiŋ	1a	t-split:fricat:retrof:deret:io-flip:rime:fusion:[de-asp]:	
205	3	枕	tʃiam	2	tsam	2a	t-split:fricat:retrof:deret:io-flip:unslide:	
205	4	疹	tʃian	2	tsan	2a	t-split:fricat:retrof:deret:io-flip:unslide:	
205	5	疹	tʃian	2	tsan	2a	t-split:fricat:retrof:deret:io-flip:unslide:	
205	6	镇	ʃian	3	tsan	3a	t-split:fricat:retrof:deret:io-flip:unslide:	
205	7	阵	ʃian	3	tsan	3b	t-split:devoi:fricat:retrof:deret:io-flip:unslide:	
205	8	振	tʃian	3	tsan	3a	t-split:retrof:deret:io-flip:unslide:	
205	8	震	tʃian	3	tsan	3a	t-split:retrof:deret:io-flip:unslide:	
205	9	沉	ʃiam	1	tsʰam	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:unslide:	
206	1	陈	ʃian	1	tsʰan	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:unslide:	
206	2	尘	ʃian	1	tsʰan	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:unslide:	
206	3	晨	ʃian	1	san	1b	t-split:devoi:retrof:deret:io-flip:unslide:	
206	3	辰	ʃian	1	san	1b	t-split:devoi:retrof:deret:io-flip:unslide:	
206	4	臣	ʃian	1	san	1b	t-split:devoi:retrof:deret:io-flip:unslide:	
206	5	趁	tʰian	3	tsʰan	3a	t-split:fricat:retrof:deret:io-flip:unslide:	
206	6	衬	tʰiam	3	tsʰan	3a	t-split:r-unslide:deret:io-flip:	
206	7	称	tʰiəŋ	3	tsiŋ	3a	t-split:retrof:deret:rime:[de-asp]:	
206	8	参	ʃiam	1	sam	1a	t-split:r-unslide:deret:io-flip:	
206	9	深	ʃiam	1	sam	1a	t-split:retrof:deret:io-flip:unslide:	
207	1	申	ʃian	1	san	1a	t-split:retrof:deret:io-flip:unslide:	
207	1	身	ʃian	1	san	1a	t-split:retrof:deret:io-flip:unslide:	
207	2	伸	ʃian	1	san	1b	[t-split]:retrof:deret:io-flip:unslide:	
207	3	神	dʃian	1	san	1b	t-split:devoi:retrof:deret:io-flip:unslide:[ts- s]:	
207	4	审	ʃiam	2	sam	2a	t-split:retrof:deret:io-flip:unslide:	
207	5	审	ʃiam	2	sam	2a	t-split:retrof:deret:io-flip:unslide:	
207	6	沈	ʃiam	2	sam	2a	t-split:retrof:deret:io-flip:unslide:	
207	7	渗	ʃiam	3	sam	3a	t-split:r-unslide:deret:io-flip:	
207	8	甚	ʃiam	2	sam	3b	t-split:t2-merge:devoi:retrof:deret:io-flip:unslide:	
207	9	肾	ʃian	2	san	2b	t-split:devoi:retrof:deret:io-flip:unslide:	
208	1	慎	ʃian	3	san	3b	t-split:devoi:retrof:deret:io-flip:unslide:	
208	2	人	ŋian	1	jan	1b	t-split:glide:io-flip:pro-d:	
208	3	仁	ŋian	1	jan	1b	t-split:glide:io-flip:pro-d:	
208	4	忍	ŋian	2	jan	2b	t-split:glide:io-flip:pro-d:	
208	5	任	ŋiam	3	jam	3b	t-split:glide:io-flip:pro-d:	
208	6	刃	ŋian	3	jan	3b	t-split:glide:io-flip:pro-d:	
208	7	认	ŋian	3	jan	3b	t-split:glide:io-flip:pro-d:	
208	8	跟	kən	1	kan	1a	t-split:io-flip:	
208	9	根	kən	1	kan	1a	t-split:io-flip:	
209	1	垦	kʰən	2	han	2a	t-split:spirant:x-weak:io-flip:	
209	1	恳	kʰən	2	han	2a	t-split:spirant:x-weak:io-flip:	
209	2	肯	kʰəŋ	2	haŋ	2a	t-split:spirant:x-weak:io-flip:	
209	3	恨	ɣən	2	han	2a	[t-split]:devoi:io-flip:	
209	4	恨	ɣən	3	han	3b	t-split:devoi:io-flip:	
209	5	恩	ʔən	1	jan	1a	t-split:glide-ins:io-flip:	
209	6	寒	piən	1	pan	1a	t-split:io-flip:unslide:	
209	7	痰	piən	3	pan	3a	t-split:io-flip:unslide:	
209	8	贫			pʰiŋ	1a	:	
209	9	贫	biən	1	pʰan	1b	t-split:asp:devoi:io-flip:unslide:	
210	1	频	biən	1	pʰan	1b	t-split:asp:devoi:io-flip:unslide:	
210	2	品	pʰiam	2	pʰan	2a	t-split:io-flip:unslide:	
210	3	聘	pʰiaŋ	3	pʰiŋ	3a	t-split:io-flip:rime:fusion:	
210	4	民	mian	1	man	1b	t-split:io-flip:unslide:	
210	5	敏	mian	2	man	2b	t-split:io-flip:unslide:	
210	6	林	liəm	1	lam	1b	t-split:io-flip:unslide:	
210	6	淋	liəm	1	lam	1a	[t-split]:io-flip:unslide:	
210	6	临	liəm	1	lam	1a	[t-split]:io-flip:unslide:	
210	7	邻	liən	1	lœn	1a	[t-split]:pre-color:color:	
210	7	邻	liən	1	lœn	1b	t-split:pre-color:color:	
210	8	鳞	liən	1	lœn	1b	t-split:pre-color:color:	
210	9	赁	niəm	3	jam	3b	t-split:glide:io-flip:[n-del]:	
211	1	吝	liən	3	lœn	3b	t-split:pre-color:color:	
211	2	津	tsian	1	tsœn	1a	t-split:pre-color:color:	
211	3	今	kiam	1	kam	1a	t-split:io-flip:unslide:	
211	4	金	kiam	1	kam	1a	t-split:io-flip:unslide:	
211	5	襟	kiam	1	kʰam	1a	t-split:io-flip:unslide:[asp]:	
211	6	禁	kiam	1	kam	1a	t-split:io-flip:unslide:	
211	7	巾	kian	1	kan	1a	t-split:io-flip:unslide:	
211	8	斤	kian	1	kan	1a	t-split:io-flip:unslide:	
211	8	筋	kian	1	kan	1a	t-split:io-flip:unslide:	
211	9	尽	tsian	2	tsœn	2b	[t-split]:pre-color:color:	

Pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
212	1	察	kiən	2	kan	2a	t-split:io-flip:unghide:	
212	2	仅	giən	3	kan	2a	[t-split]:devoi:io-flip:unghide:	
212	3	禁	kiəm	3	kam	3a	t-split:io-flip:unghide:	
212	4	进	tsiən	3	tsœn	3a	t-split:pre-color:color:	
212	5	尽	dziən	2	tsœn	3b	t-split:t2-merge:devoi:pre-color:color:	
212	6	近	giən	2	k ^h an	2b	t-split:asp:devoi:io-flip:unghide:	
212	7	劲	kiən	3	kiŋ	3b	t-split:rime:[n-ŋ]:	
212	8	侵	ts ^h iəm	1	ts ^h am	1b	[t-split]:io-flip:unghide:	
212	9	亲	ts ^h iən	1	ts ^h an	1a	t-split:io-flip:unghide:	
213	1	擒	giəm	1	k ^h am	1b	t-split:asp:devoi:io-flip:unghide:	
213	1	琴	giəm	1	k ^h am	1b	t-split:asp:devoi:io-flip:unghide:	
213	1	禽	giəm	1	k ^h am	1b	t-split:asp:devoi:io-flip:unghide:	
213	2	秦	dziən	1	ts ^h œn	1b	t-split:asp:devoi:pre-color:color:	
213	3	勤	giən	1	k ^h an	1b	t-split:asp:devoi:unghide:io-flip:	
213	3	芹	giən	1	k ^h an	1b	t-split:asp:devoi:unghide:io-flip:	
213	4	寝	ts ^h iəm	2	ts ^h am	2a	t-split:io-flip:unghide:	
213	5	心	siəm	1	sam	1a	t-split:io-flip:unghide:	
213	6	新	siən	1	san	1a	t-split:io-flip:unghide:	
213	6	薪	siən	1	san	1a	t-split:io-flip:unghide:	
213	6	辛	siən	1	san	1a	t-split:io-flip:unghide:	
213	7	欣	xiən	1	jan	1a	t-split:io-flip:[pro-x]:glide:	
213	8	信	siən	3	sœn	3a	t-split:pre-color:color:	
213	9	音	?iəm	1	jam	1a	t-split:io-flip:pro-a:glide:	
214	1	阴	?iəm	1	jam	1a	t-split:io-flip:pro-a:glide:	
214	2	因	?iən	1	jan	1a	t-split:io-flip:pro-a:glide:	
214	3	姻	?iən	1	jan	1a	t-split:io-flip:pro-a:glide:	
214	4	殷	?iən	1	jan	1a	t-split:io-flip:pro-a:glide:	
214	5	银	ŋiən	1	ŋan	1a	[t-split]:io-flip:unghide:	
214	6	寅	iən	1	in	1b	t-split:rime:	
214	7	饮	?iəm	2	jam	2a	t-split:io-flip:pro-a:glide:	
214	8	隐	?iən	2	jan	2a	t-split:io-flip:pro-a:glide:	
214	9	引	iən	2	jan	2a	[t-split]:glide:io-flip:	
215	1	尹	yən	2	wan	2a	[t-split]:y-split:glide:io-flip:	
215	2	印	?iən	3	jan	3a	t-split:io-flip:pro-a:glide:	
215	3	墩	tuən	1	tan	1a	t-split:io-flip:unghide:	
215	3	顿	tuən	1	tan	1a	t-split:io-flip:unghide:	
215	4	顿	tyən	3	tœn	3b	[t-split]:color:	
215	5	盾	duən	2	t ^h œn	2b	t-split:asp:devoi:color:	acute:
215	6	钝	duən	3	tœn	3b	t-split:devoi:color:	acute:
215	7	遁	duən	3	tœn	3b	t-split:devoi:color:	acute:
215	8	吞	t ^h œn	1	t ^h an	1a	t-split:io-flip:	
215	9	屯	duən	1	t ^h yn	1b	t-split:rime:	acute:
216	1	豚	duən	1	t ^h yn	1b	t-split:rime:	acute:
216	2	仑	luən	1	lœn	1b	t-split:color:	acute:
216	3	伦	lyən	1	lœn	1a	[t-split]:color:	
216	3	轮	lyən	1	lœn	1b	t-split:color:	
216	4	论	luən	3	lœn	3b	t-split:color:	acute:
216	5	尊	tsuən	1	tsyn	1a	t-split:rime:	acute:
216	6	遵	tsyən	1	tsyn	1a	t-split:rime:	
216	7	村	ts ^h uən	1	ts ^h yn	1a	t-split:rime:	acute:
216	8	存	dzuən	1	ts ^h yn	1b	t-split:asp:devoi:rime:	acute:
216	9	寸	ts ^h uən	3	ts ^h yn	3a	t-split:rime:	acute:
217	1	孙	suən	1	s ^h yn	1a	t-split:rime:[asp]:	acute:
217	2	损	suən	2	syn	2a	t-split:rime:	acute:
217	3	算	syən	2	sœn	2a	t-split:color:	
217	4	准	tsyən	2	tsœn	2a	t-split:retrof:deret:color:	
217	5	春	ts ^h yən	1	ts ^h œn	1a	t-split:retrof:deret:color:	
217	6	唇	dzyən	1	sœn	1b	t-split:devoi:retrof:deret:color:[ts-s]:	
217	7	纯	zyən	1	sœn	1a	[t-split]:devoi:retrof:deret:color:	
217	7	莼	zyən	1	sœn	1b	t-split:devoi:retrof:deret:color:	
217	8	蠢	ts ^h yən	2	ts ^h œn	2a	t-split:retrof:deret:color:	
217	9	顺	dzyən	3	sœn	3b	t-split:devoi:retrof:deret:color:[ts-s]:	
218	1	润	nyən	3	jœn	3b	t-split:color:pro-d:glide-ins:	
218	2	闰	nyən	3	jœn	3b	t-split:color:pro-d:glide-ins:	
218	3	滚	kuən	2	kwan	2a	t-split:lab-vel:io-flip:	
218	4	棍	kuən	2	kwan	3a	:	
218	5	昆	kuən	1	k ^h uan	1a	t-split:io-flip:[asp]:	
218	5	昆	kuən	1	k ^h uan	1a	t-split:io-flip:[asp]:	
218	6	坤	k ^h uən	1	k ^h wan	1a	t-split:lab-vel:io-flip:	
218	7	捆	k ^h uən	2	k ^h wan	2a	t-split:lab-vel:io-flip:	
218	8	困	k ^h uən	3	k ^h wan	3a	t-split:lab-vel:io-flip:	
218	9	婚	xuən	1	fan	1a	t-split:lab-vel:f-coal:io-flip:	
218	9	昏	xuən	1	fan	1a	t-split:lab-vel:f-coal:io-flip:	
219	1	晕	xyən	1	fan	1a	t-split:y-split:lab-vel:f-coal:io-flip:	
219	2	浑	yuən	1	wan	1b	t-split:io-flip:pro-b:glide:	
219	3	魂	yuən	1	wan	1b	t-split:io-flip:pro-b:glide:	
219	4	混	yuən	2	wan	3b	t-split:t2-merge:io-flip:pro-b:glide:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
219	5	温	ʔuən	1	wan	1a	t-split:io-flip:pro-a:glide:	
219	5	瘟	ʔuən	1	wan	1a	t-split:io-flip:pro-a:glide:	
219	6	文	ɿuən	1	man	1b	t-split:lab-dism:io-flip:	
219	6	纹	ɿuən	1	man	1b	t-split:lab-dism:io-flip:	
219	7	闻	ɿyən	1	man	1b	t-split:lab-dnt':lab-dism:io-flip:	
219	8	魂	ʔuən	2	wan	2a	t-split:io-flip:pro-a:glide:	
219	9	问	ɿyən	3	man	3b	t-split:lab-dnt':lab-dism:io-flip:	
220	1	均	kyən	1	kwan	1a	t-split:y-split:lab-vel:io-flip:	
220	1	钧	kyən	1	kwan	1a	t-split:y-split:lab-vel:io-flip:	
220	2	君	kyən	1	kwan	1a	t-split:y-split:lab-vel:io-flip:	
220	2	军	kyən	1	kwan	1a	t-split:y-split:lab-vel:io-flip:	
220	3	俊	tsyən	3	tsən	3a	t-split:color:	
220	4	菌	gyən	2	k ^h wan	2a	[t-split]:asp:devoi:y-split:lab-vel:io-flip:	
220	5	群	gyən	1	k ^h wan	1b	t-split:asp:devoi:y-split:lab-vel:io-flip:	
220	6	裙	gyən	1	k ^h wan	1b	t-split:asp:devoi:y-split:lab-vel:io-flip:	
220	7	勋	xyən	1	fan	1a	t-split:y-split:lab-vel:f-coal:io-flip:	
220	7	薰	xyən	1	fan	1a	t-split:y-split:lab-vel:f-coal:io-flip:	
220	8	寻	ziəm	1	ts ^h am	1b	t-split:affric:asp:devoi:io-flip:unglide:	
220	9	巡	zyən	1	ts ^h ən	1b	t-split:affric:asp:devoi:color:	
220	9	循	zyən	1	ts ^h ən	1b	t-split:affric:asp:devoi:color:	
220	9	旬	zyən	1	ts ^h ən	1b	t-split:affric:asp:devoi:color:	
221	1	讯	siən	3	sən	3a	t-split:pre-color:color:	
221	2	迅	syən	3	sən	3a	t-split:color:	
221	3	训	xyən	3	fan	3a	t-split:y-split:lab-vel:f-coal:io-flip:	
221	4	匀	yən	1	wan	1b	t-split:y-split:	
221	5	云	jiyən	1	wan	1b	t-split:y-split:io-flip:[pro-j]:glide:	
221	6	允	yən	2	wan	2b	t-split:y-split:glide:io-flip:	
221	7	爨	ʔyən	3	wan	3b	[t-split]:y-split:io-flip:pro-a:glide:	
221	8	韵	jiyən	3	wan	3b	t-split:y-split:io-flip:[pro-j]:glide:	
221	9	运	jiyən	3	wan	3b	t-split:y-split:io-flip:[pro-j]:glide:	
222	1	晕	jiyən	3	wan	3b	t-split:y-split:io-flip:[pro-j]:glide:	
222	2	孕	iəŋ	3	jan	3b	t-split:glide:io-flip:[ŋ-n]:	
222	3	帮	pəŋ	1	pəŋ	1a	t-split:rnd:	
222	4	邦	pəŋ	1	pəŋ	1a	t-split:back-asm:rnd:	
222	5	榜	pəŋ	2	pəŋ	2a	t-split:rnd:	
222	6	绑	pəŋ	2	pəŋ	2a	:	
222	7	棒	baŋ	2	p ^h aŋ	2a	[t-split]:asp:devoi:length:	
222	8	旁	baŋ	1	p ^h əŋ	1a	[t-split]:asp:devoi:rnd:	
222	9	畔	p ^h aŋ	3	p ^h un	3a	t-split:back-asm:rnd:	
223	1	忙	məŋ	1	məŋ	1b	t-split:rnd:	
223	2	方	fyəŋ	1	fəŋ	1b	[t-split]:lab-dnt':lab-dism:back-asm:rnd:	
223	3	芳	f ^h yaŋ	1	fəŋ	1a	t-split:[asp-r]:lab-dnt':lab-dism:back-asm:rnd:	
223	4	妨	f ^h yaŋ	1	fəŋ	1b	[t-split]:[asp-r]:lab-dnt':lab-dism:back-asm:rnd:	
223	5	防	vyaŋ	1	fəŋ	1b	t-split:devoi:lab-dnt':lab-dism:back-asm:rnd:	
223	6	房	vyaŋ	1	fəŋ	1b	t-split:devoi:lab-dnt':lab-dism:back-asm:rnd:	
223	7	仿	fyəŋ	2	fəŋ	2a	t-split:lab-dnt':lab-dism:back-asm:rnd:	
223	8	妨	fyəŋ	2	fəŋ	2a	t-split:lab-dnt':lab-dism:back-asm:rnd:	
223	9	仿	f ^h yaŋ	2	fəŋ	2a	t-split:[asp-r]:lab-dnt':lab-dism:back-asm:rnd:	
223	9	仿	f ^h yaŋ	2	fəŋ	2a	t-split:[asp-r]:lab-dnt':lab-dism:back-asm:rnd:	
224	1	访	f ^h yaŋ	3	fəŋ	2a	[t3-t2]:t-split:[asp-r]:lab-dnt':lab-dism:back-asm:rnd:	
224	2	方	fyəŋ	3	fəŋ	3a	t-split:lab-dnt':lab-dism:back-asm:rnd:	
224	3	当	taŋ	1	fəŋ	1a	t-split:rnd:[t-f]:	
224	4	党	taŋ	2	fəŋ	2a	t-split:rnd:[t-f]:	
224	5	挡			təŋ	2a	:	
224	6	当	taŋ	3	təŋ	3a	t-split:rnd:	
224	7	汤	t ^h aŋ	1	t ^h əŋ	1a	t-split:rnd:	
224	8	唐	daŋ	1	t ^h əŋ	1b	t-split:asp:devoi:rnd:	
224	8	堂	daŋ	1	t ^h əŋ	1b	t-split:asp:devoi:rnd:	
224	8	塘	daŋ	1	t ^h əŋ	1a	[t-split]:asp:devoi:rnd:	
224	9	糖	daŋ	1	t ^h əŋ	1b	t-split:asp:devoi:rnd:	
225	1	糖	t ^h aŋ	2	t ^h əŋ	2a	t-split:rnd:	
225	2	烫	t ^h aŋ	3	t ^h əŋ	3a	t-split:rnd:	
225	3	趟	t ^h aŋ	3	t ^h əŋ	3a	t-split:rnd:	
225	4	囊	naŋ	1	nəŋ	1b	t-split:rnd:	
225	5	廊	laŋ	1	ləŋ	1b	t-split:rnd:	
225	5	狼	laŋ	1	ləŋ	1b	t-split:rnd:	
225	5	郎	laŋ	1	ləŋ	1a	[t-split]:rnd:	
225	6	浪	laŋ	3	ləŋ	3b	t-split:rnd:	
225	7	贼	tsaŋ	1	tsəŋ	1a	t-split:rnd:	
225	7	脏	tsaŋ	1	tsəŋ	1a	t-split:rnd:	
225	7	脏	tsaŋ	1	tsəŋ	1a	t-split:rnd:	

rnd-adj:

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
225	8	葬	tsaŋ	3	tsɔŋ	3a	t-split:rnd:	
225	9	脏	dzaŋ	3	tsɔŋ	3b	t-split:devoi:rnd:	
225	9	藏	dzaŋ	3	tsɔŋ	3b	t-split:devoi:rnd:	
226	1	仓	ts ^h aŋ	1	ts ^h ɔŋ	1a	t-split:rnd:	
226	2	苍	ts ^h aŋ	1	ts ^h ɔŋ	1a	t-split:rnd:	
226	3	藏	dzaŋ	1	ts ^h ɔŋ	1b	t-split:devoi:rnd:	
226	4	桑	saŋ	1	sɔŋ	1a	t-split:rnd:	
226	5	丧	saŋ	1	sɔŋ	1a	t-split:rnd:	
226	6	噪	saŋ	2	sɔŋ	2a	t-split:rnd:	
226	7	丧	saŋ	3	sɔŋ	3a	t-split:rnd:	
226	8	张	tʃiaŋ	1	tsɔŋ	1a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
226	9	樟	tʃiaŋ	1	tsɔŋ	1a	t-split:retrof:deret:back-asm:rnd:color:	
226	9	章	tʃiaŋ	1	tsɔŋ	1a	t-split:retrof:deret:back-asm:rnd:color:	
227	1	长	tʃiaŋ	2	tsɔŋ	2a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	2	涨	tʃiaŋ	2	tsɔŋ	3a	[t2-t3]t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	3	掌	tʃiaŋ	2	tsɔŋ	2a	t-split:retrof:deret:back-asm:rnd:color:	
227	4	帐	tʃiaŋ	3	tsɔŋ	3a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	4	涨	tʃiaŋ	3	tsɔŋ	3a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	4	胀	tʃiaŋ	3	tsɔŋ	3a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	4	账	tʃiaŋ	3	tsɔŋ	3a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
227	5	丈	ɖiaŋ	2	tsɔŋ	3b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:color:	
227	6	杖	ɖiaŋ	2	tsɔŋ	3b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:color:	
227	7	仗	ɖiaŋ	2	tsɔŋ	3b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:color:	
227	8	障	tʃiaŋ	2	tsɔŋ	3a	[t2-t3]t-split:retrof:deret:back-asm:rnd:color:	
227	9	昌	ts ^h iaŋ	1	ts ^h ɔŋ	1a	t-split:retrof:deret:back-asm:rnd:color:	
228	1	肠	ɖiaŋ	1	ts ^h ɔŋ	1b	t-split:asp:devoi:fricat:retrof:deret:back-asm:rnd:color:	
228	1	长	ɖiaŋ	1	ts ^h ɔŋ	1b	t-split:asp:devoi:fricat:retrof:deret:back-asm:rnd:color:	
228	2	场	ɖiaŋ	1	ts ^h ɔŋ	1b	t-split:asp:devoi:fricat:retrof:deret:back-asm:rnd:color:	
228	3	常	ziaŋ	1	sɔŋ	1b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
228	4	尝	ziaŋ	1	sɔŋ	1b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
228	5	偿	ziaŋ	1	sɔŋ	1b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
228	6	厂	ts ^h iaŋ	2	ts ^h ɔŋ	2a	t-split:retrof:deret:back-asm:rnd:color:	
228	7	畅	tʃiaŋ	3	ts ^h ɔŋ	3a	t-split:fricat:retrof:deret:back-asm:rnd:color:	
228	8	唱	ts ^h iaŋ	3	ts ^h ɔŋ	3a	t-split:retrof:deret:back-asm:rnd:color:	
228	9	倡	ts ^h iaŋ	3	ts ^h ɔŋ	3a	t-split:retrof:deret:back-asm:rnd:color:	
229	1	商	ʃiaŋ	1	sɔŋ	1a	t-split:retrof:deret:back-asm:rnd:color:	
229	2	伤	ʃiaŋ	1	sɔŋ	1a	t-split:retrof:deret:back-asm:rnd:color:	
229	3	赏	ʃiaŋ	2	sɔŋ	2a	t-split:retrof:deret:back-asm:rnd:color:	
229	4	上	ziaŋ	2	sɔŋ	2b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
229	5	尚	ziaŋ	3	sɔŋ	3b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
229	6	上	ziaŋ	3	sɔŋ	3b	t-split:devoi:retrof:deret:back-asm:rnd:color:	
229	7	帆	niaŋ	1	nɔŋ	1b	t-split:retrof:deret:back-asm:rnd:unslide:	
229	8	嚷	niaŋ	2	jɔŋ	2b	t-split:back-asm:rnd:color:pro-d:glide-ins:	
229	8	壤	niaŋ	2	jɔŋ	2b	t-split:back-asm:rnd:color:pro-d:glide-ins:	
229	8	壤	niaŋ	2	jɔŋ	2b	t-split:back-asm:rnd:color:pro-d:glide-ins:	
229	9	让	niaŋ	3	jɔŋ	3b	t-split:back-asm:rnd:color:pro-d:glide-ins:	
230	1	刚	kaŋ	1	kɔŋ	1a	t-split:rnd:	
230	1	冈	kaŋ	1	kɔŋ	1a	t-split:rnd:	
230	1	纲	kaŋ	1	kɔŋ	1a	t-split:rnd:	
230	2	钢	kaŋ	1	kɔŋ	1a	t-split:rnd:	
230	3	缸	kaŋ	1	kɔŋ	1a	t-split:rnd:	
230	4	肛	kaŋ	1	kɔŋ	1a	t-split:back-asm:rnd:	
230	5	扛	kaŋ	1	kɔŋ	1a	t-split:back-asm:rnd:	
230	6	港	kaŋ	2	kɔŋ	2a	t-split:back-asm:rnd:	
230	7	杠	kaŋ	1	kɔŋ	3a	:	
230	8	康	k ^h aŋ	1	hɔŋ	1a	t-split:spirant:x-weak:rnd:	
230	8	糠	k ^h aŋ	1	hɔŋ	1a	t-split:spirant:x-weak:rnd:	
230	9	慷	k ^h aŋ	2	k ^h ɔŋ	2a	t-split:rnd:	
231	1	抗	k ^h aŋ	3	k ^h ɔŋ	3a	t-split:rnd:	
231	2	炕	k ^h aŋ	3	k ^h ɔŋ	3a	t-split:rnd:	
231	3	抗	ɣaŋ	1	hɔŋ	1b	t-split:devoi:x-weak:rnd:	
231	3	航	ɣaŋ	1	hɔŋ	1b	t-split:devoi:x-weak:rnd:	
231	3	行	ɣaŋ	1	hɔŋ	1b	t-split:devoi:x-weak:rnd:	
231	4	航	?aŋ	1	ɔŋ	1a	t-split:pro-a:rnd:	
231	5	昂	ŋaŋ	1	ŋɔŋ	1b	t-split:rnd:	
231	6	娘	niaŋ	1	nɔŋ	1b	t-split:back-asm:rnd:color:	
231	7	凉	liaŋ	1	lɔŋ	1b	t-split:back-asm:rnd:color:	
231	7	梁	liaŋ	1	lɔŋ	1a	[t-split]:back-asm:rnd:color:	
231	7	良	liaŋ	1	lɔŋ	1b	t-split:back-asm:rnd:color:	
231	8	粮	liaŋ	1	lɔŋ	1b	t-split:back-asm:rnd:color:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
231	8	量	liaŋ	1	lœŋ	3b	[t1-t3]:t-split:back-asm:rnd:color:	
231	9	两	liaŋ	2	lœŋ	2a	[t-split]:back-asm:rnd:color:	
232	1	亮	liaŋ	3	lœŋ	3b	t-split:back-asm:rnd:color:	
232	1	量	liaŋ	3	lœŋ	3b	t-split:back-asm:rnd:color:	
232	2	凉	liaŋ	3	lœŋ	3b	t-split:back-asm:rnd:color:	
232	3	辆	liaŋ	3	lœŋ	3b	t-split:back-asm:rnd:color:	
232	4	待	tsiaŋ	1	tsœŋ	1a	t-split:back-asm:rnd:color:	
232	5	表	tsiaŋ	1	tsœŋ	1a	t-split:back-asm:rnd:color:	
232	6	僵	kiaŋ	1	kœŋ	1a	t-split:back-asm:rnd:color:	
232	6	僵	kiaŋ	1	kœŋ	1a	t-split:back-asm:rnd:color:	
232	7	姜	kiaŋ	1	kœŋ	1a	t-split:back-asm:rnd:color:	
232	7	姜	kiaŋ	1	kœŋ	1a	t-split:back-asm:rnd:color:	
232	8	江	kaŋ	1	kɔŋ	1a	t-split:back-asm:rnd:	
232	9	浆	tsiaŋ	2	tsœŋ	2a	t-split:back-asm:rnd:color:	
233	1	讲	kaŋ	2	kɔŋ	2a	t-split:back-asm:rnd:	
233	2	酱	tsiaŋ	3	tsœŋ	3a	t-split:back-asm:rnd:color:	
233	3	将	tsiaŋ	3	tsœŋ	3a	t-split:back-asm:rnd:color:	
233	4	匠	dziaŋ	3	tsœŋ	3b	t-split:devoi:back-asm:rnd:color:	
233	5	降	kaŋ	3	kɔŋ	3a	t-split:back-asm:rnd:	
233	6	枪	ts ^h iaŋ	1	ts ^h œŋ	1a	t-split:back-asm:rnd:color:	
233	7	腔	k ^h iaŋ	1	hɔŋ	1a	t-split:spirant:x-weak:back-asm:rnd:	
233	8	墙	dziaŋ	1	ts ^h œŋ	1b	t-split:asp:devoi:back-asm:rnd:color:	
233	9	强	giaŋ	1	k ^h œŋ	1b	t-split:asp:devoi:back-asm:rnd:color:	
234	1	枪	ts ^h iaŋ	2	ts ^h œŋ	2a	t-split:back-asm:rnd:color:	
234	2	厢	siaŋ	1	sœŋ	1a	t-split:back-asm:rnd:color:	
234	2	相	siaŋ	1	sœŋ	1a	t-split:back-asm:rnd:color:	
234	2	箱	siaŋ	1	sœŋ	1a	t-split:back-asm:rnd:color:	
234	2	箱	siaŋ	1	sœŋ	1a	t-split:back-asm:rnd:color:	
234	3	乡	xiaŋ	1	hœŋ	1a	t-split:x-weak:back-asm:rnd:color:	
234	4	乡	xiaŋ	1	hœŋ	1a	t-split:x-weak:back-asm:rnd:color:	
234	5	翔	ziaŋ	1	ts ^h œŋ	1a	[t-split]:affric:asp:devoi:back-asm:rnd:color:	
234	5	洋	ziaŋ	1	ts ^h œŋ	1b	t-split:affric:asp:devoi:back-asm:rnd:color:	
234	6	洋	ziaŋ	1	ts ^h œŋ	1b	t-split:affric:asp:devoi:back-asm:rnd:color:	
234	7	降	ɣaŋ	1	hɔŋ	1b	t-split:devoi:x-weak:back-asm:rnd:	
234	8	想	siaŋ	2	sœŋ	2a	t-split:back-asm:rnd:color:	
234	9	享	xiaŋ	2	hœŋ	2a	t-split:x-weak:back-asm:rnd:color:	
234	9	响	xiaŋ	2	hœŋ	2a	t-split:x-weak:back-asm:rnd:color:	
235	1	相	siaŋ	3	sœŋ	3a	t-split:back-asm:rnd:color:	
235	2	像	ziaŋ	2	tsœŋ	3b	t-split:t2-merge:affric:devoi:back-asm:rnd:color:	
235	3	橡	ziaŋ	2	tsœŋ	3b	t-split:t2-merge:affric:devoi:back-asm:rnd:color:	
235	3	象	ziaŋ	2	tsœŋ	3b	t-split:t2-merge:affric:devoi:back-asm:rnd:color:	
235	4	向	xiaŋ	3	hœŋ	3a	t-split:x-weak:back-asm:rnd:color:	
235	5	项	ɣaŋ	2	hɔŋ	3b	t-split:t2-merge:devoi:x-weak:back-asm:rnd:	
235	6	巷	ɣaŋ	3	hɔŋ	3b	t-split:devoi:x-weak:back-asm:rnd:	
235	7	央	?iaŋ	1	jœŋ	1a	t-split:pro-a:glide-ins:back-asm:rnd:color:	
235	7	殃	?iaŋ	1	jœŋ	1a	t-split:pro-a:glide-ins:back-asm:rnd:color:	
235	7	殃	?iaŋ	1	jœŋ	1a	t-split:pro-a:glide-ins:back-asm:rnd:color:	
235	8	羊	iaŋ	1	jœŋ	1b	t-split:glide-ins:back-asm:rnd:color:	
235	9	杨	iaŋ	1	jœŋ	1b	t-split:glide-ins:back-asm:rnd:color:	
235	9	杨	iaŋ	1	jœŋ	1a	[t-split]:glide-ins:back-asm:rnd:color:	
236	1	阳	iaŋ	1	jœŋ	1a	[t-split]:glide-ins:back-asm:rnd:color:	
236	1	阳	iaŋ	1	jœŋ	1b	t-split:glide-ins:back-asm:rnd:color:	
236	2	仰	ŋiaŋ	2	jœŋ	2b	t-split:back-asm:rnd:color:pro-c:glide-ins:	
236	3	养	iaŋ	2	jœŋ	2b	t-split:glide-ins:back-asm:rnd:color:	
236	4	痒	iaŋ	2	jœŋ	2b	t-split:glide-ins:back-asm:rnd:color:	
236	5	恙	iaŋ	3	jœŋ	3b	t-split:glide-ins:back-asm:rnd:color:	
236	6	样	iaŋ	3	jœŋ	3b	t-split:glide-ins:back-asm:rnd:color:	
236	7	庄	tgiaŋ	1	tsɔŋ	1a	t-split:deret:back-asm:rnd:unlide:	
236	8	装	tgiaŋ	1	tsɔŋ	1a	t-split:deret:back-asm:rnd:unlide:	
236	9	桩	ʔaŋ	1	tsɔŋ	1a	t-split:fricat:retrof:deret:back-asm:rnd:	
237	1	壮	tgiaŋ	1	tsɔŋ	3a	t-split:r-unlide:deret:back-asm:rnd:	
237	2	状	dziaŋ	3	tsɔŋ	3b	t-split:devoi:r-unlide:deret:back-asm:rnd:	
237	3	撞	ʔaŋ	3	tsɔŋ	3b	t-split:devoi:fricat:retrof:deret:back-asm:rnd:	
237	4	疮	ts ^h iaŋ	1	ts ^h ɔŋ	1a	t-split:deret:back-asm:rnd:color:	
237	5	床	ts ^h iaŋ	1	ts ^h œŋ	1a	t-split:deret:glide-ins:back-asm:rnd:color:	
237	6	床	dziaŋ	1	ts ^h ɔŋ	1a	[t-split]:asp:devoi:deret:back-asm:rnd:color:	
237	7	闯	ts ^h iaŋ	2	ts ^h ɔŋ	3a	[t2-t3]:t-split:r-unlide:deret:back-asm:rnd:	
237	8	创	ts ^h iaŋ	3	ts ^h ɔŋ	3a	t-split:r-unlide:deret:back-asm:rnd:	
237	9	霜	giaŋ	1	sœŋ	1a	t-split:deret:back-asm:rnd:color:	
238	1	双	ɣaŋ	1	sœŋ	1a	t-split:deret:glide-ins:back-asm:rnd:color:	
238	2	爽	giaŋ	2	sɔŋ	2a	t-split:back-asm:rnd:unlide:	
238	3	光	kuɑŋ	1	kwɔŋ	1a	t-split:lab-vel:rnd:	
238	4	广	kuɑŋ	2	kwɔŋ	2a	t-split:lab-vel:rnd:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
238	5	逛	kyaŋ	3	kwaŋ	3b	[t-split]:lab-vel:[unglide]:length:	
238	6	筐	k ^h yaŋ	1	həŋ	1a	t-split:spirant:x-weak:back-asm:rnd:unglide:	
238	7	狂	gyaŋ	1	k ^h wəŋ	1b	t-split:asp:devoi:lab-vel:back-asm:rnd:unglide:	
238	8	矿	k ^h uaŋ	3	k ^h wəŋ	3a	t-split:lab-vel:rnd:	
238	9	况	xyaŋ	3	fəŋ	3a	t-split:lab-vel:f-coal:back-asm:rnd:unglide:	
239	1	慌	xuaŋ	1	fəŋ	1a	t-split:lab-vel:f-coal:rnd:	
239	1	荒	xuaŋ	1	fəŋ	1a	t-split:lab-vel:f-coal:rnd:	
239	2	黄	ɣuaŋ	1	wəŋ	1b	t-split:rnd:pro-b:glide:	
239	3	皇	ɣuaŋ	1	wəŋ	1b	t-split:rnd:pro-b:glide:	
239	3	蝗	ɣuaŋ	1	wəŋ	1a	[t-split]:rnd:pro-b:glide:	
239	4	谎	xuaŋ	2	fəŋ	2a	t-split:lab-vel:f-coal:rnd:	
239	5	晃	ɣuaŋ	2	fəŋ	2a	[t-split]:f-coal:devoi:lab-vel:rnd:	
239	6	晃	ɣuaŋ	2	fəŋ	2a	:	
239	7	汪	?uaŋ	1	wəŋ	1a	t-split:rnd:pro-a:glide:	
239	8	亡	ŋyaŋ	1	məŋ	1b	t-split:lab-dnt':lab-dism:back-asm:rnd:	
239	9	王	jyaŋ	1	wəŋ	1b	t-split:back-asm:rnd:unglide:[pro-j]:glide:	
240	1	网	ŋyaŋ	2	məŋ	2b	t-split:lab-dnt':lab-dism:back-asm:rnd:	
240	2	往	jyaŋ	2	wəŋ	2b	t-split:back-asm:rnd:unglide:[pro-j]:glide:	
240	3	忘	ŋyaŋ	3	məŋ	1b	[t3-t1]:t-split:lab-dnt':lab-dism:back-asm:rnd:	
240	4	望	ŋyaŋ	3	məŋ	3b	t-split:lab-dnt':lab-dism:back-asm:rnd:	
240	5	旺	jyaŋ	3	wəŋ	3b	t-split:back-asm:rnd:unglide:[pro-j]:glide:	
240	6	崩	pəŋ	1	paŋ	1a	t-split:io-flip:	
240	7	烹	p ^h aŋ	1	p ^h a:ŋ	1a	t-split:length:fusion:	
240	8	朋	bəŋ	1	p ^h aŋ	1b	t-split:asp:devoi:io-flip:	
240	9	棚	bəŋ	1	p ^h a:ŋ	1b	t-split:asp:devoi:io-flip:[length]:	
241	1	彭	baŋ	1	p ^h a:ŋ	1b	t-split:asp:devoi:length:fusion:	
241	2	棚	baŋ	1	p ^h a:ŋ	1b	t-split:asp:devoi:length:fusion:	
241	3	篷	buaŋ	1	p ^h uaŋ	1b	t-split:asp:devoi:	
241	3	蓬	buaŋ	1	p ^h uaŋ	1b	t-split:asp:devoi:	
241	4	捧	f ^h iuŋ	2	p ^h uaŋ	2a	t-split:lab-dnt':unglide:[fh-ph]:	
241	5	碰			p ^h uaŋ	3a	:	
241	6	盟	miaŋ	1	maŋ	1b	t-split:lab-dnt:fusion:	
241	7	蒙	muŋ	1	muŋ	1b	t-split:	
241	8	猛	maŋ	2	ma:ŋ	2b	t-split:length:fusion:	
241	9	孟	maŋ	3	ma:ŋ	3b	t-split:length:fusion:	
242	1	梦	miuŋ	3	muŋ	3b	t-split:lab-dnt:	
242	2	疯	fiuŋ	1	fuŋ	1a	t-split:lab-dnt':	
242	2	风	fiuŋ	1	fuŋ	1a	t-split:lab-dnt':	
242	3	丰	f ^h iuŋ	1	fuŋ	1a	t-split:[asp-r]:lab-dnt':	
242	4	封	fiuŋ	1	fuŋ	1a	t-split:lab-dnt':	
242	5	峰	f ^h iuŋ	1	fuŋ	1a	t-split:[asp-r]:lab-dnt':	
242	5	峰	f ^h iuŋ	1	fuŋ	1a	t-split:[asp-r]:lab-dnt':	
242	6	蜂	f ^h iuŋ	1	fuŋ	1a	t-split:[asp-r]:lab-dnt':	
242	7	冯	viuŋ	1	fuŋ	1b	t-split:devoi:lab-dnt':	
242	8	逢	viuŋ	1	fuŋ	1b	t-split:devoi:lab-dnt':	
242	9	缝	viuŋ	1	fuŋ	1b	t-split:devoi:lab-dnt':	
243	1	讽	fiuŋ	3	fuŋ	3a	t-split:lab-dnt':	
243	2	风	viuŋ	3	fuŋ	3b	t-split:devoi:lab-dnt':	
243	3	奉	viuŋ	2	fuŋ	3b	t-split:t2-merge:devoi:lab-dnt':	
243	4	俸	viuŋ	3	fuŋ	2b	[t3-t2]:t-split:devoi:lab-dnt':	
243	5	缝	viuŋ	3	fuŋ	3b	t-split:devoi:lab-dnt':	
243	6	灯	təŋ	1	taŋ	1a	t-split:io-flip:	
243	6	登	təŋ	1	taŋ	1a	t-split:io-flip:	
243	7	等	təŋ	2	taŋ	2a	t-split:io-flip:	
243	8	凳	təŋ	3	taŋ	3a	t-split:io-flip:	
243	9	邓	dəŋ	3	taŋ	3b	t-split:devoi:io-flip:	
244	1	瞪	d ^h iəŋ	3	taŋ	3b	t-split:devoi:retrof:deret:io-flip:unglide:	
244	2	腾	dəŋ	1	t ^h aŋ	1b	t-split:asp:devoi:io-flip:	
244	3	棠	dəŋ	1	t ^h aŋ	1b	t-split:asp:devoi:io-flip:	
244	4	棠	dəŋ	1	t ^h aŋ	1b	t-split:asp:devoi:io-flip:	
244	5	疼	dəŋ	1	t ^h aŋ	1b	t-split:asp:devoi:io-flip:	
244	6	能	nəŋ	1	naŋ	1b	t-split:io-flip:	
244	7	校	ləŋ	1	liŋ	1b	t-split:[schwa-i]:	
244	8	冷	laŋ	2	la:ŋ	2b	t-split:length:fusion:	
244	9	增	tsəŋ	1	tsaŋ	1a	t-split:io-flip:	
245	1	曾	tsəŋ	1	tsaŋ	1a	t-split:io-flip:	
245	2	赠	dzəŋ	3	tsaŋ	3b	t-split:devoi:io-flip:	
245	3	赠	dzəŋ	1	ts ^h aŋ	1b	t-split:asp:devoi:io-flip:	
245	4	层	dzəŋ	1	ts ^h aŋ	1b	t-split:asp:devoi:io-flip:	
245	5	憎	səŋ	1	tsaŋ	1a	t-split:io-flip:[s-ts]:	
245	6	蒸	t ^h iəŋ	1	tsiŋ	1a	t-split:retrof:deret:rime:	
245	7	争	t ^h aŋ	1	tsa:ŋ	1a	t-split:deret:length:fusion:	
245	8	争	t ^h aŋ	1	tsa:ŋ	1a	t-split:deret:length:fusion:	
245	9	睁	t ^h aŋ	1	tsa:ŋ	1a	t-split:deret:length:fusion:	
246	1	正	t ^h iaŋ	1	tsiŋ	1a	t-split:retrof:deret:io-flip:rime:fusion:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
246	2	征	t́siań	1	tsiŋ	1a	t-split:retrof:deret:io-flip:rime:fusion:	
246	3	整	t́siań	2	tsiŋ	2a	t-split:retrof:deret:io-flip:rime:fusion:	
246	4	症	t́siaŋ	3	tsiŋ	3a	t-split:retrof:deret:io-flip:rime:	
246	4	证	t́siaŋ	3	tsiŋ	3a	t-split:retrof:deret:io-flip:rime:	
246	5	郑	điań	3	tseŋ	3b	t-split:devoi:sharp:fricat:retrof:deret:pal-dism:fusion:	
246	6	正	t́siań	3	tsiŋ	3a	t-split:retrof:deret:io-flip:rime:fusion:	
246	7	政	t́siań	3	tsiŋ	3a	t-split:retrof:deret:io-flip:rime:fusion:	
246	8	称	tsʰiaŋ	1	tsiŋ	1a	t-split:retrof:deret:rime:[de-asp]:	
246	9	鼎	tʰiań	1	tsʰaŋ	1a	t-split:fricat:retrof:deret:length:fusion:unlide:	
247	1	乘	dziəŋ	1	siŋ	1b	t-split:affric:devoi:retrof:deret:rime:	
247	2	丞	ziəŋ	1	siŋ	1b	t-split:devoi:retrof:deret:rime:	
247	2	承	ziəŋ	1	siŋ	1b	t-split:devoi:retrof:deret:rime:	
247	3	橙	đań	1	tsʰaŋ	1b	t-split:asp:devoi:fricat:retrof:deret:length:fusion:	
247	4	呈	điań	1	tsʰiŋ	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:fusion:	
247	5	程	điań	1	tsʰiŋ	1b	t-split:asp:devoi:fricat:retrof:deret:io-flip:rime:fusion:	
247	6	成	ziań	1	siŋ	1b	t-split:devoi:retrof:deret:io-flip:rime:fusion:	
247	7	城	ziań	1	siŋ	1b	t-split:devoi:retrof:deret:io-flip:rime:fusion:	
247	8	减	ziań	1	siŋ	1b	t-split:devoi:retrof:deret:io-flip:rime:fusion:	
247	9	盛	ziań	1	siŋ	1b	t-split:devoi:retrof:deret:io-flip:rime:fusion:	
248	1	秤	tsʰiaŋ	3	tsʰiŋ	3a	t-split:retrof:deret:rime:	
248	2	升	siəŋ	1	siŋ	1a	t-split:retrof:deret:rime:	
248	3	生	gań	1	saŋ	1a	t-split:deret:length:fusion:	
248	4	甥	gań	1	saŋ	1a	t-split:deret:length:fusion:	
248	5	牲	gań	1	saŋ	1a	t-split:deret:length:fusion:	
248	6	声	šiań	1	siŋ	1a	t-split:retrof:deret:io-flip:rime:fusion:	
248	7	绳	dziəŋ	1	siŋ	2a	[t1-t2]:[t-split]:affric:devoi:retrof:deret:rime:	
248	8	省	giań	2	saŋ	2a	t-split:deret:pal-dism:fusion:	
248	9	剩	dziəŋ	3	tsiŋ	3b	t-split:devoi:retrof:deret:rime:	
249	1	胜	siəŋ	3	siŋ	3a	t-split:retrof:deret:rime:	
249	2	圣	šiań	3	siŋ	3a	t-split:retrof:deret:io-flip:rime:fusion:	
249	3	盛	ziań	3	siŋ	3b	t-split:devoi:retrof:deret:io-flip:rime:fusion:	
249	4	扔	riəŋ	1	jiŋ	1b	t-split:rime:pro-d:glide-ins:	
249	5	仍	riəŋ	1	jiŋ	1b	t-split:rime:pro-d:glide-ins:	
249	6	更	kań	1	kaŋ	1a	t-split:fusion:	
249	7	更	kań	1	kaŋ	1a	t-split:length:fusion:	
249	8	羹	kań	1	kaŋ	1a	t-split:fusion:	
249	9	庚	kań	1	kaŋ	1a	t-split:fusion:	
250	1	耕	kań	1	kaŋ	1a	t-split:length:fusion:	
250	2	更	kań	3	kaŋ	3a	t-split:fusion:	
250	3	坑	kʰań	1	haŋ	1a	t-split:spirant:x-weak:length:fusion:	
250	4	恒	ɣəŋ	1	haŋ	1b	t-split:devoi:x-weak:io-flip:	
250	5	衡	ɣań	1	haŋ	1b	t-split:devoi:x-weak:fusion:	
250	6	横	ɣuań	1	waŋ	1b	t-split:length:fusion:pro-b:glide:	
250	7	横	ɣuań	3	waŋ	3b	t-split:length:fusion:pro-b:glide:	
250	8	冰	piəŋ	1	piŋ	1a	t-split:rime:	
250	9	兵	piań	1	piŋ	1a	t-split:io-flip:rime:fusion:	
251	1	乘	piań	2	piŋ	2a	t-split:io-flip:rime:fusion:	
251	2	丙	piań	2	piŋ	2a	t-split:io-flip:rime:fusion:	
251	3	饼	piań	2	piŋ	2a	t-split:io-flip:rime:fusion:	
251	4	病	biań	3	piŋ	3b	t-split:devoi:io-flip:rime:fusion:	
251	5	井	piań	3	piŋ	3a	t-split:io-flip:rime:fusion:	
251	6	并	biań	2	piŋ	3b	t-split:t2-merge:devoi:io-flip:rime:fusion:	
251	7	凭	biəŋ	1	pʰaŋ	1b	t-split:asp:devoi:io-flip:unlide:	
251	8	平	biəŋ	1	pʰiŋ	1b	t-split:asp:devoi:rime:	
251	9	坪	biəŋ	1	pʰiŋ	1b	t-split:asp:devoi:rime:	
252	1	坪	biań	1	pʰiŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
252	2	瓶	biań	1	pʰiŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
252	3	屏	biań	1	pʰiŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
252	4	萍	biań	1	pʰiŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
252	5	明	miań	1	miŋ	1b	t-split:io-flip:rime:fusion:	
252	6	鸣	miań	1	miŋ	1b	t-split:io-flip:rime:fusion:	
252	7	名	miań	1	miŋ	1b	t-split:io-flip:rime:fusion:	
252	8	铭	miań	1	miŋ	1b	t-split:io-flip:rime:fusion:	
252	9	命	miań	3	miŋ	3b	t-split:io-flip:rime:fusion:	
253	1	丁	tiań	1	tiŋ	1a	t-split:io-flip:rime:fusion:	
253	2	钉	tiań	1	tiŋ	1a	t-split:io-flip:rime:fusion:	
253	3	顶	tiań	2	tiŋ	2a	t-split:io-flip:rime:fusion:	
253	4	鼎	tiań	2	tʰiŋ	2a	t-split:io-flip:rime:fusion:[asp]:	
253	5	钉	tiań	3	tiŋ	3a	t-split:io-flip:rime:fusion:	
253	6	订	tiań	3	tiŋ	3a	t-split:io-flip:rime:fusion:	
253	7	定	điań	3	tʰiŋ	3b	t-split:asp:devoi:io-flip:rime:fusion:	
253	8	听	tʰiań	1	tʰiŋ	1a	t-split:io-flip:rime:fusion:	
253	9	厅	tʰiań	1	tʰeŋ	1a	t-split:sharp:pal-dism:fusion:	
254	1	亭	điań	1	tʰiŋ	1b	t-split:asp:devoi:io-flip:rime:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
254	2	停	diań	1	t ^h ig	1b	t-split:asp:devoi:io-flip:rime:	
254	3	廷	diań	1	t ^h ig	1b	t-split:asp:devoi:io-flip:rime:	
254	4	庭	diań	1	t ^h ig	1b	t-split:asp:devoi:io-flip:rime:	
254	5	艇	diań	2	t ^h ig	2b	t-split:asp:devoi:io-flip:rime:	
254	6	挺	diań	2	t ^h ig	2b	t-split:asp:devoi:io-flip:rime:	
254	7	凝	ŋiəŋ	1	jiŋ	1b	t-split:rime:pro-c:glide-ins:	
254	8	宁	niań	1	niŋ	1b	t-split:io-flip:rime:fusion:	
254	9	凌	liəŋ	1	liŋ	1b	t-split:rime:	
254	9	菱	liəŋ	1	liŋ	1b	t-split:rime:	
254	9	陵	liəŋ	1	liŋ	1b	t-split:rime:	
255	1	灵	liań	1	liŋ	1b	t-split:io-flip:rime:fusion:	
255	2	铃	liań	1	liŋ	1b	t-split:io-flip:rime:fusion:	
255	2	零	liań	1	liŋ	1a	[t-split]:io-flip:rime:fusion:	
255	3	伶	liań	1	liŋ	1b	t-split:io-flip:rime:fusion:	
255	4	领	liań	2	liŋ	2b	t-split:io-flip:rime:fusion:	
255	5	岭	liań	2	liŋ	2b	t-split:io-flip:rime:fusion:	
255	6	令	liań	3	liŋ	3b	t-split:io-flip:rime:fusion:	
255	7	另			liŋ	3b	:	
255	8	京	kiań	1	kiŋ	1a	t-split:io-flip:rime:fusion:	
255	9	荆	kiań	1	kiŋ	1a	t-split:io-flip:rime:fusion:	
256	1	惊	kiań	1	keŋ	1a	t-split:sharp:pal-dism:fusion:	
256	2	精	tsiań	1	tsiŋ	1a	t-split:io-flip:rime:fusion:	
256	3	晶	tsiań	1	tsiŋ	1a	t-split:io-flip:rime:fusion:	
256	4	睛	tsiań	1	tsiŋ	1a	t-split:io-flip:rime:fusion:	
256	5	经	kiań	1	kiŋ	1a	t-split:io-flip:rime:fusion:	
256	6	景	kiań	2	kiŋ	2a	t-split:io-flip:rime:fusion:	
256	6	警	kiań	2	kiŋ	2a	t-split:io-flip:rime:fusion:	
256	7	井	tsiań	2	tsiŋ	2a	t-split:io-flip:rime:fusion:	
256	8	颈	kiań	2	keŋ	2a	t-split:sharp:pal-dism:fusion:	
256	9	境	kiań	2	kiŋ	2a	t-split:io-flip:rime:fusion:	
257	1	竟	kiań	3	kiŋ	2a	t-split:io-flip:rime:fusion:	
257	2	敬	kiań	3	kiŋ	3a	t-split:io-flip:rime:fusion:	
257	3	镜	kiań	3	keŋ	3a	t-split:sharp:pal-dism:fusion:	
257	4	竟	giań	3	kiŋ	3b	t-split:devoi:io-flip:rime:fusion:	
257	5	静	dziań	2	tsiŋ	3b	t-split:t2-merge:devoi:io-flip:rime:fusion:	
257	6	净	dziań	3	tsiŋ	3b	t-split:devoi:io-flip:rime:fusion:	
257	7	轻	k ^h iań	1	hiŋ	1a	t-split:spirant:x-weak:io-flip:rime:fusion:	
257	8	清	ts ^h iań	1	ts ^h iŋ	1a	t-split:io-flip:rime:fusion:	
257	9	青	ts ^h iań	1	ts ^h iŋ	1a	t-split:io-flip:rime:fusion:	
258	1	倾	k ^h iań	1	k ^h iŋ	1a	t-split:io-flip:rime:fusion:	
258	2	情	dziań	1	ts ^h iŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
258	3	晴	dziań	1	ts ^h iŋ	1b	t-split:asp:devoi:io-flip:rime:fusion:	
258	4	请	ts ^h iań	2	ts ^h iŋ	2a	t-split:io-flip:rime:fusion:	
258	5	顷	k ^h yań	2	k ^h iŋ	2a	t-split:io-flip:rime:fusion:unglide:	
258	6	庆	k ^h iań	3	hiŋ	3a	t-split:spirant:x-weak:io-flip:rime:fusion:	
258	7	兴	xiaŋ	1	hiŋ	1a	t-split:x-weak:rime:	
258	8	星	siań	1	siŋ	1a	t-split:io-flip:rime:fusion:	
258	9	腥	siań	1	siŋ	1a	t-split:io-flip:rime:fusion:	
259	1	行	yań	1	haŋ	1b	t-split:devoi:x-weak:fusion:	
259	2	刑	yań	1	jiŋ	1a	[t-split]:io-flip:rime:fusion:pro-b:glide-ins:	
259	2	型	yań	1	jiŋ	1a	[t-split]:io-flip:rime:fusion:pro-b:glide-ins:	
259	2	形	yań	1	jiŋ	1b	t-split:io-flip:rime:fusion:pro-b:glide-ins:	
259	3	省	siań	2	siŋ	2a	t-split:io-flip:rime:fusion:	
259	4	醒	siań	2	siŋ	2a	t-split:io-flip:rime:fusion:	
259	5	兴	xiaŋ	3	hiŋ	3a	t-split:x-weak:rime:	
259	6	杏	yań	2	haŋ	3b	t-split:t2-merge:devoi:x-weak:fusion:	
259	7	幸	yań	2	haŋ	3b	t-split:t2-merge:devoi:x-weak:fusion:	
259	8	性	siań	3	siŋ	3a	t-split:io-flip:rime:fusion:	
259	9	姓	siań	3	siŋ	3a	t-split:io-flip:rime:fusion:	
260	1	应	ŋiəŋ	1	jiŋ	1a	t-split:rime:pro-a:glide-ins:	
260	1	鹰	ŋiəŋ	1	jiŋ	1a	t-split:rime:pro-a:glide-ins:	
260	2	櫻	ŋań	1	jiŋ	1a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
260	2	鸶	ŋań	1	jiŋ	1a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
260	3	英	ŋiań	1	jiŋ	1a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
260	4	嬰	ŋiań	1	jiŋ	1a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
260	5	纓	ŋiań	1	jiŋ	1a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
260	6	蝇	iəŋ	1	jiŋ	1b	t-split:rime:glide-ins:	
260	7	迎	ŋiań	1	jiŋ	1b	t-split:io-flip:rime:fusion:pro-c:glide-ins:	
260	8	盈	iań	1	jiŋ	1b	t-split:io-flip:rime:fusion:glide-ins:	
260	9	赢	iań	1	jiŋ	1b	t-split:io-flip:rime:fusion:glide-ins:	
261	1	营	yań	1	jiŋ	1b	t-split:io-flip:rime:fusion:glide-ins:unglide:	
261	2	萤	yyań	1	jiŋ	1b	t-split:io-flip:rime:fusion:pro-b:glide-ins:unglide:	
261	3	影	ŋiań	2	jiŋ	2a	t-split:io-flip:rime:fusion:pro-a:glide-ins:	
261	4	应	ŋiəŋ	3	jiŋ	3a	t-split:rime:pro-a:glide-ins:	
261	5	硬	ŋań	3	ŋaŋ	3b	t-split:length:fusion:	
261	6	东	tuŋ	1	tuŋ	1a	t-split:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
261	7	冬	tug	1	tug	1a	t-split:	
261	8	懂	tug	2	tug	2a	t-split:	
261	8	董	tug	2	tug	2a	t-split:	
261	9	冻	tug	3	tug	3a	t-split:	
262	1	动	dug	2	tug	3b	t-split:t2-merge:devoi:	
262	2	洞	dug	3	tug	3b	t-split:devoi:	
262	3	通	t ^h ug	1	t ^h ug	1a	t-split:	
262	4	同	dug	1	t ^h ug	1b	t-split:asp:devoi:	
262	5	筒	dug	1	t ^h ug	1b	t-split:asp:devoi:	
262	5	铜	dug	1	t ^h ug	1b	t-split:asp:devoi:	
262	6	铜	dug	1	t ^h ug	1b	t-split:asp:devoi:	
262	7	瞳	dug	1	t ^h ug	1a	[t-split]:asp:devoi:	
262	7	童	dug	1	t ^h ug	1b	t-split:asp:devoi:	
262	8	桶	t ^h ug	2	t ^h ug	2a	t-split:	
262	9	统	t ^h ug	3	t ^h ug	2a	[t3-t2]:t-split:	
263	1	痛	t ^h ug	3	t ^h ug	3a	t-split:	
263	2	农	nug	1	nug	1b	t-split:	
263	3	脓	nug	1	nug	1b	t-split:	
263	4	浓	nug	1	nug	1b	t-split:	
263	5	弄	lug	3	lug	3b	t-split:	
263	6	笼	lug	1	lug	1b	t-split:	
263	7	窿	lug	1	lug	1b	t-split:	
263	8	隆	liug	1	lug	1b	t-split:unlide:	
263	9	龙	liug	1	lug	1b	t-split:unlide:	
264	1	拢	lug	2	lug	2b	t-split:	
264	2	院	liug	2	lug	2b	t-split:unlide:	
264	2	垆	liug	2	lug	2b	t-split:unlide:	
264	3	株	tsug	1	tsug	1a	t-split:	
264	4	擘	tsug	1	tsug	1a	t-split:	
264	5	宗	tsug	1	tsug	1a	t-split:	
264	6	踪	tsiug	1	tsug	1a	t-split:unlide:	
264	7	踪	tsug	2	tsug	2a	t-split:	
264	8	粽	tsug	3	tsog	3a	t-split:[u-o]:	
264	9	纵	tsiug	3	tsug	3a	t-split:unlide:	
265	1	囱	ts ^h ug	1	ts ^h ug	1a	t-split:	
265	1	匆	ts ^h ug	1	ts ^h ug	1a	t-split:	
265	1	聰	ts ^h ug	1	ts ^h ug	1a	t-split:	
265	1	葱	ts ^h ug	1	ts ^h ug	1a	t-split:	
265	2	丛	dzug	1	ts ^h ug	1b	t-split:asp:devoi:	
265	3	从	dziug	1	ts ^h ug	1b	t-split:asp:devoi:unlide:	
265	4	松	sug	1	sug	1a	t-split:	
265	5	松	sug	1	ts ^h ug	1b	[t-split]:[asp]:[s-ts]:	
265	6	送	sug	3	sug	3a	t-split:	
265	7	宋	sug	3	sug	3a	t-split:	
265	8	讼	ziug	3	tsug	3b	t-split:affric:devoi:unlide:	
265	8	诵	ziug	3	tsug	3b	t-split:affric:devoi:unlide:	
265	8	颂	ziug	3	tsug	3b	t-split:affric:devoi:unlide:	
265	9	中	fiug	1	tsug	1a	t-split:fricat:retrof:deret:unlide:	
265	9	忠	fiug	1	tsug	1a	t-split:fricat:retrof:deret:unlide:	
266	1	终	tsiug	1	tsug	1a	t-split:retrof:deret:unlide:	
266	2	钟	tsiug	1	tsug	1a	t-split:retrof:deret:unlide:	
266	2	钟	tsiug	1	tsug	1a	t-split:retrof:deret:unlide:	
266	3	种	tsiug	2	tsug	2a	t-split:retrof:deret:unlide:	
266	4	中	fiug	3	tsug	3a	t-split:fricat:retrof:deret:unlide:	
266	5	仲	diug	3	tsug	3b	t-split:devoi:fricat:retrof:deret:unlide:	
266	6	众	tsiug	3	tsug	3a	t-split:retrof:deret:unlide:	
266	7	重	diug	2	tsug	3a	[t2-t3]:[t-split]:devoi:fricat:retrof:deret:unlide:	
266	8	种	tsiug	3	tsug	3a	t-split:retrof:deret:unlide:	
266	9	充	ts ^h iug	1	ts ^h ug	1a	t-split:retrof:deret:unlide:	
267	1	冲	ts ^h iug	1	ts ^h ug	1a	t-split:retrof:deret:unlide:	
267	2	蠢	siug	1	ts ^h ug	1a	t-split:retrof:deret:unlide:[asp]:[s-ts]:	
267	3	虫	diug	1	ts ^h ug	1b	t-split:asp:devoi:fricat:retrof:deret:unlide:	
267	4	崇	dziug	1	sug	1b	t-split:devoi:r-unlide:deret:[ts-s]:	
267	5	重	diug	1	ts ^h ug	1b	t-split:asp:devoi:fricat:retrof:deret:unlide:	
267	6	宠	t ^h iug	2	ts ^h ug	2a	t-split:fricat:retrof:deret:unlide:	
267	7	荣	jyan	1	wig	1b	t-split:geng-zeng:io-flip:rime:[pro-j]:glide:	
267	8	戎	niug	1	jug	1b	t-split:pro-d:glide:	
267	9	绒	niug	1	jug	1b	t-split:pro-d:glide:	
268	1	茸	niug	1	jug	1a	[t-split]:pro-d:glide:	
268	2	融	iug	1	jug	1b	t-split:glide:	
268	3	蓉	iug	1	jug	1b	t-split:glide:	
268	3	蓉	iug	1	jug	1b	t-split:glide:	
268	3	熔	iug	1	jug	1a	[t-split]:glide:	
268	4	公	kug	1	kug	1a	t-split:	
268	5	功	kug	1	kug	1a	t-split:	
268	5	工	kug	1	kug	1a	t-split:	
268	5	攻	kug	1	kug	1a	t-split:	

pg	co	ch	smc	tn	mc	tn	Diachronic	Allophonic
268	6	弓	kiuŋ	1	kuŋ	1a	t-split:unglide:	
268	7	宫	kiuŋ	1	kuŋ	1a	t-split:unglide:	
268	7	躬	kiuŋ	1	kuŋ	1a	t-split:unglide:	
268	8	恭	kiuŋ	1	kuŋ	1a	t-split:unglide:	
268	9	供	kiuŋ	1	kuŋ	1a	t-split:unglide:	
269	1	拱	kiuŋ	2	kuŋ	2a	t-split:unglide:	
269	1	巩	kiuŋ	2	kuŋ	2a	t-split:unglide:	
269	2	贡	kuŋ	3	kuŋ	3a	t-split:	
269	3	供	kiuŋ	3	kuŋ	3a	t-split:unglide:	
269	4	共	giuŋ	3	kuŋ	3b	t-split:devoi:unglide:	
269	5	空	k ^h iuŋ	1	huŋ	1a	t-split:spirant:x-weak:unglide:	
269	6	孔	k ^h uŋ	2	huŋ	2a	t-split:spirant:x-weak:	
269	7	恐	k ^h iuŋ	2	huŋ	2a	t-split:spirant:x-weak:unglide:	
269	8	控	k ^h uŋ	3	kuŋ	3a	t-split:[de-asp]:	
269	9	空	k ^h uŋ	3	kuŋ	3a	t-split:[de-asp]:	
270	1	轰	xyaŋ	1	kwaŋ	1a	t-split:geng-zeng:x-lab:io-flip:unglide:[x-kh]:[de-asp]:	
270	2	烘	xuŋ	1	huŋ	1a	t-split:x-weak:	
270	3	弘	ɣuəŋ	1	waŋ	1b	t-split:io-flip:pro-b:glide:	
270	4	宏	ɣuaŋ	1	waŋ	1b	t-split:fusion:pro-b:glide:	
270	5	红	ɣuŋ	1	huŋ	1b	t-split:devoi:x-weak:	
270	6	洪	ɣuŋ	1	huŋ	1b	t-split:devoi:x-weak:	
270	7	鸿	ɣuŋ	1	huŋ	1b	t-split:devoi:x-weak:	
270	8	虹	ɣuŋ	1	huŋ	1b	t-split:devoi:x-weak:	
270	9	哄			huŋ	3b	:	
271	1	翁	?uŋ	1	juŋ	1a	t-split:pro-a:[j-ins]:	
271	2	瓮	?uŋ	3	wuŋ	3a	t-split:pro-a:glide-ins:	
271	3	窘	gyəŋ	2	k ^h wan	3a	[t2-t3]:[t-split]:asp:devoi:y-split:lab-vel:io-flip:	
271	4	琼	gyaŋ	1	k ^h iŋ	1b	t-split:asp:devoi:geng-zeng:rime:unglide:	
271	5	旁	giuŋ	1	k ^h uŋ	1b	t-split:asp:devoi:unglide:	
271	6	兄	xyaŋ	1	hiŋ	1a	t-split:y-split:x-weak:geng-zeng:rime:unglide:	
271	7	凶	xiuŋ	1	huŋ	1a	t-split:x-weak:unglide:	
271	7	凶	xiuŋ	1	huŋ	1a	t-split:x-weak:unglide:	
271	8	胸	xiuŋ	1	huŋ	1a	t-split:x-weak:unglide:	
271	9	熊	jiuŋ	1	huŋ	1b	t-split:unglide:[j-h]:	
272	1	雄	jiuŋ	1	huŋ	1b	t-split:unglide:[j-h]:	
272	2	拥	?iuŋ	1	juŋ	2a	[t1-t2]:t-split:pro-a:glide:	
272	3	永	jyaŋ	2	wiŋ	2b	t-split:geng-zeng:rime:[pro-j]:[glide]:	
272	4	勇	iuŋ	2	juŋ	2b	t-split:glide:	
272	5	用	iuŋ	3	juŋ	3b	t-split:glide:	

References

- Abramson, N. (1963). *Information theory and coding*. New York: McGraw-Hill.
- Aho, A. V., R. Sethi, and J. D. Ullman (1986). *Compilers: Principles, techniques and tools*. Reading, Massachusetts: Addison-Wesley.
- Aho, A. V. and J. D. Ullman (1972). *The theory of parsing, translation and compiling, Volume 1: Parsing*. New Jersey: Prentice-Hall.
- Aho, A. V. and J. D. Ullman (1995). *Foundations of computer Science, C edition*. New York: Computer Science Press, W. H. Freeman and Co.
- Allison, L., C. S. Wallace, and C. N. Yee (1990a). Inductive inference over macro-molecules. In *The theory and application of minimal-length encoding*, AAAI 90 Spring symposium series, pp. 50–54. Stanford, California.
- Allison, L., C. S. Wallace, and C. N. Yee (1990b). When is a string like a string? In *Proceedings of the conference on artificial intelligence and mathematics*, Lauderdale, Florida. Pages not numbered.
- Allison, L., C. S. Wallace, and C. N. Yee (1992). Finite state models in the alignment of macro-molecules. *Journal of Molecular Evolution* 35(1), 77–89.
- Allison, L. and C. N. Yee (1990). Minimum Message Length encoding and the comparison of macromolecules. *Bulletin of Mathematical Biology* 52(3), 431–453.
- Andreae, J. H. (1972). Man-machine studies reports. Technical Report UC-DSE/1–7, University of Canterbury, Christchurch, New Zealand. Series, 1972–1975.
- Angluin, D. (1978). On the complexity of minimum inference of regular sets. *Information and Control* 39, 337–350.
- Angluin, D. (1982). Inference of reversible languages. *Journal of the ACM* 29, 741–765.
- Angluin, D. (1987). Learning regular sets for queries and counterexamples. *Information and Computation* 75(2), 87–106.
- Angluin, D. and C. H. Smith (1983). Inductive inference: Theory and methods. *Computing Surveys* 15(3), 237–269.
- Arendrup, B. (1994). Chinese. In R. E. Asher (Ed.), *The encyclopaedia of language and linguistics*, pp. 516–524. Oxford: Pergamon Press.

- Arnold, G. S. and J. L. Kinsey (1977). Information theory for marginal distributions: Application to energy disposal in an exothermic reaction. *Journal of Chemical Physics* 67(8), 3530–3532.
- Austerlitz, R. (1991). Alternatives in long-range comparison. In S. L. Lamb and E. D. Mitchell (Eds.), *Sprung from some common source*, pp. 353–364. Stanford, California: Stanford University Press.
- Babcock, M. S., W. K. Olson, and E. P. D. Pednault (1990). The use of the Minimum Description Length principle to segment DNA into structural and functional domains. In *The theory and application of minimal-length encoding*, AAAI 90 Spring symposium series, pp. 40–44. Stanford, California.
- Baxter, R. A. and J. J. Oliver (1995). MDL and MML: Similarities and differences. Tech Report TR-94/207, Monash University, Computer Science Department, Clayton, Victoria.
- Bell, T. C., J. G. Cleary, and I. H. Witten (1990). *Text compression*. New Jersey: Prentice-Hall.
- Bergsland, K. and H. Vogt (1962). On the validity of glottochronology. *Current Anthropology* 3(2), 115–153.
- Biermann, A. W. and J. A. Feldman (1972). On the synthesis of finite-state machines from samples of their behaviour. *IEEE Transactions on Computers* 21, 591–597.
- Black, P. (1976). Multidimensional scaling applied to linguistic relationships. *Cahiers d'Institut de Linguistique de Louvain* 3, 43–92.
- Boulton, D. M. and C. S. Wallace (1970). A program for numerical classification. *Computer Journal* 13(1), 63–69.
- Boulton, D. M. and C. S. Wallace (1973). An information measure for hierarchical classification. *Computer Journal* 16(3), 254–261.
- Bright, W. (1992). *International encyclopaedia of linguistics*. New York: Oxford University Press.
- Bynon, T. (1983). *Historical linguistics*. Cambridge: Cambridge University Press.
- Chaitin, G. J. (1966). On the length of programs for computing finite binary sequences. *Journal of the ACM* 13, 547–569.
- Chang, K. (1974). Ancient Chinese phonology and the *ch'ieh yün*. *The Tsing Hua Journal of Chinese Studies* 10, 61–82. New series.
- Chao, Y.-R. (1930). A system of tone letters. *Le Maître Phonétique* 45, 24–27.
- Cheeseman, P. (1993). On finding the most probable model. In J. Shragar and P. Langley (Eds.), *Computational models of scientific discovery and theory formation*, pp. 73–95. San Mateo, California: Morgan Kaufmann.
- Cheeseman, P. and R. Kanefsky (1990). Evolutionary tree reconstruction. In *The theory and application of minimal-length encoding*, AAAI 90 Spring symposium series, pp. 35–39. Stanford, California.

- Chen, M. (1977). The time dimension: Contribution toward a theory of sound change. In W. S.-Y. Wang (Ed.), *The lexicon in phonological change*, pp. 197–251. The Hague: Mouton.
- Chen, M. Y. (1976). From Middle Chinese to Modern Peking. *Journal of Chinese Linguistics* 4(2/3), 113–277.
- Chen, M. Y. and J. Newman (1984a). From Middle Chinese to Modern Cantonese: Part I. *Journal of Chinese Linguistics* 12(1), 148–194.
- Chen, M. Y. and J. Newman (1984b). From Middle Chinese to Modern Cantonese: Part II. *Journal of Chinese Linguistics* 12(2), 334–388.
- Chen, M. Y. and J. Newman (1985). From Middle Chinese to Modern Cantonese: Part III. *Journal of Chinese Linguistics* 13(1), 122–170.
- Cheng, C.-C. (1982). A quantification of Chinese dialect affinity. *Studies in the Linguistic Sciences* 12(1), 29–47.
- Cheng, C.-C. (1994a). DOC: Its birth and life. In M. Y. Chen and O. J. L. Tzeng (Eds.), *In honor of William S-Y Wang: Interdisciplinary studies on language and language change*, pp. 71–86. Taiwan: Pyramid Press.
- Cheng, C.-C. and W. S.-Y. Wang (1977). Tone change in Chao-zhou Chinese: A study in lexical diffusion. In W. S.-Y. Wang (Ed.), *The lexicon in phonological change*, pp. 86–100. The Hague: Mouton.
- Cheng, R. L. (1994b). Lexical diffusion in syntactic change: The case of Taiwanese aspect and phrase markers. In M. Y. Chen and O. J. L. Tzeng (Eds.), *In honor of William S-Y Wang: Interdisciplinary studies on language and language change*, pp. 87–103. Taiwan: Pyramid Press.
- Chomsky, N. and M. Halle (1968). *The sound pattern of English*. New York: Harper and Row.
- Chrétien, C. D. (1962). The mathematical models of glottochronology. *Language* 32, 11–37.
- Cihui (1964). *Hanyu Fangyan Cihui, (A lexicon of Han Dialects)*. Beijing: Wenzhi Gaige Chubanshe. Compilation by Beijing University.
- Cleary, J. G., S. Legg, and I. Witten (1996). An MDL estimate of the significance of rules. In D. L. Dowe, K. B. Korb, and J. J. Oliver (Eds.), *ISIS: Information, Statistics and Induction in Science*, pp. 43–53. Melbourne: World Scientific.
- Clelland, C. H. (1995). Assessment of candidate PFSA models induced from symbol datasets. Tech Report TR-C95/02, Deakin University, Melbourne, Victoria.
- Clelland, C. H. and D. A. Newlands (1994). PFSA modelling of behavioural sequences by evolutionary programming. In R. J. Stonier and X. H. Yu (Eds.), *Proceedings of the conference on complex systems: Mechanism for Adaptation*, pp. 165–172. Rockhampton, Queensland: IOS Press.
- Conklin, D. and I. H. Witten (1994). Complexity based induction. *Machine Learning* 16(3), 203–226.

- Cook, D. J. and L. B. Holder (1994). Substructure discovery using Minimum Description Length and background knowledge. *Journal of Artificial Intelligence Research* 1, 231–255.
- Crowley, T. (1987). *An introduction to historical linguistics*. University of Papua New Guinea Press.
- Crowley, T., J. Lynch, J. Siegel, and J. Piau (1995). *The design of language*. Auckland: Longman Paul.
- de Saussure F. (1964). *Cours de linguistique générale, reprint*. Paris: Payot.
- Djoko, S. (1994). Substructure discovery using the Minimum Description Length principle and background knowledge. In *Proceedings of the twelfth international conference on artificial intelligence*, pp. 1442. Cambridge, MA.: AAAI Press/MIT Press.
- Dobson, A. J. and P. Black (1979). Multidimensional scaling of some lexicostatistical data. *Mathematical Scientist* 4, 55–61.
- Dyen, I. (1971). The validity of the mathematical model of glottochronology. In I. Dyen (Ed.), *Proceedings of the Yale conference: Lexicostatistics in genetic linguistics*, pp. 11–29. The Hague: Mouton.
- Dyen, I., J. B. Kruskal, and P. Black (1992). An Indo-European classification: A lexicostatistical experiment. *Transactions of the American Philosophical Society* 82(Part 5), 1–132.
- Einstein, A. (1956). Physics and reality. In *Ideas and opinions*. London: Alvin Redman.
- Ellegård, A. (1959). Statistical measurement of linguistic relationship. *Language* 35, 131–156.
- Embleton, S. M. (1981). *Incorporating borrowing rates in lexicostatistical tree reconstruction*. PhD dissertation, University of Toronto.
- Embleton, S. M. (1986). *Statistics in historical linguistics*. Bochum, Germany: Brockmeyer Verlag. Also published in *Quantitative Linguistics*, vol.30.
- Embleton, S. M. (1991). Mathematical methods of genetic classification. In S. L. Lamb and E. D. Mitchell (Eds.), *Sprung from some common source*, pp. 365–388. Stanford, California: Stanford University Press.
- Evans, T. G. (1971). Grammatical inference techniques in pattern analysis. In J. T. Tou (Ed.), *Software engineering, COINS III*, Volume 2, pp. 183–202.
- Feldman, J. A. (1972). Some decidability results in grammatical inference. *Information and Control* 20, 244–262.
- Feldman, J. A., J. Gips, J. Horning, and S. Reder (1969). Grammatical complexity and inference. Tech report CS125, Stanford University, Computer Science Department, Stanford, California.
- Fromkin, V. and R. Rodman (1983). *An introduction to language* (3rd ed.). New York: Holt, Rinehart and Winston.

- Gaines, B. R. (1971). Axioms for adaptive behaviour. *International Journal of Man-Machine Studies* 4, 169–199.
- Gaines, B. R. (1976). Behaviour structure transformations under uncertainty. *International Journal of Man-Machine Studies* 8, 337–365.
- Gaines, B. R. (1978). Maryanski's grammatical inferencer. *IEEE Transactions on Computers C-28*, 62–64.
- Gallager, R. G. (1968). *Information theory and reliable communication*. New York: Wiley.
- Garcia, P. and E. Vidal (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 920–925.
- Georgeff, M. P. and C. S. Wallace (1984). A general selection criterion for inductive inference. In T. O'Shea (Ed.), *ECAI-84: Advances in artificial intelligence*, pp. 473–481. Dordrecht: Elsevier, North Holland.
- Gleason, Jr, H. A. (1959). Counting and calculating for historical reconstruction. *Anthropological Linguistics* 2, 22–32.
- Gold, E. M. (1967). Language identification in the limit. *Information and Control* 10, 447–474.
- Gold, E. M. (1972). System identification via state characterization. *Automatica* 8, 621–636.
- Gold, E. M. (1978). Complexity of automaton identification from given data. *Information and Control* 37, 302–320.
- Goldie, C. M. and R. G. E. Pinch (1991). *Communication theory*. Cambridge: Cambridge University Press.
- Good, I. J. (1963). Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables. *Annals of Mathematical Statistics* 34, 911–934.
- Greenberg, J. H. (1987). *Language in the Americas*. Stanford, California: Stanford University Press.
- Greenberg, J. H. (1991). Some problems of Indo-European in historical perspective. In S. L. Lamb and E. D. Mitchell (Eds.), *Sprung from some common source*, pp. 125–140. Stanford, California: Stanford University Press.
- Grimes, J. E. and F. B. Agard (1959). Linguistic divergence in Romance. *Language* 35, 598–604.
- Gudchinsky, S. (1956). The ABC's of lexicostatistics (glottochronology). *Word* 12, 175–210.
- Hamming, R. W. (1980). *Coding and information theory* (2nd ed.). Englewood Cliffs, NJ.: Prentice-Hall.
- Hanson, N. R. (1958). *Patterns of discovery*. Cambridge: Cambridge University Press.

- Harms, R. T. (1990). Synchronic rules and diachronic "laws": The Saussurean dichotomy reaffirmed. In E. C. Polomé (Ed.), *Trends in linguistics: Studies and monographs: 48*, pp. 313–322. Berlin: Mouton de Gruyter.
- Hashimoto, M. (1965). *Phonology of Ancient Chinese*. PhD dissertation, Ohio State University. Reprinted as nos. 10 and 11 of *Study of languages and cultures of Asia and Africa monograph series*, Tokyo, 1978 and 1979.
- Hashimoto, O.-K. (1972). *Phonology of Cantonese*. Cambridge: Cambridge University Press.
- Henrici, A. (1973). Numerical classification of Bantu languages. *African Language Studies 14*, 82–104.
- Hobson, A. (1971). *Concepts in statistical mechanics*. New York: Gordon and Breach.
- Hock, H. H. (1986). *Principles of historical linguistics*. Berlin: Mouton de Gruyter.
- Hoijer, H. (1956). Lexicostatistics: A critique. *Language 32*, 49–60.
- Hsieh, H.-I. (1973). A new method of dialect subgrouping. *Journal of Chinese Linguistics 1*(1), 63–92. Reprinted in William S-Y Wang (Ed.), *The lexicon in phonological change*, pp.159–196. The Hague: Mouton, 1977.
- Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. *Proceedings of IRE 40*, 1098–1101.
- Hymes, D. E. (1971). Lexicostatistics and glottochronology in the nineteenth century. In I. Dyen (Ed.), *Proceedings of the Yale conference: Lexicostatistics in genetic linguistics*, pp. 122–176. The Hague: Mouton.
- Hymes, D. H. (1960). Lexicostatistics so far. *Current Anthropology 1*(1), 3–44.
- Itoga, S. Y. (1981). A new heuristic for inferring regular grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence 3*, 191–197.
- Jeffreys, H. (1946). An invariant form for the prior probability in estimations problems. *Proceedings of the Royal Society of London, Series A 186*, 453–461.
- Jianing, Z. (1990). Fojiao chuanru yu dengyuntu de xingqi, (The entrance of Buddhism into China and the emergence of rhyme-tables). In *Guoji foxue yanjiu niankan*, (*International yearbook of Buddhist studies*), pp. 252–263. Taipei.
- Johnson, R. W. (1979a). Axiomatic characterization of the directed divergences and their linear combinations. *IEEE Transactions on Information Theory IT-13*(2), 709–716.
- Johnson, R. W. (1979b). Comments on 'Prior probability and uncertainty'. *IEEE Transactions on Information Theory IT-25*, 129–132.
- Karlgren, B. *Grammatica serica; script and phonetics in Chinese and Sino-Japanese*. New York: Paragon book gallery. Reprinted from the bulletin of the Museum of Far Eastern Antiquities, Stockholm, no. 12, 1940, 1–471.

- Karlgren, B. (1954). Compendium of phonetics in ancient and archaic Chinese. *Bulletin of the Museum of Far Eastern Antiquities* 26, 211–367.
- Kashyap, R. L. (1971). Prior probability and uncertainty. *IEEE Transactions on Information Theory* IT-17, 641–650.
- Kippen, J. and B. Bel (1989). Can a computer help resolve the problem of ethnographic description? *Anthropological Quarterly* 62(3), 131–144.
- Kirkpatrick, S., C. D. Gelatt, Jr, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science* 220(4598), 671–680.
- Kolmogorov, A. N. (1965). Three approaches for defining the concept of information quantity. *Problems of Information Transmission* 1, 1–7.
- Krachtovil, P. (1968). *The Chinese language today: Features of an emerging standard*. London: Hutchinson.
- Kroeber, A. L. and C. D. Chretien (1937). Quantitative classification of Indo-European languages. *Language* 13, 83–103.
- Kudo, M. and M. Shimbo (1988). Efficient regular grammatical inference techniques by the use of partial similarities and their logical relationships. *Pattern Recognition* 21, 401–409.
- Kullback, S. (1959). *Information theory and statistics*. New York: Wiley.
- Kullback, S. (1967). A lower bound for discrimination information in terms of variation. *IEEE Transactions on Information Theory* IT-13, 126–127.
- Kullback, S. and R. A. Leibler (1951). On information and sufficiency. *Annals of Mathematical Statistics* 22, 79–86.
- Lees, R. B. (1953). The basis of glottochronology. *Language* 29, 113–127.
- Lewis, P. M. (1959). Approximating probability distributions to reduce storage requirements. *Information and Control* 2, 214–225.
- Li, R. (1952). *Qie-yun Yin-xi (The phonological system of Qie-yun)*. Beijing: Kexue Chubanshe.
- Lin, J. (1991). Divergence measures based on Shannon entropy. *IEEE Transactions on Information Theory* IT-37(1), 145–151.
- Malmqvist, G. (1968). Chou tsu-mo on the *ch'ieh-yün*. *Bulletin of the Museum of Far Eastern Antiquities* 40, 33–78. Translation of Zumo (1963).
- Maryanski, F. J. and T. L. Booth (1977). Inference of finite-state probabilistic grammars. *IEEE Transactions on Computers* C-26(6), 521–536.
- Miclet, L. (1980). Regular inference with a tail-clustering method. *IEEE Transactions on Systems, Man and Cybernetics* 10, 737–743.
- Miller, R. A. (1975). The Far East. *Current Trends in Linguistics* 13, 1214–1264.
- Milosavljevic, A., D. Haussler, and J. Jurka (1990). Clustering of macromolecular sequences by minimal length encoding. In *The theory and application of minimal-length encoding*, AAAI 90 Spring symposium series, pp. 45–49. Stanford, California.

- Muggleton, S., A. Srinivasan, and M. Bain (1991). MDL codes for non-monotonic learning. Research Memorandum TIRM-91-049, Turing Institute.
- Neil, J. R. and K. B. Korb (1996). The MML evolution of classification graphs. In D. L. Dowe, K. B. Korb, and J. J. Oliver (Eds.), *ISIS: Information, Statistics and Induction in Science*, pp. 78-89. Melbourne: World Scientific.
- Nerode, A. (1958). Linear automaton transformations. *Proceedings of the American Mathematical Society* 9, 541-544.
- Newman, J. (1987). The evolution of a Cantonese phonotactic constraint. *Australian Journal of Linguistics* 7, 43-72.
- Newman, J. (1994). The history of Wenzhou *y*. In M. Y. Chen and O. J. L. Tzeng (Eds.), *In honor of William S-Y Wang: Interdisciplinary studies on language and language change*, pp. 333-348. Taiwan: Pyramid Press.
- Newman, J. (1996). Footnotes to a history of Cantonese: Accounting for phonological irregularities. In M. Durie and M. Ross (Eds.), *The comparative method reviewed*, pp. 90-111. New York: Oxford University Press.
- Norman, J. (1988). *Chinese*. Cambridge: Cambridge University Press.
- Oliveira, A. L. and A. S. Vincentelli (1995). Inferring reduced ordered decision graphs of Minimal Description Length. In A. Prieditis and S. Russell (Eds.), *Proceedings of the 12th International conference in Machine Learning*, pp. 421-429. Lake Tahoe, CA: Morgan Kaufmann.
- Oliveira, A. L. and A. S. Vincentelli (1996). Using the Minimum Description Length principle to infer reduced ordered decision graphs. *Machine Learning* 25, 23-50.
- Oliver, J. J. (1993). Decision graphs – an extension of decision trees. Tech report TR-92/173, Monash University, Computer Science Department, Clayton, Victoria.
- Oliver, J. J. and R. A. Baxter (1994). MML and Bayesianism: Similarities and differences. Tech Report TR-94/206, Monash University, Computer Science Department, Clayton, Victoria.
- Oliver, J. J. and D. Hand (1994). Introduction to minimum encoding inference. Tech report TR-94/205, Monash University, Computer Science Department, Clayton, Victoria.
- Oncina, J. and P. Garcia (1992). Inferring regular languages in polynomial updated time. In *Pattern recognition and image analysis: Selected papers from the fourth Spanish symposium*, pp. 49-61. Singapore: World Scientific.
- Patrick, J. D. (1978). *An information measure comparative analysis of megalithic geometries*. PhD dissertation, Monash University, Computer Science Department, Melbourne.

- Patrick, J. D. (1982). Stone circle geometries: An information theory approach. In D. C. Heggie (Ed.), *Archaeoastronomy in the Old World*, pp. 231–264. Cambridge University Press.
- Patrick, J. D. (1991). SNOB: A program for discriminating between classes. Tech Report TR-91/151, Monash University, Computer Science Department, Clayton, Victoria.
- Patrick, J. D. (1996). A Minimum Message Length (MML) model for software measures. In D. L. Dowe, K. B. Korb, and J. J. Oliver (Eds.), *ISIS: Information, Statistics and Induction in Science*, pp. 385–394. Melbourne: World Scientific.
- Patrick, J. D. and K. E. Chong (1987). Real-time inductive inference for analysing human behaviour. In *Proceedings of the Australian joint AI conference*, Sydney, pp. 305–322.
- Pfahring, B. (1995). *Practical uses of the Minimum Description Length principle in inductive learning*. PhD dissertation, Vienna University of Technology, Vienna.
- Popper, K. (1969). *Conjectures and refutations*. London: Routledge and Kegan Paul.
- Popper, K. (1972). *Objective knowledge*. Oxford: Oxford University Press.
- Pulleyblank, E. G. (1970). Late Middle Chinese, part 1. *Asia Major* 15, 197–239.
- Pulleyblank, E. G. (1984). *Middle Chinese: A study in historical phonology*. Vancouver: University of British Columbia Press.
- Putnam, H. (1975). Probability and confirmation. In *Mathematics, Matter and Method*. Cambridge: Cambridge University Press.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, California: Morgan Kaufmann.
- Quinlan, J. R. and R. L. Rivest (1989). Inferring decision trees using the Minimum Description Length principle. *Information and Computation* 80, 227–248.
- Rabin, M. O. and D. Scott (1959). Finite automata and their decision problems. *IBM journal of research and development* 3(2), 115–125.
- Raman, A. V., J. Newman, and J. D. Patrick (1997). A complexity measure for diachronic Chinese phonology. Tech Report 3/97, Massey University Information Systems Department, Palmerston North, New Zealand.
- Raman, A. V. and J. D. Patrick (1997a). Beam search and simba search for PFSA inference. Tech Report 2/97, Massey University Information Systems Department, Palmerston North, New Zealand.
- Raman, A. V. and J. D. Patrick (1997b). A heuristic using cross-entropy for sk-strings. Tech Report 1/97, Massey University Information Systems Department, Palmerston North, New Zealand.

- Raman, A. V. and J. D. Patrick (1997c). Linguistic similarity measures using the Minimum Message Length principle. In R. D. Blench and M. Spriggs (Eds.), *Archaeology and language I: Theoretical and methodological orientations*, pp. 260–277. London: Routledge. Paper also read at the World Archaeological Congress-3 conference, New Delhi, December 1994.
- Raman, A. V. and J. D. Patrick (1997d). The sk-strings method for inferring PFSA. In *Proceedings of the workshop on automata induction, grammatical inference and language acquisition at the 14th international conference on machine learning — ICML'97*, Nashville, Tennessee, pp. (in press).
- Rao, C. R. and T. K. Nayak (1985). Cross-entropy, dissimilarity measures, and characterizations of quadratic entropy. *IEEE Transactions on Information Theory IT-31*(5), 589–593.
- Rea, J. A. (1990). Lexicostatistics. In E. C. Polomé (Ed.), *Trends in linguistics: Studies and monographs: 48*, pp. 217–222. Berlin: Mouton de Gruyter.
- Rich, E. and K. Knight (1991). *Artificial intelligence* (2nd ed.). New York: McGraw-Hill.
- Richetin, M. and F. Vernadet (1984). Regular inference for syntactic pattern recognition: A case study. In *Proceedings of the 7th International conference on Pattern Recognition*, pp. 1370–1372.
- Ringe, Jr, D. A. (1992). On calculating the factor of chance in language comparison. *Transactions of the American Philosophical Society 82*(Part 1), 1–110.
- Riordan, J. (1958). *An introduction to combinatorial analysis*. New York: Wiley.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica 14*, 465–471.
- Rissanen, J. (1983). A universal prior for integers and estimation by Minimum Description Length. *Annals of Statistics 11*, 416–431.
- Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society, Series B 49*(3), 223–239, (disc.) 252–265.
- Robinson, O. W. (1977). Rule reordering and lexical diffusion. In W. S.-Y. Wang (Ed.), *The lexicon in phonological change*, pp. 69–85. The Hague: Mouton.
- Ross, A. S. C. (1950). Philological probability problems. *Journal of the Royal Statistical Society, Series B 12*, 19–59.
- Ruhlen, M. (1991). The Amerind phylum and the prehistory of the New World. In S. L. Lamb and E. D. Mitchell (Eds.), *Sprung from some common source*, pp. 328–352. Stanford, California: Stanford University Press.
- Sankoff, D. (1972). Mathematical developments in lexicostatistic theory. *Current Trends in Linguistics 11*, 93–113.

- Sapir, E. (1916). *Time perspectives in aboriginal American culture: A study in method*. Anthropological series, Memoir 90, Number 19. Ottawa: Geological survey of Canada.
- Shannon, C. E. and W. Weaver (1963). *The mathematical theory of communication*. Urbana: University of Illinois Press.
- Sheinvald, J., B. Dom, and W. Niblack (1990). Detection of useless features by information theoretic criteria. In *The theory and application of minimal-length encoding*, AAAI 90 Spring symposium series, pp. 30–39. Stanford, California.
- Shevoroshkin, V. and A. M. Ramer (1991). Some recent work on the remote relations of languages. In S. L. Lamb and E. D. Mitchell (Eds.), *Sprung from some common source*, pp. 178–199. Stanford, California: Stanford University Press.
- Shore, J. E. (1979). Minimum cross-entropy spectral analysis. NRL Memo Rep. 3921, Naval Research Lab.
- Shore, J. E. and R. W. Johnson (1980). Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory IT-26*, 26–37.
- Solomonoff, R. J. (1964a). A formal theory of inductive inference: Part 1. *Information and Control* 7, 1–22.
- Solomonoff, R. J. (1964b). A formal theory of inductive inference: Part 2. *Information and Control* 7, 224–254.
- Streeter, M. L. (1977). Doc, 1971: A Chinese dialect dictionary on computer. In W. S.-Y. Wang (Ed.), *The lexicon in phonological change*, pp. 101–119. The Hague: Mouton.
- Swadesh, M. (1951). Diffusional cumulation and archaic residue as historical explanations. *Southwestern Journal of Anthropology* 7, 1–21.
- Swadesh, M. (1952). Lexicostatistic dating of prehistoric ethnic contacts. *Transactions of the American Philosophical Society* 96, 452–463.
- Thieme, P. (1964). The comparative method for reconstruction in linguistics. In D. Hymes (Ed.), *Language in culture and society; a reader in linguistics and anthropology*, pp. 585–598. New York: Harper and Row.
- Thomason, M. G. and E. Granum (1986). Dynamic programming inference of Markov networks from finite sets of samples strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 491–501.
- Thomason, M. G., E. Granum, and R. Blake (1986). Experiments in dynamic programming inference of Markov networks with strings representing speech data. *Pattern Recognition* 19, 342–352.
- Toussaint, G. T. (1975). Sharper lower bounds for discrimination information in terms of variation. *IEEE Transactions on Information Theory IT-21*(1), 99–100.

- Trombetti, A. (1905). *L'unità d' origine del linguaggio*. Bologna: Treves/Beltrami. Re-issued 1962, Bologna: *Civitas Dei*.
- Vajda, I. (1970). Note on discrimination information and variation. *IEEE Transactions on Information Theory IT-16*, 771–773.
- Vernadet, F. and M. Richetin (1984). Efficient regular grammatical inference for pattern recognition. *Pattern Recognition 17*, 245–250.
- Voyles, J. B. (1992). *Early Germanic grammar*. San Diego: Academic Press.
- Wallace, C. S. (1986). An improved program for classification. In *Proceedings of the 9th Australian Computer Science Conference: ACSC-9*, Canberra, pp. 357–366.
- Wallace, C. S. (1990). Classification by Minimum-Message-Length inference. In S. G. Akl et al. (Eds.), *Advances in computing and information — ICCI'90*, Volume LNCS 468, pp. 72–81. Niagara Falls: Springer-Verlag.
- Wallace, C. S. and D. M. Boulton (1968). An information measure for classification. *Computer Journal 11*, 185–194.
- Wallace, C. S. and D. M. Boulton (1969). The information content of a multistate distribution. *Journal of Theoretical Biology 23*, 269–278.
- Wallace, C. S. and D. L. Dowe (1993). MML estimation of the von Mises concentration parameter. Tech report TR-93/193, Monash University, Computer Science Department, Clayton, Victoria.
- Wallace, C. S. and D. L. Dowe (1994). Intrinsic classification by MML - the Snob program. In *Proceedings of the 7th Australian joint conference on Artificial Intelligence*, Armidale, NSW, pp. 37–44. World Scientific.
- Wallace, C. S. and P. R. Freeman (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series B 49*(3), 240–265.
- Wallace, C. S. and P. R. Freeman (1992). Single factor analysis by MML estimation. *Journal of the Royal Statistical Society, Series B 54*(1), 195–209.
- Wallace, C. S. and M. P. Georgeff (1984). A general objective for inductive inference. Tech Report TR-84/44, Monash University, Computer Science Department, Clayton, Victoria.
- Wallace, C. S., K. B. Korb, and H. Dai (1996). Causal discovery via MML. In L. Saitta (Ed.), *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, pp. 516–524.
- Wallace, C. S. and J. D. Patrick (1993). Coding decision trees. *Machine Learning 11*, 7–22.
- Wang, W. S.-Y. (1969). Competing changes as a cause of residue. *Language 45*, 9–25.
- Wang, W. S.-Y. and R. E. Asher (1994). Chinese linguistic tradition. In R. E. Asher (Ed.), *The encyclopaedia of language and linguistics*, pp. 524–527. Oxford: Pergamon Press.

- Wang, W. S.-Y. and C.-C. Cheng (1977). Implementation of phonological change: The Shuang-feng Chinese case. In W. S.-Y. Wang (Ed.), *The lexicon in phonological change*, pp. 148–158. The Hague: Mouton.
- Wen, H. (1995). The first step towards phonological analysis in Chinese. *Journal of Chinese Linguistics* 23(1), 137–158.
- Witten, I. H., M. N. Radford, and J. G. Cleary (1987). Arithmetic coding for data compression. *Communications of the ACM* 30(6), 520–538.
- Yiming, W. (1990). Fanqie shuo lue, (Brief remarks on Fanqie). *Wenshi Zhishi* 6, 88–91.
- Zihui (1962). *Hanyu Fangyin Zihui*, (*A Pronouncing Dictionary of Han Dialects*). Beijing: Wenzhi Gaige Chubanshe. Compilation by Beijing University.
- Zumo, Z. (1963). Qieyun de xingzhi he tade yinxi jichu. In *Yuyanxue luncong*, pp. 434–473. Beijing: Zhonghua Shuju. Revised edition pub. 1966. See also transl. (Malmqvist 1968).

ERRATA

p.xiii, para 5, line 2: Change "...the derivations for..." to "...my derivations for..."

p.24, section 2.4, line 2: Change "...we see that the length of a message to describe hypothesis..." to "...we find that the length of a message to describe the hypothesis..."

p.25, table 3.4: Change Table header for column 3 from "*OHG" to "*LNWG" and column 4 from "OF" to "OHG".

p.27, line 9: Change "...if we assume that the complexity of a language..." to "...if we assume that the complexity of a language and by implication that of its diachronic phonology..."

p.47, section 4.8, line 2/3: Change "The derivations were prepared by hand..." to "Each derivation was individually prepared by the author..."

p.56, last para, line 1: Change "...using our earlier scheme..." to "...using our initial approach..."

p.57, footnote, line 3/4: Change "(commissions)" to "(omissions)" and vice-versa.

p.60, line 15/16: Change "...Wallace and Boulton (1969)..." to "...Wallace and Boulton (1968, 1969)..."

p.60, line 17: Change "But Wallace and Freeman..." to "Wallace and Freeman..."

p.79, last line: Change "...little symbol repetition." to "...little symbol repetition within each string."

p.80, line 11: Change "...function of the the..." to "...function of the..."

p.81, section 6.4, para 2 and 3: Delete subscript "_c" on " Q_c , δ_c , q_{0c} and F_c " wherever they occur.

p.84, line 5: Change "...generating that string." to "...accepting that string."

p.85, algorithm 6.5.1, line 6: Change "*merge(p, q);*" to "*merge(p, q); renumber;*"

p.87, section 6.6.1, last sentence: Add the following footnote: "It has been pointed out that the standard best first, branch-and-bound algorithm with an admissible heuristic will be able to identify the strings in descending order of their probability. It should thus be possible to precisely construct the list of strings that we need. However, it is not unlikely that the simpler heuristic outlined in method B performs almost as well since it puts an upper bound on the number of strings that we need to consider and only ignores very improbable strings. The implementation of the method described in this footnote is, however, reserved for future work."

p.93, para 1: last sentence: Add the following footnote: "The graphs plotting the error in the inferred automaton may seem a bit confusing as it is somewhat unclear what the error term measures. Since the original automaton may have a lot of redundancy in it, inference of more compact automata will result in negative errors. We suggest that the error term be interpreted as a measure of badness of the inferred automaton with lower being better. Thus an algorithm may be construed as better than another if the plot tracing its error term is *below* the other's."

p.115, section 8.1, line 5: Change "...are the beam search method..." to "...are a variation of the beam search method..."

p.121, line 1: Add the following footnote at end of sentence: "It has been suggested that this stochastic acceptance strategy is too harsh and seems guaranteed to produce worse performance. In hindsight, we agree. A better strategy would be to shift out a random member of the beam when a candidate needs to be accepted stochastically. This is to be investigated further."

p.123, section 8.4, line 1: Add the following footnote at end of sentence: "Given that the simba search algorithm has a significant stochastic component to it, one would expect it to be run several times and the results averaged. However, this was not done in the present study as its performance was found to be poor and because

each run of the program took quite a bit of time. The possibility of evaluating the modified simba search algorithm thus, however, is definitely worth investigating and is marked as potential future work.”

p.140. last sentence of para 1: Add the following footnote: “From the computational point of view, the best way to account for this situation would be to consider all possible orderings of the sequences as equally likely and model the phonology accordingly. This, however, has not been done in this project and is left as a topic to be addressed in future work.”

p.143, last para: line 3: Change “...between DNA strands.” to “...between DNA strands. See for example work by Allison, Wallace and Yee (1990a, 1990b, 1992) and Allison and Yee (1990).”