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.
Symmetric Parallel
Class Expression Learning

TRAN, Cong An
School of Engineering and Advanced Technology
Massey University
New Zealand

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

June, 2013
For my grandparents, my parents, my wife and my son
Acknowledgements

I would like to take this opportunity to thank my supervisors, Prof. Hans W. Guesgen, Prof. Jens Dietrich and Prof. Stephen Marsland for their guidance and support. I am thankful that I could benefit from their combined research experience. Prof. Hans W. Guesgen introduced me the world of Ambient Intelligent with its humanistic ideas of helping the elderly to have a better life. Prof. Jens Dietrich attracted me to his colourful world of Software Engineering. His endless source of ideas impressed me that this is the world of the active and creative people. Prof. Stephen Marsland, a ‘diverse’ professor with the excellent background in mathematics and computer science, encouraged me into his mysterious world of Machine Learning.

I also want to thank Jens Lehmann, the research group leader of Machine Learning and Ontology Engineering (MOLE) Group at the University of Leipzig, as well as the author of the DL-Learner framework, for his kindness to grant me permission to access his DL-Learner framework repository as well as for his valuable advice to evaluate learners. I also highly appreciate Prof. Adrian Paschke, Director of RuleML Inc., for his advice on my research direction.

I would like to thank the Ministry of Education and Training of Vietnam for awarding me the scholarship for my study. Thanks also go to the School of Engineering and Advanced Technology (SEAT) for the financial support that has enabled me to attend international conferences, which not only improved my research but also enriched my life.

Thanks to all members of Massey University Smart Home (MUSE) research group for their valuable discussions and feedbacks on my research. I also
want to thank Michele Wagner for her administrative support throughout my degree.

This thesis would not be possible without the warm and support of my Palmy-based family, Mr. Vo The Truyen and his family, Mr. Nguyen Buu Huan and Mr. Nguyen Van Long, who give me a home away from home. In particular, my special thanks go to Mr. Truyen’s for treating me as their little brother and Mr. Huan for checking and giving me many helpful advices on my writing. I could not have asked for more warm and kind friends as them.

Finally, I would like to give all my deepest gratitude and respect to my family. To my grandparents and my parents for their continuous love, support and patience. To my parents-in-law for taking good care of my son in more than four years, which released me from family concerns to concentrate on my research. To my sister and brother for being with my parents during my study. To my beloved wife, Nguyen Thu Huong, and my son, Tran Cong Huan. It is my fortune to have them in my life. They are always by my side to share all the laughers and tears.
Abstract

The growth of both size and complexity of learning problems in description logic applications, such as the Semantic Web, requires fast and scalable description logic learning algorithms. This thesis proposes such algorithms using several related approaches and compares them with existing algorithms. Particular measures used for comparison include computation time and accuracy on a range of learning problems of different sizes and complexities.

The first step is to use parallelisation inspired by the map-reduce framework. The top-down learning approach, coupled with an implicit divide-and-conquer strategy, also facilitates the discovery of solutions for a certain class of complex learning problems. A reduction step aggregates the partial solutions and also provides additional flexibility to customise learnt results.

A symmetric class expression learning algorithm produces separate definitions of positive (true) examples and negative (false) examples (which can be computed in parallel). By treating these two sets of definitions ‘symmetrically’, it is sometimes possible to reduce the size of the search space significantly. The use of negative example denotations enhances learning problems with exceptions, where the negative examples (‘exceptions’) follow a few relatively simple patterns.

In general, correctness (true positives) and completeness (true negatives) of a learning algorithm are traded off against each other because these two criteria are normally conflicting. Particular learning algorithms have an inherent bias towards either correctness or completeness. The use of negative definitions enables an approach (called fortification in this thesis) to improve predictive correctness by applying an appropriate level of over-specialisation to the prediction model, while avoiding over-fitting.

The experiments presented in the thesis show that these algorithms have the potential to improve both the computation time and predictive accuracy of description logic learning when compared to existing algorithms.
## Contents

Acknowledgement .................................................. v

Abstract ................................................................................ vii

List of Figures ........................................................................ xiii

List of Tables .............................................................................. xv

1 Introduction ................................................................. 1

1.1 Introduction ........................................................................... 1

1.2 Motivation ............................................................................... 3

1.3 Scope of Study ......................................................................... 8

1.4 Aims and Objectives .............................................................. 8

1.5 Thesis Overview ................................................................. 10

2 Preliminaries and Related Work .......................................... 13

2.1 Description Logics and Web Ontology Language ............. 13

2.1.1 Description logic languages ............................................. 13

2.1.2 Description logic knowledge bases ................................. 17

2.1.3 The Web Ontology Language (OWL) ......................... 23

2.2 Description Logic and OWL Learning ............................... 26

2.2.1 Description logic learning problem .............................. 26

2.2.2 Basic approaches in DL learning ................................. 29

2.3 Related Work ........................................................................ 32

2.3.1 Description logic learning ............................................. 32
## CONTENTS

2.3.2 Parallel description logic learning .................................................. 34
2.3.3 Numerical data learning in description logics .................................. 35
2.3.4 Class Expression Learner for Ontology Engineering (CELOE) ........... 36

### 3 Evaluation Methodology

3.1 Introduction ................................................. 39
3.2 Evaluation Metrics ........................................... 40
    3.2.1 Accuracy ........................................... 40
    3.2.2 Learning time ....................................... 41
    3.2.3 Definition length .................................... 42
    3.2.4 Search space size ................................... 42
3.3 Experimental Design .......................................... 42
    3.3.1 Experimental framework ................................ 42
    3.3.2 Comparison algorithms ................................ 48
    3.3.3 Evaluation Datasets ................................... 50
3.4 Implementation and Running the Code .............................................. 58

### 4 Adaptive Numerical Data Segmentation

4.1 Introduction ............................................... 61
4.2 Motivation ................................................. 62
4.3 Description of Our Method ...................................... 65
4.4 The Algorithms ............................................. 67
4.5 Evaluation Results .......................................... 68
    4.5.1 Segmentation result ................................... 70
    4.5.2 Experimental results on the accuracy ................... 71
4.6 Conclusion .................................................. 75

### 5 Parallel Class Expression Learning

5.1 Parallelisation for Class Expression Learning ............................... 77
5.2 Description of Our Method ...................................... 79
5.3 The Algorithms ............................................. 83
5.4 Evaluation Result ............................................ 88
5.4.1 Experiment 1 - Comparison between ParCEL and CELOE 89
5.4.2 Experiment 2 - Effect of parallelisation on learning speed 96
5.4.3 Experiment 3 - Definition reduction strategy 98
5.5 Conclusion 100

6 Symmetric Class Expression Learning 103
6.1 Exceptions in Learning 103
6.2 Symmetric Class Expression Learning 106
   6.2.1 Overview of our method 106
   6.2.2 Description of our method 108
   6.2.3 The algorithm 111
   6.2.4 Counter-partial definitions combination strategies 116
6.3 Evaluation 118
   6.3.1 Experiment 1 - Combination strategies comparison 119
   6.3.2 Experiment 2 - Search tree size comparison 122
   6.3.3 Experiment 3 - Predictive accuracy and learning time 124
   6.3.4 Experiment 4 - The learnt definitions 128
6.4 Conclusion 132

7 Improving Predictive Correctness by Fortification 135
7.1 Problem Description 135
7.2 Fortification Candidates Generation 140
7.3 Fortification Strategy 143
   7.3.1 Fortification candidates scoring 143
      7.3.1.1 Training coverage scoring 144
      7.3.1.2 Fortification concept similarity scoring 145
      7.3.1.3 Fortification validation scoring 157
      7.3.1.4 Random score 160
   7.3.2 Fortification cut-off point computation 160
7.4 Evaluation 161
   7.4.1 Fortification evaluation methodology 161
   7.4.2 Experimental results 162
CONTENTS

7.5 Conclusion .................................. 176

8 Conclusions and Future Work 181
  8.1 Discussion and Contributions of the Thesis ................ 181
  8.2 Threats to Validity of the Results ....................... 184
    8.2.1 Threats to internal validity ....................... 184
    8.2.2 Threats to external validity ....................... 186
  8.3 Future Work .................................. 187

A Accessing the Implementation 191
  A.1 Software Structure .................................. 191
    A.1.1 DL-Learner architecture ......................... 191
    A.1.2 Our algorithms packages ......................... 192
  A.2 Checking Out and Compiling Code ....................... 193
    A.2.1 Checking out the project ......................... 194
    A.2.2 Compiling code ................................ 195

B Reproducing the Experimental Results 197
  B.1 System Requirements ................................ 197
  B.2 Running the Experiments ................................ 198
    B.2.1 Syntax ........................................ 198
    B.2.2 Learning configuration file naming conventions .... 199
  B.3 Learning Configuration ................................ 199
  B.4 Test Cases ....................................... 202

C List of Publications 211

Glossary 213

References 217
# List of Figures

1.1 A typical search tree produced by a top-down learning approach for the *Tweety* problem. ................................................................. 6

2.1 An example of the top-down approach in DL learning .................. 31
2.2 An example of the bottom-up approach in DL learning .................. 32

3.1 Data partition for a 10-fold cross-validation .............................. 44
3.2 An example of the inconsistency in parallel learning ................... 45
3.3 Termination of learning algorithms ............................................. 48
3.4 The MUSE dataset ontology visualised in Protege .......................... 55
3.5 RDF graph of an activity in the MUSE dataset ............................. 56
3.6 The concept hierarchy of the MUBus dataset visualised in Protege .... 57

4.1 Specialisation of numeric datatype properties ............................. 63
4.2 Segmentation of numeric datatype properties ............................. 64
4.3 Inappropriate segmentation of the data property values prevents the spe-
   cialisation of an overly general expression ................................ 64
4.4 A relation graph between examples and numeric values .................. 66
4.5 Segmentation of data property values ........................................ 67
4.6 Segmentation of the data property values in Figure 4.3 .................. 67

5.1 The specialisation using a downward refinement operator ............... 81
5.2 Parallel exploration of the search tree using two workers .............. 83
5.3 Reducer-Worker interaction. .................................................... 84
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Accuracy against learning time of CELOE and ParCEL on the Carcino-Genesis dataset using different number of workers.</td>
<td>96</td>
</tr>
<tr>
<td>5.5</td>
<td>Accuracy against learning time of CELOE and ParCEL on the UCA1 dataset using different number of workers.</td>
<td>97</td>
</tr>
<tr>
<td>5.6</td>
<td>Speed-up efficiency of ParCEL on the UCA1 dataset.</td>
<td>98</td>
</tr>
<tr>
<td>6.1</td>
<td>Exception patterns in learning</td>
<td>105</td>
</tr>
<tr>
<td>6.2</td>
<td>Different approaches to learning the definition for the extended Tweety learning problem.</td>
<td>107</td>
</tr>
<tr>
<td>6.3</td>
<td>The top-down learning step aims to find both partial definitions and counter-partial definitions</td>
<td>111</td>
</tr>
<tr>
<td>6.4</td>
<td>Top-down learning in SPaCEL with multiple workers.</td>
<td>112</td>
</tr>
<tr>
<td>7.1</td>
<td>The creation of a prediction model for a learning problem</td>
<td>137</td>
</tr>
<tr>
<td>7.2</td>
<td>A prediction scenario of the prediction model in Figure 7.1</td>
<td>138</td>
</tr>
<tr>
<td>7.3</td>
<td>Decrease of predictive accuracy caused by inappropriate fortification.</td>
<td>139</td>
</tr>
<tr>
<td>7.4</td>
<td>Fortification candidates learning</td>
<td>141</td>
</tr>
<tr>
<td>7.5</td>
<td>Family concepts hierarchy.</td>
<td>150</td>
</tr>
<tr>
<td>A.1</td>
<td>DL-Learner architecture</td>
<td>193</td>
</tr>
<tr>
<td>B.1</td>
<td>Check required softwares in the system including JDK, Subversion and Maven.</td>
<td>206</td>
</tr>
<tr>
<td>B.2</td>
<td>Install required softwares for compilation the project: JDK, Subversion and Maven</td>
<td>206</td>
</tr>
<tr>
<td>B.3</td>
<td>Check out the project from the repository</td>
<td>207</td>
</tr>
<tr>
<td>B.4</td>
<td>Compile the DL-Leaner and ParCEL core components project.</td>
<td>207</td>
</tr>
<tr>
<td>B.5</td>
<td>Compile the interface project.</td>
<td>207</td>
</tr>
<tr>
<td>B.6</td>
<td>Compile the ParCEL CLI project.</td>
<td>207</td>
</tr>
<tr>
<td>B.7</td>
<td>Command line to learn the Forte-Uncle dataset using CELOE.</td>
<td>208</td>
</tr>
<tr>
<td>B.8</td>
<td>Command line to learn the Forte-Uncle dataset using ParCEL.</td>
<td>208</td>
</tr>
<tr>
<td>B.9</td>
<td>Command line to learn the Forte-Uncle dataset using SPaCEL</td>
<td>209</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Basic concept constructors in description logics ............................. 15
2.2 Letters used to name description logic languages. ............................ 16
2.3 The semantics of basic concept constructors in description logics. .......... 18
2.4 DLs notations and OWL notations .............................................. 24
2.5 OWL constructors and the corresponding constructors in DLs ............... 25

3.1 Size and complexity of the evaluation datasets ............................... 51
3.2 Properties of the evaluation datasets .......................................... 52

4.1 Reduction of numeric data properties values resulting from the adaptive
segmentation strategy. .............................................................. 70
4.2 Training and predictive accuracies of CELOE on the ILDP dataset using
for the two segmentation strategies ........................................... 72
4.3 Training and predictive accuracies of ParCEL on the ILDP dataset using
for the two segmentation strategies ........................................... 73
4.4 Predictive accuracy of CELOE and ParCEL on the UCA1 dataset using
for the two segmentation strategies ........................................... 74

5.1 ParCEL and CELOE experimental results. ..................................... 90
5.2 Balanced accuracy of CELOE and ParCEL on unbalanced datasets .......... 94
5.3 ParCEL and CELOE experimental results with one worker ..................... 95
5.4 Speed-up of ParCEL on the UCA1 dataset .................................. 98
5.5 Definition length comparison between three reduction strategies ............ 99

6.1 Basic description learning algorithms and their usage of examples .......... 108
LIST OF TABLES

6.2 SPaCEL experimental result – Combination strategies .......................... 119
6.3 SPaCEL experimental results – The search tree size ............................. 123
6.4 SPaCEL experimental result – Learning time and predictive accuracy .... 124
6.5 Balanced predictive accuracy of unbalanced datasets .......................... 127
6.6 SPaCEL experimental result – Definition length of the learning problem 128

7.1 Fortification experimental result with CELOE ........................................ 166
7.2 Fortification experimental result on ParCEL .......................................... 170
7.3 Fortification experimental result with SPaCEL ...................................... 173
7.4 Experimental results for new cut-off points ........................................ 178

B.1 Common components in DL-Learner framework and their parameters .... 200
B.2 Actions and expected result of the test case ....................................... 204