ARIZONA STATE UNIVERSITY
GENERAL STUDIES COURSE PROPOSAL COVER FORM

Course information:
Copy and paste current course information from Class Search/Course Catalog.

<table>
<thead>
<tr>
<th>Academic Unit</th>
<th>Historical, Philosophical and Religious Studies</th>
<th>Department</th>
<th>Philosophy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>PHI</td>
<td>Number 319</td>
<td>Title Philosophy, Computing and Artificial Intelligence</td>
</tr>
<tr>
<td>Is this a cross-listed course?</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is this a shared course?</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Course description:
If so, list all academic units offering this course

Requested designation: Mathematical Studies-CS
Note: a separate proposal is required for each designation requested

Eligibility:
Permanent numbered courses must have completed the university's review and approval process.
For the rules governing approval of omnibus courses, contact Phyllis.Lucie@asu.edu or Lauren.Leo@asu.edu.

Submission deadlines dates are as follow:
For Fall 2015 Effective Date: October 9, 2014
For Spring 2016 Effective Date: March 19, 2015

Area(s) proposed course will serve:
A single course may be proposed for more than one core or awareness area. A course may satisfy a core area requirement and more than one awareness area requirements concurrently, but may not satisfy requirements in two core areas simultaneously, even if approved for those areas. With departmental consent, an approved General Studies course may be counted toward both the General Studies requirement and the major program of study.

Checklists for general studies designations:
Complete and attach the appropriate checklist
- Literacy and Critical Inquiry core courses (L)
- Mathematics core courses (MA)
- Computer/statistics/quantitative applications core courses (CS)
- Humanities, Arts and Design core courses (HU)
- Social-Behavioral Sciences core courses (SB)
- Natural Sciences core courses (SO/SC)
- Cultural Diversity in the United States courses (C)
- Global Awareness courses (G)
- Historical Awareness courses (H)

A complete proposal should include:
☒ Signed General Studies Program Course Proposal Cover Form
☒ Criteria Checklist for the area
☒ Course Catalog description
☒ Course Syllabus
☒ Copy of Table of Contents from the textbook and list of required readings/books

Respectfully request that proposals are submitted electronically with all files compiled into one PDF. If necessary, a hard copy of the proposal will be accepted.

Contact information:

Name Cindy Baade
Phone 5-7183

Mail code 4302
E-mail: cynthia.baade@asu.edu

Department Chair/Director approval: (Required)

Chair/Director name (Typed): Matthew J. Garcia
Date: 2/8/15

Chair/Director (Signature):

Rev. 1/94, 4/95, 7/98, 4/00, 1/02, 10/08, 11/11/12/11, 7/12, 5/14
Arizona State University Criteria Checklist for

MATHEMATICAL STUDIES [CS]

Rationale and Objectives

The Mathematical Studies requirement is intended to ensure that students have skill in basic mathematics, can use mathematical analysis in their chosen fields, and can understand how computers can make mathematical analysis more powerful and efficient. The Mathematical Studies requirement is completed by satisfying both the Mathematics [MA] requirement and the Computer/Statistics/Quantitative Applications [CS] requirement explained below.

The Mathematics [MA] requirement, which ensures the acquisition of essential skill in basic mathematics, requires the student to complete a course in College Mathematics, College Algebra, or Pre-calculus; or demonstrate a higher level of skill by completing a mathematics course for which a course in the above three categories is a prerequisite.

The Computer/Statistics/Quantitative Applications [CS] requirement, which ensures skill in real world problem solving and analysis, requires the student to complete a course that uses some combination of computers, statistics, and/or mathematics.* Computer usage is encouraged but not required in statistics and quantitative applications courses. At a minimum, such courses should include multiple demonstrations of how computers can be used to perform the analyses more efficiently.

*CS does not stand for computer science in this context; the "S" stands for statistics. Courses in computer science must meet the criteria stated for CS courses.

Revised April 2014
Proposer: Please complete the following section and attach appropriate documentation.

**ASU--[CS] CRITERIA**

A COMPUTER/STATISTICS/QUANTITATIVE APPLICATIONS [CS] COURSE MUST SATISFY ONE OF THE FOLLOWING CRITERIA: 1, 2, OR 3

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
</table>

1. **Computer applications**: courses must satisfy both a and b:
   
   a. Course involves the use of computer programming languages or software programs for quantitative analysis, algorithmic design, modeling, simulation, animation, or statistics.
   
   b. Course requires students to analyze and implement procedures that are applicable to at least one of the following problem domains (check those applicable):
      
      i. Spreadsheet analysis, systems analysis and design, and decision support systems.
      
      ii. Graphic/artistic design using computers.
      
      iii. Music design using computer software.
      
      iv. Modeling, making extensive use of computer simulation.
      
      v. Statistics studies stressing the use of computer software.
      
      vi. Algorithmic design and computational thinking.

   *The computer applications requirement cannot be satisfied by a course, the content of which is restricted primarily to word processing or report preparation skills, the study of the social impact of computers, or methodologies to select software packages for specific applications. Courses that emphasize the use of a computer software package are acceptable only if students are required to understand, at an appropriate level, the theoretical principles embodied in the operation of the software and are required to construct, test, and implement procedures that use the software to accomplish tasks in the applicable problem domains. Courses that involve the learning of a computer programming language are acceptable only if they also include a substantial introduction to applications to one of the listed problem domains.*
<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>2. Statistical applications:</strong> courses must satisfy a, b, and c.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a. Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Pre-calculus, or a course already approved as satisfying the MA requirement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b. The course must be focused principally on developing knowledge in statistical inference and include coverage of all of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i. Design of a statistical study.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. Summarization and interpretation of data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. Methods of sampling.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iv. Standard probability models.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v. Statistical estimation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vi. Hypothesis testing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vii. Regression or correlation analysis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c. The course must include multiple demonstrations of how computers can be used to perform statistical analysis more efficiently, if use of computers to carry out the analysis is not required.</td>
</tr>
</tbody>
</table>
### 3. Quantitative applications: courses must satisfy a, b, and c.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Pre-calculus, or a course already approved as satisfying the MA requirement.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b. The course must be focused principally on the use of mathematical models in quantitative analysis and decision making. Examples of such models are:

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

i. Linear programming.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ii. Goal programming.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

iii. Integer programming.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

iv. Inventory models.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

v. Decision theory.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

vi. Simulation and Monte Carlo methods.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

vii. Other (explanation must be attached).

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

c. The course must include multiple demonstrations of how computers can be used to perform the above applications more efficiently, if use of computers is not required by students.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
<th>Identify Documentation Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Prefix</td>
<td>Number</td>
<td>Title</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>PHI</td>
<td>319</td>
<td>Philosophy, Computing, and Artificial Intelligence</td>
</tr>
</tbody>
</table>

Explain in detail which student activities correspond to the specific designation criteria. Please use the following organizer to explain how the criteria are being met.

<table>
<thead>
<tr>
<th>Criteria (from checksheet)</th>
<th>How course meets spirit (contextualize specific examples in next column)</th>
<th>Please provide detailed evidence of how course meets criteria (i.e., where in syllabus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>This course involves the use of Prolog and logic programming to model certain aspects of intelligence within the context of an observation-thought-decision-action cycle</td>
<td>Units 1-7</td>
</tr>
<tr>
<td>1bvi</td>
<td>This course requires students to analyze and implement procedures applicable to algorithmic design and computational thinking. Students analyze and implement procedures for observation, forward reasoning, abduction, maintenance goal triggering, backward reasoning, achievement goal reduction, plan selection, and prohibition. Students analyze the differences between classical negation and negation-as-failure as well as the differences between conclusive and defeasible reasoning more generally. Students also analyze and implement procedures for lexicon building, recognizing grammatical sentences, model building, and determining the truth value of sentences relative to models.</td>
<td>Units 1-7</td>
</tr>
</tbody>
</table>
PHI 319 Philosophy, Computing and Artificial Intelligence
Philosophical problems surrounding artificial intelligence (AI). Thinking as computation and AI; the ethics, epistemology and metaphysics of computing.
Allow multiple enrollments: No Primary course component: Lecture
Repeatable for credit: No Grading method: Standard Grading
Offered by:
New College of Interdisciplinary Arts and Sciences -- School of Humanities, Arts, and Cultural Studies
Pre-requisites: ENG 102, 105 or 108 with C or better; Minimum 25 hours
College of Letters and Sciences -- College of Letters and Sciences
Pre-requisites: ENG 102, 105 or 108 with C or better; Minimum 25 hours
College of Liberal Arts and Sciences -- Historical, Philosophical & Religious Studies, Sch
Pre-requisites: ENG 102, 105 or 108 with C or better; Minimum 25 hours
Philosophy, Computing, and Artificial Intelligence

PHI 319 (Spring 2015, Session C, 15 Weeks; T,Th 12:00-1:15 Tempe ECG-G227)

Thomas A. Blackson
School of Historical, Philosophical, and Religious Studies
Arizona State University

Course Description
This course is a study of the use of certain techniques to model intelligence. The techniques consist primarily in the application of logic within the context of an observation-thought-decision-action cycle. A prior course in symbolic logic (PHI 333 or equivalent) is helpful but not required.

At the completion of this course, students will be familiar with the general idea of computation and the idea that thinking is a form of computation, will understand basic aspects of logic programming and its underlying theory, will be familiar with the application of logic programming and some of its variants within the context the agent model of intelligence, will understand basic aspects of the computer language Prolog (“PROgramming in LOGic”), will be familiar with basic aspects of natural language processing, will know some of the experiments in psychology that challenge the use of logic to model human intelligence, and will be familiar with some of the challenges to the logical approach to modeling intelligence generally.

This course satisfies CS (computer/statistics/quantitative applications) in the University Undergraduate General Studies Requirement. This course also satisfies a requirement for the Symbolic Systems Certificate. This certificate is modeled on the program at Stanford, the course of study taken by many influential figures in technology, such as Marissa Mayer (President and CEO of Yahoo!).

Required Books
There are two required books for the course: Robert Kowalski’s Computational Logic and Human Thinking: How to be Artificially Intelligent (Cambridge University Press, 2011) and Hector Levesque’s Thinking as Computation: A First Course (MIT Press, 2012).

In addition, the authors have made public some of their teaching materials: slides for Thinking as Computation, slides for a shorter and a longer course based on Computational Logic and Human
Thinking, and video lectures (which were recorded at the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona 2011) for Computational Logic and Human Thinking. These teaching materials are strictly supplementary to the books and my lecture notes.

Grade for the Course

The final grade for the course is a function of your grade on 7 assignments. Each assignment (listed below) is worth 14 out of a total of 100 points. There are 2 free points. There is no extra credit. Attendance is not required, but passing the course is unlikely without regular class attendance. Keep your graded assignments. They are your only record of your grades. Incompletes are given only to accommodate serious illnesses and family emergencies, which must be adequately documented.

The final grade for the course uses plus-minus letter grades, A+ to E. This grade is a weighted averaged computed using ASU's numerical value for the letter grades (A+ = 4.3, A = 4.0, A- = 3.7, B+ = 3.3, B = 3.00, B- = 2.67, C+ = 2.33, C = 2.00, D = 1.00, E = 0).

Here is an example to illustrate how the final grade is computed:

Assignment #1, A- 14%(3.7) = .518
Assignment #2, B+ 14%(3.3) = .462
Assignment #3, B 14%(3.0) = .42
Assignment #4, A 14%(4.0) = .56
Assignment #5, B 14%(3.0) = .42
Assignment #6, B 14%(3.0) = .42
Assignment #7, B 14%(3.0) = .42
Free points, A+ 2%(4.3) = .086

In the example, the weighted average sums to 3.306. Relative to ASU's numerical values for letter grades, 3.306 is closest to B+ (= 3.3). So, in this example, the final grade for the course is B+.

Lectures and Readings

Not all the following material is equally important. I highlight the most important points in my lectures and in my lecture notes. This (in addition to the reading in the books) is the material on which you will be graded in your assignments. Use the slides and video as supplementary. It is not necessary to understand every detail of the supplementary material. Some of this material is difficult and appropriate for a more advanced course. The lectures and lecture notes are more understandable than the books, and the books are more understandable than the slides and the video lectures.
UNIT 1: 
Thinking is Computation (The Hypothesis in the Course)
- *Thinking as Computation 1, 2; slides 1-38.
- *Computational Logic and Human Thinking "Introduction," 1; shorter 1-10; video 00:00-12:40; longer 1-33.

Logic and Logic Programming (The Technical Background)
- *Thinking as Computation 2; slides 39-55.

Assignment #1

UNIT 2
Prolog (A Computer Programming Language)

The Psychology of Logic (The Wason Selection Task and The Suppression Task)
- *Computational Logic and Human Thinking 2; longer 61-74.

Assignment #2

UNIT 3
The Fox and the Crow (The Logic Programming/Agent Model)
- *Computational Logic and Human Thinking 3, 8; longer 96-108, 157-188.

Assignment #3

UNIT 4
Negation as Failure (The Suppression Task Revisited)

Assignment #4
UNIT 5: Prohibitions and Prospective Logic Programming
  • *Computational Logic and Human Thinking* 12.

Abduction and Abductive Logic Programming
  • *Thinking as Computation* 11; slides 314-322.
  • *Computational Logic and Human Thinking* 10, "Appendix A6"; longer 215-228

Assignment #5

UNIT 6: Understanding Natural Language
  • *Thinking as Computation* 8; slides 188-228.

Assignment #6

UNIT 7: The Wason Selection Task Revisited
  • *Computational Logic and Human Thinking* 16.

Assignment #7

Contact Information
Thomas A. Blackson
Philosophy Faculty
School of Historical, Philosophical, and Religious Studies
Lattie E. Coor Hall, room 3356
PO Box 874302
Arizona State University
Tempe, AZ. 85287-4302
blackson@asu.edu, tombblackson.com, www.public.asu/~blackson
Thinking as Computation

A First Course

Hector J. Levesque

The MIT Press
Cambridge, Massachusetts
London, England
Contents

Preface

Background ......................................................... xiii
Overview of the book ................................................ xvi
Guide for the course instructor .............................. xvii
Guide for the student ............................................... xix

Acknowledgments .................................................... xxi

1 Thinking and Computation ........................................... 1
  1.1 Thinking ......................................................... 2
     1.1.1 What is thinking? ......................................... 3
  1.2 Computation ..................................................... 4
     1.2.1 Symbols and symbolic structures ...................... 4
     1.2.2 What is computation? .................................... 5
     1.2.3 Some arithmetic procedures ............................ 5
     1.2.4 The lesson .................................................. 10
  1.3 Thinking as computation ...................................... 11
     1.3.1 Leibniz and his idea ...................................... 12
     1.3.2 Propositions vs. sentences .............................. 13
     1.3.3 Using what is known: The web of belief ............. 16
Want to read more? ................................................ 20
Exercise ............................................................... 21

2 A Procedure for Thinking .......................................... 23
  2.1 Atomic and conditional sentences ....................... 23
  2.2 Logical entailment .......................................... 24
  2.3 Back-chaining ............................................... 27
     2.3.1 Using variables ........................................... 27
     2.3.2 Tracing the back-chaining .............................. 28
  2.4 Variables in queries ........................................... 31
     2.4.1 One complication: Renaming variables ............... 31
2.4.2 Another complication: Backtracking .................................................. 33
2.4.3 A more complex query ........................................................................ 34
2.5 Why is back-chaining good? .................................................................... 37
  2.5.1 Getting stuck in a loop ........................................................................ 38
Want to read more? ................................................................................. 39
Exercises ..................................................................................................... 41

3 The Prolog Language .................................................................................. 42
  3.1 Prolog programs ...................................................................................... 46
  3.2 Prolog queries ........................................................................................ 48
    3.2.1 Queries and their outcomes .............................................................. 49
    3.2.2 Conjunctive queries ........................................................................ 50
    3.2.3 Negation in queries ......................................................................... 50
    3.2.4 Tracing the back-chaining ............................................................... 52
    3.2.5 Instantiated and uninstantiated variables ........................................ 53
    3.2.6 Equality in queries ......................................................................... 53
  3.3 Prolog back-chaining ............................................................................. 56
    3.3.1 Unification ....................................................................................... 58
    3.3.2 Renaming variables .......................................................................... 59
    * 3.3.3 Back-chaining revisited ................................................................. 61
Want to read more? ...................................................................................... 61
Exercises ..................................................................................................... 63

4 Writing Prolog Programs ............................................................................. 63
  4.1 The truth in Prolog ................................................................................ 63
    4.1.1 The truth, and nothing but ............................................................. 63
    4.1.2 The whole truth .............................................................................. 64
  4.2 A blocks world ....................................................................................... 66
  4.3 Recursion in Prolog ............................................................................... 68
  * 4.4 Mathematical induction ..................................................................... 69
  4.5 Nonterminating programs ..................................................................... 72
  4.6 A more complex predicate ..................................................................... 75
    * 4.6.1 Recursion and termination, reconsidered ...................................... 75
  4.7 Efficiency in Prolog ............................................................................... 78
Want to read more? ...................................................................................... 81
Exercises ..................................................................................................... 82
7.2 Writing programs that use lists ........................................ 140
7.2.1 Some example list predicates ...................................... 141
7.3 Using the member and append predicates ......................... 145
7.3.1 The blocks world revisited ....................................... 149
Want to read more? ......................................................... 150
Exercises ........................................................................ 150

8 Case Study: Understanding Natural Language ..................... 153
8.1 Analyzing the syntax of a language ................................ 154
8.1.1 Lexicon .................................................................. 155
8.1.2 Grammar .................................................................. 156
8.1.3 Parsing and ambiguity ............................................. 157
8.2 Interpreting noun phrases ............................................. 158
8.2.1 Writing a world model ............................................. 159
8.2.2 Writing a lexicon ................................................... 159
8.2.3 Writing a parser ..................................................... 163
8.2.4 Putting the pieces together ..................................... 164
8.3 Interpreting sentences .................................................. 170
8.3.1 Yes/no questions ..................................................... 170
8.3.2 Dynamic predicates in Prolog ................................ 172
8.3.3 Simple declarative sentences ................................ 173
8.4 Nonreferential noun phrases ........................................ 174
Want to read more? ......................................................... 175
Exercises ........................................................................ 176

9 Case Study: Planning Courses of Action .......................... 179
9.1 Planning problems ....................................................... 180
9.1.1 A first example: The three coins .............................. 180
9.1.2 A second example: The monkey and bananas .......... 182
9.1.3 States and operators ................................................. 182
9.2 Generating plans ......................................................... 183
9.2.1 A general planning program .................................. 183
9.2.2 Solving the three-coins problem .............................. 184
9.2.3 Atoms as terms in Prolog ....................................... 185
9.2.4 Solving the monkey and bananas problem ............ 186
9.2.5 Bounding plan length ............................................ 187
9.2.6 A third example: The 15-puzzle .............................. 190
9.3 Scaling up: The search problem ........................................ 191
   9.3.1 Knowledge-based planning ........................................ 192
   * 9.3.2 Best-first search ................................................. 195
9.4 Scaling up: The representation problem .............................. 197
   9.4.1 Situations and fluents .......................................... 198
   9.4.2 Planning with situations and fluents ............................ 202
   9.4.3 Why is this representation useful? ............................ 203
   9.4.4 Other kinds of actions .......................................... 203
Want to read more? .................................................................. 204
Exercises ............................................................................. 205

10 Case Study: Playing Strategic Games .................................. 209
   10.1 Games as problems ...................................................... 210
      10.1.1 How a game can be defined .................................. 211
      10.1.2 A first example: Race to 21 ................................. 211
   10.2 Finding the best move .................................................. 212
      10.2.1 Game trees ......................................................... 213
      10.2.2 A general game player ........................................ 215
      10.2.3 Playing Race to 21 ............................................. 217
      10.2.4 A second example: Tic-tac-toe .............................. 218
      * 10.2.5 Playing an entire game .................................... 220
      * 10.2.6 A third example: Boxes .................................... 222
   10.3 Playing bigger games .................................................... 226
      10.3.1 Numerical game trees and minimax ........................ 227
      10.3.2 Alpha and beta cutoffs ....................................... 231
      10.3.3 The application to chess ..................................... 232
Want to read more? .................................................................. 233
Exercises ............................................................................. 234

* 11 Case Study: Other Ways of Thinking ............................... 239
   11.1 Back-chaining as subject matter .................................. 241
      11.1.1 A breadth-first thinking procedure ....................... 244
      * 11.1.2 A forward-chaining thinking procedure ............. 245
      11.1.3 Back-chaining with variables, negation, and equality 247
   11.2 Explanation .............................................................. 249
      11.2.1 Diagnosis ......................................................... 249
      11.2.2 A general explanation program ......................... 251
Contents

11.3 Learning ................................................................. 253
  11.3.1 Inducing general rules ........................................ 254
  11.3.2 An example of induction ...................................... 254
  * 11.3.3 Classification: Training and testing .................... 256
11.4 Propositional reasoning ............................................. 258
  11.4.1 Conjunctive normal form .................................... 258
  11.4.2 Satisfiability .................................................. 259
  11.4.3 Computing satisfiability ..................................... 260
  11.4.4 Logical entailment reconsidered ............................ 261
  * 11.4.5 First-order reasoning ...................................... 262
Want to read more? ..................................................... 265

12 Can Computers Really Think? ......................................... 267
  12.1 What computers can do .......................................... 268
  12.2 The Turing Test .................................................. 270
  12.3 The Chinese Room ................................................ 271
  12.4 The Summation Room ............................................. 272
  12.5 A final word ...................................................... 273
Want to read more? ..................................................... 274

Appendix A Some Computer Basics .................................... 275
  A.1 Working with computer files .................................... 275
  A.2 Files available online ......................................... 276

Appendix B Getting Started with SWI-Prolog ......................... 279
  B.1 Installing SWI-Prolog ......................................... 279
  B.2 How to load SWI-Prolog programs ............................. 280

Appendix C Getting Your Prolog Programs to Work .................. 283
  C.1 Getting program files to load ................................ 283
  C.2 Getting the right answers from queries ..................... 283
    C.2.1 Tracing the execution ................................. 283
    C.2.2 Interrupting the execution ............................. 284
  C.3 Saving a record of program execution ....................... 284

Appendix D Other Prolog Systems .................................... 285
References .................................................................. 29
Index of Technical Terms .............................................. 29
Computational Logic
and Human Thinking
How to be Artificially Intelligent

ROBERT KOWALSKI
Imperial College London
## Contents

*Preface*  
*Summary and plan of the book*  

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Introduction</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Logic on the Underground</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>The psychology of logic</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>The fox and the crow</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>Search</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Negation as failure</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>How to become a British Citizen</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>The louse and the Mars explorer</td>
<td>77</td>
</tr>
<tr>
<td>8</td>
<td>Maintenance goals as the driving force of life</td>
<td>92</td>
</tr>
<tr>
<td>9</td>
<td>The meaning of life</td>
<td>107</td>
</tr>
<tr>
<td>10</td>
<td>Abduction</td>
<td>125</td>
</tr>
<tr>
<td>11</td>
<td>The Prisoner's Dilemma</td>
<td>134</td>
</tr>
<tr>
<td>12</td>
<td>Motivations matter</td>
<td>144</td>
</tr>
<tr>
<td>13</td>
<td>The changing world</td>
<td>155</td>
</tr>
<tr>
<td>14</td>
<td>Logic and objects</td>
<td>166</td>
</tr>
<tr>
<td>15</td>
<td>Biconditionals</td>
<td>179</td>
</tr>
<tr>
<td>16</td>
<td>Computational Logic and the selection task</td>
<td>188</td>
</tr>
<tr>
<td>17</td>
<td>Meta-logic</td>
<td>198</td>
</tr>
<tr>
<td></td>
<td>Conclusions of the book</td>
<td>213</td>
</tr>
<tr>
<td></td>
<td><strong>A1</strong> The syntax of logical form</td>
<td>227</td>
</tr>
<tr>
<td></td>
<td><strong>A2</strong> Truth</td>
<td>231</td>
</tr>
<tr>
<td></td>
<td><strong>A3</strong> Forward and backward reasoning</td>
<td>247</td>
</tr>
<tr>
<td></td>
<td><strong>A4</strong> Forward and backward reasoning</td>
<td>257</td>
</tr>
</tbody>
</table>
Contents

A4 Minimal models and negation 263
A5 The resolution rule 269
A6 The logic of abductive logic programming 280
References 296
Index 303