Solution Manual

Artificial Intelligence:
Structures and Strategies for Complex Problem Solving

Sixth Edition

George F. Luger

For further lecturer material, please visit:
www.pearsoned.co.uk/luger

ISBN 0 321 54591 5

Contents

Section I

Philosophy, Sample Course Descriptions, and Examinations

SECTION I.1

Our Philosophy

As researchers in the area of artificial intelligence and practitioners in the design of expert systems and many other AI applications, we saw a need for an advanced introduction to the discipline. In creating "Artificial Intelligence: Structures and Strategies for Complex Problem Solving" we had three goals in mind:

1. To present AI technology along with its deep roots in the philosophical, mathematical, and computational traditions. AI as currently practiced is very much both part and product of the western scientific evolution.

2. To offer a broad focus on all AI, the European tradition as well as the American, Lisp language-oriented as well as Prolog, symbol-based, connectionist, and stochastic. A good programmer must be aware of all her tools.

3. Finally, we wished to base AI algorithms and techniques in their rightful place within modern computer science. Much of modern computing is a product of earlier research in AI (recursive data structures, object-based design, semantics of programming languages, and so on). Modern AI practice requires a strong foundation and grounding in traditional computing.

We intended that there be sufficient material in this book for several semesters of study. In the first semester, the foundational material is fairly clear, namely, the first 9 chapters of the book. We present all our introductory algorithms in both Lisp, Prolog, and Java in the supplementary materials; but we have found that, for an introductory quarter or semester, time permits only one language to be covered. At the University of New Mexico our CS majors have all had Lisp/Scheme in their introductory language courses, so in the 400 level AI course we teach only Prolog, and still give programming assignments in both Prolog and another language such as Lisp or Java. At other universities, of course, other options may well be more appropriate.

A second semester course in AI will of necessity be more eclectic. We prefer to cover different topics each time the advanced course is offered. We also feel an advanced course should require students to read and comment on AI research papers, and whenever we offer the advanced AI course, we collect, distribute, and require reading and analysis of 8 or 10 such papers.

In the next section we present a number of curriculum plans. First is a description of an introductory AI course, we call it an "Introduction to Artificial Intelligence". The course is divided into three sections, the first and last with evaluation through an examination, the middle section requiring the student to write a set of programs. After the course description we include two sample examinations, for the first and last thirds of the course. We also describe a typical programming assignment.

SECTION I.2

Sample Course Description: An Introduction to Artificial Intelligence

Textbook (GL), for reference purposes in the following descriptions:

Artificial Intelligence: Structures and Strategies for Complex Problem Solving By George F. Luger Addison-Wesley Pearson, 2009

Week 1: Artificial Intelligence, its roots and scope (GL, ch. 1, Intro Part II)

- AI, an attempted definition
- Historical foundations
- Overview of application areas
- An introduction to representation and search

Weeks 2 & 3: The Predicate Calculus (GL, ch. 2)

- Representation languages
- The propositional calculus and its semantics
- The predicate calculus: syntax & semantics
- Inference: soundness, completeness
- The unification algorithm

Weeks 3 & 4: Structures and strategies for state space search (GL, ch. 3)

- Quick review of graphs
- State space search
- Data-driven and goal-driven search
- Breadth-first, depth-first, and depth-first iterative deepening search

Weeks 4: Heuristic search (GL, ch. 4).

- Priority queues
- A*
- Iterative deepening A*
- Beam search
- Two-person games
- Mini-Max and alpha-beta

Week 5: Stochastic Methods (GL, ch. 5)

- Quick review of counting principles

- Elements of probability
- Applications of the stochastic technology
- Bayes' theorem and its use

Week 6: Architectures for AI problem solving (GL, ch. 6)
- Recursive specification for queues, stacks, and priority queues
- The production system
- The blackboard

Weeks 7 & 8: PROLOG (Part II of AI Algorithms, Data Structures, and Idioms)

- The PROLOG environment
- Relational specifications and rule based constraints
- Abstract data types in PROLOG
- Graph search with the production system
- A PROLOG planner

Week 9: Introduction to AI representational schemes (GL, ch. 7)
- Issues in knowledge representation
- Semantic networks
- Conceptual dependencies
- Frames, scripts, and object systems
- The hybrid design: objects with rule sets

Week 10: Rule-based, case-based, and model-based systems (GL, ch. 8)
- Production system based search
- Rule stacks and the "why" query, proof trees and the "how" query
- Models of inductive reasoning
- The Stanford Certainty Factor algebra
- Knowledge engineering

Weeks 10 & 11: Building expert systems in PROLOG (GL, ch 6, AI Algorithms….)
- Meta-predicates in PROLOG
- The role of a meta-interpreter: PROLOG in PROLOG
- Rule-stacks, proof-trees, and certainty factor algebras in PROLOG
- Exshell, a back-chaining rule interpreter in PROLOG

Week 12: Reasoning in situations of uncertainty (GL, ch. 9)
- Examples of Abductive Inference
- Non-monotonic logic, belief revision
- Certainty factor algebras and fuzzy reasoning
- Stochastic models and Bayesian belief networks

Weeks 13 & 14: Advanced AI applications (GL, select appropriate chapters)

Week 15: Course summary and review (GL, ch. 16)
- The possibility of a science of intelligence
- Limitations and future research

There are two examinations, a mid-term and a final, each one hour long There are three programming assignments:

1. Building graph search algorithms in Prolog

    a) depth-first

    b) breadth-first

    c) best-first search

2. Building graph search algorithms in Lisp

    a) depth-first

    b) breadth-first

    c) best-first search

3. Using EXSHELL to build a rule based expert reasoning system

Course credit: Mid-term and final 40% each, programming assignments 20%. Sometimes a 10-15 page paper is assigned, the AI topic of the student's choice, and then the course credit is each exam 30%, the programs, 30%, the paper 10%.

SECTION I.3

Sample Examinations

Introduction to Artificial Intelligence EXAM Number 1

Name_____________________________

No books or notes. The points for each question and percent of total credit follows the question number. Good luck.

1. (18) Consider the following story: All people that are not poor and are smart are happy. Those people that read are smart. John is wealthy. Helen can read and is wealthy. Happy people have exciting lives. Wealthy people are not poor. Find someone with an exciting life.

   (a) Translate the story into predicate calculus expressions.

   (b) Solve the problem with goal driven reasoning

   (c) Show the solution process with either the iterations of a production system or and/or graph search indicating the unifications and exactly where each is made.

2. (6) Define a production system. How can such a system be used for either data or goal driven problem solving?

3. (6) List three reasons why the production system offers an important "architecture" for computer based problem solving.

4. (8) Give the size, in terms of the branching factor B and the depth of search N, for the open list in each of the searches:

   (a) depth-first

   (b) breadth-first

   (c) best-first search

   (d) What is the size of the closed list in each of these situations?

5. (6) What is depth-first iterative deepening search, and why is it important?

6. (6) Define:

   (a) An A* (A star) algorithm

   (b) Admissibility