Luger, G.F. and Stubblefield, W.A. (1994). Artificial Intelligence Meets Modern Computer Science. In *Proceedings of AAAI-94 Fall Symposium Series*. Palo Alto: AAAI Press.

Artificial Intelligence Meets Modern Computer Science

by

George F. Luger
Department of Computer Science
University of New Mexico
Albuquerque NM

&

William A. Stubblefield
Department Computer Science
Dartmouth College
Hanover NH

abstract

The role of artificial intelligence in modern computer science education has matured in recent years. From a stand-alone discipline, often taught at the university level with little generally accepted structure or form, AI has become an important component of education in modern computer science. This evolution is important from both the computer science and artificial intelligence perspectives.

At the University of New Mexico we have directly addressed these issues. We teach artificial intelligence as a four hundred level course, indicating that it is intended for undergraduates in their final year of education or as an introductory course for graduate computer science. It is also seen as the introductory course for our graduate program emphasis in Intelligent Systems. In the next sections we present the benefits of our approach from both the computer science and artificial intelligence viewpoints. We then summarize our position, showing that many of the questions for discussion in this symposium are addressed by our approach.

AI from the CS Perspective

In the artificial intelligence class computer science students learn advanced representation and computing skills. Already prepared by lower division background skills in discrete mathematics and data structures, CS students are finally able to apply their talents to the design of solutions for complex tasks and to use more sophisticated problem solving tools. The AI topics and applications these students address include:

1. Advanced issues in representation. This is seen from a computer science viewpoint where AI students can understand and use representations in the predicate calculus, production system problem solvers, planners, and object-oriented design.

2. The power of very high level computing languages. For the first time the normal CS student comes in contact with very high level pattern matching computing. In the context of an AI course the advantages of unification coupled with recursive

search algorithms can be clearly demonstrated.

3. The design and use of meta-interpreters. This very important by-product of high level language tools offers important insights for computer science students. We design production rule interpreters, semantic net inheritance algorithms, and interpreters for an object based representation.

4. Other related AI topics. These including the control of complexity issues with heuristics, the design of context free and context dependent parsers and generators for natural language. Also presented are methods for reasoning with uncertainty and an introduction to machine learning. These are important issues INTRODUCED in the context of a first level AI class.

The AI Perspective

Because introductory AI students come from a background of computer science we find that many of the computational skills missing in AI classes a decade ago are already present when they begin the AI course. These skills come primarily from the courses in discrete mathematics and data structures. Our students have already had (at least) two courses using C++ and a general language course where they learned rudimentary skills in Common Lisp programming. Granted, they don't yet have the skills to use very high level languages properly.

The skills they bring to the AI class include.

1. Extensive knowledge of graph theory.

2. Experience with recursive data structures.

3. Language experience, including C++ and Common Lisp.

4. Preliminary knowledge of the predicate calculus.

These skills allow us to begin AI instruction at a fairly advanced level and get rather quickly into interesting topics. We present these more sophisticated AI topics as the "leading edge" of tools for contemporary computing. These are the tools that modern engineers must bring to their problem solving tasks for government, industry, and further education.

The topics covered in an introductory AI class include those tools needed for getting into more current issues in AI. The techniques presented in an introductory class allow for the design of advanced representational schemes and search algorithms that support research areas including theorem proving, natural language understanding, and machine learning. These are the topics of our advanced course offerings in AI. After an introductory course students are also prepared for the design and building of more sophisticated meta-interpreters in high level languages.

Symposium Issues

Our approach to education in AI allows for tentative answers to some of the important issues of this symposium.

1. We have a coherent introduction to the key issues of AI. Far from the " AI smorgasbord" approach, we introduce and interrelate issues in the context of modern topics and tools for computing. In this sense our product is not "disjointed or superficial." As the curriculum presented below indicates, AI topics form a cohesive and organic flow rooted in the supporting concepts of sound computer science.

2. There is no confusion between cognitive science and artificial intelligence. The different and diverging goals of these two disciplines are emphasized from the first class. We have a cognitive science course for those students desiring to acquire this knowledge. Where cognitive science and AI come close, in the use of such tools as the production system for example, these parallels are pointed out.

3. There is an integration of multiple AI "viewpoints." The context of "engineering solutions" actually offers a nice coming together of sometimes disparate AI views. It is only in the presentation of advanced issues in later AI courses that these views are seen as creating the future for research in AI.

An important topic not addressed by this symposium is the role of practice and research in AI within the larger university curriculum. Certainly the task of understanding and building intelligent artifacts relates to the issues of the disciplines of psychology, philosophy (including ethics), and education, to mention but a few. This important issue must be part of any coherent teaching of AI, but as a topic it is left to future AI symposia.

Our ideas for introductory education in AI are exemplified in the brief curriculum description that follows. They may be seen in detail in our book from Benjamin Cummings (1993): *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*.

Curriculum Description: An Introduction to Artificial Intelligence

Textbook (L&S, for reference purposes in the following descriptions):

Artificial Intelligence: Structures and Strategies for Complex Problem Solving
by
George F. Luger &
William A. Stubblefield
Benjamin Cummings, 1993

Week 1: Artificial Intelligence, Its Roots and Scope (L&S, ch1, Intro Pt II)

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

Weeks 2 & 3: The Predicate Calculus (L&S, ch2)

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

Week 4: Structures and Strategies for State Space Search (L&S ch 3)

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

Weeks 5 & 6: Heuristic Search (L&S, ch 4).

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

Weeks 7 & 8: Architectures for AI Problem Solving (L&S, ch5)

- Recursive specification for queues, stacks, and priority queues
- The production system
- The blackboard
- Planning and Triangle Tables

Weeks 9 & 10: PROLOG

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

Weeks 11 & 12: The Rule Based Expert System (L&S, ch 8)

- Production system based search
- Rule stacks and "why" queries, proof trees and "how" queries
- Models of inductive reasoning
- The Stanford Certainty Factor algebra
- Knowledge engineering
- Tieresias, a knowledge based editor

Week 13: Building a Rule Based Expert System in PROLOG (L&S, ch 13)

- 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 14: Introduction of Structured AI Representational Schemes (L&S, ch 9)

- Issues in knowledge representation
- Semantic networks
- Conceptual dependencies
- Frames, scripts, and object systems
- The hybrid design: objects with rule sets

Week 15: Course Summary and Review (L&S, ch 16)

- Video demonstrations of object based and hybrid design
- The possibility of a science of intelligence
- Future research issues and limitations of the AI paradigm

The course includes 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 search
2. Building graph search algorithms in LISP:
   a) depth first
   b) breadth first
   c) best search
3. Using EXSHELL to build a rule based expert reasoning system

Course credit: Mid-term and final 40% each, programming assignments 20%

More detail for teaching this course, including a complete set of lecture notes and programs may be obtained from Benjamin Cummings, 390 Bridge Parkway, Redwood City CA, 94065. Request the Instructor's Manual that accompanies the text. All our programs are available using anonymous ftp:

ftp ftp.cs.unm.edu

see the lisp and prolog subdirectories under ~pub/ai

Introductory AI Curriculum Materials

We have two offerings for the Type III submission. First, a complete set of video tapes for an introductory AI course and second, a complete set of programs in PROLOG and Common Lisp.

1. The set of 32 video lectures. These tapes were developed with professional camera and editing techniques of the Instructional Television Group at the University of New Mexico. They cover in detail the lectures that make up a one quarter/semester course in AI. Complete descriptions of these tapes and what they cover are available for the University of New Mexico (contact the first author: luger@cs.unm.edu).

2. A complete set of programs in Common Lisp and PROLOG are available for both an introductory and advanced course in AI. Here we list the topics for which running programs with exaqmples are available. The code is may be obtained from *ftp* *ftp.cs.unm.edu* in subdirectory *~pub/ai*.

2.1 Implementation of introductory Data Structures: sets, bags, stacks, priority queues, etc.

2.2 Control algorithms for search: depth-first, breadth-first, best-first. We present these search algorithms as "shells" where the user may add the rules for search of a particular problem domain. Several examples are demonstrated.

2.3 A general control strategy for a production system. This is demonstrated with several problem domains.

2.4 A STRIPS-like planner. This is demonstrated in the blocks world.

2.5 A rule based shell, able to process how and why queries. It uses the standard Stanford certainty factor algebra. A sample knowledge base is provided.

2.6 Semantic net and frame-based inheritance search (PROLOG and Lisp.

2.7 The creation of an object-oriented meta-interpreter. This is built, step by step, in both PROLOG and Common Lisp. We feel this is an important tool for seeing the power of very high level languages in building more sophisticated representational schemes.

2.8 The creation of a hybrid environment by attaching rule sets to the object oriented representations of 2.7.

2.9 Demonstration and justification of the main CLOS built-in predicates.

2.10 The demonstration of a number of machine learning algorithms:

a. Version space search, specific to general, general to specific, and candidate elimination (PROLOG only).

b. Explanation based learning (PROLOG)

c. ID3 (Common Lisp)

2.11 The creation and demonstration of context free, context sensitive, and recursive descent semantic net parsers (PROLOG only).

George Luger

AAAI–94 Fall Symposium Series

Improving Instruction of Introductory Artificial Intelligence

Program Committee:
Marti Hearst, Xerox PARC, Chair
Haym Hirsh, Rutgers University
Dan Huttenlocher, Cornell University
Nils Nilsson, Stanford University
Bonnie Webber, University of Pennsylvania

WORKING NOTES

November 4 – 6, 1994

The Monteleone Hotel, New Orleans, Louisiana