An Integrative Framework for Artificial Intelligence Education

Pat Langley

Institute for the Study of Learning and Expertise
Palo Alto, California

Department of Computer Science
University of Auckland, Auckland, NZ

This work was supported by ONR Grants N00014-15-1-2517 and N00014-15-1-2424.

The Crisis in AI Education

Modern students of artificial intelligence are not well served, in that introductory courses:

- Focus on topics that are well formalized and easy to teach
- Train people to consume AI technology, not to produce it
- Highlight statistics and learning, which replace classic topics
- Place little emphasis on constructing intelligent systems

The field is in danger of raising entire generations with little education in its traditional aims or methods.

In this talk, I analyze this crisis in AI instruction and propose a promising remedy.

Drawbacks of Standard AI Courses

- Students learn about AI as a collection of *isolated algorithms*, rather than as elements in integrated systems.
- There is little emphasis on *representing domain content*; e.g., search algorithms are presented in abstract terms.
- Courses seldom convey the field's *cumulative character*, the way complex abilities build on simpler ones.
- Many exercises rely on *software packages* or partial solutions that students treat as black boxes.
- Courses often *omit key topics and ideas* that played important roles historically.

Reasons for the Situation

- Standard textbooks make it easy to teach AI in this way, and inertia-bound instructors are reluctant to change.
- Computer science departments have a strong bias toward abstract analysis at the algorithm level.
- These departments denigrate ideas from cognitive psychology, which is viewed as less respectable.
- AI applications emphasize component abilities separate from other aspects of intelligence.
- Software packages make it easy for students to produce results without learning how to recreate their abilities.

Evaluation of Sample Courses

	I	R	C	P	B
Carnegie Mellon	○	⊙	○	○	⊙
Georgia Tech	○	○	○	○	○
MIT	○	*	○	⊙	⊙
Stanford	○	○	○	○	○
UC Berkeley	○	○	⊙	○	○
UCLA	○	*	○	○	○
U. Maryland	○	⊙	*	*	⊙
USC	○	*	○	○	⊙
UT Austin	*	⊙	⊙	○	⊙
U. Washington	○	○	⊙	○	○

Evaluations - poor (○), medium (⊙), good (●), unknown (*) - on five criteria: integration (I), representation (R), cumulative style (C), programming (P), and breadth (B).

New Principles for AI Instruction

- Champion a systems perspective that shows how mechanisms interact to produce intelligence.
- Give experience with encoding representational content that mechanisms interpret to produce behavior.
- Present topics in a cumulative manner, just as calculus builds on algebra, which builds on arithmetic.
- Teach students not only how to use AI methods, but how to construct them from simpler components.
- Cover important abilities exhibited in human intelligence even when difficult to formalize.

These principles will help counter the common belief that AI is a set of engineering tricks.

An Alternative Organization

Consider a course framework that borrows topics from Langley, Laird, and Rogers' (2009) review of cognitive architectures:

- Recognition and pattern matching
- Decision making and choice
- Conceptual inference and reasoning
- Execution and sequential control
- Planning and problem solving
- Integrated intelligent agents

This organization follows the design principles given earlier.

Recognition and Pattern Matching

Task statement:

- *Given:* A pattern that describes some class of situations;
- *Given:* A description of some specific situation;
- *Find:* All ways in which the pattern matches the situation.

Key ideas:

- The central role of patterns and pattern matching in AI
- The relational character of many patterns and situations
- A pattern’s ability to match a situation in multiple ways

Assignment: Pattern matcher for predicate logic, with situations stated as conjunctions of relational ground literals.

Test Cases: Check legality of puzzle moves, satisfaction of goal descriptions, recognize spatial relations.

Decision Making and Choice

Task statement:

- *Given:* A set of entities and associated descriptions;
- *Given:* A set of goals and/or evaluation criteria;
- *Find:* A selected subset of the original entities.

Key ideas / Decision making:

- Generates candidate choices from among available entities
- Evaluates and then selects from among these options
- Uses features of options and agent objectives influence choices

Assignment: Mechanism that calculates one or more scores for each choice, ranks alternatives, and makes final selection.

Test Cases: Select among categories using a utility function, choose among patterns based on recency of matched elements.

Conceptual Inference and Reasoning

Task statement:

- *Given:* A set of knowledge elements encoding expertise;
- *Given:* A set of beliefs that describe some situation;
- *Given:* An optional query to answer or goal to achieve;
- *Find:* A set of reasoning chains that link facts via knowledge.

Key ideas / Conceptual inference:

- Constructs proof-like structures that link beliefs / queries
- Involves a space of candidate structures, only some of them viable
- Must search this space to find solutions to agents' objectives

Assignment: Deductive engine that carries out AND/OR search through a space of proof trees.

Test cases: Queries about kinship relations, geometry theorems.

Execution and Sequential Control

Task statement:

- *Given:* A description of the agent's current situation;
- *Given:* Knowledge elements with conditional effects of actions;
- *Given:* Goal descriptions and/or evaluation criteria;
- *Find:* Action instances to carry out and expected changes.

Key ideas / Sequential control:

- Combines knowledge and beliefs to generate candidate actions
- Uses knowledge about agent objectives to evaluate alternatives
- Selects a subset of actions and carries them out before continuing

Assignment: Procedure that repeatedly matches, selects, executes actions; extensions for inference and task-directed processing.

Test Cases: Control first-person game agent, approach objects while avoiding obstacles, HTN to collect / assemble objects.

Planning and Problem Solving

Task statement:

- *Given:* Knowledge about conditional effects of actions;
- *Given:* A description of the agent's current situation;
- *Given:* A set of goals and/or evaluation criteria;
- *Find:* Plans to transform current state into one that satisfies goals.

Key ideas / Plan generation:

- Involves mental simulation of action sequences (not execution)
- Requires agent to search through a space of alternative plans
- Uses heuristics to guide this search and make it tractable

Assignment: A planning system that searches through space of candidate plans, using same formalism as sequential control.

Test Cases: Same as for the sequential control assignment.

Integrated Intelligent Agents

Task statement:

- Interact with a complex environment over an extended period in a goal-directed manner.

Key ideas / Intelligent agents:

- Combine pattern matching, choice, inference, execution, planning
- Monitor the progress of generated plans as they are carried out
- Detect anomalies when they arise and revise plans in response

Project: An agent architecture that integrates inference, plan generation, and sequential execution to achieve goals over time.

Test Beds: Cognitive robotics, diagnosis / repair of machinery, task-oriented dialogue.

Course Schedule

Introductory Concepts

1. Intelligence in Humans and Machines
2. Represent., Reasoning, Search, Knowledge
3. Recognition and Pattern Matching
4. Decision Making and Choice

Conceptual Inference and Reasoning

5. Multi-Step Inference
6. Deductive Reasoning
7. Satisfying Constraints
8. Qualitative and Causal Reasoning
9. Abduction and Explanation
10. Analogical Reasoning

Execution and Sequential Control

11. Reactive Control
12. Cognitive and Hierarchical Control
13. Executing and Monitoring Plans

Planning and Problem Solving

14. Problem Solving as Search
15. Heuristic Guidance
16. Generating Plans
17. Adversarial Problem Solving

Integrated Systems

18. Cognitive / Integrated Architectures
19. Application: Cognitive Robotics
20. Application: Diagnosis and Repair
21. Application: Dialogue Systems

Advanced Topics

22. Episodic Memory/Self Explanation
23. Creativity and Discovery
24. Emotion and Personality
25. Moral Reasoning

Review and Summary

Programming Assignments

1a. Implement pattern matcher for predicate logic. Test on patterns for puzzles and games, spatial and temporal relations, and word sequences.

1b. Extend pattern matcher to generate, evaluate, and choose candidates for some decision. Test on selecting moves in games and categorizing entities.

2a. Implement query-driven deductive reasoner for predicate logic. Test on answering kinship queries, proving geometry theorems, parsing sentences.

2b. Extend deductive reasoning engine to support abductive inference. Test on parsing ill-formed sentences, line drawings, and plan understanding.

3a. Implement reactive controller that matches rules, selects candidates, and executes them in environment. Test on game agents and simulated robots.

3b. Extend reactive system to include inference about situations and top-down hierarchical control. Test on game agents and simulated robots.

4a. Implement system that carries out search to find for plans that achieve goals. Test on classic planning tasks, game agents, and simulated robots.

4b. Extend planning system to use inference about situations and hierarchical task knowledge. Test on classic tasks, game agents, and simulated robots.

Advanced Topics

The course omits research progress from the past few decades in areas such as:

- Answer set programming
- Statistical relational inference
- Fast forward planning
- Monte Carlo tree search

These are advanced topics better reserved for later courses, as they make sense only after more basic concepts.

Also, there are no modules devoted to language or vision, but they appear as test cases in some exercises.

Uncertainty and Learning

Russell and Norvig (2009) devote entire sections of their book to uncertainty and learning, but:

- Probabilistic reasoning is best treated as a modulation of classical symbolic approaches.
  - Exercises can extend basic techniques to include probabilities.
  - We should not present them as standalone capabilities.
- Learning builds directly on assumptions about representations and performance, and thus depends on them.
  - Cumulative instruction implies that it be presented after them.
  - This is best taught in a follow-on course, along the same lines.

Both are relevant *across* pattern recognition, decision making, reasoning, execution, and planning.

Closing Remarks

In this talk, I proposed a novel, integrative framework for AI education by:

- Identifying problems with standard introductory AI classes
- Proposing principles that, if adopted, would overcome them
- Describing a new course that would follow these principles by:
  - Building later assignments on results from earlier ones
  - Avoiding the view of AI as a collection of isolated techniques

Course organization differs from the mainstream treatments, including the most popular textbook.

Such a system-level, cumulative approach to AI will better prepare students than algorithm-centered schemes.

End of Presentation