Notes for Meeting 12
Production Systems

Review of Stimulus-Response Systems

Recall that "stimulus-response systems" are characterized by:

- encoding knowledge as a set of stimulus-response patterns;
- operating in cycles, each of which involves:
  - accessing a pattern that matches the current stimuli;
  - executing the associated response in the environment.

This framework, associated with behaviorist psychology and reactive control in AI, uses no short-term mental structures.

Rather, it assumes direct connections between perception and action, and its limitations are obvious.

An Extended Framework

However, we can extend the stimulus-response approach to let it operate over INTERNAL "perceptions" and "actions".

a. Storing long-term knowledge in a stable associative memory that contains abstract symbolic patterns;
b. Storing short-term content in a rapidly changing memory that constrains concrete symbolic elements;
c. Operating in discrete cycles that match long-term patterns against short-term memory and take internal actions;
d. Accessing long-term memory structures is a very rapid or parallel, but altering either memory in a sequential manner.

We will call this as the "standard" model of human cognition because it has been widely adopted in the cognitive science community.

Production System Architectures

An important version of the standard theory is known as a PRODUCTION SYSTEM architecture; it assumes that:

a. Long-term knowledge consists of a set of condition-action rules.
B. Short-term memory contains list structures that denote beliefs, goals, and other dynamic content.
c. Cognition operates in cycles during which:
   1. Conditions of rules match against short-term elements;
   2. A conflict-resolution procedure selects one or more for execution;
   3. The actions of selected instantiations alter short-term memory.

Note that production systems do NOT have a logical semantics; they work entirely through side effects.

Recent production-system frameworks like ACT-R and Soar are associated with theories of the human cognitive architecture.

Conflict Resolution in Production Systems

One of the earliest production-system languages, OPS, incorporated a number of conflict-resolution strategies:

1. Refraction. Do not select a rule instance that has already applied.
2. Recency. Prefer rule instances that match more recent elements.
3. Specificity. Prefer rules with more specialized conditions.
4. Rule order. Prefer rules that were added more recently.
5. Random. Break ties among rule instances randomly.

OPS applied these strategies in a lexicographic manner, using later criteria only if ties occurred on earlier ones.

A Production System for Subtraction

```
((goal (process-all)) (top-of ?c ?digit1) (not (answer-of ?c ?any)) =>
 (*add (goal (process-column ?c))))

((goal (process-column ?c)) (top-of ?c ?digit1) (bottom-of ?c ?digit2)
 (> ?digit1 ?digit2) => (*bind ?digit3 (- ?digit1 ?digit2))
 (*add (answer-of ?c ?digit3)) (*delete (goal (process-column ?c))))

((goal (process-column ?c)) (top-of ?c ?digit1) (bottom-of ?c ?digit2)
 (< ?digit1 ?digit2) => (*add (goal (borrow ?c))))

((goal (borrow ?c1)) (top-of ?c1 ?digit1)
 (left-of ?c2 ?c1) (top-of ?c2 ?digit2) =>
 (*bind ?digit3 (+ ?digit1 10)) (*bind ?digit4 (- ?digit2 1))
 (*delete (top-of ?c1 ?digit1)) (*add (top-of ?c1 ?digit3))
 (*delete (top-of ?c2 ?digit2)) (*add (top-of ?c2 ?digit4)))
```

Applications of Production Systems

The production-system framework has seen wide application in a number of distinct arenas:

- Models of high-level human cognition (e.g., subtraction, 1981)
- Expert systems for complex cognitive tasks (e.g., R1, 1981)
- Synthetic characters for training and entertainment (e.g., TacAirSoar, 1993)
- Adaptive tutoring systems for math and science (Carnegie Learning)

Production systems have seen major success stories in each of these application areas.

History of Research on Production Systems

1. Post defines production systems for string rewriting in 1940s
2. Newell introduces production systems as a framework for cognitive modeling in 1960s.
3. Newell, Waterman, and others develop the first production system languages in early 1970s.
4. Active use of early languages in late 1970s, mostly at Carnegie Mellon, for modeling human cognition.
5. Beginning use of framework for expert systems and exploration of architectural space in 1980s.
6. Development of ACT-R and Soar in mid to late 1980s as architectures for general intelligence and cognitive modeling.
7. Work on expert systems decreases in the 1990s, but production systems remain dominant framework for modeling high-level cognition.

Advantages of Production Systems

Production systems are widely used to model human cognition because they:

- Balance the need for parallel retrieval and sequential behavior.
- Combine the benefits of stimulus-driven control (interruptible) and goal-driven behavior (directed and persistent);
- Partition knowledge into independent and modular elements in ways that support manual construction and learning; and
- Offer an appropriate level of analysis for many reasoning and problem-solving tasks tasks.

They have proved themselves useful for modeling correct and errorful cognition in many domains.

Production Systems and Learning

Learning involves changes to long-term memory, which translates into changes in production rules themselves.

The modular character of production rules offers natural support for learning by:

- Composing existing rules into more complex ones
- Generalizing the condition sides of rules
  - By removing conditions or introducing variables
- Specializing the condition sides of rules
  - By adding conditions or introducing constants
- Changing weights associated with rules

Adaptive production systems offered some of the earliest examples (in the 1970s) of cognitive systems that learned.

Disadvantages of Production Systems

Despite their many benefits, production systems also have some less desirable features:

- Production rules apply in only one direction, limiting flexibility.
- The framework cannot easily encode large structures like schemas.
- Long-term memory does not distinguish between different types of structures like concepts and skills.
- Short-term memory does distinguish between different types of structures like beliefs, goals, and intentions.

The framework can provide each of these capabilities, but it does not support them at the architecture level.

Alternative Architectural Frameworks

Although they are widely used, production systems are not the only candidates for cognitive architectures; alternatives include:

a. Schank’s dynamic memory and Minksy’s society of mind, but these are not worked out in as much detail.
b. Blackboard architectures and multi-agent systems with opaque modules.
c. Architectures like Icarus which have modular structures but that support hierarchy more directly.
d. Connectionist "architectures", but these do not model higher-level aspects of cognition.

The space of cognitive architectures is large, and production systems occupy only one of its corners.

Hierarchical Task Networks and Production Systems

A production system program can encode the same content as that in a hierarchical task, but:

- Each HTN method requires multiple rules, with task and preconditions as the condition side and one subtask as the action side.
- An HTN interpreter can call subtasks directly, while a production system must deposit elements in short-term memory.
- A production system supports persistent behavior only if conflict resolution has a recency bias.

In general, production systems encode procedural knowledge at a finer level of granularity than HTNs.

In another mapping, each path through the method hierarchy corresponds to a single production rule that matches on one cycle.

Assignments for Meeting 13
State-Driven Problem Solving

Read the articles:

- Epstein, S. L. (1999). Game playing: the next moves. Proceedings of the Sixteenth National Conference on Artificial Intelligence (pp. 987-993). Orlando, FL: AAAI Press. [required]

- Tella, J. (1997). Planning in games. Course report, Seminar on Knowledge Engineering, Computer Science Department, Helsinki University of Technology, Helsinki, Finland. [optional]

- VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M. I. Posner (Ed.), Foundations of cognitive science. Cambridge, MA: MIT Press. [optional]

- Complete the fourth exercise (due 11:59 PM on 3/2/2011) and submit solutions on course Web site.