Notes for Meeting 5
Rule-Based Deductive Reasoning

Review of Symbolic Patterns and Pattern Matching

Symbolic patterns are a special type of symbol structure that can characterize classes of situations.

A common form of symbolic pattern is a set of list structures that share variables that refer to the same item.

Symbolic pattern matching lets one find mappings from a pattern to a set of beliefs or facts.

The notion of pattern matching plays a central role in AI and cognitive science.

Reasoning

The ability to REASON is one of the hallmarks of human intelligence.

In the abstract, reasoning involves the generation of a conclusion from one or more other statements.

Reasoning utilizes some form of knowledge to drive such inferences.

The typical form of knowledge is a RULE, which is a special form of symbolic pattern.

The broad field of rule-based systems is built on this key idea.

Applications of Reasoning

We can use reasoning mechanism to automate any task that requires making inferences:

- proving theorems in logic and geometry
- solving problems in physics and thermodynamics
- diagnosing a malfunctioning device
- checking a schedule for constraint violations
- determining if you have satisfied course requirements

One can formulate many real-world tasks in terms of reasoning over symbolic rules.

The Semantic Web is an important upcoming application of rule-based processing.

Symbolic Rules

Before we can discuss processes, we must first consider representation.

We can define a rule as a two-part symbolic pattern that includes:

- Conditions or antecedents (usually a conjunctive pattern) that specify the situations in which the rule applies; and

- Effects or consequents that state the results or conclusions to draw in these situations.

Typical rules include pattern-match variables that are shared across the two parts.

Symbolic rules of this sort underlie much of AI, including many commercial applications.

Examples of Rules

Different frameworks use different formalisms for symbolic rules.

Prolog:

```
between(Block1, Block2, Block3) :-
 block(Block1), block(Block2), block(Block3),
 left-of(Block1, Block2), left-of(Block1, Block2).
```

Icarus:

```
((between ?block1 ?block2 ?block3)
 :percepts  ((block ?block1) (block ?block2) (block ?block3))
 :relations ((left-of ?block1 ?block2)
             (left-of ?block2 ?block3)) )
```

OPS:

```
((block =block1) (block =block2) (block =block3)
 (=block1 left-of =block2) (=block2 left-of =block3)
 =>
 (=block2 between =block1 and =block3))
```

Advantages of Rules

Rule-based representations are useful for building intelligent systems because of their:

- Generality (useful in many different domains)
- Modularity (manageable chunks created independently)
- Dynamic composability (can be combined at run time)
- Declarative character (rather than procedural)

Taken together, these make rules the representation of choice for many AI systems.

Deductive Reasoning

An important special case of rule-based reasoning is deductive inference, which assumes:

- A set of logical rules (typically in first-order predicate logic)

- A set of facts or given statements

Deductive methods use this content to generate new rules or facts that follow deductively.

The "logical AI" paradigm adopts deductive inference as its primary metaphor for studying intelligence.

One-Step Reasoning

The basic operation in deductive inference is one-step reasoning.

Given a logical rule and a set of facts, this involves matches the rule’s antecedents and inferring its consequents.

E.g., assume the facts (left-of A B), (left-of B C), (left-of C D), and the rule

```
    (between ?block1 ?block2 ?block3) <=
       (left-of ?block1 ?block2) (left-of ?block2 ?block3)
```

One-step reasoning would match the rule in two ways and draw two conclusions: (between A B C) and (between B C D).

Dynamic Composition

One-step reasoning has only limited usefulness, but one can also compose rules dynamically to support multi-step reasoning.

This involves chaining two or more rules by matching or unifying antecedents in some with consequents in others.

E.g., assume the facts (taller Abe Bob), (taller Bob Cal), and (taller Cal Dan), along with the rule

```
(taller ?x ?z) <= (taller ?x ?y) (taller ?y ?z)
```

One inference that follows is (taller Abe Dan), which comes from chaining this rule on itself.

Of course, longer chains of rules are possible, which gives the method considerable power.

Logical Resolution

Query-Driven Deductive Inference

Most AI work on deductive reasoning assumes a query-driven approach:

- Given: A set of inference rules, a set of facts, and a query;
- Find: A proof that derives all instances of the query.
- Most approaches to this task reason backward from the query.
- This is sometimes referred to as goal-directed reasoning or backward chaining.

Nearly all languages for logic programming operate in this manner.

Logical databases are one important application, but this approach has been used to many different ends.

Search in Deductive Inference

As Genesereth and Ginsberg note, query-driven deductive inference can require search. This has two aspects:

- selecting a rule to use when chaining off a literal (OR search)

- selecting an antecedent to chain over within a rule (AND search)

Most systems carry out depth-first search through the resulting AND/OR space to produce a proof (AND) tree.

This can lead to extensive backtracking during search, but there has been little progress on more informed search methods.

Early Work on Deductive Reasoning

Some of the earliest AI research focused on deductive reasoning:

- Newell, Shaw, and Simon’s Logic Theorist: First running AI system, it proved theorems in propositional logic, introduced notion of heuristic search, was based on studies of human reasoning.

- Slagle’s SAINT: Solved problems in symbolic integration and clarified notion of search through an AND/OR space, now a common concept in automated reasoning.

- Robinson’s resolution theorem proving: Combined unification with chaining, widely used in the automated reasoning community, led to logic programming languages like Prolog.

Data-Driven Deductive Inference

One can also carry out deductive reasoning in a bottom-up, data-driven manner:

- Given: A set of inference rules and a set of facts;

- Find: Some or all conclusions that follow deductively.

This approach is used in some AI systems, but it is much less common than query-driven methods.

Many human inferences appear to happen in an automated, bottom-up fashion, as in language, but these are not always deductive.

The Icarus inference module operates in a bottom-up manner that draws deductive conclusions from rules and percepts.

Assignments for Meeting 6
Deductive Reasoning

Read the article:

- Leake, D. (1995). Abduction, experience and goals: A model of everyday abductive explanation. Journal of Experimental and Theoretical Artificial Intelligence, 7, 407-428. [Pages 1 to 13]

- Bridewell, W., & Langley, P. (2011). A computational account of everyday abductive inference. Technical report, Institute for the Study of Learning and Expertise, Palo Alto, CA.

- Read Section 2 of the Icarus manual in preparation for the second exercise.