Foundations of Artificial Intelligence

CS472/3 — Fall 1999
Lecture #21

Bart Selman

Slide CS472–1

Today's Lecture

Knowl. Repr., cont.
Inference, cont.
Chapter 8 & 9, R&N.

Slide CS472–2

Schubert Steamroller

```
Wolves, foxes, birds, caterpillars, and snails are
animals, and there are some of each of them.

Also there are some grains, and grains are plants.

Every animal either likes to eat all plants or
all animals much smaller than itself that like
to eat some plants.

Caterpillars and snails are much smaller than birds,
which are much smaller than foxes, which are much
smaller than wolves.
```

Slide CS472–3

```
Wolves do not like to eat foxes or grains, while
birds like to eat caterpillars but not snails.

Caterpillars and snails like to eat some plants.
```

Slide CS472–4

Some logical forms:

$\forall x \ (Wolf(x) \Rightarrow animal(x))$

$\forall x \ \forall y \ ((Caterpillar(x) \land Bird(y)) \Rightarrow Smaller(x, y).$

$\exists x \ \ bird(x)$

To prove:

```
There is an animal that likes to eat
         a grain-eating animal.
```

Requires almost 150 resolution steps (minimal)

Significant challenge for early systems.

Open for about 15 years; solved in late 80s.

Relatively straightforward KB can quickly
overwhelm general resolution methods.
Resolution strategies reduce the problem somewhat,
but not completely.
As a consequence, many **practical** Knowledge Represenation
formalisms in AI use a **restricted form**
and **specialized inference.**
Can often understand them in terms of standard
first-order logic! (clear **syntax & semantics**)

KR Systems (Chapt. 10, R&N)

- **Theorem provers / logic programming**
- **Production systems**
  forward chaining / if-then-rules / **expert** systems
- **Frame systems and semantic networks**
- **Description logics**

Theorem provers / logic programming

Theorem provers: generally based on resolution

many different strategies to improve efficiency

Logic programming: program statements directly in restricted FOL (Horn clauses).

Execution: search for proof of goal/query using backward chaining with depth first-search.

In certain cases too inefficient.

Production systems

- rich history in AI
- **"expert system"** boom in 70's / 80's

Basic idea:

capture knowledge of human expert in a large set of "if-then" rules

(really, logical implication $\Rightarrow$)

"production rules"

Example: Car Diagnosis

```
Rule 1
   If   the engine is getting gas, and
        the engine will turn over,
   then
        the problem is spark plugs.

Rule 2
   If   the engine does not turn over, and
        the lights do not go on
   then
        the problem is battery or cables.
```

Slide CS472–11

```
Rule 3
   If   the engine does not turn over, and
        the lights do come on
   then
        the problem is the starter motor.

Rule 4
   If   there is gas in the fuel tank, and
        there is gas in carburetor
   then
         the engine is getting gas.
```

Slide CS472–12

A goal-driven or backward chaining expert system.

Start with goal-expression e.g. "$\exists X \;\; Problem(X)$"

Match with one of the rules (1, 2, or 3)

Depth-first backward, so pick 1.

Place premises in working memory.

Try to prove premises.

E.g. use rule 4.

Until base facts can be observed.

Slide CS472–13

E.g.:

Problem (X)

Problem (spark plugs) [rule 1]

engine getting gas & engine turns over (OK)

gas in fuel tank (OK) & gas in carb. (OK) &
engine turns over (OK) [rule 4]

bottom facts confirmed; problem "spark plugs" found.

What if one or more bottom facts were not confirmed?

Slide CS472–14

Follows roughly the process of hypothesis testing in human problem solving.

Could it be this? Could it be that? Is so, then this must hold. Otherwise, that must be true. Etc. Etc.

Chaining through rules; either backwards or forwards.

Goes back to early cognitive studies of human problem solving behavior (Newell and Simon 1961; 1972).

Rich history. Led to quite successful **expert systems** (concrete examples later)

Bottleneck: Getting the rules!

The **Knowledge Acquisition Problem.**

(all the knowledge is in the rules)

How does this form of diagnosis differ from the
earlier circuit diagnosis example?
[logical specification circuit
I/O facts
minimal fault set to restore consistency]

Slide CS472–17

Circuit example: diagnosis from **first-principles**
use full description of circuit — **deep knowledge**.

Here **"shallow knowledge".**
no full model of car.
direct encoding of **diagnostic rules**.
[**weak methods** use weak info about domain]
See also chapter 1, R&N.

Slide CS472–18

Deep Blue vs. Kasparove

Which one from "First Principles"?

Success in two opposite directions

1) Computers appear **better at** "reasoning from first principles." (possibly because of raw speed)
   Approach reduces the Knowledge Acquisition Problem.

2) Rule-based approach to the extreme:
   Case-Based Reasoning Systems
   If-then rules, but no chaining; match current situation to a known previous case & do what you did then (single rule).
   E.g. Store 50,000+ car repair scenarios
   use **closest match**

Successes in Rule-Based Reasoning

Expert systems

- DENDRAL (Buchanan *et al.*, 1969)
- MYCIN (Feigenbaum, Buchanan, Shortliffe ca. 1972)
- PROSPECTOR (Duda *et al.*, 1979)
- R1 (McDermott, 1982)

page 22 & 23 R&N.

Slide CS472–21

- DENDRAL (Buchanan *et al.*, 1969)
  - infers molecular structure from the information provided by a mass spectrometer
  - ```
    if  there are peaks at x_1 and x_2 s.t.
          x_1 + x_2 = M + 28
          x_1 - 28 is a high peak
          x_2 - 28 is a high peak
          At least one of x_1 and x_2 is high
    then there is a ketone subgroup
    ```

Slide CS472–22
    ```

- MYCIN (Feigenbaum, Buchanan, Shortliffe)
  - diagnosis of blood infections
  - 450 rules; performs as well as experts
  - incorporated **certainty factors**

```
If: (1) the stain of the organism is
        gram-positive, and
    (2) the morphology of the organism is
        coccus, and
    (3) the growth conformation of the organism
        is clumps,
then there is suggestive evidence (0.7) that the
  identity of the organism is staphylococcus.
```

Slide CS472–23

- PROSPECTOR (Duda *et al.*, 1979)
  - correctly recommended exploratory drilling at a geological site
  - rule-based system founded on probability theory
- R1 (McDermott, 1982)
  - designs configurations of computer components (DEC)
  - about 10,000 rules
  - by 1986, saving company about $40 million a year

Slide CS472–24

Cognitive Modeling with Rule-Based Systems

SOAR is a general architecture for building intelligent systems.

- Long term memory consists of rules.
- Working memory describes current state.
- All problem solving, including deciding what rule to execute, is state space search.
- Successful rule sequences are *chunked* into new rules.
- Control strategy embodied in terms of meta-rules.

Slide CS472–25

What's the difference between rule-based systems and
general theorem proving systems?
(**Horn clause** sentences only.)

Also, forward-chaining production systems
no queries; rules are constinuously fired based
on current state of the knowledge-base.
good for modeling actions / agents.
Advantages / disadvantages of rule-base approach?
What happened to expert systems?

Slide CS472–26