Foundations of Artificial Intelligence

CS472/3
Lecture #16

Bart Selman

Slide CS472–1

Today's Lecture

Synopsis of search.

Knowledge and Reasoning
R&N. Part III

Slide CS472–2

Concludes Problem Solving and Search

- **Problem Solving as Search**
- **Uninformed search:** DFS / BFS / Uniform cost search
  time / space complexity
  size search space: up to approx. 10^{11} nodes
  special case: **Constraint Satisfaction / CSPs**
    generic framework: variables & constraints
    backtrack search (DFS); propagation (forward-
    checking / arc-consistency, variable / value ordering
    (but **incomplete**; N-queens example)

Slide CS472–3

- **Informed search:** use heuristic function guide to goal
  Greedy search
  A search / provably optimal*
    Search space up to approx. 10^{25}
  Local search (incomplete)
    Greedy / hillclimbing / GSAT
    Simulated annealing
    Tabu search
    Genetic Algorithms / Genetic Programming
    search space 10^{100} to 10^{1000}

Slide CS472–4

- **Aversary search / game playing**
  Minimax
    Up to around 10^{10} nodes, 6 — 7 ply in chess.
  alpha-beta pruning
    Up to around 10^{21} nodes, 14 ply in chess.
    provably optimal
- **Hardness or Search Problems**
  Hardest problems: **critically constrained**
    at *phase transition from solvable to unsolvable.*

Slide CS472–5

Search and AI

Why such a central role?

A. Basically, because lots of task in AI are **intractable**.
Search is "only" way to handle them.

Many applications of search, in e.g.,:
Learning / Reasoning / Planning / NLU / Vision.

Good thing: much recent progress (10^{30} quite feasible;
sometimes up to 10^{1000}. **Qualitative difference
from only a few years ago!**

Slide CS472–6

Knowledge and Reasoning

R&N, Part III.

Slide CS472–7

Knowledge and Reasoning

- Human intelligence relies on a lot of background knowledge (the more you know, the easier many tasks become / **“knowledge is power”**)
- E.g. SEND + MORE = MONEY puzzle.
  — **Time flies like an arrow.**
  — **Fruit flies like bananas.**

  The spirit is willing but the flesh is weak. (English)
  The vodka is good but the meat is rotten. (Russian)

OR:
Plan a trip to L.A.

Slide CS472–8

- Q. How did we encode (domain) knowledge so far?
  Search knowledge?

Fine for limited amounts of knowledge / well-defined domains.

Otherwise: **knowledge-based systems approach**.

Slide CS472–9

Knowledge-Based Systems / Programs

General idea: represent knowledge in declarative statements
use inference / reasoning machismn to derive new information / make decisions.

Natural candidate: logical language (propositional / first-order)
combined with a logical inference mechanism

How close to human thought? (mental-models / Johnson-Laird)
In any case, appears reasonable strategy for machines

Slide CS472–10

"Advice-Taker"

1958 / 1968 — John McCarthy: "Programs with Common Sense" — agents use logical reasoning to mediate between percepts and actions.

Idea: Impart knowledge to a program in the form of declarative (logical) statements ("what" instead of "how"); program uses general reasoning mechanisms to process and act on this information.

E.g. Formalize *"x is at y"* using predicate *at*, i.e., *at(x,y)*

at **defined** by its properties, e.g., $at(x, y) \wedge at(y, z) \rightarrow at(x, z)$

Problems??

Slide CS472–11

Agent / Intelligent System Design

p. 13 R&N. Craik (1943) *The Nature of Explanation*

If the organism carries a "small-scale model" of external reality and of its own small possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of the past events in dealing with the present and future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it.

Alt. view: against representations — Brooks (1989)

Slide CS472–12

Representation Language

preferably:

— expressive and concise

— unambiguous and independent of context

— have an effective procedure to derive new information

not easy to meet these goals ...

propositional and first-order logic meet some of the criteria

incompleteness / uncertainty is key — contrast with programming languages. (see p 161 R&N).

Slide CS472–13

```
   printColor(snow) :- !, write("It's white.").
   printColor(grass) :- !, write("It's green.").
   printColor(sky) :- !, write("It's blue.").
   printColor(X) :- write("Beats me.").

Knowledge-based alternative:
   printColor(X) :-
        color(X,Y), !, write("It's "), write(Y), write(".").

   color(snow,white).    (``KB'')
   color(grass,green).
   color(sky,yellow).
```

Slide CS472–14

Logical Representation

Three components:

syntax

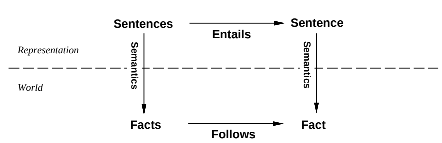
semantics (link to the world)

proof theory ("pushing symbols")

To make it work: **soundness** and **completeness**.

Slide CS472–15

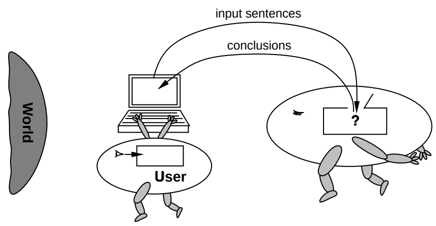
Connecting Sentences to the World



Somewhat misleading: formal semantics brings sentence down only to the primitive components (propositions). (later)

Slide CS472–16

Tenuous Link to Real World



All computer has are sentences (hopefully about the world).

Sensors can provide some grounding.

Hope KB unique model / interpretation: the real-world.

Often many more... (Aside: consider arithmetic.)

More Concrete: Propositional Logic

Syntax: build sentences from atomic propositions, using connectives $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$.

(and / or / not / implies / equivalence (biconditional))

E.g.: $((\neg P) \vee (Q \wedge R)) \Rightarrow S$

p. 167 R&N.

Semantics

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Note: $\Rightarrow$ somewhat counterintuitive.

What's the truth value of "5 is even implies Sam is smart"?

Slide CS472–19

Validity and Inference

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

Truth table for: *Premises* $\Rightarrow$ *Conclusion*.

Shows $((P \vee H) \wedge (\neg H)) \Rightarrow P$ is valid

(True in all interpretations)

We write $\models ((P \vee H) \wedge (\neg H)) \Rightarrow P$

Slide CS472–20

Models

A **model** of a set of sentences (KB) is a truth-assign.
in which each of the KB sentences evaluates to *True*.
With more and more sentences, the models of KB start looking
more and more like the "real-world" (or isomorphic to it).
If a sentence α holds (is *True*) in **all** models
of a KB, we say that α is **entailed** by the KB.
α is of interest, because *whenever KB is true in a world*
α will also be True.
We write: $KB \models \alpha$.

Slide CS472–21

Proof Theory

Purely syntactic rules for deriving the logical consequences of a set of sentences.

We write: $KB \vdash \alpha$, i.e., α can be **deduced** from KB or α is **provable** from KB.

Key property:
Both in propositional and in first-order logic we have a proof theory ("calculus") such that:

$\vdash$ **and** $\models$ **are equivalent.**

Slide CS472–22

Proof Theory

If $KB \vdash \alpha$ implies $KB \models \alpha$, we say the proof theory is **sound**.

If $KB \models \alpha$ implies $KB \vdash \alpha$, we say the proof theory is **complete**.

Why so remarkable / important?

Slide CS472–23

Soundness and Completeness

Allows computer to ignore semantics and "just push symbols"!
In propositional logic, truth tables cumbersome (at least).
In first-order, models can be infinite!

Proof theory: One or more **inference rules** with zero or more axioms (tautologies / to get things "going.").

Slide CS472–24

Example Proof Theory

One rule of inference: **Modens Ponens**

From α and $\alpha \Rightarrow \beta$ it follows that β.

Semantic soundness easily verified. (truth table)

Axiom schemas:

(Ax. I) $\alpha \Rightarrow (\beta \Rightarrow \alpha)$

(Ax. II) $((\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma)))$.

(Ax. III) $(\neg\alpha \Rightarrow \beta) \Rightarrow (\neg\alpha \Rightarrow \neg\beta) \Rightarrow \alpha$.

Note: α, β, γ stand for arbitrary sentences. So, infinite collection of axioms.

Now, α can be **deduced** from a set of sentences Φ

iff there exists a sequence of applications of **modens ponens** that leads from Φ to α (possibly using the axioms).

One can prove that:

Modens ponens with the above axioms will generate exactly all (and only those) statements logically **entailed** by Φ.

So, we have a way of generating entailed statements

in a purely syntactic manner!

(Sequence is called a proof. Finding it can be hard ...)

Example Proof

Lemma. For any α, we have $\vdash (\alpha \Rightarrow \alpha)$.

Proof.

$(\alpha \Rightarrow (\alpha \Rightarrow \alpha) \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow \alpha \Rightarrow \alpha$, (Ax. II)

$\alpha \Rightarrow (\alpha \Rightarrow \alpha) \Rightarrow \alpha$, (Ax. I)

$(\alpha \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow \alpha \Rightarrow \alpha$, (M. P.)

$\alpha \Rightarrow \alpha \Rightarrow \alpha)$ (Ax. I)

$\alpha \Rightarrow \alpha$ (M.P.)

Next time: more efficient using resolution.