CS 4700:
Foundations of Artificial Intelligence

Bart Selman
selman@cs.cornell.edu

Module: Knowledge, Reasoning, and Planning

Logical Agents
Model Theoretic Semantics
Entailment and Proof Theory
R&N: Chapter 7

Logical agents:

Agents with some representation of the complex knowledge about the world / its environment, and uses inference to derive new information from that knowledge combined with new inputs (e.g. via perception).

Key issues:

1- Representation of knowledge

What form? Meaning / semantics?

2- Reasoning and inference processes

Efficiency.

Knowledge-base Agents

Key issues:

- Representation of knowledge → knowledge base
- Reasoning processes → inference/reasoning

Knowledge base = set of sentences in a formal language representing facts about the world (*)

(*) called Knowledge Representation (KR) language

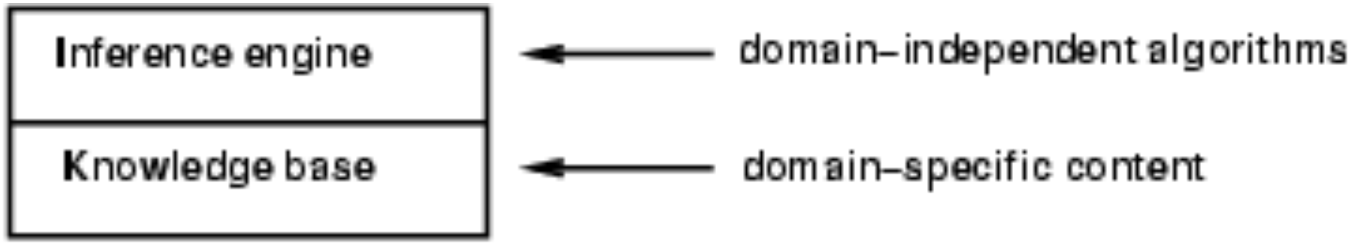
Knowledge bases

Key aspects:

- How to add sentences to the knowledge base
- How to query the knowledge base

Both tasks may involve inference – i.e. how to derive new sentences from old sentences

Logical agents – inference must obey the fundamental requirement that when one asks a question to the knowledge base, the answer should *follow* from what has been told to the knowledge base previously. (In other words the inference process should not "make things" up…)



A simple knowledge-based agent

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time
    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

The agent must be able to:

- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties of the world
- Deduce appropriate actions
-

KR language candidate:

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

How close to human thought? (mental-models / Johnson-Laird).

What is "the language of thought"?

Greeks / Boole / Frege --- Rational thought: Logic?

Why not use natural language (e.g. English)?

We want clear syntax & semantics (well-defined meaning), and, mechanism to infer new information. Soln.: Use a formal language.

“Advice-Taker”

1958 / 1968 — John McCarthy: “Programs with Common Sense” —
agents use logical reasoning to mediate between percepts and a

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??

Consider: to-the-right-of(x,y)

Agent / Intelligent System Design

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)

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.

Logical Representation

Three components:

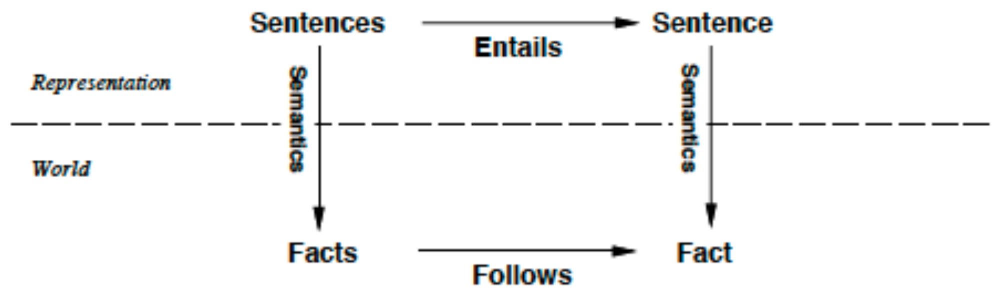
syntax

semantics (link to the world)

proof theory ("pushing symbols")

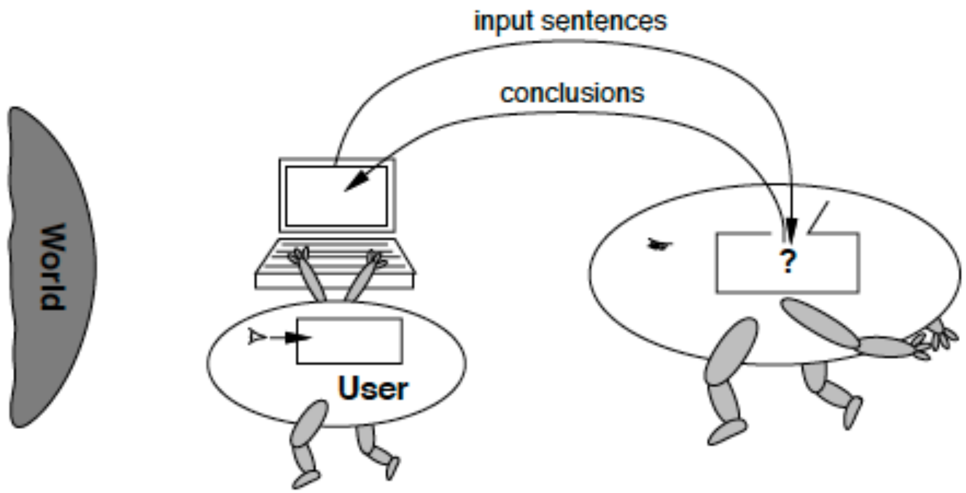
To make it work: **soundness** and **completeness**.

Connecting Sentences to the World



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

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.)

The "symbol grounding problem."

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

Semantics (as before)

P	Q	¬P	P∧Q	P∨Q	P⇒Q	P⇔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"?

True!

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: $Premises \Rightarrow Conclusion.$

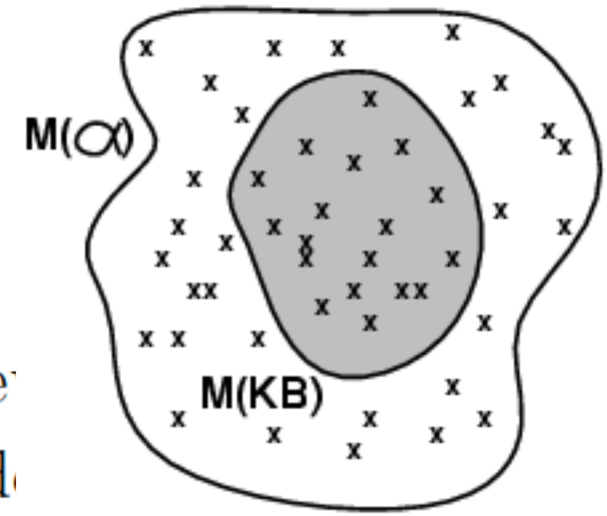
Shows $((P \vee H) \wedge (\neg H)) \Rightarrow P$ is valid

(True in all interpretations) **Logical validity / tautology.**

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

Compositional semantics

Models



A **model** of a set of sentences (KB) is a
in which each of the KB sentences e
With more and more sentences, the mod ng
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: $KB \models \alpha$.

Note: KB defines exactly the set of worlds we are interested in. I.e., our current knowledge about the world.

"KB entails \alpha"

I.e.: **Models(KB)** $\subseteq$ **Models(** α **)**

Observation about “language”

Possibly the key property of a language (both formal and natural) is that **relatively short statements can capture exponentially large sets of possible situations (“worlds”).**

This allows intelligent entities to communicate and think about the **exponential set of possible future world trajectories** and **exponential sets of possible world states** when we only have partial information.

Proof Theory

Purely syntactic rules for deriving the logical consequences o
a set of sentences. **Example soon.**

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.**

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?

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.").

Note: (1) This was Aristotle's original goal --- Construct infallible arguments based purely on the *form of statements* --- not on the "meaning" of individual propositions.

(2) Sets of models can be exponential size or worse, compared to symbolic inference (deduction). I.e., we manipulate short descriptions of exponential size sets.

Example Proof Theory

One rule of inference: **Modus 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 **Modus 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 . . .)

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

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

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

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

Proof.

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

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

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

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

more careful with parentheses ...

1. $(\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha)) \Rightarrow ((\alpha \Rightarrow (\alpha \Rightarrow \alpha)) \Rightarrow (\alpha \Rightarrow \alpha))$ from II, with $(\alpha \Rightarrow \alpha)$ for β & α for γ

2. $(\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha))$ from I, with $(\alpha \Rightarrow \alpha)$ for β

3. $((\alpha \Rightarrow (\alpha \Rightarrow \alpha)) \Rightarrow (\alpha \Rightarrow \alpha))$ by M.P. from 1 & 2

4. $(\alpha \Rightarrow (\alpha \Rightarrow \alpha))$ from I, with α for β

5. $\alpha \Rightarrow \alpha$ by M.P. from 3 & 4

Check steps yourself too

Q.E.D.

Standard syntax and semantics for propositional logic. (CS-2800; see 7.4.1 and 7.4.2.)

Syntax:

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
&\mid\ \neg\, \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \land \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \lor \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \;\Rightarrow\; \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \;\Leftrightarrow\; \mathit{Sentence}
\end{aligned}
$$

$$
\textsc{Operator Precedence} \quad : \quad \neg, \land, \lor, \Rightarrow, \Leftrightarrow
$$

Semantics

Note: Truth value of a sentence is built from its parts "compositional semantics"

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Logical equivalences

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg \alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg \beta \Rightarrow \neg \alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg \alpha \vee \beta) && \text{implication elimination} \quad (*) \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg \alpha \vee \neg \beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg \alpha \wedge \neg \beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

(*) key to go to clausal (Conjunctive Normal Form)
Implication for "humans"; clauses for machines.
de Morgan laws also very useful in going to clausal form.