Knowledge Representation: Logic

Manuela Veloso

CMU 15-381, Fall 2000

- The need for "knowledge" representation
- Symbolic knowledge
- Propositional logic and inference
- Predicate logic
- The resolution inference procedure

Acting in the World

$\boxed{\text{State and Actions}}$

- Complete intelligent agent
  - ▷ Perception - at which *state* am I?
  - ▷ Cognition - which *action* should I take?
  - ▷ Action - how do I *execute* the action?
- State recognition implies state *representation*
- Many examples
- Symbolic knowledge about the world

Knowledge Representation

- One of the goals ("the" goal?) of AI: build intelligent agents
- Need "knowledge" – meaning of facts
- Philosophy
  - ▷ Theory of *meaning* – Natural language to meaning
  - ▷ "I think therefore I am" – logical clarification of thought
- Knowledge base in AI – a set of representations of facts about the world
  - ▷ *What* to represent?
  - ▷ Which level of *detail*? – "frame problem"

Knowledge Representation

  - ▷ Many examples – any task is an example
- Knowledge level – epistemological level: what the agent *knows*
- Implementation level – how knowledge is *represented*
- The representation of knowledge affects dramatically the problem solving capability of an agent
  - ▷ Hard to realize the effects of representation choices
  - ▷ What is a *good* representation?
  - ▷ Example: mutilated checkerboard
- Many possible representations.

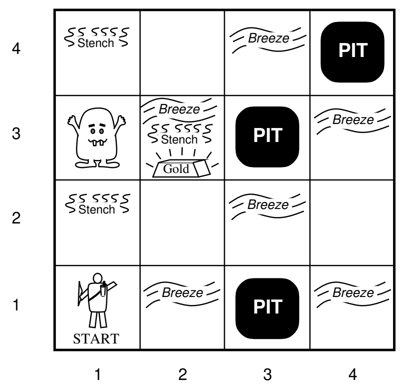
Knowledge Representation

- ▷ Logic
- ▷ Semantic networks
- ▷ Frames
- ▷ Default logics
- ▷ Probabilistic reasoning
- ▷ Graphical representations

Summary

- The wumpus world - Chapter 6, page 154
- Representing the world
- Sentences
- Propositional logic
- Syntax
- Semantics
- Proving the truth of a sentence
- Logical inference
- Discussion

The Wumpus World



Reasoning

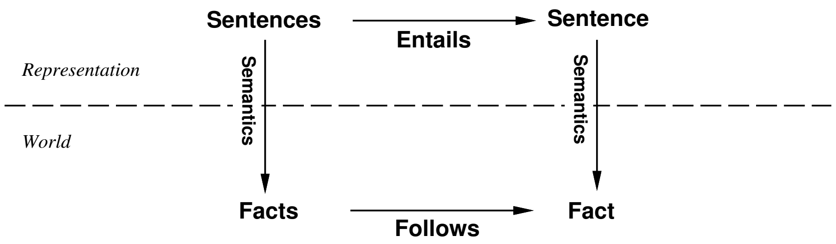
- Perceptions: stentch, breeze, glitter, bump (into wall), scream (wumpus killed)
- Actions: turn left, turn right, go forward, grab, shoot (only once), climb (at start)
- Goal: get the gold back to the start and climb out of the cave

Rules specify the actions to take at a particular state.

Need to perceive and represent the world.

Representation

- Sentences - logic
- Syntax and semantics



Sentences

- stench(2,1)
- "It's raining."
- $a \wedge \neg b$
- "Lisboa é a capital de Portugal."

Syntax

- The grammar: the rules of *legal* sentences.
- Propositional logic

Sentence ::= TerminalSentence | Non-TerminalSentence

TerminalSentence ::= **True** | **False** | *Symbol*

Non-TerminalSentence ::= Sentence Connective Sentence | $\neg$ Sentence

Connective ::= $\wedge$ | $\vee$ | $\rightarrow$

Facts into logic - simple and hard

1. Marcus was a man.
2. All Pompeians were Romans.
3. Caesar was a ruler.
4. All Romans were either loyal to Caesar or hated him.
5. Everyone is loyal to someone.

Facts into logic - simple and hard

6. People only try to assassinate rulers they are not loyal to.

7. Marcus tried to assassinate Caesar.

8. All Romans who know Marcus either hate Caesar or think that anyone who hates anyone is crazy.

9. Every country has a city that has been visited by every person in the country.

Facts into logic - simple and hard

10. For every set, there is always another set with a greater cardinality.

11. All blocks on top of blocks that have been moved or that are attached to blocks that have been moved have also been moved.

Semantics

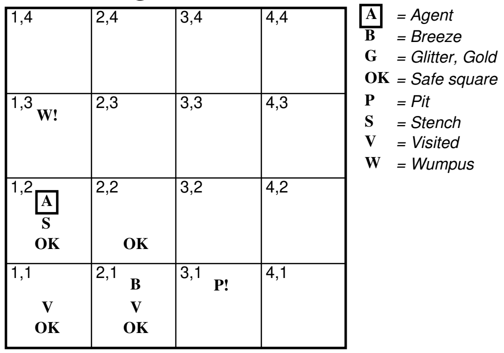
- The complete set of possible truth assignments to the symbols in the sentence: all the possible *worlds*: all **interpretations**.
- Example: $a \vee b$; (true, true), (true, false), (false, true), (true, true)
- A sentence with n propositional symbols has 2^n interpretations.
- A sentence evaluates to True or False - subsets of all interpretations.
- The truth of a sentence depends both on the interpretation and on the state of the world. Example: stench-1-2

Computing the Meaning of a Sentence

- Brute force: A truth table!
- **Valid** sentence: includes all interpretations. Examples: True, other examples?
- **Satisfiable** sentence: A sentence has at least one interpretation. Examples: ?
- **Unsatisfiable** sentence: No interpretations. Examples: False, other examples?
- If valid then satisfiable. If satisfiable, either valid or not valid.
- Either satisfiable or unsatisfiable, not both.

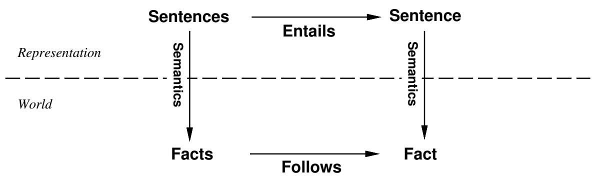
Knowledge Base

A collection of many sentences representing the complete knowledge about the world.



Contents of the knowledge base?

Entailment



The meaning of the knowledge base: other sentences may be true, i.e., other sentences can be *deduced* from the sentences in the knowledge base.

KB $| =$ S

The sentence S is “entailed” by the knowledge base.

Inference Rules for Propositional Logic

- Modus Ponens
- And-Elimination
- And-Introduction
- Or-Introduction
- Double-Negation Elimination

(see page 172 of textbook)

Logical Inference

- Not brute-force generation of the complete truth tables.
- A **proof theory** provides a set of inference rules.
- KB $\vdash$ S ; S is infered from KB by the proof theory.
- **Soundness**:
  KB $| =$ S whenever KB $\vdash$ S
  “what is infered is also true in the real world”
- **Completeness**:
  KB $\vdash$ S whenever KB $| =$ S
  “everything that is true in the real world can be infered by the proof theory”

Modus Ponens

For $d = a_i$,

$$\frac{\begin{array}{r} a_1 \wedge \ldots \wedge a_m \rightarrow b \\ c_1 \wedge \ldots \wedge c_n \rightarrow d \end{array}}{a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge c_1 \wedge \ldots \wedge c_n \rightarrow b}$$

- Modus ponens is sound
  - Proof: By contradiction.
  - Consider model where premises hold but the conclusion does not. In that case $a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge c_1 \wedge \ldots \wedge c_n$ holds but not b.
  - From second premise, then a_i holds.

Modus Ponens

- Then from first premise b also holds as all the a's hold.
- Contradiction.
- Every model in which the premises of modus ponens hold is also a model in which the conclusion holds.

Modus Ponens

For $d = a_i$,

$$\frac{\begin{array}{r} a_1 \wedge \ldots \wedge a_m \rightarrow b \\ c_1 \wedge \ldots \wedge c_n \rightarrow d \end{array}}{a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge c_1 \wedge \ldots \wedge c_n \rightarrow b}$$

- Modus ponens is complete only for Horn databases.
  - A Horn clause is one of the form $a_1 \wedge \ldots \wedge a_m \rightarrow b$, which is equivalent to saying that it has at most one positive atom.
  - Example: incomplete if disjunctions:
  - All classes meet on TTh or MWF. AI meets at 10:30am. Pat has tennis lessons Th and F at 10:30am. Can Pat take AI?

Modus Ponens

1. TTh(AI,10:30) $\vee$ MWF(AI,10:30)
2. TTh(AI,10:30) $\wedge$ busy(Pat,Th,10:30) $\rightarrow$ conflict(Pat,AI)
3. MWF(AI,10:30) $\wedge$ busy(Pat,F,10:30) $\rightarrow$ conflict(Pat,AI)
4. busy(Pat,Th,10:30)
5. busy(Pat,F, 10:30)

- Modus ponens removes busy from the premises, but cannot handle the disjunction.

Predicate Logic – First-order Logic

- The symbols a, b, c, ... are constants.
- The symbols ?a, ?b, ?c, ... are variables.
- Constants are terms.
- Variables are terms.
- (function application) If $\alpha_1 \ldots \alpha_n$ are terms and f is an n-ary function, then $(f\ \alpha_1 \ldots \alpha_n)$ is a term.
- (predicate application) If $\alpha_1 \ldots \alpha_n$ are terms and p is an n-ary predicate, then $(p\ \alpha_1 \ldots \alpha_n)$ is a formula.
- (boolean operators) If $\alpha_1 \ldots \alpha_n$ are formulae then (and $\alpha_1 \ldots \alpha_n$), (or $\alpha_1 \ldots \alpha_n$), (not α_1), and $(\leftarrow\ \alpha_1 \alpha_2)$ are also formulae.

Predicate Logic – First-order Logic

- (quantification) If α is a formula and ?x is a variable, then (forall ?x α) and (exists ?x α) are formulae. α is called the *scope* of ?x, and all occurrences of ?x in α are *bound*.
- (wffs) Occurrences of a variable which are not bound are *unbound*. A *well-formed formula (wff)* is a formula which contains no unbound variables.

Resolution Inference Rule

For $d_j = a_i$ (unify with some substitution),

... in implicative form:

$$\frac{\begin{array}{rcl} a_1 \wedge \ldots \wedge \boxed{a_i} \wedge \ldots \wedge a_m & \rightarrow & b_1 \vee \ldots \vee b_k \\ c_1 \wedge \ldots \wedge c_n & \rightarrow & d_1 \vee \ldots \vee \boxed{d_j} \vee \ldots \vee d_l \end{array}}{\begin{array}{r} a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge c_1 \wedge \ldots \wedge c_n \rightarrow \\ b_1 \vee \ldots \vee b_k \vee d_1 \vee \ldots \vee d_{j-1} \vee d_{j+1} \vee \ldots d_l \end{array}}$$

Resolution Inference Rule

For $d_j = a_i$ (unify with some substitution),

... in conjunctive form:

$$\frac{\begin{array}{rcl} \neg a_1 \vee \ldots \vee \boxed{\neg a_i} \vee \ldots \vee \neg a_m & \vee & b_1 \vee \ldots \vee b_k \\ \neg c_1 \vee \ldots \vee \neg c_n & \vee & d_1 \vee \ldots \vee \boxed{d_j} \vee \ldots \vee d_l \end{array}}{\begin{array}{r} \neg a_1 \vee \ldots \vee \neg a_{i-1} \vee \neg a_{i+1} \vee \ldots \vee \neg a_m \vee \neg c_1 \vee \ldots \vee \neg c_n \vee \\ b_1 \vee \ldots \vee b_k \vee d_1 \vee \ldots \vee d_{j-1} \vee d_{j+1} \vee \ldots d_l \end{array}}$$

Applying the Resolution Rule

- Refutation: To prove α, assume α is false and prove that there is a contradiction in the KB , i.e. prove that *false* can be derived from KB.

$$(\mathrm{KB} \wedge \neg\alpha \vdash \mathit{false}) \Leftrightarrow (\mathrm{KB} \vdash \alpha)$$

- Apply the resolution rule (*search*!) until *false* is derived.

Example

1. True $\rightarrow$ TTh(AI,10:30) $\vee$ MWF(AI,10:30)
2. TTh(AI,10:30) $\wedge$ busy(Pat,Th,10:30) $\rightarrow$ conflict(Pat,AI)
3. MWF(AI,10:30) $\wedge$ busy(Pat,F,10:30) $\rightarrow$ conflict(Pat,AI)
4. True $\rightarrow$ busy(Pat,Th,10:30)
5. True $\rightarrow$ busy(Pat,F, 10:30)
6. conflict(Pat,AI) $\rightarrow$ False

Example

- Resolve 3. and 5. resulting
  7. MWF(AI,10:30) $\rightarrow$ conflict(Pat,AI)
- Resolve 2. and 4. resulting
  8. TTh(AI,10:30) $\rightarrow$ conflict(Pat,AI)
- Resolve 1. and 7. resulting
  9. True $\rightarrow$ conflict(Pat,AI) $\vee$ TTh(AI,10:30)
- Resolve 8. and 9. resulting
  10. True $\rightarrow$ conflict(Pat,AI)
- Resolve 10. and 6. resulting
  11. True $\rightarrow$ False

Convertion to Normal Form

To perform $\vdash$, i.e., inference using the resolution inference rule, we need to show that any FOL sentence can be put into implicative normal form, or conjunctive normal form.

1. Eliminate implications.
2. Reduce scope of negation.
3. Standardize variables - each quantifier binds a unique variable.
4. Skolemize - Eliminate existential quantifiers – Introduce a Skolem function for each existentially quantified variable.

Convertion to Normal Form

5. Convert to prenex form - Move quantifiers to the left – prefix.
6. Drop the prefix.
7. Distribute `and`, $\wedge$ and `or`, $\vee$, to get a conjunction of disjunctions.
8. Split conjunction - create a separate clause to each conjunct.
9. Standardize variables apart again.

Resolution Example

Prove ¬immortal(Socrates) from the KB:

$\forall x \quad \text{man}(x) \rightarrow \text{mortal}(x)$

man(Socrates)

$\forall x \quad \neg\text{mortal}(x) \vee \neg\text{immortal}(x)$

Convertion to normal form: KB + negated fact:

1. immortal(Socrates)
2. $\neg\text{man}(x) \vee \text{mortal}(x)$
3. man(Socrates)
4. $\neg\text{mortal}(z) \vee \neg\text{immortal}(z)$

Resolution Example

- Resolve 2. and 4.:
  $\text{mortal}(x)$ matches $\text{mortal}(z)$

  results 5. $\neg\text{man}(x) \vee \neg\text{immortal}(x)$
- Resolve 5. and 3.:
  $\text{man}(x)$ matches man(Socrates)
  results 6. $\neg\text{immortal}(Socrates)$
- Resolve 1. and 6: derive *false*.

FOL – All you have to know

- Knowledge base (KB): well formed formulas (clauses).
- We want to find if a fact is *entailed* by the KB, i.e. if it holds in any model given that the facts in the KB hold.
- *Derive* facts from KB using *rules of inference*.
- *Sound* inference rules derive facts that are entailed by the KB.
- *Modus ponens* is an inference rule.
- Modus ponens is sound for FOL.
- If a fact is entailed by the KB, then a *complete* inference rule(s) will also derive it.

FOL – All you have to know

- Modus ponens is not complete for FOL.
- Modus ponens is complete for KBs of *Horn clauses*. Horn clauses are clauses with only disjunctions and only one positive term.
- *Resolution* is an inference rule. Resolution is a generalization of modus ponens.
- Resolution is sound and complete for FOL.
- There is a resolution procedure.
- Gödel's incompleteness theorem: If FOL is extended with constructs to handle mathematical induction, then there is no complete inference procedure.