Overview

- Project 1 tips
- First-order logic resolution example
- Application of theorem proving: question answering
- Uncertainty

Announcement

Project 1 due was postponed to Sunday 3/24 midnight.

Example: Drug Dealer and Customs Official

1. The customs officials searched everyone who entered the country who was not a VIP. (2) Some of the drug dealers entered the country, and they were only searched by drug dealers. (3) No drug dealer was a VIP. (4) Some of the customs officials were drug dealers.

   1. \( \forall x ((E(x) \land \neg V(x)) \rightarrow \exists y (S(x, y) \land C(y))) \)
   2. \( \exists x (E(x) \land D(x) \land \forall y (S(x, y) \rightarrow D(y))) \)
   3. \( \forall x (D(x) \rightarrow \neg V(x)) \)
   4. \( \exists x (D(x) \land C(x)) \)

Example: Clauses

(1a) \( \neg E(x) \lor V(x) \lor S(x, f(x)) \)
(1b) \( \neg E(x) \lor V(x) \lor C(f(x)) \)
(2a) \( E(a) \)
(2b) \( D(a) \)
(2c) \( \neg S(a, y) \lor D(y) \)
(3) \( \neg D(x) \lor \neg V(x) \)
(4) \( \neg D(x) \lor \neg C(x) \)

Note: clause 3 was incorrect in slide 21 – it was \( \neg D(x) \lor V(x) \)

Exercise 1: rewrite the above in LISP representation.

Exercise 2: use resolution to derive \( F \).
Exercise 1 Solution: Clauses in Lisp

\[<\text{clause-num}> <\text{pos-lit-list}> <\text{neg-lit-list}>\]

(1) \(\neg E(x) \lor V(x) \lor S(x, f(x))\)
   - (1 ((V X) (S X (f X))) ((E X)))

(2) \(\neg E(x) \lor V(x) \lor C(f(x))\)
   - (2 ((V X) (C (F X))) ((E X)))

(3) \(E(a)\)
   - (3 ((E (A))) NIL)

(4) \(D(a)\)
   - (4 ((D (A))) NIL)

(5) \(\neg S(a, y) \lor D(y)\)
   - (5 ((D Y)) ((S (A) Y)))

(6) \(\neg D(x) \lor \neg V(x)\)
   - (6 NIL ((D X) (V X)))

(7) \(\neg D(x) \lor \neg C(x)\)
   - (7 NIL ((D X) (C X)))

Exercise 2 Solution: Resolution

\[<\text{clause-num}> <\text{pos-lit-list}> <\text{neg-lit-list}>\]

(1) \(\neg E(x) \lor V(x) \lor S(x, f(x))\)

(2) \(\neg E(x) \lor V(x) \lor C(f(x))\)

(3) \(E(a)\)

(4) \(D(a)\)

(5) \(\neg S(a, y) \lor D(y)\)

(6) \(\neg D(x) \lor \neg V(x)\)

(7) \(\neg D(x) \lor \neg C(x)\)

(8) \(\neg V(a) \lor C(f(a))\)

(9) \(\neg S(a, f(a))\)

(10) \(\neg D(f(a))\)

(11) \(D(f(a))\)

(12) \(\text{False}\)

Application of the Theorem Prover: Question Answering

- Given a database of facts (ground instances) and axioms, we can pose questions in predicate calculus and answer them using resolution.
- Resolution can answer Yes/No answers, but it can be extended to answer more complex questions such as Who? or What?, etc. This is called Answer Extraction.

Project 2 Brief Overview

Write a theorem prover using resolution:

- Input: axioms and negated conclusion in standard form, in Lisp.
- Algorithm: resolution (1) two-pointer method, (2) unit-preference, (3) linear resolution.
- Improvements: (1) use set of support, (2) use deletion strategy.
- Output: success or failure, with resolution steps if successful.

Detailed specifications will be announced shortly.
Question Answering: Example

Example:

1. \( \forall x \forall y \forall z ((\text{Parent}(x,z) \land \text{Parent}(z,y)) \rightarrow \text{Grandparent}(x,y)) \)
2. \( \forall x \forall y (\text{Mother}(x,y) \rightarrow \text{Parent}(x,y)) \)
3. \( \forall x \forall y (\text{Father}(x,y) \rightarrow \text{Parent}(x,y)) \)
4. \( \text{Father}(\text{Zeus, Ares}) \)
5. \( \text{Mother}(\text{Hera, Ares}) \)
6. \( \text{Father}(\text{Ares, Harmonia}) \)

Question: "Who is a grandparent of Harmonia?"

1. \( \exists x (\text{Grandparent}(x, \text{Harmonia})) \)

Negated: \( \neg \exists x (\text{Grandparent}(x, \text{Harmonia})) \)

\( \equiv \forall x (\neg \text{Grandparent}(x, \text{Harmonia})) \)

Answer Extraction

We can introduce special predicates to extract the answers.

- **Answer predicate:**
  \( \neg \text{Grandparent}(x, \text{Harmonia}) \lor \text{Answer}(x) \)

- The answer predicate has these properties:
  - It does not resolve with anything, but it keeps track of variable bindings.
  - The theorem prover recognize a clause consisting only of the \text{Answer} predicate as \( \text{F} \).

- For example, resolution on the previous example results in:
  \( \text{Answer}(\text{Hera}) \)
  as the final clause.

Question Answering: Result

- Resolution on the previous example generates \( \text{F} \) in the end, but what that answers is the question "Is there a grandparent of Harmonia?". Of course the answer is \text{yes}, but the question is who?

- The negated question in the above examples was \( \neg \text{Grandparent}(x, \text{Harmonia}) \). Clearly, the binding which \( x \) ultimately receives is the desired answer!

- Observation: one substitution along the way, starting from \( \neg \text{Grandparent}(x, \text{Harmonia}) \), the negated conclusion, is \( \{x/\text{Hera}\} \), thus \text{Hera} must be an answer.

Exercise: use resolution to derive \( \text{F} \) in the example in the previous slide.

First-Order Logic: Summary

- Standard forms: prenex normal form, skolemization, CNF.

- Resolution: negated conclusion, substitution, unification, factors and resolvents.

- Theorem provers: two-pointer method, various deletion strategies, various speed up strategies.

- Application of theorem provers: question answering.
Uncertainty

- Problem with first-order logic: agents almost never have full access to the whole truth about their environment.
- Therefore, the agent must act under uncertainty.
- Uncertainty can also arise because of incompleteness and incorrectness in the agent's understanding of the properties in the environment.
- Incomplete, because there are too many conditions to explicitly enumerate.

There are trade-offs (playing safe can result in other annoyances), thus the right thing to do depends on both the relative importance of various goals and the likelihood (and degree to which) they will be achieved.

Example: Trying to Catch a Flight

$A_t$: plan to leave home $t$ minutes before the flight departure time.

- The traveler needs to make a decision in an uncertain environment: car can break down, traffic can be extremely congested, natural disaster, etc.
- Such worst-case scenarios are hard to explicitly enumerate: the list goes on – ran out of gas, spouse/children in an emergency, flight crews goes on a strike, etc. etc.
- Thus the traveler only has an incomplete understanding of the situation.
- The traveler can play safe by going with plan $A_{1440}$, but this can cause the traveler to wait for a long time at the airport before departure.

Acquisition of New Information and Probability

- The degree of belief changes as an agent perceives or acquires new information from the world: we call this the evidence.
- This is analogous to saying whether or not a given logical sentence is entailed by (i.e. is a logical consequence of) the knowledge base, because the truth value can change when new facts are added to the KB.
- Before the evidence is received, we talk about prior or unconditional probability.
- After the evidence is obtained, we talk about posterior or conditional probability.

Difficulties in Applying F-O-L in Uncertain Domains

For example, application of first-order logic in medical diagnosis domain can fail because of these reasons:

- Laziness: cannot list the complete set of antecedents and consequents needed to ensure an exceptionless rule, and too hard to use the enormous rules that result.
- Theoretical ignorance: medical science has no complete theory.
- Practical ignorance: even though we have all the rules, it is practically impossible to run all the tests.

Similar situation arises in law, business, dating, etc. The agent's knowledge can at best provide only a degree of belief. Probability theory is well suited for such a domain.
Example

When playing black jack,

- as new cards are drawn and shown, your degree of belief in the fact that you need more cards can change.

What about poker? or slot machine?

Rational Decisions Under Uncertainty: Decision Theory

- There are trade-offs, and an agent must first have preferences between different results when a certain plan was executed.

- Utility theory deals with such preferences: how useful is such and such result to the agent?

- Decision theory is a general theory of rational decision under uncertainty, combining probability theory and utility theory.

Decision Theory

- An agent is rational iff it chooses the action that yields the highest expected utility, averaged over all possible outcomes of the action:
  
  **Principle of Maximum Expected Utility**

- Example: backgammon (discussed earlier) – min-max trees with probabilistic levels.

Decision Theoretic Agent

```plaintext
function DT-Agent (percept) returns action

static: a set probabilistic belief about the state of the world

calculate updated probabilities for current state based on percept and past actions

calculate outcome probabilities for actions, given action descriptions and prob of current states.

select action with highest expected utility given prob of outcomes and utility information.

return action
```
Next Time

- Uncertainty: chapter 14 continued
- Probabilistic reasoning: chapter 15