Lab Tutorial 6

Course: Artificial Intelligence (503030)

I. Objectives

- Overview logic programming.
- Introducing propositional logic, first-order logic and proof methods.
- SWI-Prolog IDE and *family tree problem*.

II. Logic Programming

In logic programming, a program consists of a collection of statements expressed as formulas in symbolic logic. There are rules of inference from logic that allow a new formula to be derived from old ones, with the guarantee that if the old formulas are true, so is the new one.

For more details, visit http://en.wikipedia.org/wiki/Logic_programming or read "*An Introduction to Logic Programming Through Prolog (Prentice Hall International Series in Computer Science)*".

III. Propositional Logic

1. Interpretation

Two logic variables P and Q as sentences, we have a truth table with **5** logical connectives:

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

2. Entailment

Entailment means that one thing follows from another, $KB \vDash \alpha$. Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true.

Validity and satisfiability: a sentence α is valid if it is true in all models; a sentence is satisfiable if it is true in some model; a sentence is unsatisfiable if it is true in no models.

3. Logical equivalence

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

4. Proof Methods

- Inference rules: *(i)* transform KB to conjunctive normal form (CNF); *(ii)* apply a sequence of inference rules. In order to prove a KB entails α, we usually use contradiction, i.e., show KB $\land \neg\alpha$ unsatisfiable.
- Model checking: use truth table, but it is a NP-Complete problem.

IV. First-Order Logic (FOL)

1. Introduction

FOL (like natural language) assumes the world contains objects, relations and functions.

2. Syntax

- Constant symbols are to represent primitive objects.
- Variable symbols are to represent unknown objects.
- Predicate symbols are to represent relations. Example: married(John); love(John, Marry); ….
- Functions symbols are to represent simple objects. Example: safe(square(1, 2)); ….
- Terms are to represent complex objects. Example: love(mother(father(John)), John); ….
- Logical connectives: ¬, ∧, ∨, ⇒, ⇔.
- Equality ("="). Example: father(John) = Henry; ….

- Universal quantifiers: $\forall x\ P(x)$. Typically, $\Rightarrow$ is the main connective with $\forall$.
- Existential quantifiers: $\exists x\ P(x) \cong \neg\forall x\ \neg P(x)$. Typically, $\wedge$ is the main connective with $\exists$.

3. FOL

- Brothers are siblings: $\forall x, y\ Brother(x, y) \Rightarrow Sibling(x, y)$.
- One's mother is one's female parent: $\forall x, y\ Mother(y) = x \Leftrightarrow \left(Female(x) \wedge Parent(y, x)\right)$.

4. Inference Rules with Quantifiers

- Substitution.
- Universal Elimination.
- Existential Elimination.

5. Generalized Resolution

- Refutation proof procedure: $KB \vDash \alpha \Leftrightarrow (KB \wedge \neg\alpha \vDash false)$.
- Conversion to Normal Form:
  + Eliminate implications.
  + Move $\neg$ inwards.
  + Standardize variables.
  + Move quantifiers left.
  + Skolemize (to remove $\exists$).
  + Distribute $\wedge$ over $\vee$.
  + Flatten nested conjunctions and disjunctions.

V. SWI-Prolog

1. Information & Download

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications.

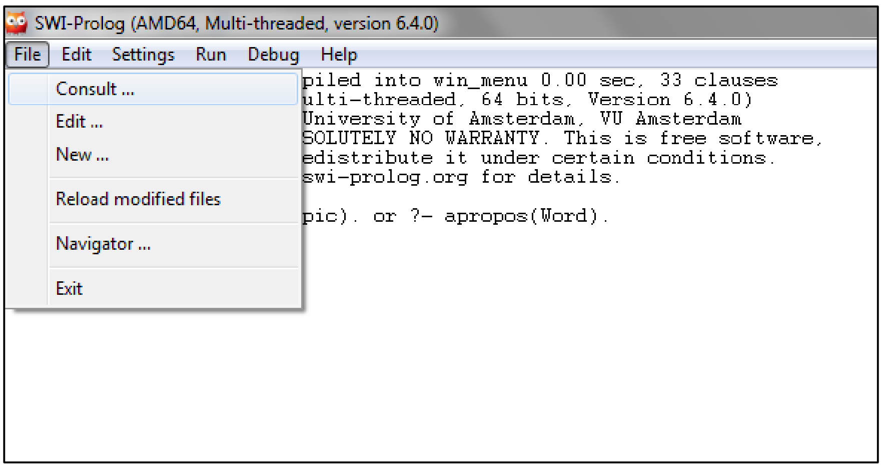
Download: http://www.swi-prolog.org/Download.html

2. Family tree problem

- Example a knowledge base of *family tree*:

```
% AI-Prolog-FamilyTreeExample, written by DHP.

male(tom).
male(bob).
male(jim).

female(liz).
female(pat).
female(ann).
female(pam).

% pam is parent of bob
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).

offspring(X,Y):- parent(Y,X).
pred(X,Y):-parent(X,Y).
pred(X,Y):-parent(X,Z),pred(Z,Y).
same(X,Y):- X=Y.
diff(X,Y):- not(same(X,Y)).
```

- Compile this KB in SWI-Prolog environment (File > Consult … > Choose *.pl* file):



- Base on the defined inference rules, "ask" the KB some question (line-by-line):

```
parent(tom,bob).
mother(tom,bob).
parent(Y,ann).
```

VI. Exercises

1. Run the demo of family tree problem in section **V.2**.

2. Determine using a truth table whether the following sentence is valid, satisfiable, or unsatisfiable:

   a. $(P \land Q) \lor \neg Q$.

   b. $((P \land Q) \Rightarrow R) \Leftrightarrow ((P \Rightarrow R) \lor (Q \Rightarrow R))$.

   c. $P \Rightarrow (Q \Rightarrow R) \Rightarrow P$.

   d. $[P \Rightarrow (Q \Rightarrow R)] \Rightarrow (P \Rightarrow \neg R)$

3. Assume that a KB contains the following rules:

   $poor \Rightarrow \neg worried,\ rich \Rightarrow scared,\ \neg rich \Rightarrow poor.$

   Use *resolution* to prove KB entails $worried \Rightarrow scared$.

4. If it rains, John brings his umbrella. If John has an umbrella, he doesn't get wet. If it doesn't rain. John doesn't get wet. Using resolution, prove that John doesn't get wet.

--- THE END ---

Resource: http://it.tdt.edu.vn/~dhphuc/teaching/artificial-intelligence/