LING 364: Introduction to Formal Semantics

Lecture 22

April 6th
Administrivia

• **Homework 5**  
  – on quantification  
  – out today  
  – due next Thursday
Exercise 1

• Truth Tables and Prolog
Exercise 1

- **Truth tables in Prolog**
- **Example**:
  - % implies(P,Q,Result)
  - implies(true,false,false).
  - implies(false,true,true).
  - implies(false,false,true).
  - implies(true,true,true).
  - % or(P,Q,Result)
  - or(true,true,true).
  - or(true,false,true).
  - or(false,true,true).
  - or(false,false,false).

Show using a Prolog query that implies/3 and or/3 are not equivalent

?- implies(P,Q,R1), or(P,Q,R2), \+ R1 = R2.

What should the outcome of this query be?
Exercise 1

• Homework Question A (3pts)
  – Using the Prolog query shown on the previous slide,
  – for what values of P and Q are implies/3 and or/3 incompatible?
  – Submit your run
Exercise 1

• Define truth table negation as follows:
  – \( \neg \text{neg}(P, \neg P) \).
  – \text{neg}(\text{true}, \text{false}).
  – \text{neg}(\text{false}, \text{true}).

<table>
<thead>
<tr>
<th>P</th>
<th>\neg P</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

• Show using a Prolog query that \( P \Rightarrow Q \) is equivalent to \( \neg P \lor Q \)

\ [-implies(P, Q, R1), \text{neg}(P, \text{NotP}), \text{or(NotP, Q, R2)}, \lor R1 = R2.]

What should the outcome of this query be?
Exercise 1

• Homework Question B (2pts)
  – Define truth table and/3 in Prolog

<table>
<thead>
<tr>
<th>P</th>
<th>^</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

• Homework Question C (4pts)
  – Using an appropriate Prolog query, and and/3,
  – Show that \( \neg(P \lor Q) = \neg P \land \neg Q \) (De Morgan’s Rule)
  – Submit your run
Exercise 1

• Homework Question D (4pts)
  – Using an appropriate Prolog query,
  – Show that \( \neg(P \land Q) = \neg P \lor \neg Q \)
  – (another side of De Morgan’s Rule)
  – Submit your run
Exercise 1

• Summary
  – Submit answers to questions A through D
  – Points:
    – A: 3pts
    – B: 2pts
    – C: 4pts
    – D: 4pts
  – Total: 13 pts
Exercise 2

• Universal Quantification and Sets
Exercise 2

- **Assume meaning grammar:**

  s(M) → qnp(M), vp(P), \{predicate2(M,P)\}.
  qnp(M) → q(M), n(P), \{predicate1(M,P)\}.
  q(findall(_X,_P1,L1),findall(_Y,_P2,L2),subset(L1,L2)) → [every].
  n(woman(_)) → [woman].
  vp(M) → v(M), np(X), \{saturate2(M,X)\}.
  v(likes(_X,_Y)) → [likes].
  np(ice_cream) → [ice, cream].

  saturate1(P,X) :- arg(1,P,X).
  saturate2(P,X) :- arg(2,P,X).
  subset([],_).
  subset([X|L1],L2) :- member(X,L2),
  subset(L1,L2).
  member(X,[X|_]).
  member(X,[_|L]) :- member(X,L).

  predicate1((findall(X,P,_,_),_),P) :-
  saturate1(P,X).
  predicate2((_,(findall(X,P,_,_),_)),P) :-
  saturate1(P,X).

  **every** has semantics:
  \{X: P_1(X)\} ⊆ \{Y: P_2(Y)\}
  e.g.
  every woman likes ice cream
  \{X: woman(X)\} ⊆ \{Y:likes(Y,ice_cream)\}
Exercise 2

• Using the meaning grammar, we can compute a meaning expression for:
  – every woman likes ice cream
  using the Prolog query:
  – ?- s(M,[every,woman,likes,ice,cream],[[]]).
  – M = findall(A,woman(A),B),findall(C,likes(C,ice_cream),D),subset(B,D)
Exercise 2

- We can evaluate this meaning expression for various possible worlds using call/1
- For example, given the database:
  - woman(mary). woman(jill).
  - likes(john,ice_cream). likes(mary,ice_cream).
  - likes(jill,ice_cream).
- we can evaluate:
  - ?- s(M,[every,woman,likes,ice,cream],[[]]), call(M).
- the call is:
  - findall(A,woman(A),B),findall(C,likes(C,ice_cream),D),subset(B,D).
- with
  - B and D being [mary,jill] and [john,mary,jill] respectively
Exercise 2

• **Homework Question A (4pts)**
  – Modify the meaning grammar to handle the sentence
    • John likes ice cream

• **Homework Question B (2pts)**
  – Evaluate *John likes ice cream* against the database from the previous slide
  – Submit your run
Exercise 2

- **Homework Question C (10pts)**
  - Treating names as Generalized Quantifiers (see below),
  - Further modify the meaning grammar to handle the sentences
    - Every woman and John likes ice cream
    - John and every woman likes ice cream
  - Evaluate the sentences and submit your runs

**Recall Lecture 21**

**Example**

every baby and John likes ice cream

\[
[NP_{NP} \text{every baby}] \text{ and } [NP_{NP} \text{John}] \text{ likes ice cream}
\]

\[
\{X: \text{baby}(X)\} \cup \{X: \text{john}(X)\} \subseteq \{Y: \text{likes}(Y, \text{ice\textunderscore cream})\}
\]

**note:** set union \( \cup \) is the translation of \textit{“and”}
Exercise 2

• Summary
  – Answer questions A, B and C
  – A: 4pts
  – B: 2pts
  – C: 10pts
  – Total: 16pts
Exercise 3

• Other quantifiers as generalized quantifiers
Exercise 3

• Other quantifiers can also be expressed using set relations between two predicates:

Example:

\[
\text{no: } \{X: P_1(X)\} \cap \{Y: P_2(Y)\} = \emptyset
\]

\(\cap\) = set intersection

\(\emptyset\) = empty set

no man smokes

\[
\{X: \text{man}(X)\} \cap \{Y: \text{smokes}(Y)\} = \emptyset
\]

should evaluate to true for all possible worlds where there is no overlap between men and smokers
Exercise 3

• Other quantifiers can also be expressed using set relations between two predicates:

Example:

\(\text{some}: \{X: P_1(X)\} \cap \{Y: P_2(Y)\} \neq \emptyset\)

\(\cap\) = set intersection

\(\emptyset\) = empty set

\(\text{some men smoke}\)

\(\{X: \text{man}(X)\} \cap \{Y: \text{smokes}(Y)\} \neq \emptyset\)
Exercise 3

• **Homework Question A (8pts)**
  – Modify the meaning grammar given in exercise 2 to handle the sentence:
  – *No woman likes ice cream*
  – Evaluate it against the database

• **Homework Question B (8pts)**
  – Modify the meaning grammar given in exercise 2 to handle the sentence:
  – *Some women like ice cream*
  – Evaluate it against the database
Exercise 3

• **Summary**
  – Submit parts A and B
  – and the runs
  – A: 8pts
  – B: 8pts
  – Total: 16pts
Summary

• PLEASE SUBMIT EVERYTHING IN ONE FILE!
• Exercises
  • 1: 13pts
  • 2: 16pts
  • 3: 16pts
• Grand total: 35pts