LING 364: Introduction to Formal Semantics

Lecture 24
April 13th
Administrivia

• **Homework 4**
  – has been returned
  – if you didn’t get an email from me, let me know

• **Homework 5**
  – usual rules
  – due tonight
  – main focus of today’s lab session: answer questions you may have
• First, quiz 5 (from last lecture) review...

Table (26) in handout is incorrect ...
• **Question 1**
  – Is **Some** UE or DE for $P_1$ and $P_2$?
    * some: $\{X: P_1(X)\} \cap \{Y: P_2(Y)\} \neq \emptyset$
  – Justify your answer using examples of valid/invalid inferences starting from
    * Some dog barks

  ![Diagram](image)

  **Answer:**
  – **UE for $P_1$**
  – Some dog barks
  – Some animal barks (OK)
  – Some Keeshond barks
  – **UE for $P_2$**
  – Some dog barks
  – Some dog makes noise (OK)
  – Some dog barks loudly
Quiz 5

- **Question 2**
  - Is No UE or DE for $P_1$ and $P_2$?
    - $no: \{X: P_1(X)\} \cap \{Y: P_2(Y)\} = \emptyset$
  - Use
    - No dog barks

- **Answer:**
  - DE for $P_1$
    - No dog barks
    - No animal barks
    - No Keeshond barks (OK)
  - DE for $P_2$
    - No dog barks
    - No dog makes noise
    - No dog barks loudly (OK)
Homework 5

• Exercise 1: Truth Tables
• Exercise 2: Universal Quantification and Sets
  – Hint for Question 3
• Exercise 3: Other quantifiers as generalized quantifiers
  – Hint
Exercise 2

• Assume meaning grammar:

\[
\begin{align*}
\text{s}(M) & \rightarrow qnp(M), \text{vp}(P), \{\text{predicate2}(M,P)\}.
\text{qnp}(M) & \rightarrow q(M), \text{n}(P), \{\text{predicate1}(M,P)\}.
\text{q}((\text{findall}(_X,_P1,L1),\text{findall}(_Y,_P2,L2),\text{subset}(L1,L2))) & \rightarrow [\text{every}].
\text{n}(\text{woman}(\_)) & \rightarrow [\text{woman}].
\text{vp}(M) & \rightarrow \text{v}(M), \text{np}(X), \{\text{saturate2}(M,X)\}.
\text{v}(\text{likes}(_X,_Y)) & \rightarrow [\text{likes}].
\text{np}(\text{ice\_cream}) & \rightarrow [\text{ice,cream}].
\end{align*}
\]

\[
\begin{align*}
\text{saturate1}(P,X) & :- \text{arg}(1,P,X).
\text{saturate2}(P,X) & :- \text{arg}(2,P,X).
\text{subset}([],\_) & :-
\text{subset}([X|L1],L2) & :- \text{member}(X,L2), \text{subset}(L1,L2).
\text{member}(X,[X|\_]) & :-
\text{member}(X,[\_|L]) & :- \text{member}(X,L).
\text{predicate1}((\text{findall}(X,P,\_),\_),P) & :-
\text{saturate1}(P,X).
\text{predicate2}((\_,(\text{findall}(X,P,\_),\_)),P) & :-
\text{saturate1}(P,X).
\end{align*}
\]

?- s(M,[every,woman,likes,ice,cream],[]).
M = \text{findall}(_A,\text{woman}(_A),_B),\text{findall}(_C,\text{likes}(_C,\text{ice\_cream}),_D),\text{subset}(_B,_D)
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 \text{John}}]] \text{ likes ice cream}
\( ({X: \text{baby}(X)} \cup {X: \text{john}(X)}) \subseteq {Y: \text{likes}(Y, \text{ice_cream})} \)

**note:** set union \((\cup)\) is the translation of “*and*”

Define set union as follows:
\[
\% L1 \cup L2 = L3 \quad \text{“}L3 \text{ is the union of } L1 \text{ and } L2\text{”}
\]
\[
\text{union(L1,L2,L3) :- append(L1,L2,L3).}
\]
Exercise 2

- We can use the rules for every woman and rewrite the rules for John in the same fashion: i.e. as a generalized quantifier:
  - 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].

- For example, we can write something like:
  - namenp((findall(X,P,L1),findall(_Y,_P2,L2),subset(L1,L2))) --> name(P), \{saturate1(P,X)\}.
  - name(john(_)) --> [john].

- Then use this in the grammar:
  - s(M) --> namenp(M), vp(P), \{predicate2(M,P)\}.

- Query:
  - ?- s(M,[john,likes,ice_cream],[[]]).
  - M = findall(_A,john(_A),_B),findall(_C,likes(_C,ice_cream),_D),subset(_B,_D)
Exercise 2

• Question becomes how do we merge the following?
  
  – ?- s(M,[every,woman,likes,ice,cream],[[]]).
  
  – M = findall(_A,woman(_A),_B),findall(_C,likes(_C,ice_cream),_D),subset(_B,_D)
  
  – ?- s(M,[john,likes,ice,cream],[[]]).
  
  – M = findall(_A,john(_A),_B),findall(_C,likes(_C,ice_cream),_D),subset(_B,_D)
  
  – Notice that predicates 2 and 3 (highlighted in blue) are the same
  
• to get:
  
  – every woman and John likes ice cream
  
Exercise 2

- One tool:
  - `predicate1((findall(X,P,_,_),P) :- saturate1(P,X)).`
- can be used to extract the first predicate
- You also have:
  - `predicate2((_,(findall(X,P,_,_),_),P) :- saturate1(P,X)).`
- You could easily write a predicate3 rule
  - note: 3rd predicate is not a `findall`...
- Then you could write a NP conjunction rule like
  - `conjnp(((P1a,P1b,union(L1a,L1b,L1)),P2,P3)) --> qnp(M1), [and], namenp(M2), {... prolog code to pick out and instantiate P1a etc... from M1 and M2}`
Exercise 3

• Other quantifiers can also be expressed using set relations between two predicates:
  Example:
  \[ no: \{X: P_1(X)\} \cap \{Y: P_2(Y)\} = \emptyset \]
  \[ \cap = \text{set intersection} \]
  \[ \emptyset = \text{empty set} \]

  \textit{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

• How to write set intersection?
  – want to define
  – intersect(L1,L2,L3) such that L3 is L1 \cap L2

• From lecture 20
  – subset([],_).
  – subset([X|_ ],L) :- member(X,L).
  – member(X,[X|_ ]).
  – member(X,[ _|L]) :- member(X,L).

• Then using member/2:
  – intersect([],_,[]).
  – intersect([X|L1],L2,L3) :- member(X,L2) -> L3 = [X|L3p], intersect(L1,L2,L3p) ; intersect(L1,L2,L3).