LING 364: Introduction to Formal Semantics

Lecture 25
April 18th
• Homework 5
  – graded and returned
Administrivia

• Today
  – review homework 5
  – also new handout
    • Chapters 7 and 8
    • we’ll begin talking about tense
Exercise 1: Truth Tables and Prolog

Question A: Using

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

for what values of P and Q are P⇒Q and PvQ incompatible?

?- implies(P,Q,R1), or(P,Q,R2), \+ R1=R2.
P = true, Q = false, R1 = false, R2 = true ? ;
P = false, Q = false, R1 = true, R2 = false ? ;
no

Let P⇒Q = R1 implies(P,Q,R1)
PvQ = R2 or(P,Q,R2)

compare values
Homework 5 Review

• **Exercise 1: Truth Tables and Prolog**
• **Question B**
• Define truth table and/3 in Prolog

Truth Table:

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

Prolog Code:

```prolog
% and(P,Q,Result)
and(true,true,true).
and(true,false,false).
and(false,true,false).
and(false,false,false).
```
Homework 5 Review

- **Exercise 1: Truth Tables and Prolog**
- **Question C**
- Show that
  \[ \neg(P \lor Q) = \neg P \land \neg Q \]
  \((De \ Morgan’s \ Rule)\)

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

<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>

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

?- or(P,Q,R1), neg(R1,NR1), neg(P,NP), neg(Q,NQ), and(NP,NQ,R2), \(+\ NR1=R2.

No
Homework 5 Review

• Exercise 1: Truth Tables and Prolog
• Question D
  – Show that
  – ¬(P ∧ Q) = ¬P ∨ ¬Q
  – (another side of De Morgan’s Rule)
• Question C was for
• ¬(P ∨ Q) = ¬P ∧ ¬Q

?- and(P,Q,R1),
  neg(R1,NR1), neg(P,NP),
  neg(Q,NQ), or(NP,NQ,R2),
\+ NR1=R2.
no
Homework 5 Review

- **Exercise 2: Universal Quantification and Sets**
- **Assume meaning grammar:**
  
  $s(M) \rightarrow \text{qnp}(M), \text{vp}(P), \{\text{predicate2}(M,P)\}$.
  $n(\text{woman}(_)) \rightarrow [\text{woman}]$.
  $\text{vp}(M) \rightarrow v(M), \text{np}(X), \{\text{saturate2}(M,X)\}$.
  $v(\text{likes}(_X,_Y)) \rightarrow [\text{likes}]$.
  $\text{np}(\text{ice_cream}) \rightarrow [\text{ice,cream}]$.
  $\text{qnp}(M) \rightarrow q(M), n(P), \{\text{predicate1}(M,P)\}$.
  $q((\text{findall}(_X,P,_,_),_),P) \rightarrow$ 
  
  `subset([],_)`.
  `subset([X|L1],L2) :- member(X,L2), subset(L1,L2)`.
  `member(X,[X|_])`.
  `member(X,[_|L]) :- member(X,L)`.
  `predicate1((\text{findall}(_X,P,_,_),_),P) :-` 
  `\text{saturate1}(P,X)`.
  `predicate2((_,(\text{findall}(_X,P,_,_),_)),P) :-` 
  `\text{saturate1}(P,X)`.

  `\text{every}` has semantics $\{X: P_1(X) \subseteq \{Y: P_2(Y)\}\}$

  every woman likes ice cream $\{X: \text{woman}(X) \subseteq \{Y:\text{likes}(Y,\text{ice_cream})\}\}$

  `--?- s(M,[\text{every},\text{woman},\text{likes},\text{ice,cream}],[])`.

  $M = \text{findall}(A,\text{woman}(A),B), \text{findall}(C,\text{likes}(C,\text{ice_cream}),D), \text{subset}(B,D)$
Homework 5 Review

- **Exercise 2: Universal Quantification and Sets**
- **Questions A and B**
  - John likes ice cream

**Simple way (not using Generalized Quantifiers)**

\[
\text{s}(P) \rightarrow \text{namenp}(X), \text{vp}(P), \{\text{saturate1}(P,X)\};
\text{namenp}(\text{john}) \rightarrow \{\text{[john]}\}.
\]

Note: very different from

\[
\text{s}(M) \rightarrow \text{qnp}(M), \text{vp}(P), \{\text{predicate2}(M,P)\}.
\]

?- s(M,[john,likes,ice,cream],[[]]).
M = likes(john,ice_cream)

?- s(M,[john,likes,ice,cream],[[]]), call(M).
M = likes(john,ice_cream)

**Database**

- woman(mary).
- woman(jill).
- likes(john,ice_cream).
- likes(mary,ice_cream).
- likes(jill,ice_cream).
Homework 5 Review

- **Exercise 2: Universal Quantification and Sets**
- **Question C**
  - (names as Generalized Quantifiers)
  - Every woman and John likes ice cream
  - \(\{X: \text{woman}(X)\} \cup \{X: \text{john}(X)\} \subseteq \{Y: \text{likes}(Y,\text{ice}\_\text{cream})\}\)
  - John and every woman likes ice cream

Treat *John* just like *every*:

```prolog
s(M) --> namenp(M), vp(P), \{predicate2(M,P)\}.
namenp((\text{findall}(X,P,L1),\text{findall}(_Y,\_P2,L2),\text{subset}(L1,L2))) --> \text{name}(P), \{\text{saturate1}(P,X)\}.
name(john(_)) --> [john].
database: john(john)
```

?- s(M,[john,likes,ice,cream],[[]]).
M = findall(A,john(A),B),findall(C,likes(C,ice\_cream),D),subset(B,D))
Homework 5 Review

• Exercise 2: Universal Quantification and Sets
• Question C
  – John and every woman likes ice cream
  – (\{X: john(X)\} \cup \{Y: woman(Y)\}) \subseteq \{Z: likes(Z,ice\_cream)\}

Define conjnp:
\[ s(M) \rightarrow \text{conjnp}(M), \text{vp}(P), \{\text{predicate2}(M,P)\}. \]
\[ \text{conjnp}(((\text{findall}(X,P1,L1),\text{findall}(Y,P2,L2),\text{union}(L1,L2,L3)\),\text{findall}(\_,\_,L4),\text{subset}(L3,L4))) \rightarrow \]
\[ \text{namenp}(M1), [\text{and}], \text{qnp}(M2), \{\text{predicate1}(M1,P1), \text{predicate1}(M2,P2), \]
\[ \text{saturate1}(P1,X), \text{saturate1}(P2,Y)\}. \]

?- s(M,[john,and,every,woman,likes,ice,cream],[]).
M = (\text{findall}(A,john(A),B), \text{findall}(C,woman(C),D),\text{union}(B,D,E)), \]
\[ \text{findall}(F,\text{likes}(F,ice\_cream),G),\text{subset}(E,G) \]
Homework 5 Review

• Exercise 3: Other Generalized Quantifiers

• Question A

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

– No woman likes ice cream

\[
\text{qnp}(M) \rightarrow q(M), n(P), \{\text{predicate1}(M,P)\}.
\]

\[
\text{q}((\text{findall}_{-X,-P1,L1}, \text{findall}_{-Y,-P2,L2}, \text{subset}(L1,L2))) \rightarrow \text{[every]}.
\]

\[
\text{q}((\text{findall}_{-X,-P1,L1}, \text{findall}_{-Y,-P2,L2}, \text{intersect}(L1,L2,[]))) \rightarrow \text{[no]}.
\]

?- s(M,[no,woman,likes,ice,cream],[]).
M = findall(_A,woman(_A),_B), findall(_C,likes(_C,ice_cream),_D), intersect(_B,_D,[])
?- s(M,[no,woman,likes,ice,cream],[]), call(M).
no
Homework 5 Review

• Exercise 3: Other Generalized Quantifiers
• Question A

  some: \( \{X: P_1(X)\} \cap \{Y: P_2(Y)\} \neq \emptyset \)
  
  – Some women like ice cream \(\text{(plural agreement)}\)
  – *Some woman likes ice cream

\[
\text{qnp}(M) \rightarrow q(M), n(P), \{\text{predicate1}(M,P)\}.
\]
\[
q((\text{findall}(_X,_P1,L1),\text{findall}(_Y,_P2,L2),\text{subset}(L1,L2))) \rightarrow \text{[every]}.\]
\[
q((\text{findall}(_X,_P1,L1),\text{findall}(_Y,_P2,L2),\text{intersect}(L1,L2,L3),\text{\textbackslash+L3=[]})) \rightarrow \text{[some]}.\]

Don’t have to implement agreement in this exercise, you could just add:
\[
n(\text{woman}(_)) \rightarrow \text{[women]}.\]
\[
v(\text{likes}(_X,_Y)) \rightarrow \text{[like]}.\]
Homework 5 Review

- **Exercise 3: Other Generalized Quantifiers**
- **Question A**
  
  \[\text{some:}\ \{X: P_1(X)\} \cap \{Y: P_2(Y)\} \neq \emptyset\]
  
  - Some women like ice cream  
  - *Some woman likes ice cream  

\[
\text{qnp}(M) \rightarrow q(M), n(P), \{\text{predicate1}(M,P)\}.
\]
\[
q((\text{findall}(_X,_P1,L1),\text{findall}(_Y,_P2,L2),\text{subset}(L1,L2))) \rightarrow [\text{every}].
\]
\[
q((\text{findall}(_X,_P1,L1),\text{findall}(_Y,_P2,L2),\text{intersect}(L1,L2,L3),\+L3=[[]))) \rightarrow [\text{some}].
\]

?- s(M,[some,women,like,ice,cream],[]), call(M).
M =
findall(_A,woman(_A),[mary,jill]),findall(_B,likes(_B,ice_cream),[john,mary,jill]),i
ntersect([mary,jill],[john,mary,jill],[mary,jill]),\+[mary,jill]=[]
• Chapter 8: Tense, Aspect and Modality
Tense

- Formal tools for dealing with the semantics of tense (Reichenbach):
  - use the notion of an **event**
  - relate
    - **utterance** or **speech time** (S),
    - **event time** (E) and
    - **reference** (R) *aka topic time* (T)
  - S, E and T are **time intervals**:
    - think of them as time lines
    - equivalently, infinite sets containing points of time
  - examples of relations between intervals:
    - precedence (<), inclusion (⊆)
Past Tense

• Example:
  – (16) Last month, I went for a hike
  – S = utterance time
  – E = time of hike

• What can we infer about event and utterance times?
  – E is within the month previous to the month of S
  – (Note: E was completed last month)

• Tense (went)
  – past tense is appropriate since E < S

• Reference/Topic time?
  – *may not seem directly applicable here*
  – T = last_month(S)
  – *think of last_month as a function that given utterance time S*
  – *computes a (time) interval*
  – name that interval T
Past Tense

• Example:
  – (16) Last month, I went for a hike

• What can we infer?
  – $T =$ reference or topic time
  – $T =$ last_month$(S)$
  – $E \subseteq T$
  – $E$ is a (time) interval, wholly contained within or equal to $T$

• Tense (went)
  – past tense is appropriate when
  – $T < S$, $E \subseteq T$
Past Tense

• Example:
  – (17) Yesterday, Noah had a rash

• What can we infer?
  – \( T = \text{yesterday}(S) \)
  – “yesterday” is relative to utterance time (S)
  – \( E = \text{interval in which Noah is in a state of having a rash} \)
  – \( E \) may have begun before \( T \)
  – \( E \) may extend beyond \( T \)
  – \( E \) may have been wholly contained within \( T \)
    – \( E \cap T \neq \emptyset \)

• Tense (had)

  – appropriate since \( T < S, E \cap T \neq \emptyset \)
Simple Present Tense

• In English
  (18a) Mary runs \((\text{simple present})\)
  has more of a \textit{habitual} reading
  – does \textbf{not} imply
    (18b) Mary is running \((\text{present progressive})\)
    – \(T = S, \text{run}(\text{mary}) \text{ true } @ T\)
      \(\text{true } @ T = \text{“at time } T\)"
      • i.e. \textit{Mary is running right now at utterance time}
    – (cf. past: \(T < S\))

• However, the simple present works when we’re talking about “states”

• Example: \((\text{has})\)
  – (18c) Noah has a rash \((\text{simple present})\)
  – rash(\text{noah}) \text{ true } @ T, T=S
  – i.e. Noah has the property of having a rash right now at utterance time

English simple present tense:
\(T=S, E\ \text{has a stative interpretation, } E \cap T \neq \emptyset\)
Simple Present Tense

• Some exceptions to the stative interpretation idea
• Historical Present
  – present tense used to describe past events
  – Example:
    • (19a) This guy comes up to me, and he says, “give me your wallet”
    • cf. This guy came up to me, and he said...
• Real-time Reporting
  – describe events concurrent with utterance time
  – Example:
    • (19b) She kicks the ball, and – it’s a goal!
    • cf. She is kicking the ball