Discourse
BSc Artificial Intelligence, Spring 2011

Raquel Fernández

Institute for Logic, Language & Computation
University of Amsterdam
Plan for Today

- Presupposition
- Topics for final projects
What’s a presupposition?

A presupposition is **background information** that is **taken for granted**. But not any kind of background information . . .

Background information, but not presuppositions in the technical senses
John wrote Harry a letter, presupposing he could read.
The theory of evolution presupposes a vast time scale.
Adolph addressed the butler as “sir”, presupposing he was the host Sir Ansel himself

Presuppositions
Martha regrets drinking John’s home brew. \( \leadsto \) Martha drank John’s home brew.
Confirm your eBay transaction (*spam email*) \( \leadsto \) you have done an eBay transaction
Carter returned to power. \( \leadsto \) Carter held power before.

Technically, a presupposition of an utterance \( U \) is background information that:

- must be assumed by the speaker and the addressee of \( U \) for \( U \) to be considered appropriate in a context;
- is associated with a specific lexical item or construction in \( U \);
- will remain a necessary assumption even if \( U \) is negated.
Presupposition vs. Entailment

Presuppositions behave differently from entailments and in that sense they are not part of the regular meaning of a sentence.

- **Semantic entailment**: \( A \) entails \( B \) \((A \models B)\) iff every situation that makes \( A \) true makes \( B \) true.

Mary’s dog was killed in an accident
\( \models \) Mary’s dog is dead
\( \rightarrow \) Mary has a dog

John managed to stop in time
\( \models \) John stopped in time
\( \rightarrow \) John tried to stop in time.

Mary’s dog was not killed in an accident
\( \not\models \) Mary’s dog is dead
\( \leftrightarrow \) Mary has a dog

John didn’t managed to stop in time
\( \not\models \) John stopped in time
\( \leftrightarrow \) John tried to stop in time

- One way to define presupposition is thus as follows:

\( A \) presupposes \( B \) iff

\( \star \) in all situations where \( A \) is true, \( B \) is true
\( \star \) in all situations where \( A \) is false, \( B \) is true
Presupposition Triggers (1)

Where do presuppositions come from? Researchers have identified a long list of presupposition triggers. Here are some of them:

- **Definite descriptions and proper names**
  
  John saw the man with two heads.
  
  $\leadsto$ there exists someone called John / there exists a man with two heads.

  The couple that won the dance contest was please.
  
  $\leadsto$ there was a dance contest / there is a couple that won it

- **Possessive case**

  Marta’s dog is tiny. $\leadsto$ Marta has a dog.
  
  Jody loves her husband. $\leadsto$ Jody has a husband / is married.

- **Factive / implicatives/ change of state verbs**

  Vincent regrets/knows/realized that Mia is married. $\leadsto$ Mia is married
  
  John forgot to open the door. $\leadsto$ John ought to / intended to lock.
  
  John stopped / continued drinking. $\leadsto$ John had been drinking.
Presupposition Triggers (2)

- **Iteratives**
  Sue took the class too. $\rightarrow$ Someone else took the class
  I would do it again $\rightarrow$ I did it before

- **Questions**
  Who broke the window? $\rightarrow$ Someone broke the window.
  Where / Why did you murder Prof. Jones? $\rightarrow$ you murdered Prof. Jones

- **Comparisons and contrasts**
  Marianne called Adolph a male chauvinist, and then HE insulted HER.
  $\rightarrow$ For Marianne to call Adolph a male chauvinist would be to insult him.
  Carol is a better linguist than Barbara. $\rightarrow$ Barbara is a linguist.

For more lexical items and constructions that are typically considered presupposition triggers see, e.g., the Presupposition entry in the Stanford Encyclopedia of Philosophy:
http://plato.stanford.edu/entries/presupposition/
Presupposition Projection

If a complex sentence contains presupposition triggers, are the presuppositions inherited by the complex sentence (*projected*)?

The *projection problem*: presuppositions survive in some contexts where entailments do not, but they disappear in some other contexts.

These are the main *projection tests* to check whether something is a presupposition:

- **Negation**: if $S$ presupposes $P$, then $\neg S$ presupposes $P$ as well.
- **Questions**: if $S$ presupposes $P$, then $P$ is a presupposition of the interrogative version of $S$ as well.

| Ed realizes that it is Wednesday. | N: Ed doesn’t realize that it is Wednesday. |
| It is Wednesday | Q: Does Ed realize that it is Wednesday? |
| ~⇒ It is Wednesday | ~⇒ It is Wednesday |
Presupposition Projection

• Conditionals

  * If \( p \) is a presupposition of sentence \( S \), then \( p \) is a presupposition of any sentence of the form \( \text{if } S, \text{ then } S' \).

  
  If Sam stopped smoking, then his marathon time should improve.
  \rightarrow Sam used to smoke.

  * If \( p \) is a presupposition of sentence \( S' \) and \( p \) is entailed by sentence \( S \), then \( p \) is not a presupposition of any sentence of the form \( \text{if } S, \text{ then } S' \).

  
  If Sam once smoked, then he has stopped smoking.
  \leftarrow Sam used to smoke.

Since the presupposition itself is hypothetical (stated in the antecedent of the conditional) it can’t be taken for granted.

The latter example shows that in some contexts presuppositions do not project. There are other contexts in which presuppositions are cancelled...
Presupposition Cancellation

Presuppositions are defeasible. Unlike entailments, they can be overtly denied:

John didn’t manage to pass his exam
→ John tried to pass his exam.
|= John didn’t pass his exam.

John didn’t manage to pass his exam, in fact he didn’t even try. ✓
John didn’t manage to pass his exam, in fact he passed it. ×
[it is not the case that he didn’t pass it]

Sometimes background knowledge cancels a presupposition. Consider the following example:

Sue cried before she finished her thesis.
→ Sue finished her thesis
Sue died before she finished her thesis.
→ Sue finished her thesis

Here the entailments of the sentence (event precedence) and background assumptions (about what is possible after death) clash with the presupposition, which leads to cancellation.
Accommodation

Presuppositions need to be compatible with the background assumptions of the interlocutors.

- If a presupposition is compatible with the hearer’s knowledge but not yet part of it, a hearer will tend to add it – accommodate it.
- For instance, the following sentence is felicitous even if the presupposition is not part of the knowledge of the hearer:

> Sorry, I’m late. My son was ill this morning. ⇝ the speaker has a son.

However, not all presupposition allow easy accommodation:

> John had dinner in New York last night, too.  
> ⇝ someone else had dinner in NY last night.  
> [surely part of anyone’s knowledge]

What seems to be needed for the sentence to be felicitous is that somebody relevant to the interlocutors had dinner in New York last night, and that this has been mentioned in the previous discourse.
Presuppositions in DRT

How can we deal with presupposition within the DRT framework? The best well-known approach is due to Rob van der Sandt.


- Like pronouns, presuppositions are anaphoric expressions: they require the presence of an antecedent in the discourse context.
- Presuppositions have more descriptive content than pronouns, and hence do not only introduce discourse referents, but full DRSs that include conditions.
- B&B (Bos 2003) mark presuppositional DRSs with an $\alpha$ operator:

  \[
  \begin{array}{c}
  \text{the woman collapses} \\
  \times \\
  \text{WOMAN}(x)
  \end{array} \quad \alpha_x \quad \begin{array}{c}
  \text{there is a woman} \\
  \text{COLLAPSE}(x)
  \end{array}
  \]
Presupposition Triggers in Prolog

The implementation uses the alfa operator (that we used to mark pronouns) to mark presupposition triggers in the lexicon and indicate what they presuppose. These are the macros for personal pronouns, the definite article, and possessive pronouns:

**Semantic Macros in** `SemLexPresupDRT.pl`

```prolog
semLex(pro,M):-
    M = [symbol:Sym,
         sem:lam(P,alfa(pro,drs([X],[pred(Sym,X)]),app(P,X)))].
semLex(det,M):-
    M = [type:def,
         num:sg,
         sem:lam(U,lam(V,alfa(def,merge(drs([X],[]),app(U,X)),app(V,X)))]).
semLex(det,M):-
    M = [type:poss(Symbol),
         num:sg,
         sem:lam(U,lam(V,alfa(pro,drs([Y],[pred(Symbol,Y)]),
                                       alfa(def,merge(drs([X],[rel(of,X,Y)]),app(U,X)),app(V,X))))].
```
Presupposition Resolution

Once we have computed the presuppositions of a sentence (taking into account the information in the lexicon and using semantic composition), we need to resolve them.

This can be done via binding (linking) or accommodation. Binding is equivalent to pronouns resolution: for each $\alpha$-DRS, we need to find an antecedent that is structurally accessible and compatible, and bind the anaphoric referent to the antecedent.

A woman snorts. She collapses

| $y$ | WOMAN(y)  
|-----|-----------
| $\vee$ | SNORT(y) |

\[ y \times \]

| $x$ | WOMAN(y)  
|-----|-----------
| $\vee$ | SNORT(y) |

\[ x = y \]

| | COLLAPSE(x) |

Result in both cases:

<table>
<thead>
<tr>
<th>$y \times$</th>
</tr>
</thead>
</table>
| WOMAN(y)  
| SNORT(y)  
| WOMAN(x)  
| COLLAPSE(x) |

$\alpha_x$

A woman snorts. The woman collapses

| $y$ | WOMAN(y)  
|-----|-----------
| $\vee$ | SNORT(y) |

\[ y \times \]

\[ \left. \frac{x}{WOMAN(x)} \middle| \right. \]

| $\alpha_x$ | COLLAPSE(x) |

After unification:

| $y$ | WOMAN(y)  
|-----|-----------
| $\vee$ | SNORT(y)  
| COLLAPSE(y) |
Accommodation

If a presupposition cannot be bound to an accessible antecedent, we may resort to accommodation: we simply add the presupposed information to the discourse context.

Accommodation follows certain constrains:

• Given the DRS structure, there may be several accommodation sites.
• Free variable check: accommodation should not lead to free discourse referents.
• Consistency: presuppositions need to be consistent with the other information conveyed by the sentence and with the previous discourse.
• Informativity: presuppositions should not be redundant, else there is no need to accommodate.

⇒ We’ll discuss accommodation further in the next class.
Implementation: Binding & Accommodation

The main level program is `presupDRT.pl`, which integrates both pronoun resolution and presupposition resolution.

- it first uses the grammar to build a representation that includes merge and alpha operators with `t/3`
- it then does merge reduction and pronoun/presupposition resolution with `resolveDrs/2` by binding alpha referents to accessible referents or, for presuppositions, performing accommodation when binding fails.

```prolog
?- presupDRT:t([sem:Drs], [mia, loves, her, husband], []).  
Drs = alfa(nam,drs([A],[pred(mia,A)]),alfa(pro,drs([X],[pred(female,X)])), alfa(def, merge(drs([B], [rel(of, B, X)]), drs([], [pred(husband, B)]))), merge(drs([C], [pred(love, C), rel(agent, C, A), rel(patient, C, B), pred(nonreflexive, C)]), drs([], [pred(event, C)]))) .

?- presupDRT.
> Mia loves her husband.
1 drs([A, B, C], [pred(female, A), pred(mia, A), rel(of, B, A), pred(husband, B), pred(love, C), rel(agent, C, A), rel(patient, C, B), pred(nonreflexive, C), pred(event, C)])```
What’s Next

• Next week (April 26) there will be no lecture.
• Get in touch with me by Friday 29 at the latest to discuss your preferences for a final project topic (send me an email or possibly make an appointment).
• On Tuesday 3 May, we will probably have a practical session – more details to be announced.
• Tuesday 10 May: wrap-up session, including report on your ongoing work in the projects.
• Friday 20 May: final projects due (written report + code, etc.).
• Monday 23 May: presentation of final projects.