Dynamic Semantics

James Pustejovsky

Brandeis University

October 29, 2024

Sentence spanning anaphora

Example:

- There is a unicorn in the garden. It is eating the flowers.

The logical representation we want to build:

- $\exists x.(\mathit{unicorn}\ x) \wedge (\mathit{inTheGarden}\ \ x) \wedge ((\mathit{eat}\ \mathit{flowers})\ x)$

Sentence spanning anaphora

Example:

- There is a unicorn in the garden. It is eating the flowers.

The logical representation we want to build:

- $\exists x.(\mathit{unicorn}\ x) \wedge (\mathit{inTheGarden}\ \ x) \wedge ((\mathit{eat}\ \mathit{flowers})\ x)$

The logical representations we can build:

- $\exists x.(\mathit{unicorn}\ x) \wedge (\mathit{inTheGarden}\ \ x)$
- $((\mathit{eat}\ \mathit{flowers})\ x)$

Sentence spanning anaphora

Example:

- There is a unicorn in the garden. It is eating the flowers.

The logical representation we want to build:

- $\exists x.(\mathit{unicorn}\ x) \wedge (\mathit{inTheGarden}\ \ x) \wedge ((\mathit{eat}\ \mathit{flowers})\ x)$

The logical representations we can build:

- $\exists x.(\mathit{unicorn}\ x) \wedge (\mathit{inTheGarden}\ \ x)$
- $((\mathit{eat}\ \mathit{flowers})\ x)$

So, quantifiers should dynamically extend their scope from one sentence to another.

Donkey sentences

Example:

- Every [farmer [who owns a donkey]] feeds it.

Donkey sentences

Example:

- Every [farmer [who owns a donkey]] feeds it.

The logical representation we want to build:

- $\forall x \forall y.(\mathit{farmer}\ x) \wedge (\mathit{donkey}\ y) \wedge ((\mathit{own}\ y)\ x) \rightarrow ((\mathit{feed}\ y)\ x)$

Donkey sentences

Example:

- Every [farmer [who owns a donkey]] feeds it.

The logical representation we want to build:

- $\forall x \forall y.(\mathit{farmer}\ x) \wedge (\mathit{donkey}\ y) \wedge ((\mathit{own}\ y)\ x) \rightarrow ((\mathit{feed}\ y)\ x)$

But:

- We would translate existential NPs (like *a donkey*) using $\exists$, not $\forall$.
- The donkey quantifier occurs inside the relative clause but needs to take scope over the matrix clause.

The dynamic turn

- **Static semantics:**
  Focus is on sentences. They express truth-conditions.
- **Dynamic semantics:**
  Focus is on discourses. Sentences are instructions for updating a discourse representation. Each new sentence of a discourse is interpreted in the context provided by the sentences preceding it.

The dynamic turn

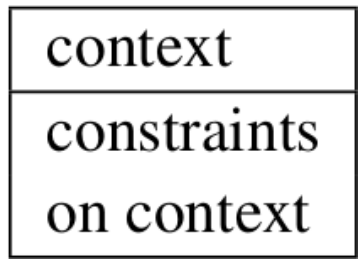
- In other words
  - **Static semantics:**
    Meaning is about truth conditions
  - **Dynamic semantics:**
    Meaning is about context change potential

Discourse Representation Theory

- Developed independently by Hans Kamp (1981) and Irene Heim (1982–as file change semantics)
- "A DRT-style representation for a piece of text consists of a context, plus a list of constraints on that context."

Discourse Representation Theory

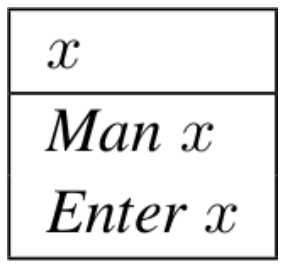
- "In the characteristic box notation of DRT this looks like:"



- "In DRT, the context consists of a list of reference markers or discourse referents.
- The constraints are assertions about these markers.
- Together they represent the information that a text provides, plus information about the anaphoric possibilities of the text."

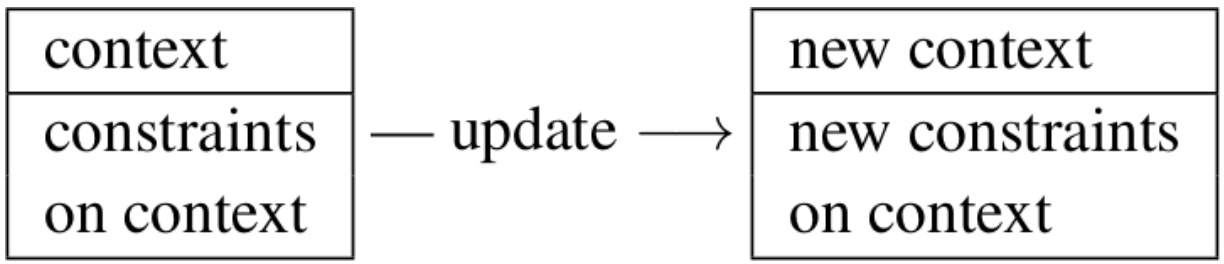
Discourse Representation Theory

- (5.1) A man entered.
- (5.2) $\exists x.(\text{Man}(x) \wedge \text{Enter}(x))$



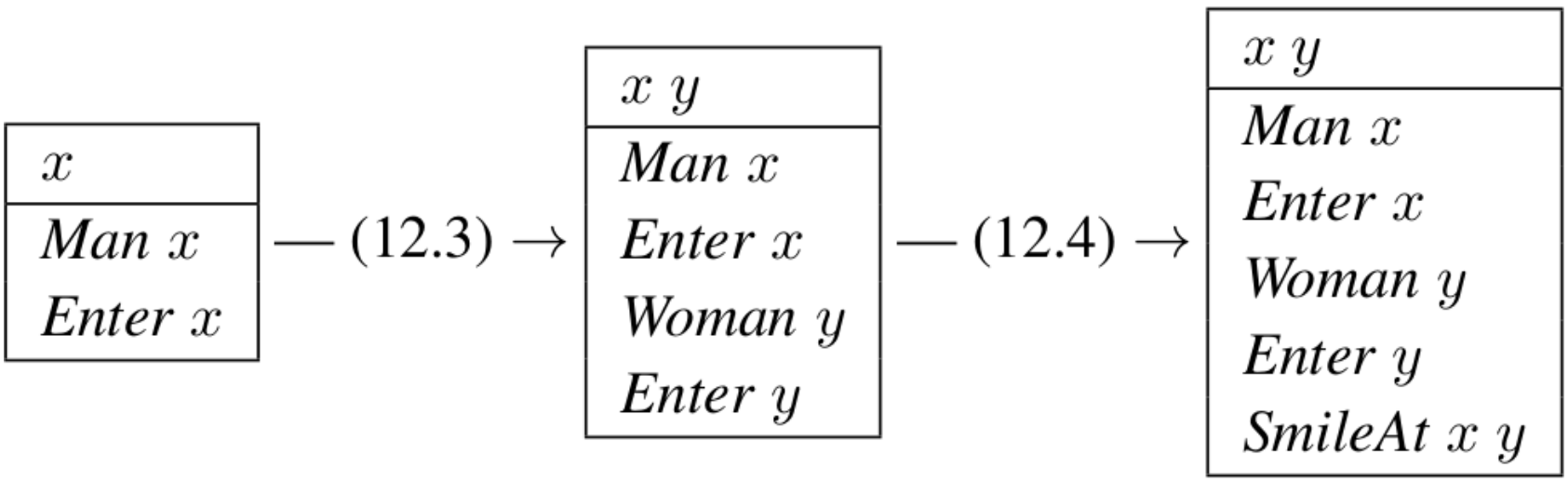
Discourse Representation Theory

- "As the information conveyed by a piece of text grows, the corresponding representation structures get 'updated'. This happens roughly as follows:"



Discourse Representation Theory

- (5.3) A woman entered.
- (5.4) He smiled at her.



Discourse Representation Theory

- How to formalize this?
  - In particular, how to formalize this in a compositional way?

  - Henk Zeevat (1989): A Compositional Approach to Discourse Representation Theory. (cf. Chapter 12.1)
  - Jeroen Groenendijk and Martin Stokhof (1991): Dynamic predicate logic.

Beyond DRT and DPL

- Jan van Eijck (2001): Incremental dynamics. (cf. Chapter 12)
  - Sentence meanings are transitions from an input context to an output context.
  - Contexts are lists of entities.
  - Existential NPs introduce new entities and add them to the context, while pronouns pick entities from the context.

Beyond DRT and DPL

- Jan van Eijck (2001): Incremental dynamics. (cf. Chapter 12)
  - Sentence meanings are transitions from an input context to an output context.
  - Contexts are lists of entities.
  - Existential NPs introduce new entities and add them to the context, while pronouns pick entities from the context.
- Philippe de Groote (2006): Towards a Montegovian acount of dynamics.

  `http://research.nii.ac.jp/salt16/proceedings/degroote.new.pdf`
  - Goal: provide Montague semantics with an appropriate notion of context
  - A sentence is interpreted w.r.t. both its left context (made of the sentences preceding it) and its right context (made of the sentences following it).
  - These two kinds of contexts are abstracted over the meaning of the sentences.

Typing left and right contexts

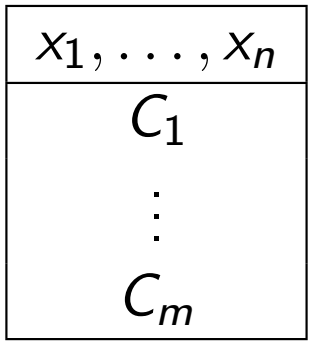
Types

$$\tau ::= e \mid t \mid \gamma \mid \tau \to \tau$$

- Left context: γ (e.g. a set of entities)
- Right context: $\gamma \to t$
- $[\![S]\!] :: \gamma \to (\gamma \to t) \to t$

Connection to DRT

Consider a DRS:



It corresponds to the following λ-expression:

$$\lambda c_L \lambda c_R . \exists x_1 \ldots x_n . C_1 \wedge \ldots \wedge C_m \wedge (c_R \ \ (c_L \cup \{x_1, \ldots, x_n\}))$$

Updating and accessing the context

- empty context $\texttt{nil} :: \gamma$
- a function $\texttt{push} :: e \to \gamma \to \gamma$ for adding an entity to a context
- a selection function $\texttt{sel} :: \gamma \to e$ that selects an entity from a context

Names and existential NPs introduce entities into the context, that pronouns can pick up later.

Updating and accessing the context

- empty context $\texttt{nil} :: \gamma$
- a function $\texttt{push} :: e \to \gamma \to \gamma$ for adding an entity to a context
- a selection function $\texttt{sel} :: \gamma \to e$ that selects an entity from a context

Names and existential NPs introduce entities into the context, that pronouns can pick up later.

Example:

- $[\![\text{John admires Mary}]\!]$
  $= \lambda c_L \lambda c_R.((\mathit{admire}\ m)\ j) \wedge (c_R\ (\texttt{push}\ m\ (\texttt{push}\ j\ c_L)))$
- $[\![\text{He smiles at her}]\!]$
  $= \lambda c_L \lambda c_R.((\mathit{smile}\ (\texttt{sel}\ c_L))\ (\texttt{sel}\ c_L)) \wedge (c_R\ c_L)$

Composition of sentence interpretations

$$[\![S_1 \,.\, S_2]\!] = \lambda c_L \lambda c_R.(([\![S_1]\!]\ c_L)\ \ \lambda c'_L.(([\![S_2]\!]\ c'_L)\ c_R))$$

Example:

- $[\![$John admires Mary . He smiles at her$]\!]$
  $= \lambda c_L \lambda c_R.(([\![$John admires Mary$]\!]\ c_L)\ \ \lambda c'_L.(([\![$He smiles at her$]\!]\ c'_L)\ c_R))$
  $= \lambda c_L \lambda c_R.((\mathit{admire}\ m)\ j) \wedge ((\mathit{smile}\ m)\ j) \wedge (c_R\ (\mathtt{push}\ m\ (\mathtt{push}\ j\ c_L)))$

Lexical expressions

	old type	new type
sentence	t	$\gamma \to (\gamma \to t) \to t \ (= t^*)$
noun	$e \to t$	$e \to t^*$
noun phrase	$(e \to t) \to t$	$(e \to t^*) \to t^*$

Lexical expressions

	old type	new type
sentence	t	$\gamma \to (\gamma \to t) \to t \ (= t^*)$
noun	$e \to t$	$e \to t^*$
noun phrase	$(e \to t) \to t$	$(e \to t^*) \to t^*$

- **Nouns**:: $e \to t^*$

$$[\![\text{unicorn}]\!] = \lambda x \lambda c_L \lambda c_R . (\mathit{unicorn}\ x) \wedge (c_R\ c_L)$$

Lexical expressions

	old type	new type
sentence	t	$\gamma \to (\gamma \to t) \to t \ (= t^*)$
noun	$e \to t$	$e \to t^*$
noun phrase	$(e \to t) \to t$	$(e \to t^*) \to t^*$

- **Nouns**:: $e \to t^*$

  $[\![\text{unicorn}]\!] = \lambda x \lambda c_L \lambda c_R.(\mathit{unicorn}\ x) \wedge (c_R\ c_L)$

- **Noun phrases**:: $(e \to t^*) \to t^*$

  $[\![\text{John}]\!] = \lambda P \lambda c_L \lambda c_R.(((P\ j)\ c_L)\ \lambda c'_L.(c_R\ (\mathtt{push}\ j\ c'_L)))$
  $[\![\text{he}]\!] = \lambda P \lambda c_L \lambda c_R.(((P\ (\mathtt{sel}\ c_L))\ c_L)\ c_R)$

Lexical expressions

	old type	new type
sentence	t	$\gamma \to (\gamma \to t) \to t \ (= t^*)$
noun	$e \to t$	$e \to t^*$
noun phrase	$(e \to t) \to t$	$(e \to t^*) \to t^*$

- **Nouns**:: $e \to t^*$

  $[\![\text{unicorn}]\!] = \lambda x \lambda c_L \lambda c_R.(\mathit{unicorn}\ x) \wedge (c_R\ c_L)$

- **Noun phrases**:: $(e \to t^*) \to t^*$

  $[\![\text{John}]\!] = \lambda P \lambda c_L \lambda c_R.(((P\ j)\ c_L)\ \lambda c'_L.(c_R\ (\mathtt{push}\ j\ c'_L)))$
  $[\![\text{he}]\!] = \lambda P \lambda c_L \lambda c_R.(((P\ (\mathtt{sel}\ c_L))\ c_L)\ c_R)$

- **Transitive verbs**:: $((e \to t^*) \to t^*) \to ((e \to t^*) \to t^*) \to t^*$

  $[\![\text{admires}]\!] = \lambda P \lambda Q.(Q\ \lambda x.(P\ \lambda y \lambda c_L \lambda c_R.((\mathit{admire}\ x)\ y) \wedge (c_R\ c_L)))$

Donkey sentences

- Every farmer who owns a donkey feeds it.
- $\lambda c_l \lambda c_R.\forall x.((\mathit{farmer}\ x) \rightarrow \forall y.((\mathit{donkey}\ y) \wedge ((\mathit{own}\ y)\ x)) \rightarrow ((\mathit{beat}\ (\mathtt{sel}\ (\mathtt{push}\ x\ (\mathtt{push}\ y\ c_L)))\ x)) \wedge (c_R\ c_L)$

The entities introduced by *every farmer* and *a donkey* are not available outside this sentence because they are pushed onto the local context (the continuation of the sentence) and not onto the global context (the continuation of the discourse).