EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Semantics & Pragmatics SoSe 2021

Lecture 15: Discourse Representation Theory II

29/06/2021, Christian Bentz

Overview

Section 1: Recap of Lecture 14

Section 2: Syntax of the DRS Language
- Formal Definition
- Different Notations
- DRS Merge

Section 3: Accessibility
- Formal Definition
- Negation
- Disjunction
- Conditional
- Quantification

Section 4: The Semantics of the DRT Language

Summary

References

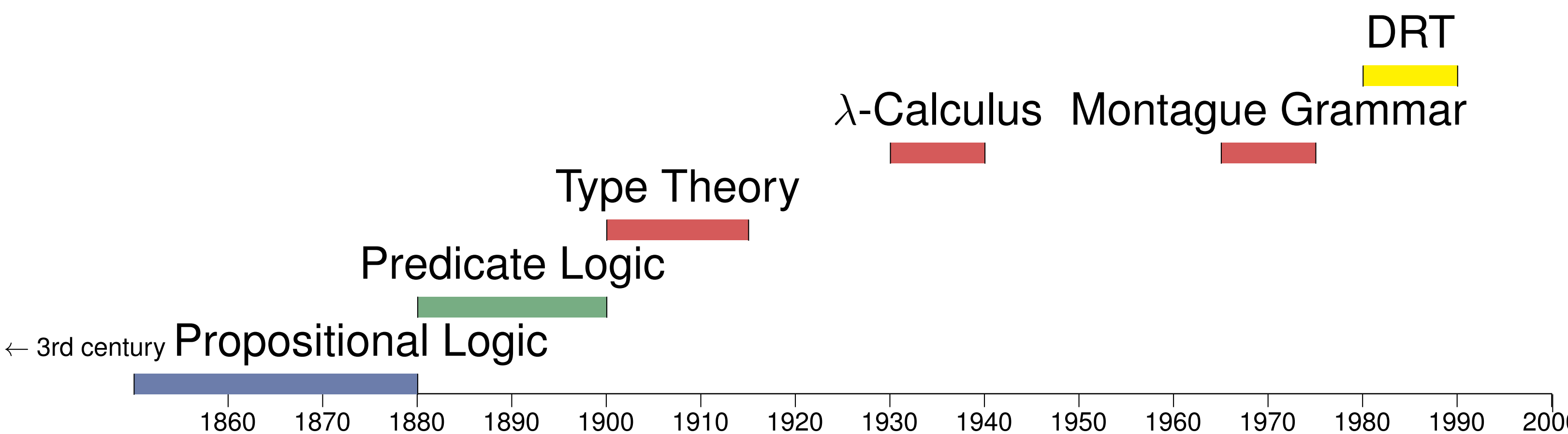
Section 1: Recap of Lecture 14

Historical Background

"In the early 1980s, **Discourse Representation Theory (DRT)** was introduced by Hans Kamp as a theoretical framework for dealing with issues in the semantics and pragmatics of anaphora and tense (Kamp 1981); a very similar theory was developed independently by Irene Heim (1982)."

Geurts & Beaver (2007), p. 1.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

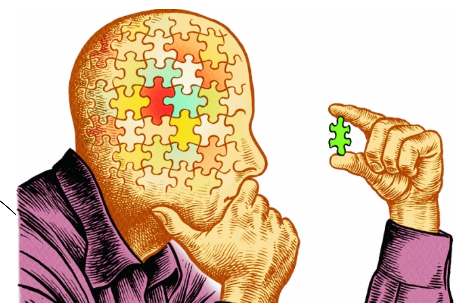


Discourse Representation Structures

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

"DRT's main (and most controversial) innovation [...] is that it introduced a level of mental representations, **called discourse representation structures (DRSs)**. The basic idea [...] is that a hearer builds up a mental representation of the discourse as it unfolds, and that every incoming sentence prompts additions to that representation."

Geurts & Beaver (2007), p. 2.



Bambi$_i$ gave Maya$_j$ flowers$_k$. She$_j$ thanked her$_i$ for them$_k$.

Anaphora Resolution

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

The problem of how hearers are able to **“resolve” anaphora**, e.g. to know which referent (**antecedent**) of the discourse a pronoun (**consequent**) is referring back to, has received attention from both syntacticians and semanticists over the course of centuries. It has resisted straightforward explanations.

If Bambi$_i$ gives Maya$_j$ flowers$_k$ she$_j$ will like them$_k$.

Note: While anaphora resolution across sentences might be considered outside the scope of classical syntax and semantics – as these theories mostly deal with single sentences – the same problems also occur within sentences.

Discourse Representation Structures

A DRS consists of **two major parts**:

1. a set of **discourse referents**,
2. a set of so-called **DRS-conditions** which capture the information about referents that has accummulated over the discourse.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

(1) John chased Jumbo.
[x, y: John(x), Jumbo(y), chased(x,y)]

(2) John chased a donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(3) A farmer chased a donkey.
[x, y: farmer(x), donkey(y), chased(x,y)]

Note: The colon ':' delimits the set of discourse referents from the set of discourse conditions.

Discourse Referents

Discourse referents are a concept similar to the *domain of discourse* in standard logic. However, note that there are *no constants* here, all entities are represented with variables (x, y, etc.). The variables then have to be assigned to proper names, definite noun phrases, indefinite noun phrases via discourse conditions.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

(4) John chased Jumbo.
[x, y: John(x), Jumbo(y), chased(x,y)]

(5) John chased a donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(6) A farmer chased a donkey.
[x, y: farmer(x), donkey(y), chased(x,y)]

Discourse Conditions

Discourse Conditions are then similar to *predicates* in standard logic (but including equations like $x = y$).

(7) John chased Jumbo.
[x, y: John(x), Jumbo(y), chased(x,y)]

(8) John chased a donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(9) A farmer chased a donkey.
[x, y: farmer(x), donkey(y), chased(x,y)]

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Merging Operation

Beyond single sentences (or parts of sentences) discourse structures can be built also for consecutive sentences by **merging** their DRSs using the **$\oplus$-operator**, which is defined as their pointwise union from a set-theoretic perspective.

(10) A farmer chased a donkey.
[x, y: farmer(x), donkey(y), chased(x,y)]

(11) He caught it.
[$\underline{v}$, $\underline{w}$: caught(v, w)]

Geurts & Beaver (2007), p. 7.

Note: The discourse referents of the second sentence are here underlined to indicate that they are in need of antecedents. Geurts & Beaver (2007) do not further explain according to which rules exactly the underlined discourse referents (v, w) are matched with the discourse referents in the former DRS (x, y). In English, this could be done, for instance, via grammatical gender and/or word order.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Complex DRS Conditions: Negation

The above example deals with **simple, i.e. non-embedded DRS conditions**. However, there are various natural language scenarios that require more **complex DRS conditions**, i.e. **embedded** DRS conditions. One such example is **negation**.

(12) John doesn't have a donkey.
$[_1$ x: John(x), $\neg[_2$ y: donkey(y), owns(x,y)]]

(13) It is grey.
[$\underline{z}$: grey(z)]

Geurts & Beaver (2007), p. 7-8.

Note: The negation here scopes over *owns a donkey*, not over *John*. This scope is reflected in the embedded DRS.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Accessibility: Informal Definition

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Every DRS is **accessible** to all and only those DRSs **whose number is bigger or equal to**[1] its own (so every DRS is accessible to itself).

(14) [x, y, $\underline{v}$, $\underline{w}$: farmer(x), donkey(y), chased(x,y), caught(v, w)]

(15) [$_1$ x, $\underline{z}$: John(x), ¬[$_2$ y: donkey(y), owns(x,y)], grey(z)]

Note: The examples are repeated from above. In the first example, all variables have access to all other variables, since they are all part of the same DRS. In the second example, on the other hand, the DRS in [$_1$. . .] is accessible to the DRS in [$_2$. . .], but not the other way around.

[1] There seems to be an error in the formulation by Geurts & Beaver (2007), p. 13. They write "[...] every DRS is accessible to all and only those DRSs whose number *does not exceed* its own." But this seems just the inverse of how accessibility is defined and used in the rest of the paper. Also, the statement does not hold for DRSs connected by logical "or".

Complex DRS Conditions: Conditionals

Similar to negation, **conditionals (material implication)** also gives rise to complex, i.e. embedded DRS structures.

(16) If John owns a donkey, he likes it.
$[_1 : [_2\ x, y: \text{John}(x), \text{donkey}(y), \text{owns}(x,y)] \rightarrow [_3\ \underline{v}, \underline{w}: \text{likes}(v,w)]]$

Note: Geurts & Beaver (2007), p. 8 put *John(x)* outside of $[_2 \ldots]$. However, it is unclear why *John(x)* would not belong to the antecedent of the conditional. In fact, Kamp (2016), p. 13 puts it inside $[_2 \ldots]$. We follow Kamp (2016) here. As to accessibility: The discourse referents x and y are accessible to v and w as before in the case of the conditional.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Complex DRS Conditions: Quantification

Quantification also involves complex DRS conditions. Namely, a quantifier Q over a discourse referent x, i.e. Qx, connects two DRSs, i.e. DRS and DRS′, such that we have DRS(Qx)DRS′. In this respect, **conditionals and quantifiers give rise to essentially the same structure**.

Geurts & Beaver (2007), p. 9.

Section 1: Recap of Lecture 14

Section 2: Syntax of the DRS Language

Section 3: Accessibility

Section 4: The Semantics of the DRT Language

Summary

References

(17) Every farmer who owns a donkey, likes it.

(18) If a farmer owns a donkey, he likes it.

$[_1 : [_2$ x,y: farmer(x), donkey(y), owns(x,y)] $(\forall x)$ $[_3$ $\underline{v}$, $\underline{w}$: likes(v,w)]]

$[_1 : [_2$ x,y: farmer(x), donkey(y), owns(x,y)] $\rightarrow$ $[_3$ $\underline{v}$, $\underline{w}$: likes(v,w)]]

Note: It is (implicitely) assumed here that in (34) we also have a pronoun as the subject of the consequent statements (*likes it*), while it is not explicitly realized here.

Section 2: Syntax of the DRS Language

Formal Definition

"DRSs are **set-theoretic objects** built from **discourse referents** [the set *U*] and **DRS-conditions** [the set *Con*]."

(i) A DRS K is a pair $\langle U_K, Con_K \rangle$, where U_K is a set of discourse referents, and Con_K is a set of DRS-conditions.

(ii) If P is an n-place predicate, and $x_1, \ldots, x_n$ are discourse referents,[2] then $P(x_1, \ldots, x_n)$ is a DRS condition.

(iii) If x and y are discourse referents, then $x{=}y$ is a DRS-condition.

(iv) If K and K' are DRSs, then $\neg K$, $K \rightarrow K'$, and $K \vee K'$ are DRS-conditions.

(v) If K and K' are DRSs and x is a discourse referent, then $K(\forall x)K'$ is a DRS-condition.

Geurts & Beaver (2007), p. 12.

[2] In the actual examples, Geurts & Beaver (2007) do not use variable x with indeces but rather x, y, z, etc.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Clause (i): DRS Basic Structure

(i) A DRS K is a pair $\langle U_K, Con_K \rangle$, where U_K is a set of discourse referents, and Con_K is a set of DRS-conditions.

(19) John chased Jumbo.
[x, y: John(x), Jumbo(y), chased(x,y)]

(20) John chased a donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(21) A farmer chased a donkey.
[x, y: farmer(x), donkey(y), chased(x,y)]

(22) John doesn't have a donkey.
$[_1$ x: John(x), $\neg[_2$ y: donkey(y), owns(x,y)]]

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Clause (ii): Simple Conditions

(ii) If P is an n-place predicate, and $x_1, \ldots, x_n$ are discourse referents, then $P(x_1, \ldots, x_n)$ is a DRS condition.

(23) John chased the donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(24) John chased a donkey.
[x, y: John(x), donkey(y), chased(x,y)]

(25) He caught it.
[v, w: caught(v, w)]

Note: In the DRT framework as outlined here by Geurts & Beaver (2007) – in contrast to standard logic – proper names (*John*), indefinite noun phrases (*a donkey*), and verbs (*chased, caught*) are all considered n-place predicates. *John(x)*, for instance, would translate as "x is a John". However, in Kamp et al. (1995: 145-146) a distinction is drawn between names (*N*) and predicate constants (*P*).

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Clause (iii): Variable Mapping

(iii) If x and y are discourse referents, then x=y is a DRS-condition.

(26) A farmer chased a donkey. He caught it.
[x, y, v, w: v=x, w=y, farmer(x), donkey(y), chased(x,y), caught(v, w)]

(27) If John owns a donkey, he likes it.
$[_1$: $[_2$ x, y, v, w: v=x, w=y, John(x), donkey(y), owns(x,y)] $\rightarrow$ $[_3$: likes(v,w)]]

Note: It should be noted again that the syntactic definition here is silent about how exactly to equate two variables if there are different options as in the examples above.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Clause (iv): Complex Conditions

(iv) If K and K' are DRSs, then $\neg K$, $K \rightarrow K'$, and $K \vee K'$ are DRS-conditions.

(28) John doesn't own a donkey.
[$_1$ x: John(x), ¬[$_2$ y: donkey(y), owns(x,y)]]

(29) If John owns a donkey, he likes it.
[$_1$: [$_2$ x, y: John(x), donkey(y), owns(x,y)] → [$_3$: likes(x,y)]]

(30) John owns a donkey or a horse.
[$_1$ x: John(x), [$_2$ y: donkey(y), owns(x,y)] ∨ [$_3$ y: horse(y), owns(x,y)]]

Note: In the last example involving disjunction, we follow Simons (1996), p. 251, who argues to deal with disjunction by assuming *just one entity y* which is either a donkey or a horse. Also, *John(x)* here has to be *outside* of the two DRSs connected by disjunction.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Clause (v): Quantification

(v) If K and K' are DRSs and x is a discourse referent, then $K(\forall x)K'$ is a DRS-condition.

(31) Every farmer who owns a donkey, likes it.
$[_1 : [_2$ x, y: farmer(x), donkey(y), owns(x,y)] ($\forall$x) $[_3$: likes(x,y)]]

(32) Some farmer who owns a donkey, likes it.
$[_1 : [_2$ x, y: farmer(x), donkey(y), owns(x,y)] ($\exists$x) $[_3$: likes(x,y)]]

Note: While in clause (v) Geurts & Beaver (2007) only define the case of the universal quantifier, at another point they state: "[...] a condition of the form $K(Qx)K'$, where Q may be any quantifier [...]", which suggests that the same definition holds for the existential quantifier.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Different DRS Notations

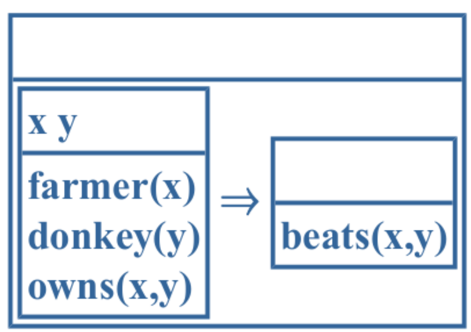
There are (at least) three **different notations** that might be used in DRT frameworks. While Geurts & Beaver (2007) use the so-called **linear notation** (which we are also following), Kamp et al. (1995) use the so-called **box notation**. However, the notation which is closest to the actual mathematical formalization of DRT is the **set-theoretic notation** (called “Official DRS” below).

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Official DRS: **<{}, <{x, y}, {farmer(x), donkey(y), owns(x,y)}> ⇒ <{}, {beats(x,y)}>} >**

Linear notation: **[: [x, y: farmer(x), donkey(y), owns(x,y)] ⇒ [: beats(x,y)]]**

Box notation:



Geurts & Beaver (2007), p. 12.

Merging of DRSs

Given the set-theoretic definition of DRSs, **merging** of two (or more) DRSs (here K and K') is defined as their **pointwise union** ($\oplus$) such that we have

$$K \oplus K' = \langle U_K \cup U_{K'}, Con_K \cup Con_{K'} \rangle. \tag{1}$$

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

(33) A farmer chased a donkey. He caught it.
[x, y: farmer(x), donkey(y), chased(x,y)] $\oplus$ [$\underline{\text{v}}$, $\underline{\text{w}}$: caught(v, w)] $=$
[x, y, $\underline{\text{v}}$, $\underline{\text{w}}$: farmer(x), donkey(y), chased(x,y), caught(v,w)];

such that

$U_K \cup U_{K'} = \{x, y, \underline{v}, \underline{w}\}$
$Con_K \cup Con_{K'} = \{$farmer(x), donkey(y), chased(x,y), caught(v,w)$\}$

Merging of DRSs

The way merging is defined in DRT it follows that there is "no principled distinction between **(clausal) conjunction** and **sentence concatenation**." Therefore, in the syntax of the DRT language, we do not need a definition involving logical "and" ($\wedge$).

Geurts & Beaver (2007), p. 12.

Section 1: Recap of Lecture 14

Section 2: Syntax of the DRS Language

Section 3: Accessibility

Section 4: The Semantics of the DRT Language

Summary

References

(34) A farmer chased a donkey. He caught it.

(35) A farmer chased a donkey **and** he caught it.

Both natural language sentences are equally represented by the DRSs repeated from above:

(36) [x, y: farmer(x), donkey(y), chased(x,y)] $\oplus$ [$\underline{\text{v}}$, $\underline{\text{w}}$: caught(v, w)] $=$
[x, y, $\underline{\text{v}}$, $\underline{\text{w}}$: farmer(x), donkey(y), chased(x,y), caught(v,w)]

Section 3: Accessibility

Accessibility: Formal Definition

"**Accessibility** is a relation between DRSs that is **transitive**[3] and **reflexive**,[4] i.e. it is a preorder. More in particular, it is the smallest preorder for which the following holds, for all DRSs K, K', and K'': if Con_K contains a condition of the form ...

- $\neg K'$, then K is accessible to K',
- $K' \vee K''$, then K is accessible to K' and K'',[5]
- $K' \rightarrow K''$, then K is accessible to K' and K' is accessible to K'',
- $K'(\forall x)K''$, then K is accessible to K' and K' is accessible to K''."

Geurts & Beaver (2007), p. 13.

[3] If a DRS K is accessible to K', and K' is accessible to K'', then K is also accessible to K'', but not the other way around.

[4] Every DRS is accessible to itself.

[5] But note that in this particular case of logical "or", K' is not accessible to K''.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Accessibility: Negation

If Con_K contains a condition $\neg K'$, then K is accessible to K'.

Schematic Representation

$$[_K \ldots \quad \neg[_{K'} \ldots]]$$

Note: The direction of the arrow gives the direction of accessibility, such that K is accessible to K', and variables in K can be used to resolve variables in K'. We can also use numbers here, i.e. 1 and 2, to mark DRSs rather than K, and K', and thus get $[_1 \ldots \ \neg[_2 \ldots]]$.

Examples

(37) John does **not** own a donkey. It is grey.
$[_1$ x, $\underline{z}$: John(x), $\neg[_2$ y: donkey(y), owns(x,y)], grey(z)]

(38) John does own a donkey. It is **not** grey.
$[_1$ x, y, $\underline{z}$: John(x), donkey(y), owns(x,y), $\neg[_2$: grey(z)]]

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Accessibility: Disjunction

If Con_K contains a condition $K' \vee K''$, then K is accessible to K' and K''.

Schematic Representation

$$[_K \ldots \quad [_{K'} \ldots] \vee \; [_{K''} \ldots]]$$

Note: In this particular case of logical “or”, K' is not accessible to K''.

Examples

(39) John owns a donkey or he is unhappy.
[$_1$ x, v: v=x, John(x), [$_2$ y: donkey(y), owns(x,y)] $\vee$ [$_3$: unhappy(v)]]

Note: This natural language construction can only be captured correctly in the DRSs since *John(x)* is in [$_1 \ldots$] rather than [$_2 \ldots$] and hence x is accessible to v. If x was in [$_2 \ldots$] it wouldn't be accessible to v.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Accessibility: Conditional

If Con_K contains a condition $K' \rightarrow K''$, then K is accessible to K' and K' is accessible to K''.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Schematic Representation

$$[_K \ldots \quad [_{K'} \ldots] \rightarrow [_{K''} \ldots]]$$

Note: By transitivity K is also accessible to K''.

Example

(40) If John owns a donkey, he likes it.
[$_1$: [$_2$ x, y, v, w: v=x, w=y, John(x), donkey(y), owns(x,y)] $\rightarrow$ [$_3$: likes(v,w)]]

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Accessibility: Quantification

If Con_K contains a condition $K'(\forall x)K''$, then K is accessible to K' and K' is accessible to K''.

Section 1: Recap of Lecture 14

Section 2: Syntax of the DRS Language

Section 3: Accessibility

Section 4: The Semantics of the DRT Language

Summary

References

Schematic Representation

$$[_K \ldots \quad [_{K'} \ldots] \ (\forall x) \ [_{K''} \ldots]]$$

Note: By transitivity K is also accessible to K''.

Example

(41) Every farmer who owns a donkey, likes it.
[$_1$: [$_2$ x, y, v: v=y, farmer(x), donkey(y), owns(x,y)] ($\forall$x) [$_3$: likes(x,v)]]

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Section 4: The Semantics of the DRT Language

Semantics of the DRT Language: **Model M**

"As usual, a **model M** is a pair $\langle D, I \rangle$, where D is a set of individuals, and I is an interpretation function that assigns sets of individuals to one-place predicates, sets of pairs to two-place predicates, and so on."

Geurts & Beaver (2007), p. 14.

Remember from Lecture on Standard Predicate Logic:

Model M

$D = \{e_1, e_2, e_3\}$
$I = \{\langle j, e_1 \rangle, \langle p, e_2 \rangle, \langle m, e_3 \rangle, \langle S, \{\langle I(j), I(m) \rangle, \langle I(p), I(m) \rangle\}\rangle\}$
$I(S) = \{\langle I(j), I(m) \rangle, \langle I(p), I(m) \rangle\}$
Translation key: j: John; p: Peter; m: morning star; Sxy: x likes y.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Semantics: **Embedding Function**

"The truth-conditional semantics of the DRS language is given by defining when an **embedding function verifies a DRS** in a given model."

Geurts & Beaver (2007), p. 14.

Valuation function $\mathbf{V}_M$ in Standard Predicate Logic

- If $V_M(\phi) = 1$, then ϕ is said to be true in model M.
- If $Aa_1, \ldots, a_n$ is an atomic sentence in L, then $\mathbf{V}_M(Aa_1, \ldots, a_n) = 1$ if and only if $\langle I(a_1), \ldots, I(a_n)\rangle \in I(A)$.
- etc.

Embedding function *f* in DRT

- f verifies a DRS K iff f verifies all conditions Con_K.
- f verifies $P(x_1, \ldots, x_n)$ iff $\langle f(x_1), \ldots, f(x_n)\rangle \in I(P)$.
- etc.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Summary

Summary

- The formal definition of **DRT syntax** consists of a set of clauses which specify the internal structure of a discourse representation structure (DRS), as well as the structure of possible DRS conditions.
- Two further important formal concepts are the **merging** of DRSs, and **accessibility relations** between DRSs, which are important for modelling **anaphora resolution**.
- The **semantics of DRT** is modelled in parallel to the model-theoretic truth-value evaluations of standard predicate logic.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

References

References

Geurts, Bart, & David Beaver (2007). Discourse Representation Theory. In: *The Stanford Encyclopedia of Philosophy*, ed. Edward N.Zalta. CSLI, Stanford University. Online at https://plato.stanford.edu/entries/discourse-representation-theory/.

Kamp, Hans (2016). *Semantics I, From Montague Grammar to DRT*. Lecture at the University of Stuttgart. online at https://www.ims.uni-stuttgart.de/archiv/kamp/files/2016.kamp.semanticsI.montague.drt.pdf.

Kamp, Hans, Genabith, Josef van, & Reyle, Uwe (1995). Discourse Representation Theory. In: Gabbay, D.M., & Guethner, F. (Eds). *Handbook of Philosophical Logic*. 2nd Edition, Volume 15. Heidelberg: Springer.

Simons, Mandy (1996). Disjunction and anaphora. In: Galloway, Teresa & Spence, Justin (Eds.) *SALT VI*. Ithaca, NY: Cornell University, p. 245-260.

Section 1: Recap of Lecture 14
Section 2: Syntax of the DRS Language
Section 3: Accessibility
Section 4: The Semantics of the DRT Language
Summary
References

Thank You.

Contact:

Faculty of Philosophy
General Linguistics
Dr. Christian Bentz
SFS Wihlemstraße 19-23, Room 1.24
chris@christianbentz.de
Office hours:
During term: Wednesdays 10-11am
Out of term: arrange via e-mail