



# **Semantics & Pragmatics SoSe 2023**

## Lecture 15: Discourse Representation Theory II

**27/06/2023, 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

DRT Implementation

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

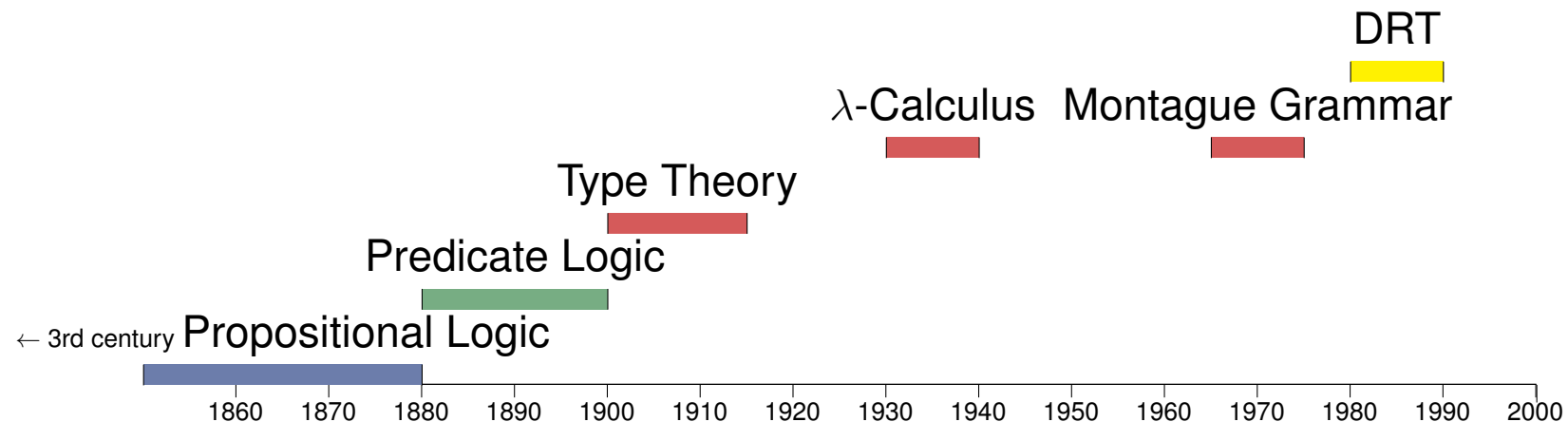
Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References





## Indefinites: A semanticist's riddle



Irene R. Heim



Barbara Partee

*One standard view among logicians is that indefinite noun phrases like ‘a tall man’ are not referring expressions, but quantifier phrases, like ‘every man’, ‘no man’, and ‘most men’. Yet in many respects, indefinite noun phrases seem to function in ordinary language much like definite noun phrases or proper names, particularly with respect to the use of pronouns in discourse. This may be simply a matter of sorting out semantics from pragmatics [...]*

Heim (1982), p. 4 citing Barbara Partee.

- (1) **A dog** came in. **It** lay down under the table.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



# Discourse Representation Structures

“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.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

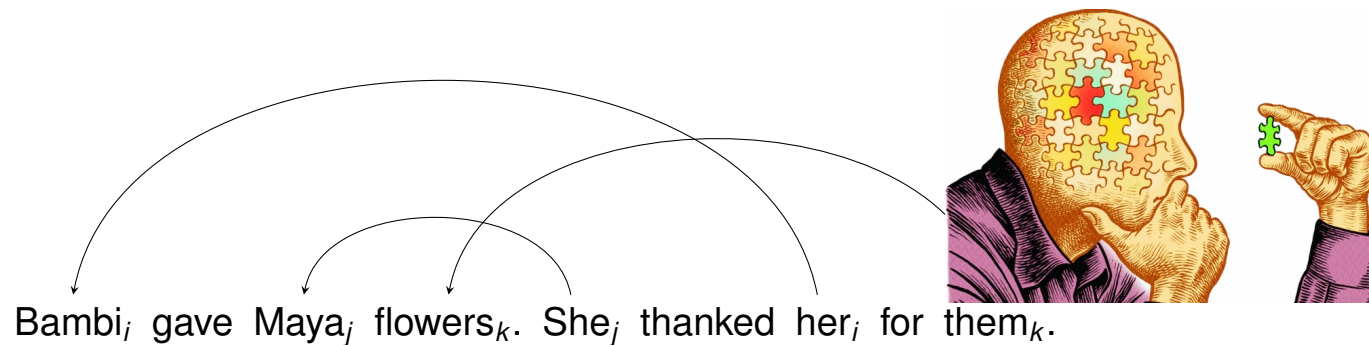
Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References





# Anaphora Resolution

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.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References

If Bambi<sub>i</sub> gives Maya<sub>j</sub> flowers<sub>k</sub> she<sub>j</sub> will like them<sub>k</sub>.

**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 accumulated over the discourse.

(2) John chased Jumbo.

[*x*, *y*: John(*x*), Jumbo(*y*), chased(*x*,*y*)]

(3) John chased a donkey.

[*x*, *y*: John(*x*), donkey(*y*), chased(*x*,*y*)]

(4) A farmer chased the 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.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References





## 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.

- (5) John chased Jumbo.  
[x, y: John(x), Jumbo(y), chased(x,y)]
- (6) John chased a donkey.  
[x, y: John(x), donkey(y), chased(x,y)]
- (7) 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

DRT  
Implementation

Summary

References



## Discourse Conditions

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

(8) John chased Jumbo.

[ $x, y$ : **John(x), Jumbo(y), chased(x,y)**]

(9) John chased a donkey.

[ $x, y$ : **John(x), donkey(y), chased(x,y)**]

(10) 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

DRT  
Implementation

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.

- (11) A farmer chased a donkey.  
 $[x, y: \text{farmer}(x), \text{donkey}(y), \text{chased}(x,y)]$
- (12) He caught it.  
 $[\underline{v}, \underline{w}: \text{caught}(v, w)]$   
 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

DRT  
Implementation

Summary

References



## Example

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

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

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

$$[x, y: \text{farmer}(x), \text{donkey}(y), \text{chased}(x,y), \text{caught}(x,y)]$$

- ▶ The first line is just the original DRSs connected with the  $\oplus$ -operator.
- ▶ In the the second line, all discourse referents which are not already represented in the former DRS are added to the set of discourse referents, and likewise for the discourse conditions (pointwise union).
- ▶ In the third line, discourse conditions are added (equations) to model the mapping of antecedents to consequents.
- ▶ In the last line, these are then “resolved”, i.e. replaced by the original discourse referents  $x$  and  $y$ .

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

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**.

(13) John doesn't own a donkey.

$[_1 x: \text{John}(x), \neg[_2 y: \text{donkey}(y), \text{owns}(x,y)]]$

(14) It is grey.

$[\underline{z}: \text{grey}(z)]$

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

DRT  
Implementation

Summary

References



## Discussion Point

We have established that anaphora resolution is not possible for *John does not own a donkey. It is grey.* In DRT this is argued to have to do with the level of embedding that negation creates.

- ▶ What about *John does not own **the** donkey. It is grey?* Would you accept this as a valid combination of sentences?
- ▶ Does this raise an issue for the DRT argumentation?

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



## Accessibility: Informal Definition

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

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References

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

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

**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  $[_1 \dots]$  is accessible to the DRS in  $[_2 \dots]$ , but not the other way around.

<sup>1</sup>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.

(17) 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  $[_2 \dots]$ . 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  $[_2 \dots]$ . We follow Kamp (2016) here. As to accessibility: The discourse referents  $x$  and  $y$  are accessible to  $v$  and  $w$  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

DRT  
Implementation

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.

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

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

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

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

**Note:** It is (implicitly) assumed here that in (18) we also have a pronoun as the subject of the consequent statements (*likes it*), while it is not explicitly realized here.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



---

## **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  $x_1, \dots, x_n$  are discourse referents,<sup>2</sup> then  $P(x_1, \dots, x_n)$  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.<sup>3</sup>
- (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.

<sup>2</sup>In the actual examples, Geurts & Beaver (2007) do not use variable  $x$  with indices but rather  $x, y, z$ , etc.

<sup>3</sup>Kamp (1995, p. 149) states that “the DRS language in which the only complex conditions are of the form  $\neg K$  has the expressive power of the full predicate calculus.” This means, in a strict sense, we would not have to define any of the other operators.

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

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.

(20) John chased Jumbo.

[ $x, y$ : John( $x$ ), Jumbo( $y$ ), chased( $x, y$ )]

(21) John chased a donkey.

[ $x, y$ : John( $x$ ), donkey( $y$ ), chased( $x, y$ )]

(22) A farmer chased a donkey.

[ $x, y$ : farmer( $x$ ), donkey( $y$ ), chased( $x, y$ )]

(23) 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

DRT  
Implementation

Summary

References



## Clause (ii): Simple Conditions

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

- (24) John chased the donkey.  
[ $x, y$ : John( $x$ ), donkey( $y$ ), chased( $x, y$ )]
- (25) John chased a donkey.  
[ $x, y$ : John( $x$ ), donkey( $y$ ), chased( $x, y$ )]
- (26) He caught it.  
[ $\underline{v}, \underline{w}$ : 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

DRT  
Implementation

Summary

References



## Clause (iii): Variable Mapping

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

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

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

(28) If John owns a donkey, he likes it.

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

**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

DRT  
Implementation

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.

(29) John doesn't own a donkey.

$[_1 x: \text{John}(x), \neg[_2 y: \text{donkey}(y), \text{owns}(x,y)]]$

(30) If John owns a donkey, he likes it.

$[_1 : [_2 x, y: \text{John}(x), \text{donkey}(y), \text{owns}(x,y)] \rightarrow [_3 : \text{likes}(x,y)]]$

(31) John owns a donkey or a horse.

$[_1 x: \text{John}(x), [_2 y: \text{donkey}(y), \text{owns}(x,y)] \vee [_3 y: \text{horse}(y), \text{owns}(x,y)]]$

**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,  $\text{John}(x)$  here should be *outside* of the two DRSs connected by disjunction. Otherwise, it might not be accessible to both of the occurrences in  $\text{owns}(x,y)$ .

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



## Discussion Point

What about the following sentence combinations:

- ▶ *John owns a donkey or a horse. It is grey.*
- ▶ *John owns a donkey or a horse. In any case, it is grey.*

Do you see a difference between the two in terms of acceptability? What could be the difference?

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

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.

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

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

(33) Some farmer who owns a donkey, likes it.

$[_1 : [_2 x, y: \text{farmer}(x), \text{donkey}(y), \text{owns}(x,y)] (\exists x) [_3 : \text{likes}(x,y)]]]$

**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

DRT  
Implementation

Summary

References



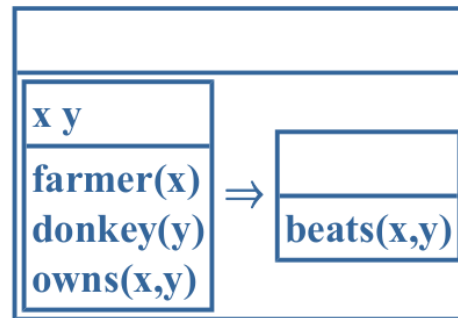
## Different DRS Notations

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).

Official DRS:  $\langle \{\}, \langle \{x, y\}, \{\text{farmer}(x), \text{donkey}(y), \text{owns}(x,y)\} \rangle \Rightarrow \langle \{\}, \{\text{beats}(x,y)\} \rangle \rangle$

Linear notation:  $[: [x, y: \text{farmer}(x), \text{donkey}(y), \text{owns}(x,y)] \Rightarrow [: \text{beats}(x,y)]]$

Box notation:



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

DRT  
Implementation

Summary

References



## 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

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

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

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

such that

$$U_K \cup U_{K'} = \{x, y, \underline{v}, \underline{w}\}$$

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

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



## 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.

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

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

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

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

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



---

## **Section 3: Accessibility**



## Accessibility: Formal Definition

“**Accessibility** is a relation between DRSs that is **transitive**<sup>4</sup> and **reflexive**,<sup>5</sup> 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  $K''$ ,<sup>6</sup>
- ▶  $K' \rightarrow K''$ , 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.

<sup>4</sup>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.

<sup>5</sup>Every DRS is accessible to itself.

<sup>6</sup>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

DRT  
Implementation

Summary

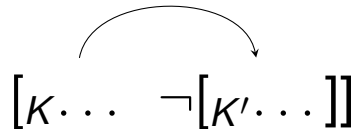
References



## Accessibility: Negation

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

### Schematic Representation



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  $[1... \neg[2...]]$ .

### Examples

(38) John does **not** own a donkey. It is grey.

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

(39) John does own a donkey. It is **not** grey.

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

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

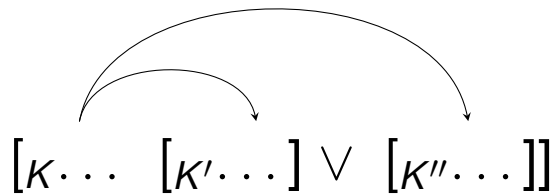
References



## Accessibility: Disjunction

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

### Schematic Representation



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

### Examples

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

**Note:** This natural language construction can only be captured correctly in the DRSs since  $\text{John}(x)$  is in  $[_1 \dots]$  rather than  $[_2 \dots]$  and hence  $x$  is accessible to  $v$ . If  $x$  was in  $[_2 \dots]$  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

DRT  
Implementation

Summary

References

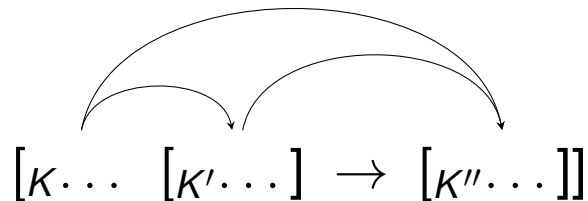




## Accessibility: Conditional

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

### Schematic Representation



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

### Example

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

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

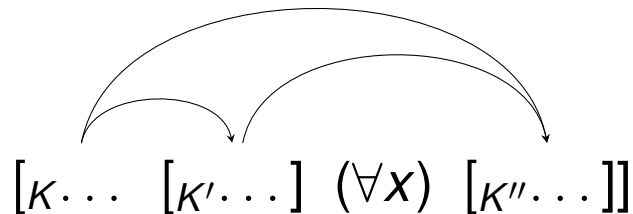
References



## Accessibility: Quantification

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

### Schematic Representation



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

### Example

- (42) Every farmer who owns a donkey, likes it.  
 $[_1 : [_2 \ x, y: \text{farmer}(x), \text{donkey}(y), \text{owns}(x,y)] (\forall x) [_3 \ \underline{v}, \underline{w}: \text{likes}(v,w)]]]$

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

DRT  
Implementation

Summary

References



## 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

DRT  
Implementation

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 $V_M$ in Standard Predicate Logic

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

## Embedding function $f$ in DRT

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

DRT  
Implementation

Summary

References



# DRT Implementation



## DRT Implementation

“Boxer is a semantic parser for English texts with many input and output possibilities, and various ways to perform meaning analysis based on **Discourse Representation Theory**. This involves the various ways that meaning representations can be computed, as well as their possible semantic ingredients.”

Bos (2015). Open-domain semantic parsing with Boxer.

John did not go to school .

```

-----
|x1
|.....|
|named(x1, john, per)|
|
|   -----   |
|   |e2 x3    | |
|   |.....| |
|   |go(e2)   | |
|   |agent(e2, x1)| |
|   |school(x3)| |
|   |to(e2, x3)| |
|   |-----| |
|-----|

```

Section 1: Recap  
of Lecture 14

Section 2: Syntax  
of the DRS  
Language

Section 3:  
Accessibility

Section 4: The  
Semantics of the  
DRT Language

**DRT  
Implementation**

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

DRT  
Implementation

Summary

References



---

## References



## References

Bos, Johan (2015). Open-domain semantic parsing with Boxer. *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*.

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

DRT  
Implementation

Summary

References



# Thank You.

## Contact:

Faculty of Philosophy

General Linguistics

Dr. Christian Bentz

SFS Wilhelmstraße 19-23, Room 1.24

[chris@christianbentz.de](mailto:chris@christianbentz.de)

Office hours:

During term: Wednesdays 10-11am

Out of term: arrange via e-mail