(Diskursrepräsentationstheorie)

3.1Preliminaries

3.2DRS Construction

After we have seen how DRT is intended to work we will have a closer look at its precise formulation. In this section we will treat the way how DRSs are **constructed** from a given text. Previously, we have treated the interaction of syntactic structure and semantic interpretation by giving each syntactic rule like X → Y Z a corresponding interpretation rule, e.g. $[\![[_X Y Z]]\!] = [\![Y]\!]([\![Z]\!])$, where $[\![Y]\!]$ and $[\![Z]\!]$ were meanings, like objects, sets of objects, sets of possible worlds, or functions built up from such entities. Now we will have to develop a system in which syntactic rules are translated into rules that built up discourse representation structures. In the following section we will then be concerned with how these DRSs are **interpreted**.

3.2.1 Syntax of English Fragment

Kamp (1981) works with a very plain illustrative fragment of English. Things are considerably refined in Kamp & Reyle (1993). I will essentially follow the latter work here, which uses a GPSG syntax to specify the English fragment. See the attached copy for the syntactic rules.

3.2.2 The Basic Steps of DRS Construction

The DRS construction rules expect as **input**

- a discourse D = S_1, S_2, ... S_n (= a finite sequence of sentences)
- an initial DRS K_0. (We may assume that K_0 represents the background common knowledge of speaker and hearer, aspects of the situation, etc.; for our purposes we assume K_0 to be empty.)

and yield as **output**

- a DRS of the discourse D.

There are two recursion steps in the construction of DRSs:

- **Extrasentential**: The sentences of D are covered sequentially, one after the other. This will lead to the well-known asymmetry for antecedents and anaphora: Antecedents have to precede their anaphora.
- **Intrasentential**: Each sentence S_i leads to changes in the DRS according to its internal syntactic structure.

These steps can be spelled out as follows:

(1) Given a discourse D = S_1, ...S_n and an initial DRS K_0, do the following:

a) For i: = 1 to n: Add sentence S_i to the conditions of K_{i-1}; go to (b)

b) Keep applying the DRS construction rules to each reducible condition of K_{i-1}
until a DRS is obtained that only contains irreducible conditions;
call this DRS K_i and go to (a).

Note that the extrasentential rules apply in the order of the sentences of the discourse, and the intrasentential rules apply in the order specified by the syntactic structure of each sentence (its "triggering configuration"). If a sentence is syntactically complex, this may lead to indeterminacy, that is, different construction orders are possible. We will see that this may be an attractive feature of DRT.

3.2.3 Names, Pronouns, and Indefinites

Assume the following discourse and an empty initial DRS K_0. (I will specify the syntactic structure only insofar as necessary.)

(2) a. $[_S[_{NP}\text{Pedro}]\ [_{VP'}\text{has a donkey}]]_{S1}$. $[_S[_{NP}\text{He}]\ [_{VP'}\text{beats it}]]_{S2}$.

b. K_0 =

The extrasentential recursion asks us to put the first sentence as a condition into the initial DRS. The DRS K_0 changes as follows:

$[_S[_{NP}[_{PN}\text{Pedro}]]\ [_{VP'}\ [_{VP}\ \text{has a donkey}]]]$

The first construction rule, CR.PN, the rule for proper names, applies to exactly this syntactic configuration.

<u>CR.PN:</u>

· Triggering configuration: $[_S[_{NP}[_{PN}\alpha]][_{VP'}\ ]]]$ or $[_{VP}[_V\ ][_{NP}[_{PN}\alpha]]]$, as a condition of DRS K.

· Introduce into universe of main DRS of K a new discourse referent d

· Introduce into condition set of main DRS of K the condition α = d

· Substitute d for $[_{NP}[_{PN}\alpha]]$ in the triggering configuration.

In our case this will yield the following DRS:

u
u = Pedro
$[_S\ \text{u}\ [_{VP'}[_{VP}[_V\text{has}]\ [_{NP}[_{DET}\text{a}][_N\text{donkey}]]]]]$

Now the construction rule CR.ID for indefinite NPs can apply:

<u>CR.ID:</u>

· Trigger: $[_S[_{NP}[_{DET}\text{a}]\ [_N\ \beta]]\ [_{VP'}\ ]]$ or $[_{VP}[_V\ ]\ [_{NP}[_{DET}\text{a}]\ [_N\ \beta]]]$, as a condition of a DRS K.

· Introduce into universe of K a new discourse referent d

· Introduce into condition set of K the condition $[_N\ \beta]$(d)

· Substitute d for $[_{NP}[_{DET}\text{a}]\ [_N\ \beta]]$ in the triggering configuration.

In our case this will lead to the following changes in the DRS K_0:

u v
u = Pedro
$[_N\ \text{donkey}](\text{v})$
$[_S\ \text{u}\ [_{VP'}[_{VP}[_V\text{has}]\ \text{v}]]]$

Conditions like the last one often are abbreviated by [u has v].

Note that CR.PN and CR.ID differ insofar as the first one introduces a discourse referent in the **main** DRS to which K belongs, whereas the second one introduces a DRS in K **itself**. In the present example this doesn't make a difference, but it will be crucial in later examples.

There are no more construction rules that could be applied. Hence we rename the DRS K_1, go back to the extrasentential recursion, and add the second sentence as a condition. After this step, K_1 looks as follows:

u v
u = Pedro
$[_N$ donkey](v)
$[_S$ u $[_{VP'}[_{VP}[_V$has] v]]
$[_S[_{NP}[_{PRO}$he]] $[_{VP'}[_V$beats] $[_{NP}[_{PRO}$it]]]]

Now we have to apply a rule for pronouns. Here it is:

<u>CR.PRO:</u>

- Trigger: $[_S[_{NP}[_{PRO}\alpha][_{VP'}$]] or $[_{VP}[_V$ $][_{NP}[_{PRO}\alpha]]]$, as condition of DRS K.
- Choose a suitable antecedent discourse referent d accessible from K.
- Substitute d for $[_{NP}[_{PRO}\alpha]]$ in the triggering configuration.

The notion of "accessible" is crucial. For now note that any discourse referent in the universe of K is accessible. Given that *he* denotes a male person and *it* may denote an animal, we get the following result after two applications of CR.PRO:

u v
u = Pedro
$[_N$ donkey](v)
$[_S$ u $[_{VP'}[_{VP}[_V$has] v]]]
$[_S$u $[_{VP'}[_V$beats] v]]

No other construction rule can be applied at this point; we rename K_1 to K_2 and go back to the extrasentential recursion. We find out that there is no further sentence in the discourse, hence K_2 (= K_1) is the final DRS.

A comment is in order: The DRS construction rules we have considered so far do not follow nicely the Fregean ideal that the meaning (the representation) of a complex expression $[\alpha\ \beta]$ is given in terms of the meaning (the representation) of the immediate parts α and β. Certain properties of the DRS construction rule prevent an easy formulation of construction rules in that format. For example, an indefinite NP like *a donkey* is associated with a discourse referent d, the requirement that d be new, and the condition *donkey*(d). Hence it is not simply an entity of a certain type.

3.2.4 Relative Clauses and Indeterminacy

Let us add here one construction for a complex syntactic construction, relative clauses. The syntactic form of relative clauses can be illustrated with as follows:

(3) Pedro owns $[_{NP}[_{DET}$a$][_N[_N$donkey] $[_{RC}[_{RPRO}$which$][_S[_{NP}$he] $[_{VP'}[_{VP}[_V$beats] $[_{NP}$e]]]]]

Note that the relative pronoun *which* corresponds to a gap e within the relative clause; the pronoun and the gap share features, e.g. +/- human.

Assume that (3) is a one-sentence discourse. After rules CR.PN and CR.ID have applied, we get the following DRS:

u v u = Pedro $[_N[_N \text{donkey}][_{RC}[_{RPRO}\text{which}][_S[_{NP}\text{he}][_{VP'}[_{VP}[_V\text{beats}][_{NP}e]]]]]](v)$ [u owns v]

The rule for relative clauses, CR.RC, and its result in the case at hand, is as follows:

<u>CR.RC:</u>

· Triggering configuration: $[_N[_N\alpha][_{RC}[_{RPRO}\ ][_S..\ e\ ...]]](d)$ in DRS K.

· Introduce new condition $[_N\alpha](d)$

· Replace the trigger configuration by $[_S...d...]$, the sentence in which the gap occurs, with the gap replaced by d.

u v u = Pedro $[_N\text{donkey}](v)$ $[_S[_{NP}\text{he}][_{VP'}[_{VP}[_V\text{beats}][_{NP}\ v]]]$ [u owns v]

Application of CR.PRO (with u as the antecedent discourse referent) will yield the following final (abbreviated) DRS:

u v u = Pedro $[_N\text{donkey}](v)$ $[_S u\ [_{VP'}[_{VP}[_V\text{beats}][_{NP}\ v]]]]$ [u owns v]

We can now illustrate an important property of intrasentential DRS-construction with the construction rule for relative clauses: The construction process may be **indeterministic**. We need this feature to account for the anaphoric possibilities with examples like the following one:

(4) a. A farmer who saw Chiquita liked her.

b. A farmer who saw her kicked Chiquita.

In both cases *her* can refer to Chiquita. To get this we must first apply CR.PN to *Chiquita* and then CR.PRO to *her* (otherwise the discourse referent for *her* is not available).

Now, if we would always first reduce the relative clause and then the main clause, we could get (4.a) but not (4.b). And if we would always first reduce the main clause and then the relative clause, we would get (b) but not (a). In order to get both (a) and (b), the order of reduction must stay undetermined. That means we may first reduce one clause as far as possible, then go to the other, and perhaps come back to the first one. But this is allowed only within a clause, not for separate sentences in a discourse — otherwise, we would be able to construct a decent DRS for a text like *Pedro owns her. He kicked Chiquita.*

3.2.5 Negation, Conditionals, and Universal Quantifiers

We turn to syntactic constructions that introduce complex conditions. First, negation, illustrated with the following example:

(5) $[_{S}$a farmer $[_{VP'}[_{AUX}$did] not $[_{VP}$own a donkey]]]

<u>CR.NEG:</u>

- Triggering configuration: $[_{S}$d $[_{VP'}[_{AUX}$] not $[_{VP}\ \alpha]]]$
- Replace triggering configuration by the condition
- Replace the triggering configuration by the condition $\neg$ $\boxed{[_{S}\ d\ [_{VP'}[_{VP}\ \alpha]]]}$

Example: We start out with the following DRS:

$[_{S}$a farmer $[_{VP'}[_{AUX}$did] not $[_{VP}$own a donkey]]]

CR.ID will yield the following DRS:

u [farmer](u) $[_{S}$u $[_{VP'}[_{AUX}$did] not $[_{VP}$own a donkey]]]

This is the input for CR.NEG, which will yield the following DRS:

u [farmer](u) $\neg$ $\boxed{[_{S}\text{u } [_{VP'}[_{VP}\text{owns a donkey}]]]}$

Further reductions via CR.ID will yield the following final DRS. Note that the condition in CR.ID that a new discourse referent must be introduced within the **local** DRS K (in contrast to names) makes a difference here!

u [farmer](u) $\neg$ $\boxed{\begin{array}{l}\text{v}\\ \text{[donkey](v)}\\ [_{S}\text{u } [_{VP'}[_{VP}\text{owns v}]]]\end{array}}$

The fact that v is introduced in the local DRS should make it inaccessible for pronouns outside of the local box, but accessible inside (see (6.a, b). Hence by determining in which DRS a discourse referent is introduced we specify the “life span” of this discourse referent, in Karttunen’s sense. Names behave different (see c); their discourse referent is always introduced in the main DRS.

(6) a. A farmer did not own a donkey. *He beat it.

b. A farmer did not own a donkey which was seen by a widow who beat it.

c. A farmer did not know Chiquita. He beat it.

Let us now turn to conditional sentences:

(7) $[_S$if $[_S$Pedro owns a donkey] then $[_S$he beats it]]

CR.COND:

· Triggering configuration: $[_S$ if $[_S\ \alpha]$ then $[_S\ \beta]]$

Replace triggering configuration by the condition $\boxed{[_S\ \alpha\ ]}$ $\boxed{[_S\ \beta\ ]}$
This yields the following result for example (7):

$[_S[_{NP}[_{PN}$Pedro$][_{VP'}[_{VP}$owns $[_{NP}[_{DET}$a$][_N$donkey$]]]]]]$	$[_S[_{NP}[_{PRO}$he$]][_{VP'}[_{VP}$beats $[_{NP}[_{PRO}$it$]]]]]]$

Applying rules CR.PN and CR.ID on the antecedent box we will get the following; note that the discourse referent for the name and the condition is introduced in the principal DRS, whereas the discourse referent for the indefinite NP is introduced in the local DRS.

u

u = Pedro

v [donkey](v) [u owns v]	$[_S[_{NP}[_{PRO}$he$]][_{VP'}[_{VP}$beats $[_{NP}[_{PRO}$it$]]]]$

We now have to apply rule CR.PRO for the consequent. Now it is time to define the notion of accessible discourse referents. It is defined in terms of the relation of one DRS being subordinated to another one. [The definition below differs slightly from the one in K&R 1993]

(8) Def: A DRS K_1 is **immediately subordinate** to a DRS K_2 iff either
a) K_2 contains the condition $\neg K_1$, or
b) K_2 contains the condition of the form K_1 K_3 or K_3 K_1, for some DRS K_3,
c) K_1 occurs in a condition K_2 K_1.

(9) Def: A DRS K_1 is **subordinate** to a DRS K_2, $K_1 < K_2$, iff either
a) K_1 is immediately subordinate to K_2, or
b) There is a K_3 such that K_3 is subordinate to K_2 and K_1 is immediately subordinate to K_3.

We write $K_1 \leq K_2$ iff $K_1 < K_2$ or $K_1 = K_2$.

(10) Def: A discourse referent d is **accessible** from a DRS-condition c in a DRS K iff d belongs to a DRS K', and $K \leq K'$.

Note that for the condition in the consequent DRS in our example u and v are accessible in this sense

- u is accessible because of clause (c.i).
- v is accessible because of clause (c.ii)

Using CR.PRO twice we get the following result:

u u = Pedro <table><tr><td>v [donkey](v) [u owns v]</td><td>$[_S[_{NP}$ u $][_{VP'}[_{VP}$beats $[_{NP}$ v]]]</td></tr></table>

Universally quantified NPs are treated like conditionals:

<u>CR.EVERY:</u>

- Triggering condition: $[_S[_{NP}[_{DET}$every] $[_N\ \beta]]\ [_{VP'}\ ]]$ or $[_{VP}[_V\ ]\ [_{NP}[_{DET}$every] $[_N\ \beta]]]$, as a condition of a DRS K.
- Introduce into conditions of K the new condition $K_1 \quad K_2$, where K_1, K_2 empty.
- Introduce into K_1 a new discourse referent d
- Introduce into K_1 the condition $[_N\ \beta](d)$.
- Introduce into K_2 a condition like the sentence of the triggering condition, but with d for $[_{NP}[_{DET}$every] $[_N\ \beta]]$.
- Delete the triggering condition.

Example:

(11) $[_S[_{NP}[_{DET}$Every] $[_N$farmer $[_{RC}$who owns a donkey]]] $[_{VP'}$beats it]]

Applying CR.EVERY we get the following DRS:

u $[_N$farmer $[_{RC}$who owns a donkey]](u)	[u $[_{VP'}$ beats it]]

And after application of CR.RC, CR.ID and CR.PRO we arrive at:

u v [farmer](u) [donkey](v) [u owns v]	[u beats v]

3.2.6 Antecedents for Anaphora

Anaphora resolution (i.e. the task of finding an antecedent for an anaphoric element, like a pronoun) is of central concern to DRT. DRT mainly covers the logical accessibility of antecedents. However, there is a lot more to anaphora resolution:

- Grammatical gender information. Many languages use grammatical gender information to restrict suitable antecedents. Every noun belongs conventionally to a gender class, and if a pronoun is to pick up a discourse referent, it has to agree with the gender class of the head noun that introduced this discourse referent.
- Natural gender information. A pronoun has to correspond to the semantic sort, e.g. the sex, of the referent of the discourse referent that is to be picked up. Example German, which sometimes can use either strategy:

(12) Das Mädchen trat herein. {Sie/es} setzte sich.
the girl.NEAT entered. She/it sat down.

Gender information may be added, using special conditions with the introduction of discourse referents, like “Feminine(d)” (see K&R 1993).

- Recency. Pronouns tend to pick up the antecedent that is closest.

(13) a. A farmer had a donkey. He also had a dog. He beat it.

b. A farmer had a donkey. If he also has a dog, he beat it.

Recency information requires us to keep track of the order in which discourse referents are introduced, or at least to be able to identify the discourse referents introduced most recently. (Cf. the notion of “focus” in Grosz & Sidner 1986).

3.3DRS Interpretation

After having seen how DRS construction works we turn to the interpretation of DRSs.

3.3.1 Models

DRSs are interpreted with respect to a model, just like a formal language like predicate logic. A model M = A, F consists of a universe A and a function F, where F has the following property:

- for every proper name α, F(α) A
- for every intransitive verb and common noun α, F(α) A
- for every transitive verb α, F(α) AxA.

Note: In K&R models are given, somewhat unconventionally, as triples U_M, $Name_M$, $Pred_M$, where U_M is the universe, where $Name_M(\alpha)$ gives the extension of names α and $Pred_M(\alpha)$ gives the extension of predicates α.

3.3.2 Truth for DRSs

We now come to a definition what it means for a DRS to be true with respect to a model M = A, F .

We first define some auxiliary concepts:

- If g is a function, then Dom(g) and Ran(g) give the domain and range of g.
- g′ is called an extension of g iff g g′. That is, g and g′ agree on the values for Dom(g), but g′ might have a larger domain.
- U(K) and C(K) denote the universe and the set of conditions of a DRS K.

A central notion for the truth definition is that a function g **verifies** a DRS K, or a DRS condition, with respect to a model M.

(14) (i) g verifies the DRS K in M iff g verifies every element of C(K) in M.

(ii) g verifies the condition γ in M iff:
a) γ is of the form d = α, and g(d) = F(α).
b) γ is of the form α(d), and g(d) F(α).
c) γ is of the form [d α], and g(d) F(α).
d) γ is of the form [d α d′] and g(d), g(d′) F(α).
e) γ is of the form ¬ K′, and there is no extension g′ of g
with Dom(g′) = Dom(g) U(K′)such that g′ verifies K′ in M.
g) γ is of the form K_1 K_2,
and for every extension g′ of g with Dom(g′) = Dom(g) U(K_1) that verifies K_1 in M,
there is an extension g″ of g′ with Dom(g″) = Dom(g′) U(K_2) that verifies K_2 in M.

Note that in the definition of what it means that g verifies a DRS K we call up the definition of what it means that g verifies a DRS condition γ (in K), and in the definition of what it means that g verifies a condition γ we call up the definition of what it means that g verifies a DRS K′ (in e and g). But this is not a *circulus vitiosus*, as K′ is contained in K and hence will always be smaller than K. We have the form of a double recursive definition.

With the help of the notion of **verification** we can define the notion of **truth**:

(15) A DRS K is **true** in a model M = A, F
iff there is a function g: U(K) → A that verifies K in M.

That is, for a DRS to be true with respect to a model we must find some embedding of the universe of the DRS into the universe of the model that verifies the DRS with respect to the model.

3.3.3 Example

Let us discuss the following discourse:

(16) Pedro owns a donkey. He loves it. If he owns a donkey, a widow beats it.

The DRS construction rules will give us the following DRS K:

(17) K =

u v

u = Pedro
[donkey](v)
[u owns v]
[u loves v]

w

[donkey](w)
[u owns w]

x

[widow](x)
[x beats w]

Assume the following model M:

· (18) A = {p, j, m, s, d1, d2, d3},

F(Pedro) = p, F(farmer) = {p, j}, F(donkey) = {d1, d2, d3}, F(widow) = {m, s}, F(owns) = { p, d1 , p, d2 , j, d3 }, F(loves) = { p, d1 }, F(beats) = { m, d1 , s, d2 }

According to the definition, K is true in M iff there is a function from U(K) = {u, v} to A which verifies K in M. The function g = { u, p , v, d1 } has the required property, that is, it verifies all the conditions of K. In particular, we have

- g verifies “u = Pedro”, as g(u) = F(Pedro) = p;
- g verifies “donkey(v)”, as g(v) F(donkey);
- g verifies “[u owns v]”, as g(u), g(v) F(owns);
- g verifies “[u loves v]”, as g(u), g(v) F(loves);
- g verifies “[w | donkey(w), [u owns w]] [x | widow(x), [x beats w]]”, as we have:
  — for every g′: {u, v} {w} → A, with g g′ that verifies K_1 in M
  — there is a g″: {u, v, w} {x} → A, with g′ g″ that verifies K_2 in M.

To check the last claim:

- Take g′ = g_1 = { u, p , v, d1 , w, d1 }; we have that g_1 verifies [w | donkey(w), [u owns w]] in M, as — g_1 verifies “donkey(w)”, as g_1(w) F(donkey); — g_1 verifies “[u owns w]”, as g_1(u), g_1(w) F(own); and there is a g″: {u, v, w} {x} → A, with g_1 g″, namely h_1 = { u, p , v, d1 , w, d1 , x, m } that verifies [x | widow(x), [x beats w] in M, as — h_1 verifies “widow(x)”, as h_1(x) F(widow); — h_1 verifies “[x beats w]”, as h_1(x), h_1(w) F(beats).
- Take g′ = g_2 = { u, p , v, d1 , w, d2 }; we have that g_2 verifies [w | donkey(w), [u owns w]] in M (as above), and there is a g″: {u, v, w} {x} → A, with g_2 g″, namely h_2 = { u, p , v, d1 , w, d2 , x, s } that verifies K_2 in M, as above.
- Take other versions of g, e.g. g_3 = { u, p , v, d1 , w, m }; they verify the condition as well, as they fail to verify the antecedent condition [w | donkey(w), [u owns w]]

Hence we have shown that K is true in M.