Handout 9: Introduction to CDRT+GQ
Semantics C (Spring 2010)

1. Dynamic Ty2

To goal is to achieve compositionality at sub-clausal level by

"[…] combining Montague Semantics and Discourse Representation into a formalism that is not only notationally adequate, in the sense that the working linguist need remember only a few rules and notations, but is also mathematically rigorous and based on ordinary type logic. […] DRT's Discourse Representation Structures (DRSs or boxes henceforth) are already present in type logic […] The presence of boxes in type logic permits us to fuse DRT and Montague Grammar in a rather evenhanded way: both theories will be recognizable in the result. […] With this unification of the theories standard techniques (such as type-shifting) that are used in Montague Grammar become available in DRT."
(Musken 1996: 144-145)

"[The unification of Montague semantics and dynamic semantics is] based on two assumptions and one technical insight. The first assumption is that meaning is compositional. The meaning of words (roughly) are the smallest building blocks of meaning and meanings may combine into larger and larger structures by the rule that the meaning of a complex expression is given by the meaning of its parts. The second assumption is that meaning is computational. Texts effect change, in particular, texts effect changes in context. The meaning of a sentence or text can be viewed as a relation between context states, much in the way that the meaning of a computer program can be viewed as a relation between program states. […] The technical insight […] is that virtually all programming concepts to be found in the usual imperative computer languages are available in classical type theory. We can do any amount of programming in type theory. This suggests that type theory is an adequate tool for studying how languages can program context change. Since there is also some evidence that type theory is also a good vehicle for modelling how the meaning of a complex expression depends on the meaning of its parts, we hope that it is adequate for a combined theory: a compositional theory of the computational aspects of natural language meaning."
(Mussken 1991: 3-4)

1. Definitions and Abbreviations

The definition of types in (1) below isolates a subset of of the set of types as the types of drefs:

- type \( t \) is the type of truth values: the domain of type \( t \) is \{T, F\}
- type \( e \) is the type of individuals; for now, we assume that \( D_e \) contains only atomic entities, i.e., there are no pluralities
- the domain of type \( s \) \( (D_s) \) models DPL's variable assignments; several axioms will be needed to ensure that the entities of type \( s \) actually behave as DPL variable assignments

Drefs:
- modeled as functions that take 'assignments' as arguments (i.e., entities of type \( s \)) and return a static object as value, e.g., an individual (type \( e \))
- so, a dref for individuals is of type \( se \)
- this is not as different from the DPL way of modeling drefs as it might seem: DPL models drefs as variables and a variable \( x \) is basically an instruction to look in the current info state, i.e., the current variable assignment \( g \), and retrieve whatever individual the current info state associates with \( x \) – that individual is, of course, \( g(x) \)
- instead of working directly with variables, we can work with their 'type-lifted' versions, i.e., instead of \( x \), we can take a dref to be a function of the form \( g \)\( (x) \), which is the (set-theoretic) projection function that projects sequence \( g \) onto coordinate \( x \)
- this is what Dynamic Ty2 does: instead of modeling discourse referents as atomic entities (variables) and info states as functions taking drefs as arguments (i.e., variable assignments), we model info states as atomic entities (of type \( s \)) and drefs as functions taking info states as arguments
- drefs are similar to the Montagovian individual concepts (e.g., the denotation of Miss America), which are functions from indices of evaluation to individuals
- both drefs and individual concepts refer only indirectly, as a function of the current context of evaluation

1.1. Preliminaries

The logic that underlies the entire unification project is Ty2 (Gallin 1975; see also Janssen 1986 and Carpenter 1998). The set of basic types is \{t, e, s\}:
- type \( t \) is the type of truth values: the domain of type \( t \) is \{T, F\}
- type \( e \) is the type of individuals; for now, we assume that \( D_e \) contains only atomic entities, i.e., there are no pluralities
- the domain of type \( s \) \( (D_s) \) models DPL's variable assignments; several axioms will be needed to ensure that the entities of type \( s \) actually behave as DPL variable assignments

Page references are to the online version.
1. **Dynamic Ty2** – the set of dref types DRefTyp and the set of types Typ.
   a. The set of basic static types BasTyp: \{f, e\} (truth-values and individuals).
   b. The set of static types STyp: the smallest set including BasTyp and s.t., if \(\sigma, \tau \in STyp\), then \((\sigma) \in STyp\).
   c. The set of dref types DRefTyp: the smallest set s.t., if \(\sigma \in STyp\), then \((\sigma) \in DRefTyp\).
   d. The set of basic types BasTyp: the smallest set including BasTyp and s.t., if \(\sigma, \tau \in DRefTyp\), then \((\sigma) \in Typ\).
   e. The set of types Typ: the smallest set including BasTyp and s.t., if \(\sigma, \tau \in Typ\), then \((\sigma) \in Typ\).

The definition in (2) provides some typical examples of expressions of various types and introduces several notational conventions.

2. **Dynamic Ty2** – basic expressions.
   For any type \(\tau \in Typ\), there is a denumerable set of \(\tau\)-constants Con, and a denumerably infinite set of \(\tau\)-variables Var, \(\{v_{0}, v_{1}, \ldots\}\), e.g.
   \[\text{Con} = \{\text{john}, \text{mary}, \text{dobby}, \ldots, a, a', \ldots, b, b', \ldots, a_{0}, a_{1}, a_{2}, \ldots\}\]
   \[\text{Con}_{\tau} = \{\text{donkey}, \text{farmer}, h\text{elf}, \text{witch}, \ldots, \text{leave, drunk, walk, …}\}\]
   \[\text{Var}_{\tau} = \{x, y, z, \ldots, k_{0}, k_{1}, \ldots\}\]
   Notational conventions:
   - \(x, x', y, y', z, z', x_{0}, x_{1}, \ldots\) are variables of type \(\tau\);
   - \(h, h', i, i', j, j', k', \ldots\) are variables of type \(\tau\);
   - \(f, f', f'', \ldots, i_{0}, i_{1}, \ldots\) are variables over terms of type \(\tau\), for any \(\tau \in STyp\);
   - \(v, v', v''\), \(x, y, z, \ldots\) are variables over terms of type \(\tau\), for any \(\tau \in Typ\).

The definition in (3) introduces the term \(i(\delta)\) of type \(\tau\) that models the DPL notion of random variable assignment:
- the formula \(i(\delta)\) requires the assignments \(i/q\) and \(j/o\) to differ at most with respect to the value they assign to \(\delta\).
- unlike Muskens (1995b, 1996), I introduce this as a basic formula of the language and not as an abbreviation, because the set DRefTyp of drefs is infinite and the abbreviation requires an infinite conjunction of formulas (as indicated in (4d) below).

Proper names with capitals, e.g., \(\text{John}\), are drefs for individuals (type \(se\)) and they are constant functions, a.k.a. specific drefs. They are defined in terms of the corresponding constant of type \(e\), e.g., \(\text{john}\).

3. **Dynamic Ty2** – terms.
   For any \(\tau \in Typ\), the set of \(\tau\)-terms Term, is the smallest set s.t.: 
   \[\text{Con} \cup \text{Var} \subseteq \text{Term} ;\]
   \[\alpha(\beta) \in \text{Term}, \text{ if } \alpha \in \text{Term} \text{, and } \beta \in \text{Term} , \text{ for any } \sigma \in \text{Typ};\]
   \[(\lambda \alpha \beta) \in \text{Term}, \text{ if } \tau = (\alpha \beta) \text{, and } \alpha \in \text{Var} \text{, and } \beta \in \text{Term} , \text{ for any } \sigma \in \text{Typ};\]
   \[(\alpha \beta) \in \text{Term}, \text{ if } \tau = (\alpha \beta) \in \text{Term} , \text{ for any } \sigma \in \text{Typ};\]
   \[(\alpha(\beta)) \in \text{Term}, \text{ if } \tau = i = \alpha, \beta \in \text{Term} , \text{ and } i(\beta) \in \text{Term} , \text{ for any } \sigma \in \text{DRefTyp} .\]

Abbreviations (the subscripts on terms indicate their type):

John := \(\lambda_i. \lambda_j. \text{john}\), Mary := \(\lambda_i. \text{mary}\), …
T := \(f', f = f'_{T}\), F := \(f = f'_{T}\),
\(\vdash x = f'_{T}\), \(\vdash f'_{T} = F\);
\(\vdash \lambda_{i} f = f'_{T}\), \(\forall (f'_{T}(f'_{T})) = \lambda_{i} f_{T}\).

Definition (4) introduces four axioms that Dynamic Ty2 models to satisfy – these axioms make sure that the entities of type \(s\) behave as variable assignments.

**Axiom 1** employs a non-logical constant udref to identify unspecific drefs, i.e., the drefs that are supposed to behave as the DPL variables, e.g., \(u, u', \ldots, a_{0}, a_{1}, \ldots\)

The constant function \(\text{John}\) (recall that \(\text{John} := \lambda_i. \text{john}\)), for example, is a specific dref: although it is of type \(se\), i.e., the type of drefs for individuals, it does not behave as a DPL variable – its value does not vary from 'assignment' to 'assignment'; if anything, specific drefs are the counterpart of DPL constants, not variables.

**Axiom 2** makes sure that all the unspecific dref names actually name different functions: if two distinct names denoted the same function, we would accidentally update both whenever we would update one of them.

**Axiom 3** ensures that, just like DPL variable assignments, two 'assignments' (i.e., two entities of type \(s\)) are different only if they assign different values to some dref \(\delta\). If they assign the same values to all drefs, the 'assignments' are identical.

**Axiom 4** ensures that we have enough 'assignments': for any given 'assignment' \(i\), any unspecific dref \(v\) and any possible dref value (i.e., static object) \(t\) of the appropriate type, there is an 'assignment' \(j\) that differs from \(i\) at \(t\) with respect to the value it assigns to \(v\) and which in fact assigns \(t\) as the value of \(v\).

4. **Dynamic Ty2** – frames, models, assignments, interpretation and truth.
   a. A *standard frame* \(F\) for Dynamic Ty2 is a set \(\{D_{\tau} : \tau \in Typ\}\) s.t. \(D_{\tau}\) and \(D_{\sigma}\) are pairwise disjoint sets and \(D_{\text{Con}} = \{f\} \text{ f is a total function from } D_{\tau} \text{ to } D_{\tau}\), for any \(\tau, \sigma \in Typ\).
   b. A *model* \(M\) for Dynamic Ty2 is a pair \(\langle F^{M} \rangle, \|F^{M} \|^{M}\) s.t.: 
   - \(F^{M}\) is a standard frame for Dynamic Ty2;
   - \(\| \|^{M}\) assigns an object \(\| \|^{M}\) to each \(\alpha \in \text{Con} \text{, for any } \tau \in Typ\) i.e., \(\| F^{M} \|^{M}\) respects typing;
   - \(M\) satisfies the following axioms:
     **Axiom 1** ("Unspecific drefs"): udref(\(\delta\)),
     for any unspecific dref name \(\delta\) of any type \(\tau \in DRefTyp\),
     e.g., \(u_{0}, u_{1}, \ldots\) but not John, Mary, …
     udref is a non-logical constant intuitively identifying the 'variable' drefs,

---

2 Equivalently, \(\vdash \lambda_{i} f_{T}\), \(\forall (f'_{T}(f'_{T})) = \lambda_{i} f_{T}\).

3 In fact, udref stands for an infinite family of non-logical constants of type \((\tau \in Typ)\) for any \(\tau \in DRefTyp\). Alternatively, we can assume a polymorphic type logic with infinite sum types, in which udref is a polymorphic function. For a discussion of sum types, see, for example, Carpenter (1998): 69 et seq.
i.e., the non-constant functions of type $\tau$ (for any $\tau \in \text{STyp}$) intended to model DPL-like variables.

**Axiom2** ("Dref have unique dref names"): $\text{dref}(\delta) \land \text{dref}(\delta') \rightarrow \delta = \delta'$, for any two distinct dref names $\delta$ and $\delta'$ of type $\tau$, for any type $\tau \in \text{DRefTyp}$.

We make sure that we do not accidentally update a dref $\delta'$ when we update $\delta$.

**Axiom3** ("Identity of assignments"): $\forall i, j \in [0, n] : i \neq j \rightarrow \alpha(i) \neq \alpha(j)$, for any type $\tau \in \text{STyp}$.

**Axiom4** ("Enough assignments"): $\forall \sigma_i, \sigma_j : (\forall v \in \text{Var} \rightarrow \text{dref}(v) \rightarrow \exists_\sigma_i (\sigma_i[v] \land \sigma_j[v]))$, for any type $\tau \in \text{STyp}$.

c. An $M$-assignment $\theta$ is a function that assigns to each variable $v \in \text{Var}$, an element $\theta(v) \in D^M_{\text{Var}}$ for any $\tau \in \text{Typ}$. Given an $M$-assignment $\theta$, if $v \in \text{Var}$, and $d \in D^M_{\text{Var}}$, then $\theta[v] = d$ is the $M$-assignment identical with $\theta$ except that it assigns $d$ to $v$.

d. The interpretation function $\llbracket \cdot \rrbracket^M$ is defined as follows:

$$
\begin{align*}
\alpha & : \llbracket \alpha \rrbracket^M = \begin{cases} 
\alpha^M & \text{if } \alpha \in \text{Con}, \text{ for any } \tau \in \text{Typ}; \\
0(\alpha) & \text{if } \alpha \in \text{Var}, \text{ for any } \tau \in \text{Typ}; \\
\alpha(\beta) & : \llbracket \alpha \rrbracket^M \llbracket \beta \rrbracket^M \\
\lambda v. \alpha & : \llbracket \lambda v. \alpha \rrbracket^M = \begin{cases} 
\alpha^M & \text{if } v \notin \text{Var}^M; \\
\alpha^M & \text{if } v \in \text{Var}^M \\
T & \text{if } \alpha^M = T; \\
F & \text{otherwise.}
\end{cases} \\
\end{cases}
\end{align*}
$$

\[\llbracket [\sigma \in \text{DRefTyp} \land \text{dref}(\sigma) \land \text{dref}(\sigma')] \rrbracket^M = T \text{ and } \llbracket [\forall v \in \text{Var} \rightarrow \text{dref}(v) \rightarrow \forall v \in \text{Var} \rightarrow \text{dref}(v)]^M = T \text{ for all } \sigma, \tau \in \text{DRefTyp} \]

\[\llbracket [\forall \exists] \rrbracket^M = T \text{ if } \exists \in \text{Term} \text{ is true in } M \text{ relative to } \theta \text{ iff } \llbracket [\exists] \rrbracket^M = T.\]

A formula $\phi \in \text{Term}$ is true in $M$ if it is true in $M$ relative to any assignment $\theta$.

Given the domain $D$, and the set of dref names, how can we construct the domain $D$, so that $D$ satisfies the axioms?

2. Translating DPL into Dynamic Ty2

We will now encode DPL (and therefore classical DRT / FCS) in Dynamic Ty2.

We do this by providing a list of abbreviations that follows closely the previous definition of DPL: the definiendum has the form of a DPL expression, while the definiens is a term of Dynamic Ty2.

2.1. Definitions and Abbreviations

Definition (5) below corresponds to the previous DPL definition.

- '$\top$' is the Dynamic Ty2 conjunction, i.e., the official, type-logical conjunction, and '$\bot$' is the Dynamic Ty2 negation, i.e., the official, type-logical negation. In contrast, dynamic conjunction '$\land$' and dynamic negation '$\neg$' are simply abbreviations.

- The DPL notion of random assignment $[\sigma]$ has as its direct correspondent the random assignment $[\sigma]$ of Dynamic Ty2.

The DPL distinction between conditions and DRSs is formulated in terms of type.

- conditions are terms of type $s$, i.e., they denote sets of 'assignments'; intuitively, conditions denote the set of 'assignments' that satisfy them.

- DRSs are terms of type $s(i)$, i.e., binary relations between 'assignments'; intuitively, a DRS $D$ is satisfied by a pair of two 'assignments' $i$ and $j$ iff the output 'assignment' $j$ is the result of non-deterministically updating the input 'assignment' $i$ with $D$.

5. DPL in Dynamic Ty2 (subscripts on terms represent their types).

a. Atomic conditions – type $s$:

$$R[u_1, \ldots, u_n] := \lambda d. R[u_1, \ldots, u_n],$$

for any non-logical constant $R$ of type $s$, $u_1, \ldots, u_n$, and $d \in D^M_{\text{Var}}$.

b. Atomic DRSs (DRSs containing exactly one atomic condition) – type $s(s)$ (corresponding to DPL atomic formulas):

$$[R[u_1, \ldots, u_n]] := \lambda d. R[u_1, \ldots, u_n],$$

$$[u_1 = u_2] := \lambda d. u_1 = u_2.$$

Expression of the dynamic negation, i.e., the official, type-logical negation. In contrast, dynamic negation $\neg$ is recursively defined in the expected way, e.g., for DPL atomic formulas, we have that $\neg R[u_1, \ldots, u_n] := \lambda j \exists d. R[u_1, \ldots, u_n] j$.

- Condition-level connectives (negation), i.e., non-atomic conditions:

$$D := \lambda d. \neg \exists j (D(\forall d j)),$$

where $D$ is a DRS (term of type $s(s)$).

i.e., $D := \lambda d. \neg \exists j (D(\forall d j))$.

- Truth: A formula $\phi \in \text{Term}$ is true in $M$ relative to $\theta$ iff $\llbracket [\phi] \rrbracket^M = T$.

A formula $\phi \in \text{Term}$ is true in $M$ if it is true in $M$ relative to any assignment $\theta$.

Given the domain $D$, and the set of dref names, how can we construct the domain $D$, so that $D$ satisfies the axioms?

---

4 Strictly speaking, the Dynamic Ty2 translation of DPL negation is defined as $\text{TR}(\neg \phi) := \neg [\lambda j. \exists d. \text{TR}(\phi)(d)],$ i.e., $\text{TR}(\neg \phi) := [\lambda j. \exists d. \text{TR}(\phi)(d)].$ $\text{TR}$ is the translation function from DPL to Dynamic Ty2 which is recursively defined in the expected way, e.g., for DPL atomic formulas, we have that $\text{TR}(\neg [\lambda j. \exists d. \text{TR}(\phi)(d)]) := \lambda j. \exists d. \text{TR}(\phi)(d)$. $\text{TR}(\neg [\lambda j. \exists d. \text{TR}(\phi)(d)]) := \lambda j. \exists d. \text{TR}(\phi)(d)$.

5 Alternatively, $[C_1, \ldots, C_n]$ can be defined using dynamic conjunction as follows:

$$[C_1, \ldots, C_n] := \lambda d. \exists j (D(\forall d j)).$$

where $[C] := \lambda d. \exists j (D(\forall d j)).$
The abbreviations introduced in definition (6) below correspond to the DPL abbreviations defined in the previous chapter.

- “∃” and “∀” are the official type-logical existential and universal quantifiers, while “◊” and “□” are the abbreviations corresponding to the dynamic (DPL-style) existential and universal quantifiers.
- following the notational conventions in the literature, I use ‘−’ and ‘ ⟨−⟩ ’ both for the official Dynamic Ty2 and for the dynamic DPL-style implication and, respectively, disjunction.

6. a. Additional abbreviations – condition-level connectives (closure, disjunction, implication):

\[ \text{[(D := [-D])]}, \]

i.e., \( [D := \lambda_i, \exists k(Dk)] \) or simply: \( D := \text{Dom}(D) \)

\[ \text{[(D = [-D])]}, \]

i.e., \( D \lor D := [-[D]] \lor [-[D]] \)

\[ \text{[(D := \lambda_i)]}, \]

i.e., \( D \lor D := \lambda_i, \exists k(Dik) \land Dk \)

equivalently: \( D \lor D := \text{Dom}(Dk) \land \text{Dom}(Dk) \)

\[ D \rightarrow D := \text{Dom}(Dk), \text{Dom}(Dk) \]

where \( (D := [j]; Dj) \)

b. Additional abbreviations – DRS-level quantifiers (multiple random assignment, existential quantification):

\[ [u_1, \ldots, u_n] \mapsto \{[u_1]; \ldots; [u_n] \} \]

\[ \text{Dom}(D) := [u]; D \]

c. Additional abbreviations – condition-level quantifiers (universal quantification):

\( \forall \alpha(D) := -([u]; [-D]) \)

i.e., \( -([u]; [-D]) \) or \( [u] \rightarrow D \) or equivalently \( \exists u([D]) \)

\[ \forall \alpha(D) := \lambda_k, \forall h([u]) \rightarrow \exists k(hk) \]

where \( [u]; \lambda_k([u]); \lambda_k([u]) \in \text{Dom}(D) \)

d. Additional abbreviations – DRSs (a.k.a. linearized ‘boxes’):

\[ [u_1, \ldots, u_n] \}

\[ C_1, \ldots, C_n \] := \lambda hju, \{ [u_1, \ldots, u_n]; [C_1, \ldots, C_n] \} / hju

where \( C_1, \ldots, C_n \) are conditions (atomic or not),

i.e., \( [u_1, \ldots, u_n] \}

\[ C_1, \ldots, C_n \] := \lambda hju, \{ [u_1, \ldots, u_n]; [C_1, \ldots, C_n] \} / hju

2.2 Cross-sentential Anaphora

7. A \( u_1 \) house-elf fell in love with a \( u_2 \) witch. 

8. He \( u_2 \) bought her \( u_3 \) an \( u_4 \) alligator purse.

- the DRT-style representation in DPL is provided in (9)
- the DRT-style representation in Dynamic Ty2 is provided in (10)
- the formula in (11) is the ‘unpacked’ type-logical term of type \( st(s) \) translating the discourse in (7-8)
- the formula in (12) provides the truth-conditions associated with the Dynamic Ty2 term in (11), derived on the basis of the definition of truth for DRSs in (5g) and the “Enough States” axiom (Axiom 4 in (4b) above)

Note that the formula in (12) capturing the truth-conditions of discourse (7-8) contains a vacuous \( \lambda \)-abstraction over input ‘assignments’, which is intuitively correct given that the discourse does not contain any item whose reference is dependent on the input context (e.g., a deictically used pronoun).

9. \[[x, y]; h elf(x), witch(y), in love(x, y)]; \]

\[ a \text{parse}(x), buy(x, y, z) \]

10. \([u_1, u_2]; h \text{elf}(u_1), \text{in love}(u_1, u_2)]; \]

\[ [u_1 \mid a \text{parse}(u_1), \text{buy}(u_1, u_2, u_3)] \]

11. \( l \ldots j h u, l \ldots j h u, h \text{elf}(u_1) \land \text{in love}(u_1, u_2) \land \text{a parse}(u_2) \land \text{buy}(u_2, u_3, u_4) \)

12. \( l \ldots j h u, l \ldots j h u, h \text{elf}(u_1) \land \text{in love}(u_1, u_2) \land \text{a parse}(u_2) \land \text{buy}(u_2, u_3, u_4) \)

How do we derive the truth conditions in (12) based on the DRS in (11), the definition of truth in (5g) and Axiom 4?

2.3 Relative-Clause Donkey Sentences

- the formula in (14) provides the DRT-style DPL translation of sentence (13)
- the corresponding Dynamic Ty2 formula is provided in (15) – note the double square brackets: the external square brackets are due to the fact that dynamic implication ‘⇒’ is a condition-level connective, so we need the extra square brackets to obtain a test, i.e., a DRS (which is a term of type \( st(s) \)), out of a condition of type \( st \).

13. Every \( u_1 \) house-elf who falls in love with a \( u_2 \) witch buys her \( u_3 \) an \( u_4 \) alligator purse.

14. \([x, y]; h \text{elf}(x), \text{in love}(x, y)]; \]

\[ a \text{parse}(z), \text{buy}(x, y, z) \]

15. \([u_1, u_2]; h \text{elf}(u_1), \text{in love}(u_1, u_2)]; \]

\[ [u_1 \mid a \text{parse}(u_1), \text{buy}(u_1, u_2, u_3)] \]

- the DRT-style representation in (15) is ‘unpacked’ as the type-logical term in (16)
- the corresponding truth-conditions are given in (17) – note again the vacuous \( \lambda \)-abstraction over ‘assignments’, followed by a static first-order formula that captures the intuitively correct truth-conditions for sentence (13)
the terms in (22) are of type st because both dynamic negation "¬" and universal quantification "∀" are condition-level connectives.

the corresponding tests – which are DRPs, i.e., terms of type s(st) – are identical if the conditions they are based on are identical.

Given the identity in (22), we can translate sentence (19) either way 10 – see (23) and (25). These equivalent translations are also equivalent to the DRT-style formulas in (24) and (26).

Note that the universal quantification over pairs of house-elves and witches is exhibited in the clearest way by (26), since any dref introduced in the antecedent of a conditional ends up being quantified over universally.11

23. \[\exists u_1[(h.elf | u_1)]\] \[\exists u_2[(\text{in}_\text{love} | u_2)]\] \[\exists u_3[(\text{a}_\text{purse} | u_3), \text{buy} | u_1, u_2, u_3)]\]

24. \[\exists u_1[(h.elf | u_1)]\] \[\exists u_2[(\text{in}_\text{love} | u_2)]\] \[\exists u_3[(\text{a}_\text{purse} | u_3), \text{buy} | u_1, u_2, u_3)]\]

25. \[\forall u_1[(h.elf | u_1)]\] \[\exists u_2[(\text{in}_\text{love} | u_2)]\] \[\exists u_3[(\text{a}_\text{purse} | u_3), \text{buy} | u_1, u_2, u_3)]\]

26. \[\exists u_1[(h.elf | u_1)]\] \[\exists u_2[(\text{in}_\text{love} | u_2)]\] \[\exists u_3[(\text{a}_\text{purse} | u_3), \text{buy} | u_1, u_2, u_3)]\]

The formula in (26) is in fact the compositional translation of the negative conditional sentences in (20) and (21) above.

The Dynamic Ty2 truth-conditions for all three sentences, provided in (27), are most easily derived from (24). Just as before, we have vacuous λ-abstraction over 'assignments', followed by a static first-order formula that captures the intuitively correct truth-conditions for the three English sentences under consideration.

27. \[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

Summary:

• Dynamic Ty2 can capture everything that DPL (hence classical DRT / FCS) does, including compositionality down to sentence-clause-level.

• moreover, we now have all the ingredients to go compositional at sub-sentential/sub-clausal level

Proper Names in Dynamic Ty2

Proper names can in principle receive two kinds of analyses in DPL / Dynamic Ty2:

• a pronoun-like analysis, whereby a proper name is basically interpreted as a deictically used pronoun, whose referent is specified by the input discourse context

\[\forall a([D \rightarrow D']) \rightarrow [a] \rightarrow [D \rightarrow D'] \rightarrow ([a] \circ D) \rightarrow D' \]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]

\[\lambda i. \exists z_1, z_2, z_3, (h.elf | z) \land \text{in}_\text{love} | x, y) \land \text{a}_\text{purse} | z \land \text{buy} | x, y, z)\]
• an indefinite-like analysis, whereby a proper name introduces a new individual-level dref whose referent is constrained to be the individual (rigidly) designated by the proper name

Muskens (1996) chooses the pronoun-like analysis of proper names, which are (basically) translated as the corresponding specific drefs.

28. Dobby\(\leadsto\) Dobby

I will choose the indefinite-like analysis and let proper names introduce an unspecific dref and an identity condition between the unspecific dref and the specific dref that is the Ty2 correspondent of the proper name.

29. Dobby\(\leadsto\) \([u\mid\nu=Dobby]\), i.e., \(\lambda_i[I], \lambda_j[I \wedge \nu=Dobby]\), i.e., \(\lambda_i[I \wedge \nu=dobby]\)

This interpretation is equivalent to the external anchoring of proper names in Kamp & Reyle (1993): 248 and it is similar to the interpretation of proper names in Kamp (1981).

Pronouns anaphoric to proper names are taken to be anaphoric to the unspecific dref introduced by the proper name, as exemplified by (30) below.

30. \(\ldots\,Dobby\ldots\,heu\ldots\)

An argument against the indefinite-like analysis (Muskens 1996: 151-153) – the resulting representation seems needlessly complex. Why not simply take the proper name to be anaphoric to its corresponding specific dref?

• the idea would be that proper names are used deictically – so, they are interpreted directly relative to the input context (as the causal chain theory of proper names would have it – see Kripke 1972 and Kaplan 1977/1989a, 1989b)

• a pronoun anaphoric to a proper name would be anaphoric to the corresponding specific dref, as shown in (31) below

The use of a pronoun anaphoric to a proper name and the use of the proper name itself are not semantically different.

31. \(\ldots\,John_{John}\ldots\,he_{John}\ldots\)

An argument against the pronoun-like analysis – the conflation of proper names and pronouns is undesirable for two reasons.

• a proper name is felicitous in a discourse initial position, while a pronoun requires a suitable context (linguistic or non-linguistic) to have previously been set up – as shown in (32) and (33) below.

32. Dobby entered The Three Broomsticks.
33. ??He\(Dobby\) entered The Three Broomsticks.

• when the proper name has been (recently) mentioned, using an anaphoric pronoun is felicitous, while using the proper name again is not

34. Dobby entered the Three Broomsticks. He\(Dobby\) ordered a butterbeer.
35. Dobby entered the Three Broomsticks. ??Dobby ordered a butterbeer.

An argument for the pronoun-like analysis – negation and anaphora to proper names vs. anaphora to indefinites:

• an indefinite introduced in the scope of negation is not anaphorically accessible to a subsequent pronoun, as shown in (36) below

36. Hermione didn't see a\(\ldots\,any\ldots\,house-elf\) at The Three Broomsticks.

• a proper name is anaphorically accessible when it occurs in the scope of negation

37. Hermione didn't see Dobby at The Three Broomsticks.
He\(Dobby\) was on vacation in the Bahamas.

• the fact that dynamic negation is defined as in (5c) above, i.e., as externally static, correctly predicts the infelicity of anaphora in (36): the pronoun, despite being co-indexed with the indefinite, ends up being interpreted as a deictic pronoun, picking up whatever the input context associates with the dref \(u\)

• the reason for the infelicity of discourse (36): the co-indexation of the indefinite and the pronoun formally encodes that the pronoun should be 'bound' by the indefinite, i.e., as far as the speaker is concerned, the indefinite and the pronoun should be co-referent / anaphorically connected (the infelicity follows directly if we work with partial variable assignments)

The indefinite-like analysis of proper names incorrectly predicts that anaphora to proper names introduced under negation is as felicitous as the corresponding anaphora to indefinites.

An indefinite-like analysis can still account for the felicitous anaphora in (37) above if we assume that pronouns can be indexed not only with unspecific drefs, but also with specific drefs like Dobby or John.

That is, in addition to the anaphoric pattern in (30) above, we should allow for the kind of connection between a pronoun and a proper name in (38).

38. \(\ldots\,Dobby\ldots\,he_{Dobby}\ldots\)

• strictly speaking, the pronoun is not co-refering with the proper name – but, in the given context, the pronoun refers to the same entity as the proper name

39. Hermione\(\ldots\,He\ldots\) didn't see Dobby\(\ldots\,He\) at The Three Broomsticks.
He\(Dobby\) was on vacation in the Bahamas.

40. \([u_{H} \mid u_{H}=Hermione, \neg [u_{D} \mid u_{D}=Dobby, see_{\text{at}_{TTB}}(u_{H}, u_{D})]]; [on_{\text{vacation}}(Dobby)]\)
3. Syntax of a Fragment of English

3.1. Indexation

"The most important requirement that we impose is that the syntactic component of the grammar assigns indices to all names, pronouns and determiners" (Muskens 1996: 159).

- we let indices be specific and unspecific drefs (recall that they are all constants of type se), e.g., u, u', u_0, u_1, Dobby etc. Just as before, the antecedents are indexed with superscripts and dependent elements with subscripts (following the convention in Barwise 1987)
- we also allow variables that have the appropriate dref type as indices on traces of movement, e.g., v_0, v_1, v_{true}, v_{false} etc.

But: variable indices appear only on traces – because they are needed only on traces.

"In Montague's PTQ [...] the Quantifying-in rules served two purposes: (a) to obtain scope ambiguities between noun phrases and other scope bearing elements, such as noun phrases, negations and intensional contexts, and (b) to bind pronouns appearing in the expression that the noun phrase took scope over. In the present set-up the mechanism of discourse referents takes over the second task." (Muskens 1996: 169)

The fact that we use distinct indices for the two purposes enables us to keep track of:

- when our indexation makes an essentially dynamic contribution to the semantics
- when it is an artifact of the particular scoping mechanism and the particular syntax-semantics interface we employ

So, it will be easy to reformulate the analyses we develop in a different syntactic formalism.

The choice of a particular (version of a particular) syntactic formalism is largely orthogonal to our concerns. The syntax-semantics interface we define is just a proof of concept, showing that our semantics is compatible with a variety of syntactic formalisms.

3.2. Phrase Structure and Lexical Insertion Rules

- the Y-model of syntax has four components: D-structure (DS), S-Structure (SS), Logical Form (LF) and Phonological Form (PF)
- we are interested in the first three, in particular in the level of LF, which provides the input to the semantic interpretation procedure
- the DS component consists of all the trees that can be generated by the phrase structure rules PS1-PS12 and the lexical insertion rules LI1-LI11 in (41)
- we could do away with rule PS1, but we will keep it as a reminder that sequencing two sentences in discourse occurs at a supra-sentential, textual level

41. Phrase structure rules and lexical insertion rules.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS 1</td>
<td>(Txt) → (Txt) CP</td>
</tr>
<tr>
<td>PS 2</td>
<td>(CP) → (CP) IP</td>
</tr>
<tr>
<td>PS 3</td>
<td>(C) IP → C</td>
</tr>
<tr>
<td>PS 4</td>
<td>(IP) → I VP</td>
</tr>
<tr>
<td>PS 5</td>
<td>(VP) → DP V'</td>
</tr>
<tr>
<td>PS 6</td>
<td>V' → V_a</td>
</tr>
<tr>
<td>PS 7</td>
<td>V' → V_{true} DP</td>
</tr>
<tr>
<td>PS 8</td>
<td>V' → V_{false} DP</td>
</tr>
<tr>
<td>PS 9</td>
<td>V_a → V_{true} DP</td>
</tr>
<tr>
<td>PS 10</td>
<td>DP → D NP</td>
</tr>
<tr>
<td>PS 11</td>
<td>NP → N (CP)</td>
</tr>
<tr>
<td>LI 1</td>
<td>D → ar, every, most, few, nor, some, any, every', ...</td>
</tr>
<tr>
<td>LI 2</td>
<td>DP → heu, sheu, itu, who, whom, ...</td>
</tr>
<tr>
<td>LI 3</td>
<td>DP → heJohn, Maryu, Johnu', ...</td>
</tr>
<tr>
<td>LI 4</td>
<td>DP → who, whom, which, Ø</td>
</tr>
<tr>
<td>LI 5</td>
<td>N → farmer, house elf, donkey, ...</td>
</tr>
<tr>
<td>LI 6</td>
<td>V → own, beat, ...</td>
</tr>
<tr>
<td>LI 7</td>
<td>V_a → sleep, walk, ...</td>
</tr>
<tr>
<td>LI 8</td>
<td>V_{false} → buy, give, ...</td>
</tr>
<tr>
<td>LI 9</td>
<td>I → Ø, doesn't, don't, -ed, -s, didn't, ...</td>
</tr>
<tr>
<td>LI 10</td>
<td>C → if, Ø</td>
</tr>
</tbody>
</table>

- subjects are assumed to be VP-internal and this is where they remain by default even at LF (they are raised out of VP only at PF) – this is how we can interpret sentential negation as having scope over quantifiers in subject position.
- similarly, V-heads move to the inflectional I-head only at PF.

3.3. Relativization and Quantifier Raising

DS and SS are connected via the obligatory movement rule of Relativization (REL).

A tree $\Theta'$ follows by REL, from a tree $\Theta$ iff $\Theta'$ is the result of replacing some sub-tree of $\Theta$ of the form $[v_X [\Theta X \Theta Y]]$, where X and Y are (possibly empty) strings and $wh$ is either who, whom or which, by a tree $[v [\Theta [wh \Theta Y]] [v X [\Theta X [\Theta X [\Theta X]]]]]$, where $v$ is a fresh variable index (not occurring in $\Theta$ as a superscript).

REL is basically CP adjunction.

42. Relativization (REL): $[v_X [\Theta X [wh \Theta Y]]] \rightarrow [v_X [\Theta X [v_X [\Theta X [\Theta X]]]]]

For example, the DP a' girl who every' boy adores has the syntactic representation in (43) below, obtained by an application of REL:

43. [a' girl who every' boy adores] [v [\Theta [wh [v [\Theta [every' boy [v [\Theta [adore [v [\Theta [true]]]]]]]]]]]]

SS is the smallest set of trees that includes DS and is closed under REL; thus, DS$$\subseteq$$SS.

We define an optional rule of Quantifier Raising (QR) (May 1977) which adjoins DPs to IPs or DPs to VPs (we need VP-adjunction for ditransitive verbs among other things) and which is basically the Montagovian Quantifying-In rule.

7
A tree $\Theta$ follows by QR from a tree $\Theta'$ iff: (a) $\Theta'$ is the result of replacing some sub-tree $\Sigma$ of $\Theta$ of the form $[\varphi \ x \ [\varphi \ y]]$ by a tree $[\varphi \ [\varphi \ z]]$, where $v$ is a fresh variable index (not occurring in $\Theta$ as a superscript); or (b) $\Theta'$ is the result of replacing some sub-tree $\Sigma$ of $\Theta$ of the form $[\varphi \ x \ [\varphi \ y]]$ by a tree $[\varphi \ [\varphi \ z]]$, where $v$ is a fresh variable index (not occurring in $\Theta$ as a superscript).

The conditions on the QR rule are that $Z$ is not a pronoun or a wh-word and that $[\varphi \ Z]$ is not a proper sub-tree of a DP sub-tree $[\varphi \ W]$ of $\Sigma$.  

4.1 Translating Basic Expressions

Table (46) below provides examples of basic meanings for the lexical items in (41) above:

- the first column contains the lexical item
- the second column its Dynamic Ty2 translation
- the third column its type, assuming the abbreviations $t := (s(t))$ and $e := (s(e))$

The abbreviated types have exactly the form we would expect them to have in Montague semantics, e.g.:

- the translation of the intransitive verb sleep is of type et
- the translation of the pronoun he is of type (et)t
- the translations of the indefinite article a and the of determiner every are of type (et)(et)t

The list of basic meanings constitutes rule TR0 of our type-driven translation procedure for the English fragment.

46. TR 0: Basic Meanings (TN – Terminal Nodes).

<table>
<thead>
<tr>
<th>Lexical Item</th>
<th>Translation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>sleep</td>
<td>$\lambda v_s. [sleep\ v(v)]$</td>
<td>$e := s(e)$</td>
</tr>
<tr>
<td>own</td>
<td>$\lambda Q, \lambda \alpha, \lambda x, \lambda v, \lambda v' [own\ v (\alpha, [v, v'])]$</td>
<td>$(et)(et)(et)$</td>
</tr>
<tr>
<td>buy</td>
<td>$\lambda Q, \lambda \alpha, \lambda x, \lambda v, \lambda v' [buy\ v (\alpha, [v, v'])]$</td>
<td>$(et)(et)(et)(et)$</td>
</tr>
<tr>
<td>house \text{elf}</td>
<td>$\lambda \alpha, \lambda x, \lambda t, \lambda v, \lambda v' [\text{house}\ -\ elf\ v (\alpha, [t, v, v'])]$</td>
<td>et</td>
</tr>
<tr>
<td>he</td>
<td>$\lambda P. P(u)$</td>
<td>$(et)$</td>
</tr>
<tr>
<td>his</td>
<td>$\lambda P. P(u)$</td>
<td>$(et)$</td>
</tr>
<tr>
<td>Bobby's</td>
<td>$\lambda P. [u</td>
<td>x = \text{Bobby}; P(u)]$</td>
</tr>
<tr>
<td>it</td>
<td>$\lambda P. \lambda P. [P (u)]$</td>
<td>$(et)$</td>
</tr>
<tr>
<td>every</td>
<td>$\lambda P. \lambda P. [\forall v (P(v) \rightarrow P(u))]$</td>
<td>$(et)(et)$</td>
</tr>
</tbody>
</table>
46. TR 0: Basic Meanings (TN – Terminal Nodes).

<table>
<thead>
<tr>
<th>Lexical Item</th>
<th>Translation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[no]₀</td>
<td>( \lambda P_{no} \lambda P_{no} , [\exists {a; , P(s; , P(u))}, , i.e., \lambda P_{no} \lambda P_{no} , [- \exists {a; , P(s; , P(u))}] )</td>
<td>(et)(et)</td>
</tr>
<tr>
<td>[who]₀₁</td>
<td>( \lambda P_{who} , P )</td>
<td>(et)</td>
</tr>
<tr>
<td>[P]₀₁ / [all]₁ / [the]₁</td>
<td>( \lambda D_{b} , D )</td>
<td>(e)</td>
</tr>
<tr>
<td>[doesn't]₀₁ / [doesn't]₁</td>
<td>( \lambda D_{b} , [-] )</td>
<td>(e)</td>
</tr>
<tr>
<td>[Ø]₀</td>
<td>( \lambda D_{b} , [\exists {D' \rightarrow D}] )</td>
<td>(t)</td>
</tr>
<tr>
<td>[and]₀</td>
<td>( \lambda D_{b} , D ; D )</td>
<td>(t)</td>
</tr>
<tr>
<td>[or]₀</td>
<td>( \lambda D_{b} , D ; [D' \vee D] )</td>
<td>(t)</td>
</tr>
</tbody>
</table>

- Transitive verbs like own take a generalized quantifier (GQ) as their direct object (type (et)(t)), which captures the fact that the default quantifier scoping is subject over object.
- Inverse scope is obtained by QR.
- Ditransitive verbs like buy are assumed to take two GQs as objects.
- The default relative scope of the two GQs (encoded in the lexical entry) is their left-to-right surface order, i.e., the first of them (e.g., the Dative GQ) takes scope over the second (e.g., the Accusative GQ).
- This seems to be empirically correct since the most salient quantifier scoping in the sentence Dobby bought every witch an alligator purse follows the left-to-right linear order: the Dative GQ takes scope over the Accusative GQ, so that the purses co-vary with the witches.
- Once again, inverse scope is obtained by QR (to VP or IP).
- The Dative GQ takes scope over the Accusative GQ despite their relative syntactic position: given the phrase structure rules PS8 and PS9 in (41) above, the Dative GQ is actually c-commanded by the Accusative GQ.
- The fact that a quantifier can take scope over another without c-commanding it is one of the advantages of working with a dynamic system, where quantifier scope is encoded in the order in which the updates are sequenced.
- Thus, in a dynamic framework, syntactic structure affects quantifier scope only to the extent to which syntactic relations (e.g., c-command) are ultimately reflected in update sequencing.
- The lexical entry for ditransitive verbs in (46) 'neutralizes' syntactic c-command: it sequences the updates contributed by the two GQ objects according to their linear order and not according to their syntactic structure.
- Defaulting to linear order (as opposed to syntactic c-command) has welcome empirical consequences: besides the fact that we capture the correlation between linear order and quantifier scope, we can also account for the fact that the Dative GQ is able to bind pronouns within the Accusative GQ without c-commanding them, as for example in Dobby gave every witch her broom.
- It is not unexpected that a dynamic system can account for pronoun binding without c-command given that donkey anaphora is a paradigmatic example of such binding without c-command.
- Pronouns of the form \( b_0 \) and traces of the form \( t_1 \) are interpreted as in Montague (1974), i.e., as the GQ-lift of their index.
- For pronouns, this index is a dref (i.e., a constant of type \( e := s \)).
- For traces, this index is a variable (also of type \( e := s \)).
- Proper names are analyzed as indefinites – see the discussion above, in particular (29).
- The only difference is that they are now translated as the corresponding GQ-lift.
- Indefinites have the type of (dynamic) generalized determiners, as needed for the definition of the compositional interpretation procedure.
- Their crucial dynamic contribution: the introduction of a new dref, which has to satisfy the restrictor property and the nuclear scope property in this order.
- The DLP-style abbreviation explicitly exhibits the existential quantification built into the indefinite.
- Every also has the type of generalized determiners and it is interpreted as expected.
- Note the square brackets surrounding the DRS – they are due to the fact that, unlike the indefinite determiner \( a \), the universal determiner \( \forall \) contributes a test – it is internally dynamic but externally static, just as in DRT / FCS / DPL.
- No also contributes a test.
- The wh-words that enter the construction of relative clauses are analyzed as identity functions over the property contributed by the relative clause.
- This property will then be 'sequenced' with the property contributed by the preceding common noun to yield a 'conjoined' property that is a suitable argument for a generalized determiner.
- The order in which the common noun and the relative clause are sequenced follows the linear surface order.
- The rule that achieves this dynamic 'conjunction' / 'sequencing' of properties generalizes both the static predicate modification rule in Heim & Kratzer (1998) and the dynamic Sequencing rule in Muskens (1996) – see (49) below.
- Non-negative inflectional heads are interpreted as identity functions over DRS meanings.
- Negative inflectional heads are interpreted as expected: their value is a test, containing a condition that negates the argument DRS.
the conditional if is a binary DRS connective: it takes two DRSs as arguments and it returns a test containing a dynamic implication condition that relates the two argument DRSs

- the coordinating elements and and or will be discussed in more detail later on.

4.2. Translating Complex Expressions

Based on TR0, we can obtain the translation of more complex LF structures by specifying how the translation of a mother node depends on the translations of its daughters.

- there are five such rules, the last of which (TR5: Coordination) will be subsequently generalized

47. TR 1 – Non-branching Nodes (NN).
   If A \sim \alpha and A is the only daughter of B, then B \sim \alpha.

48. TR 2 – Functional Application (FA).
   If A \sim \alpha and B \sim \beta and A and B are the only daughters of C, then C \sim \alpha(\beta), provided that this is a well-formed term.

49. TR 3 – Generalized Sequencing (GSeq) (i.e., Sequencing + Predicate Modification).
   If A \sim \alpha, B \sim \beta, A and B are the only daughters of C in that order (i.e., C \rightarrow A B) and \alpha and \beta are of the same type \tau of the form t or \sigma t (for any \sigma \in \text{Typ}) then C \sim \alpha; \beta if t=\tau or C \sim \lambda x. \alpha(t); \beta(t), if t=\sigma t, provided that this is a well-formed term.

50. TR 4 – Quantifying-In (QIA).
   If DP \sim \alpha, B \sim \beta and DP and B are daughters of C, then C \sim \alpha(\lambda x. \beta), provided that this is a well-formed term.

51. TR 5 – Coordination (Co).
   If A1, Conj \sim \alpha1, A2 \sim \alpha2 and A1, Conj and A2 are the only daughters of A in that order (i.e., A \rightarrow A1 Conj A2), then A \sim \beta(\alpha1(\alpha2)), provided this a well-formed term and has the same type as \alpha1 and \alpha2.

- the first rule covers non-branching nodes: the mother inherits the translation of the daughter
- the second rule is functional application: the translation of the mother is the result of applying the translation of one daughter to the translation of the other
- the third rule is a generalized sequencing (i.e., a generalized dynamic conjunction) rule
- it translates the meaning of complex texts (Txt) that are formed out of a text (Txt) and a sentence (CP) – see PS1 in (41) above; in this sense, it is a generalization of the Sequencing rule in Muskens (1996)
- it also handles predicate modification in general, e.g., it translates the meaning of an NP that is formed out of a common noun N and a relative clause CP – see PS11 in (41) above; in this sense, it is a generalization of the Predicate Modification rule in Heim & Kratzer (1998)

- the fourth rule handles Quantifying-In, both for quantifiers and for relativizers (i.e., wh-words)
- the fifth rule handles binary coordinations (to be generalized to an arbitrary finite number of coordinated elements)

The translation procedure, i.e., the relation 'tree \Theta translates as term \alpha', is formally defined as the smallest relation \sim between trees and Dynamic Ty2 terms that is conform to TR0-TR5 and is closed under type-logical identity, e.g., if tree \Theta translates as term \alpha and \alpha=\beta is true, then \Theta translates as \beta.

5. Anaphora and Quantification in Compositional DRT (CDRT)

5.1. Bound Variable Anaphora

We can capture bound anaphora in CDRT without using QR (Quantifier Raising, see (44) above) and the corresponding semantic rule QIn (Quantifying-In, see (50) above): we simply need the pronoun to be co-indexed with the antecedent, as shown in (52).

52. Every\textsuperscript{u} house-elf hates himself.

- co-indexation is enough for binding because binding in CDRT (or DPL) is taken care of by the explicit (and, in this case, unselective) quantification over 'assignments' built into the meaning of quantifiers
- if we want to obtain bound variable anaphora in a static system, co-indexation, i.e., using the same variable, is not enough: we also need to create a suitable \lambda-abstraction configuration that will ensure the semantic co-variation via selective quantification over assignments (what particular clause in the above definition of Dynamic Ty2 requires that?)

Sentence (52) receives the Dynamic Ty2 representation in (53) below – or, equivalently, the one in (54). The formulas deliver the intuitively correct truth-conditions in (55).

53. \([u1; h.elf(u1)] \rightarrow [hate(u1, u1)]\)
54. \(\forall u1[\lambda x.(h.elf(x)) \rightarrow [hate(u1, u1)]]\)
55. \(\lambda x. \forall u1[h.elf(x) \rightarrow hate(x, x)]\)

CDRT associates the correct Dynamic Ty2 translation with sentence (52) in a compositional way, as shown by the LF in (56) below.
56. Every\textsuperscript{nu} house-elf hates himself\textsubscript{nu}.

The two LFs are provided in (62) and (63) below.

57. Every\textsuperscript{nu} house-elf adores a witch.

62. every\textsuperscript{nu}, a\textsuperscript{nu} Every\textsuperscript{nu} house-elf adores a\textsuperscript{nu} witch.

What type-driven translation rules are applied at various points in the derivation?

What is the precise point where static Montagovian semantics and our CDRT semantics diverge? Why does CDRT semantics go through?

5.2. Quantifier Scope Ambiguities

The typical example in (57) below is ambiguous between two quantifier scopings:

- the surface scope every\textsuperscript{nu}, >> a\textsuperscript{nu},
- the inverse scope a\textsuperscript{nu}, >> every\textsuperscript{nu}, obtained by an application of QR

The two LFs yield the translations in (58) and (60) below, which capture the intuitively correct truth-conditions for the two readings, as shown in (59) and (61).

61. a\textsuperscript{nu}, >> every\textsuperscript{nu} : [\forall x.(h.elf(x) → adore(x, y))]
53. Quantifier Scope with Ditransitive Verbs

Consider the sentence in (64) below: the Dative GQ takes scope over and binds into the Accusative GQ – without c-commanding it.

64. [a\textsuperscript{wii} witch] give every\textsuperscript{wii} witch her\textsubscript{wii} alligator purse.

This example simultaneously exhibits two of the most interesting aspects of CDRT:

- we can have binding of pronouns without c-command and without QR, i.e., without the covert syntactic manipulations associated with the level of LF
- a quantifier can have scope over another without c-commanding it as long as the update it contributes is sequenced before the update of the other quantifier: the lexical entry for ditransitive verbs specifies that the Dative GQ update is sequenced / takes scope over the Accusative GQ update – and this is enough to nullify the fact that, syntactically, the former does not take scope over the latter
- both features of CDRT point to the fact that a finer-grained semantics should enable us to simplify the syntactic structure that we need as input to the semantic interpretation

procedure – for example, the dominance relations that the LF level encodes are not always relevant for interpretation

Following the simplified LF for possessive DPs proposed in Heim & Kratzer (1998),\textsuperscript{14} I analyze her alligator purse as the DP in (65) below.\textsuperscript{13}

65. [\text{DP} a\textsuperscript{wii} [\{a alligator purse\} [\text{of her\textsubscript{wii} ] ] ] ]

The preposition receives a translation similar to transitive verbs like own.

66. [\text{DP} \rightsquigarrow \lambda \text{Q}_{\text{acc}}. \lambda v. Q(\lambda v'. [\text{of}\textsubscript{acc}(v, v')])]

We compositionally derive the following translation for the DP in (65) (the subscript on the symbol \rightsquigarrow indicates the rule applied in translating the mother node):

67. a. [\text{of her\textsubscript{wii} }] \rightsquigarrow_{\text{E}} \lambda v. [\text{of}(v, u_1)]

b. [\text{of a purse}] \rightsquigarrow_{\text{Gw}} \lambda v. [\text{of}(\text{a purse}, v, u_1)]

c. (65) \rightsquigarrow_{\text{E}} \lambda \text{L}_{\text{acc}}. [u_2 | \text{a purse}(u_2), \text{of}(u_2, u_1)]; P(u_2)

The syntactic structure of the V\textsuperscript{vii} is provided in linearized form in (68) and is compositionally translated in (69).

The Dative GQ every\textsuperscript{wii} witch takes scope over the Accusative GQ and also binds the pronoun her\textsubscript{wii} contained in it.

68. [\text{DP} a\textsuperscript{wii} every\textsuperscript{wii} witch] [\{a alligator purse\] [\text{of her\textsubscript{wii} ] ] ]

69. [\text{DP} give every\textsuperscript{wii} witch] [\{a alligator purse\} [\text{of her\textsubscript{wii} ] ] ]

(68) \rightsquigarrow_{\text{E}} \lambda \text{L}_{\text{acc}}. [u_2 | \text{a purse}(u_2), \text{of}(u_2, u_1)]; P(u_2)

Sentence (64) is translated as shown in (70). It receives the intuitively correct truth-conditions (for its most salient reading) in (71).

70. [u_2 | u_2 = Dobby, [u_1 | \text{witch}(u_1)] \rightarrow [u_2 | \text{a purse}(u_2), \text{of}(u_2, u_1), \text{give}(u_2, u_1, u_2)]]

71. \lambda i. \exists x \exists y (z = \text{dobby} \land \forall x (\text{witch}(x) \rightarrow \exists y (\text{a purse}(y) \land \text{of}(x, y) \land \text{give}(z, x, y))),

i.e., \lambda i. \forall x (\text{witch}(x) \rightarrow \exists y (\text{a purse}(y) \land \text{of}(x, y) \land \text{give}(\text{dobby}, x, y)))

\textsuperscript{14} Although the LF in (65) is similar to the one in Heim & Kratzer (1998), the analysis is different: while Heim & Kratzer (1998) take possessives to be covertly definite DPs (and adopt a Fregean analysis of definite descriptions), I analyze them here as covertly indefinite DPs. The indefinite analysis of possessive DPs is empirically supported by the interpretation of possessives in predicative positions, e.g., John is her / Mary’s brother, which are not associated with uniqueness implications.

\textsuperscript{13} I assume that the following phrase structure and lexical insertion rules are added to the syntax of our English fragment: (PS 13) NP \rightarrow N (PP); (PS 14) PP \rightarrow P DP; (LI 12) P \rightarrow of.
5.4. Cross-sentential Anaphora

72. A house-elf fell in love with a witch.

73. He bought her an alligator purse.

5.5. Relative-clause Donkey Sentences

74. Every house-elf who falls in love with a witch buys her an alligator purse.
5.6. Conditional Donkey Sentences

75. If a\textsuperscript{h} house-elf falls in love with a\textsuperscript{h} witch, he\textsubscript{u} buys her\textsubscript{u}, an\textsuperscript{h} attillator purse.

\[
\text{CP} \\
\text{IP}
\]

\[
\lambda D_1 [[u_1, u_2 | h.elf[u_1], witch[u_2], in.love[u_1, u_2]] \rightarrow [u_1 | a.purse(u_1), buy[u_1, u_2, u_2]]]
\]

\[
\lambda D_1; !D_2 [D' \rightarrow D] \\
\lambda D_1; !D_2 [D' \rightarrow D]
\]

\[
[u_1, u_2 | h.elf[u_1], witch[u_2], in.love[u_1, u_2]]
\]

\text{a\textsuperscript{h} house-elf falls in love with a\textsuperscript{h} witch}

Show that the sentences in (76) and (77) below, which involve sentence-coordination structures, are compositionally assigned the intuitively correct interpretation

76. If a\textsuperscript{h} house-elf falls in love with a\textsuperscript{h} witch and she\textsubscript{u} likes fancy handbags, he\textsubscript{u} buys her\textsubscript{u}, an\textsuperscript{h} attillator purse.

77. If a\textsuperscript{h} farmer owns a\textsuperscript{h} donkey, he\textsubscript{u} beats it\textsubscript{u}, or he\textsubscript{u} feeds it\textsubscript{u}, poorly.

6. Translating Unselective Quantification into Dynamic Ty2

Consider again the DPL-style definition of unselective generalized quantification:

78. \[
| \text{det}(\phi, \psi) | = \{ h : \forall g, h \vdash g = h \text{ and } \text{DET}(\phi)^g, \text{Dom}(\psi) \}
\]

where \text{DET} is the corresponding static determiner and \[
\phi^g = \{ h : \exists \psi, h \vdash \psi \}
\]

and \[
\text{Dom}(\phi^g) = \{ h : \exists \psi, h \vdash \psi \}
\]

79. \[
\text{det}(\phi, \psi) = \text{det}(\chi) ; \phi, \psi
\]

In particular:

80. \[
| \text{every}(\phi, \psi) | = \{ h : \forall g, h \vdash g = h \text{ and } \text{EVERY}(\chi) ; \phi^g, \text{Dom}(\psi) \}
\]

i.e., \[
| \text{every}(\phi, \psi) | = \{ h : \forall g, h \vdash g = h \text{ and } (\exists \chi ; \phi)^g \subseteq \text{Dom}(\psi) \}
\]

81. \[
\text{no}(\phi, \psi) = \{ h : \forall g, h \vdash g = h \text{ and } \text{NO}(\chi) ; \phi^g, \text{Dom}(\psi) \}
\]

82. \[
\forall \psi (\phi \rightarrow \psi) \iff \text{exists}(\phi) \rightarrow \psi \iff (\exists \chi ; \phi) \rightarrow \psi \iff \text{every}(\phi, \psi)
\]

83. \[
\neg \exists \chi (\phi ; \psi) \iff \forall \psi (\phi \rightarrow \neg \psi) \iff (\exists \chi ; \phi) \rightarrow \text{no}(\psi, \psi)
\]

Given that the above DPL formulas are tests, they will be translated in Dynamic Ty2 as conditions, i.e., as terms of type \text{st}.

84. \[
\text{det}(D, D') := \lambda \phi, \psi. \text{DET}(\langle D, \text{Dom}(D') \rangle)
\]

where \text{DET} is the corresponding static determiner, \[
\langle D, \text{Dom}(D') \rangle := \{ h : [h; D, D'] \}
\]

85. \[
\text{det}(D, D') := \text{det}(\langle \emptyset \rangle ; D, D')
\]

In particular:

86. \[
\text{every}(D, D') = \lambda \phi, \psi. \text{EVERY}(\langle \emptyset \rangle ; D, \text{Dom}(D'))
\]

i.e., \[
\text{every}(D, D') = \lambda \phi, \psi. \forall \psi (\phi \rightarrow \psi)
\]

87. \[
\text{no}(D, D') = \lambda \phi, \psi. \text{NO}(\langle \emptyset \rangle ; D, \text{Dom}(D'))
\]

i.e., \[
\text{no}(D, D') = \lambda \phi, \psi. \forall \psi (\phi \rightarrow \neg \psi)
\]

88. \[
\text{ Det}(D, D') = \forall \psi (D \rightarrow D') \iff \forall \psi (\langle \emptyset \rangle ; D, D') \]

89. \[
\text{ Det}(D, D') = \forall \psi (D \rightarrow D') \iff \forall \psi (\langle \emptyset \rangle ; D, D')
\]

6.1. Limitations of Unselectivity: Proportions

90. \[
\text{most}(D, D') = \lambda \phi, \psi. \text{MOST}(\langle \emptyset \rangle ; D, \text{Dom}(D'))
\]

i.e., \[
\text{most}(D, D') = \lambda \phi, \psi. \forall \psi (\phi \rightarrow \psi)
\]

91. Most\textsuperscript{h} house-elves who fall in love with a\textsuperscript{h} witch buy her\textsubscript{u}, an\textsuperscript{h} attillator purse.

92. \[
\text{most}(\{u_1 | h.elf[u_1], witch[u_2], in.love[u_1, u_2], [u_1 | a.purse(u_1)], buy[u_1, u_2, u_3]\})
\]

93. \[
\lambda i, u_1, u_2 | h.elf[u_1], witch[u_2], in.love[u_1, u_2]] (\text{[a.purse]} u_1, buy[u_1, u_2, u_3])
\]

94. \[
\forall u_1, u_2, u_3 | h.elf[u_1], witch[u_2], in.love[u_1, u_2]] (\text{[a.purse]} u_1, buy[u_1, u_2, u_3])
\]

95. Every\textsuperscript{h} person who has a\textsuperscript{h} dime will put it\textsubscript{u} in the meter.

96. \[
\text{every}(\{u_1, u_2 | person[u_1], dime[u_2], have[u_1, u_2]\}, \text{put in meter}[u_1, u_2])
\]
97. \( \lambda \alpha. \{ [u_1, u_2] \mid \alpha [u_1], \text{dime}[u_2], \text{have}[u_1, u_2] \} \subseteq \{ [u_1, u_2] \mid \alpha [u_1], \text{dime}[u_2], \text{have}[u_1, u_2] \} \), i.e., by Axioms 3 and 4 ("Identity of 'assignments'" and "Enough 'assignments'").

98. The conservative definitions of unselective generalized quantification based on the non-conservative ones in (84) and (85) above are provided in (99) and (100) below.

99. **Built-in 'unselective' dynamic conservativity:**

\[
\text{det}(D, D') := \lambda \alpha. \text{DET}(\{u\}; (\{u\}; D, D'; \{u\}; D', D'))
\]

100. **Unselective generalized quantification with built-in dynamic conservativity:**

\[
\text{det}(D, D') := \lambda \alpha. \text{DET}((\{u\}; D, D'; \{u\}; D', D'))
\]

Given that the definition of conservative unselective quantification in (100) provides access to the dref \( u \) in both the restrictor and the nuclear scope of the quantification, we will use it as the basis for the CDRT definition of selective generalized quantification.

7. **Translating Selective Quantification into Dynamic Ty2**

The syntax for selective generalized quantification – the same as above:

- we use abbreviations of the form \( \text{det}(D, D') \)
- \( u \) is the 'bound' dref (recall that \( u \) is a constant of type \( \epsilon := \text{se} \), so it cannot possibly be bound in Dynamic Ty2 – hence the scare quotes on 'bound')
- \( D \) is the restrictor
- \( D' \) is the nuclear scope of the quantification.

The selective determiner \( \text{det} \), relates two sets of individuals (type \( \epsilon \)) and not two sets of 'assignments' (type \( \text{st} \)), as the unselective determiner \( \text{det} \) does:

- the fact that \( \text{det} \), relates sets of individuals solves the proportion problem

- as far as weak / strong donkey ambiguities are concerned, we account for them by means of two meanings for generalized determiners: a weak meaning \( \text{det}^w(D, D') \) and a strong meaning \( \text{det}^s(D, D') \)
- both meanings are defined in terms of the static determiner \( \text{DET} \) and both of them are conditions, i.e., terms of type \( \text{st} \)

\[
\text{det}^w(D, D') := \lambda \alpha. \text{DET}(\{u\}; \text{st}(D, D'; \{u\}; D', D'))
\]

\[
\text{det}^s(D, D') := \lambda \alpha. \text{DET}((\{u\}; D, D'; \{u\}; D', D'))
\]

where \( \text{st} := \{ [u] \} \) and

\[
\text{st}(D, D') := \lambda \alpha. \text{DET}((\{u\}; D, D'; \{u\}; D', D'))
\]

101. **Unselective generalized quantification with built-in dynamic conservativity:**

\[
\text{det}(D, D') := \lambda \alpha. \text{DET}(\{u\}; (\{u\}; D, D'; \{u\}; D', D'))
\]

where \( \text{st} := \{ [u] \} \) and

\[
\text{st}(D, D') := \lambda \alpha. \text{DET}((\{u\}; D, D'; \{u\}; D', D'))
\]

102. Every\(^{st} \) farmer who owns a\(^{st} \) donkey beats it.\(^{st} \)

The weak and strong meanings for every\(^{st} \) are provided in (104) and simplified in (105).

\[\text{every}^w(D, D') := \lambda \alpha. \text{EVERY}(\{u\}; \text{st}(D, D'; \{u\}; D', D'))\]

\[\text{every}^s(D, D') := \lambda \alpha. \text{EVERY}(\{u\}; \text{st}(D, D'; \{u\}; D', D'))\]

The weak reading of sentence (103):

\[\text{every}^w(u, \{ [u] \} \text{farmer} [u_1], \text{donkey} [u_2], \text{own} [u_1, u_2], \text{beat} [u_1, u_2])\]

103. Every\(^{st} \) farmer who owns a\(^{st} \) donkey beats it.\(^{st} \)

The weak and strong meanings for every\(^{st} \) are provided in (104) and simplified in (105).

\[\text{every}^w(D, D') := \lambda \alpha. \text{EVERY}(\{u\}; \text{st}(D, D'; \{u\}; D', D'))\]

\[\text{every}^s(D, D') := \lambda \alpha. \text{EVERY}(\{u\}; \text{st}(D, D'; \{u\}; D', D'))\]

The weak reading of sentence (103):

\[\text{every}^w(u, \{ [u] \} \text{farmer} [u_1], \text{donkey} [u_2], \text{own} [u_1, u_2], \text{beat} [u_1, u_2])\]
107. \( \lambda u_1 \lambda u_2 \lambda u_3 \lambda x. \{ [u_1, \text{farmer}(u_1), \text{donkey}(u_2), \text{own}(u_3, u_3) | u_2, \text{beat}(u_2, u_3)] \} \subseteq \)
\( \lambda u_1 \lambda u_2 \lambda u_3, \text{own}(u_3, u_3), \text{beat}(u_2, u_3) \), i.e.,
\( \lambda x_1. \{ [x_1, \text{farmer}(x_1) \land \exists y_1 (\text{donkey}(y_1) \land \text{own}(y_1, y_1)) \} \subseteq \)
\( \{ [x_1, \text{farmer}(x_1) \land \exists y_1 (\text{donkey}(y_1) \land \text{own}(y_1, x_1)) \} \}, \text{i.e.} \)
\( \lambda x_1. \forall z_1 (\text{farmer}(x_1) \land \exists y_1 (\text{donkey}(y_1) \land \text{own}(y_1, z_1))) \times \text{beat}(z_1, x_1)) \)

The strong reading of sentence (103):

108. \([u_1, \text{farmer}(u_1), \text{donkey}(u_2), \text{own}(u_3, u_3), \text{beat}(u_2, u_3)]\)

109. \([u_1, \text{farmer}(u_1), \text{donkey}(u_2), \text{own}(u_3, u_3), \text{beat}(u_2, u_3)]\)

7.2 Solving Proportions

110. most\(^{\text{SR}}\)(D, D') := \( \lambda u_1. \text{MOST}(u_1[D], \text{most}(D, D')) \)

111. House-elves who fall in love with a house-elf buy her an alligator purse.

112. most\(^{\text{SR}}\)(D, D') := \( \lambda u_1. \text{most}(D, D') \text{ of } u_1[D] \times u_1[D'] \)

8 Extending CDRT with Generalized Quantification (CDRT+GQ)

The syntax is the same. As far as the semantics is concerned, we only need to:

- replace the CDRT meanings for generalized determiners with the newly defined selective generalized determiners;
- replace the CDRT meaning for dynamic implication with the newly defined unselective generalized determiners; thus, CDRT+GQ can capture account for conditionals with overt adverbs of quantification (of course, assigning an unselective meaning to conditionals fails to account for the fact that they also exhibit weak/strong donkey ambiguities).

The CDRT+GQ meanings have the same types as the corresponding CDRT meanings, i.e., (et)(ettt) for determiners and (tttt) for conditionals/adverbs of quantification.

The meaning of the indefinite determiner a remains the same as in CDRT (redefining it in terms of selective generalized quantification would make it a test).

117. TR 0 (only the revised entries are listed): Basic Meanings (TN).

<table>
<thead>
<tr>
<th>Lexical Item</th>
<th>Translation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>every(^{\text{SR}})</td>
<td>( \lambda x. \text{every}(D, D') \times \text{det}(D, D') )</td>
<td>(et)(ettt)</td>
</tr>
<tr>
<td>most(^{\text{SR}})</td>
<td>( \lambda x. \text{most}(D, D') \times \text{det}(D, D') )</td>
<td>(tttt)</td>
</tr>
<tr>
<td>put_in_meter(^{\text{SR}})</td>
<td>( \lambda x. \text{put_in_meter}(D, D') \times \text{det}(D, D') )</td>
<td>(tttt)</td>
</tr>
</tbody>
</table>

16
8.1. Proportions and Weak / Strong Ambiguities in CDRT+GQ

118. Most house-elves who fall in love with a witch buy her an alligator purse.

9. Anaphora and Generalized Coordination in CDRT+GQ

Goals:
- provide the dynamic counterparts of the definitions of conjoolvable types and generalized conjunction and disjunction in Partee & Rooth (1983);
- show that CDRT / CDRT+GQ can account for the DP-conjunction donkey example in (119) below, from Chierchia (1995): 77, (38).
- this is one of the central examples used in Chierchia (1995) to argue for an approach to natural language that builds (part of) the dynamics into the semantic value of natural language expressions as opposed to syntactic operations on the LF of sentences / discourses
- so, mutatis mutandis, his argument that discourse dynamics should be captured semantically and not syntactically also supports CDRT+GQ.

9.1. Generalized Dynamic Conjunction and Disjunction

120. Dynamically Conjoinable Types (DCTyp).
The set of dynamically conjoinable types DCTyp is the smallest subset of Typ s.t. \( t \in \text{DCTyp} \) (where \( t = \sigma(s(t)) \)) and, if \( t \in \text{DCTyp} \), then \( \sigma(s(t)) \in \text{DCTyp} \) for any \( \sigma \in \text{Typ} \).

121. Generalized Pointwise Dynamic Conjunction \( \sqcap \) and Disjunction \( \sqcup \).
For any two terms \( \alpha \) and \( \beta \) of type \( t \), for any \( t \in \text{DCTyp} \):
- \( \alpha \sqcap \beta := (\alpha, \beta) \) if \( t = t \) and \( \alpha \sqcap \beta := \lambda \alpha, \alpha(v) \sqcap \beta(v) \) if \( t = \tau(\alpha) \);
- \( \alpha \sqcup \beta := [\alpha \lor \beta] \) if \( t = t \) and \( \alpha \sqcup \beta := \lambda \alpha, \alpha(v) \sqcup \beta(v) \) if \( t = \tau(\alpha) \).

Abbreviation. \( \alpha_1 \sqcap \alpha_2 \sqcap \ldots \sqcap \alpha_n := (\ldots (\alpha_1 \sqcap \alpha_2) \sqcap \ldots) \sqcap \alpha_n \);
\( \alpha_1 \sqcup \alpha_2 \sqcup \ldots \sqcup \alpha_n := (\ldots (\alpha_1 \sqcup \alpha_2) \sqcup \ldots) \sqcup \alpha_n \).

The translation rule GSeq (Generalized Sequencing) we have introduced in chapter 3 above is simply a restricted form of generalized dynamic conjunction \( \sqcap \).

The basic meanings for \( \text{and} \) and \( \text{or} \):

122. TR 0 (only the revised entries are listed): Basic Meanings (TN).

<table>
<thead>
<tr>
<th>Lexical Item</th>
<th>Translation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[and]_{\text{CQ}}</td>
<td>( \lambda v_1 \ldots \lambda v_n ! v_1 \sqcap \ldots \sqcap v_n )</td>
<td>( \tau(\ldots (\tau(\ldots (t_1 \lor \ldots )\ldots )\ldots )) )</td>
</tr>
<tr>
<td>[or]_{\text{CQ}}</td>
<td>( \lambda v_1 \ldots \lambda v_n ! v_1 \lor \ldots \lor v_n )</td>
<td>( \tau(\ldots (\tau(\ldots (t_1 \land \ldots )\ldots )\ldots )) )</td>
</tr>
</tbody>
</table>
9.2. Revising the Coordination Rule: Generalized Coordination

123. TR 5 (revised) – Generalized Coordination (GCo).

If \( A_1 \sim a_1, \ldots , A_n \sim a_n, \text{Conj} \sim \beta, A_{n+1} \sim a_{n+1} \) and \( A_1, \ldots , A_n, \text{Conj} \) and \( A_{n+1} \) are the only daughters of \( A \) in that order (i.e., \( A \rightarrow A_1 \ldots A_n \text{Conj} A_{n+1} \)), then \( A \sim \beta(a_1) \ldots (a_{n+1}) \), provided this a well-formed term and has the same type as \( a_1, \ldots , a_n, a_{n+1} \).

9.3. Catching and Eating a Fish in CDRT+GQ

Consider the sentences in (124) and (125) (Partee & Rooth 1983: 338, (12) and (13)).

124. John caught and ate a\(^9\) fish.

125. John hugged and kissed three\(^9\) women.

Under the most salient reading of sentence (124), John catches and eats the same fish. Similarly for (125), John hugs and kisses the same three women.

We can obtain this reading in CDRT / CDRT+GQ only by quantifying-in the direct object indefinite a\(^9\) fish. That is, CDRT+GQ incorrectly predicts that the default reading (the one without quantifying-in) should be one in which the fish that John catches and the fish that John eats are possibly different.

This is a consequence of the fact that, following Montague, transitive verbs are interpreted as taking a Q as direct object (a term of type \( \epsilon \in \text{it} \)) and not an individual dref (a term of type \( \epsilon \)) – recall that the empirical motivation for this was that the preferred relative scope of the subject and direct object is the one in which the subject scopes over the object.

What to do? Here's a possibility: translate transitive verbs as binary relations and give a type-shifting semantics for case – e.g., Accusative case on quantifiers in direct object position 'glues' together such quantifiers and transitive verbs.

What would an appropriate meaning for Accusative case be in static Montagovian semantics? How about CDRT?

The two for sentence (124) are schematically represented in (126) and (129) below, together with their respective translations and truth conditions.

126. John\(^9\) \( u \) [\( u \) caught and ate a\(^9\) fish.

127. \([u_2 \mid u=\text{John}]; [u_1 \mid \text{fish}[u_1], \text{catch}(u_2, u_1)]; [u_2 \mid \text{fish}[u_1], \text{eat}(u_2, u_1)]\]

128. \( \lambda w_1. \exists x (\text{fish}(w_x) \land \text{catch}(\text{John}, w_x)) \lor \exists y (\text{fish}(w_y) \land \text{eat}(\text{John}, w_y)) \)

129. \( [a^9 \mid \text{fish}]^\theta \) [\( \text{John}^\theta \mid [u \mid \text{caught and ate}] \theta \cdot \]

130. \( [u_1 \mid \text{fish}[u_1]]; [u_2 \mid u=\text{John}]; [\text{catch}(u_2, u_1)], [\text{eat}(u_2, u_1)]\), i.e., \( [u_1, u_2 \mid \text{fish}[u_1], u=\text{John}, \text{catch}(u_2, u_1), \text{eat}(u_2, u_1)]\)

131. \( \lambda w_1. \exists x (\text{fish}(w_x) \land \text{catch}(\text{John}, x) \land \text{eat}(\text{John}, x)) \)

9.4. Coordination and Discourse Referent Reassignment

The 'possibly distinct fish' representation in (127) above and its interpretation are unlike anything in classical DRT / FCS:

- reintroducing a dref in DRT / FCS, e.g., dref \( u \) in (127), is either banned or, if it is allowed, it is \textit{not} interpreted as reassigning a value to that dref – the output info state assigns the same value to the dref as the input info state
- in contrast, CDRT+GQ allows dref reintroduction and interprets it as reassignment of value to the dref

It seems that classical DRT / FCS is empirically better than CDRT+GQ, since the surface-based representation in (127) yields the 'same fish' interpretation in a DRT / FCS-like system.

But we cannot easily obtain a representation of the 'distinct fish' interpretation, which is the preferred one for conjunctions of intensional transitive verbs, e.g., \( \text{John needed and bought a new coat} \) (Partee & Rooth 1983: 338 observe).

We would have to postulate a mechanism whereby the indefinite object \( a \) \textit{fish} occurs twice in the LF of sentence (124) and contributes distinct drefs – i.e., we would have to syntactically simulate the semantic Montagovian analysis in (127) above.

Moreover, the necessary syntactic operations on LFs become increasingly stipulative as soon as we turn to more complex examples like the coordination donkey sentence in (119) above – which receives a straightforward reassignment-based analysis in CDRT+GQ.

Thus, it seems that we need some form of dref reassignment.

The formalization of dref reassignment in CDRT+GQ (or DPL) ultimately makes incorrect predictions – because reassignment is destructive: the previous value of the dref is lost and cannot be later accessed in discourse.

Consider for example the DP conjunction \( \text{Mary and Helen} \) in discourse (132-133) below.

132. Mary\(^9\) and Helen\(^9\), (each) bought an\(^9\) alligator purse.

133. They\(_a\) were (both) fluorescent green.

134. \([u_1, u_2 \mid u=\text{Mary, a.purse}(u_1), \text{buy}(u_1, u_2)]; [u_2, u_3 \mid u=\text{Helen, a.purse}(u_1), \text{buy}(u_2, u_3)]; [\text{fluorescent_green}(u_3)]\]

Page references are to Partee & Portner (2002).
Under the most salient reading of (132), Mary and Helen buy a purse each.

But if we analyze this sentence as shown in (134), we are able to retrieve only the purse mentioned last, i.e., Helen's purse: the destructive CDRT+GQ reassignment renders Mary's purse inaccessible for subsequent anaphora.

Summary of the problem:
- we need to provide an account of the interaction between anaphora and generalized coordination exhibited by sentence (119) and, for that, we need to allow for dref reintroduction – or, more exactly, index reusability – so that both donkey indefinite \( a^u \)
  - dog and \( a^u \) cat can be anaphorically associated with the donkey pronoun \( i^u \).
- the only way to capture index reusability in CDRT+GQ is as dref reintroduction, i.e., as destructive random (re)assignment.

Solution: index reusability does not have to be interpreted as destructive reassignment

We could in principle associate a new value with a previously used index while, at the same time, saving the old value for later retrieval by associating it with another index.

This idea can be implemented in various ways, e.g., by taking information states to be referent systems (see e.g., Vermeulen 1993 and Groenendijk, Stokhof & Veltman 1996) or stacks (see e.g., Dekker 1994, van Eijck 2001, Nouwen 2003 or Bittner 2006) – and not total variable assignments as in DPL.

Such information states are formally more complex than our current ones, so we will continue to employ total 'variable assignments' and the current notion of (destructive) random assignment.

9.5. Anaphora across VP- and DP-Conjunctions

The following three sentences are from Muskens (1996): 177-180, (52), (54) and (58).

135. A\(^v\) cat \( \{\{v\text{-}caught \, a^v \text{ fish}\} \text{ and } \{v\text{-}ate \, it^v \}\} \).

136. \([u^1, u^2] \text{ cat } [u^1, u^2], \text{ fish} [u^1, u^2], \text{ catch} [u^1, u^2] \).

137. \( \lambda z. \exists x, \exists y, (\text{cat}(x) \land \text{fish}(y) \land \text{catch}(x, y) \land \text{eat}(z, y)) \).

138. John\(^u\) has \( [a^u\{\text{cat which caught a}^u \text{ fish}\}] \) and \( [a^u \text{ cat which ate}^u \text{ it}] \).

139. \([u^1, u^2] =\text{John}; [u^1, u^2] \text{ cat } [u^1, u^2], \text{ have} [u^1, u^2], \text{ fish} [u^1, u^2], \text{ catch} [u^1, u^2]; [u^1] \text{ cat } [u^1], \text{ have} [u^1, u^2], \text{ eat} [u^1, u^2] \).

140. \( \lambda z. \exists x, \exists y, \exists z, (\text{cat}(z) \land \text{have} (\text{john}, z) \land \text{fish}(y) \land \text{catch}(x, y) \land \text{eat}(z, y)) \).

141. John\(^u\) admires \([a^u\{\text{girl who loves}^u \text{ her}^u\}] \) and \( [a^u \text{ boy who loves}^u \text{ her}] \).

142. \([u^1, u^2] =\text{John}; [u^1, u^2] \text{ girl } [u^1, u^2], \text{ admire} [u^1, u^1]; [u^1] \text{ boy } [u^1], \text{ admire} [u^1, u^2], \text{ love} [u^2, u^1] \).

143. \( \lambda x. \exists y, \exists z, (\text{girl}(x) \land \text{admire}(\text{john}, x) \land \text{boy}(y) \land \text{admire}(\text{john}, y) \land \text{love}(y, x)) \).

Given that CDRT+GQ interprets all generalized quantifiers as conditions / tests, the anaphoric connections in the structurally identical examples in (144), (145) and (146) below are correctly predicted to be felicitous.

144. \( [A^u] \text{ cat } [\{v\text{-}caught \, no^u \text{ fish}\} \text{ and } \{v\text{-}ate \, it^u \}] \).

145. John\(^u\) has \( [o^u\{\text{cat which caught no}^u \text{ fish}\}] \) and \( [o^u \text{ cat which ate}^u \text{ it}] \).

146. John\(^u\) admires \([o^u\text{girl who loves}^u] \) and \( [o^u \text{ boy who loves}^u \text{ her}] \).

9.6. DP-Conjunction Donkey Sentences

147. \([\text{Every}^{\text{str}}, u^1] \text{ boy who has a}^u \text{ dog} \) and \( [\text{every}^{\text{str}}, u^2] \text{ girl who has a}^u \text{ cat}] \) must feed it^u \( ^{18} \).

148. every\(^{\text{str}}, u^1 \text{ boy who has a}^u \text{ dog} \) —
\[ \lambda P_u. [\text{every}^{\text{str}}, u (u^1] \text{ boy } [u^1], \text{ dog } [u^1], \text{ have } [u^1, u^2], \text{ P} (u^1)) \]

149. every\(^{\text{str}}, u^2 \text{ girl who has a}^u \text{ cat} \) —
\[ \lambda P_u. [\text{every}^{\text{str}}, u (u^2] \text{ girl } [u^2], \text{ cat } [u^2], \text{ have } [u^2, u^1], \text{ P} (u^2)) \]

150. every\(^{\text{str}}, u^1 \text{ boy who has a}^u \text{ dog and every}^{\text{str}}, u^2 \text{ girl who has a}^u \text{ cat} \) —
\[ \lambda P_u. [\text{every}^{\text{str}}, u (u^1] \text{ boy } [u^1], \text{ dog } [u^1], \text{ have } [u^1, u^2], \text{ P} (u^1)), \]
\[ \text{every}^{\text{str}}, u (u^2] \text{ girl } [u^2], \text{ cat } [u^2], \text{ have } [u^2, u^1], \text{ P} (u^2)) \]

151. must feed \( \text{it}^u \) — \( \lambda x. [\text{must} \text{ feed} (x, u^2)] \)

152. every\(^{\text{str}}, u^1 \text{ boy who has a}^u \text{ dog and every}^{\text{str}}, u^2 \text{ girl who has a}^u \text{ cat} \) must feed \( \text{it}^u \) —
\[ \text{every}^{\text{str}}, u (u^1] \text{ boy } [u^1], \text{ dog } [u^1], \text{ have } [u^1, u^2], \text{ must } \text{ feed} [u^1, u^2], \text{ P} (u^1)), \]
\[ \text{every}^{\text{str}}, u (u^2] \text{ girl } [u^2], \text{ cat } [u^2], \text{ have } [u^2, u^1], \text{ must } \text{ feed} [u^1, u^2], \text{ P} (u^2)) \]

153. \( \lambda x. \exists y, \exists z, (\text{boy}(x) \land \text{dog}(y) \land \text{have}(x, y) \land \text{must} \text{ feed}(x, y) \land \text{cat}(z) \land \text{have} (\text{john}, z) \land \text{cat}(z, y) \land \text{must} \text{ feed}(x, y)) \)

\( ^{18} \) In contrast, the corresponding translation in Chierchia (1995): 96, (76b) delivers the weak truth-conditions (i.e., the donkey indefinites are assigned the weak readings), which are arguably incorrect for the most salient reading of this type of example.

In all fairness, it should be noted that Chierchia (1995): 96 aims to interpret the slightly different example: Every boy that has a dog and every girl that has a cat will beat it (see Chierchia (1995): 96, (76a)). Therefore, his implicit claim might be that this particular example is preferably interpreted by accommodating an 'anger management' kind of scenario wherein the children are advised to beat their pets rather than each other – which would favor the weak reading of the sentence.
Structurally similar sentences like (154) are infelicitous (as Chierchia 1995: 96 observes).

154. Every boy who has a dog and a girl must feed it.

155. Every boy, where boy and dog/ have[u,w], must_feed[u,u].

Following Chierchia (1995): 96, we explain their infelicity should be explained just as the infelicity of examples (144), (145) and (146) above:

- given that generalized quantifiers are conditions / tests in CDRT+GQ, the anaphoric connection between the pronoun it and the indefinite a dog cannot be successfully established in the second conjunct of the translation in (155).
- Alternatively, the infelicity of sentences like (154) above can be attributed to the fact that they fail to establish a discourse-level parallelism between the two DP-conjuncts relative to the anaphor in the VP.
- besides accounting for the infelicity of (154), this hypothesis provides an explanation for the particular indexing exhibited by the felicitous example in (147): the indefinites a dog and a cat receive the same index as a consequence of the fact that the two DP-conjuncts (or the two resulting DRSs) are related by a Parallel rhetorical relation.

10. Limitations of CDRT+GQ: Mixed Weak & Strong Donkey Sentences

156. Every person who buys a book on amazon.com and has a credit card uses it to pay for it.

157. Every man who wants to impress a woman and who has an Arabian horse teaches her to ride it.

The problem with the weak and strong CDRT+GQ meanings for determiners is that they do not distinguish between the indefinites in the restrictor: all of them receive either a weak or a strong reading.

Possible solution: we can make generalized determiners even more ambiguous, i.e., redefine them as determiners binding a sequence of drefs and specifying for each dref different from the 'primary' one whether it receives a weak or a strong reading.

- for example, a determiner of the form det weak,strong((D,D')) quantifies over three drefs u, u' and u'', the 'primary' dref is u and the drefs u' and u'' are introduced by donkey indefinites in the restrictor of the quantification and are weak and strong respectively.

Such determiners can be defined by combining the weak and strong determiner meanings that we have introduced in CDRT+GQ. But, just as in the case of DPL+GQ, the CDRT+GQ meaning for generalized determiners has to further specify the relative scope of the donkey indefinites. For example:

158. det weak,strong((D,D')) := \lambda (u,w) \cdot \text{DET}(u[D],w,\lambda([D \rightarrow (D, D')])

CDRT+GQ faces the same basic kind of problems with respect to conditionals that exhibit asymmetric readings, i.e., weak / strong ambiguities.

Kadmon's generalization – a multi-case conditional with two indefinites in the antecedent generally allows for three interpretations:

- one where the QAdverb (which is a covert always or usually in the case of bare conditionals) quantifies over pairs
- one where it quantifies over instances of the first indefinite
- one where it quantifies over instances of the second indefinite

159. If a village is inhabited by a painter, it is usually pretty. (Kadmon 1987)

160. If a drummer lives in an apartment complex, it is usually half empty. (Bäuerle & Egli 1985, apud Heim 1990: 151, (29))

161. If a woman owns a cat, she usually talks to it. (Heim 1990: 175, (91))

Various factors influence what is the most salient reading of a donkey conditional:

- Bäuerle & Egli (1985) notice that it depends on which indefinites from the antecedent are anaphorically picked up in the consequent
- Rooth (1985) and Kadmon (1987) (see also Heim 1990 and Chierchia 1995 among others) observe that the focus-background structure of the sentence also determines which indefinites receive which reading, the generalization being that the non-focused indefinite in the antecedent is the one that is bound by the QAdverb quantification.

Even more complex conditionals can occur – the most salient reading of (162) is one in which we quantify over most woman-man pairs that have some son or other (i.e., the indefinite a son receives a weak reading).

162. If a woman has a son with a man, she usually keeps in touch with him. (I. Heim, apud Chierchia 1995: 67, (14b))