Handout 9: Introduction to CDRT+GQ

Semantics C (Spring 2010)

1. Dynamic Ty2

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

> "[…] combin[ing] 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."
> (Muskens 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 may hope that it is adequate for a combined theory: a compositional theory of the computational aspects of natural language meaning."(Muskens 1991: 3-4[1])

[1] Page references are to the online version.

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 ($\boldsymbol{D}_t$) is {T, F}
- type e is the type of individuals; for now, we assume that $\boldsymbol{D}_e$ contains only atomic entities, i.e., there are no pluralities
- the domain of type s ($\boldsymbol{D}_s$) 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 $\lambda g.\ g(x)$, which is the (set-theoretic) x^{th} 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

1.2. Definitions and Abbreviations

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

- these are the types of functions from 'assignments' (type s) to static objects of *arbitrary* type
- more than sufficient for our current purposes – but see Stone & Hardt (1999) for an account of strict/sloppy readings that employs 'dynamic' drefs, i.e., drefs of type $s(s(\ldots))$ (these are just the *pointers* introduced in Janssen 1986)
- we restrict our drefs to functions from 'variable assignments' to *static* objects of arbitrary types because, if we allow for arbitrary dref types, e.g., $s(st)$, we might run into counterparts of Russell's paradox – see Muskens (1995): 179-180, fn. 10.

1. **Dynamic Ty2 –** the set of dref types **DRefTyp** and the set of types **Typ**.
   a. The set of basic static types **BasSTyp**: $\{t, e\}$ (truth-values and individuals).
   b. The set of static types **STyp**: the smallest set including **BasSTyp** and s.t., if $\sigma,\tau\in\mathbf{STyp}$, then $(\sigma\tau)\in\mathbf{STyp}$.
   c. The set of dref types **DRefTyp**: the smallest set s.t., if $\sigma\in\mathbf{STyp}$, then $(s\sigma)\in\mathbf{DRefTyp}$.
   d. The set of basic types **BasTyp**: $\mathbf{BasSTyp}\cup\{s\}$ ('variable assignments').
   e. The set of types **Typ**: the smallest set including **BasTyp** and s.t., if $\sigma,\tau\in\mathbf{Typ}$, then $(\sigma\tau)\in\mathbf{Typ}$.

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 $\tau\in\mathbf{Typ}$, there is a denumerable set of τ-constants $\mathbf{Con}_\tau$ and a denumerably infinite set of τ-variables $\mathbf{Var}_\tau = \{v_{\tau,0}, v_{\tau,1}, \ldots\}$, e.g.
   $\mathbf{Con}_e = \{john, mary, dobby, \ldots, a, a', \ldots, b, b', \ldots, a_0, a_1, a_2, \ldots\}$
   $\mathbf{Con}_{et} = \{donkey, farmer, h.elf, witch, \ldots, leave, drunk, walk, \ldots\}$
   $\mathbf{Con}_{e(et)} = \{in\_love, own, beat, have, \ldots\}$
   $\mathbf{Con}_s = \{h, h', \ldots, i, i', \ldots, j, j', \ldots, k, k', \ldots, h_0, h_1, \ldots, i_0, i_1, \ldots\}$
   $\mathbf{Con}_{se} = \{u, u', u'', \ldots, u_0, u_1, u_2, \ldots\}$
   Notational conventions:
   $x, x', \ldots, y, y', \ldots, z, z', \ldots, x_0, x_1, \ldots$ are variables of type e;
   $h, h', h'', \ldots, i, i', i'', \ldots, j, j', j'', \ldots$ are variables of type s;
   $f, f', f'', \ldots f_0, f_1, f_2, \ldots$ are variables over terms of type τ, for any $\tau\in\mathbf{STyp}$;
   $v, v', v'', \ldots, v_0, v_1, v_2, \ldots$ are variables over terms of type τ, for any $\tau\in\mathbf{Typ}$.

The definition in (3) introduces the term $i[\delta]j$ of type t that models the DPL notion of random variable assignment:

- the formula $i[\delta]j$ requires the assignments i and j to differ at most with respect to the value they assign to def δ
- unlike Muskens (1995b, 1996), I introduce this as a basic formula of the language and not as an abbreviation, because the set **DRefTyp** of dref types is infinite and the abbreviation would require an infinite conjunction of formulas (as indicated in (4d) below).

Proper names with capitals, e.g., *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., *john*.

3. **Dynamic Ty2 – terms.**
   For any type $\tau\in\mathbf{Typ}$, the set of τ-terms $\mathbf{Term}_\tau$ is the smallest set s.t.:
   $\mathbf{Con}_\tau\cup\mathbf{Var}_\tau \subseteq \mathbf{Term}_\tau$;
   $\alpha(\beta)\in\mathbf{Term}_\tau$ if $\alpha\in\mathbf{Term}_{\sigma\tau}$ and $\beta\in\mathbf{Term}_\sigma$ for any $\sigma\in\mathbf{Typ}$;
   $(\lambda v.\ \alpha)\in\mathbf{Term}_\tau$ if $\tau=(\sigma\rho)$, $v\in\mathbf{Var}_\sigma$ and $\alpha\in\mathbf{Term}_\rho$ for any $\sigma,\rho\in\mathbf{Typ}$;
   $(\alpha=\beta)\in\mathbf{Term}_\tau$ if $\tau=t$ and $\alpha,\beta\in\mathbf{Term}_\sigma$ for any $\sigma\in\mathbf{Typ}$;
   $(i[\delta]j)\in\mathbf{Term}_\tau$ if $\tau=t$ and $i,i'\in\mathbf{Var}_s$ and $\delta\in\mathbf{Term}_\sigma$, for any $\sigma\in\mathbf{DRefTyp}$.
   Abbreviations (the subscripts on terms indicate their type):
   $John_{se} := \lambda i_s.\ john_e$, $Mary_{se} := \lambda i_s.\ mary_e \ldots$;
   $\mathbf{T} := \lambda f_t.\ f = \lambda f_t.\ f$; $\quad\mathbf{F} := \lambda f_t.\ f_t = \lambda f_t.\ \mathbf{T}$;
   $\neg := \lambda f_t.\ f_t = \mathbf{F}$; $\quad\wedge := \lambda f_t f'_t.\ (\lambda f''_{tt}.\ f''(f)=f') = \lambda f''_{tt}.\ f''(\mathbf{T}))$;[2]
   $\rightarrow := \lambda f_t f'_t.\ (f \wedge f') = f$; $\quad\vee := \lambda f_t f'_t.\ \neg f \rightarrow f'$;
   $\forall v(\phi) := \lambda v.\ \phi = \lambda v.\ \mathbf{T}$; $\quad\exists v(\phi) := \lambda v.\ \neg\phi \neq \lambda v.\ \mathbf{T}$.

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

Axiom1 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., $u, u', \ldots, u_0, u_1 \ldots$

The constant function *John* (recall that $John := \lambda i.\ john_e$), 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.

Axiom2 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.

Axiom3 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 δ. If they assign the same values to all drefs, the 'assignments' are identical.

Axiom4 ensures that we have enough 'assignments': for any given 'assignment' i, any unspecific dref v and any possible dref value (i.e., static object) f of the appropriate type, there is an 'assignment' j that differs from i at most with respect to the value it assigns to v and which in fact assigns f as the value of v.

4. **Dynamic Ty2 – frames, models, assignments, interpretation and truth.**
   a. A *standard frame* $\boldsymbol{F}$ for Dynamic Ty2 is a set $\{\boldsymbol{D}_\tau: \tau\in\mathbf{Typ}\}$ s.t. $\boldsymbol{D}_e$, $\boldsymbol{D}_t$ and $\boldsymbol{D}_s$ are pairwise disjoint sets and $\boldsymbol{D}_{\sigma\tau} = \{f: f$ is a total function from $\boldsymbol{D}_\sigma$ to $\boldsymbol{D}_\tau\}$, for any $\sigma,\tau\in\mathbf{Typ}$.
   b. A *model* $\boldsymbol{M}$ for Dynamic Ty2 is a pair $<\boldsymbol{F}^{\boldsymbol{M}}, \|\cdot\|^{\boldsymbol{M}}>$ s.t.:
   - $\boldsymbol{F}^{\boldsymbol{M}}$ is a standard frame for Dynamic Ty2;
   - $\|\cdot\|^{\boldsymbol{M}}$ assigns an object $\|\alpha\|^{\boldsymbol{M}}\in\boldsymbol{D}^{\boldsymbol{M}}_\tau$ to each $\alpha\in\mathbf{Con}_\tau$ for any $\tau\in\mathbf{Typ}$, i.e., $\|\cdot\|^{\boldsymbol{M}}$ respects typing;
   - $\boldsymbol{M}$ satisfies the following axioms:
   Axiom1 ("Unspecific drefs"): $\mathbf{udref}(\delta)$,
   for any unspecific dref name δ of any type $(s\tau)\in\mathbf{DRefTyp}$,
   e.g., $u_0, u_1, \ldots$ but not *John*, *Mary*, …;
   udref is a non-logical constant[3] intuitively identifying the 'variable' drefs,

[2] Equivalently, $\wedge := \lambda f_t f'_t.\ \forall f''_{t(tt)}(f''(f, f') = f''(\mathbf{T}, \mathbf{T}))$ or $\wedge := \lambda f_t f'_t.\ \forall f''_{tt}(f = (f''(f) = f''(f')))$.

[3] In fact, **udref** stands for an infinite family of non-logical constants of type (τt) for any $\tau\in\mathbf{DRefTyp}$. 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 seqq.

i.e., the non-constant functions of type $s\tau$ (for any $\tau \in \mathbf{STyp}$) intended to model DPL-like variables.

Axiom2 ("Drefs have unique dref names"): $\mathbf{udref}(\delta) \wedge \mathbf{udref}(\delta') \rightarrow \delta \neq \delta'$,
for any two distinct dref names δ and δ' of type τ,
for any type $\tau \in \mathbf{DRefTyp}$,
i.e., we make sure that we do not accidentally update a dref δ' when we update δ.

Axiom3 ("Identity of 'assignments'"): $\forall i_s j_s (i[]j \rightarrow i=j)$.

Axiom4 ("Enough 'assignments'"): $\forall i_s \forall v_{s\tau} \forall f_\tau (\mathbf{udref}(v) \rightarrow \exists j_s (i[v]j \wedge vj=f))$,
for any type $\tau \in \mathbf{STyp}$.

c. An $\boldsymbol{M}$-*assignment* θ is a function that assigns to each variable $v \in \mathbf{Var}_\tau$ an element $\theta(v) \in \boldsymbol{D}_\tau^{\boldsymbol{M}}$ for any $\tau \in \mathbf{Typ}$. Given an $\boldsymbol{M}$-assignment θ, if $v \in \mathbf{Var}_\tau$ and $d \in \boldsymbol{D}_\tau^{\boldsymbol{M}}$, then $\theta^{v/d}$ is the $\boldsymbol{M}$-assignment identical with θ except that it assigns d to v.

d. The interpretation function $\|\cdot\|^{\boldsymbol{M},\theta}$ is defined as follows:

$\|\alpha\|^{\boldsymbol{M},\theta} = \|\alpha\|^{\boldsymbol{M}}$ if $\alpha \in \mathbf{Con}_\tau$ for any $\tau \in \mathbf{Typ}$;
$\|\alpha\|^{\boldsymbol{M},\theta} = \theta(\alpha)$ if $\alpha \in \mathbf{Var}_\tau$ for any $\tau \in \mathbf{Typ}$;
$\|\alpha(\beta)\|^{\boldsymbol{M},\theta} = \|\alpha\|^{\boldsymbol{M},\theta}(\|\beta\|^{\boldsymbol{M},\theta})$;
$\|\lambda v.\ \alpha\|^{\boldsymbol{M},\theta} = \langle \|\alpha\|^{\boldsymbol{M},\theta^{v/d}} : d \in \boldsymbol{D}^{\boldsymbol{M}}_\sigma \rangle$ if $v \in \mathbf{Var}_\sigma$;
$\|\alpha=\beta\|^{\boldsymbol{M},\theta} = \mathrm{T}$ if $\|\alpha\|^{\boldsymbol{M},\theta} = \|\beta\|^{\boldsymbol{M},\theta}$
$= \mathrm{F}$ otherwise.
$\|i[\delta]j\|^{\boldsymbol{M},\theta} = \mathrm{T}$ if $\delta \in \mathbf{Term}_\sigma$, $\sigma \in \mathbf{DRefTyp}$ and
$\|\forall v_\sigma(\mathbf{udref}(v) \wedge v \neq \delta \rightarrow vi=vj)\|^{\boldsymbol{M},\theta} = \mathbf{T}$ and
$\|\forall v_\tau(\mathbf{udref}(v) \rightarrow vi=vj\|^{\boldsymbol{M},\theta} = \mathbf{T}$ for all $\tau \neq \sigma$, $\tau \in \mathbf{DRefTyp}$
$= \mathrm{F}$ otherwise.

e. Truth: A formula $\phi \in \mathbf{Term}_t$ is *true* in $\boldsymbol{M}$ relative to θ iff $\|\phi\|^{\boldsymbol{M},\theta} = \mathrm{T}$.
A formula $\phi \in \mathbf{Term}_t$ is *true* in $\boldsymbol{M}$ iff it is true in $\boldsymbol{M}$ relative to any assignment θ.

Given the domain $\boldsymbol{D}_e$ and the set of **udref** names, how can we construct the domain $\boldsymbol{D}_s$ so that $\boldsymbol{D}_s$ 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.

- '$\wedge$' is the Dynamic Ty2 conjunction, i.e., the official, type-logical conjunction, and '$\neg$' is the Dynamic Ty2 negation, i.e., the official, type-logical negation. In contrast, dynamic conjunction ';' and dynamic negation '~' are simply abbreviations.
- yhe DPL notion of random assignment $[x]$ has as its direct correspondent the random assignment $[u]$ of Dynamic Ty2.

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

- conditions are terms of type st, i.e., they denote sets of 'assignments'; intuitively, conditions denote the set of 'assignments' that satisfy them.
- DRSs are terms of type $s(st)$, 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 st:
$R\{u_1, \ldots, u_n\} := \lambda i_s.\ R(u_1 i, \ldots, u_n i)$,
for any non-logical constant R of type $e^n t$,
where $e^n t$ is defined as follows: $e^0 t := t$ and $e^{m+1} t := e(e^m t)$
$u_1=u_2 := \lambda i_s.\ u_1 i = u_2 i$

b. Atomic DRSs (DRSs containing exactly one atomic condition) – type $s(st)$ (corresponding to DPL atomic formulas):
$[R\{u_1, \ldots, u_n\}] := \lambda i_s j_s.\ i=j \wedge R\{u_1, \ldots, u_n\}j$
$[u_1=u_2] := \lambda i_s j_s.\ i=j \wedge u_1 j = u_2 j$

c. Condition-level connectives (negation), i.e., non-atomic conditions:
$\sim D := \lambda i_s.\ \neg\exists k_s(Dik)$ [4],
where D is a DRS (term of type $s(st)$)
i.e., $\sim D := \lambda i_s.\ i \notin \mathbf{Dom}(D)$,
where $\mathbf{Dom}(D) := \{i_s: \exists j_s(Dij)\}$

d. Tests (generalizing 'atomic' DRSs):
$[C_1, \ldots, C_m] := \lambda i_s j_s.\ i=j \wedge C_1 j \wedge \ldots \wedge C_m j$ [5],
where $C_1, \ldots, C_m$ are conditions (atomic or not) of type st.

e. DRS-level connectives (dynamic conjunction):
$D_1; D_2 := \lambda i_s j_s.\ \exists h_s(D_1 ih \wedge D_2 hj)$,
where D_1 and D_2 are DRSs (type $s(st)$)

f. Quantifiers (random assignment of value to a dref):
$[u] := \lambda i_s j_s.\ i[u]j$

g. Truth:
A DRS D (type $s(st)$) is true with respect to an input info state i_s iff $\exists j_s(Dij)$,
i.e., $i \in \mathbf{Dom}(D)$ (equivalently, $i \in !D$).

[4] Strictly speaking, the Dynamic Ty2 translation of DPL negation is defined as $\mathbf{TR}(\sim\phi) := [\sim\mathbf{TR}(\phi)]$, i.e., $\mathbf{TR}(\sim\phi) := [\lambda i_s.\ \neg\exists k_s(\mathbf{TR}(\phi)ik)]$. **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 $\mathbf{TR}(R(x_1, \ldots, x_n)) := [R\{u_1, \ldots, u_n\}]$ and $\mathbf{TR}(x_1=x_2) := u_1=u_2$.

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

$[C_1, \ldots, C_m] := \lambda i_s j_s.\ ([C_1]; \ldots; [C_m])ij$, where $[C] := \lambda i_s j_s.\ i=j \wedge Cj$.

The abbreviations introduced in definition (6) below correspond to the DPL abbreviations defined in the previous chapter.

- '$\exists$' and '$\forall$' are the official type-logical existential and universal quantifiers, while '$\forall$' and '$\exists$' are the abbreviations corresponding to the dynamic (DPL-style) existential and universal quantifiers.
- following the notational conventions in the literature, I use '$\rightarrow$' and '$\vee$' 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):**

$!D := \sim[\sim D]$ [6],

i.e., $!D := \lambda i_s.\ \exists k_s(Dik)$ or simply: $!D := \mathbf{Dom}(D)$

$D_1 \vee D_2 := \sim([\sim D_1]; [\sim D_2])$,

i.e., $D_1 \vee D_2 := \sim[\sim D_1, \sim D_2]$

i.e., $D_1 \vee D_2 := \lambda i_s.\ \exists k_s(D_1ik \vee D_2ik)$;

equivalently: $D_1 \vee D_2 := \mathbf{Dom}(D_1) \cup \mathbf{Dom}(D_2)$ [7]

$D_1 \rightarrow D_2 := \sim(D_1; [\sim D_2])$,

i.e., $D_1 \rightarrow D_2 := \lambda i_s.\ \forall h_s(D_1ih \rightarrow \exists k_s(D_2hk))$,[8]

i.e., $D_1 \rightarrow D_2 := \lambda i_s.\ (D_1)i \subseteq \mathbf{Dom}(D_2)$,

where $(D)i := \{j_s: Dij\}$

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

$[u_1, \ldots, u_n] := [u_1]; \ldots; [u_n]$

$\exists u(D) := [u]; D$

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

$\forall u(D) := \sim([u]; [\sim D])$,

i.e., $\sim[u \mid \sim D]$ or $[u] \rightarrow D$ or equivalently $\sim\exists u([\sim D])$,

i.e., $\forall u(D) := \lambda i_s.\ \forall h_s(i[u]h \rightarrow \exists k_s(Dhk))$,

i.e., $\forall u(D) := \lambda i_s.\ ([u])i \subseteq \mathbf{Dom}(D)$

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

$[u_1, \ldots, u_n \mid C_1, \ldots, C_m] := \lambda i_s j_s.\ ([u_1, \ldots, u_n]; [C_1, \ldots, C_m])ij$,

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

i.e., $[u_1, \ldots, u_n \mid C_1, \ldots, C_m] := \lambda i_s j_s.\ i[u_1, \ldots, u_n]j \wedge C_1j \wedge \ldots \wedge C_mj$.

[6] Strictly speaking, DPL anaphoric closure is translated in Dynamic Ty2 as $\mathbf{TR}(!\phi) := [\sim\mathbf{TR}(\sim\phi)]$, i.e., $\mathbf{TR}(!\phi) := [\sim[\sim\mathbf{TR}(\phi)]] = [\sim[\lambda j_s.\ \neg\exists l_s(\mathbf{TR}(\phi)jl)]] = [\lambda i_s.\ \neg\exists k_s([\lambda j_s.\ \neg\exists l_s(\mathbf{TR}(\phi)jl)]ik)]$, i.e., $\mathbf{TR}(!\phi) := [\lambda i_s.\ \neg\exists k_s(i{=}k \wedge \neg\exists l_s(\mathbf{TR}(\phi)kl))] = [\lambda i_s.\ \neg(\neg\exists l_s(\mathbf{TR}(\phi)il))]$, i.e., $\mathbf{TR}(!\phi) := [\lambda i_s.\ \exists l_s(\mathbf{TR}(\phi)il)] = \mathbf{Dom}(\mathbf{TR}(\phi))$.

[7] $D_1 \vee D_2 := \sim([\sim D_1]; [\sim D_2]) = \lambda i.\ \neg\exists k(([\sim D_1]; [\sim D_2])ik) = \lambda i.\ \neg\exists kl([\sim D_1]il \wedge [\sim D_2]lk) = \lambda i.\ \neg\exists kl(i{=}l \wedge \neg\exists h(D_1ih) \wedge l{=}k \wedge \neg\exists h'(D_2lh')) = \lambda i.\ \neg(\neg\exists h(D_1ih) \wedge \neg\exists h'(D_2ih')) = \lambda i.\ \exists h(D_1ih) \vee \exists h'(D_2ih') = \lambda i.\ \exists k(D_1ik \vee D_2ik)$.

[8] $D_1 \rightarrow D_2 := \sim(D_1; [\sim D_2]) = \lambda i.\ \neg\exists k((D_1; [\sim D_2])ik) = \lambda i.\ \neg\exists kl(D_1il \wedge [\sim D_2]lk) = \lambda i.\ \neg\exists kl(D_1il \wedge l{=}k \wedge \neg\exists h(D_2lh)) = \lambda i.\ \neg\exists k(D_1ik \wedge \neg\exists h(D_2kh)) = \lambda i.\ \forall k(D_1ik \rightarrow \exists h(D_2kh))$.

2.2. Cross-sentential Anaphora

7. A^{u_1} house-elf fell in love with a^{u_2} witch.
8. He$_{u_1}$ bought her$_{u_2}$ an^{u_3} 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 $s(st)$ 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 (**Axiom4** in (4b) above)

Note that the formula in (12) capturing the truth-conditions of discourse (7-8) contains a vacuous λ-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. $[x, y \mid h.elf(x), witch(y), in\_love(x, y)]$; $[z \mid a.purse(z), buy(x, y, z)]$
10. $[u_1, u_2 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}]$; $[u_3 \mid a.purse\{u_3\}, buy\{u_1, u_2, u_3\}]$
11. $\lambda i_s j_s.\ i[u_1, u_2, u_3]j \wedge h.elf(u_1j) \wedge witch(u_2j) \wedge in\_love(u_1j, u_2j) \wedge a.purse(u_3j) \wedge buy(u_1j, u_2j, u_3j)$
12. $\lambda i_s.\ \exists x_e \exists y_e \exists z_e(h.elf(x) \wedge witch(y) \wedge in\_love(x, y) \wedge a.purse(z) \wedge buy(x, y, z))$

How do we derive the truth conditions in (12) based on the DRS in (11), the definition of truth in (5g) and **Axiom4**?

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 '$\rightarrow$' 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 $s(st)$), out of a condition of type st.

13. Every^{u_1} house-elf who falls in love with a^{u_2} witch buys her$_{u_2}$ an^{u_3} alligator purse.
14. $[x, y \mid h.elf(x), witch(y), in\_love(x, y)] \rightarrow [z \mid a.purse(z), buy(x, y, z)]$
15. $[[u_1, u_2 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}] \rightarrow [u_3 \mid a.purse\{u_3\}, buy\{u_1, u_2, u_3\}]]$

- 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 λ-abstraction over 'assignments', followed by a static first-order formula that captures the intuitively correct truth-conditions for sentence (13)

16. $\lambda i_s j_s.\ i{=}j \wedge \forall h_s(i[u_1, u_2]h \wedge h.elf(u_1h) \wedge witch(u_2h) \wedge in\_love(u_1h, u_2h)$
$\quad\quad \rightarrow \exists k_s(h[u_3]k \wedge a.purse(u_3k) \wedge buy(u_1k, u_2k, u_3k)))$
17. $\lambda i_s.\ \forall x_e \forall y_e(h.elf(x) \wedge witch(y) \wedge in\_love(x, y)$
$\quad\quad \rightarrow \exists z_e(a.purse(z) \wedge buy(x, y, z)))$

2.4. Conditional Donkey Sentences

- the conditional donkey sentence in (18) receives the same Dynamic Ty2 translation and the same truth-conditions as the relative-clause donkey sentence in (13)
- that is, Dynamic Ty2 captures the intuitive correspondence between the generalized determiner *every* and bare conditional structures (just as DPL does).

18. If a^{u_1} house-elf falls in love with a^{u_2} witch, he$_{u_1}$ buys her$_{u_2}$ an^{u_3} alligator purse.

Consider now the negative donkey sentences in (19), (20) and (21).

19. No^{x} house-elf who falls in love with a^{y} witch buys her$_{y}$ an^{z} alligator purse.
20. If a^{x} house-elf falls in love with a^{y} witch, he$_{x}$ never buys her$_{y}$ an^{z} alligator purse.
21. If a^{x} house-elf falls in love with a^{y} witch, he$_{x}$ doesn't buy her$_{y}$ an^{z} alligator purse.

We can translate the determiner *no* in sentence (19) in two different ways:

- by means of a combination of negation and existential quantification
- by means of a combination of negation and universal quantification

These two ways are equivalent given the partial duality in (22) between existential and universal quantification, inherited from DPL by Dynamic Ty2.

22. $\sim\exists u(D; D') = \forall u([D \rightarrow [\sim D']])$[9]

---

[9] $\forall u(D \rightarrow [\sim D'])$

$= \lambda i_s.\ \forall h_s([u]ih \rightarrow \exists k_s((D \rightarrow [\sim D'])hk))$

$= \lambda i_s.\ \forall h_s([u]ih \rightarrow \exists k_s(h{=}k \wedge \forall h'_s(Dhh' \rightarrow \exists k'_s([\sim D']h'k'))) = \lambda i_s.\ \forall h_s([u]ih \rightarrow \forall h'_s(Dhh' \rightarrow \exists k'_s(h'{=}k' \wedge (\sim D')k')))$

$= \lambda i_s.\ \forall h_s([u]ih \rightarrow \forall h'_s(Dhh' \rightarrow (\sim D')h'))$

$= \lambda i_s.\ \forall h_s([u]ih \rightarrow \forall h'_s(Dhh' \rightarrow \neg\exists k_s(Dh'k))$

$= \lambda i_s.\ \forall h_s([u]ih \rightarrow \forall h'_s(Dhh' \rightarrow \neg\exists k_s(Dh'k))$

$= \lambda i_s.\ \neg\exists h_s([u]ih \wedge \neg\forall h'_s(Dhh' \rightarrow \neg\exists k_s(Dh'k))$

$= \lambda i_s.\ \neg\exists h_s([u]ih \wedge \exists h'_s\neg(Dhh' \rightarrow \neg\exists k_s(Dh'k))$

$= \lambda i_s.\ \neg\exists h_s([u]ih \wedge \exists h'_s(Dhh' \wedge \exists k_s(Dh'k))$

$= \lambda i_s.\ \neg\exists h_s\exists h'_s\exists k_s([u]ih \wedge Dhh' \wedge Dh'k)$

$= \lambda i_s.\ \neg\exists k_s(([u]; D; D')ik)$

$= \sim\exists u(D; D')$

- the terms in (22) are of type *st* because both dynamic negation '~' and universal quantification '$\forall$' are condition-level connectives
- the corresponding tests – which are DRSs, 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. $[\sim\exists u_1([h.elf\{u_1\}]; \exists u_2([witch\{u_2\}, in\_love\{u_1, u_2\}]); \exists u_3([a.purse\{u_3\}, buy\{u_1, u_2, u_3\}]))]$
24. $[\sim[u_1, u_2, u_3 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}, a.purse\{u_3\}, buy\{u_1, u_2, u_3\}]]$
25. $[\forall u_1([h.elf\{u_1\}]; \exists u_2([witch\{u_2\}, in\_love\{u_1, u_2\}])$
$\quad\quad \rightarrow [\sim\exists u_3([a.purse\{u_3\}, buy\{u_1, u_2, u_3\}])])]$
26. $[[u_1, u_2 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}] \rightarrow [\sim[u_3 \mid a.purse\{u_3\}, buy\{u_1, u_2, u_3\}]]]$

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. $\lambda i_s.\ \neg\exists x_e \exists y_e \exists z_e(h.elf(x) \wedge witch(y) \wedge in\_love(x, y) \wedge a.purse(z) \wedge buy(x, y, z))$

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

---

[10] I assume that terms that are equivalent to (Dynamic Ty2 translations of DPL) translations of English sentences are also acceptable as translations.

[11] It is easily checked that the following identities hold:

$\forall u([D \rightarrow D']) = [u] \rightarrow [D \rightarrow D'] = ([u]; D) \rightarrow D' = \exists u(D) \rightarrow D'.$

- 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. $Dobby^u$ --> *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. $Dobby^u$ --> $[u \mid u{=}Dobby]$, i.e., $\lambda i_s j_s.\ i[u]j \wedge uj{=}Dobbyj$, i.e., $\lambda i_s j_s.\ i[u]j \wedge uj{=}dobby$

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. … $Dobby^u$ … he_u …

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. … $John_{John}$ … he_{John} …

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

36. Hermione didn't see a^u / any^u house-elf at The Three Broomsticks.
#He$_u$ was on vacation in the Bahamas.
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 infelicitous 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. … $Dobby^u$ … he_{Dobby} …

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

39. Hermione^{u_1} didn't see Dobby^{u_2} at The Three Broomsticks.
He$_{Dobby}$ was on vacation in the Bahamas.
40. $[u_1 \mid u_1{=}Hermione, \sim[u_2 \mid u_2{=}Dobby, see\_at\_TTB\{u_1, u_2\}]]; [on\_vacation\{Dobby\}]$

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_{se}, v'_{se}, $v_{0,se}$, $v_{1,se}$ 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.**

(**PS 1**) Txt → (Txt) CP	(**PS 5**) VP → DP V'	(**PS 9**) V_{di}' → V_{di} DP
(**PS 2**) CP → (CP) IP	(**PS 6**) V' → V_{in}	(**PS 10**) DP → D NP
(**PS 3**) CP → C IP	(**PS 7**) V' → V_{tr} DP	(**PS 11**) NP → N (CP)
(**PS 4**) IP → I VP	(**PS 8**) V' → V_{di}' DP	(**PS 12**) X → X^+ Conj X
(**LI 1**) D → *a^u, $every^u$, $most^u$, few^u, no^u, $some^u$, any^u, $a^{u'}$, $every^{u'}$, …*	(**LI 5**) N → *farmer, house-elf, donkey, …*	(**LI 9**) I → *Ø, doesn't, don't, -ed, -s, didn't, …*
(**LI 2**) DP → *he_u, she_u, it_u, $he_{u'}$, ..., he_{John}, she_{Mary}, …, t_v, $t_{v'}$, …*	(**LI 6**) V_{tr} → *own, beat, …*	(**LI 10**) C → *if, Ø*
(**LI 3**) DP → *$John^u$, $Mary^u$, $John^{u'}$, ...*	(**LI 7**) V_{in} → *sleep, walk, …*	(**LI 11**) Conj → *and, or*
(**LI 4**) DP → *who, whom, which, Ø*	(**LI 8**) V_{di} → *buy, give, …*	

- 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 Θ' *follows by REL* from a tree Θ iff Θ' is the result of replacing some sub-tree of Θ of the form $[_{CP}\ [_{IP}\ X\ [_{DP}\ wh]\ Y]\ ]$, where X and Y are (possibly empty) strings and wh is either *who*, *whom* or *which*, by a tree $[_{CP}\ [_{DP}\ wh]^v\ [_{CP}\ [_{IP}\ X\ t_v\ Y]\ ]\ ]$, where v is a fresh variable index (not occurring in Θ as a superscript).

REL is basically CP adjunction.

42. *Relativization (REL)*: $[_{CP}\ [_{IP}\ X\ [_{DP}\ wh]\ Y\ ]\ ] \rightarrow [_{CP}\ [_{DP}\ wh]^v\ [_{CP}\ [_{IP}\ X\ t_v\ Y\ ]\ ]\ ]$

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

43. $[_{DP} a^u\ \ [_{NP}[_{N}\ girl]\ \ [_{CP}[_{DP}\ who]^v\ \ [_{CP}[_{IP}[_{I}\ \text{-}s]\ \ [_{VP}\ [_{DP}\ every^{u'}\ boy]\ \ [_{V'}\ \ [_{Vtr}\ adore]\ \ t_v\ ]]]]]]]$

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

A tree Θ' *follows by QR* from a tree Θ iff: (a) Θ' is the result of replacing some sub-tree Σ of Θ of the form $[_{IP}\, X\, [_{DP}\, Z]\, Y]$ by a tree $[_{IP}\, [_{DP}\, Z]^v\, [_{IP}\, X\, t_v\, Y]\,]$, where v is a fresh variable index (not occurring in Θ as a superscript); or (b) Θ' is the result of replacing some sub-tree Σ of Θ of the form $[_{VP}\, X\, [_{DP}\, Z]\, Y]$ by a tree $[_{VP}\, [_{DP}\, Z]^v\, [_{VP}\, X\, t_v\, Y]\,]$, where v is a fresh variable index (not occurring in Θ as a superscript).

The conditions on the QR rule are that Z is not a pronoun or a *wh*-word and that $[_{DP}\, Z]$ is not a proper sub-tree of a DP sub-tree $[_{DP}\, W]$ of Σ.[12]

44. *Quantifier Raising* (*QR*):
 a. $[_{IP}\, X\, [_{DP}\, Z]\, Y\,] \rightarrow [_{IP}\, [_{DP}\, Z]^v\, [_{IP}\, X\, t_v\, Y\,]\,]$
 b. $[_{VP}\, X\, [_{DP}\, Z]\, Y\,] \rightarrow [_{VP}\, [_{DP}\, Z]^v\, [_{VP}\, X\, t_v\, Y\,]\,]$

For example, the inverse scope of *every*^{u} *house-elf adores a*$^{u'}$ *witch* can be obtained by QR to IP (of course, it could also be obtained by QR to VP):

45. $[_{IP}\ [_{DP}\ \textit{a}^{u'}\ \textit{witch}]^v\ [_{IP}\ [_{I}\ \textit{-s}]\ [_{VP}\ [_{DP}\ \textit{every}^u\ \textit{house-elf}]\ [_{V'}\ [_{V_{tr}}\ \textit{adore}]\ \ t_v\ ]]]]$

LF is the smallest set of trees that includes SS and is closed under QR; thus, SS⊆LF.

4. Type-driven Translation

- in a Fregean / Montagovian framework, the compositional aspect of the interpretation is largely determined by the types for the 'saturated' expressions, i.e., names and sentences
- abbreviate them as **e** and **t**
- an extensional static logic without pluralities (i.e., classical higher-order logic) is the simplest: **e** is *e* (atomic entities) and **t** is *t* (truth-values)
- the English verb *sleep*, for example, is represented by a term *sleep* of type (**et**), i.e., (*et*), and the generalized quantifier (GQ) *every man* by a term of type ((**et**)**t**), i.e., ((*et*)*t*)

- this setup can be complicated in various ways – for example, Lewis (1972) and Creswell (1973) introduce intensionality by letting **t** := *st*, where *s* is the type of indices of evaluation (however one wants to think of them, e.g., as worlds, <world, time> pairs etc.)
- Dynamic Ty2 complicates this by adding another basic type *s*, whose elements model DPL variable assignments
- a sentence denotes a relation between an input and an output 'assignment', i.e., **t** := (*s*(*st*))
- a name (basically) denotes an individual dref, i.e., **e** := *se* [13]

---

[12] For example, if the DP sub-tree $[_{DP}\, W]$ of Σ contains a relative clause which in its turn contains $[_{DP}\, Z]$, we do not want to QR $[_{DP}\, Z]$ all the way out of the relative clause.

[13] Relativizing the interpretation of names to 'assignments' is not different from the Montagovian interpretation of names (or the Tarskian interpretation of individual constants in first-order logic): just as a name like *John* is assigned the same individual, namely $john_e$, relative to any variable assignment g in a static Montagovian system, we interpret proper names in terms of constant functions of type *se*, e.g., the denotation of *John* is given in terms of the constant function $John_{se}$ that maps each 'assignment' i_s to the individual $john_e$.

- the English verb *sleep* is still translated by a term of type (**et**), but now this means that it takes a dref *u* of type **e** and it relates two info states *i* and *i'* of type *s* if and only if *i*=*i'* and the entity denoted by *u* in info state *i*, i.e., *ui*, has the static *sleep* property of type *et*

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*(*st*)) and **e** := (*se*)

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 of the 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).**

Lexical Item	Translation	Type **e** := *se* **t** := *s*(*st*)
$[\textit{sleep}]_{V_{in}}$	$\rightsquigarrow \lambda v_{\mathbf{e}}.\ [sleep_{et}\{v\}]$	**et**
$[\textit{own}]_{V_{tr}}$	$\rightsquigarrow \lambda Q_{(\mathbf{et})\mathbf{t}}.\lambda v_{\mathbf{e}}.\ Q(\lambda v'_{\mathbf{e}}.\ [own_{e(et)}\{v, v'\}])$	((**et**)**t**)(**et**)
$[\textit{buy}]_{V_{di}}$	$\rightsquigarrow \lambda Q'_{(\mathbf{et})\mathbf{t}}.\lambda Q_{(\mathbf{et})\mathbf{t}}.\lambda v_{\mathbf{e}}.\ Q'(\lambda v'_{\mathbf{e}}.\ Q(\lambda v''_{\mathbf{e}}.\ [buy_{e(e(et))}\{v, v', v''\}]))$	(**ett**)((**ett**)(**et**))
$[\textit{house-elf}]_{N}$	$\rightsquigarrow \lambda v_{\mathbf{e}}.\ [h.elf_{et}\{v\}]$	**et**
$[\textit{he}_u]_{DP}$	$\rightsquigarrow \lambda P_{\mathbf{et}}.\ P(u_{\mathbf{e}})$	(**et**)**t**
$[t_v]_{DP}$	$\rightsquigarrow \lambda P_{\mathbf{et}}.\ P(v_{\mathbf{e}})$	(**et**)**t**
$[\textit{he}_{Dobby}]_{DP}$	$\rightsquigarrow \lambda P_{\mathbf{et}}.\ P(Dobby_{\mathbf{e}})$	(**et**)**t**
$[\textit{Dobby}^u]_{DP}$	$\rightsquigarrow \lambda P_{\mathbf{et}}.\ [u \mid u{=}Dobby];\ P(u)$	(**et**)**t**
$[\textit{a}^u]_{D}$	$\rightsquigarrow \lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [u];\ P'(u);\ P(u)$, i.e., $\lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ \exists u(P'(u);\ P(u))$	(**et**)((**et**)**t**)
$[\textit{every}^u]_{D}$	$\rightsquigarrow \lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [([u];\ P'(u)) \rightarrow P(u)]$, i.e., $\lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [\forall u(P'(u) \rightarrow P(u))]$	(**et**)((**et**)**t**)

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

Lexical Item	Translation	**Type** $\mathbf{e} := se$ $\mathbf{t} := s(st)$
$[no^u]_{\mathrm{D}}$	$\rightsquigarrow \lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [\sim([u];\ P'(u);\ P(u))]$, i.e., $\lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [\sim\exists u(P'(u);\ P(u))]$	$\mathbf{(et)((et)t)}$
	$\rightsquigarrow \lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [([u];\ P'(u)) \rightarrow [\sim P(u)]]$, i.e., $\lambda P'_{\mathbf{et}}.\lambda P_{\mathbf{et}}.\ [\forall u(P'(u) \rightarrow [\sim P(u)])]$	
$[who]_{\mathrm{DP}}$	$\rightsquigarrow \lambda P_{\mathbf{et}}.\ P$	$\mathbf{(et)(et)}$
$[Ø]_{\mathrm{I}}$ / $[\text{-}ed]_{\mathrm{I}}$ / $[\text{-}s]_{\mathrm{I}}$	$\rightsquigarrow \lambda D_{\mathbf{t}}.\ D$	$\mathbf{tt}$
$[doesn't]_{\mathrm{I}}$ / $[didn't]_{\mathrm{I}}$	$\rightsquigarrow \lambda D_{\mathbf{t}}.\ [\sim D]$	$\mathbf{tt}$
$[if]_{\mathrm{C}}$	$\rightsquigarrow \lambda D'_{\mathbf{t}}.\lambda D_{\mathbf{t}}.\ [D' \rightarrow D]$	$\mathbf{t(tt)}$
$[and]_{\mathrm{Conj}}$	$\rightsquigarrow \lambda D'_{\mathbf{t}}.\lambda D_{\mathbf{t}}.\ D';\ D$	$\mathbf{t(tt)}$
$[or]_{\mathrm{Conj}}$	$\rightsquigarrow \lambda D'_{\mathbf{t}}.\lambda D_{\mathbf{t}}.\ [D' \vee D]$	$\mathbf{t(tt)}$

- 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^u witch her$_u$ 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 he_u and traces of the form t_v 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 $\mathbf{e} := se$)
- for traces, this index is a variable (also of type $\mathbf{e} := se$).

- 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 DPL-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 *every* 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 $\rightsquigarrow$ α and A is the only daughter of B, then B $\rightsquigarrow$ α.
48. **TR 2 – Functional Application (FA).**
    If A $\rightsquigarrow$ α and B $\rightsquigarrow$ β and A and B are the only daughters of C, then C $\rightsquigarrow$ $\alpha(\beta)$, provided that this is a well-formed term.
49. **TR 3 – Generalized Sequencing (GSeq) (i.e., Sequencing + Predicate Modification).**
    If A $\rightsquigarrow$ α, B $\rightsquigarrow$ β, A and B are the only daughters of C in that order (i.e., C $\rightarrow$ A B) and α and β are of the same type τ of the form **t** or σ**t** (for any $\sigma\in$**Typ**) then C $\rightsquigarrow$ α; β if τ=**t** or C $\rightsquigarrow$ $\lambda v_\sigma.\ \alpha(v); \beta(v)$, if τ=σ**t**, provided that this is a well-formed term.
50. **TR 4 – Quantifying-In (QIn).**
    If DP^v $\rightsquigarrow$ α, B $\rightsquigarrow$ β and DP^v and B are daughters of C, then C $\rightsquigarrow$ $\alpha(\lambda v.\ \beta)$, provided that this is a well-formed term.
51. **TR 5 – Coordination (Co).**
    If $A_1 \rightsquigarrow \alpha_1$, Conj $\rightsquigarrow$ β, $A_2 \rightsquigarrow \alpha_2$ and A_1, Conj and A_2 are the only daughters of A in that order (i.e., A $\rightarrow$ A_1 Conj A_2), then A $\rightsquigarrow$ $\beta(\alpha_1)(\alpha_2)$, provided this a well-formed term and has the same type as α_1 and α_2.

- 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 Θ *translates as* term α', is formally defined as the smallest relation $\rightsquigarrow$ between trees and Dynamic Ty2 terms that is conform to **TR0**-**TR5** and is closed under type-logical identity, e.g., if tree Θ *translates as* term α and α=β is true, then Θ *translates as* β.

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^{u_1} house-elf hates himself$_{u_1}$.

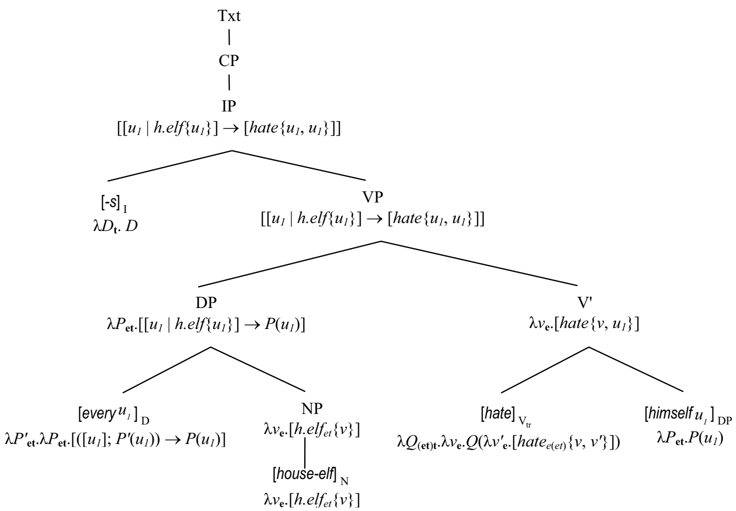
- co-indexation is enough for binding because binding in CDRT (or DPL) is taken care of by the *explicit (and, in this case, unselective) quantification over 'assignments'* built into the meaning of quantifiers
- if we want to obtain bound variable anaphora in a static system, co-indexation, i.e., using the same variable, is not enough: we also need to create a suitable λ-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. $[[u_1 \mid h.elf\{u_1\}] \rightarrow [hate\{u_1, u_1\}]]$
54. $[\forall u_1([h.elf\{u_1\}] \rightarrow [hate\{u_1, u_1\}])]$
55. $\lambda i_s.\ \forall x_e(h.elf(x) \rightarrow hate(x, x))$

CDRT associates the correct Dynamic Ty2 translation with sentence (52) in a compositional way, as shown by the LF in (56) below.

56. Every^{u_1} house-elf hates himself$_{u_1}$.



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:

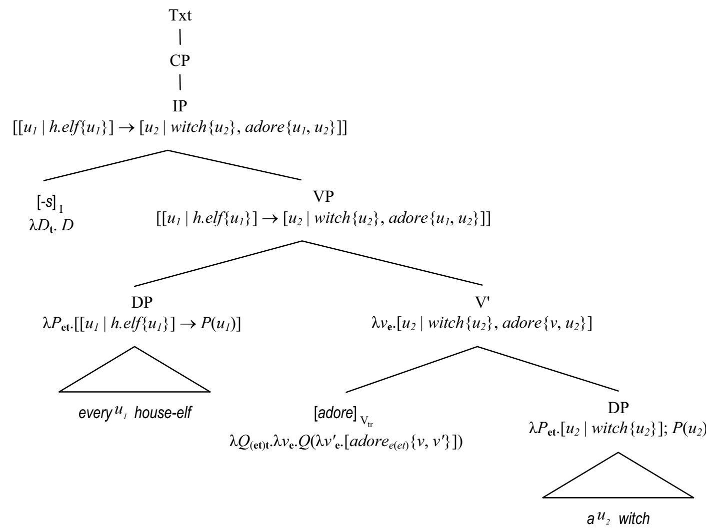
- the surface scope *every*^{u_1}>>*a*^{u_2}
- the inverse scope *a*^{u_2}>>*every*^{u_1}, obtained by an application of QR

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).

57. Every^{u_1} house-elf adores a^{u_2} witch.
58. ***every***^{u_1}**>>*a***^{u_2}: $[[u_1 \mid h.elf\{u_1\}] \rightarrow [u_2 \mid witch\{u_2\}, adore\{u_1, u_2\}]]$
59. ***every***^{u_1}**>>*a***^{u_2}: $\lambda i_s.\ \forall x_e(h.elf(x) \rightarrow \exists y_e(witch(y) \wedge adore(x, y)))$
60. ***a***^{u_2}**>>*every***^{u_1}: $[u_2 \mid witch\{u_2\}, [u_1 \mid h.elf\{u_1\}] \rightarrow [adore\{u_1, u_2\}]]$
61. ***a***^{u_2}**>>*every***^{u_1}: $\lambda i_s.\ \exists y_e(witch(y) \wedge \forall x_e(h.elf(x) \rightarrow adore(x, y)))$

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

62. ***every***^{u_1}**>>*a***^{u_2}: Every^{u_1} house-elf adores a^{u_2} witch.

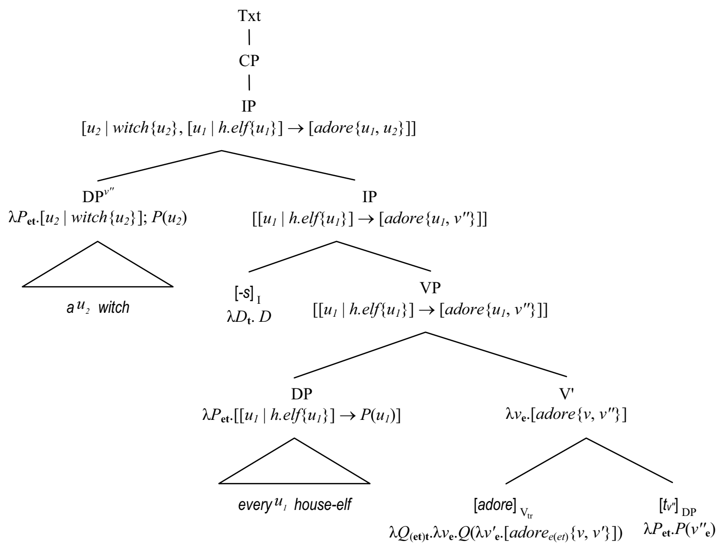


The inverse scope is obtained by applying the QR rule to the indefinite DP *a*^{u_2} *witch*, as shown in (63) below.

The application of the QR rule yields the inverse scope not because it places the indefinite DP in a c-commanding position, *but because it reverses the order of updates*.

Thus, having a syntactic level for quantifier scoping that encodes dominance in addition to linear precedence relations is a bit of an overkill.

63. ($\boldsymbol{a}^{u_2}$ >> $\boldsymbol{every}^{u_1}$) Every^{u_1} house-elf adores a^{u_2} witch.



5.3. Quantifier Scope with Ditransitive Verbs

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

64. Dobby^{u_3} gave every^{u_1} witch her$_{u_1}$ 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),[14] I analyze *her alligator purse* as the DP in (65) below.[15]

65. [$_{DP}$ *a*^{u_2} [$_{NP}$ [$_N$ *alligator purse*] [$_{PP}$ *of her*$_{u_1}$]]]

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

66. [*of*]$_P$ $\rightsquigarrow$ $\lambda Q_{\mathbf{(et)t}}.\ \lambda v_{\mathbf{e}}.\ Q(\lambda v'_{\mathbf{e}}.\ [of_{e(et)}\{v, v'\}])$

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. [$_{PP}$ *of her*$_{u_1}$] $\rightsquigarrow_{\mathbf{FA}}$ $\lambda v_{\mathbf{e}}.\ [of\{v, u_1\}]$
   b. [$_{NP}$ [$_N$ *alligator purse*] [$_{PP}$ *of her*$_{u_1}$]] $\rightsquigarrow_{\mathbf{GSeq}}$ $\lambda v_{\mathbf{e}}.\ [a.purse\{v\}, of\{v, u_1\}]$
   c. (65) $\rightsquigarrow_{\mathbf{FA}}$ $\lambda P_{\mathbf{et}}.\ [u_2 \mid a.purse\{u_2\}, of\{u_2, u_1\}];\ P(u_2)$

The syntactic structure of the V' is provided in linearized form in (68) and is compositionally translated in (69).

The Dative GQ *every*^{u_1} *witch* takes scope over the Accusative GQ and also binds the pronoun *her*$_{u_1}$ contained in it.

68. [$_{V'}$ [$_{Vdi'}$ give every^{u_1} witch] [$_{DP}$ *a*^{u_2} *alligator purse of her*$_{u_1}$]]
69. [$_{Vdi'}$ give every^{u_1} witch] $\rightsquigarrow_{\mathbf{FA}}$ $\lambda Q_{\mathbf{(et)t}}.\ \lambda v_{\mathbf{e}}.\ [[u_1 \mid witch\{u_1\}] \rightarrow Q(\lambda v''_{\mathbf{e}}.\ [give\{v, u_1, v''\}])]$
   (68) $\rightsquigarrow_{\mathbf{FA}}$ $\lambda v_{\mathbf{e}}.\ [[u_1 \mid witch\{u_1\}] \rightarrow [u_2 \mid a.purse\{u_2\}, of\{u_2, u_1\}, give\{v, u_1, u_2\}]]$

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

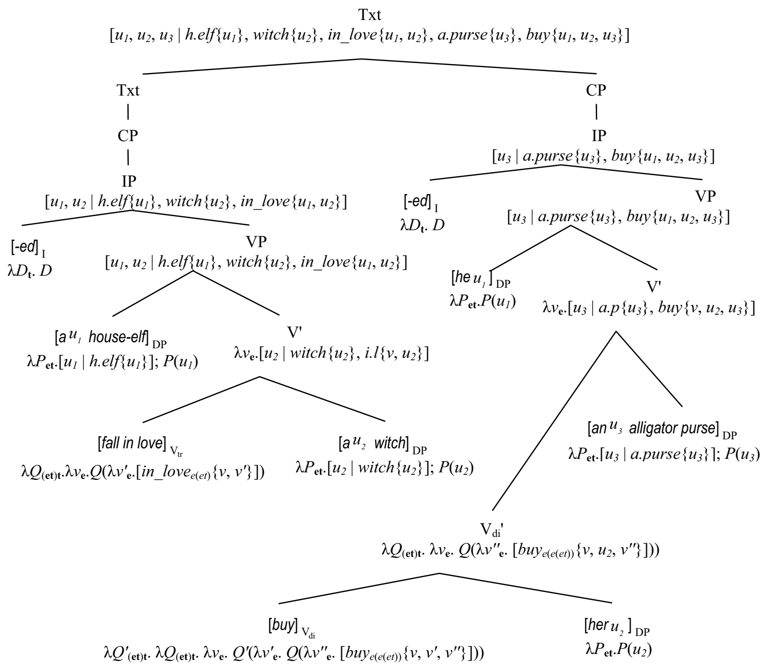
70. $[u_3 \mid u_3{=}Dobby, [u_1 \mid witch\{u_1\}] \rightarrow [u_2 \mid a.purse\{u_2\}, of\{u_2, u_1\}, give\{u_3, u_1, u_2\}]]$
71. $\lambda i_s.\ \exists z_e(z{=}dobby \wedge \forall x_e(witch(x) \rightarrow \exists y_e(a.purse(y) \wedge of(y, x) \wedge give(z, x, y))))$,
   i.e., $\lambda i_s.\ \forall x_e(witch(x) \rightarrow \exists y_e(a.purse(y) \wedge of(y, x) \wedge give(dobby, x, y)))$

[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 *in*definite 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.

[15] I assume that the following phrase structure and lexical insertion rules are added to the syntax of our English fragment: (**PS 13**) NP → N (PP); (**PS 14**) PP → P DP); (**LI 12**) P → *of*.

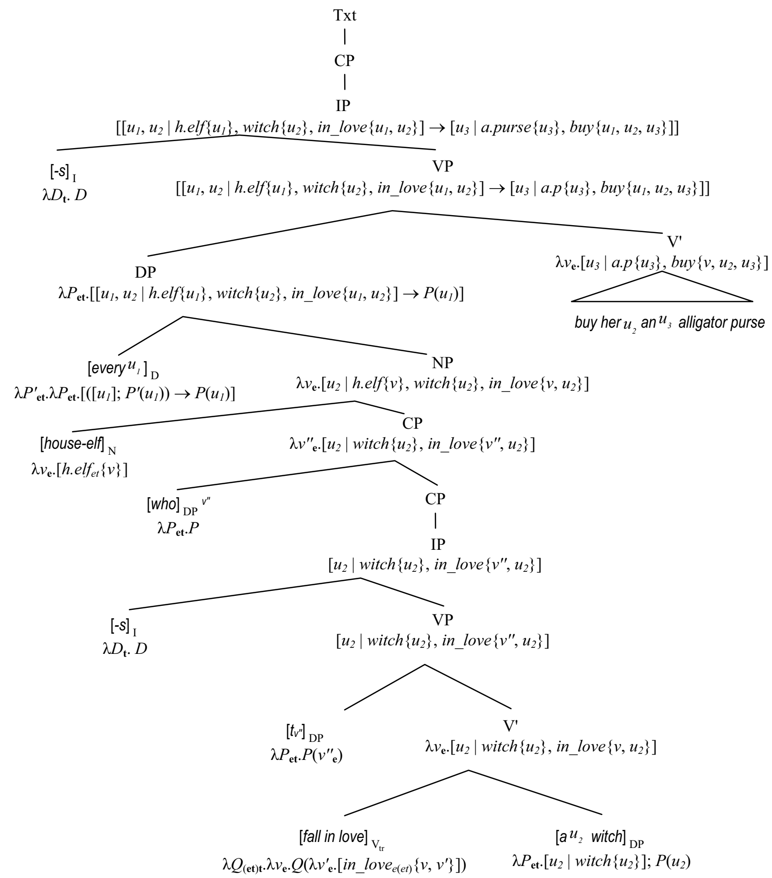
5.4. Cross-sentential Anaphora

72. A^{u_1} house-elf fell in love with a^{u_2} witch.
73. He$_{u_1}$ bought her$_{u_2}$ an^{u_3} alligator purse.



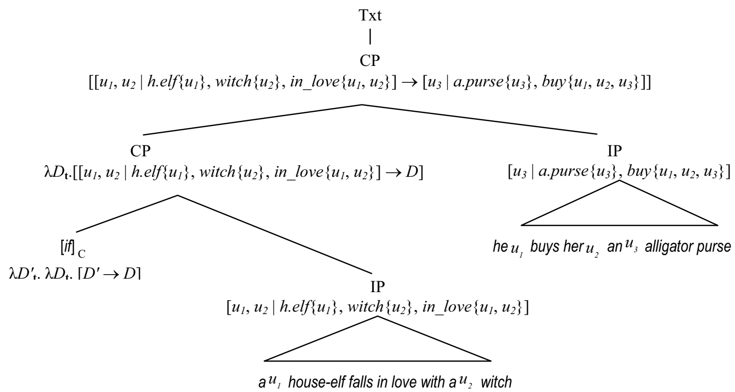
5.5. Relative-clause Donkey Sentences

74. Every^{u_1} house-elf who falls in love with a^{u_2} witch buys her$_{u_2}$ an^{u_3} alligator purse.



5.6. Conditional Donkey Sentences

75. If a^{u_1} house-elf falls in love with a^{u_2} witch, he_{u_1} buys her_{u_2} an^{u_3} alligator purse.



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

76. If a^{u_1} house-elf falls in love with a^{u_2} witch and she_{u_2} likes fancy handbags, he_{u_1} buys her_{u_2} an^{u_3} alligator purse.
77. If a^{u_1} farmer owns a^{u_2} donkey, he_{u_1} beats it_{u_2} or he_{u_1} feeds it_{u_2} poorly.

6. Translating Unselective Quantification into Dynamic Ty2

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

78. $\|\mathbf{det}(\phi, \psi)\| = \{<g, h>: g=h \text{ and } \mathbf{DET}((\phi)^g, \mathbf{Dom}(\|\psi\|))\}$,

where **DET** is the corresponding static determiner

and $(\phi)^g := \{h: \|\phi\|<g, h> = \mathrm{T}\}$

and $\mathbf{Dom}(\|\phi\|) := \{g: \text{there is an } h \text{ s.t. } \|\phi\|<g, h> = \mathrm{T}\}$.

79. $\mathbf{det}_x(\phi, \psi) := \mathbf{det}([x]; \phi, \psi)$

In particular:

80. $\|\mathbf{every}_x(\phi, \psi)\| = \{<g, h>: g=h \text{ and } \mathbf{EVERY}(([x]; \phi)^g, \mathbf{Dom}(\|\psi\|))\}$,
i.e., $\|\mathbf{every}_x(\phi, \psi)\| = \{<g, h>: g=h \text{ and } ([x]; \phi)^g \subseteq \mathbf{Dom}(\|\psi\|)\}$
81. $\|\mathbf{no}_x(\phi, \psi)\| = \{<g, h>: g=h \text{ and } \mathbf{NO}(([x]; \phi)^g, \mathbf{Dom}(\|\psi\|))\}$,
i.e., $\|\mathbf{no}_x(\phi, \psi)\| = \{<g, h>: g=h \text{ and } ([x]; \phi)^g \cap \mathbf{Dom}(\|\psi\|) = \varnothing\}$
82. $\forall x(\phi \rightarrow \psi) \Leftrightarrow \exists x(\phi) \rightarrow \psi \Leftrightarrow ([x]; \phi) \rightarrow \psi \Leftrightarrow \mathbf{every}_x(\phi, \psi)$
83. $\sim\exists x(\phi; \psi) \Leftrightarrow \forall x(\phi \rightarrow \sim\psi) \Leftrightarrow \sim([x]; \phi; \psi) \Leftrightarrow \mathbf{no}_x(\phi, \psi)$

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

84. $\mathbf{det}(D, D') := \lambda i_s.\, \mathbf{DET}(Di, \mathbf{Dom}(D'))$,
where **DET** is the corresponding static determiner,
$Di = \{j_s: Dij\}$ and
$\mathbf{Dom}(D') := \{i_s: \exists j_s(Dij)\}$.
85. $\mathbf{det}_u(D, D') := \mathbf{det}([u]; D, D')$

In particular:

86. $\mathbf{every}_u(D, D') = \lambda i_s.\, \mathbf{EVERY}(([u]; D)i, \mathbf{Dom}(D'))$,
i.e., $\mathbf{every}_u(D, D') = \lambda i_s.\, ([u]; D)i \subseteq \mathbf{Dom}(D')$.
87. $\mathbf{no}_u(D, D') = \lambda i_s.\, \mathbf{no}(([u]; D)i, \mathbf{Dom}(D'))$,
i.e., $\mathbf{no}_u(D, D') = \lambda i_s.\, ([u]; D)i \cap \mathbf{Dom}(D') = \varnothing$.
88. $\forall u(D \rightarrow D') = \exists u(D) \rightarrow D' = ([u]; D) \rightarrow D' = \mathbf{every}_u(D, D')$
89. $\sim\exists u(D; D') = \forall u(D \rightarrow \sim D') = \sim([u]; D; D') = \mathbf{no}_u(D, D')$

6.1. Limitations of Unselectivity: Proportions

90. $\mathbf{most}_u(D, D') = \lambda i_s.\, \mathbf{MOST}(([u]; D)i, \mathbf{Dom}(D'))$,
i.e., $\mathbf{most}_u(D, D') = \lambda i_s.\, |([u]; D)i \cap \mathbf{Dom}(D')| > |([u]; D)i \setminus \mathbf{Dom}(D')|$,
i.e., $\mathbf{most}_u(D, D') = \lambda i_s.\, |([u]; D; [!D'])i| > |([u]; D; [\sim D'])i|$.
91. $Most^{u_1}$ house-elves who fall in love with a^{u_2} witch buy her_{u_2} an^{u_3} alligator purse.
92. $[\mathbf{most}_{u_1}([u_2 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}], [u_3 \mid a.purse\{u_3\}, buy\{u_1, u_2, u_3\}])]$
93. $\lambda i_s.\, |([u_1, u_2 \mid h.elf\{u_1\}, witch\{u_2\}, i.l\{u_1, u_2\}]; [!([u_3 \mid a.p\{u_3\}, buy\{u_1, u_2, u_3\}])])i| >$
$|([u_1, u_2 \mid h.elf\{u_1\}, witch\{u_2\}, i.l\{u_1, u_2\}]; [\sim([u_3 \mid a.p\{u_3\}, buy\{u_1, u_2, u_3\}])])i|$,
i.e., by Axioms **3** and **4** ("Identity of 'assignments'" and "Enough 'assignments'"),
$\lambda i_s.\, |\{<x_e, y_e>: h.elf(x) \wedge witch(y) \wedge i.l(x, y) \wedge \exists z_e(a.p(z) \wedge buy(x, y, z))\}| >$
$|\{<x_e, y_e>: h.elf(x) \wedge witch(y) \wedge i.l(x, y) \wedge \neg\exists z_e(a.p(z) \wedge buy(x, y, z))\}|$

6.2. Limitations of Unselectivity: Weak / Strong Ambiguities

94. $\mathbf{every}_u(D, D') = \lambda i_s.\, \mathbf{EVERY}(([u]; D)i, \mathbf{Dom}(D'))$,
i.e., $\mathbf{every}_u(D, D') = \lambda i_s.\, ([u]; D)i \subseteq \mathbf{Dom}(D')$,
i.e., $\mathbf{every}_u(D, D') = \lambda i_s.\, ([u]; D)i \subseteq ([u]; D)i \cap \mathbf{Dom}(D')$,
i.e., $\mathbf{every}_u(D, D') = \lambda i_s.\, ([u]; D)i \subseteq ([u]; D; !D')i$,
95. $Every^{u_1}$ person who has a^{u_2} dime will put it_{u_2} in the meter.
96. $[\mathbf{every}_{u_1}([u_2 \mid person\{u_1\}, dime\{u_2\}, have\{u_1, u_2\}], [put\_in\_meter\{u_1, u_2\}])]$

97. $\lambda i_s.\ ([u_1, u_2 \mid person\{u_1\}, dime\{u_2\}, have\{u_1, u_2\}])i \subseteq$
$([u_1, u_2 \mid person\{u_1\}, dime\{u_2\}, have\{u_1, u_2\}]; [!([put\_in\_meter\{u_1, u_2\}])])i$,
i.e., by Axioms **3** and **4** ("Identity of 'assignments'" and "Enough 'assignments'"),
$\lambda i_s.\ \{<x_e, y_e>: person(x) \wedge dime(y) \wedge have(x, y)\} \subseteq$
$\{<x_e, y_e>: person(x) \wedge dime(y) \wedge have(x, y) \wedge put\_in\_meter(x, y)\}$, i.e.
$\lambda i_s.\ \forall x_e \forall y_e(person(x) \wedge dime(y) \wedge have(x, y) \rightarrow put\_in\_meter(x, y))$

6.3. Conservativity and Unselective Quantification

Assuming that the static determiner **DET** is conservative, we have:

98. $\mathbf{DET}(Di, \mathbf{Dom}(D'))$ iff $\mathbf{DET}(Di, Di \cap \mathbf{Dom}(D'))$ iff $\mathbf{DET}(Di, (D; !D')i)$

The last formula perspicuously encodes the intuition that a dynamic generalized determiner relates two sets of info states:

- the first is the set of output states compatible with the restrictor, i.e., Di
- the second is the set of output states compatible with the restrictor that, in addition, *can* be further updated by the nuclear scope, i.e., $(D; !D')i$

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:**
$\mathbf{det}(D, D') := \lambda i_s.\ \mathbf{DET}(Di, (D; [!D'])i)$
100. **Unselective generalized quantification with built-in dynamic conservativity:**
$\mathbf{det}_u(D, D') := \lambda i_s.\ \mathbf{DET}(([u]; D)i, ([u]; D; [!D'])i)$

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 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 $\mathbf{det}_u(D, D')$
- u is the 'bound' dref (recall that u is a *constant* of type $\mathbf{e} := se$, 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 $\mathbf{det}_u$ relates two sets of individuals (type *et*) and not two sets of 'assignments' (type *st*), as the unselective determiner **det** does:

- the fact that $\mathbf{det}_u$ 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 $\mathbf{det}^{wk}{}_u(D, D')$ and a *strong* meaning $\mathbf{det}^{str}{}_u(D, D')$
- both meanings are defined in terms of the static determiner **DET** and both of them are conditions, i.e., terms of type *st*

101. $\mathbf{det}^{wk}{}_u(D, D') := \lambda i_s.\ \mathbf{DET}(u[Di],\ u[(D; D')i])$
$\mathbf{det}^{str}{}_u(D, D') := \lambda i_s.\ \mathbf{DET}(u[Di],\ u[([D \rightarrow D'])i])$,
where $Di := \{j_s: Dij\}$ and
$u[Di] := \{u_{se}j_s: ([u]; D)ij\}$ $(= \{x_e: \exists j_s(([u]; D)ij \wedge x{=}uj)\})$,
i.e., $u[Di]$ is the image of the set of 'assignments' $([u]; D)i$ under the function u_{se}.

The difference between the *weak* and the *strong* lexical entry for the selective generalized determiners is localized in the nuclear scope of the quantification:

- the weak, 'existential' reading is obtained by simply sequencing (i.e., conjoining) the restrictor DRS D and the nuclear scope DRS D'
- the strong, 'universal' reading is obtained by means of the universal quantification built into the definition of dynamic implication that relates the restrictor DRS D and the nuclear scope DRS D'

Given Axiom **3** ("Identity of 'assignments'") and Axiom **4** ("Enough 'assignments'"), the weak and strong selective determiners in (101) above can be alternatively defined in terms of generalized quantification over info states – we just need to make judicious use of the anaphoric closure operator '!':

102. $\mathbf{det}^{wk}{}_u(D, D') := \lambda i_s.\ \mathbf{DET}(([u \mid !D])i,\ ([u \mid !(D; D')])i)$
$\mathbf{det}^{str}{}_u(D, D') := \lambda i_s.\ \mathbf{DET}(([u \mid !D])i,\ ([u \mid !([D \rightarrow D'])])i)$,[16]
where $Di := \{j_s: Dij\}$.

7.1. Accounting for Weak / Strong Ambiguities

103. Every^{u_1} farmer who owns a^{u_2} donkey beats it$_{u_2}$.

The weak and strong meanings for *every* are provided in (104) and simplified in (105).

104. $\mathbf{every}^{wk}{}_u(D, D') := \lambda i_s.\ \mathbf{EVERY}(u[Di],\ u[(D; D')i])$
$\mathbf{every}^{str}{}_u(D, D') := \lambda i_s.\ \mathbf{EVERY}(u[Di],\ u[([D \rightarrow D'])i])$
105. $\mathbf{every}^{wk}{}_u(D, D') := \lambda i_s.\ u[Di] \subseteq u[(D; D')i]$
$\mathbf{every}^{str}{}_u(D, D') := \lambda i_s.\ u[Di] \subseteq u[([D \rightarrow D'])i]$

The weak reading of sentence (103):

106. $[\mathbf{every}^{wk}{}_{u_1}([u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\}],\ [beat\{u_1, u_2\}])]$

[16] Given that $!([D \rightarrow D']) = D \rightarrow D'$, the strong determiner can be more simply defined as $\mathbf{det}^{str}{}_u(D, D') := \lambda i_s.\ \mathbf{DET}(([u \mid !D])i,\ ([u \mid D \rightarrow D'])i)$.

107. $\lambda i_s.\ u_1[([u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\}])i] \subseteq$
$u_1[([u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\}, beat\{u_1, u_2\}])i]$, i.e.,
$\lambda i_s.\ \{x_e\!: farmer(x) \wedge \exists y_e(donkey(y) \wedge own(x, y))\} \subseteq$
$\{x_e\!: farmer(x) \wedge \exists z_e(donkey(z) \wedge own(x, z) \wedge beat(x, z))\}$, i.e.
$\lambda i_s.\forall x_e(farmer(x) \wedge \exists y_e(donkey(y) \wedge own(x, y))$
$\rightarrow \exists z_e(donkey(z) \wedge own(x, z) \wedge beat(x, z)))$

The strong reading of sentence (103):

108. $[\mathbf{every}^{str}{}_{u_1}([u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\}],\ [beat\{u_1, u_2\}])]$
109. $\lambda i_s.\ u_1[([u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\}])i] \subseteq$
$u_1[([[u_2 \mid farmer\{u_1\}, donkey\{u_2\}, own\{u_1, u_2\} \rightarrow [beat\{u_1, u_2\}]])i]$, i.e.,
$\lambda i_s.\ \{x_e\!: farmer(x) \wedge \exists y_e(donkey(y) \wedge own(x, y))\} \subseteq$
$\{x_e\!: \forall z_e(farmer(x) \wedge donkey(z) \wedge own(x, z) \rightarrow beat(x, z))\}$, i.e.
$\lambda i_s.\forall x_e(farmer(x) \wedge \exists y_e(donkey(y) \wedge own(x, y))$
$\rightarrow \forall z_e(farmer(x) \wedge donkey(z) \wedge own(x, z) \rightarrow beat(x, z)))$, i.e.
$\lambda i_s.\forall x_e \forall z_e(farmer(x) \wedge donkey(z) \wedge own(x, z) \rightarrow beat(x, z))$

7.2. Solving Proportions

110. $\mathbf{most}^{wk}{}_u(D, D') := \lambda i_s.\ \mathbf{MOST}(u[Di],\ u[(D; D')i])$,
i.e., $\mathbf{most}^{wk}{}_u(D, D') := \lambda i_s.\ |u[Di] \cap u[(D; D')i]| > |u[Di] \setminus u[(D; D')i]|$
$\mathbf{most}^{str}{}_u(D, D') := \lambda i_s.\ \mathbf{MOST}(u[Di],\ u[([D \rightarrow D'])i])$,
i.e., $\mathbf{most}^{str}{}_u(D, D') := \lambda i_s.\ |u[Di] \cap u[([D \rightarrow D'])i]| > |u[Di] \setminus u[([D \rightarrow D'])i]|$
111. Most^{u_1} house-elves who fall in love with a^{u_2} witch buy her$_{u_2}$ an^{u_3} alligator purse.
112. $[\mathbf{most}^{str}{}_{u_1}([u_2 \mid h.elf\{u_1\}, witch\{u_2\}, in\_love\{u_1, u_2\}],[u_3 \mid a.purse\{u_3\}, buy\{u_1, u_2, u_3\}])]$
113. $\lambda i_s.\ |\{x_e\!: h.elf(x) \wedge \exists y_e(witch(y) \wedge i.l(x, y)) \wedge$
$\forall y'_e(witch(y') \wedge i.l(x, y') \rightarrow \exists z_e(a.p(z) \wedge buy(x, y', z)))\}| >$
$|\{x_e\!: h.elf(x) \wedge \exists y'_e(witch(y') \wedge i.l(x, y') \wedge \neg\exists z_e(a.p(z) \wedge buy(x, y', z)))\}|$
114. Most^{u_1} drivers who have a^{u_2} dime will put it$_{u_2}$ in the meter.
115. $[\mathbf{most}^{wk}{}_{u_1}([u_2 \mid driver\{u_1\}, dime\{u_2\}, have\{u_1, u_2\}],\ [put\_in\_meter\{u_1, u_2\}])]$
116. $\lambda i_s.\ |\{x_e\!: driver(x) \wedge \exists y_e(dime(y) \wedge have(x, y) \wedge put\_in\_meter(x, y))\}| >$
$|\{x_e\!: driver(x) \wedge \exists y_e(dime(y) \wedge have(x, y)) \wedge$
$\forall y'_e(dime(y') \wedge have(x, y') \rightarrow \neg put\_in\_meter(x, y'))\}|$

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)((et)t)** for determiners and **t(tt)** for conditionals / adverbs of quantification.

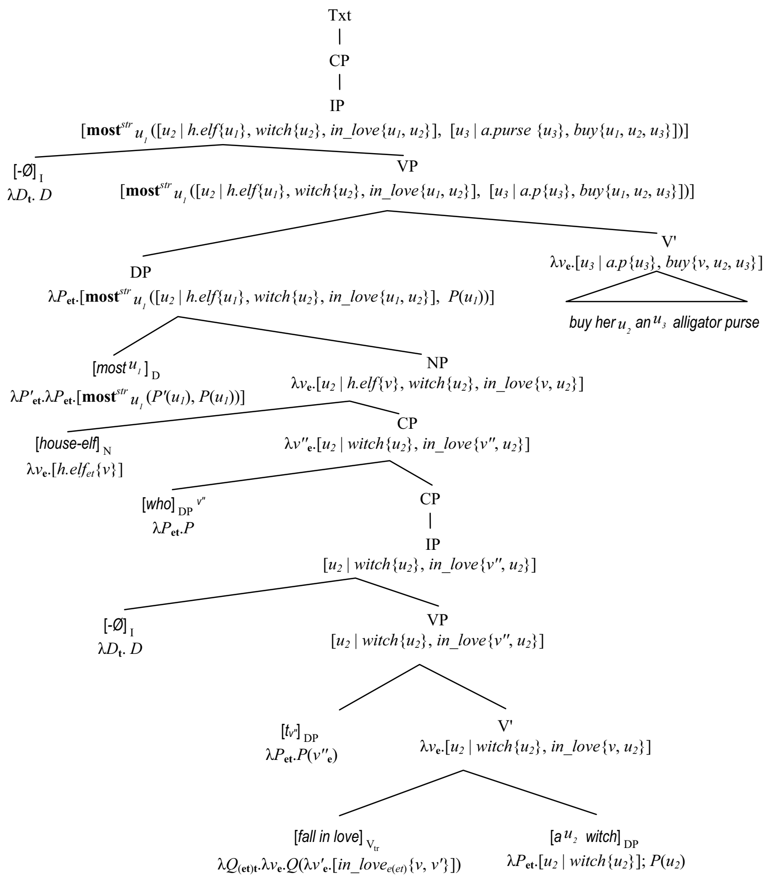
The meaning of the indefinite determiner *a* remains the same as in CDRT (redefining it in terms of selective generalized quantification would make it a test).

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

Lexical Item	Translation	Type $\mathbf{e} := se$ $\mathbf{t} := s(st)$
$[det^{wk,u}]_D$ / $[det^{str,u}]_D$ e.g., $every^{str,u}$, $no^{wk,u}$, $most^{str,u}$… (but not a^u)	$\rightsquigarrow \lambda P'_{\mathbf{et}}.\ \lambda P_{\mathbf{et}}.\ [\mathbf{det}^{wk/str}{}_u(P'(u), P(u))]$, where: $\mathbf{det}^{wk}{}_u(P'(u), P(u)) := \lambda i_s.\ \mathbf{DET}(u[P'(u)i],\ u[(P'(u); P(u))i])$ $\mathbf{det}^{str}{}_u(P'(u), P(u)) := \lambda i_s.\ \mathbf{DET}(u[P'(u)i],\ u[([P'(u) \rightarrow P(u)])i])$, where $P(u)i := \{j_s\!: P(u)ij\}$ and $u[P(u)i] := \{uj\!: ([u]; P(u))ij\}$ and **DET** is the corresponding static determiner.	**(et)((et)t)**
$[if$ (+*adv. of quant.*)$]_C$	$\rightsquigarrow \lambda D'_{\mathbf{t}}.\ \lambda D_{\mathbf{t}}.\ [\mathbf{det}(D', D)]$, where: $\mathbf{det}(D, D') := \lambda i_s.\ \mathbf{DET}(Di, (D; [!D'])i)$, where $Di := \{j_s\!: Dij\}$ and **DET** is the corresponding static determiner.	**t(tt)**
$[if]_C$ (i.e., bare *if*)	$\rightsquigarrow \lambda D'_{\mathbf{t}}.\ \lambda D_{\mathbf{t}}.\ [\mathbf{every}(D', D)]$	**t(tt)**

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

118. Most$^{\mathbf{str}, u_1}$ house-elves who fall in love with a^{u_2} witch buy her$_{u_2}$ an^{u_3} alligator purse.



9. *Anaphora and Generalized Coordination in CDRT+GQ*

Goals:

- provide the dynamic counterparts of the definitions of conjoinable 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).

119. Every^{u_1} boy who has a^{u_2} dog and every^{u_3} girl who has a^{u_2} cat must feed it$_{u_2}$.

- 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. $\mathbf{t} \in$ **DCTyp** (where $\mathbf{t} := s(st)$) and, if $\tau \in$ **DCTyp**, then $(\sigma\tau) \in$ **DCTyp** for any $\sigma \in$ **Typ**.

121. **Generalized Pointwise Dynamic Conjunction $\sqcap$ and Disjunction $\sqcup$.**
For any two terms α and β of type τ, for any $\tau \in$ **DCTyp**:

$\alpha \sqcap \beta := (\alpha; \beta)$ if $\tau = \mathbf{t}$ and $\alpha \sqcap \beta := \lambda v_\sigma.\ \alpha(v) \sqcap \beta(v)$ if $\tau = (\sigma\rho)$;

$\alpha \sqcup \beta := [\alpha \vee \beta]$ if $\tau = \mathbf{t}$ and $\alpha \sqcup \beta := \lambda v_\sigma.\ \alpha(v) \sqcup \beta(v)$ if $\tau = (\sigma\rho)$.

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 *and* and *or*:

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

Lexical Item	Translation	Type $\mathbf{e} := se$ $\mathbf{t} := s(st)$
[*and*]$_{Conj}$	$\rightsquigarrow \lambda v_1. \ldots \lambda v_n.\ v_1 \sqcap \ldots \sqcap v_n$	$\tau(\ldots(\tau\tau)\ldots)$
[*or*]$_{Conj}$	$\rightsquigarrow \lambda v_1. \ldots \lambda v_n.\ v_1 \sqcup \ldots \sqcup v_n$	$\tau(\ldots(\tau\tau)\ldots)$

9.2. Revising the Coordination Rule: Generalized Coordination

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

If $A_1 \rightsquigarrow \alpha_1, \ldots, A_n \rightsquigarrow \alpha_n$, Conj $\rightsquigarrow \beta$, $A_{n+1} \rightsquigarrow \alpha_{n+1}$ and $A_1, \ldots, A_n$, Conj and A_{n+1} are the only daughters of A in that order (i.e., $A \rightarrow A_1 \ldots A_n$ Conj A_{n+1}),

then $A \rightsquigarrow \beta(\alpha_1)\ldots(\alpha_n)(\alpha_{n+1})$, provided this a well-formed term and has the same type as $\alpha_1, \ldots, \alpha_n, \alpha_{n+1}$.

9.3. Catching and Eating a Fish in CDRT+GQ

Consider the sentences in (124) and (125) (Partee & Rooth 1983: 338, (12) and (13))[17].

124. John caught and ate a^{u_1} fish.
125. John hugged and kissed three^{u_1} 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*^{u_1} *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 GQ as direct object (a term of type (**et**)**t**) and not an individual dref (a term of type **e**) – 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^{u_2} [$_{\text{Vtr}}$ caught and ate] a^{u_1} fish.
127. $[u_2 \mid u_2{=}John]$; $[u_1 \mid fish\{u_1\}, catch\{u_2, u_1\}]$; $[u_1 \mid fish\{u_1\}, eat\{u_2, u_1\}]$

[17] Page references are to Partee & Portner (2002).

128. $\lambda i_s.\ \exists x_e(fish(x) \wedge catch(john, x)) \wedge \exists y_e(fish(y) \wedge eat(john, y))$

129. [a^{u_1} fish]$^{v''}$ [John^{u_2} [$_{\text{Vtr}}$ caught and ate] $t_{v''}$].
130. $[u_1 \mid fish\{u_1\}]$; $[u_2 \mid u_2{=}John]$; $[catch\{u_2, u_1\}, eat\{u_2, u_1\}]$,
i.e., $[u_1, u_2 \mid fish\{u_1\}, u_2{=}John, catch\{u_2, u_1\}, eat\{u_2, u_1\}]$
131. $\lambda i_s.\ \exists x_e(fish(x) \wedge catch(john, x) \wedge eat(john, x))$

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_1 in (127), is either banned or, if it is allowed, it is *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., *John needed and bought a new coat* (as Partee & Rooth 1983: 338 observe).

We would have to postulate a mechanism whereby the indefinite object *a 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 *Mary and Helen* in discourse (132-133) below.

132. Mary^{u_1} and Helen^{u_2} (each) bought an^{u_3} alligator purse.
133. They$_{u_3}$ were (both) fluorescent green.
134. $[u_1, u_3 \mid u_1{=}Mary, a.purse\{u_3\}, buy\{u_1, u_3\}]$;
$[u_2, u_3 \mid u_2{=}Helen\ , a.purse\{u_3\}, buy\{u_2, u_3\}]$;
$[fluorescent\_green\{u_3\}]$

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 indefinites *a*^{u_2} *dog* and *a*^{u_2} *cat* can be anaphorically associated with the donkey pronoun *it*$_{u_2}$.
- 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^{u_1} cat [$_{V'}$[$_{V'}$caught a^{u_2} fish] and [$_{V'}$ate it$_{u_2}$]].
136. $[u_1, u_2 \mid cat\{u_1\}, fish\{u_2\}, catch\{u_1, u_2\}, eat\{u_1, u_2\}]$
137. $\lambda i_s.\ \exists x_e \exists y_e(cat(x) \wedge fish(y) \wedge catch(x, y) \wedge eat(x, y))$

138. John^{u_4} has [$_{DP}$[$_{DP}$a^{u_1} cat which caught a^{u_2} fish] and [$_{DP}$a^{u_3} cat which ate it$_{u_2}$]].
139. $[u_4 \mid u_4{=}John]; [u_1, u_2 \mid cat\{u_1\}, have\{u_4, u_1\}, fish\{u_2\}, catch\{u_1, u_2\}];$
$[u_3 \mid cat\{u_3\}, have\{u_4, u_3\}\ eat\{u_3, u_2\}]$
140. $\lambda i_s.\ \exists x_e \exists y_e \exists z_e(cat(x) \wedge have(john, x) \wedge fish(y) \wedge catch(x, y) \wedge$
$cat(z) \wedge have(john, z) \wedge eat(z, y))$

141. John^{u_3} admires [$_{DP}$[$_{DP}$a^{u_1} girl] and [$_{DP}$a^{u_2} boy who loves her$_{u_1}$]].
142. $[u_3 \mid u_3{=}John]; [u_1 \mid girl\{u_1\}, admire\{u_3, u_1\}];$
$[u_2 \mid boy\{u_2\}, admire\{u_3, u_2\}, love\{u_2, u_1\}]$
143. $\lambda i_s.\ \exists x_e \exists y_e(girl(x) \wedge admire(john, x) \wedge boy(y) \wedge admire(john, y) \wedge love(y, x))$

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 infelicitous.

144. #A^{u_1} cat [$_{V'}$[$_{V'}$caught no^{u_2} fish] and [$_{V'}$ate it$_{u_2}$]].
145. #John^{u_4} has [$_{DP}$[$_{DP}$a^{u_1} cat which caught no^{u_2} fish] and [$_{DP}$a^{u_3} cat which ate it$_{u_2}$]].
146. #John^{u_3} admires [$_{DP}$[$_{DP}$no^{u_1} girl] and [$_{DP}$a^{u_2} boy who loves her$_{u_1}$]].

9.6. DP-Conjunction Donkey Sentences

147. [[Every$^{\mathbf{str}, u_1}$ boy who has a^{u_2} dog] and [every$^{\mathbf{str}, u_3}$ girl who has a^{u_2} cat]] must feed it$_{u_2}$.[18]
148. *every*$^{\mathbf{str}, u_1}$ *boy who has a*^{u_2} *dog* $\rightsquigarrow$
$\lambda P_{\mathbf{et}}.\ [\mathbf{every}^{str}{}_{u_1}([u_2 \mid boy\{u_1\}, dog\{u_2\}, have\{u_1, u_2\}],\ P(u_1))]$
149. *every*$^{\mathbf{str}, u_3}$ *girl who has a*^{u_2} *cat* $\rightsquigarrow$
$\lambda P_{\mathbf{et}}.\ [\mathbf{every}^{str}{}_{u_3}([u_2 \mid girl\{u_3\}, cat\{u_2\}, have\{u_3, u_2\}],\ P(u_3))]$
150. *every*$^{\mathbf{str}, u_1}$ *boy who has a*^{u_2} *dog and every*$^{\mathbf{str}, u_3}$ *girl who has a*^{u_2} *cat* $\rightsquigarrow$
$\lambda P_{\mathbf{et}}.\ [\mathbf{every}^{str}{}_{u_1}([u_2 \mid boy\{u_1\}, dog\{u_2\}, have\{u_1, u_2\}],\ P(u_1)),$
$\mathbf{every}^{str}{}_{u_3}([u_2 \mid girl\{u_3\}, cat\{u_2\}, have\{u_3, u_2\}],\ P(u_3))]$
151. *must feed it*$_{u_2}$ $\rightsquigarrow$ $\lambda v_{\mathbf{e}}.\ [must\_feed\{v, u_2\}]$
152. *every*$^{\mathbf{str}, u_1}$ *boy who has a*^{u_2} *dog and every*$^{\mathbf{str}, u_3}$ *girl who has a*^{u_2} *cat must feed it*$_{u_2}$ $\rightsquigarrow$
$[\mathbf{every}^{str}{}_{u_1}([u_2 \mid boy\{u_1\}, dog\{u_2\}, have\{u_1, u_2\}],\ [must\_feed\{u_1, u_2\}]),$
$\mathbf{every}^{str}{}_{u_3}([u_2 \mid girl\{u_3\}, cat\{u_2\}, have\{u_3, u_2\}],\ [must\_feed\{u_3, u_2\}])]$
153. $\lambda i_s.\ \forall x_e \forall y_e(boy(x) \wedge dog(y) \wedge have(x, y) \rightarrow must\_feed(x, y)) \wedge$
$\forall x'_e \forall y'_e(girl(x') \wedge cat(y') \wedge have(x', y') \rightarrow must\_feed(x', y'))$

[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$^{\mathbf{str},u_1}$ boy who has a^{u_2} dog] and [a^{u_3} girl]] must feed it$_{u_2}$.

155. [$\mathbf{every}^{str}{}_{u_1}$([$u_2$ | $boy\{u_1\}$, $dog\{u_2\}$, $have\{u_1, u_2\}$], [$must\_feed\{u_1, u_2\}$])]; [u_3 | $girl\{u_3\}$, $must\_feed\{u_3, u_2\}$]

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*$_{u_2}$ and the indefinite *a*^{u_2} *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*^{u_2} *dog* and *a*^{u_2} *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^{u_1} person who buys a^{u_2} book on amazon.com and has a^{u_3} credit card uses it$_{u_3}$ to pay for it$_{u_2}$.

157. Every^{u_1} man who wants to impress a^{u_2} woman and who has an^{u_3} Arabian horse teaches her$_{u_2}$ how to ride it$_{u_3}$.

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 $\mathbf{det}_u^{\mathbf{wk}:u',\mathbf{str}:u''}(D, D')$ 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. $\mathbf{det}_u^{\mathbf{str}:u'>>\mathbf{wk}:u''}(D_1, D_2) := \lambda i_s.\ \mathbf{DET}(u[D_1 i],\ u[([D_3 \rightarrow (D_4; D_2)])i])$
$\mathbf{det}_u^{\mathbf{str}:u'<<\mathbf{wk}:u''}(D_1, D_2) := \lambda i_s.\ \mathbf{DET}(u[D_1 i],\ u[(D_4; [D_3 \rightarrow D_2])i])$,
where D_3 is the subpart of D_1 constraining dref u'
and D_4 is the subpart of D_1 constraining dref u''.

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^{u} village is inhabited by a$^{u'}$ painter, it$_u$ is usually pretty. (Kadmon 1987)

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

161. If a^{u} woman owns a$^{u'}$ cat, she$_u$ usually talks to it$_{u'}$. (Heim 1990: 175, (91))

- the most salient reading of (159) is an asymmetric one in which we quantify over villages u inhabited by a painter; thus, that conditional is translated in CDRT+GQ by means of the selective determiner $\mathbf{most}^{wk}{}_u$ ($\mathbf{most}^{str}{}_u$ is equally adequate in this case)
- the most salient reading of (160) is an asymmetric one in which we quantify over apartment complexes u' inhabited by a drummer; hence, the conditional is translated in CDRT+GQ by means of the selective determiner $\mathbf{most}^{wk}{}_{u'}$ ($\mathbf{most}^{str}{}_{u'}$ is equally adequate in this case)
- the most salient reading of (161) is one where we quantify over woman-cat pairs; therefore, the conditional is translated in CDRT+GQ by means of the unselective determiner **most**

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 *if*+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*$^{u'}$ *son* receives a weak reading).

162. If a^{u} woman has a$^{u'}$ son with a$^{u''}$ man, she$_u$ usually keeps in touch with him$_{u''}$.

(I. Heim, apud Chierchia 1995: 67, (14b))