Quantification: QuDPs and the rest of the sentence

[Heim & Kratzer ch.7, Montague 1974, Barker 2002]

1. Introduction.

■ How can we interpret Quantificational DPs (QuDPs) in object position?

(1) Every linguist welcomed Lorenzo.
a. ⟦**linguist**⟧ = λx. LINGUIST(x)
b. ⟦**welcomed Lorenzo**⟧ = λx. WELCOME(x, l)
c. $\forall$y ⟦**linguist**⟧ (y) → ⟦**welcomed Lorenzo**⟧ (y)]
(in set terms: ⟦**linguist**⟧ ⊆ ⟦**welcomed Lorenzo**⟧)
d. $\forall$y [LINGUIST(y) → WELCOME(l)(y)]

(2) Lorenzo disappointed every linguist.
a. ⟦**linguist**⟧ = λx. LINGUIST(x)
b. ⟦ ??? ⟧ = λx. DISAPPOINT(l, x)
c. $\forall$y [⟦**linguist**⟧ (y) → ⟦ ??? ⟧ (y)]
d. $\forall$y [LINGUIST(y) → DISAPPOINT(l, y)]

■ Main avenues to solve this problem. As we saw with Modifiers, we always have two main choices: to put the burden of the interpretation in the Lexicon or to put it in the rules. Mixed possibilities also arise.

(3)

Locus of QuDP	Burden	Implementations
IN SITU	IN LEXICON	*Implementation 1: flexible types for QuDP[1] (*)Implementation 2: higher type for verb in PTQ[2]
DISPLACED	IN RULES	*Implementation 3.1: "Quantifying in" in PTQ *Implementation 3.2: Quantifier Raising in GB/Min.
IN SITU	IN RULES[3]	Implementation 4: Cooper's Quantifier Storage[4] Implementation 5: Categorial Grammar and variable-free semantics.[5] (*)Implementation 6: Continuations[6] …

[1] The version of flexible types for QuNPs that we will see here is quite simplified. A more elaborated version can be found in Hendriks, H. 1988, "Type change in semantics: The scope of quantification and coordination", in E. Klein and J. van Benthem (eds.), *Categories, Polymorphism and Unification*, pp. 96-119. ITLI, Amsterdam.

[2] Montague, Richard. "The Proper Treatment of Quantification in Ordinary English" (abbreviated PTQ). In R. H. Thomason (ed.). 1974. *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale Univ. Press.

[3] The in situ / burden-in-rules line fares like implementations 3.1 and 3.2 wrt the accomplishments we will review here.

[4] Cooper, R., 1983, *Quantification and syntactic theory*, Dordrecht: Reidel.

[5] See e.g. Jacobson, P. 1992, "Antecedent-contained deletion in variable-free semantics", *Proceedings for SALT II*.

[6] See e.g. Barker, C., 2003, "Continuations and the nature of quantification", *Natural Language Semantics* 10.

2. Implementation 1: in situ, flexible type approach for QuDPs.

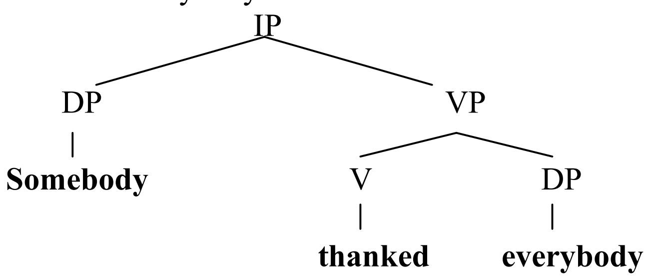
- Basic features:
  - QuDP are always interpreted where we “see” them in overt Syntax.
  - The burden of interpretation is in the Lexikon.
  - A given Det / QuDP (e.g. **some** / **some girl**) has several possible denotations, each of a different type.

- Main idea: Determiners (and, hence, the QuDPs they head) have not just one type, but several possible ones, depending on their position in the sentence (the different denotations would be related, e.g., by Type Shifting Principles).

(4) a. ⟦**everybody** $_{SU}$⟧ = $\lambda Q_{<e,t>}.\ \forall x_e\ Q(x)$
b. ⟦**everybody** $_{DO}$⟧ = $\lambda Q_{<e<e,t>>}\lambda y_e.\ \forall x_e\ Q(x)(y)$

(5) a. ⟦**somebody** $_{SU}$⟧ = $\lambda Q_{<e,t>}.\ \exists x_e\ Q(x)$
b. ⟦**somebody** $_{DO}$⟧ = $\lambda Q_{<e<e,t>>}\lambda y_e.\ \exists x_e\ Q(x)(y)$

(6) Somebody thanked everybody.



QUESTION: Give the following denotations:

(7) a. ⟦**every** $_{SU}$⟧ =
b. ⟦**every** $_{DO}$⟧ =

(8) a. ⟦**some** $_{SU}$⟧ =
b. ⟦**some** $_{DO}$⟧ =

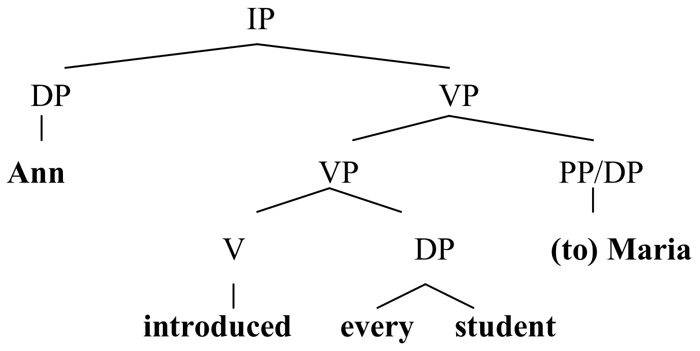
(9) a. ⟦**nobody** $_{SU}$⟧ =
b. ⟦**nobody** $_{DO}$⟧ =

(10) a. ⟦**no** $_{SU}$⟧ =
b. ⟦**no** $_{DO}$⟧ =

EXERCISE: Spell out the denotation of **every**$_{\mathbf{DOditr}}$ and do the semantic computation of the following syntactic tree:

(11) ⟦**every** $_{DOditr}$⟧ =

(12) Ann introduced every student to Maria. (≈H&K p. 181)



3. Implementation 2: higher type for verb (in PTQ).

(13) ⟦**kiss**⟧ = $\lambda Q_{<et,t>}\lambda x_e.\ Q\ (\lambda y.\text{KISS}(x,y))$

(14) John seeks a unicorn.
a. $\text{SEEK}\ (j,\ \lambda P_{<et>}\exists x[\text{UNICORN}(x)\ \&\ P(x)]\ )$

4. Implementation 3.1: Montague's "Quantifying in" (in PTQ)

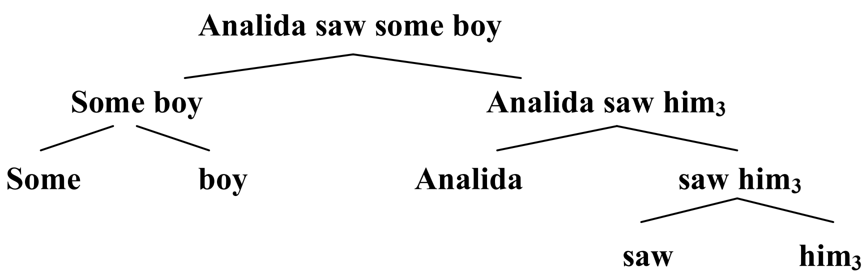
■ Basic features:

- QuDPs have always the same type, no matter their syntactic function: <et,t>
- The burden of interpretation is in the semantic rules.
- QuDP may be interpreted in positions other than where we "see" them.

■ In Montague's PTQ, we have many more semantic rules than just Functional Application. In fact, syntactic trees are built in stages by means of different syntactic rules, and each syntactic rule has its own corresponding semantic rule. One of this syntactic-semantic duplets is the rule of "Quantifying in", which is in charge of inserting QuDP in their scope position.

(15) Syn "Quantifying in": [Glossing over NP_e and agree't]
If α is an expression of type <<et>,t> and ϕ is an expression of type t, then $Syn_{Qu\text{-}in,n}$ (α,ϕ) is an expression of type t, where
$Syn_{Qu\text{-}in,n}$ (α,ϕ) comes from replacing the first occurrence of a pronoun with index n by α.

(16) Analida saw some boy.



(17) Sem "Quantifying in": [Slightly adapted to our notation]
If α is an expression of type <<et>,t> and ϕ is an expression of type t, then
$$[\![Syn_{Qu\text{-}in,n}(\alpha,\phi)]\!] = [\![\alpha]\!]\ (\lambda x.\ [\![\phi]\!]^{gx/n})$$

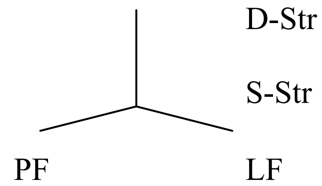
5. Implementation 3.2: Quantifier Raising (QR) in GB / Minimalist Syntax.

■ Basic features:

- QuDPs have always the same type, no matter their syntactic function: <et,t>
- The burden of interpretation is in the semantic rules (but fewer rules than in Montague).
- QuDP may be interpreted displaced, in positions other than where we "see" them.

■ In GB (and some versions of Minimalism), syntactic trees are built in stages, by merging nodes together and moving some of them up the tree. At some point in that derivation, the structure is sent to the phonological module of the grammar, encoded in a phonological representation (Phonological Form, PF) and pronounced. But the syntactic structure that serves as the basis for PF may still undergo some more derivations covertly until it reaches its final state (Logical Form, LF), which is then fed to the semantic interpretation rules.

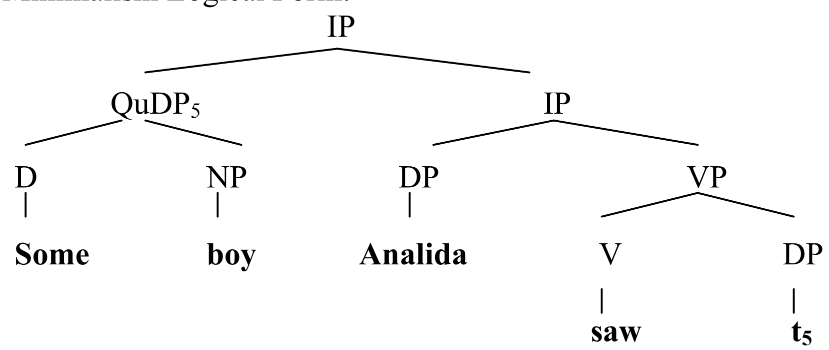
(18)



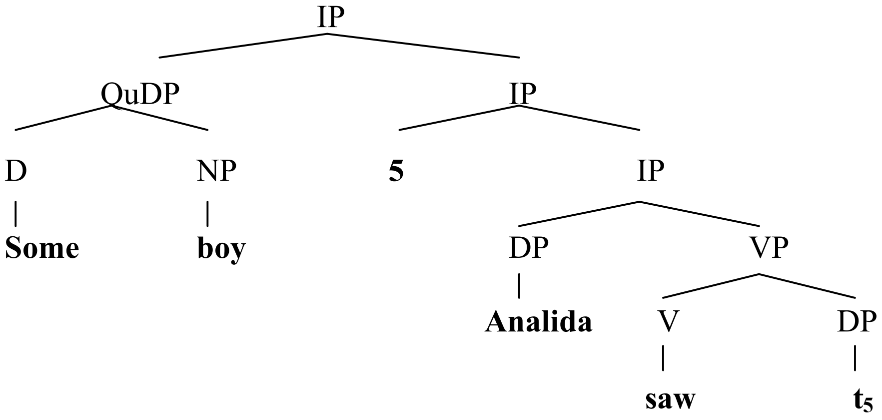
■ A QuDP that is the Theme of a transitive verb is, first, inserted in argument position. The structure is sent to PF. However, this configuration cannot be interpreted because there is a type mismatch. This type mismatch is resolved by moving the QuDP to adjunct IP position at Logical Form (by an operation called "Quantifier Raising"). Movement requires for the moved QuDP and its trace to be coindexed. The question we are interested in is how to interpret the adjunction branching node and, more concretely, the index on the moved QuDP.

(19) Analida saw some boy.

a. GB/Minimalism Logical Form:



b. Proposal to interpret the index of QuDP:



(20)	Predicate Abstraction (PA): [New Version] If α has the form α , where ~~β is a relative pronoun or **such**, and~~ $i \in \mathbb{N}$, $\cancel{\beta}_i \quad \gamma$ then $[\![\alpha]\!]^{s,g} = \lambda x \in D_e.\ [\![\gamma]\!]^{gx/i}$

6. Implementation 6: Continuations.

(21) Types:

P	<e,t>
x	e
p	<t,t>
{{NP}}	<et,t>
{{VP}}	<<et,t>,t>

(22) a. S $\rightarrow$ NP VP $\lambda p.\{\{VP\}\}\ (\lambda P.\{\{NP\}\}(\lambda x.p(Px)))$

b. S $\rightarrow$ NP VP $\lambda p.\{\{NP\}\}\ (\lambda x.\{\{VP\}\}(\lambda P.p(Px)))$

7. Three standard arguments to choose among the different implementations.

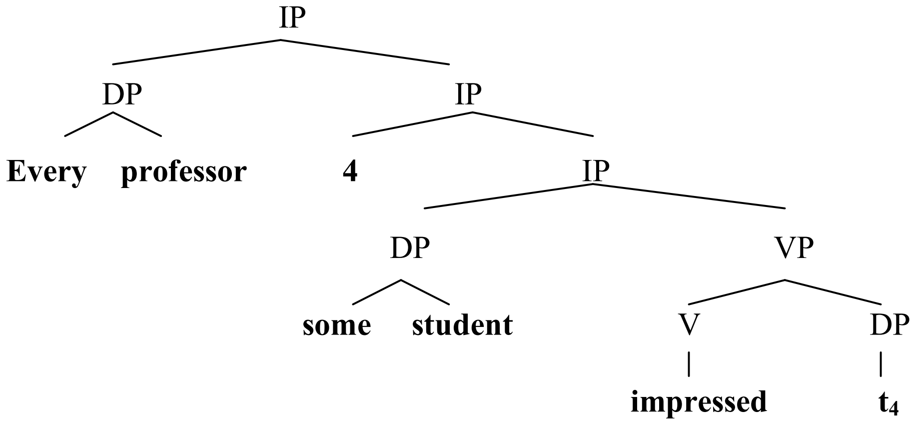
7.1. Scope ambiguity.

■ How can we derive the reading (b)?

(23) Some student impressed every professor.
a. ∃ >> ∀: "There is a student x such that, for every professor y, x impressed y."
b. ∀ >> ∃: "For every professor y, there is a (possibly different) student x such that x impressed y."

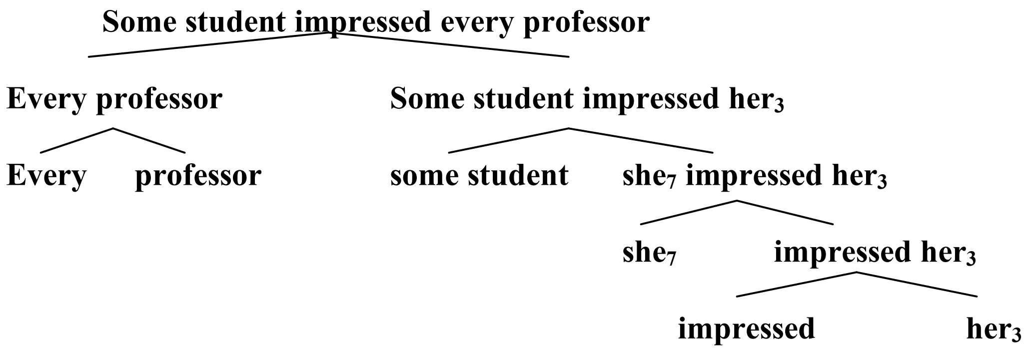
■ Implementation 3.2: Quantifier Raising may move the bottom QuDP over the top QuDP.

(24) Some student impressed every professor.



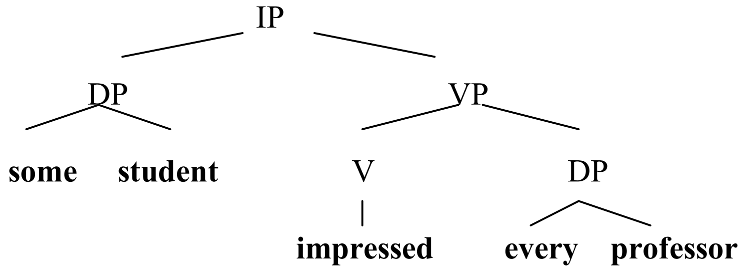
■ Implementation 3.1: the QuDP targeting the object pronoun can be "quantified in" after the QuDP targeting the subject pronoun has been "quantified in".

(25) Some student impressed every professor.



■ Implementation 1: in situ and flexible types for QuDPs. What reading(s) can you derive?

(26) Some student impressed every professor.



7.2. Antecedent Contained Deletion (ACD). [Sag 1976, May 1985]

■ Elided constituents in general need to find an antecedent in the discourse:

(27) VP-Ellipsis:
a. I went to Tanglewood during the summer, and Susanne did, too.
b. I went to Tanglewood during the summer, and Susanne did [$_{VP}$ go to Tanglewood during the summer], too.

(28) ACD:
a. I introduced Mary to everyone you did.
b. I introduced Mary to everyone (that$_1$) you did [$_{VP}$ introduce Mary to t_1]

■ If the QuDP **everyone (that$_1$) you did introduced Mary to t_1** is interpreted outside the matrix VP, an antecedent VP is easily available. If we leave the QuDP in situ, we run into an infinite regress problem, since the antecedent VP contains an ellipsis site that needs an antecedent itself. It's not clear how an in situ approach would solve that.[7]

7.3. Quantifiers that bind pronouns.

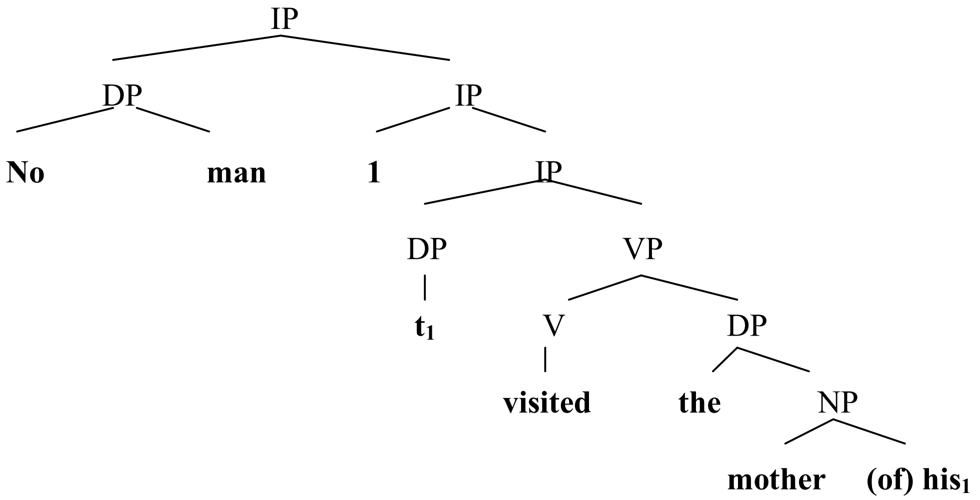
■ Example:

(29) No man$_1$ visited his$_1$ mother.

■ Implementation 3.2: Quantifier Raising.
We can always QR a QuDP, even if there is no type mismatch. Both *his* and the trace of QR are interpreted as variables bound by the moved QuDP (and similarly in implementation 3.1).

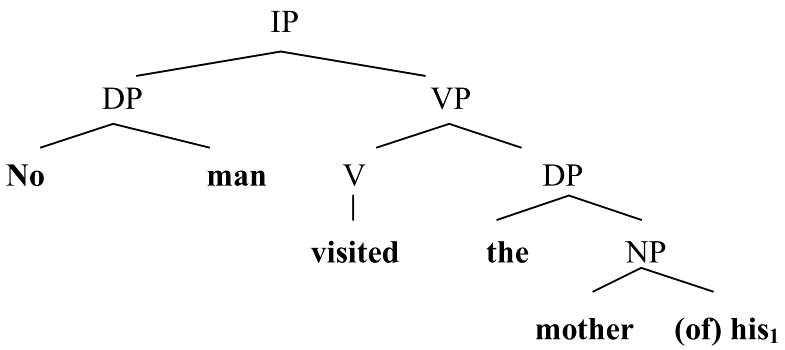
[7] Cooper's Quantifier Storage and variable-free semantics may overcome this problem.

(30) No man_1 visited his_1 mother.

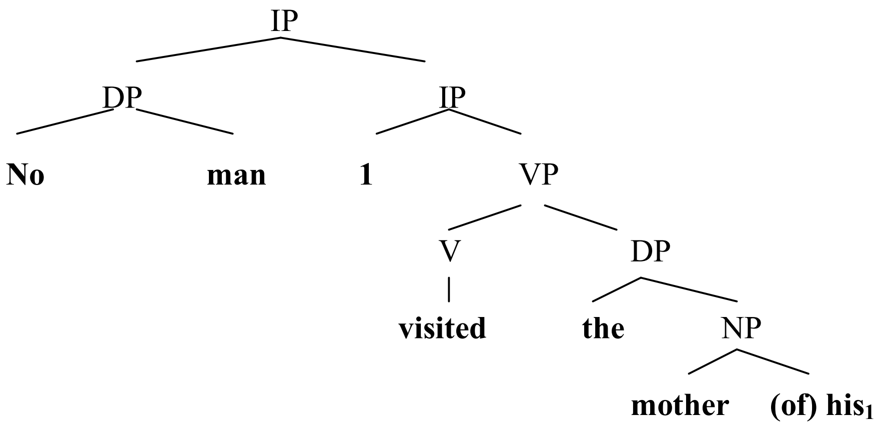


■ Implementation 1: in situ and flexible types for QuDPs.

(31) No man_1 visited his_1 mother.



(32) No man_1 visited his_1 mother.



7.4. Summary.

(33)

Place of QuDP	Burden	Implementations	Amb.	ACD	Pron
IN SITU	IN LEXICON	1: Flexible types.	*	*	*
DISPLACED	IN RULES	3.1: "Quantifying in"	√	√	√
		3.2: QR	√	√	√