Degree Quantification and Split Scope in Glue Semantics[*]

Ash Asudeh
University of Oxford
ash.asudeh@ling-phil.ox.ac.uk

February 23, 2015
University College London

1 Introduction

Certain quantifiers display a curious capability in some contexts:

(1) The company need fire no employees. (Potts 2000)

One reading of (1) is that it is not the case that the company is obligated to fire employees (Potts 2000). It's as if *no employees* has divided into separate negative and indefinite parts and these have taken separate scopes, with the negation scoping above the intensional verb and the indefinite scoping under it. This phenomenon is often called 'split scope', and has been a feature of the landscape for some time (among others, Jacobs 1980, Rullmann 1995, de Swart 2000, Heim 2001, Abels and Martí 2010).

Heim (2001) conjectures that split scope may be related to issues to do with degree quantification. She argues that DegPs have a kind of generalized quantifier interpretation but with respect to degrees: they are of type $\langle\langle d, t\rangle, t\rangle$, where d is the type of degrees. This predicts scoping interactions contrary to claims by Kennedy (1997) that degree quantifiers generally take lowest scope (with important exceptions).

Abels and Martí (2010) identify a number of generalizations, building on the previous literature. I'll return to these toward the end of the talk, but I want to focus on a generalization from Heim (2001), which I'll call Generalization H, and a version of Kennedy's Generalization, which I'll call Generalization K.

(2) **Generalization H**
Degree operators cannot scope over quantificational DPs. (Heim 2001)

(3) **Generalization K**
Degree operators can scope over intensional verbs. (Kennedy 1997)

I will present a non-QR version of (a fraction of) the DegP facts (without positing a DegP) and show how we can derive these generalizations from a single type assignment to degree operators that treats them not as type $\langle\langle d, t\rangle, t\rangle$, but rather as modifiers of type $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$. In other words, following Heim (2001), I'm focusing on the scopal behaviour of degree operators rather than on split scope per se, but the analysis I give is fundamentally different than Heim's, since it assigns the degree operator a different type. The analysis uses Glue Semantics (Dalrymple et al. 1993, Dalrymple 1999, 2001, Asudeh 2012) and its standard scoping mechanism, but in a novel way. I'll return to split scope toward the end of the talk.

[*] This work was supported by NSERC Discovery Grant #371969.

2 Overview of the Talk

1. Introduction
2. This
3. The Problem
4. Theoretical Background: LFG
5. Theoretical Background: Glue Semantics
6. Analysis
7. Conclusion

3 The Problem

Heim (2001: 217–221) points out that the scope-taking capabilities of degree phrases are often obscured by "systematic equivalences and anomalies". However, the relevant equivalences do not hold for *exactly*- and *less*-comparatives. Here is one of her examples (Heim 2001: 222):

(4) (John is $4'$ tall.) Every girl is exactly $1''$ taller than that.

She observes that a wide scope reading for the degree phrase would result in a reading such that no girl is shorter than $4'1''$, but some may be taller. This doesn't seem to be an available reading. Rather, the sentence means that every girl is $4'1''$ tall, which is the reading with the degree phrase taking narrow scope. Based on this and other examples, she concludes that degree phrases cannot outscope quantificational DPs, i.e. Generalization H.

However, with intensional verbs Heim (2001) observes a different pattern (again, one of her examples):

(5) (This draft is 10 pages.) The paper is allowed to be exactly 5 pages longer than that.

Here the degree phrase can take wide scope, which yields a reading in which the paper is 15 pages long at most. Based on this and other examples, Heim concludes that degree phrases can outscope intensional verbs.

The basic problem, then, is this: *Can we give a consistent, compositional treatment of degree phrases such that these generalizations follow?*

The **solution** that I propose rests on two main moves:

1. Degree phrases are analyzed as modifiers, not degree quantifiers.
2. They nevertheless contribute a *scope point*, through the general scoping mechanism of Glue Semantics. However, the scope point is specified such that we automatically derive a *scope trapping/freezing* effect with respect to quantified DPs.

4 Theoretical Background: Lexical-Functional Grammar

- Glue Semantics is really a theory of semantic composition and the syntax–semantics interface. It can be paired with any syntactic theory, in principle. For example, Asudeh and Crouch (2002) define Glue Semantics for HPSG and Gotham (2015) has recently been developing Glue Semantics for Minimalism. However, it was originally developed for and has been most closely associated with Lexical-Functional Grammar.
- LFG is a declarative, constraint-based linguistic theory (Kaplan and Bresnan 1982).

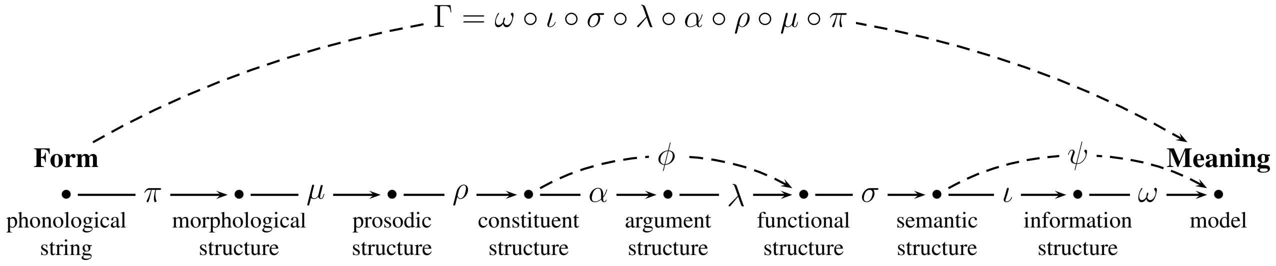


Figure 1: The Correspondence Architecture, pipeline version (Asudeh 2012)

- The motivation behind LFG is to have a theory that contributes in three ways to our understanding of language:

  1. Theory, including language universals and typology
  2. Psycholinguistics, including language acquisition
  3. Computational linguistics, including automatic parsing and generation, machine translation, and language modelling

4.1 The Correspondence Architecture

- The grammatical architecture of LFG posits that different kinds of linguistic information are modelled by distinct data structures, all of which are present simultaneously.

- Structures are related by functions, called correspondence or projection functions., which map elements of one structure to elements of another.

- This architecture is a generalization of the architecture of Kaplan and Bresnan (1982) and is called the *Parallel Projection Architecture* or *Correspondence Architecture* (Kaplan 1987, 1989, Halvorsen and Kaplan 1988, Asudeh 2006, 2012, Asudeh and Toivonen 2009).

- Syntax: constituent structure (c-structure) and functional structure (f-structure).

- C-structure is represented by phrase structure trees:

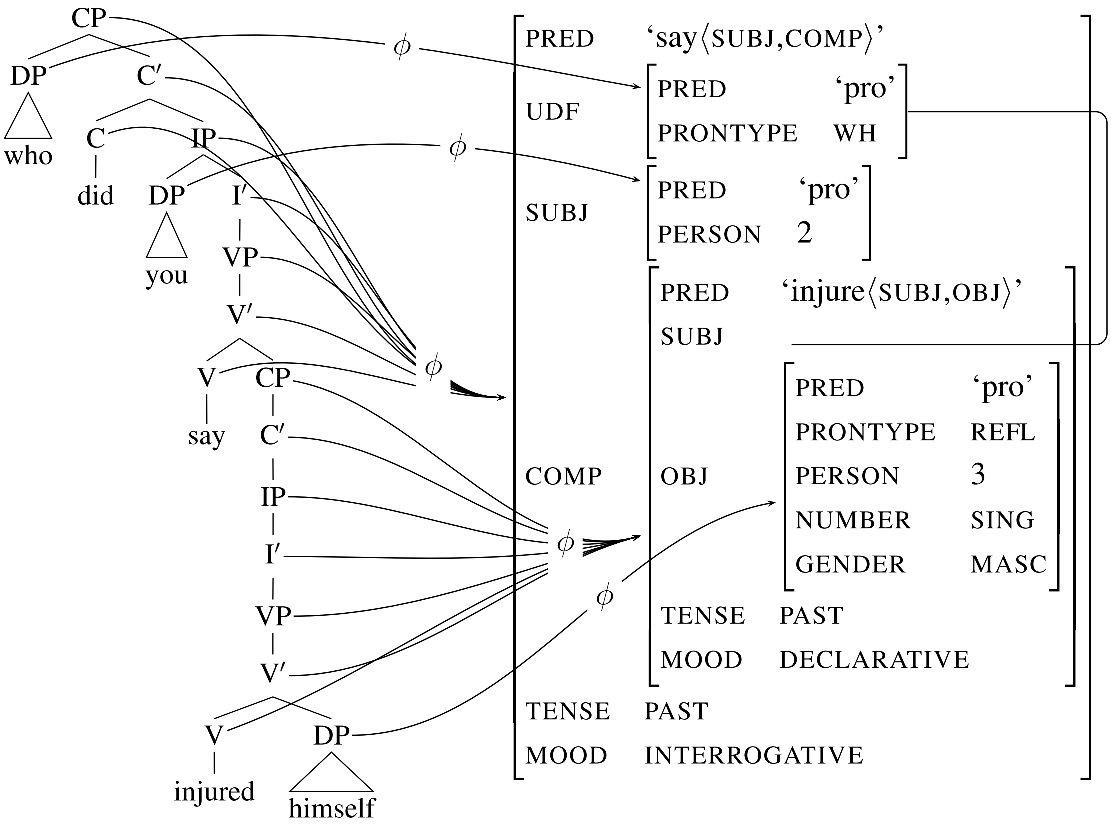
  1. Word order
  2. Dominance
  3. Constituency
  4. Syntactic categories

- F-structure is represented by feature structures (also known as attribute value matrices):

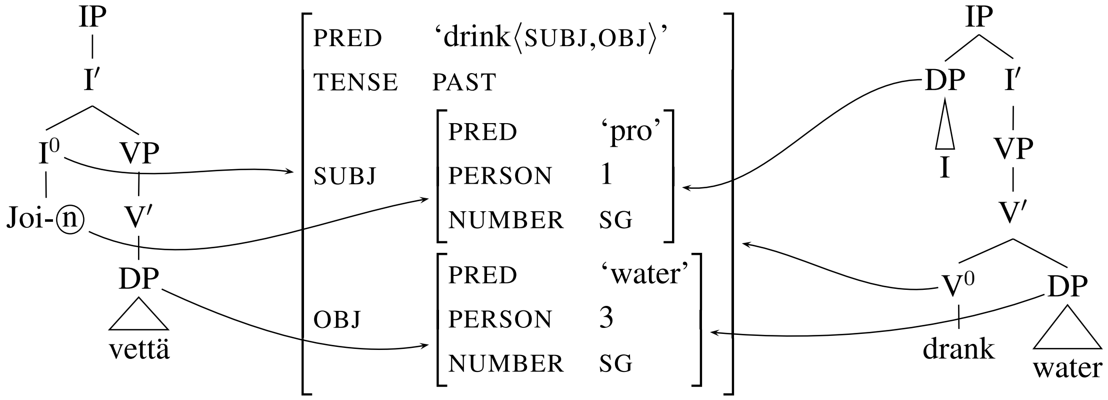
  1. Grammatical functions, such as SUBJECT and OBJECT
  2. Case
  3. Agreement
  4. Tense and aspect
  5. Local dependencies (e.g., control and raising)
  6. Unbounded dependencies (e.g., question formation, relative clause formation)

4.2 Examples

(6)



(7)



4.3 Detailed Example

(8) Kim hugged Robin.

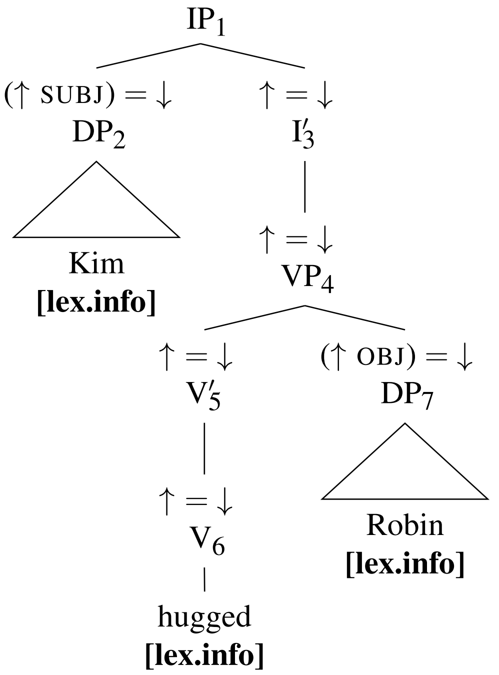
(9) **Lexical entries**

Kim D $(\uparrow \textsc{pred}) = \text{`Kim'}$
$(\uparrow \textsc{pers}) = 3$
$(\uparrow \textsc{num}) = \textsc{sg}$

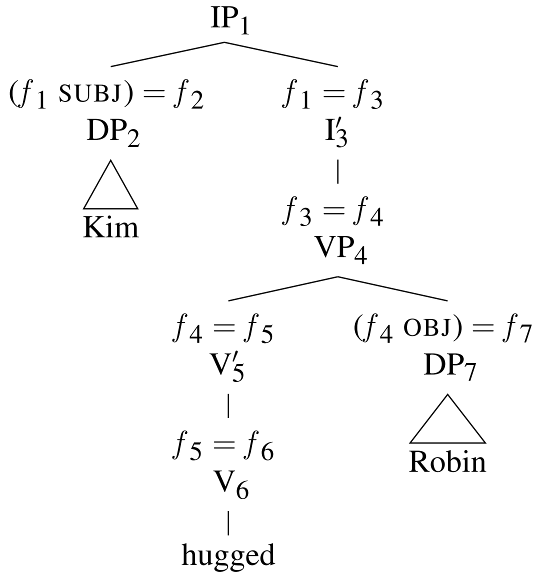
hugged V $(\uparrow \textsc{pred}) = \text{`hug}\langle\textsc{subj}, \textsc{obj}\rangle\text{'}$
$(\uparrow \textsc{tense}) = \textsc{past}$

Robin D $(\uparrow \textsc{pred}) = \text{`Robin'}$
$(\uparrow \textsc{pers}) = 3$
$(\uparrow \textsc{num}) = \textsc{sg}$

(10)



(11)



$$f_1, f_3, f_4, f_5, f_6 \left[\begin{array}{ll} \textsc{pred} & \text{`hug}\langle\textsc{subj}, \textsc{obj}\rangle\text{'} \\ \textsc{subj} & f_2 \left[\begin{array}{ll} \textsc{pred} & \text{`Kim'} \\ \textsc{pers} & 3 \\ \textsc{num} & \textsc{sg} \end{array}\right] \\ \textsc{obj} & f_7 \left[\begin{array}{ll} \textsc{pred} & \text{`Robin'} \\ \textsc{pers} & 3 \\ \textsc{num} & \textsc{sg} \end{array}\right] \\ \textsc{tense} & \textsc{past} \end{array}\right]$$

5 Theoretical Background: Glue Semantics

- Glue Semantics (Dalrymple 1999, 2001, Asudeh 2004, 2005, 2012, Lev 2007, Kokkonidis 2008) is a theory of semantic composition and the syntax–semantics interface.

- Glue *meaning constructors* are obtained from lexical items instantiated in particular syntactic structures.

  (12) $\mathcal{M} : G$

  $\mathcal{M}$ is a term from some representation of meaning, a *meaning language*, and G is a term of the Glue logic that sticks meanings together, i.e. performs composition. The colon is an uninterpreted pairing symbol.

- Linear logic (Girard 1987) serves as the glue logic (Dalrymple et al. 1993, 1999a,b).

- The meaning constructors are used as premises in a (linear logic) proof that consumes the lexical premises to produce a sentential meaning.

- A successful Glue proof for a sentence terminates in a meaning constructor of type t:

  (13) $\Gamma \vdash \mathcal{M} : G_t$

- Alternative derivations from the same set of premises: semantic ambiguity (e.g., scope)

- Linear logic is a *resource logic*: each premise in valid linear logic proof must be used exactly once.

- As discussed in detail by Dalrymple et al. (1999a), Glue Semantics is essentially a type-logical theory and is thus related to type-logical approaches to Categorial Grammar (Morrill 1994, Moortgat 1997, Carpenter 1997, Jäger 2005).

- The key difference between Glue and Categorial Grammar concerns grammatical architecture, particularly the conception of the syntax–semantics interface (Asudeh 2004, 2005, 2006). Glue Semantics posits a strict separation between syntax and semantics, such that there is a syntax that is separate from the syntax of semantic composition. Categorial Grmamar rejects the separation of syntax from semantic composition.

- I assume a small, rather weak fragment of linear logic, multiplicative intuitionistic linear logic (MILL; Asudeh 2004, 2005, 2012).

- Two proof rules of this fragment are of particular interest here: introduction and elimination for linear implication $\multimap$.

Application : Impl. Elim. Abstraction : Impl. Intro.

$$\dfrac{\begin{matrix}\vdots & \qquad & \vdots\\ a : A & \qquad & f : A \multimap B\end{matrix}}{f(a) : B}\multimap_{\mathcal{E}} \qquad \dfrac{\begin{matrix}[x : A]^1\\ \vdots\\ f : B\end{matrix}}{\lambda x.f : A \multimap B}\multimap_{\mathcal{I},1}$$

Figure 2: Linear logic proof rules with Curry-Howard correspondence

Notes on proof conventions:

- Linear logic terms are based on mappings from functional structure (syntax) to semantic structure (semantics). For example, $(\uparrow \textsc{subj})_\sigma$ is the semantic structure correspondent of the subject (SUBJ) of some predicate, where the predicate is designated by $\uparrow$.
- In proofs, these linear logic terms are abbreviated mnemonically. This is generally obvious.
- Scope points for quantifiers are represented as dependencies on universally quantified second-order variables of type t in the linear logic. For example, the term $\forall X.(a \multimap X) \multimap X$ states that any dependency on resource a that results in a type t resource can be discharged to yield just the type t resource. Notice that given the resource-sensitive nature of linear logic, the proper interpretation of this $\forall$ is as *any* (Asudeh 2005), not *all* ("pick any match for X", not "pick all matches for X").

5.1 Detailed Examples

5.1.1 General Semantic Lexicon

Expression	Type	Meaning Constructor
Kim	e	$kim : \uparrow_\sigma$
Robin	e	$robin : \uparrow_\sigma$
hugged	$e \rightarrow (e \rightarrow t)$	$\lambda y.\lambda x.hug(x,y) : (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$
everybody	$(e \rightarrow t) \rightarrow t$	$\lambda P.\forall y.[person(y) \rightarrow P(y)] : \forall X.(\uparrow_\sigma \multimap X) \multimap X$
somebody	$(e \rightarrow t) \rightarrow t$	$\lambda P.\exists x.[person(x) \wedge P(x)] : \forall Y.(\uparrow_\sigma \multimap Y) \multimap Y$

5.1.2 Example I: Transitive with Proper Name Arguments

(14) Kim hugged Robin.

(15)
$$h\left[\begin{array}{ll}\textsc{pred} & \text{`hug}\langle\textsc{subj},\textsc{obj}\rangle\text{'} \\ \textsc{subj} & k\left[\begin{array}{ll}\textsc{pred} & \text{`Kim'}\end{array}\right] \\ \textsc{obj} & r\left[\begin{array}{ll}\textsc{pred} & \text{`Robin'}\end{array}\right]\end{array}\right]$$

Instantiated lexicon/premises for (14):

Kim	e	$kim : k_\sigma$
Robin	e	$robin : r_\sigma$
hugged	$e \rightarrow (e \rightarrow t)$	$\lambda y.\lambda x.hug(x,y) : r_\sigma \multimap k_\sigma \multimap h_\sigma$

Full Proof for (14):

$$\dfrac{\dfrac{\dfrac{\dfrac{\lambda y.\lambda x.hug(x,y) : r_\sigma \multimap k_\sigma \multimap h_\sigma \qquad robin : r_\sigma}{(\lambda y.\lambda x.hug(x,y))(robin) : k_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}}}{\lambda x.hug(x,robin) : k_\sigma \multimap h_\sigma}\Rightarrow_\beta \qquad kim : k_\sigma}{(\lambda x.hug(x,robin))(kim) : h_\sigma}\multimap_{\mathcal{E}}}{hug(kim,robin) : h_\sigma}\Rightarrow_\beta$$

5.1.3 Example II: Transitive with Quantifier Object

(16) Kim hugged everybody.

(17)
$$h\left[\begin{array}{ll}\textsc{pred} & \text{‘hug}\langle\textsc{subj},\textsc{obj}\rangle\text{’}\\ \textsc{subj} & k\left[\begin{array}{ll}\textsc{pred} & \text{‘Kim’}\end{array}\right]\\ \textsc{obj} & e\left[\begin{array}{ll}\textsc{pred} & \text{‘everybody’}\end{array}\right]\end{array}\right]$$

Instantiated lexicon/premises for (16):

Kim	e	$kim : k_\sigma$
hugged	$e \to (e \to t)$	$\lambda y.\lambda x.hug(x,y) : e_\sigma \multimap k_\sigma \multimap h_\sigma$
everybody	$(e \to t) \to t$	$\lambda P.\forall y.[person(y) \to P(y)] : \forall X.(e_\sigma \multimap X) \multimap X$

Proof for (16), abridged for $\Rightarrow_\beta$:

$$\dfrac{\dfrac{\begin{array}{l}\lambda P.\forall y.[person(y) \to P(y)] :\\ \forall X.(e_\sigma \multimap X) \multimap X\end{array}}{\begin{array}{l}\lambda P.\forall y.[person(y) \to P(y)] :\\ (e_\sigma \multimap h_\sigma) \multimap h_\sigma\end{array}}\;\forall_{\mathcal{E}}[h/X] \qquad \dfrac{\dfrac{\dfrac{\begin{array}{l}\lambda y.\lambda x.hug(x,y) :\\ e_\sigma \multimap k_\sigma \multimap h_\sigma\end{array} \quad [z : e]^1}{\lambda x.hug(x,z) : k_\sigma \multimap h_\sigma}\;\multimap_{\mathcal{E}}, \Rightarrow_\beta \qquad k : kim}{hug(kim,z) : h_\sigma}\;\multimap_{\mathcal{E}}, \Rightarrow_\beta}{\lambda z.hug(kim,z) : e_\sigma \multimap h_\sigma}\;\multimap_{\mathcal{I},1}}{\forall y.[person(y) \to hug(kim,y)] : h_\sigma}\;\multimap_{\mathcal{E}}, \Rightarrow_\beta$$

Conventional, further abridged proof for (16):

$$\dfrac{\begin{array}{l}\lambda P.\forall y.[person(y) \to P(y)] :\\ \forall X.(e_\sigma \multimap X) \multimap X\end{array} \qquad \dfrac{\dfrac{\dfrac{\begin{array}{l}\lambda y.\lambda x.hug(x,y) :\\ e_\sigma \multimap k_\sigma \multimap h_\sigma\end{array} \quad [z : e]^1}{\lambda x.hug(x,z) : k_\sigma \multimap h_\sigma}\;\multimap_{\mathcal{E}}, \Rightarrow_\beta \qquad k : kim}{hug(kim,z) : h_\sigma}\;\multimap_{\mathcal{E}}, \Rightarrow_\beta}{\lambda z.hug(kim,z) : e_\sigma \multimap h_\sigma}\;\multimap_{\mathcal{I},1}}{\forall y.[person(y) \to hug(kim,y)] : h_\sigma}\;\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[h/X], \Rightarrow_\beta$$

5.1.4 Example III: Scope Ambiguity

(18) Everybody hugged somebody.

(19)
$$h\left[\begin{array}{ll}\text{PRED} & \text{‘hug}\langle\text{SUBJ},\text{OBJ}\rangle\text{’}\\ \text{SUBJ} & e\left[\begin{array}{ll}\text{PRED} & \text{‘everybody’}\end{array}\right]\\ \text{OBJ} & s\left[\begin{array}{ll}\text{PRED} & \text{‘somebody’}\end{array}\right]\end{array}\right]$$

Instantiated lexicon/premises for (18):

hugged	$e \to (e \to t)$	$\lambda y.\lambda x.hug(x,y) : s_\sigma \multimap e_\sigma \multimap h_\sigma$
everybody	$(e \to t) \to t$	$\lambda P.\forall y.[person(y) \to P(y)] : \forall X.(e_\sigma \multimap X) \multimap X$
somebody	$(e \to t) \to t$	$\lambda P.\exists x.[person(x) \wedge P(x)] : \forall Y.(s_\sigma \multimap Y) \multimap Y$

Abridged ∃∀ proof for (18):

$$\dfrac{\begin{array}{c}\lambda P.\exists x.[person(x) \wedge P(x)] :\\ \forall Y.(s_\sigma \multimap Y) \multimap Y\end{array} \quad \dfrac{\dfrac{\begin{array}{c}\lambda P.\forall y.[person(y) \to P(y)] :\\ \forall X.(e_\sigma \multimap X) \multimap X\end{array} \quad \dfrac{\begin{array}{c}\lambda y.\lambda x.hug(x,y) :\\ s_\sigma \multimap e_\sigma \multimap h_\sigma\end{array} \quad [z : s]^1}{\lambda x.hug(x,z) : e_\sigma \multimap h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \Rightarrow_\beta}}{\forall y.[person(y) \to hug(y,z)] : h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[h/X], \Rightarrow_\beta}}{\lambda z.\forall y.[person(y) \to hug(y,z)] : s_\sigma \multimap h_\sigma}{\scriptstyle \multimap_{\mathcal{I},1}}}{\exists x.[person(x) \wedge \forall y.[person(y) \to hug(y,x)] : h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[h/Y], \Rightarrow_\beta}$$

Abridged ∀∃ proof for (18):

$$\dfrac{\begin{array}{c}\lambda P.\forall y.[person(y) \to P(y)] :\\ \forall X.(e_\sigma \multimap X) \multimap X\end{array} \quad \dfrac{\dfrac{\begin{array}{c}\lambda P.\exists x.[person(x) \wedge P(x)] :\\ \forall Y.(s_\sigma \multimap Y) \multimap Y\end{array} \quad \dfrac{\dfrac{\dfrac{\begin{array}{c}\lambda y\lambda x.hug(x,y) :\\ s_\sigma \multimap e_\sigma \multimap h_\sigma\end{array} \quad [y_1 : s]^1}{\lambda x.hug(x,y_1) : e_\sigma \multimap h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \Rightarrow_\beta} \quad [z : e]^2}{hug(z,y_1) : h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \Rightarrow_\beta}}{\lambda y_1.hug(z,y_1) : s_\sigma \multimap h_\sigma}{\scriptstyle \multimap_{\mathcal{I},1}}}{\exists x.[person(x) \wedge hug(z,x)] : h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[h/Y], \Rightarrow_\beta}}{\lambda z.\exists x.[person(x) \wedge hug(z,x)] : e_\sigma \multimap h_\sigma}{\scriptstyle \multimap_{\mathcal{I},2}}}{\forall y.[person(y) \to \exists x.[person(x) \wedge hug(y,x)] : h_\sigma}{\scriptstyle \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[h/X], \Rightarrow_\beta}$$

5.2 The Resource Sensitivity Hypothesis and its Consequences

- The Resource Sensitivity Hypothesis (RSH; Asudeh 2004, 2012) stems from the resource-logical perspective on semantic composition in Glue Semantics (among others, Dalrymple et al. 1993, Dalrymple 1999, 2001, Asudeh 2012), which uses the resource logic *linear logic* (Girard 1987) to assemble meanings.

  (20) **The Resource Sensitivity Hypothesis (RSH)**:
  Natural language is resource-sensitive.

- RSH is equivalent to the claim of Linguistic Resource Sensitivity, which is in turn derived from Logical Resource Sensitivity:

  (21) **Logical Resource Sensitivity**:
  In a resource logic, premises in proofs cannot be freely *reused* or *discarded*.

  (22) **Linguistic Resource Sensitivity**:
  Natural language is resource-sensitive: elements of combination in grammars cannot be freely *reused* or *discarded*.

- The upshot of RSH is that compositional semantics is constrained by resource accounting, such that component meanings cannot go unused or be reused.

- For example, in the following sentence, the adjective *alleged* contributes a single lexical meaning resource which cannot be used twice to derive the unavailable meaning that the individual in question is only allegedly a forger.

  (23) The alleged murderer is a forger.

- The Resource Sensitivity Hypothesis paves the way to substantial simplification, since the following independent principles can be reduced to resource sensitivity (Asudeh 2012: 110–123):

  1. Bounded Closure
  2. Completeness and Coherence
  3. The Theta Criterion
  4. The Projection Principle
  5. No Vacuous Quantification
  6. The Inclusiveness Condition
  7. Full Interpretation

- Not only does RSH set the ground for eliminating these principles from our theories, it also gives us a deeper understanding of the principles, since they are reduced to the basic combinatoric logic of semantic composition.

6 Analysis

6.1 Heim's Generalization

(24) **Generalization H**
Degree operators cannot scope over quantificational DPs. (Heim 2001)

(25) Some girl is taller than 6′.

Abridged $some > DEG$ **proof for (25):**

$$\dfrac{\begin{array}{l}\textbf{some girl}\\ \lambda P.\exists x.[girl(x) \wedge P(x)] :\\ \forall Y.(g \multimap Y) \multimap Y\end{array} \qquad \dfrac{\begin{array}{l}\textbf{tall}\\ \lambda d \lambda z.tall(z, d) :\\ d \multimap g \multimap t\end{array} \quad \begin{array}{l}\textbf{-er than 6}'\\ \lambda D \lambda y.max(\lambda d.D(d)(y)) > 6' :\\ \forall X.(d \multimap g \multimap X) \multimap (g \multimap X)\end{array}}{\lambda y.max(\lambda d.tall(y, d)) > 6' : g \multimap t}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[t/X], \Rightarrow_{\beta}}{\exists x.[girl(x) \wedge max(\lambda d.tall(x, d)) > 6'] : t}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[t/Y], \Rightarrow_{\beta}$$

No $DEG > some$ **proof for (25):**

$$\dfrac{\begin{array}{l}\textbf{-er than 6}'\\ \lambda D \lambda y.max(\lambda d.D(d)(y)) > 6' :\\ \forall X.(d \multimap g \multimap X) \multimap (g \multimap X)\end{array} \qquad \dfrac{\dfrac{\dfrac{\begin{array}{l}\textbf{tall}\\ \lambda d \lambda z.tall(z, d) :\\ d \multimap g \multimap t\end{array} \quad [\delta : d]^{1}}{\lambda z.tall(z, \delta) : g \multimap t}\multimap_{\mathcal{E}} \quad \begin{array}{l}\textbf{some girl}\\ \lambda P.\exists x.[girl(x) \wedge P(x)] :\\ \forall Y.(g \multimap Y) \multimap Y\end{array}}{\exists x.[girl(x) \wedge tall(x, \delta)] : t}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[t/Y], \Rightarrow_{\beta}}{\lambda \delta.\exists x.[girl(x) \wedge tall(x, \delta)] : d \multimap t}\multimap_{\mathcal{I},1}}{}\textbf{FAIL}$$

6.2 Kennedy's Generalization

(26) **Generalization K**
Degree operators can scope over intensional verbs. (Kennedy 1997)

(27) Miss America is required to be taller than 5′.

Abridged $\Box > \mathit{DEG}$ proof for (27):

$$\dfrac{\begin{array}{c}\textbf{required}\\ \lambda p.\Box p : t \multimap r\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\textbf{tall}\\ \lambda d \lambda z.\mathit{tall}(z,d) : \\ d \multimap m \multimap t\end{array} \quad \begin{array}{c}\textbf{-er than 5}'\\ \lambda D \lambda y.\mathit{max}(\lambda d.D(d)(y)) > 5' : \\ \forall X.(d \multimap m \multimap X) \multimap (m \multimap X)\end{array}}{\lambda y.\mathit{max}(\lambda d.\mathit{tall}(y,d)) > 5' : m \multimap t}\,{\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[t/X], \Rightarrow_\beta} \quad \begin{array}{c}\textbf{Miss A.}\\ \mathit{ma} : m\end{array}}{\mathit{max}(\lambda d.\mathit{tall}(\mathit{ma},d)) > 5' : t}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta}}{\Box \mathit{max}(\lambda d.\mathit{tall}(\mathit{ma},d)) > 5' : r}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta}$$

Abridged $\mathit{DEG} > \Box$ proof for (27):

$$\dfrac{\dfrac{\begin{array}{c}\textbf{-er than 5}'\\ \lambda D \lambda y.\mathit{max}(\lambda d.D(d)(y)) > 5' : \\ \forall X.(d \multimap m \multimap X) \multimap (m \multimap X)\end{array} \quad \dfrac{\dfrac{\dfrac{\begin{array}{c}\textbf{required}\\ \lambda p.\Box p : t \multimap r\end{array} \quad \dfrac{\dfrac{\begin{array}{c}\textbf{tall}\\ \lambda d \lambda z.\mathit{tall}(z,d) : \\ d \multimap m \multimap t\end{array} \quad [\delta : d]^1}{\lambda z.\mathit{tall}(z,\delta) : m \multimap t}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta} \quad [x : m]^2}{\mathit{tall}(x,\delta) : t}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta}}{\Box \mathit{tall}(x,\delta) : r}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta}}{\lambda x.\Box \mathit{tall}(x,\delta) : m \multimap r}\,{\multimap_{\mathcal{I},2}}}{\lambda \delta \lambda x.\Box \mathit{tall}(x,\delta) : d \multimap m \multimap r}\,{\multimap_{\mathcal{I},1}}}{\lambda y.\mathit{max}(\lambda d.\Box \mathit{tall}(y,d)) > 5' : m \multimap r}\,{\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[r/X], \Rightarrow_\beta} \quad \begin{array}{c}\textbf{Miss A.}\\ \mathit{ma} : m\end{array}}{\mathit{max}(\lambda d.\Box \mathit{tall}(\mathit{ma},d)) > 5' : r}\,{\multimap_{\mathcal{E}}, \Rightarrow_\beta}$$

6.3 Heim and Kennedy's Generalizations Together

(28) Every paper is required to be exactly ten pages long.

Abridged $\mathit{every} > \mathit{DEG} > \Box$ **proof for (28):**

$$\dfrac{\dfrac{\begin{matrix}\mathit{every}' :\\ \forall Y.(p \multimap Y) \multimap Y\end{matrix} \quad \dfrac{\begin{matrix}\mathit{deg}' :\\ \forall X.(d \multimap p \multimap X) \multimap (p \multimap X)\end{matrix} \quad \dfrac{\dfrac{\dfrac{\begin{matrix}\mathit{require}' :\\ e \multimap r\end{matrix} \quad \dfrac{\dfrac{\begin{matrix}\mathit{exactly}' :\\ d \multimap p \multimap e\end{matrix} \quad [d]^1}{p \multimap e} \quad [p]^2}{e}}{\mathit{require}'(\mathit{exactly}') : r}}{p \multimap r}\multimap_{\mathcal{I},2}}{d \multimap p \multimap r}\multimap_{\mathcal{I},1}}{\mathit{deg}'(\mathit{require}'(\mathit{exactly}')) : p \multimap r}\forall_{\mathcal{E}}[r/X]}{\mathit{every}'(\mathit{deg}'(\mathit{require}'(\mathit{exactly}'))) : r}\forall_{\mathcal{E}}[r/Y]}{\text{Every paper x is such that the length l of x is such that it is necessary that l is exactly ten pages.}}\text{ i.e.}$$

Abridged $\mathit{every} > \Box > \mathit{DEG}$ **proof for (28):**

$$\dfrac{\dfrac{\begin{matrix}\mathit{every}' :\\ \forall Y.(p \multimap Y) \multimap Y\end{matrix} \quad \dfrac{\dfrac{\begin{matrix}\mathit{require}' :\\ e \multimap r\end{matrix} \quad \dfrac{\dfrac{\begin{matrix}\mathit{deg}' :\\ \forall X.(d \multimap p \multimap X) \multimap (p \multimap X)\end{matrix} \quad \begin{matrix}\mathit{exactly}' :\\ d \multimap p \multimap e\end{matrix}}{\mathit{deg}'(\mathit{exactly}') : p \multimap e}\forall_{\mathcal{E}}[e/X] \quad [p]^1}{\mathit{deg}'(\mathit{exactly}') : e}}{\mathit{require}'(\mathit{deg}'(\mathit{exactly}')) : r}}{p \multimap r}\multimap_{\mathcal{I},1}}{\mathit{every}'(\mathit{require}'(\mathit{deg}'(\mathit{exactly}'))) : r}\forall_{\mathcal{E}}[r/Y]}{\text{Every paper x is such that it is necessary that the length l of x is exactly ten pages.}}\text{ i.e.}$$

Abridged $\Box > \mathit{every} > \mathit{DEG}$ **proof for (28):**

$$\dfrac{\dfrac{\begin{matrix}\mathit{require}' :\\ e \multimap r\end{matrix} \quad \dfrac{\begin{matrix}\mathit{every}' :\\ \forall Y.(p \multimap Y) \multimap Y\end{matrix} \quad \dfrac{\begin{matrix}\mathit{deg}' :\\ \forall X.(d \multimap p \multimap X) \multimap (p \multimap X)\end{matrix} \quad \begin{matrix}\mathit{exactly}' :\\ d \multimap p \multimap e\end{matrix}}{\mathit{deg}'(\mathit{exactly}') : p \multimap e}\forall_{\mathcal{E}}[e/X]}{\mathit{every}'(\mathit{deg}'(\mathit{exactly}')) : e}\forall_{\mathcal{E}}[e/Y]}{\mathit{require}'(\mathit{every}'(\mathit{deg}'(\mathit{exactly}'))) : r}}{\text{It is necessary that every paper x is such that the length l of x is exactly ten pages.}}\text{ i.e.}$$

The following scopings are impossible, given that the system captures Generalization H ($\mathit{DEG} \not> Q$***).***

$*\ \mathit{DEG} > \mathit{QUANT} > \mathit{Verb}$

$*\ \mathit{DEG} > \mathit{Verb} > \mathit{QUANT}$

$*\ \mathit{Verb} > \mathit{DEG} > \mathit{QUANT}$

You can try deriving the readings in this section for yourself using Gianluca Giorgolo's theorem prover at `http://llilab.carleton.ca/~giorgolo/tp.html`. Enter the relevant sequent below for each of the three examples considered here (enter each all on one line):

(29) Some girl is taller than 6′.

```
tall : d.d -> g.e -> t.t,
deg : (d.d -> g.e -> X.t) -> (g.e -> X.t),
some : (g.e -> Y.t) -> Y.t
=> t.t
```

(30) Miss America is required to be taller than 5′.

```
tall : d.d -> m.e -> t.t,
deg : (d.d -> m.e -> X.t) -> (m.e -> X.t),
miss :  m.e,
require : t.t -> r.t
=> r.t
```

(31) Every paper is required to be exactly ten pages long.

```
exactly : d.d -> p.e -> e.t,
deg : (d.d -> p.e -> X.t) -> (p.e -> X.t),
every : (p.e -> Y.t) -> Y.t,
require : e.t -> r.t
=> r.t
```

7 Some Questions

1. Is $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$ a reasonable type for degree operators?

   The answer depends on one's assumptions about morphology and the relationship between morphology and syntax. If one assumes that, e.g., *taller*, is directly made up of the parts *tall* and *er* and the type of *tall* is $\langle d, \langle e, t\rangle\rangle$, then the *only* directly compositional type for *er* is one that takes $\langle d, \langle e, t\rangle\rangle$ as its input.[1]

   In other words, if one assumes strong lexicalism, the degree operator cannot directly have type $\langle\langle d, t\rangle, t\rangle$, because that input type depends on *tall* having found its entity argument before combining with the degree operator, contrary to its morphosyntax. Thus, to the extent that the type $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$ derives Heim and Kennedy's generalizations (given reasonable types for the other expressions), and strong lexicalism and compositionality require this type, then Heim and Kennedy's generalizations are theorems of that view of grammar.

2. Does the approach hold any prospect for the analysis of split scope as in (1), repeated here?

   (32) The company need fire no employees.

   Heim (2001: 225; emphasis in original):
   "What I would like to suggest instead is that scope-splitting (at least sometimes) *is* DegP-movement."

   In our terms, is there a plausible type assignment for (32) that involves assigning the type $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$ to (some part of) the object?

[1] Of course, the comparative actually takes two inputs, so the fully compositional type for the comparative morpheme would be a function from the type of measure phrases to the type of degree operators, $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$.

A plausible split-scope paraphrase of (32) is:
"It is not the case that it is necessary that the company fire some number of employees."

It is natural to treat numbers as a scale, like degrees. We then need to treat the (putative) degree operator *no employees* as somehow contributing three components:

(a) A function from the set of individuals fired by the company to the number of fired employees
$\lambda P \lambda n \lambda x.|P(x) \wedge employee(x)| = n : (e \multimap f) \multimap (d \multimap e \multimap f)$

(b) A degree modifier of the kind we have already seen, modulo number instead of degrees, which existentially quantifies over the number of fired employees
$\lambda N \lambda y.\exists n[N(n)(y)] : \forall X.(d \multimap e \multimap X) \multimap (e \multimap X)$

(c) A negative operator
$\lambda P.no(P) : \forall Y.(e \multimap Y) \multimap Y$

We then get three proofs for (32).[2] The first two proofs are simply sketches of the combinatorics, but the split scope proof on the following page has more details.

Abridged de dicto(?) $\Box > NEG > DEG$ proof for (32):

$$\dfrac{\Box : f \multimap n \qquad \dfrac{\begin{matrix} no' : \\ \forall Y.(e \multimap Y) \multimap Y \end{matrix} \qquad \dfrac{\begin{matrix} deg' : \\ \forall X.(d \multimap e \multimap X) \multimap (e \multimap X) \end{matrix} \qquad \dfrac{\begin{matrix} num.emp' : \\ (e \multimap f) \multimap (d \multimap e \multimap f) \end{matrix} \qquad \dfrac{\begin{matrix} comp' : \\ c \end{matrix} \quad \begin{matrix} fire' : \\ c \multimap e \multimap f \end{matrix}}{fire'(comp') : e \multimap f}}{num.emp'(fire'(comp')) : (d \multimap e \multimap f)}}{deg'(num.emp'(fire'(comp'))) : e \multimap f}\,\forall_{\mathcal{E}}[f/X]}{no'(deg'(num.emp'(fire'(comp')))) : f}\,\forall_{\mathcal{E}}[f/Y]}{\Box no'(deg'(num.emp'(fire'(comp')))) : n}$$

Abridged de re(?) $NEG > DEG > \Box$ proof for (32):

$$\dfrac{\begin{matrix} no' : \\ \forall Y.(e \multimap Y) \multimap Y \end{matrix} \qquad \dfrac{\begin{matrix} deg' : \\ \forall X.(d \multimap e \multimap X) \multimap (e \multimap X) \end{matrix} \qquad \dfrac{\dfrac{\dfrac{\Box : f \multimap n \qquad \dfrac{\dfrac{\dfrac{\begin{matrix} num.emp' : \\ (e \multimap f) \multimap (d \multimap e \multimap f) \end{matrix} \qquad \dfrac{\begin{matrix} comp' : \\ c \end{matrix} \quad \begin{matrix} fire' : \\ c \multimap e \multimap f \end{matrix}}{fire'(comp') : e \multimap f}}{num.emp'(fire'(comp')) : (d \multimap e \multimap f)} \qquad [d]^1}{num.emp'(fire'(comp')) : (e \multimap f)} \qquad [e]^2}{num.emp'(fire'(comp')) : f}}{\Box num.emp'(fire'(comp')) : n}}{\Box num.emp'(fire'(comp')) : e \multimap n}\,\multimap_{\mathcal{I},2}}{\Box num.emp'(fire'(comp')) : d \multimap e \multimap n}\,\multimap_{\mathcal{I},1}}{deg'(\Box num.emp'(fire'(comp'))) : (e \multimap n)}\,\forall_{\mathcal{E}}[n/X]}{no'(deg'(\Box num.emp'(fire'(comp')))) : n}\,\forall_{\mathcal{E}}[n/Y]$$

[2] In order to save space, I write c_ι for $\iota x.[company(x)]$.

Abridged split scope $NEG > \Box > DEG$ proof for (32):

$$
\dfrac{
\begin{array}{l}\textbf{no employees.3}\\ \lambda P.no(P) :\\ \forall Y.(e \multimap Y) \multimap Y\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\textbf{need}\\ \lambda p.\Box p :\\ f \multimap n\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\textbf{no employees.2}\\ \lambda N \lambda y.\exists n[N(n)(y)] :\\ \forall X.(d \multimap e \multimap X) \multimap (e \multimap X)\end{array}
\quad
\dfrac{
\begin{array}{l}\textbf{no employees.1}\\ \lambda P \lambda n \lambda x.|P(x) \wedge employee(x)| = n :\\ (e \multimap f) \multimap (d \multimap e \multimap f)\end{array}
\quad
\dfrac{
\begin{array}{l}\textbf{the company}\\ c_\iota :\\ c\end{array}
\quad
\begin{array}{l}\textbf{fire}\\ \lambda x \lambda y.\mathit{fire}(x,y) :\\ c \multimap e \multimap f\end{array}
}{\lambda y.\mathit{fire}(c_\iota, y) : e \multimap f}
}{\lambda n \lambda x.|\mathit{fire}(c_\iota, x) \wedge employee(x)| = n : (d \multimap e \multimap f)}
}{\lambda y.\exists n[|\mathit{fire}(c_\iota, y) \wedge employee(y)| = n] : (e \multimap f)}\ \forall_{\mathcal{E}}[f/X]
\quad
[x : e]^1
}{\exists n[|\mathit{fire}(c_\iota, x) \wedge employee(x)| = n] : f}
}{\Box\exists n[|\mathit{fire}(c_\iota, x) \wedge employee(x)| = n] : n}
}{\lambda x.\Box\exists n[|\mathit{fire}(c_\iota, x) \wedge employee(x)| = n] : e \multimap n}\ \multimap_{\mathcal{I},1}
}{no(\lambda x.\Box\exists n[|\mathit{fire}(c_\iota, x) \wedge employee(x)| = n]) : n}
$$

3. What prospects does this analysis hold for explaining Generalizations A–D from Abels and Martí (2010)?

   Generalization A
   Split scope readings are possible only across intensional verbs.

   To the extent that split scope is related to the degree operator facts, this is jointly derived from Generalizations H and K, which we have derived.

   Generalization B
   Split scope readings are possible only across some intensional verbs, not all.

   Heim (2001: 226) encourages us to pursue a line in which in fact these readings are in principle available (i.e, the split scoping exists), but "there is something in [the intensional verbs'] semantics that masks the truth-conditional effect of this movement".

   If this proves untenable and there really is some kind of grammatical constraint, then the best option for this system seems to be some kind of generalization of the scope trapping/freezing behaviour, such that the intensional verbs themselves introduce a scope point that prevents a proper proof for split scope.

   Generalization C
   Split scope readings are not possible in the context of extraposition (even when the intensional verb falls within the class of verbs that allow split scope).

   Again, if this is a grammatical constraint, then an option for this system seems to be some kind of generalization of the scope trapping/freezing behaviour, such that extraposition itself introduces a (limited?) scope point such that it traps split scope within its scope.

   Generalization D
   Split scope readings always involve low existential force.

   If only negative indefinites result in split scope, it follows that the low remainder will have existential force if the negation separately takes wide scope. So is this just a lexical fact?

8 Conclusion

- I have proposed a modificational type of $\langle\langle d, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$ for degree phrases.

- They are nonetheless scopal operators, because they introduce a scope point in their input type and a corresponding scope point in their output type.

- The modificational type and the general mechanisms of Glue Semantics derive that they must scope under quantificational DPs while allowing them to scope over intensional verbs.

- The system thus derives Generalizations H & K.

- Much work remains to be done in order to generalize this to a unified theory of split scope, as in Abels and Martí (2010).

References

Abels, Klaus, and Luisa Martí. 2010. A Unified Approach to Split Scope. *Natural Language Semantics* 18: 435–470.

Asudeh, Ash. 2004. Resumption as Resource Management. Ph.D. thesis, Stanford University.

—. 2005. Relational Nouns, Pronouns, and Resumption. *Linguistics and Philosophy* 28(4): 375–446.

—. 2006. Direct Compositionality and the Architecture of LFG. In Miriam Butt, Mary Dalrymple, and Tracy Holloway King, eds., *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, 363–387. Stanford, CA: CSLI Publications.

—. 2012. *The Logic of Pronominal Resumption*. Oxford: Oxford University Press.

Asudeh, Ash, and Richard Crouch. 2002. Glue Semantics for HPSG. In Frank van Eynde, Lars Hellan, and Dorothee Beermann, eds., *Proceedings of the 8th International HPSG Conference*, 1–19. Stanford, CA: CSLI Publications.

Asudeh, Ash, and Ida Toivonen. 2009. Lexical-Functional Grammar. In Bernd Heine and Heiko Narrog, eds., *The Oxford Handbook of Linguistic Analysis*, 425–458. Oxford: Oxford University Press.

Carpenter, Bob. 1997. *Type-Logical Semantics*. Cambridge, MA: MIT Press.

Dalrymple, Mary, ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Dalrymple, Mary, Vaneet Gupta, John Lamping, and Vijay Saraswat. 1999a. Relating Resource-Based Semantics to Categorial Semantics. In Dalrymple 1999, 261–280.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat. 1999b. Overview and Introduction. In Dalrymple 1999, 1–38.

Dalrymple, Mary, John Lamping, and Vijay Saraswat. 1993. LFG Semantics via Constraints. In *Proceedings of the Sixth Meeting of the European ACL*, 97–105. European Chapter of the Association for Computational Linguistics, University of Utrecht.

Girard, Jean-Yves. 1987. Linear Logic. *Theoretical Computer Science* 50(1): 1–102.

Gotham, Matthew. 2015. Towards Glue Semantics for Minimalist Syntax. Presented at *Interactions between syntax and semantics across frameworks*, Cambridge University. Slides: http://www.ucl.ac.uk/ ucjtmgg/docs/cam-slides.pdf.

Halvorsen, Per-Kristian, and Ronald M. Kaplan. 1988. Projections and Semantic Description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 1116–1122. Institute for New Generation Systems, Tokyo. Reprinted in Dalrymple et al. (1995: 279–292).

Heim, Irene. 2001. Degree Operators and Scope. In Caroline Féry and Wolfgang Sternefeld, eds., *Audiatur Vox Sapientiae: A Festschrift for Arnim von Stechow*, 214–239. Berlin: Akademie Verlag.

Jacobs, Joachim. 1980. Lexical Decomposition in Montague Semantics. *Theoretical Linguistics* 7: 121–136.

Jäger, Gerhard. 2005. *Anaphora and Type Logical Grammar*. Dordrecht: Springer.

Kaplan, Ronald M. 1987. Three Seductions of Computational Psycholinguistics. In Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod Johnson, and Paul Bennett, eds., *Linguistic Theory and Computer Applications*, 149–181. London: Academic Press. Reprinted in Dalrymple et al. (1995: 339–367).

—. 1989. The Formal Architecture of Lexical-Functional Grammar. In Chu-Ren Huang and Keh-Jiann Chen, eds., *Proceedings of ROCLING II*, 3–18. Reprinted in Dalrymple et al. (1995: 7–27).

Kaplan, Ronald M., and Joan Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Joan Bresnan, ed., *The Mental Representation of Grammatical Relations*, 173–281. Cambridge, MA: MIT Press. Reprinted in Dalrymple et al. (1995: 29–135).

Kennedy, Christopher. 1997. Antecedent Contained Deletion and the Syntax of Quantification. *Linguistic Inquiry* 28: 662–688.

Kokkonidis, Miltiadis. 2008. First-Order Glue. *Journal of Logic, Language and Information* 17(1): 43–68.

Lev, Iddo. 2007. Packed Computation of Exact Meaning Representations. Ph.D. thesis, Stanford University.

Moortgat, Michael. 1997. Categorial Type Logics. In Johan van Benthem and Alice ter Meulen, eds., *Handbook of Logic and Language*, 93–177. Cambridge, MA: MIT Press. Co-published with Elsevier Science B.V., Amsterdam.

Morrill, Glyn V. 1994. *Type Logical Grammar*. Dordrecht: Kluwer.

Potts, Chris. 2000. When Even *No*'s Neg is Splitsville. In Sandy Chung, Jim McCloskey, and Nathan Sanders, eds., *Jorge Hankamer Webfest*. Department of Linguistics, UCSC. Http://babel.ucsc.edu/Jorge/.

Rullmann, Hotze. 1995. Maximality in the Semantics of Wh-Constructions. Ph.D. thesis, University of Massachusetts, Amherst.

de Swart, Henriëtte. 2000. Scope Ambiguities with Negative Quantifiers. In *Reference and Anaphoric Relations*, 109–132. Dordrecht: Kluwer.