Lecture 3. Detailed Derivations using GL's Type Composition Logic

- Type Clashes
- Coercions
- Types for Change Predicates

Classic GL Treatment of Dot Objects

---

(123) a. Mary believes that John is sick.

b. Mary believes the story.

c. Mary believes John.

The coercion operations projecting one type from the complex type are projection operators, defined as Σ_1 and Σ_2. These two operations, together with the dot object itself form the definition of the type cluster called a lexical conceptual paradigm (lcp).

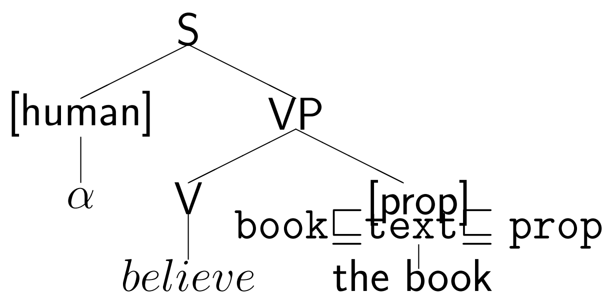
(124) $\texttt{lcp} = \{\sigma_1 \cdot \sigma_2, \Sigma_1[\sigma_1 \cdot \sigma_2] : \sigma_1, \Sigma_2[\sigma_1 \cdot \sigma_2] : \sigma_2\}$

(125)a. Σ_1[$\texttt{info}\cdot\texttt{physobj}$]:$\texttt{info}$

b. Σ_2[$\texttt{info}\cdot\texttt{physobj}$]:$\texttt{physobj}$

c. $\texttt{info}\cdot\texttt{physobj\_lcp} = \{\texttt{info}\cdot\texttt{physobj}$, $\texttt{info}$, $\texttt{physobj}\}$

$\texttt{book} \sqsubseteq \texttt{text} \sqsubseteq \texttt{prop}$



(126)

$$\frac{\Sigma_1(\texttt{info}\cdot\texttt{physobj}) : \texttt{info} \quad , \quad \Theta[\texttt{info} \sqsubseteq \texttt{prop}] : \texttt{info} \rightarrow \texttt{prop}}{\Theta[\texttt{info} \sqsubseteq \texttt{prop}](\Sigma_1(\texttt{info}\cdot\texttt{physobj})) : \texttt{prop}}$$

(127)a. Mary believes the book.

b. $\mathbf{believe}(\hat{}\Theta(\Sigma_1(\mathbf{the\text{-}book})))(\mathbf{Mary}) \Rightarrow$

c. $\mathbf{believe'}(\hat{}\Theta(\mathbf{the\text{-}book}\!:\!\mathtt{info}))(\mathbf{Mary}) \Rightarrow$

d. $\mathbf{believe'}(\hat{}\mathbf{the\text{-}book}\!:\!\mathtt{prop})(\mathbf{Mary})$

(128) Mary sold the book to John.

(129)a. Mary sold the book to John.

b. $\mathbf{sell}(\mathbf{John})(\Theta(\Sigma_2(\mathbf{the\text{-}book})))(\mathbf{Mary}) \Rightarrow$

c. $\mathbf{sell}(\mathbf{John})(\Theta(\mathbf{the\text{-}book}\!:\!\mathtt{physobj}))(\mathbf{Mary}) \Rightarrow$

d. $\mathbf{sell}(\mathbf{John})(\mathbf{the\text{-}book}\!:\!\mathtt{ind})(\mathbf{Mary})$

Reference to different Aspects of Dot Objects

(130)a. John read every book in the library.

b. John stole every book in the library.

(131)a. Mary answered every question in the class.

b. Mary repeated every question in the class.

Object-Elaboration (O-Elab). 1

- $\lambda y \lambda x[\text{O-Elab}(x, y)]$
- Predication involving the simple aspect of adot object is an object elaboration of the dot object.
- E.g., For a book of type $p \bullet i$, we can pick out or elaborate one aspect of the variable $v : p \bullet i$ with a variable of either dot element (component type).

Object-Elaboration (O-Elab). 2

- •-types are just mereological sums of their aspects; hence they are idempotent, associative and commutative;
- O-Elab is an antisymmetric and transitive proper-part-of relation;
- Assume in addition that x is of type σ and y is of type σ and we have O-elab(z, x) and O-elab(z, y), then $x = y$; i.e. parts of an object singled out for predication that are of the same aspect are identical.

Type Clash Motivates Coercion

A type clash between two constituents A and B occurs whenever:

- if A is a function that is supposed to apply to B, then the glb of the type τ of the λ-abstracted variable in A and the type of B is $\bot$, or
- if B is a function that is supposed to apply to A, then the glb of the type τ of λ-abstracted variable in B and the type of A is $\bot$.

Example: A and B will have a type clash, when A is $\lambda x F x$ with $x : p$ and B is y with $y : i$, where $p \sqcap i = \bot$.

Resolving the Type Clash

---

(132) Head Typing Principle:
Given an environment X with constituents A and B, and type assignments $A\colon \alpha$ and $B\colon \beta$ in clashing type contexts for A and B respectively, if A is the syntactic head of X, then typing of A must be preserved in any composition rule for A and B to produce a type for X.

Example of Type Clash

(133) a heavy book

(134) $\lambda P \lambda x[\text{heavy}(x) \wedge P(x)]$

x: p;

P : $p \multimap t$;

heavy : $(p \multimap t) \multimap (p \multimap t)$.

(135) $\lambda v \text{book}(v)$: $(p \bullet i) \multimap t$

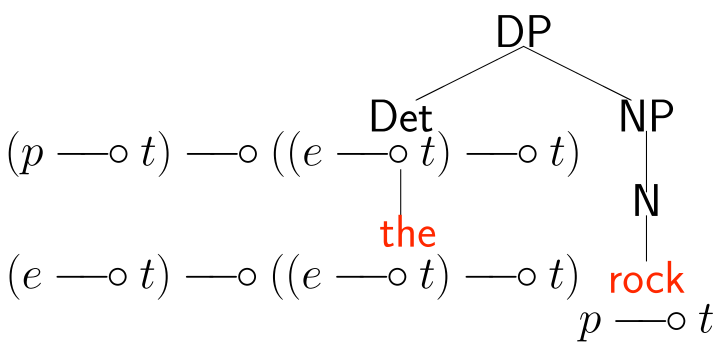
Resolving the Type Clash

(136) Head Typing Principle:
Given an environment X with constituents A and B, and type assignments $A\colon \alpha$ and $B\colon \beta$ in clashing type contexts for A and B respectively, if A is the syntactic head of X, then typing of A must be preserved in any composition rule for A and B to produce a type for X.

(137) $\lambda P \lambda x \exists z[\text{heavy}(z) \wedge O\text{-}elab(z, x) \wedge P(x)]$
where $z\colon p$, $x\colon p \bullet i$ and $P\colon (p \bullet i) \multimap t$.

Derivation using Accommodation

(138) The rock is heavy.



(139) $(e \multimap t) \multimap ((e \multimap t) \multimap t) \sqcap p \multimap t =$

the: $(p \multimap t) \multimap ((e \multimap t) \multimap t)$

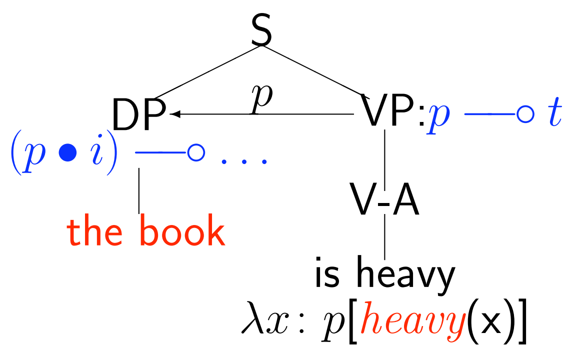
(140)

$$\frac{\lambda x \phi[t], c(x \colon \alpha, t \colon \beta), \;\; \alpha \sqcap \beta \neq \bot}{\lambda x \phi[t], c * (x, t \colon \alpha \sqcap \beta)}$$

Derivation involving Coercions: Exploitation

(141) The book is heavy.

(142)



Step-by-Step Derivation: 1

1. $[\![\text{the}]\!] = \lambda Q \lambda P \exists x(Q[x] \wedge P[x]), \langle P, Q \colon e \multimap t, x \colon e \rangle$
2. $[\![\text{book}]\!] = \lambda v \mathsf{book}(v), \langle v \colon p \bullet i \rangle$
3. $[\![\text{the book}]\!] = \lambda Q \lambda P \exists x(Q[x] \wedge P[x]),$
   $\langle P, Q \colon e \multimap \underline{t}, x \colon e \rangle [\lambda v \mathsf{book}(v), \langle v \colon p \bullet i \rangle]$

   As $e \sqcap (p \bullet i) = p \bullet i$, by Accommodation, which revises the typing context, we get:
   $\lambda Q \lambda P \exists x(Q[x] \wedge P[x]),$
   $\langle P, Q \colon (p \bullet i) \multimap \underline{t}, x \colon p \bullet i \rangle [\lambda v \mathsf{book}(v), \langle v \colon p \bullet i \rangle]$

Step-by-Step Derivation: 2

5. Now we use Application and Merging Contexts to get:
6. $[\![$the book$]\!] =$
   $\lambda P \exists x(\text{book}(x) \wedge P[x]), \langle x : p \bullet i,\ P : (p \bullet i) \multimap \underline{t}\rangle$
7. $[\![$is heavy$]\!] = \lambda u \text{heavy}(u),\ \ \langle u : p\rangle$
8. The syntax dictates:
   $\lambda P \exists x(\text{book}(x) \wedge P[x]),$
   $\langle P : e \multimap \underline{t}, x : p \bullet i\rangle[\lambda u \text{heavy}(u),\ \ \langle u : p\rangle]$
9. By $\bullet$-Exploitation:
   $\{\lambda P \exists x(\exists v(\text{book}(v) \wedge \text{O-Elab}(x, v)) \wedge P[x])$
   $\langle v : p \bullet i, x : p\rangle\}[\lambda u \text{heavy}(u),\ \ \langle u : p\rangle]$

Step-by-Step Derivation: 3

10. By Merging Contexts and Application:
    $\exists x(\exists v(\text{book}(v) \wedge \text{O-Elab}(x,v)) \wedge \lambda u \text{heavy}(u)[x]), \langle x\colon p,\ u\colon p,\ v\colon p \bullet i\rangle$

11. By Application:
    $\exists x(\exists v(\text{book}(v) \wedge \text{O-Elab}(x,v)) \wedge \text{heavy}(x)), \quad \langle v\colon p \bullet i, x\colon p\rangle$

    $\Longrightarrow$ The book is heavy.

Coercion by •-Exploitation

(143)

$$\frac{\{\lambda P \phi(P(x)),\ c(P\colon (\alpha \bullet \beta) \multimap \gamma)\}[\psi,\ c'(\psi\colon \begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} \multimap \gamma)],}{\{\lambda P \phi[\frac{\exists v(\Delta(\phi,x)[\frac{v}{x}] \wedge \textsf{O-Elab}(x,v))}{\Delta(\phi,x)}],\ c*(x\colon \begin{bmatrix}\alpha \sqcap \alpha' \\ \beta \sqcap \beta'\end{bmatrix}, v\colon \alpha \bullet \beta)\}[\psi, c']}$$

Coercion by Type Shifting •-Exploitation. 1

(144) John's mother burned his book on magic before he fully understood it.

(145) $\lambda x[\text{burn}(x)] \ : \ p \multimap (e \multimap t)$

$\lambda x[\text{understand}(x)] \ : \ i \multimap (human \multimap t)$

$\lambda v \text{book}(v) \ : (p \bullet i) \multimap t$

- The verb *burn*'s object argument must be a physical object, and as the Head Typing Principle dictates, although the object DP enters the composition with type $p \bullet i$, there must be some way to coerce it into having the right type, to satisfy the typing context and thereby allow the λ-conversion from the verb to go through.
- To do this, we apply a kind of $\bullet$-Exploitation on the generalized quantifier to coerce it into the right type.
- $\lambda\mathcal{P}\lambda w\mathcal{P}[\lambda u(\mathsf{burn}(w,u))], \langle\mathcal{P}\colon (p \multimap \underline{t}) \multimap \underline{t}, u\colon p, w\colon p\rangle[\lambda P\exists x(\mathsf{book}(x) \wedge P(x)),\ \langle P\colon (p \bullet i) \multimap \underline{t}, x\colon p \bullet i\rangle]$

Coercion by Type Shifting •-Exploitation. 2

$$\frac{\{\lambda\mathcal{P}\phi,\ c(\mathcal{P}\colon (\begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} \multimap \gamma) \multimap \delta)\}[\lambda P\psi(P[x]),\ c'(P\colon (\alpha\bullet\beta) \multimap \gamma)]}{\{\lambda\mathcal{P}\phi,\ c\}[\lambda P\psi\{\frac{\exists v(\Delta(\psi,x)\{\frac{v}{x}\}\wedge \textsf{O-Elab}(x,v))}{\Delta(\psi,x)}\},\ c'*(v\colon \alpha\bullet\beta, x\colon \begin{bmatrix}\alpha\sqcap\alpha' \\ \beta\sqcap\beta'\end{bmatrix})]}$$

Type-Shift •-Exploitation applies

1. $\lambda\mathcal{P}\lambda w\mathcal{P}[\lambda u(\text{ burn}(w,u))], \langle\mathcal{P}\colon (p \multimap \underline{t}) \multimap \underline{t}, u\colon p, w\colon p\rangle\ [\lambda P\exists x(\exists v(\text{book}(v) \wedge \text{O-Elab}(x,v)) \wedge P[x]),\ \langle P\colon p \multimap \underline{t}, x\colon p, v\colon p \bullet i\rangle]$
2. Applying *Merging* and *Application*, we get the following expression:
3. $\lambda w\ \lambda P\ \exists x(\exists v(\text{book}(v) \wedge \text{O-Elab}(x,v) \wedge P[v]))[\lambda u(\text{ burn}(w,u))], \langle P\colon p \multimap \underline{t}, x\colon p, v\colon p \bullet i, u\colon p, w\colon p\rangle$
4. $\lambda w\exists x\exists v(\text{book}(v) \wedge \text{O-Elab}(x,v)) \wedge \text{burn}(w,v),\ \langle w\colon p, x\colon p, v\colon p \bullet i\rangle]$

Derivation involving Coercions: Introduction

1. Mary read the book.
2. John read the rumor about his ex-wife.
3. Mary read the subway wall.

Example Walk-through of $\bullet$-Introduction

(146) a heavy book

(147) $\lambda P \lambda x[\text{heavy}(x) \wedge P(x)]$

x: p;

$P : p \multimap t$;

heavy : $(p \multimap t) \multimap (p \multimap t)$.

(148) $\lambda v \text{book}(v)$: $(p \bullet i) \multimap t$

Resolving the Type Clash

(149) Head Typing Principle:

Given an environment X with constituents A and B, and type assignments $A: \alpha$ and $B: \beta$ in clashing type contexts for A and B respectively, if A is the syntactic head of X, then typing of A must be preserved in any composition rule for A and B to produce a type for X.

(150) $\lambda P \lambda x \exists z[\textsf{heavy}(z) \wedge O\text{-}elab(z, x) \wedge P(x)]$

where $z: p$, $x: p \bullet i$ and $P: (p \bullet i) \multimap t$.

Rule of •-Introduction

$$\frac{\{\lambda P \phi(P[x]), c(P\colon \begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} \multimap \gamma)\}[\psi,\ c'(\psi\colon (\alpha \bullet \beta) \multimap \gamma)], \mathsf{head}(\psi)}{\{\lambda P \phi[\frac{\exists v(\Delta(\phi,x)[\frac{v}{x}] \wedge \mathsf{O\text{-}Elab}(v,x))}{\Delta(\phi,x)}], c * (v\colon \begin{bmatrix}\alpha \sqcap \alpha' \\ \beta \sqcap \beta'\end{bmatrix}, x : \alpha \bullet \beta)\}\ [\psi,\ c']}$$

Step-by-Step Derivation: 1

1. $[\![\text{a}]\!] = \lambda Q \lambda P \exists x (Q[x] \wedge P[x]), \langle P, Q : e \multimap t, x : e \rangle$
2. $[\![\text{book}]\!] = \lambda v \text{book}(v), \langle v : p \bullet i \rangle$
3. $[\![\text{heavy}]\!] = \lambda P \lambda x [\text{heavy}(x) \wedge P(x)]$
4. $\bullet$-Introduction applies to the Adjective Phrase:
5. $[\![\text{heavy}]\!] = \lambda P \lambda x \exists z [\text{heavy}(z) \wedge O\text{-}elab(z, x) \wedge P(x)]$
   where $z : p$, $x : p \bullet i$ and $P : (p \bullet i) \multimap t$.
6. This now combines with the head noun *book*:
   $\lambda x \exists z ((\text{Heavy}(z) \wedge \text{O-elab}(z, x) \wedge \text{Book}(x))$
7. $[\![\text{heavy book}]\!] =$
   $\lambda x \exists z ((\text{Heavy}(z) \wedge \text{O-elab}(z, x) \wedge \text{Book}(x))$

8. $[\![$a heavy book$]\!] =$
$\lambda P \lambda x \exists z((\mathsf{Heavy}(z) \wedge \mathsf{O\text{-}elab}(z, x) \wedge \mathsf{Book}(x) \wedge P[x]),$
$z : p$, $x : p \bullet i$ and $P : (p \bullet i) \multimap t$.

Rule of Type Shift •-Introduction

$$\frac{\{\lambda\mathcal{P}\phi, c(\mathcal{P}\colon ((\alpha \bullet \beta) \multimap \gamma) \multimap \delta)\ [\lambda P\psi(P[x]), c'(P\colon \begin{bmatrix}\alpha\\ \beta\end{bmatrix} \multimap \gamma)]}{\{\lambda\mathcal{P}\phi,\}\ [\lambda P\psi\{\frac{\exists v(\Delta(\psi,x)\{\frac{v}{x}\}\wedge \textsf{O-Elab}(v,x))}{\Delta(\psi,x)}\}, c' * (x\colon \alpha\bullet\beta, v\colon \begin{bmatrix}\alpha \sqcap \alpha'\\ \beta\sqcap\beta'\end{bmatrix})}$$

Example Walk-through of Type Shift •-Introduction

- John read every wall.

The rule of •-I^{TS} transforms the logical form for the DP *every wall* into:

$$\lambda P\ \forall x[\exists v[\mathsf{wall}(v) \wedge \mathsf{O\text{-}elab}(v, x)] \rightarrow P(x)],$$
$$\langle x : p \bullet i,\ v : p\rangle$$

Exploitation of Qualia ($\otimes$). 1

1. begin a cigarette (i.e., smoking)
2. enjoy the book (i.e., reading)
3. enjoy the sonata (i.e., listening or playing)
4. finish the coffee (i.e., drinking)
5. finish the house (i.e., building)

Exploitation of Qualia ($\otimes$). 2

- If σ and $\tau_1, \cdots \tau_n$ are types, then so is $(\sigma \otimes_{R_1,\ldots R_n} (\tau_1 \cdots \tau_n))$.
- $\textsc{phys} \otimes_{\mathsf{Telic}} \textsc{smoke}$
- $p \bullet i \otimes_{\mathsf{A,T}} (\textsc{write}, \textsc{read})$

Example Walk-through of Qualia $\otimes$-Exploitation

- Zac enjoyed the book.

1. *enjoy*:
   $\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v\textsf{enjoy}(u,v)), \langle\mathcal{P}\colon (\textsc{event} \multimap \underline{t}) \multimap \underline{t}, v\colon \textsc{event}, u\colon \textsc{agent}\rangle$
2. *the book*:
   $\lambda\exists x(\textsf{Book}(x) \wedge P[x]),\ \langle x\colon (p \bullet i) \otimes_{A,T} (\textsc{write},\ \textsc{read})\rangle$
3. putting the two together:
   $\{\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v\textsf{enjoy}(u,v)),$

$\langle\mathcal{P}\colon (\textsc{event} \multimap \underline{t}) \multimap$

$\underline{t}, v\colon \textsc{event}, u\colon \textsc{agent}\rangle\}[\lambda\exists x(\mathsf{Book}(x) \wedge P[x]),\ \langle x\colon (p \bullet i) \otimes_{A,T} (\textsc{write, read})\rangle]$

We'll assume that *enjoy* prefers the telic of its object. So QC^{TS} exploitation on the telic role will give us:

1. $\{\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v\mathsf{enjoy}(u,v)), \langle\mathcal{P}\colon (\textsc{event} \multimap \underline{t}) \multimap \underline{t}, v\colon \textsc{event}, u\colon \textsc{agent}\rangle\}$

   $[\lambda P\exists e\exists x(\mathsf{book}(x) \wedge (\mathsf{Telic}(x,e) \wedge P[e]), \langle x\colon (p \bullet i) \otimes (\textsc{write, read}), u\colon \textsc{event},\ e\colon \textsc{read}\rangle].$

2. Now by Accommodation,

   $\{\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v\mathsf{enjoy}(u,v)), \langle\mathcal{P}\colon (\textsc{event} \multimap \underline{t}) \multimap \underline{t}, v\colon \textsc{event} \sqcap \textsc{read}, u\colon \textsc{agent}\rangle\}$

   $[\lambda P\exists e\exists x(\mathsf{book}(x) \wedge (\mathsf{Telic}(x,e) \wedge P[e]), \langle x\colon (p \bullet i) \otimes (\textsc{write, read}), u\colon \textsc{event},\ e\colon \textsc{read}\rangle]$

3. Now by Application:
$\lambda u \exists e \exists x(\mathsf{book}(x) \wedge (\mathsf{Telic}(x, e) \wedge \mathsf{enjoy}(u, e)),$
$\langle x\colon (p \bullet i) \otimes (\textsc{write}, \textsc{read}),\ e\colon \textsc{read}\rangle$