Lecture 2. Generative Typing

- Levels and Type Construction
- Type Composition Logic
- Mechanisms of Selection

GL's Theoretical Starting Points

- The human conceptual apparatus (i.e. the ability to categorize and represent the world) is one of generative categorization and compositional thought (as opposed to extensional).
- The human linguistic capacity reflects our ability to categorize and represent the world in the particular ways we do.
- Therefore, language is a natural manifestation of our generative construction of the world through the categories it employs.

Level of Types in the Major Categories

1. Noun

   N: rock, water, woman, tiger, tree

   F: knife, beer, husband, dancer

   C: book, lunch, university, temperature

2. Verb

   N: fall, walk, rain, put, have

   F: donate, spoil, quench

   C: read, perform

3. Adjective

N: red, large, flat

F: useful, good, effective

C: rising

Mechanisms of Selection

- Pure Selection: The type a function requires is directly satisfied by the argument.
- Exploitation: The type a function requires is inherited by the argument.
- Coercion: The type a function requires is wrapped around the argument, embedding it within the required type.

GL Lexical Structure

(83)

$$\begin{bmatrix} \alpha \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = x \\ \ldots \end{bmatrix} \\ \text{EVENTSTR} = \begin{bmatrix} \text{EVENT1} = e1 \\ \text{EVENT2} = e2 \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} = \textbf{what } x \textbf{ is made of} \\ \text{FORMAL} = \textbf{what } x \textbf{ is} \\ \text{TELIC} = e_2\textbf{: function of } x \\ \text{AGENTIVE} = e_1\textbf{: how } x \textbf{ came into being} \end{bmatrix} \end{bmatrix}$$

Type Composition Logic:
Pustejovsky (2001), Asher and Pustejovsky (1998, 2005)

The Type Language

(84)a. e the general type of entities; t the type of truth values.

(σ, τ range over all simple types, and subtypes of e.)

b. If σ and τ are types, then so is $(\sigma \rightarrow \tau)$

c. If σ and τ are types, then so is $(\sigma \bullet \tau)$

d. If σ and $\tau_1, \cdots \tau_n$ are types, then so is $(\sigma \otimes_{R_1,\ldots R_n} (\tau_1 \cdots \tau_n))$.

Qualia Structure as Types

(85) TYPE FEATURE STRUCTURE:

$$\begin{bmatrix} x: \alpha \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} : \beta \\ \text{FORMAL} : \alpha \\ \text{TELIC} : \tau \\ \text{AGENTIVE} : \sigma \end{bmatrix} \end{bmatrix}$$

Qualia as Types

(86)
$$\begin{bmatrix} x: & \alpha \\ & \otimes\beta \\ & \otimes\tau \\ & \otimes\sigma \end{bmatrix}$$

Natural Entities

Entities formed from the application of the FORMAL and/or CONST qualia roles:

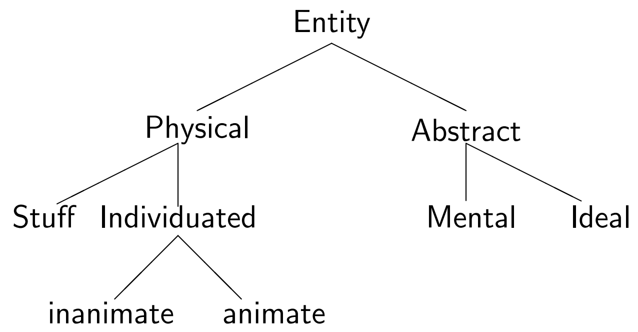
For the predicates below, e_N is structured as a join semi-lattice, $\langle e_N, \sqsubseteq \rangle$;

(87)a. *physical*, *human*, *stick*, *lion*, *pebble*

b. *water*, *sky*, *rock*

Natural Entity Types as a Lattice

(88)



Natural Predicate Types

Predicates formed with Natural Entities as arguments:

(89)a. *fall*: $e_N \rightarrow t$

b. *touch*: $e_N \rightarrow (e_N \rightarrow t)$

c. *be under*: $e_N \rightarrow (e_N \rightarrow t)$

Expressed as typed arguments in a λ-expression:

(90)a. $\lambda x : e_N[\mathit{fall}(\mathsf{x})]$

b. $\lambda y : e_N \lambda x : e_N[\mathit{touch}(\mathsf{x,y})]$

c. $\lambda y : e_N \lambda x : e_N[\mathit{be\text{-}under}(\mathsf{x,y})]$

Functional Entity Types: e_F

Entities formed from the Naturals by adding the AGENTIVE or TELIC qualia roles:

(91) Expressed as types:

a. Functional Entity: $x : e_N \otimes T$

b. Functional Predicate: $P : e_N \otimes T \rightarrow t$

Functional Entity Types

Examples of types in e_F.

(92)a. *beer*: $liquid \otimes drink$

b. *knife*: $phys \otimes cut$

c. *house*: $phys \otimes live\_in$

(93) TELIC and AGENTIVE constraints on the Natural Type HUMAN:

a. *boss*, *friend*;

b. *dancer*, *husband*, *president*;

Functional Predicate Types

Predicates formed with Functional Entities as arguments:

(94)a. *spoil*: $e_N \otimes T \rightarrow t$

b. *fix*: $e_N \otimes T \rightarrow (e_N \rightarrow t)$

Expressed as typed arguments in a λ-expression:

(95)a. $\lambda x: e_F[$*spoil*(x)$]$

b. $\lambda y: e_F \lambda x: e_N[$*fix*(x,y)$]$

(96)a. The beer spoiled.

b. Mary fixed the watch.

Corpus Data on spoil (Patrick Hanks, p.c.)

In both BNC and Associated Press, over 80% of Direct Objects of spoil are Events. Typically, they are Events that one would expect to enjoy. The implicature is that, by spoiling an Event, one kills the enjoyability of it. One might say that spoil is a causative antonym of enjoy.

The lexical set of significant direct objects of spoil (from Waspbench, augmented by CPA) include:

fun, enjoyment, magic, pleasure, holiday, party, Christmas, birthday, dinner, evening, morning, day, half-hour, event, occasion, view, performance, opera, game, match, ...

Complex Entity Types

Entities formed from the Naturals and Functionals by reifying a specific relation between the entities, i.e., the dot, •.

(97) a. Mary doesn't believe the book.

b. John bought his book from Mary.

c. The police burnt a controversial book.

(98) a. John wrote the exam last night in under 10 minutes.

b. The exam lasted more than three hours this morning.

Dot Objects: e_C

(99) Expressed as types:

a. Complex Entity: $x : e_i \bullet e_j$

b. Complex Predicate: $P : x : e_i \bullet e_j \rightarrow t$

Introduce a coherence relation between (at least) two natural or functional types, and reify that as a type.

(100)a. PHYS•INFO: *book*, *DVD*;

b. EVENT•EVENT: *construction*, *examination*;

c. PHYS•APERTURE: *door*, *window*.

Complex Predicate Types

Predicates formed with Complex Entity Types as arguments:

(101) $read$: $phys \bullet info \rightarrow (e_N \rightarrow t)$

Expressed as typed arguments in a λ-expression:

(102) $\lambda y: phys \bullet info \; \lambda x: e_N[read(\text{x,y})]$

(103) Mary read the book.

Maintaining Compositionality

- Generative Mechanisms of Argument Selection:
  - Selection
  - Accommodation
  - Coercion:
    (i) Introduction
    (ii) Exploitation
- Qualia-based Type Structure:
  - Natural,
  - Functional,
  - Complex.

Classic GL Coercion

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

b. Mary believes the story.

c. Mary believes John.

Generative Mechanisms of Argument Selection

- Pure Selection: The type a function requires is directly satisfied by the argument.
- Accommodation: The type a function requires is inherited by the argument.
- Coercion: The type a function requires is imposed on the argument type. This is accomplished by either:
  - Exploitation: selecting part of the argument's type structure to satisfy the function's typing;
  - Introduction: wrapping the argument with the type the function requires.

Pure Selection as Application

(105)

$$\frac{\Gamma \vdash \alpha \rightarrow \beta, \quad \Gamma \vdash \alpha}{\Gamma \vdash \beta}$$

(106)

$$\frac{\lambda x \phi[t], \quad c(x : \alpha, t : \alpha)}{\phi[t/x], c}$$

Accommodation of an Argument

(107)

$$\frac{\Gamma \vdash \alpha \rightarrow \beta, \quad \Gamma \vdash \gamma, \alpha \sqcap \gamma \neq \bot}{\Gamma' \vdash \alpha \sqcap \gamma \rightarrow \beta}$$

(108)

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

Merging Contexts:

$$\frac{\{\lambda x \phi, c\}[t, c']}{\lambda x \phi[t], (c + c')}$$

Type Coercion

- Exploitation: selecting part of the argument's type structure to satisfy the function's typing;
- Introduction: wrapping the argument with the type the function requires.

Head Typing Principle

---

(109) Given a compositional environment X with constituents A and B, and type assignments $A\colon \alpha$ and $B\colon \beta$ in the type contexts for A and B respectively that clash, if A is the syntactic head in the environment, then the typing of A must be preserved in any composition rule for A and B to produce a type for X.

Coercion of an Argument: Exploitation of •

(110)

$$\frac{\Gamma \vdash \alpha \to \beta, \quad \Gamma \vdash \gamma, \gamma' \bullet \alpha = \gamma}{\Gamma', \gamma' \bullet \alpha, \vdash \alpha \to \beta}$$

(111)

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

Coercion of an Argument: Introduction of •

(112)

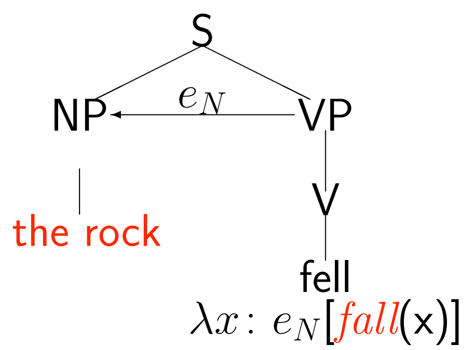
$$\frac{\Gamma \vdash \alpha \bullet \gamma \rightarrow \beta, \quad \Gamma \vdash \gamma}{\Gamma', \vdash \alpha \bullet \gamma \rightarrow \beta, \gamma \bullet \alpha = \gamma'}$$

(113)

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

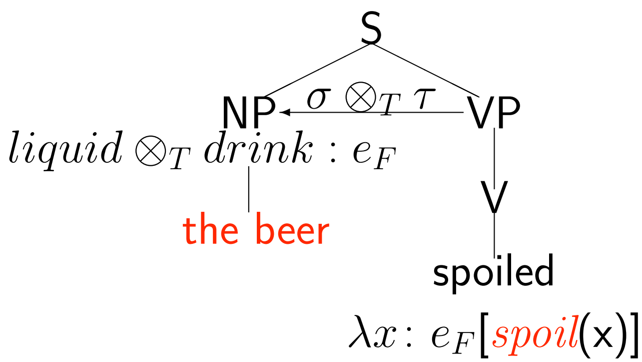
Pure Selection: Natural Type

(114)



Pure Selection: Functional Type

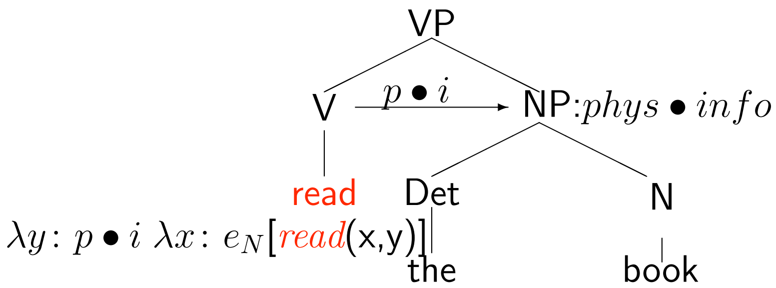
(115)



$$liquid \otimes_T drink \sqsubseteq \sigma \otimes_T \tau$$

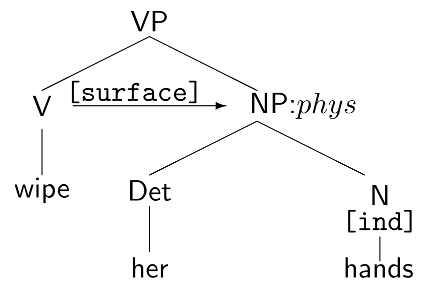
Pure Selection: Complex Type

(116)



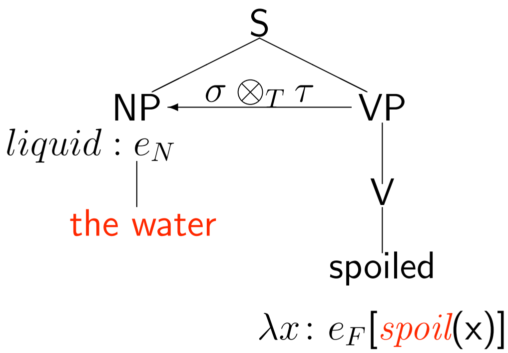
Type Accommodation: Natural

(117)



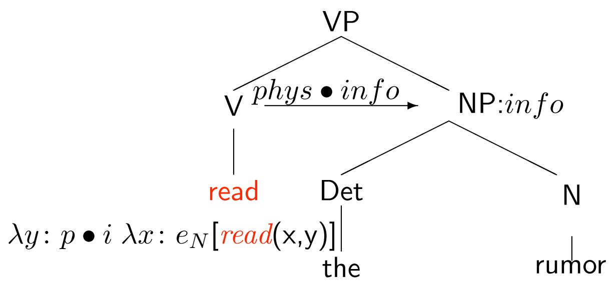
Type Coercion: Natural to Functional Introduction

(118)



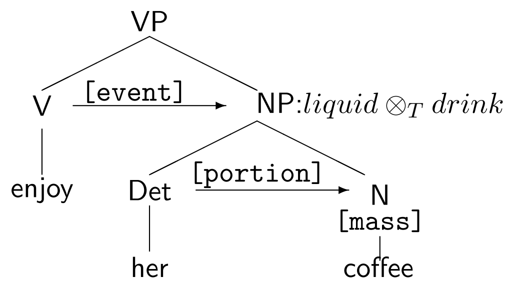
Type Coercion: Natural to Complex Introduction

(119)



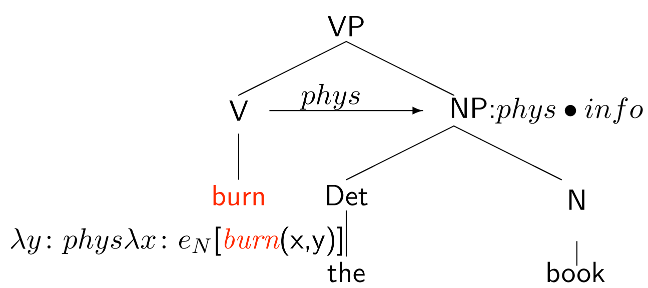
Type Coercion: Functional Exploitation

(120)



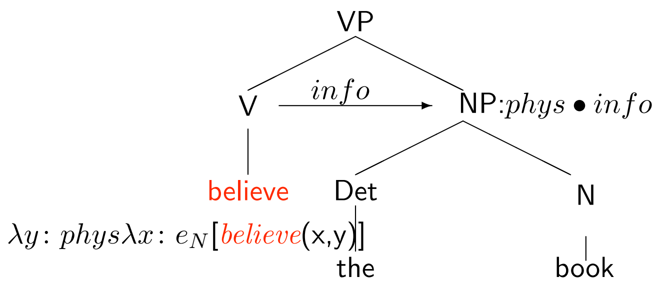
Type Coercion: Complex Exploitation

(121)



Type Coercion: Complex Exploitation

(122)



Types and Composition of Local Contexts

Compositionality mediated through richer selectional mechanisms:

		TYPE	
CONTEXT	Natural	Functional	Complex
Selection	die(x)	fix(x,y)	read(x,y)
Accommodation	wipe(x,hand)	spill(beer)	steal(x,book)
Coercion	begin(rock)	spoil(water)	read(x,joke)