

Generative Lexicon Theory: Integrating Theoretical and Empirical Methods

James Pustejovsky
Brandeis University

Elisabetta Ježek
University of Pavia

July 11-15, 2016
NASSLLI 2016
Rutgers University



- July 11: Introduction to GL and Data Analytics
- July 12: Qualia Structure
- July 13: Event Structure
- July 14: Argument Structure
- July 15: Meaning Composition

Introduction to Generative Lexicon

- Basic concepts in GL
 - Motivation
 - Notation and Language: typed feature structures
 - Meaning Composition in GL
- Polysemy and the Lexicon-Pragmatics Interface
- Evidence-based linguistics and data analytics

Qualia Structure

- What is a Quale?
- What motivates Qualia?
- Default Qualia and context updating
- Methodology to identify Qualia
- Data for each Quale
- Qualia and Conventionalized Attributes
- Qualia for Verbs

Lab on Qualia identification and extraction

Event Structure

- Events as Structured Objects
- Event Types
 - States
 - Transitions
 - Point Verbs
 - Processes
- Events as Labeled Transition Systems
- Dynamic Event Models

Lab on identification of event types

Argument Structure

- Argument Types in GL
 - True Arguments
 - Shadow Arguments
 - Hidden Arguments
- Argument Structure Representation
- Arguments and Defaulting

Lab on argument typing and coercion

Meaning composition

- Basic Assumptions
- Simple Function Application
- Coercion
- Subselection
- Co-composition

Argument Structure

- Argument Types in GL
 - True Arguments
 - Shadow Arguments
 - Hidden Arguments
- Arguments and Defaulting
- Dynamic Argument Structure
- Basic Mechanisms for Strong Compositionality

Lab on argument typing and coercion

Arguments in GL

- In accordance with standard theories of argument structure, GL assumes that among the entities that take part in an event (so-called **participants**), some are selected by the corresponding predicate in the language as **arguments**, while other are not (the **adjuncts**).
- Arguments **saturate** (i.e. complete, specify) the meaning of the predicate, and are part of its Qualia , while adjuncts provide supplementary (optional) information.

Non-completeness constraint in GL

- An important feature in GL argument theory is that there is no requirement that for every argument of the relation expressed by a verb there is a corresponding position in the surface syntactic structure containing the verb.
- Arguments **may remain unexpressed** under certain conditions.

Argument types in GL

- We will consider an inventory of GL argument types which includes **three basic types**:
 - **True** Arguments.
 - **Default** Arguments.
 - **Shadow** Arguments.
 - **Hidden** Arguments.

True Arguments

- True Arguments are event participants that are selected by the predicate as **parts of its meaning**, and must be expressed syntactically, otherwise the predicate cannot be interpreted because its meaning remains incomplete.
- True arguments do not express temporal properties of the event itself, such as presence or absence of change (**event structure**) nor its ontological type, such as motion, perception, communication (**qualia structure**), at least not directly.
- True arguments are **informative** with respect to these properties.

- After she locked **the front door**, she went to bed.
- *After she locked, she went to bed.
- *I forgot to lock.
- *Did you lock?
- *What is she doing? She is locking.

Default Arguments

- **Default Arguments:** Arguments which participate in the logical expressions in the qualia, but which are not necessarily expressed syntactically;
- John built the house out of bricks.

Shadow Arguments

- A Shadow Argument is an event participant which is **semantically incorporated** (i.e. shadowed) in the meaning of the verb (i.e. in its *qualia*).
- As such, it is obligatorily left out of the surface structure, because it is redundant (i.e. it repeats information) from the point of view of its contribution to the interpretation of the predicate.
- She phoned the office.
- *She phoned the office on the phone.

Hidden Arguments

- *Hidden Arguments* (Badia and Sauri 2001, Jezek and Melloni 2011, Pustejovsky and Jezek 2011) are an addition to the standard 1995 model.
- A Hidden Argument encodes a participant that cannot be expressed in the syntax under any condition, but **plays a role in the interpretation of the sentences** in which the predicate appears.
- John photographed the whole scene.

Hidden arguments and anaphora

- Mary translated the Italian poem in two weeks. **It** is a beautiful piece of work.
- I copied your passport for the office, but **it** is too blurred to use.
- Cathie sliced the bread. We each got **one**.

Argument Structure Representations in GL

In a GL lexical entry, argument types are directly **encoded** in the representation of argument structure, together with their **semantic type**.

$$\left[\begin{array}{l} \textit{lock.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \textit{human} \\ \text{ARG2} = \textit{phys_obj} \\ \text{S-ARG1} = \textit{lock} \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \textit{phone.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \textit{human} \\ \text{ARG2} = \textit{human} \\ \text{S-ARG1} = \textit{phone} \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \textit{photograph.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \textit{human} \\ \text{ARG2} = \textit{entity} \\ \text{H-ARG1} = \textit{photograph} \end{array} \right] \end{array} \right]$$

- The type of an argument is a value selected from **an inventory of types** in the language (Asher and Pustejovsky 2006, Pustejovsky 2011).
- In addition to the Montague types, *e* and *t*, GL assumes a richer subtyping over the entity domain than is typically assumed in type theory.
- Among these we find: *human*, *physical_object*, *artifact*, *material*, *substance*, *information*, *location*.

GL Types for Composition

ABSTRACT ENTITY, ANIMATE, ARTIFACT, ATTITUDE,
DOCUMENT, DRINK, EMOTION, ENTITY, EVENT,
FOOD, HUMAN, HUMAN GROUP, IDEA, INFORMATION,
LOCATION, OBLIGATION, ORGANIZATION, PATH,
PHYSICAL OBJECT, PROPERTY, PROPOSITION, RULE,
SENSATION, SOUND, SUBSTANCE, TIME PERIOD,
VEHICLE

- (1) a. The spokesman denied the statement (PROPOSITION).
- b. The child threw the stone (PHYSICAL OBJECT).
- c. The audience didn't believe the rumor (PROPOSITION).

Coercion using Types

- (2) a. The president denied the attack (EVENT → PROPOSITION).
- b. The White House (LOCATION → HUMAN) denied this statement.
- c. The Boston office called with an update (EVENT → INFO).

Semantic types and co-composition

The semantic types of the arguments contribute to the assignment of a specific interpretation to the selecting verb in the context of use, i.e. they **co-determine** verb meaning.

$$\left[\begin{array}{l} \textit{devour.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \textit{animate} \\ \text{ARG2} = \textit{food} \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \textit{devour.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \textit{human} \\ \text{ARG2} = \textit{document} \end{array} \right] \end{array} \right]$$

Semantic types and co-composition

- Complements are regarded as **arguments** if they **contribute functionally to the way the verb is interpreted in the context of use**; otherwise, they are **adjuncts** (Jezek et al. 2014).
- This principle distinguishes a **typed predicate-argument structure** (T-PAS, cf. Jezek et al. 2014) from a **pattern of use** of the verb (CPA, Pustejovsky and Hanks, 2005).

Typed predicate-argument structure:

$$\left[\begin{array}{l} die.v \\ ARGSTR = \left[ARG1 = animate \right] \end{array} \right]$$

Corpus-derived pattern of use (from PDEV, Hanks 2013):

[[*Human*]] die ([[Adv[Manner]] — in infancy — childbirth)
([Adv[Location]] — in [[Location]] — at [[Location]])
([Adv[Manner]] — of [[Eventuality]] — from [[Eventuality]])
([Adv[Time]] — at [[Number = age]]) ([Adv[Time]] — in
[[Number = year of death]]) ([Adv[Manner]] — at birth)
([Adv[Time]] — on [[Time Point = date]]) ([Adv[Time]] — at
[[Time Point = Time of day]]) ([Adv[Manner]] — in
[[Eventuality]]) ([Adv[Time]] — last day — week — ...))

Thematic Roles and GL

- Thematic role constraints on arguments, such as *Agent*, *Patient*, *Theme*, *Experiencer*, *Instrument*, etc. are not included in the standard GL argument structure representation.
- The view taken in standard GL is that argument selection is a **typing mechanism**.
- In Pustejovsky and Ježek 2012, we show how information on participants **roles** is complementary to typing information and relevant in distinguishing selectional properties of verbs, and may be integrated in the model.

Shadow as Instrument

- Shadow Instrument is a type of *artifact*.
- bike to the lake shore. (move by S-arg)
- Shadow Instrument is a type of *substance*.
- glue two pieces of wood together. (put together using S-arg)

Shadow as Theme

- Shadow Theme is a type of *artifact*.
- dress the kids before breakfast. (put **S-arg** on the kids)
- Shadow Theme is a type of *substance*
- butter the toast. (put **S-arg** on toast, cover toast with **S-arg**)

Shadows as types playing specific roles

$$\left[\begin{array}{l} \textit{bike.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \dots \\ \text{S-ARG:instrument} = \textit{bike} \end{array} \right] \end{array} \right]$$
$$\left[\begin{array}{l} \textit{butter.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{ARG1} = \dots \\ \text{S-ARG:theme} = \textit{butter} \end{array} \right] \end{array} \right]$$

Arguments and Defaulting

- Optionality of arguments is accounted for in GL through the mechanisms of **defaulting**.
- **Defaulting** allows an argument to **freely appear or not appear** depending on specific conditions. (Pustejovsky and Jezek 2012).
- An argument which has been defaulted is called a **Defaulted Argument**, i.e. D-ARG.
- There are two main types of **Defaulted arguments** (Jezek 2016):
 - **Pragmatically Defaulted Arguments** (PD-ARG);
 - **Lexically Defaulted Arguments** (LD-ARG).

Pragmatically Defaulted Arguments (PD-ARG)

- PD-ARGs are omissions of an argument which denotes an **instance** of the class selected by the predicate (i.e. the semantic type), which is salient in the situational context.
- The predicates they are arguments of force us **to grab something from the context to saturate their meaning**.

Examples of PD-ARG

- Marc decided to take the left trail and we followed (...) behind.
- When he found out (about ...), it was too late.
- John started (...) at 8.30 and finished (...) at 5pm.
- She tried (to ...) but did not succeed (in ...).

Lexically Defaulted Arguments (LD-ARG)

- A Lexically Defaulted Argument (LD-ARG) is an event participant that is lexically selected as an argument by the predicate but can left unexpressed when it is to be generically understood as the **class** of entities selected by the predicate.
- Saturation of a LD-ARG is not a contextual matter, i.e. there is not need to retrieve or construct a specific discourse referent.

Examples of LD-ARG

- John ate at 5pm.
- John read in the car.
- John parked ten yards after the last house.
- *John listened in the car.

Difference between PD- and LD-ARG

- With LD-ARG there is no need to retrieve or construct the particular thing that e.g. *x ate* and *parked*.
- The unexpressed object is understood generically as the class of entities (*food, information, vehicle*) selected by the predicate.
- Only at this condition (i.e. the condition that the object is understood generically as a class, and the focus is on the action being performed), can the argument be lexically defaulted.

Difference between PD- and LD-ARG

- If a specific instance of the class needs to be mentioned (a pizza, a letter, the BMW etc.), lexical defaulting cannot apply, unless the specific instance can be reconstructed pragmatically, in which case, however, the correct interpretation of the omission is **pragmatically** and not *lexically* defaulted argument.
- Mary prepared John's favorite dish and he ate too much (of it).

Defaulted Arguments

Defaulted Locations

- My plane arrived late. (where?)
- John left before breakfast. (from where?)

Defaulted Materials or Substances

- John built a house. (with what?)

Defaulted Instruments (Searle 1980)

- Mary cut the grass (with what?)

Constraints on Pragmatic Defaulting

- They were ordered to abandon the room.
- *They were ordered to abandon.
- They were ordered to leave (the room).

Lexically Defaulted Arguments and Predictability

- Lexical defaults encode a participant which class is **predictable** from the meaning of the verb.
- Predictability does not seem to be a sufficient condition.
- *garment* is predictable as object of *wear* but cannot be left out.

Defaulting and Shadows

- When a **specialization** of a shadow argument is made in discourse by way of adding new information, the shadow argument is extracted from the predicate and expressed in the form of a defaulted argument.
- “She phones the office on the mobile phone.”

Dynamic Argument Structure

- Dynamic Argument Structure (DAS) is a systems of representation of argument structure, which **keeps track** of how the arguments to an event **contribute to the change** being expressed.
- It has been developed in the context of representing dynamically **creation** and **destruction** events as encoded in predicates.
- Jezek and Pustejovsky 2016.

Dynamic Argument Structure

- DAS identifies the following **four properties** for each argument to a predicate:
 - the **mode** of change it undergoes;
 - its semantic **role**;
 - its semantic **type**;
 - and whether it is **syntactically** realized.

We assume three primary **modes of change**:

- **Modification**: an argument undergoes *modification* if there is an identifiable attribute whose value is changed during the event.
- **Initiation**: an argument undergoes *initiation* if it is brought into existence as predicated by the verb,
- **Termination**: an argument undergoes *termination* if the converse is true.
- **Merge**: when two objects are brought together to bring about a new object, we say that they are *merged*.

Semantic Roles for Creation and Destruction Predicates

- **Resource**: the material or object used to bring about the change of state or result.
- **Result**: the outcome of the change of state brought about by the event.
- **Source**: used for Dowty's representational Source.
- **Medium**: the resource used by an animate object acting intentionally in the event bringing about a representational artifact.

Dynamic Program Variable

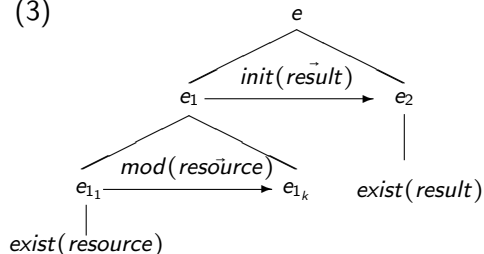
- DAS may encode one or more dynamic program variable.
- A **program variable** is a syntactically covert variable that **keeps track of the current state of what has been “acted on”**.
- It is annotated with a vector, \vec{x} .
- As it is never expressed, it is encoded as a hidden argument in GL representation format.
- Pustejovsky and Moszkowicz (2011), Pustejovsky and Jezek (2011), Pustejovsky (2013).

Dynamic Argument Structure for *build*

$$\left[\begin{array}{l} \textit{build.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{D-ARG:resource} = \textit{exist(phys)} \\ \text{H-ARG1:resource} = \textit{mod(phys)} \\ \text{H-ARG2:result} = \textit{init(phys)} \\ \text{T-ARG:result} = \textit{exist(artifact)} \end{array} \right] \end{array} \right]$$

Event Structure with Dynamic Argument annotation for *build*

(3)

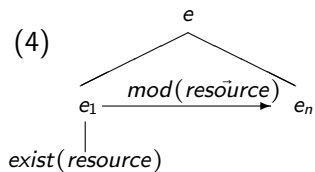


Dynamic Argument Structure for *knit*

$$\left[\begin{array}{l} \textit{knit.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{H-ARG:resource} = \textit{exist(phys)} \\ \text{H-ARG1:resour̄ce} = \textit{mod(phys)} \end{array} \right] \end{array} \right]$$

Event Structure with Dynamic Argument annotation for *knit*

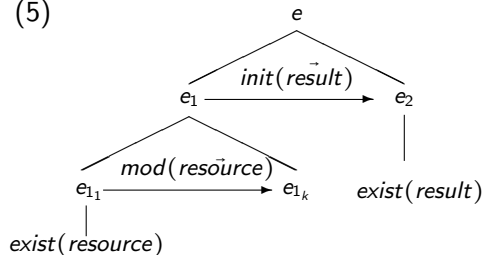
Mary knitted yesterday evening.



Event Structure with Dynamic Argument annotation for *knit*

Mary knitted a sweater last night.

(5)

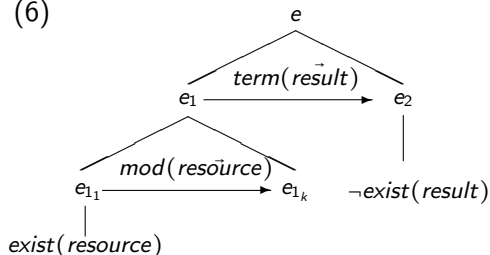


Dynamic Argument Structure for *destroy*

$$\left[\begin{array}{l} \textit{destroy.v} \\ \text{ARGSTR} = \left[\begin{array}{l} \text{T-ARG:resource} = \textit{exist(phys)} \\ \text{H-ARG1:res\ddot{o}urce} = \textit{mod(phys)} \\ \text{H-ARG2:res\ddot{u}lt} = \textit{term(phys)} \\ \text{H-ARG3:result} = \neg\textit{exist(phys)} \end{array} \right] \end{array} \right]$$

Event Structure with Dynamic Argument annotation for *destroy*

(6)

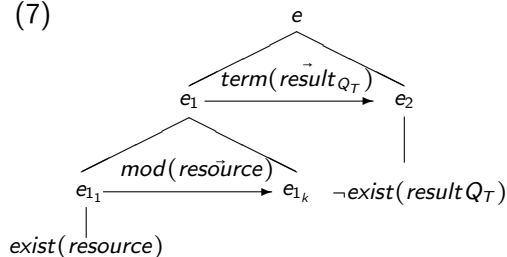


Event Structure with Dynamic Argument annotation for *break*

Frost broke the rock into smaller pieces.

(cf *The earthquake destroyed Mary's house into pieces.)

(7)



Conventional Compositionality

- (a) $BasTyp \subseteq T$
- (b) If $\sigma, \tau \in T$, then $\sigma \rightarrow \tau \in T$.

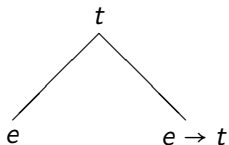
Following Montague, $BasTyp$ is:

$$BasTyp = \{Ind, Bool\} \text{ or } \{e, t\}$$

where e is the type of individuals and t is the type of propositions.

- (a) Function Application: If α is of type a , and β is of type $a \rightarrow b$, then $\beta(\alpha)$ is of type b .

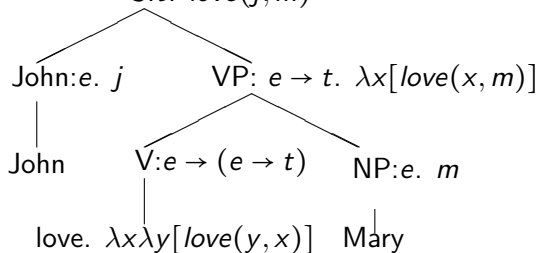
Conventional Compositionality



- (i) walk: $e \rightarrow t$
- (ii) walk slowly: $e \rightarrow t$
- (iii) eat fish: $e \rightarrow t$
- (iv) tell Mary where to buy her house: $e \rightarrow t$

Conventional Compositionality

(a) John loves Mary. $S:t. \text{love}(j, m)$



Sense Enumerative Lexicon (SEL)

1. Different senses of a word are separate lexical entries
2. Different senses behave differently in composition
1. Lexical entry encodes both syntactic and semantic properties
semantics

love: $\text{love}(\theta_1, \theta_2)$

θ_1 : HUMAN, θ_2 : HUMAN

1. John loves Mary.
2. Apply $\text{love}(\theta_1, \theta_2)$ to Mary
3. $\implies \text{love}(\theta_1, \text{Mary})$
4. Apply $\text{love}(\theta_1, \text{Mary})$ to John
5. $\implies \text{love}(\text{John}, \text{Mary})$

Lexical ambiguity is handled through separate lexical entries 1/2

1. **bake** (change-of-state): $\text{bake}(\theta_1, \theta_2)$
2. **bake** (create) : $\text{bake}(\theta_1, \theta_3)$

1. John **baked** a potato.
2. Apply $\text{bake}(\theta_1, \theta_2)$ to a potato
3. $\implies \text{bake}(\theta_1, \text{a_potato})$
4. Apply $\text{bake}(\theta_1, \text{a_potato})$ to John
5. $\implies \text{bake}(\text{John}, \text{a_potato})$

Lexical ambiguity is handled through separate lexical entries 2/2

1. **bake** (change-of-state): $\text{bake}(\theta_1, \theta_2)$
2. **bake** (create) : $\text{bake}(\theta_1, \theta_3)$

1. John **baked** a cake.
2. Apply $\text{bake}(\theta_1, \theta_3)$ to a cake
3. $\implies \text{bake}(\theta_1, \text{a_cake})$
4. Apply $\text{bake}(\theta_1, \text{a_cake})$ to John
5. $\implies \text{bake}(\text{John}, \text{a_cake})$

GL provides **richer typing** for arguments:

1. Identifies specific predicates in the body of the expression that are **characteristic functions of an argument**;
2. pulls this subset of predicates out of the body, and creates a *pretest* to the expression as a **restricted quantification over a domain of sorts**, denoted by that set of predicates.

Types from Predicative Content

$$\lambda x_2 \lambda x_1 [\Phi_1, \dots, \overbrace{\Phi_{x_1}}^{\tau}, \dots, \overbrace{\Phi_{x_2}}^{\sigma}, \dots, \Phi_k]$$

$$\lambda x_2 : \sigma \lambda x_1 : \tau [\Phi_1, \dots, \Phi_k - \{\Phi_{x_1}, \Phi_{x_2}\}]$$

σ and τ have now become **reified** as types on the arguments.

A Flexible Strategy of Selection

Arguments can be viewed as encoding **pretests** for performing the action in the predicate.

If the **argument condition** (i.e., its type) is not satisfied, the predicate either:

- **fails** to be interpreted (strong selection);
- **coerces** its argument according to a given set of strategies.

- (8) a. **LEXICAL TYPING STRUCTURE**: giving an explicit type for a word positioned within a type system for the language;
- b. **ARGUMENT STRUCTURE**: specifying the number and nature of the arguments to a predicate;
- c. **EVENT STRUCTURE**: defining the event type of the expression and any subeventual structure it may have;
- d. **QUALIA STRUCTURE**: a structural differentiation of the predicative force for a lexical item.

- (9) a. FORMAL: the basic category of which distinguishes the meaning of a word within a larger domain;
- b. CONSTITUTIVE: the relation between an object and its constituent parts;
- c. TELIC: the purpose or function of the object, if there is one;
- d. AGENTIVE: the factors involved in the object's origins or "coming into being" .

Arguments and Body in GL

$$\begin{array}{c} \text{Args} \\ \overbrace{\underbrace{\text{AS}}_{\lambda x_n \dots \lambda x_1} \underbrace{\text{ES}}_{\lambda e_m \dots \lambda e_1}} \\ \text{Body} \\ \overbrace{[Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4; C]} \end{array}$$

AS: Argument Structure

ES: Event Structure

Q_i : Qualia Structure

C: Constraints

Strong Compositionality

If all you have for composition is **function application**, then you need to create as many **lexical entries** for an expression as there are **environments** it appears in. (**Weak Compositionality**)

Two ways to overcome this:

1. **Type Shifting Rules**: Geach rule, Rooth and Partee (1982), Partee (1987), Groenendijk and Stokhof (1989).
2. **Type Coercion Operations**: Moens and Steedman (1988), Pustejovsky (1989), Jacobson (1992), Dölling (1992), Copestake and Briscoe (1992), Hendriks (1993), Egg (1994), Ramsey (1996), de Swart (1998).

- (10) a. **PURE SELECTION** (Type Matching): the type a function requires is directly satisfied by the argument;
- b. **ACCOMMODATION**: the type a function requires is inherited by the argument;
- c. **TYPE COERCION**: the type a function requires is imposed on the argument type. This is accomplished by either:
- i. **Exploitation**: taking a part of the argument's type to satisfy the function;
 - ii. **Introduction**: wrapping the argument with the type required by the function.

Two Kinds of Coercion in Language

- **Domain-shifting**: The domain of interpretation of the argument is shifted;
- **Domain-preserving**: The argument is coerced but remains within the general domain of interpretation.

Domain-Shifting Coercion

1. Entity shifts to event:
I enjoyed the beer
2. Entity shifts to proposition:
I doubt John.

Domain-Preserving Coercion

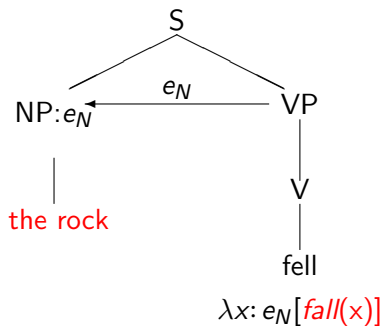
1. **Count-mass shifting**: There's chicken in the soup.
2. **NP Raising**: Mary and every child came.
3. **Natural-Artifactual shifting**: The water spoiled.
4. **Natural-Complex shifting**: She read a rumor.
5. **Complex-Natural shifting**: John burnt a book.
6. **Artifactual-Natural shifting**: She touched the phone.

Direct Argument Selection

- The spokesman denied the **statement** (**PROPOSITION**).
- The child threw the **ball** (**PHYSICAL OBJECT**).
- The audience didn't believe the **rumor** (**PROPOSITION**).

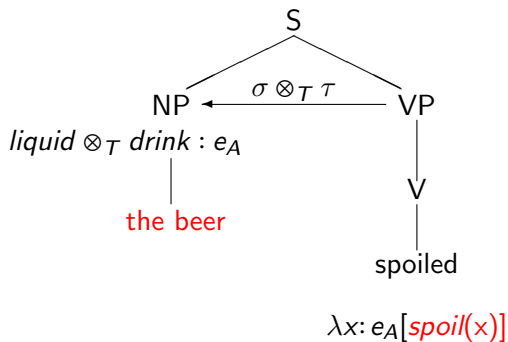
Natural Selection

1. The rock fell.



Pure Selection: Artifactual Type

1. The beer spoiled.



Pure Selection: Complex Type

1. John read the book.

