Compositionality and the Theory of Argument Selection

James Pustejovsky

LOT Winter School, Nijmegen

January 8-12, 2007

Lecture 5. Dynamic Semantics and The Syntax-Semantics Mapping

Types of Opposition: II

---

(1)a. For a binary predicate, P, the opposition is $\neg P$.

b. If the language lexicalizes both forms, then $\langle P, \neg P\rangle$, $\langle P, Q\rangle$, $\langle \neg Q, Q\rangle$.

c. For a binary adjective, $\lambda x \lambda e[dead(e, x)]$ is equivalent to $\lambda x \lambda e[\neg alive(e, x)]$.

(2)a. Polar Opposites over Scale: $sick/healthy$ and $tall/short$:

b. defined in terms of a sortal array with distinguished elements.

Principle of Sortal and Property Inertia

(3) 1. A sortal fluent f_S is not affected by the matrix predicate, unless explicitly asserted by the predication in the sentence.

2. A property fluent f_D is not affected by the matrix predicate, unless explicitly asserted by the predication in the sentence.

(4) a. $\lambda x \lambda e[man(e, x)]$

b. $\lambda x \lambda e[rock(e, x)]$

c. $\lambda x \lambda e \lambda e'[fall\_act(e', x) \wedge fall\_result(e, x) \wedge e' < e]$

Computing Event Persistence Structure

(5)a. The Principle of Inertia; objects and their properties tend to remain as they are unless explicitly affected;

b. Qualia Selection Thesis; modifiers selectively bind to specific qualia of the head noun.

(6)a. Assign each predicate an event description; $\{\delta_i\}$. The set of event descriptions will be referred to as Δ.

b. We denote the event description assigned to the matrix predicate of the clause, P, as the *core event structure*.

Given Δ, and the construction of the core event structure, for each event-denoting predicate in the expression, we apply a single test, $gate$, defined as follows.

(7) a. GATE: For an event description, $\delta \in \Delta$, in the domain of the matrix predicate P, δ is $gated$ by P only if the property denoted by δ is either $initiated$ or $terminated$ by successful assertion of P.

b. PERSIST: If δ is not gated, then it is said to $persist$ relative to the matrix predicate, P.

(8) a. Associate each event description to the event introducing it.

b. If an event description does not take wide scope (such as all those that are gated), then it is narrow scope, and is associated only with the appropriate subevents.

c. All persisting events are factored out of the expression in the event structure. They will be said to take wide *persistence* scope (p-scope) over the event description.

Examples of Event Persistence Structure

(9)a. The argument persists;

b. The head of the argument does not persist;

c. The head of the argument persists, but there are properties of the head introduced by predication that do not persist.

d. The head of the argument persists, but there are inherent properties of the head expressed in the referring expression that do not persist.

(10)a. Mary saw John.

b. A man sat on a bench.

(11) a. Mary built a house.

b. Mary ate a cookie.

(12) a. John closed the door.

b. Mary cleaned the table.

c. John painted the house.

d. A man sat down on a bench.

(13) a. People filled the empty hall.

b. Mary cleaned the dirty table.

(14) a. Mary fixed the tire.

b. Mary fixed the flat tire.

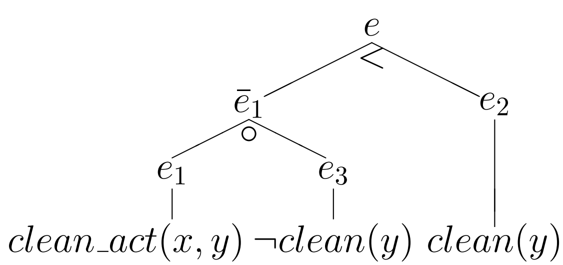
Example 1

(15) Mary cleaned the table.

(16) $\Delta = \{$mary(e$_1$,x), table(e$_2$,y), clean_act(e$_3$,x,y), $\neg$clean(e$_4$,y), clean(e$_5$,y)$\}$

From Δ, we construct an event structure associated with the matrix predicate of the sentence, shown in (57):

(17)

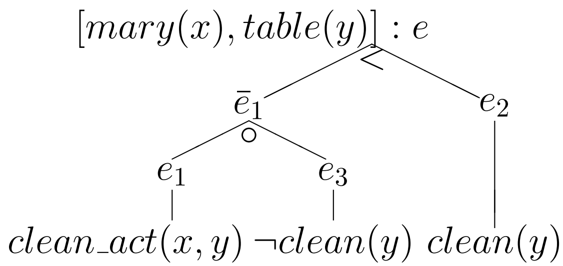


Then we apply the operation *gate*:

(18)a. $gate$(mary) fails;

b. $gate$(table) fails;

(19)



Example 2

(20) Mary cleaned the dirty table.

(21) $\Delta = \{$mary(e$_1$,x), table(e$_2$,y), clean_act(e$_3$,x,y),
$\neg$clean(e$_4$,y),
clean(e$_5$,y), dirty(e$_6$,y)$\}$

Again, we apply the operation $gate$:

(22)a. $gate$(mary) fails;

b. $gate$(table) fails;

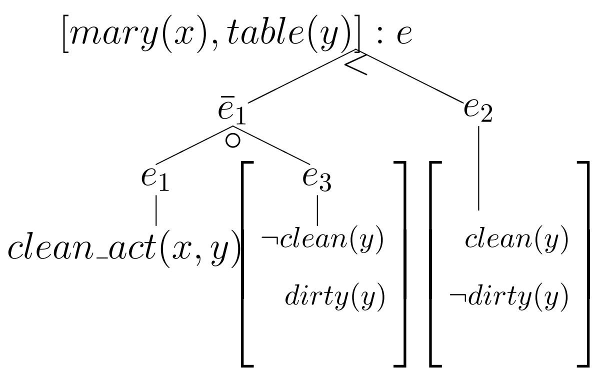
c. $gate$(dirty) succeeds;

There are two opposition structures for an adjective like *dirty*:

(23)a. $\langle dirty, \neg dirty \rangle$: Binary opposition

b. $\langle dirty, clean \rangle$: Polar opposition

(24)



Stateless and Event-based Selection

(25) a. Stateless Selection: Selection is performed independent of the operation performed over the argument. Selection is without state information.

b. Event-based Selection: Selection is performed over a trace of the operation performed over the argument. Selection is sensitive to the states the argument participates in.

Dynamic Lexical Semantics: I

---

(26) Whenever the program α is performed succesfully, ϕ holds in the discourse.

$$[\alpha]\phi$$

(27) It is possible to run the program α such that ϕ holds in the discourse.

$$\langle\alpha\rangle\phi$$

(28) Given the precondition of ψ, whenever the program α is performed succesfully, ϕ holds in the discourse.

$$\psi \rightarrow [\alpha]\phi$$

Lexicalizing the statement of change

---

(29) a. kill: $\neg dead(y) \rightarrow [kill(x, y)]dead(y)$

b. break: $\neg broken(y) \rightarrow [break(x, y)]broken(y)$

c. clean: $[clean(x, y)]clean(y)$

(30) Stateless Selection

a. $kill$: $anim \rightarrow (e_N \rightarrow t)$

b. $\lambda y : anim\ \lambda x : e_N$[$kill$(x,y)]

(31) Stateless Selection with Dynamic Interpretation

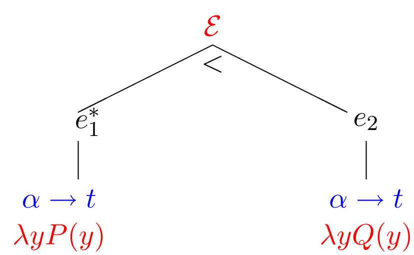
a. $kill$: $anim \rightarrow (e_N \rightarrow t)$

b. $\lambda y : anim\ \lambda x : e_N(\neg dead(y)[kill(x, y]dead(y))$

Dynamic Selection

(32)a. Let $\bar{\alpha}$ refer to the trace of the type α through the event structure, $\mathcal{E}$, associated with the predicate, $\bar{\alpha} \longrightarrow t$.

(33)



(34)a. The temperature is 25 degrees: $e \longrightarrow t$

b. The temperature is rising: $\bar{e} \longrightarrow t$

Event-based Selection

Now a predicate can be represented as two kinds of functional types:

(35) a. Stateless: $\alpha \rightarrow t$. Reference only to the argument.
b. Stateful: $\bar{\alpha} \rightarrow t$. Reference to the trace of the argument through the event structure.

Gates

Traces let us now refer to change as a property of the argument of the predicate. Hence, a change predicate has a different functional type from a stateless predicate.

(36) Gates:
Let us define a pair of type operators, $\ulcorner$ and $\urcorner$, applied over a trace, that initiate or terminate a process or state. We will call the resulting transformations, *gating functions*.

Dynamic Lexical Semantics: III

(37) a. If a is a type, then $\ulcorner a$ and $a \urcorner$ are types.

(38) a. If $a \longrightarrow b$ is a type, then $\ulcorner a \longrightarrow b$ and $a \urcorner \longrightarrow b$ are types.

(39) a. If $a \otimes c \longrightarrow b$ is a type, then $\ulcorner a \otimes c \longrightarrow b$ and $a \urcorner \otimes c \longrightarrow b$ are types.

b. If $a \otimes c \longrightarrow b$ is a type, then $a \otimes \ulcorner c \longrightarrow b$ and $a \otimes c \urcorner \longrightarrow b$ are types.

(40) a. If $a \bullet c \longrightarrow b$ is a type, then $\ulcorner a \bullet c \longrightarrow b$ and $a \urcorner \bullet c \longrightarrow b$ are types.

b. If $a \bullet c \longrightarrow b$ is a type, then $a \bullet \ulcorner c \longrightarrow b$ and $a \bullet c\urcorner \longrightarrow b$ are types.

(41) Composition of Gates:

a. $(\ulcorner a \longrightarrow t) \circ (a\urcorner \longrightarrow t) = \ulcorner a\urcorner \longrightarrow t$.

b. Hence, $\ulcorner a\urcorner$ could be said to designate the total history of type a.

Typing the Predicate as a Change

(42) Given a discourse where ϕ holds, the successful performance of the predicate α brings about $\neg\phi$ in the discourse.

$$\phi \rightarrow [\alpha]\neg\phi$$

Now assume that a gate can mark which argument undergoes the update:

(43) Gating embeds change into predicate's type:

a. $kill$: $anim^{\urcorner} \rightarrow (e_N \rightarrow t)$

b. $\lambda y : anim^{\urcorner}\ \lambda x : e_N[kill(\mathrm{x},\mathrm{y})]$

c. $\neg dead(y)[kill(x,y)]dead(y)$

Dynamic Lexical Semantics: III

(44)a. The door opened.

b. The window closed.

(45) State Transitions:

a. $open$: $phys \bullet \ulcorner aperture \rightarrow t$

b. $close$: $phys \bullet aperture \urcorner \rightarrow t$

(46) $\neg open(x)[open]open(x)$

(47) Predicates as terminating and initiating functions:

fill, empty

Two-Gate Transitions

(48) John gave a book to Mary.

(49) State Transitions:

a. $give$:

$phys \rightarrow (human \otimes \ulcorner have \rightarrow (human \otimes have \urcorner \rightarrow t))$

(50) a. $have(x, y)[give(x, y, z)]\neg have((x, y)$

b. $\neg have(z, y)[give(x, y, z)]have((z, y)$

Gates are Methods for Qualia Structure

To formally identify specific actions or relations with objects or entities, e.g., qualia roles and affordances:

1. Gates over Natural

    animal: be born, die

    apple: grow, eat

2. Gates over Artifactual

    prisoner: arrest, escape

    audience: assemble, disperse

    cake: bake, eat

3. Gates over Complex

door: build, destroy, open, close

talk: begin, end, prepare

Examples of Gates

(51)a. The prisoner escaped from the prison.

c. The audience left the theatre.

$$\begin{bmatrix} \textbf{prisoner} \\ \text{QS} = \begin{bmatrix} \text{F} = human(x) \\ \text{T=A} = \exists e[captive(e,x)] \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{audience} \\ \text{QS} = \begin{bmatrix} \text{F} = human(x) \\ \text{T=A} = \exists e,y[attend(e,x,y)] \end{bmatrix} \end{bmatrix}$$

(52)a. prisoner: $human \otimes captive$

b. audience: $human \otimes attend$

(53)a. capture$_1$: $e_N \otimes \ulcorner captive \rightarrow (e_N \rightarrow t)$

b. escape: $e_N \otimes captive\urcorner \rightarrow t$

c. capture$_2$: $e_N \otimes free\urcorner \rightarrow (e_N \rightarrow t)$

Spoiling an Event

(54) And why is it whenever you choose to roll out the seventy foot or more of polythene , which has humped up your overdraft considerably , a perfect tranquil morning is spoilt by a freak hurricane.

(55) in that the singers may learn their parts more thoroughly and thus, knowing the music perfectly, may act with greater confidence and not spoil the opera.

(56)a. $spoil$: $\epsilon \otimes T^{\neg} \rightarrow (e_N \rightarrow t)$

b. $\lambda e : \epsilon_F \lambda x : e_N[spoil(\mathrm{e})(\mathrm{x})]$

Mechanisms of Selection and Coercion with Gates

For a given argument, α, in the runtime of an event $\mathcal{E}$, one of the three situations holds:

i. α has the denotative integrity as selected by the predicate associated with $\mathcal{E}$;

ii. α acquires the denotative integrity as a result of the successful runtime of $\mathcal{E}$;

iii. α stops having the denotative integrity as a result of the successful runtime of $\mathcal{E}$.

Coercion with Gates

(1) Verbs encode specific gating functions:

a. thing: $phys$

b. animal: $anim$

c. die: $anim^{\urcorner} \rightarrow t$

(2) Selection of a Gate:

a. The animal died.

b. $\lambda x : anim^{\urcorner}[die(\mathsf{x})](the\ animal : anim)$

(3) Introduction of a Gate:

a. The thing died.

b. $\lambda x : anim^{\urcorner}[die(\mathsf{x})](the\ thing: phys)$

(1) a. man: $human$

b. prisoner: $human \otimes captive$

c. escape: $e_N \otimes captive^{\urcorner} \rightarrow t$

(2) Selection of a Gate:

a. The prisoner/captive escaped.

b. $\lambda x : e_N \otimes captive^{\urcorner}$[$escape$(x)]

($the\ prisoner$: $human \otimes captive$)

(3) Introduction of a Gate:

a. The man escaped.

b. $\lambda x : e_N \otimes captive^{\urcorner}$[$escape$(x)]

($the\ man$: $human$)

Type Characterization of Verb Classes

(1) a.Let $\tilde{\alpha} =_{df} \ulcorner\alpha \vee \alpha\urcorner$.

1. b. An unaccusative can be characterized as having the type $\tilde{\alpha} \rightarrow t$.
2. c. An unergative can be characterized as having the type $\alpha \rightarrow t$.

Derivation involving Gating: Introduction

1. Mary cleaned the car.
2. John filled the glass.
3. Mary broke vase.

Encoding Change on the Argument

(57) John cleaned the car.

(58) a. $\lambda y \lambda x(clean(x, y)), \langle y\colon phys, x\colon phys\rangle$
b. clean: $phys \rightarrow (phys \rightarrow t)$

Encoding the change, we would have the introduction of a gated predicate over the internal argument y:

(59) clean: $phys \otimes {}^{\ulcorner}clean \rightarrow (phys \rightarrow t)$

Example Walk-through of Gate-Introduction

Derivation involving Gating: Exploitation

1. an escaped prisoner.
2. a former mayor.
3. Mary release the prisoner.

Gate-Exploitation is an Elimination Operation

an escaped prisoner

(60) $\lambda P \lambda x \exists e[\text{escaped}(e, x) \wedge P(e, x)]$
x: p;
escaped: $(p \otimes captive^{\neg} \rightarrow t) \rightarrow (p \otimes captive^{\neg} \rightarrow t)$.

(61) $\lambda v \text{prisoner}(v)$: $(human \otimes captive) \rightarrow t$

(62) $[\![\text{escaped}]\!] = \lambda P \lambda x \lambda e_2 \exists e_1[\neg captive(e_2, x) \wedge captive(e_1, x) \wedge e_1 \leq e_2 \wedge P(e_2, x)]$;

(63) $[\![\text{prisoner}]\!] = \lambda x \lambda e[human(x,) \wedge captive(e, x)]$

(64) Apply Gate Exploitation, giving:

$[\![$escaped prisoner$]\!] =$
$\lambda x \lambda e_2 \exists e_1 [\neg captive(e_2, x) \;\wedge\; captive(e_1, x) \wedge$
$e_1 \leq e_2 \wedge human(e_2, x)];$

Temporal Adjectives as Gate-Exploitation

a former boxer

(65) former: $(p \otimes E^{\urcorner} \rightarrow t) \rightarrow (p \otimes E^{\urcorner} \rightarrow t)$.

(66) λvboxer(v) : $(human \otimes_T box) \rightarrow t$

(67) $[\![$former$]\!]$ =
$\lambda P \lambda x \lambda e_2 \exists e_1[\neg P(e_2, x) \wedge P(e_1, x) \wedge e_1 \leq e_2]$;

(68) $[\![$boxer$]\!] = \lambda x \lambda e[human(x,) \wedge box(e, x)]$

(69) Apply Gate Exploitation, giving:
$[\![$former boxer$]\!]$ =
$\lambda x \lambda e_2 \exists e_1[\neg box(e_2, x) \wedge box(e_1, x) \wedge e_1 \leq e_2 \wedge human(e_2, x)]$;

Gating in Discourse

(70)a. The prisoner escaped from the prison.

b. The escapee has been put in police custody.

c. The audience left the theatre.

(71) $\begin{bmatrix} \textbf{audience} \\ \text{QS} = \begin{bmatrix} \text{F} = human(x) \\ \text{T=A} = \exists e, y[attend(e,x,y)] \end{bmatrix} \end{bmatrix}$

(72)a. $prisoner$: $human \otimes_A captive$

b. $audience$: $human \otimes_{T/A} attend$

c. $pedestrian$: $human \otimes_A walk\_street$

(73) The audience$_i$ left the music hall.

(74) $$\lambda x \lambda e \begin{array}{|l|}\hline \pi, \pi_1 \\ \hline \\ \pi : \begin{array}{|l|}\hline x, y, e' \\ \hline \\ \textit{leave\_act}(e, x, y) \\ \textit{audience}(e', x) \\ \\ \hline \end{array} \\ \\ \pi_1 : \begin{array}{|l|}\hline \mathsf{s} \\ \hline \\ \neg \textit{in}(s, x, y) \\ \\ \hline \end{array} \\ \\ \textit{Result}(\pi, \pi_1) \\ \\ \hline \end{array}$$

What is contributed through lexical inference?

(75)a. $*$It$_i$ then went home.

b. They$_i$ then went home.

c. It$_i$/They$_i$ had just heard Bernard Haitink's last performance.

How is the anaphoric restriction on (75a) stated?

(76)a. John burned the letter$_i$ after he got it$_i$.

b. It$_i$ still makes him mad.

Exploiting the Event Structure in Discourse

(77) a. Mary was upset.

b. John comforted her.

(78) a. Mary was upset.

b. John told her a very bad joke.

The Syntax-Semantics Mapping

Event Decomposition and Linking Theory
(Pustejovsky, 1995)

---

a. Event Headedness: A way of indicating a foregrounding and backgrounding of sub-event. The arguments of a headed event must be expressed.

b. Argument Covering: An argument x is covered only if:

(i) x is linked to a position in s-structure; or

(ii) x is logically bound to a covered argument y; or

(iii) x is existentially closed by virtue of its type.

c. Qualia Saturation: A qualia structure is saturated only if all arguments in the qualia are $covered$.

Event Decomposition in GL

(79)a. Q_i: $R(e_1^*, x, y) \longrightarrow x$:SUBJ, y:OBJ

b. Q_j: $P(e_2, y) \longrightarrow$ shadowed

(80)a. Q_i: $R(e_1, x, y) \longrightarrow$ shadowed

b. Q_j: $P(e_2^*, y) \longrightarrow y$:SUBJ

Qualia Structure with Opposition

$$
\begin{bmatrix}
\textbf{kill} & \\
\text{EVENTSTR} = & \begin{bmatrix} \text{E}_0 = \textbf{e}_0\textbf{:state} \\ \text{E}_1 = \textbf{e}_1\textbf{:process} \\ \text{E}_2 = \textbf{e}_2\textbf{:state} \\ \text{RESTR} = <_\alpha \\ \text{HEAD} = \textbf{e}_1 \end{bmatrix} \\
\text{ARGSTR} = & \begin{bmatrix} \text{ARG1} = \boxed{1} \begin{bmatrix} \textbf{ind} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \\ \text{ARG2} = \boxed{2} \begin{bmatrix} \textbf{animate\_ind} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \end{bmatrix} \\
\text{QUALIA} = & \begin{bmatrix} \textbf{cause-lcp} \\ \text{FORMAL} = \textbf{dead}(\textbf{e}_2, \boxed{2}) \\ \text{AGENTIVE} = \textbf{kill\_act}(\textbf{e}_1, \boxed{1}, \boxed{2}) \\ \text{PRECOND} = \neg\textbf{dead}(\textbf{e}_0, \boxed{2}) \end{bmatrix}
\end{bmatrix}
$$

Four Types of Logical Parameters for Lexical Items

1. True Arguments (ARG): Syntactically realized parameters of the lexical item;
2. Default Arguments (D-ARG): Parameters which participate in the logical expressions in the qualia, but which are not necessarily expressed syntactically; e.g. "John built the house *with bricks*".
3. Shadow Arguments (S-ARG): Logical parameters which are semantically incorporated into the lexical item. They can be expressed only by operations of subtyping; e.g. " Mary buttered her toast *with an*

expensive butter."

4. Optional Arguments: Parameters which modify the logical expression, but are part of situational or propositional interpretation, not any particular lexical item's semantic representation. These include adjunct expressions of temporal or spatial modification.

Expression of Argument Types

1. a. [Mary]$_{arg1}$ built [a house]$_{arg2}$.
   b. [Mary]$_{arg1}$ built [a house]$_{arg2}$ [with wood]$_{d-arg1}$.
   c. [Mary]$_{arg1}$ built [a wooden$_{d-arg1}$ house]$_{arg2}$] [with pine]$_{s-arg1}$.
2. a. [Mary]$_{arg1}$ arrived.
   b. [Mary]$_{arg1}$ arrived [home]$_{d-arg1}$.

Argument Coherence

(81) The relation expressed by the causing event and that expressed by the resulting event must make reference to at least one parameter in common. This reference can be direct or indirect:

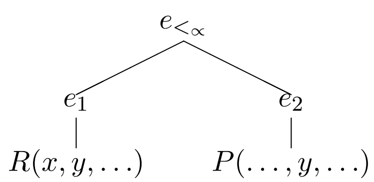
a. Direct Causation :

$$\left[\text{QUALIA} = \left[\begin{array}{l}\text{FORMAL} = \alpha\_\mathbf{result}(\mathbf{e}_2,\mathbf{y}) \\ \text{AGENTIVE} = \alpha\_\mathbf{act}(\mathbf{e}_1,\mathbf{x},\mathbf{y})\end{array}\right]\right]$$

b. Indirect (Constitutive) Causation :

$$\left[\text{QUALIA} = \left[\begin{array}{l}\text{CONST} = \mathbf{part\_of}(\mathbf{z},\mathbf{y}) \\ \text{FORMAL} = \alpha\_\mathbf{result}(\mathbf{e}_2,\mathbf{y}) \\ \text{AGENTIVE} = \alpha\_\mathbf{act}(\mathbf{e}_1,\mathbf{x},\mathbf{z})\end{array}\right]\right]$$

(82)



(83)
$$\begin{bmatrix} \alpha \\ \text{EVENTSTR} = \begin{bmatrix} \text{E}_1 = \textbf{e}_1\textbf{:process} \\ \text{E}_2 = \textbf{e}_2\textbf{:state} \\ \text{RESTR} = <_{\propto} \\ \text{HEAD} = \end{bmatrix} \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = \boxed{1} \\ \text{ARG2} = \boxed{2} \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \textbf{default-causative-lcp} \\ \text{FORMAL} = \alpha\_\textbf{result}(\textbf{e}_2, \boxed{2}) \\ \text{AGENTIVE} = \alpha\_\textbf{act}(\textbf{e}_1, \boxed{1}, \boxed{2}) \end{bmatrix} \end{bmatrix}$$

(84)a. Left-headed events : e.g., Direct causative accomplishments, such as *kill*, *murder*, etc.

b. Right-headed events : e.g., Direct causative achievements, such as *die*, *arrive*.

c. Headless events : e.g., causative/unaccusative verbs, such as *sink*, *break*, *burn*.

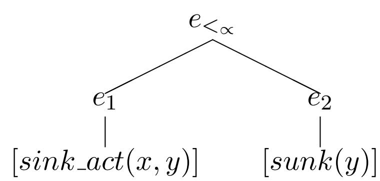
(85)a. I nemici hanno affondato la nave.

“The enemy sank the boat.”

b. La nave è affondata.

"The boat sank."

(86)



(87)