Computational Models of Events

Lecture 3: Sub-atomic and Dynamic Models of Events

James Pustejovsky
Brandeis University

ESSLLI 2018 Summer School
Sofia, Bulgaria
August 6-10, 2018

Today's Outline

- Kowalski and Sergot (1986): Event Calculus
- Fernando: Segmented Event Logic
- Moens and Steedman (1988), Pustejovsky (1991)
- Pustejovsky and Moszkowicz (2011): Dynamic Interval Temporal Logic
- Naumann (1999): Dynamic Event Semantics

Event Calculus

Lambalgen and Hamm (2005)

- what distinguishes this approach to semantics: comparison with standard model theoretic semantics
- cognitive semantics as usually conceived
- psycholinguistic approaches to semantics
- event calculus: a fusion between these approaches

Non-cognitive formal approaches to natural language semantics

"I distinguish two topics: first, the description of possible languages or grammars as abstract semantic systems whereby symbols are associated with aspects of the world; and second, the description of the psychological and sociological facts whereby a particular one of these abstract semantic systems is the one used by a person or a population. Only confusion comes of mixing these two topics. . . . Semantics without truth conditions is not semantics." (David Lewis)

Characteristic features

- necessary and sufficient truth conditions (Lewis), possible worlds semantics (Montague, Dowty, ...),
- semantic representations are sets in classical *models* $(D; R, \ldots; f, \ldots; a, \ldots)$, or possible world structures, i.e. families of such models related by an accessibility relation
- models often viewed as ‘general metaphysics’, general description of how the world is
- ‘sense relations’ modelled by entailment: deeper analysis of lexicography
- hence uses logical techniques
- example: (branching) tense logic for modelling the future tense

Tense logic

- modal logic with operators P ('in the past'), F ('in the future'), $H = \neg P \neg$ ('always in the past'), $G = \neg F \neg$ ('always in the future')
- interpreted on *tense structure* $(T, <, V)$, where $<$ is at least antisymmetric and transitive, and each proposition letter q is interpreted as subset $V(q)$ of T
- for $t \in T$, $(T, <, V) \models Pq[t]$ iff $\exists s < t(s \in V(q))$ and $(T, <, V) \models Fq[t]$ iff $\exists s > t(s \in V(q))$
- axioms e.g. $q \rightarrow GPq$, $q \rightarrow HFq$ (minimal tense logic); $PPq \rightarrow Pq$
- unicity of the past: $Pq \rightarrow H(Pq \vee q \vee Fq)$ – still allows branching future
- tries to explain tenses purely temporally–but we have seen that richer structure is involved

Cognitively inspired approaches to semantics: 'Languages of the mind'

"Conceptual Semantics ... is concerned most directly with the form of the internal mental representations that constitute conceptual structure and with the formal relations between this level and other levels of representation. ... Conceptual Semantics is thus a prerequisite to [truth conditional] semantics: the first thing one must know about an English sentence is its translation into conceptual structure. Its truth conditions should then follow from its conceptual structure plus rules of inference, which are stated as well in terms of conceptual structure." (Ray Jackendoff)

Characteristic features

- semantic representations are *mental* entities – what is their 'most general' theory?
- criticism of traditional formal approaches to semantics
  - takes dim view of set theoretic models and truth conditions
  - prototypes
  - role of analogy/metaphor . . .
- tries to anchor semantics in 'conceptual structure'
- 'componential analysis in terms of supposed primitives of conceptual structure/language of the mind: EVENT, PATH, STATE, GOAL, CAUSE ...

  a. $[_S[_{NP}$ John$][_{VP}$ ran $[_{PP}$ into $[_{NP}$ the room$]]]]$

  b. $[_{Event}$ GO $([_{Thing}$ JOHN$],[_{Path}$ TO $([_{Place}$ IN $([_{Thing}$ ROOM$])])])]$
- but why do cognitive linguists reject formal/logical methods?

Psychologists' views on semantics

- emphasis on *algorithms*–e.g. 'sets of possible worlds' irrelevant because non-computable
- e.g. meaning of an expression is *algorithm* which tests whether object falls under the expression (Miller and Johnson-Laird, *Language and Perception*)
- psychologists' aim is to understand issues like language comprehension and production, in quantitative terms (e.g reaction times, error rates)
- psychologists are very fond of network architectures such as spreading activation nets
- compare Marr's three levels of inquiry/division of labour
  - information processing task
  - algorithm
  - neural implementation

Event calculus: a language for mental representations

- human processing of temporal notions is in terms of goals/plans/actions
- this also requires a theory of causality and change, which comes in two forms
  - instantaneous change
  - continuous change

Event calculus, and what it talks about

- actions and events: e, . . . (‘break’)
- time–varying properties or *fluents*: f, . . . (‘being broken’), possibly with parameters
- individuals (‘John’)
- instants of time, interpreted as real numbers
- various other real quantities (e.g. position, velocity)
- a *goal* is a desired state of affairs
- a *plan* is a sequence of actions which achieves some goal

Event calculus: primitive predicates ...

- predicates such as $<$ over the reals
- instantaneous change
  1. $Initially(f)$
  2. $Happens(e, t)$
  3. $Initiates(e, f, t)$
  4. $Terminates(e, f, t)$
  5. $Clipped(s, f, t)$
  6. $HoldsAt(f, t)$
- continuous change
  1. $Releases(e, f, t)$
  2. $Trajectory(f_1, t, f_2(x), d)$

Axioms for the event calculus, instantaneous change only

Axiom'

$$Initially(f) \;\wedge\; \neg Clipped(0, f, t) \rightarrow HoldsAt(f, t)$$

Axiom'

$$Happens(e, t) \wedge Initiates(e, f, t) \wedge \; t < t' \; \wedge \neg Clipped(t, f, t') \\ \rightarrow HoldsAt(f, t')$$

Axiom'

$$Happens(e, s) \wedge t < s < t' \wedge Terminates(e, f, s) \;\rightarrow\; Clipped(t, f, t')$$

Axioms for the event calculus, full version

Axiom
$Initially(f) \rightarrow HoldsAt(f, 0)$

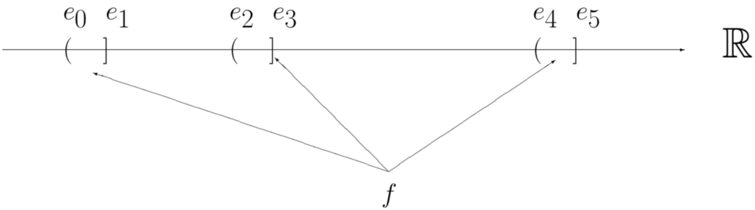
Axiom
$HoldsAt(f, r) \wedge r < t \wedge \neg\exists s < r\, HoldsAt(f, s) \wedge \neg Clipped(r, f, t) \rightarrow HoldsAt(f, t)$

Axiom
$Happens(e, t) \wedge Initiates(e, f, t) \wedge t < t' \wedge \neg Clipped(t, f, t') \rightarrow HoldsAt(f, t')$

Axiom
$Happens(e, t) \wedge Initiates(e, f_1, t) \wedge t < t' \wedge t' = t + d \wedge Trajectory(f_1, t, f_2, d) \wedge \neg Clipped(t, f_1, t') \rightarrow HoldsAt(f_2, t')$

Axiom
$Happens(e, s) \wedge t < s < t' \wedge (Terminates(e, f, s) \vee Releases(e, f, s)) \rightarrow Clipped(t, f, t')$

Typical models of the event calculus



- left-open because fluent f does not hold at the moment it is initiated
- a version of Zeno's paradox: there cannot be *both* a last moment at which f does not hold and a first moment at which f holds
- best to assume last moment at which f does not hold

Event calculus: states and scenarios

- *goal* of the form $?HoldsAt(f, t)$ or $?Happens(e, t)$
- *scenario* describes cognitive representation of agent and environment in language of event calculus
- scenario must be theory of specific syntactic form to be plausible as memory structure
- syntactic form of scenario defined in two steps

Definition

A state $S(t)$ at time t is a conjunction of literals involving only

1. literals of the form $(\neg)HoldsAt(f, t)$, for t fixed and possibly different f,
2. equalities between fluent terms, and between event terms
3. equations and inequalities involving real numbers

Scenario

Definition

A *scenario* is a conjunction of statements of the form

1. $Initially(f)$,
2. $\forall t(S(t) \rightarrow Initiates(e, f, t))$,
3. $\forall t(S(t) \rightarrow Terminates(e, f, t))$,
4. $\forall t(S(t) \rightarrow Releases(e, f, t))$,
5. $\forall t, s(S(t, s) \wedge Happens(e_0, s) \rightarrow Happens(e, t))$,
6. $S(f_1, f_2, t, d) \rightarrow Trajectory(f_1, t, f_2, d)$,

where the $S(t), \ldots$ are states in the sense of definition 1.

Causation and continuous change

- axioms for instantaneous change formalize principle of *inertia*: after the cause has stopped acting, the caused state does not change
- this principle is not valid for continuous causation
- $Releases(e, f, t)$ stipulates that when e happens, f is no longer subject to the principle of inertia
- example: crossing the street
  $HoldsAt(distance(x),\ t) \rightarrow$
  $Trajectory(crossing,t,distance(x+d),d)$

Lexical entry for the accomplishment 'cross the street'

1. $Happens(start, t_0)$
2. $HoldsAt(crossing,\ now)$
3. $Initially(one\text{–}side)$
4. $Initially(distance(0))$
5. $HoldsAt(distance(m),\ t) \wedge HoldsAt(crossing,\ t) \rightarrow Happens(reach,\ t)$
6. $Initiates(start,\ crossing,\ t)$
7. $Releases(start,\ distance(0),\ t)$
8. $Initiates(reach,\ other\text{–}side,\ t)$
9. $Terminates(reach,\ crossing,\ t)$
10. $HoldsAt(distance(x),\ t) \rightarrow Trajectory(crossing, t, distance(x+d), d)$
11. $HoldsAt(distance(x_1), t) \wedge HoldsAt(distance(x_2),\ t) \rightarrow x_1 = x_2.$

Plans contained in a lexical entry

- consider the goal $?HoldsAt(other\text{–}side, t)$, $t \geq now$
- want to *derive* plan for achievement of this goal
- do this by backward chaining using axioms of the event calculus and the scenario
- e.g. by axiom 3 *reach* event must have occurred,
- by scenario 5 this can only be if *distance m* has been covered
- by axiom 4 this distance can be covered only if the activity *crossing* persists for sufficiently long, *etc.*
- compare this semantic representation with set theoretic representation, such as

$$\{(a, b) \mid cross(a, b)\},\ s = \text{‘the street’}$$

Vendler Event Classes + Semelfactive

- STATE: John loves his mother.

Vendler Event Classes + Semelfactive

- STATE: John loves his mother.
- ACTIVITY: Mary played in the park for an hour.

Vendler Event Classes + Semelfactive

- STATE: John loves his mother.
- ACTIVITY: Mary played in the park for an hour.
- ACCOMPLISHMENT: Mary wrote a novel.

Vendler Event Classes + Semelfactive

- STATE: John loves his mother.
- ACTIVITY: Mary played in the park for an hour.
- ACCOMPLISHMENT: Mary wrote a novel.
- ACHIEVEMENT: John found a Euro on the floor.

Vendler Event Classes + Semelfactive

- STATE: John loves his mother.
- ACTIVITY: Mary played in the park for an hour.
- ACCOMPLISHMENT: Mary wrote a novel.
- ACHIEVEMENT: John found a Euro on the floor.
- POINT: John knocked on the door (for 2 minutes).

Fernando's Segmented Event Theory

Fernando (2008, 2013)

- Timelines interpreting interval temporal logic formulas are segmented into strings which serve as semantic representations for tense and aspect.
- The strings have bounded but refinable granularity, suitable for analyzing (im)perfectivity, durativity, telicity, and various relations including branching.

Fernando's Segmented Event Theory

A sentence in the simple past, such as (1a), uttered at (speech) time S can be pictured as a timeline (1b), describing an event E (Ernest explaining) prior to S.

(1) a. Ernest explained.

b. | E | S | (depicting $E \prec S$)

We can view the event E in (1b) as an unbroken point, wholly to the left of S, $E \prec S$. By contrast, in the timeline (2a) for the progressive (2b), E splits into three boxes, the middle of which contains also a *reference time* R (Reichenbach, 1947).[1]

(2) a. | E | E,R | E | (depicting $R \sqsubset E$)

b. Ernest explaining

Fernando's Segmented Event Theory

We begin with temporal formulas, which for the sake of brevity, we hereafter call *fluents*. A fluent such as E, R or S can occur as a whole, as E and S do in (1b), or as segmented, as E does in (2a). We formulate the notions of *whole* and *segmented* model-theoretically in section 2, defining a map $\varphi \mapsto \varphi_\circ$ on fluents φ through which the picture (2a) is sharpened to (3) with $E_\circ$ segmented.

(3) | $E_\circ$ | $E_\circ$,R | $E_\circ$ | (segmented $E_\circ$, whole R)

The map $\varphi \mapsto \varphi_\circ$ is essentially a universal grinder (the right half of an adjoint pair with a universal packager, max)

$$\frac{\text{whole}}{\text{segmented}} \approx \frac{\text{count}}{\text{mass}}$$

Fernando's Segmented Event Theory

Fix a set Φ of fluents. Fluents in Φ are interpreted relative to a Φ-*timeline*, a triple $\mathfrak{A} = \langle T, \prec, \models \rangle$ consisting of a linear order $\prec$ on a non-empty set T of (temporal) points, and a binary relation $\models$ between intervals I (over $\prec$) and fluents $\varphi \in \Phi$. An interval is understood here to be a nonempty subset I of T with no holes — i.e. $t \in I$ whenever $t_1 \prec t \prec t_2$ for some pair of points t_1, t_2 in I.[2] $I \models \varphi$ is pronounced "φ holds at I" or "I satisfies φ" (in $\mathfrak{A}$).

A fluent φ is said to be $\mathfrak{A}$-*segmented* if for all intervals I and I' such that $I \cup I'$ is an interval, φ holds at I and at I' precisely if it does at their union

$$I \models \varphi \text{ and } I' \models \varphi \iff I \cup I' \models \varphi.$$

A simple way for a fluent φ to be $\mathfrak{A}$-segmented is by holding at an interval I precisely if it holds at all points of I

$$I \models \varphi \iff (\forall t \in I)\ \{t\} \models \varphi$$

in which case we say φ is $\mathfrak{A}$-*pointed*.[3] A fluent is $\mathfrak{A}$-*singular* if at most one interval satisfies it. Generalizing $\mathfrak{A}$-singular fluents, we call a fluent φ $\mathfrak{A}$-*whole* if for all intervals I and I' such that $I \cup I'$ is an interval,

$$I \models \varphi \text{ and } I' \models \varphi \text{ implies } I = I'.$$

Fernando's Segmented Event Theory

Semelfactives, activities (= processes), achievements (= culminations) and accomplishments (= culminated processes) are commonly differentiated on the basis of durativity and telicity (Moens and Steedman, 1988; Pulman, 1997).

(12) a. A semelfactive is non-durative and atelic

b. An activity is durative but atelic

c. An achievement is non-durative but telic

d. An accomplishment is telic and durative

Under the present approach based on strings, (12) can be sharpened to (13).

(13) a. A φ-semelfactive $\trianglerighteq$ $\boxed{\langle\supset\rangle\varphi}$

b. A φ-activity $\trianglerighteq$ $\boxed{\varphi}\boxed{\varphi}\boxed{\varphi}^{+}$ (presupposing φ is $\mathfrak{A}$-segmented)

c. A ψ-achievement $\trianglerighteq$ $\boxed{\neg\psi}\boxed{\psi}$

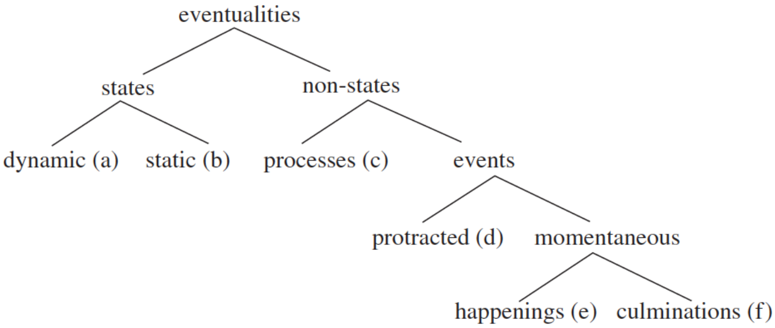
d. An accomplishment built from a φ-activity culminating in ψ

$$\trianglerighteq \boxed{\varphi, \neg\psi}\boxed{\varphi, \neg\psi}\boxed{\varphi, \neg\psi}^{+}\boxed{\psi}$$

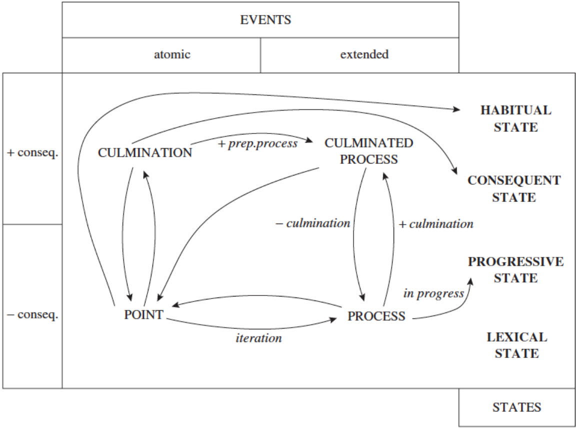
Fernando's Segmented Event Theory

$r \in$**Allen**	$s_r \in (2^{\{e,e'\}})^+$
$e = e'$	$\boxed{e, e'}$
e s e'	$\boxed{e, e'}\boxed{e'}$
e si e'	$\boxed{e, e'}\boxed{e}$
e f e'	$\boxed{e'}\boxed{e, e'}$
e fi e'	$\boxed{e}\boxed{e, e'}$
e d e'	$\boxed{e'}\boxed{e, e'}\boxed{e'}$
e di e'	$\boxed{e}\boxed{e, e'}\boxed{e}$
e o e'	$\boxed{e}\boxed{e, e'}\boxed{e'}$
e oi e'	$\boxed{e'}\boxed{e, e'}\boxed{e}$
e m e'	$\boxed{e}\boxed{e'}$
$e < e'$	$\boxed{e}\boxed{}\boxed{e'}$
e mi e'	$\boxed{e'}\boxed{e}$
$e > e'$	$\boxed{e'}\boxed{}\boxed{e}$

Bach Eventuality Typology (Bach, 1986)



Event Transition Graph (Moens and Steedman 1988)



Incremental Theme Verbs

- Certain NP's measure out the event. They are direct objects consumed or created in increments over time (cf. *eat an apple* vs. *push a chart*) (Tenny 1994).

Incremental Theme Verbs

- Certain NP's measure out the event. They are direct objects consumed or created in increments over time (cf. *eat an apple* vs. *push a chart*) (Tenny 1994).
- In *Mary drank a glass of wine* "every part of the glass of wine being drunk corresponds to a part of the drinking event" (Krifka 1992)

Incremental Theme Verbs

- Certain NP's measure out the event. They are direct objects consumed or created in increments over time (cf. *eat an apple* vs. *push a chart*) (Tenny 1994).
- In *Mary drank a glass of wine* "every part of the glass of wine being drunk corresponds to a part of the drinking event" (Krifka 1992)
- "Incremental themes are arguments that are completely processed only upon termination of the event, i.e., at its end point" (Dowty 1991).

Degree Achievements

- Verbs with variable aspectual behavior: they seems to be change of state verbs like other achievements , but allow durational adverbs (Dowty 1979, Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).

Degree Achievements

- Verbs with variable aspectual behavior: they seems to be change of state verbs like other achievements , but allow durational adverbs (Dowty 1979, Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- No implication that exactly the same change of state took place over and over again (no semelfactives).

Degree Achievements

- Verbs with variable aspectual behavior: they seems to be change of state verbs like other achievements , but allow durational adverbs (Dowty 1979, Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- No implication that exactly the same change of state took place over and over again (no semelfactives).
- Scalar predicates: verbs which lexically specify a change along a scale inasmuch as they denote an ordered set of values for a property of an event argument (Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).

Degree Achievements

- Verbs with variable aspectual behavior: they seems to be change of state verbs like other achievements , but allow durational adverbs (Dowty 1979, Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- No implication that exactly the same change of state took place over and over again (no semelfactives).
- Scalar predicates: verbs which lexically specify a change along a scale inasmuch as they denote an ordered set of values for a property of an event argument (Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- For example *cool*, *age*, *lenghten*, *shorten*; *descend*.
- *Let the soup cool for 10 minutes*.
- *I went on working until the soup cooled*.

Points

- Moens and Steedman 1988 analyze point expressions as those that are not normally associated to a consequent state (consequent state defined as no transition to a new state in the world – according to Moens and Steedman a point is an event whose consequences are not at issue in the discourse).

Points

- Moens and Steedman 1988 analyze point expressions as those that are not normally associated to a consequent state (consequent state defined as no transition to a new state in the world – according to Moens and Steedman a point is an event whose consequences are not at issue in the discourse).
- Semelfactives (Smith 1990, Rothstein 2004).

Points

- Moens and Steedman 1988 analyze point expressions as those that are not normally associated to a consequent state (consequent state defined as no transition to a new state in the world – according to Moens and Steedman a point is an event whose consequences are not at issue in the discourse).
- Semelfactives (Smith 1990, Rothstein 2004).
- **arrived/landed for five minutes*, *knocked/tapped for five minutes*.

Points

- Moens and Steedman 1988 analyze point expressions as those that are not normally associated to a consequent state (consequent state defined as no transition to a new state in the world – according to Moens and Steedman a point is an event whose consequences are not at issue in the discourse).
- Semelfactives (Smith 1990, Rothstein 2004).
- **arrived/landed for five minutes*, *knocked/tapped for five minutes*.
- Points admit iterative readings under coercive contexts (Moens and Steedman 1988).

Aspectual Composition

- Bare plurals and mass-terms arguments can make a sentence with a telic predicate behave as if it were 'durative' or 'imperfective' in aspect (Verkuyl 1972).

Aspectual Composition

- Bare plurals and mass-terms arguments can make a sentence with a telic predicate behave as if it were 'durative' or 'imperfective' in aspect (Verkuyl 1972).
- *John drank a glass of beer* (perfective).

Aspectual Composition

- Bare plurals and mass-terms arguments can make a sentence with a telic predicate behave as if it were 'durative' or 'imperfective' in aspect (Verkuyl 1972).
- *John drank a glass of beer* (perfective).
- *John drank beer* (imperfective).

Aspectual Composition

- Bare plurals and mass-terms arguments can make a sentence with a telic predicate behave as if it were 'durative' or 'imperfective' in aspect (Verkuyl 1972).
- *John drank a glass of beer* (perfective).
- *John drank beer* (imperfective).

Aspectual Coercion

- “A person *leads* somebody somewhere” (PROCESS) vs. “A road *leads* somewhere” (STATE)

Aspectual Coercion

- “A person *leads* somebody somewhere” (PROCESS) vs. “A road *leads* somewhere” (STATE)
- “An object *falls* to the ground” (TRANSITION) vs. “A case *falls* into a certain category” (STATE)

(1) a. EVENT → STATE | PROCESS | TRANSITION

Subatomic Event Structure

Pustejovsky (1991)

(2) a. EVENT → STATE | PROCESS | TRANSITION
    b. STATE: → e

Subatomic Event Structure

Pustejovsky (1991)

(3) a. EVENT $\rightarrow$ STATE | PROCESS | TRANSITION

b. STATE: $\rightarrow$ e

c. PROCESS: $\rightarrow$ $e_1 \ldots e_n$

(4) a. EVENT → STATE | PROCESS | TRANSITION

b. STATE: → e

c. PROCESS: → $e_1 \ldots e_n$

d. TRANSITION$_{ach}$: → STATE STATE

Subatomic Event Structure

Pustejovsky (1991)

(5) a. EVENT $\rightarrow$ STATE | PROCESS | TRANSITION

b. STATE: $\rightarrow$ e

c. PROCESS: $\rightarrow$ $e_1 \ldots e_n$

d. TRANSITION$_{ach}$: $\rightarrow$ STATE STATE

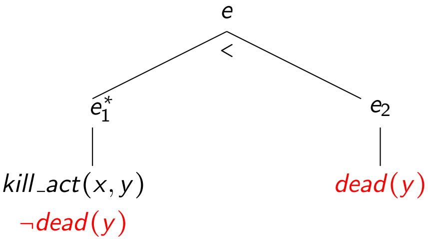
e. TRANSITION$_{acc}$: $\rightarrow$ PROCESS STATE

Qualia Structure for Causative

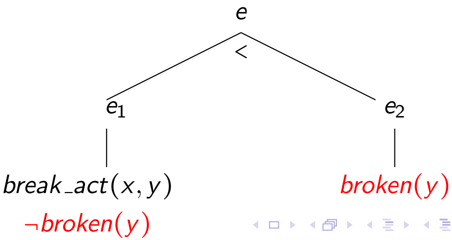
Pustejovsky (1995)

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

(6) kill



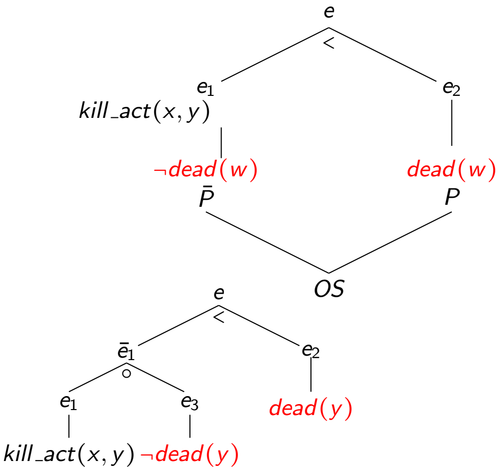
(7) break



Qualia Structure with Opposition Structure

$$
\left[\begin{array}{l}
\textbf{kill} \\
\text{EVENTSTR} = \left[\begin{array}{l}
\color{red}{E_0 = \mathbf{e_0{:}state}} \\
E_1 = \mathbf{e_1{:}process} \\
E_2 = \mathbf{e_2{:}state} \\
\text{RESTR} = <_{\propto} \\
\text{HEAD} = \mathbf{e_1}
\end{array}\right] \\
\text{ARGSTR} = \left[\begin{array}{l}
\text{ARG1} = \boxed{1}\left[\begin{array}{l}\textbf{ind} \\ \text{FORMAL} = \textbf{physobj}\end{array}\right] \\
\text{ARG2} = \boxed{2}\left[\begin{array}{l}\textbf{animate\_ind} \\ \text{FORMAL} = \textbf{physobj}\end{array}\right]
\end{array}\right] \\
\text{QUALIA} = \left[\begin{array}{l}
\textbf{cause-lcp} \\
\text{FORMAL} = \color{red}{\mathbf{dead}(e_2, \boxed{2})} \\
\text{AGENTIVE} = \mathbf{kill\_act}(e_1, \boxed{1}, \boxed{2}) \\
\text{PRECOND} = \color{red}{\neg\mathbf{dead}(e_0, \boxed{2})}
\end{array}\right]
\end{array}\right]
$$

Opposition is Part of Event Structure



Dynamic Extensions to GL

- Qualia Structure: Can be interpreted dynamically

Dynamic Extensions to GL

- Qualia Structure: Can be interpreted dynamically
- Dynamic Selection: Encodes the way an argument participates in the event

Dynamic Extensions to GL

- Qualia Structure: Can be interpreted dynamically
- Dynamic Selection: Encodes the way an argument participates in the event
- Tracking change: Models the dynamics of participant attributes

Inherent Dynamic Aspect of Qualia Structure

- Parameters of a verb, P, extend over sequential frames of interpretation (subevents).

Inherent Dynamic Aspect of Qualia Structure

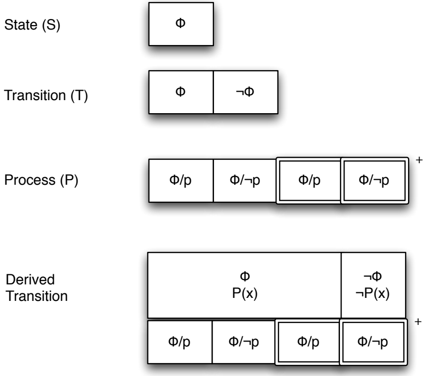
- Parameters of a verb, P, extend over sequential frames of interpretation (subevents).
- P is decomposed into different subpredicates within these events:

Inherent Dynamic Aspect of Qualia Structure

- Parameters of a verb, P, extend over sequential frames of interpretation (subevents).
- P is decomposed into different subpredicates within these events:

$$\text{Verb}(\text{Arg}_1\text{Arg}_2) \Longrightarrow \lambda y \lambda x \boxed{P_1(x,y)}_A \boxed{P_2(y)}_F$$

Frame-based Event Structure



2nd Conference on CTF, Pustejovsky (2009)

Dynamic Event Structure

- Events are built up from multiple (stacked) layers of primitive constraints on the individual participants.

Dynamic Event Structure

- Events are built up from multiple (stacked) layers of primitive constraints on the individual participants.
- There may be many changes taking place within one atomic event, when viewed at the subatomic level.

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\ U\psi$.

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\ U\psi$.
- Program composition:

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\diamond\phi$, $\square\phi$, $\phi\ U\psi$.
- Program composition:
  1. They can be ordered, $\alpha; \beta$ (α is followed by β);

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\, U\psi$.
- Program composition:
  1. They can be ordered, $\alpha;\beta$ (α is followed by β);
  2. They can be iterated, a^* (apply a zero or more times);

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\, U\psi$.
- Program composition:
  1. They can be ordered, $\alpha;\beta$ (α is followed by β);
  2. They can be iterated, a^* (apply a zero or more times);
  3. They can be disjoined, $\alpha \cup \beta$ (apply either α or β);

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\ U\psi$.
- Program composition:
  1. They can be ordered, $\alpha; \beta$ (α is followed by β);
  2. They can be iterated, a^* (apply a zero or more times);
  3. They can be disjoined, $\alpha \cup \beta$ (apply either α or β);
  4. They can be turned into formulas
     $[\alpha]\phi$ (after every execution of α, ϕ is true);
     $\langle\alpha\rangle\phi$ (there is an execution of α, such that ϕ is true);

Dynamic Interval Temporal Logic

(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\ U\psi$.
- Program composition:
  1. They can be ordered, $\alpha; \beta$ (α is followed by β);
  2. They can be iterated, a^* (apply a zero or more times);
  3. They can be disjoined, $\alpha \cup \beta$ (apply either α or β);
  4. They can be turned into formulas
     $[\alpha]\phi$ (after every execution of α, ϕ is true);
     $\langle\alpha\rangle\phi$ (there is an execution of α, such that ϕ is true);
  5. Formulas can become programs, $\phi?$ (test to see if ϕ is true, and proceed if so).

Dynamic Event Structure

(8) a. Mary was sick today.
    b. My phone was expensive.
    c. Sam lives in Boston.

Dynamic Event Structure

(9) a. Mary was sick today.
    b. My phone was expensive.
    c. Sam lives in Boston.

We assume that a *state* is defined as a single frame structure (event), containing a proposition, where the frame is temporally indexed, i.e., $e^i \rightarrow \phi$ is interpreted as ϕ holding as true at time i. The frame-based representation from Pustejovsky and Moszkowicz (2011) can be given as follows:

Dynamic Event Structure

(10) $\boxed{\phi}^{i}_{e}$

(13) $\boxed{\phi}^{i}_{e}$

Propositions can be evaluated over subsequent states, of course, so we need an operation of concatenation, +, which applies to two or more event frames, as illustrated below.

Dynamic Event Structure

(16) $\boxed{\phi}_e^i$

Propositions can be evaluated over subsequent states, of course, so we need an operation of concatenation, +, which applies to two or more event frames, as illustrated below.

(17) $\boxed{\phi}_e^i + \boxed{\phi}_e^j = \boxed{\phi}_e^{[i,j]}$

(19) $\boxed{\phi}^{i}_{e}$

Propositions can be evaluated over subsequent states, of course, so we need an operation of concatenation, +, which applies to two or more event frames, as illustrated below.

(20) $\boxed{\phi}^{i}_{e} + \boxed{\phi}^{j}_{e} = \boxed{\phi}^{[i,j]}_{e}$

Semantic interpretations for these are:

Dynamic Event Structure

(22) $\boxed{\phi}^{i}_{e}$

Propositions can be evaluated over subsequent states, of course, so we need an operation of concatenation, +, which applies to two or more event frames, as illustrated below.

(23) $\boxed{\phi}^{i}_{e} + \boxed{\phi}^{j}_{e} = \boxed{\phi}^{[i,j]}_{e}$

Semantic interpretations for these are:

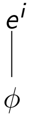
(24) a. $[\![\boxed{\phi}]\!]_{M,i} = 1$ iff $V_{M,i}(\phi) = 1$.

b. $[\![\boxed{\phi}\boxed{\phi}]\!]_{M,\langle i,j \rangle} = 1$ iff $V_{M,}(\phi) = 1$ and $V_{M,j}(\phi) = 1$, where $i < j$.

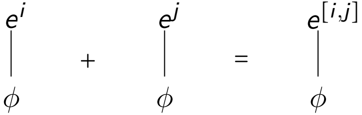
Dynamic Event Structure

(25)

(26)



Tree structure for event concatenation:



Labeled Transition System (LTS)

The dynamics of actions can be modeled as a Labeled Transition Systems (LTS).

Labeled Transition System (LTS)

The dynamics of actions can be modeled as a Labeled Transition Systems (LTS).

An LTS consists of a 3-tuple, $\langle S, Act, \rightarrow \rangle$, where

Labeled Transition System (LTS)

The dynamics of actions can be modeled as a Labeled Transition Systems (LTS).

An LTS consists of a 3-tuple, $\langle S, Act, \rightarrow \rangle$, where

(31) a. S is the set of states;
b. Act is a set of actions;
c. $\rightarrow$ is a total transition relation: $\rightarrow \subseteq S \times Act \times S$.

Labeled Transition System (LTS)

The dynamics of actions can be modeled as a Labeled Transition Systems (LTS).

An LTS consists of a 3-tuple, $\langle S, Act, \rightarrow \rangle$, where

(33) a. S is the set of states;
b. Act is a set of actions;
c. $\rightarrow$ is a total transition relation: $\rightarrow \subseteq S \times Act \times S$.

(34) $(e_1, \alpha, e_2) \in \rightarrow$

cf. Fernando (2001, 2013)

Labeled Transition System (LTS)

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition.

Labeled Transition System (LTS)

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition.

As a shorthand for

Labeled Transition System (LTS)

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition.

As a shorthand for

(37) a. $(e_1, \alpha, e_2) \in \rightarrow$, we will also use:

Labeled Transition System (LTS)

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition.

As a shorthand for

(38) a. $(e_1, \alpha, e_2) \in \rightarrow$, we will also use:

b. $e_1 \xrightarrow{\alpha} e_3$

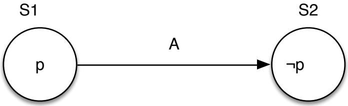
Labeled Transition System (LTS)

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition.

As a shorthand for

(39) a. $(e_1, \alpha, e_2) \in \rightarrow$, we will also use:

b. $e_1 \xrightarrow{\alpha} e_3$

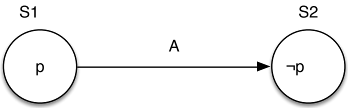


If reference to the state content (rather than state name) is required for interpretation purposes, then as shorthand for: $(\{\phi\}_{e_1}, \alpha, \{\neg\phi\}_{e_2}) \in \rightarrow$, we use:

Labeled Transition System (LTS)

If reference to the state content (rather than state name) is required for interpretation purposes, then as shorthand for: $(\{\phi\}_{e_1}, \alpha, \{\neg\phi\}_{e_2}) \in \rightarrow$, we use:

(41) $\boxed{\phi}_{e_1} \xrightarrow{\alpha} \boxed{\neg\phi}_{e_2}$



Temporal Labeled Transition System (TLTS)

With temporal indexing from a Linear Temporal Logic, we can define a Temporal Labeled Transition System (TLTS). For a state, e_1, indexed at time i, we say $e_1@i$.
$(\{\phi\}_{e_1@i}, \alpha, \{\neg\phi\}_{e_2@i+1}) \in \rightarrow_{(i,i+1)}$, we use:

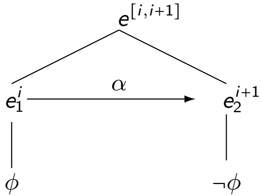
Temporal Labeled Transition System (TLTS)

With temporal indexing from a Linear Temporal Logic, we can define a Temporal Labeled Transition System (TLTS). For a state, e_1, indexed at time i, we say $e_1@i$.
$(\{\phi\}_{e_1@i}, \alpha, \{\neg\phi\}_{e_2@i+1}) \in \rightarrow_{(i,i+1)}$, we use:

(43) $\boxed{\phi}^{i}_{e_1} \xrightarrow{\alpha} \boxed{\neg\phi}^{i+1}_{e_2}$

Dynamic Event Structure

(44)



(45) Mary awoke from a long sleep.

Dynamic Event Structure

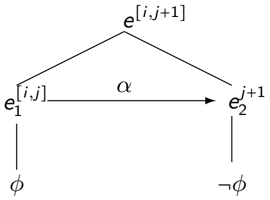
(47) Mary awoke from a long sleep.

The state of being asleep has a duration, $[i, j]$, who's valuation is gated by the waking event at the "next state", $j + 1$.

(49) Mary awoke from a long sleep.

The state of being asleep has a duration, $[i, j]$, who's valuation is gated by the waking event at the "next state", $j + 1$.

(50)



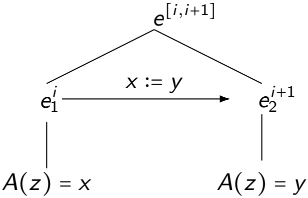
Simple First-order Transition

(51) $x := y$ (ν-transition)
"x assumes the value given to y in the next state."
$\langle M, (i, i+1), (u, u[x/u(y)])\rangle \models x := y$
iff $\langle M, i, u\rangle \models s_1 \wedge \langle M, i+1, u[x/u(y)]\rangle \models x = y$

Simple First-order Transition

(53) $x := y$ (ν-transition)
"x assumes the value given to y in the next state."
$\langle M, (i, i+1), (u, u[x/u(y)])\rangle \models x := y$
iff $\langle M, i, u\rangle \models s_1 \wedge \langle M, i+1, u[x/u(y)]\rangle \models x = y$

(54)



Processes

With a ν-transition defined, a *process* can be viewed as simply an iteration of basic variable assignments and re-assignments:

With a ν-transition defined, a *process* can be viewed as simply an iteration of basic variable assignments and re-assignments:

(56)