Semantic Theory
week 6 – Event semantics

Noortje Venhuizen

University of Groningen/Universität des Saarlandes

Summer 2015

A problem with verbs and adjuncts

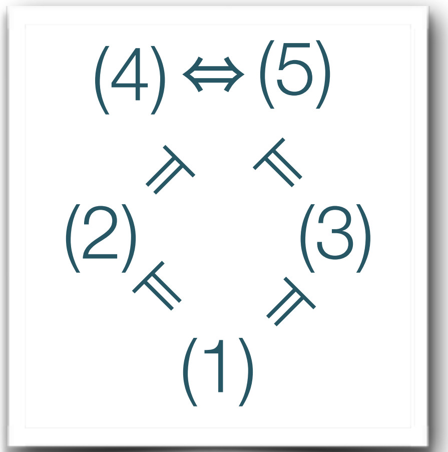
(1) *The gardener killed the baron* $\mapsto$ $kill_1(g', b')$ $kill_1 :: \langle e, \langle e, t \rangle \rangle$

(2) *The gardener killed the baron in the park* $\mapsto$ $kill_2(g', b', p')$ $kill_2 :: \langle e, \langle e, \langle e, t \rangle \rangle$

(3) *The gardener killed the baron at midnight* $\mapsto$ $kill_3(g', b', m')$ $kill_3 :: \langle e, \langle e, \langle e, t \rangle \rangle$

(4) *The gardener killed the baron at midnight in the park* $\mapsto$ $kill_4(g', b', m', p')$ $kill_4 :: \ldots$

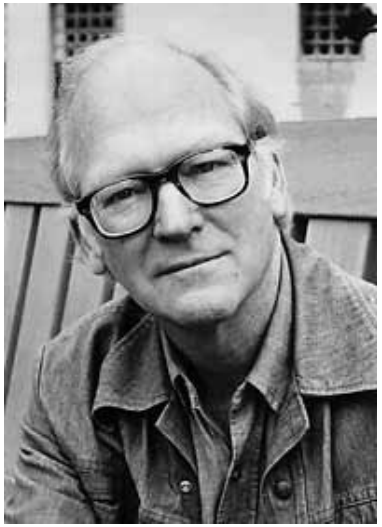
(5) *The gardener killed the baron in the park at midnight* $\mapsto$ $kill_4(g', b', p', m')$ $kill_5 :: \ldots$



Q: How to explain the systematic logical entailment relations between the different uses of "kill"?

Davidson's solution: verbs introduce events.

Verbs expressing events have an additional event argument, which is not realised at linguistic surface:



- kill ↦ λyλxλe(kill'(e,x,y)) :: ⟨e,⟨e,⟨e,t⟩⟩⟩ *arity = n+1*

Sentences denote sets of events:

- λyλxλe(kill'(e,x,y))(b')(g') ⇒$^{\beta}$ λe(kill'(e, g', b')) :: ⟨e,t⟩

Existential closure turns sets of events into truth conditions

- λP∃e(P(e)) :: ⟨⟨e,t⟩,t⟩
- λP∃e(P(e))(λe(kill'(e,g',b'))) ⇒$^{\beta}$ ∃e(kill'(e,g',b')) :: t

Davidson (1967, 1980)

Davisonian events and adjuncts

Adjuncts express two-place relations between events and the respective "circumstantial information": time, location, …

- at midnight $\mapsto$ $\lambda P \lambda e(P(e) \wedge \mathrm{time}(e,m))$ :: $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$
- at midnight $\mapsto$ $\lambda P \lambda e(P(e) \wedge \mathrm{location}(e,p))$ :: $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$

The gardener killed the baron at midnight in the park

$\mapsto \exists e\,(\mathrm{kill}(e, g', b') \wedge \mathrm{time}(e, m) \wedge \mathrm{location}(e, p))$

$\Leftrightarrow \exists e\,(\mathrm{kill}(e, g', b') \wedge \mathrm{location}(e, p) \wedge \mathrm{time}(e, m))$

$\models \exists e\,(\mathrm{kill}(e, g', b') \wedge \mathrm{time}(e, m))$

$\models \exists e\,(\mathrm{kill}(e, g', b') \wedge \mathrm{location}(e, p))$

$\models \exists e\,(\mathrm{kill}(e, g', b'))$

Compositional derivation of event-semantic representations

the gardener killed the baron

$\lambda x_e \lambda y_e \lambda e_e [\text{ kill}(e, y, x)](b')(g') \Rightarrow^{\beta} \lambda e [\text{ kill}(e, g', b')]$

α

… at midnight

$\lambda F_{\langle e,t \rangle} \lambda e_e [F(e) \wedge \text{time}(e, m')](\lambda e' [\text{ kill}(e', g', b')]) \Rightarrow^{\beta} \lambda e [\text{ kill}(e, g, b) \wedge \text{time}(e,m')]$

… in the park

$\lambda F_{\langle e,t \rangle} \lambda e_e [F(e) \wedge \text{location}(e, p')] (\lambda e'[\text{kill}(e', g', b') \wedge \text{time}(e', m')]) \Rightarrow^{\beta}$

$\lambda e [\text{kill}(e, g', b') \wedge \text{time}(e, m') \wedge \text{location}(e, p')]$

Existential closure

$\lambda P_{\langle e,t \rangle} \exists e(P(e))(\lambda e'(K \wedge T \wedge L) \Rightarrow^{\beta} \exists e [\text{ kill}(e, g', b') \wedge \text{time}(e, m') \wedge \text{location}(e, p')]$

Model structures with events

To interpret events, we need enriched *ontological* information

Ontology: The area of philosophy identifying and describing the basic “categories of being” and their relations.

A model structure with events is a triple $M = \langle U, E, V \rangle$, where

- U is a set of “standard individuals” or “objects”
- E is a set of events
- $U \cap E = \varnothing$,
- V is an interpretation function like in first order logic

Sorted (first-order) logic

A variable assignment g assigns individuals (of the correct sort-specific domain) to variables:

- $g(x) \in U$ for $x \in VAR_U$ $VAR_U = \{ x, y, z, \ldots , x_1, x_2, \ldots \}$ (Object variables)
- $g(e) \in E$ for $e \in VAR_E$ $VAR_E = \{ e, e', e'', \ldots, e_1, e_2, \ldots \}$ (Event variables)

Quantification ranges over sort-specific domains:

- $[\![\exists x\, \Phi]\!]^{M,g} = 1$ iff there is an $a \in U$ such that $[\![\Phi]\!]^{M,g[x/a]} = 1$
- $[\![\exists e\, \Phi]\!]^{M,g} = 1$ iff there is an $a \in E$ such that $[\![\Phi]\!]^{M,g[e/a]} = 1$
- (universal quantification analogous)

Advantages of Davidsonian events

- [x] Intuitive representation and semantic construction for adjuncts
- [ ] Uniform treatment of verb complements
- [ ] Uniform treatment of adjuncts and post-nominal modifiers
- [ ] Coherent treatment of tense information
- [ ] Highly compatible with analysis of semantic roles

Uniform treatment of verb complements

(1) *Bill saw an elephant*

$\mapsto \exists e\ \exists x\ (\text{see}(e, b', x) \wedge \text{elephant}(x))$ — see :: $\langle e,\langle e,\langle e,t\rangle\rangle$

(2) *Bill saw an accident*

$\mapsto \exists e\ \exists e'\ (\text{see}(e, b, e') \wedge \text{accident}(e'))$ — see :: $\langle e,\langle e,\langle e,t\rangle\rangle$

(3) *Bill saw the children play*

$\mapsto \exists e\ \exists e'\ (\text{see}(e, b, e') \wedge \text{play}(e', \text{the-children}))$ — see :: $\langle e,\langle e,\langle e,t\rangle\rangle$

Uniform treatment of adjuncts and post-nominal modifiers

Treatment of adjuncts as predicate modifiers, analogous to attributive adjectives:

- red ↦ λF λx [F(x) ∧ red*(x)] ⟨⟨e,t⟩,⟨e,t⟩⟩
- in the park ↦ λF λe [F(e) ∧ location(e, park)] ⟨⟨e,t⟩,⟨e,t⟩⟩

(1) *The murder in the park…*

↦ λFλe[F(e) ∧ location(e, park)] (λe [murder(e)])

(2) The fountain in the park ….

↦ λFλx[F(x) ∧ location(x, park)] (λy [fountain(y)])

Advantages of Davidsonian events

- ☑ Intuitive representation and semantic construction for adjuncts
- ☑ Uniform treatment of verb complements
- ☑ Uniform treatment of adjuncts and post-nominal modifiers
- ☐ Coherent treatment of tense information
- ☐ Highly compatible with analysis of semantic roles

Classical Tense Logic

- *John walks* walk(john)
- *John walked* P(walk(john))
- *John will walk* F(walk(john))

Syntax like in first-order logic, plus

- if Φ is a well-formed formula, then PΦ, FΦ, HΦ, GΦ are also well-formed formulae.

Φ has always been the case

Φ is always going to be the case

Φ happened in the past

Φ will happen in the future

Classical Tense Logic (cont.)

Tense model structures are quadruples $M = \langle U, T, <, V \rangle$ where

- U is a non-empty set of individuals (the “universe”)
- T is a non-empty sets of points in time
- $U \cap T = \varnothing$
- $<$ is a linear order on T
- V is a value assignment function, which assigns to every non-logical constant α a function from T to appropriate denotations of α

$$[\![P\Phi]\!]^{M,\,t,\,g} = 1 \text{ iff there is a } t' < t \text{ such that } [\![\Phi]\!]^{M,\,t',\,g} = 1$$

$$[\![F\Phi]\!]^{M,\,t,\,g} = 1 \text{ iff there is a } t' > t \text{ such that } [\![\Phi]\!]^{M,\,t',\,g} = 1$$

Temporal Relations and Events

(1) *The door opened, and Mary entered the room.*

(2) *John arrived. Then Mary left.*

(3) *Mary left, before John arrived.*

(4) *John arrived. Mary had left already.*

Q: How to formalize temporal relations *between events*?

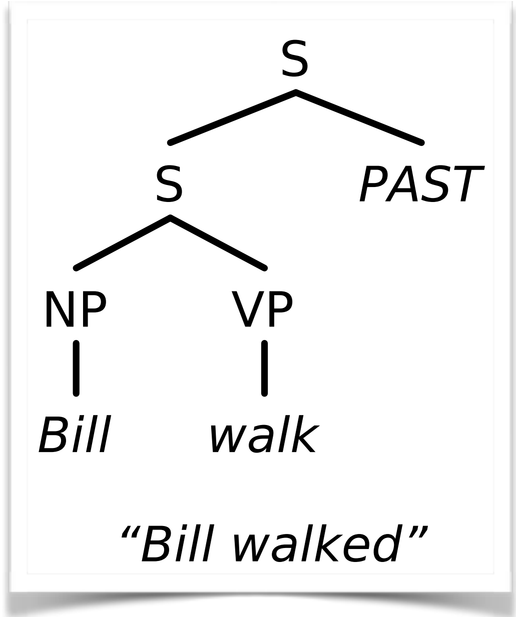
Temporal Event Structure

A model structure with events and temporal precedence is defined as $M = \langle U, E, <, e_u, V \rangle$, where

- $U \cap E = \varnothing$,
- $< \subseteq E \times E$ is an asymmetric relation (temporal precedence)
- $e_u \in E$ is the utterance event
- V is an interpretation function like in standard FOL
- Overlapping events: $e \cdot e'$ iff neither $e < e'$ nor $e' < e$

Tense in Semantic Construction

We can represent inflection as an abstract tense operator reflecting the temporal location of the reported event relative to the utterance event.



PAST $\mapsto \lambda P.\exists e\ [P(e) \wedge e < e_u] : \langle\langle e, t\rangle, t\rangle$

PRES $\mapsto \lambda P.\exists e\ [P(e) \wedge e \cdot e_u] : \langle\langle e, t\rangle, t\rangle$

Tense in Semantic Construction

Standard function application results in integration of temporal information and binding of the event variable (i.e., replacing E-CLOS):

- walk $\mapsto$ λx λe [walk(e, x)]
- Bill walk $\mapsto$ λx λe [walk(e, x)](b') $\Rightarrow^{\beta}$ λe [walk(e, b')]
- Bill walk PAST
  $\mapsto$ λE ∃e [E(e) ∧ e < e_u](λe' [walk(e', b)])
  $\Rightarrow^{\beta}$ ∃e [λe' [walk(e', b)](e) ∧ e < e_u]
  $\Rightarrow^{\beta}$ ∃e [walk(e, b) ∧ e < e_u]