Multimodal Semantics for Affordances and Actions

Lecture 2: Modeling Human-Object Interactions

James Pustejovsky and Nikhil Krishnaswamy

ESSLLI 2022 Summer School

Galway, Ireland

August 8-19, 2022

Extending Qualia to Modeling Affordances

The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill. It implies the complementarity of the animal and the environment. (J. J. Gibson, 1979/1986)

- Affordance: a correlation between an agent who acts on an object with a systematic or prototypical effect.
- Gibson (1979), Turvey (1992), Stoffregen (2003) Steedman (2002), Sahin et al (2007), Krippendorff (2010);

Major Issues in Interpreting Affordances

- Do affordances exist independently of perception? Perceivers are aware of action possibilities, but the affordance has an existence independent of that perception and do not arise as a consequence of mental operations. They are action-referential properties of the environment that may or may not be perceived.
- Are affordances action related only or more general? We either limit the term to action-affordances or include state-affordances as well.

Action Domain Types

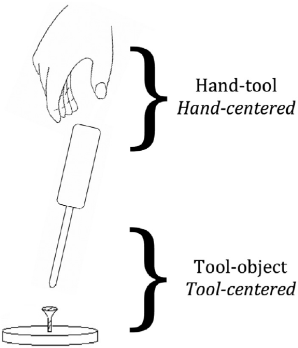
Osiuraka et al. (2017)

- Non-tool Use: actions consisting in moving one object from one location to another (i.e., object transport). The mere action of grabbing an object (i.e., reach-grasping) is also be included within this category.
- Tool Use: notion of tool encompasses a wide range of objects, whether manufactured or not, manipulable or not.

Reference Frame Types

- Hand-tool Interface: based on the agent's biomechanical and morphological characteristics. For instance, a hammer is graspable by a human adult but not by a baby. Thus, the interface is centered on the agent.
- Tool-object Interface: independent of the agent's characteristics. The relationship is centered on objects external to the agent and the interaction is made possible because of the compatibility between the characteristics of the tool and the object.

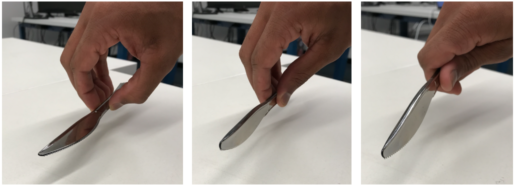
Reference Frames in Affordances



Learning Affordances for Different Objects - Grasping 1/2

Affordances involve:

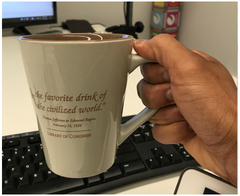
- Human-object configurations;
- An action;
- A resulting process or state.



Learning Affordances for Different Objects - Grasping 2/2

Affordances involve:

- Human-object configurations;
- An action;
- A resulting process or state.



Affordances are event-functions

Event-denoting expressions

- Verbal event predicates:
  - The missile sank the ship.
  - The car hit the pedestrian.
- Nominal events:
  - The meal was after the speech.
  - An alarm went off during the workshop.
- Nominalizations:
  - The explosion occurred at noon.
  - The arrival of the train was late.
- Adjectival event predicates:
  - Mary closed the open door.
  - They observed the moving truck.
- Prepositional Phrase predicates:
  - John is on board.
  - Sophie is in the house.

- Stage-level Agentive Nominals :
    - The plaza is filled with pedestrians and merchants.
    - A jogger was injured during the crash.
    - A policeman questioned the witness.
- Occupational Agentive Nominals :
    - A policeman questioned the witness.
    - The plaza is filled with pedestrians and merchants.
- Resultative Nominals:
    - Debris covered the field for yards.
    - Sophie placed the laundry in the basket / on the bed.
- Artifactual Nominals:
    - This is a good knife. The coffee is ready.
    - Sergei is at the piano.

Semantics of Function and Purpose

There are two levels of accessibility that can be identified in a Telic role value, as illustrated below.

(1) a. local modality: the conditions under which the activity can be performed on the object;
b. global modality: what is done with the object, and the resulting state.

Hidden events and the Telic

- The Telic relation (T) encodes information about the intended use or function of an object.

Hidden events and the Telic

- The Telic relation (T) encodes information about the intended use or function of an object.
- It expresses the relation that allows us to grasp what an entity is by knowing what it is used for.

Hidden events and the Telic

- The Telic relation (T) encodes information about the intended use or function of an object.
- It expresses the relation that allows us to grasp what an entity is by knowing what it is used for.
- It encodes a potential activity of the object.

Hidden events and the Telic

- The Telic relation (T) encodes information about the intended use or function of an object.
- It expresses the relation that allows us to grasp what an entity is by knowing what it is used for.
- It encodes a potential activity of the object.
- First systematic mention of Telic in Pustejovsky and Anick (1988) as hidden event.

Telic Qualia Role as an Affordance Structure

Motivation for Qualia relations comes from the idea that there is a *hidden event* in the lexical representation associated with nouns denoting objects made for a particular purpose:

(2)
a. a door is for walking through
b. a window is for seeing through
c. a book is for reading
d. a beer is for drinking
e. a cake is for eating
f. a car is for driving
g. a table is for putting things on
h. a desk is for working on
i. a pen is for writing with

Habitats and Simulations

Pustejovsky (2013)

- Habitat: a representation of an object situated within a partial minimal model; Enhancements of the qualia structure.
- With multi-dimensional affordances that determine how habitats are deployed and how they modify or augment the context.
- Compositional combinations of procedural (simulation) and operational (selection, specification, refinement) knowledge.
- A habitat:
  - embeds;
  - orients;
  - positions.

Natural Predicate Types

Predicates formed with Natural Entities as arguments:

1. *fall*: $e_N \rightarrow t$
2. *touch*: $e_N \rightarrow (e_N \rightarrow t)$
3. *be under*: $e_N \rightarrow (e_N \rightarrow t)$

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

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

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

Artifactual Entity Types

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

1. Artifact Entity: $x : e_N \otimes_a \sigma$
   x exists because of event σ
2. Functional Entity: $x : e_N \otimes_t \tau$
   the purpose of x is τ
3. Functional Artifactual Entity: $x : (e_N \otimes_a \sigma) \otimes_t \tau$
   x exists because of event σ for the purpose τ

a. *beer*: $(liquid \otimes_a brew) \otimes_t drink$

b. *knife*: $(phys \otimes_a make) \otimes_t cut$

c. *house*: $(phys \otimes_a build) \otimes_t live\_in$

Artifactual Predicate Types

Predicates formed with Artifactual Entities as arguments:

1. *spoil*: $e_N \otimes_t \tau \rightarrow t$
2. *fix*: $e_N \otimes_t \tau \rightarrow (e_N \rightarrow t)$

a. $\lambda x{:}\, e_A[\ spoil(\mathrm{x})]$

b. $\lambda y{:}\, e_A \lambda x{:}\, e_N[\ fix(\mathrm{x,y})]$

- The beer spoiled.
- Mary fixed the watch.

Complex Entity Types (Dot Objects)

Pustejovsky (1994)

When a single word or phrase has the ability to appear in selected contexts that are contradictory in type specification.

If a lexical expression, α, where $\sigma \sqcap \tau = \perp$:

1. [___] $_\sigma$ X
2. [___] $_\tau$ Y

are both well-formed predications, then α is a dot object (complex type).

Entities formed from the Naturals and Artifactuals by a product type between the entities, i.e., the dot, $\bullet$.

1. a. Mary doesn't believe the book.
   b. John sold his book to Mary.
2. a. The exam started at noon.
   b. The students could not understand the exam.

$$(3)\ \begin{bmatrix} \textit{cake} \\ \text{QUALIA} = \begin{bmatrix} \text{F} = \textbf{food} \\ \text{T} = \textbf{eat(human,food)} \end{bmatrix} \end{bmatrix}$$

$$(4)\ \begin{bmatrix} \textit{pen} \\ \text{QUALIA} = \begin{bmatrix} \text{F} = \textbf{tool} \\ \text{T} = \textbf{write\_with} \end{bmatrix} \end{bmatrix}$$

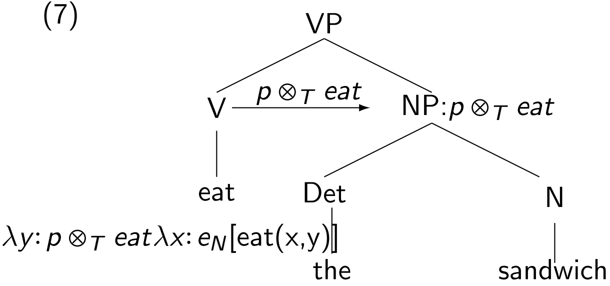
$$(5)\ \begin{bmatrix} \textit{singer} \\ \text{QUALIA} = \begin{bmatrix} \text{F} = \textbf{human} \\ \text{T} = \textbf{sing(human, song)} \end{bmatrix} \end{bmatrix}$$

Conventional Type Composition

(6)

- **S**:t — $grab(w, c)$
  - **DP**
    - a_woman — $w : human$
  - **VP**:$animate \to t$ — $\lambda x[grab(x, c)]$
    - **V**
      - grabbed — $phys \to (animate \to t)$ — $\lambda y \lambda x[grab(x, y)]$
    - **DP**
      - a_cup — $c : phys$

(7)



VP
V
$p \otimes_T eat$
NP:$p \otimes_T eat$
eat
Det
N
$\lambda y{:}\, p \otimes_T eat\, \lambda x{:}\, e_N[\mathrm{eat}(x,y)]$
the
sandwich

Habitats and Simulations Pustejovsky (2013)

- Habitat: a representation of an object situated within a partial minimal model; Enhancements of the qualia structure.
- With multi-dimensional affordances that determine how habitats are deployed and how they modify or augment the context.
- Compositional combinations of procedural (simulation) and operational (selection, specification, refinement) knowledge.
- A habitat:
  - embeds;
  - orients;
  - positions.

Teleotopology

- The function of space: the actions associated with a region or an object (inherently or opportunistically), i.e., Telic role values.
- The space of function: the regions defined by the Telic actions performed by an agent, or supervenient on the Telic state of an artifact, teleotopology.

(8) $\mathcal{C} \rightarrow [\pi]\mathcal{R}$

The TELIC of *sandwich*:

(9) $$\lambda x \begin{bmatrix} \textbf{sandwich} \\ \text{AS} = \begin{bmatrix} \text{ARG1} = x : e \end{bmatrix} \\ \text{QS} = \begin{bmatrix} \text{F} = phys(x) \\ \text{T} = \lambda y \lambda e[\mathcal{C} \rightarrow [eat(e,y,x)]\mathcal{R}_{eat}(x)]] \\ \text{A} = \exists z[make(z,x)] \end{bmatrix} \end{bmatrix}$$

Objects can have Inherent Functions

Telic Qualia Role

- For a noun such as *sandwich*, we have a set of contexts, $\mathcal{C}$, for the object denoted by x, under which, when an individual y eats x, there is a resulting state of nourishment, which we will notate as $\mathcal{R}_{eat}$.
- This says that if the context is satisfied, then every eating of that substance will result in a "nourishing."

$$\lambda x[\text{FORMAL}(x) = phys(x) \wedge$$
$$\text{TELIC}(x) = \lambda y \lambda e[\mathcal{C} \rightarrow [eat(e, y, x)]\mathcal{R}_{eat}(x)]]$$

$$
\text{(10)}\quad \lambda x \exists y \begin{bmatrix} \textbf{chair} & & \\ \text{AS} = \begin{bmatrix} \text{ARG1} = x : e \end{bmatrix} & & \\ \text{QS} = \begin{bmatrix} \text{F} = \mathit{phys}(x) \\ \text{T} = \lambda z, e[\mathit{sit\_in}(e, z, x)] \end{bmatrix} & & \end{bmatrix}
$$

$$
\lambda x \begin{bmatrix} \textbf{chair}_{hab} \\ \\ \text{F} = [\mathit{phys}(x), \mathit{on}(x, y_1), \mathit{in}(x, y_2), \mathit{orient}(x, \mathit{up})] \\ \text{C} = [\mathit{seat}(x_1), \mathit{back}(x_2), \mathit{legs}(x_3), \mathit{clear}(x_1)] \\ \text{T} = \lambda z \lambda e[\mathcal{C} \rightarrow [\mathit{sit}(e, z, x)]\mathcal{R}_{sit}(x)] \\ \text{A} = [\mathit{made}(e', w, x)] \end{bmatrix}
$$

Lexical Semantic Relations

Verbal Subtypes: Means

(11) CUT

saw ax slice

(12) a. Mary **cut** the {wood/ tree/ bread}.

b. Mary **sawed** the wood (='cut with a saw').

c. Mary **axed** the tree (='cut with an ax').

d. Mary **sliced** the bread (='cut with a knife').

- A category B is a "means subtype" of a category A, $B \sqsubseteq_m A$, if and only if B has a more specific subtype for the INSTRUMENT semantic role for category A.

(13) a. $\textsc{saw} \sqsubseteq \textsc{instrument} \Longrightarrow \textsc{saw} \sqsubseteq_m \textsc{cut}$

b. $\textsc{ax} \sqsubseteq \textsc{instrument} \Longrightarrow \textsc{ax} \sqsubseteq_m \textsc{cut}$

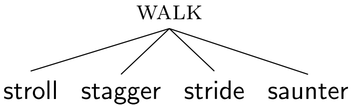
c. $\textsc{knife} \sqsubseteq \textsc{instrument} \Longrightarrow \textsc{slice} \sqsubseteq_m \textsc{cut}$

Manner Subtyping for Verbs

Verbal Subtypes: Manner

- A category B is a "manner subtype" (troponym) of a category A, $B \sqsubseteq_t A$, if and only if B specifies a particular manner in which to perform A.
- The motion verbs *stroll*, *stagger*, *stride*, and *saunter* are troponyms of *walk*, because they each denote a certain manner of walking.

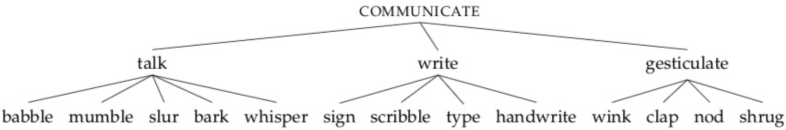
(14)



Manner Subtyping for Verbs

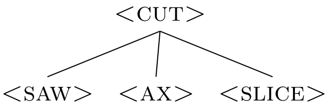
Mixing Means and Manner

(30)



Objects as Action Modalities
Nominal Subtypes: Means

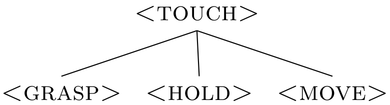
(15)



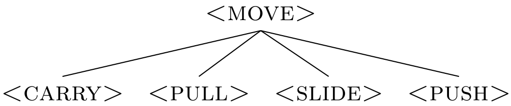
(16) a. This object is **cuttable** (the {wood/ tree/ bread}).
b. The wood is **sawable** (=‘cuttable with a saw’).
c. The tree is **axable** (=‘cuttable with an ax’).
d. The bread is **sliceable** (=‘cuttable with a knife’).

Action Modalities as Types

(17)

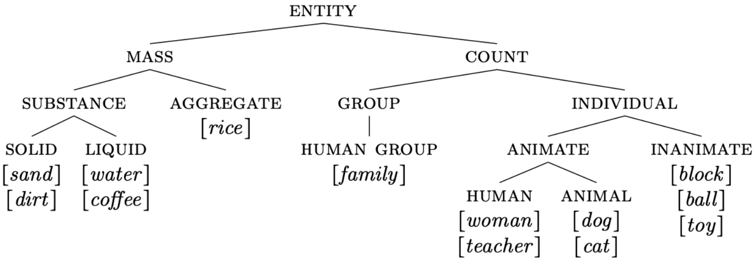


(18)

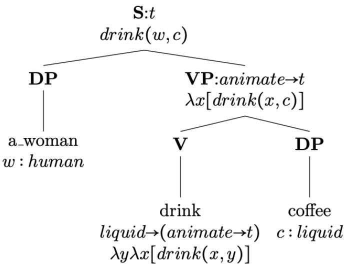


Categorical Knowledge of Objects

Pustejovsky and Batiukova (2019)



Conventional Type Composition



- How does a human drink a liquid?
- Where is the liquid located?
- How is the container oriented?

Categorical and Modal Object Knowledge

Qualia Structure (Pustejovsky, 1995)

- *Formal*: encoding taxonomic information about the lexical item (IS-A relation);
- *Constitutive*: encoding information on the parts and constitution of an object (PART-OF or MADE-OF relation);
- *Telic*: encoding information on purpose and function (USED-FOR or FUNCTIONS-AS relation);
- *Agentive*: encoding information about the origin of the object (CREATED-BY relation).

Telic Qualia Role as part of Affordance Structure

Motivation for Qualia relations comes from the idea that there is a *hidden event* in the lexical representation associated with nouns denoting objects made for a particular purpose:

(19) a. a door is for walking through
b. a window is for seeing through
c. a book is for reading
d. a beer is for drinking
e. a cake is for eating
f. a car is for driving
g. a table is for putting things on
h. a desk is for working on
i. a pen is for writing with

Habitats and Simulations

Pustejovsky (2013)

- Habitat: a representation of an object situated within a partial minimal model; Enhancements of the qualia structure.
- With multi-dimensional affordances that determine how habitats are deployed and how they modify or augment the context.
- Compositional combinations of procedural (simulation) and operational (selection, specification, refinement) knowledge.
- A habitat:
  - embeds;
  - orients;
  - positions.

Different Habitats for Object Use

Top: *Spoon* allowing holding (left) and stirring (right).
Bottom: *Knife* allowing spreading (left) and cutting (right)

Habitat-Affordance Pairs

If **Habitat** *then* **Action**

spoon

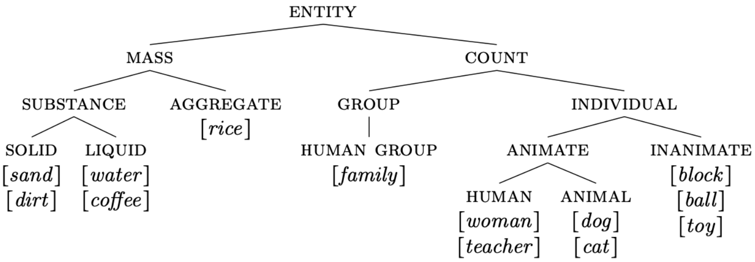
(20) a. If *spoon's concavity is vertical*, then it can *support containment of a substance*;
b. If *spoon's major axis is vertical*, then it can *support mixing*.

knife

(21) a. If *knife's zero convexity (sheet) is horizontal*, then it can *support spreading of a substance*;
b. If *knife's zero convexity (sheet) is vertical*, then it can *support cutting or separating*.

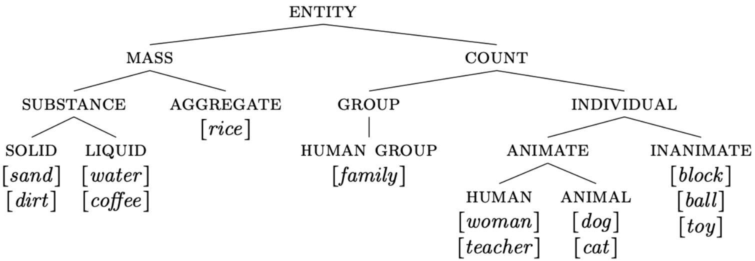
Categorical Knowledge of Objects

Pustejovsky and Batiukova (2019)



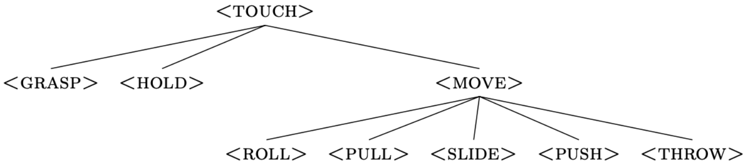
Modal Action Hierarchy

Entities as Types



Modal Action Hierarchy

Refactoring Entity Types as Affordances



Habitats are Contexts of Use

Pustejovsky (2013)

- $\lambda x \begin{bmatrix} \textbf{chair} \\ \text{AS} = \begin{bmatrix} \text{ARG1} = x : e \end{bmatrix} \\ \text{QS} = \begin{bmatrix} \text{F} = phys(x) \\ \text{T} = \lambda z, e[sit\_in(e,z,x)] \end{bmatrix} \end{bmatrix}$
- $\mathcal{H} \rightarrow [\pi]\mathcal{R}$: if the habitat $\mathcal{H}$ is satisfied, then every time the activity π is performed, the resulting state $\mathcal{R}$ will occur.
- $\lambda\mathcal{H}\lambda x \begin{bmatrix} \textbf{chair} \\ \text{F} = [phys(x), on(x,y_1), in(x,y_2), orient(x,up)] \\ \text{C} = [seat(x_1), back(x_2), legs(x_3), clear(x_1)] \\ \text{T} = \lambda z\lambda e[\mathcal{H} \rightarrow [sit(e,z,x)]\mathcal{R}_{sit}(x)] \\ \text{A} = [made(e', w, x)] \end{bmatrix}$

Visual Object Concept Modeling Language (VoxML)

Pustejovsky and Krishnaswamy (2016)

- Encodes afforded behaviors for each object
  - Gibsonian: afforded by object structure
    - grasp, move, lift, etc.
  - Telic: goal-directed, purpose-driven
    - drink from, read, etc.
- Voxeme
  - Object Geometry: Formal object characteristics in R3 space
  - Habitat: Conditioning environment affecting object **affordances** (behaviors attached due to object structure or purpose);
  - Affordance Structure:
    - What can one do to it
    - What can one do with it
    - What does it enable

VoxML Attributes

LEX	OBJECT's lexical information
TYPE	OBJECT's geometrical typing
HABITAT	OBJECT's habitat for actions
AFFORD_STR	OBJECT's affordance structure
EMBODIMENT	OBJECT's agent-relative embodiment

The LEX attribute contains the subcomponents PRED, the predicate lexeme denoting the object, and TYPE, the object's type according to Generative Lexicon.

Object Geometry for $[[\textsc{cup}]]$

$$(22)\quad \begin{bmatrix} \textbf{cup} \\ \text{TYPE} = \begin{bmatrix} \text{HEAD} = \textbf{cylindroid[1]} \\ \text{COMPONENTS} = \textbf{surface, interior} \\ \text{CONCAVITY} = \textbf{concave} \\ \text{ROTATIONAL\_SYMMETRY} = \{Y\} \\ \text{REFLECTION\_SYMMETRY} = \{XY, YZ\} \end{bmatrix} \end{bmatrix}$$

Various habitats identified with $[[\textsc{cup}]]$

$$
(23)\quad \begin{bmatrix} \textbf{cup} \\ \textsc{habitat} = \begin{bmatrix} \textsc{Intrinsic} = [2]\begin{bmatrix} \textsc{constr} = \{Y > X, Y > Z\} \\ \textsc{up} = align(Y, \mathcal{E}_Y) \\ \textsc{top} = top(+Y) \end{bmatrix} \\ \textsc{Extrinsic} = [3]\begin{bmatrix} \textsc{up} = align(Y, \mathcal{E}_{\perp Y}) \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

$$
(24)\quad \begin{bmatrix} \textbf{cup} & \\ \text{AFF\_STR} = & \begin{bmatrix} A_1 = H_{[2]} \rightarrow [put(x, on([1]))]support([1], x) \\ A_2 = H_{[2]} \rightarrow [put(x, in([1]))]contain([1], x) \\ A_3 = H_{[2]} \rightarrow [grasp(x, [1])]hold(x, [1]) \\ A_4 = H_{[3]} \rightarrow [roll(x, [1])]\mathcal{R} \end{bmatrix} \end{bmatrix}
$$

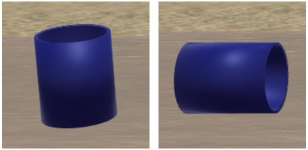


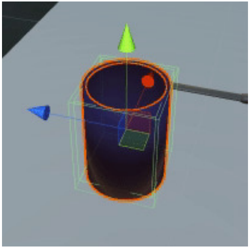
Figure: Cup in different habitats allowing sliding and holding (left) and rolling (right).

Semantic type, habitat, and affordances

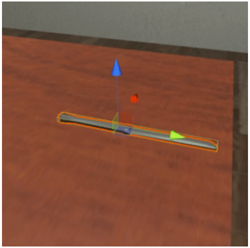
$$
\left[\begin{array}{l}
\textbf{cup} \\
\text{LEXICAL} = \left[\begin{array}{l} \text{PREDICATE} = \textbf{cup} \\ \text{TYPE} = \textbf{physobj} \bullet \textbf{artifact} \end{array}\right] \\
\text{TYPE} = \left[\begin{array}{l} \text{HEAD} = \textbf{cylindroid[1]} \\ \text{COMPONENTS} = \textbf{surface, interior} \\ \text{CONCAVITY} = \textbf{concave} \\ \text{ROTATIONAL\_SYMMETRY} = \{Y\} \\ \text{REFLECTION\_SYMMETRY} = \{XY, YZ\} \end{array}\right] \\
\text{HABITAT} = \left[\begin{array}{l} \text{INTRINSIC} = [2]\left[\begin{array}{l} \text{CONSTR} = \{Y > X, Y > Z\} \\ \text{UP} = align(Y, \mathcal{E}_Y) \\ \text{TOP} = top(+Y) \end{array}\right] \\ \text{EXTRINSIC} = [3]\left[\text{UP} = align(Y, \mathcal{E}_{\perp Y}) \right] \end{array}\right] \\
\text{AFF\_STR} = \left[\begin{array}{l} \text{A}_1 = H_{[2]} \rightarrow [put(x, on([1]))]support([1], x) \\ \text{A}_2 = H_{[2]} \rightarrow [put(x, in([1]))]contain([1], x) \\ \text{A}_3 = H_{[2]} \rightarrow [grasp(x, [1])]hold(x, [1]) \\ \text{A}_4 = H_{[3]} \rightarrow [roll(x, [1])]\mathcal{R} \end{array}\right] \\
\text{EMBOD} = \left[\begin{array}{l} \text{SCALE} = \textbf{<agent} \\ \text{MOVABLE} = \textbf{true} \end{array}\right]
\end{array}\right]
$$

VoxML - cup

$$
\begin{bmatrix}
\textbf{cup} & \\
\text{LEX} = & \begin{bmatrix} \text{PRED} = \textbf{cup} \\ \text{TYPE} = \textbf{physobj, artifact} \end{bmatrix} \\
\text{TYPE} = & \begin{bmatrix} \text{HEAD} = \textbf{cylindroid[1]} \\ \text{COMPONENTS} = \textbf{surface, interior} \\ \text{CONCAVITY} = \textbf{concave} \\ \text{ROTATSYM} = \{Y\} \\ \text{REFLECTSYM} = \{XY, YZ\} \end{bmatrix} \\
\text{HABITAT} = & \begin{bmatrix} \text{INTR} = [2]\begin{bmatrix} \text{CONSTR} = \{Y > X, Y > Z\} \\ \text{UP} = align(Y, \mathcal{E}_Y) \\ \text{TOP} = top(+Y) \end{bmatrix} \\ \text{EXTR} = [3]\begin{bmatrix} \text{UP} = align(Y, \mathcal{E}_{\perp Y}) \end{bmatrix} \end{bmatrix} \\
\text{AFFORD\_STR} = & \begin{bmatrix} A_1 = H_{[2]} \rightarrow [put(x, on([1]))]support([1], x) \\ A_2 = H_{[2]} \rightarrow [put(x, in([1]))]contain([1], x) \\ A_3 = H_{[2]} \rightarrow [grasp(x, [1])] \\ A_4 = H_{[3]} \rightarrow [roll(x, [1])] \end{bmatrix} \\
\text{EMBODIMENT} = & \begin{bmatrix} \text{SCALE} = \textbf{<agent} \\ \text{MOVABLE} = \textbf{true} \end{bmatrix}
\end{bmatrix}
$$

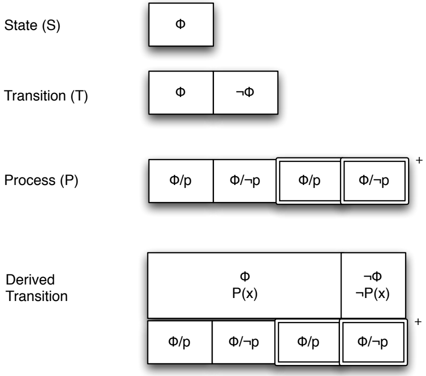


$$
\begin{bmatrix}
\textbf{knife} \\
\text{LEX} = \begin{bmatrix} \text{PRED} = \textbf{knife} \\ \text{TYPE} = \textbf{physobj, artifact} \end{bmatrix} \\
\text{TYPE} = \begin{bmatrix} \text{HEAD} = \textbf{rectangular\_prism[1]} \\ \text{COMPONENTS} = \textbf{handle[2], blade} \\ \text{CONCAVITY} = \textbf{flat} \\ \text{ROTATSYM} = nil \\ \text{REFLECTSYM} = \{XY\} \end{bmatrix} \\
\text{HABITAT} = \begin{bmatrix} \text{INTR} = {\scriptstyle [3]} \begin{bmatrix} \text{CONSTR} = \{X > Y, X \gg Z\} \\ \text{FRONT} = front(+X) \end{bmatrix} \\ \text{EXTR} = \ldots \end{bmatrix} \\
\text{AFFORD\_STR} = \begin{bmatrix} A_1 = H_{[3]} \rightarrow [grasp(x, [1])] \\ A_2 = H_{[3]} \rightarrow [grasp(x, [2]) \rightarrow grasp(x, [1])] \end{bmatrix} \\
\text{EMBODIMENT} = \begin{bmatrix} \text{SCALE} = \textbf{<agent} \\ \text{MOVABLE} = \textbf{true} \end{bmatrix}
\end{bmatrix}
$$



$$
\left[\begin{array}{l}
\textbf{slide} \\
\text{LEX} = \left[\begin{array}{l} \text{PRED} = \textbf{slide} \\ \text{TYPE} = \textbf{process} \end{array}\right] \\
\text{TYPE} = \left[\begin{array}{l}
\text{HEAD} = \textbf{process} \\
\text{ARGS} = \left[\begin{array}{l} \text{A}_1 = \textbf{x:agent} \\ \text{A}_2 = \textbf{y:physobj} \\ \text{A}_3 = \textbf{z:physobj} \end{array}\right] \\
\text{BODY} = \left[\begin{array}{l} \text{E}_1 = grasp(x,y) \\ \text{E}_2 = [while(hold(x,y), \\ \qquad while(EC(y,z), move(x,y)))] \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

$$
\left[\begin{array}{l}
\textbf{put} \\
\text{LEX} = \left[\begin{array}{l} \text{PRED} = \textbf{put} \\ \text{TYPE} = \textbf{transition\_event} \end{array}\right] \\
\text{TYPE} = \left[\begin{array}{l}
\text{HEAD} = \textbf{transition} \\
\text{ARGS} = \left[\begin{array}{l} \text{A}_1 = \textbf{x:agent} \\ \text{A}_2 = \textbf{y:physobj} \\ \text{A}_3 = \textbf{z:location} \end{array}\right] \\
\text{BODY} = \left[\begin{array}{l} \text{E}_1 = grasp(x,y) \\ \text{E}_2 = [while((\neg at(y,z) \wedge hold(x,y)), move(x,y)] \\ \text{E}_3 = [at(y,z) \rightarrow ungrasp(x,y)] \end{array}\right]
\end{array}\right]
\end{array}\right]
$$



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: $\bigcirc\phi$, $\diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.

(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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- 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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- 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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- 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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- 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);

(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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- 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: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\mathcal{U}\psi$.
- 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

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

(26) 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

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

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

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.

Dynamic Event Structure

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

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.

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

Dynamic Event Structure

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

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.

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

Semantic interpretations for these are:

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

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.

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

Semantic interpretations for these are:

(41) 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

(42)

$$\begin{array}{c} e^i \\ | \\ \phi \end{array}$$

Dynamic Event Structure

(43)

$$\begin{array}{c} e^{i} \\ | \\ \phi \end{array}$$

Tree structure for event concatenation:

$$\begin{array}{c} e^{i} \\ | \\ \phi \end{array} \quad + \quad \begin{array}{c} e^{j} \\ | \\ \phi \end{array} \quad = \quad \begin{array}{c} e^{[i,j]} \\ | \\ \phi \end{array}$$

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

(48) 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

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

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

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

(54) 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

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

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

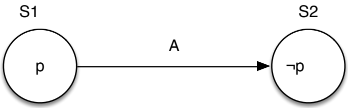
Labeled Transition System (LTS)

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

As a shorthand for

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

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



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:

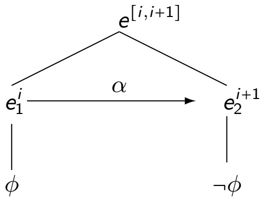
With temporal indexing from a Linear Temporal Logic, we can define a Temporal Labeled Transition System (TLTS). For a state, e_1, indexed at time i, we say $e_1@i$.

$(\{\phi\}_{e_1@i}, \alpha, \{\neg\phi\}_{e_2@i+1}) \in \rightarrow_{(i,i+1)}$, we use:

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

Dynamic Event Structure

(59)



Dynamic Event Structure

(60) Mary awoke from a long sleep.

Dynamic Event Structure

(62) 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", $j + 1$.

Dynamic Event Structure

(64) 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", $j + 1$.

(65)

$$
\begin{tikzcd}
& e^{[i,j+1]} \arrow[dl, dash] \arrow[dr, dash] & \\
e_1^{[i,j]} \arrow[rr, "\alpha"] \arrow[d, dash] & & e_2^{j+1} \arrow[d, dash] \\
\phi & & \neg\phi
\end{tikzcd}
$$

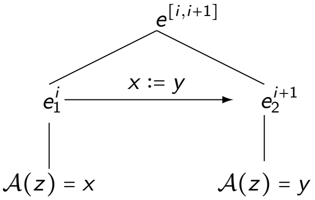
Simple First-order Transition

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

Simple First-order Transition

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

(69)



Processes

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

Processes

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

(71)

$$e$$

$$e_1 \xrightarrow{\nu} e_2 \quad \dots \quad \xrightarrow{\nu} \quad e_n$$

Dynamic Event Structures for Motion Predicates

- Manner construction languages
  Path information is encoded in directional PPs and other adjuncts, while verb encode manner of motion

- Manner construction languages
  Path information is encoded in directional PPs and other adjuncts, while verb encode manner of motion
  English, German, Russian, Swedish, Chinese
- Path construction languages

Dynamic Event Structures for Motion Predicates

- Manner construction languages
  Path information is encoded in directional PPs and other adjuncts, while verb encode manner of motion
  English, German, Russian, Swedish, Chinese
- Path construction languages
  Path information is encoded in matrix verb, while adjuncts specify manner of motion
  Modern Greek, Spanish, Japanese, Turkish, Hindi

Defining Motion (Talmy 1985)

(72) a. The *event* or situation involved in the change of location ;

Defining Motion (Talmy 1985)

(73) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);

Defining Motion (Talmy 1985)

(74) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;

Defining Motion (Talmy 1985)

(75) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;
d. A distinguished point or region of the path (the *ground*);

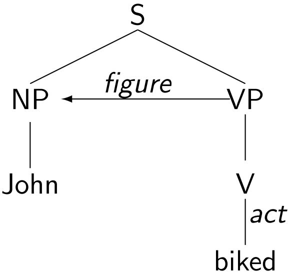
Defining Motion (Talmy 1985)

(76) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;
d. A distinguished point or region of the path (the *ground*);
e. The *manner* in which the change of location is carried out;

Defining Motion (Talmy 1985)

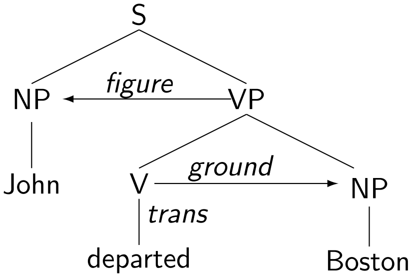
(77) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;
d. A distinguished point or region of the path (the *ground*);
e. The *manner* in which the change of location is carried out;
f. The *medium* through which the motion takes place.

(78)



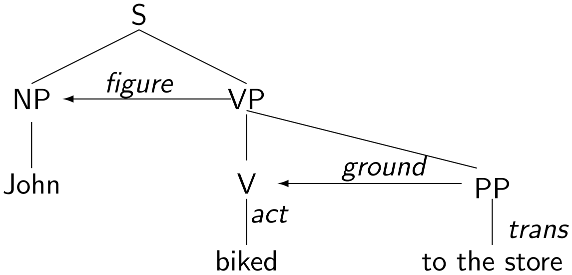
Path Predicates

(79)



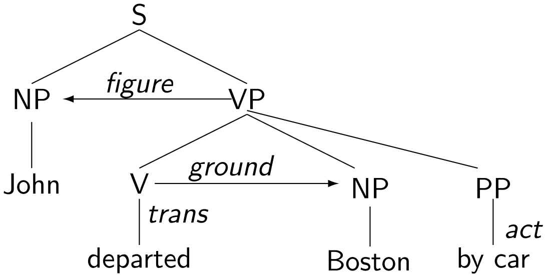
Manner with Path Adjunction

(80)



Path with Manner Adjunction

(81)



(82) a. Isabel climbed for 15 minutes.

Path+manner Predicates (Talmy 2000) 1/2

(84) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

Path+manner Predicates (Talmy 2000) 1/2

(86) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

(87) a. There is an action (e) bringing about an iterated non-distinguished change of location;

(88) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

(89) a. There is an action (e) bringing about an iterated non-distinguished change of location;
b. The figure undergoes this non-distinguished change of location;

Path+manner Predicates (Talmy 2000) 1/2

(90) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

(91) a. There is an action (e) bringing about an iterated non-distinguished change of location;
b. The figure undergoes this non-distinguished change of location;
c. The figure creates (leaves) a path by virtue of the motion.

Path+manner Predicates (Talmy 2000) 1/2

(92) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

(93) a. There is an action (e) bringing about an iterated non-distinguished change of location;
b. The figure undergoes this non-distinguished change of location;
c. The figure creates (leaves) a path by virtue of the motion.
d. The action (e) is performed in a certain manner.

Path+manner Predicates (Talmy 2000) 1/2

(94) a. Isabel climbed for 15 minutes.
b. Nicholas fell 100 meters.

(95) a. There is an action (e) bringing about an iterated non-distinguished change of location;
b. The figure undergoes this non-distinguished change of location;
c. The figure creates (leaves) a path by virtue of the motion.
d. The action (e) is performed in a certain manner.
e. The path is oriented in an identified or distinguished way.

Path+manner Predicates (Talmy 2000) 2/2

Unlike pure manner verbs, this class of predicates admits of two compositional constructions with adjuncts.

Path+manner Predicates (Talmy 2000) 2/2

Unlike pure manner verbs, this class of predicates admits of two compositional constructions with adjuncts.

(98) **Manner of motion verb with path adjunct;**
John climbed to the summit.

Path+manner Predicates (Talmy 2000) 2/2

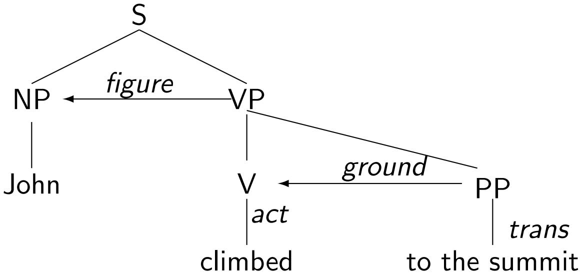
Unlike pure manner verbs, this class of predicates admits of two compositional constructions with adjuncts.

(100) **Manner of motion verb with path adjunct;**
John climbed to the summit.

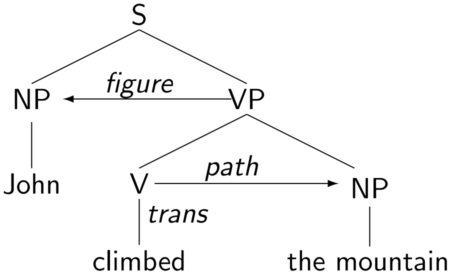
(101) **Manner of motion verb with path argument;**
John climbed the mountain.

With Path Adjunct

(102)



(103)



Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.

Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.
  The change in value is tested.

Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.
  The change in value is tested.
- Manner of motion verbs iterate a change in location from state to state.

Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.
  The change in value is tested.
- Manner of motion verbs iterate a change in location from state to state.
  The value is assigned and reassigned.

Directed Motion

(104) $\boxed{loc(z) = x}_{e_1} \xrightarrow{\nu} \boxed{loc(z) = y}_{e_2}$

$x \neq y?$

Directed Motion

(106) $\boxed{loc(z) = x}_{e_1} \xrightarrow{\nu} \boxed{loc(z) = y}_{e_2}$

$x \neq y?$

When this test references the ordinal values on a scale, $\mathcal{C}$, this becomes a *directed ν-transition ($\vec{\nu}$)*, e.g., $x \preccurlyeq y$, $x \succcurlyeq y$.

Directed Motion

(108) $\overset{\overset{x \neq y?}{\curvearrowleft}}{\boxed{loc(z) = x}}_{e_1} \xrightarrow{\nu} \boxed{loc(z) = y}_{e_2}$

When this test references the ordinal values on a scale, $\mathcal{C}$, this becomes a *directed ν-transition ($\vec{\nu}$)*, e.g., $x \preccurlyeq y$, $x \succcurlyeq y$.

(109) $\vec{\nu} =_{df} \overset{\overset{\mathcal{C}?}{\curvearrowleft}}{e_i} \xrightarrow{\nu} e_{i+1}$

Directed Motion

(110)

$$
\begin{tikzcd}
& e^{[i,i+1]} \arrow[dl, dash] \arrow[dr, dash] & \\
e_1^i \arrow[loop above, "x \preccurlyeq y?"] \arrow[rr, "x := y"] \arrow[d, dash] & & e_2^{i+1} \arrow[d, dash] \\
\mathcal{A}(z) = x & & \mathcal{A}(z) = y
\end{tikzcd}
$$

- Manner-of-motion verbs introduce an assignment of a location value:
  $loc(x) := y; y := z$

Change and Directed Motion

- Manner-of-motion verbs introduce an assignment of a location value:
  $loc(x) := y; y := z$
- Directed motion introduces a dimension that is measured against:
  $d(b, y) < d(b, z)$

Change and Directed Motion

- Manner-of-motion verbs introduce an assignment of a location value:
  $loc(x) := y; y := z$
- Directed motion introduces a dimension that is measured against:
  $d(b, y) < d(b, z)$
- Path verbs introduce a pair of tests:
  $\neg\phi? \ldots \phi?$

Change and the Trail it Leaves

Change and the Trail it Leaves

- The execution of a change in the value to an attribute $\mathcal{A}$ for an object x leaves a trail, τ.

Change and the Trail it Leaves

- The execution of a change in the value to an attribute $\mathcal{A}$ for an object x leaves a trail, τ.
- For motion, this trail is the created object of the path p which the mover travels on;

Change and the Trail it Leaves

- The execution of a change in the value to an attribute $\mathcal{A}$ for an object x leaves a trail, τ.
- For motion, this trail is the created object of the path p which the mover travels on;
- For creation predicates, this trail is the created object brought about by order-preserving transformations as executed in the directed process above.

Motion Leaving a Trail

(111) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$loc(x) := y$

(112) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$$loc(x) := y$$

b. Name this value b (this will be the beginning of the movement);

$$b := y$$

Motion Leaving a Trail

(113) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$$loc(x) := y$$

b. Name this value b (this will be the beginning of the movement);

$$b := y$$

c. Initiate a path p that is a list, starting at b;

$$p := (b)$$

Motion Leaving a Trail

(114) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$$loc(x) := y$$

b. Name this value b (this will be the beginning of the movement);

$$b := y$$

c. Initiate a path p that is a list, starting at b;

$$p := (b)$$

d. Then, reassign the value of y to z, where $y \neq z$

$$y := z, y \neq z$$

Motion Leaving a Trail

(115) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$$loc(x) := y$$

b. Name this value b (this will be the beginning of the movement);

$$b := y$$

c. Initiate a path p that is a list, starting at b;

$$p := (b)$$

d. Then, reassign the value of y to z, where $y \neq z$

$$y := z, y \neq z$$

e. Add the reassigned value of y to path p;

(116) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$loc(x) := y$

b. Name this value b (this will be the beginning of the movement);

$b := y$

c. Initiate a path p that is a list, starting at b;

$p := (b)$

d. Then, reassign the value of y to z, where $y \neq z$

$y := z, y \neq z$

e. Add the reassigned value of y to path p;

$p := (p, z)$

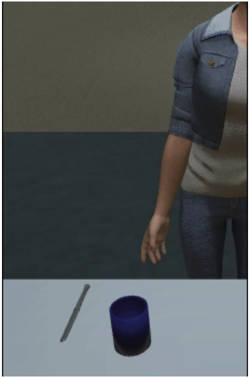
f. Kleene iterate steps (d) and (e).

VoxML - grasp

$$\begin{bmatrix} \textbf{grasp} \\ \text{LEX} = \begin{bmatrix} \text{PRED} = \textbf{grasp} \\ \text{TYPE} = \textbf{transition\_event} \end{bmatrix} \\ \text{TYPE} = \begin{bmatrix} \text{HEAD} = \textbf{transition} \\ \text{ARGS} = \begin{bmatrix} \text{A}_1 = \textbf{x:agent} \\ \text{A}_2 = \textbf{y:physobj} \end{bmatrix} \\ \text{BODY} = \begin{bmatrix} \text{E}_1 = grasp(x, y) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

VoxML - grasp cup

- Continuation-passing style semantics for composition
- Used within conventional sentence structures and between sentences in discourse in MSG



Events as Dynamic Programs

Pustejovsky and Moszkowicz (2011)