Multimodal Semantics for Affordances and Actions

Lecture 1: Multimodal Communication

James Pustejovsky and Nikhil Krishnaswamy

ESSLLI 2022 Summer School

Galway, Ireland

August 8-19, 2022

The Big Picture Goal

- As HCI becomes more multimodal (language, gesture, gaze, posture, facial expressions), we need to understand the role of affordances and human-object interactions.
- We examine how our knowledge of object interactions is rarely reflected in linguistic descriptions of actions (or images).
- We demonstrate how object and situational conditions on actions need to be identified and encoded, just as importantly as the actions themselves.

Course Learning Goals

- Identify requirements involved in developing a semantics for human-computer (HCI) and human-robot interactions (HRI).
  - We study Human-human interactions (HHI) in multimodal communication.
- Modeling human-object interactions for communication
  - object properties and behaviors
  - actions associated with objects
- Developing the notion of embodiment: an agent's actions are *situated* by the context and constrained by its *embodiment* in the context.
  - embodiment in a simulation
  - an embodied environment allows us to bridge and integrate formal symbolic and statistically oriented reasoning approaches, to create grounding and situated reasoning.

Course Outline

- Monday: Components of Multimodal Communication
- Tuesday: Modeling Human-Object Interactions
- Wednesday: Modeling Multimodal Common Ground
- Thursday: Communicating with Multimodal Common Ground
- Friday: Reasoning with and about Affordances

Monday: The Components of Multimodal Communication

- Achieving and maintaining common ground
  - shared conceptual space
- Context-aware interpretation of communicative acts
  - language, gesture, gaze
- Recognizing Object-specific knowledge and behavior
- Objects and actions are situated in the interaction
- Agents are embodied in the interaction:
  - all actions (communicative or physical) are interpreted through embodiment.
- Generative Lexicon object semantics

Situated Semantic Grounding and Embodiment

- Task-oriented dialogues are embodied interactions between agents, where language, gesture, gaze, and actions are situated within a common ground shared by all agents in the communication.
- Situated semantic grounding assumes shared perception of agents with co-attention over objects in a situated context, with co-intention towards a common goal.
- VoxWorld : a multimodal simulation framework for modeling Embodied Human-Computer Interactions and communication between agents engaged in a shared goal or task.

Situated Meaning

Mother and son interacting in a shared task of icing cupcakes

Situated Meaning in a Joint Activity

- Son: *Put it there (gesturing with co-attention)?*
- Mother: *Yes, go down for about two inches.*
- Mother: *OK, stop there. (co-attentional gaze)*
- Son: *Okay. (stops action)*
- Mother: *Now, start this one (pointing to another cupcake).*

Situated Meaning

Elements from the Common Ground

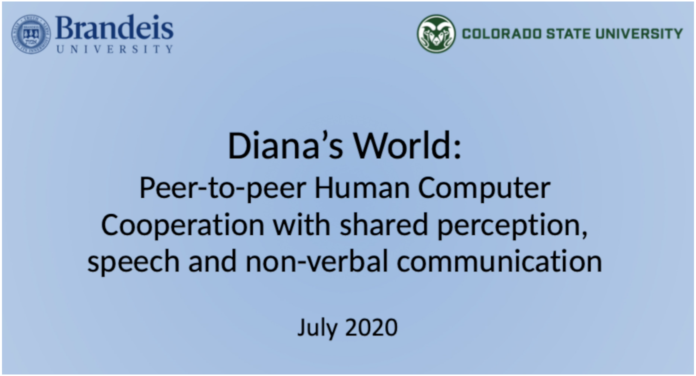
Agents	mother, son
Shared goals Beliefs, desires, intentions	baking, icing Mother knows how to ice, bake, etc. Mother is teaching son
Objects	Mother, son, cupcakes, plate, knives, pastry bag, icing, gloves
Shared perception	the objects on the table
Shared Space	kitchen

The Challenge of Situated Grounding

1. Human-computer/robot interactions require at least the following capabilities:
   - Robust recognition and generation within multiple modalities
     - language, gesture, vision, action;
   - understanding of contextual grounding and co-situatedness;
   - appreciation of the consequences of behavior and actions.

2. Multimodal simulations provide an approach to modeling human-computer communication by situating and contextualizing the interaction, thereby visually demonstrating what the computer/robot sees and believes.

Diana’s World



Link

The Meaning of Embodiment in Communication

- Agent has situated meaning for the objects and actions in the environment;
- Recognition of the human's embodiment; agent has awareness of people's linguistic and gestural expressions, facial expressions, and actions.
- Self-embodiment of the agent: the agent has "spatial presence" within the domain of the interaction

Shared Conceptual Space

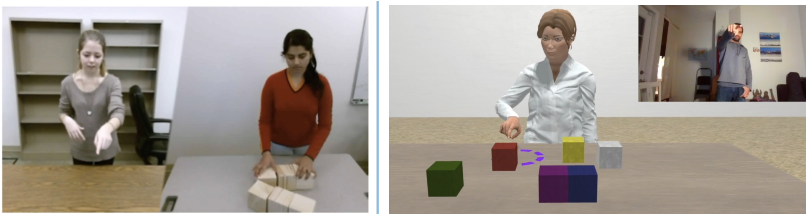


Figure: *Left:* Human-human collaborative interaction; *Right:* Human-avatar interaction.

Embodiment and Situated Meaning

- Elements of Situated Meaning
  - Identifying the *actions and consequences* associated with objects in the environment.
  - Encoding a multimodal expression contextualized to the *dynamics of the discourse*
  - *Situated grounding*: Capturing how multimodal expressions are anchored, contextualized, and situated in context
- Modalities Deployed
  - gesture recognition and generation
  - language recognition and generation
  - affect, facial recognition, and gaze
  - action generation

Recognition of Human's Embodiment

Awareness of the partner's:

- linguistic and gestural expressions
- gestural expressions
- facial expressions
- gaze and eye tracking
- actions

The agent continuously constructs and maintains a representation of the embodiment of its human partner.

Agent Self-embodiment

"spatial presence" within the domain of the interaction

- facial "countenance"
- explicit effectors for action
- explicit sensors for audio and visual input.
- Constraints on its behavior are imposed by the physical extents and limitations of the embodiment (e.g., how far it can reach, degrees of freedom on the joints, etc.).

A non-verbal interaction between a human and IVA using gesture, gaze, and action.

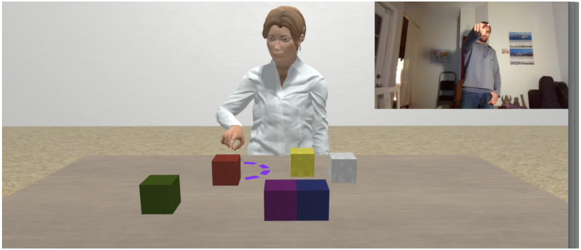


Figure: IVA Diana engaging in an embodied HCI with a human user.

Link

Modeling Human Object Interactions (HOIs)

- The objects in a dialogue carry much more semantic information than conventionally assumed.
- This includes knowledge for how the objects can be manipulated by an agent in space and time, their *Gibsonian affordances*, and how they can be used, their *Telic affordances*.
- Such information also includes knowledge of how an object is situated in the environment relative to an agent for specific purposes and actions, that is, its *habitat*.
- We show how affordance encoding and recognition can improve object and action classification in HCI tasks.

Captions Don't Describe Human-Object Interactions

Neither do conventional semantic representations

"Woman drinking coffee."

(1) a. $drink(w, c)$

b. $\exists x \exists y[\text{woman}(x) \wedge coffee(y) \wedge \text{drink}(x, y)]$

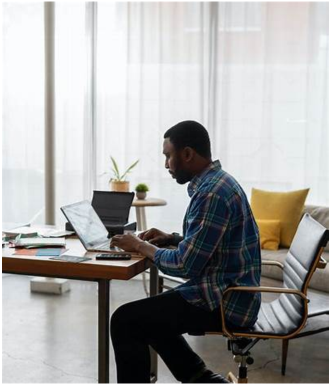
c. $\text{EVENT}(drink) \wedge \text{AGENT}(woman) \wedge \text{PATIENT}(coffee)$

What the Caption Leaves Out

Dense Paraphrase

- *A woman drinking coffee.*
- A upright seated woman is holding in her hand, a cup filled with coffee while she drinks it.
- The cup is upright so the container portion (inside) is able to hold coffee.
- She is holding the cup by an attached handle.
- The cup is tilted towards her and touches her partially open mouth, in order to allow drinking.

Captions Don't Describe Human-Object Interactions



"A man working at a desk."

What the Caption Leaves Out

- *A man working at a desk.*
- A upright man is seated in a chair, typing with both hands on the keyboard of a laptop, which is on the top surface of a table.
- The chair he is seated in is close enough to the table for him to reach the keyboard.
- The laptop is open, with the keyboard exposed flat and the screen facing the man.
- The man is facing the computer screen and keyboard and the desk.

Affordance and Goal Recognition

1. Perceived purpose is an integral component of how we interpret situations and reason about utterances in communicative contexts.

- Events are purposeful and directed;
- Places are functional;
- Objects are usable and manipulable.

2. Affordances are latent action structures of how an agent interacts with objects in the environment, in different modalities:

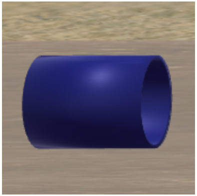
- language, gesture, vision, action;

3. Qualia Structure provides a link to such latent actions structures associated with objects in utterances and the context.

Encoding Object Behavior

- Context of objects is described by their properties.
- Object properties cannot be decoupled from the events they facilitate.
  - *Affordances* (Gibson, 1979)
  - *Qualia* (Pustejovsky, 1995)

"He **slid** the cup across the table. Liquid spilled out."
"He **rolled** the cup across the table. Liquid spilled out."



Visual Object Concept Modeling Language (VoxML)

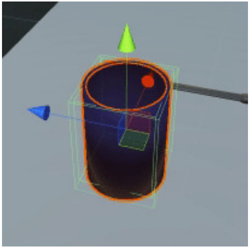
Pustejovsky and Krishnaswamy (2016)

- Encodes afforded behaviors for each object
  - Gibsonian: afforded by object structure (Gibson,1977,1979)
    - grasp, move, lift, etc.
  - Telic: goal-directed, purpose-driven (Pustejovsky, 1995, 2013)
    - 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 - cup

cup

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



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

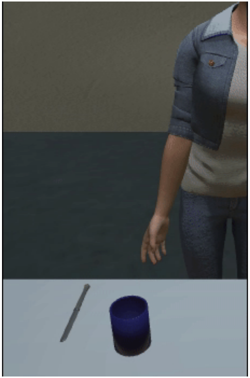
$$\begin{bmatrix} \textbf{put} \\ \text{LEX} = \begin{bmatrix} \text{PRED} = \textbf{put} \\ \text{TYPE} = \textbf{transition\_event} \end{bmatrix} \\ \text{TYPE} = \begin{bmatrix} \text{HEAD} = \textbf{transition} \\ \text{ARGS} = \begin{bmatrix} \text{A}_1 = \textbf{x:agent} \\ \text{A}_2 = \textbf{y:physobj}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

VoxML - grasp

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

VoxML - grasp cup

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



Multimodal Simulations

- Human understanding depends on a wealth of common-sense knowledge; humans perform much reasoning qualitatively.
- To simulate events, every parameter must have a value
  - "Roll the ball." How fast? In which direction?
  - "Roll the block." Can this be done?
  - "Roll the cup." Only possible in a certain orientation.
- VoxML: Formal semantic encoding of properties of objects, events, attributes, relations, functions.
- VoxSim: What can situated grounding do? (Krishnaswamy, 2017)
  - Exploit numerical information demanded by 3D visualization;
  - Perform qualitative reasoning about objects and events;
  - Capture semantic context often overlooked by unimodal language processing.

VoxWorld Architecture

Pustejovsky and Krishnaswamy (2016), Krishnaswamy (2017), Pustejovsky et al (2017), Narayana et al (2018)

- Dynamic interpretation of actions and communicative acts:
  - Dynamic Interval Temporal Logic (DITL)
  - Dialogue Manager
- VoxML: Visual Object Concept Modeling Language
- EpiSim: Visualizes agent's epistemic state and perceptual state in context;
  - Public Announcement Logic
  - Public Perception Logic
- VoxSim: 3D visualizer of actions, communicative acts, and context.
  - Built on Unity Game Engine

Dynamic Discourse Interpretation

- Common Ground Structure
  - Co-belief
  - Co-perception
  - Co-situatedness
- Multimodal communication act:
  - language
  - gesture
  - action
- Dynamic tracking and updating of dialogue with:
  - Discourse Sequence Grammar
  - Gesture Grammar
  - Action Grammar

Multimodal Communicative Acts

- A communicative act, performed by an agent, a, is a tuple of expressions from the modalities available to a, involved in conveying information to another agent.
- We restrict this to the modalities of speech, S, gesture, G, facial expression F, gaze Z, an explicit action A.
    1. $C_a = \langle S, G, F, Z, A \rangle$
- These modal channels can be aligned or unaligned in the input.

Co-belief and Co-perception in the Common Ground

- *Public announcement logic* (PAL)
  - $[\alpha]\varphi$ denotes that an agent "α *knows* φ".
  - Public Announcement: $[!\varphi_1]\varphi_2$
  - Any proposition, φ, in the common knowledge held by two agents, α and β, is computed as: $[(\alpha \cup \beta)^*]\varphi$.
- *Public perception logic* (PPL)
  - $[\alpha]_\sigma\varphi$ denotes that agent "α perceives that φ".
  - $[\alpha]_\sigma\hat{x}$ denotes that agent "α perceives that there is an x."
  - Public Display: $[!\varphi_1]_\sigma\varphi_2$
  - The co-perception by two agents, α and β includes φ : $[(\alpha \cup \beta)^*]_\sigma\varphi$

The situated common ground consists of the following state information:

(2) a. **A**: The agents engaged in communication;
b. **B**: The shared belief space;
c. **P**: The objects and relations that are jointly perceived in the environment;
d. $\mathcal{E}$: The embedding space that both agents occupy in the communication.

(3) $\boxed{\begin{array}{l} \textbf{A:}a_1, a_2 \quad \textbf{B:}\Delta \quad \textbf{P:}b \\ \hline \\ \mathcal{S}_{a_1} = \text{"You}_{a_2} \text{ see it}_b\text{"} \end{array}}_{\mathcal{E}}$

- State Monad: $M\alpha = State \rightarrow (\alpha \times State)$
- Context is a stack of items and the type of left contexts is a list of entities, $[e]$.
- Right contexts will be interpreted as continuations: a discourse that requires a left context to yield a truth value., of type $[e] \rightarrow t$.
- Hence, context transitions are of type $[e] \rightarrow [e] \rightarrow t$;
- Given the current discourse, T, and a new expression, C, C updates T as follows:
- $[\![\overline{(\mathbf{T.C})}]\!]^{M,cg} = \lambda k.[\![\overline{\mathbf{T}}]\!](\lambda n.[\![\overline{\mathbf{C}}]\!](\lambda m.k(m\ n)))$

- Representing how gestures denote
- Encoding co-perception of situated objects under reference
- Situated alignment of expressions from distinct modalities

Gesture Types in Multimodal Interactions

1. Deixis (pointing) gestures, generated to request information regarding an object, a location, or a direction when performing a specific action;
2. Iconic action gestures, generated to request clarification on how (what manner of action) to perform a specific task;
3. Affordance-denoting gestures, generated to describe how the agent can interact with an object, even when it does not know what it is or what it might be used for;
4. Direct situated actions, where the agent responds to a command or request by acting in the environment directly.

Gestures used in VoxWorld

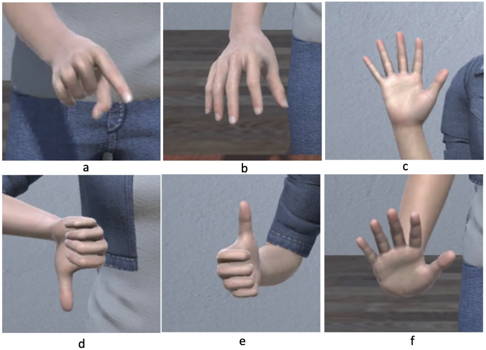
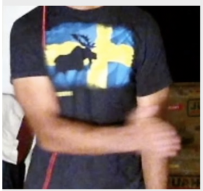


Figure: Some of the gestures generated by VoxWorld: pointing, grab, five, no, yes, push back.

- On the left, a human is action gesturing to move an object to the left:
- On the right, the IVA is performing the identical gesture.



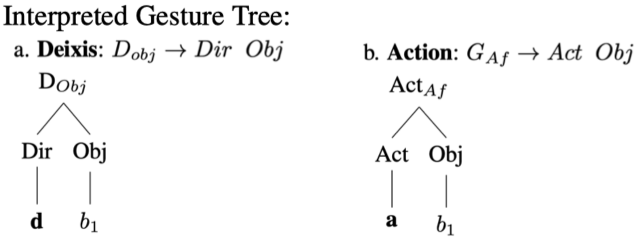
Actions as Described by Gesture

Kendon (2004), Lascarides and Stone (2009)

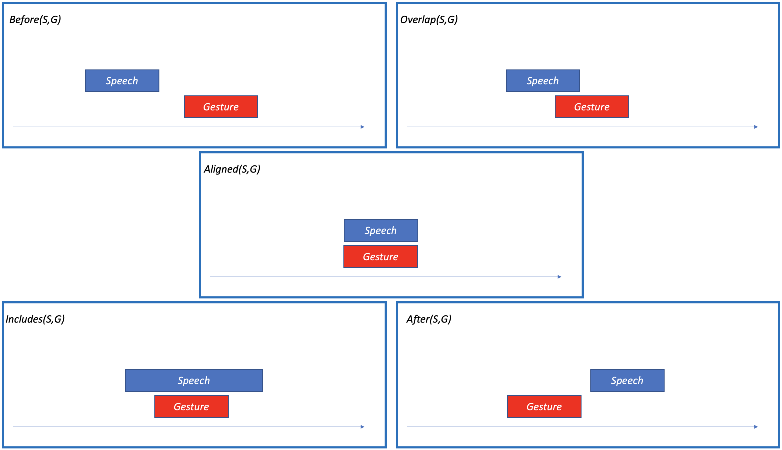
- $G \rightarrow (Prep)\ (Pre\_stroke\ Hold)$ Stroke $Retract$

The stroke is the content-baring phase, **d**, and in a pointing gesture, will convey the deictic orientational information.

- $[\![point]\!] = [\![End(cone(\mathrm{d}))]\!]$



Aligning Speech and Gesture in Dialogue



Reasoning in Multimodal Simulations

- Human understanding depends on a wealth of common-sense knowledge; humans perform much reasoning qualitatively.
- To simulate events, every parameter must have a value
  - "Roll the ball." How fast? In which direction?
  - "Roll the block." Can this be done?
  - "Roll the cup." Only possible in a certain orientation.
- VoxML: Formal semantic encoding of properties of objects, events, attributes, relations, functions.
- VoxSim: What can situated grounding do? (Krishnaswamy, 2017)
  - Exploit numerical information demanded by 3D visualization;
  - Perform qualitative reasoning about objects and events;
  - Capture semantic context often overlooked by unimodal language processing.

Transfer Learning of Object Affordances

- Gibsonian/Telic affordances are associated with abstract properties:
  - spheres roll, sphere-like entities probably do too;
  - small cups are graspable, small cylindroid-shaped objects probably are too.
- Similar objects have similar habitats/affordances:
- This informs the way you can talk about items in context:
  - Q: "What am I pointing at?"
  - A: "I don't know, but it looks like {a ball/a container/etc.}

Affordance-denoting gestures

▸ Play!

Reasoning with Affordances

Learning how to stack a cube

- An agent can interact with various objects and see how they behave differently under the same circumstances.
- An agent can learn to distinguish objects based on behaviors.

Generative Lexicon: Type-driven Compositionality

- A compositional semantics that has a rich library of object knowledge
- The lexicon is viewed as genuinely generative:
  - Specific lexical mechanisms derive an infinite number of word senses from a finite number of meaning elements.
  - It incorporates and dictates essential components of other linguistic levels: syntax and semantics.
- Decompositional approach to word meaning: the lexicon comprises several levels of representation and *sublexical* components.

  Decomposition is applied to uncover compositional/relational aspects of lexical semantics.
  - *Diana almost built a staircase.*
    - almost(build_act, be_built)
    - [build_act, almost(be_built)]

Lexical structures in GL

- Argument structure: specification of number and type of the predicate's arguments.
  - Rich argument typology, including unexpressed arguments.
- Event structure: overall event type of the predicate, its parts (*subevents*), their relative ordering and prominence.
  - *Dynamic Event Structure*: scalar properties of events.
- Qualia structure: decomposed representation of lexical meaning in terms of four dimensions:
  a. FORMAL (F): basic semantic typing (**is_a** relation: *fence* is a kind of 'barrier'); features that distinguish the object within a larger domain
  b. AGENTIVE (A): factors involved in the origin of an object ('build' for *fence*).
  c. TELIC (T): purpose/function of the object ('separate/ prevent from entering or leaving' for *fence*).
  d. CONSTITUTIVE (C): relation between an object and its proper parts ('wood/metal' for *fence*), or what it is a part of.

Lexical structures integrated in a lexical entry

$$\begin{bmatrix} \textbf{build} & \\ \text{AS} = & \begin{bmatrix} \text{ARG}_1 = x\textbf{: animate} \\ \text{ARG}_2 = y : \textbf{phys\_obj} \\ \text{D-ARG}_1 = w : \textbf{phys\_obj} \end{bmatrix} \\ \text{ES} = & \begin{bmatrix} \text{E}_0 = \textbf{e}_0\textbf{: transition} \\ \text{E}_1 = \textbf{e}_1 \textbf{ process} \\ \text{E}_2 = \textbf{e}_2 \textbf{ state} \end{bmatrix} \\ \text{QS} = & \begin{bmatrix} \text{FORMAL} = \textbf{be\_built(e}_2\textbf{,y)} \\ \text{AGENTIVE} = \textbf{build\_act(e}_1\textbf{,x,w)} \end{bmatrix} \end{bmatrix}$$

- ES/ QS: ES subevents are identified as processes and resultant states in the QS
- QS/ AS: QS (formal role) provides the semantic typing of the arguments in the AS
- ES/ AS: AS arguments are involved in different parts of the ES

Semantic types

- Semantic types: the kind of entity denoted by a lexical item.
  - *table* is a kind of FURNITURE → semantic type
- Ontology: what entities exist and how they can be grouped and related within a hierarchy.

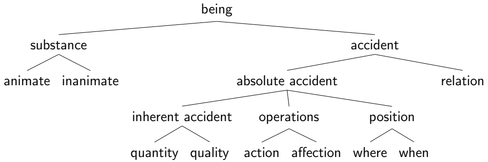


Figure: *Aristotelian ontology*

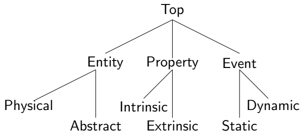


Figure: *Classic Upper Ontology*

Linguistic motivation of semantic types

- Which semantic categories are linguistically relevant?
- A piece of conceptual information can be considered linguistic if it affect other modules of grammar (syntax and morphology).
- Sem. properties systematically shared by arguments of different Vs:
  - ANIMATE
    - a. The *chef* was chopping onions.
    - b. The *girl* played with the blocks.
  - EVENT
    - a. John finished {the *dinner*/ *eating/preparing the dinner*}.
    - b. He was late for the *lecture*.
  - LOCATION
    - a. The stack is on *the edge of the table*.
    - b. Put the block *there*.
  - TIME
    - a. Stir the soup for *10 minutes*.
    - b. Ananya played with blocks *today*.

Integrating GL with Distributional Semantics

1. Context vectors for words encode syntagmatic relations between words

$$w_i \quad \textbf{Verb} \quad w_j$$

Integrating GL with Distributional Semantics

1. Context vectors for words encode syntagmatic relations between words

$$w_i \quad \textbf{Verb} \quad w_j$$

2. Abstractions or clusters over contexts give rise to paradigmatic relations between words

Integrating GL with Distributional Semantics

1. Context vectors for words encode syntagmatic relations between words

$$w_i \quad \textbf{Verb} \quad w_j$$

2. Abstractions or clusters over contexts give rise to paradigmatic relations between words

$$\begin{cases} v_1 \\ v_2 \\ \ldots \\ v_n \end{cases} \quad \textbf{Verb} \quad \begin{cases} w_1 \\ w_2 \\ \ldots \\ w_n \end{cases}$$

(4) a. Natural types:

(5) a. Natural types:

- *Simple*: Natural kind concepts consisting of reference only to Formal or Constitutive qualia roles;

GL Type Structures

(6) a. Natural types:

- *Simple*: Natural kind concepts consisting of reference only to Formal or Constitutive qualia roles;
- *Functional*: Additional reference to Telic (purpose or function)

(7) a. Natural types:

- *Simple*: Natural kind concepts consisting of reference only to Formal or Constitutive qualia roles;
- *Functional*: Additional reference to Telic (purpose or function)

b. Artifactual types: Concepts making reference to Agentive (origin) for a specific Telic (purpose or function);

(8) a. Natural types:

- *Simple*: Natural kind concepts consisting of reference only to Formal or Constitutive qualia roles;
- *Functional*: Additional reference to Telic (purpose or function)

b. Artifactual types: Concepts making reference to Agentive (origin) for a specific Telic (purpose or function);

c. Complex types: Concepts integrating reference to a logical coherence relation between types from the other two levels.

Kinds of Compositionality

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

Kinds of Compositionality

1. Weak Compositionality:
   If all you have for composition is function application, then you need to create as many lexical entries for an expression as there are environments it appears in.
2. True Compositionality: Enrich the mechanisms of making larger meanings by taking advantage of all espressions in the phrase; type coercion, qualia exploitation, co-composition.

Modes of Composition

(9) a. PURE SELECTION (Type Matching): the type a function requires is directly satisfied by the argument;

Modes of Composition

(10) a. PURE SELECTION (Type Matching): the type a function requires is directly satisfied by the argument;
b. ACCOMMODATION: the type a function requires is inherited by the argument;

Modes of Composition

(11) a. PURE SELECTION (Type Matching): the type a function requires is directly satisfied by the argument;
b. ACCOMMODATION: the type a function requires is inherited by the argument;
c. TYPE COERCION: the type a function requires is imposed on the argument type. This is accomplished by either:

Modes of Composition

(12) a. PURE SELECTION (Type Matching): the type a function requires is directly satisfied by the argument;

b. ACCOMMODATION: the type a function requires is inherited by the argument;

c. TYPE COERCION: the type a function requires is imposed on the argument type. This is accomplished by either:

i. *Exploitation*: taking a part of the argument's type to satisfy the function;

Modes of Composition

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

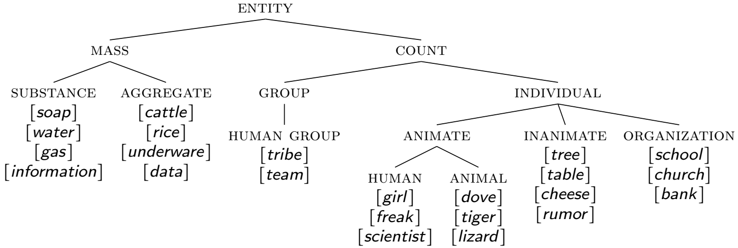
Notation and Language: typed feature structures

$$\begin{bmatrix} \alpha \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = x \\ \ldots \end{bmatrix} \\ \text{EVENTSTR} = \begin{bmatrix} \text{EVENT1} = e1 \\ \text{EVENT2} = e2 \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} = \textbf{what } x \textbf{ is made of} \\ \text{FORMAL} = \textbf{what } x \textbf{ is} \\ \text{TELIC} = e_2\textbf{: function of } x \\ \text{AGENTIVE} = e_1\textbf{: how } x \textbf{ came into being} \end{bmatrix} \end{bmatrix}$$

Type list

Semantic type	Example
ENTITY	thing
–MASS	soap, luggage, cattle
–COUNT	cow, table, tribe, bank
EVENT	happening, situation
–STATE	happiness, depression, love, be sick, be German
–DYNAMIC EVENT	demonstration, arrival, learn, build, jump
PROPERTY	
–INDIVIDUAL-LEVEL	tall, intelligent, respectful
–STAGE-LEVEL	hungry, tired, bored
PROPOSITION	(He told me) that you left
–INFORMATION	data/datum, commentary, rumor, message, summary, handout
TIME	tonight, soon, after dark, the day we met
LOCATION	upstairs, world-wide, here, downtown, in the yard
DIRECTION	towards, via, down
QUANTITY	seven, (a) few, (a) little, numerous, great deal, severely
MANNER	fast, happily, cruelly, with joy

- Only simple inheritance



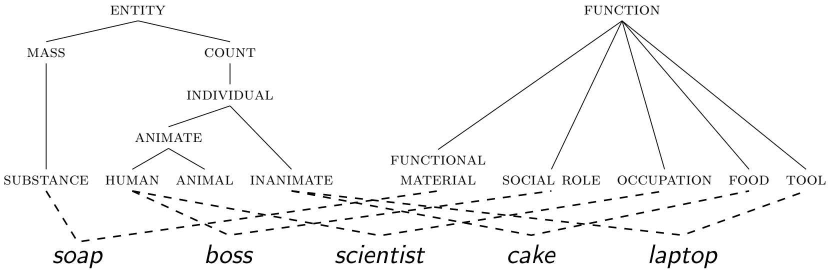
Qualia-based types: natural and artifactual

- FORMAL is the head type: an entity can have a function or not but, if it exists, it must 'be something' (physical object, substance, etc.).
- When the property encoded in the FORMAL role persists, other properties may not persist, but if it does not persist no other property persists because the entity no longer exists.
  - *broken camera*: PHYSICAL OBJECT with no function
  - *former boss*: HUMAN INDIVIDUAL with no function
- Type constructor tensor ($\otimes$): introduces types containing other qualia on top of the formal:

  a. *cake*: phys_obj $\otimes_C$ $\{flour, sugar\}$ $\otimes_A$ bake $\otimes_T$ eat
  b. *tree*: phys_obj $\otimes_C$ trunk $\otimes_C$ foliage
  c. *beer*: liquid $\otimes_C$ $\{water, yeast\}$ $\otimes_A$ brew $\otimes_T$ drink

Qualia-based types: natural and artifactual

- Multiple inheritance in artifactual types



Qualia-based types: complex (dot objects)

- Complex types (dot objects) have more than one semantic type specified in their formal role.
- **dot** (•) is the type constructor that creates a complex type $a \bullet b$ from any two types a and b.

a. *novel*: PHYSICAL_OBJECT • INFORMATION

$$\begin{bmatrix} \textbf{novel} \\ \text{QS} = \begin{bmatrix} \text{F} = \textbf{phys\_obj} \\ \text{C} = \textbf{page, cover...} \\ \text{T} = \textbf{containing\_information} \\ \text{A} = \textbf{printing, binding...} \end{bmatrix} \end{bmatrix} \bullet \begin{bmatrix} \textbf{novel} \\ \text{QS} = \begin{bmatrix} \text{F} = \textbf{information} \\ \text{C} = \textbf{narrative} \\ \text{T} = \textbf{entertain} \\ \text{A} = \textbf{writing} \end{bmatrix} \end{bmatrix}$$

- For a dot object to be well-formed, there must exist a relation between its contituent types:
  a. *book*: [F = contain(PHYS_OBJ, INFORMATION)]
  b. *Toyota*: [F = produce(PRODUCER, PRODUCT)]
  c. *chicken*: [F = used_as(ANIMAL, FOOD)]
  d. *exam*: [F = ask(EVENT, HUMAN, QUESTION)]

What is a *Quale*?

- A *Quale* (singular of *Qualia*) indicates **a single aspect of a word's meaning**, defined on the basis of the relation between the concept expressed by the word and another concept that the word evokes.

What is a *Quale*?

- A *Quale* (singular of *Qualia*) indicates **a single aspect of a word's meaning**, defined on the basis of the relation between the concept expressed by the word and another concept that the word evokes.
- Among the conceptual relations that a word may activate Qualia relations as defined in GL are those that are exploited in our understanding of linguistic expressions.

What is a *Quale*?

- A *Quale* (singular of *Qualia*) indicates **a single aspect of a word's meaning**, defined on the basis of the relation between the concept expressed by the word and another concept that the word evokes.
- Among the conceptual relations that a word may activate Qualia relations as defined in GL are those that are exploited in our understanding of linguistic expressions.
- *fresh bread* = "bread which has been baked recently."

The Formal

- The Formal (F) encodes the relation between the entity denoted by the word and the category it belongs to.

The Formal

- The Formal (F) encodes the relation between the entity denoted by the word and the category it belongs to.
- This relation enables one to grasp the nature of an entity by discriminating it from other kinds.

The Formal

- The Formal (F) encodes the relation between the entity denoted by the word and the category it belongs to.
- This relation enables one to grasp the nature of an entity by discriminating it from other kinds.
- What type of entity is x denoting? *rock* denotes a natural kind, *table* denotes an artifact, *car* denotes a vehicle, *park* denotes a location, *water* denotes a liquid, *plant* denotes a living thing, *fish* denotes an animal, *hand* denotes a body part, *glass* denotes a container, and so on.

The Formal

- More classifications are possible for the same type of object: for example, a *knife* can denote both a weapon or a kitchenware.

The Formal

- More classifications are possible for the same type of object: for example, a *knife* can denote both a weapon or a kitchenware.
- Classifications at different levels of generalization are available for reference:

The Formal

- More classifications are possible for the same type of object: for example, a *knife* can denote both a weapon or a kitchenware.
- Classifications at different levels of generalization are available for reference:
- a liquid such as <u>water</u>.

The Formal

- More classifications are possible for the same type of object: for example, a *knife* can denote both a weapon or a kitchenware.
- Classifications at different levels of generalization are available for reference:
- a liquid such as <u>water</u>.
- fluids such as <u>water</u> or air.

The Formal

- More classifications are possible for the same type of object: for example, a *knife* can denote both a weapon or a kitchenware.
- Classifications at different levels of generalization are available for reference:
- a liquid such as <u>water</u>.
- fluids such as <u>water</u> or air.
- substances such as <u>fluids</u>, salts, glucose and carbon dioxide.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.
- Size and dimensional properties.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.
- Size and dimensional properties.
- Shape and form.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.
- Size and dimensional properties.
- Shape and form.
- Color.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.
- Size and dimensional properties.
- Shape and form.
- Color.
- Position.

Formal factors for the class of nouns denoting concrete entities

- Spatial characteristics, intrinsic orientation.
- Size and dimensional properties.
- Shape and form.
- Color.
- Position.
- Surface.

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color$_F$)

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color_F)
- a long <u>carrot</u>

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color$_F$)
- a long <u>carrot</u>
- (Dimension$_F$)

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color$_F$)
- a long <u>carrot</u>
- (Dimension$_F$)
- a round <u>plate</u>

Values for Formal factors of nouns denoting concrete entities

- a red block
- (Color$_F$)
- a long carrot
- (Dimension$_F$)
- a round plate
- (Shape$_F$)

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color_F)
- a long <u>carrot</u>
- (Dimension_F)
- a round <u>plate</u>
- (Shape_F)
- a red <u>pen</u>

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color$_F$)
- a long <u>carrot</u>
- (Dimension$_F$)
- a round <u>plate</u>
- (Shape$_F$)
- a red <u>pen</u>
- (Color$_F$) or T/C (depending on contextual interpretation)

Values for Formal factors of nouns denoting concrete entities

- a red block
- (Color$_F$)
- a long carrot
- (Dimension$_F$)
- a round plate
- (Shape$_F$)
- a red pen
- (Color$_F$) or T/C (depending on contextual interpretation)
- a flat screen

Values for Formal factors of nouns denoting concrete entities

- a red block
- (Color_F)
- a long carrot
- (Dimension_F)
- a round plate
- (Shape_F)
- a red pen
- (Color_F) or T/C (depending on contextual interpretation)
- a flat screen
- (Shape_F)

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color_F)
- a long <u>carrot</u>
- (Dimension_F)
- a round <u>plate</u>
- (Shape_F)
- a red <u>pen</u>
- (Color_F) or T/C (depending on contextual interpretation)
- a flat <u>screen</u>
- (Shape_F)
- a thick <u>wall</u>

Values for Formal factors of nouns denoting concrete entities

- a red <u>block</u>
- (Color_F)
- a long <u>carrot</u>
- (Dimension_F)
- a round <u>plate</u>
- (Shape_F)
- a red <u>pen</u>
- (Color_F) or T/C (depending on contextual interpretation)
- a flat <u>screen</u>
- (Shape_F)
- a thick <u>wall</u>
- (Dimension_F)

Values for Formal factors of nouns denoting concrete entities

- the length of the <u>table</u>

Values for Formal factors of nouns denoting concrete entities

- the length of the <u>table</u>
- (Dimension$_F$)

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building

Values for Formal factors of nouns denoting concrete entities

- the length of the <u>table</u>
- (Dimension$_F$)
- the facade of the <u>building</u>
- (Orientation$_F$)

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building
- (Orientation$_F$)
- wipe the floor

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building
- (Orientation$_F$)
- wipe the floor
- (Surface$_F$)

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building
- (Orientation$_F$)
- wipe the floor
- (Surface$_F$)
- a large round table

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building
- (Orientation$_F$)
- wipe the floor
- (Surface$_F$)
- a large round table
- (Size$_F$) (Shape$_F$)

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension$_F$)
- the facade of the building
- (Orientation$_F$)
- wipe the floor
- (Surface$_F$)
- a large round table
- (Size$_F$) (Shape$_F$)
- *a round and square table

Values for Formal factors of nouns denoting concrete entities

- the length of the table
- (Dimension_F)
- the facade of the building
- (Orientation_F)
- wipe the floor
- (Surface_F)
- a large round table
- (Size_F) (Shape_F)
- *a round and square table
- (Shape_F) (Shape_F)

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.

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)

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

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:

(14) 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

(15) 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

(16) 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*.

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

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

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

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. $\lambda x\colon e_A[\ spoil(\mathrm{x})]$

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

- The beer spoiled.
- Mary fixed the watch.

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 $\sigma \sqcap \tau = \bot$:

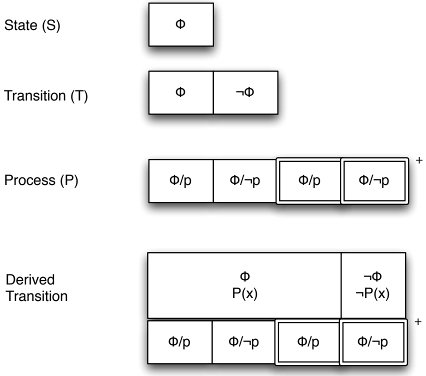
1. [___] $_\sigma$ X
2. [___] $_\tau$ Y

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.

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').

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

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:

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: $\bigcirc\phi$, $\diamond\phi$, $\square\phi$, $\phi\,\mathcal{U}\psi$.
- 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);

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);

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, ϕ? (test to see if ϕ is true, and proceed if so).

Dynamic Event Structure

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

(18) 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

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

Dynamic Event Structure

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

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

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

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.

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

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

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.

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

Semantic interpretations for these are:

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

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.

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

Semantic interpretations for these are:

(33) 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

(34)

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

Dynamic Event Structure

(35)

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

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

(40) 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

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

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

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

(46) 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

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

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

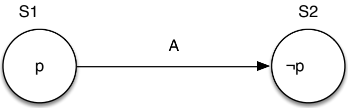
Labeled Transition System (LTS)

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

As a shorthand for

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

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



Temporal Labeled Transition System (TLTS)

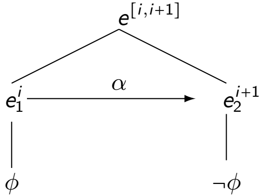
With temporal indexing from a Linear Temporal Logic, we can define a Temporal Labeled Transition System (TLTS). For a state, e_1, indexed at time i, we say $e_1@i$.
$(\{\phi\}_{e_1@i}, \alpha, \{\neg\phi\}_{e_2@i+1}) \in \rightarrow_{(i,i+1)}$, we use:

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:

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

(51)



Dynamic Event Structure

(52) Mary awoke from a long sleep.

Dynamic Event Structure

(54) 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", $j + 1$.

(56) 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", $j + 1$.

(57)

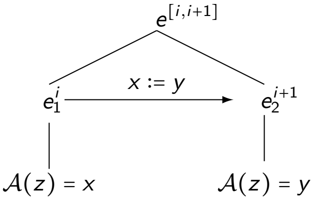
$$\begin{tikzcd}
 & e^{[i,j+1]} \arrow[dl, no head] \arrow[dr, no head] & \\
e_1^{[i,j]} \arrow[rr, "\alpha"] \arrow[d, no head] & & e_2^{j+1} \arrow[d, no head] \\
\phi & & \neg\phi
\end{tikzcd}$$

Simple First-order Transition

(58) $x := y$ (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 $\langle \mathcal{M}, i, u \rangle \models s_1 \wedge \langle \mathcal{M}, i+1, u[x/u(y)] \rangle \models x = y$

(60) $x := y$ (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 $\langle \mathcal{M}, i, u \rangle \vDash s_1 \wedge \langle \mathcal{M}, i+1, u[x/u(y)] \rangle \vDash x = y$

(61)



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:

(63)
$$\begin{array}{c} e \\ e_1 \xrightarrow{\nu} e_2 \;\ldots\; \xrightarrow{\nu} e_n \end{array}$$

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

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

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)

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

Defining Motion (Talmy 1985)

(65) 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)

(66) 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)

(67) 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)

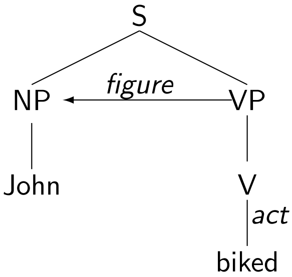
(68) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;
d. A distinguished point or region of the path (the *ground*);
e. The *manner* in which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

Defining Motion (Talmy 1985)

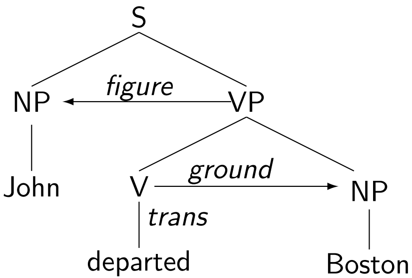
(69) a. The *event* or situation involved in the change of location ;
b. The object (construed as a point or region) that is undergoing movement (the *figure*);
c. The region (or *path*) traversed through the motion;
d. A distinguished point or region of the path (the *ground*);
e. The *manner* in which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

Manner Predicates

(70)

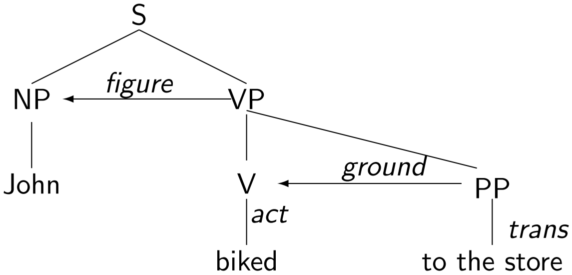


(71)



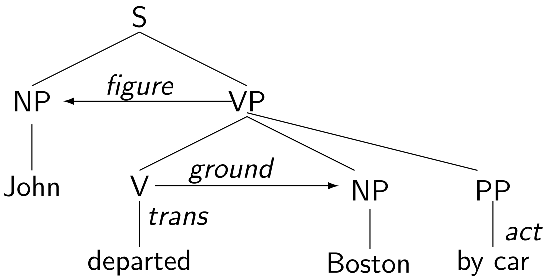
Manner with Path Adjunction

(72)



Path with Manner Adjunction

(73)



Path+manner Predicates (Talmy 2000) 1/2

(74) a. Isabel climbed for 15 minutes.

Path+manner Predicates (Talmy 2000) 1/2

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

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

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

Path+manner Predicates (Talmy 2000) 1/2

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

(81) 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

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

(83) 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

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

(85) 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

(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;
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.

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

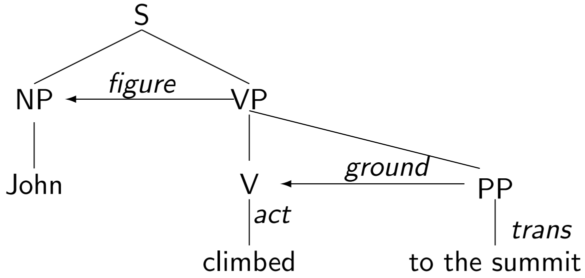
Path+manner Predicates (Talmy 2000) 2/2

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

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

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

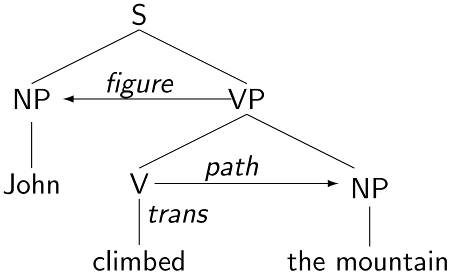
(94)



S
NP
figure
VP
John
V
ground
PP
act
trans
climbed
to the summit

With Path Argument

(95)



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.

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

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

$x \neq y?$

Directed Motion

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

$x{\neq}y?$

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

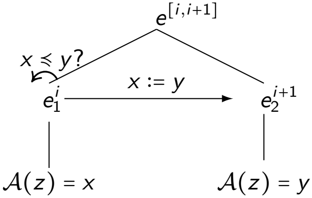
(100) $\overset{\overset{x \neq y?}{\curvearrowleft}}{\boxed{loc(z) = x}}_{e_1} \xrightarrow{\nu} \boxed{loc(z) = y}_{e_2}$

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

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

Directed Motion

(102)



Change and Directed Motion

- 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

(103) MOTION LEAVING A TRAIL:

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

$loc(x) := y$

Motion Leaving a Trail

(104) 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);

$b := y$

(105) 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

(106) 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

(107) 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;

Motion Leaving a Trail

(108) 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).

Quantifying the Resulting Trail

chartParse.pdf

Quantifying the Resulting Trail

chartParse.pdf

Quantifying the Resulting Trail

chartParse.pdf