Frame theory with first-order comparators: Modeling the lexical meaning of verbs of change with frames

Sebastian Löbner, Düsseldorf

1. Background

1.1 The Düsseldorf/CRC 991[1] frames program

- **Frame Hypothesis** (Barsalou, 1992)
  Frames (i.e. recursive attribute-value structures with constraints) constitute the **universal format** of human cognitive representation.
- Attributes in frames are **functional**, one- or more-place.
- Any frame analysis strives for **cognitive plausibility**, aiming at a model of real cognitive representation.
- A **global ontology** serves as a model for all frames.

1.2 A global ontology for frames

The ontology is basically defined in terms of attributes. The universe is sorted (for convenience; the sorts are possibly derivable from the attributes).

DEFINITION 1. **First-order sorted ontology**
A first-order sorted **ontology** $\mathfrak{O}$ is a quadruple $\langle U, \mathcal{A}, \mathcal{S}, \mathcal{C} \rangle$ such that

a. U, the **universe**, is a non-empty set of individuals, the universe.

b. $\mathcal{A}$, the set of **attributes**, is a set of non-empty partial functions $a: U^n \longrightarrow U$.

c. $\mathcal{S}$, the system of **sorts**, is a partition of U. The attributes respect sorts:
for every $a: U^n \longrightarrow U$, there are sorts $s, s_1, \ldots, s_n \in \mathcal{S}$ such that $dom_i(a) \subseteq s_i$ for $i = 1,\ldots,n$ and $cod(a) \subseteq s$. [2]

d. $\mathcal{C}$, the set of **classes**, is a proper subset of $\wp(U)$.
For every $c \in \mathcal{C}$, there is an $s \in \mathcal{S}$ with $c \subseteq s$. $\mathcal{S} \subseteq \mathcal{C}$.
For every $x \in U$, $\{x\} \in \mathcal{C}$. $\{\{x\} \mid x \in U\}$ is the set of atoms in $\mathcal{C}$.

Closure conditions on the set $\mathcal{A}$ of attributes

e. $\mathcal{A}$ is closed under functional composition.

f. If $a \in \mathcal{A}$ is injective, there is a partial function $a^{-1} \in \mathcal{A}$, such that $\forall x, y \in U$, $a^{-1}(y) = x$ iff $a(x) = y$.

Closure conditions on the set $\mathcal{C}$ of classes

g. If $a \in \mathcal{A}$, $c \in \mathcal{C}$, $c \subseteq dom(a)$, then $a[c] \in \mathcal{C}$
if $a \in \mathcal{A}$, $c \in \mathcal{C}$, $c \subseteq cod(a)$, then $a^{-1}[c] \in \mathcal{C}$

h. For every $c, c' \in \mathcal{C}$, $c \cap c' \in \mathcal{C}$.

Remark: Sorts and classes may carry additional structure; they may, e.g., be linearly ordered or form a joint semilattice.

---

[1] CRC 991 “The Structure of Representations in Language, Cognition, and Science”

[2] dom(a) is the domain of the function a, i.e. the set of all individuals a returns a value for.
$dom_i(a)$ is the i-th projection of the domain of a, if a is n-place, $1 \le i \le n$.
cod(a) is the codomain of a, i.e. the set of all individuals x that are values of a for some n-tuple of arguments.
a[c], for a class c, is the image of c under a; $a^{-1}[c]$ is the preimage of c under a.

1.3 Frames

DEFINITION 2. **Frame structure related to the ontology 𝔇**
A frame is a connected di[rected]graph with (optional) labels on vertices and mandatory labels on arcs; for a frame on the ontology 𝔇, the labels denote classes and attributes in the ontology.

A frame structure related to the ontology $\mathfrak{D} = \langle U, \mathcal{A}, \mathcal{S}, \mathcal{C}\rangle$ is a structure ⟨V, A, att, C, cl⟩, such that

a. **V** is a finite non-empty set of **indices** (vorteces). The indices are variables or names for individuals in U.

b. **A** is a finite set of **attribute labels** that denote attributes in $\mathcal{A}$. **att** is a function that maps pairs of an n-tuple of indices (n≥1) and an attribute label to an index. All indices are arguments or values of att.

c. C is a set of class labels that denote classes in $\mathcal{C}$; cl is a partial function that assigns **class labels** to indices. All class labels are values of cl for some index..

d. With $E =_{df} \{\langle \vec{x}, y\rangle \mid \exists a \in A\ att(\vec{x}, a) = y\}$, ⟨V, E⟩ is a **connected digraph** with n-to-1 hyperedges.

1.4 Frames and first order predicate logic

DEFINITION 3. **PL1 language associated with an ontology** $\mathfrak{D} = \langle U, \mathcal{A}, \mathcal{S}, \mathcal{C}\rangle$
PL-𝔇 is a first-order predicate logic language with the following elements:

a. individual terms, including individual variables and individual constants for individuals in U

b. class constants: terms for classes in $\mathcal{C}$

c. n-place function constants: terms for the attributes in $\mathcal{A}$

d. ∈ for statements of the form ⌜t ∈ c⌝, with individual term t and class term c.

e. = for statements of the form ⌜$t_1 = t_2$⌝ with individual terms t_1, t_2.

f. ∧ propositional conjunction

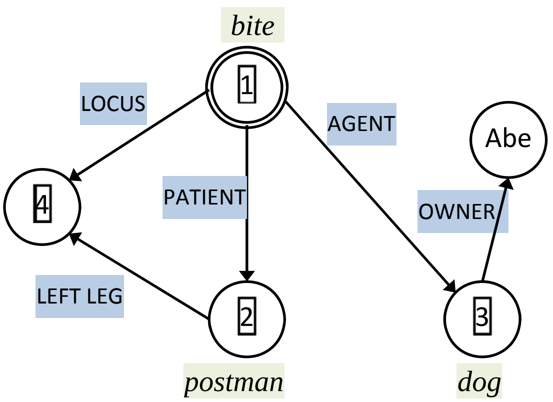
g. ∃ existential quantifier

Remark: The ontology 𝔇 provides a standard model for PL-𝔇.

DEFINITION 4. **Canonical satisfaction formula**
If ⟨ I, A, C, att, cl⟩ is a frame based on the ontology $\mathfrak{D} = \langle U, \mathcal{A}, \mathcal{S}, \mathcal{C}\rangle$, the canonical satisfaction formula is the conjunction of the following statements:

(i) ⌜$a(i_1, \dots, i_n) = j$⌝ if $att(i_1, \dots, i_n, a) = j$

(ii) ⌜i∈c⌝ if cl(i) = c.



graph definition	satisfaction formula
cl('$\boxed{1}$') = '*bite*'	$\boxed{1}$ ∈ *bite*
att('$\boxed{1}$', 'AGENT') = '$\boxed{3}$'	∧ AGENT($\boxed{1}$) = $\boxed{3}$
att('$\boxed{1}$', 'PATIENT') = '$\boxed{2}$'	∧ PATIENT($\boxed{1}$) = $\boxed{2}$
att('$\boxed{1}$', 'LOCUS') = '$\boxed{4}$'	∧ LOCUS($\boxed{1}$) = $\boxed{4}$
cl('$\boxed{2}$') = '*postman*'	∧ $\boxed{2}$ ∈ *postman*
att('$\boxed{2}$', 'LEFT LEG') = '$\boxed{4}$'	∧ LEFT LEG($\boxed{2}$) = $\boxed{4}$
cl('$\boxed{3}$') = '*dog*'	∧ $\boxed{3}$ ∈ *dog*
att('$\boxed{3}$', 'OWNER') = 'Abe'	∧ OWNER($\boxed{3}$) = Abe

Remark: A frame structure is essentially a two-dimensional expression (in need of interpretation).

Typographic conventions : ATTRIBUTES *classes/sorts* vertex indices

2. Comparators

- Comparators are two-place attributes on S×S for arbitrary sorts S. They return **comparison values**.
- **Comparison values** like '=' | '≠'; '<' | '=' | '>'; '⊏' | '⊐' are not relations; they are atomic objects ontologically of the same status as truth-values (actually truth-values *are* comparison values). The values that a comparator returns must be mutually exclusive (otherwise the comparator is not a function). If a comparator returns '<', '=', or '>', it cannot return '≤' (but a different comparator may return either '≤' or '>'). We may use numbers to encode comparison values, but they won't carry their algebraic meanings.
- Comparators are **cognitively plausible** operators: cognitive systems (even the most primitive perceptual systems) are able to compare their input. Categorization employs the standard '= | ≠' comparator.
- Depending on the sort, there may be more than one comparator available.
- Comparators allow the modeling of elementary 2-place relations with 2-place functions, e.g. equality, scalar ordering, merological ordering, topological relations, etc.

DEFINITION 5. **First-order comparators**
First-order comparators in an ontology $\mathfrak{O} = \langle U, \mathcal{A}, \mathcal{S}, \mathcal{C} \rangle$ are two-place attributes with both arguments of the same sort; they return comparison values.

a. For every sort $s \in \mathcal{S}$, the standard comparator $©_s$ is defined as
$©_s(x, y) =_{df} 1 \mid 0$ iff $x = \mid \neq y$

b. If $<$ is a partial or total ordering on $s \in \mathcal{S}$, $©_{s,<}$ is defined as
$©_{s,<}(x, y) =_{df} 0 \mid 1 \mid 2$ iff $x < \mid = \mid > y$

Remark: In the associated PL1 language, one can use the conventional notation for comparisons – e.g. '$x > y$' for '$©_{s<}(x, y) = 2$' – extending the syntax of the PL1 language appropriately.

3. Time

Introduction of time into the ontology allows us to approach the temporal structure of events in lexical verb semantics (and derivatively, in compositional semantics).

3.1 Tensed ontologies

- Introduce a sort *time*. Times are intervals on the time axis.
- Introduce functions that map events on times.
- Introduce time-dependency for attributes whose values change in time.

DEFINITION 6: **Tensed ontology**
A tensed ontology $\mathfrak{O}^t = \langle U, \mathcal{A}, \mathcal{S}, \mathcal{C} \rangle$ fulfills the following additional conditions:

a. There is a sort *time* in $\mathcal{S}$, of **time intervals** (including points in time) with the usual properties and the (mutually exclusive) temporal Allen relations.

b. There is a comparator $©_{Allen}$ that assigns values for the Allen relations to pairs of times.

c. There is a sort *event* in $\mathcal{S}$, of **events**.

d. Three attributes map events on time: TE(e) = the time occupied by the event e (a.k.a. τ(e)),
TB(e) = the time before the event e, TA(e) = the time after the event e;
for every event e: TB(e) < TE(e) < TA(e); the three times need not be adjacent..

e. **Time-dependent attributes**: There are two-place attributes a: $s_1 \times time \longrightarrow s_2$, for some sorts s_1, s_2 (e.g. WEIGHT, PRICE, TEMPERATURE, AGE).

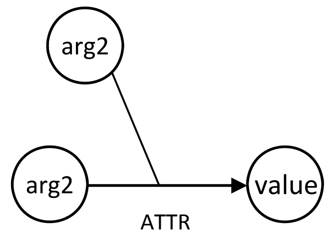
f. **Homogeneity** condition for attributes with a time argument:
If an attribute a assigns the value v to a time t (and possibly further arguments), then a assigns the same value v to all non-empty subintervals of t.

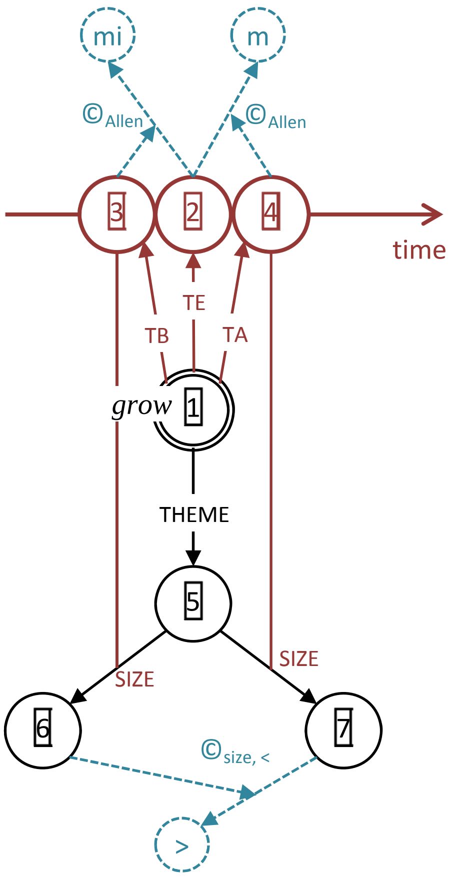
3.2 Lexical frames for dynamic verbs

The use of temporal comparators in a tensed ontology allows the frame representation of the situation structure of certain aspectual types of verbs, in particular of verbs of change.

Example: A frame/AVM for the lexical meaning of *grow*
The time axis is not part of the frame; it is added for better readability.[3]

Diagram for two-place attributes





satisfaction formula (canonical)

[1] ∈ *grow*
∧ TE([1]) = [2]
∧ TB([1]) = [3]
∧ TA([1]) = [4]
∧ THEME([1]) = [5]
∧ $©_{Allen}$([2], [3]) = mi
∧ $©_{Allen}$([2], [4]) = m
∧ SIZE([5], [3]) = [6]
∧ SIZE([5], [4]) = [7]
∧ $©_{size, <}$([7], [6]) = 2

satisfaction formula (simplified equivalent)

[1] ∈ *grow*
∧ TE([1]) mi TB([1])
∧ TE([1]) m TA([1])
∧ SIZE(THEME([1]), TA([1])) >
SIZE(THEME([1]), TB([1]))

[1] *grow*
TE [2] *time*
TB [3] [*time* / $©_{Allen}$([2]) m]
TA [4] [*time* / $©_{Allen}$([2]) mi]
THEME [5] [*object* / SIZE([3]) [6] *size* / SIZE([4]) [7] [*size* / $©_{size,<}$([6]) 2]]

Matrix representation

- use 'attr(arg1)' for 'λy attr(arg1, y)'
- add *class/sort/type* information: this information is derivative of the attributes, hence redundant

[3] 'm' denotes Allen's "meet" relation: t m t' iff$_{df}$ t is wholly before t' and adjacent to t'; 'mi' denotes the inverse of m.