

# REPRESENTATION OF CONCEPTS AS FRAMES

Wiebke Petersen

University of Düsseldorf

Germany

*petersew@uni-duesseldorf.de*

## ABSTRACT

Concepts can be represented as frames, i.e., recursive attribute-value structures. Frames assign unique values to attributes. Concepts can be classified into four groups with respect to both relationality and referential uniqueness: sortal, individual, proper relational, and functional concepts. The paper defines frames as directed graphs with labeled nodes and arcs and it discusses the graph structures of frames for sortal and relational concepts. It aims at a classification of frame graphs that reflects the given concept classification. By giving a new definition of type signatures, the status of attributes in frames is clarified and the connection between functional concepts, their sortal uses, and their associated attributes is explained.

**Keywords:** Frames, knowledge representation, cognitive linguistics

## 1. Introduction

According to Barsalou (1992), frames, understood as recursive attribute-value structures, are used as a general format in accounting for the content of mental concepts. The attributes in a concept frame are the general properties or dimensions by which the respective concept is described (e.g., COLOR, SPOKESPERSON, HABITAT ...).<sup>1</sup> Their values are concrete or underspecified specifications (e.g., [COLOR: **red**], [SPOKESPERSON: **Ellen Smith**], [HABITAT: **jungle**] ...). For example, *ball* can be characterized by [SHAPE: **round**], specifying its concrete shape, and [COLOR: **color**], specifying that it has a color which is not further specified. The attribute values can themselves be complex frames and thus described by additional attributes. E.g., the value **jungle** of the attribute HABITAT can be further specified by attributes like AVERAGE TEMPERATURE OF RAIN SEASON. Frames are thus recursive, and it is this feature that renders them flexible enough to represent information of any desired grade of detail.

Barsalou & Hale (1993) argue that frame theory is independent with respect to various theories of categorization such as checklist theory (cf. Katz 1972; Lyons 1977), exemplar theory (cf. Rosch & Mervis 1975; Brooks 1978), prototype theory (cf. Rosch 1973, 1975; Smith & Medin 1981) or connectionist networks (cf. McClelland & Rumelhart 1981; Shanks 1991). Frames are rather

---

<sup>1</sup> Throughout the paper we will mark types by using small, bold letters, while attributes are written in small capitals.

a model for the representation of concepts and therefore establish an alternative to pure feature-list representations. The advantage of frames over predicates of First Order Logic is that they do not force one to stipulate a fixed arity and that substructures can be addressed via labeled symbols instead of ordered argument positions.

Being motivated primarily by empirical research, Barsalou's focus in developing his frame theory was not on giving a formal theory. However, a formal theory of frames is necessary if they are to be employed in knowledge management or language-processing systems and it is the project of developing such a formal theory that concerns us here. For our account of concept decomposition we will use Barsalou's (1992) cognitive frame theory as a starting point. We will show how frames can be represented by labeled graphs and will establish a type system based on them. Our aim is to develop a formal theory of frames that enables us to describe all kinds of concepts and that is plausible as an adequate basis for a frame-based cognitive semantics explaining both decompositional and compositional phenomena in a unified way.

In aiming at the decomposition of *concepts* that are expressible by nouns, our approach aligns with well-established graph-based knowledge representation formalisms that focus on *situations* such as frame semantics (Fillmore 1982) and on *propositions* as in conceptual graph theory (Sowa 1984).

### ***1.1. Frame-based knowledge representation***

Frame structures appeared in several disciplines in the 1970s. In Cognitive Science, their introduction led to a paradigm change (cf. Fahlmann 1977; Minsky 1975): Instead of being taken as atomic units, concepts came to be understood as classes of highly structured entities describable in terms of recursive attribute-value structures. Feature lists and binary features represented a preliminary stage in this process (cf. Chomsky & Halle 1968). The frame perspective also became prominent in Artificial Intelligence (AI) and Linguistics. One of the best-known frame-based knowledge representation languages of AI is KL-ONE (Brachman & Schmolze 1985), which is the predecessor of a whole family of knowledge representation languages, the so-called description logics (cf. Donini et. al. 1996; Baader et. al. 2004).

In Linguistics, frames were first introduced in Fillmore's case grammar in order to represent verbs and the relational roles of their arguments (Fillmore 1968). This early work laid the foundations for the development of frame semantics (Fillmore 1982). Kay (1979) introduced the idea of describing language signs with complex frame structures and proposed frame unification for their manipulation. These frame structures are now known in Computational Linguistics (CL) as feature structures and are heavily used in unification-

based grammars (cf. Shieber et. al. 1983; Shieber 1986). Inspired by the work of Ait-Kaci on  $\psi$ -terms (Ait-Kaci 1984), type hierarchies with appropriateness conditions were introduced in CL in order to restrict the set of admissible typed feature structures (Carpenter 1992).

Further knowledge representation structures that are related to frames are Semantic Networks (cf. Quilian 1968; Helbig 2006) and Conceptual Graphs (cf. Sowa 1984, 2000).

## 1.2 A classification of concepts

Concepts can be distinguished with respect to both their relationality and their referential uniqueness (Löbner 1985). Sortal and individual concepts are non-relational and thus typically have no possessor argument. Sortal concepts (e.g., *apple*) denote classical categories and have no unique referents. Individual concepts (e.g., *Mary*), in contrast, have unique referents. Proper relational and functional concepts are both relational in that their referents are given by a relation to a possessor (e.g., *brother of Tom*, *mother of Tom*). It is characteristic of functional concepts (e.g., *mother*) that they establish a right-unique mapping from possessors to referents and thus are uniquely referring.<sup>2</sup> In contrast, unique reference is not generally implied for proper relational concepts (e.g., *brother*). Figure 1 shows the resulting concept classification.

The meaning of a given concept may be shifted: E.g., the concept *mother* which, in its normal use, is uniquely referring and relational (*the mother of Tom*) and thus functional can be also used in contexts like *Mothers like gambling* or *The mothers of the constitution were wise*, where it is used as a sortal or proper relational concept, respectively. Such meaning shifts are always context triggered.

	non-unique reference	unique reference
non-relational	SC: sortal concepts: person, house, verb, wood	IC: individual concepts: Mary, pope, sun
relational	RC: proper relational concepts: brother, argument, entrance	FC: functional concepts: mother, meaning, distance, spouse

Figure 1: classification of concepts

Most languages reflect the classification of concepts. E.g., in English, nouns expressing concepts without unique reference (SCs and RCs) are usually used

<sup>2</sup> Note that throughout this paper the term *functional concept* is always used in the sense of describing a concept that establishes a *functional mapping*. Hence, *functional* in this paper does not mean that the concept denotes objects which have a special function.

without a definite article. Nouns expressing relational concepts (RCs and FCs) are usually used in possessive constructions, where the possessor is specified synthetically (*the cat's paw*) or analytically (*the paw of the cat*). However, there is a considerable variation in the expression of definiteness and possession across languages.<sup>3</sup>

## 2. Frame graphs

Our concept-decomposition framework should be formally explicit and cognitively adequate. Therefore, we aim at keeping our frame model as simple and rigid as possible. We do not want to introduce any elements into our model language for merely technical or computational reasons. In Petersen & Werning (2007) we explain how our frame model can be extended to account for cognitive typicality effects. By using oscillatory neural networks as a biologically motivated model, we show how frames might be implemented in the cortex.

Since frames for concepts are recursive attribute-value-structures, each attribute of a frame establishes a relation between the objects denoted by the concept and the value of the attribute; e.g., the attribute *SEX* in the frame for *woman* assigns the value **female** to each denoted object. In accordance with the examples in Barsalou (1992), we assume that attributes in frames assign unique values to objects and thus describe functional relations. The values themselves can be complex frames. Section 3 discusses attributes in frames in greater detail.

We model frames as connected directed rooted graphs with labeled nodes (types) and arcs (attributes). Our definitions follow the notational conventions in Carpenter (1992).

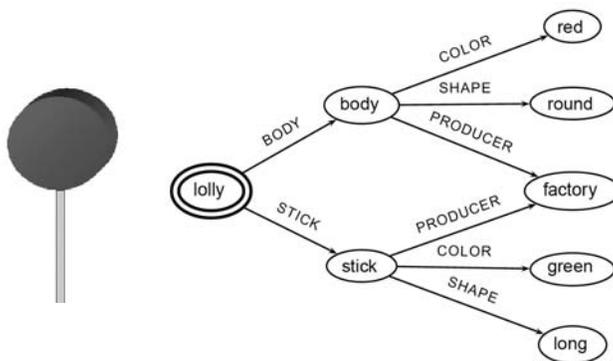


Figure 2: lolly frame

<sup>3</sup> The four concept classes (sortal, individual, proper relational, and functional) are mutually exclusive and jointly exhaustive. For more details on our concept classification, on its linguistic reflections, and on context-triggered meaning-shifts have a look at the webpages of the research group FOR600 *Functional Concepts and Frames* (<http://www.phil-fak.uni-duesseldorf.de/fff>).

Figure 2 shows the graph of an example frame representing knowledge about lollies with long green sticks and round red bodies produced in factories. The double-encircled node **lolly** is the central node of the frame; it shows that the graph represents a frame about lollies. The outgoing arcs of the **lolly**-node stand for the attributes of the represented lollies and point to their values. Hence, each denoted lolly has a stick and a body. The values of the attributes **BODY** and **STICK** are themselves complex frames, both having three attributes, namely **COLOR**, **SHAPE**, and **PRODUCER**. The fact that the stick and the body of each denoted lolly are produced in the same factory is indicated by the single **factory**-node to which the two **PRODUCER**-arcs from **stick** and **body** point. The single **factory**-node excludes the possibility that the body is produced in a candy factory in Belgium while the stick is produced in a paper mill in Canada.

**Definition 1:** Given a set **TYPE** of types and a finite set **ATTR** of attributes. A frame is a tuple  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  where:

- $Q$  is a finite set of nodes,
- $\bar{q} \in Q$  is the central node,
- $\delta : \text{ATTR} \times Q \rightarrow Q$  is the partial transition function,
- $\theta : Q \rightarrow \text{TYPE}$  is the total node typing function,
- $\leftrightarrow \subseteq Q \times Q$  is the symmetric and anti-reflexive inequality relation.

Furthermore, the underlying undirected graph  $(Q, E)$  with edge set  $E = \{\{q_1, q_2\} \mid \exists a \in \text{ATTR} : \delta(a, q_1) = q_2\}$  is connected.

The underlying directed graph of a frame is the graph  $(Q, \vec{E})$  with edge set  $\vec{E} = \{(q_1, q_2) \mid \exists a \in \text{ATTR} : \delta(a, q_1) = q_2\}$ .

If  $\theta(\bar{q}) = t$ , we say that the frame is of type  $t$ ; and if  $\theta(q) = t$  is true for a node  $q$ , we call the node  $q$  a  $t$ -node. Furthermore, if  $\delta(a, q_1) = q_2$  is true for a frame, we say that the frame has an  $a$ -arc from  $q_1$  to  $q_2$ ; the  $a$ -arc is an *outgoing arc* for node  $q_1$  and an *incoming arc* for  $q_2$ . Contrary to other frame definitions, we do not demand that all nodes of a frame can be reached via directed arcs from its central node.<sup>4</sup>

The types are ordered in a type hierarchy, which induces a subsumption order on frames: “We think of our types as organizing feature structures into natural classes. [...] Thus it is natural to think of the types as being organized in an inheritance hierarchy based on their generality”, (Carpenter 1992: 11).

<sup>4</sup> The claim that all nodes of a frame can be reached from its central node is common in most frame theories (cf. Carpenter 1992; Barsalou 1992) because they usually consider only frames for sortal concepts.

**Definition 2:** A type hierarchy  $(\text{TYPE}, \sqsupseteq)$  is a finite partial ordered set which forms a join semilattice, i.e., for any two types there exists a least upper bound. A type  $t_1$  is a subtype of a type  $t_2$  if  $t_1 \sqsupseteq t_2$ .

**Definition 3:** Given a type hierarchy  $(\text{TYPE}, \sqsupseteq)$  and a finite set  $\text{ATTR}$  of attributes. A frame  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  subsumes a frame  $F' = (Q', \bar{q}', \delta', \theta', \leftrightarrow')$ , notated as  $F \sqsubseteq F'$ , if and only if there exists a total function  $h: Q \rightarrow Q'$  such that:

- $h(\bar{q}) = \bar{q}'$ ;
- if  $q \in Q$ ,  $a \in \text{ATTR}$ , and if  $\delta(a, q)$  is defined, then  $h(\delta(a, q)) = \delta'(a, h(q))$ ;
- for each  $q \in Q$ :  $\theta(q) \sqsubseteq \theta'(h(q))$ ;
- if  $q_1 \leftrightarrow q_2$ , then  $h(q_1) \leftrightarrow' h(q_2)$ .

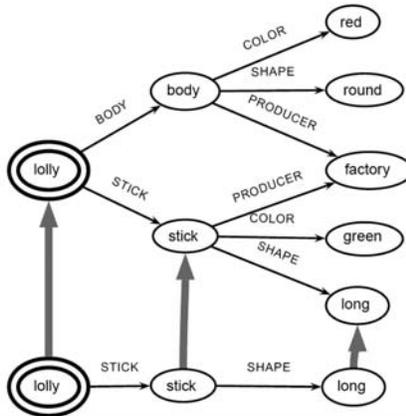


Figure 3: subsumption example

An example of the subsumption relation is given in Figure 3: It shows a rather unspecific *lolly*-frame subsuming the more specific *lolly*-frame from Figure 2. Bold arrows mark the function  $h$  from Definition 3. The example shows that Definition 3 captures our general understanding of subsumption: When a concept A subsumes a concept B then A is more general than B, i.e., A imposes less restrictions on the objects it denotes than B.

The definition of frames as labeled graphs yields the problem that two frames with different node sets are always different, even if all their labels match. E.g., the two frames  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  and  $F' = (Q', \bar{q}', \delta', \theta', \leftrightarrow')$  with  $Q = \{a, b\}$ ,  $\bar{q} = a$ ,  $\delta(G, a) = b$ ,  $\theta(a) = s$ ,  $\theta(b) = t$  and  $Q' = \{c, d\}$ ,  $\bar{q}' = c$ ,  $\delta'(G, c) = d$ ,  $\theta'(c) = s$ ,  $\theta'(d) = t$  are unequal due to the different node sets ( $Q \neq Q'$ ) although they can both be drawn as  $\textcircled{s} \xrightarrow{G} \textcircled{t}$ . Since the two frames  $F$  and  $F'$  subsume each other, i.e.,  $F \sqsubseteq F'$  and  $F' \sqsubseteq F$ , the subsumption relation defines no partial order on frames, but merely a preorder.

Looking at the lolly example in Figure 2 it is obvious that the information represented in a frame does not depend on the concrete set of nodes. It depends rather on how the nodes are connected by directed arcs and how the nodes and arcs are labeled. However, it is not possible to simply replace the nodes in the frame definition by their labels, since two distinct nodes of a graph can be labeled with the same type. E.g., we could modify the *lolly*-frame in Figure 2 so that the stick and the body of the described lollies were produced in two distinct factories, where one is located in Belgium and one in Canada. The frame in the middle of Figure 4 shows another example of a frame with several equally typed nodes. Therefore, it is convenient to define the *alphabetic variance* relation: A frame  $F$  is an *alphabetic variant* of a frame  $F'$  (written as  $F \sim F'$ ) if and only if  $F \sqsubseteq F'$  and  $F' \sqsubseteq F$  are both true. The alphabetic variance relation is an equivalence relation over the collection of frames. It follows immediately that subsumption modulo the alphabetic variance relation defines a partial order on the equivalent classes of frames. From now on, to simplify matters, we will not distinguish between a frame and its equivalence class under alphabetic variance.

In order to characterize the graphs underlying frames, we use the following terminology:

**Definition 4:** A node  $q_0 \in Q$  is said to be a root of a frame  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  if for each  $q \in Q$  there is a finite sequence of attributes  $a_1 \dots a_n \in \text{ATTR}^*$  with  $\delta(a_n, \dots, \delta(a_2, \delta(a_1, q_0) \dots)) = q$ , i.e.,  $q_0$  and  $q$  are connected by a finite directed path.

**Definition 5:** A node  $q \in Q$  is said to be a source of a frame  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  if  $q$  has no incoming arc (i.e.,  $q$  has indegree 0). Analogously,  $q$  is a sink of a frame if  $q$  has no outgoing arc (i.e.,  $q$  has outdegree 0).

The frame in Figure 2 has exactly one source, namely the node labeled **lolly**, and five sinks, i.e., the nodes labeled **red**, **round**, **factory**, **green**, **long**. The source of this frame is simultaneously a root of the frame.

A frame is said to be *acyclic* if the underlying directed graph is acyclic, i.e., if it is not possible to find a way along directed arcs leading from a node back to itself. It is obvious that an acyclic frame has at most one root. Our experience in decomposing concepts into frames indicates that frames for concepts are generally acyclic. Through our involvement in the research group FOR600 *Functional Concepts and Frames*, we have access to more than a hundred frame graphs of different concepts that were drawn by approximately twenty (test) persons; none of the frames are cyclic. However, there are some rare self-referential concepts like *egoist* or *narcissist* whose frame graphs have to be cyclic.<sup>5</sup> In spite of these exceptions, we consider only concepts with acyclic frames in this paper.

<sup>5</sup> Thanks to Magdalena Schwager for pointing out the problem of self-referential concepts to us.

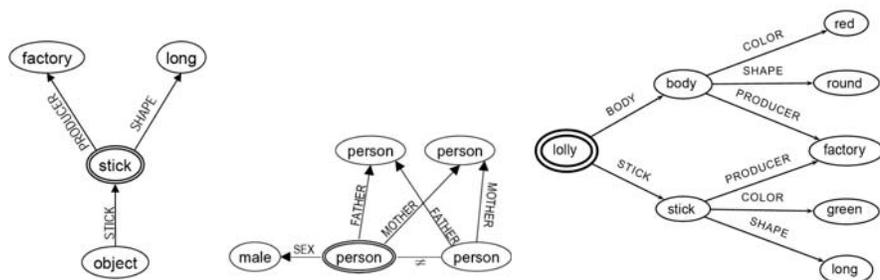


Figure 4: frames for different concepts (left: *stick*; middle: *brother*; right: *lolly*)

Figure 4 shows three frames belonging to concepts of three different concept classes (again the central nodes are double-encircled).<sup>6</sup> The right frame for the sortal concept *lolly* has already been discussed above. The left frame represents the functional concept *stick* and the frame in the middle corresponds to the proper relational concept *brother*. The *stick*-frame characterizes a stick by *being the stick of an object* (i.e., by a functional relation) and by additional sortal features like *being long* and *being produced in a factory*. Functional concepts differ fundamentally from sortal concepts, since their potential referents are the values of an attribute which is identical with the functional concept. Although the *stick*-frame seems to be a substructure of the frame for the sortal concept *lolly*, the fundamental difference is encoded inherently in the graph structure of the frames: The central node of the functional frame, i.e., the frame for the functional concept *stick*, has an incoming arc while that of the sortal frame for *lolly* has solely outgoing arcs. Both frames characteristically have a root. It is the incoming arc (labeled by an attribute corresponding to a functional concept) which establishes the functional relation from potential possessors to the referents of the functional concept.

The frame for the proper relational concept *brother* is more complex. It describes a brother as a male person for which a second person exists with whom it shares mother and father. The undirected arc between the two **person**-nodes labeled with  $\neq$  indicates the inequality relation and ensures that the two nodes can never be unified.<sup>7</sup> The peculiarity of this frame is that the two nodes labeled **person** cannot be reached along directed paths from each other and that there is no third node from which both nodes can be reached. Thus, the potential referents of the central **person**-node are characterized by the sortal feature *male* and especially by the existence of referents for the non-central source of type **person**, which represents the possessor

<sup>6</sup> Throughout this paper we do not deal with individual concepts since they require a rather different treatment: The graphs underlying their frames do not differ but their central nodes are not labeled by arbitrary types but by particular entity types. Petersen & Werning (2007) give some examples of frames for individual concepts.

<sup>7</sup> The inequality relation becomes important as soon as information combining procedures like unification are applied to frames, as these procedures have to preserve explicitly stated inequalities.

argument of the proper relational concept *brother*. The connection between the central node and the node for the possessor argument is established indirectly via the shared values of the FATHER- and MOTHER-attributes. Since the relation between a person and his or her mother (or father) is a many-to-one relation, the *brother*-frame does not set up a functional relation between the possessor argument and the referents of the central node. It is characteristic for a *proper relational frame*, i.e., a frame for a proper relational concept, that it has a node which is a source but from which the central node is not reachable along directed arcs.

The example frames show that what type of concept is represented by an acyclic frame is determined by the properties of the central node and the question whether or not the frame has a root or a source. In the remainder of this section we therefore use the eight binary features “± has source” (±ES), “± has root” (±ER), “± central node is a source” (±CS), and “± central node is a root” (±CR) to classify directed acyclic graphs with central nodes. In order to gain a complete list of possible classes we apply the attribute exploration technique known from Formal Concept Analysis (FCA) (Ganter & Wille 1999), which is implemented in the software *Concept Explorer*.<sup>8</sup> During an attribute exploration, *Concept Explorer* successively presents implications of properties (in the terminology of FCA: attribute implications) which the user must either accept or reject (by offering a counter example). The process ends when the canonical universe of the properties is completed (Osswald & Petersen 2003), i.e., the closure of sets of compatible properties is determined. The procedure guarantees that the number of implications presented is minimal. Figure 5 shows the result of the exploration: The implicational statements on the left are those which we affirmed during the exploration process. The resulting concept lattice is given on the right side of the figure.

- (1) necessarily +ES;
- (2) if -ER, then -CR;
- (3) if -CS, then -CR;
- (4) if +ER and -CR, then -CS;
- (5) if +CS and -CR then -ER;
- (6) if +CS and +ER then +CR;
- (7) if +CR then +CS and +ER.

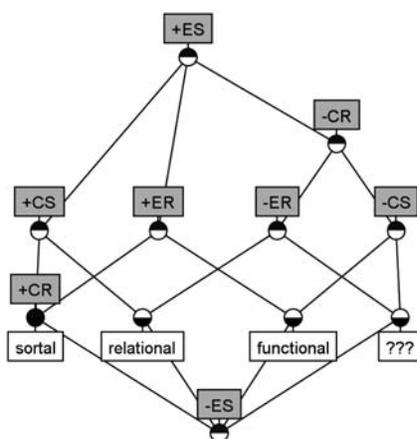


Figure 5: basis of implications and concept lattice

<sup>8</sup> <http://conexp.sourceforge.net/>

The seven implicational statements are true for the following reasons: The second and the third statement follow immediately from the meaning of the existential quantifier and the definition of roots and sources. Since an acyclic directed graph can be physically modeled by a system of tubes where each tube has a slope, the remaining statements can be easily verified: Roots in such a tube system can be easily recognized by the fact that water flows through all tubes if it is poured in at a root. A source in such a tube system is a tube junction that is not reached by water poured into the system at any other point. Since water cannot run upwards, it cannot flow in circles. Hence, there is never more than one root in a tube system; and if there is a root, it is necessarily the only source of the system as well. The statements (1) and (4)-(7) follow immediately from these considerations.

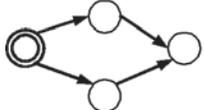
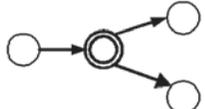
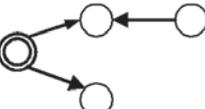
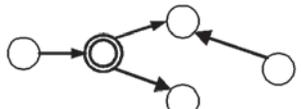
CR	CS	ER	ES	typical graph	frame class
+	+	+	+		sortal
-	-	+	+		functional
-	+	-	+		proper relational
-	-	-	+		???

Figure 6: classification of acyclic frames

Taking into account the implicational statements in Figure 5, there are four property distributions remaining that are consistent. Hence, the chosen properties classify acyclic frames into four classes. Figure 6 lists them and shows a typical graph with the required properties for each distribution.

**Definition 6:** A sortal frame is an acyclic frame whose central node is a root. A relational frame is an acyclic frame with a source which is not the central node. A proper relational frame is an acyclic frame with at least two sources of which one is the central node. A functional frame is an acyclic frame with a root which is not the central node.

From our experience in modeling concepts with frames, we expect that, at least in typical unmarked cases, the first three frame classes in Figure 6 correspond to the concept classes discussed in section 1.2.

**Conjecture:** *In general, sortal concepts are represented by sortal frames, functional concepts are represented by functional frames, and (proper) relational concepts are represented by (proper) relational frames. However, not every arbitrary acyclic frame models a cognitively relevant or even lexicalized concept.*

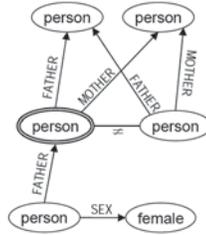


Figure 7: frame for *father of a niece*

The fourth frame class does not correspond as nicely to a concept class as the others do. We assume that frames for this class model non-lexicalized concepts like *father of a niece* whose frame is given in Figure 7.

We only found one frame belonging to the fourth class that seems to be lexicalized, namely the one for *brother-in-law* (and analogically for *sister-in-law*). The corresponding frame is shown on the left in Figure 8.<sup>9</sup> Nevertheless, *brother-in-law* only *appears* to be a concept whose frame belongs to the fourth frame class as the right frame of Figure 8 illustrates. Since the *spouse*-relation is a symmetric one-to-one relation, the direction of the *SPOUSE*-arc of the frame can be reversed. The content of the left frame can be paraphrased as *male person who is the spouse of someone who has a sibling* and the right one as *male person whose spouse has a sibling*. Since *brother-in-law* is a proper relational concept that takes one possessor argument (*my brother-in-law*), the paraphrases of the two frames show that the left one analyzes *brother-in-law* incorrectly as a relational concept with an extra argument.

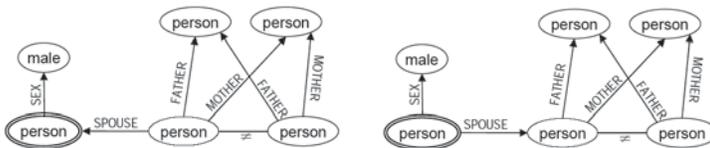


Figure 8: two frames for *brother-in-law*

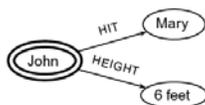
<sup>9</sup> Strictly speaking, *brother-in-law* is polysemous; it means either *brother of spouse* or *husband of sibling*. We only consider the latter meaning here.

From the frames in this section we can draw two main conclusions: First, the arguments of relational concepts are modeled in frames as sources that are not identical to the central node. Second, the functionality of functional concepts is modeled by an incoming arc at the central node.

### 3. Attributes and type signatures

As Guarino (1992) points out, frame-based knowledge engineering systems as well as feature-structure-based linguistic formalisms normally force a radical choice between attributes and types. Therefore, frames like the one in Figure 2 are common, where the rather unspecific value *stick* is assigned to the attribute STICK. The parallel naming of the attribute STICK and the type *stick* suggests a systematic relationship between the attribute and the type that is not captured by the formalism.

A second problem addressed in Guarino (1992) concerns the question which binary relations should be expressed by attributes. If one allows attributes to be unrestricted arbitrary binary relations, this leads to frames like the following one, which was first discussed in Woods (1975):



Although HEIGHT and HIT can be represented by binary predicates, the ontological status of the link established between **John** and **6 feet** and between **John** and **Mary** differs fundamentally.

As stated before, we presuppose that attributes of frames establish many-to-one, i.e., functional relations between the nodes they are attached to and their values. The question arises how attributes and functional concepts are connected. All sample attributes we have used so far (STICK, COLOR, ...) correspond to functional concepts. Guarino (1992) distinguishes between the *denotational* and the *relational* interpretation of a relational concept. This distinction can be used to explain how functional concepts can act as concepts and as attributes: Let there be a universe  $\mathcal{U}$  and a set of functional concepts  $\mathcal{F}$ . A functional concept (like any concept) denotes a set of entities:

$$\Delta: \mathcal{F} \rightarrow 2^{\mathcal{U}}$$

(e.g.,  $\Delta(\text{mother}) = \{m \mid m \text{ is the mother of someone}\}$ ).

A functional concept also has a relational interpretation:

$$\rho: \mathcal{F} \rightarrow 2^{U \times U}$$

(e.g.,  $\rho(\text{mother}) = \{(p, m) \mid m \text{ is the mother of } p\}$ ).

Additionally, the denotational and the relational interpretation of a functional attribute have to respect the following *consistency postulate* (Guarino 1992): Any value of a relationally interpreted functional concept is also an instance of the denotation of that concept. E.g., if  $(p, m) \in \rho(\text{mother})$ , then  $m \in \Delta(\text{mother})$ . Furthermore, the relational interpretation of a functional concept  $f$  is a function, i.e., if  $(a, b), (a, c) \in \rho(f)$ , then  $b = c$ .

These considerations allow us to clarify the ontological status of attributes in frames: Attributes in frames are relationally interpreted functional concepts. Hence, attributes are not frames themselves and therefore are unstructured. Frames decompose concepts into relationally interpreted functional concepts. Thus, functional concepts embody the concept type on which categorization is based. The differentiation between the denotational and the relational interpretation of functional concepts is consistent with Barsalou's view on attributes: "I define an attribute as a concept that describes an aspect of at least some category members. For example, *color* describes an aspect of *birds*, and *location* describes an aspect of *vacations*. A concept is an attribute only when it describes an aspect of a larger whole. When people consider *color* in isolation (e.g., thinking about their favorite color), it is not an attribute but is simply a concept", Barsalou (1992: 30).

In the theory of typed feature structures, it is common to enrich the plain type hierarchy by an appropriateness specification. It regulates which attributes are appropriate for feature structures of a special type and restricts the values of the appropriate attributes (Carpenter 1992).<sup>10</sup> We adapt this technique in order to restrict the class of admissible frames. However, we consequently dismiss the artificial distinction between attributes and types in our definition of type signatures: In contrast to the standard definition (Carpenter 1992: 86) the attribute set is merely a subset of the type set. Hence, attributes occur in two different roles: as names of binary functional relations between types and as types themselves.

**Definition 7:** *Given a type hierarchy  $(\text{TYPE}, \sqsubseteq)$  and a set of attributes  $\text{ATTR} \subseteq \text{TYPE}$  an appropriateness specification on  $(\text{TYPE}, \sqsubseteq)$  is a partial function  $\text{Approp}: \text{ATTR} \times \text{TYPE} \rightarrow \text{TYPE}$  such that for each  $a \in \text{ATTR}$  the following holds:*

<sup>10</sup> Type signatures can be automatically induced from sets of untyped feature structures, i.e., frames where the central node is a root and in which only the maximal paths are typed. With FCAType, an implemented system for such inductions is available (Kilbury et. al. 2006; Petersen 2006).

- attribute introduction: *There is a type*  $\text{Intro}(a) \in \text{TYPE}$  *with:*
  - $\text{Approp}(a, \text{Intro}(a)) = a$  *and*
  - *for every*  $t \in \text{TYPE}$  : *if*  $\text{Approp}(a, t)$  *is defined, then*  $\text{Intro}(a) \sqsubseteq t$ .
- specification closure: *If*  $\text{Approp}(a, s)$  *is defined and*  $s \sqsubseteq t$ , *then*  $\text{Approp}(a, t)$  *is defined and*  $\text{Approp}(a, s) \sqsubseteq \text{Approp}(a, t)$ .
- attribute consistency: *If*  $\text{Approp}(a, s) = t$ , *then*  $a \sqsubseteq t$ .

*A type signature is a tuple*  $(\text{TYPE}, \sqsubseteq, \text{ATTR}, \text{Approp})$ , *where*  $(\text{TYPE}, \sqsubseteq)$  *is a type hierarchy, ATTR*  $\subseteq \text{TYPE}$  *is a set of attributes, and*  $\text{Approp} : \text{ATTR} \times \text{TYPE} \rightarrow \text{TYPE}$  *is an appropriateness specification.*

The first two conditions on an appropriateness specification are standard in the theory of type signatures (Carpenter 1992), except that we tighten up the attribute introduction condition. We claim that the introductory type of an attribute  $a$  carries the appropriateness condition  $a : a$ . With the attribute-consistency condition we ensure that Guarino’s consistency postulate holds and that Barsalou’s view on frames, attributes, and values is modeled appropriately: “At their core, frames contain attribute-value sets. Attributes are concepts that represent aspects of a category’s members, and values are subordinate concepts of attributes”, (Barsalou 1992: 43). Hence, the possible values of an attribute are subconcepts of the denotationally interpreted functional concept. This is reflected in the type signature by the condition that the possible values of an attribute are restricted to subtypes of the type corresponding to the attribute.

We call a frame *well-typed* with respect to a type signature if all attributes of the frame are licensed by the type signature and if additionally the attribute values are consistent with the appropriateness specification.

**Definition 8:** *Given a type signature*  $(\text{TYPE}, \sqsubseteq, \text{ATTR}, \text{Approp})$ , *a frame*  $F = (Q, \bar{q}, \delta, \theta, \leftrightarrow)$  *is well-typed with respect to the type signature, if and only if for each*  $q \in Q$  *the following holds:*

*If*  $\delta(a, q)$  *is defined, then*  $\text{Approp}(a, \theta(q))$  *is also defined and*  $\text{Approp}(a, \theta(q)) \sqsubseteq \theta(\delta(a, q))$ .

The definition of the appropriateness specification guarantees that every arc in a well-typed frame points to a node that is typed by a subtype of the type corresponding to the attribute labeling the arc. The decomposition of concepts into frames requires that the frame in question be well-typed.

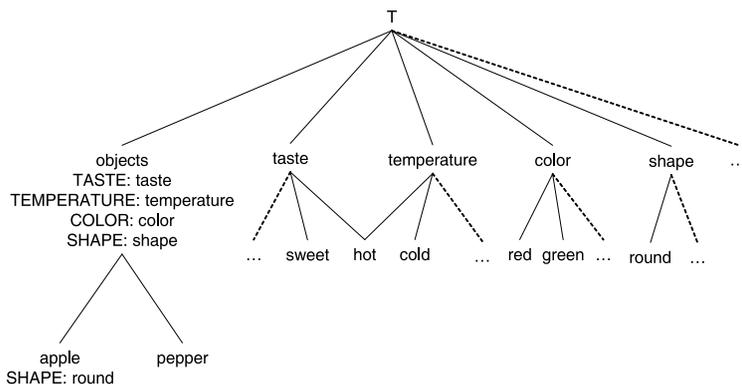


Figure 9: example type signature

A small example type signature is given in Figure 9. The appropriateness specification is split up into single appropriateness conditions: The expression TASTE: **taste** at type **objects** means that the attribute TASTE is appropriate for frames of type **objects** and its value is restricted to frames of type **taste** or subtypes of **taste**. The attribute conditions are passed on downwards. Hence, the type **apple** inherits the appropriateness condition TASTE: **taste** from its upper neighbor **objects**. It also inherits the appropriateness condition SHAPE: **shape**, but tightens it up to SHAPE: **round**, which is permissible by the specification closure condition. The definition of the type signature makes sure that the permissible values of an attribute are subtypes of the attribute type. Hence, the possible values of TASTE, i.e., **sweet**, **hot**, **sour**, and so forth, are subtypes of the type **taste**. Notice that the subtypes of an attribute type are not generally attribute types themselves. Figure 10 shows two frames, where the first one is well-typed while the second one violates the appropriateness condition SHAPE: **round** at type **apple**.

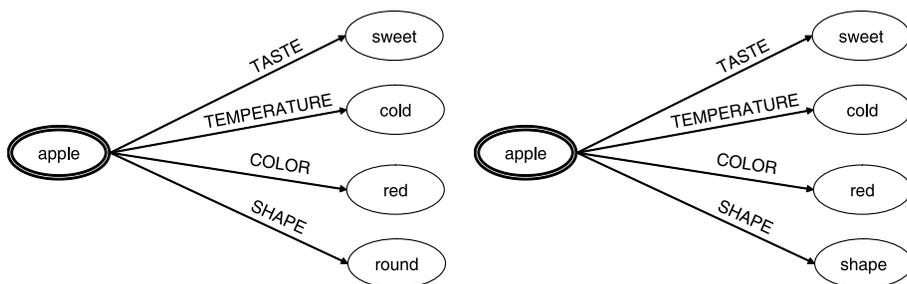


Figure 10: well-typed versus non well-typed frame

Due to the recursive structure of frames, it is possible to specify the attribute values with frames of any desired complexity. Since the length of attribute paths in frames is not restricted, the frames can fan out in depth as depicted in the upper part of Figure 11. However, Barsalou allows frames to grow in an additional respect: “Within a frame, each attribute may be associated with its own frame of more specific attributes. [...] These secondary attributes often have frames as well. [...] Even these attributes [of frames of secondary attributes] continue to have frames”, Barsalou (1992: 33). The possibility of further specifying attributes as well as their values by additional attributes results in the double recursive and self-similar structure of Barsalou’s frames, which is depicted in the lower part of Figure 11.

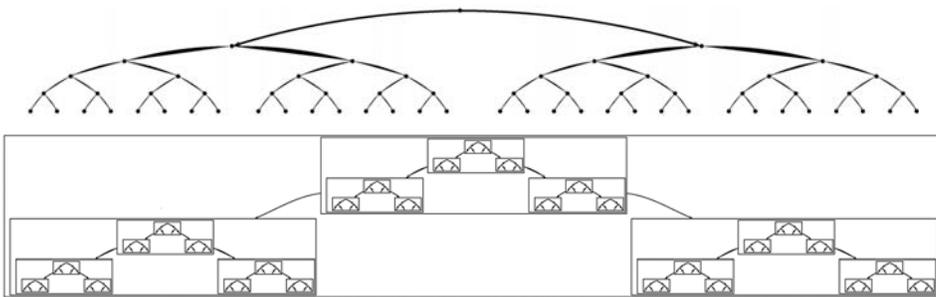


Figure 11: Fanning out of classical frames (top) and Barsalou’s self-similar frames (bottom)

Our approach to frames, which reflects the parallelism of the denotational and the relational interpretation of functional concepts in the definition of type signatures, captures Barsalou’s idea about frames, but avoids the double recursive structure. Since attributes are types at the same time, further attribute-value pairs can specify them; this is in accordance with Barsalou’s claim that “frames represent all types of concepts, whether they are free-standing concepts, such as *bird* and *vacation*, or whether they are attributes such as *color* for *bird* and *location* for *vacation*”, (Barsalou 1992: 31). However, this further specification will only take place if the attribute is used as a type, i.e., if it labels a frame node, and never when it is used as a functional attribute and labels a frame arc. Our *lolly*-frame in Figure 2 exemplifies this perspective: The attribute *stick* labels an arc as well as a node, but it is the value of the attribute to which further attribute arcs are attached, such that it constitutes the sortal *stick*-subframe in Figure 12.

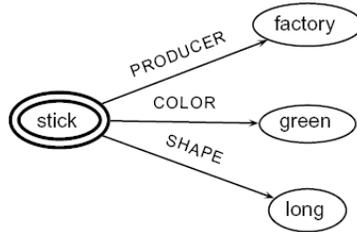


Figure 12: Frame for the sortal reading of *stick*

The attributes *PRODUCER*, *COLOR* and *SHAPE* are attributes of sticks and not of the attribute *STICK*, since *STICK* is the partial function that assigns sticks to objects. Note that the *stick*-frame in Figure 12 differs from the *stick*-frame in Figure 4 (left) in that it does not relate the stick to an object to which it is attached. If it is not embedded into a larger object frame (e.g., a *lolly*-frame) it models the sortal reading of *stick* as in the sentence *these days sticks are mostly produced in big factories*. Such context-triggered meaning-shifts from relational concepts to non-relational readings are very common; the frame structures of the concepts help to explain and visualize them. However, we would like to emphasize that the *stick*-frame of Figure 4 (left) must not be confused with the attribute *STICK* itself: *Stick* is a functional concept whose functional frame is given in Figure 4 (left); although it is functional, it denotes – like the sortal *stick*-frame in Figure 12 – sticks. However, in contrast to the sortal frame its denotation is determined by a functional relation from a possessor argument (here the potential referents of the **object**-node). The attribute *STICK* is the relationally interpreted functional concept *stick* and therefore a function; it is not a frame.

The ability to give explicit frames for functional concepts that differ fundamentally from frames for the sortal readings of those concepts is a novelty. It is made possible by our novel definition of frames, which no longer demands that the central node be a root of the frame graph. We know of no other explicit approach to frames for functional concepts.

Having motivated our approach to type signatures we will now sketch how it offers an elegant solution to problems in grammar engineering that occur when frames are employed as semantic representations. To model how adjectives modify the meaning of a noun it has to be explained that in a phrase like *red body* the value **red** is assigned to the attribute *COLOR*, while *round body* modifies the value of the attribute *SHAPE*. An unsatisfactory solution would be to have a single rule for each adjective dimension, i.e., for each attribute. Instead, we propose to introduce the notion of a *minimal upper attribute* of a type. An upper attribute of a type is an attribute which is a supertype of the type with respect to the type hierarchy. Hence, a minimal upper attribute of a type is a minimal element of the set of upper attributes of the type. According to the type signature

in Figure 9, the minimal upper type of **red** is **color** and the one for **round** is **shape**. Hence, we can formulate a single rule for all those cases: Simplified, it states that in a frame which represents the meaning of a phrase consisting of an adjective and a noun, the type corresponding to the adjective is assigned as a value to the minimal upper attribute of the adjective type. Such a rule would even capture some interesting cases of ambiguity. Consider the polysemous adjective *hot*, which means either *being very warm* or *being very spicy*. In a type hierarchy the type **hot** could be placed such that it is a subtype of the attribute type **temperature** as well as of the attribute type **taste** (cf. Figure 9). Then **hot** has two minimal upper attributes and the above mentioned rule applied to the phrase *hot pepper* would result in two frames: one representing a very spicy pepper [TASTE: **hot**] and one representing a very warm pepper [TEMPERATURE: **hot**], which could be part of a dish. Due to space limits, we cannot go into more detail here, but a publication focusing on this issue is in preparation.

#### 4. Conclusion

We have presented a new approach to frames which discards the claim that the central node of a frame is its root, which is a common claim in standard frame theories. In addition, we have dismissed the artificial distinction between types and attributes in type signatures. Those two adaptations enable us to give a classification of acyclic frame graphs that mirrors the classification of concepts into sortal, proper relational and functional concepts. In particular, the promising fact that in our frame theory the structure of a functional (but also of a proper relational) concept differs fundamentally from that of the corresponding sortal concept assures us that our modifications to standard frame theory can open up new insights into concept decomposition.

However, a lot of work still has to be done. We need to develop a unification operation in order to account for frame composition. Individual concepts also have to be accommodated in our frame theory. Furthermore, we expect the discovery of new ways to explain phenomena from fields such as composition, metonymy, metaphors, and meronymy. Finally, powerful software devices have to be developed in order to test our frame model in real knowledge engineering or language processing tasks.

#### Acknowledgments

I would like to thank my colleagues in the research group *Functional Concepts and Frames* for invaluable discussion and input. Special thanks go to Peter Bücken and Rafael Cieslik for their help editing this text and to an anonymous reviewer for helpful hints concerning its readability. Research for this paper was made possible by the grant FOR600 of the German Research Foundation.

## Bibliography

- Ait-Kaci, H. (1984): *A Lattice Theoretic Approach to Computations Based on a Calculus of Partially Ordered Type Structures*. PhD thesis, University of Pennsylvania.
- Baader, F. & D. Calvanese & D. McGuinness & D. Nardi & P. Patel-Schneider (eds.) (2004): *The Description Logics Handbook*. Cambridge: Cambridge University Press.
- Barsalou, L. W. & C. R. Hale (1993): Components of conceptual representation: From feature lists to recursive frames. In I. Van Mechelen, J. Hapton, R. Michalski, and P. Theuns (eds.): *Categories and concepts: Theoretical views and inductive data analysis*. San Diego, CA: Academic Press. 97-144.
- Barsalou, L. W. (1992): Frames, concepts, and conceptual fields. In A. Lehrer and E. F. Kittay (eds.): *Frames, fields, and contrasts: New essays in semantic and lexical organization*. Hillsdale, NJ: Lawrence Erlbaum Associates. 21-74.
- Brachman, R. J. & J. G. Schmolze (1985): An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2). 171-216.
- Brooks, L. R. (1978): Nonanalytic concept formation and memory for instances. In E. Rosch and B. B. Lloyd (eds.): *Cognition and Categorization*. Hillsdale: Erlbaum.
- Chomsky, N. & M. Halle (1968): *The Sound Pattern of English*. New York: Harper and Row.
- Carpenter, B. (1992): *The Logic of Typed Feature Structures*. Cambridge: CUP.
- Donini, F. M. & M. Lenzerini & D. Nardi & A. Schaerf (1996): Reasoning in description logics. In G. Brewka (ed.): *Principles of Knowledge Representation*. Stanford, CA: CSLI Publications. 193–238.
- Fahlmann, S. E. (1977): *A System for Representing and Using Real-world Knowledge*. PhD thesis, MIT.
- Fillmore, C. J. (1982): Frame semantics. In *Linguistics in the Morning Calm*. Seoul: Hanshin Publishing Co. 111-137.
- Fillmore, C. J. (1968): The case for case. In E. Bach and R. Harms (eds.): *Universals in linguistic theory*. New York: Holt, Rinehart & Winston. 1-88.
- Ganter, B. & R. Wille (1999): *Formal Concept Analysis. Mathematical Foundations*. Berlin: Springer.
- Guarino, N. (1992): Concepts, attributes and arbitrary relations: some linguistic and ontological criteria for structuring knowledge bases. *Data Knowl. Eng.*, 8(3). 249–261.
- Helbig, H. (2006): *Knowledge Representation and the Semantics of Natural Language*. Berlin: Springer.
- Katz, J. (1972): *Semantic Theory*. New York: Harper & Row.
- Kay, M. (1979): Functional grammar. In *Proceedings of the 5th Meeting of the Berkeley Linguistics Society*. 142-158.
- Kilbury, J. & W. Petersen & C. Rumpf (2006): Inheritance-based models of the lexicon. In D. Wunderlich (ed.): *Advances in the Theory of the Lexicon*. Berlin: Mouton de Gruyter. 429-477.

- Löbner, S. (1985): Definites. *Journal of Semantics*, 4(4). 279–326.
- Lyons, J. (1977): *Semantics* (2 volumes). New York: CUP.
- McClelland, J. L. & D. E. Rumelhart (1981): An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88. 375-407.
- Minsky, M. (1975): A framework for representing knowledge. In P. H. Winston (ed.): *The Psychology of Computer Vision*. New York: McGraw-Hill. 211-277.
- Osswald, R. & W. Petersen (2003): A Logical Approach to Data-Driven Classification. In A. Günter, R. Kruse, and B. Neumann (eds.): *KI-2003: Advances in Artificial Intelligence*. Lecture Notes in Artificial Intelligence 2821, Springer. 267-281.
- Petersen, W. (2006): FCAType – a System for Type Signature Induction. In S. B. Yahia and E. M. Nguifo (eds.): *Proceedings of 4th International Conference on Concept Lattices and their Applications (CLA'2006)*. Faculté des Sciences de Tunis, Université Centrale.
- Petersen, W. & M. Werning (2007): Conceptual Fingerprints: Lexical Decomposition by Means of Frames. In U. Priss, S. Polovina, and R. Hill (eds.): *Proceedings of the ICCS 2007*, LNAI 4604. Berlin: Springer. 415–428.
- Quillian, M. R. (1968): Semantic memory. In M. Minsky (ed.): *Semantic Information Processing*. Cambridge, MA: MIT Press. 227–270.
- Rosch, E. & C. B. Mervis (1975): Family resemblances: studies in the internal structure of categories. *Cognitive Psychology*, 7. 573-605.
- Rosch, E. (1973): Natural categories. *Cognitive Psychology*, 4. 328-350.
- Rosch, E. (1975): Cognitive reference points. *Cognitive Psychology*, 7. 532-547.
- Shanks, D. R. (1991): Categorization by a connectionist network. In *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17. 433-443.
- Shieber, S. M. (1986): *An Introduction to Unification-Based Approaches to Grammar*. Stanford, CA: CSLI Lecture Notes 4.
- Shieber, S. & H. Uszkoreit & F. Pereira & J. Robinson & M. Tyson (1983): The formalism and implementation of PATR-II. In B. J. Grosz and M. Stickel (eds.): *Research on Interactive Acquisition and Use of Knowledge*. Menlo Park, CA: SRI International. 39-79.
- Smith, E. E. & D. L. Medin (1981): *Categories and Concepts*. Cambridge, MA: Harvard University Press.
- Sowa, J. (1984): *Conceptual structures*. Reading, MA: Addison-Wesley.
- Sowa, J. (2000): *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove: Brooks & Cole.
- Woods, W. A. (1975): What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. M. Collins (eds.): *Representation and understanding: Studies in cognitive science*. New York: Academic Press. 35-82.