

The Logic of Categorization

Pei Wang

Department of Computer and Information Sciences, Temple University

Philadelphia, PA 19122

<http://www.cis.temple.edu/~pwang/>

pwang@joda.temple.edu

Abstract

The AI system NARS contains a categorization model, in the sense that categorization and reasoning are two aspects of the same mechanism. As a theory of categorization, the NARS model unifies several existing theories. In this paper, the logic used in NARS is briefly described, and the categorization model is compared with the others.

Introduction

NARS (Non-Axiomatic Reasoning System) is an AI system based on the theory that intelligence means adaptation with insufficient knowledge and resources (Wan94; Wan95; Wan01).

Though NARS is usually presented as a reasoning system, it can also be seen as a computational model of categorization. In NARS, knowledge is represented in the framework of a *categorical logic*, so that all relations are variations of categorical relations, such as *Inheritance* and *Similarity*. With insufficient knowledge, the truth value of a statement is not binary.

According to the *experience-grounded semantics* of NARS, the meaning of a term (concept) is determined by its existing (categorical) relations with other terms. In this way, in general the meaning of a term depends both on its extension (instances, exemplars) and intension (properties, features), though the dependency on the two aspects is not necessarily equal. Also, limited by available resources, when each time a term is used, only part of its meaning is involved, selected according to a priority distribution.

Given this semantics, the reasoning activity of the system (deduction, induction, abduction, revision, comparison, analogy, ...) constantly changes the meaning of terms, according to the experience of the system.

Non-Axiomatic Reasoning System (NARS)

Under the 5-page length limitation, this paper only gives a brief introduction to the most relevant aspects of NARS. For publications and an on-line demo of NARS, see <http://www.cogsci.indiana.edu/farg/peiwang/papers.html>.

Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

Knowledge representation

NARS does not use First-Order Predicate Logic or its variations. Instead, it uses a kind of *categorical logic* (also called *term logic*), in which each statement is about a categorical relation between a subject term and a predicate term, where a *term* is the name of a concept. In NARS, the most fundamental categorical relation is the *Inheritance* relation, written as “ \subset ”. In the ideal form, it is a reflexive and transitive binary relation defined among terms. Intuitively, a statement $S \subset P$ says that S is a *specialization* of P , and P is a *generalization* of S . This roughly corresponds to “ S is a kind of P ” in English. For example, “Bird is a kind of animal” can be represented as $bird \subset animal$. In this way, Inheritance defined in NARS is like the “sub-category vs. super-category” relation. Given $S \subset P$, we often say that the *instances* associated with S have the *properties* associated with P .

The Inheritance relation is treated as basic, because all other relations can be converted into Inheritance relations with the help of compound terms. For example, an arbitrary relation R among three terms A, B, C and C is usually written as $R(A, B, C)$, which can be equivalently rewritten as one of the following Inheritance statements (i.e., they have the same meaning and truth value):

- $(A, B, C) \subset R$, where the subject term is a compound (A, B, C) , an ordered tuple. This statement says “The relation among A, B, C (in that order) is a special case of the relation R .”
- $A \subset R(*, B, C)$, where the predicate term is a compound $R(*, B, C)$ with a “wildcard”, $*$. This statement says “ A is such an x that satisfies $R(x, B, C)$.”
- $B \subset R(A, *, C)$. Similarly, “ B is such an x that satisfies $R(A, x, C)$.”
- $C \subset R(A, B, *)$. Again, “ C is such an x that satisfies $R(A, B, x)$.”

This treatment of relation is similar to the case in set theory, where a relation is defined as a set of tuples.

Since in NARS all knowledge is represented as Inheritance relations, we can think the knowledge base of NARS as an “Inheritance network”, where each node represents a term, and each link represents an Inheritance statement from one term to another. It means that the memory of NARS consists of a set of concepts, each of which is named by a term.

The body of a concept consists of a set of statements (with truth values attached), in which the term is the subject or the predicate. These statements are the knowledge the system has on the concept. Inference tasks are also stored within the concepts whose names appear in the tasks. In the current version of NARS, there are two types of tasks: questions to be answered and pieces of new knowledge to be digested.

Semantics

In NARS, the *extension* and *intension* of a term T are defined as sets of terms that have direct Inheritance relations with T :

$$T^E = \{x \mid x \subset T\}; \quad T^I = \{x \mid T \subset x\}$$

Intuitively, they include all known specialization (instances) and generalizations (properties) of T , respectively. Under this definition, “extension” and “intension” is a dual relation among terms. That is, T_1 is in the extension of T_2 , if and only if T_2 is in the intension of T_1 .

From the reflexivity and transitivity of Inheritance, it can be proven that

$$(S \subset P) \iff (S^E \subseteq P^E) \iff (P^I \subseteq S^I)$$

where the first statement is about an Inheritance relation between two terms, while the last two are about subset relations between two sets (extensions and intensions of terms).

The above theorem identifies $S \subset P$ with “ P inherits the extension of S , and S inherits the intension of P ”. Since an Inheritance statement is a summary of multiple other Inheritance statements, we can use the latter as *evidence* for the former. We need the concept “evidence” to define the truth value of “imperfect” Inheritance statements, which is uncertain because the system has insufficient knowledge and resources. These uncertain statements are the ones actually appear in NARS, while the idealized (binary) Inheritance relation is just a theoretical notion used to build the semantics of NARS.

According to the above theorem, for a statement $S \subset P$ and a term M , we have

- if M is in the extensions of both S and P , it is positive evidence for the statement (because as far as M is concerned, P indeed inherits the extension of S);
- if M is in the extensions of S but not the extension of P , it is negative evidence (because as far as M is concerned, P fails to inherit the extension of S);
- if M is in the intensions of both P and S , it is positive evidence for the statement (because as far as M is concerned, S indeed inherits the intension of P);
- if M is in the intension of P but not the intension of S , it is negative evidence (because as far as M is concerned, S fails to inherit the intension of P);
- otherwise, M is not directly relevant to the statement.

Therefore, when the experience of the system is given as a set of statements in which each statement is either an (ideal) Inheritance or its negation, then for $S \subset P$, the amount of positive evidence is

$$w^+ = |S^E \cap P^E| + |P^I \cap S^I|$$

the amount of negative evidence is

$$w^- = |S^E - P^E| + |P^I - S^I|$$

and the amount of all evidence is

$$w = w^+ + w^- = |S^E| + |P^I|$$

Another categorical relation, *Similarity*, is defined as symmetric Inheritance, so that $S = P$ is a summary of $S \subset P$ and $P \subset S$. The (positive and negative) evidence of either Inheritance relation is also the evidence for the Similarity relation. Therefore, for $S = P$ we have

$$w^+ = |S^E \cap P^E| + |P^I \cap S^I|$$

$$w^- = |S^E - P^E| + |P^E - S^E| + |P^I - S^I| + |S^I - P^I|$$

$$w = w^+ + w^- = |S^E \cup P^E| + |P^I \cup S^I|$$

The *truth value* of a statement in NARS is a pair of numbers in $[0, 1]$, $\langle f, c \rangle$. f is the *frequency* of the statement, defined as

$$f = w^+ / w$$

so it indicates the proportion of positive evidence among all evidence. c is the *confidence* of the statement, defined as

$$c = w / (w + 1)$$

so it indicates the proportion of current evidence among evidence in the near future (after a unit-weight evidence is collected). When $f = 1$, it means that all known evidence is positive; when $f = 0$, it means that all known evidence is negative; when $c = 0$, it means that the system has no evidence on the statement at all (and f is undefined); when $c = 1$, it means that the system already has all the evidence on the statement, so that it will not be influenced by future experience. Therefore, “absolute truth” has a truth value $\langle 1, 1 \rangle$, and in NARS $S \subset P \langle 1, 1 \rangle$ can be written as $S \subset P$, as we did earlier in the discussion. Under the “insufficient knowledge” assumption, such a truth value cannot be reached by empirical knowledge, though it can be used for analytical knowledge (such as theorems in mathematics), as well as serve as idealized situation in semantic discussions. For more discussion on evidence and truth value in NARS, see (Wan95).

In NARS, the *meaning* of a term consists of its extension and intension, that is, its categorical relations with other terms in the system, including both its instances and its properties. In the memory structure mentioned previously, the meaning of a term is exactly the concept body associated with the term.

The above semantics of NARS is called “experience-grounded”, because both “truth” and “meaning” are defined as functions of the experience of the system. This semantics is fundamentally different from the traditional model-theoretic semantics used by most AI reasoning systems, where truth and meaning are defined by an “interpretation”, a mapping between items in the system and items in a model.

Inference

The truth value is *defined* in terms of amount of evidence collected in idealized situation (so the evidence itself is certain), but it is not actually obtained in that way for the statements in NARS. In real situation, the input knowledge comes into the system with truth value assigned by the user or other knowledge sources (according to the above definition), and derived knowledge is produced recursively by the built-in inference rules, which have truth value functions that determine the truth values of the conclusions according to those of the premises. The truth-value functions are designed according to the above semantics. In NARS, an inference rule is valid as long as its conclusion is based on the evidence provided by the premises. This, once again, is different from the definition of validity of inference rules in model-theoretic semantics.

Typical inference rules in categorical logic take a *syllogistic* form, that is, given a pair of statements, if they share a common term, a conclusion between the other two (not shared) terms is derived. In this aspect, NARS is more similar to Aristotle's Syllogism than to First-Order Predicate Logic. Different combinations of premises correspond to different inference rules, and use different truth-value functions. NARS has inference rules for *deduction, induction, abduction, revision, comparison, analogy, compound term formation*, and so on. An introduction of the rules is beyond the scope of this paper. For our current discussion, it is enough to know that a rule can generate new statements, as well as can revise the truth value of existing statements.

After initialization, NARS runs by repeating the following steps in a cycle:

1. *select* a concept to work on;
2. within the selected concept, *select* an inference task, with can either be a question to be answered, or a piece of new knowledge to be digested;
3. within the selected concept, according to the selected task, *select* a piece of (previous) knowledge;
4. *derive* new tasks and knowledge from the selected task and knowledge by the applicable rules;
5. *return* the selected concept, task, and knowledge back into the memory;
6. *insert* the new tasks and knowledge into the memory.

In the above step 1, 2, and 3, the selection of items (concept, task, and knowledge) is done according to the priority distributions among the items involved. The probability for an item to be selected is proportional to its priority value. In step 4, the priority of the new items are determined by the inference rule. In step 5, the priority values of the selected items are adjusted before returned into the memory.

NARS maintains priority distributions among components of its memory (concepts, tasks, and knowledge) because of its assumption of insufficient resources. Under that assumption, the system cannot afford the time to use all available knowledge to process all existing tasks, nor can it afford the space to keep all derived knowledge and tasks. Therefore, it has to allocate its resources among the items.

The allocation is uneven, and reflects the relative importance of the items for the system. (Wan96b)

In all the three cases (for concept, task, and knowledge), the priority of an item depends on a long-term factor and a short-term factor.

The long-term factor is initially determined by the "quality" of the item. It depends on, in a concept, how rich its content is; in a task, how urgent it is to user; in a piece of knowledge, how confident it is, and so on. Then after each time an item is processed, this factor is adjusted according to its performance in this step. Therefore, in the long run, useful concepts and knowledge become more accessible. Since the system's memory has a constant size, and new items are input and derived all the time, items with low priority value are removed from the system when the memory is full.

The short-term factor mainly reflects the relevance of an item to the current context. The inference activity of the system "activates" the items that are directly related to the tasks under processing. On the other hand, there is a "decay" process going on, so that if an items has not been involved with the inference activity for a while, its priority value is decreased.

The above resource allocation mechanism makes NARS to show "attention" and "forgetting" phenomena. Also, the meaning of a concept is not a simple set of knowledge and task, but a set with a priority values defined among the components, so that some components contribute more than the others to what a concept means to the system at a given time.

Categorization in NARS

In this section we discuss the major aspects of the categorization model proposed by NARS, and compare it with the other theories.

Extension and intension

To define the meaning of a concept as its relation with other concepts is not really a new idea. For example, this is also the case in semantic networks (Qui68). What makes NARS special is that, formalized as a categorical logic, all empirical statements become instances of the Inheritance relation defined above, therefore become *categorical* statements — $S \subset P$ says that S is a sub-concept of P , and P is a super-concept of S . As a result, the meaning of a concept in NARS can be simply defined by the extension and intension of the concept, and extension/intension is defined by Inheritance relation.

According to *Prototype Theory*, a concept is characterized by *properties* shared by most of its members (Ros73). According to *Exemplar Theory*, a concept is determined by a set of *instances* (Nos91). Using the terminology of NARS, Exemplar Theory defines the meaning of a concept as its extension, and Prototype Theory (as well as "feature list" theory) defines the meaning of a concept as its intension.

In this way, the problem is like the "chicken-and-egg" question — do we get instances first, then generalize properties from them, or get properties first, then determine instances according to them? The answer provided by NARS is: both. Whenever the system gets a piece of new knowledge, since its form is $S \subset P$ (with a certain truth value),

it always adds something new to the extension of P , and to the intension of S . Therefore, to the system as a whole, extension and intension are symmetric, and are developed together — they are just two opposite directions of a link.

On the other hand, for a concrete concept, it is quite possible that its meaning is mainly determined by its extension, while in another concept, by its intension. It depends on how the concept is learned and used in the past. In general, both extension and intension contributes to the meaning of a concept, though not necessarily to the same extent.

Since meaning of a concept is determined by its relations with other concepts, it depends on the role the concept plays in the conceptual structure of the system. Consequently, NARS is also consistent with the “Theory Theory” of categorization (MM85). However, concretely speaking, in NARS the “role in a theory” is indicated by nothing but instances and properties of the concept. In this sense, NARS provides a more detailed model for how a concept is related to a conceptual structure.

When the truth value of a statement is determined in NARS, both extensional evidence and intensional evidence are taken into account, as implied by the previous definition of evidence. In this way, Inheritance is different from similar relations like “subset” in set theory (which is defined by extension only) and “inheritance” in object-oriented programming (which is defined by intension only). As discussed in (Wan95), in a closed world where the system has sufficient knowledge and resources, the extension and intension of a concept uniquely determine each other, so that to consider one of the two is enough. However, for environments where the system has insufficient knowledge and resources, the two are no longer perfectly synchronized, so it is necessary to consider both. When NARS is asked to decide whether an entity e belongs to a concept C , the system may do so by comparing e with known instances of C , or by checking whether e has the properties associated with C , or both — it depends on what knowledge is available to the system, and which part of it is active at the time.

In summary, one feature that distinguishes NARS from the other models of categorization is: since NARS has multiple types of inference rules, the same categorical relation can be built in different ways. Therefore, NARS provides a possibility to unify the various models of categorization, not by simply putting them together, but by building a unified mechanism, which under different conditions show different features.

As mentioned above, in NARS concepts have priority values attached, and the long-term factor in the priority value mainly depends on the usefulness of the concept. Usually, a concept with unbalanced extension/intension is not very useful — a concept with a big extension and a small intension is like a collection of objects with few common properties, and a concept with a big intension and a small extension is like a collection of features which few object can satisfy. A concept with balanced extension/intension tend to be close to the “basic level” of categorization (Ros78), which, according to typical human experience, include many instances with many common properties, therefore are usually more useful for various future situations. In this way, NARS

provides an explanation to the basic-level phenomenon, and also uses it for categorization and inference control.

Fluid concept

Concepts in NARS approach the idea that Hofstadter calls “fluid concept” — that is, “concepts with flexible boundaries, concepts whose behavior adapts to unanticipated circumstances, concepts that will bend and stretch — but not without a limit” (HtFARG95).

Though many people agree that “invariant representations of categories do not exist in human cognitive systems” (Bar87), the existing models of categorization do not provide a clear picture about the *dynamics* of concepts.

In NARS, Inheritance is always a matter of degree, determined by the system’s experience about the terms in the relation. In this aspect, it agrees with fuzzy logic (Zad65) and several psychological models of categorization (MR92). The difference between NARS and fuzzy logic is in the interpretation of fuzziness, as well as how it changes over time. Since the first issue has been addressed by a previous publication (Wan96a), here we will focus on the second one.

As described above, in NARS, the meaning of concepts changes as the system get new information, as well as a result of the internal inference activity. At a given time, the system only uses part of the available knowledge, according to the priority distribution.

A concept changes its meaning both in long-term and in short-term.

In the long-term, the process corresponds to concept learning and evolving. Instead of assuming the learning process converges to a “correct representation” or “true meaning” of the concept, in NARS the process is a never-ended adaptation, and the result is determined by the experience of the system. In NARS, “experience” not only includes “external experience” (i.e., what information the system has got from the environment), but also includes “internal experience” (i.e., what the system has done on the information). According to the previous description about the inference process and the definition of meaning in NARS, we see that the inference activity changes the meaning of the concepts involved, simply by adding new links, removing old links, and revising the truth value and priority value of existing links.

In the short-term, the system shows context-sensitivity. Under the assumption of insufficient resources, NARS almost never tries to use all available knowledge to process a task. Instead, only “partial meaning” will be used, and which part actually get selected depends on the priority distribution. Roughly speaking, the selected ones tend to be “useful” and “relevant”, judged by the system according to past experience and current context, and there is also a random factor in the selection mechanism. As a result, the system may use the same concept with different meaning in different situations, by focusing on different part of the available knowledge on the concept.

Such a categorization model can explain many psychological and linguistic phenomena. For example, a concept is used as a metaphor when only a small part of its meaning

is used, and the other part is deliberately ignored, due to a certain context.

Of course, “fluid concept” does not mean that every arbitrary usage of a concept is equally possible. When the environment is stable, the system’s experience on certain concepts may become stable, too, and consequently those concepts may develop their “hard cores” — a small set of relations that can efficiently process most tasks related to those concepts. Such a hard core is close to what we usually call the “essence”, or the “definition”, of the concept. Since in NARS nothing is ever absolutely stable, the above words are only used relatively.

Categorization and reasoning

NARS is an attempt to provide a unified normative theory for intelligence, both in human and in computer (Wan95). As a reasoning system, it unifies various types of inference (Wan01). Here we see that it also unifies categorization and reasoning.

As described above, in NARS all categorization related information processing is carried out by the inference rules, therefore categorization is reasoning. On the other hand, since NARS is a kind of categorical logic, in each inference step the premises and conclusions are all categorical statements (i.e., the Inheritance relation defined previously), therefore reasoning is categorization. The two terms just focus on different aspects of the same process.

In traditional reasoning (and computing in general) systems, model-theoretic semantics is used, so that the syntax and semantics of a language is separated. The formalized inference engine knows syntax only, and has nothing to do with the meaning of the concepts. Searle’s “Chinese Room” argument is actually about this distinction (Sea80). In NARS, with experience-grounded semantics, inference is based on both syntax and semantics. For example, the deduction rule derives conclusion $A \subset C$ from premises $A \subset B$ and $B \subset C$ (truth-value function omitted), which is a syntactic rule, because the inference engine only needs to recognize the fact that the predicate term of the first premises is also the subject term of the second premise, and to generate the conclusion accordingly. However, the same rule is also a semantic rule, because the premises reveal partial meaning of the terms involved, the conclusion add new meaning to the terms in it, and the inference rule, with its truth-value function, is justified according to the semantic analysis of this type of inference.

Beside the semantics used, NARS is also different from traditional reasoning system in its formal language. As we have seen, NARS uses a categorical logic, where all relations are converted into categorical relations. On the contrary, in predicate logic, categorical relations and non-categorical relations are separated, and usually need to be handled differently. Such examples can be found in many knowledge-based systems, where “terminological knowledge” (categorical relations) and “factual knowledge” (non-categorical relations) are represented and processed differently (BS85). As a result, in such systems categorization and reasoning do not have the same relationship as in NARS.

Though NARS is not finished yet, and we cannot discuss all of its categorization-related aspects in this paper, we can see that it provides a promising theory of categorization, and may unify various aspects of cognition into a single model.

References

- L. Barsalou. The instability of graded structure: implications for the nature of concepts. In U. Neisser, editor, *Concepts and Conceptual Development: Ecological and intellectual factors in categorization*, chapter 5, pages 101–140. Cambridge University Press, Cambridge, 1987.
- R. Brachman and J. Schmolze. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9:171–216, 1985.
- D. Hofstadter and the Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New Nork, 1995.
- G. Murphy and D. Medin. The role of theories in conceptual coherence. *Psychological Review*, 92(3):289–316, 1985.
- D. Medin and B. Ross. *Cognitive Psychology*. Harcourt Brace Jovanovich, Fort Worth, 1992.
- R. Nosofsky. Typicality in logically defined categories: exemplar-similarity versus rule instantiation. *Memory and Cognition*, 17:444–458, 1991.
- M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*. The MIT Press, Cambridge, Massachusetts, 1968.
- E. Rosch. On the internal structure of perceptual and semantic categories. In T. Moore, editor, *Cognitive Development and the Acquisition of Language*, pages 111–144. Academic Press, New York, 1973.
- E. Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, New jersey, 1978.
- J. Searle. Minds, brains, and programs. *The Behavioral and Brain Sciences*, 3:417–424, 1980.
- P. Wang. From inheritance relation to nonaxiomatic logic. *International Journal of Approximate Reasoning*, 11(4):281–319, November 1994.
- P. Wang. *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence*. PhD thesis, Indiana University, 1995.
- P. Wang. The interpretation of fuzziness. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(4), 1996.
- P. Wang. Problem-solving under insufficient resources. In *Working Notes of the AAI Fall Symposium on Flexible Computation*, Cambridge, Massachusetts, November 1996.
- P. Wang. Abduction in non-axiomatic logic. In *Working Notes of the IJCAI workshop on Abductive Reasoning*, pages 56–63, Seattle, Washington, August 2001.
- L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.