Return to Term Logic

Pei Wang
Center for Research on Concepts and Cognition
Indiana University
http://www.cogsci.indiana.edu/farg/pwang.html
pwang@cogsci.indiana.edu

Abstract

Term logic is characterized by subject–predicate statements and syllogistic inference rules. This kind of logic, when properly extended, provides a natural and consistent model for multiple types of inference, including deduction, abduction, induction and revision. This paper briefly describes how such a logic works in the NARS project.

1 Introduction

There are two major traditions in formal logic: term logics and propositional/predicate logics, exemplified respectively by the Syllogism of Aristotle and the First-Order Predicate Logic of Frege, Russell, and Whitehead.

Term logic is different from predicate logic in both its knowledge representation language and its inference rules.

Term logic uses subject–predicate statements, in each of which two terms are linked together by a copula:

$$S \subset P$$

where S is the subject term of the statement, and P is the predicate term. Intuitively, this statement says that S is a specialization (instantiation) of P, and P is a generalization (abstraction) of S. This roughly corresponds to "S is P" in English.

Term logic uses syllogistic inference rules, in each of which two statements sharing a common term generate a new statement linking the two unshared terms.

When Aristotle introduced the deduction/induction distinction, he presented it in term logic, and so did Peirce when he added abduction into the picture [Peirce, 1931]. According to Peirce, the deduction/induction/abduction triad is defined formally in terms of the position of the shared term:

DEDUCTION

$$\frac{\begin{array}{c} M \subset P \\ S \subset M \end{array}}{S \subset P}$$

INDUCTION

$$\frac{\begin{array}{c} M \subset P \\ M \subset S \end{array}}{S \subset P}$$

ABDUCTION

$$\frac{\begin{array}{c} P \subset M \\ S \subset M \end{array}}{S \subset P}$$

Defined in this way, the difference among the three is purely syntactic: in deduction, the shared term is the subject of one premise and the predicate of the other; in induction, the shared term is the subject of both premises; in abduction, the shared term is the predicate of both premises. If we only consider combinations of premises with one shared term, these three exhaust all the possibilities.

It is well-known that only deduction can generate sure conclusions, while the other two are fallible. When discussing the pragmatic usage of non-deductive inference, Peirce proposed that induction is used for generalization, and abduction is used for explanation. Currently in AI, however, the situation is just the reverse of what Peirce did: people take "generalization" and "explanation" as the definition of induction and abduction, respectively, and then look for corresponding syntactic rules for them, usually in the framework of First-Order Predicate Logic and its variations. This happens because, with the rising of mathematical logic, predicate logic has replaced term logic as the dominant paradigm of formal logic.

The main claim of this position paper is: though predicate logic has done a good job in modeling binary deduction, term logic provides a better framework when induction and abduction are the target.

2 Induction and Abduction in NARS

NARS is an intelligent reasoning system I developed. In this position paper, I only introduce the part of it that is directly relevant to the "Workshop on Abduction and Induction in AI" of IJCAI'97. For detailed descriptions of the system, see [Wang, 1994; Wang, 1995], which are also available via the Web at *http://www.cogsci.indiana.edu/farg/peiwang/papers.html.*

2.1 Syntax and Semantics

NARS is designed to be adaptive, and to work with insufficient knowledge and resources. The system answers questions in real time according to available knowledge, which may be incomplete (with respect to the questions to be answered), uncertain, and inconsistent.

In NARS, each statement has the form

$$S \subset P \ {<} F,\ C {>}$$

where "$S \subset P$" is defined as before, and "$< F,\ C >$" is a pair of real numbers, referred to as the *frequency* and the *confidence* of the statement, respectively. Both "F" and "C" are real numbers in [0, 1], and they altogether represent the truth value of the statement.

When both F and C reach their maximum value, 1, the statement indicates a *complete inheritance* relation from S to P. We write this special case as "$S \sqsubset P$". By definition, this relation is reflexive and transitive.

We further define the *extension* and *intension* of a term T as sets of terms:

$$E_T = \{x | x \sqsubset T\} \text{ and } I_T = \{x | T \sqsubset x\}$$

respectively. It can be proven that

$$S \sqsubset P \text{ iff } E_S \subseteq E_P \text{ and } I_P \subseteq I_S$$

This is why "$\sqsubset$" is called a "complete inheritance" relation — "$S \sqsubset P$" means that P completely inherits the extension of S, and S completely inherits the intension of P.

As mentioned before, "$\sqsubset$" is a special case of "$\subset$". In NARS, according to the assumption of insufficient knowledge, inheritance relations are usually incomplete. Though complete inheritance relations do not appear as given knowledge to the system, we can use them to define positive and negative evidence of a statement in idealized situations, just like we usually *define* measurements of physical quantities in highly idealized situations, then *use* them in actual situations according to the definition.

If the given knowledge to the system is a set of complete inheritance relations, then the *weight* of positive evidence and total (positive and negative) evidence are defined, respectively, as

$$W^+ = |E_S \cap E_P| + |I_P \cap I_S|,\ W = |E_S| + |I_P|$$

Finally, the truth value mentioned previously can be defined as

$$F = W^+/W,\ C = W/(W+1)$$

Intuitively, F is the proportion of positive evidence among all evidence, and C is the proportion of current evidence among evidence in the near future (after a unit-weight evidence comes). When C is 0, it means that the system has no evidence on the proposed inheritance relation at all (and F is undefined); the more evidence the system gets (no matter positive or negative), the more confident the system is on this judgment.

2.2 Inference Rules

Though the truth value of a statement is *defined* in terms of counting in extension and intension of the two terms, it is not actually *obtained* in this way in NARS. Instead, inference rules have truth-value functions attached to calculate the truth values of the conclusions from those of the premises.

In NARS, the triad of inference becomes the following:

DEDUCTION

$$\begin{array}{l} M \subset P \ {<}F_1,\ C_1{>} \\ S \subset M \ {<}F_2,\ C_2{>} \\ \hline S \subset P \ {<}F_d,\ C_d{>} \end{array}$$

$$\begin{array}{l} F_d = F_1F_2/(F_1 + F_2 - F_1F_2) \\ C_d = C_1C_2(F_1 + F_2 - F_1F_2) \end{array}$$

INDUCTION

$$\begin{array}{l} M \subset P \ {<}F_1,\ C_1{>} \\ M \subset S \ {<}F_2,\ C_2{>} \\ \hline S \subset P \ {<}F_i,\ C_i{>} \end{array}$$

$$\begin{array}{l} F_i = F_1 \\ C_i = F_2C_1C_2/(F_2C_1C_2 + 1) \end{array}$$

ABDUCTION

$$\begin{array}{l} P \subset M \ {<}F_1,\ C_1{>} \\ S \subset M \ {<}F_2,\ C_2{>} \\ \hline S \subset P \ {<}F_a,\ C_a{>} \end{array}$$

$$\begin{array}{l} F_a = F_2 \\ C_a = F_1C_1C_2/(F_1C_1C_2 + 1) \end{array}$$

The above functions are determined in two steps: first, from the definition of the truth value in NARS, we get the boundary conditions of the functions, that is, the function values when the premises are complete inheritance relations. Second, these boundary conditions are extended into general situations according to some principles, such as the continuity of the function, the independence of the premises, and so on. For detailed discussion, see [Wang, 1994; Wang, 1995].

Intuitively, deduction extends the transitivity of complete inheritance relation into (incomplete) inheritance relation in general; induction establishes inheritance relations based on shared extension; abduction establishes inheritance relations based on shared intension. Inductive and abductive conclusions are always uncertain (i.e., their confidence values cannot reach 1 even if the premises have confidence 1).

Each time one of the above rules is applied, the truth value of the conclusion is evaluated solely according to the evidence summarized in the premises. To get more confident conclusions, the following revision rule is used to combine evidence from different sources:

REVISION

$$\frac{S \subset P \ <F_1,\, C_1> \quad S \subset P \ <F_2,\, C_2>}{S \subset P \ <F_r,\, C_r>}$$

$$F_r = \frac{F_1C_1(1-C_2)+F_2C_2(1-C_1)}{C_1(1-C_2)+C_2(1-C_1)}$$
$$C_r = \frac{C_1(1-C_2)+C_2(1-C_1)}{C_1(1-C_2)+C_2(1-C_1)+(1-C_1)(1-C_2)}$$

These two functions are derived from the relation between weight of evidence and truth value, and the additivity of weight of evidence during revision [Wang, 1994; Wang, 1995].

2.3 Inference Control

Since all statements have the same format, different types of inference can easily be mixed. The premises used by the induction rule may be generated by the deduction (or abduction, and so on) rule, and the conclusions of the induction rule may be used as premises by the other rules. The revision rule may merge an inductive conclusion with a deductive (or abductive, and so on) conclusion.

NARS processes many inference tasks at the same time by time-sharing, and in each time-slice a task (a question or a piece of new knowledge) interacts with a piece of available knowledge. The task and knowledge are chosen probabilistically according to priority distributions reflecting the urgency of each task and the salience of each piece of knowledge. The priority distributions are adjusted after each step, according to the result the system obtained in that step. By doing so, the system tries to spend more resources on more valuable works. Consequently, in each inference step the system does not decide what rule to use first, then look for corresponding knowledge. Instead, it picks up two statements that share a common term, and decides what rule to apply according to the position of the shared term.

In general, a question-answering procedure in NARS consists of many inference steps. Each step carries out a certain type of inference, such as deduction, induction, abduction, and so on. These steps are linked together in run-time in a context-dependent manner, so the processing of a question or a piece of new knowledge does not follow a predetermined algorithm. If the same task appears at different time, the processing processes and results may be different, depending on the available knowledge, the order by which the pieces of knowledge are accessed, and the time-space supply to the task.

When the system run out of space, it will remove terms and statements with the lowest priority.

This control mechanism makes NARS possible to answer questions in real time, to handle unexpected tasks, and to use its limited resources efficiently.

3 An Example

Let us see a simple example. Assume that the following is the relevant knowledge NARS has at a certain moment:

$$robin \subset feathered\text{-}creature \ <1.00, 0.90> \tag{1}$$
("Robin has feather.")

$$bird \subset feathered\text{-}creature \ <1.00, 0.90> \tag{2}$$
("Bird has feather.")

$$swan \subset bird \ <1.00, 0.90> \tag{3}$$
("Swan is a kind of bird.")

$$swan \subset swimmer \ <1.00, 0.90> \tag{4}$$
("Swan can swim.")

$$gull \subset bird \ <1.00, 0.90> \tag{5}$$
("Gull is a kind of bird.")

$$gull \subset swimmer \ <1.00, 0.90> \tag{6}$$
("Gull can swim.")

$$crow \subset bird \ <1.00, 0.90> \tag{7}$$
("Crow is a kind of bird.")

$$crow \subset swimmer \ <0.00, 0.90> \tag{8}$$
("Crow cannot swim.")

Then the system is asked to evaluate the truth value of

$$robin \subset swimmer$$

which is like asking "Can robin swim?"

To make the discussion simple, let us assume a certain priority distribution, according to which the premises are chosen in the following order.

[Step 1] From (1) and (2), by abduction, the system gets:

$$robin \subset bird \ <1.00, 0.45> \tag{9}$$

"Having feather" gives the system evidence to believe that robin is bird.

[Step 2] From (3) and (4), by induction, the system gets:

$$bird \subset swimmer \ <1.00, 0.45> \tag{10}$$

Swan provides positive evidence for "Bird swims".

[Step 3] From (9) and (10), by deduction, the system gets:

$$robin \subset swimmer \ <1.00, 0.20> \tag{11}$$

As an answer to a user question, this result is reported to the user, while the system continues to work on it when resources are available. Here the system is answering "Yes" to "Can robin swim?", though it is far from confident about this answer, and it is going to look for more evidence.

[Step 4] From (5) and (6), by induction, the system gets:

$$bird \subset swimmer \ <1.00, 0.45> \tag{12}$$

Gull also provides positive evidence for "Bird swims".

[Step 5] (10) and (12) look identical, but since they came from different sources, they are not redundant and can be merged by the revision rule to get:

$$bird \subset swimmer \ <1.00, 0.62> \tag{13}$$

Evidence from different sources accumulate to support a more confident conclusion.

[Step 6] From (7) and (8), by induction, the system gets:

$$bird \subset swimmer \ <0.00, 0.45> \tag{14}$$

Crow provides negative evidence for "Bird swims".

[Step 7] From (13) and (14), by revision, the system gets:

$$bird \subset swimmer \ <0.67, 0.71> \tag{15}$$

A compromise is formed by considering both positive and negative evidence, and the positive evidence is stronger.

[Step 8] From (9) and (15), by deduction, the system gets:

$$robin \subset swimmer \ <0.67, 0.32> \tag{16}$$

Because this conclusion is a more confident answer to the user question than (13), it is reported, too. In this way, the system can change its mind after more knowledge and resources become available.

It needs to be mentioned that a typical run in NARS is much more complex than this. In the above description, we have omitted the conclusions that are irrelevant to the user question.

4 Discussion

By using a term logic, NARS gets the following properties that distinguish it from other AI systems doing induction and abduction:

- In the framework of term logic, different types of inference (deduction, abduction, induction, and so on) are defined syntactically, and their relationship is simple and elegant.
- Compared to predicate Logic, term logic has a language that is closer to natural language, and inference rules that are closer to everyday human reasoning.
- With the definition of extension and intension introduced in NARS, it becomes easy to define truth value as function of available evidence, to consistently represent uncertainty from various sources, and to design truth-value functions accordingly.
- Induction and abduction become *duals* in the sense that they are completely symmetric to each other. The difference is that induction collects evidence from the extensions of the terms in the conclusion, while abduction collects evidence from the intensions of the terms. Intuitively, they still correspond to generalization and explanation, respectively.
- With the help of the revision rule, induction and abduction at the problem-solving level becomes incremental and open-ended processes, and they do not follow predemetmined algorithms.
- The choice for inference rule is knowledge-driven and context-sensitive. Though in each inference step, different types of inference are well-defined and clearly distinguished, the processing of user tasks typically consists of different types of inference. Solutions the user gets are seldom purely inductive, abductive, or deductive.

Though First-Order Predicate Logic is still better for binary deductive reasoning, for the enterprise of AI, term logic may provide a better platform. However, it does not mean that we should simply go back to Aristotle. NARS has extended the traditional term logic in the following aspects:

1. from binary to multi-valued,
2. from monotonic to revisible,
3. from extensional to both extensional and intensional,
4. from deduction only to multiple types of inference,
5. from atomic terms to compound terms.

The last issue is beyond the scope of this paper, but it needs to be mentioned briefly. Term logic is often criticized for its poor expressibility. Obviously, many statements cannot be put into the "$S \subset P$" format where S and P are simple words. However, this problem can be solved by allowing compound terms. This is similar to the situation in natural language: most (if not all) declarative sentences can be parsed into a subject phrase and a predicate phrase, which can either be a word, or a structure consisting of multiple words. Though the new version of NARS containing compound terms is still under developing, it seems that the expressibility of the term-oriented language can be greatly enriched in this way.

Because term logic has been ignored by mainstream AI for a long time, it is still too early to conclude its power and limitation. However, according to available evidence, at least we can say that it shows many novel properties, and some, if not all, of previous criticisms on term logic can be avoided if we properly extend the logic.

Acknowledgement

I would like to thank the anonymous reviewers for their helpful comments.

References

[Peirce, 1931] Peirce, C. (1931). *Collected Papers of Charles Sanders Peirce*, volume 2. Harvard University Press, Cambridge, Massachusetts.

[Wang, 1994] Wang, P. (1994). From inheritance relation to nonaxiomatic logic. *International Journal of Approximate Reasoning*, 11(4):281–319.

[Wang, 1995] Wang, P. (1995). *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence.* PhD thesis, Indiana University.