Non-Axiomatic Reasoning System (Version 4.1)

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

Introduction

NARS (Non-Axiomatic Reasoning System) is an intelligent reasoning system. It answers questions according to the knowledge originally provided by its user. What makes it different from conventional reasoning systems is its ability to learn from its experience and to work with insufficient knowledge and resources.

The NARS 4.1 demo is a Java applet. It comes with help information and simple examples to show how the system does deduction, induction, abduction, analogy, belief revision, membership evaluation, relational inference, backward inference, new concept formation, and so on, in a unified manner.

The demo also allows its user to create new examples to test the system, as well as to see the internal structure and process when the system is running. The on-line help document contains links to relevant publications.

A previous version of the system, NARS 3.0, is described in detail in (Wang, 1995), which, and other related publications, are available at the author's web page.

The System

NARS is based on the conjecture that what we call "intelligence" can be built into a computer system by making it to adapt to its experience, that is, to answer questions according to available knowledge and resources.

Concretely, it means that the system should open to new knowledge and questions in real time, and answer questions according to its available knowledge when the knowledge and resources are insufficient to provide a perfect answer.

Knowledge Representation

NARS does not use first-order predicate logic. Instead, each piece of knowledge in NARS has the form "$SrP < f, c >$". Here S is the subject term, and P is the predicate term. In the simplest situation, both of them are words. r is an inheritance relation. In this demo, three types of inheritance relations can be used:

- "$S \subset P$" means that "S is a special type of P";
- "$S \in P$" means that "S is an instance of P";
- "$S = P$" means that "S and P are similar to each other".

"$< f, c >$" is the truth value of the sentence, where f is the "frequency", a real number in [0, 1], indicating the ratio of positive evidence among all evidence of the relation, and c is the "confidence", a real number in (0, 1), indicating the amount of evidence the system has on the relation.

Each question that can be asked to the system has the form SrP. A question looks just like a piece of knowledge, except that there is no truth value, and that S or P (but not both) can be a special symbol "?". A question without "?" is like a "yes/no" question — the system is asked to evaluate the truth value of the given relation. A question with "?" is like a "what" question — the system is asked to find a term that have more positive evidence and less negative evidence for the given relation.

Since the confidence of a piece of knowledge cannot reach 1.0, no answer is absolutely sure. Instead, the system needs to compare the available candidates to choose a "best answer", which may be overturned by new knowledge or further consideration.

Inference Rules

The following basic rules are involved in this demo. Each of them in NARS takes two pieces of existing knowledge as premises, and derive a piece of new knowledge as conclusion. The premises must share at least one common term.

Revision

$$\begin{array}{c} S \subset P < f_1, c_1 > \\ S \subset P < f_2, c_2 > \\ \hline S \subset P < f, c > \end{array}$$

Deduction

$$\begin{array}{c} S \subset M < f_1, c_1 > \\ M \subset P < f_2, c_2 > \\ \hline S \subset P < f, c > \end{array}$$

Abduction

$$\begin{array}{c} S \subset M < f_1, c_1 > \\ P \subset M < f_2, c_2 > \\ \hline S \subset P < f, c > \end{array}$$

Induction

$$\begin{array}{c} M \subset S < f_1, c_1 > \\ M \subset P < f_2, c_2 > \\ \hline S \subset P < f, c > \end{array}$$

Analogy

$$\begin{array}{c} S \subset M < f_1, c_1 > \\ M = P < f_2, c_2 > \\ \hline S \subset P < f, c > \end{array}$$

Comparison

$$\begin{array}{c} S \subset M < f_1, c_1 > \\ P \subset M < f_2, c_2 > \\ \hline S = P < f, c > \end{array}$$

Since by definition $S \in P$ is identical to $\{S\} \subset P$, rules on the "$\in$" relation can be derived from those on the "$\subset$" relation.

In each rule, there is a truth value function that calculate the strength and confidence of the conclusion ($< f, c >$) from those of the premises ($< f_1, c_1 >$ and $< f_2, c_2 >$). Different rule use different function.

According to how the confidence c is calculated, the above rules can be put into three categories:

1. In Deduction and Analogy, the confidence of the conclusion can be very close to the confidence of a premise, so these types of inference can produce relatively sure answers.
2. In Abduction, Induction, and Comparison, the confidence of the conclusion is always much lower than that of the premises, so these types of inference are more tentative.
3. Revision is the only rule where the confidence of the conclusion is higher than that of the promises, because this rule merges the evidence of the premises into that of the conclusion.

Besides these basic rules, NARS 4.1 also has compound-term composition and decomposition rules, such as "$S \subset (P1 \cap P2)$ if and only if $S \subset P1$ and $S \subset P2$". Another type of rule is backward inference rule that derive question from question and knowledge, such as from available knowledge "$S \subset M$" and question "$? \in M$" to derive a new question "$? \in S$", whose answer and the knowledge can derive an answer to the original question. This kind of rule allows the system to work in a goal-directed manner.

Control Mechanism

Because of the assumption of real time input, NARS cannot work on a task at a time, but must allow multiple tasks to be under processing at the same time. Because of the assumption of insufficient knowledge and resources, it cannot assume that all tasks will be processed to their "logical end", or to be solved by considering all relevant knowledge in the system.

Instead, the system processes multiple inference tasks by time-sharing. Each task is given a priority value, which indicates the frequency for it to be processed for a time slice. After a task is selected for processing, a piece of knowledge is also selected according to a priority distribution, then the derived task and knowledge are put back into the task pool and knowledge base, and the priority of the involved task and knowledge is adjusted according to the feedback obtained in this inference step.

When an answer is found for a user question, it is reported, then the system continue to look for a better one, if the task still have a high enough priority.

The Demonstration

NARS has been implemented several times. The current version, 4.1, is a Java applet which is available at the author's web page. There is also a file for download, which contains both the code and the documentation.

User Interface

The user interface of NARS 4.1 allows the user to provide knowledge and questions to the system in a text field. The system will return answers to the questions in another window. Since the timing of input influences the system's processing, the user can also specify the number of inference steps allowed between input events.

The user can let the system to work step by step, or to run continuously. The user can open several display windows to watch the internal inference process, as well as the content and priority distribution of the task pool and knowledge base.

There are several system parameters the user can adjust to change the system's behavior, such as the forgetting rate of the knowledge base, and so on.

There is an on-line User's Guide that explains how the demo can be used.

Examples

The NARS 4.1 demo has a set of examples attached, and each of which shows a basic function or property of the system. By observing how the examples are processed in the system, the user can get direct experience on how the system works.

The examples include: input and output, context sensitivity, deduction, induction, abduction, mixed inference, confidence processing, backward inference, contradiction handling, similarity evaluation, compound term formation, Hempel's paradox, relation operators, and fuzzy concept formation.

In the on-line documentation, each example comes with a simple explanation about the system's processing and the result, as well as links to related publications.

All of these examples can be given to the system by copy/paste. When a user becomes familiar enough to the system, he or she can create new examples to test the system, as long as they can be expressed in the formal language of NARS.

These examples show that NARS is different from other reasoning systems in terms of the knowledge representation language, the semantics of the language, the inference rules, the knowledge base structure, the control mechanism, and the relation with users. NARS provides a unified solution to many problems that are traditionally handled in isolation to one another.

References

Wang, P. 1995. Non-Axiomatic Reasoning System: Exploring the essence of intelligence. Ph.D. diss., Dept. of Computer Science and Program of Cognitive Science, Indiana Univ.