Unified Inference in Extended Syllogism

Pei Wang (pwang@cogsci.indiana.edu)
Center for Research on Concepts and Cognition, Indiana University

(Received: .....; Accepted: .....)

1. Term Logic vs. Predicate Logic

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

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

Term logic represents knowledge in subject–predicate statements. In the simplest form, such a statement contains *two terms*, linked together by an *inheritance relation*:

$$S \subset P$$

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 a kind of 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.

In Aristotle's syllogism (Aristotle, 1989), all statements are binary (that is, either true or false), and all valid inference rules are deduction, therefore when both premises are true, the conclusion is guaranteed to be true. 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/abduction/induction triad is defined formally in terms of the position of the shared term:

Deduction	**Abduction**	**Induction**
$M \subset P$ $S \subset M$ $S \subset P$	$P \subset M$ $S \subset M$ $S \subset P$	$M \subset P$ $M \subset S$ $S \subset P$

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 abduction, the shared term is the predicate of both premises; in induction, the shared term is the subject of both premises. If we only consider combinations of premises with one shared term, these three exhaust all the possibilities (the order of the premises does not matter for our current purpose).

It is well-known that only deduction can generate sure conclusions, while the other two are fallible. However, it seems that abduction and induction, expressed in this way, do happen in everyday thinking, though they do not serve as rules for proof or demonstration, as deduction does. In seeking for a semantic justification for them, Aristotle noticed that induction corresponds to generalization, and Peirce proposed that abduction is for explanation.

Later, when Peirce's focus turned to the pragmatic usage of non-deductive inference in scientific inquiry, he viewed abduction as the process of hypothesis generation, and induction as the process of hypothesis confirmation.

Peirce's two ways to classify inference, syllogistic and inferential (in the terminology introduced by Flach and Kakas, this volume), correspond to two levels of observation in the study of reasoning. The "syllogistic" view is at the micro level, and is about a single inference step. It indicates what conclusion can be derived from given premises. On the other hand, the "inferential" view is at the macro level, and is about a complete inference process. It specifies the logical relations between the initial premises and the final result, without saying how the result is obtained, step by step.

Though the syllogistic view is constructive and appealing intuitively, some issues remain open, as argued by Flach and Kakas (this volume):

1. Abduction and induction, when defined in the syllogistic form, have not obtained an appropriate semantic justification.

2. The expressive capacity of the traditional term-oriented language is much less powerful than the language used in predicate calculus.

I fully agree that these two issues are crucial to term logic.

For the first issue, it is well known that abduction and induction cannot be justified in the model-theoretic way as deduction. Since "presuming truth in all models" seems to be the only existing definition for the validity of inference rules, in what sense abductive and inductive conclusions are "better", or "more reasonable" than arbitrary conclusions? In what semantic aspect they differ from each other?

The second issue actually is a major reason for syllogism to lose its dominance to "mathematical logic" (mainly, first-order predicate logic) one hundred years ago. How much of our knowledge can be put into the form "S is a kind of P", where S and P are English words? Currently almost all logic textbooks treat syllogism as an ancient and primary form of logic, and as a subset of predicate logic in terms of functionalities. It is no surprise that even in the works that accept the "syllogistic" view of Peirce on abduction and induction, the actual formal language used is not that of syllogism, but predicate logic (see the work of Christiansen, this volume).

In the following, I want to show that when properly extended, syllogism (or term logic, I am using these two names interchangeably here) provides a more natural, elegant, and powerful framework, in which deduction, abduction, and induction can be unified in syntax, semantics, and pragmatics. Furthermore, this new logic can be implemented in a computer system for the purpose of artificial intelligence.

2. Extended Syllogism in NARS

NARS is an intelligent reasoning system. In this chapter, I only introduce the part of it that is directly relevant to the theme of this book. For more comprehensive descriptions of the system, see (Wang, 1994; Wang, 1995).

2.1. Syntax and Semantics

NARS is designed to be adaptive to its environment, 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 with internal conflicts.

In NARS, each statement has the form

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

where "$S \subset P$" is used as in the previous section, and "$< F,\ C >$" is a pair of real numbers in [0, 1], representing the truth value of the statement. F is the *frequency* of the statement, and C is the *confidence*.

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

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

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

respectively. It can be proven that

$$(S \sqsubseteq P) \Longleftrightarrow (E_S \subseteq E_P) \Longleftrightarrow (I_P \subseteq I_S)$$

where the first relation is an inheritance relation between two terms, while the last two are including relations between two sets.

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

As mentioned before, "$\sqsubseteq$" is a special case of "$\subset$". In NARS, according to the assumption of insufficient knowledge, inheritance relations are usually incomplete. To adapt to its environment, even incomplete inheritance relations are valuable to NARS, and the system needs to know how incomplete the relation is, according to given knowledge. 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.

For a given statement "$S \subset P$" and a term M, when M is in the extension of S, it can be used as evidence for the statement. If M is also in the extension of P, then the statement is true, as far as M is concerned, otherwise it is false, with respect to M. In the former case, M is a piece of positive evidence, and in the latter case, it is a piece of negative evidence. Similarly, if M is in the intension of P, it becomes evidence for the statement, and it is positive if M is also in the intension of S, but negative otherwise.

For example, if we know that iron (M) is metal (S), then iron can be used as evidence for the statement "Metal is crystal" ($S \subset P$). If iron (M) is crystal (P), it is positive evidence for the above statement (that is, as far as its instance iron is concerned, metal is crystal). If iron is not crystal, it is negative evidence for the above statement (that is, as far as its instance iron is concerned, metal is not crystal). Therefore, syntactically, "Metal is crystal" is an inductive conclusion defined

in term logic; semantically, the truth value of the conclusion is determined by checking for inherited instance (extension) from the subject to the predicate; pragmatically, the conclusion "Metal is crystal" is a generalization of "Iron is crystal", given "Iron is metal" as background knowledge.

Similarly, if we know that metal (P) is crystal (M), then crystal can be used as evidence for the statement "Iron is metal" ($S \subset P$). If iron (S) is crystal (M), it is positive evidence for the above statement (that is, as far as its property crystal is concerned, iron is metal). If iron is not crystal, it is negative evidence for the above statement (that is, as far as its property crystal is concerned, iron is not metal). Therefore, syntactically, "Iron is metal" is an abductive conclusion defined in term logic; semantically, the truth value of the conclusion is determined by checking for inherited property (intension) from the predicate to the subject; pragmatically, the conclusion "Iron is metal" is an explanation of "Iron is crystal", given "Metal is crystal" as background knowledge.

The perfect parallelism of the above two paragraphs indicates that induction and abduction, when defined in term logic as above, becomes a dual.

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

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

where set theory notations are used.

Finally, the truth value mentioned previously is 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 is collected). 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.

Now we can see that while in traditional binary logics, the truth value of a statement *qualitatively* indicates whether there exists negative evidence for the statement, in NARS the truth value *quantitatively* measures available positive and negative evidence.

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, the user specifies truth values of input statements (according to the semantics described in the previous section), and the inference rules in NARS have truth-value functions attached to calculate the truth values of the conclusions from those of the premises in each inference step.

In NARS, the triad of inference becomes the following:

Deduction	**Abduction**	**Induction**
$M \subset P \ <F_1,\ C_1>$ $S \subset M \ <F_2,\ C_2>$ ——————— $S \subset P \ <F,\ C>$	$P \subset M \ <F_1,\ C_1>$ $S \subset M \ <F_2,\ C_2>$ ——————— $S \subset P \ <F,\ C>$	$M \subset P \ <F_1,\ C_1>$ $M \subset S \ <F_2,\ C_2>$ ——————— $S \subset P \ <F,\ C>$
$F = \frac{F_1 F_2}{F_1+F_2-F_1F_2}$ $C = C_1 C_2 (F_1 + F_2 - F_1 F_2)$	$F = F_2$ $C = \frac{F_1 C_1 C_2}{F_1 C_1 C_2 + 1}$	$F = F_1$ $C = \frac{F_2 C_1 C_2}{F_2 C_1 C_2 + 1}$

The above truth-value 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 certain principles, such as the continuity of the function, the independence of the premises, and so on. For detailed discussions, see (Wang, 1994; Wang, 1995).

From a semantic point of view, the truth value of an inheritance relation is determined in different ways in different inference types: deduction extends the transitivity of complete inheritance relation into (incomplete) inheritance relation in general; abduction establishes inheritance relations based on shared intension; induction establishes inheritance relations based on shared extension. Though intuitively we can still say that deduction is for proving, abduction is for explanation, and induction is for generalization, these characteristics are no longer essential. Actually here labels like "generalization" and "explanation" are more about the *pragmatics* of the inference rules (that is, what the user can use them for) than about their *semantics* (that is, how the truth values of their conclusions are determined).

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. Abductive and inductive conclusions are always uncertain (i.e., their confidence values cannot reach 1 even if the premises have confidence 1), because they never check the extension or intension of the two terms *exhaustively* in one step, as deductive conclusions do.

To get more confident conclusions, the following revision rule is used to combine evidence from different sources:

Revision

$$\frac{\begin{array}{c} S \subset P \ <F_1,\ C_1> \\ S \subset P \ <F_2,\ C_2> \end{array}}{S \subset P \ <F,\ C>}$$

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

$$C = \frac{C_1(1-C_2)+C_2(1-C_1)}{C_1(1-C_2)+C_2(1-C_1)+(1-C_1)(1-C_2)}$$

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). The conclusion is more confident than either premise, because it is based on more evidence. The frequency of the conclusion is more stable in the sense that it is less sensitive (compared to either premise) to (a given amount of) future evidence.

Using this rule to combine abductive and inductive conclusions, the system can obtain more confident and stable generalizations and explanations.

2.3. Inference Control

Since all statements have the same format, different types of inference can easily be mixed in an inference procedure. 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 values 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 important and promising tasks, with more reliable and useful knowledge. 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, abduction, induction, revision, and so on. These steps are linked together in run-time in a context-sensitive 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 in different context, 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 resource supplied to the task.

When the system runs out of space, it removes terms and statements with the lowest priority, therefore some knowledge and tasks may be permanently forgotten by the system. When the system is busy (that is, working on many urgent tasks at the same time), it cannot afford the time to answer all questions and to consider all relevant knowledge, so some knowledge and tasks may be temporally forgot by the system. Therefore the quality of the answers the system can provide not only depends on the available knowledge, but also depends on the context in which the questions are asked.

This control mechanism makes it possible for NARS 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:

$$robin \subset feathered\text{-}creature \ <1.00, 0.90> \quad (1)$$

("Robin has feather.")

$$bird \subset feathered\text{-}creature \ <1.00, 0.90> \quad (2)$$

("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:

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

Here "having feather" gives the system evidence to believe that robin is a kind of bird, though the confidence of the conclusion is low, because it is only based on a single piece of evidence.

[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". Again, the confidence is low.

[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-asked 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:

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

Evidences 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 to the user, 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 the previous description, where we have omitted the conclusions that are irrelevant to the user question, and we have assumed an order of inference that directly leads to the desired result.

For example, in Step 2 and 4, NARS actually also gets a symmetric inductive conclusion

$$swimmer \subset bird \ \ <1.00, 0.45> \tag{17}$$

which can be combined to become

$$swimmer \subset bird \ <1.00, 0.62> \tag{18}$$

However, in Step 6 there is no symmetric inductive conclusion generated — since crow is not a swimmer, no matter it is a bird or not, it provides no evidence for $swimmer \subset bird$. From the definition of (positive and negative) evidence introduced earlier, it is not hard to see that in induction and abduction, positive evidence for "$X \subset Y$" are also positive evidence for "$Y \subset X$", but negative evidence for the former is not counted as evidence for the latter.

In practical situations, the system may wonder around, and jump from task to task. However, these behaviors are reasonable in the sense that all conclusions are based on available evidence, and the choice of task and knowledge at each step is determined by a priority distribution, which is formed by the system's experience and current environmental factors (such as user requirements).

4. Discussion

In this book, the current chapter is the only one that belongs to the term logic tradition, while all the others belong to the predicate logic tradition. Instead of comparing NARS with the other approaches introduced in the volume one by one, I will compare the two paradigms and show their difference in handling abduction and induction, because this is the origin of many minor differences between NARS and the other works.

Compared with deduction, a special property of abduction and induction is the uncertainty they introduced into their conclusions, that is, even when all the premises are completely true and an abduction (or induction) rule is correctly applied, there is no guaranty that the conclusion is completely true.

When abduction and induction are formalized in binary logic, as in the most chapters of this book, their conclusions become defeasible, that is, a conclusion can be falsified by any single counter-evidence (see Lachiche, this volume). The philosophical foundation and implication of this treatment of induction can be found in Popper's work (Popper, 1959). According to this approach, an inductive conclusion is a universally quantified formula that implies all positive evidence but no negative evidence. Though we can found many practical problems that can be put into this framework, I believe that there are much more that cannot — in empirical science and everyday life, it is not very easy to get a non-trivial "rule" without counter-example. Abduction is

similar. Staying in binary logic means that we are only interested in explanations that explains all relevant facts, which are not very common, neither.

To generate and/or evaluate generalizations and explanations with both positive and negative evidence usually means to measure the evidence quantitatively, and the ones that with more positive evidence and less negative evidence are preferred (other things being equal). A natural candidate theory for this is "probabilistic logic" (a combination of first-order predicate logic and probability theory).

Let us use induction as an example. In predicate logic, a general conclusion "Ravens are black" can be represented as a universally quantified proposition $(\forall x)(Raven(x) \rightarrow Black(x))$. To extend it beyond binary logic, we attach a probability to it, to allow it to be "true to a degree". Intuitively, each time a black raven is observed, the probability should be increased a little bit, while when a non-raven is observed, the probability should be decreased a little bit.

Unfortunately, Hempel has found a paradox in this naive solution (Hempel, 1943). $(\forall x)(Raven(x) \rightarrow Black(x))$ is logically identical to $(\forall x)(\neg Black(x) \rightarrow \neg Raven(x))$. Since the probability of the latter is increased by any non-black non-raven (such as a green shirt), so do the former. This is highly counter-intuitive.

This chapter makes no attempt to survey the huge amount of literature on Hempel's "Raven Paradox". What I want to mention is the fact that all the previous solutions are proposed within the framework of first-order predicate logic. I will show that this problem is actually caused by the framework itself, and the paradox does not appear in term logic.

In first-order predicate logic, every general conclusion is represented by a proposition which contains at least one universally quantified variable, such as the x in the previous example. This variable can be substituted by any constant in the domain, and the resulting proposition is either true or false. If we call the constants that make it true "positive evidence" and those make it false "negative evidence", then everything must belongs to one of the two category, and nothing in the domain is irrelevant. Literally, $(\forall x)(Raven(x) \rightarrow Black(x))$ states that "For everything in the domain, either it is a raven, or it is not black". Though it is a meaningful statement, there is a subtle difference between it and "Ravens are black" — the latter is about ravens, not about everything.

The situation in term logic is different. In term logic "Ravens are black" can be represented as $raven \subset black\_thing$, and "Non-black things are not raven" as $(thing - black\_thing) \subset (thing - raven)$. According to the definition, these two statements share common negative evidence (non-black ravens), but the positive evidence for the former

(black ravens) and the latter (non-black non-ravens) are completely different (here we only consider the extension of the concepts). The two statements have the same truth value in binary (extensional) term logic, because there a truth value merely qualitatively indicates whether there is negative evidence for the statement. In a non-binary term logic like NARS, they do not necessarily have the same truth value anymore, so in NARS a green shirt has nothing to do with the system's belief about whether ravens are black, just like crow, as a non-swimmer, provides no evidence for $swimmer \subset bird$, no matter it is a bird or not (see the example in the previous section).

The crucial point is that in term logic, general statements are usually not about everything (except when "everything" or "thing" happen to be the subject or the predicate), and the domain of evidence is only the extension of the subject (and the intension of the predicate, for a logic that consider both extensional inference and intensional inference). I cannot see how first-order predicate logic can be extended or revised to do a similar thing.

In summary, my argument goes like this: the real challenge of abduction and induction is to draw conclusions with conflicting and incomplete evidence. To do this, it is necessary to distinguish positive evidence, negative evidence, and irrelevant information for a given statement. This task can be easily carried out in term logic, though it is hard (if possible) for predicate logic.

Another advantage of term logic over predicate logic is the relation among deduction, abduction, and induction. As described previously, in NARS the three have a simple, natural, and elegant relationship, both in their syntax and semantics. Their definitions and the relationship among them becomes controversial in predicate logic, which is a major issue discussed in the other chapters of this book.

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

- 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.

- With the definition of extension and intension introduced in NARS, it becomes easy to define truth value as a function of available evidence, to consistently represent uncertainty from various sources, and to design truth-value functions accordingly. As a result, different types of inference can be justified by the same experience-grounded semantics, while the difference among them is still visible.

- Abduction and induction become *dual* in the sense that they are completely symmetric to each other, both syntactically and semantically. The difference is that abduction collects evidence from the intensions of the terms in the conclusion, while induction collects evidence from the extensions of the terms. Intuitively, they still correspond to generalization and explanation, respectively.

- With the help of the revision rule, abduction and induction at the problem-solving level becomes incremental and open-ended processes, and they do not follow predetermined algorithms.

- The choice of 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.

Here I want to claim (though I have only discussed part of the reasons in this chapter) that, though First-Order Predicate Logic is still better for binary deductive reasoning, term logic provides a better platform for the enterprise of AI.

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 revisable,
3. from extensional to both extensional and intensional,
4. from deduction only to multiple types of inference,
5. from atomic terms to compound terms.

Though the last issue is beyond the scope of this chapter, it needs to be addressed 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. In this way, term logic can be expended to represent more complex knowledge.

For example, in the previous section "Non-black things are not raven" is represented as $(thing - black\_thing) \subset (thing - raven)$, where

both the subject and the predicate are compound terms formed by simpler terms with the help of the *difference* operator. Similarly, "Ravens are black birds" can be represented as $raven \subset (black\_thing \cap bird)$, where the predicate is the *intersection* of two simpler terms; "Sulfuric acid and sodium hydroxide can neutralize each other" can be represented as $(sulfuric\_acid \times sodium\_hydroxide) \subset neutralization$, where the subject is a *Cartesian product* of two simpler terms.

Though the new version of NARS containing compound terms is still under development, it is obvious that the expressibility of the term-oriented language can be greatly enriched by recursively applying logical operators to form compound terms from simpler terms.

Finally, let us re-visit the relationship between the micro-level (inference step) and macro-level (inference process) perspectives of abduction and induction, in the context of NARS. As described previously, in NARS the words "abduction" and "induction" are used to name (micro-level) inference rules. Though the conclusions derived by these rules still intuitively correspond to explanation and generalization, such a correspondence does not accurately hold at the macro-level. If NARS is given a list of statements to start with, then after many inference steps the system may reach a conclusion, which is recognized by human observers as an explanation (or generalization) of some of the given statements. Such a situation is usually the case that the abduction (or induction) rule has played a major role in the process, though it is rarely the only rule involved. As shown by the example, the answers reported to the user by NARS are rarely pure abductive (or deductive, inductive, and so on). In summary, though different types of inference can be clearly distinguished in each step (at the micro level), a multiple-step inference procedure usually consists of various types of inference, so cannot be accurately classified.

As mentioned at the beginning of this chapter, Peirce introduced the deduction/induction/abduction triad in two levels of reasoning: syllogistic (micro, single step) and inferential (macro, complete process). I prefer to use the triad in the first sense, because it has an elegant and natural formalization in term logic. On the other hand, I doubt that we can identify a similar formalization at the macro level when using "abduction" for hypothesis generation, and "induction" for hypothesis confirmation. It is very unlikely that there is a single, universal method for inference processes like hypothesis generation or confirmation. On the contrary, these processes are typically complex, and vary from situation to situation. For the purpose of Artificial Intelligence, we prefer a constructive explanation, than a descriptive explanation. It is more likely for us to achieve this goal at the micro level than at the macro level.

Because term logic has been ignored by mainstream logic and AI for a long time, it is still too early to draw conclusions about 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 the previous criticisms on term logic can be avoided if we properly extend the logic.

References

Aristotle (1989). *Prior Analytics*. Hackett Publishing Company, Indianapolis, Indiana. Translated by R. Smith.

Hempel, C. (1943). A purely syntactical definition of confirmation. *Journal of Symbolic Logic*, 8:122–143.

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

Popper, K. (1959). *The Logic of Scientific Discovery*. Basic Books, New York.

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

Wang, P. (1995). *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence*. PhD thesis, Indiana University. Available at http://www.cogsci.indiana.edu/farg/peiwang/papers.html.