Non-Axiomatic Logic (NAL) Specification

Pei Wang

September 15, 2010

Contents

List of Tables

Acknowledgment

Thanks to Jeff Thompson for many helpful comments and discussions.

Chapter 1

Introduction

This document provides a complete and up-to-date specification of *Non-Axiomatic Logic* (*NAL*).

1.1 NAL and NARS

NAL is the logic part of *NARS* (*Non-Axiomatic Reasoning System*).

NARS is an AI project aims at a general-purpose thinking machine.

NARS is designed according to the theory that *intelligence is the ability for a system to adapt to its environment while working with insufficient knowledge and resources* [Wang, 1995a, Wang, 2006].

NARS is developed in the framework of reasoning system. The logic part of NARS is NAL, a formal logic, consisting of a formal language *Narsese* and a set of formal inference rules, plus a semantics. The control part of NARS mainly consists of a memory mechanism and an inference control mechanism.

NARS is an attempt to provide a normative model of *general intelligence*, rather than a descriptive model of *human intelligence*, though the latter is a special case of the former, therefore these two types of model are similar in various (though not all) aspects.

As a *normative* model, NAL starts from some basic principles, then derives a concrete design for what a system *should* do to adapt when its knowledge and resources are insufficient with respect to its tasks.

1.2 Structure of NAL

NAL is established in multiple layers, each of which extends the logic by adding new grammar and inference rules, with proper addition of the semantics. Consequently, each layer has a higher expressive and inferential power than the previous ones, so as to give the corresponding NARS a higher level of intelligence.

In the current design, there are 8 layers. Consequently, each of the logic is named as NAL-n, and the corresponding formal language is named Narsese-n, with n being a number between 1 and 8.

This document starts at the meta-language of NAL. Using it, NAL-1 to NAL-8 are introduced one by one, with formal and semi-formal specifications of its addition in language, semantics, and inference rules.

1.3 Specifying NAL

This specification only explains what NAL is and does, rather than why it is designed in this way, what kind of overall functionality is produced, or how it differs from other systems. For those contents, references are provided by citing previous publications on NARS. All the NARS publications referred, except the book [Wang, 2006], are available online at the project website `http://sites.google.com/site/narswang/`.

This document is under constant revision. As an up-to-date description of an on-going research project, this specification of NAL is not identical to the previous publications on NAL in all details. Wherever such a difference occurs, this document should be considered as representing the current opinion of the author.

This document does not address the control part of NARS, which is described in [Wang, 2006, Chapter 6], as well as [Wang, 1996c, Wang, 2004b, Wang, 2009b]. Currently NARS is an open-source project, hosted at `http://code.google.com/p/open-nars/`.

There are still some open issues in the design of NAL. In the document, they are introduced in the footnotes.[1]

[1] Even after all the known issues are resolved, whether NAL is "complete" depends on a new notion of *completeness*, because the traditional notion cannot be applied to non-axiomatic logics. The new notion should be based on a formal definition of adaptive system, whose interaction with the environment is described as streams of sentences in a formal language. In that situation, NAL will be considered as "complete" if (1) Narsese is shown to be powerful enough to describe all possible interactions between a system and its environment, and (2) NAL inference rules are shown to be powerful enough to describe all possible adaptive behaviors of a system.

References

[Wang, 2006, Chapter 2], [Wang, 1995a, Wang, 2007a]

Chapter 2

IL-1: Inheritance Logic

NAL is *described* using several meta-theories, though cannot be *reduced* into any of them, that is, results in NAL and results in any of its meta-theories are distinct, though there are partial overlaps and intuitive similarity here or there. The meta-theories include set theory, formal language theory, first-order predicate logic, and inheritance logic (also known as NAL-0). Since only the last one is not well known, it is specified here.

Inheritance Logic, or *IL*, is an *idealized version* of NAL, in the sense that it is similar to NAL in language, semantics, and inference rule, though it assumes sufficient knowledge and resources. Therefore it is not a "non-axiomatic" logic, but a tool used when building such a logic. For each layer n ($1 \leq n \leq 8$), the corresponding IL-n will be defined first, then the effect of insufficient knowledge and resources is introduced, to turn IL-n into NAL-n. This chapter defines IL-1, the simplest inheritance logic.

2.1 Language: term and inheritance

IL-1, like all members of the IL-NAL family, is a "term logic". This type of logic is characterized by its usage of *categorical* sentences and *syllogistic* inference rules. Therefore, it is also known as "categorical logic" or "syllogistic logic".

Definition 1 *The basic form of a* term *is a* word*, a string of letters in a finite alphabet.*

There is no additional requirement on the alphabet. In this document the alphabet is that of English, plus digits 0 to 9 and a few special signs, such as hyphen ('-').

Definition 2 *The* inheritance copula*, '$\rightarrow$', is a binary relation from one term to another term, and defined by* being reflexive and transitive*.*

There is no additional requirement associated with the inheritance copula beside the above definition.

Definition 3 *The basic form of a* statement *is an* inheritance statement*, "$S \rightarrow P$", where S is the* subject term*, and P is the* predicate term*.*

The "subject-copula-predicate" form of statement is what traditionally called *categorical sentences*.

Definition 4 *IL-1 is defined on a formal language whose sentences are inheritance statements.*

The above definitions are summarized in Table 2.1, using a variant of the Backus-Naur Form (BNF).

$\langle sentence \rangle$	::=	$\langle statement \rangle$
$\langle statement \rangle$	::=	$\langle term \rangle \langle copula \rangle \langle term \rangle$
$\langle copula \rangle$	::=	'$\rightarrow$'
$\langle term \rangle$	::=	$\langle word \rangle$
$\langle word \rangle$	:	a string in a given alphabet

Table 2.1: The Grammar Rules of IL-1

When embedded in expressions, "$S \rightarrow P$" is often written as "$(S \rightarrow P)$" to avoid misunderstanding.

The above formal language is used in IL-1 both for internal representation and external communication.

2.2 Semantics: truth and meaning

Intuitively, "$S \rightarrow P$" states that S is a *specialization* of P, and P is a *generalization* of S. It roughly corresponds to "S is a kind of P" in English.

Definition 5 *A sentence in IL has a binary truth-value, as a proposition in propositional logic.*

The following theorems directly follow from the definitions.

Theorem 1 *For any term X, statement "$X \rightarrow X$" is true.*

Theorem 2 *For any term X, Y, and Z,*

$$((X \to Y) \wedge (Y \to Z)) \subset (X \to Z)$$

In this theorem, IL sentences are treated as propositions, and "$\wedge$" and "$\subset$" are the "conjunction" and "implication" connectives in propositional logic, respectively.

The inheritance relation is neither symmetric nor anti-symmetric. That is, for different X and Y, given "$X \to Y$" alone, the truth-value of "$Y \to X$" cannot be determined.

The initial knowledge of the system, obtained from the environment, is defined as its "experience."

Definition 6 *For a system implementing IL-1, its* experience*, K, is a non-empty and finite set of sentences in IL. In each sentence in K, the subject term and the predicate term are different.*

K can be also represented as a (directed and unweighted) graph, with terms as vertices and statements as edges.

Definition 7 *Given experience K, the system's* beliefs*, K^*, is the transitive closure of K, excluding sentences whose subject and predicate are the same term.*

Therefore, K^* is also a non-empty and finite set of sentences in IL-1, which includes K, as well as the sentences derived from K according to the transitivity of the inheritance relation. In systems implementing IL or NAL, the words "belief" and "knowledge" are usually treated as exchangeable with each other. Therefore, K^* can also be called the *knowledge base* of the system.

Definition 8 *Given experience K, the* truth-value *of a statement is true if it is in K^*, or in the form of $X \to X$, otherwise it is false.*

Therefore there are two types of truth in IL-1: *empirical* and *literal* (or call them *synthetic* and *analytic*, respectively). The former is "true according to experience," and the latter is "true by definition." Truth in these two categories have no overlap.

In IL-1, all analytic truths are *positive*, in the form of "$X \to X$". Synthetic truths may be either positive (on what is *true*) or negative (on what is *false*). In IL-1, negative knowledge are implicitly represented: they are not sentences in IL-1, but propositions in its meta-language. The amount of positive knowledge (i.e., number of beliefs in K^*) increases monotonically with the increase of the experience K, but that is not the case for negative knowledge, which is implicitly defined by the former as "statements that not known to be true" (the Closed-World Assumption).

For a term T that does not appear in K, all statements having T in them are false, except "$T \rightarrow T$".

Definition 9 *Given experience K, let the set of all terms appearing in K to be the* vocabulary *of the system, V_K. Then, the* extension *of a term T is the set of terms $T^E = \{x \mid (x \in V_K) \wedge (x \rightarrow T)\}$. The* intension *of T is the set of terms $T^I = \{x \mid (x \in V_K) \wedge (T \rightarrow x)\}$.*

Obviously, both T^E and T^I are determined with respect to K, so they can also be written as T^E_K and T^I_K. In the following, the simpler notions are used, with the experience K implicitly assumed.

Since "extension" and "intension" are defined in a symmetric way in IL, for any result about one of them, there is a dual result about the other. Each belief of the system reveals part of the intension for the subject term and part of the extension for the predicate term.

Theorem 3 *For any term $T \in V_K$, $T \in (T^E \cap T^I)$. If T is not in V_K, $T^E = T^I = \{\}$, though "$T \rightarrow T$" is still true.*

Definition 10 *Given experience K, the* meaning *of a term T consists of its extension and intension.*

Therefore, the meaning of a term is its relation with other terms, according to the experience of the system. A term T is "meaningless" to the system, if $T^E = T^I = \{\}$ (that is, it has never got into the experience of the system), otherwise it is "meaningful". The larger the extension and intension of a term are, the "richer" its meaning is.

Theorem 4 *If both S and P are in V_K, then $(S \rightarrow P) \equiv (S^E \subseteq P^E) \equiv (P^I \subseteq S^I)$.*

Here "$\equiv$" is the "if and only if" connective in propositional logic.

If "$S \rightarrow P$" is false, it means that the inheritance is incomplete — either $(S^E - P^E)$ or $(P^I - S^I)$ is not empty. However, it does not mean that S and P share no extension or intension.

Theorem 5 $(S^E = P^E) \equiv (S^I = P^I)$.

This means that in IL-1 the extension and intension of a term are mutually determined. Consequently, one of the two uniquely determines the meaning of a term.

Consequently, IL-1 gets an "experience-grounded semantics", since the truth-values of its statements and the meanings of its terms are determined by the experience of the system, except in trivial cases (analytical truths and meaningless terms). No ontological assumption is made about the outside world. To the system, the world is nothing but what the experience reveals.

2.3 Inference: deriving and matching

IL-1 has a single inference rule that derives new knowledge from experience, justified by the transitivity of the inheritance relation. This rule is *syllogistic*, in the sense that it takes two premises, B_1 and B_2, that share a term M, and derives a conclusion between the other two terms S and P. It is shown in Table 2.2.

$B_2 \setminus B_1$	$M \to P$	$P \to M$
$S \to M$	$S \to P$	
$M \to S$		$P \to S$

Table 2.2: The Inference Rule of IL-1

Definition 11 *For different terms S and P, a* question *that can be answered by an IL-based system has one of the following three forms: (1) $S \to P?$, (2) $S \to ?$, and (3) $? \to P$. The '?' in the last two is a "query variable" to be instantiated. A belief $S \to P$ is an answer to any of the three. If no such an answer can be found in K^*, "NO" is answered.*

The first form of question asks for an *evaluation* of a given statement, while the other two ask for a *selection* of a term with a given relation with another term. If there are more than one answers to (2) and (3), they are equally good. Literal truth "$X \to X$" is a trivial answer to such a question, so it is not allowed.

The matching rule is shown in Table 2.3, with Q for question and B for matching belief.

Similar to negative knowledge, in IL-1 questions are not represented as sentences in object language, but in the meta-language only. IL-1 does not accept question "What is not T?".

$B \setminus Q$	$S \to P?$	$S \to ?$	$? \to P$
$S \to P$	$S \to P$	$S \to P$	$S \to P$

Table 2.3: The Matching Rule of IL-1

References

[Wang, 2006, Chapter 3], [Wang, 1994, Wang, 1995a]

Chapter 3

NAL-1: Evidential Inference

NAL-1 turns IL-1 into a non-axiomatic logic, under the Assumption of Insufficient Knowledge and Resources (AIKR).

3.1 Evidence and uncertainty

As shown by Theorem 4, a *perfect* inheritance is equivalent to a *complete* subset relation between the extension or intension of the two terms. It is natural to extend a *complete* subset relation into a *partial* subset relation, and, by the above equivalence, it also extends a *perfect* inheritance into an *imperfect* inheritance.

Furthermore, since the subset relation can be seen as a summary of a set of inheritance statements, an inheritance statement can also be seen as a summary of inheritance statements. Based on this observation, "evidence" of an inheritance statement is introduced.

Definition 12 *For an inheritance statement "$S \to P$", its* evidence *are terms in S^E and P^I. Among them, terms in $(S^E \cap P^E)$ and $(P^I \cap S^I)$ are* positive evidence*, and terms in $(S^E - P^E)$ and $(P^I - S^I)$ are* negative evidence*.*

Here '$\cap$' and '$-$' are the *intersection* and *difference* of sets, respectively, as defined in set theory.

Evidence is defined in this way, because as far as a term in positive evidence is concerned, the inheritance statement is correct; as far as a term in negative evidence is concerned, the inheritance statement is incorrect.

Since according to the previous definition, terms in the extension or intension of a given term are equally weighted, the amount of evidence can be simply measured by the size of the corresponding set.

Definition 13 *For "$S \to P$", the amount of positive, negative, and total evidence is, respectively,*

$$\begin{aligned} w^+ &= |S^E \cap P^E| + |P^I \cap S^I| \\ w^- &= |S^E - P^E| + |P^I - S^I| \\ w &= w^+ + w^- \\ &= |S^E| + |P^I| \end{aligned}$$

When comparing competing beliefs and deriving new conclusions, *relative* measurements are usually preferred over *absolute* measurements, because the evidence of a premise normally cannot be directly used as evidence for the conclusion. Also, it is often more convenient for the measurements to take values from a finite range, while the amount of evidence has no upper bound.

Definition 14 *The truth-value of a statement consists of a pair of real numbers in [0, 1]. One of the two is called* frequency*, defined as $f = w^+/w$; the other is called* confidence*, defined as $c = w/(w + k)$, where k is the "evidential horizon" of the system, a positive constant.*

Informally speaking, frequency is the proportion of positive evidence among all evidence; confidence is the proportion of current available evidence among available evidence in the near future, after the coming of new evidence of amount k. This *evidential horizon* k is a "personality parameter" of the system, in the sense that in different NAL-based systems, it can take different values, and in general it is hard (if possible) to say what value is the best.

In this two-factor truth-value, the frequency factor indicates the ratio between positive and negative evidence, and the confidence factor indicates the ratio between current and future evidence. Since it is impossible to consider infinite future, the evidential horizon k is introduced to restrict "future" into a constant "near future". Since what matters is the *relative* confidence of beliefs, they should be measured against the same evidential horizon, though the exact distance to the horizon (the k value) is not always important.

The above definition implies that in a truth-value, the frequency factor and the confidence factor are *independent* of each other, in the sense that given the value of one, the value of the other is not determined, or even bounded.

The frequency value will be restricted in an interval within the evidential horizon, until the coming evidence reaching amount k.

Definition 15 *The* frequency interval *of a statement* $[l, u]$ *contains its frequency value from the current moment to the moment when the new evidence has amount* k*. The* lower frequency l *is* $w^+/(w+k)$*, and the* upper frequency u *is* $(w^+ + k)/(w+k)$*.*

The frequency of a statement does not necessarily converge to a limit. Even if it does, the limit is not necessarily in the frequency interval at every previous moment.

Definition 16 *The* ignorance *of a statement is measured by the* width *of the frequency interval, i.e.,* $i = u - l$*.*

Theorem 6 *For a statement, its* confidence *and* ignorance *are complement to each other, that is,* $c + i = 1$*.*

The interval representation of uncertainty provides a mapping between the "accurate representation" and the "inaccurate representation" of uncertainty, because "inaccuracy" corresponds to willingness to change a value within a certain range. If in a situation there are only N words that can be used to specify the uncertainty of a statement, and all numerical values are equally possible, the most informative way to communicate is to evenly divide the [0, 1] interval into N section: [0, 1/N], [1/N, 2/N], ..., [(N-1)/N, 1], and to use a label for each section. A special situation of this is to use a single number, with its accuracy, to carry out both frequency and confidence information.

In summary, NAL uses three functionally equivalent representations for the uncertainty (or degree of belief) of a statement:

Amounts of evidence: $\{w^+, w\}$, where $0 \le w^+ \le w$, or using $w^- = w - w^+$ to replace one of the two;

Truth value: $\langle f, c\rangle$, where both f and c are real numbers in [0, 1], and are independent of each other;

Frequency interval: $[l, u]$, where $0 \le l \le u \le 1$, or using $i = u - l$ to replace one of the two.

Among all possible values of the measurements, there are two extreme cases that only appear in the meta-language, and a normal case that actually happen in Narsese:

Null evidence: This is indicated by $w = 0$, $c = 0$, or $i = 1$. It means the system knows nothing at all about the statement, so does not need to be actually represented in the system.

Full evidence: This is indicated by $w = \infty$, $c = 1$, or $i = 0$. It means the system already knows everything about the statement, which cannot occur in a non-axiomatic logic.

Normal evidence: This is indicated by $0 < w$, $0 < c < 1$, or $0 < i < 1$. It means the statement is supported by finite amount of evidence, which is the normal case for every belief in NAL.

Though the extreme cases never appear in actual beliefs of the system, they can be discussed in the meta-language of NAL, as limit cases of the actual beliefs, and therefore play important roles in system design.

This is why IL can be considered as an idealized version of NAL, while still being a meta-logic of it. The beliefs of IL is supported by "full positive evidence", and therefore having *binary* truth-value. On the contrary, in NAL each belief may have both *positive* and *negative* evidence, and the impact of *future* evidence must be considered, too. Therefore, the truth-value *true* of IL can be mapped into truth-value $\langle 1, 1\rangle$ of NAL, since the former assumes that there is neither negative evidence nor future evidence.

For the normal case, formulas for inter-conversion among the three forms are displayed in Table 3.1.

to \ from	$\{w^+, w\}$	$\langle f, c\rangle$	$[l, u]$ (and i)
$\{w^+, w\}$		$w^+ = kfc/(1-c)$ $w = kc/(1-c)$	$w^+ = k\,l/i$ $w = k(1-i)/i$
$\langle f, c\rangle$	$f = w^+/w$ $c = w/(w+k)$		$f = l/(1-i)$ $c = 1-i$
$[l, u]$	$l = w^+/(w+k)$ $u = (w^+ + k)/(w+k)$	$l = fc$ $u = 1 - c(1-f)$	

Table 3.1: The Mappings Among Measurements of Uncertainty

3.2 Grammar and semantics

The grammar of Narsese-1, the language used in NAL-1, is that of IL-1, except that a binary "statement" plus its truth-value becomes a multi-valued "judgment". Also, "question" is included in the object-level of the language, as a statement without truth-value, and may contain variable to be instantiated.

$\langle sentence \rangle$	::=	$\langle judgment \rangle \mid \langle question \rangle$
$\langle judgment \rangle$	::=	$\langle statement \rangle \langle truth\text{-}value \rangle$
$\langle question \rangle$	::=	$\langle statement \rangle \mid$ '?' $\langle copula \rangle \langle term \rangle \mid \langle term \rangle \langle copula \rangle$ '?'
$\langle statement \rangle$	::=	$\langle term \rangle \langle copula \rangle \langle term \rangle$
$\langle copula \rangle$	::=	'$\rightarrow$'
$\langle term \rangle$	::=	$\langle word \rangle$
$\langle truth\text{-}value \rangle$	:	a pair of real number in $[0, 1] \times (0, 1)$
$\langle word \rangle$	:	a string in a given alphabet

Table 3.2: The Grammar of Narsese-1

The truth-value of each judgment is defined by a chunk of evidence represented by IL-1 sentences. In communications between the system and its environment, the other two types of uncertainty representation can also be used in place of the truth-value of a judgment, though within the system they will be translated to (from) truth-value.

Similarly, the definition of "meaning" in NAL-1 also comes from that in IL-1.

Definition 17 *A judgment "$S \rightarrow P \langle f, c \rangle$" indicates that S is in the extension of P and that P is in the intension of S, with the truth-value of the judgment specifying their grades of membership.*

Consequently, the extension and intension of a term in NAL-1 are no longer ordinary sets with well-defined boundaries (as in IL-1), but sets with (two-dimensional) grades of membership.

Definition 18 *The actual* experience *of a system implementing NAL-1 is a stream of Narsese-1 sentences. The experience defined in IL-1 is renamed* idealized experience *in NAL-1.*

What differs *idealized* experience from *actual* experience is:

1. The former contains *true* statements only, while the latter contains questions and *multi-valued* judgments,

2. The former is a *set* (without internal order or duplicated elements), while the latter is a *stream* (where order matters, and duplicate elements are possible).

Since NAL-1 works under AIKR, the transitive closure of its (actual) experience is not defined. The system may not have the resources to exhaust all possible conclusions derivable from given experience, nor can it be assumed that the conclusions will converge to a stable set of beliefs, since new experience comes constantly, and consists of sentences with unrestricted content.

Definition 19 *The* evidential base *of a truth-value is the set of sentences in the experience from which the truth-value is derived.*

Therefore, the evidential base of an input sentence (in the experience of the system) is a set containing itself, while the evidential base of a derived conclusion is the union of the evidential bases of the premises. If the same sentence appears multiple times in experience, each occurrence corresponds to a separate evidential base.

In the actual implementation of NAL, the evidential base of a truth-value is represented by a "stamp" containing sequential numbers of input sentences, with a maximum length. To calculate the union of two evidential bases, the two stamps are interwoven, and the overflow part is ignored. The system decides if two truth-values are based on overlapping evidence by checking if their stamps contain any common element, which may fail to recognize overlapping evidence for beliefs derived from many input sentences, which, though not desired, is inevitable for a system with AIKR.

3.3 Forward inference

As a syllogistic logic, a typical forward inference rule in NAL takes two judgments as premise, and derives a judgment as conclusion, with a truth-value function to calculate the truth-value of the conclusion from those of the premises. That is, it looks like

$$\{premise_1\langle f_1,\, c_1\rangle,\ premise_2\langle f_2,\, c_2\rangle\} \vdash conclusion\langle f,\, c\rangle$$

where $\langle f,\, c\rangle$ is calculated by a truth-value function from $\langle f_1,\, c_1\rangle$ and $\langle f_2,\, c_2\rangle$. Alternatively, the rule can be put into a table where each row and column corresponds to a premise, as in IL-1.

In NAL-1, all the premises and conclusions are inheritance statements, and the two premises share at least one common term. Furthermore, to avoid circular inference, the premises cannot have common stamp elements.

Because the two premises share at least one term, their contents are semantically related to each other. NAL never infers on two arbitrary premises and only considers their truth-values in deriving a conclusion.

For a pair of judgments that do share at least one common term, their structures and the position of the shared term determine the content of the conclusion, as well as the truth-value function.

A truth-value function is usually designed (with a few exceptions) by treating the related measurements in [0, 1] as extended Boolean values, by the following procedure:

1. According to the experience-grounded semantics, decide the uncertainty values of the conclusion for each combination of the values in the premises, when all of them are binary values 0 or 1.

2. Represent each value in the conclusion as a Boolean function of the values in the premises, using Boolean operators "*and*", "*or*", and "*not*". Among the Boolean functions satisfying the given condition, the function selected usually is the simplest, and with an intuitive justification.

3. Assuming variables $x_1, ..., x_n$ are *mutually independent* (i.e., the value of one cannot be bounded by the value of the others), the Boolean operators are extended from $\{0, 1\}$ to [0, 1]:

   Definition 20

$$\begin{aligned} not(x_i) &= 1 - x_i \\ and(x_1, ..., x_n) &= x_1 \times ... \times x_n \\ or(x_1, ..., x_n) &= 1 - (1 - x_1) \times ... \times (1 - x_n) \end{aligned}$$

   When the operators are applied in truth-value functions, the independence requirement is satisfied when the two premises have distinct evidential bases, since the two factors in a truth-value (frequency and confidence) are already independent of each other in this sense.

4. Rewrite the uncertainty functions as truth-value functions if they are not in that form, using the mappings between truth-values and other uncertainty measurements in Table 3.1.

In term logics, when two judgments share exactly one common term, they can be used as premises in an inference rule that derives an inheritance relation between the other two (unshared) terms. When the copula is directed, like *inheritance*, there are four possible combinations of premises and conclusions, as listed in Table 3.3. For each combination of premises, there are two conclusions, corresponding to the two directions of inheritance between the two terms that only appear on one premise. The involved inference type include *deduction*, *abduction*, *induction*, and *exemplification*.

$J_2 \setminus J_1$	$M \to P \langle f_1, c_1 \rangle$	$P \to M \langle f_1, c_1 \rangle$
$S \to M \langle f_2, c_2 \rangle$	$S \to P < F_{ded} >$	$S \to P < F_{abd} >$
	$P \to S < F'_{exe} >$	$P \to S < F'_{abd} >$
$M \to S \langle f_2, c_2 \rangle$	$S \to P < F_{ind} >$	$S \to P < F_{exe} >$
	$P \to S < F'_{ind} >$	$P \to S < F'_{ded} >$

Table 3.3: The Basic Syllogistic Rules

In the table, F_{nnn} indicates the truth-value function that calculates the truth-value of the conclusion, and F'_{nnn} is F_{nnn} with the order of the premises switched. The associated truth-value functions are given in Table 3.4, together with the type of inference. The function F_{ded} is derived from the transitivity of the inheritance relation, while the other three are derived from the definition of evidence.

Deduction	Boolean version:	f	$=$	$and(f_1, f_2)$
		c	$=$	$and(f_1, c_1, f_2, c_2)$
F_{ded}	truth-value version:	f	$=$	$f_1 \times f_2$
		c	$=$	$f_1 \times c_1 \times f_2 \times c_2$
Abduction	Boolean version:	w^+	$=$	$and(f_1, c_1, f_2, c_2)$
		w^-	$=$	$and(f_1, c_1, not(f_2), c_2)$
F_{abd}	truth-value version:	f	$=$	f_2
		c	$=$	$\frac{f_1 \times c_1 \times c_2}{f_1 \times c_1 \times c_2 + k}$
Induction	Boolean version:	w^+	$=$	$and(f_1, c_1, f_2, c_2)$
		w^-	$=$	$and(not(f_1), c_1, f_2, c_2)$
F_{ind}	truth-value version:	f	$=$	f_1
		c	$=$	$\frac{c_1 \times f_2 \times c_2}{c_1 \times f_2 \times c_2 + k}$
Exemplification	Boolean version:	w^+	$=$	$and(f_1, c_1, f_2, c_2)$
		w^-	$=$	0
F_{exe}	truth-value version:	f	$=$	1
		c	$=$	$\frac{f_1 \times c_1 \times f_2 \times c_2}{f_1 \times c_1 \times f_2 \times c_2 + k}$

Table 3.4: The Truth-value Functions of the Basic Syllogistic Rules

In term logics, "conversion" is an inference from a single premise to a conclusion by interchanging the subject and predicate terms of the premise. The conversion rule in NAL is defined in Table 3.5.

$$\boxed{\{P \to S\ \langle f_0,\, c_0\rangle\} \vdash S \to P\ \langle F_{cnv}\rangle}$$

Table 3.5: The Conversion Rules of NAL-1

By definition, statements "$S \to P$" and "$P \to S$" have the same positive evidence, but distinct negative evidence. However, in conversion inference directly letting $w^+ = w_0^+$ and $w^- = 0$ lead to the undesired result that "$P \to S\ \langle 1,\, 1\rangle$" derives "$S \to P\ \langle 1,\, 1\rangle$". Instead, in NAL inference rules evidence for a premise should not be taken as evidence of the same amount for the conclusion (except in a few special rules to be introduced later). A proper truth-value function for the conversion rule can be obtained by treating the conclusion as derived by abduction from premises "$P \to S\langle f_0,\, c_0\rangle$" and "$S \to S\langle 1,\, 1\rangle$", or by induction from premises "$P \to P\langle 1,\, 1\rangle$" and "$P \to S\langle f_0,\, c_0\rangle$". Both of them lead to the function in Table 3.6, which also means that in conversion the premise only provides positive evidence (with the amount of $f_0 \times c_0$) to the conclusion.

Conversion	Boolean version:	w^+	$=$	$and(f_0, c_0)$
		w^-	$=$	0
F_{cnv}	truth-value version:	f	$=$	1
		c	$=$	$\frac{f_0 \times c_0}{f_0 \times c_0 + k}$

Table 3.6: The Truth-value Function of the Conversion Rule

3.4 Revision and choice

In NAL, *revision*, given in Table 3.7, indicates the inference step in which evidence from different sources for the same statement is accumulated. It is applicable when the two premises contains the same statement, and their stamps contain no common element. The two premises are still kept as valid beliefs after the revision.

It is the only two-premise rule in NAL where the evidence of the premises can be directly taken, with the same type and amount, as the evidence of the conclusion (because they all contain the same statement). Therefore, the truth-value function, given in Table 3.8, is not designed according to the general procedure introduced previously, but comes directly from the additivity of the amount of evidence.

$J_2 \setminus J_1$	$S\langle f_1, c_1\rangle$
$S\langle f_2, c_2\rangle$	$S\langle F_{rev}\rangle$

Table 3.7: The Revision Rule

Revision	evidence version:	w^+	$=$	$w_1^+ + w_2^+$
		w	$=$	$w_1 + w_2$
F_{rev}	truth-value version:	f	$=$	$\frac{f_1 c_1(1-c_2)+f_2 c_2(1-c_1)}{c_1(1-c_2)+c_2(1-c_1)}$
		c	$=$	$\frac{c_1(1-c_2)+c_2(1-c_1)}{c_1(1-c_2)+c_2(1-c_1)+(1-c_1)(1-c_2)}$

Table 3.8: The Truth-value Function of the Revision Rule

As in IL-1, judgment "$S \to P\langle f, c\rangle$" provides a *candidate answer* to evaluative question "$S \to P$?", as well as to selective questions "$S \to$?" and "? $\to P$". However, unlike the situation of IL-1, in NAL-1 all candidates are not equally good. The *choice rule* of NAL chooses the better answer between two candidates.

For an *evaluative question* "$S \to P$?", both candidate answers contain the same statement "$S \to P$", though have different truth-values. Between them, the better one is the one with a higher *confidence* value. This is the case because an adaptive system prefers an evaluation supported by more evidence.

For a *selective question* "$S \to$?" or "? $\to P$", the two candidate answers usually suggest different instantiations T_1 and T_2 for the query variable in the question. Between them, the better one is the one with a higher *expectation* value, which is a prediction of the frequency for the statement to be confirmed in the near future. This prediction is based on the past frequency, but more *conservative*, by taking the confidence factor into account. The expectation function is given in Table 3.9.

Expectation	frequency-interval version:	e	$=$	$(l+u)/2$
	evidence-amount version:	e	$=$	$(w^+ + k/2)/(w+k)$
F_{exp}	truth-value version:	e	$=$	$c(f-1/2)+1/2$

Table 3.9: The Expectation Function

In summary, the choice rule is formally defined in Table 3.10, where $S_1 \langle f_1, c_1 \rangle$ and $S_2 \langle f_2, c_2 \rangle$ are two competing answers to a question, and $S \langle F_{cho} \rangle$ is the chosen one. When S_1 and S_2 are the same statement, the one with a higher confidence value is chosen, otherwise the one with a higher expectation value is chosen. It is also a special rule because no new conclusion is derived.

$J_2 \setminus J_1$	$S_1 \langle f_1, c_1 \rangle$
$S_2 \langle f_2, c_2 \rangle$	$S \langle F_{cho} \rangle$

Table 3.10: The Choice Rule

3.5 Backward inference

Backward inference happens when a judgment and a question are taken as premises, and a *derived question* is produced as result. The question derivation rules are specified by the following general principle, or *meta-rule*, using the other (forward inference) rules defined previously.

Question derivation: A question Q and a judgment J will give rise to a new question Q' if and only if an answer for Q can be derived from J and an answer for Q', by a forward inference rule.

Therefore, if a question cannot be properly answered by the choice rule, backward inference is used to recursively "reduce" the question into derived questions, until all of them have direct answers. Then these answers, together with the judgments involved in the previous backward inference, will derive an answer to the original question by forward inference.

In NAL-1, all backward inference rules are obtained by turning the forward syllogistic rules in Table 3.3 in a reverse direction, and the corresponding backward-inference rules are in Table 3.11, where P can be a query variable (marked by '?').

This table turns out to be identical to Table 3.3, if the truth-value functions and the question/judgment difference are ignored.

References

[Wang, 2006, Chapter 3], [Wang, 1994, Wang, 1995b, Wang, 1996a, Wang, 1996b, Wang, 2000, Wang, 2001b, Wang, 2004a, Wang, 2005, Wang, 2009c]

$J \setminus Q$	$M \rightarrow P$	$P \rightarrow M$
$S \rightarrow M\langle f,\, c\rangle$	$S \rightarrow P$ $P \rightarrow S$	$S \rightarrow P$ $P \rightarrow S$
$M \rightarrow S\langle f,\, c\rangle$	$S \rightarrow P$ $P \rightarrow S$	$S \rightarrow P$ $P \rightarrow S$

Table 3.11: The Backward Basic Syllogistic Rules

Chapter 4

NAL-2: Similarity and Sets

In this chapter and the following ones, first the language of IL is extended, then the inference rule of NAL is extended to handle the new items in the language under AIKR.

4.1 Similarity

Definition 21 *For any terms S and P,* similarity *'$\leftrightarrow$' is a copula defined by*

$$(S \leftrightarrow P) \equiv ((S \rightarrow P) \wedge (P \rightarrow S))$$

Since '$\equiv$' and '$\wedge$' are the *equivalence* and *conjunction* connectives in propositional logic, respectively, the expression in the definition is not a statement in IL, but in its meta-language, though it introduce similarity statement '$S \leftrightarrow P$' into IL.

Theorem 7 *Similarity is a reflexive, symmetric, and transitive relation between two terms.*

Theorem 8 $(S \leftrightarrow P) \supset (S \rightarrow P)$

Here '$\supset$' is the *implication* connective in propositional logic. Since in all the following definitions and theorems, symbols like S, P, and M are used for arbitrary terms, they will not be explicitly declared as so.

Theorem 9 $(S \leftrightarrow P) \equiv (S \in (P^E \cap P^I)) \equiv (P \in (S^E \cap S^I))$

Theorem 10 $(S \leftrightarrow P) \equiv (S^E = P^E) \equiv (S^I = P^I)$

That is, "$S \leftrightarrow P$" means the two terms have the same meaning, or are *identical* to each other.

To extend the binary similarity statement in IL-2 to the similarity judgment in NAL-2, the evidence of a similarity statement is defined, alike the evidence of an inheritance statement.

Definition 22 *For similarity statement "$S \leftrightarrow P$", its positive evidence is in $(S^E \cap P^E)$ and $(P^I \cap S^I)$, and its negative evidence is in $(S^E - P^E)$, $(P^E - S^E)$, $(P^I - S^I)$, and $(S^I - P^I)$.*

In NAL-2 a similarity statement is true to a degree, where the amounts of evidence and truth-value are defined in the same way as in NAL-1. In the following, the word "identical" will be reserved for terms S and P when they are related by the binary "$S \leftrightarrow P$" in IL, which is an extreme case of "similar" in both IL and NAL.

Corresponding to the basic syllogistic rules in NAL-1, in NAL-2 there are three combinations of inheritance and similarity, corresponding to *comparison*, *analogy*, and *resemblance*, respectively, as indicated by the names of truth-value functions in Table 4.1. To make the table (as well as the following inference tables) simpler, the truth-values of the premises are omitted in the table, though it is obvious that the truth-value of J_1 and J_2 are $\langle f_1, c_1\rangle$ and $\langle f_2, c_2\rangle$, respectively.

$J_2 \setminus J_1$	$M \to P$	$P \to M$	$M \leftrightarrow P$
$S \to M$		$S \leftrightarrow P\langle F_{com}\rangle$	$S \to P\langle F'_{ana}\rangle$
$M \to S$	$S \leftrightarrow P\langle F_{com}\rangle$		$P \to S\langle F'_{ana}\rangle$
$S \leftrightarrow M$	$S \to P\langle F_{ana}\rangle$	$P \to S\langle F_{ana}\rangle$	$S \leftrightarrow P\langle F_{res}\rangle$

Table 4.1: The Similarity-related Syllogistic Rules

The associated truth-value functions are given in Table 4.2.

4.2 Compound terms

To represent more complicated experience, "compound terms" are needed.

Definition 23 *A* compound term *$(con\ c_1\ \cdots\ c_n)$ is a term formed by a* term connector*, con, that connects one or more terms $c_1, \cdots, c_n$, called the* component(s) *of the compound. The order of the components usually matters.*

Comparison	Boolean version:	$w^+ = and(f_1, c_1, f_2, c_2)$
		$w = and(or(f_1, f_2), c_1, c_2)$
F_{com}	truth-value version:	$f = \frac{f_1 \times f_2}{f_1 + f_2 - f_1 \times f_2}$
		$c = \frac{(f_1 + f_2 - f_1 \times f_2) \times c_1 \times c_2}{(f_1 + f_2 - f_1 \times f_2) \times c_1 \times c_2 + k}$
Analogy	Boolean version:	$f = and(f_1, f_2)$
		$c = and(c_1, f_2, c_2)$
F_{ana}	truth-value version:	$f = f_1 \times f_2$
		$c = c_1 \times f_2 \times c_2$
Resemblance	Boolean version:	$f = and(f_1, f_2)$
		$c = and(or(f_1, f_2), c_1, c_2)$
F_{ana}	truth-value version:	$f = f_1 \times f_2$
		$c = (f_1 + f_2 - f_1 \times f_2) \times c_1 \times c_2$

Table 4.2: The Truth-value Functions of the Similarity-related Rules

Definition 24 *Each term in NAL has a syntactical* complexity. *The complexity of an atomic term (i.e., word) is 1. The complexity of a compound term is 1 plus the sum of the complexity of its components.*

Sometimes the "infix" format of a compound term can be used to write $(con\ c_1\ \cdots\ c_n)$ as $(c_1\ con\ \cdots\ con\ c_n)$, and the syntactical complexity of the two forms are the same.

When introducing term operators with two or more components in the following, usually they are only defined with two components, and the general case (for both the above prefix representation and the infix representation) is translated into the two-component case by the following definition.

Definition 25 *If* $c_1\ \cdots\ c_n$ *(*$n > 2$*) are terms, and con is a term connector defined as taking two or more arguments, then both* $(con\ c_1\ \cdots\ c_n)$ *and* $(c_1\ con\ \cdots\ con\ c_n)$ *are defined recursively as* $(con\ (con\ c_1\ \cdots\ c_{n-1})\ c_n)$*, though the latter form has a higher syntactical complexity.*

In Narsese, all term connectors are defined in the grammar, and with predetermined (experience-independent) meaning.

The meaning of a compound term is related to the meaning of the components, so identical components form identical compounds.

Definition 26 *In IL, two compound terms are* identical *if they have the same term connector and pairwise identical components.*

$$((c_1 \leftrightarrow d_1) \wedge \cdots \wedge (c_n \leftrightarrow d_n)) \supset ((con\ c_1\ \cdots\ c_n) \leftrightarrow (con\ d_1\ \cdots\ d_n))$$

Since the interrelations among components influence the meaning of a compound, identical compound terms do not necessarily have identical pairwise components. An exception is the compounds that have a sole component.

Definition 27 *In IL, two compound terms with sole components are* identical *if and only if they have the same term connector and identical components.*

$$(c \leftrightarrow d) \equiv ((con\ c) \leftrightarrow (con\ d))$$

Just like there are analytical truth and empirical truth, the meaning of a compound term has two parts, an *analytical* part and an *empirical* part, where the former is determined by its definitional relation with its components and other analytical truths about the term, while the latter comes from the system's experience when the compound term is used as a whole.

All compound terms can be used by the inference rules as atomic terms. When doing so, their internal structures are ignored. Furthermore, compound terms can directly appear in the (idealized or actual) experience of the system.

Consequently, in NAL the meaning of a compound term is not completely reducible to the meanings of its components plus the meaning of the term connector, though related to them.

4.3 Sets and derivative copulas

Definition 28 *If T is a term, the* extensional set *with T as the only component, $\{T\}$, is a compound term, and its meaning is defined by*

$$(\forall x)((x \rightarrow \{T\}) \equiv (x \leftrightarrow \{T\})).$$

That is, a compound term with such a form is like a set defined by a sole element or individual. The compound therefore has a special property: all terms in the extension of $\{T\}$ must be identical to it, and no term can be more *specific* than it (though it is possible for some terms to be more specific than T).

This compound term uses a special format, with '{ }' as term connector.

Theorem 11 *For any term T, $\{T\}^E \subseteq \{T\}^I$.*

On the other hand, $\{T\}^I$ is not necessarily included in $\{T\}^E$.

An *instance* copula, '$\circ\!\!\rightarrow$', is another way to represent the same information.

Definition 29 *The instance statement "$S \circ\!\!\rightarrow P$" is defined by the inheritance statement "$\{S\} \rightarrow P$."*

Theorem 12 $((S \circ\!\!\rightarrow M) \wedge (M \rightarrow P)) \supset (S \circ\!\!\rightarrow P)$.

However, "$S \rightarrow M$" and "$M \circ\!\!\rightarrow P$" does not imply "$S \circ\!\!\rightarrow P$."

Theorem 13 $(S \circ\!\!\rightarrow \{P\}) \equiv (S \rightarrow P)$.

"$T \circ\!\!\rightarrow \{T\}$" follows as a special case. On the other hand, the statement "$T \circ\!\!\rightarrow T$" is not an analytical truth, though may be an empirical one.

According to the duality between extension and intension, another special compound term and the corresponding copula are defined.

Definition 30 *If T is a term, the* intensional set *with T as the only component, $[T]$, is a compound term, and its meaning is defined by*

$$(\forall x)(([T] \rightarrow x) \equiv ([T] \leftrightarrow x)).$$

That is, a compound term with such a form is like a set defined by a sole attribute or feature. The compound therefore has a special property: all terms in the intension of $[T]$ must be identical to it, and no term can be more *general* than it (though it is possible for some terms to be more general than T).

This compound term also uses a special format, with '[ ]' as term connector.

Theorem 14 *For any term T, $[T]^I \subseteq [T]^E$.*

On the other hand, $[T]^E$ is not necessarily included in $[T]^I$.

A *property* copula, '$\rightarrow\!\!\circ$', is another way to represent the same information.

Definition 31 *The property statement "$S \rightarrow\!\!\circ P$" is defined by the inheritance statement "$S \rightarrow [P]$."*

Theorem 15 $(S \rightarrow M) \wedge (M \rightarrow\!\!\circ P) \supset (S \rightarrow\!\!\circ P)$.

However, "$S \rightarrow\!\!\circ M$" and "$M \rightarrow P$" does not imply "$S \rightarrow\!\!\circ P$."

Theorem 16 $([S] \rightarrow\!\!\circ P) \equiv (S \rightarrow P)$.

"$[T] \rightarrow\!\!\circ T$" follows as a special case. On the other hand, the statement "$T \rightarrow\!\!\circ T$" is not an analytical truth, though may be an empirical one.

An *instance-property* copula, '$\circ\!\!\rightarrow\!\!\circ$', is defined by combining '$\circ\!\!\rightarrow$' and '$\rightarrow\!\!\circ$'.

Definition 32 *The instance-property statement "$S \circ\!\!\rightarrow\!\!\circ P$" is defined by the inheritance statement "$\{S\} \rightarrow [P]$."*

Intuitively, it states that an instance S has a property P.

Theorem 17 $(S \circ\!\!\rightarrow\!\!\circ P) \equiv (\{S\} \rightarrow\!\!\circ P) \equiv (S \circ\!\!\rightarrow [P])$

4.4 Grammar and inference rules

In summary, while all the grammar rules of Narsese-1 are still valid in NAL-2, there are additional grammar rules of Narsese-2, as listed in Table 4.3.

$\langle copula \rangle$	$::=$	'$\leftrightarrow$' | '$\circ\!\!\rightarrow$' | '$\rightarrow\!\!\circ$' | '$\circ\!\!\rightarrow\!\!\circ$'
$\langle term \rangle$	$::=$	'$\{$'$\langle term \rangle$'$\}$' | '$[$'$\langle term \rangle$'$]$'

Table 4.3: The New Grammar Rules of Narsese-2

Since each derivative copula is fully defined in terms of the inheritance copula, its semantics and relevant inference rules can be derived from those in NAL-1. To simplify the implementation of the system, derivative copulas *instance*, *property* and *instance-property* are only used in the input/output interface, and within the system they are translated into *inheritance*. Therefore there is no need to introduce inference rules for them. The same thing cannot be done to the copula *similarity*. Though in IL-2 the binary form of *similarity* is defined in terms of the *inheritance*, in NAL-2 *similarity* judgments usually cannot be translated into equivalent *inheritance* judgments. Therefore, NAL-2 uses five copulas in its interface language, but only keep two of them (*inheritance* and *similarity*) in its internal representation, without losing any power in expression and inference.

References

[Wang, 2006, Chapter 4], [Wang, 1994, Wang, 1995a, Wang, 2009a]

Chapter 5

NAL-3: Intersections and Differences

In NAL-3, compound terms are composed by combining the extension or intension of existing terms in certain way.

5.1 Intersections

Definition 33 *Given terms* T_1 *and* T_2*, their* extensional intersection*,* $(T_1 \cap T_2)$*, is a compound term defined by*

$$(\forall x)((x \to (T_1 \cap T_2)) \equiv ((x \to T_1) \wedge (x \to T_2))).$$

From right to left, the equivalence expression defines the extension of the compound, i.e., "$(x \to T_1) \wedge (x \to T_2)$" implies "$x \to (T_1 \cap T_2)$"; from left to right, it defines the intension of the compound, i.e., "$(T_1 \cap T_2) \to (T_1 \cap T_2)$" implies "$(T_1 \cap T_2) \to T_1$" and "$(T_1 \cap T_2) \to T_2$."

Theorem 18

$$(T_1 \cap T_2)^E = T_1^E \cap T_2^E, \ (T_1 \cap T_2)^I = T_1^I \cup T_2^I$$

In the above expressions, the '$\cap$' sign is used in two different senses. On the right-side of the first expression, it indicates the intersection of sets, but on the left-side of the two expressions, it is the term connector of extensional intersections.

Definition 34 *Given terms* T_1 *and* T_2*, their* intensional intersection*,* $(T_1 \cup T_2)$*, is a compound term defined by*

$$(\forall x)(((T_1 \cup T_2) \to x) \equiv ((T_1 \to x) \wedge (T_2 \to x))).$$

From right to left, the equivalence expression defines the intension of the compound, i.e., "$(T_1 \rightarrow x) \wedge (T_2 \rightarrow x)$" implies "$(T_1 \cup T_2) \rightarrow x$"; from left to right, it defines the extension of the compound, i.e., "$(T_1 \cup T_2) \rightarrow (T_1 \cup T_2)$" implies "$T_1 \rightarrow (T_1 \cup T_2)$" and "$T_2 \rightarrow (T_1 \cup T_2)$."

Theorem 19

$$(T_1 \cup T_2)^I = T_1^I \cap T_2^I,\ (T_1 \cup T_2)^E = T_1^E \cup T_2^E$$

The duality of *extension* and *intension* in NAL corresponds to the duality of *intersection* and *union* in set theory — *intensional intersection* corresponds to *extensional union*, and *extensional intersection* corresponds to *intensional union*.

Both operators can be extended to take more than two arguments. Since '$\cap$' and '$\cup$' are both associative and symmetric, the order of their components does not matter.

Theorem 20

$$\begin{array}{l}(T_1 \cap T_2) \leftrightarrow (T_2 \cap T_1) \\ (T_1 \cup T_2) \leftrightarrow (T_2 \cup T_1)\end{array}$$

Theorem 21

$$\begin{array}{l}(T_1 \cap T_2) \rightarrow T_1 \\ T_1 \rightarrow (T_1 \cup T_2)\end{array}$$

Theorem 22

$$\begin{array}{l}(T \cup T) \leftrightarrow T \\ (T \cap T) \leftrightarrow T\end{array}$$

Theorem 23

$$\begin{array}{rcllr}
T_1 \rightarrow M & \wedge & \neg((T_1 \cup T_2) \rightarrow M) & \supset & \neg(T_2 \rightarrow M) \\
\neg(T_1 \rightarrow M) & \wedge & (T_1 \cap T_2) \rightarrow M & \supset & T_2 \rightarrow M \\
M \rightarrow T_1 & \wedge & \neg(M \rightarrow (T_1 \cap T_2)) & \supset & \neg(M \rightarrow T_2) \\
\neg(M \rightarrow T_1) & \wedge & M \rightarrow (T_1 \cup T_2) & \supset & M \rightarrow T_2
\end{array}$$

Here '$\neg$' is the negation operator in propositional logic.

Theorem 24

$$\begin{array}{lcl}
S \rightarrow P & \supset & (S \cup M) \rightarrow (P \cup M) \\
S \rightarrow P & \supset & (S \cap M) \rightarrow (P \cap M) \\
S \leftrightarrow P & \supset & (S \cup M) \leftrightarrow (P \cup M) \\
S \leftrightarrow P & \supset & (S \cap M) \leftrightarrow (P \cap M)
\end{array}$$

In the results of the above theorem, M can be any term in V_K. The same is assumed for some other theorems to be introduced later.

Definition 35 *If* $T_1, \cdots, T_n$ *($n \geq 2$) are different terms, a* compound extensional set $\{T_1, \cdots, T_n\}$ *is defined as* $(\cup \{T_1\} \cdots \{T_n\})$*; a* compound intensional set $[T_1, \cdots, T_n]$ *is defined as* $(\cap [T_1] \cdots [T_n])$*.*

In this way, extensional sets and intensional sets can both have multiple components. The former defines a term by enumerating its *instances*, and the latter by enumerating its *properties*. The order of the components does not matter. These multi-component sets no longer have the property of single-component sets that their extension or intension is minimum.

Theorem 25

$$\begin{array}{rcl}(\forall x)((\{x\} \rightarrow \{T_1, \cdots, T_n\}) & \equiv & ((x \leftrightarrow T_1) \vee \cdots \vee (x \leftrightarrow T_n))) \\ (\forall x)(([T_1, \cdots, T_n] \rightarrow [x]) & \equiv & ((T_1 \leftrightarrow x) \vee \cdots \vee (T_n \leftrightarrow x)))\end{array}$$

5.2 Differences

Definition 36 *If* T_1 *and* T_2 *are different terms, their* extensional difference*,* $(T_1 - T_2)$*, is a compound term defined by*

$$(\forall x)((x \rightarrow (T_1 - T_2)) \equiv ((x \rightarrow T_1) \wedge \neg(x \rightarrow T_2))).$$

From right to left, the equivalence expression defines the extension of the compound, i.e., "$(x \rightarrow T_1) \wedge \neg(x \rightarrow T_2)$" implies "$x \rightarrow (T_1 - T_2)$"; from left to right, it defines the intension of the compound, i.e., "$(T_1 - T_2) \rightarrow (T_1 - T_2)$" implies "$(T_1 - T_2) \rightarrow T_1$" and "$\neg((T_1 \cap T_2) \rightarrow T_2)$."

Obviously, $(T_2 - T_1)$ can also be defined, but it will be different from $(T_1 - T_2)$.

Theorem 26

$$(T_1 - T_2)^E = T_1^E - T_2^E, \ (T_1 - T_2)^I = T_1^I$$

Definition 37 *If* T_1 *and* T_2 *are different terms, their* intensional difference*,* $(T_1 \ominus T_2)$*, is a compound term defined by*

$$(\forall x)(((T_1 \ominus T_2) \rightarrow x) \equiv ((T_1 \rightarrow x) \wedge \neg(T_2 \rightarrow x))).$$

From right to left, the equivalence expression defines the intension of the compound, i.e., "$(T_1 \rightarrow x) \wedge \neg(T_2 \rightarrow x)$" implies "$(T_1 \ominus T_2) \rightarrow x$"; from left to right, it defines the extension of the compound, i.e., "$(T_1 \ominus T_2) \rightarrow (T_1 \ominus T_2)$" implies "$T_1 \rightarrow (T_1 \ominus T_2)$" and "$\neg(T_2 \rightarrow (T_1 \ominus T_2))$."

Theorem 27

$$(T_1 \ominus T_2)^I = T_1^I - T_2^I,\ (T_1 \ominus T_2)^E = T_1^E$$

Theorem 28

$$\begin{array}{l} (T_1 - T_2) \rightarrow T_1 \\ T_1 \rightarrow (T_1 \ominus T_2) \end{array}$$

Theorem 29

$$\begin{array}{l} M \rightarrow (T_1 - T_2) \supset \neg(M \rightarrow T_2) \\ (T_1 \ominus T_2) \rightarrow M \supset \neg(T_2 \rightarrow M) \end{array}$$

Unlike the *intersection* operators, the *difference* operators cannot take more than two arguments. Also, neither $(T - T)$ nor $(T \ominus T)$ is a valid term.

Theorem 30

$$\begin{array}{rcccr} T_1 \rightarrow M & \wedge & \neg((T_1 \ominus T_2) \rightarrow M) & \supset & T_2 \rightarrow M \\ \neg(T_1 \rightarrow M) & \wedge & \neg((T_2 \ominus T_1) \rightarrow M) & \supset & \neg(T_2 \rightarrow M) \\ M \rightarrow T_1 & \wedge & \neg(M \rightarrow (T_1 - T_2)) & \supset & M \rightarrow T_2 \\ \neg(M \rightarrow T_1) & \wedge & \neg(M \rightarrow (T_2 - T_1)) & \supset & \neg(M \rightarrow T_2) \end{array}$$

Theorem 31

$$\begin{array}{rcl} S \rightarrow P & \supset & (S - M) \rightarrow (P - M) \\ S \rightarrow P & \supset & (M - P) \rightarrow (M - S) \\ S \rightarrow P & \supset & (S \ominus M) \rightarrow (P \ominus M) \\ S \rightarrow P & \supset & (M \ominus P) \rightarrow (M \ominus S) \\ S \leftrightarrow P & \supset & (S - M) \leftrightarrow (P - M) \\ S \leftrightarrow P & \supset & (M - P) \leftrightarrow (M - S) \\ S \leftrightarrow P & \supset & (S \ominus M) \leftrightarrow (P \ominus M) \\ S \leftrightarrow P & \supset & (M \ominus P) \leftrightarrow (M \ominus S) \end{array}$$

Theorem 32

$$\begin{array}{ccc} (\{T_1,\ \cdots,\ T_n\} - \{T_n\}) & \leftrightarrow & \{T_1,\ \cdots,\ T_{n-1}\} \\ ([T_1,\ \cdots,\ T_n] \ominus [T_n]) & \leftrightarrow & [T_1,\ \cdots,\ T_{n-1}] \end{array}$$

5.3 Grammar and inference rules

The additional grammar rules of Narsese-3 are listed in Table 5.1.

The previous grammar rule for extensional set and intensional set becomes a special case of the new rule.

$\langle term \rangle$	$::=$	$`\{' \langle term \rangle^{+} `\}'$
		$\mid `[' \langle term \rangle^{+} `]'$
		$\mid `(\cap' \langle term \rangle \langle term \rangle^{+} `)'$
		$\mid `(\cup' \langle term \rangle \langle term \rangle^{+} `)'$
		$\mid `(-' \langle term \rangle \langle term \rangle `)'$
		$\mid `(\ominus' \langle term \rangle \langle term \rangle `)'$

Table 5.1: The New Grammar Rules of Narsese-3

$J_2 \setminus J_1$	$M \to T_1$	$T_1 \to M$
$T_2 \to M$		$(T_1 \cup T_2) \to M \ \langle F_{int} \rangle$ $(T_1 \cap T_2) \to M \ \langle F_{uni} \rangle$ $(T_1 \ominus T_2) \to M \ \langle F_{dif} \rangle$ $(T_2 \ominus T_1) \to M \ \langle F'_{dif} \rangle$
$M \to T_2$	$M \to (T_1 \cap T_2) \ \langle F_{int} \rangle$ $M \to (T_1 \cup T_2) \ \langle F_{uni} \rangle$ $M \to (T_1 - T_2) \ \langle F_{dif} \rangle$ $M \to (T_2 - T_1) \ \langle F'_{dif} \rangle$	

Table 5.2: The Composition Rules of NAL-3

Each inference rule in Table 5.2 introduce a compound term in conclusion. Such a rule is applicable only when T_1 and T_2 are different, and do not have each other as component. Also, the two premises cannot be based on overlapping evidence.

The truth-value functions in Table 5.2 are defined in Table 5.3, in an extended Boolean version.

The frequency functions are obtained based on the assumption that the two premises have independent truth-values, which is assumed when the two do not use overlapping evidence. When T_1 and T_2 are either highly similar or opposite to each other, the compound terms produced by these rules may not have much practical value. However, even when it happens, it is an issue to be handled by inference control, not by logic.

In the confidence functions, each case for the conclusion to reach its maximum is separately considered. The *plus* operator is used in place of an *or* operator, because the two cases involved are mutually exclusive, rather than independent of each other.

F_{int}	:	**Intersection**
f	$=$	$and(f_1, f_2)$
c	$=$	$or(and(not(f_1), c_1), and(not(f_2), c_2)) + and(f_1, c_1, f_2, c_2)$
F_{uni}	:	**Union**
f	$=$	$or(f_1, f_2)$
c	$=$	$or(and(f_1, c_1), and(f_2, c_2)) + and(not(f_1), c_1, not(f_2), c_2)$
F_{dif}	:	**Difference**
f	$=$	$and(f_1, not(f_2))$
c	$=$	$or(and(not(f_1), c_1), and(f_2, c_2)) + and(f_1, c_1, not(f_2), c_2)$

Table 5.3: The Truth-value Functions of the Composition Rules

References

[Wang, 2006, Chapter 4], [Wang, 2004c, Wang, 2007b]

Chapter 6

NAL-4: Products, Relations, and Images

NAL-4 has the capability of representing and processing arbitrary relations among terms that cannot be treated by the copulas.

6.1 Products and relations

Intuitively, a "product" is a compound term consisting of a sequence of components.

Definition 38 *For two terms T_1 and T_2, their* product $(T_1 \times T_2)$ *is a compound term defined by*

$$((S_1 \times S_2) \rightarrow (P_1 \times P_2)) \equiv ((S_1 \rightarrow P_1) \wedge (S_2 \rightarrow P_2)).$$

This definition can be extended to allow more than two components in a product. The product connector allows duplicate components. The order of components matters. The prefix format can be used for products.

Theorem 33

$$(S \rightarrow P) \equiv ((M \times S) \rightarrow (M \times P)) \equiv ((S \times M) \rightarrow (P \times M))$$

$$(S \leftrightarrow P) \equiv ((M \times S) \leftrightarrow (M \times P)) \equiv ((S \times M) \leftrightarrow (P \times M))$$

Theorem 34

$$((S_1 \times S_2) \leftrightarrow (P_1 \times P_2)) \equiv ((S_1 \leftrightarrow P_1) \wedge (S_2 \leftrightarrow P_2))$$

As a special case of the definition of product, when the terms involved are products with common components, the system can "concatenate" them into longer products with more than two components:

Theorem 35

$$(((\times, S_1, S_2) \leftrightarrow (\times, P_1, P_2)) \wedge ((\times, S_1, S_3) \leftrightarrow (\times, P_1, P_3)))$$
$$\equiv ((\times, S_1, S_2, S_3) \leftrightarrow (\times, P_1, P_2, P_3))$$

Theorem 36

$$\{(x \times y) \,|\, x \in T_1^E, y \in T_2^E\} \subseteq (T_1 \times T_2)^E$$

$$\{(x \times y) \,|\, x \in T_1^I, y \in T_2^I\} \subseteq (T_1 \times T_2)^I$$

The '$\subseteq$' cannot be replaced by '$=$' in the above theorem, because $(T_1 \times T_2)^E$ and $(T_1 \times T_2)^I$ may contain other terms that are not products.

Definition 39 *A* relation *is a term R such that there is a product $(T_1 \times T_2)$ satisfying "$(T_1 \times T_2) \rightarrow R$" or "$R \rightarrow (T_1 \times T_2)$".*

Since "$(T_1 \times T_2) \rightarrow (T_1 \times T_2)$", a product is a relation, though a relation is not necessarily a product. In NAL, a relation can be an atomic term.

Though in the meta-language of NAL, a copula (*inheritance* or *similarity*) is a "relation" as defined set theory, it is not a *relation* in Narsese, as defined above. A copula has a fixed meaning provided in the meta-language of NAL, while a relation has an experience-grounded meaning, as other terms in Narsese.

6.2 Images

Definition 40 *For a relation R and a product $(\times\ T_1\ T_2)$, the* extensional image *operator, '$\perp$', and* intensional image *operator, '$\top$', of the relation on the product are defined as the following, respectively:*

$$((\times\ T_1\ T_2) \rightarrow R) \equiv (T_1 \rightarrow (\perp\ R\ \diamond\ T_2)) \equiv (T_2 \rightarrow (\perp\ R\ T_1\ \diamond))$$

$$(R \rightarrow (\times\ T_1\ T_2)) \equiv ((\top\ R\ \diamond\ T_2) \rightarrow T_1) \equiv ((\top\ R\ T_1\ \diamond) \rightarrow T_2)$$

where '$\diamond$' is a special symbol indicating the location of T_1 or T_2 in the product, and it can appear in any place, except the first (which is the relation), in the component list. When it appears at the second place, the image can also be written in infix format as $(R \perp T_2)$ or $(R \top T_2)$.

The above definition can be extended to include products with more than two components, where the image can only be written in the prefix format.

In general, $(R \perp T)$ and $(R \top T)$ are different, but there are situations where they are the same.

Theorem 37

$$T_1 \leftrightarrow ((T_1 \times T_2) \perp T_2)$$

$$T_1 \leftrightarrow ((T_1 \times T_2) \top T_2)$$

Theorem 38

$$((R \perp T) \times T) \rightarrow R$$

$$R \rightarrow ((R \top T) \times T)$$

The '$\rightarrow$' in the above theorem cannot be replaced by the '$\leftrightarrow$'.

Theorem 39

$$\begin{array}{lcl} S \rightarrow P & \supset & (S \perp M) \rightarrow (P \perp M) \\ S \rightarrow P & \supset & (S \top M) \rightarrow (P \top M) \\ S \rightarrow P & \supset & (M \perp P) \rightarrow (M \perp S) \\ S \rightarrow P & \supset & (M \top P) \rightarrow (M \top S) \end{array}$$

6.3 Grammar and inference rules

In summary, NAL-4 introduces the new grammar rules in Table 6.1.

$\langle term \rangle$	$::=$	$`(\times' \langle term \rangle \langle term \rangle^{+}`)'$
		$\mid `(\perp' \langle term \rangle \langle term \rangle^{*} ` \diamond' \langle term \rangle^{*} `)'$
		$\mid `(\top' \langle term \rangle \langle term \rangle^{*} ` \diamond' \langle term \rangle^{*} `)'$

Table 6.1: The New Grammar Rules of Narsese-4

There is no new inference rule directly defined in NAL-4, except the equivalence and implication propositions in the definitions and theorems, which will be turned into inference rules later.

References

[Wang, 2006, Chapter 4], [Wang, 2004c, Wang, 2007b]

Chapter 7

NAL-5: Statements as Terms

When a statement is treated as a term, there are *statements on statements*, as well as inference on this kind of *higher-order* statements.

7.1 Inference: higher-order vs. first-order

The new grammar rules of Narsese-5 are listed in Table 7.1. It includes "higher-order statements" (statements on statements), so that NAL-5 can carry out "higher-order inference" (inference on higher-order statements), while NAL-4 is "first-order" (where *statement* and *term* are distinct).

$\langle term \rangle$	::=	'(' $\langle statement \rangle$ ')'
$\langle statement \rangle$	::=	$\langle term \rangle$
		| '($\neg$' $\langle statement \rangle$ ')'
		| '($\wedge$' $\langle statement \rangle \langle statement \rangle^{+}$ ')'
		| '($\vee$' $\langle statement \rangle \langle statement \rangle^{+}$ ')'
$\langle copula \rangle$	::=	'$\Rightarrow$' | '$\Leftrightarrow$'

Table 7.1: The New Grammar Rules of Narsese-5

In IL-5 and NAL-5, a statement can be treated as a term, and a term can also be used as a statement. However, it does not mean that there is no difference between *term* and *statement*. In IL and NAL, a statement has both meaning and truth-value, while a non-statement term only has meaning, no truth-value.

The "propositional attitudes", such as "believe" and "know", are represented in Narsese as relations between a ordinary term and a statement, so the corresponding statements are higher-order statements.

Compound statements can be formed using statement connectors *negation* ('$\neg$'), *conjunction* ('$\wedge$'), and *disjunction* ('$\vee$').

The two copulas, *implication* ('$\Rightarrow$') and *equivalence* ('$\Leftrightarrow$'), are "higher-order", because they are defined between two statements. In their binary form, '$\Rightarrow$' and '$\Leftrightarrow$' are different from '$\supset$' and '$\equiv$', though their intuitive meanings ("if" and "if-and-only-if", respectively) are similar. The former two belong to the object language (Narsese), while the latter two belong to the meta-language of Narsese (propositional calculus).

Definition 41 *If S_1 and S_2 are statements, "$S_1 \Rightarrow S_2$" is true if and only if in IL S_2 can be derived from S_1.*

The derivation in the above definition can consists of any (finite) number of inference steps.

Theorem 40 *The* implication *copula, '$\Rightarrow$', is a reflexive and transitive relation from one statement to another statement.*

Since the above theorem of implication is parallel to the definition of inheritance in IL-1, higher-order inference in IL-5 can be defined as *partially isomorphic* to first-order inference. The correspondences are listed in Table 7.2.

First-Order IL	**Higher-Order IL**
inheritance	implication
similarity	equivalence
subject	antecedent
predicate	consequent
extension	sufficient condition
intension	necessary condition
extensional intersection	conjunction
intensional intersection	disjunction

Table 7.2: Isomorphism of First-Order and Higher-Order IL

The definitions of the new notions in Table 7.2 are in the following.

Definition 42 *An* implication statement *consists of two statements related by the implication copula. In implication statement "$A \Rightarrow C$", A is the* antecedent*, and C is the* consequent*.*

Definition 43 *Given idealized experience K expressed in the formal language of IL-5, the* sufficient conditions *of a statement T is the set of statements $T^S = \{x \mid x \in V_K \wedge x \Rightarrow T\}$; the* necessary conditions *of T is the set of statements $T^N = \{x \mid x \in V_K \wedge T \Rightarrow x\}$.*

Definition 44 *For an implication statement "$A \Rightarrow C$", its* evidence *are statements in A^S and C^N. Among them, statements in $(A^S \cap C^S)$ and $(C^N \cap A^N)$ are* positive evidence*, while statements in $(A^S - C^S)$ and $(C^N - A^N)$ are* negative evidence*.*

Definition 45 Equivalence *copula, '$\Leftrightarrow$', is defined by*

$$(A \Leftrightarrow C) \equiv ((A \Rightarrow C) \wedge (C \Rightarrow A))$$

The amounts of evidence and the truth-value for a higher-order statement are defined in the same way from evidence as for a first-order statement.

Definition 46 *When S_1 and S_2 are different statements, their* conjunction*, $(S_1 \wedge S_2)$, is a compound statement defined by*

$$(\forall x)((x \Rightarrow (S_1 \wedge S_2)) \equiv ((x \Rightarrow S_1) \wedge (x \Rightarrow S_2))).$$

Their disjunction*, $(S_1 \vee S_2)$, is a compound statement defined by*

$$(\forall x)(((S_1 \vee S_2) \Rightarrow x) \equiv ((S_1 \Rightarrow x) \wedge (S_2 \Rightarrow x))).$$

The above two statement connectors are symmetric, and can be extended to take more than two arguments.

Because of this isomorphism between copulas, there are isomorphic inference rules in NAL-5 for the following rules defined previously (and each pair of rules uses the same truth-value function):

- The NAL-1 rules for deduction, abduction, induction, exemplification, and conversion.
- The NAL-2 rules for comparison, analogy, and resemblance.
- The NAL-3 rules for the composition and decomposition of intersections.
- The backward inference rules corresponding to the above forward inference rules.

The term connectors for (extensional/intensional) set, product, and (extensional/intensional) image are not involved in the isomorphism between first-order and higher-order terms. They treat higher-order terms just like first-order terms, and there is no special rule added. Similarly, the revision rule and the choice rule work the same way on first-order and higher-order statements.

Though *implication* and *equivalence* are isomorphic to *inheritance* and *similarity*, respectively, they are not the same. The higher-order copulas indicate the substitutability between statements in *truth-value*, while the first-order copulas indicate the substitutability between terms in *meaning*. They both specify the extent to which one item *can be used as* another, though in different ways.

7.2 Implication as conditional statement

Another group of rules are introduced by the identity between an implication statement $(S_1 \Rightarrow S_2)$ and an inference process $(\{S_1\} \vdash S_2)$.

By definition, in NAL a judgment "$S \langle f,\ c\rangle$" states that "The degree of belief the system has on statement S, according to available evidence, is measured by $\langle f,\ c\rangle$". Assume that the available evidence currently used on the evaluation of S can be written as a compound statement E, then the same meaning can be represented by "$E \Rightarrow S \langle f,\ c\rangle$", that is, "The degree of belief the system has on statement 'If E is true, then S is true' is measured by $\langle f,\ c\rangle$". In this way, a statement "S" is equivalently translated into an implication statement "$E \Rightarrow S$".

This translation is a conceptual one, not an actual one, since E is not really a statement in Narsese. Even so, this conceptual translation can be used to justify certain inference rules. The implicit condition E can be added into the premises, so as to change the premise combinations into the ones for which we already have inference rules. Finally, the implicit condition is dropped from the conclusion. Table 7.3 contains several rules obtained in this way (truth-values of the premises are omitted).

premises	**add condition**	**conclusion**	**drop condition**
$M \Rightarrow P,\ M$	$M \Rightarrow P,\ E \Rightarrow M$	$E \Rightarrow P \langle F_{ded}\rangle$	$P \langle F_{ded}\rangle$
$P \Rightarrow M,\ M$	$P \Rightarrow M,\ E \Rightarrow M$	$E \Rightarrow P \langle F_{abd}\rangle$	$P \langle F_{abd}\rangle$
$M \Leftrightarrow P,\ M$	$M \Leftrightarrow P,\ E \Rightarrow M$	$E \Rightarrow P \langle F'_{ana}\rangle$	$P \langle F'_{ana}\rangle$

Table 7.3: The Conditional Syllogistic Rules (1)

Similarly, when the two premises can be seen as derived from the same evidence, the evidence can be used as the common virtual condition of the two, and some conclusions can be derived accordingly, as in Table 7.4.[1]

premises	**add condition**	**conclusion**	**drop condition**
$P,\ S$	$E \Rightarrow P,\ E \Rightarrow S$	$S \Rightarrow P\ \langle F_{ind} \rangle$	$S \Rightarrow P\ \langle F_{ind} \rangle$
$P,\ S$	$E \Rightarrow P,\ E \Rightarrow S$	$S \Leftrightarrow P\ \langle F_{com} \rangle$	$S \Leftrightarrow P\ \langle F_{com} \rangle$
$P,\ S$	$E \Rightarrow P,\ E \Rightarrow S$	$E \Rightarrow (P \wedge S)\ \langle F_{int} \rangle$	$P \wedge S\ \langle F_{int} \rangle$
$P,\ S$	$E \Rightarrow P,\ E \Rightarrow S$	$E \Rightarrow (P \vee S)\ \langle F_{uni} \rangle$	$P \vee S\ \langle F_{uni} \rangle$

Table 7.4: The Conditional Syllogistic Rules (2)

For practical purpose, the two middle-columns in the above tables of conditional rules can be ignored, and the rules can be treated as directly go from the first column (as premises) to the last column (as conclusions).

All together, NAL has three groups of syllogistic rules (deduction, abduction, and induction), one defined on inheritance statements, one on implication statements, and one on a mixture of the two, though the same truth-value functions are used.

Theorem 41 *For any statements S_1, S_2, and S_3,*

$$(S_1 \Rightarrow (S_2 \Rightarrow S_3)) \equiv ((S_1 \wedge S_2) \Rightarrow S_3)$$

that is, a conditional statement of a conditional statement is equivalent to a conditional statement with a conjunction of the conditions.

This equivalence give NAL the rules in Table 7.5.

The truth-values of the premises are omitted in the rules in Table 7.5. As before, the induction rule is applied only when the two premises are based on the same evidence. These rules can be seen as generalizations of the corresponding rules in the previous two tables by adding a condition C into J_1. Table 7.6 gives further extension of these rules by adding another condition S into J_2.

In each group of the syllogistic rules, abduction and induction can be obtained from deduction by switching a (different) premise and the conclusion, so they are "reversed deduction" in different ways.

[1] NAL does not take two arbitrary judgments as premises in an inference step. Instead, the P and S in Table 7.4 must be semantically related to each other in some way. In the current implementation, the *conjunction* statements are introduced only in NAL-6, while the *implication* and *equivalence* statements are introduced only in NAL-7. It is still unclear when the *disjunction* statements should be introduced to get non-trivial results that cannot be produced in another way.

J_1	J_2	J	F
$(C \wedge M) \Rightarrow P$	M	$C \Rightarrow P$	F_{ded}
$(C \wedge M) \Rightarrow P$	$C \Rightarrow P$	M	F_{abd}
$C \Rightarrow P$	M	$(C \wedge M) \Rightarrow P$	F_{ind}

Table 7.5: The Conditional Syllogistic Rules (3)

J_1	J_2	J	F
$(C \wedge M) \Rightarrow P$	$S \Rightarrow M$	$(C \wedge S) \Rightarrow P$	F_{ded}
$(C \wedge M) \Rightarrow P$	$(C \wedge S) \Rightarrow P$	$S \Rightarrow M$	F_{abd}
$(C \wedge S) \Rightarrow P$	$S \Rightarrow M$	$(C \wedge M) \Rightarrow P$	F_{ind}

Table 7.6: The Conditional Syllogistic Rules (4)

In NAL, *conjunction* and *disjunction* are not defined by truth table. With the help of the isomorphism and the implicit condition technique, the following theorem can be proved.

Theorem 42

$$\begin{aligned} (S_1 \wedge S_2) &\supset S_1 \\ S_1 &\supset (S_1 \vee S_2) \end{aligned}$$

7.3 Negation

Since the negation connector in NAL-5 takes one argument, it is not directly isomorphic to the (extensional/intensional) difference connectors defined in NAL-3. Instead, it is defined directly from evidence.

Definition 47 *If S is a statement, its* negation, $(\neg S)$, *is a compound statement, and its truth-value is obtained by switching the positive and negative evidence of S.*

Intuitively, the negation of a statement S can either means "It is not the case as S", or "It is the opposite case of S". In a binary logic (like IL), these two interpretations coincide, but it is not the case in a multi-valued logic. In NAL the latter interpretation is used.

The definition leads to the negation rule defined in Table 7.7.

The truth-value function is in Table 7.8.

$$\boxed{\{S\langle f_0,\, c_0\rangle\} \vdash (\neg\, S)\langle F_{neg}\rangle}$$

Table 7.7: The Negation Rule

Negation	evidence version:	$w^+ = w_0^-$ $w^- = w_0^+$
F_{neg}	truth-value version:	$f = 1 - f_0$ $c = c_0$

Table 7.8: The Truth-value Function of the Negation Rule

Theorem 43 $(\neg(\neg\, S)) \equiv S$

Theorem 44 *When the truth-values of statements S_1 and S_2 are determined independently, and they decide the truth-values of the related compound statements, then De Morgan's laws hold, that is,*

$$\neg(S_1 \wedge S_2) \equiv (\neg S_1) \vee (\neg S_2) \;\; \textit{and} \;\; \neg(S_1 \vee S_2) \equiv (\neg S_1) \wedge (\neg S_2)$$

Theorem 45

$$\begin{array}{lcl} (S_1 \wedge (\neg(S_1 \wedge S_2))) & \supset & (\neg S_2) \\ ((\neg S_1) \wedge (S_1 \vee S_2)) & \supset & S_2 \end{array}$$

Theorem 46 $(S_1 \Leftrightarrow S_2) \equiv ((\neg S_1) \Leftrightarrow (\neg S_2))$

By definition, the evidence of $(\neg(S_1 \Rightarrow S_2))$ is obtained by switching the positive and negative evidence of $(S_1 \Rightarrow S_2)$, which is the same as the evidence of $(S_1 \Rightarrow (\neg S_2))$. The same is true for the *equivalence* copula.

Theorem 47

$$\begin{array}{lcl} (\neg(S_1 \Rightarrow S_2)) & \equiv & (S_1 \Rightarrow (\neg S_2)) \\ (\neg(S_1 \Leftrightarrow S_2)) & \equiv & (S_1 \Leftrightarrow (\neg S_2)) \end{array}$$

When the truth-value of "$S_1 \Rightarrow S_2$" is determined by the induction rule in Table 7.4 from the observations of the truth-values of S_1 and S_2, an observation provides positive evidence if both S_1 and S_2 are true, negative evidence if S_1 is true and S_2 is false, and no evidence if S_1 is false. It follows that "$S_1 \Rightarrow S_2$" and "$(\neg S_2) \Rightarrow (\neg S_1)$" have the same negative evidence,

$$\boxed{\{S_1 \Rightarrow S_2\langle f_0, c_0\rangle\} \vdash (\neg S_2) \Rightarrow (\neg S_1)\langle F_{cnt}\rangle}$$

Table 7.9: The Contraposition Rule

but completely distinct positive evidence. This leads to the *contraposition rule* defined in Table 7.9.

In contraposition, though the negative evidence of the premise is taken to be negative evidence of the conclusion, they do not have the same amount, since the former is only taken as indirect evidence for the latter in NAL. This situation is similar to the situation of conversion, defined in NAL-1. The truth-value function of contraposition is given in Table 7.10.

Contraposition	evidence version:	$w^+ = 0$ $w^- = and(not(f_0), c_0)$
F_{cnt}	truth-value version:	$f = 0$ $c = \frac{(1-f_0)c_0}{(1-f_0)c_0+k}$

Table 7.10: The Truth-value Function of the Contraposition Rule

7.4 Analytical truths of IL applied in NAL

The analytical truths in IL have been introduced by the definitions and theorems, as propositions in the meta-languages of Narsese. Given the definition of the *implication* and *equivalence* copulas in IL, '$\Rightarrow$' and '$\Leftrightarrow$', in the current context they are exchangeable with the *implication* and *equivalence* connectives in propositional logic, '$\supset$' and '$\equiv$', respectively, though they are not defined in the same way.

Though binary IL truths correspond to NAL judgments with truth-value $\langle 1, 1\rangle$, such a judgment can only appear in the meta-theoretical discussions about NAL, not as a belief actually stored in the system, given AIKR. Even so, there are meta-rules that allow the IL definitions and theorems to be used in NAL inference.

Theorem 48 *An IL analytical truth S can be used as a judgment "$S\langle 1, r\rangle$" by a NAL inference rule as an implicit premise, to derive an empirical conclusion from another empirical premise. The parameter* r *is a "reliance factor" in [0, 1].*

The reliance factor is necessary, because many analytical truths are introduced to define the analytical (literal) meaning of compound terms. Though these definitions remain true in IL, in NAL the meaning of a compound term also depends on the system's empirical knowledge about it, which can be more or less from the related analytical definitions. Consequently, the analytical truths are not absolutely reliable when applied under AIKR, even though they still contribute to the meaning of the terms involved.[2]

References

[Wang, 2006, Chapter 5], [Wang, 2001a, Wang, 2004c]

[2]In the current implementation, $r = 1$ for inference rules that do not introduce new terms not in the premise, or rules where the theorems involved are equivalence statements. On the other hand, $r \leq 1$ for rules where the the theorems involved are implication statements. Furthermore, in the latter case, the implication statements are used for deduction only.

Chapter 8

NAL-6: Inference with Variable Terms

8.1 Variable terms

Definition 48 *A* query variable *is named by a word (or a number) preceded by '?', and can only appear in a question; an* independent variable *or* dependent variable *is named by a word (or a number) preceded by '#', and can appear in any type of sentence. The name of a dependent variable also contains a list of independent variables, and the list can be empty.*

Therefore, in NAL variable terms are distinguished from non-variable terms in syntax, and so are different types of variable. All the types of variables in NAL are summarized in Table 8.1.

$\langle term \rangle$	$::=$	$\langle variable \rangle$
$\langle variable \rangle$	$::=$	$\langle independent\text{-}variable \rangle$
		$\mid \langle dependent\text{-}variable \rangle$
		$\mid \langle query\text{-}variable \rangle$
$\langle independent\text{-}variable \rangle$	$::=$	$`\#'\langle word \rangle$
$\langle dependent\text{-}variable \rangle$	$::=$	$`\#'[\langle word \rangle`('\langle independent\text{-}variable \rangle^*`)']$
$\langle query\text{-}variable \rangle$	$::=$	$`?'[\langle word \rangle]$

Table 8.1: The New Grammar Rules of Narsese-6

Definition 49 *The* scope *of a variable is the smallest statement that contains all occurrences of the variable.*

In a sentence with multiple variables, each of them uses a different name, therefore its scope does not need to be explicitly specified. The scope of a variable can be embedded in that of another variable.

Definition 50 *The meaning of a variable term is determined locally by its relations with the other terms within its scope.*

On the contrary, a non-variable term is *constant*, in the sense that at any given moment, its occurrences in the whole system have the same meaning, determined by its (empirical and analytical) relations with the other term *in the whole system.* The name of a variable term is unique in a sentence, while the name of a constant term is unique in a system.

Definition 51 *For a judgment containing variable terms in it, its truth-value is defined by the truth-values of the statements obtained by replacing the variable terms by constant terms satisfying the meanings of the variables. Especially, an* independent variable *can be replaced by* any *constant satisfying the condition, and a* dependent variable *can be replaced by* a single *constant satisfying the condition. A dependent variable may depend on some independent variables when picking the constant it replaces.*

In IL and NAL, an independent variable is used to describe the property of a group of terms, typically in the extension or intension of a term; a dependent variable is used to describe the property of a unspecified term, which may depend on some independent variables. As a result, an independent variable normally appears in both sides of an implication or equivalence copula, as extension or intension of two terms. A dependent variable normally appear in two components of a conjunction, also as extension or intension of two terms. Therefore, the following are the simplest statements with variable terms:

$$\begin{array}{ll} (\#x \rightarrow S) \Rightarrow (\#x \rightarrow P) & \quad (\#x() \rightarrow S) \wedge (\#x() \rightarrow P) \\ (S \rightarrow \#x) \Rightarrow (P \rightarrow \#x) & \quad (S \rightarrow \#x()) \wedge (P \rightarrow \#x()) \end{array}$$

In this way, an independent variable is used to indicate the inclusion of the extension (or intension) of one term in that of another; a dependent variable is used to indicate the overlap of the extensions (or intensions) of two terms.

Definition 52 *A variable is* open *in a compound term if its scope goes beyond the compound, otherwise it is* closed *in the compound term. A compound term is a variable if it contains open variables.*

The IL-6 definition of query variable is an extension of the query variable implicitly introduced in IL-1 as forms of questions in "$S \rightarrow ?$" and "$? \rightarrow P$". With the new definition, there can be multiple query variables in a question, and a query variable can appear in other positions other than top-level subject or predicate. Even so, the rule of its processing remains the same, that is, all occurrences of a query variable can be substituted by the same constant term.

Both a dependent variable and a query variable can be *anonymous*, without a name, so each occurrence of it is taken to be a different term. An anonymous dependent variable does not have to appear in two components of a conjunction.

8.2 Variable elimination and introduction

Definition 53 *For given terms R, s, t, a* substitution *$R\{s/t\}$ produces a new term by replacing all occurrences of s by t in R, which is usually a compound term.*

Theorem 49 *If a true statement S contains independent variable $\#v$, then the statement $S\{\#v/t\}$ is true for any (constant or variable) term t.*

Theorem 50 *If a true statement S contains a (constant or variable) term t, and does not contain dependent variable $\#v()$, then the statement $S\{t/\#v()\}$ is true.*

Some independent-variable elimination rules are given in Table 8.2, and each of them can be seen as carrying a substitution $\{\#x/M\}$, followed by an inference defined previously. A complete list of such rules include almost all the two-premise rules with a common term, where "a common term" now is replaced by "two terms that can be instantiated by the same constant".

$\{(\#x \rightarrow S) \Rightarrow (\#x \rightarrow P),\ M \rightarrow S\}$	$\vdash$	$M \rightarrow P\ \langle F_{ded} \rangle$
$\{(\#x \rightarrow S) \Rightarrow (\#x \rightarrow P),\ M \rightarrow P\}$	$\vdash$	$M \rightarrow S\ \langle F_{abd} \rangle$
$\{(\#x \rightarrow S) \Leftrightarrow (\#x \rightarrow P),\ M \rightarrow S\}$	$\vdash$	$M \rightarrow P\ \langle F'_{ana} \rangle$

Table 8.2: Sample Independent-Variable Elimination Rules

The reverse of *independent-variable elimination* is *independent-variable introduction*, as given in Table 8.3. These rules are justified in the same way as the rules in NAL-1 and NAL-2, except that here the "extensional

$\{M \to P,\ M \to S\}$	$\vdash$	$(\#x \to S) \Rightarrow (\#x \to P)\ \langle F_{ind} \rangle$
$\{M \to P,\ M \to S\}$	$\vdash$	$(\#x \to S) \Leftrightarrow (\#x \to P)\ \langle F_{com} \rangle$

Table 8.3: Sample Independent-Variable Introduction Rules

inheritance" and "intensional inheritance" between S and P are separated, due to the using of an independent variable.

The rule in Table 8.4 introduces a dependent variable into conjunction, which can be seen as the conjunction-composition rule defined in Table 7.4 followed by a substitution $\{M/\#x()\}$.

$\{M \to P,\ M \to S\}$	$\vdash$	$(\#x() \to P) \wedge (\#x() \to S)\ \langle F_{int} \rangle$

Table 8.4: Sample Dependent-Variable Introduction Rule

The reverse of the rule in Table 8.4 can be seen as a special type of unification to match a dependent variable with a constant, as given in Table 8.5. Conceptually, the inference is a comparison followed by an analogy. First, in "$(\#x() \to P) \wedge (\#x() \to S)$" the anonymous term provides evidence for a similarity statement "$P \leftrightarrow S$", then the latter is used with "$M \to S$" by the analogy rule to derive "$M \to P$". Therefore, the truth-value function F^w_{ana} is just the analogy function F_{ana}, except that the confidence of the second premise is taken to be the weight of confidence of the corresponding similarity statement.

$\{M \to S,\ (\#x() \to P) \wedge (\#x() \to S)\}$	$\vdash$	$M \to P\ \langle F^w_{ana} \rangle$

Table 8.5: Sample Dependent-Variable Elimination Rule

The rules in Table 8.4 and Table 8.5 are only about the extensions of S and P. Similarly, there are rules that only process the intensions of the terms involved. As required before, in NAL a dependent variable is only introduced into a conjunction, and an independent variable into both sides of an implication or equivalence.

Variables can be introduced into statements where other variables exist. When an independent variable is introduced, the existing dependent variables become its function. The rules for multiple variables in Table 8.6 can be extended to handle more than two variables.

$\{(\#x \to P) \Rightarrow (M \to (\perp\ R\ \#x \diamond)),\ M \to S\}$ $\vdash ((\#y \to S) \wedge (\#x \to P)) \Rightarrow (\#y \to (\perp\ R\ \#x \diamond))\ \langle F_{ind} \rangle$
$\{(\#x \to P) \Rightarrow (M \to (\perp\ R\ \#x \diamond)),\ M \to S\}$ $\vdash (\#y() \to S) \wedge ((\#x \to P) \Rightarrow (\#y() \to (\perp\ R\ \#x \diamond)))\ \langle F_{int} \rangle$
$\{(\#x() \to P) \wedge (M \to (\perp\ R\ \#x() \diamond)),\ M \to S\}$ $\vdash ((\#y \to S) \Rightarrow ((\#x(\#y) \to P) \wedge (\#y \to (\perp\ R\ \#x(\#y) \diamond)))\ \langle F_{ind} \rangle$
$\{(\#x() \to P) \wedge (M \to (\perp\ R\ \#x() \diamond)),\ M \to S\}$ $\vdash (\#y() \to S) \wedge (\#x() \to P) \wedge (\#y() \to (\perp\ R\ \#x() \diamond))\ \langle F_{int} \rangle$

Table 8.6: Sample Multi-Variable Introduction Rules

The revision rule is also extended to unify independent variables. For example, statements $(\#x \to S) \Rightarrow (\#x \to P)$ and $(\#y \to S) \Rightarrow (\#y \to P)$ can be merged together. On the other hand, this rule cannot be applied on two judgments containing $((\#x() \to S) \wedge (\#x() \to P))$, since the dependent variables in them do not necessarily correspond to the same (constant) term, even though they share the same name.

References

[Wang, 2006, Chapter 5]

Chapter 9

NAL-7: Temporal Inference

NAL-7 introduces *time* into the logic.

9.1 Time and events

In an implementation of NAL, call it NARS, time appears as a total order among the system's internal and external activities and events.

Definition 54 Time *within NARS can be measured by an internal clock, with the unit being certain recurrent activity in the system, such as its inference cycle.*

Such a time measurement is relative to the system's internal activity, and independent of the hardware/software speed of the implementation, so different copies of NARS may have different "subjective time" associated to an activity in their common environment.

Definition 55 *The real-time* experience *of a NARS is a sequence of Narsese sentences, separated by non-negative numbers indicating the interval between the arriving time of subsequent sentences.*

Therefore, each input sentence can be associated with a moment in the system clock, and the experience of the system can be represented as a stream of Narsese sentences, with the time interval between adjacent sentences marked according to the system clock.

So far, there have been three notions of *experience* used in NAL:

- In IL, *idealized* experience is defined as a *set* of (true) statements, with the Closed-World Assumption. The order of sentences does not matter, and is ignored by the logic.

- In NAL-1 to NAL-6, *actual* experience is defined as a *stream* of sentences of the corresponding Narsese (i.e., Narsese-1 to Narsese-6), without the Closed-World Assumption. The timing in the stream is omitted in the language, and ignored by the inference rules (though it matters for the inference control mechanism).

- Since NAL-7, *real-time* experience explicitly indicates time in the input stream, using the internal clock. It covers the previous notions of experience as special cases: actual experience corresponds to the situation where there is one input per moment, and idealized experience is where all inputs arrive at the very beginning.

In NARS, the meaning of a term and the truth-value of a statement can be *produced* from many different real-time or actual experiences expressed in Narsese, though they are *defined* by an idealized experience consisting of IL statements.

Definition 56 *An* event *is a statement with a time-dependent truth-value, that is, the evidential support summarized in its truth-value is valid only for a certain period of time.*

Accurately speaking, almost all empirical statements are time dependent, and few statements are about relations holding forever. However, for practical purposes, it is not always necessary for a system to take the temporal attributes of a statement into consideration. Therefore, whether a *statement* should be treated as an *event* may change from context to context, and events are just statements whose temporal attributes are specified. On the contrary, the time interval of a "non-event" statement is unspecified, except that it includes the current moment, as well as all the moments of relevance.

If the time interval of a truth-value is referred to as when the corresponding event "happens", then between two events E_1 and E_2, their basic temporal relation can be one of the following three cases:

- E_1 happens before E_2 happens,

- E_1 happens after E_2 happens,

- E_1 happens when E_2 happens.

Obviously, "before" and "after" are the opposite directions of the same temporal relation.

Definition 57 *There are two basic temporal relations between two events: "before" (which is irreflexive, antisymmetric, and transitive) and "when" (which is reflexive, symmetric, and transitive).*

If the temporal relation between two events is more complicated than these cases (for example, if their time intervals overlap partially), it is always possible to divide an event into subevents (such as talking about "when E_1 starts" and "when E_2 ends"), then describe their temporal relations in detail.

If event E_1 is represented as "before" event E_2, the time interval between "E_1 finishes" and "E_2 starts" is omitted as negligible, even if the duration of this interval is not zero. When the interval is not negligible, it should be represented as an event E_3, which happens after E_1 and before E_2.

Similarly, when two events are described as happening at the same time, it does not mean that their time intervals perfectly overlap, but that their difference in timing is negligible. If an absolute time, such as the system clock, is used to represent the temporal property of an event, then a moment in that time dimension can be treated as a special event, and these two events are described as happening at the same time. In this sense, the system clock corresponds to a sequence of events, each of which having the same duration.

By using a relational description for temporal attributes of event, NAL can be applied to fields where phrases like "at the same time" and "immediately after" are used with very different scales, scopes, and accuracies. This treatment is consistent with the general semantic principle of Narsese, that is, the language is not used *to represent the world as it is*, but *to summarize the experience as the system needs*.

The above representation of temporal information only supports some basic types of temporal inference, and more complicated and accurate types of temporal inference can be carried out by explicitly specifying temporal relations as terms, and their relationships as implication and equivalence statements.

9.2 Temporal operators and copulas

Since in NAL temporal attributes is optional in statements, the two temporal relations are never used alone, without any logical relations between the events. Instead, they are used in combination with certain copulas and term connectors that have been introduced before.

First, "E_1 happens before E_2 happens" and "E_1 happens when E_2 happens" both assume "E_1 and E_2 happen (at some time)", which is "$E_1 \wedge E_2$" plus temporal information.

Definition 58 *The conjunction connector has two temporal variants: "sequential conjunction" (",") and "parallel conjunction" (";"). "(E_1, E_2)" corresponds to compound event "E_1 then E_2", and "$(E_1; E_2)$" corresponds to compound event "E_1 and E_2".*

Like ordinary conjunction, either of the two temporal operators can take more than two components, and is associative.

Similarly, there are temporal variants of copulas *implication* and *equivalence*.

Definition 59 *For an implication statement "$S \Rightarrow T$" between events S and T, three different temporal relations can be distinguished:*

1. *If S happens* before *T happens, the statement is called "predictive implication," and is rewritten as "$S \mathbin{/\!\!\Rightarrow} T$", where S is called a* sufficient precondition *of T, and T a* necessary postcondition *of S.*

2. *If S happens* after *T happens, the statement is called "retrospective implication," and is rewritten as "$S \mathbin{\backslash\!\!\Rightarrow} T$", where S is called a* sufficient postcondition *of T, and T a* necessary precondition *of S.*

3. *If S happens* when *T happens, the statement is called "concurrent implication," and is rewritten as "$S \mathbin{|\!\!\Rightarrow} T$", where S is called a* sufficient co-condition *of T, and T a* necessary co-condition *of S.*

Definition 60 *Three "temporal equivalence" (predictive, retrospective, and concurrent) relations can be defined.*

1. *"$S \mathbin{/\!\!\Leftrightarrow} T$" (or equivalently, "$T \mathbin{\backslash\!\!\Leftrightarrow} S$") means that S is an* equivalent precondition *of T, and T an* equivalent postcondition *of S.*

2. *"$S \mathbin{|\!\!\Leftrightarrow} T$" means that S and T are* equivalent co-conditions *of each other.*

3. *To simplify the language, "$T \mathbin{\backslash\!\!\Leftrightarrow} S$" is always represented as "$S \mathbin{/\!\!\Leftrightarrow} T$", so the copula "$\backslash\!\!\Leftrightarrow$" is not actually included in the grammar of Narsese.*

As explained in NAL-5, judgment "$S\langle f,\ c\rangle$" can be equivalently rewritten as "$E \Rightarrow S\langle f,\ c\rangle$", where E is a virtual compound statement summarizing the currently available evidence. Now if statement S is an event, its temporal attribute can be specified relative to E, taking as an event that is currently occurring. Since in Narsese E is implicitly assumed, the temporal implication operators serve here as *tense*, which are operators indicating the temporal nature of truth-values. In this way, adjectives like "past," "present," and "future" can be represented in Narsese.

Definition 61 *The* tense *of a sentence indicates the occurring time of an event with respect to "events happening now", a special event. The temporal implication symbols '\⇒', '|⇒', and '/⇒' are also used in a sentence to indicate "past tense", "present tense", and "future tense", respectively.*

What makes the situation complicated is that tenses are always used with respect to the using time of a sentence, and in a real-time system "now" changes constantly, so "future" gradually becomes "present", then "past". Furthermore, while "present" is unique, the moments referred to as "past" and "future" are not. The same judgment may have different truth-values while having the same "past" or "future" tense, and it may not be considered as conflicting evidence, because each of them is actually about a different moment.

Because of the above reasons, the above qualitative "tense" is not used in the internal representation of a belief. Instead, when a belief in NARS corresponds to an event, its truth-value is associated with its *happening time* represented by the system clock. Additionally, the creation time of a sentence, either from outside (input) or inside (derivation) is recorded, too. When a sentence is expressed in Narsese for communication, this temporal information is translated into (and from) a tense (which has three possible values), with respect to the current time when the communication happens. The clock values are not directly used in communication, because of their system-specific nature.

The new statements introduced in NAL-7 are summarized in Table 9.1.

$\langle judgment \rangle$	::=	$\langle statement \rangle [\langle tense \rangle] \langle truth\text{-}value \rangle$
$\langle question \rangle$	::=	$\langle statement \rangle [\langle tense \rangle]$
$\langle statement \rangle$	::=	'(,' $\langle statement \rangle \langle statement \rangle^{+}$')'
		| '(;' $\langle statement \rangle \langle statement \rangle^{+}$')'
$\langle tense \rangle$	::=	'/⇒' | '\⇒' | '|⇒'
$\langle copula \rangle$	::=	'/⇒' | '\⇒' | '|⇒' | '/⇔' | '|⇔'

Table 9.1: The New Grammar Rules of Narsese-7

In summary, temporal information is represented in NARS in multiple ways:

Absolute representation. Some sentences have "tense" associated, to indicate the time-dependency of their truth-values. In the communication interface, tense is represented qualitatively, with respect to the current time; within the system, this information is represented by clock values, indicating the moment for which the truth-values are established.

Relative representation. Some compound terms (implication, equivalence, and conjunction) may have temporal order specified among its components (each taken to be an event).

Explicit representation. When the above representation cannot satisfy the accuracy requirement when temporal information is needed, it is always possible to introduce terms to explicitly represent an event, as well as its beginning, ending, and duration.

9.3 Temporal inference

Different types of temporal information are treated in different ways in NARS.

In the reasoning process of NARS, the terms that **explicitly** represent temporal information are treated just like other term, and temporal relations are handled like other relations. There is no special inference rule needed.

When temporal information is represented **relatively**, the existing rules are modified by taking the temporal properties in the premises and conclusions into consideration while processing their local properties.

The temporal orders within copula and operators are handled by the inference rules based on the properties of the two basic temporal relations. Consequently, some of the inference rules in NAL-7 are variants of the rules defined in NAL-5 and NAL-6. In them, the only additional function of these rules is to decide the temporal property of the conclusions according to that of the premises, and the truth-value functions remain the same. For example, the following is a deduction rule introduced in NAL-5,

$$\{(C \wedge M) \Rightarrow P,\ S \Rightarrow M\} \vdash (C \wedge S) \Rightarrow P \langle F_{ded} \rangle$$

Now it has a variant in NAL-7, as listed in Table 9.2. Since the logical factor and the temporal factor are independent of each other in the rules, these variant rules can be obtained by considering the two factors separately, then combining them in the conclusion. An alternative way is to see the above rules as the combinations of the NAL rules introduced previously and a

$$\boxed{\{(M,\ C) \not\Rightarrow P,\ S \not\Rightarrow M\} \vdash (S,\ C) \not\Rightarrow P\ \langle F_{ded}\rangle}$$

Table 9.2: Sample Temporal Inference Rule

"meta-rule" of temporal inference, based on the properties of the two basic temporal relations.

When temporal information is represented **absolutely** as clock values, there are three basic cases:

- When the premises are about the same moment t, the conclusion is about the same moment.
- When one premise is about moment t, and the other one is timeless, the conclusion is about moment t.
- When the premises are about the different moments t_1 and t_2, one of them need to be "casted" into another.

In a tensed sentence, the truth-value is about a given moment, indicated by a clock value. When there is no other information, this truth-value can also be used for nearby moments. When a truth-value about moment t_1 is used for moment t_2, its confidence is decreased by multiplying a "discount factor" d:

$$d = 1 - \frac{|t_1 - t_2|}{|t - t_1| + |t - t_2|}$$

where t is the moment when the judgment is made, and when $t = t_1 = t_2$, $d = 1$.

A tensed sentence can also be cased into a timeless sentence, as a form of "temporal induction". Since a sentence at a certain moment provides for the sentence in all moments, the frequency of the conclusion is the same as that of the premise, and the confidence of the conclusion, c, is determined by that of the premise, c_0, as $c = c_0/(c_0 + k)$, where k is the evidential horizon defined in NAL-1. In other words, the confidence value is multiplied to a "discount factor" $1/(c_0 + k)$.

Both absolute and relative temporal information are present in rules like

$$\{S\ ,\ S \not\Rightarrow P\} \vdash P\ \langle F_{ded}\rangle$$

where if the two premises are about moment t, the conclusion is about moment $t + 1$.

Another group of NAL-7 rules are variants of the following inference rules defined in NAL-5:

$$\{P,\ S\} \vdash S \Rightarrow P\ \langle F_{ind} \rangle$$

$$\{P,\ S\} \vdash S \Leftrightarrow P\ \langle F_{com} \rangle$$

Though these rules do not apply to arbitrary P and S, they are applicable when the two are temporally related events. When P and S are events happening at the same time, the conclusions are "$S\ |\!\!\Rightarrow\ P\ \langle F_{ind} \rangle$" and "$S\ |\!\!\Leftrightarrow\ P\ \langle F_{com} \rangle$"; when S happens right before P, the conclusions are "$S\ /\!\!\!\Rightarrow P\ \langle F_{ind} \rangle$" and "$S\ /\!\!\!\Leftrightarrow P\ \langle F_{com} \rangle$". Here the situation is different from the above temporal meta-rule in that without a temporal relation, these rules will not be applied.

References

[Wang, 2006, Chapter 5], [Wang, 2004c, Wang, 2007b]

Chapter 10

NAL-8: Procedural Inference

NAL-8 interprets certain events as operations of the system itself, and uses them to achieve goals.

10.1 Operations and goals

Definition 62 *An* operation *of a system is an event that the system can actualize. In Narsese, an operation is represented as an* operator *(a special term whose name starts with '$\Uparrow$') followed by an* argument list *(a sequence of terms), which can be empty. Within the system, operation "($\Uparrow op\ a_1 \cdots a_n$)" is treated as statement "($\times\ a_1 \cdots a_n) \rightarrow op$", where op belongs to a special type of term, which has a procedural interpretation.*

Therefore operation is system dependent: the *operations* of a system will be observed as *events* by other systems. An *operator* is a system-specific term connector. For a system implementing NAL-8, its list of operators remains constant, though not specified in Narsese.

While statements are *declarative* knowledge and events are *episodic* knowledge, operations are *procedural* knowledge, in the sense that the meaning of an operation is not only revealed by how it is related to the other terms in Narsese (according to the system's experience), but also by what it *does* to the "body" of the system, as well as to the environment.

An operation usually distinguishes input and output among its arguments. When an operation is described abstractly, its input arguments are typically independent variables, and its output are dependent variables.

Such an operation corresponds to a function that maps certain input values into output values. Optionally, an operation may bring the system some Narsese sentences as feedback.

Since the main purpose of operations is for the system to achieve various consequences, their meaning, or the system's beliefs on them, is usually represented as (temporal or not) *implication* or *equivalence* statements, which indicate the conditions, causes, and effects of an operation. Typically, it takes the following form:

$$(condition, operation) \mathbin{/\!\!\!\!\Rightarrow} consequence$$

where *condition* and *consequence* are both events. This form is common, because it is a simplified version of

$$condition \mathbin{/\!\!\!\!\Rightarrow} (operation \mathbin{/\!\!\!\!\Rightarrow} consequence)$$

so the *condition* is not really applied on the *operation*, but on its relation with the *consequence*.

For an operation to be meaningful and useful for the system, it will have some consequence that is eventually *observable*, that is, trigger certain input judgments, as the feedback of the operation, in the system's experience.

As other statements, the truth-value of the above statement indicates the evidential support for the stated relationship. The system usually has multiple such statements for each operation. Under AIKR, in NAL the conditions and consequences of an operation are never exhaustively specified in each belief about it. Instead, each belief only records its (limited) experience on the relation between the operation and the *stated* events.

Compound operations work like (object-level) programs, which organize primitive operations into hierarchical control structures. The basic control structures include

Sequential execution, formed by the *sequential conjunction* operator on operations;

Parallel execution, formed by the *parallel conjunction* operator on operations;

Conditional execution, formed by the *implication* (or *equivalence*) copula between events and operations;

Repeated execution, formed recursively by conditional execution.

These control structures give Narsese the capability of a general-purpose programming language. Furthermore, the *equivalence* copula can be used to give a compound operation a simple name.

Operations can make changes both within a system and in its outside environment, with consequences expressible as Narsese statements. However, not all activities in the system can be perceived and controlled in NAL in this way.

Definition 63 *A* goal *is a sentence containing an event the system is attempting to realize by carrying out operations.*

Given the inevitable uncertainty in the event, to "realize it" actually means "to make it as close to absolute truth as possible."

NARS usually has multiple goals, and they may conflict with one another, in the sense that the achieving of a goal makes another one harder to be achieved. Therefore, the system must make decisions about whether to pursue various goals or whether to take various operations.

Definition 64 *The* desire-value *of an event measures the extent to which a desired state is implied by the event, that is, the desire-value of event E is the truth-value of the implication statement $E \Rightarrow D$, where D is a virtual statement describing the desired state of the system, a summary of its current goals.*

Here D is "virtual", in the sense that it is not a concrete statement in Narsese, but a conceptual one in the meta-language, used in the design of the system. By it, the derived-values of the events involved are reduced to truth-values, whose calculations have been specified by the truth-value functions. Here is the situation is like in NAL-5 where a "virtual evidence" is introduced so that the truth-value of a statement can be taken as the truth-value for the statement to be implied by the available evidence. In both situations, the evaluation of a statement is interpreted as an evaluation of it and another (virtual) statement, which is coherent with the semantic principle of NARS that the meaning of an item is revealed by its relations with other items, rather than being an intrinsic property of the item itself. Intuitively speaking, the truth-value of a statement evaluates its relation with the "source" (where it comes from), while the desire-value is about the "destination" (where it leads to).

A desire-value is attached to every statement in the system, because it may become a goal in the future, if it is not already a goal. This value shows the system's "attitude" about the situation in which the statement is true.[1] The desire-value of a goal is always explicitly expressed, though the

[1] This desire value will eventually be attached to every term, to represent the system's "feeling" about it. If the term is not a statement, its desire value will be determined by the beliefs in which it appears.

desire-values of other statements are often omitted unless they are relevant to a discussion.

Now the questions in NAL can not only be about the truth-value of a statement, but also about its desire-value. To more clearly separate different types of sentences, in Narsese-8 a punctuation mark is added at the end of each sentence: '.' for judgment, '!' for goal, '?' for question (on truth-value), and '@' for quest, that is, question on desire-value. The new grammar rules introduced in NAL-8 are summarized in Table 10.1.

$\langle sentence \rangle$	::=	$\langle judgment \rangle \mid \langle goal \rangle \mid \langle question \rangle$
$\langle judgment \rangle$	::=	$\langle statement \rangle$'.' $[\langle tense \rangle]\langle truth\text{-}value \rangle$
$\langle goal \rangle$	::=	$\langle statement \rangle$'!' $\langle desire\text{-}value \rangle$
$\langle question \rangle$	::=	$\langle statement \rangle$'?' $[\langle tense \rangle]$
		$\mid \langle statement \rangle$'@' $[\langle tense \rangle]$
$\langle statement \rangle$	::=	'($\Uparrow$' $\langle word \rangle\ \langle term \rangle^*$ ')'
$\langle desire\text{-}value \rangle$	::=	$\langle truth\text{-}value \rangle$

Table 10.1: The New Grammar Rules of Narsese-8

10.2 Inference on operations and goals

Since operations and goals are events, the previously defined inference rules on events work on them, too.

Inference on an operation can derive new beliefs about its preconditions and postconditions. Furthermore, compound operations are selectively formed from useful combinations of operations, and become "skills" of the system that can be executed efficiently, without step-by-step deliberation.

Inference on a goal also derives new beliefs about how it can be realized, as well as reveals its by-products and side-effects. Especially, for a given goal G, the inference engine can find a *plan*, which is a compound operation Op that achieves the goal (i.e., to have a high expectation value for "$Op \Rightarrow G$"). By executing the plan, and adjusting it when necessary, the internal or external environment is changed to turn the goal into reality. When repeatedly appearing compounds of operations are memorized, repeated planning is avoided, and the system learns a new skill.

When a goal is an operation, it can be directly realized by executing the operator on the arguments. If a goal cannot be directly satisfied in this way, by backward inference it can increase the desire-values of certain events. For a given event, the desire-values coming from different goals are merged together using the revision rule, just like how truth-values from different evidential bases are merged.

The *decision-making* rule will turn candidate goals with high desire-value and plausibility into goals being actually pursued by the system.

Definition 65 *The plausibility of goal G is the truth-value of implication statement "$\# \Rightarrow G$", that is, "there is a way to achieve G."*

The Decision-making Rule A candidate goal G is actually pursued by the system, when its expected desirability p_G and expected plausibility d_G satisfy condition $p_G(d_G - 0.5) + 0.5 > t$, where t is a threshold larger than 0.5.

The above "decision-making function" has the same form as the expectation function, with desirability as frequency and plausibility as confidence.

If a goal G has been decided to be actively pursued, the system will also derive a question with the same content to check if the desired event has already happened. If that turns out to be the case, the goal will be directly satisfied by a judgment, and therefore its desire-value will be greatly reduced.

10.3 Sensorimotor interface

As a reasoning system, NARS communicates its environments in Narsese, a formally defined language.

On the top of that, NAL-8 introduces an interface between NARS and an external system, a tool, or a "body", by allowing an out-going *command* to be represented and processed as a NARS *operation*. Here the only requirement is that the command can be put into the form of "$(\Uparrow op \ \ a_1 \cdots a_n)$", with all the arguments represented as terms in NARS.

In this way, NARS, as a general-purpose "mind", can be embedded within, or connected with, various host systems with different sensorimotor mechanisms, either in a physical world or in a virtual world (which also exists in a physical world, though is described abstractly). For a given host, a special interface module needs to be built, which registers all the relevant commands in the host that is exposed to the control of NARS, so that whenever NARS decides to execute an operation, the corresponding command is sent to the corresponding actuator in the host system.

Similarly, the sensors in the host are also formalized as operators, invoked by Narsese questions, and the result of the operations will be received as new experience (input knowledge) to the system. Driving by questions derived both from goals and from other questions, the system's observation is not a merely passive process which accepts whatever comes from the environment, but an active process directed by the system's goal-achieving activities.

NARS leaves the low-level sensorimotor management to the host system, which still contribute to the perception and action processes, by allowing operations defined on multiple levels of abstraction (with different granularity and scope), as well as using anticipations and goals to selectively process incoming information. With a sensorimotor mechanism connected to NARS, the effect of an operation can be anticipated, checked, and confirmed, and the feedback will provide information for various types of learning.

Though the integrated system (NARS plus host) as a whole can have experience with multiple modalities, the NARS part of the system remains amodal in design. On the other hand, the *content* of the system's beliefs and concepts will depend on its "body".

10.4 Self-monitoring and self-control

The sensorimotor mechanism described above can be expanded into the system itself. It means that a NARS implementation can be equipped with sensors and actuators that perceive and modify the internal state of the system itself. These sensors and actuators are invoked by commands issued in NARS, and their results are feedback to the system, represented as Narsese sentences.

Consequently, such a NARS has both an "external experience" and an "internal experience", and the two are represented and processed in similar ways. Like its knowledge of the world, the system's knowledge of itself is also a summary of its experience, and restricted by its sensorimotor and information-processing capability. There is no new grammar and inference rules needed, but system-specific operations.

From the view power of NARS, the sensors and actuators can be roughly divided into two types, those that are mostly about its "body" and those that are mostly about its "mind". When NARS is implemented in a robot, there will be various sensors to monitor its energy level, damage of parts, etc., which do not change how the reasoning/learning process, but provide goals to be achieved and means to achieve them. Though these sensors work on the body of the system, they are not that different from the sensors that work on the outside environment. On the other hand, there are also sensors on the reasoning/learning process, which express information about the state of the system in a format (Narsese sentences) that can be processed by the system. These sensors are very different from the "ordinary" ones, since they directly produce conceptual level results, without another categorization process. Furthermore, their results can be self-referential. Similarly, there are "physical" actuators and "mental" actuators. The the latter are inevitably carried out by physical processes, they are known to the system only at an abstract level, without their physical details.

Even though sensorimotor mechanism is system-specific, and optional to NARS, we can still expect a small common cognitive capabilities needed in most intelligent systems.

- There should be sensors to measure certain indicators of system's overall status, such as how busy it is and how much its current goals have been achieved. This kind of information will be used by the control mechanism to adjust resource allocation, among other things.
- There should be sensors and actuators to explicitly detect and adjust the inference process, by "paying attention" to certain concepts and sentences.
- There should be sensors to report certain properties of specific data items. For example, the system may want to explicitly consider and change the truth-value or desire-value of a statement.
- There should be sensors to remember the concept-level activities of the system.

Before such a self-control mechanism is implemented, the inference control in NARS is pure *autonomic*. In each inference step, the task to be carried out and the belief to be used are selected according to several factors to achieve the highest overall efficiency, and this process is governed by algorithms that coded in the programming language of the system, and are beyond the reach of the inference rules. With the above self-control mechanism, however, the system can think about its own thinking process, and adjust it as allowed by its internal sensorimotor mechanism, according to its experience. This introduces *voluntary* control (according to knowledge represented declaratively in Narsese) that supplement (though not replace) the autonomic control (according to knowledge represented procedurally in the programming language of NARS). In the future, NARS can be implemented in systems where the resources to be managed is not limited to processing time and storage space of information. For example, a robot should manage its own energy usage. This kind of task can also be carried out by special-purpose operations.

Since the sensorimotor mechanism is only directly accessible to the system, and its effect cannot be fully expressed and duplicated via communication by other systems. Consequently, NARS will have *consciousness*, that is, subjective experience that can only be partially communicated to and understood by other systems. Even to the system itself, since its "inside-oriented" and "outside-oriented" operations are separated from each other, and there is no one-to-one mapping between the two, two separate concept systems will be developed to describe its internal and external processes, and there will be a "mind-body" gap in between.

References

[Wang, 2006, Chapter 5], [Wang, 2004c, Wang, 2007b]

Chapter 11

Summary

11.1 Narsese grammar and semantics

The complete grammar rules of Narsese are listed in Table 11.1.
Additional notes about the Narsese grammar:

- Confidence values 0 and 1 are used in the meta-language of Narsese only, and cannot appear in actual sentences in the system.
- In the communication between the system and its environment, a truth-value can be replaced by amounts of evidence or frequency interval.
- In the communication between the system and its environment, copulas "$\circ\!\!\rightarrow$", "$\rightarrow\!\!\circ$", and "$\circ\!\!\rightarrow\!\!\circ$" are also valid.
- Most prefix operators in compound term and compound statement can also be used in the infix form.

The symbols used in Narsese grammar are listed in Table 11.2.

11.2 NAL Inference Rules

The inference rules of NAL are summarized into several categories, according to their syntactic features.

(A) Two-premise inference rules: each of these rules takes two premises J_1 and J_2, and derive a conclusion J, with a truth-value calculated from the truth-values of the premises by a function F.

(A.1) First-order syllogistic rules, in Table 11.3, are defined on copulas *inheritance* and *similarity.*

(A.2) Higher-order syllogistic rules, in Table 11.4, are defined on copulas *implication* and *equivalence.*

(A.3) Conditional syllogistic rules, in Table 11.5, are based on the nature of conditional statements.

(A.4) Composition rules, in Table 11.6, introduce new compounds into the conclusion.

(A.5) Decomposition rules, in Table 11.7, are the opposite operation of the composition rules. Each decomposition rule comes from a high-level theorem of the form $(st_1 \wedge st_2) \supset st_3$, where st_1 is a statement about a compound, st_2 is a statement about a component of the compound, while st_3 is the statement about the other component. As a two-premise inference rule, in the first step the truth-values of st_1 and st_2 are used to calculate the truth-value of $(st_1 \wedge st_2)$ (using F_{int}), then the resulting truth-value is used by an Implication Rule (to be defined in the following) to decide the truth-value of st_3.

(B) One-premise inference rules: each of these rules takes one premise J_0, and derive a conclusion J, with a truth-value calculated from the truth-value of the premise by a function F.

(B.1) Conversion rules, in Table 11.8, are rules only need to consider the evidence provided by the premise.

(B.2) Equivalence rules, in Table 11.9, come from theorems of the form "$statement_1 \equiv statement_2$". Each of them can be used in inference as equivalence statement "$statement_1 \Leftrightarrow statement_2 \langle 1, r \rangle$".

(B.3) Term reduction rules, in Table 11.10, come from theorems of the form "$term_1 \leftrightarrow term_2$". Each of them can be used in inference to reduce term $term_1$ into a simpler term $term_2$, and turns a premise into a conclusion with the same truth-value.

(B.4) Implication rules, in Table 11.11, come from theorems in the form of "$statement_1 \supset statement_2$". Each of them can be used in inference as implication statement "$statement_1 \Rightarrow statement_2 \langle 1, r \rangle$".

(B.5) Inheritance rules, in Table 11.12, come from theorems in the form of "$term_1 \rightarrow term_2$". Each of them can be used as two implications "$(X \rightarrow term_1) \supset (X \rightarrow term_2)$" and "$(term_2 \rightarrow X) \supset (term_1 \rightarrow X)$", by the above Implication Rules.

(C) Meta-level inference rules: Each of these rules specifies how to use the other rules defined above for additional usages.[1]

(C.1) Question derivation. A question Q and a judgment J produce a derived question Q', if and only if the answer to Q', call it J', can be used with J to derive an answer to Q by a two-premise inference rule; a question Q by itself produces a derived question Q', if and only if the answer to Q', call it J', can be used to derive an answer to Q by a one-premise inference rule.

(C.2) Goal derivation. A goal G and a judgment J produce a derived goal G', if and only if the solution to G', call it J', can be used with J to derive a solution to G by a two-premise inference rule; a question G by itself produces a derived goal G', if and only if the solution to G', call it J', can be used to derive a solution to G by a one-premise inference rule. In both cases, the desire-value of G' is derived as the truth-value of $G' \Rightarrow D$ from the desire-value of G, as the truth-value of $G \Rightarrow D$, as well as the truth-value of J (if it is involved).

(C.3) Variable substitution. All occurrences of an independent variable term in a statement can be substituted by another term (constant or variable); all occurrences of constant in a statement can be substituted by a dependent variable term (constant or variable). The reverse cases of substitution are limited to Table 8.3 and 8.5. A query variable in a question can be substituted by a constant term in a judgment.

(C.4) Temporal attributes. Temporal inference is carried out by processing the logical factor and the temporal factor in the premises in parallel. The former is based on the inference rules, and the latter on the properties of the two basic temporal relations. When both factors can be decided, they are combined in the conclusion, otherwise no conclusion is derived.

(D) Direct-processing rules: Each of these rules directly processes a new inference task, based on the information local to the content of the task.

---

[1]Beside the following meta-rules, it may be possible to summarize some other rules into meta-rules. For instance, the rules in 7.5, 7.6, and 8.6 probably should be replaced by the following meta-rules:

$$\text{If } \{P_1, P_2\} \vdash C\langle F_n\rangle, \text{ then } \{(A \Rightarrow P_1), P_2\} \vdash (A \Rightarrow C)\langle F_n\rangle$$

$$\text{If } \{P_1, P_2\} \vdash C\langle F_n\rangle, \text{ then } \{(A \wedge P_1), P_2\} \vdash (A \wedge C)\langle F_n\rangle$$

(D.1) Revision/update. When the system gets a new judgment (or goal), it is used with an existing judgment (or goal) by the *revision* rule, under the conditions that (1) the two have the same content (top-level statement), (2) the content does not contain dependent variable term, and (3) the two have distinct evidential bases. The conclusion has the same content, but a higher confidence value. When the statement is an event, and the new judgment is significantly different from the old one, the operation is *update*, and the old belief is adding a past tense, and the new one becomes the current belief.[2]

(D.2) Choice. A judgment provides an answer to a question, and a solution to a goal, with the same content. When there are multiple candidate answers or solutions, the one with high *expectation* and low *complexity* is chosen if the question contains query variables,[3] otherwise the one with the highest *confidence* value is chosen.

(D.3) Decision. A candidate goal G is actually pursued by the system, when its expected desirability p_G and expected plausibility d_G satisfy condition $p_G(d_G - 1/2) + 1/2 > t$, where t is a threshold larger than 1/2. When the goal is an operation, it is executed.

11.3 NAL Truth-value Functions

All truth-value functions are summarized in Table 11.13, in their simplest form. Different types of uncertainty measurements are mixed in the functions, and their relations are given in Table 3.1.

The functions are clustered into groups, according to the syntactic feature of the rules using them. The functions used in the syllogistic rules are divided into *strong* functions and *weak* functions. In a rule using a strong function, the confidence of the conclusion has an upper bound 1, and the rule remains valid in its binary form; in a rule using a weak function, the confidence of the conclusion has an upper bound $1/(1+k)$ (since w has an upper bound 1), and the rule is invalid in its binary form. A NAL inference rule with a strong truth-value function will be a valid inference rule in IL if when the truth-values are omitted (so the premises and conclusion become binary), which is not the case for the rules with weak truth-value functions.

[2]There are situations where *revision* and *update* are both applicable. The system will either to both or use additional information to select between the two interpretations of the situation.

[3]When resource restriction is taken into consideration, the syntactic *complexity* of the candidates should also be taken into account, together with the expectation value of the candidate, and simpler answers should be preferred. In the current implementation, the choice rule compare two candidates by their *expectation/complexity* ratio.

References

[Wang, 2006], [Wang, 1995a, Wang, 2007a]

$\langle \textit{sentence} \rangle$	$::=$	$\langle \textit{judgment} \rangle \mid \langle \textit{question} \rangle \mid \langle \textit{goal} \rangle$
$\langle \textit{judgment} \rangle$	$::=$	$\langle \textit{statement} \rangle$ ‘.’ $[\langle \textit{tense} \rangle] \langle \textit{truth-value} \rangle$
$\langle \textit{question} \rangle$	$::=$	$\langle \textit{statement} \rangle$ ‘?’ $[\langle \textit{tense} \rangle]$
$\langle \textit{goal} \rangle$	$::=$	$\langle \textit{statement} \rangle$ ‘!’ $\langle \textit{desire-value} \rangle$
$\langle \textit{statement} \rangle$	$::=$	‘(’ $\langle \textit{term} \rangle \langle \textit{copula} \rangle \langle \textit{term} \rangle$ ‘)’ $\mid \langle \textit{term} \rangle$
		$\mid$ ‘($\neg$’ $\langle \textit{statement} \rangle$ ‘)’
		$\mid$ ‘($\wedge$’ $\langle \textit{statement} \rangle \langle \textit{statement} \rangle^{+}$ ‘)’
		$\mid$ ‘($\vee$’ $\langle \textit{statement} \rangle \langle \textit{statement} \rangle^{+}$ ‘)’
		$\mid$ ‘(,’ $\langle \textit{statement} \rangle \langle \textit{statement} \rangle^{+}$ ‘)’
		$\mid$ ‘(;’ $\langle \textit{statement} \rangle \langle \textit{statement} \rangle^{+}$ ‘)’
		$\mid$ ‘($\Uparrow$’ $\langle \textit{word} \rangle \langle \textit{term} \rangle^{*}$ ‘)’
$\langle \textit{copula} \rangle$	$::=$	‘$\rightarrow$’ $\mid$ ‘$\leftrightarrow$’ $\mid$ ‘$\Rightarrow$’ $\mid$ ‘$\Leftrightarrow$’ $\mid$ ‘$/\!\!\Rightarrow$’ $\mid$ ‘$\backslash\!\!\Rightarrow$’ $\mid$ ‘$|\!\!\Rightarrow$’ $\mid$ ‘$/\!\!\Leftrightarrow$’ $\mid$ ‘$|\!\!\Leftrightarrow$’
$\langle \textit{tense} \rangle$	$::=$	‘$/\!\!\Rightarrow$’ $\mid$ ‘$\backslash\!\!\Rightarrow$’ $\mid$ ‘$|\!\!\Rightarrow$’
$\langle \textit{term} \rangle$	$::=$	$\langle \textit{word} \rangle \mid \langle \textit{variable} \rangle \mid \langle \textit{statement} \rangle$
		$\mid$ ‘{’ $\langle \textit{term} \rangle^{+}$ ‘}’ $\mid$ ‘[’ $\langle \textit{term} \rangle^{+}$ ‘]’
		$\mid$ ‘($\cap$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle^{+}$ ‘)’
		$\mid$ ‘($\cup$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle^{+}$ ‘)’
		$\mid$ ‘($-$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle$ ‘)’
		$\mid$ ‘($\ominus$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle$ ‘)’
		$\mid$ ‘($\times$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle^{+}$ ‘)’
		$\mid$ ‘($\perp$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle^{*}$ ‘$\diamond$’ $\langle \textit{term} \rangle^{*}$ ‘)’
		$\mid$ ‘($\top$’ $\langle \textit{term} \rangle \langle \textit{term} \rangle^{*}$ ‘$\diamond$’ $\langle \textit{term} \rangle^{*}$ ‘)’
$\langle \textit{variable} \rangle$	$::=$	$\langle \textit{independent-variable} \rangle$
		$\mid \langle \textit{dependent-variable} \rangle$
		$\mid \langle \textit{query-variable} \rangle$
$\langle \textit{independent-variable} \rangle$	$::=$	‘#’ $\langle \textit{word} \rangle$
$\langle \textit{dependent-variable} \rangle$	$::=$	‘#’ $[\langle \textit{word} \rangle$ ‘(’ $\langle \textit{independent-variable} \rangle^{*}$ ‘)’$]$
$\langle \textit{query-variable} \rangle$	$::=$	‘?’ $[\langle \textit{word} \rangle]$
$\langle \textit{truth-value} \rangle$	:	a pair of real number in $[0, 1] \times (0, 1)$
$\langle \textit{desire-value} \rangle$	:	the same as $\langle \textit{truth-value} \rangle$
$\langle \textit{word} \rangle$	:	a string in a given alphabet

Table 11.1: The Complete Grammar of Narsese

type	**symbol**	**layer**	**name**
sentence punctuation	.	NAL-1	judgment
	?	NAL-1	question
	!	NAL-8	goal
copula	$\rightarrow$	NAL-1	inheritance
	$\leftrightarrow$	NAL-2	similarity
	$\circ\!\!\rightarrow$	NAL-2	instance
	$\rightarrow\!\!\circ$	NAL-2	property
	$\circ\!\!\rightarrow\!\!\circ$	NAL-2	instance-property
	$\Rightarrow$	NAL-5	implication
	$\Leftrightarrow$	NAL-5	equivalence
	$/\!\!\Rightarrow$	NAL-7	predictive implication
	$\backslash\!\!\Rightarrow$	NAL-7	retrospective implication
	$\vert\!\!\Rightarrow$	NAL-7	concurrent implication
	$/\!\!\Leftrightarrow$	NAL-7	predictive equivalence
	$\vert\!\!\Leftrightarrow$	NAL-7	concurrent equivalence
term operator	$\{\}$	NAL-2	extensional set
	$[\,]$	NAL-2	intensional set
	$\cap$	NAL-3	extensional intersection
	$\cup$	NAL-3	intensional intersection
	$-$	NAL-3	extensional difference
	$\ominus$	NAL-3	intensional difference
	$\times$	NAL-4	product
	$\perp$	NAL-4	extensional image
	$\top$	NAL-4	intensional image
	$\diamond$	NAL-4	image place-holder
statement operator	$\neg$	NAL-5	negation
	$\wedge$	NAL-5	conjunction
	$\vee$	NAL-5	disjunction
	,	NAL-7	sequential conjunction
	;	NAL-7	parallel conjunction
term prefix	#	NAL-6	variable
	?	NAL-6	query
	$\Uparrow$	NAL-8	command

Table 11.2: The Symbols in Narsese Grammar

$J_2 \setminus J_1$	$M \to P$	$P \to M$	$M \leftrightarrow P$
$S \to M$	$S \to P\langle F_{ded}\rangle$ $P \to S\langle F'_{exe}\rangle$	$S \to P\langle F_{abd}\rangle$ $P \to S\langle F'_{abd}\rangle$ $S \leftrightarrow P\langle F'_{com}\rangle$	$S \to P\langle F'_{ana}\rangle$
$M \to S$	$S \to P\langle F_{ind}\rangle$ $P \to S\langle F'_{ind}\rangle$ $S \leftrightarrow P\langle F_{com}\rangle$	$S \to P\langle F_{exe}\rangle$ $P \to S\langle F'_{ded}\rangle$	$P \to S\langle F'_{ana}\rangle$
$S \leftrightarrow M$	$S \to P\langle F_{ana}\rangle$	$P \to S\langle F_{ana}\rangle$	$S \leftrightarrow P\langle F_{res}\rangle$

Table 11.3: The First-Order Syllogistic Rules

$J_2 \setminus J_1$	$M \Rightarrow P$	$P \Rightarrow M$	$M \Leftrightarrow P$
$S \Rightarrow M$	$S \Rightarrow P\langle F_{ded}\rangle$ $P \Rightarrow S\langle F'_{exe}\rangle$	$S \Rightarrow P\langle F_{abd}\rangle$ $P \Rightarrow S\langle F'_{abd}\rangle$ $S \Leftrightarrow P\langle F'_{com}\rangle$	$S \Rightarrow P\langle F'_{ana}\rangle$
$M \Rightarrow S$	$S \Rightarrow P\langle F_{ind}\rangle$ $P \Rightarrow S\langle F'_{ind}\rangle$ $S \Leftrightarrow P\langle F_{com}\rangle$	$S \Rightarrow P\langle F_{exe}\rangle$ $P \Rightarrow S\langle F'_{ded}\rangle$	$P \Rightarrow S\langle F'_{ana}\rangle$
$S \Leftrightarrow M$	$S \Rightarrow P\langle F_{ana}\rangle$	$P \Rightarrow S\langle F_{ana}\rangle$	$S \Leftrightarrow P\langle F_{res}\rangle$

Table 11.4: The Higher-Order Syllogistic Rules

J_1	J_2	J	F
S	$S \Leftrightarrow P$	P	F_{ana}
S	P	$S \Leftrightarrow P$	F_{com}
$M \Rightarrow P$	M	P	F_{ded}
$P \Rightarrow M$	M	P	F_{abd}
P	M	$M \Rightarrow P$	F_{ind}
$(C \wedge M) \Rightarrow P$	M	$C \Rightarrow P$	F_{ded}
$(C \wedge M) \Rightarrow P$	$C \Rightarrow P$	M	F_{abd}
$C \Rightarrow P$	M	$(C \wedge M) \Rightarrow P$	F_{ind}
$(C \wedge M) \Rightarrow P$	$S \Rightarrow M$	$(C \wedge S) \Rightarrow P$	F_{ded}
$(C \wedge M) \Rightarrow P$	$(C \wedge S) \Rightarrow P$	$S \Rightarrow M$	F_{abd}
$(C \wedge S) \Rightarrow P$	$S \Rightarrow M$	$(C \wedge M) \Rightarrow P$	F_{ind}

Table 11.5: The Conditional Syllogistic Rules

J_1	J_2	J	F
$M \to T_1$	$M \to T_2$	$M \to (T_1 \cap T_2)$	F_{int}
		$M \to (T_1 \cup T_2)$	F_{uni}
		$M \to (T_1 - T_2)$	F_{dif}
		$M \to (T_2 - T_1)$	F'_{dif}
$T_1 \to M$	$T_2 \to M$	$(T_1 \cup T_2) \to M$	F_{int}
		$(T_1 \cap T_2) \to M$	F_{uni}
		$(T_1 \ominus T_2) \to M$	F_{dif}
		$(T_2 \ominus T_1) \to M$	F'_{dif}
$M \Rightarrow T_1$	$M \Rightarrow T_2$	$M \Rightarrow (T_1 \wedge T_2)$	F_{int}
		$M \Rightarrow (T_1 \vee T_2)$	F_{uni}
$T_1 \Rightarrow M$	$T_2 \Rightarrow M$	$(T_1 \vee T_2) \Rightarrow M$	F_{int}
		$(T_1 \wedge T_2) \Rightarrow M$	F_{uni}
T_1	T_2	$T_1 \wedge T_2$	F_{int}
		$T_1 \vee T_2$	F_{uni}

Table 11.6: The Composition Rules

st_1	st_2	st_3
$\neg(M \to (T_1 \cap T_2))$	$M \to T_1$	$\neg(M \to T_2)$
$M \to (T_1 \cup T_2)$	$\neg(M \to T_1)$	$M \to T_2$
$\neg(M \to (T_1 - T_2))$	$M \to T_1$	$M \to T_2$
$\neg(M \to (T_2 - T_1))$	$\neg(M \to T_1)$	$\neg(M \to T_2)$
$\neg((T_1 \cup T_2) \to M)$	$T_1 \to M$	$\neg(T_2 \to M)$
$(T_1 \cap T_2) \to M$	$\neg(T_1 \to M)$	$T_2 \to M$
$\neg((T_1 \ominus T_2) \to M)$	$T_1 \to M$	$T_2 \to M$
$\neg((T_2 \ominus T_1) \to M)$	$\neg(T_1 \to M)$	$\neg(T_2 \to M)$
$\neg(S_1 \wedge S_2)$	S_1	$\neg S_2$
$S_1 \vee S_2$	$\neg S_1$	S_2

Table 11.7: The Decomposition Rules

J_0	J	F
S	$\neg S$	F_{neg}
$S \to P$	$P \to S$	F_{cnv}
$S \Rightarrow P$	$P \Rightarrow S$	F_{cnv}
$S \Rightarrow P$	$(\neg P) \Rightarrow (\neg S)$	F_{cnt}

Table 11.8: The Conversion Rules

$statement_1$	$statement_2$
$S \leftrightarrow P$	$(S \rightarrow P) \wedge (P \rightarrow S)$
$S \Leftrightarrow P$	$(S \Rightarrow P) \wedge (P \Rightarrow S)$
$S \leftrightarrow P$	$\{S\} \leftrightarrow \{P\}$
$S \leftrightarrow P$	$[S] \leftrightarrow [P]$
$S \rightarrow \{P\}$	$S \leftrightarrow \{P\}$
$[S] \rightarrow P$	$[S] \leftrightarrow P$
$(S_1 \times S_2) \rightarrow (P_1 \times P_2)$	$(S_1 \rightarrow P_1) \wedge (S_2 \rightarrow P_2)$
$(S_1 \times S_2) \leftrightarrow (P_1 \times P_2)$	$(S_1 \leftrightarrow P_1) \wedge (S_2 \leftrightarrow P_2)$
$S \rightarrow P$	$(M \times S) \rightarrow (M \times P)$
$S \rightarrow P$	$(S \times M) \rightarrow (P \times M)$
$S \leftrightarrow P$	$(M \times S) \leftrightarrow (M \times P)$
$S \leftrightarrow P$	$(S \times M) \leftrightarrow (P \times M)$
$(\times\ T_1\ T_2) \rightarrow R$	$T_1 \rightarrow (\perp R \diamond T_2)$
$(\times\ T_1\ T_2) \rightarrow R$	$T_2 \rightarrow (\perp R\ T_1 \diamond)$
$R \rightarrow (\times\ T_1\ T_2)$	$(\top R \diamond T_2) \rightarrow T_1$
$R \rightarrow (\times\ T_1\ T_2)$	$(\top R\ T_1 \diamond) \rightarrow T_2$
$\neg(S_1 \wedge S_2)$	$(\neg S_1) \vee (\neg S_2)$
$\neg(S_1 \vee S_2)$	$(\neg S_1) \wedge (\neg S_2)$
$S_1 \Leftrightarrow S_2$	$(\neg S_1) \Leftrightarrow (\neg S_2)$
$\neg(S_1 \Rightarrow S_2)$	$S_1 \Rightarrow (\neg S_2)$
$\neg(S_1 \Leftrightarrow S_2)$	$S_1 \Leftrightarrow (\neg S_2)$

Table 11.9: The Equivalence Theorems

$term_1$	$term_2$
$\neg(\neg T)$	T
$(\cup\ \{T_1\}\ \cdots\ \{T_n\})$	$\{T_1,\ \cdots,\ T_n\}$
$(\cap\ [T_1]\ \cdots\ [T_n])$	$[T_1,\ \cdots,\ T_n]$
$(\{T_1,\ \cdots,\ T_n\} - \{T_n\})$	$\{T_1,\ \cdots,\ T_{n-1}\}$
$([T_1,\ \cdots,\ T_n] \ominus [T_n])$	$[T_1,\ \cdots,\ T_{n-1}]$
$((T_1 \times T_2) \perp T_2)$	T_1
$((T_1 \times T_2) \top T_2)$	T_1
$S_1 \Rightarrow (S_2 \Rightarrow S_3)$	$(S_1 \wedge S_2) \Rightarrow S_3$

Table 11.10: The Reduction Theorems

$statement_1$	$statement_2$
$S \leftrightarrow P$	$S \rightarrow P$
$S \Leftrightarrow P$	$S \Rightarrow P$
$S_1 \wedge S_2$	S_1
S_1	$S_1 \vee S_2$
$S \rightarrow P$	$(S \cup M) \rightarrow (P \cup M)$
$S \rightarrow P$	$(S \cap M) \rightarrow (P \cap M)$
$S \leftrightarrow P$	$(S \cup M) \leftrightarrow (P \cup M)$
$S \leftrightarrow P$	$(S \cap M) \leftrightarrow (P \cap M)$
$S \Rightarrow P$	$(S \vee M) \Rightarrow (P \vee M)$
$S \Rightarrow P$	$(S \wedge M) \Rightarrow (P \wedge M)$
$S \Leftrightarrow P$	$(S \vee M) \Leftrightarrow (P \vee M)$
$S \Leftrightarrow P$	$(S \wedge M) \Leftrightarrow (P \wedge M)$
$S \rightarrow P$	$(S - M) \rightarrow (P - M)$
$S \rightarrow P$	$(M - P) \rightarrow (M - S)$
$S \rightarrow P$	$(S \ominus M) \rightarrow (P \ominus M)$
$S \rightarrow P$	$(M \ominus P) \rightarrow (M \ominus S)$
$S \leftrightarrow P$	$(S - M) \leftrightarrow (P - M)$
$S \leftrightarrow P$	$(M - P) \leftrightarrow (M - S)$
$S \leftrightarrow P$	$(S \ominus M) \leftrightarrow (P \ominus M)$
$S \leftrightarrow P$	$(M \ominus P) \leftrightarrow (M \ominus S)$
$M \rightarrow (T_1 - T_2)$	$\neg(M \rightarrow T_2)$
$(T_1 \ominus T_2) \rightarrow M$	$\neg(T_2 \rightarrow M)$
$S \rightarrow P$	$(S \perp M) \rightarrow (P \perp M)$
$S \rightarrow P$	$(S \top M) \rightarrow (P \top M)$
$S \rightarrow P$	$(M \perp P) \rightarrow (M \perp S)$
$S \rightarrow P$	$(M \top P) \rightarrow (M \top S)$

Table 11.11: The Implication Theorems

$term_1$	$term_2$
$(T_1 \cap T_2)$	T_1
T_1	$(T_1 \cup T_2)$
$(T_1 - T_2)$	T_1
T_1	$(T_1 \ominus T_2)$
$((R \perp T) \times T)$	R
R	$((R \top T) \times T)$

Table 11.12: The Inheritance Theorems

type	**inference**	**name**	**function**
same-statement	revision	F_{rev}	$w^+ = w_1^+ + w_2^+$
			$w^- = w_1^- + w_2^-$
single-premise	negation	F_{neg}	$w^+ = w_0^-$
			$w^- = w_0^+$
	conversion	F_{cnv}	$w^+ = and(f_0, c_0)$
			$w^- = 0$
	contraposition	F_{cnt}	$w^+ = 0$
			$w^- = and((not(f_0), c_0)$
strong syllogism	deduction	F_{ded}	$f = and(f_1, f_2)$
			$c = and(f_1, f_2, c_1, c_2)$
	analogy	F_{ana}	$f = and(f_1, f_2)$
			$c = and(f_2, c_1, c_2)$
	resemblance	F_{res}	$f = and(f_1, f_2)$
			$c = and(or(f_1, f_2), c_1, c_2)$
weak syllogism	abduction	F_{abd}	$w^+ = and(f_1, f_2, c_1, c_2)$
			$w = and(f_1, c_1, c_2)$
	induction	F_{ind}	$w^+ = and(f_1, f_2, c_1, c_2)$
			$w = and(f_2, c_1, c_2)$
	exemplification	F_{exe}	$w^+ = and(f_1, f_2, c_1, c_2)$
			$w = and(f_1, f_2, c_1, c_2)$
	comparison	F_{com}	$w^+ = and(f_1, f_2, c_1, c_2)$
			$w = and(or(f_1, f_2), c_1, c_2)$
term composition	intersection	F_{int}	$f = and(f_1, f_2)$
			$c = or(and(not(f_1), c_1), and(not(f_2), c_2))$
			$+ and(f_1, c_1, f_2, c_2)$
	union	F_{uni}	$f = or(f_1, f_2)$
			$c = or(and(f_1, c_1), and(f_2, c_2))$
			$+ and(not(f_1), c_1, not(f_2), c_2)$
	difference	F_{dif}	$f = and(f_1, not(f_2))$
			$c = or(and(not(f_1), c_1), and(f_2, c_2))$
			$+ and(f_1, c_1, not(f_2), c_2)$

Table 11.13: The Truth-Value Functions of NAL

Bibliography

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

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

[Wang, 1995b] Wang, P. (1995b). Reference classes and multiple inheritances. *International Journal of Uncertainty, Fuzziness and and Knowledge-based Systems*, 3(1):79–91.

[Wang, 1996a] Wang, P. (1996a). Heuristics and normative models of judgment under uncertainty. *International Journal of Approximate Reasoning*, 14(4):221–235.

[Wang, 1996b] Wang, P. (1996b). The interpretation of fuzziness. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(4):321–326.

[Wang, 1996c] Wang, P. (1996c). Problem-solving under insufficient resources. In *Working Notes of the AAAI Fall Symposium on Flexible Computation*, pages 148–155, Cambridge, Massachusetts.

[Wang, 2000] Wang, P. (2000). Unified inference in extended syllogism. In Flach, P. and Kakas, A., editors, *Abduction and Induction: Essays on Their Relation and Integration*, pages 117–129. Kluwer Academic Publishers, Dordrecht.

[Wang, 2001a] Wang, P. (2001a). Abduction in non-axiomatic logic. In *Working Notes of the IJCAI workshop on Abductive Reasoning*, pages 56–63, Seattle, Washington.

[Wang, 2001b] Wang, P. (2001b). Confidence as higher-order uncertainty. In *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 352–361, Ithaca, New York.

[Wang, 2004a] Wang, P. (2004a). The limitation of Bayesianism. *Artificial Intelligence*, 158(1):97–106.

[Wang, 2004b] Wang, P. (2004b). Problem solving with insufficient resources. *International Journal of Uncertainty, Fuzziness and and Knowledge-based Systems*, 12(5):673–700.

[Wang, 2004c] Wang, P. (2004c). Toward a unified artificial intelligence. In *Papers from the 2004 AAAI Fall Symposium on Achieving Human-Level Intelligence through Integrated Research and Systems*, pages 83–90, Washington DC.

[Wang, 2005] Wang, P. (2005). Experience-grounded semantics: a theory for intelligent systems. *Cognitive Systems Research*, 6(4):282–302.

[Wang, 2006] Wang, P. (2006). *Rigid Flexibility: The Logic of Intelligence*. Springer, Dordrecht.

[Wang, 2007a] Wang, P. (2007a). From NARS to a thinking machine. In Goertzel, B. and Wang, P., editors, *Advance of Artificial General Intelligence*, pages 75–93. IOS Press, Amsterdam.

[Wang, 2007b] Wang, P. (2007b). The logic of intelligence. In Goertzel, B. and Pennachin, C., editors, *Artificial General Intelligence*, pages 31–62. Springer, Berlin.

[Wang, 2009a] Wang, P. (2009a). Analogy in a general-purpose reasoning system. *Cognitive Systems Research*, 10(3):286–296.

[Wang, 2009b] Wang, P. (2009b). Case-by-case problem solving. In *Proceedings of the Second Conference on Artificial General Intelligence*, pages 180–185.

[Wang, 2009c] Wang, P. (2009c). Formalization of evidence: A comparative study. *Journal of Artificial General Intelligence*, 1:25–53.