Introduction to NARS

(Non-Axiomatic Reasoning System)

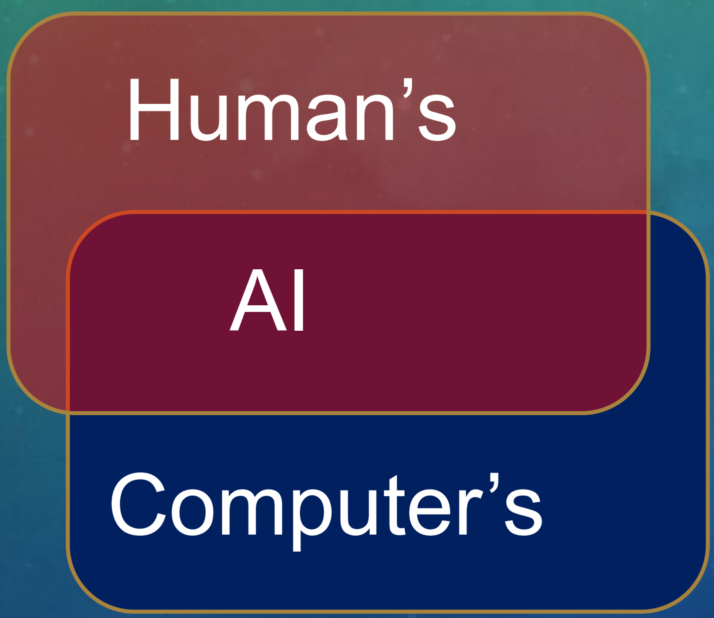
Xiang Li
Temple University

Intelligence as a Category

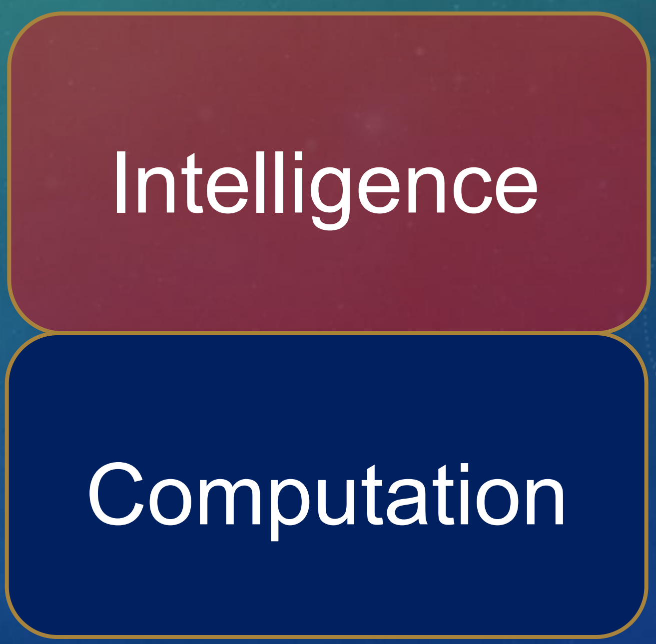
	Human intelligence
	Computer intelligence
Intelligence	Animal intelligence
	Collective intelligence
	Alien intelligence

Intelligence & Computation

As problem-solving capabilities



As principles and mechanisms



What is not Intelligent

- Innate behavior, or instinct
- Exhaustive search
- Information retrieval
- Repeated routines
- Algorithm following (numerical calculating, sorting, fixed mapping, …)

A Working Definition

“Intelligence” is *the capability of a system to adapt to its environment and to work with insufficient knowledge and resources*

Assumption of Insufficient Knowledge and Resources (AIKR):

- To rely on *finite* processing capacity
- To work in *real time*
- To *open* to unexpected tasks

Presumption of Computation

"Computation" specifies fixed input-output mapping, which is accurately repeatable, predictable, and terminatable

Assumption on knowledge and resources:

- The system is equipped with problem-specific algorithms
- The time-space expenses of the algorithms are affordable

Relative Rationality

- Under AIKR, no more traditional correctness or optimality
- "Rational" becomes relative to available knowledge and resources, so is subjective and evolving, but not arbitrary
- AIKR makes adaptation necessary, which is neither desired nor possible in computational model of problem solving

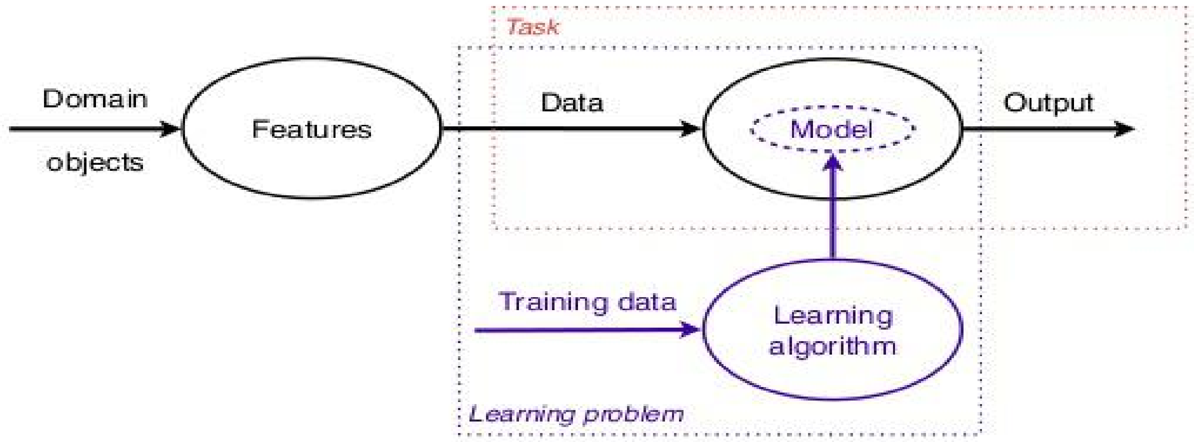
Adaptation

- To predict the future according to the past, though the past and the future are surely different (Hume's Problem)
- To relatively satisfy the (effectively) infinite resource demand using the finite supply, though it is never enough
- An "adaptive system" is one that tries to adapt, though no success is guaranteed

Machine Learning

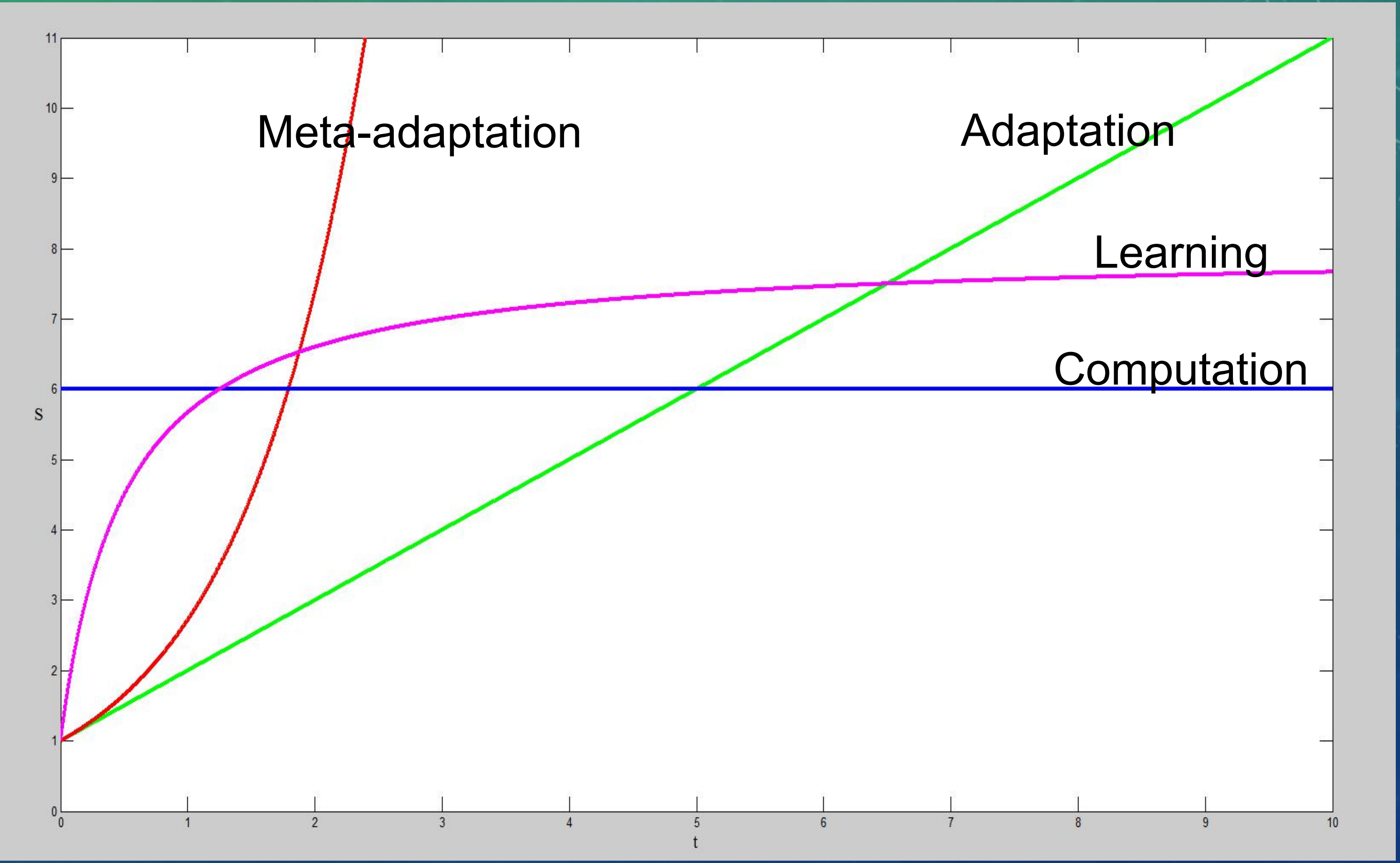
Figure 3, p.11

How machine learning helps to solve a task



An overview of how machine learning is used to address a given task. A task (red box) requires an appropriate mapping – a model – from data described by features to outputs. Obtaining such a mapping from training data is what constitutes a learning problem (blue box).

Learning and Adaptation

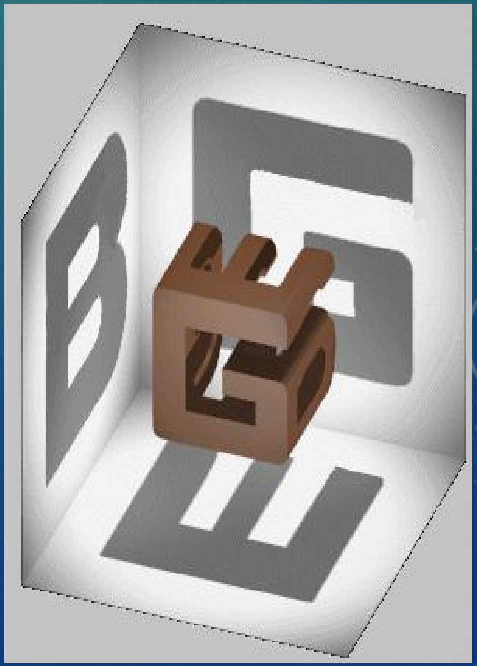


To Model Multiple Functions

Hybrid

Integrated

Unified



Formal Model

The necessity of a formal model between an informal theory and a physical realization

- Multidimensional space, vector, tensor, trajectory, neural network, …
- Data structure and algorithm, control structures, computational complexity, …
- Formal language and inference rules, …

Reasoning System Framework

- a *language* for representation
- a *semantics* of the language
- a set of inference *rules*
- a *memory* structure
- a *control* mechanism

Advantages:

- Domain independence
- Rich expressing power
- Justifiability of the rules
- Flexibility in combining the rules

Existing Issues

- Uncertainty, inconsistency, and implicity in commonsense knowledge
- "Symbol grounding" problem
- Validity of non-deductive inference: induction, abduction, analogy, revision, …
- Combinatorial explosion
- Integration with other cognitive functions

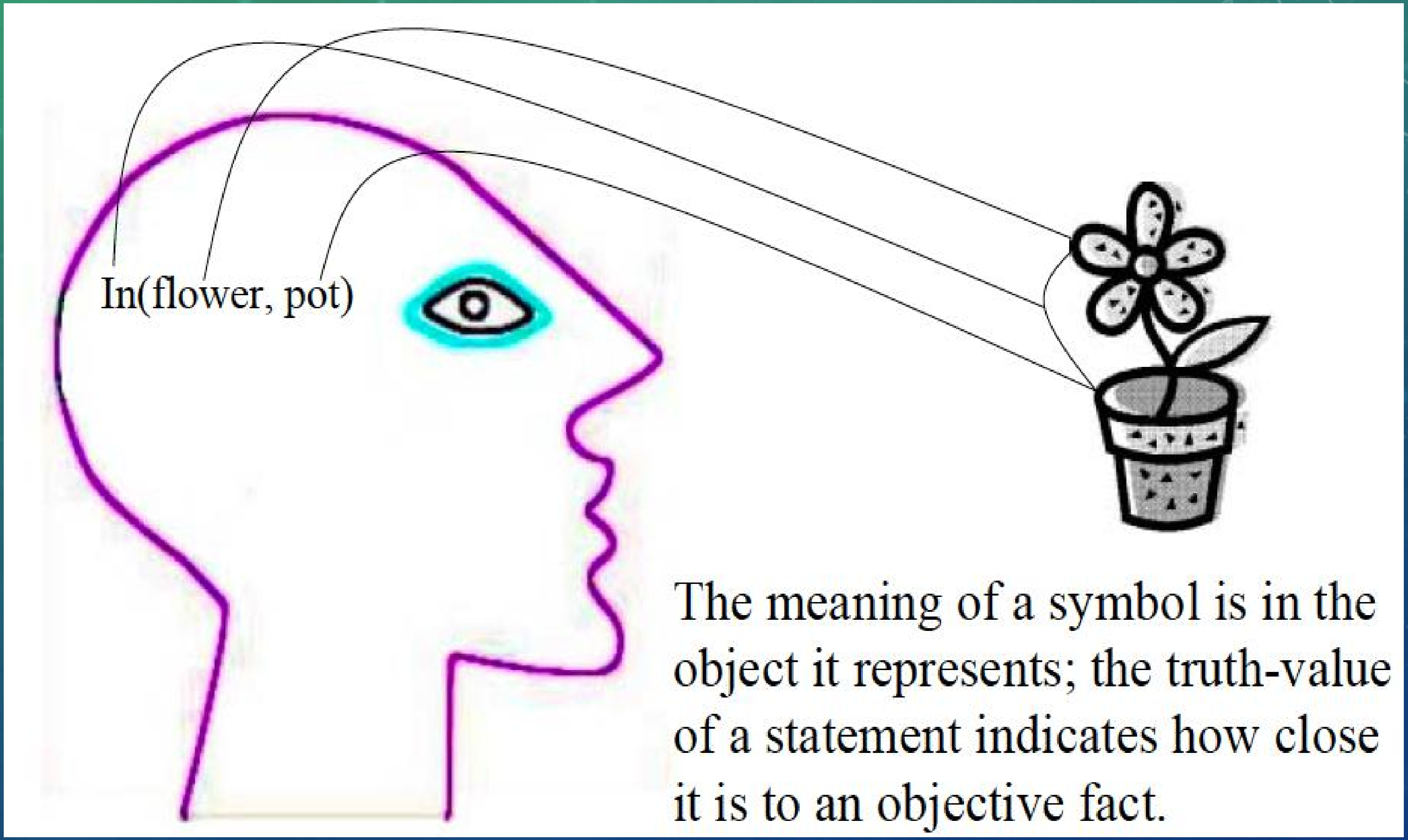
Non-Axiomatic Reasoning System

- NARS has a logic part and a control part, with a “logic” in the original sense
- NARS is fully based on AIKR
- NARS is a normative model built on a descriptive foundation
- NARS has a designed meta-level and an acquired object-level

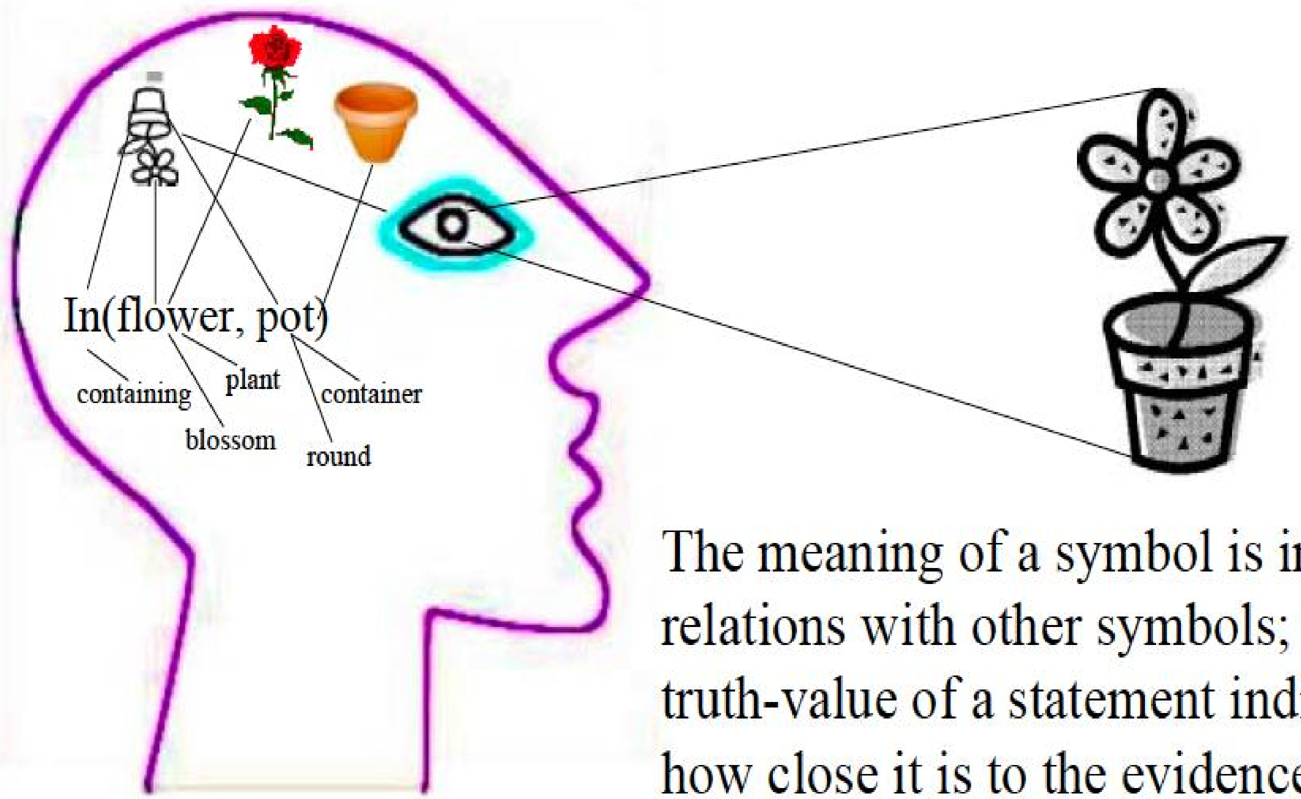
Meaning and Truth

- Correspondence: the “meaning” of a symbol is the object/event it refers to; the “truth-value” of a statement measures its agreement with the reality
- Experience-grounded: the “meaning” of a symbol is its experienced relation; the “truth-value” of a statement measures its evidential support

Model-Theoretic Semantics



Experience-Grounded Semantics



The meaning of a symbol is in its relations with other symbols; the truth-value of a statement indicates how close it is to the evidence.

Representation Language

- Predicate calculus: good for theorem proving, but unsuitable for commonsense reasoning
- Term logic: closer to everyday thinking, easy to extend to include compounds terms and non-deductive rules, with graphical interpretation, coherent with conceptual model, …

Term and Statement

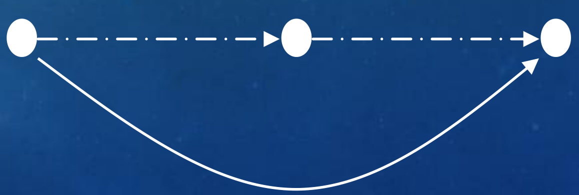
Term: word, as name of a concept

Statement: subject-copula-predicate

$S \rightarrow P$

as specialization-generalization

Copula *inheritance* is reflexive and transitive



Experience-based Truth

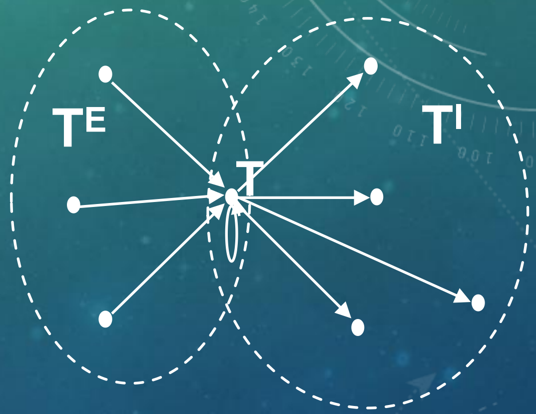
- Experience ***K***: a finite set of statements
- Beliefs ***K****: the transitive closure of ***K***
- A statement is *true* if
  either it is in ***K****
  or it has the form of $X \rightarrow X$
  otherwise it is *false*

Extension and Intension

For a given term T,

its *extension* $T^E = \{x \mid x \rightarrow T\}$

its *intension* $T^I = \{x \mid T \rightarrow x\}$



Theorem:

$(S \rightarrow P) \Leftrightarrow (S^E \subseteq P^E) \Leftrightarrow (P^I \subseteq S^I)$

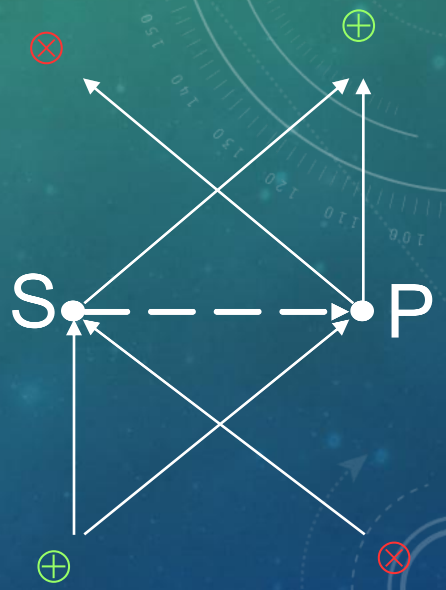
Evidence

Positive evidence of $S \longrightarrow P$:

$\{x \mid x \in (S^E \cap P^E) \cup (P^I \cap S^I)\}$

Negative evidence of $S \longrightarrow P$:

$\{x \mid x \in (S^E - P^E) \cup (P^I - S^I)\}$



Amount of evidence:

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

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

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

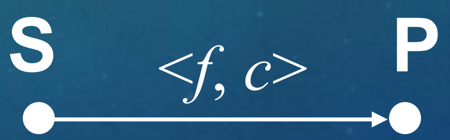
Truth-Value Defined

In NARS, the truth-value of a statement is a pair of real numbers in [0, 1], and measures the evidential support to the statement.

$S \rightarrow P <f, c>$

frequency: $f = w^+/w$

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



Truth-Value Produced

- Actual experience: a stream of statements with truth-value, where the confidence is in (0, 1)
- Each inference rule has a truth-value function, and the truth-value of the conclusion is determined only by the evidence provided by the premises

Compared to Probability

Based on similar intuition, probability theory has axioms that requires consistent probability assignments

The confidence in NARS cannot be interpreted as a probability in the same sample space as the frequency

The operations are different

Compared to Fuzzy Logic

Truth is a matter of degree

In fuzzy logic, degrees of membership are purely subjective, and the operations on them have little justification

In NARS, truth-value has an evidence-based interpretation, by which randomness, fuzziness, and ignorance are unified

Truth-value Function Design

1. Treat all involved variables as Boolean

2. For each value combination in premises, decide the values in conclusion

3. Build Boolean functions among the variables

4. Extend the operators to real-number:

$$\text{not}(x) = 1 - x$$

$$\text{and}(x, y) = x * y$$

$$\text{or}(x, y) = 1 - (1 - x) * (1 - y)$$

Types of Inference Rules

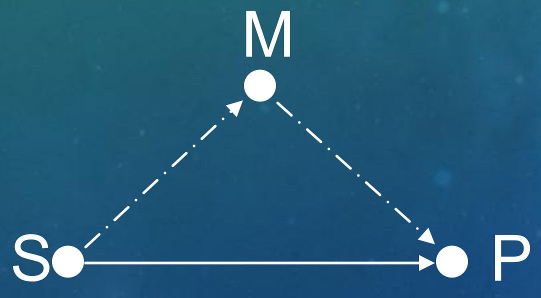
- **Local Inference**: revising beliefs or choosing an answer for a question
- **Forward inference**: from existing beliefs to new beliefs (deduction, induction, abduction, …)
- **Backward inference**: from existing questions and beliefs and to derived questions

Deduction

$$\frac{\begin{array}{l} M \rightarrow P\ [f_1, c_1] \\ S \rightarrow M\ [f_2, c_2] \end{array}}{S \rightarrow P\ [f, c]}$$

$f = \text{and}(f_1, f_2)$

$c = \text{and}(f_1, f_2, c_1, c_2)$

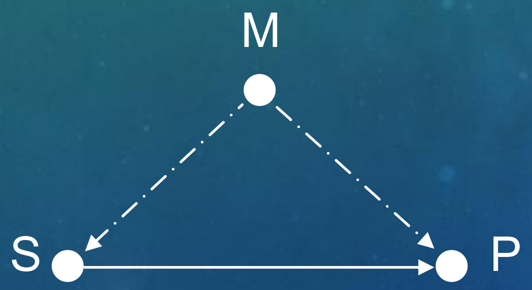


$$\frac{\begin{array}{l} \textit{bird} \rightarrow \textit{animal}\ [1.00, 0.90] \\ \textit{robin} \rightarrow \textit{bird}\ [1.00, 0.90] \end{array}}{\textit{robin} \rightarrow \textit{animal}\ [1.00, 0.81]}$$

Induction

$$\frac{M \rightarrow P\ [f_1, c_1] \qquad M \rightarrow S\ [f_2, c_2]}{S \rightarrow P\ [f, c]}$$

$\boldsymbol{w^+} = \text{and}(f_1, f_2, c_1, c_2)$
$\boldsymbol{w} = \text{and}(f_2, c_1, c_2)$



swan → *bird* [1.00, 0.90]
swan → *swimmer* [1.00, 0.90]

---

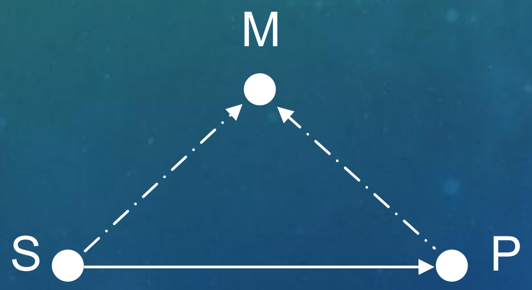
bird → *swimmer* [1.00, 0.45]

Abduction

$$\frac{P \rightarrow M\ [f_1, c_1] \quad S \rightarrow M\ [f_2, c_2]}{S \rightarrow P\ [f, c]}$$

$\boldsymbol{w^+} = \text{and}(f_1, f_2, c_1, c_2)$

$\boldsymbol{w} = \text{and}(f_1, c_1, c_2)$



$$\frac{\textit{seabird} \rightarrow \textit{swimmer}\ [1.00, 0.90] \quad \textit{gull} \rightarrow \textit{swimmer}\ [1.00, 0.90]}{\textit{gull} \rightarrow \textit{seabird}\ [1.00, 0.45]}$$

Revision

$$S \longrightarrow P\,[f_1, c_1]$$
$$S \longrightarrow P\,[f_2, c_2]$$

$$S \longrightarrow P\,[f, c]$$

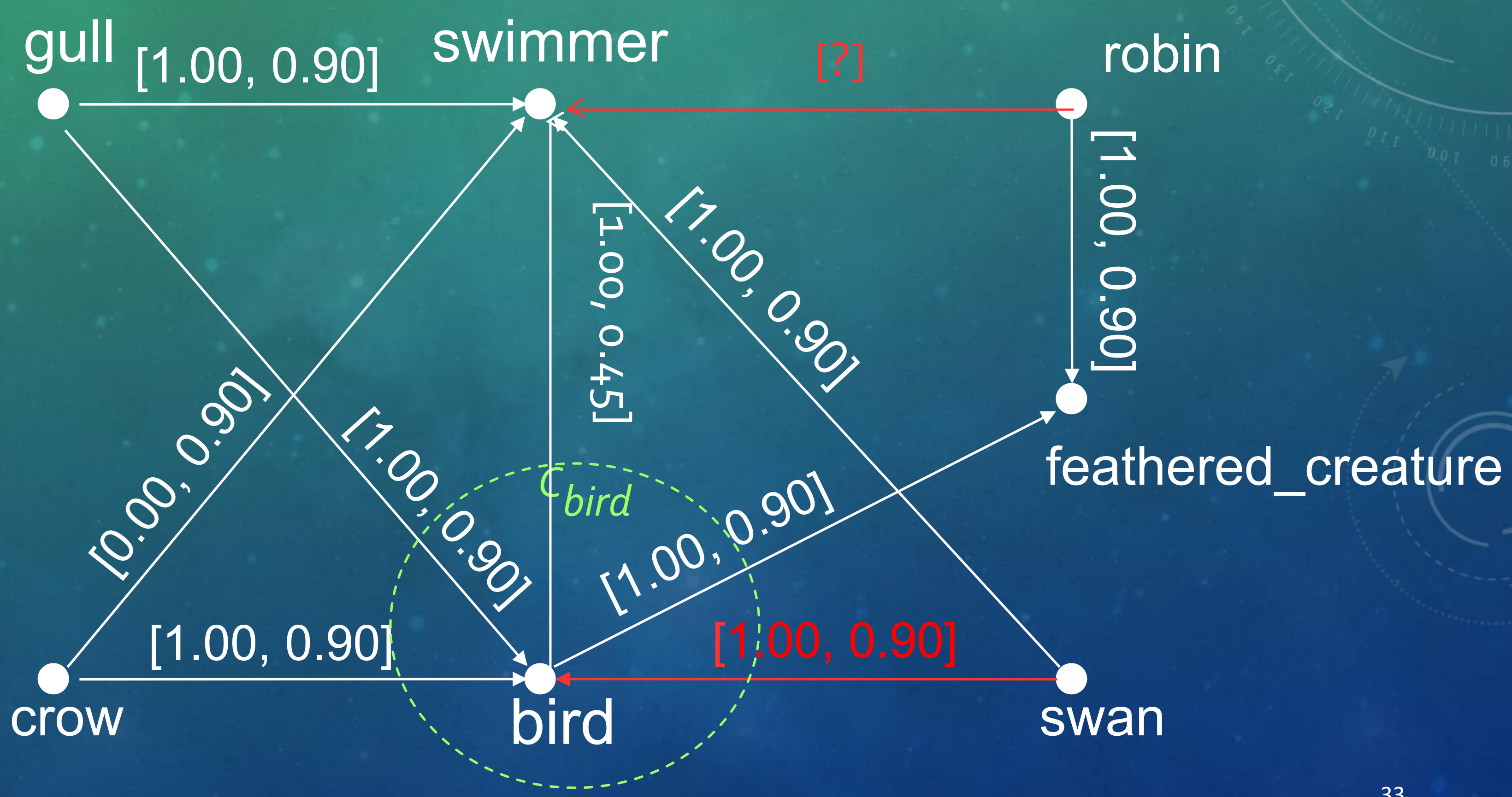
$$\boldsymbol{w}^+ = w^+_1 + w^+_2$$
$$\boldsymbol{w} = w_1 + w_2$$

bird ⟶ *swimmer* [1.00, 0.62]
bird ⟶ *swimmer* [0.00, 0.45]

---

bird ⟶ *swimmer* [0.67, 0.71]

Memory as a Network



Memory Structure

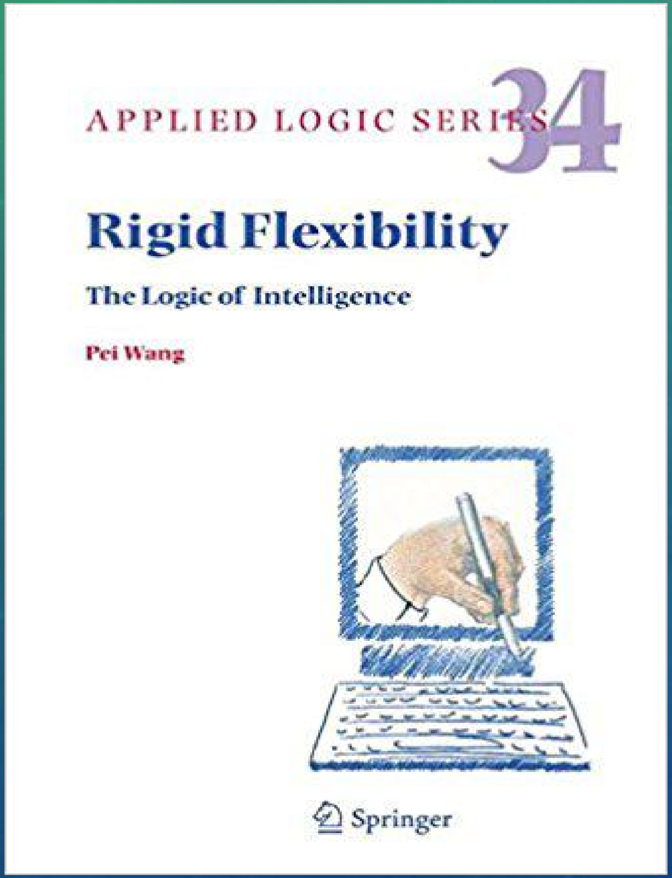
- A *task* is either a question or a piece of new knowledge
- *A belief* is accepted knowledge
- The tasks and beliefs are clustered into *concepts* according to the terms
- Concepts are prioritized in the memory; tasks and beliefs are prioritized within each concept

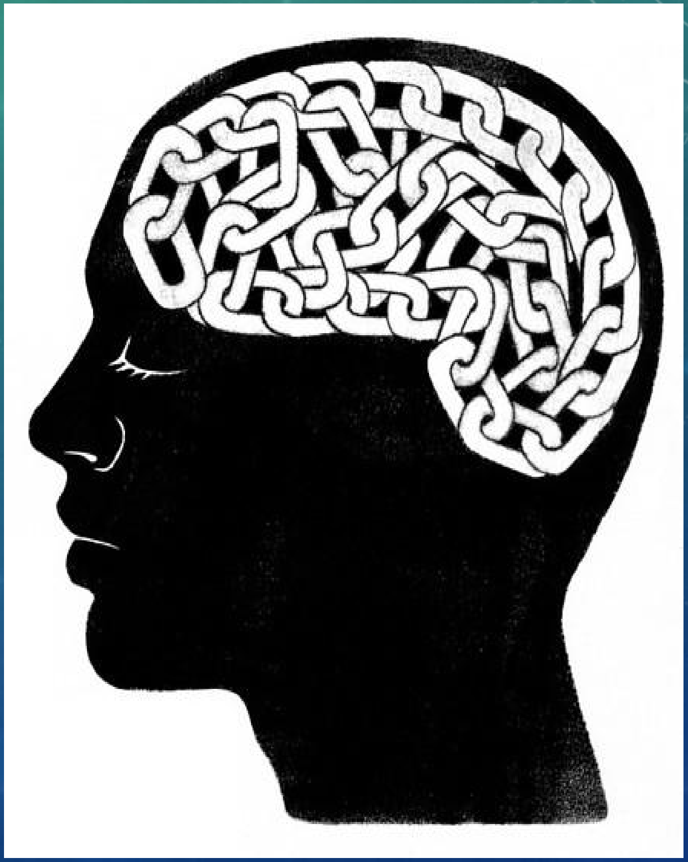
Control Strategy

- In each step, a task interacts with a belief according to applicable rules
- The task and belief are selected probabilistically, biased by priority
- Factors influence the priority of an item: its quality, its usefulness in history, and its relevance to the current context

Non-algorithmic task processing

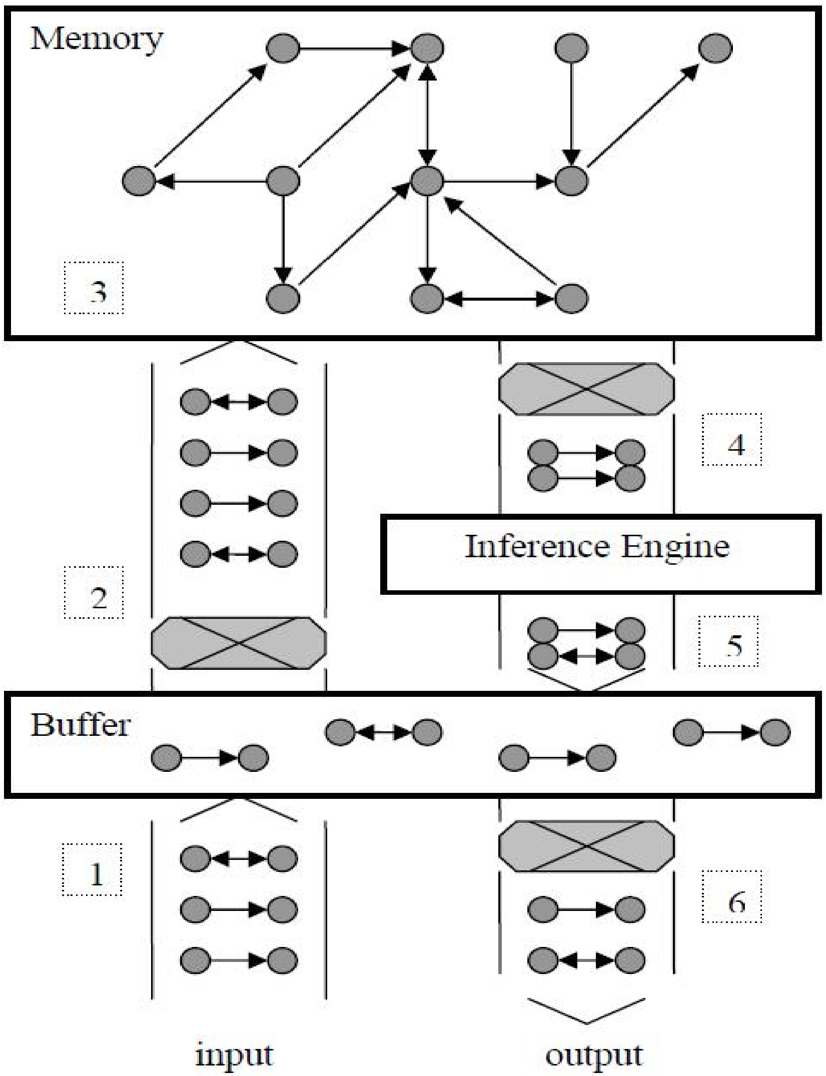
Rigidity and Flexibility





Rigidity vs. Flexibility

Architecture and Routine



1. Input tasks are added into the task buffer.

2. Selected tasks are inserted into the memory.

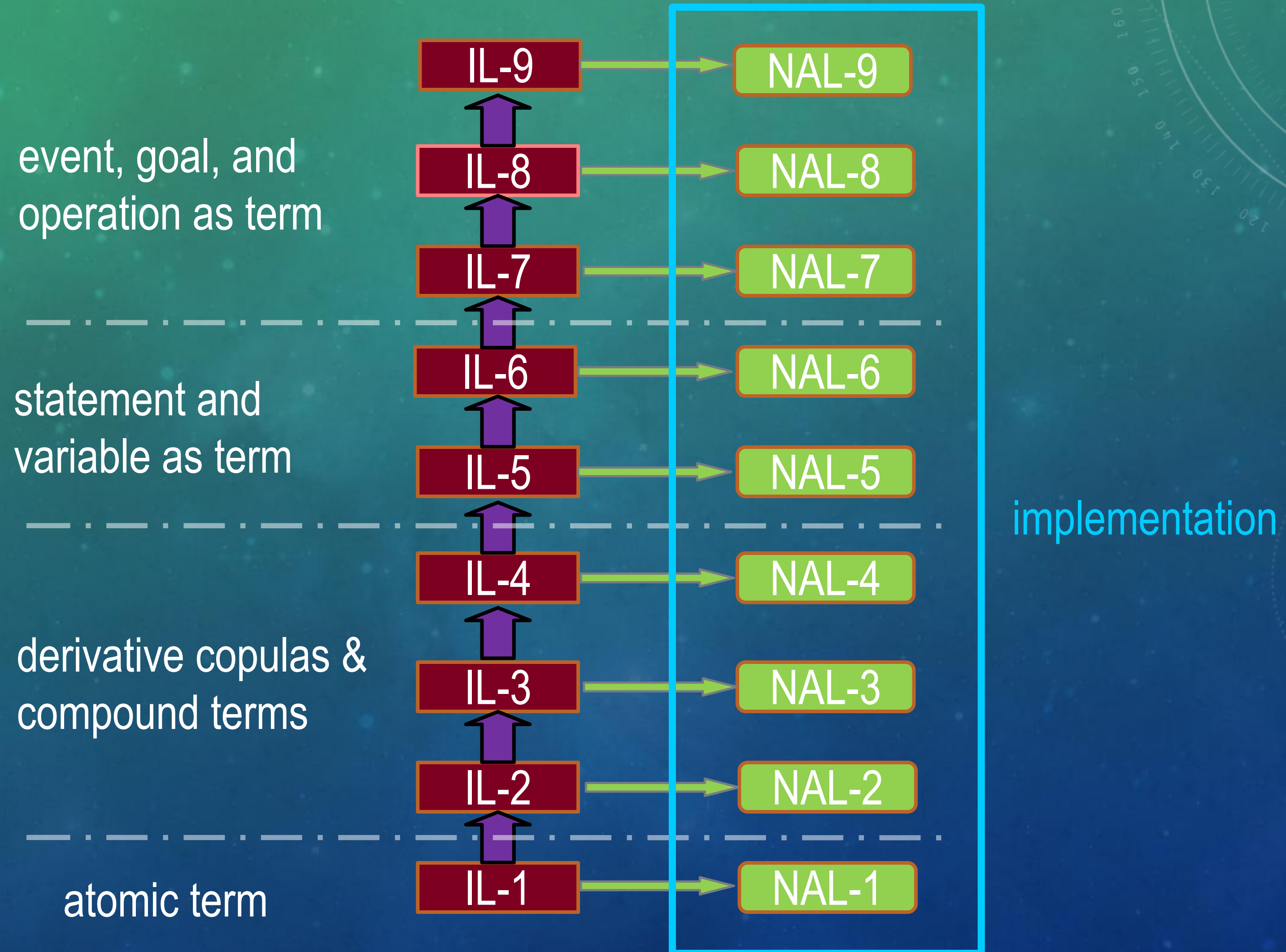
3. Inserted tasks in memory may also produce beliefs and concepts, as well as change existing ones.

4. In each working cycle, a task and a belief are selected from a concept, and feed to the inference engine as premises.

5. The conclusions derived from the premises by applicable rules are added into the buffer as derived tasks.

6. Selected derived tasks are reported as output tasks.

The Layers of the Logic



Copulas & Compound Terms

Ideas from set theory:

- Variants of the *inheritance* copula: *similarity*, *instance*, and *property*
- Compound terms: *sets*, *intersections*, *differences*, *products*, and *images*
- New inference rules for *comparison*, *analogy*, plus compound-term *composition* and *decomposition*

Higher-Order Reasoning

Ideas from propositional/predicate logic:

- Copulas: *implication* and *equivalence*
- Compound statements: *negation*, *conjunction*, and *disjunction*
- *Conditional inferences as implication*
- Variable terms as symbols

NAL as a Meta-logic

- NARS can represent the words, phrase, and sentences of another language as terms
- NARS can represent the inference rules of another logic as implication statements
- Natural language processing: combined syntax, semantics, and pragmatics
- Mathematical reasoning: local axiomatic subsystem

Procedural Reasoning

Ideas from logic programming:

- *Events* as statements with temporal relations (*sequential* and *parallel*)
- *Operations* as executable events, with a sensorimotor interface
- *Goals* as events to be realized
- *Mental operations* are integrated into the inference process

NARS as an agent

From question-answering to goal-achieving

Causal inference, prediction, explanation

Planning, skill acquisition, self-programming

Self-awareness and self-control

Emotion and feeling

Unifications in NARS

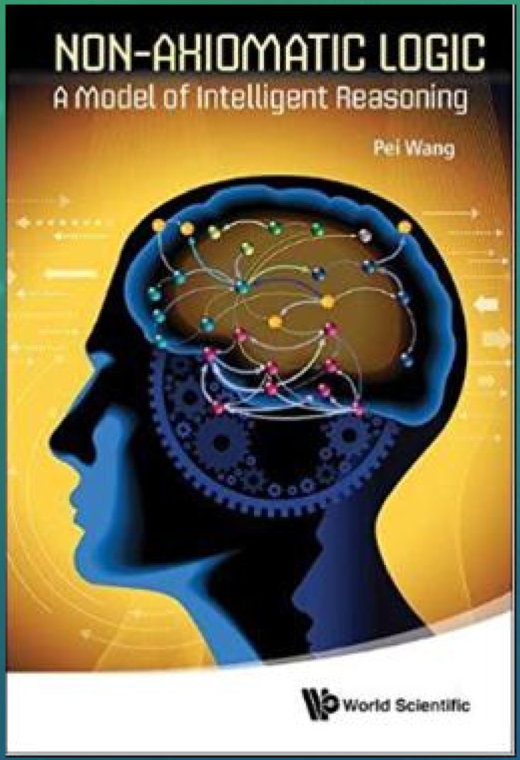
- Fully based on AIKR
- Unified representational language
- Complete inferential power
- *Reasoning as learning, planning, perceiving, problem solving, decision making, …*
- Using other software & hardware by plug-and-play

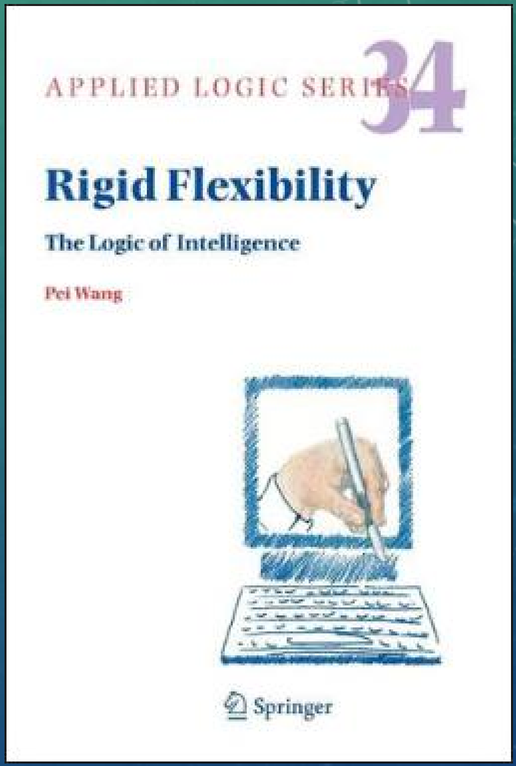
Learning as Reasoning

- Learning (adaptation) is the long-term effects of the inference processes
- Task processing does not follow any fixed algorithm, but is data-driven, real-time, lifelong, context-sensitive, and incremental
- The system uses whatever knowledge and resources available, and provides justifiable solutions with numerical evaluation

Unified Explanation

Creativity, originality, intentionality, forecasting, interpreting, guessing, imagining, understanding, attention, association, forgetting, inspiration, intuition, motivation derivation, skill acquisition, tool using, language acquisition, local maximization, faith forming, communication, emotion, aesthetics, gaming playing, self awareness, self control, consciousness …





https://cis.temple.edu/~wangp/papers.html

NAL-1(Revision)

//Bird is a type of swimmer.

<bird --> swimmer>.

//Bird is probably not a type of swimmer.

<bird --> swimmer>. %0.10;0.60%

//Bird is very likely to be a type of swimmer.

<bird --> swimmer>. %0.87;0.91%')

NAL-1(Deduction)

//Bird is a type of animal.

<bird --> animal>.

//Robin is a type of bird.

<robin --> bird>.

//Robin is a type of animal.

<robin --> animal>. %1.00;0.81%'

NAL-1(Abduction)

// Sport is a type of competition.

<sport --> competition>.

// Chess is a type of competition.

<chess --> competition>. %0.90%

// I guess sport is a type of chess.

<sport --> chess>. %1.00;0.42%')

// I guess chess is a type of sport.

<chess --> sport>. %0.90;0.45%

NAL-1(Induction)

// Swan is a type of swimmer.

<swan --> swimmer>. %0.90%

// Swan is a type of bird.

<swan --> bird>.

// I guess bird is a type of swimmer.

<bird --> swimmer>. %0.90;0.45%'

// I guess swimmer is a type of bird.

<swimmer --> bird>. %1.00;0.42%'

NAL-1(Yes/No Question)

// Bird is a type of swimmer.

<bird --> swimmer>.

// Is bird a type of swimmer?

<bird --> swimmer>?

// Bird is a type of swimmer.

<bird --> swimmer>. %1.00;0.90%

NAL-1(Conversion)

// Bird is a type of swimmer.

<bird --> swimmer>.

// Is swimmer a type of bird?

<swimmer --> bird>?

// I guess swimmer is a type of bird.

<swimmer --> bird>. %1.00;0.47%

NAL-1(“What” Question)

// Bird is a type of swimmer.

<bird --> swimmer>. %1.00;0.80%

// What is a type of swimmer?

<?x --> swimmer>?

// Bird is a type of swimmer.

<bird --> swimmer>. %1.00;0.80%'

NAL-1(Choice Rule)

1. The confidence values of two candidates are same

Choose the one with higher frequency

2. The frequency values of two candidates are same

Choose the one with higher confidence

3. Either of frequency or confidence is same

Choose the one with higher expectation

Truth-value version: $e = c \times (f - 1/2) + 1/2$

NAL-2(Revision)

// Robin is similar to swan.

<robin <-> swan>.

// I think robin is not similar to swan.

<robin <-> swan>. %0.10;0.60%

// Robin is probably similar to swan.

<robin <-> swan>. %0.87;0.91%')

NAL-2(Comparison)

// Sport is a type of competition.

<sport --> competition>.

// Chess is a type of competition.

<chess --> competition>. %0.90%

// I guess chess is similar to sport.

<chess <-> sport>. %0.90;0.45%

NAL-2(Analogy)

// Swan is a type of swimmer.

<swan --> swimmer>.

// Gull is similar to swan.

<gull <-> swan>.

// I think gull is a type of swimmer.

<gull --> swimmer>. %1.00;0.81%'

NAL-2(Analogy)

// Gull is a type of swimmer.

<gull --> swimmer>.

// Gull is similar to a swan.

<gull <-> swan>.

// I believe a swan is a type of swimmer.

<swan --> swimmer>. %1.00;0.81%'

NAL-2(Analogy)

// Gull is a type of swimmer.

<gull --> swimmer>.

// Gull is similar to a swan.

<gull <-> swan>.

// I believe a swan is a type of swimmer.

<swan --> swimmer>. %1.00;0.81%'

NAL-2(Instance copula)

// Tweety is a bird.

<Tweety {-- bird>.

{Tweety} --> bird>. %1.00;0.90%

NAL-2(Instance copula)

// Ravens are black.

<raven --] black>.

<raven --> [black]>.

NAL-3(Compound Term)

A compound term (con C1, C2,…,Cn) is a term formed by term connector, con, that connects one or more terms C1,….,Cn (n >=1) call the component(s) of the compound

Example:

// Robin is a type of bird or type of swimmer

<robin --> (|,bird,swimmer)>.

// Robin is a type of bird and a type of swimmer

<robin --> (&,bird,swimmer)>.

// Robin is a nonswimming mammal.

<robin --> (-,mammal,swimmer)>.

NAL-3(Set Representation)

Definition 7.10. Given different terms $T_1, \ldots, T_n$ ($n \geq 2$), an extensional

set $\{T_1, \ldots, T_n\}$ is defined as $(\cup \{T_1\} \ldots \{T_n\})$; an intensional set $[T_1 \ldots T_n]$ is defined as $(\cap [T_1] \ldots [T_n])$.

NAL-3(Set operation)

// PlanetX is Mars, Pluto, or Venus.

<planetX --> {Mars,Pluto,Venus}>. %0.90%

// PlanetX is probably Pluto or Saturn.

<planetX --> {Pluto,Saturn}>. %0.70%

// PlanetX is Mars, Pluto, Saturn, or Venus.

<planetX --> {Mars,Pluto,Saturn,Venus}>. %0.97;0.81%'

// PlanetX is probably Pluto.

<planetX --> {Pluto}>. %0.63;0.81%'

NAL-3(Set operation)

// PlanetX is Mars, Pluto, or Venus.

<planetX --> {Mars,Pluto,Venus}>. %0.90%

// PlanetX is probably neither Pluto nor Saturn.

<planetX --> {Pluto,Saturn}>. %0.10%

// PlanetX is either Mars or Venus.

<planetX --> {Mars,Venus}>. %0.81;0.81%

NAL-4(Relation Terms)

// water dissolve salts

<(*, water, salt) --> dissolve>.

// water is something that dissolves salts

<water --> (/, dissolve, _ , salt)>.

// Salt is something that can be dissolved by water

<salt --> (/, dissolve, water, _)>.

NAL-4(Relation Terms)

// water dissolve salts

<rain --> water>.

// water is something that dissolves salts

<water --> (/, dissolve, _ , salt)>.

// Salt is something that can be dissolved by water

<(*, rain, salt) --> dissolve>.

NAL-4(Relation Terms)

// Neutralization is a relation between an acid and a base.

<neutralization --> (*,acid,base)>.

// Something that can neutralize a base is an acid.

<(\,neutralization,_,base) --> acid>. %1.00;0.90%

// Something that can be neutralized by an acid is a base.

<(\,neutralization,acid,_) --> base>. %1.00;0.90%

NAL-5(Statements as Terms)

// If robin can fly then robin is a type of bird.

<<robin --> [flying]> ==> <robin --> bird>>.

// If robin can fly then robin may not a type of bird.

<<robin --> [flying]> ==> <robin --> bird>>. %0.00;0.60%

// If robin can fly then robin is a type of bird.

<<robin --> [flying]> ==> <robin --> bird>>. %0.86;0.91%

NAL-5(Deduction)

// If robin is a type of bird then robin is a type of animal.

<<robin --> bird> ==> <robin --> animal>>.

// If robin can fly then robin is a type of bird.

<<robin --> [flying]> ==> <robin --> bird>>.

// If robin can fly then robin is a type of animal.

<<robin --> [flying]> ==> <robin --> animal>>. %1.00;0.81%

NAL-5(Detachment)

// If robin is a type of bird then robin can fly.

<<robin --> bird> ==> <robin --> animal>>.

// Robin is a type of bird.

<robin --> bird>.

// Robin is a type of animal.

<robin --> animal>. %1.00;0.81%

NAL-5(Analogy)

// If robin is a type of bird then robin is a type of animal.

<<robin --> bird> ==> <robin --> animal>>.

// Usually, robin is a type of bird if and only if robin can fly.

<<robin --> bird> <=> <robin --> [flying]>>. %0.80%

// If robin can fly then probably robin is a type of animal.

<<robin --> [flying]> ==> <robin --> animal>>. %0.80;0.65%

NAL-5(Compound composition)

// If robin is a type of bird then robin is a type of animal.

<<robin --> bird> ==> <robin --> animal>>.

// If robin can fly then robin is a type of animal.

<<robin --> [flying]> ==> <robin --> animal>>. %0.9%

// If robin can fly and is a type of bird then robin is a type of animal.

<(&&,<robin --> [flying]>,<robin --> bird>) ==> <robin --> animal>>. %1.00;0.81%

//If robin can fly or is a type of bird then robin is a type of animal.

<(||,<robin --> [flying]>,<robin --> bird>) ==> <robin --> animal>>. %0.90;0.81%

NAL-5(Compound composition)

// If robin is a type of bird then robin is a type of animal.

<<robin --> bird> ==> <robin --> animal>>.

// If robin can fly then robin is a type of animal.

<<robin --> [flying]> ==> <robin --> animal>>. %0.9%

// If robin can fly and is a type of bird then robin is a type of animal.

<(&&,<robin --> [flying]>,<robin --> bird>) ==> <robin --> animal>>.
%1.00;0.81%

//If robin can fly or is a type of bird then robin is a type of animal.

<(||,<robin --> [flying]>,<robin --> bird>) ==> <robin --> animal>>. %0.90;0.81%

NAL-6(Variable)

```
\\ Every key opens every lock.

<(&&, <{$x} --> key>, <{$y} --> lock>) ==> <(*, $x, $y) --> open>>.

\\ There is a key that opens every lock

<(&&, <{#x} --> key>, <{$y} --> lock>) ==> <(*, #x, $y) --> open>>.

\\ Every key opens some lock (that depends on the key).

<(&&, <{$x} --> key>, <{#y} --> lock>) ==> <(*, $x, #y) --> open>>.

\\There is a key that opens a lock.

<(&&, <{#x} --> key>, <{#y} --> lock>) ==> <(*, #x, #y) --> open>>.
```

NAL-6(Variable Unification)

// If something is a bird, then it is a flyer.

<<$x --> bird> ==> <$x --> flyer>>.

// If something is a bird, then it is not a flyer.

<<$y --> bird> ==> <$y --> flyer>>. %0.00;0.70%

// If something is a bird, then usually, it is a flyer.

<<$1 --> bird> ==> <$1 --> flyer>>. %0.79;0.92%'

NAL-6(Variable Unification)

// If something is a swan, then it is a bird.

```
<<$x --> swan> ==> <$x --> bird>>. %1.00;0.80%
```

// If something is a swan, then it is a swimmer.

```
<<$y --> swan> ==> <$y --> swimmer>>. %0.80%
```

NAL-6(Variable Unification)

// I believe that if something is a swan, then it is a bird or a swimmer.

```
<<$1 --> swan> ==> (||,<$1 --> bird>,<$1 --> swimmer>)>. %1.00;0.72%
```

// I believe that if something is a swan, then usually, it is both a bird and a swimmer.

```
<<$1 --> swan> ==> (&&,<$1 --> bird>,<$1 --> swimmer>)>. %0.80;0.72%
```

// I guess if something is a swimmer, then it is a bird.

```
<<$1 --> swimmer> ==> <$1 --> bird>>. %1.00;0.37%
```

// I guess if something is a bird, then it is a swimmer.

```
<<$1 --> bird> ==> <$1 --> swimmer>>. %0.80;0.42%
```

NAL-6(Variable Elimination)

\\ Tweety has wings.

<{Tweety} --> [with-wings]>.

\\ If something can chirp and has wings, then it is a bird.

<(&&,<$x --> [chirping]>,<$x --> [with-wings]>) ==> <$x --> bird>>.

\\ If Tweety can chirp, then it is a bird.

<<{Tweety} --> [chirping]> ==> <{Tweety} --> bird>>. %1.00;0.81%')

NAL-6(Second level variable unification)

// There is a lock which is opened by all keys

(&&,<#1 --> lock>,<<$2 --> key> ==> <#1 --> (/,open,$2,_)>>). %1.00;0.90%

// key1 is a key

<{key1} --> key>. %1.00;0.90%

// there is a lock which is opened by key1

(&&,<#1 --> (/,open,{key1},_)>,<#1 --> lock>). %1.00;0.81%'

NAL-7(Events as Statements)

An event is a statement with a time-dependent truthvalue, that is, the evidential support summarized in its truth-value is valid only in its duration, which is a certain period of time between the moment the event starts and the moment it ends

NAL-7(Events as Statements)

The temporal relation between two atomic events E1 and E2 has the following three basic cases.

1. E1 happens before E2 happens

<E1 =/> E2>.

2. E1 happens after E2 happens

<E1 =\> E2>.

3. E1 happens when E2 happens

<E1 =|> E2>.

NAL-7(Tense for single Event)

// John entered room 101

<(*,John,room_101) --> enter>. :\:

// John is entering room 101

<(*,John,room_101) --> enter>. :|:

// John will enter room 101

<(*,John,room_101) --> enter>. :/

:

NAL-7(Tense for single Event)

// John entered room 101

<(*,John,room_101) --> enter>. :\:

// John is entering room 101

<(*,John,room_101) --> enter>. :|:

// John will enter room 101

<(*,John,room_101) --> enter>. :/

:

NAL-7(Temporal deduction/explification)

// Someone needs to open door 101 before entering room 101

<<(*, $x, room_101) --> enter> =\> <(*, $x, door_101) --> open>>. %0.9%

// Someone needs to hold the key 101 before opening door 101

<<(*, $y, door_101) --> open> =\> <(*, $y, key_101) --> hold>>. %0.8%

// Someone needs to hold the key 101 before entering room 101

<<(*,$1,room_101) --> enter> =\> <(*,$1,key_101) --> hold>>. %0.72;0.58%

// Someone needs to hold the key 101 before entering room 101

<<(*,$1,key_101) --> hold> =/> <(*,$1,room_101) --> enter>>. %1.00;0.37%

NAL-7(Inference on Tense)

// Before John entered room 101, John hold the key 101

```
<<(*,John,key_101) --> hold> =/> <(*,John,room_101) --> enter>>.
```

// John is holding the key 101

```
<(*,John,key_101) --> hold>. :|:
```

// John will enter room 101

```
<(*,John,room_101) --> enter>. :/: %1.00;0.81%'
```

NAL-8(Operations and Goals as Events)

1. An operation of NARS is an event that the system can actualize - it is true when

a corresponding procedure is executed by the system

2. A goal is a sentence containing an event that the system

desires to realize.

NAL-8(Operations and Goals as Events)

1. An operation of NARS is an event that the system can actualize - it is true when

a corresponding procedure is executed by the system

2. A goal is a sentence containing an event that the system

desires to realize.

NAL-8(Toothbrush)

```
// Toothbrush is made of plastic

<(*,toothbrush,plastic) --> made_of>.

// If something is made of plastic and we use the lighter to it, this thing will be

// heated

<(&/,<(*,$1,plastic) --> made_of>,(^lighter,{SELF},$1)) =/> <$1 --> [heated]>>.

// If something is heated then this thing will be melted

<<$1 --> [heated]> =/> <$1 --> [melted]>>.

// If something is melted, at the same time it becomes pliable

<<$1 --> [melted]> <|> <$1 --> [pliable]>>.
```

NAL-8(Toothbrush)

```
// If I reshape something pliable, it will be hardened

<(&/,<$1 --> [pliable]>,(^reshape,{SELF},$1)) =/> <$1 --> [hardened]>>.

// If something is hardened it becomes unscrewing

<<$1 --> [hardened]> =|> <$1 --> [unscrewing]>>.

// Toothbrush is an object

<toothbrush --> object>.

// I want an unscrewing object

(&&,<#1 --> object>,<#1 --> [unscrewing]>)!

10000
```

NAL-8(Toothbrush)

// I have a goal which is to use lighter to the toothbrush

(^lighter,{SELF},toothbrush)! %1.00;0.18%

// I have another goal which is to reshape the toothbrush

(^reshape,{SELF},toothbrush)! %1.00;0.06%

Emotion

//If something is wanted by SELF, and SELF anticipates the
//opposite to happen, SELF feels fear

Input:<(&&, (^want, {SELF}, #1, FALSE), (^anticipate, {SELF}, #1)) =|> (^feel, {SELF}, fear)>.

//At the same time when SELF feels fear, it generate an
//motivation which to run away, run is also an operator in NARS

Input: <(^feel,{SELF}, fear) =|> <(*, {SELF}, <(*, {SELF}) --> ^run>) --> ^want>>.

//SELF doesn't want to be hurt

Input: (--,<{SELF} --> hurt>)!

//If wolf is getting close to SELF, SELF will get hurt

Input: <(&/,<(*, {SELF}, wolf) --> close_to>,+42) =/> <{SELF} --> [hurt]>>.

Emotion

//Wolf is getting close to self

Input: <(*, {SELF}, wolf) --> close_to>. :|:

//Result: SELF takes the action run, based on the knowledge
//where SELF runs when it feels fear, SELF also feels the emotion
//fear

EXECUTE (^run,{SELF})