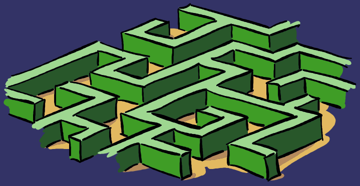
NARS

An Artificial General Intelligence Project

Pei Wang

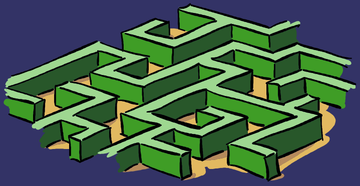
&

The OpenNARS Team



“Intelligence” Interpreted

- Mainstream AI treats “Intelligence” as a collection of problem-specific and domain-specific functions
- AGI takes “Intelligence” as a general-purpose capability that should be treated as a whole
- AGI research still includes different research objectives

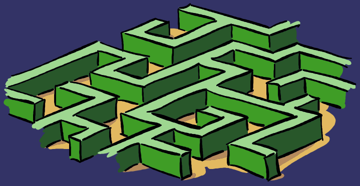


Basic Assumption

"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

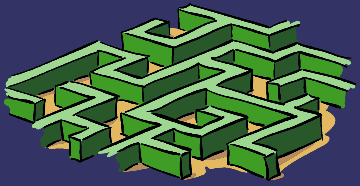


Reasoning System Framework

- 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
- flexibility

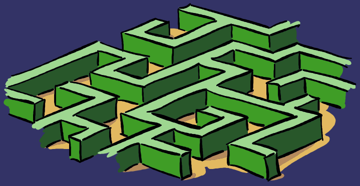


Fundamental Issue

Under AIKR, the system cannot guarantee absolute correctness or optimum anymore. Now what is the standard of *validity* or *rationality*?

Validity and rationality become *relative* to the available knowledge and resources.

Desired features: *general, adaptive, flexible, robust, scalable*

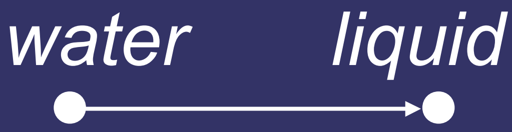


Knowledge Representation

Term: word, as name of a concept

Statement: subject-copula-predicate

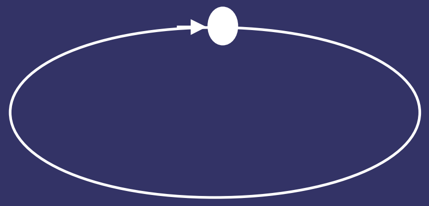
S → P

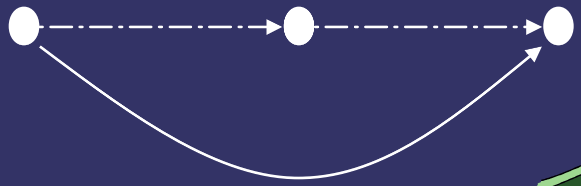


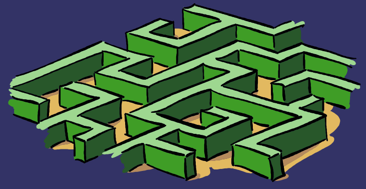
water *liquid*

as specialization-generalization

Copula *inheritance* is reflexive and transitive







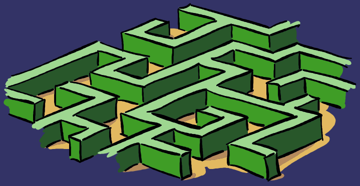
Binary Truth-value

- Experience ***K***: a finite set of statements
- Beliefs ***K****: the transitive closure of ***K***
- A statement is *true* if

  either it is in ***K****

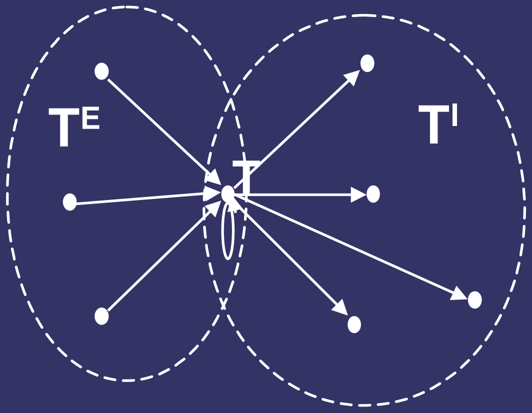
  or it has the form of X → X

  otherwise it is *false*

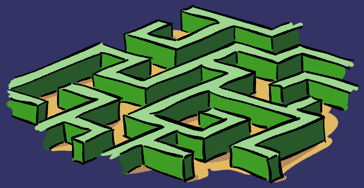


Extension and Intension

For a given term T,
its *extension* $T^E = \{x \mid x \rightarrow T\}$
its *intension* $T^I = \{x \mid T \rightarrow x\}$



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



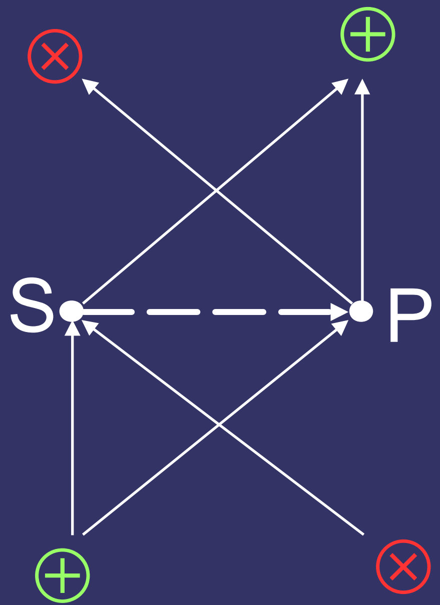
Evidence

Positive evidence of $S \rightarrow P$:

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

Negative evidence of $S \rightarrow P$:

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

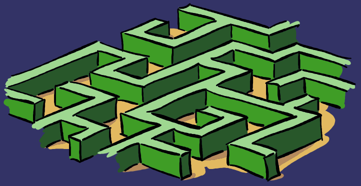


Amount of evidence:

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

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

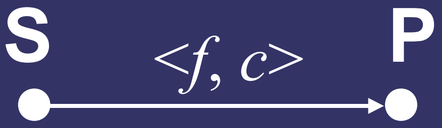
total: $w = w^+ + w^- = |S^E| + |P^I|$



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.

$S \rightarrow P <f, c>$



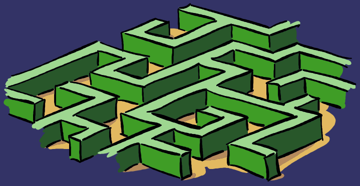
S

$<f, c>$

P

frequency: $f = w^+/w$

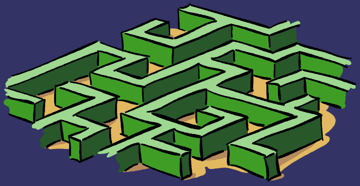
confidence: $c = w / (w +1)$



The Semantics of NARS

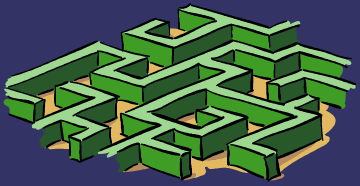
NARS has an experience-grounded semantics:

- The truth-value of a statement is defined by its evidence, not by a fact
- The meaning of a term is defined by its extension and intension, not by a reference



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



Truth-value Function Design

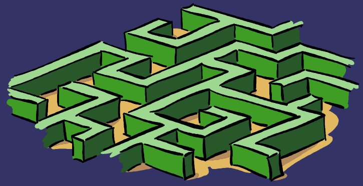
1. Treat all involved variables as Boolean

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:

$$\begin{aligned} \mathrm{not}(x) &= 1 - x \\ \mathrm{and}(x, y) &= x * y \\ \mathrm{or}(x, y) &= 1 - (1 - x) * (1 - y) \end{aligned}$$



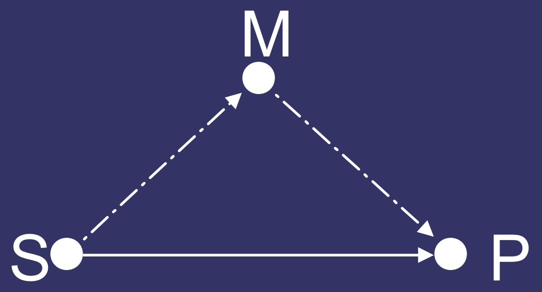
Deduction

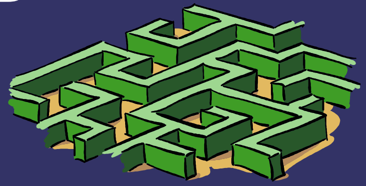
$$\begin{array}{c} M \to P\ [f_1, c_1] \\ S \to M\ [f_2, c_2] \\ \hline S \to P\ [f, c] \end{array}$$

$$\begin{array}{c} \text{bird} \to \text{animal}\ [1.00, 0.90] \\ \text{robin} \to \text{bird}\quad [1.00, 0.90] \\ \hline \text{robin} \to \text{animal}\ [1.00, 0.81] \end{array}$$

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

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





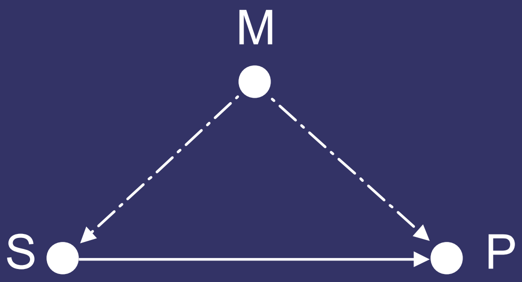
Induction

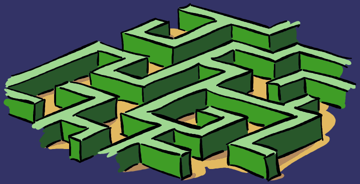
$$\frac{\begin{array}{c} M \rightarrow P\ [f_1, c_1] \\ M \rightarrow S\ [f_2, c_2] \end{array}}{S \rightarrow P\ [f, c]}$$

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

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

$$\frac{\begin{array}{ll} \text{Swan} \rightarrow \text{bird} & [1.00, 0.90] \\ \text{swan} \rightarrow \text{swimmer} & [1.00, 0.90] \end{array}}{\text{bird} \rightarrow \text{swimmer}\ [1.00, 0.45]}$$





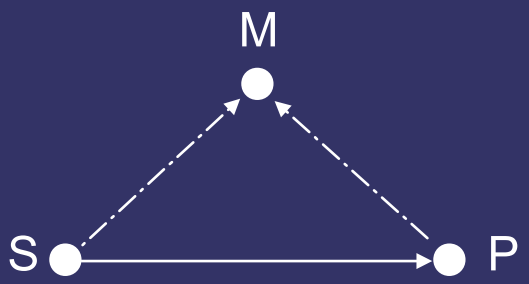
Abduction

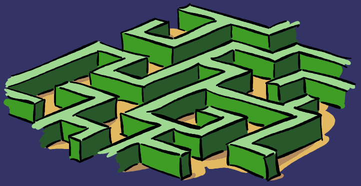
$$\begin{array}{c} P \rightarrow M\ [f_1, c_1] \\ S \rightarrow M\ [f_2, c_2] \\ \hline S \rightarrow P\ [f, c] \end{array}$$

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

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

$$\begin{array}{r} \text{seabird} \rightarrow \text{swimmer}\ [1.00, 0.90] \\ \text{gull} \rightarrow \text{swimmer}\ [1.00, 0.90] \\ \hline \text{gull} \rightarrow \text{seabird}\quad [1.00, 0.45] \end{array}$$



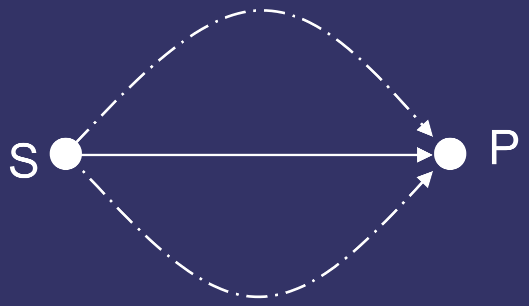


Revision

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

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

$$\boldsymbol{w} = w_1 + w_2$$

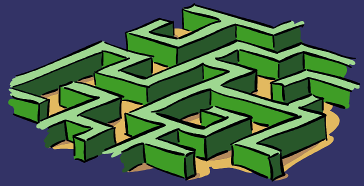


bird → swimmer [1.00, 0.62]

bird → swimmer [0.00, 0.45]

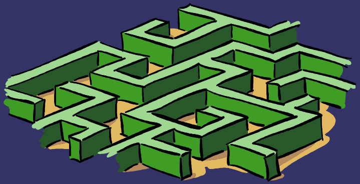
---

bird → swimmer [0.67, 0.71]



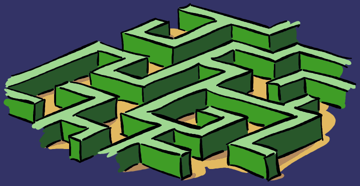
Types of Inference Rules

- **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

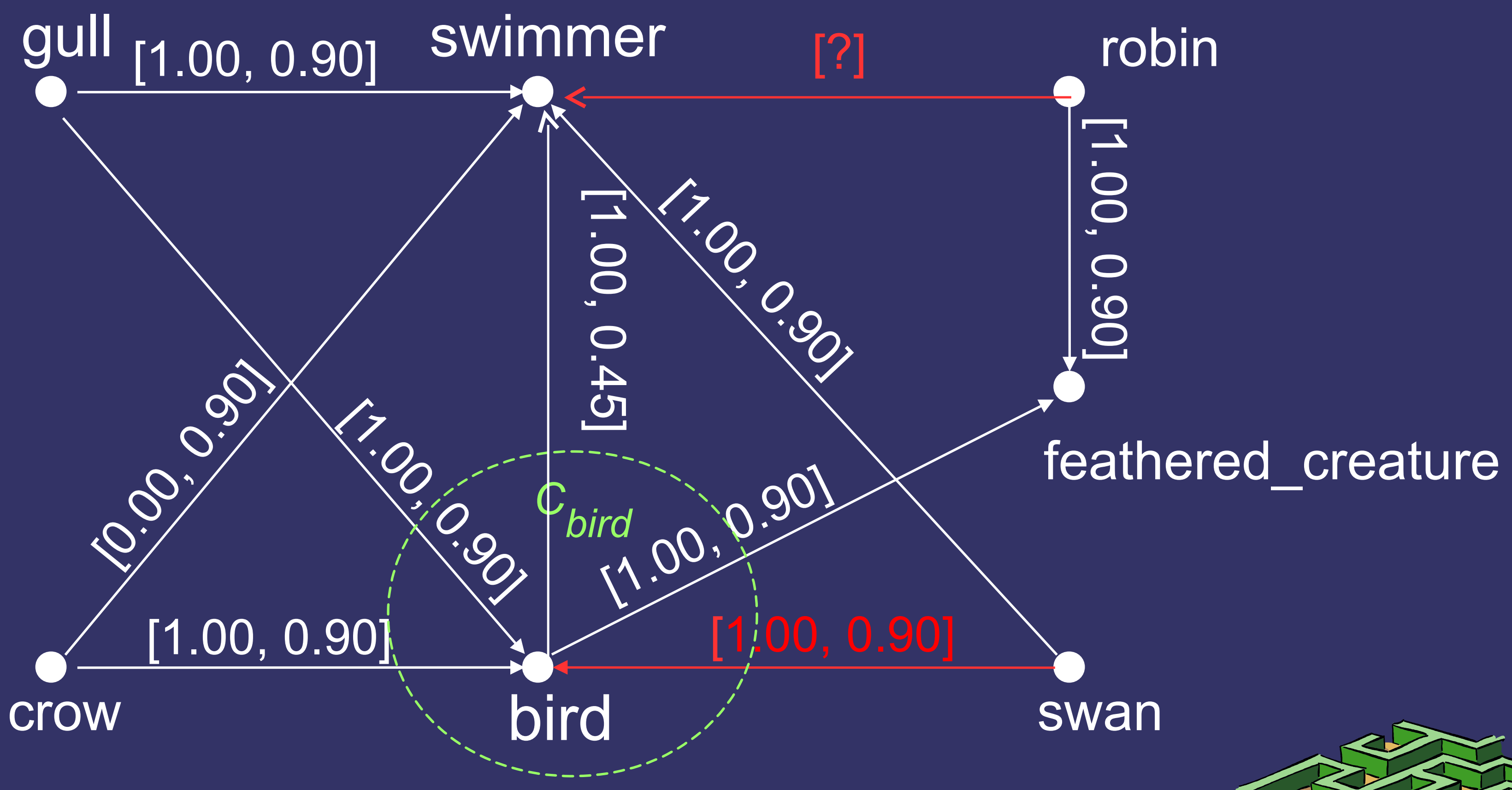


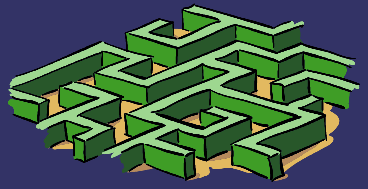
Memory Structure

- 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

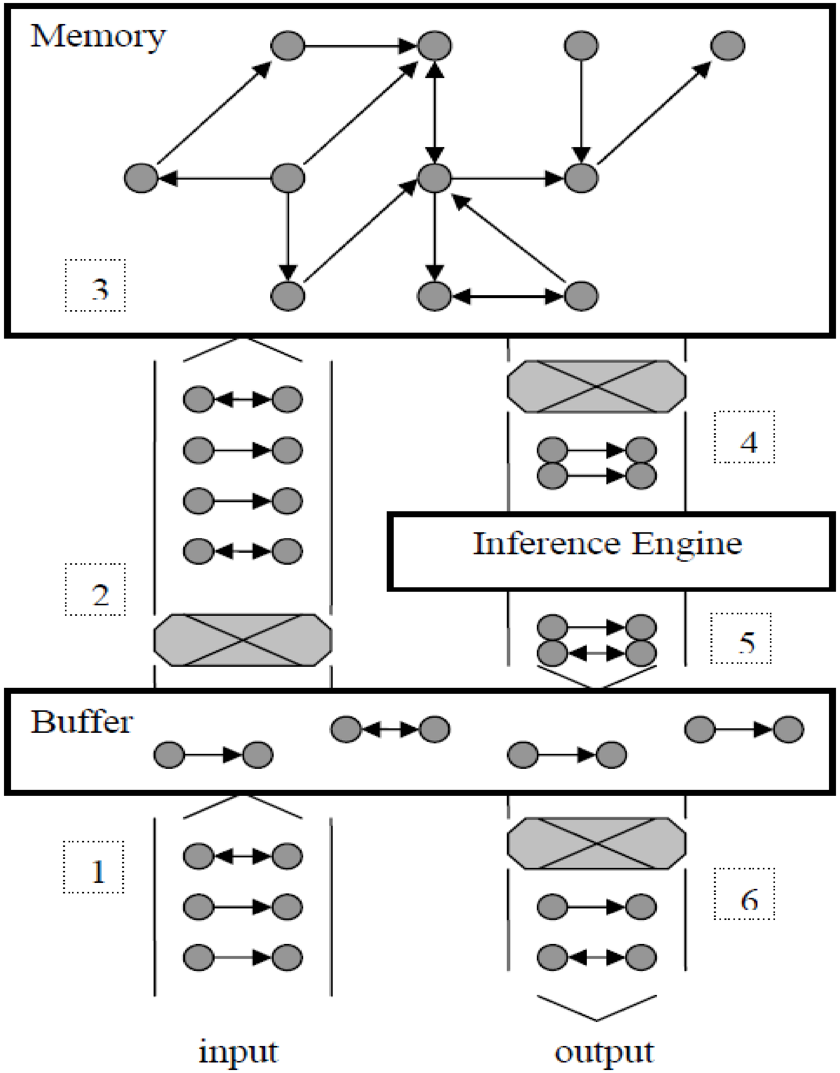


Memory as a Network





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.

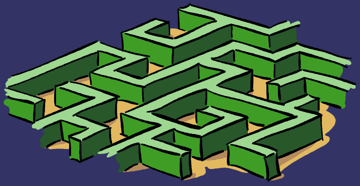
5. The conclusions derived from the premises by applicable rules are added into the buffer as derived tasks.

6. Selected derived tasks are reported as output tasks.

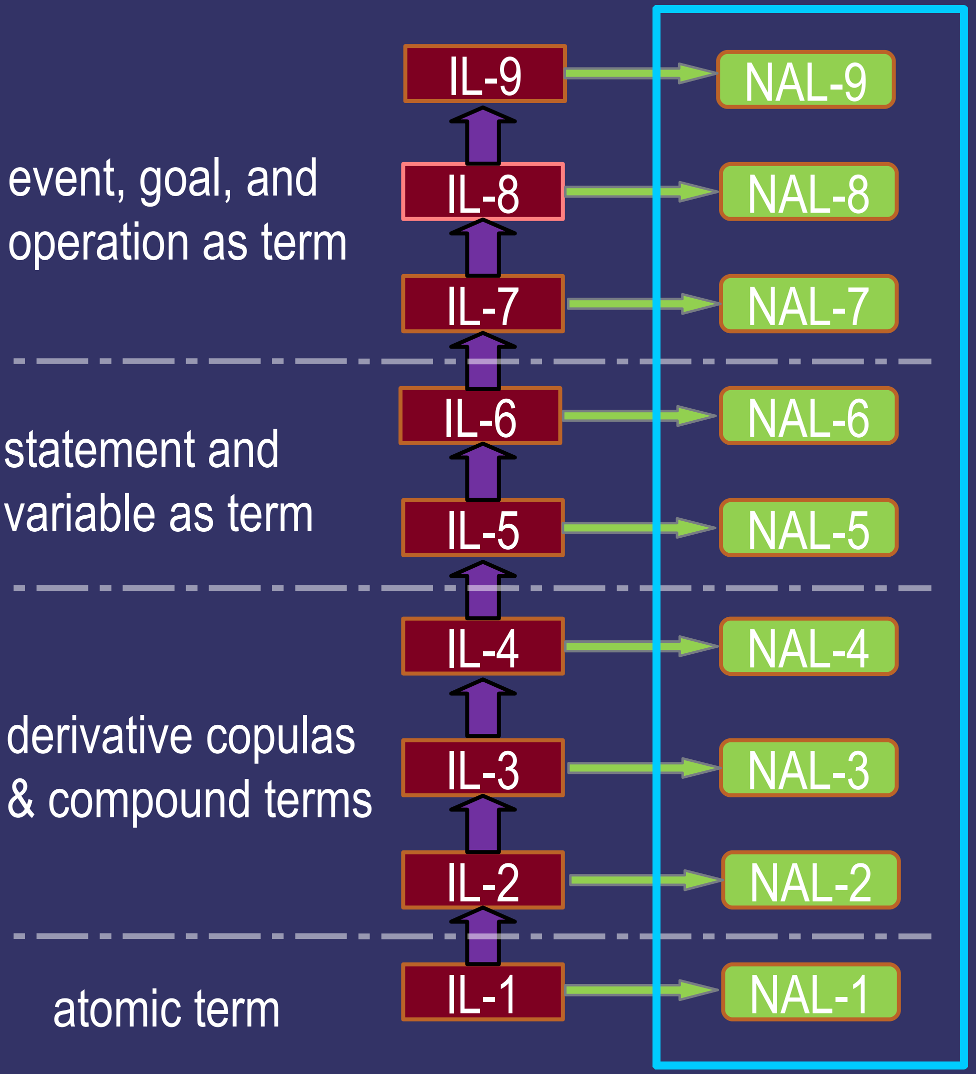
Control Strategy

- 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



The Layers of the Logic



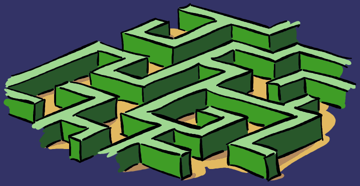
event, goal, and
operation as term

statement and
variable as term

implementation

derivative copulas
& compound terms

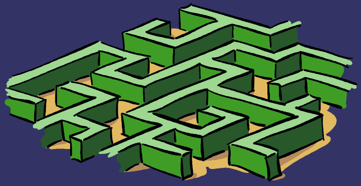
atomic term



Copulas & Compound Terms

Ideas from set theory:

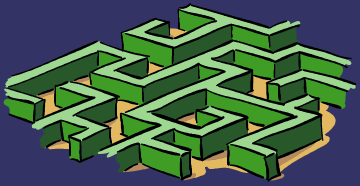
- Variants of the *inheritance* copula: *similarity*, *instance*, and *property*
- Compound terms: *sets*, *intersections*, *differences*, *products*, and *images*
- New inference rules for *comparison*, *analogy*, plus compound-term *composition* and *decomposition*



Model of Concept

Every concept in NARS is *fluid*:

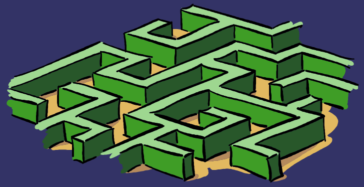
- Its meaning is determined neither by reference nor definition, but by experienced relations
- Each relation is a matter of degree
- Meaning changes by history and context



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



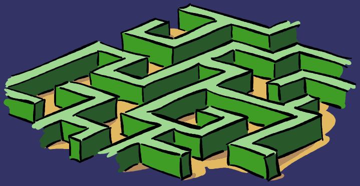
NAL as a Meta-logic

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

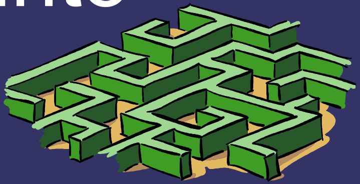
Mathematical reasoning: local axiomatic subsystem



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

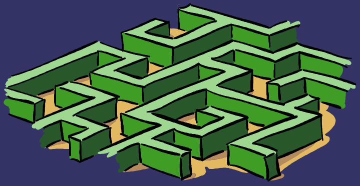
From question-answering to goal-achieving

Causal inference, prediction, explanation

Planning, skill acquisition, self-programming

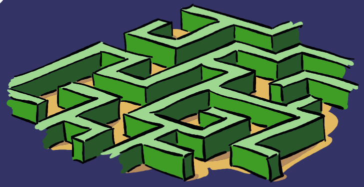
Self-awareness and self-control

Emotion and feeling



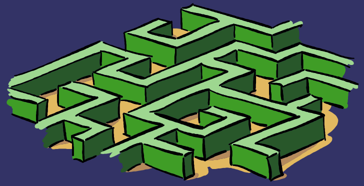
Unifications in NARS

- 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



Implementation

- NARS has been mostly implemented in the open-source project OpenNARS
- Working examples exist as proof of concept, though only cover simple cases
- The system shows many human-like properties, though it is not a psychological model

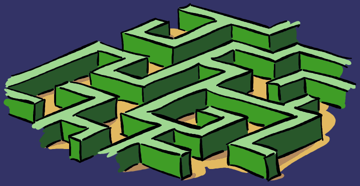


Potential Applications

NARS is not designed for any specific application, though some of its components can be customized for practical applications

Suitable domains:

- AIKR is applicable
- Tasks expressible as reasoning
- Tools have compatible interface



Publications & reports:
http://www.cis.temple.edu/~pwang/

Participations and cooperations are welcome!