Artificial General Intelligence

4. Dynamic Resource Allocation

Dr. Pei Wang

Dr. Patrick Hammer

Dr. Robert Johansson

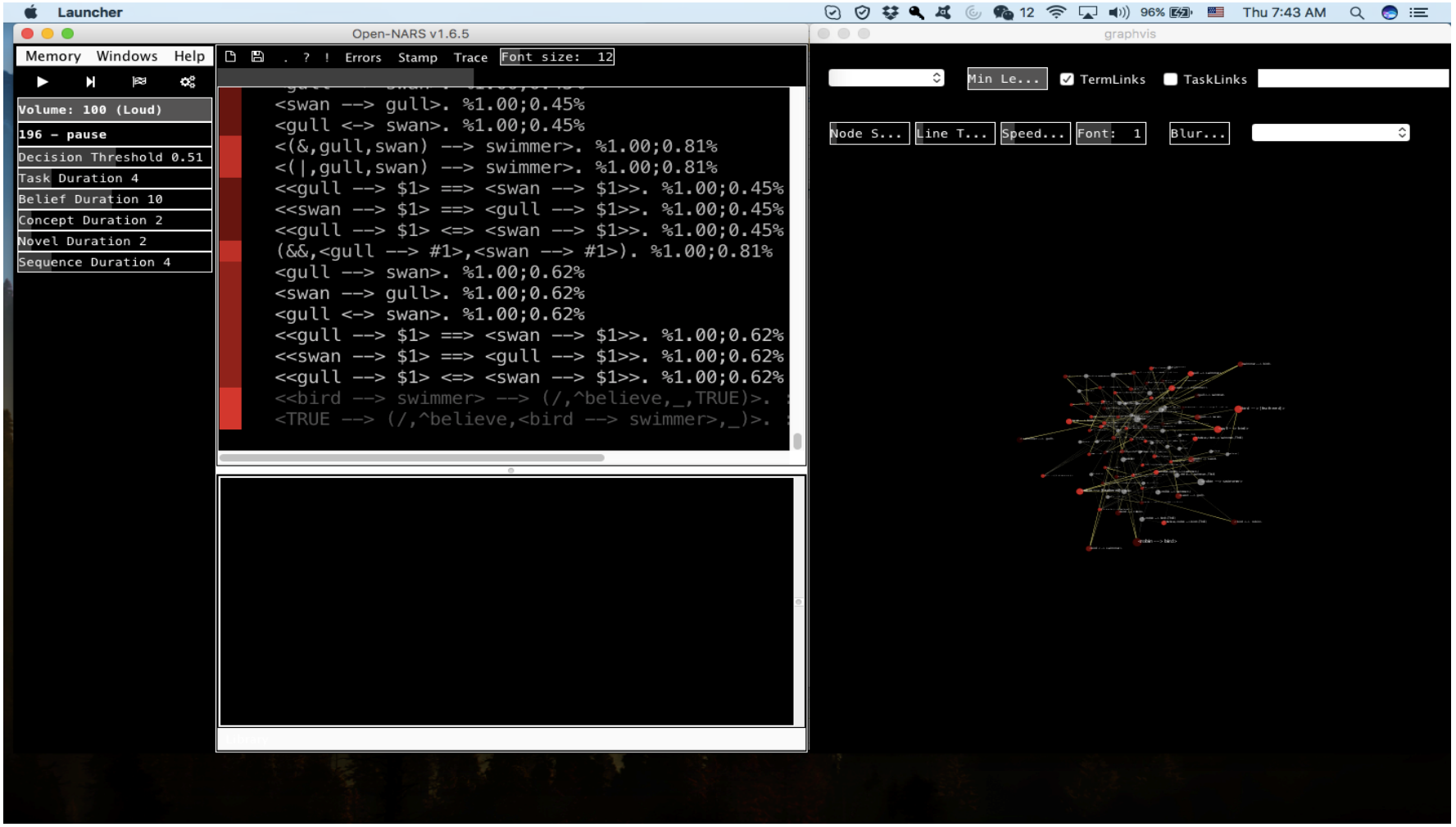
Digital Futures (KTH, SU, RISE)

Stockholm, Fall 2022

From NAL to NARS

- NARS (Non-Axiomatic Reasoning System) realizes adaptation under AIKR
- NARS has code for the grammar rules of Narsese and inference rules of NAL, plus a memory structure, a control mechanism, etc.
- NARS has been extended and refined over 3 decades (open-source since 2008), with around 50 versions in several programming languages

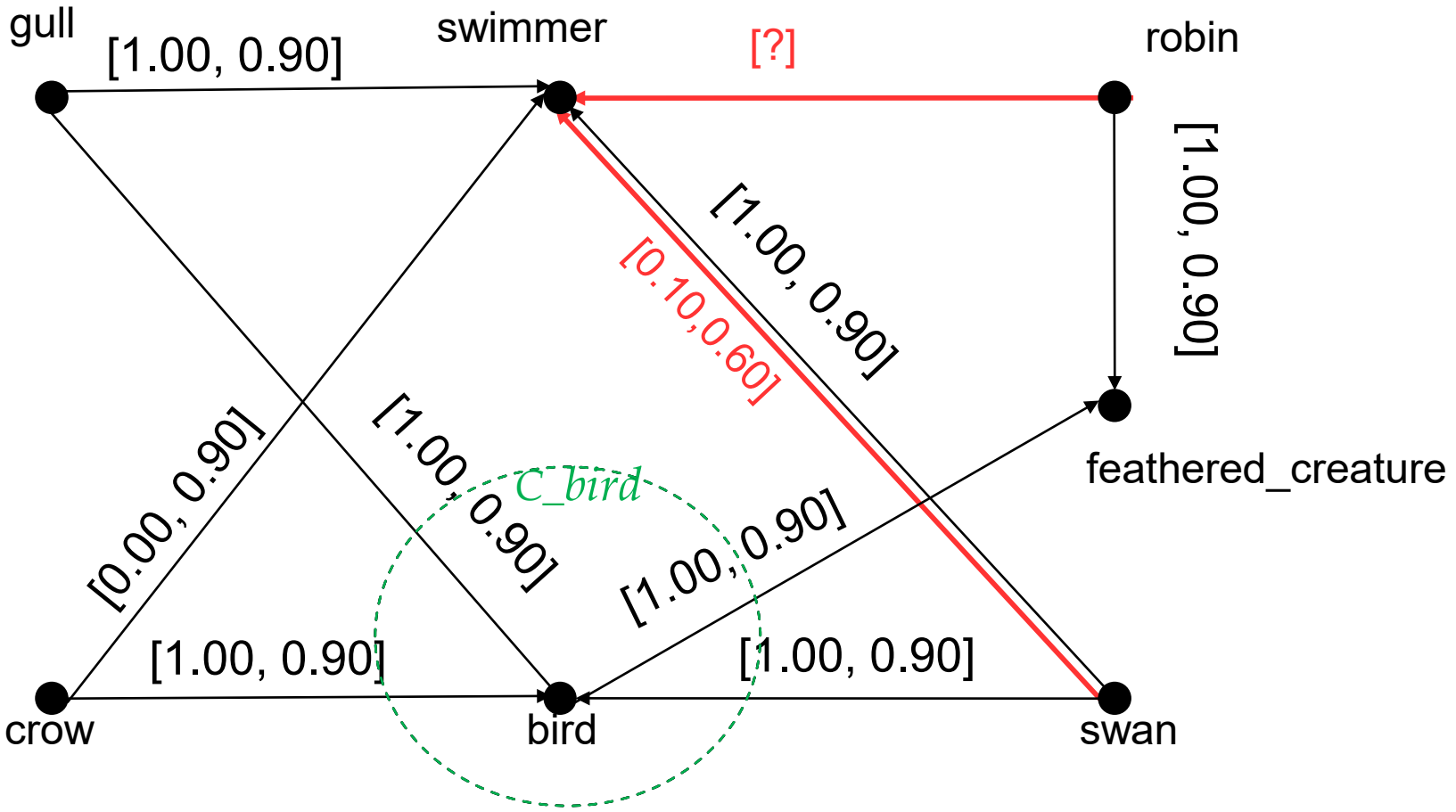
OpenNARS


Launcher
Thu 7:43 AM
Open-NARS v1.6.5
Memory Windows Help
Errors Stamp Trace Font size: 12
Volume: 100 (Loud)
196 - pause
Decision Threshold 0.51
Task Duration 4
Belief Duration 10
Concept Duration 2
Novel Duration 2
Sequence Duration 4
<swan --> gull>. %1.00;0.45%
<gull <-> swan>. %1.00;0.45%
<(&,gull,swan) --> swimmer>. %1.00;0.81%
<(|,gull,swan) --> swimmer>. %1.00;0.81%
<<gull --> $1> ==> <swan --> $1>>. %1.00;0.45%
<<swan --> $1> ==> <gull --> $1>>. %1.00;0.45%
<<gull --> $1> <=> <swan --> $1>>. %1.00;0.45%
(&&,<gull --> #1>,<swan --> #1>). %1.00;0.81%
<gull --> swan>. %1.00;0.62%
<swan --> gull>. %1.00;0.62%
<gull <-> swan>. %1.00;0.62%
<<gull --> $1> ==> <swan --> $1>>. %1.00;0.62%
<<swan --> $1> ==> <gull --> $1>>. %1.00;0.62%
<<gull --> $1> <=> <swan --> $1>>. %1.00;0.62%
<<bird --> swimmer> --> (/,^believe,_,TRUE)>.
<TRUE --> (/,^believe,<bird --> swimmer>,_)>.
graphvis
Min Le...
TermLinks
TaskLinks
Node S...
Line T...
Speed...
Font: 1
Blur...

To implement NAL-1

- NAL-1 allows NARS to answer simple questions
- Each input sentence is taken as a *task* to be processed using the system's *beliefs* (knowledge)
- A *judgment* task can be turned into a belief, revise the existing beliefs, as well as derive new judgments via forward inference
- A *question* task can be answered by a belief, as well as derive new questions via backward inference then get answers via forward inference

Tasks, beliefs, and concepts



Task processing

- Each task is processed by interacting with certain beliefs in the same concept, step by step
- In each step, the selected task and belief will decide the applicable inference rules
- So the processing of a task (a.k.a. the solving of a problem) is fully determined by the beliefs selected for it, as well as by the selection order
- Logic decides possibilities, and control turns certain possibilities into reality

The challenge of AIKR

- For a novel task, the system may not have a predetermined algorithm
- The system usually does not have the time to exhaustively use all beliefs on each task
- The situation changes unpredictably, so the system cannot depend on planning in advance
- Random and arbitrary selections are not adaptive
- NARS attempts to achieve all tasks as much as possible, according to its experience

Controlled concurrency

- NARS processes tasks concurrently by time-sharing with inference steps as unit
- Priority-based resource allocation happens among tasks through probabilistic selection
- For a selected task, the beliefs used to process it are also selected probabilistically, biased by their priority values
- Repeated probabilistic selections implement concurrent processing with different speed

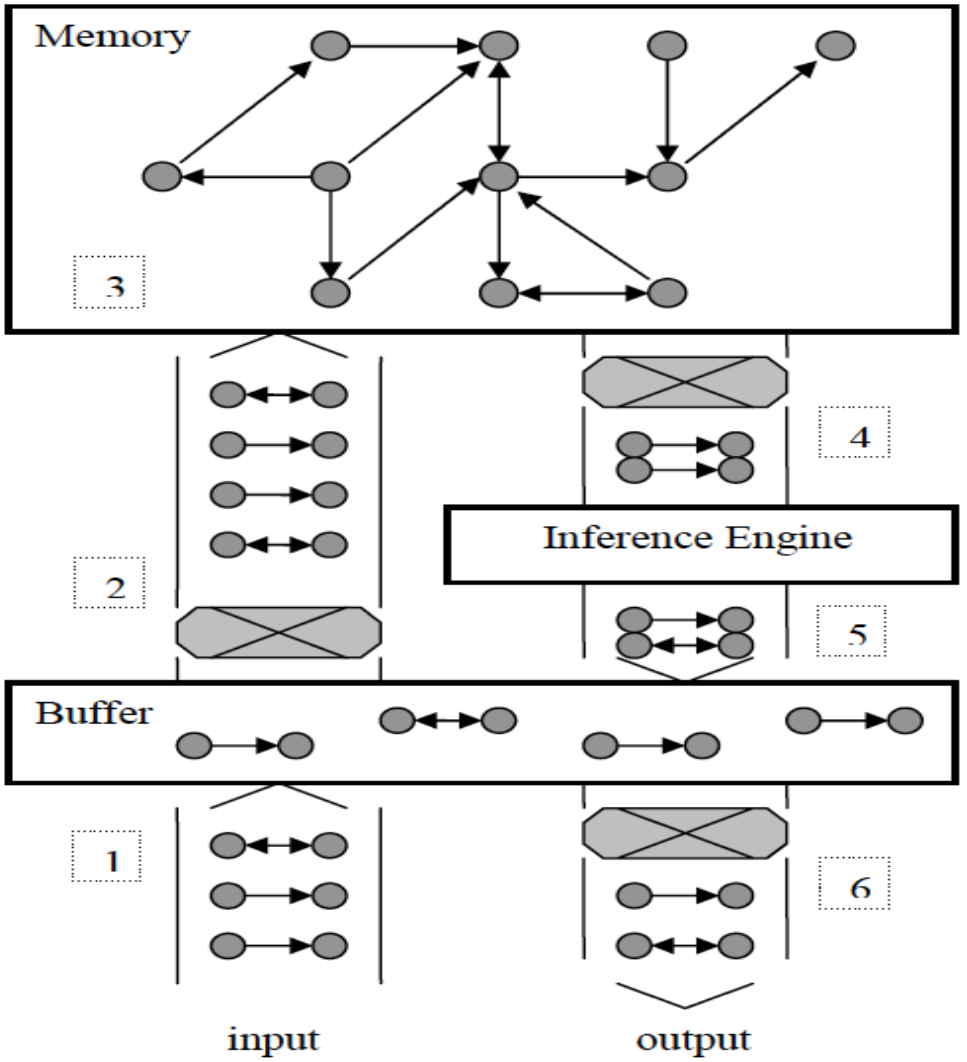
Priority values

- The *priority* of a data item (task, belief, concept) indicates its rank in resource competition
- A priority value summarizes factors like *urgency* (for task), *usefulness* (for belief), *relevance* (for concept), and so on
- Each priority value decays until reaching a level determined by the innate *quality* of the item
- Low priority items are removed from memory when there is a shortage of space

Major data structures and parts

- **Bag**: a constant-size probabilistic priority-queue combined with a hash-table, with functions like *put(key, priority)*, *get(key)*, *select()*, etc.
- **Concept**: a storage-processing unit identified by a term, with a task bag and a belief bag
- **Memory**: a concept bag
- **NARS**: a memory, an inference engine, a task buffer, plus a user interface

Architecture and working cycle



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.

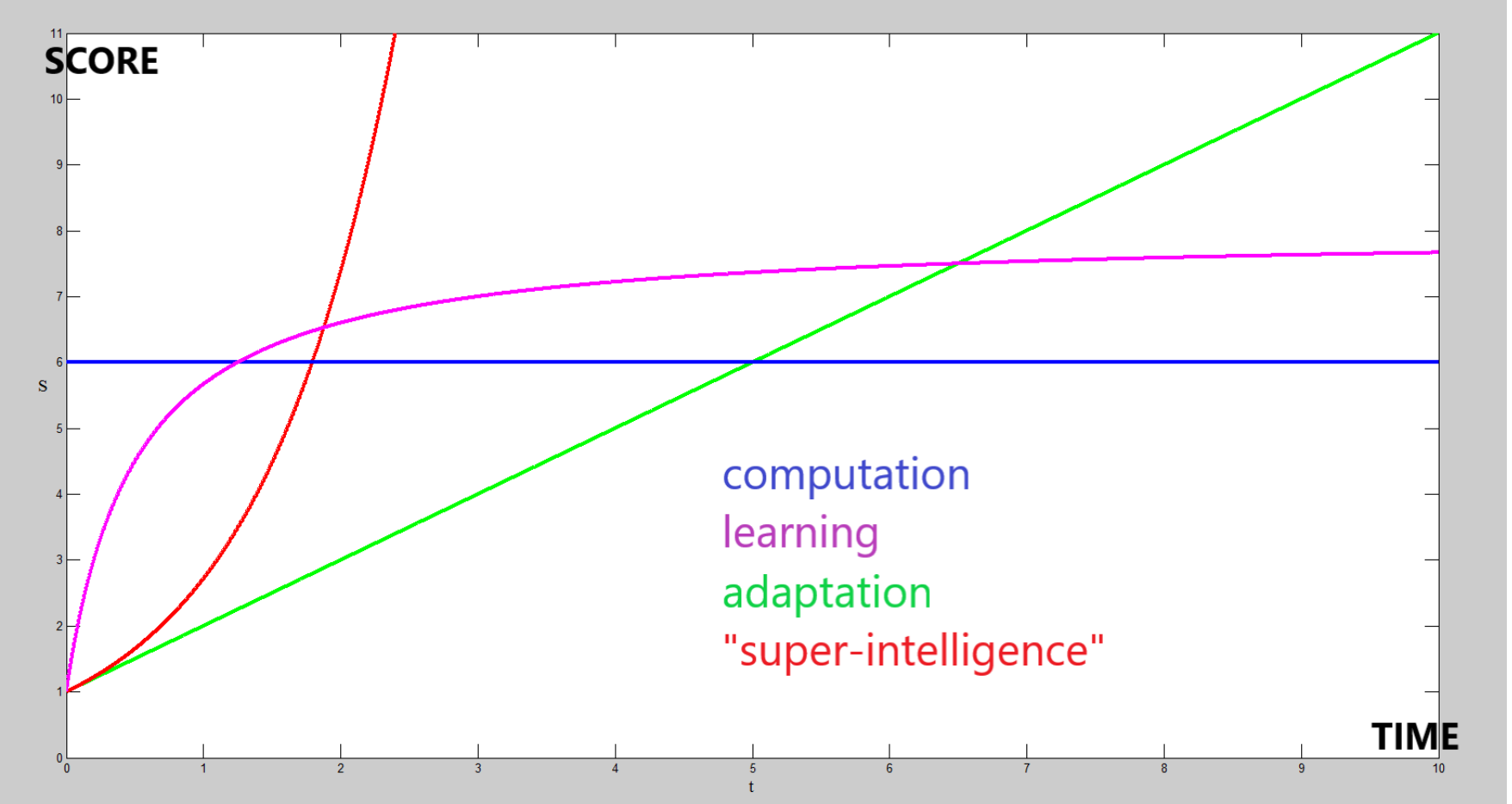
5. The conclusions derived from the premises by applicable rules are added into the buffer as derived tasks.

6. Selected derived tasks are reported as output tasks.

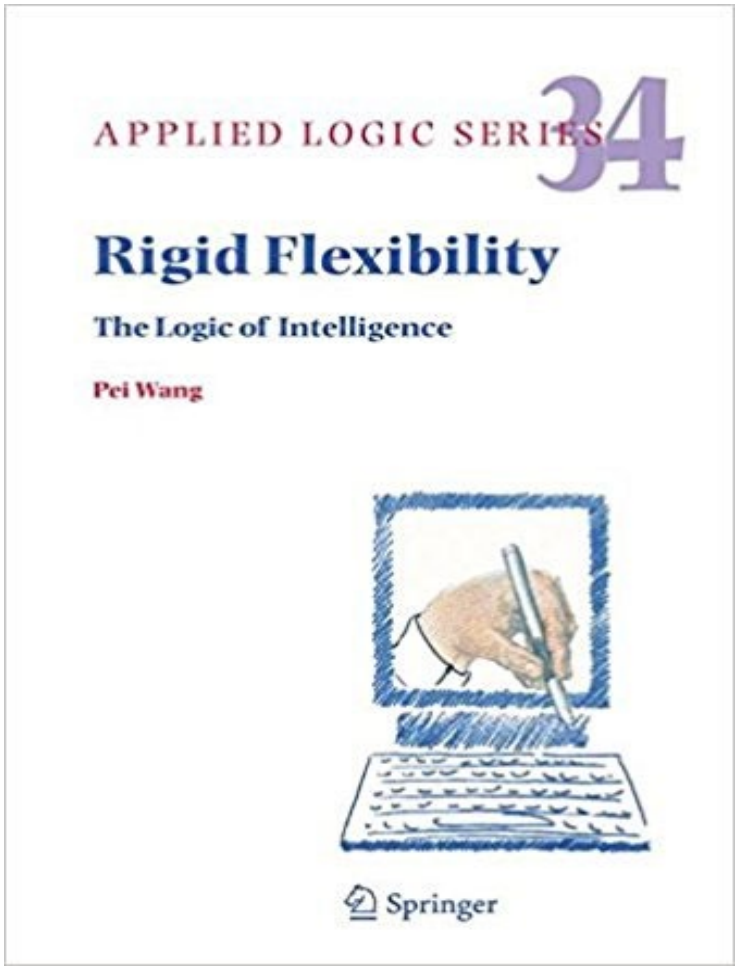
Novel properties

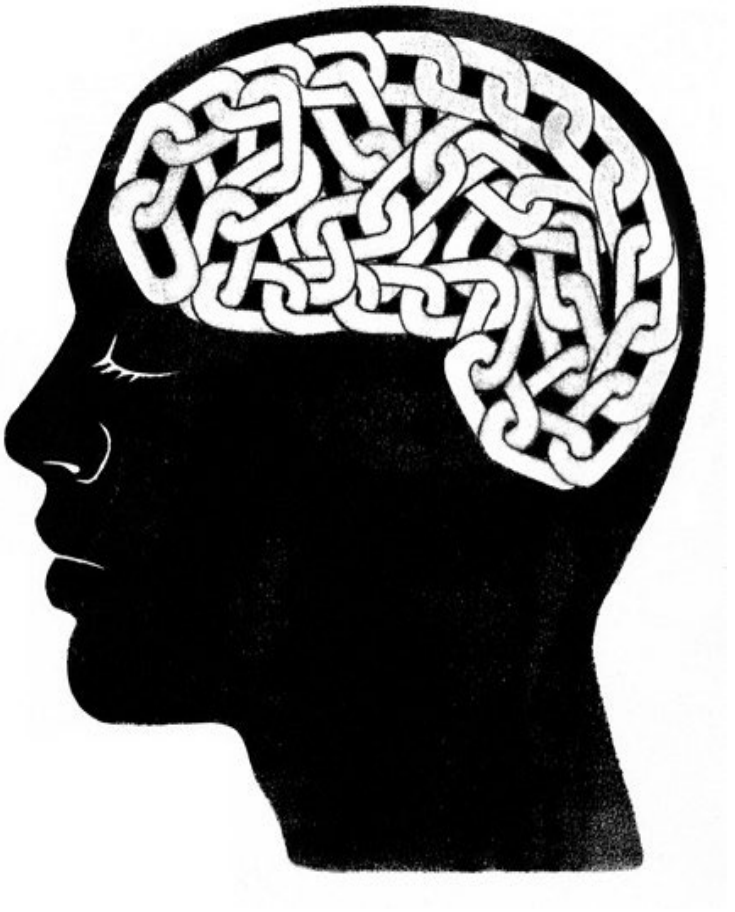
- NARS treats each task in a case-by-case manner, rather than follows a predetermined algorithm
- The processing path of a task is determined by the beliefs selected to interact with it, so is data-driven, context-sensitive, and unrepeatable
- A task is paused or terminated for resource reasons, not according to its achieving-level
- Traditional computability and complexity analysis are no longer applicable at the task level

Compared to the traditional models



Rigid flexibility





Suggested Readings

- Allen Newell, Herbert Simon, Computer Science as Empirical Inquiry: Symbols and Search
- Paul Smolensky, On the proper treatment of connectionism
- Peter Kugel, Thinking may be more than computing
- Pei Wang, Non-Axiomatic Logic, Chapter 5
- Pei Wang, Rigid Flexibility, Chapter 6, 12