NAL / NARchy

UNSTABLE - USE AT YOUR OWN RISK

Terms

Terms are the fundamental unit of data representation. They can be classified into a few categories, which resemble Prolog's symbolic semantics:

- **Atom** - indivisible symbols
- **Variable** - can unify with other Terms, depending on the variable type (**# $? %**)
- **Compound** - consists of a tree/DAG of **subterms** connected by **operators**

Examples of NAL Terms

```
a

(b --> $c)

e(#d)

((f ==> g) <-> (?h,i,j))
```

Term Normalization and Validity

Term construction is valid within the syntactic assumptions of the operators involved -- and after recursive application of algebraic reductions and contained evaluated functors.

A term eventually becomes **normalized** before use in inference processes.

- **Variable normalization** of compound terms containing variables involves the anonymized rewrite of the unique variables in a canonically ordered sequence according to their first appearance in the compound.
- Not all terms are valid, particularly when logical relations are involved. In such cases, construction attempts result in the **Null** term.

Concepts

Each concept is uniquely identified by a term - but 'Concept' refers specifically to additional runtime metadata associated with the term. Every concept has a term, but not every term necessarily becomes **conceptualized**.

```
c = concept(t)
```

In certain cases, such as temporal terms, groups of terms that differ only by secondary numeric component will map to the same Concept, supporting certain belief aggregation purposes despite the terms being uniquene.

Tasks

Tasks are **prioritized** procedures which may or may not be executed, and which may or may not spawn additional tasks.

In other words, Tasks are probabalistic forking co-routines which compete with other tasks to be active in a system.

Control

A homeostatic control system manages the evolution of the open-ended runtime state by determining the relative effective ranking of potential system **Tasks**.

- Consider biological organism *central metabolic control systems* which are responsible for maintaining precise conditions upon which all its chemical processes depend upon for reliability.

Types of Tasks

Control Tasks

- procedures which generally lead to the derivation of new inference tasks, or that are meant to affect system state in a particulary way.

Logic Tasks

- upon being processed, affect Concept **belief** (.), **goal** (!), and **question** (? and @).

Task Equivalence and Merging

Instantiated tasks are an **idempotent** representation of their procedure, which precisely specifies its equality (and hashing) so that duplicate instances may be **merged** with an existing.

The survival of a task generally depends on its continued prioritization, which is ultimately allocated and deallocated by the control system in iterative processes. Therefore repeated merging operations are an expected method of sustaining an equivalent Task instance even while it executes (ex: in another thread).

Example:

```
ConceptFire(term)
Premise(tasklink, termlink)
```

NAL Truth

[0.0..1.0] **freq** x (0.0..1.0) **conf**

NAL Punctuation

NAL Operators

Priority

Priority (sometimes abbreviated as '**pri**') is measured as a scalar quantity between 0 and 1.0 (inclusive). Values above 1.0 and below 0.0 are generally clamped to the valid range.

A **NaN** value is taken to signify a 'deleted' priority which has effectively zero priority but also that it may be removed. Therefore a deleted priority has 'less priority' than 0.

Canonically this is represented with a 32-bit floating point value with a system-determined non-negligible minimum value epsilon (ex: 0.0001) used to cull negligible subthreshold operations.

Bags

Bags, in NARS, refer specifically to a probabalistic variation of the conventional Bag data structure in which prioritized items compete for entrance, rank, and survivability within its specified capacity constraints.

Bags provide storage and retrieval methods for a changing set of admitted items, as well as a method of sampling (and/or draining) its contents probabalistically. Bags exist in several implementations with unique operational advantages and characteristics.

Bags are applied ubiquitously throughout the system in a variety of ways, particularly when some contended resource must be managed or optimized.

Bag Pressure and Balanced Forgetting

Using measured bag admission pressure to determine future forgetting rates

Priority Dynamics

User **input tasks**, and their specific prioritizations, are meant to be the **primary source of budget preference**. The spread of priority throughout the system generally flows from parent tasks to its generated children tasks. If all tasks obey a **<=1.0 conservation policy** then budget can be expected to remain near unity, or under-unity.

Over-unity conditions, while tolerable in some conditions, may lead to uncontrolled feedback -- an analog of **livelock**. In an economic analogy, it may be seen as a 'hyperinflation' scenario, in which even high priority is valueless. Degenerate under-unity situations may result in inactivity -- an analog of **deadlock**.

However, both of these conditions can be detected and managed by **external control intervention**.

Belief Tables

Temporal Belief Tables

Dynamic Belief Tables

Concept Cache

Task Equalization

TODO describe: Input classifier -> filter rebudget -> exe bag

Graphical EQ and channel mix analogy

TermLinks and TaskLinks

Premise Formation

ArrayBag

HijackBag

Meta-NAL Derivation

Temporal Coding and Temporalization

Term Functors

- special terms: TRUE, FALSE, NULL
- their purpose in the term evaluation context
- evaluation conditions

Parallelization

NARS can be fractally decomposed in various granularities to achieve the maximum computational efficiency that a particular implementation offers.

Within limits, multithreading an individual "node" offers the possibility for concurrent reasoner threads to share common computation results which can synergize their effective combined power.

In computer architectures which transcend the limits of an individual multithreadable node, multiple nodes may communicate and coordinate their activity in similar ways, albeit with longer latencies and reduced bandwidth.

InterNARS

NAgent Sensor/Motor Interface

NARquery Fluent API

Telemetry and Performance Analysis

Unit Testing

Areas of Further Research

- Control System Optimization
- Implementation Optimization
    - Virtual machine (JIT, GC) customization using NAL-specific intrinsics and first-class logic data types
    - Online lossy and lossless compression
    - API abstractions facilitating self modification and experimentation
- User-interface
    - 3D visualization
    - Sonification
    - Natural language
    - EEG brainwave and biofeedback
- Integration with external systems
- ...