**NARS in Python – Technical Documentation**

**Object Classes and Data Structures**

**Task**A Task can be *input* or *derived*. Derived Tasks contain sentences which have 2 or more pieces of evidence in its evidential base.

Each Task has a Stamp, which contains the Task’s metadata.

The Task’s Evidential Base is an array of IDs, representing the sentences from which is was derived.

**Concept***Conceptualizing* is the process of creating a new Concept, which is named by a term.

Each Concept contains:

* A *belief* Table holding processed *judgments* about the Concept. Atomic Concepts (that contain no copula) will have empty belief Tables, but are term-linked to Statement Concepts which may have non-empty Tables.
* A *desire* Table holding processed *goals* about the Concept. Atomic Concepts (that contain no copula) will have empty desire Tables, but are term-linked to Statement Concepts which may have non-empty Tables.
* A Bag of *task-links*, which link to Tasks related to the Concept.
* A Bag of *term-links*, which link to other Concepts related to the Concept by a shared immediate term.

**Tables**Tables (belief table and desire table) are stored in Concepts. They are double-ended priority queus that store Narsese Sentences sorted by Confidence. When the Table overflows, the Sentence with the lowest Confidence is purged.

**Bag**The Bag consists of an array of buckets (1-100), and a pointer that points to the currently selected bucket. Objects can be placed into the bag, where they are first wrapped inside a Bag Item with a Priority value, and then inserted into the corresponding bucket based on Priority.

When an item is to be randomly removed from the bag:

* The pointer moves to the next non-empty bucket
* A random number is generated
* If the random number passes the bucket’s probability threshold, an item is removed randomly (uniformly) from the bucket. Otherwise, the pointer moves to the next non-empty bucket.
* This process is repeated until an item is removed.

Items are also stored inside a lookup table, where the key is the hash of the Bag Item’s contained object. *Concept* data structure is defined so its hash is simply the hash of its term string; in this way, Concepts can be directly selected from the bag using the term string.

**Buffer**TBD

**Algorithms**

**Main Control Loop:**

The system either *Observes* a task from its experience buffer, or it *Considers* a Concept from its Memory. The proportion of time the system spends on either process depends on a system parameter, its *Mindfulness*.

**Task Processing:***Initial processing* occurs the first time a task is selected.

*Continued processing* occurs after initial processing, and subsequently whenever the task is selected again.

* **Judgment:**
  + *Initial Processing*
    1. The Judgment’s immediate subterms (subject and predicate) are conceptualized.
    2. The Judgment itself is conceptualized, and bidirectionally term-linked to its subject and predicate concepts.
    3. The Judgment undergoes Revision with the most confident belief in
    4. The Judgment is added directly to the belief table.
    5. **END PROCESSING**
  + *Continued Processing*
    1. First, the Judgment’s corresponding Concept is peeked.
    2. Then, a related belief is accepted from the current Concept (Local Inference: *Revision*) and a related Concept (Forward Inference: *Deduction*)
    3. **Revision / Forward Inference**  
       A belief related to the Judgment Task is selected. Both the Task and related belief are fed into the Inference Engine, which returns a derived Task. The input sentences’ evidential bases are assumed to be distinct, and the Inference Engine merges their evidential bases into the resultant Task.
       1. Revision
          - The highest-confidence belief from the Statement’s Table is selected
          - If the Judgment Task hasn’t interacted with the belief, the Inference Engine does inference and returns a new Task.
       2. Deduction
          - The Judgment’s Subject and Predicate Concepts are peeked. A term-link is peeked from each Concept, returning semantically related Statement Concepts (these may be the same as the Task Judgment).
          - If the Judgment Task hasn’t interacted with the belief, the Inference Engine does inference and returns a new Task.
* **Question:**
  + *Initial Processing*
    1. First, the Question’s corresponding Concept is activated.
    2. Get an answer to the question, by peeking at the highest-confidence belief in the Concept’s belief table.
    3. If the task is an *input* task, the answer is printed as OUTPUT from NARS.
  + *Continued Processing*
    1. TBD
* **Goal:**
  + *Initial Processing*
    1. TBD
  + *Continued Processing*
    1. TBD

**Inference Engine:**

The Inference Engine takes as input one Task and one Belief, and outputs one or more Tasks resulting from the inputs. It assumes the Sentences from the input have distinct (non-overlapping) Evidential Bases; however, it does merge the parents’ Evidential Bases into the derived Sentence’s/Task’s Evidential Base.

1. The relation between the input Task and Belief sentences is identified.
2. One or more derived Tasks are generated using the appropriate inference rule.
3. Tautologies are discarded.
4. The input Task is marked as having interacted with the input Belief.