T-720-ATAI-2016

Advanced Topics in Artificial Intelligence:

NARS

Jordi Bieger

School of Computer Science | Center for Analysis and Design of Intelligent Agents

NARS

- Non-Axiomatic Reasoning System
- Dr. Pei Wang
- 1995 – now

- Non-Axiomatic:
  - All domain knowledge comes from experience and is subject to revision.

AIKR

- Assumption of Insufficient Knowledge and Resources
  - Finite
    - Computational resources and knowledge
  - Real-time
    - Anything can happen at any time and tasks often have time requirements (e.g. deadline, ASAP, urgent)
  - Open
    - The system must be open to any representable content

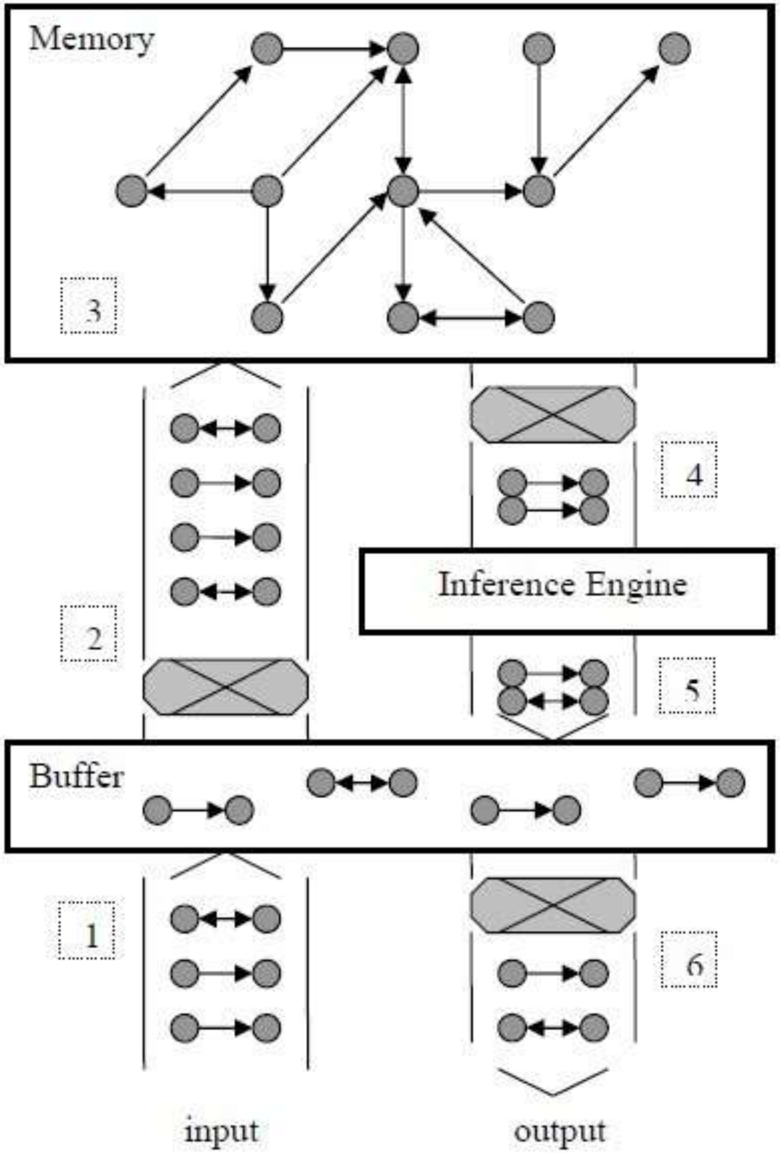
AIKR consequences

- No objective truth
- No guaranteed completeness, consistency, predictability, repeatability, etc.
- “A rational system is one that attempts to adapt to its environment, though its attempts may fail.”
- No model-theoretic semantics, because meaning and truth-values cannot be constants referring to an outside world
- No world model, but summary of experience
- No binary truth or consistent probability distributions

AIKR objections

- Idealization
  - AIKR is also idealization, but not all idealizations are created equal.
  - You cannot idealize away fundamental properties of the thing you're modeling.
- Simplification
  - Divide-and-conquer doesn't work, because AIKR is fundamental to everything

Control



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.

NAL

- Non-Axiomatic Logic:
  1. Inheritance
  2. Sets and variants of inheritance
  3. Intersection and difference
  4. Products, images and relations
  5. Statements as terms
  6. Variables
  7. Temporal
  8. Procedural
  9. Meta

Term logic

- A term is a name of a concept.
  - e.g. “cat”
- A copula is a link between two concepts:
  - e.g. inheritance
- A proposition relates two concepts to each other:
  - subject-copula-predicate
  - e.g. “cats are animals”
- A syllogism is an inference in which one proposition is concluded from two others:
  - Major premise: “cats are animals”
  - Minor premise: “Tom is a cat”
  - Conclusion: “Tom is an animal”

Meaning

- The empirical meaning of a term is formed through the system's experience and subject to revision.
- It is defined by a term's relation to other terms.
- The extension of a term is the set of its specializations: $T^E = \{x \mid x \text{ --> } T\}$
- The intension of a term is the set of its generalizations: $T^I = \{x \mid T \text{ --> } x\}$
- S --> P iff $S^E \subseteq P^E$ and $P^I \subseteq S^I$
- Positive evidence: terms in $(S^E \cap P^E)$ and $(P^I \cap S^I)$
- Negative evidence: terms in $(S^E - P^E)$ and $(P^I - S^I)$

Degree of belief

- Binary truth values and consistent probability distributions violate AIKR.
- Degree of belief depends can be represented as:
  - amount of evidence $\{w, w^+\}$
    - where $0 \leq w^+ \leq w$
  - frequency and confidence <f,c>
    - where $f \in [0,1]$ and $c \in (0,1)$
  - frequency interval [l,u]
    - where $0 \leq l \leq u \leq 1$

Degree of belief

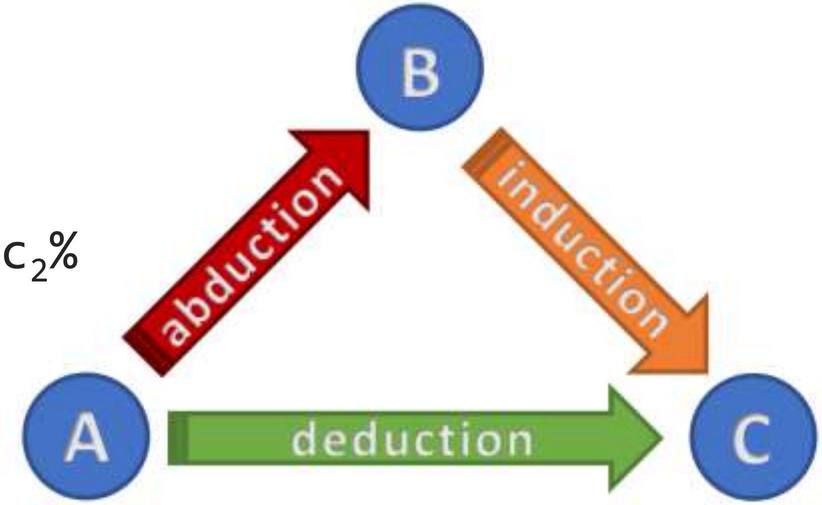
to \ from	$\{w^+, w\}$	$< f, c >$	$[l, u]$
$\{w^+, w\}$		$w^+ = k\frac{fc}{1-c}$ $w = k\frac{c}{1-c}$	$w^+ = k\frac{l}{u-l}$ $w = k\frac{1-(u-l)}{u-l}$
$< f, c >$	$f = \frac{w^+}{w}$ $c = \frac{w}{w+k}$		$f = \frac{l}{1-(u-l)}$ $c = 1 - (u - l)$
$[l, u]$	$l = \frac{w^+}{w+k}$ $u = \frac{w^++k}{w+k}$	$l = fc$ $u = 1 - c(1 - f)$	

Narsese

- The language used to express things in NAL.
- A statement is delimited by < >
- A sentence ends in punctuation: . ? ! @
- A sentence can be followed by a truth value corresponding to <f,c>: %1;0.9%
- Examples:
  - `<cat --> animal>. %1;0.9%`
    - roughly: I’m 90% sure that cats are always animals.
  - `<cat --] smart>. %0.6;0.8%`
    - roughly: I’m 80% sure that 60% of cats are smart (or cats are 60% smart).

Inheritance and Inference

- Deduction:
  - $<\$a \text{ --> } \$b>.\ \%f_1;\ c_1\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<\$a \text{ --> } \$c>.\ \%f_1f_2;\ f_1c_1f_2c_2\%$
- Abduction:
  - $<\$a \text{ --> } \$c>.\ \%f_1;\ c_1\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<\$a \text{ --> } \$b>.\ \%f_1;\ c_1f_2c_2/(c_1f_2c_2 + k)\%$
- Induction:
  - $<\$a \text{ --> } \$b>.\ \%f_1;\ c_1\%$
  - $<\$a \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ f_1c_1c_2/(f_1c_1c_2 + k)\%$



Inheritance and Inference

- Conversion:
  ```
  <$a --> $b>. %f_1;c_1%
  <$b --> $a>. %1; f_1c_1/(f_1c_1 + k)%
  ```
- Exemplification:
  ```
  <$a --> $c>. %f_1;c_1%
  <$b --> $c>. %f_2;c_2%
  <$a --> $b>. %1; f_1c_1f_2c_2/(f_1c_1f_2c_2 + k)%
  ```
- Revision:
  ```
  <$a --> $b>. {w^+_1;w_1} // not valid Narsese
  <$a --> $b>. {w^+_2;w_2}
  <$a --> $b>. {w^+_1+w^+_2;w_1+w_2}
  ```

Statements

- Judgment
  - <cat --> animal>**.** // declarative knowledge
- Question
  - <cat --> animal>**?** // ask about truth
  - <**?x** --> animal>**?** // ask about value (what? where? etc.)
- Goal
  - <SELF --] happy>**!** // try to become happy
- Desire query
  - <SELF --] happy>@ // how much is this goal desired?

Backward inference

- Forward inference:
  - judgment + judgment = judgment
- Backward inference:
  - question + judgment = question or truth-value

Similarity

- Comparison:
  - $<\$a\ \texttt{-->}\ \$c>.\ \%f_1;\ c_1\%$
  - $<\$b\ \texttt{-->}\ \$c>.\ \%f_2;\ c_2\%$
  - $<\$a\ \texttt{<->}\ \$b>.\ \%w^{+}=\ f_1\&c_1\&f_2\&c_2;\ w=(f_1|f_2)\&c_1\&c_2\%$
- Analogy*:
  - $<\$a\ \texttt{-->}\ \$b>.\ \%f_1;\ c_1\%$
  - $<\$a\ \texttt{<->}\ \$c>.\ \%f_2;\ c_2\%$
  - $<\$c\ \texttt{-->}\ \$b>.\ \%f_1 f_2;\ c_1 f_2 c_2\%$
- Deduction:
  - $<\$a\ \texttt{<->}\ \$b>.\ \%f_1;\ c_1\%$
  - $<\$a\ \texttt{<->}\ \$c>.\ \%f_2;\ c_2\%$
  - $<\$b\ \texttt{<->}\ \$c>.\ \%\ f_1 f_2;\ c_1 c_2(f_1+f_2\ -\ f_1 f_2)\%$

Compound terms and sets

- A compound term is a term with internal structure.
  - e.g. (operator, component, component)
- Meaning has a literal and empirical part
- Syntactic complexity

Variants of inheritance

- Instance:
  - <Jerry {-- mouse>.
  - <{Jerry} --> mouse>.
- Property:
  - <mouse --] brown>.
  - <mouse --> [brown]>.
- Instance-property:
  - <Jerry {-] brown>.
  - <{Jerry} --> [brown]>.

Equivalence

$S \leftrightarrow P$	$\{S\} \leftrightarrow \{P\}$
$S \leftrightarrow P$	$[S] \leftrightarrow [P]$
$S \rightarrow \{P\}$	$S \leftrightarrow \{P\}$
$[S] \rightarrow P$	$[S] \leftrightarrow P$
$S \circ\!\!\rightarrow P$	$\{S\} \rightarrow P$
$S \rightarrow\!\!\circ P$	$S \rightarrow [P]$
$S \circ\!\!\rightarrow\!\!\circ P$	$\{S\} \rightarrow [P]$

Table 4.3: The Equivalence Rules of NAL-2

Intersection and difference

- Extensional intersection
  - `($x & $y & $z)`
  - `(&, $x, $y, $z)`
- Intensional intersection
  - `($x | $y | $z)`
  - `(|, $x, $y, $z)`
- Extensional difference
  - `($x - $y)`
  - `(-, $x, $y)`
- Intensional difference
  - `($x ~ $y)`
  - `(~, $x, $y)`

Compound sets

- Compound extensional set:
  - {a, b, c} = (|, {a}, {b}, {c})
- Compound intensional set:
  - [a, b, c] = (&, [a], [b], [c])

NAL-3 inference

- Intersection:
  - $<\$a \text{ --> } \$c>.\ \%f_1;\ c_1\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<(\$a \mid \$b) \text{ --> } \$c>.$
    - $\%f_1 f_2;\ (\sim f_1 \& c_1 | \sim f_2 \& c_2) + f_1 f_2 c_1 c_2\%$
- Union:
  - $<\$a \text{ --> } \$c>.\ \%f_1;\ c_1\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<(\$a \,\&\, \$b) \text{ --> } \$c>.$
    - $\%f_1 | f_2;\ (f_1 c_1 | f_2 c_2) + \sim f_1 \& \sim f_2 \& c_1 \& c_2\%$
- Difference:
  - $<\$a \text{ --> } \$c>.\ \%f_1;\ c_1\%$
  - $<\$b \text{ --> } \$c>.\ \%f_2;\ c_2\%$
  - $<(\$a \sim \$b) \text{ --> } \$c>.$
    - $\%f_1 \& \sim f_2;\ (\sim f_1 \& c_1 | f_2 \& c_2) + f_1 \& \sim f_2 \& c_1 \& c_2\%$

NAL-3 inference

- Intersection:
  - `<$a --> $b>.` %f_1; c_1%
  - `<$a --> $c>.` %f_2; c_2%
  - `<$a --> ($b & $c)>.`
    - %$f_1 f_2$; $(\sim f_1 \& c_1 | \sim f_2 \& c_2) + f_1 f_2 c_1 c_2$%
- Union:
  - `<$a --> $b>.` %f_1; c_1%
  - `<$a --> $c>.` %f_2; c_2%
  - `<$a --> ($b | $c)>.`
    - %$f_1 | f_2$; $(f_1 c_1 | f_2 c_2) + \sim f_1 \& \sim f_2 \& c_1 \& c_2$%
- Difference:
  - `<$a --> $b>.` %f_1; c_1%
  - `<$a --> $c>.` %f_2; c_2%
  - `<$a --> ($b - $c)>.`
    - %$f_1 \& \sim f_2$; $(\sim f_1 \& c_1 | f_2 \& c_2) + f_1 \& \sim f_2 \& c_1 \& c_2$%

Products and relations

- Product
  - (*, a, b, c)
  - (a * b * c)
- A term R is a relation if there is a product P so that:
  - P --> R,
    - <(acid * base) --> neutralization>.
    - "Acid and base neutralize each other"
  - R --> P
    - <neutralization --> (acid * base)>.
    - "Neutralization happens between acid and base"

Products and relations

- Given an operator and component, an image can identify the other component in the relation.
- Extensional image
  - <(*, a, b) --> rel>.
  - <a --> (/, rel, _, b)>.
  - <b --> (/, rel, a, _)>.
- Intensional image
  - <(\, rel, _, b) --> a>.
  - <(\, rel, a, _) --> b>.

NAL-4 equivalence

$S \rightarrow P$	$(S \times M) \rightarrow (P \times M)$
$S \leftrightarrow P$	$(S \times M) \leftrightarrow (P \times M)$
$(\times\ T_1\ T_2) \rightarrow R$	$T_1 \rightarrow (\perp\ R \diamond T_2)$
$R \rightarrow (\times\ T_1\ T_2)$	$(\top\ R \diamond T_2) \rightarrow T_1)$

Table 4.7: The Equivalence Rules of NAL-4

Terms as statements

- Example:
  - `<({John} * {<whale --> mammal>}) --> know>.`
  - “John knows that whales are mammals.”
- Negation:
  - <(--, <whale --> fish>)>.
  - “Whales are not fish.”
- Conjunction:
  - <(&&, <whale --> mammal>, <whale --> swimmer>)>.
- Disjunction:
  - <(||, <whale --> mammal>, (--, <whale --> mammal>))>.

Higher-order copula

- Implication
  - <<weather --] raining> ==> <grass --] wet>>.
- Equivalence
  - <<(a * b) --> lesser> <=> (b * a --> greater)>.

Variables

- Independent variables
  - `<(<$x --> key> && <$y --> lock>) ==> <($x * $y) --> open>>.`
  - “Every key can open every lock.”
- Dependent variables
  - `<(<#x --> key> && <#y --> lock>) ==> <($x * $y) --> open>>.`
  - “There is a key that can open a lock.”
- More examples:
  - `<(<$x --> key> && <#y --> lock>) ==> <($x * $y) --> open>>.`
  - “For every key there’s a lock it can open.”
  - `<(<#x --> key> && <$y --> lock>) ==> <($x * $y) --> open>>.`
  - “There is a key that can open all locks.”

Temporal statements

- An event is a statement whose time attributes are specified, and whose truth value holds in a certain time interval.
- The time interval of a non-event is unspecified but includes “now”.
- All time is specified relatively. Events can occur before, after, or concurrently with each other.
- Events may need to be divided up to specify more complicated time relations.

Temporal conjunction

- Sequential conjunction:
  - (E &/ F) // E happens before F
- Parallel conjunction:
  - (E &/ F) // E and F are both true at the same time
- Non-temporal conjunction:
  - (E && F) // E and F are both true at some unspecified time (always?)

Temporal implication

- Predictive implication:
  - E =/> F // if we saw E, we expect to see F in the future
  - E is a sufficient precondition, F is a necessary postcondition
- Retrospective implication:
  - E =\> F // if we see E, we expect F has happened in the past
  - E is a sufficient postcondition, F is a necessary precondition
- Concurrent implication:
  - E =|> F // if E happens, we expect to F happen also (at the same time)
  - E is a sufficient co-condition, F is a necessary co-condition

Temporal equivalence

- Predictive equivalence :
  - E </> F // E is true iff F is true later
  - E is a equivalent precondition, F is a equivalent postcondition
- Retrospective equivalence :
  - Not defined in Narsese, but F <\> E would have been the same as E </> F.
- Concurrent equivalence:
  - E <|> F // E happens iff F happens at the same time
  - E and F are equivalent co-conditions

Tense

- We can add past, present and future tense to sentences in Narsese.

- Past:
  - `<({Bush} * {US} --> president)> :\: %1;%.99%`
- Present:
  - `<({Obama} * {US} --> president)> :|: %1;%.99%`
- Future
  - `<({Trump} * {US} --> president)> :/: %1;%.2%`

Procedural statements

- An operation is a special kind of event that can be executed by the system.
- The operations of one system will be observed as events by others.
- Input: typically independent variables
- Output: typically dependent on input

Procedural example

- `<(*, A, B, C) --> sum>. // a+b=c`
- `<<(add, $x, $y, #z)> =/> <(*, $x, $y, #z) --> sum>>.`
  - If I add $x and $y, I expect to know/observe that $x+$y=#z in the future.
- `<<(sub, $x, $y, #z)> /=> <(*, $y, #z, $x) --> sum>>.`
  - If I subtract $y from $x, I expect to know/observe that $y+#z=$x in the future.
- `<<(div, $x, $y, #z)> \=> (--, <0 {-- $y>)>.`
  - If I'm going to divide $x by $y, then $y must not have been 0.

Goals and desire values

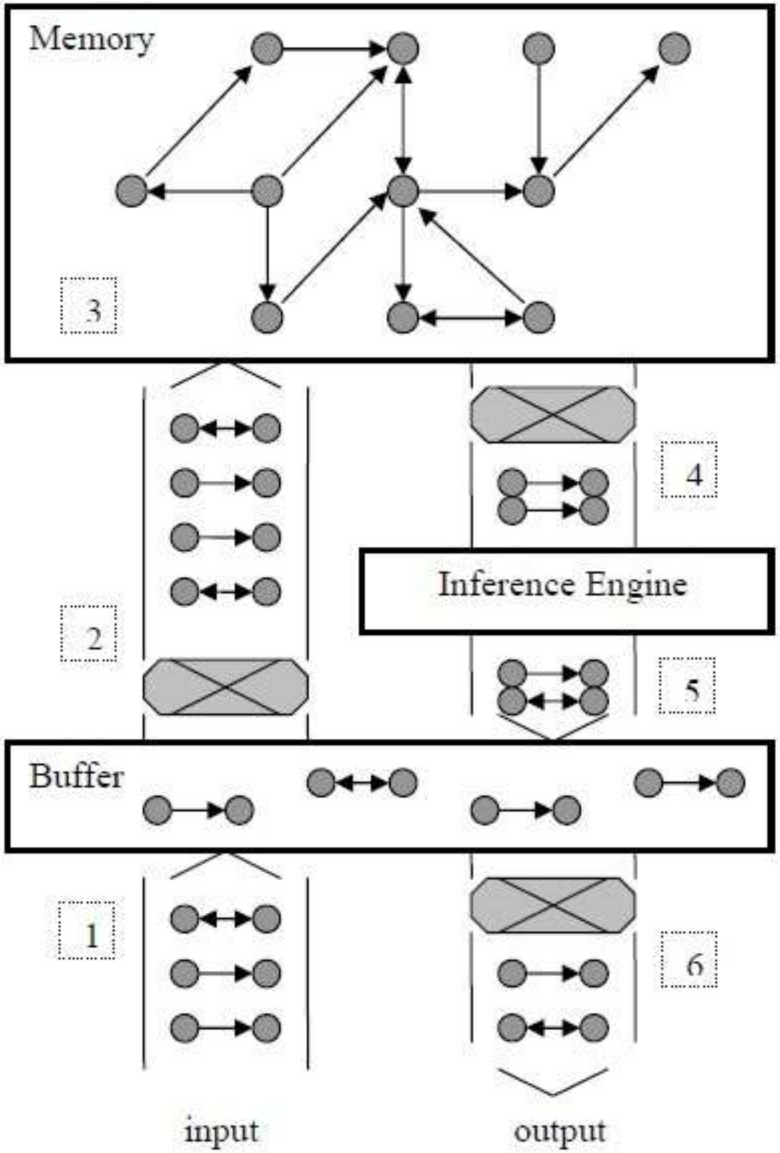
- Like questions, goals don’t have a truth value, and may contain variables to be instantiated.
- `<SELF --> [happy]>! // make yourself happy`
- Goals have (fluid) desire values to decide which ones to prioritize, defined as the truth value of <G ==> D>. where G is the goal and D is a virtual “desired” state.
- Desire values can be attached to any statement, because any statement may become a goal.

Control

- How to allocate limited resources?
- Relatively underdeveloped in NARS.

- Every statement is a task. (The task is for the statement to be processed.)
- Probabilistic priority queues
- Memory
  - Concept
    - Task-link
    - Term-link (beliefs only)

Control



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

- Probabilistically select a concept C from the memory
- Probabilistically select a tasklink from C, which specifies the task T to be used
- Probabilistically select a termlink from C, which specifies the belief B to be used
- With T and B as premises, trigger the applicable inference rules to derive new tasks and add them into the task buffer
- Probabilistically select some tasks from the task buffer for pre-processing

Budget

- Priority
  - Current access chance is increased when the item is activated but decays over time.
- Durability
  - Controls the decay rate
- Quality
  - Long-term general value to the system that determines “residual priority”

Budgets

- Beliefs are judgments in memory
  - Quality: high confidence, extreme frequency, low complexity
  - Priority: new evidence, solution contribution, components
  - Durability: new evidence, solution contribution
- Tasks are new judgments, goals or questions
  - Quality: like belief
  - Priority and durability: user specified:
    - $p;d;q$ <statement>! // where p;d;q in [0,1];(0,1);[0,1]
- Concepts are created for new terms
  - Quality: complexity, link quality, usefulness
- Links link to relevant tasks and terms
  - Quality: initially from linked thing, adjusted over time

Assignments Schedule

- Programming assignment 4
  - Due: 31.03.2016 23:59
  - Grade: 5%
- Programming assignment 5
  - Make teams: before next class
  - Due: 04.04.2016 09:00
- Essay / Research proposal
  - Pick topic: 22.03.2016 12:00
  - Due: 08.04.2016
  - Presentation: 08.04.2016

Questions?