Interesting Usage Scenarios

Help

>help ~~→ P~~ } prints help.
>help {fun} }

Uses processors ~~A~~ to generate ~~a~~ & output ~~the~~ responses.
uses up W.M. like anything else, and thus may distract it from other things.

Interruption & Reminding

1. Interrupt it enough that a calculation request leaves its W.M. (but is still in S.T.M)
2. > I asked you a question!
3. Processor looks through S.T.M for most recent unanswered ~~q~~ ~~q~~ ~~query~~ request.
4. If found: "Oh yes, you asked me: 3+5". Otherwise: "sorry I've forgotten".

Learning to solve Expressions - the long way

1. > 3+?=8 ⇒ ~~knows '?' represents a number~~
   ⇒ doesn't know what to do.
2. > Try ? = a number ⇒ Now knows ? = a number (automatically assumes between 1 & 100)
   ⇒ Once, tries a random number in that range
   ⇒ succeeds or fails.
   ⇒ Already knows how to validate an expression.
3. > keep trying ⇒ Repeats the previous attempt multiple times until it succeeds. Using a new random # each time.

~~ie. repeat { pick random number, try it.~~

Interesting implication about learnings being represented as sequences of actions.

Now adds "repeat" to this sequence.

ie: { '?' is a number (1...100)
- pick a random number
- try it.

2/

Interesting Usage Examples

(... contd.)

4. Over time uses these experiments } to learn.

   ⇒ Uses ~~the~~ general desire to learn patterns for: "What x will result in y?"

   x: thing or action

   y: thing or input.

⋮

10. New generalised model enables me to ask:

    > 1005 + ? = 2134

    ← Can't answer this using initial simplistic model. So needs to learn a new technique which can overall the simplistic approach.

Double guessing

> ? + ? = 8

(Assuming already learned from 3+?=8) And has learned generic mode)

⇒ ~~The~~ Generic advanced model doesn't work, so falls back to old habits:
- → guesses each ? as a random number in range 1..100.
- → keeps trying until succeeds and outputs the result.

THIS is a generalisation of a learned approach.

Generic Token

> 3 + x = 8
  3 + ? = 8

} should treat x/y as a token and associate it with the learning.

Waking up

- when going unconscious, loose all state in WM, but persist STM + LTM.
- when waking up, uncover prior state from STM