system_blade_of_grass_ccz

Blade of grass / tree model

Precondition: region w/ set of resources

Encoded Instructions for combining resources

Kernel
- bootstrap - capable of creating new organism
- minimal
- dispersable
- information to define key elements of organism

Process:

kernels distributed into environment

kernels bootstrap

into from kernels define/direct combination of resources from environment into larger machine

larger machines ultimately produce more kernels that are distributed into world

Observations
- requires locking process... combinations can be established that are unlikely to be reversed by random events

- requires instructions... locked data structures that can drive machine like operations

- requires memory

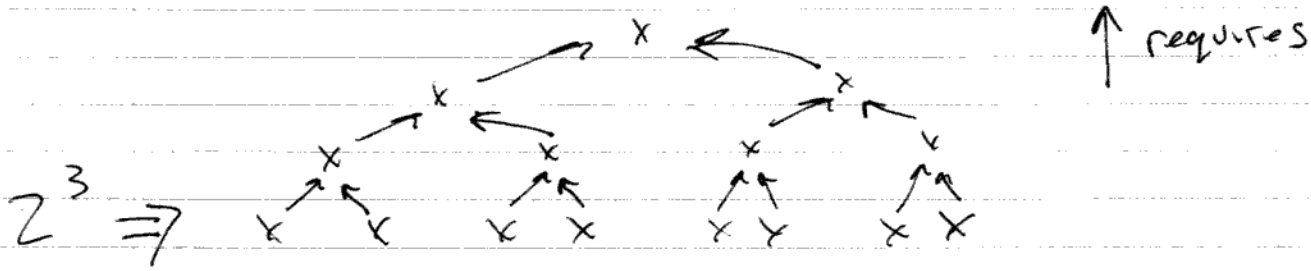
Blade of Grass model cont.

Observations cont...

- Happens at different scales

Q) How to ~~scale~~ expand in size, time, etc

A) Hierarchical Preconditions



Q) How to scale?

A) Lower layers are natural relationships in world and x's are provided by world. all that is required is catylist to enable lower layers

Concept:
A subsect bility to organization must be wide spread in environment then instructions just drive higher level organization

~~The world and the organ~~
the world is like an unmarked part of the organism. An essential part of the organism that is not always associated with it.

Grass blade model cont...

- implications....

bottom layers of hierarchy are primed by world.... primed and ready to go

these could be thought of as building blocks

the building blocks are diverse ... extensive enough to support X number of layers above them

Q) what about sample vs everything in hierarchy

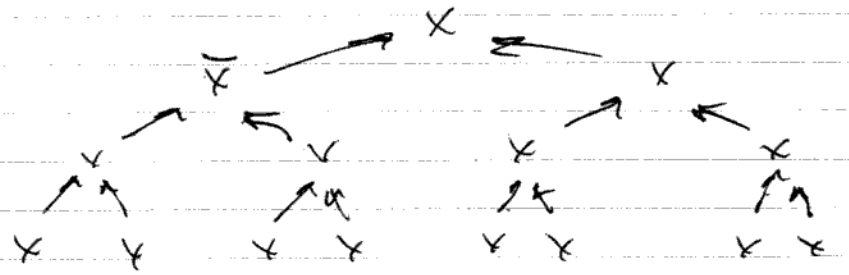
System is primed for reinforcement learning. attn

Some sort of memory or histeresis is required

Presence of node in upper hierarch will allow lower nodes to be instanciated

Then you need a memory component to potentiate lower nodes to future presence of trigger node or related trigger node

New but related question



$\overline{X}$ is a precondition, a state,

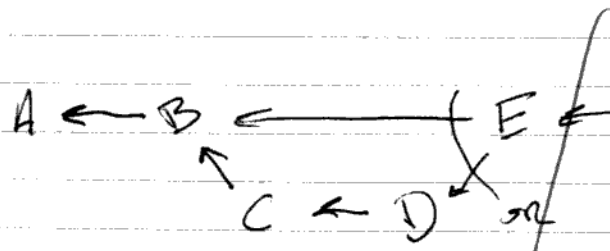
but seems likely $\overline{X}$ is combination like $\overline{X} = A \cdot B \cdot C$

Could be

$\overline{X} =$ A or B or C

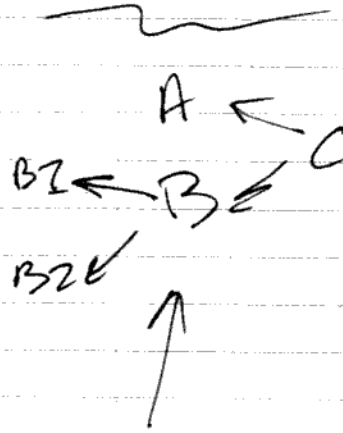
What if

$\overline{X}$ is not a single state but an instance, a particle?



Shows kernel – world – kernel dependency/ordering

Q) OR, AND



C – actual particle or state

~C – machine/potential/trigger to build a "C"

Both of above are important

Should ~C only exist if A or B exist?

How many B's should be sitting around in world space waiting to produce "C"'s?

Should ~~this~~ the number of "B"s be function of potentialization described in previous page?

Q) what are the potential problems w/ this universe model?

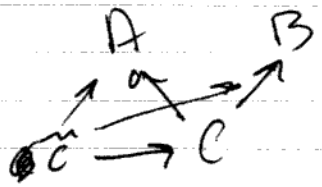
- Reliance / Dependence on frequency of particles....
  the rate of particles will impact the likely hood of triggering
  The rate must be optimized for proper logic

- Need to track existance and entanglement of many many particles

Point -
Existance of Particle & Particle Factory

non in sense particle factory is a book keeping mechanism to properly allocate resources.

IF



Q) Do A, B change state when State C is in existance.

- That would seem to match compounding of matter → Atoms share electrons producing a more stable compond -- the individual atoms are not free to reassociate as they originally were without reversing the sharing.