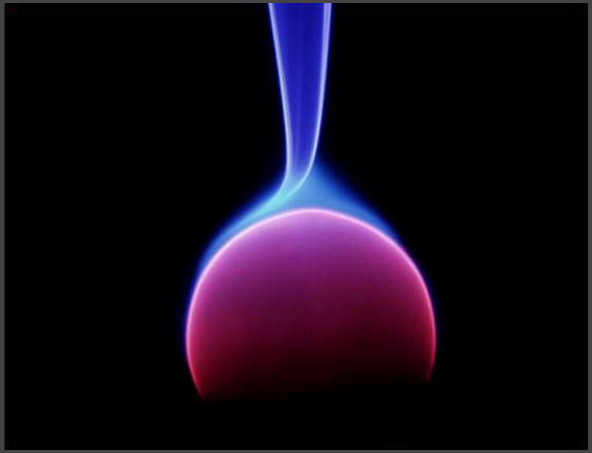
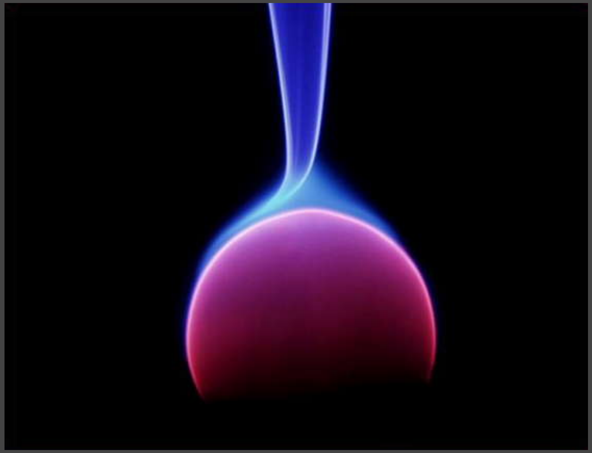
Implementing the Rosen Diagram



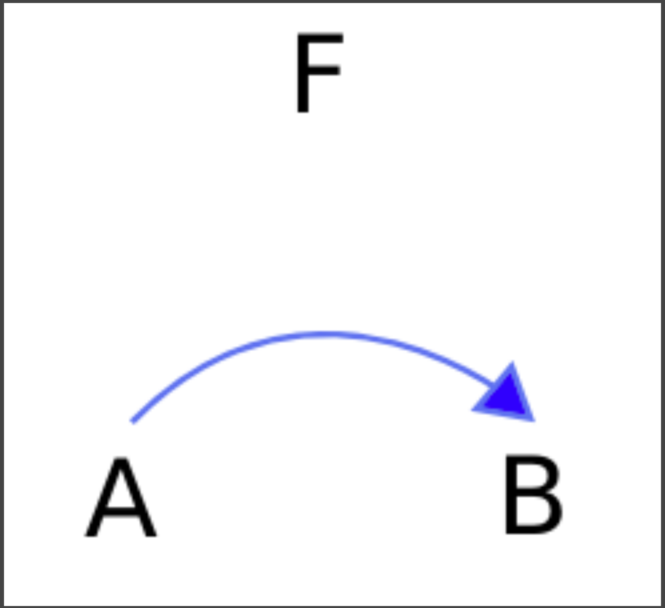
a first attempt at a mutually self-reflective program architecture

Implementing the Rosen Diagram

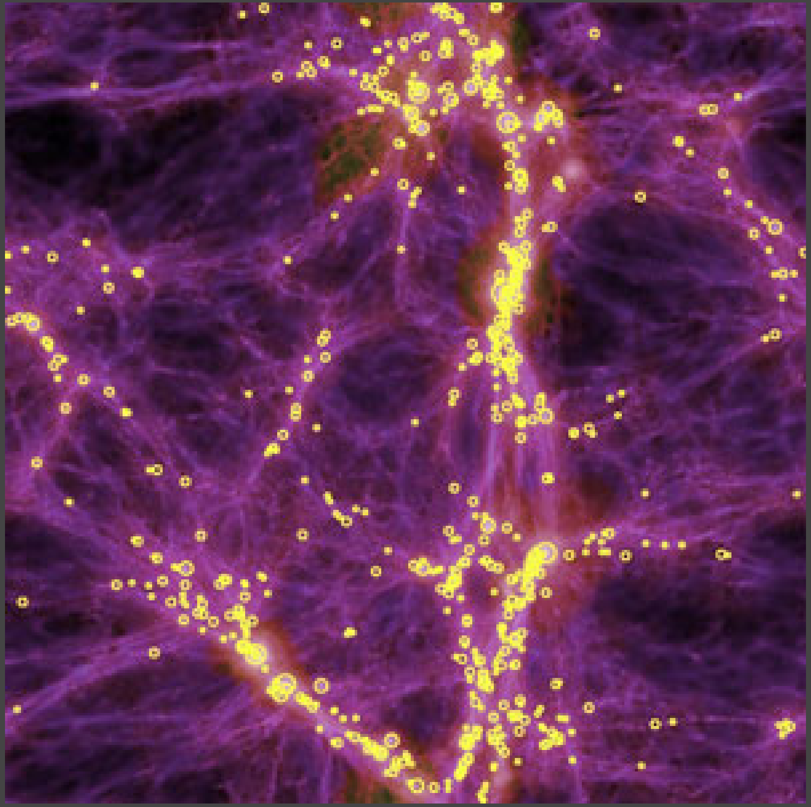


a journey into madness

The traditional modeling relation



B's are produced from A's according to F.
F is static for all time.
Systems are modeled by correlating a step from one time step to the next with a transition from A to B through F.



The history of physics can be viewed as the quest for an F that totally describes the behavior of the universe.

This whole situation is about models.

Our "laws of physics" are just models of the behavior of the universe we observe.

What are we building our models out of?

The metaphor provided by Newton is still the dominant modeling relation we use to construct models of behavior in all fields.

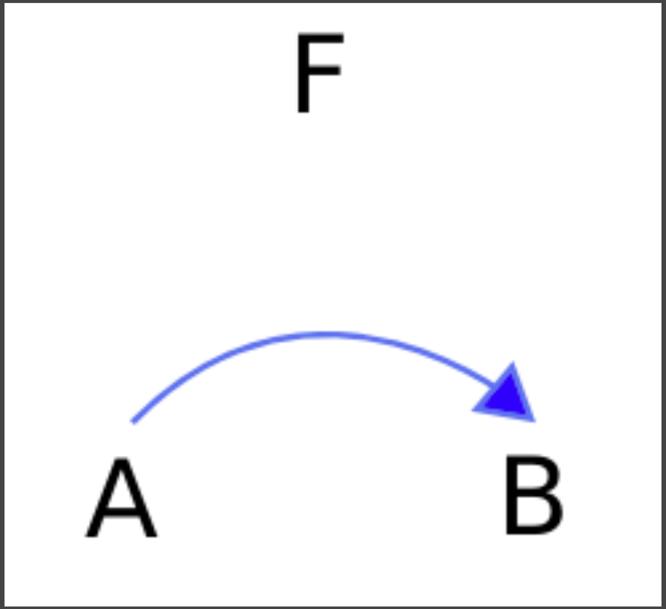
The reason biological systems have been so resistant to modeling is that the models we are using have fundamental limitations in the ways elements can be entailed within the system.

ROSEN'S CLAIM

What is the Rosen Diagram?

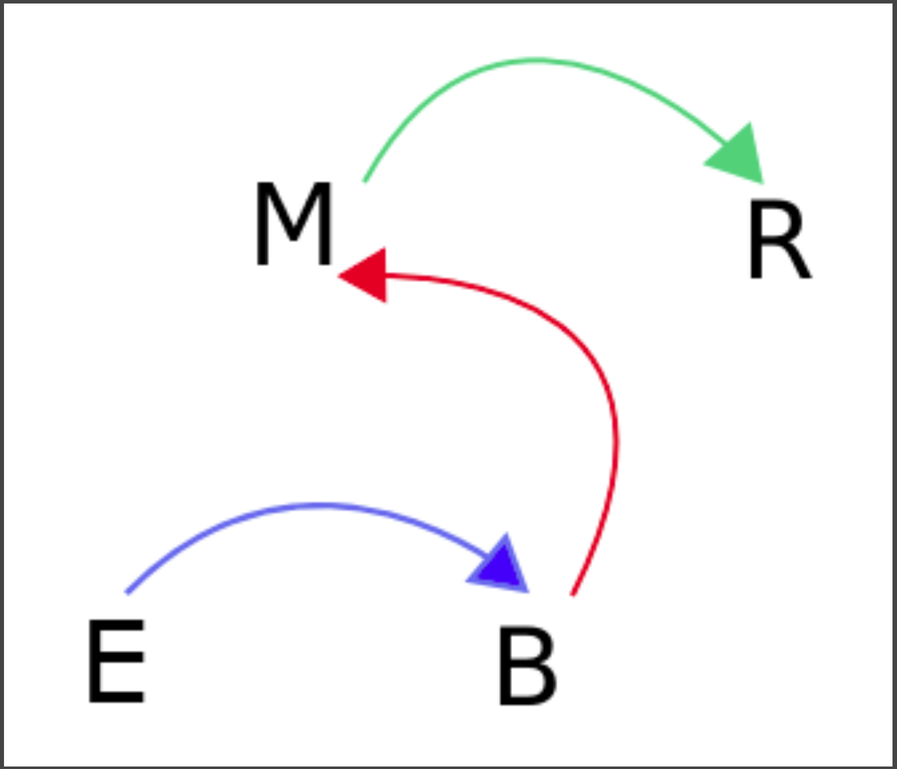
- M generates B from E
- R generates M from B
- B generates R from M

Rosen's category theoretical diagram of the simplest "organism".



$$\begin{tikzcd} & F & \\ A \arrow[rr, bend left] & & B \end{tikzcd}$$

Newton



$$\begin{tikzcd} & M \arrow[r, bend left] & R \\ E \arrow[r, bend left] & B \arrow[u, bend right] & \end{tikzcd}$$

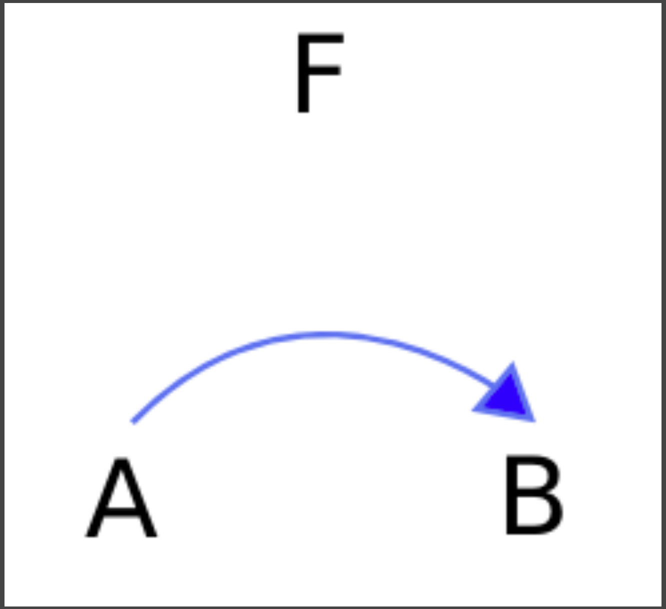
Rosen

There are three roles:

Actor	What operates on the other two, reading from the source and producing the target.
Source	What the Actor bases its decisions on.
Target	What is actually operated on/replaced/renewed etc.

Adding entailment of the Actor

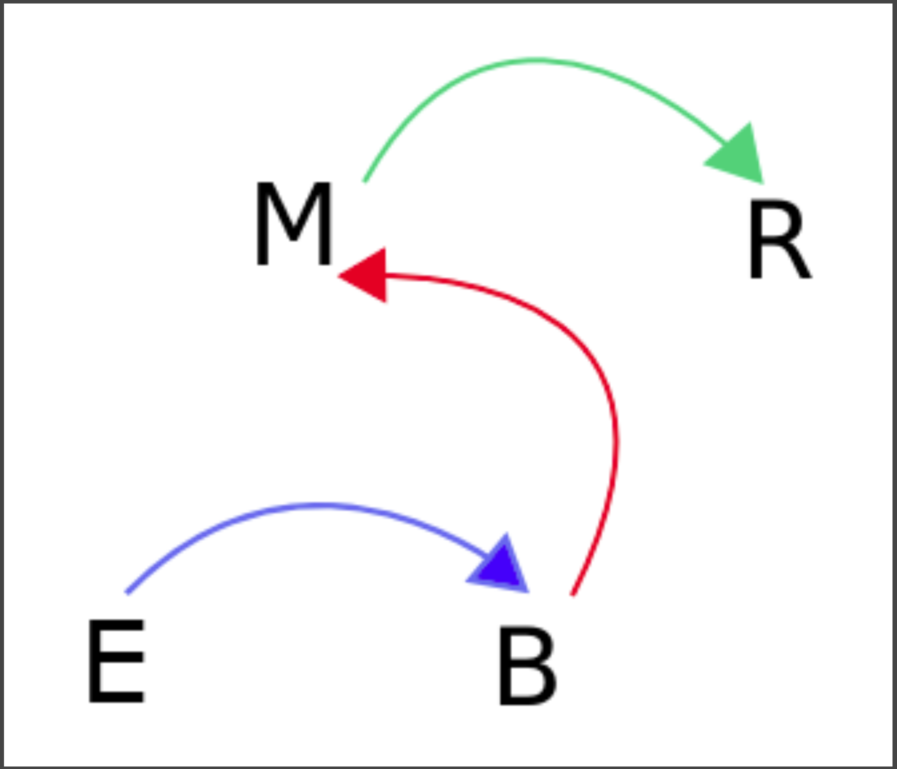
- In Newton's model, F is static, forever producing B's from A's in exactly the same way.



$$\begin{tikzcd} & F & \\ A \arrow[rr, bend left] & & B \end{tikzcd}$$

Newton

- In Rosen's model, M is entailed by R, which itself is entailed by B.

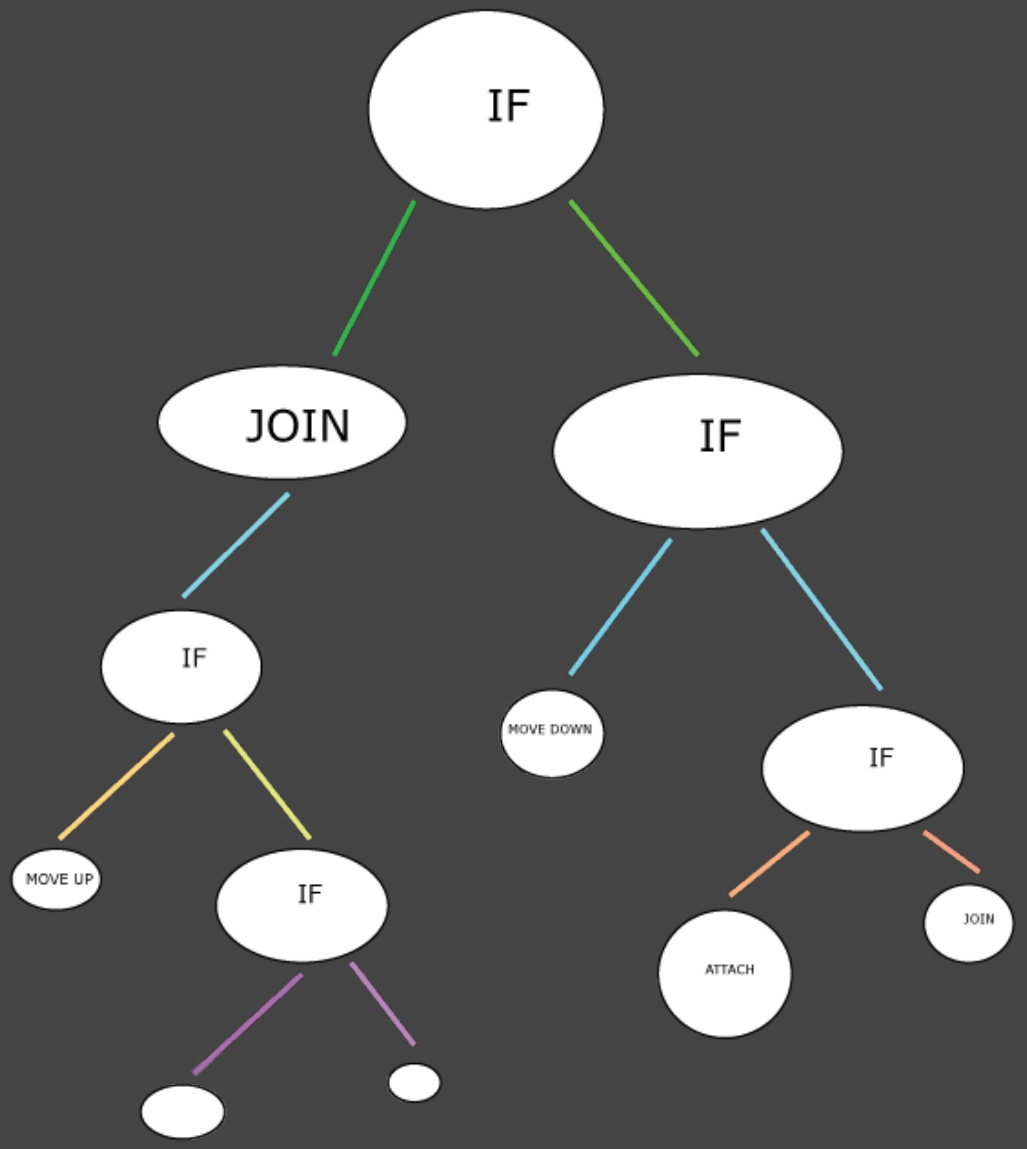


$$\begin{tikzcd} & M \arrow[r, bend left] & R \\ E \arrow[r, bend left] & B \arrow[u, bend right] & \end{tikzcd}$$

Rosen

What can play all three roles at once?

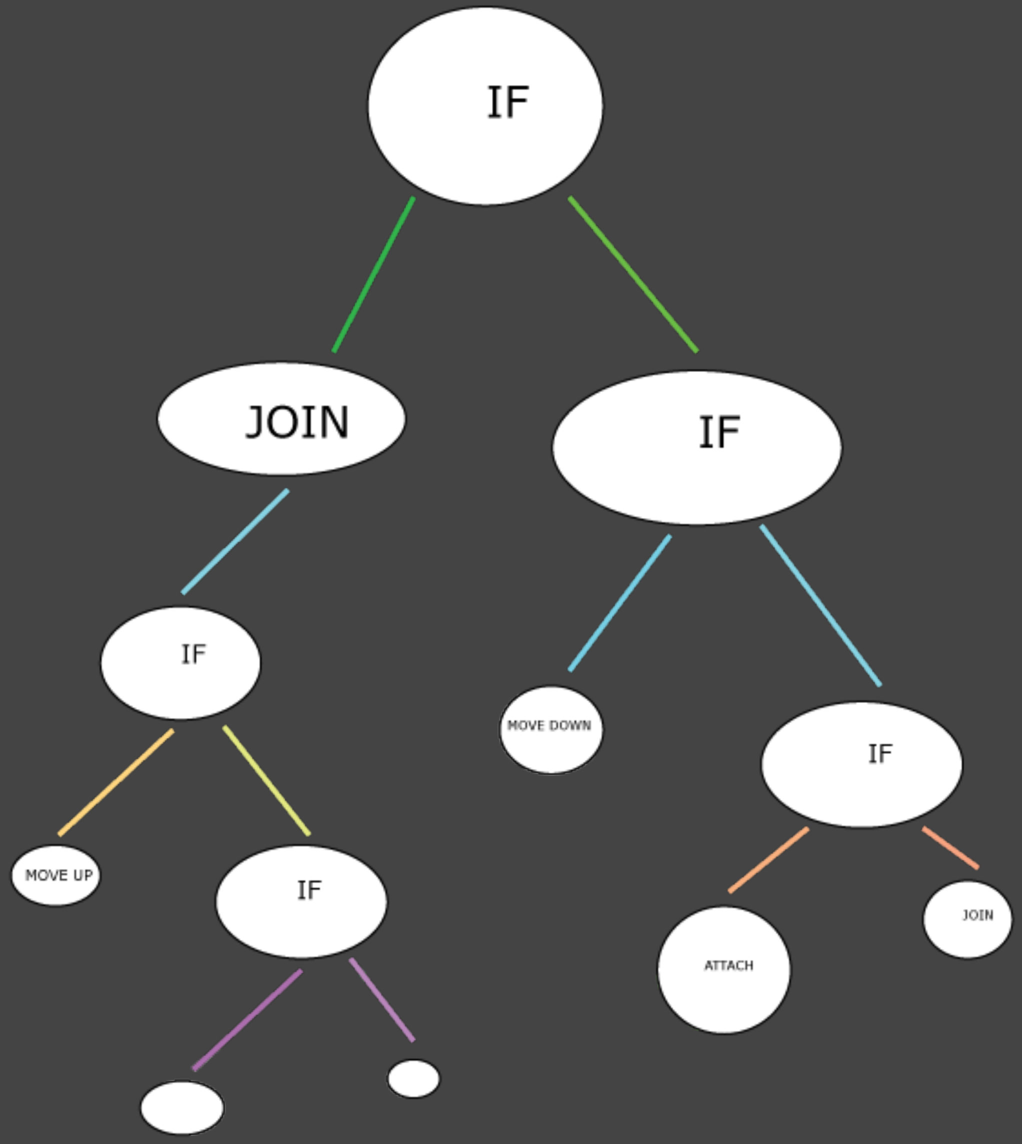
Actor	What operates on the other two, reading from the source and producing the target.
Source	What the Actor bases its decisions on.
Target	What is actually operated on/replaced/renewed etc.

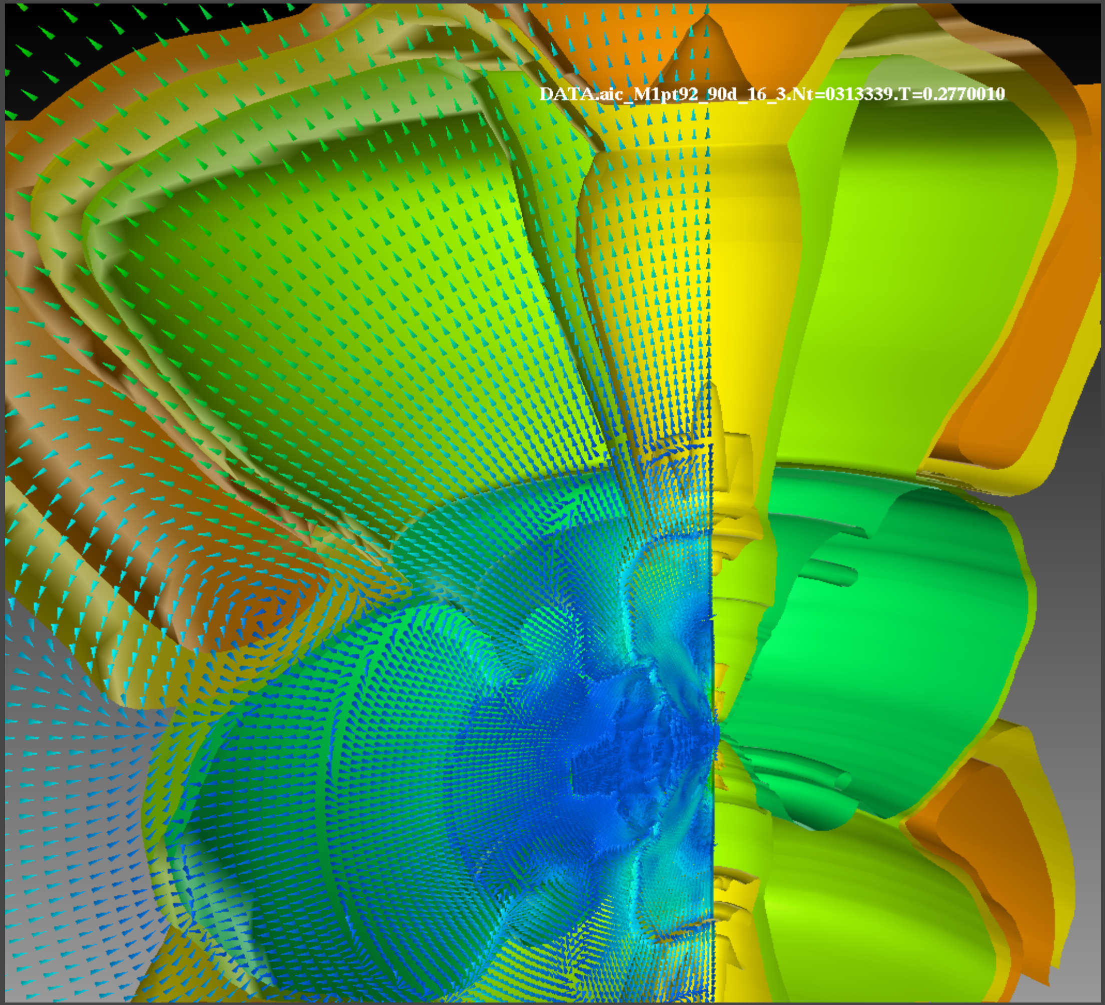


Answer: Executable Trees

Executable Trees

- Operate on other trees.
- Can be read from.
- Can be modified and operated on.





But also, there should be some sense of entropy, or wear in the system.

SIMULATED ENTROPY

Each node tracks the number of times it has been part of the execution flow, and is removed if it is over some limit.

The limit depends on the level of the node: deeper nodes have a lower limit, and are therefore removed more quickly.

Operations: Grow

The grow operation is the only way new nodes are added to the tree. The type of node it adds to the target tree is based on what type of node it points to in the focus tree.

Grow attempts to fill out an unbalanced tree, and adds from the root if the tree is full (attaching the old tree as a subtree).

Operations: Move

The move operation shift the pointer up or down the tree, in either the focus or the target. Every operation is fully defined, so there is a separate operation for moving down the target tree and moving down the focus tree, for example.

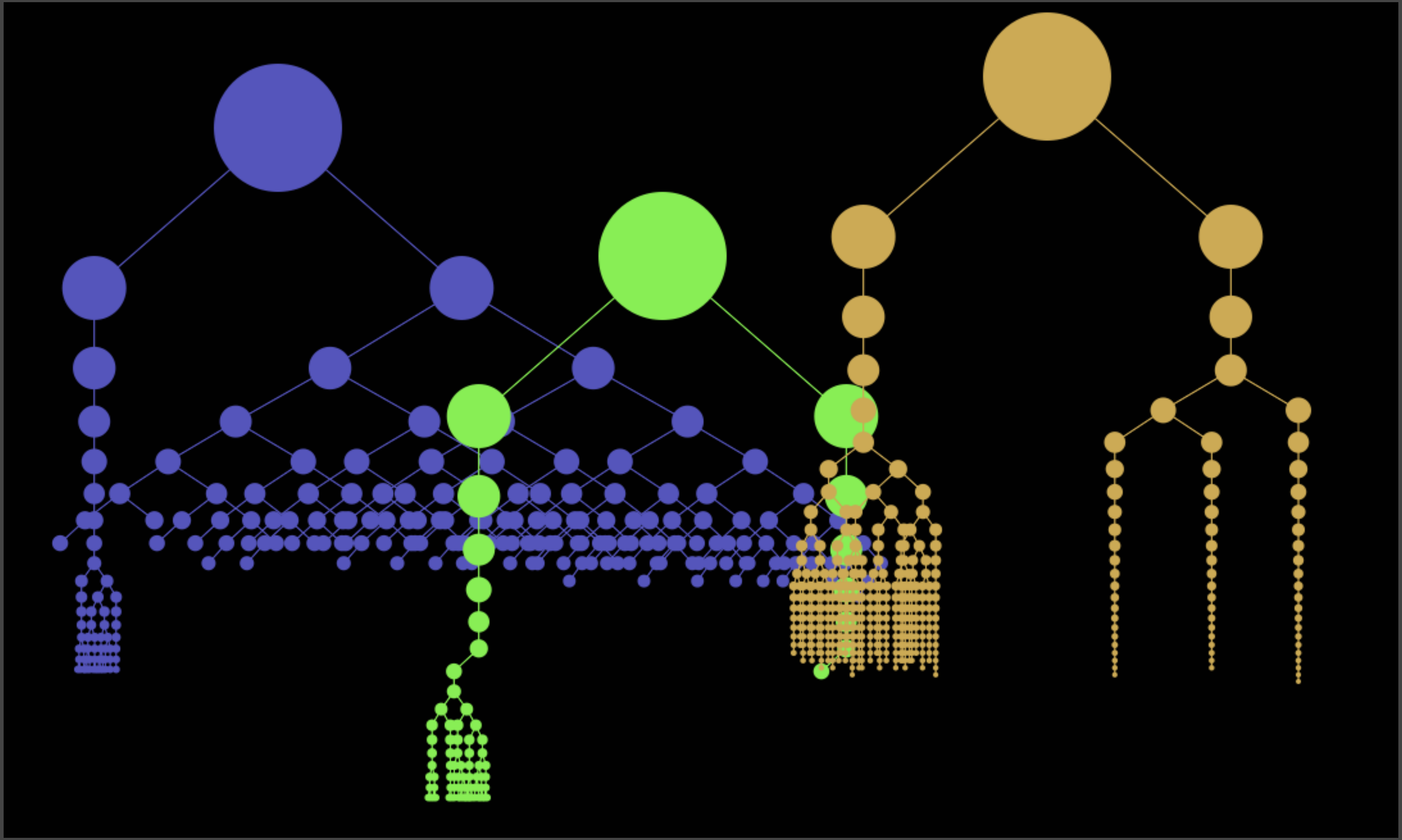
Operations: Renew

The renew operation removes the wear on its target node, but at the expense of inducing wear on itself. This is the only way to counteract the entropy imposed by repeated execution.

Operations: If

The if operation chooses between two different execution paths depending on what type of node it is sensitive towards. There is a different If operation for each possible type of node.

```
(moveup-target
  (grow
    (renew
      (movedown-focus-1
        (if-if
          (movedown-focus-2
            (movedown-focus-2
              (movedown-target-2
                (moveup-focus
                  (if-movedown
                    (moveup-target
                      (if-renew
                        (movedown-target-2 @)
                        (if-if @ @)))
                    (movedown-target-2
                      (movedown-target-1
                        (if-moveup @ @))))))))
          (if-moveup
            (if-movedown
              (movedown-focus-2
                (if-grow
                  (renew
                    (if-moveup
                      (if-renew
                        (movedown-target-1 @)
                        (grow @))
                      (moveup-focus
                        (if-moveup @ @))))
                  (movedown-focus-1
                    (moveup-focus
                      (if-grow
                        (grow @)
                        (grow @))))))
              (movedown-target-2
                (if-if
                  (moveup-focus
                    (if-renew
                      (if-movedown
                        (moveup-target @)
                        (if-if @ @))
                      (movedown-target-1
                        (movedown-target-2 @))))
                  (if-moveup
                    (moveup-target
                      (movedown-target-1
                        (grow @)))
                    (moveup-focus
                      (moveup-target
                        (renew @)))))))
            (movedown-focus-1
              (movedown-focus-1
                (grow
```



PROBLEMS:

- What does it do? I'm not sure.
- How can it be used? Unclear.
- Is it even working correctly? I don't know.
- How can the "environment" be affected by anything the organism does if there are no arrows pointing back into the environment? I might just throw an arrow in there anyway.

Overall

- Quite an adventure.
- *Something* occurred.
- If I can add some way for the organism to have outputs or operators on the environment (rather than simply sensors), coupled with the environment it could be an interesting signal processor. It's whole usefulness could come from the ability to supply it with a tailored set of sensors and effectors on the environment.

Final Thoughts

Organisms are inherently physical.

A model/simulation by definition has extraneous parts we willingly ignore. Only the real system includes all of its physical being in the behavior we observe.

All the parts of the organismal system contribute to maintaining and sustaining the operation of that system in the face of entropy.