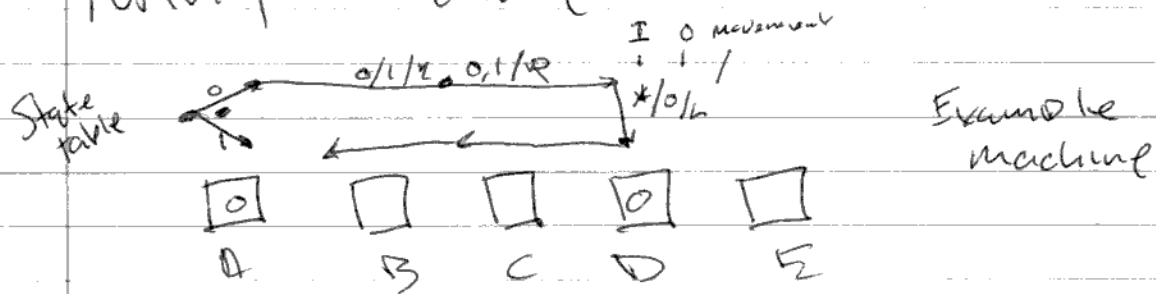


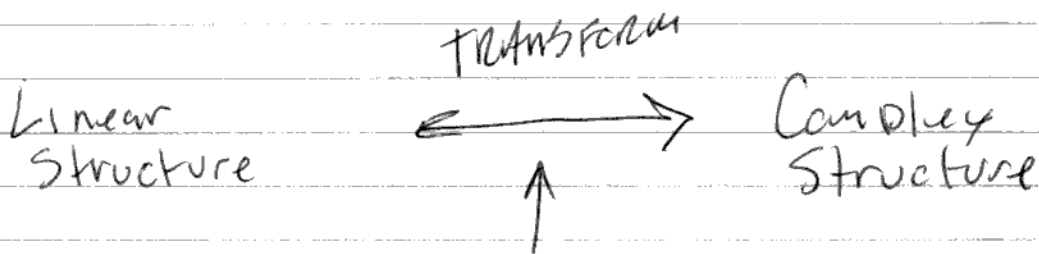
Turing machine



This is a copy $A \rightarrow D$

Not a function style / Copy abstraction

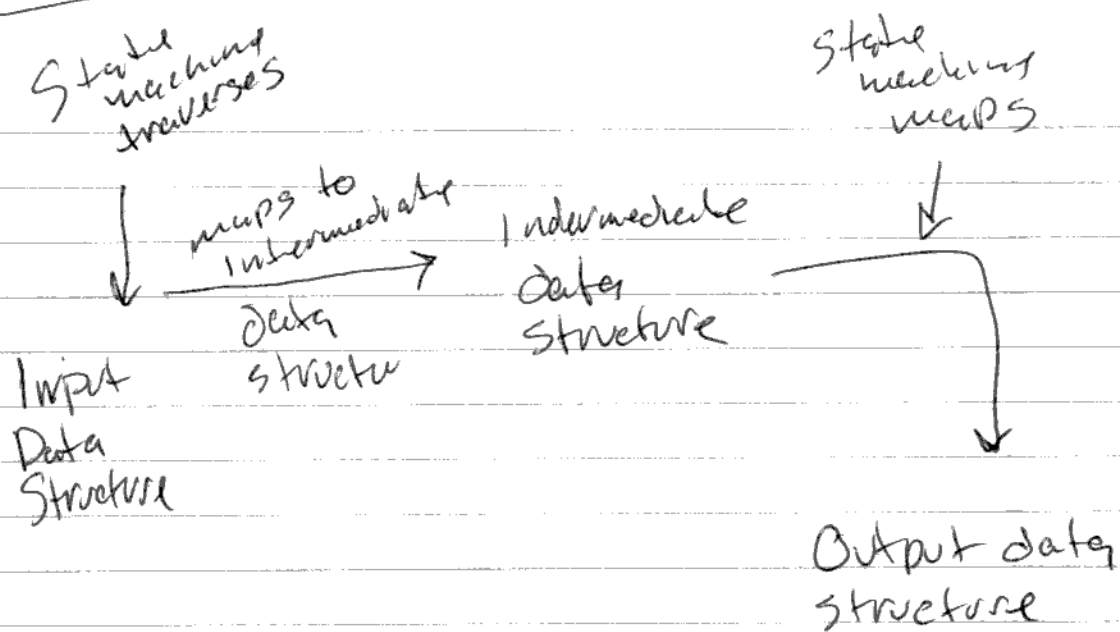
huge state structure required



Transform
Provided
by States
in state space

State Space provides the transforms that convert a linear (tape) structure to and from a more complex data structure

Turing Machine



ALL Three data structures must be traversable. (state, node, Branch)

State machine encodes the steps for traversal

The leaf nodes represent yet another state space.

The state machine handles transitions between the traversable branch space and the traversable node space

This ~~mean~~ means there is an equivalence between structure in the node, branch and state spaces

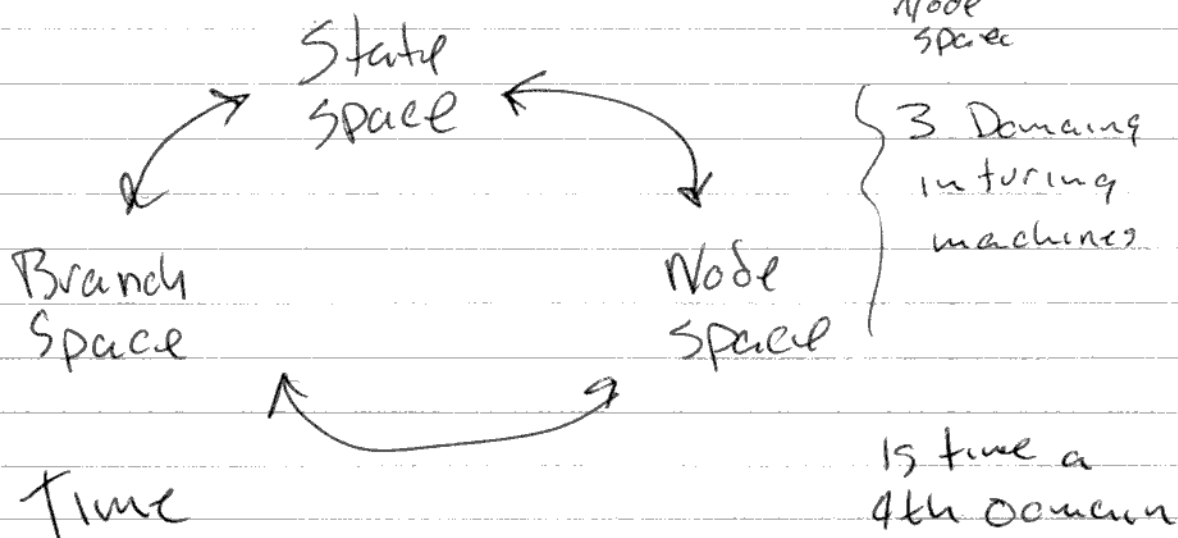
Turing Machine

State space

Branch space

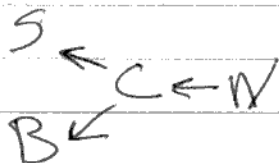


Node space



Any traversable structure in one of the three spaces can be stored or recreated in a combination of the other two.

Computation is simply the process of flowing the structure from 2 to 1 or back?



Compression is simply transform ~~between combinations~~ of traversable structure between S, B, N

turing machine

Does it generalize?

Any traversable data structure

Can be represented as a ~~combination~~
combination of two other
traversable data structures

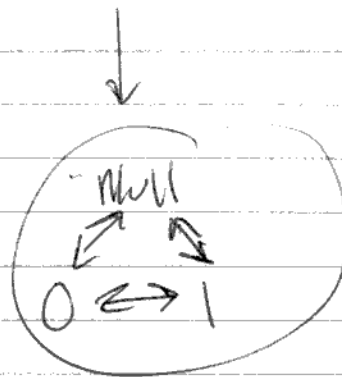
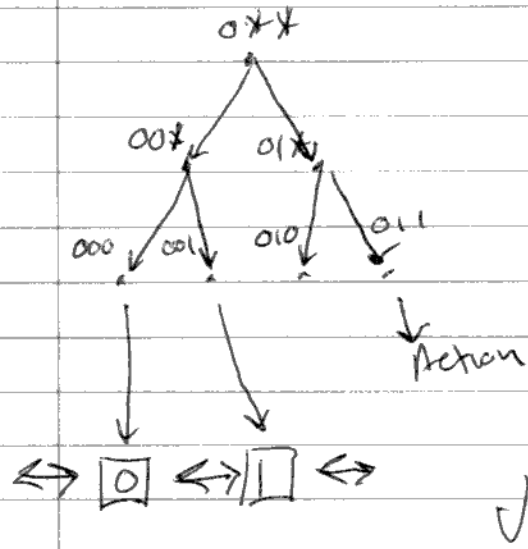
Combination = mapping

A	B		C
a1	b2	\Rightarrow	c5
a2	b7	\Rightarrow	c1

~~Nice~~ ~~Nov 16~~ ~~Hebbs~~



turing machine



turing machine

Transforming is also selecting
subset.