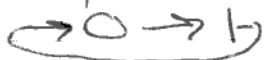


There is a connection between

- Turing machines (3 spaces)
 - o (1) state space / state data structure
 - o (1) linear / list space w/ ∞ nodes
 - o (1) binary enum data struct w/ each node



- Data structure transformation
 - o turing machines can perform any calc
 - o input and output data structures are on the tape
 - o the 3 spaces (state machine, tape and state at position) are evaluated together to transform data
 - o ~~complex~~ transforms from complex structure to linear structure happen when providing input to tape and for receiving output
 - o the state table in a turing machine must encapsulate
 - 1) transform from linear tape to meaningful structure on input
 - 2) transform that alters the complex structure in some way
 - 3) transform that converts meaningful structure back to linear array

There is a connection between (cont)

— Traversing data structures + addressability
o the turing machine operates by traversing data structure 1 to produce data structure 2

— Compression
o the transform nature of turing machine consisting of
1) input data
2) State machine capturing complex data structure in its definition
construction

Seems very similar to compression in terms of

- 1) creating a machine that can produce a data structure from minimum set of variable data
- 2) Storing only the variable data