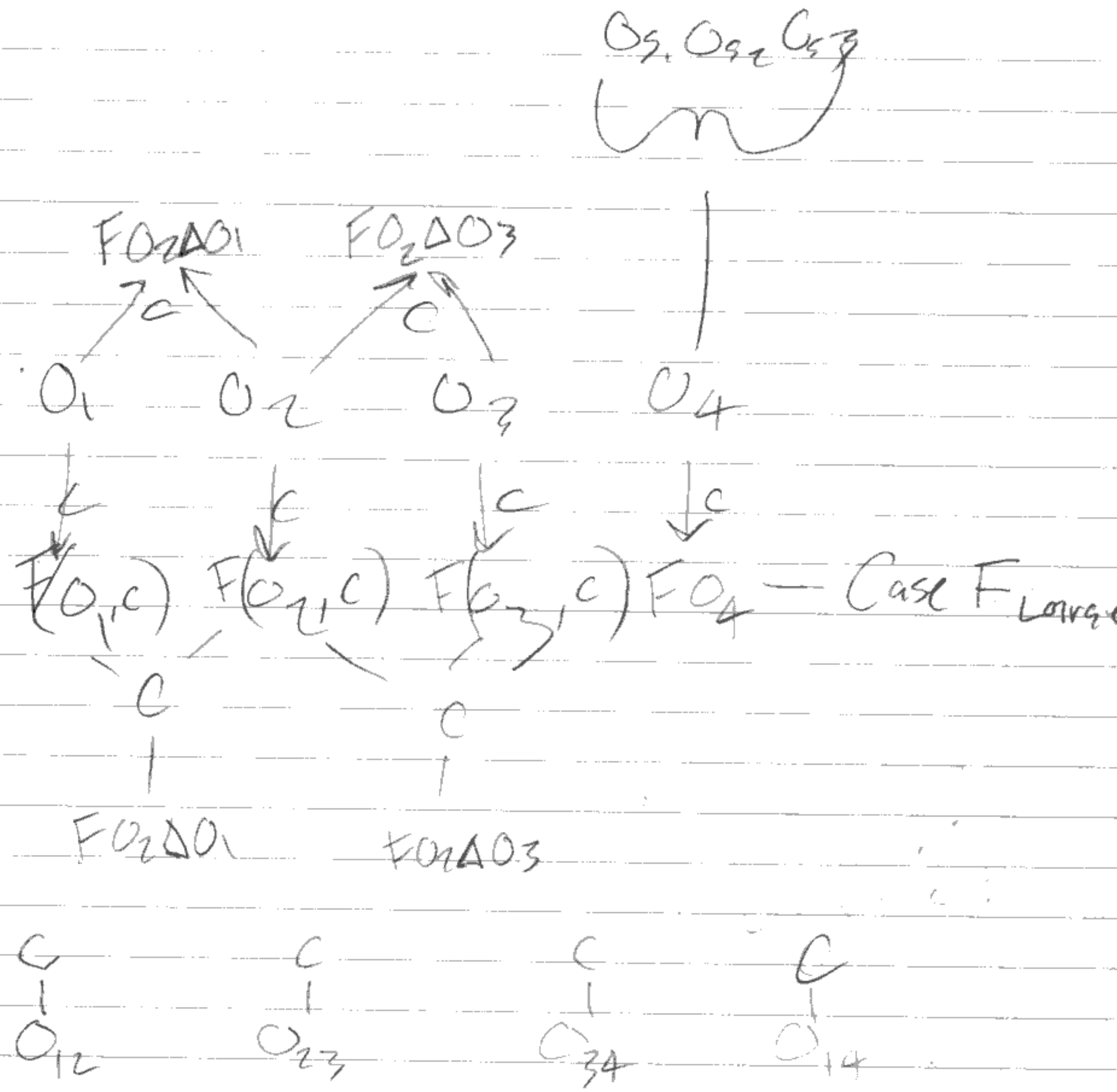
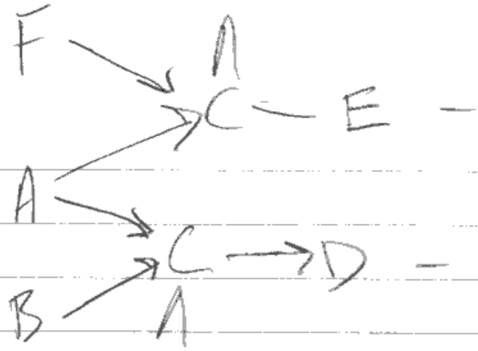
Case sub ~~objects~~
Components Large

EF



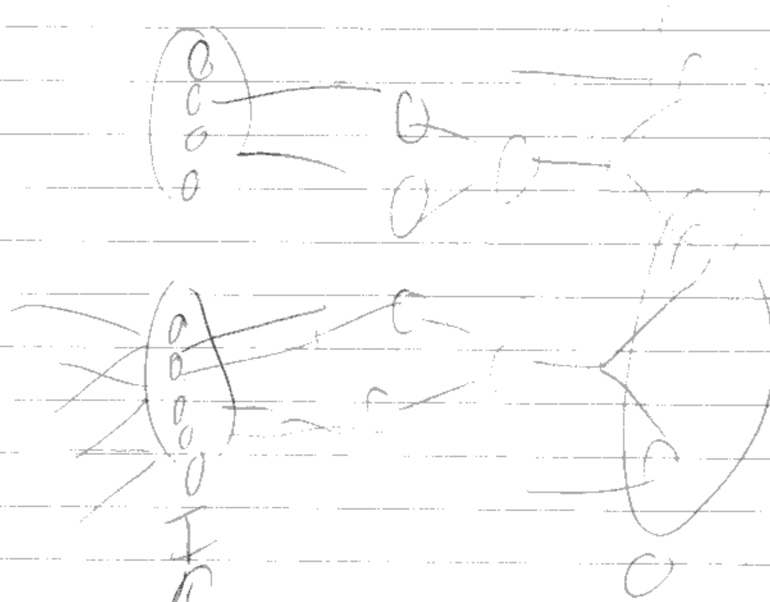
F = Fitness C = Compute



— D exists only if $A \wedge B$

Computes Always Exist

This is a sufficiently defined search



Can map J to C

S = state space

$A \subseteq B$ A subset B

Rule$(\text{IN} \subseteq S, \text{OUT} \subseteq S)$

$G(\text{IN} \subseteq S + \text{out} \subseteq S)$ ← evaluate rule against this

Q) How do we pick in, out?

?? Select in, out such that $G(\text{IN}, \text{OUT})$ has loop?

State Space

$$\# \text{states} = \overset{\#\text{tokens}}{\underset{0}{\overset{i}{\cancel{\#\text{tokens}}\ \sum}}} \left\{ (t(i).max - t(i).min) \cdot \ldots \right\}$$

Limit statespace such that more than one rule will not be triggered by a state?

one miss by a rule excludes the rule?

T = Token Space/Set

$\text{Machine}(T) = G_T$

$V(G)$ = Token Combinations

$E(G)$ = State transitions

$\text{Factory}_R(T) \Rightarrow \text{Machine}(T)$

$\text{Interest Level}\left(\text{Factory}_R(T)\right) \Leftrightarrow IL\left(\text{Machine}(T)\right) \Leftrightarrow IL(G_T)$

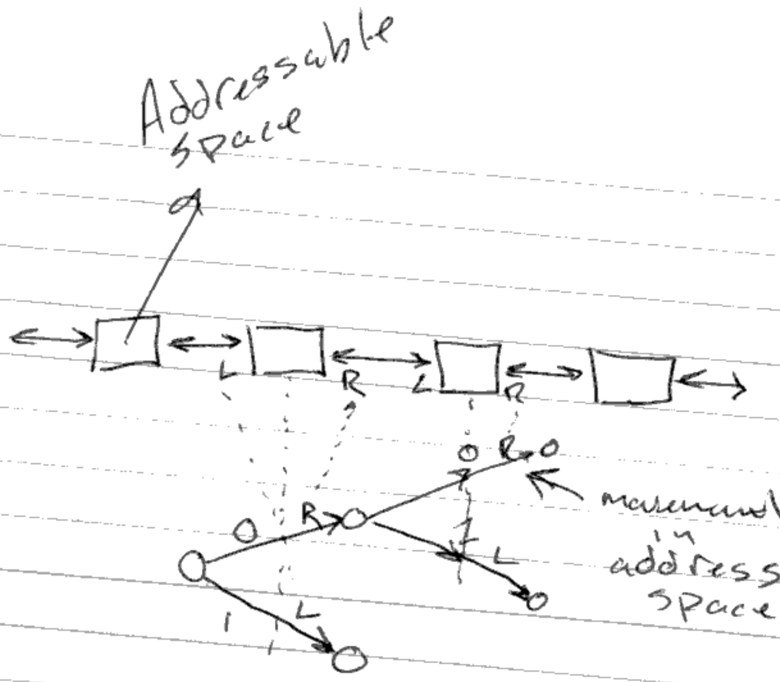
$IL(G_T) \Leftrightarrow \text{Information Density}(G_T)$

```
R.each{|r|
  machine_T = Factory(r, t)

  T.each do |t|
    G_t = Machine t
  end

  f = G_t <=> G̅_t
}
```

? Compare full G(T)?

Can't. O(Big)

Compare what part G(T)



State on memory
Action on Address
Action on State
Constraints

State Space

2 movements from each ~~space~~ state

Address Space

Infinite states

infinite
linear movement
2 movements

Addressable Spaces

Addressability

Paradox

maximum information density
in terms of states / independent domains
seems to equate to a ~~binary~~ hierarchy

Domains C B A	
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

$$\frac{2^n}{n} = \frac{\text{States}}{\text{Domains}}$$

3 bits ~~6~~ / 8 states
2 states/bit

yet there seems to be a great deal of redundant information in the system

For example ~~B~~ C becomes very predictable from A, B.

how can the structure be predictable yet have high information content

$$\frac{2^n}{n} = \frac{\text{states}}{\text{Domains}}$$

$$\frac{2^n}{2 \cdot n} = \frac{\text{States}}{\text{tokens}}$$

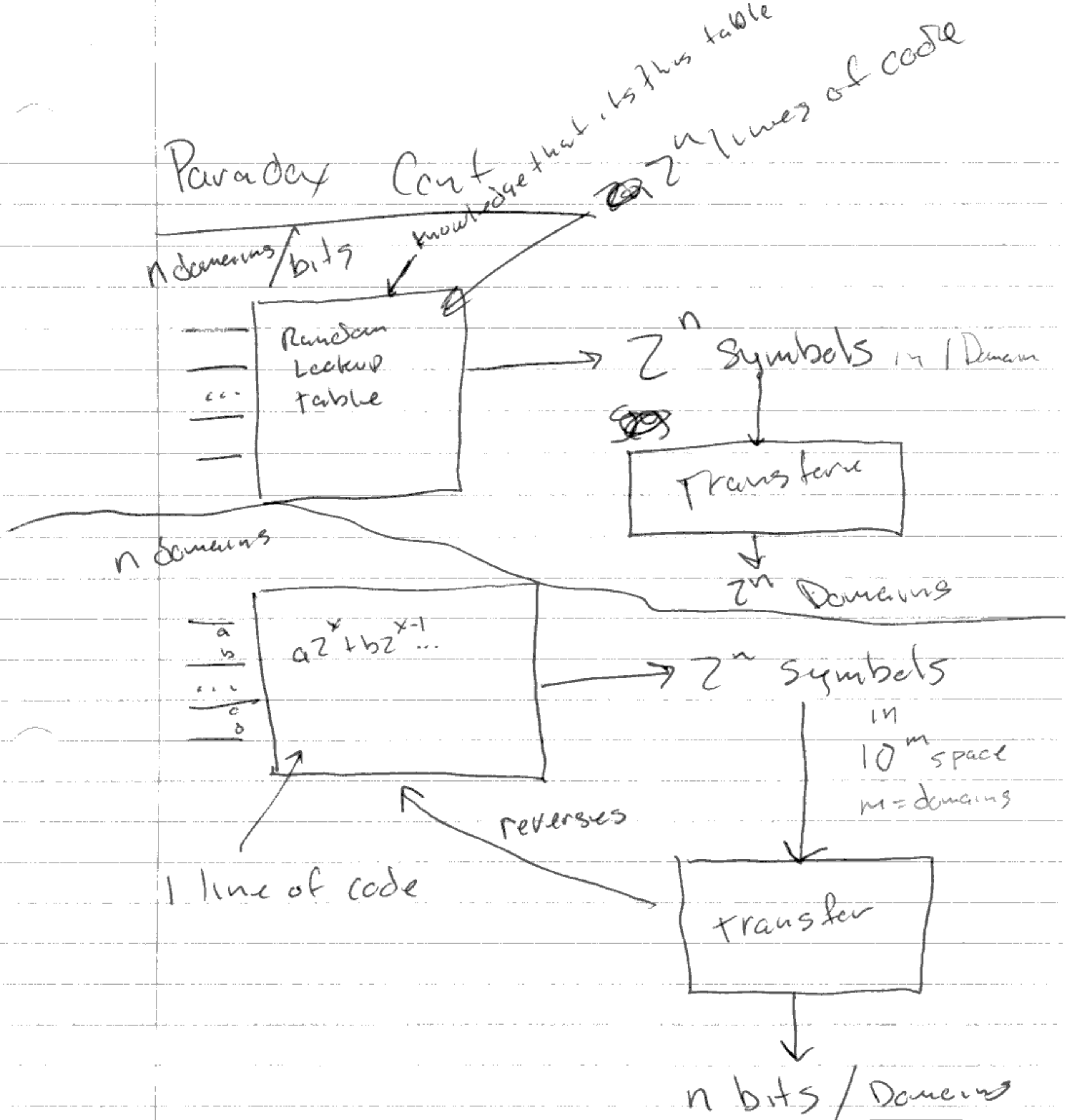
$$\frac{m^n}{m \cdot n} = \frac{\text{States}}{\text{tokens}}$$

m = tokens/Domain
n = # domains

$m = 0$
$m = 1$ $D \Rightarrow 0$ as $n \Rightarrow \infty$
$m = 2$ $D \Rightarrow \infty$ as $n \Rightarrow \infty$
$n = 1$ $D \Rightarrow 1$ as $m \Rightarrow \infty$
$n = 2$ $D \Rightarrow \frac{\infty}{2}$ as $m \Rightarrow \infty$

Q) what is the correct m and n
if each state has a cost "a"
each domain has a cost "b"

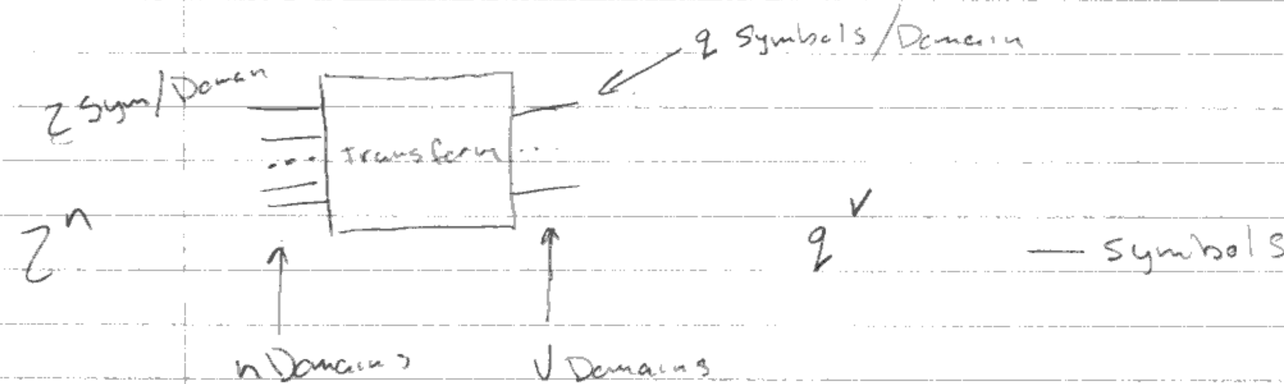
Paradox Cont



top example has maximum information density as states/domains

Paradox Cont

A — B



If $v=1$, $q=2^n$ = # symbols, Transform is:
2^n symbol lookup table

If $v=n$, $q=2$, transform is 1 lookup/Domain

Conditions:
Symbols@A <= # symbols@B
unique a = unique b

Q) how many logic gates to convert input to output?