11/07/18

Pumping lemma for CFG:

$\forall$ CFG L $\exists p$ $\forall w$ ($len(w) \geq p \rightarrow$

$\exists u, v, x, y, z$ ($w = uvxyz$ & $len(vy) > 0$

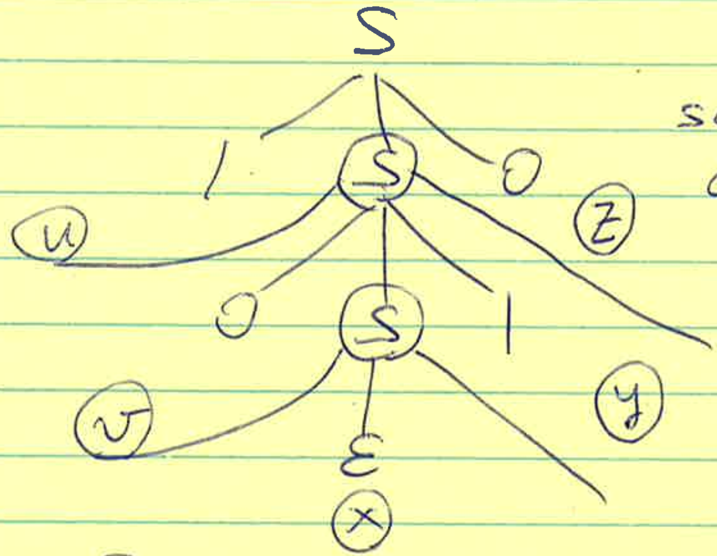
& $len(vxy) \leq p$ & $\forall i$ ($uv^i x y^i z \in L$)))

11/07/18
-2-

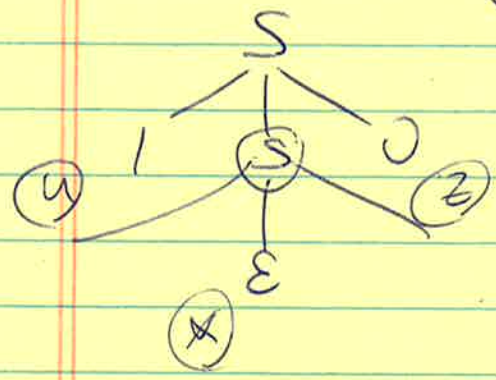
1. $S \to \varepsilon$
2. $S \to 0S1$
3. $S \to 1S0$

$\boxed{11010}$

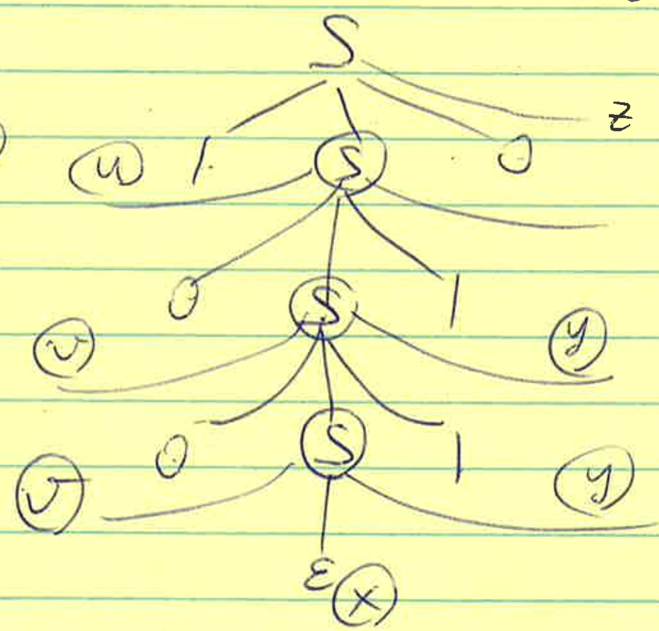
We select 2 lowest repetitions of the same variable on the same branch



$u = 1$
$v = 0$
$x = \Lambda$
$y = 1$
$z = 0$



10



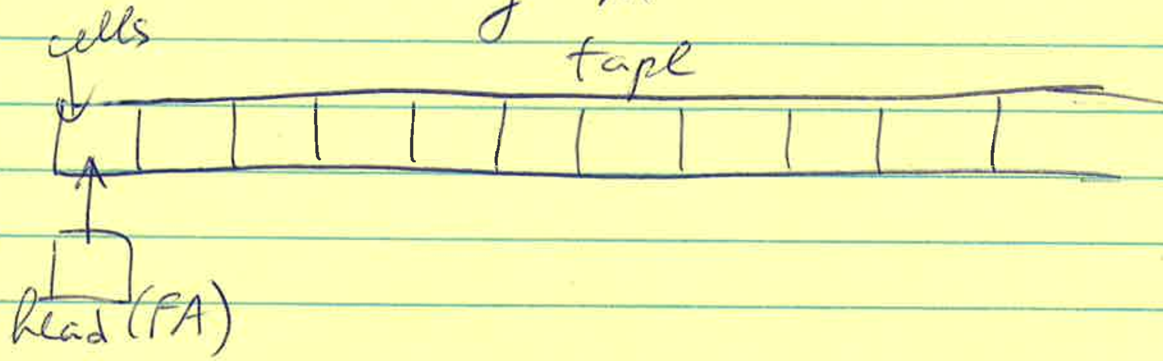
100110

11/07/18
-3-

Turing Machines

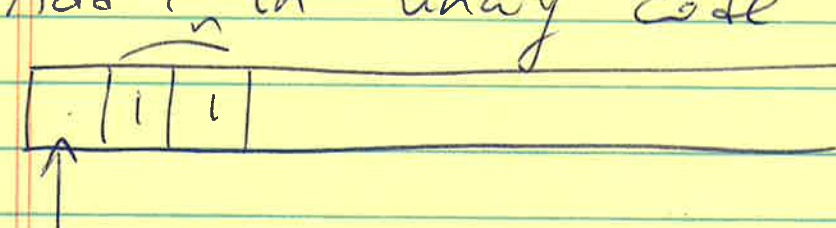
- ~~wh~~ why?
- Who was Turing?
- What are Turing machines?

1932
1934
1936 ⟷
1939

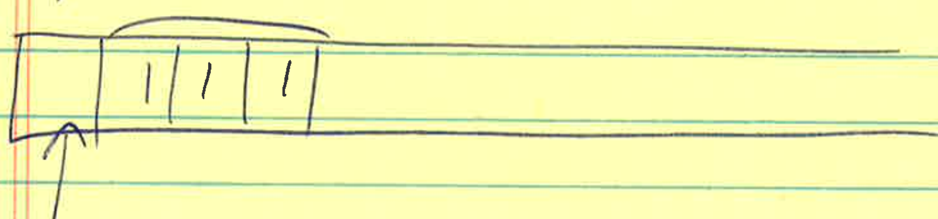


unary code: 0 — ⊔ 1 — 1 2 — 11
3 — 111, 4 — 1111, ...

Add 1 in unary code



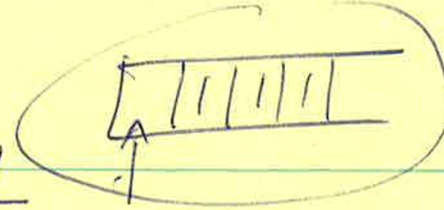
we have

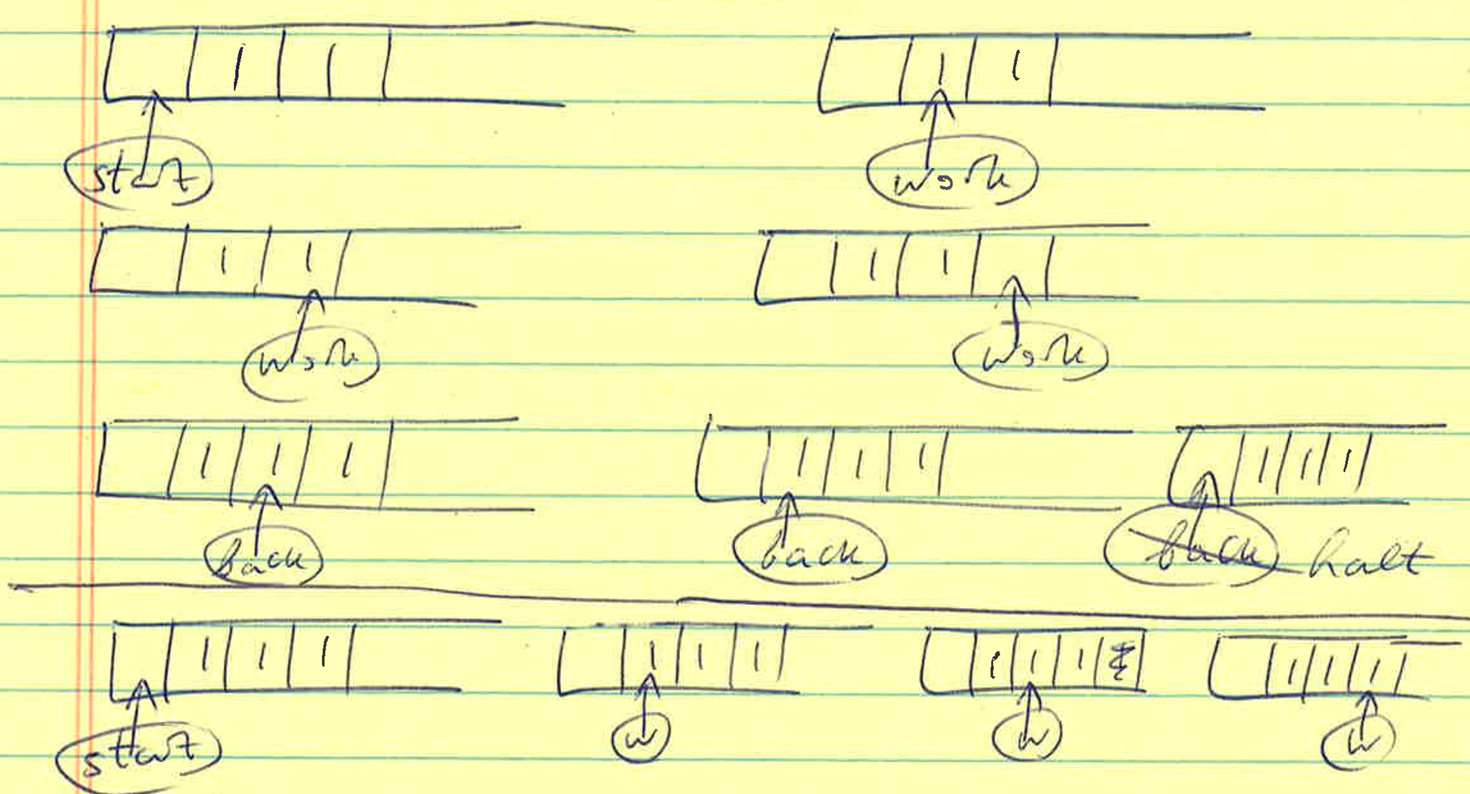


we want.

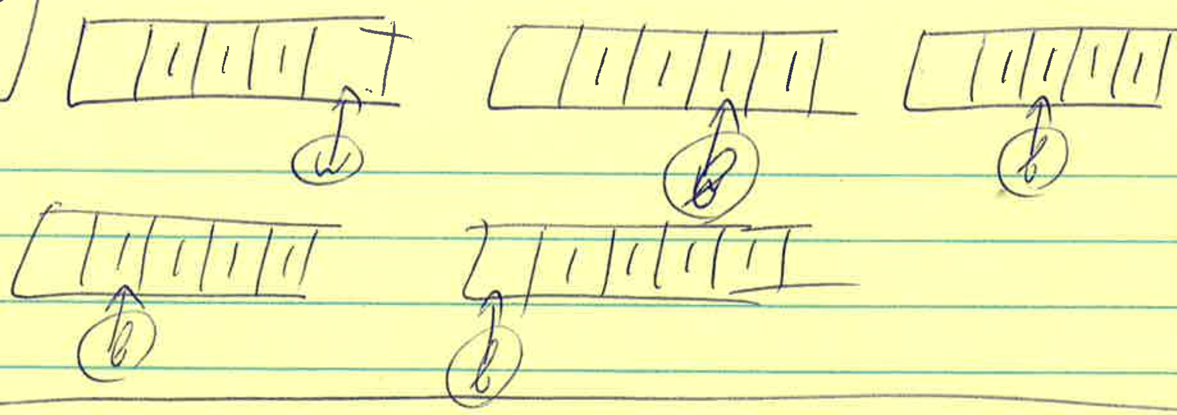
11/07/18
-4-

start, ␣ → R, work
work, 1 → R
work, ␣ → 1, back, L
back, 1 → L
back, ␣ → halt.

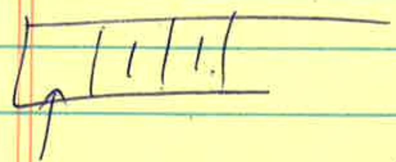




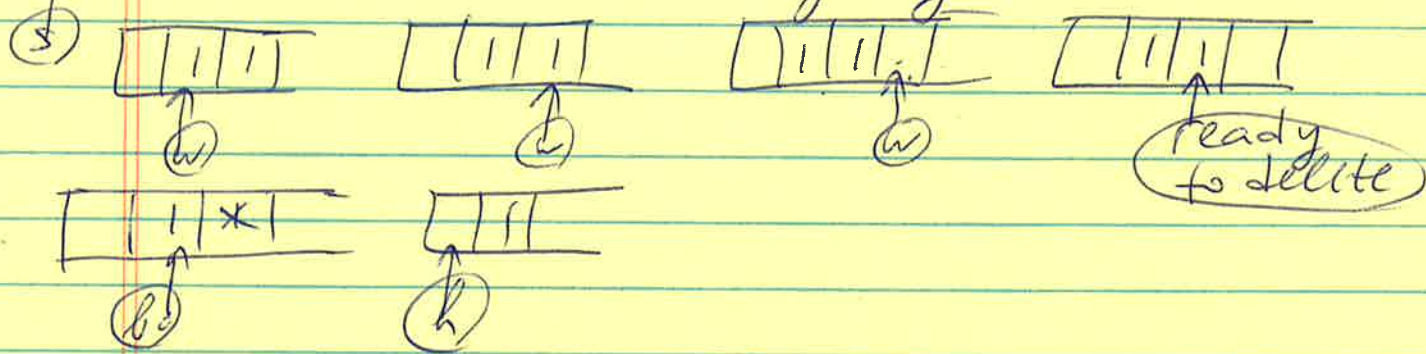
11/07/18
-5-



Subtract 1 in unary code:



we go right until we see blank,
then we go back one step
~~and~~ then replace | w/ blank
& start going back



11/07/18
-6

start, ⌴ → R, working
working, 1 → R
working, ⌴ → L, readyToDelete
readyToDelete, 1 → ⌴, back, L
back, 1 → L , back, ⌴ → halt

Binary numbers: $6_{10} = 110$

4 2 1

We write starting w/ most significant digit
Most computers start w/ least sign. digit

0 1 1

110
11

because all arithm. operations start with least significant digits

$$\begin{array}{r} 13 \\ 18 \\ \hline 31 \end{array}$$

| ⌴ | 0 | 1 | 1 | 1 |

How do you add 1 in binary code?

16 8 4 2 1

$$\begin{array}{r} 10111 \\ 1 \\ \hline 11000 \end{array} \quad \begin{array}{l} = 23_{10} \\ \\ = 24_{10} \end{array}$$

16 8

$$\begin{array}{r} 111 \\ 1 \\ \hline 1000 \end{array} \quad 7$$

Algorithm: ~~you ad~~ You start at least significant digit, replace 1s with 0s until you see 0 or ⌴, then you replace them with 1, and stop.

++

hardware supported

11/07/18

-7-

(start, ⌴ → R, working

working, 1 → 0, R

working, 0 → 1, back, L

back, 0 → L

back, ⌴ → halt