CSCE 355
4/10/2024

TMs & the Church-Turing thesis.

Algorithms = TMs

Machine primitives:

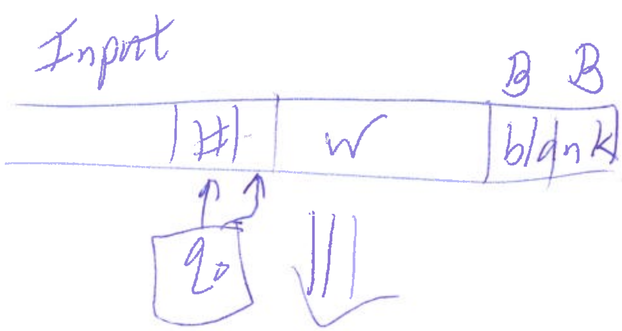
- moving data
- ~~copy~~ copying data
- arith ops
- logic, comparisons

By request:

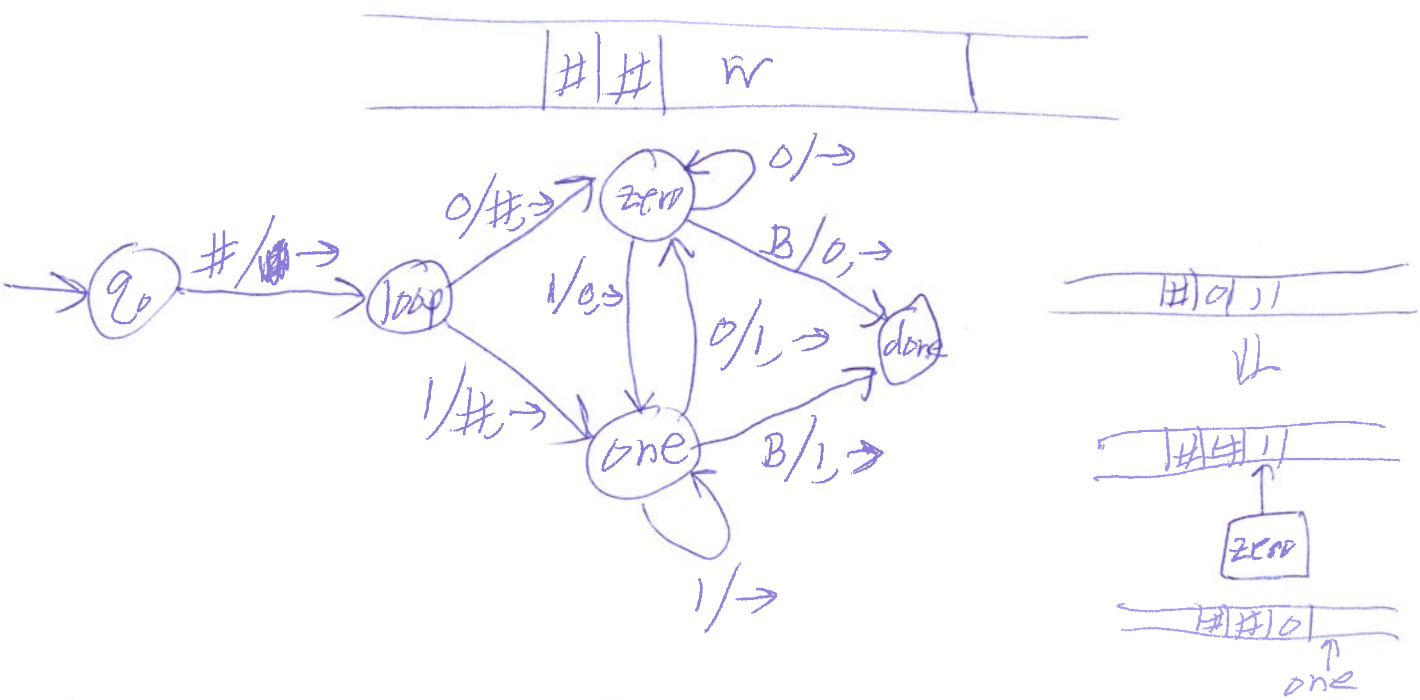
- simulating a PDA

Moving data:

TM to do this:



$w \in \{0,1\}^*$



Simulating a PDA:

Input alphabet is $\{a,b,c\}$ for example

Tape alphabet is $\{a,b,c,\mathring{a},\mathring{b},\mathring{c},B\} \cup \Gamma$

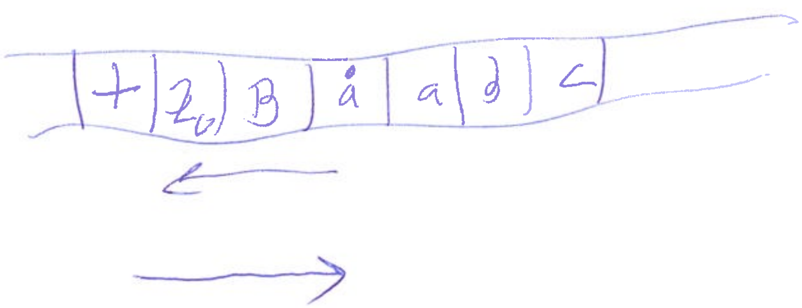
stack alphabet

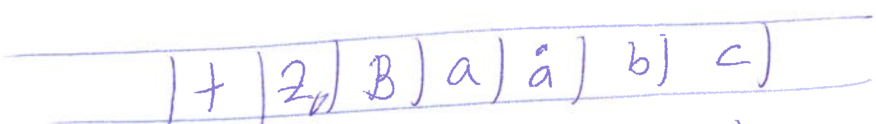
Input: $w \in \{a,b,c\}^*$

$w = aabc$

$\Downarrow$

$\delta(q,a,Z_0)=(r,+Z_0)$

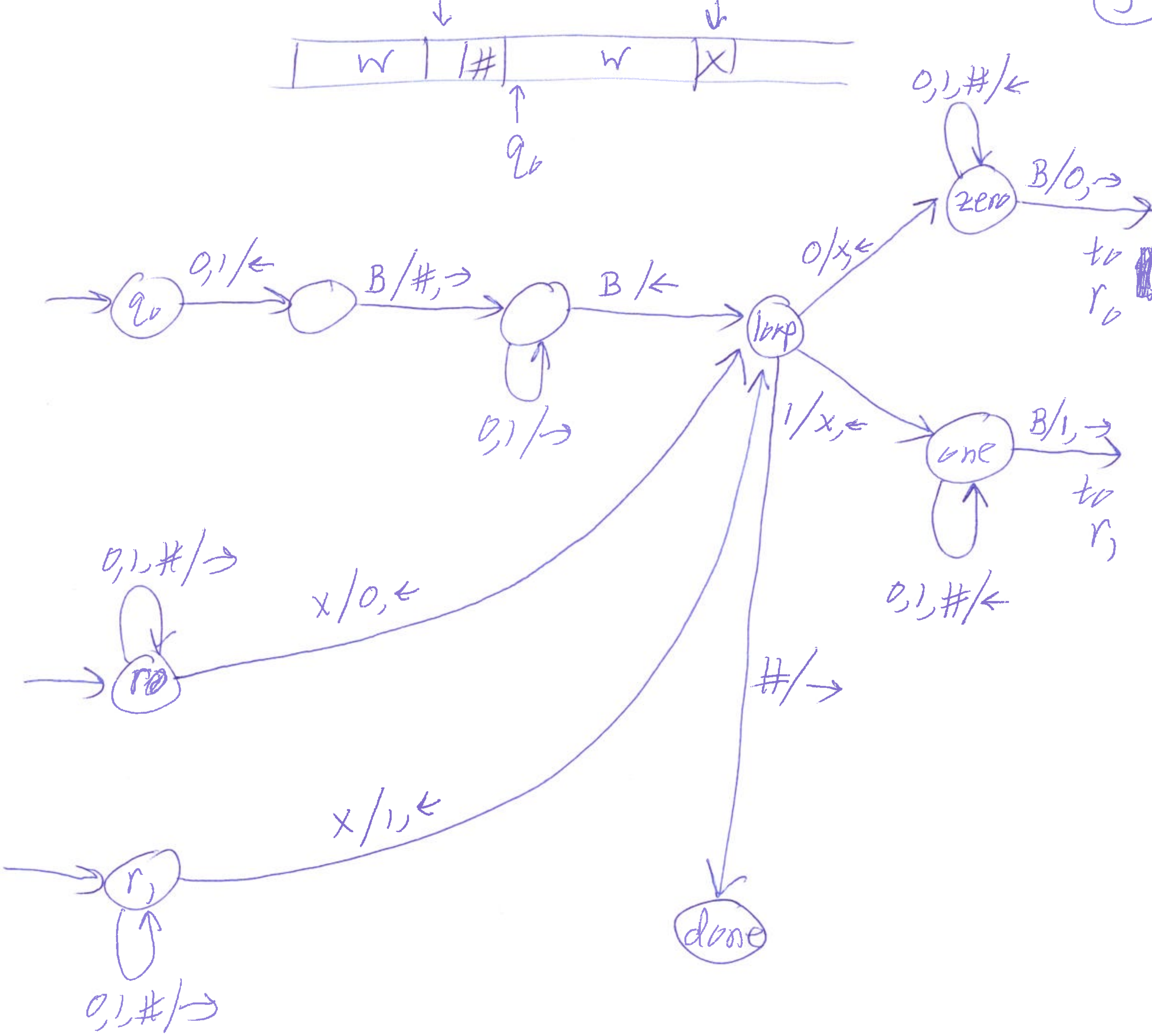




etc.

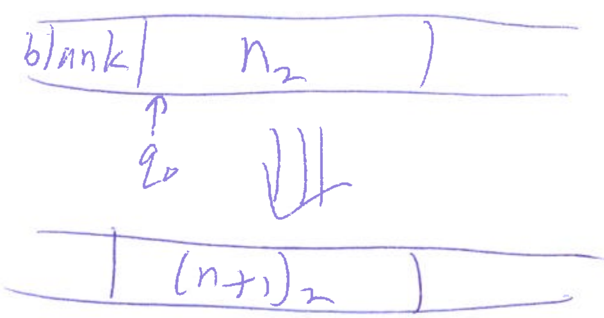
Copying data: Input: $w \in \{0,1\}^*$

Output: $w\#w$



Arith ops: Increment & decrement

Increment:



natural (number n in base 2)

Eg:

$n_2 = 11011$

$(n+1)_2 = 11100$

Truncated subtraction

while $m_2 \neq 0$ and $n_2 \neq 0$

  dec m_2

  dec n_2

[when one becomes 0, the other number has $|m_2 - n_2|$]

Comparison: Input | m_2 # n_2 |

~~Outpu~~ End in one of 3 states:

$<$	$=$	$>$
if $m_2 < n_2$	if $m_2 = n_2$	$m_2 > n_2$

while $m_2 \neq 0$ & $n_2 \neq 0$:

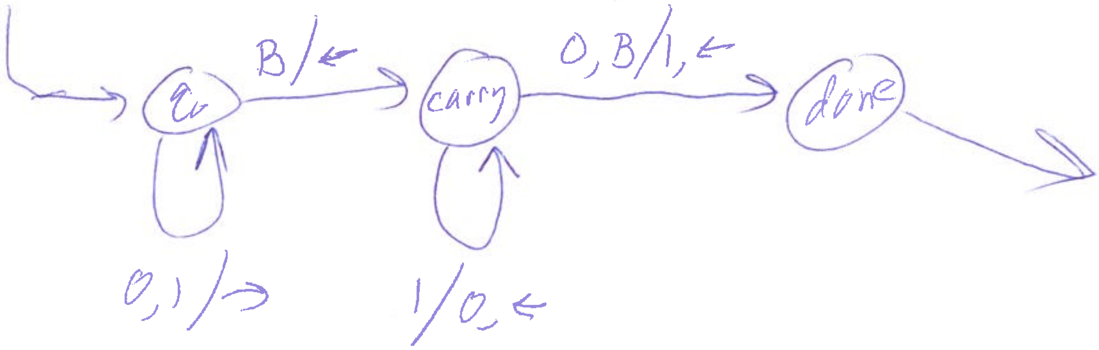
  dec m_2

  dec n_2

if $m_2 \neq 0$ goto $>$

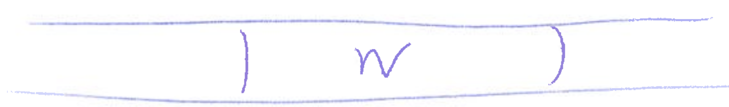
if $n_2 \neq 0$ goto $<$

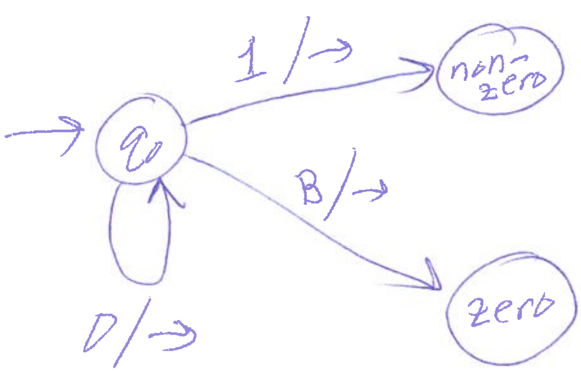
else goto $=$



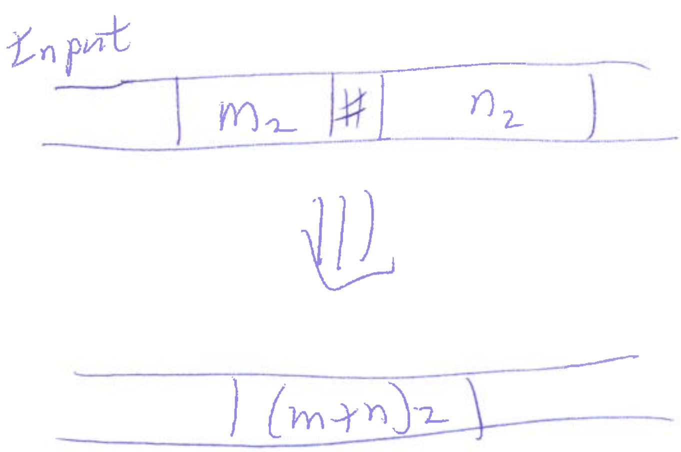
Decrement is similar — different possibilities for decrementing zero:

Testing for zero:



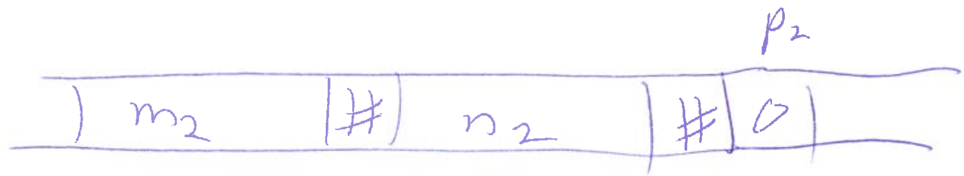


Addition:



Idea:
while $n_2 \neq 0$:
dec n_2
inc m_2

Multiplying:

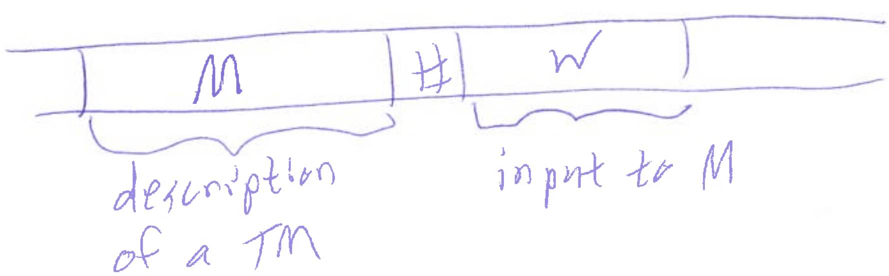


while $n_2 \neq 0$
    dec n_2
    add m_2 into p_2 (copying m_2 first)
// p_2 holds the product

Division is repeated subtraction.

Now assume the Church-Turing thesis to express TMs as high-level algos.

A universal TM U



U simulates M on input w