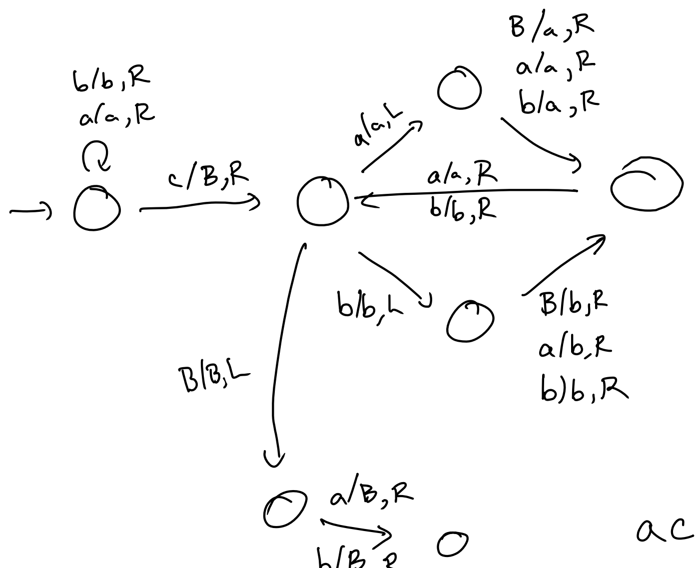
Turing Computable functions

$f(b^n a b^n) = b^n a^2 b^n$ ← last time.

$f(xcx) = xx$ $\quad x \in \{a,b\}^*$

$f(abac\,aba) = abaaba$



aca

aBa

aaaB

Arithmetic on TMs

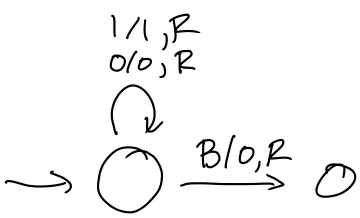
For binary #s,

make a TM for

$f(x) = 2x$

	binary		decimal
like	$\overset{8\,4\,2\,1}{101}$	=	5
binary	1010	=	10

To multiply by 2, we just add 0 on the right end of the #.



$f(x) = x+1$

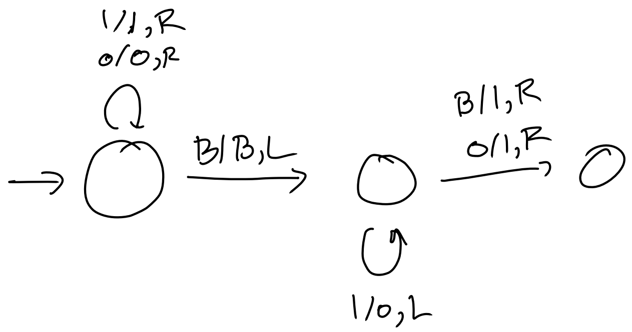
if $x = 11010$

then $x+1 = 11011$

if $x = 11011$

$x+1 = 11100$

$$\begin{array}{r} 11011 \\ +\quad 1 \\ \hline 11100 \end{array}$$



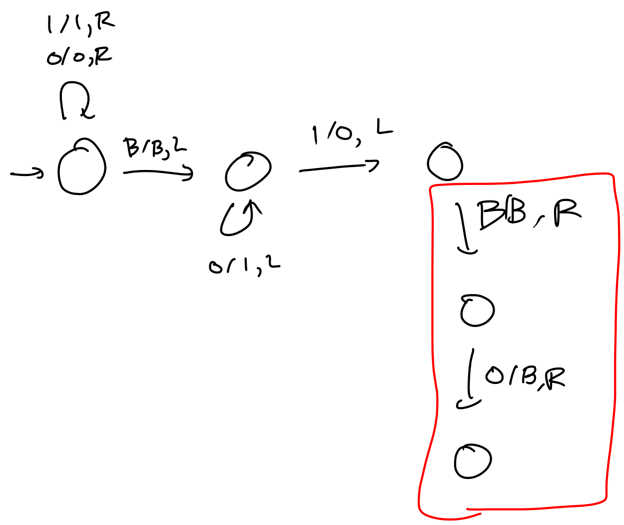
You try $f(x) = x-1$ (assume $x > 0$)

$x = 101101$

$x-1 = 101100$

$x = 101100$

$x-1 = 101011$



1000

0111

What about functions of several variables?

plus(37,42) = 79

We could write on the tape

37,42

or 37#42

or, if convenient, we could stack them

like $\frac{3}{4}\ \frac{7}{2}$

Some simple functions on stacked inputs.

A TM for "bitwise AND"

00110 AND 10101

we do it like

$$\begin{array}{r} 0\,0\,1\,1\,0 \\ \underline{\text{AND}}\quad 1\,0\,1\,0\,1 \\ \hline 0\,0\,1\,0\,0 \end{array}$$

$\frac{0}{0}/0,R$

$\frac{0}{1}/0,R$

$\frac{1}{0}/0,R$

$\frac{1}{1}/1,R$

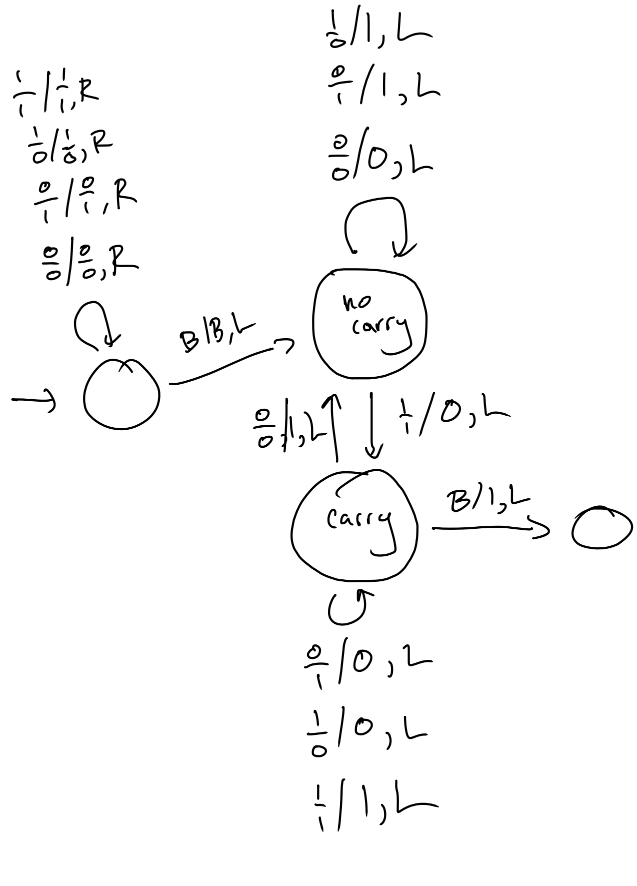
OR is similar.

Bitwise addition (assume we have same # of digits)

$$
\begin{array}{rccccc}
 & & 1 & & & \\
 & 0 & 0 & 1 & 1 & 0 \\
+ & 1 & 0 & 1 & 0 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 1
\end{array}
$$

$\frac{0}{1}$ becomes 1 (if there was no carry)

We need to keep track of if there was a carry or not.



in no carry, if we see $\frac{0}{0}$, we make it 0, & no carry.

$\frac{0}{1}$,

in carry if we see $\frac{0}{0}$, make it 1, no carry