can move left or right

tape

R/W head

finite-state controller

CS 600

blank space, spelled □
Hmmm $\equiv$ Prolog $\equiv$ Racket

Turing Machine
0 jeqzn r0 0

(proofs in)
Predicate Calculus
p :- p.

$\lambda$
Calculus
$((\lambda(x)(x\ x))\ (\lambda(x)(x\ x)))$
Turing Machines FTW!

(1) \( L = \{a^Nb^N | N > 0\} \quad // \text{equality} \)

(2) \( L = \{a^Nb^{2N} | N > 0\} \quad // \text{multiplication} \)

(3) \( L = \{a^Nb^Mc^{(N+M)} | N, M > 0\} \quad // \text{addition} \)
Turing Machines FTW!

(1) \( L = \{\text{All valid Unicalc programs}\} \) ✔

i.e., HW 10
Turing Machines FTW?

(1) \( L = \{ \text{All valid Unicalc programs} \} \)  ✔
i.e., HW 10

(2) \( L = \{ \text{All programs that accept all valid Unicalc programs} \} \)  ✗
Autograding!
Kurt Gödel
This sentence is false.
The Halting Problem

(1) $L = \{\text{All programs that halt and give an answer}\}$

def HC( f, data ):
    if f(data) halts:
        return True
    else:
        return False
The Halting Problem

(1) \( L = \{ \text{All programs that halt and give an answer} \} \)

```python
def flip(f):
    if HC(f, f):
        while True:
            pass
    else:
        return True

flip(flip) !!!
```

undecidable
Hmmm \equiv \text{Prolog} \equiv \text{Racket}

\Downarrow

\text{Turing Machine} \equiv \text{(proofs in) Predicate Calculus} \equiv \lambda \text{Calculus}

0 \text{ jeqzn r0 0} \equiv p :- p. \equiv ( (\lambda (x) (x x)) (\lambda (x) (x x)) )
Turing Machine $\equiv$ (proofs in) Predicate Calculus $\equiv$ $\lambda$ Calculus

0 jeqzn r0 0

Programs $\equiv$ Data

p :- p.

$(\lambda (x) (x x))$

$(\lambda (x) (x x))$
Evaluations
I see you!

www.buzzfeed.com/katienotopoulos/how-anybody-can-secretly-save-your-snapchat-videos
Given a computational problem:

1. Is there a solution?
2. What is it?
3. How good is it?
Programs $\equiv$ Data

What now?