# The CS 5 Post

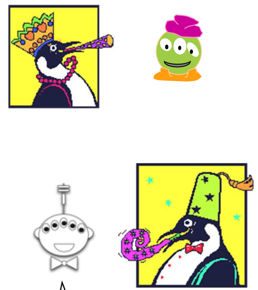
## ALIEN INVASION!!!

Claremont (AP): A party at a private college here was disrupted when uninvited aliens burst through the gates.

"Every year, we celebrate *Long Tall Penguins*," explained an angry student. "We get together, dress like the stuffiest professors, and chip bits off an iceberg to cool our drinks. This year, just as we were about to chill the mackerel, two strange alien creatures ran into the courtyard, picked everyone up, and took turns stacking us in piles."

But another student claimed that the aliens were just misunderstood. "They love to play Connect 4, and since we were wearing black and white clothes, they thought we were playing pieces. They stacked us up in a 5-ply lookahead formation. It was fun!"

According to police, no charges will be filed because the aliens are not subject to Solar jurisdiction.

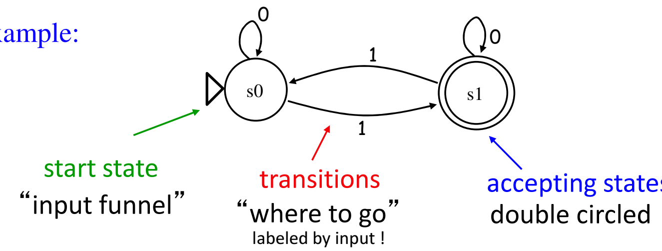


Read 7.1-7.2!

# Simplest Model of Computation

*Finite State Machines*

Example:




# FSM's Can't Count!

Because they're finite, FSMs can only count ***finitely high*** !

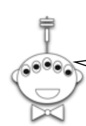
They can handle ***modulo***, but not arbitrary, arithmetic

**Computable with FSMs**

- Even/odd sums or differences
- Multiples of other integers
- Finite constraints on the input:
  - Third digit is a 1
  - Third-to-last digit is a 1
  - Third digit == third-to-last digit
  - etc.

***Uncomputable* with FSMs**

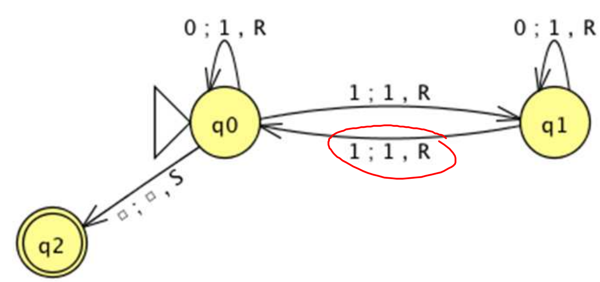
- Equal numbers of two values
- A given difference between two values
- Palindromes
- *Anything modeled by a potentially unbounded* `while` *or* `for` *loop*



Can ***computers*** handle *abitrary* arithmetic?

# FSM's Can't Count!

**So let's build a better machine!**



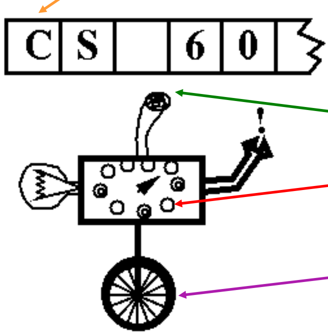

**"Turing Machine"**

# Turing Machines

Alan Turing: "Logical Computing Machines"

A simple model of *universal* computation

Starts on the leftmost character




The *tape*: an unbounded amount of memory. Consists of *cells*, each containing exactly one character (e.g. 0, 1, or (blank))

Read/write head for the tape

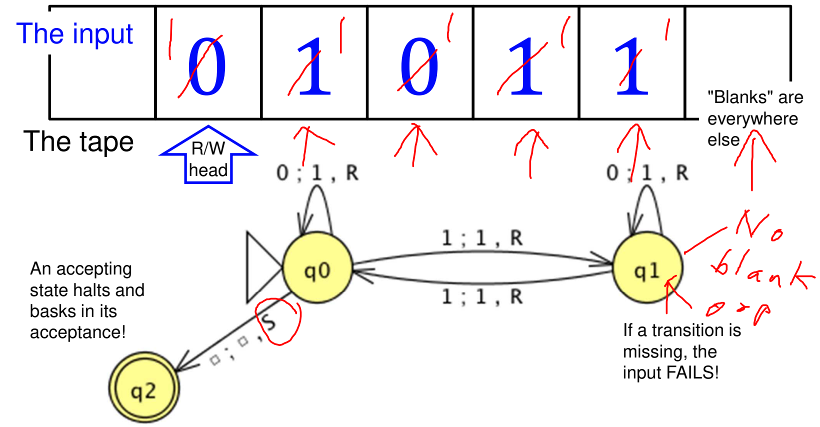
The control: a finite amount of memory, the *control states*—some are accepting, some are not.

Ability to move left and right

The complete state of a TM is determined by:

- The control state
- The symbol now under the head
- The symbols to the right of the head
- The symbols to the left of the head

# Turing Machines

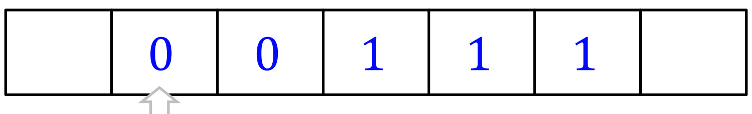



**A Turing Machine rule:** 0 ; 1 , R

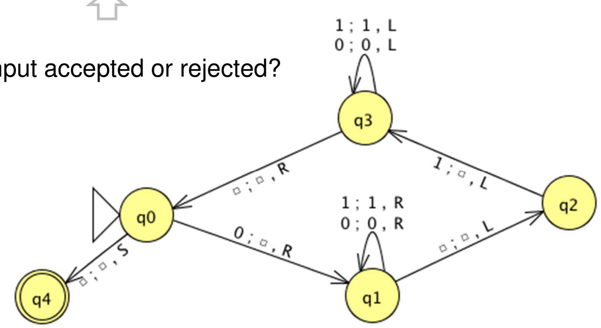
READ SYMBOL WRITE SYMBOL MOTION

Try it in JSFLAP!

# Worksheet




Is *this* input accepted or rejected?

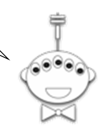



What inputs are accepted *in general*? How does it work?

**Extra:** How could you change this to accept *palindromes?* (just a thought experiment)

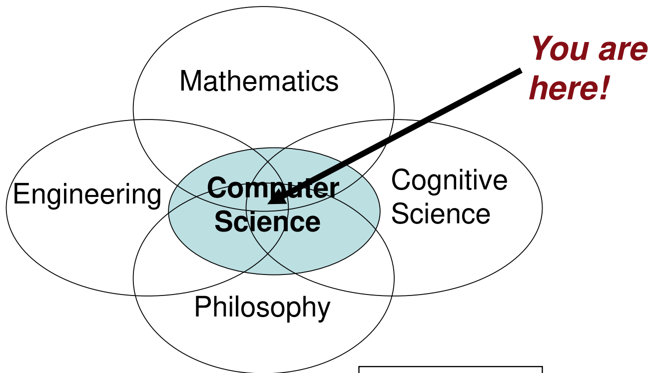
# The Alien's Life Advice

Accept contradictions



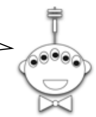
The world isn't 100% math!

# Uncomputability!



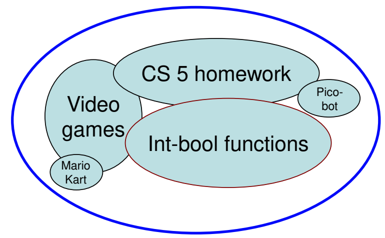

Notice that CS has more eccentricity than the others!

I don't understand your elliptic statement!



# Uncomputable Functions?

There are **well-defined *computational* problems** that **no computer program** can solve even with ∞ memory!




Well-defined computational tasks

```
def is_even(X):
  return X%2 == 0

def is_three(X):
  return X == 3
```

Programs

# Uncomputable Functions?

There are **well-defined *computational* problems** that **no computer program** can solve even with ∞ memory!

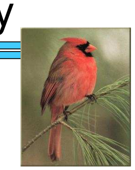
There are more of these…

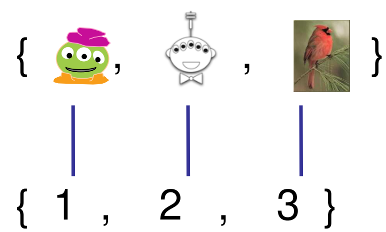
Well-defined computational tasks

than these!

Programs

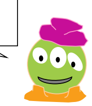
# A Brief Digression on Cardinality



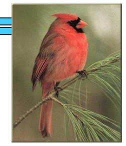



I saw some of this in Math 19 already…

Hang on though, there's some new stuff here!



## A Brief Digression on Cardinality

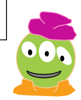


{  }

{ 1 , 2 , 3 }

A perfect matching is called a "bijection"





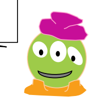


## To Infinity and Beyond!

N = {1, 2, 3, 4, 5, … } ← The "natural" numbers

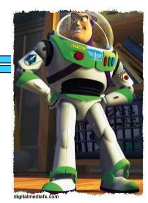
E = {2, 4, 6, 8, 10, … } ← The positive evens







## To Infinity and Beyond!



N = {1, 2, 3, 4, 5, … } ← The "natural" numbers

E = {2, 4, 6, 8, 10, … } ← The positive evens

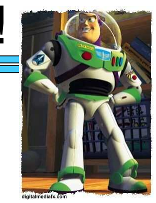
Z = {…, -4, -3, -2, -1, 0, 1, 2, 3, 4, …} ← The integers

Q = { -3/42, 1/2, 2/3, … } ← The rationals





## To Infinity and Beyond!



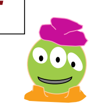
N = {1, 2, 3, 4, 5, … } ← The "natural" numbers

E = {2, 4, 6, 8, 10, … } ← The positive evens

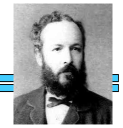
Z = {…, -4, -3, -2, -1, 0, 1, 2, 3, 4, …} ← The integers

Q = { -3/42, 1/2, 2/3, … } ← The rationals

These sets are said to be ***countably infinite.***



# Cantor Diagonalization



Georg Cantor
1845-1918

Proving that the set [0, 1) is uncountably infinite!

[0, 1) = The set of real numbers between 0 and 1

Your claim: you have a way to list all real numbers in order so you can match them to the integers

Cantor's claim: you left something off the list

# Cantor Diagonalization

Georg Cantor
1845-1918

Your list:

```
1   0.10000000000000000000…
2   0.33333333333333333333…
3   0.31415926535897932384 6…
4   0.4242424242424242424
5   0.91711902109999999999…
```

Cantor's number:

```
N   0.21356…
```

# The Tragic Story of Georg Cantor

**"I don't know what predominates in Cantor's theory—philosophy or theology, but I am sure that there is no mathematics there."**
- Leopold Kronecker

Leopold Kronecker
1823-1891

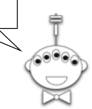
**"No one shall expel us from the paradise that Cantor has created for us."**
- David Hilbert

David Hilbert 1862-1943

# A Bag of Reals

I'm going to reach into this bag of real numbers and pick one out!

Wow, that's a fun game!



# What's Computation Got to Do, Got to Do With It?

Plan:

- Show that the number of Python programs is *countably infinite* *(a small infinity)*

- Show that the number of possible "computational tasks" is *uncountably infinite* *(a **big** infinity)*!

Conclusion: ?

# Computational Tasks?

Plan:

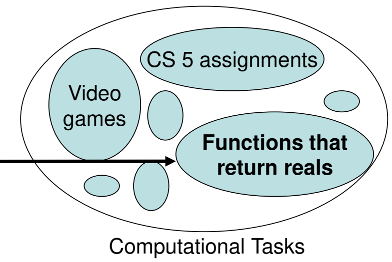
- Show that the number of Python programs is *countably infinite (a small infinity)*

- Show that the number of possible "computational tasks" is *uncountably infinite* *(a **big** infinity)*!

positive integer input

```
def pi():
  return 3.14159265…
```

Real output!




# Counting Python Programs

1. The empty string is a Python program. So is "a".

2. After "z" we could write "+", "-", etc. Most of those are illegal Python programs—but we don't care!

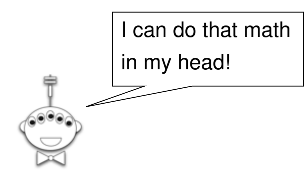
3. Now we do the 2-character programs, then 3, etc.

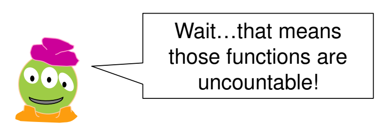
4. Lots of these programs do nothing—but again we don't care!

# Functions

Consider all the **constant** mathematical functions $f(N) = x$, where $x$ is a real number from 0 to 1:

- $f(N) = 0.5$
- $g(N) = 0.707107\ldots$
- $h(N) = 0.314159\ldots$



# Functions and Programs

We know that programs are countable...

…and even simple functions are uncountable…

…so there must be more functions than programs…

…and therefore there are functions that can't be computed!

# Next Time