CS 5 finale

Looking back! Evals, Ideas
Looking ahead? Options...

CS 5: Final lecture now!
I’ll be back...

CS 5, on the verge of termination

CS 5 Final Projects
due this Fri. eve 5/3 @ 8 (Sat. w/ Euro)
LAC hrs today & tutoring hours through Friday...

CS 5 Review Session
This time on Thurs., 5/9 in B442
- to go over the practice exam problems + q’ns

CS 5 Final Exam
comprehensive – some choice of problems
Tues. 5/14 at 2pm or 7pm
Accommodations nearby rooms
Two pages of notes are OK...

The CS view of the world...

Input ——> Algorithm ——> Output

CS’s challenge

Uncomputable functions

Input ——> Algorithm ——> Output

Sometimes an algorithm simply does not exist...
more precisely: every possible algorithm contains bugs!

Haltchecking is uncomputable.

hc ( f )
~ returns whether f() halts or not

hc always has a bug!
**Haltchecking** is uncomputable.

It is impossible to write a (bug-free) function \( \text{hc} (f) \) that determines if a function \( f \) halts when run:

1. \( \text{hc}(f) \) returns True if \( f() \) halts and
2. \( \text{hc}(f) \) returns False if \( f() \) loops infinitely

Suppose \( \text{hc}(f) \) worked for all \( f \)

Create this BFF:

```python
def BFF():
    if \( \text{hc}(\text{BFF}) \) == True:
        while 1+1==2: print 'Ha!'
    else:
        return  # halt!
```

Is \( \text{hc}(\text{BFF}) \) == True?
Is \( \text{hc}(\text{BFF}) \) == False?

\( \text{hc} \) always has a bug  Proven!

```
def f1():
    return 42

def f2():
    while 1+1==2:
        print('Ha')
    return 42

def f3():
    if f1() or f2():
        return 5
```

\( \text{hc}(f1) \)
\( \text{hc}(f2) \)
\( \text{hc}(f3) \)

What should \( \text{hc} \) return for each of these?
And this is important because ...

∞ loops are **undetectable**

...some are detectable, but some are not — and there's no way to know!

bugs are **inevitable**

...infinite loops are just one type of bug... 
In general, they're all undetectable

programming is **not automatable**...

...not perfect programming, at least

---

**Meaningful functions?**

Input → Algorithm → Output

but nearly all *meaningful* functions *are* computable...

---

**CS 5's examples...**

Input → Algorithm → Output

Connect 4 Board → intelligent move
Input text → Markov model + new text
current gener. of “life” cells → next gener. of “life” cells
"Science without religion is lame, religion without science is blind."
"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."
"Duct tape is like the force, it has a light side, a dark side, and it holds the world together"
"If you die in an elevator, be sure to push the Up button."
"All generalizations are false, including this one."
"Clearly you've never been to Singapore!"
"Luke, I am your father." "To be, or not to be." "You shall not pass!"

(… all with authors …)

Markov-generated wisdom!

Connect-4 tiebreaking...

scoreBoard would give each of these 50.0 but which board is really better for black?

Ex. Cr.: write a scoreBoard4Tourney f’n.

Fortunately, nearly all meaningful functions are computable...

but this doesn’t mean we know how to compute them (yet)!

for example …
**Computer Vision Problem**

**Input** → **Algorithm** → **Output**

- **Input**: Image
- **Algorithm**: Smiling detection
- **Output**: Whether the image is of a woman, smiling.

### Actual Image

- **Actual input**: Pixels
- **Actual image**: Image of the Mona Lisa
- **Actual output**: Contents

- **Happily, this problem is being solved...**
- **but why is it so challenging?**

**Notes**

- 2D array of ints
- 16th century woman, smiling (maybe...)
- Mona Lisa being an image... of what?
What's red?

Goal: a coke-can collecting robot...

Subgoal for July

Analysis of scenes consisting of non-overlapping objects from the following set:

- balls
- bricks with faces of the same or different colors or textures
- cylinders.

Each face will be of uniform and distinct color and/or texture.
Background will be homogeneous.

Extensions for August

The first priority will be to handle objects of the same sort but with complex surfaces and backgrounds, e.g., cigarette pack with writing and bands of different color, or a cylindrical battery.

Then extend class of objects to objects like tools, cups, etc.

Try it! Illusions? What computations is your brain doing to cause them? Why?
Are these four lines parallel?
Try it! Illusions? What computations is your brain doing to cause them?

Vision is more challenging than it might seem on first "glance"!

Why?

we don't always give our own vision system credit for all the work it's doing…
Let it go!

Door == coke can? We'll work around it!

The coke-can collector: seeking...

so many pixels!

Actual output: contents people, walking...

CS 5: the past...

Functions & variables
Recursion
Representations (binary, ascii)
Circuit design & Hmmm
Loops, 2d arrays
Dictionaries
Objects and Classes
Computability

Caesar cipher
4-bit multiplier
Mandelbrot, Life
Markov Text Gen.
Date, C4, Project
Finite state machines
Uncomputable functions
However you're CSing, enjoy!

6/28 3:18:53

But what if I'm nostalgic for CS 5 itself?

Consider grutoring for CS5 next term or beyond...

6/28 3:18:53

However you're CSing, enjoy!

No matter what path you choose, it's likely to be in binary...
Thank you for joining CS5!

Good luck on all finals (projects, exams, papers...)

Final Projects: due *Friday* evening...

Exam: **Tues, 5/14 @ 2pm or 7pm**

Here in Shan B442

**REVIEW**: Optional review session covering the practice final and any other questions...

This time next week in Shan B442