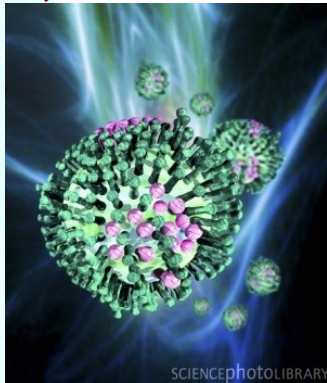


Uncomputability, Viruses, OOP



Avian Flu Virus

One-Slide Summary

- In a **proof by contradiction**, to show that A cannot exist you show that A implies the existence of X , some known-to-be-impossible thing. To show that A is **undecidable** or **uncomputable**, pick X to be the halting problem (i.e., **halts?**).
- Determining if a program is a **worm** or a **virus** is undecidable. Sad face.
- **Object-oriented programming** encapsulates state and methods together into **objects**. This **hides** implementation details (cf. **inheritance**) while allowing methods to operate on many types of input.

#2

Outline

- Proof By Contradiction
- Undecidability Examples
- Worms and Viruses
- OOP
 - Sketchpad
 - Simula
 - PARC
 - Smalltalk



#3

From the “blaming the victim” dept

- *Some* (but by no means all or even most) students expect the TAs to help them get started on a problem when they “don’t even know how to begin”.
- At this point in CS 150 this is *unacceptable*
 - I have not made this clear, so let me do so now.
- Happiness in CS 150 = managing expectations
- You should *not* expect that merely spending time on an assignment will cause you to master it (cf. entitlement).
 - Merely spending time at a piano won’t make you a pianist.
 - You are not doing yourself any long-term favors (e.g., exam2, the final) if you can’t even get started without the TAs.

#4

Concrete Suggestions

- If a problem initially mystifies you:
 1. Review lecture notes from Class 4, slide 35+
 2. **Step back and take out a piece of paper.**
 3. **Write down the inputs (and their types: list, int, etc.).**
 4. **Write down the outputs (and their types).**
 5. **Write down some example input-output pairs.**
 6. **Is it a recursive procedure or not?**
 7. **If it is, what gets smaller? Hint: pick one of the inputs.**
 8. **Does it have any conditional behavior? If so, what?**
 9. **Write out in your own words in English what the procedure should do.**
 10. **Come to Weimer’s office hours.**
- I have instructed the TA’s to skip “what should I do?” questions from people who cannot show 2-9 on a piece of paper.

#5

Informal Proof

(define (**paradox**)
 (if (halts? ‘paradox)
 (loop-forever)
 #t))

If paradox halts, the if test is true and it evaluates to (loop-forever) - it doesn’t halt!

If paradox doesn’t halt, the if test is false, and it evaluates to #t. It halts!

#6

Proof by Contradiction

Goal: Show A cannot exist.

1. Show X is nonsensical.
2. Show that if you have A you can make X .
3. Therefore, A must not exist.

$X = \text{paradox}$

$A = \text{halts? algorithm}$

#7

How convincing is our Halting Problem proof?

```
(define (paradox)
  (if (halts? 'paradox)
      (loop-forever)
      #t))
```

If `contradict-halts` halts, the `if` test is true and it evaluates to `(loop-forever)` - it doesn't halt!

If `contradict-halts` doesn't halt, the `if` test is false, and it evaluates to `#t`. It halts!

This "proof" assumes Scheme exists and is consistent! Scheme is too complex to believe this...we need a simpler model of computation (in two weeks).

#8

Evaluates-to-3 Problem

Input: A procedure specification P

Output: **true** if evaluating (P) would result in 3; **false** otherwise.

```
> (eval-to-three '(lambda () (+ 2 1)))
```

```
#t
```

```
> (eval-to-three '(lambda () (* 2 2)))
```

```
#f
```

Is "Evaluates to 3" computable?

#9

Proof by Contradiction

Goal: Show A cannot exist.

1. Show X is nonsensical.
2. Show that if you have A you can make X .
3. Therefore, A must not exist.

$X = \text{halts? algorithm}$

$A = \text{evaluates-to-3? algorithm}$

#10

Undecidability Proof

Suppose we could define `evaluates-to-3?` that decides it. Then we could define `halts?`:

```
(define (halts? P)
  (evaluates-to-3?
   '(lambda () (begin (P) 3))))
```

if `#t`: it evaluates to 3, so we know (P) must halt.

if `#f`: the only way it could not evaluate to 3, is if (P) doesn't halt. (Note: assumes (P) cannot produce an error.)

#11

Hello-World Problem

Input: An expression specification E

Output: **true** if evaluating E would print out "Hello World!"; **false** otherwise.

Is the *Hello-World Problem* computable?

#12

Uncomputability Proof

Suppose we could define `prints-hello-world?` that solves it. Then we could define `halts?`:

```
(define (halts? P)
  (prints-hello-world?
   '(begin ((remove-prints P))
           (print "Hello World!"))))
```

#13

Proof by Contradiction

Goal: Show A cannot exist.

1. Show X is nonsensical.
2. Show that if you have A you can make X .
3. Therefore, A must not exist.

$X = \text{halts?}$ algorithm

$A = \text{prints-hello-world?}$ algorithm

#14

Liberal Arts Trivia: British History

- This 968-1016 king of England spent the majority of his reign in a defensive war against Danish invaders. His nickname means "evil counsel", "bad plan", "folly", or "ill-advised" in Old English. The invective is actually focused on those around him, who were expected to provide the young king with good advice.



#15

Liberal Arts Trivia: Modern Cinema and English Language

- Name the two-time Emmy award-winning actor who appeared in movies with titles "synonymous" to these:
 - Fiber Fabrication
 - Immoral Metropolis
 - Fail Difficult
 - Invincible
 - The Right-To-Remain-Silent Component
 - Catastrophe
 - The Distance-From-Kevin-Bacon Awareness

#16

From Paul Graham's "Undergraduation":

My friend Robert learned a lot by writing network software when he was an undergrad. One of his projects was to connect Harvard to the Arpanet; it had been one of the original nodes, but by 1984 the connection had died. Not only was this work not for a class, but because he spent all his time on it and neglected his studies, he was kicked out of school for a year. ... When Robert got kicked out of grad school for writing the Internet worm of 1988, I envied him enormously for finding a way out without the stigma of failure. ... It all evened out in the end, and now he's a professor at MIT. But you'll probably be happier if you don't go to that extreme; it caused him a lot of worry at the time.

3 years of probation, 400 hours of community service, \$10,000+ fine

#17

Worm Detection Problem

Input: A program P and input I

Output: **true** if evaluating $(P I)$ would cause a remote computer to be "infected".

Virus Detection Problem

Input: A program specification P

Output: **true** if evaluating (P) would cause a file on the host computer to be "infected".

#18

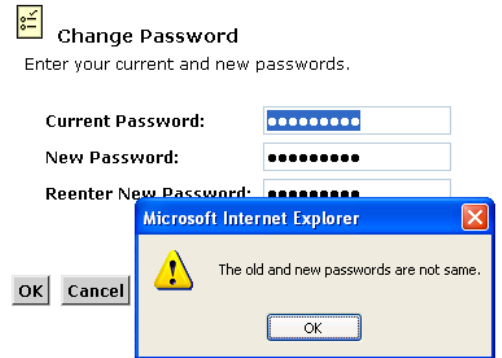
Morris Internet Worm (1988)

- P = fingerd
 - Program used to query user status
 - Worm also attacked other programs
- I = “nop⁴⁰⁰ pushl \$68732f pushl \$6e69622f movl sp,r10 pushl \$0 pushl \$0 pushl r10 pushl \$3 movl sp,ap chmk \$3b”
 - (is-worm? ‘(P I)) should evaluate to #t
- Worm infected several thousand computers (~10% of Internet in 1988)

#19

Computer Security Paradoxes

- Is is-virus? computable?



#20

Uncomputability Proof

Suppose we could define is-virus? Then:

```
(define (halts? P)
  (is-virus?
   '(lambda ()
      (begin ((remove-infects P))
              (infect-files))))))
```

#21

Uncomputability Proof

```
(define (halts? P)
  (is-virus?
   '(lambda ()
      (begin ((remove-infects P))
              (infect-files))))))
```

#t: Since it is a virus, we know (infect-files) was evaluated, and P must halt.

Can we make **remove-infects**?

#f: The (infect-files) would not evaluate, so P must not halt.

Yes, just remove all file writes.

#22

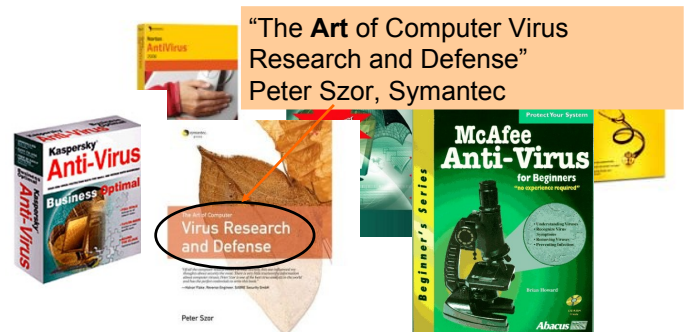
“Solving” Undecidable Problems

- No perfect solution exists:
 - **Undecidable** means there is no procedure that:
 - 1. Always gives the correct answer
 - and also 2. Always terminates
- Must give up one of these to “solve” undecidable problems
 - Giving up #2 is not acceptable in most cases
 - Must give up #1
- Or change the problem: e.g., detect file infections during an execution

#23

Conclusion?

- Anti-Virus programs cannot exist!



#24

Actual is-virus? Programs

- Give the wrong answer sometimes
 - “False positive”: say P is a virus when it isn’t
 - “False negative”: say P is safe when it is
- Database of known viruses: if P matches one of these, it is a virus
- Clever virus authors can make viruses that change each time they propagate
 - Emulate program for a **limited number** of steps; if it doesn’t do anything bad, assume it is safe

#25

Proof Recap

- If we had is-virus? we could define halts?
- We know halts? is undecidable
- Hence, we can’t have is-virus?
- Thus, we know is-virus? is undecidable

#26

History of Object-Oriented Programming

Pre-History: MIT’s Project Whirlwind (1947-1960s)



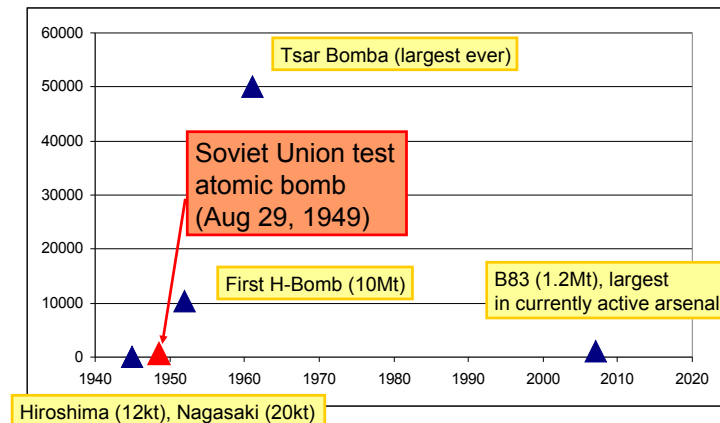
Jay Forrester

#27

#28

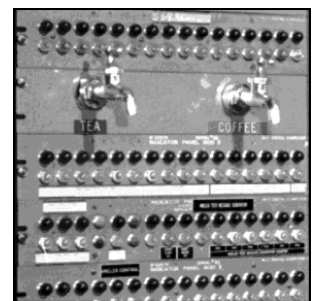
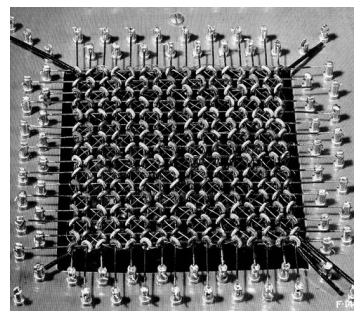
Why Whirlwind?

From an earlier class ...



#29

Whirlwind Innovations



Magnetic Core Memory
(first version used vacuum tubes)

... the first computer that operated in **real time**, used **video displays for output**, and the first that was not simply an electronic replacement of older mechanical systems. Its development led directly to the United States Air Force’s Semi Automatic Ground Environment (SAGE) system, and indirectly to almost all business computers and minicomputers in the 1960s.

#30

Sketchpad

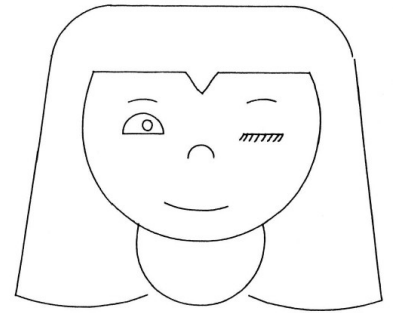
- Ivan Sutherland, 1963 (PhD thesis supervised by Claude Shannon)
- Interactive drawing program
- Light pen
- Turing Award 1988



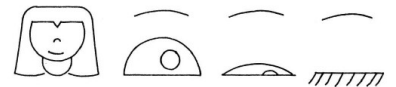
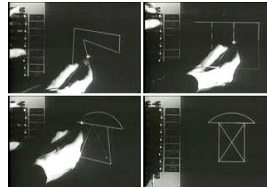
<http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-574.pdf>

#31

Components in Sketchpad



Actual Sketchpad:



#32

Objects in Sketchpad

In the process of making the Sketchpad system operate, a few very general functions were developed which make no reference at all to the specific types of entities on which they operate. These general functions give the Sketchpad system the ability to operate on a wide range of problems. The motivation for making the functions as general as possible came from the desire to get as much result as possible from the programming effort involved. For example, the general function for expanding instances makes it possible for Sketchpad to handle any fixed geometry subpicture. The rewards that come from implementing general functions are so great that the author has become reluctant to write any programs for specific jobs.

Each of the general functions implemented in the Sketchpad system abstracts, in some sense, some common property of pictures independent of the specific subject matter of the pictures themselves.

Ivan Sutherland, *Sketchpad: a Man-Machine Graphical Communication System*, 1963 (major influence on Alan Kay developing OOP in 1970s)

#33

Liberal Arts Trivia: Geography

- Name the most populous metropolitan area on Earth. It contains 23 special wards, each one of which is governed as a city, totaling 35 million people. It houses a National Diet Building and an orange-colored "copy" of the Eiffel Tower. It sustained damage in a 1923 earthquake and later firebombing.



Liberal Arts Trivia: Music

- This traditionally-female vocal range lies below mezzo-soprano, and is usually viewed as the deepest female singing voice. The typical range lies between the F below middle C (F3) to two Fs above middle C (F5). Ruth in Gilbert and Sullivan's *Pirates of Penzance* and Lel in Rimsky-Korsakov's *The Snow Maiden* are example operatic roles.

#35

Simula

- Considered the first "object-oriented" programming language
- Language designed for simulation by Kristen Nygaard and Ole-Johan Dahl (Norway, 1962)
- Had special syntax for defining classes that package state and procedures together

#36

Counter in Simula

```
class counter;
integer count;
begin
  procedure reset(); count := 0; end;
  procedure next();
    count := count + 1; end;
integer procedure current();
  current := count; end;
end
```

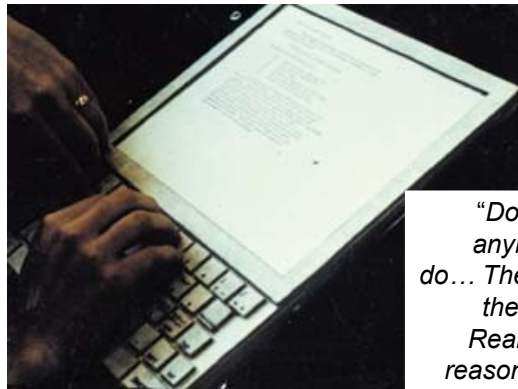
#37

XEROX Palo Alto Research Center (PARC)

1970s:

- Bitmapped display
- Graphical User Interface
 - Steve Jobs paid \$1M to visit PARC (bought their stock), and returned to make Apple Lisa/Mac
- Ethernet
- First personal computer (Alto)
- PostScript Laser Printers
- **Object-Oriented Programming**

#38



Dynabook, 1972
(Just a model)

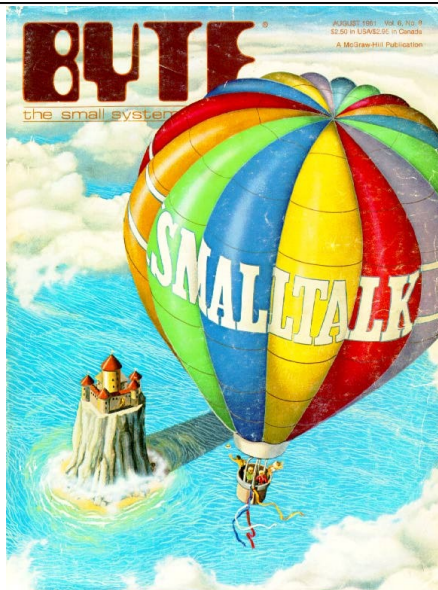
"Don't worry about what anybody else is going to do... The best way to predict the future is to invent it. Really smart people with reasonable funding can do just about anything that doesn't violate too many of Newton's Laws!"
— Alan Kay, 1971

#39

Dynabook 1972

- Tablet computer
- Intended as tool for learning
- Kay wanted children to program it also
- Hallway argument, Kay claims you could define "the most powerful language in the world in a page of code"
- Proof: Smalltalk
 - Scheme is as powerful, but takes two pages
 - Before the end of CS 150, we will see an equally powerful language that fits in ¼ page

#40



BYTE Magazine,
August 1981

#41

Smalltalk

- Everything is an *object*
- Objects communicate by sending and receiving *messages*
- Objects have their own state (which may contain other objects)
- How do you do 3 + 4?
send the object **3** the message **" + 4 "**

#42

Counter in Smalltalk

class name counter

instance variable names count

new count <- 0

next count <- count + 1

current ^ count

#43

Counter in Python

class counter:

```
def __init__(self): self._count = 0
```

```
def reset(self): self._count = 0
```

```
def next(self): self._count = self._count + 1
```

```
def current(self): return self._count
```

counter() creates a new counter using the `__init__` method
`_count` is the instance variable (`_` is just a naming convention)

#44

Who was the first object-oriented programmer?

By the word operation, we mean any process which alters the mutual relation of two or more things, be this relation of what kind it may. This is the **most general definition**, and would include all subjects in the universe. Again, it might act upon other things besides number, were **objects found whose mutual fundamental relations** could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine... Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

Ada, Countess of Lovelace, around 1843

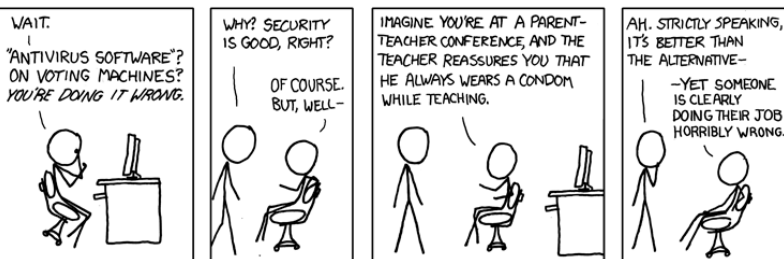
#45

#46

Homework

- Read Chapter 12 for Wednesday
- PS 7 due Monday April 6

PREMIER ELECTION SOLUTIONS (FORMERLY DIEBOLD)
HAS BLAMED OHIO VOTING MACHINE ERRORS ON PROBLEMS
WITH THE MACHINES' MCAFEE ANTIVIRUS SOFTWARE.



#47