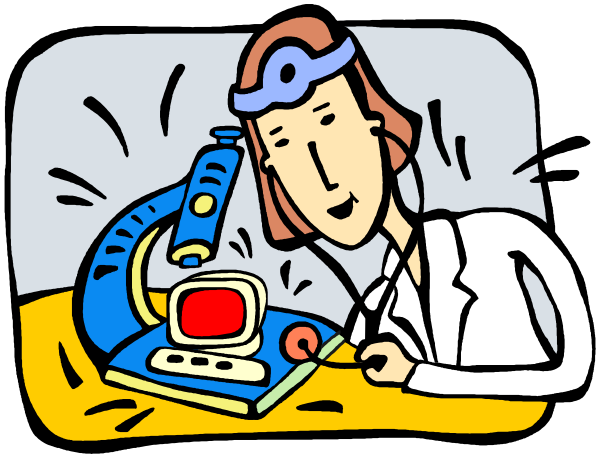
Welcome to CS 42!

Principles and Practice of Computer Science



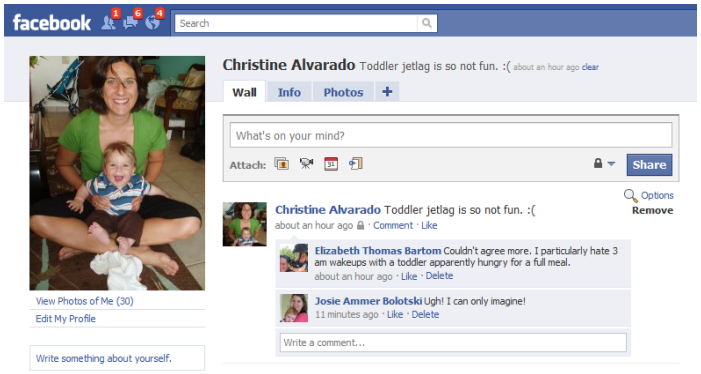
"Computer Science is no more about computers than
astronomy is about telescopes,
biology is about microscopes, or
chemistry is about beakers and test tubes."
— **Fellowes and Parberry**

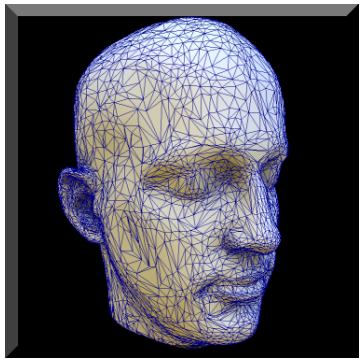
Interactions
&&
properties

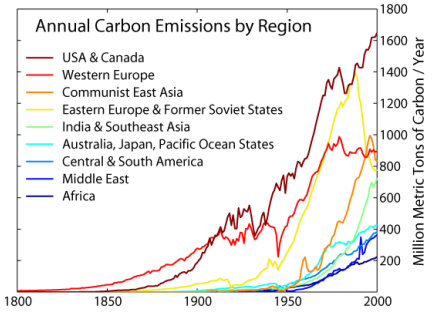
- matter and energy — physics
- molecules — chemistry
- cells and organisms — biology
- rule-based systems — mathematics
- information — computer science

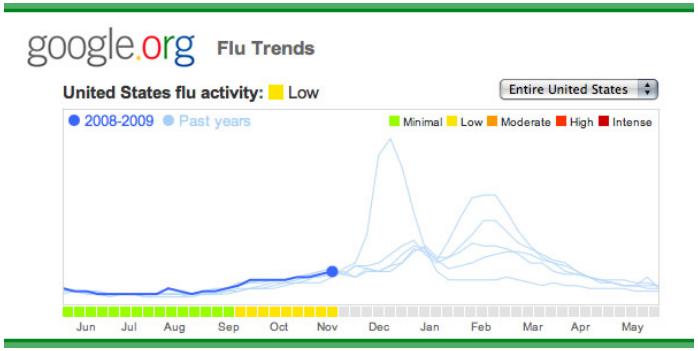
42 = 5 + 60 – Java

What is Information?









Computer Science Fiction?

http://www.ted.com/talks/tan_le_a_headset_that_reads_your_brainwaves.html

Why CS ?

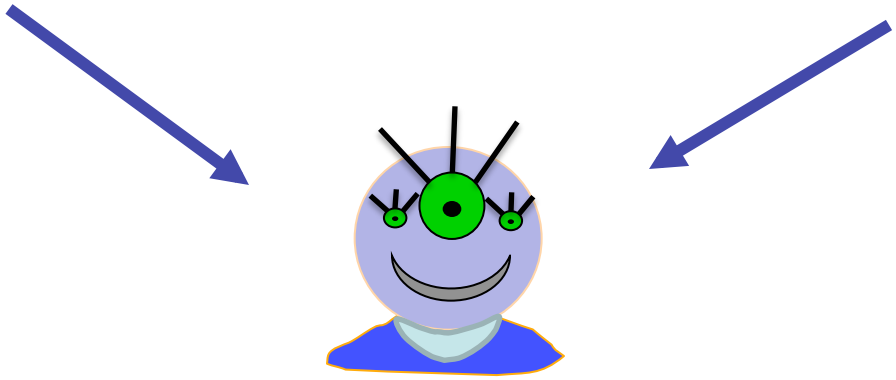
- Information is life's fundamental *building block*.

Genetic Code: DNA

`GTAGCACAITTAGC…`

Our senses and experiences

More coffee required…



us

- CS is a set of fundamental techniques for *understanding and leveraging* this information…

"constructing with"

A Few Aspects of CS

Mathematics

Proposition 4.5
If $\Gamma \vdash \mathcal{E}[(\lambda x{:}A'.\, M)\, M'] : A$ *then* $\Gamma \vdash \mathcal{E}[(\lambda x{:}A'.\, M)\, M'] \equiv \mathcal{E}[[M'/x]M] : A$

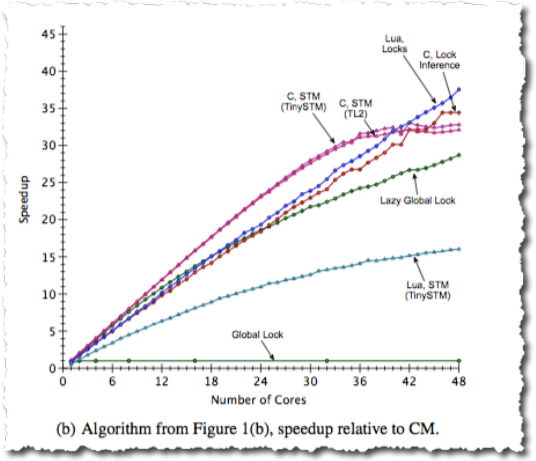
PROOF. By simultaneous induction on the proof of the assumption, and cases on the last rule used.

—Case: Rule 19

$$\frac{\Gamma \vdash (\lambda x{:}A'.\, M) : \Pi x{:}A_1'.\, A_1'' \qquad \Gamma \vdash M' : A_1'}{\Gamma \vdash (\lambda x{:}A'.\, M)\, M' : [M'/x]A''}$$

where $A = [M'/x]A_1''$ and $\mathcal{E} = \diamond$. By Theorem 3.20 and inversion we have $\Gamma \vdash A'$, $\Gamma, x : A' \vdash M \Uparrow B''$, $\Gamma \vdash \Pi x{:}A'.\, B'' \leq \Pi x{:}A_1'.\, A_1''$, $\Gamma \vdash M' \Uparrow B'$, and $\Gamma \vdash B' \leq A_1'$. By inversion of Rule 10 we have $\Gamma \vdash A_1' \leq A'$ and $\Gamma, x : A_1' \vdash B'' \leq A_1''$. By Theorem 3.18 we have $\Gamma, x : A' \vdash M : B''$ and by subsumption $\Gamma \vdash M' : A'$, so by Rule 48 we have $\Gamma \vdash (\lambda x{:}A'.\, M)\, M' \equiv [M'/x]M : [M'/x]B''$. By Proposition 3.7 $\Gamma \vdash [M'/x]B'' \leq [M'/x]A_1''$, so by subsumption we have $\Gamma \vdash (\lambda x{:}A'.\, M)\, M' \equiv [M'/x]M : [M'/x]A_1''$

Experiments



(b) Algorithm from Figure 1(b), speedup relative to CM.

Applications (and HCI)

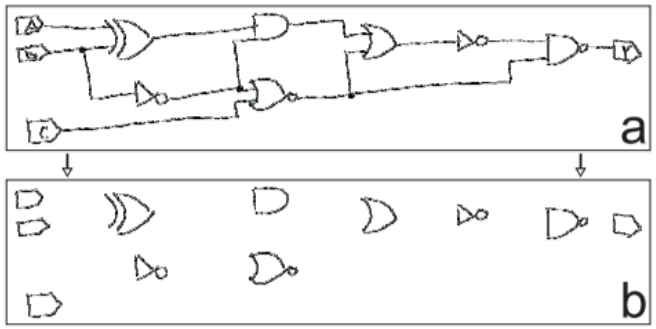
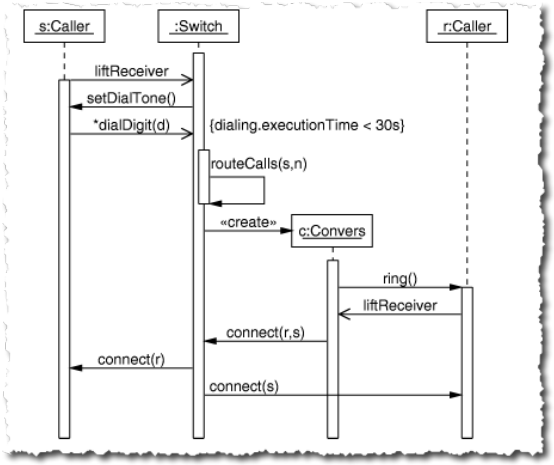
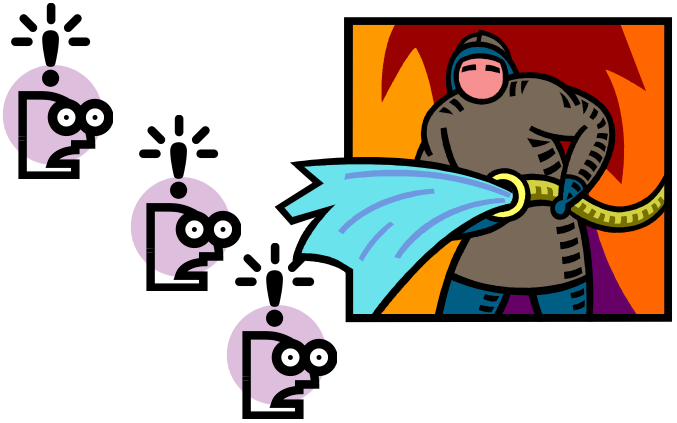


Figure 1: (a) A digital logic sketch. (b) The strokes in the sketch that are classified as gates.

Design and Engineering

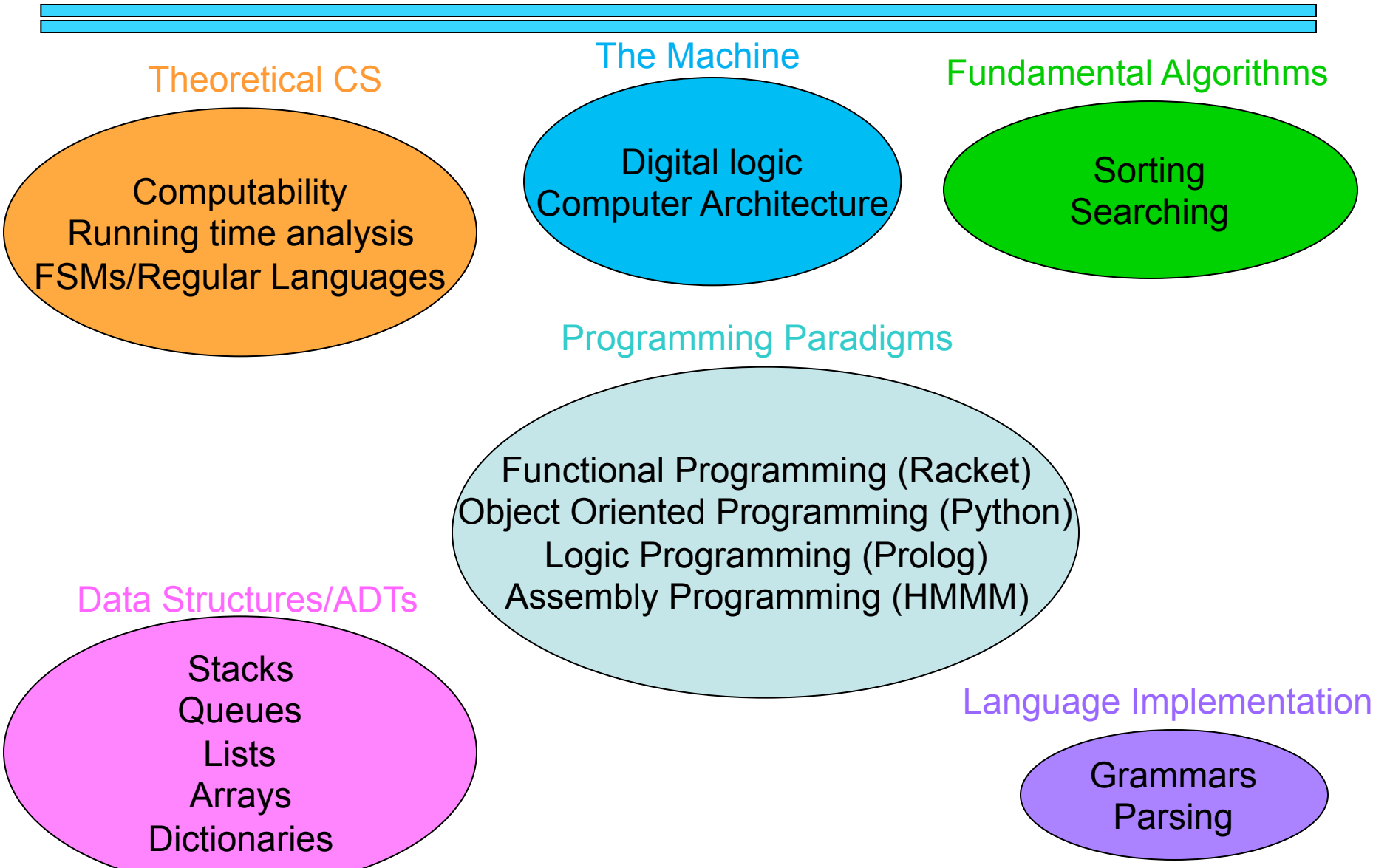


42 == 5 + 60 - Java



- You are in CS 42 because you have significant CS experience
  - I will assume you know Java, or could pick it up without trouble.
- CS 42 combines the best of CS 5 and CS 60, but it moves fast
  - If you feel you would be better placed in CS 5, talk to me ASAP
- CS 42 prepares you for:
  - Any course with CS 60 as a prereq
  - Future computational work
  - A happy and fulfilling life :-)

A Modular Course, for a Modular Classroom



Syllabus, briefly

Lectures **T/R:** 1:15-2:30 pm

Key skills, topics, and their motivation

Insight into the HW problems (what, ***why***, how)

Required! Let me know if you won't make it

Tutoring and Office Hours http://www.cs.hmc.edu/cs42/tutorhours.html (LAC Lab)

Can't make these hours? Just contact me by email **`cs42help@cs.hmc.edu`**

HW **Monday nights:** due by 11:59 pm

Syllabus, more briefly

Lectures **T/R:** 1:15-2:30 pm

Key skills, topics, and their motivation

Insight into the HW problems (what, ***why***, how)

Required! Let me know if you won't make it

Tutoring and Office Hours

OFFICE/TUTOR HOURS START TONIGHT

`cs42help@cs.hmc.edu`

HW

HOMEWORK 1 DUE MONDAY!

Grading

Grades

Based on points percentage

~ 60% Assignments
~ 30% Exams
~ 10% Participation/"quizzes"

```
if perc >= .95: grade = 'HP'
elif perc >= .70: grade = 'P'
```

Exams

(page of notes OK)

Midterm 1	October 11-13
Midterm 2	November 17-22
Final	TBA

Note!

To pass CS 42, you ***must*** have a passing grade on each of the three individual components of the course (exams, HW, and participation/"quizzes"). Failing one component results in failing CS 42, even with a passing total score. Missing more than 5 "quizzes" will result in failing CS 42.

Homework

Assignments

~ 4 problems/week ~ 100 points extra credit available

Due Monday evening - by 11:59 pm.
http://www.cs.hmc.edu/submit

"Late Days"
You have 3 ***CS 42 Euros*** to use…
…for 3 different homeworks

Collaboration

Some problems are specified "individual-only."

Others offer the option of working in a pair.

- You don't have to
- You must share the work equally - typing and coaching
- You should make ONLY ONE submission for both of you
- Be sure to indicate who your partner was at the submission site!
- You may want to email your partner the code, too.

Honor Code →

Honor Code

- You are *encouraged* to **discuss** problems with other students – or tutors - or any instructors.
- You may **not** share written, electronic or verbal solutions with other students (present or past):
  - No internet code or other copying of code except those provided by the course.
  - No transcribing of code from paper, whiteboards, blackboards, or other media.

You will have the option of working in pairs for MANY of each week's problems: the same guidelines apply for each pair.

Read over the syllabus and honesty policy – sign & submit HP today!

Scenario 0

It's 10:30pm on Monday night, and you remember you haven't even started the week's homework.

The assignment asks you to construct an animated simulation of a Turing Machine. To remind yourself what that is, you google "Turing Machine." To your surprise, the 42nd link is very relevant; the professor used the same problem in CS 65 three years ago, and the sample solutions are still online!

With time running short, you copy the professor's solution, making just a few changes in the comments. "It'll be OK," you think. "I understand this code just fine, and could have written it if I had more time."

Scenario 1

The week's homework looks long, but all the problems are marked as permitting Pairs.

"We'll split it up," your roommate says. "I'll do the first two problems, and you do the last two. Then we'll meet, paste our solutions together, read them over together to make sure we agree, and submit them together."

"It's OK," you think. "It says we can work in pairs. Besides, this is how teams work in the real world."

Scenario 2a

The homework asks you to write an AI program that plays Checkers against a human.

A quick Google search leads you to code for an open-source Checkers-playing program. You copy just the code to display the board, and add comments containing the URL of the original code and the names of the original authors. Now you can start the AI part.

“It’s OK,” you think. “In the real world, people reuse code all the time; it’s considered bad form to reinvent the wheel. Besides, everyone knows it’s not plagiarism if I cite my sources!”

Scenario 2b

The week's homework asks you to write an AI program that plays Checkers against a human.

By sheer coincidence, you spent the past summer writing a Chess-playing program. You copy the code you wrote to display the board and make a few tweaks to remove the chess-specific bits. Now you can start the AI part.

"It's OK," you think. "Playing good checkers is hard; displaying the board isn't an interesting or important part of the homework. Besides, I wrote this code myself, so it's obvious I know how to do it.

Scenario 3

The professor hands out a 2-hour take-home exam. You take it 8:13–10:13pm, in your dorm room. The instructions say “Do not use a computer during the test except to view materials posted on the course web page.”

At 8:12pm, you start iTunes playing a relaxing two-hour compilation of your favorite mp3s. At 8:47pm, you get an IM from your cousin, asking whether he should buy an iPad or a laptop; you dash off a quick reply. At 9:31pm, you look up “pseudocode” in the on-line dictionary to make sure you’re spelling it right. At 9:35pm, you email the professor to ask whether “b” in Question 9, line 4 was supposed to be “β.”

At 10:13pm precisely, you stop work on the test. You turn it in the next day, confident that you haven’t cheated.

Scenario 4

The last homework problem this week, a control algorithm for an automated Robot Butler, is labeled “Individual.” It looks hard.

You and your friend start discussing the problem, and quickly realize how the lecture on Depth-First Search relates to robot-butler control. You grab some scrap paper, start drawing examples and sketching out key lines of code, and in the end, agree on how to solve the problem by breaking it into three key steps.

You both walk to the lab, separately write Butlering code at different computers, and each submit. “It’s OK,” you think. “In the end, the code I submitted was written by me, individually.”

Lecture Slides

- I will post them the night before the lecture. I will not print them out (but you can if you like)
- I will post slides with notes after class

Submissions Site

http://www.cs.hmc.edu/submit

IMPORTANT NOTE: To log off of the submissions site, you must quit all windows in your browser

The most important things you'll learn today:

`http://www.cs.hmc.edu/cs42`

`cs42help@cs.hmc.edu`

`http://www.cs.hmc.edu/submit`

Taxonomy of Programming Models

What mood am I in today?

- Programming
  - Declarative Programming
    - Logic Programming — **Prolog**
    - Functional Programming — **Racket**
      - algorithms via functions
  - Imperative Programming
    - Object-Oriented Programming — **Python**
    - Low-level Programming — **Assembly(HMMM)**, **Picobot**

more abstraction away from the machine ⟷ closer to the machine

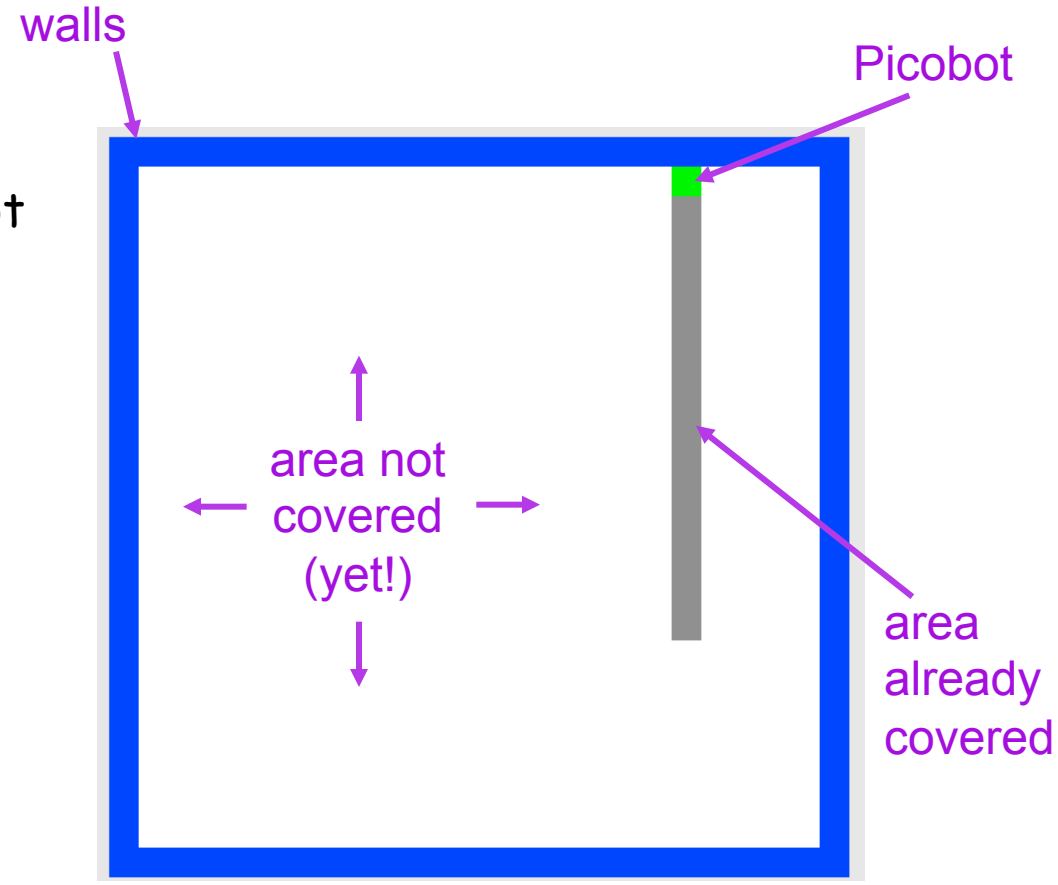
HW problems 1 and 2: Programming?

Picobot

http://www.cs.hmc.edu/picobot

as envisioned by an HMC clinic

inspiration?



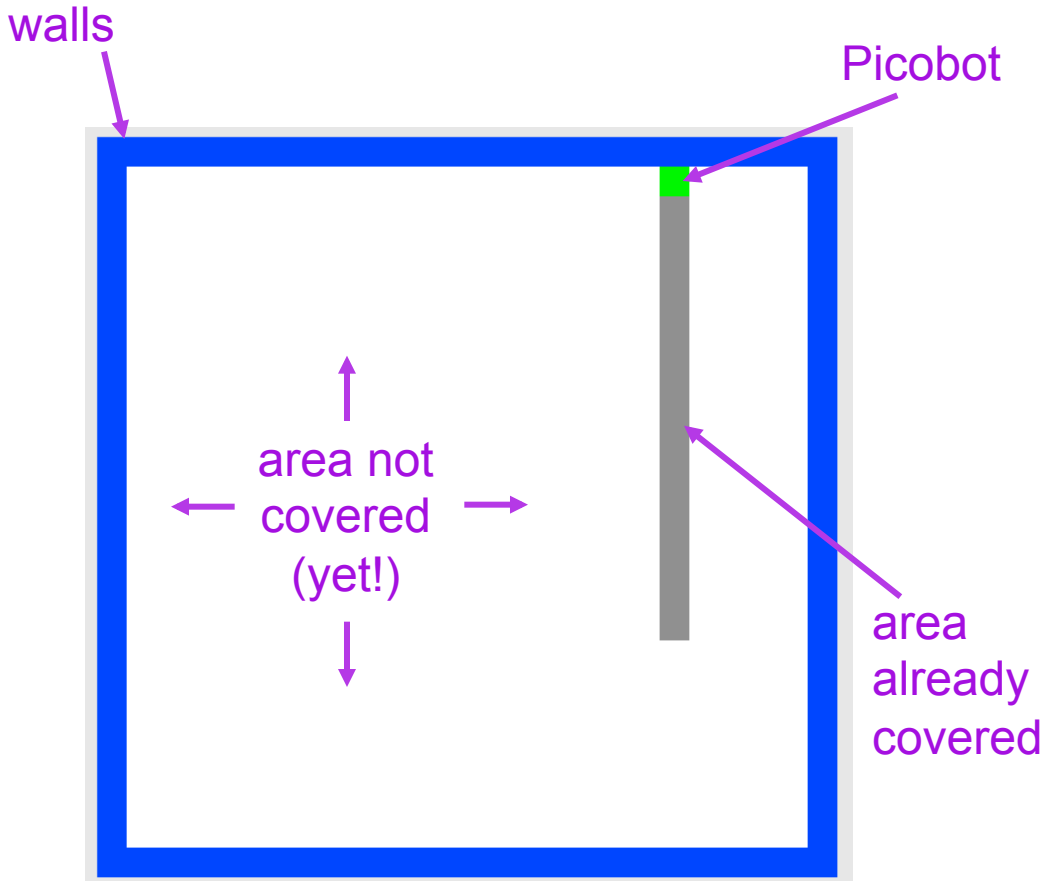
Goal: whole-environment coverage with only ***local sensing***…

HW problems 1 and 2: Programming?

Picobot

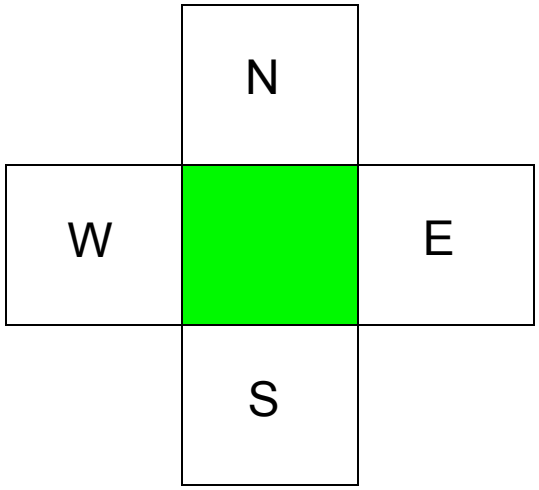
iRobot's Roomba vacuum

inspiration!

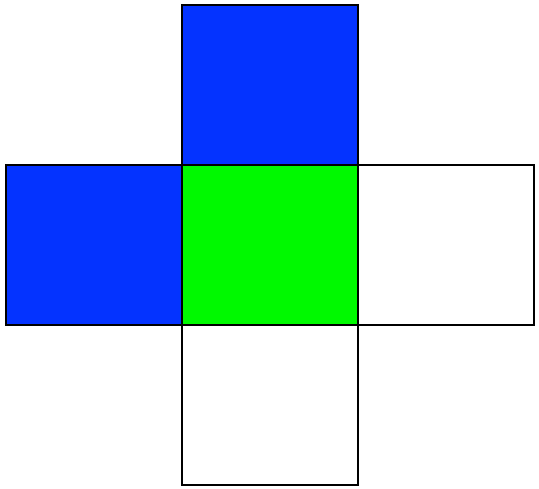


Goal: whole-environment coverage with only ***local sensing***…

Surroundings



Picobot can only sense things directly to the N, E, W, and S



For example, here its surroundings are

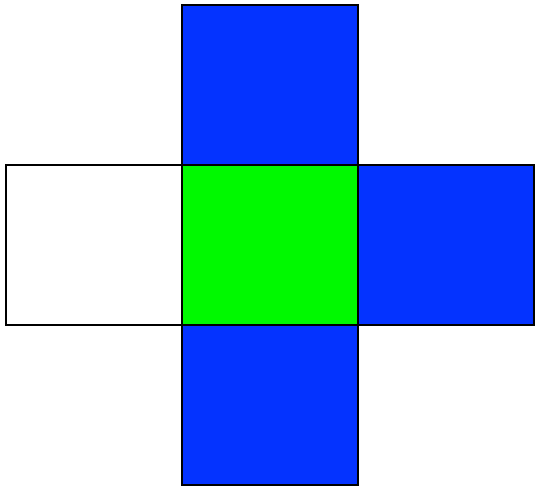
Nx**W**x

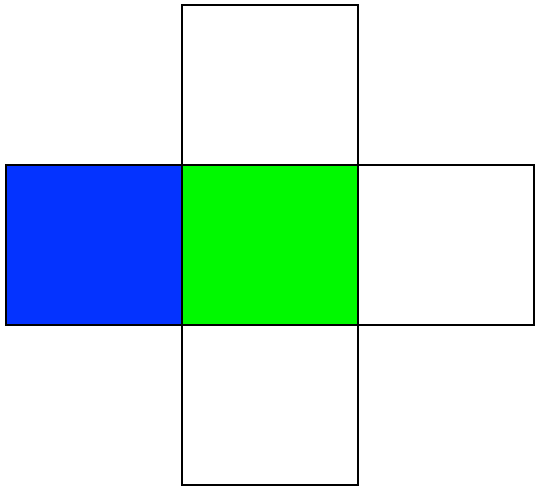
N E W S

Surroundings are always in NEWS order.

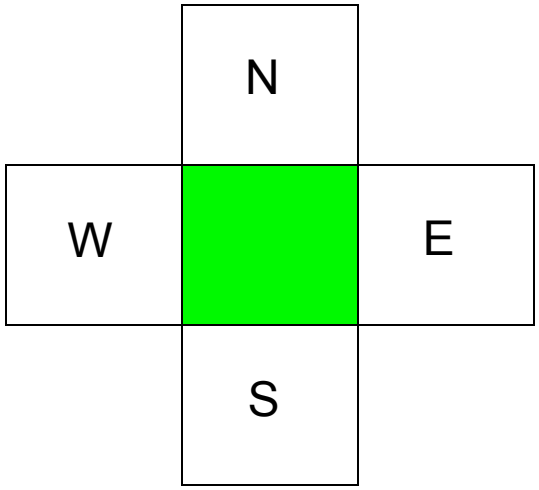
Surroundings

Describe these surroundings



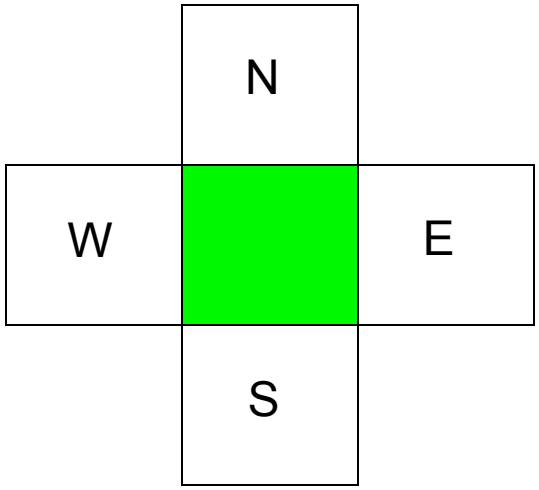


Surroundings



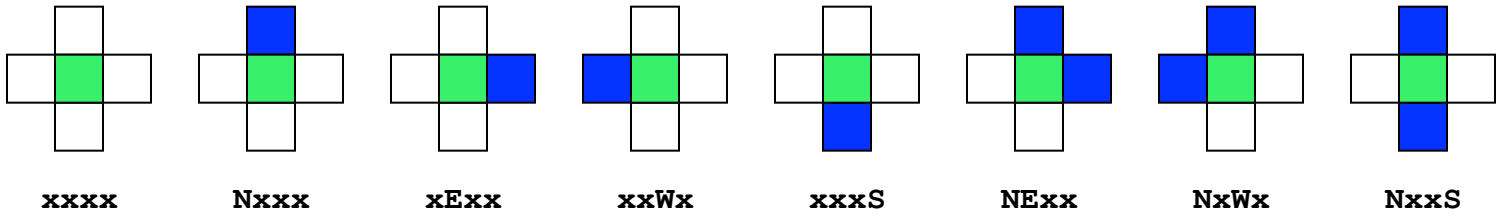
How many distinct surroundings are there?

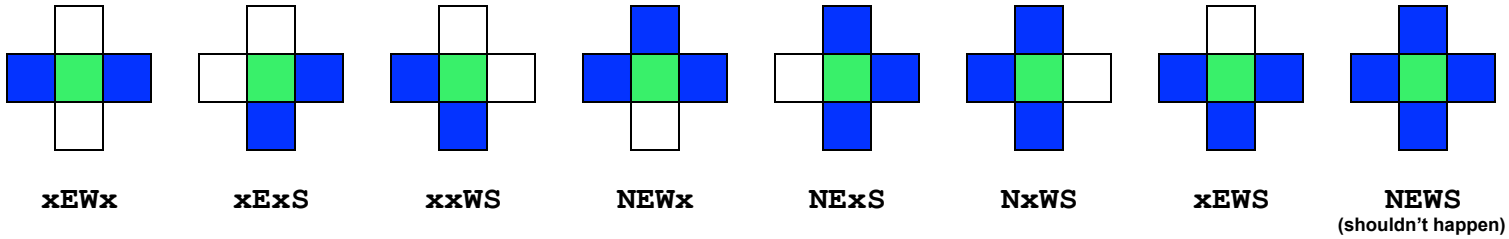
Surroundings



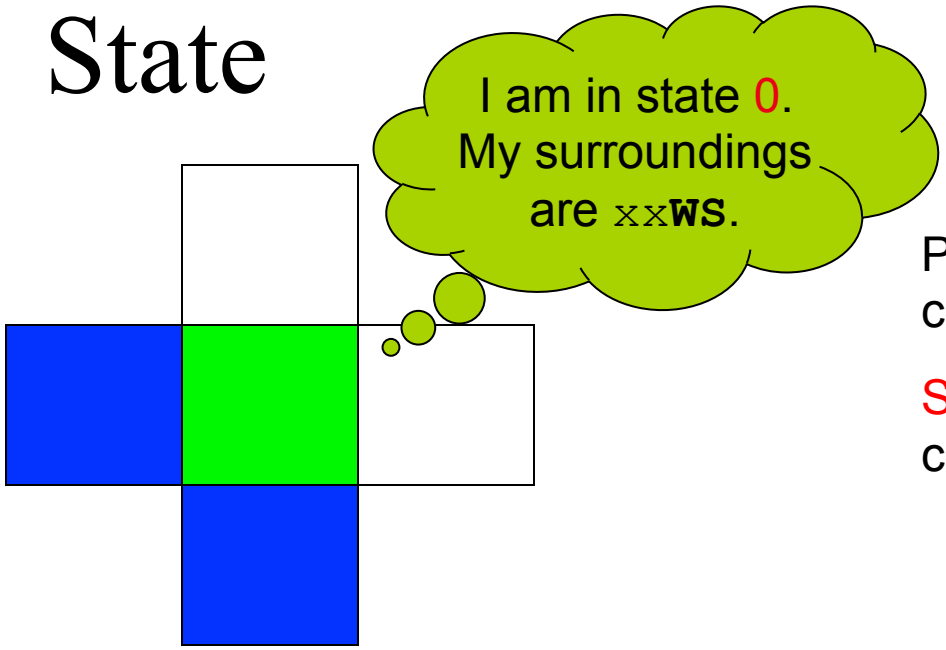
How many distinct surroundings are there?

2^4 == 16 possible …





State



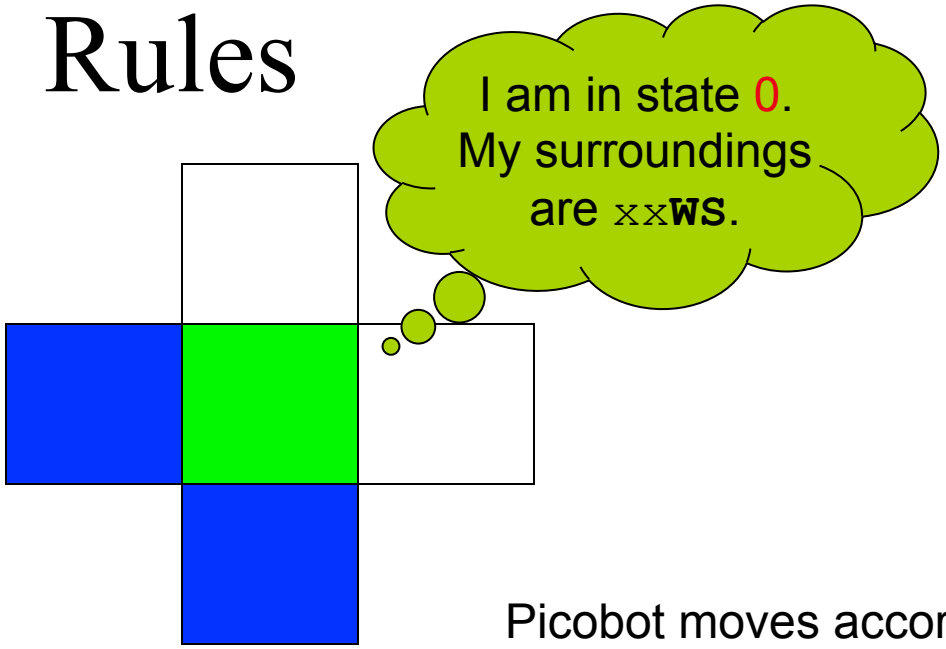
Picobot's memory is a single number, called its state.

State is the *internal context* of computation.

Picobot always starts in state 0.

State and **`surroundings`** represent everything the robot knows about the world

Rules



Aha!
I should move N.
I should enter state 0.

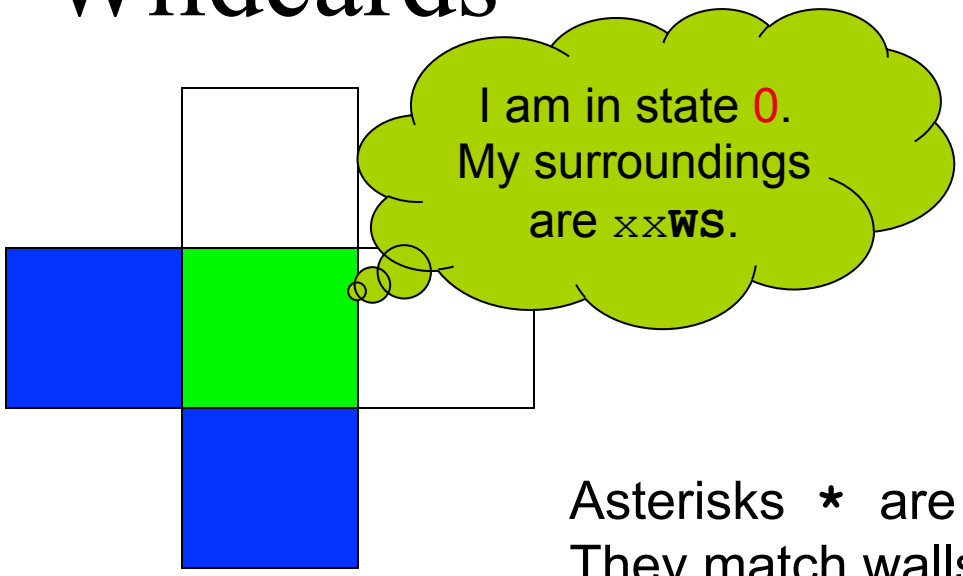
Picobot moves according to a set of rules:

state	surroundings		direction	new state
0	xx**WS**	→	N	0

If I'm in state 0
seeing xx**WS**,

Then I move **N***orth, and change*
to state 0.

Wildcards



Aha! This matches x***

Asterisks * are wild cards.
They match walls ***or*** empty space:

state	surroundings		direction	new state
0	x***	→	N	0

here, EWS may be wall or empty space

What will this set of rules do to Picobot?

state	surroundings		direction	new state
0	`x***`	->	N	0
0	`N***`	->	X	1
1	`***x`	->	S	1
1	`***S`	->	X	0

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

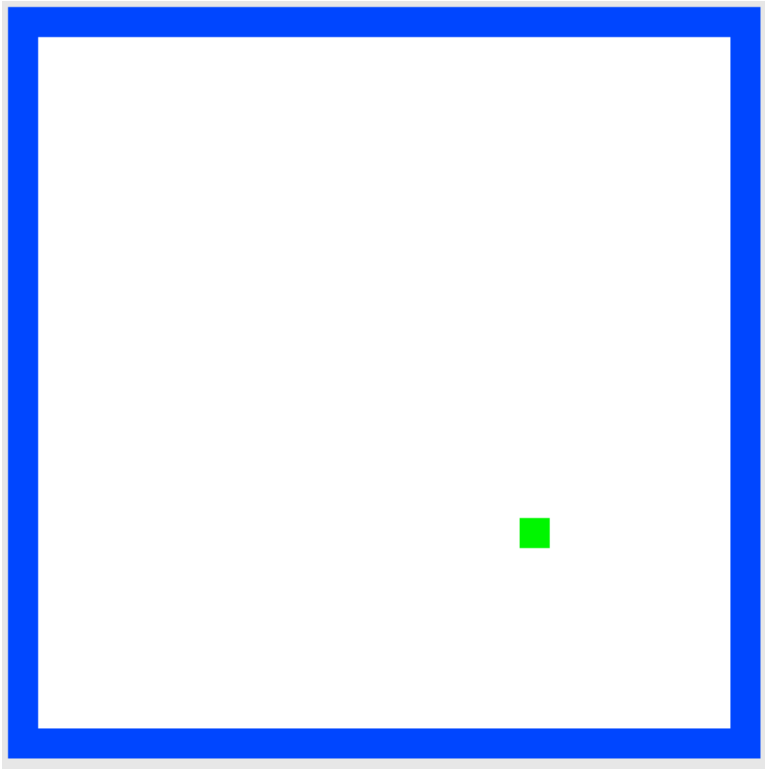
Only one rule is allowed per state and surroundings.

To do

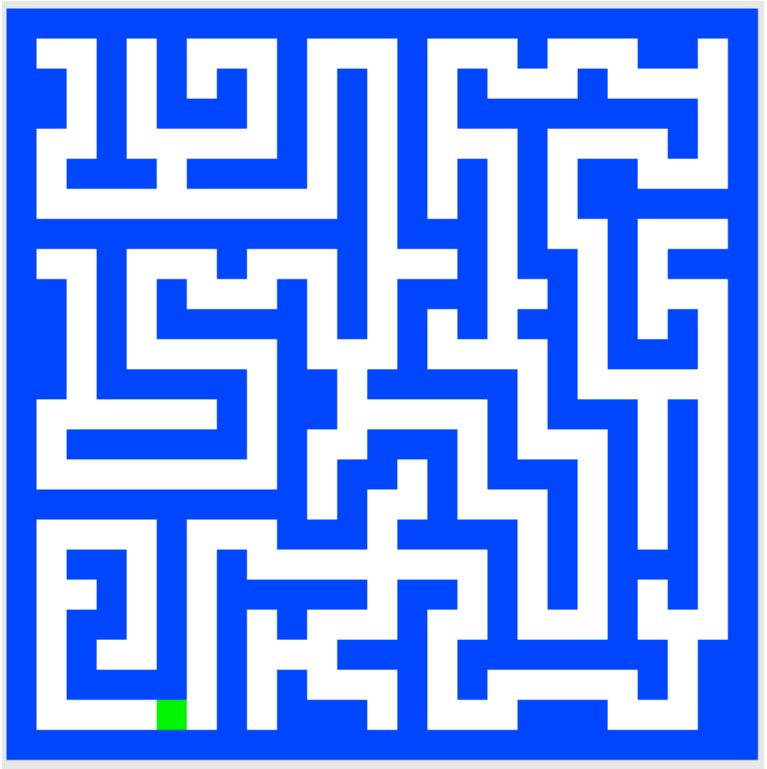
Write rules that will always cover these two rooms.

(*separate* sets of rules are encouraged…)

hw1, Problem #1



hw1, Problem #2



but your rules should work *regardless* of Picobot's starting location