INTRODUCTION TO COMPUTER SCIENCE · WEEK 1

What is Intelligence?

A first-semester class, in fourteen weeks, that answers its title with a definition — and earns it.

proof · notation · syntax · truth · meaning · semantics

WHAT THIS CLASS IS FOR

A deep understanding of the basic principles — deep enough to position generative AI, and whatever comes next, properly.

Not how to use a device. Not how to prompt a chat bot. Not even, mainly, how to code.

The principles were true before this technology arrived and will still be true after whatever replaces it. That is why they are worth a semester, and why nothing in this class mentions a product, a vendor, or a year.

By the end you will be able to say, about any machine that talks, exactly what it is and exactly what it cannot be — and to check the answer yourself.

THE QUESTION

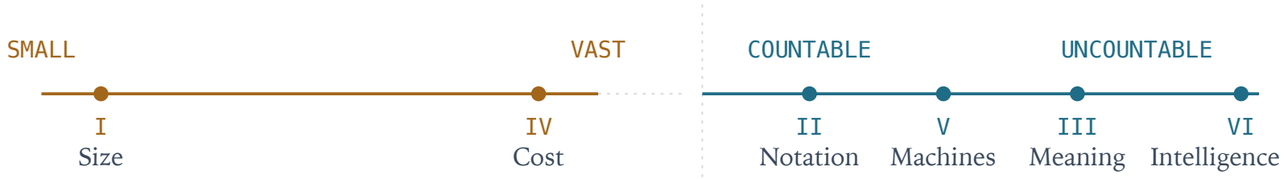
Is it intelligent? — badly posed, as the talk said. We replace it with questions you can answer: what language, what semantics, what metric, what budget, who checks.

THE SHORT ANSWER

Intelligence is developing new formal languages — or new properties in existing ones — which requires discovering and understanding promising *unproven* truth. Fourteen weeks to earn it.

THE AXIS

Everything in this class sits somewhere on one line.



the six stations of this book, on its one axis

the six stations of the semester

Finite. Bits, and how fast a few of them outnumber the atoms in the universe. Week 2 and 3.

Countable. Everything you can write down: programs, proofs, machines — and the machine that reads them. Weeks 4 to 6.

Uncountable. What you mean by it. Behaviours, truths. Cantor, Gödel, Turing, Rice — then the compiler and the operating system built anyway. Weeks 7 to 10. Then cost, machines, and the answer.

FOURTEEN WEEKS

The route.

WK	STATION	LECTURE	LIVE IN THE TERMINAL
1	—	What is Intelligence? The talk, then this	./selfie
2	I	Size: bits, orders of magnitude, state spaces	-c selfie.c
3	I	Everything is bits: numbers, text, kilo vs kibi	examples/
4	II	Notation: formal languages, EBNF, C* and RISC-U	-s
5	II	Countability: a binary is a number; the fixed point	make self-self-check
6	II	The machine: RISC-U, mipster as the universal machine	make emu, -d
7	III	Uncountability: the diagonal, twice; truths without proof	—
8	III	Self-reference I: the compiler, the fixed point, Gödel	make self
9	III	Self-reference II: halting, universality, Rice	-d on a loop
10	III	Systems: OS by emulation ≡ OS by virtualization	make emu-emu, os-emu
11	IV	Cost: P vs NP, SAT, a solver in 400 lines, Landauer	make sat
12	IV	Formal methods: machine code in, formula out	make rotor, bitme
13	V	Machines: what an LLM is; hallucination; the gate	a prompt, and a compiler
14	VI	What is intelligence? The definition; the exam	—

THE SPECIMEN

One file, small enough to read to the end: selfie.

A compiler for a tiny subset of C, an emulator for a tiny subset of RISC-V, and a hypervisor — twelve thousand lines, one file, written in the language it compiles.

Three commands: the compiler compiles itself, the emulator executes itself, the hypervisor hosts itself. Understand the first and you understand compilers; the second, machines; the third, operating systems.

And a workshop of tools around it that turn a program into a logical formula and hand it to a solver — where this class meets the frontier, in week 12.

```
$ ./selfie -c selfie.c -o selfie1.m -m 2 -c selfie.c
./selfie: selfie compiling selfie.c to 64-bit RISC-U
…
./selfie: 64-bit mipster executing 64-bit RISC-U bin
selfie1.m: selfie compiling selfie.c to 64-bit RISC-
…
$ diff -s selfie1.m selfie2.m
Files selfie1.m and selfie2.m are identical
```

BEFORE NEXT WEEK

Install selfie, and run it once.

1. Get a terminal. Every laptop has one; if you only have a browser, the repository's README shows the cloud option.
2. Download selfie from `github.com/cksystemsteaching/selfie` and type `make` in its directory. You need a C compiler; the README says which.
3. Type `./selfie`. It answers with its synopsis, one line, in a formal language. Bring that line to week 2.
4. Read the introduction of the book, and the Selfie chapter, and type the two commands in it.

THE BOOK

What is Intelligence? — this class is its first six parts at talk resolution, one chapter per week or so. It is in the repository under `book/`.

EXERCISES

This class has no formal assignments. It has a recommended exercise list per week — reading, the commands of the session to rerun, a few paper exercises — and one thing to try on the autograder, which is how the compiler and systems classes are graded.

HOW TO READ THIS CLASS

Slow yourself down.

Formal languages are not casual. Everything matters, even the tiniest detail, and the trick to learning them is to take steps so small they are almost painful.

Every negative result in this class — there are five — is read twice: what it forbids, and what it opens. The second reading is the one to remember.

> *Whatever intelligence is, humour is the only way.*
>
> THE LAST SLIDE OF THE TALK, AND OF WEEK 14