What is COMPUTATION?

Input
Human Genome (3 billion base pairs)

Program

Output
How might you write such a program?

Suppose I want to login to a system:
The system needs to accept or reject my password.

Usual approach:
Store $s = \text{hash}(\text{realpw})$.
Check $\text{hash}(\text{pw}) = s$.

Consider this code:
/* returns NULL terminated string of arbitrary length */
char * hash(char *r);

/* pw and s are NULL terminated */
void func(char *pw, char *s) {
    while (1) {
        if (strcmp(pw, s) == 0) {
            printf("pw accepted");
        }
        pw = hash(pw);
    }
}

Determine whether func(pw, s) prints "pw accepted"
How might you write such a program?

Some programs are easy, some are hard, some are impossible.

1. Computation without storage. (Finite automata, regular expressions)
2. Computation with storage. (Pushdown automata, context-free grammars)
3. Limits of computation. (Turing machines, undecidability)
4. Complexity of computation. (Complexity classes, reductions)

Assessment
40%: Homework
40%: Quizzes
15%: Final Exam
5%: Class Participation
Homework
8 assignments
100 points each
N hours late: $2^{N/6}$ point penalty
LaTeX typeset: 10% bonus

Quizzes
4 quizzes
Self-administered
Assigned in class, due in 24 hours
No late turn-ins

Final Exam
Take home, self-administered
Comprehensive
Assigned on 12/12, due on 12/15
No late turn-ins

Class Participation
Keep the discussion going.
Office hours: Monday, Tuesday, Thursday 2:30-4.
Skipping assessments: give me 2 weeks notice.
Homework schedule: mostly weekly.
Lecture slides: on webpage after each lecture.
Recommended reading: given each lecture.

Reading: Chapter 0.

Questions!
We need formal specifications.

Symbols compose an alphabet

\[ \Sigma = \{0, 1\} \]

\[ \Sigma = \{a, b, c, d, e, f\} \]

An alphabet is used to make strings

\[ s_1 = 1001010 \]

\[ s_2 = \text{feed} \]
String operations

\[ s = 1001 \]
\[ ss = 10011001 \]
\[ s^2 = 10011001 \]
\[ |s| = 4 \quad |\varepsilon| = 0 \]
\[ 0001^R = 1000 \]

A set of strings makes a language

\[ L_1 = \{01, 10, 011\} \]
\[ L_2 = \{\text{cad, feed}\} \]

A set of languages makes a language class

\[ L_1 = \{0, 000, \ldots\} \]
\[ L_2 = \{00, 0000, \ldots\} \]
\[ C = \{L_1, L_2\} \]

Language definitions can be colloquial

\[ L_3 = \{ss|s \text{ has more 0s than 1s}\} \]
Simple model of computation: the automaton.

Consider an elevator.
If we end in state 3, the automaton $A$ accepts. Otherwise it rejects.

Language $B$ of all button sequences that end on the 3rd floor.

Automaton $A$ recognizes $L(A) = B$

Accepts 12133
Rejects 31213
Claim:
In any set of n hats, all are the same color.

“Proof” by induction:
Base case: If n = 1, all hats are the same color.
Inductive case: Assume that the claim holds for all n ≤ k. Examine any set of n = k+1 hats. Remove one hat from the set; the remaining set has k hats and thus is of a single color. Replace the hat and remove a different hat from the set; the resulting set has k hats and thus is of a single color. The two sets overlap, thus they must be of the same color. Therefore, all k+1 hats are the same color.  □