Alexander Woo
awoo@uidaho.edu
(208)-885-6741 (office)

8/20/18

The Halting Problem

Thm: There cannot a computer program that takes as input a computer program (as a long string in C) and input to that program and outputs whether this program will halt or not given that input.

Pf: Suppose for contradiction there was such a program, packaged as a C function

```
boolean haltp(char* program,
              char* input)
   :
   :
end
```

Write a new program:

```
void breaker(char* x)
  if haltp(x, x)
      go into infinite loop
  else
      terminate
```

Call

```
breaker( this string )
```

This is a contradiction, since breaker breaks itself. if it terminates, ~~if won~~ haltp returns true and it loops; if it loops, haltp returns false and it terminates.

~~There~~ Conclusion: there is no such program.

A false theorem:

"Theorem": Every positive integer can be described in less than 20 words.

"Pf": Suppose not - then there exists a smallest pos. int. not describable in <20 words. We can describe this integer as "The smallest pos. int. not. describable in less than 20 words" □

What's wrong is that we haven't been precise about what "describe" means.

This throws into question the previous argument

Goal for semester: Give a mathematical formalization of "computer program", "input," "output," "feeding a program as input to another program" - so that we can be sure this argument works.

Outline for semester:

singular: automaton
plural: automata

1 We make mathematical def'ns of several kinds of automata - formal mathematical models of computers w/ varying levels of complexity and capabilities.

2 Automata will take a string as input and decide "true" or "false" for the input.

3 We can think about the set of strings that an automaton says "true" to. This set is a language. We will study what properties a language must have to be accepted by a particular type of automaton.

4 E.g: { strings of a's and b's where the number of a's is divisible by the number of b's }

abaa is in this language (3 is div by 1)

abaab is not (3 is not div by 2)

E.g. { strings of ('s and)'s ~~where~~ which form proper strings of parens - i.e. #('s = #)'s, and at any point, we have seen at least as many ('s as)'s. }

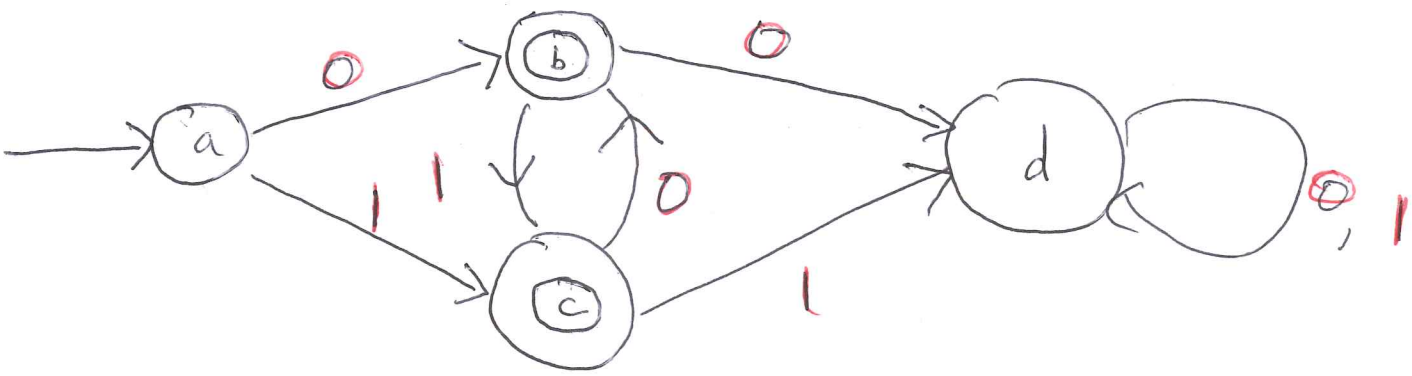
4 Think about a language in terms of rules for generating all possible strings in the language. This collection of rules is a grammar.

Study how type of automaton

restrictions on language

type of grammar

Deterministic Finite Automata (DFA)



0 1 0 1 1 1 0

a b c b c d d d → not an accept state, so false

A DFA has a finite number of states

One is the initial state (a)

When fed a letter, it ~~changes state~~ goes to some state depending on the current state and the letter it is fed.

Feeding a string just means feeding letters one at a time. If, at the end of the string, we are in a final state (or accept state), it says "true" and otherwise "false"

We just worked out - this DFA accepts exactly the strings which strictly alternate 0's and 1's.