Tuples ≡ set $(a,b) = \{a, \{a,b\}\}$

CSE322 Theory of Computation (L2)

Recap of last lecture

Strings - reverse(x·y)
properties = reverse(y)·reverse(x)

Ex: defn. of reverse

- |reverse(x·y)| = |x| + |y|

prefix of length t

suffix of length s

Language L = set of some strings

Today

Finite automaton (DFA)

$\varepsilon \rightarrow bb \in L$

$abba \in L$

$aaaa \equiv (a, (a, (a, (a, \varepsilon))))$

Define language L over the alphabet Σ={a,b} in the following manner:

- e is in L. • If x is in L, then axa and bxb are in L.

a·x·a

L= {ε, aa, bb, axa, aba, bbaa, abab, abba, ...}
aaaa, ...
aaba ∉ L

Prove: For any x in L, |x| is divisible by 4.

∴ proof is wrong. claim is incorrect.

Level 1: Proof by induction on the length of x.

Base case: The fact holds for |x|=0, since the empty string is in L and has zero length which is divisible by 4.

x → y', z, y', z'
arbitrary / general instances of L.

I.H.: For any x in L of length <= n, |x| is divisible by 4.

Started with x that already satisfies IH. y'=baxab z'=bbxbb

I.S.: Take any x in L of length = n. Let y=aaxaa, z=abxba, etc. The length of all of these are |x|+4. Since 4 divides |x|, 4 also divides |x|+4. This shows that |y|, |z|, etc. are divisible by 4. (QED).

IS is supposed to show that ... "For any w in L of larger length, the desired property (4 divides |w|) holds."

Define language L over the alphabet $\{a,b\}$ in the following manner:
e is in L. If x is in L, then axa and bxb are in L.

Prove: For any x in L, $|x|$ is even.

Level 1: Proof by induction on the length of x.

Base case: The fact holds for $|x|=0$, since the empty stirng is in L and has even length.

arbitrary/ general string of longer lengths → IH on another string.

I.H.: For any x in L of length <= n, $|x|$ is even.

I.S.: Take any x in L of length = n+1, where $|x| = n+1 >= 1$.

Since x is not empty, therefore, x must be constructed as either

1. aya for some y in L, or

2. bzb for some z in L

$|x| = 2+|y|$ ∴ $|y| = n-1$, ∴ By IH, $|y|$ is even ⇒ n-1 is even,

For case 1, $|y| <= n$ and y is in L. By IH, $|y|$ is even. Thus, $|x|=|y|+2$ is also even. Case 2 is similar.

What to do with a language?

Construct a machine/algorithm to decide membership.

Questions:

Are all languages "solvable"?

Can we say a solvable language is "easier" compared to another one?

What is the best way to "solve" a language?

A simple iterative one-pass function

```
boolean my_func(input) {
    // define few local vars
    for i in input {
        switch(i) {
            case ...:
            case ...:
            case ...:
            case ...:
        }
    }
    return ... // True or False
}
```

string

constant

"$x = y + z$" x, y, z are binary rep. of some integer I_x, I_y, I_z & $I_x = I_y + I_z$

but not unbounded data structure
linked list, stacks, queues, trees

single pass

"11 = 10 + 01" → True

"1111 = +01" → False

"111 = 111 + 111" → False

"101 = 011 + 010" → True

$L = \{ x : \text{my\_func}(x) \to \text{True} \}$

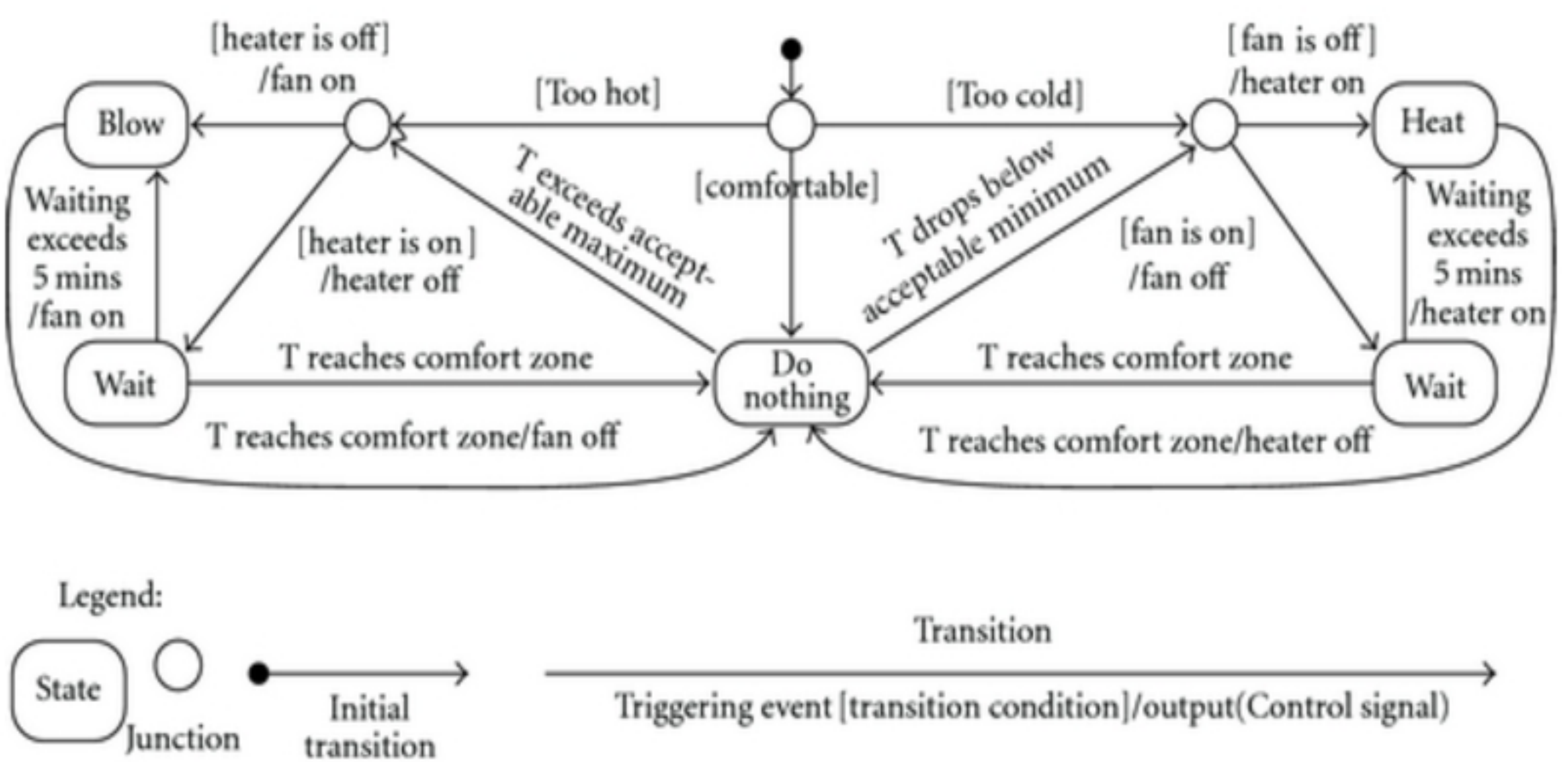
language ≡ Boolean problem
≡ Decision problem

(Deterministic) Finite Automata

Finite state machine DFA

McCulloch, W. S.; Pitts, E. (1943).
"A logical calculus of the ideas imminent in nervous activity"

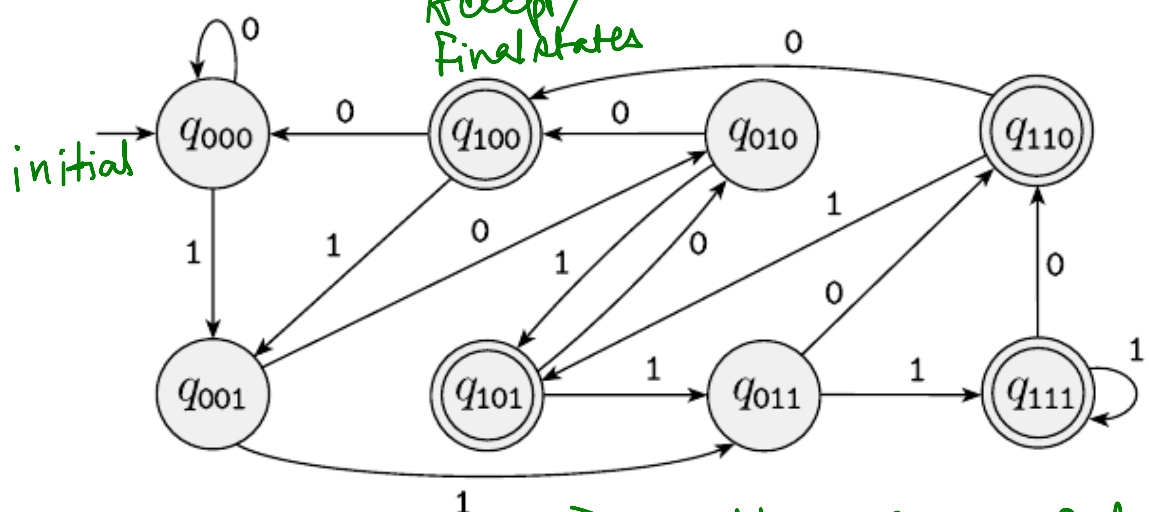
Rabin, M. O.; Scott, D. (1959).
"Finite automata and their decision problems.".



Formalization of DFA

$\Sigma = \{0, 1\}$

Q: set of states
Σ: alphabet
$q0$: starting state
$F \subseteq Q$: set of final states
$\delta: Q \times \Sigma \rightarrow Q$



DFA(11011) = $q_{000} \xrightarrow{1} q_{001} \xrightarrow{1} q_{011} \xrightarrow{0} q_{110} \xrightarrow{1} q_{101} \xrightarrow{1} q_{011}$
rejects 11011

DFA(1010101010011) = reject

D accepts s if the 3rd last bit of s is 1.

$\delta(q_{000}, 0) = q_{000}$
$\delta(q_{101}, 1) = q_{011}$

$L(D) = \{ s : D \text{ accepts } s \}$

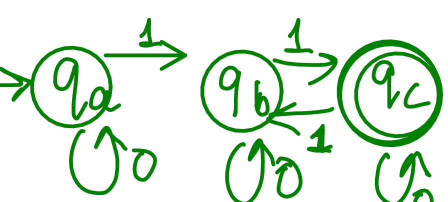
DFA = ⟨Q, Σ, δ, q0, F⟩

δ	0	1
→ q_{000}	q_{000}	q_{001}
q_{001}	q_{010}	...
q_{010}	..	..
q_{011}		
* q_{100}		
* q_{101}		
* q_{110}		
* q_{111}		

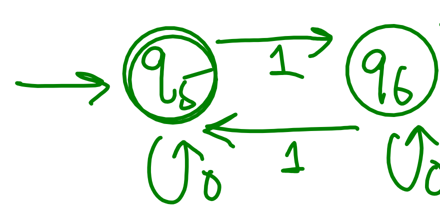
DFA "accepts" s (string) if its ends up in an accept state after reading s.
DFA "rejects" s if D doesn't accept S.

Q: Formally write a DFA to check if even number of 1s.

$L = \{x : \#1(x) \geq 2 \ \& \ \#1(x) \text{ is even}\}$



$L = \{011, 11, 110, \dots\}$



$L = \{x : \#1(x) \text{ is even}\}$

$L = \{\varepsilon, 011, 11, 1010, 0, 00, \dots\}$

? Questions to ask ?

* Can DFAs be constructed for every problem ?
* For problems that allow DFAs, how to construct a "best" DFA?
* What happens if we allow DFA++ ?

Language of DFA

$$r_0 \xrightarrow{s_1} r_1 \xrightarrow{s_2} r_2 \rightarrow \cdots \rightarrow r_{n-1} \xrightarrow{s_n} r_n$$

M $= \langle Q, \Sigma, \delta, q_0, F \rangle$ accepts string $s = s_1 s_2 \cdots s_n$ where $s_i \in \Sigma$, if there exists states $r_0, \ldots, r_n \in Q$ such that

1. $r_0 = q_0$
2. $r_n \in F$
3. $\forall\, i = 1 \ldots n,\ \delta(r_{i-1}, s_i) = r_i$

$$\left.\begin{matrix} \delta(r_0, s_1) = r_1 \\ \delta(r_1, s_2) = r_2 \\ \vdots \end{matrix}\right.$$

M recognizes L' / L' is language of M / L' = L(M)

$L' = \{x : M \text{ accepts } x\}$

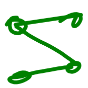
(Discrete) Computational Problems

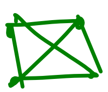
Function problems: output is many-valued

Decision problems: output is Boolean — if element has a property, element is a yes instance of the problem.

Language L = set of strings, from a universe, which satisfy some property — $L = \{x : D(x) \text{ returns true}\}$

2COLORABLE


yes


no

$L = \{ab, abba, ba, bba, \ldots\}$ — $D(x)$: Is $x \in L$?

Language and Decision problems are EQUIVALENT!

Decision problem = question of deciding membership of input in some particular language.

$LSORT = \{\langle x_1 \cdots x_k ; i ; y\rangle : x_1 \cdots x_k$ are integers, $i \in \{1 \ldots k\}$, $y \in \{x_1 \cdots x_k\}$, & y is the i-th smallest integer in $\{x_1 \cdots x_k\}\}$

(Discrete) Computational Problems

Decision problems: output is Boolean, decide if input has a specified property.

- Does input x represent adj.mat. of a conn. graph?
- Does input x represent two coprime integers?

Language and Decision problems are EQUIVALENT!

Input: list LI of integers

Q: Does LI have any duplicate? Yes if duplicate exists.

Represent the above problem as membership of L:

$L = \{ \langle x1 \ldots xk \rangle$: k is integer, x_i are integers &

there exists distinct i and j s.t. $x_i = x_j$ $\}$