BU CS 332 – Theory of Computation

https://forms.gle/gsUoYPKnehDafk3dA

Lecture 8:

- More on non-regularity
- Turing Machines

Reading:

"Myhill-Nerode" note

Sipser Ch 3.1, 3.3

Mark Bun

September 29, 2022

Last Time: Distinguishing Set Method

Definition: Strings x and y are **distinguishable** by L if there exists a "distinguishing extension" z such that exactly one of xz or yz is in L.

Definition: A set of strings S is **pairwise distinguishable** by L if every pair of distinct strings $x, y \in S$ is distinguishable by L.

Theorem: If S is pairwise distinguishable by L, then every DFA recognizing L needs at least $|S|$ states

Corollary: If S is an infinite pairwise dist. set. for L
Then L is not regular

Now you try!

Use the distinguishing set method to show that the following languages are not regular

$L_3 = \{1^{n^2} \mid n \geq 0\}$

$1111 \in L_3$

$11111 \notin L_3$

$S = \{1^{n^2} \mid n \geq 0\}$

Let $x = 1^{n^2}$ $y = 1^{m^2}$ Assume WLOG $m > n$

Set $z = 1^{2n+1}$

Then:

$xz = 1^{n^2} \circ 1^{2n+1} = 1^{n^2+2n+1} = 1^{(n+1)^2} \in L_3$

$yz = 1^{m^2} \circ 1^{2n+1} = 1^{m^2+2n+1} \notin L_3$

$m^2 < \underbrace{m^2 + 2n + 1}_{\text{not a perfect square!}} < (m+1)^2 = m^2+2m+1$

Reusing a Proof

Finding a distinguishing set can take some work…

Let's try to reuse that work!

How might we show that $00100111 \in BALANCED$

$$BALANCED = \{w \mid w \text{ has an equal \# of 0s and 1s}\}$$

is not regular?

$L(0^*1^*)$

$$\overbrace{\{0^n1^n \mid n \geq 0\}}^{\text{Not Regular}} = \overbrace{BALANCED}^{???} \cap \overbrace{\{w \mid \text{all 0s in } w \text{ appear before all 1s}\}}^{\text{REGULAR}}$$

Claim: BALANCED is not regular

Proof. Assume FTSOC BALANCED is regular

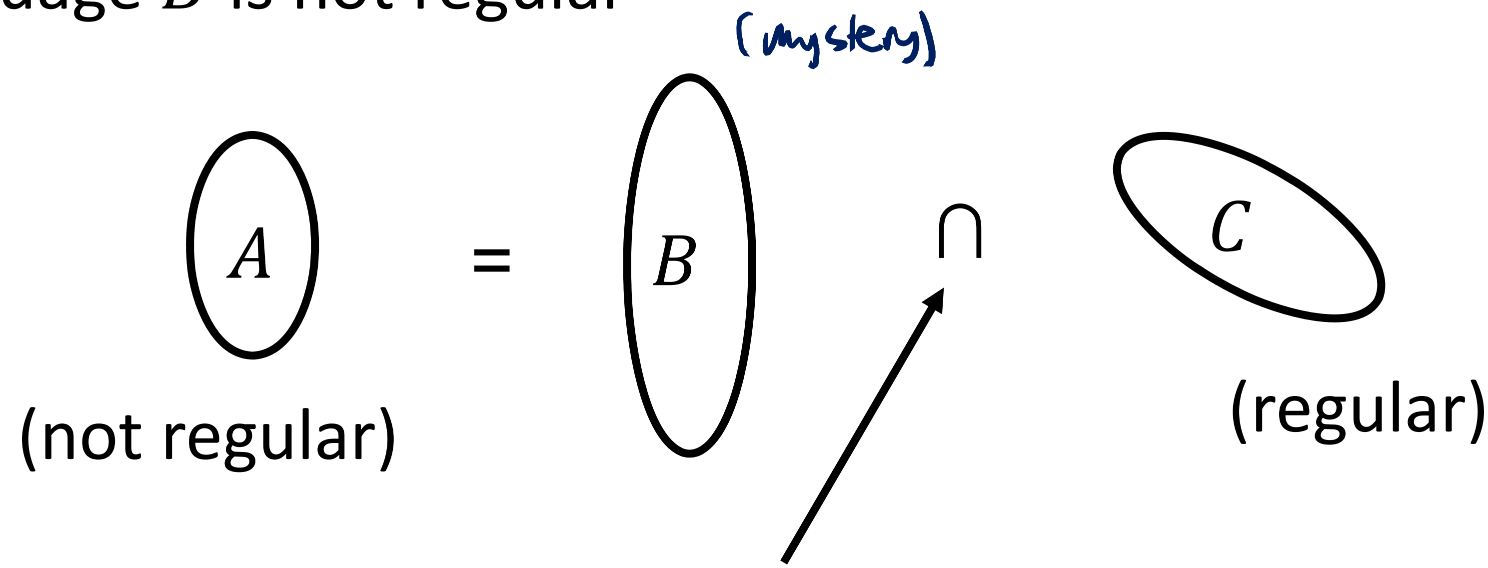
$\Rightarrow BALANCED \cap \{w \mid \text{all 0's appear before all 1's}\}$ is regular (closure under $\cap$)

$\Rightarrow \{0^n1^n \mid n \geq 0\}$ is regular $\times$ conclude BALANCED not regular.

Using Closure Properties

Six closure properties: $\boxed{\cup}$, $\cap$, $\boxed{\circ}$, reverse, complement, $\boxed{\text{star}}$

If A is not regular, we can show a related language B is not regular

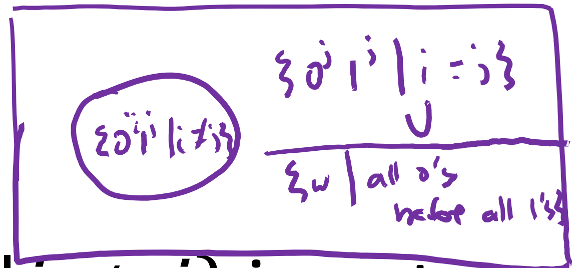


any of $\{\circ, \cup, \cap\}$ or, for one language, $\{\neg, {}^R, *\}$

<u>By contradiction:</u> If B is regular, then $B \cap C$ $(= A)$ is regular.

But A is not regular so neither is B!

Example



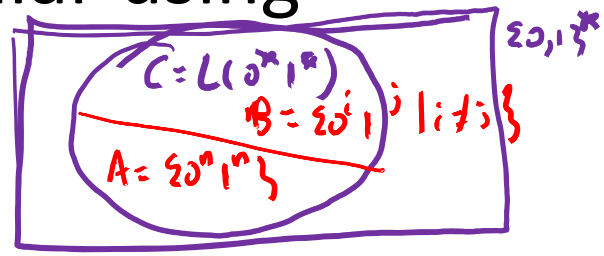
Prove $B = \{0^i 1^j | i \neq j\}$ is not regular using

- nonregular language

  $A = \{0^n 1^n | n \geq 0\}$ and

- regular language

  $C = \{w \mid \text{all 0s in } w \text{ appear before all 1s}\}$ $= L(0^* 1^*)$



$A = C \setminus B$

$= C \cap \overline{B}$

Which of the following expresses A in terms of B and C?

a) $A = B \cap C$

b) $A = \overline{B} \cap C$

c) $A = B \cup C$

d) $A = \overline{B} \cup C$

Proof that B is nonregular

Assume for the sake of contradiction that B is regular

We know: $A = \bar{B} \cap C$

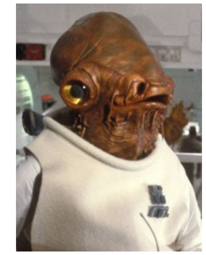
$\Rightarrow \bar{B}$ is regular (reg. langs. closed under complement)

$\Rightarrow \bar{B} \cap C$ is regular (closure under intersection)

$\Rightarrow A$ is regular

$\times$ conclude B is nonregular

!DANGER!



Let $B = \{0^i 1^j | i \neq j\}$ and write $B = A \cup C$ where

- nonregular language

  $A = \{0^i 1^j | i > j \geq 0\}$ and

- nonregular language

  $C = \{0^i 1^j | j > i \geq 0\}$ and

Does this let us conclude B is nonregular?

NOT the case that nonregular are closed under union

Ex: Let A be any nonregular language

Let $C = \overline{A}$ nonregular [why? If $\overline{A}$ were regular, then $\overline{(\overline{A})}$ would be regular, but $A = \overline{(\overline{A})}$]

$B = A \cup C \Rightarrow B = \Sigma^*$ is regular

Turing Machines

Turing Machines – Motivation

We’ve seen finite automata as a restricted model of computation

Finite Automata / Regular Expressions

- Can do simple pattern matching (e.g., substrings), check parity, addition
- Can’t perform unbounded counting $\{0^n 1^n \mid n \geq 0\}$
- Can’t recognize palindromes

Somewhat more powerful (not in this course):

Pushdown Automata / Context-Free Grammars

- Can count and compare, parse math expressions
- Can’t recognize $\{a^n b^n c^n \mid n \geq 0\}$

Turing Machines – Motivation

Goal:

Define a model of computation that is

1) General purpose. Captures all algorithms that can be implemented in any programming language.

2) Mathematically simple. We can hope to prove that things are not computable in this model.

A Brief History

1900 – Hilbert's Tenth Problem

Given: $p(x,y,z) = 2x^2y + 4z - 2x$

Does there exist $(x,y,z) \in \mathbb{Z}^3$ s.t. $p(x,y,z) = 0$?

$x = 1, \; y = 1, \; z = 0$

Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.

An algorithm

David Hilbert 1862-1943

1928 – The *Entscheidungsproblem*

Wilhelm Ackermann 1896-1962

The "Decision Problem"

Is there an algorithm which takes as input a formula (in first-order logic) and decides whether it's logically valid?

Input:
Mathematical statement

Is the math statement true or false?

David Hilbert 1862-1943

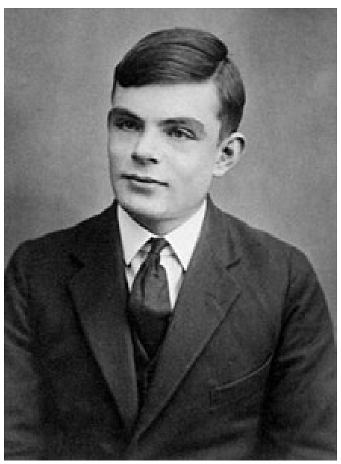
1936 – Solution to the *Entscheidungsproblem*

Alonzo Church 1903-1995

"An unsolvable problem of elementary number theory“

Model of computation: λ-calculus (CS 320)

~ regular expression



Alan Turing 1912-1954

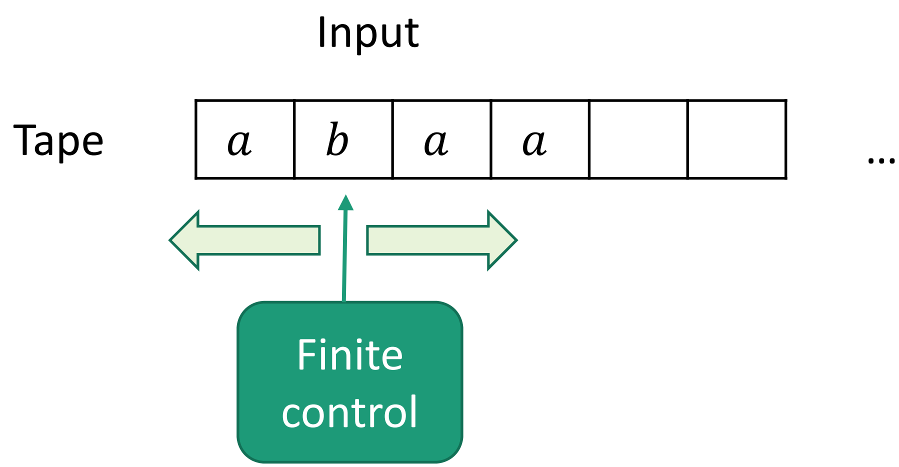
“On computable numbers, with an application to the *Entscheidungsproblem*”

Model of computation: Turing Machine

~ Finite automata

Turing Machines

The Basic Turing Machine (TM)



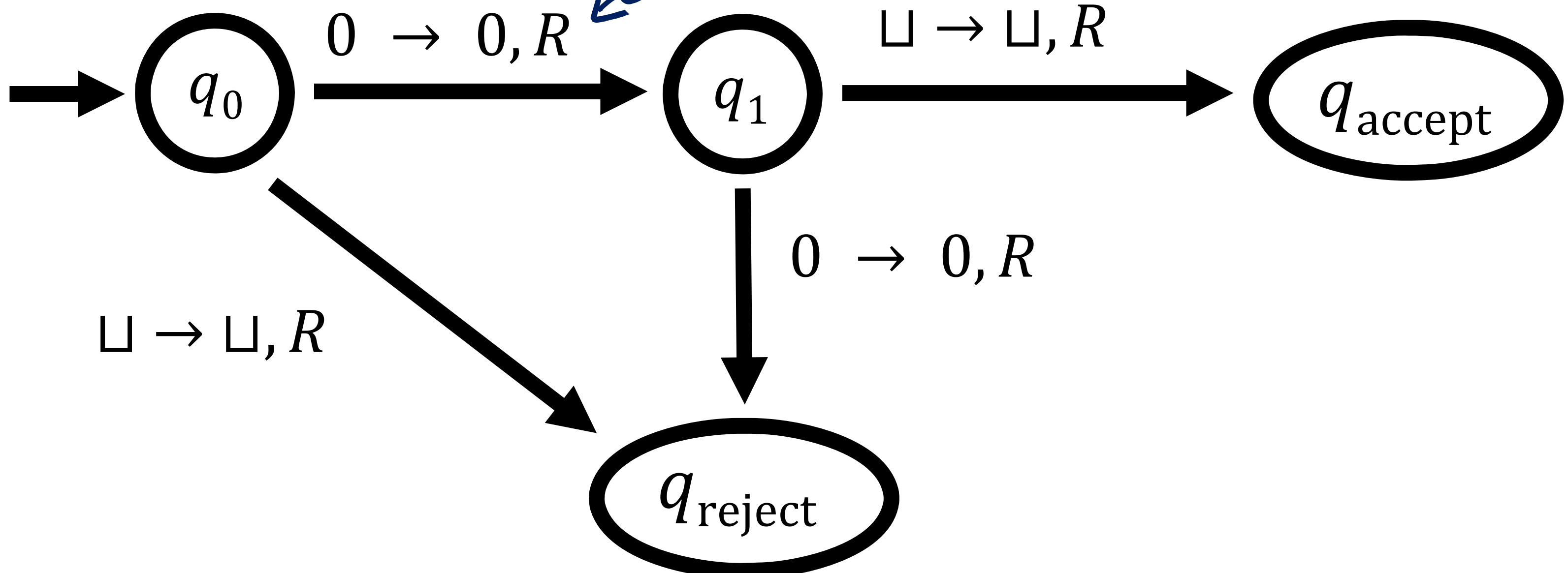
- Input is written on an infinitely long tape
- Head can both read and write, and move in both directions
- Computation halts as soon as control reaches “accept” or “reject” state

Example

If TM reads this symbol...

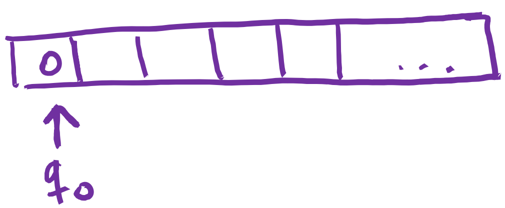
then it writes this symbol

and moves its head this way

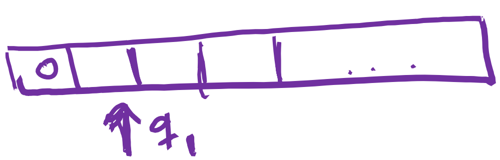


On input 0:

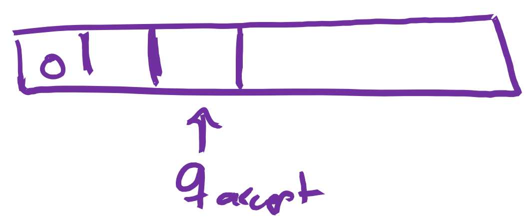
1)



2)

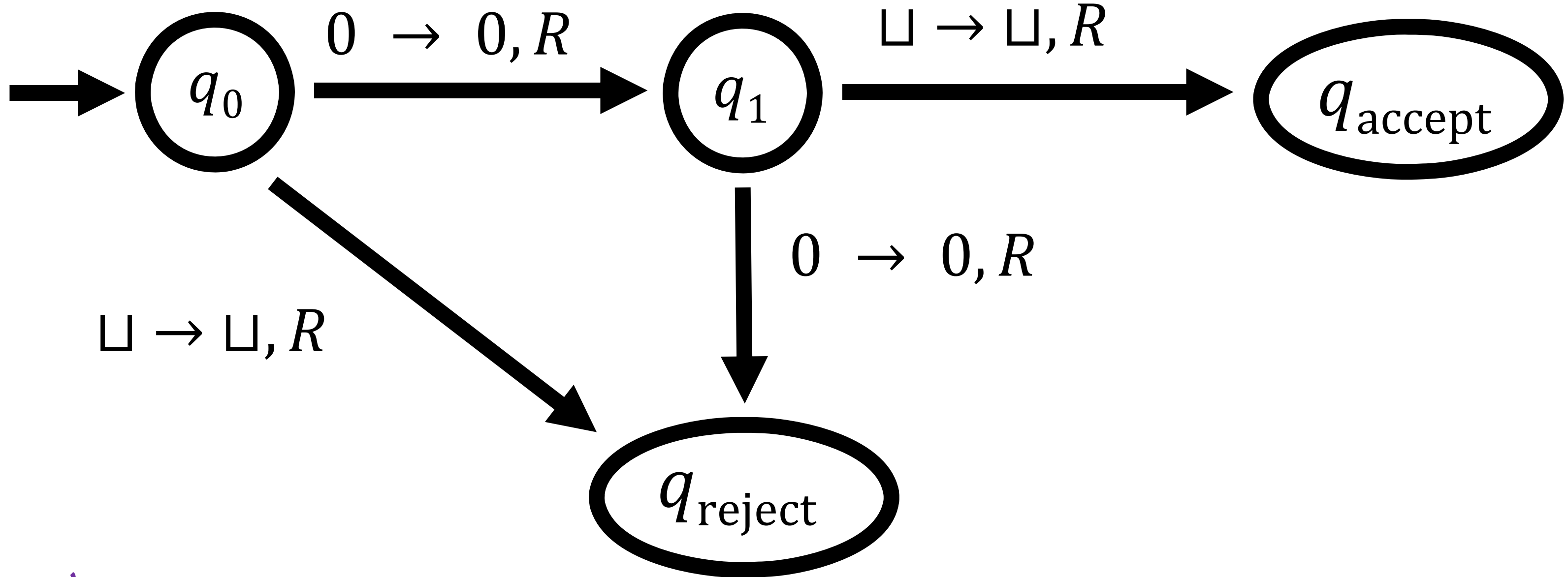


3)



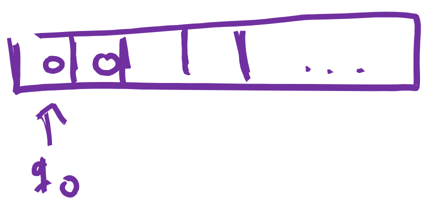
TM accepts!

Example

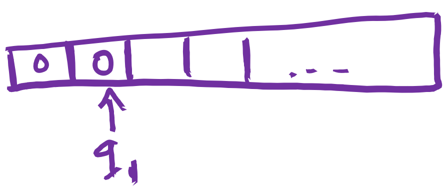


On input: 00

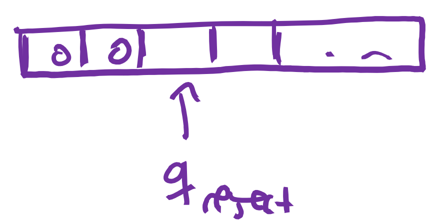
1)



2)



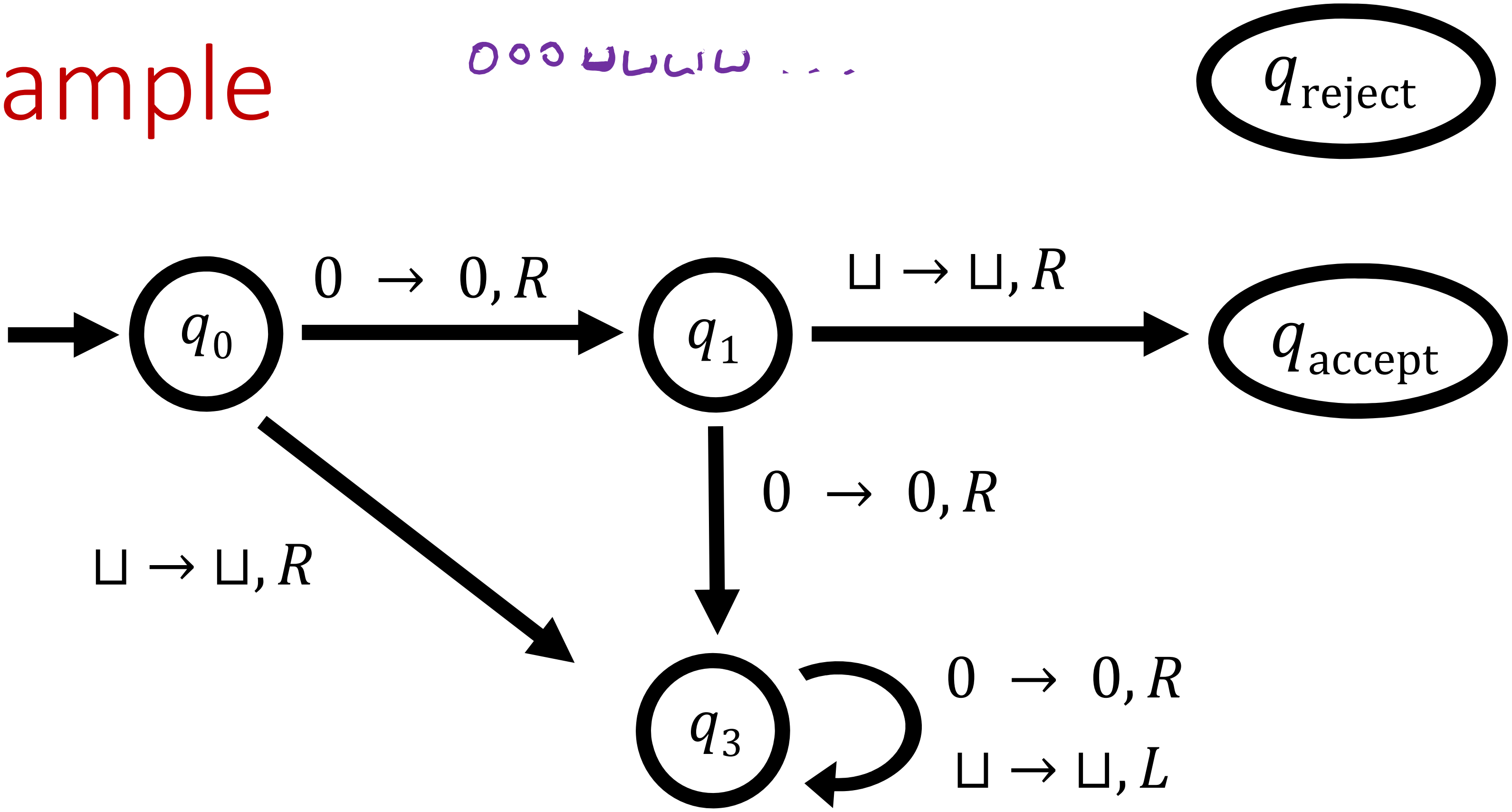
3)



TM rejects!

Example

000⊔⊔⊔⊔⊔ . . .



What does this TM do on input 000?

a) Halt and accept
b) Halt and reject
c) Halt in state q_3
d) Loop forever without halting

Three Levels of Abstraction

Programming

High-Level Description

An algorithm (like CS 330)

Python, Java

Implementation-Level Description

Describe (in English) the instructions for a TM

- How to move the head
- What to write on the tape

C

Low-Level Description

State diagram or formal specification

Assembly
Machine code

Example

Determine if a string $w \in \{0\}^*$ is in the language

$A = \{0^{2^n} \mid n \geq 0\}$

High-Level Description

Repeat the following forever:

- If there is exactly one 0 in w, accept
- If there is an odd (> 1) number of 0s in w, reject
- Delete half of the 0s in w

Example

Determine if a string $w \in \{0\}^*$ is in the language

$$A = \{0^{2^n} \mid n \geq 0\}$$

$\cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \sqcup\sqcup\sqcup\sqcup$

Implementation-Level Description

1. While moving the tape head left-to-right:
   a) Cross off every other 0 (i.e. replace 0 with x)
   b) If there is exactly one 0 when we reach the right end of the tape, accept
   c) If there is an odd (> 1) number of 0s when we reach the right end of the tape, reject
2. Return the head to the left end of the tape
3. Go back to step 1

Example

Determine if a string $w \in A = \{0^{2^n} \mid n \geq 0\}$

Low-Level Description