CSIS-616 - Chapter 4, Part 1

Ralph W. Crosby, PhD.

College of Charleston

Thursday, October 15, 2020

Definition of Algorithm

Church-Turing Hypothesis

- The *Entscheidungsproblem* (Hilbert)
- Turing - Turing Machines as an idealized model of computation
- Church - Lambda calculus

`https://plato.stanford.edu/entries/church-turing/`

So what is an algorithm?

- The question is whether a given question (problem) is decidable?
- If it is, can a method be devised to compute the decision?
- This method is what we refer to as an algorithm

Definitions of Language Recognizers

Regular Languages

$M_{RL} = \{Q, \Sigma, \delta, q_0, F\}$
$\delta : Q \times \Sigma \rightarrow Q$

Context Free Languages

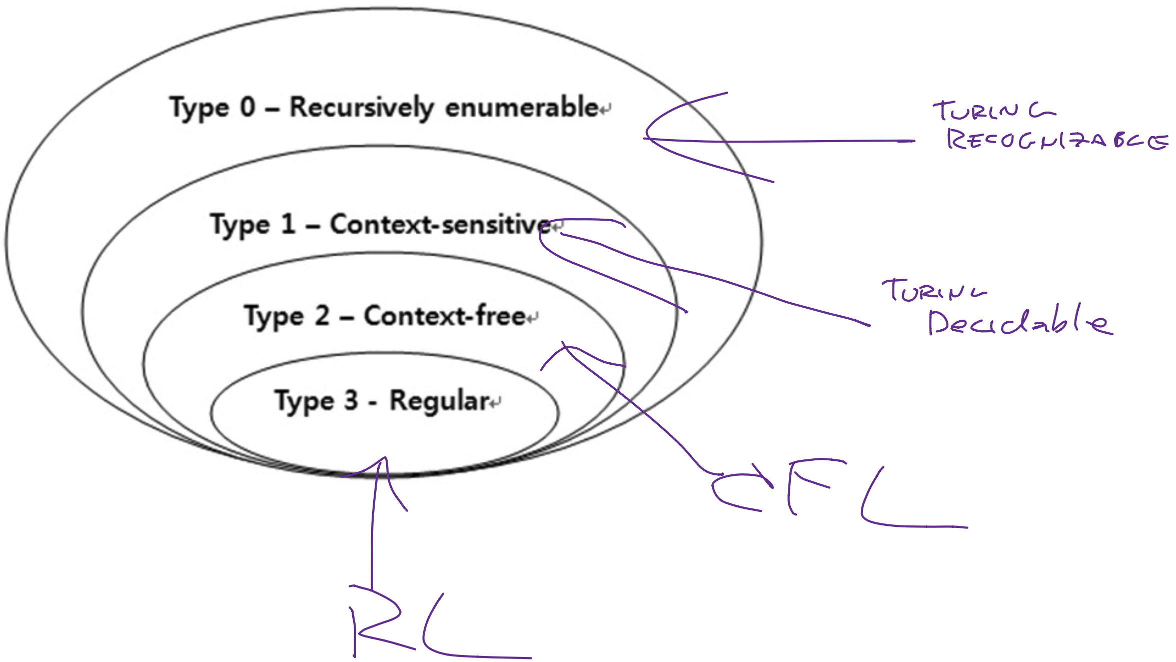
- CFL: $M_{CFL} = \{V, \Sigma, R, S\}$
- PDA: $M_{PDA} = \{Q, \Sigma, \Gamma, \delta, q_0, F\}$
  $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$

Turing Machines

$M_{TM} = \{Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject}\}$
$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

- Recursive Languages - (Turing Decidable)
- Recursively Enumerable Languages - (Turing Recognizable)

Chomsky's Hierarchy



Decidability

Is a problem solvable - Hilbert's Question

Why do we care?

- Don't want to waste time on things we can't solve
- Bound what we can solve using a general purpose computer

Decidable Languages

- Can we determine if a given language is decidable?

Decidability and Regular Languages

Acceptance Problem

$$A_{DFA} = \{< B, w > | B \text{ is a DFA that accepts string } w\}$$

$$A_{NFA} = \{< B, w > | B \text{ is an NFA that accepts string } w\}$$ CONVERT TO DFA USE

$$A_{REX} = \{< B, w > | B \text{ is a regular expression that generates } w\}$$

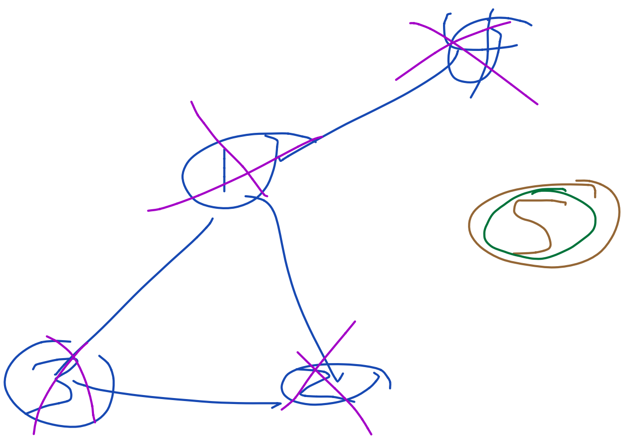
SIMULATE B ON A TURING MACHINE WITH INPUT w.
- LOAD w ONTO TAPE
- START AT LEFT OF TAPE
- EVERY TRANSITION WILL MOVE RIGHT
- AT END
  - ACCEPT
  - REJECT

  STATES

DOESN'T WORK
w CONTAINS symbols NOT IN Σ

Emptyness Problem

$$E_{DFA} = \{< A > | A \text{ is a DFA and } L(A) = \emptyset\}$$



#1 2 3 4 5 #12#

13#

14#

R

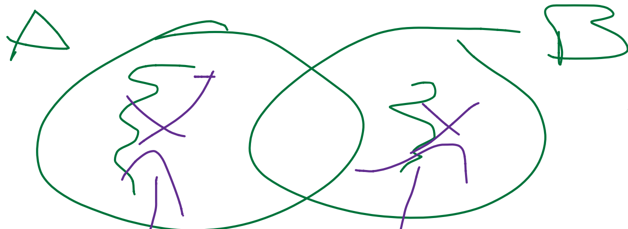
State Diagram

- Mark Start State
- Mark all states that have transitions from our start state
  - Repeat for all marked states
- If no marked states are accept states ⇒ accept

Decidability and Regular Languages

Equality Problem

$$EQ_{DFA} = \{< A, B > | A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$$



$\longrightarrow A \neq B$

$\longrightarrow A = B$

$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$

$L(C) = \emptyset$

$\uparrow$ E_{DFA}

Rust - Functional Programming

- *iter*() function

FOR VAR IN COLLECTIONS
}
MOVE VAR OUT OF COLLECTION

for VAR IN
collection.iter(){
}
RETURN Refer
TO ITEMS

- *enumerate*() function

Let i = 0;
for x in Array.iter() {
i = i + 1;
}

for (i,x) in Array.iter().enumerate(){
i IS AVAILABLE STARTING AT 0
}

- Closures - ONE LINE FUNCTION

Let c = |v| v+1;
c(10) == 11

LET v1 = 20;
LET c1 = |q| q+v1;
println!(c1(5))
25 printed

- *map*() function

Let c1 = collection.iter().map(|c| c+1).collect();
Let collection = vec![1,2,3]
USE THE LINE ABOVE
[2,3,4]

- *for_each*() function

SIMILAR TO MAP
- DOESN'T RETURN THE ITEM TO THE ITERATOR
collection.iter().for_each(print_it);

Decidability and Context Free Languages

Acceptance Problem

$$A_{CFG} = \{< G, w > | G \text{ is a CFG that generates string } w\}$$

Chomsky Normal Form
- Binary Tree
- $2n-1$ Leaves
  ↳ n is # productions

- Convert G to CNF
- List all $2n-1$ derivations
- If any derivation generates w ⇒ Accept
  otherwise reject

Decidability and Context Free Languages

Emptyness Problem

$$E_{CFG} = \{< G > | G \text{ is a CFG and } L(G) = \emptyset\}$$

- SIMILAR MARKING ALGORITHM
- MARK ALL TERMINAL SYMBOLS IN W
- SCAN RULES, MARK ALL VARIABLES THAT APPEAR IN A RULE
  - REPEAT FOR ALL TERMINALS & MARKED VARIABLES
- IF START VARIABLE IS NOT MARKED → ACCEPT OTHERWISE REJECT

$E \to W S$
$W \to W x$
$S \to S S$
→ REJECT

$E = W S$
$Q = x$
ACCEPT

Decidability and Context Free Languages

Equality Problem

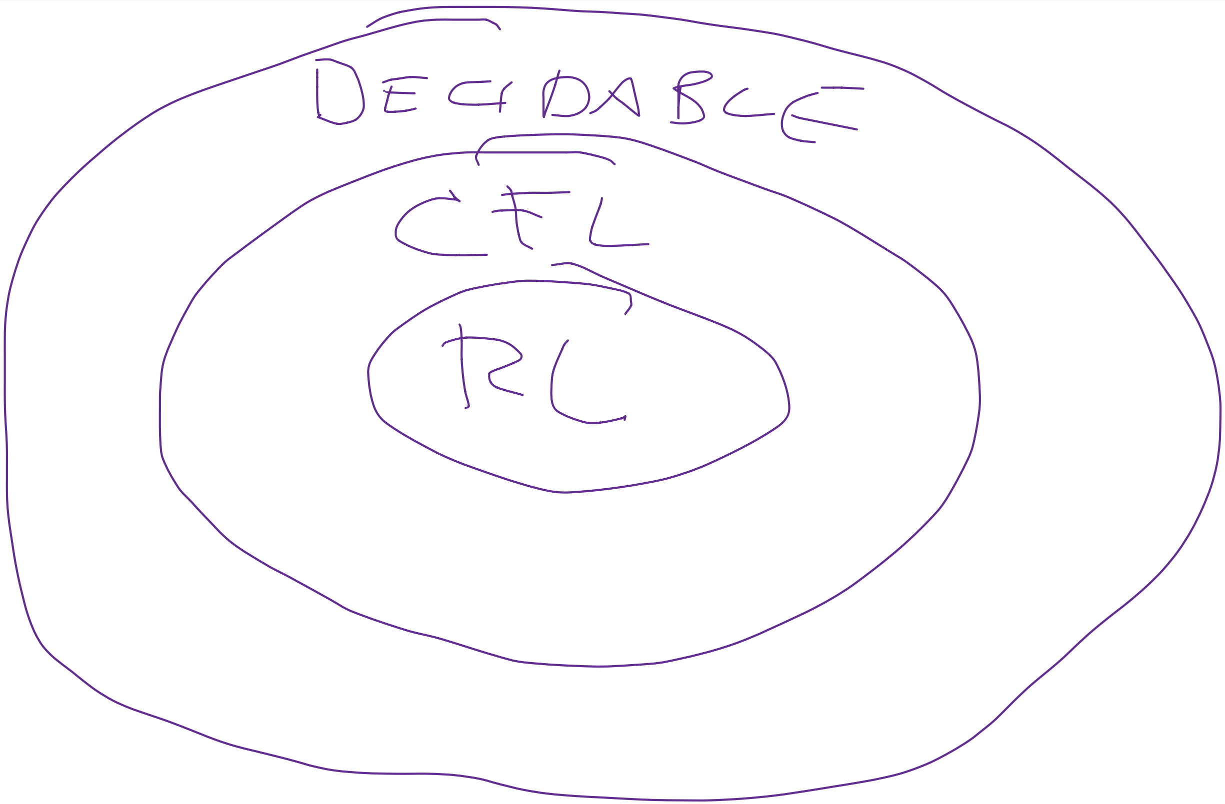
$$EQ_{CFG} = \{< G, H > | G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$$

CONCAT, UNION, STAR
↳ CLOSED FOR CFG
- SAME "VEN" DIAGRAM proof FOR CFG'S

Decidability and Context Free Languages

Decidability Problem

Every context-free language is decidable



- Represent a DFA, CFL as a Turing machine
- Test if Turing machine is decidable