18.404 Notes

Jason Chen
Lecturer: Michael Sipser

Last Updated: February 5, 2021

Contents

0.1 Decision Problems 2
0.1.1 DFAs 2
0.1.2 CFGs 3

0.1 Decision Problems

Sometimes, we want to work with objects that are not strings. If A is an object, then let's say that $\langle A \rangle$ is an **encoding** of A into a string over Σ. If we have multiple objects $A_1, \ldots, A_k$, then we can encode all of them into one string using $\langle A_1, \ldots, A_k \rangle$. Note that $\langle A_1, \ldots, A_k \rangle$ is not necessarily just $\langle A_1 \rangle, \ldots, \langle A_k \rangle$ concatenated, because it might not be clear where each one ends.

0.1.1 DFAs

Acceptance Problem

Given a DFA, we would like to determine whether it will accept some string w. In particular, let

$$A_{\text{DFA}} = \{\langle B, w \rangle \mid B \text{ is a DFA that accepts } w\}.$$

Therefore, if testing if $a \in A$ is equivalent to determining if some DFA accepts some word.

1 **Proposition**

A_{DFA} is decidable.

Proof. Let M be the following Turing machine with input $s = \langle B, w \rangle$:

- Test if $s = \langle B, w \rangle$ rejects or not.
- Simulate B on W.
- If B accepts, ACCEPT, else REJECT. □

We can define a similar language for NFAs.

2 **Corollary**

$A_{\text{NFA}} = \{\langle B, w \rangle \mid B \text{ is a NFA that accepts } w\}$ is decidable.

To prove this, we can just do the same thing as above, and simulate the NFA. However, we can do something clever and just reduce it to the previous problem.

Proof (Sketch). Convert B to a DFA B' and then use the fact that A_{DFA} is decidable. □

Emptiness Problem

3 **Proposition**

Let $E_{\text{DFA}} = \{\langle B\rangle \mid B \text{ is a DFA where } L(B) = \varnothing\}$. Then E_{DFA} is decidable.

Proof. The idea is to just perform a graph search on the states. Consider the TM M with input $\langle B\rangle$, described by:

- Mark the start state
- Repeat until nothing new is marked:
  - Mark state q if some previously marked state points to q.
- Accept if no accept node is marked, otherwise reject. □

Equivalence Problem

4 **Proposition**

Let $EQ_{\text{DFA}} = \{\langle B, C\rangle \mid B, C \text{ are DFAs and } L(B) = L(C)\}$. Then EQ_{DFA} is decidable.

Proof. The key idea is that if $L(B) = L(C)$, then we know that the symmetric difference is empty, i.e.

$$L(B) \,\Delta\, L(C) = \overline{L(B)} \cap L(C) \cup L(B) \cap \overline{L(C)} = \varnothing.$$

Therefore, we consider the following TM with input $\langle B, C\rangle$:

- Let D be the DFA such that
$$L(D) = \overline{L(B)} \cap L(C) \cup L(B) \cap \overline{L(C)}.$$
- Test if $\langle D\rangle \in E_{\text{DFA}}$, i.e. if $L(D)$ is empty. If yes then accept; reject if no. □

0.1.2 CFGs

5 **Proposition**

Let $A_{\text{CFG}} = \{\langle G, w\rangle \mid G \text{ is a CFG such that } w \in L(G)\}$. Then A_{CFG} is decidable.

Proof. Consider the following TM with input $\langle G, w\rangle$:

- Convert G to Chomsky's Normal form. We know that to derive a string of length n, it will take $2n-1$ steps ($n-1$ to get the correct number of symbols, and n to convert variables).
- Try all the derivations of length $2n-1$. If any of them yield w, then accept, else reject. □

6 **Proposition**

Every CFL is decidable.

Proof. Let A be a CFL generated by the CFG G. Consider the TM M_G with input w:

- Test if $\langle G, w\rangle \in A_{\text{CFG}}$.
- If yes then accept; reject if no. □

7 **Proposition**

Let $E_{\mathsf{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \varnothing\}$. Then E_{CFG} is decidable.

Proof. Suppose we are given a list of rules. We can mark all the terminals, and then if everything on the right side of a rule is marked, then this means we can get the left side. Repeat until nothing new is marked. Then, if the initial variable is marked then ACCEPT; else REJECT. □

8 **Proposition**

Let $EQ_{\mathsf{CFG}} = \{\langle G, H\rangle \mid G, H \text{ are CFGs and } L(G) = L(H)\}$. Then EQ_{CFG} is undecidable.

We can't use the same trick of using the symmetric difference we used last time for DFAs, because CFGs are not closed under complements or intersection! We'll prove this proposition later.

9 **Proposition**

Let $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM accepting } w\}$. Then A_{TM} is Turing recognizable.

Proof. The idea is to just make the machine do it. In particular, the following machine recognizes A_{TM}:

0. On input $\langle M, w\rangle$:
1. Simulate M on w.
2. If M accepts, then ACCEPT, if M rejects, then REJECT. □