CS 4510: Automata and Complexity **Fall 2019**

Lecture 6: Undecidability and The Pumping Lemma

September 9, 2019

Lecturer: Santosh Vempala *Scribe: Xiaofu Niu, Aditi Laddha*

6.1 Undecidability

6.1.1 The Acceptance Problem

Definition 6.1 Acceptance Language
$L_A = \{\langle M, x\rangle,\ M \text{ is valid TM description and } M \text{ accepts } x\}$

Theorem 6.2 *L_A is undecidable.*

Proof:
We prove this theorem by contradiction. Suppose there exists a TM D that decides L_A. On given input $\langle M, x\rangle$, D decides whether M accepts x. Consider a new TM $\hat{D}$ that takes the description of some TM $\langle M\rangle$ as input.

```
Run D on input ⟨M, ⟨M⟩⟩
if D accepts then
    D̂ rejects
else
    D̂ accepts
```

The contradiction occurs when we run $\hat{D}$ on input $\hat{D}$. According to the definition of D, $\hat{D}$ accepts $\langle\hat{D}\rangle$ if and only if $\hat{D}$ rejects $\langle\hat{D}\rangle$. ■

6.1.2 The Halting Problem

Definition 6.3 Halting Language
$L_{HALT} = \{\langle M, x\rangle,\ M \text{ is valid TM description and } M \text{ halts on } x\}$

Theorem 6.4 *L_{HALT} is undecidable.*

Proof:
We prove this theorem by reduction. Suppose there exists a TM H that decides L_{HALT}. On given input $\langle M, x\rangle$, H decides whether M halts on (either accept or reject) x. Consider a new TM A that takes the

description of some TM $\langle M \rangle$ and a string x as input.

```
Run H on input ⟨M, x⟩
if H rejects then
    A rejects
else
    Run M on x
    if M accepts x then
        A accepts
    else
        A rejects
```

Based on the definition, we have constructed a TM A that decides the acceptance problem L_A. However, we have proved that L_A is undecidable. Therefore, the assumption is wrong. ∎

6.2 A Pumping Lemma for Deterministic Finite Automata

In previous lectures, we have seen Turing machines that decide $L_1 = \{0^i 1^i : i \in \mathbb{N}\}$ and $L_2 = \{0^i : i \text{ is a power of } 2\}$.

Question: Given a language L, can you prove that there is no DFA that accepts L?

- $\mathbf{L_1}$: Assume that there exists a DFA $D = (Q, \Sigma, \delta, q_0, F)$ that accepts L_1 and $|Q| = p$. Then consider the string $x = 0^n 1^n$ where $n > p$. Let $q_0 q_1 q_2 \dots q_i \dots q_j \dots q_{2n}$ be the sequence of states that D goes through when computing with input x. In other words, $\delta(q_i, x_{i+1}) = q_{i+1}, \forall i \in \{0, \dots, 2n-1\}$ and $q_{2n} \in F$. This sequence consists of $n+1$ states corresponding to state transitions on seeing the first n 0's. But the DFA has $p < n+1$ states and hence there must be a repeated state. Let q_j be the first repeated state such that there exists $i < j$ with $q_i = q_j$. Let l be the substring of x that takes D from q_i to q_j.

  $$x = 0^i \underbrace{0\dots0}_{l} 0^{n-j} 1^n$$

  Let y be defined as

  $$y = 0^i \underbrace{0\dots0}_{l} \underbrace{0\dots0}_{l} 0^{n-j} 1^n$$

  On input y, 0^i takes D from state q_0 to q_i, and l takes D from q_i to q_i. If we add one more l after the first one, D will still be in state q_i and $0^{n-j}1^n$ takes D from state q_i to $q_{2n} \in F$. So, y is accepted by D but number of 0's in $y = n + |l| > n =$ number of $1's$ in y. A contradiction.

- $\mathbf{L_2}$: Assume that there exists a DFA $D = (Q, \Sigma, \delta, q_0, F)$ that accepts L_2 and $|Q| = p$. Then consider the string $s = 0^n$ where n is a power of 2 and $n \geq p$. with $q_0 q_1 q_2 \dots q_i \dots q_j \dots q_n$ as the sequence of states that D goes through when computing with input s. This sequence consists of $n+1$ states but the D has $p < n+1$ states and hence there must be a repeated state. Let q_j be the first repeated state such that there exists $i < j$ with $q_i = q_j$. Let y be the substring of s that takes D from q_i to q_j. Then,

  $$s = 0^i \underbrace{0\dots0}_{y} 0^{n-j}$$

  $$s_1 = 0^i \underbrace{0\dots0}_{y} \underbrace{0\dots0}_{y} 0^{n-j}$$

$$s_2 = 0^i \underbrace{0 \dots 0}_{y} \underbrace{0 \dots 0}_{y} \underbrace{0 \dots 0}_{y} 0^{n-j}$$

Then s_1 and s_2 are also accepted by D and hence their lengths must be powers of 2. Let $|s_1| = 2^m$ and $|s_2| = 2^l$. Also, $k < m < l$ as $|y| > 0$. In other words,

$$|x| + |y| + |z| = 2^k$$
$$|x| + 2|y| + |z| = 2^m$$
$$|x| + 3|y| + |z| = 2^l$$

This implies $|y| = 2^l - 2^m = 2^m - 2^k \Rightarrow 2^m = 2^{l-1} + 2^{k-1}$ which can only be true if $l = k$. A contradiction.

6.2.1 Pumping Lemma for DFA

Lemma 6.5 *If L is a regular language, then there exists an integer $p > 0$ such that for any string $s \in L$ with $|s| \geq p$, s can be written as $s = xyz$ satisfying the following conditions:*

1. $\forall i \geq 0$, $xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$

Proof:
Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L. Let $p = |Q|$. Consider any string $s \in L$ with $|s| = n \geq p$. Let $q_0 q_1 q_2 \dots q_i \dots q_j \dots q_n$ be the sequence of states that D goes through when computing with input s. In other words, $\forall i \in \{0, \dots, n-1\}, \delta(q_i, s_{i+1}) = q_{i+1}$ and $s \in L \Rightarrow q_n \in F$. This sequence consists of $n+1$ states but the DFA D has only $p < n+1$ states. So, there must be a repeated state. Let q_j be the first repeated state such that $q_i = q_j$ for some $i < j$. Then $j \leq p$ because j is the first repeated state. Then let $x = s_1 s_2 \dots s_i$, $y = s_{i+1} \dots s_j$ and $z = s_{j+1} \dots s_n$.

$$s = \underbrace{s_1 s_2 \dots s_i}_{x} \underbrace{s_{i+1} \dots s_j}_{y} \underbrace{s_{j+1} \dots s_n}_{z}$$

Also, $|y| > 0$ as the DFA must read at least 1 symbol to transition from q_i back to q_i.

On input xy^iz for any $i \geq 0$, x takes D from state q_0 to q_i, the first y takes D from state q_i to q_i and so do the subsequent y's. Reading xy^i will leave D in the state q_i for all $i \geq 0$ and z takes D from state q_j to q_n where $q_n \in F$. Hence, $xy^iz \in L$.

■

6.3 Reference

- Ch 1.2 Nondeterminism, "Introduction to the Theory of Computation"
- Ch 4.2 Undecidability, "Introduction to the Theory of Computation"
- Ch 1.4 Pumping Lemma, "Introduction to the Theory of Computation"