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

Lecture 5: Undecidability

September 4, 2019

Lecturer: Santosh Vempala *Scribe: Xiaofu Niu*

5.1 Nondeterministic Turing Machine

Question: Is NTM more powerful than TM?

Theorem 5.1 *Every NTM can be converted to a TM.*

Proof: Think about the computation of NTM as a tree. Each node in the tree represents a single computational state, while each edge represents a transition to the next possible computation. The branching factor of this computation tree is finite. For any input string S, if there exists a path in the computation tree which ends in a node whose state is accepting state, then S is accepted. Because the width of the computation tree is finite at any finite depth, we can convert this tree to a single path which represents the computation of a TM using BFS. ■

5.2 Undecidability

Question: Is every language decidable by some TM?

Possible hard languages:

- Traveling Salesman Problem
- 2-Player Game

There exist some languages that is undecidable by TMs, but both listed examples are decidable. Actually, most languages are undecidable. Next we will learn how to prove that there exists undecidable languages.

5.2.1 Countable

Definition 5.2 Countable *A set S is called countable if each element in the set can be assigned a unique natural number. In other words, there is a 1-1 mapping from S to natural numbers* $f : S \longrightarrow \mathbb{N}$.

Here are some examples of countable sets.

Example 1. The set of all integers is countable.

Proof: We use the following function to map it to natural number.

$$F(z) = \begin{cases} 2|z| & z \leqslant 0 \\ 2|z|+1 & z > 0 \end{cases}$$

■

Example 2. The set $Q = \{\frac{a}{b} | b \in \mathbb{N}, a \in F\}$ is countable.

Proof: Let's draw an infinite matrix where the column is a and row is b. The number $\frac{a}{b}$ occurs in the bth column and ath row. This is apparently a 1-1 unique mapping. Next we turn this matrix into a list by traversing this matrix by diagonals.

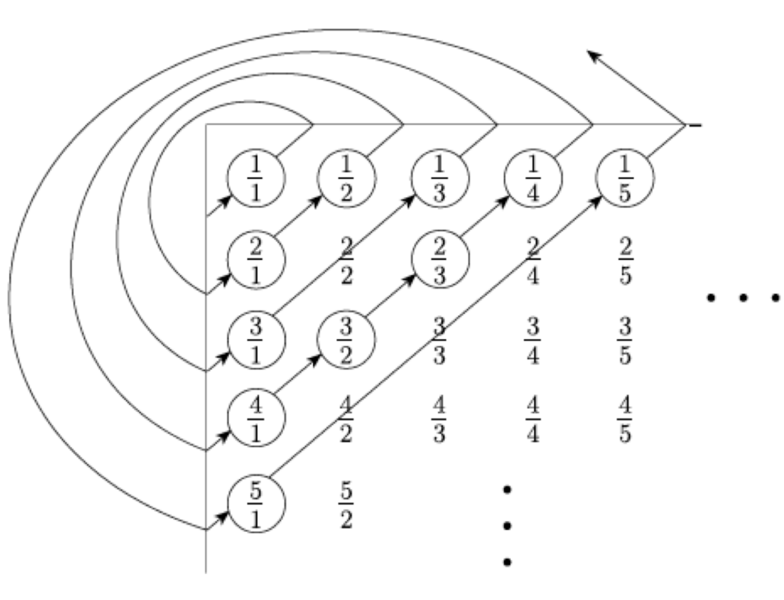

Figure 5.1: A corresponce of $\frac{a}{b}$

■

Example 3. Σ is a finite set of alphabet. Σ^* is countable.

Proof: Let's take the binary as example. For any binary strings, we add a '1' before it and map it to a natural number.

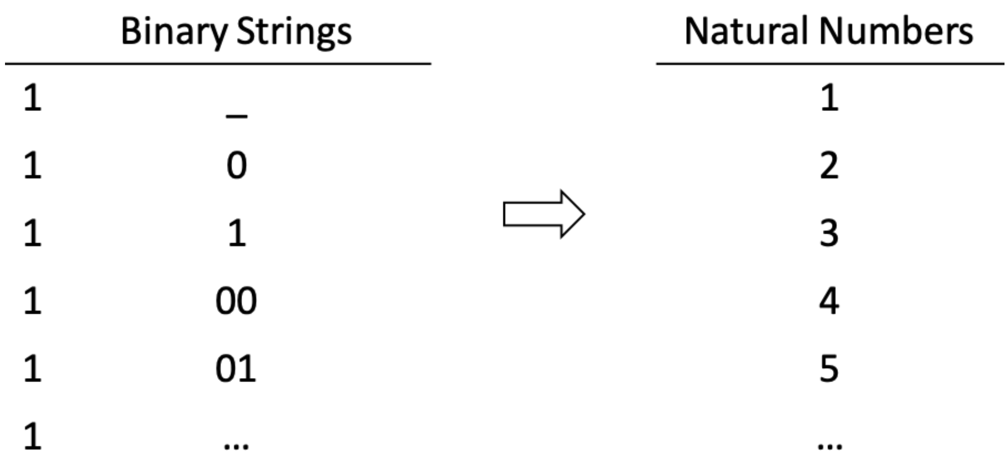

Figure 5.2: Mapping of binary strings to natural number

■

5.2.2 Undecidable

After defining countable set, we can try to answer the question raised at the beginning of this section: is every language decidable by some TM? The answer is no. The reason is that there are uncountably many languages yet only countably many TMs. Since one TM can only recognize one language and there are more languages than TMs, some languages are not decidable by TM.

Corollary 5.3 *Some languages are not decidable by Turing machine.*

Next we will prove that the set of all languages is uncountable and that the set of all TMs are countable. Let's start with the easy one.

Corollary 5.4 *The set of all TMs is countable.*

Proof: This conclusion is obvious. Every Turing machine has an encoding into a string over a finite set of alphabet. We have proven that the set of all strings is countable, therefore the set of TMs is countable. ■

Theorem 5.5 *The set of all languages is uncountable.*

Proof: We prove this theorem by contradiction. Suppose the set of all languages is countable. There must exist a mapping from all strings to natural numbers. In other words, there exists an order of all languages. Next we create a matrix in which we list all strings across the columns, $X_1, X_2, X_3...$, and all languages down the rows, $L_1, L_2, L_3....$ The entries tell whether the language in a given row includes the string in a given column. If string X_i belongs to language L_j, the entry at ith column and jth row is 1. Otherwise it is 0.

Let's define a language L: $X_i \in L \; iff \; X_j \notin L_j$. Based on the assumption, this language must occur in the list $L_1, L_2, L_3...$ with some index k. In the matrix we just defined, it is a row in which all entries is the opposite of diagonal entries. The question is, what should be the entry at the kth row and the kth column when the string X_k belongs to L_k if and only if it does not belongs to L_k? This is the contradiction. Therefore, our assumption that the set of all languages is countable is wrong.

	X_1	X_2	X_3	...	X_k	...
L_1	1	0	1			
L_2	1	0	0			
L_3	0	0	1			
...				...		
L_k	0	1	0	...	?	
...						

Figure 5.3: A contradiction occurs at '?'

■

The approach we used is called the **diagonalization method**. It is a useful method to prove the undecidability of a language.

5.3 Reference

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