CS310 : Automata Theory 2019

Lecture 32: Post's Correspondance Problem

Instructor: S. Akshay

IITB, India

26-03-2019

Recap

Turing machines and computability

1. Definition of Turing machines: high level and low-level descriptions
2. Variants of Turing machines
3. Decidable and Turing recognizable languages
4. Church-Turing Hypothesis
5. Undecidability and a proof technique by diagonalization
   - A universal TM lang $L^A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$
6. Reductions: a powerful way to show undecidability.
7. Rice's theorem, its proof and its applications.

Undecidability beyond Turing machines

Are only problems about Turing machines undecidable?

Undecidability beyond Turing machines

Are only problems about Turing machines undecidable?

- Computers, C-programs, counter machines
- Problems on CFLs: Given CFG G, is $L(G) = \Sigma^*$?
- Problems on Tiling
- Problems on String Matching

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \ldots s_n\}$ and $B = \{t_1, \ldots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \dots s_n\}$ and $B = \{t_1, \dots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \dots, i_m \leq n$ such that

$$s_{i_1} \dots s_{i_m} = t_{i_1} \dots t_{i_m}$$

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \dots s_n\}$ and $B = \{t_1, \dots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \dots, i_m \leq n$ such that

$$s_{i_1} \dots s_{i_m} = t_{i_1} \dots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \dots s_n\}$ and $B = \{t_1, \dots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \dots, i_m \leq n$ such that

$$s_{i_1} \dots s_{i_m} = t_{i_1} \dots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$ 2, 3, 1!

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \ldots s_n\}$ and $B = \{t_1, \ldots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \ldots, i_m \leq n$ such that

$$s_{i_1} \ldots s_{i_m} = t_{i_1} \ldots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$ 2, 3, 1!
- $A = \{0011, 11, 1101\}$ and $B = \{101, 1, 110\}$

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \ldots s_n\}$ and $B = \{t_1, \ldots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \ldots, i_m \leq n$ such that

$$s_{i_1} \ldots s_{i_m} = t_{i_1} \ldots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$ 2, 3, 1!
- $A = \{0011, 11, 1101\}$ and $B = \{101, 1, 110\}$
- $A = \{0011, 11, 1101\}$ and $B = \{101, 011, 110\}$

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \ldots s_n\}$ and $B = \{t_1, \ldots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \ldots, i_m \leq n$ such that

$$s_{i_1} \ldots s_{i_m} = t_{i_1} \ldots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$ 2, 3, 1!
- $A = \{0011, 11, 1101\}$ and $B = \{101, 1, 110\}$
- $A = \{0011, 11, 1101\}$ and $B = \{101, 011, 110\}$
- $A = \{100, 0, 1\}$ and $B = \{1, 100, 0\}$

A simple programming exercise

A string matching problem

Given two lists $A = \{s_1, \dots s_n\}$ and $B = \{t_1, \dots, t_n\}$, over the same alphabet, is there a sequence of combining elements that produces the same string in both lists?

- Does there exist a finite sequence $1 \leq i_1, \dots, i_m \leq n$ such that

$$s_{i_1} \dots s_{i_m} = t_{i_1} \dots t_{i_m}$$

Consider the following lists

- $A = \{110, 0011, 0110\}$ and $B = \{110110, 00, 110\}$ 2, 3, 1!
- $A = \{0011, 11, 1101\}$ and $B = \{101, 1, 110\}$
- $A = \{0011, 11, 1101\}$ and $B = \{101, 011, 110\}$
- $A = \{100, 0, 1\}$ and $B = \{1, 100, 0\}$

Can you write an algorithm for solving this?

Post's correspondance problem

A domino game

Given a collection of dominos:

$$\left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]$$

Post's correspondance problem

A domino game

Given a collection of dominos:

$$\left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]$$

- A match is a list of these dominos (with possible repetitions) such that the string formed in top is same as string formed by bottom row.

Post's correspondance problem

A domino game

Given a collection of dominos:

$$\left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]$$

- A match is a list of these dominos (with possible repetitions) such that the string formed in top is same as string formed by bottom row.
- Does this collection of dominos have a match?

Post's correspondance problem

A domino game

Given a collection of dominos:

$$\left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]$$

- A match is a list of these dominos (with possible repetitions) such that the string formed in top is same as string formed by bottom row.
- Does this collection of dominos have a match?
- Same as the string matching exercise!
- Called Post's Correspondance Problem or PCP.

Post's correspondance problem

A domino game

Given a collection of dominos:

$$\left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]$$

- A match is a list of these dominos (with possible repetitions) such that the string formed in top is same as string formed by bottom row.
- Does this collection of dominos have a match?
- Same as the string matching exercise!
- Called Post's Correspondance Problem or PCP.

Theorem

The Post's correspondance problem is undecidable!

Post's correspondance problem

Theorem

The Post's correspondance problem is undecidable.

Proof Idea:

- Encode TM computation histories!

Post's correspondance problem

Theorem

The Post's correspondance problem is undecidable.

Proof Idea:

- Encode TM computation histories!
- Each transition as a domino!

Post's correspondance problem

Theorem

The Post's correspondance problem is undecidable.

Proof Idea:

- Encode TM computation histories!
- Each transition as a domino!
- Simulate the run using the dominos.

Proof of undecidability of PCP:1

Simplifying assumptions

- Assume that the tape of TM is one-way infinite and never attempts to move left off its left-end.

Proof of undecidability of PCP:1

Simplifying assumptions

- Assume that the tape of TM is one-way infinite and never attempts to move left off its left-end.
- If $w = \varepsilon$, then use $\sqcup$ instead of w.

Proof of undecidability of PCP:1

Simplifying assumptions

- Assume that the tape of TM is one-way infinite and never attempts to move left off its left-end.
- If $w = \varepsilon$, then use $\sqcup$ instead of w.
- Modify PCP so that match must start with a given domino, say the first one. Call this MPCP.

Proof of undecidability of PCP:2

We define a reduction from A_{TM} to $(M)PCP$. Let an instance of A_{TM} be

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
- $w = w_1, \ldots w_n$.

Proof of undecidability of PCP:2

We define a reduction from A_{TM} to $(M)PCP$. Let an instance of A_{TM} be

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
- $w = w_1, \ldots w_n$.

We build instance P' of MPCP in several steps:

Proof of undecidability of PCP:2

We define a reduction from A_{TM} to $(M)PCP$. Let an instance of A_{TM} be

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
- $w = w_1, \ldots w_n$.

We build instance P' of MPCP in several steps:

Step 1: fix first domino in P'

$$\left[\frac{\#}{\# q_0 w_1 \cdots w_n \#}\right]$$

Proof of undecidability of PCP:2

We define a reduction from A_{TM} to $(M)PCP$. Let an instance of A_{TM} be

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
- $w = w_1, \ldots w_n$.

We build instance P' of MPCP in several steps:

Step 1: fix first domino in P'

$$\left[\frac{\#}{\# q_0 w_1 \cdots w_n \#}\right]$$

Because we are reducing to MPCP, the match must start with this domino!

Proof of undecidability of PCP:2

We define a reduction from A_{TM} to $(M)PCP$. Let an instance of A_{TM} be

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
- $w = w_1, \ldots w_n$.

We build instance P' of MPCP in several steps:

Step 1: fix first domino in P'

$$\left[\frac{\#}{\# q_0 w_1 \cdots w_n \#}\right]$$

Because we are reducing to MPCP, the match must start with this domino!How do we proceed?

Proof of undecidability of PCP:3

Step 2: encode transitions of TM into dominos!

Proof of undecidability of PCP:3

Step 2: encode transitions of TM into dominos!

For every $a, b, c \in \Gamma$ and every $q, q' \in Q$, $q \neq q_{rej}$,

- if $\delta(q, a) = (q', b, R)$ then add domino to P':

$$\left[\frac{qa}{bq'}\right]$$

Proof of undecidability of PCP:3

Step 2: encode transitions of TM into dominos!

For every $a, b, c \in \Gamma$ and every $q, q' \in Q$, $q \neq q_{rej}$,

- if $\delta(q, a) = (q', b, R)$ then add domino to P':

$$\left[\frac{qa}{bq'}\right]$$

- if $\delta(q, a) = (q', b, L)$ then add domino to P':

$$\left[\frac{cqa}{q'cb}\right]$$

Proof of undecidability of PCP:3

Step 2: encode transitions of TM into dominos!

For every $a, b, c \in \Gamma$ and every $q, q' \in Q$, $q \neq q_{rej}$,

- if $\delta(q, a) = (q', b, R)$ then add domino to P':

$$\left[\frac{qa}{bq'}\right]$$

- if $\delta(q, a) = (q', b, L)$ then add domino to P':

$$\left[\frac{cqa}{q'cb}\right]$$

- add all dominos (i.e, for all $a \in \Gamma \cup \{\#\}$) to P':

$$\left[\frac{a}{a}\right]$$