Complexity

Lecture 35
Sections 14.1 - 14.2

Robb T. Koether

Hampden-Sydney College

Mon, Nov 21, 2016

Outline

The Post Correspondence Problem

Definition (The Post Correspondence Problem)

Given two sets of n strings over an alphabet Σ,

$$\{w_1, w_2, w_3, \ldots, w_n\}$$

and

$$\{v_1, v_2, v_3, \ldots, v_n\},$$

is it possible to satisfy the equation

$$w_i w_j w_k \cdots w_m = v_i v_j v_k \cdots v_m,$$

where each w_i and v_i is any string from the respective sets? Repetitions are allowed and not every string need be used, but they must be indexed in the same order.

The Post Correspondence Problem

Theorem

The Post Correspondence Problem is undecidable.

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

  and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

- Can it be done?

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

  and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

- Can it be done?
- If so, then we must begin with (w_2, v_2) or (w_5, v_5).

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

  and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

- Can it be done?
- If so, then we must begin with (w_2, v_2) or (w_5, v_5). Why?

The Post Correspondence Problem

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

- Can it be done?
- If so, then we must begin with (w_2, v_2) or (w_5, v_5). Why?
- And we must end with (w_3, v_3), (w_4, v_4), or (w_6, v_6).

Example

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and let the sets be

$$\{\mathbf{ba}, \mathbf{c}, \mathbf{cb}, \mathbf{b}, \mathbf{a}, \mathbf{ba}, \mathbf{ac}\}$$

and

$$\{\mathbf{c}, \mathbf{ca}, \mathbf{bb}, \mathbf{cb}, \mathbf{ac}, \mathbf{a}, \mathbf{b}\}.$$

ba	**c**	**cb**	**b**	**a**	**ba**	**ac**
c	**ca**	**bb**	**cb**	**ac**	**a**	**b**

- Can it be done?
- If so, then we must begin with (w_2, v_2) or (w_5, v_5). Why?
- And we must end with (w_3, v_3), (w_4, v_4), or (w_6, v_6). Why?

The Post Correspondence Problem

Theorem

The Post Correspondence Problem is undecidable.

The Post Correspondence Problem

Theorem

The Post Correspondence Problem is undecidable.

- The Post Correspondence Problem (PCP) can be reduced to many other decision problems.
- Thus, the undecidability of PCP implies the undecidability of many other problems.

Outline

Time and Space Complexity

- There are two basic ways to measure complexity.
  - Time complexity - How much time does a program require?
  - Space complexity - How much memory does a program require?

Time and Space Complexity

- There are two basic ways to measure complexity.
  - Time complexity - How much time does a program require?
  - Space complexity - How much memory does a program require?
- For a Turing machine,
  - Time complexity is measured by the number of transitions executed.
  - Space complexity is measured by the number of tape cells required.

Time and Space Complexity

- There are two basic ways to measure complexity.
  - Time complexity - How much time does a program require?
  - Space complexity - How much memory does a program require?
- For a Turing machine,
  - Time complexity is measured by the number of transitions executed.
  - Space complexity is measured by the number of tape cells required.
- We will consider only time complexity.

Time Complexity

- Typically, the number of transitions required by a Turing machine depends on the input.
- We are interested in the time complexity as a function of the size (length) of the input.
- If n is the size of the input, then we seek a function $T(n)$ for the time complexity.

Time Complexity

- Typically, the number of transitions required by a Turing machine depends on the input.
- We are interested in the time complexity as a function of the size (length) of the input.
- If n is the size of the input, then we seek a function $T(n)$ for the time complexity.
- But even for inputs of the same length, the times could be different.

Time Complexity

- Typically, the number of transitions required by a Turing machine depends on the input.
- We are interested in the time complexity as a function of the size (length) of the input.
- If n is the size of the input, then we seek a function $T(n)$ for the time complexity.
- But even for inputs of the same length, the times could be different.
- We define $T(n)$ to be the worst case (maximal time) for all inputs of length n.

Example (The Turing Machine INCR)

- Recall the Turing machine INCR that incremented the input.
- n is the number of bits in the number.
- n transitions are needed to reach the right end of the number.
- At most, n transitions are needed to change 1's to 0's and then a 0 to 1.
- The worst case is when all the bits are 1. That case requires one additional transition, to write a 1 at the left end of the string of 0's.
- Thus, $T(n) = 2n + 1$.

O, Ω, and Θ

- In general, it is too tedious, and often not possible, and not really necessary to compute exactly the function $T(n)$,
- Our primary concern is not the exact value of $T(n)$, but

  how fast $T(n)$ increases as n increases.
- Thus, it is enough to be able to say that $T(n) \in O(f(n))$ or $T(n) \in \Theta(f(n))$ for some known function $f(n)$.

O, Ω, and Θ

Definition (O (At least as fast as...))

The function $T(n) \in O(f(n))$ for some function $f(n)$ if there exists a constant c such that $T(n) \leq cf(n)$ for all $n \geq n_0$ for some n_0.

O, Ω, and Θ

Definition (O (At least as fast as...))

The function $T(n) \in O(f(n))$ for some function $f(n)$ if there exists a constant c such that $T(n) \leq cf(n)$ for all $n \geq n_0$ for some n_0.

Definition (Ω (At least as slow as...))

The function $T(n) \in \Omega(f(n))$ for some function $f(n)$ if there exists a constant c such that $T(n) \geq cf(n)$ for all $n \geq n_0$ for some n_0.

O, Ω, and Θ

Definition (O (At least as fast as...))

The function $T(n) \in O(f(n))$ for some function $f(n)$ if there exists a constant c such that $T(n) \leq cf(n)$ for all $n \geq n_0$ for some n_0.

Definition (Ω (At least as slow as...))

The function $T(n) \in \Omega(f(n))$ for some function $f(n)$ if there exists a constant c such that $T(n) \geq cf(n)$ for all $n \geq n_0$ for some n_0.

Definition (Θ (Just as fast as...))

The function $T(n) \in \Theta(f(n))$ for some function $f(n)$ if there exists a constants c_1 and c_2 such that $c_1 f(n) \leq T(n) \leq c_2 f(n)$ for all $n \geq n_0$ for some n_0.

Outline

The Satisfiability Problem

Definition (The Satisfiability Problem)

Given a boolean expression *e* in Conjunctive Normal Form, the Satisfiability Problem (SAT) asks whether *e* is true for some choice of boolean values of its variables, i.e, is *e* “satisfiable?”

Conjunctive Normal Form

Definition (Conjunctive Normal Form)

A boolean expression e is in conjunctive normal form (CNF) if

$$e = t_1 \wedge t_2 \wedge \cdots \wedge t_n$$

where for each term (or clause) t_i,

$$t_i = s_{i1} \vee s_{i2} \vee \cdots \vee s_{im},$$

where each s_{ij} is a boolean variable or its negation.

- Disjunctive normal form (DNF) is similar, with $\wedge$ and $\vee$ reversed.
- Some definitions require that every variable x appear in every clause either as x or as $\overline{x}$.

Conjunctive Normal Form

Example (Examples)

- Let the variables be x_1, x_2, and x_3.
- An example

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_3})$$

- Convert the expression

$$(x_1 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3)$$

  from CNF to DNF.

Converting CNF to DNF

- To convert e from CNF to DNF,
  - Apply DeMorgan's Law to e.

$$\begin{aligned}\overline{e} &= \overline{(x_1 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3)} \\ &= (\overline{x_1} \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \\ &= (\overline{x_1} \wedge x_2 \wedge x_3) \vee (\overline{x_1} \wedge \overline{x_2} \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}).\end{aligned}$$

Converting CNF to DNF

- To convert e from CNF to DNF,
  - Write the truth table for $\overline{e}$ and e.

x_1	x_2	x_3	$\overline{e}$	e
1	1	1	0	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	1	0
0	1	0	0	1
0	0	1	1	0
0	0	0	0	1

Converting CNF to DNF

- To convert e from CNF to DNF,
  - Select the combinations that make e true.

x_1	x_2	x_3	$\overline{e}$	e
1	1	1	0	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	1	0
0	1	0	0	1
0	0	1	1	0
0	0	0	0	1

Converting CNF to DNF

- To convert e from CNF to DNF,
  - Write e in DNF based on the table.

$$e = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3}) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \\ \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \vee (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}).$$

Converting CNF to DNF

- To convert e from CNF to DNF,
  - Write e in DNF based on the table.

$$e = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3}) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \vee (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}).$$

- The procedure can be reversed to convert DNF to CNF.

Outline

The Satisfiability Problem

Definition (The Satisfiability Problem)

Given a boolean expression e in CNF, the Satisfiability Problem (SAT) asks whether e is satisfiable.

The Satisfiability Problem

- One method to decide the problem is to try "true" and "false" for each of the n variables.
- There are 2^n possible combinations, so the run time is exponential.
- Another method is to convert e from CNF to DNF, at which point the answer is obvious.
- How efficiently can we convert from CNF to DNF?

Outline

Assignment

Homework

- Section 14.1 Exercises 1, 2.
- Section 14.2 Exercises 4, 5.