Review: Elements of Computation Theory

XIAOGANG JIN

AI Institute, College of Computer Science
Zhejiang University
E-Mail: xiaogangj@cise.zju.edu.cn

Outline

Regular Language

- Regular Expression
- Deterministic finite automata(DFA)
- Non-deterministic finite automata(NFA)
- Closure properties and Pumping Theorem
- State Minimization (graduated course)

Regular Language

- Regular Expression
- Deterministic finite automata(DFA)
- Non-deterministic finite automata(NFA)
- Closure properties and Pumping Theorem
- State Minimization (graduated course)

Question:

- Give a DFA or NFA $\Rightarrow$ Regular Expression (Example 2.3.2 P 81);
- Give a Regular Expression $\Rightarrow$ DFA or NFA (Theorem 2.3.1 P 75);
- Show a given language be regular or non-regular?
  (Yes, regular expression, DFA, NFA and closure property; No, Pumping theorem or closure property)

2.4.8 Are the following statements true od false? Explain you answer in each case. (P 91)

- Every subset of a regular language is regular. (**F**)
- Every regular language has a regular proper subset. (**F**) (Hint: $\emptyset$)
- If L is regular, then so is $\{xy \mid x \in L \text{ and } y \notin L\}$. (**T**) (Hint:$\{xy \mid x \in L \text{ and } y \notin L\}$=$L \circ \overline{L}$)
- $\{w \mid w = w^R\}$ is regular. (**F**)
- If L is regular, then so is $\{w \mid w \in L \text{ and } w^R \in L\}$. (**T**) (Hint: $L = L \cap L^R$.)
- If C is any set of regular languages, then $\cup C$ is a regular language. (**F**)
- $\{xyx^R \mid x, y \in \Sigma^*\}$ is regular. (**T**) (Hint: $\{xyx^R \mid x, y \in \Sigma^*\} = \Sigma^*$. By letting $x = e$, y can vary over all the strings of Σ^*.)

Outline

1. Regular Language
2. Context-free Language(CFL)
3. Turing Machine and Recursive Enumerable Language
4. Undecidablity
5. $\mathcal{P}$ and $\mathcal{NP}$ Problems
6. $\mathcal{NP}-$ Complete Problems

Context-free Language

- Context-free Grammar(CFG)
- Pushdown automata
- Closure properties and Pumping Theorem

Context-free Language

- Context-free Grammar(CFG)
- Pushdown automata
- Closure properties and Pumping Theorem

Question:

- Give a context-free language $\Rightarrow$ Context-free Grammar
- Give a context-free language $\Rightarrow$ PDA
- Give a context-free grammar $\Rightarrow$ PDA (Lemma 3.4.1, Example 3.4.1 P 136)
- Show a given language be context-free or non-context free?

Example: Show that language $L = \{a^i b^j c^k \mid j \geq i + k\}$ is context-free.

Solution:

□ $\{a^i b^j c^k \mid j \geq i + k\} = a^i b^i \circ b^* \circ b^k c^k$

□ Language L can be generated by the following CFG:

$S \to XYZ$, $X \to aXb$, $Y \to bY$, $Y \to bYc$, $X \to e$, $Y \to e$, $Z \to e$

3.1.9 Show that the following language are context-free by exhibiting context-free grammars generating each (P 122):

- $\{a^m b^n \mid m \geq n\}$
  $\{S \rightarrow aSb, S \rightarrow aS, S \rightarrow e\}$
- $\{a^m b^n c^p d^q \mid m + n = p + q\}$
  Let $m + n = p + q = N$, then $n = N - m$, $p = N - q$.
  $a^m b^n c^p d^q = a^m b^{N-m} c^{N-q} d^q$
  In case of $m \geq q$, $a^m b^n c^p d^q = a^q a^{m-q} b^{N-m} c^{N-m} c^{m-q} d^q$
  $S \rightarrow aSd$, $S \rightarrow A$, $A \rightarrow aAc$, $A \rightarrow B$, $B \rightarrow bBc$, $B \rightarrow e$
  In case of $m < q$,we can obtain the similar results.
- $\{w \in \{a, b\}^* : w$ has twice as many b's as a's $\}$
  $\{S \rightarrow e$, $S \rightarrow Sabb \mid aSbb \mid abSb \mid abbS$,
  $S \rightarrow Sbab \mid bSab \mid baSb \mid babS$, $S \rightarrow Sbba \mid bSba \mid bbSa \mid bbaS$ $\}$
- $\{uawb : u, w \in \{a, b\}^* |u| = |w|\}$
  $\{S \rightarrow Tb, T \rightarrow aTa, T \rightarrow aTb, T \rightarrow bTa, T \rightarrow bTb, T \rightarrow a\}$

3.5.1 Use closure under union to show that the following language are context-free (P 148).
a) $\{a^m b^n \mid m \neq n\}$
$\{a^m b^n \mid m \neq n\} = \{a^m b^n \mid m > n\} \cup \{a^m b^n \mid m < n\}$
or $\{a^m b^n \mid m > n\} = a \circ a^* \circ \{a^m b^n \mid m = n\}$

b) $\{a, b\}^* - \{a^n b^n : n \geq 0\}$
This language cab be expressed as
$\{a^m b^n \mid m \neq n\} \cup \Sigma^* a \Sigma^* b \Sigma^* a \Sigma^* \cup \Sigma^* b \Sigma^* a \Sigma^* b \Sigma^*$

3.5.8 Show that the language $\{ww \mid w \in \{a,b\}^*\}$ is not context-free.

Proof: Let $L = \{ww \mid w \in \{a,b\}^*\}$, and $s = a^p b^p a^p b^p$, $s \in L$. p is the pumping length. Use the Pumping Theorem to prove that. If L is CFL, such that $s = uvxyz$, $|vxy| \leq p$.

(1) Consider that the substring vxy of s is over the midpoint of s. Pump the s as uxz, the string s is as $a^p b^i a^j b^p$ form, where i and j are not equal to p at the same time. Such that the s is not the form ww.

(2) If vxy is placed before the midpoint of s, by the Pumping Theorem, when $s = uv^2xy^2z$, the b has to be put the first place of the last part of s after the midpoint, such that the s is not the form ww. Similarly, if vxy is placed after the midpoint of s, when $s = uv^2xy^2z$, the a has to be moved to the last place of the first part of s before the midpoint, also the s is not the form ww. So, L is not CFL .

3.5.14 Which of the following languages are context-free? Explain briefly in each case.

a) $\{a^m b^n c^p \mid m = n \text{ or } n = p \text{ or } m = p\}$

This language is context-free.

$\{a^m b^n c^p \mid m = n \text{ or } n = p \text{ or } m = p\}$

$= \{a^m b^n c^p \mid m = n\} \cup \{a^m b^n c^p \mid n = p\} \cup \{a^m b^n c^p \mid m = p\}$

and

$\{a^m b^n c^p \mid m = n\} = \{a^n b^n\} \circ c^*$

$\{a^m b^n c^p \mid n = p\} = a^* \circ \{b^n c^n\}$

$\{a^m b^n c^p \mid m = p\}$ can be generated by the following CFG:

$\{S \to aSb, S \to T, T \to cT, T \to e\}$.

b) $\{a^m b^n c^p \mid m \neq n \text{ or } n \neq p \text{ or } m \neq p\}$
This language is context-free.
$\{a^m b^n c^p \mid m \neq n \text{ or } n \neq p \text{ or } m \neq p\}$
$= \{a^m b^n c^p \mid m \neq n\} \cup \{a^m b^n c^p \mid n \neq p\} \cup \{a^m b^n c^p \mid m \neq p\}$
and
$\{a^m b^n c^p \mid m \neq n\} = \{a^m b^n c^p \mid m > n\} \cup \{a^m b^n c^p \mid m < n\}$

c) $\{a^m b^n c^p \mid m = n \text{ and } n = p \text{ and } m = p\}$ This language is not context-free.
$\{a^m b^n c^p \mid m = n \text{ and } n = p \text{ and } m = p\}$=$\{a^n b^n c^n \mid n \geq 0\}$ (Pump Theorem).

d) $L = \{w \in \{a, b, c\}^* : w$ does not contain equal numbers of occurrences of a, b and $c\}$
This language is context-free.
$L = \{w \in \{a, b, c\}^* : w$ has different numbers of a's and b's$\}$
$\cup \{w \in \{a, b, c\}^* : w$ has different numbers of b's and c's$\}$
$\cup \{w \in \{a, b, c\}^* : w$ has different numbers of a's and c's$\}$.
and the language
$\{w \in \{a, b, c\}^* : w$ has different numbers of a's and b's$\}$ is context-free.

3.5.15 Suppose that L is context-free and R is regular. Is $L - R$ necessarilly context-free? What about $R - L$? Justify your answers.
(1) $L - R$ is context-free.
$L - R = L \cap \bar{R}$ and Theorem 3.5.2 (P 144)
(2) we can not conclude that $R - L$ is context-free.
$R = a^*b^*c^*$, $L = \{a^m b^n c^p \mid m \neq n$ or $n \neq p$ or $m \neq p\}$ then
$R - L = \{a^n b^n c^n\}$ is not context-free.

Outline

TM and R.E. Language

- Turing Machine
- Grammar
- Numerical Functions
  Basic Functions, composition,function defined recursively; primitive recursive functions,primitive recursive predicate; minimalizable, μ*-recursive*

TM and R.E. Language

- Turing Machine
- Grammar
- Numerical Functions
  Basic Functions, composition,function defined recursively; primitive recursive functions,primitive recursive predicate; minimalizable, μ-recursive

Question:

a) design a Turing Machine to compute a function or decide (semidecide) a language;
b) Given a TM $\rightarrow$ function;
c) Show a function be a primitive recursive function.

Outline

Undecidablity

- Church-Turing Thesis
- Chomsky hierarchy
- Universal Turing Machine
- Halting Problem
- Some Undecidable problems

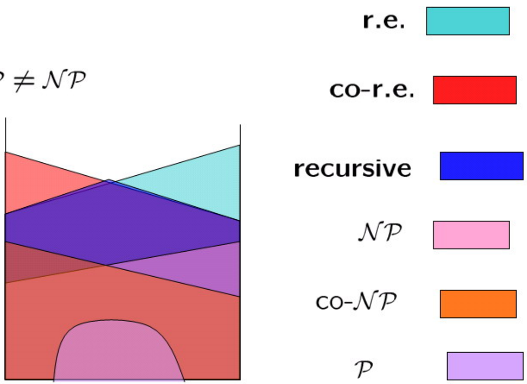
Undecidablity

- Church-Turing Thesis
- Chomsky hierarchy
- Universal Turing Machine
- Halting Problem
- Some Undecidable problems

Question:

Establish reduction between some undecidable problems

I Asumming $\mathcal{P} \neq \mathcal{NP}$



Example: (Halting Problem) Show that it is undecidable whether a TM halts on all inputs.

Solution: Consider some TM M with input w. We can then create a second TM M_0 that works as follows:
Given any input w_0, M_0 ignores w_0 and simulates running M on w (M is hard-coded into M_0). M_0 halts on all inputs M halts on w. We know that the problem of determining whether a TM halts on a single input is undecidable and thus, in general, we cannot decide whether M halts on w. It follows that it is undecidable whether M_0 halts on all inputs.

Outline

$\mathcal{P}$ and $\mathcal{NP}$ Problems

- $\mathcal{P}$ Problems

$$\mathcal{P} = \bigcup_{p} \mathrm{TIME}(p), \text{ where } p \text{ is a polynomial}$$

- $\mathcal{NP}$ Problems

$$\mathrm{NTIME}(t) = \{L : L \text{ is decided in time } t \text{ by some NTM}\}$$

$$\mathcal{NP} = \bigcup_{p} \mathrm{NTIME}(p), \text{ where } p \text{ is a polynomial}$$

$\mathcal{P}$ and $\mathcal{NP}$ Problems

- $\mathcal{P}$ Problems

$$\mathcal{P} = \bigcup_p \mathrm{TIME}(p), \text{ where } p \text{ is a polynomial}$$

- $\mathcal{NP}$ Problems

$$\mathrm{NTIME}(t) = \{L : L \text{ is decided in time } t \text{ by some NTM}\}$$

$$\mathcal{NP} = \bigcup_p \mathrm{NTIME}(p), \text{ where } p \text{ is a polynomial}$$

Question:

Show a given language be $\mathcal{P}$ Problem or $\mathcal{NP}$ Problem

Outline

- $\mathcal{NP}-$ Complete Problems

- $\mathcal{NP}-$ Complete Problems

Question:

Establish reduction between some $\mathcal{NP}-$ Complete Problems