NOTES ON LANGUAGES

MIKE ZABROCKI

1. Introduction

The symbol Σ will represent an alphabet or a set of letters. 'Languages' are sets of words (i.e. subsets of $\Sigma^* = \{w_1 w_2 \cdots w_k | w_i \in \Sigma\}$.

Let A be a language and set $a_n = |\{w | w \in A, |w| = n\}|$

The general question that we want to consider is "Under what conditions can we say something about the sequence $a_0, a_1, a_2, \ldots$?"

If A is a language then set $G_A(q) = \sum_{n \geq 0} a_n q^n$ is the generating function for A.

Exercise 1. *Find $G_A(q)$ and a_n where A is the language Σ^* with $|\Sigma| = n$.*

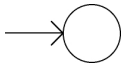
Is it always possible to give a formula for a_k or $G_A(q)$? Given a description of a language A is it possible to find a formula for a_n or $G_A(q)$? Can we at least compute a_n in a "reasonable" amount of time for a given n?

2. Regular languages

Finite automata or Deterministic Finite Automata (DFA) - $M = (Q, \Sigma, \delta, q_0, F)$ where

(1) Q - set of states
(2) Σ - alphabet
(3) $\delta : Q \times \Sigma \to Q$ - transition function
(4) q_0 - start state (an element of Q)
(5) F - a subset of Q called the accept states (or final states)

Usually a finite state automata is represented as a graph. A start state is represented as



an accept state is indicated by

Example 2. *Insert example of DFA here.*

An example of a FA.

We say that M accepts $w = w_1 \cdots w_n$ if there exists a sequence of states $r_0, r_1, \ldots, r_n$ satisfying

(1) $r_0 = q_0$
(2) $\delta(r_i, w_{i+1}) = r_{i+1}$
(3) $r_n \in F$.

The language that M recognizes is $\{w | M \text{ accepts } w\}$.

The empty string is denoted by ε. Machines that have $q_0 \in F$ accept ε.

A language recognized by a finite automata is called *regular.*

Regular languages are closed under

- union - $A \cup B$
- concatenation - $A \circ B = \{wv | w \in A, v \in B\}$
- star - $A^* = \{w^{(1)} \cdots w^{(k)} \mid w^{(i)} \in A\}$
- intersection
- complementation
- set difference

Define an *NFA* (Non-deterministic Finite Automata) to be a finite automata where we relax the condition that the transition function now

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$$

where $\mathcal{P}(Q)$ represents the power set of Q.

An NFA accepts a word w if there is a sequence of states $r_0, \cdots, r_m$ such that

(1) $r_0 = q_0$
(2) $r_{i+1} \in \delta(r_i, w_{i+1})$ or $r_{i+1} \in \delta(r_i, \varepsilon)$
(3) $r_m \in F$.

The language that an NFA accepts is $\{w | M \text{ accepts } w\}$.

Exercise 3. *Insert NFA here*

Which of the following words are accepted by the NFA above?: $\varepsilon, a, b, bb, baa, baba, abba, babba$.

Looking at the conditions closely, one would assume that the class of languages which is accepted by NFAs is larger than regular languages. It turns out that they are the same. That is,

Theorem 4. *If A is a regular language, then A is accepted by an NFA. Conversely, if A is the language that is accepted by an NFA then A is regular.*

That is, for every NFA there is a DFA which accepts the same language. The proof of this theorem is constructive in the sense it defines a method for converting an NFA to a DFA.

Exercise 5. *Find a DFA which is equivalent to the NFA from the previous exercise.*

Given this, it is easy to see why regular languages are closed under union, concatenation and star operations.

Exercise 6. *Give an outline of a proof that $A \cup B$ is regular given that A and B are regular by defining an NFA which accepts $A \cup B$ in terms of the DFA which accepts A and B.*

Exercise 7. *Give an outline of a proof that $A \circ B$ is regular given that A and B are regular by defining an NFA which accepts $A \circ B$ in terms of the DFA which accepts A and B.*

Exercise 8. *Give an outline of a proof that A^* is regular given that A are regular by defining an NFA which accepts A^* in terms of the DFA which accepts A.*

Regular languages are also represented by *regular expressions*.

Definition 9. *An expression R is regular if it is*

(1) $\emptyset$
(2) ε
(3) $a \in \Sigma$
(4) $R_1 \cup R_2$ *or sometimes* $R_1 + R_2$
(5) $R_1 \circ R_2$ *or sometimes* $R_1 R_2$
(6) R_1^* *where R_1, R_2 are themselves regular expressions.*

Exercise 10. *Find a regular expression which represents the DFA of example 2.*

Exercise 11. *Create a DFA which represents the language accepted by the following regular expression.*

$$a^* \cup (a^* b(b \cup a(aa^* b)^* b)a)^*$$

Of course, not all languages are regular.

$$C = \{w | w \text{ has an equal number of } 0s \text{ and } 1s\}$$
$$D = \{w | w \text{ has an equal number of 01 and 10 substrings}\}$$

C is not regular, but D is!

Lemma 12. *(The Pumping Lemma for regular languages) If A is regular, then there is a $p \geq 0$ such that if $w \in A$ and $|w| \geq p$ then $w = xyz$ where*

(1) $xy^i z \in A$
(2) $|y| > 0$
(3) $|xy| < p$

If A does not have this property then A is not regular.

Exercise 13. *Use the pumping lemma to show that C (given above) is not regular.*

Exercise 14. *Find a DFA which accepts the language of D.*

Exercise 15. *The following is a short list of examples of regular languages*

(1) $\{w \in \{0,1\}^* \,|\, w \text{ has a single } 1\}$
(2) $\{w \in \{0,1\}^* \,|\, w \text{ has a at least one } 1\}$
(3) $\{w \in \{0,1\}^* \,|\, w \text{ contains } 001 \text{ as a substring}\}$
(4) $\{w \in \Sigma^* \,|\, |w| = 2n, n \geq 0\}$
(5) $\{w \in \Sigma^* \,|\, |w| = 3n, n \geq 0\}$
(6) $\{w \in \Sigma^* \,|\, |w| \leq 4\}$
(7) $\{w \in \Sigma^* \,|\, |w| = 4\}$
(8) $\{w \in \Sigma^* \,|\, w \text{ starts and ends with the same symbol}\}$

Find a regular expression representing each of the languages above.

3. Context free languages

A *context free grammar* is a tuple $G = (V, \Sigma, R, S)$ where

(1) V is a finite set of variables
(2) Σ is the alphabet, usually called terminals ($\Sigma \cap V = \emptyset$)
(3) R a finite set of rules
(4) S a start symbol

A *rule* here is a variable and a string of variables and terminals (including the empty string ε).

We say that u yields w or $u \Rightarrow w$ if $u = xAy$ and $w = xzy$ and $A \to z$ is a rule. we write $u \overset{*}{\Rightarrow} w$ if there is a sequence of u_i such that $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow w$.

The language of the grammar is $\{w \in \Sigma^* | S \overset{*}{\Rightarrow} w\}$. A language is called context free if there is a context free grammar which accepts the language.

A derivation of a string w in a grammar G is called a leftmost derivation if at each step the leftmost remaining variable is the one replaced.

Example 16. *If $A = \{0^n 1^n : n \geq 0\}$ then a grammar which accepts A as the language is $G = (\{S\}, \{0,1\}, \{S \to 0S1 \,|\, \varepsilon\}, S)$.*

Example 17. *If $A = \{0^n 1^n : n \geq 0\} \cup \{1^n 0^n : n \geq 0\}$ then a grammar which accepts A as the language is $G = (\{S, S_1, S_2\}, \{0,1\}, R, S)$ where R is given as*

$$S \to S_1 \,|\, S_2$$
$$S_1 \to 0S_1 1 \,|\, \varepsilon$$
$$S_2 \to 1S_2 0 \,|\, \varepsilon$$

Exercise 18. *Prove regular languages are context free.*

Exercise 19. *The following is a short list of languages which are context free but not regular. For each give an unambiguous context free grammar which recognizes the language.*

- $\{a^n b^n : n \geq 0\}$
- $\{w \,|\, w \in \{a,b\}^*, w \text{ is a palindrome}\}$
- *Dyck words* - $\{w \in \{(,)\}^* \,|\, w \text{ has balanced parentheses}\}$.
- $\{w \in \{0,1\}^* \,|\, |w| = 2n+1 \text{ and the middle symbol is } 0\}$
- $\{w \in \{0,1\}^* \,|\, w \text{ contains the same number of } 1s \text{ as } 0s\}$
- $\{w \in \{0,1\}^* \,|\, w \text{ contains more } 1s \text{ than } 0s\}$
- $\{w \in \{a,b\}^* \,|\, w \neq a^n b^n\}$
- $\{w \in \{a,b\}^* \,|\, |w|_a = 2|w|_b\}$

A grammar is ambiguous if there are two or more leftmost derivations of the same word.

Sometimes ambiguous grammars have an unambiguous counterpart which generates the same language. Some context free languages only have ambiguous grammars and are called *inherently ambiguous*.

Example 20. *The following is a long list of languages which are context free but inherently ambiguous (see* [2]*).*

- $\{a^i b^j c^k \,|\, i,j,k \geq 0 \text{ and either } i = j \text{ or } j = k\}$
- $\{w \,|\, w \in \{a,b,c\}^*, |w|_a = |w|_b \text{ or } |w|_a = |w|_c\}$
- $\{w \,|\, w \in \{x, \bar{x}, y, \bar{y}\}^*, |w|_x = |w|_{\bar{x}} \text{ or } |w|_y = |w|_{\bar{y}}\}$
- $\{w \,|\, w \in \{a,b,c\}^*, |w|_a \neq |w|_b \text{ or } |w|_a \neq |w|_c\}$
- $\{w_1 w_2 \,|\, w_1, w_2 \in \{a,b\}^*; w_1 = rev(w_1), w_2 = rev(w_2)\}$
- $\{a^n b v_1 a^n v_2 \,|\, n \geq 1, v_1, v_2 \in \{a,b\}^*\}$
- $\{a^{n_1} b a^{n_2} b \cdots a^{n_{2k}} b \,|\, (\forall j, n_{2j} = n_{2j-1}) \text{ or } (n_1 = n_{2k} \text{ and } \forall j, n_{2j} = n_{2j+1})\}$
- $\{a^{n_1} b a^{n_2} b \cdots a^{n_{2k}} b \,|\, (n_1 = 1, \forall j, n_{2j} = 2n_{2j-1}) \text{ or } (\forall j, n_{2j} = 2n_{2j+1})\}$
- $\{a^{n_1} b a^{n_2} b \cdots a^{n_k} b \,|\, \exists j, n_j \neq j\}$

- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, \exists j, n_j < j\}$
- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, \exists j, n_j > j\}$
- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, \exists j, n_j = j\}$
- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, \exists j, n_j \neq k\}$
- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, k \geq 2, \exists j, n_{j+1} \neq n_j\}$
- $\{a^{n_1}ba^{n_2}b\cdots a^{n_k}b \,|\, k \geq 2, \exists j, n_{j+1} \neq 2n_j\}$
- $\{\tilde{n}_1\tilde{n}_2\cdots\tilde{n}_k \,|\, n_1 \neq 1 \text{ or } \exists j, n_{j+1} \neq n_j + 1\}$ *where* $\tilde{n}_i$ *is the binary representation of the non-negative integer* n_i *in* $\{0,1\}^*c$ *where where the* c *is the end of word marker.*
- $\{ucv_1wv_2 \,|\, u, v_1, v_2, w \in \{a,b\}^*, w = rev(u)\}$

Exercise 21. *Given the grammar with rules*

$$< EXPR > \rightarrow < EXPR > + < EXPR > \;|\; < EXPR > \times < EXPR > \;|\; (< EXPR >) \,|\, a \,|\, b,$$

give a word in this language which can be derived ambiguously.

Context free languages are close under

- union
- concatenation
- Kleene star

They are not closed under intersection or complementation. The intersection of a context free language and a regular language is context free.

Lemma 22. *(The pumping lemma for context free languages) if* A *is a context free language, then there is a number* p *where, if* s *is any string in* A *of length at least* p*, then* s *may be divided into five pieces* $s = uvxyz$ *satisfying the conditions:*

(1) *for each* $i \geq 0$*,* $uv^ixy^iz \in A$*,*
(2) $|vy| > 0$*, and*
(3) $|vxy| \leq p$*.*

Exercise 23. *Using the pumping lemma prove that the following languages are not context free.*

- $\{a^nb^nc^n \,|\, n \geq 0\}$
- $\{0^n1^n0^n1^n \,|\, n \geq 0\}$
- $\{0^n10^{2n}10^{3n} \,|\, n \geq 0\}$
- $\{w\#x \,|\, w \text{ is a substring of } x\}$

Exercise 24. *Show that* $L_1 = \{a^ib^ic^k \,|\, i, k \geq 0\}$ *is context free. Let* $L_2 = \{a^ib^kc^k \,|\, i, k \geq 0\}$*. Show that* $L_1 \cap L_2$ *is not context free.*

Context free grammars are equivalent to a class of automata called push-down automata (PDA).

A *pushdown automata* is a six-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where Q, Σ, Γ, and F are all finite sets, and

(1) Q is the set of states

(2) Σ is the input alphabet
(3) Γ is the stack alphabet
(4) $\delta : Q \times (\Sigma \cup \varepsilon) \times (\Gamma \cup \varepsilon) \to \mathcal{P}(Q \times (\Gamma \cup \varepsilon))$ is the transition function,
(5) $q_0 \in Q$ is the start state
(6) $F \subseteq Q$ is the set of accept states.

The language that the automata $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ accepts is the set of words $w = w_1 w_2 \cdots w_m$ where each $w_i \in (\Sigma \cup \varepsilon)$ and sequences $r_0, r_1, \ldots, r_m \in Q$ and strings $s_0, s_1, \ldots, s_m \in \Gamma^*$ exist that satisfies

(1) $r_0 = q_0$ and $s_0 = \varepsilon$
(2) $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, where $s_i = at$ and $s_{i+1} = bt$ for some $a, b \in (\Gamma \cup \varepsilon)$ and $t \in \Gamma^*$.
(3) $r_m \in F$.

4. Indexed Languages

An indexed grammar is a five-tuple (V, Σ, I, R, S) where

(1) V is a set of variables
(2) Σ is the alphabet of terminals
(3) I is the index set (a finite alphabet)
(4) R is a set of rules of the form
    (a) $A \to \alpha$ with $A \in V$ and $\alpha \in (V \cup \Sigma)^*$
    (b) $A \to Bf$ with $A, B \in V$
    (c) $Af \to \alpha$ with $f \in I$ and $\alpha \in (V \cup \Sigma)^*$
(5) S is the start symbol.

Let β and γ be in $(VI^* \cup \Sigma)^*$, $\delta \in I^*$ and $X_i \in V \cup \Sigma$. The productions of this grammar are

(1) $A \to X_1 X_2 \cdots X_k$ is a production of type (1) then
$$\beta A \delta \gamma \Rightarrow \beta X_1 \delta_1 X_2 \delta_2 \cdots X_k \delta_k \gamma$$
where $\delta_i = \delta$ if $X_i \in V$ and $\delta_i = \varepsilon$ if X_i is in Σ.
(2) If $A \to Bf$ is a production of type (2) then $\beta A \delta \gamma \Rightarrow \beta B f \delta \gamma$.
(3) If $Af \to X_1 X_2 \cdots X_k$ is a production of type (3), then
$$\beta A f \delta \gamma \Rightarrow \beta X_1 \delta_1 X_2 \delta_2 \cdots X_k \delta_k \gamma$$
(same convention for δ_i).

The language of the grammar is then $\{w \mid S \Rightarrow w \text{ and } w \in \Sigma^*\}$.

Exercise 25. *$G = (\{S, T, A, B, C\}, \{a, b, c\}, \{f, g\}, R, S)$ where R consists of the following rules:*
$$S \to Tg$$
$$T \to Tf$$

$$T \to ABC$$
$$Af \to aA$$
$$Bf \to bB$$
$$Cf \to cC$$
$$Ag \to a$$
$$Bg \to b$$
$$Cg \to c$$

Show that $a^3b^3c^3$ is a word in the language of this grammar and give a description of the language.

Context free languages are indexed because every context free grammar is an indexed grammar without the indices.

Example 26. *The following is a short list of languages which are indexed but are not context-free.*

(1) $\{0^n 1^n 2^n \,|\, n \geq 1\}$
(2) $\{0^{n^2} \,|\, n \geq 1\}$
(3) $\{0^n \,|\, n \text{ is composite}\}$
(4) $\{0^{2^n} \,|\, n \geq 1\}$
(5) $\{ww \,|\, w \in \{0,1\}^*\}$
(6) $\{a^i b a^j b \cdots a^k b \,|\, i \geq j \geq \cdots \geq k\}$
(7) $\{a^n b^{n^2} \,|\, n \geq 1\}$

Exercise 27. *Give grammars for each of the languages in the previous example.*

Exercise 28. *Using the pumping lemma for context-free languages, show that each of the languages in the previous example are not context-free.*

Indexed languages are computationally equivalent to a machine called a nested stack automata (NSA). When the NSA is in 'read mode' in the stack it has the ability of creating a new stack. This new stack must be destroyed before any more of the previous stack can be read. This process of creating a new stack can be repeated and it allows the creation of new stacks to an arbitrary depth.

References

[1] A. Aho, Indexed Grammars– An Extension of Context Free Grammars, Journal of the Association for Computing Machinery, Vol. 15, No. 4, (1968), 647–671.
[2] P. Flajolet, Analytic Models and Ambiguity of Context-Free Languages, Theoretical Computer Science, 49 (1987), 283-309.
[3] J. Hopcroft and J. Ullman. Introduction to automata theory, languages, and computation. Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.
[4] M. Sipser, Introduction to the Theory of Computation, PWS Publishing Company, 1997.