COMS3261: Computer Science Theory

Fall 2013

Mihalis Yannakakis

Lecture 9, 10/2/13

Context–Free Languages

- Defined originally by Chomsky in 1950's along with context-free grammars for natural language processing
- Then applied to specify programming languages – BNF syntax; led to automation of parsing, compilation

- Will talk about two types of representations:
- Context-Free Grammars: Recursive definition of sets of strings
  (e.g. recall recursive definition of regular expressions)
- Pushdown Automata

Example: Palindromes

- Recursive (inductive) definition of palindromes over alphabet {0,1} (similar for arbitrary alphabet)
- Basis: ε, 0, 1 are palindromes
- Induction (recursion): If w is a palindrome then 0w0 and 1w1 are also palindromes
- (implicit rule: Nothing else is a palindrome)

Context-free Grammar for Palindromes

$$\left.\begin{array}{l} \bullet\ S \to \varepsilon \\ \bullet\ S \to 0 \\ \bullet\ S \to 1 \\ \bullet\ S \to 0S0 \\ \bullet\ S \to 1S1 \end{array}\right\} \text{productions (rules)}$$

head $\to$ body

terminals: 0,1 (the alphabet Σ)

variables (nonterminals) : S (in general many)

start symbol (variable): S

grammar defines what strings S represents

Derivation of strings

- Start with the symbol S, and derive other strings by using productions as rewriting rules replacing an occurrence of a head by the body of a production, until it is no more possible

- Example: $S \Rightarrow \underbrace{1S1}_{S \to 0S0} \Rightarrow \underbrace{10S01}_{S \to 0} \Rightarrow 10001$

Example: English fragment

- $S \to NP\ VP$ (Sentence = Noun-Phrase Verb-Phrase)
- $NP \to A\ N$ (Noun-Phrase = Article Noun)
- $VP \to V\ NP$ (Verb-Phrase = Verb Noun-Phrase)
- $A \to$ a $A \to$ the
- $N \to$ child $N \to$ dog
- $V \to$ likes $V \to$ sees

- a child sees a dog
- the child likes the dog

Formal Definition

- Context-free grammar G = (V, T, P, S)
- V = set of variables
- T = set of terminals (= alphabet)
- P = set of productions: rules of form
  variable $\rightarrow$ string in $(V \cup T)^*$
- S = start symbol (in V)

- Notational shorthand convention: can combine productions with same head with a | separating the bodies
- $S \rightarrow \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$

Typographical conventions

- Variables: capital
- Terminals: lower case in beginning of alphabet, digits
- Strings of variables and terminals: Greek letters
- Terminal strings: English lower case letters towards end of alphabet (x,y,z,w,..)

Derivations of a CFG

- Derivation: Start with start symbol S, and derive other strings by using productions as rewriting rules replacing an occurrence of a head by the body of a production
- Example: $S \Rightarrow \underbrace{1S1 \Rightarrow}_{S \to 0S0} \underbrace{10S01 \Rightarrow}_{S \to 0} 10001$

Generally, if $\alpha, \beta \in (V \cup T)^*$ and $A \to \gamma \in P$ then $\alpha A \beta \Rightarrow_G \alpha\gamma\beta$

($\alpha A \beta$ derives $\alpha\gamma\beta$) Usually omit subscript G if clear.

Context-free: can replace A regardless of context

$\Rightarrow^*$: reflexive transitive closure of $\Rightarrow$: derives in 0, 1 or more steps

• Example: $S \Rightarrow^* S$, $S \Rightarrow^* 10001$

Language of CFG

- Sentential forms: strings of $(V \cup T)^*$ derived from S
- Language of a CFG G:
  $L(G) = \{ w \in T^* \mid S \Rightarrow_G^* w \}$
  = set of terminal strings that can be derived from start symbol S

Proof of correctness of a CFG

Proof that example grammar G has L(G) = {palindromes}

1. w palindrome $\Rightarrow$ w in L(G)

   By induction on length of w:

   $|w| = 0$ or $1 \Rightarrow w = \varepsilon, 0, 1$ and then $S \Rightarrow w$

   $|w| \geq 2 \Rightarrow$ first and last letter are same $\Rightarrow w = 0x0$ or $w=1x1$ and x also a palindrome.

   Since x is shorter, $S \Rightarrow^* x$ by induction hypothesis.

   Therefore $S \Rightarrow 0S0 \Rightarrow^* 0x0$ and $S \Rightarrow 1S1 \Rightarrow^* 1x1$.

2. w in L(G) $\Rightarrow$ w palindrome:

   Similar, by induction on length of a derivation.

More Examples

1. $\{0^n1^n \mid n \geq 0\}$

   $S \rightarrow \varepsilon \mid 0S1$

2. $\{a^nb^n c^md^m \mid n,m \geq 0\}$

   $S \rightarrow L \mid R$

   $L \rightarrow \varepsilon \mid aLb$

   $R \rightarrow \varepsilon \mid cRd$

More Examples

3. All strings over {a,b}, i.e. {a,b}*

$S \to \varepsilon$ | aS | bS

4. All nonempty strings over {a,b,0,1} that start with a letter (cf. identifiers in a programming language)

   i.e., (a+b)(a+b+0+1)* , eg. aab01a

$I \to$ a | b | Ia | Ib | I0 | I1

- Every regular language has a context-free grammar

i.e. Regular languages $\subseteq$ Context-free languages

(will do as HW via grammars; will show later via automata)

Example: Arithmetic expressions

- T = { a,b,0,1,+,*,(,)}
- V={E,I}, S=E
- Productions

  $E \to$ I | E+E | $E * E$ | (E)

  $I \to$ a | b | Ia | Ib | I0 | I1

Leftmost, Rightmost Derivations

- A sentential form may have many occurrences of variables, we can replace any one of them

- Leftmost derivation: replace always the leftmost variable
- $E \Rightarrow E+E \Rightarrow E*E+E \Rightarrow I*E+E \Rightarrow a*E+E \Rightarrow a*I+E \Rightarrow a*b+E \Rightarrow a*b+I \Rightarrow a*b+a$

- Rightmost derivation: replace always the rightmost variable
- $E \Rightarrow E+E \Rightarrow E+I \Rightarrow E+a \Rightarrow E*E+a \Rightarrow E*I+a \Rightarrow E*b+a \Rightarrow I*b+a \Rightarrow a*b+a$

- Left / right sentential form