Context Free Languages

Mark Greenstreet, CpSc 421, Term 1, 2006/07

Lecture Outline

Context Free Languages

- 0^n1^n – One More Time
- Formal Definition
- More Examples

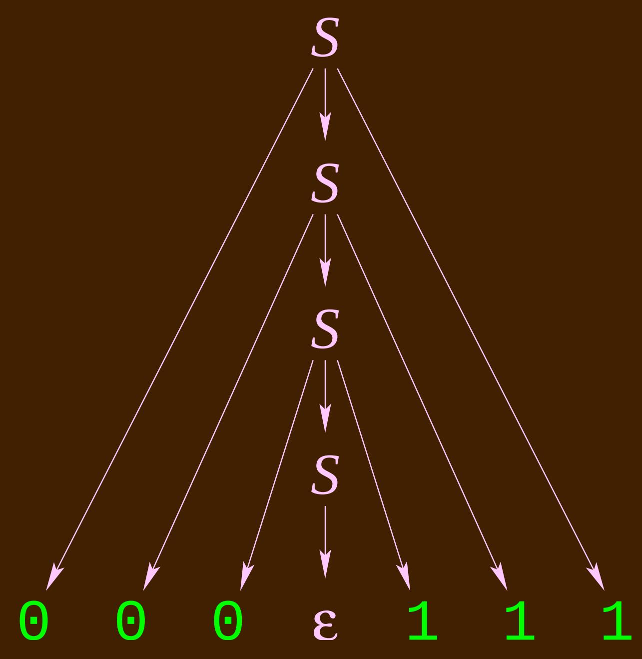
0^n1^n

- Let $A = 0^n1^n$. A is not regular.
- Here's an inductive definition of the language. A string, w, is in A iff
  - $w = \epsilon$, or
  - There is a string, $x \in A$ such that $w = 0x1$.
- Can we formalize this approach?
  - Why formalize?
    We formalize the definition of languages so we can reason about properties that every language in some class has. That way, we don't have to prove properties individually.
  - Why not inductive definitions with English? Because it's not possible/practical to determine what can and cannot be said in English. How would you write an English sentence to state something that can't be said in English?

A Notation for Describing $0^n 1^n$

- $A \rightarrow \epsilon \mid \texttt{0}\, A\, \texttt{1}$
- A string is in the language if we can derive it from A using these two rules.
- Example: `000111`

$$\begin{aligned} A &\rightarrow \texttt{0}\, A\, \texttt{1} \\ &\rightarrow \texttt{00}\, A\, \texttt{11} \\ &\rightarrow \texttt{000}\, A\, \texttt{111} \\ &\rightarrow \texttt{000}\, \epsilon\, \texttt{111} = \texttt{000111} \end{aligned}$$



$\#0$'s $= \#1$'s

- Let B be the language of strings that have an equal number of 0's and 1's.
- From the Sept. 8 notes,
  - $w = \epsilon$; or
  - There is a string x in B such that $w = \texttt{0}x\texttt{1}$ or $w = \texttt{1}x\texttt{0}$; or
  - There are strings x and y in B such that $w = xy$.
- Can we write this in our new notation?

$\#0$'s $= \#1$'s

- Let B be the language of strings that have an equal number of 0's and 1's.
- From the Sept. 8 notes,
  - $w = \epsilon$; or
  - There is a string x in B such that $w = \texttt{0}x\texttt{1}$ or $w = \texttt{1}x\texttt{0}$; or
  - There are strings x and y in B such that $w = xy$.
- Can we write this in our new notation?

$$
\begin{array}{lcl}
B & \rightarrow & \epsilon \\
 & | & \texttt{0}\ B\ \texttt{1} \\
 & | & \texttt{1}\ B\ \texttt{0} \\
 & | & B\ B
\end{array}
$$

$\#0$'s $< \#1$'s

- Let C be the language of strings that have an fewer 0's than 1's.
- From the HW 0 solution set: String w is in C iff
  - There are strings x and y in B such that $w = x1y$, where B is the set of all string with an equal number of ones and zeros as defined in the problem statement.
  - There are strings x and y in C such that $w = xy$.
- In our new notation, this is:

$\#0$**'s** $< \#1$**'s**

- Let C be the language of strings that have an fewer 0's than 1's.
- From the HW 0 solution set: String w is in C iff
  - There are strings x and y in B such that $w = x1y$, where B is the set of all string with an equal number of ones and zeros as defined in the problem statement.
  - There are strings x and y in C such that $w = xy$.
- In our new notation, this is:

$$\begin{array}{rcl} C & \rightarrow & B\,\texttt{1}\,B \quad | \quad C\,C \\ B & \rightarrow & \epsilon \quad | \quad B\,B \\ & | & \texttt{0}\,B\,\texttt{1} \quad | \quad \texttt{1}\,B\,\texttt{0} \end{array}$$

Formalizing Our Notation

- A context-free grammar (CFG) is a 4-tuple, (V, Σ, R, S) where . . .

Arithmetic Expressions

$G = (V, \Sigma, R, Expr)$, where

$$
\begin{aligned}
V &= \{Expr, ExprList, NonEmptyExprList\} \\
\Sigma &= \{\texttt{INTEGER}, \texttt{IDENTIFIER}, \texttt{PLUS}, \texttt{MINUS}, \\
&\quad \texttt{TIMES}, \texttt{DIVIDE}, \texttt{EXP}, \\
&\quad \texttt{LPAREN}, \texttt{RPAREN}, \texttt{COMMA}\}
\end{aligned}
$$

$$
\begin{aligned}
Expr \rightarrow\ & \texttt{INTEGER} && \mid \texttt{IDENTIFIER} \\
\mid\ & Expr\ \texttt{PLUS}\ Expr && \mid Expr\ \texttt{MINUS}\ Expr \\
\mid\ & Expr\ \texttt{TIMES}\ Expr && \mid Expr\ \texttt{DIVIDE}\ Expr \\
\mid\ & Expr\ \texttt{EXP}\ Expr && \mid \texttt{LPAREN}\ Expr\ \texttt{RPAREN} \\
\mid\ & \texttt{IDENTIFIER}\ \texttt{LPAREN}\ ExprList\ \texttt{RPAREN}
\end{aligned}
$$

$$ExprList \rightarrow \epsilon \mid NonEmptyExprList$$

$$NonEmptyExprList \rightarrow Expr$$

$$\mid NonEmptyExprList\ \texttt{COMMA}\ Expr\ .$$

Arithmetic Terminals

Regular Expressions:

INTEGER	$\equiv$	DIGITDIGIT*			
DIGIT	$\equiv$	$0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9$			
IDENTIFIER	$\equiv$	ISTARTITAIL*			
ISTART	$\equiv$	$A \cup B \cup \ldots \cup Z \cup a \cup b \cup \ldots \cup z$			
ITAIL	$\equiv$	ISTART $\cup$ DIGIT			
PLUS	$\equiv$	+	MINUS	$\equiv$	-
TIMES	$\equiv$	*	DIVIDE	$\equiv$	/
EXP	$\equiv$	$\wedge$	COMMA	$\equiv$	,
LPAREN	$\equiv$	(	RPAREN	$\equiv$	)

Arithmetic Example

$2 + 3 * 4$

$$
\begin{aligned}
\mathit{Expr} &\Rightarrow \mathit{Expr}\ \texttt{PLUS}\ \mathit{Expr} \\
&\Rightarrow \texttt{INTEGER}\ \texttt{PLUS}\ \mathit{Expr} \\
&\Rightarrow \texttt{INTEGER}\ \texttt{PLUS}\ \mathit{Expr}\ \texttt{TIMES}\ \mathit{Expr} \\
&\Rightarrow \texttt{INTEGER}\ \texttt{PLUS}\ \texttt{INTEGER}\ \texttt{TIMES}\ \mathit{Expr} \\
&\Rightarrow \texttt{INTEGER}\ \texttt{PLUS}\ \texttt{INTEGER}\ \texttt{TIMES}\ \texttt{INTEGER}
\end{aligned}
$$