Context Free Grammars and Languages

Grammar Rules

Consider the following set of rules:

- $< \text{assg-stmt} >::=< \text{var} >:=< \text{arith-expr} >$

Grammar Rules

Consider the following set of rules:

- $<$ assg-stmt $>$::=$<$ var $>$:=$<$ arith-expr $>$
- $<$ arith-expr $>$::=$<$ var $>$ $|$ $<$ const $>$ $|$ $<$ arith-expr $><$ arith-op $><$ arith-expr $>$

Grammar Rules

Consider the following set of rules:

- $<$ assg-stmt $>::=<$ var $>:=<$ arith-expr $>$
- $<$ arith-expr $>::=<$ var $>$ | $<$ const $>$ | $<$ arith-expr $><$ arith-op $><$ arith-expr $>$
- $<$ arith-op $>::= +|-|*|/$

Grammar Rules

Consider the following set of rules:

- $<$ assg-stmt $>::=<$ var $>:=<$ arith-expr $>$
- $<$ arith-expr $>::=<$ var $>$ | $<$ const $>$ | $<$ arith-expr $><$ arith-op $><$ arith-expr $>$
- $<$ arith-op $>::= +| - | * |/$
- $<$ const $>::= 0|1|2|3|4|5|6|7|8|9$

Grammar Rules

Consider the following set of rules:

- $< \text{assg-stmt} >::=< \text{var} >:=< \text{arith-expr} >$
- $< \text{arith-expr} >::=< \text{var} > \mid < \text{const} > \mid < \text{arith-expr} >< \text{arith-op} >< \text{arith-expr} >$
- $< \text{arith-op} >::= +\mid - \mid * \mid /$
- $< \text{const} >::= 0|1|2|3|4|5|6|7|8|9$
- $< \text{var} >::= a|b|c| \ldots |x|y|z$

Grammar Rules

- Objects of the form $< \times \times \times >$ are called *nonterminal symbols*.

Grammar Rules

- Objects of the form $< \times \times \times >$ are called *nonterminal symbols*.
- Each nonterminal symbol generates a set of strings over finite alphabet $\Sigma = \{0, 1, \ldots, 9, a, b, \ldots, z, +, -, *, /\}$.

Grammar Rules

- Objects of the form $< \times \times \times >$ are called *nonterminal symbols*.
- Each nonterminal symbol generates a set of strings over finite alphabet $\Sigma = \{0, 1, \ldots, 9, a, b, \ldots, z, +, -, *, /\}$.
- The *language* generated by the set of rules or the *grammar*: Arithmetic expressions formed by using the 4 basic arithmetic operators over the integers $0 - 9$ and variables.
  Infinitely many strings in this language represented in a finite manner using the grammar.

Grammar Rules

- Objects of the form $< \times \times \times >$ are called *nonterminal symbols*.
- Each nonterminal symbol generates a set of strings over finite alphabet $\Sigma = \{0, 1, \ldots, 9, a, b, \ldots, z, +, -, *, /\}$.
- The *language* generated by the set of rules or the *grammar*: Arithmetic expressions formed by using the 4 basic arithmetic operators over the integers $0 - 9$ and variables.
  Infinitely many strings in this language represented in a finite manner using the grammar.
- These rules define a syntax for the language.

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$
- $\rightarrow < var > := < arith - expr > < arith - op > < arith - expr >$

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op >< arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op >< arith - expr >< arith - op >< arith - expr >$

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op >< arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op >< arith - expr >< arith - op >< arith - expr >$
- $\rightarrow x := y + z - 3$.

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$
- $\rightarrow < var > := < arith - expr > < arith - op > < arith - expr >$
- $\rightarrow < var > := < arith - expr > < arith - op > < arith - expr > < arith - op > < arith - expr >$
- $\rightarrow x := y + z - 3$.
- *Sentential forms*: The expressions with non-terminal symbols in the intermediary derivation steps.

Derivation of Arithmetic Expressions from the Rules

- Can you derive $x = y + z - 3$ using the rules?
- $< asst - stmt > \rightarrow < var > := < arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op >< arith - expr >$
- $\rightarrow < var > := < arith - expr >< arith - op ><$ $arith - expr >< arith - op >< arith - expr >$
- $\rightarrow x := y + z - 3$.
- *Sentential forms*: The expressions with non-terminal symbols in the intermediary derivation steps.
- Can you give two ways to derive $x = y + z - 3$?
  $x = (y + z) - 3$, $x = y + (z - 3)$

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.
- N is a finite set (the nonterminal symbols),

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.
- N is a finite set (the nonterminal symbols),
- Σ is a finite set (the terminal symbols) disjoint from N,

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.
- N is a finite set (the nonterminal symbols),
- Σ is a finite set (the terminal symbols) disjoint from N,
- P is a finite subset of $N \times (N \cup \Sigma)^*$ (the productions)

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.
- N is a finite set (the nonterminal symbols),
- Σ is a finite set (the terminal symbols) disjoint from N,
- P is a finite subset of $N \times (N \cup \Sigma)^*$ (the productions)
- S is the start symbol

Context Free Grammar (CFG)

- A CFG is denoted as $G = (N, \Sigma, P, S)$.
- N is a finite set (the nonterminal symbols),
- Σ is a finite set (the terminal symbols) disjoint from N,
- P is a finite subset of $N \times (N \cup \Sigma)^*$ (the productions)
- S is the start symbol
- Finite representation for a set of possibly infinite strings.

Notational Conventions

- Nonterminals usually denoted by capital letter $A, B, \ldots$,

Notational Conventions

- Nonterminals usually denoted by capital letter $A, B, \ldots,$
- Terminal symbols: $a, b, \ldots$

Notational Conventions

- Nonterminals usually denoted by capital letter $A, B, \ldots,$
- Terminal symbols: $a, b, \ldots$
- Strings in $(N \cup \Sigma)^*$: $\alpha, \beta, \gamma, \ldots$

Notational Conventions

- Nonterminals usually denoted by capital letter $A, B, \ldots,$
- Terminal symbols: $a, b, \ldots$
- Strings in $(N \cup \Sigma)^*$: $\alpha, \beta, \gamma, \ldots$
- Productions: Usually written as $A \to \alpha$ instead of (A, α).

Notational Conventions

- Nonterminals usually denoted by capital letter $A, B, \ldots,$
- Terminal symbols: $a, b, \ldots$
- Strings in $(N \cup \Sigma)^*$: $\alpha, \beta, \gamma, \ldots$
- Productions: Usually written as $A \to \alpha$ instead of (A, α).
- Suppose there are several productions from the same nonterminal: $A \to \alpha_1, A \to \alpha_2, A \to \alpha_3$. Then shorten this as $A \to \alpha_1|\alpha_2|\alpha_3$.

Derivations

- If $\alpha, \beta \in (N \cup \Sigma)^*$, then β is *derivable* from α in 1 step $[\alpha \rightarrow_G^1 \beta]$ if
  There exists a production $A \rightarrow \gamma$ such that $\alpha = \alpha_1 A \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$.

Derivations

- If $\alpha, \beta \in (N \cup \Sigma)^*$, then β is *derivable* from α in 1 step $[\alpha \to_G^1 \beta]$ if
  There exists a production $A \to \gamma$ such that $\alpha = \alpha_1 A \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$.
- Define $\to_G^*$ to be the reflexive transitive closure of $\to_G^1$:
  $\alpha \to_G^0 \alpha$ for all α,
  $\alpha \to_G^{n+1} \beta$ if there is a γ such that $\alpha \to_G^n \gamma$ and $\gamma \to_G^1 \beta$,
  $\alpha \to_G^* \beta$ if there is an $n \geq 0$ such that $\alpha \to_G^n \beta$.

Sentential Forms and Sentences

- A string derivable from start symbol S: sentential form;
  A sentential form with no nonterminal symbols is a *sentence*.

Sentential Forms and Sentences

- A string derivable from start symbol S: sentential form;
  A sentential form with no nonterminal symbols is a *sentence*.
- $L(G) = \{x \in \Sigma^* | S \rightarrow_G^* x\}$.

Sentential Forms and Sentences

- A string derivable from start symbol S: sentential form;
  A sentential form with no nonterminal symbols is a *sentence*.
- $L(G) = \{x \in \Sigma^* | S \rightarrow_G^* x\}$.
- $B \subseteq \Sigma^*$ is a Context Free Language (CFL) if $B = L(G)$ for a CFG G.

Example 1

Set $\{a^n b^n | n \geq 0\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSb | \epsilon$.

Example 1

Set $\{a^n b^n | n \geq 0\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSb | \epsilon$.
- Can you derive $a^3 b^3$?

Example 1

Set $\{a^n b^n | n \geq 0\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSb|\epsilon$.
- Can you derive $a^3 b^3$?
- $S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaabbb$.

Example 1

Set $\{a^n b^n | n \geq 0\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSb | \epsilon$.
- Can you derive $a^3 b^3$?
- $S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaabbb$.
- Can you have multiple derivations of $a^3 b^3$? *Unambiguous grammar* - more on this later.

Example 2

Set of palindromes over $\{a, b\}$, or $\{x \in \{a, b\}^* | x = x^{\text{rev}}\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSa|bSb|a|b|\epsilon$.

Example 2

Set of palindromes over $\{a, b\}$, or $\{x \in \{a, b\}^* | x = x^{\text{rev}}\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSa|bSb|a|b|\epsilon$.
- First 2 productions: for balancing the outer ends of the string

Example 2

Set of palindromes over $\{a, b\}$, or $\{x \in \{a, b\}^* | x = x^{\text{rev}}\}$ is a CFL (not regular!)

- Generated by the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}, \Sigma = \{a, b\}, P = S \rightarrow aSa|bSb|a|b|\epsilon$.
- First 2 productions: for balancing the outer ends of the string
- Last 3 productions: for finishing derivations.
  $S \rightarrow a|b$ are used to finishing odd length strings,
  $S \rightarrow \epsilon$ is used for finishing even length strings.