Context-Free Grammar & Language

A Context-Free Grammar

A **context-free grammar (CFG)** is a **finite description** or **specification** of a language. A **language** is called **context-free** if it can be described by a **CFG**.

Definition

A **context-free grammar** $\mathbf{G}$ is a **4-tuple** $(\mathbf{V}, \mathbf{\Sigma}, \mathbf{R}, \mathbf{S})$ of data, where

- $\mathbf{V}$ is a **finite set** of **nonterminals** or **variables**.
- $\mathbf{\Sigma}$ is the **object language alphabet**. Elements of Σ are called the **terminals** or **constants**.

Definition

- $\mathbf{R}$ is a **finite subset** of $\mathbf{V} \times (\mathbf{V} \cup \mathbf{\Sigma})^*$, known as the set of **production** or **inference rules**. An element $(\mathbf{A}, \alpha) \in \mathbf{R}$ is written as $\mathbf{A} \rightarrow \alpha$ and is read as 'A **produces** α'. If $(\mathbf{A}, \alpha_\mathbf{1}), \cdots, (\mathbf{A}, \alpha_\mathbf{k}) \in \mathbf{R}$, it may be written as $\mathbf{A} \rightarrow \alpha_\mathbf{1} \mid \alpha_\mathbf{2} \mid \cdots \mid \alpha_\mathbf{k}$.
- $\mathbf{S}$ is a **distinguished** element of V and is called the **start symbol** or the **axiom**.

An Example

The **context-free grammar** of the language $\mathbf{L = \{a^n b^n \ : \ n \geq 0\}}$ is

$$(\{\mathbf{S}\}, \{\mathbf{a}, \mathbf{b}\}, \{\mathbf{S} \rightarrow \mathbf{aSb} \mid \varepsilon\}, \mathbf{S}).$$

How Does it Work?

- Define a **binary relation** '$\Rightarrow$' over $(\mathbf{V} \cup \mathbf{\Sigma})^*$ in the following way.

$$(\alpha\mathbf{A}\beta, \alpha\gamma\beta) \in \Rightarrow, \ \textbf{if } \mathbf{A} \rightarrow \gamma \in \mathbf{R},$$

where $A \in V$. We write $\alpha\mathbf{A}\beta \ \Rightarrow \ \alpha\gamma\beta$ and read '$\alpha A\beta$ produces $\alpha\gamma\beta$ in one step'.

- The **binary relation** '$\stackrel{*}{\Rightarrow}$' is the **reflexive-transitive closure** of '$\Rightarrow$'. By $\alpha \stackrel{*}{\Rightarrow} \beta$ we mean that either $\alpha = \beta$, or α produces β in **finite** number of steps.

Language of a Grammar

Let $\mathbf{G} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{R}, \mathbf{S})$ be a **CFG**. The **language specified** by the **grammar** G is

$$\mathbf{L}(\mathbf{G}) = \{\mathbf{x} \in \mathbf{\Sigma}^* \ : \ \mathbf{S} \ \stackrel{*}{\Rightarrow} \mathbf{x}\}.$$

Definition

A **language** $\mathbf{L}$ over an alphabet $\mathbf{\Sigma}$ is called a **context-free language (CFL)**, if there is a **CFG**, $\mathbf{G} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{R}, \mathbf{S})$, such that $\mathbf{L} = \mathbf{L}(\mathbf{G})$.

A Small Proof

$$\mathbf{L} = \{\mathbf{a^n b^n} : \mathbf{n} \geq \mathbf{0}\} = \mathbf{L}(\{\mathbf{S}\}, \{\mathbf{a}, \mathbf{b}\}, \{\mathbf{S} \rightarrow \mathbf{aSb} \mid \varepsilon\}, \mathbf{S}).$$

Proof ($L(G) \subseteq L$):
The proof is by **induction** on the **length of derivation**. We shall prove a **more general result** to do the induction.

Claim: Every **sentential form**[a] has a's followed by equal number of b's. There may be an **S** between them.

[a] α is a **sentential form** if $\mathbf{S} \stackrel{*}{\Rightarrow} \alpha$; and α is a **sentence** if $\alpha \in \mathbf{\Sigma}^*$.

A Small Proof

The **base cases** are **sentential forms** of length **zero** and **three** after **one step derivation**.

If the claim holds after **n** steps of derivation and the nonterminal **S** is at the middle, the claim is true after the **next step of derivation** due to the rules of **R**.

A Small Proof

Proof ($L \subseteq L(G)$):
The proof is by **induction** on the **length of strings** of L.

In the **base case** the string of length zero can be derived from **G**.

Let all strings of length $\mathbf{2k}$ can be derived from **G**. A string of length $\mathbf{2(k+1)}$ is of the form **axb** and can be derived as

$$\mathbf{S} \Rightarrow \mathbf{aSb} \stackrel{*}{\Rightarrow} \mathbf{axb}.$$

Example

Let the grammar be $(\{S\}, \{a, b\}, \{S \rightarrow aSb \mid \varepsilon\}, S)$.

- $S \Rightarrow \varepsilon$.
- $S \Rightarrow aSb \Rightarrow ab$.
- $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$.

This process is called the **derivation** of the strings of the language. The **grammar** and the **language** is called **context-free** because the **repalcement** of a **nonterminal** does not depend on its **context**.

Example

Consider the language of **arithmetic expressions** over constants (**c**) and usual binary operators $\{+, -, *, /, (,)\}$. The grammar $\mathbf{G} = (\{\mathbf{E}, \mathbf{F}, \mathbf{T}\}, \{+, -, *, /, (,), \mathbf{c}\}, \mathbf{R}, \mathbf{E})$, where

$$
\begin{aligned}
& R: \\
& E \rightarrow E+T \mid E-T \mid T \\
& T \rightarrow T*F \mid T/F \mid F \\
& F \rightarrow (E) \mid c
\end{aligned}
$$

The **terminal** '**c**' is for any constant.

Derivation of $c * (c - c)$

$$
\begin{aligned}
E \Rightarrow T &\Rightarrow T * F \\
&\Rightarrow F * F \\
&\Rightarrow c * F \\
&\Rightarrow c * (E) \\
&\Rightarrow c * (E - T) \\
&\Rightarrow c * (T - T) \\
&\Rightarrow c * (F - T) \\
&\Rightarrow c * (c - T) \\
&\Rightarrow c * (c - F) \Rightarrow c * (c - c)
\end{aligned}
$$

Derivation of $c * (c - c)$ is not Unique

$$
\begin{aligned}
E \Rightarrow T \;&\Rightarrow\; T * F \\
&\Rightarrow\; T * (E) \\
&\Rightarrow\; T * (E - T) \\
&\Rightarrow\; T * (E - F) \\
&\Rightarrow\; T * (E - c) \\
&\Rightarrow\; T * (T - c) \\
&\Rightarrow\; T * (F - c) \\
&\Rightarrow\; T * (c - c) \\
&\Rightarrow\; F * (c - c) \;\Rightarrow\; c * (c - c)
\end{aligned}
$$

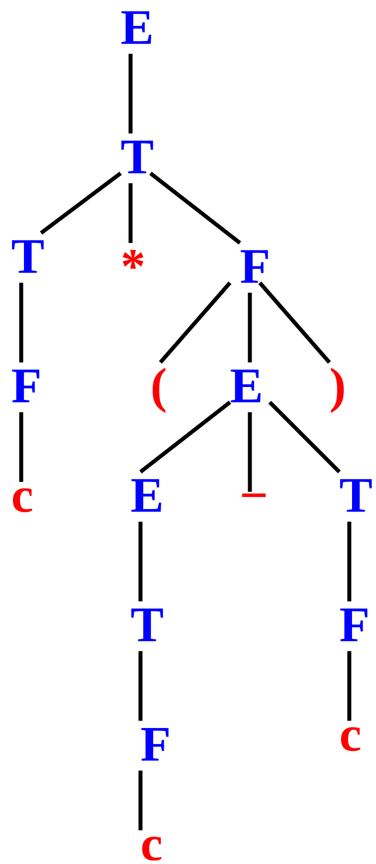

Figure 1: **Parse Tree of** $c * (c - c)$

Unique Parse Tree

There is **exactly one parse tree** for any valid arithmetic expression and the given grammar.

Another Grammar for Arithmetic Expression

Let $\mathbf{G} = (\{\mathbf{E}\}, \{+, -, *, /, (,), \mathbf{c}\}, \mathbf{R}, \mathbf{E})$ where

$R:$

$$E \Rightarrow E + E \mid E - E \mid E * E \mid E/E \mid (E) \mid c$$

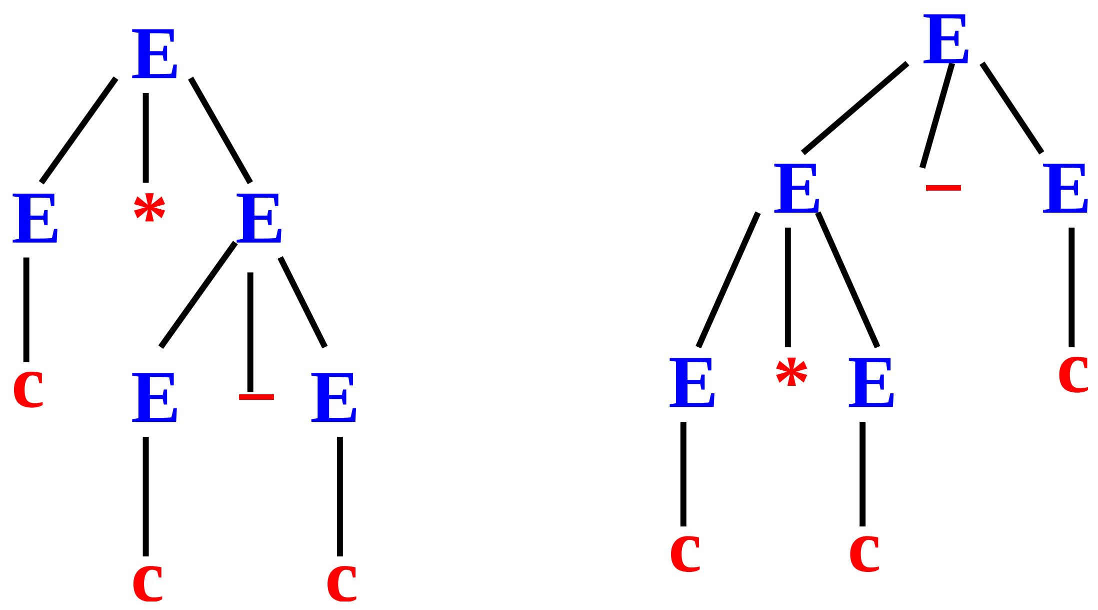

Figure 2: **Two Parse Trees of** $c * c - c$

Ambiguous Grammar

A grammar **G** is called **ambiguous** if there is a **sentence** $\mathbf{x} \in \mathbf{\Sigma}^*$ so that it has **more than one parse trees**.

The second grammar for the arithmetic expressions is **ambiguous**.

A CFL **L** is called **inherently ambiguous** if it **does not** have **any nonambiguous CFG**.

An Inherently Ambiguous Language

$$\{a^i b^j c^k \ : \ i = j \text{ or } j = k\}$$

Any string of the form $a^n b^n c^n$ can be generated in two possible ways.

Finite Union of CFL

If $\mathbf{L_1}$ and $\mathbf{L_2}$ are **CFL**s, then so is $\mathbf{L_1 \cup L_2}$.

Construction: Let $\mathbf{G_1 = (V_1, \Sigma, R_1, S_1)}$ and $\mathbf{G_2 = (V_2, \Sigma, R_2, S_2)}$ be two **CFG**s such that $\mathbf{L(G_1) = L_1}$ and $\mathbf{L(G_2) = L_2}$. The grammar for $\mathbf{L_1 \cup L_2}$ is

$$\mathbf{G = (V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \rightarrow S_1 | S_2\}, S)},$$

where $S \notin V_1 \cup V_2$.

Pumping Lemma of CFL

Let $\mathbf{L}$ be a **context-free language**. There is a positive number $\mathbf{p}$, called the **pumping length**, such that if $\mathbf{w} \in \mathbf{L}$ and the **length** of $\mathbf{w}$ is at least $\mathbf{p}$, then $\mathbf{w}$ can be written as $\mathbf{w = uvxyz}$ so that

1. $|\mathbf{vxy}| \leq \mathbf{p}$,
2. $|\mathbf{vy}| > \mathbf{0}$,

and $\mathbf{w = uv^k xy^k z \in L}$ for all $\mathbf{k \geq 0}$.

Pumping Lemma of CFL

If the language is **context-free**, then the **pumping lemma**.

If the **pumping lemma does not hold**, the language is **not context-free**.

Use of Pumping Lemma : An Example

$\mathbf{L = \{a^n \ : \ } n \mathbf{\ is\ a\ prime\}}$ is not a context-free language.

Assume that the language $\mathbf{L}$ be context-free. Let the **pumping length** be $\mathbf{p}$. Let $\mathbf{n_0 \geq p}$ be a prime number. $\mathbf{a^{n_0} \in L}$ and $\mathbf{a^{n_0} = xyuvw}$. Let $\mathbf{|yv| = b}$; then $\mathbf{|xuw| = n_0 - b}$. By pumping lemma $\mathbf{a^{n_0+kb}}$, for all $\mathbf{k \geq -1}$ are elements of L. Therefore $\mathbf{a^{n_0+n_0b} \in L}$, but then $\mathbf{n_0 + n_0b}$ is not a prime - contradiction.

CFL is not Closed Under Intersection

- It is not dfficult to prove using the **pumping lemma** that the language $\mathbf{l = \{a^n b^n c^n \ : \ n \geq 0\}}$ is not context-free.
- But the following **two languages** are contxt-free.

$$\mathbf{L_1 = \{a^n b^n c^m \ : \ m, n \geq 0\}}$$

$$\mathbf{L_2 = \{a^n b^m c^m \ : \ m, n \geq 0\}}$$

- $\mathbf{L = L_1 \cap L_2}$, the collection of context-free languages is **not closed under intersection**.