

Context-free grammars

We move on from regular languages to **context-free languages**, a language family that contains all regular languages but also some non-regular ones. For example

$$\{a^n b^n \mid n \geq 0\}$$

is a context-free language. (On the other hand there are many simple languages which are not context-free, for example $\{a^n b^n c^n \mid n \geq 0\}$.)

Context-free languages are recognized by automata with access to an infinite memory that is organized as a stack (LIFO, last-in-first-out). These automata are called **pushdown automata**.

We also have a generative model for the family, so-called **context-free grammars**.

Context-free languages have applications in compiler design (parsers). The syntax of programming languages is often given in the form of context-free grammar, or equivalent Backus-Naur form (BN-form).

Example. Here is an example of a context-free grammar. It uses **variables** A and B , and **terminal** symbols a and b . We have following **productions** (also called rewrite rules):

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

In a word containing variables and terminals we can replace any variable with the word on the right hand side of a production for that variable. For example, in the word

$$aBabAbA$$

we may replace the first occurrence of variable A by AbA , obtaining word

Example. Here is an example of a context-free grammar. It uses **variables** A and B , and **terminal** symbols a and b . We have following **productions** (also called rewrite rules):

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

In a word containing variables and terminals we can replace any variable with the word on the right hand side of a production for that variable. For example, in the word

$$aBabAbA$$

we may replace the first occurrence of variable A by AbA , obtaining word

$$aBabAbAbA$$

Then we can decide to replace the variable B by b which gives the word

Example. Here is an example of a context-free grammar. It uses **variables** A and B , and **terminal** symbols a and b . We have following **productions** (also called rewrite rules):

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

In a word containing variables and terminals we can replace any variable with the word on the right hand side of a production for that variable. For example, in the word

$$aBabAbA$$

we may replace the first occurrence of variable A by AbA , obtaining word

$$a\textcolor{red}{B}abAbAbA$$

Then we can decide to replace the variable B by b which gives the word

$$a\textcolor{red}{b}abAbAbA$$

Rewriting one variable is called a **derivation step** and we denote

$$aBabAbA \Rightarrow aBabAbAbA \Rightarrow ababAbAbA$$

Derivation is non-deterministic: usually we have many choices how to proceed.

Example. Here is an example of a context-free grammar. It uses **variables** A and B , and **terminal** symbols a and b . We have following **productions** (also called rewrite rules):

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

In a word containing variables and terminals we can replace any variable with the word on the right hand side of a production for that variable. For example, in the word

$$aBabAbA$$

we may replace the first occurrence of variable A by AbA , obtaining word

$$a\textcolor{red}{B}abAbAbA$$

Then we can decide to replace the variable B by b which gives the word

$$a\textcolor{red}{b}abAbAbA$$

Rewriting one variable is called a **derivation step** and we denote

$$aBabAbA \Rightarrow aBabAbAbA \Rightarrow ababAbAbA$$

Derivation is non-deterministic: usually we have many choices how to proceed.

We can continue derivation as long as there exist variables in the word. Once the word contains only terminal symbols the derivation terminates.

Example continues:

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

One variable is called the **start symbol**- In this example, let us choose A as the start symbol. All derivations start with the start symbol A , *i.e.*, initially the word is A .

The **language defined by the grammar** contains all words of terminal symbols that can be obtained from the start symbol by applying the productions on the variables.

Example continues:

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

One variable is called the **start symbol**- In this example, let us choose A as the start symbol. All derivations start with the start symbol A , *i.e.*, initially the word is A .

The **language defined by the grammar** contains all words of terminal symbols that can be obtained from the start symbol by applying the productions on the variables.

For example, the word $aabaababa$ is in the language defined by our grammar since we have the following valid derivation:

$$A \Rightarrow$$

Example continues:

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

One variable is called the **start symbol**- In this example, let us choose A as the start symbol. All derivations start with the start symbol A , *i.e.*, initially the word is A .

The **language defined by the grammar** contains all words of terminal symbols that can be obtained from the start symbol by applying the productions on the variables.

For example, the word $aabaababa$ is in the language defined by the grammar since we have the following valid derivation:

$$A \Rightarrow AbA \Rightarrow BbA \Rightarrow aBabA \Rightarrow aaBaabA \Rightarrow aabaabA \Rightarrow aabaabB \Rightarrow aabaabaBa \Rightarrow aabaababa$$

Example continues:

$$A \longrightarrow AbA$$

$$A \longrightarrow B$$

$$B \longrightarrow aBa$$

$$B \longrightarrow b$$

One variable is called the **start symbol**- In this example, let us choose A as the start symbol. All derivations start with the start symbol A , *i.e.*, initially the word is A .

The **language defined by the grammar** contains all words of terminal symbols that can be obtained from the start symbol by applying the productions on the variables.

For example, the word $aabaababa$ is in the language defined by the grammar since we have the following valid derivation:

$$A \Rightarrow AbA \Rightarrow BbA \Rightarrow aBabA \Rightarrow aaBaabA \Rightarrow aabaabA \Rightarrow aabaabB \Rightarrow aabaabaBa \Rightarrow aabaababa$$

On the other hand, the word aaa is not in the language because the only way to produce letter a is to use the production

$$B \longrightarrow aBa$$

and it creates two a 's, so the number of a 's has to be even.

Here's the formal definition of context-free grammars.

A context-free grammar $G = (V, T, P, S)$ consists of

- two disjoint alphabets, V and T , containing **variables** (=nonterminals) and **terminals**, respectively,
- a finite set P of **productions** of the form

$$A \longrightarrow \alpha$$

where $A \in V$ is a variable, and $\alpha \in (V \cup T)^*$ is a word of terminals and variables,

- a **start symbol** $S \in V$.

For any $\alpha, \beta \in (T \cup V)^*$ we denote

$$\alpha \Rightarrow \beta$$

if β is obtained from α by rewriting one variable in α using some production from P . This is called a **derivation step**. More precisely,

$$xAy \Rightarrow xwy$$

for $A \in V$ and $x, y, w \in (V \cup T)^*$ if and only if

$$A \longrightarrow w \in P.$$

For any $\alpha, \beta \in (T \cup V)^*$ we denote

$$\alpha \Rightarrow \beta$$

if β is obtained from α by rewriting one variable in α using some production from P . This is called a **derivation step**. More precisely,

$$xAy \Rightarrow xwy$$

for $A \in V$ and $x, y, w \in (V \cup T)^*$ if and only if

$$A \longrightarrow w \in P.$$

We write

$$\alpha \Rightarrow^* \beta$$

if there is a sequence of derivation steps

$$\alpha = \alpha_1 \Rightarrow \alpha_2 \Rightarrow \alpha_3 \Rightarrow \cdots \Rightarrow \alpha_n = \beta$$

that starts with α and leads to β . (The number of derivation steps can be zero, so always $\alpha \Rightarrow^* \alpha$.) We say that α **derives** β .

Notation $\alpha \Rightarrow^+ \beta$ means that α derives β using at least one derivation step, and $\alpha \Rightarrow^n \beta$ means a derivation with n derivation steps.

A word $\alpha \in (V \cup T)^*$ is called a **sentential form** if it can be derived from the start symbol S , *i.e.*, if

$$S \Rightarrow^* \alpha.$$

A word $\alpha \in (V \cup T)^*$ is called a **sentential form** if it can be derived from the start symbol S , *i.e.*, if

$$S \Rightarrow^* \alpha.$$

The language $L(G)$ generated by grammar G consists of all sentential forms that contain only terminals. In other words, a word is in $L(G)$ if and only if

- it can be derived from the start symbol S using the productions in P , and
- it contains only terminal symbols.

In short,

$$L(G) = \{w \in T^* \mid S \Rightarrow^* w\}.$$

A language is called a **context-free language** if it is $L(G)$ for some context-free grammar G .

Example 1. Consider the grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$, and P contains productions

$$\begin{aligned} S &\longrightarrow aSb \\ S &\longrightarrow \varepsilon \end{aligned}$$

To make notations shorter we may use the following convention: productions of the same variable may be combined on one line, separated by symbol $|$. So we may write

$$S \longrightarrow aSb \mid \varepsilon$$

What is $L(G)$?

Example 2. Consider the grammar $G = (V, T, P, E)$ where $V = \{E, N\}$, $T = \{+, *, (,), 0, 1\}$, and P contains the productions

$$\begin{aligned} E &\longrightarrow E + E \mid E * E \mid (E) \mid N \\ N &\longrightarrow 0N \mid 1N \mid 0 \mid 1 \end{aligned}$$

Example 2. Consider the grammar $G = (V, T, P, E)$ where $V = \{E, N\}$, $T = \{+, *, (,), 0, 1\}$, and P contains the productions

$$\begin{aligned} E &\longrightarrow E + E \mid E * E \mid (E) \mid N \\ N &\longrightarrow 0N \mid 1N \mid 0 \mid 1 \end{aligned}$$

For example, all following words are in the language $L(G)$:

$$\begin{aligned} &0 \\ &0 * 1 + 111 \\ &(1 + 1) * 0 \\ &(1 * 1) + (((0000)) * 1111) \end{aligned}$$

For instance $(1 + 1) * 0$ is derived by

$$E \Rightarrow$$

Example 2. Consider the grammar $G = (V, T, P, E)$ where $V = \{E, N\}$, $T = \{+, *, (,), 0, 1\}$, and P contains the productions

$$\begin{aligned} E &\longrightarrow E + E \mid E * E \mid (E) \mid N \\ N &\longrightarrow 0N \mid 1N \mid 0 \mid 1 \end{aligned}$$

For example, all following words are in the language $L(G)$:

$$\begin{aligned} &0 \\ &0 * 1 + 111 \\ &(1 + 1) * 0 \\ &(1 * 1) + (((0000)) * 1111) \end{aligned}$$

For instance $(1 + 1) * 0$ is derived by

$$E \Rightarrow E * E \Rightarrow (E) * E \Rightarrow (E + E) * E \Rightarrow^3 (N + N) * N \Rightarrow^3 (1 + 1) * 0$$

Example 2. Consider the grammar $G = (V, T, P, E)$ where $V = \{E, N\}$, $T = \{+, *, (,), 0, 1\}$, and P contains the productions

$$\begin{aligned} E &\longrightarrow E + E \mid E * E \mid (E) \mid N \\ N &\longrightarrow 0N \mid 1N \mid 0 \mid 1 \end{aligned}$$

For example, all following words are in the language $L(G)$:

$$\begin{aligned} &0 \\ &0 * 1 + 111 \\ &(1 + 1) * 0 \\ &(1 * 1) + (((0000)) * 1111) \end{aligned}$$

For instance $(1 + 1) * 0$ is derived by

$$E \Rightarrow E * E \Rightarrow (E) * E \Rightarrow (E + E) * E \Rightarrow^3 (N + N) * N \Rightarrow^3 (1 + 1) * 0$$

Here are some words that are **not** in the language $L(G)$:

$$\begin{aligned} &1(1 \\ &() \\ &1 * 1 * 1 * 1* \end{aligned}$$

- From variable N all nonempty strings of symbols 0 and 1 can be derived.
- From variable E we can derive all well-formed arithmetic expressions containing operations '*' and '+', parentheses, and strings of 0's and 1's derived from variable N .

Example 3. Grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$ and P contains productions

$$S \longrightarrow aSbS \mid bSaS \mid \varepsilon$$

What is $L(G)$?

Example 3. Grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$ and P contains productions

$$S \longrightarrow aSbS \mid bSaS \mid \varepsilon$$

What is $L(G)$?

Clearly every word that belongs to $L(G)$ must contain equally many a 's and b 's. (Every production adds the same number of a 's and b 's to the sentential form.)

Let us prove (on the blackboard) that $L(G)$ contains all words $w \in \{a, b\}^*$ with equally many a 's and b 's, *i.e.*, $L(G)$ is the language

$$L = \{w \in \{a, b\}^* \mid w \text{ has equally many } a\text{'s and } b\text{'s} \}$$

Example 3. Grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$ and P contains productions

$$S \longrightarrow aSbS \mid bSaS \mid \varepsilon$$

What is $L(G)$?

Clearly every word that belongs to $L(G)$ must contain equally many a 's and b 's. (Every production adds the same number of a 's and b 's to the sentential form.)

Let us prove (on the blackboard) that $L(G)$ contains all words $w \in \{a, b\}^*$ with equally many a 's and b 's, *i.e.*, $L(G)$ is the language

$$L = \{w \in \{a, b\}^* \mid w \text{ has equally many } a\text{'s and } b\text{'s} \}$$

For example, the word *ababba* has the derivation:

$S \Rightarrow$

Example 3. Grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$ and P contains productions

$$S \longrightarrow aSbS \mid bSaS \mid \varepsilon$$

What is $L(G)$?

Clearly every word that belongs to $L(G)$ must contain equally many a 's and b 's. (Every production adds the same number of a 's and b 's to the sentential form.)

Let us prove (on the blackboard) that $L(G)$ contains all words $w \in \{a, b\}^*$ with equally many a 's and b 's, *i.e.*, $L(G)$ is the language

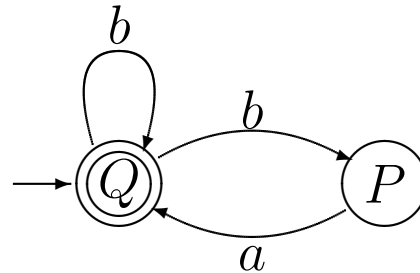
$$L = \{w \in \{a, b\}^* \mid w \text{ has equally many } a\text{'s and } b\text{'s} \}$$

For example, the word *ababba* has the derivation:

$$S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow ababbSaS \Rightarrow ababbaS \Rightarrow ababba$$

Let us prove that every regular language can be generated by a context-free grammar.

Example. Consider the language L recognized by the NFA A



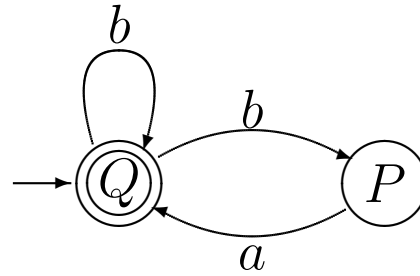
Let us construct a grammar G that generates L . The variables of G are the states $\{Q, P\}$ and the terminals are $\{a, b\}$. The start symbol is the initial state Q . Every transition of A is simulated by a production:

$$\begin{aligned} Q &\longrightarrow bQ \\ Q &\longrightarrow bP \\ P &\longrightarrow aQ \end{aligned}$$

For each transition $q \xrightarrow{x} p$ in A we have the production $q \longrightarrow xp$ in G

So whenever the automaton reads a letter x and goes to state p the grammar produces the same letter x and changes the variable to the current state p .

Example continues.



$$\begin{aligned} Q &\longrightarrow bQ \\ Q &\longrightarrow bP \\ P &\longrightarrow aQ \end{aligned}$$

Derivations by G correspond step-by-step to computations in A :

In A : $Q \xrightarrow{b} Q \xrightarrow{b} P \xrightarrow{a} Q$

In G : $Q \Rightarrow bQ \Rightarrow bbP \Rightarrow bbaQ$

In general, $q \xrightarrow{w} p$ in A if and only if So $q \Rightarrow^* wp$ in G .

To terminate the simulation we add the transition

$$Q \longrightarrow \varepsilon$$

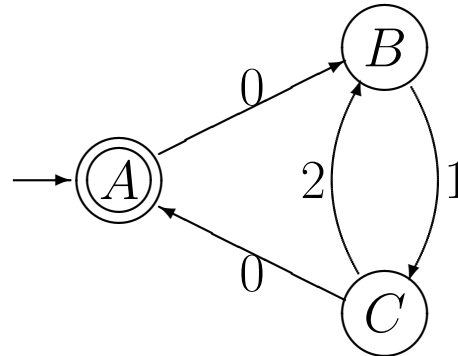
for the final state Q . Accepting computations $Q \xrightarrow{w} Q$ now have corresponding terminating derivations $Q \Rightarrow^* wQ \Rightarrow w$.

The construction can be done on any NFA, proving the following theorem:

Theorem. Every regular language is a context-free language.

Proof.

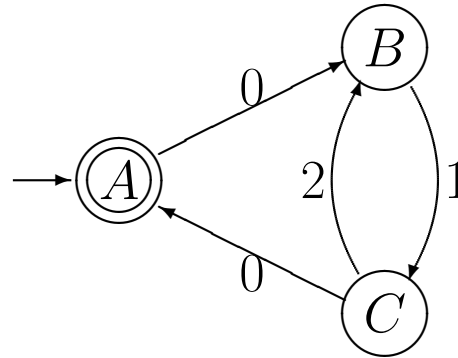
Example. Let us construct a grammar for the language recognized by



$G = (V, T, P, S)$ where

- variables are $V =$
- terminals are $T =$
- the start symbol is $S =$
- the productions are

Example. Let us construct a grammar for the language recognized by



$G = (V, T, P, S)$ where

- variables are $V = \{A, B, C\}$
- terminals are $T = \{0, 1, 2\}$
- the start symbol is $S = A$
- the productions are

$A \longrightarrow 0B$

$B \longrightarrow 1C$

$C \longrightarrow 2B$

$C \longrightarrow 0A$

$A \longrightarrow \varepsilon$

A grammar is called **right linear** if all productions are of forms

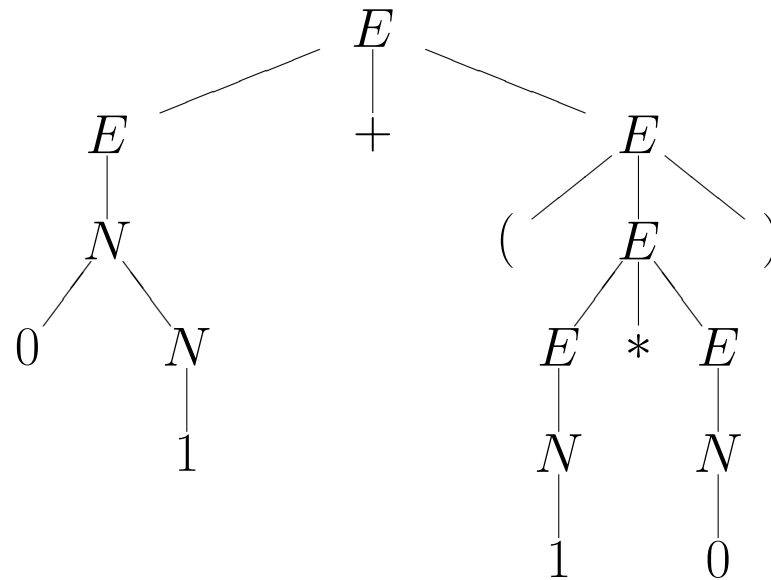
$$\begin{aligned} A &\longrightarrow wB, \\ A &\longrightarrow w \end{aligned}$$

for terminal words $w \in T^*$. In other words, the right hand side of a production may contain only one variable, and it has to be the last symbol.

Our construction always produces a right linear grammar, so every regular language is generated by a right linear grammar. The converse is also true: a language generated by a right linear grammar is always regular.

Derivation trees

Derivations by context-free grammars can be visualized using **derivation trees**, also called **parse trees**. These are rooted, oriented trees. Here is a derivation tree for one of our sample grammars:



All interior nodes are labeled using variables. The label of the root is the start symbol E of the grammar. The children of a node are the letters on the right-hand-side of some production for the parent's variable.

For example, the root has children E , $+$, E , in this order, corresponding to the production $E \longrightarrow E + E$ in the grammar.

A **derivation tree** for grammar $G = (V, T, P, S)$ is a tree whose nodes are labeled by symbols from the set

$$V \cup T \cup \{\varepsilon\}$$

in such a way that

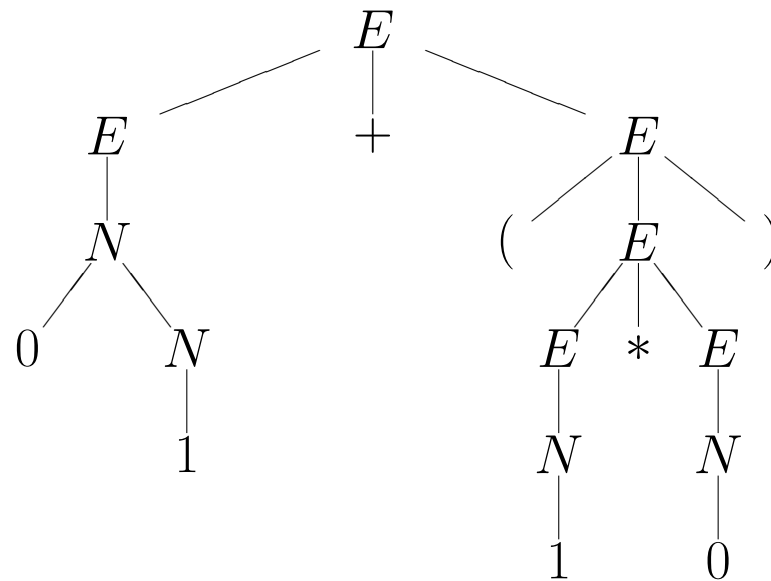
- interior nodes are labeled by variables, *i.e.*, elements of V ,
- the root is labeled by the start symbol S ,
- if $X_1, X_2, \dots, X_n$ are the labels of the children of a node labeled by variable A , ordered from left to right, then

$$A \longrightarrow X_1 X_2 \dots X_n$$

is a production in P . If $n \geq 2$ then all $X_i \neq \varepsilon$.

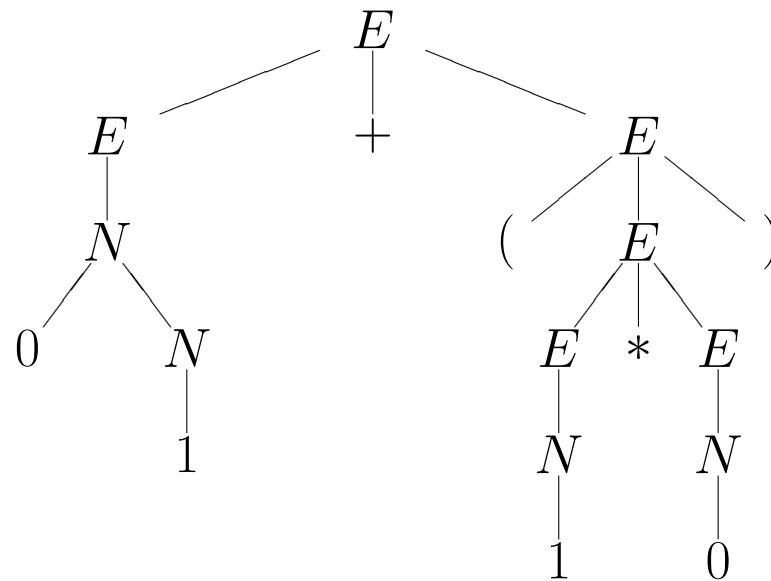
Note that label ε is needed to allow productions $A \longrightarrow \varepsilon$ in the derivation tree.

If one reads the labels of the **leaves** from left to right, one obtains a word over $(V \cup T)^*$. This word is called the **yield** of the derivation tree. For example, the yield of



is $01 + (1 * 0)$.

If one reads the labels of the **leaves** from left to right, one obtains a word over $(V \cup T)^*$. This word is called the **yield** of the derivation tree. For example, the yield of



is **01 + (1 * 0)**.

The yield is always a sentential form, and every sentential form is a yield of a derivation tree (obvious?)

Remark: the yield is obtained by the depth first traversal of the tree, when the subtrees of each node are processed in the left-to-right order, outputting the symbols at all encountered leaves. A program to print the yield of a tree rooted at node x :

Yield(x)

Begin

 if x is a leaf then print the label of x

 else

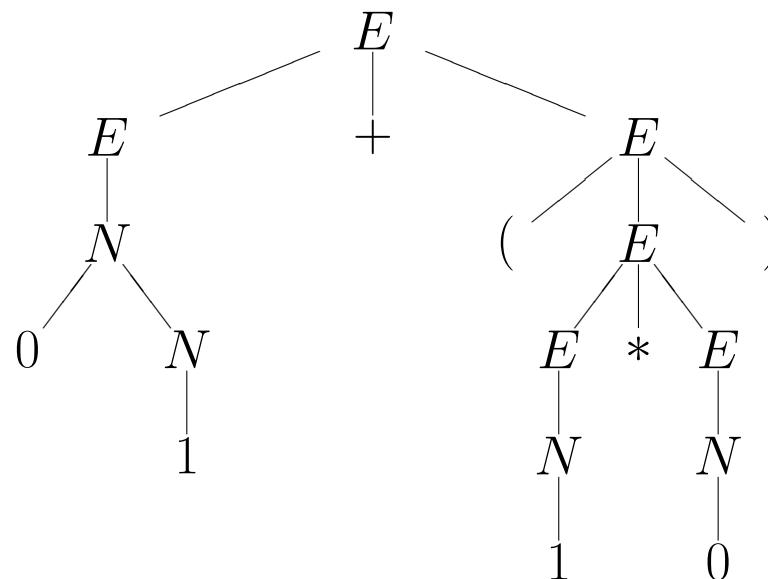
 begin

 let $x_1, \dots, x_n$ be the children of x in the left-to-right order

 for $i=1$ to n call Yield(x_i)

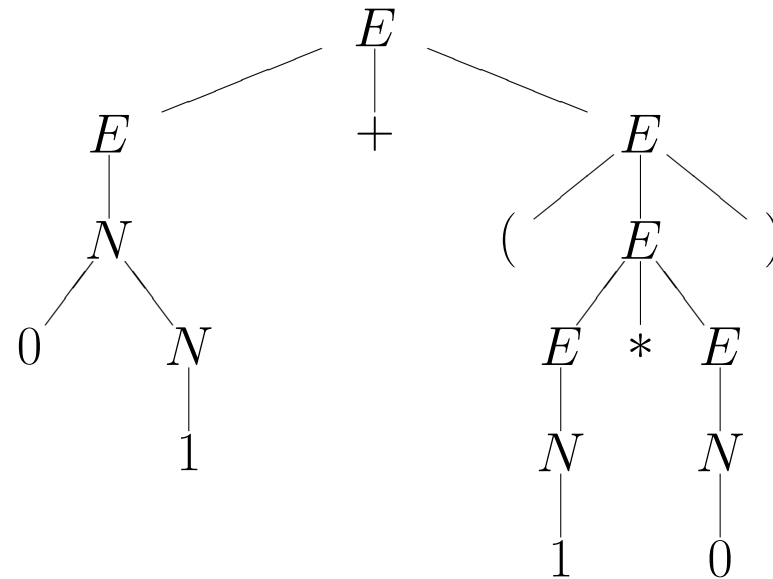
 end

End

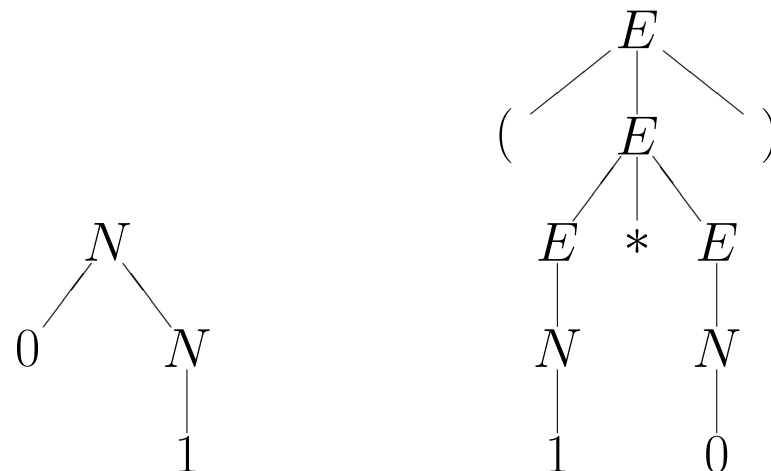


A **subtree** of a derivation tree is the tree formed by some node and all its descendants. A subtree is like a derivation tree except that its root may have a different label than the start symbol. Such a tree is called an **A-tree**, where A is the label of the root.

Example.



has, as subtrees, the following N -tree and E -tree



Theorem. Let $G = (V, T, P, S)$ be a context-free grammar. Then $A \Rightarrow^* \alpha$, for some $A \in V$ and $\alpha \in (V \cup T)^*$, if and only if α is the yield of an A -tree.

In particular, $w \in L(G)$ if and only if w is the yield of a derivation tree and w contains only terminal letters.

Proof. An easy induction on the length of the derivation/size of the derivation tree.

Example. Let us construct the derivation tree corresponding to the derivation

$$\begin{aligned} E &\Rightarrow E + E \Rightarrow N + E \Rightarrow 1 + E \Rightarrow 1 + E * E \\ &\Rightarrow 1 + N * E \Rightarrow 1 + 1N * E \Rightarrow 1 + 10 * E \\ &\Rightarrow 1 + 10 * N \Rightarrow 1 + 10 * 1 \end{aligned}$$

Example. Let us construct the derivation tree corresponding to the derivation

$$\begin{aligned} E &\Rightarrow E + E \Rightarrow N + E \Rightarrow 1 + E \Rightarrow 1 + E * E \\ &\Rightarrow 1 + N * E \Rightarrow 1 + 1N * E \Rightarrow 1 + 10 * E \\ &\Rightarrow 1 + 10 * N \Rightarrow 1 + 10 * 1 \end{aligned}$$

Note that the same tree represents several derivations. For example, the derivation

$$\begin{aligned} E &\Rightarrow E + E \Rightarrow E + E * E \Rightarrow E + N * E \\ &\Rightarrow N + N * E \Rightarrow N + 1N * E \Rightarrow N + 1N * N \\ &\Rightarrow 1 + 1N * N \Rightarrow 1 + 1N * 1 \Rightarrow 1 + 10 * 1 \end{aligned}$$

has the same derivation tree. The two derivations only differ in the order of rewriting the variables.

Example. Let us construct the derivation tree corresponding to the derivation

$$\begin{aligned} E &\Rightarrow E + E \Rightarrow N + E \Rightarrow 1 + E \Rightarrow 1 + E * E \\ &\Rightarrow 1 + N * E \Rightarrow 1 + 1N * E \Rightarrow 1 + 10 * E \\ &\Rightarrow 1 + 10 * N \Rightarrow 1 + 10 * 1 \end{aligned}$$

Note that the same tree represents several derivations. For example, the derivation

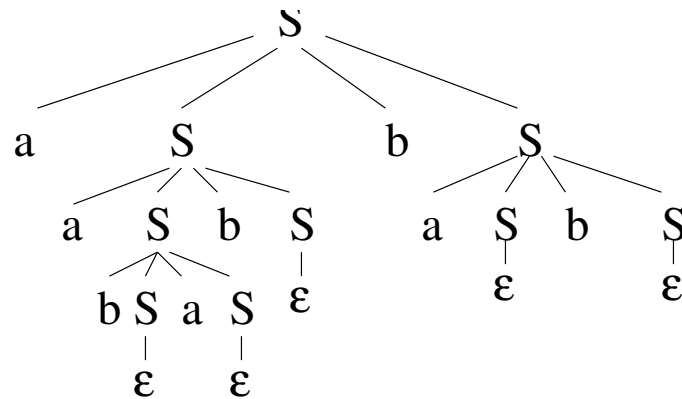
$$\begin{aligned} E &\Rightarrow E + E \Rightarrow E + E * E \Rightarrow E + N * E \\ &\Rightarrow N + N * E \Rightarrow N + 1N * E \Rightarrow N + 1N * N \\ &\Rightarrow 1 + 1N * N \Rightarrow 1 + 1N * 1 \Rightarrow 1 + 10 * 1 \end{aligned}$$

has the same derivation tree. The two derivations only differ in the order of rewriting the variables.

A derivation is called **leftmost** if at every derivation step the leftmost variable of the sentential form is rewritten. The first derivation is leftmost, and the second one is not. **Rightmost** derivations are defined analogously: One always replaces the rightmost variable.

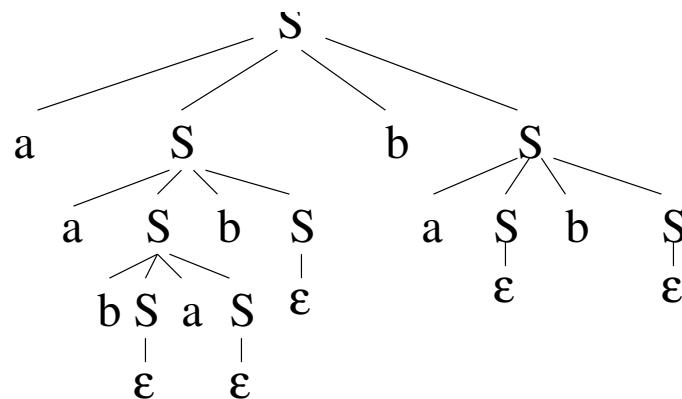
Every **derivation tree defines a unique leftmost derivation** (and a unique rightmost derivation). The leftmost derivation rewrites the variables in the order in which the depth first traversal of the tree from left to right encounters them. (The rightmost derivation corresponds to the depth first traversal from right to left.)

Example. Let us write the leftmost derivation corresponding to the derivation tree



Every **derivation tree defines a unique leftmost derivation** (and a unique rightmost derivation). The leftmost derivation rewrites the variables in the order in which the depth first traversal of the tree from left to right encounters them. (The rightmost derivation corresponds to the depth first traversal from right to left.)

Example. Let us write the leftmost derivation corresponding to the derivation tree



$S \Rightarrow aSbS \Rightarrow aaSbSbS \Rightarrow aabSaSbSbS \Rightarrow aabaSbSbS \Rightarrow aababSbS \Rightarrow aababbS \Rightarrow aababbaSbS \Rightarrow aababbabS \Rightarrow aababbabS$

Conclusion: There is a one-to-one correspondence between derivation trees and leftmost (rightmost) derivations.