Limitations of Context-free Languages

- Context-free grammars and pushdown automata are not able to generate or accept all possible languages.
- Some examples of languages for which it can be proved that they are not context-free (i.e., that there is no context-free grammar generating them):

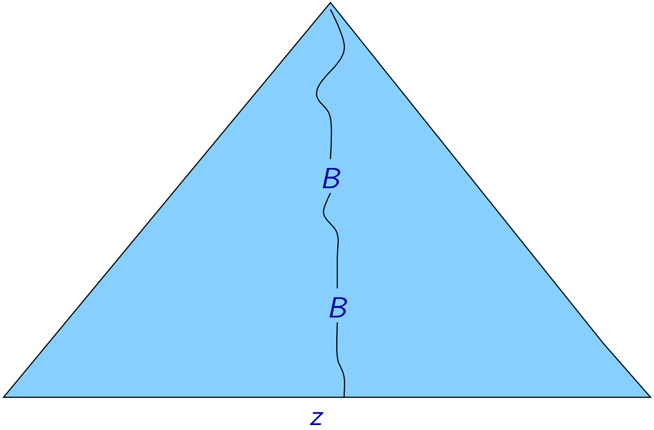
  $L_1 = \{a^n b^n c^n \mid n \geq 0\}$
  $L_2 = \{ww \mid w \in \{a, b\}^*\}$

Pumping Lemma for Context-free Languages

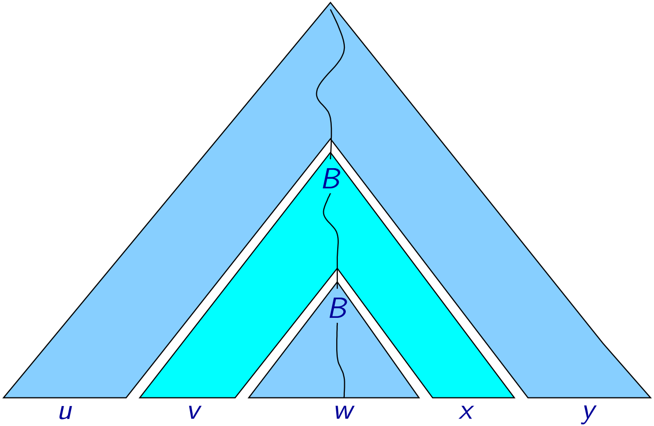
Consider some arbitrary context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$:

- Consider an arbitrary word $z \in \mathcal{L}(\mathcal{G})$.
- For a derivation of word z using grammar $\mathcal{G}$, there is a corresponding derivation tree.
- Consider individual branches of this derivation tree.
- It is possible that on some branch of the tree, some nonterminal $B \in \Pi$ occurs at least two times.
- Assume now, that the given tree contains at least one such branch with a repeated occurrences of a nonterminal.

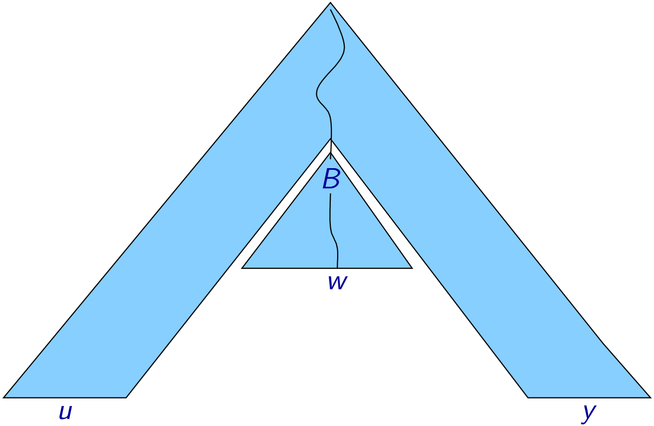
Pumping Lemma for Context-free Languages



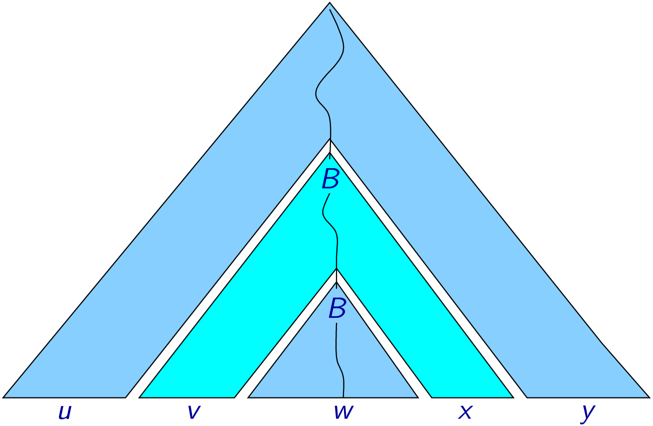
Pumping Lemma for Context-free Languages



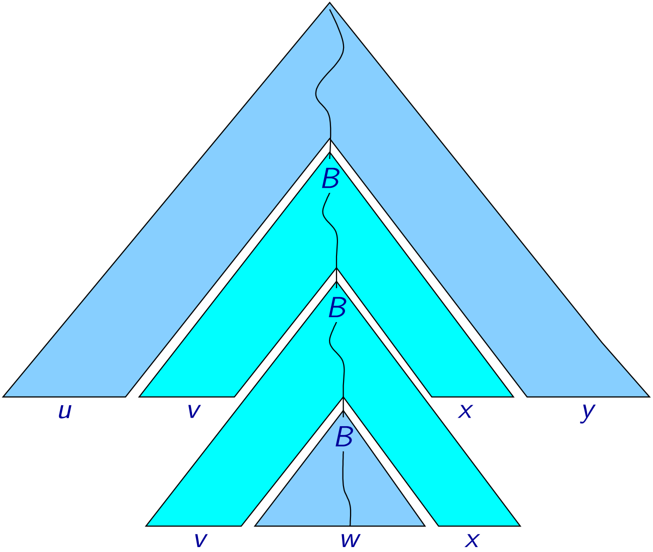
Pumping Lemma for Context-free Languages



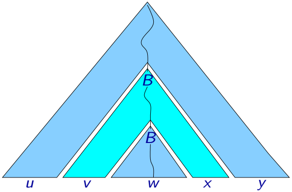
Pumping Lemma for Context-free Languages



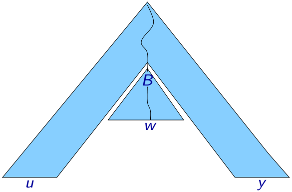
Pumping Lemma for Context-free Languages



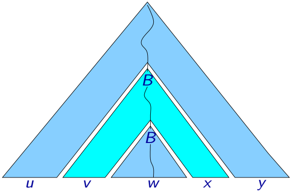
Pumping Lemma for Context-free Languages



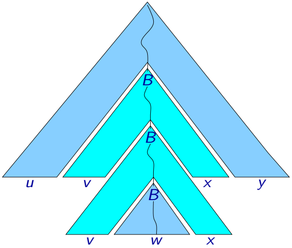
Pumping Lemma for Context-free Languages



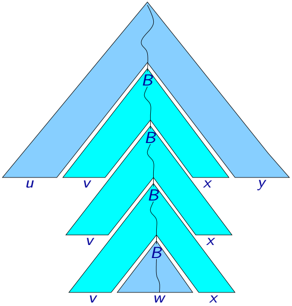
Pumping Lemma for Context-free Languages



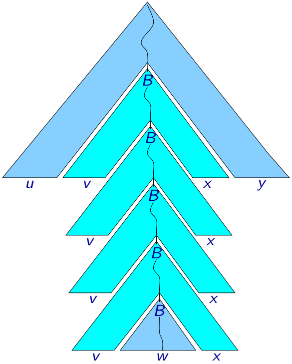
Pumping Lemma for Context-free Languages



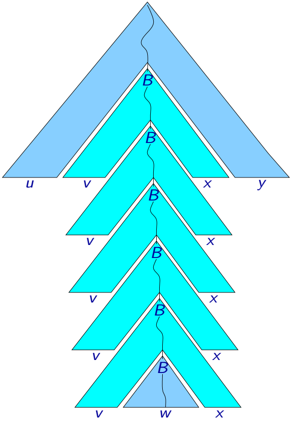
Pumping Lemma for Context-free Languages



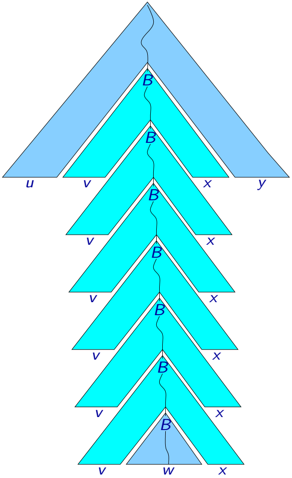
Pumping Lemma for Context-free Languages



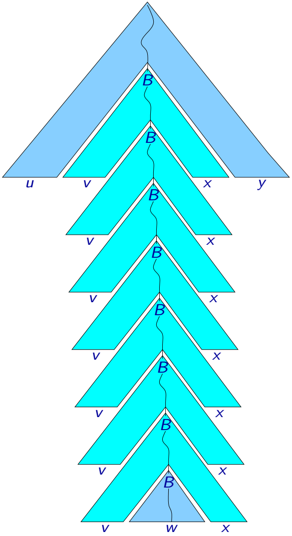
Pumping Lemma for Context-free Languages



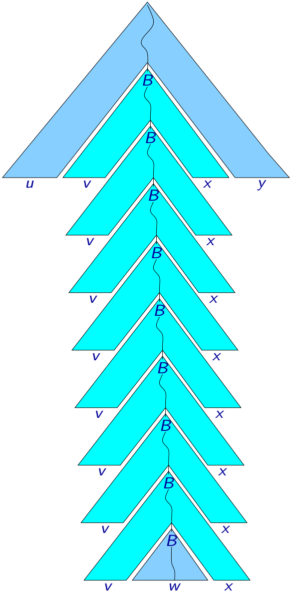
Pumping Lemma for Context-free Languages



Pumping Lemma for Context-free Languages



Pumping Lemma for Context-free Languages



Pumping Lemma for Context-free Languages

We can see that if the given derivation tree representing a derivation of a word z contains some branch where some nonterminal is repeated at least twice, then the word z could be decomposed into five subwords u, v, w, x, y such that:

- all words of the form uv^iwx^iy, where $i \geq 0$,
  i.e., words uwy, $uvwxy$, $uvvwxxy$, $uvvvwxxxy$, $uvvvvwxxxxy$, ...
  also belong to the language $\mathcal{L}(\mathcal{G})$.

Pumping Lemma for Context-free Languages

Of course, in general a given derivation tree representing a derivation of word z need not contain a branch where some nonterminal occurs at least twice.

However, in this case the word z can not be too long:

- Let k be the number of nonterminal in the given grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$, i.e., $k = |\Pi|$.
- Obviously, every branch then contains at most k nonterminals and one terminal.
- So the length of each branch is at most k.
- The number of children of each node is at most the length of the longest right-hand side of a rule from the set P.

  Let ℓ be the length of this longest right-hand side, i.e.,

$$\ell = \max\{ |\alpha| \mid (A \rightarrow \alpha) \in P \}$$

Pumping Lemma for Context-free Languages

- It hold in general that a tree with branches of length k where every node has at most ℓ childen, can have at most ℓ^k leafs.
- So we see that the length of the word z can be at most ℓ^k.

It follows from this that for a given grammar $\mathcal{G}$ there exists a constant p such that:

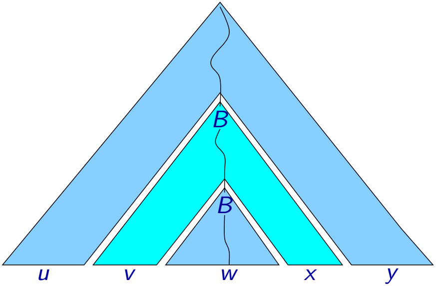
- for every word $z \in \mathcal{L}(\mathcal{G})$, such that $|z| \geq p$, it holds that a derivation tree representing a derivation of the word z in grammar $\mathcal{G}$ must contain at least one branch where some nonterminal occurs at least twice.

We can see that this constant can be computed for the given grammar $\mathcal{G}$. In particular, we can put $p = \ell^k + 1$.

Pumping Lemma for Context-free Languages

Moreover, we can choose a branch, a repeated nonterminal B, and two particular occurrences of this nonterminal B in such way that it holds that:

- at least one of words v and x is nonempty, i.e., $|vx| \geq 1$,
- the total length of words v, w, x is bounded from above by some constant q, i.e., $|vwx| \leq q$, and the value of this constant q depends only on the grammar $\mathcal{G}$, not on a particular word z.



It is obvious that for the given word z we can find such derivation tree where for every subtree holds that:

- if a given subtree has a root labelled with nonterminal B and the given subtree contains one more occurrence of the nonterminal B, then the word generated by the subtree with the root in this second occurrence is shorter than the word generated by the whole subtree.

Pumping Lemma for Context-free Languages

In such tree, we can choose such nonterminal B and such subtree with a root labelled with B satisfying the following:

- The given subtree contains at least one other occurrence on the nonterminal B.
- None of the branches of the subtree contains nonterminal B more than twice
  — once in the tree of the subtree and at most one additional occurrence.
- No other nonterminal occurs twice on any of branches of the subtree.

It is obvious that such subtree satisfies the following:

- the length of all its branches is at most $k + 1$

So the given subtree has at most ℓ^{k+1} leafs.

We can put constant q equal to ℓ^{k+1}.

Pumping Lemma for Context-free Languages

So we have proven the following proposition:

Pumping lemma (version 1)

If language L is context-free then
there exist constants p and q such that
for every word $z \in L$ such that $|z| \geq p$
there exist words u, v, w, x, y such that
$z = uvwxy$, $|vx| \geq 1$, $|vwx| \leq q$, and
for each $i \geq 0$ it holds that $uv^iwx^iy \in L$.

Pumping Lemma for Context-free Languages

We can note that if the constants p and q exist then the given condition also holds for any bigger values.

So intead of two values p and q we can consider just one value $n = \max\{p, q\}$, and to simplify the formulation of the pumping lemma a little bit:

Pumping lemma (version 2)

If language L is context-free then
  there exists a number $n \in \mathbb{N}$ such that
    for every word $z \in L$ such that $|z| \geq n$
      there exist words u, v, w, x, y such that
        $z = uvwxy$, $|vx| \geq 1$, $|vwx| \leq n$, and
        and for each $i \geq 0$ it holds that $uv^iwx^iy \in L$.

Pumping Lemma for Context-free Languages

Deciding whether the given proposition holds can be viewed as a game played by two players:

1. Player I chooses $n \in \mathbb{N}$.
2. Player II chooses a word $z \in L$ such that $|z| \geq n$.
3. Player I chooses words $u, v, w, x, y \in \Sigma^*$ such that $z = uvwxy$, $|vx| \geq 1$, and $|vwx| \leq n$.
4. Player II chooses $i \in \mathbb{N}$.
5. If $uv^iwx^iy \in L$ then Player I wins, otherwise Player II wins.

If L is context-free then Player I has a winning strategy. (So if Player II has a winning strategy, then L is not context-free.)

Pumping Lemma for Context-free Languages

Example: Language $L = \{a^n b^n c^n \mid n \geq 0\}$.

- Player I chooses $n \in \mathbb{N}$.
- Player II chooses word $a^n b^n c^n$.
- Player I chooses words $u, v, w, x, y \in \Sigma^*$ such that $z = uvwxy$, $|vx| \geq 1$, $|vwx| \leq n$.
- Player II chooses $i = 0$.
- Player II wins because the word $z' = uwy$ does not belong to the language L: words v and x necessarily contain at most two from symbols a, b, c. Moreover, at least one of words v and x is nonempty. So it holds in z' that at least one of values $|z'|_a$, $|z'|_b$ and $|z'|_c$ is strictly smaller than n, and at least one of them is equal to n.

So the language L is not context-free.

Closure Properties of the Class of Context-Free Languages

- We have already seen that the class of context-free languages is closed with respect to the **union**, **concatenation**, and **iteration**, i.e., it holds for all context-free languages L_1 and L_2 that also languages

$$L_1 \cup L_2 \qquad L_1 \cdot L_2 \qquad L_1^*$$

are context-free.

- It is not hard to see that the class of context-free languages is also closed for example with respect to the **reverse** and to the **intersection with a regular language**, i.e., if language L_1 is context-free and language L_2 is regular, then also languages

$$L_1^R \qquad L_1 \cap L_2$$

are context-free.

Closure Properties of the Class of Context-Free Languages

- However, context-free languages are not closed with respect to the **intersection**:

  Consider languages

  $$L_1 = \{a^n b^n c^k \mid n, k \geq 0\} \qquad L_2 = \{a^k b^n c^n \mid k, n \geq 0\}$$

  These languages are context-free because $L_1 = \mathcal{L}(\mathcal{G}_1)$ and $L_2 = \mathcal{L}(\mathcal{G}_2)$:

  $\mathcal{G}_1$:

  $$\begin{aligned} S_1 &\to DC \\ D &\to \varepsilon \mid aDb \\ C &\to \varepsilon \mid cC \end{aligned}$$

  $\mathcal{G}_2$:

  $$\begin{aligned} S_2 &\to AE \\ A &\to \varepsilon \mid aA \\ E &\to \varepsilon \mid bEc \end{aligned}$$

  It is obvious that

  $$L_1 \cap L_2 \;=\; \{a^n b^n c^n \mid n \geq 0\}$$

  We have already seen that this language is not context-free.

Closure Properties of the Class of Context-Free Languages

- It follows from the previous that context-free languages are not closed with respect to the **complement**:

  If context-free languages would be closed with respect to the complement, they would be also closed with respect to the intersection because

$$L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$$

Closure Properties of the Class of Context-Free Languages

An example of a language, which is context-free, but whose complement is not context-free:

the complement of the language $\{a^n b^n c^n \mid n \geq 0\}$

This complement is context-free since it can be represented as the union of three context-free languages:

- L_1 — the complement of the regular language $a^* b^* c^*$
- $L_2 = \{a^m b^n c^p \mid m, n, p \geq 0,\ m \neq n\}$
- $L_3 = \{a^m b^n c^p \mid m, n, p \geq 0,\ n \neq p\}$

For example, the language L_2 is generated by the following grammar:

$$
\begin{aligned}
S &\rightarrow ADC \mid DBC \\
A &\rightarrow a \mid aA \\
B &\rightarrow b \mid bA \\
C &\rightarrow \varepsilon \mid cC \\
D &\rightarrow \varepsilon \mid aDb
\end{aligned}
$$

Deterministic Context-Free Languages

Definition

A language L is a **deterministic context-free language** if it is accepted by a deterministic pushdown automaton.

Remark: In the above definition, we can consider both automata accepting by an accepting state and automata accepting by an empty stack where a special endmarker $\dashv$ is added at the end of a word on its input tape.

(We have already seen that both these types of automata can be easily transformed into each other.)

Deterministic Context-Free Languages

The class of deterministic context-free languages is closed with respect to the **complement**.

But it is not closed with respect to the **intersection**:

- Languages

  $$L_1 = \{a^n b^n c^k \mid n, k \geq 0\} \qquad L_2 = \{a^k b^n c^n \mid k, n \geq 0\}$$

  are deterministic context-free languages.

  Their intersection is the language

  $$L = \{a^n b^n c^n \mid n \geq 0\}$$

  that is not even context-free (so obviously, it is not deterministic context-free).

Deterministic Context-Free Languages

- It immediately follows from the previous discussion that deterministic context-free languages are not closed with respect to the **union**:

  If context-free languages would be closed with respect to the union, they would be also closed with respect to the intersection, since

$$L_1 \cap L_2 \;=\; \overline{(\overline{L_1} \cup \overline{L_2})}$$

- However, deterministic context-free languages are closed with respect to both the **intersection** and the **union** with a **regular language**.

  I.e., if language L_1 is deterministic context-free and language L_2 is regular then also languages

$$L_1 \cap L_2 \qquad\qquad L_1 \cup L_2$$

  deterministic context-free.

Example: The following two languages are deterministic context-free (and so they are also context-free):

- $L_1 = \{a^m b^n c^p \mid m, n, p \geq 0,\ m \neq n\}$
- $L_2 = \{a^m b^n c^p \mid m, n, p \geq 0,\ n \neq p\}$

Their union is the language

$$L_3 = \{a^m b^n c^p \mid m, n, p \geq 0,\ (m \neq n) \vee (n \neq p)\}$$

It is obvious that language L_3 is context-free.

Deterministic Context-Free Languages

But language L_3 is not deterministic context-free:

- Let us assume that L_3 would be deterministic context-free.
- Then also the language $L_5 = L_3 \cup L_4$, is the complement of the language represented by regular expression $a^* b^* c^*$, would be also deterministic context-free.
- However, this would mean that also the complent of the language L_5 is deterministic context-free. But this is not possible, since this complement is the language

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

  which is not even context-free.

Deterministic Context-Free Languages

Deterministic context-free languages are also not closed with respect to the **reverse**.

Example: It is not difficult to see that the followin language L over the alphabet $\Sigma = \{a, b, c, d, e\}$ is deterministic context-free:

$$L = \{ d a^n b^n c^k \mid n, k \geq 0 \} \cup \{ e a^k b^n c^n \mid n, k \geq 0 \}$$

It can be shown that the reverse of this language, i.e., the language

$$L^R = \{ c^k b^n a^n d \mid n, k \geq 0 \} \cup \{ c^n b^n a^k e \mid n, k \geq 0 \}$$

is not deterministic context-free.

Nondeterministic and Deterministic Pushdown Automata

Additional remarks concerning nondeterministic and deterministic pushdown automata:

- For every nondeterministic pushdown automaton it is possible to construct an equivalent nondeterministic pushdown automaton **with one state** of the control unit.

  This is not the case for deterministic pushdown automata.

- For every nondeterministic pushdown automaton it is possible to contruct an equivalent nondeterministic pushdown automaton **without ε-transitions**.

  This is not the case for deterministic pushdown automata.