# TDDD14/TDDD85 Lecture 11: Closure Properties and Pumping Lemma

Jonas Wallgren

**Abstract**

In this lecture some closure properties of CFLs will be presented. The pumping lemma for CFLs will be presented and exemplified.

## 1 Introduction

In lecture 4 some closure properties for regular languages were presented and in lecture 6 the pumping lemma for regular languages was discussed.

In this lecture corresponding features for context-free languages will be discussed. First some closure properties are presented, then the pumping lemma, and finally some more closure properties that need the pumping lemma for their proofs.

## 2 Some closure properties

Let in this section the two languages $L_1$ and $L_2$ have the grammars
$G_1 = \langle N_1, \Sigma_1, P_1, S_1 \rangle$ and $G_2 = \langle N_2, \Sigma_2, P_2, S_2 \rangle$.

**Theorem 1.** *CFLs are closed under union*[1].

*Proof.* Construct a new grammar:
$G_3 = \langle N_1 \cup N_2 \cup \{S_3\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S_3 \to S_1 | S_2\}, S_3 \rangle$.
Now, starting from $S_3$ all derivations starting with either $S_1$ or $S_2$ can be performed, i.e. all strings in $L(G_1)$ and $L(G_2)$ can be derived, i.e.
$L(G_3) = L(G_1) \cup L(G_2)$. □

**Theorem 2.** *CFLs are closed under concatenation.*

*Proof.* Construct a new grammar:
$G_4 = \langle N_1 \cup N_2 \cup \{S_4\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S_4 \to S_1 S_2\}, S_4 \rangle$.
Now, starting from $S_4$ all derivations consist of one part derived from $S_1$ in $L(G_1)$ followed by one part derived from $S_2$ in $L(G_2)$, i.e. $L(G_4) = L(G_1)L(G_2)$. □

[1] This is an alternative, common, way to express that the set of CFLs is closed under union. It does not express a property of every single CFL.

**Theorem 3.** *CFLs are closed under the star operation.*

*Proof.* Construct a new grammar:
$G_5 = \langle N_1 \cup \{S_5\}, \Sigma_1, P_1 \cup \{S_5 \to S_1S_5|\varepsilon\}, S_5\}$.
Now, starting from $S_5$ either $\varepsilon$ can be derived or one string from $L(G_1)$ followed once again by what can be derived from $S_5$. So, $L(G_5) = (L(G_1))^*$. □

Since a CFL can be defined either by a grammar and by a PDA we could have tried to prove these properties "on the automata side", but a grammar is a more natural tool for specifying a CFL, so our proofs come closer to the natural thinking. But there are always two ways to look at such problems.

# 3 Pumping lemma

We will first motivate the pumping lemma by an illustration. Then we will state the lemma in two forms. Finally some examples will be given.

## 3.1 Motivation

The main idea in the pumping lemma for regular languages—if something very big must fit into something limited then some parts have to loop, and then must be able to loop arbitrarily many times—also holds for the the pumping lemma for CFLs.

See the top figure on the last page. The whole triangle symbolizes the outline of a derivation tree. S is the root of the tree, the start symbol. The sequence uvwxy stands for parts of the derived string, the sequence of leaves of the tree. The squiggly curve down the middle stands for one path down the tree from the root to one single leaf. Even if the grammar is large it is finite. It has a limited number of nonterminals. If a string is very long at least some such path down the tree must encounter the same nonterminal, symbolized with A here, at least twice. It means that in the grammar there must be at least two alternatives for A ($A \to something_{1\_(using\_A)}|something_{2\_(not\_using\_A)}$). Following the squiggly path from S the derivation for the first A used the first grammar rule and the derivation for the second A used the second one.

The middle figure shows the situation when the derivation for the first occurence of A in the tree uses the second rule. Directly the small derivation subtree appears and there is no v and x in the string.

The bottom figure shows the situation when also the derivation for the second A uses the first rule, and the derivation for the third A then appearing uses the second rule. In this case v and x occur twice in the derived string.

## 3.2 Direct form

From the ideas in the figures this form of the pumping lemma can be stated, which will be given here without proof:

$$
\begin{array}{ll|l}
L\ is\ CF \rightarrow & \exists k > 0 : & 1 \\
& \quad \forall z \in L : |z| \geqslant k \rightarrow & 2 \\
& \quad\quad \exists u, v, w, x, y \in \Sigma^* : z = uvwxy \land vx \neq \varepsilon \land |vwx| \leqslant k \rightarrow & 3 \\
& \quad\quad\quad \forall i \geqslant 0 : uv^iwx^iy \in L & 4
\end{array}
$$

Line 1: If L is context-free then there is some constant (pumping length) such that
Line 2: if a string in L is of enough length then
Line 3: it can be splitted into uvwxy—such that vx isn't empty and uvw isn't too long—so that
Line 4: the vx parts can be repeated arbitrarily number of times and the string still is in L.

## 3.3 Inverted form

Like for regular pumping instead of using the lemma on the form "if a then b" we will use the inverted form "if not b then not a". The pumping lemma in this form follows this reasoning:
If this test holds
 Line 1: For all constants (pumping lengths)
 Line 2: there is a string of enough length and
 Line 3: for all ways to split it into uvwxy—such that vx isn't empty and uvw isn't too long—
 Line 4: you can pump the vx parts to get the string outside of the language.
then the language is not context-free.

**Lemma 1.** *(Inverted pumping lemma for CFLs)*

$$
\begin{array}{ll|l}
(\forall k > 0 : & & 1 \\
\quad \exists z \in L : |z| \geqslant k \land & & 2 \\
\quad\quad \forall u, v, w, x, y \in \Sigma^* : z = uvwxy \land vx \neq \varepsilon \land |vwx| \leqslant k \rightarrow & & 3 \\
\quad\quad\quad \exists i \geqslant 0 : uv^iwx^iy \notin L) & \rightarrow L\ is\ not\ CF & 4
\end{array}
$$

## 3.4 Examples

Some examples of using the pumping lemma.

**Example 1.** *Prove that $L_6 = \{a^nb^nc^n | n \geqslant 0\}$ is not context-free.*
*For the pumping length k let $z = a^kb^kc^k$.*
*Now, since vwx cannot contain more than k symbols vx cannot contain both a:s, b:s, and c:s. They have the same number of occurences in z but not all of them will change when pumping so the pumped string will be out of $L_6$. E.g. if $u = a^p, v = a^q, w = a^r, x = a^s, y = a^{k-p-q-r-s}b^kc^k$, then pumping to $uv^2wx^2y$ will give the string $a^{k+q+s}b^kc^k$. So $L_6$ is not context-free.*

**Example 2.** *Prove that the language $L_7 = \{a^mb^na^mb^n | m, n \geqslant 0\}$ is not context-free.*
*For the pumping length k let $z = a^kb^ka^kb^k$.*
*I'll call each of $a^k$ and $b^k$ a* block.

- *If $v$ or $x$ contains both $a$ and $b$ then pumpig up ($uv^2wx^2y$) will in $v^2$ or $x^2$ result in a mixture of two symbols and $z$ will break the original $a^*b^*a^*b^*$ pattern in $L_7$.*
- *If $v$ and $x$ are in the same block then pumping up ($uv^2wx^2y$) will make that block longer than the other block with the same symbol.*
- *If $v$ and $x$ are in neighbouring blocks then there will be different numbers of both a:s and b:s in the blocks. E.g. if $u = a^p, v = a^q, w = a^{k-p-q}b^r, x = b^s, y = b^{k-r-s}a^kb^k$ then $uv^2wx^2y = a^{k+q}b^{k+s}a^kb^k$.*

*So, for every possible split of $z$ it is pumpable out of $L_7$. $L_7$ is not context-free.*

# 4 Some more closure properties

In this section we will introduce a number of languages ($L_8, L_9, \ldots$), their grammars($G_8, G_9, \ldots$), and a number of automata ($M_{10}, M_{11}, \ldots$)

**Theorem 4.** *CFLs are not closed under intersection.*
*Let $L_8 = \{a^mb^mc^n | m, n \geqslant 0\}$ and $L_9 = \{a^mb^nc^n | m, n \geqslant 0\}$. Both languages are context-free. They can be defined with the following grammars:*

$$G_8: \quad S \to AC \qquad G_9: \quad S \to AC$$
$$A \to aAb|\varepsilon \qquad A \to aA$$
$$C \to cC|\varepsilon \qquad C \to bCc|\varepsilon$$

*Now, $L_8 \cap L_9 = \{a^nb^nc^n | n \geqslant 0\}$*
*Since $L_8$ and $L_9$ are context-free and the result was proved not context-free in* **Example 1** *context-free languages are not closed under intersection.*

**Theorem 5.** *CFLs are closed under intersection with regular languages.*

*Proof.* Let CFL $L_{10}$ be accepted by a PDA $M_{10}$ and regular $L_{11}$ by a DFA $M_{11}$. A PDA $M_{12}$ can be constructed where each state is a pair of one state in $M_{10}$ and one state in $M_{11}$. The transitions in $M_{12}$ follow simultaneously the transitions in $M_{10}$ and $M_{11}$. A string is accepted in $M_{12}$ if it is accepted in both $M_{10}$ and $M_{11}$. The stack is handled as in $M_{10}$—since $M_{11}$ doesn't use the stack the $M_{10}$ stack handling won't be disturbed. □

**Theorem 6.** *CFLs are not closed under complement.*

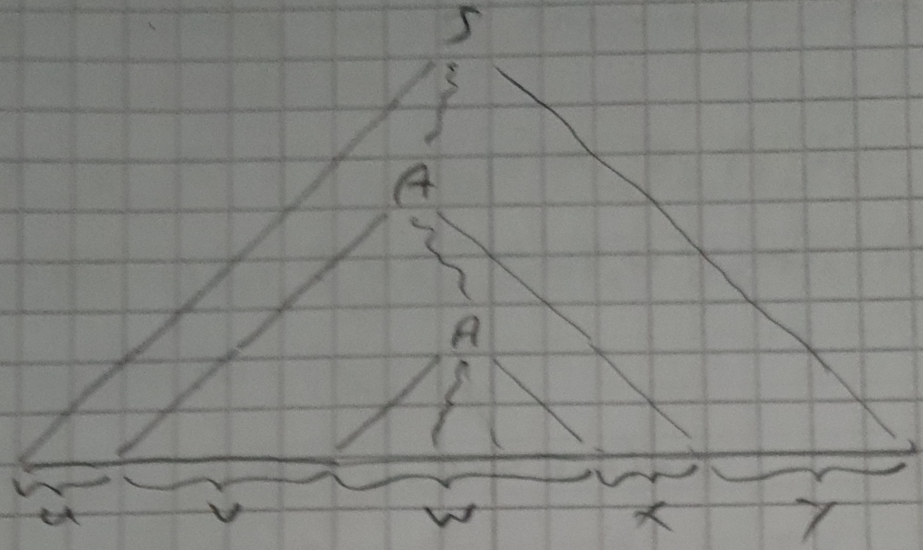
*Proof.* $L_{13} \cap L_{14} = \sim (\sim L_{13} \cup \sim L_{14})$
If CFLs were closed under complement then the union of two CFLs would have been constructed, which would be context-free, and then the complement of that result should have been context-free. But the result is the intersection that, according to **Theorem 4**, does not preserve context-freeness. □

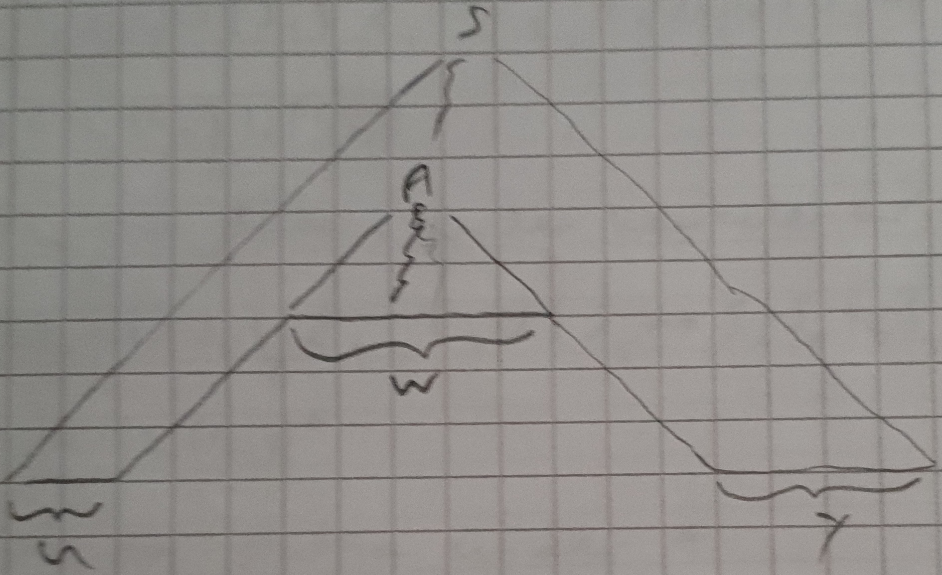
**Example 3.** *$L_{15} = \{ww | w \in \{a, b\}^*\}$ is not context-free.*
*($L_{15}$ contains strings consisting of two equal parts, e.g. aabaab, ababaababa, and bbbbabbbba.)*

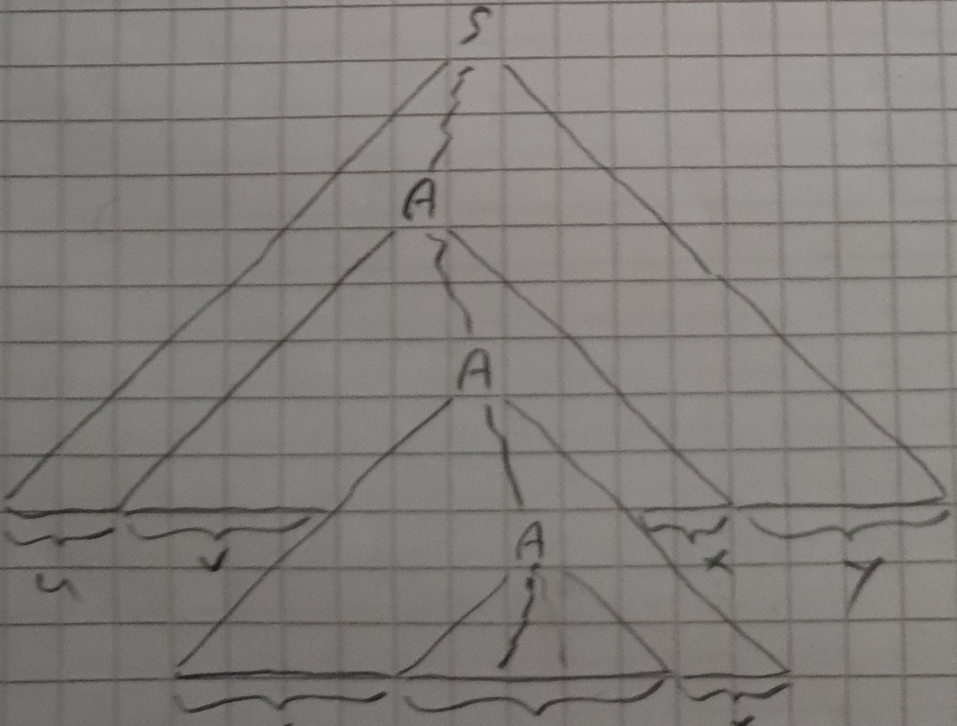
*Some of the strings in* $L_{15}$ *are the same as some strings of the form* $a^*b^*a^*b^*$. $L_{15} \cap a^*b^*a^*b^* = L_7$, *with* $L_7$ *from* **Example 2**. *Since CFLs are closed under intersection with regular languages* $L_{15}$ *would be context-free if* $L_{15}$ *was context-free.*

# 5 More to think about

1. Can we use the pumping lemma for context-free languages to prove that a given language is context-free?
2. Can every language which is not context-free be proven so by using the pumping lemma introduced in this lecture?


S
A
A
u
v
w
x
y


S
A
w
u
y


S
A
A
A
u
v
x
y
v
w
x