IN2080

Non-Context-Free Languages

Einar Broch Johnsen

Universitetet i Oslo

1st March 2022



Department of Informatics



University of Oslo

Repetition

Definition (Context-Free Grammar)

A *context-free grammar* is a 4-tuple (V, Σ, R, S) where

1. V is a finite set of *variables*
2. Σ is a finite set disjoint from V of *terminals*
3. R is a finite set of *rules*, each consisting of a variable and a string of variables and terminals
4. and S is the *start variable*

A language generated by a context-free grammar is a *context-free language*

Context-Free Grammar

$S \to aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \varepsilon$

Palindromes over $\{a, b, c\}$

$S \to 01S10 \mid \varepsilon$

$\{(01)^n(10)^n \mid n \geq 0\}$

$S \to 1 \mid 1B1 \mid 0 \mid 0B0$
$B \to 1B \mid 0B \mid \varepsilon$

All $\{0, 1\}$ words which begin and end with the same digit

Chomsky Normal Form

Every context-free grammar can be rewritten into Chomsky normal form:

Definition (Chomsky Normal Form)

A grammar in Chomsky normal form consists only of rules of the following form:

$$S \to \varepsilon$$
$$A \to BC$$
$$A \to d$$

where S is the start variable, A, B, C are variables with B, C distinct from S, and d is a terminal.

PDA

A pushdown automaton (PDA) is, informally,
a NFA with an additional stack to store information.

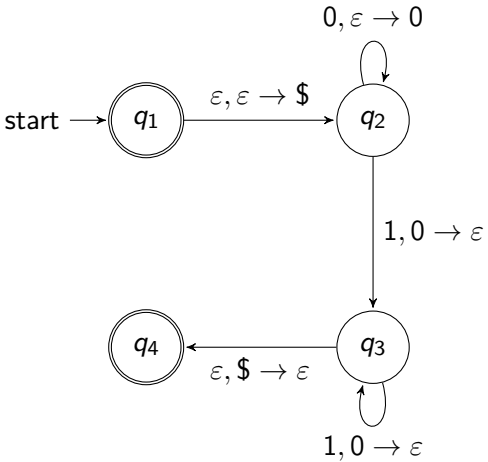
Definition (PDA)

A PDA is a tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where Q, Σ, Γ, F are finite states and

1. Q is a set of states,
2. Σ is the input alphabet,
3. Γ is the stack alphabet,
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is the transition function
5. $q_0 \in Q$ is the start state, and
6. $F \subseteq Q$ is the set of accepting states.

Examples:


start
1


start
q_1
$\varepsilon, \varepsilon \to \$$
q_2
$0, \varepsilon \to 0$
$1, 0 \to \varepsilon$
q_3
q_4
$\varepsilon, \$ \to \varepsilon$
$1, 0 \to \varepsilon$

PDAs and CFLs

Last lecture we saw that PDAs and CFLs are closely related:

Theorem

A language is context-free if and only if a PDA recognizes it.

- Proof was quite involved, check last lecture/book for details (a very good exercise!)
- So PDAs and CFLs have a similar relationship as NFAs and regular languages

Non-Context-Free Languages

- Today, we deal with non-context-free languages; i.e., we are interested in seeing whether a language is non-context free.
- Recall our discussion on nonregular languages, in particular the pumping lemma:

Lemma (Pumping Lemma for Regular Languages)

If A is a regular language, then there is a number p, called the pumping length, where if w is a word in A of length $\geq p$ then w can be divided into three parts, $w = xyz$, such that

1. $xy^i z \in A$ *for every* $i \geq 0$,
2. $|y| > 0$,
3. $|xy| \leq p$.

Pumping Lemma for Regular Languages

Lemma (Pumping Lemma for Regular Languages)

If A is a regular language, then there is a number p, called the pumping length, where if w is a word in A of length $\geq p$ then w can be divided into three parts, $w = xyz$, such that

1. $xy^iz \in A$ *for every* $i \geq 0$,
2. $|y| > 0$,
3. $|xy| \leq p$.

- Each component x, y, z corresponds to one part in an accepting path in the DFA.
- y corresponds to a cycle in the DFA
- x corresponds to a path from the initial node to the start of the cycle.
- z corresponds to a path from the end of the cycle to an accepting state.
- Condition (3) in the lemma was a useful tool when proving the nonregularity of a language.

Pumping Lemma for CFLs

Lemma (Pumping Lemma for CFLs)

For every context-free language A there exists a number p (called the pumping length) where, if w is a word in A of length $\geq p$, then w can be divided into five parts, $w = uvxyz$, satisfying the following conditions:

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

$\rightarrow$ Things are a bit more involved this time!

$\rightarrow$ For RL we considered a repetition in the sequence of transitions of a DFA.

$\rightarrow$ For CFL we consider a repetition in the derivation of the parse tree.

Pumping Lemma for CFLs

Lemma (Pumping Lemma for CFLs)

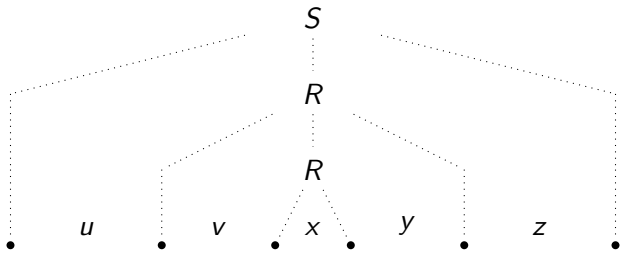
For CFL A there exists a pumping length p where, if $w \in A$ of length $\geq p$, then w can be divided into five parts, $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

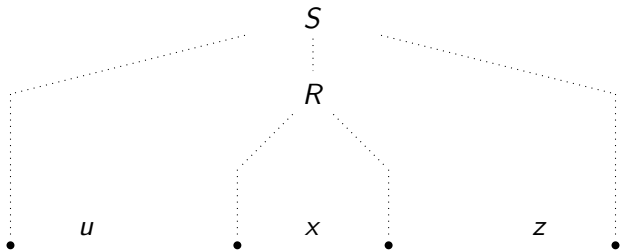
- **Proof idea:** Let G be the CFG generating A and τ a parse tree of w. If w is sufficiently large, we can argue that, along a path, a variable must occur twice.

Pumping Lemma - Proof

Parse tree for w:

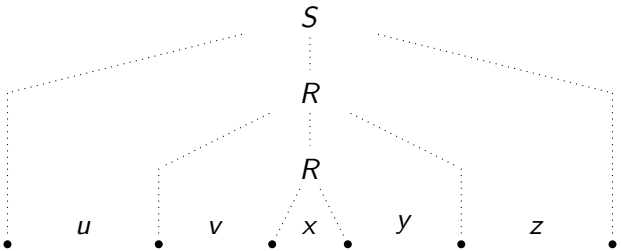


Pumping Lemma - Proof

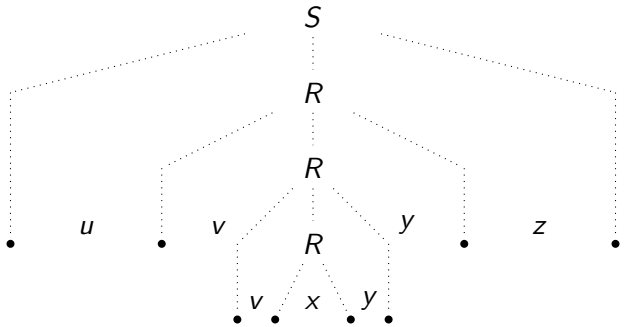


$\rightarrow uv^0xy^0z = uxz$

Pumping Lemma - Proof



Pumping Lemma - Proof



$\rightarrow uv^2xy^2z$, and so on

Pumping Lemma - Proof

Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- Let G be the CFG generating A and b be the maximal number of symbols on the right side of any rule in G.
- We can assume $b \geq 2$, since any CFG is equivalent to a CFG in Chomsky normal form.
- Thus, in a parse tree, any node can have maximally b children.
- $\Rightarrow$ If a parse tree has height h, then it has at most b^h leaves.

Pumping Lemma - Proof

Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- If a parse tree has height h, then it has at most b^h leaves.
- Hence if a string has length $b^h + 1$, then the parse tree must have height at least $h + 1$.
- Let $|V|$ be the number of variables in the grammar G and set the pumping length $p = b^{|V|+1}$.
- If $|w| \geq p$, then its parse tree must have height at least $|V| + 1$.

Pumping Lemma - Proof

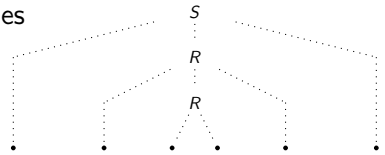
Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- If $|w| \geq p$, then its parse tree must have height at least $|V| + 1$.
- Let τ be the parse tree for w with the smallest number of nodes
- It has height at least $|V| + 1$, so the longest path from S to a leaf has length $\geq |V| + 1$, i.e., it contains $\geq |V| + 2$ nodes.
- The path must have at least $|V| + 1$ variables.
  But G only has $|V|$ variables, hence one must occur twice!



Pumping Lemma - Proof

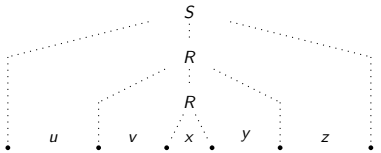
Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- But the path could be much longer than $|V| + 1$, so we choose R to be a repeating variable from the *lowest* $|V| + 1$ variables.
- We get the situation we looked at before and subdivide $w = uvxyz$.



Pumping Lemma - Proof

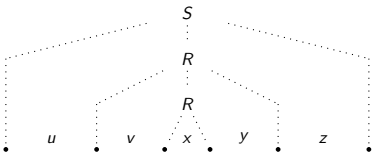
Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- The first occurrence of R generates vxy, while the second generates x (remember, only using rules from G),
- Thus we can replace one subtree with the other and get valid parse trees for uxz and $uv^i xy^i z$, $i > 1$.
- This proves condition 1.



Pumping Lemma - Proof

Lemma (Pumping Lemma for CFLs)

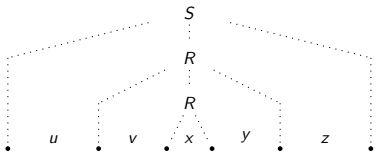
CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- For condition 2, assume, to the contrary, that $v = y = \varepsilon$.
- Then replacing the first R (generating vxy) with the second R (generating x) gives us a *smaller* parse tree for w.

  Contradiction! We assumed the parse tree τ was the smallest!



Pumping Lemma - Proof

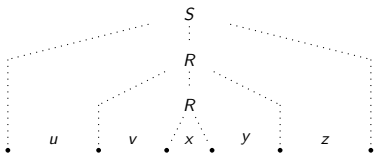
Lemma (Pumping Lemma for CFLs)

CFL A, $w \in A$ with $|w| \geq p$, then $w = uvxyz$, such that:

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Proof:

- Condition 3: We chose the Rs such that they lie in the *lowest* $|V| + 1$ variables on the path.
- Hence the subtree generated by the first R (which generates vxy) has height at most $|V| + 1$.
- Thus that subtree has at most $b^{|V|+1}$ leaves.
  $\Rightarrow |vxy| \leq b^{|V|+1} = p$ □



Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_1 = \{a^n b^n c^n \mid n \geq 0\}$.

- Assume A_1 is context-free. Let p be its pumping length and consider $w = a^p b^p c^p$.
- $|w| = 3p \geq p$ and $w \in A_1$ ✓
- Now let us consider how we can subdivide w into $uvxyz$.
- Condition 2: v or y must be nonempty.

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_1 = \{a^n b^n c^n \mid n \geq 0\}$.

- **Case 1:** Either v or y contains more than one type of symbol.
- Then uv^2xy^2z does not have the symbols in the correct order. **Contradiction!**

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_1 = \{a^n b^n c^n \mid n \geq 0\}$.

- **Case 2:** Both v and y contains only one type of symbol.
- Then v and y cannot contain both a's and b's or both b's and c's.
- But then uv^2xy^2z cannot have an equal number of a's, b's, *and* c's. **Contradiction!**
- w cannot be pumped.
- A_1 is, in fact, *context-sensitive* (more on that in a later lecture)

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_2 = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$.

- Assume A_2 context-free, p pumping length, and $w = a^p b^p c^p$. Again, $w = uvxyz$.
- **Case 1:** Same argument as before. If either v or y contains more than one type of symbol, then uv^2xy^2z will not have the right order of symbols. **Contradiction!**

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_2 = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}, w = a^p b^p c^p$.

- **Case 2:** If both v and y contain only one type of symbol, then we cannot use the same reasoning as in the last example (the number of a's, b's, and c's are different now!)
- However, since each contains only one type of symbol, one symbol a, b, or c does not occur in v or y.

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_2 = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}, w = a^p b^p c^p$.

- **Case 2a:** a does not appear. Then consider the string uxz. This string has the same amount (p) of a's as s, but fewer b's and c's. **Contradiction!**
- **Case 2b:** b does not appear. Then either a's or c's must occur in v or y (both cannot be empty). If a's appear, then uv^2xy^2z has more a's than b's. If c's appear, then uxz has more b's than c's. **Contradiction!**
- **Case 2c:** c does not appear. Then the string uv^2xy^2z contains more a's or b's than c's. **Contradiction!**

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_3 = \{\omega\omega \mid \omega \in \{0,1\}^*\}$.

- Assume A_3 is CF, p pumping length, and consider $w = 0^p10^p1$.
- Let $u = 0^{p-1}$, $v = 0$, $x = 1$, $y = 0$, and $z = 0^{p-1}1$. Then $w = uvxyz$.
- But this word can be pumped! uv^ixy^iz is still a word in A_3.
- We need to be more careful with our choice of w.

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_3 = \{\omega\omega \mid \omega \in \{0,1\}^*\}$.

- Let $w = 0^p 1^p 0^p 1^p$. We will make use of condition 3, so $|vxy| \leq p$.
- Two options: vxy is contained in the first or second half of w, or covers the middle.
- If vxy is contained in the first half, then pumping up to uv^2xy^2z "pushes" a 1 to the first position of the second half. But this is not a word in A_3.

  Example: 000111 000111, where v and y are green.
  Pumping up to uv^2xy^2z yields 0000111 1000111

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^i xy^i z \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_3 = \{\omega\omega \mid \omega \in \{0,1\}^*\}$.

- Let $w = 0^p 1^p 0^p 1^p$. We will make use of condition 3, so $|vxy| \leq p$.
- Two options: vxy is contained in the first or second half of w, or covers the middle.
- If vxy is contained in the second half, then pumping up to uv^2xy^2z "pushes" a 0 to the last position of the first half. **Contradiction.**

  Example: 000111 000111, where v and y are green.
  Pumping up to uv^2xy^2z yields 0001110 00000111

Pumping Lemma - Examples

Lemma (Pumping Lemma for CFLs)

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Language: $A_3 = \{\omega\omega \mid \omega \in \{0,1\}^*\}$.

- Now assume vxy covers the middle. Then pumping w down to uxz yields something of the form $0^p1^i0^j1^p$. Since v and y are not both empty, i and j cannot both equal p.
  $\Rightarrow$ not a word in A_3
- Hence w cannot be pumped
  $\Rightarrow$ A_3 is not context-free