SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

FOUNDATIONS OF COMPUTER SCIENCE
LECTURE 7: Non-Context-free languages

Prof. Daniele Gorla

Are all possible languages context-free?

We proved that $\{a^n b^n \mid n \geq 0\}$ is not regular

→ intuitively a DFA/NFA has no way to remember how many *a*'s has received so far (for arbitray *n*), since it has finitely many states

Similarly, we shall now prove that $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free

→ here the intuitive idea is that a PDA can use the stack for accepting as many *b*'s as the *a*'s (indeed, $\{a^n b^n \mid n \geq 0\}$ is C.F.)

→ but in doing so it empties the stack

→ so there is no more way of remembering how many *a*'s and *b*'s received

→ no way for accepting the very same number of *c*'s

Like for regular languages, we shall have a Pumping Lemma for C.F. languages

To better understand how it works, we first introduce the notion of *parse tree* of a string

Parse trees for a C.F. Grammar

Consider the following grammar for arithmetic expressions over the letter *a*

EXPR ::= EXPR + EXPR | EXPR x EXPR | (EXPR) | a

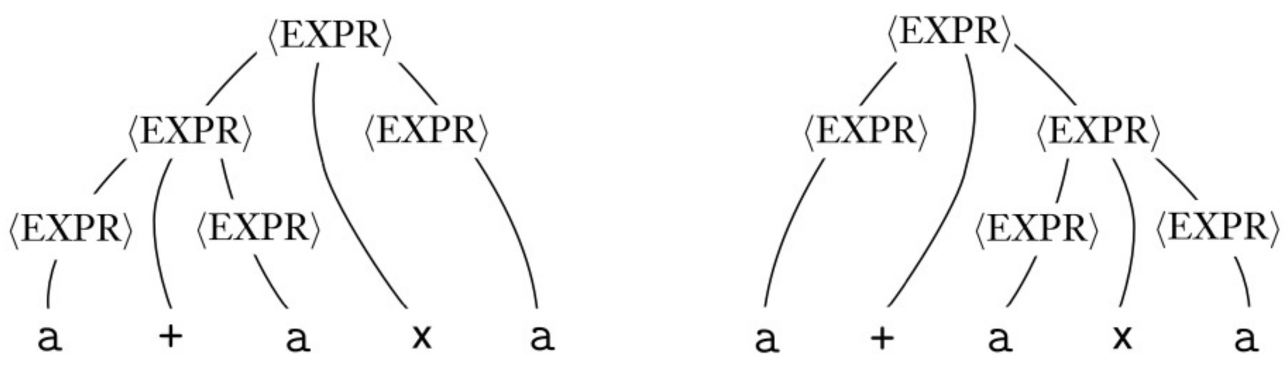
and the expression a + a x a

This string can be derived in two ways:

EXPR ⇒ EXPR x EXPR ⇒ EXPR + EXPR x EXPR ⇒⇒⇒ a + a x a

EXPR ⇒ EXPR + EXPR ⇒ EXPR + EXPR x EXPR ⇒⇒⇒ a + a x a

The derivations can be more easily depicted as a trees, whose leaves are terminal symbols, internal nodes are variables, and the root is the starting variable:



The Pumping Lemma for C.F. languages

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

THEOREM 2.34

Pumping lemma for context-free languages If A is a context-free language, then there is a number p (the pumping length) where, if s is any string in A of length at least p, then s may be divided into five pieces $s = uvxyz$ satisfying the conditions

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

- Notice that cond. 2 states that at least one between v and y (the pumped parts) must be non-empty (needed for the meaningfulness of the lemma)

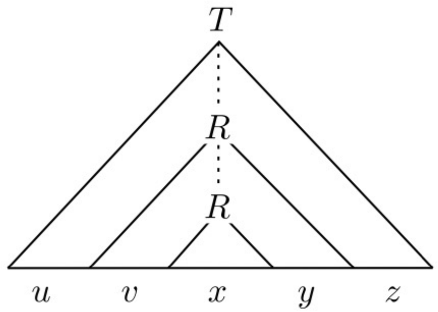
- Cond. 3 is similar to cond. 3 for the pumping lemma for regular languages and will be useful in proving that some langauges are not C.F.

- The P.L., more precisely:

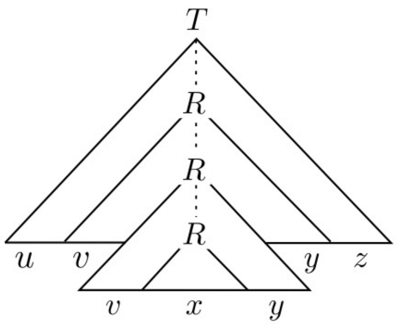
$$A \text{ C.F.} \Rightarrow \exists\, p \in \mathbb{N}\ \forall s \in A\ (\,|s| \geq p\ \Rightarrow\ \exists\, u,v,x,y,z \text{ s.t. } (\,s = uvxyz\ \wedge\ |vy| > 0\ \wedge\ |vxy| \leq p\ \wedge\ \forall i \in \mathbb{N}.\ uv^ixy^iz \in A)\,)$$

The Proof, intuitively

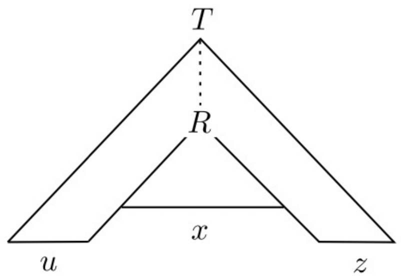
- Let A be a CFL, G be a CFG that generates it, and s be a ‘‘very long’’ string in A
- Because s is in A, it is derivable from G and so it has a parse tree
- Since s is ‘‘very long’’, the parse tree must contain some ‘‘long’’ path from the start variable at the root to one of the terminal symbols at a leaf
- On this ‘‘long’’ (i.e., greater than the number of variables) path, some variable R must repeat because of the pigeonhole principle:



- So, we may cut s into five pieces $uvxyz$
- Hence, we may repeat (or cancel) the second and fourth pieces at will to obtain strings that are still in the language:



u vv x yy z , i.e. pump vx twice



u x z, i.e. pump vy 0 times

- Thus, uv^ixy^iz is in A, for any $i \geq 0$.

The Proof, formally (1)

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

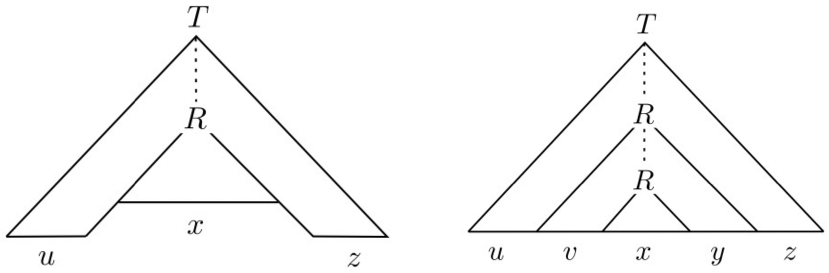
- Let G be a CFG for the CFL A
- Let b be the maximum number of symbols in the right-hand side of a rule in G longest RHS
- In any parse tree for G, every internal node has at most b children

  → a tree of height h has at most b^h leaves
- Hence, a string of length b^h+1 has parse trees of heigth at least $h+1$
- Let the pumping length p be $b^{|V|+1}$, where V are the variables of G
- Now, if a string $s \in A$ is long p or more, its parse trees must all be at least $|V|+1$ high (indeed, $b^{|V|+1} \geq b^{|V|}+1$)
- Let τ be a parse tree for s (if many exist, choose one with the smallest number of nodes)
- The longest path in τ has length at least $|V|+1$ and so it has at least $|V|+2$ nodes

  → the last one is a terminal, all the others (at least $|V|+1$) are variables
- Since G has only $|V|$ variables, some variable R appears more than once on this path
- Let R be the first variable that repeats along this path
- Then, by working like in the intuitive proof, we have condition 1 of the Lemma.

The Proof, formally (2)

- For condition 2, we work by contradiction:
  - Imagine that $v = y = \varepsilon$
  - The parse tree obtained by substituting the smaller subtree for the larger would have fewer nodes than τ does and would still generate s:



  - This isn't possible because we chose τ to be a parse tree for s with the smallest number of nodes

so there cannot be a smaller tree equivalent to the one chosen

- For condition 3:
  - In the parse tree for s, the upper occurrence of R generates vxy look ad u v x y z tree
  - We chose R so that both occurrences fall within the bottom $|V|+1$ variables on the path
  - So the subtree where R generates vxy is at most $|V|+1$ high. trivially, in the example |V| = 2 and the tree has an height of 3
  - A tree of this height can generate a string of length at most $b^{|V|+1} = p$.

Q.E.D.

Example of use

Like for regular languages, we use the contrapositive:

$$\forall p \in \mathbb{N}\, \exists s \in A\ (\, |s| \geq p \wedge \forall u,v,x,y,z\ (s \neq uvxyz \vee |vy| = 0 \vee |vxy| > p \vee \exists i \in \mathbb{N}.\ uv^ixy^iz \notin A)$$

$$\Rightarrow A \text{ is not C.F.}$$

or equivalently:

$$\forall p \in \mathbb{N}\, \exists s \in A\ (\, |s| \geq p \wedge \forall u,v,x,y,z\ ((s = uvxyz \wedge |vy| > 0 \wedge |vxy| \leq p) \Longrightarrow \exists i \in \mathbb{N}.\ uv^ixy^iz \notin A)\ )$$

$$\Rightarrow A \text{ is not C.F.}$$

EXAMPLE:

Let us show that the language $B = \{a^nb^nc^n \mid n \geq 0\}$ is not context free.

- Fix a generic p
- Let $s = a^pb^pc^p$ ($s \in B$ and $|s| > p$)
- Let s be split into $uvxyz$, with either v or y nonempty:
  - If both v and y contain only one type of alphabet symbol:

    → uv^2xy^2z cannot contain an equal numbers of a's, b's, and c's !!
  - If either v or y contains more than one type of symbol

    → uv^2xy^2z may contain an equal number of a's, b's, and c's but not in the correct order

A more delicate example of use

We prove that $C = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$ is not a CFL

Like before, consider a generic p and use again the string $s = a^p b^p c^p$, a decomposition $uvxyz$ and the two cases considered before:

I. When v or y contains more types of symbol, uv^2xy^2z has symbols in the wrong order

II. When both v and y contain only one type of symbol, one of the symbols doesn't appear therein

1. *a doesn't appear:* Consider $uv^0xy^0z = uxz$ that contains p a's but less than p b's or c's
   → it is not a member of C !!
2. *c doesn't appear:* Consider uv^2xy^2z that contains p c's but more than p a's or b's
   → it is not a member of C !!
3. *b doesn't appear:*
   - If a appears in v or y, the string uv^2xy^2z contains more a's than b's, so it is not in C
   - If c appears in v or y, the string uv^0xy^0z contains more b's than c's, so it is not in C

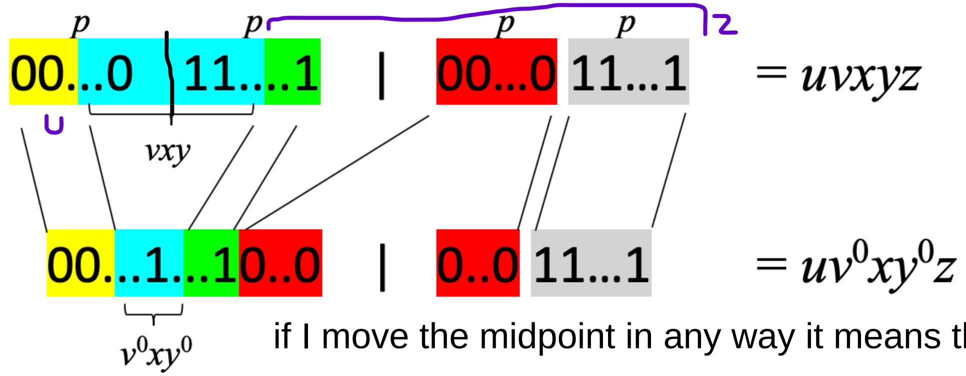
An example where condition 3 is crucial

Prove that $D = \{ww \mid w \in \{0,1\}^*\}$ is not a CFL

Fix a p and consider the string $s = 0^p1^p0^p1^p$ (it is of the form ww, for $w = 0^p1^p$, so it belongs to D)

Let us now divide s into $uvxyz$, where $|vxy| \leq p$

1. $|vxy|$ must be even, otherwise uv^2xy^2z would have an odd length (and so cannot belong to D)
2. If vxy lies within the leftmost 0^p1^p, then uv^0xy^0z could be decomposed in two substrings, one ending with 0 and the other one with 1 (so, it would not be of the form $w'w'$, for some w'):



if I move the midpoint in any way it means that the form is not ww

3. Dually, if vxy lies within the rightmost 0^p1^p, then uv^0xy^0z could be decomposed in two substrings, one starting with 0 and the other one with 1 (so, it would not be of the form $w'w'$, for some w')
4. If vxy lies within 1^p0^p, then uv^0xy^0z is of the form $0^p1^i0^j1^p$ (for at least one between i and j strictly smaller than p). Hence, it cannot belong to D, since it can only be seen as the justapoxition of two (equal length) strings, one ending with 0 and the other one with 1 (if $i < j$), or one starting with 0 and the other one with 1 (if $i > j$) or with different numbers of 0's and 1's (if $i=j$).