Limits of Context-Free Grammars/Languages

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated
2. easy kind of automata: finite automata (regular expressions)

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated
2. easy kind of automata: finite automata (regular expressions)
3. problem: $\{a^n b^n | n \in \mathbb{N}\}$ is not regular

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated
2. easy kind of automata: finite automata (regular expressions)
3. problem: $\{a^n b^n | n \in \mathbb{N}\}$ is not regular
4. more sophisticated way: context-free grammars

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated
2. easy kind of automata: finite automata (regular expressions)
3. problem: $\{a^n b^n | n \in \mathbb{N}\}$ is not regular
4. more sophisticated way: context-free grammars
5. is everything producible by context-free grammars?

Can we express everything with CFLs?

1. HALT-Problem is somehow complicated
2. easy kind of automata: finite automata (regular expressions)
3. problem: $\{a^n b^n \mid n \in \mathbb{N}\}$ is not regular
4. more sophisticated way: context-free grammars
5. is everything producible by context-free grammars?
6. what is with $\{a^n b^n c^n \mid n \in \mathbb{N}\}$?

The Pumping Lemma for CFLs

Reminder: How to prove non-regularity?

- no DFA/NFA/regexp exists for the language
- contradict the supposition that it is regular using the Pumping Lemma
- prove that the index of the Myhill-Nerode-relation is infinite

Pumping Lemma for Context-Free Languages

Lemma

L *context-free language* $\Rightarrow \exists p \in \mathbb{N} \forall z \in L^{\geq p} \exists u, v, w, x, y \in \Sigma^* :$

1. $z = uvwxy$
2. $vx \neq \varepsilon$
3. $|vwx| \leq p$
4. $\forall i \in \mathbb{N}_0 : uv^i wx^i y \in L$

Cont.

Definition

Parse Tree for a word z producible by a grammar in CNF:

- nodes $N = V \cup \Sigma$
- start with S
- if the rule $L \longrightarrow \ell$ is applied: ℓ is a child of L
- if the rule $L \longrightarrow R_1 R_2$ is applied: R_1 and R_2 are children

Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!

Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!
- set $p = 2^{n+1}$

Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!
- set $p = 2^{n+1}$
- $z \in L(G)$, $|z| \geq n$

Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!
- set $p = 2^{n+1}$
- $z \in L(G)$, $|z| \geq n$
- any parse tree for z has depth at least $n + 1$

Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!
- set $p = 2^{n+1}$
- $z \in L(G)$, $|z| \geq n$
- any parse tree for z has depth at least $n + 1$
- $\Rightarrow$ the longest path has length at least $n + 1$

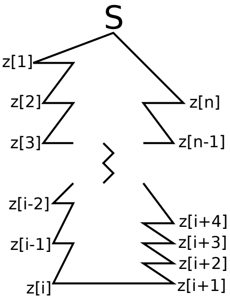
Proof of the Pumping Lemma vor CFL

$G = (V, \Sigma, S, P)$ CFG with $|V| = n$

- we need to proof that the conclusion really holds!
- set $p = 2^{n+1}$
- $z \in L(G)$, $|z| \geq n$
- any parse tree for z has depth at least $n + 1$
- $\Rightarrow$ the longest path has length at least $n + 1$
- only n variables $\Rightarrow$ one occurs at least twice

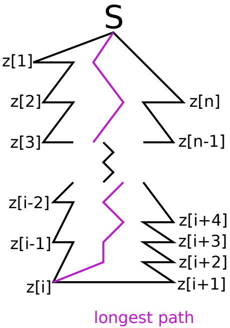
Cont.

read longest path from bottom to top and take the first two variables X occurring twice



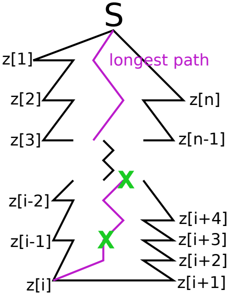
Cont.

read longest path from bottom to top and take the first two variables X occurring twice



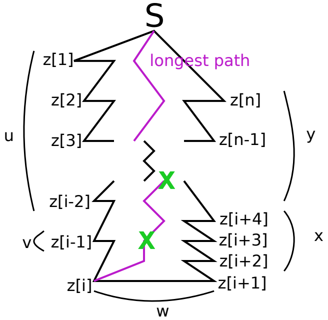
Cont.

read longest path from bottom to top and take the first two variables X occurring twice



Cont.

read longest path from bottom to top and take the first two variables X occurring twice



Choose u, v, w, x, y at the bottom of the tree with

- w is generated by lower occurrence of X
- v is generated by left child of upper occurrence of X
- x is generated by right child of upper occurrence of X
- u is part before v
- y is part after x

Cont.

- trick:

Cont.

- trick:
- both subtrees beneath the both S are valid derivations

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one
- $\Rightarrow$ we doubled v and x and are still in the language

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one
- $\Rightarrow$ we doubled v and x and are still in the language
- we can do this as often as we want

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one
- $\Rightarrow$ we doubled v and x and are still in the language
- we can do this as often as we want
- $\Rightarrow$ claim proven for all $i > 0$

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one
- $\Rightarrow$ we doubled v and x and are still in the language
- we can do this as often as we want
- $\Rightarrow$ claim proven for all $i > 0$
- for $i = 0$: exchange the upper one by the lower one

Cont.

- trick:
- both subtrees beneath the both S are valid derivations
- exchange the lower one by the upper one
- $\Rightarrow$ we doubled v and x and are still in the language
- we can do this as often as we want
- $\Rightarrow$ claim proven for all $i > 0$
- for $i = 0$: exchange the upper one by the lower one
- v, x are gone!

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Proof:

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Proof:

- Contradiction: Suppose L is context free

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Proof:

- Contradiction: Suppose L is context free
- PL $\Rightarrow$ exists $p \in \mathbb{N}$ (take it)

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Proof:

- Contradiction: Suppose L is context free
- PL $\Rightarrow$ exists $p \in \mathbb{N}$ (take it)
- set $z = a^p b^p a^p$

Nasty Language II

Context-Freedom is not the jack of all trades device.

Lemma

The language $L = \{a^n b^n a^n \mid n \in \mathbb{N}_0\}$ is not context-free.

Proof:

- Contradiction: Suppose L is context free
- PL $\Rightarrow$ exists $p \in \mathbb{N}$ (take it)
- set $z = a^p b^p a^p$
- $\Rightarrow w \in L, |z| \geq p$

Cont.

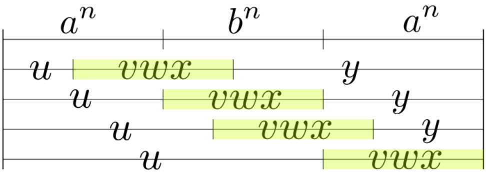
$$\underbrace{a^n}\;\underbrace{b^n}\;\underbrace{a^n}$$

Cont.

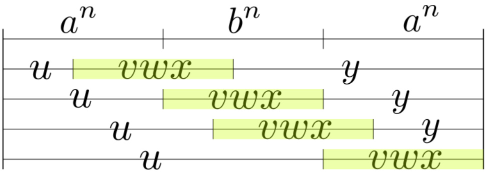
$$\overset{a^n}{\vdash\!\!\text{———}\!\!\dashv}\ \overset{b^n}{\vdash\!\!\text{———}\!\!\dashv}\ \overset{a^n}{\vdash\!\!\text{———}\!\!\dashv}$$

we have to check all decompositions into $uvwxy$

Cont.

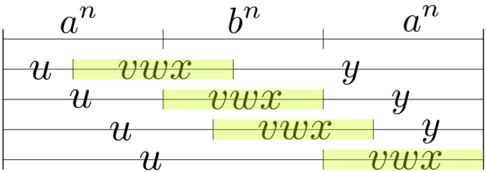


Cont.



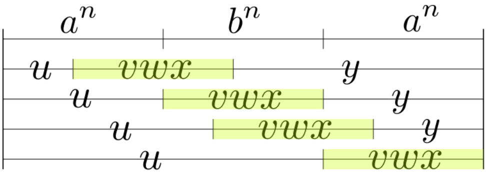
- case 1: no change in third part

Cont.



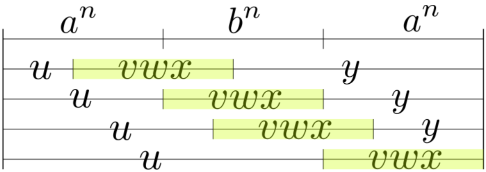
- case 1: no change in third part
- case 2: no change in third part

Cont.



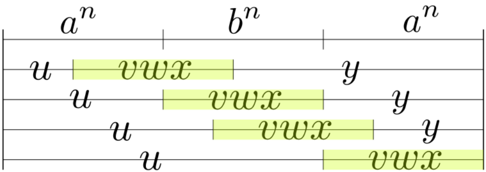
- case 1: no change in third part
- case 2: no change in third part
- case 3: no change in first part

Cont.



- case 1: no change in third part
- case 2: no change in third part
- case 3: no change in first part
- case 4: no change in first part

Cont.



- case 1: no change in third part
- case 2: no change in third part
- case 3: no change in first part
- case 4: no change in first part

contradiction to (4)! $\square$