Computability and Complexity
COSC 4200

Non-Context-Free Languages

Pumping Lemma for Context-Free Languages

If A is a context-free language, then there is a number p (the pumping constant) such that for every $w \in A$ with $|w| \geq p$, w can be divided into five pieces $w = uvxyz$ satisfying the following conditions:

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

Pumping Lemma for Context-Free Languages

If A is a context-free language, then there is a number p (the pumping constant) such that for every $w \in A$ with $|w| \geq p$, w can be divided into five pieces $w = uvxyz$ satisfying the following conditions:

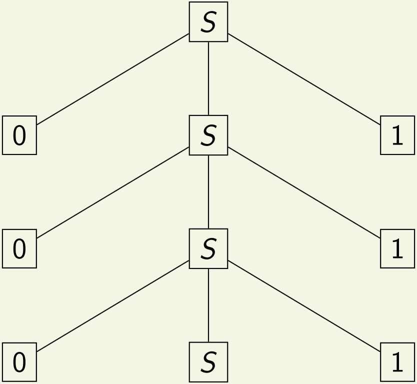
1. For all $i \geq 0$, $uv^ixy^iz \in A$.
2. $|vy| > 0$.
3. $|vxy| \leq p$.

Comments.

- Condition (2) says that $v \neq \epsilon$ or $y \neq \epsilon$.
- Condition (1) says that uxz, $uvxyz$, $uvvxyyz$, $uvvvxyyyz$, ... are all in A.

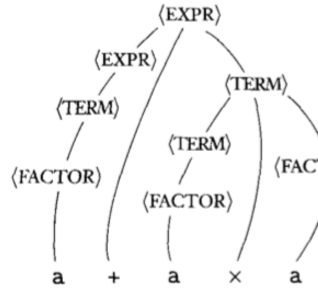
Parse Trees

$$S \to 0S1|\epsilon$$



Parse Trees

$$
\begin{aligned}
\langle \text{EXPR} \rangle &\to \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle \\
\langle \text{TERM} \rangle &\to \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle \\
\langle \text{FACTOR} \rangle &\to (\langle \text{EXPR} \rangle) \mid \mathtt{a}
\end{aligned}
$$



Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

- At most b leaves are 1 step from the start variable.

Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

- At most b leaves are 1 step from the start variable.
- At most b^2 leaves are within 2 steps from the start variable.

Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

- At most b leaves are 1 step from the start variable.
- At most b^2 leaves are within 2 steps from the start variable.
- $\vdots$
- At most b^h leaves are within h steps from the start variable.

Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

- At most b leaves are 1 step from the start variable.
- At most b^2 leaves are within 2 steps from the start variable.
- $\vdots$
- At most b^h leaves are within h steps from the start variable.

Therefore, if the height of the parse tree is at most h, the length of the string generated is at most b^h.

Proof. Let G be a CFG for A.
Let b be the max # symbols on the RHS of a rule (assume $b \geq 2$).

In any parse tree for G, a node can have no more than b children.

- At most b leaves are 1 step from the start variable.
- At most b^2 leaves are within 2 steps from the start variable.
- $\vdots$
- At most b^h leaves are within h steps from the start variable.

Therefore, if the height of the parse tree is at most h, the length of the string generated is at most b^h.

Conversely, if a generated string is at least $b^h + 1$ long, each of its parse trees must be at least $h + 1$ high.

Recall $|V|$ is the $\#$ of variables in G. We set the pumping constant p to be

$$p = b^{|V|+1}.$$

Recall $|V|$ is the $\#$ of variables in G. We set the pumping constant p to be

$$p = b^{|V|+1}.$$

If a string $s \in A$ and $|s| \geq p$, any parse tree for s must be at least $|V| + 1$ high, because

$$p = b^{|V|+1} \geq b^{|V|} + 1.$$

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.
We know:

- τ must be at least $|V| + 1$ high
  $\Rightarrow \exists$ a path from the root to a leaf of length $\geq |V| + 1$

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.
We know:

- τ must be at least $|V| + 1$ high
  $\Rightarrow \exists$ a path from the root to a leaf of length $\geq |V| + 1$
- this path has $\geq |V| + 2$ nodes: one terminal, the others variables

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.
We know:

- τ must be at least $|V| + 1$ high
  $\Rightarrow \exists$ a path from the root to a leaf of length $\geq |V| + 1$
- this path has $\geq |V| + 2$ nodes: one terminal, the others variables
- therefore there are $\geq |V| + 1$ variables

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.
We know:

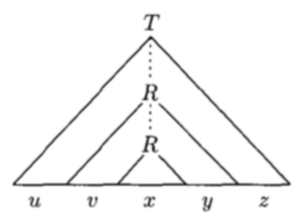
- τ must be at least $|V| + 1$ high
  $\Rightarrow \exists$ a path from the root to a leaf of length $\geq |V| + 1$
- this path has $\geq |V| + 2$ nodes: one terminal, the others variables
- therefore there are $\geq |V| + 1$ variables
- there are only $|V|$ variables in G, so some variable appears more than once on the path

Let τ be a parse tree for $s \in A$, $s \geq p$. If there are multiple parse trees, choose τ to be one with the smallest number of nodes.
We know:

- τ must be at least $|V| + 1$ high
  $\Rightarrow \exists$ a path from the root to a leaf of length $\geq |V| + 1$
- this path has $\geq |V| + 2$ nodes: one terminal, the others variables
- therefore there are $\geq |V| + 1$ variables
- there are only $|V|$ variables in G, so some variable appears more than once on the path
- let R be a variable that repeats among the lowest $|V| + 1$ variables on the path

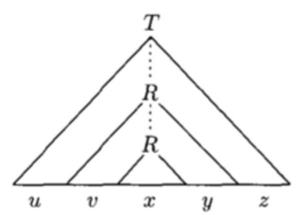
We divide s into $uvxyz$ as in the diagram at right.

- The upper occurence of R generates vxy and has a larger subtree.
- The lower occurence of R generates x and has a smaller subtree.



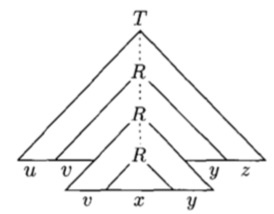
We divide s into $uvxyz$ as in the diagram at right.

- The upper occurence of R generates vxy and has a larger subtree.
- The lower occurence of R generates x and has a smaller subtree.



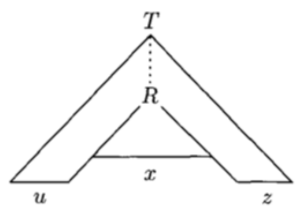
We may replace the smaller subtree with a copy of the larger subtree: This generates the string uv^2xy^2z.

Repeating this process, we generate uv^ixy^iz for all $i \geq 2$.



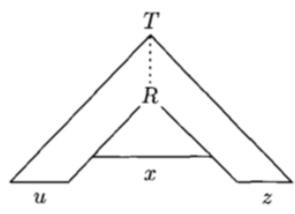
We can also replace the larger subtree by the smaller subtree.

This generates $uxz = uv^0xy^0z$.



We can also replace the larger subtree by the smaller subtree.

This generates $uxz = uv^0xy^0z$.

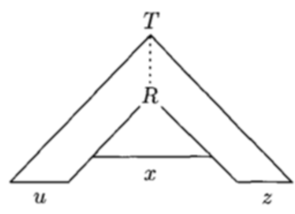


---

1. We have established condition 1: $uv^ixy^iz \in A$ for all $i \geq 0$.

We can also replace the larger subtree by the smaller subtree.

This generates $uxz = uv^0xy^0z$.

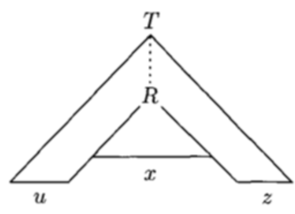


---

1. We have established condition 1: $uv^ixy^iz \in A$ for all $i \geq 0$.
2. Condition 2: we need to show $|vy| > 0$. If $v = y = \epsilon$, then the parse tree above is a smaller tree than τ that also generates s, contradicting our choice that τ was the smallest parse tree.

We can also replace the larger subtree by the smaller subtree.

This generates $uxz = uv^0xy^0z$.



---

1. We have established condition 1: $uv^ixy^iz \in A$ for all $i \geq 0$.
2. Condition 2: we need to show $|vy| > 0$. If $v = y = \epsilon$, then the parse tree above is a smaller tree than τ that also generates s, contradicting our choice that τ was the smallest parse tree.
3. Condition 3: we need to show $|vxy| \leq p$. Recall R generates vxy and R is in the bottom $|V| + 1$ variables. This means that the number of leaves in the subtree rooted at R is at most $b^{|V|+1}$. Thus R can generate a string of length $\leq b^{|V|+1}$. Since $p = b^{|V|+1}$, condition 3 holds. $\square$

Pumping Lemma for Context-Free Languages

If A is a context-free language, then there is a number p (the pumping constant) such that for every $w \in A$ with $|w| \geq p$, w can be divided into five pieces $w = uvxyz$ satisfying the following conditions:

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

Comments.

- Condition (2) says that $v \neq \epsilon$ or $y \neq \epsilon$.
- Condition (1) says that uxz, $uvxyz$, $uvvxyyz$, $uvvvxyyyz$, ... are all in A.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$. Then let $uvxyz = w$ such that $|vy| > 0$ and $|vxy| \leq p$.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$. Then let $uvxyz = w$ such that $|vy| > 0$ and $|vxy| \leq p$.
We will consider two cases, based whether v and y contain more than one type of alphabet symbol.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$. Then let $uvxyz = w$ such that $|vy| > 0$ and $|vxy| \leq p$.
We will consider two cases, based whether v and y contain more than one type of alphabet symbol.

1. Suppose v and y each contain only one type of alphabet symbol. Then the string uv^2xy^2z will not contain equal numbers of a's, b's, and c's (because it will have more of the symbols that are present in v and y – but this does not include all three symbols). Therefore it is not a member of A.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$. Then let $uvxyz = w$ such that $|vy| > 0$ and $|vxy| \leq p$.
We will consider two cases, based whether v and y contain more than one type of alphabet symbol.

1. Suppose v and y each contain only one type of alphabet symbol. Then the string uv^2xy^2z will not contain equal numbers of a's, b's, and c's (because it will have more of the symbols that are present in v and y – but this does not include all three symbols). Therefore it is not a member of A.
2. Suppose that v or y contains more than one type of alphabet symbol. Then the string uv^2xy^2z will not be a member of $a^* b^* c^*$ and therefore also not a member of A.

Example. $A = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

Proof. Suppose that A is context-free, and let p be the pumping constant for A. Let $w = a^p b^p c^p$. Then let $uvxyz = w$ such that $|vy| > 0$ and $|vxy| \leq p$.

We will consider two cases, based whether v and y contain more than one type of alphabet symbol.

1. Suppose v and y each contain only one type of alphabet symbol. Then the string uv^2xy^2z will not contain equal numbers of a's, b's, and c's (because it will have more of the symbols that are present in v and y – but this does not include all three symbols). Therefore it is not a member of A.
2. Suppose that v or y contains more than one type of alphabet symbol. Then the string uv^2xy^2z will not be a member of $a^*b^*c^*$ and therefore also not a member of A.

In either case, we have a contradiction of the context-free pumping lemma. Therefore the assumption that A is context-free must be false, so A is not context-free. $\square$

Example. $D = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Proof. Suppose that D is context-free, and let p be the pumping constant for D. Let $s = 0^p 1^p 0^p 1^p$.

Example. $D = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Proof. Suppose that D is context-free, and let p be the pumping constant for D. Let $s = 0^p 1^p 0^p 1^p$. Then let $uvxyz = s$ such that $|vy| > 0$ and $|vxy| \leq p$.

We consider three cases:

Example. $D = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Proof. Suppose that D is context-free, and let p be the pumping constant for D. Let $s = 0^p1^p0^p1^p$. Then let $uvxyz = s$ such that $|vy| > 0$ and $|vxy| \leq p$.

We consider three cases:

1. Suppose vxy lies in the first half of s. Then pumping v and y will move a 1 into the first position of the second half. Therefore $uv^2xy^2z \notin D$.

Example. $D = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Proof. Suppose that D is context-free, and let p be the pumping constant for D. Let $s = 0^p1^p0^p1^p$. Then let $uvxyz = s$ such that $|vy| > 0$ and $|vxy| \leq p$.

We consider three cases:

1. Suppose vxy lies in the first half of s. Then pumping v and y will move a 1 into the first position of the second half. Therefore $uv^2xy^2z \notin D$.
2. Similarly, if vxy lies in the second half of s, uv^2xy^2z will have a 0 in the last position of the first half, so $uv^2xy^2z \notin D$.

Example. $D = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Proof. Suppose that D is context-free, and let p be the pumping constant for D. Let $s = 0^p 1^p 0^p 1^p$. Then let $uvxyz = s$ such that $|vy| > 0$ and $|vxy| \leq p$.

We consider three cases:

1. Suppose vxy lies in the first half of s. Then pumping v and y will move a 1 into the first position of the second half. Therefore $uv^2xy^2z \notin D$.
2. Similarly, if vxy lies in the second half of s, uv^2xy^2z will have a 0 in the last position of the first half, so $uv^2xy^2z \notin D$.
3. The remaining case is when vxy straddles the midpoint of s. In this case, pumping down, we get $uxz = 0^p 1^i 0^j 1^p$, where $i < p$ or $j < p$. Hence $uxz \notin A$.

This contradicts the context-free pumping lemma, and therefore D is not context-free. $\square$

Closure Properties of CFL

The Context-Free Languages are Closed Under:

- Union
- Concatenation
- Star

What about intersection?

What about intersection?

$$A = \{a^n b^n c^n \mid n \geq 0\} \notin \mathrm{CFL}.$$

Let

$$B = \{a^n b^n c^m \mid n, m \geq 0\},$$

$$C = \{a^n b^m c^m \mid n, m \geq 0\}.$$

Then $A = B \cap C$ and $B, C \in \mathrm{CFL}$:

What about intersection?

$$A = \{a^n b^n c^n \mid n \geq 0\} \notin \mathrm{CFL}.$$

Let

$$B = \{a^n b^n c^m \mid n, m \geq 0\},$$

$$C = \{a^n b^m c^m \mid n, m \geq 0\}.$$

Then $A = B \cap C$ and $B, C \in \mathrm{CFL}$:

CFG for B	CFG for C
$S \to LR$	$S \to LR$
$L \to aLb \mid \epsilon$	$L \to aL \mid \epsilon$
$R \to cR \mid \epsilon$	$R \to bRc \mid \epsilon$

Therefore CFL is *not* closed under intersection.

However, it can be shown that the context-free languages are closed under intersection with the regular languages:

Theorem

If $A \in \mathrm{CFL}$ and $B \in \mathrm{REG}$, then $A \cap B \in \mathrm{CFL}$.

However, it can be shown that the context-free languages are closed under intersection with the regular languages:

Theorem

If $A \in \mathrm{CFL}$ *and* $B \in \mathrm{REG}$*, then* $A \cap B \in \mathrm{CFL}$*.*

Proof.

Homework :) $\square$

Are the context-free languages closed under complement?

Are the context-free languages closed under complement? *No.*

Are the context-free languages closed under complement? *No.*

We will give two different proofs of this.

Are the context-free languages closed under complement? *No.*

We will give two different proofs of this.

Proof 1. Suppose that CFL is closed under complement.
Let

$$A = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}.$$

Then $A \in \mathrm{CFL}$, so $A^c \in \mathrm{CFL}$ by assumption.

Are the context-free languages closed under complement? *No.*

We will give two different proofs of this.

Proof 1. Suppose that CFL is closed under complement.
Let

$$A = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}.$$

Then $A \in \mathrm{CFL}$, so $A^c \in \mathrm{CFL}$ by assumption.

Because CFL is closed under intersection with REG, we also have

$$A^c \cap a^* b^* c^* = \{a^n b^n c^n \mid n \geq 0\} \in \mathrm{CFL},$$

a contradiction. $\square$

Proof 2. Let
$$B = \{ww \mid w \in \{0,1\}^*\}.$$
Then $B \notin \mathrm{CFL}$. However, we can show that

$$\begin{aligned} B^c &= \Big\{xy \;\Big|\; |x| = |y| \text{ and } x \neq y\Big\} \\ &\quad \bigcup \Big\{x \in \{0,1\}^* \;\Big|\; |x| \text{ is odd}\Big\} \end{aligned}$$

is context-free!

Proof 2. Let

$$B = \{ww \mid w \in \{0,1\}^*\}.$$

Then $B \notin \mathrm{CFL}$. However, we can show that

$$\begin{aligned} B^c &= \left\{xy \;\middle|\; |x| = |y| \text{ and } x \neq y\right\} \\ &\quad \bigcup \left\{x \in \{0,1\}^* \;\middle|\; |x| \text{ is odd}\right\} \end{aligned}$$

is context-free!

Here is a CFG for B^c:

$$\begin{aligned} S &\rightarrow AB \mid BA \mid A \mid B \\ A &\rightarrow CAC \mid 0 \\ B &\rightarrow CBC \mid 1 \\ C &\rightarrow 0 \mid 1 \end{aligned}$$

1. Starting with $S \to A$ or $S \to B$, we can derive all strings of odd length.
2. Starting with $S \to AB$, we can derive all strings of the form

$$r0su1v$$

where $|r| = |s|$ and $|u| = |v|$.

3. Starting with $S \to BA$, we can derive all strings of the form

$$u1vr0s$$

where $|u| = |v|$ and $|r| = |s|$.

If $w = xy$ with $|x| = |y|$ and $x \neq y$, then w can be derived in the second or third forms above. $\square$

Suppose

$$xy = x_1 0 x_2 y_1 1 y_2$$

where $x = x_1 0 x_2$, $y = y_1 1 y_2$, $|x_1| = |y_1|$, $|x_2| = |y_2|$.

Suppose

$$xy = x_1 0 x_2 y_1 1 y_2$$

where $x = x_1 0 x_2$, $y = y_1 1 y_2$, $|x_1| = |y_1|$, $|x_2| = |y_2|$.

Then we generate $xy = u0vr1s$ as above using

$$u = x_1,$$

$$v = \text{first } |x_1| \text{ bits of } x_2 y_1,$$

$$r = \text{remaining } |y_2| \text{ bits of } x_2 y_1,$$

$$s = y_2.$$

$$\begin{aligned} xy &= \boxed{\quad x \quad}\,\boxed{\quad y \quad} \\ &= \boxed{x_1}\,0\,\boxed{\; x_2 \;}\,\boxed{y_1}\,1\,\boxed{\; y_2 \;} \\ &= \boxed{u}\,0\,\boxed{v}\,\boxed{\quad r \quad}\,1\,\boxed{\; s \;} \end{aligned}$$

Summary of CFL Closure Properties

CFL is closed under:

- union
- concatenation
- star

CFL is not closed under:

- intersection
- complement