COMP4141 Theory of Computation

Lecture 6 CFLs (cont.) & Grammars

Ron van der Meyden

CSE, UNSW

Revision: 2014/03/19

(Credits: David Dill, Thomas Wilke, Kai Engelhardt, Peter Höfner, Rob van Glabbeek)

Theorem

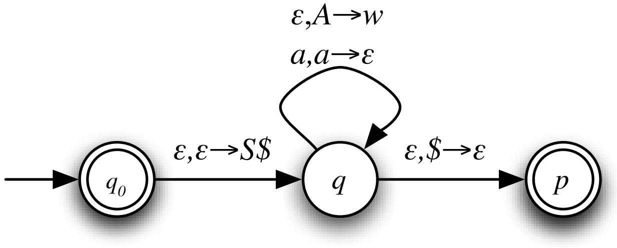
$L \subseteq \Sigma^*$ is context-free iff some PDA recognises L.

CFG $\longrightarrow$ PDA

Proof idea: Let the PDA guess a derivation, then match what it derived against the input word.

Refinement of the idea: Let the PDA guess a derivation while matching terminals from the input word.

CFG $\longrightarrow$ PDA



$\varepsilon, A \rightarrow w$
$a, a \rightarrow \varepsilon$
$\varepsilon, \varepsilon \rightarrow S\$$
$\varepsilon, \$ \rightarrow \varepsilon$
q_0
q
p

for every rule $A \rightarrow w$ and terminal a. The non-terminal handling transitions are abbreviations for detours through auxiliary states to build up w on the stack one symbol at a time.
The details are in [Sipser2006].

PDA $\longrightarrow$ CFG

Start with a PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$.

Design a CFG $G = (V, \Sigma, R, S)$ with a non-terminal A_{pq} for each pair $(p, q) \in Q \times Q$ with the idea that $A_{p,q}$ generates all those strings that take P from state p with an empty stack to state q with an empty stack:

$$V = \{ \ A_{pq} \mid p, q \in Q \ \}$$
$$S = A_{q_0 q_{\text{accept}}}$$

(This also means that strings generated by A_{pq} take P from state p with stack content γ to state q with stack content γ without ever dipping into γ.)

PDA $\longrightarrow$ CFG, cont.

First we transform the machine into one that

1. has a single accept state $F = \{q_{\text{accept}}\}$
2. empties its stack before terminating
3. has only transitions that either push or pop a symbol from the stack (but not both in a single transition, nor transitions that neither push nor pop.)

PDA $\longrightarrow$ CFG, cont.

R consists of the following rules:

- $A_{pq} \to aA_{rs}b$ if $p \xrightarrow{a,\epsilon\to t} r$ and $s \xrightarrow{b,t\to\epsilon} q$
  for some $p, q, r, s \in Q$, $t \in \Gamma$ and $a, b \in \Sigma_\epsilon$
- $A_{pq} \to A_{pr}A_{rq}$, for $p, q, r \in Q$
- $A_{pp} \to \epsilon$, for $p \in Q$

Lemma

If $A_{pq} \Rightarrow^*_G x \in \Sigma^*$ then x can take P from state p with an empty stack to state q with an empty stack.

Lemma

If $x \in \Sigma^*$ can take P from state p with an empty stack to state q with an empty stack then $A_{pq} \Rightarrow^*_G x$.

The lemmas are proven in [Sipser2006].

PDA $\longrightarrow$ CFG, cont.

R consists of the following rules:

- $A_{pq} \to aA_{rs}b$ if $p \xrightarrow{a,\epsilon\to t} r$ and $s \xrightarrow{b,t\to\epsilon} q$
  for some $p, q, r, s \in Q$, $t \in \Gamma$ and $a, b \in \Sigma_\epsilon$
- $A_{pq} \to A_{pr}A_{rq}$, for $p, q, r \in Q$
- $A_{pp} \to \epsilon$, for $p \in Q$

Lemma

If $A_{pq} \Rightarrow^*_G x \in \Sigma^*$ then x can take P from state p with an empty stack to state q with an empty stack.

Lemma

If $x \in \Sigma^*$ can take P from state p with an empty stack to state q with an empty stack then $A_{pq} \Rightarrow^*_G x$.

The lemmas are proven in [Sipser2006].

How to pump CFLs

Similar to regular languages, CFLs can be pumped.

Theorem (Pumping Lemma)

If $L \subseteq \Sigma^*$ is context-free then there exists $p \in \mathbb{N}$ (the pumping length) where, if $w \in L$ with $|w| \geq p$, then w may be split into five pieces, $w = uvxyz$, satisfying the following conditions:

1. $uv^ixy^iz \in L$, *for all* $i \in \mathbb{N}$,
2. $|vy| > 0$, *and*
3. $|vxy| \leq p$.

Proof idea only.

In large enough derivation trees we must necessarily find paths from the root to a terminal symbol that are longer than the number of non-terminals. So there must be a repetition. Take the last repetition, i.e., two occurrence of the same non-terminal. To pump down, cut off the subtree rooted at the first occurrence and insert the subtree rooted at the second occurrence. To pump up, cut at the second occurrence and insert the subtree rooted at the first occurrence. □

Example

Let $L(G_1) = \{\ a^i b^i c^i \mid i > 0\ \}$.

Assume that $L(G_1)$ is a CFL, that is, there is a CFG G with $L(G) = L(G_1)$. We use the pumping lemma to derive a contradiction.

Let p be G's pumping length. Consider $w = \mathtt{a}^p\mathtt{b}^p\mathtt{c}^p$ and let $w = uvxyz$ satisfy the conditions of the pumping lemma.

If vxy is in $\mathtt{a}^*\mathtt{b}^*$ then uxz is not in $L(G_1)$ because, by condition 2 vy contains at least one symbol, so uxz has either fewer than p copies of a or b but exactly c copies of p.
If vxy is in $\mathtt{b}^*\mathtt{c}^*$ we reason analogously.

Due to condition 3 of the pumping lemma there are no other cases. Each case lead to a contradiction. Thus $L(G_1)$ cannot be context-free.

Playing games

Exploits of the pumping lemma can be seen as a game played against an imaginary adversary:

1. We pick $L \subseteq \Sigma^*$.
2. The adversary picks a pumping length $p > 0$.
3. We pick a word $w \in L$ with $|w| \geq p$.
4. The adversary partitions w into $uvxyz$ such that $|vxy| \leq p$ and $vy \neq \epsilon$.
5. We win if we can find an $i \in \mathbb{N}$ such that $uv^ixy^iz \notin L$. Our profit is a proof of L not being context-free.

For playing some instances of the game, check out JFLAP.

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

This may surprise some because $L = \{\ ww^{\mathcal{R}} \mid w \in \{0,1\}^*\ \}$ is a CFL. (Proof eg via the CFG $S \to 0S0 \mid 1S1 \mid \epsilon$.)

Let p be the pumping length for L.

As a rule of thumb, we suggest words w based on p such that only a small number of cases for partitioning w into $uvxyz$ satisfying $|vxy| \leq p$ and $vy \neq \epsilon$ emerge. This is often achieved by making all the different regions in w sufficiently long such that vxy must be located in at most two of them.

Sipser discusses why 0^p10^p1 does *not* work. We note that it doesn't follow the rule of thumb because vxy could range over three of the four regions, e.g., 0^p10^p.

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

This may surprise some because $L = \{\ ww^{\mathcal{R}} \mid w \in \{0,1\}^*\ \}$ is a CFL. (Proof eg via the CFG $S \to 0S0 \mid 1S1 \mid \epsilon$.)

Let p be the pumping length for L.

As a rule of thumb, we suggest words w based on p such that only a small number of cases for partitioning w into $uvxyz$ satisfying $|vxy| \leq p$ and $vy \neq \epsilon$ emerge. This is often achieved by making all the different regions in w sufficiently long such that vxy must be located in at most two of them.

Sipser discusses why 0^p10^p1 does *not* work. We note that it doesn't follow the rule of thumb because vxy could range over three of the four regions, e.g., 0^p10^p.

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

This may surprise some because $L = \{\ ww^{\mathcal{R}} \mid w \in \{0,1\}^*\ \}$ is a CFL. (Proof eg via the CFG $S \to 0S0 \mid 1S1 \mid \epsilon$.)

Let p be the pumping length for L.

As a rule of thumb, we suggest words w based on p such that only a small number of cases for partitioning w into $uvxyz$ satisfying $|vxy| \leq p$ and $vy \neq \epsilon$ emerge. This is often achieved by making all the different regions in w sufficiently long such that vxy must be located in at most two of them.

Sipser discusses why 0^p10^p1 does *not* work. We note that it doesn't follow the rule of thumb because vxy could range over three of the four regions, e.g., 0^p10^p.

$L = \{ \, ww \mid w \in \{0,1\}^* \, \}$ is not a CFL cont.

Instead, we (and Sipser, and everybody else) pick $w = 0^p 1^p 0^p 1^p$. Let $uvxyz = w$ such that $|vxy| \leq p$ and $vy \neq \epsilon$. As in the failed attempt, w contains four regions, but this time vxy can range over at most two of them.

If vxy is located in a single one of the regions, i.e., $vxy \in 0^* \cup 1^*$ pumping either way takes us out of L.

$L = \{\ ww \mid w \in \{0,1\}^*\ \}$ **is not a CFL cont.**

Otherwise, if vxy is stretches across two adjacent regions we distinguish three cases:

1. $u \in 0^*$ and $vxy \in 0^*1^*$, that is, vxy doesn't straddle the midpoint of w but is contained in the first half of w. Pumping down leads to a word the midpoint of which is located inside the third region, 0^p. So the left half of uxz ends in a 0 but the right half ends in a 1. Thus $uxz \notin L$.
2. $u \in 0^p1^p0^*$ and $vxy \in 0^*1^*$, that is, vxy is contained in the second half of w. This case is similar to the previous one.
3. $vxy \in 1^*0^*$, that is, vxy straddles the midpoint of w. Pumping down leads to $uxz = 0^p1^i0^j1^p$ for some $i, j \leq p$ that cannot both be p. Again, $uxz \notin L$. □

Theorem

CFLs are closed under union.

Proof.

Let $L_1, L_2 \subseteq \Sigma^*$ be context-free.

For $i = 1, 2$ let $G_i = (V_i, \Sigma, P_i, S_i)$ be a CFG such that $L(G_i) = L_i$ such that w.l.o.g. $V_1 \cap V_2 = \emptyset$.

Consider $G = (V_1 \cup V_2 \cup \{S\}, \Sigma, P, S)$ for some fresh non-terminal $S \notin V_1 \cup V_2 \cup \Sigma$ where

$$P = P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}$$

This grammar clearly generates $L_1 \cup L_2$. $\square$

Theorem

CFLs are closed under concatenation.

Proof.

With L_1, L_2, G_1, G_2, G as in the previous proof, except for

$$P = P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}$$

This grammar clearly generates $L_1 L_2$. $\square$

Theorem

CFLs are closed under Kleene star (*).

Proof: exercise.

Theorem

CFLs are not closed under intersection.

A counter example suffices. We introduce two languages

$$L_1 = \{\ a^i b^i c^j \mid i > 0 \wedge j > 0\ \}$$
$$L_2 = \{\ a^j b^i c^i \mid i > 0 \wedge j > 0\ \}$$

which are context-free but whose intersection

$$L_1 \cap L_2 = \{\ a^i b^i c^i \mid i > 0\ \}$$

is not as we've shown. It remains to be shown that the L_i are indeed CFLs.

Consider the two CFGs

$$G_1 : S \to AB \qquad\qquad G_2 : S \to AB$$
$$A \to aAb \mid ab \qquad\qquad A \to aA \mid a$$
$$B \to cB \mid c \qquad\qquad B \to bBc \mid bc$$

In G_1, the non-terminal A generates all strings of the form $a^i b^i$, and B generates all strings of cs.

In G_2, the non-terminal B generates all strings of the form $b^i c^i$, and A generates all strings of as. $\square$

Complementation

That CFLs are not closed under complementation follows from de Morgan's laws:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}\ .$$