COS210 - Theoretical Computer Science
Context-Free Languages (Part 2)

Context-Free Grammar: Complement of Non-Regular L

Consider the non-regular language $L = \{0^n1^n : n \geq 0\}$ over $\Sigma = \{0, 1\}$

We want to prove that $\overline{L}$ is **context-free** where $\overline{L}$ is the **complement** of L

What is the possible **form of strings** w in $\overline{L}$?
We can distinguish the following cases:

- $w = 0^m1^n$ where $m < n$ (less 0s than 1s) **(Case 1)**
- $w = 0^m1^n$ where $m > n$ (more 0s than 1s) **(Case 2)**
- w contains the substring 10 (0 occurs after 1) **(Case 3)**

$\overline{L}$ is the **union** of the languages represented by the above cases

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_1 = (V_1, \Sigma, R_1, S_1)$ of the language

$$C_1 \;=\; \{0^m 1^n : m < n\} \;\; \textbf{(Case 1)}$$

less 0s than 1s

If a string w is in C_1, then it is of one of the following forms:

- $w = 1 \quad (S_1 \to 1)$
- it is an element of C_1 with a **1 concatenated at the end** $(S_1 \to S_1 1)$
- it is an element of C_1 with a **0 concatenated to the front** and a **1 concatenated at the end** $(S_1 \to 0 S_1 1)$

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_1 = (V_1, \Sigma, R_1, S_1)$ of the language

$$C_1 = \{0^m 1^n : m < n\} \quad \textbf{(Case 1)}$$

less 0s than 1s

If a string w is in C_1, then it is of one of the following forms:

- $w = 1 \quad (S_1 \to 1)$
- it is an element of C_1 with a **1 concatenated at the end** $(S_1 \to S_1 1)$
- it is an element of C_1 with a **0 concatenated to the front** and a **1 concatenated at the end** $(S_1 \to 0S_1 1)$

So we can build any string in C_1 with the following **composite rule**:

$$S_1 \to 1 | S_1 1 | 0 S_1 1$$

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_2 = (V_2, \Sigma, R_2, S_2)$ of the language

$$C_2 \;=\; \{0^m 1^n : m > n\} \quad \textbf{(Case 2)}$$

more 0s than 1s

If a string w is in C_2, then it is of one of the following forms:

- $w = 0 \quad (S_2 \to 0)$
- it is an element of C_2 with a **0 concatenated at front** $(S_2 \to 0S_2)$
- it is an element of C_2 with a **0 concatenated to the front** and a **1 concatenated at the end** $(S_2 \to 0S_21)$

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_2 = (V_2, \Sigma, R_2, S_2)$ of the language

$$C_2 \;=\; \{0^m 1^n : m > n\} \quad \textbf{(Case 2)}$$

more 0s than 1s

If a string w is in C_2, then it is of one of the following forms:

- $w = 0 \quad (S_2 \to 0)$
- it is an element of C_2 with a **0 concatenated at front** $(S_2 \to 0S_2)$
- it is an element of C_2 with a **0 concatenated to the front** and a **1 concatenated at the end** $(S_2 \to 0S_21)$

So we can build any string in C_2 with the following **composite rule**:

$$S_2 \to 0|0S_2|0S_21$$

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_3 = (V_3, \Sigma, R_3, S_3)$ of the language

$$C_3 = \{w \in \Sigma^* : w \text{ contains the substring } 10\} \textbf{ (Case 3)}$$

0 occurs after 1

If a string w is in C_3, then it is of the following form:

- $w = X10X'$ where X and X' are **arbitrary strings** over Σ

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_3 = (V_3, \Sigma, R_3, S_3)$ of the language

$$C_3 = \{w \in \Sigma^* : w \text{ contains the substring } 10\} \textbf{ (Case 3)}$$

0 occurs after 1

If a string w is in C_3, then it is of the following form:

- $w = X10X'$ where X and X' are **arbitrary strings** over Σ

Rule for building arbitrary strings:

$$X \rightarrow \epsilon | 0X | 1X$$

Context-Free Grammar: Complement of Non-Regular L

Defining the grammar $G_3 = (V_3, \Sigma, R_3, S_3)$ of the language

$$C_3 = \{w \in \Sigma^* : w \text{ contains the substring } 10\} \textbf{ (Case 3)}$$

0 occurs after 1

If a string w is in C_3, then it is of the following form:

- $w = X10X'$ where X and X' are **arbitrary strings** over Σ

Rule for building arbitrary strings:

$$X \rightarrow \epsilon|0X|1X$$

So we can build any string in C_3 with the following two rules:

- $S_3 \rightarrow X10X$
- $X \rightarrow \epsilon|0X|1X$

Context-Free Grammar: Complement of Non-Regular L

Now we combine the rules to a grammar that defines the **union** of the cases 1, 2 and 3

Let $G = (V, \Sigma, R, S)$ with $V = \{S, S_1, S_2, S_3, X\}$, $\Sigma = \{0, 1\}$, and

$S \to S_1 \mid S_2 \mid S_3$	**(Choose Case)**
$S_1 \to 1 \mid S_1 1 \mid 0 S_1 1$	**(Case 1)**
$S_2 \to 0 \mid 0 S_2 \mid 0 S_2 1$	**(Case 2)**
$X \to \epsilon \mid 0X \mid 1X$	**(Case 3)**
$S_3 \to X10X$	**(Case 3)**

We know that $L(G) = \overline{L}$ because

- $S_1 \overset{*}{\Longrightarrow} 0^m 1^n$, for all $m < n$
- $S_2 \overset{*}{\Longrightarrow} 0^m 1^n$, for all $m > n$
- $X \overset{*}{\Longrightarrow} u$, for all $u \in \Sigma^*$
- $S_3 \overset{*}{\Longrightarrow} w$, w contains the substring 10

Context-Free Grammar: Verification of Addition

Consider the following language

$$L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$$

number of c's is the **sum** *of number of a's and number of b's*

This language is **non-regular** (proof via pumping lemma)

However, we will show that the language is **context-free**

- there often exist **multiple context-free grammars** for a given language
- we will construct one such grammar for L and the **reduce** it in terms of the number of rules

Context-Free Grammar: Verification of Addition

$$L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$$

We can build any string of L by considering the following cases:

1) $\epsilon = a^0 b^0 c^{0+0} \in L$
   $(S \to \epsilon)$

2) $a^n c^n = a^n b^0 c^{n+0} \in L$
   $(S \to A, \;\; A \to \epsilon | aAc)$

3) $b^m c^m = a^0 b^m c^{0+m} \in L$
   $(S \to B, \;\; B \to \epsilon | bBc)$

4) $a^n b^m c^m c^n = a^n b^m c^{n+m} \in L$
   $(S \to A, \;\; A \to \epsilon | aAc | B, \;\; B \to \epsilon | bBc)$

Example for Case 4:
S
A
aAc
a**aAc**c
aa**B**cc
aa**bBc**cc
aab**ϵ**ccc = aabccc

Context-Free Grammar: Verification of Addition

$$L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$$

So we can describe the language L with the following grammar.
$G = (V, \Sigma, R, S)$, $V = \{S, A, B\}$, and $\Sigma = \{a, b, c\}$, and the following R

$$S \to \epsilon | A | B$$
$$A \to \epsilon | aAc | B$$
$$B \to \epsilon | bBc$$

This works because

- $A \stackrel{*}{\Longrightarrow} a^n B c^n$, for any $n \geq 0$
- $B \stackrel{*}{\Longrightarrow} b^m c^m$, for any $m \geq 0$

so

- $S \stackrel{*}{\Longrightarrow} a^n b^m c^m c^n = a^n b^m c^{n+m}$
- and no other $w \in \{a, b, c\}^*$ can be derived from S.
- therefore $L(G) = L$

Context-Free Grammar: Verification of Addition

While the grammar G works we can actually simplify the grammar.

- Observe that $S \implies A \implies B \implies \epsilon$
- So we don't need the ϵ-option for the S-rules and A-rules
- Moreover, we don't need the B-option for the S-rule
- Hence, we can replace the set of rules R by R'

R :

$$S \to \epsilon|A|B$$
$$A \to \epsilon|aAc|B$$
$$B \to \epsilon|bBc$$

R' :

$$S \to A$$
$$A \to aAc|B$$
$$B \to \epsilon|bBc$$

Context-Free Grammar: Verification of Addition

While the grammar G works we can actually simplify the grammar.

- Observe that $S \implies A \implies B \implies \epsilon$
- So we don't need the ϵ-option for the S-rules and A-rules
- Moreover, we don't need the B-option for the S-rule
- Hence, we can replace the set of rules R by R'

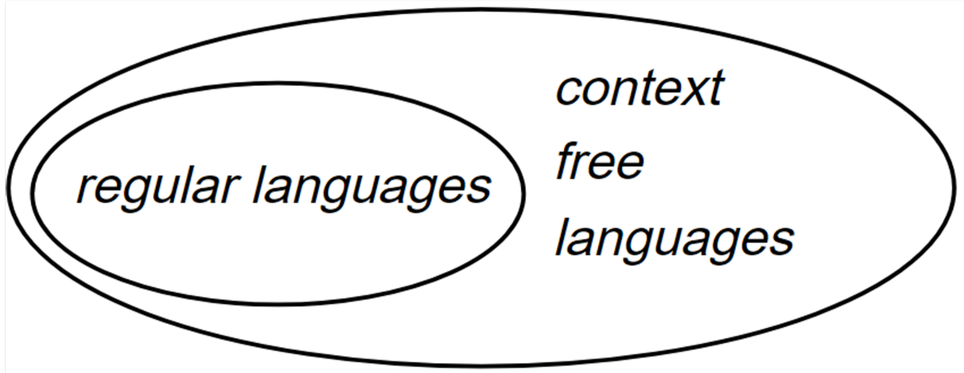
$R:$	$R':$
$S \to \epsilon \vert A \vert B$	$S \to A$
$A \to \epsilon \vert aAc \vert B$	$A \to aAc \vert B$
$B \to \epsilon \vert bBc$	$B \to \epsilon \vert bBc$

But is it also clear now that we don't need S, so we can rather use the grammar $G'' = (V, \Sigma, R'', A)$ with the following R'' :

$$A \to aAc|B$$
$$B \to \epsilon|bBc$$

Regular Languages are Context-Free

We have seen that **not** all context-free languages are regular



Example:

$$L = \{0^n 1^n : n \geq 0\}$$

However, **all regular languages are context-free** which we will prove now

Regular Languages are Context-Free: Proof

Theorem

Let L be a **regular language** *over an alphabet Σ, then L is* **context-free**

Proof:

- Since L over Σ is **regular** there exists a **deterministic finite automaton** $M = (Q, \Sigma, \delta, q, F)$ with $L(M) = L$
- In order to show that L is **context-free** we construct a **grammar** $G = (V, \Sigma, R, S)$ such that
  - $L(G) = L(M) = L$, which means
  - $w \in L(M)$ if and only if $w \in L(G)$, which means
  - M accepts w if and only if $S \overset{*}{\Longrightarrow} w$

Regular languages are context-free: Proof

Proof cont:
We construct the grammar $G = (V, \Sigma, R, S)$ based on the automaton $M = (Q, \Sigma, \delta, q, F)$ with $L(M) = L$ as follows:

- $V = Q$, i.e. the variables of G are the states of M,
- $S = q$, i.e. the start variable of G is the initial state of M,
- R consists of the following rules:
  - for all $A, B \in Q$, $a \in \Sigma$ with $\delta(A, a) = B$
    $A \rightarrow aB$ is a rule
    (each transition $A \xrightarrow{a} B$ becomes a rule $A \rightarrow aB$)
  - for all $A \in F$ where F are the accepting states of M
    $A \rightarrow \epsilon$ is a rule

Our constructed grammar looks as follows: $G = (Q, \Sigma, R, q)$

Regular Languages are Context-Free: Proof

Proof cont:

- Now that we have constructed the grammar G we need to show that:

$$L(G) = L(M)$$

$$(M \text{ accepts } w \text{ if and only if } S \overset{*}{\Longrightarrow} w)$$

- We do this by proving that for all $w \in \Sigma^*$

$$w \in \Sigma^* : w \in L(M) \Rightarrow w \in L(G) \quad \textbf{(Part 1)}$$

$$\text{and} \quad w \in \Sigma^* : w \in L(G) \Rightarrow w \in L(M) \quad \textbf{(Part 2)}$$

Regular Languages are Context-Free: Proof of Part 1

Show that for all $w \in \Sigma^* : w \in L(M) \Rightarrow w \in L(G)$

- Let $w = w_1 w_2 \ldots w_n$ be an arbitrary string in $L(M)$
- When M reads in w it visits the states $r_0, r_1, \ldots r_n$ where:
  - $r_0 = q$, $r_n \in F$, and
  - $r_{i+1} = \delta(r_i, w_{i+1})$ for $0 \leq i \leq n-1$

$$\underbrace{r_0}_{=q} \xrightarrow{w_1} r_1 \xrightarrow{w_2} \ldots \xrightarrow{w_{n-1}} r_{n-1} \xrightarrow{w_n} \underbrace{r_n}_{\in F}$$

- Given our construction of G we have the following rules in R
  - $r_i \rightarrow w_{i+1} r_{i+1}$ for all $0 \leq i \leq n-1$, and
  - $r_n \rightarrow \epsilon$
- We can derive w as follows:

$$\begin{aligned} S = q = r_0 &\Longrightarrow w_1 r_1 \\ &\Longrightarrow w_1 w_2 r_2 \\ &\ldots \\ &\Longrightarrow w_1 w_2 \ldots w_n r_n \\ &\Longrightarrow w_1 w_2 \ldots w_n \end{aligned}$$

- Consequently, $w_1 w_2 \ldots w_n \in L(G)$

Regular Languages are Context-Free: Proof of Part 2

Show that for all $w \in \Sigma^* : w \in L(G) \Rightarrow w \in L(M)$

- Let w be an arbitrary string of $L(G)$
- This means that $S \stackrel{*}{\Longrightarrow} w$, and S is equal to initial state of M
- We have that $S \neq w$ because a string w of $L(G)$ only contains terminals, whereas S is a variable
- Hence, $S \stackrel{*}{\Longrightarrow} w$ can only mean that
  - there exists a $k \geq 2$ and a sequence $u_1, u_2 \ldots, u_k$ in $(V \cup \Sigma)^*$ with
  - $S = u_1$, $w = u_k$, and $u_1 \Longrightarrow u_2 \Longrightarrow \ldots \Longrightarrow u_k$
- Each one step derivation ($\Longrightarrow$) must of one of the following forms
  - $A \to aB$, where $A, B \in V = Q$ and $B = \delta(A, a)$
  - $A \to \epsilon$
- Based on this we can construct a path in M by considering that for $u_1 \Longrightarrow u_2$ we would use a rule of the form

$$U_1 \to a_1 U_2 \text{ where } U_1, U_2 \in V = Q \text{ and } U_2 = \delta(U_1, a_1)$$

Regular Languages are Context-Free: Proof of Part 2

- $u_1 \implies u_2$ means

$$U_1 \to a_1 U_2 \text{ where } U_1, U_2 \in V = Q \text{ and } U_2 = \delta(U_1, a_1)$$

- This corresponds to taking the transition $U_1 \xrightarrow{a_1} U_2$ in M
- If we repeat this for the remaining positions of $u_1, \ldots, u_k$, we obtain a run from the initial state S of the DFA M to an accepting state of M
  - The last state U_k of the run is accepting

    since in order for u_k to be terminal (in Σ^*) the rule of the form $U_k \to \epsilon$ would have been used
  - and only states in the accepting set F where given an ϵ-rule
  - It follows that $w \in L(M)$

- We now have that for all $w \in \Sigma^* : w \in L(G) \Rightarrow w \in L(M)$ and $w \in L(M) \Rightarrow w \in L(G)$
- It follows that $L(M) = L(G)$ $\square$

Regular Languages are Context-Free

Theorem

Let L be a **regular language** *over an alphabet Σ, then L is* **context-free**