Advanced Automata Theory 1 Chomsky Hierarchy and Grammars

Frank Stephan
Department of Computer Science
Department of Mathematics
National University of Singapore
fstephan@comp.nus.edu.sg
Languages

Language = Set of Strings over an Alphabet. Alphabet $\Sigma$, for example $\Sigma = \{0, 1, 2\}$. Always finite.

Finite languages
$L_1 = \emptyset$, no elements.
$L_2 = \{\varepsilon\}$, set consisting of empty string.
$L_3 = \{00, 01, 02, 10, 11, 12, 20, 21, 22\}$, all elements of length 2.
$L_4 = \{\varepsilon, 0, 00, 000, 0000\}$, all strings of 0s up to length 4.
$L_5 = \{01, 001, 02, 002\}$, all strings consisting of one or two 0s followed by a 1 or 2.
Operations with Languages

Union:
\[ L \cup H = \{ u : u \in L \lor u \in H \}; \]
\[ \{00, 01, 02\} \cup \{01, 11, 21\} = \{00, 01, 02, 11, 21\}; \]
\[ \{0, 00, 000\} \cup \{00, 000, 0000\} = \{0, 00, 000, 0000\}. \]

Intersection:
\[ L \cap H = \{ u : u \in L \land u \in H \}; \]
\[ \{0, 00, 000\} \cap \{00, 000, 0000\} = \{00, 000\}; \]
\[ \{00, 01, 02\} \cap \{01, 11, 21\} = \{01\}. \]

Set Difference:
\[ L - H = \{ u : u \in L \land u \notin H \}; \]
\[ \{00, 01, 02\} - \{01, 11, 21\} = \{00, 02\}. \]

Concatenation:
\[ 000 \cdot 1122 = 0001122; \]
\[ L \cdot H = \{ v \cdot w : v \in L \land w \in H \}; \]
\[ \{0, 00\} \cdot \{1, 2\} = \{01, 001, 02, 002\}. \]
Kleene Star and Plus

Definition
$L^* = \{\varepsilon\} \cup L \cup L \cdot L \cup L \cdot L \cdot L \cup \ldots$
$= \{w_1 \cdot w_2 \cdot \ldots \cdot w_n : n \geq 0 \land w_1, w_2, \ldots, w_n \in L\}$;
$L^+ = L \cup L \cdot L \cup L \cdot L \cdot L \cup \ldots$
$= \{w_1 \cdot w_2 \cdot \ldots \cdot w_n : n > 0 \land w_1, w_2, \ldots, w_n \in L\}$.

Examples
$\emptyset^* = \{\varepsilon\}$.
$\Sigma^*$ is the set of all words over $\Sigma$.
$\{0\}^* = \{\varepsilon, 0, 00, 000, 0000, \ldots\}$.
$\{00, 01, 10, 11\}^*$ are all binary words of even length.
$\varepsilon \in L^+$ iff $\varepsilon \in L$.

Notation
Often $w^*$ in place of $\{w\}^*$;
Often $w \cdot L$ in place of $\{w\} \cdot L$. 
Regular Languages

Regular expressions are either finite sets listed by their elements or obtained from other regular expressions by forming the Kleene star, Kleene plus, union, intersection, set-difference or concatenation.

A language is regular iff it can be described by a regular expression.

Regular sets have many different regular expressions.

For example, \{0, 00\} \cdot \{1, 2\} and \{01, 001, 02, 002\} describe the same set. Also 0* and (00)* \cup 0 \cdot (00)* describe the same set.

Intersections and set difference are traditionally not used in regular expressions, as they can be replaced by combining other operations.

The complement of a language \(L\) is \(\Sigma^* - L\).
Quiz

Which of the following regular expressions describe the same set?

1. \( \{00, 000\}^*; \)
2. \( \{000, 0000\}^*; \)
3. \( 00 \cdot 0^*; \)
4. \( 000 \cdot 0^*; \)
5. \( \{000, 0000\} \cup (000000 \cdot 0^*); \)
6. \( \{00, 01, 02, 10, 11, 12\}; \)
7. \( 0^*1^*2^*; \)
8. \( (0^*1^*2^*)^*; \)
9. \( (\{0, 1\} \cdot \{0, 1, 2\}^*) \cap (\{0, 1, 2\} \cdot \{0, 1, 2\}). \)
Grammars

Grammar \( (N, \Sigma, P, S) \) describes how to generate the words in a language; the language \( L \) of a grammar consists of all the words in \( \Sigma^* \) which can be generated.

**N**: Non-terminal alphabet, disjoint to \( \Sigma \).

**S \in N** is the start symbol.

**P** consists of rules \( l \rightarrow r \) with each rule having at least one symbol of \( N \) in the word \( l \).

\( v \Rightarrow w \) iff there are \( x, y \) and rule \( l \rightarrow r \) in \( P \) with \( v = xly \) and \( w = xry \). \( v \Rightarrow^* w \): several such steps.

The grammar with \( N = \{S\} \), \( \Sigma = \{0, 1\} \) and \( P = \{S \rightarrow SS, S \rightarrow 0, S \rightarrow 1\} \) permits to generate all nonempty binary strings.

\( S \Rightarrow SS \Rightarrow SSS \Rightarrow 0SS \Rightarrow 01S \Rightarrow 011 \).
Examples

Example 1.6
At least three symbols, \(0\)s followed by \(1\)s, at least one \(0\) and one \(1\).
\[N = \{S, T\}, \Sigma = \{0, 1\}, \text{ startsymbol } S, \ P \text{ has } S \to 0T1,\]
\[T \to 0T, \ T \to T1, \ T \to 0, \ T \to 1.\]

Example 1.7
All words with as many \(0\)s as \(1\)s.
\[N = \{S\}, \Sigma = \{0, 1\}, S \to SS|0S1|1S0|\varepsilon.\]
The symbol | separates alternatives.

Example 1.8
All words of odd length.
\[N = \{S, T\}, \Sigma = \{0, 1, 2\}, \text{ startsymbol } S,\]
\[S \Rightarrow 0T|1T|2T|0|1|2, \ T = 0S|1S|2S.\]
The Chomsky Hierarchy

Grammar \((N, \Sigma, P, S)\) generating \(L\).

CH0: No restriction. Generates all recursively enumerable languages.

CH1 (context-sensitive): Every rule is of the form \(uAw \to uvw\) with \(A \in N\), \(u, v, w \in (N \cup \Sigma)^*\) and \(v = \varepsilon\) is only possible if \(A = S\) and \(S\) does not occur on any right side of a rule.

Easier formalisation: If \(l \to r\) is a rule then \(|l| \leq |r|\), that is, \(r\) is at least as long as \(l\). Special rule (as above) for the case that \(\varepsilon \in L\).

CH2 (context-free): Every rule is of the form \(A \to w\) with \(A \in N\) and \(w \in (N \cup \Sigma)^*\).

CH3 (regular): Every rule is of the form \(A \to wB\) or \(A \to w\) with \(A, B \in N\) and \(w \in \Sigma^*\).

\(L\) is called context-sensitive / context-free / regular iff it can
Examples

Regular grammar for Example 1.6:
\( N = \{S, T\}, \Sigma = \{0, 1\}, \text{ startsymbol } S, S \rightarrow 0S|00T|01T, T \rightarrow 1T|1. \)

Grammar for Example 1.7 is context-free.
Grammar for Example 1.8 is regular.

Example 1.12.
Context-Sensitive Grammar for \( \{0^n1^n2^n : n \in \mathbb{N}\} \).
\( N = \{S, T, U\}, \Sigma = \{0, 1, 2\}, \text{ startsymbol } S, S \rightarrow 012|0T12|\varepsilon, T \rightarrow 0T1U|01U, U1 \rightarrow 1U, U2 \rightarrow 22. \)
Regular Grammar ⇒ Expression

Regular grammar $(\{S, T\}, \{0, 1, 2, 3\}, P, S)$ with $S \to 0S|1T|2$ and $T \to 0T|1S|3$.

Let $L_S = \{w : (S \to w) \in P\} = \{2\}$ and $L_T = \{3\}$.

Let $L_{S,S} = \{w : (S \to wS) \in P\} = \{0\}$, $L_{S,T} = \{1\}$, $L_{T,S} = \{1\}$, $L_{T,T} = \{0\}$.

Regular Expression:
$$(L_{S,S})^* \cdot (L_{S,T} \cdot (L_{T,T})^* \cdot L_{T,S} \cdot (L_{S,S})^*)^* \cdot (L_S \cup L_{S,T} \cdot (L_{T,T})^* \cdot L_T)$$
giving $0^* \cdot (10^*10^*)^* \cdot (2 \cup 10^*3)$.

Equivalent expression:
$$(L_{S,S} \cup L_{S,T} \cdot (L_{T,T})^* \cdot L_{T,S})^* \cdot (L_S \cup L_{S,T} \cdot (L_{T,T})^* \cdot L_T)$$
giving $(0 \cup 10^*1)^* \cdot (2 \cup 10^*3)$.
Regular Expression ⇒ Grammar

Given \((\{0,1\}^* \cdot 2 \cdot \{0,1\}^* \cdot 2) \cup \{0,2\}^* \cup \{1,2\}^*\).

Choose Non-Terminals \(S, T, U, V, W\) with

\[
L_S = L_T \cup L_V \cup L_W;
\]
\[
L_T = \{0,1\}^* \cdot 2 \cdot \{0,1\}^* \cdot 2 = \{0,1\}^* \cdot 2 \cdot L_U;
\]
\[
L_U = \{0,1\}^* \cdot 2;
\]
\[
L_V = \{0,2\}^*;
\]
\[
L_W = \{1,2\}^*.
\]

Grammar \((\{S, T, U, V, W\}, \{0,1,2\}, P, S)\) with these rules:

\[
S \rightarrow T|V|W,
\]
\[
T \rightarrow 0T|1T|2U,
\]
\[
U \rightarrow 0U|1U|2,
\]
\[
V \rightarrow 0V|2V|\varepsilon,
\]
\[
W \rightarrow 1W|2W|\varepsilon.
\]
The Pumping Lemma

Theorem
Let $L$ be a regular language. There is a constant $k$ such that every $w \in L$ with $|w| > k$ equals to $xyz$ with $y \neq \varepsilon$ and $|xy| \leq k$ and $xy^*z \subseteq L$.

Tighter versions will be shown later.

Example
$L = 0110 \cdot \{2, 3\}^* \cup 001100 \cdot \{22, 33\}^* \cdot 11 \cup 0011001100 \cdot \{2, 3\}$. Then constant $k$ is 11.
If $w \in L$ and $|w| > 11$ then there are at least two occurrences of $2, 3$ in $w$.
So split $w$ into $xyz$ such that $y$ is the first block of two digits from $2, 3$ occuring in $w$.
Then $xy^*z \subseteq L$. 

Advanced Automata Theory 1 Chomsky Hierarchy and Grammars – p. 13
Structural Induction

To show that all regular sets satisfy the Pumping Lemma, one does the following.

Show that $H$ satisfies the Pumping Lemma for all finite $H$.

Show that if $H$ satisfies the Pumping Lemma, so does $H^*$.

Show that if $H_1, H_2$ satisfy the Pumping Lemma, so does $H_1 \cdot H_2$.

Show that if $H_1, H_2$ satisfy the Pumping Lemma, so does $H_1 \cup H_2$.

Other operations (intersection, set difference) can be ignored, as one can make all regular languages without using them.
**Pumping Position and Length**

**Examples 1.21**

Let \( L = \{ w \in \{ 0, 1 \} : w \) has as many 0s as 1s \}. \]

Satisfies Pumping-Lemma Without Constraint on Pumping-Position and Length.

Given \( w \in \{ 0, 1 \}^* - \{ 0 \}^* - \{ 1 \}^* \). Then \( w = xyz \) with \( y \in \{ 01, 10 \} \)

If \( w \in L \) then \( xy^*z \subseteq L \).

Does not Satisfy Pumping Lemma with Constraint on Pumping-Position and Length.

Let \( k \) be the pumping constant and consider \( 0^{k+1}1^{k+1} \). Pumping before position \( k \) expands or reduces the number of 0s without keeping the number of 1s the same.
Context-Free Languages

Pumping-Lemma for Context-Free Languages
Assume that $L$ is a context-free language. Then there is a constant $k$ such that for all $u \in L$ with $|u| > k$ there is a representation $vwxyz$ of $u$ with $|wxy| \leq k$ and $w \neq \varepsilon \lor y \neq \varepsilon$ and $vw^nxynz \in L$ for all $n \in \mathbb{N}$.

Applications
Showing that certain languages are not context-free or regular.

$L = \{u : u$ is a decimal number where every digit appears as often as the other digits$\}$. This language is not context-free.

$L = \{3^n7^n : n \in \{1, 2, 3, \ldots \}\}$. This language is context-free but not regular.
Example 1.20
The set \( L = \{0^p : p \text{ is a prime} \} \) is not context-free.

Let \( k \) be the pumping constant and \( p \) be a prime number larger than \( k \).

Now \( 0^p = vwxyz \) with \( wy \neq \varepsilon \) and \( vw^rxy^rz \in L \) for all \( r \).

Let \( q = |wy| \), note that \( q > 0 \).

Now \( vw^{p+1}xy^{p+1}z \in L \) and has length \( p + p \times q \).

This is \( p \times (1 + q) \) and is not a prime.

Hence \( 0^{p+p\times q} \notin L \), a contradiction to the Pumping Lemma.

So \( L \) does not satisfy the Pumping Lemma for context-free languages.
Exercise 1.22

Exercise
Let \( L = \{0^n 1^n 2^n : n \in \mathbb{N}\} \).

Show that this language is not context-free using the Pumping Lemma for context-free languages.

Comment
This is a classical result and standard exercise in the field. This example often comes up and it is useful to remember it. It will be used in varied form for various further results.
Exercise 1.23

Let $L \subseteq \{0\}^*$. Show that the following conditions are equivalent for $L$.

1. $L$ is regular;
2. $L$ is context-free;
3. $L$ satisfies the Pumping Lemma for regular languages;
4. $L$ satisfies the Pumping Lemma for context-free languages.

Show that, however, the bound on the length of the pumped word is necessary. Show that $\{0^p : p \text{ is not a power of two}\}$ satisfies the Pumping Lemma without that bound, although its not regular.
Exercise 1.24

Let \( f(n) = a \ast n \ast n + b \ast n + c \) with \( a, b, c \in \mathbb{N} \) and consider
\( L = \{ 0^n1^{f(n)} : n \in \mathbb{N} \} \).

1. Case \( a = 0 \land b = 0 \): Show that \( L \) is regular in this case.

2. Case \( a = 0 \land b > 0 \): Show that \( L \) is context-free by giving the corresponding grammar; show that \( L \) is not regular using the regular Pumping Lemma.

3. Case \( a > 0 \) — Consider the specific case \( a = 1, b = 2, c = 4 \) (the others would be similar): Show that \( L \) is context-sensitive but not context-free; for context-sensitivity, it is sufficient if the rules \( l \rightarrow r \) just satisfy \( |l| \leq |r| \); for not being context-free, use the context-free Pumping-Lemma.
Exercise 1.25

Let $L = \{0^n1^m2^k : n < m \lor m < k\}$ and $H = \{0^n1^m2^{n+m} : n, m \in \mathbb{N}\}$.

Construct context-free grammars for both languages.

Comment

These examples are also typical examples for context-free languages. Although one controls three items, namely the number of 0, number of 1 and number of 2, these are arranged in a way that one at each time only needs to take care of the dependency between two of them. Therefore the languages are still context-free.