Theory of Computation 2
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 finite Alphabet $\mathbf{\Sigma}$.
Examples for Finite Languages: $\emptyset$, $\{\varepsilon, \mathbf{0}, \mathbf{11}\}$.

Union: $\mathbf{L} \cup \mathbf{H} = \{\mathbf{u} : \mathbf{u} \in \mathbf{L} \textbf{ or } \mathbf{u} \in \mathbf{H}\}$.
Intersection: $\mathbf{L} \cap \mathbf{H} = \{\mathbf{u} : \mathbf{u} \in \mathbf{L} \textbf{ and } \mathbf{u} \in \mathbf{H}\}$.
Set Difference: $\mathbf{L} - \mathbf{H} = \{\mathbf{u} : \mathbf{u} \in \mathbf{L} \textbf{ and } \mathbf{u} \notin \mathbf{H}\}$.
Concatenation: $\mathbf{000} \cdot \mathbf{1122} = \mathbf{0001122}$;
$\mathbf{L} \cdot \mathbf{H} = \{\mathbf{v} \cdot \mathbf{w} : \mathbf{v} \in \mathbf{L} \textbf{ and } \mathbf{w} \in \mathbf{H}\}$.
Kleene Star: $\mathbf{L}^* = \{\varepsilon\} \cup \mathbf{L} \cup \mathbf{L} \cdot \mathbf{L} \cup \mathbf{L} \cdot \mathbf{L} \cdot \mathbf{L} \cup \ldots$
$= \{\mathbf{w_1} \cdot \mathbf{w_2} \cdot \ldots \cdot \mathbf{w_n} : \mathbf{n} \geq \mathbf{0} \textbf{ and } \mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n} \in \mathbf{L}\}$.
Kleene Plus: $\mathbf{L}^+ = \mathbf{L} \cup \mathbf{L} \cdot \mathbf{L} \cup \mathbf{L} \cdot \mathbf{L} \cdot \mathbf{L} \cup \ldots$
$= \{\mathbf{w_1} \cdot \mathbf{w_2} \cdot \ldots \cdot \mathbf{w_n} : \mathbf{n} > \mathbf{0} \textbf{ and } \mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n} \in \mathbf{L}\}$.

Regular languages are those which can be formed from finite languages using union, concatenation and Kleene star. Regular expressions write down such definitions explictly.

Growth

A set $\mathbf{L}$ has polynomial growth iff there is a polynomial $\mathbf{p}$ such that for all $\mathbf{n}$, there are at most $\mathbf{p(n)}$ words in $\mathbf{L}$ which are shorter than $\mathbf{n}$.
Examples are $\{\mathbf{00}\}^*$, $\{\mathbf{0}\}^* \cup \{\mathbf{1}\}^*$ and $\{\mathbf{0}\}^* \cdot \{\mathbf{1}\}^* \cdot \{\mathbf{2}\}^*$.
A set $\mathbf{L}$ has exponential growth iff there are constants $\mathbf{h}, \mathbf{k}$ such that for each $\mathbf{n}$ there are at least $\mathbf{2^n}$ words in $\mathbf{L}$ which are shorter than $\mathbf{hn + k}$.
Examples are $\{\mathbf{0}, \mathbf{1}\}^*$ and $\{\mathbf{0000}, \mathbf{1111}\}^* \cdot \{\mathbf{2222}\}$.

Theorem
Every regular set has either polynomial or exponential growth.

Proof by structural induction: Let $\mathbf{P(L)}$ denote that $\mathbf{L}$ has either polynomial or exponential growth.

$\mathbf{P(L)}$ is true for all finite sets $\mathbf{L}$. If $\mathbf{P(L)}$ and $\mathbf{P(H)}$ are true, so are $\mathbf{P(L \cup H)}$, $\mathbf{P(L \cdot H)}$ and $\mathbf{P(L^*)}$.

Grammars

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

$\mathbf{N}$: Non-terminal alphabet, disjoint to $\mathbf{\Sigma}$.

$\mathbf{S} \in \mathbf{N}$ is the start symbol.

$\mathbf{P}$ consists of rules $\mathbf{l} \to \mathbf{r}$ with each rule having at least one symbol of $\mathbf{N}$ in the word $\mathbf{l}$.

$\mathbf{v} \Rightarrow \mathbf{w}$ iff there are $\mathbf{x}, \mathbf{y}$ and rule $\mathbf{l} \to \mathbf{r}$ in $\mathbf{P}$ with $\mathbf{v} = \mathbf{xly}$ and $\mathbf{w} = \mathbf{xry}$. $\mathbf{v} \Rightarrow^* \mathbf{w}$: several such steps.

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

$\mathbf{S} \Rightarrow \mathbf{SS} \Rightarrow \mathbf{SSS} \Rightarrow \mathbf{0SS} \Rightarrow \mathbf{01S} \Rightarrow \mathbf{011}$.

Conventions 1

These conventions are followed often, but not strictly.

Terminal symbols or alphabet symbols: $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots$;
Variables for terminals: $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$;
Variables for sets of alphabets: $\boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \boldsymbol{\Delta}$;

Non-Terminals: $\mathbf{S}, \mathbf{T}, \ldots$; Start symbol $\mathbf{S}$;
Variables for terminals: $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$;

Words over terminals:
$\varepsilon, \mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots, \mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{02}, \mathbf{11}, \mathbf{20}, \mathbf{03}, \ldots$;
Symbol repetition in words: $\mathbf{0^8 1^9}$, $\mathbf{0^n 1^n 2^n}$;
Variables for words over terminals: $\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$;
Words over non-terminals and terminals: same letters, sometimes capital.

Conventions 2

Rules: $\mathbf{S \to 00|11|ST|T, T \to TT|222}$;
Variables for rules: $\mathbf{l \to r}$, $\mathbf{A \to w}$;
Derivation one step: $\mathbf{v \Rightarrow w}$;
Derivation arbitrary steps (including none): $\mathbf{v \Rightarrow^* w}$;

Languages: $\emptyset$, $\{\mathbf{00, 11}\}$, $\mathbf{\Sigma^*}$, $\{\mathbf{0, 1}\}^*$, regular expressions;
Variables for Languages: $\mathbf{L, H, K, \ldots}$;

States of automata (next lecture): $\mathbf{o, p, q, r, s, t}$; start: $\mathbf{s}$;
Automata transition function: δ;
Variables of states of automata: often same symbols;

Natural numbers:
$\mathbf{0, 1, 2, \ldots, 9, 10, 11, \ldots, 99, 100, 101, 102, \ldots}$;
Variables of natural numbers: $\mathbf{m, n, i, j, k, \ldots}$;

Variables for functions: $\mathbf{f, g, h, \ldots}$;
Set of natural numbers: $\mathbb{N}$.

Examples

Example 2.3

At least three symbols, $\mathbf{0}$s followed by $\mathbf{1}$s, at least one $\mathbf{0}$ and one $\mathbf{1}$.

$\mathbf{N} = \{\mathbf{S}, \mathbf{T}\}$, $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}\}$, start symbol $\mathbf{S}$,

$\mathbf{P}$ has $\mathbf{S} \to \mathbf{0T1}$, $\mathbf{T} \to \mathbf{0T}$, $\mathbf{T} \to \mathbf{T1}$, $\mathbf{T} \to \mathbf{0}$, $\mathbf{T} \to \mathbf{1}$.

Example 2.4

All words with as many $\mathbf{0}$s as $\mathbf{1}$s.

$\mathbf{N} = \{\mathbf{S}\}$, $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}\}$, $\mathbf{S} \to \mathbf{SS}|\mathbf{0S1}|\mathbf{1S0}|\varepsilon$.

The symbol $|$ separates alternatives.

Example 2.5

All words of odd length.

$\mathbf{N} = \{\mathbf{S}, \mathbf{T}\}$, $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$, start symbol $\mathbf{S}$,

$\mathbf{S} \to \mathbf{0T}|\mathbf{1T}|\mathbf{2T}|\mathbf{0}|\mathbf{1}|\mathbf{2}$, $\mathbf{T} \to \mathbf{0S}|\mathbf{1S}|\mathbf{2S}$.

The Chomsky Hierarchy

Grammar $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ generating $\mathbf{L}$.

CH0: No restriction, all recursively enumerable languages.

CH1 (context-sensitive): Every rule is of the form $\mathbf{uAw} \rightarrow \mathbf{uvw}$ with $\mathbf{A} \in \mathbf{N}$, $\mathbf{u}, \mathbf{v}, \mathbf{w} \in (\mathbf{N} \cup \mathbf{\Sigma})^*$ and $\mathbf{v} = \varepsilon$ is only possible if $\mathbf{A} = \mathbf{S}$ and $\mathbf{S}$ never on right side of a rule.

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

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

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

$\mathbf{L}$ is called context-sensitive / context-free / regular iff it can be generated by a grammar of respective type.

Examples

Recall: A language $\mathbf{L}$ is called context-sensitive / context-free / regular iff it can be generated by a grammar of respective type.

Regular grammar for Example 2.3:
$\mathbf{N} = \{\mathbf{S}, \mathbf{T}\}$, $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}\}$, start symbol $\mathbf{S}$,
$\mathbf{S} \rightarrow \mathbf{0S}|\mathbf{00T}|\mathbf{01T}$, $\mathbf{T} \rightarrow \mathbf{1T}|\mathbf{1}$.

Grammar for Example 2.4 is context-free. This was the example of all words with same number of $\mathbf{0}$ and $\mathbf{1}$.

Grammar for Example 2.5 is regular. This was the example of all words of odd length.

Example 2.9.
Context-Sensitive Grammar for $\{\mathbf{0^n 1^n 2^n} : \mathbf{n} \in \mathbb{N}\}$.
$\mathbf{N} = \{\mathbf{S}, \mathbf{T}, \mathbf{U}\}$, $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$, start symbol $\mathbf{S}$,
$\mathbf{S} \rightarrow \mathbf{012}|\mathbf{0T12}|\varepsilon$, $\mathbf{T} \rightarrow \mathbf{0T1U}|\mathbf{01U}$, $\mathbf{U1} \rightarrow \mathbf{1U}$, $\mathbf{U2} \rightarrow \mathbf{22}$.

Expression $\Rightarrow$ Regular Grammar

Theorem

Every regular language is generated by a regular grammar.

The next slide will provide the following details:

- Every finite set is generated by a regular grammar;
- If two regular grammars with disjoint sets of non-terminals generate $\mathbf{L}$ and $\mathbf{H}$ then one can combine these two grammars to new regular grammars for $\mathbf{L} \cup \mathbf{H}$, $\mathbf{L} \cdot \mathbf{H}$ and $\mathbf{L}^*$, respectively.

One can always rename the non-terminals in order to achieve that two grammars do not use the same non-terminals; thus one can prove by structural induction that every regular set $\mathbf{L}$ satisfies the property "$\mathbf{L}$ is generated by some regular grammar."

Constructing the Grammars

If $\mathbf{L} = \{\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n}\}$ then grammar $(\{\mathbf{S}\}, \mathbf{\Sigma}, \{\mathbf{S} \rightarrow \mathbf{w_1}|\mathbf{w_2}|\ldots|\mathbf{w_n}\}, \mathbf{S})$ generates $\mathbf{L}$.

Assume that the regular grammar $(\mathbf{N_1}, \mathbf{\Sigma}, \mathbf{P_1}, \mathbf{S_1})$ generates $\mathbf{L}$ and the regular grammar $(\mathbf{N_2}, \mathbf{\Sigma}, \mathbf{P_2}, \mathbf{S_2})$ generates $\mathbf{H}$ and $\mathbf{N_1} \cap \mathbf{N_2} = \emptyset$. $\mathbf{A}, \mathbf{B}$ are always non-terminals and $\mathbf{w} \in \mathbf{\Sigma}^*$. Let $\mathbf{S} \notin \mathbf{N_1} \cup \mathbf{N_2} \cup \mathbf{\Sigma}$ be a new symbol.

The regular grammar $(\{\mathbf{S}\} \cup \mathbf{N_1} \cup \mathbf{N_2}, \mathbf{\Sigma}, \{\mathbf{S} \rightarrow \mathbf{S_1}|\mathbf{S_2}\} \cup \mathbf{P_1} \cup \mathbf{P_2}, \mathbf{S})$ generates $\mathbf{L} \cup \mathbf{H}$.

Let $\mathbf{P} = \{\mathbf{A} \rightarrow \mathbf{wB} : \mathbf{A} \rightarrow \mathbf{wB} \textbf{ is in } \mathbf{P_1} \cup \mathbf{P_2}\} \cup \{\mathbf{A} \rightarrow \mathbf{wS_2} : \mathbf{A} \rightarrow \mathbf{w} \textbf{ is in } \mathbf{P_1}\} \cup \{\mathbf{A} \rightarrow \mathbf{w} : \mathbf{A} \rightarrow \mathbf{w} \textbf{ is in } \mathbf{P_2}\}$; the regular grammar $(\mathbf{N_1} \cup \mathbf{N_2}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S_1})$ generates $\mathbf{L} \cdot \mathbf{H}$.

Let $\mathbf{P} = \mathbf{P_1} \cup \{\mathbf{S} \rightarrow \mathbf{S_1}|\varepsilon\} \cup \{\mathbf{A} \rightarrow \mathbf{wS} : \mathbf{A} \rightarrow \mathbf{w} \textbf{ is in } \mathbf{P_1}\}$; the regular grammar $(\mathbf{N_1} \cup \{\mathbf{S}\}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ generates $\mathbf{L}^*$.

Example 2.11

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:
$S \to T|V|W$,
$T \to 0T|1T|2U$,
$U \to 0U|1U|2$,
$V \to 0V|2V|\varepsilon$,
$W \to 1W|2W|\varepsilon$.

Regular Grammar ⇒ Expression

Let $\mathbf{R_1, R_2, \ldots, R_n}$ be an explicit list of rules in the regular grammar and define inductively for $\mathbf{m = 0, 1, \ldots, n}$ and all non-terminals $\mathbf{C, D}$ the sets $\mathbf{E_{C,D,m}, E_{C,m}}$ defined as follows:

- $\mathbf{E_{C,D,m}}$ is the set of all words $\mathbf{v \in \Sigma^*}$ for which there is a derivation $\mathbf{C \Rightarrow^* vD}$ using only the rules $\mathbf{R_1, R_2, \ldots, R_m}$;
- $\mathbf{E_{C,m}}$ is the set of all words $\mathbf{v \in \Sigma^*}$ for which there is a derivation $\mathbf{C \Rightarrow^* v}$ using only the rules $\mathbf{R_1, R_2, \ldots, R_m}$.

It will be proven by induction that all these sets can be generated by regular expressions. The base case is that $\mathbf{E_{C,C,0} = \{\varepsilon\}}$, as one can derive $\mathbf{C \Rightarrow^* C}$ without applying any rule, that $\mathbf{E_{C,D,0} = \emptyset}$ when $\mathbf{C \neq D}$ and that $\mathbf{E_{C,0} = \emptyset}$.

Inductive Step

For $\mathbf{m = 0, 1, \ldots, n-1}$, define all $\mathbf{E_{C,D,m+1}}$ and $\mathbf{E_{C,m+1}}$ using the corresponding expressions on level $\mathbf{m}$.

If $\mathbf{R_{m+1}}$ is $\mathbf{A \to w}$:

- $\mathbf{E_{C,D,m+1} = E_{C,D,m}}$;
- $\mathbf{E_{C,m+1} = E_{C,m} \cup (E_{C,A,m} \cdot \{w\})}$.

If $\mathbf{R_{m+1}}$ is $\mathbf{A \to wB}$:

- $\mathbf{E_{C,D,m+1} =}$
  $\mathbf{E_{C,D,m} \cup (E_{C,A,m} \cdot \{w\} \cdot (E_{B,A,m} \cdot \{w\})^* \cdot E_{B,D,m})}$;
- $\mathbf{E_{C,m+1} = E_{C,m} \cup (E_{C,A,m} \cdot \{w\} \cdot (E_{B,A,m} \cdot \{w\})^* \cdot E_{B,m})}$.

The final regular expression is $\mathbf{E_{S,n}}$ where $\mathbf{S}$ is the start symbol.

Example 2.13

Regular grammar $(\{\mathbf{S},\mathbf{T}\},\{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{3}\},\mathbf{P},\mathbf{S})$ with $\mathbf{S} \to \mathbf{0S}|\mathbf{1T}|\mathbf{2}$ and $\mathbf{T} \to \mathbf{0T}|\mathbf{1S}|\mathbf{3}$.
Let $\mathbf{L_S} = \{\mathbf{w} : (\mathbf{S} \to \mathbf{w}) \in \mathbf{P}\} = \{\mathbf{2}\}$ and $\mathbf{L_T} = \{\mathbf{3}\}$.
Let $\mathbf{L_{S,S}} = \{\mathbf{w} : (\mathbf{S} \to \mathbf{wS}) \in \mathbf{P}\} = \{\mathbf{0}\}$, $\mathbf{L_{S,T}} = \{\mathbf{1}\}$, $\mathbf{L_{T,S}} = \{\mathbf{1}\}$, $\mathbf{L_{T,T}} = \{\mathbf{0}\}$.

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

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

Exercises

Construct regular grammars for the below regular expressions or vice versa.

Exercise 2.6. Make a regular grammar for $\mathbf{0^*10^*10^*10^*10^*10^*20^*}$.

Exercise 2.12. Make a regular grammar for $(\{\mathbf{00},\mathbf{11},\mathbf{22}\}\cdot\{\mathbf{33}\}^*)^*$.

Exercise 2.14. Let $(\{\mathbf{S},\mathbf{T},\mathbf{U}\},\{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4}\},\mathbf{P},\mathbf{S})$ be a grammar where the set $\mathbf{P}$ contains the rules $\mathbf{S}\rightarrow\mathbf{0S}|\mathbf{1T}|\mathbf{2}$, $\mathbf{T}\rightarrow\mathbf{0T}|\mathbf{1U}|\mathbf{3}$ and $\mathbf{U}\rightarrow\mathbf{0U}|\mathbf{1S}|\mathbf{4}$. Make a regular expression describing this language.

The Pumping Lemma

Theorem 2.15 (a)
Let $\mathbf{L}$ be a regular language. There is a constant $\mathbf{k}$ such that every $\mathbf{w} \in \mathbf{L}$ with $|\mathbf{w}| > \mathbf{k}$ equals to $\mathbf{xyz}$ with $\mathbf{y} \neq \varepsilon$ and $|\mathbf{xy}| \leq \mathbf{k}$ and $\mathbf{xy^*z} \subseteq \mathbf{L}$.

Tighter versions will be shown later.

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

Proof by Structural Induction

If $\mathbf{k}$ is larger than the length of all members of $\mathbf{L}$ then $\mathbf{L}$ satisfies the Pumping Lemma with constant $\mathbf{k}$.

If $\mathbf{L}, \mathbf{H}$ satisfy the Pumping Lemma with constant $\mathbf{k}$ so does $\mathbf{L} \cup \mathbf{H}$.

If $\mathbf{L}, \mathbf{H}$ satisfy the Pumping Lemma with constant $\mathbf{k}$ then $\mathbf{L} \cdot \mathbf{H}$ satisfies the Pumping Lemma with constant $\mathbf{2k}$: if $\mathbf{v} \in \mathbf{L}$ and $\mathbf{w} \in \mathbf{H}$ satisfy $|\mathbf{vw}| > \mathbf{2k}$ then either $|\mathbf{v}| \leq \mathbf{k}$ and one can pump inside the first $\mathbf{k}$ symbols of $\mathbf{w}$ or $|\mathbf{v}| > \mathbf{k}$ and one can pump inside the first $\mathbf{k}$ symbols of $\mathbf{v}$.

If $\mathbf{L}$ satisfies the Pumping Lemma with constant $\mathbf{k}$ so does $\mathbf{L}^*$: If $\mathbf{v} = \mathbf{w_1 w_2 \ldots w_n}$ with $|\mathbf{v}| > \mathbf{k}$ and $\mathbf{w_1, w_2, \ldots, w_n} \in \mathbf{L}$ then either $\mathbf{1} \leq |\mathbf{w_1}| \leq \mathbf{k}$ and $\mathbf{w_1^* w_2 \ldots w_n} \subseteq \mathbf{L}$ or $|\mathbf{w_1}| > \mathbf{k}$ and one can pump inside $\mathbf{w_1}$. Here wlog $\mathbf{w_1} \neq \varepsilon$.

Weaker Version of Pumping Lemma

Corollary 2.16

If $\mathbf{L}$ is regular then there is a constant $\mathbf{k}$ such that for all $\mathbf{u} \in \mathbf{L}$ longer than $\mathbf{k}$ there are $\mathbf{x}, \mathbf{y}, \mathbf{z}$ with $\mathbf{y} \neq \varepsilon$, $\mathbf{u} = \mathbf{xyz}$ and $\mathbf{xy^*z} \subseteq \mathbf{L}$.

Exercise 2.17

Let $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, \ldots$ be the list of prime numbers in ascending order. Show that $\mathbf{L} = \{\mathbf{0^n} : \mathbf{n} > \mathbf{0}$ and $\mathbf{n} \neq \mathbf{p_1} \cdot \mathbf{p_2} \cdot \ldots \cdot \mathbf{p_m}$ for all $\mathbf{m}\}$ satisfies Corollary 2.16 but does not satisfy Theorem 2.15 (a).

Exercise 2.18

Assume that $(\mathbf{N}, \boldsymbol{\Sigma}, \mathbf{P}, \mathbf{S})$ is a regular grammar and $\mathbf{h}$ is a constant such that $\mathbf{N}$ has less than $\mathbf{h}$ elements and for all rules of the form $\mathbf{A} \rightarrow \mathbf{wB}$ or $\mathbf{A} \rightarrow \mathbf{w}$ with $\mathbf{A}, \mathbf{B} \in \mathbf{N}$ and $\mathbf{w} \in \boldsymbol{\Sigma}^*$ it holds that $|\mathbf{w}| < \mathbf{h}$. Show that Theorem 2.15 (a) holds with the constant $\mathbf{k}$ being $\mathbf{h^2}$.

Pumping Position and Length

Example 2.20

Let $\mathbf{L = \{w \in \{0,1\}^* : w}$ has as many $\mathbf{0}$s as $\mathbf{1}$s$\}$.

$\mathbf{L}$ satisfies Corollary 2.16.

Given $\mathbf{w \in \{0,1\}^* - \{0\}^* - \{1\}^*}$. Let $\mathbf{w = xyz}$ with $\mathbf{y \in \{01, 10\}}$.

If $\mathbf{w \in L}$ then $\mathbf{xy^*z \subseteq L}$.

$\mathbf{L}$ does not satisfy Theorem 2.15 (a), as there is a constraint on the position where the word is pumped.

Let $\mathbf{k}$ be the pumping constant and consider $\mathbf{0^{k+1}1^{k+1}}$. Pumping before position $\mathbf{k}$ expands or reduces the number of $\mathbf{0}$s while keeping the number of $\mathbf{1}$s the same.

Context-Free Languages

Theorem 2.15 (b): Pumping-Lemma for CTF
Assume that $\mathbf{L}$ is a context-free language. Then there is a constant $\mathbf{k}$ such that for all $\mathbf{u} \in \mathbf{L}$ with $|\mathbf{u}| > \mathbf{k}$ there is a representation $\mathbf{vwxyz}$ of $\mathbf{u}$ with $|\mathbf{wxy}| \leq \mathbf{k}$ and $\mathbf{w} \neq \varepsilon$ or $\mathbf{y} \neq \varepsilon$ and $\mathbf{vw^nxy^nz} \in \mathbf{L}$ for all $\mathbf{n} \in \mathbb{N}$.

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

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

$\mathbf{L} = \{\mathbf{3^n7^n} : \mathbf{n} \in \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots\}\}$.
This language is context-free but not regular.

Primes

Example 2.19

The set $\mathbf{L} = \{\mathbf{0^p} : \mathbf{p}$ is a prime$\}$ is not context-free.

Let $\mathbf{k}$ be the pumping constant and $\mathbf{p}$ be a prime number larger than $\mathbf{k}$.

Now $\mathbf{0^p} = \mathbf{vwxyz}$ with $\mathbf{wy} \neq \varepsilon$ and $\mathbf{vw^rxy^rz} \in \mathbf{L}$ for all $\mathbf{r}$.

Let $\mathbf{q} = |\mathbf{wy}|$, note that $\mathbf{q} > \mathbf{0}$.

Now $\mathbf{vw^{p+1}xy^{p+1}z} \in \mathbf{L}$ and has length $\mathbf{p} + \mathbf{p} \cdot \mathbf{q}$.

This is $\mathbf{p} \cdot (\mathbf{1} + \mathbf{q})$ and is not a prime.

Hence $\mathbf{0^{p+p \cdot q}} \notin \mathbf{L}$, a contradiction to the Pumping Lemma.

So $\mathbf{L}$ does not satisfy the Pumping Lemma for context-free languages.

Theorem 2.21

Let $\mathbf{L} \subseteq \{\mathbf{0}\}^*$. The following conditions are equivalent for $\mathbf{L}$.

(a) $\mathbf{L}$ is regular;

(b) $\mathbf{L}$ is context-free;

(c) $\mathbf{L}$ satisfies the Pumping Lemma for regular languages;

(d) $\mathbf{L}$ satsifies the Pumping Lemma for context-free languages.

Clearly (a) implies (b),(c) and (b),(c) both imply (d).

Proof of (d) implies (a)

Assume that $\mathbf{k}$ is the pumping constant for the context-free Pumping Lemma. Then, for every word $\mathbf{u} \in \mathbf{L}$ with $|\mathbf{u}| > \mathbf{k}$, one can split $\mathbf{0^n}$ into $\mathbf{vwxyz}$ such that $|\mathbf{wxy}| \leq \mathbf{k}$ and $\mathbf{wy} \neq \varepsilon$ and $\mathbf{vw^hxy^hz} \in \mathbf{L}$ for all $\mathbf{h}$.

This in particular holds when $\mathbf{h-1}$ is a multiple of $\mathbf{k!/|wy|}$. As words from $\mathbf{0^*}$ commute with each other, $\mathbf{0^n} \cdot (\mathbf{0^{k!}})^* \subseteq \mathbf{L}$.

For each remainder $\mathbf{m} \in \{\mathbf{0}, \mathbf{1}, \ldots, \mathbf{k!-1}\}$, let

$$\mathbf{n_m} = \min\{\mathbf{i} : \exists \mathbf{j}\, [\mathbf{i} > \mathbf{k} \text{ and } \mathbf{i} = \mathbf{m} + \mathbf{jk!} \text{ and } \mathbf{0^i} \in \mathbf{L}]\}$$

and let $\mathbf{n_m} = \infty$ when there is no such $\mathbf{i}$, that is, $\min \emptyset = \infty$.

Now $\mathbf{L}$ is the union of $\mathbf{L} \cap \{\varepsilon, \mathbf{0}, \mathbf{00}, \ldots, \mathbf{0^k}\}$ and those sets $\mathbf{0^{n_m}} \cdot (\mathbf{0^{k!}})^*$ where $\mathbf{m} < \mathbf{k!}$ and $\mathbf{n_m} < \infty$. Thus $\mathbf{L}$ is regular.

Exercise 2.22

Construct context-free grammars for the sets
$\mathbf{L} = \{0^n1^m2^k : n < m \text{ or } m < k\}$,
$\mathbf{H} = \{0^n1^m2^{n+m} : n, m \in \mathbb{N}\}$ and
$\mathbf{K} = \{w \in \{0, 1, 2\}^* : w$ has a subword of the form 20^n1^n2 for some $n > 0$ or $w = \varepsilon\}$.

Which of the versions of the Pumping Lemma (Theorems 2.15 (a) and 2.15 (b) and Corollary 2.16) are satisfied by $\mathbf{L}$, $\mathbf{H}$ and $\mathbf{K}$, respectively.

Exercise 2.23

Exercise

Let $\mathbf{L} = \{\mathbf{0^n 1^n 2^n} : \mathbf{n} \in \mathbb{N}\}$.

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.

Additional Exercises

Exercise 2.24

Let $\mathbf{L = \{0^h 1^i 2^j 3^k : (h \neq i}$ and $\mathbf{j \neq k)}$ or $\mathbf{(h \neq k}$ and $\mathbf{i \neq j)\}}$ be given. Construct a context-free grammar for $\mathbf{L}$ and determine which of versions of the Pumping Lemma (Corollary 2.16 and Theorems 2.15 (a) and 2.15 (b)) are satisfied by $\mathbf{L}$.

Exercise 2.25

Consider the linear grammar $\mathbf{(\{S\}, \Sigma, \{S \to 00S|S1|S2|3\}, S)}$ and construct for the language $\mathbf{L}$ generated by the grammar the following: a regular grammar for $\mathbf{L}$ and a regular expression for $\mathbf{L}$.

Grammars and Growth

For the following exercises, let $\mathbf{f(n)}$ be the number of words $\mathbf{w} \in \mathbf{L}$ with $|\mathbf{w}| < \mathbf{n}$. To answer the questions, either construct a grammar witnessing that such an $\mathbf{L}$ exists or prove that it cannot exist.

Exercise 2.26
Is there a context-free language $\mathbf{L}$ with $\mathbf{f(n) = \lfloor \sqrt{n} \rfloor}$?

Exercise 2.27
Is there a regular $\mathbf{L}$ with $\mathbf{f(n) = n(n+1)/2}$?

Exercise 2.28
Is there a context-sensitive $\mathbf{L}$ with $\mathbf{f(n) = n^n}$, where $\mathbf{0^0 = 0}$?

Exercise 2.29
Is there a regular $\mathbf{L}$ with $\mathbf{f(n) = (3^n - 1)/2 + \lfloor n/2 \rfloor}$?

Exercise 2.30
Is there a regular $\mathbf{L}$ with $\mathbf{f(n) = \lfloor n/3 \rfloor + \lfloor n/2 \rfloor}$?

Length of Pumps

Call a nonempty $\mathbf{y}$ a pump for $\mathbf{xyz} \in \mathbf{L}$ iff $\{\mathbf{x}\} \cdot \{\mathbf{y}\}^* \cdot \{\mathbf{z}\} \subseteq \mathbf{L}$. For the below languages $\mathbf{L}$, find the smallest $\mathbf{k}$ such there is a $\mathbf{h}$ for which all words $\mathbf{w} \in \mathbf{L}$ with $|\mathbf{w}| \geq \mathbf{h}$ have a pump of length up to $\mathbf{k}$. State $\mathbf{h}, \mathbf{k}$ and explain the answer.

Exercise 2.31: $\mathbf{L} = \{\mathbf{000}, \mathbf{111}, \mathbf{222}\}^* \cap \{\mathbf{0000}, \mathbf{1111}, \mathbf{2222}\}^* \cap \{\mathbf{00000}, \mathbf{11111}, \mathbf{22222}\}^*$.

Exercise 2.32: $\mathbf{L}$ is the set of all words where the length is $\mathbf{1}$ or $\mathbf{3}$ or $\mathbf{7}$ or $\mathbf{9}$ modulo $\mathbf{10}$.

Exercise 2.33: $\mathbf{L}$ is the set of all words where the length is neither $\mathbf{1}$ nor $\mathbf{3}$ nor $\mathbf{7}$ nor $\mathbf{9}$ modulo $\mathbf{10}$.

Exercise 2.34: $\mathbf{L} = \{\mathbf{001100110011}\} \cdot \{\mathbf{222}\}^* \cup \{\mathbf{0011}\} \cdot \{\mathbf{2222}\}^* \cup \{\mathbf{001100110011001100110011}\}$.

Exercise 2.35: $\mathbf{L}$ is the set of all decimal numbers without leading zeroes which are multiples of $\mathbf{512}$.

Decimal Numbers

For the following sets $\mathbf{H}$, find an infinite $\mathbf{L} \subseteq \mathbf{H}$ such that $\mathbf{L}$ is of complexity as low as possible and $\mathbf{L}$ satisfies the given property. Choose $\mathbf{L}$ to be in the best level of the Chomsky Hierarchy (regular better than context-free better than context-sensitive). Give grammars or regular expressions and use pumping lemmas to show that it does not go better.

Exercise 2.36: Find infinite $\mathbf{L} \subseteq \mathbf{H}$ for $\mathbf{H = \{10^n 20^m 1 : n \geq m \geq 1 \text{ and } n + m \text{ is even}\}}$ such that all members of $\mathbf{L}$ are square numbers.

Exercise 2.37: Find infinite $\mathbf{L} \subseteq \mathbf{H}$ for $\mathbf{H = \{10^n 30^m 30^k 1 : 2n \geq m + k \text{ and } 3 \text{ divides } n + m + k\}}$ such that all members of $\mathbf{L}$ are third powers (cubes).

Exercise 2.38: Find infinite $\mathbf{L} \subseteq \mathbf{H}$ for $\mathbf{H = \{1\} \cdot \{0\}^+ \cdot \{3\} \cdot \{0\}^+ \cdot \{3\} \cdot \{0\}^+ \cdot \{1\} \cdot \{0\}^+}$ such that all members of $\mathbf{L}$ are third powers (cubes).