Advanced Automata Theory 3
Combining Languages

Frank Stephan

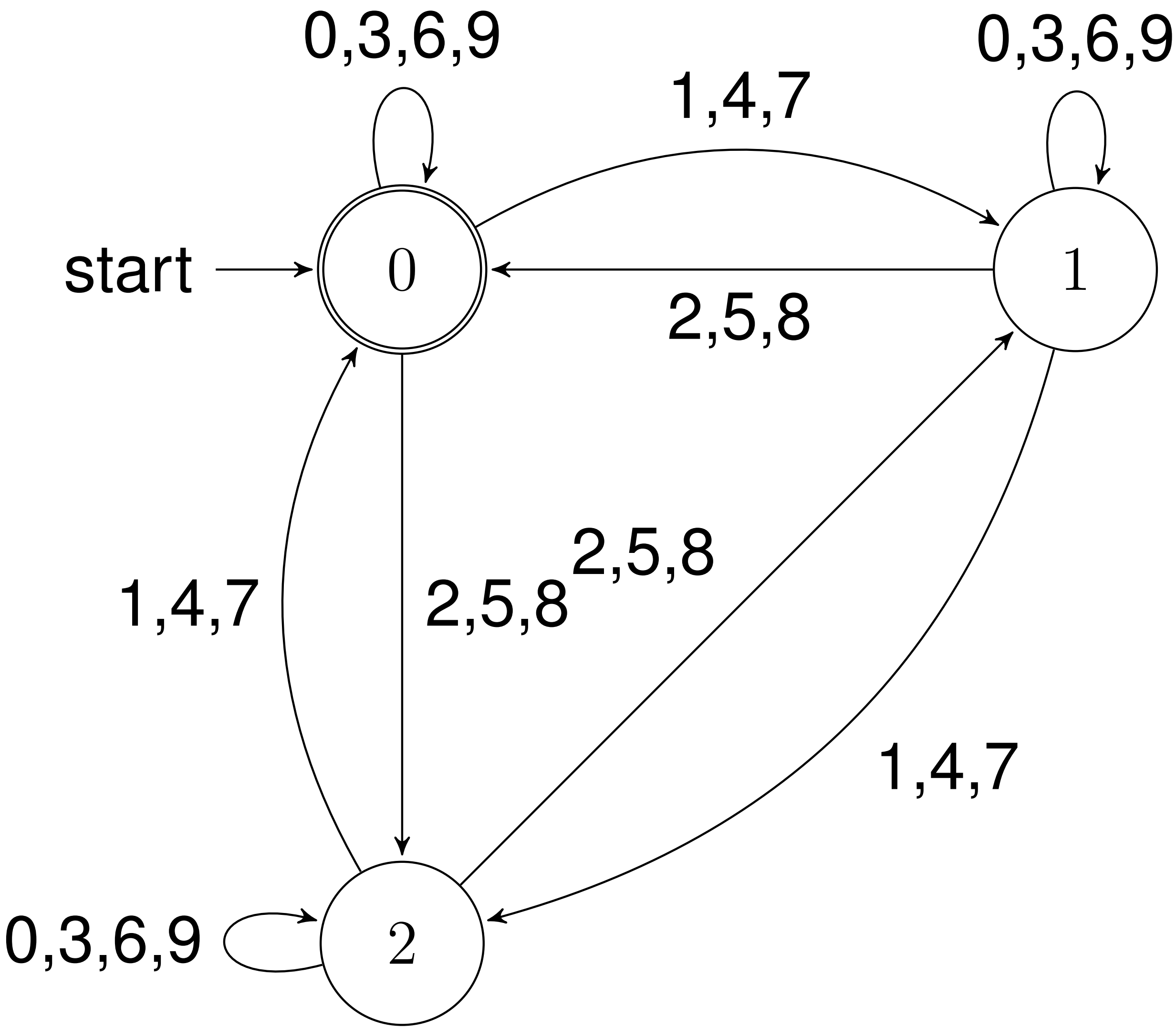
Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition 1



Repetition 2

Theorem
Let $\mathbf{L}$ be any language (subset of $\mathbf{\Sigma}^*$).
$\mathbf{L}$ is generated by a regular grammar $\Leftrightarrow$
$\mathbf{L}$ is generated by a regular expression $\Leftrightarrow$
$\mathbf{L}$ is recognised by a dfa $\Leftrightarrow$
$\mathbf{L}$ is recognised by an nfa $\Rightarrow$
$\mathbf{L}$ satisfies the Block Pumping Lemma $\Rightarrow$
$\mathbf{L}$ satisfies the Pumping Lemma with bound $\Rightarrow$
$\mathbf{L}$ satisfies the Pumping Lemma without bound.

The last three $\Rightarrow$ cannot be inverted.

$\{\mathbf{w} : \mathbf{w}$ does not start with $\mathbf{010}$ or $\mathbf{w}$ has length $\mathbf{n^2}$ for some $\mathbf{n}\}$ satisfies the Pumping Lemma with bound but not the Block Pumping Lemma.

$\{\mathbf{w} : |\mathbf{w}|$ is not a power of $\mathbf{2}\}$ satisfies the Pumping Lemma without bound but not the one with bound.

Repetition 3

If $\mathbf{L}$ is a regular set then there is a constant $\mathbf{k}$ such that for all strings $\mathbf{u_0, u_1, \ldots, u_k}$ with $\mathbf{u_1, u_2, \ldots, u_{k-1}}$ not empty and $\mathbf{u_0 u_1 \ldots u_k \in L}$ there are $\mathbf{i, j}$ with $\mathbf{0 < i \leq j < k}$ and

$$\mathbf{(u_0 u_1 \ldots u_{i-1}) \cdot (u_i u_{i+1} \ldots u_j)^* \cdot (u_{j+1} u_{j+2} \ldots u_k) \subseteq L.}$$

So if one splits a word in $\mathbf{L}$ into $\mathbf{k+1}$ parts then one can select some parts in the middle of the word which can be pumped.

Example: $\mathbf{\{1,2\}^* \cdot \{0\} \cdot \{1,2\}^* \cdot \{0\} \cdot \{1,2\}^*}$ satisfies the Block Pumping Lemma with $\mathbf{k = 4}$; splitting a word in this language into $\mathbf{u_0 u_1 u_2 u_3 u_4}$, either $\mathbf{u_1}$ or $\mathbf{u_2}$ or $\mathbf{u_3}$ does not contain $\mathbf{0}$ and can be pumped.

Repetition 4

Any nfa with $\mathbf{n}$ states can be replaced by a complete dfa with $\mathbf{2^n}$ states. Alternatively one can use an incomplete dfa, which might reject input due to δ being undefined on some pair $(\mathbf{q}, \mathbf{a})$; such a dfa can be made using $\mathbf{2^n - 1}$ states.

The bound $\mathbf{2^n}$ for the size of the dfa is tight (except for the case that the alphabet is unary, say $\Sigma = \{\mathbf{0}\}$).

Product Automata

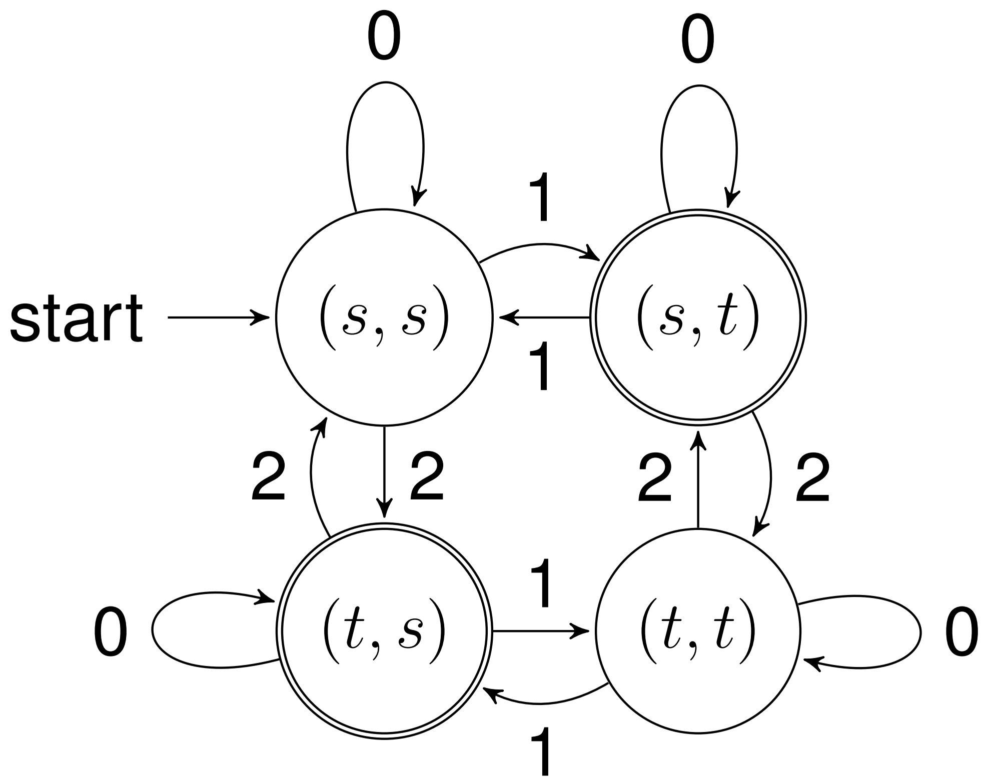
Let $(\mathbf{Q_1}, \mathbf{\Sigma}, \delta_1, \mathbf{s_1}, \mathbf{F_1})$ and $(\mathbf{Q_2}, \mathbf{\Sigma}, \delta_2, \mathbf{s_2}, \mathbf{F_2})$ be dfas which recognise $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively.

Consider $(\mathbf{Q_1} \times \mathbf{Q_2}, \mathbf{\Sigma}, \delta_1 \times \delta_2, (\mathbf{s_1}, \mathbf{s_2}), \mathbf{F})$ with $(\delta_1 \times \delta_2)((\mathbf{q_1}, \mathbf{q_2}), \mathbf{a}) = (\delta_1(\mathbf{q_1}, \mathbf{a}), \delta_2(\mathbf{q_2}, \mathbf{a}))$. This automaton is called a product automaton and one can choose $\mathbf{F}$ such that it recognises the union or intersection or difference of the respective languages.

Union: $\mathbf{F} = \mathbf{F_1} \times \mathbf{Q_2} \cup \mathbf{Q_1} \times \mathbf{F_2}$;
Intersection: $\mathbf{F} = \mathbf{F_1} \times \mathbf{F_2} = \mathbf{F_1} \times \mathbf{Q_2} \cap \mathbf{Q_1} \times \mathbf{F_2}$;
Difference: $\mathbf{F} = \mathbf{F_1} \times (\mathbf{Q_2} - \mathbf{F_2})$;
Symmetric Difference: $\mathbf{F} = \mathbf{F_1} \times (\mathbf{Q_2} - \mathbf{F_2}) \cup (\mathbf{Q_1} - \mathbf{F_1}) \times \mathbf{F_2}$.

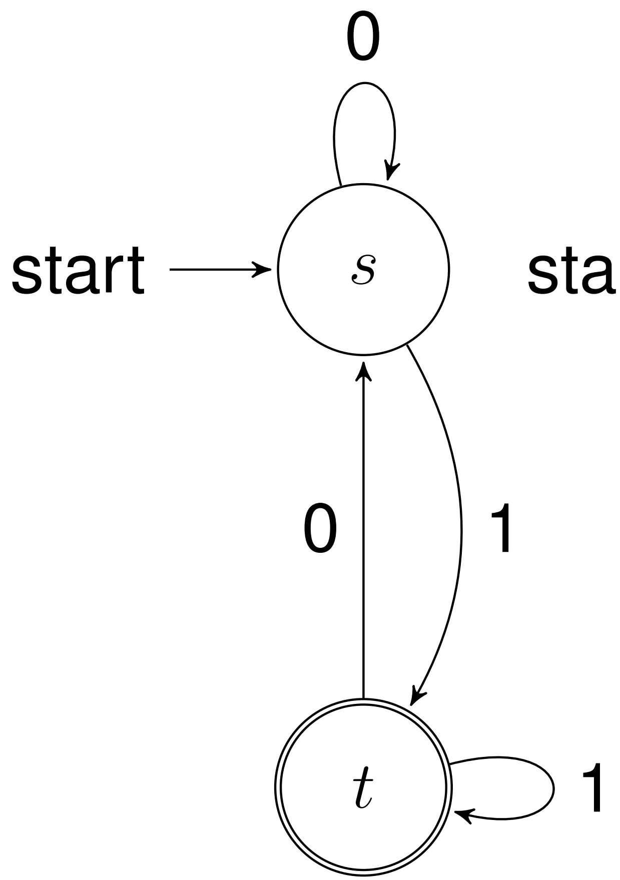
Example

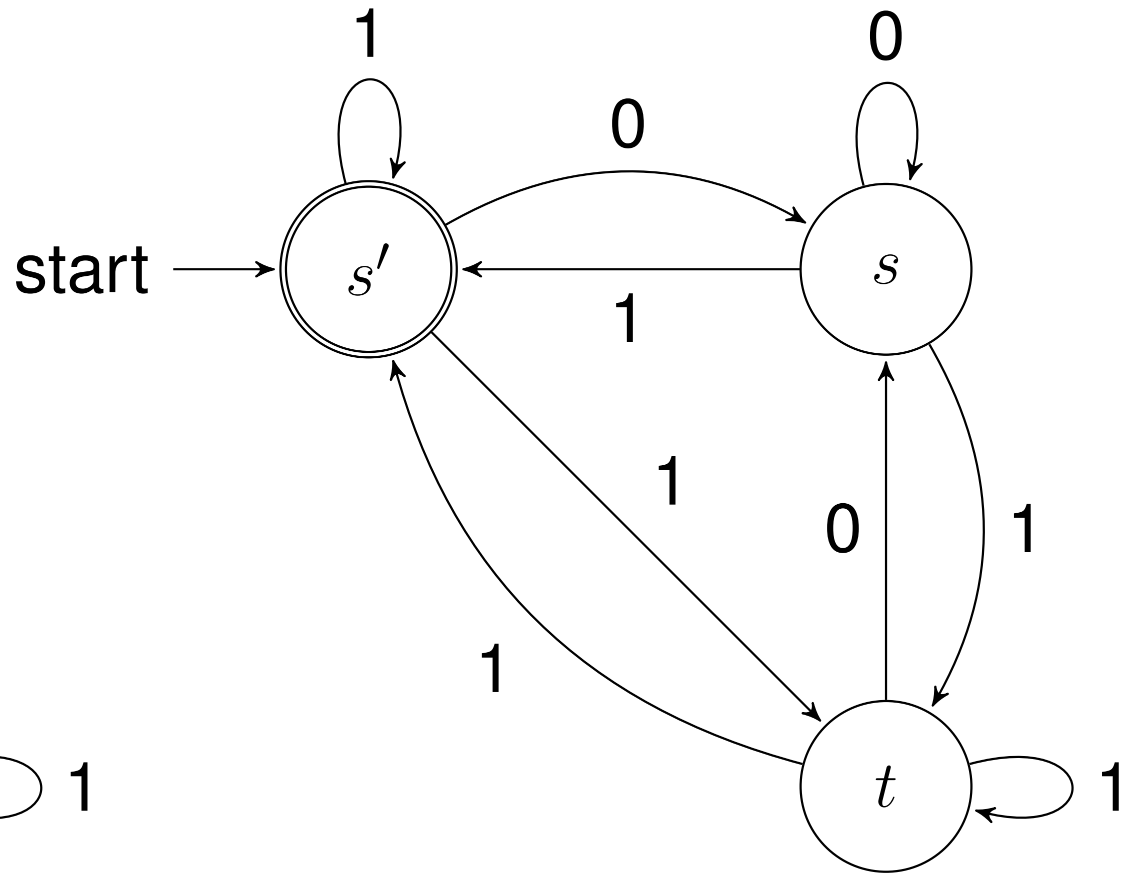
Let the first automaton recognise the language of words in $\{0, 1, 2\}$ with an even number of 1s and the second automaton with an even number of 2s. Both automata have the accepting and starting state s and a rejection state t; they change between s and t whenever they see 1 or 2, respectively. Example of a product automaton.



Kleene Star

Assume $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{F})$ is an nfa recognising $\mathbf{L}$. Now $\mathbf{L}^*$ is recognised by $(\mathbf{Q} \cup \{\mathbf{s}'\}, \mathbf{\Sigma}, \mathbf{\Delta}, \mathbf{s}', \{\mathbf{s}'\})$ where $\mathbf{\Delta} = \delta \cup \{(\mathbf{s}', \mathbf{a}, \mathbf{p}) : (\mathbf{s}, \mathbf{a}, \mathbf{p}) \in \delta\} \cup \{(\mathbf{p}, \mathbf{a}, \mathbf{s}) : (\mathbf{p}, \mathbf{a}, \mathbf{q}) \in \delta$ for some $\mathbf{q} \in \mathbf{F}\} \cup \{(\mathbf{s}', \mathbf{a}, \mathbf{s}') : \mathbf{a} \in \mathbf{L}\}$.





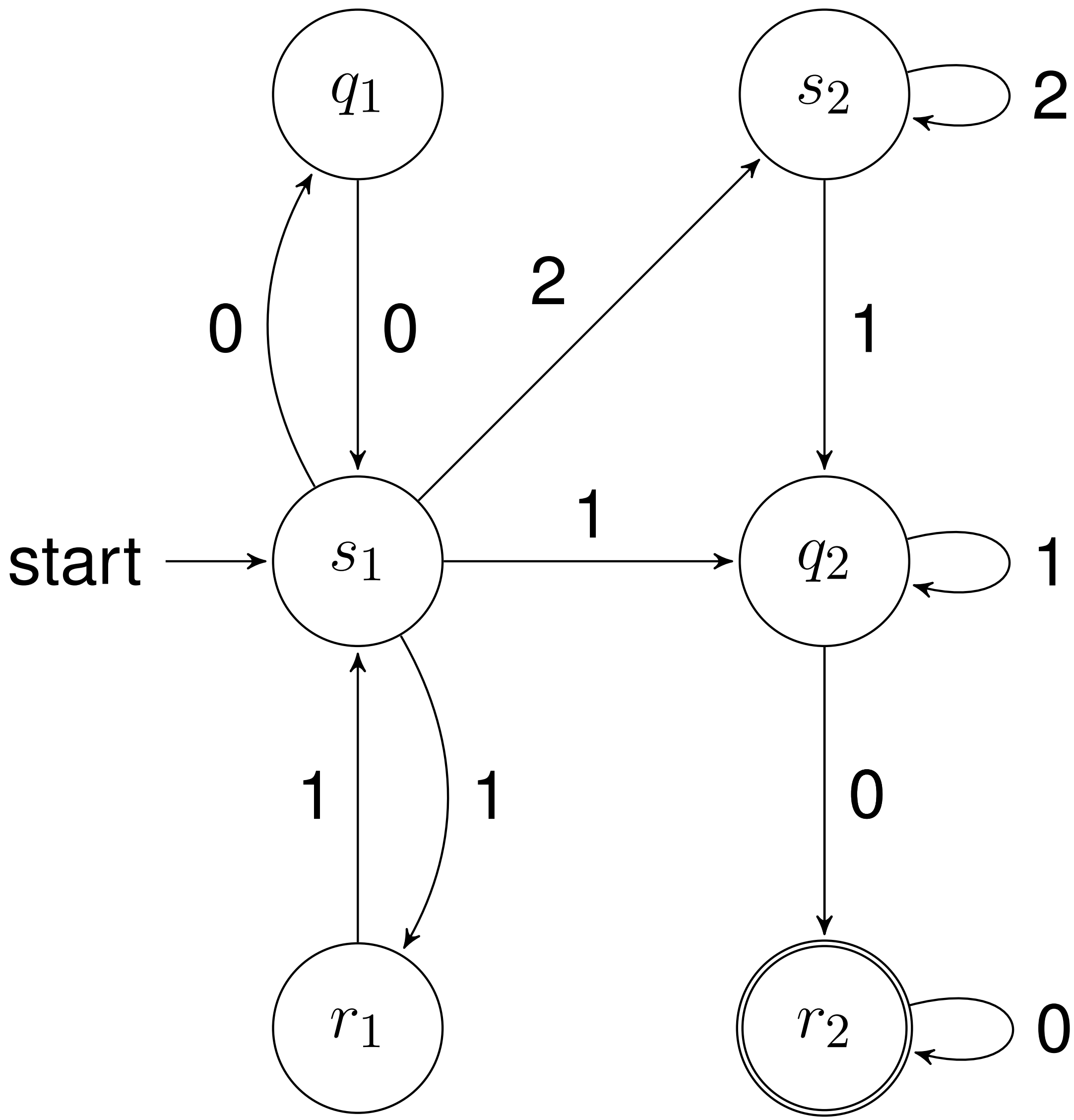
Concatenation

Assume $(\mathbf{Q_1}, \mathbf{\Sigma}, \delta_\mathbf{1}, \mathbf{s_1}, \mathbf{F_1})$ and $(\mathbf{Q_2}, \mathbf{\Sigma}, \delta_\mathbf{2}, \mathbf{s_2}, \mathbf{F_2})$ are nfas recognising $\mathbf{L_1}$ and $\mathbf{L_2}$ with $\mathbf{Q_1} \cap \mathbf{Q_2} = \emptyset$ and assume $\varepsilon \notin \mathbf{L_2}$. Now $(\mathbf{Q_1} \cup \mathbf{Q_2}, \mathbf{\Sigma}, \delta, \mathbf{s_1}, \mathbf{F_2})$ recognises $\mathbf{L_1} \cdot \mathbf{L_2}$ where $(\mathbf{p}, \mathbf{a}, \mathbf{q}) \in \delta$ whenever $(\mathbf{p}, \mathbf{a}, \mathbf{q}) \in \delta_\mathbf{1} \cup \delta_\mathbf{2}$ or ($\mathbf{p} \in \mathbf{F_1}$ and $(\mathbf{s_2}, \mathbf{a}, \mathbf{q}) \in \delta_\mathbf{2}$).

If $\mathbf{L_2}$ contains ε then one can consider the union of $\mathbf{L_1}$ and $\mathbf{L_1} \cdot (\mathbf{L_2} - \{\varepsilon\})$.

Example

$\mathbf{L_1 \cdot L_2}$ with $\mathbf{L_1 = \{00, 11\}^*}$ and $\mathbf{L_2 = 2^*1^+0^+}$.



Exercise 3.3

The previous slides give upper bounds on the size of the dfa for a union, intersection, difference and symmetric difference as n^2 states, provided that the original two dfas have at most n states.

Give the corresponding bounds for nfas: If L and H are recognised by nfas having at most n states each, how many states does one need at most for an nfa recognising (a) the union $L \cup H$, (b) the intersection $L \cap H$, (c) the difference $L - H$ and (d) the symmetric difference $(L - H) \cup (H - L)$?

Give the bounds in terms of "linear", "quadratic" and "exponential". Explain the bounds.

Exercises Combining DFAs and NFAs

Exercise 3.4

Let $\mathbf{\Sigma = \{0,1,2,3,4,5,6,7,8,9\}}$. Construct a (not necessarily complete) dfa recognising the language $\mathbf{\Sigma \cdot \{aa : a \in \Sigma\}^* \cap \{aaaaa : a \in \Sigma\}^*}$. It is not needed to give a full table for the dfa, but a general schema and an explanation how it works.

Exercise 3.5

Make an nfa for the intersection of the following languages: $\mathbf{\{0,1,2\}^* \cdot \{001\} \cdot \{0,1,2\}^* \cdot \{001\} \cdot \{0,1,2\}^*}$; $\mathbf{\{001,0001,2\}^*}$; $\mathbf{\{0,1,2\}^* \cdot \{00120001\} \cdot \{0,1,2\}^*}$.

Exercise 3.6

Make an nfa for the union $\mathbf{L_0 \cup L_1 \cup L_2}$ with $\mathbf{L_a = \{0,1,2\}^* \cdot \{aa\} \cdot \{0,1,2\}^* \cdot \{aa\} \cdot \{0,1,2\}^*}$ for $\mathbf{a \in \{0,1,2\}}$.

Exercise 3.7

Consider two context-free grammars with terminals $\mathbf{\Sigma}$, disjoint non-terminals $\mathbf{N_1}$ and $\mathbf{N_2}$, start symbols $\mathbf{S_1} \in \mathbf{N_1}$ and $\mathbf{S_2} \in \mathbf{N_2}$ and rule sets $\mathbf{P_1}$ and $\mathbf{P_2}$ which generate $\mathbf{L}$ and $\mathbf{H}$, respectively. Explain how to form from these a new context-free grammar for

(a) $\mathbf{L} \cup \mathbf{H}$,

(b) $\mathbf{L} \cdot \mathbf{H}$ and

(c) $\mathbf{L}^*$.

Write down the context-free grammars for $\{\mathbf{0^n 1^{2n}} : \mathbf{n} \in \mathbb{N}\}$ and $\{\mathbf{0^n 1^{3n}} : \mathbf{n} \in \mathbb{N}\}$ and form the grammars for the the union, concatenation and star explicitly.

Example 3.8

The language $\{0\}^* \cdot \{1^n 2^n : n \in \mathbb{N}\}$ is context-free.

Grammar $(\{S, T\}, \{0, 1, 2\}, P, S)$ with P be given by $S \to 0S|T|\varepsilon$ and $T \to 1T2|\varepsilon$.

The language $\{0^n 1^n : n \in \mathbb{N}\} \cdot 2^*$ is context-free.

$L = \{0^n 1^n 2^n : n \in \mathbb{N}\}$ is not context-free but the intersection of the two above.

The complement of L is the union of $\{0^n 1^m 2^k : n < k\}$, $\{0^n 1^m 2^k : n > k\}$, $\{0^n 1^m 2^k : m < k\}$, $\{0^n 1^m 2^k : m > k\}$, $\{0^n 1^m 2^k : n < m\}$, $\{0^n 1^m 2^k : n > m\}$ and $\{0, 1, 2\}^* \cdot \{10, 20, 21\} \cdot \{0, 1, 2\}^*$.

Each of these languages is context-free. Grammar for the first of them: $S \to 0S2|S2|T2, T \to 1T|\varepsilon$. The union is also context-free. Hence L has a context-free complement.

Context-Free Intersects Regular

Theorem 3.9.
If $\mathbf{L}$ is context-free and $\mathbf{H}$ is regular then $\mathbf{L} \cap \mathbf{H}$ is context-free.

Construction.
Let $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ be a context-free grammar generating $\mathbf{L}$ with every rule being either $\mathbf{A} \to \mathbf{w}$ or $\mathbf{A} \to \mathbf{BC}$ with $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbf{N}$ and $\mathbf{w} \in \mathbf{\Sigma}^*$.

Let $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{F})$ be a dfa recognising $\mathbf{H}$.

Let $\mathbf{S}' \notin \mathbf{Q} \times \mathbf{N} \times \mathbf{Q}$ and make the following new grammar $(\mathbf{Q} \times \mathbf{N} \times \mathbf{Q} \cup \{\mathbf{S}'\}, \mathbf{\Sigma}, \mathbf{R}, \mathbf{S}')$ with rules $\mathbf{R}$:
$\mathbf{S}' \to (\mathbf{s}, \mathbf{S}, \mathbf{q})$ for all $\mathbf{q} \in \mathbf{F}$;
$(\mathbf{p}, \mathbf{A}, \mathbf{q}) \to (\mathbf{p}, \mathbf{B}, \mathbf{r})(\mathbf{r}, \mathbf{C}, \mathbf{q})$ for all rules $\mathbf{A} \to \mathbf{BC}$ in $\mathbf{P}$ and all $\mathbf{p}, \mathbf{q}, \mathbf{r} \in \mathbf{Q}$;
$(\mathbf{p}, \mathbf{A}, \mathbf{q}) \to \mathbf{w}$ for all rules $\mathbf{A} \to \mathbf{w}$ in $\mathbf{P}$ with $\delta(\mathbf{p}, \mathbf{w}) = \mathbf{q}$.

Exercises 3.10 and 3.11

Construct context-free grammars for the following intersections between the context-free set $\mathbf{L}$ of all words which contain as many $\mathbf{0}$ as $\mathbf{1}$ and a regular set. Here a grammar for $\mathbf{L}$ is

$$(\{\mathbf{S}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{S} \rightarrow \mathbf{SS}|\varepsilon|\mathbf{0S1}|\mathbf{1S0}\}, \mathbf{S}).$$

Exercise 3.10

Give a context-free grammar for $\mathbf{L} \cap \{\mathbf{00} \cdot \mathbf{1}^+\}^*$;

Exercise 3.11

Give a context-free grammar for $\mathbf{L} \cap \mathbf{0}^*\mathbf{1}^*\mathbf{0}^*\mathbf{1}^*$.

Context-Sensitive and Concatenation

Let $\mathbf{L_1}$ and $\mathbf{L_2}$ be context-sensitive languages not containing ε. Let $(\mathbf{N_1}, \mathbf{\Sigma}, \mathbf{P_1}, \mathbf{S_1})$ and $(\mathbf{N_2}, \mathbf{\Sigma}, \mathbf{P_2}, \mathbf{S_2})$ be two context-sensitive grammars generating $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively, where $\mathbf{N_1} \cap \mathbf{N_2} = \emptyset$ and where each rule $\mathbf{l} \rightarrow \mathbf{r}$ satisfies $|\mathbf{l}| \leq |\mathbf{r}|$ and $\mathbf{l} \in \mathbf{N_e^+}$ for the respective $\mathbf{e} \in \{\mathbf{1}, \mathbf{2}\}$. Let $\mathbf{S} \notin \mathbf{N_1} \cup \mathbf{N_2} \cup \mathbf{\Sigma}$.

Now $(\mathbf{N_1} \cup \mathbf{N_2} \cup \{\mathbf{S}\}, \mathbf{\Sigma}, \mathbf{P_1} \cup \mathbf{P_2} \cup \{\mathbf{S} \rightarrow \mathbf{S_1 S_2}\}, \mathbf{S})$ generates $\mathbf{L_1} \cdot \mathbf{L_2}$.

If $\mathbf{v} \in \mathbf{L_1}$ and $\mathbf{w} \in \mathbf{L_2}$ then $\mathbf{S} \Rightarrow \mathbf{S_1 S_2} \Rightarrow^* \mathbf{v S_2} \Rightarrow^* \mathbf{vw}$. Furthermore, the first rule has to be $\mathbf{S} \Rightarrow \mathbf{S_1 S_2}$ and from then onwards, each rule has on the left side either $\mathbf{l} \in \mathbf{N_1^+}$ so that it applies to the part generated from $\mathbf{S_1}$ or it has in the left side $\mathbf{l} \in \mathbf{N_2^+}$ so that $\mathbf{l}$ is in the part of the word generated from $\mathbf{S_2}$. Hence every intermediate word $\mathbf{z}$ in the derivation is of the form $\mathbf{xy} = \mathbf{z}$ with $\mathbf{S_1} \Rightarrow^* \mathbf{x}$ and $\mathbf{S_2} \Rightarrow^* \mathbf{y}$.

Context-Sensitive and Kleene-star

Let $(\mathbf{N_1}, \mathbf{\Sigma}, \mathbf{P_1}, \mathbf{S_1})$ and $(\mathbf{N_2}, \mathbf{\Sigma}, \mathbf{P_2}, \mathbf{S_2})$ be context-sensitive grammars for $\mathbf{L} - \{\varepsilon\}$ with $\mathbf{N_1} \cap \mathbf{N_2} = \emptyset$ and all rules $\mathbf{l} \rightarrow \mathbf{r}$ satisfying $|\mathbf{l}| \leq |\mathbf{r}|$ and $\mathbf{l} \in \mathbf{N_1^+}$ or $\mathbf{l} \in \mathbf{N_2^+}$, respectively. Let $\mathbf{S}, \mathbf{S'}$ be symbols not in $\mathbf{N_1} \cup \mathbf{N_2} \cup \mathbf{\Sigma}$.

Now consider $(\mathbf{N_1} \cup \mathbf{N_2} \cup \{\mathbf{S}, \mathbf{S'}\}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ where $\mathbf{P}$ contains the rules $\mathbf{S} \rightarrow \mathbf{S'}|\varepsilon$ and $\mathbf{S'} \rightarrow \mathbf{S_1 S_2 S'} \,|\, \mathbf{S_1 S_2} \,|\, \mathbf{S_1}$ plus all rules in $\mathbf{P_1} \cup \mathbf{P_2}$.

This grammar generates $\mathbf{L}^*$.

Exercise 3.14.
Construct a grammar for $\{\mathbf{0^n 1^n 2^n} : \mathbf{n} > \mathbf{0}\}^+$. Try to keep it small (use more intuition than algorithms).

Context-Sensitive and Intersection

Theorem.
The intersection of two context-sensitive languages is context-sensitive.

Construction.
Let $(\mathbf{N_k}, \mathbf{\Sigma}, \mathbf{P_k}, \mathbf{S})$ be grammars for $\mathbf{L_1}$ and $\mathbf{L_2}$. Now make a new non-terminal set $\mathbf{N} = (\mathbf{N_1} \cup \mathbf{\Sigma} \cup \{\#\}) \times (\mathbf{N_2} \cup \mathbf{\Sigma} \cup \{\#\})$ with start symbol $\binom{\mathbf{S}}{\mathbf{S}}$ and following types of rules:
(a) Rules to generate and manage space;
(b) Rules to generate a word $\mathbf{v}$ in the upper row;
(c) Rules to generate a word $\mathbf{w}$ in the lower row;
(d) Rules to convert a string from $\mathbf{N}$ into $\mathbf{v}$ provided that the upper components and lower components of the string are both $\mathbf{v}$.

Type of Rules

(a): $\binom{\mathbf{S}}{\mathbf{S}} \to \binom{\mathbf{S}}{\mathbf{S}}\binom{\#}{\#}$ for producing space; $\binom{\mathbf{A}}{\mathbf{B}}\binom{\#}{\mathbf{C}} \to \binom{\#}{\mathbf{B}}\binom{\mathbf{A}}{\mathbf{C}}$ and $\binom{\mathbf{A}}{\mathbf{C}}\binom{\mathbf{B}}{\#} \to \binom{\mathbf{A}}{\#}\binom{\mathbf{B}}{\mathbf{C}}$ for space management.

(b) and (c): For each rule in $\mathbf{P_1}$, for example, for $\mathbf{AB} \to \mathbf{CDE} \in \mathbf{P_1}$, and all symbols $\mathbf{F}, \mathbf{G}, \mathbf{H}, \ldots$ in $\mathbf{N_2} \cup \mathbf{\Sigma} \cup \{\#\}$, one has the corresponding rule $\binom{\mathbf{A}}{\mathbf{F}}\binom{\mathbf{B}}{\mathbf{G}}\binom{\#}{\mathbf{H}} \to \binom{\mathbf{C}}{\mathbf{F}}\binom{\mathbf{D}}{\mathbf{G}}\binom{\mathbf{E}}{\mathbf{H}}$. So rules in $\mathbf{P_1}$ are simulated in the upper half and rules in $\mathbf{P_2}$ are simulated in the lower half and they use up $\#$ if the left side is shorter than the right one.

(d): Each rule $\binom{\mathbf{a}}{\mathbf{a}} \to \mathbf{a}$ for $\mathbf{a} \in \mathbf{\Sigma}$ is there to convert a matching pair $\binom{\mathbf{a}}{\mathbf{a}}$ from $\mathbf{\Sigma} \times \mathbf{\Sigma}$ (a nonterminal) to $\mathbf{a}$ (a terminal).

Grammar for $0^n 1^n 2^n$ with $n > 0$

Grammar L_1: $S \rightarrow S2|0S1|01$.
Grammar L_2: $S \rightarrow 0S|1S2|12$.

Grammar for Intersection.
A, B, C stand for any members of $\{S, 0, 1, 2, \#\}$.
$N = \{\binom{A}{B} : A, B \in \{S, 0, 1, 2, \#\}\}$.
Rules: $\binom{S}{S} \rightarrow \binom{S}{S}\binom{\#}{\#}$;
$\binom{A}{B}\binom{\#}{C} \rightarrow \binom{\#}{B}\binom{A}{C}$; $\binom{A}{C}\binom{B}{\#} \rightarrow \binom{A}{\#}\binom{B}{C}$;
$\binom{S}{A}\binom{\#}{B} \rightarrow \binom{S}{A}\binom{2}{B}$; $\binom{S}{A}\binom{\#}{B}\binom{\#}{C} \rightarrow \binom{0}{A}\binom{S}{B}\binom{1}{C}$;
$\binom{S}{A}\binom{\#}{B} \rightarrow \binom{0}{A}\binom{1}{B}$;
$\binom{A}{S}\binom{B}{\#} \rightarrow \binom{A}{0}\binom{B}{S}$; $\binom{A}{S}\binom{B}{\#}\binom{C}{\#} \rightarrow \binom{A}{1}\binom{B}{S}\binom{C}{2}$;
$\binom{A}{S}\binom{B}{\#} \rightarrow \binom{A}{1}\binom{B}{2}$;
$\binom{0}{0} \rightarrow 0$; $\binom{1}{1} \rightarrow 1$; $\binom{2}{2} \rightarrow 2$.

Deriving 001122

$$\binom{S}{S} \Rightarrow^* \binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#} \Rightarrow \binom{S}{S}\binom{2}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#} \Rightarrow^*$$
$$\binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#} \Rightarrow \binom{S}{S}\binom{2}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#} \Rightarrow^*$$
$$\binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{S}\binom{S}{\#}\binom{1}{\#}\binom{\#}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$
$$\binom{0}{S}\binom{S}{\#}\binom{\#}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{S}\binom{0}{\#}\binom{1}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$
$$\binom{0}{0}\binom{0}{S}\binom{1}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{0}\binom{0}{0}\binom{1}{S}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$
$$\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{S}\binom{2}{2}\binom{2}{\#} \Rightarrow \binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{S}\binom{2}{\#}\binom{2}{2} \Rightarrow$$
$$\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{1}\binom{2}{2}\binom{2}{2} \Rightarrow^* 001122.$$

Exercise 3.17

Consider the language $\mathbf{L = \{00\} \cdot \{0,1,2,3\}^* \cup \{1,2,3\} \cdot \{0,1,2,3\}^* \cup \{0,1,2,3\}^* \cdot \{02,03,13,10,20,30,21,31,32\} \cdot \{0,1,2,3\}^* \cup \{\varepsilon\} \cup \{01^n2^n3^n : n \in \mathbb{N}\}}$.

Which versions of the Pumping Lemma does it satisfy:

- Regular Pumping Lemma (with / without bounds);
- Context-Free Pumping Lemma (with / without bounds);
- Block Pumping Lemma (for regular languages)?

Determine the exact position of $\mathbf{L}$ in the Chomsky hierarchy.

Mirror Images

Define $(a_1a_2\ldots a_n)^{mi} = a_n\ldots a_2a_1$ as the mirror image of a string. A word w with $w = w^{mi}$ is called a palindrome.

It follows from the definition of context-free and context-sensitive, that if L is context-free / context-sensitive so is L^{mi}. This can be achieved by replacing every rule $l \to r$ by $l^{mi} \to r^{mi}$.

For example, the mirror image of the language of the words 0^n1^{3n+3} is given by language of the words $1^{3n+3}0^n$. While L is generated by a context-free grammar with one non-terminal S and rules $S \to 0S111 \,|\, 111$, L^{mi} is then generated by a similar grammar with the rules $S \to 111S0 \,|\, 111$.

Exercise 3.18

Recall that $\mathbf{x^{mi}}$ is the mirror image of $\mathbf{x}$, so $\mathbf{(01001)^{mi} = 10010}$. Furthermore, $\mathbf{L^{mi} = \{x^{mi} : x \in L\}}$.
Show the following two statements:
(a) If an nfa with $\mathbf{n}$ states recognises $\mathbf{L}$ then there is also an nfa with up to $\mathbf{n+1}$ states recognising $\mathbf{L^{mi}}$.
(b) Find the smallest nfas which recognise $\mathbf{L = 0^*(1^* \cup 2^*)}$ as well as $\mathbf{L^{mi}}$.

Palindromes

The members of the language $\{\mathbf{x} \in \Sigma^* : \mathbf{x} = \mathbf{x}^{\mathbf{mi}}\}$ are called palindromes. A palindrome is a word or phrase which looks the same from both directions.

An example is the German name “OTTO”; furthermore, when ignoring spaces and punctuation marks, a famous palindrome is the phrase “A man, a plan, a canal: Panama.” This palindrome is due to Leigh Mercer (1893-1977).

The grammar with the rules $\mathbf{S} \rightarrow \mathbf{aSa}|\mathbf{aa}|\mathbf{a}|\varepsilon$ with $\mathbf{a}$ ranging over all members of Σ generates all palindromes; so for $\Sigma = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ the rules of the grammar would be $\mathbf{S} \rightarrow \mathbf{0S0} \,|\, \mathbf{1S1} \,|\, \mathbf{2S2} \,|\, \mathbf{00} \,|\, \mathbf{11} \,|\, \mathbf{22} \,|\, \mathbf{0} \,|\, \mathbf{1} \,|\, \mathbf{2} \,|\, \varepsilon$.

The set of palindromes is not regular.

Exercises 3.20-3.22

Exercise 3.20

Let $\mathbf{w} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}\}^*$ be a palindrome of even length and $\mathbf{n}$ be its decimal value. Prove that $\mathbf{n}$ is a multiple of $\mathbf{11}$. Note that it is essential that the length is even, as for odd length there are counter examples (like $\mathbf{111}$ and $\mathbf{202}$).

Exercise 3.21

Given a context-free grammar for a language $\mathbf{L}$, is there also one for $\mathbf{L} \cap \mathbf{L}^{\mathbf{mi}}$? If so, explain how to construct the grammar; if not, provide a counter example where $\mathbf{L}$ is context-free but $\mathbf{L} \cap \mathbf{L}^{\mathbf{mi}}$ is not.

Exercise 3.22

Is the following statement true or false? Prove your answer: Given a language $\mathbf{L}$, the language $\mathbf{L} \cap \mathbf{L}^{\mathbf{mi}}$ equals to $\{\mathbf{w} \in \mathbf{L} : \mathbf{w}$ is a palindrome$\}$.

Pumping Lemmas

Definition 3.24
Let $\mathbf{PUMP_{sw}}$ contain all languages whose large members can be pumped somewhere (satisfy Corollary 1.20).
Let $\mathbf{PUMP_{st}}$ contain all languages whose large members can be pumped near start (satisfy Theorem 1.19 (a)).
Let $\mathbf{PUMP_{bl}}$ contain all languages which satisfy the block pumping lemma (Theorem 2.9).

Proposition 3.25
The classes $\mathbf{PUMP_{sw}}$ and $\mathbf{PUMP_{st}}$ are closed under union, concatenation, Kleene star and Kleene Plus.

For $\mathbf{PUMP_{st}}$ this was proven in Theorem 1.19 (a) and the proof also works with minor modifications for $\mathbf{PUMP_{sw}}$.

Example 3.26

Let $\mathbf{L} = \{\mathbf{0^h 1^k 2^m 3^n : h = 0}$ or $\mathbf{k = m = n}\}$ and $\mathbf{H} = \{\mathbf{00}\} \cdot \{\mathbf{1}\}^* \cdot \{\mathbf{2}\}^* \cdot \{\mathbf{3}\}^*$.

Both languages are in $\mathbf{PUMP_{sw}}$ and $\mathbf{PUMP_{st}}$ and $\mathbf{H}$ is regular. For pumping, one just pumps the first symbol. If it is $\mathbf{0}$ then it can be multiplied or removed; in the case that all $\mathbf{0}$ get removed, one has the regular language $\{\mathbf{1}\}^* \cdot \{\mathbf{2}\}^* \cdot \{\mathbf{3}\}^*$; if the first symbol is in $\{\mathbf{1, 2, 3}\}$ then the word is in $\{\mathbf{1}\}^* \cdot \{\mathbf{2}\}^* \cdot \{\mathbf{3}\}^*$ and pumping the first letter does not change the membership in the language.

The intersection of $\mathbf{L}$ and $\mathbf{H}$ is the language $\{\mathbf{0^2 1^n 2^n 3^n : n} \in \mathbb{N}\}$ which does not satisfy any of the pumping lemma's given in class; in particular $(\mathbf{L} \cap \mathbf{H}) \notin \mathbf{PUMP_{sw}}$ and $(\mathbf{L} \cap \mathbf{H}) \notin \mathbf{PUMP_{st}}$.

More Results

Proposition 3.27

If $\mathbf{L}$ is in $\mathbf{PUMP_{sw}}$ or $\mathbf{PUMP_{bl}}$, then also $\mathbf{L^{mi}}$ is in the respective class.

Example

The language $\{\mathbf{u} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^* : \mathbf{u}$ contains a square$\}$ is in $\mathbf{PUMP_{sw}}$ and $\mathbf{PUMP_{st}}$, but its complement is not.

Exercise 3.28

Show that $\mathbf{PUMP_{bl}}$ is closed under union and concatenation. Furthermore, show that the language $\mathbf{L} = \{\mathbf{v3w4} : \mathbf{v}, \mathbf{w} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^*$ and if $\mathbf{v}, \mathbf{w}$ are both square-free then $|\mathbf{v}| \neq |\mathbf{w}|$ or $\mathbf{v} = \mathbf{w}\}$ is in $\mathbf{PUMP_{bl}}$ while $\mathbf{L}^+$ and $\mathbf{L}^*$ are not.

Theorem 3.29

If $\mathbf{L}, \mathbf{H}$ are in $\mathbf{PUMP_{bl}}$ so is $\mathbf{L} \cap \mathbf{H}$.

Proof of Theorem 3.29

Assume that $\mathbf{L}, \mathbf{H}$ are block pumpable with constant $\mathbf{c}$. Let $\mathbf{c}'$ be so large that if one colours the pairs of a set of $\mathbf{c}'$ elements with two colours then this set has a monochromatic subset with at least $\mathbf{c}$ elements.

Let a word $\mathbf{x} \in \mathbf{L} \cap \mathbf{H}$ with a set $\mathbf{I}$ of $\mathbf{c}'$ breakpoints be given.

If a pair of breakpoints $\mathbf{i}, \mathbf{j} \in \mathbf{I}$ split $\mathbf{x}$ into $\mathbf{u} \cdot \mathbf{v} \cdot \mathbf{w}$ such that $\mathbf{u} \cdot \mathbf{v}^* \cdot \mathbf{w} \subseteq \mathbf{L}$ then let the colour be white else let the colour be red.

There is a monochromatic subset $\mathbf{J}$ of $\mathbf{I}$ containing at least $\mathbf{c}$ breakpoints. By choice of $\mathbf{c}$, a pair of the breakpoints must have white colour and by choice of $\mathbf{J}$, all pairs have. Furthermore, one pair must also split $\mathbf{x}$ into $\mathbf{u} \cdot \mathbf{v} \cdot \mathbf{w}$ with $\mathbf{u} \cdot \mathbf{v}^* \cdot \mathbf{w} \subseteq \mathbf{H}$. Now $\mathbf{u} \cdot \mathbf{v}^* \cdot \mathbf{w} \subseteq \mathbf{L} \cap \mathbf{H}$ and $\mathbf{L} \cap \mathbf{H}$ is in $\mathbf{PUMP_{bl}}$ with constant $\mathbf{c}'$.

Example

Let $\mathbf{L}$ be all words with even number of $\mathbf{1}$ and $\mathbf{H}$ all words with odd number of $\mathbf{2}$. $\mathbf{L}, \mathbf{H}$ are in $\mathbf{PUMP_{bl}}$ with constant $\mathbf{3}$. Now $\mathbf{c}' = \mathbf{6}$.

The word $\mathbf{00(a)01(b)1012(c)2(d)1121(e)00(f)00202}$ has six breakpoints and is in $\mathbf{L} \cap \mathbf{H}$.

A pair of breakpoints is white iff an even number of $\mathbf{1}$ is in between. $(\mathbf{a}, \mathbf{e})$, $(\mathbf{a}, \mathbf{f})$, $(\mathbf{b}, \mathbf{c})$, $(\mathbf{b}, \mathbf{d})$, $(\mathbf{c}, \mathbf{d})$, $(\mathbf{e}, \mathbf{f})$ are white pairs. The set $\{(\mathbf{b}), (\mathbf{c}), (\mathbf{d})\}$ is monochromatic, all of its pairs are white.

Among the white pairs, $(\mathbf{b}, \mathbf{d})$ and $(\mathbf{e}, \mathbf{f})$ satisfy that they split the word into $\mathbf{u} \cdot \mathbf{v} \cdot \mathbf{w}$ with $\mathbf{u} \cdot \mathbf{v}^* \cdot \mathbf{w} \subseteq \mathbf{H}$.

Now $\mathbf{0001} \cdot (\mathbf{10122})^* \cdot \mathbf{11210000202} \subseteq \mathbf{L} \cap \mathbf{H}$.

Additional Exercises

A language is called linear iff it has a grammar where every rule is either of the form $\mathbf{A} \to \mathbf{u}$ or of the form $\mathbf{A} \to \mathbf{vBw}$; here $\mathbf{A}, \mathbf{B}$ are nonterminals and $\mathbf{u}, \mathbf{v}, \mathbf{w}$ are terminal words.

Exercise 3.30
Show that the intersection of a linear language and a regular language is linear.

Exercise 3.31
A linear grammar is called balanced iff for every rule of the form $\mathbf{A} \to \mathbf{vBw}$ it holds that $|\mathbf{v}| = |\mathbf{w}|$ and a language is called balanced linear iff it is generated by a balanced linear grammar. Is the intersection of two balanced linear languages again balanced linear? Prove the answer.

Exercise 3.32
Provide an example of a language which is linear but not balanced linear. Prove the answer.

Exercises

In the following, one considers regular expressions consisting of the symbol $\mathbf{L}$ of palindromes over $\{0, 1, 2\}$ and the mentioned operations. What is the most difficult level in the hierarchy "regular, linear, context-free, context-sensitive" such expressions can generate. It can be used that $\{10^i10^j10^k1 : i \neq j, i \neq k, j \neq k\}$ is not context-free.

Exercise 3.33: Expressions containing $\mathbf{L}$ and $\cup$ and finite sets.

Exercise 3.34: Expressions containing $\mathbf{L}$ and $\cup$ and $\cdot$ and Kleene star and finite sets.

Exercise 3.35: Expressions containing $\mathbf{L}$ and $\cup$ and $\cdot$ and $\cap$ and Kleene star and finite sets.

Exercise 3.36: Expressions containing $\mathbf{L}$ and $\cdot$ and set difference and Kleene star and finite sets.