Theory of Computation 5
Combining Languages

Frank Stephan

Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition 1

If $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{F})$ is a non-deterministic finite automaton (nfa) then δ has a set of values (not always single value), that is, for $\mathbf{p} \in \mathbf{Q}$ and $\mathbf{a} \in \mathbf{\Sigma}$ there can be several $\mathbf{q} \in \mathbf{Q}$ such that the nfa can go from $\mathbf{p}$ to $\mathbf{q}$ on symbol $\mathbf{a}$.

A run of an nfa on a word $\mathbf{a_1 a_2 \ldots a_n}$ is a sequence $\mathbf{q_0 q_1 q_2 \ldots q_n} \in \mathbf{Q}^*$ such that $\mathbf{q_0} = \mathbf{s}$ and $\mathbf{q_{m+1}} \in \delta(\mathbf{q_m}, \mathbf{a_{m+1}})$ for all $\mathbf{m} < \mathbf{n}$.

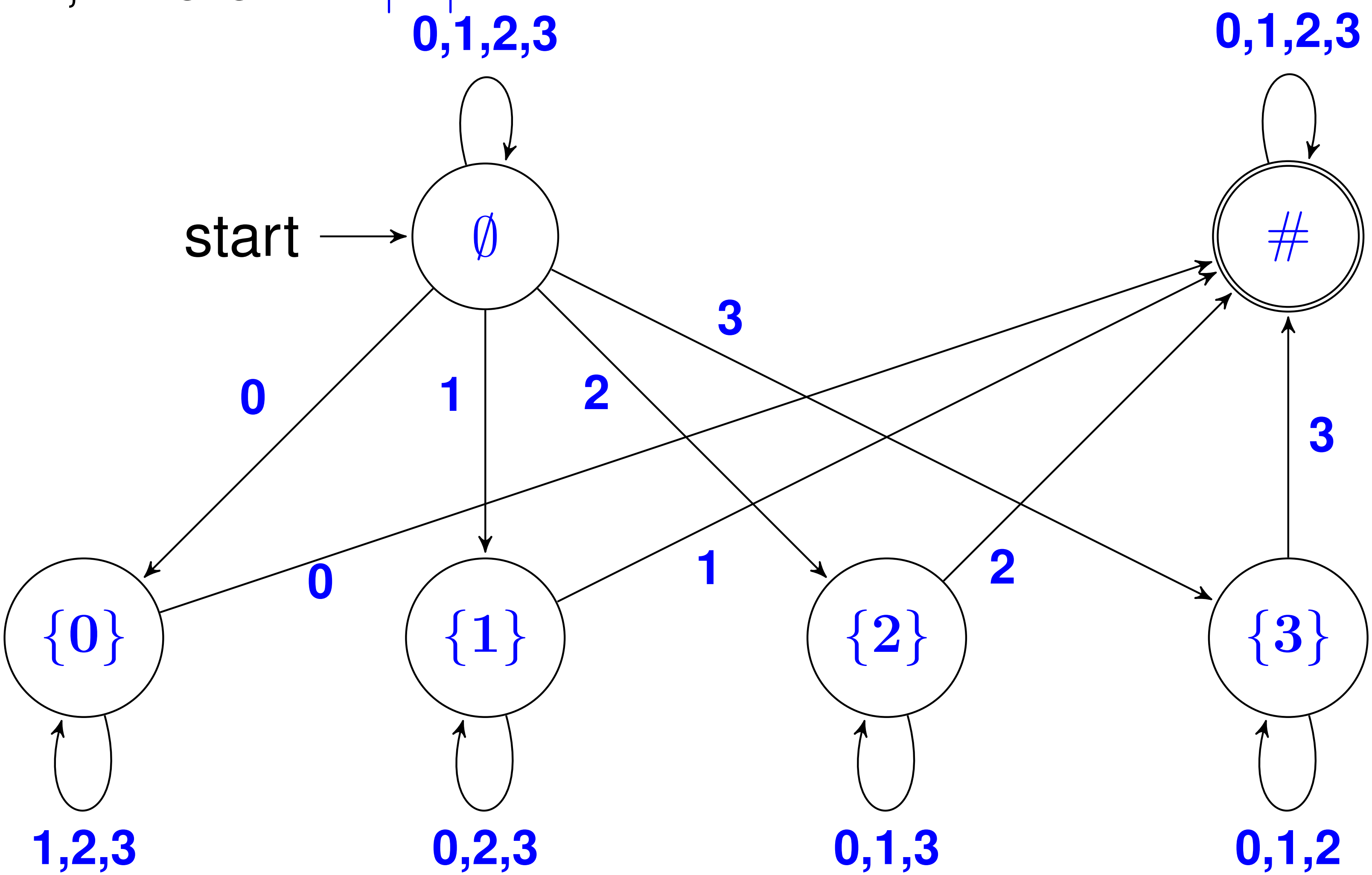
If $\mathbf{q_n} \in \mathbf{F}$ then the run is “accepting” else the run is “rejecting”.

The nfa accepts a word $\mathbf{w}$ iff it has an accepting run on $\mathbf{w}$; this is also the case if there exist other rejecting runs.

δ as relation: $(\mathbf{p}, \mathbf{a}, \mathbf{q}) \in \delta$ iff nfa can go on $\mathbf{a}$ from $\mathbf{p}$ to $\mathbf{q}$.
δ as set-valued function: $\delta(\mathbf{p}, \mathbf{a}) = \{\mathbf{q}$: nfa can go on $\mathbf{a}$ from $\mathbf{p}$ to $\mathbf{q}\}$.

Repetition 2

The language $\{w : \text{some letter appears twice}\}$ has an nfa with $n + 2$ states while a dfa needs $2^n + 1$ states; here for $n = 4$, where $n = |\Sigma|$.



Repetition 3

Given an nfa, one let for given state $\mathbf{q}$ and symbol $\mathbf{a}$ the set $\delta(\mathbf{q},\mathbf{a})$ denote all states $\mathbf{q}'$ to which the nfa can transit from $\mathbf{q}$ on symbol $\mathbf{a}$.

Theorem 4.5 [Büchi; Rabin and Scott]
For each nfa $(\mathbf{Q},\mathbf{\Sigma},\delta,\mathbf{s},\mathbf{F})$ with $\mathbf{n}=|\mathbf{Q}|$ states, there is an equivalent dfa $(\{\mathbf{Q}':\mathbf{Q}'\subseteq\mathbf{Q}\},\mathbf{\Sigma},\delta',\{\mathbf{s}\},\mathbf{F}')$ with $\mathbf{2^n}$ states such that $\mathbf{F}'=\{\mathbf{Q}'\subseteq\mathbf{Q}:\mathbf{Q}'\cap\mathbf{F}\neq\emptyset\}$ and

$$\forall\mathbf{Q}'\subseteq\mathbf{Q}\,\forall\mathbf{a}\in\mathbf{\Sigma}\,[\delta'(\mathbf{Q}',\mathbf{a}) \;=\; \bigcup_{\mathbf{q}'\in\mathbf{Q}}\delta(\mathbf{q}',\mathbf{a}) \;=\; \{\mathbf{q}''\in\mathbf{Q}:\exists\mathbf{q}'\in\mathbf{Q}'\,[\mathbf{q}''\in\delta(\mathbf{q}',\mathbf{a})]\}].$$

As the number of states is often overshooting, it is good to minimise the resulting automaton with the algorithm of Myhill and Nerode.

Repetition 4

The following statements are all equivalent to "L is regular":

(a) L is generated by a regular expression;

(b) L is generated by a regular grammar;

(c) L is recognised by a determinisitic finite automaton;

(d) L is recognised by a non-determinisitic finite automaton;

(e) L and $\Sigma^* - L$ both satisfy the Block Pumping Lemma;

(f) L satsifies Jaffe's Matching Pumping Lemma;

(g) L has only finitely many derivatives.

Product Automata

Let $(\mathbf{Q_1}, \mathbf{\Sigma}, \delta_\mathbf{1}, \mathbf{s_1}, \mathbf{F_1})$ and $(\mathbf{Q_2}, \mathbf{\Sigma}, \delta_\mathbf{2}, \mathbf{s_2}, \mathbf{F_2})$ be dfas which recognise $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively.

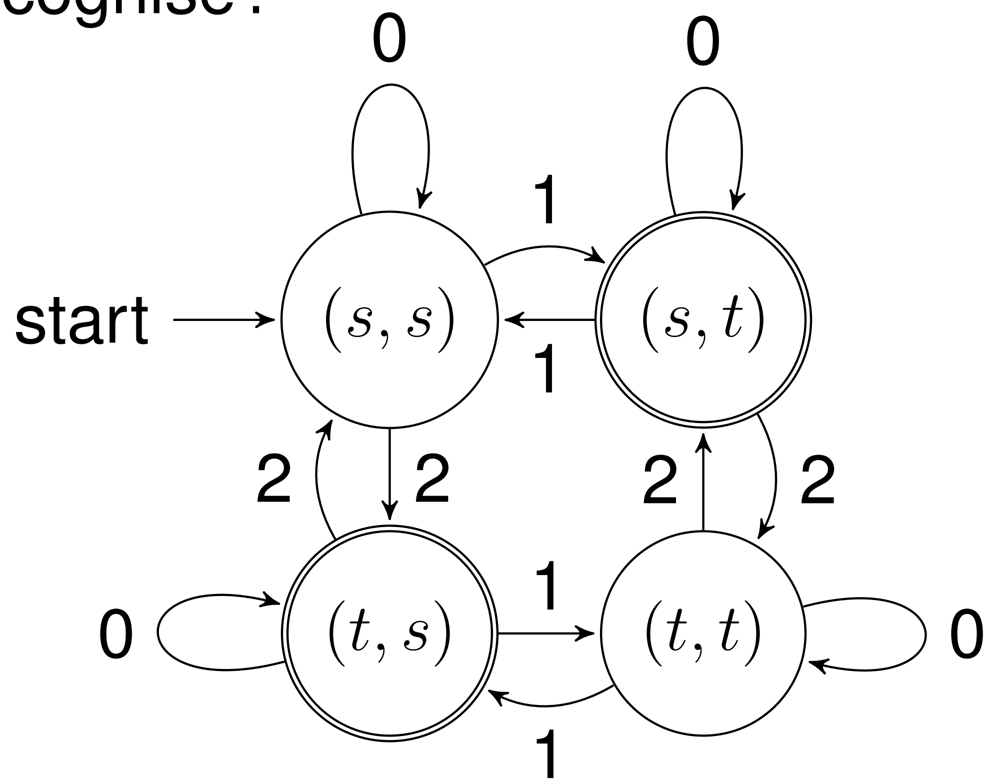
Consider $(\mathbf{Q_1} \times \mathbf{Q_2}, \mathbf{\Sigma}, \delta_\mathbf{1} \times \delta_\mathbf{2}, (\mathbf{s_1}, \mathbf{s_2}), \mathbf{F})$ with $(\delta_\mathbf{1} \times \delta_\mathbf{2})((\mathbf{q_1}, \mathbf{q_2}), \mathbf{a}) = (\delta_\mathbf{1}(\mathbf{q_1}, \mathbf{a}), \delta_\mathbf{2}(\mathbf{q_2}, \mathbf{a}))$. 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

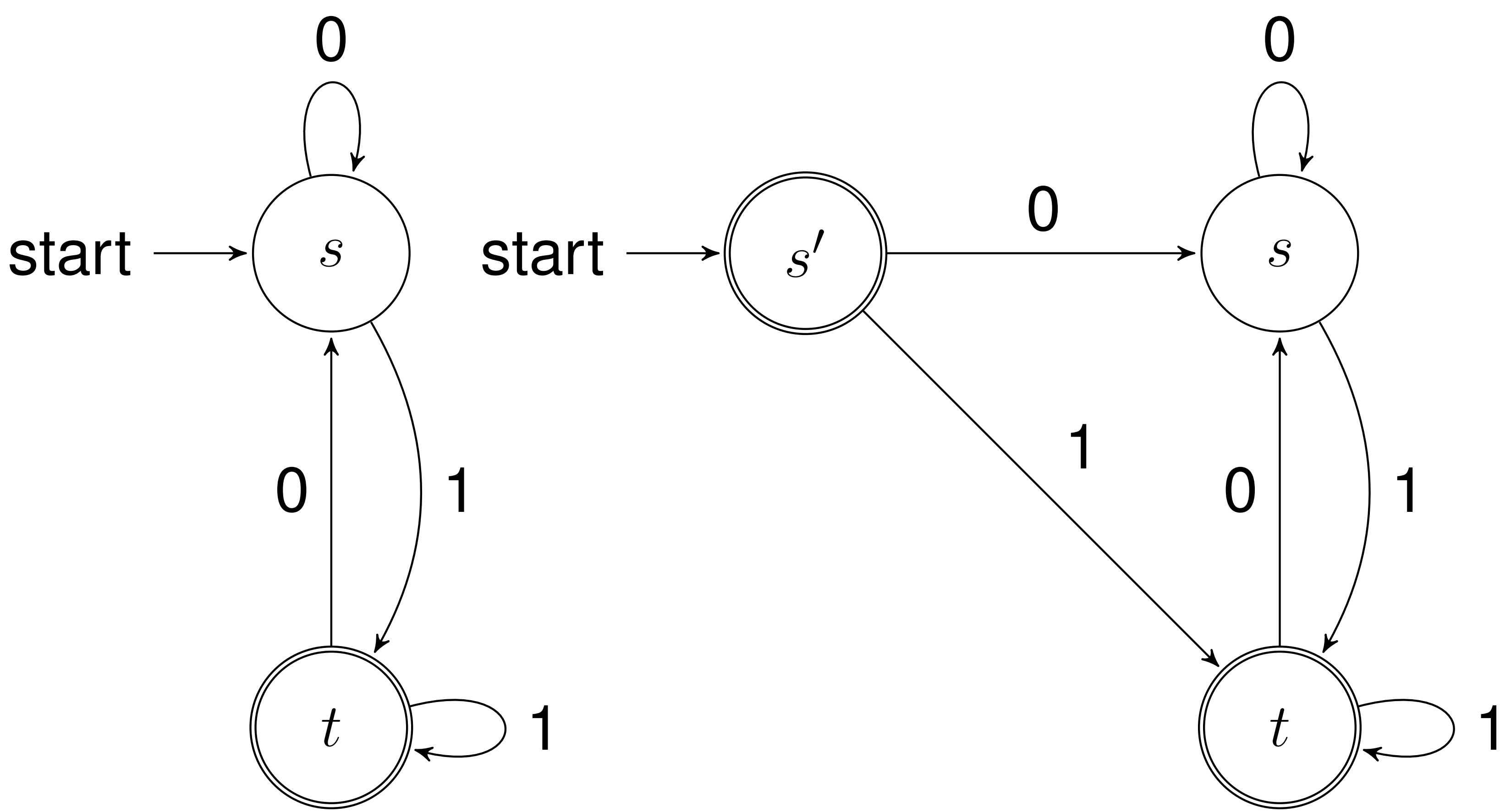
For $\mathbf{a} = \mathbf{1}, \mathbf{2}$, let automaton $(\{\mathbf{s}, \mathbf{t}\}, \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}, \delta_{\mathbf{a}}, \mathbf{s}, \{\mathbf{s}\})$ recognise when there is an even number of $\mathbf{a}$; if input $\mathbf{b}$ equals $\mathbf{a}$ then state is changed else state remains unchanged.

Quiz: Which Boolean combination does this product automaton recognise?



Kleene Star

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



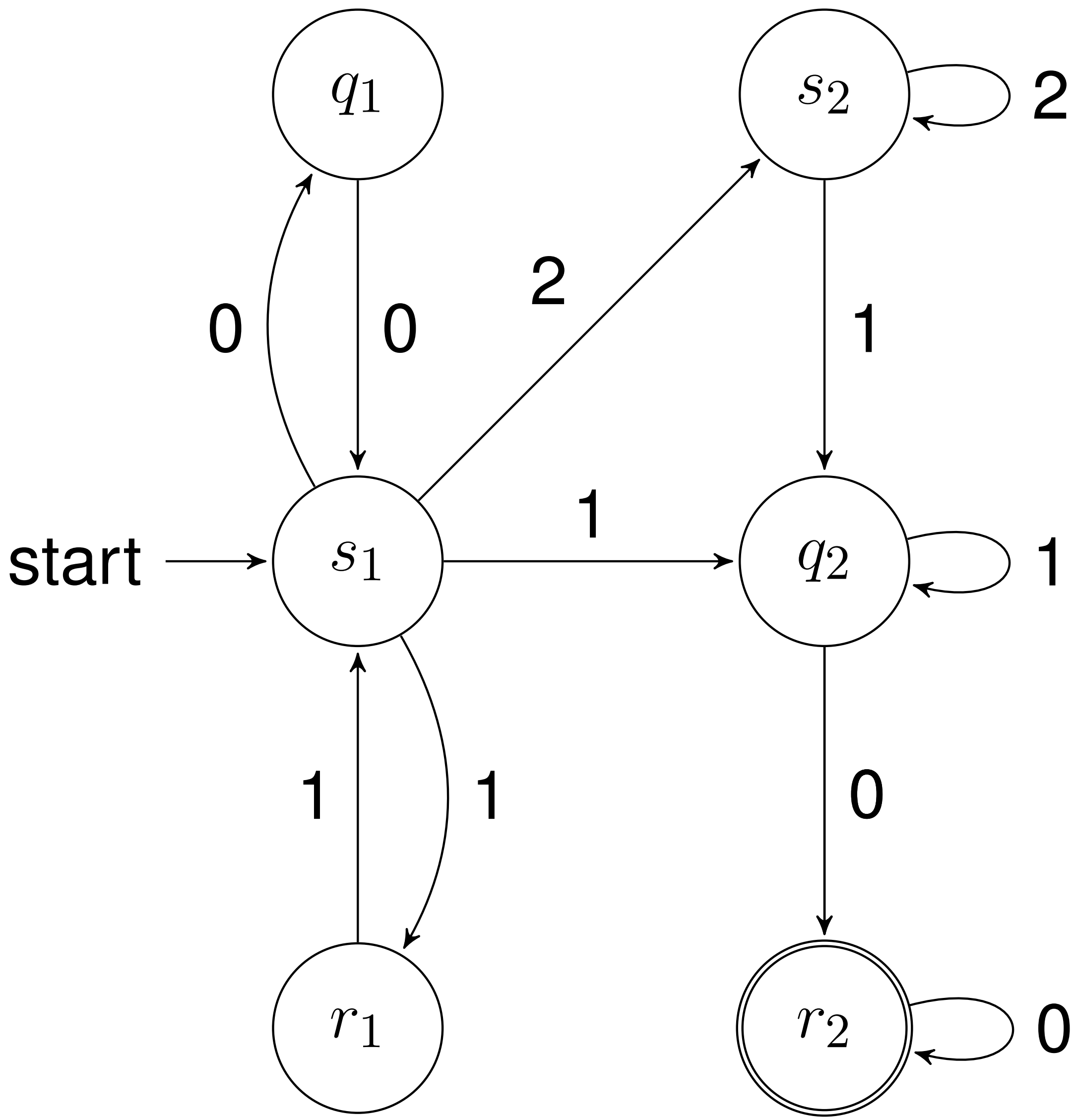
Concatenation

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 5.3

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

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

Give the bounds in terms of “linear”, “quadratic” and “exponential”. Explain your bounds.

Sample Automata

Exercise 5.4

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

Exercise 5.5

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

Exercise 5.6

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

Exercise 5.7

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

(a) $L \cup H$,

(b) $L \cdot H$ and

(c) L^*.

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

Example 5.8

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 5.9

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 5.10 and 5.11

Recall that the language $\mathbf{L}$ of all words which contain as many $\mathbf{0}$s as $\mathbf{1}$s is context-free; a grammar for it is $(\{\mathbf{S}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{S} \to \mathbf{SS}|\varepsilon|\mathbf{0S1}|\mathbf{1S0}\}, \mathbf{S})$.

Exercise 5.10

Construct a context-free grammar for $\mathbf{L} \cap (\mathbf{001}^+)^*$.

Exercise 5.11

Construct 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-senstive grammers generating $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively, where $\mathbf{N_1} \cap \mathbf{N_2} = \emptyset$ and where each rule $\mathbf{l} \to \mathbf{r}$ 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 $(\mathbf{N_1} \cup \mathbf{N_2} \cup \{\mathbf{S}\}, \mathbf{\Sigma}, \mathbf{P_1} \cup \mathbf{P_2} \cup \{\mathbf{S} \to \mathbf{S_1 S_2}\}, \mathbf{S})$ 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 $\mathbf{l} \to \mathbf{r}$ 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 $\mathbf{S} \to \mathbf{S'}|\varepsilon$ and $\mathbf{S'} \to \mathbf{S_1 S_2 S'} \,|\, \mathbf{S_1 S_2} \,|\, \mathbf{S_1}$ plus all rules in $\mathbf{P_1} \cup \mathbf{P_2}$.

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

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

(b) and (c): For each rule in $\mathbf{P_1}$, for example, for $\mathbf{AB} \rightarrow \mathbf{CDE} \in \mathbf{P_1}$, and all symbols $\mathbf{F}, \mathbf{G}, \mathbf{H}, \ldots$ in $\mathbf{N_2}$, one has the corresponding rule $\binom{\mathbf{A}}{\mathbf{F}}\binom{\mathbf{B}}{\mathbf{G}}\binom{\#}{\mathbf{H}} \rightarrow \binom{\mathbf{C}}{\mathbf{F}}\binom{\mathbf{D}}{\mathbf{G}}\binom{\mathbf{E}}{\mathbf{H}}$. 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 $\binom{\mathbf{a}}{\mathbf{a}} \rightarrow \mathbf{a}$ 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: $S \to S2|0S1|01$.
Grammar L_2: $S \to 0S|1S2|12$.

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

Deriving 001122

$$\binom{S}{S} \Rightarrow^* \binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#} \Rightarrow \binom{S}{S}\binom{2}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#} \Rightarrow^*$$

$$\binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#} \Rightarrow \binom{S}{S}\binom{2}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#} \Rightarrow^*$$

$$\binom{S}{S}\binom{\#}{\#}\binom{\#}{\#}\binom{\#}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{S}\binom{S}{\#}\binom{1}{\#}\binom{\#}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$

$$\binom{0}{S}\binom{S}{\#}\binom{\#}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{S}\binom{0}{\#}\binom{1}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$

$$\binom{0}{0}\binom{0}{S}\binom{1}{\#}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow \binom{0}{0}\binom{0}{0}\binom{1}{S}\binom{1}{\#}\binom{2}{\#}\binom{2}{\#} \Rightarrow$$

$$\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{S}\binom{2}{2}\binom{2}{\#} \Rightarrow \binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{S}\binom{2}{\#}\binom{2}{2} \Rightarrow$$

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

Exercises 5.14 and 5.17

Exercise 5.14

Let $\mathbf{L} = \{\mathbf{0^n1^n2^n} : \mathbf{n} \in \mathbb{N}\}$ and construct a context-sensitive grammar for $\mathbf{L}^*$.

Exercise 5.17

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

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 $(\mathbf{a_1 a_2 \ldots a_n})^{\mathbf{mi}} = \mathbf{a_n \ldots a_2 a_1}$ as the mirror image of a string.

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

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

Exercise 5.18

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." originating from the time when the canal in Panama was built.

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

Exercise 5.20

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 5.21

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

Exercises

Exercise 5.22

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$\}$.

Exercise 5.23

Let $\mathbf{L} = \{\mathbf{w} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^* : \mathbf{w} = \mathbf{w}^{\mathbf{mi}}\}$ and consider $\mathbf{H} = \mathbf{L} \cap \{\mathbf{012}, \mathbf{210}, \mathbf{00}, \mathbf{11}, \mathbf{22}\}^* \cap (\{\mathbf{0}, \mathbf{1}\}^* \cdot \{\mathbf{1}, \mathbf{2}\}^* \cdot \{\mathbf{0}, \mathbf{1}\}^*)$. This is the intersection of a context-free and regular language and thus context-free. Construct a context-free grammar for $\mathbf{H}$.

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

Exercise 5.25: Expressions containing $\mathbf{L}$ and $\cup$ and $\cdot$ and Kleene star and finite sets.

Exercise 5.26: Expressions containing $\mathbf{L}$ and $\cup$ and $\cdot$ and and $\cap$ and Kleene star and finite sets.

Exercise 5.27: Expressions containing $\mathbf{L}$ and $\cdot$ and set difference and Kleene star and finite sets.

Next Week's Midterm Examination

Topics

Defining and proving using structural induction

Making and analysing finite automata

Converting regular languages from one form into another form, Deterministic versus non-deterministic finite automata, Bounds on number of states

Pumping lemmas: Usage for proofs; Properties

Combining finite automata

Basic properties of context-free grammars: Making of such grammars, Usage of pumping lemma for context-free languages

Revise lecture notes; Try exercises and compare with solutions by fellow students

Example of Inductive Definition

$\varepsilon <_{ll} \mathbf{0} <_{ll} \mathbf{1} <_{ll} \mathbf{00} <_{ll} \mathbf{01} <_{ll} \mathbf{10} <_{ll} \mathbf{11} <_{ll} \mathbf{000} <_{ll} \ldots$; use this length-lexicographical order $<_{ll}$ to define $\mathbf{sw}(\mathbf{reg\,exp})$:

$$
\begin{array}{rcl}
\mathbf{sw}(\emptyset) & = & \infty; \\
\mathbf{sw}(\{\mathbf{w_1}, \ldots, \mathbf{w_n}\}) & = & \mathbf{min}_{ll}\{\mathbf{w_1}, \ldots, \mathbf{w_n}\}; \\
\mathbf{sw}(\sigma \cup \tau) & = & \begin{cases} \mathbf{sw}(\sigma) & \text{if } \mathbf{sw}(\tau) = \infty; \\ \mathbf{sw}(\tau) & \text{if } \mathbf{sw}(\sigma) = \infty; \\ \mathbf{min}_{ll}\{\mathbf{sw}(\sigma), \mathbf{sw}(\tau)\} & \text{otherwise;} \end{cases} \\
\mathbf{sw}(\sigma \cdot \tau) & = & \begin{cases} \infty & \text{if } \mathbf{sw}(\sigma) = \infty \\ & \text{or } \mathbf{sw}(\tau) = \infty; \\ \mathbf{sw}(\sigma) \cdot \mathbf{sw}(\tau) & \text{otherwise;} \end{cases} \\
\mathbf{sw}(\sigma^*) & = & \varepsilon.
\end{array}
$$

One can show by structural induction: $|\mathbf{sw}(\sigma)| \leq |\sigma|$ where $|\infty| = \mathbf{1}$ and $|\emptyset| = \mathbf{1}$ and $|\varepsilon| = \mathbf{0}$.