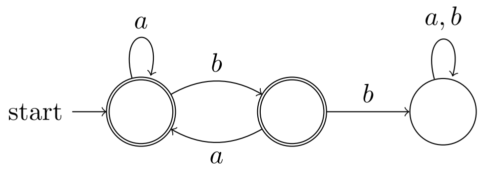
Technische Universität München **Winter term 2011/12**
I7
Prof. J. Esparza / J. Křetínský **20.10.2011**

Automata and Formal Languages – Homework 2

Due 27.11.2011.

Exercise 2.1

Let $\mathcal{A}$ be the following finite automaton:



(a) Transform the automaton $\mathcal{A}$ into an equivalent regular expression, then transform this expression into an NFA (with ε-transitions), remove the ε-transitions, and determinize the automaton.

(b) Use JFLAP to perform the same transformations. Is there any difference?

(c) Using JFLAP check that your resulting automaton is equivalent to the original one.

Exercise 2.2

Let r be the regular expression $((0+1)(0+1))^*$ over $\Sigma = \{0,1\}$, where $+$ denotes choice.

(a) Describe $\mathcal{L}(r)$ in words.

(b) Give a regular expression r' such that $\mathcal{L}(r') = \Sigma^* \setminus \mathcal{L}(r)$.

Exercise 2.3

For any language L, let L_{pref} and L_{suf} denote the languages of all prefixes and all suffixes, respectively, of words in L. E.g. for $L = \{abc, d\}$, we have $L_{\text{pref}} = \{abc, ab, a, \varepsilon, d\}$ and $L_{\text{suf}} = \{abc, bc, c, \varepsilon, d\}$.

(a) Given an automaton $\mathcal{A}$, construct automata $\mathcal{A}_{\text{pref}}$ and $\mathcal{A}_{\text{suf}}$ so that $\mathcal{L}(\mathcal{A}_{\text{pref}}) = \mathcal{L}(\mathcal{A})_{\text{pref}}$ and $\mathcal{L}(\mathcal{A}_{\text{suf}}) = \mathcal{L}(\mathcal{A})_{\text{suf}}$.

(b) Consider a regular expression $r = (ab+b)^*c$. Give a regular expression r_{pref} so that $\mathcal{L}(r_{\text{pref}}) = \mathcal{L}(r)_{\text{pref}}$.

Exercise 2.4

For $n \in \mathbb{N}_0$ let $\text{msbf}(n)$ be the language of all words over $\{0,1\}$ which represent n w.r.t. to the *most significant bit first* representation where an arbitrary number of leading zeros is allowed. For example:

$$\text{msbf}(3) = \mathcal{L}(0^*11) \text{ and } \text{msbf}(0) = \mathcal{L}(0^*).$$

Similarly, let $\text{lsbf}(n)$ denote the language of all *least significant bit first* representations of n with an arbitrary number of following zeros, e.g.:

$$\text{lsbf}(6) = \mathcal{L}(0110^*) \text{ and } \text{lsbf}(0) = \mathcal{L}(0^*).$$

(a) Construct and compare DFAs representing all even natural numbers $n \in \mathbb{N}_0$ w.r.t. the unary encoding (i.e., $n \mapsto a^n$), the msbf encoding, and the lsbf encoding.

(b) Construct a DFA representing the language $\{w \in \{0,1\}^* \mid \text{lsbf}^{-1}(w) \text{ is divisible by } 3\}$.

(c) Give regular expressions corresponding to the languages in (a) and (b).

Exercise 2.5

Let Σ_1, Σ_2 be two alphabets. A map $h : \Sigma_1^* \to \Sigma_2^*$ is called a *homomorphism* if it respects the empty word and concatenation, i.e.,

$$h(\varepsilon) = \varepsilon \text{ and } h(w_1 w_2) = h(w_1)h(w_2) \text{ for all } w_1, w_2 \in \Sigma_1^*.$$

Assume that $h : \Sigma_1^* \to \Sigma_2^*$ is a homorphism. Note that h is completely determined by its values on Σ_1.

(a) Let $\mathcal{A}$ be a finite automaton over the alphabet Σ_1. Describe how to constuct a finite automaton accepting the language

$$h(\mathcal{L}(\mathcal{A})) := \{h(w) \mid w \in \mathcal{L}(\mathcal{A})\}.$$

(b) Let $\mathcal{A}'$ be a finite automaton over the alphabet Σ_2. Describe how to construct a finite automaton accepting the language

$$h^{-1}(\mathcal{L}(\mathcal{A}')) := \{w \in \Sigma_1^* \mid h(w) \in \mathcal{L}(\mathcal{A}')\}.$$

(c) Recall that the language $\{0^n 1^n \mid n \in \mathbb{N}\}$ is context free, but not regular. Use the preceding results to show that $\{(01^k 2)^n 3^n \mid k, n \in \mathbb{N}\}$ is also not regular.

Exercise 2.6

For alphabets Σ and Δ, a *substitution* is a mapping $f : \Sigma \to 2^{\Delta^*}$ assigning to each letter $a \in \Sigma$ a language $L_a \subseteq \Delta^*$. Then by setting $f(\varepsilon) = \varepsilon$ and $f(wa) = f(w)f(a)$ we can define $f(L) = \bigcup_{w \in L} f(w)$.

Note that a homomorphism can be identified with a subsitution where all L_a's are singletons.

Consider the following example. Let $\Sigma = \{Name, Telephone, :, \#\}$ and $\Delta = \{A, \ldots, Z, 0, 1, \ldots, 9, :, \#\}$. The substitution f is given by

$$\begin{aligned}
&f(Name) = \mathcal{L}((A + \cdots + Z)^*) \\
&f(:) = \{:\} \\
&f(Telephone) = \mathcal{L}(0049(1 + \ldots + 9)(0 + 1 + \ldots + 9)^{10} + 00420(1 + \ldots + 9)(0 + 1 + \ldots + 9)^8) \\
&f(\#) = \{\#\}
\end{aligned}$$

(a) Draw a DFA recognizing $L = Name : Telephone\{\#Telephone\}^*$.

(b) Sketch an NFA-reg recognizing $f(L)$.

(c) Given an automaton recognizing L', a substitution f', and automata recognizing $f'(a)$ for every a, construct an automaton recognizing $f'(L')$.

Exercise 2.7

For L_1, L_2 regular languages over an alphabet Σ, the *left quotient* of L_1 by L_2 is defined by

$$L_2 \diagdown L_1 := \{v \in \Sigma^* \mid \exists u \in L_2 : \ uv \in L_1\}$$

(a) Use the fact that regular languages are closed under homomorphisms, inverse homomorphisms, concatenation and intersection to prove they are closed under quotienting.

(b) Given finite automata $\mathcal{A}_1, \mathcal{A}_2$, construct an automaton $\mathcal{A}$ such that

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_2) \diagdown \mathcal{L}(\mathcal{A}_1)$$

(c) Is there any difference when taking the *right quotient* $L_1 \diagup L_2 := \{u \in \Sigma^* \mid \exists v \in L_2 : \ uv \in L_1\}$?