Technische Universität München **Winter term 2019/20**
I7
Prof. J. Křetínský / M. Lazić / S. Sickert-Zehnter / C. Weil-Kennedy **12.12.2019**

Automata and Formal Languages — Exercise Sheet 10

Exercise 10.1
Construct a finite automaton for the Presburger formula $\exists y.\ x = 3y$ using the algorithms of the chapter.

Exercise 10.2
Let $\Sigma = \{a, b\}$. Give formulations in plain English of the languages described by the following formulas of FO(Σ), and give a corresponding regular expression:

(a) $\exists x.\ \mathit{first}(x)$

(b) $\forall x.\ \mathit{false}$

(c) $\neg\exists x.\exists y.\ \big(x < y \wedge Q_a(x) \wedge Q_b(y)\big) \wedge \forall x.\ \big(Q_b(x) \rightarrow \exists y.\ x < y \wedge Q_a(y)\big) \wedge \exists x.\neg\exists y.\ x < y$

Exercise 10.3
Give a MSO(Σ) formula $\mathsf{Odd\_Card}(X)$ expressing that the cardinality of the set of positions X is odd.

Exercise 10.4
Let Σ be a finite alphabet. A language $L \subseteq \Sigma^*$ is *star-free* if it can be expressed by a star-free regular expression, i.e. a regular expression where the Kleene star operation is forbidden, but complementation is allowed. For example, Σ^* is star-free since $\Sigma^* = \overline{\emptyset}$, but $(aa)^*$ is not.

(a) Give star-free regular expressions and FO(Σ) sentences for the following star-free languages:

(i) Σ^+.

(ii) $\Sigma^* A \Sigma^*$ for some $A \subseteq \Sigma$.

(iii) A^* for some $A \subseteq \Sigma$.

(iv) $(ab)^*$.

(v) $\{w \in \Sigma^* \mid w \text{ does not contain } aa\ \}$.

(b) Show that finite and cofinite languages are star-free.

(c) Show that for every sentence $\varphi \in \mathrm{FO}(\Sigma)$, there exists a formula $\varphi^+(x, y)$, with two free variables x and y, such that for every $w \in \Sigma^+$ and for every $1 \leq i \leq j \leq w$,

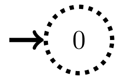
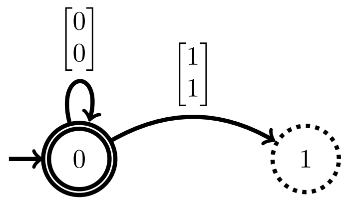
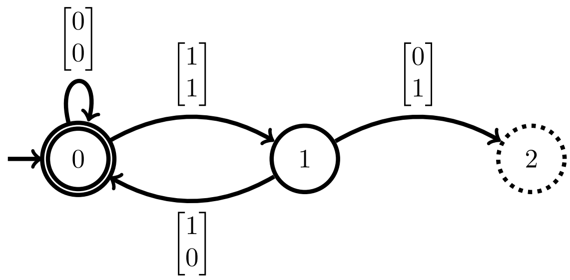
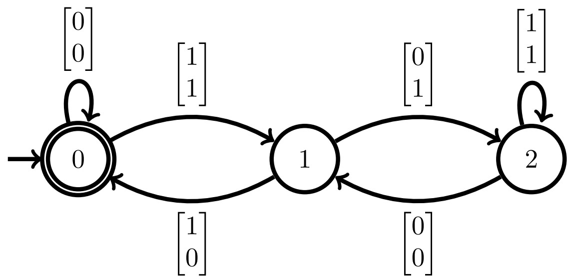
$$w \models \varphi^+(i, j) \quad \text{iff} \quad w_i w_{i+1} \cdots w_j \models \varphi\ .$$

(d) Give a polynomial time algorithm that decides whether the empty word satisfies a given sentence of FO(Σ).

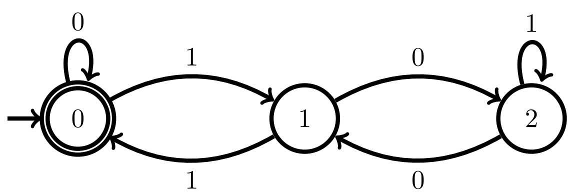
(e) Show that every star-free language can be expressed by an FO(Σ) sentence.

Solution 10.1

We can rewrite the formula as $\exists y.\ x - 3y = 0$. We first use *EqtoDFA* to obtain an automaton for $x - 3y = 0$:

Iter.	**Current automaton**	W
0		$\{0\}$
1		$\{1\}$
2		$\{2\}$
3		$\emptyset$

It remains to project the automaton on x, i.e. on the first component of the letters. We obtain:



Solution 10.2

(a) All nonempty words. The regular expression is $(a + b)(a + b)^*$

(b) The empty word. The regular expression is ϵ.

(c) The first conjunct expresses that no a precedes a b. The corresponding regular expression is b^*a^*. The second conjunct states that every b is followed (not necessarily immediately) by an a; this excludes the words of b^*. Finally, the third conjunct expresses that the last letter exists (and, by the second conjunct, must be an a), which excludes the empty word. So the regular expression is b^*aa^*

Solution 10.3

We first give formulas $\text{First}(x, X)$ and $\text{Last}(x, X)$ expressing that x is the first/last position among those in X. We also give a formula $\text{Next}(x, y, X)$ expressing that y is the succesor of x in X. It is then easy to give a formula $\text{Odd}(Y, X)$ expressing that Y is the set of odd positions of X (more precisely, Y contains the first position among those in X, the third, the fifth, etc.). Finally, the formula $\text{Odd\_card}(X)$ expresses that the last position of X belongs to the set of odd positions of X.

$$\begin{aligned}
\text{First}(x, X) &:= x \in X \wedge \forall y\, y < x \to y \notin X \\
\text{Last}(x, X) &:= x \in X \wedge \forall y\, y > x \to y \notin X \\
\text{Next}(x, y, X) &:= x \in X \wedge y \in X \wedge x < y \wedge \neg \exists z\, x < z \wedge z < y \wedge z \in X \\
\text{Odd}(Y, X) &:= \forall x (x \in Y \leftrightarrow \big(\text{First}(x, X) \vee \exists z\, \exists u\, z \in Y \wedge \text{Next}(z, u, X) \wedge \text{Next}(u, x, X)) \\
\text{Odd_card}(X) &= \exists Y\, \big(\text{Odd}(Y, X) \wedge \forall x\ \text{Last}(x, X) \to x \in Y\big)
\end{aligned}$$

Solution 10.4

(a) (i) $\overline{\emptyset} \cdot \Sigma$ and $\exists x\ \text{first}(x)$.

(ii) $\overline{\emptyset} \cdot A \cdot \overline{\emptyset}$ and $\exists x\ \bigvee_{a \in A} Q_a(x)$.

(iii) $\overline{\Sigma^* \overline{A} \Sigma^*}$ and $\forall x\ \bigvee_{a \in A} Q_a(x)$.

(iv) $\overline{b\Sigma^* + \Sigma^* a + \Sigma^* aa \Sigma^* + \Sigma^* bb \Sigma^*}$ and

$$\begin{aligned}
&(\neg \exists x\ \text{first}(x))\ \vee \\
&\big((\exists x\ \text{first}(x) \wedge Q_a(x)) \wedge (\exists y\ \text{last}(y) \wedge Q_b(y))\ \wedge \\
&\quad (\forall x\ \forall y\ (Q_a(x) \wedge y = x + 1) \to Q_b(y)) \wedge\ (\forall x\ \forall y\ (Q_b(x) \wedge y = x + 1) \to Q_a(y)) \big)\,.
\end{aligned}$$

(v) $\overline{\Sigma^* aa \Sigma^*}$ and $\forall x\ \forall y\ (Q_a(x) \wedge y = x + 1) \to \neg Q_a(y)$.

(b) Every finite language $L = \{w_1, w_2, \ldots, w_m\}$ can be expressed as $w_1 + w_2 + \cdots + w_m$. For every cofinite language L, there exists a finite language A such that $L = \overline{A}$. Since star-free regular expressions allow for complementation, cofinite languages are also star-free. $\square$

(c) We build φ^+ using the following inductive rules:

$$\begin{aligned}
(x < y)^+(i, j) &= x < y \\
Q_a(x)^+(i, j) &= Q_a(x) \\
(\neg \psi)^+(i, j) &= \neg \psi^+(i, j) \\
(\psi_1 \vee \psi_2)^+(i, j) &= \psi_1^+(i, j) \vee \psi_2^+(i, j) \\
(\exists x\ \psi)^+(i, j) &= \exists x\ (i \leq x \wedge x \leq j) \wedge \psi^+(i, j)\ .
\end{aligned}$$

(d)

Input: sentence $\varphi \in \text{FO}(\Sigma)$.
Output: $\varepsilon \models \varphi$?

has-empty(φ):
  if $\varphi = \neg \psi$ **then**
    return $\neg$has-empty(ψ)
  else if $\varphi = \psi_1 \vee \psi_2$ **then**
    return has-empty(ψ_1) $\vee$ has-empty(ψ_2)
  else if $\varphi = \exists\ \psi$ **then**
    return *false*

(e)

Input: star-free regular expression r.
Output: sentence $\varphi \in \mathrm{FO}(\Sigma)$ s.t. $L(\varphi) = L(r)$.

`formula(`r`)`*:*
  if $r = \varepsilon$ **then**
    return $\forall x$ false
  else if $r = a$ for some $a \in \Sigma$ **then**
    return $(\exists x \text{ true}) \wedge (\forall x \text{ first}(x) \wedge Q_a(x))$
  else if $r = \overline{s}$ **then**
    return $\neg$`formula(`s`)`
  else if $r = s_1 + s_2$ **then**
    return `formula(`s_1`)` $\vee$ `formula(`s_2`)`
  else if $r = s_1 \cdot s_2$ **then**
    return $(\forall x \text{ false} \wedge (\varepsilon \in L(s_1)) \wedge (\varepsilon \in L(s_2))) \vee$
      (`formula(`s_1`)` $\wedge (\varepsilon \in L(s_2))) \vee$
      $((\varepsilon \in L(s_1)) \wedge$ `formula(`s_2`)`$) \vee$
      $(\exists x, y, y', z \text{ first}(x) \wedge y' = y + 1 \wedge \text{last}(z) \wedge \texttt{formula(}s_1\texttt{)}^{+}(x, y) \wedge \texttt{formula(}s_2\texttt{)}^{+}(y', z))$