Automata and Formal Languages — Exercise Sheet 7

Exercise 7.1
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) $\forall x.\ \mathit{last}(x)$

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

Exercise 7.2
Let $\Sigma = \{a, b\}$.

(a) Give an MSO(Σ) sentence for aa^*b^*.

(b) Give an MSO(Σ) sentence for the set of words with an a at every odd position.

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

(d) Give an MSO(Σ) sentence for the set of words with an even number of occurrences of a's.

Exercise 7.3
Recall the syntax of MSO(Σ):

$$\varphi := Q_a(x) \mid x < y \mid x \in X \mid \neg\varphi \mid \varphi \lor \varphi \mid \exists x\ \varphi \mid \exists X\ \varphi$$

We have introduced $y = x + 1$ ("y is the successor position of x") as an abbreviation

$$y = x + 1 :=\ x < y \land \neg\exists z\ (x < z \land z < y)$$

Consider now the variant MSO$'(\Sigma)$ in which, instead of an abbreviation, $y = x + 1$ is part of the syntax and replaces $x < y$. In other words, the syntax of MSO$'(\Sigma)$ is

$$\varphi := Q_a(x) \mid y = x + 1 \mid x \in X \mid \neg\varphi \mid \varphi \lor \varphi \mid \exists x\ \varphi \mid \exists X\ \varphi$$

Prove that MSO$'(\Sigma)$ has the same expressive power as MSO(Σ) by finding a formula of MSO$'(\Sigma)$ with the same meaning as $x < y$.

Exercise 7.4
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 with $\Sigma = \{a, b\}$:

  (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 $\mathrm{FO}(\Sigma)$.

(e) Show that every star-free language can be expressed by an $\mathrm{FO}(\Sigma)$ sentence.

Solution 7.1

(a) All nonempty words. The regular expression is $\Sigma\Sigma^*$

(b) The empty word and words of one letter. The regular expression is $\epsilon + \Sigma$.

(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 7.2

(a) $\exists x\, Q_a(x) \;\wedge\; \big(\forall x \forall y\; (Q_a(x) \wedge Q_b(y)) \rightarrow x < y\big)$

(b) We first define a formula that asserts that a set contains the odd positions:

$$\text{odd}(P) = \forall p\colon (p \in P \leftrightarrow (\text{first}(p) \vee \exists q\colon (p = q + 2 \wedge q \in P))).$$

The sentence for the given language is:

$$\exists O\colon (\text{odd}(O) \wedge (\forall p\colon p \in O \rightarrow Q_a(p)).$$

(c) 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.

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

The subformula $\exists x\, x \in X$ is added to $\text{Odd\_card}(X)$ to make sure that X is not the empty set. Indeed $\exists Y\, \big(\text{Odd}(Y, \emptyset) \wedge \forall x\ \text{Last}(x, \emptyset) \rightarrow x \in Y\big)$ evaluates to true for Y the empty set (thanks to Jakob Schulz for pointing this out).

(d) Let $\text{Even\_card}(X) = \exists Y\, \big(\text{Odd}(Y, X) \wedge \forall x\ \text{Last}(x, X) \rightarrow x \notin Y\big)$. Then the solution is

$$\exists X\colon \text{Even\_card}(X) \wedge (\forall x\colon x \in X \leftrightarrow Q_a(x)).$$

Solution 7.3

Observe that $x < y$ holds iff there is a set Y of positions containing y and satisfying the following property: every $z \in Y$ is either the successor of x, or the successor of another element of Y. Formally:

$$x < y := \exists Y\; \big(y \in Y\big) \;\wedge\; \Big(\forall z\; z \in Y \leftrightarrow \big(z = x + 1 \vee \exists u \in X\; z = u + 1\big)\Big)$$

Solution 7.4

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

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

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

Notice that the FO sentences presented here are correct even if Σ is more than $\{a, b\}$. However the regular expression of (iv) does require $\Sigma = \{a, b\}$. For example if $\Sigma = \{a, b, c\}$ we would have c in the language of the star-free expression, but c is not in $(ab)^*$.

(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. □

(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 $\varphi \in \mathrm{FO}(\Sigma)$.
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) Given a star-free regular expression r, we build sentence $\varphi_r \in \mathrm{FO}(\Sigma)$ s.t. $L(\varphi_r) = L(r)$ using the following inductive rules:

$r = \emptyset \ \rightarrow\ \varphi_r = \exists x\ false$
$r = \varepsilon \ \rightarrow\ \varphi_r = \forall x\ false$
$r = a \ \rightarrow\ \varphi_r = (\exists x\ true) \wedge (\forall x\ first(x) \wedge Q_a(x))$
$r = \overline{s} \ \rightarrow\ \varphi_r = \neg\varphi_s$
$r = s_1 + s_2 \ \rightarrow\ \varphi_r = \varphi_{s_1} \vee \varphi_{s_2}$
$r = s_1 \cdot s_2 \ \rightarrow\ \varphi_r = (\varphi_{s_1} \wedge \varepsilon \in L(s_2)) \vee (\varepsilon \in L(s_1) \wedge \varphi_{s_2}) \vee (\exists x, y, y', z\ \mathrm{first}(x) \wedge y' = y + 1 \wedge \mathrm{last}(z) \wedge \varphi_{s_1}^+(x, y) \wedge \varphi_{s_2}^+(y', z))$

where $\varepsilon \in L(s_i)$ is syntactic sugar for *true* or *false*, and we can decide which of these it stands for using the algorithm of (d).