Technische Universität München **Winter term 2015/16**
I7
Prof. J. Esparza / Dr. D. Suwimonteerabuth / Dr. C. Broadbent **28.01.2016**

Automata and Formal Languages — Sample Solution 15

Due 01.02.2016

Solution 15.1

(a) $\mathcal{I}(X) = \{2 \cdot i \mid i \in \mathbb{N}\}$

(b) $\mathcal{I}(X) = \{2, 4\}$, $\mathcal{I}(x) = 2$, and $\mathcal{I}(y) = 4$

(c) $\mathcal{I}(X) = \{1\}$ and $\mathcal{I}(Y) = \{3\}$

Solution 15.2

The solutions provided here are straightforwardly translated from the problem statements. As discussed in the class, there are (much) shorter formulae than these.

Note: See previous exercises for the macro definitions.

(a) $\exists x\ (\text{first}(x) \wedge Q_a(x) \wedge \forall y\, (x < y \rightarrow Q_b(y)))$

(b) $\forall x\, (Q_a(x) \vee Q_b(x))$

(c) $\forall x\ (\text{first}(x) \rightarrow Q_a(x) \wedge \exists y\, (y = x + 1 \wedge (Q_a(x) \rightarrow Q_b(y)) \wedge (Q_b(x) \rightarrow Q_a(y))))$

(d) $\forall x\, (Q_a(x) \rightarrow \exists y\, (x < y \wedge Q_b(y))) \wedge \forall x\, \neg Q_c(x)$

(e) $\exists X\ (\forall x \in X\ Q_b(x) \wedge \forall y \notin X\ Q_a(y))$

(f) $\exists X\ (\text{Even}(X) \wedge \forall x \in X\ Q_a(x))$

(g) $\neg\, (\exists x\, Q_a(x) \wedge \exists y\ y = x + 1 \wedge Q_a(y) \wedge \exists z\ z = y + 1 \wedge Q_a(z))$

(h)

$$\begin{aligned} \neg\exists X(\text{Cons}(X) \quad & \wedge \quad \exists x\ \text{first}(x, X) \wedge Q_a(x) \\ & \wedge \quad \exists y\ \text{last}(x, X) \wedge Q_a(y) \\ & \wedge \quad \exists z\ x < z \wedge z < y \wedge Q_a(z) \\ & \wedge \quad \forall w\ x < w \wedge w < z \rightarrow Q_c(w) \\ & \wedge \quad \forall w\ z < w \wedge w < y \rightarrow Q_c(w)) \end{aligned}$$

(i) $\exists x\ \forall y\ (x < y \rightarrow \neg Q_a(y))$

(j) $\forall x\ \exists y\ (x < y \wedge Q_a(y))$

Solution 15.3

Recall the syntax of ω-regular expressions $s ::= r^\omega \mid r \cdot s_1 \mid s_1 + s_2$, where r is a regular expression. We can write down a recursive algorithm as follows:

Algorithm 1: ω-RegToMSO

Data: ω-regular expression s
Result: sentence ψ such that $L(\psi) = L_\omega(s)$
if $s = r^\omega$ **then**
  return Omega(RegToMSO(r));
else if $s = r \cdot s_1$ **then**
  return Concat(RegToMSO(r), ω-RegToMSO(s_1));
else if $s = s_1 + s_2$ **then**
  return ω-RegToMSO(s_1) $\vee$ ω-RegToMSO(s_2);

---

where

$$
\begin{aligned}
\text{Concat}(\varphi, \psi) \quad &:= \quad \exists X\ \exists Y\ \forall x\ (x \in X \vee x \in Y) \\
&\quad\ \wedge \forall x\ \forall y\ (x \in X \wedge y \in Y \rightarrow x < y) \\
&\quad\ \wedge \varphi^X \wedge \varphi^Y \\
\text{Block}(Y, X) \quad &:= \quad \exists x \in X \exists z \in X (\text{Next}(x, z, X) \wedge \forall y\ (y \in Y \leftrightarrow (x \leq y \wedge y < z))) \\
\text{Omega}(\varphi) \quad &:= \quad \exists X\ \forall x\ (\text{first}(x) \rightarrow x \in X) \wedge\ \forall Y\ (\text{Block}(Y, X) \rightarrow \varphi^Y)
\end{aligned}
$$

The "projection" formula φ^X is inductively defined as follows:

- If $\varphi = Q_a(x)$, $x < y$, $x \in X$, $\neg\varphi_1$, or $\varphi_1 \vee \varphi_2$, then $\varphi^X = \varphi$
- If $\varphi = \exists x\ \varphi_1$, then $\varphi^X = \exists x \in X\ \varphi_1^X$
- If $\varphi = \exists Y\ \varphi_1$, then $\varphi^X = \exists Y\ (\forall x\ (x \in Y \rightarrow x \in X) \wedge \varphi_1^X)$