Prof. Javier Esparza
Philipp Czerner

Technical University of Munich
Chair for Foundations of Software Reliability

Automata and Formal Languages

Winter Term 2023/24 – Exercise Sheet 9

Exercise 9.1.

Let $\Sigma := \{a, b\}$. Construct the following MSO(Σ) formulae.

(a) $\text{Pos}_a(X)$: X contains exactly the positions with an a.

(b) $\text{Between}(X, Y)$: between every two elements of X there is an element of Y.

(c) $\text{Without}(X, Y, Z)$: $Z = X \setminus Y$.

(d) $\text{Min}(X, x)$, $\text{Max}(X, x)$: x is the minimum/maximum of X.

(e) $\text{EvenSize}(X)$: $|X|$ is even.

(f) φ with $L(\varphi) = \{w \in \Sigma^* : |w|_a, |w|_b \text{ even}\}$.

Solution.

(a) $\text{Pos}_a(X) = \forall x(x \in X \leftrightarrow Q_a(x))$

(b) $\text{Between}(X, Y) = \forall x \forall y(x \in X \wedge y \in X \rightarrow \exists z(z \in Y \wedge x < z \wedge z < y))$

(c) $\text{Without}(X, Y, Z) = \forall x(x \in Z \leftrightarrow (x \in X \wedge \neg(x \in Y)))$

(d) $\text{Min}(X, x) = x \in X \wedge \forall y(y \in X \rightarrow (x = y \vee x < y))$, for Max replace $<$ with $>$.

(e)
$$\begin{aligned}\text{EvenSize}(X) = \exists Y \exists Z \big(&\text{Without}(X, Y, Z) \wedge \text{Between}(Y, Z) \wedge \text{Between}(Z, Y) \\ &\wedge \exists x \exists y(\text{Min}(X, x) \wedge \text{Min}(Y, x) \wedge \text{Max}(X, y) \wedge \text{Max}(Z, y))\big)\end{aligned}$$

(f) $\varphi = \exists X \exists Y(\text{Pos}_a(X) \wedge \text{Pos}_b(Y) \wedge \text{EvenSize}(X) \wedge \text{EvenSize}(Y))$

Exercise 9.2.

Consider the logic PureMSO(Σ) with syntax

$$\varphi := X \subseteq Q_a \mid X < Y \mid X \subseteq Y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists X.\ \varphi$$

Notice that formulas of PureMSO(Σ) do not contain first-order variables. The satisfaction relation of PureMSO(Σ) is given by:

$$\begin{array}{llll}(w, \mathcal{J}) & \models & X \subseteq Q_a & \text{iff} \quad w[p] = a \text{ for every } p \in \mathcal{J}(X) \\ (w, \mathcal{J}) & \models & X < Y & \text{iff} \quad p < p' \text{ for every } p \in \mathcal{J}(X),\ p' \in \mathcal{J}(Y) \\ (w, \mathcal{J}) & \models & X \subseteq Y & \text{iff} \quad p \in \mathcal{J}(Y) \text{ for every } p \in \mathcal{J}(X)\end{array}$$

with the rest as for MSO(Σ).

Prove that MSO(Σ) and PureMSO(Σ) have the same expressive power for sentences. That is, show that for every sentence ϕ of MSO(Σ) there is an equivalent sentence ψ of PureMSO(Σ), and vice versa.

Solution. Given a sentence ψ of PureMSO(Σ), let ϕ be the sentence of MSO(Σ) obtained by replacing every subformula of ψ of the form

$$\begin{array}{lll}X \subseteq Y & \text{by} & \forall x\ (x \in X \rightarrow x \in Y) \\ X \subseteq Q_a & \text{by} & \forall x\ (x \in X \rightarrow Q_a(x)) \\ X < Y & \text{by} & \forall x\ \forall y\ (x \in X \wedge y \in Y) \rightarrow x < y\end{array}$$

Clearly, ϕ and ψ are equivalent. For the other direction, let

$$\text{empty}(X) := \forall Y \; X \subseteq Y$$

and

$$\text{sing}(X) := \neg\text{empty}(X) \land \forall Y \; (Y \subseteq X) \to (\text{empty}(Y) \lor Y = X).$$

Intuitively, $\text{empty}(X)$ is true iff X is the empty set and $\text{sing}(X)$ is true iff X is a set of size one.

Let ϕ be a sentence of $\text{MSO}(\Sigma)$. Assume without loss of generality that for every first-order variable x the second-order variable X does not appear in ϕ (if necessary, rename second-order variables appropriately). Let ψ be the sentence of $\text{PureMSO}(\Sigma)$ obtained by replacing every subformula of ϕ of the form

$\exists x \; \psi'$	by	$\exists X \; (\text{sing}(X) \land \psi'[X/x])$
		where $\psi'[X/x]$ is the result of substituting X for x in ψ'
$Q_a(x)$	by	$X \subseteq Q_a$
$x < y$	by	$X < Y$
$x \in Y$	by	$X \subseteq Y$

Clearly, ϕ and ψ are equivalent.

Exercise 9.3.

Let $r \geq 0$ and $n \geq 1$. Give a Presburger formula φ such that

$$(x, y) \in \text{Sol}(\varphi) \quad \text{iff} \quad x > y \text{ and } x - y \equiv r \pmod{n}$$

Give an automaton that accepts the solutions of φ for $r = 1$ and $n = 2$. (It is not necessary to use the algorithm from the lecture to construct this automaton.)

Solution. Let $0 \leq r' < n$ such that $r' \equiv r \pmod{n}$. Since n and r are fixed constants, r' is also a fixed constant. Further, since n is a constant, we can multiply a variable by n via iterated addition. The required formula is then given by:

$$\varphi(x, y) := (x > y) \land \exists a \; (x = y + n \cdot a + r').$$

Let $k \in \mathbb{N}$ and $x, y \in \Sigma^k$ be LSBF encodings of some naturals. First note that $\text{val}(x) - \text{val}(y) \equiv 1 \pmod{2}$ iff $\text{val}(x)$ and $\text{val}(y)$ are such that one is odd and the other one is even. Thus, the first bit of x and y should be different. Moreover, $\text{val}(x) > \text{val}(y)$ iff there exists $\ell \in \{1, \ldots, k\}$ such that $x_\ell = 1$, $y_\ell = 0$, and $x_i \geq y_i$ for every $\ell < i \leq k$. These observations yield the following automaton: