Technische Universität München **Winter term 2021/22**
Prof. J. Esparza / A. R. Balasubramanian / C. Weil-Kennedy **I7**

Automata and Formal Languages — Exercise Sheet 8

Exercise 8.1

Let c a channel. A process can send a message m to the channel with the instruction $c \,!\, m$. A process can also consume the first message of the channel with the instruction $c \,?\, m$. If the channel is full when executing $c \,!\, m$, then the process blocks and waits until it can send m. When a process executes $c \,?\, m$, it blocks and waits until the first message of the channel becomes m.

Suppose there are two processes being executed concurrently that communicate through a channel c. Channel c is a queue that can store up to 1 message. The two processes follow these two algorithms respectively:

```
process(1):
  while true do
    c ! m
    /* critical section */
    c ? m
```

```
process(2):
  while true do
    c ? m
    c ? m
    /* critical section */
    c ! m
```

a. Model the program by constructing a network of three automata:
   - One for process 1, using the alphabet $\Sigma_1 = \{c?m, c!m, cs_1\}$,
   - One for process 2, using the alphabet $\Sigma_2 = \{\overline{c?m}, \overline{c!m}, cs_2\}$,
   - One for the channel c of size 1, that is initially empty, using the alphabet $\Sigma_c = \{c?m, c!m, \overline{c?m}, \overline{c!m}\}$.

b. Construct the asynchronous product $\mathcal{P}$ of the three automata obtained in (a). The alphabet of the automaton $\mathcal{P}$ should be $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_c$.

c. Consider the state of the asynchronous product $\mathcal{P}$ where both processes are in the critical section. Is this state reachable? Give a short justification based on automaton $\mathcal{P}$.

Exercise 8.2

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. *Hint: Given a star-free regular expression r, build sentence $\varphi_r \in \mathrm{FO}(\Sigma)$ s.t. $L(\varphi_r) = L(r)$. You may use (c) and (d).*

Exercise 8.3

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).

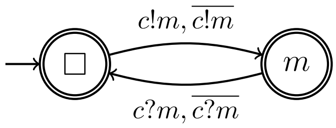
Prove that MSO(Σ) and PureMSO(Σ) have the same expressive power for sentences. That is, show that for every sentence ϕ of MSO(Σ) there is an equivalent sentence ψ of PureMSO(Σ), and vice versa.

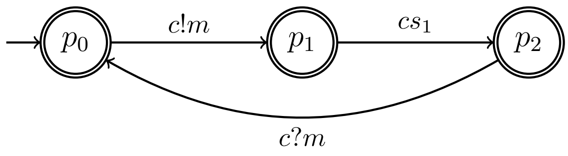
Exercise 8.4

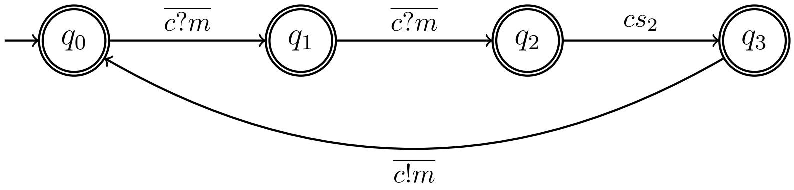
Let $r \geq 0$ and $n \geq 1$. Give a Presburger formula φ such that $\mathcal{J} \models \varphi$ iff $\mathcal{J}(x) > \mathcal{J}(y)$ and $\mathcal{J}(x) - \mathcal{J}(y) \equiv r \pmod{n}$. Give an automaton that accepts the solutions of φ for $r = 1$ and $n = 2$.

Solution 8.1

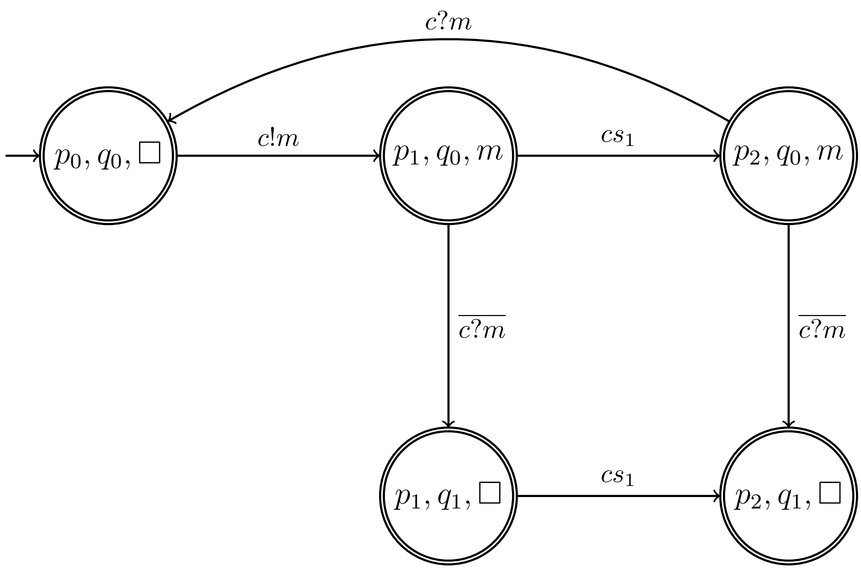
a. The automata for the channel, `process`(1) and `process`(2) are respectively:







b. The asynchronous product $\mathcal{P}$ is given below:



c. The state where both processes are in the critical section is not reachable, since $\mathcal{P}$ does not contain any of the states $(p_1, q_2, \square)$ and (p_1, q_2, m).

Solution 8.2

(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

$$(\neg \exists x \; \text{first}(x)) \; \vee$$

$$\big((\exists x \; \text{first}(x) \wedge Q_a(x)) \wedge (\exists y \; \text{last}(y) \wedge Q_b(y)) \; \wedge$$

$$(\forall x \, \forall y \; (Q_a(x) \wedge y = x + 1) \rightarrow Q_b(y)) \wedge \; (\forall x \, \forall y \; (Q_b(x) \wedge y = x + 1) \rightarrow Q_a(y)) \big) \, .$$

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

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:

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

(d) We use $false$ as syntactic sugar for $x < x$.

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 $\varphi = \psi_1 \lor \psi_2$ **then**
    return has-empty(ψ_1) $\lor$ has-empty(ψ_2)
  else if $\varphi = \exists\ \psi$ **then**
    return *false*

(e)

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

formula(r):
  if $r = \emptyset$ **then**
    return $\exists x$ false
  else if $r = \varepsilon$ **then**
    return $\forall x$ false
  else if $r = a$ for some $a \in \Sigma$ **then**
    return ($\exists x$ true) $\land$ ($\forall x$ first(x) $\land Q_a(x)$)
  else if $r = \overline{s}$ **then**
    return $\neg$formula(s)
  else if $r = s_1 + s_2$ **then**
    return formula(s_1) $\lor$ formula(s_2)
  else if $r = s_1 \cdot s_2$ **then**
    $\varphi_1 \leftarrow$ formula(s_1)
    $\varphi_2 \leftarrow$ formula(s_2)
    return ($\varphi_1 \land$ has-empty(φ_2)) $\lor$
        (has-empty(φ_1) $\land \varphi_2$) $\lor$
        ($\exists x, y, y', z$ first(x) $\land\ y' = y + 1 \land$ last(z) $\land\ \varphi_1^+(x, y) \land \varphi_2^+(y', z)$)

where the $\varphi_i^+(x, y)$ can be computed with an algorithm induced from (c).

Solution 8.3

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

$$\text{sing}(X) := \neg\text{empty}(X) \wedge \forall Y\,(Y \subseteq X \wedge \neg\text{empty}(Y)) \rightarrow X = Y.$$

Let ϕ be a sentence of MSO(Σ). 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 appropiately). Let ψ be the sentence of PureMSO(Σ) obtained by replacing every subformula of ϕ of the form

$$\begin{array}{lll} \exists x\; \psi' & \text{by} & \exists X\,(\text{sing}(X) \wedge \psi'[X/x]) \\ & & \text{where } \psi'[X/x] \text{ is the result of substituting } X \text{ for } x \text{ in } \psi' \\ Q_a(x) & \text{by} & X \subseteq Q_a \\ x < y & \text{by} & X < Y \\ x \in Y & \text{by} & X \subseteq Y \end{array}$$

Clearly, ϕ and ψ are equivalent.

Solution 8.4

Recall that since n is a constant, we can multiply a variable by n via iterated addition. The formula is as follows:

$$\varphi(x,y) := (x > y) \wedge \exists a\,\exists b \bigvee_{0 \leq r' < n} [(x = y + n \cdot a + r') \wedge (r = n \cdot b + r')].$$

Note that the right conjunct can be replaced with $r' = r$ if $r < n$.

Let $k \in \mathbb{N}$ and $x, y \in \Sigma^k$, 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 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: