Automata and Formal Languages — Homework 10

Due 09.01.2018

Exercise 10.1

It is late in Munich and you are craving for nuggets. Since the U-bahn is stuck at Sendliger Tor, you have no idea how hungry you will be when reaching the restaurant. Since nuggets are only sold in boxes of 6, 9 and 20, you wonder if it will be possible to buy exactly the amount of nuggets you will be craving for when arriving at the restaurant. You also wonder whether it is always possible to buy the exact amount of nuggets if one is hungry enough. Luckily, you can answer these questions since you are quite knowledgeable about Presburger arithmetic and automata theory.

For every finite set $S \subseteq \mathbb{N}$, let us say that $n \in \mathbb{N}$ is an *S-number* if n can be obtained as a linear combination of elements of S. For example, if $S = \{6, 9, 20\}$, then 67 is an S-number since $67 = 3 \cdot 6 + 1 \cdot 9 + 2 \cdot 20$, but 25 is not. For some sets S, there are only finitely many numbers which are not S-numbers. When this is the case, we say that the largest number which is not an S-number is the *Frobenius number* of S. For example, 7 is the Frobenius number of $\{3, 5\}$, and $S = \{2, 4\}$ has no Frobenius number.

To answer your questions, it suffices to come up with algorithms for Frobenius numbers and to instantiate them with $S = \{6, 9, 20\}$.

(a) Give an algorithm that decides, on input $n \in \mathbb{N}$ and a finite subset $S \subseteq \mathbb{N}$, whether n is an S-number.

(b) Give an algorithm that decides, on input $S \subseteq \mathbb{N}$ (finite), whether S has a Frobenius number. [Hint:]

(c) Give an algorithm that computes, on input $S \subseteq \mathbb{N}$ (finite), the Frobenius number of S (assuming it exists).

(d) ★ Show that $S = \{6, 9, 20\}$ has a Frobenius number, and identify this number.

Exercise 10.2

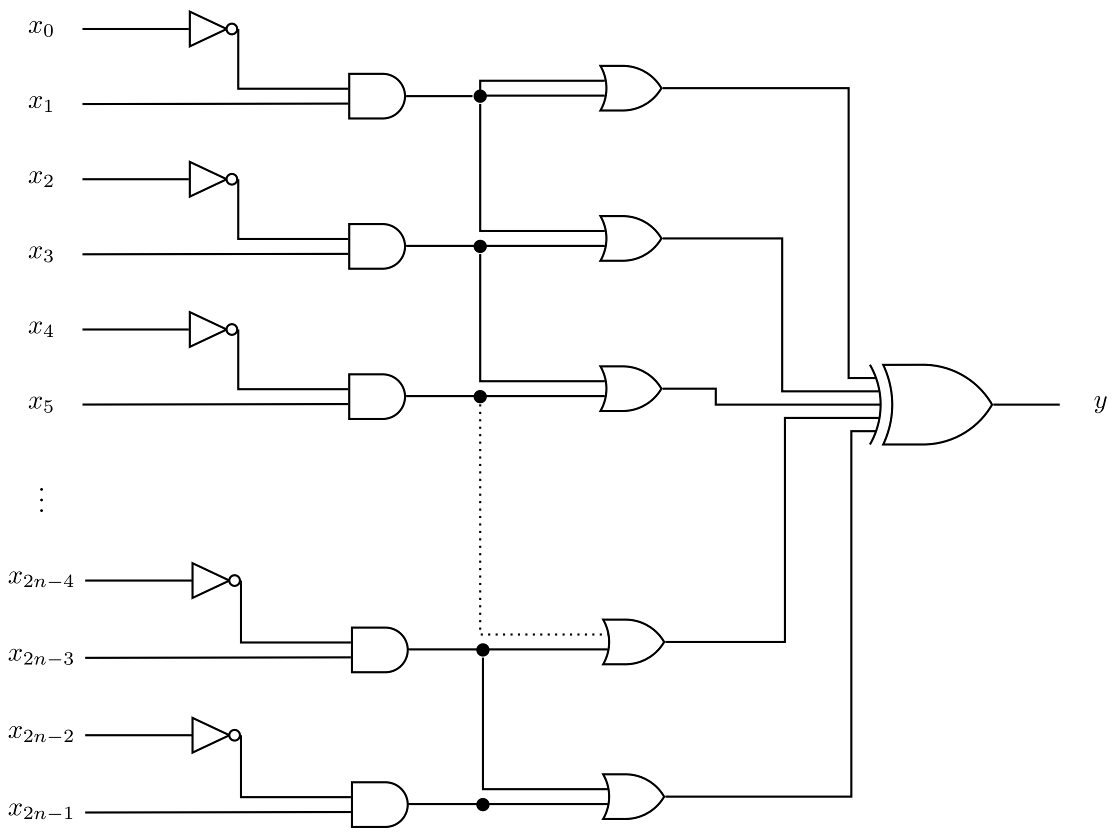
Let $\Sigma = \{a, b\}$. Give an MSO(Σ) sentence for the following languages:

(a) The set of words with an a at every odd position.

(b) The set of words with an even number of occurences of a's.

(c) The set of words of odd length with an even number of occurences of a's.

Exercise 10.3
Let $n \in \mathbb{N}_{>0}$. Consider the following circuit C_n:



In case you are not familiar with the above symbols, the first, second, third and fourth layers of gates are respectively NOT, AND, OR and XOR gates. Note that $\mathrm{XOR}(z_1, z_2, \ldots, z_n) = z_1 \oplus z_2 \oplus \cdots \oplus z_n$. For every $n \in \mathbb{N}_{>0}$, let $X_n = \{x \in \{0,1\}^n : C_n$ outputs 1 on input $x\}$. Let $X = \bigcup_{n \in \mathbb{N}_{>0}} X_n$. In other words, X is the set of assignments that satisfy some circuit of the infinite family of circuits $\{C_n : n \in \mathbb{N}_{>0}\}$.

(a) Give an MSO sentence ϕ such that $L(\phi) = X$.

(b) ★ Use `MONA` to obtain an automaton accepting X.

To solve (a), you should consider constructing a formula for each layer of gates. For example, the following predicate asserts that Out contains precisely the indices of wires set to 1 past the first layer of NOT gates:

$$\begin{aligned} \mathrm{layer1}(Out) = \forall p\ (\mathrm{even}(p) &\rightarrow ((p \in Out) \leftrightarrow Q_0(p))) \wedge \\ (\mathrm{odd}(p) &\rightarrow ((p \in Out) \leftrightarrow Q_1(p))) \end{aligned}$$

For this question, you may assume that positions begin at 0 instead of 1 in MSO. This is the case in `MONA`.

† Question 10.3 is adapted from an example of Henriksen et al. *Mona: Monadic second-order logic in practice.* Proc. International Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 1995.