Apr 14, 2021

CS 228 Spring 2021 Assignment

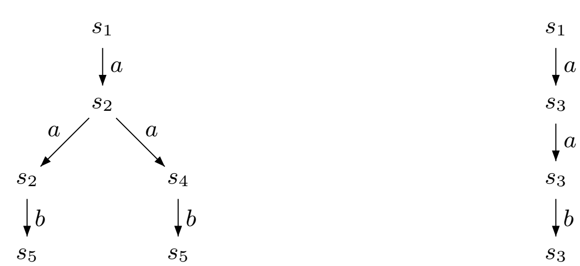
---

1. Consider languages $L_1 = b(a^*bb)^*$ and $L_2 = b(a^+bb)^*$. Can you give a FO formula φ for either of these languages? When you can, give the formula. When you cannot, give an intuitive reason why.

2. Let us define a new kind of automaton inspired from logic, called $\forall\exists$ automaton as follows: A $\forall\exists$ automaton is a finite state automaton $(Q, \Sigma, \Delta, S, F)$ where the set of states Q is partitioned into two sets $Q_\forall$ and $Q_\exists$. Let there be n states in Q. The transitions coming out of a $Q_\exists$ state are called "or" transitions, and the transitions coming out of a $Q_\forall$ state are called "and" transitions.

   An "or" transition has the form $\Delta(q, a) = q_{i_1} \vee \ldots \vee q_{i_j}$, $q \in Q_\exists$, while, an "and" transition has the form $\Delta(q, a) = q_{j_1} \wedge \ldots \vee q_{j_l}$, $q \in Q_\forall$.

   For example, let $Q = \{s_1, s_2, s_3, s_4, s_5\}$, $S = \{s_1\}$, $F = \{s_5\}$, with transitions $\Delta(s_1, a) = s_2 \vee s_3$, $\Delta(s_2, a) = s_2 \wedge s_4$, $\Delta(s_2, b) = s_5$, $\Delta(s_4, a) = s_4$, $\Delta(s_4, b) = s_5$, $\Delta(s_3, a) = s_3$, $\Delta(s_3, b) = s_3$. The following are two run trees of the word aab:



   Note that each time an "and" transition $\delta(q, a) = q_{i_1} \wedge \ldots \wedge q_{i_j}$ is used, we spawn j threads, and maintain all the states $q_{i_1}, \ldots, q_{i_j}$. All these threads then evolve in parallel. A word w is accepted by a run tree if a final state is encountered in all the threads when you finish reading w. An "or" transition is the usual non-deterministic choice you have, you can pick any one of the choices. In the above example, the first run tree is accepting, while the second is not. A word is accepted if it has atleast one accepting run tree.

   Compare the expressiveness of $\forall\exists$ automata and NFA. That is, given a $\forall\exists$ automaton A, does there exist a NFA B such that $L(A) = L(B)$? Conversely, given a NFA A, does there exist a $\forall\exists$ automaton B such that $L(A) = L(B)$?

3. A certain logician, Prof.Calculus is interested in specifying properties of systems, by observing them over intervals of time $[b, e]$, where $b, e \in \mathbb{N}$. He assumed that his systems are observable at all discrete points of time $0, 1, 2, \ldots, j$, where $j \in \mathbb{N}$. Let $Var = \{p_1, p_2, \ldots, p_n\}$ be the underlying set of propositional variables that are needed to model the system under consideration. Each variable p_i can take a value 0 or 1 at all the observable points. For example, if Prof.Calculus wants to observe a system over an interval $[0, 3]$, with $Var = \{p_1, p_2\}$ he must know the values of variables in Var at the time points 0,1,2,3. This gives a behaviour σ of the system in the interval [0,3]. A possible behaviour σ could be this:

$$p_1 : 0\ 1\ 0\ 1$$
$$p_2 : 1\ 0\ 1\ 1$$

σ says that p_2 is true at the first point, both p_1, p_2 are true at the last point, and at all points other than the last, p_1, p_2 toggle. Let $|\sigma|$ denote the length of behaviour σ, and let $\sigma(i)$ denote its behaviour at point i, $0 \leq i \leq |\sigma| - 1$, and let $dom(\sigma) = \{0, 1, \ldots, |\sigma| - 1\}$. In the above case, $|\sigma| = 4, \sigma(1) = (p_1 = 1, p_2 = 0)$, and $dom(\sigma) = \{0, 1, 2, 3\}$.
Let small letters $p, q, \ldots$ denote propositional variables, and $P, Q, \ldots$ denote boolean combinations of propositional variables. Prof.Calculus designed a logic (lets call it $\mathcal{CL}$ as short form for Calculus Logic) which has the following syntax: For $c \in \mathbb{N}$ and $\sim \in \{>, =\}$,

$$\varphi : \lceil P \rceil^0 \mid \lceil\lceil P \rceil \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi.\varphi \mid len \sim c \mid \int P \sim c$$

For a chosen behaviour σ, an interval $[b, e] \in dom(\sigma) \times dom(\sigma)$, with $b \leq e$, we inductively define the satisfaction of a formula φ in the logic of Calculus denoted $\sigma, [b, e] \models \varphi$ as follows:

$\sigma, [i, i] \models p$ iff $\sigma(i)(p) = 1$
$\sigma, [b, e] \models \lceil P \rceil^0$ iff $b = e$ and $\sigma, b \models P$
$\sigma, [b, e] \models \lceil\lceil P \rceil$ iff $b < e$ and $\sigma, i \models P$ for all $b \leq i < e$
$\sigma, [b, e] \models \neg\varphi$ iff $\sigma, [b, e] \not\models \varphi$
$\sigma, [b, e] \models \varphi_1 \wedge \varphi_2$ iff $\sigma, [b, e] \models \varphi_1$ and $\sigma, [b, e] \models \varphi_2$
$\sigma, [b, e] \models \varphi_1.\varphi_2$ iff for some $m, b \leq m \leq e, \sigma, [b, m] \models \varphi_1$ and $\sigma, [m, e] \models \varphi_2$

$len \sim c$ and $\int P \sim c$ are called measurements. len stands for the length of the interval chosen. $\int P$ is the number of times P holds good in a chosen interval $[b, e]$, and this number is compared with c in evaluation of $\int P \sim c$. For a given behaviour σ over $[b, e]$, define

$$\int_b^e P = \sum_{i=b}^{e-1} x_i$$

where $x_i = 1$ if $\sigma, i \models P$, and $x_i = 0$ otherwise.

$$\sigma, [b, e] \models len \sim c \text{ iff } e - b \sim c$$
$$\sigma, [b, e] \models \int P \sim c \text{ iff } \int_b^e P \sim c$$

In our example, we thus have $\sigma, [0, 3] \models (len = 3)$, $\sigma, [0, 3] \models (\int p_2 = 2)$. A formula $\varphi \in \mathcal{CL}$ is satisfiable iff one can find a behaviour σ of some finite length such that $\sigma, [0, |\sigma| - 1] \models \varphi$.

(a) Write the property "the interval has even length" in $\mathcal{CL}$. Justify.
(b) Construct a finite state automaton (DFA/NFA) that accepts all behaviours σ which satisfy the formula $(\lceil p \rceil^0 . \lceil\lceil \neg q \rceil . \lceil q \rceil^0) \rightarrow (\lceil\lceil r \rceil . \lceil r \rceil^0)$. Explain your construction.