Roll Number :

CS 228 Spring 2023 End-semester Examination

17-04-2023

- ***If you need to make any assumptions, state them clearly.***
- ***Write your answers in the spaces provided in the question booklet.***
- ***If needed, you may cite results/proofs covered in class without reproducing them.***
- ***Penalty for Copying: FR grade***

1. **Good old propositional logic** *[10 marks]*

   (a) Convert the following argument into a propositional statement, i.e., $\Sigma \vdash F$. If the laws are good and their enforcement is strict, then crime will diminish. If strict enforcement of laws will make crime diminish, then our problem is a practical one. The laws are good. Therefore our problem is a practical one. (Hint: needed propositional variables G, S, D, P)

(b) Write a formal proof proving the statement in the previous problem.

2. **LTL and cousins** *[20 marks]*
   Lets begin with LTL and first order logic whose models are infinite words.
   (a) For each LTL formula φ over a set of propositional variables AP, show that there exists a first order logic sentence ψ such that $L(\varphi) = L(\psi)$.

   Hint : Use the semantics of LTL, and use structural induction to come up with a translation T such that each LTL formula φ can be translated into an FO formula $\psi = T(\varphi)$. For example, for the LTL formula a, $T(a) = \exists x(Q_a(x) \wedge first(x))$, where you expand first appropriately.

(b) Part (a) shows that LTL $\subseteq$ FO. Now consider the following variant of LTL, called quantified LTL (QLTL) defined as follows.

Syntax: any LTL φ over $AP \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists p.\varphi$, where $p \in AP$.

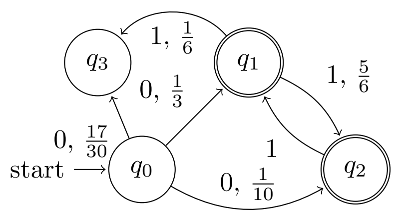
Semantics: Given an infinite word $w \in \Sigma^\omega$, $\Sigma = 2^{AP}$, $w, i \models \exists q.\varphi$ iff there exists $w' \in \Sigma^\omega$ with $w'[j] \cap (AP \backslash \{q\}) = w[j] \cap (AP \backslash \{q\})$ for all indices j, such that $w', i \models \varphi$. $w[j]$ denotes the jth position of w.

(1) Show that the language $L = (\emptyset\emptyset)^*\{p\}^\omega$ is QLTL-definable.

(2) For every Büchi automaton $\mathcal{A}$ over $\Sigma = 2^{AP}$, there exists a QLTL formula φ such that $L(\varphi) = L(\mathcal{A})$.

3. **Probably Regular** *[10 marks]*
A probabilistic automaton A is given by $(Q, \Sigma = \{0,1\}, (T_0, T_1), q_0, F)$ where Q is a set of states, the input alphabet is $\{0,1\}$, q_0 is the initial state, F is a set of final states and T_0, T_1 are two $n \times n$ matrices where $n = |Q|$. The matrices have the following properties:
   - each row and each column is labeled by the set of states $q_0, q_1, \cdots, q_{n-1}$
   - All the entries of the matrices are in the interval $[0,1]$, i.e. $\forall i, j \in \{0, 1, \cdots, n-1\}$, $0 \leq T_0(q_i, q_j) \leq 1$ and $0 \leq T_1(q_i, q_j) \leq 1$.
   - Each row sums up to 1, i.e. $\forall i \in \{0, 1, \cdots, n-1\}$, $\sum_{j=0}^{n-1} T_0(q_i, q_j) = 1$ and $\sum_{j=0}^{n-1} T_1(q_i, q_j) = 1$.

The idea is that the transitions of the automaton are probabilistic, i.e. when the automaton is in the state q_i and sees a letter $b \in \{0,1\}$ then it goes to state q_j with probability $T_b(q_i, q_j)$.



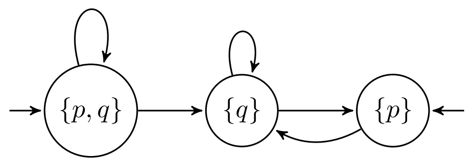
Let $w \in \{0,1\}^*$, $w = a_1 a_2 \cdots a_n$ and let $T_w = T_{a_1} \times T_{a_2} \cdots \times T_{a_n}$ where $\times$ is matrix multiplication. Let $Nice(w) = \sum_{f \in F} T_w(q_0, f)$, i.e. the total probability of reaching any of the final states starting from q_0 on word w. We say that A τ-accepts w iff $Nice(w) \geq \tau$. Let $L_\tau(A) = \{w \mid A\ \tau\text{-accepts } w\}$. As an example, for the automaton A above and $w = 01$, $Nice(01) = \frac{5}{18} + \frac{1}{10}$.
Prove that for any probabilistic automaton, $L_1(A)$ is regular.

4. **Alternation** *[15 marks]*
Recall alternating finite automata (AFA) done in the class. Let us now consider this on infinite words with the Büchi acceptance condition. An Alternating Büchi automaton (ABA) is a tuple $A = (Q, q_0, \Sigma, \delta, F)$ where $\delta(q, a)$ for each $q \in Q, a \in \Sigma$ is a CNF over Q. Assume wlg that no states are terminal, that is, on each symbol of Σ, there is at least one outgoing transition from each state.
An infinite word $w \in \Sigma^\omega$ is accepted by A iff there is at least one run tree T for w, such that all the branches of T visit states of F infinitely often. Formally, T is an accepting run tree iff for each infinite branch α of T, $Inf(\alpha) \cap F \neq \emptyset$.
   (a) Consider the ABA $A = (\{p, q, r\}, \{p\}, \{a, b\}, \delta, \{p, r\})$ such that
   $\delta(p, a) = p \wedge q, \delta(p, b) = p, \delta(q, a) = q, \delta(q, b) = r, \delta(r, a) = \delta(r, b) = r.$
   What is $L(A)$?

(b) For every ABA $A = (Q, q_0, \Sigma, \delta, F)$, show that there exists a NBA B such that $L(A) = L(B)$.

Hint: when you construct the NBA B, visiting the good states of B infinitely often must simulate the fact that there is a run tree T of A such that each branch of T sees states from F infinitely often. Recall the product construction we did in class for NBA, where we track visiting a good state in the first automaton, and then in the second automaton and keep alternating this. Recall we used a bit $\in \{1, 2\}$ to track, and used this in the acceptance condition. Here, you need to track the branches of T; any run tree T is finitely branching. Perhaps you cant use a bit here, since you don't know exactly how many children are there at a given level of T; however, you can use the fact that the children at any level is just a subset of Q.

5. **LTL model checking** *[20 marks]*
Consider the transition system TS given below.



(a) Give a persistence property P_{pers} such that $TS \not\models P_{pers}$. Write an algorithm that checks $TS \not\models P_{pers}$.

(b) Consider the LTL formula $\varphi = \neg \bigcirc (q \wedge \bigcirc \Box (p \wedge q))$. Following the steps done in class, check if $TS \models \varphi$. To do this,

- construct a NBA A for $\neg\varphi$,
- construct the product TS' of TS and A,
- write down a persistence property ψ and argue that $TS' \models \psi$ iff $TS \models \varphi$.

6. **Let's Count!** *[10 marks]*

(a) Consider the following Boolean logic formula over variables $x_1, ..., x_n$.

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \ldots \wedge (\neg x_{n-1} \vee x_n) \wedge (\neg x_n \vee x_1)$$

How many assignments to the variables satisfy the above formula?

(b) Consider the following Boolean logic formula over variables $x_1, ..., x_n$.

$$(\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3) \vee \ldots \vee (\neg x_{n-1} \wedge x_n) \vee (\neg x_n \wedge x_1)$$

How many assignments to the variables satisfy the above formula?

ROUGH WORK/ EXTRA SHEET

ROUGH WORK/ EXTRA SHEET