CS 228 Minor Autumn 2023 End-semester Examination

21-11-2023 Total Marks: 80

- ***If you need to make any assumptions, state them clearly.***
- ***If needed, you may cite results/proofs covered in class without reproducing them.***

1. *[2+2+5+6=15 marks]* Let $\Sigma = \{a, b, c\}$ be a finite set of propositions. For $k \geq 1, k \in \mathbb{N}$, consider the language L_k of ω-words $w \in [2^{\Sigma}]^{\omega}$ describing the specification

   "w has no subword where the proposition a appears k times but the proposition b does not appear"

   (a) Give a word $w \in L_k$

   $w = \{a, b\}^{\omega}$

   (b) Give a word $w \notin L_k$

   $\{a\}^{\omega}$

   (c) Give a family of LTL formulae φ_k of depth $k \in \mathbb{N}$ such that $L(\varphi_k) = L_k$. Give an intuitive argument why $L(\varphi_k) = L_k$.

   Consider the complement of the property : that is, there is a subword where a appears k times but b does not appear. Call this property M_k. We define LTL formulae inductively.

   i. Let $\gamma_0 = a \wedge \neg b, \gamma_k = (a \wedge \neg b \wedge \bigcirc((\neg a \wedge \neg b) \mathsf{U} \gamma_{k-1}))$
   ii. Let $\varphi_k = \Diamond \gamma_k$.
   iii. Then $\neg \varphi_k$ captures L_{k+1} (or φ_k captures M_{k+1}).
   iv. For instance
      - when $k = 0$, $\varphi_0 = \Diamond(a \wedge \neg b)$ captures words where there is a subword having an a but no b (so M_1).
      - If $k = 1$, $\varphi_1 = \Diamond \gamma_1 = \Diamond[(a \wedge \neg b) \wedge \bigcirc[(\neg a \wedge \neg b) \mathsf{U} (a \wedge \neg b)]]$ captures words having a subword with two occurrences of a but no b (so M_2).
      - If $k = 2$, $\varphi_2 = \Diamond \gamma_2 = \Diamond[(a \wedge \neg b) \wedge \bigcirc[(\neg a \wedge \neg b) \mathsf{U} \gamma_1]]$ which is

      $$\Diamond \gamma_2 = \Diamond[(a \wedge \neg b) \wedge \bigcirc[(\neg a \wedge \neg b) \mathsf{U} \{(a \wedge \neg b) \wedge \bigcirc((\neg a \wedge \neg b) \mathsf{U} (a \wedge \neg b)\}]]$$

      captures words having a subword with three occurrences of a but no b (so M_3), and so on.

   (d) For any $k \in \mathbb{N}$, give an argument that φ_k has to be of depth k in order to capture L_k as follows.
   Construct two ω-words w_1, w_2 such that
   (i) $w_1 \in L_k, w_2 \notin L_k$, and (ii) either $w_1, w_2 \models \varphi_{k-1}$ or $w_1, w_2 \not\models \varphi_{k-1}$.

Justify your choice of words and explain.

Define

$$v_k = \{a\}\{c\}w_k, \quad w_k = [\{a\}\{c\}]^k[\{b\}\{c\}[\{a\}\{c\}]^k]^\omega$$

So,

$$v_0 = \{a\}\{c\}w_0, w_0 = [\{b\}\{c\}]^\omega,$$

$$v_1 = \{a\}\{c\}w_1, w_1 = \{a\}\{c\}[\{b\}\{c\}\{a\}\{c\}]^\omega,$$

$$v_2 = \{a\}\{c\}w_2, w_2 = [\{a\}\{c\}]^2[\{b\}\{c\}(\{a\}\{c\})^2]^\omega$$

$$v_3 = \{a\}\{c\}w_3, w_2 = [\{a\}\{c\}]^3[\{b\}\{c\}(\{a\}\{c\})^3]^\omega$$

and so on.

Then $v_k \in M_{k+1}, w_k \notin M_{k+1}$ (equivalently, $v_k \notin L_{k+1}, w_k \in L_{k+1}$).

Argue that $v_k, w_k \vDash \varphi_{k-1}$ for all $k \geq 1$.

So, $v_2, w_2 \vDash \varphi_1 = \Diamond\gamma_1 = \Diamond[(a \wedge \neg b) \wedge \bigcirc[(\neg a \wedge \neg b)\mathsf{U}(a \wedge \neg b)]]$

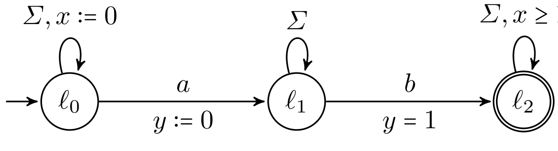
$v_3, w_3 \vDash \varphi_2 = \Diamond[(a \wedge \neg b) \wedge \bigcirc[(\neg a \wedge \neg b)\mathsf{U}\{(a \wedge \neg b) \wedge \bigcirc(\ (\neg a \wedge \neg b)\mathsf{U}\ (a \wedge \neg b)\}]]$

and so on.

2. *[2+2+2+2+5+5+5+5+5+2=35 marks]* Given a finite alphabet Σ, a finite timed word ρ over Σ is a finite sequence of pairs $(\sigma, \tau) \in (\Sigma \times \mathbb{R}_{\geq 0})^*$: $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ where $\tau_i \leq \tau_j$ for all $1 \leq i \leq j \leq n$. Here, $\mathbb{R}_{\geq 0}$ denotes the non-negative reals. The pair (σ_j, t_j) represents that action σ_j happened at time $t_j \in \mathbb{R}_{\geq 0}$. A set of timed words is called a timed language. Timed languages are generated by timed automata, which are an extension of finite state automata with real valued variables called clocks.

Timed Automata. Given below is a timed automaton A with 3 locations ℓ_0, ℓ_1, ℓ_2 where ℓ_0 is initial and ℓ_2 is an accepting location. x, y are clocks. Let the alphabet $\Sigma = \{a, b, c\}$. To begin, we are at the initial location ℓ_0 with clocks x, y initialized to 0. We can stay in ℓ_0 for some real time $t \geq 0$, after which a transition can be taken. Whenever you stay at a location and elapse time t, all your clocks progress by that time t.

If the loop at ℓ_0 is taken on some symbol from Σ, the clock x is reset to 0 (the meaning of $x := 0$ on the loop means reset x to value 0). After the transition, we are back in location ℓ_0, with clock values $x = 0, y = t$. Likewise, if we transit from ℓ_0 to ℓ_1, the clock y is reset to 0. The loop on ℓ_1 does not reset either clock. The transition from ℓ_1 to ℓ_2 is possible only if the value of y is 1. Lastly, the loop on ℓ_2 can be taken only when $x \geq 1$.



Here is a sample run :

1. Spend 0.3 time units at ℓ_0. This gives $x = y = 0.3$.

2. Take an a transition on the loop at ℓ_0. Then $x = 0, y = 0.3$. Then we have read a at time 0.3.
3. Next, spend 0.8 time units in ℓ_0. Then $x = 0.8, y = 1.1$.
4. Take the transition from ℓ_0 to ℓ_1 on a. Then $x = 0.8, y = 0$, and we have read a at time 1.1.
5. Next, stay in ℓ_1 for .5 time units. Then $x = 1.3, y = 0.5$ and then take the loop on symbol c. Now we have spent a total of 1.6 units of time since we started from ℓ_0. Then we have read c at time 1.6.
6. Next, stay in ℓ_1 for 0 time units and then take the loop on symbol b. Then we have read b at time 1.6.
7. Next, stay in ℓ_1 for 0.4 time units. The transition to ℓ_2 is not yet enabled since $y = 0.9 < 1$. You can take the loop at ℓ_1 many more times till y grows to 1. In this example, let's simply spend 0.1 more time at ℓ_1 obtaining $x = 1.8, y = 1$. The guard $y = 1$ is true. Take the transition to ℓ_2 on b. Then we have read b at time 2.1.
8. At ℓ_2, stay 1.1 time units and take the loop on a. Then we have $x = 2.9, y = 2.1$ and we have read a at time 3.2.
9 By now, we have read the timed word $(a, 0.3)(a, 1.1)(c, 1.6)(b, 1.6)(b, 2.1)(a, 3.2)$ which is accepted by A.

A accepts a timed language $L(A)$ consisting of finite timed words which have an accepting run from ℓ_0 to ℓ_2 : that is, all timed words which contain an occurrence of event a followed by a b after a delay of one time unit.

Metric Temporal Logic(MTL). Now, we look at an extension of logic LTL whose models are finite timed words. This logic is called Metric Temporal Logic (MTL). MTL is a real-time extension of LTL where the modalities ("until" U and "since" S) are guarded with intervals. Formulae of MTL are built from Σ using Boolean connectives and time constrained versions U_I and S_I of the standard U, S modalities, where I is an open, half-open or closed time interval whose end points are in $\mathbb{N} \cup \{0, \infty\}$. Given a set of propositional variables Σ, the syntax of MTL is as follows.

$\varphi ::= a \,|\top\, |\varphi \wedge \varphi \mid \neg\varphi \mid \varphi \mathsf{U}_I \varphi \mid \varphi \mathsf{S}_I \varphi$, where $a \in \Sigma$ and I is an interval.

For a timed word $\rho = (\sigma_1, \tau_1)(\sigma_2, \tau_2) \dots (\sigma_n, \tau_n)$, a position $1 \le i \le n$, an MTL formula φ, the satisfaction of φ at a position i of ρ, denoted $\rho, i \models \varphi$, is defined below.

- $\rho, i \models a$ iff $\sigma_i = a$,
- $\rho, i \models \varphi_1 \wedge \varphi_2$ iff $\rho, i \models \varphi_1$ and $\rho, i \models \varphi_2$,
- $\rho, i \models \neg\varphi$ iff $\rho, i \not\models \varphi$,
- $\rho, i \models \varphi_1 \mathsf{U}_I \varphi_2$ iff $\exists j > i$, $\rho, j \models \varphi_2, \tau_j - \tau_i \in I$, and $\rho, k \models \varphi_1$ forall $i < k < j$.
- $\rho, i \models \varphi_1 \mathsf{S}_I \varphi_2$ iff $\exists j < i$, $\rho, j \models \varphi_2, \tau_i - \tau_j \in I$, and $\rho, k \models \varphi_1$ forall $j < k < i$.

The language of an MTL formula φ is defined as $L(\varphi) = \{\rho | \rho, 1 \models \varphi\}$.

As we did in LTL, we can define the derived operators

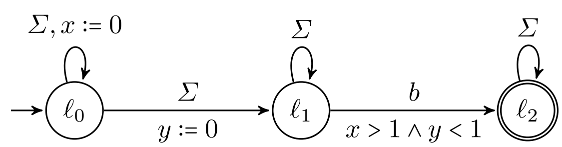
$$\Diamond_I \varphi = \mathit{true}\ \mathsf{U}_I \varphi, \quad \text{and} \quad \Box_I\, \varphi = \neg(\Diamond_I \neg\varphi)$$

Likewise we can define

$$\diamondminus_I \varphi = \mathit{true}\ \mathsf{S}_I \varphi \quad \text{and} \quad \boxminus_I\, \varphi = \neg(\diamondminus_I \neg\varphi)$$

As an example, consider the formula $\varphi_1 = a\mathsf{U}_{(0,1]} b$ and $\varphi_2 = \Diamond_{[0,\infty)}(a\mathsf{S}_{[1,1]} b)$ and the timed word $w = (a,0.2)(a,0.22)(a,0.7)(b,1)(a,1.1)(c,1.2)(a,2)$. Then $w,1 \models \varphi_1$ but $w,1 \not\models \varphi_2$, in fact, $w,i \not\models a\mathsf{S}_{[1,1]} b$ for $i = 5,7$. However, $w,7 \models (a \vee c)\mathsf{S}_{[1,1]} b$.

With this background on timed automata and MTL, consider the timed automaton B below with $\Sigma = \{a,b,c\}$, and answer the following questions. In each case, provide justification for your answers.



(a) Consider the timed word $w_1 = (a,0)(b,0)(c,.9)(c,1.2)(b,1.9)(a,\pi)$. Is $w_1 \in L(B)$?

No.

(b) Consider the timed word $w_2 = (a,0.3)(b,0.6)(c,.9)(c,1.4)(b,1.5)(a,\pi)$. Is $w_2 \in L(B)$?

Yes. Take the a at the loop at ℓ_0 and b from ℓ_0 to ℓ_1.

(c) Give a timed word in $L(B)$ (other than w_1, w_2).

$w_2 = (a,0.3)(b,0.6)(c,.92)(c,1.4)(b,1.5)(a,\pi)$

(d) Give a timed word not in $L(B)$ (other than w_1, w_2). $(a,0)(b,0.001)(c,.9)(c,1.2)(b,1.9)(a,\pi)$

(e) What is $L(B)$?

All timed words where there is a b at some time t and there is no event at time $t-1$, $t > 1$, and some event in $(t-1,t)$.

(f) Consider the MTL formula $\varphi = \Diamond_{[0,\infty)} \diamondminus_{[1,1]} b$. Is $L(B) \cap L(\varphi) = \varnothing$?

No. You can find a word in $L(B)$ which has some event at a time t and b at time $t-1$ (note that once you reach ℓ_2 at time $\leq t-1$, this can be done as there are no time constraints).

(g) Consider the MTL formula $\varphi = \Diamond_{[0,\infty)} \boxminus_{[1,1]} \neg b$. Is $L(B) \cap L(\varphi) = \varnothing$?

No. $(a, 0.3)(b, 0.6)(c, .92)(c, 1.4)(b, 1.5)(a, 2)(a, 2.8)$ is in $L(B)$ and also in $L(\varphi)$ (from the last a, in the interval of length 1 in the past, there is no b).

(h) Consider the MTL formula $\varphi = \Diamond_{[0,\infty)} \neg(\Diamond_{[1,1]} b)$. What is the relationship between $L(B)$ and $L(\varphi)$?

One can find a word in both. Simply create a word in $L(B)$ which has some event at some time t, and there is no b at $t + 1$.

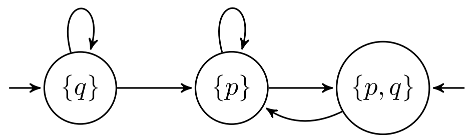
(i) Consider the MTL formula $\varphi = \Diamond_{[0,\infty)} [b \wedge (b \Rightarrow \neg \diamondminus_{[1,1]} \Sigma)]$. What is the relationship between $L(B)$ and $L(\varphi)$?

One can find a word in both. Simply create a word in $L(B)$ which has a b at time $t + 1$ for some $t \geq 0$ and there is no event at time t.

(j) Express $\bigcirc_{(1,2)} b$ in MTL.

$false \ \mathsf{U}_{(1,2)} b$.

3. *[4+5+1=10 marks]*
Consider the transition system TS given below.



Let $\varphi = \Box(p \to \bigcirc(\Diamond \Box q))$.

(a) draw an NBA $A_{\neg\varphi}$ for $\neg\varphi$,

(b) construct $TS' = TS \otimes A_{\neg\varphi}$,

(c) write an appropriate persistence property P_{pers} to be checked on TS'.

Finally, your answer for TS satisfying (or not) φ must be linked to TS' satisfying (or not) P_{pers}. Using this conclude whether or not $TS \models \varphi$.

4. *[10 marks]* Mr. X owns the stock of 3 different companies, A, B and C. Each day, he checks the relative values of his stocks and orders them from the most to the least valuable (assume that the values of two stocks can never be the same). Mr. X decided to sell the stock of a company if it ever goes down for two days in a row. For example, if the stock record in 3 consecutive days is CBA, ACB, ABC, then Mr. X will sell C. Let Σ be the set of all permutations of A, B, C. Prove that the language of words over Σ that describes those stock records that do not lead to the sale of any stock owned by Mr. X is regular.

Regularity is easy, since Mr.X's decision whether to sell a stock depends only on the current stock and only on the orders of the previous two days.
For any word w the information about a word v needed to determine if $vw \in L$ is
whether $v \in L$, and if yes, then

– the highest stock value on the last day, and
– the stock that dropped from the first to the second place on the last day, if any.

This gives an automaton with 11 states.

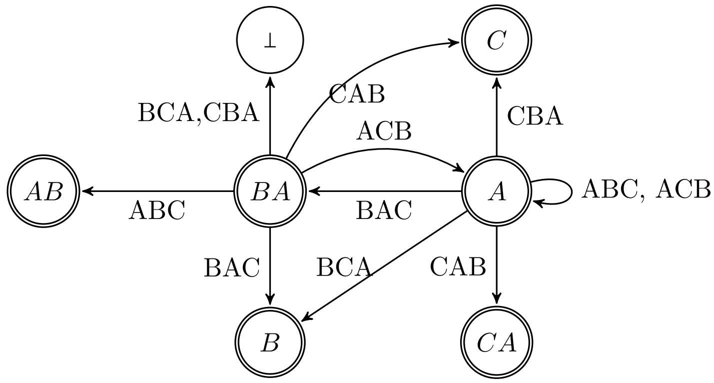
(a) an initial state 0,

(b) 3 states A, B, C indicating the highest priced stock and no stock dropped from the first to second place last day,

(c) 6 states $AB, AC, \ldots$ indicating the highest priced stock was A and B dropped from first to second place the last day

(d) a reject state $\perp$.

The transition structure is intuitive. Here is a part of it



5. *[5 marks]*
A propositional logic formula is said to be in CNF if it can be written as a conjunction of disjunctions. Each disjunct of literals is called a clause. For example, $(x \vee y \vee \neg z) \wedge (\neg x \vee \neg g \vee a \vee b)$ is in CNF and has 2 clauses.
A propositional logic formula is said to be in k-CNF form if each clause has exactly k-literals (including repetitions). For example, $(x \vee \neg y \vee z \vee x) \wedge (\neg y \vee \neg z \vee x \vee \neg d)$ is in 4-CNF.
Show that any CNF formula can be rewritten in 3-CNF preserving satisfiability. That is, if φ is in CNF, we can obtain a formula φ' in 3-CNF such that φ' evaluates to true under all assignments which satisfy φ. Conversely, from any assignment β which satisfies φ', we can find an assignment γ which satisfies φ.

6. *[5 marks]*
Give FO formulae for each of the following specifications interpreted as infinite words. Your signature allows the predicates $Q_a, <$ and S.
   - Infinitely Often a
   - Eventually forever a
   - Each a is followed by at least two b's
   - There is exactly one a in the word
   - After some long prefix, the word has no a's