Cyber Physical Systems - Discrete Models
Exercise Sheet 14 Solution

Alper Ari
aa508@uni-freiburg.edu

Onur Sahin
os141@uni-freiburg.de

February 7, 2023

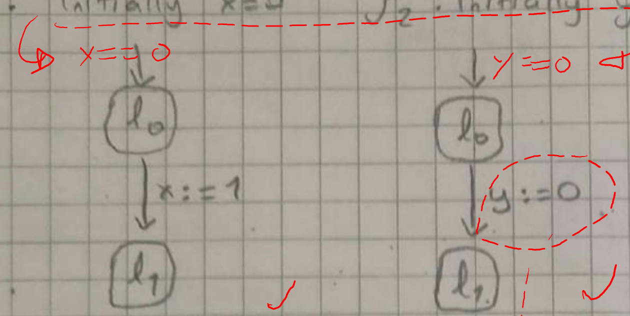
1	2	3	4	Σ
4.5/8	8/8	7/7	6/6	25.5

Exercise 1: Transition Systems, Program Graphs, Interleaving 4.5/8

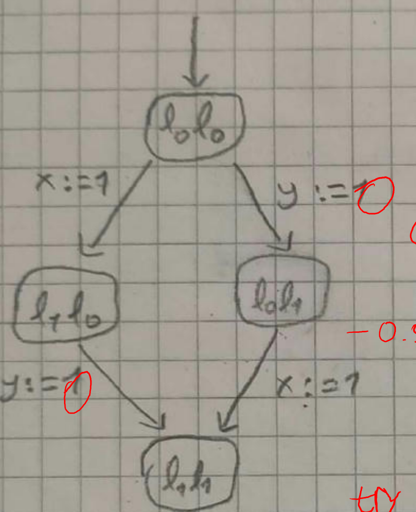
1)

a) P_1: Initially $x=0$ P_2: Initially $y=0$

Then you should have initial conditions $x==0$ and $y==0$ in your PG −1

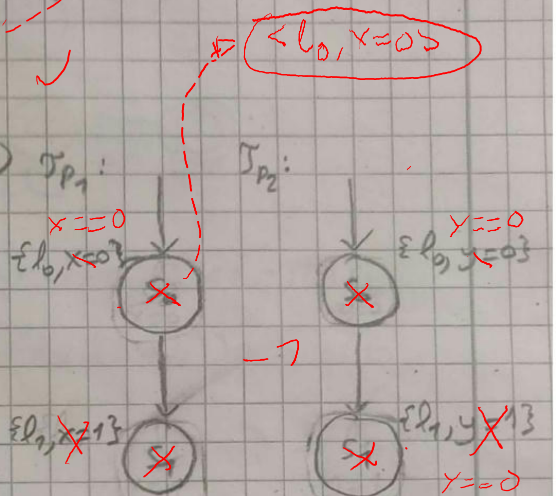


b) $P_1 \,|||\, P_2$: Initially $x=0$ and $y=0$



−0.5

c) T_{P_1}: T_{P_2}:



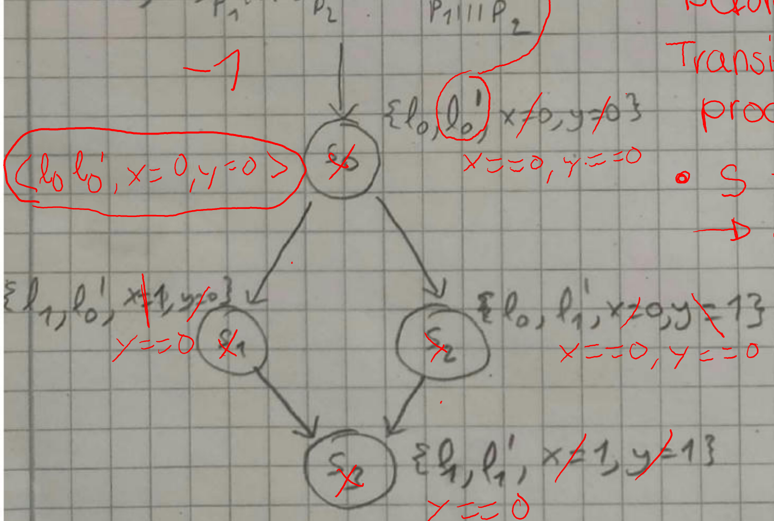
−1

AP = $\{l_0, l_1, x=0, x=1\}$ AP = $\{l_0, l_1, y=0, y=1\}$

try to be consistent with the naming. where does the ' come from?

d) $T_{P_1} \,|||\, T_{P_2} = T_{P_1|||P_2}$

−1



AP: $\{l_0, l_1, l_0', l_1', x=0, x=1, y=0, y=1\}$

Before the exam: Look up Transition System semantics of a program graph.

- $S = Loc \times Eval(Var)$

→ don't name the states $s_0, s_1, \ldots$ in this case! The names of states are tuples consisting of the PG location and the variable valuations.

- $AP = Loc \cup Cond(Var)$

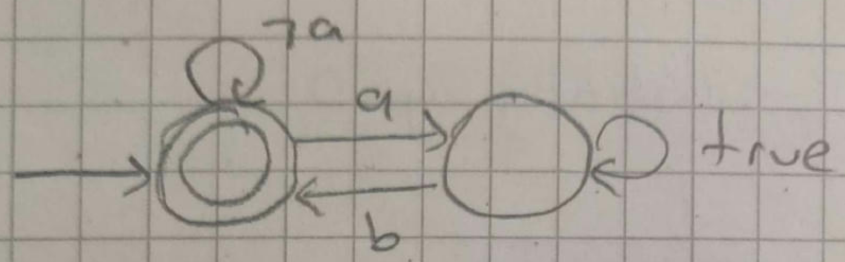
The set Cond(Var) of a PG are conditions on the variables. In your PGs, Cond(Var) = $\{x==0, y==0\}$ (your initial conditions). Don't use valuations as APs!

Exercise 2: From LTL to NBA and Back

2) 8/8

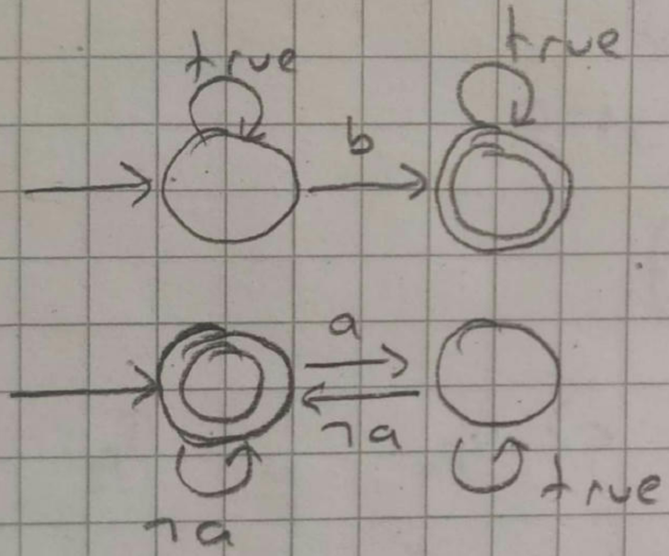
a)

2/2



b)

2/2



2/2 c) $\Box(\neg b \vee \bigcirc(\neg a \, \mathcal{U} \, \neg a \wedge \neg b))$

2/2 d) $c \wedge \bigcirc((a \wedge c) \vee (\neg a \wedge \neg b)) \wedge \bigcirc\bigcirc a$

Exercise 3: LT Properties for a Program

$$\mathrm{AP} = \{x = 0, x > 1\}$$

Assuming that "x is equal to 0" doesn't hold for $\mathrm{AP} = \{x = 0, x > 1\}$ and only holds for $\mathrm{AP} = \{x = 0\}$. Similar assumption is made for "x differs from 0" as well.

Part A

- a) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \text{false}\right\}$
- b) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \{x = 0\} = A_0\right\}$
- c) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \{x > 1\} = A_0\right\}$
- d) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \{x = 0\} = A_0 \land \exists i \in \mathbb{N}_1 . \{x = 1\} = A_i\right\}$
- e) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \overset{\infty}{\forall} i \in \mathbb{N}.\ \{x > 1\} \neq A_i\right\}$
- f) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \overset{\infty}{\exists} i \in \mathbb{N}.\ \{x > 1\} = A_i\right\}$
- g) $\left\{A_0 A_1 \ldots \in \left(2^{\mathrm{AP}}\right)^\omega \mid \text{true}\right\}$

Part B

- a) It's a safety property because it satisfies the condition that all traces that are not in the language has a bad prefix. Since there are no traces in the property, any finite trace is a bad prefix for this language.
- b) It's a safety property with $\mathrm{BadPref} = \left\{A_0 A_1 \ldots A_n \in \left(2^{\mathrm{AP}}\right)^+ \mid \{x = 0\} \neq A_0\right\}$
- c) It's a safety property with $\mathrm{BadPref} = \left\{A_0 A_1 \ldots A_n \in \left(2^{\mathrm{AP}}\right)^+ \mid \{x > 0\} \neq A_0\right\}$
- d) It's not a safety property. One counter example is $\sigma = \{x = 0\}^\omega$. σ does not satisfy this lt property, however all of it's finite prefixes can be extended to satisfy the property by appending $\{x > 1\}$ at some point. Therefore, this trace doesn't have any prefix that is a bad prefix of the language of this property.
- e) It's not a safety property because it's a liveness property. It's a liveness property because for any finite prefix we can extend it so that $\{x > 1\}$ does not appear infinitely often.
- f) Similar to part e), this is also a liveness property because we can extend any finite prefix such that $\{x > 1\}$ appears infinitely often. Therefore it's not a safety property.
- g) It's a safety property, because there are no traces that is not in the language of this property. Therefore it satisfies the safety property condition trivially.

Exercise 4: Fair Equivalence

6/6

a)

If $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2$ and $\psi_1 \underset{\text{fair}}{\equiv} \psi_2$, then $(\varphi_1 \vee \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 \vee \psi_2)$.

We know that $(\varphi_1 \vee \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 \vee \psi_2)$ is equivalent to
$\text{fair} \to (\varphi_1 \vee \psi_1) \equiv \text{fair} \to (\varphi_2 \vee \psi_2)$.

We need to show that:
1. $\text{Words}(\text{fair} \to (\varphi_1 \vee \psi_1)) \subseteq \text{Words}(\text{fair} \to (\varphi_2 \vee \psi_2))$
2. $\text{Words}(\text{fair} \to (\varphi_2 \vee \psi_2)) \subseteq \text{Words}(\text{fair} \to (\varphi_1 \vee \psi_1))$

Then the equivalence holds.

Without loss of generality, we can prove just the lemma 1. Lemma 2. can be proven in the same fashion.

Let $\sigma \vDash \text{Words}(\text{fair} \to (\varphi_1 \vee \psi_1))$. Then:
1. $\sigma \nvDash \text{fair}$: Then $\sigma \vDash \text{fair} \to (\varphi_2 \vee \psi_2)$ holds trivially.
2. $\sigma \vDash \text{fair}$: Then we also know that $\sigma \vDash (\varphi_1 \vee \psi_1)$. Then we have the following cases:
   1. $\sigma \vDash \varphi_1$: Then from $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2$ we can claim $\sigma \vDash \varphi_2$ as well. And therefore $\sigma \vDash (\varphi_2 \vee \psi_2)$. Hence, $\sigma \vDash \text{fair} \to (\varphi_2 \vee \psi_2)$.
   2. $\sigma \vDash \psi_1$: Then from $\psi_1 \underset{\text{fair}}{\equiv} \psi_2$ we can claim $\sigma \vDash \psi_2$ as well. And therefore $\sigma \vDash (\psi_2 \vee \psi_2)$. Hence, $\sigma \vDash \text{fair} \to (\varphi_2 \vee \psi_2)$.

Since $\forall \sigma \in \text{Words}(\text{fair} \to (\varphi_1 \vee \psi_1)).\ \sigma \in \text{Words}(\text{fair} \to (\varphi_2 \vee \psi_2))$, $\text{Words}(\text{fair} \to (\varphi_1 \vee \psi_1)) \subseteq \text{Words}(\text{fair} \to (\varphi_2 \vee \psi_2))$. Applying the same steps for the other direction we can conclude that $(\varphi_1 \vee \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 \vee \psi_2)$ ■

b)

If $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2$, then $(\bigcirc \varphi_1) \underset{\text{fair}}{\equiv} (\bigcirc \varphi_2)$.

We know that $(\bigcirc \varphi_1) \underset{\text{fair}}{\equiv} (\bigcirc \varphi_2)$ is equivalent to
$\text{fair} \to (\bigcirc \varphi_1) \equiv \text{fair} \to (\bigcirc \varphi_2)$.

We need to show that:
1. $\text{Words}(\text{fair} \to (\bigcirc \varphi_1)) \subseteq \text{Words}(\text{fair} \to (\bigcirc \varphi_2))$
2. $\text{Words}(\text{fair} \to (\bigcirc \varphi_2)) \subseteq \text{Words}(\text{fair} \to (\bigcirc \varphi_1))$

Then the equivalence holds.

Without loss of generality, we can prove just the lemma 1. Lemma 2. can be proven in the same fashion.

Let $\sigma \vDash \text{fair} \to (\bigcirc \varphi_1)$. Then:

1. $\sigma \not\vDash \text{fair}$: Then $\sigma \vDash \text{fair} \to (\bigcirc\, \varphi_2)$ holds trivially.
2. $\sigma \vDash \text{fair}$: Then we also know that $\sigma \vDash (\bigcirc\, \varphi_1)$. Let $\sigma' = \sigma[1..]$. We know that $\sigma' \vDash \varphi_1$. Also $\sigma' \vDash \text{fair}$ because σ' is a suffix of σ. So $\sigma' \vDash \text{fair} \to \varphi_1$. From $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2$, we can conclude that $\sigma' \vDash \text{fair} \to \varphi_2$. And because $\sigma' \vDash \text{fair}$, $\sigma' \vDash \varphi_2$ also holds. Thus $\sigma \vDash (\bigcirc\, \varphi_2)$. Hence we conclude $\sigma \vDash \text{fair} \to (\bigcirc\, \varphi_2)$.

Since $\forall \sigma \in \text{Words}(\text{fair} \to (\bigcirc\, \varphi_1)).\ \sigma \in \text{Words}(\text{fair} \to (\bigcirc\, \varphi_2))$, $\text{Words}(\text{fair} \to (\bigcirc\, \varphi_1)) \subseteq \text{Words}(\text{fair} \to (\bigcirc\, \varphi_2))$. Applying the same steps for the other direction we can conclude that $(\bigcirc\, \varphi_1) \underset{\text{fair}}{\equiv} (\bigcirc\, \varphi_2)$ ■

c)

If $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2\ \psi_1 \underset{\text{fair}}{\equiv} \psi_2$, then $(\varphi_1 U \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 U \psi_2)$.

We know that $(\varphi_1 U \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 U \psi_2)$ is equivalent to $\text{fair} \to (\varphi_1 U \psi_1) \equiv \text{fair} \to (\varphi_2 U \psi_2)$.

We need to show that:
1. $\text{Words}(\text{fair} \to (\varphi_1 U \psi_1)) \subseteq \text{Words}(\text{fair} \to (\varphi_2 U \psi_2))$
2. $\text{Words}(\text{fair} \to (\varphi_2 U \psi_2)) \subseteq \text{Words}(\text{fair} \to (\varphi_1 U \psi_1))$

Then the equivalence holds.

Without loss of generality, we can prove just the lemma 1. Lemma 2. can be proven in the same fashion.

Let $\sigma \vDash \text{fair} \to (\varphi_1 U \psi_1)$. Then:
1. $\sigma \not\vDash \text{fair}$: Then $\sigma \vDash \text{fair} \to (\varphi_2 \vee \psi_2)$ holds trivially.
2. $\sigma \vDash \text{fair}$: Then it must follow that $\sigma \vDash \varphi_1 U \psi_1$. There exists an $i \in \mathbb{N}$ such that $\sigma[i..] \vDash \psi_1$ and $\forall j \in \mathbb{N}.\ j < i \to \sigma[j..] \vDash \varphi_1$.

   Since every suffix of σ is also fair, from $\varphi_1 \underset{\text{fair}}{\equiv} \varphi_2$ we can conclude $\forall j \in \mathbb{N}.\ j < i \to \sigma[j..] \vDash \varphi_2$. And similarly from $\psi_1 \underset{\text{fair}}{\equiv} \psi_2$ we can conlude $\sigma[i..] \vDash \psi_2$. Therefore $\sigma \vDash \varphi_2 U \psi_2$ where the break point is i. Then $\sigma \vDash \text{fair} \to \varphi_2 U \psi_2$.

Since $\forall \sigma \in \text{Words}(\text{fair} \to (\varphi_1 U \psi_1)).\ \sigma \in \text{Words}(\text{fair} \to (\varphi_2 U \psi_2))$, $\text{Words}(\text{fair} \to (\varphi_1 U \psi_1)) \subseteq \text{Words}(\text{fair} \to (\varphi_2 U \psi_2))$. Applying the same steps for the other direction we can conclude that $(\varphi_1 U \psi_1) \underset{\text{fair}}{\equiv} (\varphi_2 U \psi_2)$ ■