Exercise 1

Consider the following transition system $TS$

and the regular safety property

$$p_{safe} = \text{"always if } a \text{ is valid and } b \land \neg c \text{ was valid somewhere before, then neither } a \text{ nor } b \text{ holds thereafter at least until } c \text{ holds"}$$

As an example, it holds:

$$\{b\} \emptyset \{a, b\} \{a, b, c\} \in \text{pref}(p_{safe})$$
$$\{a, b\} \emptyset \{b, c\} \in \text{pref}(p_{safe})$$
$$\{b\} \{a, c\} \{a\} \{a, b, c\} \in \text{BadPref}(p_{safe})$$
$$\{b\} \{a, c\} \{a\} \{a\} \in \text{BadPref}(p_{safe})$$

a) Define an NFA $A$ such that $L(A) = \text{MinBadPref}(p_{safe})$.

b) Decide whether $TS \models p_{safe}$ using the $TS \otimes A$ construction.

Provide a counterexample if $TS \not\models p_{safe}$. 

---

Exercise 2 (Model Checking Regular Safety Properties): (3 points)

Consider the following transition system $TS$ and the regular safety property $p_{safe}$ as defined above. 

As an example, it holds:

$$\{b\} \emptyset \{a, b\} \{a, b, c\} \in \text{pref}(p_{safe})$$
$$\{a, b\} \emptyset \{b, c\} \in \text{pref}(p_{safe})$$
$$\{b\} \{a, c\} \{a\} \{a, b, c\} \in \text{BadPref}(p_{safe})$$
$$\{b\} \{a, c\} \{a\} \{a\} \in \text{BadPref}(p_{safe})$$

Provide a counterexample if $TS \not\models p_{safe}$. 

---

Software Validation and Verification
Third Exercise Sheet – Regular Properties

Exercise 1 (Realizability & Fairness): (2 points)

Consider the transition system $TS$ on the right (where atomic propositions are omitted). Decide which of the following fairness assumptions $F_i$ are realizable for $TS$.

1. $F_1 = (\{s_0\}, \{s_1\}, \{s_2\})$
2. $F_2 = (\{s_0\}, \{s_1\}, \{s_2\}, \{s_3\})$
3. $F_3 = (\{s_0\}, \{s_1\}, \{s_2\}, \{s_3\}, \{\epsilon\})$

Justify your answers!
Exercise 2

a) Give the language for the following three NBA:

b) Give an NBA for:
   - "initially a occurs, and at some point b occurs" with $\Sigma = \{a, b, c\}$.
   - "if a occurs somewhere, then afterwards (b occurs infinitely often iff c occurs infinitely often)."
Exercise 3

a) Provide NBA $A_1$ and $A_2$ for the languages given by the expressions $(AC + B)^*B^\omega$ and $(B^*AC)^\omega$.

b) Apply the product construction to obtain a GNBA $G$ and an NBA $A$ with $L(A)_\omega = L(A_1)_\omega \cap L(A_2)_\omega$.
   
   **Hint:** Do not apply simplifications in these steps

b) Justify, why $L(G)_\omega = \emptyset$ where $G$ denotes the GNBA accepting the intersection.

Exercise 2 (LTL and Fairness): (3 points)

Consider the transition system TS above with the set $AP = \{a, b, c\}$ of atomic propositions. Note that this is a single transition system with two initial states. Consider the LTL fairness assumption $fa = \land(\land(a^\land b))\land(\land(a^\land \lnot b))$. 

a) Determine the fair paths in TS, i.e., the initial, infinite paths satisfying $fa$

b) For each of the following LTL formulae:

   1. $\textbf{1} = b \textbf{U} \land \lnot b$
   2. $\textbf{2} = b \textbf{W} \land \lnot b$
   3. $\textbf{3} = (b \land \land(\land(a^\land \lnot b)))$

   determine whether $\textbf{TS} \models \textbf{1}$. In case $\textbf{TS} \not\models \textbf{1}$, indicate a path $\pi$ for which $\pi \not\models \textbf{1}$.

   c) Redo the previous task but ignore the fairness assumption.
Exercise 4

Formally prove that there is no DBA \( \mathcal{A} \) over the alphabet \( \Sigma = \{a, b\} \) that accepts the language

\[
\mathcal{L} := \mathcal{L}_\omega((a + b)^* \cdot a^*).
\]
Exercise 5

Let the $\omega$-regular LT properties $P_1$ and $P_2$ over the set of atomic propositions $AP = \{a, b\}$ be given by

$P_1 := \text{“if a holds infinitely often, then b holds finitely often”}$

$P_2 := \text{“a holds infinitely often and b holds infinitely often”}$

The model is given by the transition system TS as follows:

![Transition System Diagram]

Algorithmically check whether $TS \models P_1$ and $TS \models P_2$. For this, proceed as follows.

a) Derive suitable NBA $A_{P_1}, A_{P_2}$, where suitable means “appropriate for part b)-d”).
   
   Hint: For $P_1$ you can find an automaton with 3 states and for $P_2$ 4 states suffice. Derive the automata directly.

b) Outline the reachable fragments of the product transition systems $TS \otimes A_{P_1}$ and $TS \otimes A_{P_2}$.

c) Decide whether $TS \models P_1$ by checking an appropriate persistence property via nested depth-first search on $TS \otimes A_{P_1}$. Document all changes to the contents of $U$, $V$, $\pi$ and $\xi$ (the state sets and stacks of the nested depth-first search, see lecture). If the property is violated, provide a counterexample based on the execution of the algorithm.

d) Decide whether $TS \models P_2$ by checking an appropriate persistence property via SCC analysis on $TS \otimes A_{P_2}$. If the property is violated, provide a counterexample based on your analysis.