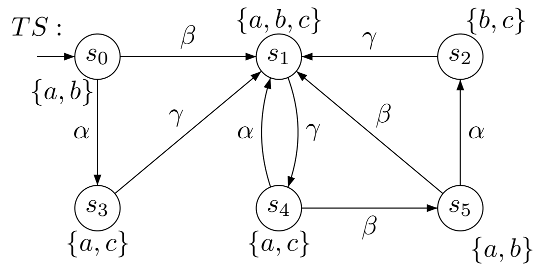
Software Validation and Verification
Third Exercise Sheet – Regular Properties

Exercise 1

Consider the following transition system TS



and the regular safety property

$$P_{\texttt{safe}} = \begin{array}{c}\text{“always if } a \text{ is valid and } b \wedge \neg c \text{ was valid somewhere before,} \\ \text{then neither } a \text{ nor } b \text{ holds thereafter at least until } c \text{ holds”}\end{array}$$

As an example, it holds:

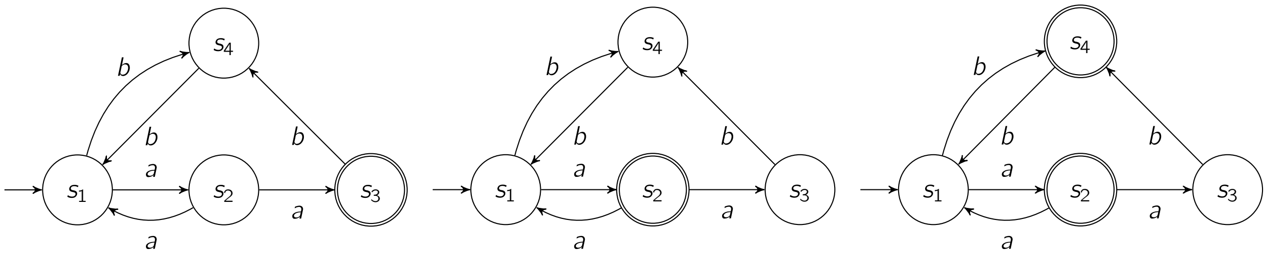
$$\{b\}\emptyset\{a,b\}\{a,b,c\} \in \mathit{pref}(P_{\texttt{safe}})$$
$$\{a,b\}\{a,b\}\emptyset\{b,c\} \in \mathit{pref}(P_{\texttt{safe}})$$
$$\{b\}\{a,c\}\{a\}\{a,b,c\} \in \mathit{BadPref}(P_{\texttt{safe}})$$
$$\{b\}\{a,c\}\{a,c\}\{a\} \in \mathit{BadPref}(P_{\texttt{safe}})$$

a) Define an NFA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathit{MinBadPref}(P_{\texttt{safe}})$.

b) Decide whether $TS \models P_{\texttt{safe}}$ using the $TS \otimes \mathcal{A}$ construction.
Provide a counterexample if $TS \not\models P_{\texttt{safe}}$.

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 $\mathcal{A}_1$ and $\mathcal{A}_2$ for the languages given by the expressions $(AC + B)^* B^\omega$ and $(B^* AC)^\omega$.

b) Apply the product construction to obtain an GNBA $\mathcal{G}$ and an NBA $\mathcal{A}$ with $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(\mathcal{A}_1) \cap \mathcal{L}_\omega(\mathcal{A}_2)$.
Hint: Do not apply simplifications in these steps

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

Exercise 4

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

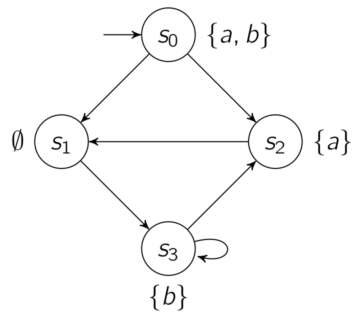
$$\mathcal{L} := \mathcal{L}_\omega((a+b)^*.a^\omega).$$

Exercise 5

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

$$P_1 := \text{“if } a \text{ holds infinitely often, then } b \text{ holds finitely often”}$$
$$P_2 := \text{“} a \text{ holds infinitely often and } b \text{ holds infinitely often”}$$

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



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

a) Derive *suitable* NBA $\mathcal{A}_{P_1}$, $\mathcal{A}_{P_2}$, 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 $TS \otimes \mathcal{A}_{P_1}$ and $TS \otimes \mathcal{A}_{P_2}$.

c) Decide whether $TS \models P_1$ by checking an appropriate persistence property via nested depth-first search on $TS \otimes \mathcal{A}_{P_1}$. Document *all* changes to the contents of U, V, π and ξ (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 $TS \otimes \mathcal{A}_{P_2}$. If the property is violated, provide a counterexample *based on your analysis*.