Technische Universität München **Summer term 2018**
J. Esparza / M. Blondin / P. Ashok

Model checking — Endterm

- You have **120 minutes** to complete the exam.
- Answers must be written in a **separate booklet**. Do not answer on the exam.
- Please let us know if you need more paper.
- Write your name and Matrikelnummer on every sheet.
- Write with a non-erasable **pen**. Do not use red or green.
- You are not allowed to use auxiliary means other than pen and paper.
- You can obtain **40 points**. You need **17 points** to pass.

Question 1 LTL and Büchi automata ($\mathbf{2+2+2+2=8}$ points)
Consider the following LTL formulae over the set of atomic propositions $AP = \{p, q\}$:

$$\phi_1 = \mathbf{FG}(p \,\mathbf{U}\, q) \qquad \phi_2 = \mathbf{FG}(\neg p \to q) \qquad \phi_3 = \mathbf{G}(\neg p \vee (p \,\mathbf{R}\, q))$$

(a) Is there a word satisfying ϕ_1 but not ϕ_2? If so, exhibit such a word and if not, briefly explain why it does not exist.

(b) Is there a word satisfying ϕ_2 but not ϕ_1? If so, exhibit such a word and if not, briefly explain why it does not exist.

(c) Is there a word satisfying all three formulae? If so, exhibit such a word and if not, briefly explain why it does not exist.

(d) Give a Büchi automaton accepting exactly the words satisfying ϕ_1. Make sure it accepts the following words: $\{p, q\}^\omega, \{p\}\{q\}^\omega$ and rejects the following words: $\emptyset^\omega, \{p\}^\omega$.

Question 2 CTL ($\mathbf{1+1+1+1=4}$ points)
Consider the CTL formulas $\mathbf{EF}p, \mathbf{EFAG}p, \mathbf{AGEF}p, \mathbf{AGAF}p, \mathbf{AG}p$. Draw

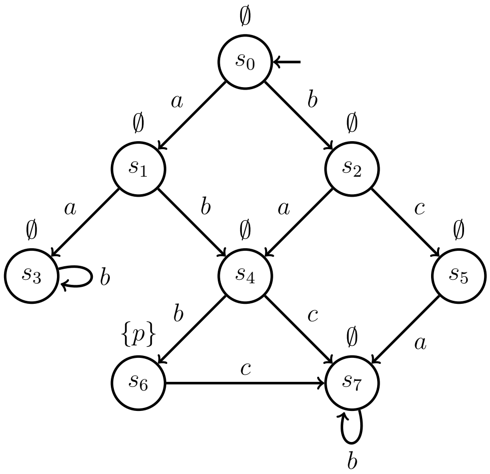
(a) a Kripke structure $\mathcal{K}_1$ satisfying $\mathbf{EF}p$ but not $\mathbf{EFAG}p$;

(b) a Kripke structure $\mathcal{K}_2$ satisfying $\mathbf{EFAG}p$ but not $\mathbf{AGEF}p$;

(c) a Kripke structure $\mathcal{K}_3$ satisfying $\mathbf{AGEF}p$ but not $\mathbf{AGAF}p$;

(d) a Kripke structure $\mathcal{K}_4$ satisfying $\mathbf{AGAF}p$ but not $\mathbf{AG}p$.

Question 3 Partial order reduction ($\mathbf{1+1+1+1+1=5}$ points)
Consider the labelled Kripke structure $\mathcal{K} = (S, A, \to, r, AP, \nu)$ where $S = \{s_0, \ldots, s_7\}$, $A = \{a, b, c\}$, $r = \{s_0\}$, $AP = \{p\}$, and $\to$ and ν are graphically represented below. Observe that p holds only at state s_6 and nowhere else.



(a) Give the largest relation $I \subseteq A \times A$ satisfying the three properties of an independence relation (irreflexivity, symmetry, and the "diamond property") and explain why it is the largest.

(b) Give the largest invisibility set $U \subseteq A$.

(c) Does $red(s_0) = \{a\}$ satisfy condition C_1 for I and U? Justify your answer.

(d) Does $red(s_4) = \{b\}$ satisfy all of C_0–C_3 for I and U? Justify your answer.

(e) Does $red(s_2) = \{a\}$ satisfy all of C_0–C_3 for I and U? Justify your answer.

Recall: the conditions that $red(s)$ has to satisfy are

- C0: $red(s) = \emptyset$ iff $en(s) = \emptyset$.
- C1: Every path starting at s satisfies: no action dependent on some action in $red(s)$ can be executed without an action from $red(s)$ occurring first.
- C2: If $red(s) \neq en(s)$ then all actions in $red(s)$ are invisible.
- C3: For all cycles in the reduced Kripke structure the following holds: if $a \in en(s)$ for some state s in the cycle, then $a \in red(s)$ for some (possibly other) state s' in the cycle.

Question 4 BDDs (3 + 3 = 6 points)
Assume that you are given a Kripke structure with states $S = \{s_0, s_1, \ldots, s_7\}$.

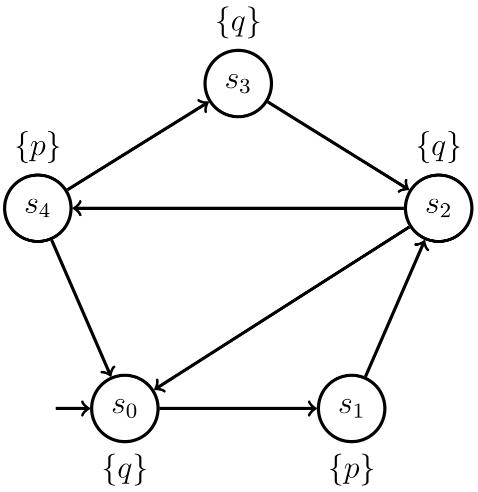
(a) Compute a multi-BDD representing the two subsets of states $P = \{s_0, s_1, s_3, s_5, s_7\}$ and $Q = \{s_0, s_2, s_6, s_7\}$. Encode each state of S using three bits in the obvious way:

$$s_0 \mapsto 000, s_1 \mapsto 001, \ldots, s_7 \mapsto 111.$$

Use the ordering $b_0 < b_1 < b_2$ where b_0 is the most significant bit and b_2 is the least significant bit of the binary encoding.

(b) Compute the set $P \cap Q$ using the BDD intersection algorithm. Show the recursion tree.

Question 5 Abstraction refinement (2 + 1 + 2 = 5 points)
Consider the labelled Kripke structure $\mathcal{K} = (S, A, \rightarrow, r, AP, \nu)$ where $AP = \{p, q\}$, and $S, A, \rightarrow$ and ν are graphically represented as follows:



Let $\approx_p$ be the equivalence relation over S given by $s \approx_p t$ iff $\nu(s) = \nu(t)$.

(a) Construct the Kripke structure $\mathcal{K}'$ obtained by abstracting S w.r.t. $\approx_p$.

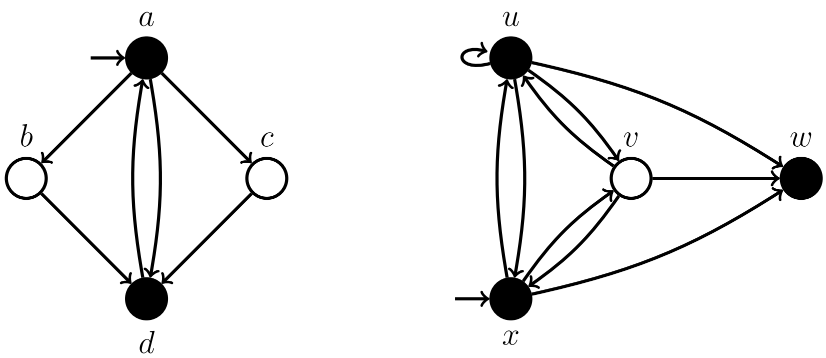
(b) Give a counterexample showing that $\mathcal{K}'$ does not satisfy $\mathbf{GF}p$.

(c) Following the procedure described in the course, use the counterexample to refine $\mathcal{K}'$ into a Kripke structure $\mathcal{K}''$.

(d) **2 Bonus points**: Keep refining the abstraction until you prove that the property holds.

Question 6 Simulations and Bisimulations ($2+2=4$ points)

Consider the three following Kripke structures $\mathcal{K}_1$ (left) and $\mathcal{K}_2$ (right):



States coloured black satisfy proposition p and others do not. For (a) and (b), if your answer is *yes*, then give a simulation relation, and if it is *no*, then explain why not. For (c), give a bisimulation relation.

(a) Does $\mathcal{K}_2$ simulate $\mathcal{K}_1$?

(b) Does $\mathcal{K}_1$ simulate $\mathcal{K}_2$?

(c) **2 Bonus points:** Give a Kripke structure $\mathcal{K}_3$ bisimilar to $\mathcal{K}_2$ but smaller than $\mathcal{K}_2$. Explain why they are bisimilar.

Question 7 Pushdown systems (3 + 3 + 2 = 8 points)

Consider the following recursive program with a global boolean variable `x`:

```
            boolean x;

            procedure foo;                procedure bar;
f0:           x := not x;         b0:       if x then
                                              call foo;
f1:           if x then                     endif;
                call foo;
              else                b1:       return;
                call bar;
              endif;

f2:           return;
```

(a) Model the program, where the value of `x` is not initialized, with a pushdown system $\mathcal{P} = (P, \Gamma, \Delta)$. Give explicit enumerations of the set of control states P, the stack alphabet Γ, and the set of rules Δ.
Hint: Δ contains 10 rules.

(b) Let E be the set of all configurations of $\mathcal{P}$ with empty stack. Give a $\mathcal{P}$-automaton recognizing the language E. Use the saturation rule to compute a $\mathcal{P}$-automaton recognizing the language $pre^*(E)$. For each transition added by the saturation rule, explain how it is generated.
Hint: The $\mathcal{P}$-automaton for $pre^*(E)$ should have 10 transitions.

(c) Give a regular expression for the set of all initial configurations of the program, where we assume that `foo` is the main procedure and, as above, `x` is not initialized. Is there an initial configuration from which it is impossible to terminate? Briefly justify your answer.

Solution 1   LTL and Büchi automata $(2+2+2+2=8$ points)

$\phi_1 = \mathbf{FG}(p\ \mathbf{U}\ q)$ — eventually, $\emptyset$ must stop occurring and q must appear infinitely often.
$\phi_2 = \mathbf{FG}(\neg p \rightarrow q)$ — eventually always $p \vee q$.
$\phi_3 = \mathbf{G}(\neg p \vee (p\ \mathbf{R}\ q))$ — equivalent to $\mathbf{G}(\neg p \vee (p \wedge q))$.

(a) No. $p\ \mathbf{U}\ q \implies p \vee q$. Hence $\mathbf{FG}(p\ \mathbf{U}\ q) \implies \mathbf{FG}(p \vee q) \equiv \mathbf{FG}(\neg p \rightarrow q)$.

(b) Yes. $\{p\}^\omega$ satisfies ϕ_2 but not ϕ_1.

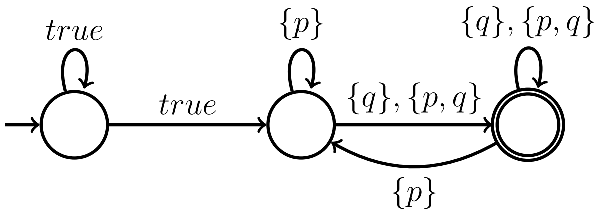
(c) Yes. $\{p, q\}^\omega$ satisfies all three.
(a) is satisfied because $\mathbf{G}(p \wedge q) \implies \mathbf{G}(p\ \mathbf{U}\ q)$;
(b) is satisfied because $p \wedge q \implies p \vee q$; and
(c) is satisfied because $\phi_3 \implies \mathbf{G}(\neg p \vee (p \wedge q))$ and the word ensures $p \wedge q$ at all points.

(d) It should accept

- $\{p, q\}^\omega$
- $\emptyset\{p, q\}^\omega$
- $\{p\}\{q\}^\omega$
- $(\{p\}\{q\})^\omega$
- $\{q\}^\omega$

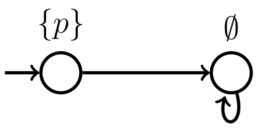
and it should reject

- $\emptyset^\omega$
- $\{p\}^\omega$

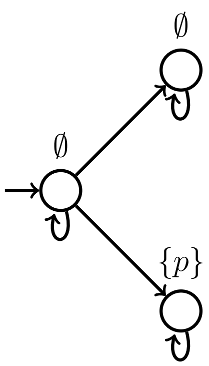


Solution 2   CTL $(1+1+1+1=4$ points)

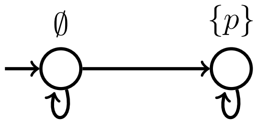
(a) $\mathbf{EF}p$ but not $\mathbf{EFAG}p$



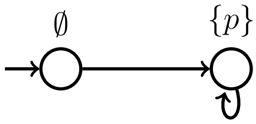
(b) $\mathbf{EFAG}p$ but not $\mathbf{AGEF}p$



(c) **AGEF**p but not **AGAF**p



(d) **AGAF**p but not **AG**p



Solution 3 Partial order reduction (1 + 1 + 1 + 1 + 1 = 5 points)

(a) $I = \{(a, c), (c, a), (b, c), (c, b)\}$. It cannot include $\{(a, b), (b, a)\}$ because the diamond property is violated in s_3.

(b) $\{a\}$

(c) No, C1 is violated because b can be executed before a.

(d) No, C2 is violated because b is visible.

(e) (i) C0 is satisfied because $red(s_2)$ is not empty.

(ii) C1 is satisfied. The only path from s_2 which doesn't execute a is $s_2 \xrightarrow{c} s_5 \xrightarrow{a} s_7 \ldots$ and in this path, no action dependent on a is executed before a (since $(a, c) \in I$).

(iii) C2 is satisfied because a is invisible.

(iv) C3 is satisfied because in the reduced Kripke structure, the only two cycles are at s_3 and s_7, and red of both these states will be non-empty because en is non-empty.

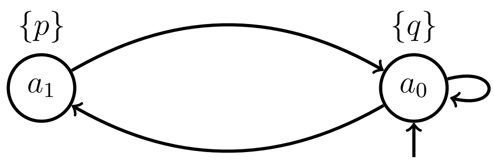
Solution 4 BDDs (3 + 3 = 6 points)

...

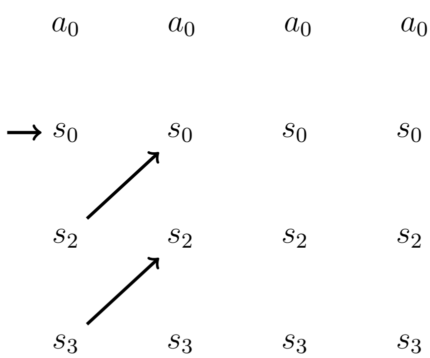
Solution 5 Abstraction refinement (2 + 1 + 2 = 5 points)

(a) First abstraction:

$$a_0 = \{s_0, s_2, s_3\},$$
$$a_1 = \{s_1, s_4\}$$

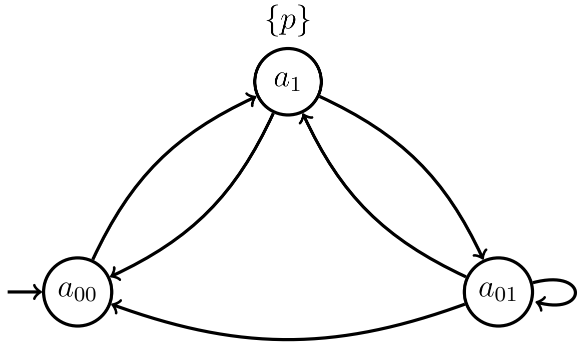


(b) Counter-example: $a_0{}^\omega$. We have $|a_0| = 3$, so we unroll the loop 4 times:

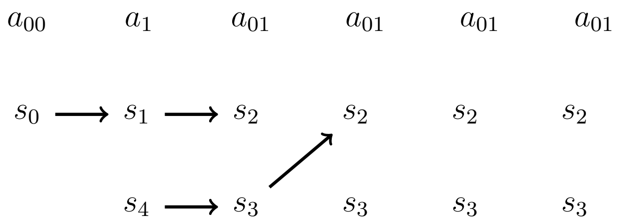


Fails to concretize in 1 step, so we realize that we need to refine. The states which are reachable from the initial state should be distinguished from the states which still have successors. We introduce:

$$a_{00} = \{s_0\},$$
$$a_{01} = \{s_2, s_3\},$$
$$a_1 = \{s_1, s_4\}.$$



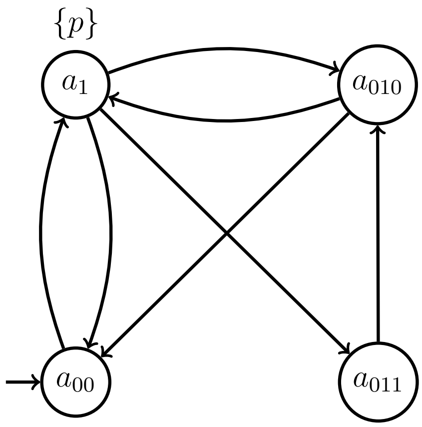
Counter-example: $a_{00} a_1 a_{01} {a_{01}}^{\omega}$:



We split s_2 and s_3, and introduce:

$$a_{010} = \{s_2\},$$
$$a_{011} = \{s_3\}.$$

We obtain the following which satisfies $\mathbf{GF}p$:

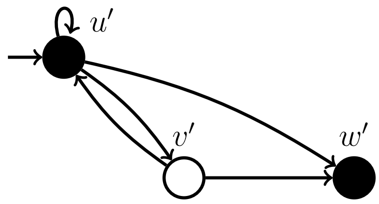


Solution 6 Simulations and Bisimulations (2 + 2 = 4 points)

(a) Yes: $\{(a, x), (b, v), (c, v), (d, w)\}$.

(b) No, we prove by contradiction. Assume there $\mathcal{K}_1$ simulates $\mathcal{K}_2$ and let H be the simulation. Since x and a are the respective initial states, $(x, a) \in H$. Since $(x, a) \in H$ and $x \to u$ where u is black, there must exist a black state in $\mathcal{K}_1$ with a transition from a. The only candidate in this case is d. Hence, $(u, d) \in H$. By a similar argument, if $(u, d) \in H$ and $u \to v$ where v is white, then there must exist a white state in $\mathcal{K}_1$ with a transition from d — which is not the case. Hence $\mathcal{K}_1$ does not simulate $\mathcal{K}_2$.

(c) Merge x and u in $\mathcal{K}_2$ to get $\mathcal{K}_3$ as shown below.



Define the bisimulation relation is as follows: $H = \{(x, u'), (u, u'), (v, v'), (w, w')\}$. Then H must be a simulation from $\mathcal{K}_2$ to $\mathcal{K}_3$ and $H' = \{(t, s) \mid (s, t) \in H\}$ must be a simulation from $\mathcal{K}_3$ to $\mathcal{K}_2$. First we prove that H is indeed a simulation from $\mathcal{K}_2$ to $\mathcal{K}_3$. We have

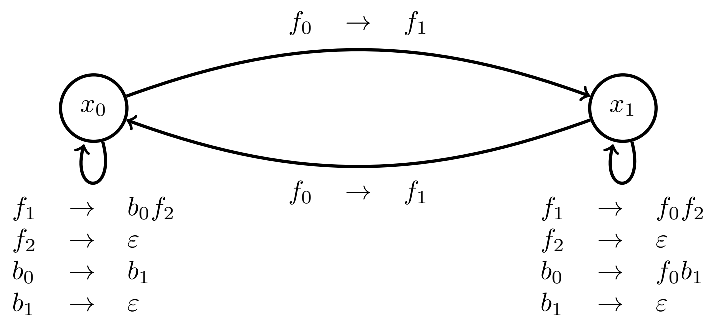
(i) $(x, u') \in H$, $x \to u$ and $u' \to u'$, $(u, u') \in H$.
(ii) $(x, u') \in H$, $x \to v$ and $u' \to v'$, $(v, v') \in H$.
(iii) $(x, u') \in H$, $x \to w$ and $u' \to w'$, $(w, w') \in H$.
(iv) $(u, u') \in H$, $u \to u$ and $u' \to u'$, $(u, u') \in H$.
(v) $(u, u') \in H$, $u \to v$ and $u' \to v'$, $(v, v') \in H$.
(vi) $(u, u') \in H$, $u \to w$ and $u' \to w'$, $(w, w') \in H$.
(vii) $(u, u') \in H$, $u \to x$ and $u' \to u'$, $(x, u') \in H$.
(viii) $(v, v') \in H$, $v \to u$ and $v' \to u'$, $(u, u') \in H$.
(ix) $(v, v') \in H$, $v \to w$ and $v' \to w'$, $(w, w') \in H$.
(x) $(v, v') \in H$, $v \to x$ and $v' \to u'$, $(x, u') \in H$.

Now we prove H' is a simulation from $\mathcal{K}_3$ to $\mathcal{K}_2$.

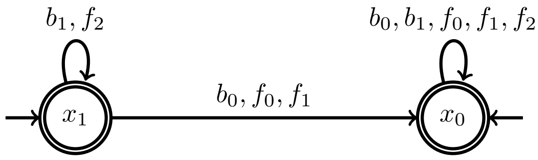
(i) $(u', x) \in H'$, $u' \to u'$ and $x \to u$, $(u', u) \in H'$.
(ii) $(u', x) \in H'$, $u' \to v'$ and $x \to v$, $(v', v) \in H'$.
(iii) $(u', x) \in H'$, $u' \to w'$ and $x \to w$, $(w', w) \in H'$.
(iv) $(v', v) \in H'$, $v' \to u'$ and $v \to u$, $(u', u) \in H'$.
(v) $(v', v) \in H'$, $v' \to w'$ and $v \to w$, $(w', w) \in H'$.

Solution 7 Pushdown systems (3 + 3 + 2 = 8 points)

(a) The stack alphabet is $\Gamma = \{f_0, f_1, f_2, b_0, b_1\}$ and the pushdown system is as follows:



(b)



(c) The regular expression is $x_0 f_0 + x_1 f_1$. No, there is no such configuration since the $\mathcal{P}$-automaton obtained in (b) accepts both $x_0 f_0$ and $x_1 f_1$.