Examination Algorithms for Model Checking (2IW55)

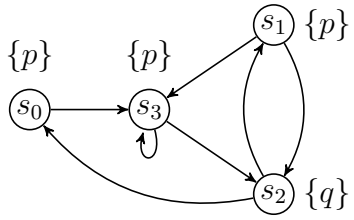
20 January, 2011, 14:00 – 17:00

Important notes:

- The exam consists of four questions.
- Weighting: 1: **30**, 2: **20**, 3: **20**, 4: **30**.
- Carefully read and answer the questions. The book, the course notes and other written material may be used during this examination.

---

1. Consider the following Kripke Structure K:



Consider the following formulae, where p and q are atomic propositions:

(**A**) **A** $[q$ **R** $p]$
(**B**) **AG**(**AF**q)

(a) Determine the set of states in K where (**A**) holds using the **symbolic model checking algorithm** for CTL model checking. Use set notation to represent states instead of BDDs. Show all intermediate steps.

Solution: The symbolic model checking algorithm relies on fixpoint computations for computing the set of states for which the formula holds. We first convert **A** $[q$ **R** $p]$ *to* $\neg\mathbf{E}[(\neg q)$ **U** $(\neg p)]$. *The encoding, using fixpoints, for this formula is as follows:*

$$S \setminus \mu Z.(\neg p \cup (\neg q \cap \mathbf{EX}Z))$$

The solution to Z can be obtained by the following approximation scheme:

$$\begin{aligned}
Z_0 &= \emptyset \\
Z_1 &= \neg p \cup (\neg q \cap \mathbf{EX} Z_0) \\
&= \{s_2\} \cup (\{s_0, s_1, s_3\} \cap \mathbf{EX}\emptyset) \\
&= \{s_2\} \\
Z_2 &= \neg p \cup (\neg q \cap \mathbf{EX} Z_1) \\
&= \{s_2\} \cup (\{s_0, s_1, s_3\} \cap \mathbf{EX}\{s_2\}) \\
&= \{s_2\} \cup (\{s_0, s_1, s_3\} \cap \{s_1, s_3\}) \\
&= \{s_2\} \cup \{s_1, s_3\} \\
&= \{s_1, s_2, s_3\} \\
Z_3 &= \neg p \cup (\neg q \cap \mathbf{EX} Z_1) \\
&= \{s_2\} \cup (\{s_0, s_1, s_3\} \cap \mathbf{EX}\{s_1, s_2, s_3\}) \\
&= \{s_2\} \cup (\{s_0, s_1, s_3\} \cap \{s_0, s_1, s_2, s_3\}) \\
&= S
\end{aligned}$$

Hence, the formula holds in $S \setminus S = \emptyset$, i.e., in no state.

(b) Consider the fairness constraint $\mathcal{F} = \{\{s_2\}, \{s_3\}\}$. Determine the set of states in K where (**B**) holds fairly under $\mathcal{F}$ using the **labelling algorithm** for fair CTL. Show the intermediate steps.

Solution: We must compute $\mathbf{AG}(\mathbf{AF}q)$; for this, we first convert this formula to existential form: $\mathbf{AG}(\mathbf{AF}q) \equiv \neg\mathbf{EF}(\mathbf{EG}\neg q) \equiv \neg\mathbf{E}[\mathit{true}\ \mathbf{U}\ \mathbf{EG}\neg q]$. Since we are dealing with a $\mathbf{EU}$ formula under fairness constraints, we first determine the set of states where $\mathbf{EG}\mathit{true}$ holds fairly. As the entire graph is an SCC, every state is the start of a fair path. We thus compute $\mathbf{E}[\mathit{true}\ \mathbf{U}\ (fair \wedge \mathbf{EG}\neg q)]$.

We next compute where $\mathbf{EG}\neg q$ holds fairly. First, we compute the non-trivial SCCs within set of states where $\neg q$ holds, i.e., $\{s_0, s_1, s_3\}$. The non-trivial SCCs are s_3, but it is a non-fair SCC. As a result, $\mathbf{EG}\neg q$ holds fairly no-where; it is the equivalent of false, and no state is labelled with $\mathbf{EG}\neg q$. We use this result in computing $\mathbf{E}[\mathit{true}\ \mathbf{U}(fair \wedge \mathbf{EG}\neg q)]$. Computing the set of states where this formula holds is trivial: no state satisfies $\mathbf{EG}\neg q$; thus, no state leads to this set, so no states are labelled with $\mathbf{E}[\mathit{true}\ \mathbf{U}\mathit{false}]$. Thus, all states are labelled with $\neg\mathbf{E}[\mathit{true}\ \mathbf{U}\ \mathit{false}]$, i.e., our original formula. Hence, our formula holds fairly in all states.

2. Consider the PBES $\mathcal{E}$ given below:

$$\begin{aligned}
\Big(\nu Z(b{:}Bool, c{:}Bool, n{:}Nat) \ = \ & (\neg b \wedge c \wedge Z(c, b, n)) \vee (b \wedge \neg c \wedge Z(\neg b, \neg c, n)) \\
\vee \ & \exists j{:}Nat.\exists m{:}Nat.(\neg(b \vee c) \wedge m = j \wedge Z(c, \neg b, m))\Big)
\end{aligned}$$

(a) Identify the set of "positive redundant" parameters in the PBES $\mathcal{E}$ and, if possible, simplify $\mathcal{E}$ as a result of your analysis. Show all necessary computations.

Solution: We construct a marked influence graph $G = (V, E, M)$ by analysing $\mathcal{E}$. Let ϕ denote the right-hand side of the equation for Z in $\mathcal{E}$. We obtain:

$$\begin{aligned} \mathsf{sig}(\phi) &= \{b, c\} \\ \mathsf{dep}(\phi) &= \{Z(c, b, n), Z(\neg b, \neg c, n), Z(c, \neg b, m)\} \end{aligned}$$

From this, and from the signature of the equation for Z, we derive the following graph:

- $V = \{b, c, n\}$
- $E = \{(b, b), (b, c), (c, b), (c, c), (n, n)\}$, *which is obtained from* $\mathsf{dep}(\phi)$
- $M = \{b, c\}$, *which is obtained from* $\mathsf{sig}(\phi)$

Next, perform a backwards reachability analysis on G, identifying vertices in V that can reach a vertex in M; this happens to be exactly the set M. Hence, the vertices in $V \setminus M = \{n\}$ are positively redundant. As a result, we can simplify $\mathcal{E}$ by removing the variable n (and the expressions updating this variable), and the existential quantifiers:

$$\begin{aligned} \Big(\nu \tilde{Z}(b{:}Bool, c{:}Bool) = \; & (\neg b \wedge c \wedge \tilde{Z}(c, b)) \vee (b \wedge \neg c \wedge \tilde{Z}(\neg b, \neg c)) \\ \vee \; & (\neg(b \vee c) \wedge \tilde{Z}(c, \neg b))\Big) \end{aligned}$$

The solution to Z can be found by solving $\tilde{Z}$: $\tilde{Z}(b, c) = Z(b, c, n)$ for all values of b, c, n.

(b) Solve the PBES $\mathcal{E}$ using symbolic approximation. Show all intermediate steps.

Solution: We need to approximate the solution to Z. Since we are looking for the greatest fixpoint solution to the equation for Z, the approximation starts with the function $Z_0(b, c, n) = \mathsf{true}$ for all b, c, n. We denote the next approximants by $Z_i(b, c, n)$:

$$\begin{aligned} Z_1(b, c, n) &= (\neg b \wedge c \wedge Z_0(c, b, n)) \vee (b \wedge \neg c \wedge Z_0(\neg b, \neg c, n)) \\ &\quad \vee \exists j{:}Nat.\exists m{:}Nat.(\neg(b \vee c) \wedge m = j \wedge Z_0(c, \neg b, m)) \\ &= (\neg b \wedge c) \vee (b \wedge \neg c) \vee \exists j{:}Nat.\exists m{:}Nat.(\neg b \wedge \neg c \wedge m = j) \\ &= (\neg b \wedge c) \vee (b \wedge \neg c) \vee (\neg b \wedge \neg c) \\ &= (\neg b \wedge (c \vee \neg c)) \vee (b \wedge \neg c) \\ &= \neg b \vee \neg c \end{aligned}$$

$$\begin{aligned} Z_2(b, c, n) &= (\neg b \wedge c \wedge Z_1(c, b, n)) \vee (b \wedge \neg c \wedge Z_1(\neg b, \neg c, n)) \\ &\quad \vee \exists j{:}Nat.\exists m{:}Nat.(\neg(b \vee c) \wedge m = j \wedge Z_1(c, \neg b, m)) \\ &= (\neg b \wedge c \wedge (\neg c \vee \neg b)) \vee (b \wedge \neg c \wedge (b \vee c)) \\ &\quad \vee \exists j{:}Nat.\exists m{:}Nat.(\neg b \wedge \neg c \wedge m = j \wedge (\neg c \vee b)) \\ &= (\neg b \wedge c) \vee (b \wedge \neg c) \vee (\neg b \wedge \neg c) \\ &= Z_1(b, c, n) \end{aligned}$$

Since $Z_2(b, c, n) = Z_1(b, c, n)$, the solution to Z is the function given by Z_2.

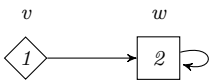
3. Let $\mathcal{G} = (V, E, p, (V_\Diamond, V_\Box))$ denote a parity game, in which $V_\Diamond$ is the set of vertices owned by player Even, and $V_\Box$ is the set of vertices owned by player Odd. Read the three statements below carefully, and, for each statement, give either a short proof or a counterexample.

   (a) For all parity games $\mathcal{G}$ with $V_\Box = \emptyset$, player Even wins all vertices in $\mathcal{G}$.

   Solution: this statement is not true. In order to prove this, we provide a counterexample. For instance, the Parity Game $\mathcal{G}$, consisting of a single vertex that belongs to player Even, which has a self-loop and an arbitrary odd priority; the entire game $\mathcal{G}$ is then won by player Odd.

   (b) For all parity games $\mathcal{G}$ and all sets $U \subseteq V$, the following property holds: for all vertices $v, w \in V$ such that $v \to w$, if $v \in Attr_\Diamond(\mathcal{G}, U)$ then also $w \in Attr_\Diamond(\mathcal{G}, U)$.

   Solution: this statement is not true. Consider the Parity Game $\mathcal{G}$ given below.

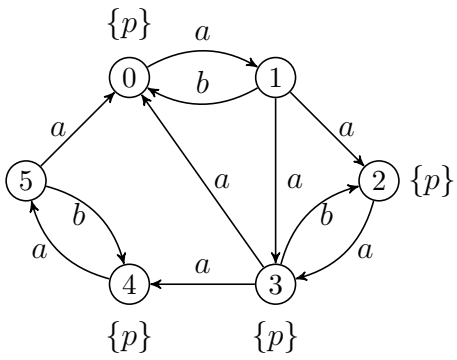


   Choose $U = \{v\}$. Then the statement does not hold for $v \in U$, as one can easily show that $Attr_\Diamond(\mathcal{G}, U) = U$ and $w \notin U$.

   (c) Let $\mathcal{G}$ be an arbitrary parity game. Let $U \subseteq V$ such that for all $v \in U$, v has only a selfloop (*i.e.*, for all $w \in V$, if $v \to w$ then $w = v$), and there is an odd $k \in \mathbb{N}$, such that $k = p(v)$ for all $v \in U$ (*i.e.*, each vertex in U has the same odd priority). Then the Small Progress Measures algorithm requires lifting the game parity progress measure for $\mathcal{G}$ at least $|U|^2$ times to become stable.

   Solution: this statement is true. Let k be the priority for the vertices in U. This means there are at least $|U|$ vertices of priority k in $\mathcal{G}$. The k^{th} position of the set $\mathbb{M}_\mathcal{G}^\top$ is thus a superset of the set $\{i \mid i \leq |U|\}$. Let μ_0 be the initial progress measure for $\mathcal{G}$, and let $u \in U$ be an arbitrary vertex. Let $\mu_{i+1} = Lift_u(\mu_i)$ for all $i \leq |U|$. Then $\mu_{i+1} \neq \mu_i$ for $i \leq |U|$. Since all vertices in U only have self-loops, they do not depend on other vertices. Thus, each vertex in U can be lifted, like u, for $|U| + 1$ times, irrespective of the order in which vertices are lifted. In total, this means that the progress measure can be lifted for at least $|U| \times (|U| + 1) > |U|^2$ times before stabilising.

4. Consider the mixed Kripke Structure K, and the equation system $\mathcal{E}$.



$$
\begin{aligned}
(\mu X_0 &= Y_0)\\
(\mu X_1 &= Y_1)\\
(\mu X_2 &= Y_2)\\
(\mu X_3 &= Y_3)\\
(\mu X_4 &= Y_4)\\
(\mu X_5 &= Y_5)\\
(\nu Y_0 &= Y_2 \wedge Y_3)\\
(\nu Y_1 &= X_0)\\
(\nu Y_2 &= Y_0 \wedge Y_4)\\
(\nu Y_3 &= X_2 \vee (Y_1 \wedge Y_5))\\
(\nu Y_4 &= Y_0)\\
(\nu Y_5 &= X_4)
\end{aligned}
$$

(a) Let the modal μ-calculus formula ϕ be defined as:

$$\mu X.\nu Y.\Big(\langle b\rangle X \vee (p \wedge [a][a]Y)\Big)$$

Prove or disprove that the BES $\mathcal{E}$ given above can be obtained from the translation of the model checking problem for $K \models \phi$ (up-to Boolean equivalence of the right-hand side expressions). Motivate your answer by means of a computation.

The BES is the result of the translation of the model checking problem; some proposition logic is used. The left-hand sides follow the subformula ordering of

the fixpoints. We show the translation:

$$
\begin{aligned}
\mu X_0 &= \mathsf{RHS}(0, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_0\\
\mu X_1 &= \mathsf{RHS}(1, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_1\\
\mu X_2 &= \mathsf{RHS}(2, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_2\\
\mu X_3 &= \mathsf{RHS}(3, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_3\\
\mu X_4 &= \mathsf{RHS}(4, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_4\\
\mu X_5 &= \mathsf{RHS}(5, \nu Y.(\langle b\rangle X \vee (p \wedge [a][a]Y))) = Y_5\\
\nu Y_0 &= \mathsf{RHS}(0, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(0, \langle b\rangle X) \vee (\mathsf{RHS}(0, p) \wedge \mathsf{RHS}(0, [a][a]Y))\\
&= \mathit{false} \vee (\mathit{true} \wedge \mathsf{RHS}(1, [a]Y))\\
&= Y_2 \wedge Y_3\\
\nu Y_1 &= \mathsf{RHS}(1, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(1, \langle b\rangle X) \vee (\mathsf{RHS}(1, p) \wedge \mathsf{RHS}(1, [a][a]Y))\\
&= X_0 \vee (\mathit{false} \wedge \mathsf{RHS}(2, [a]Y) \wedge \mathsf{RHS}(3, [a]Y))\\
&= X_0\\
\nu Y_2 &= \mathsf{RHS}(2, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(2, \langle b\rangle X) \vee (\mathsf{RHS}(2, p) \wedge \mathsf{RHS}(2, [a][a]Y))\\
&= \mathit{false} \vee (\mathit{true} \wedge \mathsf{RHS}(3, [a]Y))\\
&= Y_0 \wedge Y_4\\
\nu Y_3 &= \mathsf{RHS}(3, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(3, \langle b\rangle X) \vee (\mathsf{RHS}(3, p) \wedge \mathsf{RHS}(3, [a][a]Y))\\
&= X_2 \vee (\mathit{true} \wedge \mathsf{RHS}(4, [a]Y) \wedge \mathsf{RHS}(0, [a]Y))\\
&= X_2 \vee (Y_1 \wedge Y_5)\\
\nu Y_4 &= \mathsf{RHS}(4, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(4, \langle b\rangle X) \vee (\mathsf{RHS}(4, p) \wedge \mathsf{RHS}(4, [a][a]Y))\\
&= \mathit{false} \vee (\mathit{true} \wedge \mathsf{RHS}(5, [a]Y))\\
&= Y_0\\
\nu Y_5 &= \mathsf{RHS}(5, \langle b\rangle X \vee (p \wedge [a][a]Y))\\
&= \mathsf{RHS}(5, \langle b\rangle X) \vee (\mathsf{RHS}(5, p) \wedge \mathsf{RHS}(5, [a][a]Y))\\
&= X_4 \vee (\mathit{false} \wedge \mathsf{RHS}(0, [a]Y))\\
&= X_4
\end{aligned}
$$

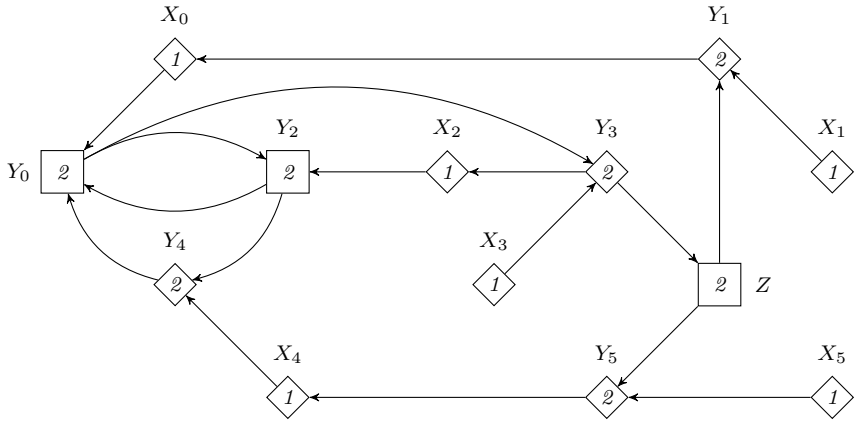
(b) Give the Parity Game obtained by translating $\mathcal{E}$ and solve the game using either the recursive algorithm or the Small Progress Measures algorithm. Show all the intermediate steps.

Solution: We must first transform the equation system $\mathcal{E}$ to SRF. This is achieved by appending an extra equation $\nu Z = Y_1 \wedge Y_5$ to $\mathcal{E}$, and replacing the subformula $Y_1 \wedge Y_5$ in the equation for Y_3 by Z. The resulting equation system

is the following:

$$
\begin{aligned}
(\mu X_0 &= Y_0)\\
(\mu X_1 &= Y_1)\\
(\mu X_2 &= Y_2)\\
(\mu X_3 &= Y_3)\\
(\mu X_4 &= Y_4)\\
(\mu X_5 &= Y_5)\\
(\nu Y_0 &= Y_2 \wedge Y_3)\\
(\nu Y_1 &= X_0)\\
(\nu Y_2 &= Y_0 \wedge Y_4)\\
(\nu Y_3 &= X_2 \vee Z)\\
(\nu Y_4 &= Y_0)\\
(\nu Y_5 &= X_4)\\
(\nu Z &= Y_1 \wedge Y_5)
\end{aligned}
$$

Next, we transform the equation system to a Parity Game. All μ-equations will be represented by vertices with priority 1. The ν-equations are represented by vertices with priority 2. The vertices representing the conjunctive equations are owned by player Odd; all other equations are owned by player Even:



We first apply the recursive algorithm; this turns out to be the easiest algorithm to apply in this case. The least priority occurring in the game is 1. The set of vertices associated to this priority is the set $U = \{X_0, X_1, X_2, X_3, X_4, X_5\}$. Next, we compute the attractor set into for this set of vertices. Since we are considering vertices with an odd priority, we compute $Attr_\Box(U)$:

$$
\begin{aligned}
Attr^0_\Box &= U\\
Attr^1_\Box &= U \cup \{Y_1, Y_5\}\\
Attr^2_\Box &= (U \cup \{Y_1, Y_5\}) \cup \{Z\}\\
Attr^3_\Box &= (U \cup \{Y_1, Y_5, Z\}) \cup \{Y_3\}\\
Attr^4_\Box &= (U \cup \{Y_1, Y_3, Y_5, Z\}) \cup \{Y_0\}\\
Attr^5_\Box &= (U \cup \{Y_0, Y_1, Y_3, Y_5, Z\}) \cup \{Y_2, Y_4\}
\end{aligned}
$$

Since $Attr_{\Box}(U) = Attr^{5}_{\Box}(U)$ contains all vertices in the game, the recursive call to the recursive algorithm takes the empty game as its argument, immediately returning $(\emptyset, \emptyset)$. As a result, the original game has as its solution that all vertices are won by player Odd, and no vertex is won by player Even. In terms of the original equation system $\mathcal{E}$, this means that all equations have solution false.

For the SPM algorithm, we observe that we are considering measures in the set $\mathbb{M}^{\top}_{\mathcal{G}} = (\{0\} \times \{i \mid i \leq 6\} \times \{0\}) \cup \{\top\}$. Let $\mu_0 = \lambda v{:}V.(0,0,0)$. The strategy, updating the measures of the vertices in the order $X_0, X_2, Y_1, Z, Y_3, Y_0, Y_2$ leads to the measures $\top$ after 49 (= 7×(6+1)) applications of Lift. Once the measures for these vertices are $\top$, updating the measure in the following order: $Y_4, X_4, Y_5, X_5, X_3, X_1$ leads to measure $\top$ for all vertices. The number of applications Lift then totals to 55.

□