COMP3153/9153
Homework 3

Fixed Points, Abstraction Refinement, BDD's

Due: April 17, 2020, 10am
Submission guidelines are given at the end of this document.

Exercise 1 (Abstraction Refinement) (30 Marks)

For Questions 1 and 2 consider the transition system M in Figure 1. Each state represents the three Boolean variables a, b, c.

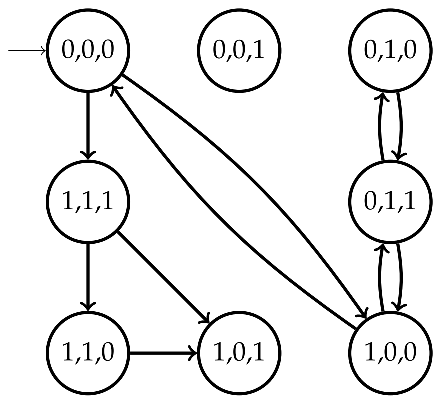


Figure 1: Transition System M

For Questions 3 and 4 consider the transition system TS and its abstraction TS' in Figure 2. Each state representing the three Boolean variables x, y, z.

Question 1 Assume two predicates p_1 and p_2 with

$$p_1 \equiv (a = 1) \wedge (b = 1 \vee c = 1),$$

$$p_2 \equiv (a = 0 \wedge b = c)\ .$$

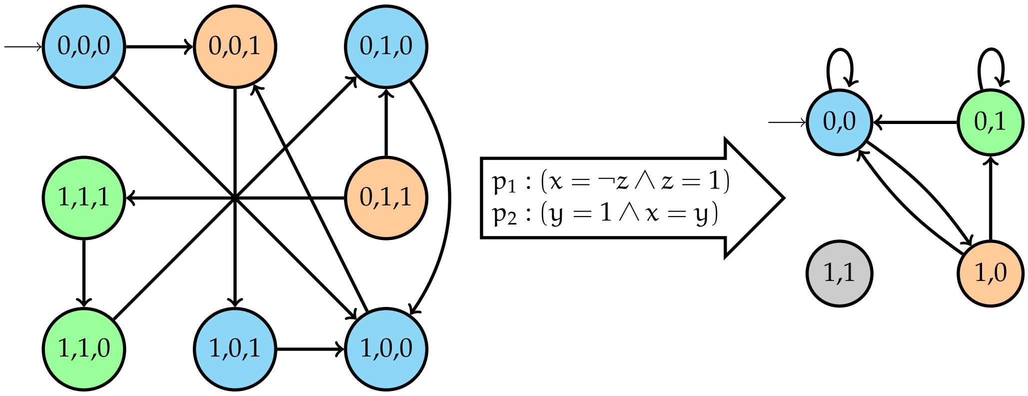


Figure 2: Transition System TS and Abstraction TS′

These predicates can be used to define an abstraction (two states in M are equivalent if the yield the same values for p_1 and p_2). Construct the resulting abstract transition system M' using p_1, p_2.

Some Definitions A labeling L' of an abstraction M' is *consistent* with the abstraction function α if $L(s) = L'(\alpha(s))$, for all states s.

A labeling L' is *consistent* with predicate p if $L_p(s) = L'_p(\alpha(s))$, for all s, where $L_p(s) =$ TRUE for states satisfying p, and FALSE otherwise.

Intuitively the former says that "an abstract state has same labels as each of its concrete states". The latter means that "an abstract state satisfies exactly the same predicates as each of its concrete states".

Question 2 Is the abstraction M'

1. consistent with p_1,
2. consistent with $q_1 \equiv a = b$,
3. consistent with $q_2 \equiv (a \vee b) \wedge (b \vee c)$?

For each case either give an example why it is not consistent or give the labeling showing consistency.

Question 3 Give an ACTL formula that is true in TS, but not in TS′. Explain why by giving the spurious counter-example.

You can assume that states are labelled with atomic propositions according to their colour in Figure 2, i.e., Blue, Green, Orange and Grey. A Blue state is for instance (0,0,0) in TS, or (0,0) in TS′ etc.

Question 4 Give the refinement TS″ of TS′ using the additional predicate $p_3 \equiv y \neq z$.

Exercise 2 (Data Flow Analysis) (25 Marks)

This exercise is about Available Expression Analysis of source code. Consider the following pseudo-code in a simple WHILE language:

```
[a:=x+b]^1;
[y:=a*b]^2;
while [y>a+b]^3 do (
    [x:=a+1]^4;
    [y:=a+b]^5
    if [(a+1 > a+b)]^6
            then [a:=x+1]^7;
            else [x:=a+1]^8;
);
```

Question 5 Give the Control Flow Graph for the above program.

Question 6 Give the table of `gen` and `kill` sets for each statement in the program.

Question 7 Give the data flow equation for each node's entry and each node's exit(s). You may find it helpful to give the condition node 6 has two different exits (AE_{exit_true} and AE_{exit_false}).

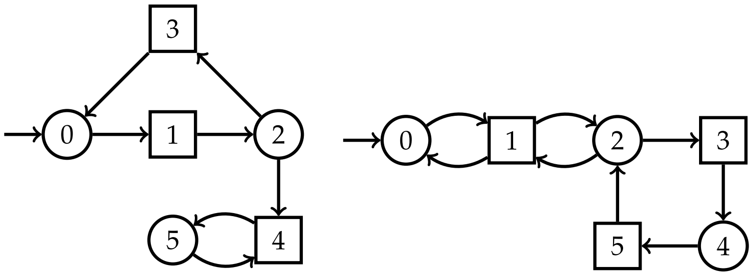
Question 8 Compute the least fixed point of the equation set, i.e., the result of the available expression analysis:

1. Give the resulting table AE_{entry}, AE_{exit} for all nodes.
2. Give the sequence of equations you evaluated to get to the fixed point.

   Note: for the last question above, you can give each equation in this Question 7 a number. Then just give the sequence of those numbers to enable someone else to reproduce your results by following the same evaluation steps.

Exercise 3 (Fixed Points) (30 Marks)

The automata in Figure 3 depict two *two-player games* G_1 and G_2. The round-shaped nodes, set S_1, belong to Player 1 (P_1) and the square-shaped nodes, set S_2 to Player 2 (P_2). The game is played by moving a token from one node to another according to the following rule: when a token is in an S_i-node q for $i \in \{1, 2\}$, Player i chooses a target state q' of an outgoing edge from q, and the token moves to q'. We assume that the set of edges is $E \subseteq (S_1 \times S_2) \cup (S_2 \times S_1)$, which implies the Players alternate turns. The initial node of the game is an S_1-node, 0, and by moving the token infinitely often according to the rule the players build infinite sequences of nodes in $(S_1.S_2)^\omega$. The objective for Player 1



(a) Game G_1 (b) Game G_2

Figure 3: Automata for Exercise 4

is either i) to *force* the game in a designated set of states (*reachability game*) or ii) to *avoid* a set of bad states (*safety game*). Of course the Players do not co-operate and Player 2's objective is to prevent Player 1 from achieving its objective.

A *strategy* π_1 for Player 1 is a mapping from $S_1.(S_2.S_1)^*$ to S_2, i.e., if the token is in an S_1-node, π_1 gives the choice of the next node. The choice should be allowed in the current node of the game: if the history of the game is $w.q$ with $q \in S_1$, then $\pi_1(w.q)$ must be a target of an outgoing edge from q. Given a strategy π_1 for Player 1, a π_1 *outcome* is an infinite sequence $q_0 q_1 q_2 \cdots q_n \cdots \in (S_1.S_2)^\omega$ such that for $i \geqslant 0$, $(q_i, q_{i+1}) \in E$ and $q_{2i+1} = \pi_1(q_0.q_1.\cdots.q_{2i})$ (Player 1 plays according to the strategy π_1). Notice that they may be more than one π_1 outcome as Player 2's choices are not constrained.

We first consider *reachability games*. The objective for Player 1 is to force the game into a set of nodes $G \subseteq S_1 \cup S_2$. We want to decide whether Player 1 has a strategy π_1 such that every π_1 outcome contains a node in G. In this is the case π_1 is a *winning* strategy.

Question 9 In Game G_1 (Figure 3(a)), is there a winning strategy for Player 1 to force node 3? If yes, give a winning strategy.

Question 10 The Pre operator is defined on sets of nodes $Y \subseteq S_1 \cup S_2$ by:

$$\mathsf{Pre}(Y) = \{q \mid \exists (q, q') \in E, q' \in Y\}.$$

We let $\overline{X} = (S_1 \cup S_2) \setminus X$ be the complement of X in $S_1 \cup S_2$. Given $X \subseteq S_1 \cup S_2$, we define f by:

$$f(X) = (G \cup X) \cup (S_1 \cap \mathsf{Pre}(X)) \cup \left(S_2 \cap \overline{\mathsf{Pre}(\overline{X})}\right)$$

Show that f is monotone.

Question 11 Let $\mathcal{W}$ be the set of winning nodes in a game, i.e., the nodes from where Player 1 has a winning strategy. Show that $\mathcal{W}$ is a fixed point of f.

Question 12 We know that $\mathcal{W}$ is either the least fixed point or the greatest fixed point. Still we have to find the good choice. Using G_1, how can you decide whether $\mathcal{W} = \mu X.f(X)$ or $\nu X.f(X)$?

Question 13 Using the results of Questions 10–12, propose an algorithm to compute the winning nodes of finite games. Give the result of your algorithm for G_1.

We now focus on *safety* games. We want to decide whether Player 1 has a strategy π_1 such that every π_1 outcome does not contain a node in set B (B is the set of *bad* nodes we want to avoid). In this is the case π_1 is a *winning* strategy.

Question 14 In Game G_2 (Figure 3(b)), does Player 1 have a strategy to win the safety game with $B = \{5\}$?

Question 15 Propose a fixed point characterisation in the form $X = h(X)$ of the set of winning nodes $\mathcal{W}'$ for safety games.

Question 16 Assuming the set of winning nodes $\mathcal{W}'$ is either the least fixed point or the greatest fixed point of the characterisation $X = h(X)$ you proposed before, using the result derived above, decide $\mathcal{W}' = \nu X.h(X)$ or $\mathcal{W}' = \mu X.h(X)$.

Question 17 Using the results of Questions 15–16, propose an algorithm to solve finite safety games and give the result of your algorithm for G_2.

Exercise 3 (Binary Decision Diagrams) (25 Marks)

Question 18 Consider the boolean function

$$f_2(\overline{x}) = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$$

Give the binary decision tree for f_2 for the order o of the variables $x_1 < x_2 < x_3 < x_4$ i.e., x_1 is at the top of the tree.

Question 19 Give the ORBDD for f_2 with the previous order o (x_1 at the top).

Question 20 Now consider the order o' defined by $x_1 < x_3 < x_2 < x_4$. Give the ORBDD for f_2 with this order o'.

Question 21 Consider the function

$$f_n(x_1, x_2, \cdots, x_{2n-1}, x_{2n}) = \bigwedge_{i=1}^{n} (x_{2i-1} \vee x_{2i})$$

Let o_n be the ordering $x_1 < x_2 < \cdots < x_{2n-1} < x_{2n}$. Prove that the number of nodes in the ORBDD for f_n with the order o_n is $2n + 2$.

Question 22 Now consider the order o'_n defined by $x_1 < x_3 < \cdots < x_{2n-1} < x_2 < x_4 < x_{2n}$ (from top to bottom: first the odd indices and then the even ones). Prove that the number of nodes in the ORBDD for f_n with the order o'_n is 2^{n+1}.

Submission Guidelines

- Due time: April 17, 2020, 10am. **No late submission allowed**.
- Submit one PDF file (`hw3.pdf`) using the CSE `give` system by typing the command `give cs3153 hw3 hw3.pdf` on a CSE terminal. Alternatively use the online submission page.
- It is highly recommended that you use LaTeX to prepare your document. A guide is provided on the course website.
