Practice Exam – Sample Solutions

Johannes Kloos

July 17, 2014

How to read these sample answers: I will give answers in two parts. First, I will write down the answer in the way I would give it in an exam – in this way, you can see what our expectations are. Sometimes, I will add some more explanation afterwards; I would not require this in an exam, but it may help you understand how I found the answer.

Problem 1.a.

You are given a (reduced ordered) BDD b. How can you test in constant time whether the formula represented by b is satisfiable?

Sample answer: We know that every satisfying assignment of the BDD corresponds to a path from the root node to $\boxed{1}$. So, if the BDD is unsatisfiable, there are only paths to $\boxed{0}$. But in that case, because it's reduced, the BDD would actually be exactly $\boxed{0}$. Therefore, to check if b is satisfiable, we check if it's not $\boxed{0}$.

Problem 1.b.

Suppose you are performing predicate abstraction with n predicates. We said that the number of states in the abstraction is finite. How many states can there be at most in this finite abstraction?

Sample answer: The states in a predicate abstraction are given as the sets of predicates that hold. Since the set of predicates has size n, the number of subsets of this set is 2^n, meaning there can be at most 2^n different states.

Comments: There is another definition of predicate abstraction, namely boolean abstraction, where the states are given as Boolean combinations of the predicates. Since there are n predicates, there can be at most as many different states as there are different Boolean combinations of n predicates. But this is the number of different Boolean formulas with n variables, so it's 2^{2^n}.

The sample answer is what follows from the description of predicate abstraction in the lecture. If you have arrived at the other answer, we will accept that as well.

Problem 1.c.

Let us define a new temporal modality $\varphi_1 \exists \mathcal{U}^* \varphi_2$ with the following semantics: A state s of a labelled transition system satisfies $\varphi_1 \exists \mathcal{U}^* \varphi_2$ if there is a trace $s_0 \to s_1 \to \cdots \to s_k$ for some $k \geq 0$ such that $s_0 = s$ and s_k satisfies $\varphi_1 \mathcal{U} \varphi_2$. Show that you can express $\varphi_1 \exists \mathcal{U}^* \varphi_2$ as an STL formula.

Sample answer 1: Here's a picture of the situation we're trying to solve:

$$s_0 \to s_1 \to \cdots \to \underbrace{s_k \to \cdots s_{n-1}}_{\text{Here, } \varphi_1 \text{ holds}} \to s_n,$$

where φ_2 holds on s_n.

From the lecture, we know that $\exists \diamond \varphi$ means "in state s, there is a path from s to s', where φ holds on s'." This means that $\exists \diamond (\varphi_1 \mathcal{U} \varphi_2)$ describes what we want, and this can be expanded to $\mathsf{true} \exists \mathcal{U} (\varphi_1 \exists \mathcal{U} \varphi_2)$.

Sample answer 2: By looking at the definition of $\exists \mathcal{U}$, we see that $\varphi_1 \exists \mathcal{U} \varphi_2$ holds in state s if there is a path $s = s_0 \to \cdots \to s_n$ such that φ_1 holds on $s_0, \ldots, s_{n-1}$ and φ_2 holds on s_n. The definition of $\exists \mathcal{U}^*$ can then be read as:

$\varphi_1 \exists \mathcal{U}^* \varphi_2$ holds in s if there is a path $s = s_0 \to \cdots \to s_n$ and some k, $0 \leq k < n$, such that for $i = k, \ldots, n-1$, φ_1 holds in s_i, and φ_2 holds in s_n. But note that we can always choose $k = n - 1$, so that the condition reduces to: There is a path $s_0 \to \cdots \to s_n$ such that φ_2 holds in s_n. But this is exactly $\exists \diamond \varphi_2$.

Comments: This question actually has two valid answers. We would accept either answer as correct and give full points.

Problem 1.d.

We showed in class that two states are bisimilar iff they agree on all STL formulas. Extend that proof (you need only to write the new parts) to show that two states are bisimilar iff they agree on all CTL formulas.

Sample answer: The $\Leftarrow$ direction is clear, since all STL formulas are CTL formulas. For $\Rightarrow$, there's one new case, namely $\varphi = \exists \Box \varphi'$.

Let s, s' be bisimilar states, and assume that φ holds on s. Then there is an infinite path $s_0 \to s_1 \to \cdots$ with $s_0 = s$ such that φ' holds for every s_i. By bisimulation, we can construct a path $s'_0 \to s'_1 \to \cdots$ such that $s'_0 = s'$, all s'_i are bisimilar to the respective s_i, and, by the induction hypothesis, φ holds on all the s'_i. But this means that $\exists \Box \varphi$ holds on s'. If we assume that φ holds on s', by a similar argument, it also holds on s.

Problem 1.e.

Suppose you have a Boolean formula φ and you are interested in finding two distinct satisfying assignments to this formula. Show how you can get two distinct satisfying assignments by making at most two queries to a SAT solver.

Sample answer: Here's the algorithm: Given φ. First, call the SAT solver on φ to get an assignment $\mathcal{A}_1$. If none exists, we're done. Otherwise, suppose that φ has the variables $x_1, \ldots, x_n$ and $\mathcal{A}_1$ assigns x_i to a_i. Define $\hat{x}_i$ to be x_i if $a_i = 1$ and $\neg x_i$ if $a_i = 0$. Then it's easy to see that $b := \bigwedge_{i=1}^{n} \hat{x}_i$ has exactly one satisfying assignment, namely $\mathcal{A}_1$. So, call the SAT solver again, with $\varphi \wedge \neg b$. If it returns UNSAT, we're done. Otherwise, we get a satisfying assignment $\mathcal{A}_2$. But since $\mathcal{A}_2$ satisfies $\neg b$, we know that it can't be $\mathcal{A}_1$.

Comments: The formula b is known as a *blocking clause*. This method is used to implement model enumeration in SAT solvers.

Problem 2.1.

Consider an invariant verification problem with a formula $A(x)$ defining the initial states, a formula $B(x)$ defining the bad states and a transition relation $T(x, x')$. Suppose there is a formula $I(x)$ with the following properties:

1. $A(x) \implies I(x)$,
2. $I(x) \wedge B(x)$ is unsatisfiable, and
3. $\mathsf{Post}(I(x)) \implies I(x)$.

Argue that there is no trajectory that starts from the initial states and ends up in a bad state.

Sample answer: We show that if $s \to \cdots \to s'$, and $A(s)$ (i.e., s is an initial state), then $I(s')$. Point (2) then gives us that $B(s')$ does not holds, so s' is not a bad state. Since this holds for all reachable states, we're done.

So, suppose $s = s_0 \to \cdots \to s_n$ for some $n \geq 0$, and $s_i \to s_{i+1}$ means $T(s_i, s_{i+1})$ is satisfied. We do induction on n.

For $n = 0$, we have $s_n = s$, so the claim follows by point (1). For $n > 0$, we have by IH that $I(s_{n-1})$. Since $s_{n-1} \to s_n$, we have that $s_n \in \mathsf{Post}(I(x))$. By point (3), this implies $I(s_n)$.

Comments: I is an inductive invariant. The key idea in the proof is to look at possible violating traces and see that every trace stays inside I.

Problem 2.2.

Let $\mathcal{K} = (S, \to)$ be a transition system and let $s_0 \in S$ be an initial state. A transition $s \to t$ is called reachable is s is reachable from s_0. A set of transitions is called a *transition invariant* if it contains every reachable transition. Give an enumerative model checking algorithm to check that a given set of transitions is a transition invariant.

Sample answer: Algorithm:

Assume that I is the given set of transitions, given as a set of pairs (s, t).
$D := \varnothing$, where D is the set of seen states.
$Q := [s_0]$, a queue of states that need to be visited.
While Q is not empty:
  Take one state s from Q
  If $s \in D$, continue the loop.
  For every outgoing transition $s \to t$:
    If $(s, t) \notin I$, return FAIL
    Add t to Q
  Add s to D

It works like the default BFS enumerative model checking algorithm, only checking a transition condition instead of a state condition.

Comments: My approach here was to start with a standard enumerative model checking algorithm and to check what parts needed changing.

Problem 3.1.

Show that $(\mathbb{N}^k, \leq)$ is a well quasi order. You can use the observation that the Cartesian product of two wqos is a wqo.

Sample answer: We prove the claim by induction on k. If $k = 0$, $\mathbb{N}^k = \{()\}$ is the set containing only the empty vector, and the claim is trivial.

If $k = 1$, $\mathbb{N}^k = \mathbb{N}$. $\leq$ is obviously a pre-order. Now, let s be a sequence on $\mathbb{N}$. For all i, either $s_i \leq s_{i+1}$ or $s_i > s_{i+1}$. If $s_i > s_{i+1}$ for all i, we get a contradiction (see problem 4). Therefore, there's some i such that $s_i \leq s_{i+1}$. Thus, $\leq$ is a wqo.

If $k > 1$, we know that $(\mathbb{N}^{k-1}, \leq)$ is a wqo by IH, and that $(\mathbb{N}, \leq)$ is a wqo from above. By the hint, $(\mathbb{N}^{k-1} \times \mathbb{N}, \leq')$ is a wqo, where $(x, x') \leq' (y, y')$ iff $x \leq y$ and $x' \leq y'$. But that's exactly $(\mathbb{N}^k, \leq)$.

Comments: The hint implies that it will be useful to split $\mathbb{N}^k$ into parts, so we try induction and use the hint as the induction step. For the base cases, the easiest approach is to do the "simple cases" $k = 0$ and $k = 1$ explicitly; this pays off in the induction step here.

Problem 3.2.

Let $(S, \leq)$ be a wqo and let $U_0, U_1, \ldots$ be a sequence of upward closed sets such that $U_0 \subseteq U_1 \subseteq \ldots$. Prove that there is some i such that $U_i = U_{i+1}$.

Sample answer: Suppose, for a contradiction, that $U_0 \subsetneq U_1 \subsetneq \ldots$. Then for every i, we can choose some $x_i \in U_{i+1} \setminus U_i$.

By wqo, there are i, j with $i < j$ and $x_i \leq x_j$. By choice of the x_i, we have that $x_i \in U_{i+1}$, and because U_{i+1} is upward closed, also $x_j \in U_{i+1}$. But since $U_{i+1} \subsetneq U_{i+2} \subsetneq \cdots \subsetneq U_j$, we also have $x_j \in U_j$. Now, by choice of x_j, $x_j \notin U_j$ – contradiction.

Comments: In problems about well-quasi-orders and wellfounded relations, it often pays off to try to argue by contradiction and look at some infinite sequence s that does not have the required properties, e.g., $s_i \not\leq s_j$ for all $i < j$ in the case of wqo.

Problem 3.3.

Let $\mathcal{K} = (S, \to, \leq)$ be a WSTS, and U an upward closed sets of states. Prove or disprove: $\mathsf{Post}(U)$ is upward closed.

Sample answer: This is wrong. Here's a counter-example: Take $S = \mathbb{N}$, and define $a \to b$ to hold iff $b = 0$. Then $\mathsf{Post}(\mathbb{N}) = \{0\}$, which is not upward-closed. But $(S, \to)$ is a WSTS: $(\mathbb{N}, \leq)$ is a wqo, and we have

$$\mathrm{pre}(U) = \begin{cases} \varnothing & 0 \notin U \\ \mathbb{N} & 0 \in U \end{cases},$$

where both $\mathbb{N}$ and $\varnothing$ are upward closed.

Comments: Since the definition of WSTS explicitly uses the Pre set, it seems likely that the claim is incorrect. To construct a counter-example, I looked for a transition relation whose Post is never upward-closed. The easiest non-upward-closed set I could think of is $\{0\} \subseteq \mathbb{N}$, and this gives the transition relation. Proving that $(\mathbb{N}, \leq, \to)$ is a WSTS is routine.

Problem 4.1.

Is the relation $>$ on natural numbers well-founded? What about $>$ on integers? (Give a short justification or a counterexample in each case).

Sample answer: $>$ on $\mathbb{N}$ is well-founded: Note that an infinite sequence $a_0 > a_1 > \cdots$ gives rise to an infinite set $\{a_i \mid i \geq 1\}$ where $a_i < a_0$ and all a_i are distinct. But there can be at most a_0 numbers in that set – contradiction.

$>$ on $\mathbb{Z}$ is not well-founded: $0 > -1 > -2 > -3 > \cdots$.

Problem 4.2.

Let $S = (X, \to)$ be a system with set of states X and transition relation $\to$. Let $x_0 \in X$ be a state of S, and *Reach* the set of states reachable from x_0. We say S *terminates* from initial state x_0 is there is no infinite sequence of states $x_0 \to x_1 \to \cdots$.

Show that S terminates from x_0 if $\to \cap (\mathit{Reach} \times \mathit{Reach})$ is well-founded.

Sample answer: Suppose $x_0 \to x_1 \to \cdots$ is an infinite sequence of states. Clearly, $x_i \in \mathit{Reach}$ for all i. Write $\to' := \to \cap \mathit{Reach} \times \mathit{Reach}$. Then $x_0 \to' x_1 \to' \cdots$. But this gives us an infinite sequence, so $\to'$ is not well-founded.

Problem 4.3.

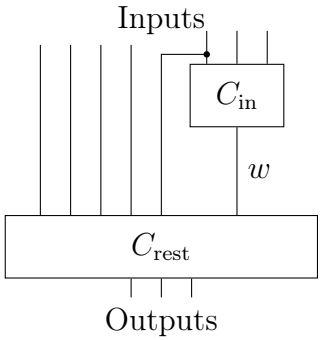
A *ranking function* is a map $r : X \to \mathbb{N}$ such that whenever $s \to t$, we have $r(s) > r(t)$. Show that if we can define a ranking function for S, then S terminates from every initial state.

Sample answer: Suppose $s_0 \to s_1 \to \cdots$ is an infinite execution. Then $r(s_0) > r(s_1) > \cdots$. But this gives us an infinite descending sequence of natural numbers. By 3.1, $\mathbb{N}$ is well-founded, so this is not possible.

Problem 5.

Automatic test pattern generation.

Sample answer: Idea: We treat the value on a given wire as the output of a Boolean formula combining the inputs, and use a SAT solver to calculate inputs that should force that wire to have value 0 or 1.



To calculate the inputs for wire w, we transform the "input circuit" C_{in} that gives a value to that wire into a CNF φ using the Tseitin transform from the exercises. We name the input variables $x_1, \ldots, x_n$ and the output variable y.

To generate the input for $y = 0$, we SAT-solve $\varphi \wedge \neg y$, and to generate to input for $y = 1$, we SAT-solve $\varphi \wedge y$. It one of these formulas is not satisfiable, we have detected a stuck-at fault at the logic level. If we do this for every wire, we get all the inputs.

Polynomial time: The Tseitin transform takes linear time in the size of the input, and we call it k times. SAT solving is considered "almost free" (we treat it as constant-time for the analysis) and is called twice per wire.

Comments: The main problem here is to understand what the problem is actually about. The point is that we want to ensure that we can force wire w to carry both true and false. Since w is driven by a single combinatorial circuit, this reduces to forcing that circuit to output a given value, and since combinatorial circuits can be given by Boolean formulas, SAT solving seems like a promising approach.

Note that we could do better by using a smarter encoding and reusing the results from the Tseitin transformation. But under exam conditions, this answer is absolutely sufficient.