Prof. Dr. Andreas Podelski
Dominik Klumpp
Elisabeth Henkel

Hand in until December 7th, 2022
23:59 via ILIAS
Discussion: December 12th/13th, 2022

Tutorial for Cyber-Physical Systems - Discrete Models

Exercise Sheet 7

Exercise 1: Linear-Time Properties 7 Points

The goal of this exercise is to help you better understand the representation of properties as sets of traces, as well as the notion of satisfaction by a transition system.

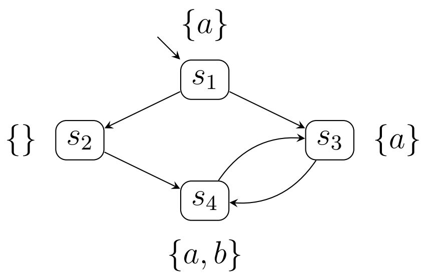
Assume $AP = \{a, b\}$. For each of the properties P_i, complete the following tasks:

(a) Formalize P_i as a set of traces using set comprehension.
For example: "always a" can be formalized as $\{A_0 A_1 A_2 \cdots \mid \forall i.\, a \in A_i\}$.

(b) Give an example of a trace that satisfies P_i.

(c) Give an example of a trace that does not satisfy P_i.

(d) State whether or not the transition system below satisfies P_i.



(P_1) Always (at any point of time) a or b holds.

(P_2) Always (at any point of time) a and b holds.

(P_3) b never holds before a holds.

(P_4) Every time a holds there will be eventually a point of time where b holds.

(P_5) At exactly three points of time, a holds.

(P_6) If there are infinitely many points of time where a holds, then there are infinitely many points of time where b holds.

(P_7) There are only finitely many points of time where a holds.

Exercise 2: Complement of LT-Properties 4 Points

This exercise is supposed to reveal some interesting (and possibly counter-intuitive) facts about LT properties and their complement.

Determine if the following statements hold for every trace $\tau \in (2^{AP})^\omega$, transition system T over AP and property $E \subseteq (2^{AP})^\omega$.

If a statement holds, give a proof. Otherwise give a counterexample.

(a) If $\tau \models \neg E$ holds, it follows that $\tau \not\models E$ holds.

(b) If $\tau \not\models E$ holds, it follows that $\tau \models \neg E$ holds.

(c) If $T \models \neg E$ holds, it follows that $T \not\models E$ holds.

(d) If $T \not\models E$ holds, it follows that $T \models \neg E$ holds.

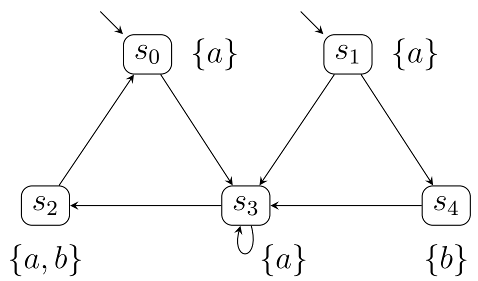
Notes:

- The negation of a property E is defined as $\neg E := (2^{AP})^\omega \setminus E$
- A trace τ satisfies a property E, $\tau \models E$ if and only if $\tau \in E$

Exercise 3: Invariant checking I 4 Points

In the lecture, you have seen an algorithm for invariant checking by forward depth-first search. This algorithm is displayed in algorithm 1.

Apply this algorithm to the following transition system whose set of atomic propositions is $AP = \{a, b\}$. The invariant Φ to be checked is the propositional logical formula a.



Whenever you iterate over a set of states, always take state s_i before state s_j if i is smaller than j.

Present the execution of the algorithm by writing down the contents of the set U and the stack π directly before every call to the function **DFS**.

Algorithm 1: DFS-based invariant checking

input : a finite transition system $\mathcal{T}$ and a propositional formula Φ
output: "yes" if $\mathcal{T} \models$ "always Φ", otherwise "no" and a counterexample

$U := \emptyset$; `// set of states`
$\pi := \varepsilon$; `// stack of states`
forall $s \in I$ **do**
  if DFS(s, Φ) **then**
    return("no", *reverse*(π)); `// path from` s `to error state`
  end
end
return("yes"); `//` $\mathcal{T} \models$ `''always Φ''`

---

function DFS(s, Φ)
  push(s, π);
  if $s \notin U$ **then**
    $U := U \cup \{s\}$; `// mark` s `as reachable`
    if $s \not\models \Phi$ **then**
      return("true"); `//` s `is an error state`
    else
      forall $s' \in Post(s)$ **do**
        if DFS(s', Φ) **then**
          return("true"); `//` s' `lies on a path to an error state`
        end
      end
    end
  end
  pop(π);
  return("false");
end

Exercise 4*: Invariant checking II 2 Bonus Points

The "DFS-based invariant checking" algorithm presented in algorithm 1 (and in the lecture) always computes a minimal counterexample (minimal in the sense that you cannot remove the last state). However, the algorithm does not necessarily compute a counterexample of minimal length (there might be two minimal counterexamples of different lengths). What is an example that shows that the counterexample that is returned does not always have minimal length? For this purpose, provide the following:

- A transition system that has three states s_0, s_1, s_2.
- An invariant.
- The counterexample with non-minimal length that is computed by the algorithm that uses the following strategy for iterating over a set of states: always take state s_i before state s_j if i is smaller than j.
- A counterexample of minimal length.