Master d'Informatique Fondamentale - 2016-2017
Sémantique et Vérification (M1)

ENS DE LYON

TD de Sémantique et Vérification

II– Linear Time Properties

Simon Castellan
`simon.castellan@ens-lyon.fr`

Exercise 1.

We are given three (primitive) processes P_1 , P_2 , and P_3 with shared integer variable x. The program of process P_i is as follows:

```
Process P_i:
for k_i = 1, ... ,10 do
    LOAD(x);
    INC(x);
    STORE(x);
```

That is, P_i executes ten times the assignment $x := x+1$. The assignment $x := x+1$ is realised using the three actions LOAD(x), INC(x) and STORE(x). Consider now the parallel program:

```
Parallel program P:
x := 0;
P_1 || P_2 || P_3;
```

Does P have an execution that halts with the terminal value $x = 2$?

Exercise 2.

Consider the following mutual exclusion algorithm that was proposed 1966 as a simplification of Dijkstra's mutual exclusion algorithm in case there are just two processes:

```
Dijkstra's algorithm for two processes:
    boolean array b = [0; 1];
    integer k = 1, i, j;
    /* This is the program for computer i, which may be either 0 or 1, computer
       j ≠ i is the other one, 1 or 0                                        */
C0: b(i) := false;
C1: if k ≠ i then
C2:     if ¬b(j) then goto C2;
        else k := i; goto C1;
    else critical section;
    b(i) := true;
    remainder of program;
    goto C0;
```

Here *C0*, *C1*, and *C2* are program labels, and the word "computer" should be interpreted as process.

1. Give the program graph representations for a single process. (A pictorial representation suffices.)
2. Give the reachable part of the transition system of $P_1 \parallel P_2$.
3. Check whether the algorithm indeed ensures mutual exclusion.

Exercise 3.

Consider the set AP of atomic propositions defined by $AP = \{x = 0, x > 1\}$ and consider a nonterminating sequential computer program P that manipulates the variable x. Formulate the following informally stated properties as LT properties:

1. false
2. initially x is equal to zero
3. initially x differs from zero
4. initially x is equal to zero, but at some point x exceeds one
5. x exceeds one only finitely many times
6. x exceeds one infinitely often
7. true

Exercise 4.

Each transition system TS (that probably has a terminal state) can be extended such that for each terminal state s in TS there is a new state s_{stop} , transition $s \to s_{stop}$ and s_{stop} is equipped with a self-loop, i.e., $s_{stop} \to s_{stop}$. The resulting "equivalent" transition system obviously has no terminal states.

1. Give a formal definition of this transformation $TS \mapsto TS^\star$
2. Prove that the transformation preserves trace-equivalence, i.e., show that if TS_1, TS_2 are transition systems (possibly with terminal states) such that $Traces(TS_1) = Traces(TS_2)$, then $Traces(TS_1^\star) = Traces(TS_2^\star)$.

Exercise 5.

Recall the definition of AP-deterministic transition systems. Let TS and TS' be transition systems with the same set of atomic propositions AP. Prove the following relationship between trace inclusion and finite trace inclusion:

1. For AP-deterministic TS and TS' :

$$Traces(TS) = Traces(TS') \text{ if and only if } Traces_{fin}(TS) = Traces_{fin}(TS').$$

2. Give concrete examples of TS and TS' where at least one of the transition systems is not AP-deterministic, but

$$Traces(TS) \not\subseteq Traces(TS') \text{ and } Traces_{fin}(TS) = Traces_{fin}(TS').$$