Concurrency theory
Exercise sheet 8

TU Braunschweig

Roland Meyer, Elisabeth Neumann

Winter term 2018/19

Out: January 9 Due: January 16

Submit your solutions until Wednesday, January 16, 12:00 am. You may submit in groups up to three persons.

Exercise 1: Sequential consistency

In the memory model **SC (sequential consistency)**, we assume that access to the main memory is atomic. More formally, the transition relation $\rightarrow_{\mathsf{SC}}$ is defined similar to $\rightarrow_{\mathsf{TSO}}$, but the rule (STORE) is replaced by the rule (SCSTORE).

$$\text{(SCSTORE)}\ \frac{<\text{inst}> = \text{mem}[r] \leftarrow r', a = val(r), v = val(r')}{(pc, b, buf) \rightarrow_{\mathsf{SC}} (pc', val[a := v], buf)}$$

Note that the buffer will never be used, i.e. early reads and updates from the buffer never occur.

a) Explain the following statement and argue that it is true: There is a correspondence between all executions of a multi-threaded program under SC and the single execution of all single-threaded programs obtained by shuffling the source code of the threads.

b) Let *prog* be a program. We define fency(*prog*) as the program that we obtain from *prog* by inserting an mfence instruction directly after every store operation (i.e. mem[r] $\leftarrow$ r').

Argue whether the following statement is correct: The program *prog* executed under SC has the same behavior as fency(*prog*) does under TSO.

Here, you may use control-state reachability (see below) as a suitable definition for "having the same behavior".

Exercise 2: SC reachability

The (control-state) reachability problem for SC is defined as follows.

SC-Reachability

Given: Program *prog* over DOM, program counter *pc*

Decide: Is there a computation $cf_0 \rightarrow^*_{\mathsf{SC}} (pc, buf, val)$ for some *buf*, *val*?

a) Reduce SC-Reachability to Petri net coverability. Explain which places are needed by the net, and how each instruction in the program can be simulated by Petri net transitions.

b) Conclude that SC-Reachability can be solved in PSPACE. Here, you may assume that the size of DOM is encoded in unary.
Hint: Reachability in 1-safe Petri nets is PSPACE-complete.

Exercise 3: Bounded round reachability

Describe the general case for the bounded round TSO-reachability problem that was described in the lecture. Let P be a parallel program with $n \in \mathbb{N}$ threads and a bound $k \in \mathbb{N}$ on the number of rounds that each thread can make. Explain how to construct a program P' such that for each program counter pc in P and its equivalent program counter pc' in P', the following holds.

$$pc \text{ is TSO-reachable in } P \text{ iff } pc' \text{ is SC-reachable in } P'.$$

Note: You do not have to give a formal construction. It is sufficient to list the additional global variables needed, explain their meaning and how they are used by P'.