SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Jan Reineke
Sebastian Hahn, M.Sc.

Verification of Real-Time Systems SS 2015

Assignment 9

Deadline: July 2, 2015, before the lecture

Exercise 9.1: Block-Level Relative Competitiveness (3+4+10=17 Points)

Prove the following statements:

(a) For $k \geq l \geq 2$, $FIFO(k)$ is not (α, β)-block-miss-competitive relative to $LRU(l)$ for any α and β.

(b) For $log_2 k + 1 < l$, $PLRU(k)$ is not (α, β)-block-miss-competitive relative to $LRU(l)$ for any α and β.

(c) For $k \geq l$, $FIFO(k)$ is $\left(\frac{\lceil \frac{k}{l-1} \rceil - 1}{\lceil \frac{k}{l-1} \rceil}, 1\right)$-block-hit-competitive relative to $LRU(l)$. *Hint:* You might find it useful to partition an access sequence s into subsequences $s_1 s_2 \dots s_n s_{post} = s$ such that s_i ends with the i^{th} miss to a block b.

Exercise 9.2: Persistence Analysis (4 Points)

Prove local consistency of the *conditional must* analysis. The concretization function is given as

$$\gamma(cm) := \{c_1(b_1, h_1)c_2 \dots c_n \mid \forall i < n : c_{i+1} = upd(c_i, b_i) \wedge age(c_n, b_i) \leq cm(b_i)\}.$$

Exercise 9.3: Multi-Scope Persistence Analysis (1+4+2+(3)=7+(3) Points)

In the lecture, we mentioned that persistence analysis can be done *scope*-aware, i.e. a block is classified as persistent in a program scope (a part of the program) and not for the whole program.

(a) Provide a short example that illustrates the need of multiple scopes, i.e. one scope is not enough to classify all persistent blocks as persistent.

(b) Choose one type of persistence analysis. Extend the analysis to handle multiple scopes. Provide the lattice, update, join, and classification function. *Hint:* You might want to define the update function not only for accesses but also for scope entries and exit.

(c) Perform your new analysis on your example from (a).

(d) *Optional:* Give the concretization function for your analysis in (b). Can you use the concrete trace semantics we defined in the lecture? If not, how could you extend the concrete trace semantics to suit your concretization function?