Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 7 (due: 2.09.13) - 25 points

Exercise 1: **(2)**

Give a formal definition of $irt(i)$ and $Irt(i)$, which give the index of the i-th return statement in *c.pr* as well as the list of statements between the i-th and $i+1$-th return statement, including the i-th but not the $i+1$-th return statement, respectively.

Exercise 2: **(2)**

Give the configuration c^0 according to some C0 program P satisfying the context conditions in which the C0 computation begins.

Exercise 3: **(4)**

Recall that the element symbol is overloaded a couple of times in our lecture. Give the formal definitions of:

(a) (1 point) $e' \in e$ where $e', e \in L(E)$ and e' is a subexpression of e

(b) (1 point) $s \in f$ where $s \in L(St), f \in FN$ and s is a statement in the body of f.

(c) (1 point) $e \in f$ where $e \in L(E), f \in FN$ and e is an expression occurring in the body of f.

(d) (1 point) $x \in f$ where $x \in L(id), f \in FN$ and x is a local variable defined in f.

Exercise 4: **(2)**

Recall the following auxiliary functions, which are used to define the semantics of statements in a C0-configuration c where e is an expression:

$lv(e, c)$**:** The subvariable of e (only defined if e is a subvariable)

$etype(e, c)$**:** The type of e

$va(e, c)$**:** The value of e

Give recursive definitions for $lv(e, c)$, $etype(e, c)$ and $va(e, c)$ when:

(a) (1 point) $e = e'.n$ and n is a field of $etype(e', c)$

(b) (1 point) $e = true$

(c) (1 point (bonus)) $e = null$.

Exercise 5: **(2)**

Prove that $inv\_conf(c) \wedge lv(e, c) \in SV(c) \wedge vtype(lv(e, c)) = etype(e, c)$ implies $inv\_expr(e, c)$

Exercise 6: **(2)**

Fill in the following program such that `fib(n)` returns the n-th fibonacci number modulo 2^{32}.

```
int fib(int n) {


    return
}
```

Then formally prove that your program is correct.

Exercise 7: **(2)**

Prove that $hd(c.pr) \in L(St)$ is invariant w.r.t. the execution of valid statements.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 7 (due: 2.09.13) - 25 points

Exercise 8: **(5)**

Execute the following sequence of statements, starting with configuration c. Keep track of only the things that change. For the program rest, only write the next statement in detail and use abbreviations otherwise. Make sure it is clear what your abbreviations abbreviate. Stop when the program rest is "return x". Configuration c is specified as follows: $c.rd = 0$, $c.st = main$, $c.nh = 0$, $c.ht(x) = \square$, $c.m()$, $c.pr =$ "x=pow(2);return x".

```
typedef struct{int exp; int val} pow_struct;
int pow(int base, int n) {
    pow_struct p;
    p = new pow_struct*;
    p.exp = 0;
    p.val = 1;
    while p.exp < n {
        p.exp = p.exp + 1;
        p.val = p.val * p.base
    };
    return p.val
}
```

Exercise 9: **(4)**

Recall the exercise where you had to prove or refute that a certain set is a tree region. Many of you used a property of tree regions but none of you proved it.

(a) (2 points) Prove that for a tree region A and a node $xs \circ x \in A$, we have both $xs \in A$ and $xs \circ 0, \ldots, xs \circ x - 1 \in A$.

(b) (1 point) Use that lemma to prove that if $xs \circ x \in A$ but $xs \notin A$, A is not a tree region.

(c) (1 point) Use that lemma to prove that if $xs \circ x \in A$ but $xs \circ 0 \notin A$, A is not a tree region.

(d) (1 point (bonus)) Refute: $\{\epsilon, 20, 21, 22, 23\}$ is a tree region.

(e) (1 point (bonus)) Refute: $\{\epsilon, 2, 20, 21, 22, 23\}$ is a tree region.

(f) (1 point (bonus)) Refute: $\{0, 1, 2, 20, 21, 22, 23\}$ is a tree region.