UFR-IMAG
Université Joseph Fourier

Programming Language and Compiler Design,
2010/2011

Marion Daubignard
Yassine Lakhnech
Laurent Mounier

Homework - Version B.

Exercise 1

We define the syntactic category of bits $B = \{0, 1\}$. By b we denote a meta-variable ranging over B. We define inductively a set of bitstrings BS by the following BNF:

$bs := b \mid 0\ bs\ 0 \mid 01\ bs\ 10$

where bs is a meta-variable ranging over the set of bitstrings BS. Of the following two statements, one is wrong and one is right. You get to tell which is which and justify your answers by either a proof (by induction) or a counter-example.

1. In the set of bistrings BS, all elements contain at least one 0.
2. Every bitstring in the set BS is a palindrome.

A palindrome is a sequence of symbols that reads the same from right to left or left to right (e.g. 'Rise to vote sir' is one, 1001 is one too).

The first statement is false. Indeed, the first rule provides that 1 belongs to BS, which is a counter-example.

The second statement is true, let's prove it by induction.
Basic case: the atoms of this set are bits in B. Moreover, 0 and 1 are indeed palindromes, so that the property holds for atoms.
Case of the rule 'if $bs \in BS$ then $0\ bs\ 0 \in BS$:
our induction hypothesis is that bs is a palindrome, and we have to show that $0\ bs\ 0$ is one too, which is true (I skip the details).
Case of the rule 'if $bs \in BS$ then $01\ bs\ 10 \in BS$:
our induction hypothesis is that bs is a palindrome, and we have to show that $01\ bs\ 10$ is one too, which is true (I skip the details).
Conclusion : we have proven by structural induction on the set BS that every element of BS is a palindrome.

Exercise 2

We consider the following program.

```
begin   var z1 := 43;
        var z2 := 5 * (z1 - 1);
        proc toto is z1 := 3 * (z2 + z1);
        begin var z2 := 1;
        proc toto is z1 := z2 + 45;
        proc q is call toto;
        call q;
        end
        call toto;
end
```

You have been presented three different semantics for the While language with blocks and procedures: one with dynamic links for variables and procedures, another with dynamic links for variables but static links for procedures, and finally one with static links for variables and procedures.
What values are associated to z_1 and z_2 at the end of this program according to each of the three semantics you know ? Justify your answer (you can either draw the tree or precise the state or the variable environment and the storage function after each ';').

Dynamic-dynamic semantics:

		σ, env_P
begin	var $z_1 := 43$;	$\sigma_1 = \sigma[z_1 \mapsto 43], env_P$
	var $z_2 := 5 * (z_1 - 1)$;	$\sigma_2 = \sigma_1[z_2 \mapsto 210] = \sigma[z_1 \mapsto 43, z_2 \mapsto 210], env_P$
	proc $toto$ is $z_1 := 3 * (z_2 + z_1)$;	$\sigma_2, env_P^1 = env_P[toto \mapsto (z_1 := 3 * (z_2 + z_1))]$
	begin var $z_2 := 1$;	$\sigma_3 = \sigma_2[z_2 \mapsto 1], env_P^1$
	proc $toto$ is $z_1 := z_2 + 45$;	$\sigma_3, env_P^2 = env_P^1[toto \mapsto z_1 := z_2 + 45]$
	proc q is call $toto$;	$\sigma_3, env_P^3 = env_P^2[q \mapsto call\ toto]$
	call q;	$\sigma_4 = \sigma_3[z_1 \mapsto 46], env_P^3$
	end	$\sigma_5 = \sigma_4[z_2 \mapsto \sigma_2(z_2)] = \sigma[z_1 \mapsto 46, z_2 \mapsto 210], env_P^1 *$
	call $toto$;	$\sigma_6 = \sigma[z_1 \mapsto 768, z_2 \mapsto 210], env_P^1$
end		σ, env_P

Ask yourself why in *, the procedure environment is env_P^1... It is because we apply the sequence rule to the body of the big block, and in the sequence rule, S_1 and S_2 are evaluated in the same environment!

Dynamic-static semantics:

		σ, env_P
begin	var $z_1 := 43$;	$\sigma_1 = \sigma[z_1 \mapsto 43], env_P$
	var $z_2 := 5 * (z_1 - 1)$;	$\sigma_2 = \sigma_1[z_2 \mapsto 210] = \sigma[z_1 \mapsto 43, z_2 \mapsto 210], env_P$
	proc $toto$ is $z_1 := 3 * (z_2 + z_1)$;	$\sigma_2, env_P^1 = env_P[toto \mapsto (z_1 := 3 * (z_2 + z_1), env_P)]$
	begin var $z_2 := 1$;	$\sigma_3 = \sigma_2[z_2 \mapsto 1], env_P^1$
	proc $toto$ is $z_1 := z_2 + 45$;	$\sigma_3, env_P^2 = env_P^1[toto \mapsto (z_1 := z_2 + 45, env_P^1)]$
	proc q is call $toto$;	$\sigma_3, env_P^3 = env_P^2[q \mapsto (call\ toto, env_P^2)]$
	call q;	$\sigma_4 = \sigma_3[z_1 \mapsto 46], env_P^3$
	end	$\sigma_5 = \sigma_4[z_2 \mapsto \sigma_2(z_2)] = \sigma[z_1 \mapsto 46, z_2 \mapsto 210], env_P^1$
	call $toto$;	$\sigma_6 = \sigma[z_1 \mapsto 768, z_2 \mapsto 210], env_P^1$
end		σ, env_P

Static-static semantics:

		env_V, sto, env_P
begin	var $z_1 := 43$;	$env_V^1 = env_V[z_1 \mapsto 1], sto^1 = sto[1 \mapsto 43], env_P$
	var $z_2 := 5 * (z_1 - 1)$;	$env_V^2 = env_V^1[z_2 \mapsto 2], sto^2 = sto^1[2 \mapsto 210], , env_P$
	proc $toto$ is $z_1 := 3 * (z_2 + z_1)$;	$env_V^2, sto^2, env_P^1 = env_P[toto \mapsto (z_1 := 3 * (z_2 + z_1), env_P, env_V^2)]$
	begin var $z_2 := 1$;	$env_V^3 = env_V^2[z_2 \mapsto 3], sto^3 = sto^2[3 \mapsto 1], , env_P^1$
	proc $toto$ is $z_1 := z_2 + 45$;	$env_V^3, sto^3, env_P^2 = env_P^1[toto \mapsto (z_1 := z_2 + 45, env_P^1, env_V^3)]$
	proc q is call $toto$;	$env_V^3, sto^3, env_P^3 = env_P^2[q \mapsto (call\ toto, env_P^2, env_V^3)]$
	call q;	$env_V^3, sto^4 = sto^3[1 \mapsto 46] = sto[1 \mapsto 46, 2 \mapsto 210, 3 \mapsto 1], env_P^3$
	end	$env_V^4 = env_V[z_1 \mapsto 1, z_2 \mapsto 2], sto^5 = sto[1 \mapsto 46, 2 \mapsto 210], env_P^1$
	call $toto$;	$env_V^4, sto^6 = sto[1 \mapsto 768, 2 \mapsto 210], env_P^1$
end		env_V, sto, env_P