

CAP - Exercises: semantics

Laure Gonnord

Sept. 2016

Abstract syntax

Recall the abstract syntax of the course for expressions :

$e ::=$	c	<i>constant</i>
	x	<i>variable</i>
	$e + e$	<i>addition</i>
	$e \times e$	<i>multiplication</i>
	\dots	

and the mini-while language :

$S(Smt) ::=$	$x := expr$	assign
	$skip$	do nothing
	$S_1; S_2$	sequence
	$\text{if } b \text{ then } S_1 \text{ else } S_2$	test
	$\text{while } b \text{ do } S \text{ done}$	loop

EXERCISE ► **Semantics of arithmetic expressions**

Show the two following properties :

1. Let $a \in \mathbf{Aexp}$ a given arithmetic expression. Let σ, σ' be two states. Show that if $(\forall x \in X, \sigma(x) = \sigma'(x))$, then $\mathcal{A}[a]\sigma = \mathcal{A}[a]\sigma'$.
2. Let $a' \in \mathbf{Aexp}$, show that :

$$\mathcal{A}[a[a'/x]]\sigma = \mathcal{A}[a]\sigma[x \mapsto \mathcal{A}[a']\sigma]$$

EXERCISE ► **Repeat**

We want to add the command **repeat** S **until** b to the mini-while language seen in the course.

1. Give semantics rules to define **repeat** S **until** b without using **while**.
2. Show that the constructions :
 - (a) **repeat** S **until** b and
 - (b) $S; \text{if } b \text{ then skip else (repeat } S \text{ until } b)$.are semantically equivalent.
3. Give a program transformation to transform any program with the **repeat** S **until** b construction into another one without this construction.