Definition of Security Properties – Exercises

Ana Matos
Software Security 2020/2021, IST

November 8, 2020

Aim To become familiar with basic notations and concepts for defining and understanding a semantics of a programming language, and for expressing and proving semantic properties of programs.

To gain intuitions on the gap between semantic and syntactic properties.

1 Noninterference, informally

Consider the definition of the property Input-Output Deterministic Noninterference, as given in classes. For each of the programs below, state whether they satisfy the property, and justify your answer rigorously.

1. $x := 42$
2. $x := 42; x := y$
3. $x := y; x := 42$
4. if $y = 0$ then $x := 42$ else $x := 42$
5. if $x = 0$ then $x := 42$ else $x := 4242$
6. $x := y$; if $x = 0$ then $x := 42$ else $x := 4242$
7. $x := 0$; while $y = 0$ do $x := 1$
8. $x := y$; while $x \neq 0$ do $x := x - 1$
9. $x := 0$; while $x \neq y$ do $x := x + 1$

2 Formal semantics

Consider the Semantics of Arithmetic Expressions, and the Natural Semantics (based on the big-step transition system), for the WHILE language, as was presented in class. The following exercises are closely based on examples and exercises that appear in Nielson & Nielson 1992.

1. Construct a derivation tree that allows to determine the result of evaluating the following statements, or explain why it does not exist:

   (a) $y := 1; \text{while } x \neq 1 \text{ do } (y := y \times x; x := x - 1)$
   on a memory such that x has the value 3.

   (b) $z := 0; \text{while } y \leq x \text{ do } (z := z + 1; x := x - y)$
   on a state where x has the value 17 and y has the value 5.

   Can you determine a state ρ such that the derivation tree obtained for the above statements does exist/not exist?

2. Consider the following programs.

   (a) while $x \neq 1$ do $(y := y * x; x := x - 1)$

   (b) while $1 \leq x$ do $(y := y * x; x := x - 1)$

   (c) while $1 = 1$ do skip

   For each statement determine whether or not it terminates for all choices of initial memory and whether or not it always loops. Try to argue for your answer using the axioms and rules of the semantics.

3 Semantic Properties

1. For each of the programs S in Question 1 of Section 1 above, say whether the following properties hold:

   i) there exist two memories ρ_1 and ρ_2 such that $\rho_1(x) = \rho_2(x)$, and $\langle S, \rho_1 \rangle \to \rho_1'$ and $\langle S, \rho_2 \rangle \to \rho_2'$, but $\rho_1'(x) \neq \rho_2'(x)$.

   ii) for all pairs of memories ρ_1 and ρ_2 such that $\rho_1(x) = \rho_2(x)$, we have that $\langle S, \rho_1 \rangle \to \rho_1'$ and$\langle S, \rho_2 \rangle \to \rho_2'$, implies $\rho_1'(x) = \rho_2'(x)$.

   Justify using the rules of the semantics.

2. Two statements S_1 and S_2 are said to be *semantically equivalent* when for all states ρ and ρ' we have that

$$\langle S_1, \rho \rangle \to \rho' \text{ if and only if } \langle S_2, \rho \rangle \to \rho'$$

   Show whether the following pairs of statements are semantically equivalent.

   (a) $S_1; S_2$ and $S_2; S_1$

   (b) $S_1; (S_2; S_3)$ and $(S_1; S_2); S_3$

   (c) (advanced) while t do S and if t then $(S; \text{while } t \text{ do } S)$ else skip

4 Approximating semantic properties

Consider the *High (H) - Low (L)* security policy for confidentiality, where the variable x has level L and variable y has level H, and the security property expressed in ii), of Question 1, Section 3.

1. The following *"Rules of thumb"* define a mechanism for accepting or rejecting programs of the WHILE language:

   - The "reading level" of an arithmetic expression or test is H if it contains a variable of level H. Otherwise it has level L.
   - You cannot assign an expression of "reading level" H to a variable of level L.

   (a) For each of the programs in Question 1 of Section 1 above, state whether rejecting programs according to the following *"Rules of thumb"* would produce false negatives (reject insecure programs), false positives (reject secure programs), with respect to (Deterministic Input-Output) Noninterference.

   (b) Is the above defined mechanism sound/complete/precise?

2. Repeat the previous question, but now consider the additional rules:

   - You cannot assign to a variable of level L, while in the branch of a conditional whose test has "reading level" H.
   - You cannot assign to a variable of level L, while in the body of a loop whose test has "reading level" H.

3. Write the analogous *"Rules of thumb"* for Integrity, and answer the corresponding questions above.