1 Control Flow Analysis for an object-oriented language

Program ::= Class* Exp
Class ::= class Id Var* Method* end
Var ::= var Id
Method ::= method Id ( Id* ) Exp end
Exp ::= Term
Term ::= Int | Exp Op Exp | false | true | Id := Exp |
       if Exp then Exp else Exp end |
       this | null | new Id | Exp.Id(Exp*)
Op ::= + | − | ∗ | & | < | =
Id ::= ⟨identifier⟩
Int ::= ⟨integer⟩

Consider the object-oriented mini-language defined above. It implements standard semantics, assuming the following rules:

- All variables are initialized with null.
- Assignments evaluate to the expression on the right-hand side.
- You may assume that all instance variables and formal arguments have distinct names. Further, this is never used outside classes; when used within a class C, it is renamed to this-C.

Define a constraint based 0-CFA for this language which determines for each expression to elements of which type(s) it might evaluate. Possible types are \( \text{Bool}, \text{Int}, \text{C} \in \text{CName}, \) where \( \text{CName} \) is the set of all classes defined in a program.

1. What are \( C(l) \) and \( r(x) \) in this setting?
2. Define for each kind of expression the set of constraints \( C \) it generates.
3. Consider the following type-incorrect program:

   \[
   \begin{align*}
   \text{class } & C \\
   \text{method } & n(i) \\
   & i+1 \\
   & \text{end} \\
   & \text{end} \\
   (\text{new } C).n(\text{true})
   \end{align*}
   \]

   Give the constraints that are generated for this program together with a minimal solution.
4. How can the results of the 0-CFA be used to reject programs which are not type-correct?

2 Correctness of 0-CFA

1. The following statement was crucial in the correctness proof for 0-CFA (cf. Slide 47 or Fact 3.11 on p. 160):

   \[
   \left( (\mathcal{C}, \tilde{p}) \models \text{it}^{l_1} \land \mathcal{C}(l_1) \subseteq \mathcal{C}(l_2) \right) \Rightarrow (\mathcal{C}, \tilde{p}) \models \text{it}^{l_2} \tag{1}
   \]

   Prove the statement formally.
2. Reconsider the decision to use $\overline{\text{Val}} = \mathcal{P}(\text{Term})$ in the correctness proof. Alternatively, we could have chosen $\overline{\text{Val}} = \mathcal{P}(\text{Exp})$. Show that the specification of the CFA may be modified accordingly, but that then the statement 1 above (and hence the correctness result) would fail.

Submission

- Deadline: 12.07.2010, 14:00, per mail to bieniusa@informatik.uni-freiburg.de, or on paper to Annette Bieniusa, Geb. 079, Room 000-14.
- Late submissions will not be marked.
- Do not forget to put your name on the exercise sheet.