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 | Id | Exp Op Exp | false | true | Id := Exp |
      if Exp then Exp else Exp end | Exp;Exp |
      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 \( \text{this-}C \).

Define a 0-CFA for this language which determines for each expression to elements of which type(s) it might evaluate. Possible types are \( \text{Bool} \), \( \text{Int} \), and \( C \in \text{CName}_* \), 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:

   class C
   method n(i)
      i+1
   end
   end

   (new C).n(true)

   Add labels and 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?

Solution

1. We define \( T = \{ \text{Int, Bool} \} \cup \text{CName}_* \) and \( r : \text{Var}_* \rightarrow \mathcal{P}(T) \), \( C : \text{Lab}_* \rightarrow \mathcal{P}(T) \).
2. The constraints could be defined as follows:
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):

   \[ (\widehat{C}, \widehat{\mu}) \models it^{l_1} \land \widehat{C}(l_1) \subseteq \widehat{C}(l_2) \] \[ \Rightarrow \ (\widehat{C}, \widehat{\mu}) \models it^{l_2} \]  

   \hspace{2cm} (1)

Similarly for all other binary operators.

3. The labeled program could look like this:

   ```plaintext
   class C
   method n(i)
      (i^2 + 1)\(^3\)
   end
   end
   
   (new C\(^4\).n(true\(^5\)))\(^6\)
   ```

   The constraints for this program are

   \[ r(i) \subseteq C(1) \]
   \[ \{\text{Int}\} \subseteq C(2) \]
   \[ \{C\} \subseteq C(4) \]
   \[ \{\text{Bool}\} \subseteq C(5) \]

   \[ C \in C(4) \Rightarrow C(5) \subseteq r(i) \]
   \[ C \in C(4) \Rightarrow C(3) \subseteq C(6) \]

   A minimal solution is given by:

   \[ C(1) = \{\text{Bool}\} \]
   \[ C(2) = \{\text{Int}\} \]
   \[ C(3) = \{\text{Int}\} \]
   \[ C(4) = \{C\} \]
   \[ C(5) = \{\text{Bool}\} \]
   \[ C(6) = \{\text{Int}\} \]
   \[ r(i) = \{\text{Bool}\} \]

4. If we annotate the program with the inferred type information, we could run a type checker. The type checker would then detect the type error in the sum.

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):

   \[ (\widehat{C}, \widehat{\mu}) \models it^{l_1} \land \widehat{C}(l_1) \subseteq \widehat{C}(l_2) \] \[ \Rightarrow \ (\widehat{C}, \widehat{\mu}) \models it^{l_2} \]  

   \hspace{2cm} (1)
Prove the statement formally.

2. Reconsider the decision to use $\widehat{\text{Val}} = \mathcal{P}(\text{Term})$ in the correctness proof. Alternatively, we could have chosen $\widehat{\text{Val}} = \mathcal{P}(\text{Exp})$. 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.

Solution

1. Proof by each case.

   - $[\text{con}]$ $(\widehat{C}, \widehat{\rho}) \models c^1$ always $\Rightarrow (\widehat{C}, \widehat{\rho}) \models c^2$
   - $[\text{var}]$ $(\widehat{C}, \widehat{\rho}) \models x^i \land \widehat{C}(l_1) \subseteq \widehat{C}(l_2) \Leftrightarrow \rho(x) \subseteq \widehat{C}(l_1) \subseteq \widehat{C}(l_2)$
     $\Rightarrow (\widehat{C}, \widehat{\rho}) \models x^i$
   - $[\text{fn}]$ $(\widehat{C}, \widehat{\rho}) \models (\text{fn} \ x \rightarrow e_0)^i \land \widehat{C}(l_1) \subseteq \widehat{C}(l_2) \Leftrightarrow (\text{fn} \ x \rightarrow e_0) \subseteq \widehat{C}(l_1) \subseteq \widehat{C}(l_2)$
     $\Rightarrow (\widehat{C}, \widehat{\rho}) \models (\text{fn} \ x \rightarrow e_0)^i$

   All other cases proceed similarly.

2. Define $\hat{v} \in \widehat{\text{Val}} = \mathcal{P}(\text{Exp})$.

   - $[\text{con}]$ $(\widehat{C}, \widehat{\rho}) \models c^i$ always
   - $[\text{var}]$ $(\widehat{C}, \widehat{\rho}) \models x^i \iff \widehat{\rho}(x) \subseteq \widehat{C}(l)$
   - $[\text{fn}]$ $(\widehat{C}, \widehat{\rho}) \models (\text{fn} \ x \rightarrow e_0)^i \iff \{ (\text{fn} \ x \rightarrow e_0)^i \} \subseteq \widehat{C}(l)$
   - $[\text{fun}]$ $(\widehat{C}, \widehat{\rho}) \models (\text{fun} \ f \ x \rightarrow e_0)^i \iff \{ (\text{fun} \ f \ x \rightarrow e_0)^i \} \subseteq \widehat{C}(l)$
   - $[\text{app}]$ $(\widehat{C}, \widehat{\rho}) \models (t_1^i \ t_2^j)^i \ 	ext{iff} \ (\widehat{C}, \widehat{\rho}) \models t_1^i \land (\widehat{C}, \widehat{\rho}) \models t_2^j$
     $\land (\forall (\text{fun} \ f \ x \rightarrow e_0)^j \in \widehat{C}(l_1) : (\widehat{C}, \widehat{\rho}) \models t_0^j \land \widehat{C}(l_2) \subseteq \widehat{\rho}(x) \land \widehat{C}(l_0) \subseteq \widehat{C}(l))$
     $\land (\forall (\text{fun} \ f \ x \rightarrow e_0)^j \in \widehat{C}(l_1) : (\widehat{C}, \widehat{\rho}) \models t_0^j \land \widehat{C}(l_2) \subseteq \widehat{\rho}(x) \land \widehat{C}(l_0) \subseteq \widehat{C}(l))$
     $\land (\text{fun} \ f \ x \rightarrow e_0)^j \subseteq \widehat{\rho}(f)$

   All other rules remain unchanged.

   For an example where statement 1 fails consider $it = (\text{fn} \ x \rightarrow e_0)$, and $ie_1 = it^1, ie_2 = it^2$.

   Assume, that $(\widehat{C}, \widehat{\rho}) \models ie_1$, i.e. $\{ (\text{fn} \ x \rightarrow e_0)^i \} \subseteq \widehat{C}(l_1)$. Now, choose $\widehat{C}(l_1) = \widehat{C}(l_2) = \{ (\text{fn} \ x \rightarrow e_0)^i \}$. Then, the condition of the statement holds but $(\widehat{C}, \widehat{\rho}) \models ie_2$ does not hold because $\{ ie_2 \} \notin \widehat{C}(l_2)$. 