Correlated-calls analysis: transfer functions

October 6, 2014

Given an IFDS problem P, our goal is to define the IDE problem $P^{\Subset}$ that considers correlated calls for P. Let N^* be P's control-flow supergraph, D the set of data-flow facts, and T the set of all types. To encode $P^{\Subset}$, we define the edge functions

$$\begin{gathered}\mathsf{EdgeFn}^{\Subset} = \{\mathsf{callStartEdges}^{\Subset},\, \mathsf{callReturnEdges}^{\Subset},\, \mathsf{endReturnEdges}^{\Subset},\\ \mathsf{otherSuccEdges}^{\Subset},\, \mathsf{otherSuccEdgesPhi}^{\Subset}\}.\end{gathered}$$

Each function $f^{\Subset} \in \mathsf{EdgeFn}^{\Subset}$ that defines $P^{\Subset}$ has a corresponding function $f \in \mathsf{EdgeFn}$ that defines P. For example, the corresponding function for $\mathsf{callStartEdges}^{\Subset}$ is $\mathsf{callStartEdges}$. Note that each f returns a set of data-flow facts $d \subset D$, whereas each $f^{\Subset}$ returns a set of pairs $p \subset D \times F$ of data-flow facts and micro functions.

Let $n_1,\, n_2 \in N^*$, $d_1 \in D \cup \Lambda$. The set $R^{\Subset}$ denotes the correlated-call receivers. For a given function $f^{\Subset} \in \mathsf{EdgeFn}^{\Subset}$, let

- $m(n)$ be the enclosing method of a node n;
- D_2 be the set of data-flow facts that is returned by $f(n_1,\, d_1,\, n_2)$[1];
- $R(n)$ be the set of correlated-call receivers declared in $m(n)$;
- $r(n)$ be the receiver `r`, if the node n corresponds to a call site `r.m()`;
- $\tau(n)$ be the static types corresponding to the receiver `r`, if n is a call site `r.m()`.

Recall that a micro functions in a correlated-calls analysis is a map from receivers $R^{\Subset}$ to update functions of type $T \to T$. Update functions are represented with intersection and union sets I and U. For example, the identity micro function

$$\mathsf{id}^{\Subset} = \langle \bot_T,\, \top_T \rangle$$

has the intersection set $\bot_T = T$ and union set $\top_T = \varnothing$. Therefore, the identity function should be interpreted as $\lambda\tau\,.\,\tau \cap \bot_T \cup \top_T$. The top micro function

$$\top^{\Subset} = \langle \top_T,\, \top_T \rangle,$$

and the bottom micro function

$$\bot^{\Subset} = \langle \bot_T,\, \bot_T \rangle.$$

In the following, the function id should be interpreted as follows: for the sets $S \subseteq R^{\Subset}$ and $F \subset T \to T$,

$$\mathsf{id}[\{(s, f) \,|\, s \in S,\, f \in F\}] = D_2 \times \{(s, f) \,|\, s \in S,\, f \in F\} \cup \left\{(r,\, \mathsf{id}^{\Subset}) \,|\, r \in R^{\Subset} \setminus S\right\}.$$

In other words, id maps all receivers of S to the corresponding functions in F, and all other receivers to identity micro functions. We will denote $\mathsf{id}[\varnothing]$ as id.

[1] Or by $f(n_1,\, d_1)$, if f is $\mathsf{otherSuccEdges}$ or $\mathsf{otherSuccEdgesPhi}$.

Using the above information, we can now define the correlated-calls-analysis edge functions:

$$\mathsf{callStartEdges}(n_1,\, d_1,\, n_2) = \begin{cases} \mathsf{id}\,[\{(r(n_1),\, \langle\tau(n_1),\, \top_T\rangle)\} \cup (R(n_2) \times \{\bot^{\Subset}\})] & \text{if } m(n_2) \text{ is not static and } r(n_1) \in R^{\Subset} \\ \mathsf{id}\,[R(n_2) \times \{\bot^{\Subset}\}] & \text{otherwise} \end{cases} \tag{1}$$

$$\mathsf{callReturnEdges}(n_1,\, d_1,\, n_2) = \begin{cases} \mathsf{id}\,[\{(r(n_1),\, \top^{\Subset})\}] & \text{if } m(n_2) \text{ is not static and } r(n_1) \in R^{\Subset} \\ \mathsf{id} & \text{otherwise} \end{cases} \tag{2}$$

$$\mathsf{endReturnEdges}(n_1,\, d_1,\, n_2) = \mathsf{id} \tag{3}$$

$$\mathsf{otherSuccEdges}(n_1,\, d_1) = \begin{cases} \mathsf{id}\,[R(n_1) \times \bot^{\Subset}] & \text{if } n_1 \text{ is a return instruction} \\ \mathsf{id} & \text{otherwise} \end{cases} \tag{4}$$

$$\mathsf{otherSuccEdgesPhi}(n_1,\, d_1) = \mathsf{id} \tag{5}$$

The definitions of the edge functions can be interpreted as follows.

(1) On a call-start edge, we set all variables in the target method to the set of all types $\bot^{\Subset}$. If the call is not static (i.e. it has a receiver), we map the receiver to its static-type set.

(2) Set the receivers to the empty set (TODO: this is wrong).

(3) We do not change anything on end-return edges. We need to set the local variables to $\bot^{\Subset}$, but we want to do this on the actual `return` nodes. Because of the way the IR works in WALA, we access the return nodes through otherSuccEdges.

(4) If we encounter a `return` node, we set all local variables of the exiting method to the set of all types. Otherwise, we do not change anything.

(5) We do not have to do anything special for phi nodes.