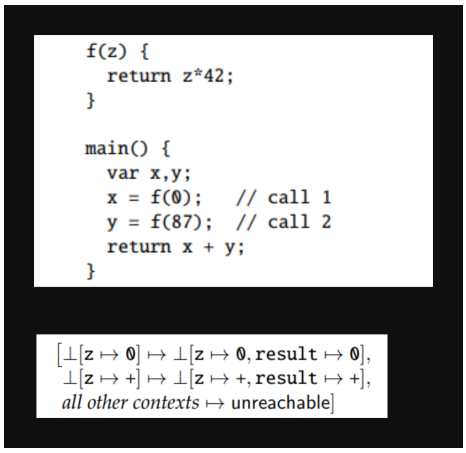
functional

Se modela el contexto como los posibles estados que puede tomar las variables.

Ejemplo: Los 2 llamados a la función definen el contexto. En este caso con los valores 0 y + (para sign analysis) el resto unreachable



Call node y function node

The constraint rule for an entry node v of a function $f(b_1, \ldots, b_n)$ and a call $w \in pred(v)$ to the function is the same as in the call strings approach, except for the condition on c:

$$s_w^{c'} \sqsubseteq [\![v]\!](c) \quad \text{where } c = s_w^{c'}$$

(The abstract state $s_w^{c'}$ is defined as in Section 8.3.) This rule shows that at the call w in context c', the abstract state $s_w^{c'}$ is propagated to the function entry node v in a context that is identical to $s_w^{c'}$. This makes sense because contexts are abstract states with the functional approach.

Parameter passing at function calls is modeled in the same way as in context-insensitive analysis, but now taking the call contexts into account. Assume w is a call node and v is the entry node of the function $f(b_1, \ldots, b_n)$ being called. The abstract state $s_w^{c'}$ defined by

$$s_w^{c'} = \begin{cases} \text{unreachable} & \text{if } [\![w]\!](c') = \text{unreachable} \\ \bot[b_1 \mapsto eval([\![w]\!](c'), E_1^w), \ldots, b_n \mapsto eval([\![w]\!](c'), E_n^w)] & \text{otherwise} \end{cases}$$

En palabras $S_w^{c'}$ es el estado formado a partir de los datos del nodo w en el contexto c'.

Dependiendo la elección del contexto, c' puede ser un call string (si k=1 seria simplemente un call node) o en el caso funcinal sería un estado que define valores de las variables de entrada de la función a la que pertenece el nodo w.

Si con dicho contexto c', w no es alcanzable entonces $[w](c') = unreachable$

After call node

Assume v is an after-call node that stores the return value in the variable X, and that v' is the associated call node and $w \in pred(v)$ is the function exit node. The constraint rule for v merges the abstract state from the v' and the return value from w, while taking the call contexts and reachability into account:

$$[\![v]\!](c) = \begin{cases} \text{unreachable} & \text{if } [\![v']\!](c) = \text{unreachable} \ \vee \ [\![w]\!](s_{v'}^c) = \text{unreachable} \\ [\![v']\!](c)[X \mapsto [\![w]\!](s_{v'}^c)(\texttt{result})] & \text{otherwise} \end{cases}$$

To find the relevant context for the function exit node, this rule builds the same abstract state as the one built at the call node.

Del mismo modo que en call string, el resultado se define en el mismo contexto del call node $[v](c) = [v'](c)[...]$ y usando el resultado de la función a partir de

dicho contexto del call node (del estado formado por el contexto del call node en este caso): $[w](S_w^{c'})(result)$