Taint Analysis

Let $O = \{A, B, C, ...\}$ and $V_t \subseteq var \times O$

$$\text{GEN}(var := expr) = \begin{cases} (var, o) & \text{for } \exists\ o \subseteq O : o \text{ in } expr \\ & \text{or } \exists\ o \subseteq V_t(var') : var' \text{ in } expr \end{cases}$$

$$\text{GEN}(\text{if}(expr)?t : e)) = \Big\{ \{(var, o)\} \text{ from } GEN(var := expr) \quad \text{for } \forall\ var : var \text{ in } t \vee var \text{ in } e, var := ...$$

$$\text{GEN}(I) = \{\emptyset\}$$

$$\text{KILL}(var := expr) = \begin{cases} (var, o) & \text{for } \exists\ o' \subseteq O : o' \text{ in } expr \wedge o' \neq o \\ & \text{or } \exists\ o' \subseteq V_t(var') :\ var' \text{ in } expr \wedge o' \neq o \\ & \text{or } \forall\ e \in expr : e \in INTS \end{cases}$$

$$\text{KILL}(\text{if}(expr)?t : e)) = \Big\{ \{(var, o)\} \text{ from } KILL(var := expr) \quad \text{for } \forall\ var : var \text{ in } t \vee var \text{ in } e, var := ...$$

$$\text{KILL}(I) = \{\emptyset\}$$

$$\text{JOIN}(S_1, S_2) = TRANSFER(S_1)\ \cup\ TRANSFER(S_2)$$

$$\text{TRANSFER}(S, I) = S - KILL(I)\ \cup\ GEN(I)$$

1 Worklist Algorithm

Algorithm 1: Worklist algorithm for Taint Analysis

```
   Input  : CFG cfg
   Output: Set of possible tainted variables
 1 Function analyze
 2 |   worklist.add(cfg.entry.get(0));
 3 |   while !worklist.empty do
 4 |   |   i = worklist.remove;
 5 |   |   itpt.put(i, {∅});
 6 |   |   for s : i.successors do
 7 |   |   |   worklist.add(s);
 8 |   |   end
 9 |   end
10 |   worklist.add(cfg.entry.get(0));
11 |   while !worklist.empty do
12 |   |   i = worklist.remove;
13 |   |   ptb = itpt.get(i);
14 |   |   pta = transfer(ptb, i);
15 |   |   for s : i.successors do
16 |   |   |   worklist.add(s);
17 |   |   |   pte = join(pta, itpt.get(s));
18 |   |   |   iptp.put(s, pte);
19 |   |   end
20 |   end
21 end
```