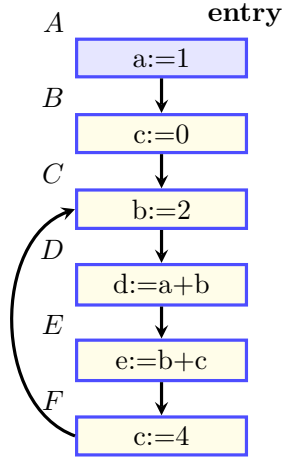


## Definitions

**Definition 1.1.** 一个有限有向图  $G = (N, E)$  由有限集合  $N$  和集合  $E \subseteq N \times N$  构成. 对定义的补充: 取  $N$  上两个顶点  $n_1$  和  $n_2$ , 若有一条  $n_1$  到  $n_2$  有向边  $e$ , 那么  $e \in E$ .

**Definition 1.2.** 一个程序图是一个有限有向图  $G = (N, E)$ , 其中  $N$  表示程序中的所有指令 (instruction) 构成的集合,  $E$  表示指令之间的控制流 (control flow) 构成的集合.

**Example 1.3.** 通篇将使用下面的例子来阐述.



**Definition 1.4.** **Constant pool**表示由一群有序对 (order pair)  $(v, c)$  构成的集合, 记为  $\mathbf{P}$ . 其中  $v$  表示程序中某个具体的变量,  $c$  表示某个常量.  $G$  上某顶点  $i$  处的 Constant pool 记为  $\mathbf{P}_i$ , 若  $(v, c) \in \mathbf{P}_i$ , 那么表示在程序动态运行时变量  $v$  可以取到常量  $c$ , 所以是可以存在  $(v, c_1), (v, c_2) \in \mathbf{P}_i$

**Definition 1.5.** 给定  $G$  上某个顶点  $i$  和从 entry 到  $i$  的一条路径  $\pi = (p_1, \dots, p_n)$ , 其中  $p_1$  为 entry,  $p_n$  为顶点  $i$ . 用  $\mathbf{P}_i^\pi$  表示**限制在路径  $\pi$  上  $i$  点的 constant pool**, 注意其含义是在路径  $\pi$  确定的情况下, 沿着  $\pi$  执行到顶点  $i$  处时, 这一点的程序状态是确定的, 所以若  $(v, c) \in \mathbf{P}_i^\pi$ , 则不会同时存在  $(v, c') \in \mathbf{P}_i^\pi$ .

**Example 1.6.** 以上图的  $D$  点举例, 如果我们关注的路径是  $\pi_1 = (A, B, C, D)$ , 那么

$$\mathbf{P}_D^{\pi_1} = \{(a, 1), (c, 0), (b, 2)\}.$$

如果我们关注的路径是  $\pi_2 = (A, B, C, D, E, F, C, D)$ , 那么

$$\mathbf{P}_D^{\pi_2} = \{(a, 1), (c, 4), (b, 2), (d, 3), (e, 2)\}.$$

**Definition 1.7.** 给定程序图  $G = (N, E)$  中的某个顶点  $i$ , 那么它的**Propagated constant pool**表示为

$$\mathcal{P}_i = \bigcap_{k \in K} \mathbf{P}_i^{\pi_k}.$$

其中  $G$  上总共有  $K$  条 entry 到  $i$  的路径.

**Definition 1.8.** 给定程序图  $G = (N, E)$ , 定义集合  $V$  表示  $G$  上所有的变量, 集合  $C$  表示图上所有的常量和集合  $U = V \times C$  表示可能出现在任意顶点上 constant pool 中 order pairs. 那么一个常量传播函数表示为

$$f: \bar{N} \times \mathfrak{P}(U) \rightarrow \mathfrak{P}(U).$$

给定某个具体顶点  $i$  和 constant pool  $\mathbf{P}$ , 若  $(v, c) \in f(i, \mathbf{P})$  当且仅当 (1) 在  $(v, c) \in \mathbf{P}$  且顶点  $i$  处没有对  $v$  进行赋值或者 (2) 对  $v$  赋值表达式的结果是常量  $c$ .

前面的 Propagated constant pool 的定义可以改写为

$$\mathcal{P}_i = \bigcap_{u \in F^i} u,$$

其中  $F^i = \{f(p_{k_n}, f(p_{k_{n-1}}, \dots, f(p_1, \mathbf{P}_\varepsilon)) \dots), \dots\}$ ,  $(p_{k_1}, \dots, p_{k_n})$  是一个 entry  $\varepsilon$  到  $i$  的路径  $\pi_k, k \in K$ ,  $\mathbf{P}_\varepsilon$  表示 entry 处的初始化的 constant pool.

**Example 1.9.** 例如在上图  $A$  点的 constant pool 是一个  $\emptyset$ , 那么

$$f(A, \emptyset) = \{(a, 1)\}.$$

如果我们把得到的结果当做  $B$  点的 constant pool, 那么就有

$$f(B, f(A, \emptyset)) = \{(a, 1), (c, 0)\}$$

**Definition 1.10.** 一个 semilattice (半格)  $\mathcal{S}$  由一个非空集合  $S$  和一个二元运算  $\cdot$  构成, 其中  $\cdot$  需要满足下面的条件

1.  $x \cdot x = x$ ,
2.  $x \cdot y = y \cdot x$ ,
3.  $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ .

通常我们把  $\cdot$  用  $\wedge$  或者  $\vee$  来表示.

**Definition 1.11.** 给定一个 semilattice  $\mathcal{S}$ . 在其上构造一个 partial order set, 我们定义若  $x \wedge y = x$ , 则  $x \leq y$ , 这个特殊的 poset (后面用它代称 partial order set) 我们称之为 meet semilattice. 对偶地定义若  $x \vee y = y$ , 则  $x \leq y$ , 这个 poset 我们称之为 join semilattice.

**Definition 1.12.** 给定一个 meet semilattice  $\mathcal{S}$ , 且  $\mathcal{S}$  里面有一个最大元 1 (maximum element), 即对于任意的  $x \in \mathcal{S}$ , 都有  $x \wedge 1 = x$ . 我们称  $\mathcal{S}$  为一个 finite meet semilattice. 同理给定一个 join semilattice 里面有一个最小元 0 (minimum element), 则称其为 finite join semilattice.

**Lemma 1.13.** 下面等式成立 homomorphism???

$$f(i, x \wedge y) = f(i, x) \wedge f(i, y).$$

其中  $y, u \in \mathfrak{P}(U)$ .

证明. 分四种情况来分别说明

1. 若  $(v, c) \in x$  和  $(v, c) \in y$ . 那么  $(v, c) \in x \wedge y$ , 若  $(v, c) \in f(i, x \wedge y)$ , 则满足  $f$  定义提到的条件 (1)(2), 那么在满足 (1)(2) 的前提下有  $(v, c) \in f(i, x)$  和  $(v, c) \in f(i, y)$ , 即  $(v, c) \in f(i, x) \wedge f(i, y)$ . 同理若  $(v, c) \notin f(i, x \wedge y)$ , 也可以得到  $(v, c) \notin f(i, x) \wedge f(i, y)$ .
2. 若  $(v, c) \in x$  和  $y$  里面没有关于  $v$  的 variable-constant pair. 同下
3. 若  $x$  里面没有关于  $v$  的 variable-constant pair 和  $(v, c) \in y$ . 同下
4. 若  $(v, c_1) \in x$  和  $(v, c_2) \in y$ . 那么  $(v, c_1) \notin x \wedge$  且  $(v, c_2) \notin x \wedge y$ . 若  $(v, c_3) \in f(i, x \wedge y)$ , 则满足  $f$  定义中条件 (2). 那么满足 (2) 的前提下, 有  $(v, c_3) \in f(i, x)$  和  $(v, c_3) \in f(i, y)$ , 即  $(v, c_3) \in f(i, x) \wedge f(i, y)$ . 若  $f(i, x \wedge y)$  中没有关于  $v$  的 variable-constant pair, 那么  $f$  中没有对  $v$  进行重新赋值, 则  $(v, c_1) \in f(i, x)$  和  $(v, c_2) \in f(i, y)$ , 即  $f(i, x) \wedge f(i, y)$  里面也没有关于  $v$  的 variable-constant pair.

由于我们选取的  $i$  和  $x, y$  都是任意的, 综上原式成立. □

**Definition 1.14.** 给定有向图  $G = (N, E)$  上一个顶点  $i$ , 若  $s \in N$  且  $(i, s) \in E$ , 则  $s$  为  $i$  的立即后继(immediate successor), 把  $i$  的所有立即后继记为  $IS(i)$ ; 同理若  $p \in N$  且  $(p, i) \in E$ , 则  $p$  为  $i$  的立即前驱(immediate predecessor), 把所有的立即前驱记为  $IP(i)$ .

## Algorithm

---

**Algorithm 1:** global analysis

---

```
input  : A program graph  $G = (N, E)$ 
output The propagated constant pool  $\mathcal{P}_i$  of every node  $i$ 
:
1 begin
    /* 我们只关注整个图只有一个 entry  $\varepsilon$ . 经典 worklist 的应用 */
2    $W \leftarrow \{(\varepsilon, \mathbf{P}_\varepsilon)\};$ 
    /* 所有顶点的 propagated constant pool 初始化为整个 lattice 里面的最大值 */
3   foreach  $i$  in  $N$  do
4        $\mathcal{P}_i = 1$ 
5   while  $W \neq \emptyset$  do
6        $X \leftarrow (i, \mathbf{P}_i) \in W;$ 
7        $W \leftarrow W - \{(i, \mathbf{P}_i)\};$ 
        /* 这里做 meet 是很自然的 */
8       if  $X \wedge \mathcal{P}_i < \mathcal{P}_i$  then
9            $\mathcal{P}_i \leftarrow X \wedge \mathcal{P}_i;$ 
            /* 考虑所有后继 */
10           $W \leftarrow W \cup \{(s, f(i, \mathcal{P}_i)) \mid s \in \text{IS}(i)\}$ 
```

---

**Theorem 2.1.** 算法 1 的步骤是有限的.

**Theorem 2.2.** 算法 1 计算得到的结果  $\mathcal{P}_i = \bigwedge_{u \in F^i} u$ , 其中  $F^i = \{f(p_n, f(p_{n-1}, \dots, f(p_1, \mathbf{P}_\varepsilon)) \dots), \dots\}$ .

证明. 注意上面的  $\bigwedge$  在这里是和  $\bigcap$  等价的, 因为我们的 semilattice 的 underlying set 是  $\mathfrak{B}(U)$ . 对于从  $\varepsilon$  到  $i$  的任意一条路径  $\pi_k = (p_{k_1}, \dots, p_{k_n})$ , 其中  $p_{k_1} = \varepsilon, p_{k_n} = i$ , 那么

$$\bigwedge_{u \in F^i} u = \bigwedge_{k \in K} f(p_{k_n}, f(p_{k_{n-1}}, \dots, f(p_{k_1}, \mathbf{P}_\varepsilon)) \dots).$$

我们注意到对于任意的  $k \in K$  都有  $p_{k_n} = i$ , 那么应用 lemma 1.13 就有

$$f(i, \bigwedge_{k \in K} f(p_{k_{n-1}}, \dots, f(p_{k_1}, \mathbf{P}_\varepsilon)) \dots).$$

我们在想这个  $\bigwedge$  能不能继续往里面推呢? 对于任意的  $k \in K$ , 集合  $\{p_{k_{n-1}}\}$  可能就不是一个单点集了 (single set), 这个集合就是  $i$  的所有的立即前驱  $\text{IP}(i)$ . 我们又可以尝试用 lemma 1.13 合并一些项

$$f(i, \bigwedge_{i^{-1} \in \text{IP}(i)} f(i^{-1}, \bigwedge_{k \in K \text{ and } p_{k_{n-1}} = i^{-1}} f(p_{k_{n-2}}, \dots, f(p_{k_1}, \mathbf{P}_\varepsilon)) \dots))$$

那么最后我们可以给出一般式

$$f(i, \bigwedge_{i^{-1} \in \text{IP}(i)} f(i^{-1}, \bigwedge_{k \in K \text{ and } p_{k_{n-1}} = i^{-1}} f(i^{-2}, \dots \bigwedge_{k \in K \text{ and } p_{k_3} = i^{-(n-3)}} f(i^{-(n-3)}, \bigwedge_{k \in K \text{ and } p_{k_2} = i^{-(n-2)}} f(i^{-(n-2)}, f(p_{k_1}, \mathbf{P}_\varepsilon)) \dots)))$$

体会这个一般式子, 我们已经证明了. □