

# IFDS: Dataflow Analysis via Graph Reachability

枫聆

2022 年 1 月 4 日

## 目录

**1 Definitions**

**2**

## Definitions

**Annotation 1.1.** 在数据流分析中的“精确”一词，实际上等价于“meet over all vaild path”。

- 在过程内分析 (intraprocedural) 中，一条“vaild path”就是指从某个 procedural 的 CFG 上从 entry node 到特定的点这样一条路径。
- 在过程间分析 (interprocedural) 中，一条“vaild path”就是指当从 main function 开始，且某个 procedural 结束之后返回调用它的 procedural，直到某个特定程序点的这样一条路径。

上述东西没有什么新意，但是让各种名词形式化有利于表达。

**Definition 1.2.** 数据流分析中的可能会出现所有不同的数据值组成的集合  $D$ (underlying set) 称为 dataflow facts. 对于可能分析得到的结果是 dataflow facts 的一个子集，通常我们把所用可能得到的结果记为  $2^D$ 。

**Definition 1.3.** 数据流的值可以表示成位向量 (bit-vectors)，其中每个 bit 可以表示一个具体的 dataflow fact, 且可以每个传递函数可以用相应的位运算来表示，这样的一类数据流分析问题我们称之为 **locally separable problems**. i.e. reaching-definitions, available expressions, live variables.

**Annotation 1.4.** **怎么理解“separable”?** separable 对应的是逻辑位运算过程不同位 bit 是不会相互影响的，也就是两个不同 dataflow facts 是不会相互依赖的。例如在 reaching-definitions 中两个不同变量定义的作用域是不会相互影响的。

**Definition 1.5.** 若 dataflow facts  $D$  是一个有限 (finite) 集合，且每一个 transfer function  $f: 2^D \rightarrow 2^D$  都是可分配的 (distribute, 即满足 join or meet semilattice homomorphism)，这样一类过程间 (interprocedural) 数据流分析问题称为 **interprocedural, finite, distribute, subset problems**，或简称为 **IFDS problems**。

**Definition 1.6.** 设程序  $P$  的每个 procedural  $p$  对应图表示为  $G_p = (N_p, E_p)$ ，其中  $N_p$  表示  $p$  上所有 (atomic) statements,  $E_p$  表示  $p$  的控制流.  $G_p$  中结点种类分为

- 唯一的 start node  $s_p$ ;
- 唯一的 exit node  $e_p$ ;
- 过程调用结点 call node,  $G_p$  中的所有 call nodes 构成的集合记为  $\text{Call}_p$ ;
- 过程调用的返回位置结点 return-site node, 即紧跟在 call node 后面,  $G_p$  中的所有 return-site nodes 构成的集合记为  $\text{Ret}_p$
- 其余的结点与通常 flowgraph 上结点保持一致。

特别地,  $G_p$  上每一对 call node  $c$  和 return-site node  $r$  直接有一条  $c$  到  $r$  的有向边, 称为 **call-to-return-side edge**. 设  $G^* = (N^*, E^*)$ ,  $G^*$  由所有 procedurals 图表示  $G_1, G_2, \dots, G_p, \dots$  和两类特殊的边构成

- **call-to-start edge**: 从  $G_{p_1}$  中 call node 到对应  $G_{p_2}$  的 start node  $s_p$  的有向边.
- **exit-to-return-side edge**: 从  $G_{p_2}$  的 exit node  $e_{p_2}$  到对应  $G_{p_1}$  的 return-side node.

称  $G^*$  为 **supergraph**.

**Annotation 1.7.** 理解 supergraph 只需要了解几个特殊的点

- return-site 这类结点可能是抽象出来的, 在一定程度有利于对 call node 的细化分析.
- 放置 call-to-return-side edge 的目的是用于过程内分析, i.e. local variables.
- 一个 procedural 执行返回会抽象到 exit node 上统一返回, 即 exit-to-return-side 的作用.

**Definition 1.8.** 设  $f: 2^D \rightarrow 2^D$  某个 IFDS problem 中一个 distribute transfer function, 它的一个关系表示  $R_f \subset (D \cup \{0\}) \times (D \cup \{0\})$  (representation relation) 为

$$\begin{aligned} R_f = & \{0, 0\} \\ & \cup \{ (0, y) \mid y \in f(\emptyset) \} \\ & \cup \{ (x, y) \mid y \in f(\{x\}) \text{ and } y \notin f(\emptyset) \}. \end{aligned}$$

其中 0 表示  $\emptyset$ .