

# 数据流图的形式规范\*

秦晓

(浙江大学计算中心, 杭州 310027)

**摘要** 本文提出了属性图,以描述数据流图的拓扑结构和性质,并提出了边(流)组成关系以给出父子数据流图平衡的形式定义,由此得到的数据流图的形式规范强于文献中现有的规范.本文提出的形式规范方法适用于不同类型的数据流图,包括 DeMarco 型数据流图、实时系统数据流图等.

**关键词** 数据流图,形式规范,属性图,边(流)组成关系.

数据流图(DFD)语言是需求分析阶段最常用的一种图形语言.软件开发人员运用这一语言,由抽象到具体、逐层分解,绘出所开发的软件系统的分层 DFD.如同程序设计语言一样,DFD 语言的规范主要包括语法和语义的两个方面.目前来看,大多数关于 DFD 语言的规范<sup>[1-3]</sup>都是非形式的,一般只有几条用自然语言表达的作图规则,这些非形式规范的主要问题是:笼统、含糊和不准确.如果要使计算机能够理解和处理 DFD,就需要采用形式规范技术严格地、准确地定义 DFD 语言的语法和语义.

DFD 语言的形式规范的用途和优点主要包括:

- 作为 DFD 语言使用者共同参照的文本.他们将这个文本作为共同约定和遵循的规范,避免写出互不理解、难以交流的文档.
- DFD 一致性和完整性的自动分析.使用 DFD 语言书写的文档(分层 DFD)可能包含错误,当 DFD 很多时靠人工检查是很费力费时的,并且还可能遗漏一些错误.如果采用形式方法严格地表达 DFD 语言的语法和语义,就有可能研制软件工具来完成 DFD 的一致性和完整性检查.
- 为 DFD 编辑分析工具研制人员提供共同参照的文本.这个文本使不同研制人员对 DFD 语言有清晰的、一致的理解,从而避免编出不相容的或不认同的程序.
- DFD 语言的改进.对 DFD 语言形式规范的研究有助于我们深刻理解这一语言的基本性质(非形式规范难以做到这点),发现语言内部的不合理因素,找到改进语言的依据.
- DFD 语言的标准化.现有的标准<sup>[8]</sup>是非形式的,容易造成不同的理解.要严格地无二义性地定义 DFD 语言,只有依靠形式规范技术.
- DFD 的自动生成.在自动生成 DFD(布局+布线)的过程中要依据形式规范以确保生

\* 本文 1993-05-31 收到

作者秦晓,31岁,讲师,主要研究领域为软件工程,可视语言.

本文通讯联系人:秦晓,杭州 310027,浙江大学 262 信箱

成符合用户描述的 DFD.

本文提出了属性图和一组相关的概念,用它们作为元语言,以给出 DFD 语言的形式规范. 简单地说,一个属性图是一个有向图,其结点和边分为若干类,每类结点和边都分别由对应的一个属性集描述. 一个 DFD 被定义为一个属性图,前者的结点和边及其特性一一对应于后者的结点和边及它们的属性. 分层 DFD 中各 DFD 的边(流)由哪些元素组成,用边(流)的组成关系形式地说明,并基于这个关系给出了父子 DFD 平衡的形式定义.

第 1 节简要地评述相关的工作,第 2 节是属性图和一组有关的基本概念的定义,在第 3 节我们将详细地描述 DFD 语言的形式规范,最后总结本文.

## 1 相关的工作

Adler 运用代数方法形式地定义了 DFD 的处理分析(process decomposition)<sup>[4]</sup>. 在分解一个处理前,先建立这个处理的输入输出连接矩阵,矩阵的行为输入、列为输出,如果一个输入被用于产生一个输出,则在矩阵中这个输入和这个输出对应的行列上作一标记,否则不作标记. 这个矩阵描述了数据流之间的“用于产生(is-used-to-produce)”关系. 由这个矩阵可以导出一个有向无回路图,这个图即描述了这个处理的分解. Adler 仅仅考虑了 DeMarco 型 DFD<sup>[1]</sup>的处理分解的形式规范,就整个 DFD 语言而言,这个规范是远远不够的.

Tao 和 Kung 提出用有向图和一个称为优先关系的二元关系来给出 DFD 语言的形式规范<sup>[5]</sup>. 用有向图的结点表示处理、数据存储和外部实体,并用有向图的边表示数据流(DF). 数据变换和控制变换统一作为处理对待,连续 DF、离散 DF 和控制流也统一表示为 DF. 优先关系是 DF 集合上的一个二元关系,它与 Adler 的“用于产生”关系是一致的,即 DF  $d_1$  优先于 DF  $d_2$ ,当且仅当  $d_1$  用于产生  $d_2$ . 基于这个关系,可以得出关于 DFD 一致性和完整性的判定方法. 这里的一致性是指父子图平衡,而完整性包括三方面内容:(1)每个 DFD 是弱连通的;(2)每个处理和数据存储都既有输入也有输出;(3)每个 DF 都直接或间接地来源于和流到外部实体. Tao 和 Kung 的形式规范方法的主要问题是:(1)由于将 DeMarco 型 DFD 和实时系统 DFD<sup>[2,3]</sup>作统一处理,所以无法表达它们在语法和语义上存在的差别;(2)在语篇分析(涉及两个以上 DFD)方面,只考虑了父子图平衡;(3)由于没有考虑 DFD 中图元的各种属性(如标识符、几何属性等等),所以不能表达 DFD 应满足的、有关这些属性的约束条件;(4)以固定的 DFD 语言为研究对象,没有提供足够的表达手段以适应 DFD 语言的扩充或改进.

Lee 和 Tan 提出用 Petri 网的一个变型——位置-转移网来形式地定义 DFD 语言<sup>[6]</sup>. 其基本思想是:用位置代表外部实体、数据存储的 DF;用单个转移表示不分解的原始处理,而用一个转移组表示非原始处理;定义一个输入函数和一个输出函数,它们分别指明每个处理(即转移或转移组)的输入和输出(即位置),本质上等同于 Adler 的“用于产生”关系;基于上述概念,给出一组关于 DFD 一致性和完整性的约束条件,其包含的检查项目与 Tao 和 Kung 所考虑的基本相同. 由此易见, Lee 和 Tan 的形式规范方法也同样存在前述四个问题.

## 2 基本定义

虽然从抽象的拓扑结构来看,DFD 和图论中的有向图是一致的,但是它们有一个重要的不同:DFD 的结点分为多种类型,如处理、数据存储和外部实体等,实时系统 DFD 还有多种类型的边,如 DF、控制流等;而有向图只有单一类型的结点和边.象 DFD 这样的图称为带多类型结点和边的图,在工程领域里可以找到许多这种类型的图.下面给出这种图的形式定义.

**定义 2.1.** 一个带多类型结点和边的有向图,简称 MVE 图,是一个七元组  $G=(V, E, \psi, T_V, T_E, \delta, \sigma)$ ,其中,  $V$  是一个非空的结点集合,  $E$  是边集合;  $\psi: E \rightarrow V \times V$ , 是一个从边集到结点有序偶集的函数;  $T_V$  是  $V$  中结点的类型的集合,  $T_E$  是  $E$  中边的类型的集合,  $T_V \cap T_E = \emptyset$ ;  $\delta: V \rightarrow T_V$  是一个函数,说明  $V$  中每个结点的类型;  $\sigma: E \rightarrow T_E$  是一个函数,指出  $E$  中每条边的类型.  $\square$

例如,图 1 所示 DFD 可以定义为一个七元组  $G_1=(V, E, \psi, T_V, T_E, \delta, \sigma)$

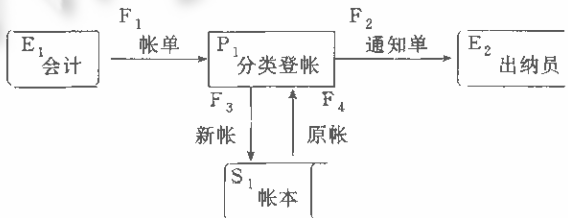


图1 DFD一例

其中,  $V = \{E_1, E_2, P_1, S_1\}$ ;  $E = \{F_1, F_2, F_3, F_4\}$ ;  $\psi = \{(F_1, (E_1, P_1)), (F_2, (P_1, E_2)), (F_3, (P_1, S_1)), (F_4, (S_1, P_1))\}$ ;  $T_V = \{EE, PROC, DS\}$ , 其中, EE: 外部实体类型, PROC: 处理类型, DS: 数据存储类型;  $T_E = \{DF\}$ , 其中 DF: 数据流类型;  $\delta = \{(E_1, EE), (E_2, EE), (P_1, PROC), (S_1, DS)\}$ ;  $\sigma: \forall e \in E, \sigma(e) = DF$ .

**定义 2.2.** 设  $G=(V, E, \psi, T_V, T_E, \delta, \sigma)$  是一个 MVE 图,  $t \in T_V$  是一个结点类型, 定义  
(1)  $t$  的秩为  $G$  中类型为  $t$  的结点的数目, 记作  $rank(t)$ ;  
(2)  $t$  的最大入度  $max_i(t) = \max\{\deg_i(v) \mid v \in \delta^{-1}(t)\}$ , 即  $\delta^{-1}(t)$  中各结点入度的最大者, 这里  $\delta^{-1}(t) = \{v \in V \mid \delta(v) = t\}$

$t$  的最小入度  $min_i(t) = \min\{\deg_i(v) \mid v \in \delta^{-1}(t)\}$ ;

(3)  $t$  的最大出度  $max_o(t) = \max\{\deg_o(v) \mid v \in \delta^{-1}(t)\}$  即  $\delta^{-1}(t)$  中各结点出度的最大者  
 $t$  的最小出度  $min_o(t) = \min\{\deg_o(v) \mid v \in \delta^{-1}(t)\}$   $\square$

在图 1 中,  $rank(EE) = 2$ ,  $max_i(PROC) = min_i(PROC) = 2$ ,  $max_o(DS) = min_o(DS) = 1$ .

**定义 2.3.** 设  $G=(V, E, \psi, T_V, T_E, \delta, \sigma)$  是一个 MVE 图. 笛卡尔积  $T_V \times T_V \times T_E$  的任意一个子集称为  $G$  上的边约束关系. 如果  $R$  是  $G$  上的一个边约束关系, 且对于任意  $e \in E$ ,  $\psi(e) = (v_1, v_2)$ , 有  $(\delta(v_1), \delta(v_2), \sigma(e)) \in R$ , 则称  $G$  是满足边约束关系  $R$  的.  $\square$

例如, 图 1 中的 DFD 满足边约束关系  $\{(EE, PROC, DF), (PROC, EE, DF), (DS, PROC, DF), (PROC, DS, DF), (PROC, PROC, DF)\}$ .

**定义 2.4** 设  $G=(V,E,\phi,T_V,T_E,\delta,\sigma)$  是一个无孤立结点的 MVE 图,  $V'$  是  $V$  的一个子集.

(1)若  $V'$  中每个结点的出度均大于 0,且对于任意  $v \in V'$ , 都有  $u \in V'$ , 使得从  $u$  可达  $v$ , 则称  $V'$  中结点为  $G$  的源点, 称  $V'$  为  $G$  的源点集, 并称  $G$  是有源的;

(2)若  $V'$  中每个结点的入度均大于 0,且对于任意  $v \in V'$ , 都有  $u \in V'$ , 使得从  $v$  可达  $u$ , 则称  $V'$  中结点为  $G$  的汇点, 称  $V'$  为  $G$  的汇点集, 并称  $G$  是有汇的;

(3)设  $V'$  是  $G$  的一个源(汇)点集, 若  $V'$  的任何真子集都不是  $G$  的源(汇)点集, 则称  $V'$  是  $G$  的极小源(汇)点集. □

在图 1 中,  $\{E_1\}$  是这个 DFD 的一个源点集, 并且是极小的; 而  $\{E_2\}$  是它的一个极小汇点集.

**定义 2.5.** 设  $G=(V,E,\phi,T_V,T_E,\delta,\sigma)$  是一个 MVE 图,  $A_1, A_2, \dots, A_n$  是一组属性, 它们的值域分别是  $D_1, D_2, \dots, D_n$ . 记  $A = \{A_1, A_2, \dots, A_n\}$  和  $D = \{D_1, D_2, \dots, D_n\}$ . 如果定义一个函数  $\eta: T_V \cup T_E \rightarrow \mathcal{P}(A)$  ( $A$  的幂集), 对于任意  $t \in T_V \cup T_E$ ,  $\eta(t) = \{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$ ,  $A_{i_j}$  ( $1 \leq j \leq k$ ) 的值域是  $D_{i_j}$ , 这里不妨规定:  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ ; 并定义一个函数集  $\Gamma = \{f_t | t \in T_V \cup T_E\}$ , 使得  $f_t$  为

$$f_t: \delta^{-1}(t) \rightarrow D_{i_1} \times D_{i_2} \times \dots \times D_{i_k}, \text{ 当 } t \in T_V$$

或

$$\sigma^{-1}(t) \rightarrow D_{i_1} \times D_{i_2} \times \dots \times D_{i_k}, \text{ 当 } t \in T_E.$$

那么, 称五元组  $(G, A, D, \eta, \Gamma)$  是一个属性图. □

函数  $\eta$  说明每个结点类型和边类型带有什么属性, 而函数  $\Gamma$  指出每个结点和每条边的各属性的值.

下面以图 1 为例说明. 令  $A = \{\text{编码, 名称}\}$ . 编码的值域  $D_1 = \{E_1, E_2, P_1, S_1, F_1, F_2, F_3, F_4\}$ , 名称的值域  $D_2 = \{\text{会计, 出纳员, 分类登帐, 帐本, 帐单, 通知单, 新帐, 原帐}\}$ . 定义  $\eta(EE) = \eta(\text{PROC}) = \eta(\text{DS}) = \eta(\text{DF}) = A$ . 又,  $\Gamma = \{f_{EE}, f_{\text{PROC}}, f_{\text{DS}}, f_{\text{DF}}\}$ , 其中各函数的定义见图 2.

结点或边	函数值
$f_{EE}$	$E_1$ (E <sub>1</sub> , 会计)
	$E_2$ (E <sub>2</sub> , 出纳员)
$f_{\text{PROC}}$	$P_1$ (P <sub>1</sub> , 分类登帐)
$f_{\text{DS}}$	$S_1$ (S <sub>1</sub> , 帐本)
	$F_1$ (F <sub>1</sub> , 帐单)
$f_{\text{DF}}$	$F_2$ (F <sub>2</sub> , 通知单)
	$F_3$ (F <sub>3</sub> , 新帐)
	$F_4$ (F <sub>4</sub> , 原帐)

图2 函数集P中的函数的定义

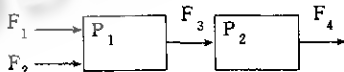


图3 非顶层DFD一例

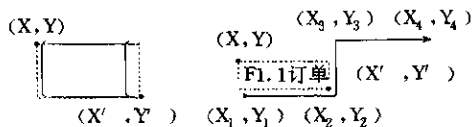


图4 DFD结点类型和边类型的几何属性示意图

### 3 DFD 语言的形式规范

#### 3.1 DFD 的属性图表示

一个 DFD 被定义为一个属性图  $(G, A, D, \eta, \Gamma)$ , 其中,  $G = (V, E, \phi, T_v, T_E, \delta, \sigma)$  是一个 MVE 图  $T_v = \{EE, DS, CS, DT, CT, \downarrow, \uparrow\}$ , 这里, CS 是控制存储类型, DT 和 CT 分别是数据变换和控制变换类型;  $\downarrow$  和  $\uparrow$  是两个特殊的结点类型, 仅出现在非顶层 DFD 中. 对于那些起始结点或终止结点在父图中的流(边), 都不画出它们的起始结点或终止结点, 如图 3 中的  $F_1, F_2, F_4$ . 为了表达的统一性, 我们规定这些流(边)都有虚设的起始结点或终止结点以代表父图中对应结点, 并规定前者的类型为  $\downarrow$ , 后者的类型为  $\uparrow$ , 不论父图中对应结点是什么类型. 例如, 图 3 中  $F_1$  和  $F_2$  的虚起始结点类型为  $\downarrow$ , 而  $F_4$  的虚终止结点类型为  $\uparrow$ .

$T_E = \{DDF, CDF, CF\}$ , 其中 DDF、CDF 和 CF 分别是离散 DF、连续 DF 和控制流类型. 各结点类型所带有的属性不妨统一规定为:

$$(ID, Name, X, Y, X', Y')$$

其中, ID 和 Name 分别是结点的标识符(编码)和名称;  $(X, Y)$  和  $(X', Y')$  分别是结点图符所占据的最小矩阵区域的左上角和右下角坐标(见图 4).

各边类型所带有的属性不妨统一规定为:

$$(ID, Name, X, Y, X', Y', TNum, X_1, Y_1, X_2, Y_2, \dots, X_n, Y_n)$$

这里, ID 和 Name 分别是流的标识符和名称; 如图 4 所示, 按照[8]用水平线、垂直线或由这两种线组成的折线表示流, TNum 是表示流的(折)线的端点数目,  $TNum \leq n$  ( $n$  为最多允许的端点数目), 而  $(X_1, Y_1), \dots, (X_n, Y_n)$  分别是这些端点的坐标,  $(X, Y)$  和  $(X', Y')$  分别是流的标识符和名称所占据的最小矩阵区域的左上角和右下角坐标.

#### 3.2 单个 DFD 内的规范

1. 顶层 DFD 必须包含外部实体, 不包含  $\downarrow$  和  $\uparrow$  类型结点

$$\text{rank}(EE) > 0, \quad \text{rank}(\downarrow) = \text{rank}(\uparrow) = 0$$

2. 非顶层 DFD 必须包含  $\downarrow$  和  $\uparrow$  类型结点, 不包含外部实体

$$\text{rank}(\downarrow) > 0, \text{rank}(\uparrow) > 0, \text{rank}(EE) = 0$$

3. 对结点的入度和出度的限制

$$(1) \forall v \in \delta^{-1}(EE), \text{deg}_i(v) + \text{deg}_o(v) > 0$$

$$(2) \max_i(\downarrow) = 0, \min_o(\downarrow) = \max_o(\downarrow) = 1; \max_o(\uparrow) = 0, \min_i(\uparrow) = \max_i(\uparrow) = 1$$

$$(3) \min_i(DT) > 0, \min_o(DT) > 0; \min_i(CT) > 0, \min_o(CT) > 0;$$

$$\min_i(DS) > 0, \min_o(DS) > 0; \min_i(CS) > 0, \min_o(CS) > 0$$

(4)  $\max_i(DT) < N_1, \max_i(CT) < N_2$ , 这里  $N_1, N_2$  是人为规定的阈值, 用以避免变换间联系太复杂.

4. 每个 DFD 必须包含数据变换, 即:

$$\text{rank}(DT) > 0, \text{对于非顶层 DFD}, \text{rank}(DT) > 1.$$

5. 集合  $\{v | v \in \delta^{-1}(EE) \text{ 且 } \text{deg}_o(v) > 0\}$  和  $\{v | v \in \delta^{-1}(EE) \text{ 且 } \text{deg}_i(v) > 0\}$  分别是顶层 DFD 的源点集和汇点集;  $\delta^{-1}(\downarrow)$  和  $\delta^{-1}(\uparrow)$  分别是非顶层 DFD 的极小源点集和极小

汇点集.

6. 每个 DFD 满足边约束关系  $\{(EE,DT,DDF), (EE,DT,CDF), (EE,DT,CF), (EE,CT,CF), (DT,EE,DDF), (DT,EE,CDF), (DT,EE,CF), (DT,DS,DDF), (DT,DS,CDF), (DT,CS,CF), (DT,DT,DDF), (DT,DT,CDF), (DT,CT,CF), (CT,DT,CF), (CT,CT,CF), (CT,CS,CF), (CT,EE,CF), (DS,DT,DDF), (DS,DT,CDF), (CS,CT,CF), (\downarrow,DT,DDF), (\downarrow,DT,CDF), (\downarrow,DT,CF), (\downarrow,CT,CF), (DT,\uparrow,DDF), (DT,\uparrow,CDF), (DT,\uparrow,CF), (CT,\uparrow,CF)\}$

7. 对标识符和名称的一些限制

(1)  $\forall a, b \in V, a \neq b$ , 有  $ID(a) \neq ID(b), Name(a) \neq Name(b)$ . (2)  $\forall e_1, e_2 \in E, e_1 \neq e_2, \psi(e_1) = (a, b), \psi(e_2) = (c, d)$ , 若  $\sigma(e_1) \neq \sigma(e_2)$ , 则  $ID(e_1) \neq ID(e_2), Name(e_1) \neq Name(e_2)$ ; 否则, 若  $ID(e_1) = ID(e_2)$ , 则  $a = c$  或  $\delta(a) = \delta(c) = \downarrow$ . (3)  $\forall v \in V, \forall e \in E$ , 有  $ID(v) \neq ID(e), Name(v) \neq Name(e)$ .

8. 对几何属性的一些限制

$\forall a, b \in V$ ,  $a$  和  $b$  不能重叠, 要满足条件:  $X(a) > X'(b)$  或  $X(b) > X'(a)$  或  $Y(a) > Y'(b)$  或  $Y(b) > Y'(a)$ . 不难推出边与边、边与结点不交叉不重叠的充分条件, 这里不再赘述.

3.3 多个 DFD 间的规范

3.3.1 关于标识符和名称的限制

(1) 设  $DFD_1$  是分层 DFD 中任何一个非顶层 DFD,  $V_1, E_1$  分别是其结点集和边,  $\delta_1$  和  $\sigma_1$  分别说明其结点和边的类型;  $DFD_2$  是  $DFD_1$  的任何一个祖先 DFD,  $V_2, E_2$  分别为其结点集和边集,  $\delta_2$  和  $\sigma_2$  分别说明其结点和边的类型. 那么,

$\forall a \in V_1, \forall b \in V_2$ , 有  $ID(a) \neq ID(b), Name(a) \neq Name(b)$ ;

$\forall c \in E_1, \forall d \in E_2$ , 若  $ID(c) = ID(d)$ , 则  $\sigma_1(c) = \sigma_2(d)$ , 并且  $\delta_1(v_1) = \downarrow$  或  $\delta_1(v_2) = \uparrow$ , 这里  $v_1, v_2$  分别是  $c$  的起始结点和终止结点.

(2) 如果(1)中的  $DFD_2$  是分层 DFD 中任意一个 DFD, 但不是  $DFD_1$  的祖先, 那么,

$\forall a \in \delta_1^{-1}(DT) \cup \delta_1^{-1}(CT), \forall b \in \delta_2^{-1}(DT) \cup \delta_2^{-1}(CT)$ , 有  $ID(a) \neq ID(b), Name(a) \neq Name(b)$ ;

$\forall c \in E_1, \forall d \in E_2$ , 若  $ID(c) = ID(d)$ , 则  $\sigma_1(c) = \sigma_2(d)$ , 同时有 ①  $\delta_1(v_1) = \delta_2(v'_1) = DS$  或  $CS, ID(v_1) = ID(v'_1)$ , 这里  $v_1, v'_1$  分别是  $c, d$  的起始结点; 或 ②  $\delta_1(v_2) = \delta_2(v'_2) = DS$  或  $CS, ID(v_2) = ID(v'_2)$ , 这里  $v_2, v'_2$  分别是  $c, d$  的终止结点; 或 ③  $\delta_1(v_1) = \downarrow$  或  $\delta_2(v'_1) = \downarrow$  或  $\delta_1(v_2) = \uparrow$  或  $\delta_2(v'_2) = \uparrow$ .

3.3.2 父子 DFD 保持平衡

如果上面的  $DFD_2$  是  $DFD_1$  的父 DFD, 则它们必须是平衡的.

记  $F$  为所有 DDF 类型流的集合,  $S$  为所有该类型流组成元素的集合. 定义边(流)组成关系如下:  $\forall a \in F, \forall b \in S, (a, b) \in R$  当且仅当  $b$  是  $a$  的组成元素.

又记  $v_p$  是  $DFD_1$  对应的  $DFD_2$  中的 DT 类型结点, 它的输入流元素集和输出流元素集分别是:

$I(v_p) = \{b \in S \mid \exists e \in E_2, e \text{ 的终止结点是 } v_p \text{ 且 } (e, b) \in R\}$  和  $O(v_p) = \{b \in S \mid \exists e \in E_2, e$

的起始结点是  $v_p$  且  $(e, b)R$

又,  $DFD_1$  的输入流元素集和输出流元素集分别是:

$$I(DFD_1) =_{df} \bigcup_{\substack{v \in V_1, \\ \delta(v) = \downarrow}} I(v) \text{ 和 } O(DFD_1) =_{df} \bigcup_{\substack{v \in V_1, \\ \delta(v) = \uparrow}} O(v)$$

如果  $I(v_p) = I(DFD_1)$  且  $O(v_p) = O(DFD_1)$ , 则称  $DFD_1$  和  $DFD_2$  关于 DDF 类型流是平衡的. 对于其他类型流可以依此类推. 如果  $DFD_1$  和  $DFD_2$  关于各种类型是平衡的, 则称它们是平衡的.

**结束语:** 属性图既可以刻画 DFD 的拓扑结构, 也可以描述 DFD 的其他性质, 如几何性质; 边(流)组成关系能够用来形式地表达父子 DFD 平衡这一概念. 本文提出的以属性图和边(流)组成关系为元语言的形式规范方法不仅适用于 DeMarco 型 DFD, 而且适用于实时系统 DFD. 采用这种方法得到的形式规范强于迄今已发表的文献中的形式规范<sup>[4-7]</sup>要更准确的严格. 这种方法还具有较好的适应性, 能适应 DFD 语言的改进或扩充, 因为它允许按需要设定 DFD 的结点类型集和边类型集以及它们的属性集.

### 参考文献

- 1 DeMarco T. Structured analysis and system specification. Yourdon Press, 1978.
- 2 Yourdon E. Modern structured analysis. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
- 3 Peters L. Advanced structured analysis and design. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- 4 Adler M. An algebra for data flow diagram process decomposition. IEEE Trans. SE, 1988; 14(2): 169—183.
- 5 Tao Y, Kung C. Formal definition and verification of data flow diagrams. J. Systems & Software. 1991; 16(1): 29—36.
- 6 Lee P, Tan K P. Modelling of visualised data-flow diagrams using petri net model. Software Eng. J., 1992; 4—12.
- 7 Tse T H and Pong L. Towards a formal foundation for DeMarco data flow diagrams. Computer J., 1989; 32(1): 1—12.
- 8 GB1526—89(ISO5807—1985), 国家技术监督局, 1989.

## FORMAL SPECIFICATION OF DATA FLOW DIAGRAMS

Qin Xiao

(Zhejiang University, Hangzhou 310027)

**Abstract** The attributed diagram is presented to specify the structure and properties of data flow diagrams (DFD), and a binary relation called the edge (flow) composition relation is proposed as the base on which the notion of balance between a DFD and its child DFD is formally defined. By using the above-mentioned approach, this paper provides a formal specification of DFD which is stronger than those found in the literature. Moreover, this approach applies to different sorts of DFD, including DeMarco DFD, DFD for real-time systems and so on.

**Key words** Data flow diagrams, formal specification, attributed diagram, edge(flow) composition relation.