测试数据选择理论 20 年

暴建民

(黑龙江大学应用数学研究所　哈尔滨　150080)

杨孝宗　杨　楠　王申科

(哈尔滨工业大学计算机科学与工程系　哈尔滨　150006)

摘要　本文将测试数据选择理论发展的 20 年划分成 3 个阶段——形成阶段、发展阶段、完善阶段,给出了每一阶段具有代表性的研究成果和特点,讨论了此理论在软件工程中的应用.

关键词　测试数据选择理论,充分性标准,规格说明书.

软件测试是当今计算机科学与工程中起着至关重要作用的领域之一,而如何选择数据来进行有效的测试是完成软件测试的关键. 自 1975 年,J. B. Goodenough 和 S. L. Gerhart[1] 提出测试数据选择应面向理论至今已有 20 年的历史. 它已引起了计算机、通信和自动化等领域的广泛关注和兴趣. 测试数据选择理论(以下简称测试理论)的发展动力主要基于以下几点:可靠性测试—系统的生命之本;可信性测试标准—系统的生命所在;付出劳动巨大—软件开发花费的大约 50% 都用于软件测试;酿成后果惨重—法国 J. C. Laprie 等人作了一个统计,在法国,保险公司为由计算机引起的事故付出的赔偿费远远超过数据处理商的利润.

本文旨在对测试理论中最有实用价值的理论成果作介绍,并探讨了其在工程中的应用.

1　测试理论形成阶段

第 1 阶段:1975～1983 年为测试理论的形成阶段,其主要特点是寻找能保证测试正确性的测试数据选择形式化系统.

1.1　理想测试理论的基本定义和定理

给定程序 F,具有输入域 D,输出域 R,输出需求 $OK(d)=OUT(d,F(d))$ 和测试选择标准 C:

(1)$\forall\ d\in D$,则 F(若是终止的)产生程序的输出 $F(d)\in R$.

(2)F 的输出规格说明书定义为 $OUT(x,y)$,其中 $x\in D, y\in R$. F 在输入 d 点是正确

---

* 本文研究得到国家自然科学基金资助. 作者暴建民,1963 年生,讲师,主要研究领域为分布容错计算机,软件测试. 杨孝宗,1939 年生,教授,博士导师,主要研究领域为计算机系统可靠性,容错技术,Ada 语言. 杨楠,1969 年生,助工,主要研究领域为软件可靠性技术. 王申科,1961 年生,博士,副教授,主要研究领域为容错计算机结构,软件可靠性技术.

本文通讯联系人:暴建民,哈尔滨 150080,黑龙江大学应用数学研究所

本文 1996-06-20 收到修改稿

的,引入$OK(d)$谓词定义:$OK(d)$为真↔$F(d)$是正确的输出↔$F(d)$存在且$OUT(d,F(d))$为真.

(3)若有$T\subseteq D$,且:$(\forall t\in T)OK(t)\rightarrow(\forall d\in D)OK(d)$,则说$T$是一个理想测试,记为$Ideal(T)$.

(4)定义了成功$Succ$:对于测试集T是成功的.$Succ(T)\leftrightarrow(\forall t\in T)(OK(t))$.

(5)定义了某一测试准则C的一些特性:$Complete$(完全的)、$Vaild$(有效的)和$Reliable$(可靠的):

设T是D的一个子集,则:

$Complete(T,C)\leftrightarrow T$满足$C$.

$Reliable(C)\leftrightarrow(\forall T_1,T_2)(T_1\subseteq D\cap T_2\subseteq D\cap Complete(T_1,C)\cap Complete(T_2,C))\rightarrow(Succ(T_1)\cap Succ(T_2))\cup(\neg Succ(T_1)\cap\neg Succ(T_2))$.

$Valid(C)\leftrightarrow(\forall d\in D)\neg OK(d)\rightarrow(\exists T\subseteq D)Complete(T,C)\cap\neg Succ(T)$.

$Ideal(C)\leftrightarrow Reliable(C)\cap Valid(C)$.

$Ideal(T)\leftrightarrow Ideal(C)\cap Complete(T,C)$.

从以上定义得到测试基本定理:如果有程序F存在既可靠又有效的测试标准C,且有一满足C的测试集T,若F在T上是正确的,则F在D上是正确的(D是F的定义域,$T\subseteq D$).

基本定理:$(\exists T\subseteq D)(\exists C)(Complete(T_1,C)\wedge Reliable(C)\wedge(Succ(T))\supseteq(\forall d\in D)OK(d)$

1.2 Weyuker 和 Ostrand 理论[2]

定理(可靠性和有效性关系定理). $(\forall C)(Val(C)\vee Rel(C))$.

推论. 一个标准C是一致有效的和一致可靠的当且仅当测试集选择了整个输入域.

为此,提出了半正确的概念.即不追求理想测试获取程序正确性的目标,而设法通过测试暴露出程序中存在的某类错误或验证此类错误不存在.从而为寻求半正确测试准则奠定了基础.谓词$Revealing$:$Revealing(C,S)\Leftrightarrow(\exists d\in S)(\neg OK(d))\rightarrow(\forall T\subseteq S)(C(T)\rightarrow\neg SUCC(T))$(其中$C$是一测试标准,$S$是输入域的一个子集,$C(T)$表示测试集$T$满足标准$C$).

1.3 Howden 路径测试可靠性理论[3~5]

可靠性测试集定义. P是计算函数F的一个程序,其定义域为D.如果$T\subseteq D$,那么T是可靠性测试集,当$(\forall X\in T)P(X)=F(X)\rightarrow(\forall X\in D)P(X)=F(X)$

定理. 对给定计算函数F的任意一个程序P,其定义域为D.如果$T\subset D$,那么不存在可计算过程H用以产生可靠性测试集T,使得

$$(\forall X\in T)P(X)=F(X)\rightarrow(\forall X\in D)P(X)=F(X).$$

可靠测试策略定义. 设P是一个程序,H是一个测试策略.$\{T_i\}_{i=1}^n$是输入域通过H得到的子集.如果任意地从每一T_i中选取一元素构成测试T是一个可靠测试集,则H是一个可靠测试策略.

可靠测试策略定理. 设P是计算函数F的一个程序,H是P的一个测试策略,$\{T_i\}_{i=1}^n$是H产生的输入子系列.则H是P的可靠测试策略当且仅当P是正确的或是存在某一

$T_i,(\forall\ X\in T_i)P(X)\neq F(X)$.

1.4 Gourlay 测试的数学框架[6]

设所有程序集合为 P;所有规格说明书为 φ;所有测试的集合为 τ. 某程序 p 相对于某规格说明书 s 的正确性用 $p\ corr\ s$ 表示;p 在测试 t 上相对于 s 的正确性用 $p\ OK_t\ s$ 表示. 显然,$corr\subseteq P\times\varphi$;$OK\subseteq\tau\times P\times\varphi$.

定义了测试系统:一个测试系统是五元组$\langle P,\varphi,\tau,corr,OK\rangle$,其中 P,φ,τ 是上述定义的集合.

$corr\subseteq P\times\varphi,OK\subseteq\tau\times P\times\varphi$,且$\forall\ p\forall\ s\forall\ t(p\ corr\ s\rightarrow p\ OK_t\ s)$,则$\langle P,\varphi,\tau,corr,OK\rangle$构成一个测试系统.

定义了测试方法:一个测试方法是一个函数 $M,M:P\times\varphi\rightarrow\tau$.

如果一种测试方法 M 比另一种测试方法 N 强,我们说 $M\geqslant N$:$M\geqslant N\Leftrightarrow\forall\ p\forall\ s(p\ OK_M\ s\Rightarrow p\ OK_N\ s)$,测试方法可靠性定义为:$p\ rel_M\ s\Leftrightarrow(p\ OK_M\ s\Rightarrow p\ corr\ s)$

2 测试理论发展阶段

第 2 阶段:1983～1989 年为测试理论的发展阶段,其主要特点是形成了较完善的功能测试方法和数据流测试标准.

2.1 功能测试理论[7,8]

功能测试理论是把有效性问题划分成一系列的较小的问题,从而克服了解决问题过程的不确定性. 功能是由基本的或由其它功能成分所组成. 组成成分一般分为 2 大类:方式和结构. 方式有 3 种:代数、条件和重复. 如功能 f_1 和 f_2 可以用以上 3 种方式合成新功能 f_3:

$f_3(x)=[f_1(x)]^2+3f_1(x)f_2(x)$　(代数方式)

$f_3(x)=$if $b(x)$ then $f_1(x)$ else $f_2(x)$　(条件方式)

$f_3(x)=$repeat while $b(x)$　(重复方式)

$x:=f_1(x)$

endrepeat.

功能合成是用正确的功能成分序列或由正确的状态传递序列组成,一般用功能序列结构图(如图 1)或状态传递序列图(如图 2)表示.

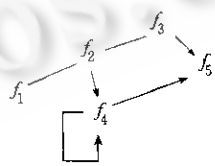

图1 功能序列结构

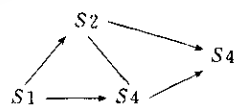

图2 状态传递序列

功能分析就是要通过可用的 ORACLE 来断定功能或系统状态序列的正确. 功能分析理论要求只需检查所有功能或系统状态序列的一个较小子集,就能确定错误序列是否发生. 理论上要证明如果在子集中无错误序列,那么整个系统就不存在错误序列.

2.2 路径选择标准

2.2.1 Rapps and Weyuker 的标准家族[9,10]

他们以数据流分析的概念为基础,提出了包含著名的 3 种测试标准和一些新的路径选择标准,特别是发现并证明了这些标准的偏序关系,为测试程度的判断在理论上提供了依据.

2.2.2　Ntafos' k 元组需求标准[11,12]

为了克服仅依赖控制流信息本身的缺陷，他定义了一类依据数据流的路径选择标准，称为 K 元组需求标准. 此类标准要求选择的一路径集能覆盖定义和使用交替改变链，称为 $K-DR$ 相互作用. 在一个 $K-DR$ 相互作用中的每个定义都要在此链中达到下一个使用，并且此链中下一个定义也在使用出现的同一节点出现. 因此 $K-DR$ 相互作用信息沿一条子路的传播称为对 $K-DR$ 相互作用子路.

2.2.3　Laski 和 Korel 标准[13]

Laski 和 Korel 给出了 3 种基于数据流的路径选择标准，他们强调了对于给定的一个节点，它可能含有几个不同的变量的使用，而每个使用又可能是对出现在不同节点的几个定义都是可达的，他们提出的标准是关心沿着可达节点定义的各种组合来选择子路. 提出的 3 种标准是到达覆盖标准 R—C(reach coverage)，正文覆盖标准 C—C(context coverage)和有序正文覆盖标准 OC—C(ordered context coverage).

2.2.4　Clarke Podgurkski Richardson 和 Zeil 修正标准[14]

将第 2.2.1～2.2.4 节中提出的标准分别称为 A,B,C,D 标准，Clarke 等人在保留原有标准基本含义的前提下，提出了对 B,C 标准的修正，并得出了较理想的包含层次关系，同时发现了 3 种标准间的相互关系.

我们用虚线箭头表示 D 组研究发现的包含层次关系定理；用实线箭头(不在框内)表示 A 组人员提出和研究发现的关系；而实框和虚框分别表示 B 组和 C 组人员提出并研究出的包含层次关系定理，图 3 是在 A,B,C 三组定义的基础上，由 A,B,C,D 四研究小组发现的，并用他们的姓名简写命名的包含层次图，图 4 是由 A,B,C,D 四研究小组提出修正后的包含层次图.

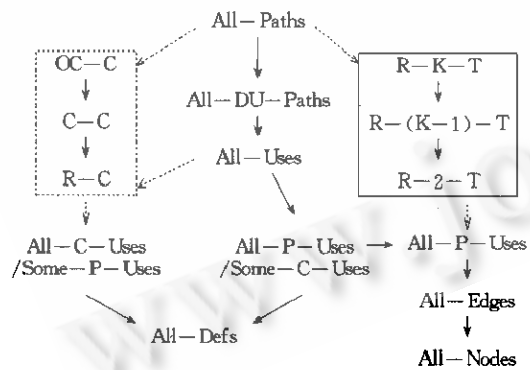

图3　RW—LK—N—CPRS包含层次图

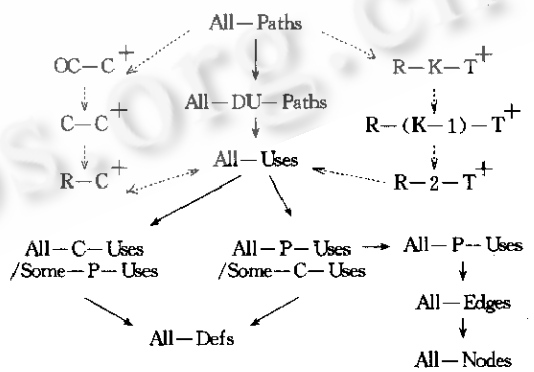

图4　修正的RW—LK—N—CPRS包含层次图

为了说明图 3 和图 4 的含义，我们首先定义包含关系. 标准 $C1$ 包含标准 $C2$：如果对每个模块的控制流图 $G(M)$，G 的任意完全路径集，若满足 $C1$ 标准也一定满足 $C2$ 标准. 标准 $C1$ 严格包含标准 $C2$(记为 $C1\rightarrow C2$)：如果只要标准 $C1$ 包含标准 $C2$，且存在一些 $G(M)$ 图，在其中有满足 $C2$ 标准的一些完全路径，但不满足标准 $C1$. 标准 $C1$ 和 $C2$ 是不可比的：如果标准 $C1$ 和 $C2$ 关系为：$C1$ 包含 $C2$ 不成立，$C2$ 包含 $C1$ 也不成立，标准 $C1$ 和标准 $C2$ 是相互包含关系：如果 $C1$ 包含 $C2$ 和 $C2$ 包含 $C1$ 同时成立.

定理 1. 由 A,B,C(包括 D)组定义的标准，它们具有如图 3 所示的严格包含关系所定

义的偏序关系.进一步,标准 Ci 严格包含标准 Cj 是指当且仅当在图 3 中,或者 Ci 和 Cj 具有明确的偏序表示,或者由 Ci 沿偏序关系图能传递到 Cj.

定理 2. 由 D 组定义的标准(包括 A,B,C),它们具有如图 4 所示的严格包含关系(或相互包含关系)所定义的偏序关系.进一步,标准 Ci 和标准 Cj 具有严格包含关系(或相互包含关系)是指当且仅当在图 4 中,或者 Ci 和 Cj 具有明确的偏序表示(或相互包含关系),或者由 Ci 沿偏序关系图能传递到 Cj.

3　测试理论完善阶段

第 3 阶段:1989～1994 年是测试理论完善阶段,其主要特点是形成了与规格说明书相关的理论.考虑到知识的系统性,我们将公理化体系也在此介绍.

3.1　测试公理化性质[15,16]

软件测试数据充分性公理:

公理 1. (APP)每个程序相对于此标准都有一个充分测试集.

公理 2. (NA)在不需要穷尽测试的条件下,相对于此标准的一些程序是可以被充分测试的.

公理 3. (MON)对任意测试集相对于此标准是充分的;那么此测试集的超集相对于此标准也是充分的.

公理 4. (IES)相对于此标准空集不是充分测试集.

公理 5. (AE)一个测试集对某个程序是充分的,相对于此标准,它不是对所有的程序都是充分的.

公理 6. (GMC)具有相似形的程序间,对于同一测试集它们并不一定都是充分的.

公理 7. (AD)相对于某一测试标准,测试集对于一个程序是充分的,并不一定对此程序的每个组成部分也充分.

公理 8. (AC)相对于某一测试标准,测试集对于第 1 个程序是充分的,且产生的结果值对第 2 个程序也是充分的,但对由第 1 个程序和第 2 个程序组成的序列程序不一定是充分的.

不依赖规格说明书性质关系定理:SI 表示不依赖规格说明书;REN:重名;$COMP$:复杂性;SC:语句覆盖.

定理 1. 如果 C 满足 APP_{SI} 和 IES_{SI},那么 C 满足 AD_{SI}.

定理 2. 如果 C 满足 APP_{SI} 和 AE_{SI},那么 C 满足 NA_{SI}.

定理 3. 如果 C 满足 APP_{SI} 和 GMC_{SI},那么 C 满足 NA_{SI}.

定理 4. 如果 C 满足 APP_{SI} 和 AC_{SI},那么 C 满足 NA_{SI}.

定理 5. 如果 C 满足 $COMP_{SI}$,那么 C 满足 NA_{SI}.

定理 6. 如果 C 满足 SC_{SI} 和 AE_{SI},那么 C 满足 IES_{SI}.

定理 7. 如果 C 满足 SC_{SI} 和 NA_{SI},那么 C 满足 AE_{SI}.

定理 8. 如果 C 满足 APP_{SI} 和 SC_{SI},那么 C 满足 AD_{SI}.

定理 9. 如果 C 满足 SC_{SI} 和 $COMP_{SI}$,那么 C 满足 AE_{SI}.

依赖规格说明书性质关系定理:

定理 10. 如果 C 是不依赖规格说明书的标准，$NA \Leftrightarrow NA_{SI}$.

定理 11. 如果 C 是不依赖于程序的标准并满足 NI 和 IES，那么 C 满足 AD.

定理 12. 如果 C 是满足 NAL 和 SC 的标准，那么 C 是依赖程序的.

定理 13. (a)存在标准 C 满足 APP 和 AE，但不满足 NA；

(b)存在标准 C 满足 APP 和 GMC，但不满足 NA；

(c)存在标准 C 满足 APP 和 AC，但不满足 NA；

(d)存在标准 C 满足 $COMP$，但不满足 NA.

定理 14. 如果 C 满足 SC 和 $COMP$，那么 C 满足 AE.

3.2 程序测试故障检测能力评估[17~19]

3.2.1 基本概念

$C1$ 窄化 $C2$：设 $C1$，$C2$ 是标准. 对于 (P,S)，如果对于每个子域 $D \in SD_{C2}(P,S)$，存在一个子域 $D' \in SD_{C1}(P,S)$ 满足 $D' \in D$.

$C1$ 覆盖 $C2$：设 $C1$，$C2$ 是标准，对于 (P,S)，若对于每个子域 $D \in SD_{C2}(P,S)$，存在一个非空子域集 $\{D_1,\ldots,D_n\}$ 属于 $SD_{C1}(P,S)$ 并且满足 $D_1 \cup \ldots \cup D_n = D$.

$C1$ 划分 $C2$：设 $C1$，$C2$ 是标准，对于 (P,S)，若对于每个子域 $D \in SD_{C2}(P,S)$，存在一个非空两两不相交子域集 $\{D_1,\ldots,D_n\}$ 属于 $SD_{C1}(P,S)$ 并且满足 $D_1 \cup \ldots \cup D_n = D$.

$C1$ 真覆盖 $C2$：设 $SD_{C1}(P,S)=\{D_1^1,\ldots,D_m^1\}$，$SD_{C2}(P,S)=\{D_1^2,\ldots,D_n^2\}$. 在 (P,S) 意义下，如果存在一个多集 $M=\{D_{1,1}^1,\ldots,D_{1,k_1}^1,\ldots,D_{n,1}^1,\ldots,D_{n,k_n}^1\}$ 并满足下列条件：

$M \subseteq SD_{C1}(P,S)$ 且 $D_1^2=D_{1,1}^1 \cup \ldots \cup D_{1,k_1}^1,\ldots,D_n^2=D_{n,1}^1 \cup \ldots \cup D_{n,k_n}^1$.

$C1$ 真划分 $C2$：设 $SD_{C1}(P,S)=\{D_1^1,\ldots,D_m^1\}$，$SD_{C2}(P,S)=\{D_1^2,\ldots,D_n^2\}$. 在 (P,S) 意义下，如果存在一个多集 $M=\{D_{1,1}^1,\ldots,D_{1,k_1}^1,\ldots,D_{n,1}^1,\ldots,D_{n,k_n}^1\}$ 并满足下列条件：

$M \subseteq SD_{C1}(P,S)$ 且 $D_1^2=D_{1,1}^1 \cup \ldots \cup D_{1,k_1}^1,\ldots,D_n^2=D_{n,1}^1 \cup \ldots \cup D_{n,k_n}^1$ 且对于每个 i，集合 $\{D_{i,1}^1,\ldots,D_{i,k_i}^1\}$ 是两两不相交的.

3.2.2 5 种关系评估

一个测试是基于子域，对于 (P,S)，有一个非空子域多集，$SD_C(P,S)$，假定测试选择满足基于子域标准 C，首先根据 $SD_C(P,S)$ 划分域，然后随机地选择 D_i 中一个元素满足 $D_i \in SD_C(P,S)$，令 $d_i=|D_i|$，m_i 是 D_i 中产生失效的输入个数. 每一个测试情形是按照统一分布从每个子域中独立选取.

$M1$，$M2$，$M3$ 给出了用这种测试选择的方式选取的测试组至少产生一个错误的概率.

G. Frankl 和 J. Weyuker 将输入域按测试标准划分为：(1)分支输入域集；(2)路径输入域集；(3)数据流输入域集；(4)弱变换输入域集；(5)基于规范化集；(6)穷尽输入域集. 提出了故障检测能力形式化分析模型：

$$M1(C,P,S)=\max_{1\leqslant i\leqslant k}\left(\frac{m_i}{d_i}\right)$$

$$M2(C,P,S)=1-\prod_{i=1}^{k}\left(1-\frac{m_i}{d_i}\right)(k \text{ 为在标准 } C \text{ 下划分子域的个数})$$

$$M3(C,P,S,n)=1-\prod_{i=1}^{k}\left[1-\left(\frac{m_i}{d_i}\right)\right]^n(\text{每个子域选择 } n \text{ 个情形}).$$

表 1 关系定理表

	$M1(C1)\geqslant M1(C2)$	$M2(C1)\geqslant M2(C2)$	$M3(C1)\geqslant M3(C2)$
$C1$ 窄化 $C2$	不一定	不一定	不一定
$C1$ 覆盖 $C2$	不一定	不一定	不一定
$C1$ 划分 $C2$	总成立	不一定	不一定
$C1$ 真覆盖 $C2$	不一定	总成立	不一定
$C1$ 真划分 $C2$	总成立	总成立	不一定

4 结 论

测试理论形成阶段中还有许多科学家从不同的角度去研究测试,如 Geller 采用拓广断言,Kennaway 把测试建立在不确定计算机程序的语义问题上,并取得了很大的成果,但是也面临着一个不愿接受的结论:测试是不能保证程序无错误,只能提高程序的可靠性.

测试理论发展阶段中研究的核心问题是通过功能划分和提出的一系列标准来指导测试,同时也回答了何时测试才算充分. 实质上,功能测试也是一种测试标准. 功能测试覆盖标准从某种意义上来说比分支测试、语句测试要求更充分,功能测试标准需要对一个程序功能的所有程序语句组合都需要测试一次. 功能测试已在网络协议验证和通讯工程上得到广泛的应用. 我们在测试自行研制实现的分布容错机 HIT－501 通讯子网时,就是采用了状态序列图方法进行的.[20]依据充分性测试的关键问题是如何使功能的组合数或标准要求的测试情形尽可能的少. 如在使用 Rapps－Weyuker 标准家族时,All－uses,all－p－uses/some－c－uses,all－c－uses/some－p－uses 的复杂性为 $O(t^2)$(t 是二元判定语句数),按照 *Knuth* 统计的结果表明,中等大小的程序一般也要有 1 000 个二元分支语句,所需测试实例至少需要 10 000(在最坏情况下),从而使标准难以实用化. 我们在实现此标准的过程中(如图 5),通过使用特殊赋值语句($y:=y$),使得 m 节点满足所有在此节点前的所有变量都以 $y:=y$ 形式出现于 m 点,此标准的复杂度将会变为 $O(t)$,从而使标准实用化.

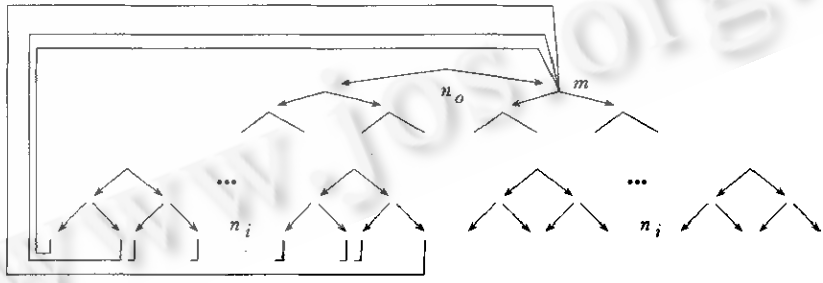

图5

测试理论完善阶段是尽可能使测试定理化和数量化,测试理论的集中点仍是 Gourlay 理论框架中所涉及的元素. Frankl 和 Weyuker 基于检测故障能力建立了层次化软件测试方法. 最近国外的研究结果表明:弱变换测试和 all－uses 数据流覆盖标准间是可证明不可比较的. 而有关 All－uses,All－DU－paths 和 All－Edges 测试标准的关系,在不作限制和不确定假设的前提下,它们的关系是与要求的相关强度一致的. 最后要说明一点,由于本文试图综述一个领域,限于篇幅,难免挂一漏万,许多文章未能列入,不妥之处,望读者指正.

参考文献

1 Goodenough J B, Gerhart S L. Toward a theory of test data selection. IEEE Trans. Software Eng., 1975,**1(2)**: 156～173.

2 Weyuker E J, Ostrand T J. Theories of proper testing and the application of revealing subdomains. IEEE Trans. Software Eng., 1980,**6(5)**:236～246.

3 Howden W E. Reliability of the path analysis testing strategy. IEEE Trans. Software Eng., 1976,**2(2)**:208～215.

4 Howden W E. Symbolic testing and the DISSECT symbolic evaluation system. IEEE Trans. Software Eng., 1977, **3(4)**:266～278.

5 Howden W E. Functional program testing. IEEE Trans. Software Eng., 1980,**6(2)**:162～169.

6 Gourlay J S. A mathematical framework for the investigation of testing. IEEE Trans. Software Eng., 1983,**9(5)**: 686～709.

7 Howden W E. A functional approach to program testing and Analysis. IEEE Trans. Software Eng., 1986,**12(10)**: 997～1005.

8 Howden W E. Functional programming testing and analysis. New York: McGraw-Hill, 1987.

9 Weyuker E J. The complexity of data flow criteria for test data selection. Information Processing Letters, 1984,**19 (8)**:103～109.

10 Rapps S, Weyuker E J. Selecting software test data using data flow information. IEEE Trans. Software Engineering, 1985,**11(4)**:367～375.

11 Ntafos S C. On required element testing. IEEE Trans. Software Eng., 1984,**10(6)**:795～803.

12 Ntafos S C. A comparison of some structural testing strategies. IEEE Trans. Software Eng., 1988,**14(6)**:868～874.

13 Laski J W, Korel B. A data flow oriented program testing strategy. IEEE Trans. Software Eng., 1983,**9(3)**:347～354.

14 Clarke L, Podgurski A, Richardson D J *et al*. A formal evaluation of data flow path selection criteria. IEEE Trans. Software Eng., 1989,**15(11)**:1319～1331.

15 Weyuker E J. Axiomatizing software test data adequacy. IEEE Trans. Software Eng., 1986,**12(12)**:1128～1138.

16 Parrish A, Zweben S. An analysis and refinement of software test data adequacy properties. IEEE Trans. Software Eng., 1991,**17(6)**:561～581.

17 Frankl P G, Weyuker E J. A formal analysis of the fault detecting ability of testing methods. IEEE Trans. Software Eng., 1993,**19(3)**:202～213.

18 Frankl P G, Weiss S N. An experimental comparison of the effectiveness of branch testing and data flow testing. IEEE Trans. Software Eng., 1993,**19(8)**:775～787.

19 Frankl P G, Weyuker E J. Provable improvements on branch testing. IEEE Trans. Software Eng., 1993,**19 (10)**:962～975.

20 暴建民,黄仲伟,杨孝宗. 分布式容错计算机通讯系统的实现. 小型微型计算机系统,1996,**17(6)**:59～63.

THE TWENTY YEARS FOR THE THEORY OF TEST DATA SELECTION

Bao Jianmin

(Institute of Applied Mathematics Heilongjiang University Harbin 150080)

Yang Xiaozong　Yang Nan　Wang Shenke

(*Department of Computer Science and Engineering　Harbin Institute of Technology　Harbin*　150006)

Abstract　It divides the twenty years of the theory of test data selection into three periods——formation period, development period, perfectness period, gives representive research results and characters of each period, and discusses the application of the theory in software engineering.

Key words　Theory of test data selection, adequacy criteria, specification.