第3卷第4期
1992年11月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 3, No. 4
Nov., 1992

从软件功能实例规格说明到软件功能形式规格说明的自动转换

吕 建 徐家福

(南京大学计算机软件研究所,南京 210008)

THE AUTOMATIC TRANSFORMATION FROM THE EXAMPLE SPECIFICATION TO THE FORMAL FUNCTIONAL SPECIFICATION

Lü Jian and Xu Jiafu

(*Institute of Computer Software, Nanjing University, Nanjing* 210008)

Abstract The acquisition of the formal functional specification is a very important in software automation. This paper presents a method which combines the concept formation and the antecedent derivation mechanism to accomplish the automatic transformation from the example specification to the formal functional specification and the plausibility of the generated formal specification is guaranteed in some sense.

摘要 软件功能形式规格说明的获取是软件自动化领域中十分重要的问题.本文采用概念学习与前件推导机制相结合的方法完成从软件功能实例规格说明到软件功能形式规格说明的自动转换,并能在某种意义下保证转换结果具有一定的合理性.

§1. 引 言

软件功能形式规格说明由于其语义的完备性和无歧义性而易于表达、分析和验证软件及其开发过程的各种性质,便于软件的形式化和自动化研究,如 Floyd 的归纳断言方法[1],Dijkstra 的 WP 方法[2]和 Z. Manna 的演绎综合方法[3]均以软件功能的形式规格说明作为基础.

然而,从实用的角度,由于形式性和易理解性难以共存而使得软件功能形式规格说明的获取并非易事.因此,软件功能形式规格说明的获取成了基于软件自动化的软件开发新风范(paradigm)投入实用的主要障碍之一[4],并成为软件自动化研究的首要问题[5].

本文采用概念学习与前件推导机制相结合的方法完成从易于书写、易于理解、由正负实例

本文1990年9月8日收到,1990年11月25日定稿.本文作者吕建,副教授,1988年博士毕业于南京大学,目前从事软件自动化,人工智能,软件方法学方面的研究工作.徐家福,教授,现任南京大学软件研究所所长,主要从事程序语言,软件自动化,新型程序设计,自然语言理解方面的研究工作.

集刻划的软件功能实例规格说明到由输入/输出断言刻划的软件功能形式规格说明的自动转换,并使所生成的软件功能形式规格说明具有一定的合理性.

§2. 问题描述

为讨论方便起见,我们以函数作为软件的数学模型,比较严格地定义了函数功能形式规格说明,函数功能实例规格说明和从软件功能实例规格说明到软件功能形式规格说明转换的含义.

定义 1:函数 F 的说明是一三元组,即 $FD(F)=<FM(F),DO_F,RA_F>$,其中,

$FM(F)$ 表示函数格式,其一般形式是 $Y=F(X)$,F 是函数名,X 是输入变量的元组,Y 是输出变量的元组,即 $X=(x_1,\cdots,x_m)$,$Y=(y_1,\cdots,y_n)(m,n\geqslant 1)$;

DO_F 表示函数 F 的输入的取值范围,其一般形式是 $D_1\times\cdots\times D_m$,每一 $D_i(1\leqslant i\leqslant n)$ 是高级语言中的数据类型;

RA_F 表示函数 F 的输出的取值范围,其一般形式是 $R_1\times\cdots\times R_n$,每一 $R_i(1\leqslant i\leqslant n)$ 是高级语言中的数据类型.

定义 2:函数 F 功能的形式规格说明是一二元组,即 $P(F)=<FD(F),SP_F>$,其中 $FD(F)$ 是函数 F 的说明;SP_F 是函数功能的形式描述,它包括 IC_F 和 OC_F 两部分;其中,

IC_F 是一谓词表达式,表示 DO_F 上的关系,记为 $IC_F(X)$,称为输入断言.凡满足输入断言的输入值称为合法输入;

OC_F 是一谓词表达式,表示 $DO_F\times RA_F$ 上的关系,记为 $OC_F(X,Y)$,称为输出断言;给定合法输入 X',凡满足 $OC_F(X',Y')$ 的输出 Y' 称为关于 X' 的合法输出.

定义 3:函数 F 功能的实例规格说明是一二元组,即 $E(F)=<FD(F),EX_F>$,其中,$FD(F)$ 是函数 F 的说明;EX_F 是函数功能的实例描述,它包括三部分:

$In\text{-}F$:函数输入的正负实例集;函数输入的正实例集的一般形式是 $In\text{-}F\text{-}P=\{P\text{-}X_i \mid P\text{-}X_i\in DO_F\wedge P\text{-}X_i$ 是合法输入$\}$;函数输入的负实例集的一般形式 $In\text{-}F\text{-}N=\{N\text{-}X_i\mid N\text{-}X_i\in DO_F\wedge N\text{-}X_i$ 是非法输入$\}$;要求 $In\text{-}F\text{-}P\cap In\text{-}F\text{-}N=\varnothing$;如果 $In\text{-}F\text{-}P=DO_F$,则用 $TRUE$ 表示.

$Out\text{-}F$:函数输出的正负实例集;函数输出的正实例集的一般形式是 $Out\text{-}F\text{-}P=\{P\text{-}Y_i\mid P\text{-}Y_i\in RA_F\wedge P\text{-}Y_i$ 是合法输出$\}$;函数输出的负实例集的一般形式是 $Out\text{-}F\text{-}N=\{N\text{-}Y_i\mid N\text{-}Y_i\in RA_F\wedge N\text{-}Y_i$ 是非法输出$\}$;要求 $Out\text{-}F\text{-}P\cap Out\text{-}F\text{-}N=\varnothing$;如果 $Out\text{-}F\text{-}P=RA_F$,则用 $TRUE$ 表示.

$Map\text{-}F$:输入输出关系的正负实例集;输入输出关系的正实例集的一般形式是 $Map\text{-}F\text{-}P=\{(P\text{-}X_i,P\text{-}Y_j)_P\mid P\text{-}X_i\in In\text{-}F\text{-}P\wedge P\text{-}Y_j\in Out\text{-}F\text{-}P\wedge P\text{-}Y_j=F(P\text{-}X_i)\}$;输入输出关系的负实例集的一般形式是 $Map\text{-}F\text{-}N=\{(P\text{-}X_i,P\text{-}Y_j)_N\mid P\text{-}X_i\in In\text{-}F\text{-}P\wedge P\text{-}Y_j\in RA_F\wedge P\text{-}Y_j\neq F(P\text{-}X_i)\}$;要求 $Map\text{-}F\text{-}P\cap Map\text{-}F\text{-}N=\varnothing$;

定义 4:从函数 F 功能的实例规格说明 $E(F)$ 到函数 F 功能的形式规格说明 $P(F)$ 的转换是指由 $E(F)$ 得到满足以下条件的 $P(F)$:

(1) $IC_F(X)$ 是恰当的谓词表达式,即

$$\begin{cases}\forall X\in DO_F(X\in In\text{-}F\text{-}P\Rightarrow IC_F(X))\\ \forall X\in DO_F(X\in In\text{-}F\text{-}N\Rightarrow \overline{IC}_F(X))\end{cases}$$

〈2〉$OC_F(X,Y)$是恰当的谓词表达式，即

$$\begin{cases}\forall X\in DO_F,\forall Y\in RA_F((X,Y)\in Map\text{-}F\text{-}P\Rightarrow OC_F(X,Y))\\ \forall X\in DO_F,\forall Y\in RA_F((X,Y)\in Map\text{-}F\text{-}N\Rightarrow \overline{OC}_F(X,Y))\end{cases}$$

〈3〉$IC_F(X)$与$OC_F(X,Y)$是一致的，即

$$\forall X\in DO_F,\exists Y\in RA_F(IC_F(X)\Rightarrow OC_F(X,Y))$$

以上三个条件在某种意义下保证所生成的函数功能形式规格说明的合理性；条件〈1〉说明了$IC_F(X)$是输入的正实例集的推广，但不能包括负实例集中的元素；条件〈2〉说明了$OC_F(X,Y)$是输入输出关系的正实例集的推广，但不能包括负实例集中的元素；条件〈3〉则说明了$IC_F(X)$与$OC_F(X,Y)$推广的程度必须一致.

§3. 转换方法

根据定义4，完成从函数功能实例规格说明$E(F)$到函数功能形式规格说明的转换要解决两方面的问题，其一是获取问题，即从$E(F)$获取相应的$P(F)$，本文采用概念学习的方法解决；其二是合理性验证问题，即对所生成的$P(F)$验证其合理性条件〈1〉,〈2〉,〈3〉均是成立的，对于条件〈1〉和〈2〉，它们由相应的概念学习过程来保证，而对于一致性问题，则需采用一定的自动演绎机制进行验证，本文采用前件推导机制.

1. 获取问题与概念学习

概念学习是人工智能中研究得十分广泛的一种归纳推理方法，它从某一概念的一组正负实例出发，形成此概念的一般描述.

形式地说，概念学习问题是一三元组〈P,N,L〉，其中，P是概念的正实例的集合，N是概念的负实例的集合，L是用来描述概念的语言；概念学习问题的解是满足以下条件的关系：

〈1〉它由语言L所表达；

〈2〉它由P中所有正实例所满足；

〈3〉它不被N中任一负实例所满足.

函数功能形式规格说明的生成主要是由$E(F)$获取$IC_F(X)$和$OC_F(X,Y)$. 对于$IC_F(X)$，它可归结为概念学习问题〈$In\text{-}F\text{-}P$,$In\text{-}F\text{-}N$,谓词表达式〉的求解；其合理性条件由概念学习问题的解的定义所保证；

而对于$OC_F(X,Y)$，一般说来，它可分解为两部分，其一是$OC_F^1(Y)$，它主要对输出Y自身的性质加以刻划，它可归结为概念学习问题〈$Out\text{-}F\text{-}P$,$Out\text{-}F\text{-}N$,谓词表达式〉的求解；其二是$OC_F^2(X,Y)$，它主要对X与Y的关系加以刻划，它可归结为概念学习问题〈$Map\text{-}F\text{-}P$,$Map\text{-}F\text{-}N$,谓词表达式〉的求解；结果$OC_F(X,Y)=OC_F^2(X,Y)\wedge OC_F(Y)$;$OC_F(X,Y)$的合理性可说明如下：任给$(\overline{X},\overline{Y})\in Map\text{-}F\text{-}P$，根据概念学习问题解的定义，有$OC_F^2(\overline{X},\overline{Y})$成立. 另外，根据$Map\text{-}F\text{-}P$的定义，$\overline{Y}\in Out\text{-}F\text{-}P$，所以$OC_F^1(\overline{Y})$成立，所以$OC_F^2(\overline{X},\overline{Y})\wedge OC_F^1(\overline{Y})$成立，另一方面，如果$(\overline{X},\overline{Y})\in Map\text{-}F\text{-}N$，那么由定义$\overline{OC}_F^2(\overline{X},\overline{Y})$，所以$\overline{OC}_F(\overline{X},\overline{Y})$成立.

2. 概念学习问题的求解

概念学习问题的求解采用解空间搜索方法[6]来完成，其基本点如下：

〈1〉解空间的概念

给定概念学习问题$\langle P,N,L\rangle$,定义其所有解构成的集合为解空间SP.对于解空间中的元素可定义直接包含关系如下:对于SP中的两个元素p,q,q直接包含p当且仅当(a)p是q的子关系(b)SP中不存在r,使得p是r的子关系而r也是q的子关系;q直接包含p记为$p\subset q$.由于直接包含关系是一偏序,因此,解空间可由有向无环的解图来刻划,其每一节点均是SP中的元素,节点p到q有弧当且仅当$p\subset q$.由直接包含关系可定义包含关系$p\subset\subset q$,$p\subset\subset q$当且仅当解图中有p到q的通路.

解空间SP中某一元素p是其极小元素当且仅当SP中不存在元素q使得$q\subset p$;解空间中某一元素r是其极大元素当且仅当SP中不存在元素s使得$r\subset S$;显然,解空间中的极大元素和极小元素并不是唯一的.

解空间SP是良构的当且仅当解图中每一直接包含关系所构成的链均有极大元素和极小元素,一般说来,如果SP是有限的,那么它是良构的.

对于良构的解空间,称其所有极小元素所构成的集合为其下限集S,所有极大元素所构成的集合为其上限集G;上限集和下限集具有这样的性质:解空间中任一元素均可由上限集或下限集中的元素通过直接包含关系而得到.

(2)求解过程

给定概念学习问题$\langle P,N,L\rangle$和其上限集G与下限集S,那么

a. $<P\cup\{x\},N,L>$的上限集$pg(x,S,G)$和下限集$PS(x,S,G)$可按下法求得;

Ⅰ. $pg(x,S,G)=\{g\in G|g(x)\}$

Ⅱ. $ps(x,S,G)=\{r|pup(x,S,G,r)\}$

其中,$pup(x,S,G,r)$成立当且仅当

* r是S中某一元素或包含S中的某一元素,且
* r被$pg(x,S,G)$中的某一元素所包含,且
* r被正实例所满足,即$r(x)$成立,且
* 不存在被r所包含的关系r_1,它同时具有以上三个性质.

b. $<P,N\cup\{x\},L>$的下限集$ns(x,S,G)$和上限集$np(x,S;G)$的求法如下;

Ⅰ. $ns(x,S,G)=\{s\in S|\bar{s}(x)\}$

Ⅱ. $ng(x,S,G)=\{r|nup(x,S,G,r)\}$

其中,$nup(x,S,G,r)$成立当且仅当

* r是G中某一元素或被G中某一元素所包含;且
* r包含$ns(x,S,G)$中的某一元素;且
* r不被负实例所满足,即$\bar{r}(x)$成立.
* 不存在包含r的关系r_1,它同时具有以上三个性质.

而概念学习问题的求解过程是:在一定的背景知识的指导下,根据所给的正负实例集对上限集和下限集进行不断修改的过程,直到上限集与下限集均为单一的、相同的关系构成为止,此关系即为概念学习问题的解.

3. 一致性验证

由于函数功能形式规格说明的输入断言$IC_F(X)$和输出断言$OC_F(X,Y)$是由三个不同的概念学习问题的解组合而成的,因此,有可能会使得$IC_F(X)$与$OC_F(X,Y)$的一般化的程度不

一致而导致函数功能形式规格说明的不一致性，即存在满足 $IC_F(X)$ 的合法输入 $\overline{X}$，而在 RA_F 中不存在相应的 $\overline{Y}$，使得 $OC_F(\overline{X},\overline{Y})$ 成立；因此，所生成的函数功能形式规格说明的验证是十分重要的.

根据定义4，所谓一致性验证就是采用自动演绎机制说明 $\forall X\in DO_F(IC_F(X)\Rightarrow\exists Y\in RA_F\ OC_F(X,Y))$ 成立；传统的定理证明技术的不足之处在于如果一致性条件不成立，那么，它便不能提供任何有用的信息对所生成的 $IC_F(X)$ 和 $OC_F(X,Y)$ 加以调整，为此，我们采用了前件推导机制.

在一阶理论中，前件可定义如下：设有两公式 $\forall x_1,\cdots,x_n[P\Rightarrow Q]$ 和 $A(x_1,\cdots,x_i)$，其中，$A(x_1,\cdots,x_i)$ 仅包含自由变量 $x_1,\cdots,x_i(1\leqslant i\leqslant n)$；公式 $A(x_1,\cdots,x_i)$ 是 $\forall x_1,\cdots,x_n[P\Rightarrow Q]$ 的前件当且仅当 $\forall x_1,\cdots,x_n[P\Rightarrow(A(x_1,\cdots,x_i)\Rightarrow Q)]$ 成立. 在实际使用中，一般要求所导出的前件尽量的弱和简单.

D. R. Smith[7]提出了导出前件的形式系统，我们已在 Sun-3 工作站上实现了这一系统并应用于算法设计自动化系统 NDADAS[8]之中.

使用前件推导机制来验证一致性条件会出现两种情况：其一是一致性条件成立，即公式 $\forall X\in DO_F[IC_F(X)\Rightarrow\exists Y\in RA_F\ OC_F(X,Y)]$ 的 $\{X\}$－前件为 $TRUE$；其二是一致性条件不成立，即公式 $\forall X\in DO_F[IC_F(X)\Rightarrow\exists Y\in RA_F\ OC_F(X,Y)]$ 的 $\{X\}$－前件为某一公式 $J(X)$，那么根据前件的定义可得 $\forall X\in DO_F[IC_F(X)\Rightarrow(J(X)\Rightarrow\exists Y\in RA_F\ OC_F(X,Y)]$ 成立，即 $\forall X\in DO_F[IC_F(X)\wedge J(X)\Rightarrow\exists Y\in RA_F\ OC_F(X,Y)]$ 成立，因此，可采用 $IC_F(X)\wedge J(X)$ 作新的输入断言，问题是验证合理性条件〈1〉对于新的输入断言是否成立. 由于 $IC_F(X)$ 的合理性已得到验证，因此只要验证所有的正实例均满足 $J(X)$ 即可. 此验证是比较简便的.

另一方面，用户可能只给出 *Out-F* 和 *Map-F*，由系统求出 $OC_F(X,Y)$ 后，由前件推导机制给出 $IC_F(X)$.

§4. 举 例

下面给出一个简单的实例来说明以上转换方法的使用.

1. 函数功能实例规格说明

y＝F(x)

DO_F	:SEQ(INT)(整型序列)
RA_F	:INT
In-F-P	:TRUE
Out-F-P	:TRUE
Map-F-P	:<3,2,1>→3
	<3,4,1>→4
Map-F-N	:<3,2,1>→2
	<3,4,1>→5

其中，In-F-P 和 Out-F-P 均为 TRUE 反映了用户在开始考虑问题时的重点在于输入输出关系.

2. 转换过程

1〉背景知识

谓词表达式::=基本谓词|谓词表达式∧谓词表达式

基本谓词::=In(e,l)|Out(e,l)|

Grt(e,l)|Bnd(e,l)

(其中,e表示整型变量,l表示整型序列)

基本知识

$$\forall l\in SEQ(INT)\forall e\in INT(\exists e_1\in l(e_1=e))\Rightarrow In(e,l)$$

$$\forall l\in SEQ(INT)\forall e\in INT(\forall e_1\in l(e_1\neq e))\Rightarrow Out(e,l)$$

$$\forall l\in SEQ(INT)\forall e\in INT(\forall e_1\in l(e_1\leqslant e))\Rightarrow Grt(e,l)$$

$$\forall l\in SEQ(INT)\forall e\in INT(\exists e_1\in l(e_1> e))\Rightarrow Bnd(e,l)$$

2)输入输出关系的归纳

对每一实例,按其特征向量〈属于关系,大小关系〉分析,然后用概念学习方法进行归纳.

取第一个正实例〈3,2,1〉→〈3〉,其特征向量为〈$\exists e_1\in\langle 3,2,1\rangle(e_1=3)$,$\forall e_1\in\langle 3,2,1\rangle(e_1\leqslant 3)$〉,与之相应的解图为:

$$\begin{tikzcd}[column sep=tiny]
 & & \langle \exists e_1\in\langle 3,2,1\rangle(e_1=3), \forall e_1\in\langle 3,2,1\rangle(e_1\leq 3)\rangle \arrow[dl] \arrow[dr] & & \\
 & \langle In(y,x), \forall e_1\in\langle 3,2,1\rangle(e_1\leq 3)\rangle \arrow[dl] \arrow[dr] & & \langle \exists e_1\in\langle 3,2,1\rangle(e_1=3), Grt(y,x)\rangle \arrow[dl] \arrow[dr] & \\
\langle ANY, \forall e_1\in\langle 3,2,1\rangle(e_1\leq 3)\rangle \arrow[dr] & & \langle In(y,x), Grt(y,x)\rangle \arrow[dl] \arrow[dr] & & \langle \exists e_1\in\langle 3,2,1\rangle(e_1=3), ANY\rangle \arrow[dl] \\
 & \langle ANY, Grt(y,x)\rangle \arrow[dr] & & \langle In(y,x), ANY\rangle \arrow[dl] & \\
 & & \langle ANY, ANY\rangle & &
\end{tikzcd}$$

取第二个正实例〈3,4,1〉→4,其特征向量为〈$\exists e_1\in\langle 3,4,1\rangle(e_1=4)$,$\forall e_1\in\langle 3,4,1\rangle(e_1\leqslant 4)$〉,去掉上图中不包含此特征向量的节点可得如下解图:

$$\begin{tikzcd}
 & \langle In(y,x), Grt(y,x)\rangle \arrow[dl] \arrow[dr] & \\
\langle ANY, Grt(y,x)\rangle \arrow[dr] & & \langle In(y,x), ANY\rangle \arrow[dl] \\
 & \langle ANY, ANY\rangle &
\end{tikzcd}$$

取第一个负实例〈3,2,1〉→〈2〉,其特征向量为〈$\exists e_1\in\langle 3,2,1\rangle(e_1=2)$,$\exists e_1\in\langle 3,2,1\rangle(e_1>2)$〉,去掉上图中包含此特征向量的节点可得下图:

$$\begin{tikzcd}
\langle In(y,x), Grt(y,x)\rangle \arrow[d] \\
\langle ANY, Grt(y,x)\rangle
\end{tikzcd}$$

取第二个负实例〈3,4,1〉→〈5〉,其特征向量为〈$\forall e_1\in\langle 3,4,1\rangle(e_1\neq 5)$,$\forall e_1\in\langle 3,4,1\rangle(e_1\leqslant 5)$〉,去掉上图中包含此特征向量的节点可得:

$$\langle In(y,x), Grt(y,x)\rangle$$

因此，$OC_F^2(x,y)=In(y,x)\wedge Grt(y,x)$

即通常意义下 $OC_F(x,y)=y\in x\wedge y\geqslant all(x)$

结果有 $IC_F(X)=TRUE$，$OC_F(x,y)=TRUE\wedge y\in x\wedge y\geqslant all(x)=y\in x\wedge y\geqslant all(x)$

3）一致性验证

采用前件推导机制导出 $\forall\ x\in SEQ(INT)\exists\ y\in INT(TRUE\Rightarrow(y\in x\wedge y\geqslant all(x)))$，即 $\forall\ x\in SEQ(INT)\exists\ y\in INT(y\in x\wedge y\geqslant all(x))$ 的 $\{x\}$－前件，其结果为 $x\neq\varnothing$；

因此，最终所生成的函数功能形式规格说明应为

$$
\begin{aligned}
&y=F(x)\\
&DO_F:SEQ(INT)\\
&RA_F:INT\\
&IC_F(x):x\neq\varnothing\\
&OC_F(x,y):y\in x\wedge y\geqslant all(x)
\end{aligned}
$$

即 F 是一求整型序列最大值的函数.

§5. 讨 论

通常，以输入输出实例作为规格说明手段的程序综合方法是直接从输入输出实例综合出高级语言程序，其理论基础是归纳推理. 一般说来，归纳推理要解决两方面的核心问题，其一是归纳结论的生成，其二是归纳结论合理性的确认. 目前，人们普遍认为，计算机可解决第一方面的问题，即归纳结论的自动生成，而归纳结论的确认则必须由人完成. 这就要求所生成的归纳结论应是一种从语法和语义上均接近于人的思维的符号描述，并且描述分量应易于理解而且结构手段简明易读，这便是归纳推理中十分重要的易理解性假设(Comprehensibility postulate)[9].

从以上观点来看从输入输出实例直接归纳出高级语言程序的方法，其归纳确认过程可归为以下图式：

$$\text{输入输出实例}\xrightarrow[\text{归纳}]{}\text{程序}\xrightarrow[\text{理解}]{}\text{功能描述}\xrightarrow[\text{确认}]{}\text{结果}$$

而任给一程序，要完全理解其功能往往是比较困难的，特别当所生成程序的结构较差时，更加深了这方面的困难. 本文的另一目的就是试图将归纳结论的确认过程从程序级提高到形式化功能规格说明级：

$$\text{输入输出实例}\xrightarrow[\text{归纳}]{}\text{形式化功能规格说明}\xrightarrow[\text{确认＋自动转换}]{}\text{程序}$$

由于形式化功能规格说明可借助于各种自动演绎机制分析和验证其各种性质，比程序有较好的易理解性，并且从形式化功能规格说明到可执行程序的自动转换可由各种算法设计自动化系统(如 NDADAS 系统)完成，且可保证所生成的程序相对于形式化功能规格说明是正确的；因此，可以提高最终所生成的程序的合理性.

本文给出了采用概念学习与前件推导机制相结合的方法，完成从软件功能实例规格说明到软件功能形式规格说明自动转换的基本思想，其主要目的在于探讨软件功能形式规格说明的自动获取问题与提高基于输入输出实例的程序综合方法的确认级别. 进一步的工作包括：研

究各种数据结构的性质及其相互关系,在已实现的前件推导机制的基础上构作实验性转换系统,并将之与算法设计自动化系统 NDADAS 相连,通过实验改进、提高和完善本文所提出的方法.

参考文献

[1]R. W. Floyd, Assigning Meanings to Programs, Proc, Amer. Math. Soc. Symp. in Applied Mathematics, 19 (1967), 19-31.

[2]E. W. Dijkstra, A Discipline of Programming, Prentice-Hall, 1976.

[3]Z. Manna, R. Waldinger, A Deductive Approach to Program Synthesis, ACM Trans. on Programming Language and Sys. 2(1), 1980, 90-121.

[4]W. W. Agresti, What are the New Paradigms? W. W. Agresti(ed) New Paradigms for Software Development, 1986.

[5]R. Balzer, A 15 Year Perspective on Automatic Programming, IEEE Trans. on Software Engineering, Vol. SE-11, No. 11, 1985, 1257-1267.

[6]M. R. Genesereth, N. J. Nilsson, Logical Foundations of Artificial Intelligence, Morgan Kaufmann Publisher Inc. 1987.

[7]D. R. Smith, Derived Preconditions and Their Use in Program Synthesis, D. W. Loveland(ed), Six Conf. on Automated Deductions, Lecture Notes in Computer Science 138.

[8]吕建,算法设计自动化系统 NDADAS 的设计与实现,南京大学博士论文.

[9]R. S. Michalski, A Theory and Methodology of Inductive Learning, R. S. Michalski, etc(ed) Machine Learning, An Artificial Intelligence Approach, 1983.