# 一种基于规则的语言的公理语义

魏振春[1,2],韩江洪[1,2],陆 阳[1,2],刘小平[1]
WEI Zhen-chun[1,2],HAN Jiang-hong[1,2],LU Yang[1,2],LIU Xiao-ping[1]

1.合肥工业大学 计算机与信息学院,合肥 230009
2.教育部安全关键工业测控技术工程研究中心,合肥 230009
1.School of Computer & Information,Hefei University of Technology,Hefei 230009,China
2.Engineering Research Center of Safety Critical Industrial Measurement and Control Technology,Ministry of Education,Hefei 230009,China
E-mail:wzc@ialab.hfut.edu.cn

**WEI Zhen-chun,HAN Jiang-hong,LU Yang,et al.Axiomatic semantics of rule-based language.Computer Engineering and Applications,2008,44(20):86-88.**

**Abstract:** In order to exactly describe logic relations among objects of discrete event control systems and write programs,a rule-based language,Logic Rule Description Language (LRDL) is put forward.Formal syntax of LRDL is defined by EBNF.Axiomatic semantics of LRDL is presented and proved formally based on axiomatic system of Hoare logic,which offers theoretical basis for the proof of the correctness of programs written with LRDL.

**Key words:** rule;Logic Rule Description Language(LRDL);axiomatic semantics;Hoare logic;formal syntax

**摘 要:**为了准确描述离散事件控制系统对象之间的逻辑关系和编写控制程序,提出了一种基于规则的语言——逻辑规则描述语言(LRDL)。用 EBNF 给出了 LRDL 的语法定义,基于 Hoare 逻辑的公理系统,形式化地给出并证明了 LRDL 的公理语义,为用 LRDL 编写的程序的正确性证明提供了理论依据。

**关键词:**规则;逻辑规则描述语言;公理语义;Hoare 逻辑;形式语法
**DOI:**10.3778/j.issn.1002-8331.2008.20.027 **文章编号:**1002-8331(2008)20-0086-03 **文献标识码:**A **中图分类号:**TP301.2

## 1 引言

离散事件控制系统(DECS,Discrete Event Control Systems)是一类广泛存在于工业控制系统中,由离散事件驱动,并由离散事件按照一定运行规则相互作用,来导致状态演化的动态控制系统。DECS 的基本处理对象是系统状态和即时事件,其动态运行过程建立在系统各对象的逻辑关系基础上,对系统行为进程起决定作用的是一系列离散事件,系统所遵循的是复杂的人为规则。

产生式规则表示法是由美国数学家波斯特(Post)于 1943 年提出的,它依据人类大脑记忆模式中各种知识块之间大量存在的因果关系,以 IF…THEN…的形式表示出来[1]。产生式规则表示法形式单一,便于理解,规则之间相互独立,规则的结构化较好,有利于知识的提取和形式化,其求解问题的过程同人类的认知过程相似,因而广泛应用于各类专家系统及人工智能、模糊控制等领域。

结合 DECS 的特点和产生式规则表示法的优点,提出了 DECS 的规则化描述方法[2,3],采用产生式规则表示 DECS 对象之间的逻辑关系,DECS 的控制过程即基于规则的系统[4]的运行过程。为了准确描述系统对象之间的逻辑关系和编写 DECS 的控制程序,提出了一种基于规则的语言——逻辑规则描述语言(LRDL,Logic Rule Description Language)。用 EBNF 给出了 LRDL 的语法定义,借助 Hoare 逻辑的公理系统,形式化地给出并证明了 LRDL 的公理语义。

## 2 LRDL 的形式语法

语法语义是程序设计语言的基础,语法规定合法程序的形式,即表示构成语言的各个记号之间的组合规律。用 BNF(Backus-Naur Form)的扩展形式 EBNF 给出 LRDL 的形式语法定义如下(限于篇幅,这里只给出了部分语法):

<程序>::=<单条语句>|<语句块>|<单条语句><程序>|<语句块><程序>
<语句块>::=<块起始符>{<单条语句>}+<else 语句>
<单条语句>::=<左件>→<右件>;
<else 语句>::=<else 因子>→<右件>;|<else 因子>→;
<块起始符>::=@

**基金项目:**国家教育部博士点基金(the Fund for the Doctoral Program of the Ministry of Education of China under Grant No.20050359004);国家教育部新世纪优秀人才计划项目(the New Century Excellent Talent Foundation from MOE of China under Grant No.NCET-04-050562);安徽省科技公关计划项目(the Key Technologies R&D Program of Anhui Province,China under Grant No.06012069B)。
**收稿日期:**2007-10-08 **修回日期:**2007-12-24

<左件>::=<左件因子>{&<左件因子>}*

<右件>::=<右件因子>{&<右件因子>}*

<因子>::=　[<因子名>:<因子值>]

语句是语言的基本组成单元，也是运行时的基本处理单元。在LRDL中,每个语句都是以一条规则的形式出现,所以这样的语句又称为逻辑规则表达式,简称规则式或规则。每个语句都由左件和右件两部分组成,左件相当于条件,右件相当于动作。每个语句都相当于一个蕴含式"左件→右件",其逻辑含义是:当左件满足,即左件的逻辑值为真时,执行右件的动作。语句分为单条语句和else语句两大类,else语句相当于左件为else因子的一类特殊语句。程序中可以包含单条语句,也可以包含由一组语句组成的语句块。语句块由一组规则组成,也被称为规则组。语句块以块起始符开始,以else语句结束,中间包含一条或一条以上单条语句。一个语句块中的语句通常用来描述同一个被控对象,即同一个语句块中的语句右件拥有相同的因子名。每个语句块中,当else语句之前的所有语句的左件都没有成功匹配(逻辑值为假),则else语句的右件被执行。else语句可以为空,即else语句的右件为空。单条语句又称为"确定条件语句",因为语句中的条件是明确给出的,只要满足条件该条语句即被执行;else语句又称为"剩余条件语句",只有语句块中所有的单条语句都不满足时,else语句才被执行。在规则化描述方法中引入规则组,将对同一对象的控制规则组合在一起构成规则组,既保证了逻辑的完备性,又可以实现表达的简洁性和执行的高效性。每个语句都以分号";"结尾。因子是逻辑规则表达式的基本组成单元,因子由因子名和因子值两部分组成,分别代表一个对象和其状态值[3]。

## 3　公理语义与Hoare逻辑

语法正确的语句,其语义未必正确。语义规定符合语法的程序的意义,即在执行时所产生的效果,语义远比语法复杂。对程序语义的研究已经取得了许多成果[5,6],形式语义学研究如何用严格的形式化的方式,定义程序设计语言的语义。

公理语义学是形式语义学的四大分支之一,公理语义学通过使用数学中的公理化方法,用公理系统定义程序设计语言的语义。公理语义渊源自Floyd在1967年发表的论文"Assigning meaning to programs"。Hoare在1969年发表的文章"An axiomatic basis for computer programming"中提出了一种改进的逻辑语义方法,被称为Hoare逻辑。

公理语义对程序设计语言中的每个成份定义了一对断言(Assertion):前置断言(Precondition)和后置断言(Postcondition),也称前置条件和后置条件。前置断言是某个语言成份在执行前满足的谓词逻辑公式(或条件),而后置断言则是该语言成份执行后应该满足的谓词逻辑公式(或条件)。

Hoare逻辑中的逻辑公式形式为(Hoare三元组):

$\{P\}Q\{R\}$

其中$P$为前置条件,$Q$代表一个程序或程序中的一条语句,$R$为后置条件。它的含义是:如果在程序(或语句)$Q$执行前,前置条件$P$成立,且$Q$的执行能终止,则在$Q$执行后,必有后置条件$R$成立。这也就是$Q$的部分正确性。

Hoare逻辑的四条基本公理是赋值公理、条件公理、组合公理和循环公理[5,6]。

(1)赋值公理

假定赋值语句取如下形式:

$$x:=e \tag{1}$$

赋值公理如下:

$$\{P[e/x]\}x:=e\{P\} \tag{2}$$

其中$P[e/x]$表示把谓词$P$中$x$的所有自由出现均代之以$e$,并把$P$中所有约束变量换成$e$中没有的名字。

(2)条件公理

条件语句:

$$\text{if } B \text{ then } Q1 \text{ else } Q2 \tag{3}$$

的公理形式是

$$\frac{\{P\wedge B\}Q1\{R\},\{P\wedge\neg B\}Q2\{R\}}{\{P\}\ \text{if } B \text{ then } Q1 \text{ else } Q2\{R\}} \tag{4}$$

其含义是,若横线上面的断言都成立,则横线下面的断言也成立。横线上面以逗号分开的诸断言按合取计算。

(3)组合公理

计算机语言的程序一般都是由多条语句组合而成,因此有语句组合问题。不失一般性,考虑两个语句的组合公理:

$$\frac{\{P\}Q1\{R\},\{R\}Q2\{T\}}{\{P\}Q1;Q2\{T\}} \tag{5}$$

这里的分号表示两个语句的组合。

(4)循环公理

循环语句有多种形式，这里采用一种形式作为代表,即while语句:

$$\text{while } B \text{ do } Q \tag{6}$$

其中$B$表示条件,$Q$表示语句。其公理形式为:

$$\frac{\{P\wedge B\}Q\{R\}}{\{P\}\ \text{while } B \text{ do } Q\{R\wedge\neg B\}} \tag{7}$$

证明中常用的公理还包括推断公理、析取公理和合取公理。

(5)推断公理

$$\frac{T\Rightarrow P,\{P\}Q\{R\},R\Rightarrow S}{\{T\}Q\{S\}} \tag{8}$$

(6)析取公理

$$\frac{\{P1\}Q\{R\},\{P2\}Q\{R\}}{\{P1\vee P2\}Q\{R\}} \tag{9}$$

(7)合取公理

$$\frac{\{P\}Q\{R1\},\{P\}Q\{R2\}}{\{P\}Q\{R1\wedge R2\}} \tag{10}$$

公理语义学是程序正确性研究的理论基础。利用Hoare逻辑,可以证明程序的正确性,也即证明Hoare三元组中的程序语句"符合"前后条件的要求。如果在前置条件满足的情况下执行语句,而且语句的执行终止,则后置条件满足。这个证明并不保证语句终止,因此,这样证明的正确性称为"部分正确性"。程序的终止性需要另外证明。如果证明了程序的部分正确性,又证明了终止性,那么程序就是"完全正确的"。

## 4　LRDL的公理语义

LRDL中的语句和Hoare逻辑中的逻辑公式存在着内在联系。LRDL中的语句由左件和右件两部分组成,左件相当于条件,右件相当于动作/结论,当左件满足时,执行右件的动作。右件因子本身以"[因子名:因子值]"这样断言的方式给出,其实质是给出了执行右件后相应因子名所对应的对象所处的状态或所满足的条件。因此,每条LRDL的语句对应于Hoare逻辑中的一个逻辑公式,LRDL语句的左件对应于Hoare逻辑公式中的前置条件,LRDL语句的右件对应于Hoare逻辑公式中的后置条件。借助Hoare逻辑的公理系统,可以形式化地表示LRDL

的公理语义。

(1)单条语句规则

$$\{P\}P\rightarrow R\{R\} \tag{11}$$

其含义是:若规则 $P\rightarrow R$ 执行前 $P$ 满足,则规则 $P\rightarrow R$ 执行后 $R$ 满足。根据规则的定义,该断言成立。

(2)规则组

$$\{P\}G\{R\} \tag{12}$$

其中 $G$ 为规则组。

从两个方面进行论述。

不失一般性,假设规则组 $G$ 为:

$G$=($P1\rightarrow R1$

$P2\rightarrow R2$

else$\rightarrow R3$)

①规则组相当于条件语句的嵌套

规则组 $G$ 表示为条件语句的嵌套,相当于

if $P1$

then $R1$

else if $P2$

then $R2$

else $R3$

然后运用条件公理进行处理。

②将 else 语句等价转换为确定条件规则

根据规则组的语法和语义定义,$G$ 中的 else 语句

else$\rightarrow R3$

等价于确定条件规则

$\neg P1\wedge\neg P2\rightarrow R3$

此时规则组 $G$ 可以等价转换为 $G'$

$G'$=($P1\rightarrow R1$

$P2\rightarrow R2$

$\neg P1\wedge\neg P2\rightarrow R3$)

$G'$中的规则都是单条语句规则。

(3)析取条件

$$\frac{\{P1\}P1\rightarrow R\{R\},\{P2\}P2\rightarrow R\{R\}}{\{P1\vee P2\}P1\vee P2\rightarrow R\{R\}} \tag{13}$$

**证明** 由已知,

$\{P1\}P1\rightarrow R\{R\}$

$\{P2\}P2\rightarrow R\{R\}$

分别在两式中增加加强条件:

$\{P1\}(P1\rightarrow R)\wedge(P2\rightarrow R)\{R\}$

$\{P2\}(P1\rightarrow R)\wedge(P2\rightarrow R)\{R\}$

进行等价转换(根据$(P1\rightarrow R)\wedge(P2\rightarrow R)\Leftrightarrow P1\vee P2\rightarrow R$),

$\{P1\}P1\vee P2\rightarrow R\{R\}$

$\{P2\}P1\vee P2\rightarrow R\{R\}$

运用析取公理,

$\{P1\vee P2\}P1\vee P2\rightarrow R\{R\}$

得证。

(4)合取结论

$$\frac{\{P\}P\rightarrow R1\{R1\},\{P\}P\rightarrow R2\{R2\}}{\{P\}P\rightarrow R1\wedge R2\{R1\wedge R2\}} \tag{14}$$

**证明** 由已知,

$\{P\}P\rightarrow R1\{R1\}$

$\{P\}P\rightarrow R2\{R2\}$

分别在两式中增加加强条件:

$\{P\}(P\rightarrow R1)\wedge(P\rightarrow R2)\{R1\}$

$\{P\}(P\rightarrow R1)\wedge(P\rightarrow R2)\{R2\}$

进行等价转换(根据$(P\rightarrow R1)\wedge(P\rightarrow R2)\Leftrightarrow P\rightarrow R1\wedge R2$),

$\{P\}P\rightarrow R1\wedge R2\{R1\}$

$\{P\}P\rightarrow R1\wedge R2\{R2\}$

运用合取公理,

$\{P\}P\rightarrow R1\wedge R2\{R1\wedge R2\}$

得证。

## 5 结束语

语法语义是程序设计语言的基础,本文提出的 LRDL 具有简洁、规范的语法形式和清晰、严格的语义描述。文中用 EBNF 给出了 LRDL 的形式语法,借助 Hoare 逻辑的公理系统,形式化地给出并证明了 LRDL 的几种主要语句形式:单条语句规则、规则组、析取条件规则和合取结论规则的公理语义,从而为用 LRDL 编写的程序的正确性证明提供了理论依据。

## 参考文献:


[1] Giarratano J C,Riley G D.Expert systems:principles and programming[M].4th ed.北京:机械工业出版社,2005.

[2] Wei Zhenchun,Han Jianghong,Lu Yang,et al.Hierarchical modeling and control of discrete event control systems based on rule description method[C]//Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation,2006:2179-2183.

[3] 韩江洪,魏振春,张本宏,等.总线式车身控制系统的规则化建模方法[J].汽车工程,2006,28(12):1121-1124.

[4] Hayes-Roth F.Rule-based systems[J].Communications of the ACM,1985,28(9):921-932.

[5] 陆汝钤.计算机语言的形式语义[M].北京:科学出版社,1992.

[6] Winskel G.程序设计语言的形式语义[M].宋国新,邵志清,译.北京:机械工业出版社,2004.


(上接 72 页)


[7] Chen Shyi-Ming.Measures of similarity between vague sets[J].Fuzzy Sets and Systems,1995,74(2):217-223.

[8] Chen Shyi-Ming.Similarity measure between Vague sets between elements[J].IEEE Trans on Systems,Man and Cybernetics,1997,27(1):153-158.

[9] Hong D H,Kimc.A note on similarity measures between Vague sets and between elements[J].Information Sciences,1999,115:83-96.

[10] Szmidt E,Kacprzyk J.Distances between intuitionistic fuzzy sets[J].Fuzzy Sets and Systems,2000,114(3):505-518.

[11] Li D F,Cheng C T.New similarity measures of intuitionistic fuzzy sets and application to pattern recognitions[J].Pattern Recognition Letters,2002,23(13):221-225.

[12] 李凡,吕泽华,蔡立晶.基于 Fuzzy 集的 Vague 集的模糊熵[J].华中科技大学学报,2003,31(2):1-3.