主动数据库中基于图的规则的指称语义研究*)

Denotational Semantics for Graph-Based Rules in Active Databases

徐长醒　刘云生　许贵平

(华中理工大学计算机学院　武汉430074)

Abstract　Many active database systems propose various rule systems, which are normally described in an informal manner. The informal description makes it difficult to understand the behavior of rules and the differences of various rule systems. In this paper, we describe a denotational semantic for a kind of graph-based rule in an active database. The kind of active rules had been formally defined and analyzed in our previous paper, their execution may be both inter-rule and intra-rule parallelism through a nested transaction model.

Keywords　Active database, Active rule, Denotational semantics, Parallel

1　引言

主动数据库的主动特性一般使用 E-C-A(Event-Condition-Action)规则模型描述。我们用 CA 规则表示 ECA 规则模型中的 Condition 和 Action 部分，ECA 规则语言的语义可描述为：事件 E 的发生可触发 CA 规则的执行，该 CA 规则的执行又可导致其它事件的发生，进而触发其它 CA 规则的执行，形成触发规则集。主动规则源于人工智能的知识表示的产生式系统(OPS5[1])，许多主动数据库规则系统的执行语义均源于 OPS5产生式规则语言的 recognize-act cycle 算法(如图1)。

```
initial match(test rule conditions)
repeat until no rules match or halt is executed
    perform conflict resolution (pick a triggered rule)
    act (execute the rule's action)
    match(test rule conditions)
end
```

图1　recognize-act cycle 算法

该算法显示了专家系统中规则的处理过程：在匹配步，确定规则的条件是否满足当前工作存储器中的事实数据，如果存在事实满足该条件，则相应的规则被激活或触发，被触发的所有规则集形成冲突集；在冲突消解步，使用冲突消解策略从冲突集中选出一合适的规则；在执行步，执行被选出规则；规则的执行又将改变工作存储器中的事实数据，从而触发新的规则，以上三步循环处理直至没有新的规则被触发执行。

上述的冲突消解策略强制规则串行执行，而支持规则并发执行的规则系统是系统的必然发展。在文[2,3]中，我们研究了一类基于图的主动规则模型 E-RG(Event-Rule Graph)，其中，规则图包含 CA 规则和由 CA 规则间的基本时序关系导出的控制结构。该规则模型支持两类并发性，即规则间的并发性和规则内的并发性[3,4]。E-RG 规则模型的应用优势包括：(1)由最基本的时间关系引入了规则图 RG 中的控制结构(S、Y 和 P 关系)，易于保证规则集的"终止性"；(2)E-RG 规则由 CA 规则构成，CA 规则可并发执行，这表现为规则内的并发性；(3)相同的 CA 规则集使用不同的控制结构可对应于不同的规则图 RG，从而增加了 CA 规则使用的灵活性；(4)E-RG 规则执行模型需要非平坦的嵌套事务模型，多个 E-RG 规则可并发地执行，这表现为规则间的并发性。

本文先介绍 E-RG 规则模型的语法和非形式语义，在此基础上研究 E-RG 规则语言的形式语义，它是正确实现和理解规则语言的关键，由于其重要性，规则语言的形式语义的研究已成为数据库领域的一个热点。文[5,6]中概述了相关研究概况。

2　基于图的规则模型

本节首先简要介绍 E-RG 规则模型中的规则图(RG)的概念，详细的说明请参见文[3]。然后，介绍 E-RG 规则的语法和非形式语义，作为下节研究其形式语义的基础。

2.1　规则图(RG)

在实际应用中，同一事件可触发多个 CA 规则，我们将 CA 规则作为结点，并由 CA 规则间的基本执行时序导出的"规则关系"作为边，构成"规则图"。"规则关系"包括"依次关系"S、"同步关系"Y 与"并发关系"P(Parallel)[3]。在此基础上，形式地定义了"规则时序图"(图2)和"规则图"(图3)的概念。

规则时序图是有限的有向无环图。规则时序图通过文[3]给出的转换算法可转换为语义无损的规则图。图3所示规则图的含义为：规则 r_1 先于 r_2、r_3、r_4 执行，r_1 结束后，r_2、r_3、r_4 可并发执行；r_2 先于 r_5 执行；r_2、r_3、r_4 的执行在 r_6 同步，即 r_2、r_3、r_4 都完成才能执行 r_6。

2.2　E-RG 主动规则模型

基于图的主动规则模型 E-RG 可形式地描述为三元组(Event, RG, Coupling)。E-RG 规则的执行语义简单地说就是：事件 E 的发生触发规则图 RG 的执行，RG 的执行满足 RG 所含的时序关系。规则事件包括"原语事件"和"复杂事件"，执行的耦合方式(Coupling modes)在主动数据库系统 HiPAC[7]中首先被提出，包括'立即'、'推迟'和'分离'方式，前二种方式被触发的规则作为触发事务(或规则)的子事务，第三种方式形成独立的事务。

多样的触发耦合方式提供了丰富的执行策略，形式地描述所有的耦合方式超出了本文的范围，我们主要刻画"推迟"

*)国家自然科学基金资助项目：60073045。徐长醒　博士生，主要研究方向为现代数据库技术及实时嵌入式操作系统。刘云生　教授，博士生导师。许贵平　博士生。

耦合方式的 E-RG 规则的指称语义。作为下节描述的基础，我们在此给出 E-RG 规则的定义形式(图4)，为直观起见，定义中使用的规则图对应于图3所示的规则图。

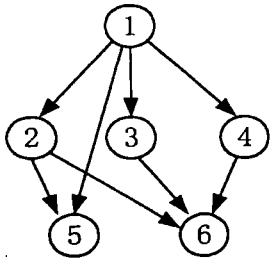

图2　规则时序图

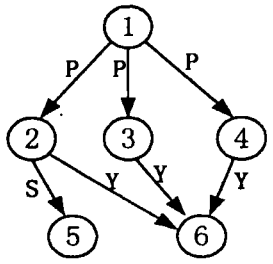

图3　规则图

```
create rule rule-name
on event
rule-graph
    r1:
        P:r2,r3,r4;
    r2:
        S:r5;
    r6:
        Y:r2,r3,r4;
end rule-name
```

图4　E-RG 规则

3　主动规则 E-RG 的指称语义

在指称语义理论中，规则执行的形式语义使用语义函数描述，函数的定义使用 λ-演算语言[8,9]。我们首先定义适合于 E-RG 规则的语义域，语义域抽象了规则定义的语法项，然后定义支撑函数(或辅助函数)描述语法项之间的相关操作。在定义了语义域和支撑函数的基础上，最后我们给出语义函数的定义。我们描述的规则语义基于关系数据模型，主动规则的执行是面向元组的。

下面的描述中，2^S 表示集合 S 的幂集合；$\langle e_1,e_2\cdots e_n\rangle \downarrow i, i \in [1,n]$ 表示在 n 元组 $\langle e_1,e_2,\cdots,e_n\rangle$ 的第 i 项上的投影操作。

3.1　语义域

·设 Σ 是数据库状态域，如果 σ 是状态域 Σ 中的一个状态，那么 $\sigma=\{t_1,t_2,\cdots,t_n\}$，其中 t_i 是元组。我们假定元组具有唯一的不可重用标识。

·设 Δ 是改变事件集域。如果 δ 是域 Δ 中的某改变事件集，那么 $\delta=[PE,CE]$，其中 $PE=\{pe_1,pe_2,\cdots,pe_n\}$，$CE=\{ce_1,ce_2,\cdots,ce_m\}$，这里 pe_i 是原语事件，ce_j 是复杂事件。

·设 *CAR* 是 *CA* 规则域。如果 *car* 是规则域 *CAR* 中的一个 *CA* 规则，那么 *car* 是一函数，该函数以改变事件集 δ 和数据库状态 σ 作为参数，结果返回一布尔值、新的变化事件集以及新的数据库状态。也就是：

$car:\Delta\times\Sigma\rightarrow\{true,false\}\times\Delta\times\Sigma$

如果规则 car' 的条件为真(car(δ,σ)↓1=true)，那么 car(δ,σ)↓2和 car(δ,σ)↓3就是执行规则 car. 后的新结果；如果规则 car 的条件为假(car(δ,σ)↓1=false)，那么 car(δ,σ)↓2=[ϕ,ϕ]并且 car(δ,σ)↓3=σ。

·设 *P* 是并发关系域。如果 p 是域 *P* 中的一个并发关系，那么 $p=\{car_i\alpha\{car_j,\cdots car_k,\cdots car_n\},\cdots\}$ 其中 car 是 CA 规则，符号 α 表示事件发生的直接先于关系[3]。

·设 Y 是同步关系域。如果 y 是域 Y 中的一同步关系，那么 $y=\{\{car_j,car_k,\cdots,car_n\}\alpha car_i,\cdots\}$ 其中 car 是 CA 规则，符号 α 表示事件发生的直接先于关系。

·设 S 是依次关系域。如果 s 是域 S 中的一依次关系，那么 $s=\{car_i\ \alpha\ car_j,\cdots\}$ 其中 car_i 是 *CA* 规则，符号 α 表示事件发生的直接先于关系。

·设 *O* 是规则关系域(*rule relations*)。即：

$o=p+y+s$

·设 *R* 是 *E-RG* 规则域(*E-RG rules*)。即：

$O=2^{CAR}\times O$

如果 erg_i 是域 R 中的 E-RG 规则，那么 $erg_i=\langle\{car_1,car_2,\cdots,car_m\},o_i\rangle$

·设 *RE* 是规则-事件偶对集域。即：$RE\subset 2^{R\times\Delta}$

如果 *re* 是域 *RE* 中的规则-事件偶对集，那么 $re=\{\langle erg_1,\delta_1\rangle,\langle erg_2,\delta_2\rangle,\cdots,\langle erg_n,\delta_n\rangle\}$。

3.2　辅助函数

规则的指称语义使用语义函数和辅助函数表达，函数的定义使用 λ-演算，该语言基于两种基本的概念：函数抽象和函数施用[8]。下面我们给出函数的 λ-表达式以及函数意义的概要说明。

$Run-erg:R\times\Sigma\times RE\rightarrow\Sigma\times RE$

$Run-erg=\lambda\, erg_i,\sigma,\{\langle erg_1,\delta_1\rangle,\langle erg_2,\delta_2\rangle,\cdots,\langle erg_n,\delta_n\rangle\}.$
　　let $\sigma'=Run-rg(\{car_1,car_2,\cdots,car_n\},o_i)\downarrow 1$ in
　　let $\delta'=Run-rg(\{car_1,car_2,\cdots,car_n\},o_i)\downarrow 2$ in
　　$\langle\sigma',Add\text{-}Change(\delta',\{\langle erg_1,\delta_1\rangle,\cdots,\langle erg_i,[\phi,\phi]\rangle,\cdots,\langle erg_n,\delta_n\rangle\})$

$Add-Change:\Delta\times RE\rightarrow RE$

$Add-Change=\lambda\delta,\{\langle erg_1,\delta_1\rangle,\langle erg_2,\delta_2\rangle,\cdots,\langle erg_n,\delta_n\rangle\}.$
　　$\{\langle erg_1,\delta\rangle,\langle erg_2,\delta\rangle,\cdots,\langle erg_n,\delta\rangle\}$

函数 Run－erg 模拟了 *E-RG* 规则的执行过程，*E-RG* 规则的执行可以接连(*cascade*)触发其它规则的执行。

函数 Add－Change 模拟了在 *E-RG* 规则执行后被触发的规则集的改变。

$Run-rg:2^{CAR}\times O\rightarrow 2\Sigma\times\Delta$

$Run-rg=Least-Fixed-Point(\lambda F.\ \lambda(\{car_2,car_2,\cdots car_m\},o)$
　　if $Select(\{car_1,car_2,\cdots car_m\},o)=\phi$ then $\langle\{\sigma\},\delta\rangle$
　　else let $CAR'=Select(\{car_1,car_2,\cdots car_m\},o)$ in
　　$Y_{car_1\in CAR'}F(car_i(\delta,\sigma))$

$Select:2^{CAR}\times O\rightarrow 2^{CAR}$

$Select=\lambda\{car_1,car_2,\cdots,car_m\},o.$
$\{car_i\mid If\exists\ car_j\ni\langle car_j,car_i\rangle\in o,$ then the rule car,had been completed. $1\leqslant i,j\leqslant m\}$

函数 Run－rg 被定义为泛函 *F* 的最小不动点，该函数模拟了规则图(*rule graph*)的执行。函数 Run－rg 在此显示了 *E-RG* 规则执行的内部并发(*intra-rule parallels*)。

函数 Select 模拟了规则管理器从规则图中选出可执行的 *CA* 规则的情形，这些被选出的 *CA* 规则可以并发地执行。

$Eligible:RE\rightarrow 2^R$

(下转第62页)

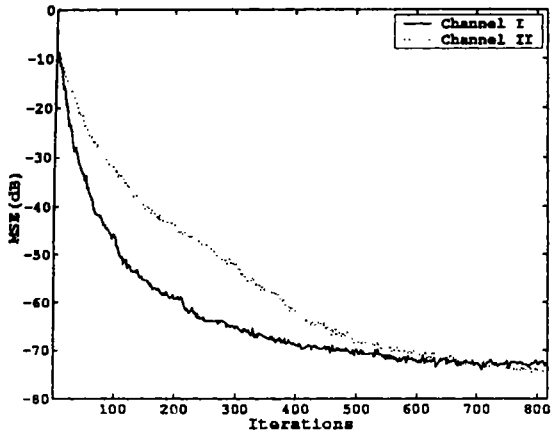

图6 40次独立实现的平均均方误差与迭代次数

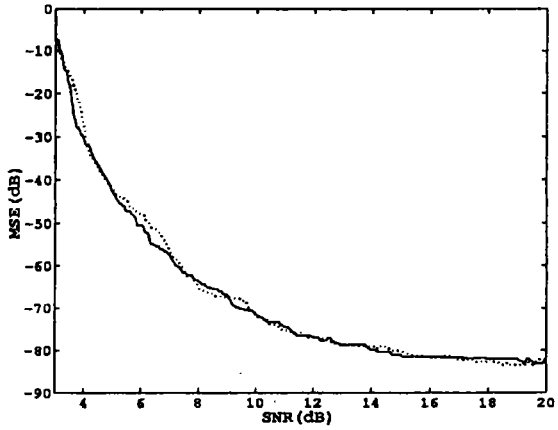

图7 已恢复的 $\hat{x}_2$ 均方误差性能与信噪比，图中实线为信道 I，虚线为信道 II

4 Proakis J G. Digital Communications, 3rd Edition. New York: McGraw-Hill, 1995

5 Takens F. Detecting strange attractors in turbulence in Dynamical Systems and Turbulence. D. Rand and I. Young (ed.), Berlin: Springer-Verlag, 1981. 366～381

6 Gencay R, Liu T. Nonlinear modeling and prediction with feedforward and recurrent networks. Physica D, 1997, 108: 119～134

7 Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1989, 1: 270～280

8 Haykin S. Adaptive Filter Theory, 3rd Edition. NJ: Prentice Hall, 1996

9 Qureshi S U H. Adaptive equalization. Proceedings of IEEE, 1985, 73: 1349～1387

(上接第79页)

Eligible = $\lambda\{\langle erg_i, \delta_1\rangle, \langle erg_2, \delta_2\rangle, \cdots, \langle erg_n, \delta_n\rangle\}$.
{erg_i | all rules are triggered by any event in set of event change Δ. $1 \leqslant i \leqslant n$}

函数 Eligible 模拟了被触发的 *E-RG* 规则的处理方法，相当于 recognize-act cycle 算法中的冲突消解，但是在这里可以获得多个 *E-RG* 规则，而非一个，该函数将应用于下面的语义函数。

3.3 语义函数

$M: 2^R \rightarrow \Delta \times \Sigma \rightarrow 2^\Sigma$

$M[\![\{erg_1, erg_2, \cdots, erg_n\}]\!] = \lambda\delta, \sigma. M'(\langle \sigma, Distrib(\delta, \{erg_1, erg_2, \cdots, erg_n\})\rangle)$

$Distrib: \Delta \times 2^R \rightarrow RE$

$Distrib = \lambda\delta, \{erg_1, erg_2, \cdots, erg_n\}. \{\langle erg_1, \delta\rangle, \langle erg_2, \delta\rangle, \cdots, \langle erg_n, \delta\rangle\}$

$M': = \Sigma \times RE \rightarrow 2^\Sigma$

$M' = Least-Fixed-Point(\lambda F.$
$\lambda\langle \sigma, \{\langle erg_1, \delta_1\rangle, \langle erg_1, \delta_2\rangle, \cdots, \langle erg_n, \delta_n\rangle\}\rangle.$
$if\ Eligible(\{\langle erg_1, \delta_1, \langle erg_1, \delta_2\rangle, \cdots, \langle erg_n, \delta_n\rangle\}) = \phi then\{\sigma\}$
$else\ let\ R' = Eligible(\{\langle erg_1, \delta_1\rangle, \langle erg_2, \delta_2\rangle, \cdots, \langle erg_n, \delta_n\rangle\})\ in$
$Y_{erg_1 \in R'} F(Run-erg(erg_1, \sigma, \{\langle erg_1, \delta_1\rangle, \langle erg_1, \delta_2\rangle, \cdots, \langle erg_n, \delta_n\rangle\})))$

函数 M 模拟了主动规则的处理语义。函数 M' 被定义为泛函 F 的最小不动点，该函数显示了 E-RG 规则执行的规则间的并发性。

总结 本文着重研究主动规则模型 E-RG 的指称语义，作为正确实现和理解规则语言的基础，基于上述研究结果，我们实现的 E-RG 主动规则系统，集成于主动实时数据库原型系统 ARTs-Ⅱ项目，该项目得到国防预研基金以及国家自然科学基金资助。规则执行的两种并发性分别应用操作系统的进程和线程模型实现，限于本文的篇幅，规则系统的实现细节我们将在其他文章中详细介绍。

参 考 文 献

1 Brownston L, Farrel R, Kant E, et al. Programming Expert Systems in OPS5, Addison-Wesley, Reading, Massachusetts, 1985.

2 Liu yunsheng, Xu changxing, Xu guiping, An E-RG Rule Model in Active Database Systems. In: 2000 Intl. Conf. on Intelligent Information Processing, the 16th IFIP World Computer Congress, Beijing, China, 2000, Aug. 21～25

3 Xu ChangXing, Liu YunSheng, Xu GuiPing. A Graph-based Rule Model in Active Database Systems. Journal of HuaZhong University of Science and Technology, 2000, 28(11), 31～33

4 Saygin, Y. Ulusoy, O. Chakravarthy, S. , Concurrent Rule Execution in Active Database, Information Systems, 1998, 23 (1), 39-64

5 Paton N, et al. Formal Specification of Active Database Functionality: A Survey. In: Sellis T, ed. Proc. Second Int'l Workshop Rules in Database Systems, RIDS, 1995. 21～35

6 Bertino E, Guerrini G, Merlo I, Trigger Inheritance and Overriding in an Active Object Database system, IEEE Transaction on Knowledge and Data Engineering, 2000, 12 (4): 588～608

7 Chakravarthy S, et al. HiPAC: A research project in active time-constrained database management final technical report: [Technical Report XAIT-89-2]. Reference Number 187, Xerox Advanced Information Technology, 1989

8 Watt D A. Programming Language Syntax and Semantics, Prentice-Hall, New York, 1991

9 Widom J. A Denotational Semantics for the Starburst Production Rule Language, SIGMOD Record, 1992, 21(3): 339～351