

ECA 规则的模型和行为特定理论

姜跃平 汪卫 施伯乐

董继润

(复旦大学计算机系 上海 200433)

(山东大学计算机系 济南 250100)

摘要 本文给出了ECA规则系统的一般性模型,通过扩展系统状态和改变的定义,形式化地描述了规则处理的语义,并着重研究规则系统的行为特征(可终止性和行为一致性),文中给出的形式化定义和判定方法可作为规则静态分析工具的理论基础。

关键词 数据库,主动,规则,模型,语义,行为。

ECA(事件-条件-动作)规则^[1-3]具有很强的语义表达能力,可以支持完整性保持^[1,5,6]、派生数据维护^[7]、生产监控、市场监控和决策支持系统^[2]等各种应用,作为主动数据库系统的主动服务机制已得到广泛的接受。

简单地说,ECA规则的职能是:当规则事件发生时,系统实时地或在规定时刻检查规则的条件,如满足则执行规则的动作。ECA规则与CA规则^[4]和产生式规则^[5,7]的主要区别是:它的事件作为一个相对独立的成分,具有专门的监测机制,有更强的能力描述数据库内外的各种状态变化。但是,由于规则的行为十分复杂,规则设计人员很难预计他们设计的规则会给数据库状态带来什么样的影响,缺乏对规则设计和分析的方法论支持已成为主动数据库的发展瓶颈。^[2]

规则行为的复杂性尤其体现在以下2个方面:

- 1)终止:规则间的循环触发,可能造成对某些规则的处理无法终止。
- 2)行为一致:如多条规则被同时触发,不同的处理顺序可能导致不同的终止状态。

我们统称这些问题为行为特定问题。由于规则的条件求值和动作执行结果依赖于它被处理时的数据库状态,我们无法仅仅根据规则的定义,精确地判定对一组规则的处理必定是不可终止的或行为不一致的。但是对于一个规则集合 R ,我们找到了正确的方法来判定在什么情况下对 R 的处理能够保证终止或行为一致。以此为基础,可以为ECA规则系统构造关于规则行为的静态分析工具,帮助设计人员改善他所设计的规则。

与其它ECA规则模型^[1,3,4]不同的是,我们扩展了数据库系统状态(State)和改变(Transition)的定义,形式化地描述了规则处理的语义。已有的对规则行为特定理论的研究

× 本文研究得到国家自然科学基金资助。作者姜跃平,1970年生,博士生,主要研究领域为主动数据库,面向对象数据库,汪卫,1970年生,博士生,主要研究领域为面向对象数据库,施伯乐,1935年生,教授,博士生导师,主要研究领域为数据库理论,知识库,面向对象数据库,董继润,1935年生,教授,主要研究领域为数据库理论,软件工程。

本文通讯联系人:姜跃平,上海200433,复旦大学计算机系

本文1996-03-18收到修改稿

是面向 Datalog, CA 等传统规则的^[8,9], 其唯一不动点理论仅考虑一组固定的规则, 而 ECA 规则系统的规则处理要且只要涉及所有相继被触发的规则; 传统的可终止理论必须以改变序列的单调性为前提, 不能适应数据操作和 ECA 规则行为的一般情形. 而我们的研究面向 ECA 规则, 不论是问题描述, 还是判定方法都能适应 ECA 规则的一般模型.

本文第 1 节扼要介绍规则模型, 形式化地描述了规则语义; 第 2 节和第 3 节分别给出了可终止性和行为一致性的问题描述和判定方法; 最后是结语和展望.

1 规则模型

作为主动数据库系统的组成部分, 一个典型的 ECA 规则系统(ERS)可以表示为三元组 (R, m_e, m_c) , 其中 R 是规则库, m_e 是事件监测机制, m_c 是条件计算和动作执行机制, 而一条 ECA 规则可表示为

$$E \circ C \xrightarrow{P} A$$

其中 E, C, A 分别代表规则的事件、条件和动作; P 代表描述规则行为或状态的附加性质, 如 E - C 耦合方式和 C - A 耦合方式等.

1.1 事件、条件、动作

只有规则的事件 E 发生, 条件 C 才可能得到测试, 这时称规则被触发了. ERS 不仅支持原子事件, 而且支持复合事件. 原子事件包括数据库改变、事务边界、时序(Temporal)事件和外部事件(如硬件中断)等; 复合事件由原子事件或其它事件合成.^[1-3] 由于有相对独立的监测机制并引进了时序事件、复合事件等新成分, ERS 能够对更多的数据库内外的状态改变作出反应, 不再局限于传统的数据更新类操作.^[5-7,8]

规则的条件可以等价地视为数据库查询的条件部分. 在关系数据库中, 一个查询的条件总能用一个元组关系演算公式(TRCF)表示^[9], 因而规则条件也总可以表示为一个 TRCF. 通过应用德·摩根定律, 我们总可以把一个 TRCF 中的否定操作符移到最里层, 使之只出现在关系谓词前面. 下文中, “TRCF”均指经过这种改写后的 TRCF.

为了后面的讨论, 我们引入一些记号: 如果在 $TRCF\alpha$ 中, 关系 t 对应的谓词存在一个前面有(没有)否定操作符的出现, 则称 α 含有负 t (正 t), 记为 “ $-t \in \alpha$ ”(“ $t \in \alpha$ ”); 如果关系 t 的第 i 个属性 $t.c$ 所对应的变量 $t[i]$ 出现在 $TRCF\alpha$ 中, 则称 α 含有 $t.c$, 记为 “ $t.c \in \alpha$ ”. 此外, 我们用 $TRCF(r)$ 来表示规则 r 的条件所对应的 TRCF.

动作由一系列数据库操作或非数据行为(如警告信息显示)组成. 对规则条件进行计算, 如果为真执行其动作——这一过程称为对规则的“处理”或“应用”. 如果动作的执行改变了系统状态(见 1.2 节), 则可能导致某些事件的发生, 从而触发某些规则, 这时对该规则的处理可能要引发一个包含其它新触发规则的规则应用序列; 反之, 则不会触发其它规则, 对该规则的处理即告结束.

通常在动作里也有选择运算, 或者作为独立的选择操作, 或者作为数据改变操作的条件. 这样的选择同样可以用 TRCF 来表示, 我们把这些 TRCF 连同 $TRCF(r)$ 合记为 $TRCF^*(r)$.

规则 r 的动作里的数据改变操作可以表示为集合 $O(r)$:

$$O(r) = \{\langle I, t \rangle\} \cup \{\langle D, t \rangle\} \cup \{\langle U, t, c \rangle\}$$

其中 $\langle I, t \rangle$ 表示向关系 t 中插入元组, $\langle D, t \rangle$ 表示从关系 t 中删除元组, $\langle U, t, c \rangle$ 表示更新关系 t 的元组在属性 c 上的值.

为了后面的讨论, 我们给出以下定义.

定义 1.1. 源关系集 SR 和源属性集 SC 分别是规则的条件和动作要读用的关系集合和属性集合. 给定规则 r ,

$$\begin{aligned} SR+(r) &= \{t \mid t \in TRCF^*(r)\}; & SR-(r) &= \{t \mid -t \in TRCF^*(r)\}; \\ SR(r) &= SR+(r) \cup SR-(r); & SC(r) &= \{t, c \mid t, c \in TRCF^*(r)\}. \end{aligned}$$

定义 1.2. 目标关系集 TR 是规则动作要作元组增删的关系集合; 目标属性集 TC 是规则动作要更新(Update)的属性集合. 给定规则 r ,

$$\begin{aligned} TR+(r) &= \{t \mid \langle I, t \rangle \in O(r)\}; & TR-(r) &= \{t \mid \langle D, t \rangle \in O(r)\}; \\ TR(r) &= TR+(r) \cup TR-(r); & TC(r) &= \{t, c \mid \langle U, t, c \rangle \in O(r)\}. \end{aligned}$$

1.2 形式语义

为了形式化地描述规则处理的语义, 我们先扩展 2 个传统概念: 状态(State)和改变(Transition).

定义 1.3. 一个基于 ERS 的主动数据库系统的状态(以下简称状态) s 是一个二元组 $\langle s_D, s_R \rangle$, 其中 s_D 是数据库状态, s_R 是规则库状态. 为统一表示 s_D 和 s_R , 假设有一个特殊的单维关系 T_R , 其元组为当前所有正等待处理的规则. 如果数据库 $D = \{T_1, T_2, \dots, T_n\}$, 则 s, s_D, s_R 都可以用向量表示:

$$s_D = (t_1, t_2, \dots, t_n), \quad s_R = (t_R), \quad s = (t_1, t_2, \dots, t_n, t_R)$$

其中 t_i 和 t_R 分别是 T_i 和 T_R 的取值($1 \leq i \leq n$). T_R 的特殊性在于: 如果 T_R 中的某规则 r 又被触发, 则 T_R 中将有 2 个元组对应于规则 r . 我们称 T_R 中的 1 个元组为 1 个被触发规则实例.

定义 1.4. 改变描述状态的变化. 一个基于 ERS 的主动数据库系统的改变(以下简称改变) u 是一个二元组 $\langle u_D, u_R \rangle$, 其中 u_D 是数据库改变, u_R 是规则库改变. 因为对关系的更新(Update)总可以用删除和插入等价表示, 所以 u, u_D, u_R 都可以表示为一个矩阵:

$$u_D = \begin{pmatrix} I_1 & I_2 & \dots & I_n \\ D_1 & D_2 & \dots & D_n \end{pmatrix}, \quad u_R = \begin{pmatrix} I_R \\ D_R \end{pmatrix}, \quad u = \begin{pmatrix} I_1 & I_2 & \dots & I_n & I_R \\ D_1 & D_2 & \dots & D_n & D_R \end{pmatrix}$$

其中 I_k 表示插入 T_k 的元组, D_k 表示从 T_k 删除的元组($1 \leq k \leq n$); I_R 表示新的被触发规则实例, D_R 表示新近得到处理的被触发规则实例. 此外, u_D 描述数据库操作的净效应, 即若元组 α 先被插入 T_i , 继而删除, 则在 u_D 中, $\alpha \in D_i$ 且 $\alpha \notin T_i$; 反之同理. 因而有 $I_i \cap D_i = \emptyset$, $1 \leq i \leq n$, 及 $I_R \cap D_R = \emptyset$.

如果对状态 s 施以改变 u , 我们将得到新状态 s' :

$$s' = u(s) = (t_1 \cup I_1 - D_1, t_2 \cup I_2 - D_2, \dots, t_n \cup I_n - D_n, t_R \cup I_R - D_R).$$

另外, 如果对状态 s_0 施加一系列改变 $\{u_1, \dots, u_k\}$, 最终得到状态 s_k , 这一过程可表示为:

$$s_0 \xrightarrow{u_1} s_1 \xrightarrow{u_2} \dots \xrightarrow{u_{k-1}} s_{k-1} \xrightarrow{u_k} s_k = u'_k(s_0).$$

需要注意的是, 数据库改变之间可能存在冲突, 即可能存在 u_{i_1} 和 u_{i_2} , 对某关系 T_k , 有 I_{k_1}

$\cap D_k \neq \emptyset$. 这种冲突使得改变序列的数据库部分并不是单调递增的. 由于文献[8]的理论不容许这种情况, 所以我们的讨论更具有—般性.

现在我们借助状态和改变来描述规则处理的语义. 设 s 为规则 r 被处理前的状态, 应用 r 之后, 我们将得到新状态 s' , 这一处理步可表示为 $s \rightarrow s'$ 或 $u \rightarrow u'$, 其中 u, u' 分别是 s 和 s' 相对于某个初始状态 s_0 的改变. 如果 r 的动作并没被执行, 或并没造成数据库状态的改变, 则有 $u_d' = u_d, T_R' = T_R - \{r\}$. 由初始状态 s_0 开始, 连续应用 k 条规则 $\{r_1, \dots, r_k\}$, 可以表示为

$$u_0 \xrightarrow{r_1} u_1 \rightarrow \dots \xrightarrow{r_{k-1}} u_{k-1} \xrightarrow{r_k} u_k,$$

其中 u_0 的每个关系都是 \emptyset , 最终状态是 $u_k(s_0)$. 需要注意的是, 设 R_0 中有 k 个被触发规则实例, 则由 $s_0 = \langle D_0, R_0 \rangle$ 开始的整个规则处理序列所包含的处理步可能远远不止 k 个, 甚至会因为规则间的循环触发而不能终止.

2 可终止性

由于数据库状态的改变或其它事件的作用, ERS 的某些规则被触发, 由此开始了一个规则处理过程: 规则动作的执行和后继事件可能触发新的规则, ERS 要依次处理所有被触发实例. 如果到某一时刻, 没有在等待处理的被触发规则, 则称该处理过程已终止 (Terminate).

定义 2.1. 给定 $ERS = \langle R, m_r, m_t \rangle$, 如果从任一状态出发的规则处理过程都肯定能终止, 则称规则库 R 是可终止的 (Terminable).

我们借助触发依赖图来判定规则处理的—可终止性.

2.1 触发依赖图 (FDG)

定义 2.2. 任意规则集 R 的触发依赖图 G_R 定义如下:

- (1) 任意规则 $r \in R$, 在 G_R 中有顶点 v_r 与之—对应.
- (2) 任意 $r_1 \in R$ 和 $r_2 \in R$, 若 r_1 的动作的执行可能导致 r_2 的事件发生 (即触发 r_2), 则在 G_R 中有从 v_{r_1} 到 v_{r_2} 的弧.
- (3) 除此之外, G_R 中没有其它的顶点和弧.

根据定义, 触发依赖图 G_R 有以下性质:

- (1) G_R 为有向图, 不一定连通, 且可能存在自环 (Self-loop);
- (2) 若规则 r 的动作中不含有改变数据库状态的操作, 则 G_R 无自 V_r 出发的弧;
- (3) 若规则 r 的事件中不含有数据状态改变类的原子事件, 则 G_R 无到 V_r 的弧.

在 CRS 中^[8], 规则 r_1 能否触发 r_2 完全取决于 r_1 的动作和 r_2 的条件; 而在 ERS 中, 则取决于 r_1 的动作和 r_2 的事件.

2.2 可终止规则库的判定

基于 ERS 的主动数据库需要有正确的理论方法以保证用户设计的规则库是可终止的. 为描述我们的判定法则, 先引入 2 个定义.

定义 2.3. 给定规则集合 R , 若对任意 $r_1 \in R$ 和 $r_2 \in R (r_1 \neq r_2)$, 有 $TR+(r_1) \cap TR-(r_2) = \emptyset$, 且 $TC(r_1) \cap TC(r_2) = \emptyset$, 则称 R 是无动作冲突的.

观察 2.1. 任意规则应用序列 $u_0 \xrightarrow{r_1} u_1 \rightarrow \dots \xrightarrow{r_{k-1}} u_{k-1} \xrightarrow{r_k} u_k$, 若其中的任意 $r_i \in R (1 \leq i$

$\leq k$),且 R 是无动作冲突的,则相应的数据库改变序列(U_d ,见 1.2 节)是单调递增的.

定义 2.4. 给定规则集合 R ,对任意 $r \in R$,任意 $t, c \in TC(r)$,若所有 $(I, t) \in O(r)$ 和 $(U, t, c) \in O(r)$ 对 t, c 的赋值都是简单赋值(即赋值表达式中不含有函数和运算符),则称 R 是赋值简单的.

观察 2.2. 若 R 是赋值简单的,则 R 中规则动作的执行不会产生当前状态的数据库中所没有的新文字(Literal).

定理 2.1. 规则库 R 是可终止的,如果其触发依赖图 G_R 是(1)无环的,或者(2)任意环 C 的顶点集合 V_C 所对应的规则集合 R_C 是无动作冲突的和赋值简单的.

证明:(1) G_R 无环.设 R 有 n 条规则,注意到 G_R 可能存在的最长路径的长度 $\leq n-1$,容易证明从任意状态开始的规则处理序列经有限个规则应用步后必然终止.

(2) G_R 有环.考察其任意环 C ,因为起始状态的数据库里的文字是有限的,根据观察 2.1 和观察 2.2, R_C 中的任意规则 r 在被重复触发应用有限遍后,必然导致 r 的动作的执行再不能改变数据库状态,从而再不能触发新的规则.因 G_R 的环的顶点数 $\leq n$,设 G_R 有 m 个环,设环上任意规则可被触发应用的最大遍数为 K_{max} ,则 G_R 中对应于触发应用序列的最长路径的长度 $\leq (n-1) + m \cdot n \cdot K_{max}$.以后的证明与情况(1)一致,限于篇幅从略. \square

注意到 FDG、无动作冲突规则集和赋值简单规则集都可以通过对规则定义的静态分析得到,说明可以构造基于定理 2.1 的可终止性静态分析工具,对于不满足定理条件的规则库,分析工具可以把在 FDG 上发现的所有环提供给用户,帮助用户验证是否有某种具体因素使环能够避免陷入无限循环.

3 行为一致性定义

定义 3.1. 给定规则库 R ,如果对于从任意状态开始的规则处理序列,无论 ERS 以何种(合法的)顺序处理被触发实例,序列总能终止在同一个状态,则称 R 是行为一致的(Behaviour-consistent)或具备行为一致性.

定义 3.2. 任意规则 r_1 和 r_2 ,如果 $r_1=r_2$ 或满足如下条件,则称 r_1 和 r_2 是可交换的:

- (1) $TR(r_1) \cap SR(r_2) = \emptyset \wedge TC(r_1) \cap SC(r_2) = \emptyset$,并且
- (2) $TR+(r_1) \cap TR-(r_2) = \emptyset \wedge TC(r_1) \cap TC(r_2) = \emptyset$,并且
- (3) 条件(1)和(2)对 $i=1, j=2$ 和 $i=2, j=1$ 都成立.

引理 3.1. 如果规则 r_1 和 r_2 是可交换的,则互换 r_1 和 r_2 的处理顺序不会影响所得到的

状态.即若有 $u_0 \xrightarrow{r_1} u_1 \xrightarrow{r_2} u_2$ 和 $u_0 \xrightarrow{r_2} u_1' \xrightarrow{r_1} u_2'$,则 $u_2 = u_2'$.

证明:(1) $r_1=r_2$,证明是平凡的.

(2) $r_1 \neq r_2$.根据定义 1.1、定义 1.2 和定义 3.2 的条件(1)可知, r_1 的动作对 r_2 的条件和动作所要读用的数据不做任何改变,因而 r_1 的动作不会影响 r_2 的条件值和 r_2 的动作对状态的改变.反之也一样.亦即

$$f(u_1, r_2) = f(u_0, r_2), f(u_1', r_1) = f(u_0, r_1),$$

其中 $f(u_i, r_j)$ 表示从 u_i 出发应用 r_j 所产生的状态改变,这一改变与 u_i 的复合产生 u_{i+1} .又由定义 3.2 的条件(2)可知, r_1 和 r_2 对状态的改变是无冲突的.亦即如果用“ \circ ”表示 2 个改

变之间的复合运算, 则 $f(u_0, r_2) \circ f(u_1', r_1)$ 等价于它们在每个关系 (见 1.2 节) 上的并 (Union). 所以有 $f(u_0, r_2) \circ f(u_1', r_1) = f(u_1', r_1) \circ f(u_0, r_2)$, 因而, $u_2 = u_0 \circ f(u_0, r_1) \circ f(u_1, r_2) = u_0 \circ f(u_1', r_1) \circ f(u_0, r_2) = f(u_0, r_2) \circ f(u_1', r_1) = u_2'$. \square

定理 3.1. 给定规则库 R , 若 R 是可终止的, 并且任意 $r_1 \in R$ 和 $r_2 \in R$, r_1 和 r_2 是可交换的, 则 R 是行为一致的.

证明: 设初始状态为 s_0 , 其被触发规则实例集 $t_{R_0} = \{r_1, r_2, \dots, r_k\}$, 对任意 2 个规则处理序列 $\alpha: u_0 \xrightarrow{r_a} u_{11} \xrightarrow{r_c} \dots \xrightarrow{r_1} u_{1m}$ 和 $\beta: u_0 \xrightarrow{r_b} u_{21} \xrightarrow{r_d} \dots \xrightarrow{r_k} u_{2n}$, α 和 β 中必都有 t_{R_0} 中的 $r_i (1 \leq i \leq k)$ 的一次出现, 且 $u_{1m}(s_0)$ 和 $u_{2n}(s_0)$ 分别是 α 和 β 的终止状态. 由于任意 2 条规则是可交换的, 经过有限步的顺序交换后, 必能把 t_{R_0} 中的 $r_i (1 \leq i \leq k)$ 调整到序列的第 i 个处理步上, 此时的序列为

$$\begin{aligned} \alpha: u_0 &\xrightarrow{r_1} u'_{11} \xrightarrow{r_2} \dots \xrightarrow{r_k} u'_{1k} \xrightarrow{\dots} u_{1m} \text{ 和} \\ \beta: u_0 &\xrightarrow{r_1} u'_{21} \xrightarrow{r_2} u'_{2k} \xrightarrow{\dots} u_{2n}, \end{aligned}$$

其中 $u'_{1i} = u'_{2i} (1 \leq i \leq k)$. 特别地, 如果记 $u'_{1k} = u'_{2k} = u_{s1}$, 则 α 和 β 自 u_{s1} 以后的子序列可记为 $\alpha_1: u_{s1} \xrightarrow{r_1} \dots \xrightarrow{r_m} u_{1m}$ 和 $\beta_1: u_{s1} \xrightarrow{r_1} \dots \xrightarrow{r_n} u_{2n}$, 其初始被触发规则实例集 $t_{s1} = t_{R_0} \cup I_{s1} - D_{s1}$ (见 1.2 节). 不妨称 u_{s1} 为 α 和 β 的新出发点. 对 α_1 和 β_1 进行类似的有限次变换后又得到第 2 个新出发点 u_{s2} . 设 $m \leq n$, 新出发点的个数必然是有限的, 必存在 $t < m$, $\alpha_t: u_{s1} \xrightarrow{r_1} \dots \xrightarrow{r_t} u_{1m}$; 而此时, $\beta_t: u_{s1} \xrightarrow{r_1} \dots \xrightarrow{r_t} u_{1m}$ 也使 β 序列终止 (见第 2 节), 故 $u_{1m} = u_{2n}$. \square

由于可交换规则也可以通过对规则的静态分析得到, 因而可以构造基于定理 3.1 的行为一致性分析工具, 自动检查规则库的行为一致性, 把造成潜在的行为不一致的规则找出来, 帮助规则设计人员对规则库加以改进.

4 结语及展望

本文给出了一个 ECA 规则系统的一般性模型, 它支持复合事件和关系演算条件, 并基于扩展的状态和改变, 形式化地描述了规则处理的语义. 本文的讨论重点是规则行为特定问题, 即规则处理是否能保证终止及是否能保证终止在一个唯一状态. 我们给出了问题的描述和相关的判定方法.

在实际应用中, 尤其是在规则库较大的时候, 定理 2.1 和定理 3.1 的条件可能较难满足. 因而对不能保证终止和行为一致的规则库, ERS 需要一种更加宽松的判定原则, 使之能确定一个规则处理序列所可能涉及到的规则集合 (而不是整个规则库) 是否能保证终止和行为一致. 关于这个问题的进一步研究将反映在我们以后的文章里. 虽然我们的模型和理论是基于关系数据模型的, 但其中的主要思想可适应面向对象数据模型. 关于面向对象数据模型的方法、封装性给 ECA 规则的行为带来的具体影响, 也是我们下一步研究的一项重点内容.

参考文献

1. Ishikawa H, Kubota K. An active object-oriented database: a multi-paradigm approach to constraint management.

- Proc. of 19th VLDB, 1993. 467~478.
- 2 Simon E, Dittrich A K. Promises and realities of active database systems. Proc. of 21st VLDB, 1995. 642~653.
 - 3 Chakravarthy S *et al.* Composite events for active databases. Proc. of 20th VLDB, 1994. 606~617.
 - 4 Hanson E N. Rule condition testing and action execution in ariel. Proc. of ACM SIGMOD, 1992. 49~58.
 - 5 Stonebraker M. The integration of rule systems and database systems. IEEE TKDE, 1992, 4(5):415~423.
 - 6 姜跃平, 董继润. 完整性约束规则的自动生成. 计算机科学, 1994, 4:52~56.
 - 7 Widom J *et al.* Implementing set-oriented production rules as an extension to starburst. Proc. of 17th VLDB, 1991. 275~285.
 - 8 Zhou Yuli, Hsu Meichun. A theory for rule triggering systems. EDBT90 Lecture Notes in Computer Science 416, 1990. 409~421.
 - 9 Ullman J D. Principles of database and knowledge-base systems. USA, Computer Science Press, INC., 1988.

MODEL AND BEHAVIORAL DETERMINISM THEORY FOR ECA RULES

JIANG Yueping WANG Wei SHI Baile

(Department of Computer Science Fudan University Shanghai 200433)

DONG Jirun

(Department of Computer Science Shandong University Ji'nan 250100)

Abstract This paper introduces a generic model for ERS (ECA rule systems). By extending the definition of system state and transition, it formally describes the semantics of rule processing. It furtherly investigates the behavioral determinism of a given ECA rulebase, including termination and behavioral consistency. The relative definitions and decision criteria which are given in this paper can form the theoretical base of static analysis tools for an ERS.

Key words Database, active, rule, model, semantics, behavior.