

基于图转换的模型重构冲突检测方法

陈军冰^{1,2}, 王志坚¹

(1. 河海大学计算机及信息工程学院, 南京 210098; 2. 河海大学科学研究院, 南京 210098)

摘 要: 根据冲突分类规则, 对非删除规则产生的关键对的计算进行优化, 以避免计算两个非删除规则的交叠。探讨了对于一种给定带否定应用条件(NACs)的图转换系统(GTS)可直接计算出关键对的方法, 并给出直接构建带NACs的关键对的方式。针对每个冲突原因, 存在一个有效关键对, 可用最小上下文来表示冲突。有效关键对集合仅为关键对集合的子集。该方法因而可简化系统中冲突的汇合性分析, 从而可简化静态冲突检测方法。

关键词: 模型重构; 图转换; 关键对; 冲突检测

Model Reconstruction Refactoring Conflict Detection Methods Based on Graph Transformation

CHEN Jun-bing^{1,2}, WANG Zhi-jian¹

(1. Computer & Information Engineering College of Hohai University, Nanjing 210098, China;

2. Research Academy of Hohai University, Nanjing 210098, China)

【Abstract】 According to the conflict classification, we can optimize the computation of the non-deletion rules. It avoids calculating the overlap of the two non-deleting rules. Further This paper discusses to the Graph Transformation System(GTS) with NACs, may calculate critical pairs, we it provides a construction for critical pairs with NACs. There It exists an essential critical pairs for every conflict reason, which can use the available smallest context expresses the conflict. Since the essential critical pairs are a subset of the critical pairs, the essential critical pairs can simplify the static conflict detection method.

【Key words】 model reconstruction refactoring; graph transformation; critical pair; conflict detection

1 前言概述

由于模型重构过程中重构应用规则的并行应用会导致重构应用规则之间产生冲突现象^[1]。关键对^[2]是一对有冲突的转换, 模型转换规则之间的冲突关系可以通过图转换中的关键对进行分析。有三种图转换规则应用产生冲突^[2], 前两两个涉及图结构, 最后一个涉及图属性。Tom Mens^[1]文献^[1]研究了使用关键对分析(Critical Pair Analysis)检测结构重构的冲突。该算法需要对所有规则对进行遍历, 算法效率不高。文献^[3]Leen Lambers等^[3]通过计算非删除规则(non-deleting)应用时产生的关键对, 给出了有效的计算冲突关键对方法。

本文运用图转换方法就删除-使用冲突的算法优化, 以及NAC否定应用条件(NAC)约束下冲突检测方法、有效关键对方法进行冲突检测方法分析。

2 基本冲突检测方法与算法

定义 21.1: 关键对(critical pair):

一对关键对是一对冲突的有向图转换 $K \xRightarrow{p1,m1} P_1$ 和 $K \xRightarrow{p2,m2} P_2$, 且满足 m_1 和 m_2 是联合满射。

定义 2.2 胶合条件(gluings condition):

给定一个图生产 $p=(L \xleftarrow{l} K \xrightarrow{r} R)$, 一个图 G , 以及一个带有 $X=(V_X, E_X, s_X, t_X)$ 对所有的 $X \in \{L, K, R, G\}$ 的匹配 $m: L \rightarrow G$, 进行下面的如下定义:

(1)• 胶合点 GP 是在 L 中未被 p 删除的节点和边, 即: $GP=l_V(V_K) \cup l_E(E_K)=l(K)$ 。

(2)• 标识点 IP 是那些在 L 中被 m 标识的节点和边, 即 $IP=\{v \in V_L \mid \exists w \in V_L, w \neq v: m_V(v)=m_V(w)\} \cup \{e \in E_L \mid \exists f \in E_L, f \neq e: m_E(e)=m_E(f)\}$ 。

(3)• 悬摆点 DP 是在 L 中的那些节点, 其在 m 下的像是在图 G 中不属于 $m(L)$ 的一条边的源或目标的节点, 即 $DP=\{v \in V_L \mid \exists e \in E_G \setminus m_E(E_L): s_G(e)=m_V(v) \text{ 或 } t_G(e)=m_V(v)\}$ 。

如果所有的标识点和所有悬摆点也都是胶合点, 那么 p 和 m 满足胶合条件, 即 $IP \cup DP \subseteq GP$ 。

定义 2.3: 否定应用条件(NAC)

NAC 如图 1 所示。

设 M 为所有单射的图映射的集合。

一个在 L 上的否定应用条件或 $NAC(n)$ 是一个图映射 $n: L \rightarrow N$ 。一个图映射 $g: L \rightarrow G$ 满足于 L 上的 $NAC(n)$, 即 $g \sqsubseteq NAC(n)$ 当且仅当不存在 $q: N \rightarrow G$, 导致 $q \circ n = g$ 。对于一个规则 $p: L \xleftarrow{l} K \xrightarrow{r} R$, 一个 L (相对于 R) 上的 $NAC(n)$ 叫做关于 p 的左 NAC (相对于右)。 $NAC_{p,L}(NAC_{p,R})$ 是一个关于 p 的 NAC_s 的左(右)集合。 $NAC_p=(NAC_{p,L}, NAC_{p,R})$ 包括在 p 上的左 NAC 和右 NAC 集合, 称之为关于 p 的 NAC_s 的集合。

基金项目: 国家科技支撑计划基金资助项目(2008BA29B03); 河海大学自然科学基金资助项目(理科类)(2008432311); 国家科技支撑计划 2008BA29B03

作者简介: 陈军冰(1969-), 男, 副研究员、在职博士研究生、副研究员, 主研方向: 软件重构, 软件工程、软件重构等方面的研究; 王志坚, 博士、教授、博士生导师

收稿日期: 20082010-02-08 **E-mail:** jichen88@sina.com

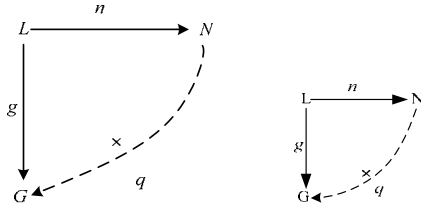


图1 NAC

设 M 为所有单射的图映射的集合。一个在 L 上的否定应用条件 $NAC(n)$ 是一个图映射 $n: L \rightarrow N$ 。一个图映射 $g: L \rightarrow G$ 满足于 L 上的 $NAC(n)$ ，即 $g \models NAC(n)$ 当且仅当不存在 $q: N \rightarrow G \in M$ ，导致 $q \circ n = g$ 。

对于一个规则 $p: L \xleftarrow{l} K \xrightarrow{r} R$ ，一个 L (相对于 R) 上的 $NAC(n)$ 叫做关于 p 的左 NAC (相对于右)。 $NAC_{p,L}(NAC_{p,R})$ 是一个关于 p 的 $NACs$ 的左 (右) 集合。 $NAC_p = (NAC_{p,L}, NAC_{p,R})$ 包括在 p 上的左 NAC 和右 NAC 集合，称为关于 p 的 NAC_s 的集合。

定义 2.4：冲突

两 2 个图转换 $G \xRightarrow{p_1, m_1} H_1$ 带 NAC_{p_1} 和 $G \xRightarrow{p_2, m_2} H_2$ 带 NAC_{p_2} ，如果不是并行独立的，那么它们就是冲突的。也即，如果不存在 $h_{12}: L_1 \rightarrow D_1$ 满足 $(d_1 \circ h_{12} = m_1$ 并且 $e_2 \circ h_{12} \in NAC_{p_1})$ ，或不存在 $h_{21}: L_2 \rightarrow D_1$ 满足 $(d_1 \circ h_{21} = m_2$ 并且 $e_1 \circ h_{21} \in NAC_{p_2})$ 。关键对冲突如图 2 所示。

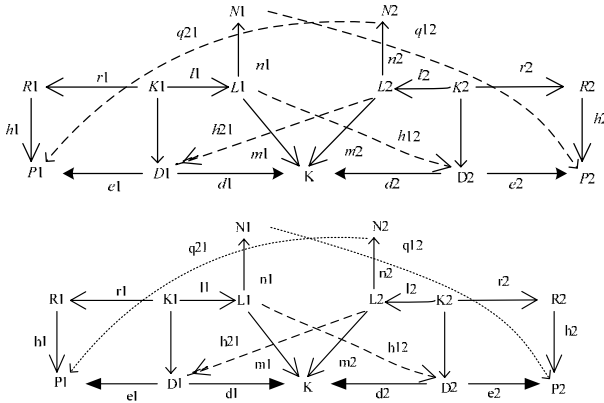


图2 关键对冲突

冲突检测基本算法：

通过关键对的定义，直接计算两 2 个关键对规则集合 $p_1: L_1 \leftarrow K_1 \rightarrow R_1$ 和 $p_2: L_2 \leftarrow K_2 \rightarrow R_2$ 是否冲突，算法步骤如下：

- (1) 穷举所有的联合满射 $m_1 \wedge m_2$ (保证关键对定义中的最小)；
- (2) 分别判断 $m_1 \wedge m_2$ 是否满足胶合条件 (以保证 $p_1 \wedge p_2$ 可以正常执行)；
- (3) 根据三 3 种类型冲突条件检测是否冲突，若冲突则归入冲突集合中。

这一算法对所有规则进行遍历，计算规则之间的关键对，算法效率不高。

3 非删除规则的冲突检测方法的优化

在冲突检测基本算法中，根据规则对应的图结构特征，可以进行改进。

3.1 第一第 1 步优化

首先需要检测规则是删除的还是非删除的。用这个方法可以避免计算两 2 个非删除规则的交叠 (见引理 3.1)。

引理 3.1 已知两 2 个非删除规则 $p_1: L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1$ 和 $p_2: L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2$ ，每一对有向图转换 $H_1 \xRightarrow{p_1, m_1} G \xRightarrow{p_2, m_2} H_2$ 是并行独立的。

如果 l 是同构的，那么规则 $p: L \xleftarrow{l} K \xrightarrow{r} R$ 是非删除的。检测 l 是否为一个同构实际上只要检测 l 的相关图 C 是一个空集。如果图以它们它们的邻接矩阵这一形式存储，那么相关图 C 的算法复杂度是 L 中的点和边数字的平方。

3.2 第二第 2 步优化

在检测了删除规则后，我们只保留那些只有一个非删除规则和一个删除规则的规则对。对于这样的规则对，只运算那些导致关键对的交叠。这可以通过在交叠中鉴别出至少一个被删除规则删除的元素。在下面的定义和定理中，详细地阐明了怎样处理这样一个交叠。

定义 3.15 [冲突条件 - 与兼容性条件]

已知一个无删除规则 $p_1: L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1$ 和一个规则 $p_2: L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2$ ，有 $l_2: K_2 \rightarrow L_2$ 边界图 B_2 和相关图 C_2 。

(1) 当且仅当 SS 满足下列条件的最小图时，图 SS 和两 2 个映射 $o_1: S \rightarrow L_1$ 和 $o_2: S \rightarrow C_2$ 满足冲突条件：1) SS 是 $L_1 \times C_2$ 的一个子图，伴随映射 $o_1: S \rightarrow L_1$ 和 $o_2: S \rightarrow C_2$ ；2) $o_2(S)$ 不是 B_2 的一个子图。

(2) 两 2 个映射 $m_1: L_1 \rightarrow K$ 和 $m_2: L_2 \rightarrow K$ 满足兼容性条件当且仅当：1) m_1 和 m_2 是联合满射；2) $m_1 \circ o_1 = m_2 \circ o_2$ ；3) $m_1(m_2)$ 满足 $P_1(P_2)$ 的胶合条件。

定理 3.1 已知一个无删除规则 $p_1: L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1$ 和一个规则 $p_2: L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2$ ，总有以下成立：

(1) 每一个 (S, o_1, o_2) 满足冲突条件和每一对图映射 $m_1: L_1 \rightarrow K$ ， $m_2: L_2 \rightarrow K$ 满足兼容性条件，则那么可以引出关键对 $P_1 \xleftarrow{p_1, m_1} K \xRightarrow{p_2, m_2} P_2$ 。

(2) 对于每一个关键对 $P_1 \xleftarrow{p_1, m_1} K \xRightarrow{p_2, m_2} P_2$ 存在一个图 S 和 2 个映射 $o_1: S \rightarrow L_1$ ， $o_2: S \rightarrow C_1$ ，则 (S, o_1, o_2) 满足冲突条件，且匹配 $m_1: L_1 \rightarrow K$ ， $m_2: L_2 \rightarrow K$ 满足兼容性条件。

冲突条件与兼容性条件如图 3 所示。

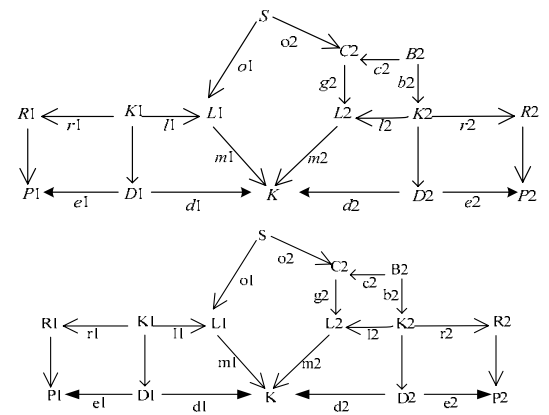


图3 冲突条件 - 与兼容性条件

对于每一个关键对 $P_1 \xleftarrow{p_1, m_1} K \xRightarrow{p_2, m_2} P_2$ 存在一个图 S 和两个映射 $o_1: S \rightarrow L_1$ ， $o_2: S \rightarrow C_1$ ，这 (S, o_1, o_2) 满足冲突条件，且匹配 $m_1: L_1 \rightarrow K$ ， $m_2: L_2 \rightarrow K$ 满足兼容性条件。

3.3 优化算法

根据定义 3.15 满足冲突条件：

(1) S 是 $L_1 \times C_2$ 的子图。 (C_2 是被 p_2 删除的部分， L_1

是 p_1 要使用的部分)。

(2) $o_2(S)$ 不是 B_2 的子图。(防止删除的点不是为了满足胶合条件而人为加上去的点)。

根据定义 3.15 满足兼容性条件：

(1) $m_1 \wedge m_2$ 是联合满射。(保证最小化)；

(2) $m_1 \circ o_1 = m_2 \circ g_2 \circ o_2$ (满足 K 中有 L_1, C_2 的重叠部分，所以因此冲突)；

(3) $m_1 \wedge m_2$ 满足胶合条件。

优化算法的具体步骤如下：

(1) 计算出 $C_1 \wedge C_2$ ；

(2) 穷举所有 S 的可能性，然后推出 (K, m_1, m_2) ；

(3) 保证 $m_1 \wedge m_2$ 联合满射；

(4) 满足胶合条件；

(5) 求出关键对。

基本算法中由于要计算所有的导致交叠产生的对 (m_1, m_2) ，因而因此，对于每对规则的关键对计算量是 L_1, L_2 中端点数和边数总和的指数集。使用优化方法，我们就可直接得出这对规则的关键对。这是因为只有一个图 S 满足这些规则的冲突条件，只有一对规则对下的映射满足与图 S 相关的一致性条件。在由两 2 个非删除规则构成的对中，由于只对子图进行计算，所以因此算法的复杂度仅为求出交叠后的 L_1 和 L_2 中其他部分的节点和边总和的平方。

4 NAC 条件检测方法

图转换系统(Graph Transformation System, GTS)(Gts)中的每一对规则可导致一些关键对。通过计算每一对规则可得出一个 gtsGTS 的所有关键对。对一个给定带 NAC_s 的 gtsGTS^[4]，可直接计算出关键对。在此只给出直接构建的方式^[4]。用类似的方法可以给出一个更为有效的构建禁止-产生冲突和产生-禁止冲突关键对的方式，当然还需继续研究。关键对构建如图 4 所示。

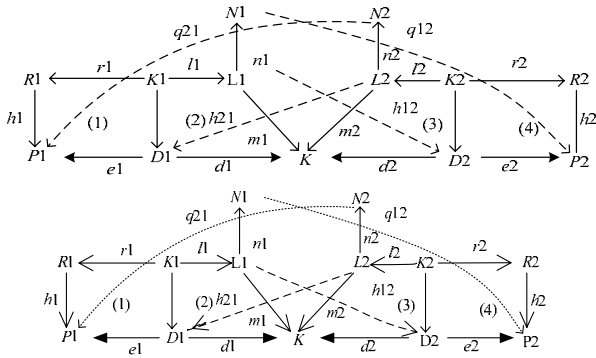


图 4 关键对构建

给定一对含 NAC_s 的规则 $(p_1: L_1 \leftarrow K_1 \rightarrow R_1, p_2: L_2 \leftarrow K_2 \rightarrow R_2)$ ：

步骤 1： 考虑所有带 $NAC_s NAC_s$ 的联合满射映射对 $(m_1: L_1 \rightarrow K, m_2: L_2 \rightarrow K)$ ：

a. (1) 检验 (l_1, m_1) 和 (l_2, m_2) 的相关性条件，如果满足，那么就在图 4 中的(2)(3)构建推出补集(PO-complements) D_1, D_2 ，在图 4 中的(1)、(4)构建推出 P_1, P_2 ；。

b. (2) 检测有向图转换对是否产生删除-使用冲突或使用-删除冲突，即是否会产生关键对。

步骤 2： 考虑 p_1 的每一个 $NAC n_1: L_1 \rightarrow N_1$ 时，每一个单射 q_{12} 构成的联合满射对 $(h_2: R_2 \rightarrow P_2, q_{12}: N_1 \rightarrow P_2)$ ：

a. (1) 检验 (h_2, r_2) 的相关性条件，若满足，则在图 4 中由(4)构建推出补集(PO complement) D_2 ；

b. (2) 若 $m_2 \models NAC_{p_2}$ ，那么在图 4 中的(3)中构建推出 K ，并中止；

c. (3) 检验 $h_{12}: L_1 \rightarrow D_2$ 的存在性，即 $e_2 \circ h_{12} = q_{12} \circ n_1$ 是否成立(e_2 单射也就意味着 h_{12} 唯一)；若不存在，则中止；

d. (4) 定义 $m_1 = d_2 \circ h_{12}: L_1 \rightarrow K$ ，若 $m_1 \models NAC_{p_1}$ ，则中止；

e. (5) 检验 (m_1, l_1) 的相关性条件，若满足，则在图 4 中的(2)构建推出补集(PO complement) D_1 ；

f. (6) 在图 54 中的(1)构建 P_1 如 PO ，导出 $P_1 \leftarrow K \Rightarrow P_2$ 的关键对。

步骤 3： 考虑 p_1 的每一个 $NAC n_2: L_2 \rightarrow N_2$ 时，每一个单射 q_{21} 构成的联合满射对 $(h_1: R_1 \rightarrow P_1, q_{21}: N_2 \rightarrow P_1)$ ，与步骤 2 类似。

5 有效关键对(essential critical pairs)方法的检测冲突检测

考虑到规则冲突结构中存在三 3 种冲突：规则自身并行运行两 2 次、对称冲突、非对称冲突。对应的图结构中存在对称结构和非对称结构。针对对称结构，可以进一步减少冲突检测的计算。

5.1 有效关键对定义

针对每个冲突原因，存在一个有效关键对，通过它可用最小上下文来表示由这一冲突原因导致的冲突。

定义 5.16 冲突条件(conflict condition)

给定一对有向转换 $H_1 \xleftarrow{p_1, m_1} G \xrightarrow{p_2, m_2} H_2$ 。

* (1) 若不存在 $s_1: S_1 \rightarrow B_1 \in M$ 使得 $c_1 \circ s_1 = o_1$ 成立，那么对于结构 $(S_1, o_1: S_1 \rightarrow C_1, q_{12}: S_1 \rightarrow L_2)$ ，拉回(pullback) $(m_1 \circ g_1, m_2)$ 满足冲突条件，如图 5 所示。

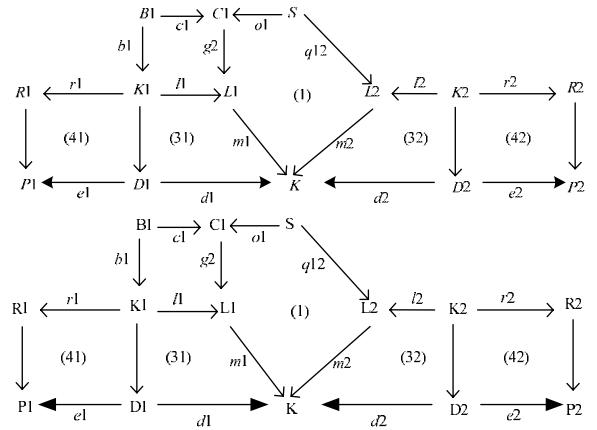


图 5 冲突条件：拉回 $(m_1 \circ g_1, m_2)$

* (2) 若不存在 $s_2: S_2 \rightarrow B_2 \in M$ ，使得 $c_2 \circ s_2 = o_2$ 成立，那么对于结构 $(S_2, q_{12}: S_2 \rightarrow L_1, o_2: S_2 \rightarrow C_2)$ ，拉回(pullback) $(m_1, m_2 \circ g_2)$ 满足冲突条件。

定义 5.27 (冲突原因范围)

给定产生冲突的有向转换对 $H_1 \xleftarrow{p_1, m_1} G \xrightarrow{p_2, m_2} H_2$ ，冲突原因

生成有如下三 3 种情况：

* (1) 在结构 $(S_1, g_1 \circ o_1, q_{12})$ 中, (S_1, o_1, q_{12}) 满足冲突条件, (S_2, q_{21}, o_2) 不满足。

* (2) 在结构 $(S_2, q_{21}, g_2 \circ o_2)$ 中, (S_1, o_1, q_{12}) 不满足冲突条件, (S_2, q_{21}, o_2) 满足。

* (3) 在结构 (S_1, s_1, s_2) 中, (S_1, o_1, q_{12}) 和 (S_2, q_{21}, o_2) 都满足冲突条件(对称冲突)。

定义 5.38 (有效关键对)

一个有向转换对 $P_1 \xleftarrow{p_1, m_1} K \xrightarrow{p_2, m_2} P_2$ 通过规则 (p_1, p_2) 进行, 若满足: $P_1 \xleftarrow{p_1, m_1} K \xrightarrow{p_2, m_2} P_2$ 产生冲突, 并且 (K, m_1, m_2) 是冲突原因范围中所定义的冲突原因生成 $(S_1, g_1 \circ o_1, q_{12})$ 、 $(S_2, q_{21}, g_2 \circ o_2)$, (S_1, s_1, s_2) 的推出, 那么 $P_1 \xleftarrow{p_1, m_1} K \xrightarrow{p_2, m_2} P_2$ 是有效关键对。

证明: 每个有效关键对是产生冲突的有向转换, 有效关键对 (m_1, m_2) 通过推出来构建, 因而 (m_1, m_2) 联合满射。

5.2 有效关键对的性质

在图转换系统中, 检测出所有有效关键对就可检测出所有冲突。因此因而, 我们得出有效关键对集合具有以下 3 个性质:

(1) 首先系统中的每一个冲突都可用一个有效关键对来表示。

其次, (2) 依据冲突原因范围, 每一个有效关键对都是唯一的。

最后, (3) 基于有效关键对可证明局部汇合。

有效关键对的原理不仅简化了静态冲突检测, 而且, 由于有效关键对集合仅为关键对集合的子集, 因而因此需检验严格汇合的关键对数减少了, 从而简化了系统中冲突的汇合性

分析。

6 总结结束语

AGG 允许自动检测一个 GTS 中的关键对集合。本文提出的优化方法能有效检测冲突被得到, 这是因为因为两个原因: 首先, 规则被分析, 检查它们是否真的产生冲突, 其次, 并且用一个积极的方法计算每一个冲突的最小上下文。AGG 允许自动检测一个 gts 中的关键对集合, 其算法需要对所有规则对进行遍历。下一步的研究方向是对本文所探讨的优化方法可以提高检测效率。NAC 条件检测方法、有效关键对方法均有进一步研究的必要的深入探讨。

参考文献

- [1] Mens T, Taentzer T G, Runge O. Detecting structural Structural rRefactoring Cconflicts uUsing Ccritical pPair aAnalysis[J]. Electronic Notes in Theoretical Computer Science, 2005, 127(3): 113.-128.
- [2] D. Plump D. Critical Pairs in Term Graph Rewriting[C]// Proc. of Mathematical Foundations of Computer Science, . [S. l.]: Springer-Verlag, Lecture Notes in Computer Science 841, pp. 556-566. Springer-Verlag, 1994.
- [3] Lambers, L., Ehrig, H., Orejas, F.: Efficient detection Detection of cConflicts in gGraph-based mModel tTransformation[J]. Electronic Notes in Theoretical Computer Science, 2006, 152(2) (2006) .pp.: 97—109.
- [4] Lambers, L., Ehrig, H., Orejas, F.: . Conflict Detection for Graph Transformation with Negative Application Conditions[C]//Proc. of ICGT 200'06. [S. l.]: Springer-Verlag LNCS, vol. 4178, pp. 61–76, 2006..

编辑 金胡考

(上接第 39 页)

- [8] Genrich H J. Predicate/Transition Nets[C]//Proc. of the 4th Advanced Course on Petri Nets: Central Models and Their Properties, Advances in Petri Nets. Bad Honnef, Germany: [s. n.], 1986.

- [9] Ratzer V A, Wells L, Lassen M H, et al. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets[C]//Proc. of International Conference on Applications and Theory of Petri Nets. Eindhoven, Netherlands, [s. n.], 2003.

编辑 金胡考