

基于 tableau 的自动推理技术综述 *

刘全^{1,2,3} 孙吉贵^{2,3} 于万钧^{2,3}

(苏州大学计算机科学与技术学院 苏州 215006)¹ (吉林大学计算机科学与技术学院 长春 130012)²
(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)³

摘要 tableau 方法是一种接近于逻辑系统表示的自动推理方法,由于其直观性和通用性,易于计算机实现,成为目前最普及的自动推理方法之一。在提高系统效率方面,主要在相关技术和策略、理论和方法等方面进行了分析,并展望了未来的研究方向。

关键词 Tableau, 技术, 策略, 理论, 方法

Automated Reasoning Based on Tableau

LIU Quan^{1,2,3} SUN Ji-Gui^{2,3} YU Wan-Jun^{2,3}

(College of Computer Science and Technology, Suzhou University, Suzhou 215006)¹

(College of Computer Science and Technology, Jilin University, Changchun 130012)²

(Ministry of Education Laboratory of Symbol Calculation and Knowledge Engineering, Jilin University, Changchun 130012)³

Abstract Tableau is an automated reasoning method approaching logic system. Tableau method has become one of the widespread automated reasoning due to its universal and audio-visual properties and easy to implement. In recent years many research results have been brought in related technology and tactics, theory and method in order to improve greatly in efficiency. Finally proposed is possible research trend of the development of tableau in the future.

Keywords Tableau method, Technology, Theory, Method

自动推理作为自动定理证明的扩展是人工智能研究的基础工作,许多重要的人工智能系统,诸如专家系统、问题解答系统、学习系统、程序的自动综合与合成、医疗诊断、信息检索等,都是以推理系统为其核心部分,所以自动推理的研究,将对人工智能的其它分枝产生深远的影响,它所提出的推理方法已被应用于人工智能的各个领域。

1955年,由 Beth 提出“tableau”术语,当时的主要目的是寻找一个反例。Beth 明确论述了 tableau 方法与相继式算子及自然演绎的关系,给出了一个基于 tableau 的 Gentzen 消除理论的证明。从中可以看出,tableau 方法的实质是将语义结构中的二元关系显式地表现出来。换句话说,就是通过引入相应的谓词,将二元关系的性质用逻辑公式来表示。对于不同的逻辑系统,所使用的 tableau 规则是相同的,只是对公式构造集进行扩展,使之更接近相应的逻辑系统。由于 tableau 方法具有较强的通用性和直观性,从 20 世纪 60 年代开始,引起了以 Smullyan^[1]、Fitting^[2] 为代表的计算机科学家的广泛兴趣,同归结一样,被认为是重要的自动推理方法之一。多年来,人们也一直在寻求各种各样的 tableau 方法,以求提高效率和易于计算机实现。

1 tableau 及扩展规则

定义 1 (tableau) 令 $\{A_1, A_2, \dots, A_n\}$ 为公式的有限集合。

(1) 下列分枝树为公式 $\{A_1, A_2, \dots, A_n\}$ 的一个 tableau:

A_1

A_2

⋮

A_n

(2) 如果 T 为 $\{A_1, A_2, \dots, A_n\}$ 的一个 tableau, 且 T' 为 T 应用 tableau 扩展规则后的结果, 那么 T' 也是 $\{A_1, A_2, \dots, A_n\}$ 的一个 tableau。

利用 tableau 方法进行定理证明, 将被证明的公式分为四类, 即 α -、 β -、 δ -、 γ -公式, 各类公式对应的规则如下:

定义 2 (tableau 扩展规则)。

$$\frac{\alpha}{\alpha_1} \quad \frac{\beta}{\beta_1 | \dots | \beta_n}$$

⋮

α_n

$$\frac{\gamma}{\gamma_1(y)} \quad \frac{\delta}{\delta_1(f(x_1, \dots, x_n))}$$

y 是一个自由变量 f 是一个新的 Skolem 函数符号,
 x_1, \dots, x_n 是分枝中出现的自由变量

2 相关技术和策略的研究

2.1 含量词 tableau 改造技术

在 tableau 证明中, γ -公式对其中包含的自由变量需要进行多次不同的实例化。典型的例子是群论中的结合律 $(\forall x)(\forall y)(\forall z)((x \cdot y) \cdot z \approx x \cdot (y \cdot z))$, 应用该定律即使证明一个非常简单的定理, 也要对其中的 x, y, z 进行多次不同的实例化替换。因此在 tableau 实现时, 对 γ -规则应用次数 h

*) 国家自然科学基金(60073039)、教育部骨干教师基金和吉林省自然科学基金资助课题(2000540)。刘全 博士, 副教授, 主要研究方向为人工智能、自动推理、地理信息系统。孙吉贵 博士, 教授, 博士生导师, 主要研究方向为人工智能、约束程序设计、智能决策支持系统。

的限制至关重要,限制的次数太少,不能满足推理所需要的实例,使一个可满足的公式得不到 tableau 证明。限制的次数过多,由于 γ 规则使用次数的不确定性,必然会导致大量的冗余推理产生,使证明过程的搜索空间膨胀。

文[2]中提出了循环控制 γ 规则应用次数 h 的技术,即 h 从 1 开始,如果未能找到 tableau 过程, h 的值逐渐增大,直到找到 tableau 或达到最高限定次数为止。实现过程如下,其中 VarLim 为最高限定次数:

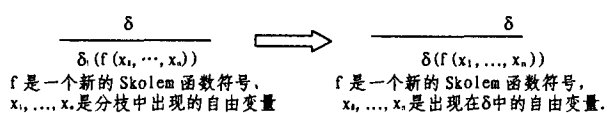
```

increase_prove(Fml, H): -(Fml, [], [], [], H).
increase_prove(Fml, H): - NewH is H + 1, H <=
VarLim,
increase_prove(Fml, NewH).
    
```

证明器始于 `increase_prove(Fml, N=1)`。H 的搜索值逐渐变为 $N, N+1, \dots$ 。

tableau 在扩展中 γ 规则将全称变量替换成一个任意变量,由于替换的任意性,可导致在同一 tableau 证明中, γ 规则被多次使用,使得 tableau 结构中出现多个自由变量。另外, δ 规则要求被 Skolem 化的函数符号是新的,且函数中包含分枝中出现的所有自由变量。显然,由于 γ 规则的不确定性以及 δ 规则的限制,可使一个简单的证明变得很复杂,延迟了 tableau 的封闭时间。

在文[3]中,对 δ 规则进行了改进,称 δ^+ 规则。改进后的系统对 δ 公式在实现推理空间上有较大的改进。



δ^+ 规则与 δ 规则的区别是:在 δ 规则中 x_1, \dots, x_n 是指出现在分枝中的自由变量,在 δ^+ 规则中 x_1, \dots, x_n 是指出现在 δ 中的自由变量。

2.2 tableau 非相关部分剪枝技术

tableau 非相关部分剪枝 (Pruning Irrelevant Parts of a Proof) 技术是由 Oppacher 提出的 tableau 压缩技术,通过这种技术可以有效地减少 tableau 证明的搜索空间,在文[4]也称为层次剪切 (level cut) 技术,在文[5]中考虑了一种非 Horn 集转换的特殊情况。

如果一个 tableau 分枝 B 应用 β 规则扩展后,其中一个封闭的子 tableau T_i 在不利用扩展分枝 β_i 的情况下仍然是封闭的,那么其它的任何包含扩展分枝 $\beta_j (j \neq i)$ 的子 tableau $T_j (j \neq i)$ 都是封闭的,整个分枝 B 也是封闭的。这种技术的优点在于①封闭规则改变使得所有的包含 B 的 tableau 分枝作为一个子分枝去考虑;②包含 B_j 之一的所有分枝被剪枝,即受到 β 规则影响的分枝不需要再扩展。情况如图 1 所示。

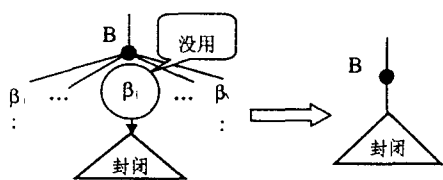


图 1 tableau 非相关部分剪枝技术

2.3 因子分解技术

含因子的 tableau 与归结定理证明^[6]相关。假设分枝 B 未封闭,但存在一个替换 σ ,使得 $B' \subseteq B\sigma, B'$ 已经封闭,那么可

以判定 B 也是封闭的。因子分解技术在文[7]作为模型删除而引入,文[8]用于关系算子中,但由于格式的限制,只能用于深度优先选择函数中。考虑一个封闭 tableau 包含节点 N_1 和 N_2 ,假如 N_2 的所有后继节点也是 N_1 的后继节点,那么 N_2 下面的部分封闭 tableau T 也可以作为一个解节点,由 N_1 节点来访问,因为在 N_2 下的 T 中,所有的扩展和推理都可能在 N_1 下的 T 中使用。如果 N_1 的文字和 N_2 的文字能合一,或 N_2 的后继节点集合是 N_1 的后继节点集合的子集,那么将子目标 N_1 标记为解。

考虑公式集 $\{p \vee q, p \vee \neg q, \neg p \vee q, \neg p \vee \neg q\}$ 应用因子分解技术产生一个较短的证明,见图 2,因子分解步用弧线表示。显然为了保证有效性,规则中必须加以限定,禁止循环解出现,文[9]中提出了回归合并 tableau 方法,用因子依赖关系对循环解问题进行了有效解决。

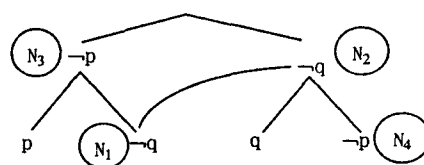


图 2 因子分解技术实现公式集 $\{p \vee q, p \vee \neg q, \neg p \vee q, \neg p \vee \neg q\}$ 的证明过程

2.4 包含删除策略

定义 3(公式树压缩) 一个公式树 T 称为公式树 T' 的压缩,如果对于某一 $n \geq 0, T'$ 是通过向 T 的 n 非叶节点附加 n 公式树而得到的。

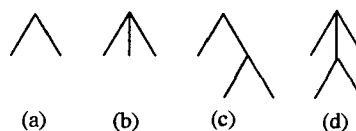


图 3 树压缩概念说明

在图 3 中,(a)图是(d)图的压缩树,但不是(c)图的压缩树;(c)图是(d)图的压缩树。

定义 4(公式树的包含) 如果 T 是比 T' 的公式树压缩更一般的,则称公式树 T 包含公式树 T' 。

定义 5(包含删除) 在 tableau 搜索树 T 中,对于任何一对节点 N 和 N' ,如果 N 的 tableau 目标树包含 N' 的 tableau 目标树,那么以 N' 为根搜索树的整个子树从 T 中删除。

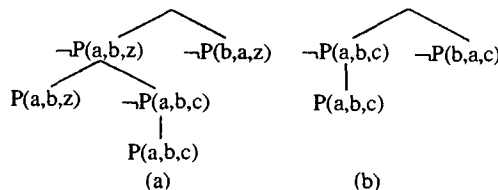


图 4 两个 tableau 的目标树的包含关系

包含删除策略保持了证明搜索的完备性,在图 4 中,(a)和(b)均为子句集合 $\{\forall x \forall y \forall z (\neg P(x, y, z) \vee \neg P(y, x, z)), \forall x \forall y \forall z \forall v (P(x, y, z) \vee \neg P(x, y, v)), P(a, b, c)\}$ 的 tableau,由于 z 是自由变量,(a)中的目标树 $(\oplus \neg P(b, a, z))$ 包含(b)中的目标树 $(\oplus \neg P(b, a, c))$,当(b)封闭时,(a)也是封闭的。这样(b)和它的所有可能的扩展能被完全忽略掉。

3 理论和方法的研究

3.1 子句 tableau 方法

在 tableau 实现中,许多用来提高系统效率的方法都要求将公式化成子句形式。这主要有两方面的原因:一方面将公式化为子句形式时,不影响它的可满足性,只是公式的大小呈幂指数的增加。另一方面,也是更重要的方面,由于子句形式的公式存在统一的结构,规则更为简单,更容易探索出简单而实用的消除冗余的技术。因此近年来人们在子句 tableau 方面取得了一定的成果^[10~12]。

定义 6(子句 tableau) 对于子句集 S 的子句 tableau T 为一个文字树 $\langle t, \lambda \rangle$, 任何节点 t 的后继子节点序列标记为文字 K_1, \dots, K_n , 存在一个替换 σ 和一个子句 $L_1 \vee \dots \vee L_n \in S$ 使得 $K_i = L_i \sigma (1 \leq i \leq n)$ 。对于任何 $N_i (1 \leq i \leq n), K_1 \vee \dots \vee K_n$ 称为 N_i 的 tableau 子句。

利用子句 tableau 进行证明的首要一步是将带符号公式转化为子句,通常采用结构保留子句转换方法^[13,14],随着自动定理证明方法的发展,人们在大量的实践中认识到,仅仅证明一个蕴涵式恒真或公式集不可满足是不够的,还要讨论其可满足性,并且给出其模型,即使公式取真的变量赋值,这就是基于子句 tableau 的模型生成研究。目前 tableau 的子句转换问题即为 2-SAT 问题^[15],一般的公式模型生成问题和很多实际问题密切相关,将取代狭义的 SAT 问题成为新的研究热点。在子句 tableau 推理效率的研究方面,人们探索较多的是包含删除策略,这种方法主要在 tableau 中找出被包含的公式树的模型,然后进行消减,而不影响公式的可满足性,但由于包含模型的查找困难、不易判别,因此该方法受到一定的局限性。在此基础上,Bibel 提出了连接 tableau 方法,对公式的扩展进行了限制,除了开始节点外,每个节点都是一个互补的后继节点。这样就可以将互补的节点删除,产生一个目标树,对于任何深度优先最右分枝选择函数,用 $[L]F$ 替换合取 $L \wedge F$,用 $L_1 \dots L_n$ 代替析取 $L_1 \vee \dots \vee L_n$,在大括号中存在模型删除链,即将整个链删除。如果一个公式是可满足的,那么它最终对应着一个空目标树。连接 tableau 方法存在的问题是,对扩展分枝深度的限制方面^[16],限制的次数太多,可能产生大量的冗余分枝,限制的太少,有可能找不到相应的证明。最近文^[15]研究了一种基于公平策略的不回溯的深度优先证明搜索方法,这种方法使用了权重排序和 tableau 模型包含相结合,在一定程度上避免了循环证明的问题。

3.2 超 tableau 方法

基于超扩展规则的 tableau 算子受到广泛的关注^[17~19]。超算子具有使几个推理步结合成一个的共享特征,这使证明搜索速度得以提高。它的主要优点在于:在开始时不必计算中间结果,这使考虑的搜索空间会相应变小。事实上,早在文^[20]超 tableau 就被提及过,但这个研究没有产生它所应有的影响,直到 1988 年,作为模型产生理论被文^[21,22]重新提出,它基本上是超 tableau 的一个变种。

在归结中,超归结能被看成是语义归结的一个实例,这对于超 tableau 同样适用。

定义 7(超 tableau) 用在 B 上的扩展步的每个子句必须从集合 $\{C | C \in S, I_B \neq C \text{ 且 } \overline{f(C)} \subseteq B\}$ 中取得。每一个扩展步,包含扩展子句的被选文字的补分枝标记为封闭的 tableau 称为超 tableau。

EP-tableau^[23] 是对不含函数符号的非子句一阶逻辑的正超 tableau 的变种,主要是:①称为约束量词(PRQ 公式)的正公式的值约束非子句版本,特别是形如 $(\forall x)(\phi(x) \rightarrow \Psi)$ 的 γ -类型公式,并且用在一个分枝 B 上,对于某一 σ ,使得 $I_B \models \phi \sigma$;

②只有含 $I_B \models \Psi$ 的公式 Ψ 被加入到分枝 B 上;③使用修改的 δ 规则,即枚举目前的模型域

$$\begin{array}{c} \delta(x) \\ \hline \delta_i(c) \quad \dots \quad \delta_i(c_n) \quad \delta_i(\text{sk}o\phi) \\ \text{c}_i, \dots, \text{c}_n \text{ 是目前分枝上的所有常量} \end{array}$$

文^[23]对值约束情况进行了改进,替换只应用到输入子句的新实例中,即如果一个在 tableau 分枝上文字 L 的一个实例 $L\sigma$ 要求被扩展或推理,那么 σ 应用到包含 L 的子句的实例中,并且结果被加入到输入子句集中,使输入子句动态增加。文^[24]有一个不同的方法,不需要修改输入子句,而是修改 tableau 本身。

3.3 决策图 tableau(TDD)方法

二值决策图技术(Binary Decision Diagrams,简称 BDD)是一种表示布尔公式的有效数据结构。1986 年由 Bryant^[25]提出,它的主要优点是能表达一个很大的可满足的公式模型。将 tableau 规则与 BDD 技术相结合(Tableau Decision Diagrams,简称 TDD 方法)来实现 tableau 证明,在效率上有很大的提高。在文^[26]中介绍了一些很适用的 BDD 方法,这些方法主要用于硬件验证中^[27]。TDD 方法不仅可以用于经典逻辑中,而且可以用于直觉逻辑、模态逻辑^[28]等非经典逻辑中。

定义 8(二值决策图 BDD) 二值决策图 F 递归地定义为 1、0 或三元组 $(s\Phi, F_0, F_1)$ 。

二值决策图 BDD 可以如图 5 表示。三元组 $(s\Phi, F_0, F_1)$ 表示为 $s\Phi \rightarrow F_0/F_1$ 。

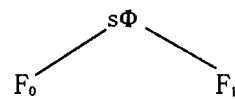


图 5 BDD 中的一个节点

BDD 技术是基于三元 if-then-else 连接的。例如: $p \wedge q$ 等价于 if p then (if q then 1 else 0) else 0。假设出现在 ϕ 中的原子为 $\{p_1, \dots, p_m\}$, 表示为 $\phi(p_1, \dots, p_m)$, 那么,反复地应用 $\phi(p_1, \dots, p_m) = \text{if } p_1 \text{ then } \phi(1, \dots, p_m) \text{ else } \phi(0, \dots, p_m)$, 用函数替换公式中的自由变量,即可得到相应的 BDD 表示。

通常 BDD 假定是递减有序的(ROBDD),递减意味着 BDD 的语法树可以通过识别同构子树和应用基于 (if i then j else j) = j 的简化而得到,有序意味着对于给定的原子的全序关系 $<$,对于任何 q ,一定有 if p then $p < q$ 成立。

BDD 技术可以被扩展到多值逻辑中,即多值决策图(Many-Valued Decision Diagrams,简称 MDD),对于 n 值逻辑 case i of

$$0 : j_0; \quad \frac{1}{n-1} : j_{\frac{1}{n-1}}; \quad \dots \quad 1 : j_1; \quad \left\{ \begin{array}{ll} j_0 & \text{if } i=0 \\ j_{\frac{1}{n-1}} \dots & \text{if } i=\frac{1}{n-1} \\ \dots & \\ j_1 & \text{if } i=1 \end{array} \right.$$

esac

定义 9(决策图 tableau TDD) 给定一个 BDD F ,那么与 F 相关的 tableau $T(F)$ 定义如下:

- (1) $T(0)$ 为空 tableau \emptyset ;
- (2) $T(1)$ 为只包含空路径 $\{\emptyset\}$ 的 tableau;
- (3) $T(s\Phi \rightarrow F_0/F_1) = \{\{s\Phi\} \cup \{p \in T(F_0) \cup T(F_1)\}\}$

在 TDD 方法中,对 α 公式的处理方法是:如果 A 和 B 都是 BDD,那么对于 A 中的 1 叶节点用 BDD 中的 B 来代替。

对 β -公式的处理方法是:将 β -规则改成

$$\frac{A \vee B}{A \mid \rightarrow A \wedge B}$$

以 $\{1\}(p \vee q)$ 为例,对 tableau 与 BDD、MDD 关系描述如图 6。

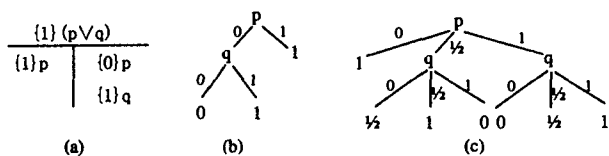


图 6 tableau 与 BDD、MDD 关系描述图

在(b)中,then 和 else 分枝分别标记为 1 和 0,一个来自于节点 p 标记为 i 的边可以看作将真值 i 赋给 p ,对于 BDD 中,以 1 结束的分枝为开放分枝,以 0 结束的分枝为封闭分枝,将该方法扩展到 MDD 中,对于三值 lukasiewicz 逻辑 $\{1/2\}(p \rightarrow q)$ 的 MDD 推理如图(c)。

3.4 翻译成整数规划的 tableau 方法

翻译成整数规划(Integer Programming, 简称 IP)的 tableau 方法的主要思想是:将不需要化成任何范式的公式 ϕ 翻译成 0-1-IP C ,使得公式 ϕ 是不可满足的当且仅当 C 是可行的。Tableau 规则用于劈分和改造 ϕ ,利用 IP 方法检查结果 tableau 是否是封闭的。

对于命题 CNF 公式,每个形如

$$p_1 \vee \dots \vee p_k \vee \neg p_{k+1} \vee \dots \vee \neg p_m \quad (1 \leq k \leq m)$$

对应的约束为

$$p_1 + \dots + p_k + (1 - p_{k+1}) + \dots + (1 - p_m) \geq 1$$

一个 tableau 分枝 B 是封闭的翻译成 0-1-IP 即为: B 是封闭的当且仅当约束集合 $\{p \geq 1; p \in B, p \text{ 为原子}\} \cup \{p \leq 1; \neg p \in B, p \text{ 为原子}\}$ 是可行的。翻译成 0-1-IP 后,一个 tableau T 是封闭问题转换成析取规划问题: T 是封闭的当且仅当构成每个分枝的 IP 有一个解。

定义 10(与 Φ 相关的 0-1-IP) 令 $\Phi = \{\phi_1, \dots, \phi_k\}$ 为公式集, C_0, \dots, C_n 为根据下列规则形成的约束序列:

(1) $C_0 = \{\boxed{\geq 1} \phi_1, \dots, \boxed{\geq 1} \phi_k\}$ 。

(2)a) C_m 来自于 C_{m-1} ,通过对 $\Psi \in C_{m-1}$ 应用 α^- 、 β^- 、 δ^- 、 γ -规则。

b) 有一个替换 σ 使得 $C_m = C_{m-1} \cup (C_{m-1}\sigma)$ 。

那么 C_n 为与 Φ 相关的 0-1-IP。

令 $\Phi = \{p \vee \neg p\}$,那么 $C_0 = \{\boxed{\geq 1}(p \vee \neg p)\}$ 。应用 β 规则得到 $C_1 = \{\boxed{\geq 1-j} p, \boxed{\geq 1+j-1} \neg p\}$,这样可得到与 Φ 相关的 0-1-IP 规则,即

$$C_3 = \{p \geq 1-j, 1-p \geq 1+j-1\}$$

由于 C_3 是可行的,因此 $p \vee \neg p$ 是可满足的。

结论 随着人工智能技术的不断发展,自动推理技术在效率方面要求越来越高,基于 tableau 的推理系统将在技术、策略、理论和方法等方面得到更进一步的发展和提高。对于自动推理来说,考核推理效率的一个重要的指标就是推理所用的时间和空间。在 tableau 推理中也同样存在着这方面的问题。例如对 γ -公式实例化的次数的限定,和 δ -公式中函数符号和自由变量的限制,限制的控制处理不得当可使一个简单的证明变得很复杂,延迟了 tableau 的封闭时间,甚至得不到证明。另外,由于等词的对称性,对于含等词的 tableau 有时会出现搜索空间膨胀以及死循环的现象。向非经典逻辑扩展方面,虽然经典逻辑的相应规则和方法可以直接应用到非

经典逻辑,但有时会出现大量的冗余分枝,使计算机无法处理。

由于 tableau 方法存在通用性、直观性及易于计算机实现等特点,因此具有广阔的前景,近年来,在 tableau 技术、策略及方法上有大量的研究,取得了非常可喜的成果,推动了自动推理技术的发展,将在诊断、自然语言理解、专家系统、推理数据库等方面有广泛的应用前景。

参 考 文 献

- Smullyan R M. First-order logic. Revise second edition. New York: Springer-Verlag, 1994
- Fitting M. First-order logic and automated theorem proving. New York: Springer-Verlag, 1996
- Hahnle R, Many F. Transformations between signed and classical clause logic. In: Proc. of ISMVL 99, IEEE Press, 1999. 248~255
- Baumgartner P, Furbach U, Niemelä I. Hyper tableaux. In: Alferes J J, Pereira L M, eds. In: Proc. European Workshop: Logics in Artificial Intelligence, JELIA, Vol. 1126 of LNCS, Springer-Verlag, 1996. 1~17
- Koshimura M. A proof of completeness for non-horn magic sets and its application to proof condensation. In: Murray N ed. Position Paper, International Conference on Automated Reasoning with Analytic Tableaux and Related Methods. Technical Report 99-1, Institute for Programming and logixs, Department of Computer Science, University at Albany-SUNY, 101~116
- 刘叙华. 基于归结方法的自动推理. 北京: 科学出版社, 1994
- Kowalski R A, Kuehner D. Linear resolution with selection function. Artificial Intelligence. 1971, 2: 227~260
- Bibel W. Automated theorem proving. Second edition, Vieweg Verlag, Braunschweig, 1987
- Wallace K, Wrightson G. Regressive merging in model elimination tableau-based theorem provers. Journal of the Interest Group in Pure and Applied Logics. 1995, 3(6): 921~938
- Leta R. Using matings for pruning connection tableaux. In: Proc. 15th Intl. Conf. on Automated Deduction (CADE). Lindau, Vol. 1421 of LNCS, Springer-Verlag, 381~396
- 刘全, 孙吉贵. 提高一阶多值逻辑 tableau 推理效率的布尔剪枝方法. 计算机学报, 2003, 26(9): 1165~1170
- Waler A. Connections in nonclassical logics. Handbook of Automated Reasoning. Elsevier Science, 2001
- Hahnle R. Transformations between signed and classical clause logic. In: Proc. 29th Intl. Symposium on Multiple-valued Logics, Freiburg, Germany, 1999. 248~255
- Hahnle R. Short conjunctive normal forms in finitely-valued logics. Journal of Logic and Computation, 1994, 4(6): 905~927
- Becker B. Depth-first proof search without backtracking for free variable clausal tableaux. Journal of Logic and Computation, 2000, 6(6): 819~834
- Letz R, Mayr K, Goller C. Controlled Integration of the cut rule into connection tableaux calculi. Journal of Automated
- Bry F, Yahya A. Minimal model generation with positive unit hyper-resolution tableaux. In: Miglioli p, Moscato U, Mundici D and Ornaghi M eds. Theorem Proving with Tableaux and Related Methods, 5th International Workshop, TABLEAUX'96, Italy: Terrasini, 1996. 143~159
- Shults B. A framework for using knowledge in tableau proof. In: Galmiche D, ed. In: Proc. Intl. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods. France: Springer-Verlag, 1997. 328~342
- Baumgartner P. Hyper tableau-the next generation. In harrie de Swart, ed. Automated Reasoning with Analytic Tableaux and Related Methods, volume 1397 of Springer LNAI, 1998

(下转第 15 页)

的动态查询、动态加入、退出算法,有利于网格 workflow 引擎对于网格 workflow 任务的动态调度。创新的将网格组织和资源集成考虑,有利于两者的建模和管理。

4.3 调度问题

网格 workflow 中的调度问题是最重要的问题,不仅仅影响网格 workflow 执行成功与否和效率的高低,同时也涉及到网格 workflow 的资源管理情况,调度引擎中非常重要的功能是根据 workflow 模型生成相应的工作流实例任务,然后根据定义好的调度策略进行调度。当前网格调度中大部分调度策略和算法都是针对元任务的,即调度任务相互之间没有关联,显然不满足网格 workflow 中任务之间是否具有时序或者因果的约束条件,另外网格中考虑具有关联任务的调度采用基于 DAG 的调度方法,对算法缺乏优化,同时调度单一,没有提供多种策略,难以满足网格 workflow 应用的不同需求。

4.4 基于网格 workflow 的网格服务合成

单一的网格服务不一定能够满足各种网格应用需求,如果针对所有的应用都专门开发一个与之对应的网格服务可能代价很高,也是不现实的,因此基于网格 workflow 的网格服务合成能够为网格应用带来极大的方便。随着 WSRF^[21] 的推出,网格研究和 Web Service 研究正走向融合,现在网格服务合成很多采用 Web Service 合成方案,例如 BPELWS^[22]、WSC^[23],利用这些标准虽然能够较方便地定义企业内部、企业之间的业务流程是如何相互协作的,但是这些流程需要用户手工来明确定义,自动化程度不高,因此需要考虑如何利用网格 workflow 来方便网格服务的合成。

结论 网格还在飞速地发展变化之中,网格研究和 Web Services 正走向融合,WSRF 的规范即将颁布,从目前的草案看,WSRF 对于网格应用会带来很大的影响,对于网格 workflow 的相关技术的影响和调整也是业界需重点关注的问题,具体有网格应用的模式会产生什么样的变化,导致网格 workflow 的过程模型会有什么的变化,在 WSRF 框架下,网格的组织和管理如何体现等问题。

参考文献

- Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publisher, 1999
- Kesselman F C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. of Supercomputer Application, 2001, 15(3): 200~222
- Foster I. What is the Grid? A Three Point Checklist. GRIDTo-day, July 20, 2002
- 徐志伟. 织女星网格总体思路. 织女星网格文档, VGD-1, 2001
- Messina P. Metacomputing and Data-Intensive Applications. In: Proc. of the Intl. Conf. on Worldwide Computing and Its Applications, March 1997
- Bivens H P, Beiriger J I. GALE: Grid Access Language for HPC Environments. [http:// sass3186.sandia.gov/hpbiven](http://sass3186.sandia.gov/hpbiven), 2002
- Beiriger J, et al. Constructing The ASCI Grid. In: Proc. of 9th High Performance and Distributed Computing Conf. Pittsburgh, PA, August 2000
- SETI@HOME. <http://setiathome.ssl.berkeley.edu>
- Krishnan S, Wagstrom P, von Laszewski G. GSFL: A Workflow Framework for Grid Services. ANL/MCS-P980-0802, Argonne, 2002
- <http://www.gridforum.org/>
- Bivens H P. Grid Workflow. Global Grid Forum, 2001
- Marinescu D C. A Grid workflow Management Architecture. GGF white paper
- Foster I, Kesselman C, Nick J, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002
- Krishnan S, Wagstrom P, von Laszewski G. GSFL: A Workflow Framework for Grid Services. ANL/MCS-P980-0802, Argonne, 2002
- Marinescu D C. A Grid workflow Management Architecture. GGF white paper
- Graham G E, Evans D, Bertram I, McRunjob. A High Energy Physics Workflow Planner for Grid. Computing in High Energy and Nuclear Physics, La Jolla, California, March 2003
- Aalst J, van der W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 18(3): 21~66
- 张绍华. 网格 workflow 关键技术研究: [复旦大学博士论文]. 2004. 6
- Czajkowski K, Foster I, Karonis N, et al. A Resource Management Architecture for Metacomputing Systems. In 4th Workshop on Job Scheduling Strategies for Parallel Processing, Springer-Verlag, 1998. 62~82
- Frey J, Tannenbaum T, Foster I, et al. Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing, 2002, 5(3): 237~246
- WSFR, <http://www.globus.org/wsrfl/>
- Arkin A, et al. Business Process Modeling Language (BPML), Version 1.0. 2002
- Arkin A, Askary S, Fordin S, et al. Web Service Choreography Interface (WSCCI) 1.0. Standards proposal by BEA Systems, Intalio, SAP, and Sun Microsystems, 2002
- Brown F M. Towards the automation of set theory and its logic. Artificial Intelligence, 1978, 10(3): 281~316
- Manthey R, Bry F. SATCHMO: A theorem prover implemented in prolog. In: Lusk E, Overbeek R, eds, In: Proc. 9th Conf. on Automated Deduction. New York: Springer-Verlag, 1988. 415~434
- Fujita H, Hasegawa R. A model generation theorem prover in KL1 using a ramifiedstack algorithm. In: Furukawa K, ed. Proc. 8th Intl. Conf. on Logic Programming. France: MIT Press, 1991. 535~548
- Bry F, Torge S. A deduction method complete for refutation and finite satisfiability. In: Dix J, Del Cerro L F and Furbach U, ed. Proc. 6th European Workshop on Logics in AI (JELIA). Springer-Verlag, 1998. 122~136
- Beckert B, Hahnle R. Analytic tableaux. In Bible W, Schmitt. Automated Deduction: A basis for Application. Kluwer, 1998. 11~41
- Bryant R E. Grapg-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, 1986, C35: 677~691
- Brace K S, Rudell R L, Bryant R E. Efficient implementation of a BDD package. In: Proc. 27th ACM/IEEE Design Automation Conference, Los Alamitos, 1990. 40~45
- Burch J R, Clarke E M, Dill D L. Sequential circuit verification using symbolic model checking. In: Proc. 27th Design Automation Conf. (DAC'90), 1990. 46~51
- 刘全, 孙吉贵. 非经典逻辑的语义 tableau 方法. 计算机科学, 2002, 29(5): 72~75

(上接第 4 页)

- Brown F M. Towards the automation of set theory and its logic. Artificial Intelligence, 1978, 10(3): 281~316
- Manthey R, Bry F. SATCHMO: A theorem prover implemented in prolog. In: Lusk E, Overbeek R, eds, In: Proc. 9th Conf. on Automated Deduction. New York: Springer-Verlag, 1988. 415~434
- Fujita H, Hasegawa R. A model generation theorem prover in KL1 using a ramifiedstack algorithm. In: Furukawa K, ed. Proc. 8th Intl. Conf. on Logic Programming. France: MIT Press, 1991. 535~548
- Bry F, Torge S. A deduction method complete for refutation and finite satisfiability. In: Dix J, Del Cerro L F and Furbach U, ed. Proc. 6th European Workshop on Logics in AI (JELIA). Springer-