基于知识Petri网和归结规则的推理方法

谭开成, 罗继亮†, 章宏彬, 林鑫杰, 叶剑虹

(华侨大学 信息科学与工程学院, 福建 厦门 361021; 福建省电机控制与系统优化调度工程技术研究中心, 福建 厦门 361021)

摘要: 本文提出了一种基于知识Petri网和归结规则的推理方法. 通过知识Petri网描述命题逻辑知识库, 将归结规则映射到知识Petri网上, 根据库所和变迁的连接关系, 定义了知识Petri网中的归结结构. 利用归结结构, 给出了基于知识Petri网的归结推理算法和扩展知识库的推理算法, 并利用Wumpus实例验证了推理算法. 该推理方法是可靠且完备的, 能够利用知识Petri网的网络结构降低计算复杂性.

关键词: Petri网; 归结规则; 推理

引用格式: 谭开成, 罗继亮, 章宏彬, 等. 基于知识Petri网和归结规则的推理方法. 控制理论与应用, 2023, 40(1): 172 – 177

DOI: 10.7641CTA.2021.10283

Reasoning method based on knowledge Petri nets and resolution rules

TAN Kai-cheng, LUO Ji-liang†, ZHANG Hong-bin, LIN Xin-jie, YE Jian-hong

(College of Information Science and Engineering, Huaqiao University, Xiamen Fujian 361021, China;
Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen Fujian 361021, China)

Abstract: In this paper, a reasoning method based on knowledge Petri nets and resolution rules is proposed. First, a knowledge base in the form of propositional logic is described as a knowledge Petri net. Second, a resolution structure is defined in a knowledge Petri net according to the input and output relationship of places and transitions. Third, a reasoning algorithm is presented to perform resolution operations by resolution structures. Further, a reasoning algorithm is given for an extended knowledge base. Finally, the Wumpus world problem is taken as an example to illustrate and verify the reasoning algorithm. It is proved that the reasoning method is sound and complete. The computational complexity involved in reasoning processes can be reduced by utilizing network structures of a knowledge Petri net.

Key words: Petri nets; resolution rules; reasoning

Citation: TAN Kaicheng, LUO Jiliang, ZHANG Hongbin, et al. Reasoning method based on knowledge Petri nets and resolution rules. *Control Theory & Applications*, 2023, 40(1): 172 – 177

1 引言

人工智能的核心主题是设计可以合理思考和行为的智能体(Agent). 然而, Agent通常工作在部分可观的环境中, 需要根据有限感知信息来估计未知的环境状态, 这要求Agent能够基于环境知识进行推理. 逻辑推理是人工智能中最持久的研究领域之一, 逻辑推理是证明在知识库Kb下语句q的永真性, 即确定$\mathrm{Kb}\to q$是否对每个可能状态都成立. 假设在Kb中存在n个逻辑变量, 那么需要检查2^n个状态. 因此, 推理算法的复杂度相对于逻辑变量的个数是指数级增长的.

分离和归结是逻辑推理中两个基础性规则, 分别采用霍恩(Horn)子句和合取范式的逻辑语句来表示的知识库. 文献[1]提出的霍恩子句实际上是一个if-then规则, 文献[2]将该规则称为Modus-Ponens规则. 文献[3–4]提出了归结规则, 文献[5]利用该规则设计了一种归结算法, 并证明了它的完备性, 但该算法的复杂度随着逻辑语句的数量而爆炸性增长. 文献[6]比较了运用Σ^*和Q^*两种搜索策略来做归结的效果, 但两种策略都没有为归结提供足够的指导. 文献[7]将蕴涵结构的子句集设计成Petri网模型, 通过变迁激发序列表示相应的推理过程. 文献[8]利用了Petri网并行和异步设计了有色Petri网, 表示蕴涵结构的Horn子句集, 利用并行激发变迁来提高推理效率. 文献[9]提出了一种运用于多值逻辑的归结方法, 并证明了该方法是可靠且完备的. 文献[10]将动态逻辑以Petri网进行描述, 提出了一种基于归结的证明方法, 并证明了该方法的可

收稿日期: 2021−04−07; 录用日期: 2021−09−23.

†通信作者. E-mail: jlluo@hqu.edu.cn; Tel.: +86 13110595996.

本文责任编委: 曾志刚.

国家自然科学基金项目(61973130), 福建省自然科学基金项目(2017J01117)资助.

Supported by the National Natural Science Foundation of China (61973130) and the National Natural Science Foundation of Fujian Province (2017J01117).

靠性和终止性, 但不能保证完备性. 文献[11]将归结规则运用于时序逻辑, 将逻辑语言转换为规范形式, 并给出在规范形式下归结推理方法. 针对似然推理, 文献[12]提出了一种模糊时序约束逻辑, 并结合有效归结规则来处理这个逻辑中的查询. 文献[13]在一阶逻辑中引入相似度, 提出了一种基于相似度的归结方法, 该方法是归结规则在模糊逻辑的扩展. 文献[14–17]通过模糊Petri网对基于知识的模糊系统进行建模, 设计在模糊Petri网上的推理引擎. 文献[18]对贝叶斯网络进行Petri网建模, 通过网络结构来表示概率的传播. 文献[19–21]通过设计监控库所, 对Petri网施加线性约束的方法来设计逻辑控制器. 文献[22]将布尔变量设计成环状的Petri网, 将知识设计成监控库所, 从而得到推理Petri网, 并开发了一种推理方法.

在前期工作[23]中, 笔者开发了一种基于知识Petri网的推理引擎, 该推理算法需遍历所有变迁来找到推理使能的变迁, 没有充分利用知识Petri网的拓扑结构, 且无法保证推理的完备性. 本文将知识Petri网与归结规则相结合, 定义了知识Petri网中的归结结构, 针对归结结构, 给出了在知识Petri网上对其进行归结操作的推理算法, 同时, 将完成推理的知识Petri网定义为完备知识Petri网, 在其基础上进行拓展知识库的推理, 推理效率提升更为显著. 与命题逻辑中的推理方法相比, 由于Petri网是图形化语言, 这使得本文提出的推理方法更加直观易懂. 此外, 知识Petri网的网络结构准确描述了命题符号之间的依赖关系, 而本文推理方法利用这种依赖关系, 实现了降低推理的计算复杂度. 该推理方法是可靠且完备的, 可运用于工程实践中.

不同于文献[8]中以Petri网建模推理过程, 用于判断某个命题是否成立, 本文以Petri网建模知识库, 得到的知识Petri网是受知识库约束的, 并将归结推理结果以归结库所的形式添加进知识Petri网, 使得完备知识Petri网包含了知识库的所有隐藏知识, 可以描述满足该知识库规则的每个合法状态. 此外, 不同于文献[9–13]将归结规则与非经典逻辑结合, 其知识表示形式为非经典逻辑语言, 本文研究的推理方法属于经典逻辑领域的确定性推理方法, 其知识表示形式为命题逻辑语言, 因而该推理方法适用于知识可用命题逻辑语言描述的系统, 如专家系统. 当然, 对于存在不确定性的系统, 本文方法不再适用.

2　基本概念

知识库Kb包括世界运作的一般知识和Agent对世界的感知语句. 推理算法是根据Kb导出蕴涵语句, 只导出蕴涵语句的推理算法被称为可靠的, 可以导出任一蕴涵语句的推理算法被称为完备的.

文字是原子语句或原子语句的否定式, 子句是文字的析取, 以子句的合取式表达的语句称为合取范式(conjunctive normal form, CNF). 任何一个语句逻辑上都等价于某些子句的合取式, 故都可以写成合取范式的形式. 归结规则是当两个子句中存在一对互补文字时, 可得到一个新的子句, 该新子句包含除了两个互补文字以外的原始子句中的所有文字. 归结规则与任何一个完备的搜索算法相结合可以得到完备的推理算法.

Petri网结构是一个四元组, 记为$N=(P,T,F,W)$, 其中: P是库所集; T是变迁集; $F\subseteq(P\times T)\cup(T\times P)$是变迁和库所组成的有序二元组的集合, 表示库所与变迁之间的有向弧; $W:F\to\mathbb{Z}^+$是有向弧与权重映射的正整数集. 如果$(x,y)\in F$, 则x是y的输入, y是x的输出. $x^{\cdot}$表示x的输出集, ${}^{\cdot}x$表示x的输入集. Petri网记作$G=(N,m_0)$, 其中m_0为初始标识, $m_0(p)$表示库所p在初始时的托肯数.

线性约束(η,b)要求Petri网在它规定的合法标识中演化, 其中η是$|P|$维行向量, b是整数, 要求每个标识m满足$\eta\cdot m\leqslant b$. 行向量$\bar{\eta}=\eta\cdot C$称为变迁权重向量, 其中C是Petri网的关联矩阵. 根据库所不变量监控 (supervision-based-on-place-invariant, SBPI) 方法, 设计监控库所p, 初始标识$m_0(p)=b-\eta\cdot m_0$; 当$\bar{\eta}(t)>0$, 添加弧(p,t), 弧上权值为$\bar{\eta}(t)$; 当$\bar{\eta}(t)<0$, 添加弧(t,p), 弧上权值为$|\bar{\eta}(t)|$.

对于一个命题符号α, 它的符号Petri网G_α[23]为: 库所集$P_\alpha=\{\tilde{\alpha},\alpha,\bar{\alpha}\}$, 变迁集$T_\alpha=\{t_\alpha,t_{\bar{\alpha}}\}$, 有向弧集$F_\alpha=\{(\tilde{\alpha},t_\alpha),(\tilde{\alpha},t_{\bar{\alpha}}),(t_\alpha,\alpha),(t_{\bar{\alpha}},\bar{\alpha})\}$, 弧权值均为1, 初始标识$m_0(\tilde{\alpha})=1$, $m_0(\alpha)=m_0(\bar{\alpha})=0$.

对于一个子句$c=l_1\vee l_2\vee\cdots\vee l_k$, 其中$l_i$表示文字, 该子句中文字个数为$k$, 其语义约束[23]为: $\bar{l}_1+\bar{l}_2+\cdots+\bar{l}_k\leqslant k-1$, 其中$\bar{l}_i$表示文字$l_i$的互补文字.

知识Petri网的设计方法[23]主要分为两个部分:

1) 对于知识库的符号集中每个命题符号, 设计符号Petri网;

2) 对于知识库里的每一条子句, 根据SBPI方法为其语义约束设计成监控库所.

3　知识Petri网归结算法

归结需要遍历查询子句之间是否存在互补符号, 这导致了指数级的排列组合的计算, 而知识Petri网的结构能够帮助简化计算.

定义 1　给定一个知识库Π和其知识Petri网G_Π, c是Π中任意一个子句, 根据c的语义约束设计的监控库所p称为c的子句库所, 记作p_c; 而c称为子句库所p_c的子句.

定义 2　给定一个知识库Π和其知识Petri网G_Π, Σ_Π是Π的符号集, 任意一个符号$\alpha\in\Sigma_\Pi$, 如果一个子句库所是t_α的输入, 那么该库所称为α的正关联库所, 由α的正关联库所组成的集合称为α的正关联库

所集; 如果一个子句库所是$t_{\bar{\alpha}}$的输入, 那么该库所称为α的负关联库所, 由α的负关联库所组成的集合称为α的负关联库所集.

定义 3 给定一个知识库Π和其知识Petri网G_Π, Σ_Π是Π的符号集, 任意一个符号$\alpha \in \Sigma_\Pi$, 若p_c是α的一个正关联库所, $p_{c'}$是α的一个负关联库所, 则无序二元组$(p_c, p_{c'})$是α的一个归结结构. 由α的归结结构组成的集合, 记为K_α.

定义 4 给定α的一个归结结构$(p_c, p_{c'})$, 它的归结库所$p_{c\wedge c'}$是按照下列步骤设计得到的一个库所:

1) 如果一个变迁是p_c或$p_{c'}$的输出变迁, 并且该变迁不是t_α或$t_{\bar{\alpha}}$, 则画一条从$p_{c\wedge c'}$到该变迁的弧, 弧上的权值是1;

2) $p_{c\wedge c'}$的初始标识是输出变迁的个数减1, 即$m_0(p_{c\wedge c'}) = |p^{\cdot}_{c\wedge c'}| - 1$.

引理 1 归结结构$(p_c, p_{c'})$设计的归结库所$p_{c\wedge c'}$和子句$c'' = c \wedge c'$的子句库所是相同的.

证 任意一个符号$\alpha \in \Sigma_\Pi$, 设$(p_c, p_{c'})$是α的归结结构. p_c和$p_{c'}$的子句表示为: $c = \bar{\alpha} \vee l_1 \vee \cdots \vee l_i$, $c' = \alpha \vee n_1 \vee \cdots \vee n_j$, 其中$l_i$和$n_j$表示文字. p_c和$p_{c'}$输出集分别表示为: $p^{\cdot}_c = \{t_\alpha, t_{\bar{l}_1}, \cdots, t_{\bar{l}_i}\}$, $p^{\cdot}_{c'} = \{t_\alpha, t_{\bar{n}_1}, \cdots, t_{\bar{l}_j}\}$. 令$p_{c\wedge c'}$是根据$(p_c, p_{c'})$设计的归结库所, 其输出集为$p^{\cdot}_{c\wedge c'} = \{t_{\bar{l}_1}, \cdots, t_{\bar{l}_i}\} \cup \{t_{\bar{n}_1}, \cdots, t_{\bar{n}_j}\}$, 初始标识为$m_0(p_{c\wedge c'}) = |p^{\cdot}_{c\wedge c'}| - 1$. 令$c''$为子句$c$与$c'$归结得到的新子句, $c'' = c \wedge c' = l_1 \vee \cdots \vee n_j$, 其子句库所为$p_{c''}$, 输出集$p^{\cdot}_{c''}$和初始标识$m_0(p_{c''})$与归结库所$p_{c\wedge c'}$对应相同. 故可得: 归结库所$p_{c\wedge c'}$是子句$c'' = c \wedge c'$的子句库所, $c'' = c \wedge c'$是$p_{c\wedge c'}$的子句.

证毕.

定义 5 给定一个知识Petri网和归结库所p, 如果存在$\alpha \in \Sigma_\Pi$, t_α和$t_{\bar{\alpha}}$都是p的输出变迁, 那么p称为永真归结库所.

引理 2 永真归结库所的子句是永真子句.

证 假设p为永真归结库所, 那么存在$\alpha \in \Sigma_\Pi$, t_α和$t_{\bar{\alpha}}$都是p的输出变迁, p的子句可表示为$c = \alpha \vee \bar{\alpha} \vee l_1 \vee \cdots \vee l_k$, 其中$l_k$为文字, 其语义线性约束为: $\bar{\alpha} + \alpha + \bar{l}_1 + \cdots + \bar{l}_k \leqslant k + 1$, 又因为$\bar{\alpha}$和$\alpha$是互补符号, 满足约束关系$\bar{\alpha} + \alpha \leqslant 1$, 故而上述语义线性约束可写成$\bar{l}_1 + \cdots + \bar{l}_k \leqslant k$, 又因为每一个文字的取值为0或1, 故该线性约束不等式是恒成立的, 这样的归结库所的子句是永真子句. 证毕.

定义 6 给定一个知识Petri网和归结库所p, 如果存在子句库所p_c的输出集$p^{\cdot}_c$是$p^{\cdot}$的子集, 那么将p称为冗余归结库所.

引理 3 冗余归结库所的子句的语义线性约束是冗余的.

证 假设p为冗余归结库所, 则存在子句库所p_c的输出集$p^{\cdot}_c$是$p^{\cdot}$的子集. 令输出集$p^{\cdot}_c = \{t_{l_1}, \cdots, t_{l_j}\}$, $p^{\cdot} = \{t_{l_1}, \cdots, t_{l_j}, \cdots, t_{l_{j+k}}\}$. p_c的子句为$\bar{l}_1 \vee \cdots \vee \bar{l}_j$, 语义线性约束为$l_1 + \cdots + l_j \leqslant j - 1$. p的子句为$\bar{l}_1 \vee \cdots \vee \bar{l}_j \cdots \vee \bar{l}_{j+k}$, 语义线性约束为$l_1 + \cdots + l_j + \cdots + l_{j+k} \leqslant j + k - 1$. 又因为每一个文字的取值为0或1, 可得不等式$l_{j+1} + \cdots + l_{j+k} \leqslant k$. 故由$p_c$的语义线性约束可以推导$p$的语义线性约束, p的语义线性约束是冗余的. 证毕.

由此, 本文给出了基于知识Petri网的归结算法, 如算法1所示.

算法 1 知识Petri网归结算法.

输入: 知识Petri网G_Π, 知识库Π的符号集Σ_Π.

输出: 储存推理结果的知识库Π^r, 算法结束的知识Petri网G^r_Π.

步骤 1 令Γ_r储存等待做归结的归结结构, Π^r是储存推理结果的知识库. 初始时, $\Gamma_r = \varnothing$, $\Pi^r = \varnothing$;

步骤 2 $\Gamma_r = \bigcup\limits_{\alpha\in\Sigma_\Pi} K_\alpha$, 将符号集$\Sigma_\Pi$中所有符号的归结结构放入$\Gamma_r$;

步骤 3 **while** $\Gamma_r \neq \varnothing$ **do**

步骤 4 从Γ_r中任意取出一个归结结构$(p_c, p_{c'})$, 并将其从Γ_r中删除, 由定义4设计归结库所$p_{c\wedge c'}$;

步骤 5 判断$p_{c\wedge c'}$是否为永真归结库所或冗余归结库所: 若是, 将$p_{c\wedge c'}$从G_Π中舍去; 否则, 将$p_{c\wedge c'}$对应子句$c \wedge c'$添加进Π^r_n, 并令$\Gamma_r = \Gamma_r \cup \{(p_{c'''}, p_{c''''}) | p_{c'''} = p_{c\wedge c'} \vee p_{c''''} = p_{c\wedge c'}\}$, 将归结库所$p_{c\wedge c'}$满足的归结结构放入$\Gamma_r$;

步骤 6 **end while**

步骤 7 返回推理结果Π^r和知识Petri网G^r_Π.

在算法1中, 给出了知识Petri网的归结推理. 其中, 在步骤2中, 得到等待做归结操作的归结结构集; 在步骤3–6中, 为归结结构设计归结库所; 在步骤3进行算法终止条件的判断, 若无等待做归结操作的归结结构, 则算法结束; 在步骤5中, 将保留的归结库所满足的归结结构放入归结结构集, 实现对归结结构的完备搜索.

在算法1中, 主要的计算是搜索归结结构, 它的计算复杂度是$\mathrm{O}(n \cdot k^2)$, 其中n表示符号集中符号的个数, k表示所有符号中最大的正关联库所集或负关联库所集的大小. 对单个符号α而言, 最坏情况是经过k^2次组合, 得到k^2个归结结构, 一共n个符号, 所以总的时间复杂度为$\mathrm{O}(n \cdot k^2)$. 然而, 传统的归结算法需要将知识库中的析取子句两两取出判断能否进行归结, 最坏的情况下, 知识库共有$2nk$个子句, 两两比对则需要判断

$$C^2_{2nk} = \frac{2nk \cdot (2nk - 1)}{2} = 2n^2k^2 - nk$$

次. 可见, 基于知识Petri网的归结算法能大大减少计

算复杂度.

定理 1　知识Petri网归结算法是可靠的.

证　由引理1, 归结库所的子句与子句归结运算的结果是相同的, 子句归结运算是可靠的, 那么知识Petri网归结算法中的归结操作也是可靠的.　　证毕.

定理 2　知识Petri网归结算法是完备的.

证　在算法1中, 初始时, 遍历符号集中的符号, 得到所有符号的归结结构. 之后每设计得到一个有效的归结库所, 便找到其满足的所有归结结构, 如此可以实现对归结结构的完备性搜索. 归结规则与完备的搜索算法相结合得到推理算法是完备的.　　证毕.

4　完备知识Petri网归结算法

算法1对知识Petri网进行完备性归结推理, 其结果仍然是一个知识Petri网. 如果获得了新的知识库Π', 可继续利用归结操作进行推理.

定义 7　若G_{Π}^{r}是知识Petri网G_{Π}经过算法1归结操作得到的知识Petri网, 称其为知识库Π的完备知识Petri网.

定义 8　若Π^{r}是知识Petri网G_{Π}经过算法1归结操作得到的知识库, 称其为知识库Π的归结知识库.

下面给出在完备知识Petri网上扩展新知识库的归结推理算法.

算法 2　完备知识Petri网归结算法.

输入: 知识库Π的完备知识Petri网G_{Π}^{r}, 知识库Π的符号集Σ_{Π}, 新知识库Π', 知识库Π'的符号集$\Sigma_{\Pi'}$.

输出: 储存推理结果的知识库Π_n^r, 算法结束的知识Petri网$G_{\Pi_n}^{r}$.

步骤 1　令$\Pi_n = \Pi \wedge \Pi'$, Γ_r储存等待做归结的归结结构, Π_n^r是储存推理结果的知识库. 初始时, $\Gamma_r = \varnothing$, $\Pi_n^r = \varnothing$, $\Sigma_{\Pi_n} = \Sigma_{\Pi} \cup \Sigma_{\Pi'}$ 表示Π_n的符号集;

步骤 2　**for all** $\alpha \in \Sigma_{\Pi} \wedge \alpha \notin \Sigma_{\Pi'}$ **do**

步骤 3　根据符号Petri网的设计方法, 在G_{Π}上设计α的符号Petri网;

步骤 4　**end for**

步骤 5　**for all** $c \in \Pi'$ **do**

步骤 6　根据c的语义线性约束, 按照SBPI方法设计c的子句库所p_c;

步骤 7　$\Gamma_r = \Gamma_r \cup \{(p_{c'}, p_{c''}) | p_{c'} = p_c \vee p_{c''} = p_c\}$, 将子句库所$p_c$满足的归结结构放入$\Gamma_r$;

步骤 8　**end for**

步骤 9　**while** $\Gamma_r \neq \varnothing$ **do**

步骤 10　从Γ_r中任意取出一个归结结构$(p_c, p_{c'})$, 并将其从Γ_r中删除, 由定义4设计归结库所$p_{c\wedge c'}$;

步骤 11　判断$p_{c\wedge c'}$是否为永真归结库所或冗余归结库所: 若是, 将$p_{c\wedge c'}$从G_{Π}中舍去; 否则, 将$p_{c\wedge c'}$对应子句$c \wedge c'$添加进Π_n^r, 并令$\Gamma_r = \Gamma_r \cup \{(p_{c'''}, p_{c''''}) | p_{c'''} = p_{c\wedge c'} \vee p_{c''''} = p_{c\wedge c'}\}$, 将归结库所$p_{c\wedge c'}$满足的归结结构放入$\Gamma_r$;

步骤 12　**end while**

步骤 13　返回推理结果Π_n^r和知识Petri网$G_{\Pi_n}^{r}$.

算法2给出了基于完备知识Petri网的推理过程. 与算法1不同的是, 步骤1得到了知识库Π和Π'的全部符号集; 步骤2–4对存在于Π'中而不在Π的符号设计符号Petri网; 步骤5–8对新知识库Π'中的每一条子句设计子句库所, 将该子句库所满足的归结结构放入Γ_r, 得到初始的等待做归结的归结结构集. 步骤9–13与算法1相同, 取归结结构进行归结, 并将归结库所满足的归结结构放入Γ_r中, 直至Γ_r中无归结结构.

定理 3　完备知识Petri网归结算法是可靠的.

定理 4　完备知识Petri网归结算法是完备的.

算法2也是可靠和完备的, 证明过程与算法1相似, 其证明从略.

算法2是在之前推理的基础上继续进行推理, 其输入为完备知识Petri网, 该知识Petri网的子句库所已全部完成了归结操作, 因而在扩展知识库时, 只需要遍历新设计的子句库所, 找到满足的归结结构, 而不用对所有子句库所进行归结操作. 相较于算法1, 算法2利用了之前推理的结果, 减少了对知识Petri网中归结结构的搜索, 在推理上具有更好的可扩展性. 但在本质上, 两个算法是相同的, 都能得到完备知识Petri网, 即实现对知识库中隐藏知识的完备性挖掘.

5　实验验证

Wumpus世界是人工智能中的一个经典案例. 为了更好的可读性将Wumpus世界进行简化, 如图1所示, 该环境有4个房间, 房间中可能存在怪兽. Agent在环境中探索, 利用传感器感知臭气, 避开怪兽.

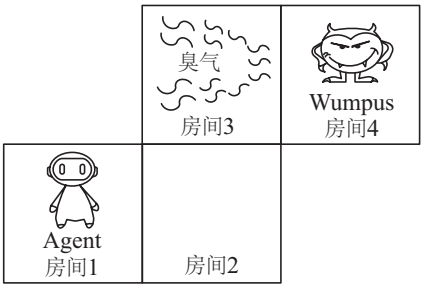

图 1　Wumpus世界

Fig. 1　The Wumpus world

Wumpus世界的知识规则为: 怪兽相邻的房间有臭气. 用符号w_i表示i号房间有怪兽, s_i表示i号房间有臭气, A_i表示i号房间相邻房间的集合, 则该Wumpus世界的知识规则形式化描述为

$$\Pi = \wedge_{i=1}^{4}(s_i \Leftrightarrow \vee_{j\in A_i} w_j). \tag{1}$$

式(1)的含义为: 当且仅当某个房间相邻的房间中有怪兽时, 该房间有臭气.

根据逻辑等价规则, 可将式(1)转换为合取范式, 表1给出了10条子句及其语义约束. 根据知识Petri网的设计方法[23], 设计表1中子句的子句库所, 得到Π的知识Petri网G_Π, 如图2中黑色部分所示.

表 1 各子句的语义约束和子句库所

Table 1 Semantic constraints and clause place of each clause

知识库	子句	语义约束	子句库所
Π	$c_1 = \bar{s}_1 \vee w_2$	$s_1 + \bar{w}_2 \leqslant 1$	p_{c_1}
	$c_2 = s_1 \vee \bar{w}_2$	$\bar{s}_1 + w_2 \leqslant 1$	p_{c_2}
	$c_3 = \bar{s}_2 \vee w_1 \vee w_3$	$s_2 + \bar{w}_1 + \bar{w}_3 \leqslant 2$	p_{c_3}
	$c_4 = s_2 \vee \bar{w}_1$	$\bar{s}_2 + w_1 \leqslant 1$	p_{c_4}
	$c_5 = s_2 \vee \bar{w}_3$	$\bar{s}_2 + w_3 \leqslant 1$	p_{c_5}
	$c_6 = \bar{s}_3 \vee w_2 \vee w_4$	$s_3 + \bar{w}_2 + \bar{w}_4 \leqslant 2$	p_{c_6}
	$c_7 = s_3 \vee \bar{w}_2$	$\bar{s}_3 + w_2 \leqslant 1$	p_{c_7}
	$c_8 = s_3 \vee \bar{w}_4$	$\bar{s}_3 + w_4 \leqslant 1$	p_{c_8}
	$c_9 = \bar{s}_4 \vee w_3$	$s_4 + \bar{w}_3 \leqslant 1$	p_{c_9}
	$c_{10} = s_4 \vee \bar{w}_3$	$\bar{s}_4 + w_3 \leqslant 1$	$p_{c_{10}}$
Π_1	$c_{11} = \bar{s}_1$	$s_1 \leqslant 0$	$p_{c_{11}}$
Π_2	$c_{12} = \bar{s}_2$	$s_2 \leqslant 0$	$p_{c_{12}}$
Π_3	$c_{13} = s_3$	$\bar{s}_3 \leqslant 0$	$p_{c_{13}}$

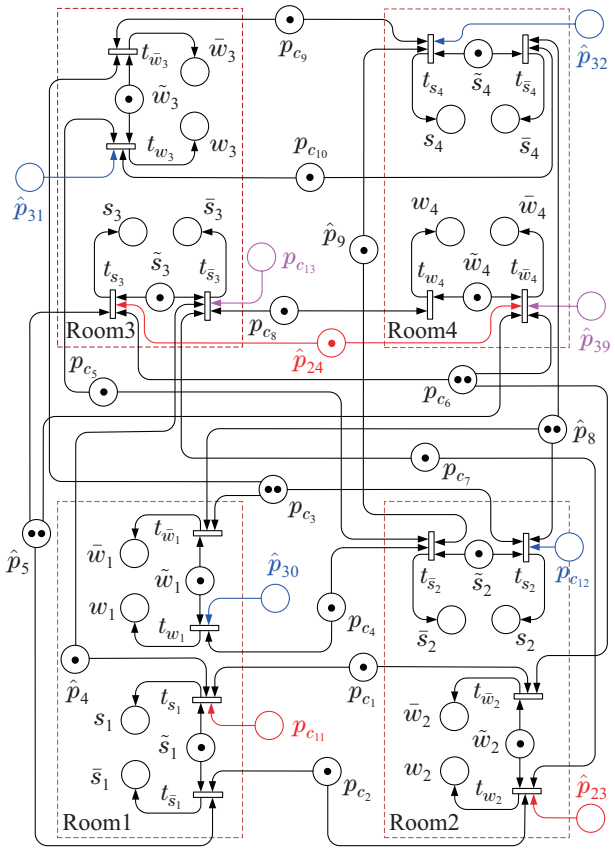

图 2 Wumpus世界的完备知识Petri网

Fig. 2 The complete knowledge Petri nets of the Wumpus world

根据算法1, 在知识Petri网G_Π上进行归结推理. 利用G_Π的网络结构搜索归结结构, 如(p_{c_1}, p_{c_2})为s_1的一个归结结构, 对其进行归结操作, 得到归结库所$\hat{p}_1$, 根据永真归结库所和冗余归结库所的定义, 判定$\hat{p}_1$ 为永真归结库所, 将其舍去. 又如归结结构(p_{c_1}, p_{c_7}), 归结操作得到归结库所$\hat{p}_4$, 该归结库所既不是永真归结库所, 也不是冗余归结库所, 保留在知识Petri网上, 其对应子句为$\bar{s}_1 \vee s_3$.

整个算法1共搜索了22个归结结构, 其中前10个归结结构是由G_Π中10个子句库所组成的二元组, 只有4个归结库所即不是永真归结库所, 也不是冗余归结库所, 被保留下来. 后12个归结结构是新设计的4个归结库所与其他子句库所组成的二元组, 其归结库所要么为永真归结库所, 要么为冗余归结库所, 皆被舍去. 此时, G_Π中无新的归结结构, 算法1结束, 返回知识库Π的完备知识Petri网G_Π^r, 如图2所示, 其中设计的4个归结库所为$\hat{p}_4, \hat{p}_5, \hat{p}_8$和$\hat{p}_9$, 归结知识库为$\Pi^r = (\bar{s}_1 \vee s_3) \wedge (s_1 \vee \bar{s}_3 \vee w_4) \wedge (\bar{s}_2 \vee s_4 \vee w_1) \wedge (\bar{s}_4 \vee s_2)$.

Agent探索Wumpus世界, 初始在Room1, 没有感知到臭气, 感知信息为$c_{11} = \bar{s}_1$, 将感知信息作为新的知识库$\Pi_1 = \bar{s}_1$. 根据算法2, 将算法1得到的Π的完备知识Petri网G_Π^r、新的知识库Π_1及相应的符号集作为输入, 在G_Π^r上进行扩展知识库Π_1的归结推理, 此时$\Pi_n = \Pi \wedge \Pi_1$. 算法2共搜索了7个归结结构, 这7个归结结构为$p_{c_{11}}$或新得到的归结库所与其他子句库所组成的二元组. 算法2结束, 返回完备知识Petri网, 如图2所示, 其中设计的归结库所为$\hat{p}_{23}$和$\hat{p}_{24}$, 归结知识库为$\Pi_n^r = \bar{w}_2 \wedge (\bar{s}_3 \vee w_4)$, 推理出Room2中没有怪兽. Agent下一个探索的房间为Room2.

Agent进入Room2, 根据算法2, 利用获得的感知信息$c_{12} = \bar{s}_2$继续在完备知识Petri网上进行归结推理. 算法2搜索了9个归结结构, 设计的归结库所中, 在完备知识Petri网上只保留了$\hat{p}_{30}, \hat{p}_{31}$和$\hat{p}_{32}$, 归结知识库为$\Pi_n^r = \bar{w}_1 \wedge \bar{w}_3 \wedge \bar{s}_4$. 推理出Room3中没有怪兽, Agent继续探索Room3. Agent在Room3的感知信息为$c_{13} = s_3$, 根据算法2搜索了4个归结结构, 设计的归结库所中, 仅保留了归结库所$\hat{p}_{39}$, 归结知识库为$\Pi_n^r = w_4$, 推理出Room4中有怪兽.

整个实验最终得到的完备知识Petri网如图2所示. 算法1和算法2共搜索了42个归结结构, 设计的归结库所中, 在完备知识Petri网上保留了10个归结库所, 其余32个归结库所为永真归结库所或冗余归结库所, 被舍去. 舍去永真归结库所和冗余归结库所, 使得保留的归结库所是真正起着约束作用的监控库所, 这样的监控库所是有限的, 且随着推理的进行, 这样的监控库所是随之减少的. 所以舍去永真归结库所和冗余归结库所, 保证了算法1和算法2具有终止性, 不会陷入局部性结论.

由实验结果可以得出, 推理得到的$\bar{w}_1, \bar{w}_2, \bar{w}_3, \bar{w}_4$等, 与图1中Wumpus世界的情况一致, 可见推理结果与事实相符; 同时, 最终的完备知识Petri网中共存在23个子句库所, 包括语义约束设计的13个子句库所和归结操作设计的10个归结库所, 传统的归结算法需要将知识库中子句两两组合判断能否归结, 则需搜索$C_{23}^2 = 253$个二元组, 而根据本文提出的推理算法, 利用知识Petri网的网络结构, 只需搜索42个归结结构, 可见本文推理方法具有更高的计算效率.

当待推理问题较为复杂, 解空间较大时, 本文推理方法仍是有效的, 且能有效降低计算复杂性. 这是因为归结规则本身就是可靠的, 知识Petri网的网络结构描述了命题符号之间的依赖关系, 利用知识Petri网的网络结构搜索归结结构, 就能减少不必要的遍历.

6　结论

本文提出了一种基于知识Petri网和归结规则的推理方法, 该方法是可靠且完备的. 通过定义永真归结库所和冗余归结库所, 保证了算法的可终止性. 此外, 利用知识Petri网的网络结构能够有效降低归结推理的计算复杂性.

未来将考虑Agent动作Petri网模型, 与知识Petri网相结合, 在一个统一的模型上求解知识推理与动作规划问题.

参考文献:

[1] HORN A. On sentences which are true of direct unions of algebras. *The Journal of Symbolic Logic*, 1951, 16(1): 14 – 21.

[2] MCGEE V. A counterexample to modus ponens. *The Journal of Philosophy*, 1985, 82(9): 462 – 471.

[3] DAVIS M, PUTNAM H. A computing procedure for quantification theory. *Journal of the ACM (JACM)*, 1960, 7(3): 201 – 215.

[4] DAVIS M, LOGEMANN G, LOVELAND D. A machine program for theorem-proving. *Communications of the ACM*, 1962, 5(7): 394 – 397.

[5] ROBINSON J. A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 1965, 12(1): 23 – 41.

[6] WILSON G, MINKER J. Resolution, refinements, and search strategies: A comparative study. *IEEE Computer Architecture Letters*, 1976, 25(8): 782 – 801.

[7] LAUTENBACH K. *On Logical and Linear Dependencies*. Sankt Augustin, Germany: GMD, 1985.

[8] MURATA T, ZHANG D. A high-level Petri net model for parallel interpretation of logic programs. *IEEE Transactions on Software Engineering*, 1988, 14(4): 481 – 493.

[9] KHANG T. A resolution method for linguistic many-valued logic. *Applied Mathematics & Information Sciences*, 2013, 7(3): 1193 – 1200.

[10] NALON C, LOPES B, DOWEK G, et al. A calculus for automatic verification of Petri nets based on resolution and dynamic logics. *Electronic Notes in Theoretical Computer Science*, 2015, 312: 125 – 141.

[11] FARAHANI H, MOSHIRI S. Temporal logic of common knowledge and its resolution-based proof method. *Journal of Intelligent & Fuzzy Systems*, 2018, 35(5): 5507 – 5522.

[12] VIEDMA M, MORALES R, SANCHEZ I. Fuzzy temporal constraint logic: A valid resolution principle. *Fuzzy Sets and Systems*, 2001, 117(2): 231 – 250.

[13] FONTANNA F, FORMATO F. A similarity-based resolution rule. *International Journal of Intelligent Systems*, 2002, 17(9): 853 – 872.

[14] LIU H, YOU J, YOU X, et al. Linguistic reasoning Petri nets for knowledge representation and reasoning. *IEEE Transactions on Systems Man & Cybernetics Systems*, 2016, 46(4): 499 – 511.

[15] LIU H, YOU J, LI Z, et al. Fuzzy Petri nets for knowledge representation and reasoning: A literature review. *Engineering Applications of Artificial Intelligence*, 2017, 60: 45 – 56.

[16] YUE W, GUI W, XIE Y. Experiential knowledge representation and reasoning based on linguistic Petri nets with application to aluminum electrolysis cell condition identification. *Information Sciences*, 2020, 529: 141 – 165.

[17] WANG L, DAI W, AI J, et al. Reliability evaluation for manufacturing system based on dynamic adaptive fuzzy reasoning Petri net. *IEEE Access*, 2020, 8: 167276 – 167287.

[18] LAUTENBACH K, PINL A. A Petri net representation of Bayesian message flows: Importance of Bayesian networks for biological applications. *Natural Computing*, 2011, 10(2): 683 – 709.

[19] LUO Jiliang, SHAO Hui, WU Weimin, et al. Synthesis of logical controllers and discrete-event systems supervisory control theory. *Control Theory & Applications*, 2018, 35(1): 86 – 91.
(罗继亮, 邵辉, 吴维敏, 等. 逻辑控制器设计与离散事件系统监控理论. 控制理论与应用, 2018, 35(1): 86 – 91.)

[20] LUO J, WU W, ZHOU M, et al. Structural controller for logical expression of linear constraints on Petri nets. *IEEE Transactions on Automatic Control*, 2019, 65(1): 397 – 403.

[21] LUO J, ZHANG Q, CHEN X, et al. Modeling and race detection of ladder diagrams via ordinary Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, 48(7): 1166 – 1176.

[22] LUO H, LUO J, LIN W. Reasoning via Petri nets. *The 8th International Conference on Intelligent Control and Information Processing*. Hangzhou, China: IEEE, 2017: 162 – 167.

[23] LUO J, TAN K, LUO H, et al. Inference approach based on Petri nets. *Information Sciences*, 2021, 547: 1008 – 1024.

作者简介:

谭开成　硕士研究生, 目前研究方向为离散事件系统和Petri网理论与应用等, E-mail: 19013082050@stu.hqu.edu.cn;

罗继亮　教授, 博士, 目前研究方向为离散事件系统和Petri网理论与应用等, E-mail: jlluo@hqu.edu.cn;

章宏彬　硕士研究生, 目前研究方向为离散事件系统和Petri网理论与应用等, E-mail: hopingzhang@foxmail.com;

林鑫杰　硕士研究生, 目前研究方向为离散事件系统和Petri网理论与应用等, E-mail: 1148268947@qq.com;

叶剑虹　副教授, 博士后, 目前研究方向为物联网云计算、大数据的数据挖掘、故障诊断与预判等, E-mail: leafever@163.com.