科技导报 2007 年 第 25 卷 第 7 期 (总第 229 期)

基于模糊 Petri 网的动态知识表示与推理方法

危胜军,胡昌振,孙明谦

北京理工大学计算机网络攻防对抗技术实验室,北京 100081

[摘要] 针对专家系统知识库中的知识具有模糊特性以及知识库需要频繁更新的特点,设计了一种基于模糊 Petri 网的动态知识表示与推理方法。该方法利用实际环境中的数据,通过学习来调整知识模型的权值、阈值和确信度等参数,从而实现知识库的动态实时更新。此方法同时结合了产生式知识表示方式和神经网络知识表示方式的优点:知识模型结构清晰,参数意义明确;具有学习和并行推理能力。仿真实验结果表明,经过训练调整后的知识模型具有更高的准确度。

[关键词] 知识表示;知识学习;模糊 Petri 网;模糊推理

[中图分类号] TP182 [文献标识码] A [文章编号] 1000-7857(2007)07-0013-05

A Method of Dynamic Knowledge Representation and Reasoning Based on Fuzzy Petri Nets

WEI Shengjun, HU Changzhen, SUN Mingqian

Laboratory of Computer Network Defense Technology, Beijing Institute of Technology, Beijing 100081, China

Abstract: A method of dynamic knowledge representation and reasoning based on Fuzzy Petri Nets is proposed for problems of knowledge of Fuzzy characteristics, where frequent updating of knowledge is required in an expert system. Using the data obtained from the actual environment, the parameters of the knowledge model such as weights, threshold and reliability can be adjusted by training, so the knowledge can be updated dynamically. Some useful factors of knowledge representation based on production rules and Neural Networks are integrated into the knowledge model in a clear way, with well defined meanings for parameters, and with a learning and parallel reasoning ability. The result of simulation shows that the precision of the knowledge model is improved after training.

Key Words: knowledge representation; knowledge learning; Fuzzy Petri nets; Fuzzy reasoning

CLC Number: TP182 **Document Code:** A **Article ID:** 1000-7857(2007)07-0013-05

专家系统知识库中的知识具有模糊特性,基于模糊 Petri 网(Fuzzy Petri Nets, FPN)的知识表示方法提供了一种对模糊知识有效表示和推理的工具[1]。FPN 知识模型具有并行推理能力且模型结构清晰、参数意义明确,但是模型不具有学习能力,不便于知识模型的动态实时更新,对知识模型的参数调整只能人工进行;基于神经网络的知识表示方法具有学习能力,能够自动调整知识模型的参数,但它是一种隐式的表示方法,模型中的参数没有明确的意义,这给知识模型的分析造成不便。文献[2-4]将神经网络的学习功能引入 FPN 中,实现了对 FPN 知识模型阈值的动态调整;文献[2-3]给出了较严格条件下 FPN 知识模型的权值学习算法;文献[4]给出了在没有阈值条件下的权值和确信度的学习算法。

上述各种动态知识表示和推理方法均具有较大的局限性。本文提出的基于 FPN 的知识表示与推理方法通过将知识模型进行分层处理后,利用反向传播学习算法实现了对知识模型的权值、阈值和确信度等参数

收稿日期:2007-01-23

作者简介:危胜军,男,北京市海淀区中关村南大街 5 号北京理工大学软件学院,讲师,研究方向为网络安全;E-mail: shj_w@163.com

的同时调整。

1 基本概念

定义 1 FPN 定义成一个十元组[2]

$$FPN=(\boldsymbol{P}, \boldsymbol{T}, \boldsymbol{D}, \boldsymbol{I}, \boldsymbol{O}, M, Th, \boldsymbol{W}, f, \beta)$$

其中，$\boldsymbol{P}=\{p_1,p_2,\cdots,p_n\}$为库所的有限集合；$\boldsymbol{T}=\{t_1,t_2,\cdots,t_m\}$为变迁的有限集合；$\boldsymbol{D}=\{d_1,d_2,\cdots,d_n\}$表示命题的有限集合；$|\boldsymbol{P}|=|\boldsymbol{D}|$，$\boldsymbol{P}\cap\boldsymbol{T}\cap\boldsymbol{D}=\varphi$；$\boldsymbol{I}$, $\boldsymbol{O}$: $\boldsymbol{T}\rightarrow\boldsymbol{P}$，是输入、输出函数，反映变迁到库所的映射；$M$: $\boldsymbol{P}\rightarrow[0,1]$，是标识函数，给库所 $p_i\in\boldsymbol{P}$ 分配一个标识 $M(p_i)$，为该库所对应模糊命题的真值；Th: $\boldsymbol{T}\rightarrow[0,1]$，对变迁 $t(t\in\boldsymbol{T})$ 定义一个阈值 $Th(t)=\lambda$；$\boldsymbol{W}=\{w_1,w_2,\cdots,w_r\}$，是规则的权值向量，反映规则中前提条件对结论的支持程度，$0\leqslant w_j\leqslant 1$，$\sum_{j=1}^{r} w_j=1$；$f$: $\boldsymbol{T}\rightarrow[0,1]$，给变迁赋予规则的确信度，$f(t)=\mu$；$\beta$: $\boldsymbol{P}\rightarrow\boldsymbol{D}$，将库所节点与命题一一对应。

定义 2 若存在变迁 t_1，库所 $p\in\boldsymbol{I}(t_1)$，但不存在变迁 t_2，$p\in\boldsymbol{O}(t_2)$，则称库所 p 为初始库所；若存在变迁 t_1，库所 $p\in\boldsymbol{O}(t_1)$，但不存在变迁 t_2，$p\in\boldsymbol{I}(t_2)$，则称库所 p 为终止库所。

2 基于 FPN 的动态知识表示与推理

2.1 基于 FPN 的知识表示

基于 FPN 的知识表示有“与”和“或”两种基本形式[1]（见图 1），任何复杂的知识模型都可以转换成这两种基本表示的连接组合。

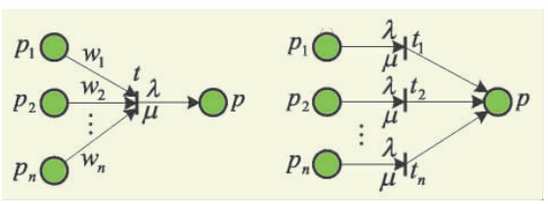
1）“与”形式对应的产生式规则为

if d_1 and d_2 and$\cdots$and d_n then d $(\mu,\lambda,w_1,w_2,\cdots,w_n)$；

2）“或”形式对应的产生式规则为

if d_1 or d_2 or$\cdots$or d_n then d (μ,λ)；

其中，$d_j(j=1,2,\cdots,n)$是规则的模糊前提命题；d 是结论命题；μ 是规则的确信度；λ 是规则的应用阈值；w_j $(j=1,2,\cdots,n)$为权值。知识的 FPN 模型中，μ，λ，w_j 等参数与产生式规则中的相关参数一一对应，都可显式直观地表现出来，具有确定的意义，因此可以直接基于模型对知识进行分析。



(a)“与”形式 (a) Model of "and"　　(b)“或”形式 (b) Model of "or"

图 1　FPN 模型的两种基本形式

Fig. 1　Two kinds of FPN basic model

2.2 基于 FPN 的模糊推理

基于 FPN 的模糊推理过程是知识的 FPN 模型从初始标识开始，所有满足条件的变迁按顺序并行激发，向输出库所产生新的标识，将 tolen 向目标转移，直到运行结束，结束时的终止库所的标识值即为推理结果。推理过程中变迁 t（设变迁 t 有 n 个输入库所）激发产生的新标识值由下式计算

$$z(t)=\begin{cases} f(t)\cdot\sum_{j=1}^{n}[M(p_j)\cdot w_j], \forall p_j\in\boldsymbol{I}(t), \sum_{j=1}^{n}[M(p_j)\cdot w_j]\geqslant Th(t) \\ 0, \qquad\qquad \forall p_j\in\boldsymbol{I}(t), \sum_{j=1}^{n}[M(p_j)\cdot w_j]<Th(t) \end{cases} \tag{1}$$

对于具有多个（假设有 m 个）输入变迁的库所 p，所有变迁同时激发后，该库所的标识值为这些变迁分别激发后产生的最大标识值。即

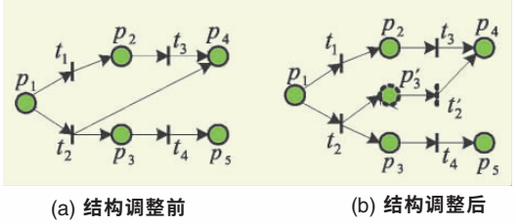
$$M(p)=\max\{z(t_1),z(t_2),\cdots,z(t_m)\} \tag{2}$$

2.3 参数学习算法

2.3.1 FPN 知识模型的分层处理

与神经网络的学习过程相同，FPN 知识模型的学习也是针对每一层逐层进行。由于 FPN 知识模型的网络结构可能不像神经网络层次分明，具有对称性，因此在一定的情况下需要对模型的结构进行调整，使得知识模型的层次结构更加清晰。在图 2(a)中，由于模糊推理时 p_4 的标识值由变迁 t_2 和 t_3 决定，因此这两个变迁在学习算法的反向误差传播时应同时受到影响，需要将这两个变迁划分到同一层中；同时 p_3 的标识值仅由变迁 t_2 确定，需要将 t_2 单独划分在一层中。

为了解决这一问题，需要对知识模型的结构适当调整，本文通过添加虚拟的库所和变迁来达到这一目的。所添加的虚拟库所对应的权值和确信度为 1，阈值为 0，这些值都是确定值，在学习过程中保持不变。这种处理既不影响模型的输入输出关系，也达到了分层的目的。图 2(b)显示了对图 2(a)分层处理后的网络结构，添加了虚拟库所 p_3' 和虚拟变迁 t_2'，使得模型的变迁分为清晰的两层，但是保持了原来的映射关系。这样处理后学习就可以针对这两层变迁顺利进行。



(a) 结构调整前 (a) Before structure adjustment　　(b) 结构调整后 (b) After stucture adjustment

图 2　知识的 FPN 模型结构调整示例

Fig. 2　An example of structure adjustment for FPN knowledge model

文献[5]给出了一种上述模型分层的计算机编程实现算法，描述如下。

1）初始化。建立所有初始库所的集合，设为 PM；$\forall p \in P-PM$，置 $M(p)=0$；设 $i=1$，i 是循环变量，其值表示模型的分层数。

2）建立变迁集合 $T_i=\{t \in T \mid \forall p \in I(t), p \in PM\}$。

将 T 中所有输入库所都属于 PM 的变迁 t 构成集合 T_i，T_i 也表示了同一层中所有变迁组成的集合。

3）若 $\exists t \in T_i$，$\exists p \in O(t)$，$\exists t_1 \in T-T_i, p \in O(t_1)$，则 $T_i=T_i-\{t\}$。

以上保证若一个库所是多个变迁的输出库所，则这些变迁被划分在同一层中。

4）置 $PM=PM \cup \{p \in O(t) \mid \forall t \in T_i\}$。

将 T_i 中所有变迁的输出库所都插入到集合 PM 中。

5）若 $T_i=\varphi$，则向模型中添加虚拟库所和虚拟变迁，返回步骤 2）执行。否则 $T=T-T_i$。

6）若 $T=\varphi$，则 $n=i$（n 为模型的最终分层数），结束。否则，$i=i+1$，返回步骤 2）循环执行。

2.3.2 模糊推理中连续函数的建立

式（1）、式（2）中标识值的计算和标识值的求最大运算都不是连续的，不能直接采用梯度下降法进行学习，因此需要采用连续函数模拟标识值的计算和标识值的求最大运算。参照神经网络的设计方法，将不连续的符号函数采用连续的 S 型函数替换。S 型函数的表达式如下

$$y(x)=1/(1+e^{-h(x-k)})=\begin{cases}1, \text{当 } x \geqslant k, \text{则 } e^{-h(x-k)} \cong 0\\ 0, \text{当 } x<k, \text{则 } e^{-h(x-k)} \to \infty\end{cases} \quad (3)$$

其中 $h(h \geqslant 200)$ 为常量。采用 S 型函数模拟标识值计算和标识值求最大运算如下。

1）标识值计算的连续函数模拟

设 $x=\sum_{j=1}^{n} M(p_j) \cdot w_j$，$k=Th(t)$，则变迁 t 激发产生的标识值为

$$z(t) \cong f(t) \cdot x \cdot y(x)=f(t) \cdot x/(1+e^{-h(x-Th(t))})=\begin{cases}f(t) \cdot x, & x \geqslant Th(t)\\ 0, & x<Th(t)\end{cases} \quad (4)$$

2）标识值求最大（max）运算连续函数模拟[5]

$$M(p)=\max\{z(t_1), z(t_2)\} \cong z(t_1)/(1+e^{-h(z(t_1)-z(t_2))})+z(t_2)/(1+e^{-h(z(t_2)-z(t_1))})=\begin{cases}z(t_1), & z(t_1) \geqslant z(t_2)\\ z(t_2), & z(t_1)<z(t_2)\end{cases} \quad (5)$$

2.3.3 BP 学习算法

推理中标识值的计算和标识值求最大运算采用连续函数模拟，因此可以直接采用 BP（Back Propagation）算法进行学习，学习过程针对知识模型中的每个变迁 t 进行，具体算法如下。

设攻击知识的 FPN 模型为 n 层，有 b 个终止库所 $p_1, p_2, \cdots, p_b$。用 r 个样本数据进行学习。设误差函数为

$$E=\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{b}\left[M_i(p_j)-M_i'(p_j)\right]^2 \quad (6)$$

式中，$M_i(p_j)$ 和 $M_i'(p_j)$ 分别表示终止库所 p_j 的第 i 个样本的实际标识值和期望标识值。

设 θ 为待学习的参数，θ 表示权值 w，阈值 λ 和确信度 μ；$t_i^{(n)}$ 是 FPN 模型第 n 层的一个变迁，若 $p_j^{(n)} \in O(t_i^{(n)})$，则显然 $p_j^{(n)}$ 是终止库所。用 BP 算法计算一阶梯度如下

$$\frac{dE}{d\theta_{ix}^{(n)}}=\frac{dE}{d(M^{(n)}(p_j))}\frac{d(M^{(n)}(p_j))}{d\theta_{ix}^{(n)}}=\delta^{(n)}\frac{d(M^{(n)}(p_j))}{d\theta_{ix}^{(n)}} \quad (7)$$

式中，$\delta^{(n)}=\frac{dE}{d(M^{(n)}(p_j))}$，$x=1,2,\cdots,m-1$。

若 $t_i^{(n-1)}$ 是 FPN 模型第 $n-1$ 层的一个变迁，设 $p_j^{(n-1)} \in O(t_i^{(n-1)})$，如果 $p_j^{(n-1)}$ 为终止库所，则一阶梯度的计算方法与第 n 层的计算方法相同；如果 $p_j^{(n-1)}$ 为非终止库所，则 $\exists t_i^{(n)} \in T_n$，且 $p_j^{(n-1)} \in I(t_i^{(n)})$，同时 $\exists p_l^{(n)} \in O(t_i^{(n)})$，$p_l^{(n)}$ 为终止库所，采用误差反向传播，$n-1$ 层的变迁的一阶梯度计算为

$$\frac{dE}{d\theta_{ix}^{(n-1)}}=\frac{dE}{d(M^{(n)}(p_l))}\frac{d(M^{(n)}(p_l))}{d(M^{(n-1)}(p_j))}\frac{d(M^{(n-1)}(p_j))}{d\theta_{ix}^{(n-1)}}=\delta^{(n-1)}\frac{d(M^{(n-1)}(p_j))}{d\theta_{ix}^{(n-1)}} \quad (8)$$

式中，$\delta^{(n-1)}=\delta^{(n)}\frac{d(M^{(n)}(p_l))}{d(M^{(n-1)}(p_j))}$，$x=1, 2, \cdots, m-1$。

依次对 $n-2, \cdots, 2, 1$ 层计算出 $dE/d\theta_{ix}^{(q)}$，则各参数根据下式进行调整

$$\theta_{ix}^{(q)}(k+1)=\theta_{ix}^{(q)}(k)-\eta dE/d\theta_{ix}^{(q)} \quad (9)$$

式中，$q=n, \cdots, 2, 1$，η 为学习速率，k 为学习的步次。针对权值 w 的调整，要求 $w_{im}^{(q)}(k+1)=1-\sum_{x=1}^{m-1} w_{ix}^{(q)}(k+1)$，保证变迁 $t_i^{(q)}$ 的输入弧上的权值总和为 1。

3 仿真实验

下面以某网络入侵检测系统中 FPN 攻击知识模型为实例，采用本文的学习算法对知识模型的参数进行调整来说明学习算法的可行性和有效性。

3.1 建立 FPN 知识模型

一般情况下，初始 FPN 知识模型根据专家的经验

建立。检测系统中建立的某个初始 FPN 攻击知识模型如图 3 所示。抛开攻击知识的具体内容，仅通过一个抽象的模型来说明学习过程。模型对应了 3 条产生式规则，分别用来检测 3 种网络攻击发生的可能性，相应的内容及参数初始值如表 1 所示。

命题 d_1 对应的真值表示攻击 1 发生的可能性；命题 d_2 对应的真值表示攻击 2 发生的可能性；命题 d_3 对应的真值表示攻击 3 发生的可能性。

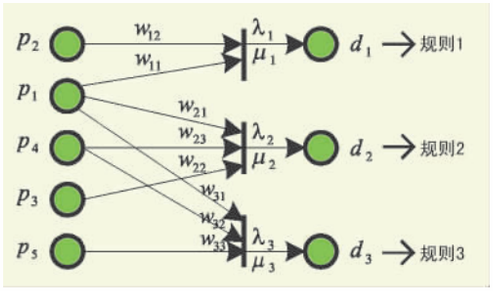


图 3 攻击知识的初始 FPN 模型

Fig. 3 Initial FPN model of attack knowledge

表 1 图 3 中 FPN 知识模型说明

Table 1 The explanation of FPN knowledge model in Fig. 3

规则	内容	参数初始值		
		权值	阈值	确信度
1	if p_1 and p_2 then d_1	w_{11}=0.2, w_{12}=0.8	λ_1=0.24	μ_1=0.96
2	if p_1 and p_3 and p_4 then d_2	w_{21}=0.1, w_{22}=0.5, w_{23}=0.4	λ_2=0.31	μ_2=0.96
3	if p_1 and p_4 and p_5 then d_3	w_{31}=0.1, w_{32}=0.3, w_{33}=0.6	λ_3=0.18	μ_3=0.97

3.2 模型参数学习与测试

表 2 中的数据是通过在实际网络环境中模拟这 3 种攻击产生的真实数据，数据标记了相应的攻击类型。表中数据约 500 条，将数据分为两部分，一部分用于知识模型训练，另一部分用于知识模型测试。

训练时标识计算函数中的 h=300，学习速率 η=0.04。模型在经过 300 次左右的训练后收敛，表 3 给出了模型在训练 320 次后的各参数值。

用测试数据对训练前和训练后的知识模型进行测试，测试结果如表 4 所示。表中数据为训练前后的知识模型在推理结束时各终止库所的标识值，即各种攻击发生的可能性。其值越大表示该攻击发生的可能性越大，表 5 是对表 4 求平均后的结果。

3.3 结果分析

对表 5 的数据进行分析得到如下结论。

1）当用攻击类型 1 的数据对模型进行测试，检测为攻击 1 的可能性略有提高（93%以上），误检为其他两种

表 2 在实际网络环境中模拟攻击采集到的数据

Table 2 The data collected in actual network environment by attack simulation

攻击类型	$M(P_1)$	$M(P_2)$	$M(P_3)$	$M(P_4)$	$M(P_5)$	$M(d_1)$	$M(d_2)$	$M(d_3)$
攻击 1	0.934	0.971	0	0.914	0.321	1	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0.941	0.965	0	0.916	0.492	1	0	0
攻击 2	0.726	0	0.878	0.963	0.124	0	1	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0.767	0	0.865	0.955	0.121	0	1	0
攻击 3	0.941	0	0	0.901	0.913	0	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0.938	0	0	0.892	0.911	0	0	1

表 3 知识模型训练后的各参数值

Table 3 Values of Parameters in knowledge model after training

规则	权值（w）			阈值（λ）	确信度（μ）
1	w_{11}=0.183 7	w_{12}=0.816 3		λ_1=0.230 1	μ_1=0.967 5
2	w_{21}=0.112 4	w_{22}=0.721 8	w_{23}=0.165 8	λ_2=0.228 7	μ_2=0.952 9
3	w_{31}=0.111 5	w_{32}=0.195 2	w_{33}=0.693 3	λ_3=0.191 5	μ_3=0.963 8

表 4 知识模型训练前后的输出结果比较

Table 4 Output comparison between fore-learning and aft-learning

攻击类型	攻击发生的可能性								
	训练前			训练后			理想值		
	$M(d_1)$	$M(d_2)$	$M(d_3)$	$M(d_1)$	$M(d_2)$	$M(d_3)$	$M(d_1)$	$M(d_2)$	$M(d_3)$
攻击 1	0.925 1	0.440 6	0.517 6	0.932 9	0.244 4	0.246 7	1	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0.916 5	0.437 6	0.505 3	0.945 8	0.238 9	0.231 0	1	0	0
攻击 2	0	0.860 9	0.422 8	0	0.833 8	0.142 1	0	1	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	0.857 6	0.415 6	0	0.846 5	0.137 6	0	1	0
攻击 3	0	0.432 6	0.880 8	0	0.115 6	0.877 3	0	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	0.427 5	0.875 2	0	0.112 3	0.880 1	0	0	1

表 5 模型训练前后输出结果的平均值比较

Table 5 The comparison between average output of model before learning and after learning

攻击类型	平均结果								
	训练前			训练后			理想值		
	$M(d_1)$	$M(d_2)$	$M(d_3)$	$M(d_1)$	$M(d_2)$	$M(d_3)$	$M(d_1)$	$M(d_2)$	$M(d_3)$
攻击 1	92.3%	43.7%	50.6%	93.9%	23.1%	22.7%	1	0	0
攻击 2	0%	86.4%	42.1%	0%	85.8%	13.6%	0	1	0
攻击 3	0%	42.4%	87.6%	0%	11.3%	87.9%	0	0	1

攻击的可能性大幅度降低。

2)当用攻击类型 2 的数据对模型进行测试,检测为攻击 2 的可能性基本保持不变(85%以上),误检为攻击1 的可能性为 0, 误检为攻击 3 的可能性大幅度降低。

3)当用攻击类型 3 的数据对模型进行测试,检测为攻击 3 的可能性略有提高(87%以上),误检为攻击 1 的可能性为 0,误检为攻击 2 的可能性大幅度降低。

以上结论说明了模型在学习后, 其准确度大幅度提高, 但是和理想情况相比还需要进一步改善知识模型。

4 结论

专家系统知识库中的知识依赖于专家的经验建立,因此知识具有模糊特性,并且由于专家经验知识的局限,需要对建立后的知识模型进行调整。如果能够从实际环境中获取数据, 利用学习算法从这些数据中学习知识,是一种最直接和有效的方法。本文提出的方法能够满足这一要求,当实际环境发生变化后,利用学习算法就可以对相应的参数进行动态实时调整。相对于以往的许多动态知识表示方法, 本文的方法具有更加宽松的条件,应用更加灵活,实用性更强。

参考文献(References)

[1] CHEN S M, KE J S, CHANG J F. Knowledge representation using fuzzy Petri nets [J]. IEEE Trans Knowledge and Data Engineering, 1990, 2(3): 311–319.

[2] LI Xiaoou, YU Wen, LARA–ROSANO F. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework [J]. IEEE Transactions on Systems, Man, and Cybernetic–Part C: Applications and Reviews, 2000, 30(4): 442–449.

[3] LI Xiaoou, LARA–ROSANO F. Adaptive fuzzy Petri nets for dynamic knowledge representation and inference [J]. Expert Systems with Applications, 2000, 19 (3): 235–241.

[4] TSANG E C C, YEUNG D S, LEE J W T. Learning capability in fuzzy Petri nets [J]. Proceeding of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, 1999: 355–360.

[5] 鲍培明. 基于 BP 网络的模糊 Petri 网的学习能力[J]. 计算机学报, 2004, 27(5): 695–702.
BAO Peiming. Learning capability in fuzzy Petri nets based on BP net[J]. Chinese Journal of Computers, 2004, 27(5): 695–702.

(责任编辑 代 丽)

·科技智力游戏·

好玩的数学——猜数

丙把数 a 和 b 分配给逻辑推理能力很强的甲、乙两人,甲、乙两人都知道自己的数,不知道对方的数,但两人都被告知对方的数是与自己手中数相差 1 的正整数。现在,有一台报时器每隔 10 秒钟发出一次嘟嘟声。如果甲、乙中有谁知道了对方的数,他就必须在某次嘟嘟声响结束后立即宣布这一点,于是游戏结束。

游戏开始了。在报时器第 15 次声响后,甲报出了乙的数。请问:丙分配给甲乙的数各是多少? 为什么?

2007 年第 5 期“谁的脸上沾有泥巴”答案

我们分析一下游戏开始后发生的情况。

如果孩子中只有 1 个脸上有泥巴,除了这个有泥巴的孩子不知道外,其他人都能够看到并且知道。根据老师的提示,所有孩子都知道至少有一个孩子脸上有泥巴。于是,脸上有泥巴的这个孩子在看到其他孩子脸上都干净后, 会推出只有自己脸上有泥巴。所以,他会在第 1 次报时器响后马上举手。由此所有孩子都会明白:如果在第 1 次报时器响后没有孩子举手,则意味着脸上有泥巴的孩子不止一个。

如果孩子中有 2 个脸上有泥巴, 那么对这 2 个孩子来说,除自己之外,他们都只看到 1 个孩子脸上有泥巴。于是,根据上面的“脸上有泥巴的孩子不止一个”,这 2 个孩子都会推出自己脸上有泥巴。所以,他们会在第 2 次报时器响后马上举手。由此所有孩子都会明白:如果在第 2 次报时器响后没有孩子举手,则意味着脸上有泥巴的孩子不止 2 个。

如果孩子中有 3 个脸上有泥巴, 那么对这 3 个孩子来说,除自己之外,他们都只看到 2 个孩子脸上有泥巴。于是,根据上面的“脸上有泥巴的孩子不止两个”,这 3 个孩子都会推出自己脸上有泥巴。所以,他们会在第 3 次报时器响后马上举手。由此所有孩子都会明白:如果在第 3 次报时器响后没有孩子举手,则意味着脸上有泥巴的孩子不止 3 个。

依此类推下去。对这个著名的“脸上沾有泥巴的孩子”问题可得出一般性的结论:假定有 n 个孩子脸上有泥巴,那么在第 $1\sim(n-1)$ 次报时器响后都没有孩子举手,而在第 n 次报时器响后这 n 个孩子会同时举手。

因此,如果第 12 次声响后,所有脏脸孩子都举了手,那么可以确定:教室里脸上沾有泥巴的孩子一共有 12 个。

(供稿 韩雪涛 责任编辑 黄永明)