计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200100079

LFNDIT:从不确定状态变换学习布尔网络

黄　羿[1,2]　孔世明[2]　王以松[1]　张明义[3]　马新强[1,2]
1 贵州大学计算机科学与技术学院　贵阳 550025
2 重庆文理学院人工智能学院　重庆 402160
3 贵州科学院　贵阳 550001
(cqhy@21cn.com)

摘　要　布尔网络是一种重要的基因调控数学模型,从布尔网络的状态变换推断其结构以发现基因之间的调控关系是布尔网络研究中长期关注的重要问题。已有的归纳逻辑程序算法不能从布尔网络的不确定(解释)状态变换学习推断其网络结构。为此,文中提出了非确定解释转换学习(Learning From Non-deterministic interpretation Transitions,LFNDIT)算法从布尔网络异步更新语义下的解释变换学习其网络结构。首先将异步更新语义下的不确定解释变换集转换成确定解释变换集,然后利用Inoue等提出的从1步解释转换学习(Learning From 1-step state transition,LF1T)算法计算其对应的正规逻辑程序(布尔网络)。该算法的完备性得到了证明,初步的实验结果表明,该方法能有效地从不确定状态变换计算布尔网络的结构,从而为发现布尔网络的结构提供了新的思路。

关键词:归纳逻辑程序;布尔网络;异步布尔网络;正规逻辑程序;LFNDIT算法
中图法分类号　TP391

LFNDIT:Learning Boolean Networks from Nondeterministic Interpretation Transitions

HUANG Yi[1,2],KONG Shi-ming[2],WANG Yi-song[1],ZHANG Ming-yi[3] and MA Xin-qiang[1,2]
1 College of Computer Science and Technology,GuizhouUniversity,Guiyang 550025,China
2 College of Artificial Intelligence,Chongqing University of Arts and Sciences,Chongqing 402160,China
3 Guizhou Academy of Sciences,Guiyang 550001,China

Abstract　Boolean network is an important mathematical model for gene regulation. It is an important issue that inferring structure from the interpretation transitions of Boolean network to discover the regulatory relationship between genes. Thus,researchers in the field of Boolean networks have been paying attention for a long time. Existing inductive logic program algorithms cannot infer the network structure from a set of nondeterministic state transitions. To this end,LFNDIT is proposed to learn the structure from state transitions under the asynchronous update semantics of Boolean network. First it translates a set of uncertain state transitions into the set of certain state transitions,and then uses the LF1T learning algorithm proposed by Inoue et al to calculate the corresponding normal logic program (Boolean network). The completeness of LFNDIT is proofed. The preliminary experimental results show that the algorithm can effectively calculate the Boolean network structure from the uncertain state transitions,thus it provides a new idea for discovering Boolean network structure.

Keywords　Inductive logical programming,Boolean network,ABN,Normal logic programming,LFNDIT algorithm

1　引言

布尔网络(Boolean Network,BN)[1-2]是由布尔变量集合及变量所对应的布尔函数组成的离散动态系统。该模型可定性地描述基因调控网络节点间的相互作用关系,因此在生物信息学领域得到了广泛的应用。尽管布尔网络是一个简单的模型,但表现出了复杂的行为,学术界对其进行了广泛的研究[3-9]。布尔网络推断问题是该领域研究中的基础问题之一,

到稿日期:2020-01-11　返修日期:2020-04-07　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61976065);重庆市高技术产业重大产业技术研发项目(2018148208);重庆市技术创新与应用发展重点项目(cstc2019jscx-fxydX0094);重庆英才计划创新创业示范团队(CQYC201903167)
The work was supported by the Nature Science Foundation of China(61976065),Key Industrial Technology Development Project of Chongqing Development and Reform Commission,China(2018148208),Key Technological Innovation and Application Development Project of Chongqing,China (cstc2019jscx-fxydX0094) and Innovation and Entrepreneurship Demonstration Team of Yingcai Program of Chongqing, China (CQYC201903167).
通信作者:王以松(yswang@gzu.edu.cn)

引起了生物信息学及计算机等领域许多学者的兴趣[10-18]。

布尔网络状态转换集可分为确定状态转换集和非确定状态转换集。确定状态转换集是指集合中每一状态的后继状态唯一；非确定状态转换集合意味着该集合中存在至少一个状态，而其后继状态不唯一。不考虑噪声等外界干扰因素的情形下，布尔网络状态转换集合为确定还是非确定是由更新模式决定的。

布尔网络的更新模式分为同步和异步两种。同步更新指每一时刻网络中所有节点的状态根据其指定的布尔函数进行同时更新。同步更新模式下，一个状态（t 时刻网络中所有节点的值）的后继状态（$t+1$ 时刻网络中所有节点的值）是唯一的。但由于生物系统中存在不同的时间刻度，考虑到这一实际情形，异步更新方式被提出。所谓异步更新指网络中所有节点的值不再根据其对应的布尔函数同时更新，而是根据不同的时间刻度进行更新。这种更新方式直接导致了一个状态的后继状态不唯一，即一个状态有多个可能的后继状态。异步布尔网络又可分为确定异步布尔网络（DARBNs）、一般异步布尔网络（GARBNs）和确定一般异步布尔网络（DGRBNs）[19]。

Inoue 的研究[20-21]表明，布尔网络可表示为正规逻辑程序（NLP），并且网络的吸引子可由正规逻辑程序的支承语义模型表示。上述推断问题被描述为从给定状态转换集合学习正规逻辑程序，从而可利用归纳逻辑程序（ILP）[22-23]的学习范例。一个典型的归纳学习任务可描述为：给定以逻辑程序表示的观察 O 及背景知识 BG，当 $BG \not\models O$ 时，需要从 O 和 BG 中学习出新的假设 H，使得 $BG \cup H \models O$。Inoue 等[24]提出了从 1 步解释转换学习（LF1T）自底向上的学习框架，该框架用于从确定的状态转换集合学习同步布尔网络。由于此算法存在根据同一状态转换集合的不同输入顺序学到的 NLP 程序有所不同且学到的程序存在冗余规则的问题，Ribeiro 等[25]提出了 LF1T 自顶向下的算法。该算法从确定的状态转换集合学习，根据完全状态转换（即系统所有可能的状态转换）集合学到的 NLP 程序不受该状态转换集合不同输入顺序的影响。

但不论 LF1T 是自顶向下的算法还是自底向上的算法，都只能从确定的转换集合中学习，因此不能用于异步布尔网络的学习。在归纳逻辑程序研究领域，Huang 等[26]提出了从非确定的解释转换集合学习以析取逻辑程序表示的异步布尔网络的算法（LFDT）。但 LFDT 算法要求不确定的状态转换集合 E 要满足集合包含意义下的极小关系，即 $\exists\langle I,J\rangle,\langle I,J'\rangle \in E$ 使得 $J \subseteq J'$ 或者 $J' \subseteq J$。Ribeiro 等的研究[27]表明可用多值逻辑程序表示异步布尔网络的状态转换规则，并提出 GULA 算法从给定的不确定转换集学习多值逻辑程序。

不论是以析取逻辑程序或是多值逻辑程序表示的异步布尔网络状态转换规则，都不能反映该异步布尔网络对应的同步更新模式下的状态转换，因此需要一种新的方法根据异步布尔网络的状态转换集学习其对应的同步布尔网络。

异步布尔网络与对应的同步布尔网络只是更新方式不同，但布尔函数没有发生变化。本文提出了一种将不确定状态转换集变为确定状态转换集合的方法，并讨论了其相关性质。根据该方法对 LF1T 自顶向下算法进行了修改，讨论了新算法的可靠性和完备性。本文提出了表示 3 种不同更新方式的语义并讨论了 3 种语义映射下状态转换集的关系。

2　背景知识

2.1　布尔网络

一个布尔网络可表示为 $BN=(V,F)$，其中 $V=\{v_1,\cdots,v_n\}$ 是节点的有限集合且 $F=\{f_1,\cdots,f_n\}$ 是节点对应的布尔函数集合。用 v_i^t 表示 t 时刻 v_i 的状态，节点的状态取值为 1 或 0。向量 $s(t)=(v_1^t,\cdots,v_n^t)$ 被称为 t 时刻 BN 的全局状态或简称 t 时刻 BN 的状态。n 个节点的布尔网络每一时刻有 2^n 个可能的状态。$t+1$ 时刻节点 v_i 的状态由调节函数 $v_i^{t+1}=f_i(v_{i_1}^t,\cdots,v_{i_k}^t)$ 决定，其中 $v_{i_1}^t,\cdots,v_{i_k}^t$ 是 v_k 的输入节点集。BN 的轨迹指通过一系列状态转换得到的状态序列。吸引子是不动点或一个周期震荡，可表示为状态集合 $\{s_0,s_1,\cdots,s_{p-1}\}$，其中 $s_{i+1}=F(s_i)(i=0,\cdots,p-2)$ 且 $s_0=F(s_{p-1})$。

布尔网络的更新有 3 种模式，即同步更新、异步更新和一般异步更新。根据更新方式的不同可将布尔网络分为同步布尔网络（SBN）、异步布尔网络（ABN）和一般布尔网络（GBN）。同步布尔网络中所有节点的状态同时进行更新。这种更新方式下，状态转换是唯一的。异步布尔网络中每时刻选择网络中的一个节点进行更新，一般布尔网络中每时刻随机选择 0 到 n 个节点进行更新。

2.2　正规逻辑程序

本节主要介绍正规逻辑程序的基本概念。这里考虑一阶语言并用 B 表示 *Herbrand* 基，原子 a 和原子的否定 $\neg a$ 分别被称为正文字和负文字；正文字和负文字统称为文字。一个正规逻辑程序是如下形式规则的集合：

$$a \leftarrow a_1 \wedge \cdots \wedge a_m \wedge \neg a_{m+1} \cdots \neg a_n \tag{1}$$

其中，a 和 $a_i(0 \leqslant m \leqslant n)$ 是原子。任何形如式(1)的规则 r，a 被称为 r 的头，记为 $hd(r)$，箭头右侧文字的合取称为 r 的体，记为 $bd(r)=\{a_1,\cdots,a_m,\neg a_{m+1},\cdots,\neg a_n\}$。规则的体 $bd(r)$ 进一步被分为正体与负体；正体是规则体中原子的集合，记为 $bd^+(r)=\{a_1,\cdots,a_m\}$；负体是规则体中负文字的集合，记为 $bd^-(r)=\{a_{m+1},\cdots,a_n\}$。一个逻辑程序 P 所有基实例的集合记为 $ground(P)$。令 r_1,r_2 是两条规则，称 r_1 包孕 r_2 如果 $hd(r_1)=hd(r_2)$ 且 $bd(r_1) \subseteq bd(r_2)$；也称 r_1 比 r_2 更一般化，r_2 比 r_1 更特殊化。

给定两条规则 r_1 和 r_2，且 r_1 包孕 r_2。r_1 基于 r_2 的最小特殊化记为 $lst(r_1,r_2)$：

$$lst(r_1,r_2)=\{hd(r_1) \leftarrow bd(r_1) \cup \{\neg l\} \mid l \in bd(r_2)-bd(r_1)\} \tag{2}$$

一个程序 P 的 *Herbrand* 基记为 B_P 是 P 中所有基原子的集合。P 的一个 *Herbrand* 解释 I 是 B_P 的子集。一个解释 I 是 P 的模型，如果 P 中的每条规则 r，$bd^+(r) \subseteq I$ 且 $bd^-(r) \cap I=\emptyset$ 蕴含 $hd(r) \in I$ 成立。给定程序 P 和解释 I，直接后继算子是映射 $T_P:2^B \to 2^B$：

$$T_P(I)=\{hd(r) \mid r \in ground(P), bd^+(r) \subseteq I, bd^-(r) \cap I=\emptyset\} \tag{3}$$

正规逻辑程序 P 的支承类 S 是 *Herbrand* 解释的非空集

合并满足 $S=\{T_P(I)\mid I\in S\}$。程序 P 的支承类 S 是严格的，即 S 的任何子集都不是 P 的支承类。

为了简便起见，在没有说明的情形下，总是假设逻辑程序是基程序。由于知道一个布尔网络的节点集，该网络的状态能用解释表示，因此后文将布尔网络的状态称为解释，状态转换称为解释转换。

2.3 布尔网络与正规逻辑程序

给定布尔网络 $N=(V,F)$，其中 $V=\{v_1,\cdots,v_n\}$ 且 $F=\{f_1,\cdots,f_n\}$。可将 F 中的每个布尔表达式转换为多条正规逻辑规则，从而用正规逻辑程序来表示布尔网络。为了实现这个转变，首先将 f_i 变为析取范式。布尔表达式总是能变成析取范式，因此假定每个 f_i 已经是析取范式的形式。

$$v_i(t)=\bigvee_{j=1}^{l_i}B_{i,j}(t),B_{i,j}(t)=\bigwedge_{k=1}^{m_j}v_{i,j,k}(t)\wedge\bigwedge_{k=m_j+1}^{n_j}\neg v_{i,j,k}(t) \tag{4}$$

其中，$v_{i,j,k}\in V$ 且 $n_j\geqslant m_j\geqslant 0$，$j=1,\cdots,l_i$。注意，节点 v_i，l_i 和 j 可以是 0。这种情形下 v_i 被称为常量节点。令 $V_c\subseteq V$ 是 V 中常量节点的集合。

给定布尔网络 N，可将 N 转变为正规逻辑程序 $\pi(N)$，其中：

$$\pi(N)=\{v_i\leftarrow B_{i,j}\mid v_i\in(V-V_c),1\leqslant j\leqslant l_i\} \tag{5}$$

其中，$B_{i,j}=\bigwedge_{k=1}^{m_j}v_{i,j,k}$，即去掉了式(4)中表达式 $B_{i,j}(t)$ 中每个文字的时间参数。t 时刻网络的状态 $s(t)=(v_1(t),\cdots,v_n(t))$ 可用解释 $I^t\subseteq V$ 表示，其中 $I^t=\{v_i\in V\mid v_i(t)=1\}$。

此时，$I^{t+1}=T_{\pi(N)}(I^t)$ 成立。一个布尔网络 N 转变为正规逻辑程序 P 后，该网络的吸引子可由正规逻辑程序的严格支承类表示[22]。

3 布尔网络的语义

首先将 T_P 语义[23]扩展至异步及一般操作，在此基础上根据同步布尔网络及异步和一般布尔网络更新的模式，给出3种不同模式下的更新语义。

定义 1(同步更新模式的 T_P 算子) 给定原子集 A 及正规逻辑程序 P，令解释 $I\in 2^A$。定义同步更新变换下的直接后继算子 $T_P^s:2^A\to 2^A$ 如下：

$$T_P^s(I)=\{hd(r)\mid I\models bd(r),r\in P\} \tag{6}$$

定义 2(异步更新模式的 T_P 算子) 给定原子集 A 及正规逻辑程序 P。令解释 $I\in 2^A$，$A_1=\{\{q\}\mid q\in A\}$。定义异步更新变换下的直接后继算子 $T_P^{asy}:2^A\times A_1\to 2^A$ 如下：

$$T_P^{asy}(I,V_c)=(I-V_c)\cup(V_c\cap T_P^s(I)) \tag{7}$$

定义 3(一般更新模式的 T_P 算子) 给定原子集 A 及正规逻辑程序 P。令解释 $I\in 2^A$，$V_c\subseteq A$。定义一般更新变换下的直接后继算子 $T_P^g:2^A\times 2^A\to 2^A$ 如下：

$$T_P^g(I,V_c)=(I-V_c)\cup(V_c\cap T_P^s(I)) \tag{8}$$

根据上述定义，定义 2 是定义 3 的特例。而定义 1 的 $T_P^s(I)$ 算子可表示为 $T_P^g(I,A)$。

推论 1 令 A 是原子集，P 是正规逻辑程序。

1) $T_P^s(I)=T_P^g(I,A)$；

2) $I=T_P^g(I,\varnothing)$。

证明：

1) $T_P^g(I,A)=(I-A)\cup(A\cap T_P^s(I))$
$=\varnothing\cup(A\cap T_P^s(I))=T_P^s(I)$

2) $T_P^g(I,\varnothing)=(I-\varnothing)\cup(\varnothing\cap T_P^s(I))$
$=I\cup(\varnothing\cap T_P^s(I))$
$=I$

通过上面的定义，给定当前解释及选择更新的节点集合，得到当前解释在不同更新方式下的直接后继状态，但在异步和一般更新方式下，当前解释的直接后继解释不唯一。下文的定义将得到给定解释在这两种更新模式下所有可能的后继解释。

定义 4 给定原子集 A 及正规逻辑程序 P。异步更新模式下，一个解释 I 的所有可能后继解释算子 $T_P^{asy}:2^A\to 2^{2^A}$ 如下：

$$T_P^{asy}(I)=\{T_P^{asy}(I,\{q\})\mid\forall q\in A\} \tag{9}$$

定义 5 给定原子集 A 及正规逻辑程序 P。定义一般更新模式下，一个解释 I 的所有可能后继解释算子 $T_P^g:2^A\to 2^{2^A}$ 如下：

$$T_P^g(I)=\{T_P^g(I,V_c)\mid\forall V_c\subseteq A\} \tag{10}$$

根据上述定义，$T_P^s(I)\in T_P^g(I)$ 且 $T_P^{asy}(I)\subseteq T_P^g(I)$。

根据上文的 T_P 算子分别定义布尔网络的同步、异步和一般语义。

给定原子集 A，令 $RA=\{hd(r)\leftarrow bd^+(r)\cup\neg bd^-(r)\mid hd(r)\in A,bd^+(r)\subseteq A,bd^-(r)\subseteq A,bd^+(r)\cap bd^-(r)=\varnothing\}$，$\mathscr{P}=\{P\mid P\in 2^{RA}\}$，$\varepsilon=\{E\mid E\in 2^{2^A\times 2^A}\}$。

定义 6(同步更新语义) 给定原子集 A，定义同步更新语义 $Tr_{syn}:\mathscr{P}\to\varepsilon$，$P\mapsto E$ 如下：

$$Tr_{syn}^P=\{\langle I,J\rangle\mid J=T_P^s(I),\forall I\subseteq A\} \tag{11}$$

定义 7(异步更新语义) 给定原子集 A，定义异步更新语义 $Tr_{asyn}:\mathscr{P}\to\varepsilon$，$P\mapsto E$ 如下：

$$Tr_{asyn}^P=\{\langle I,J\rangle\mid J\in T_P^{as}(I),\forall I\subseteq A\} \tag{12}$$

定义 8(一般更新语义)给定原子集 A，定义一般更新语义 $Tr_{gen}:\mathscr{P}\to\varepsilon$，$P\mapsto E$ 如下：

$$Tr_{gen}^P=\{\langle I,J\rangle\mid J\in T_P^g(I),\forall I\subseteq A\} \tag{13}$$

通过上述定义可以将一个正规逻辑程序 P 在同步语义下映射为一个确定的解释转换集合，在异步和一般语义下映射为不确定的解释转换集合。而这些集合分别表示以正规逻辑程序表示的布尔网络在不同更新语义下所有可能的解释转换。

定义 6－定义 8 与 Chatain 等[28]定义的同步、异步、一般语义比较，这里定义的 3 种语义可以包含点吸引子，而文献[28]中的语义不包含点吸引子。本文定义的语义与文献[26]中定义的语义的不同点在于，一个解释在文献[26]的同步语义下存在多个可能的后继解释，而在本文定义的同步语义下，一个解释只有一个后继解释。

为了能从不确定的解释转换集合中学习表示布尔网络更新的正规逻辑程序，首先需要对解释转换集合进行变换。下节将介绍解释集合变换操作及相关的性质。

4 解释集合变换

本节定义解释转换集合的基本操作，该操作将一个不确定的解释转换集合变为一个确定的解释转换集合。

定义 9(解释对称差)　令 I,J 是 *Herbrand* 解释表示的两个状态。定义解释(状态)对称差 $\Delta(I,J)$ 如下：

$$\Delta(I,J)=(I-J)\cup(J-I) \tag{14}$$

直观上，如果给定一个布尔网络在异步或一般更新语义下的全部解释转换，就能计算出该网络同步更新下的所有解释转换。例如考虑 3 个节点$\{p,q,r\}$的布尔网络中有解释转换 $s_1:\{q,r\}\to\{p,r\}$，$s_2:\{q,r\}\to\{p,q\}$。经过 s_1 转换后节点 p,q 的状态发生了改变，而经过 s_2 转换后节点 p 和 r 的状态发生了改变。如果允许 s_1 和 s_2 中节点的状态变化同时发生则会得到新的解释转换 $s_3:\{q,r\}\to\{p\}$。因为通过 s_1，p 的值从 0 变到了 1，q 的值从 1 变到了 0，根据 s_2，p 的值从 0 变到了 1，而 r 的值从 1 变到了 0。这表明如果选择$\{p,q,r\}$同时更新，即在同步更新下有解释转换 s_3。下面形式地给出了上述计算过程。

给定解释转换序列 $u=\{\langle I,J_1\rangle,\cdots,\langle I,J_n\rangle\}$，令

$$CJ_I^u=(I\cap\bigcap_i J_i)\cup((\bigcup_i J_i)-I) \tag{15}$$

根据式(15)有以下命题。

命题 1　$CJ_I^u=J$ 成立对任意的解释转换集合 $u=\{\langle I,J\rangle\}$。

证明：首先考虑解释 I 与解释 J 的交集为空集的情形。如果 $I\cap J=\emptyset$ 则 $I-J=I$。根据式(15)，$CJ_I^u=(I-I)\cup J=J$。

其次，考虑 $I\cap J\neq\emptyset$ 的情形。根据式(15)，$CJ_I^u=(I\cap J)\cup(J-(I\cap J))=J$。

命题 2　给定原子集 A，令 I,J_1,J_2 分别为 2^A 的子集。令 $u=\{\langle I,J_1\rangle,\langle I,J_2\rangle\}$，下面的等式成立。

$$\begin{aligned}CJ_I^u&=(I-(I-J_1)-(I-J_2))\cup(\bigcup_{i=1}^{2}(\Delta(I,J_i)\cap J_i))\\&=(I\cap J_1\cap J_2)\cup(\bigcup_{i=1}^{2}(\Delta(I,J_i)\cap J_i))\end{aligned} \tag{16}$$

证明：这里只需要证明 $I-(I-J_1)-(I-J_2)=I\cap J_1\cap J_2$。

$$\begin{aligned}I-(I-J_1)-(I-J_2)&=(I\cap\overline{I\cap\overline{J_1}})-(I-J_2)\\&=(I\cap J_1)\cap\overline{I\cap\overline{J_2}}\\&=I\cap J_1\cap J_2\end{aligned}$$

根据式(16)，CJ_I^u 可表示为：

$$CJ_I^u=(I\cap J_1\cap J_2)\cup(\bigcup_{i=1}^{2}(J_i-I)) \tag{17}$$

根据命题 2，有以下推论。

推论 2　给定原子集 A，令 $I,J_1,\cdots,J_n$ 分别为 2^A 的子集。令 $u=\{\langle I,J_1\rangle,\cdots,\langle I,J_n\rangle\}$，则下面的等式成立。

$$\begin{aligned}CJ_I^u&=(I-(I-J_1)-\cdots-(I-J_n))\cup(\bigcup_{i=1}^{n}(\Delta(I,J_i)\cap J_i))\\&=(I\cap\bigcap_{i=1}^{n}J_i)\cup(\bigcup_{i=1}^{n}(\Delta(I,J_i)\cap J_i))\end{aligned} \tag{18}$$

下面的定理表明，根据集合 $u=\{\langle I,J\rangle,\langle I,J'\rangle\}$ 转换后得到的集合 J_I^u，采用批量计算和迭代计算的结果相同。

命题 3　给定原子集 A。令 I,J_1,J_2 分别是 2^A 的子集。令 $u=\{\langle I,J_1\rangle,\langle I,J_2\rangle\}$ 。下面的等式成立。

$$\begin{aligned}&(I-(I-J_1)-(I-J_2))\cup(\bigcup_{i=1}^{2}(J_i-I)\\&=(((I-(I-J_1))\cup(J_1-I))-(I-J_2))\cup(J_2-I)\end{aligned} \tag{19}$$

证明：根据命题 2，

$$\text{等式左边}=(I\cap J_1\cap J_2)\cup(J_1-I)\cup(J_2-I) \tag{20}$$

根据命题 1，

$$\begin{aligned}&(((I-(I-J_1))\cup(J_1-I))-(I-J_2))\cup(J_2-I)\\&=(J_1-(I-J_2))\cup(J_2-I)\\&=(J_1\cap\overline{(I-J_2)})\cup(J_2-I)\\&=(J_1\cap\overline{(I-J_2)})\cup(J_2-I)\\&=(J_1\cap(\bar{I}\cup J_2))\cup(J_2-I)\\&=(J_1\cap\bar{I})\cup(J_1\cap J_2)\cup(J_2-I)\\&=(J_1\cap J_2)\cup(J_1-I)\cup(J_2-I)\end{aligned} \tag{21}$$

下面证明集合(20)=(21)。

对比式(20)、式(21)会发现唯一的不同在于式(20)是 $I\cap J_1\cap J_2$，而式(21)中是 $J_1\cap J_2$，因此只需对这部分进行证明。

显然 $\forall q\in(20)$，则 $q\in(21)$。

$\forall q\in(21)$考虑如下情形：

(1)$q\in J_1-I$ 或者 $q\in J_2-I$，则 $q\in(20)$。

(2)对 $q\in J_1\cap J_2$，如果 $q\in J_1\cap J_2\cap I$，则 $q\in(20)$。

(3)$q\in J_1\cap J_2$，如果 $q\in J_1\cap J_2$ 且 $q\notin I$，意味着 $q\in(J_1-I)$或者 $q\in(J_2-I)$，所以 $q\in(20)$。

示例 1　令 $A=\{p,q,r\}$。令 $\langle I,J_1\rangle=\langle\emptyset,\{r\}\rangle$，$\langle I,J_2\rangle=\langle\emptyset,\emptyset\rangle$。

$$(I-(I-J_1)-(I-J_1))\cup(J_1-I)\cup(J_2-I)=\{r\}$$

$$((I-(I-J_1)\cup(J_1-I))-(I-J_2))\cup(J_2-I)=\{r\}$$

推论 3　给定原子集 A。令 $I,J_1,\cdots,J_n$ 分别是 2^A 的子集，$u=\{\langle I,J_1\rangle,\cdots,\langle I,J_n\rangle\}(n\geqslant 2)$。下面的等式成立。

$$\begin{aligned}&(I-(I-J_1)-\cdots-(I-J_n))\cup(\bigcup_{i=1}^{n}(\Delta(I,J_i)\cap J_i))\\&=((\cdots((I-(I-J_1))\cup(J_1-I))\cdots)-(I-J_n))\cup(J_n-I)\end{aligned} \tag{22}$$

证明：基始。当 $n=2$ 时，根据命题 3，等式成立。

假设 $n=k-1$ 时等式成立。下面证明 $n=k$ 时成立。

根据推论 2，要证明 $n=k$ 时等式成立，即证明下面的等式成立即可。

$$\begin{aligned}&(I\cap\bigcap_{i=1}^{n}J_i)\cup(\bigcup_{i=1}^{n}(\Delta(I,J_i)\cap J_i))\\&=(((I\cap\bigcap_{i=1}^{n-1}J_i)\cup(\bigcup_{i=1}^{n-1}(\Delta(I,J_i)\cap J_i)))-(I-J_n))\cup(J_n-I)\end{aligned} \tag{23}$$

令 $A=(I\cap\bigcap_{i=1}^{n-1}J_i)$，$B=\bigcup_{i=1}^{n-1}(\Delta(I,J_i)\cap J_i))$。等式(23)右边可改写为：$((A\cup B)-(I-J_n))\cup(J_n-I)$。

接下来证明$((A\cup B)-(I-J_n))\cup(J_n-I)$与等式(23)左边相等。

$$\begin{aligned}&((A\cup B)-(I-J_n))\cup(J_n-I)\\&=((A\cup B)\cap\overline{I\cap\overline{J_n}})\cup(J_n-I)\\&=((A\cup B)\cap(I\cap J_n))\cup(J_n-I)\\&=((A\cap\bar{I})\cup(B\cap\bar{I})\cup(A\cap J_n)\cup(B\cap J_n))\cup(J_n-I)\\&=(B\cup(A\cap J_n)\cup(B\cap J_n))\cup(J_n-I)\end{aligned}$$

$$=(A\cap J_n)\cup B\cup(J_n-I)$$

$$=(I\cap\bigcap_{i=1}^{n}J_i)\cup\bigcup_{i=1}^{n}(\Delta(I,J_i)\cap J_i)$$

上面讨论了如果一个解释有多个可能的后继解释，则通过计算可将这些可能的后继解释转换为一个新的解释。因此，一个不确定的解释转换集通过这种转换会得到一个新的解释转换集合，这个新的解释转换集中每个解释的后继解释是唯一的。下面给出确定状态解释集和非确定状态解释集的定义。

定义 10（确定解释转换集合） 一个解释转换集合 $E=\{\langle I_1,J_1\rangle,\cdots,\langle I_n,J_n\rangle\}$ 是确定解释转换集合，如果 E 中 $\exists\langle I_i,J_i\rangle,\langle I_j,J_j\rangle\in E$ $(1\leqslant i,j\leqslant n$ 且 $i\neq j)$，使得 $I_i=I_j$ 且 $J_i\neq J_j$。

定义 11（非确定状态解释集合） 一个解释转换集合 $E=\{\langle I_1,J_1\rangle,\cdots,\langle I_n,J_n\rangle\}$ 是非确定解释转换集合，如果 E 中 $\exists\langle I_i,J_i\rangle,\langle I_j,J_j\rangle\in E$ $(1\leqslant i,j\leqslant n$ 且 $i\neq j)$ 使得 $I_i=I_j$ 且 $J_i\neq J_j$。

给定一个解释集合 E，令 $E^i=\{I\mid\langle I,J\rangle\in E\}$ 及 $T_E^l=\{\langle I',J'\rangle\mid\langle I',J'\rangle\in E$ 且 $I'=I\}$。

给定原子集 A，$E\subseteq 2^A\times 2^A$，定义 12 给出转换函数 $dtm(E)$：$2^A\times 2^A\to 2^A\times 2^A$。

定义 12 给定解释转换集合 E，令 $u=T_E^l$。定义解释转换函数 dtm 如下：

$$dtm(E)=\{\langle I,CJ_I^u\rangle\mid\forall I\in E^i\}$$

由命题 1 可得到推论 4。

推论 4 E 是确定状态转换集，则 $dtm(E)=E$。

示例 2 令 $A=\{p,q\}$，$E=\{\langle\{p\},\{q\}\rangle\}$。根据定义 12，$dtm(E)=\{\langle\{p\},\{q\}\rangle\}=E$。

令 $A=\{p,q\}$，$E=\{\langle\{p\},\{q\}\rangle,\langle\{p\},\{r\}\rangle\}$。根据定义 12，$dtm(E)=\{\langle\{p\},\{qr\}\rangle\}\neq E$。因为 E 是一个非确定解释转换集合。

命题 4 给定正规逻辑程序 P 和原子集 A，P 中出现的原子都是 A 的元素。令 $E_s=Tr_{syn}^P$，$E_{as}=Tr_{asyn}^P$，$E_g=Tr_{gen}^P$，则 $dtm(E_{as})=dtm(E_g)=dtm(E_s)=E_s$。

证明：首先根据推论 4，$dtm(E_s)=E_s$。

接下来证明 $dtm(E_s)=dtm(E_{as})$，即证明 $\forall I\in E^i$，$CJ_I^u=T_P^s(I)$，其中 $u=\{\langle I,T_P^{as}(I,\{q\})\rangle\mid q\in A\}$。

(1)根据定义 2 及定义 4，$\forall J=T_P^{as}(I,\{q\})$：

1)对 $q\in I$，$J=I$ 如果 $\{q\}\cap T_P^s(I)\neq\emptyset$。

2)对 $q\in I$，$J=I-\{q\}$ 如果 $\{q\}\cap T_P^s(I)=\emptyset$。

3)对 $q\notin I$，$J=I\cup\{q\}$ 如果 $\{q\}\cap T_P^s(I)\neq\emptyset$。

4)对 $q\notin I$，$J=I$ 如果 $\{q\}\cap T_P^s(I)=\emptyset$。

(2)根据(1)有 $u=S_1\cup S_2\cup S_3$，其中 $S_1=\{\langle I,I\rangle\}$，$S_2=\{\langle I,I-\{q\}\rangle\mid q\in I$ 且 $\{q\}\cap T_P^s(I)=\emptyset\}$，$S_3=\{\langle I,I\cup\{q\}\rangle\mid q\notin I$ 且 $\{q\}\cap T_P^s(I)\neq\emptyset\}$。

记 $C_E^l=\{J'\mid\langle I',J'\rangle\in E$ 且 $I'=I\}$。

$\bigcap C_{S_2}^l=I-A_1$。

其中，$A_1=\{q\mid q\in I$ 且 $q\cap T_P^s(I)=\emptyset\}$。

$\bigcap C_{S_3}^l=(I-A_2)=I$ 其中 $A_2=\{q\mid q\notin I$ 且 $q\cap T_P^s(I)\neq\emptyset\}$。

(3)根据推论 3，有：

$$CJ_I^u=(I\cap J_1\cap\cdots\cap J_n)\cup(\bigcup_{i=1}^{n}(J_i-I))$$

$$=I\cap C_{S_1}^l\cap(\cap C_{S_2}^l)\cap(\cap C_{S_3}^l)\cup(\bigcup_{i=1}^{n}(J_i-I))$$

$$=(I-A_1)\cup A_2$$

根据(2)有 $\forall q\in(I-A_1)\cup A_2$，$q\in T_P^s(I)$。对于 $q\in A_2$，结论显然成立；对于 $q\in(I-A_1)$，如果 $q\notin T_P^s(I)$，则 $q\in A_1$ 矛盾。

而 $\forall q\in T_P^s(I)$ 则 $q\in CJ_I^u$。原因如下：如果 $q\notin I$，则 $q\in A_2$；如果 $q\in I$，则 $q\in(I-A_1)$，否则矛盾。

最后证明 $dtm(E_g)=dtm(E_s)$。对于 $I\in E_g^l$，令 $u=\{\langle I,T_P^g(I,V_c)\rangle\mid V_c\in 2^A\}$。也可将其写成 $u=\{\langle I,J_1\rangle,\cdots,\langle I,J_n\rangle\}$。

$$CJ_I^u=(I\cap J_1\cap\cdots\cap J_n)\cup(\bigcup_{i=1}^{n}(J_i-I))$$

$$=(I-A_1)\cup A_2$$

其中，$A_1=\{q\mid q\in A$ 且 $\exists\langle I,J\rangle\in u$，使得 $q\in I$ 且 $q\notin J\}$，$A_2=\{q\mid q\in A$ 且 $\exists\langle I,J\rangle\in u$，使得 $q\notin I$ 且 $q\in J\}$。

$CJ_I^u=T_P^s(I)$ 的证明过程与(3)的证明过程类似。

命题 4 表明，根据异步更新布尔网络的解释转换集合学习布尔网络时，可以通过部分数据推测出该网络以正规逻辑程序表示的布尔函数，因此有如下推论。

推论 5 给定正规逻辑程序 P 及原子集 A。令 $\bigcup_{i=1}^{n}V_{c_i}=A$，$I\in 2^A$ 及 $E=\{\langle I,T_P^g(I,V_{c_i})\rangle\mid 1\leqslant i\leqslant n\}$，则 $dtm(E)=T_P^s(I)$。

令 $ET=\{\langle I,T_P^g(I,V_{c_i})\rangle\mid\forall I\in 2^A,\forall 1\leqslant i\leqslant n\}$，则 $dtm(ET)=Tr_{syn}^P$。

5 LFNDIT 算法

本节给出了学习任务及学习算法。

定义 13（给定解释转换集合的学习任务） 给定解释转换集合 E，学习一个正规逻辑程序 P 使得 $E\subseteq Tr_{gen}^P$。

根据定义 12 提出的非确定解释转换集合变为确定解释转换集合的操作，并结合 LF1T 自顶向下算法[19]给出能完成定义 13 学习任务的非确定解释转换学习(LFNDIT)的算法伪代码。算法 1 中的 2—12 行为 LF1T 算法的伪代码。

算法 1 非确定解释转换学习算法 LFNDIT

输入：解释转换集合 E，原子集 A

输出：正规逻辑程序 P

```
1. E_syn:=dtm(E)
2. P:={q←|q∈A}
3. for each⟨I,J⟩∈E_syn do
4.   for each q∈A do
5.     if q∉J then
6.       {P'={q←bd(r)|I⊨bd(r)且 hd(r)=q,∀r∈P}
7.       np=lst(P',I)
8.       P:=P-P'
9.       for each r∈np do
10.          if ∃r'∈Ps.tr'<r then
11.            P:=P∪{r}
12. return P
```

算法 2　$dtm(E)$转换算法

输入：原子集 A，$E\subseteq 2^A\times 2^A$

输出：确定解释转换集合 E_{syn}

1. $E_{syn}:=\emptyset$
2. for each $I\in E^i$ do
3.　$S1:=\emptyset$
4.　$S2:=\emptyset$
5.　for each $J\in T_E^I$ do
6.　$S1:=S1\cup(\Delta(I,J)\cap I)$
7.　$S2:=S2\cup(\Delta(I,J)\cap J)$
8.　$C_{syn}^I:=(I-S1)\cup S2$
9.　$E_{syn}:=E_{syn}\cup\{\langle I,C_{syn}^I\rangle\}$

算法 3　P 关于 I 的最小特化 $lst(P,I)$

输入：正规逻辑程序 P，解释 $I\subseteq 2^A$

输出：正规逻辑程序 np

1. $np:=\emptyset$
2. for each $r\in P$ do
3.　$np:=np\cup\{hd(r)\leftarrow bd(r)\cup\{l\}\mid l\in((I\cup\neg(A-I))-bd(r))\}$
4. return np

接下来讨论观察到的数据集与根据该数据集学习到的逻辑程序 P 在 3 种不同语义下产生的状态转换集合之间的关系。

推论 6　给定一个确定转换集 E，$P=LFNDIT(E)$。对集合 E 的每个转换$\langle I,J\rangle$满足 $J=T_P^s(I)$。

一个程序 P 关于 E 完备，如果$\forall\langle I,J\rangle\in E\langle I,J\rangle\in Tr_P^g$；程序 P 关于 E 可靠，如果$\forall\langle I,J\rangle\in Tr_P^g$，$\langle I,J\rangle\in E$。这里由于 P 是通过 $LFNDIT(E)$学习出来的，因此可说 LFNDIT 算法关于 E 是否完备和可靠。如果 E 是确定状态转换集合，则 LFNDIT(E)关于 E 是完备、可靠的[18]。如果 E 是不确定状态转换集合，通常情形下算法关于 E 是完备但不是可靠的。下面证明算法的完备性。

命题 5(LFNDIT 算法的完备性)　给定状态集合 E，算法 LFNDIT 关于 E 完备。即 $\forall\langle I,J\rangle\in E$，$\langle I,J\rangle\in Tr_{LFNDIT(E)}^g$。

证明：要证明 LFNDIT 关于 E 完备，只需要证明$\forall\langle I,J\rangle\in E$，$\exists V_c\subseteq A$，使得 $T_{LFNDIT(E)}^g(I,V_c)=J$。令 $C_E^I=\{J\mid\langle I,J\rangle\in E\}$，令 $P=LFNDIT(E)$。

(1)令 $E'=dtm(E)$，$LFNDIT(E)$关于 E'是完备和可靠的。

(2)$\forall I\in E^i$，$\exists\langle I',J'\rangle\in E'$，使得 $I'=I$ 且 $J'=CJ_I^u$，其中 $u=\{\langle I',J'\rangle\mid\langle I',J'\rangle\in E$ 且 $I'=I\}$。

$\forall\langle I,J\rangle\in E$。令 $V_c=\Delta(I,J)$，则 $V_c=s_1\cup s_2$，其中 $s_1=\Delta(I,J)\cap I$，$s_2=\Delta(I,J)\cap J$。则 $J=(I-V_c)\cup s_2$。

根据式(15)，如果原子 $q\in V_c\cap I$，则 $q\notin CJ_I^u$；如果 $q\in V_c\cap J$，则 $q\in CJ_I^u$。

$$\begin{aligned}T_P^g(I,V_c)&=(I-\Delta(I,J))\cup(V_c\cap T_P^s(I))\\&=(I-\Delta(I,J))\cup(V_c\cap J_I^s)\\&=(I-V_c)\cup s_2\\&=J\end{aligned}$$

(3)由(2)可得，$\forall\langle I,J\rangle\in E$，总存在 V_c，使得 $T_P^g(I,V_c)=J$。

考虑最坏情形，即 n 个节点的布尔网络解释总数为 2^n，每个解释的可能后继解释有 2^n 个，因此解释转换集合有 $2^n\cdot 2^n$ 个元素。根据文献[24]，LFNDIT 算法得到最坏情形的时间复杂度为 $O((n+1)\cdot 4^n)$。

6　实验

LFNDIT 算法原型由 python3.6 编程实现。先根据文献[23]提出的转换方法将文献[3]中的 mammalian cell，fission yeast cell，budding yeast cell，Arabidopsis thaliana 这 4 个布尔网络分别转换为对应的正规逻辑程序。然后根据得到的逻辑程序生成 4 个布尔网络在同步、异步、一般语义下的解释转换集合，并将其作为测试数据集。由于生成 T-helper cell 和 T-cell receptor 一般语义下解释转换集的时间过长，因此实验中不考虑这两个布尔网络。所有的实验在 Intel Xeon sliver (4110，2.1GHz)上进行，内存为 64GB，操作系统为 Centos7。表 1 列出了每个布尔网络在不同的语义下生成的状态转化的数目，算法从不同解释转换集学习所花费的时间及最终学得的规则条数。实验结果表明同一个布尔网络在 3 种不同的解释转换集合上学习出的规则条数是相同的。这说明本文提出的将不确定解释转换集转变为确定转换集的方法是正确的。

表 1　LFNDIT 算法的测试结果

Table 1　Test results of LFNDIT algorithm

布尔网络	语义	状态转换数	时间/s	规则数
mammalian cell	同步	1024	7.64	22
	异步	5096	8.34	22
	一般	30971	9.84	22
fission yeast cell	同步	1024	8.61	24
	异步	5168	8.76	24
	一般	33721	11.54	24
budding yeastcell	同步	4096	69.47	54
	异步	24064	75.35	54
	一般	260557	162.37	54
Aboraposis thaliana	同步	32678	116.59	28
	异步	491520	2613.14	28
	一般	6992019	23259.35	28

由于 mammalian cell 和 fission yeast cell 两个网络的节点数都是 10，从表 1 可以看出，此时算法花费的时间与布尔函数表示为逻辑程序规则数有关，规则数越多，算法花费的学习时间越长。Fission yeast cell 网络在一般语义下的解释转换个数比 budding yeast cell 在异步语义下的转换个数要多，但 budding yeast cell 网络的节点个数是 12，因此解释转换数目相同的情形下，算法所花费的时间与网络节点数相关，节点数越多，所花费的时间越长。

结束语　本文提出了表示布尔网络 3 种不同更新方式的语义，分别是同步、异步和一般语义；根据这 3 种语义可将一个正规逻辑程序映射到不同的解释转换集合；讨论了 3 种语义映射下解释转换集合的关系。同时提出了将不确定解释转换集合变为确定解释转换集合的方法，并证明了其相关性质；并根据该方法对 LF1T 自顶向下算法进行了修改，修改后的算法 LFNDIT 解除了 LFDT 算法对解释转换集合的限制。利用 LFNDIT 从不确定的解释转换集合学习以正规逻辑程序表示的布尔网络，讨论了算法的可靠性和完备性。实验结

果表明，给定一个布尔网络在3种语义下的解释转换集合，LFNDIT根据这3个集合学出的规则集是相同的，这表明文中提出的转换方法是正确的。利用本文中的算法从不确定的解释转换集学到的规则数目比LFDT和GULA算法学到的规则数少。下一步工作考虑将算法进行并行化以便处理大规模的状态转换集合以及从部分状态转换集合学习。

参 考 文 献

[1] KAUFFMAN S A. Metabolic stability and epigenesis in randomly constructed genetic nets[J]. Journal of Theoretical Biology, 1969, 22(3): 437-467.

[2] KAUFFMAN S A. Origins of order in evolution: self-organization and selection[J]. In Understanding Origins, 10. 1007/978-94-015-8054-0. Springer, 1992, (Chapter 8): 153-181.

[3] DUBROVA E, TESLENKO M. A sat-based Algorithm for finding attractors in synchronous Boolean networks[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2011, 8(5): 1393-1399.

[4] VELIZ-CUBA A. Reduction of boolean network models[J]. Journal of Theoretical Biology, 2011, 289: 167-172.

[5] CHENG D Z, QI H S. Controllability and observability of boolean control networks[J]. Automatica, 2009, 45(7): 1659-1667.

[6] JARRAH A S, LAUBENBACHER R, VELIZ-CUBA A. The dynamics of conjunctive and disjunctive boolean network models [J]. Bulletin of Mathematical Biology, 2010, 72(6): 1425-1447.

[7] KRAWITZ P, SHMULEVICH I. Basin entropy in boolean network ensembles[J]. Physical Review Letters, 2007, 98(15): 158701.

[8] MENINI L, POSSIERI C, TORNAMBÉ A. Boolean network analysis through the joint use of linear algebra and algebraic geometry[J]. Journal of theoretical biology, 2019, 472: 46-53.

[9] TONELLO E, FARCOT E, CHAOUIYA C. Local negative circuits and cyclic attractors in Boolean networks with at most five components[J]. SIAM Journal on Applied Dynamical Systems, 2019, 18(1): 68-79.

[10] LIANG S D, FUHRMAN S, SOMOGYI R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures[C]// Pacific Symposium on Biocomputing. 1998, 3: 18-29.

[11] HAIDER S, PAL R. Boolean network inference from time series data incorporating prior biological knowledge[J]. BMC genomics, 2012, 13(6): S9.

[12] OSTROWSKI M, PAULEVÉ L, SCHAUB T, et al. Boolean network identification from multiplex time series data[C]// International Conference on Computational Methods in Systems Biology. Springer, 2015: 170-181.

[13] OSTROWSKI M, PAULEVÉ L, SCHAUB T, et al. Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming[J]. Biosystems, 2016, 149: 139-153.

[14] LÄHDESMÄKI H, SHMULEVICH I, YLI-HARJA O. On learning gene regulatory networks under the boolean network model[J]. Machine Learning, 2003, 52(1/2): 147-167.

[15] STALIN M, MIGUEL C, EUGENIO A, et al. Griffin: A Tool for Symbolic Inference of Synchronous Boolean Molecular Networks[J]. Frontiers in Genetics, 2018, 9: 39.

[16] SHI N, ZHU Z, TANG K, et al. ATEN: And/Or tree ensemble for inferring accurate Boolean network topology and dynamics [J]. Bioinformatics, 2020, 36(2): 578-585.

[17] RÉDA C, WILCZYŃSKI B. Automated inference of gene regulatory networks using explicit regulatory modules[J]. Journal of Theoretical Biology, 2020, 486: 110091.

[18] YUE J, YAN Y, CHEN Z, et al. Identification of predictors of Boolean networks from observed attractor states[J]. Mathematical Methods in the Applied Sciences, 2019, 42(11): 3848-3864.

[19] GERSHENSON C. Classification of random boolean networks [J]. arXiv: cs/0208001, 2002.

[20] INOUE K. Logic programming for boolean networks [C]// Twenty-Second International Joint Conference on Artificial Intelligence. 2011.

[21] INOUE K, SAKAMA C. Oscillating behavior of logic programs [M]// Correct Reasoning. Springer, Berlin, Heidelberg, 2012: 345-362.

[22] MUGGLETON S. Inductive logic programming[J]. New Generation Computing, 1991, 8(4): 295-318.

[23] MUGGLETON S, DE RAEDT L. Inductive logic programming: Theory and methods[J]. The Journal of Logic Programming, 1994, 19: 629-679.

[24] INOUE K, RIBEIRO T, SAKAMA C. Learning from interpretation transition[J]. Machine Learning, 2014, 94(1): 51-79.

[25] RIBEIRO T, INOUE K. Learning prime implicant conditions from interpretation transition[M]// Inductive Logic Programming. Springer, Cham, 2015: 108-125.

[26] HUANG Y, WANG Y, ZHANG Y, et al. Learning Disjunctive Logic Programs from Interpretation Transition[C]// 26th International Conference on Inductive Logic Programming. 2016: 34-40.

[27] RIBEIRO T, FOLSCHETTE M, MAGNIN M, et al. Learning Dynamics with Synchronous, Asynchronous and General Semantics[C]// International Conference on Inductive Logic Programming. Springer, Cham, 2018: 118-140.

[28] CHATAIN T, HAAR S, PAULEVÉ L. Boolean networks: beyond generalized asynchronicity[C]// International Workshop on Cellular Automata and Discrete Complex Systems. Springer, Cham, 2018: 29-42.

HUANG Yi, born in 1976, Ph.D candidate, professor, is a member of China Computer Federation. Her main research interests include artificial intelligence, knowledge representation and reasoning.

WANG Yi-song, born in 1975, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include artificial intelligence, knowledge representation and reasoning.