# 华东师范大学

## 计算机逻辑基础论文

# 多值逻辑介绍和DS3模型的构建

姓名：赵晗瑜

学号：10205101530

# *1 非经典逻辑概述*

## 1.1 什么是非经典逻辑？[2]

非经典逻辑是一种与经典逻辑相对的一种现代逻辑分支，基于二值经典逻辑，非经典逻辑按照“拓展”和“脱轨”两种方式发展起来。

按照“拓展”方式发展起来的非经典逻辑分支称为：“**扩展的逻辑**”，扩展的意思即在经典逻辑的命题和谓词演算中添加新的公理、规则以及新的逻辑算子。这样的非经典逻辑分支主要包括：

- **模态逻辑**$(modal\ Logic)$：即研究必然、可能以及其相关概念的逻辑的一个分支，是处理用模态如“可能”、“或许”、“可以”、“一定”、“必然”等限定的句子的逻辑。
- **时态逻辑**$(tense\ Logic)$：时态逻辑的基本成分是时态语句，时态逻辑主要包括两类，其一是线性时间逻辑$LTL$；其二是分支时间逻辑$CTL$。$LTL$和$CTL$都是进行模型检测的一种研究工具，模型检测就是对问题$M,s\vdash\varphi$是否成立进行计算的过程，其中：$M$是所考虑系统的一个适当模型;$s$是该模型的一个状态; $\vdash$是满足关系; $\varphi$是一个$LTL$或$CTL$公式。
- **规范逻辑**$(deontic\ Logic)$：规范逻辑是一种含有必须、允许等规范词的规范命题以及规范演绎系统的现代逻辑分支。
- **优先逻辑**$(Logic\ of\ Logic)$：优先逻辑即逻辑现在性，描述事物之间在逻辑上的优先和非优先地位。
- **问题逻辑**$(Logic\ of\ Question)$：问题逻辑是认知逻辑的一个分支，研究在问题和答案内所产生的各种逻辑问题。

按照“脱轨”方式发展起来的非经典逻辑分支称为：“**脱轨逻辑**”，脱轨的意思即使用脱轨逻辑的系统对经典逻辑的公理和规则进行了限制和根本性的修改，使其从根本上脱离了二值经典逻辑的轨道。这样的非经典逻辑分支主要包括：

- **多值逻辑**$(many-valued\ Logic)$：多值逻辑是一种有多于两个的可能真值的一种非经典逻辑系统，在多值逻辑中有三种命题真值的解释，分别是：①三值逻辑的解释；②n值逻辑的解释；③可数无穷多值逻辑的解释。
- **模糊逻辑** $(Fuzzy\ Logic)$：模糊逻辑是一种研究模糊推理的有效性的一类非经典逻辑系统，模糊逻辑借助于隶属度函数的概念，区分模糊集合，处理模糊关系，解决因“排中律”产生的种种不确定性问题。
- **量子逻辑** $(Quantum\ Logic)$：量子逻辑是一种全新的、非布尔型的非经典逻辑，其语义和语法由一种具有非分配性和不对易性特征的代数结构所确定，其形式具有多样性，技术应用具有广泛的可能性。

- **直觉主义逻辑** $(Iutuitional\ Logic)$：直觉主义逻辑是一种具有存在性质的且适合其他形式的数学构造主义，其公式语法类似于命题逻辑或一阶逻辑，与经典逻辑连结词是可互定义的不同，直觉主义逻辑的连结词是不可互定义的，使用$\rightarrow, \wedge, \vee, \perp$作为基本连结词。

总之，非经典逻辑通过拓展真值来刻画更具多样性的事实和知识，它更加适用于表征更加复杂的现实世界。首先，非经典逻辑命题的真值并不是非此即彼的，是具有多值性的，超越了0和1所解释的有限值逻辑和无限值逻辑；其次，非经典逻辑的真值在其确立过程中是不确定的，其命题的真值对语境具有依赖性，分别表现为真值对情境、时间、认识主体、主题心理意向性等方面的依赖；最后，当$\varGamma \rightarrow s$，$\varGamma \cup \varDelta \rightarrow s$也必然会成立，即当系统中有新的事实加入时，得出的结论不会因此丧失，而是继续保持，即非经典逻辑的推理是具有非单调的特性的。以上特征针对所有非经典逻辑的系统都成立，非经典逻辑是一种对于经典逻辑的完善，逐渐地体现出在计算机科学技术或人工智能等领域的研究意义和价值。

## 1.2 为什么会出现非经典逻辑，它发展的动力是什么？[1]

经典逻辑系统更加注重推理的一种“形式化”，它是一种严谨精确的摒弃日常语言的歧义和模糊性的以人工符号的形式表征为基础的逻辑系统，经典逻辑系统具有其独特的特征，但正是因为这些独特特征所存在的不足催生着非经典逻辑的诞生和发展。

首先是经典逻辑的**二值性**，在经典逻辑的世界中，事件只有两种状态，即“真”或“假”，是一种非黑即白的确定性判断，例如电源开关只有“开”或“关”两种状态，而不允许其处于半开半闭的状态，基于此思想设计的逻辑门电路促进了人工智能自动推演系统的发展。在某种程度上，经典逻辑的二值性已经可以充分地进行知识的表征，但是它不足以获取所有人类可以做的自然推理。因此克服经典逻辑的真值的局限性成为亟待解决的问题。

其次是经典逻辑的**实质蕴含**特性，为了解决“实质蕴涵怪论”问题，逻辑学家刘易斯致力于构造严格蕴涵，创建了模态逻辑系统，模态逻辑系统是针对实质蕴涵的强化，解决了“实质蕴涵怪论”问题。那么，什么是实质蕴涵怪论呢?

在命题$A \rightarrow B$中，当$A$为$T$，$B$为$F$时，该命题为假，那么在这样一种前提条件下，命题$A \rightarrow (B \rightarrow A)$(真命题被任一命题所蕴含)和命题$\neg A \rightarrow (A \rightarrow B)$(假命题蕴含任一命题)均成立，但这样的命题却被许多逻辑学家认为不符合人们的生活常识和知觉，这就是“实质蕴涵怪论”。

最后是经典逻辑命题真值表现出的**单一性**，考虑一个在经典逻辑系统中具有矛盾的命题$p$，命题$p$表示“鸟会飞”。现假设在现实世界中命题$p$为真，即在现实世界中鸟会飞，但是鸵鸟也是一种鸟，它却不会飞，因此该命题在现实语境边界上是假。我们再假设命题$p$为假，显然该命题错误，因为大多数鸟都是会飞的。针对这一种矛盾，可知经典逻辑系统的命题真值表现出的单一性收到了挑战。而与之相顺应的是，逻辑学家克里普克等人创立的“可能世界语义学”严密且清晰地揭示了多种模态公理系统的直观背景，在这个非经典逻辑框架下，如果命题$p$为真，那么至少存在一个 **可能世界**$W_0$使得$p$成立，其中$W_0$是剔除所有不会飞的鸟的世界。因此，可能世界语义学指出了经典逻辑中亟待解决的问题——**命题真值所表现出的单一性**。

总的来说，尽管经典逻辑的形式化和公理化方法发展迅速，但是其不能表征世界的所有状态以及人们的演绎推理能力，因此经典逻辑获取更加丰富的表征能力的需要以及经典语义学修正的需要催生着非经典逻辑的诞生于不断研究发展。

## 1.3 研究非经典逻辑有什么意义？

非经典逻辑是一种语境可演绎化的分析方法，被广泛应用于 **知识表征、自然语言分析、常识推理以及逻辑编程**等领域。随着大数据时代的到来，非经典逻辑越来越能跟更多数学分支相结合，多种多样的数学思想的应用使其具有更强的表征力。再者，非经典逻辑越来越能与具体的语境相结合，越来越能够揭示逻辑推理和语境要素之间的关系。

# 2 非经典逻辑与经典逻辑的联系和区别

## 2.1 联系

非经典逻辑是在经典逻辑的基础上发展起来的，是一种对经典逻辑的有用却富有表征性的**补充和完善**，针对经典逻辑系统中**二值性、实质蕴涵性、单一性**等特性所揭露出来的问题，非经典逻辑一一提出了对应的解决方案。从要克服经典逻辑的真值的局限性开始，非经典逻辑开始逐渐浮出水面，以经典逻辑系统为基础，以能够表征经典逻辑所不能表征的世界的状态和能够表征人类更全面的演绎推理的方法为根本导向，逐渐萌生、发展、壮大、爆发。尽管现如今的非经典逻辑仍具有很多缺点，但伴随这些缺点不断完善是我们的目标。

## 2.2 区别

下表是经典逻辑和非经典逻辑分别在推理方法、辖域取值、运算法则、逻辑算符、单调性等方面的区别

| | 经典逻辑 | 非经典逻辑 |
|---|---|---|
| 推理方法 | 演绎逻辑推理 | 归纳逻辑推理 |
| 辖域取值 | 二值逻辑 | 多值和模糊推理 |
| 运算法则 | 否定之否定、德摩根定律等 | 不成立 |
| 逻辑算符 | 5个连词、2个量词（是非判断） | 更多，模态算符“必然”、“可能”，“应该”和“偶然” |
| 单调性 | 单调：已知事实均为充分可信，不含随着新事实的出现而使原有事实变为假的可能性 | 非单调：否定之否定的辩证发展过程 |

### 2.2.1 在侧重点上

经典逻辑和非经典逻辑虽然都注重形式化并且形成了严格的逻辑演算系统，但是非经典逻辑更加注重的是数学之外的自然语言形式化和非形式推理的形式化。并且，非经典逻辑的很多分支不仅在形式上注重有效，而且更加注重其所应用的情境的意义，这是一种以更新的角度探讨系统形式化的方式，因此它面临的挑战和困难也比经典逻辑更大。

### 2.2.2 在辖域取值上

经典逻辑是二值的，“非此即彼”的思维方式带来了很大的局限性；而非经典逻辑允许一个命题有三值甚至更多值，使得一些诸如模糊语句、模态句得到了充分的表征，而这些语句是二值逻辑所无暇顾及的。因此，非经典逻辑在辖域取值方面的拓展拓深了逻辑领域。

### 2.2.3 在应用范围上

非经典逻辑应用的范围更加广泛，被广泛应用于人工智能方面的研究，以常识推理(一种非单调逻辑)为例。常识推理即人们基于不完全的信息推出某些结论，当得到更完善的信息后，可以改变甚至收回原来的结论。

不完全信息的推理研究知识是人工智能研究的一个重要问题，而当代计算机建立在精确逻辑和二值基础上，因此常识知识和专家知识有着不完全性和不精确性。因此，经典逻辑在处理常识推理时显得无能为力，人工智能需要从日常推理中抽象出一个较为完善的非单调系统，在往后的发展中，弗协调逻辑、多值逻辑、模糊逻辑等逻辑系统均被引入到人工智能来处理模糊性和不完全信息的推理，这就体现了经典逻辑和非经典逻辑在应用范围上的不同。

# *3 多值逻辑(many-valued logic)的详细介绍*

多值逻辑是一个命题真值有两个以上的逻辑家族。也就是说，多值逻辑可以表达除了真或假之外的其他可能的真值。多值逻辑的思想来源于亚里士多德有关未来偶然时间的思考中。现如今，多值逻辑被广泛应用于解决不同领域的问题，尤其是三值逻辑和四值逻辑被广为人知地利用在计算机科学地应用中，同时，人们还注意到，所谓的“模糊逻辑”被归类为一种多值逻辑。

现如今有多种三值逻辑的定义，像是Kleene和Lukasiewicz的3值逻辑学($D=\{1\}$)、$LP$和$RM_3$($D$={1,i},其中i被当作同真同假)、$K_3$、$L_3$（$D$={1,i},其中i被当作非真非假），其中D表示命题结果的取值集合,它们均有自己的漏洞和不足。

第一个尝试将三值逻辑形式化的是Łukasiewicz。他的系统现在被称作是Łukasiewicz’s three-valued logic，被标记为“$L_3$”，第三个真值被称作是：“未决定的”或“可能的”。

Łukasiewicz认为未来的命题中应该包含第三个标记为$I$的真值，它既不是True也不是False。

$L_3$的语言包含合取($\wedge$)、析取($\vee$)、蕴含($\to_L$)、和否定($\sim$)。三值逻辑的真值表可表示如下：

| A | ~A |
|---|---|
| T | F |
| I | I |
| F | T |

| A | B | A ∧ B | A ∨ B | $A \to_L B$ |
|---|---|---|---|---|
| T | T | T | T | T |
| T | F | F | T | F |
| T | I | I | T | I |
| F | T | F | F | T |
| F | F | F | F | T |
| F | I | F | I | T |
| I | T | I | T | T |
| I | F | F | I | I |
| I | I | I | I | T |

需要注意的是，在经典逻辑中两个重要的法则：$A\vee\sim A$和$\sim(A\wedge\sim A)$在$K_3$下均不成立，事实上，它们的值均为$I$。

$L_3$的**Hilbert系统**如下所示,将$\wedge$和$\vee$用$\sim$和$\to_L$进行定义:$A \vee B =_{def} (A \to B) \to B$、$A \wedge B = def \sim (\sim A \vee \sim B)$ ,由此可得到以下公理和推论:

**公理**

$$(L1) A \to (B \to A)$$

$$(L2)(A \to B) \to ((B \to C) \to (A \to C))$$

$$(L3)((A \to \sim A) \to A) \to A$$

$$(L4)(\sim A \to \sim B) \to (B \to A)$$

**推论**

$$(MP) \vdash A, \vdash A \to B \Rightarrow \vdash B$$

Kleene还利用递归函数的理论提出了$K_3$这一种三值逻辑，$K_3$与$L_3$的不同之处在于$K_3$对于蕴含$\to_K$的解释，$K_3$的真值表如下:

| A | ~A |
|---|---|
| T | F |
| I | I |
| F | T |

| A | B | A ∧ B | A ∨ B | A $\to_K$ B |
|---|---|---|---|---|
| T | T | T | T | T |
| T | F | F | T | F |
| T | I | I | T | I |
| F | T | F | F | T |
| F | F | F | F | T |
| F | I | F | I | T |
| I | T | I | T | T |
| I | F | F | I | I |
| I | I | I | I | I |

可以看出$K_3$的真值表和$L_3$的真值表只有在A和B同为$I$时$A \to B$真值的不同，$L_3$中的为T，而$K_3$中的为$I$。

在$K_3$中，第三个真值被叫做：“undefined”，因此，$K_3$可以被应用在程序理论中。由于没有针对$K_3$的可重定义，因此没有一个$K_3$的 **Hilbert**系统。

$K_3$通常被称作 *Kleene's strong three-valued logic*，在其他文献中，*Kleene's strong three-valued logic* 也出现过，里面说如果任意的复合formula是$I$的话，那么这个formula就是$I$，Kleene的weak three-valued logic和Bochvar's three-valued logic是等价的。

**四值逻辑**适合于一个必须处理不完整和不一致信息的计算机系统，Belnap提出了一种四值逻辑，这种四值逻辑可以对计算机的内部状态进行形式化。在四值逻辑中有四种状态，分别是:$T$、$F$、$None$、$Both$，这四种状态作为一个计算机的输入，而基于这些输入状态，计算机可以计算出合适的输出。

---

(T)　　表示一个命题是true

(F)　　表示一个命题是false

(N)　　表示一个命题既不是true也不是false

(B)　　表示一个命题既是true也是false

---

在这里，(N)和(B)分别表示(None)和(Both)。可以看出，$(N)$对应于 **不完整性**（**incompleteness**），而$(B)$对应于**不一致性**（**inconsistency**）。因此，四值逻辑可以看作是三值逻辑的一个自然扩展。

事实上，Belnap的四值逻辑可以对不完整的信息(N)和不一致的信息(B)进行建模，他提出了两种四值逻辑，分别是$A4$和$L4$。其中，$A4$只可以处理原子命题，而$L4$可以处理更加复杂的复合命题公式。

$A4$基于如右图所示的 *approximation lattice*，

$$\begin{array}{ccccc} & & B & & \\ & / & & \backslash & \\ T & & \mathbf{A4} & & F \\ & \backslash & & / & \\ & & N & & \end{array}$$

，在这里，B是最小上边界，N是最大下边界。

$L4$基于如右图所示的 *Logical lattice*，

$$\begin{array}{ccccc} & & T & & \\ & / & & \backslash & \\ N & & \mathbf{L4} & & B \\ & \backslash & & / & \\ & & F & & \end{array}$$

。

$L4$使用$\sim, \wedge, \vee$逻辑符号并且基于$4 = \{T, F, N, B\}$的真值集合，$L4$的一个特征是逻辑符号的单调性。

令$f$是一个逻辑函数，当且仅当$a \subseteq b \Rightarrow f(a) \subseteq f(b)$时，$f$是单调的。同时，为了确保合取和析取的单调性，它们必须满足如下定理:

$$a \wedge b = a \Leftrightarrow a \vee b = b$$

$$a \wedge b = b \Leftrightarrow a \vee b = a$$

$L4$的真值表如下：

| | N | F | T | B |
|---|---|---|---|---|
| ~ | B | T | F | N |

| ∧ | N | F | T | B |
|---|---|---|---|---|
| N | N | F | N | F |
| F | F | F | F | F |
| T | N | F | T | B |
| B | F | F | B | B |

| ∨ | N | F | T | B |
|---|---|---|---|---|
| N | N | N | T | T |
| F | N | F | T | B |
| T | T | T | T | T |
| B | T | B | T | B |

Belnap为$L4$针对$\{T, F, N, B\}$进行了如下定义,其中$s$表示一个映射，它的输入域是所有的命题集合，包括A、B、(A ∧ B)、(A ∨ B)、～A等（原子命题或原子命题的组合），它的输出域是一个定义好的集合，在这里是$\{T, F, N, B\}$。

于是针对所有的$s$，有如下公式:

$$s(A \wedge B) = s(A) \wedge s(B)$$

$$s(A \vee B) = s(A) \vee s(B)$$

$$s(\sim A) = \sim s(A)$$

进一步地，Belnap用如下方法定义$\rightarrow$:

$$A \rightarrow B \Leftrightarrow s(A) \leq s(B)$$

$\rightarrow$可以被公理化为以下内容:

$$(A_1 \wedge \ldots \wedge A_m) \rightarrow (B_1 \vee \ldots \vee B_n)(A_i \ shares \ some \ B_j)$$

$$(A \vee B) \rightarrow C \leftrightarrow (A \rightarrow C) \ and \ (B \rightarrow C)$$

$$A \to B \Leftrightarrow \sim B \to \sim A$$

$$A \vee B \leftrightarrow B \vee A, A \wedge B \leftrightarrow B \wedge A$$

$$A \vee (B \vee C) \leftrightarrow (A \vee B) \vee C$$

$$A \wedge (B \wedge C) \leftrightarrow (A \wedge B) \wedge C$$

$$A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$$

$$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$$

$$(B \vee C) \wedge A \leftrightarrow (B \wedge A) \vee (C \wedge A)$$

$$(B \wedge C) \vee A \leftrightarrow (B \vee A) \wedge (C \vee A)$$

$$\sim\sim A \leftrightarrow A$$

$$\sim (A \wedge B) \leftrightarrow \sim A \vee \sim B, \sim (A \vee B) \leftrightarrow \sim A \wedge \sim B$$

$$A \to B, B \to C \Leftrightarrow A \to C$$

$$A \leftrightarrow B, B \leftrightarrow C \Leftrightarrow A \leftrightarrow C$$

$$A \to B \Leftrightarrow A \leftrightarrow (A \wedge B) \Leftrightarrow (A \vee B) \leftrightarrow B$$

注意到$(A \wedge \sim A) \to B$和 $A \to (B \vee \sim B)$ 不能用以上公理推导出来。事实上，Belnap的四值逻辑和 *tautological entailment*系统是等价的。

# *4 构造逻辑模型和具体事例*

在上学期学习《数据库系统及其应用》这门课的时候，我曾经学习到这样一个知识点:SQL中的bool类型的值有三种，分别是true、false和unknown，并且true AND unknown的结果是unknown，true OR unknown的结果是true，我当时就对为什么是这样的结果不解，也不太理解这个unknown这个真值有什么实际的用处。

现在学习到了多值逻辑，回想起这个问题，上网又查到 **不完全信息数据库**是数据库理论的一个重要研究方向，又回想起当时学习数据库 “关系代数”的时候，常常需要将SQL语句转换为关系代数等内部表示，查阅相关文献后，我打算利用三值逻辑构建一套逻辑语言，进行数据库中关系演算的模拟，这种模拟适用于不完全信息数据库。

下面为构造思路和过程，将此三值逻辑系统命名为**DS3（Database system 3-valued Logic）**，DS3可以处理有空值(NULL)参与的条件逻辑查询的表达式:

首先，定义关系数据库的条件查询结果可能出现的三种情况:

---

| | |
|---|---|
| T | 表示条件成立 |
| F | 表示条件不成立 |
| ? | 表示不能确定条件是否成立 |

---

然后，令$A$表示命题逻辑的连接词集合$\{\wedge, \vee, \sim, \to_D$，$\xi\}$，其中，$\xi$表示处理NULL空值的逻辑连接词。令 $W$表示所有真值的空间，即$\{T, F, ?\}$，令 $S$表示$W$的一个子集。

那么，DS3的逻辑结构可以定义为:$< W, S, \{\Psi_a; a \in A\} >$，其中$W$是$\{T, F, ?\}$，$\{\Psi_a; a \in A\}$表示对每一个逻辑连接词a，$\Psi_a$表示对应的真值函数，$S$代表指定值的集合：$\{T\}$，这个集合是我们认为命题成立时的结果取值。

针对$\Psi_a$函数的含义，举以下例子进行说明:

例如，$\Psi_{\sim}$表示一个一元的函数，其中$\Psi_{\sim}(0) = 1, \Psi_{\sim}(1) = 0, \Psi_{\sim}(?) = ?$；而$\Psi_{\wedge}$表示一个二元的函数，其中:

$$\Psi_{\wedge}(a, b) = \begin{cases} T & a = b = T \\ F & a = F \; or \; b = F \\ ? & otherwise \end{cases}$$

我们定义一个映射$f$，$f$表示从命题公式到$W$的一个映射，即输入域是全体命题公式，输出域是$W = \{T, F, ?\}$。那么我们可以利用$f$进行从全体命题公式到$W$的一个递归的映射(通过应用相应的连接词对应的函数)。举一个例子:

对于任意可能的$f$，有$f(x \wedge y) = \Psi_{\wedge}(f(x), f(y))$，其中

$$\Psi_{\wedge}(a, b) = \begin{cases} T & a = b = T \\ F & a = F \; or \; b = F \\ ? & otherwise \end{cases}$$

那么对于命题公式$x \wedge y$，我们以下几种可能的解释，它们的区别仅在于对原子命题$x$和$y$的取值上:

$$\begin{gathered} f_1(x) = F, f_1(y) = F \\ f_2(x) = T, f_2(y) = T \\ f_3(x) = F, f_3(y) = T, \\ f_4(x) = T, f_4(y) = F \\ f_5(x) = ?, f_5(y) = ? \\ f_6(x) = ?, f_6(y) = F \\ f_7(x) = ?, f_7(y) = T \\ f_8(x) = T, f_8(y) = ? \\ f_9(x) = F, f_9(y) = ? \end{gathered}$$

例如，$f_3(x \wedge y) = \Psi(f_3(x), f_3(y)) = f_{\wedge}(1, 0) = 0$。

进一步地，拓展到$n$值逻辑，有$f(a(P_1, \ldots, P_n)) = \Psi_a(f(P_1), \ldots f(P_n))$。其中$a$表示$n$元的逻辑连接词。

而针对 **语义推理**$P_1, P_2, P_3, \ldots, P_n \models A$，它是有效的，当且仅当不存在一个映射$f$使得$f(P_i) \in S$时，$f(A) \notin S$。即对于所有的$f(A) \in S$，$A$是逻辑真值。其中$S = \{ T \}$ 即为我们认为命题成立时的取值。**而通俗一点地说即为：前提成立时结论也成立。**下面列出针对命题连接词$\wedge, \vee, \sim, \rightarrow_D, \xi$的真值表:

| x | ~x |
|---|---|
| T | F |
| ? | ? |
| F | T |

| x | y | x ∧ y | x ∨ y | x →$_D$ y |
|---|---|---|---|---|
| T | T | T | T | T |
| T | F | F | T | F |
| T | ? | ? | T | ? |
| F | T | F | F | T |
| F | F | F | F | T |
| F | ? | F | ? | T |
| ? | T | ? | T | T |
| ? | F | F | ? | ? |
| ? | ? | ? | ? | ? |

| x | ξx |
|---|---|
| T | F |
| ? | T |
| F | F |

以上就是我对基于$\wedge, \vee, \sim, \rightarrow_D, \xi$三值逻辑谓词演算系统DS3系统的构建思路和过程，现用该系统解决一个小小的SQL查询语句的问题，我们仅针对SQL中条件表达式的结果进行计算:

给定三个命题分别为:$x=1, y=2, z=NULL$,现SQL查询语句中产生了如下条件表达式的求值需求:

$$((x=1) \rightarrow (z>x)) \vee (x>z) \wedge (x<y)$$

我们利用上述定义系统进行如下运算:

$$\begin{gathered} f(((x=1) \rightarrow (z>x)) \vee (x>z) \wedge (x<y)) \\ = \Psi_{\wedge}(f(((x=1) \rightarrow (z>x)) \vee (x>z)), f(x<y)) \\ = \Psi_{\wedge}(\Psi_{\vee}(f(x=1 \rightarrow z>x), f(x>z)), f(x<y)) \\ = \Psi_{\wedge}(\Psi_{\vee}( \quad \Psi_{\rightarrow}(f(x=1), f(z>x)), \quad f(x>z)), f(x<y)) \end{gathered}$$

现映射f的输入域全部变为原子命题，可以结束递归，将真值表中的值带入计算:

$$\begin{gathered} = \Psi_{\wedge}(\Psi_{\vee}(\Psi_{\rightarrow}(T, ?), ?), T) \\ = \Psi_{\wedge}(\Psi_{\vee}(?, ?), T) \\ = \Psi_{\wedge}(?, T) \\ =? \end{gathered}$$

因此，条件表达式最终的结果是?，即不能判断该表达式是真是假。

在数据库的查询优化中，常常需要将$SQL$语句转化为关系代数，上述构造了一个可以处理有$NULL$值参与条件表达式的三值谓词逻辑演算系统：$DS3$，使用该系统，在进行$SQL$语句的查询时，可以先将查询需求形式化，然后利用上述方法进行条件表达式真假值的判断，由于引入了？这一真值，我觉得使用$DS3$进行形式化演算的方法对研究不完全信息数据库具有重要的作用。

# *6 附录*


[1]郭贵春, 崔帅. 非经典逻辑的本质及其意义[J]. 江海学刊(南京), 2016, (3):51-57.

[2]余静.传统逻辑、经典逻辑与非经典逻辑[J].湘潭师范学院学报(社会科学版),1991,(1): 42-46

[3]高航.一个非经典逻辑群体:哲学逻辑[J].文化学刊,2017,(1): 144-147

[4]季秋,王万森,王新. 人工智能中逻辑学的研究[C]//.逻辑与认知学术研讨会会议论文集.[出版者不详],2004:24-28.

[5]ganò, L. (2000). Labelled Natural Deduction Systems for Propositional Non-Classical Logics. In: Labelled Non-Classical Logics. Springer, Boston, MA. https://doi.org/10.1007/978-1-4757-3208-5_3

[6]Bacon, A. Non-classical Metatheory for Non-classical Logics. *J Philos Logic* **42**, 335–355 (2013). https://doi.org/10.1007/s10992-012-9223-9

[7]Akama, S., Murai, T., & Kudo, Y. (2018). Non-classical logics. *Intelligent Systems Reference Library, 142*, 51-84. https://doi.org/10.1007/978-3-319-72691-5_3

[8]张怀博. 多值逻辑的历史发展与哲学问题探究[D]. 湖北:华中科技大学,2014. DOI:10.7666/d.D612740.

[9]Priest, Graham. *An Introduction to Non-Classical Logic : From If to Is*, Cambridge University Press, 2008. *ProQuest Ebook Central*, https://ebookcentral.proquest.com/lib/ecnu/detail.action?docID=336084.