

## 第 3 章

# 认知逻辑理论

逻辑学是研究思维形式及其规律的科学,包括传统逻辑、辩证逻辑、数理逻辑等<sup>[79]</sup>。逻辑学能为其他学科提供逻辑分析、逻辑推理、逻辑论证、逻辑批判等工具,因此是一门工具性学科。学习逻辑学,就是要掌握思维的形式结构及其规律,并能熟练运用逻辑的符号对思维推理或论证过程用形式化方法表示和分析<sup>[80-81]</sup>。正如欧洲近代哲学的奠基人之一笛卡儿的思想所述,各门具体学科研究的是其相关的具体内容,而逻辑学是研究思维本身的形式法则。因此,逻辑学是一种客观性和普遍性的思想,无论以什么为研究对象,都可以抽象出它的思维形式。形式逻辑对思维的研究是从形式结构着手的,对思维内容它并不研究的,或者说它是透过形式来表现内容的<sup>[82]</sup>;而计算机模拟人的思维形式就是从人的思维的逻辑形式入手的,例如人工智能的显著特征之一就是逻辑推理。智能系统中主体知识库的表达、推理模式的建立就是一阶逻辑、直觉逻辑、模态逻辑、多值逻辑和模糊逻辑等的充分运用<sup>[83]</sup>,所以逻辑学理论是计算机科学的基础理论之一,计算机科学蓬勃地发展又反过来推动了逻辑学的进一步发展。随着逻辑学的不断发展,它的表达力也越来越强大,它的逻辑论证能力也不断扩展<sup>[84]</sup>。关于逻辑的论证性,有不少学者进行了研究<sup>[85,89]</sup>,论证逻辑是逻辑学中一个不可分割的部分,在论证逻辑真当中起着重要的作用<sup>[89-90]</sup>。近年来,逻辑学研究中产生的新理论与新方法在数学、语言学<sup>[90-91,86,92-93]</sup>、法学<sup>[94]</sup>、经济学<sup>[95]</sup>,尤其是在计算机科学及人工智能领域中<sup>[96-99]</sup>已有不少的应用。因此,逻辑学越来越被这些领域尤其是计算机科学和人工智能领域的研究者关注,越来越多的学者在研究数学、计算机科学的同时还研究逻辑学,或者逻辑学研究者研究逻辑学在这些领域的应用<sup>[100-103,97,99]</sup>。

作为模态逻辑的一个分支,动态认知逻辑是认知逻辑融合动态逻辑形成的。动态认知逻辑是知识、行为以及它们之间的相互关系的逻辑。它不是一种逻辑而是一整套逻辑,可用来描述理性主体系统中静态和动态的两方面信息。

本章主要参考文献[83,104]来介绍动态认知逻辑的理论基础。

### 3.1 命题逻辑

有“真”或“假”两个值的陈述句,或者说不是真就是假的陈述句称为命题。下面给出命题逻辑的概念。

**定义 3.1: 命题公式** 假设  $P$  是命题变元集(它的元素通常用  $p, q, r, p_1, p_2, p_3$  等来组成),最简单的基本命题语言可用公式  $\varphi$  表示为

$$\varphi ::= | p | \neg \varphi | \varphi \vee \varphi, \quad \text{其中 } p \in P$$

这个定义的含义是,一个公式可以由 3 种方式构成:元子命题  $p$  是一个公式,如果  $\varphi$  是一个命题公式, $\varphi$  的否定  $\neg \varphi$  和  $\varphi \vee \varphi$  也是公式。这是命题公式最简洁的表示,因为其他联结词均可由以上两个联结词直接或间接定义。也可以解释为命题公式由命题变元或命题变元和联结词构成。命题基本联结词有  $\neg$ 、 $\vee$ 、 $\wedge$ 、 $\rightarrow$  等,分别表示否定、析取、合取、蕴含。之所以用否定和析取就能表示所有基本的联结词,是因为

$$\varphi \wedge \psi = \neg(\neg \varphi \vee \neg \psi)$$

$$\varphi \rightarrow \psi = \neg \varphi \vee \psi$$

命题逻辑的语义是由命题公式的真值来定义的。命题逻辑不能完全表达认知信息,认知概念的处理方法与模态逻辑的模态词相似,所以认知逻辑被称为认知模态逻辑。

### 3.2 模态逻辑

模态逻辑是在命题逻辑或谓词逻辑中加入模态词形成的,根据对模态词的不同解释可形成认知逻辑、道义逻辑、时态逻辑、动态逻辑、可证性逻辑、信念逻辑等,这就使得它的表达力在大大增强。因此模态逻辑在很多行业都有着不同程度的应用,使逻辑学得到不断的丰富和发展。模态逻辑的最大特点是发展了关系语义<sup>[105]</sup>,关系语义指的是框架、模型、可满足性和有效性等概念,在社会科学、经济学、数学等领域有着成功的应用。同时还在可能世界中引入了程序,于是模态语言理论在计算机科学中得到了广泛的应用,成为计算机科学领域非常有用的工具。尤其是认知逻辑和时态逻辑,时态逻辑描述了随着时间变化中的逻辑<sup>[98]</sup>、公开宣告和认知行为引起知识在认知活动中的变化,因此动态认知逻辑表达了公开宣告等认知行为导致的知识变化的逻辑模型及其模型的变

化<sup>[106]</sup>。这与计算机程序中一个操作引起的各参与方拥有的信息的变化刚好吻合,于是计算机程序的模态逻辑描述成为可能。

模态逻辑是为了形式化模态概念在基本逻辑的基础上加入了模态算子即模态词而形成,最典型的是模态词“必然”(用“ $\square$ ”表示)和“可能”(用“ $\diamond$ ”表示)<sup>[107]</sup>。在命题逻辑的基础上加上模态词称为命题模态逻辑,在谓词逻辑的基础上加上模态词称为谓词模态逻辑。模态逻辑比命题逻辑、谓词逻辑的表达力强得多。

**定义 3.2: 命题模态公式** 假设  $P$  是命题变元集,最基本的模态语言  $\square$  可用 BNF(Backus-Naur form) 范式表示为

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \psi \mid \square \varphi \quad \text{其中 } p \in P$$

这个公式的意思是一个真的常量  $p$  是一个公式,一个命题变元是基本模态公式,如果  $\varphi$  是一个模态公式, $\neg \varphi$  的否定和  $\varphi$  的析取以及必然  $\varphi$  也是模态公式。这是模态公式的最为简洁的表示,因为

$$\begin{aligned} \varphi \wedge \psi &= \neg(\neg \varphi \vee \neg \psi) \\ \varphi \rightarrow \psi &= \neg \varphi \vee \psi \\ \diamond \varphi &= \neg \square \neg \varphi \end{aligned}$$

定义中的“ $\square$ ”和“ $\diamond$ ”在基本模态语言中表示必然和可能。狭义上讲,模态逻辑研究涉及“必然”和“可能”表达式使用的推理。然而,术语“模态逻辑”被更广泛地用于涵盖具有相似规则和各种不同符号的一系列逻辑。根据对“ $\square$ ”的解释不同形成不同的模态语言,最为著名的几种模态逻辑是:认知逻辑、道义逻辑、时态逻辑、信念逻辑<sup>[108]</sup>。

模态语言之间的相互区别在于模态词的不同。模态语言以一个模态词集(在基本模态语言中是 $\{\square, \diamond\}$ )加上命题变元集  $P$  为基础构成。如果一个模态词有两个元素 $\otimes$ 和 $\oplus$ ,使得“ $\otimes \varphi := \neg \oplus \neg \varphi$ ”,则称 $\otimes$ 和 $\oplus$ 互为对偶。显然, $\square$ 和 $\diamond$ 互为对偶,但是  $K_a$ (主体  $a$  知道)和  $B_a$ (主体  $a$  相信)之间没有对偶关系。模态语言的语义是用可能世界的语义学来解释的,如此构建的语义理论称为克里普克(Kripke)语义学。它有两个层面的解释:框架的和模型的。克里普克框架和模型是定义模态逻辑语义最常用的结构。下面定义两个概念。

**定义 3.3: 基本模态语言  $\square$  的框架** 基本模态语言  $\square$  的框架是一个二元组  $F = (W, R)$ ,其中  $W$  是一个非空的可能世界集, $R$  是  $W$  上的一个二元关系。 $W$  是可能世界  $w$  的集合,它的元素通常称为  $w, v, s, t$  等, $wRv$  或  $Rwv$  表示关系  $R$  在世界  $w, v$  之间成立。在世界  $W$  成立的元素可以用一个非空的原子命

题集  $P$  来描述,这些原子命题代表了相关世界的基本事实,因此任意一个可能世界  $w(w \in W)$  都可以看作一个非空的原子公式集。根据模态逻辑具体应用领域不同,  $W$  中的元素,它或者表示状态(state)或者是节点(node)或者是情景(situation)或者是点(point)等。  $W$  上的  $R$  是二元的,描述可能世界之间的可及关系。  $wRv$  这种关系在集合理论上写作:  $(w, v) \in R$ 。如果假定  $R$  至少包含一种关系,框架就是一个关系结构,  $W$  则是这个关系结构的域。

**定义 3.4: 基本模态语言  $\square$  的模型** 基本模态语言  $\square$  的模型是一个二元组  $\mathfrak{M} = (F, V)$ , 其中  $F$  是一个框架,  $V$  是一个函数,它指派给每个命题符号  $p$  成立的一个子集  $V(p)$ , 其中  $p \in P$ 。形式上来说,  $V$  是一个映射:  $p \rightarrow \mathcal{P}(W)$ , 其中  $\mathcal{P}(W)$  表示  $W$  的幂集。函数  $V$  被称为一个赋值。

给定模型  $\mathfrak{M} = (F, V)$ , 称  $\mathfrak{M}$  是以框架  $F$  为基础的模型, 或者  $F$  是模型  $\mathfrak{M}$  所依据的框架。从形式上看, 模型比框架多了一个赋值或者说模型是一个带有值函数的框架, 模型也可以看作一种关系结构, 即  $(W, R, V)$ 。定义模型是为了解释模态语言。假定将模型  $\mathfrak{M}$  中的一个世界标记为  $w$ , 一个点模型就可以写作:  $\mathfrak{M}, w$ 。

**定义 3.5: 模态语言  $\square$  的语义归纳** 令  $\mathfrak{M} = (W, R, V)$  是一个模型且  $w \in W$ , 将“公式  $\varphi$  在模型  $\mathfrak{M}$  中的世界  $w$  为真”, 记作  $\mathfrak{M}, w \models \varphi$ , 则模态语言  $\square$  的语义归纳定义如下。

- $\mathfrak{M}, w \models p$  当且仅当  $w \in V(p)$  ;
- $\mathfrak{M}, w \models \neg \varphi$  当且仅当  $\mathfrak{M}, w \not\models \varphi$  ;
- $\mathfrak{M}, w \models \varphi \vee \psi$  当且仅当  $\mathfrak{M}, w \models \varphi$  或者  $\mathfrak{M}, w \models \psi$  ;
- $\mathfrak{M}, w \models \square \varphi$  当且仅当对任意一个  $v \in W$ , 如果  $Rwv$ , 那么  $\mathfrak{M}, v \models \varphi$ 。

显然这与命题语言不同, 模态语言的模型是基于一个世界来描述一个公式的真, 对任意一个公式  $\varphi$  对一个世界来说或者为真或者为假, 因此有  $\mathfrak{M}, w \models \varphi$  或者  $\mathfrak{M}, w \not\models \varphi$ 。对于变元  $p$  的赋值定义扩展到任意一个公式  $\varphi$ , 有  $V(\varphi) = \{w \mid \mathfrak{M}, w \models \varphi\}$ , 意思是说任意一个公式或命题的值可以看作使它为真的世界的集合。“ $\mathfrak{M}, w \models \varphi$ ”也称为公式  $\varphi$  在模型  $\mathfrak{M}$  中的  $w$  世界是可满足的, “ $\mathfrak{M}, w \not\models \varphi$ ”称为公式  $\varphi$  在模型  $\mathfrak{M}$  中的  $w$  世界是不可满足的。关于  $\mathfrak{M}, w \models \square \varphi$ , 是说模型  $\mathfrak{M}$  中与世界  $w$  可及的所有世界  $v$  都满足  $\varphi$ , 所以  $\varphi$  必然为真。例如, 在一个克里普克模型  $\mathfrak{M} = (W, R, V)$  中, 给定  $W = \{w, v, s\}$ , 关系  $R$  有  $Rsv, Rsw$ 。  $\mathfrak{M}, s \not\models p$ , 如图 3.1 所示。

如图 3.1 所示, 模型  $\mathfrak{M}$  中的世界或者称为状态或称为点  $v, s, w$ , 用圆圈表

图 3.1 模型  $\mathfrak{M}$  中  $\mathfrak{M}, s \Box p, \mathfrak{M}, s \Diamond q$ 

示,在这些世界或状态或点成立的命题写在其中,例如状态  $v$  满足的命题是  $p$ 、 $q$  这两个命题,或者说这两个命题在世界  $v$  是真的,即  $\mathfrak{M}, v \models p$  和  $\mathfrak{M}, v \models q$ 。可及状态用箭头表示。状态  $s$  的可及状态是  $v$  和  $w$ ,  $\mathfrak{M}, s \Box p$ , 因为  $\mathfrak{M}, v \models p$  和  $\mathfrak{M}, w \models p$ 。  $\mathfrak{M}, s \Box q$ , 因为  $\mathfrak{M}, v \models q$  和  $\mathfrak{M}, w \models q$ 。但是,  $\mathfrak{M}, s \Diamond q$ , 因为存在一个  $v$ , 使得  $Rsv$ , 并且  $\mathfrak{M}, v \models q$ 。所以说在一个世界必然  $p$  为真依赖于它的可及世界中  $p$  是不是都为真。在一个世界可能  $p$  为真, 是存在一个与它可及的世界满足  $p$ 。在一个世界一个模态公式是否为真, 依赖于与它可及的其他世界的事实, 这是模态逻辑的一条重要的性质。

## 3.3 动态认知逻辑

### 3.3.1 认知逻辑概述

认知逻辑的发展和应用已经远远超越了哲学的起源, 广泛应用到经济学、语言学、计算机科学等相关的学科, 已成为这些领域重要的形式化工具<sup>[109]</sup>, 在计算机科学中已经大量应用到人工智能、分布式计算和计算机安全领域<sup>[110]</sup>。目前, 动态认知逻辑在哲学、社会学、计算机科学中有着广泛的应用<sup>[88,96,111-135]</sup>。国内有不少学者对其进行研究<sup>[88,90-91,106,111-117]</sup>。动态认知逻辑的发展也趋向于实际应用<sup>[145-148]</sup>。

动态认知逻辑是模态逻辑的一个重要分支。动态认知逻辑是动态逻辑与认知逻辑的结合, 它能表示这两方面的知识, 一是主体的知识, 二是主体的行为执行引起主体变化的知识。通过其克里普克(Kripke)语义学, 认知逻辑受到模态逻辑发展的影响很大。在认知逻辑中对认知概念的处理与逻辑模态词“必然”与“可能”的处理方法类似,  $\Box\varphi$  表示知道  $\varphi$ , 例如, 主体  $a$  知道  $\varphi$ , 写作“ $K_a\varphi$ ” (本书中用  $K$  表示知道算子, 不同的字体表示不同的含义)。有时也用  $\Box\varphi$  表示相信  $\varphi$ , 主体  $a$  相信  $\varphi$ , 写作: “ $B_a\varphi$ ” (这里的  $B$  表示相信算子)。  $K$  和  $B$  是认知逻辑中主要的认知算子, 认知逻辑主要的研究就是围绕这一特征进行的。在认知逻辑中, 主体  $a$  知道某一事实, 这一事实就被认为是真的, 形式化的表达就是:  $K_a\varphi \rightarrow \varphi$ 。而相信就不是这样了,  $B_a\varphi \rightarrow \varphi$  是假的。

认知逻辑系统采用逻辑系统 5 (自反性和欧性)。认知逻辑基本的表达知识

的逻辑是基于一个原子命题集  $P$  和一个有限主体集  $A$  上的,原子命题  $p, q, r$  等描述在现实世界中的事件的状态,例如,在计算机程序运行中的每一个状态。在后面的描述中, $p$  是  $P$  中一个任意的原子命题, $a$  是  $A$  中的一个任意主体。

**定义 3.6: 基本的认知语言** 令  $P$  是原子命题集, $A$  是主体集。多主体认知逻辑语言  $\mathcal{K}$  用 BNF 范式表示如下:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \psi \mid K_a \varphi$$

在这个公式中, $p \in P, a \in A$ 。这个公式的含义是,原子命题是公式,公式使用  $\neg$  (否定)、 $\vee$  (析取)或  $K_a$  (认知算子)构成复杂公式。根据定义, $(p \wedge \neg K_a K_a (p \wedge K_a \neg p))$ 和  $\neg K_a \neg (p \wedge K_a (q \wedge \neg K_a r))$ 是公式。另外,也会使用一些典型的元素来表示原子命题,例如, $q, r$  或  $p', q', r', p''$ 等主体也可以用其他的变量来表示,例如, $b, c$  和  $a', b'$ 等在后面的实例中,也用别的符号来表示主体,诸如  $S$  表示发送者, $R$  表示接收者等。通常采用标准缩写和惯例来表示公式,例如: $(\varphi \vee \psi) = \neg(\neg \varphi \wedge \neg \psi)$ ,符号  $\bigvee$  作为  $p \vee \neg p$  的简写和  $\neg$ 。而且, $(\varphi \rightarrow \psi) = (\neg \varphi \vee \psi)$ ;  $(\varphi \leftrightarrow \psi)$ 是双向蕴含的简写。

这里仅仅增加一个一元算子来对命题逻辑进行一个最为简单的扩展。对任意一个主体  $a, K_a \varphi$  表示主体  $a$  知道  $\varphi$ 。主体  $a$  可以是在博弈中的一个玩家、一个机器人、一个机器或者是一个过程或进程。在认知定义中,主体  $a$  不知道  $\neg \varphi$  表示为  $(\neg K_a \neg \varphi)$ 。这个表达有时也解释为  $\varphi$  是与  $a$  的知识是一致的,或主体  $a$  可能知道  $\varphi$ ,写作  $\hat{K}_a \varphi = \neg K_a \neg \varphi$ 。对  $A$  中的任意一个主体群  $B$ ,如果  $B$  中每一个主体都知道  $\varphi$ ,写作  $E_B \varphi$ 。因此,增加  $E_B$  这个算子对每个  $B \subseteq A$  有

$$E_B \varphi = \bigwedge_{b \in B} K_b \varphi$$

因此,形如  $p \wedge \neg K_a p$  表示  $p$  是真的而且主体  $a$  不知道它。公式  $(\neg K_a K_b p \wedge \neg K_a \neg K_b p)$ 表示主体  $a$  不知道是否  $b$  知道  $p$ 。形如  $K_a (p \rightarrow E_B p)$ 表示主体  $a$  知道如果  $p$  为真,那么在主体集  $B$  中的每一个主体知道  $p$ 。

对认知逻辑的语义解释是用克里普克模型。文献[149]中阐明克里普克的可能世界语义的哲学思想是“可能世界”是“世界可能呈现的状态或可能会采取的方式”。在克里普克模型中,可能世界或称为状态,可及关系是不可区分的关系。形如“ $K_a p$ ”主体  $a$  知道  $p$  用克里普克模型图表示  $a$  可及的状态  $w$  和  $w'$  中, $\varphi$  都为真,如图 3.2 所示。

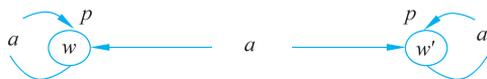


图 3.2 模型  $\mathcal{M}$  中,主体  $a$  知道  $p$

如图 3.2 所示,对主体  $a$ ,状态  $w$  和  $w'$  都是可及的( $R_a w w'$ ),表示  $a$  对状态  $w$  和  $w'$  不可区分,在  $w$  和  $w'$  中  $p$  为真,这里将在这个状态为真的公式标在状态圆圈上面。“ $K_a p$ ”主体  $a$  知道  $p$  也就是说主体  $a$  可及的所有世界里  $p$  都为真。在应用克里普克模型分析问题,把所有可能的情况看作一个世界或一个状态,具体用一个例子来说明。假如主体  $a$  住在上海,主体  $b$  住在北京, $a$  显然知道上海是否在下雨,而  $b$  是不清楚的。但是  $b$  知道  $a$  知道上海是在下雨或没有下雨。实际上  $a$  知道上海是在下雨。用  $p$  表示上海在下雨, $\neg p$  表示上海没有下雨。这两种情况用两个状态 1 和 0 来表示。事实状态用下画线标明。

规定在状态 1 中  $p$  为真,在状态 0 中  $p$  为假。这个模型如图 3.3 所示,状态 0 和 1 对  $b$  可及,也就是说  $b$  不能区分状态 0 和 1, $b$  不知道  $p$  的值。在认知模型中,所有的关系是等价关系,所有的状态是自反的对称的和传递的。所以  $b$  知道  $a$ 、知道  $p$ ,或者  $a$  知道  $\neg p$ 。 $a$  对这两个状态不可及,所以  $a$  对这两个状态是可以区分的,也就是说, $a$  知道  $p$  或者  $a$  知道  $\neg p$ 。事实上  $a$  知道  $p$ 。为了图形的简洁,常常省略自反和对称箭头,上面的模型精简为如图 3.4 所示的状态。

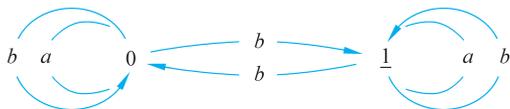


图 3.3 模型表明  $K_b(K_a p \vee K_a \neg p) \wedge \neg K_b p$

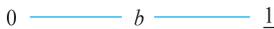


图 3.4 图 3.3 的精简

从上面的例子可以得出基本认知语言的语义如下。

**定义 3.7: 模态语言  $K$  的语义归纳** 给定主体集  $A$ , 原子命题集  $P$ , 认知模型为  $\mathfrak{M} = (W, R, V)$  且状态  $w \in W$ , 基本认知模态语言  $K$  的语义归纳定义如下:

- $\mathfrak{M}, w$                     当且仅当  $\mathfrak{M}, w$     ;
- $\mathfrak{M}, w \ p$                 当且仅当  $w \in V(p)$ ;
- $\mathfrak{M}, w \ \neg \varphi$             当且仅当  $\mathfrak{M}, w \ \varphi$ ;
- $\mathfrak{M}, w \ \varphi \vee \psi$         当且仅当  $\mathfrak{M}, w \ \varphi$  或者  $\mathfrak{M}, w \ \psi$ ;
- $\mathfrak{M}, w \ K_a \varphi$           当且仅当对任意一个  $w' \in W$ , 如果  $R_a w w'$ , 那么  $\mathfrak{M}, w'$   $\varphi$ 。

从这个语义可以看出,认知语言也是基于一个世界来解释的,认知公式  $K_a \varphi$  的真依赖于主体  $a$  的所有可及世界  $\varphi$  是否都为真。

在图 3.2 中,在认知模型  $\mathfrak{M}$  中,状态之间可及关系  $R_a$  是自反的(例如,任意一个  $w$ , 有  $R_a w w$ ),因此  $\mathfrak{M}$  满足  $K_a \varphi \rightarrow \varphi$ 。在认知模型  $\mathfrak{M}$  中,在一集合  $W$

上的关系  $R$  是一系列可及关系之一,对所有的  $a \in A, R_a$  有以下几种情况:

(1) 所有克里普克模型的类有时表示为  $\mathfrak{K}$ , 因此  $\varphi$  与  $\neg\varphi$  是一致的。

(2) 关系  $R_a$  是连续的,如果对所有的  $w \in W$ ,都存在一个  $w'$  使得  $R_a w w'$ 。在克里普克模型  $\mathfrak{M} = (W, R, V)$  中,如果所有的关系  $R_a$  是连续的,就可以表示为  $\mathfrak{K}_c$ 。

(3) 关系  $R_a$  是自反的,如果对所有的  $w \in W$ ,都有  $R_a w w$ 。在克里普克模型  $\mathfrak{M} = (W, R, V)$  中,如果所有的关系  $R_a$  是自反的,就可以表示为  $\mathfrak{K}_r$ 。

(4) 关系  $R_a$  是传递的,如果对所有的  $w, v, t \in W$ ,如果有  $R_a w v$  和  $R_a v t$ ,那么  $R_a w t$ 。在克里普克模型  $\mathfrak{M} = (W, R, V)$  中,如果所有的关系  $R_a$  是传递的,就可以表示为  $\mathfrak{K}_t$ 。自反且传递的模型类表示为  $\mathfrak{K}_{rt}$ 。

(5) 关系  $R_a$  是欧性的,如果对所有的  $w, v$  和  $t$ ,如果有  $R_a w v$  和  $R_a w t$ ,那么  $R_a v t$ 。在克里普克模型  $\mathfrak{M} = (W, R, V)$  中,如果所有的关系  $R_a$  是欧性的,就可以表示为  $\mathfrak{K}_e$ 。连续且传递和欧性的模型类表示为  $\mathfrak{K}_{cte}$ 。

(6)  $R_a$  是对称关系对所有的  $w, w'$ ,如果  $R_a w w'$ ,那么  $R_a w' w$ 。

(7)  $R_a$  是等价关系如果  $R_a$  是自反的传递的和对称的,称为等价关系。同样地,如果  $R_a$  是自反的和欧性的, $R_a$  也称为等价关系。具有等价关系的克里普克模型类表示为  $\mathfrak{K}_e$ 。在本书中的认知模型主要应用模型类  $\mathfrak{K}_e$ 。

在不影响任意主体知识的情况下,两个认知状态  $\mathfrak{M}, w$  和  $\mathfrak{M}', w'$  如何不同?换句话说,一个模型和另一个模型从哪个粒度上可以区分,认知语言  $\mathcal{L}$  如何去表达这个问题。下面引出一个基本概念。

**定义 3.8: 互模拟** 给定两个模型,  $\mathfrak{M} = (W, R, V)$  和  $\mathfrak{M}' = (W', R', V')$ , 一个非空的关系  $\subseteq W \times W'$  是一个互模拟,当且仅当,对所有的  $w \in W$  和  $w' \in W'$  有  $(w, w') \in \sim$ , 且满足下列条件:

① 若  $w \in V(p)$  当且仅当,  $w' \in V'(p)$ , 对所有的  $p \in P$ 。

② 对任意  $a \in A$  和所有的  $v \in W$ , 如果  $R_a w v$ , 那么, 存在一个  $v' \in W'$ , 使得  $R'_a w' v'$  和  $(v, v') \in \sim$ 。

③ 对任意  $a \in A$  和所有的  $v' \in W'$ , 如果  $R'_a w' v'$ , 那么, 存在一个  $v \in W$ , 使得  $R_a w v$  和  $(v, v') \in \sim$ 。

写作  $(\mathfrak{M}, w) \sim (\mathfrak{M}', w')$ , 当且仅当与  $w$  和  $w'$  相关联的  $\mathfrak{M}$  和  $\mathfrak{M}'$  存在互模拟, 称  $(\mathfrak{M}, w)$  和  $(\mathfrak{M}', w')$  互模拟。如果模型  $\mathfrak{M}$  和  $\mathfrak{M}'$  之间有一个互模拟关系相连接, 那么  $\mathfrak{M}$  和  $\mathfrak{M}'$  是互模拟, 写作  $\mathfrak{M} \sim \mathfrak{M}'$ 。

显然, 上面的 3 个条件是说, 对于  $(\mathfrak{M}, w)$  和  $(\mathfrak{M}', w')$  要互模拟, 在  $\mathfrak{M}, w$  和  $\mathfrak{M}', w'$  都要满足原子命题为真, 不仅是在  $w$  和  $w'$ , 而且在所有可及状态都是递

归的。第二个条件是说,保留从  $\mathfrak{M}, w$  到  $\mathfrak{M}', w'$  的认知公式,第三个条件是保留  $\mathfrak{M}', w'$  到  $\mathfrak{M}, w$  的知识。在认知语言中是不能区分互模拟模型,更精确地说,如果  $(\mathfrak{M}, w) \equiv_{\kappa} (\mathfrak{M}', w')$ , 当且仅当,  $\mathfrak{M}, w \models \varphi$ , 对所有的  $\varphi \in \mathcal{L}_{\kappa}$ , 有  $\mathfrak{M}', w' \models \varphi$ 。对所有的点模型  $\mathfrak{M}, w$  和  $\mathfrak{M}', w'$ , 如果  $(\mathfrak{M}, w) \preceq (\mathfrak{M}', w')$  那么  $(\mathfrak{M}, w) \equiv_{\kappa} (\mathfrak{M}', w')$ 。也就是说,如果这两个点模型是互模拟的那么它们在语言  $\mathcal{L}_{\kappa}$  上是等价的。

一个逻辑系统是一公式集,这一公式集在模型类上是有效的。这个公式集被称为公理系统,它是一个被称为公理的公式集合和推理规则构成。推理规则提供了从公理出发推演出系统内定理的工具。由公理推演出定理的过程被称作证明。因此,认知逻辑系统可以看作一个认知模态公式集,它包括所有命题重言式并且在 MP、US、和 N 规则下封闭。封闭的意思是说,在这个逻辑系统中其他公式可以从其推演。认知逻辑系统的公理如下:

$K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi)$  (分配公理)

$K_a\varphi \rightarrow \varphi$  (知道公理)

若  $\varphi$ , 则  $K_a\varphi$  (必然性规则, 又称 N 规则)

若  $\varphi$  和  $\varphi \rightarrow \psi$ , 则  $\psi$  (MP 规则)

$K_a\varphi \rightarrow K_a K_a\varphi$  (正内省公理)

$\neg K_a\varphi \rightarrow K_a \neg K_a\varphi$  (负内省公理)

分配公理是说主体知道所知道东西的所有逻辑后承。也就是说,在该蕴含式中知道算子可以进行分配的。知道公理是说,主体知道  $\varphi$ , 那么  $\varphi$  为真。如果说主体知道  $\varphi$ , 而  $\varphi$  又不是事实,这就导致不一致。而事实也是如此,人们所知道的东西都必须是事实。N 规则是说如果主体知道一个逻辑系统中的所有可证公式,则就会知道系统内所有定理。

### 3.3.2 群体知识

前面已经提到了普遍知识“每一个人都知道”的概念,在这一节中,介绍一些为多主体系统的群体知识的概念。群体知识在多主体系统中有重要的作用,因为主体之间的影响和互动很多是建立在群体知识的基础之上的。知道知识可以直接由经验获得,也可以通过交流获得,群体知识主要由交流获得。这里主要强调后面的应用中要用到的公共知识。

前面介绍了普遍知识  $E_B\varphi$  表示对每个  $B \subseteq A$ ,  $B$  中每一个主体都知道  $\varphi$ 。但这并不说明每一个主体都知道别的主体也知道  $\varphi$ 。这个称为普遍知识,即团体每个成员都知道的知识。公共知识是不仅每个成员知道,而且每个成员都知

道群内的其他成员都知道这个知识。例如,交通规则“红灯停,绿灯行”是驾驶员的公共知识,就是说不仅每个驾驶员知道这条规则,并且每个驾驶员都知道每个驾驶员都知道每个驾驶员都知道这条规则,等等。假如有某个驾驶员虽然自己知道这条规则,但不知道是否别人都知道,或者说他怀疑可能有人不知道,那么这个驾驶员就会缺乏安全感。这意味着该规则能否真正成为公共知识是驾驶员们遵守规则安全驾驶的前提条件。所以公共知识是群知识的迭代,即

$$E_B E_B E_B \cdots \varphi = E_B^n \varphi, \quad n \geq 0$$

这里用算子  $C$  来表示公共知识,群体  $B$  的公共知识表示为

$$C_B \varphi = \bigwedge_{n=0}^{\infty} E_B^n \varphi$$

因此,可以给出含有公共知识的认知语言  $\mathcal{K}_C$  的定义。

**定义 3.9: 含有公共知识的认知语言  $\mathcal{K}_C$  的语法** 令  $P$  是原子命题集,  $A$  是主体集,  $B \subseteq A, a, b, c \in A$ , 语言  $\mathcal{K}_C$  为含有公共知识的多主体的认知逻辑语言, 用 BNF 表示为

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid K_a \varphi \mid C_B \varphi$$

有时,把  $B$  里面的小群体的公共知识表示为  $C_a \varphi, C_{ab} \varphi, C_{abc} \varphi, \dots$  因此,  $\mathcal{K}_C$  是用公共知识(Common Knowledge)扩展  $\mathcal{K}$  形成的。

**定义 3.10: 含有公共知识的认知语言  $\mathcal{K}_C$  的语义** 对原子命题集  $P$  和主体集  $A$  给定一个模型  $\mathfrak{M} = (W, R, V)$  且状态  $w \in W$ , 含有公共知识的认知语言  $\mathcal{K}_C$  的语义归纳定义如下:

- $\mathfrak{M}, w \models p$  当且仅当  $w \in V(p)$ ;
- $\mathfrak{M}, w \models \neg \varphi$  当且仅当  $\mathfrak{M}, w \not\models \varphi$ ;
- $\mathfrak{M}, w \models \varphi \vee \psi$  当且仅当  $\mathfrak{M}, w \models \varphi$  或者  $\mathfrak{M}, w \models \psi$ ;
- $\mathfrak{M}, w \models K_a \varphi$  当且仅当对任意一个  $w' \in W$ , 若  $R_a w w'$ , 则  $\mathfrak{M}, w' \models \varphi$ ;
- $\mathfrak{M}, w \models C_B \varphi$  当且仅当对任意一个  $w' \in W$ , 若  $R_B w w'$ , 则  $\mathfrak{M}, w' \models \varphi$ 。

### 3.3.3 公开宣告逻辑

动态认知逻辑是认知逻辑和动态逻辑相结合而形成的逻辑。公开宣告是动态认知逻辑的基础系统,在文献[150]中有详细的阐述。动态逻辑以模态逻辑为基础发展并形成。在计算机系统中,一个程序的调用和执行或者一个消息的发送是一个行动,一个模态可以表示一个行动,动态逻辑是一种多模态逻辑。因此,动态逻辑可以表示计算机中程序执行和程序调用。动态认知逻辑是一个

非常活跃的研究领域。它主要能够刻画由行动引起的主体知识变化,而且能够形式化这些变化的信息。这些知识和由行动引起的知识变化都要分别表示。近年来,认知逻辑在计算机以及人工智能和社会科学等许多学科领域中都有着不同程度的应用,主要表示群体知识和多主体之间的互动引起的认知信息的变化。这种互动性的认知活动存在两个方面:一方面是从初始状态获取信息,另一方面是获取信息后的行为产生新的状态。从知识的角度来说,又可以看作知识的处理和知识变化的处理,即从初始状态到最终状态的变化情况。知识的动态变化可以理解为“知识—行为—新状态—新知识”的过程,这使得认知逻辑和动态逻辑可因此而结合。目前的做法是用模态算子来表示知道和引起知识变化的行动,使“知道”和“行动”同在一个系统中存在并且互相影响,行动的对象可以是一个知识的命题,这就刻画了知识以及知识的动态变化。

公开宣告是日常信息交流的一种形式,类似于现实生活中的广播。在动态逻辑中,公开宣告被认为是一种行为,于是,公开宣告逻辑(public announcement logic, PAL)就是包含公开宣告的动态认知逻辑。对公开宣告逻辑的研究促进了动态认知逻辑的进一步发展。对于多主体之间的通信的系统,公共知识及其相关知识是必不可少的。公共知识结合行为模型更丰富了逻辑的表达性和完整性<sup>[151-152]</sup>。在用餐密码协议的验证中,还有学者把符号技术应用在知识逻辑的模型检测中<sup>[153]</sup>,为匿名广播提供了一种机制。下面介绍公开宣告逻辑的语法和语义。

如果某人诚实地对一群朋友说“一棵树上开着红色的花”。“一棵树上开着红色的花”这个事实就成为了他们的公共知识。这是对事实的一种宣告,宣告的事实就变成了公共知识。但这容易造成一种错误的直觉:不管此人宣告什么,所说的内容都会变成公共知识,但是这对某些认知命题并不成立。例如, $a$ 告诉 $b$ :“John 当选为美国总统了,可是你还不知道这个消息。” $a$ 说的这句话可以形式地表达为 $p \wedge \neg K_b p$ 。在 $a$ 说这句话前, $p \wedge \neg K_b p$ 是真的,但当 $a$ 说了后, $b$ 就知道了“John 当选为美国总统了”,也就是说在公开宣告之后, $p \wedge \neg K_b p$ 就变成了假命题。如果一个命题在公开宣告之后就变成了假命题,这个宣告就称为不成功的宣告。在后面的应用中,只讨论宣告一个事实后变成了公共知识的这种情况。下面给出公开宣告逻辑的语法。

**定义 3.11: 公开宣告逻辑  $\text{KC}[\Box]$  的语法** 给定一个原子命题集  $P$  和一个有限的主体集  $A$ , 公开宣告逻辑  $\text{KC}[\Box]$  用 BNF 表示为

$$\varphi ::= \mid p \mid \neg \varphi \mid \varphi \vee \psi \mid K_a \varphi \mid C_B \varphi \mid [\Box] \psi$$

其中, $B \subseteq A$  和  $a \in A$ ,  $p \in P$ ,  $\varphi$  和  $\psi$  表示两个变量以及可能不同或相同的两个公式。语言  $\text{KC}[\Box]$  在语言  $\text{KC}$  基础上增加一个符号“ $[\Box]$ ”,表示增加一个动态模态算子。

归纳构造  $[\varphi]\psi$  表示：宣告  $\varphi$  或用  $\varphi$  更新后， $\psi$  成立。对其他的动态现象来说，更新是一个更普遍的概念。这里的宣告总是意味着公开和真实的宣告。

公开宣告  $\varphi$  的作用是将认知状态限制到使  $\varphi$  成立的那些状态以及可及状态。宣告  $\varphi$  被看作带相应的动态模态算子  $[\varphi]$  的认知状态转换器。为了对动态算子的语义解释，需要增加一个句子  $|\varphi|_{\mathfrak{M}}$  表示模型  $\mathfrak{M}$  中满足  $\varphi$  的状态或点，形式化表示为  $|\varphi|_{\mathfrak{M}} = \{\omega \in (\mathfrak{M}) \mid \mathfrak{M}, \omega \models \varphi\}$ ，其中  $\omega$  表示域。

**定义 3.12：含有公共知识的公开宣告逻辑  $\text{KC}[\Box]$  的语义** 对原子命题  $P$  和主体集  $A$ ，给定一个模型  $\mathfrak{M} = (W, R, V)$  且状态  $\omega \in W$ ，含有公共知识的公开宣告逻辑  $\text{KC}[\Box]$  的语义归纳定义如下：

- $\mathfrak{M}, \omega \models p$  当且仅当  $\omega \in V(p)$ ；
- $\mathfrak{M}, \omega \models \neg \varphi$  当且仅当  $\mathfrak{M}, \omega \not\models \varphi$ ；
- $\mathfrak{M}, \omega \models \varphi \vee \psi$  当且仅当  $\mathfrak{M}, \omega \models \varphi$  或者  $\mathfrak{M}, \omega \models \psi$ ；
- $\mathfrak{M}, \omega \models K_a \varphi$  当且仅当对任意一个  $\omega' \in W$ ，若  $R_a \omega \omega'$ ，则  $\mathfrak{M}, \omega' \models \varphi$ ；
- $\mathfrak{M}, \omega \models C_B \varphi$  当且仅当对任意一个  $\omega' \in W$ ，若  $R_B \omega \omega'$ ，则  $\mathfrak{M}, \omega' \models \varphi$ ；
- $\mathfrak{M}, \omega \models [\varphi]\psi$  当且仅当若  $\mathfrak{M}, \omega \models \varphi$ ，则  $\mathfrak{M}|\varphi, \omega \models \psi$ 。

其中， $\mathfrak{M}|\varphi = (W', R', V')$  定义如下：

- $W' = |\varphi|_{\mathfrak{M}}$
- $R' = R \cap (|\varphi|_{\mathfrak{M}} \times |\varphi|_{\mathfrak{M}})$
- $V'(p) = V(p) \cap |\varphi|_{\mathfrak{M}}$
- $\mathfrak{M}, \omega \models \langle \varphi \rangle \psi$  当且仅当  $\mathfrak{M}, \omega \models \varphi$  且  $\mathfrak{M}|\varphi, \omega \models \psi$

从上面的定义可以看到，表示公开宣告后，主体知识的变化是在模型中把认知状态限制到  $\varphi$  成立的那些状态去，而且保留这些状态已有的认知可及关系。

这里用 3 个玩家玩卡片的游戏来说明公开宣告如何产生模型的变化。甲、乙和丙分别从 3 张卡片当中抽取一张卡，每人只知道自己的卡，不知道别人的卡，3 张卡片上的数字分别是 0、1、2，这是公共知识。事实上，甲有卡 0，乙有卡 1，丙有卡 2。甲现在说：“我没有卡 1”。

用 012 表示甲有卡 0，乙有卡 1，丙有卡 2。这副卡是大家都知道的，但玩家只能看到自己的卡，其他的玩家也只有一张卡。他们只知道自己的卡上的数字而且别人的卡的数字一定是和自己的不同的。换句话说，012 和 021 对于甲来说是一样的，也就是甲对这两个状态是不可区分的，因为他不知道乙是 1 还是 2，也不知道丙是 2 还是 1。同样，012 和 210 对于乙也是一样的。所以对所有

的玩家来说,总共有 6 种不同的情况。因为只知道自己的卡而产生的等价关系和玩家实际拥有什么卡的真实情况,可以得到认知状态(Hexa,012),其中 Hexa 表示模型名称,真实状态用下画线标明,如图 3.5 所示。

分别用  $a$ 、 $b$ 、 $c$  代表甲、乙、丙三人。图 3.5 中 012 是事实状态,021 是  $a$  认为的可能状态,因为认为乙可能是 1 或 2,丙或许是 2 或许是 1,这两个状态对于甲来说不可区分。乙只知道自己是卡 1,他认为甲可能是 0 也可能是 2,丙可能是 2 或 0,012 和 210 对于乙来说不可区分,乙认为如果甲是 2,那么甲对 210 和 201 不可区分,以此类推,共有 6 个状态。状态之间连线上的主体表示他们对这两个状态不可区分,也就是说,这两个状态对他们来说是等价的。为了图的简洁,用线代表了箭头,省略了自反的箭头。

用  $0_a$  表示甲有卡 0,  $1_b$  表示乙有卡 1,以此类推。认知公式  $K_a \neg (K_b 0_a \vee K_b 1_a \vee K_b 2_a)$  表示甲知道乙不知道他的卡。甲认为乙可能有卡 2,但实际上,乙有卡 1,形式化为  $1_b \wedge \hat{K}_a 2_b$ 。甲宣告“我没有卡 1 ( $\neg 1_a$ )。”这个宣告将把模型限制到使这个公式为真的状态上。那些认为  $1_a$  的状态以及与它相连的连线删去,模型变为如图 3.6 所示状态。

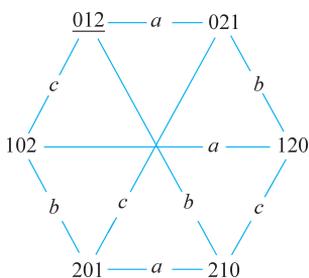


图 3.5 模型 Hexa 甲有卡 0,乙有卡 1,丙有卡 2 以及他们不可区分的状态

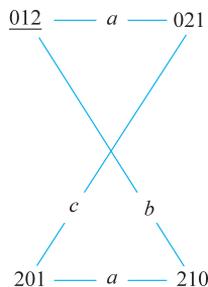


图 3.6 甲宣告“我没有卡 1”后变化后的模型

接下来乙说：“我还是不知道甲的卡号。”这句话可以形式化为  $(\neg (K_b 0_a \vee K_b 1_a \vee K_b 2_a))$ 。

所以模型仅存对于乙不可区分的状态。乙的宣告导致的认知模型变化为如图 3.7 所示的状态。

可以看出,乙这一宣告对于甲来说是很有启发性的,假如他现在知道乙的卡。不过,乙还是不知道他的。如果甲宣布他现在知道乙的卡,那就对  $K_a 1_b \vee K_a 2_b$  在状态 012 和 210 成立的就可以区分了。这毫无疑问,甲一旦知道乙的卡

也就知道丙的卡了。这一宣告无疑表明卡的真实状态是  $012$  (即  $0_a 1_b 2_c$ )，所以最后模型就剩下一个状态了，如图 3.8 所示。

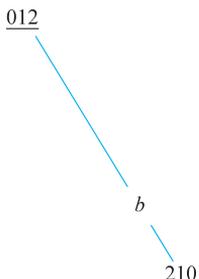


图 3.7 乙宣告“我不知道甲的卡号”后变化后的模型

012

图 3.8 甲宣告“我知道乙的卡号”后变化后的模型

这就没有进一步能宣告的信息了。整个模型的变化过程如图 3.9 所示。

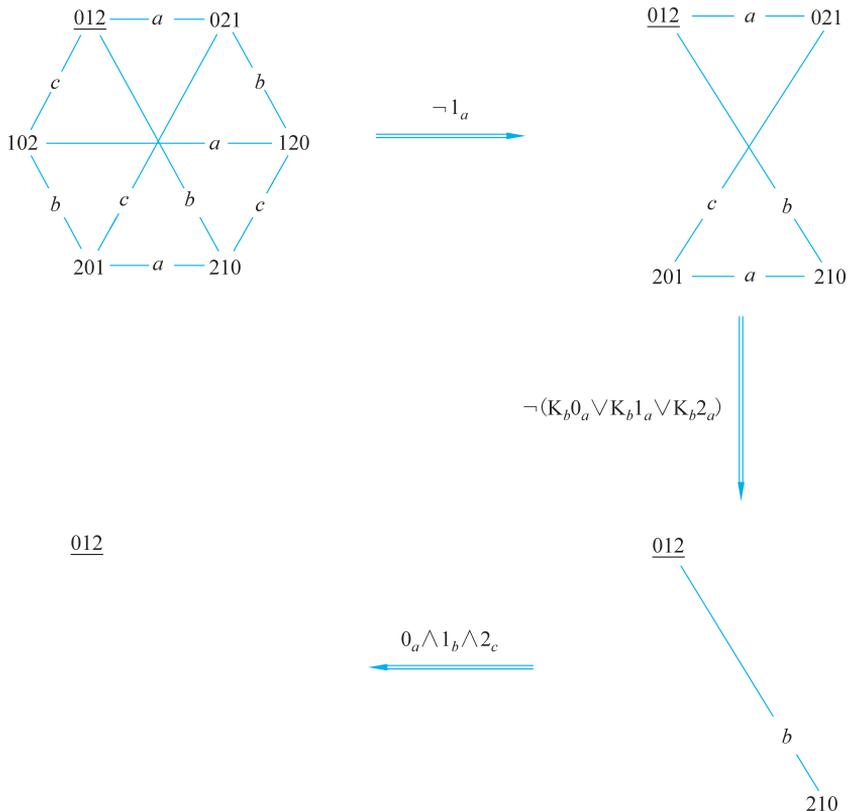


图 3.9 3 次宣告后模型的变化过程

如果把一个不安全的网络中的一个消息发送操作看作一个宣告,那么就可以应用这种分析方法表明参与的各个主体的知识的变化。

公开宣告的公理(符号“ $\leftrightarrow$ ”表示等价):

$$K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi) \quad (\text{分配公理})$$

$$K_a\varphi \rightarrow \varphi \quad (\text{知道公理})$$

$$\text{如果 } \varphi, \text{那么 } K_a\varphi \quad (\text{必然性规则,即 N 规则})$$

$$\text{如果 } \varphi \text{ 和 } \varphi \rightarrow \psi, \text{那么 } \psi \quad (\text{MP 规则})$$

$$K_a\varphi \rightarrow K_a K_a\varphi \quad (\text{正内省公理})$$

$$\neg K_a\varphi \rightarrow K_a \neg K_a\varphi \quad (\text{负内省公理})$$

$$[\varphi]p \leftrightarrow (\varphi \rightarrow p) \quad (\text{原子命题持久性})$$

$$[\varphi] \neg \psi \leftrightarrow (\varphi \rightarrow \neg [\varphi]\psi) \quad (\text{宣告与否定公式})$$

$$[\varphi](\psi \wedge \chi) \leftrightarrow ([\varphi]\psi \wedge [\varphi]\chi) \quad (\text{宣告与合取公式})$$

$$[\varphi]K_a\psi \leftrightarrow (\varphi \rightarrow K_a[\varphi]\psi) \quad (\text{宣告和知道})$$

$$[\varphi][\psi]\chi \leftrightarrow [\varphi \wedge [\varphi]\psi]\chi \quad (\text{宣告组合})$$

### 3.3.4 认知行为

3.3.3 节介绍了模型更新的一种方式——公开宣告。公开宣告作为一种认知行为对参与的所有主体传达了相同的信息并导致了模型域的限制。如果想要对不同的主体传达不同的信息就要采用更复杂的更新方式称之为认知行为(epistemic actions)。这种更新可能使模型域不变但是可及关系精化,也可能使模型域变大,甚至复杂度是由非互模拟状态的数量来衡量。

在公开宣告中,  $[\varphi]\psi$  表示公开宣告  $\varphi$  后,  $\psi$  成立。在认知行为逻辑  $[\varphi]\psi$  中  $\varphi$  是一个行为或动作,当  $\varphi$  发生,  $\psi$  成立。模型中每一个可能的行为构成一个状态。下面,用一个例子来说明。假定有网络中有 3 个主体  $a$ 、 $b$ 、 $c$ ,主体  $a$  发送一个消息给主体  $b$ ,  $c$  代表一个攻击者,  $b$  不确定他收到的消息是  $a$  发送的还是  $c$  发送的。假设消息为  $p$ ,  $S_a^b p$  表示主体  $a$  发送一个消息  $p$  给主体  $b$ 。这个过程中,  $a$  和  $c$  都知道是  $a$  做了这个动作,而  $b$  不知道消息来源于  $a$  还是  $c$ 。模型表示如图 3.10 所示。

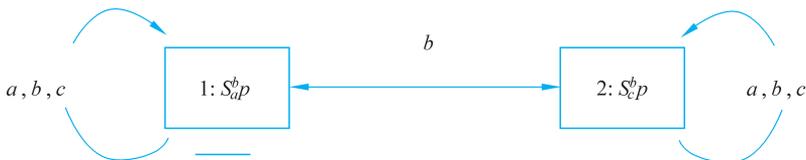


图 3.10  $a$  和  $c$  都知道是  $a$  发送消息  $p$  给  $b$ , 而  $b$  不知道消息来源于  $a$  还是  $c$

通常情况下,用方框表示可能发生的行为,用圆圈表示克里普克的世界或状态。用下画线表示事实行为。行为更新的克里普克模型是由世界或状态和行为成对构成。例如,图 3.10 中在状态 1,主体  $a$  发送一个消息  $p$  给主体  $b$ ,在状态 2,主体  $c$  发送一个消息  $p$  给主体  $b$ 。当然一个动作的发生还需要前提条件的,比如  $a$  发送一个消息  $p$  给主体  $b$ ,那么首先  $a$  要有消息  $p$ ,这就是他执行发送行为的前提条件。

根据上面的例子给出认知行为逻辑的语法和语义。

**定义 3.13: 认知行为逻辑语言  $\mathcal{L}_{\text{Act}}$  (Act 表示行为 Action) 的语法** 根据上面的例子,假定一个原子命题集  $P$  和一个有限的主体集  $A$ ,  $\alpha$  代表行为,认知行为逻辑  $\mathcal{L}_{\text{Act}}$  用 BNF(Backus-Naur form) 表示为

$$\begin{aligned} \varphi ::= & \quad | p | \neg \varphi | \varphi \vee \varphi | K_a \varphi | C_B \varphi | [\alpha] \psi \\ \alpha ::= & \quad S_a^b p | \alpha \vee \alpha | \alpha ; \alpha \end{aligned}$$

当  $B \subseteq A$  和  $a, b \in A, p \in P, S_a^b p \in \alpha$ 。  $S_a^b p$  是行为发送,  $\alpha \vee \alpha$  表示行为的选择,  $\alpha ; \alpha$  表示这两个行为顺序执行。

**定义 3.14: 认知行为逻辑语言  $\mathcal{L}_{\text{Act}}$  的语义** 对原子命题  $P$  和主体集  $A$ , 给定一个模型  $\mathfrak{M} = (W, R, V)$  且状态  $w \in W$ , 认知行为逻辑  $\mathcal{L}_{\text{Act}}$  的语义归纳定义如下:

$\mathfrak{M}, w$	当且仅当 $\mathfrak{M}, w \models p$ ;
$\mathfrak{M}, w \models p$	当且仅当 $w \in V(p)$ ;
$\mathfrak{M}, w \models \neg \varphi$	当且仅当 $\mathfrak{M}, w \not\models \varphi$ ;
$\mathfrak{M}, w \models \varphi \vee \psi$	当且仅当 $\mathfrak{M}, w \models \varphi$ 或者 $\mathfrak{M}, w \models \psi$ ;
$\mathfrak{M}, w \models K_a \varphi$	当且仅当对任意一个 $w' \in W$ , 如果 $R_a w w'$ , 那么 $\mathfrak{M}, w' \models \varphi$ ;
$\mathfrak{M}, w \models C_B \varphi$	当且仅当对任意一个 $w' \in W$ , 如果 $R_B w w'$ , 那么 $\mathfrak{M}, w' \models \varphi$ ;
$\mathfrak{M}, w \models [\alpha] \psi$	当且仅当对所有的 $\mathfrak{M}', w'$ , 如果 $(\mathfrak{M}, w) \models \alpha \mid (\mathfrak{M}', w')$ , 那么 $\mathfrak{M}', w' \models \psi$ ;
$\mathfrak{M}, w \models [S_a^b p] \psi$	当且仅当对所有的 $\mathfrak{M}', w'$ , 如果 $\mathfrak{M}, w \models \text{Pre}(S_a^b p)$ 和 $(\mathfrak{M}, w) \models S_a^b p \mid (\mathfrak{M}', w')$ , 那么 $\mathfrak{M}', w' \models \psi$ .

$$|\alpha \vee \beta| = |\alpha| \vee |\beta|$$

$$|\alpha ; \beta| = |\alpha| \circ |\beta|$$

其中参数含义如下:

$\text{Pre}(S_a^b p)$  表示执行动作  $S_a^b p$  的前提条件。  $|S_a^b p|$  表示动作  $S_a^b p$  可执行。

$(\mathfrak{M}, \omega) \mid S_a^b p$  表示动作  $S_a^b p$  在点  $(\mathfrak{M}, \omega)$  是可执行的。

$\mid \alpha \vee \beta$  表示动作  $\alpha$  和  $\beta$  之间的一个不确定的选择, 或  $\alpha$  被执行或  $\beta$  被执行。

$\mid \alpha; \beta$  表示动作  $\alpha$  与  $\beta$  顺序执行, “ $\circ$ ”表示顺序, 执行完  $\alpha$  后再执行  $\beta$ 。

### 3.3.5 行为模型

这一节引入“行为模型”的方法来描述认知行为。为了阐明这个方法, 用一个具体的例子来说明。假如有  $a$ 、 $b$ 、 $c$  3 个玩家玩掷硬币的游戏,  $c$  为裁判。 $c$  向空中抛一块硬币, 在空中抓住它并用手掌蒙住盖在桌上, 谁也没有看到硬币的哪一面朝上的。这时, 有两种可能的状态: 硬币的正面朝上(H), 硬币的背面向上(T), 用两个状态表示这个情况, 如图 3.11 所示。状态 1 满足硬币正面朝上, 状态 2 满足硬币背面向上。这个模型用 Toss 来表示, 有 Toss, 1 H, Toss, 2 T。

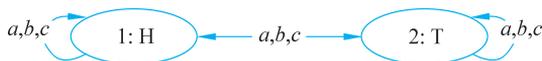


图 3.11  $a$ 、 $b$ 、 $c$  都不知道硬币的哪一面朝上

这时, 谁也不知道硬币究竟哪一面朝上, 所以每个参与者都对这两个状态可及, 这两个状态对每个参与者都是不可区分的。这时候,  $c$  忍不住偷偷松开手掌看了一下, 看到硬币正面朝上, 但是  $a$ 、 $b$  没有看见这一行为, 从而认为什么也没有发生, 用  $\sigma$  表示这个偷看行为, 用  $\tau$  来表示什么也没有发生, 这个行为模型用  $\langle \cdot, \sigma \rangle$  表示。满足这个行为前提条件的点称为行为点, 这个结构  $(\cdot, \sigma)$  称为行为模型, 如图 3.12 所示。

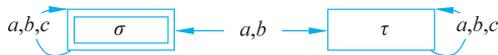


图 3.12  $c$  偷看了硬币,  $a$ 、 $b$  以为什么也没有发生, 行为模型  $(\cdot, \sigma)$

只有  $c$  才知道  $\sigma$  发生了,  $a$ 、 $b$  不能区分  $\sigma$  和  $\tau$ , 用双重的矩形来表示真实的行为。显然,  $c$  偷看到硬币朝上, 这个行为只能在满足 H 的状态发生, 也就是硬币朝上的那个状态, 不满足 H 的状态是不可能发生这一行为的。表示在行为  $\sigma$  后, 模型更新如图 3.13 所示。

也就是说, 只有在状态 1 才能发生  $\sigma$ , 在状态 1 和 2 都可以发生  $\tau$ 。

这个更新过程如图 3.14 所示。这个更新是积更新, 这个偷看行为只有  $c$  才知道,  $a$  和  $b$  都不知道, 并且只能在状态 1 发生, 行为  $\sigma$  需要一个前提条件 H, 只

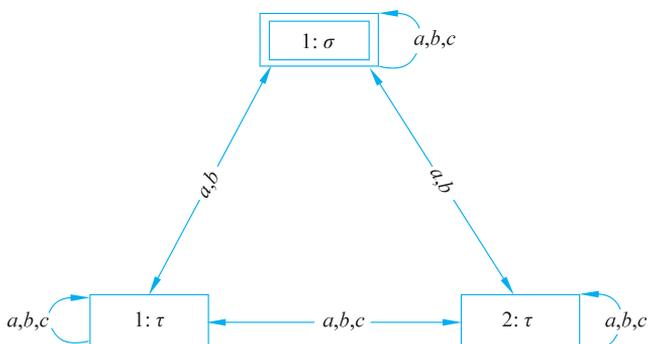


图 3.13 行为  $\sigma$  发生后的结果模型

有状态 1 满足, 即  $\text{Pre}(\sigma) = H$ 。这个更新模型称为行为模型, 这两个行为加上各自的前提条件, 描述了认知行为。

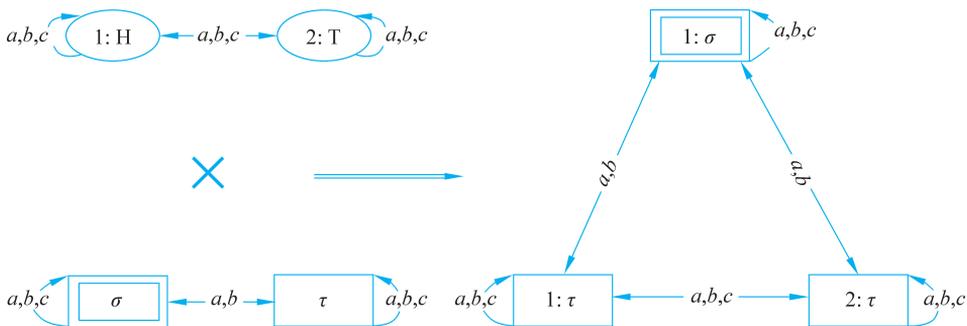


图 3.14 行为更新过程

图 3.14 展示了从认知状态的初始结构和行为模型的结构到结果认知状态的结构。在初始状态时,  $a, b$  不能区分状态 1 和 2, 也就是不能区分 H 和 T 哪一个为真, 也不能区分认知行为  $\sigma$  和  $\tau$ 。也就是说, 不能区分两个状态和两个行为, 即一个行为在一个状态被执行, 另一个行为在另一个状态执行, 而在结果状态仍然不能区分。行为  $\sigma$  在状态 1 执行, 行为  $\tau$  在状态 1 和 2 都可以执行。 $a, b$  仍然不能区分。尽管  $c$  在初始状态不能区分状态 1 和 2, 但在结果状态, 是能够区分的。也就是说, 虽然在初始状态不能区分这两个状态, 但是只要能区分这两个行为, 这两个行为的结果也是能够区分的, 即在结果模型中  $(1, \sigma)$  和  $(2, \tau)$  对  $a$  和  $b$  来说是一样的, 表示为  $(1, \sigma) \sim_{ab} (2, \tau)$ , 因为在初始状态 1 和 2 对  $a$  和  $b$  来说不可区分  $1 \sim_{ab} 2$ , 行为模型中,  $\sigma$  和  $\tau$  对  $a$  和  $b$  来说不可区分  $\sigma \sim_{ab} \tau$ 。

总结起来说：两个结果状态对一个主体不可区分，当且仅当，它们是在两个已经不可区分的状态中发生的两个不可区分的行为的结果。形式化为对任意的主体  $a$ ，有  $(1, \sigma) \sim_a (2, \tau)$ ，当且仅当  $1 \sim_a 2$  和  $\sigma \sim_a \tau$ 。

说明：这个对  $(1, \sigma)$  的成立表明  $\sigma$  在状态 1 能够被执行，仅当在初始模型中状态 1 满足执行  $\sigma$  的前提条件，即  $\text{Toss}, 1 \text{ Pre}(\sigma)$ 。

这个构造被看作一个认知状态和一个行为模型的限制模态积。两个模态结构的模态积，通过它们的域的笛卡儿积或者它们的一部分完成一些计算并保留在这些结构上的信息编码，见上例。这个积是受限制的，这是因为没有做完全的笛卡儿积，当  $\sigma$  在状态 1 能够被执行，仅仅才允许对  $(1, \sigma)$  成立。由于这个行为是认知行为，因此在这个对  $(1, \sigma)$  中的事实值是在原来状态 1 中成立的事实值，这个是不能改变的。显然，在结果结构中的指派点是由原始结构的指派点构成的对。模型  $(\text{Toss}, 1)$  描述为  $1 \sim_{ab} 2$ ，和行为模型  $(\ , \sigma)$  描述为  $\sigma \sim_{ab} \tau$ 。计算它们的限制模态积如图 3.13。这个笛卡儿积构成 3 个点，这是因为  $\text{Toss}, 1 \text{ H}, \text{Toss}, 2 \text{ T}$ ，即  $\text{Toss}, 1 \text{ Pre}(\sigma), \text{Toss}, 1 \text{ Pre}(\tau), \text{Toss}, 2 \text{ Pre}(\tau)$ 。尽管笛卡儿积是由 4 个点构成，然而这个限制模态积只有 3 个点： $(1, \sigma), (1, \tau), (2, \tau)$ ，因为  $\text{Toss}, 1 \text{ Pre}(\sigma), \text{Toss}, 1 \text{ Pre}(\tau), \text{Toss}, 2 \text{ Pre}(\tau)$ 。

**定义 3.15: 行为模型** 给定主体集  $A$  和原子命题集  $P$ ，令  $\mathcal{L}$  是任意的模态逻辑语言。一个  $\mathcal{L}$  行为模型  $\mathcal{M}$  是一个结构  $(W, \sim, \text{Pre})$ ， $W$  是行为的域或者说行为集，对每一个主体  $a \in A$ ， $\sim_a$  是在  $W$  上的等价关系，使得  $\text{Pre}: \mathcal{L} \rightarrow \mathcal{P}(W)$  是对每一个行为  $\delta \in \mathcal{L}$  的前提条件的函数，指派一个前提条件  $\text{Pre}(\delta) \in \mathcal{P}(W)$ 。一个指定的  $\mathcal{L}$  行为模型是一个结构  $(W, \delta) (\delta \in \mathcal{L})$  (在表示行为模型时常常省略  $\mathcal{L}$ )。

**定义 3.16: 行为模型逻辑的语法** 给定主体集  $A$  和原子命题集  $P$ ，令  $\mathcal{L}$  是任意的行为模型逻辑语言，用 BNF 表示为

$$\begin{aligned} \varphi &::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid K_a \varphi \mid C_B \varphi \mid [\alpha] \psi \\ \alpha &::= ( \ , \delta) \mid \alpha \vee \alpha \end{aligned}$$

当  $B \subseteq A$  和  $a \in A, p \in P, \delta \in \mathcal{L}$ ， $( \ , \delta)$  是点行为模型。 $\alpha \vee \alpha$  表示行为的选择。

**定义 3.17: 行为模型逻辑  $\mathcal{L}$  的语义** 对原子命题集  $P$  和主体集  $A$ ，给定一个模型  $\mathfrak{M} = (W, R, V)$  且状态  $w \in W$ ，有认知状态  $(\mathfrak{M}, w)$ ，行为模型  $\mathcal{M} = ( \ , \sim, \text{Pre})$ ， $\alpha$  是行为， $\varphi$  是公式。行为模型逻辑  $\mathcal{L}$  语义归纳定义如下：

$$\begin{aligned} \mathfrak{M}, w \models \varphi & \quad \text{当且仅当 } \mathfrak{M}, w \models \varphi ; \\ \mathfrak{M}, w \models p & \quad \text{当且仅当 } w \in V(p) ; \\ \mathfrak{M}, w \models \neg \varphi & \quad \text{当且仅当 } \mathfrak{M}, w \not\models \varphi ; \end{aligned}$$

$\mathfrak{M}, w \models \varphi \vee \psi$	当且仅当 $\mathfrak{M}, w \models \varphi$ 或者 $\mathfrak{M}, w \models \psi$ ;
$\mathfrak{M}, w \models K_a \varphi$	当且仅当对任意一个 $w' \in W$ , 如果 $R_a w \tau w'$ , 则 $\mathfrak{M}, w' \models \varphi$ ;
$\mathfrak{M}, w \models C_B \varphi$	当且仅当对任意一个 $w' \in W$ , 如果 $R_B w \tau w'$ , 则 $\mathfrak{M}, w' \models \varphi$ ;
$\mathfrak{M}, w \models [\alpha] \psi$	当且仅当对所有的 $\mathfrak{M}', w'$ , 如果 $(\mathfrak{M}, w) \mid \alpha \mid (\mathfrak{M}', w')$ , 则 $\mathfrak{M}', w' \models \psi$ ;
$(\mathfrak{M}, w) \mid \mathfrak{M}, \delta \mid (\mathfrak{M}', w')$	当且仅当 $\mathfrak{M}, w \models \text{Pre}(\delta)$ 和 $(\mathfrak{M}', w') = (\mathfrak{M} \otimes \mathfrak{M}', (\tau, \delta))$ ;

$$\mid \alpha \vee \beta \mid = \mid \alpha \mid \vee \mid \beta \mid$$

$\text{Pre}(\delta)$ 表示执行动作  $\delta$  的前提条件。 $\mid \alpha \mid$ 表示动作  $\alpha$  可执行。

在这里行为的执行是一个积更新,是一个点模型到点模型的映射。积更新后结果模型中的世界是世界与行为对。

$\mathfrak{M}' = (\mathfrak{M} \otimes \mathfrak{M}', (\tau, \delta))$ 是认知模型和行为模型的限制积,  $\mathfrak{M}' = (W', R', V')$ 有

$W' = \{(\tau, \delta) \mid \tau \in W, \delta \in \Sigma, \text{和 } \mathfrak{M}, \tau \models \text{Pre}(\delta)\}$ ;

$(\tau, \delta) \sim'_a (\tau', \delta')$  当且仅当  $\tau \sim_a \tau'$ 和  $\delta \sim_a \delta'$ ;

$(\tau, \delta) \in V'(p)$  当且仅当  $\tau \in V(p)$ 。

在结果模型中,一个主体不能区分两个世界,当且仅当在这之前它就不能区分这两个世界以及在这两个世界执行的两个不同的行为。

### 3.3.6 非单调逻辑

认知推理在计算机科学以及人工智能中有许多重要的应用。传统的推理,主要指演绎推理和完全归纳推理还有单调的模态逻辑推理等,它们都具有一条很重要的性质——单调性。在单调推理方法中,都是基于一个重要的假设:主体总是记得它之前的知识。推理的单调性是,增加任何新的前提都不会推翻已经得出的结论形式化为,若  $\Gamma \models \varphi$ , 则  $\Gamma, \psi \models \varphi$ 。这里  $\Gamma$  表示任意公式集,  $\psi, \varphi$  是任意公式,意思是说:若  $\Gamma$  能够推出  $\varphi$ , 当增加新的前提  $\psi$  时,结论  $\varphi$  仍然成立。推广而言,一旦一个结论得出,那么它总是有效的,即使增加新的信息。单调认知逻辑应用在密码协议中,表示当主体知道某消息后它将一直知道,不管后面它的信息增加多少它也不会丢失这些消息。在这种逻辑下,主体知道某知识就不会失去,也就是说,主体拥有的知识总是单调增加的。事实上,在现实生活中,主体的知识可能具有非单调性,例如在文献[84]中阐明了在一个论辩系统中可有可废止理论,那么就要采用非单调推理。由于主体的知识、接收的信息、拥