分 类 号: TP309　　密　级: 公开

论文编号: 2016010041

贵 州 大 学

2022届博士研究生学位论文

基于遗忘的反应式系统最弱充分条件研究

学科专业: 软件工程

研究方向: 软件工程技术与人工智能

导　　师: 王以松

研 究 生: 冯仁艳

中国·贵州·贵阳

2022年5月

目　录

表　格

插　图

摘　要

反应式系统是在对应用程序的即时响应（responsive）、回弹性（resilient）、弹性（elastic）以及消息驱动（message driven）要求的基础上产生和发展起来的不终止的系统。随着反应式系统越来越复杂，系统正确性和系统及其系统描述（规范，specification）之间的一致性越来越难以得到保证。模型检测是一个保证系统正确性行之有效的方法之一，此时反应式系统被表示为一个Kripke结构。然而在模型检测中，若系统不满足给定的规范（即与规范不一致），如何更新系统使其能够与规范一致是长时间以来的一个重要问题。与这个问题密切相关的两个概念是最强必要条件（the strongest necessary condition, SNC）和最弱充分条件（the weakest sufficient condition, WSC），其分别对应于形式化验证中的最强后件（the strongest post-condition, SP）和最弱前件（the weakest precondition, WP）。然而现有的方法不能计算像反应式系统这样不终止系统的SNC和WSC。此外，随着系统的更新和演化，现有的规范不可避免地会与新的知识相冲突。此时，如何将之前融入的元素在不影响其他信息的情况下“移除”也是个亟待解决的问题。

系统规范的描述语言以时序逻辑为主。其中计算树逻辑（Computation tree logic，CTL）是一种重要的分支时间时序逻辑，其具有模型检测能多项式时间完成的特性，因此被广泛用于系统规范描述中。但是CTL具有表达能力不够强的缺陷，μ-演算（μ-calculus）是一种比CTL表达能力更强但模型检测更加复杂的逻辑语言。尽管如此，CTL和μ-演算的语义都是Kripke语义，这与反应式系统被表示为Kripke结构有着直接的联系。因而，本文以这两种逻辑语言为研究背景，探索这两种语言下反应式系统上述问题的解决方法。

遗忘是一种知识抽取的技术，其被应用于信息隐藏、冲突解决和计算逻辑差等领域。本文从遗忘的角度出发解决上述提到的问题，主要研究成果如下：

1. 研究CTL下的遗忘的概念及其相关性质。首先，从模型在某个原子命题集合上互模拟的角度给出了遗忘的定义；其次，探讨并证明了遗忘算子的代数属性，包括模块性、交换性和同质性；第三，基于Zhang等人提出的四条公设，给出CTL遗忘的四条公设，表示定理表明这四条公设对遗忘是充分且必要的，即：遗忘的结果满足四条公设，且满足那四条公设的公式为遗忘的结果。

2. 提出一种基于归结的方法计算CTL下的遗忘。该方法使用Zhang等人提出的归结系统$R^{\succ,S}_{\mathrm{CTL}}$（clausal resolution calculus for CTL），在这个过程中需要将CTL公式转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句（separated normal form with global clauses for CTL）的集合，最后再将具有索引的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句转换为CTL公式。在这一过程中需要计算遗忘的公式总是和各个

过程的输出保持互模拟等价关系。

3. 给出了CTL下遗忘封闭的情形——约束的CTL。在这种情形下限制了公式的长度为n、公式所依赖的模型个数为有限个及构成公式的原子命题是有限的。此时，公式的模型可以用其特征公式——CTL公式来表示。因此，遗忘的结果可以由其所有模型在给定原子命题集合上的特征公式的析取来表示。

4. 研究了μ-演算下的遗忘。μ-演算是一种具有均匀插值（uniform interpolation）性质，证明了μ-演算下的遗忘与均匀插值是等价的，这意味着μ-演算下的遗忘是封闭的。此外，研究了μ-演算下遗忘的基本属性和复杂性，为均匀插值的研究提供了新的思路。

5. 给出了遗忘与WSC（SNC）和知识更新（knowledge update）的关系。WSC对模型的验证和修改具有重要作用，现有方法只能计算可终止模型的WSC，而像反应式系统这类不可终止的系统的WSC如何计算没有有效的方法。本文通过遗忘给出了计算WSC（SNC）的方法，利用遗忘定义了CTL和μ-演算下的知识更新，并证明其满足Katsuno等人提出的知识更新的八条公设。

6. 利用Prolog实现了2中提到的基于归结的计算CTL下的遗忘的方法，并做了相应的实验。从标准数据集和随机产生的数据集里做了两组实验，分别为计算遗忘和SNC。实验表明公式越长或遗忘的原子个数越多，效率越低；此外，在随机产生的公式的大部分情况下能计算出SNC。

这些结果为时序逻辑下的遗忘理论的研究提供了框架，并为模型更新提供了辅助工具——WSC。

关键词： 遗忘理论，最强必要条件，最弱充分条件，知识更新，人工智能

Abstract

With the software (hardware) systems of a computer becoming more and more complex, it gets hard to guarantee the correctness of systems and the consistency between systems and its specification. Model checking is a valid method to ensure the correctness of systems. However, it is an important problem in model checking to fix the system to make it consistent with its specification when the system does not satisfy the given specification. Moreover, in such a scenario, two logical notions introduced by E. Dijkstra are highly informative: the *strongest necessary condition* (SNC) and the *weakest sufficient condition* (WSC) of a given specification. These correspond to the *strongest post-condition* (SP) and the *weakest precondition* (WP) of such specification, respectively. Besides, the specification at hand is an unavoidable conflict with the new knowledge when the information of the system becomes clearer. In this case, another problem that needs to solve is to "*eliminate*" the containing elements without affecting the other information.

Forgetting, a technique to distill knowledge, was used to hide information, solve the conflict, compute logic differences, and so on. This paper solves the above problems from the point of forgetting. The major contributions are as follows:

1. Given the definition and properties of forgetting in CTL. First, this paper defines forgetting from the point of bisimulation over the given signature. Second, we explore the algebraic properties, including modularity, commutativity, and homogeneity, of forgetting. Third, the expression theorem shows that there is an "if and only if" relation between forgetting and the forgetting postulates proposed by Zhang et al., i.e., the result of forgetting satisfies the forgetting postulates, and the formula which satisfies the forgetting postulates is the result of forgetting.

2. Proposed a resolution-based method to compute the forgetting in CTL. This approach bases the resolution system proposed by Zhang et al., and the CTL formula is transformed into a set of $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ clauses (separated normal form with global clauses for CTL) at first. At the end of the approach, the $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ clauses are transformed into a CTL formula to obtain the forgetting result. It is noteworthy that the output of each process is bisimilar equivalent of the input CTL formula.

3. We outline a situation in which the forgetting is closed. In this case, the length of formulas are limited to integer n, and the number of atoms for formulas and Kripke structures is finite. To prove that it is closed, we define the characterizing formula (i.e., a CTL formula)

of the (finite) initial K-structure and show that each CTL formula is equivalent to a disjunction of the characterizing formulas of its models. This fact means that the result of forgetting some atoms from a CTL formula always exists.

4. Explored the forgetting in μ-calculus. μ-calculus is a kind of logic which have uniform interpolation. This paper shows that the uniform interpolation and forgetting in μ-calculus are equivalent, this means that the forgetting in μ-calculus is closed which is the biggest difference between μ-calculus and CTL. Moreover, the properties and complexity results related to forgetting are given, which proposes a new point for studying uniform interpolation.

5. Given the relation between forgetting and WSC (SNC or knowledge update). WSC is important to the verification and modification of the system. However， the existing methods can only compute the WSC of a terminable system, and the WSC can not be obtained in non-terminable systems (e.g., the reactive system). This paper shows how to compute the WSC of the reactive system and define the knowledge update using forgetting to satisfy the postulates proposed by Katsuno et al..

6. Implemented the algorithm proposed in 2, and some experiments are shown. Two experiments, computing forgetting, and SNC, were performed for standard and randomly generated datasets. Experiments show that the longer the formula is or the more atoms are forgotten, the lower the efficiency is. Moreover, SNC can be calculated in most cases of randomly generated formulas.

Its significance mainly provides a framework for the study of forgetting theory under temporal logic and provides an auxiliary tool for the model update.

Keywords: forgetting， the strongest necessary condition， the weakest sufficient condition， knowledge update

第一章　绪论

本章首先介绍研究的研究背景，分析保障系统正确的重要性，并阐述研究意义。其次，综合分析遗忘理论、最强必要（最弱充分）条件等关键技术的国内外研究动态，以及遗忘理论在形式化验证中的应用研究趋势。然后，围绕研究对象凝练出关键问题与目标。进一步，介绍本文的核心研究内容以及研究取得的主要成果。最后，给出具体章节组织安排。

1.1　研究背景与意义

1.1.1　研究背景

形式化验证是一种广泛应用在硬件[2-4]和软件系统中[5-6]的有别于测试的、采用数学方法证明系统满足给定特性的验证（verification）技术。软件和硬件的缺陷会导致严重的后果，如表1.1中列出的几个重大事件。近年来，为了减少系统（尤其是像火箭发射系统和卫星发射系统等关键领域的系统）错误带来的损失，形式化方法的研究与应用越来越受到人们的重视。包括INTEL、AMD、IBM、NVIDIA、CADENCE、Motorola、西门子和微软等在内的大型公司纷纷引入了形式化验证方法。与此同时，学术界也在形式化验证领域取得了突破性的成果，比如：剑桥大学进行了ARM6处理器的验证[7] 为类似于ARM这样的处理器提供了潜在的形式化验证方法，德国的Verisoft项目验证了一个一万行的操作系统内核[8]。

表 1.1: 由系统故障引起的重大事件概览

时间	事故原因	损失
1991年	美国爱国者导弹系统舍入错误	28名士兵死亡、100人受伤等
1996年	阿丽亚娜5型运载火箭因软件不同飞行条件下代码重用	其与其他卫星在瞬间毁灭
1999年	火星探测器用错度量单位	探测器坠毁并造成了3.27亿美元的损失
2011年	温州7.23动车信号设备在设计上存在严重的缺陷	动车脱节脱轨、多人失去生命

反应式系统是一种特殊的应用系统，它不终止且与环境有着持续不断的交互。其应用十分广泛：上至航空电子等安全攸关的领域，下至生活息息相关的汽车电子。必须在指定时间期限内完成对外部事件的检测和目标事件的响应是安全攸关反应式系统

的核心要求。因而，近年来包括形式化方法在内的多种方法用于确保反应式系的正确性。

形式化验证有两种主要的验证方法：自动定理证明（Automated theorem proving）和模型检测（Model checking)。在自动定理证明中，系统模型和规范（specification）被同一形式化语言分别描述为φ_{imp}和φ_{spec}，剩下的任务就是证明性质，如:“$\varphi_{imp} \rightarrow \varphi_{spec}$”或“$\varphi_{imp} \leftrightarrow \varphi_{spec}$”，即证明模型系统$\varphi_{imp}$是否满足给定的规范$\varphi_{spec}$。常用的自动定理证明方式有归结（Resolution）[9]和常用于模态逻辑的基于表推理（tableau）[10]的方法。计算机程序和系统验证（verification of computer programs and systems）是自动定理证明的新领域，它使用基于规则的方法来验证程序的正确性。然而，当要验证的程序为循环语句的时候，所需要的不变式（invariant）的获取是个困难的问题。因此，为了避免像Hoare逻辑[11]、动态逻辑[12]和分离逻辑[13]在形式化验证中寻找不变式问题，Fangzhen Lin提出将一个程序（program）转换为一阶理论，然后再使用一阶理论中的自动定理证明方法来验证[14]。

形式化验证的模型检测首先由Clarke提出，并用于解决并发系统验证问题[15]。Clarke和Emerson指出，在有限状态的并发系统中使用时态逻辑的推演系统（deductive system）中的公理和推理规则进行构造性证明（proof construction）的方法来证明该系统是否满足给定的规范是不必要的[16]。因为在有限状态并发系统中，该并发系统可以被看作是一个Kripke结构$\mathscr{M}$，与此同时，一个规范被表示成一个逻辑公式φ。此时，该验证问题就变成检验一个Kripke结构是否满足该公式，即模型检测（$\mathscr{M} \models \varphi$）：判断$\mathscr{M}$是否是$\varphi$的一个模型。

近年来，模型检测问题在知识表示与推理（KRR）领域的推进下取得了丰富的科研成果，例如：基于SAT的有界（bounded）模型检测[17]和基于OBDD的符号模型检测[18]已经使得模型检测问题在时间和空间效率上取得了很大的进步，在一定程度上缓解了其固有的状态空间爆炸问题。此外，大量优质的模型验证器，如：NuSMV[1]、SPIN[2]、Uppaal[3] 等，也相继发展起来，并且大部分的验证器都可以用来验证多种时态逻辑描述的公式。

时态逻辑作为描述系统规范的形式化语言，它研究状态随时间变化系统的逻辑特性。由于软件和硬件系统运行的本质是状态变化的过程，所以时态逻辑在软件程序验证和硬件验证中应用得相当广泛。计算树逻辑（Computation Tree Logic, CTL）是分支时态逻辑的一种，其模型检测是多项式时间可行的。然而，CTL表达系统性质的能力不如μ-演算（μ-calculus），如：“某给定的系统中存在一条路径使得该路径上的第偶数个状态满足特定的性质”这一规范是不能用其他时态逻辑表示的[19]。充分考虑这两种逻辑语言自身的特性，本文主要研究CTL和μ-演算。因此，本文所说的公式指CTL

[1]http://nusmv.fbk.eu/

[2]http://spinroot.com/spin/whatispin.html

[3]http://www.uppaal.org/

（或μ-演算）公式，即用来描述一个规范（或性质）的公式是CTL（或μ-演算）公式。

在实际应用中，对于给定的模型检测问题$\mathscr{M} \models \varphi$，当$\mathscr{M}$满足$\varphi$时一般的验证器都会返回“yes”以表示满足，当$\mathscr{M}$不满足$\varphi$时验证器会给出一个使得$\varphi$不被$\mathscr{M}$满足的负例。此时，使用什么样的信息对$\mathscr{M}$进行修正使得其满足给定的规范是一个重要的问题。

从知识抽取（或“消除”）的角度来看。出于安全考虑，查看信息时需要将有的信息隐藏而只抽取关注部分信息。此外，随着时间推移，由于某些原因使得系统的部分信息过时，就需要将这样的过时信息在不影响其他信息的情况下“消除”。考虑如下示例：

例 1.1 (汽车制造企业模型)**.** 一个汽车制造企业能够生产两种汽车：小轿车(se)和跑车(sp)。每隔一段时间，该企业都会做一个生产决策(d)，即：合理的生产计划。刚开始的时候，该企业做出了具有三个选择的方案：（1）先生产足够的se，然后在再产sp；（2）先生产足够的sp，然后再生产se；（3）同时生产se和sp。这一过程可以由图1.1中的Kripke结构（带标志的状态转换图）$\mathscr{M} = (S, R, L)$形式化地展现出来，其中:

- $V = \{d, s, se, sp\}$为该工厂所需要考虑的原子命题的集合；
- $S = \{s_0, s_1, s_2, s_3, s_4\}$为状态空间；
- $R = \{(s_0, s_1), (s_1, s_2), (s_1, s_3), (s_1, s_4), (s_2, s_0), (s_3, s_0), (s_4, s_0)\}$为状态转换关系的集合；
- $L : S \to 2^V$为标签函数，具体地：$L(s_0) = \{d\}$、$L(s_1) = \{s\}$、$L(s_2) = \{se\}$、$L(s_3) = \{sp\}$和$L(s_4) = \{se, sp\}$。

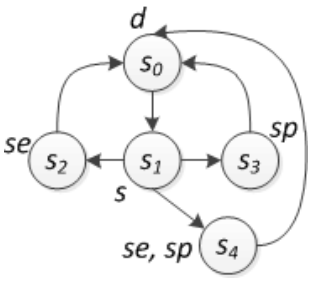


图 1.1: 汽车制造企业模型

假定现在由于经济危机或者是环境需要，导致该企业不能再生产跑车。这意味这所有关于sp的规范和Kripke结构都不再需要考虑，因此应该“移除”掉。

日常生活中也有很多上述例子中的场景，如：商业交易过程、软件开发过程等[20]。但是对于给定原子命题的集合，从这些大型系统（或规范）中“移除”掉一些原子而

保持与这些原子无关的性质是一个复杂的问题。此外，在这种情形下，两个重要的概念：最强必要条件（SNC）和最弱充分条件（WSC）问题也随之产生，其中SNC是指最一般的结论，WSC指最特殊的诱因。

基于上述存在的问题，本文在下面的研究意义部分给出对应的解决方案和其意义所在。

1.1.2 研究意义

最强后件（SP）和最弱前件（WP）（分别对应于上文提到的SNC和WSC）是形式化验证中的两个重要的概念，其不仅被用于汇编语言程序推理[21]和制定验证条件[22]，还被应用于形式化验证过程中的负例生成[23]和系统精化（refinement）[24]。当$\mathscr{M} \not\models \varphi$时，若知道某个性质$\psi$使得若$\mathscr{M}$按照此性质进行修改后得到的新模型$\mathscr{M}'$能满足$\varphi$，即$\psi$为使得$\mathscr{M} \models \varphi$成立的充分条件（$\mathscr{M} \models \psi \rightarrow \varphi$）。然而，现有的方法不能直接应用于计算当$\mathscr{M}$为不终止系统（如：反应式系统（reactive system））时使得“$\mathscr{M} \models \varphi$”为真的最强必要条件（SNC）和最弱充分条件（WSC）（其详细原因将在下文指出）。此时，探索在$\mathscr{M}$下使得φ满足的定义在某个符号集合上的*SNC*和*WSC*将更进一步完善模型检测问题，同时也为基于*WSC*的负例生成和精化提供了理论依据和新的计算方法。

本文将探索一种称为遗忘（forgetting）的方法来计算充分（必要）条件。正如下文将要说到的，遗忘理论作为知识表示与推理（KR）中的重要理论，具有较长的科研历史，且在许多逻辑中都有了较为成熟的研究。然而，在时序逻辑方面的研究目前还不成熟。因此，本文的研究将为时序逻辑下遗忘理论的研究提供一个理论框架。与此同时，借助遗忘理论计算上述形式化验证问题中的充分（必要）条件，这架起了*KR*与形式化验证的桥梁。

1.2 相关研究工作回顾

遗忘是一种重要的知识抽取工具，它具有均匀插值（uniform interpolation）和二阶量化消解（second-order quantifier elimination，SOQE）两种称谓。在很长一段时间内，遗忘被用于描述逻辑中本体（ontology）摘要的提取、敏感信息的隐藏和软件工程中计算两个文件的逻辑差（logical differences）。此外，其也被用于包括信念更新（belief update）、修改（repair）、规划（planning）和知识独立性的其他领域。

反应式系统是一种不终止的、与环境有着持续不断交互的系统，其应用广泛、种类繁多。本文探究如何使用遗忘计算反应式系统下的SNC（WSC），下面就与本文密切相关的反应式系统、遗忘理论和SNC（WSC）进行回顾。

1.2.1　反应式系统

随着应用程序在各个领域的应用，对其需求已经发生了重大变化，继而引起模式变化。以前，一个大型的应用程序通常需要数十台服务器，如此高的代价换来的效果和维护成本不成正比：秒级的响应时间，但需要数小时的维护时间。当前，移动和云端技术的成熟使应用程序不拘泥于以前的方式，被部署到了移动设备和具有多核心处理器的云端集群上。用户对应用程序的响应时间要求从秒级变到毫秒级，且具有100%的成功率，而数据量也从之前的GB级扩展到现在的PB级。原有的软件架构已经无法满足今天的需求。

要满足现有的要求，应用程序需要剧本以下特质：即时响应性（Responsive）、回弹性（Resilient）、弹性（Elastic）以及消息驱动（Message Driven）[4]。反应式系统是一种与环境有着持续不断交互的系统，具有不终止的性质[19]，它满足上述的几种要求，且从传统的静态模式逐步向动态、开放、自适应、服务化的软件系统演化[25]。反应式系统由来已久[26-27]，且种类繁多，如微处理器（microprocessor）、计算机操作系统（computer operating systems），航空交通管制系统（air traffic control systems）、车载电子设备（on-board avionics）以及其他嵌入式系统。

此外，反应式系统的应用非常广泛：上至航空电子等安全攸关的领域，下至生活息息相关的汽车电子。安全攸关反应式系统的核心要求是：必须在指定时间期限内完成对外部事件的检测和目标事件的响应，否则会产生灾难性的后果[28]。为了确保反应式系统的正确性和实时性，不同的方法被提出，包括：模型检查方法[19,29-30]、目标驱动的运行时需求建模框架[25]、基于图模型的实时规则调度方法（graph-based real-time rule scheduling,简称GBRRS）[28]、SPARDL[31]和选择性测试方法[32]等。

在上面的方法中，大都是验证（检测）给定的系统是否具有某种性质，而当系统不满足该性质时对如何寻找可靠的信息（充分条件）来对系统进行修改没有探讨。反应式系统是一种不终止的系统，通常被看作一个Kripke结构。基于此，本文探索如何计算该充分条件使得系统在该条件下满足给定的性质。

1.2.2　遗忘理论

遗忘这一词源于Lin等人关于一阶逻辑（first-order logic，FOL）的工作[33]，在此之前的研究中多提到的是均匀插值[34-35]和SOQE[36]。

在命题逻辑中（propositional logic，PL），从公式φ里遗忘掉一个原子命题p通常记为$Forget(\varphi,\{p\})$，得到的结果为$\varphi[p/\bot]\vee\varphi[p/\top]$（其与$\exists p\varphi$等价），其中$\varphi[X/Y]$为将$\varphi$中$X$的全部出现替换为$Y$得到的结果。从公式$\varphi$中遗忘掉有限的原子命题集合$P$被定

[4]https://www.reactivemanifesto.org/zh-CN

义如下：

$$Forget(\varphi,\emptyset)=\varphi,$$
$$Forget(\varphi,P\cup\{q\})=Forget(Forget(\varphi,\{q\}),P).$$

在FOL中，遗忘通常被看为SOQE问题的一个实例。特别地，从FOL公式φ中遗忘掉一个n-元谓词P得到的结果为一个二阶公式$\exists R\varphi[P/R]$[33]。从这个角度看来，遗忘就是找到一个与二阶公式$\exists R\varphi[P/R]$等价的一阶公式。然而，二阶逻辑的表达能力是严格大于一阶逻辑的，因而可以容易得出FOL下的遗忘不是封闭的，也就是说从有的一阶公式中遗忘掉某些谓词得到的结果不可以用一阶公式来表示。作为FOL的一个子类，描述逻辑公式的遗忘也不总是存在的[37]，甚至对最基本的描述逻辑ALC而言，遗忘的存在性问题都是不可判定的。尽管如此，描述逻辑作为一种在语义网领域很重要的语言，其子类（包括ALCOHI和ALCOIH）中的遗忘通常被用来抽取视图（review）[38-42]。

现有的计算一阶逻辑和描述逻辑遗忘的方法有基于归结（resolution）和基于Ackermann引理的方法[43]。其中基于归结的方法是一种基于子句的归结方法，归结规则是其基础。通常在这种方法中首先要把公式转换为其子句形式，然后再使用归结规则，最后将得到的子句集合中包含有要遗忘的谓词（原子命题）“移除”掉后得到的结果可能就为遗忘的结果（在后文中会详细介绍与本文相关的归结规则和转换规则）。基于Ackermman引理的方法主要是直接或间接（扩展）使用下面的Ackermann引理得到的。

引理 1.1 (引理6.1[43]). *给定关系变元X和一阶公式$\alpha(\bar{x},\bar{z})$和$\beta(X)$，其中$\bar{x}$和$\bar{z}$为普通变元构成的多元组、$\bar{x}$中变元的个数与X的参数个数相同、且α中不包括X。*

- *若$\beta(X)$关于X是正的，即：X在$\beta(X)$中的每次出现前面都有偶数个“$\neg$”符号，则：*
$$\exists X\{\forall\bar{x}[X(\bar{x})\rightarrow\alpha(\bar{x},\bar{z})]\wedge\beta(X)\}\equiv\beta(X)^{X(\bar{x})}_{\alpha(\bar{x},\bar{z})}。$$
- *若$\beta(X)$关于X是负的，即：X在$\beta(X)$中的每次出现前面都有奇数个“$\neg$”符号，则：*
$$\exists X\{\forall\bar{x}[\alpha(\bar{x},\bar{z})\rightarrow X(\bar{x})]\wedge\beta(X)\}\equiv\beta(X)^{X(\bar{x})}_{\alpha(\bar{x},\bar{z})}。$$

其中，$\beta(X)^{X(\bar{x})}_{\alpha(\bar{x},\bar{z})}$表示将$\beta(X)$中$X(\bar{x})$的全部出现用$\alpha(\bar{x},\bar{z})$来替换得到的公式。

知识遗忘（knowledge forgetting）在模态逻辑S5中首先被提出并被用于推理智能体的知识状态（知识或者信念）[44]。模态逻辑中的遗忘与经典逻辑下的遗忘不同，因为

模态逻辑系统中引入了模态词，此时就不能以简单的谓词（命题）替换的方式获取遗忘的结果，如：

例 1.2. [45] 令S5公式$\varphi = \mathrm{K}p \wedge \neg\mathrm{K}q \wedge \neg\mathrm{K}\neg q$，则如果使用命题逻辑下的计算方法得到的结果为$\varphi[q/\top] \vee \varphi[q/\bot] \equiv \bot$。这显然是不正确的，因为在遗忘$q$之后智能体的知识库不应该变得不一致。

为此，新的计算方法和四个能精确描述知识遗忘的基本公设被给出，这几个公设为：削弱（weaking）、正支持（positive persistence）、负支持（negative persistence）和无关性（irrelevance）。此外，模态谓词逻辑下信息不完备的知识遗忘[46]、模态一阶逻辑S5中的遗忘也得到了研究[47]，Fang等人也讨论了关于多模态（multi-modal）K_n、D_n、T_n、$K45_n$、$KD45_n$情形下遗忘的存在性[48-50]——这些逻辑里的遗忘总是存在的，其中n为智能体的个数。

均匀插值作为遗忘的一个对偶概念，这里有必要介绍一下模态逻辑系统的这一性质的研究现状。S5、K和KD模态逻辑系统具有均匀插值性质[51]，而一些模态逻辑系统没有均匀插值性质，如：模态逻辑量化的S5[52]和K4、和S4及其扩展都没有均匀插值性质[53]，因而其遗忘也不是封闭的。因此，在研究这些具有均匀插值性质的模态逻辑下的遗忘时可以借鉴S5系统下的遗忘方法，也可以参考K系统下的基于归结计算均匀插值的方法。对于那些没有均匀插值的模态逻辑系统可以考虑模态逻辑下的Ackermann引理[43]。

在非单调推理（non-monotonic reasoning）环境中，科研工作者们也从遗忘需要满足的基本条件的视角研究了基于回答集语义的逻辑程序的遗忘，这些工作包括Zhang、Wang等人发表在AI、AAAI和JAIR上的文章[54? -60]，Eiter、Gonccalves等人的综述[61-62]。

遗忘有很多应用，下面列出几点：

- 计算后继状态公理：在规划问题中，根据最强必要条件和最弱充分条件有利于求出后继状态公理[63]。
- 信息隐藏：在有的关键领域，为实现隐私保护，敏感信息必须被隐藏。现有的方法包括基于本体[37]和基于DataSecOps的个人信息保护[64]，要做到隐私保护，只需要将那些敏感的概念（concept）和角色（role）符号隐藏（遗忘）就行了。值得注意的是个人信息保护涉及被遗忘权[65];
- 知识更新：在许多场景，知识不是一层不变的，随着时间或空间的推移，会有新的知识加入，如何用新加入的信息更新原有知识而保证知识库的一致性是知识更新需要解决的问题。此外，知识更新也需要满足一些基本条件，在这些基本条件中，Katsuno和Mendelzon提出的$(U1)$-$(U8)$较为常用；

- 提取本体的概要：当一个本体工程师想要快速了解并测试一个本体的内容时，能事先快速地提取出该本体的概要是非常有用的。当一个本体含有很多无关信息时，这使得事半功倍;

- 知识归并：知识库通常来自于多个信息源，这些分布的信息大多会存在冲突（不一致），因而不能简单地将他们放在一起。归并考虑当这些信息矛盾时如何将他们整合[66-70]，基于遗忘的归并尽可能少地遗忘原子以保持知识的一致性[71]。

- 逻辑独立性：生活中许多场景都需要我们判断哪些信息是无关的（或相关的），即信息的独立性[72-75]，如：知识归并中能知道哪些信息是无关的，那么知识归并将变得容易一些。在智能推理中，智能体也需要判断哪些东西（如：命题或文字）是无关的。除了无关性这一名字，还有其他名字揭示了这一本质[?]，如：独立性（independence）、非冗余的（irredundancy）、非影响的（influenceability）、分离的（separability）以及交互性（interactivity）。基于遗忘的独立性表明被遗忘的原子（文字、公式）与得到的结果无关，更一般地，当遗忘的结果与原公式等价时，该公式与被遗忘的原子（文字、公式）无关[71]。

1.2.3 SNC和WSC

正如上文所说，WSC（SNC）对于软件工程中系统的形式化验证有着及其重要的作用。一般说来，最强必要条件（SNC）是最一般的推论（the most general consequence），即：命题成立时能推出的最强的后件（the strongest post-condition, SP），SNC能够蕴涵所有的必要条件；最弱充分条件（WSC）是最特殊的诱因（the most specific abduction），即：使得命题成立的最弱的前提条件（the weakest precondition，WP），WSC能被所有的充分条件蕴涵。

特别地，给定一个程序（program）S和某一状态（state）的规范（specification）Q，则S关于Q的WSC是一个能够描述S初始状态的规范，其中S满足以下两个条件：

（i）S必须终止，

（ii）S执行完成后必须到达能满足Q的状态。

Dijkstra提出了四条规则来计算这样的规范，程序语言里的四种语句（即：赋值语句（assignment statement）、顺序语句（sequence statement）、条件语句（conditional statement）和循环语句（loop statement））分别对应了这四种规则[76]。此外，这两个概念还可用于系统精化[24]、模型检测中的负例产生[23]、汇编语言程序的推理[21]和制定验证条件[22]。在上述计算WSC的方法中，有一个必须满足的要求是“S必须终止”，这就是上文中说到的当系统为反应式系统这类不终止系统时现有方法不能计算WSC的主要原因。

在知识表示与推理中，SNC和WSC为因果理论中后继状态公理的计算提供了一种方法[63]，且SNC和WSC都可以用遗忘来计算[77-78]。之后，SNC和WSC被扩展到FOL下，且用SOQE实现了SNC和WSC的计算[78]。

这里对SNC和WSC的发展和应用只做了简单的概述，在背景知识部分再对其在命题逻辑和模态逻辑这两种情形进行详细介绍（包括其定义和算法）。

1.3　研究目标及主要结果

相关研究工作表明现存方法不能解决反应式系统下的WSC（SNC）的求解。然而，WSC是一种进行系统修改的重要知识，寻求一种有效的求解方法有利于系统正确性的确保。在知识表示与推理中，一种称为遗忘的技术可以用于计算给定理论（公式）的WSC（SNC）。但是，就如上面所述时序逻辑下的遗忘理论尚处于不成熟阶段，没有一个统一的理论框架。此外，如何用遗忘来计算给定系统模型和性质的WSC（SNC）也是一个重要问题。

基于此，本文从遗忘理论的角度出发，研究反应式系统下*SNC*和*WSC*的计算方法，从而为计算不终止类系统下定义在某个符号集上的*SNC*和*WSC*提供了新的方法，架起形式化验证与*KRR*之间的桥梁。为了实现这一目标，本文**主要研究内容及结果**如下：

(1) *CTL*和μ-演算的遗忘理论

本文研究了CTL中遗忘理论的方法和性质，特别是其遗忘结果的存在性、复杂性等，为探索用遗忘理论计算SNC和WSC奠定论基础。具体说来，遗忘理论具有削弱（Weakening，(W)）、正维持（Positive Persistence，(PP)）、负维持（Negative Persistence，(NP)）、无关性（Irrelevance，(IR)）等基本准则[44]。本文探索CTL和μ-演算的遗忘理论的以上准则，并探讨其与存在性之间的关系。此外，本文研究了CTL和μ-演算中计算遗忘结果的方法，探讨了CTL和μ-演算与遗忘相关问题的复杂性结果，为研究计算SNC和WSC的性质、算法以及基本准则等作好铺垫。具体说来，有以下两点：

- **CTL的遗忘理论：** CTL不具有Crig插值（Crig interpolation）性质，因而不具有均匀插值（uniform interpolation）性质[79]，即：存在一个两个公式φ和ψ，若$\varphi \models \psi$，不存在由φ和ψ的公共元素构成的公式θ使得$\varphi \models \theta$且$\theta \models \psi$。在这种情况下，本文除了探讨了上述CTL中遗忘的性质，还研究了CTL子类的遗忘，特别是能保证其遗忘结果仍然是CTL可表达的子类。在这些子类中，一个特殊的子类为约束CTL（bunded CTL）：每个公式的模型是有限个，且每一个模型都能用一个CTL公式表达。因此，其遗忘理论是封闭的。

- **μ-演算的遗忘理论：** 与CTL不同，μ-演算虽然表达能力比CTL强，其可满足性问题也比CTL的复杂，但是μ-演算具有均匀插值性质[80]。这意味着对于在任意

的μ-演算句子中“遗忘”掉任意的原子命题的集合得到的结果仍然是μ-演算公式。本文给出了μ-演算下遗忘理论的主要框架：包括上述遗忘理论的性质和计算遗忘的方法。特别地，证明了μ-演算下的遗忘与均匀插值是一对对偶概念，即本文中的遗忘具有的性质均匀插值也具有，这为研究均匀插值提供了另一种途径。此外，本文还证明了当μ-公式为析取μ-公式时，计算遗忘理论可以在多项式时间内完成，这为μ-演算下遗忘的计算提供了一种有效的方法。

(2) 计算*CTL*遗忘的算法

基于上述研究结果，设计并用Prolog实现了一个计算CTL遗忘结果的原型系统，并从实验角度研究其计算代价以启发快速的计算遗忘结果的算法。具体说来，给出了一种基于归结的计算CTL遗忘的算法，并使用Prolog实现了该算法。此外，对于约束CTL的情形，本文也提出了一种基于模型的算法。

(3) 遗忘理论在反应式系统的形式化验证和知识更新中的应用

反应式系统用Kripke结构表示，Kripke结构是公式的模型。因而，基于上述研究，给出了使用遗忘计算给定有限系统模型（Kripke结构）在给定条件下SNC和WSC的方法。如上所述，一个有限的系统模型能够被一个CTL公式描述，遗忘可以看作以公式和原子命题为运算对象的函数，因而可以使用遗忘计算其SNC和WSC。此外，知识更新是一种使用新发现的性质更新已有理论的技术，本文探讨了如何使用遗忘更新CTL和μ-演算表达的知识。表明了使用遗忘定义的知识更新满足现有的知识更新的八条准则。

针对上述几个内容，解决了以下3个**关键问题**：

(1) *CTL*的遗忘什么情形下存在？如何计算遗忘？

CTL是一种分支时序逻辑，已有文献表明CTL不具有均匀插值性质。与此同时，CTL还引入了时态算子（temporal operator）。在此情形下，研究CTL的遗忘就不能像已有的经典命题逻辑和模态逻辑S5那样，因为遗忘与均匀插值是一对对偶概念，即：它们可以互相证明彼此的存在性。为此，本文深度剖析现存的归结规则，提出了一种基于归结的计算遗忘的方法。该方法表明，当所有在转换为CTL标准形式过程中引入的新的原子命题都被“消除”掉时，使用这一方法得到的结果即为遗忘的结果，即：这一方法是可靠的。尽管CTL下的遗忘不是封闭的，但是本文给出：当被归结的原子命题只同时出现在同一模态词下的命题公式里时，遗忘总是存在的。

此外，针对约束CTL的遗忘，证明了其遗忘结果可以由有限个模型的特征公式（一种CTL公式）的析取来表示，所以这种情形下的遗忘是封闭（存在）的。

(2) 遗忘理论与反应式系统的*SNC*和*WSC*的关系

在经典命题逻辑和一阶逻辑中，遗忘理论与SNC和WSC的关系分别已经被Lin

和Doherty等人分别提出[77-78]。特别地，经典命题逻辑中的SNC和WSC被用于规划问题中的后继状态公理的计算。这里所说的SNC和WSC都是在给定的命题公式或一阶公式下的。本文给出，当给定一个有限反应式系统（Kripke structure）时，将该系统表示为其特征公式时就可使用上述所说的CTL和μ-演算的遗忘理论计算SNC和WSC。

(3) *CTL和μ-演算的遗忘在推理问题上的复杂性*

计算复杂性理论致力于将可计算问题根据它们本身的复杂性分类。研究表明，在经典命题逻辑中：CNF（Conjunctive normal form）公式的遗忘的推理问题最难是Π_2^p-完全的，DNF（Disjunctive normal form）公式的遗忘的推理问题是co-NP-完全的；在命题模态逻辑S5中，遗忘的模型检测问题是NP-完全的，对应的推理问题是Π_2^p-完全的。基于此，本文从现有复杂性结果和自动机理论研究CTL和μ-演算的遗忘在模型检测和推理问题上的复杂性。研究表明μ-演算下关于遗忘的模型检测是EXPTIME的，推理问题的复杂性都是EXPTIME-完全的。而CTL下关于遗忘的模型检测：当只考虑特殊段CTL_{AF}（公式中只包括时态算子AF）时，其模型检测的复杂性为NP-完全的，而有的推理问题的复杂性在co-NP-完全的和Π_2^p-完全的之间。

1.4　论文组织结构

本文研究了计算树逻辑和μ-演算下的遗忘理论，并探讨如何使用遗忘技术来计算SNC（WSC）和知识更新。全文共分为九章，组织结构如图1.2所示，各章节内容的具体安排如下：

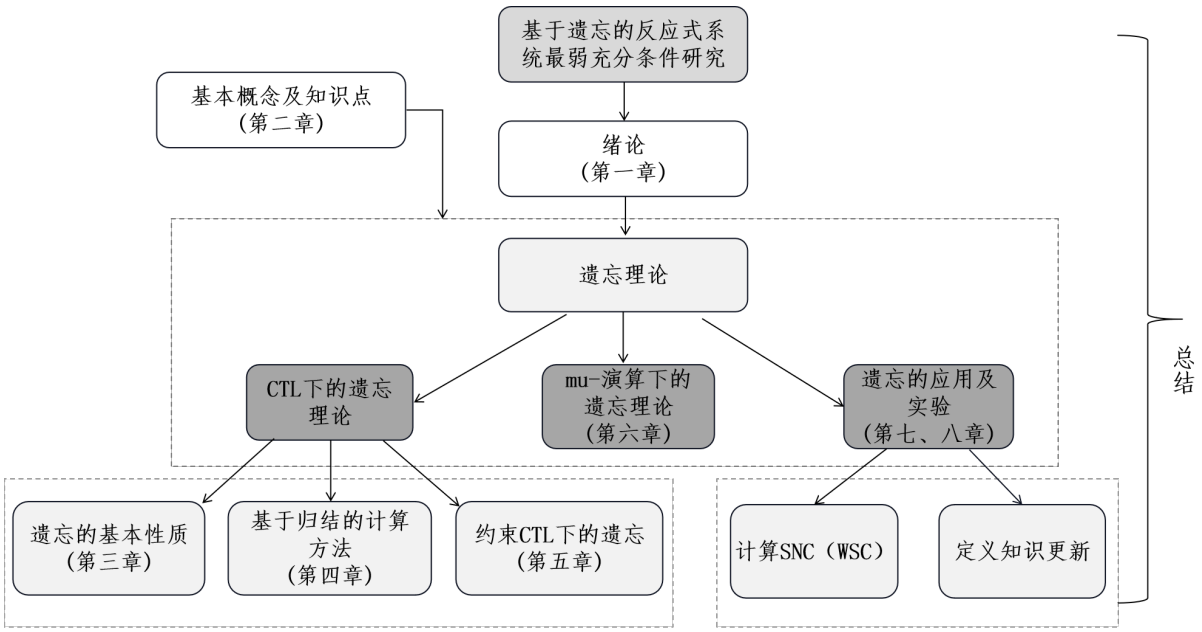


图 1.2: 本文的章节内容组织结构图

第一章为绪论，首先阐述了本文的研究背景及意义，并分析给出了存在的问题，

凝练出本文研究需要解决的关键问题。基于上述分析，阐述了本文的研究内容和研究取得的主要成果。最后给出了本文的章节组织结构安排。

第二章为背景知识，介绍了本文研究所涉及逻辑语言的语法语义及相关技术。首先，给出了经典命题逻辑下解释（赋值）的定义，并基于此描述了解释本文研究的语言的模型结构——Kripke结构。在解释清楚语言所依赖的模型结构之后，本章给出CTL和μ-演算的语法和语义。最后，除了最强必要条件和最弱充分条件在经典逻辑和模态逻辑S5下的研究，本章还介绍了与上述逻辑语言密切相关的两个技术：遗忘和CTL下的归结。

第三章给出*CTL*下遗忘的定义及其基本性质。首先，给出原子命题集合上模型间互模拟的定义，并介绍这一定义的基本属性；其次，根据互模拟来定义CTL下的遗忘，并研究遗忘的属性，包括表达性属性（representation theory）和复杂性结果等。此外，本章还指出CTL下的遗忘是不封闭的，即存在有的公式的遗忘的结果不能用CTL来表示。

第四章为基于归结的*CTL*遗忘计算。基于第二章的归结规则，本章探讨如何使用该归结系统计算CTL下的遗忘。首先，将CTL公式转换为归结规则需要的子句形式——$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句（后文详细介绍）；其次，使用归结规则计算所有可能的需要遗忘的原子命题上的归结结果；随后，移除那些包含要遗忘的原子命题的子句，并给出一种一般化的Ackermann引理消除一些新引入的原子命题；最后，将得到的结果转换成CTL公式。此外，基于上述过程提出计算CTL下的遗忘的算法，并分析该算法的时间和空间复杂性。

第五章为约束*CTL*下的遗忘。在第三章所说，CTL下的遗忘不是封闭的，因此在本章探讨了约束CTL遗忘是封闭的情形。为此，本章提出了一种约束的互模拟，并给出给定深度的计算树在给定原子命题集合下的特征公式，继而给出有限情况下Kripke结构在给定原子命题集合下的特征公式。最后说明约束情形下CTL公式的遗忘结果总是CTL公式可表达的。

第六章为μ-演算下的遗忘。μ-演算是一种表达能力比CTL强的时序逻辑，其具有均匀插值性质。本章给出μ-演算下遗忘的定义和基本属性，包括模块性、交换性及同质性。此外，探讨了μ-演算下的遗忘是封闭的，且对任意析取μ-公式的遗忘可以在多项式时间内计算出来。最后，给出μ-演算下关于遗忘算子的模型检测和推理问题的复杂性结果。

第七章为遗忘理论的应用。讲述如何将遗忘应用于计算SNC（WSC）和定义知识更新。此时，有限状态的反应式系统模型的WSC（SNC）可以通过遗忘来计算，且通过遗忘定义的知识更新满足Katsuno等人提出的知识更新应该满足的基本准则。

第八章为实验结果。给出基于归结的算法实现模型的实验结果及分析。

第九章为总结与展望。首先总结了本文的研究工作，进一步，展望了未来研究工

作的方向和重点。

第二章　基础知识

本章主要介绍本文用到的符号、术语以及逻辑理论基础，包括：反应式系统、*Kripke*结构、时序逻辑（尤其是计算树逻辑（*CTL*）和μ-演算）和遗忘理论等。首先，介绍解释时序逻辑语言所需的模型结构，即*Kripke*结构。其次，主要介绍时序逻辑中本文探讨的计算树逻辑和μ-演算。为了后文使用基于归结的方法计算*CTL*下的遗忘，本章详细介绍*CTL*下的归结系统及探索这些规则潜在的基本性质。此外，遗忘理论是本文的研究重点，其概念、性质及在各个研究领域的研究情况将会被当作本章的重点详细介绍。

为了方便，本文将命题变量（也叫原子命题）的集合记作$\mathscr{A}$，$V \subseteq \mathscr{A}$ 是$\mathscr{A}$的子集。此外，规定$\overline{V}$是V在$\mathscr{A}$上的补，也即是$\overline{V} = \mathscr{A} - V$。

2.1　反应式系统

反应式系统是一种不终止的、与环境有着持续不断交互的系统。其种类繁多，如微处理器（microprocessor）、计算机操作系统（computer operating systems），航空交通管制系统（air traffic control systems）、车载电子设备（on-board avionics）以及其他嵌入式系统。

保证这些系统的正确性至关重要，而系统设计正变成验证系统的同义词。

"*The design of modern information processing systems like digital circuits or protocols is becoming more and more difficult. A large part of the design more and more a synonym for verifying systems.*"

–Klaus Schneider

模型检测是理论计算机科学中最为成功的技术之一，也是反应式系统验证的一种重要手段，而时序逻辑是其规范描述语言[27]。在模型检查中，反应式系统通常用Kripe结构表示[19,81]。本文主要探讨其规范描述语言和Kripke结构，因而如何将反应式系统转换为Kripke结构的方法可以参见文章[82-84].

下文将详细描述用于描述反应式系统的Kripke结构和描述反应式系统规范的时序逻辑语言，本文主要指CTL和μ-演算。

2.2　Kripke结构

Kripke结构作为一种表示转换系统（transition system）的数学模型，在理论计算机科学领域有着广泛的应用，尤其是作为解释时序逻辑公式的模型结构，本节将

对Kripke结构做详细描述。

2.2.1　真假赋值和K-解释

经典命题语言$\mathscr{L}^p$由以下三类符号构成：

- 命题符号：一般用小写拉丁字母p，q，r，...来表示，且这些命题符号来源于$\mathscr{A}$；
- 联结符号：$\neg$（否定），$\wedge$（合取），$\vee$（析取），$\rightarrow$（蕴涵），$\leftrightarrow$（等值于）；
- 标点符号：(（左括号），)（右括号）。

$\mathscr{L}^p$的原子公式集合为命题符号集合$Atom(\mathscr{L}^p)$，公式的集合为$\mathscr{F}(\mathscr{L}^p)$。$\varphi \in \mathscr{F}(\mathscr{L}^p)$当且仅当它能由（有限次使用）以下的三条规则生成[85]：

- 如果$\varphi \in Atom(\mathscr{L}^p)$，则$\varphi \in \mathscr{F}(\mathscr{L}^p)$。
- 如果$\varphi \in \mathscr{F}(\mathscr{L}^p)$，则$(\neg\varphi) \in \mathscr{F}(\mathscr{L}^p)$。
- 如果φ、$\varphi' \in \mathscr{F}(\mathscr{L}^p)$，则$(\varphi * \varphi') \in \mathscr{F}(\mathscr{L}^p)$。其中，$* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$。

此外，也称“ture”和“false”为原子公式，分别记为“$\top$”和“$\bot$”。原子命题或其否定称为**文字**，有限个文字的析取称为**子句**，有限个文字的合取称为**项**。

例 2.1. 下面几个字符串为$\mathscr{L}^p$的公式：

- $(q \vee p)$；
- $(((\neg p) \leftrightarrow (q \vee r)) \rightarrow (r \wedge p))$。

而字符串$p \wedge \vee q$不属于集合$\mathscr{F}(\mathscr{L}^p)$。

为了方便，称$\mathscr{L}^p$的公式为**命题公式**（在不引起歧义的情况下也称之为公式）。此外，规定联结符号的优先级有助于简化公式（省略掉冗余的标点符号）。为此，规定在下面的序列中，每个左边的联结符号优先于右边的联结符号。

$$\neg \qquad \wedge \qquad \vee \qquad \rightarrow \qquad \leftrightarrow$$

此时，例 2.1中的公式$(((\neg p) \leftrightarrow (q \vee r)) \rightarrow (r \wedge p))$可写为$(\neg p \leftrightarrow q \vee r) \rightarrow r \wedge p$。

在讨论了命题公式的语法结构之后，接下来将讨论其语义解释。

定义 2.1 (真假赋值). 真假赋值是以所有命题符号的集为定义域，以真假值的集$\{0,1\}$为值域的函数$v:\mathscr{A}\to\{0,1\}$。

为了方便，后文中也用$\top$代表1，$\bot$代表0（此时真假赋值为$v:\mathscr{A}\to\{\bot,\top\}$），且满足对任意的真假赋值$v$都有$\top^v=\top$和$\bot^v=\bot$。由该定义可知，一个真假赋值要同时给$\mathscr{A}$中的所有命题符号指派一个真假值，所以真假赋值的个数为$2^{|\mathscr{A}|}$。真假赋值v给公式φ指派的值记作φ^v，定义如下：

定义 2.2 (公式的真假值). 真假赋值v给公式指派的真假值递归定义如下：

- $p^v\in\{\bot,\top\}$，其中$p\in\mathscr{A}$。
- $(\neg\varphi)^v=\begin{cases}\top, & \text{如果}\varphi^v=\top;\\ \bot, & \text{否则。}\end{cases}$
- $(\varphi\wedge\psi)^v=\begin{cases}\top, & \text{如果}\varphi^v=\psi^v=\top;\\ \bot, & \text{否则。}\end{cases}$
- $(\varphi\vee\psi)^v=\begin{cases}\top, & \text{如果}\varphi^v=\top\text{或}\psi^v=\top;\\ \bot, & \text{否则。}\end{cases}$
- $(\varphi\to\psi)^v=\begin{cases}\top, & \text{如果}\varphi^v=\bot\text{或}\psi^v=\top;\\ \bot, & \text{否则。}\end{cases}$
- $(\varphi\leftrightarrow\psi)^v=\begin{cases}\top, & \text{如果}\varphi^v=\psi^v;\\ \bot, & \text{否则。}\end{cases}$

对于任意的命题公式φ和真假赋值v，当$\varphi^v=\top$时，称v是公式φ的一个模型，也可以记为$v\models\varphi$，读作v满足φ。一般地，当存在一个真假赋值v使得$v\models\varphi$，则称公式φ是可满足的。如果φ是可满足的，且$\neg\varphi$是不可满足的，则称φ是有效的。可满足问题（SAT）是一个NP-完全问题，当公式可满足时关注如何验证和求解极小模型[86-88]，而当公式不可满足时，极大可满足和极小不可满足就成为研究的重点[89-90]。

值得注意的是，一个真假赋值也可以看作是原子命题的集合I，其表示出现在I中的原子命题为真，反之为假。

模态逻辑是经典逻辑的扩充，它是经典逻辑中引进“必然”和“可能”这两种模态词得到的。如上所述，命题的真假值只有两种，命题是真的（1）或是假的（0）。而在模态逻辑中，把命题区分为必然真的命题和并非必然真的命题，把假命题区分为必然假的和并非必然假的命题。对于任何命题φ，可以有两种模态命题：“φ是必然的”和“φ是可能的”。值得注意的是，时序逻辑也是模态逻辑的一种[43]。尽管如此，本文

在说模态逻辑的时候通常指不带有时序操作符的情况，说时序逻辑时指带有时序操作符的情况。

本文所说的模态逻辑为命题单模态逻辑（propositional mono-modal logic）。模态公式的集合$\mathscr{F}^{\mathscr{M}}$是包含“$\top$”和“$\bot$”的满足如下条件的最小集：

- $\mathscr{A} \subseteq \mathscr{F}^{\mathscr{M}}$；
- 如果$\varphi \in \mathscr{F}^{\mathscr{M}}$，则$(\neg\varphi)$，$(\mathrm{K}\varphi) \in \mathscr{F}^{\mathscr{M}}$；
- 如果φ，$\psi \in \mathscr{F}^{\mathscr{M}}$，则$(\varphi * \psi) \in \mathscr{F}^{\mathscr{M}}$，其中$* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$。

令$\mathrm{B} = \neg\mathrm{K}\neg$，则$\mathrm{B}\varphi \in \mathscr{F}^{\mathscr{M}}$。其中，K和B叫做模态符号，分别表示“必然”和“可能”。

可能世界语义（或*Kripke*语义）是标准的命题模态逻辑语义[91]。Kripke语义是定义在Kripke结构上的，一个Kripke结构是一个三元组(S, R, L)（下一节中将详细介绍）。其中，S是状态的非空集合，$R \subseteq S \times S$是可达性关系。特别地，当R是一个等价关系的时候（模态逻辑S5中），一个Kripke结构可以写成一个二元组$\langle W, w\rangle$，其中W是状态的非空集合，w是W中的元素，每个状态是原子命题的集合。此时，称$\mathscr{M} = \langle W, w\rangle$为一个K-解释（K-interpretation）[44]。

定义 2.3. 给定一个K-解释$\mathscr{M} = \langle W, w\rangle$，其与*S5*中的公式的可满足关系被归纳地定义为：

- $\mathscr{M} \not\models \bot$，$\mathscr{M} \models \top$；
- $\mathscr{M} \models p$当且仅当$p \in w$，其中$p \in \mathscr{A}$；
- $\mathscr{M} \models \neg\varphi$当且仅当$\mathscr{M} \not\models \varphi$；
- $\mathscr{M} \models \varphi \supset \psi$当且仅当$\mathscr{M} \not\models \varphi$或$\mathscr{M} \models \psi$；
- $\mathscr{M} \models \mathrm{K}\varphi$当且仅当$\forall w' \in W$有$\langle W, w'\rangle \models \varphi$。

$\mathscr{M} = \langle W, w\rangle$称为公式$\varphi$的K-模型（K-model），当且仅当$\mathscr{M} \models \varphi$。此外，如果存在一个$\mathscr{M} = \langle W, w\rangle$使得公式$\mathscr{M} \models \varphi$，则称公式$\varphi$是可满足的。如果$\mathscr{M} \models \varphi$对于所有的$\mathscr{M} = \langle W, w\rangle$都成立，则称$\varphi$是有效的。

2.2.2　Kripke结构

给定一个可数无限索引的集合Ind，一个初始Ind-Kripke结构是一个五元组$\mathscr{M} = (S, R, L, [_], s_0)$，其中：

- S是状态的非空集合，s_0是$\mathscr{M}$的初始状态（下面详细介绍）；
- $R \subseteq S \times S$是状态转换函数，且对任意的$s \in S$，存在$s' \in S$使得$(s,s') \in R$；
- $L : S \to 2^{\mathscr{A}}$是一个标签函数；
- $[_] : \mathrm{Ind} \to 2^{S \times S}$是一个将索引集合Ind中的元素$ind$映射为后继函数$[ind]$使得对任意的$s \in S$都存在唯一一个状态$s' \in S$使得$(s,s') \in [ind] \cap R$。

Kripke结构$\mathscr{M} = (S,R,L)$上的*路径*是$\mathscr{M}$上的状态构成的无限序列$\pi = (s_0, s_1, s_2, \ldots)$，其中对任意的$j \geq 0$都有$(s_j, s_{j+1}) \in R$。用$s' \in \pi$表示$s'$是路径$\pi$上的一个状态。特别地，用$\pi_s$表示从$s$开始的$\mathscr{M}$上的一条路径。如果对$\mathscr{M}$中任意的状态$s'$都有一条路径$\pi_s$使得$s' \in \pi_s$，那么称$s$为*初始状态*。Ind-Kripke 结构$\mathscr{M} = (S,R,L,[_])$上的一条*索引路径*$\pi_s^{\langle ind \rangle}$（$ind \in \mathrm{Ind}$）是一条路劲$(s_0(=s), s_1, s_2, \ldots)$且对任意的$j \geq 0$有$(s_j, s_{j+1}) \in [ind]$。

此外，可由初始Ind-Kripke结构得到其他几种特殊的Kripke结构：

- 从初始Ind-Kripke结构$\mathscr{M}$中去掉$[_]$元素得到的结构称为初始Kripke结构$\mathscr{M} = (S,R,L,s_0)$；
- 从初始Ind-Kripke结构$\mathscr{M}$中去掉初始状态s_0得到的结构称为Ind-Kripke结构$\mathscr{M} = (S,R,L,[_])$；
- 从初始Ind-Kripke结构$\mathscr{M}$中同时去掉$[_]$和s_0得到的结构称为Kripke结构$\mathscr{M} = (S,R,L)$。

通常一个转换系统（transition system）（包括反应式系统）能够被抽象为一个Kripke结构[20]。一个*(Ind-)结构*是一个二元组$\mathscr{K} = (\mathscr{M}, s)$，其中$\mathscr{M}$是一个初始(Ind-)Kripke结构，$s$是$\mathscr{M}$中的一个状态。如果$s$为$\mathscr{M}$上的初始状态，则称$\mathscr{K}$为*初始(Ind-)结构*。上面的关于(索引)路径的概念对于这些结构也可相似地定义。

*树*是一种只有一个根节点（没有其他节点指向且可达于其他任意节点的节点）的无环图。给定一个初始Kripke结构$\mathscr{M} = (S,R,L,s_0)$和一个状态$s \in S$，定义在$\mathscr{M}$上以$s$为根节点的深度为为$n$（$n \geq 0$）的*计算树*$\mathrm{Tr}_n^{\mathscr{M}}(s)$递归定义如下[92]：

- $\mathrm{Tr}_0^{\mathscr{M}}(s)$ 是只有一个节点s（其标签为$L(s)$）的树。
- $\mathrm{Tr}_{n+1}^{\mathscr{M}}(s)$是以$s$为根节点（标签为$L(s)$）的树，并且满足若$(s,s') \in R$，则节点$s$有一棵子树$\mathrm{Tr}_n^{\mathscr{M}}(s')$。

2.3　时序逻辑

时序逻辑是一种描述系统规范的形式化语言，它研究状态随时间变化而变化的系统的逻辑特性。由于软件和硬件系统运行的本质是状态变化的过程，所以时序逻辑在程序验证和硬件验证中应用得相当广泛。计算树逻辑（Computation Tree Logic, CTL）是分支时序逻辑的一种，其模型检测是多项式时间可行的。然而，CTL表达系统性质的表达能力不如μ-演算（μ-calculus），如："某给定的系统中存在一条路径使得该路径上的第偶数个状态满足特定的性质"这一规范是不能用除了μ-演算以外的其他时序逻辑表示的[19]。充分考虑这两种逻辑语言自身的特性，本节主要介绍CTL和μ-演算这两种时序逻辑。因此，本文所说的公式指CTL（或μ-演算）公式，即用来描述一个规范（或性质）的公式是CTL（或μ-演算）公式。

2.3.1　计算树逻辑（CTL）

CTL由Clark和Emerson等人于1986年提出[93]。这里给出带索引的CTL公式定义，而CTL公式是该种公式的子类。带索引的CTL的语言$\mathscr{L}$由下面的几类符号构成：

- 原子命题的集合$\mathscr{A}$；
- 可数无限索引集合Ind；
- 命题常量**start**；
- 常量符号：$\top$和$\bot$，分别表示"真"和"假"；
- 联结符号：$\vee$和$\neg$，分别表示"析取"和"否定"；
- 路径量词：A、E和E_{ind}，分别表示"所有"、"存在"和"存在索引为$ind \in \mathrm{Ind}$"的路径；
- 时序操作符：X、F、G、U和W，分别表示"下一个状态"、"将来某一个状态"、"将来所有状态"、"直到"和"除非"；
- 标点符号："("和")"。

带索引的CTL公式（在不引起歧义时也叫做公式）的时序算子与CTL公式的时序算子相同，是路径量词和时序操作符的组合（路径量词在前，时序操作符在后），如：AX、EX、E_{ind}X、AF等。此时，语言$\mathscr{L}$的*存在范式(existential normal form, ENF)*可以用巴科斯范式递归定义如下：

$$\phi ::= \mathbf{start} \mid \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \mathrm{EX}\phi \mid \mathrm{EG}\phi \mid \mathrm{E}(\phi\ \mathrm{U}\ \phi) \mid \mathrm{E}_{\langle ind \rangle}\mathrm{X}\phi \mid \mathrm{E}_{\langle ind \rangle}\mathrm{G}\phi \mid \mathrm{E}_{\langle ind \rangle}(\phi \mathrm{U} \phi)$$

其中，$p \in \mathscr{A}$，$ind \in \mathrm{Ind}$。$\mathscr{L}$中其他形式的公式可以通过下面的定义（使用上述定义中的形式）得到：

$$\varphi \wedge \psi \stackrel{def}{=} \neg(\neg\varphi \vee \neg\psi) \tag{2.1}$$

$$\varphi \rightarrow \psi \stackrel{def}{=} \neg\varphi \vee \psi \tag{2.2}$$

$$\mathrm{A}(\varphi \mathrm{U} \psi) \stackrel{def}{=} \neg\mathrm{E}(\neg\psi \mathrm{U}(\neg\varphi \wedge \neg\psi)) \wedge \neg\mathrm{EG}\neg\psi \tag{2.3}$$

$$\mathrm{A}(\varphi \mathrm{W} \psi) \stackrel{def}{=} \neg\mathrm{E}((\varphi \wedge \neg\psi)\mathrm{U}(\neg\varphi \wedge \neg\psi)) \tag{2.4}$$

$$\mathrm{E}(\varphi \mathrm{W} \psi) \stackrel{def}{=} \neg\mathrm{A}((\varphi \wedge \neg\psi)\mathrm{U}(\neg\varphi \wedge \neg\psi)) \tag{2.5}$$

$$\mathrm{AF}\varphi \stackrel{def}{=} \mathrm{A}(\top \mathrm{U} \psi) \tag{2.6}$$

$$\mathrm{EF}\varphi \stackrel{def}{=} \mathrm{E}(\top \mathrm{U} \psi) \tag{2.7}$$

$$\mathrm{AX}\varphi \stackrel{def}{=} \neg\mathrm{EX}\neg\varphi \tag{2.8}$$

$$\mathrm{AG}\varphi \stackrel{def}{=} \neg\mathrm{EF}\neg\varphi \tag{2.9}$$

没有索引和**start**的公式称为CTL公式。此外，对于给定的公式φ，其否定范式（negation normal form, NNF）是将否定联结词“$\neg$”的出现通过上述定义变化到只出现在原子命题之前的形式。

与经典命题逻辑一样，给联结符号规定优先级，有助于减少过多括号使用带来的符号冗余。带索引的CTL中的联结符号的优先级如下序列所示，左边的联结符号优先于右边的联结符号：

$$\neg, \mathrm{EX}, \mathrm{EF}, \mathrm{EG}, \mathrm{AX}, \mathrm{AF}, \mathrm{AG}, \mathrm{E}_{\langle ind \rangle}\mathrm{X}, \mathrm{E}_{\langle ind \rangle}\mathrm{F}, \mathrm{E}_{\langle ind \rangle}\mathrm{G}, \wedge, \vee, \quad \mathrm{EU}, \mathrm{AU}, \mathrm{EW}, \mathrm{AW}, \mathrm{E}_{\langle ind \rangle}\mathrm{U}, \mathrm{E}_{\langle ind \rangle}\mathrm{W}, \rightarrow .$$

给定一个不包含“$\rightarrow$”的公式φ和原子命题p，若如果p在φ中的出现之前有偶数个否定$\neg$，则称p在φ中的出现为*正出现*，否则为*负出现*。若φ中所有p的出现都为正出现（或负出现），则称φ关于p是正的（或负的）。此外，对于给定的公式集合，如果该集合中的所有公式关于p都是正的（或负的），则说该集合关于p是正的（或负的）。

带索引的CTL的语义定义在Kripke结构上，可以严格地描述如下。

定义 2.4 (带索引的CTL的语义). *给定带索引的CTL公式φ，初始Ind-Kripke结构$\mathscr{M} = (S, R, L, [_], s_0)$和状态$s \in S$。$(\mathscr{M}, s)$与$\varphi$之间的可满足关系$(\mathscr{M}, s) \models \varphi$定义如下：*

- $(\mathscr{M}, s) \models \textbf{\textit{start}}$ *当且仅当* $s = s_0$；
- $(\mathscr{M}, s) \not\models \bot$；
- $(\mathscr{M}, s) \models p$ *当且仅当* $p \in L(s)$；

- $(\mathscr{M},s) \models \varphi_1 \vee \varphi_2$当且仅当$(\mathscr{M},s) \models \varphi_1$或$(\mathscr{M},s) \models \varphi_2$；
- $(\mathscr{M},s) \models \neg\varphi$当且仅当$(\mathscr{M},s) \not\models \varphi$；
- $(\mathscr{M},s) \models \mathrm{EX}\varphi$当且仅当存在$S$中的一个状态$s_1$，使得$(s,s_1) \in R$且$(\mathscr{M},s_1) \models \varphi$；
- $(\mathscr{M},s) \models \mathrm{EG}\varphi$当且仅当存在$\mathscr{M}$上的一条路径$\pi_s = (s_1 = s, s_2, \ldots)$，使得对每一个$i \geq 1$都有$(\mathscr{M},s_i) \models \varphi$；
- $(\mathscr{M},s) \models \mathrm{E}(\varphi \mathrm{U} \psi)$当且仅当存在$\mathscr{M}$上的一条路径$\pi_s = (s_1 = s, s_2, \ldots)$，使得对某一个$i \geq 1$有$(\mathscr{M},s_i) \models \psi$，同时对任意的$1 \leq j < i$有$(\mathscr{M},s_j) \models \varphi$；
- $(\mathscr{M},s) \models \mathrm{E}_{\langle ind \rangle}\mathrm{X}\psi$ 当且仅当对索引路劲$\pi_s^{\langle ind \rangle}$，$(\mathscr{M},s') \models \psi$ 且$(s,s') \in [ind]$;
- $(\mathscr{M},s) \models \mathrm{E}_{\langle ind \rangle}\mathrm{G}\psi$ 当且仅当对任意的$s' \in \pi_s^{\langle ind \rangle}$, $(\mathscr{M},s') \models \psi$;
- $(\mathscr{M},s) \models \mathrm{E}_{\langle ind \rangle}(\psi_1 \mathrm{U} \psi_2)$ 当且仅当存在$\pi_s^{\langle ind \rangle} = (s = s_1, s_2, \ldots)$中的$s_j\,(1 \leq j)$使得$(\mathscr{M},s_j) \models \psi_2$且对任意的$s_k \in \pi_s^{\langle ind \rangle}$，若$1 \leq k < j$，则$(\mathscr{M},s_k) \models \psi_1$。

与Browne和Bolotov等人的工作类似，本文只将初始Ind-结构作为模型的候选项[92,94]。换句话说，对于给定的Ind-结构$(\mathscr{M},s)$和带索引的CTL公式φ，如果$(\mathscr{M},s) \models \varphi$且$s = s_0$，则称$(\mathscr{M},s)$为公式$\varphi$的一个模型。更清楚地说，对于给定的初始Ind-结构$\mathscr{K} = (\mathscr{M},s_0)$，如果$\mathscr{K} \models \varphi$，则称$\mathscr{K}$是$\varphi$的一个模型。

令φ、φ_1和φ_2为公式，为了符号的统一，这里列出文中出现的一些记号的含义。

- $Mod(\varphi)$：公式φ的所有模型构成的集合；
- 可满足：如果$Mod(\varphi) \neq \emptyset$，则称$\varphi$是可满足的；
- 逻辑蕴涵：若$Mod(\varphi_1) \subseteq Mod(\varphi_2)$，则称$\varphi_1$逻辑地蕴涵$\varphi_2$，记为$\varphi_1 \models \varphi_2$；
- 逻辑等值：当$\varphi_1 \models \varphi_2$且$\varphi_2 \models \varphi_1$时，即$Mod(\varphi_1) = Mod(\varphi_2)$，则称$\varphi_1$和$\varphi_2$为逻辑等值公式（简称为等值公式），记作$\varphi_1 \equiv \varphi_2$。

值得注意的是，上述的记号也适用于讨论的对象为公式的集合的情形。此外，给定一个公式的集合Π和一个初始Ind-结构$\mathscr{K}$，若对于Π中的任意一个公式φ都有$\mathscr{K} \models \varphi$，则$\mathscr{K} \models \Pi$。若$\Pi$为公式的有限集合，则用$\bigvee\Pi$（$\bigwedge\Pi$）表示$\Pi$中公式的析取（合取），本文也用$\Pi$表示$\bigwedge\Pi$。

对于给定的公式φ，将出现在φ中的原子命题的集合记为$Var(\varphi)$。此外，给定公式φ和原子命题的集合V，如果存在一个公式ψ使得$Var(\psi) \cap V = \emptyset$且$\varphi \equiv \psi$，那么说$\varphi$与$V$中的原子命题无关，简称为$V$-无关（$V$-*irrelevant*），写作$\mathrm{IR}(\varphi,V)$。一种特殊的

形式是$Var(\varphi) \subseteq V$，此时称φ为集合V上的公式。可以类似定义公式的集合与原子命题集合的无关性，也即是：如果对于公式的集合Π中的任意一个公式φ，IR(φ, V)都成立，则Π与V中的原子命题无关，记为IR(Π, V)。

CTL公式与结构之间的可满足性与定义 2.4类似，只是将Ind-结构替换为结构。除此之外，相应的符号（记号）也与上述情况类似。

根据上面的定义，以下结论是显然的。

引理 2.1. 给定两个公式φ 和ψ，则下列结论成立：

(i) $\mathrm{AG}(\varphi \wedge \psi) \equiv (\mathrm{AG}\varphi) \wedge (\mathrm{AG}\psi)$；

(ii) $\mathrm{AGAG}(\varphi) \equiv \mathrm{AG}(\varphi)$；

(iii) $\mathrm{AG}(\boldsymbol{start} \to \varphi) \equiv \varphi$。

证明. (i) ($\Rightarrow$) 对$\mathrm{AG}(\varphi \wedge \psi)$的任意模型$(\mathscr{M}, s)$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \varphi \wedge \psi$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \varphi$ 和$(\mathscr{M}, s_i) \models \psi$
$\Rightarrow (\mathscr{M}, s) \models (\mathrm{AG}\varphi) \wedge (\mathrm{AG}\psi)$.

($\Leftarrow$) 对$(\mathrm{AG}\varphi) \wedge (\mathrm{AG}\psi)$的任意模型$(\mathscr{M}, s)$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \varphi$ 和$(\mathscr{M}, s_i) \models \psi$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \varphi \wedge \psi$
$\Rightarrow (\mathscr{M}, s) \models \mathrm{AG}(\varphi \wedge \psi)$。

(ii) ($\Rightarrow$) 对$\mathrm{AGAG}(\varphi)$的任意模型$(\mathscr{M}, s)$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \mathrm{AG}(\varphi)$
$\Rightarrow (\mathscr{M}, s) \models \mathrm{AG}(\varphi)$。

($\Leftarrow$) 对$\mathrm{AG}(\varphi)$的任意模型$(\mathscr{M}, s)$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \varphi$
$\Rightarrow \forall \pi' = (s_i, s_{i+1}, \ldots)$，对任意的$i \geq 0$ 和$j \geq i$有$(\mathscr{M}, s_j) \models \varphi$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \mathrm{AG}(\varphi)$
$\Rightarrow (\mathscr{M}, s) \models \mathrm{AGAG}(\varphi)$。

(iii) ($\Rightarrow$) 对$\mathrm{AG}(\mathbf{start} \to \varphi)$的任意模型$(\mathscr{M}, s)$
$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，有对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \mathbf{start} \to \varphi$
$\Rightarrow (\mathscr{M}, s) \models \varphi$ （因为$(\mathscr{M}, s) \models \mathbf{start}$）

($\Leftarrow$) 对φ的任意模型$(\mathscr{M}, s)$
$\Rightarrow (\mathscr{M}, s) \models \mathbf{start} \to \varphi$ （因为$(\mathscr{M}, s) \models \mathbf{start}$）
$\Rightarrow (\mathscr{M}, s') \models \mathbf{start} \to \varphi$ （因为对任意的$s' \neq s$，$(\mathscr{M}, s') \not\models \mathbf{start}$）

$\Rightarrow \forall \pi = (s = s_0, s_1, \ldots)$，对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \mathbf{start} \rightarrow \varphi$
$\Rightarrow (\mathscr{M}, s') \models \mathrm{AG}(\mathbf{start} \rightarrow \varphi)$。 □

给定两个带索引的CTL公式φ 和ψ，若φ是可满足的当且仅当ψ是可满足的，则称φ 和ψ 是等价可满足的（*equi-satisfiable*）。可以看出，相互等价（等值）的公式是等价可满足的，但是反过来却不成立。例：公式$\mathrm{E}_{\langle 1\rangle}\mathrm{X}p$和$\mathrm{E}_{\langle 2\rangle}\mathrm{X}p$是等价可满足的，但是他们不是等价的。基于此，下面的引理是显然的。

引理 2.2. *令φ为带索引的CTL公式*

(i) *φ 和φ' 是等价可满足的，其中φ' 是由φ通过用新的索引命名现有的一些索引得到的公式，即φ是将出现在其中的一些索引用不出现在其中的索引全部替换得到，且不同的索引对应不同新索引。*

(ii) *$\varphi \models \varphi''$，其中$\varphi''$ 是移除掉φ中的索引得到的公式。*

证明. (i) 不失一般性地，令$\varphi' = \varphi[i/j]$ 为用新的索引j 替换φ中的索引得到的公式。

($\Rightarrow$) 对φ的任意模型$(\mathscr{M}, s)$，其中$\mathscr{M} = (S, R, L, [_], s)$
$\Rightarrow (\mathscr{M}', s) \leftrightarrow_{\emptyset} (\mathscr{M}, s)$ 和$(\mathscr{M}', s) \models \varphi'$，其中$\mathscr{M}' = (S, R, L, [_]', s)$ 且对任意的$x \in \mathrm{Ind}$

$$[x]' = \begin{cases} [i], & \text{若} x = j; \\ [x], & \text{否则。} \end{cases}$$

可以类似地证明对φ'的任意模型$(\mathscr{M}, s)$，存在$(\mathscr{M}', s)$ 使得$(\mathscr{M}', s) \leftrightarrow_{\emptyset} (\mathscr{M}, s)$ 和$(\mathscr{M}', s) \models \varphi$。

(ii) 这可从带索引的CTL的语义得出。 □

值得注意的是，引理中的(ii)的逆命题不成立。例如：令$\varphi = \mathrm{E}_{\langle 1\rangle}\mathrm{X}p \wedge \mathrm{E}_{\langle 1\rangle}\mathrm{X}\neg p$，显然$\varphi'' = \mathrm{EX}p \wedge \mathrm{EX}\neg p$是可满足的，但是$\varphi$不可满足。因此，$\varphi'' \not\models \varphi$。

2.3.2 CTL的标准形式

已有结果表明，任意的CTL公式能够在多项式时间内被转换为CTL的全局子句分离的范式（separated normal form with global clauses for CTL，$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句）[95-96]。

$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句是具有下面几种形式的公式：

$$
\begin{array}{ll}
\mathrm{A}\mathrm{G}(\mathbf{start} \to \bigvee_{j=1}^{k} m_j) & (\text{初始句，initial clause}) \\
\mathrm{A}\mathrm{G}(\top \to \bigvee_{j=1}^{k} m_j) & (\text{全局子句，global clause}) \\
\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n} l_i \to \mathrm{A}\mathrm{X}\bigvee_{j=1}^{k} m_j) & (\text{A-步子句，A-step clause}) \\
\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n} l_i \to \mathrm{E}_{\langle ind\rangle}\mathrm{X}\bigvee_{j=1}^{k} m_j) & (\text{E-步子句，E-step clause}) \\
\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n} l_i \to \mathrm{A}\mathrm{F} l) & (\text{A-某时子句，A-sometime clause}) \\
\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n} l_i \to \mathrm{E}_{\langle ind\rangle}\mathrm{F} l) & (\text{E-某时子句，E-sometime clause})
\end{array}
$$

其中k和n都是大于0的常量，**start**是命题常量符号，l_i（$1 \le i \le n$）、m_j（$1 \le j \le k$）和l都是文字，且$ind \in \mathrm{Ind}$。从上述标准形式中，可以看到每个$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句都是$\mathrm{AG}(P \to G)$形式。因此在没有歧义的情况下，下文中将使用$P \to G$指代这些子句。此外，除了额外说明，本文通常将$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句和子句统称为子句。

一个CTL公式φ可以通过表 2.1中的规则将其转换为一个$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合，记为$T_{\varphi}$。

表 2.1: 转换规则

Trans(1) $\dfrac{q \to \mathrm{E}T\varphi}{q \to \mathrm{E}_{\langle ind\rangle}T\varphi}$;	**Trans(2)** $\dfrac{q \to \mathrm{E}(\varphi_1 \mathrm{U} \varphi_2)}{q \to \mathrm{E}_{\langle ind\rangle}(\varphi_1 \mathrm{U} \varphi_2)}$;	**Trans(3)** $\dfrac{q \to \varphi_1 \wedge \varphi_2}{q \to \varphi_1, q \to \varphi_2}$;
Trans(4) $\dfrac{q \to \varphi_1 \vee \varphi_2 \text{ (如果}\varphi_2\text{不是子句)}}{q \to \varphi_1 \vee p, p \to \varphi_2}$;		**Trans(5)** $\dfrac{q \to D}{\top \to \neg q \vee D}$; $\dfrac{q \to \bot}{\top \to \neg q}$; $\dfrac{q \to \top}{\{\}}$
Trans(6) $\dfrac{q \to Q\mathrm{X}\varphi \text{ (如果}\varphi\text{不是子句)}}{q \to Q\mathrm{X}p, p \to \varphi}$;		**Trans(7)** $\dfrac{q \to Q\mathrm{F}\varphi \text{ (如果}\varphi\text{不是文字)}}{q \to Q\mathrm{F}p, p \to \varphi}$;
Trans(8) $\dfrac{q \to Q(\varphi_1 \mathrm{U} \varphi_2) \text{ (如果}\varphi_2\text{不是文字)}}{q \to Q(\varphi_1 \mathrm{U} p), p \to \varphi_2}$;		**Trans(10)** $\dfrac{q \to Q\mathrm{G}\varphi}{q \to p, p \to \varphi, p \to Q\mathrm{X}p}$;
Trans(9) $\dfrac{q \to Q(\varphi_1 \mathrm{W} \varphi_2) \text{ (如果}\varphi_2\text{ 不是文字)}}{q \to Q(\varphi_1 \mathrm{W} p), p \to \varphi_2}$;		
Trans(11) $\dfrac{q \to Q(\varphi \mathrm{U} l)}{q \to l \vee p, p \to \varphi, p \to Q\mathrm{X}(l \vee p), q \to Q\mathrm{F}l}$;		**Trans(12)** $\dfrac{q \to Q(\varphi \mathrm{W} l)}{q \to l \vee p, p \to \varphi, p \to Q\mathrm{X}(l \vee p)}$.

在表 2.1中，$T \in \{\mathrm{X}, \mathrm{G}, \mathrm{F}\}$，$ind$是规则中引入的新的索引且$Q \in \{\mathrm{A}, \mathrm{E}_{\langle ind\rangle}\}$；$q$是一个原子命题，$l$是一个文字，$D$是文字的析取（即子句），$p$是新的原子命题；$\varphi$，$\varphi_1$，和$\varphi_2$都是CTL公式。

规则**Trans(1)**和规则**Trans(2)**为每一个存在路径量词E引入一个新的索引ind；规则**Trans(3)**到规则**Trans(5)**通过引入新的替换规则将复杂的公式用新的原子命题替换；规则**Trans(6)**到规则**Trans(12)**用于移除掉那些不能出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$中的时序操作符[97]。

给定一个CTL公式φ，将其转换为一个$\mathrm{SNF}^{g}_{\mathrm{CTL}}$字句集合的主要步骤如下：

(1) 将公式CTL转换为其NNF（negation normal form）[1]形式，记为$nnf(\varphi)$；

[1]如果公式中的否定符号“¬”仅出现在原子命题之前，且联结符号只有“∨”和“∧”这两种，则称该公式

(2) 使用表 2.2中的等值公式化简$nnf(\varphi)$，得到$simp(nnf(\varphi))$；

(3) 使用表 2.1中的规则将$\{\mathrm{AG}(\mathbf{start} \to z), \mathrm{AG}(z \to simp(nnf(\varphi)))\}$化简为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合$T_\varphi$。形式化地，$T_\varphi$由导出（*derivation*）序列生成：

$$T_0 = \{\mathrm{AG}(\mathbf{start} \to p), \mathrm{AG}(p \to \mathbf{simp}(\mathbf{nnf}(\varphi)))\}, T_1, \ldots, T_n = T_\varphi$$

其中

- p是一个新的原子命题，*i.e.* $p \notin \{\mathbf{start}\} \cup Var(\varphi)$；
- $T_{t+1} = (T_t - \{\psi\}) \cup R_t$ $(t \geq 0)$，其中ψ为T_t中的非$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句，且$R_t$ 是使用一条匹配的归则作用到ψ上得到的结果；
- T_n中的每个公式都为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$字句形式。

表 2.2: CTL化简规则，其中$Q \in \{\mathrm{A}, \mathrm{E}\}$且$T \in \{\mathrm{X}, \mathrm{G}, \mathrm{F}\}$。

$(\varphi \wedge \top) \to \varphi$;	$(\varphi \wedge \bot) \to \bot$;	$(\varphi \vee \top) \to \top$;	$(\varphi \vee \bot) \to \varphi$;
$\neg\top \to \bot$;	$\neg\bot \to \top$;	$QT\bot \to \bot$;	$QT\top \to \top$;
$Q(\varphi \mathrm{U} \bot) \to \bot$;	$Q(\varphi \mathrm{U} \top) \to \top$;	$Q(\bot \mathrm{U} \varphi) \to \varphi$;	$Q(\top \mathrm{U} \varphi) \to Q\mathrm{F}\varphi$;
$Q(\varphi \mathrm{W} \bot) \to Q\mathrm{G}\varphi$;	$Q(\varphi \mathrm{W} \top) \to \top$;	$Q(\bot \mathrm{W} \varphi) \to \varphi$;	$Q(\top \mathrm{W} \varphi) \to \top$.

下文也将$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$记为由$\varphi$通过转换规则获得的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合。

引理 2.3. *给定CTL公式φ，*

(i) *对φ中的每一个路径量词E，有且仅有一个新的索引在转换过程中被引入，即$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$ 中对φ中的每一个E都有唯一一个带索引的存在路径量词。*

(ii) *$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$中不存在两个E-某时子句有相同的索引，即若$P_i \to \mathrm{E}_{\langle j_i \rangle}\mathrm{F}l_i$ $(i = 1,2)$在$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$ 中，则$j_1 \neq j_2$。*

(iii) *原子命题$p \in Var(\varphi)$不会出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$ 中的子句蕴含式的左手边。*

证明. (i) 显然规则**Trans(1)** 和**Trans(2)**为每一个E路径量词引入一个新的索引。再者，一旦路径量词E被索引标记了，它就不会被其他索引标记。

(ii) 由(i)可知，每个E被唯一的索引标记。此外在转换过程中不会产生新的E-某时子句。因此结论成立。

(iii) 从转换规则容易看出φ中的原子命题不会出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的左边。 □

是NNF形式的公式。

值得注意的是，每条转换规则的前件φ和结果ψ分别都是形如AGφ 和AGψ的公式。此外，由规则**Trans(11)**可知在$\mathrm{SNF}^{g}_{\mathrm{CTL}}(\varphi)$中，E-某时子句和E-步子句可能有相同的索引。

下面通过一个简单的例子[96]来展示上述转换步骤：

例 2.2. 令$\varphi = \neg\mathrm{AF}p \wedge \mathrm{AF}(p \wedge \top)$，下面给出将$\varphi$转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$的详细步骤。

(1) 将公式φ转换为其NNF形式：$\mathrm{EG}\neg p \wedge \mathrm{AF}(p \wedge \top)$；

(2) 化简(1)中的公式为：$\mathrm{EG}\neg p \wedge \mathrm{AF}p$；

(3) 使用表 2.1中的规则转化$\{\mathrm{AG}(\mathbf{start} \rightarrow z), \mathrm{AG}(z \rightarrow (\mathrm{EG}\neg p \wedge \mathrm{AF}p))\}$，详细步骤如下：

1. $\mathbf{start} \rightarrow z$
2. $z \rightarrow \mathrm{EG}\neg p \wedge \mathrm{AF}p$
3. $z \rightarrow \mathrm{EG}\neg p$ $(2, \mathbf{Trans(3)})$
4. $z \rightarrow \mathrm{AF}p$ $(2, \mathbf{Trans(3)})$
5. $z \rightarrow \mathrm{E}_{\langle 1\rangle}\mathrm{G}\neg p$ $(3, \mathbf{Trans(1)})$
6. $z \rightarrow x$ $(5, \mathbf{Trans(10)})$
7. $x \rightarrow \neg l$ $(5, \mathbf{Trans(10)})$
8. $x \rightarrow \mathrm{E}_{\langle 1\rangle}\mathrm{G}x$ $(5, \mathbf{Trans(10)})$
9. $\top \rightarrow \neg z \vee x$ $(6, \mathbf{Trans(5)})$
10. $\top \rightarrow \neg x \vee \neg p$ $(7, \mathbf{Trans(5)})$

因此，得到的φ对应的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$公式为：

1. $\mathbf{start} \rightarrow z$　2. $z \rightarrow \mathrm{AF}p$　3. $x \rightarrow \mathrm{E}_{\langle 1\rangle}\mathrm{G}x$　4. $\top \rightarrow \neg z \vee x$　5. $\top \rightarrow \neg x \vee \neg p$.

2.3.3 μ-演算

μ-演算是一种表达能力与S2S[2]相同的逻辑语言，线性时序逻辑（linear temporal logic，LTL）、CTL和CTL*能表达的性质都能用μ-演算来表示。μ-演算是模态逻辑的扩展，构成μ-演算语言的符号有：

- 原子命题符号的集合：$\mathscr{A}$；
- 变元符号的可数集：$\mathscr{V}$；

[2]无限完全二叉树下的一元二阶理论（monadic second order theory of the infinite complete binary tree），简称为S2S。

- 常量符号：$\bot$和$\top$；
- 布尔联结符号：$\vee$，$\wedge$，和$\neg$;
- 路径量词符号：A和E;
- 时序操作符号：X 用于用于表示“下一个状态”；
- 不动点符号：μ和ν，分别表示“最小不动点”和“最大不动点”。

通常认为AX和EX的优先级比布尔连接符高[98]，为了保证文章的统一性，本文规定各类符号之间的如下优先级（从左到右优先级逐渐变低）：

$$\neg \qquad \mathrm{EX} \qquad \mathrm{AX} \qquad \wedge \qquad \vee \qquad \mu \qquad \nu.$$

此时可如下定义μ-演算公式（简称为μ-公式或公式）：

$$\varphi := \top \mid \bot \mid p \mid \neg p \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathrm{EX}\varphi \mid \mathrm{AX}\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

其中$p \in \mathscr{A}$且$X \in \mathscr{V}$。AX和EX为模态操作。称出现在$\mu X.\varphi$和$\nu X.\varphi$中的变元X是*受约束的*（bound），不受约束的变元称为*自由变元*。原子命题和变元符号及其各自的否定称为*文字*，出现在公式φ中的原子命题的集合记为$Var(\varphi)$。

注意：μ-演算公式的定义通常考虑动作的集合Act和一组与集合Act中的动作a相关的模态词“$\langle a\rangle$”[99-101]。为了方便，本文考虑公式里只有一个动作的情形，但是本文的结论可以扩展到一般的情形。此时，模态词中的动作a可以省略，且公式$\mathrm{EX}\varphi$（或$\mathrm{AX}\varphi$）与公式$\langle a\rangle\varphi$（或$[a]\varphi$）[101]相同。

由上述定义可以看出，“$\neg$”符号只能出现在原子命题符号的前面。但在μ-演算公式的一般定义中，“$\neg$”符号可以出现在变元符号的前面，但是要求变元符号前的“$\neg$”符号的个数为偶数。尽管如此，这两种方式定义的公式具有相同的表达能力。

对于给定的公式φ，若出现在其中的自由变元与受约束变元不同，且每个变元最多被约束一次，则称公式φ是*取名恰当的*（well-named）。此外，若公式$\delta X.\varphi(X)$（$\delta \in \{\mu,\nu\}$）中变元X的每次出现都是在EX或AX的辖域[3]内，则称变元在公式$\delta X.\varphi(X)$中是*受保护的*（guarded）。一个没有自由变元出现的公式称为μ-*句子*（sentence）。*在本文中所谈到的公式指的是取名恰当的、受保护的*μ-*句子*。

与CTL公式类似，μ-演算公式的语义定义在Kripke结构上。但是，与CTL不同的是，这里不要求$\mathscr{M} = (S,R,L,r)$中的r为初始状态，且这里的r称为根（root），R为S上的任意二元关系，但仍然称$\mathscr{M} = (S,R,L,r)$为初始Kripke结构。

[3]给定公式$*\varphi$（$* \in \{\neg, \mathrm{EX}, \mathrm{AX}, \mu X, \nu X\}$），则称$\varphi$为$*$在公式$*\varphi$中的**辖域**。对于公式$\varphi * \psi$（$* \in \{\vee,\wedge\}$），则分别称$\varphi$和$\psi$为他们之间的$*$在$\varphi * \psi$中的左辖域和右辖域。

注意：虽然这里的Kripke结构不要求其二元关系是完全的，但是这里的情况更加一般化，其结论也能推广到完全的二元关系的情形。

定义 2.5. 给定μ-演算公式φ、初始*Kripke*结构$\mathscr{M}$和一个从$\mathscr{V}$中的变量到$\mathscr{M}$中状态的赋值函数$v:\mathscr{V}\to 2^S$。公式在$\mathscr{M}$和v上的解释是S的一个子集$\|\varphi\|_v^{\mathscr{M}}$（当$\mathscr{M}$在上下文中是显然的，则可以省去上标）：

$$
\begin{aligned}
&\|p\|_v=\{s\mid p\in L(s)\}\ ,\ \|\top\|_v=S\ ,\ \|\bot\|_v=\emptyset,\\
&\|\neg p\|_v=S-\|p\|_v,\\
&\|X\|_v=v(X),\\
&\|\varphi_1\vee\varphi_2\|_v=\|\varphi_1\|_v\cup\|\varphi_2\|_v,\\
&\|\varphi_1\wedge\varphi_2\|_v=\|\varphi_1\|_v\cap\|\varphi_2\|_v,\\
&\|\mathrm{EX}\varphi\|_v=\{s\mid\exists s'.(s,s')\in R\wedge s'\in\|\varphi\|_v\},\\
&\|\mathrm{AX}\varphi\|_v=\{s\mid\forall s'.(s,s')\in R\Rightarrow s'\in\|\varphi\|_v\},\\
&\|\mu X.\varphi\|_v=\bigcap\{S'\subseteq S\mid\|\varphi\|_{v[X:=S']}\subseteq S'\},\\
&\|\nu X.\varphi\|_v=\bigcup\{S'\subseteq S\mid S'\subseteq\|\varphi\|_{v[X:=S']}\}.
\end{aligned}
$$

其中，$v[X:=S']$是一个赋值函数，它除了$v[X:=S'](X)=S'$之外，和v完全相同。也即是，对任意的$Y\in\mathscr{V}$：

$$
v[X:=S'](Y)=\begin{cases}S', & \text{若}Y=X;\\ v(Y), & \text{否则。}\end{cases}
$$

直观地，$\|\varphi\|_v^{\mathscr{M}}$为在$\mathscr{M}$和$v$上$\varphi$为真的状态集合。称由$\mathscr{M}$和其赋值$v$构成的对$(\mathscr{M},v)$为赋值。在下文中，若$s\in\|\varphi\|_v$，则称$s$“满足”$\varphi$，记为$(\mathscr{M},s,v)\models\varphi$。此时，若$(\mathscr{M},r,v)\models\varphi$，则称$(\mathscr{M},r,v)$为$\varphi$的一个模型。当公式$\varphi$为$\mu$-句子时，可以将赋值函数$v$省略，记为$(\mathscr{M},s)\models\varphi$。记$Mod(\varphi)$为$\varphi$的模型的集合，即$Mod(\varphi)=\{(\mathscr{M},r)\mid(\mathscr{M},r,v)\models\varphi\}$（也可简写为$Mod(\varphi)=\{\mathscr{M}\mid(\mathscr{M},r,v)\models\varphi\}$）。其他记号与CTL情形类似，这里不再赘述。

由Tarski-Knaster定理可知，集合U子集上的单调函数f的最小和最大不动点可以由超穷归纳法得到，即最小不动点$\mu(f)=\bigcup\mu_\alpha(f)$，其中：

- $\mu_0(f)=\emptyset$，
- $\mu_{\alpha+1}=f(\mu_\alpha(f))$，$\alpha$为后继序数，
- $\mu_\lambda=\bigcup_{\alpha<\lambda}\mu_\alpha(f)$，$\lambda$为极限序数。

最大不动点$\nu(f)=\bigcap\nu_\alpha(f)$，其中

- $\nu_0(f) = U$，
- $\nu_{\alpha+1} = f(\nu_\alpha(f))$，$\alpha$为后继序数，
- $\nu_\lambda = \bigcap_{\alpha<\lambda} \nu_\alpha(f)$，$\lambda$为极限序数。

由公式的定义可知，变量前面不会出现¬符号，所以在$\mu X.\varphi(X)$中，$\varphi(X)$ 是关于X的单调函数。因此，$\|\mu X.\varphi(X)\|_v$是单调函数$\|\varphi\|_v : 2^S \to 2^S$（也写作$Y \mapsto \|\varphi\|_{v[X:=Y]}$）的最小不动点。下面给出如何使用上述定义计算最小不动点和最大不动点。

例 2.3. (1) 令$\varphi = \nu X.\psi$、$\mathscr{M} = (S,R,L,r)$为初始Kripke结构，其中$\psi = (j \wedge ch) \wedge \mathrm{AX}(\neg j \wedge \neg ch) \wedge \mathrm{AXAX}X$，则：

- $\nu_0(\|\psi\|) = S$，
- $\nu_1(\|\psi\|) = \|j\| \cap \|ch\| \cap \|\mathrm{AX}(\neg j \wedge \neg ch)\| \cap \|\mathrm{AXAX}\nu_0(\|\psi\|)\| = \{s \mid s \in L(j) \cap L(ch)$和$\forall t.(s,t) \in R, t \in S - (L(j) \cup L(ch))\}$，
- ...

(2) 令$\varphi = \nu X.\psi$、$\mathscr{M} = (S,R,L,r)$为初始Kripke结构，其中$\psi = (j \wedge ch) \wedge \mathrm{EX}(\neg j \wedge \neg ch) \wedge \mathrm{EXEX}X$，则：

- $\nu_0(\|\psi\|) = S$，
- $\nu_1(\|\psi\|) = \|j\| \cap \|ch\| \cap \|\mathrm{AX}(\neg j \wedge \neg ch)\| \cap \|\mathrm{AXAX}\nu_0(\|\psi\|)\| = \{s \mid s \in L(j) \cap L(ch)$和$\exists t.(s,t) \in R, t \in S - (L(j) \cup L(ch))\}$，
- $\nu_2(\|\psi\|) = \|j\| \cap \|ch\| \cap \|\mathrm{AX}(\neg j \wedge \neg ch)\| \cap \|\mathrm{AXAX}\nu_1(\|\psi\|)\|$，
- ...

当给定了初始Kripke结构，根据上述过程，不难计算出最大和最小不动点。

2.3.4 μ-公式的析取范式

Janin等人首先提出了μ-演算的析取范式[102]，后来被逐步完善，本文使用文章[80]中的析取μ-公式的定义。

在给出该定义之前，事先给出μ-公式的另一种定义，称为*覆盖-语法*（cover-syntax）。该定义是将上述μ-公式的定义中的EX用*覆盖操作*（cover operator）的集合来替换得到。在覆盖-语法中，

- $Cover(\emptyset)$是公式；
- 对任意的$n \geq 1$，若$\varphi_1, \ldots, \varphi_n$是公式，则$Cover(\varphi_1, \ldots, \varphi_n)$是公式。

对于给定的初始结构$\mathscr{M} = (S, R, L, r)$和赋值函数$v$：

- $(\mathscr{M}, r, v) \models Cover(\emptyset)$当且仅当$r$没有任何的后继状态；
- $(\mathscr{M}, s, v) \models Cover(\varphi_1, \ldots, \varphi_n)$当且仅当
  - 对任意的$i = 1, ..., n$，存在$(s,t) \in R$使得$(\mathscr{M}, t, v) \models \varphi_i$;
  - 对任意的$(s,t) \in R$，存在$i \in \{1, ..., n\}$使得$(\mathscr{M}, t, v) \models \varphi_i$。

尽管覆盖-语法在形式上与上一小节中μ-公式的定义有所不同，但是已经证明这两种定义是等价的[80]。这主要因为$Cover$公式与EX公式之间可以通过下面的等式转换：

$$Cover(\varphi_1, \ldots, \varphi_n) \Leftrightarrow \mathrm{EX}\varphi_1 \wedge \cdots \wedge \mathrm{EX}\varphi_n \wedge \mathrm{AX}(\varphi \vee \cdots \vee \varphi_n),$$

反之，

$$\mathrm{EX}\varphi \Leftrightarrow Cover(\varphi, \top).$$

基于此，可以给出析取μ-公式的形式定义如下：

定义 2.6 (析取μ-公式[80]). 析取μ-公式的集合$\mathscr{F}_d$是包含$\top$、$\bot$和不矛盾的文字的合取且封闭于下面几条规则的最小集合:

(1) 析取式（*disjunctions*）：若$\alpha, \beta \in \mathscr{F}_d$，则$\alpha \vee \beta \in \mathscr{F}_d$；

(2) 特殊合取式（*special conjunctions*）：若$\varphi_1, \ldots, \varphi_n \in \mathscr{F}_d$且$\delta$为不矛盾的文字的合取，则$\delta \wedge Cover(\varphi_1, \ldots, \varphi_n) \in \mathscr{F}_d$；

(3) 不动点操作（*fixpoint operators*）：若$\varphi \in \mathscr{F}_d$，且对任意的公式$\psi$，$\varphi$不含有形如$X \wedge \psi$的子公式，则$\mu X.\varphi$和$\nu X.\varphi$都在$\mathscr{F}_d$中。

例 2.4. 容易检查$\nu X.(j \wedge ch) \wedge \mathrm{EX}(\neg j \wedge \neg ch) \wedge \mathrm{EXEX}X$ 和$\nu X.(j \wedge ch) \wedge \mathrm{AX}(\neg j \wedge \neg ch) \wedge \mathrm{AXAX}X$ 都不是析取μ-公式。而$j \wedge ch \wedge \mathrm{EX}(\neg j \wedge \neg ch)$、$\mu X.(j \wedge ch) \wedge \mathrm{EX}X$ 和$\nu X.(j \wedge ch) \wedge \mathrm{EXEX}X$ 都为析取μ公式，因为：

$$j \wedge ch \wedge \mathrm{EX}(\neg j \wedge \neg ch) \equiv j \wedge ch \wedge Cover(\neg j \wedge \neg ch, \top),$$

$$\mu X.(j \wedge ch) \wedge \mathrm{EX}X \equiv \mu X.(j \wedge ch) \wedge Cover(X, \top),$$

且

$$\nu X.(j \wedge ch) \wedge \mathrm{EXEX}X \equiv \nu X.(j \wedge ch) \wedge Cover(Cover(X, \top), \top).$$

表 2.3: 归结规则

(SRES1) $\dfrac{P \to \mathrm{AX}(C \vee l), Q \to \mathrm{AX}(D \vee \neg l)}{P \wedge Q \to \mathrm{AX}(C \vee D)}$;	**(SRES2)** $\dfrac{P \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee l), Q \to \mathrm{AX}(D \vee \neg l)}{P \wedge Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)}$;
(SRES3) $\dfrac{P \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee l), Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(D \vee \neg l)}{P \wedge Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)}$;	**(SRES4)** $\dfrac{\mathbf{start} \to C \vee l, \mathbf{start} \to D \vee \neg l}{\mathbf{start} \to C \vee D}$;
(SRES5) $\dfrac{\top \to C \vee l, \mathbf{start} \to D \vee \neg l}{\mathbf{start} \to C \vee D}$;	**(SRES6)** $\dfrac{\top \to C \vee l, Q \to \mathrm{AX}(D \vee \neg l)}{Q \to \mathrm{AX}(C \vee D)}$;
(SRES7) $\dfrac{\top \to C \vee l, Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(D \vee \neg l)}{Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)}$;	**(SRES8)** $\dfrac{\top \to C \vee l, \top \to D \vee \neg l}{\top \to C \vee D}$;
(RW1) $\dfrac{\bigwedge_{i=1}^{n} m_i \to \mathrm{AX}\ \bot}{\top \to \bigvee_{i=1}^{n} \neg m}$;	**(RW2)** $\dfrac{\bigwedge_{i=1}^{n} m_i \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}\ \bot}{\top \to \bigvee_{i=1}^{n} \neg m}$;
(ERES1) $\dfrac{\Lambda \to \mathrm{EXEG}l, Q \to \mathrm{AF}\neg l}{Q \to \mathrm{A}(\neg \Lambda \mathrm{W} \neg l)}$;	**(ERES2)** $\dfrac{\Lambda \to \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{G}l, Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{F}\neg l}{Q \to \mathrm{E}_{\langle ind \rangle}(\neg \Lambda \mathrm{W} \neg l)}$.

2.4　CTL下的归结

归结是一种用于判定给定的命题公式（或一阶公式）是否可满足的规则，该技术可以追溯到1960年Davis等的工作[103]，之后被Robinson加以完善[9]。对于给定的公式，归结给出一个反驳定理证明过程。

在看见了归结在在命题逻辑和一阶逻辑中取得成就之后，科研工作者们开始将精力致力于其他非经典逻辑中，并取得了相当显著的理论成果，如：模态逻辑（K系统，Q系统，T系统，S4和S5系统）中的归结[104]和时序逻辑（尤其是线性时序逻辑（LTL）和CTL）中的归结[94,105]。

这里主要介绍与本文直接相关的CTL下的归结。CTL下的归结起源于BolotovF的研究[94]，之后被Zhang等人完善[96]。不论是在BolotovF的工作还是在Zhang等人的工作中，其关键点都是将CTL公式转换为一个$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合。本文使用Zhang等人提出的规则[96]，如表 2.3所示。

在表 2.3中P和Q是文字的合取，C和D是文字的析取，l是一个文字，称每条规则横线下面的公式为横线上面的公式关于文字l的归结结果。此外，$\Lambda = \bigvee_{i=1}^{n} \bigwedge_{i=1}^{m_i} P_j^i$、$P_j^i$是文字的析取，其中$1 \leq i \leq n$和$1 \leq j \leq m$。

规则**SRES1-8**称为步-归结规则（step resolution rule）、**RW1-2**称为重写规则（rewrite rule）、**ERES1-2**称为可能归结规则（eventuality resolution rule）。值得注意的是，规则**ERES1** 的前提“$\Lambda \to \mathrm{EXEG}l$”表示如下子句的集合$\Lambda_{\mathrm{EG}}$：

$$
\begin{array}{lcl}
P_1^1 \to *C_1^1, & & P_1^n \to *C_1^n, \\
\vdots & & \vdots \\
P_{m_1}^1 \to *C_{m_1}^1, & \cdots & P_{m_n}^n \to *C_{m_n}^n,
\end{array}
$$

其中，对任意的i $(1 \leq i \leq n)$，

- 存在一个索引$ind \in \mathscr{I}$ 使得$*$ 要么为空符号，要么为$\{\mathrm{AX}, \mathrm{E}_{\langle ind \rangle}\mathrm{X}\}$中的一个，
- $\left(\bigwedge_{j=1}^{m_i} C_j^i\right) \rightarrow l$成立，且
- $\left(\bigwedge_{j=1}^{m_i} C_j^i\right) \rightarrow \left(\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} P_j^i\right)$成立。

上面的最后两个条件确保了子句集合Λ_{EG}能够蕴涵$\Lambda \rightarrow \mathrm{EXEG}l$。规则**ERES2**的第一个前提与**ERES1**的类似。**ERES1**的结果能通过表 2.1中的转换规则转换成等价可满足的全局和A-步子句的集合：

$$\{w_{\neg l}^{\mathrm{A}} \rightarrow \mathrm{AX}(\neg l \vee \bigvee_{j=1}^{m_i} \neg P_j^i) \mid 1 \leq i \leq n\},$$
$$\{\top \rightarrow \neg Q \vee \neg l \vee \bigvee_{j=1}^{m_i} \neg P_j^i \mid 1 \leq i \leq n\},$$
$$\{\top \rightarrow \neg Q \vee \neg l \vee w_{\neg l}^{\mathrm{A}}, w_{\neg l}^{\mathrm{A}} \rightarrow \mathrm{AX}(\neg l \vee w_{\neg l}^{\mathrm{A}})\}.$$

ERES2的结果则通过表 2.1中的规则转换成等价可满足的全局和E-步子句的集合：

$$\{w_{\neg l}^{ind} \rightarrow \mathrm{E}_{\langle ind \rangle}\mathrm{X}(\neg l \vee \bigvee_{j=1}^{m_i} \neg P_j^i) \mid 1 \leq i \leq n\},$$
$$\{\top \rightarrow \neg Q \vee \neg l \vee \bigvee_{j=1}^{m_i} \neg P_j^i \mid 1 \leq i \leq n\},$$
$$\{\top \rightarrow \neg Q \vee \neg l \vee w_{\neg l}^{\mathrm{A}}, w_{\neg l}^{ind} \rightarrow \mathrm{E}_{\langle ind \rangle}\mathrm{X}(\neg l \vee w_{\neg l}^{ind})\}.$$

值得注意的是在转换**ERES1-2**的结果为子句集合的过程中会引入一个新的原子命题，即$w_{\neg l}^{\mathrm{A}}$和$w_{\neg l}^{ind}$[96]。从这个角度来看，每个归结规则的前件和结果都是子句形式。

对于给定的CTL公式，使用上述的归结规则可以导出一个子句的集合。形式化地说，源于$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集合$S$的一个推导（derivation）是一个满足如下条件的$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集合的序列$S_0, S_1, S_2, \ldots,$：

- $S_0 = S$，且
- $S_{i+1} = S_i \cup \{\alpha\}$ $(i \geq 0)$，其中$\alpha \notin S_i$是对S_i的某些子句使用一条归结规则得到的结果。

$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集合$S$的一个反驳是一个源于$S$的推导$S_0, S_1, S_2, \ldots, S_i$，且$S_i$（$i \geq 0$）中包含一个矛盾——公式$\top \rightarrow \bot$ 或**start** $\rightarrow \bot$。

为了判定CTL公式φ的可满足性，基于归结的判定过程用于检查T_φ是否有反驳存在。定理5.6、5.30和6.1[96]已经证明这一过程是可靠和完备的。因而，下面的推论显然成立。

推论 2.1. *给定两个CTL公式φ和ψ。则$\varphi \models \psi$ 当且仅当$T_{\varphi\wedge\neg\psi}$有一个反驳。*

例 2.5 (例 2.2的扩展)**.** 对例 2.2中的子句使用表 2.3中的归结规则，得到如下子句：

(1) **start** $\rightarrow x$	$(1,4,\mathbf{SRES5})$
(2) $w \rightarrow \mathrm{AX}(p \vee \neg x)$	$(2,3,5,\mathbf{ERES1})$
(3) $\top \rightarrow \neg z \vee p \vee \neg x$	$(2,3,5,\mathbf{ERES1})$
(4) $\top \rightarrow \neg z \vee l \vee w$	$(2,3,5,\mathbf{ERES1})$
(5) $w \rightarrow \mathrm{AX}(x \vee w)$	$(2,3,5,\mathbf{ERES1})$
(6) $\top \rightarrow \neg z \vee \neg x$	$(5,(3),\mathbf{SRES8})$
(7) **start** $\rightarrow \neg x$	$(1,(6),\mathbf{SRES5})$
(8) **start** $\rightarrow \bot$	$((1),(7),\mathbf{SRES4})$.

由于在这一推导中有一个子句集合包含一个矛盾，即：**start** $\rightarrow \bot$，所以T_φ存在一个反驳。因此，φ是不可满足的。

2.5　遗忘理论和SNC（WSC）

这部分主要介绍遗忘理在经典逻辑和模态逻辑S5下的定义，以及基于遗忘的SNC（WSC）的计算方法。

2.5.1　经典逻辑的遗忘

遗忘一词起源于经典逻辑（包括命题逻辑和一阶逻辑）[33]，给定一个命题公式φ和一个原子命题p，下面将介绍如何从φ中遗忘（forget）掉p。在第1.2.2节中说过，从φ中遗忘掉p 得到的结果为$Forget(\varphi,\{p\}) \equiv \varphi[p/\top] \vee \varphi[p/\bot]$。

例 2.6. 某学校有a和b两个食堂，学生要么去a食堂吃饭要么去b食堂吃饭，如果想吃烤鱼（fish，f）就去a食堂吃饭，如果想吃炒饭（rice，r）就去b食堂吃饭。这一知识可表示为命题公式$\varphi = (a \vee b) \wedge (f \rightarrow a) \wedge (r \rightarrow b)$。如果此时不考虑鱼，即：由于某种原因$a$食堂就不再卖烤鱼了，此时就应该“遗忘”烤鱼。这一计算过程表示如下：

$$\begin{aligned} Forget(\varphi,\{f\}) &\equiv \varphi[f/\top] \vee \varphi[f/\bot] \\ &\equiv [(a \vee b) \wedge (\top \rightarrow a) \wedge (r \rightarrow b)] \vee [(a \vee b) \wedge (\bot \rightarrow a) \wedge (r \rightarrow b)] \end{aligned}$$

$$\equiv [(a \vee b) \wedge a \wedge (r \rightarrow b)] \vee [(a \vee b) \wedge (r \rightarrow b)]$$
$$\equiv (a \vee b) \wedge (r \rightarrow b).$$

直观上来看，这个结果应该比原始的公式φ还要弱，但是能够蕴含同样的任何不包含f的句子（sentence），也就是说遗忘只能影响与f相关的语义。这一性质可由互模拟这一词来表示。对于解释之间的互模拟来说，给定原子命题p，如果对任意的$q \in \mathscr{A} - \{p\}$有$q \in I_1$当且仅当$q \in I_2$，则称解释$I_1$与$I_2$是$p$互模拟的，记为：$I_1 \sim_p I_2$。

在一阶逻辑中，一阶逻辑语言$\mathscr{L}_f$的解释有两种：$\mathscr{L}_f$和结构有联系或没有联系，互模拟的定义就要困难一些。这里考虑和结构有联系的情形，一个一阶结构由论域（domain）、指定的个体、关系和函数构成。此时，$\mathscr{L}_f$中的个体符合、n-元关系符号和m-元函数符号分别被解释为这个结构中指定的论域中的个体、论域上的n-元关系和m-元全函数（即处处有定义的函数）。对于给定的一阶结构M和$X \in$ {个体符号，元组，关系符号，函数符号}，$M[X]$表示结构M对X的解释，且$M[(a_1, a_2, \ldots, a_i)] = (M[a_1], M[a_2], \ldots, M[a_i])$。

给定实例化（ground atom）原子$P(\vec{t})$（$\vec{t}$是一个n元组）、M_1和M_2为一阶结构，则$M_1 \sim_{P(\vec{t})} M_2$ 当且仅当除了$P(\vec{t})$的真值M_1和M_2相同，即：

(i) M_1和M_2有相同的论域，且每个函数符号被解释成相同的函数；

(ii) 对于和P不同的任意关系符号Q，$M_1[Q] = M_2[Q]$;

(iii) 令$\vec{u} = M_1[\vec{t}]$，则对于该论域中任意与$\vec{u}$不同的元组$\vec{d}$，$\vec{d} \in M_1[P]$ 当且仅当$\vec{d} \in M_2[P]$。

一阶逻辑中遗忘实例化原子的形式化定义[33]为：

定义 2.7. *给定一个句子（sentence）φ和实例化原子p，φ'是从φ中遗忘掉p的结果当且仅当对任意的结构M，M是φ'的模型当且仅当存在一个φ的模型M'使得$M \sim_p M'$。*

从句子φ中遗忘掉实例化原子$P(\vec{t})$比命题逻辑下的遗忘多了一步，即事先将φ中的所有$P(\vec{t'})$的出现用$(\vec{t} = \vec{t'} \wedge P(\vec{t})) \vee (\vec{t} \neq \vec{t'} \wedge P(\vec{t'}))$来替换，这一结果记为$\varphi[P(\vec{t})]$。

例 2.7. 令$\varphi = J(mo) \vee J(fa) \vee B(sm)$、$p = J(mo)$，则：

$$\varphi[p] \equiv (mo = mo \wedge J(mo)) \vee (mo \neq mo \wedge J(mo)) \vee$$
$$(mo = fa \wedge J(mo)) \vee (mo \neq fa \wedge J(fa)) \vee B(sm).$$

$$Forget(\varphi, p) \equiv \varphi[p][p/\top] \vee \varphi[p][p/\bot]$$

$$\equiv (mo = mo \wedge \top) \vee (mo \neq mo \wedge \top) \vee (mo = fa \wedge \top) \vee (mo \neq fa \wedge J(fa)) \vee B(sm)$$
$$\vee (mo = mo \wedge \bot) \vee (mo \neq mo \wedge \bot) \vee (mo = fa \wedge \bot) \vee (mo \neq fa \wedge J(fa)) \vee B(sm)$$
$$\equiv (mo = mo) \vee (mo \neq mo) \vee (mo = fa) \vee (mo \neq fa \wedge J(fa)) \vee B(sm).$$

然而，遗忘掉一整个关系（谓词）“P”而不是其实例得到的结果是一个二阶公式，且结构间在谓词上的互模拟与上述在实例上的有所不同：对于谓词P和结构M_1、M_2，$M_1 \sim_P M_2$ 当且仅当：

(i) M_1和M_2有相同的论域，且每个函数符号被解释成相同的函数；

(ii) 对于和P不同的任意关系符号Q，$M_1[Q] = M_2[Q]$。

也即是排除了实例情形下的第三个条件，因为此时考虑的是整个谓词。而遗忘掉谓词的定义与遗忘掉实例的定义类似，只是将$M \sim_p M'$变为$M \sim_P M'$。

研究表明，从句子φ中遗忘掉谓词P的结果为$Forget(\varphi, P) = (\exists R)\varphi[P/R]$[33]，其中$P$是$n$-元谓词，$R$是$n$-元谓词变量。正如前面所说的，一阶逻辑下的遗忘不是封闭的，此时不一定能找到一个与$(\exists R)\varphi[P/R]$等价的一阶公式。

本文采用了基于归结的方法来计算CTL中的遗忘，因此这里给出命题逻辑下基于归结的遗忘定义[106]。

定义 2.8. 给定命题公式φ和原子命题p，

$$Forget(\varphi, p) = \{C \in CNF(\varphi) \mid p \text{ 不出现在} C \text{中}\} \cup Res(CNF(\varphi), p)$$

其中$CNF(\varphi)$表示形成φ的合取范式的子句构成的集合，$Res(S, p) = \{C_1 \vee C_2 \mid C_1 \vee p, C_2 \vee \neg p \in S\}$。

从定义2.8不难看出计算从φ中遗忘p的结果可以分为三个步骤：（1）计算φ的合取范式，并得到$CNF(\varphi)$；（2）计算$Res(CNF(\varphi), p)$；（3）去除$CNF(\varphi)$包含p的子句。遗忘的定义种类很多，本文的定义采用的是上述所说的互模拟方式，因此这里不再赘述其他定义，感兴趣的读者可以参考Eiter的文章[61]。

在描述逻辑中，如果遗忘的结果是可以用当前讨论的描述逻辑来表示的，则该结果就是一个均匀插值。而判定均匀插值的存在性通常是很费时的，如：在$\mathscr{ALC}$和$\mathscr{EL}$中是双指数时间的。因此，不难看出描述逻辑中的遗忘通常也是很困难的，尽管如此也有很多方法克服这些问题，其中扩展描述语言（如：从$\mathscr{ALC}$到$\mathscr{ALC}_v$[107]）或引入新的辅助符号[108]是常用的方法。一些计算遗忘的工具是：基于skolem化和SOQE的SCAN [4]、基于归结的Lethe[109]和基于Ackermann引理的FAME[110]。

[4]http://www.mettel-prover.org/scan/index.html

2.5.2 模态逻辑S5的遗忘

由于时序逻辑是模态逻辑的一种，其语义是Kripke语义，这里介绍与其密切相关且基础的模态逻辑S5中的遗忘。与经典逻辑中的遗忘相似，S5中的遗忘也是用互模拟的概念来定义。

原子命题的集合w_1和w_2是V-互模拟的，当且仅当$w_1 - V = w_2 - V$，记为$w_1 \sim_V w_2$，其中$w_1, w_2, V \subseteq \mathscr{A}$。给定原子命题的集合$V \subseteq \mathscr{A}$、两个κ-解释$\mathscr{M} = \langle W, w\rangle$和$\mathscr{M}' = \langle W, s\rangle$，则称$\mathscr{M}$和$\mathscr{M}'$是$V$-互模拟的（记为$\mathscr{M} \leftrightarrow_V \mathscr{M}'$）[45]，当且仅当存在一个二元关系$\sigma \subseteq W \times W'$使得$(w, w') \in \sigma$，且：

(i) $\forall w_1 \in W$，$\exists w_2 \in W'$使得$(w_1, w_2) \in \sigma$；

(ii) $\forall w_2 \in W'$，$\exists w_1 \in W'$使得$(w_1, w_2) \in \sigma$；

(iii) 若$(w_1, w_2) \in \sigma$，则$w_1 \sim_V w_2$。

条件(i)和(ii)分别称为前向条件（forth condition）和后向条件（back condition）。需要注意的是，即使M和M'有V-互模拟关系，M和M'也可能有不同数量的世界个数。除此之外，从定义中不难看出，如果$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，则有$Atom(W) - V = Atom(W') - V$，其中$Atom(X)$（$X$是可能世界的集合）是由出现在$X$中的世界中的原子构成的集合。从定义中还可得出$\leftrightarrow_V$是一个等价关系。

S5关于V-互模拟是不变的（invariant）：如果两个κ-解释M和M'有V-互模拟关系，那么对于任何不包含V中任何原子的公式φ，M和M'同时满足或不满足公式φ。S5中的知识遗忘（knowledge forgetting）定义如下[45]：

定义 2.9 (knowledge forgetting). *给定模态S5公式φ和$V \subseteq \mathscr{A}$。如果下面的等式成立，则称知识集$KForget(\varphi, V)$是从φ遗忘掉V得到的结果：*

$$Mod(KForget(\varphi, V)) = \{\mathscr{M}' \mid \exists \mathscr{M} \in Mod(\varphi), \mathscr{M} \leftrightarrow_V \mathscr{M}'\}.$$

Zhang等人还提出了能精确描述知识遗忘的四个基本条件（公设），给定两个公式φ和$\varphi' = KForget(\varphi, V)$，$V \subseteq \mathscr{A}$是原子命题的集合。知识遗忘满足以下的性质：

(**W**) 削弱（Weaking）：$\varphi \models \varphi'$；

(**NP**) 正支持（Positive Persistence）：如果IR(ϕ, V)并且$\varphi \models \phi$，则$\varphi' \models \phi$；

(**PP**) 负支持（Negative Persistence）：如果IR(ϕ, V)并且$\varphi \not\models \phi$，则$\varphi' \not\models \phi$；

(**IR**) 无关性（Irrelevance）：IR(φ', V)。

直观地说，(**W**)和(**IR**)表明“遗忘”削弱了公式φ且得到的结果与V无关，(**PP**)和(**NP**)表明对任意与V无关的公式ϕ，$\varphi \models \phi$当且仅当$\varphi' \models \phi$。总而言之，遗忘得到的结果能推出所有与V无关且能被φ推出的结果，且不能推出所有与V无关且不能被φ推出的结果。从数据库和安全的层面讲，遗忘相当于从已有的关系表中构建出一个视图，达到了隐私保护的作用。

这四个条件与知识遗忘的关系为[45]：

定理 2.1. *给定公式φ和φ'，$V \subseteq \mathscr{A}$为原子命题的集合。下面的陈述是等价的：*

(i) $\varphi' \equiv KForget(\varphi, V)$,

(ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi \text{且} IR(\phi, V)\}$,

*(iii) 若φ、φ'和V为(i)和(ii)中提到的符号，则公设(**W**)、(**PP**)、(**NP**)和(**IR**)成立.*

在本文中也将说明CTL和μ-演算的遗忘也有上述性质。值得注意的是任意的S5公式都能转换为与之等价的模态合取范式（MCNF）[111]，其模态子句形式为：

$$C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n,$$

或具有如下形式的公式的析取——模态析取范式（MDNF）[45,47]:

$$\varphi_0 \wedge \mathrm{K}\varphi_1 \wedge \mathrm{B}\varphi_2 \wedge \cdots \wedge \mathrm{B}\varphi_n \tag{2.10}$$

其中φ_i（$0 \leq i \leq n$）为命题逻辑公式，C_i（$0 \leq i \leq n-1$）为经典子句，C_n为CNF公式，且任意φ_i和C_i都可能缺失。从子句2.10遗忘掉原子命题p可以转换成命题逻辑中的遗忘，即：

$$\begin{aligned} &KForget(\varphi_0 \wedge \mathrm{K}\varphi_1 \wedge \mathrm{B}\varphi_2 \wedge \cdots \wedge \mathrm{B}\varphi_n) \\ &\equiv Forget(\varphi_0, \{p\}) \wedge \mathrm{K}(Forget(\varphi_1, \{p\})) \wedge \bigwedge_{i=1}^{n} \mathrm{B}(Forget(\varphi_1 \wedge \varphi_i, \{p\})). \end{aligned}$$

由此可以得出任意S5公式下的遗忘都能转换为命题逻辑中的遗忘，而命题逻辑下的遗忘已有算法和实现，这将在计算SNC和WSC部分给出。

2.5.3　遗忘的计算方法

在第 2.5.1和2.5.2小节中详细介绍了经典逻辑和模态逻辑S5下遗忘的定义和一些直接的计算方法。总的来说，这些方法分为两类：“代替”的方法和归结的方法。其中“代替”法是将要遗忘的原子命题在公式里用“$\top$”或“$\bot$”代替，归结的方法主要使

用归结规则来“消除”需要遗忘的原子命题。然而，在上文中并没有对归结方法进行详细的介绍，所以这部分给出命题情形下归结方法的详细描述。

归结方法取决于子句的形式，子句的形式决定了归结规则的复杂性。经典命题逻辑中的子句形式比较单一，就只有一种——文字的析取，因此归结规则就比较简单，即：

$$\frac{C_1 \vee p \qquad C_2 \vee \neg p}{C_1 \vee C_2},$$

其中C_1和C_2是子句，p是原子命题。在这种情况下，基于归结的方法就如定义 2.8那样简单。在一阶逻辑中，将公式转换为子句形式的过程比较复杂，而归结规则也相对复杂一些。但是在一阶情形下归结，归结系统R[43]是可靠的且归结反驳是完备的。

在上一节中已经说明任意的S5公式能够转化成模态子句$C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n$的合取，因此S5下的归结系统RS5[104]如下：

$$(\mathrm{KB})\frac{C \vee \mathrm{K}(l \vee D) \qquad C' \vee \mathrm{B}(\neg l \vee D', E)}{C \vee C' \vee \mathrm{B}(D \vee D', \neg l \vee D', E)}; \qquad (\mathrm{K}\bot)\frac{C \vee \mathrm{K}\bot}{C};$$

$$(\mathrm{KK})\frac{C \vee \mathrm{K}(l \vee D) \qquad C' \vee \mathrm{K}(\neg l \vee D')}{C \vee C' \vee \mathrm{K}(D \vee D')}; \qquad (\mathrm{B}\bot)\frac{C \vee \mathrm{B}(\bot, E)}{C};$$

$$(\mathrm{K})\frac{C \vee \mathrm{K}(l \vee D) \qquad C' \vee \neg l}{C \vee C' \vee D}; \qquad (Clas)\frac{C \vee l \qquad C' \vee \neg l}{C \vee C'};$$

$$(\mathrm{B})\frac{C \vee \mathrm{B}(l \vee D, \neg \vee D', E)}{C \vee \mathrm{B}(D \vee D', l \vee D, \neg \vee D', E)}; \qquad (Fact)\frac{E[D \vee D \vee C])}{E[D \vee C]}.$$

其中l为文字，C、C'、D、D'为子句，E为子句的集合；对于子句的集合S，$\mathrm{B}(S)$表示$\mathrm{B}(\bigwedge S)$；$E[\psi]$表示ψ是E的子公式。

基于上述归结系统RS5，计算遗忘的算法被提出。为了更清楚地描述该算法，这里还需要介绍两个概念：模态子句的包蕴（subsume）和清除（suppressing）。给定两个模态子句$C = C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n$和$C' = C'_0 \vee \mathrm{K}C'_1 \vee \cdots \vee \mathrm{K}C'_{m-1} \vee \mathrm{B}C'_m$，如果满足下面三个条件，则说$C$包蕴$C'$：

- C_0包蕴C'_0，即$Lit(C) \subseteq Lit(C')$；
- $\forall C_i$（$1 \leq i \leq n-1$），$\exists C'_j$（$1 \leq j \leq m-1$）使得C_i包蕴C_j；
- 对C'_m中的任意合取项e'，存在C_n中的一个合取项e使得e包蕴e'。

其中$Lit(X)$为出现在X中文字的集合。

“清除”操作主要是用于移除那些包含要遗忘的原子命题的公式。具体地，令ϕ为子句，$V \subseteq \mathscr{A}$为原子命题的集合，ϕ在V上的清除操作记为$Supp(V,\phi)$，且：

$$Supp(V,\phi) = \begin{cases} \top, & \text{若存在}V\text{中的元素}p\text{使得}p \in Var(\phi); \\ \phi, & \text{否则。} \end{cases}$$

令$\phi = C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n$为模态子句，$\phi$在$V$上的清除操作也记为$Supp(V,\phi)$，且：

$$Supp(V,C_0) \vee \left(\bigvee_{1 \le i \le n-1} \mathrm{K}Supp(V,C_i) \right) \vee \mathrm{B} \left(\bigwedge_{\alpha \text{ is a conjunct of } C_n} Supp(V,\alpha) \right).$$

模态S5下基于归结的算法如算法 2.1所示。在该算法中，第7-9行用于移除具有形式$p \vee D$或$\neg p \vee D$（$p \in V$）的子句，以免产生无用的结果，因为这些结果在第11行也会被移除。

算法 2.1 S5下基于归结的遗忘计算

输入：

Γ, V：　S5公式，原子命题的集合

输出：

$KForget(\Gamma, V)$：从Γ中遗忘掉V中原子的结果

1: 将Γ转换为模态子句的集合Γ'；
2: $\Gamma_2 = \{C \mid C \in \Gamma', Var(C) \cap V = \emptyset\}$，$\Gamma_1 = \Gamma' - \Gamma_2$
3: **if** $V = \emptyset$ **then**
4:　　跳转到11;
5: **end if**
6: 从V中随机选择一个原子p，且令$V = V - \{p\}$;
7: 化简Γ_1（$C_1, C' \in \Gamma$）：
8:　　若C'包蕴C_1，则从Γ_1中删除C_1；
9:　　若C_1形如$p \vee D$或$\neg p \vee D$，则从Γ_1中删除C_1（D为模态子句）
10: 跳转到2;
11: $\Gamma_3 = \{Supp(V,\phi) \mid \phi \in \Gamma_1\}$;
12: **return** $\Gamma \cup \Gamma_3$.

2.5.4　基于遗忘的SNC（WSC）计算

SNC和WSC的定义最先由Lin提出[77]，这部分给出其在命题逻辑和一阶逻辑下的形式化定义和计算方法。

定义 2.10. 令φ是一个命题公式，$V \subseteq \varphi$，q是一个出现在φ中但是不出现在V中的命题。对于V上的公式ϕ，若$\varphi \models q \rightarrow \phi$（$\varphi \models \phi \rightarrow q$），则称公式$\phi$是$q$在$V$和$\varphi$上的必要条件（充分条件）。如果对于任意$q$在$V$和$\varphi$上的必要条件（充分条件）$\phi'$都有$\varphi \models \phi \rightarrow \phi'$（$\varphi \models \phi' \rightarrow \phi$），则称$\phi$是$q$在$V$和$\varphi$上的最强必要条件（最弱充分条件）。

SNC和WSC是一个对偶概念，且任意公式的SNC（WSC）都能转换成原子命题的形式计算[77]，因此这里只讨论原子命题情形下的SNC的定义及其计算。

定理 2.2. 给定命题公式φ、原子命题集合$V \subseteq Var(\varphi)$和原子命题$q \in (Var(\varphi) - V)$。令$V' = Var(\varphi) - (V \cup \{q\})$，则

- q在V和φ上的*SNC*是$Forget(\varphi[q/\top], V')$；
- q在V和φ上的*WSC*是$\neg Forget(\varphi[q/\bot], V')$。

定理 2.2表明可以用遗忘计算SNC和WSC。基于遗忘的计算SNC（WSC）的详细算法为算法 2.2，其中一个子句集合的极小集（minimal set of clauses）为满足下面性质的集合：

- 所有的单元子句都被替换为$\top$；
- 没有一个子句被集合中的另一个子句包蕴。

此外，对于公式集合S，$S[X/Y]$为将S中每个公式中X的出现全都替换成Y，即$S[X/Y] = \{\varphi[X/Y] \mid \varphi \in S\}$。

在一阶逻辑中，SNC和WSC的定义和命题逻辑下相似，也可用遗忘来计算[78]。不同的是，一阶逻辑中的遗忘计算比较复杂，且遗忘的结果不一定能用一阶语言表示出来。

定理 2.3. 对任意一阶公式α、关系符号的集合P和句子Th:

- α在P和Th上的*SNC*为$\exists\overline{\Phi}.[Th \wedge \alpha]$,
- α在P和Th上的*WSC*为$\forall\overline{\Phi}.[Th \rightarrow \alpha]$,

其中$\overline{\Phi}$是出现在$Th \wedge \alpha$且不出现在P中的关系符号的集合。

正如前面所说，一阶逻辑下的遗忘主要使用归结和SOQE的方法来计算，但由于本文不涉及相关知识，所以这里就不详细介绍一阶逻辑下遗忘的计算。

在模态逻辑S5中，SNC和WSC也可以通过遗忘来计算：

算法 2.2 命题逻辑下基于遗忘的SNC计算

输入:

Γ, V, q：子句集合，原子命题的集合，出现在Γ且不出现在V中的原子命题

输出:

ϕ：　　V上的公式（ϕ是q在集合V和Γ上的最强必要条件）

1: $T_1 = \{C \mid C \in \Gamma$是$V$上的一个子句$\}$，$T_2 = \Gamma - T_1$
2: 将出现在T_2中的q用$\top$代替，并将得到的结果和T_1分别转换成为子句集合的极小集T_3和T_0.
3: 令$V' = Var(T_3) - V$;
4: **if** $V' = \emptyset$ **then**
5:　跳转到post-processing;
6: **end if**
7: 从V'中随机选择一个原子p，且令$V' = V' - \{p\}$;
8: 将$T_3[p/\top] \cup T_3[p/\bot]$转换为极小集得到结果$T_3$，跳转到4;
9: post-processing：根据下面步骤化简T_3：
10:　　移除T_3中被T_0包蕴的子句；
11:　　对T_3中的每个子句α，将$(T_3 - \{\alpha\}) \cup T_0$转换为极小子句集$T_\alpha$；
12:　　如果α被T_α中的某个子句包蕴，则将α从T_3中删除；
13: **return** T_3中子句的合取

定理 2.4. *给定S5公式Γ和原子命题集合$V \subseteq \mathscr{A}$，$q \in Var(\Gamma) - V$，则：*

(i) *q在V和Γ上的SNC为$KForget(\Gamma \wedge q, Var(\Gamma) - V)$；*

(ii) *q在V和Γ上的WSC为$\neg KForget(\Gamma \wedge \neg q, Var(\Gamma) - V)$。*

此时，由算法 2.1不难得出计算SNC和WSC的算法，这里就不再赘述。

2.6　本章小结

围绕本文的研究工作，本章首先介绍了最基本的真假赋值的概念，给出了命题逻辑公式的解释，随后给出了时序逻辑依赖的Kripke结构的定义。其次，本章详细介绍了带索引的CTL和μ-演算公式的语法和语义。CTL公式是带索引的CTL公式的子类，基于此又介绍了CTL公式的标准形式——$\mathrm{SNF}^g_{\mathrm{CTL}}$子句，并详细介绍了如何将CTL公式转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合。基于此，对CTL中的归结过程和归结系统做了详细的介绍，为下文使用基于归结的方法计算CTL中的遗忘做下铺垫。最后，本章详细介绍了本文密切相关的经典逻辑和模态逻辑下的遗忘的定义、基本公设及相关算法，并基于此给出这些逻辑系统下SNC（WSC）的定义和使用遗忘计算SNC（WSC）的详细算法。本章中所介绍的内容为后续章节提供了基本模型与定义，是开展后续研究工作的理论出发点。

第三章　遗忘理论

本章首先通过扩展互模拟的概念，给出*CTL*下遗忘理论的定义。其次，探索遗忘理论的一般通用属性，这些属性包括：模块化（*Modularity*）性质、交换性（*Commutativity*）、同质性（*Homogeneity*）等属性。

从一个公式中“遗忘”掉一些原子命题得到的结果应该不“违背”定义在其他原子命题集合上的公式，也就是说对于其他原子命题集合上的公式，原始公式能够逻辑蕴涵它当且仅当遗忘得到的结果能过逻辑蕴涵它。从模型的角度来讲，遗忘得到的结果的模型与原始公式的模型在“除去”被遗忘的那些原子命题之后是能够相互模拟的。互模拟描述的是两个在行为上能够相互替代的转换系统[20]。在本文中，转换系统被描述成为Kripke结构。因此为了描述遗忘理论，这部分给出在给定原子命题集合上Ind-结构之间互模拟的定义及其性质。

基于互模拟的概念，给出了CTL下遗忘的定义。与后面章节将要讲述的约束CTL下的遗忘相对应，这部分探索没有约束的遗忘理论的一般属性。

本章其余部分组织如下：首先，第3.1节定义一种V-互模拟的概念及其性质，并定义互模拟等价。其次，第3.2节定义CTL下的遗忘，并探索遗忘的基本性质和公设。最后，进行本章工作总结。

3.1　V-互模拟

这部分给出定义在给定原子命题集合V上的互模拟的概念，本文称之为V-互模拟。尽管在文章[44]中给出了相似的概念，但是如在基础知识部分所述，$S5$的语义是定义在一种特殊的Kripke结构（K-解释）下的，因而不具有一般性。这里探讨一种更加一般的V-互模拟。

定义 3.1 (V-互模拟). 给定原子命题集合$V \subseteq \mathscr{A}$、索引集合$I \subseteq Ind$和初始Ind-结构$\mathscr{M}_i = (S_i, R_i, L_i, [_]_i, s_0^i)$ $(i = 1,2)$。对关系$\mathscr{B}_V \subseteq S_1 \times S_2$和任意的$s_1 \in S_1$和$s_2 \in S_2$，若$(s_1, s_2) \in \mathscr{B}_V$蕴涵下列条件，则称$\mathscr{B}_V$是$\mathscr{M}_1$和$\mathscr{M}_2$之间的一个$V$-互模拟关系：

(i) $L_1(s_1) - V = L_2(s_2) - V$；

(ii) $\forall r_1 \in S_1$，若$(s_1, r_1) \in R_1$，则$\exists r_2 \in S_2$使得$(s_2, r_2) \in R_2$和$(r_1, r_2) \in \mathscr{B}_V$；

(iii) $\forall r_2 \in S_2$，若$(s_2, r_2) \in R_2$，则$\exists r_1 \in S_1$使得$(s_1, r_1) \in R_1$和$(r_1, r_2) \in \mathscr{B}_V$。

若$\mathscr{M}_1$和$\mathscr{M}_2$之间存在一个V-互模拟关系$\mathscr{B}_V$使得$(s_1,s_2)\in\mathscr{B}_V$，则称两个Ind-结构$\mathscr{K}_1=(\mathscr{M}_1,s_1)$和$\mathscr{K}_2=(\mathscr{M}_2,s_2)$是$V$-互模拟的（也称$\mathscr{K}_1$和$\mathscr{K}_2$关于$V$是互模拟的），记为$\mathscr{K}_1\leftrightarrow_V\mathscr{K}_2$。令$i\in\{1,2\}$，$\pi_i=(s_{i,1},s_{i,2},\ldots)$为$\mathscr{M}_i$上的路径，若对任意的$j\geq 1$都有$\mathscr{K}_{1,j}\leftrightarrow_V\mathscr{K}_{2,j}$，则称这两条路径是$V$-互模拟的，记为$\pi_1\leftrightarrow_V\pi_2$，其中$\mathscr{K}_{i,j}=(\mathscr{M}_i,s_{i,j})$。

直观地说，若两个状态在不考虑$V\subseteq\mathscr{A}$中的元素时，其行为是相同的，则称这两个状态在V上是“互模拟”的。当$V=\emptyset$，V-互模拟的三个条件即为CTL中的互模拟要满足的条件。下文中当初始Ind-Kripke结构能从上下文中清楚地知道时，简写$\mathscr{K}_1\leftrightarrow_V\mathscr{K}_2$为$s_1\leftrightarrow_V s_2$。

下面例子呈现结构之间的V-互模拟。

例 3.1. 令$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$为三个Ind-结构（其索引对应的后继函数与V-互模拟无关，所以在图里没给出），分别如图中的$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$所示。它们之间的互模拟关系也如图中标记所示，即$\mathscr{K}_1\leftrightarrow_{\{sp\}}\mathscr{K}_2$，$\mathscr{K}_2\leftrightarrow_{\{se\}}\mathscr{K}_3$和$\mathscr{K}_1\leftrightarrow_{\{sp,se\}}\mathscr{K}_3$。此外，可以看出$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$之间是互不互模拟[20]的，即不$\emptyset$-互模拟。

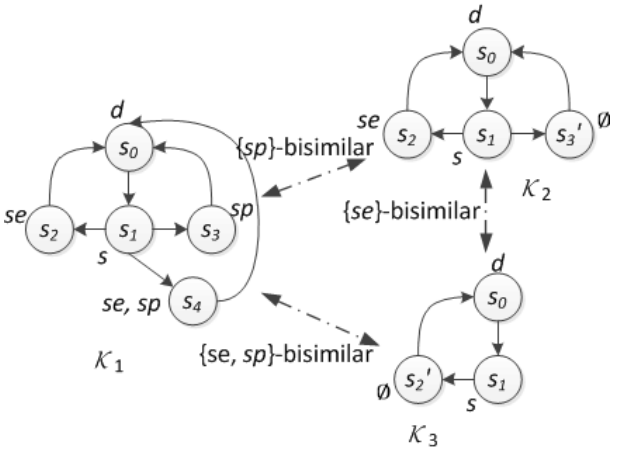


图 3.1: κ-结构之间的V-互模拟关系示意图

V-互模拟给出了两个结构之间相互模仿的行为关系，下面的命题给出了这种关系一些关键的性质。

命题 3.1. 给定集合$V_i\subseteq\mathscr{A}$、状态s_i'、路径π_i'和*Ind*-结构$\mathscr{K}_j=(\mathscr{M}_j,s_j)$，其中$i=1,2$，$j=1,2,3$。如果$\mathscr{K}_1\leftrightarrow_{V_1}\mathscr{K}_2$且$\mathscr{K}_2\leftrightarrow_{V_2}\mathscr{K}_3$，则：

(i) $\mathscr{K}_1\leftrightarrow_{V_1\cup V_2}\mathscr{K}_3$;

(ii) 若$V_1\subseteq V_2$，则$\mathscr{K}_1\leftrightarrow_{V_2}\mathscr{K}_2$;

(iii) $s_1'\leftrightarrow_{V_i}s_2'$ $(i=1,2)$ 蕴涵$s_1'\leftrightarrow_{V_1\cup V_2}s_2'$;

(iv) $\pi_1' \leftrightarrow_{V_i} \pi_2'$ $(i=1,2)$ 蕴涵$\pi_1' \leftrightarrow_{V_1\cup V_2} \pi_2'$；

(v) 对$\mathscr{M}_1$上的每条路径π_{s_1}，存在$\mathscr{M}_2$上的一条路径π_{s_2}使得$\pi_{s_1} \leftrightarrow_{V_1} \pi_{s_2}$，反之也成立。

证明. (i) 令$\mathscr{M}_j = (S_j, R_j, L_j, [_]_j, s_0^j)$ $(j=1,2,3)$，$\mathscr{B}$为s_1和s_2之间的V_1-互模拟关系，即$s_1 \leftrightarrow_{V_1} s_2$ 通过$\mathscr{B}$形成V_1-互模拟关系，$s_2 \leftrightarrow_{V_2} s_3$通过$\mathscr{B}''$形成$V_2$-互模拟关系。令$\mathscr{B}' = \{(w_1, w_3) \mid (w_1,w_2) \in \mathscr{B}$ 和$(w_2,w_3) \in \mathscr{B}''\}$。为了证明$\mathscr{K}_1 \leftrightarrow_{V_1\cup V_2} \mathscr{K}_3$，这里证明$\mathscr{B}'$ 是一个包含(s_1,s_3) 的$V_1 \cup V_2$-互模拟关系。由于$(s_1,s_2) \in \mathscr{B}$ 和$(s_2,s_3) \in \mathscr{B}''$，所以$(s_1,s_3) \in \mathscr{B}'$。对于所有$(w_1,w_3) \in \mathscr{B}'$：

(a) 存在一个$w_2 \in S_2$使得$(w_1,w_2) \in \mathscr{B}$ 且$(w_2,w_3) \in \mathscr{B}''$，因此由$w_1 \leftrightarrow_{V_1} w_2$可知，$L_1(w_1) - V_1 = L_2(w_2) - V_1$，且由$w_2 \leftrightarrow_{V_2} w_3$可知$L_2(w_2) - V_2 = L_3(w_3) - V_2$。所以有$L_1(w_1) - (V_1 \cup V_2) = L_3(w_3) - (V_1 \cup V_2)$.

(b) $\forall u_1 \in S_1$，若$(w_1,u_1) \in R_1$，则$\exists u_2 \in S_2$ 使得$(w_2,u_2) \in R_2$ 和$(u_1,u_2) \in \mathscr{B}$（由$\mathscr{B}'$的定义可知$(w_1,w_2) \in \mathscr{B}$ 且$(w_2,w_3) \in \mathscr{B}''$）；因而$\exists u_3 \in S_3$ 使得$(w_3,u_3) \in R_3$ 且$(u_2,u_3) \in \mathscr{B}''$，所以由$\mathscr{B}'$的定义可知$(u_1,u_3) \in \mathscr{B}'$。

(c) $\forall u_3 \in S_3$，若$(w_3,u_3) \in R_3$，则$\exists u_2 \in S_2$ 使得$(w_2,u_2) \in R_2$ 和$(u_2,u_3) \in \mathscr{B}_2$；因此$\exists u_1 \in S_1$ 使得$(w_1,u_1) \in R_1$ 且$(u_1,u_2) \in \mathscr{B}$，所以由$\mathscr{B}'$的定义可知$(u_1,u_3) \in \mathscr{B}'$。

(ii) 假定$\mathscr{B}_{V_1}$是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_1-互模拟关系，且$(s_1,s_2) \in \mathscr{B}_{V_1}$。这里证明$\mathscr{B}_{V_1}$也是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_2-互模拟关系。对任意的$(w_1,w_2) \in \mathscr{B}_{V_1}$，有：

- 因为$L_1(w_1) - V_1 = L_2(w_2) - V_1$ 和$V_1 \subseteq V_2$，所以$L_1(w_1) - V_2 = L_2(w_2) - V_2$；
- $\forall r_1 \in S_1$，若$(w_1,r_1) \in R_1$，因为$\mathscr{B}_{V_1}$是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_1-互模拟关系，则$\exists r_2 \in S_2$ 使得$(w_2,r_2) \in R_2$ 且$(r_1,r_2) \in \mathscr{B}_{V_1}$；和
- $\forall r_2 \in S_2$，若$(w_2,r_2) \in R_2$，因为$\mathscr{B}_{V_1}$是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_1-互模拟关系，则$\exists r_1 \in S_1$使得$(w_1,r_1) \in R_1$ 且$(r_1,r_2) \in \mathscr{B}_{V_1}$。

由于$V_i \subseteq (V_1 \cup V_2)$（$i=1,2$），则(iii)是(ii)的一种特殊情况，因而(iv)从(iii)可以容易得到。

(v) 可以从V-互模拟的定义容易得到，因为$s_1 \leftrightarrow_{V_1} s_2$ （即：$\mathscr{K}_1 \leftrightarrow_{V_1} \mathscr{K}_2$）。 □

在命题3.1中，性质(iii)-(v)是V-互模拟的标准属性，含义比较直观。性质(i)表示如果一个结构$\mathscr{K}_2$分别与另外的两个结构$\mathscr{K}_1$和$\mathscr{K}_3$具有V_1和V_2-互模拟关系，则这$\mathscr{K}_1$和$\mathscr{K}_3$是$V_1 \cup V_2$-互模拟的（如图 3.1所示）。如后文所示，这一性质对遗忘理论性质的探索至

关重要。性质(ii)表示若两个结构关于某一集合是互模拟的，则这两个结构关于该集合的超集是互模拟的。

从互模拟的定义上来看，如果两个结构是V-互模拟的，那么对于与V中的原子命题无关的公式φ来说，这两个结构同时满足或不满足φ。这一性质可以形式化地描述如下：

定理 3.1. 令$V \subseteq \mathscr{A}$是原子命题的集合，$\mathscr{K}_i\ (i=1,2)$是两个具有V-互模拟的*Ind*-结构，即：$\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$，$\Phi$是一个*CTL*公式且$IR(\Phi,V)$。则有$\mathscr{K}_1 \models \Phi$当且仅当$\mathscr{K}_2 \models \Phi$。

证明. 这一结论可以从CTL公式的结构归纳地来证明。此外，不失一般性地可以假设$Var(\Phi)\cap V=\emptyset$，$\mathscr{K}_1=(\mathscr{M},s)$和$\mathscr{K}_1=(\mathscr{M}',s')$。

情形1：$\Phi = p$ （$p \in \mathscr{A}-V$）.

$(\mathscr{M},s)\models\Phi$当且仅当$p\in L(s)$ （可满足关系的定义）

$\Leftrightarrow p\in L'(s')$ （$s\leftrightarrow_v s'$）

$\Leftrightarrow (\mathscr{M}',s')\models\Phi$。

情形2：$\Phi=\neg\psi$.

$(\mathscr{M},s)\models\Phi$当且仅当$(\mathscr{M},s)\not\models\psi$

$\Leftrightarrow (\mathscr{M}',s')\not\models\psi$ （归纳假设）

$\Leftrightarrow (\mathscr{M}',s')\models\Phi$。

情形3：$\Phi=\psi_1\vee\psi_2$.

$(\mathscr{M},s)\models\Phi$

$\Leftrightarrow (\mathscr{M},s)\models\psi_1$或$(\mathscr{M},s)\models\psi_2$

$\Leftrightarrow (\mathscr{M}',s')\models\psi_1$或$(\mathscr{M}',s')\models\psi_2$ （归纳假设）

$\Leftrightarrow (\mathscr{M}',s')\models\Phi$。

情形4：$\Phi=\mathrm{EX}\psi$.

$(\mathscr{M},s)\models\Phi$

$\Leftrightarrow$ 存在一条路径$\pi=(s,s_1,\ldots)$使得$(\mathscr{M},s_1)\models\psi$

$\Leftrightarrow$ 存在一条路径$\pi'=(s',s'_1,\ldots)$使得$\pi\leftrightarrow_V\pi'$ ($s\leftrightarrow_V s'$，命题 3.1)

$\Leftrightarrow s_1\leftrightarrow_V s'_1$ ($\pi\leftrightarrow_V\pi'$)

$\Leftrightarrow (\mathscr{M}',s'_1)\models\psi$ （归纳假设）

$\Leftrightarrow (\mathscr{M}',s')\models\Phi$。

情形5：$\Phi=\mathrm{EG}\psi$.

$(\mathscr{M},s)\models\Phi$

$\Leftrightarrow$ 存在一条路径$\pi=(s=s_0,s_1,\ldots)$使得对于任意的$i\geq 0$都有$(\mathscr{M},s_i)\models\psi$

$\Leftrightarrow$ 存在一条路径$\pi'=(s'=s'_0,s'_1,\ldots)$使得$\pi\leftrightarrow_V\pi'$ ($s\leftrightarrow_V s'$，命题 3.1)

$\Leftrightarrow$ 对于任意的$i \geq 0$都有$s_i \leftrightarrow_V s'_i$ $(\pi \leftrightarrow_V \pi')$
$\Leftrightarrow$ 对于任意的$i \geq 0$都有$(\mathscr{M}, s'_i) \models \psi$ （归纳假设）
$\Leftrightarrow (\mathscr{M}', s') \models \Phi$。

情形6：$\Phi = \mathrm{E}(\psi_1 \mathrm{U} \psi_2)$.

$(\mathscr{M}, s) \models \Phi$
$\Leftrightarrow$ 存在一条路径$\pi = (s = s_0, s_1, \ldots)$和$i \geq 0$使得$(\mathscr{M}, s_i) \models \psi_2$，且对所有的$0 \leq j < i$都有$(\mathscr{M}, s_j) \models \psi_1$
$\Leftrightarrow$ 存在一条路径$\pi' = (s' = s'_0, s'_1, \ldots)$使得$\pi \leftrightarrow_V \pi'$ $(s \leftrightarrow_V s'$，命题 3.1)
$\Leftrightarrow (\mathscr{M}', s'_i) \models \psi_2$，且对于所有的$0 \leq j < i$都有$(\mathscr{M}', s'_j) \models \psi_1$ （归纳假设）
$\Leftrightarrow (\mathscr{M}', s') \models \Phi$。 □

值得注意的是，上述定理中的公式Φ不包含索引。否则，Ind-结构中的索引函数可能会影响公式的可满足性。例：令$\phi = \mathrm{E}_{\langle 1 \rangle} \mathrm{X} p$、$\mathscr{K} = (\mathscr{M}, s)$ 和$\mathscr{M} = (S, R, L, [_], s_0)$，其中$S = \{s_0, s_1\}$, $L(s_0) = \emptyset$、$L(s_1) = \{p\}$、$R = \{(s_0, s_1), (s_0, s_0), (s_1, s_1), (s_1, s_0)\}$ 和$[1] = \{(s_0, s_1), (s_1, s_1)\}$。容易检查$\mathscr{K} \models \phi$。令$\mathscr{K}' = (\mathscr{M}', s)$ 和$\mathscr{M}' = (S, R, L, [_]', s_0)$，其中$[1]' = \{(s_0, s_0), (s_1, s_1)\}$。显然$\mathscr{K} \leftrightarrow_{\{q\}} \mathscr{K}'$，$\mathrm{IR}(\phi, \{q\})$。但是，$\mathscr{K}' \not\models \phi$。

例 3.2. 令$\varphi_1 = d \wedge \mathrm{EF} se \wedge \mathrm{AG}(se \rightarrow \mathrm{AX} d)$和$\varphi_2 = d \wedge \mathrm{AX} se$是两个CTL公式，且$\mathrm{IR}(\varphi_1, \{sp\})$和$\mathrm{IR}(\varphi_2, \{sp\})$成立。因此可以验证图3.1中的$\mathscr{K}_1$和$\mathscr{K}_2$都满足$\varphi_1$，但是都不满足$\varphi_2$。

定义 3.2 (互模拟等价，bisimilar equivalence)**.** 给定原子命题的集合$V \subseteq \mathscr{A}$，公式φ和ψ。若对任意的$\mathscr{K} \models \varphi$都存在一个$\mathscr{K}' \models \psi$使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，且对任意的$\mathscr{K}' \models \psi$都存在一个$\mathscr{K} \models \varphi$使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，则称公式$\varphi$和$\psi$是$V$-互模拟等价的(bisimilar equivalence)，记为$\varphi \equiv_V \psi$。

由定义3.1和3.2，和命题3.1可容易得出下列引理。

引理 3.1. 对任意的$V \subseteq \mathscr{A}$，$\leftrightarrow_V$和$\equiv_V$为等价关系。

证明. 由命题3.1(i)可知$\leftrightarrow_V$是传递的。显然也是自反和对称的。因此是等价关系。

关系$\equiv_V$显然是自反和对称的。假设$\varphi \equiv_V \psi$和$\psi \equiv_V \xi$。则有对任意的$\mathscr{K} \models \varphi$，由$\varphi \equiv_V \psi$可知存在一个$\mathscr{K}' \models \psi$ 使得 $\mathscr{K}' \leftrightarrow_V \mathscr{K}$，且由$\psi \equiv_V \xi$可知存在一个$\mathscr{K}'' \models \xi$ 使得 $\mathscr{K}' \leftrightarrow_V \mathscr{K}''$。又因为$\leftrightarrow_V$是一个等价关系，因此有$\mathscr{K} \leftrightarrow_V \mathscr{K}''$。类似地，对任意的$\mathscr{K}'' \models \xi$，存在$\mathscr{K} \models \varphi$使得 $\mathscr{K}'' \leftrightarrow_V \mathscr{K}$。这蕴含$\equiv_V$是传递的。因此$\equiv_V$是等价关系。
□

此外，上面的定义和命题3.1蕴涵下面的推论。

推论 3.1. 令V、V_1、V_2为$\mathscr{A}$的子集，φ和ψ为公式。

(i) 若$\varphi \equiv \psi$，则$\varphi \equiv_V \psi$。

(ii) 若φ和ψ不包括索引，且$\varphi \equiv_\emptyset \psi$，则$\varphi \equiv \psi$。

(iii) 若$\varphi \equiv_{V_i} \psi\ (i=1,2)$，则$\varphi \equiv_{V_1 \cup V_2} \psi$。

(iv) 若$\varphi \equiv_{V_1} \psi$和$V_1 \subseteq V_2$，则$\varphi \equiv_{V_2} \psi$。

证明. (i) 对任意φ（或ψ）的模型$\mathscr{K}$和$V \subseteq \mathscr{A}$，存在一个$\mathscr{K} \leftrightarrow_V \mathscr{K}$。因此，$\varphi \equiv_V \psi$。

(ii) 对任意φ的模型$\mathscr{K}$，存在ψ的一个模型$\mathscr{K}'$使得$\mathscr{K} \leftrightarrow_\emptyset \mathscr{K}'$。显然IR$(\psi, \emptyset)$，因此由定理 3.1可知$\mathscr{K} \models \psi$。类似地，对任意的$\mathscr{K}' \models \psi$，存在$\mathscr{K} \models \varphi$使得$\mathscr{K} \leftrightarrow_\emptyset \mathscr{K}'$，因此$\mathscr{K}' \models \varphi$。

(iii) 对任意的$\mathscr{K} \models \varphi$，存在$\mathscr{K}' \models \psi$使得$\mathscr{K} \leftrightarrow_{V_i} \mathscr{K}'\ (i=1,2)$。因此，由命题3.1(i)可知$\mathscr{K} \leftrightarrow_{V_1 \cup V_2} \mathscr{K}'$。类似地，对任意的$\mathscr{K}' \models \psi$，存在$\mathscr{K} \models \varphi$使得$\mathscr{K} \leftrightarrow_{V_1 \cup V_2} \mathscr{K}'$。因此，$\varphi \equiv_{V_1 \cup V_2} \psi$。

(iv) 可类似于(iii)证明。 □

请注意，在上述结论(ii)中“φ和ψ中不包含索引”是必要的。否则，令$\varphi = \mathrm{E}_{\langle 1 \rangle}\mathrm{X}p$和$\psi = \mathrm{E}_{\langle 2 \rangle}\mathrm{X}p$，可以证明$\varphi \equiv_\emptyset \psi$，但是$\varphi \not\equiv \psi$。

命题 3.2. 令φ为一个*CTL*公式。则$\varphi \equiv_U T_\varphi$，其中$T_\varphi = \mathrm{SNF}^g_{\mathrm{CTL}}(\varphi)$和$U = Var(T_\varphi) - Var(\varphi)$。

证明. 为了讨论方便，转换过程产生了一个公式集合的序列，$T_0, T_1, \ldots, T_n = T_\varphi$，其中$p$是不出现在$\varphi$中的原子命题，$T_0 = \{\mathrm{AG}(\mathbf{start} \to p), \mathrm{AG}(p \to \mathbf{simp}(\mathbf{nnf}(\varphi)))\}$且对任意的$i$（$0 \le i < n$）有$T_{i+1} = (T_i - \{\psi\}) \cup R_i$（$Trans(\psi)$返回的结果为$R_i$）。此外，在这一过程中，所有的公式都是其否定范式的形式。

为了证明命题中的结论，只需证明，对任意的i（$0 \le i < n$）有$T_i \equiv_{V'} T_{i+1}$成立。由于T_{i+1}是由T_i通过表2.1中的规则作用于T_i中的某一个公式得到，因此证明过程分为两个部分：（1）从φ到T_0部分；（2）对表2.1中的规则做归纳的部分。为了方便，下面假设$\mathscr{M}_1 = (S_1, R_1, L_1, [_], s_1)$和$\mathscr{M}_2 = (S_2, R_2, L_2, [_]_2, s_2)$。

（1）这里将证明$\varphi \equiv_{\{p\}} T_0$。

($\Rightarrow$) $\forall(\mathscr{M}_1, s_1) \in Mod(\varphi)$，可以构造一个Ind-Kripke结构$\mathscr{M}_2 = (S_2, R_2, L_2, [_], s_2)$使得$\mathscr{M}_2$除了$L_2(s_2) = L_1(s_1) \cup \{p\}$（默认不出现在$\varphi$中的原子命题都不出现在状态的标签中），其他的元素都与$\mathscr{M}_1$中元素相同。显然，$(\mathscr{M}_2, s_2) \models T_0$且$(\mathscr{M}_1, s_1) \leftrightarrow_{\{p\}} (\mathscr{M}_2, s_2)$。

($\Leftarrow$) $\forall(\mathscr{M}_1, s_1) \in Mod(T_0)$，由**start**的语义可以知道$(\mathscr{M}_1, s_1) \models \varphi$。

（2）这里将证明对任意的i（$0 \le i < n$）有$T_i \equiv_{V'} T_{i+1}$成立，其中$T_{i+1} = (T_i - \{\psi\}) \cup R_i$。为了方便，用$\psi \to_t R_i$表示$R_i$是使用规则$t$在公式$\psi$上得到的结果，且$T_i = X \cup \{\psi\}$

（显然，$T_{i+1}=X\cup R_i$）。下面证明规则$t\in\{$**Trans(1), Trans(4), Trans(6)**$\}$的情形，其他情形可以类似地证明。

（a）$t=$**Trans(1)**。

$(\Rightarrow)\ \forall(\mathscr{M}_1,s_1)\in Mod(T_i)$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{EX}\varphi)$
$\Rightarrow (\mathscr{M}_1,s_1)\models X$，且对任意路径$\pi$上的状态$s_{1,j}$ $(j\geq 1)$有：$(\mathscr{M},s_{1,j})\not\models\neg q$或存在一个状态$s_{1,j+1}$使得$(s_{1,j},s_{1,j+1})\in R_1$且$(\mathscr{M},s_{1,j+1})\models\varphi$。

由此构造一个初始Ind-Kripke结构$\mathscr{M}_2$使得$\mathscr{M}_2$与$\mathscr{M}_1$相同，除了对使用规则**Trans(1)**在公式$\mathrm{AG}(q\rightarrow \mathrm{EX}\varphi)$上而引入的新索引$ind$有$[ind]_2=\bigcup_{s\in S}R_s\cup R_y$。其中：

- $R_{s_{1,j}}=\{(s_{1,j},s_{1,j+1}),(s_{1,j+1},s_{1,j+2}),\dots\}$ $(j\geq 1)$，其满足“若$(\mathscr{M}_1,s_{1,j})\models q$，则$(\mathscr{M}_1,s_{1,j+1})\models\varphi$”且“对于任意的$i\geq j$，若$(s_{1,i},s')\in R_s$ $(s\neq s_{1,j})$，则$s'=s_{1,i+1}$”；
- $R_y=\{(s_x,s_y)\mid s_x\in S$，若$\forall(s'_1,s'_2)\in\bigcup_{s\in S}R_s$，$s'_1\neq s_x$，则找一个状态$s_y\in S_2$使得$(s_x,s_y)\in R_2\}$。

显然，$(\mathscr{M}_1,s_1)\leftrightarrow_{\langle\emptyset,\{ind\}\rangle}(\mathscr{M}_2,s_2)$
$\Rightarrow$ 对任意从s_2开始的路径$\pi=(s_2=s_{2,1},s_{2,2},\dots)$，如果$s_{2,j}\in\pi$，则$(\mathscr{M}_2,s_{2,j})\models\neg q$或者$(\mathscr{M}_2,s_{2,j})\models \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi$
$\Rightarrow (\mathscr{M}_2,s_2)\models \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi)$
$\Rightarrow (\mathscr{M}_2,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi)$

$(\Leftarrow)\ \forall(\mathscr{M}_1,s_1)\in Mod(T_{i+1})$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi)$
$\Rightarrow (\mathscr{M}_1,s_1)\models X$且$(\mathscr{M}_1,s_1)\models \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi)$
$\Rightarrow$ 对任意的以s_1为始点的路径上的任意状态$s_{1,j}$，$(\mathscr{M}_1,s_{1,j})\models\neg q$或$(\mathscr{M}_1,s_{1,j})\models \mathrm{EX}\varphi$
$\Rightarrow (\mathscr{M}_1,s_1)\models \mathrm{AG}(q\rightarrow \mathrm{EX}\varphi)$
$\Rightarrow (\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{EX}\varphi)$。

（b）$t=$**Trans(4)**。

$(\Rightarrow)\ \forall(\mathscr{M}_1,s_1)\in Mod(T_i)$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \varphi_1\vee\varphi_2)$
$\Rightarrow (\mathscr{M}_1,s_1)\models X$且$\forall s'_1\in S_1$，$(\mathscr{M}_1,s'_1)\models q\rightarrow\varphi_1\vee\varphi_2$
$\Rightarrow (\mathscr{M}_1,s'_1)\models\neg q$或$(\mathscr{M}_1,s'_1)\models\varphi_1\vee\varphi_2$。

如下构造初始Ind-Kripke结构$\mathscr{M}_2=(S_2,R_2,L_2,[_]_2,s_2)$：

- $S_2=S_1$，$R_2=R_1$，$[_]_2$与$[_]_1$一样且$s_2=s_1$；
- L_2与L_1类似，除了：对任意的$s'_1\in S_2$，若$(\mathscr{M}_1,s'_1)\models\neg q$，则$L_2(s'_1)=L_1(s'_1)$，否则“若$(\mathscr{M}_1,s'_1)\models\varphi_1$，则$L_2(s'_1)=L_1(s'_1)$，否则$L_2(s'_1)=L_1(s'_1)\cup\{p\}$”。

显然，$(\mathscr{M}_2,s'_1)\models(q\rightarrow\varphi_1\vee p)\wedge(p\rightarrow\varphi_2)$且$(\mathscr{M}_1,s_1)\leftrightarrow_{\{p\}}(\mathscr{M}_2,s_2)$，因而$(\mathscr{M}_2,s_1)\models T_{i+1}$。

($\Leftarrow$) $\forall(\mathscr{M}_1,s_1)\in Mod(T_{i+1})$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \varphi_1\vee p)\wedge \mathrm{AG}(p\rightarrow\varphi_2)$。显然，$(\mathscr{M}_1,s_1)\models T_i$。

（c）t=**Trans(6)**。

这里证明$\mathrm{E}_{\langle ind\rangle}\mathrm{X}$的情形，AX的情形可以类似地证明。

($\Rightarrow$) $\forall(\mathscr{M}_1,s_1)\in Mod(T_i)$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi)$
$\Rightarrow (\mathscr{M}_1,s_1)\models X$且对任意的$s_1'\in S,(\mathscr{M}_1,s_1')\models q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi$
$\Rightarrow (\mathscr{M}_1,s_1')\models \neg q$或者存在一个状态$s'$使得$(s_1',s')\in[ind]$且$(\mathscr{M}_1,s')\models\varphi$

如下构造初始Ind-Kripke结构$\mathscr{M}_2=(S_2,R_2,L_2,[_]_2,s_2)$：

- $S_2=S_1$，$R_2=R_1$，$[_]_2$与$[_]_1$一样且$s_2=s_1$；
- L_2与L_1类似，除了：对任意的$s_1'\in S_2$，若$(\mathscr{M}_1,s_1')\models\neg q$，则$L_2(s_1')=L_1(s_1')$，否则“若$(\mathscr{M}_1,s_1')\models q$，则$L_2(s')=L_1(s')\cup\{p\}$ $((s_1',s')\in R_2)$”。

显然，$(\mathscr{M}_2,s_2)\models \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}p)\wedge\mathrm{AG}(p\rightarrow\varphi)$, $(\mathscr{M}_2,s_2)\models T_{i+1}$且$(\mathscr{M}_1,s_1)\leftrightarrow_{\{p\}}(\mathscr{M}_2,s_2)$ $(s_2=s_1)$。

($\Leftarrow$) $\forall(\mathscr{M}_1,s_1)\in Mod(T_{i+1})$，即$(\mathscr{M}_1,s_1)\models X\wedge \mathrm{AG}(q\rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}p)\wedge \mathrm{AG}(p\rightarrow\varphi)$。显然，$(\mathscr{M}_1,s_1)\models T_i$。 □

例 3.3. 令$\varphi=\mathrm{A}((p\wedge q)\mathrm{U}(f\vee m))\wedge r$。$T_\varphi$是下面公式构成的集合：

$$1:\mathbf{start}\rightarrow z,\quad 2:\top\rightarrow\neg z\vee r,\quad 3:\top\rightarrow\neg x\vee f\vee m,\quad 4:\top\rightarrow\neg z\vee x\vee y,$$
$$5:\top\rightarrow\neg y\vee p,\quad 6:\top\rightarrow\neg y\vee q,\quad 7:z\rightarrow \mathrm{AF}x,\quad 8:y\rightarrow\mathrm{AX}(x\vee y)$$

其中x,y,z为新引入的原子命题。

3.2 CTL遗忘及其性质

这部分将给出CTL下遗忘的定义及其相关属性。

定义 3.3 (遗忘，forgetting). 令V是$\mathscr{A}$的一个子集，Φ是一个公式。如果一个公式ψ满足下面条件，则称ψ为从Φ中遗忘掉V后得到的结果：

- ψ与V中的原子命题无关（即：$IR(\psi,V)$）；
- $Mod(\psi)=\{\mathscr{K}\mid \mathscr{K}$是一个初始Ind-结构$,\exists\mathscr{K}'\in Mod(\phi)$ 使得 $\mathscr{K}'\leftrightarrow_V\mathscr{K}\}$

从定义3.3可以看出，如果有两个公式ψ和ψ'都是从Φ中遗忘掉V中元素后得到的结果，则有$Mod(\psi) \equiv Mod(\psi')$。从这个角度来看，可以说从$\Phi$中遗忘掉$V$中元素后得到的结果之间是语义等价的。将遗忘的结果记为$\mathrm{F}_{\mathrm{CTL}}(\phi, V)$，不做其他说明的情况下，这表示从$\phi$中遗忘掉$V$是CTL可表示的。此外，当$V$中只包含一个元素的时候，可以省略掉集合符号，即：$\mathrm{F}_{\mathrm{CTL}}(\Phi, \{p\}) \equiv \mathrm{F}_{\mathrm{CTL}}(\Phi, p)$。

在上述遗忘的定义中说明了如果公式ψ的任意一个模型$\mathscr{K}$都能找到φ的一个模型$\mathscr{K}'$使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，则称$\psi$为从$\varphi$中遗忘掉$V$中原子命题后得到的结果。为刻画S5逻辑下该概念的直观含义，Zhang等人提出了如下遗忘理论特性——这些特性被称为遗忘理论公设（forgetting postulates）[44]。类似地，给定CTL公式φ、$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$和原子命题集合$V \subseteq \mathscr{A}$和$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$，CTL下遗忘理论公设如下：

(W) 削弱：$\varphi \models \varphi'$；

(PP) 正支持：对任意与V无关的公式η，若$\varphi \models \eta$则$\varphi' \models \eta$；

(NP) 负支持：对任意与V无关的公式η，若$\varphi \not\models \eta$则$\varphi' \not\models \eta$；

(IR) 无关性：$\mathrm{IR}(\varphi', V)$。

直观地说，**(W)**和**(IR)**表明“遗忘”削弱了公式φ且得到的结果与V无关，**(PP)**和**(NP)**表明对任意与V无关的公式η，$\varphi \models \eta$当且仅当$\varphi' \models \eta$。总而言之，遗忘得到的结果能蕴涵所有与V无关且能被φ蕴涵的结果，且不能蕴涵所有与V无关且不能被φ蕴涵的结果。此外，这些公设不都是独立的，如：NP由**W**和**PP**可以得出，但是这里把它们都列出来以表达更加直观的含义。从数据库和安全的层面讲，遗忘相当于从已有的关系表中构建出一个视图，达到了隐私保护的作用。下面的定理表上述公设对CTL的遗忘是充分且必要的。

定理 3.2 (表达性定理，Representation Theorem)**.** *给定CTL公式φ和φ'，$V \subseteq \mathscr{A}$为原子命题的集合。下面的陈述是等价的：*

(i) $\varphi' \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$,

(ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi$*和*$IR(\phi, V)\}$,

*(iii) 若φ、φ'和V为(i)和(ii)中提到的符号，则公设(**W**)、(**PP**)、(**NP**)和(**IR**)成立.*

证明. $(i) \Leftrightarrow (ii)$. 为了证明这个结论，只需证明如下等式成立：

$$Mod(\mathrm{F}_{\mathrm{CTL}}(\varphi, V)) = Mod(\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi, V)\}).$$

($\Rightarrow$) 对$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的任意模型$\mathscr{K}'$
$\Rightarrow \exists\mathscr{K}$使得$\mathscr{K}\models\varphi$且$\mathscr{K}\leftrightarrow_V\mathscr{K}'$ (定义 3.3)
$\Rightarrow \forall\phi$，若$\varphi\models\phi$且$\mathrm{IR}(\phi,V)$，则$\mathscr{K}'\models\phi$ (定理 3.1)
$\Rightarrow \mathscr{K}'\models\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}$

($\Leftarrow$) 因为$\mathrm{IR}(\mathrm{F}_{\mathrm{CTL}}(\varphi,V),V)$且$\varphi\models\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$，由定义 3.3可知$\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}\models\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。

$(ii)\Rightarrow(iii)$.令$A=\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}$。首先，由于对任意的$\phi'\in A$都有$\varphi\models\phi'$，所以$\varphi\models\varphi'$。其次，对任意的公式$\phi$，若$\mathrm{IR}(\phi,V)$且$\varphi\models\phi$则$\phi\in A$，因此$\varphi'\models\phi$. 第三，对任意的公式$\phi$，若$\mathrm{IR}(\phi,V)$且$\varphi\not\models\phi$则$\phi\notin A$。因此$\varphi'\not\models\phi$。最后，因为对任意的$\phi'\in A$都有$\mathrm{IR}(\varphi',V)$，所以$\mathrm{IR}(\phi',V)$。

$(iii)\Rightarrow(ii)$.一方面，由(**PP**) 和(**NP**)可知，对任意的公式ϕ'且$\mathrm{IR}(\phi',V)$，$\varphi\models\phi'$当且仅当$\varphi'\models\phi'$。所以对任意的$\phi'\in\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}$都有$\varphi'\models\phi'$，因而$\varphi'\models\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}$。另一方面，由(**W**) 和(**IR**)可知$\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}\models\varphi'$。因此，$\varphi'\equiv\{\phi\mid\varphi\models\phi,\mathrm{IR}(\phi,V)\}$。 □

尽管上面的表达性定理描述了遗忘及其基本准则之间“当且仅当”的关系，值得注意的是上面的定义 3.3并不表示遗忘的结果一定存在。事实上，存在一个CTL公式和原子命题的集合，使得从该公式里遗忘掉集合里的元素的结果不可用CTL公式表示。例：令p 和x为两个不同的原子命题，$\varphi(p,x)$[1]为下面公式合取[79]：

$$\mathrm{AG}(\neg x\wedge\neg\mathrm{AG}p\rightarrow\neg\mathrm{AX}\neg x),\qquad \mathrm{AG}(\neg\mathrm{AX}\neg x\rightarrow\mathrm{AX}x),$$
$$\mathrm{AG}(\mathrm{AX}x\rightarrow\neg x\wedge\neg\mathrm{AG}p),\qquad \mathrm{AG}(x\rightarrow\neg\mathrm{AG}p),\qquad \mathrm{AG}(\mathrm{AFAG}p).$$

Maksimova证明$\varphi(p,x)\wedge\varphi(p,y)\models x\leftrightarrow y$且不存在CTL公式$\psi$使得$Var(\psi)=\{p\}$且$\varphi(p,x)\models x\leftrightarrow\psi$，即CTL不具有Beth属性。这一结论蕴涵如下命题：

命题 3.3. $\mathrm{F}_{\mathrm{CTL}}(x\wedge\varphi(p,x),\{x\})$ *在CTL中是不可表示的。*

证明. 令$\psi(p)=\mathrm{F}_{\mathrm{CTL}}(x\wedge\varphi(p,x),\{x\})$为一个CTL公式。有
$\varphi(p,x)\wedge\varphi(p,y)\vdash x\leftrightarrow y$
$\Rightarrow\varphi(p,x)\wedge x\vdash\varphi(p,y)\rightarrow y$
$\Rightarrow\varphi(p,x)\wedge x\vdash\psi(p)$ 和$\psi(p)\vdash\varphi(p,y)\rightarrow p(y)$ (定理 3.2)
$\Rightarrow\varphi(p,x)\vdash x\rightarrow\psi(p)$，和$\varphi(p,y)\vdash\psi(p)\rightarrow p(y)$，这意味着$\varphi(p,x)\vdash\psi(p)\rightarrow p(x)$
$\Rightarrow\varphi(p,x)\vdash x\leftrightarrow\psi(p)$，这是一个矛盾。 □

[1] $\varphi(p,x)$ 表示具有原子命题集合$Var(\varphi)=\{p,x\}$的公式。

本段和下一段在related work的国内外研究现状中。事实上，遗忘结果的存在性（可表达性）与均匀插值（或Craig插值）性质密切相关。从形式上说，如果一个逻辑系统$\mathscr{L}$中任意的公式φ和ψ，若$\varphi \models_{\mathscr{L}} \psi$，则存在一个公式$\xi$使得$\varphi \vdash_{\mathscr{L}} \xi$、$\xi \vdash_{\mathscr{L}} \psi$和$Var(\xi) \subseteq Var(\varphi) \cap Var(\psi)$，则$\mathscr{L}$具有或Craig插值性质，若$\xi$与$\psi$无关，则$\mathscr{L}$具有均匀插值性质。研究表明，LTL、CTL和CTL*不具有均匀插值性质[79,112]。

从命题公式φ中遗忘掉原子命题p为：$Forget(\varphi,\{p\}) \equiv \varphi[p/\bot] \vee \varphi[p/\top]$。值得注意的是，本文遗忘的定义与Lin等人于1994提出命题逻辑下的遗忘一致。换句话说，本文将命题逻辑下的遗忘扩展到了CTL下。下面命题展示了上述结论：

定理 3.3. *给定一个命题公式φ和原子命题的集合$V \subseteq \mathscr{A}$，则下面逻辑等式成立。*

$$\mathrm{F}_{\mathrm{CTL}}(\varphi,V) \equiv Forget(\varphi,V).$$

证明. 为了证明上述结论成立，只需要证明$Mod(\mathrm{F}_{\mathrm{CTL}}(\varphi,V)) = Mod(Forget(\varphi,V))$。

一方面，对于$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的任意一个模型$(\mathscr{M},s)$，由遗忘的定义可知存在一个$\varphi$的模型$(\mathscr{M}',s')$使得$(\mathscr{M},s) \leftrightarrow_V (\mathscr{M}',s')$。因而有$L(s)-V = L'(s')-V$（其中$L \in \mathscr{M}$，$L' \in \mathscr{M}'$），这意味着$(\mathscr{M},s) \models Forget(\varphi,V)$。

另一方面，对于$Forget(\varphi,V)$的任意一个模型$(\mathscr{M},s)$（$\mathscr{M} = (S,R,L,[_],s)$），存在一个$\varphi$的模型$(\mathscr{M}',s')$（$\mathscr{M}' = (S',R',L',[_]',s')$）使得$L(s)-V = L'(s')-V$。此时可以构建一个初始Ind-结构$(\mathscr{M}_1,s_1)$使得$\mathscr{M}_1 = (S_1,R_1,L_1,[_],s_1)$，其中：

- $S_1 = (S-\{s\}) \cup \{s_1\}$；
- R_1由将R出现的s替换为s_1得到；
- 对于S_1中的任意的一个状态s^*：

$$L_1(s^*) = \begin{cases} L'(s'), & \text{如果} s^* = s_1; \\ L(s^*), & \text{否则。} \end{cases}$$

显然，$(\mathscr{M}_1,s_1)$是φ的一个模型且$s_1 \leftrightarrow_V s$。因此，$(\mathscr{M},s)$是$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的一个模型。□

遗忘的另一个重要属性与V-无关性密切相关。直观地说，对于给定的公式$\psi = \varphi \wedge (q \leftrightarrow \alpha)$，如果IR$(\varphi \wedge \alpha,\{q\})$，那么从$\psi$中遗忘掉$q$后得到的结果为$\varphi$。这一性质与后文中将要介绍的$SNC$（$WSC$）的计算密切相关。但是由于其也是遗忘的性质，因而本文将其放在此处来探讨。

引理 3.2. *给定两个公式φ和α，且$q \notin (Var(\varphi) \cup Var(\alpha))$。则$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge (q \leftrightarrow \alpha),q) \equiv \varphi$。*

证明. 令$\varphi' = \varphi \wedge (q \leftrightarrow \alpha)$。对于任意$\mathrm{F}_{\mathrm{CTL}}(\varphi', q)$的模型$(\mathscr{M}, s)$，由遗忘的定义可知存在一个初始Ind-结构$(\mathscr{M}', s')$使得$(\mathscr{M}, s) \leftrightarrow_{\{q\}} (\mathscr{M}', s')$且$(\mathscr{M}', s') \models \varphi'$。$(\mathscr{M}', s') \models \varphi$显然成立。此外，由于$\mathrm{IR}(\varphi, \{q\})$且$(\mathscr{M}, s) \leftrightarrow_{\{q\}} (\mathscr{M}', s')$，由定理 3.1可知$(\mathscr{M}, s) \models \varphi$。

为了证明另一个方向，令$\mathscr{M} = (S, R, L, [_], s)$且$(\mathscr{M}, s) \in Mod(\varphi)$。下面构造初始Ind-结构$(\mathscr{M}', s)$使得$\mathscr{M}' = (S, R, L', [_], s)$，其中：

$$
\begin{aligned}
&L' : S \to \mathscr{A} \text{ 且 } \forall s^* \in S, \text{ 若 } (\mathscr{M}, s^*) \not\models \alpha, \text{ 则} L'(s^*) = L(s^*) - \{q\} \text{ 否则 } L'(s^*) = L(s^*) \cup \{q\}, \\
&\text{若 } (\mathscr{M}, s) \models \alpha, \text{ 则} L'(s) = L(s) \cup \{q\}, \text{ 否则 } L'(s) = L(s) - \{q\}.
\end{aligned}
$$

可以看出$(\mathscr{M}', s) \models \varphi$、$(\mathscr{M}', s) \models q \leftrightarrow \alpha$且$(\mathscr{M}', s) \leftrightarrow_{\{q\}} (\mathscr{M}, s)$。因此，$(\mathscr{M}', s) \models \varphi \wedge (q \leftrightarrow \alpha)$。所以，由$(\mathscr{M}', s) \leftrightarrow_{\{q\}} (\mathscr{M}, s)$可知$(\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q)$。 □

除了上述性质，遗忘还有其他一些一般属性。下面将详细介绍这些属性。

根据遗忘的定义可以看出，从一个公式里遗忘掉某个原子命题集合中的元素是将该集合看作一个整体来遗忘的。下面的结论说明，遗忘可以将原子命题中的元素拿出来一个一个的遗忘，而不是作为一个整体。

命题 3.4 (模块性，Modularity). *对于给定的公式φ，原子命题集合V，和原子命题p（$p \notin V$），下面的结论成立：*

$$\mathrm{F}_{\mathrm{CTL}}(\varphi, \{p\} \cup V) \equiv \mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\varphi, p), V).$$

证明. 要证明上述结论成立，只需证明等式左右两边的公式有相同的模型。

一方面，令$\mathscr{M}_1 = (S_1, R_1, L_1, [_], s_1)$是一个初始Ind-Kripke结构，$(\mathscr{M}_1, s_1)$是$\mathrm{F}_{\mathrm{CTL}}(\varphi, \{p\} \cup V)$的一个模型。由遗忘的定义可知，存在$\varphi$的一个模型$(\mathscr{M}, s)$（$\mathscr{M} = (S, R, L, [_], s)$）使得$(\mathscr{M}_1, s_1) \leftrightarrow_{\{p\} \cup V} (\mathscr{M}, s)$。此时，可以如下构建一个初始Ind-结构$(\mathscr{M}_2, s_2)$使得$\mathscr{M}_2 = (S_2, R_2, L_2, [_], s_2)$且：

(1) 对于s_2情形：令s_2是满足下面条件的状态：

- $p \in L_2(s_2)$当且仅当$p \in L_1(s_1)$，
- 对于任意的$q \in V$，$q \in L_2(s_2)$当且仅当$q \in L(s)$，
- 对于其他的原子命题q'，$q' \in L_2(s_2)$当且仅当$q' \in L_1(s_1)$当且仅当$q' \in L(s)$。

(2) 其他情形：

- 对于所有的满足$w \in S$，$w_1 \in S_1$且$w \leftrightarrow_{\{p\} \cup V} w_1$的状态对$(w, w_1)$，如下构造$w_2 \in S_2$：

* $p \in L_2(w_2)$当且仅当$p \in L_1(w_1)$，
* 对于任意的$q \in V$，$q \in L_2(w_2)$当且仅当$q \in L(w)$，
* 对于其他的原子命题q'，$q' \in L_2(w_2)$当且仅当$q' \in L_1(w_1)$当且仅当$q' \in L(w)$。

– 对于$(w_1', w_1) \in R_1$，若w_2是基于w_1构造而成，且w_2'是基于w_1'构造而成，则令$(w_2', w_2) \in R_2$。

(3) 删除掉S_2和R_2中重复的元素。

则$(\mathscr{M}, s) \leftrightarrow_{\{p\}} (\mathscr{M}_2, s_2)$且$(\mathscr{M}_2, s_2) \leftrightarrow_V (\mathscr{M}_1, s_1)$。所以，$(\mathscr{M}_2, s_2) \models \mathrm{F}_{\mathrm{CTL}}(\varphi, p)$。因此$(\mathscr{M}_1, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\varphi, p), V)$。

另一方面，假定$(\mathscr{M}_1, s_1)$是$\mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\varphi, p), V)$的一个模型，则存在一个初始-Ind结构$(\mathscr{M}_2, s_2)$使得$(\mathscr{M}_2, s_2) \models \mathrm{F}_{\mathrm{CTL}}(\varphi, p)$且$(\mathscr{M}_2, s_2) \leftrightarrow_V (\mathscr{M}_1, s_1)$，且存在$(\mathscr{M}, s)$使得$(\mathscr{M}, s) \models \varphi$且$(\mathscr{M}, s) \leftrightarrow_{\{p\}} (\mathscr{M}_2, s_2)$。因此，由命题 3.1(i)可知$(\mathscr{M}, s) \leftrightarrow_{\{p\} \cup V} (\mathscr{M}_1, s_1)$，所以，$(\mathscr{M}_1, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\varphi, \{p\} \cup V)$。 □

从上面的命题不难看出，从公式中遗忘掉原子命题集合中的元素，可以将该集合拆成两个集合后遗忘。

推论 3.2. *对于给定的公式φ，原子命题集合V_1和V_2，下面的结论成立：*

$$\mathrm{F}_{\mathrm{CTL}}(\varphi, V_1 \cup V_2) \equiv \mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\varphi, V_1), V_2).$$

如同被遗忘的原子命题集合能被拆成两个集合的遗忘一样，下面将介绍在有些情况下从时序词的公式中遗忘掉一些原子命题可以将这些时序词提到遗忘操作的前面。

命题 3.5. *令$V \subseteq \mathscr{A}$为原子命题的集合，ϕ为CTL公式，则下面等式成立：*

(i) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V) \equiv \mathrm{AXF}_{\mathrm{CTL}}(\phi, V)$；

(ii) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi, V) \equiv \mathrm{EXF}_{\mathrm{CTL}}(\phi, V)$；

(iii) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{AF}\phi, V) \equiv \mathrm{AFF}_{\mathrm{CTL}}(\phi, V)$；

(iv) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{EF}\phi, V) \equiv \mathrm{EFF}_{\mathrm{CTL}}(\phi, V)$；

(v) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{AG}\phi, V) \equiv \mathrm{AGF}_{\mathrm{CTL}}(\phi, V)$；

(vi) $\mathrm{F}_{\mathrm{CTL}}(\mathrm{EG}\phi, V) \equiv \mathrm{EGF}_{\mathrm{CTL}}(\phi, V)$。

证明. (i) $(\Rightarrow)$ $(\mathscr{M},s)\models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi,V)$

$\Rightarrow$ 有$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}',s')$且$(\mathscr{M}',s')\models \mathrm{AX}\phi$

$\Rightarrow (\mathscr{M},s)\leftrightarrow_V(\mathscr{M}',s')$，且对任意的$(s',s'')\in R'$有$(\mathscr{M}',s'')\models\phi$ $(R'\in\mathscr{M}')$

$\Rightarrow$ 对任意的$(s,s_1)\in R$，存在$(s',s_1')\in R'$ 和$(\mathscr{M},s_1)\leftrightarrow_V(\mathscr{M}',s_1')$，且对任意的$(s',s'')\in R'$有$(\mathscr{M}',s'')\models\phi$

$\Rightarrow$ 对任意的$(s,s_1)\in R$，有$(\mathscr{M},s_1)\leftrightarrow_V(\mathscr{M}',s_1')$且$(\mathscr{M}',s_1')\models\phi$

$\Rightarrow$ 对任意的$(s,s_1)\in R$，$(\mathscr{M},s_1)\models \mathrm{F}_{\mathrm{CTL}}(\phi,V)$

$\Rightarrow (\mathscr{M},s)\models \mathrm{AXF}_{\mathrm{CTL}}(\phi,V)$。

$(\Leftarrow)$ $(\mathscr{M},s)\models \mathrm{AXF}_{\mathrm{CTL}}(\phi,V)$

$\Rightarrow$ 对任意的$(s,s')\in R$，$(\mathscr{M},s')\models \mathrm{F}_{\mathrm{CTL}}(\phi,V)$ $(R\in\mathscr{M})$

$\Rightarrow$ 对任意的$(s,s')\in R$，有$(\mathscr{M},s')\leftrightarrow_V(\mathscr{M}',s'')$且$(\mathscr{M}',s'')\models\phi$

$\Rightarrow$ 对任意的$i\geq 0$，有$(\mathscr{M},s_i')\leftrightarrow_V(\mathscr{M}_i',s_i'')$且$(\mathscr{M}_i',s_i'')\models\phi$，其中$\{s_0',s_1',\ldots\}=\{s'\mid(s,s')\in R\}$且$\mathscr{M}_i'=(S_i',R_i',L_i',[_]_i',s_i'')$ (当$i\neq j$时，假定$S_i'\cap S_j'=\emptyset$)

$\Rightarrow (\mathscr{M}^*,s)\leftrightarrow_V(\mathscr{M},s)$ 且$(\mathscr{M}^*,s)\models \mathrm{AX}\phi$，其中$\mathscr{M}^*=(S^*,R^*,L^*,[_],s)$且

- $S^*=\{s\}\cup\bigcup_{i\geq 0}S_i'$，
- $R^*=\{(s,s_i'')\mid i\geq 0\}\cup\bigcup_{i\geq 0}R_i'$，
- $L^*=\bigcup_{i\geq 0}L_i'$ 和$L^*(s)=L(s)$，其中$L\in\mathscr{M}$。

$\Rightarrow (\mathscr{M},s)\models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi,V)$。

(ii) $(\Rightarrow)$ $(\mathscr{M},s)\models \mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi,V)$

$\Rightarrow$ 存在$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}',s')$ 且$(\mathscr{M}',s')\models \mathrm{EX}\phi$

$\Rightarrow$ 存在$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}',s')$，且对一些$(s',s'')\in R'$有$(\mathscr{M}',s'')\models\phi$ $(R'\in\mathscr{M}')$

$\Rightarrow$ 对一些$(s,s_1)\in R$，有$(s',s_1')\in R'$ 和$(\mathscr{M},s_1)\leftrightarrow_V(\mathscr{M}',s_1')$，且对一些$(s',s'')\in R'$有$(\mathscr{M}',s'')\models\phi$ （$R\in\mathscr{M}$）

$\Rightarrow$ 对一些$(s,s_1)\in R$，有$(\mathscr{M},s_1)\leftrightarrow_V(\mathscr{M}',s_1')$ 和$(\mathscr{M}',s_1')\models\phi$

$\Rightarrow$ 对一些$(s,s_1)\in R$，有$(\mathscr{M},s_1)\models \mathrm{F}_{\mathrm{CTL}}(\phi,V)$

$\Rightarrow (\mathscr{M},s)\models \mathrm{EXF}_{\mathrm{CTL}}(\phi,V)$。

$(\Leftarrow)$ $(\mathscr{M},s)\models \mathrm{EXF}_{\mathrm{CTL}}(\phi,V)$

$\Rightarrow$ 对一些$(s,s')\in R$，$(\mathscr{M},s')\models \mathrm{F}_{\mathrm{CTL}}(\phi,V)$ $(R\in\mathscr{M})$

$\Rightarrow$ 对一些$(s,s')\in R$，有$(\mathscr{M},s')\leftrightarrow_V(\mathscr{M}',s'')$ 和$(\mathscr{M}',s'')\models\phi$，其中$\mathscr{M}'=(S',R',L',[_]',s'')$

$\Rightarrow (\mathscr{M}^*,s)\leftrightarrow_V(\mathscr{M},s)$ 和$(\mathscr{M}^*,s)\models \mathrm{EX}\phi$，其中$\mathscr{M}^*=(S^*,R^*,L^*,[_],s)$，

- $S^*=S\cup S'$，
- $R^*=\{(s,s'')\}\cup R\cup R'$，

- $L^* = L \cup L'$ 和$L^*(s) = L(s)$，其中$L \in \mathscr{M}$。

$\Rightarrow (\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi, V)$。

(iii)和(iv)可以分别类似(i)和(ii)来证明。

(v) ($\Rightarrow$) $(\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AG}\phi, V)$

$\Rightarrow$ 存在$(\mathscr{M}', s') \models \mathrm{AG}\phi$且$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$

$\Rightarrow$ 对$\mathscr{M}$上的每一条路径$\pi = (s = s_1, s_2, \ldots)$，存在$\mathscr{M}'$上的一条路径$\pi' = (s' = s_1, s_2', \ldots)$使得$\pi \leftrightarrow_V \pi'$，反之也成立，且对任意的$i \geq 1$有$(\mathscr{M}', s_i') \models \varphi$

$\Rightarrow$ 对π上的任意s_i'（$i \geq 1$），有$(\mathscr{M}', s_i') \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow$ 对π'上的任意s_i（$i \geq 1$），有$(\mathscr{M}, s_i) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$ $\quad (\mathrm{IR}(\mathrm{F}_{\mathrm{CTL}}(\phi, V), V))$

$\Rightarrow$ 对任意的$t \in S$，有$(\mathscr{M}, t) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V) \quad (S \in \mathscr{M})$

$\Rightarrow (\mathscr{M}, s) \models \mathrm{AGF}_{\mathrm{CTL}}(\phi, V)$。

($\Leftarrow$) $(\mathscr{M}, s) \models \mathrm{AGF}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow$ 对$\mathscr{M}$上的每一条路径$\pi = (s = s_0, s_1, \ldots)$，和对任意的$i \geq 0$有$(\mathscr{M}, s_i) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow \forall t \in S$，$(\mathscr{M}, t) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V) \quad (S \in \mathscr{M})$

$\Rightarrow \forall t \in S$，有$(\mathscr{M}, t) \leftrightarrow_V (\mathscr{M}', t')$且$(\mathscr{M}', t') \models \phi$

$\Rightarrow \forall s_i \in S$ （$i \geq 0$），有$(\mathscr{M}, s_i) \leftrightarrow_V (\mathscr{M}_i', s_i')$且$(\mathscr{M}_i', s_i') \models \phi$，其中$\mathscr{M}_i' = (S_i', R_i', L_i', [_]_i', s_i')$

$\Rightarrow (\mathscr{M}^*, s^*) \leftrightarrow_V (\mathscr{M}, s)$ 和$(\mathscr{M}^*, s^*) \models \mathrm{AG}\phi$，其中$\mathscr{M}^* = (S^*, R^*, L^*, [_], s^*)$，

- $S^* = \bigcup S_i'$，
- $s* = s_0'$，
- $R^* = \{(s_x', s_y') \mid (s_x, s_y) \in R,\ x, y \geq 0\} \cup \bigcup_{i \geq 0} R_i'$，其中$R \in \mathscr{M}$，
- 对任意的$t_i \in S_i'$，$L^*(t_i) = L_i'(t_i)$。

$\Rightarrow (\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AG}\phi, V)$。

(vi) 可类似(v)来证明。 □

3.3 本章小结

本章基于现有不同环境下的互模拟，给出了Ind-结构下的V-互模拟的定义。Ind-结构间的V-互模拟描述的是两个Ind-结构除了V中的元素之外，它们的状态转换行为是能够互相模拟的。这与遗忘理论所描述的“遗忘掉不想考虑的原子命题应该不影响剩余原子命题上的结论”一致。因此，我们使用V-互模拟刻画了原始公式与遗忘结果的模型之间的关系，从而得到了遗忘的定义。遗忘理论作为本文主要探讨的对象，本章通过研究V-互模拟的一些基本性质，探索了遗忘理论应有的一般属性，这些属性包括：

模块化性质、交换性、同质性及命题逻辑也满足的一些属性。除了这些基本性质，本章还说明了本文所给出的遗忘的定义是命题逻辑下遗忘定义的扩展。这些都为后文探索如何使用遗忘计算最强必要条件和最弱充分条件奠定了坚实的基础。

第四章　计算CTL下的遗忘：基于归结的方法

已有结果显示，任意的*CTL*公式可以转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合。归结是一种以子句为计算对象的判断可满足性的方法，本章提出一种基于归结的计算遗忘的方法。其主要思想是：首先将给定的*CTL*公式转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合（这一部分在第二章已经给出了详细的介绍），其次在相应的原子命题集合V上使用归结规则得到归结结果，最后“消除”之前引入的索引和***start***，最终得到遗忘的结果。其主要流程图如图*4.1*所示。正如本章所要说明的那样，*CTL*不具有均匀插值这一属性，基于归结的方法在有的情况下是不能计算出遗忘结果的。然而，在有些*CTL*子类下，本章提出的方法能够计算出其遗忘结果。

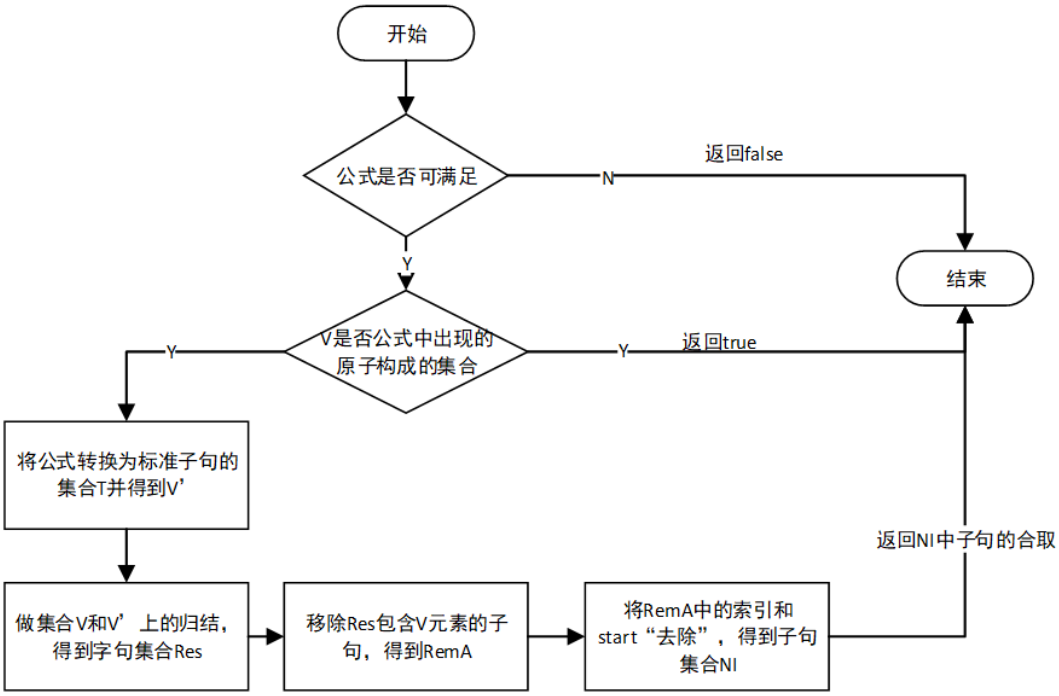


图 4.1: 基于归结的遗忘的主要流程图

4.1　引言

虽然在第二章详细介绍了命题逻辑和模态逻辑S5下基于归结的计算遗忘的方法，但值得注意的是CTL公式没有像这两种逻辑一样有标准的自身语言子句形式，而CTL的子句是有索引的。这就注定了CTL下基于归结的遗忘与上述两者的不同，

而且也应该要复杂一些（虽有不同，但也可以借鉴）。本章展示如何使用第2.4节中表2.3中的归结规则来计算CTL下的遗忘。

为了计算从CTL公式φ中遗忘掉集合V中的原子命题，需要解决如下两个主要问题：

(1) 如何表示CTL公式和带索引的CTL公式之间的关系？如在第二章中所展示的那样，将一个CTL公式转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合会引入新的原子命题和索引。虽然已有的研究说明了CTL公式可以转换为带索引的公式的集合并保证其可满足性，然而并没有表明这两种形式的公式之间的模型具有怎样的联系。本章从互模拟等价的角度探讨CTL公式和归结等过程得到的（带索引）公式之间的联系，为计算遗忘提供理论基础。

(2) 如何“移除”无关的原子命题（包括需要遗忘的原子命题和转换过程中引入的新的原子命题），以及如何“消除”索引？为此，本章给出“移除”原子命题的一般操作，并提出一种一般化的Ackermann引理。为了“消除”索引，探索几个逻辑等价关系，详见第4.2节。

本章其余部分组织如下：首先，第4.2节给出归结过程的详细介绍。其次，第节介绍如何将第4.2节得到的结果转换成CTL公式，具体分为两部分：移除索引和移除新引入的原子命题。第4.4节给出计算遗忘的算法，并分析算法的时间和空间复杂性。最后总结本章的主要工作。

4.2 CTL归结UF

这部分给出如何使用归结规则（表2.3）来计算CTL中的遗忘。这个过程的主要思想是将转换过程中获得的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句在表2.3中归结规则上做穷尽的归结，即直到不产生新的归结结果。值得注意的是该归结过程涉及的归结原子命题是需要遗忘的和转换过程中引入的新的原子命题，这是与判定可满足性下做所有原子命题的归结的不同。

令T为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合，$p$为原子命题。$T$在$p$上的*展开*（记为$\mathrm{UF}(T,p)$）是集合$T$和如下集合的并集：

$$\{\alpha \mid \alpha \text{ 是} T \text{ 中的公式关于文字} l \in \{p, \neg p\} \text{ 的归结结果}\}.$$

对于原子命题的集合V，定义$\mathrm{UF}(T,\emptyset) = T$且$\mathrm{UF}(T,\{p\} \cup V) = \mathrm{UF}(\mathrm{UF}(T,p),V)$。直观地说，$T$在$p$上的展开是对$p$做穷尽的归结，也就是直到对$p$使用规则**SRES1-8**不再产生新的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句为止（即使使用了重写规则和可能规则之后也不会产生新的子句）。

下面的命题展示了对任意的CTL公式φ，从$\mathrm{UF}(T_\varphi,V)$ 中移除掉含有V中元素的子句得到的结果在不考虑V中元素和新引入的元素的情况下是互模拟等价的。下文中记$ERes(\varphi,V)=\{\alpha\in \mathrm{UF}(T_\varphi,V)\mid Var(\alpha)\cap V=\emptyset\}$。

命题 4.1. *令φ 为一个CTL公式，$V\subseteq\mathscr{A}$为原子命题的集合。则$T_\varphi\equiv_U \mathrm{ERes}(\varphi,V)$，其中$U=Var(\mathrm{UF}(T_\varphi,V))-(Var(\varphi)-V)$。*

证明. 从两个方面来证明这一结论：**(F1)** $T_\varphi\equiv_U \mathrm{UF}(T_\varphi,V)$，**(F2)** $\mathrm{UF}(T_\varphi,V)\equiv_U ERes(\varphi,V)$。为了方便，定义如下由$\mathrm{SNF}^g_{\mathrm{CTL}}$ 子句集合构成的序列：$T_0=T_\varphi,T_1,T_2,\ldots,T_n=\mathrm{UF}(T_\varphi,V)$，其中$T_{i+1}=T_i\cup R_i$ $(0\leq i<n)$、R_i 是对$\Pi\subseteq T_i$使用一条匹配的规则r且该规则归结的原子命题为$p\in V$，这一过程记为$\Pi\rightarrow_r R_i$。

(F1) 为了证明$T_\varphi\equiv_U \mathrm{UF}(T_\varphi,V)$，这里证明对任意的$0\leq i<n$ 有$T_i\equiv_U T_{i+1}$。

(1) 若$r\in\{$**(SRES1)**, …, **(SRES8)**, (**RW1**), (**RW2**)$\}$，则$T_i\equiv_{\{p\}} T_{i+1}$，其中。

一方面，显然$\Pi\models R_i$，所以$T_i\models T_{i+1}$。另一方面$T_i\subseteq T_{i+1}$，所以$T_{i+1}\models T_i$。

(2) 这里证明若$\Pi\rightarrow_r R_i$且$r=$**(ERES1)**，则$T_i\equiv_{\{l,w^{\mathrm{A}}_{\neg l}\}} T_{i+1}$，其中$l=p$或$l=\neg p$，$w^{\mathrm{A}}_{\neg l}$是与子句$Q\rightarrow \mathrm{AF}\neg l$相关的新的原子命题，$l$是文字（即：$p$或者$\neg p$）。

在文章[113]中已经证明$\Pi\models R_i$，因此有$T_{i+1}=T_i\cup\Lambda^{\mathrm{A}}_{\neg l}$，其中$\Lambda^{\mathrm{A}}_{\neg l}$是通过使用表2.1中的转换规则作用到$R_i$上得到的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合（请查看文章[114]获取更加详细的描述）。显然，对所有的$(\mathscr{M}_1,s_1)\in Mod(T_i=X\cup\Pi)$都存在一个$(\mathscr{M}_2,s_2)\in Mod(T_{i+1}=T_i\cup\Lambda^{\mathrm{A}}_{\neg l})$使得$(\mathscr{M}_1,s_1)\leftrightarrow_{\langle\{p,w^{\mathrm{A}}_{\neg l}\},\emptyset\rangle}(\mathscr{M}_2,s_2)$，且对任意的$(\mathscr{M}_2,s_2)\in Mod(T_{i+1}=T_i\cup\Lambda^{\mathrm{A}}_{\neg l})$也存在一个$(\mathscr{M}_1,s_1)\in Mod(T_i=X\cup\Pi)$使得$(\mathscr{M}_1,s_1)\leftrightarrow_{\{p,w^{\mathrm{A}}_{\neg l}\}}(\mathscr{M}_2,s_2)$。又$\{p,w^{\mathrm{A}}_{\neg l}\}\subseteq U$，由命题3.2可知$T_i\equiv_U T_{i+1}$。

当规则为**(ERES2)**时可以类似地证明。

总之，$V\subseteq U$，因此由推论3.1(iii)可知对任意的$0\leq i<n$ 有$T_i\equiv_U T_{i+1}$。又因为$\equiv_U$为一个等价关系，所以$T_\varphi\equiv_U \mathrm{UF}(T_\varphi,V)$。

(F2) 不失一般性地，假设$V=\{p\}$，C_i $(i=1,2)$为经典子句，$l=p$ 或$l=\neg p$。显然$\mathrm{UF}(T_\varphi,V)\models ERes(\varphi,V)$，这里证明对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$ （其中$\mathscr{M}=(S,R,L,[_],s)$），存在一个初始Ind-结构$\mathscr{K}'=(\mathscr{M}',s')$ 使得$\mathscr{K}\leftrightarrow_U\mathscr{K}'$ 且$\mathscr{K}'\models\mathrm{UF}(T_\varphi,V)$。

因为p 只出现在$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的右手边，这里从下面几点证明上述结论成立。

(1) 假定$\mathrm{UF}(T_\varphi,V)$有全局子句，则对于任意的$C=\top\rightarrow C_1\vee l\in\mathrm{UF}(T_\varphi,V)$：

(a) 如果不存在$C'\in\mathrm{UF}(T_\varphi,V)$ 使得C 和C' 在p上是可归结的，则$\mathrm{UF}(T_\varphi,V)$中不存在除了Pt-某时子句之外的子句C' 包含文字$\neg l$，其中$Pt\in\{\mathrm{A},\mathrm{E}\}$。

若对任意其他的子句C'，$p \notin Var(C')$，则对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$可如下构造$(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$（即$s'=s$），其中$L'$与$L$一样，除了对任意的$s_1\in S$，若$(\mathscr{M},s_1)\not\models C_1\vee l$则“若$l=p$令$L'(s_1)=L(s_1)\cup\{p\}$，否则令$L'(s_1)=L(s_1)-\{p\}$”。显然$(\mathscr{M},s)\leftrightarrow_{\{p\}}(\mathscr{M}',s')$和$(\mathscr{M}',s')\models C'\wedge C$。

若$C'=Q\rightarrow Pt\mathrm{F}\neg l$，不失一般性地，令$l=p$。对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，如下构造$(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$，其中$L'$与$L$相同，除了$L'(s)=L(s)\cup\{p\}$，且对任意具有$(s,s')\in R$关系的状态$s'\in S'$，令$L'(s)=L(s)-\{p,q\}$，其中$q\in(Var(\mathrm{UF}(T_\varphi,V))-Var(\varphi))$是负出现在$C_1$中的原子命题，且对任意其他的非初始状态$s''\in S$，$L'(s'')=(L(s'')-\{Q\})\cup\{p\}$（$Q$在$Pt$-某时子句中是一个原子）。显然$(\mathscr{M},s)\leftrightarrow_{\{p,q\}}(\mathscr{M}',s')$和$(\mathscr{M}',s')\models C'\wedge C$。

(b) 若存在子句$C'\in\mathrm{UF}(T_\varphi,V)$使得$C$和$C'$在$p$上是可归结的：

(i) 若$C'=Q\rightarrow Pt\mathrm{X}(C_2\vee\neg l)$（令$Pt=\mathrm{A}$，$Pt=\mathrm{E}$可类似地证明），则有$Q\rightarrow\mathrm{AX}(C_1\vee C_2)\in\mathrm{UF}(T_\varphi,V)$。因此，对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，可如下构造$\mathscr{K}'=(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$（即$s'=s$），其中$L'$与$L$一样，除了对任意的$s_1\in S$，若$(\mathscr{M},s_1)\not\models Q$，则对任意的$(s_1,s_2)\in R$若$(\mathscr{M},s_2)\not\models C_1$则“若$l=p$，则令$L'(s_2)=L(s_2)\cup\{p\}$，否则令$L'(s_2)=L(s_2)-\{p\}$”，否则，若$(\mathscr{M},s_2)\models C_1\wedge\neg C_2$，则“若$l=p$，则令$L'(s_2)=L(s_2)-\{p\}$，否则令$L'(s_2)=L(s_2)\cup\{p\}$”；否则，若$(\mathscr{M},s_2)\models\neg C_1\wedge C_2$，则“若$l=p$，则令$L'(s_2)=L(s_2)\cup\{p\}$，否则$L'(s_2)=L(s_2)-\{p\}$”。容易检查$\mathscr{K}\leftrightarrow_{\{p\}}\mathscr{K}'$且$\mathscr{K}'\models C'\wedge C$。

(ii) 若$C'=Q\rightarrow Pt\mathrm{F}\neg l$。不失一般性地，假设$l=p$。$C$和$C'$在$p$上是可归结的，则一定存在子句的集合$\{P_1^1\rightarrow *C_1^1,\ldots,P_{m_1}^1\rightarrow *C_{m_1}^1,P_1^n\rightarrow *C_1^n,\ldots,P_{m_n}^1\rightarrow *C_{m_n}^1\}$使得$*$要么要么为空字符串，要么为$\{\mathrm{AX},\mathrm{E}_{\langle ind\rangle}\mathrm{X}\}$中的一个（$\neg C_1\rightarrow l$为子句集合中的一个）使得$\bigvee_{i=1}^n\bigwedge_{j=1}^{m_i}P_j^i\rightarrow\mathrm{EXEG}l$。因此，通过使用规则ERES1（ERES2规则类似）可以得到一个子句$C''=\top\rightarrow\neg Q\vee\neg p\vee C_1$，因此在使用规则SRES8在$C$和$C''$上后得到子句$\top\rightarrow\neg Q\vee C_1$。在这种情况下，对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，可以如下构造$\mathscr{K}'=(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$（即$s'=s$），其中$L'$与$L$一样，除了对任意的$s_1\in S$，若$(\mathscr{M},s_1)\models Q$，则令$L'(s_1)=L(s_1)-\{p\}$，否则令$L'(s_1)=L(s_1)\cup\{p\}$。可以检查$\mathscr{K}\leftrightarrow_{\{p\}}\mathscr{K}'$和$\mathscr{K}'\models C'\wedge C$。

(ii) 可以类似地证明其他类型的子句，且得到$\mathscr{K}\leftrightarrow_U\mathscr{K}'$和$\mathscr{K}'\models\mathrm{UF}(T_\varphi,V)$。

(2) 考虑Pt-步子句的情形。令$C\in\mathrm{UF}(T_\varphi,V)$为$Q\rightarrow\mathrm{AX}(C_1\vee l)$。不失一般性地，假设存在某些子句$C'\in\mathrm{UF}(T_\varphi,V)$使得$C$和$C'$在$p$上是可归结的（$l=p$）。

若$C'=Q_1\rightarrow Pt\mathrm{X}(C_2\vee\neg l)$（令$Pt=\mathrm{E}_{ind}$，$Pt=\mathrm{A}$的情形可以类似地证明），因此有$Q\wedge Q_1\rightarrow\mathrm{E}_{ind}\mathrm{X}(C_1\vee C_2)\in\mathrm{UF}(T_\varphi,V)$。所以对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，

如下构造$\mathscr{K}'=(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$（即$s'=s$），其中$L'$与$L$一样，除了对任意的$s_1\in S$

(i) 若$(\mathscr{M},s_1)\not\models Q\wedge Q_1$则“若$(\mathscr{M},s_1)\models\neg Q\wedge Q_1$则（若对$(s_1,s_2')\in\pi_s^{\langle ind\rangle}$有$(\mathscr{M},s_2')\not\models C_2$则令$L'(s_2')=L(s_2')-\{p\}$否则令$L'(s_2')=L(s_2')$），否则若$(\mathscr{M},s_1)\models Q\wedge\neg Q_1$则对任意的$(s_1,s_2)\in R$（若$(\mathscr{M},s_2)\not\models C_1$则令$L'(s_2)=L(s_2)\cup\{p\}$，否则令$L'(s_2')=L(s_2')$），否则令$L'(s_2')=L(s_2')$”。

(ii) 否则若$(\mathscr{M},s_1)\models Q\wedge Q_1$，则对$(s_1,s_2')\in\pi_s^{\langle ind\rangle}$有$(\mathscr{M},s_2')\models C_1\vee C_2$。因此，若$(\mathscr{M},s_2')\models C_1\wedge\neg C_2$则$L'(s_2')=L(s_2')-\{p\}$，否则若$(\mathscr{M},s_2')\models\neg C_1\wedge C_2$则令$L'(s_2')=L(s_2')\cup\{p\}$，否则令$L'(s_2')=L(s_2')$。对其他满足$(s_1,s_2)\in R$和$s_2\neq s_2'$的状态$s_2$，若$(\mathscr{M},s_1)\models Q$和$(\mathscr{M},s_2)\models\neg C_1$，则令$L'(s_2)=L(s_2)\cup\{p\}$，否则令$L'(s_2')=L(s_2')$。

容易检查$\mathscr{K}\leftrightarrow_{\{p\}}\mathscr{K}'$和$\mathscr{K}'\models C'\wedge C$，其中$\mathscr{K}'=(\mathscr{M}',s')$。

若$C'=Q_1\rightarrow Pt\mathrm{F}\neg l$（令$Pt=\mathrm{A}$，$Pt=\mathrm{E}$的情形可类似地证明）。若$C$和$C'$在$p$上是可归结的，则必须存在一个包含子句$\neg C_1\rightarrow l$的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合$\{P_1^1\rightarrow *C_1^1,\ldots,P_{m_1}^1\rightarrow *C_{m_1}^1,P_1^n\rightarrow *C_1^n,\ldots,P_{m_n}^1\rightarrow *C_{m_n}^1\}$使得$*$为空字符串或集合$\{\mathrm{AX},\mathrm{E}_{\langle ind\rangle}\mathrm{X}\}$中的一个，且$\bigvee_{i=1}^n\bigwedge_{j=1}^{m_i}P_j^i\rightarrow\mathrm{EXEG}l$。在这种情况下，对惹妞的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，如下构造$(\mathscr{M}',s')$：令$\mathscr{M}'=(S,R,L',[_],s)$（即$s'=s$），其中$L'$和$L$一样，除了对任意的状态$s'\in S'$，若$L'(s')\models Q$和$(s',s'')\in R$，则令$L'(s'')=L(s'')\cup\{p\}$，若$(\mathscr{M},s)\models Q_1$，则令$L'(s)=L(s)-\{p\}$。显然$(\mathscr{M},s)\leftrightarrow_{\{p\}}(\mathscr{M}',s')$和$(\mathscr{M}',s')\models C'\wedge C$成立。

因此，对任意的$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$（$\mathscr{M}=(S,R,L,[_],s)$），存在一个初始Ind-结构$\mathscr{K}'=(\mathscr{M}',s')$使得$\mathscr{K}\leftrightarrow_U\mathscr{K}'$和$\mathscr{K}'\models\mathrm{UF}(T_\varphi,V)$成立。 □

例 4.1 (例3.3的延续). 令$V=\{p,r\}$。则$\mathrm{UF}(T_\varphi,V\cup\{x,y,z\})$除了例3.3中的子句，还包含如下子句：

(1) **start** $\rightarrow r$	$(1,2,\mathbf{SRES5})$	(2) **start** $\rightarrow x\vee y$	$(1,4,\mathbf{SRES5})$
(3) $\top\rightarrow\neg z\vee y\vee f\vee m$	$(3,4,\mathbf{SRES8})$	(4) $y\rightarrow\mathrm{AX}(f\vee m\vee y)$	$(3,8,\mathbf{SRES6})$
(5) $\top\rightarrow\neg z\vee x\vee p$	$(4,5,\mathbf{SRES8})$	(6) $\top\rightarrow\neg z\vee x\vee q$	$(4,6,\mathbf{SRES8})$
(7) $y\rightarrow\mathrm{AX}(x\vee p)$	$(5,8,\mathbf{SRES6})$	(8) $y\rightarrow\mathrm{AX}(x\vee q)$	$(6,8,\mathbf{SRES6})$
(9) **start** $\rightarrow f\vee m\vee y$	$(3,(2),\mathbf{SRES5})$	(10) **start** $\rightarrow x\vee p$	$(5,(2),\mathbf{SRES5})$
(11) **start** $\rightarrow x\vee q$	$(6,(2),\mathbf{SRES5})$	(12) $\top\rightarrow p\vee\neg z\vee f\vee m$	$(5,(3),\mathbf{SRES8})$
(13) $\top\rightarrow q\vee\neg z\vee f\vee m$	$(6,(3),\mathbf{SRES8})$	(14) $y\rightarrow\mathrm{AX}(p\vee f\vee m)$	$(5,(4),\mathbf{SRES6})$
(15) $y\rightarrow\mathrm{AX}(q\vee f\vee m)$	$(6,(4),\mathbf{SRES6})$	(16) **start** $\rightarrow f\vee m\vee p$	$(5,(9),\mathbf{SRES5})$
(17) **start** $\rightarrow f\vee m\vee q$	$(6,(9),\mathbf{SRES5})$		

在从$\mathrm{UF}(T_\varphi, V\cup\{x,y,z\})$中移除掉包含$V$中元素的子句后，得到$ERes(\varphi,V)$，其包含如下子句：

$$\begin{aligned}
&\mathbf{start}\to z,\quad \mathbf{start}\to f\vee m\vee q,\quad \mathbf{start}\to x\vee y,\quad \mathbf{start}\to q\vee x,\quad \mathbf{start}\to f\vee m\vee y,\\
&\top\to f\vee m\vee\neg x,\quad \top\to q\vee f\vee m\vee\neg z,\quad \top\to f\vee m\vee\neg z\vee y,\\
&\top\to q\vee x\vee\neg z,\quad \top\to x\vee y\vee\neg z,\quad \top\to q\vee\neg y,\quad z\to \mathrm{AF}x,\\
&y\to \mathrm{AX}(q\vee f\vee m),\quad y\to \mathrm{AX}(x\vee q),\quad y\to \mathrm{AX}(x\vee y),\quad y\to \mathrm{AX}(f\vee m\vee y).
\end{aligned}$$

可以看出，尽管$ERes(\varphi,V)$中不包含具有索引的公式，但是有的子句包含出现在T_φ中的新原子命题。

4.3　转换$\mathrm{SNF}^g_{\mathrm{CTL}}$子句为CTL公式

在上一节中描述了归结过程，但是正如归结规则和转换规则所示，在这两个过程中会引入索引和新的原子命题。本节介绍如何移除掉这些新引入的元素。

下面的引理表明，可以移除掉$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集合中的索引而保持互模拟等价。

引理 4.1. 令$j\in\mathscr{I}$，$\psi_i,\varphi_i\ (1\le i\le n)$为*CTL*公式。有：

(i) $\{\psi_i\to \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i \mid 1\le i\le n\}\equiv\{(\bigwedge_{i\in S}\psi_i)\to \mathrm{E}_{\langle j\rangle}\mathrm{X}(\bigwedge_{i\in S}\varphi_i)\mid S\subseteq\{1,\dots,n\}\}$，

(ii) $\{\psi_i\to \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i \mid 1\le i\le n\}\equiv_\emptyset\{(\bigwedge_{i\in S}\psi_i)\to \mathrm{EX}(\bigwedge_{i\in S}\varphi_i)\mid S\subseteq\{1,\dots,n\}\}$，

(iii) $\{(\psi_1\to \mathrm{E}_{\langle j\rangle}\mathrm{F}\varphi_1),(\psi_2\to \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_2)\}\equiv_\emptyset$

$$(\psi_1\to\varphi_1\vee\mathrm{EXEF}\varphi_1)\wedge(\psi_2\to\mathrm{EX}\varphi_2)\wedge(\psi_1\wedge\psi_2\to((\varphi_1\wedge\mathrm{EX}\varphi_2)\vee\mathrm{EX}(\varphi_2\wedge\mathrm{EF}\varphi_1))).$$

证明. (i) ($\Rightarrow$) 对等式左边公式的任意模型$(\mathscr{M},s_0)$，若$(\mathscr{M},s_0)\models\bigwedge_{i=1}^m P_{j_i}$（$j_i\in\{1,\dots,n\}$，$1\le m\le n$），则存在$s_0$的下一个状态$s_1$使得$(s_0,s_1)\in[j]$和$(\mathscr{M},s_1)\models\bigwedge_{i=1}^m\varphi_{j_i}$。由$[j]$的定义可知$(s_0,s_1)\in R$，因此$(\mathscr{M},s_0)\models\bigwedge_{i=1}^m P_{j_i}\to \mathrm{E}_{\langle j\rangle}\mathrm{X}(\bigwedge_{i=1}^m\varphi_{j_i})$。

($\Leftarrow$) 显然等式左边的集合为右边的子集，因此：$\{(\bigwedge_{i\in S}\psi_i)\to \mathrm{E}_{\langle j\rangle}\mathrm{X}(\bigwedge_{i\in S}\varphi_i)\mid S\subseteq\{1,\dots,n\}\}\models\{\psi_i\to \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i\mid 1\le i\le n\}$。

(ii) ($\Rightarrow$) 对等式左边公式的任意模型$(\mathscr{M},s_0)$，若$(\mathscr{M},s_0)\models\bigwedge_{i=1}^m P_{j_i}$（$j_i\in\{1,\dots,n\}$，$1\le m\le n$），则存在$s_0$的下一个状态$s_1$使得$(s_0,s_1)\in[j]$且$(\mathscr{M},s_1)\models\bigwedge_{i=1}^m\varphi_{j_i}$。由$[j]$的定义可知$(s_0,s_1)\in R$，因此$(\mathscr{M},s_0)\models\bigwedge_{i=1}^m P_{j_i}\to \mathrm{EX}(\bigwedge_{i=1}^m\varphi_{j_i})$。

($\Leftarrow$) 对等式右边公式的任意模型$(\mathscr{M},s_0)$，若$(\mathscr{M},s_0)\models\bigwedge_{i=1}^m P_{j_i}$（$j_i\in\{1,\dots,n\}$，$1\le m\le n$），则存在$s_0$的下一个状态$s_1$使得$(\mathscr{M},s_1)\models\bigwedge_{i=1}^m\varphi_{j_i}$。容易构造一个初始Ind-结

算法 4.1 RM-index(Σ)

Input: A finite set Σ of $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ clauses
Output: A set of CTL formulas

foreach *maximal subset* Δ *of* E*-step clauses in* Σ *with a same index* $\langle i\rangle$ **do**
  if *There is an* E*-sometime clause* $\alpha \in \Sigma$ *with the index* $\langle i\rangle$ **then**
    foreach $\beta \in rei(\Delta)$ **do** $\Sigma \leftarrow \Sigma \cup rfi(\alpha,\beta)$ $\Sigma \leftarrow \Sigma - \{\alpha\}$
  end
  $\Sigma \leftarrow \Sigma - \Delta \cup rxi(\Delta)$
end
return Σ

构$(\mathscr{M}',s_0)$（$\mathscr{M}'=(S,R,L,[_]',s_0)$）使得$(\mathscr{M}',s_0)$与$(\mathscr{M},s_0)$相同，除了$(s_0,s_1)\in[j]'$，即$(\mathscr{M},s_0)\leftrightarrow_{\emptyset}(\mathscr{M}',s_0)$且$(\mathscr{M}',s_0)\models\{\psi_i\rightarrow \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i \mid 1\le i\le n\}$。

(iii)的证明与(ii)的证明类似。 □

本文将引理4.1中(i)、(ii)、(iii)等号$\equiv_*$（$*\in$｛空字符串，空集｝）的右边分别用$rei(\{\alpha_i \mid 1\le i\le n\})$、$rxi(\{\alpha_i \mid 1\le i\le n\})$、$rfi(\{\beta_1,\alpha_2\})$来表示，其中$\alpha_i=\psi_i\rightarrow \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i$ $(1\le i\le n)$和$\beta_1=\psi_1\rightarrow \mathrm{E}_{\langle j\rangle}\mathrm{F}\varphi_1$。

因为$\mathrm{EX}\varphi_1\wedge\mathrm{EX}\varphi_2\not\models\mathrm{EX}(\varphi_1\wedge\varphi_2)$，引理4.1(i)的目的是为了保证若$\psi_1$和$\psi_2$在当前状态满足，则$\varphi_1$和$\varphi_2$被同一条路径满足，这可以推广到多个$\psi_i$ $(1\le i\le n)$的情形。(iii)表示可以将每一个E-某时子句和一个E-步子句结合得到新的满足互模拟等价的CTL公式。(ii)表明拥有相同索引的E-步子句可以结合成新的满足互模拟等价的CTL公式。这一过程可由算法4.1实现。下面的推论是上述引理的一个简单应用，它表明当移除索引之后Σ 和RM-index(Σ)是互模拟等价的。

推论 4.1. *令φ 为一个CTL公式、$V\subseteq\mathscr{A}$为原子命题的集合、$\Sigma=\mathrm{ERes}(\mathrm{UF}(\varphi,V\cup U),V)$，其中$U=Var(T_\varphi)-Var(\varphi)$。则$RM\text{-}index(\Sigma)\equiv_{\emptyset}\Sigma$。*

通过下面的定理，可以移除掉一些新引入的原子命题而保持互模拟等价。

引理 4.2 (一般化的Ackermann引理，Generalised Ackermann's Lemma)**.** *令x为一个原子命题、$\Delta=\{\mathrm{AG}(\top\rightarrow\neg x\vee C_1),\ldots,\mathrm{AG}(\top\rightarrow\neg x\vee C_n),\mathrm{AG}(x\rightarrow B_1),\ldots,\mathrm{AG}(x\rightarrow B_m)\}$为只包含一个$x$的CTL公式的集合（$n,m\ge 1$）、$\Gamma$为$x$正出现在其中的有限个CTL公式的集合。则有:*

$$\Gamma\cup\Delta\equiv_{\{x\}}\Gamma\left[x/\bigwedge(\{C_i \mid 1\le i\le n\}\cup\{B_i \mid 1\le i\le m\})\right]. \tag{4.1}$$

证明. 令$\varphi=\bigwedge(\{C_i \mid 1\le i\le n\}\cup\{B_i \mid 1\le i\le m\})$。不失一般性地，令$\Gamma$为一个CTL公式，$\mathscr{M}=(S,R,L,[\_],s_0)$，$\psi_i(x)$ $(i=\{1,2\})$为x正出现在其中的CTL公式。

($\Rightarrow$) 对公式$\Gamma\wedge\bigwedge\Delta$的任意模型$(\mathscr{M},s_0)$，这里证明通过归纳公式$\Gamma$的结构证明$(\mathscr{M},s_0)\models\Gamma[x/\varphi]$。

基始. 令$\Gamma = x$，因为$(\mathscr{M}, s_0) \models x$，则显然$(\mathscr{M}, s_0) \models \varphi$成立。

归纳步. (1) 令$\Gamma = \psi_1(x) \wedge \psi_2(x)$。由归纳假设可得$(\mathscr{M}, s_0) \models \Gamma[x/\varphi]$。

(2) 令$\Gamma = \mathrm{EX}\psi_1(x)$。

$(\mathscr{M}, s_0) \models \Gamma \wedge \bigwedge \Delta$

$\Rightarrow$ 存在$(s_0, s_1) \in R$使得$(\mathscr{M}, s_1) \models \psi_1(x)$和$(\mathscr{M}, s_1) \models \bigwedge \Delta$成立

$\Rightarrow$ 由归纳假设可知$(\mathscr{M}, s_1) \models \psi_1(x)[x/\varphi]$

$\Rightarrow (\mathscr{M}, s_0) \models \mathrm{EX}\psi_1(x)[x/\varphi]$

$\Rightarrow (\mathscr{M}, s_0) \models (\mathrm{EX}\psi_1(x))[x/\varphi]$。

(3) 令$\Gamma = \mathrm{AX}\psi_1(x)$。

$(\mathscr{M}, s_0) \models \Gamma \wedge \bigwedge \Delta$

$\Rightarrow$ 对任意的$(s_0, s_1) \in R$，有$(\mathscr{M}, s_1) \models \psi_1(x)$和$(\mathscr{M}, s_1) \models \bigwedge \Delta$

$\Rightarrow$ 对任意的$(s_0, s_1) \in R$，由归纳假设可知$(\mathscr{M}, s_1) \models \psi_1(x)[x/\varphi]$

$\Rightarrow (\mathscr{M}, s_0) \models \mathrm{AX}\psi_1(x)[x/\varphi]$

$\Rightarrow (\mathscr{M}, s_0) \models (\mathrm{AX}\psi_1(x))[x/\varphi]$。

(4) 令$\Gamma = \mathrm{E}(\psi_1(x)\mathrm{U}\psi_2(x))$。

$(\mathscr{M}, s_0) \models \Gamma \wedge \bigwedge \Delta$

$\Rightarrow$ 存在一条路径$\pi = (s_0, s_1, \ldots) \in R$使得对于某个$j \geq 0$有$(\mathscr{M}, s_j) \models \psi_2(x)$，对任意的$0 \leq i < j$有$(\mathscr{M}, s_i) \models \psi_1(x)$，且对所有的$x \geq 0$有$(\mathscr{M}, s_x) \models \bigwedge \Delta$

$\Rightarrow$ 由归纳假设可知，存在一条路径$\pi = (s_0, s_1, \ldots) \in R$使得对于某个$j \geq 0$有$(\mathscr{M}, s_j) \models \psi_2(x)[x/\varphi]$，和对任意的$0 \leq i < j$有$(\mathscr{M}, s_i) \models \psi_1(x)[x/\varphi]$

$\Rightarrow (\mathscr{M}, s_0) \models \mathrm{E}((\psi_1(x)[x/\varphi])\,\mathrm{U}\,(\psi_2(x)[x/\varphi]))$

$\Rightarrow (\mathscr{M}, s_0) \models (\mathrm{E}(\psi_1(x)\mathrm{U}\psi_2(x)))[x/\varphi]$。

可以类似证明其他情况。

($\Leftarrow$) 对$\Gamma[x/\varphi]$的任意模型$(\mathscr{M}, s_0)$（$\mathscr{M} = (S, R, L, [_], s_0)$），构造一个初始Ind-Kripke结构$\mathscr{M}' = (S, R, L', [_], s_0)$，其中$L'$与$L$一样，除了对任意$s' \in S'$，若$(\mathscr{M}', s') \models \varphi$，则令$L'(s') = L(s) \cup \{x\}$，否则令$L'(s') = L(s) - \{x\}$（即$(\mathscr{M}', s_0) \models x \leftrightarrow \varphi$）。

容易证明$(\mathscr{M}, s_0) \leftrightarrow_{\{x\}} (\mathscr{M}', s_0')$且$(\mathscr{M}', s_0') \models \Gamma \cup \Delta$。 □

在这种情形下，记$\mathrm{GAL}(\Gamma \cup \Delta, \{x\}) = \Gamma[x/\bigwedge(\{C_i \mid 1 \leq i \leq n\} \cup \{B_i \mid 1 \leq i \leq m\})]$。对于CTL公式的集合$\Sigma$，用$\mathrm{GAL}(\Sigma, \{x\})$表示$\mathrm{GAL}(\Gamma \cup \Delta, \{x\})$，其中$\Delta \subseteq \Sigma$为与引理4.2中$\Delta$有相同性质的出现在$\Sigma$中有唯一负出现$x$的公式的集合，$\Gamma \subseteq \Sigma$是$x$正出现在其中的公式的集合。对于原子命题集合$V$，定义

$$\mathrm{GAL}(\Sigma, V \cup \{x\}) = \mathrm{GAL}(\mathrm{GAL}(\Sigma, \{x\}), V).$$

算法 4.2 CTL-forget(φ, V)

Input: A CTL formula φ and a set V of atoms
Output: A conjunction of formulas

if $\varphi \equiv \bot$ **then return** $\bot$ **if** $V = Var(\varphi)$ **then return** $\top$ $T_\varphi \leftarrow \mathrm{SNF}^g_{\mathrm{CTL}}(\varphi)$; // 将φ转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句
$\Sigma \leftarrow \mathrm{UF}(T_\varphi, V \cup U)$ where $U = Var(T_\varphi) - Var(\varphi)$; // 展开
$\Sigma \leftarrow ERes(\Sigma, V)$; // 移除包含V中元素的子句
$\Sigma \leftarrow$ RM-index(Σ) ; // 从Σ移除索引
$\Sigma \leftarrow \mathrm{GAL}(\Sigma, Var(\Sigma) - Var(\varphi))$; // 移除留存的新的原子命题
Replacing each initial clause "$\mathrm{AG}(\mathbf{start} \rightarrow \varphi)$" in Σ by $\mathrm{AG}\varphi$; // 去除**start**
return Σ

例 4.2 (例4.1的延续). 首先考虑原子命题x、$\Delta = \{\top \rightarrow f \vee m \vee \neg x\}$和$\Gamma = ERes(\varphi, V) - \Delta$。$\Gamma$中包含$x$的公式关于$x$都为正的，因此$\Gamma[x/(f \vee m)]$包含如下公式：

$$\begin{aligned}&\mathbf{start} \rightarrow z, \quad \mathbf{start} \rightarrow f \vee m \vee q, \quad \mathbf{start} \rightarrow f \vee m \vee y,\\ &\top \rightarrow q \vee f \vee m \vee \neg z, \quad \top \rightarrow f \vee m \vee y \vee \neg z, \quad \top \rightarrow q \vee \neg y, \quad z \rightarrow \mathrm{AF}(f \vee m),\\ &y \rightarrow \mathrm{AX}(q \vee f \vee m), \quad y \rightarrow \mathrm{AX}(f \vee m \vee y).\end{aligned}$$

第二步考虑原子命题z、$\Delta' = \{\top \rightarrow q \vee f \vee m \vee \neg z, \top \rightarrow f \vee m \vee y \vee \neg z, z \rightarrow \mathrm{AF}(f \vee m)\}$和$\Gamma' = \Gamma[x/(f \vee m)] - \Delta'$，其中$z$正出现在$\Gamma'$中。因此，$\Gamma'' = \Gamma'[z/(q \vee f \vee m) \wedge (f \vee m \vee y) \wedge \mathrm{AF}(f \vee m)]$包含如下公式：

$$\begin{aligned}&\mathbf{start} \rightarrow (q \vee f \vee m) \wedge (f \vee m \vee y) \wedge \mathrm{AF}(f \vee m), \quad \mathbf{start} \rightarrow f \vee m \vee q, \quad \mathbf{start} \rightarrow f \vee m \vee y,\\ &\top \rightarrow q \vee \neg y, \quad y \rightarrow \mathrm{AX}(q \vee f \vee m), \quad y \rightarrow \mathrm{AX}(f \vee m \vee y).\end{aligned}$$

不难证明$ERes(\varphi, V) \equiv_{\{x,z\}} \Gamma''$。因为$\Gamma''$包含一个公式其关于$y$既不是正的也不是负的，因此这里不能对$\Gamma''$和$y$使用上述过程。

4.4 算法及其复杂性分析

现在可以给出计算CTL下遗忘的算法——算法4.2。该算法的输入为一个CTL公式φ和一个原子命题的集合，输出为一个与φ互模拟等价的CTL公式。

定理 4.1. 令φ为一个*CTL*公式、$V \subseteq \mathscr{A}$、Σ =*CTL-forget*(φ, V)和$U = Var(\Sigma) - Var(\varphi)$。则

(i) $\Sigma \equiv_{V \cup U} \varphi$，

(ii) 若$U = \emptyset$，则$\Sigma \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$。

证明. (i) 这一结论直接来源于命题3.2和4.1，引理4.1和4.2。

(ii) 若$U=\emptyset$，则由(i)可知$\Sigma \equiv_V \varphi$。又由于Σ和$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$都是$V$-无关的且都不包含索引，因此由$\mathrm{F}_{\mathrm{CTL}}(\varphi,V) \equiv_V \varphi$可知$\Sigma \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。 □

例 4.3 (例4.2的延续)**.** 容易看出CTL-forget$(\varphi,\{p,r\})$包含下面的公式

$$(q \vee f \vee m) \wedge (f \vee m \vee y) \wedge \mathrm{AF}(f \vee m), \quad \mathrm{AG}(\top \rightarrow q \vee \neg y),$$
$$\mathrm{AG}(y \rightarrow \mathrm{AX}(q \vee f \vee m)), \quad \mathrm{AG}(y \rightarrow \mathrm{AX}(f \vee m \vee y)).$$

尽管如此，有的CTL公式的遗忘结果总是存在的，如下面的结论所示。

命题 4.2. *给定CTL公式φ，若φ满足下面约束：φ中不包括操作符$Pt\mathscr{T}$（其中$Pt \in \{\mathrm{A},\mathrm{E}\}$且$\mathscr{T} \in \{\mathrm{U},\mathrm{G}\}$），且对于任意的原子命题$p \in V$，若$p$和$\neg p$出现在同一时序算子的范围内；则*$\mathrm{ERes}(\varphi,V) \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。

证明. 不失一般性地假设$V=\{p\}$。对任意上述所说形式的CTL公式φ，假定$\varphi = \varphi_1 \wedge \mathrm{AXEF}\varphi_2$，其中$p \notin Var(\varphi_1)$且$\varphi_2$是一个包含子句$C_1 = \neg p \vee \psi_1$和$C_2 = p \vee \psi_2$的CNF (conjunctive normal form)公式。φ可以被转换为包含集合$\Pi = \{\top \rightarrow \neg x \vee p \vee \psi_1, \top \rightarrow \neg x \vee \neg p \vee \psi_2\}$的子句的集合$\Sigma$，其中$x$为新引入的原子命题，$\psi_i$（$i=1,2$）为经典子句。除此之外，$\Sigma$中不包含其他含有$p$的公式。

由归结过程可产生子句$\top \rightarrow \neg \vee \psi_1 \vee \psi_2$，由定理4.2可知，$x$可以被$\psi_1 \vee \psi_2$替换。又因为公式$\varphi$中不包含$Pt\mathscr{T}$时序算子，因而不会引入嵌套原子命题（同时出现在$\rightarrow$两边的原子命题），此时对新引入的其余的原子命题都可使用定理4.2。因此，由定理4.1可知CTL-forget$(\varphi,V) \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。 □

已有结果表明，转换过程和归结过程会终止[96]。此外，移除原子命题、移除索引、替换V'中的原子命题和转换到CTL过程都会终止，因此算法4.2会终止。其具体的时间和空间复杂性如下面的结论所示。

命题 4.3. *给定CTL公式φ和原子命题集合$V \subseteq \mathscr{A}$。算法4.2的时间和空间复杂性为$O((m+1)2^{4(n+n')})$，其中$n=|Var(\varphi)|$、$n'=|V'|$为新引入的原子命题的个数、m为引入的索引个数。*

证明. 由于转换过程在多项式时间内完成，移除原子命题、移除索引、转换到CTL过程和替换V'中的原子命题最多都只需要扫描归结过程得到的子句集合就能完成。因此，算法的复杂性主要依赖于归结过程。

对于给定的公式φ和V，归结过程产生的子句个数为$(m+1)2^{4(n+n')} + (m*(n+n') + n + n' + 1)2^{2(n+n')+1})$。 □

在上述结论中值得注意的是m的大小不会大于公式φ中出现的时序算子的个数，因此可以得出算法4.2的计算复杂性仅与出现在φ中的原子命题和时序算子的个数相关。

4.5 本章小结

本章探索了如何使用Zhang等人提出的归结系统计算CTL下的遗忘。为此，首先使用Zhang提出的转换规则将CTL公式转换成$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合，这一步骤在第二章讲述CTL的标准形式时已经给出。基于此，使用归结规则在要遗忘的原子命题集合上做穷尽的归结。然后提出几个互模拟等价的消除索引的等式消除索引，并提出一般化的Ackermann引理用于移除一些新引入的原子命题。最后，给出了计算遗忘的算法，并分析了算法的复杂性，表明了使用该算法计算遗忘的时间和空间复杂性关于公式和要遗忘的原子命题的个数是指数时间的。

第五章　基于模型的方法计算CTL下的遗忘

遗忘与均匀插值是一对对偶概念，已有研究表明*CTL*不具有均匀插值性质[79]，这就表明*CTL*中的遗忘理论不是封闭的[1]。此时，探索*CTL*下遗忘理论封闭的情形对深入应用遗忘理论有重要意义。为此，本章首先提出有限初始结构的特征公式；其次，表明*CTL*公式的遗忘结果在此情形下可以表示成其模型的特征公式的吸取；最后，提出一种基于模型的方法计算遗忘。

5.1　引言

这一段在基础知识部分CTL那节。 计算树逻辑是由Clarke和Emerson提出的一种分支时间时序逻辑，它能很好的描述并发系统的一些性质，包括：互斥属性和安全属性等。Emerson和Halpern证明CTL具有小模型属性：如果一个公式是可满足的，那么它在一个小的有限模型下是可满足的[115]。具体说来，对于给定的CTL公式φ，如果公式的长度[2]为n（记为：$|\varphi| = n$），则存在一个状态数为$n8^n$的初始结构$(\mathscr{M}, s_0)$使得$(\mathscr{M}, s_0) \models \varphi$。

此外，现实情况下能处理的系统都是有限的，且在某一固定环境下所涉及到的原子命题是有限的。因此，在这部分讨论一种约束的CTL，即：（1）出现在CTL公式中的原子命题的个数是有限的（即$\mathscr{A}$是有限的）；（2）初始结构的状态空间S是一个有限的固定状态空间$\mathscr{S} = \{b_1, \ldots, b_m\}$的子集（即$S \subseteq \mathscr{S}$），且使得对于任意约束长度的CTL公式$\varphi$，若$\varphi$是可满足的，则存在一个初始结构$(\mathscr{M}, s_0)$使得$(\mathscr{M}, s_0) \models \varphi$且其状态空间是$\mathscr{S}$的子集。由此可见，在这种情形下只有有限个初始结构应该被考虑。下文将表明在这一约束条件下CTL中的遗忘是封闭的。

本章其余部分组织如下：首先，第5.2节定义一种有界V-互模拟的概念，然后给出初始结构的特征公式。其次，第5.3节描述约束CTL下的遗忘是封闭的，并给出计算方法。进一步，第5.4节给出基于模型的计算遗忘的算法，并分析算法的复杂性。最后，进行本章工作总结。

5.2　描述初始结构

本节介绍与一个初始结构相关的CTL公式——特征公式是如何得到的。对于一个给定的初始结构，其特征公式与其计算树的特征公式密切相关。为此，本节首先介绍

[1]对于给定的逻辑语言$\mathscr{L}$和该语言上的操作$\mathscr{O}$，若$\mathscr{O}$作用到$\mathscr{L}$中的元素后得到的结果仍然在$\mathscr{L}$中，则称$\mathscr{O}$在$\mathscr{L}$下是封闭的。

[2]给定公式φ，出现在该公式里的符号的个数为公式的长度，记为$|\varphi|$。

计算树之间的V-互模拟关系，然后给出计算树的特征公式的定义，最后给出初始结构的特征公式。

5.2.1 计算树V-互模拟

在第三章中的定义3.1给出了Ind-结构（$(\mathscr{M},s_0)$中的$\mathscr{M}$为初始Ind-Kripke结构）之间的V-互模拟定义及其相关性质，结构（$(\mathscr{M},s_0)$中的$\mathscr{M}$为初始Kripke结构）之间的V-互模拟有相同的定义。本节给出结构之间的V-互模拟在有限结构下与下面将要给出的有界V-互模拟——与深度n关联的V-互模拟等价。但是为了方便，本章仍然引入有界V-互模拟的概念。且不难证明Ind-结构之间V-互模拟有的性质结构之间V-互模拟也有，从而第三章遗忘有的性质在本章的约束情形下基本也有[116]，这里不再赘述。

首先给出能够描述一定深度$n\in\mathbb{N}$的计算树之间的V-互模拟关系，记为$\mathscr{B}_n^V$。令$V\subseteq\mathscr{A}$是原子命题的集合，$i\in\{1,2\}$，$\mathscr{M}_i=(S_i,R_i,L_i,s_0^i)$是初始Kripke结构，$\mathscr{K}_i=(\mathscr{M}_i,s_i)$是结构。$\mathscr{B}_n^V$被递归定义如下：

- 若$L_1(s_1)-V=L_2(s_2)$，则$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_0^V$；
- 对任意$n\geq 0$，若满足下面几个条件，则$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_{n+1}^V$成立：
  - $(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_0^V$；
  - 对任意$(s_1,s_1')\in R_1$，存在$(s_2,s_2')\in R_2$使得$(\mathscr{K}_1',\mathscr{K}_2')\in\mathscr{B}_n^V$；
  - 对任意$(s_2,s_2')\in R_2$，存在$(s_1,s_1')\in R_1$使得$(\mathscr{K}_1',\mathscr{K}_2')\in\mathscr{B}_n^V$。

其中$\mathscr{K}_i'=(\mathscr{M}_i,s_i')$。

当所谈及的原子命题的集合V很显然的时候，上述$\mathscr{B}_n^V$中的V可以省略，记为$\mathscr{B}_n$。此外，当讨论的$\mathscr{M}_i\ (i=1,2)$是显然的时候，可以使用$(s_1,s_2)\in\mathscr{B}_n$代替$((\mathscr{M}_1,s_1),(\mathscr{M}_2,s_2))\in\mathscr{B}_n$。此时，有界$V$-互模拟关系就可以定义如下：

定义 5.1 (有界V-互模拟)**.** 令V是$\mathscr{A}$的一个子集，$i\in\{1,2\}$，$\mathscr{K}_1$和$\mathscr{K}_2$是结构。

- $\mathscr{K}_1$和$\mathscr{K}_2$是有界V-互模拟的，当且仅当对所有的$n\geq 0$都有$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_n$。若$\mathscr{K}_1$和$\mathscr{K}_2$是有界$V$-互模拟的，则记为$\mathscr{K}_1\overset{B}{\leftrightarrow}_V\mathscr{K}_2$。
- 对$\mathscr{M}_i$上的路径$\pi_i=(s_{i,1},s_{i,2},\dots)$，若对于任意的$j\in\mathbb{N}_{\geq 1}$[3]都有$\mathscr{K}_{1,j}\overset{B}{\leftrightarrow}_V\mathscr{K}_{2,j}$，则$\pi_1\overset{B}{\leftrightarrow}_V\pi_2$。其中$\mathscr{K}_{i,j}=(\mathscr{M}_i,s_{i,j})$。

值得注意的是，满足有界V-互模拟关系的结构之间有且仅有一个有界V-互模拟关系，即$\mathscr{B}_{k+1}=\mathscr{B}_k$（$k\in\mathbb{N}$）。

[3]$\mathbb{N}$为整数的集合，$\mathbb{N}_{\geq 1}$是大于等于1的整数的集合。

引理 5.1. *对于有限初始Kripke结构，有界V-互模拟是一个V-互模拟。*

证明. 令$V \subseteq \mathscr{A}$、$\mathscr{K}_i = (\mathscr{M}_i, s_i)$ $(i \in \{1,2\})$和$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$为有限的初始-Kripke结构。因为$S_i$ $(i = 1,2)$是有限的，所以$\mathscr{B}_n \subseteq S_1 \times S_2$ $(n \geq 0)$都是有限的。由于对任意的$n \geq 0$都有$\mathscr{B}_{n+1} \subseteq \mathscr{B}_n$。因此，存在一个最小数$k$使得$\mathscr{B}_{k+1} = \mathscr{B}_k$（用$\mathscr{B}$表示）。

这里证明对任意的$r_1 \in S_1$和$r_2 \in S_2$，有$(r_1, r_2) \in \mathscr{B}$蕴涵

(a) $L_1(r_1) - V = L_2(r_2) - V$；

(b) $\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$和$(w_1, w_2) \in \mathscr{B}$；和

(c) $\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$和$(w_1, w_2) \in \mathscr{B}$。

首先，由对任意的$n \geq 0$都有$(r_1, r_2) \in \mathscr{B}_n$可知$(r_1, r_2) \in \mathscr{B}_0$，这意味着$L_1(r_1) - V = L_2(r_2) - V$。因此(a)成立。

其次，令$w_1 \in S_1$和$(r_1, w_1) \in R_1$。有
$(r_1, r_2) \in \mathscr{B}$、$w_1 \in S_1$和$(r_1, w_1) \in R_1$
$\Rightarrow$ 由$\mathscr{B}_{k+1} = \mathscr{B}$知$(r_1, r_2) \in \mathscr{B}_{k+1}$、$w_1 \in S_1$和$(r_1, w_1) \in R_1$
$\Rightarrow$ 由定义可知$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$和$(w_1, w_2) \in \mathscr{B}_k$
$\Rightarrow$ 由$\mathscr{B} = \mathscr{B}_k$可知$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$和$(w_1, w_2) \in \mathscr{B}$
$\Rightarrow$ (b)成立。

可以类似地证明(c)也成立。因此，$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_2$之间的一个V-互模拟关系。　□

由引理5.1可知，$\mathscr{B}$既是一个有界V-互模拟关系也是一个V-互模拟关系，因此下面的推论是显然成立的。

推论 5.1. *令$V \subseteq \mathscr{A}$ 和$\mathscr{K}_i = (\mathscr{M}_i, s_i)$ $(i \in \{1,2\})$，其中$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$是有限的初始Kripke结构。则$s_1$和$s_2$是有界$V$-互模拟的当且仅当$s_1 \leftrightarrow_V s_2$。*

本章只涉及有限的初始Kripke结构，因此，本章用$\leftrightarrow_V$（V-互模拟）表示$\overset{B}{\leftrightarrow}_V$（有界$V$-互模拟）。

给定原子命题集合$V \subseteq \mathscr{A}$和初始Kripke结构$\mathscr{M}_i$（$i = 1,2$）。如果下面条件同时被满足，则称$\mathscr{M}_1$的计算树$\mathrm{Tr}_n(s_1)$和$\mathscr{M}_2$的计算树$\mathrm{Tr}_n(s_2)$是V-互模拟的（记为$(\mathscr{M}_1, \mathrm{Tr}_n(s_1)) \leftrightarrow_V (\mathscr{M}_2, \mathrm{Tr}_n(s_2))$，简写为$\mathrm{Tr}_n(s_1) \leftrightarrow_V \mathrm{Tr}_n(s_2)$）：

- $L_1(s_1) - V = L_2(s_2) - V$,
- 对$\mathrm{Tr}_n(s_1)$的任意子树$\mathrm{Tr}_{n-1}(s_1')$，都存在$\mathrm{Tr}_n(s_2)$的一棵子树$\mathrm{Tr}_{n-1}(s_2')$使得$\mathrm{Tr}_{n-1}(s_1') \leftrightarrow_V \mathrm{Tr}_{n-1}(s_2')$，且

- 对任意$\mathrm{Tr}_n(s_2)$的子树$\mathrm{Tr}_{n-1}(s_2')$，都存在$\mathrm{Tr}_n(s_1)$的一棵子树$\mathrm{Tr}_{n-1}(s_1')$使得$\mathrm{Tr}_{n-1}(s_1') \leftrightarrow_V \mathrm{Tr}_{n-1}(s_2')$。

在上述定义中，当$n=0$时，只需考虑第一个条件。

命题 5.1. *给定原子命题集合$V \subseteq \mathscr{A}$和结构$(\mathscr{M}_i, s_i)$（$i=1,2$）。则：*

$$(s_1,s_2) \in \mathscr{B}_n \text{当且仅当对任意的} 0 \leq j \leq n \text{有} Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)。$$

证明. 这里从下面两个方面来证明这一结论：

($\Rightarrow$) 这里证明“如果$(s_1,s_2) \in \mathscr{B}_n$，则对于任意的$0 \leq j \leq n$有$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$”成立。$(s_1,s_2) \in \mathscr{B}_n$意味着$Tr_n(s_1)$和$Tr_n(s_2)$的根有同样的标签（除了$V$里的元素之外）。此外，对任意的$(s_1,s_{1,1}) \in R_1$，存在一个$(s_2,s_{2,1}) \in R_2$使得$(s_{1,1},s_{2,1}) \in \mathscr{B}_{n-1}$；且对任意的$(s_2,s_{2,1}) \in R_2$，存在一个$(s_1,s_{1,1}) \in R_1$使得$(s_{1,1},s_{2,1}) \in \mathscr{B}_{n-1}$。因此，由定义可知$Tr_1(s_1) \leftrightarrow_V Tr_1(s_2)$。递归地使用上述方法可得对任意的$0 \leq j \leq n$都有$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$。

($\Leftarrow$)这里证明“如果对于任意的$0 \leq j \leq n$有$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$，则$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$”成立。由$Tr_0(s_1) \leftrightarrow_V Tr_0(s_2)$可知$L(s_1)-V = L'(s_2)-V$，因而$(s_1,s_2) \in \mathscr{B}_0$。由$Tr_1(s_1) \leftrightarrow_V Tr_1(s_2)$可知$L(s_1)-V = L'(s_2)-V$，且对于一棵树根的任意后继状态$s$，都能找到另一棵树根的一个后继状态$s'$使得$(s,s') \in \mathscr{B}_0$。因此有$(s_1,s_2) \in \mathscr{B}_1$。同理可证$(s_1,s_2) \in \mathscr{B}_2, \ldots,$ $(s_1,s_2) \in \mathscr{B}_n$。 □

命题5.1表明如果任意两个初始结构中的两个状态s_1和s_2能够在$\overline{V}$上相互模拟对方直到n步，当且仅当分别以s_1和s_2为根的计算树能在$\overline{V}$上相互模拟直到深度为n。由此可知，如果同一初始结构的两个状态s和s'不是V-互模拟的，则存在一个数$k \in \mathbb{N}$使得分别以s和s'为根的计算树$\mathrm{Tr}_k(s)$和$\mathrm{Tr}_k(s')$不是V-互模拟的。

命题 5.2. *给定原子命题集合$V \subseteq \mathscr{A}$、初始Kripke结构$\mathscr{M}$和两个状态$s,s' \in S$。若$s \not\leftrightarrow_V s'$，则存在一个最小整数$k$使得$Tr_k(s)$和$Tr_k(s')$不是$V$-互模拟的。*

证明. 若$s \not\leftrightarrow_V s'$，则存在一个最小的数$c$使得$(s_i,s_j) \notin \mathscr{B}_c$。因此，由命题5.1可知，存在一个最小整数$m$（$m \leq c$）使得$\mathrm{Tr}_m(s_i)$和$\mathrm{Tr}_m(s_j)$不是$V$-互模拟的。令$k=m$可得上述结论。 □

5.2.2 计算树的特征公式

由上面小节的讨论可知，V-互模拟可以将计算树分别开来[4]。本节讨论如何使用CTL公式描述一棵计算树，且表明具有（或没有）V-互模拟关系之间的计算树的特征公式又有怎样的关系。为此，首先给出计算树的特征公式的定义。

[4]相似的方法在Mycielski等人的文章中已被使用[117]，在这篇文章中一元公式的结构通过等价类$\equiv_{\bar{k}}$被描述为Hintikka公式[118]。另一个类似的工作是Yankov-Fine构造[119]。

定义 5.2. 给定原子命题集合$V \subseteq \mathscr{A}$、初始*Kripke*结构$\mathscr{M} = (S, R, L, s_0)$和状态$s \in S$。定义在$V$上的计算树$Tr_n(s)$的特征公式（记为$\mathscr{F}_V(Tr_n(s))$，$n \geq 0$）被递归定义如下：

$$\mathscr{F}_V(Tr_0(s)) = \bigwedge_{p \in V \cap L(s)} p \wedge \bigwedge_{q \in V - L(s)} \neg q,$$

$$\mathscr{F}_V(Tr_{k+1}(s)) = \bigwedge_{(s,s') \in R} \mathrm{EX}\mathscr{F}_V(Tr_k(s')) \wedge \mathrm{AX}\left(\bigvee_{(s,s') \in R} \mathscr{F}_V(Tr_k(s')) \right) \wedge \mathscr{F}_V(Tr_0(s)) \quad (k \geq 0)。$$

由定义5.2可知，计算树的特征公式从三个方面展示了计算树的信息：（1）只考虑V中的原子命题；（2）突出了树节点的内容，即：对于任意原子命题$p \in V$，若p在节点的标签中，则其正出现在特征公式中，否则负出现在特征公式中；（3）公式中的时序算子表示了状态之间的转换关系。通俗地讲，$\mathscr{F}_V(\mathrm{Tr}_0(s))$表明了节点$s$的在$V$上的内容；EX的合取部分和AX部分保证以$s$的每个直接后继状态$s'$为根深度为$k$的计算树都有一个CTL公式来描述。

下面的结论表明，若两个计算树是V-互模拟的，则他们在V上的特征公式是逻辑等价的。

引理 5.2. 给定原子命题集合$V \subseteq \mathscr{A}$、初始*Kripke*结构$\mathscr{M} = (S, R, L, s_0)$和$\mathscr{M}' = (S', R', L', s'_0)$、$s \in S$、$s' \in S'$且$n \geq 0$。若$Tr_n(s) \leftrightarrow_{\overline{V}} Tr_n(s')$，则$\mathscr{F}_V(Tr_n(s)) \equiv \mathscr{F}_V(Tr_n(s'))$。

证明. 通过归纳计算树的深度n来证明。

基始（$n = 0$）：对任意的$s_x \in S$和$s'_x \in S'$，若$\mathrm{Tr}_0(s_x) \leftrightarrow_{\overline{V}} \mathrm{Tr}_0(s'_x)$，则由$L(s_x) - \overline{V} = L'(s'_x) - \overline{V}$可知$\mathscr{F}_V(\mathrm{Tr}_0(s_x)) \equiv \mathscr{F}_V(\mathrm{Tr}_0(s'_x))$。

归纳步（$n > 0$）：假设对任意的$0 \leq m \leq n$若$\mathrm{Tr}_m(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_m(s')$，则$\mathscr{F}_V(\mathrm{Tr}_m(s)) \equiv \mathscr{F}_V(\mathrm{Tr}_m(s'))$。这里要证明若$\mathrm{Tr}_{n+1}(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n+1}(s')$，则$\mathscr{F}_V(\mathrm{Tr}_{n+1}(s)) \equiv \mathscr{F}_V(\mathrm{Tr}_{n+1}(s'))$。

由归纳假设可知，对任意的$k = m$、$s_k \in S$和$s'_k \in S'$，若$\mathrm{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k}(s'_k)$，则$\mathscr{F}_V(\mathrm{Tr}_{n-k}(s_k)) \equiv \mathscr{F}_V(\mathrm{Tr}_{n-k}(s'_k))$。因此，要证原结论成立，只需要证明若$\mathrm{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k+1}(s'_{k-1})$，则$\mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s_{k-1})) \equiv \mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s'_{k-1}))$。其中，$(s_{k-1}, s_k) \in R$且$(s'_{k-1}, s'_k) \in R'$。显然，由计算树的特征公式可知：

$$\mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s_{k-1})) = \left(\bigwedge_{(s_{k-1},s_k) \in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_{n-k}(s_k)) \right) \wedge \mathrm{AX}\left(\bigvee_{(s_{k-1},s_k) \in R} \mathscr{F}_V(\mathrm{Tr}_{n-k}(s_k)) \right) \wedge \mathscr{F}_V(\mathrm{Tr}_0(s_{k-1}))$$

且

$$\mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s'_{k-1})) = \left(\bigwedge_{(s'_{k-1},s'_k)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_{n-k}(s'_k))\right) \wedge \mathrm{AX}\left(\bigvee_{(s'_{k-1},s'_k)\in R} \mathscr{F}_V(\mathrm{Tr}_{n-k}(s'_k))\right) \wedge \mathscr{F}_V(\mathrm{Tr}_0(s'_{k-1})).$$

又因为$\mathrm{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_V \mathrm{Tr}_{n-k+1}(s'_{k-1})$，所以对任意的$(s_{k-1},s_k)\in R$存在$(s'_{k-1},s'_k)\in R'$使得$\mathrm{Tr}_{n-k}(s_k)\leftrightarrow_V \mathrm{Tr}_{n-k}(s'_k)$，且对任意的$(s'_{k-1},s'_k)\in R'$存在$(s_{k-1},s_k)\in R$使得$\mathrm{Tr}_{n-k}(s_k)\leftrightarrow_V \mathrm{Tr}_{n-k}(s'_k)$。因此，由归纳假设可知$\mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s_{k-1}))\equiv\mathscr{F}_V(\mathrm{Tr}_{n-k+1}(s'_{k-1}))$。 □

此外，对于初始Kripke结构$\mathscr{M}$上的状态s和s'，若$(\mathscr{M},s)$是定义在V上的根为s'深度为n的计算树的特征公式，则s和s'至少属于$\mathscr{B}_n$，即：s和s'能想互模拟至少到第n层深度。

引理 5.3. 令$V\subseteq\mathscr{A}$、$\mathscr{M}=(S,R,L,s_0)$、$\mathscr{M}'=(S',R',L',s'_0)$、$s\in S$、$s'\in S'$且$n\geq 0$，则：

(i) $(\mathscr{M},s)\models\mathscr{F}_V(Tr_n(s))$；

(ii) 若$(\mathscr{M},s)\models\mathscr{F}_V(Tr_n(s'))$，则$Tr_n(s)\leftrightarrow_V Tr_n(s')$。

证明. (i) **基始**（$n=0$）：从树的特征公式定义可知$\mathscr{F}_V(\mathrm{Tr}_0(s))$是显然的。

归纳步（$n>0$）：假设对任意的$k\geq 0$，$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_k(s))$，下面证明$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_{k+1}(s))$，即：

$$(\mathscr{M},s)\models\left(\bigwedge_{(s,s')\in R}\mathrm{EX}T(s')\right)\wedge\mathrm{AX}\left(\bigvee_{(s,s')\in R}T(s')\right)\wedge\mathscr{F}_V(\mathrm{Tr}_0(s)).$$

其中$T(s')=\mathscr{F}_V(\mathrm{Tr}_k(s'))$。由基始可知$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_0(s))$。由归纳假设可知，对任意的$(s,s')\in R$有$(\mathscr{M},s')\models\mathscr{F}_V(\mathrm{Tr}_k(s'))$。因此有$(\mathscr{M},s)\models\mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_k(s')$，从而$(\mathscr{M},s)\models\bigwedge_{(s,s')\in R}\mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_k(s'))$。

同理，对任意的$(s,s')\in R$都有$(\mathscr{M},s')\models\bigvee_{(s,s')\in R}\mathscr{F}_V(\mathrm{Tr}_k(s'))$。因此，

$$(\mathscr{M},s)\models\mathrm{AX}\left(\bigvee_{(s,s')\in R}\mathscr{F}_V(\mathrm{Tr}_k(s'))\right)$$

从而可知对任意的$n\geq 0$，$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s))$。

(ii) **基始**（$n=0$）：若$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_0(s'))$，则$L(s)-\overline{V}=L'(s')-\overline{V}$。因此$\mathrm{Tr}_0(s)\leftrightarrow_{\overline{V}}\mathrm{Tr}_0(s')$。

归纳步（$n>0$）：假定若$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'))$，则$\mathrm{Tr}_{n-1}(s)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s')$。下面证明若$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s'))$，则$\mathrm{Tr}_n(s)\leftrightarrow_{\overline{V}}\mathrm{Tr}_n(s')$。

(a) 由基始知$L(s)-\overline{V}=L'(s')-\overline{V}$；

(b) 因为$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s'))$，所以$(\mathscr{M},s)\models\mathrm{AX}\left(\bigvee_{(s',s'_1)\in R}\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1))\right)$。由此，对于任意的$(s,s_1)\in R$，存在$(s',s'_1)\in R'$使得$(\mathscr{M},s_1)\models\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1))$。由归纳假设可知$\mathrm{Tr}_{n-1}(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s'_1)$。即：$\forall(s,s_1)\in R$，$\exists(s',s'_1)\in R'$使得$\mathrm{Tr}_{n-1}(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s'_1)$。

(c) 因为$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s'))$，所以$(\mathscr{M},s)\models\bigwedge_{(s',s'_1)\in R'}\mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1))$。由此，对于任意的$(s',s'_1)\in R'$，存在$(s,s_1)\in R$使得$(\mathscr{M},s_1)\models\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1)$。由归纳假设可知$\mathrm{Tr}_{n-1}(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s'_1)$。即：$\forall(s',s'_1)\in R'$，$\exists(s,s_1)\in R$使得$\mathrm{Tr}_{n-1}(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s'_1)$。

□

5.2.3　初始结构的特征公式

由V-互模拟的定义和命题5.2可以自然地得到一个V-互模拟的补概念——V-可区分。特别地，在命题5.2中，若初始Kripke结构$\mathscr{M}$的两个状态s和s'不是$\overline{V}$-互模拟的（即：$s\not\leftrightarrow_{\overline{V}}s'$），则称$s$和$s'$是$V$-可区分的。且用$\mathrm{dis}_V(\mathscr{M},s,s',k)$表示状态$s$和$s'$在命题5.2中所说的最小数$k$下是$V$-可区分的。正如下文所说，$V$-可区分这一概念是定义初始结构特征公式的重要概念。

此外，对于给定的初始Kripke结构$\mathscr{M}$和原子命题集合V，若在$\mathscr{M}$中存在两个状态s和s'是V-可区分的，则称$\mathscr{M}$是V-可区分的。而对于一个V-可区分的初始Kripke结构$\mathscr{M}$，存在一个一个最小的数k使得对于该结构上的任意两个状态s和s'，若s和s'是可区分的，则$(s,s')\notin\mathscr{B}_k$。本文称这样的数为$\mathscr{M}$关于$V$的特征数，记为$ch(\mathscr{M},V)$，其定义如下：

$$ch(\mathscr{M},V)=\begin{cases}\max\{k\mid s,s'\in S\text{ 且}\mathrm{dis}_V(\mathscr{M},s,s',k)\}, & \mathscr{M}\text{是}V\text{-可区分的；}\\ \min\{k\mid\mathscr{B}_k=\mathscr{B}_{k+1},k\geq 0\}, & \text{否则。}\end{cases}$$

由$ch(\mathscr{M},V)$定义可知，对于任意的$\mathscr{M}$和V，$ch(\mathscr{M},V)$总是存在的，这体现在两个方面：（1）若$\mathscr{M}$是V-可区分的，则存在两个状态s和s'是V-可区分的，由命题5.2可知，

存在一个数k使得$\mathrm{dis}_V(\mathscr{M},s,s',k)$成立；（2）若对于任意$k\geq 0$和$\mathscr{M}$上的两个状态$s$和$s'$都有$(s,s')\in\mathscr{B}_k$且$\mathscr{B}_k=\mathscr{B}_{k+1}$，则$ch(\mathscr{M},V)=0$。

非形式化地说，特征数$c=ch(\mathscr{M},V)$将$\mathscr{M}$上的状态分为两大类：第一类中的任意两个状态s和s'是V-可区分的，且$(s,s')\notin\mathscr{B}_c$；第二类中状态都是$V$-不可区分的。这也在计算树的特征公式上：

引理 5.4. 令$V\subseteq\mathscr{A}$、$\mathscr{M}=(S,R,L,s_0)$、$k=ch(\mathscr{M},V)$且$s\in S$，则

(i) $(\mathscr{M},s)\models\mathscr{F}_V(Tr_k(s))$；

(ii) 对任意的$s'\in S$，$(\mathscr{M},s)\leftrightarrow_{\overline{V}}(\mathscr{M},s')$当且仅当$(\mathscr{M},s')\models\mathscr{F}_V(Tr_k(s))$。

证明. (i) 这由引理5.3易知。

(ii) 令$\phi=\mathscr{F}_V(\mathrm{Tr}_k(s))$（$k$为$\mathscr{M}$关于$V$的特征数）。由(i)知$(\mathscr{M},s)\models\phi$，从而对任意的$s'\in S$，若$s\leftrightarrow_{\overline{V}}s'$，由定理3.1和$\mathrm{IR}(\phi,\mathscr{A}-V)$知$(\mathscr{M},s')\models\phi$。

假定$(\mathscr{M},s')\models\phi$。若$s\nleftrightarrow_{\overline{V}}s'$，则$\mathrm{Tr}_k(s)\not\leftrightarrow_{\overline{V}}\mathrm{Tr}_k(s')$，因而由引理5.3可知$(\mathscr{M},s')\not\models\phi$，这与假定矛盾。 □

由此，可定义初始结构的特征公式如下。

定义 5.3 (特征公式). 给定原子命题集合$V\subseteq\mathscr{A}$和初始结构$\mathscr{K}=(\mathscr{M},s_0)$，其中$c=ch(\mathscr{M},V)$。对任意$\mathscr{M}$上得状态$s'\in S$，记$T(s')=\mathscr{F}_V(Tr_c(s'))$。则$\mathscr{K}$关于$V$的特征公式$\mathscr{F}_V(\mathscr{K})$为：

$$T(s_0)\wedge\bigwedge_{s\in S}\mathrm{AG}\left(T(s)\rightarrow\bigwedge_{(s,s')\in R}\mathrm{EX}T(s')\wedge\mathrm{AX}\left(\bigvee_{(s,s')\in R}T(s')\right)\right).$$

有时为了凸显出初始Kripke结构及其初始状态，也把特征公式写为$\mathscr{F}_V(\mathscr{M},s_0)$。显然，$\mathrm{IR}(\mathscr{F}_V(\mathscr{M},s_0),\overline{V})$。此外，在特征公式的定义中，使用了深度为$c$（即：特征数）的计算树的特征公式意在表明对任意$\mathscr{M}$上的两个状态$s$和$s'$，$s$和$s'$是$V$-可区分的当且仅当$\mathscr{F}_V(\mathrm{Tr}_c(s))\not\equiv\mathscr{F}_V(\mathrm{Tr}_c(s'))$。特别地，$T(s_0)$确保了初始结构的初始状态被CTL公式描述；其余部分表明了结构$\mathscr{M}$上状态之间的转换关系。下面的例子给出了计算特征公式的一般步骤：

例 5.1 (例3.1的延续). 考虑图5.1中左边的初始结构$\mathscr{K}_2=(\mathscr{M},s_0)$（其最初出现在图3.1中）。左边的为$\mathscr{M}$上的四棵计算树：从左到右表示以$s_0$为根、深度分别为0、1、2和3的计算树（为简化图，计算树的标签没有给出，但是每个树节点的标签可从$\mathscr{K}_2$找到）。令$V=\{d\}$，则$\overline{V}=\{s,se\}$。

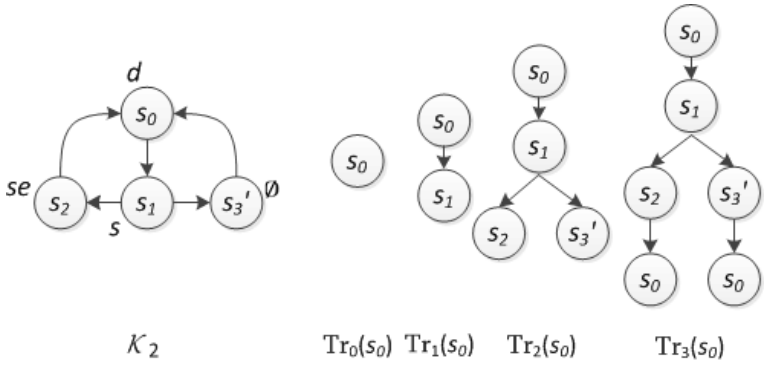


图 5.1: 初始结构$\mathscr{K}_2$（源于图3.1）及其计算树示意图

因为$L(s_1)-\overline{V}=L(s_2)-\overline{V}$，所以有$\mathrm{Tr}_0(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_0(s_2)$。由于存在$(s_1,s_2)\in R$使得对任意的$(s_2,s')\in R$都有$L(s_2)-\overline{V}\neq L(s')-\overline{V}$，所以$\mathrm{Tr}_1(s_1)\not\leftrightarrow_{\overline{V}}\mathrm{Tr}_1(s_2)$。由此可知$s_1$和$s_2$是$V$-可区分的，且$\mathrm{dis}_V(\mathscr{M},s_1,s_2,1)$。

同样，我们可得到：$\mathrm{dis}_V(\mathscr{M},s_0,s_1,0)$、$\mathrm{dis}_V(\mathscr{M},s_1,s_3',1)$、$\mathrm{dis}_V(\mathscr{M},s_0,s_2,0)$和$\mathrm{dis}_V(\mathscr{M},s_0,s_3',0)$。此外，$s_2\leftrightarrow_{\overline{V}}s_3'$。因此可以计算$\mathscr{M}$关于$V$的特征数为：

$$ch(\mathscr{M},V)=\max\{k\mid s,s'\in S\text{ 且}\mathrm{dis}_V(\mathscr{M},s,s',k)\}=1.$$

所以，可以由以下步骤计算$\mathscr{K}_2$关于V的特征公式：

$$\begin{aligned}
&\mathscr{F}_V(\mathrm{Tr}_0(s_0))=d, \qquad \mathscr{F}_V(\mathrm{Tr}_0(s_1))=\neg d,\\
&\mathscr{F}_V(\mathrm{Tr}_0(s_2))=\neg d, \qquad \mathscr{F}_V(\mathrm{Tr}_0(s_3'))=\neg d,\\
&\mathscr{F}_V(\mathrm{Tr}_1(s_0))=\mathrm{EX}\neg d\wedge\mathrm{AX}\neg d\wedge d\equiv\mathrm{AX}\neg d\wedge d,\\
&\mathscr{F}_V(\mathrm{Tr}_1(s_1))=\mathrm{EX}\neg d\wedge\mathrm{EX}\neg d\wedge\mathrm{AX}(\neg d\vee\neg d)\wedge\neg d\equiv\mathrm{AX}\neg d\wedge\neg d,\\
&\mathscr{F}_V(\mathrm{Tr}_1(s_2))=\mathrm{EX}d\wedge\mathrm{AX}d\wedge\neg d\equiv\mathrm{AX}d\wedge\neg d,\\
&\mathscr{F}_V(\mathrm{Tr}_1(s_3'))\equiv\mathscr{F}_V(\mathrm{Tr}_1(s_2)),\\
&\mathscr{F}_V(\mathscr{M},s_0)\equiv\mathrm{AX}\neg d\wedge d\wedge\\
&\qquad\mathrm{AG}(\mathrm{AX}\neg d\wedge d\rightarrow\mathrm{AX}(\mathrm{AX}\neg d\wedge\neg d))\wedge\\
&\qquad\mathrm{AG}(\mathrm{AX}\neg d\wedge\neg d\rightarrow\mathrm{AX}(\mathrm{AX}d\wedge\neg d))\wedge\\
&\qquad\mathrm{AG}(\mathrm{AX}d\wedge\neg d\rightarrow\mathrm{AX}(\mathrm{AX}\neg d\wedge d)).
\end{aligned}$$

下面的定理表示上述定义的特征公式确实描述了一个初始结构，此时对系统结构的操作就可转换为对其特征公式的操作，如下文将要讲的给定系统下的最弱充分条件计算。更直观地说，特征公式保持了给定初始结构在原子命题集合V上的所有特性，即：具有$\overline{V}$-互模拟的两个初始结构关于V的特征公式逻辑等价。

定理 5.1. 令$V \subseteq \mathscr{A}$、$\mathscr{M} = (S,R,L,s_0)$且$\mathscr{M}' = (S',R',L',s_0')$，则：

(i) $(\mathscr{M}',s_0') \models \mathscr{F}_V(\mathscr{M},s_0)$ 当且仅当$(\mathscr{M},s_0) \leftrightarrow_{\overline{V}} (\mathscr{M}',s_0')$；

(ii) 若$s_0 \leftrightarrow_{\overline{V}} s_0'$则$\mathscr{F}_V(\mathscr{M},s_0) \equiv \mathscr{F}_V(\mathscr{M}',s_0')$。

证明. (i) 令$\mathscr{F}_V(\mathscr{M},s_0)$为$(\mathscr{M},s_0)$关于$V$的特征公式。显然，IR$(\mathscr{F}_V(\mathscr{M},s_0),\overline{V})$。为了证明上述结论成立，下面证明首先证明$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathscr{M},s_0)$。

令$c = ch(\mathscr{M},V)$，由引理5.3可知$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s_0))$。下面证明特征公式里的另一部分，即：$(\mathscr{M},s_0) \models \bigwedge_{s\in S} \mathrm{AG}\, G(\mathscr{M},s)$，其中

$$G(\mathscr{M},s) = \mathscr{F}_V(\mathrm{Tr}_c(s)) \rightarrow \left(\bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1)) \right) \wedge \mathrm{AX} \left(\bigvee_{(s,s_1)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_1)) \right).$$

为此，下面证明$(\mathscr{M},s_0) \models \mathrm{AG}\, G(\mathscr{M},s)$。考虑下面两种情况：

- 若$(\mathscr{M},s_0) \not\models \mathscr{F}_V(\mathrm{Tr}_c(s))$，显然$(\mathscr{M},s_0) \models G(\mathscr{M},s)$；
- 若$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s))$：
  $(\mathscr{M},s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s))$
  $\Rightarrow s_0 \leftrightarrow_{\overline{V}} s$ （引理5.4）
  $\forall (s,s_1) \in R$：
  $(\mathscr{M},s_1) \models \mathscr{F}_V(\mathrm{Tr}_c(s_1))$ （$s_1 \leftrightarrow_{\overline{V}} s_1$）
  $\Rightarrow (\mathscr{M},s) \models \bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))$
  $\Rightarrow (\mathscr{M},s_0) \models \bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))$ （IR$(\bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1)),\overline{V})$, $s_0 \leftrightarrow_{\overline{V}} s$）
  $\forall (s,s_1) \in R$：
  $(\mathscr{M},s_1) \models \bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))$
  $\Rightarrow (\mathscr{M},s) \models \mathrm{AX}\left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))\right)$
  $\Rightarrow (\mathscr{M},s_0) \models \mathrm{AX}\left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))\right)$ （IR$(\mathrm{AX}\left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))\right),\overline{V})$, $s_0 \leftrightarrow_{\overline{V}} s$）
  $\Rightarrow (\mathscr{M},s_0) \models G(\mathscr{M},s)$。

对任意其他能从s_0可达的状态s'，都可以类似地证明$(\mathscr{M},s') \models G(\mathscr{M},s)$。因此，对任意的$s \in S$，$(\mathscr{M},s_0) \models \mathrm{AG}\, G(\mathscr{M},s)$，从而$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathscr{M},s_0)$。

下面从两个方面证明(i)成立：

($\Leftarrow$) 证明：若$s_0 \leftrightarrow_{\overline{V}} s_0'$，则$(\mathscr{M}',s_0') \models \mathscr{F}_V(\mathscr{M},s_0)$。因为$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathscr{M},s_0)$且IR$(\mathscr{F}_V(\mathscr{M},s_0),\overline{V})$，由定理3.1可知$(\mathscr{M}',s_0') \models \mathscr{F}_V(\mathscr{M},s_0)$。

($\Rightarrow$) 证明：若$(\mathscr{M}', s_0') \models \mathscr{F}_V(\mathscr{M}, s_0)$，则$s_0 \leftrightarrow_{\overline{V}} s_0'$。为此，下面证明对任意的$n \geq 0$，$Tr_n(s_0) \leftrightarrow_{\overline{V}} Tr_n(s_0')$。

基始（$n = 0$）：由特征公式的定义，显然$Tr_0(s_0) \equiv Tr_0(s_0')$成立。

归纳步骤（$n > 0$）：假定对任意的$k > 0$都有$\mathrm{Tr}_k(s_0) \leftrightarrow_{\overline{V}} \mathrm{Tr}_k(s_0')$，下面证明$\mathrm{Tr}_{k+1}(s_0) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{k+1}(s_0')$。令$(s_0, s_1), (s_1, s_2), \ldots, (s_{k-1}, s_k) \in R$且$(s_0', s_1'), (s_1', s_2'), \ldots, (s_{k-1}', s_k') \in R'$，即对于任意的$0 \leq i \leq k-1$，$s_{i+1}$（$s_{i+1}'$）是$s_i$（$s_i'$）的直接后继状态。由归纳假设可知，只需证明$\mathrm{Tr}_1(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_1(s_k')$。

(a) 由归纳假设可知$L(s_k) - \overline{V} = L'(s_k') - \overline{V}$。

在讨论其他点时，首先考虑下面事实（**fact**）：

$(\mathscr{M}', s_0') \models \mathscr{F}_V(\mathscr{M}, s_0)$
$\Rightarrow \forall s' \in S'$，$\forall s \in S$，
$(\mathscr{M}', s') \models \mathscr{F}_V(\mathrm{Tr}_c(s)) \rightarrow \left(\bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))\right) \wedge \mathrm{AX}\left(\bigvee_{(s,s_1)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_1))\right)$
(I) $(\mathscr{M}', s_0') \models \mathscr{F}_V(\mathrm{Tr}_c(s_0)) \rightarrow \left(\bigwedge_{(s_0,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))\right) \wedge \mathrm{AX}\left(\bigvee_{(s_0,s_1)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_1))\right)$
(II) $(\mathscr{M}', s_0') \models \mathscr{F}_V(\mathrm{Tr}_c(s_0))$　(已知)
(III) $(\mathscr{M}', s_0') \models \left(\bigwedge_{(s_0,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))\right) \wedge \mathrm{AX}\left(\bigvee_{(s_0,s_1)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_1))\right)$　((I),(II))

(b) 这里证明$\forall (s_k, s_{k+1}) \in R$，存在$(s_k', s_{k+1}') \in R'$使得$L(s_{k+1}) - \overline{V} = L'(s_{k+1}') - \overline{V}$。
(1) $(\mathscr{M}', s_0') \models \bigwedge_{(s_0,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))$　(III)
(2) $\forall (s_0, s_1) \in R$，$\exists (s_0', s_1') \in R'$使得$(\mathscr{M}', s_1') \models \mathscr{F}_V(\mathrm{Tr}_c(s_1))$　(1)
(3) $\mathrm{Tr}_c(s_1) \leftrightarrow_{\overline{V}} \mathrm{Tr}_c(s_1')$　((2), 引理5.3)
(4) $L(s_1) - \overline{V} = L'(s_1') - \overline{V}$　((3), $c \geq 0$)
(5) $(\mathscr{M}', s_1') \models \mathscr{F}_V(\mathrm{Tr}_c(s_1)) \rightarrow \left(\bigwedge_{(s_1,s_2)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_2))\right) \wedge \mathrm{AX}\left(\bigvee_{(s_1,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))\right)$ **(fact)**
(6) $(\mathscr{M}', s_1') \models \left(\bigwedge_{(s_1,s_2)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_2))\right) \wedge \mathrm{AX}\left(\bigvee_{(s_1,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))\right)$　((2), (5))
(7) $(\mathscr{M}', s_k') \models \left(\bigwedge_{(s_k,s_{k+1})\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}))\right) \wedge \mathrm{AX}\left(\bigvee_{(s_k,s_{k+1})\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}))\right)$　(与(6)类似)
(8) $\forall (s_k, s_{k+1}) \in R$，$\exists (s_k', s_{k+1}') \in R'$使得$(\mathscr{M}', s_{k+1}') \models \mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}))$　(7)
(9) $\mathrm{Tr}_c(s_{k+1}) \leftrightarrow_{\overline{V}} \mathrm{Tr}_c(s_{k+1}')$　((8), 引理5.3)
(10) $L(s_{k+1}) - \overline{V} = L'(s_{k+1}') - \overline{V}$　((9), $c \geq 0$)

(c) 这里证明$\forall (s_k', s_{k+1}') \in R'$，存在$(s_k, s_{k+1}) \in R$使得$L(s_{k+1}) - \overline{V} = L'(s_{k+1}') - \overline{V}$。
(1) $(\mathscr{M}', s_k') \models \mathrm{AX}\left(\bigvee_{(s_k,s_{k+1})\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}))\right)$　(上面的(7))
(2) $\forall (s_k', s_{k+1}') \in R'$，$\exists (s_k, s_{k+1}) \in R$使得$(\mathscr{M}', s_{k+1}') \models \mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}'))$　(1)
(3) $\mathrm{Tr}_c(s_{k+1}) \leftrightarrow_{\overline{V}} \mathrm{Tr}_c(s_{k+1}')$　((2), 引理5.3)
(4) $L(s_{k+1}) - \overline{V} = L'(s_{k+1}') - \overline{V}$　((3), $c \geq 0$)

(ii) 由引理5.2和5.4易知。

□

5.3 遗忘封闭性及复杂性

当给定的CTL公式的长度（符号的个数）为n，由小模型理论可知定义在状态个数为$k = n8^n$的状态空间$\mathscr{S} = \{s_1, s_2, \ldots, s_k\}$上的初始Kripke结构就能保证公式的可满足性[115]。对于其他拥有同样大小的状态空间上的任意初始结构，都能在$\mathscr{S}$状态空间上找到一个初始结构与之互模拟，且由定理5.1可知他们有相同的特征公式。因此，只有有限个初始结构作为该公式的候选模型。从而下面结论成立。

引理 5.5. *给定CTL公式φ，下面等式成立：*

$$\varphi \equiv \bigvee_{(\mathscr{M},s_0)\in Mod(\varphi)} \mathscr{F}_{\mathscr{A}}(\mathscr{M}, s_0).$$

证明. 令$(\mathscr{M}', s_0')$为φ的模型。由定理5.1可知$(\mathscr{M}', s_0') \models \mathscr{F}_{\mathscr{A}}(\mathscr{M}', s_0')$，则：

$$(\mathscr{M}', s_0') \models \bigvee_{(\mathscr{M},s_0)\in Mod(\varphi)} \mathscr{F}_{\mathscr{A}}(\mathscr{M}, s_0).$$

另一方面，假定$(\mathscr{M}', s_0')$为$\bigvee_{(\mathscr{M},s_0)\in Mod(\varphi)} \mathscr{F}_{\mathscr{A}}(\mathscr{M}, s_0)$的模型。则存在$(\mathscr{M}, s_0) \in Mod(\varphi)$使得$(\mathscr{M}', s_0') \models \mathscr{F}_{\mathscr{A}}(\mathscr{M},\ s_0)$。由定理5.1可知$(\mathscr{M}, s_0) \leftrightarrow_{\emptyset} (\mathscr{M}', s_0')$，从而由定理3.1可知$(\mathscr{M}, s_0)$是$\varphi$的一个模型。 □

这一结论表明：任意的CTL公式都与其模型的特征公式的吸取逻辑等价。这对遗忘理论的封闭性提供了重要的理论支撑，也即是从公式里遗忘掉原子命题集合V中的元素只需找到与给定公式的模型V-互模拟的那些模型就能确定遗忘的结果。形式化地，对于给定的公式φ和原子命题集合V，从φ中遗忘掉V中的元素得到的结果为：

$$\bigvee_{\mathscr{K}\in\{\mathscr{K}'\mid\exists\mathscr{K}''\in Mod(\phi),\ \mathscr{K}''\leftrightarrow_V\mathscr{K}'\}} \mathscr{F}_{\overline{V}}(\mathscr{K}).$$

下面分析遗忘在CTL段$\mathrm{CTL}_{\mathrm{AF}}$下各种任务的复杂性结果，其中$\mathrm{CTL}_{\mathrm{AF}}$表示CTL公式只包含时序算子AF的子类。这一类公式在并发系统中的互斥和等待等属性描述中有重要作用[20]，尽管这类公式是相对简单的，但是下面将要说明判定一个模型是否为从此类公式中遗忘掉原子命题集合得到的结果的模型是NP-完全的。

命题 5.3 (模型检测). *给定一个结构$(\mathscr{M}, s_0)$、原子命题集合$V \subseteq \mathscr{A}$和公式$\varphi \in CTL_{\mathrm{AF}}$，判定$(\mathscr{M}, s_0)$是否为$\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$的模型是NP-完全的。*

证明. 在NP中. 假定$(\mathscr{M},s_0)\models \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$，则存在一个初始结构$(\mathscr{M}',s_0')$使得 (a) $(\mathscr{M}',s_0')\models \varphi$且 (b) $(\mathscr{M},s_0)\leftrightarrow_V(\mathscr{M}',s_0')$。已有结果表明，(a)能在$\mathscr{M}'$和$\varphi$的大小的多项式时间内完成[29]。条件(b)也可用Baier等人的推论7.45[20]的方法证明能在多项式时间内完成。又因为猜$(\mathscr{M},s_0)$多项式大小的初始结构$(\mathscr{M}',s_0')$能在多项式时间内完成，因此，这一问题在NP中。

NP-难. 已经证明命题逻辑下遗忘的模型检测是NP-难的，由于命题逻辑下的遗忘是CTL的一种[45]。因此，上述问题是NP-难的。 □

关于遗忘的逻辑蕴涵问题也是值得考虑的，下面讨论$\mathrm{CTL}_{\mathrm{AF}}$段下关于遗忘的逻辑蕴涵的复杂性。下文用记号$\varphi\models^{?}\psi$表示“$\varphi$是否逻辑蕴涵$\psi$”。

定理 5.2 (Entailment). *令φ和ψ为CTL_{AF}中的两个公式，V为原子命题的集合。则：*

(i) *判定$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\models^{?}\psi$是co-NP-完全的，*

(ii) *判定$\psi\models^{?}\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$是$\Pi_2^{\mathrm{p}}$-完全的，*

(iii) *判定$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\models^{?}\mathrm{F}_{\mathrm{CTL}}(\psi,V)$是$\Pi_2^{\mathrm{p}}$-完全的。*

证明. (i) 困难属性. 已有结论表明，判定公式φ是否是可满足的是NP-完全的[120]。要困证难属性，通过令$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\equiv\top$，只需证明$\psi$是有效的，即：原问题是co-NP-难的。

成员属性. 定理6.2表明$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\models\psi$当且仅当$\varphi\models\psi$且$\mathrm{IR}(\psi,V)$。在$\mathrm{CTL}_{\mathrm{AF}}$中，判定$\varphi\models\psi$是co-NP的[120]。这里证明$\mathrm{IR}(\psi,V)$是否成立是co-NP的。不失一般性地，假定$\psi$是可满足的，则$\psi$有一个大小为$|\psi|$的多项式的模型。这里讨论该问题的补问题：判定$\psi$是$V$-有关的，即$\neg\mathrm{IR}(\psi,V)$。显然，$\neg\mathrm{IR}(\psi,V)$当且仅当存在$\psi$的一个模型$(\mathscr{M},s_0)$和大小为$|\psi|$的多项式的初始结构$(\mathscr{M}',s_0')$使得$(\mathscr{M},s_0)\leftrightarrow_V(\mathscr{M}',s_0')$且$(\mathscr{M}',s_0')\not\models\psi$。因此，判定$\neg\mathrm{IR}(\psi,V)$是否成立有如下两个步骤：(1)猜两个大小为$|\psi|$的多项式的初始结构$(\mathscr{M},s_0)$和$(\mathscr{M}',s_0')$使得$(\mathscr{M},s_0)\models\psi$且$(\mathscr{M}',s_0')\not\models\psi$，和(2)检查$(\mathscr{M},s_0)\leftrightarrow_V(\mathscr{M}',s_0')$。显然，(1)和(2)都是能在多项式时间内完成的。

(ii) 成员属性. 考虑这一问题的补问题。猜一个大小为$|\psi|$的多项式的ψ的模型$(\mathscr{M},s_0)$且检查是否$(\mathscr{M},s_0)\not\models\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。由命题5.3 可知这一问题在$\Sigma_2^{\mathrm{p}}$中。因此，原问题在$\Pi_2^{\mathrm{p}}$中。

困难属性. 令$\psi\equiv\top$。则这一问题被规约为判定$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的有效性问题。由于命题遗忘是CTL遗忘的特殊情形，因此该问题的困难属性直接来源于[121]。

(iii) 成员属性. 假定$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\not\models\mathrm{F}_{\mathrm{CTL}}(\psi,V)$。则存在一个初始结构$(\mathscr{M},s)\models\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$且$(\mathscr{M},s)\not\models\mathrm{F}_{\mathrm{CTL}}(\psi,V)$，即存在一个初始结构$(\mathscr{M}_1,s_1)\leftrightarrow_V(\mathscr{M},s)$使得$(\mathscr{M}_1,s_1)\models\varphi$ 但是对其他$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}_2,s_2)$ 使得$(\mathscr{M}_2,s_2)\not\models\psi$。由命题5.3的证明可知，$(\mathscr{M},s)$和$(\mathscr{M}_1,s_1)$

算法 5.1 A Model-based CTL Forgetting Procedure

Input: A CTLformula φ and a set V of atoms
Output: $\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$

```
ψ ← ⊥ foreach initial K-structure 𝒦 (over 𝒜 and 𝒮) do
    if 𝒦 ⊭ φ then continue
    foreach initial K-structure 𝒦' with 𝒦 ↔_V 𝒦' do
        ψ ← ψ ∨ ℱ_V(𝒦')
    end
end
return ψ
```

能在$|\varphi|$、$|\psi|$和$|V|$的多项式时间内完成。显然，猜使得$(\mathscr{M}_1, s_1) \leftrightarrow_V (\mathscr{M}, s)$成立的大小为$|\varphi|$多项式的$(\mathscr{M}, s)$和$(\mathscr{M}_1, s_1)$可以在多项式时间内完成，且对任意的$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}_2, s_2)$，检查$(\mathscr{M}_2, s_2) \not\models \psi$可以在$|\psi|$和$|\mathscr{M}_2|$的多项式时间内完成。因此，该问题是$\Pi_2^{\mathrm{p}}$的。

困难属性. 由于$\mathrm{IR}(\mathrm{F}_{\mathrm{CTL}}(\psi, V), V)$，因此$\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \models \mathrm{F}_{\mathrm{CTL}}(\psi, V)$当且仅当$\varphi \models \mathrm{F}_{\mathrm{CTL}}(\psi, V)$。所以由(ii)不难证明该问题。 □

定理5.2蕴涵下列结论。

推论 5.2. 令φ和ψ为CTL_{AF}中的两个公式，V原子公式的集合。则

(i) 判定$\psi \equiv^{?} \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$是$\Pi_2^{\mathrm{p}}$-完全的，

(ii) 判定$\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \equiv^{?} \varphi$是*co*-NP-完全的，

(iii) 判定$\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \equiv^{?} \mathrm{F}_{\mathrm{CTL}}(\psi, V)$是$\Pi_2^{\mathrm{p}}$-完全的。

5.4 基于模型的遗忘计算方法

前几节中讲述的过程显然给出了一种计算遗忘的方法，本节将这些过程组织起来构成一个算法。该算法与第4.4节中基于语法的算法不同，它是一种基于模型的算法（算法5.1），也就是说本节的算法通过计算所有可能的模型来计算遗忘，且其正确性可由引理5.5和定理5.1保证。

下面的例子来源于研究背景和意义部分，这里给出如何计算从规范中遗忘掉一些原子命题。

例 5.2. 对于图1.1中的$\mathscr{K}_1$，假定一个已知的规范（性质）为$\alpha = \mathrm{EF}(se \wedge sp)$。显然$\mathscr{K}_1$满足$\alpha$。若想移除掉$sp$，即从$\alpha$中遗忘掉$sp$，则$\mathrm{F}_{\mathrm{CTL}}(\alpha, \{sp\}) \equiv \mathrm{EF}se$。此时，该汽车制造公司可以用新的规范EF$se$（这保证驾车最终会被生产）引导未来汽车的生产。

正如下面命题展示的那样，通过计算所有可能的模型是非常低效的，但是这为从理论的角度探索遗忘有重要作用。

命题 5.4. 令φ为*CTL*公式，$V \subseteq \mathscr{A}$为原子命题集合，状态空间的大小为$|\mathscr{S}| = m$，$|\mathscr{A}| = n$和$|V| = x$。则使用算法*5.1*计算从φ中遗忘掉V中原子的空间复杂度为$O((n-x)m^{2(m+2)}2^{nm}\log m)$，且时间复杂性至少与空间复杂性相同。

证明. 假定每个状态或原子命题占据$\log m$的空间且$n \leq m$，则一个状态对(s,s')占据$2\times\log m$位（bit）。对任意的$B \subseteq \mathscr{S}$和$s_0 \in B$，构造如下初始结构$(\mathscr{M},s_0)$，其中$\mathscr{M} = (B,R,L,s_0)$，则$R$中至多有$\frac{|B|^2}{2}$个状态对且$L$中至多$|B|\times n$个$(s,A)$对（$A \subseteq \mathscr{A}$）。因此，$(\mathscr{M},s_0)$至多占据$(|B|+|B|^2+|B|\times n)\times\log m$位。此外，对于状态集合$B$，初始状态有$|B|$种选择，有$|B|^{|B|}$种关系$R$，$(2^n)^{|B|}$种标签函数$L$。

在最坏的情形下（即$|B| = m$），可构成$m\times(m^m\times 2^{nm}\times m)$个初始结构。因此，最多有$m^{m+2}\times 2^{nm}$个初始结构，且最多花费$(m^{m+2}\times 2^{nm}\times(m+m^2+nm))\times\log m$位。

令$k = n-x$，对任意有$i \geq 1$个状态的初始结构$\mathscr{K} = (\mathscr{M},s_0)$，其中$\mathscr{M} = (B,R,L,s_0)$。在最坏的情形下（即$ch(\mathscr{M},V) = i$时），可以花费$N(i) = P_i(s_0)+i\times(P_i(s)+i\times P_i(s'))$位来存储$\mathscr{K}$在$\overline{V}$上的特征公式。其中$s',s \in B$，$P_i(y)$是存储$\mathscr{F}_{\overline{V}}(\mathrm{Tr}_i(y))$（$y \in B$）所用的空间（这里假定EX和AX使用了相同的存储单位）。

$\mathscr{F}_{\overline{V}}(\mathrm{Tr}_i(y))$（$0 \leq n \leq i$）被递归地定义如下：

$$
\begin{aligned}
&(1)n = 0, \qquad P_0(y) = k\\
&(2)n = 1, \qquad P_1(y) = k+i\times k = k+i\times P_0(y)\\
&(3)n = 2, \qquad P_2(y) = k+i\times(k+i\times k) = k+i\times P_1(y)\\
&\cdots \qquad\qquad \cdots\\
&(i+1)n = i, \quad P_i(y) = k+i\times P_{i-1}(y).
\end{aligned}
$$

因此，有

$$
\begin{aligned}
P_i(y) &= k+i\times k+i^2\times k+\cdots+i^i\times k = \frac{i^i-1}{i-1}k,\\
N(i) &= P_i(s_0)+i\times(P_i(s)+i\times P_i(s'))\\
&= (i^2+i+1)P_i(y)\\
&= (i^2+i+1)\frac{i^i-1}{i-1}k.
\end{aligned}
$$

在最坏的情形下（即具有m个状态的初始结构）有$m^{m+2}\times 2^{nm}$个初始结构，需要$(m^{m+2}\times 2^{nm}\times N(m))\times\log m$位来存储遗忘的结果。

因此，空间复杂性为$O((n-x)m^{2(m+2)}2^{nm}\times\log m)$。 □

5.5 本章小结

本章讨论了一种基于模型的计算约束CTL下遗忘的计算。为此，本章第一节首先提出了一种有界V-互模拟概念，并证明了该有界V-互模拟与V-互模拟在有限结构下是等价的。此外，定义了给定深度的计算树在给定原子命题集合上的特征公式，由此定义了有限结构的特征公式。结论表明初始结构能够满足给定的特征公式当且仅当该初始结构与特征公式对应的初始结构在给定原子命题集合上互模拟。基于此，得出了任意公式语义等价于其所有模型的特征公式的吸取，因为可以计算遗忘的结果。最后，本章给出了基于模型的计算遗忘的算法，并分析该算法关于公式的大小是指数空间的。

第六章　μ-演算的遗忘理论

本章探索μ-演算中的遗忘理论。μ-演算是描述转换系统性质的重要逻辑语言，其具有表达能力强的优点：μ-演算是一种表达能力与$S2S$[1]相同的逻辑语言，且LTL、CLT和CTL^*能表达的属性都能用μ-演算来表示。

已有研究表明μ-演算具有均匀插值性质，这蕴涵了μ-演算下的遗忘理论研究本质上与CTL下的不同。其次，本章首先给出μ-演算下遗忘的定义。其次，表明μ-演算下的遗忘是封闭的，这是其与CTL下的遗忘理论的最大的不同。最后，模型检测问题作为形式化验证的重要方法，本章给出μ-演算下遗忘理论的模型检测和推理问题的复杂性结果。

6.1　引言

μ-演算是一种表达能力较强的逻辑语言，它能表达CTL不能表示的性质，例如：Kripke结构中有一条路径，在这条路径上基数位置上的状态满足公式$\neg q \wedge \neg p$，但是偶数位置上的状态满足$q \wedge p$。这一性质不能用CTL公式来表示，但是可以用μ-演算公式表示如下：

$$\varphi = \nu X.(p \wedge q) \wedge \mathrm{EX}(\neg p \wedge \neg q) \wedge \mathrm{EXEX}X.$$

这种情形在日常生活中是很常见的，如：偏序关系$(N, \leq)$（自然数集上的小于等于关系）构成的Kripke结构，其基数节点为基数、偶数节点为偶数。事实上，CTL不能表达具有有规则的性质[122]，其主要原因是

> 对于给定的原子命题p，任意包含n个“X”时序词的命题时序公式（*proposinal temporal formula*，*PTL*）对于序列“$p^i(\neg p)p^w$”有相同的真值，其中$i > n$。

因而得出如下结论：

> 对任意的$m \geq 2$，性质“p在所有状态s_i上为真（$i = k * m$，整数$k \geq 0$）”不能用PTL中的公式来表示。

均匀插值是一个重要的逻辑概念，其有以下含义：给定两个具有$\varphi \models \psi$关系的公式φ和ψ，如果存在公式θ使得$\varphi \models \theta$、$\theta \models \psi$且$Var(\theta) \subseteq Var(\varphi) \cap Var(\psi)$，则称公式$\theta$是$\varphi$和$\psi$的Craig插值。若$\theta$与$\psi$无关，而只与$Var(\varphi) \cap Var(\psi)$有关，则称$\theta$为$\varphi$关

[1]无限完全二叉树下的一元二阶理论（monadic second order theory of the infinite complete binary tree），简称为S2S。

于$Var(\varphi)\cap Var(\psi)$的均匀插值。均匀插值的定义[80]如下（注意：这里的$Var(\varphi)$表示出现在φ中的原子命题和变元的集合）：下面两个结论放到基础知识部分。

定义 6.1. 给定一个μ-句子φ和集合$V\subseteq Var(\varphi)$，φ关于V的均匀插值是满足下列条件的μ-句子θ：

- $\varphi\models\theta$；
- 对任意的公式ψ，若$Var(\varphi)\cap Var(\psi)\subseteq V$且$\varphi\models\psi$，则$\theta\models\psi$；
- $Var(\theta)\subseteq V$。

从直观上来说，φ关于$Var(\varphi)\cap Var(\psi)$的均匀插值是从$\varphi$中“移除”掉不在$Var(\varphi)\cap Var(\psi)$中的元素而保留其在$Var(\varphi)\cap Var(\psi)$上的结论得到的结果。这与遗忘有着密切的关系。

再者，在背景知识部分已经指出，任意的μ-演算公式都能转换成析取μ-公式，在这种形式下的公式的均匀插值是容易计算的，即：

定理 6.1 (定理3.6[80]). 析取μ-公式φ的均匀插值$\exists p\varphi$与μ-公式$\varphi[p/\top,\neg p/\bot]$等价，其中$\varphi[p/\top,\neg p/\bot]$是同时将$p$及其否定$\neg p$分别用$\top$和$\bot$替换得到。

尽管上面的定义中使用的$Var(\varphi)$表示出现在φ中的原子命题和变元的集合，但是均匀插值指的是与原子命题相关的部分，即：对任意的μ-句子φ和原子命题p，存在一个μ-句子$\widetilde{\exists}p.\varphi$使得$\widetilde{\exists}p.\varphi$是$\varphi$关于$\{p\}$的一个均匀插值。这表明在讨论均匀插值时，不考虑变元，即：不是φ关于某个变元的均匀插值。因此，在下文中$Var(\varphi)$仍然用于表示出现在公式φ的原子命题的集合。

本章将要说明本文所定义的μ-演算下的遗忘与文章[80]中定义的均匀插值是一对对偶概念，此时本文给出的遗忘的性质无疑也是均匀插值所具有的性质，这为μ-演算的均匀插值的探索提供了另一种思路。此外，借助于均匀插值的计算方法，本文也给出了计算遗忘的方法。这形成了遗忘和均匀插值之间相辅相成的作用。

本章的组织结构如下。首先，给出μ-演算下遗忘的定义；然后，探讨遗忘的一般性质，并给出其与均匀插值的关系；最后给出与遗忘相关问题的复杂性。

6.2 μ-演算遗忘

与CTL情形下的遗忘相似，这里先给出V互模拟的定义。不失一般性地，令$\mathscr{M}_i=(S_i,R_i,L_i,r_i)$，$i$为自然数集$\mathbb{N}$中的元素。

定义 6.2 (V-互模拟). 给定原子命题集合$V\subseteq\mathscr{A}$和两个*Kripke*结构$\mathscr{M}_1$和$\mathscr{M}_2$。若下面几个条件满足，则称$\mathscr{B}\subseteq S_1\times S_2$是$\mathscr{M}_1$和$\mathscr{M}_2$的$V$-互模拟关系：

- $r_1 \mathscr{B} r_2$，
- 对任意的$s \in S_1$和$t \in S_2$，若$s \mathscr{B} t$则对任意的$p \in \mathscr{A} - V$有$p \in L_1(s)$当且仅当$p \in L_2(t)$，
- $(s,s') \in R_1$和$s \mathscr{B} t$蕴涵存在一个t'使得$s' \mathscr{B} t'$和$(t,t') \in R_2$，且
- 若$s \mathscr{B} t$且$(t,t') \in R_2$，则存在一个s'使得$(s,s') \in R_1$和$t' \mathscr{B} s'$。

一方面，与CTL下的V-互模拟不同的是，这里要求$r_1 \mathscr{B} r_2$（即：$(r_1,r_2) \in \mathscr{B}$）。如果$\mathscr{M}_1$和$\mathscr{M}_2$之间存在一个$V$-互模拟关系$\mathscr{B}$则称这两个Kripke结构$\mathscr{M}_1$和$\mathscr{M}_2$及由这两个Kripke结构构成的结构$(\mathscr{M}_1,r_1)$和$(\mathscr{M}_2,r_2)$是$V$-互模拟的，分别记为$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$和$(\mathscr{M}_1, r_1) \leftrightarrow_V (\mathscr{M}_2,r_2)$。显然，Kripke结构之间的$V$-互模拟与结构之间的$V$-互模拟是等价的：$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$当且仅当$(\mathscr{M}_1,r_1) \leftrightarrow_V (\mathscr{M}_2,r_2)$，因此在下文中只讨论Kripke结构之间$V$-互模拟的性质，而结构之间的互模拟的性质与之相同。不难看出初始结构之间的V-互模拟是结构之间的V-互模拟的一个特例。

另一方面，V-互模拟与$\mathscr{L}$-互模拟[2] [100]类似，不同的是V-互模拟只考虑原子命题，且当$\mathscr{L}$为原子命题集合时是$\mathscr{L}$-互模拟补命题（这里默认除了原子命题之外的符号都是相同的，因此只考虑原子命题）。此外，已有结果表明$\mathscr{L}$-句子（符号只出现在$\mathscr{L}$中的μ-句子）关于$\mathscr{L}$-互模拟是不变的，即若$\mathscr{M}$和$\mathscr{M}'$是$\mathscr{L}$-互模拟的，则对于$\mathscr{L}$-句子φ有$\mathscr{M} \models \varphi$当且仅当$\mathscr{M}' \models \varphi$[98,100]。因此，若IR$(\varphi,V)$且$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，则$\mathscr{M} \models \varphi$当且仅当$\mathscr{M}' \models \varphi$。本文称这一性质为$V$-不变性。

例 6.1. 如图6.1中的两个Kripke结构$\mathscr{M} = (S,R,L,r)$和$\mathscr{M}' = (S',R',L',r')$，其中:

- $S = \{s_0,s_1,s_2\}$，$S' = \{t_0,t_1\}$，$r = s_0$，$r' = t_0$；
- $R = \{(s_0,s_1),(s_1,s_0),(s_0,s_2),(s_2,s_0)\}$，$R' = \{t_0,t_1\}$；
- $L(s_0) = \{ch,j\}$，$L(s_1) = \{ch\}$，$L(s_2) = \emptyset$，$L'(t_0) = \{j\}$，$L'(t_1) = \emptyset$。

由于$\mathscr{M}$和$\mathscr{M}'$之间存在一个二元$\{ch\}$-互模拟$\mathscr{B} = \{(s_0,t_0),(s_1,t_1),(s_2,t_1)\}$，所以$\mathscr{M} \leftrightarrow_{\{ch\}} \mathscr{M}'$。

μ-句子互模拟是不变的，μ-公式则不是[123]。这里定义一种对μ-公式是不变的互模拟。

定义 6.3 (Var-V-互模拟)**.** 给定原子命题集合V、赋值$(\mathscr{M},v)$和$(\mathscr{M}',v')$。$(\mathscr{M},v)$和$(\mathscr{M}',v')$之间的一个*Var-V-互模拟关系*$\mathscr{B}$是$\mathscr{M}$和$\mathscr{M}'$之间一个V-互模拟关系，且满足：若$(s,s') \in \mathscr{B}$，则对任意$X \in \mathscr{V}$，$s \in v(X)$ 当且仅当$s' \in v'(X)$。

[2] $\mathscr{L}$-互模拟是一种将定义6.2中的V替换为$\mathscr{L}$的二元关系。

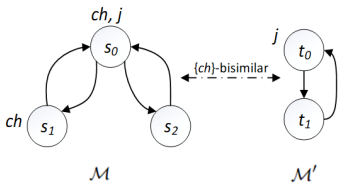


图 6.1: 两个$\{ch\}$-互模拟的Kripke结构示意图

若$(\mathscr{M},v)$和$(\mathscr{M}',v')$之间存在一个Var-V-互模拟关系$\mathscr{B}$，则称$(\mathscr{M},v)$和$(\mathscr{M}',v')$是Var-V-互模拟的，记为$(\mathscr{M},v)\leftrightarrow_V(\mathscr{M}',v')$。对$\mathscr{M}$中的状态集$S_1\subseteq S$，用$\mathscr{B}(S_1)$表示通过$\mathscr{B}$从$S_1$映射得来的状态集，即：$\mathscr{B}(S_1)=\{s'\mid(s,s')\in\mathscr{B},\ s\in S_1\}$。

例 6.2 (例 6.1的延续)**.** 令$\mathscr{M}$和$\mathscr{M}'$为图 6.1中的Kripke结构，$v:\mathscr{V}\to 2^S$和$v':\mathscr{V}\to 2^{S'}$为将$\mathscr{V}$中的变量分别赋值到$\mathscr{M}$和$\mathscr{M}'$的状态集上的赋值函数。可以检查下面的结论成立：

- 若对任意的$X\in\mathscr{V}$，$v(X)=\{s_0,s_1,s_2\}$且$v'(X)=\{t_0,t_1\}$，则$(\mathscr{M},v)\leftrightarrow_{\{ch\}}(\mathscr{M}',v')$；
- 若对任意的$X\in\mathscr{V}-\{X_1\}$，$v(X_1)=\{s_0\}$、$v'(X_1)=\{t_1\}$、$v(X)=\{s_0,s_1,s_2\}$且$v'(X)=\{t_0,t_1\}$，则$(\mathscr{M},v)\not\leftrightarrow_{\{ch\}}(\mathscr{M}',v')$；这是因为$(s_0,t_0)\in\mathscr{B}$且$s_0\in v(X_1)$，但是$t_0\notin v'(X_1)$。

显然，对任意的集$V\subseteq\mathscr{A}$，每个Var-V-互模拟与一个V-互模拟对应，即：$(\mathscr{M},v)\leftrightarrow_V(\mathscr{M}',v')$ 蕴涵$\mathscr{M}\leftrightarrow_V\mathscr{M}'$。另一方面，Kripke结构$\mathscr{M}$和$\mathscr{M}'$之间的每个$V$-互模拟可被扩展为$(\mathscr{M},v)$和$(\mathscr{M}',v')$一个Var-$V$-互模拟，其中对任意的$X\in\mathscr{V}$，$v(X)=S$且$v'(X)=S'$，$S$和$S'$分别为$\mathscr{M}$和$\mathscr{M}'$中的状态集合。

可以容易证明$\leftrightarrow_V$有如下性质。

命题 6.1. *令$V,V_1\subseteq\mathscr{A}$为原子命题的集合，$(\mathscr{M}_1,v_1)$、$(\mathscr{M}_2,v_2)$和$(\mathscr{M}_3,v_3)$为赋值，则：*

(i) *$\leftrightarrow_V$是赋值之间的等价关系；*

(ii) *若$(\mathscr{M}_1,v_1)\leftrightarrow_V(\mathscr{M}_2,v_2)$和$V\subseteq V_1$，则$(\mathscr{M}_1,v_1)\leftrightarrow_{V_1}(\mathscr{M}_2,v_2)$；*

(iii) *若$(\mathscr{M}_1,v_1)\leftrightarrow_V(\mathscr{M}_2,v_2)$且$(\mathscr{M}_2,v_2)\leftrightarrow_{V_1}(\mathscr{M}_3,v_3)$，则$(\mathscr{M}_1,v_1)\leftrightarrow_{V\cup V_1}(\mathscr{M}_3,v_3)$。*

证明. (i) 这里从自反性、对称性和传递性来证明该关系是一个等价关系。

(1) $\leftrightarrow_V$是自反的。容易检查对任意的赋值$(\mathscr{M},v)$都有$(\mathscr{M},v)\leftrightarrow_V(\mathscr{M},v)$。

(2) $\leftrightarrow_V$是对称的。这里证明对任意的$(\mathscr{M}_1,v_1)$和$(\mathscr{M}_2,v_2)$，若$(\mathscr{M}_1,v_1)\leftrightarrow_V(\mathscr{M}_2,v_2)$，则$(\mathscr{M}_2,v_2)\leftrightarrow_V(\mathscr{M}_1,v_1)$。假设$(\mathscr{M}_1,v_1)$和$(\mathscr{M}_2,v_2)$之间的Var-$V$-互模拟关系$\mathscr{B}$，构造如下二元关系：$\mathscr{B}_1=\{(s,t)\mid(t,s)\in\mathscr{B}\}$。现在从下面几点证明$\mathscr{B}_1$是$\mathscr{M}_2$和$\mathscr{M}_1$之间的一个$V$-互模拟关系：

- 由于$r_1 \mathscr{B} r_2$，所以$r_2 \mathscr{B}_1 r_1$，
- 对任意的$s \in S_1$和$t \in S_2$，若$t \mathscr{B}_1 s$，则$s \mathscr{B} t$，因此对于任意的$p \in \mathscr{A} - V$，$p \in L_1(s)$当且仅当$p \in L_2(t)$，且
- 因为$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_2$之间的V-互模拟关系，所以V-互模拟的第三和第四个点很容易能够证明。

此外，若$(s,t) \in \mathscr{B}$，则对任意的$X \in \mathscr{V}$，$s \in v_1(X)$ 当且仅当$t \in v_2(X)$。因此，若$(t,s) \in \mathscr{B}_1$，则对任意的$X \in \mathscr{V}$，$t \in v_2(X)$ 当且仅当$s \in v_1(X)$。所以，$(\mathscr{M}_2, v_2) \leftrightarrow_V (\mathscr{M}_1, v_1)$。

(3) $\leftrightarrow_V$是传递的。这里证明若$(\mathscr{M}_1, v_1) \leftrightarrow_V (\mathscr{M}_2, v_2)$ 和$(\mathscr{M}_2, v_2) \leftrightarrow_V (\mathscr{M}_3, v_3)$，则$(\mathscr{M}_1, v_1) \leftrightarrow_V (\mathscr{M}_3, v_3)$。假定$(\mathscr{M}_1, v_1)$和$(\mathscr{M}_2, v_2)$之间的Var-$V$-互模拟关系为$\mathscr{B}_1$，$(\mathscr{M}_2, v_2)$和$(\mathscr{M}_3, v_3)$之间的Var-$V$-互模拟关系为$\mathscr{B}_2$，构造如下二元关系：可以类似(2)证明$\mathscr{B}$为$\mathscr{M}_1$和$\mathscr{M}_3$的一个$V$-互模拟关系，即：$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_3$。

此外，若$(s,z) \in \mathscr{B}$，则存在$t \in S_2$使得$(s,t) \in \mathscr{B}_1$ 和$(t,z) \in \mathscr{B}_2$，所以，对任意的$X \in \mathscr{V}$，$s \in v_1(X)$ 当且仅当$t \in v_2(X)$ 当且仅当$z \in \mathscr{B}$。因此，$(\mathscr{M}_1, v_1) \leftrightarrow_V (\mathscr{M}_3, v_3)$。

(ii) 假设$\mathscr{B}_V$是$(\mathscr{M}_1, v_1)$ 和$(\mathscr{M}_2, v_2)$之间的一个Var-V-互模拟关系。这里证明$\mathscr{B}_V$是$(\mathscr{M}_1, v_1)$ 和$(\mathscr{M}_2, v_2)$之间的一个Var-V_1-互模拟关系。显然：

- $(r_1, r_2) \in \mathscr{B}_V$，
- 对任意的$w_1 \in S_1$和$w_2 \in S_2$，若$(w_1, w_2) \in \mathscr{B}_V$，因为$V \subseteq V_1$且对任意的$p \in \mathscr{A} - V$，$p \in L_1(w_1)$ 当且仅当$p \in L_2(w_2)$，所以，对任意的$p \in \mathscr{A} - V_1$，$p \in L_1(w_1)$ 当且仅当$p \in L_2(w_2)$，
- 若$(w_1, r_1) \in R_1$ 和$(w_1, w_2) \in \mathscr{B}_V$，因为$\mathscr{B}_V$ 是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V-互模拟关系，所以，$\exists r_2 \in S_2$使得$(w_2, r_2) \in R_2$ 和$(r_1, r_2) \in \mathscr{B}_V$，
- 若$(w_2, r_2) \in R_2$ 和$(w_1, w_2) \in \mathscr{B}_V$，因为$\mathscr{B}_V$ 是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V-互模拟关系，所以，$\exists r_1 \in S_1$ 使得$(w_1, r_1) \in R_1$ 和$(r_1, r_2) \in \mathscr{B}_V$。

因此，$\mathscr{B}_V \mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_1-互模拟关系。

此外，若$(s,t) \in \mathscr{B}_V$，则对任意的$X \in \mathscr{V}$，$s \in v_1(X)$ 当且仅当$t \in v_2(X)$。因此，$(\mathscr{M}_1, v_1) \leftrightarrow_{V_1} (\mathscr{M}_2, v_2)$。

(iii) 由(ii)可知，$(\mathscr{M}_1, v_1) \leftrightarrow_V (\mathscr{M}_2, v_2)$ 蕴涵$(\mathscr{M}_1, v_1) \leftrightarrow_{V \cup V_1} (\mathscr{M}_2, v_2)$，由(i)的传递性可知$(\mathscr{M}_1, v_1) \leftrightarrow_{V \cup V_1} (\mathscr{M}_3, v_3)$。 □

直观地说，(i)表示$\leftrightarrow_V$是Kripke结构的集合上的自反、对称和传递关系。(ii)表示如果一个Kripke结构和其他两个Kripke结构互相V和V_1互模拟，则这两个Kripke结构$V \cup V_1$-互模拟。这跟CTL情形下的互模拟类似，正如下文将要说到的那样，这一性质有助于证明μ-演算下遗忘的模块属性。

此时，μ-演算下的遗忘如下定义：

定义 6.4 (μ-演算下的遗忘)**.** 令$V \subseteq \mathscr{A}$和ϕ为μ-句子。若下面等式成立，则称V上的μ-句子ψ是从ϕ中遗忘掉V后得到的结果，记为$\mathrm{F}_\mu(\phi, V)$：

$$Mod(\psi) = \{\mathscr{M} \mid \exists \mathscr{M}' \in Mod(\phi) \ \&\ \mathscr{M}' \leftrightarrow_V \mathscr{M}\}.$$

定义6.4表明如果ψ和ψ'都是从ϕ中遗忘掉V中的原子命题得到的结果，则$Mod(\psi) = Mod(\psi')$，也就是说遗忘的结果之间是语义等价的（即有相同的模型）。

D'Agostino 研究了μ-演算下的均匀插值，并指出μ-算具有均匀插值性质[80,100-101]。换句话说，这意味着对任意的μ-句子φ和有限的原子命题的集合$V \subseteq Var(\varphi)$，都存在一个V-无关且与φ最接近的μ-句子$\widetilde{\exists} V\varphi$。值得注意的是，上述定义的遗忘$\mathrm{F}_\mu(\phi, V)$与$\widetilde{\exists} V\varphi$[80]语义等价.

6.3 μ-演算遗忘的性质

这部分展示μ-演算下遗忘的语义属性。特别地，这里将证明上述μ-演算下的遗忘的定义与遗忘的那几条规则具有“当且仅当的关系”，且从任意μ-句子中遗忘掉任意原子命题的集合的结果总是一个μ-句子。此外，也研究了遗忘算子的代数属性，包括分解性（decomposition）、切片性（slice）和同质性（homogeneity）。

定理 6.2. 给定原子命题$q \in \mathscr{A}$和μ-句子ϕ，则存在一个μ-句子ψ使得$IR(\psi, \{q\})$且$\psi \equiv \mathrm{F}_\mu(\phi, \{q\})$。

证明. 已有结果表明（定理3.1[100]），对任意的μ-句子ϕ和原子命题p，存在一个$\{p\}$-无关的μ-句子ϕ'（即$\mathrm{IR}(\phi', \{p\})$）使得：

$$\mathscr{M} \models \phi' \text{ 当且仅当} \exists \mathscr{M}' \in Mod(\phi) \text{ 使得} \mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}'.$$

这与本文遗忘的定义一致，因此上述结论成立。 □

与模态S5和CTL情形类似，下面给出μ-演算下遗忘的基本公设：

(**W**) 削弱：$\varphi \models \varphi'$;

(PP) 正支持：对任意的μ-句子η，若$\mathrm{IR}(\eta,V)$和$\varphi \models \eta$则$\varphi' \models \eta$；

(NP) 负支持：对任意的μ-句子η，若$\mathrm{IR}(\eta,V)$和$\varphi \not\models \eta$则$\varphi' \not\models \eta$；

(IR) 无关性：$\mathrm{IR}(\varphi',V)$。

其中$V \subseteq \mathscr{A}$、φ为μ-句子、φ'是从φ中遗忘掉V后得到的结果。

定理 6.3 (表示性定理)**.** *给定μ-句子φ、φ'和ϕ，$V \subseteq \mathscr{A}$为原子命题的集合。下面的几个陈述是等价的：*

(i) $\varphi' \equiv \mathrm{F}_\mu(\varphi,V)$,

(ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi$ *且*$IR(\phi,V)\}$,

*(iii) 若φ、φ'及V和(i)、(ii)中的符号表示相同公式和原子命题的集合，则**(W)**、**(PP)**、**(NP)** 和**(IR)** 成立。*

证明. $(i) \Leftrightarrow (ii)$. 为了证明这一结论成立，只需证明：

$$Mod(\mathrm{F}_\mu(\varphi,V)) = Mod(\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}).$$

$(\Rightarrow)$ 对$\mathrm{F}_\mu(\varphi,V)$的任意模型$\mathscr{M}'$
$\Rightarrow \exists \mathscr{M}$使得$\mathscr{M} \models \varphi$和$\mathscr{M} \leftrightarrow_V \mathscr{M}'$ (定义6.4)
$\Rightarrow$ 对于任意与V-无关且$\varphi \models \phi$的ϕ都有$\mathscr{M}' \models \phi$
$\Rightarrow \mathscr{M}' \models \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$

$(\Leftarrow)$ 由于$\mathrm{IR}(\mathrm{F}_\mu(\varphi,V),V)$和$\varphi \models \mathrm{F}_\mu(\varphi,V)$，由定义6.4可知$\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\} \models \mathrm{F}_\mu(\varphi,V)$。

$(ii) \Rightarrow (iii)$. 为了方便，令$A = \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$。首先，对任意的$A$中的公式$\phi'$都有$\mathrm{IR}(\phi',V)$，所以有$\mathrm{IR}(A,V)$。因此，$\mathrm{IR}(\varphi',V)$。其次，对任意的$\phi' \in A$，都有$\varphi \models \phi'$，所以$\varphi \models \varphi'$。第三，$\forall \phi$且$\mathrm{IR}(\phi,V)$，若$\varphi \models \phi$则$\phi \in A$，因而$\varphi' \models \phi$。最后，$\forall \phi$且$\mathrm{IR}(\phi,V)$，若$\varphi \not\models \phi$则$\phi \notin A$。因此，由定义6.4和$V$-无关性可知$\varphi' \not\models \phi$。

$(iii) \Rightarrow (ii)$. (1) $\varphi' \models \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$ (**(PP)**)
(2) $\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\} \models \varphi'$ (**(W)**，**(IR)**)
$\Rightarrow \varphi' \equiv \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$ ((1)，(2)). □

定理6.3 表明基本公设对μ-演算下的遗忘是充分且必要的：基本公设能描述遗忘的结果，遗忘的结果具有基本公设里的性质。这与S5和CTL下的情形相同。

除了上述的表示性定理，后文将说明公设**(IR)**对计算SNC 和WSC是重要的。对于μ-句子$\psi = \varphi \wedge (q \leftrightarrow \alpha)$，$\varphi \wedge \alpha$是$\{q\}$-无关的，则从$\psi$中遗忘掉$q$得到的结果是$\varphi$。正

如将在第七章中展示，这一性质有助于将任意公式的SNC（WSC）转换为命题下的SNC（WSC）。这一性质可形式化如下：

引理 6.1. *令φ和α为两个μ-句子，q为原子命题且$q \notin Var(\varphi) \cup Var(\alpha)$。则*$\mathrm{F}_\mu(\varphi \wedge (q \leftrightarrow \alpha), q) \equiv \varphi$。

证明. 令$\varphi' = \varphi \wedge (q \leftrightarrow \alpha)$。对$\mathrm{F}_\mu(\varphi', q)$的任意一个模型$\mathscr{M}$，存在一个Kripke结构$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_{\{q\}} \mathscr{M}'$和$\mathscr{M}' \models \varphi'$。显然有$\mathscr{M}' \models \varphi$，又因为$\mathrm{IR}(\varphi, \{q\})$和$\mathscr{M} \leftrightarrow_{\{q\}} \mathscr{M}'$，所以$\mathscr{M} \models \varphi$。

令$\mathscr{M} \in Mod(\varphi)$，其中$\mathscr{M} = (S, R, L, s)$。如下构造$\mathscr{M}' = (S, R, L', s)$：

$$
\begin{aligned}
&L' : S \to 2^{\mathscr{A}} \ and \ \forall s^* \in S, L'(s^*) = L(s^*) - \{q\} \ if \ (\mathscr{M}, s^*) \not\models \alpha, \\
&\text{否则} L'(s^*) = L(s^*) \cup \{q\}, \\
&L'(s) = L(s) \cup \{q\} \ if \ (\mathscr{M}, s) \models \alpha, \ and \ L'(s) = L(s) \text{否则}.
\end{aligned}
$$

显然$\mathscr{M}' \models \varphi$、$\mathscr{M}' \models q \leftrightarrow \alpha$且$\mathscr{M}' \leftrightarrow_{\{q\}} \mathscr{M}$。因此，$\mathscr{M}' \models \varphi \wedge (q \leftrightarrow \alpha)$，又因为$\mathscr{M}' \leftrightarrow_{\{q\}} \mathscr{M}$和$\mathrm{IR}(\mathrm{F}_\mu(\varphi \wedge (q \leftrightarrow \alpha), q), \{q\})$，所以$\mathscr{M} \models \mathrm{F}_\mu(\varphi \wedge (q \leftrightarrow \alpha), q)$。 □

正如在第一章中所说的，遗忘在经典命题逻辑中首先被提出，并应用于各种领域。这里给出经典命题逻辑与μ-演算下的遗忘之间的联系。

首先，回顾一下从命题公式φ中遗忘原子命题p得到的结果为$Forget(\varphi, \{p\}) \equiv \varphi[p/\bot] \vee \varphi[p/\top]$，且$Forget(\varphi, V \cup \{p\})$被递归地定义为：$Forget(Forget(\varphi, \{p\}), V)$，其中$Forget(\varphi, \emptyset) = \varphi$。此外，对于给定的Kripke结构$\mathscr{M} = (S, R, L, r)$和命题公式$\psi$，$\mathscr{M} \models \psi$当且仅当$L(r) \models \psi$。经典命题逻辑与$\mu$-演算下的遗忘之间的联系如下：

定理 6.4. *令φ为命题公式，$V \subseteq \mathscr{A}$为原子命题的集合，则*

$$\mathrm{F}_\mu(\varphi, V) \equiv Forget(\varphi, V).$$

证明. 令$\mathscr{M} = (S, R, L, r)$和$\mathscr{M}' = (S', R', L', r')$为Kripke结构。

($\Rightarrow$) 对任意的$\mathscr{M} \in Mod(\mathrm{F}_\mu(\varphi, V))$

$\Rightarrow$ 由定义3.3可知存在$\mathscr{M}' \in Mod(\varphi)$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，且$\mathscr{M}$和$\mathscr{M}'$之间的$V$-互模拟关系为$\mathscr{B}$

$\Rightarrow r \mathscr{B} r'$

$\Rightarrow \mathscr{M} \models Forget(\varphi, V)$ ($\mathrm{IR}(Forget(\varphi, V), V)$, V-无关性)

($\Leftarrow$) 对任意的$\mathscr{M} \in Mod(Forget(\varphi, V))$

$\Rightarrow \exists \mathscr{M}' \in Mod(\varphi)$使得$\forall p \in \mathscr{A} - V$，$p \in L(r)$当且仅当$p \in L'(r')$ ($Forget$的定义)

如下构造Kripke 结构$\mathscr{M}_1 = (S_1, R_1, L_1, r_1)$：

* $S_1 = (S - \{r\}) \cup \{r_1\}$,

* R_1与R相同，除了r被r_1替换，且

* L_1与L相同，除了$L_1(r_1) = L'(r')$。

$\Rightarrow \mathscr{M}_1 \models \varphi$且$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}$
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\varphi, V)$　　(IR($\mathrm{F}_\mu(\varphi, V), V$)-无关性）　　□

定理6.4表明μ-演算下的遗忘是命题逻辑下遗忘的扩展，这提示我们是否命题情形下遗忘拥有的性质μ-演算下的遗忘也具有。下面的性质在命题逻辑、S5[44] 和CTL中都成立，这里证明其在μ-演算中也成立。

命题 6.2. *给定μ-句子φ、φ_i和ψ_i（$i = 1, 2$），$V \subseteq \mathscr{A}$为原子命题的集合。则：*

(i) $\mathrm{F}_\mu(\varphi, V)$*是可满足的当且仅当*$\varphi$*是可满足的；*

(ii) *若*$\varphi_1 \equiv \varphi_2$，*则*$\mathrm{F}_\mu(\varphi_1, V) \equiv \mathrm{F}_\mu(\varphi_2, V)$；

(iii) *若*$\varphi_1 \models \varphi_2$，*则*$\mathrm{F}_\mu(\varphi_1, V) \models \mathrm{F}_\mu(\varphi_2, V)$；

(iv) $\mathrm{F}_\mu(\psi_1 \vee \psi_2, V) \equiv \mathrm{F}_\mu(\psi_1, V) \vee \mathrm{F}_\mu(\psi_2, V)$,

(v) $\mathrm{F}_\mu(\psi_1 \wedge \psi_2, V) \models \mathrm{F}_\mu(\psi_1, V) \wedge \mathrm{F}_\mu(\psi_2, V)$。

证明. (i) ($\Rightarrow$) 假设$\mathscr{M}$是$\mathrm{F}_\mu(\varphi, V)$的模型，由$\mathrm{F}_\mu$的定义可知，存在$\varphi$的一个模型$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$。

($\Leftarrow$) 假定$\mathscr{M}$是φ的模型，则存在一个Kripke 结构$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，因此由$\mathrm{F}_\mu$的定义可知$\mathscr{M}' \models \mathrm{F}_\mu(\varphi, V)$。

(ii) 和(iii) 可以类似地证明。

(iv) ($\Rightarrow$) $\forall \mathscr{M} \in Mod(\mathrm{F}_\mu(\psi_1 \vee \psi_2, V))$
$\Rightarrow \exists \mathscr{M}' \in Mod(\psi_1 \vee \psi_2)$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$和$\mathscr{M}' \models \psi_1$（或$\mathscr{M}' \models \psi_2$）
$\Rightarrow \exists \mathscr{M}_1 \in Mod(\mathrm{F}_\mu(\psi_1, V))$使得$\mathscr{M}' \leftrightarrow_V \mathscr{M}_1$，或者$\exists \mathscr{M}_2 \in Mod(\mathrm{F}_\mu(\psi_2, V))$使得$\mathscr{M}' \leftrightarrow_V \mathscr{M}_2$
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1, V) \vee \mathrm{F}_\mu(\psi_2, V)$。

($\Leftarrow$) $\forall \mathscr{M} \in Mod(\mathrm{F}_\mu(\psi_1, V) \vee \mathrm{F}_\mu(\psi_2, V))$
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1, V)$或$\mathscr{M} \models \mathrm{F}_\mu(\psi_2, V)$
$\Rightarrow \exists \mathscr{M}_1$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}_1$，且$\mathscr{M}_1 \models \psi_1$或者$\mathscr{M}_1 \models \psi_2$　　（定义6.4）
$\Rightarrow \mathscr{M}_1 \models \psi_1 \vee \psi_2$

$\Rightarrow \exists \mathscr{M}_2$使得$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$和$\mathscr{M}_2 \models \mathrm{F}_\mu(\psi_1 \vee \psi_2, V)$
$\Rightarrow \mathscr{M} \leftrightarrow_V \mathscr{M}_2$ （命题6.1）
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1 \vee \psi_2, V)$ （定义6.4）。

(v)可以像(iv)一样证明。 □

命题6.2(i) 表明从一个μ-句子中遗忘掉一些原子命题不影响该句子的可满足性；从(ii)可以看出，如果两个句子是等价的，则他们遗忘相同原子命题得到的结果是等价的；(iv)指出析取公式$\varphi_1 \vee \varphi_2$的遗忘可以由分开计算遗忘后在析取而得到；而正如(v) 中指出的那样，合取公式$\psi_1 \wedge \psi_2$的遗忘不能分别计算再合取，这甚至对于命题公式都是不成立。例：令$\varphi = p \wedge (q \vee \neg p)$，从$\varphi$中遗忘掉$p$的结果为$q$，但是$Forget(p, \{p\}) \wedge Forget(q \vee \neg p, \{p\}) \equiv \top$。显然二者不等价。

下面是关于μ-演算下遗忘算子的其他性质。

命题 6.3 (分解性)**.** *给定μ-句子φ、原子命题的集合V和原子命题p且$p \notin V$，则：*

$$\mathrm{F}_\mu(\varphi, \{p\} \cup V) \equiv \mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, \{p\}), V).$$

证明. 令$\mathscr{M}_1 = (S_1, R_1, L_1, s_1)$为$\mathrm{F}_\mu(\varphi, \{p\} \cup V)$的模型。由遗忘的定义可知，存在$\varphi$的一个模型$\mathscr{M}$（$\mathscr{M} = (S, R, L, s)$）使得$\mathscr{M}_1 \leftrightarrow_{\{p\} \cup V} \mathscr{M}$。如下构造Kripke结构$\mathscr{M}_2 = (S_2, R_2, L_2, s_2)$：

(1) 对于s_2，令s_2为满足下列条件的状态：

- $p \in L_2(s_2)$当且仅当$p \in L_1(s_1)$，
- 对任意的$q \in V$，$q \in L_2(s_2)$当且仅当$q \in L(s)$，
- 对于其他的原子命题q'，$q' \in L_2(s_2)$当且仅当$q' \in L_1(s_1)$当且仅当$q' \in L(s)$。

(2) 其他情形：假定$\mathscr{M}_1$和$\mathscr{M}$有$\{p\} \cup V$-互模拟关系$\mathscr{B}$。

(i) 对任意的$w \in S$和$w_1 \in S_1$且$(w, w_1) \in \mathscr{B}$，令$w_2 \in S_2$和

- $p \in L_2(w_2)$当且仅当$p \in L_1(w_1)$，
- 对任意的$q \in V$，$q \in L_2(w_2)$当且仅当$q \in L(w)$，
- 对其他原子命题q'，$q' \in L_2(w_2)$当且仅当$q' \in L_1(w_1)$当且仅当$q' \in L(w)$。

(ii) 若$(w_1', w_1) \in R_1$，且w_2是由w_1构造，$w_2' \in S_2$由w_1'构造，则$(w_2', w_2) \in R_2$。

- 删除S_2和R_2中重复的元素。

可以容易检查$\mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}_2$和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$。因此，$(\mathscr{M}_2, s_2) \models \mathrm{F}_\mu(\varphi, p)$，所以$(\mathscr{M}_1, s_1) \models \mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, p), V)$。

另一方面，假设$\mathscr{M}_1$是$\mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, p), V)$的一个模型

$\Rightarrow \exists \mathscr{M}_2$使得$\mathscr{M}_2 \models \mathrm{F}_\mu(\varphi, p)$和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$　(定义6.4)

$\Rightarrow \exists \mathscr{M}$使得$\mathscr{M} \models \varphi$和$\mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}_2$　(定义6.4)

因此，由命题6.1可知$\mathscr{M} \leftrightarrow_{\{p\} \cup V} \mathscr{M}_1$，因此$\mathscr{M}_1 \models \mathrm{F}_\mu(\varphi, \{p\} \cup V)$。　□

下面这一性质为上述命题的推论。

推论 6.1 (切片性). *给定μ-句子φ和原子命题的集合$V_i \subseteq \mathscr{A}$ $(i = 1, 2)$，有：*

$$\mathrm{F}_\mu(\varphi, V_1 \cup V_2) \equiv \mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, V_1), V_2).$$

F_μ的另一个属性是关于AX和EX时序算子的：形如AXφ或EXφ的μ-句子的遗忘可以提到AX和EX后面计算。而对于$\mu X.\varphi$和$\nu X.\varphi$就没有这样的性质，因为φ显然不是一个μ-句子。

命题 6.4 (同质性). *给定原子命题集合$V \subseteq \mathscr{A}$和μ-句子ϕ，则：*

(i) $\mathrm{F}_\mu(\mathrm{AX}\phi, V) \equiv \mathrm{AX}\mathrm{F}_\mu(\phi, V)$.

(ii) $\mathrm{F}_\mu(\mathrm{EX}\phi, V) \equiv \mathrm{EX}\mathrm{F}_\mu(\phi, V)$.

证明. 令$\mathscr{M} = (S, R, L, s)$、$\mathscr{M}_i = (S_i, R_i, L_i, s_i)$（$i \in \mathrm{N}$）且$\mathscr{M}' = (S', R', L', s')$。

(i) $(\Rightarrow)$ $\mathscr{M} \models \mathrm{F}_\mu(\mathrm{AX}\phi, V)$

$\Rightarrow$ 有$\mathscr{M} \leftrightarrow_V \mathscr{M}'$且$\mathscr{M}' \models \mathrm{AX}\phi$，即$\mathscr{M}$和$\mathscr{M}'$之间存在$V$-互模拟关系$\mathscr{B}_V$

$\Rightarrow \mathscr{M} \leftrightarrow_V \mathscr{M}'$，且对任意的$(s', s'') \in R'$有$(\mathscr{M}', s'') \models \phi$

$\Rightarrow$ 对任意的$(s, s_1) \in R$，存在$(s', s'_1) \in R'$使得$(s_1, s'_1) \in \mathscr{B}_V$，且对任意的$(s', s'') \in R'$有$(\mathscr{M}', s'') \models \phi$

$\Rightarrow$ 对任意的$(s, s_1) \in R$，有$(s_1, s'_1) \in \mathscr{B}_V$和$(\mathscr{M}', s'_1) \models \phi$

$\Rightarrow$ 对任意的$(s, s_1) \in R$，$(\mathscr{M}, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow (\mathscr{M}, s) \models \mathrm{AX}\mathrm{F}_{\mathrm{CTL}}(\phi, V)$，即$\mathscr{M} \models \mathrm{AX}\mathrm{F}_{\mathrm{CTL}}(\phi, V)$。

$(\Leftarrow)$ $(\mathscr{M}, s) \models \mathrm{AX}\mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow$ 对任意的$(s, s_1) \in R$，$(\mathscr{M}, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$ $(R \in \mathscr{M})$

$\Rightarrow$ 对任意的$(s, s_1) \in R$，有$(\mathscr{M}, s_1) \leftrightarrow_V (\mathscr{M}', s')$和$(\mathscr{M}', s') \models \phi$，即$\mathscr{M}$和$\mathscr{M}'$之间存在$V$-互模拟关系$\mathscr{B}_V$

$\Rightarrow$ 对任意的$i \geq 0$，有$(s_i, s'_i) \in \mathscr{B}'_V$且$(\mathscr{M}'_i, s'_i) \models \phi$，即$\mathscr{M}$和$\mathscr{M}'_i$之间存在$V$-互模拟关系$\mathscr{B}'_V$其中$\{s_0, s_1, \ldots\} = \{s' \mid (s, s') \in R\}$和$\mathscr{M}'_i = (S'_i, R'_i, L'_i, [_]'_i, s'_i)$（当$i \neq j$时，假定$S'_i \cap S'_j = \emptyset$）

$\Rightarrow (\mathscr{M}^*, s) \leftrightarrow_V (\mathscr{M}, s)$和$(\mathscr{M}^*, s) \models \mathrm{AX}\phi$，其中$\mathscr{M}^* = (S^*, R^*, L^*, [_], s)$和

- $S^* = \{s\} \cup \bigcup_{i\geq 0} S'_i$，
- $R^* = \{(s, s'_i) \mid i \geq 0\} \cup \bigcup_{i\geq 0} R'_i$，
- $L^* = \bigcup_{i\geq 0} L'_i$和$L^*(s) = L(s)$，其中$L \in \mathscr{M}$。

$\Rightarrow (\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V)$。

(ii) ($\Rightarrow$) $(\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi, V)$

$\Rightarrow$ 有$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$且$(\mathscr{M}', s') \models \mathrm{EX}\phi$，即$\mathscr{M}$和$\mathscr{M}'$之间存在$V$-互模拟关系$\mathscr{B}_V$

$\Rightarrow$ 有$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$，且对一些$(s', s'') \in R'$有$(\mathscr{M}', s'') \models \phi$（$R' \in \mathscr{M}'$）

$\Rightarrow$ 对一些$(s, s_1) \in R$，有$(s', s'_1) \in R'$和$(s_1, s'_1) \in \mathscr{B}_V$，且对一些$(s', s'') \in R'$有$(\mathscr{M}', s'') \models \phi$（$R \in \mathscr{M}$）

$\Rightarrow$ 对一些$(s, s_1) \in R$，有$(s_1, s'_1) \in \mathscr{B}_V$和$(\mathscr{M}', s'_1) \models \phi$

$\Rightarrow$ 对一些$(s, s_1) \in R$，有$(\mathscr{M}, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow (\mathscr{M}, s) \models \mathrm{EXF}_{\mathrm{CTL}}(\phi, V)$。

($\Leftarrow$) $(\mathscr{M}, s) \models \mathrm{EXF}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow$ 对一些$(s, s') \in R$，$(\mathscr{M}, s') \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$（$R \in \mathscr{M}$）

$\Rightarrow$ 对一些$(s, s') \in R$，有$(\mathscr{M}, s') \leftrightarrow_V (\mathscr{M}', s'')$和$(\mathscr{M}', s'') \models \phi$，其中$\mathscr{M}' = (S', R', L', [_]', s'')$

$\Rightarrow (\mathscr{M}^*, s) \leftrightarrow_V (\mathscr{M}, s)$和$(\mathscr{M}^*, s) \models \mathrm{EX}\phi$，其中$\mathscr{M}^* = (S^*, R^*, L^*, [_], s)$，

- $S^* = S \cup S'$，
- $R^* = \{(s, s'')\} \cup R \cup R'$，
- $L^* = L \cup L'$和$L^*(s) = L(s)$，其中$L \in \mathscr{M}$。

$\Rightarrow (\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi, V)$。

□

AX（或EX）在F_μ上的同质性表明，在从AXφ（或EXφ）遗忘掉V中的原子命题等价于将F_μ提到AX和EX后面计算的结果。特别地，当命题6.4中的公式ϕ为命题公式时，从$Q\mathrm{X}\phi$ ($Q \in \{\mathrm{E}, \mathrm{A}\}$)中遗忘原子命题可以使用命题逻辑的遗忘计算方法来计算。

6.4 计算复杂性

析取μ-公式φ的均匀插值为$\widetilde{\exists} p\varphi$（$p \in \mathscr{A}$），且与$\varphi[p/\top, \neg p/\top]$等价[80]，其中$\varphi[p/\top, \neg p/\top]$表示将$\varphi$中的$p$和$\neg p$同时用$\top$替换。正如之前说过的$\mathrm{F}_\mu(\varphi, V)$与均匀插值$\widetilde{\exists} V\varphi$等价[80]。因此，下面的命题容易证明。

推论 6.2. *给定μ-句子φ和原子命题$p \in \mathscr{A}$。若φ是一个析取μ-句子，$\mathrm{F}_\mu(\varphi,\{p\})$能在线性时间内计算。*

证明. content... □

这种情况下，可以首先将一个μ-句子转化为析取μ-公式，再去计算遗忘。下面的例子给出如何计算从析取μ-公式中遗忘“ch”。

例 6.3. 令$\varphi_1 = j \wedge ch \wedge Cover(\neg j \wedge \neg ch, \top)$、$\varphi_2 = \mu X.(j \wedge ch) \wedge Cover(X, \top)$、$\varphi_3 = \nu X.(j \wedge ch) \wedge Cover(Cover(X, \top), \top)$。令$V = \{ch\}$，我们能容易地计算从这些公式里遗忘掉$V$。

(1) $\mathrm{F}_\mu(\varphi_1, V) \equiv j \wedge Cover(\neg j, \top) \equiv j \wedge \mathrm{EX}(\neg j)$；

(2) $\mathrm{F}_\mu(\varphi_2, V) \equiv \mu X.j \wedge Cover(X, \top) \equiv \mu X.j \wedge \mathrm{EX}X$；

(3) $\mathrm{F}_\mu(\varphi_2, V) \equiv \nu X.j \wedge Cover(Cover(X, \top), \top) \equiv \nu X.j \wedge \mathrm{EX}(\mathrm{EX}X)$。

尽管如此，关于遗忘的模型检测（即：检查一个结构是否为从μ-句子中遗忘掉某个原子命题的集合的模型）也是困难的。

命题 6.5 (模型检测)**.** *给定一个有限的Kripke 结构$\mathscr{M}$、一个μ-句子φ和原子命题的集合$V \subseteq \mathscr{A}$。有：*

(i) *判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$在*EXPTIME*中；*

(ii) *若φ是一个析取μ-公式，则判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$在*NP$\cap$*co-*NP*中。*

证明. 对于一个μ-公式φ，如果该公式为一个析取μ-公式可在多项式时间内构造一个μ-自动机（也叫模态自动机[98]）A_φ，否则需要指数时间构造其对应的μ-自动机[3]. 这里证明(ii)，(i)可以类似地证明。

令A_φ为一个μ-自动机且满足对任意的Kripke $\mathscr{N}$，A_φ接受$\mathscr{N}$当且仅当$\mathscr{N} \models \varphi$，其中$A_\varphi = (Q, \Sigma_p, \Sigma_r, q_0, \delta, \Omega)$、$\Sigma_p = Var(\varphi)$。不是一般性地，假定$V \subseteq Var(\varphi)$和$V = \{p\}$。因此，构造一个$\mu$-自动机$\mathscr{B} = (Q, \Sigma_p - V, \Sigma_r, q_0, \delta', \Omega)$使得对任意的$q \in Q$和$L \subseteq \Sigma_p - V$，

$$\delta'(q, L) = \delta(q, L) \cup \delta(q, L \cup \{p\}).$$

已有结论表明，对任意的Kripke结构$\mathscr{N}$，$\mathscr{B}$接受$\mathscr{N}$当且仅当存在φ的一个模型$\mathscr{N}'$使得$\mathscr{N} \leftrightarrow_{\{p\}} \mathscr{N}'$[100]，即$\mathscr{B}$对应于和$\mathrm{F}_\mu(\varphi, V)$等价的$\mu$-句子。

在这种情况下，判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$问题被归约到判定是否$\mathscr{B}$接受$\mathscr{M}$的问题。而$\mathscr{B}$从根$r$接受一个Kripke结构$\mathscr{M} = (S, R, L, r)$当且仅当Eve在参数游戏（parity game）$\mathscr{G}(\mathscr{M}, \mathscr{A})$上有一个从$(r, q^0)$开始的赢的策略，这一问题在NP$\cap$co-NP[98]中。 □

[3] 个人通信: Giovanna D'Agostino, 2020.

给定μ-句子φ和ψ，V为原子命题的集合。从知识进化的角度来看，以下推理问题（在命题逻辑里也有研究[124]）是值得探索的：

(i) [Var-weak] φ在ψ的原子命题上的约束至多有ψ强，即$\psi \models \mathrm{F}_\mu(\varphi, V)$；

(ii) [Var-strong] φ在ψ的原子命题上的约束至少有ψ强，即$\mathrm{F}_\mu(\varphi, V) \models \psi$；

(iii) [Var-entailment] φ在$Var(\varphi) \cap Var(\psi)$上的约束比$\psi$在$Var(\varphi) \cap Var(\psi)$上的约束强，即$\mathrm{F}_\mu(\varphi, V) \models \mathrm{F}_\mu(\psi, V)$,

值得注意的是，在(i)和(ii)中，$Var(\varphi) - V = Var(\psi)$，在(iii)中，$V \subseteq (Var(\varphi) \cap Var(\psi))$。

定理 6.5 (Entailment). *给定μ-句子φ和ψ，V为原子命题的集合，则下面的判定问题为*EXPTIME*-完全的。*

(i) 判定$\mathrm{F}_\mu(\varphi, V) \models^? \psi$,

(ii) 判定$\psi \models^? \mathrm{F}_\mu(\varphi, V)$,

(iii) 判定$\mathrm{F}_\mu(\varphi, V) \models^? \mathrm{F}_\mu(\psi, V)$。

证明. 这里给出(i)的证明，其他的结论能够类似地证明。

在EXPTIME中. 令A_φ和A_ψ分别为φ和ψ的μ-自动机，由命题6.5的证明可从A_φ构造$\mathrm{F}_\mu(\varphi, V)$的$\mu$-自动机$\mathscr{B}$。由命题7.3.2[125]可知，可以在线性时间内构造A_ψ的补自动机C，因此可以在线性时间内构造C和$\mathscr{B}$的交自动机$A_{C \cap \mathscr{B}}$。此时，判定问题$\mathrm{F}_\mu(\varphi, V) \models^? \psi$被规约称判定$A_{C \cap \mathscr{B}}$接受的语言是否为空，这一问题时EXPTIME-完全的(定理7.5.1[125])。

因此，判定是否$\mathrm{F}_\mu(\varphi, V) \models^? \psi$是EXPTIME的。

EXPTIME-难: 对任意的μ-句子存在一个等价的μ-自动机，且对任意的μ-自动机存在一个等价的μ-句子[98]。因此，判定问题$\mathrm{F}_\mu(\varphi, V) \models^? \psi$规约成其对应的$\mu$-自动机是否为空的问题。因此，困难属性直接来源于[98,125]。 □

与上面的几个推论问题类似，下面考虑这几个等价问题：Lang等人提出的“var-independence”和“var-equivalence”问题[121]，及“Var-match”问题:

(i) [Var-independence] 公式φ是否独立于原子命题的集合V，即$\mathrm{F}_\mu(\varphi, V) \equiv \varphi$,

(ii) [Var-match] φ在ψ的原子命题上的约束与ψ等价，即$\mathrm{F}_\mu(\varphi, V) \equiv \psi$,

(iii) [Var-equivalence] φ和ψ在原子命题V上的约束是否等价，即$\mathrm{F}_\mu(\varphi, V) \equiv \mathrm{F}_\mu(\psi, V)$。

对于φ和ψ，其原子命题上的约束是一样的。

推论 6.3. 给定μ-句子φ和ψ，V为原子命题的集合。则下面的判定问题为EXPTIME-完全的。

(i) 判定$\psi \equiv^{?} \mathrm{F}_{\mu}(\varphi, V)$，

(ii) 判定$\mathrm{F}_{\mu}(\varphi, V) \equiv^{?} \varphi$，

(iii) 判定$\mathrm{F}_{\mu}(\varphi, V) \equiv^{?} \mathrm{F}_{\mu}(\psi, V)$。

6.5 本章小结

本章介绍了μ-演算下的遗忘。μ-演算是一种比CTL表达能力强的逻辑系统，本章说明了μ-演算下的遗忘是封闭的且与均匀插值是一对对偶概念。此外，本章也研究了μ-演算下的代数属性，表明μ-演算下的遗忘也具有分解性、切片性和同质性。表示性定理给出了遗忘与四条基本公设之间当且仅当的关系。最后，遗忘在模型检测和推理问题上的复杂性结果表明遗忘操作是很困难的，即指数时间的。

第七章　基于遗忘的SNC和知识更新

本章针对第一章提出的问题(2)探讨使用遗忘的方法来解决计算反应式系统*WSC*（*SNC*）和知识更新。首先给出*SNC*（*WSC*）的定义，然后证明任意公式的*SNC*（*WSC*）可以规约到命题下的*SNC*（*WSC*）。并证明*SNC*和*WSC*是一对对偶概念，因此探讨其中一个就足以表示另一个的性质。其次，从遗忘的角度给出了计算*SNC*（*WSC*）的方法。基于此，当给定的反应式系统模型为有限状态时，可以事先将该模型用其特征公式表示出来，然后使用遗忘来计算其*SNC*（*WSC*）。最后，提出使用遗忘定义知识更新的方法。

7.1　引言

遗忘可以用于从本体中抽取摘要、敏感信息的隐蔽、计算逻辑差等，这在相关工作部分有详细介绍。此外，WSC和SNC在智能规划和形式化验证里有重要作用。比如，在2003年，Lin使用WSC和SNC计算规划问题中的后继状态公理。在第二章中详细介绍了命题逻辑和模态逻辑S5下使用遗忘计算WSC（SNC）的方法和算法。本章研究CTL和μ-演算下如何使用遗忘计算反应式系统的SNC（WSC），及使用遗忘定义知识更新。即：

- 对于给定的公式φ、原子命题q和原子命题集合$V \subseteq Var(\varphi)$，若满足$q \in Var(\varphi) - V$，则q在φ和V上的SNC等价于从公式$\varphi \wedge q$中遗忘掉除了V中元素的原子命题后得到结果；WSC做为SNC的一个对偶概念，有相似的结论。而不终止的系统（包括反应是系统你）在本章看作是一个有限的初始结构，根据第五章中的介绍，任意有限的初始结构都能用其特征公式来表示，因而可以使用遗忘的方法来计算其SNC（WSC）。

- 基于遗忘的知识更新通过最小的改变现有知识的模型来使其适应新信息，从这个角度上可以看作是基于模型的更新，且使用这种方法的更新满足katsuno等人提出的八条基本准则[1]。

本章其余部分组织如下：首先，第7.2节给出最强必要条件和最弱充分条件的定义，然后探索这两个概念之间及原子命题和任意公式之间这两个概念的联系，最后给出基于遗忘的计算方法。其次，第7.3节提出基于遗忘的知识更新方法，并证明其与基于偏序关系的方法之间的等价性。最后，总结本章的研究工作。

7.2　最强必要条件和最弱充分条件

这部分介绍如何使用遗忘理论计算最强必要条件和最弱充分条件。直观地说，最强必要条件指最一般的结果（the most general consequence），最弱充分条件指最特殊的诱因（the most specific abduction）。下面给出其形式化定义，本章所说的公式指的是CTL公式，且这章所有的定义和结论都可以平凡地移到μ-句子下。

定义 7.1 (充分和必要条件). 给定两个公式ϕ和ψ，$V \subseteq Var(\phi)$，$q \in Var(\phi) - V$ 和$Var(\psi) \subseteq V$。

- 若$\phi \models q \to \psi$，则称ψ是q在V和ϕ上的必要条件（necessary condition，NC）；
- 若$\phi \models \psi \to q$，则称ψ是q在V和ϕ上的充分条件（sufficient condition，SC）；
- 若ψ是q在V和ϕ上的必要条件，且对于任意的q在V和ϕ上的必要条件ψ'都有$\phi \models \psi \to \psi'$，则称$\psi$是$q$在$V$和$\phi$上的最强必要条件（strongest necessary condition，SNC）；
- 若ψ是q在V和ϕ上的充分条件，且对于任意的q在V和ϕ上的充分条件ψ'都有$\phi \models \psi' \to \psi$，则称$\psi$是$q$在$V$和$\phi$上的最弱充分条件（weakest sufficient condition，WSC）；

从上述定义可以看出，SNC（WSC）是q在V和ϕ上的NC（SC）中最强（最弱）的一个，即：对任意的NC（或SC）ψ'，$\phi \models \text{SNC} \to \psi'$（$\phi \models \psi' \to \text{WSC}$）。此外，如果公式$\psi$和$\psi'$都是$q$在$V$和$\phi$上的SNC（WSC），则$\psi \equiv \psi'$。下面的命题表明SNC和WSC是一对对偶概念。

命题 7.1 (对偶性). 令V、q、φ和ψ为定义*7.1*出现的符号。则ψ是q在V和ϕ上的*SNC*（*WSC*）当且仅当$\neg\psi$是$\neg q$在V和ϕ上的*WSC*（*SNC*）。

证明. （i）假设ψ是q在V和ϕ上的SNC。则$\varphi \models q \to \psi$，因而$\varphi \models \neg\psi \to \neg q$，即$\neg\psi$是$\neg q$在$V$和$\phi$上的SC. 设$\psi'$是$\neg q$在$V$和$\phi$上的SC：$\varphi \models \psi' \to \neg q$。则$\varphi \models q \to \neg\psi'$，即$\neg\psi'$是$q$在$V$和$\phi$上NC。因此，由假设可知$\varphi \models \psi \to \neg\psi'$，所以$\varphi \models \psi' \to \neg\psi$。这证明了$\neg\psi$是$\neg q$在$V$和$\phi$上的WSC。可以类似地证明另一部分。

（ii）WSC的情形可以类似SNC的情形给出证明。 □

在定义7.1中将q替换为任意的公式α，$V \subseteq Var(\alpha) \cup Var(\phi)$，则定义7.1被推广到任意公式的最强必要条件和最弱充分条件的定义。下面的命题表示了原子命题的充分（必要）条件与公式的充分（必要）条件之间的关系：通过计算原子命题的充分（必要）条件来计算公式的充分（必要）条件。

命题 7.2. 给定公式Γ和α，$V \subseteq Var(\alpha) \cup Var(\Gamma)$，$q$是不出现在$\Gamma$和$\alpha$中的原子命题。集合$V$上的公式$\varphi$是$\alpha$在$V$和$\Gamma$上的$SNC$（$WSC$）当且仅当$\varphi$是$q$在$V$和$\Gamma'$上的$SNC$（$WSC$），其中$\Gamma' = \Gamma \cup \{q \leftrightarrow \alpha\}$。

证明. 这里给出SNC部分的证明，WSC部分的证明与其类似。

对于任意的公式β，记$SNC(\varphi,\beta,V,\Gamma)$为“$\varphi$是$\beta$在$V$和$\Gamma$上的SNC”，$NC(\varphi,\beta,V,\Gamma)$为“$\varphi$是$\beta$在$V$和$\Gamma$上的NC”。

($\Rightarrow$) 证明“若$SNC(\varphi,\alpha,V,\Gamma)$，则$SNC(\varphi,q,V,\Gamma')$”。由$SNC(\varphi,\alpha,V,\Gamma)$和$\alpha \equiv q$可知$\Gamma' \models q \rightarrow \varphi$，即：$\varphi$是$q$在$V$和$\Gamma'$上的NC。假设$\varphi'$是$q$在$V$和$\Gamma'$上的任意NC，由于$\alpha \equiv q$和$IR(\alpha \rightarrow \varphi', \{q\})$，因此，$\mathrm{F}_{\mathrm{CTL}}(\Gamma', q) \models \alpha \rightarrow \varphi'$。由引理6.1可知$\Gamma \models \alpha \rightarrow \varphi'$，即：$NC(\varphi', \alpha, V, \Gamma)$。

($\Leftarrow$) 证明“若$SNC(\varphi,q,V,\Gamma')$，则$SNC(\varphi,\alpha,V,\Gamma)$”。由$SNC(\varphi,q,V,\Gamma')$、$IR(\alpha \rightarrow \varphi, \{q\})$和(**PP**)可知$\mathrm{F}_{\mathrm{CTL}}(\Gamma', \{q\}) \models \alpha \rightarrow \varphi$，又由引理6.1可知$\Gamma \models \alpha \rightarrow \varphi$，即：$NC(\varphi, \alpha, V, \Gamma)$。设$\varphi'$是$\alpha$在$V$和$\Gamma$上的任意NC。由$\alpha \equiv q$和$\Gamma' = \Gamma \cup \{q \equiv \alpha\}$可知$\Gamma' \models q \rightarrow \varphi'$，即：$NC(\varphi', q, V, \Gamma')$。又因为$SNC(\varphi,q,V,\Gamma')$、$IR(\varphi \rightarrow \varphi', \{q\})$和(**PP**)，所以$\mathrm{F}_{\mathrm{CTL}}(\Gamma', \{q\}) \models \varphi \rightarrow \varphi'$。由引理6.1可知$\Gamma \models \varphi \rightarrow \varphi'$，因此$SNC(\varphi,\alpha,V,\Gamma)$成立。 □

为了对给定原子命题集合下公式的最弱充分条件有个直观的认识，下面给出一个简单的例子。

例 7.1 (例3.1的延续)**.** 本例来源于图3.1中的初始结构$\mathscr{K}_2$。令$\psi = \mathrm{EX}(s \wedge (\mathrm{EX}se \vee \mathrm{EX}\neg d))$、$\varphi = \mathrm{EX}(s \wedge \mathrm{EX}\neg d)$、$\mathscr{A} = \{d, s, se\}$和$V = \{s, d\}$。下面证明$\varphi$是$\psi$在$V$和$\mathscr{K}_2$上的WSC：

(i) 由已知有$\varphi \models \psi$和$Var(\varphi) \subseteq V$。此外，$(\mathscr{M}, s_0) \models \varphi \wedge \psi$，因此$\mathscr{K}_2 \models \varphi \rightarrow \psi$，即：$\varphi$是$\psi$在$V$和$\mathscr{K}_2$上的SC;

(ii) 这里证明“对任意的ψ在V和$\mathscr{K}_2$上的SC φ'都有$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$”。易知若$\mathscr{K}_2 \not\models \varphi'$，则$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$。假设$\mathscr{K}_2 \models \varphi'$。由$\varphi'$是$\psi$在$V$和$\mathscr{K}_2$上的SC可知$\varphi' \models \mathrm{EX}(s \wedge \phi)$，其中$\phi$是使得$\phi \models \mathrm{EX}se \vee \mathrm{EX}\neg d$成立的公式。又$\mathrm{IR}(\varphi', \overline{V})$，所以$\phi \models \mathrm{EX}\neg d$。因此，$\varphi' \models \varphi$且$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$。

如何使用遗忘计算SNC（WSC）是本章讨论的关键问题。下面首先给出其理论基础，然后再做直观的解释。

定理 7.1. 给定公式φ、原子命题的集合$V \subseteq Var(\varphi)$和原子命题$q \in Var(\varphi) - V$。

(i) $\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$是$q$在$V$和$\varphi$上的$SNC$;

(ii) $\neg\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \neg q, (Var(\varphi) \cup \{q\}) - V)$是$q$在$V$和$\varphi$上的$WSC$。

证明. (i) 令$\mathscr{F}=\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q,(Var(\varphi)\cup\{q\})-V)$。

“NC”部分：由遗忘的定义可知$\varphi\wedge q\models\mathscr{F}$。因此，$\varphi\models q\rightarrow\mathscr{F}$，即：$\mathscr{F}$是$q$在$V$和$\varphi$上的NC。

“SNC”部分：假设ψ'为q在V和φ上的任意NC，即：$\varphi\models q\rightarrow\psi'$。由定理3.1和$IR(\psi',(Var(\varphi)\cup\{q\})-V)$可知，若$\varphi\wedge q\models\psi'$，则$\mathscr{F}\models\psi'$。由假设可知$\varphi\models q\rightarrow\psi'$，所以$\varphi\wedge\mathscr{F}\models\psi'$，因此$\varphi\models\mathscr{F}\rightarrow\psi'$。

由上面两部分可知，$\mathscr{F}$是q在V和φ上的SNC。

(ii) 令$\mathscr{F}=\neg\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge\neg q,(Var(\varphi)\cup\{q\})-V)$。由命题7.1可知，对任意的命题$q'$，$\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q',(Var(\varphi)\cup\{q'\})-V)$是$q'$在$V$和$\varphi$上的SNC，当且仅当$\neg\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q',(Var(\varphi)\cup\{q'\})-V)$是$\neg q'$在$V$和$\varphi$上的WSC。由(i)可知$\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q',(Var(\varphi)\cup\{q'\})-V)$是$q'$在$V$和$\varphi$上的SNC，所以$\neg\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q',(Var(\varphi)\cup\{q'\})-V)$是$\neg q'$在$V$和$\varphi$上的WSC。令$q=\neg q'$，可得$\mathscr{F}$是$q$在$V$和$\varphi$上的WSC。 □

令$\mathscr{F}=\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q,(Var(\varphi)\cup\{q\})-V)$。上面的定理可以直观地解释如下：由遗忘理论的定义可知$\varphi\wedge q\models\beta$，这说明$\mathscr{F}$是$q$在$V$和$\varphi$上的NC；对任意的与$(Var(\varphi)\cup\{q\})-V$无关的公式$\psi$，若$\varphi\wedge q\models\psi$，则由表达性定理可知$\beta\models\psi$。

由第五章可知，任意有限的初始结构都能由一个CTL公式表示，所以由上面的定理自然地就能得到给定有限初始结构下的SNC和WSC。

推论 7.1. *令$\mathscr{K}=(\mathscr{M},s)$为初始结构，其中$\mathscr{M}=(S,R,L,s_0)$为有限原子命题集合$\mathscr{A}$上的初始-Kripke结构，$V\subseteq\mathscr{A}$且$q\in V'=\mathscr{A}-V$。则：*

(i) $\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K})\wedge q,V')$*是$q'$在$V$和$\mathscr{K}$上的SNC；*

(ii) $\neg\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K})\wedge\neg q,V')$*是$q'$在$V$和$\mathscr{K}$上的WSC。*

例 7.2 (例 5.1的延续)**.** 考虑引言中的例子。在例 5.1计算了$\mathscr{K}_2$关于$V=\{d\}$的特征公式为：

$$\begin{aligned}\mathscr{F}_V(\mathscr{M},s_0)\equiv\ &\mathrm{AX}\neg d\wedge d\wedge\\ &\mathrm{AG}(\mathrm{AX}\neg d\wedge d\rightarrow\mathrm{AX}(\mathrm{AX}\neg d\wedge\neg d))\wedge\\ &\mathrm{AG}(\mathrm{AX}\neg d\wedge\neg d\rightarrow\mathrm{AX}(\mathrm{AX}d\wedge\neg d))\wedge\\ &\mathrm{AG}(\mathrm{AX}d\wedge\neg d\rightarrow\mathrm{AX}(\mathrm{AX}\neg d\wedge d)).\end{aligned}$$

7.3 知识更新

本小节介绍遗忘理论的另一个应用：知识更新（Knowledge update）。具体说来，本节将使用遗忘理论定义知识更新，使得用这种方式定义的知识更新满足下面由Katsuno和Mendelzon的基本条件：

- (U1) $\Gamma \diamond \varphi \models \varphi$；
- (U2) 若$\Gamma \models \varphi$，则$\Gamma \diamond \varphi \equiv \Gamma$；
- (U3) 若Γ和φ都是可满足的，则$\Gamma \diamond \varphi$是可满足的；
- (U4) 若$\Gamma_1 \equiv \Gamma_2$且$\varphi_1 \equiv \varphi_2$，则$\Gamma_1 \diamond \varphi_1 \equiv \Gamma_2 \diamond \varphi_2$；
- (U5) $(\Gamma \diamond \varphi) \wedge \psi \models \Gamma \diamond (\varphi \wedge \psi)$；
- (U6) 若$\Gamma \diamond \varphi \models \psi$且$\Gamma \diamond \psi \models \varphi$，则$\Gamma \diamond \varphi \equiv \Gamma \diamond \psi$；
- (U7) 若Γ有唯一一个模型，则$(\Gamma \diamond \varphi) \wedge (\Gamma \diamond \psi) \models \Gamma \diamond (\varphi \vee \psi)$；
- (U8) $(\Gamma_1 \vee \Gamma_2) \diamond \varphi / (\Gamma_1 \diamond \varphi) \vee (\Gamma_2 \diamond \varphi)$。

其中，$\diamond$为知识更新操作，$\varphi \diamond \psi$表示用ψ更新φ得到的结果。

本小节假设所有的初始结构都是有限的，即：状态来源于有限的状态空间且$\mathscr{A}$为有限的原子命题的集合。下面定理显然成立：

定理 7.2. 给定公式φ和原子命题集$V \subseteq \mathscr{A}$。存在一个公式ψ使得：

$$(\mathscr{M}, s_0) \models \psi \text{ 当且仅当存在} (\mathscr{M}', s_0') \in Mod(\varphi) \text{ 使得} (\mathscr{M}, s_0) \leftrightarrow_V (\mathscr{M}', s_0')$$

其中$(\mathscr{M}, s_0)$和$(\mathscr{M}', s_0')$都是有限的初始结构。

证明. 令$\psi = \mathrm{F}_\mu(\varphi, V)$（或$\psi = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$）。由遗忘的定义可知，对任意的$\mathscr{K} \models \psi$存在一个$\mathscr{K}' \models \varphi$使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，且对每一个$\mathscr{K}' \in Mod(\varphi)$都有$\mathscr{K}' \models \psi$。此时，容易证明对任意的有限初始结构$\mathscr{K}$，若$\mathscr{K} \models \psi$，则存在一个$\mathscr{K}'$使得$\mathscr{K}' \models \varphi$且$\mathscr{K} \leftrightarrow_V \mathscr{K}'$。

此外，对任意的$\mathscr{K}' \in Mod(\varphi)$，由(**W**)可知存在$\mathscr{K}' \models \psi$。又$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，所以$\mathscr{M} \models \psi$。 □

定理7.2表明模型结构被限制到有限初始结构下的μ-演算下遗忘理论也是封闭的。此外，由第五章可知，任意$\mathscr{A}$上的有限初始结构$\mathscr{K}$都能用一个CTL公式——特征公式$\mathscr{F}_{\mathscr{A}}(\mathscr{K})$来表示，此公式也是$\mu$-句子。

给定公式φ和ψ，$V_{min} \subseteq \mathscr{A}$为使得$\mathrm{F}_{\mathrm{CTL}}(\varphi, V_{min}) \wedge \psi$可满足的极小子集。此外，记

$$\bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}) \wedge \psi)$$

为所有$\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}) \wedge \psi$的模型集合的并集。此时，可如下定义$\mu$-演算下的知识更新操作：

定义 7.2. 给定公式Γ和φ。知识更新操作$\diamond_{CTL}$定义如下：

$$Mod(\Gamma \diamond_{CTL} \varphi) = \bigcup_{\mathscr{K} \in Mod(\Gamma)} \bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}) \wedge \varphi),$$

其中，$\mathscr{F}_{\mathscr{A}}(\mathscr{K})$是$\mathscr{K}$在$\mathscr{A}$上的特征公式，$V_{min} \subseteq \mathscr{A}$是使得$\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min})$可满足的极小子集。

从直观上来说，$\Gamma \diamond_{\mathrm{CTL}} \varphi$表示通过极小化改变$\Gamma$的模型到$\varphi$的模型来更新$\Gamma$。换句话说，定义7.2通过极小化改变$\Gamma$的每个模型，使得该模型能够满足$\varphi$来更新原有的知识$\Gamma$。从这个角度看，这样定义的知识更新是一种基于模型的知识更新方法。

此外，μ-演算下的知识更新也可以通过像命题逻辑里的那样来定义：令I，J_1和J_2为三个解释，即：原子命题的集合；则J_1比J_2更接近I（记为：$J_1 \leq_{I,pma} J_2$）当且仅当$\mathrm{Diff}(I, J_1) \subseteq \mathrm{Diff}(I, J_2)$，其中$\mathrm{Diff}(X, Y) = \{p \in \mathscr{A} \mid X(p) \neq Y(p)\}$。那么命题逻辑知识更新——用$\psi$更新$\Gamma$，即为$\psi$的关于偏序关系$\leq_{I,pma}$的所有极小模型的集合（$I$是$\Gamma$的模型），即：

$$Mod(\Gamma \diamond_{pma} \psi) = \bigcup_{I \in Mod(\Gamma)} Min(Mod(\psi), \leq_{I,pma}).$$

其中，$Min(Mod(\psi), \leq_{I,pma})$是$\psi$的关于偏序关系$\leq_{I,pma}$的极小模型的集合。

类似地，这里定理有限初始结构之间关于另一个初始结构的偏序关系。

定义 7.3. 给定三个有限初始结构$\mathscr{K}$、$\mathscr{K}_1$和$\mathscr{K}_2$，$\mathscr{K}_1$比$\mathscr{K}_2$更接近$\mathscr{K}$（记为$\mathscr{K}_1 \leq_{\mathscr{K}} \mathscr{K}_2$）当且仅当对任意使得$\mathscr{K}_2 \leftrightarrow_{V_2} \mathscr{K}$成立的$V_2 \subseteq \mathscr{A}$都存在一个$V_1 \subseteq V_2$使得$\mathscr{K}_1 \leftrightarrow_{V_1} \mathscr{K}$。$\mathscr{K}_1 <_{\mathscr{K}} \mathscr{K}_2$当且仅当$\mathscr{K}_1 \leq_{\mathscr{K}} \mathscr{K}_2$且$\mathscr{K}_2 \not\leq_{\mathscr{K}} \mathscr{K}_1$。

给定有限初始结构的集合M和有限初始结构$\mathscr{K}$，用$Min(M, \leq_{\mathscr{K}})$表示$M$中关于偏序关系$\leq_{\mathscr{K}}$的极小有限初始结构的集合。则$\leq_{\mathscr{K}}$与知识更新操作$\diamond_{\mathrm{CTL}}$有如下关系。

定理 7.3. 给定μ-句子Γ和φ，则：

$$Mod(\Gamma \diamond_{CTL} \varphi) = \bigcup_{\mathscr{K} \in Mod(\Gamma)} Min(Mod(\varphi), \leq_{\mathscr{K}}).$$

证明. 对每一个初始结构$\mathscr{K}' \in Mod(\Gamma \diamond_{\mathrm{CTL}} \varphi)$，这里证明存在一个$\mathscr{K} \in Mod(\Gamma)$使得$\mathscr{K}' \in Min(Mod(\varphi), \leq_{\mathscr{K}})$。由定义7.2可知，存在$\mathscr{K} \in Mod(\Gamma)$使得$\mathscr{K}' \in Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}) \wedge \varphi)$。此外，存在一个特殊的$V' \subseteq \mathscr{A}$（即：$V' = V_{min}$）使得$\mathscr{K}' \leftrightarrow_{V'} \mathscr{K}$和$\mathscr{K}' \in Mod(\varphi)$。因为$V'$是使得$\mathscr{K}' \leftrightarrow_{V'} \mathscr{K}$成立的极小子集，因此对任意使得$\mathscr{K}'' \leftrightarrow_{V_{min}} \mathscr{K}$成立的$\varphi$的模型$\mathscr{K}''$，由遗忘理和特征公式论的定义可知$\mathscr{K}' \leq_{\mathscr{K}} \mathscr{K}''$。因此，$\mathscr{K}' \in Min(Mod(\varphi), \leq_{\mathscr{K}})$。

对每一个初始结构$\mathscr{K}' \in \bigcup_{\mathscr{K}\in Mod(\Gamma)} Min(Mod(\varphi), \leq_{\mathscr{K}})$，存在$\mathscr{K} \in Mod(\Gamma)$使得$\mathscr{K}' \in Min(Mod(\varphi), \leq_{\mathscr{K}})$。设$V_{min}$是使得$\mathscr{K}' \leftrightarrow_{V_{min}} \mathscr{K}$成立的极小子集。根据$\leq_{\mathscr{K}}$的定义可知，不存在其他$\mathscr{K}'' \in Mod(\varphi)$使得$\mathscr{K}'' \leftrightarrow_{V'} \mathscr{K}$且$V' \subset V_{min}$。因而$\mathscr{K}' \in Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}) \wedge \varphi)$，所以$\mathscr{K}' \in Mod(\Gamma \diamond_{\mathrm{CTL}} \varphi)$。 □

从定理7.3可以看出，通过遗忘理论定义的知识更新操作与通过有限初始结构间的偏序关系定义的知识更新一致，且通过遗忘理论定义的知识更新操作满足Katsuno和Mendelzon提出的八条基本条件。

定理 7.4. *知识更新操作$\diamond_{CTL}$满足Katsuno和Mendelzon提出的基本条件(U1)-(U8)。*

证明. (U1). 由定理7.3可知$Mod(\Gamma \diamond_{\mathrm{CTL}} \varphi) \subseteq Mod(\varphi)$，因此$\Gamma \diamond_{\mathrm{CTL}} \varphi \models \varphi$。

(U2). 首先证明$\Gamma \diamond_{\mathrm{CTL}} \varphi \models \Gamma$。对$\Gamma \diamond_{\mathrm{CTL}} \varphi$的任意一个模型$\mathscr{K}$，存在一个$\mathscr{K}_1 \in Mod(\Gamma)$和$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。又$\Gamma \models \varphi$，因此$V_{min} = \emptyset$，即$\mathscr{K} \models \Gamma$。类似地，对$\Gamma$的每一个模型$\mathscr{K}$，存在一个$\mathscr{K}_1 \in Mod(\Gamma \diamond_{\mathrm{CTL}} \varphi)$和一个$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。又$\Gamma \models \varphi$，因此$V_{min} = \emptyset$。所以，$\Gamma \models \Gamma \diamond_{\mathrm{CTL}} \varphi$。

容易证明$\diamond_{\mathrm{CTL}}$满足(U3)和(U4)。

(U5). 对$(\Gamma \diamond_{\mathrm{CTL}} \varphi) \wedge \psi$的每一个模型$\mathscr{K}$，存在一个$\mathscr{K}_1 \in Mod(\Gamma)$和一个$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。此外，可知$\mathscr{K} \models \varphi \wedge \psi$，因此，$\mathscr{K} \models \Gamma \diamond_{\mathrm{CTL}} (\varphi \wedge \psi)$。

(U6). 这里给出$\Gamma \diamond_{\mathrm{CTL}} \varphi \models \Gamma \diamond_{\mathrm{CTL}} \psi$的证明，另一个方向可以类似地证明。对$\Gamma \diamond_{\mathrm{CTL}} \varphi$的每一个模型$\mathscr{K}$，$\mathscr{K}$也是$\psi$的模型，且存在$\mathscr{K}_1 \in Mod(\Gamma)$和$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。因此，$\mathscr{K}$是$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V_{min}) \wedge \psi$的模型，也即是$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V_{min}) \wedge \psi$可满足的。设$V \subset V_{min}$是使得$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V) \wedge \psi$可满足的原子命题的集合，由$\Gamma \diamond_{\mathrm{CTL}} \psi \models \varphi$可知$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V) \wedge \varphi$是可满足的，这与$V_{min}$是使得$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V_{min}) \wedge \varphi$可满足的极小子集。因此，$V_{min}$是使得$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}_1), V_{min}) \wedge \psi$可满足的极小子集，所以，$\mathscr{K}$是$\Gamma \diamond_{\mathrm{CTL}} \psi$的模型。

(U7). 设Γ有唯一的模型$\mathscr{K}$。对每一个$\mathscr{K}_1 \in Mod((\Gamma \diamond_{\mathrm{CTL}} \varphi) \wedge (\Gamma \diamond_{\mathrm{CTL}} \psi))$，存在两个关于$\leq_{\mathscr{K}_1}$的极小子集$V_1$和$V_2$使得$\mathscr{K} \leftrightarrow_{V_1} \mathscr{K}_1$和$\mathscr{K} \leftrightarrow_{V_2} \mathscr{K}_1$成立，即：$\mathscr{K}_1$是$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_1) \wedge \varphi$和$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_2) \wedge \psi$的模型。因此，$\mathscr{K}_1 \leftrightarrow_{V_1 \cap V_2} \mathscr{K}$，即$\mathscr{K}_1$是$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_1 \cap V_2)$的模型。所以，$V_1 = V_2$，否则$V_1$（或$V_2$）不是关于$\leq_{\mathscr{K}_1}$的极小子集。此外，$\mathscr{K}_1$也是$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_1) \wedge (\varphi \vee \psi)$的模型。

设$V_3 \subset V_1$是使得$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_3) \wedge (\varphi \vee \psi)$可满足的原子命题集合，则有$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_3) \wedge \varphi$或$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_3) \wedge \psi$是可满足的。不失一般性地，设$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_3) \wedge \varphi$是可满足的，则$V_1$不是关于$\leq_{\mathscr{K}_1}$的极小子集，与前面的描述矛盾。因此，$V_1$是使得$\mathrm{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_1) \wedge (\varphi \vee \psi)$可满足的极小子集。所以，$\mathscr{K}_1$是$\Gamma \diamond_{\mathrm{CTL}} (\varphi \vee \psi)$的模型。

(U8) 对每一个$\mathscr{K} \in Mod((\Gamma_1 \vee \Gamma_2) \diamond_{\mathrm{CTL}} \varphi)$都存在一个$\mathscr{K}_1 \in Mod(\Gamma_1)$（或$\mathscr{K}_1 \in Mod(\Gamma_2)$）和一个$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。因此，$\mathscr{K} \models (\Gamma_1 \diamond_{\mathrm{CTL}} \varphi) \vee (\Gamma_2 \diamond_{\mathrm{CTL}} \varphi)$成立。

类似地，对每一个$\mathscr{K} \in Mod((\Gamma_1 \diamond_{\mathrm{CTL}} \varphi) \vee (\Gamma_2 \diamond_{\mathrm{CTL}} \varphi))$都存在一个$\mathscr{K}_1 \in Mod(\Gamma_1)$（或$\mathscr{K}_1 \in Mod(\Gamma_2)$）和一个$V_{min}$使得$\mathscr{K} \leftrightarrow_{V_{min}} \mathscr{K}_1$。因此，$\mathscr{K} \models (\Gamma_1 \vee \Gamma_2) \diamond_{\mathrm{CTL}} \varphi$成立。　□

7.4　本章小结

本章讨论了遗忘在计算最强必要条件（最弱充分条件）和定义知识更新。特别地，本章首先给出了SNC（WSC）的定义，表明SNC和WSC是一对对偶概念，因而只要知道其一就能知道另一个。其次，结论表明任意公式的SNC（WSC）可以转换成原子命题的SNC（WSC）来计算，并给出使用遗忘的计算SNC（WSC）的定理。最后，分别给出使用遗忘和偏序关系的方法定义知识更新，并证明了这两种定义等价且满足Katsuno等人提出的八条准则。

第八章　实验结果

本节给出所提出的基于归结的遗忘计算的实验结果，并分析实验结果。第四章提出的算法*4.2*用*Prolog*语言实现，并在*Linux*服务器上进行了实验，该服务器是具有*8*个*Intel*核和*32GB*内存的*i7CPU*，其锁频和主频分别为*4770 K*和*3.50 GHz*。每次计算的时间限制到*1200*秒以内。实验分为两个部分：*(1)* 在随机数据集和标准数据集上的遗忘；*(2)* 在随机数据集上命题逻辑公式和*CTL*公式的*SNC*计算。所有的实验数据和实验结果都可以从网上获取[1]。

此外，在这部分*3-CNF*公式φ的长度（记为$|\varphi|$）表示φ中子句的个数。

8.1　算法实现

8.2　遗忘实验分析

这部分的实验数据分为两组：一组来源于标准数据集，一组是随机生成的数据。标准数据集来源于CTL-RP [2]。但是由于数据集里的大部分公式是不可满足的，这种情形下遗忘的结果总是为$\perp$。因此，这里对数据集进行了简单的处理：从标准数据集里抽取了“sample01”文件中的s001.ctl、s002.ctl、s003.ctl和s004.ctl文件，并从这些公式里取前面的两个子公式的合取构成新的公式，分别称为s001、s002、s003和s004。此外，从s001.ctl中取前三个子公式的合取构成新的公式s001-3。

计算CTL-forget(φ, V)所使用的CPU时间（单位：秒(s)，不指出时也默认为秒）如表8.1所示，其中$\varphi \in \{s001, s002, s003, s004, s001\text{-}3\}$，$|V| \in \{1,2,3,4\}$。从中可以看出公式长度越长、被遗忘的原子命题个数越多，则计算所需要的时间越长。

[1] https://github.com/fengrenyan/forgetting-in-CTL/tree/main/Appendix
[2] https://sourceforge.net/projects/ctlrp/

表 8.1: 计算CTL-forget(φ, V)所使用的CPU时间（单位：秒(s)）

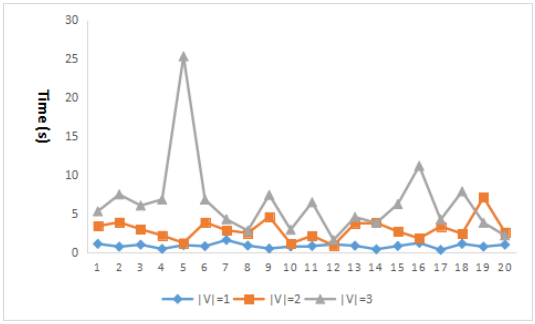
φ \ $\lvert V\rvert$	1	2	3	4
s001	0.0505	0.1053	0.2259	0.3680
s002	0.3645	1.0416	5.6372	10.0184
s003	97.5341	71.5396	190.1157	423.5793
s004	77.5086	77.4246	101.1284	118.7461
s001-3	681.2883	613.1859	1617.047	2356.949

除了上述标准数据集中的公式，我们也做了具有以下形式的公式的遗忘实验:

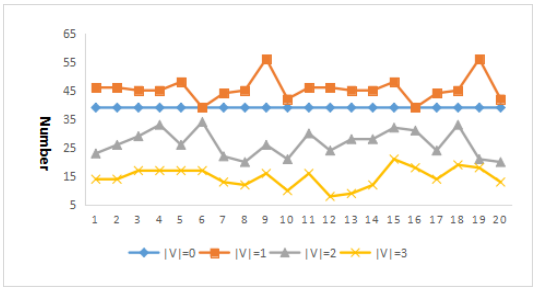
$$\varphi = \varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$$

其中φ_i（$i = 1,2,3$）是随机产生的定义在原子命题集合$\mathscr{A}$上的3-CNF公式，且$|\varphi_1| = |\varphi_2| = |\varphi_3|$、$|\mathscr{A}| = 4$。这里做了六组计算$\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$的实验，即$|\varphi_i| \in \{12,16\}$和$|V| \in \{1,2,3\}$的组合，每一组有二十个公式。

$|\varphi_i| = 12$时的实验结果如图8.1所示，图8.1(a) 展示了计算遗忘所需要的时间，图8.1(b)展示了在计算过程“移除原子命题”后$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数。其中$x$轴表示第几个公式，$y$轴分别表示时间和数量。从图8.1里面可以看出，需要遗忘的原子命题个数越多，所用时间越长且在“移除原子命题”后剩余的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数越少。当$|\varphi_i| = 16$时的实验结果如图8.2所示，其与$|\varphi_i| = 12$时有相似的结果。

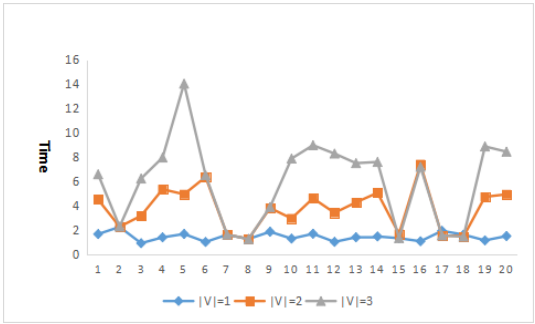


(a) 计算遗忘需要的CUP时间

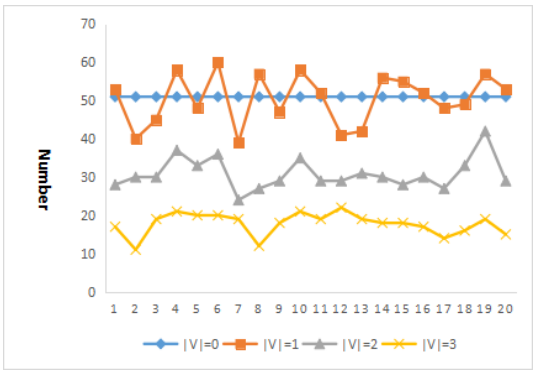


(b) $\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数

图 8.1: 计算CTL-forget(φ, V)使用的时间和在“移除原子命题”步骤后$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数，其中$\varphi_i = 12$。



(a) 计算遗忘需要的CUP时间



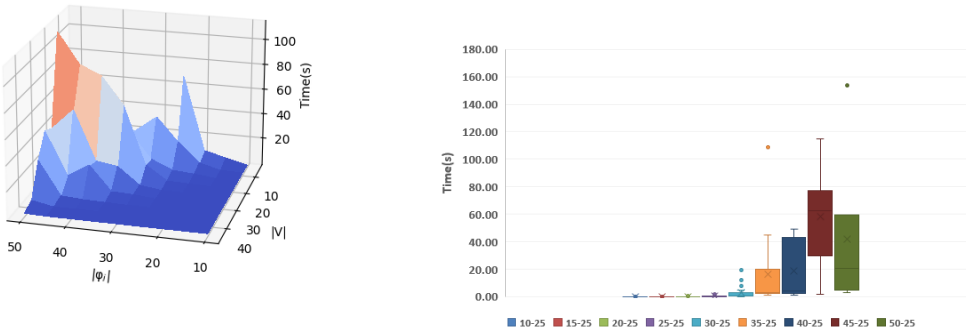
(b) $\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数

图 8.2: 计算CTL-forget(φ, V)使用的时间和在“移除原子命题”步骤后$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数，其中$\varphi_i = 16$。

8.3 SNC计算结果分析

这部分实验分析使用基于遗忘的方法计算CTL公式的SNC，分为两组实验：分别计算经典命题公式和CTL公式的SNC，即：计算q在V和$\varphi \wedge q$上的SNC（$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, Var(\varphi) - V \cup \{q\})$），其中$V \subseteq Var(\varphi)$、$q \in Var(\varphi \wedge q) - V$。这些公式$\varphi$都是随机生成的定义在原子命题集合$\mathscr{A}$上的公式、$V$也是在计算过程中随机生成的、$q \notin Var(\varphi)$是一个固定的原子命题且$|\mathscr{A}| = 50$。

首先测试随机3-CNF命题公式。令$|V|$的取值范围为$\{5, 10, \ldots, 40, 45\}$，3-CNF公式的子句个数$nc$范围为$\{10, 15, \ldots, 45, 50\}$。在每种情形当中，计算20个随机实例$(\varphi, q, V)$：$\varphi$为$\mathscr{A}$上的公式，且$V \subseteq Var(\varphi)$。计算SNC的平均CPU时间如图8.3所示。



(a) 平均CPU时间(s)　　(b) $|V| = 25$时所使用CPU时间箱线图

图 8.3: 计算3-CNF公式SNC的CPU时间

从图8.3(a)可看出，随着$|\varphi|$增大或$|V|$的减小时间消耗越大。直观地说，越大的$|\varphi|$或者越小的$|V|$意味着$\mathrm{F}_{\mathrm{CTL}}(\varphi, \overline{V})$更难计算。这与上一小节中的结论相符合。图8.3(b)展示了当$|V| = 25$、$nc \in \{10, 15, \ldots, 45, 50\}$时20个随机实例的箱线图。这同样证明了$nc$越大SNC越难计算。

其次，测试具有如下形式的CTL公式的SNC的计算：

$$\varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$$

其中φ_i（$i = 1, 2, 3$）为随机产生的定义在$\mathscr{A}$上的3-CNF公式，且满足$|\varphi_1| = |\varphi_2| = |\varphi_3|$。在这种情形下，每个实例$(\varphi, q, V)$是随机产生的，其中$\varphi = \varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$、$V \subseteq Var(\varphi)$、$|\varphi| \in \{5, 6, \ldots, 13, 14\}$、且$|V| \in \{15, 16, \ldots, 23, 24\}$。值得注意的是在实例$(\varphi, q, V)$中，$q$可能没有在$V$和$\varphi \wedge q$上的SNC。

图8.4(a)展示了每种情形计算40个实例的SNC的平均CPU时间。与命题公式的情形相似，越大的$|\varphi|$或者越小的$|V|$意味着$\mathrm{F}_{\mathrm{CTL}}(\varphi, \overline{V})$更难计算。此外，图8.4(b)展示了每种

情形下40个实例中SNC存在的占比，即：$|\varphi|$越小或者$|V|$越小则SNC存在的占比越大。特别地，当$|\varphi_i| = 5$且$|V| = 16$时，SNC存在的占比为80%。

(a) 平均CPU时间(s)　　(b) 存在SNC的公式占比(%)

图 8.4: 计算CTLSNC的平均时间和存在SNC的公式占比.

综上所述，算法4.2大多数情况下能计算出SNC（WSC），且当需要遗忘的原子个数很少或公式长度较小的时候计算效率很高。

8.4　本章小结

本章从两个角度来评估第四章提出的算法的实现。从标准数据集中获取的公式和随机产生的公式下遗忘的计算对需要遗忘的原子命题的个数和公式的长度都很敏感：要遗忘的原子命题个数越多或公式越长，计算效率越低。此外，关于计算SNC的实验表明本文提出的算法在实验给出的数据集中的大部分情况下是能计算出SNC的。三维图和箱线图都表示计算SNC体现了跟计算遗忘一样的规律，这与SNC是由遗忘来计算是一致的。

第九章　总结与展望

本章首先总结文中针对*CTL*和μ-演算下的遗忘理论做出的研究工作，概括文中使用的方法及取得的研究成果。其次，讨论分析本文工作中存在的不足之处，并基于此对本文的后续研究内容进行了展望。

9.1　工作总结

随着计算机系统日益变得复杂，描述系统规范的语言也变得越来越复杂。随着信息的更新，系统的规范也随着改变，因而急需有效的方法来提取相关原子命题下的知识。遗忘是一种知识提取的方法，本文从语法和语义的角度探索了广泛应用于并行系统的CTL下的遗忘。此外，也探索了表达能力更强的μ-演算下的遗忘。具体地，本文的主要工作总结如下：

按照1.3节的方式重新总结下面的内容。

(1) 从语义的角度给出了CTL和μ-演算下的遗忘的定义，并探索了遗忘的基本性质。首先，给出了一般化的互模拟——在给定集合上的互模拟的定义。互模拟是描述两个系统行为的概念，若两个系统具有互模拟关系，则这两个系统具有相同的行为，即：在对应的状态上对相同的原子命题有相同的解释。公式刻画了其代表的模型的行为，从公式中遗忘掉给定的原子命题应该不影响该公式在其他原子命题在其模型上的行为表现，也即是新得到的公式的模型除了被遗忘的原子命题之外与原公式的模型有互模拟关系，即给定集合上的互模拟。基于此，遗忘的概念由给定集合上的互模拟给出。特别地，对于给定的原子命题集合V，若两个系统具有V-互模拟关系，则对于与V无关的公式，这两个系统同时满足或不满足该公式。其次，指出CTL下的遗忘是不封闭的，而μ-演算下的遗忘是封闭的，即存在CTL公式的遗忘结果是不可用CTL公式来表示的。最后，探讨遗忘的代数属性，指出遗忘具有模块属性、交换属性和同质属性，这为计算遗忘提供了方便。

(2) 提出并实现了基于归结的CTL下遗忘的计算方法。本文采用了Zhang等人的归结系统计算CTL下的遗忘，并给出计算遗忘的算法。具体地，该算法以CTL公式和原子命题集合V为输入，输出一个CTL公式。该算法主要包括四个步骤：转换CTL公式为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合、计算归结结果、移除包含$V$中元素的子句及将得到子句集合转换为CTL公式。为此，本文给出了如何消除转换过程中引入的索引的方法，即移除掉索引后保持公式之间的互模拟等价。此外，为了消除转换过程中引入的新原子命题，提出了一般的Ackermann引理。

(3) 探索了约束情形下遗忘的封闭性。CTL公式具有小模型性质，也即是对给定

的CTL公式，若该公式是可满足的，则其存在一个该公式大小指数的一个模型。因而本文讨论这种具有约束大小的公式，并讨论有限状态空间下的CTL的遗忘。在这种情形下，CTL公式的模型的个数是有限的，CTL的遗忘是封闭的，且其遗忘的结果等于其所有模型的特征公式的吸取。此外，还给出了这种情形下计算遗忘的算法。尽管该算法从效率上来说是低效的，但是它是一种可靠且完备的方法。

(4) 使用遗忘计算SNC（WSC）和定义知识更新。对于给定的公式和原子命题集合，若遗忘掉除这些原子命题之外的原子命题的结果可用CTL公式表示，则该结果一定是SNC（WSC），即：CTL下可以用遗忘来计算SNC（WSC）。在约束情形下SNC（WSC）一定可以用遗忘方法来计算，因为这种情形下遗忘是封闭的。此外，当给定有限反应式系统的情形下，可以将该系统表示成特征公式，然后再使用遗忘来计算。最后，提出了两种定义知识更新的方法：基于遗忘的定义和基于模型的偏序关系的定义，并证明这两种方法定义的知识更新是等价的且满足Katsuno等人提出的八条基本准则。

(5) 实现了第四章提出的基于归结的算法，并做了实验，得到“遗忘的原子命题个数越少，计算效率越高”的结论。

9.2　研究展望

本文探讨了对系统设计至关重要的抽取信息的方法——遗忘，并使用该方法计算SNC（WSC）和定义知识更新。文中指出CTL下的遗忘不是封闭的，并提出了基于归结的计算遗忘的方法。下面问题值得将来继续研究：

(1) 探索CTL中遗忘封闭的子类。在系统规范描述中，有时候用到的公式不一定很复杂，也不一定需要用到所有的时序词。为此，探索简单并足够表达某些性质的CTL公式的子类，在这些子类下遗忘是容易计算的。

(2) 探索如何使用计算出来的WSC（SNC）更新（修改）系统模型。特别地，当一个系统$\mathscr{M}$不满足规范ϕ时，可以计算在某个命题集合V上的最弱充分条件ψ使得$\mathscr{M}$满足，即$\mathscr{M} \models \psi \rightarrow \phi$且$\psi$是$V$上的公式。这时，如何使用获得的WSC来更新系统得到新系统$\mathscr{M}'$使其满足ϕ也是重要的。

(3) 尽管CTL_{AF}段下遗忘的模型检测和推理判定问题的复杂性已经给出，但是更多关于遗忘的问题的复杂性应该被探索：整个CTL下的遗忘的模型检测和推理判定问题的复杂性。

(4) 研究算法 4.2的完备性。尽管在第四中给出了算法正确性的分析，但是并没有讨论算法的完备性。在将来的工作中可以研究算法 4.2的完备性，并探索其他可靠且完备的算法。

参考文献

[1] KATSUNO H, MENDELZON A O. On the difference between updating a knowledge base and revising it[C]//ALLEN J F, FIKES R, SANDEWALL E. Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991. [S.l.]: Morgan Kaufmann, 1991: 387-394.

[2] LAM W K. 硬件设计验证–基于模拟与形式的方法[M]. 北京: 机械工业出版社, 2007.

[3] 吕毅. 时序逻辑电路的形式验证方法研究[D]. 北京: 中国科学院研究生院(计算技术研究所), 2000.

[4] JANICK B夏宇闻. System Verilog验证方法学[M]. 北京: 北京航空航天大学出版社学, 2007.

[5] 袁志斌. 软件开发的形式化工程方法[M]. 北京: 清华大学出版社, 2008.

[6] 古天龙. 软件开发的形式化方法[M]. 北京: 高等教育出版社, 2005.

[7] FOX A C J. Formal specification and verification of ARM6[C/OL]//BASIN D A, WOLFF B. Lecture Notes in Computer Science: volume 2758 Theorem Proving in Higher Order Logics, 16th International Conference, TPHOLs 2003, Rom, Italy, September 8-12, 2003, Proceedings. Springer, 2003: 25-40. https://doi.org/10.1007/10930755_2.

[8] DAUM M, SCHIRMER N, SCHMIDT M. Implementation correctness of a real-time operating system[C/OL]//HUNG D V, KRISHNAN P. Seventh IEEE International Conference on Software Engineering and Formal Methods, SEFM 2009, Hanoi, Vietnam, 23-27 November 2009. IEEE Computer Society, 2009: 23-32. https://doi.org/10.1109/SEFM.2009.14.

[9] ROBINSON J A. A machine-oriented logic based on the resolution principle[J/OL]. Journal of the ACM (JACM), 1965, 12(1):23-41. http://doi.acm.org/10.1145/321250.321253.

[10] HUGHES G E, CRESSWELL M J, CRESSWELL M M. A new introduction to modal logic[M]. [S.l.]: Psychology Press, 1996.

[11] HOARE C A R. An axiomatic basis for computer programming[J/OL]. Commun. ACM, 1969, 12(10):576-580. https://doi.org/10.1145/363235.363259.

[12] HAREL D. Lecture notes in computer science: volume 68 first-order dynamic logic [M/OL]. Springer, 1979. https://doi.org/10.1007/3-540-09237-4.

[13] REYNOLDS J C. Separation logic: A logic for shared mutable data structures[C/OL]// 17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings. IEEE Computer Society, 2002: 55-74. https://doi.org/10.1109/LICS.2002.1029817.

[14] LIN F. A formalization of programs in first-order logic with a discrete linear order [J/OL]. Artificial Intelligence, 2016, 235:1-25. https://doi.org/10.1016/j.artint.2016.01.014.

[15] CLARKE E M. The birth of model checking[C/OL]//GRUMBERG O, VEITH H. Lecture Notes in Computer Science: volume 5000 25 Years of Model Checking - History, Achievements, Perspectives. Springer, 2008: 1-26. https://doi.org/10.1007/978-3-540-69850-0_1.

[16] CLARKE E M, EMERSON E A. Design and synthesis of synchronization skeletons using branching time temporal logic[C]//Workshop on Logic of Programs. [S.l.]: Springer, 1981: 52-71.

[17] BIERE A, CIMATTI A, CLARKE E M, et al. Bounded model checking[J/OL]. Adv. Comput., 2003, 58:117-148. https://doi.org/10.1016/S0065-2458(03)58003-2.

[18] BURCH J R, CLARKE E M, MCMILLAN K L, et al. Symbolic model checking: 1020 states and beyond[J]. Information and computation, 1992, 98(2):142-170.

[19] SCHNEIDER K. Texts in theoretical computer science. an EATCS series: Verification of reactive systems - formal methods and algorithms[M/OL]. Springer, 2004. https://doi.org/10.1007/978-3-662-10778-2.

[20] BAIER C, KATOEN J. Principles of model checking[M]. [S.l.]: MIT Press, 2008.

[21] LEGATO W J. A weakest precondition model for assembly language programs[M]. [S.l.]: April, 2002.

[22] LEINO K R M. Efficient weakest preconditions[J/OL]. Information Processing Letters, 2005, 93(6):281-288. https://doi.org/10.1016/j.ipl.2004.10.015.

[23] DAILLER S, HAUZAR D, MARCHÉ C, et al. Instrumenting a weakest precondition calculus for counterexample generation[J/OL]. J. Log. Algebraic Methods Program., 2018, 99:97-113. https://doi.org/10.1016/j.jlamp.2018.05.003.

[24] WOODCOCK J, MORGAN C. Refinement of state-based concurrent systems[C/OL]// BJØRNER D, HOARE C A R, LANGMAACK H. Lecture Notes in Computer Science: volume 428 VDM '90, VDM and Z - Formal Methods in Software Development, Third International Symposium of VDM Europe, Kiel, FRG, April 17-21, 1990, Proceedings. 1990: 340-351. https://doi.org/10.1007/3-540-52513-0_18.

[25] 简云松. 一个目标驱动的运行时需求管理框架的研究与实现[D]. 北京: 清华大学, 2012.

[26] HAREL D, PNUELI A. On the development of reactive systems[C/OL]//APT K R. NATO ASI Series: volume 13 Logics and Models of Concurrent Systems - Conference proceedings, Colle-sur-Loup (near Nice), France, 8-19 October 1984. Springer, 1984: 477-498. https://doi.org/10.1007/978-3-642-82453-1_17.

[27] PNUELI A. Applications of temporal logic to the specification and verification of reactive systems: A survey of current trends[M/OL]//DE BAKKER J W, DE ROEVER W P, ROZENBERG G. Lecture Notes in Computer Science: volume 224 Current Trends in Concurrency, Overviews and Tutorials. Springer, 1986: 510-584. https://doi.org/10.1007/BFb0027047.

[28] 王娟娟, 乔颖, 熊金泉, 等. 多核环境下基于图模型的实时规则调度方法[J]. Journal of Software, 2019, 30(2).

[29] CLARKE E M, GRUMBERG O, PELED D A. Model checking[M/OL]. MIT Press, 2001. http://books.google.de/books?id=Nmc4wEaLXFEC.

[30] CLARKE E, GRUMBERG O, LONG D. Model checking[C]//NATO ASI DPD. [S.l.: s.n.], 1996: 305-349.

[31] 王政. 嵌入式周期控制系统的建模与分析[D]. [出版地不详]: 华东师范大学, 2012.

[32] 李书浩, 王戟, 董威, 等. 反应式系统面向性质测试的方法框架[J]. 电子学报, 2004, 32(S1):226-230.

[33] LIN F, REITER R. Forget it[C]//Working Notes of AAAI Fall Symposium on Relevance. [S.l.: s.n.], 1994: 154-159.

[34] VISSER A. Uniform interpolation and layered bisimulation[M]//Gödel'96: Logical foundations of mathematics, computer science and physics—Kurt Gödel's legacy, Brno, Czech Republic, August 1996, proceedings. [S.l.]: Association for Symbolic Logic, 1996: 139-164.

[35] KONEV B, WALTHER D, WOLTER F. Forgetting and uniform interpolation in large-scale description logic terminologies[C/OL]//BOUTILIER C. IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009. 2009: 830-835. http://ijcai.org/Proceedings/09/Papers/142.pdf.

[36] ACKERMANN W. Untersuchungen über das eliminationsproblem der mathematischen logik[J]. Mathematische Annalen, 1935, 110(1):390-413.

[37] KONEV B, LUTZ C, WALTHER D, et al. Model-theoretic inseparability and modularity of description logic ontologies[J/OL]. Artificial Intelligence, 2013, 203:66-103. https://doi.org/10.1016/j.artint.2013.07.004.

[38] WANG Z, WANG K, TOPOR R W, et al. Forgetting for knowledge bases in DL-Lite[J]. Annuals of Mathematics and Artificial Intelligence, 2010, 58(1-2):117-151.

[39] LUTZ C, WOLTER F. Foundations for uniform interpolation and forgetting in expressive description logics[C/OL]//WALSH T. IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. IJCAI/AAAI, 2011: 989-995. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-170.

[40] KONEV B, LUDWIG M, WALTHER D, et al. The logical difference for the lightweight description logic $\mathscr{EL}$[J]. Journal of Artificial Intelligence Research, 2012, 44:633-708.

[41] ZHAO Y, SCHMIDT R A. Role forgetting for ALCOQH(δ)-ontologies using an ackermann-based approach[C/OL]//SIERRA C. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, 2017: 1354-1361. https://doi.org/10.24963/ijcai.2017/188.

[42] ZHAO Y, SCHMIDT R A, WANG Y, et al. A practical approach to forgetting in description logics with nominals[C/OL]//The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020: 3073-3079. https://aaai.org/ojs/index.php/AAAI/article/view/5702.

[43] GABBAY D M, SCHMIDT R A, SZALAS A. Studies in logic : Mathematical logic and foundations: volume 12 second-order quantifier elimination - foundations, computational aspects and applications[M/OL]. College Publications, 2008. http://collegepublications.co.uk/logic/mlf/?00009.

[44] ZHANG Y, ZHOU Y. Knowledge forgetting: Properties and applications[J]. Artificial Intelligence, 2009, 173(16-17):1525-1537.

[45] ZHANG Y, ZHOU Y. Properties of knowledge forgetting[C]//PAGNUCCO M, THIELSCHER M. Proceedings of NMR 2008. Sydney, Australia: [s.n.], 2008: 68-75.

[46] 文习明. 信息不完备下的知识遗忘[J]. 现代计算机(专业版), 2019, 11:8-13.

[47] LIU Y, WEN X. On the progression of knowledge in the situation calculus[C]//IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence. Barcelona, Catalonia, Spain: IJCAI/AAAI, 2011: 976-982.

[48] FANG L, LIU Y, VAN DITMARSCH H. Forgetting in multi-agent modal logics[J/OL]. Artificial Intelligence, 2019, 266:51-80. https://doi.org/10.1016/j.artint.2018.08.003.

[49] 文习明, 方良达, 余泉, 等. 多智能体模态逻辑系统$KD45_n$中的知识遗忘[J]. 计算机科学, 2019, 46(7):195-205.

[50] 文习明, 方良达, 余泉, 等. 多智能体模态逻辑系统K_n中的知识遗忘[J]. 逻辑学研究, 2019, 02(12):43-62.

[51] IEMHOFF R. Uniform interpolation and sequent calculi in modal logic[J/OL]. Archive for Mathematical Logic, 2019, 58(1-2):155-181. https://doi.org/10.1007/s00153-018-0629-0.

[52] FINE K. Failures of the interpolation lemma in quantified modal logic[J/OL]. The Journal of Symbolic Logic, 1979, 44(2):201-206. https://doi.org/10.2307/2273727.

[53] SCHUMM G F. Some failures of interpolation in modal logic[J/OL]. Notre Dame journal of formal logic, 1986, 27(1):108-110. https://doi.org/10.1305/ndjfl/1093636529.

[54] ZHANG Y, FOO N Y. Solving logic program conflict through strong and weak forgettings[J]. Artificial Intelligence, 2006, 170(8-9):739-778.

[55] EITER T, WANG K. Semantic forgetting in answer set programming[J/OL]. Artificial Intelligence, 2008, 172(14):1644-1672. https://doi.org/10.1016/j.artint.2008.05.002.

[56] WONG K S. Forgetting in logic programs[D]. [S.l.]: The University of New South Wales, 2009.

[57] WANG Y, ZHANG Y, ZHOU Y, et al. Knowledge forgetting in answer set programming [J/OL]. Journal of Artificial Intelligence Research, 2014, 50:31-70. https://doi.org/10.1613/jair.4297.

[58] WANG Y, WANG K, ZHANG M. Forgetting for answer set programs revisited [C/OL]//ROSSI F. IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. 2013: 1162-1168. http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6807.

[59] DELGRANDE J P. A knowledge level account of forgetting[J/OL]. Journal of Artificial Intelligence Research, 2017, 60:1165-1213. https://doi.org/10.1613/jair.5530.

[60] GONÇALVES R, KNORR M, LEITE J, et al. On the limits of forgetting in answer set programming[J]. Artificial Intelligence, 2020, 286(0):103307.

[61] EITER T, KERN-ISBERNER G. A brief survey on forgetting from a knowledge representation and reasoning perspective[J]. KI-Künstliche Intelligenz, 2019, 33(1):9-33.

[62] GONÇALVES R, KNORR M, LEITE J. Forgetting in answer set programming–a survey [J]. arXiv preprint arXiv:2107.07016, 2021.

[63] LIN F. Compiling causal theories to successor state axioms and strips-like systems [J/OL]. Journal of Artificial Intelligence Research, 2003, 19:279-314. https://doi.org/10.1613/jair.1135.

[64] 王文宇. 基于DataSecOps 的个人信息遗忘[J]. 信息安全研究, 2021, 7(9):856-860.

[65] 朱嘉玮. 从“被遗忘权”看个人信息保护[J]. 现代商贸工业, 2020, 8.

[66] LIBERATORE P, SCHAERF M. Arbitration (or how to merge knowledge bases)[J/OL]. IEEE Trans. Knowl. Data Eng., 1998, 10(1):76-90. https://doi.org/10.1109/69.667090.

[67] KONIECZNY S, PÉREZ R P. Merging information under constraints: A logical framework[J/OL]. J. Log. Comput., 2002, 12(5):773-808. https://doi.org/10.1093/logcom/12.5.773.

[68] MAYNARD-ZHANG P, LEHMANN D. Representing and aggregating conflicting beliefs[J/OL]. J. Artif. Intell. Res., 2003, 19:155-203. https://doi.org/10.1613/jair.1206.

[69] KONIECZNY S, LANG J, MARQUIS P. Da2 merging operators[J]. Artificial Intelligence, 2004, 157(1-2):49-79.

[70] EVERAERE P, KONIECZNY S, MARQUIS P. Disjunctive merging: Quota and gmin merging operators[J/OL]. Artif. Intell., 2010, 174(12-13):824-849. https://doi.org/10.1016/j.artint.2010.05.001.

[71] 徐岱. 遗忘及应用[D]. [出版地不详]: 北京大学, 2011.

[72] SANDU G. On the logic of informational independence and its applications[J/OL]. J. Philos. Log., 1993, 22(1):29-60. https://doi.org/10.1007/BF01049180.

[73] SANDU G. The logic of informational independence and finite models[J/OL]. Log. J. IGPL, 1997, 5(1):79-95. https://doi.org/10.1093/jigpal/5.1.79.

[74] VÄÄNÄNEN J A. On the semantics of informational independence[J/OL]. Log. J. IGPL, 2002, 10(3):339-352. https://doi.org/10.1093/jigpal/10.3.339.

[75] SEVENSTER M. On the computational consequences of independence in propositional logic[J/OL]. Synth., 2006, 149(2):257-283. https://doi.org/10.1007/s11229-005-3878-5.

[76] DIJKSTRA E W. Guarded commands, Nondeterminacy and Formal Derivation of Programs[J/OL]. Commun. ACM, 1975, 18(8):453-457. https://doi.org/10.1145/360933.360975.

[77] LIN F. On strongest necessary and weakest sufficient conditions[J/OL]. Artificial Intelligence, 2001, 128(1-2):143-159. https://doi.org/10.1016/S0004-3702(01)00070-4.

[78] DOHERTY P, LUKASZEWICZ W, SZALAS A. Computing strongest necessary and weakest sufficient conditions of first-order formulas[C]//NEBEL B. Proceedings of IJCAI'01. [S.l.]: Morgan Kaufmann, 2001: 145-154.

[79] MAKSIMOVA L. Temporal logics of “the next” do not have the beth property[J]. Journal of Applied Non-Classical Logics, 1991, 1:73-76.

[80] D’AGOSTINO G, LENZI G. On modal μ-calculus with explicit interpolants[J/OL]. Journal of Applied Logic, 2006, 4(3):256-278. https://doi.org/10.1016/j.jal.2005.06.008.

[81] ZHU J, WANG H, XU Z, et al. A new model for model checking: cycle-weighted kripke structure[J/OL]. Frontiers Comput. Sci. China, 2010, 4(1):78-88. https://doi.org/10.1007/s11704-009-0066-7.

[82] SCHNEIDER K. A verified hardware synthesis of esterel programs[C]//IFIP Working Conference on Distributed and Parallel Embedded Systems. [S.l.]: Springer, 2000: 205-214.

[83] SCHNEIDER K. Embedding imperative synchronous languages in interactive theorem provers[C/OL]//2nd International Conference on Application of Concurrency to System Design (ACSD 2001), 25-30 June 2001, Newcastle upon Tyne, UK. IEEE Computer Society, 2001: 143. https://doi.org/10.1109/CSD.2001.981772.

[84] SCHNEIDER K. Proving the equivalence of microstep and macrostep semantics [C/OL]//CARREÑO V, MUÑOZ C A, TAHAR S. Lecture Notes in Computer Science: volume 2410 Theorem Proving in Higher Order Logics, 15th International Conference, TPHOLs 2002, Hampton, VA, USA, August 20-23, 2002, Proceedings. Springer, 2002: 314-331. https://doi.org/10.1007/3-540-45685-6_21.

[85] 陆钟万. 面向计算机科学的数理逻辑[M]. 北京: 北京大学出版社, 1989.

[86] BEN-ELIYAHU-ZOHARY R, ANGIULLI F, FASSETTI F, et al. Decomposing minimal models[C/OL]//BARTÁK R, MCCLUSKEY T L, PONTELLI E. CEUR Workshop Proceedings: volume 1648 Proceedings of the Workshop on Knowledge-based Techniques for Problem Solving and Reasoning co-located with 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), New York City, USA, July 10, 2016. CEUR-WS.org, 2016. http://ceur-ws.org/Vol-1648/paper1.pdf.

[87] BEN-ELIYAHU-ZOHARY R, ANGIULLI F, FASSETTI F, et al. Modular construction of minimal models[C/OL]//BALDUCCINI M, JANHUNEN T. Lecture Notes in Computer Science: volume 10377 Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings. Springer, 2017: 43-48. https://doi.org/10.1007/978-3-319-61660-5_5.

[88] 张丽, 王以松, 谢仲涛, 等. 基于MiniSAT 的命题极小模型计算方法[J]. 计算机研究与发展, 2021, 58(11):2515.

[89] LEI Z, CAI S. Solving (weighted) partial maxsat by dynamic local search for SAT [C/OL]//LANG J. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. ijcai.org, 2018: 1346-1352. https://doi.org/10.24963/ijcai.2018/187.

[90] LEI Z, CAI S. Solving set cover and dominating set via maximum satisfiability[C/OL]// The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020: 1569-1576. https://aaai.org/ojs/index.php/AAAI/article/view/5517.

[91] KRIPKE S A. Semantical analysis of modal logic i normal modal propositional calculi [J]. Mathematical Logic Quarterly, 1963, 9(5-6):67-96.

[92] BROWNE M C, CLARKE E M, GRÜMBERG O. Characterizing finite Kripke structures in propositional temporal logic[J]. Theoretical Computer Science, 1988, 59(1-2): 115-131.

[93] CLARKE E M, EMERSON E A, SISTLA A P. Automatic verification of finite-state concurrent systems using temporal logic specifications[J/OL]. ACM Trans. Program. Lang. Syst., 1986, 8(2):244-263. https://doi.org/10.1145/5397.5399.

[94] BOLOTOV A, FISHER M. A clausal resolution method for CTL branching-time temporal logic[J/OL]. Journal of Experimental & Theoretical Artificial Intelligence, 1999, 11(1):77-93. https://doi.org/10.1080/095281399146625.

[95] ZHANG L, HUSTADT U, DIXON C. First-order resolution for CTL[R]. [S.l.]: Technical Report ULCS-08-010, Department of Computer Science, University of Liverpool, 2008.

[96] ZHANG L, HUSTADT U, DIXON C. A resolution calculus for the branching-time temporal logic CTL[J]. ACM Transactions on Computational Logic (TOCL), 2014, 15 (1):1-38.

[97] ZHANG L, HUSTADT U, DIXON C. CTL-RP: A computation tree logic resolution prover[J/OL]. AI Commun., 2010, 23(2-3):111-136. https://doi.org/10.3233/AIC-2010-0463.

[98] BRADFIELD J C, WALUKIEWICZ I. The μ-calculus and model checking[M/OL]// CLARKE E M, HENZINGER T A, VEITH H, et al. Handbook of Model Checking. 2018: 871-919. https://doi.org/10.1007/978-3-319-10575-8_26.

[99] KOZEN D. Results on the propositional μ-calculus[J/OL]. Theoretical Computer Science, 1983, 27:333-354. https://doi.org/10.1016/0304-3975(82)90125-6.

[100] D'AGOSTINO G, HOLLENBERG M. Uniform interpolation, automata and the modal μ-calculus[J]. Logic Group Preprint Series, 1996, 165.

[101] D'AGOSTINO G, HOLLENBERG M. Logical questions concerning the μ-calculus: Interpolation, lyndon and los-tarski[J/OL]. The Journal of Symbolic Logic, 2000, 65 (1):310-332. https://doi.org/10.2307/2586539.

[102] JANIN D, WALUKIEWICZ I. Automata for the modal μ-calculus and related results[C/OL]//WIEDERMANN J, HÁJEK P. Lecture Notes in Computer Science: volume 969 Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings. 1995: 552-562. https://doi.org/10.1007/3-540-60246-1_160.

[103] DAVIS M, PUTNAM H. A computing procedure for quantification theory[J/OL]. Journal of the ACM (JACM), 1960, 7(3):201-215. http://doi.acm.org/10.1145/321033.321034.

[104] ENJALBERT P, DEL CERRO L F. Modal resolution in clausal form[J/OL]. Theoretical Computer Science, 1989, 65(1):1-33. https://doi.org/10.1016/0304-3975(89)90137-0.

[105] CAVALLI A R, DEL CERRO L F. A decision method for linear temporal logic[C/OL]// SHOSTAK R E. Lecture Notes in Computer Science: volume 170 7th International Conference on Automated Deduction, Napa, California, USA, May 14-16, 1984, Proceedings. Springer, 1984: 113-127. https://doi.org/10.1007/978-0-387-34768-4_7.

[106] DELGRANDE J P. Towards a knowledge level analysis of forgetting[C/OL]//BARAL C, GIACOMO G D, EITER T. Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July

20-24, 2014. AAAI Press, 2014. http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7979.

[107] KOOPMANN P, SCHMIDT R A. Uniform interpolation of -ontologies using fixpoints [C/OL]//FONTAINE P, RINGEISSEN C, SCHMIDT R A. Lecture Notes in Computer Science: volume 8152 Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013. Proceedings. Springer, 2013: 87-102. https://doi.org/10.1007/978-3-642-40885-4_7.

[108] ZHAO Y. Automated semantic forgetting for expressive description logics[D/OL]. The University of Manchester (United Kingdom), 2018. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.740373.

[109] KOOPMANN P. Practical uniform interpolation for expressive description logics [D/OL]. University of Manchester, UK, 2015. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.674705.

[110] ZHAO Y, SCHMIDT R A. FAME: an automated tool for semantic forgetting in expressive description logics[C/OL]//GALMICHE D, SCHULZ S, SEBASTIANI R. Lecture Notes in Computer Science: volume 10900 Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings. Springer, 2018: 19-27. https://doi.org/10.1007/978-3-319-94205-6_2.

[111] BIENVENU M. Prime implicates and prime implicants in modal logic[C/OL]// Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada. AAAI Press, 2007: 379-384. http://www.aaai.org/Library/AAAI/2007/aaai07-059.php.

[112] D'AGOSTINO G. Interpolation in non-classical logics[J]. Synthese, 2008, 164(3): 421-435.

[113] BOLOTOV A. Clausal resolution for branching-time temporal logic.[D]. [S.l.]: Manchester Metropolitan University, 2000.

[114] ZHANG L, HUSTADT U, DIXON C. A refined resolution calculus for CTL[C]// International Conference on Automated Deduction. [S.l.]: Springer, 2009: 245-260.

[115] EMERSON E A, HALPERN J Y. Decision procedures and expressiveness in the temporal logic of branching time[J/OL]. Journal of computer and system sciences, 1985, 30(1):1-24. https://doi.org/10.1016/0022-0000(85)90001-7.

[116] FENG R, ACAR E, SCHLOBACH S, et al. On sufficient and necessary conditions in bounded CTL: A forgetting approach[C]//Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning: volume 17. [S.l.: s.n.], 2020: 361-370.

[117] MYCIELSKI J, ROZENBERG G, SALOMAA A. Lecture notes in computer science: volume 1261 structures in logic and computer science, A selection of essays in honor of andrzej ehrenfeucht[C/OL]. Springer, 1997. https://doi.org/10.1007/3-540-63246-8.

[118] HINTIKKA J. Distributive normal forms in the calculus of predicates[J]. Cambridge University Press, 1953, 20(2).

[119] YANKOV V A. Three sequences of formulas with two variables in the positive propositional logic[J]. Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya, 1968, 32(4):880-883.

[120] MEIER A, THOMAS M, VOLLMER H, et al. The complexity of satisfiability for fragments of CTL and CTL*[J]. International Journal of Foundations of Computer Science, 2009, 20(05):901-918.

[121] LANG J, LIBERATORE P, MARQUIS P. Propositional independence: Formula-variable independence and forgetting[J/OL]. Journal of Artificial Intelligence Research, 2003, 18:391-443. https://doi.org/10.1613/jair.1113.

[122] WOLPER P. Temporal logic can be more expressive[J/OL]. Information and control, 1983, 56(1/2):72-99. https://doi.org/10.1016/S0019-9958(83)80051-5.

[123] JANIN D, WALUKIEWICZ I. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic[C]//International Conference on Concurrency Theory. [S.l.]: Springer, 1996: 263-277.

[124] WANG Y. On forgetting in tractable propositional fragments[J/OL]. CoRR, 2015, abs/1502.02799. http://arxiv.org/abs/1502.02799.

[125] COMON H. Tree automata techniques and applications[J]. http://www. grappa. univ-lille3. fr/tata, 1997.

致　　谢

此时此刻，真是难以表达自己内心的五味杂陈。

如果世界上有幸运儿，我想我应该是其中一个。作为贵州边远山村的孩子，能走到今天我是幸运的。而这些幸运都是由我身边的老师、亲人和朋友一点点累积的。

从小就听妈妈在耳边说：“你好好读书，只要你能读，我砸锅卖铁都会供你上学”。这句话看着简单，但却藏着深深的期许。妈妈小时候成绩极好，但是由于家庭贫寒和重男轻女的思想让她错过了上学的机会。因而将自己的希望寄托在我的身上，当然更多的是为了我的将来考虑，不想让我走她的老路。对我来说，父母是人生道路上最初感染我的人。小时候每当看见爸爸妈妈辛辛苦苦的干活却过不上体面的生活，心里就暗暗发誓以后好好读书，将来让爸爸妈妈过上好一点的生活。爸爸虽然不爱说话，但是心里也是及希望我考上大学的。也是这些生活的点滴让我走到了今天，尽管途中很多困难，但都咬牙挺过。

老师，是我人生道路上指引方向、解决疑惑的人。从小学到现在，我的班主任或者导师在我的学习道路上都扮演着重要的角色。让我最记忆深刻的是高中班主任张焱老师及大学班导师谭熹微老师，他们是我在人生岔路口走向正途的引导人。而让我最受益的导师当属硕士及博士导师王以松老师，他不仅是我学术上的导师，也是我人生方向的引领者。王老师严谨的科研态度让我获益良多，从小我就是个粗心大意的人，现在能在知识表示与推理这个要求严谨的方向有点点成果王老师功不可没。从硕士走向博士，王老师也给了我很好的指向，并给了我出国留学的机会。说到留学，两位阿姆斯特丹自由大学的导师（Erman Acar 和Stefan Schlobach）也是非常感谢的，他们给了我很多做科研的指导，尤其是Erman Acar对我的英文写作有很多帮助。此外，也感谢国防科技大学的刘万伟老师对我博士期间论文的指导。

人是群居动物，良好的群体环境不仅对生活大有裨益，对学术更加有帮助。感谢人工智能团队在我这七年（硕士+博士）的学习生涯中的帮助，从他们身上我学到了很多，他们的帮助让我感到了幸福和温暖。同时，感谢实验室的师兄弟姐妹在日常生活中给予我的帮助，读书期间的温馨和睦相处和茶余饭后时的谈笑风生都让我感受到家的温暖。

感谢贵州大学计算机学院的老师为本文工作所提供的支持和帮助；感谢秦永彬教授、彭长根教授、张明义教授等为本文的选题和研究方案的设计所提出的宝贵意见和建议。

最后，感谢参与该博士学位论文评审、答辩的诸位专家学者，感谢您们为提高我的博士学位论文质量所提出的宝贵修改意见和建议。

攻读博士学位期间科研和论文情况

一、参与科研项目

1. 国家自然科学基金重点项目：数据共享应用的块数据融合分析理论与安全管控模型研究，项目基金号U1836205
2. 国家自然科学基金：不完全知识的遗忘理论研究及应用，项目基金号
3. 国家自然科学基金：析取逻辑程序归纳学习研究及应用，项目基金号

二、发表论文

[1] **Renyan Feng**, Erman Acar, Stefan Schlobach, Yisong Wang, Wanwei Liu. On Sufficient and Necessary Conditions in Bounded CTL: A Forgetting Approach [C]. KR2020, 17(1): 361-370. (CCF B类会议)

附：学位论文原创性声明和关于学位论文使用授权的声明

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究在做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

论文作者签名：_____ 日 期：_____年 月

关于学位论文使用授权的声明

本人完全了解贵州大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权贵州大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密论文在解密后应遵守此规定）

论文作者签名：_____导师签名：_____日 期：_____年 月