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系统。随着反应式系统越来越复杂，系统正确性、系统及其系统规范（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），寻找使得系统满足规范所需的最弱信息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它信息的情况下“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技术，其被应用于信息隐藏、冲突解决、多智能体中的知识融合、本体摘要提取和计算逻辑差等领域。特别地，在情景演算中，遗忘被用于求解WSC和SNC，从而计算智能规划中的后继状态公理。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. 研究了CTL下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，且满足这四条公设的公式为遗忘的结果。为了计算CTL遗忘，本文提出了两种计算方法：

- 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$R_{\mathrm{CTL}}^{\succ,S}$

（clausal resolution calculus for CTL）。在此方法中，首先将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$\mathrm{SNF}^{g}_{\mathrm{CTL}}$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。

- 基于模型的CTL遗忘计算方法。本文证明了CTL遗忘结果不总是存在的，即：存在一些公式遗忘原子命题集的结果不可用CTL公式表示。因此，探讨了遗忘总是存在的约束CTL。在这种方法中，规定了公式的长度为整数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。

2. 研究了μ-演算遗忘。μ-演算具有均匀插值（uniform interpolation）性质，本文证明了μ-句子遗忘与均匀插值是等价的。此外，提出一种新的互模拟，并证明μ-公式对这种互模拟是不变的（invariant）。最后，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。

3. 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，而不能计算像反应式系统这类不可终止系统的WSC。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。

4. 用Prolog实现了1中提到的基于归结的计算CTL遗忘的算法，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，在随机产生的公式中，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

A reactive system is a nonterminal system generated and developed on the basis of responsive, resilient, elastic, and message-driven requirements of the application program. As the reaction system becomes more and more complex, the correctness of a system and the consistency between the system and its specification become more and more difficult to be guaranteed. Model checking is an effective method to ensure a system's correctness, and a reactive system is represented as a Kripke structure in such an environment. However, in model checking, if a system does not meet the given specification (that is, it does not conform to the specification), it has been an important problem for a long time to find the weakest information needed to make the system meet the specification. The strongest necessary condition (SNC) and the weakest sufficient condition (WSC) are two crucial concepts that correspond to the strongest post-condition (SP) and the weakest precondition (WP) in verification, respectively. However, existing methods cannot calculate the SNC and WSC of nonterminal systems, such as reactive systems. In addition, as systems update and evolve, existing specifications inevitably conflict with new knowledge. In this sense, how to "remove" the previously integrated atoms without affecting other information is also an urgent problem to be solved.

Temporal logic is one of the important logic describing the specification of systems. Computation tree logic (CTL) is an important branch-time temporal logic, which is widely used in system specification description because of its characteristic that model checking can be done in polynomial time. However, μ-calculus is a logic language with more vital expression ability but more complex model checking than CTL. Nevertheless, CTL and μ-calculus are interpreted under Kripke semantics, which is directly related to the reactive system being expressed as a Kripke structure. Therefore, this paper takes these two logical languages as the research background to explore the solutions to the above problems of reactive systems.

Forgetting was first formally defined in propositional logic (PL) and FOL by Lin and Reiter. As a technique for distilling knowledge, it has been explored in various logic languages and is widely used in AI, such as information hiding, conflict resolving, knowledge integration in multi-agent, ontology abstract extraction, and logical difference computation. In particular, forgetting is used to solve WSC and SNC in a scenario of situation calculus to compute the successor state axioms in planning. This paper aims to solve the above-mentioned problems from the perspective of forgetting. The main contributions of the paper 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 decomposition, slice, 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. To calculate CTL forgetting, this paper proposes two approaches:

- A resolution-based method. This approach is based on 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. Throughout the process, the formula to be calculated always maintains bisimilar equivalence with the output of each process.
- A model-based method. This paper shows that the result of forgetting in CTL does not always exist, i.e., forgetting some atoms from some formula can not be expressed by a CTL formula. However, it proves that the forgetting always exists in bounded CTL, i.e., the size (the number of signatures in a formula) of formulas are limited to an integer, the set of atoms that construct the formulas, and the state space are finite. Therefore, the model of a formula can be expressed by its characterizing formula, and then the forgetting result is the disjunction of the characterizing formulas of its models.

2. 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 are equivalent for μ-sentence. In addition, we define a general bisimulation on valuations, $\langle X, V\rangle$-bisimulation, in which X is a set of variables and V is a set of atoms. It is proved that μ-calculus is invariant under $\langle X, V\rangle$-bisimulation. Finally, the properties and complexity results related to forgetting are given, which proposes a new point for studying uniform interpolation.

3. 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..

4. Implemented the algorithm proposed in 1, and some experiments are shown. Two experiments, i.e., computing forgetting and WSC, were performed for standard and randomly generated datasets. Experiments show that the longer the formula is, or the more atoms are for-

gotten, 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

首先，本章介绍研究背景，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]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。软件和硬件系统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。INTEL、AMD、IBM、NVIDIA、CADENCE、Motorola、西门子和微软等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：剑桥大学对ARM6处理器进行了验证[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型运载火箭软件在不同飞行条件下的代码重用	火箭与其它卫星毁灭
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（reactive system）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）被同一种形式化语言分别描述为φ_{imp}和φ_{spec}，然后证明其推理“$\varphi_{imp} \rightarrow \varphi_{spec}$”或“$\varphi_{imp} \leftrightarrow \varphi_{spec}$”的正确性，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，当程序为循环语句时，验证所需要的不变式（invariant）获取是个困难的问题。因此，为了避免类似于在Hoare逻辑[11]、动态逻辑[12]和分离逻辑[13]形式化验证过程中寻找不变式问题，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）的方法证明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，并发系统可以被抽象为一个Kripke结构$\mathscr{M}$，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（KR）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。与此同时，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。

时序逻辑（也叫时态逻辑）作为一种描述系统规范的形式化语言，其研究状态随时间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，所以时态逻辑在软件和硬件系统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完成的。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，且这条路径上处于偶数位置的状态满足特定的性质”这一规范是不能用CTL来表示的[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

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

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

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

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。

在模型检测中，反应式系统通常用Kripke结构表示[19-20]。当给定了反应式系统的Kripke结构模型$\mathscr{M}$和规范φ，就存在模型检测问题$\mathscr{M} \models^{?} \varphi$：

$$\text{系统输出}=\begin{cases} Yes, & \text{若}\mathscr{M} \models \varphi; \\ No\text{和负例}, & \text{否则。} \end{cases}$$

当$\mathscr{M}$不满足φ时，使用什么信息对$\mathscr{M}$进行修正，使得其满足给定的规范φ是一个重要的问题。这就是寻找最弱前件（weakest precondition，WP）问题[21]，在人工智能（artificial intelligence，AI）中也称为最弱充分条件（weakest sufficient condition，WSC），与之对偶的另一个概念是最强必要条件（strongest necessary condition，SNC）[22-23]。

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部分信息。此外，随着时间推移，由于系统的部分信息会因某些原因而过时，需要将这样的信息在不影响其它信息的情况下“消除”。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(s)的方案：

(1) 先生产足够的se，然后在再生产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（带标签的状态转换图）$\mathscr{M}=(S,R,L)$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集；
- $S=\{s_0,s_1,s_2,s_3,s_4\}$为状态空间；
- $R=\{(s_0,s_1),(s_1,s_2),(s_1,s_3),(s_1,s_4),(s_2,s_0),(s_3,s_0),(s_4,s_0)\}$为状态转换关系集；
- $L:S\to 2^V$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$L(s_4)=\{se,sp\}$。

假定，由于经济危机或者战略调整，导致该企业不能再生产跑车。这意味着所有规范和Kripke结构都不再需要考虑sp的，因此应该“移除”。

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[24]。但是对于给定原子命题集，从这些大型系统（或规范）中“移除”一些原子命题，而保留与这些原子命题无关的性质是一个复杂的问题。

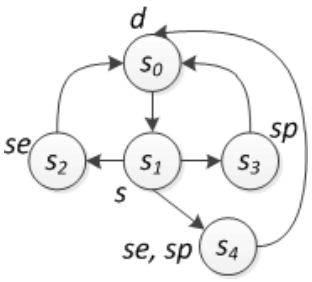


图 1.1: 汽车制造企业模型

1.1.2 研究意义

最强后件（the strongest post-condition，SP）和最弱前件（WP）（分别对应于上文提到的SNC和WSC）是形式化验证中两个重要的概念，其不仅被用于汇编语言程序推理[25]和制定验证条件[26]，还被应用于形式化验证过程中的负例生成[27]和系统精化（refinement）[28]。当模型$\mathscr{M}$不满足规范φ（$\mathscr{M} \not\models \varphi$）时，找到某个性质$\psi$，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$\varphi$是必要的，即$\psi$是使得$\mathscr{M} \models \varphi$成立的充分条件（$\mathscr{M} \models \psi \rightarrow \varphi$）。然而，现有方法不能直接应用于如下情形：计算当$\mathscr{M}$为不终止系统（如：反应式系统）时使得“$\mathscr{M} \models \varphi$”为真的最强必要条件和最弱充分条件（其详细原因将在下文指出）。此时，探索如何计算“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。如下问所说，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，借助遗忘计算上述形式化验证问题中的充分（必要）条件，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）和知识独立性的其它领域。

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，一个大型应用程序的运行通常需要数十台服务器，这么高昂的代价换来的是效果和维护成本不成正比：秒级的响应时间，需要数小时的维护。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，它们可以被部署到移动设备和具有多核心处理器的云端集群上。但是，用户对应用程序的响应时间要求从秒级变到毫秒级，相应的成功率要求为100%；程序需要处理的数据量也从之前的GB级扩展到现在的PB级。原有的软件架构已经无法满足当前的需求。

在这种背景下，应用程序需要具备以下特质：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]。此外，其同时满足上述几种要求，且从传统的静态模式逐步向动态、开放、自适应、服务化的模式演化[29]。反应式系统由来已久[30-31]，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）以及其它嵌入式系统。

此外，反应式系统的应用非常广泛：包括航空电子等安全攸关的领域和生活息息相关的汽车电子等领域。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[32]。为了确保反应式系统的正确性和实时性，研究者们提出了不同的解决方法，包括：模型检测方法[19,33-34]、目标驱动的运行时需求建模框架[29]、基于图模型的实时规则调度方法（graph-based real-time rule scheduling,简称GBRRS）[32]、SPARDL[35]和选择性测试方法[36]等。

在上面的方法中，多数是验证（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（称为充分条件）来对系统进行修改至关重要。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[37]，在此之前的研究中多提到的是均匀插值[38-39]和SOQE[40]。

在命题逻辑中（propositional logic，PL），公式φ遗忘一个原子命题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$\varphi$

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

中X的全部出现替换为Y得到的结果。公式φ中遗忘有限原子命题集P的定义如下：

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

在FOL中，遗忘通常被看作SOQE问题的一个实例：从FOL公式φ中遗忘一个n-元谓词P，得到的结果为一个二阶公式$\exists R\varphi[P/R]$[37]，其中R为n-元变量。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严格大于一阶逻辑，因而可以容易得出FOL下的遗忘不是封闭（存在）的结论，也就是从某些一阶公式中遗忘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[41]，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，其子类中的遗忘通常被用来抽取视图（review）[42-46]。这些子类包括：ALCOI（the basic ALC extended with nominals and inverse roles）和ALCOIH（ALCOI extended with role hierarchies, the universal role and role conjunction）。

现有计算一阶逻辑和描述逻辑遗忘的方法有基于归结和基于Ackerm- ann引理的方法[47]。其中基于归结的方法是一种基于子句归结的方法，其基础是归结规则。通常在这种方法中首先把公式转换为其子句形式，然后使用归结规则，最后移除含有要遗忘的谓词（原子命题）的子句，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[47]). *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“¬”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“¬”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）[48]。因为模态逻辑系统中引入了模态词，模态逻辑中的

遗忘不同于经典逻辑下的遗忘。所以，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. [49] 令S5公式$\varphi = \mathrm{K}p \wedge \neg\mathrm{K}q \wedge \neg\mathrm{K}\neg q$，则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[50]、模态一阶逻辑S5中的遗忘也得到了研究[51]，Fang等人讨论了关于多模态（multi-modal）K_n、D_n、T_n、$K45_n$、$KD45_n$情形下遗忘的存在性[52-54]——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。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。

其在模态逻辑和时序逻辑系统中取得了一系列研究成果：LTL、CTL和CTL*不具有均匀插值性质[55-56]，S5、K和KD模态逻辑系统具有均匀插值性质[57]，而一些模态逻辑系统没有均匀插值性质。如：量词模态逻辑S5（quantified modal logic S5）[58]、K4、和S4及其扩展都没有均匀插值性质[59]，因而其遗忘也不是封闭的。研究具有均匀插值性质的模态逻辑遗忘可以借鉴S5系统的遗忘方法，也可以参考K系统下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[47]。

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[60-67]，和Eiter、Gonccalves等人的综述[68-69]。

遗忘有很多应用，这里列出下面几点：

- 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[23]。
- 信息隐藏：在某些关键领域，为实现隐私保护，必须隐藏敏感信息。现有方法包括基于本体[41]和基于DataSecOps的个人信息保护[70]。要做到隐私保护，只需要隐藏（遗忘）那些敏感的概念（concept）和角色（role）符号。值得注意的是，个人信息保护涉及被遗忘权[71];
- 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，能事先快速地摒弃许多无关信息，并提取出本体的概要是非常有用的；
- 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），因而不能简单地将它们放在一起。归并考虑当这些信息矛盾时如何将它们整合[72-76]，且基于遗忘的归并可以尽可能少地遗忘原子以保持知识的一致性[77]。
- 逻辑独立性：生活中许多场景都需要判断哪些信息是无关的（或相关的），即信息的独立性[78-81]，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，还有其它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。所以，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[77]。

1.2.3 最强必要条件（SNC）和最弱充分条件（WSC）

正如上文所说，WSC（SNC）对于软件工程中系统的形式化验证非常重要。此外，SNC和WSC还可用于系统精化[28]、模型检测中的负例产生[27]、汇编语言程序的推理[25]和制定验证条件[26]。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后件（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前提条件（WP），WSC能被所有的充分条件蕴涵。

给定一个程序（program）S和某一状态（state）的规范（specification）Q，若S关于Q的WSC是一个能够描述S初始状态的规范，那么前提是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[21]。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[23]，且SNC和WSC都可以用遗忘来计算[22,82]。随后，SNC和WSC被扩展到FOL下，

且用SOQE实现了SNC和WSC的计算[82]。

本文将在背景知识部分对SNC和WSC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，现存方法不能求解反应式系统下的WSC（SNC）。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确保系统的正确性。在知识表示与推理中，遗忘技术可以用于计算给定理论（公式）的WSC（SNC）。但是，如上所述，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*KR*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

本文探究了CTL和μ-演算中遗忘的定义和性质，特别是其遗忘结果的存在性、复杂性等，为探索用遗忘计算SNC和WSC奠定理论基础。具体说来，遗忘具有削弱、正维持、负维持、无关性等基本准则[48]。本文探索CTL和μ-演算遗忘的以上四个准则，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[55]，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，本文除了研究上述CTL遗忘的性质，还针对CTL子类，特别是能保证其遗忘结果仍然是CTL可表达的子类的遗忘进行了研究。在这些子类中，一个特殊的子类是约束CTL（bunded CTL）：每个公式具有有限个模型，且每一个模型都能用一个CTL公式表达。因此，其遗忘是封闭的。
- **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[83]。这意味着，μ-句子遗忘任意原子命题集得到的结果仍然是μ-句子。本文给出了μ-演算下遗忘的主要框架：包括上述遗忘的性质和计算遗忘的方法。与CTL不同，μ-演算公式含有变元。为此，本文提出一种新的互模拟，并证明μ-公式对这种互模拟是不变的（invariant）。特别地，证明了μ-演算遗忘与均匀插值是一个对偶概念，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，计算遗忘可以在多项式时间内完成，这为μ-演算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，设计并使用Prolog实现了计算CTL遗忘结果的原型系统，并从理论和实验角度分析其计算复杂性。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，给出了基于遗忘的计算“有限系统模型（Kripke结构）在给定条件下的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该系统的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，其引入了时序算子，已有文献表明CTL不具有均匀插值性质。因为遗忘与均匀插值是一对对偶概念，研究CTL的遗忘不能像已有的经典命题逻辑和模态逻辑S5那样。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，并证明：当所有在转换为CTL标准形式过程中引入的新原子命题都被“消除”时，使用这一方法得到的结果即为遗忘结果。尽管CTL遗忘不是封闭的，但本文给出：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，Lin 和Doherty 等人分别提出了遗忘与SNC（WSC）的关系[22,82]。特别地，经典命题逻辑中的SNC和WSC被用于计算规划问题中的后继状态公理。本文给出，给定一个有限反应式系统（Kripke structure），并将该系统表示为其特征公式，就可使用上述的CTL和μ-演算遗忘计算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）公式遗忘的推理问题最难是Π_2^p-完全的，DNF （Disjunctive normal form）公式遗忘的蕴涵问题是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，对应的蕴涵问题是Π_2^p-完全的。本文从现有复杂性结果和自动机理论基础上研究CTL和μ-演算遗忘在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，蕴涵问题都在EXPTIME中，且有的蕴含问题是EXPTIME-完全的。CTL下关于遗忘的推理问题：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；而蕴涵问题的复杂性在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：

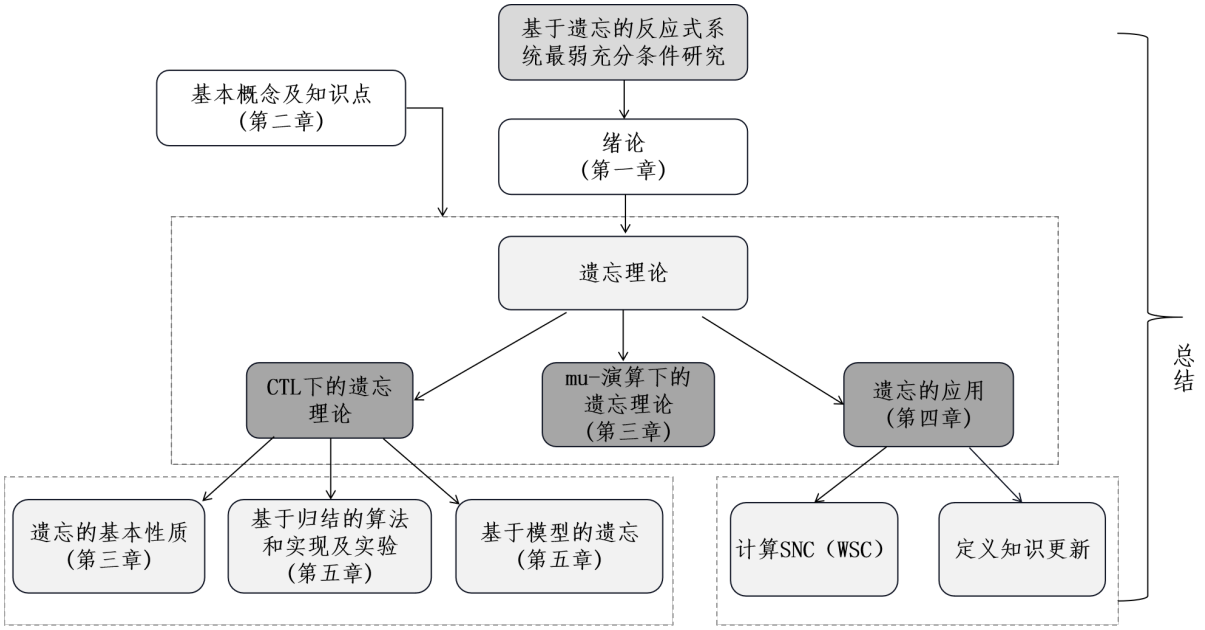


图 1.2: 本文的章节内容组织结构图

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，以及相关语言的模型结构——Kripke结构。然后，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*和μ-演算遗忘的定义及其基本性质。首先，给出一种新的互模拟定义，并介绍其基本属性；其次，使用互模拟来定义CTL和μ-演算下的遗忘，并研究遗忘的属性，包括表达性属性（representation theory）和复杂性结果等。本章指出CTL遗忘是不封闭的，即存在某些公式的遗忘结果不能用CTL来表示。此外，证明了μ-句子遗忘的结果总是存在的，且不含有不动点操作的这类μ-公式遗忘的结果总在这个类里面。

第四章为遗忘理论在反应式系统中的应用。讲述如何将遗忘应用于计算反应式系统在给定条件下的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等人提出的八条基本准则。

第五章为*CTL*遗忘的计算方法。讲述了两种计算CTL遗忘的方法：基于模型和基于归结的方法。此外，给出了基于归结的算法Prolog实现描述和实验结果分析。在基于模型的计算CTL遗忘中：第 5.1节探讨了约束CTL遗忘是封闭的情形；为此，提出了一种约束的互模拟，并定义给定深度的计算树在给定原子命题集下的特征公式，继而给出有限情况下Kripke结构在给定原子命题集下的特征公式；最后说明，在约束情形下，CTL公式的遗忘结果总是CTL公式可表达的。在基于归结的计算CTL遗忘中：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；其次，使用归结规则计算要遗忘的原子命题上所有可能的归结结果；随后，移除那些含有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，根据上述过程，提出计算CTL遗忘的算法，并分析该算法的时间和空间复杂性。

第六章为总结与展望。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，介绍反应式系统和解释时序逻辑语言所需的模型结构，即*Kripke*结构。其次，介绍时序逻辑中本文探讨的计算树逻辑和μ-演算的语法和语义、*CTL*归结系统及这些规则的基本性质。此外，遗忘理论是本文的研究重点，其概念、性质及密切相关的逻辑系统中的研究情况将会被当作本章的重点详细介绍。

本文将命题变量（也叫原子命题）集记作$\mathscr{A}$，$V \subseteq \mathscr{A}$ 是$\mathscr{A}$的子集。此外，规定$\overline{V}$是V在$\mathscr{A}$上的补，即$\overline{V} = \mathscr{A} - V$。

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）以及其它嵌入式系统。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，时序逻辑是描述系统规范的语言[31]。在模型检测中，反应式系统通常用Kripe结构表示[19-20]。本文主要探讨规范描述语言和Kripke结构之间的关系，如何将反应式系统转换为Kripke结构的方法可以参见文章[84-86].

下文介绍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：CTL和μ-演算。

2.2　Kripke结构

Kripke结构作为一种描述转换系统（transition system）的数学模型，在理论计算机科学领域有着广泛的应用，它是解释时序逻辑公式的模型结构，本节将对Kripke结构做详细介绍。

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，且这些命题符号属于$\mathscr{A}$；
- 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）。

$\mathscr{L}^p$的原子公式集是$Atom(\mathscr{L}^p)=\mathscr{A}$，公式集为$\mathscr{F}(\mathscr{L}^p)$。$\varphi\in\mathscr{F}(\mathscr{L}^p)$当且仅当它能由（有限次使用）以下的三条规则生成[87]：

- 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，“$\top$”和“$\bot$”和也是原子公式，分别称作“真”和“假”。原子命题及其否定形式称为文字（literal），有限个文字的析取称为子句（clause），有限个文字的合取称为项（term）。

例 2.1. 下面几个字符串是$\mathscr{L}^p$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$\mathscr{F}(\mathscr{L}^p)$。

在本文中，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，即：省略掉冗余的标点符号。为此，规定如下符号优先级，且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$(\neg p\leftrightarrow q\vee r)\rightarrow r\wedge p$。

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

后文中也用$\top$代表1，$\bot$代表0（此时真假赋值为$v:\mathscr{A}\rightarrow\{\bot,\top\}$），且满足对任意真假赋值$v$，都有$\top^v=\top$和$\bot^v=\bot$。由定义 2.1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$p \in \mathscr{A}$。
- $(\neg\varphi)^v = \begin{cases} \top, & \text{如果}\varphi^v = \bot; \\ \bot, & \text{否则。} \end{cases}$
- $(\varphi \wedge \psi)^v = \begin{cases} \top, & \text{如果}\varphi^v = \psi^v = \top; \\ \bot, & \text{否则。} \end{cases}$
- $(\varphi \vee \psi)^v = \begin{cases} \top, & \text{如果}\varphi^v = \top\text{或}\psi^v = \top; \\ \bot, & \text{否则。} \end{cases}$
- $(\varphi \to \psi)^v = \begin{cases} \top, & \text{如果}\varphi^v = \bot\text{或}\psi^v = \top; \\ \bot, & \text{否则。} \end{cases}$
- $(\varphi \leftrightarrow \psi)^v = \begin{cases} \top, & \text{如果}\varphi^v = \psi^v; \\ \bot, & \text{否则。} \end{cases}$

对任意命题公式φ和真假赋值v，当$\varphi^v = \top$时，称v是公式φ的一个模型，记为$v \models \varphi$，读作“v满足φ”。一般地，若存在一个真假赋值v使得$v \models \varphi$，则称公式φ是可满足的。如果$\neg\varphi$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[88-90]；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[91-92]。

值得注意的是，一个真假赋值（本文也称作解释）也可以看作是原子命题集I，其表示：属于I的原子命题解释为真，反之为假。

模态逻辑在经典逻辑中引进“必然”和“可能”这两种模态词。如上所述，命题的真假值只有两种，命题是真的或是假的。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”。值得一提的是，时序逻辑也是模态逻辑的一种[47]。尽管如此，在本文中，模态逻辑通常指不带有时序操作符（参见下文）的情况，时序逻辑特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）。模态公式集$\mathscr{F}^{\mathscr{M}}$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$* \in \{\wedge, \vee, \to, \leftrightarrow\}$。

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[93]。Kripke语义定义在Kripke结构上，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，如果R是一个等价关系（模态逻辑S5中），那么一个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集。此时，称$\mathscr{M} = \langle W, w\rangle$为一个K-解释（K-interpretation）[48]。

定义 2.3. 给定一个K-解释$\mathscr{M} = \langle W, w\rangle$，其与$S5$公式的可满足关系归纳定义如下：

- $\mathscr{M} \not\models \bot$，$\mathscr{M} \models \top$；
- $\mathscr{M} \models p$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$\mathscr{M} \models \varphi$，则称公式$\varphi$是可满足的。如果对所有的$\mathscr{M} = \langle W, w\rangle$，$\mathscr{M} \models \varphi$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集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（参见下文）；
- $R \subseteq S \times S$是状态转换函数，且对任意$s \in S$，存在$s' \in S$使得$(s, s') \in R$；
- $L : S \to 2^{\mathscr{A}}$是一个标签函数；
- $[_] : \mathrm{Ind} \to 2^{S \times S}$是一个函数，其使得对任意$ind \in \mathrm{Ind}$，若$s \in S$，则存在唯一一个$s \in S$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$\pi = (s_0, s_1, s_2, \ldots)$，且满足对任意$j \geq 0$，$(s_j, s_{j+1}) \in R$。用$s' \in \pi$表示$s'$是路径$\pi$上的一个状态。特别地，用$\pi_s$表示以$s$为起点的$\mathscr{M}$上的一条路径，即：$\pi_s = (s = s_0, s_1, \ldots)$。如果对任意$s' \in S$，都存在路径$\pi_s$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$\pi_s^{\langle ind\rangle}$（$ind \in \mathrm{Ind}$）是一条路径$(s_0(= s),\ s_1,\ s_2,\ \ldots)$，且对任意$j \geq 0$，有$(s_j, s_{j+1}) \in [ind]$。

此外，初始Ind-Kripke结构与其它几种特殊的Kripke结构的关系如下：

- 初始Kripke结构$\mathscr{M} = (S, R, L, s_0)$：从初始Ind-Kripke结构$\mathscr{M}$中去掉$[_]$元素得到；
- Ind-Kripke结构$\mathscr{M} = (S, R, L, [_])$：从初始Ind-Kripke结构$\mathscr{M}$中去掉初始状态$s_0$得到；

- 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。

通常一个转换系统（包括反应式系统）能够被抽象为一个Kripke结构[24]。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$s$是$\mathscr{M}$的初始状态，则称$\mathscr{K}$是*初始(Ind-)结构*。在这些结构中，（索引）路径这一概念可以类似地定义。

树是一种只有一个根节点（没有其它节点指向，且可达于其它任意节点的节点）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$s\in S$，$\mathscr{M}$上以s为根节点、深度为n（$n\geq 0$）的*计算树*$\mathrm{Tr}_n^{\mathscr{M}}(s)$递归定义如下[94]：

- $\mathrm{Tr}_0^{\mathscr{M}}(s)$ 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，并且若$(s,s')\in R$，则$s$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。本节介绍CTL和μ-演算这两种时序逻辑。

2.3.1　计算树逻辑（CTL）

计算树逻辑是由Clarke和Emerson等人于1986年提出的一种分支时间时序逻辑[95]，它能很好的描述并发系统的一些性质，包括：互斥性和安全性等。此外，Emerson和Halpern 证明了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[96]。具体说来，对于给定的CTL公式φ，如果公式的长度[1]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$(\mathscr{M},s_0)\models\varphi$。

这里给出带索引的CTL公式定义，CTL公式是这种公式的子类。带索引的CTL的语言$\mathscr{L}$由下面的几类符号构成：

- 原子命题集$\mathscr{A}$；
- 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$ind\in$ Ind”的路径；

[1]给定公式φ，该公式里符号的个数是公式的长度，记为$|\varphi|$。

- 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，$p \in \mathscr{A}$，$ind \in \mathrm{Ind}$。$\mathscr{L}$中其它形式的公式可以通过如下定义（使用上述定义中的形式）得到：

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

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

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

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

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

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

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

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

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

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“¬”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，规定联结符号的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中各类符号的优先级如下，且从左到右优先级逐渐降低：

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

给定一个不包含“$\rightarrow$”的公式φ和原子命题p。在φ中，若p的前面有偶数个否定¬，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，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，其递归定义如下。

定义 2.4 (带索引的CTL的语义). 给定公式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：

- $(\mathscr{M},s)\models$ ***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$(\mathscr{M},s_i)\models\psi$，且对任意$1\leq j<i$，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$\pi_s^{\langle ind\rangle}=(s,s',\ldots)$，有$(\mathscr{M},s')\models\psi$；
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}\mathrm{G}\psi$ 当且仅当对任意$s'\in\pi_s^{\langle ind\rangle}$, $(\mathscr{M},s')\models\psi$；
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}(\psi_1\mathrm{U}\psi_2)$ 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$s_k\in\pi_s^{\langle ind\rangle}$，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[94,97]，即：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。

令φ、φ_1和φ_2为公式，这里列出文中出现的一些记号及其含义。

- $Mod(\varphi)$：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$\varphi_1\equiv\varphi_2$；
- $Var(\varphi)$：出现在φ中的原子命题集。

上述的记号也适用于讨论的对象为公式集的情形。此外，给定一个公式集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$\bigvee\Pi$和$\bigwedge\Pi$分别表示$\Pi$中公式的析取和合取，在本文中$\Pi$也表

示$\bigwedge\Pi$。

此外，给定公式φ和原子命题集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IR$(\varphi,V)$。一种特殊的形式是$Var(\varphi)\subseteq V$，此时称$\varphi$为集合$V$上的公式。可以类似定义公式集与原子命题集的无关性，即：对于公式集合Π中的任意一个公式φ，如果IR(φ,V)均成立，那么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，显然以下结论成立。

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

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*）。由此可见，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，但是它们不是等价的。所以，可以得到下面的引理。

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

证明. (i) 令$\varphi'=\varphi[i/j]$为用新索引j替换φ中的索引i得到的公式。

($\Rightarrow$) 对φ的任意模型$(\mathscr{M},s)$（$\mathscr{M}=(S,R,L,[_],s)$），存在$(\mathscr{M}',s)\leftrightarrow_{\emptyset}(\mathscr{M},s)$，使得$(\mathscr{M}',s)\models\varphi'$（$\mathscr{M}'=(S,R,L,[_]',s)$）且对任意$x\in$ 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(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CTL公式能够在多项式时间内被转换为CTL全局子句分离范式（separated normal form with global clauses for CTL，$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句）[98-99]。$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句是具有下面几种形式的公式：

$\mathrm{AG}(\mathbf{start} \rightarrow \bigvee_{j=1}^{k} m_j)$	(初始句，initial clause)
$\mathrm{AG}(\top \rightarrow \bigvee_{j=1}^{k} m_j)$	(全局子句，global clause)
$\mathrm{AG}(\bigwedge_{i=1}^{n} l_i \rightarrow \mathrm{AX}\bigvee_{j=1}^{k} m_j)$	(A-步子句，A-step clause)
$\mathrm{AG}(\bigwedge_{i=1}^{n} l_i \rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{X}\bigvee_{j=1}^{k} m_j)$	(E-步子句，E-step clause)
$\mathrm{AG}(\bigwedge_{i=1}^{n} l_i \rightarrow \mathrm{AF}l)$	(A-某时子句，A-sometime clause)
$\mathrm{AG}(\bigwedge_{i=1}^{n} l_i \rightarrow \mathrm{E}_{\langle ind\rangle}\mathrm{F}l)$	(E-某时子句，E-sometime clause)

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$\mathrm{AG}(P \rightarrow G)$形式。因此在不引起歧义的情况下，下文中使用$P \rightarrow G$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中的规则转换为一个$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集，记为$T_\varphi$。

表 2.1: 转换规则

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

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；φ，φ_1，和φ_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)**通过引入新的原子命题替换复杂的公式；规则**Trans(6)**到规则**Trans(12)**用于移除那些不能出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$中的时序算子[100]。

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 形式，记为$nnf(\varphi)$；
(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$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集$T_\varphi$，其中$T_\varphi$由如下导出（*derivation*）序列生成：

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

使得

- p是一个新的原子命题，即：$p \notin \{\mathbf{start}\} \cup Var(\varphi)$；
- $T_{t+1} = (T_t - \{\psi\}) \cup R_t$ $(t \geq 0)$，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上得到的结果集；
- T_n中的每个公式都是$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句形式。

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$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集。

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。此外，一旦路径量词E被索引标记了，它就不会被其它索引标记。

(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。 □

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$\mathrm{AG}\varphi$ 和$\mathrm{AG}\psi$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[99]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中的规则转换$\{\mathrm{AG}(\mathbf{start} \rightarrow z), \mathrm{AG}(z \rightarrow (\mathrm{EG}\neg p \wedge \mathrm{AF}p))\}$，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$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句为：

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-演算来表达。μ-演算是模态逻辑的扩

[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。

展，构成μ-演算语言的符号有：

- 原子命题符号集：$\mathscr{A}$；
- 变元符号的可数集：$\mathscr{V}$；
- 常量符号：$\bot$和$\top$；
- 布尔联结符号：$\vee$，$\wedge$，$\rightarrow$和$\neg$;
- 路径量词符号：A和E;
- 时序操作符号X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[101]，本文规定各类符号之间的优先级如下（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）：

$$\varphi ::= p \mid X \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathrm{AX}\varphi \mid \nu X.\varphi$$

其中$p \in \mathscr{A}$且$X \in \mathscr{V}$。此外，公式$\nu X.\varphi$中的X总是正出现在φ中，即：φ中X的每一次出现之前都有偶数个否定符号“$\neg$”。AX和EX为时序算子。称出现在$\mu X.\varphi$和$\nu X.\varphi$中的变元X是*受约束的*（bound），且受约束的变元称为*约束变元*，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*（literal），出现在公式φ中的*原子命题和变元的集合*记为$Var(\varphi)$。

其它公式定义如下：

- $\bot =_{def} p \wedge \neg p$, $\top =_{def} \neg\bot$,
- $\varphi \wedge \psi =_{def} \neg(\neg\varphi \vee \neg\psi)$,
- $\varphi \rightarrow \psi =_{def} \neg\varphi \vee \psi$,
- $\mathrm{EX}\varphi =_{def} \neg\mathrm{AX}\neg\varphi$,
- $\mu X.\varphi(X) =_{def} \neg\nu X.\neg\varphi(\neg X)$,

其中$p \in \mathscr{A}$，$\varphi(X)$表示X出现在公式φ中，$\varphi(\neg X)$是将φ中X的所有出现同时用$\neg X$替换得到的公式。

注意：在μ-演算公式的定义中，通常考虑动作集Act和一组与$a \in Act$相关的模态词“$\langle a \rangle$”[102-104]。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$[a]\varphi$）[104]相同。

对于给定的公式φ，若出现在其中的自由变元与约束变元不同，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）。任意μ-公式可以转换为取名恰当且受保护的公式[105]，所以，在本文中所谈到的公式指的是取名恰当的、受保护的μ-公式。

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，这里也称$\mathscr{M}=(S,R,L,r)$为Kripke结构。

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，其结论也能推广到二元关系是完全的情形。

定义 2.5. 给定μ-演算公式φ、*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$\|\varphi\|_v^{\mathscr{M}}$（如果在上下文中$\mathscr{M}$是明确的，则可以省去上标）：

$$\begin{aligned}
&\|p\|_v=\{s\mid p\in L(s)\},\\
&\|X\|_v=v(X),\\
&\|\varphi_1\vee\varphi_2\|_v=\|\varphi_1\|_v\cup\|\varphi_2\|_v,\\
&\|\mathrm{AX}\varphi\|_v=\{s\mid\forall s'.(s,s')\in R\Rightarrow s'\in\|\varphi\|_v\},\\
&\|\nu X.\varphi\|_v=\bigcup\{S'\subseteq S\mid S'\subseteq\|\varphi\|_{v[X:=S']}\}.
\end{aligned}$$

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$Y\in\mathscr{V}$：

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

直观上，$\|\varphi\|_v^{\mathscr{M}}$为在$\mathscr{M}$和$v$上$\varphi$为真的状态集合。由$\mathscr{M}$、其赋值函数$v$和$\mathscr{M}$上的状态$s$构成的三元组$(\mathscr{M},s,v)$称为为赋值（当$s$为$\mathscr{M}$的根时，$(\mathscr{M},s,v)$简写为$(\mathscr{M},v)$，也称其为一个赋值）。在下文中，若$s\in\|\varphi\|_v$，则称$s$“满足”$\varphi$，记为$(\mathscr{M},s,v)\models\varphi$。用$(\mathscr{M},v)\models\varphi$代替$(\mathscr{M},r,v)\models\varphi$。当公式$\varphi$为$\mu$-句子时，可以将赋值函数$v$省略。

引理 2.4. 令$\mathscr{M}=(S,r,R,L)$、$\mathscr{M}_s=(S,s,R,L)$、$v:\mathscr{V}\to 2^S$、$s\in S$且φ为μ-公式，则：

[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$\psi$为它们之间的$*$在$\varphi*\psi$中的左辖域和右辖域。

(i) 对任意$s_1 \in S$，$s_1 \in \|\varphi\|_v^{\mathscr{M}}$当且仅当$s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$；

(ii) $(\mathscr{M}, s, v) \models \varphi$当且仅当$(\mathscr{M}_s, v) \models \varphi$。

证明. (i) **基始.** (a) $\varphi = p$，其中$p \in \mathscr{A}$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in \{s' \mid p \in L(s')\}$

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

(b) $\varphi = X$，其中$X \in \mathscr{V}$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in v(X)$

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

归纳步. (a) $\varphi = \neg\varphi_1$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \notin \|\varphi_1\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \notin \|\varphi_1\|_v^{\mathscr{M}_s}$ (归纳假设)

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

(b) $\varphi = \varphi_1 \vee \varphi_2$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in \|\varphi_1\|_v^{\mathscr{M}}$或$s_1 \in \|\varphi_2\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in \|\varphi_1\|_v^{\mathscr{M}_s}$或$s_1 \in \|\varphi_2\|_v^{\mathscr{M}_s}$ (归纳假设)

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

(c) $\varphi = \mathrm{AX}\varphi_1$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in \{s' \mid \forall s''.(s', s'') \in R \Rightarrow s'' \in \|\varphi_1\|_v^{\mathscr{M}}\}$

$\Leftrightarrow s_1 \in \{s' \mid \forall s''.(s', s'') \in R \Rightarrow s'' \in \|\varphi_1\|_v^{\mathscr{M}_s}\}$ (归纳假设)

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

(d) $\varphi = \nu X.\varphi_1$。

$s_1 \in \|\varphi\|_v^{\mathscr{M}}$

$\Leftrightarrow s_1 \in \bigcup\{S' \subseteq S \mid S' \subseteq \|\varphi_1\|_{v[X:=S']}^{\mathscr{M}}\}$

$\Leftrightarrow$ 存在$S' \subseteq S$，使得$s_1 \in S'$和$S' \subseteq \|\varphi_1\|_{v[X:=S']}^{\mathscr{M}}$

$\Leftrightarrow$ 存在$S' \subseteq S$，使得$s_1 \in S'$和$S' \subseteq \|\varphi_1\|_{v[X:=S']}^{\mathscr{M}_s}$ (归纳假设)

$\Leftrightarrow s_1 \in \bigcup\{S' \subseteq S \mid S' \subseteq \|\varphi_1\|_{v[X:=S']}^{\mathscr{M}_s}\}$

$\Leftrightarrow s_1 \in \|\varphi\|_v^{\mathscr{M}_s}$。

(ii) $(\mathscr{M},s,v)\models\varphi$
$\Leftrightarrow s\in\|\varphi\|_v^{\mathscr{M}}$
$\Leftrightarrow s\in\|\varphi\|_v^{\mathscr{M}_s}$ (i)
$\Leftrightarrow (\mathscr{M}_s,v)\models\varphi$。 □

因此，只将$(\mathscr{M},v)$作为公式的模型，记$Mod(\varphi)$为φ的模型的集合，即$Mod(\varphi)=\{(\mathscr{M},v)\mid(\mathscr{M},r,v)\models\varphi\}$（当$\varphi$为$\mu$-句子时，也可简写为$Mod(\varphi)=\{\mathscr{M}\mid(\mathscr{M},r,v)\models\varphi\}$）。其它记号与CTL情形类似，这里不再赘述。

2.3.4　μ-公式的析取范式

μ-演算的析取范式由Janin等人提出[83,105]。在给出μ-演算析取公式的定义之前，事先给出μ-公式的另一种定义，称为*覆盖-语法*（cover-syntax）。在覆盖-语法语法中，用*覆盖操作*（cover operator）集替换上述μ-公式的定义中的EX。在覆盖-语法中，

- $Cover(\emptyset)$是公式；
- 对任意$n\geq 1$，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$i=1,...,n$，存在$(s,t)\in R$使得$(\mathscr{M},t,v)\models\varphi_i$；
  - 对任意$(s,t)\in R$，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，但已证明这两种定义是等价的[83]，且*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[83]). *析取μ-公式集$\mathscr{F}_d$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 2.3. 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

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

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$\varphi$关于$Var(\varphi) \cap Var(\psi)$的均匀插值。均匀插值的形式化定义[83]如下：

定义 2.7. 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：

- $\varphi \models \theta$；
- 对任意公式ψ，若$Var(\varphi) \cap Var(\psi) \subseteq V$且$\varphi \models \psi$，则$\theta \models \psi$；
- $Var(\theta) \subseteq V$。

直观上，φ关于$Var(\varphi) \cap Var(\psi)$的均匀插值是从$\varphi$中“移除”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。

已有结论表明析取μ-公式的均匀插值是容易计算的[83]，即：

定理 2.1. 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。

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）是否可满足的技术，该技术可以追溯到1960年Davis等的工作[106]，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。

表 2.3: $R_{\mathrm{CTL}}^{\succ,S}$归结系统

(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)** $\dfrac{\Lambda \to \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{G}l, Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{F}\neg l}{Q \to \mathrm{E}_{\langle ind \rangle}(\neg \Lambda \mathrm{W} \neg l)}$.

归结在在命题逻辑和一阶逻辑中取得了丰富的成果，致使科研工作者们开始将精力致力于其它非经典逻辑中，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[107]和时序逻辑（尤其是线性时序逻辑（LTL）和CTL）中的归结[97,108]。

这里主要介绍与本文直接相关的CTL归结。CTL归结起源于BolotovF的研究[97]，之后被Zhang等人完善[99]。不论是在BolotovF的工作还是在Zhang等人的工作中，关键点都是将CTL公式转换为一个$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集。本文使用Zhang等人提出的归结系统$R_{\mathrm{CTL}}^{\succ,S}$[99]，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$1 \le i \le n$和$1 \le j \le m$。

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“$\Lambda \to \mathrm{EXEG}l$”表示如下子句集$\Lambda_{\mathrm{EG}}$：

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

其中，对任意i $(1 \le i \le n)$，

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

上面的最后两个条件确保了子句集Λ_{EG}能够蕴涵$\Lambda \to \mathrm{EXEG}l$。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-步子句集：

$$\{w^{\mathrm{A}}_{\neg l} \to \mathrm{AX}(\neg l \vee \bigvee_{j=1}^{m_i} \neg P^i_j) \mid 1 \le i \le n\},$$
$$\{\top \to \neg Q \vee \neg l \vee \bigvee_{j=1}^{m_i} \neg P^i_j \mid 1 \le i \le n\},$$
$$\{\top \to \neg Q \vee \neg l \vee w^{\mathrm{A}}_{\neg l}, w^{\mathrm{A}}_{\neg l} \to \mathrm{AX}(\neg l \vee w^{\mathrm{A}}_{\neg l})\}.$$

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-步子句集：

$$\{w^{ind}_{\neg l} \to \mathrm{E}_{\langle ind\rangle}\mathrm{X}(\neg l \vee \bigvee_{j=1}^{m_i} \neg P^i_j) \mid 1 \le i \le n\},$$
$$\{\top \to \neg Q \vee \neg l \vee \bigvee_{j=1}^{m_i} \neg P^i_j \mid 1 \le i \le n\},$$
$$\{\top \to \neg Q \vee \neg l \vee w^{\mathrm{A}}_{\neg l}, w^{ind}_{\neg l} \to \mathrm{E}_{\langle ind\rangle}\mathrm{X}(\neg l \vee w^{ind}_{\neg l})\}.$$

注意，在转换**ERES1-2**的结果为子句集的过程中会引入一个新的原子命题，即$w^{\mathrm{A}}_{\neg l}$和$w^{ind}_{\neg l}$[99]。因而，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集，即：源于$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集$S$的一个推导（derivation）是一个满足如下条件的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集序列$S_0, S_1, S_2, \ldots,$：

- $S_0 = S$，且
- $S_{i+1} = S_i \cup \{\alpha\}$ $(i \ge 0)$，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。

$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集$S$的一个反驳是一个源于$S$的推导$S_0, S_1, S_2, \ldots, S_i$，且$S_i$（$i \ge 0$）包含一个矛盾：公式$\top \to \bot$或**start** $\to \bot$。

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[99]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 2.4 (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：

$$
\begin{array}{ll}
(1)\ \textbf{start} \to x & (1,4,\textbf{SRES5}) \\
(2)\ w \to \mathrm{AX}(p \vee \neg x) & (2,3,5,\textbf{ERES1}) \\
(3)\ \top \to \neg z \vee p \vee \neg x & (2,3,5,\textbf{ERES1}) \\
(4)\ \top \to \neg z \vee l \vee w & (2,3,5,\textbf{ERES1}) \\
(5)\ w \to \mathrm{AX}(x \vee w) & (2,3,5,\textbf{ERES1}) \\
(6)\ \top \to \neg z \vee \neg x & (5,(3),\textbf{SRES8}) \\
(7)\ \textbf{start} \to \neg x & (1,(6),\textbf{SRES5}) \\
(8)\ \textbf{start} \to \bot & ((1),(7),\textbf{SRES4}).
\end{array}
$$

显然，产生了一个矛盾。因此，T_φ存在一个反驳，即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在经典逻辑和模态逻辑S5下的定义，以及基于遗忘的SNC（WSC）计算方法。

2.5.1　经典逻辑遗忘

遗忘一词起源于经典逻辑（包括命题逻辑和一阶逻辑）[37]。给定一个命题公式φ和一个原子命题p，从φ中遗忘p 得到的结果为$Forget(\varphi,\{p\}) \equiv \varphi[p/\top] \vee \varphi[p/\bot]$。

例 2.5. 某学校有a和b两个食堂，学生只能去a或去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$\varphi = (a \vee b) \wedge (f \to a) \wedge (r \to b)$。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（$f$）。这一计算过程表示如下：

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

直观上来看，这个结果应该比原始公式φ弱，但是能够蕴含同样的任何不包含f的句子（sentence），也就是说遗忘只影响与f相关的语义。这一性质可由互模拟一词来表示，解释之间的互模拟为：给定原子命题p，如果对任意$q \in \mathscr{A} - \{p\}$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。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[37]为：

定义 2.8. *给定一个句子（sentence）φ和实例化原子p，φ'是从φ中遗忘p的结果当且仅当对任意结构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，并且将这一结果记为$\varphi[P(\vec{t})]$。

例 2.6. 令$\varphi = J(mo) \vee J(fa) \vee B(sm)$、$p = J(mo)$，则：

$$\begin{aligned}\varphi[p] \equiv\ & (mo = mo \wedge J(mo)) \vee (mo \neq mo \wedge J(mo)) \vee \\ & (mo = fa \wedge J(mo)) \vee (mo \neq fa \wedge J(fa)) \vee B(sm).\end{aligned}$$

$$\begin{aligned}&Forget(\varphi, p) \equiv \varphi[p][p/\top] \vee \varphi[p][p/\bot] \\ &\equiv (mo = mo \wedge \top) \vee (mo \neq mo \wedge \top) \vee (mo = fa \wedge \top) \vee (mo \neq fa \wedge J(fa)) \vee B(sm) \\ &\vee (mo = mo \wedge \bot) \vee (mo \neq mo \wedge \bot) \vee (mo = fa \wedge \bot) \vee (mo \neq fa \wedge J(fa)) \vee B(sm) \\ &\equiv (mo = mo) \vee (mo \neq mo) \vee (mo = fa) \vee (mo \neq fa \wedge J(fa)) \vee B(sm).\end{aligned}$$

然而，遗忘一个关系（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。而遗忘谓词的定义与遗忘实例的定义类似，只是将$M \sim_p M'$变为$M \sim_P M'$。

研究表明，从句子φ中遗忘谓词P的结果为$Forget(\varphi, P) = (\exists R)\varphi[P/R]$[37]，其中$P$是$n$-元谓词，$R$是$n$-元谓词变量。正如前文所说，一阶逻辑遗忘不是封闭的，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[109]。

定义 2.9. 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2.9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，本文采用上述互模拟的方式。因此，这里不再赘述其它定义，感兴趣的读者可以参考Eiter的文章[68]。

在描述逻辑中，如果遗忘的结果可以用当前讨论的描述逻辑来表示，则该结果就是一个均匀插值。而判定均匀插值是否存在通常是困难的，如：在ALC和EL中是双指数时间的。因此，描述逻辑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ALC到ALC_ν[110]）或引入新的辅助符号[111]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[112]和基于Ackermann引理的FAME[113]。

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遗忘。与经典逻辑遗忘相似，S5遗忘也用互模拟来定义。

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

原子命题集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$w_1, w_2, V \subseteq \mathscr{A}$。给定原子命题集$V \subseteq \mathscr{A}$、两个K-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$\mathscr{M} \leftrightarrow_V \mathscr{M}'$）[49]，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）。值得注意的是，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$X$中组成世界的原子命题构成的集合。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：如果两个K-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[49]：

定义 2.10 (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$\varphi' = KForget(\varphi, V)$，$V \subseteq \mathscr{A}$是原子命题集。知识遗忘满足以下性质：

(**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)。

直观上，(**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，遗忘结果能推出所有与V无关且能被φ推出的结果，但不能推出所有与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一个视图，达到了隐私保护的作用。

这四个性质与知识遗忘的关系如下所述[49]：

定理 2.2. 给定公式φ和φ'，$V \subseteq \mathscr{A}$为原子命题集。下面的陈述是等价的：

(i) $\varphi' \equiv KForget(\varphi, V)$,
(ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi$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。

任意S5公式都能转换为与之等价的模态合取范式（MCNF）[114]，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）[49,51]:

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

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。从公式(2.10)中遗忘原子命题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公式遗忘都能转换为命题逻辑公式遗忘，而命题逻辑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，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，只有一种——文字的析取。因此，归结规则比较简单，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 2.9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。但是在一阶情形下的归结系统R[47]是可靠的且归结反驳是完备的。

上一节已经说明任意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的归结系统RS5[107]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为子句集；对于子句集S，$\mathrm{B}(S)$表示$\mathrm{B}(\bigwedge S)$；$E[\psi]$表示ψ是E的子公式。

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（subsume）$C'$：

- 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”（suppressing）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，$V \subseteq \mathscr{A}$为原子命题集，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：

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

基于上述归结系统RS5，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，原子命题集

输出:

$KForget(\Gamma, V)$：从Γ中遗忘V中原子的结果

1: 将Γ转换为模态子句集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[22]，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 2.11. 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$\varphi \models q \to \phi$（$\varphi \models \phi \to q$），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）$\phi'$，有$\varphi \models \phi \to \phi'$（$\varphi \models \phi' \to \phi$），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具有对偶关系，且任意公式的SNC（WSC）都能转换成原子命题的形式计算[22]。因此，这里只讨论原子命题情形下SNC（WSC）的定义及其计算。

定理 2.3. 给定命题公式φ、原子命题集$V \subseteq Var(\varphi)$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 2.3表明可以用遗忘计算SNC和WSC。基于遗忘的计算SNC（WSC）的详细算法如算法 2.2所示，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$S[X/Y] = \{\varphi[X/Y] \mid \varphi \in S\}$。

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，原子命题集，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

在一阶逻辑中，SNC（WSC）的定义和命题逻辑下相似，也可用遗忘来计算[82]。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 2.4. *对任意一阶公式α、关系符号集P和句子Th:*

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

其中$\overline{\Phi}$是出现在$Th \wedge \alpha$中，且不出现在P中的关系符号集。

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，这里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：

定理 2.5. 给定*S5*公式Γ和原子命题集$V \subseteq \mathscr{A}$，$q \in Var(\Gamma) - V$，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，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$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集。随后对CTL中的归结系统和归结过程做了详细介绍，为下文使用基于归结的方法计算CTL遗忘做铺垫。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遗忘的定义、基本公设及相关算法，并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　CTL和μ-演算遗忘理论

从一个公式中“遗忘”一些原子命题得到的结果应该不“违背”定义在其它原子命题集上的公式，即：对于其它原子命题集上的公式，原公式能够逻辑蕴涵它当且仅当遗忘结果能逻辑蕴涵它。从模型的角度来讲，遗忘结果的模型与原公式的模型在“除去”被遗忘的原子命题之后是相互模拟的。互模拟描述的是两个在行为上能够相互替代的转换系统[24]。在本文中，转换系统（包括反应式系统）被描述成为*Kripke*结构。因此，为了定义*CTL*遗忘，这部分给出在给定原子命题集上*Ind*-结构之间互模拟的定义及其性质。此外，μ-演算是一种表达能力比*CTL*强的语言，且其公式中包含自由变元。因而，本章扩展上述提到的互模拟到变元上。

根据上述两种互模拟，给出了*CTL*和μ-演算遗忘的定义。与后面章节将要讲述的约束*CTL*遗忘相对应，本章探索没有约束的遗忘的一般性质。

本章其余部分组织如下：第*3.1*节首先定义V-互模拟和互模拟等价，并探索其性质；其次，定义*CTL*遗忘，并探索遗忘的基本性质和公设。第*3.2*节定义变元-命题-互模拟，并证明其对μ-公式具有不变性；然后给出μ-公式遗忘的定义及相关性质。最后，总结本章工作。

3.1　CTL遗忘理论

3.1.1　互模拟

这部分给出定义在给定原子命题集V上的互模拟概念，本文称之为V-互模拟。尽管zhang等人在S5中给出了相似的概念[48]，但是如在基础知识部分所述，*S*5的语义定义在一种特殊的Kripke结构（K-解释）下，其不具有一般性。因此，这里定义一种更一般的V-互模拟。

定义 3.1 (V-互模拟). 给定原子命题集$V \subseteq \mathscr{A}$、索引集合$I \subseteq Ind$和初始*Ind*-结构$\mathscr{M}_i = (S_i, R_i, L_i, [_]_i, s_0^i)\ (i = 1,2)$。$\mathscr{B}_V \subseteq S_1 \times S_2$为二元关系，对任意$s_1 \in S_1$和$s_2 \in S_2$，若$(s_1, s_2) \in \mathscr{B}_V$，则：

(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$(r_1, r_2) \in \mathscr{B}_V$。

那么，称$\mathscr{B}_V$是$\mathscr{M}_1$和$\mathscr{M}_2$之间的一个V-互模拟关系。

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$j\geq 1$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$\mathscr{K}_{i,j}=(\mathscr{M}_i,s_{i,j})$。

直观上，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$\overline{V}$上是“互模拟”的。当$V=\emptyset$，V-互模拟即为一般的互模拟。下文中，若能从上下文中明确初始Ind-Kripke结构，则将$\mathscr{K}_1\leftrightarrow_V\mathscr{K}_2$简写为$s_1\leftrightarrow_V s_2$。

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如图中虚线所示，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[24]的，即不$\emptyset$-互模拟。

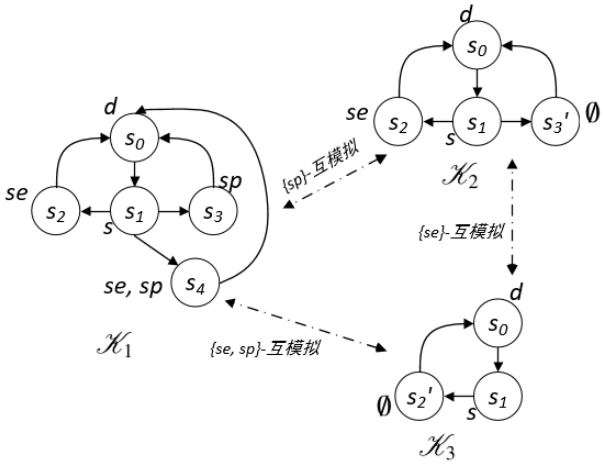


图 3.1: κ-结构之间的V-互模拟关系示意图

下述命题给出了V-互模拟关系的一些关键性质。

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$(w_2,w_3) \in \mathscr{B}''$，其中$w_i \in S_i$ $(i=1,2,3)\}$。为了证明$\mathscr{K}_1 \leftrightarrow_{V_1 \cup V_2} \mathscr{K}_3$，下证$\mathscr{B}'$ 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$(s_1,s_3) \in \mathscr{B}'$。对所有$(w_1,w_3) \in \mathscr{B}'$，存在$w_2 \in S_2$使得$(w_1,w_2) \in \mathscr{B}$和$(w_2,w_3) \in \mathscr{B}''$，且：

(a) 存在一个$w_2 \in S_2$使得$(w_1,w_2) \in \mathscr{B}$ 且$(w_2,w_3) \in \mathscr{B}''$。由$w_1 \leftrightarrow_{V_1} w_2$可知；$L_1(w_1) - V_1 = L_2(w_2) - V_1$；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$(u_1,u_2) \in \mathscr{B}$；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$(s_1,s_2) \in \mathscr{B}_{V_1}$。下证$\mathscr{B}_{V_1}$ 也是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的一个V_2-互模拟关系。对任意$(w_1,w_2) \in \mathscr{B}_{V_1}$，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$(r_1,r_2) \in \mathscr{B}_{V_1}$；
- $\forall r_2 \in S_2$，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) 因为$s_1 \leftrightarrow_{V_1} s_2$ （即：$\mathscr{K}_1 \leftrightarrow_{V_1} \mathscr{K}_2$），可以从$V$-互模拟的定义易证(v)。 □

在命题3.1中，性质(iii)-(v)的含义比较直观。性质(i)表示如果一个结构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$\mathscr{K}_1$和$\mathscr{K}_3$ 是$V_1 \cup V_2$-互模拟的（如图3.1所示）。这一性质对遗忘性质的证明至关重要。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。

从互模拟的定义来看，如果两个结构是V-互模拟的，那么对于与V中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$V \subseteq \mathscr{A}$是原子命题集，$\mathscr{K}_i$ $(i=1,2)$是两个具有V-互模拟关系的Ind-结构，即：$\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$。若$\Phi$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. 下面通过归纳CTL公式的结构证明这一结论。假设$Var(\Phi) \cap V = \emptyset$，$\mathscr{K}_1 = (\mathscr{M},s)$和$\mathscr{K}_1 = (\mathscr{M}',s')$。

基始. $\Phi = p$ （$p \in \mathscr{A} - V$）。

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

归纳步. (a) $\Phi=\neg\psi$。

$(\mathscr{M},s)\models\Phi$当且仅当$(\mathscr{M},s)\not\models\psi$
$\Leftrightarrow(\mathscr{M}',s')\not\models\psi$　　(归纳假设)
$\Leftrightarrow(\mathscr{M}',s')\models\Phi$。

(b) $\Phi=\psi_1\vee\psi_2$。

$(\mathscr{M},s)\models\Phi$
$\Leftrightarrow(\mathscr{M},s)\models\psi_1$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)
$\Leftrightarrow(\mathscr{M}',s')\models\Phi$。

(c) $\Phi=\mathrm{EX}\psi$。

$(\mathscr{M},s)\models\Phi$
$\Rightarrow$存在一条路径$\pi=(s,s_1,\dots)$使得$(\mathscr{M},s_1)\models\psi$
$\Rightarrow$存在一条路径$\pi'=(s',s'_1,\dots)$使得$\pi\leftrightarrow_V\pi'$且$(\mathscr{M},s_1)\models\psi$　　($s\leftrightarrow_V s'$，命题 3.1)
$\Rightarrow s_1\leftrightarrow_V s'_1$且$(\mathscr{M},s_1)\models\psi$　　($\pi\leftrightarrow_V\pi'$)
$\Rightarrow(\mathscr{M}',s'_1)\models\psi$　　(归纳假设)
$\Rightarrow(\mathscr{M}',s')\models\Phi$。

另一个方向可类似证明。

(d) $\Phi=\mathrm{EG}\psi$。

$(\mathscr{M},s)\models\Phi$
$\Rightarrow$存在一条路径$\pi=(s=s_0,s_1,\dots)$，使得对任意$i\geq 0$，都有$(\mathscr{M},s_i)\models\psi$
$\Rightarrow$存在一条路径$\pi'=(s'=s'_0,s'_1,\dots)$，使得$\pi\leftrightarrow_V\pi'$，且对任意$i\geq 0$，都有$(\mathscr{M},s_i)\models\psi$
($s\leftrightarrow_V s'$，命题 3.1)
$\Rightarrow$对于任意$i\geq 0$，都有$s_i\leftrightarrow_V s'_i$和$(\mathscr{M},s_i)\models\psi$　　($\pi\leftrightarrow_V\pi'$)
$\Rightarrow$对于任意$i\geq 0$，都有$(\mathscr{M},s'_i)\models\psi$　　(归纳假设)
$\Rightarrow(\mathscr{M}',s')\models\Phi$。

另一个方向可类似证明。

(e) $\Phi=\mathrm{E}(\psi_1\mathrm{U}\psi_2)$。

$(\mathscr{M},s)\models\Phi$
$\Rightarrow$存在一条路径$\pi=(s=s_0,s_1,\dots)$和$i\geq 0$，使得$(\mathscr{M},s_i)\models\psi_2$，且对所有的$0\leq j<i$都有$(\mathscr{M},s_j)\models\psi_1$
$\Rightarrow$存在一条路径$\pi'=(s'=s'_0,s'_1,\dots)$，使得$\pi\leftrightarrow_V\pi'$和$(\mathscr{M}',s'_i)\models\psi_2$，且对所有的$0\leq j<$

i都有$(\mathscr{M}', s'_j) \models \psi'_1$ ($s \leftrightarrow_V s'$，归纳假设，命题 3.1)
$\Rightarrow (\mathscr{M}', s') \models \Phi$。

另一个方向可类似证明。 □

上述定理中，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$[1] = \{(s_0, s_1), (s_1, s_1)\}$。显然，$\mathscr{K} \models \phi$。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$\mathrm{IR}(\varphi_2, \{sp\})$。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)**.** 给定原子命题集$V \subseteq \mathscr{A}$，公式φ和ψ。若对任意$\mathscr{K} \models \varphi$，都存在一个$\mathscr{K}' \models \psi$，使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$；且对任意$\mathscr{K}' \models \psi$，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$V \subseteq \mathscr{A}$，$\leftrightarrow_V$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$\psi \equiv_V \xi$。则对任意$\mathscr{K} \models \varphi$，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，$\mathscr{K} \leftrightarrow_V \mathscr{K}''$。类似地，对任意$\mathscr{K}'' \models \xi$，存在$\mathscr{K} \models \varphi$ 使得$\mathscr{K}'' \leftrightarrow_V \mathscr{K}$。这表明$\equiv_V$是传递的。因此，$\equiv_V$是等价关系。 □

此外，由互模拟等价的定义和上述结论可得以下推论。

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$\mathscr{K} \leftrightarrow_V \mathscr{K}$。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$(\psi,\emptyset)$，由定理 3.1可知$\mathscr{K} \models \psi$。类似地，对任意$\mathscr{K}' \models \psi$，存在$\mathscr{K} \models \varphi$使得$\mathscr{K} \leftrightarrow_{\emptyset} \mathscr{K}'$，因此$\mathscr{K}' \models \varphi$。

(iii) 对任意$\mathscr{K} \models \varphi$，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$\mathscr{K}' \models \psi$，存在$\mathscr{K} \models \varphi$，使得$\mathscr{K} \leftrightarrow_{V_1 \cup V_2} \mathscr{K}'$。因此，$\varphi \equiv_{V_1 \cup V_2} \psi$。

同理可证(iv)。 □

请注意，在上述结论(ii)中“φ和ψ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. 令$T_0, T_1, \ldots, T_n = T_\varphi$为转换过程产生的公式集合序列。$T_0 = \{\mathrm{AG}(\mathbf{start} \rightarrow p), \mathrm{AG}(p \rightarrow \mathbf{simp}(\mathbf{nnf}(\varphi)))\}$，其中$p$是不出现在$\varphi$中的原子命题，且对任意$i$（$0 \leq i < n$），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，所有公式都是否定范式形式。

为了证明命题中的结论，只需证明，对任意i（$0 \leq i < n$），$T_i \equiv_U T_{i+1}$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）从T_0到T_φ的部分。假设$\mathscr{M}_1 = (S_1, R_1, L_1, [_]_1, s_1)$和$\mathscr{M}_2 = (S_2, R_2, L_2, [_]_2, s_2)$。

（1）下证$\varphi \equiv_{\{p\}} T_0$。

$(\Rightarrow)$ $\forall(\mathscr{M}_1, s_1) \in Mod(\varphi)$，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），$\mathscr{M}_2$中的其它元素都与$\mathscr{M}_1$中的元素相同。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**的语义可知$(\mathscr{M}_1, s_1) \models \varphi$。

（2）下证对任意i（$0 \leq i < n$），有$T_i \equiv_U T_{i+1}$成立，其中$T_{i+1} = (T_i - \{\psi\}) \cup R_i$。这里，用$\psi \rightarrow_t R_i$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，$T_{i+1} = X \cup R_i$，$X = T_i - \{\psi\}$）。下面证明规则$t \in \{$**Trans(1), Trans(4), Trans(6)**$\}$的情形，其它情形可以类似地证明。

（a）$t =$ **Trans(1)**。

$(\Rightarrow)$ $(\mathscr{M}_1, s_1) \in Mod(T_i)$，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$\pi = (s_{1,1}, s_{1,2}, \ldots)$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AG$(q \to \text{EX}\varphi)$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$i \geq j$，若$(s_{1,i}, s') \in R_s$ $(s \neq s_{1,j})$，则$s' = s_{1,i+1}$”；
- $R_y = \{(s_x, s_y) \mid s_x \in S$，若对任意$(s'_1, s'_2) \in \bigcup_{s\in S} R_s$，有$s'_1 \neq s_x$，则找一个状态$s_y \in S_2$，使得$(s_x, s_y) \in R_2\}$。

显然，$(\mathscr{M}_1, s_1) \leftrightarrow_{\emptyset} (\mathscr{M}_2, s_2)$

$\Rightarrow$ 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$(\mathscr{M}_2, s_{2,j}) \models \text{E}_{\langle ind\rangle}\text{X}\varphi$

$\Rightarrow (\mathscr{M}_2, s_2) \models \text{AG}(q \to \text{E}_{\langle ind\rangle}\text{X}\varphi)$

$\Rightarrow (\mathscr{M}_2, s_1) \models X \wedge \text{AG}(q \to \text{E}_{\langle ind\rangle}\text{X}\varphi)$

$(\Leftarrow) \forall(\mathscr{M}_1, s_1) \in Mod(T_{i+1})$，即$(\mathscr{M}_1, s_1) \models X \wedge \text{AG}(q \to \text{E}_{\langle ind\rangle}\text{X}\varphi)$

$\Rightarrow (\mathscr{M}_1, s_1) \models X$且$(\mathscr{M}_1, s_1) \models \text{AG}(q \to \text{E}_{\langle ind\rangle}\text{X}\varphi)$

$\Rightarrow$ 对任意以s_1为起点的路径上的任意状态$s_{1,j}$，$(\mathscr{M}_1, s_{1,j}) \models \neg q$或$(\mathscr{M}_1, s_{1,j}) \models \text{EX}\varphi$

$\Rightarrow (\mathscr{M}_1, s_1) \models \text{AG}(q \to \text{EX}\varphi)$

$\Rightarrow (\mathscr{M}_1, s_1) \models X \wedge \text{AG}(q \to \text{EX}\varphi)$。

（b）$t =$ **Trans(4)**。

$(\Rightarrow) (\mathscr{M}_1, s_1) \in Mod(T_i)$，即$(\mathscr{M}_1, s_1) \models X \wedge \text{AG}(q \to \varphi_1 \vee \varphi_2)$

$\Rightarrow (\mathscr{M}_1, s_1) \models X$，且$\forall s'_1 \in S_1$，$(\mathscr{M}_1, s'_1) \models q \to \varphi_1 \vee \varphi_2$

$\Rightarrow (\mathscr{M}_1, s'_1) \models \neg q$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$[_]_1$相同，且$s_2 = s_1$；
- L_2与L_1相同，除了：对任意$s'_1 \in S_2$，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，$(\mathscr{M}_2, s'_1) \models (q \to \varphi_1 \vee p) \wedge (p \to \varphi_2)$且$(\mathscr{M}_1, s_1) \leftrightarrow_{\{p\}} (\mathscr{M}_2, s_2)$，所以，$(\mathscr{M}_2, s_2) \models T_{i+1}$。

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

（c）t=**Trans(6)**。

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

$(\Rightarrow) (\mathscr{M}_1, s_1) \in Mod(T_i)$，即$(\mathscr{M}_1, s_1) \models X \wedge \text{AG}(q \to \text{E}_{\langle ind\rangle}\text{X}\varphi)$

$\Rightarrow (\mathscr{M}_1, s_1) \models X$，且对任意$s'_1 \in S$，$(\mathscr{M}_1, s'_1) \models q \to \text{E}_{\langle ind\rangle}\text{X}\varphi$

$\Rightarrow (\mathscr{M}_1, s_1') \models \neg q$，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相同，除了：对任意$s_1' \in S_2$，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，$(\mathscr{M}_2, s_2) \models \mathrm{AG}(q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}p) \wedge \mathrm{AG}(p \to \varphi)$、$(\mathscr{M}_2, s_2) \models T_{i+1}$且$(\mathscr{M}_1, s_1) \leftrightarrow_{\{p\}} (\mathscr{M}_2, s_2)$ $(s_2 = s_1)$。

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

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

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

其中x, y, z为新引入的原子命题。

3.1.2　遗忘算子及其性质

这部分给出CTL遗忘的定义及其相关性质。

定义 3.3 (遗忘，forgetting). 令V是$\mathscr{A}$的子集，Φ是公式。如果公式ψ满足下面条件：

- 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 $\mathscr{K}' \leftrightarrow_V \mathscr{K}\}$。

那么，称ψ为从Φ中遗忘V后得到的结果。

从定义3.3可以看出，如果两个公式ψ和ψ'都是Φ遗忘V中元素后得到的结果，则$Mod(\psi) \equiv Mod(\psi')$。从这个角度来看，$\Phi$遗忘$V$中元素后得到的所有结果是语义等价的。将遗忘结果记为$\mathrm{F}_{\mathrm{CTL}}(\phi, V)$，不做其它说明的情况下，这表示$\phi$遗忘$V$是CTL可表示的。此外，当$V$中只包含一个元素时，可以省略集合符号，即：

$$\mathrm{F}_{\mathrm{CTL}}(\Phi, \{p\}) \equiv \mathrm{F}_{\mathrm{CTL}}(\Phi, p).$$

遗忘的定义表明，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$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，那么称$\psi$为$\varphi$遗忘$V$中原子命题后得到的结果。为刻画S5逻辑下遗忘的

直观含义，Zhang等人提出了遗忘的特性——这些特性被称为遗忘理论公设（forgetting postulates）[48]。类似地，给定CTL公式φ、$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$、原子命题集$V \subseteq \mathscr{A}$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：$\mathrm{IR}(\varphi', V)$。

其中**(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，遗忘结果能蕴涵所有与V无关且能被φ蕴涵的结果，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一个视图，达到隐私保护的作用。下面的定理表明上述公设对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'，$V \subseteq \mathscr{A}$为原子命题集。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. $(i) \Leftrightarrow (ii)$. 通过证明如下等式证明该结论。

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

$(\Rightarrow)$ $\mathscr{K}' \models \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$
$\Rightarrow \exists \mathscr{K}$ 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$\phi' \in A$，都有$\varphi \models \phi'$，所以，$\varphi \models \varphi'$。其次，对任意公式$\phi$，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$\phi' \in A$，都有$\mathrm{IR}(\varphi', V)$，所以，$\mathrm{IR}(\phi', V)$。

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

定理 3.2并不表明遗忘的结果一定存在。事实上，存在一个CTL公式和原子命题集，使得从该公式里遗忘该集合得到的结果不可用CTL公式表达。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[55]：

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

Maksimova证明了$\varphi(p,x) \wedge \varphi(p,y) \models x \leftrightarrow y$，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性质。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)
$\Rightarrow \varphi(p,x) \vdash x \rightarrow \psi(p)$且$\varphi(p,y) \vdash \psi(p) \rightarrow p(y)$，这表明$\varphi(p,x) \vdash \psi(p) \rightarrow p(x)$
$\Rightarrow \varphi(p,x) \vdash x \leftrightarrow \psi(p)$，这是一个矛盾。 □

从命题公式φ中遗忘原子命题p为：$Forget(\varphi, \{p\}) \equiv \varphi[p/\bot] \vee \varphi[p/\top]$。本文遗忘的定义对公式为命题逻辑公式时，与Lin等人于1994提出的命题逻辑遗忘一致，即：本文的CTL遗忘扩展了命题逻辑遗忘。

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

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

证明. 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$L \in \mathscr{M}$，$L' \in \mathscr{M}'$），这表明$(\mathscr{M},s) \models Forget(\varphi,V)$。

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$\varphi$的一个模型$(\mathscr{M}',s')$（$\mathscr{M}' = (S',R',L',[_]',s')$），使得$L(s) - V = L'(s') - V$。如下构建一个初始Ind-结构$(\mathscr{M}_1,s_1)$，使得$\mathscr{M}_1 = (S_1,R_1,L_1,[_],s_1)$，其中：

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

- $S_1 = (S - \{s\}) \cup \{s_1\}$；
- R_1由将R出现的s替换为s_1得到；
- 对于S_1中任意一个状态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。□

遗忘的另一个重要性质与V-无关性密切相关。对于给定的公式$\psi = \varphi \wedge (q \leftrightarrow \alpha)$，如果$\mathrm{IR}(\varphi \wedge \alpha, \{q\})$，那么$\psi$遗忘$q$后得到的结果为$\varphi$。这一性质与后文将要介绍的*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。

引理 3.2. *给定两个公式φ和α，原子命题$q \notin (Var(\varphi) \cup Var(\alpha))$，则：*

$$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q) \equiv \varphi.$$

证明. ($\Rightarrow$) 令$\varphi' = \varphi \wedge (q \leftrightarrow \alpha)$。对于$\mathrm{F}_{\mathrm{CTL}}(\varphi', q)$的任意模型$(\mathscr{M}, s)$，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$(\mathscr{M}', s') \models \varphi'$。显然，$(\mathscr{M}', s') \models \varphi$。此外，由于$\mathrm{IR}(\varphi, \{q\})$且$(\mathscr{M}, s) \leftrightarrow_{\{q\}} (\mathscr{M}', s')$，由定理 3.1可知$(\mathscr{M}, s) \models \varphi$。

($\Leftarrow$) 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：

$$L' : S \to 2^{\mathscr{A}} \text{ 且 } \forall s^* \in S, \text{ 若 } (\mathscr{M}, s^*) \not\models \alpha, \text{ 则} L'(s^*) = L(s^*) - \{q\} \text{ 否则 } L'(s^*) = L(s^*) \cup \{q\},$$
$$\text{若 } (\mathscr{M}, s) \models \alpha, \text{ 则} L'(s) = L(s) \cup \{q\}, \text{ 否则 } L'(s) = L(s) - \{q\}.$$

显然，$(\mathscr{M}', s) \models \varphi$、$(\mathscr{M}', s) \models q \leftrightarrow \alpha$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，$(\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q)$。□

根据遗忘的定义可以看出，从一个公式里遗忘某个原子命题集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，遗忘原子命题集可分解为遗忘单个原子命题。

命题 3.4 (分解性，Decomposition)**.** *对于给定的公式φ，原子命题集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. 一方面，令$\mathscr{M}_1=(S_1,R_1,L_1,[_],s_1)$是一个初始Ind-Kripke结构，且$(\mathscr{M}_1,s_1)$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$(\mathscr{M}_1,s_1)\leftrightarrow_{\{p\}\cup V}(\mathscr{M},s)$。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$p\in L_1(s_1)$，
- 对任意$q\in V$，$q\in L_2(s_2)$当且仅当$q\in L(s)$，
- 对其它原子命题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) 其它情形：

- 对所有状态对(w,w_1)，若$w\in S$，$w_1\in S_1$且$w\leftrightarrow_{\{p\}\cup V}w_1$，如下构造$w_2\in S_2$：
  - $p\in L_2(w_2)$当且仅当$p\in L_1(w_1)$，
  - 对任意$q\in V$，$q\in L_2(w_2)$当且仅当$q\in L(w)$，
  - 对其它原子命题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由w_1构造而成，且w_2'由w_1'构造而成，则令$(w_2',w_2)\in R_2$。

(3) 删除S_2和R_2中重复的元素。

由此可知，$(\mathscr{M},s)\leftrightarrow_{\{p\}}(\mathscr{M}_2,s_2)$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，若$(\mathscr{M}_1,s_1)$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，$(\mathscr{M}_1,s_1)\models\mathrm{F}_{\mathrm{CTL}}(\varphi,\{p\}\cup V)$。 □

由上面的命题不难看出，从公式中遗忘原子命题集中的元素，可以将该集合拆成两个集合遗忘。

推论 3.2. *对于给定的公式φ，原子命题集V_1和V_2，有如下结论：*

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

类似于被遗忘的原子命题集能被拆成两个集合的遗忘，下面介绍在某些情况下，

从带时序算子的公式中遗忘一些原子命题，可以将这些时序算子提到遗忘操作的前面。

命题 3.5. 令$V \subseteq \mathscr{A}$为原子命题集，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) $(\Rightarrow)$ $(\mathscr{M}, s) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V)$
$\Rightarrow$ 存在$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$且$(\mathscr{M}', s') \models \mathrm{AX}\phi$
$\Rightarrow (\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$，且对任意$(s', s'') \in R'$，有$(\mathscr{M}', s'') \models \phi$ $(R' \in \mathscr{M}')$
$\Rightarrow$ 对任意$(s, s_1) \in R$，存在$(s', s_1') \in R'$，使得$(\mathscr{M}, s_1) \leftrightarrow_V (\mathscr{M}', s_1')$，且对任意$(s', s'') \in R'$，有$(\mathscr{M}', s'') \models \phi$
$\Rightarrow$ 对任意$(s, s_1) \in R$，有$(\mathscr{M}, s_1) \leftrightarrow_V (\mathscr{M}', s_1')$且$(\mathscr{M}', s_1') \models \phi$
$\Rightarrow$ 对任意$(s, s_1) \in R$，$(\mathscr{M}, s_1) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$
$\Rightarrow (\mathscr{M}, s) \models \mathrm{AXF}_{\mathrm{CTL}}(\phi, V)$。

$(\Leftarrow)$ $(\mathscr{M}, s) \models \mathrm{AXF}_{\mathrm{CTL}}(\phi, V)$
$\Rightarrow$ 对任意$(s, s') \in R$，有$(\mathscr{M}, s') \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$ $(R \in \mathscr{M})$
$\Rightarrow$ 对任意$(s, s') \in R$，有$(\mathscr{M}, s') \leftrightarrow_V (\mathscr{M}', s'')$且$(\mathscr{M}', s'') \models \phi$
$\Rightarrow$ 对任意$i \geq 0$，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$(\mathscr{M}', s'') \models \phi$ $(R' \in \mathscr{M}')$
$\Rightarrow$ 存在$(s, s_1) \in R$和$(s', s_1') \in R'$，使得$(\mathscr{M}, s_1) \leftrightarrow_V (\mathscr{M}', s_1')$，且存在$(s', s'') \in R'$，使得$(\mathscr{M}', s'') \models \phi$ $(R \in \mathscr{M})$

$\Rightarrow$ 存在$(s,s_1)\in R$和$(\mathscr{M},s_1)\leftrightarrow_V(\mathscr{M}',s_1')$，使得$(\mathscr{M}',s_1')\models\phi$

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

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

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

$\Rightarrow$ 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$\mathscr{M}^*=(S^*,R^*,L^*,[_],s)$，

- $S^*=S\cup S'$，
- $R^*=\{(s,s'')\}\cup R\cup R'$，
- $L^*=L\cup L'$ 且$L^*(s)=L(s)$，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$i\geq 1$，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$(\mathscr{M},s_i)\models \mathrm{F}_{\mathrm{CTL}}(\phi,V)$ $(\mathrm{IR}(\mathrm{F}_{\mathrm{CTL}}(\phi,V),V))$

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

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

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

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

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

$\Rightarrow \forall t\in S$，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$t_i\in S_i'$，$L^*(t_i)=L_i'(t_i)$。

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

(vi) 可类似(v)来证明。 □

推论 3.3. 给定命题集合$V \subseteq \mathscr{A}$，命题公式φ和*CTL*公式$\mathrm{AG}\psi$，则

$$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \mathrm{AG}\psi) \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \psi, V) \wedge \mathrm{AGF}_{\mathrm{CTL}}(\psi, V).$$

证明. $(\mathscr{M}, s_0) \models \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \mathrm{AG}\psi)$
$\Leftrightarrow$ 存在$(\mathscr{M}', s_0') \leftrightarrow_V (\mathscr{M}, s_0)$，使得$(\mathscr{M}', s_0') \models \varphi \wedge \mathrm{AG}\psi$
$\Leftrightarrow$ 存在$(\mathscr{M}', s_0') \leftrightarrow_V (\mathscr{M}, s_0)$，使得$(\mathscr{M}', s_0') \models \varphi \wedge \psi$且$(\mathscr{M}', s_0') \models \mathrm{AG}\psi$
$\Leftrightarrow (\mathscr{M}, s_0) \models \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \psi, V)$且$(\mathscr{M}, s_0) \models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AG}\psi, V)$
$\Leftrightarrow (\mathscr{M}, s_0) \models \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \psi, V) \wedge \mathrm{AGF}_{\mathrm{CTL}}(\psi, V)$。 □

3.2 μ-演算遗忘理论

μ-演算是一种表达能力较强的逻辑语言，它能表达CTL不能表达的一些性质，例如：Kripke结构中有一条路径，在这条路径上，基数位置的状态满足公式$\neg q \wedge \neg p$，但是偶数位置的状态满足$q \wedge p$。这一性质不能用CTL公式来表达，但是可以用μ-演算公式表达如下：

$$\varphi = \nu X.(p \wedge q) \wedge \mathrm{EX}(\neg p \wedge \neg q) \wedge \mathrm{EXEX}X.$$

这种情形在日常生活中是很常见的，如：偏序关系$(\mathbb{N}, \leq)$（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不能表达具有有规律的性质[115]，其主要原因是

> 对于给定的原子命题p，任意包含n个“X”时序算子的命题时序公式（*proposinal temporal formula*，*PTL*），对序列“$p^i(\neg p)p^w$”有相同的真值，其中$i > n$。

因而得出如下结论：

> 对任意$m \geq 2$，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。

本小节给出μ-演算遗忘的定义，并说明本文所定义的μ-演算遗忘与文章[83]中定义的均匀插值具有对偶关系。此时，本文给出的遗忘性质也是均匀插值所具有的性质，这为μ-演算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，本文给出计算遗忘的方法。形成了遗忘和均匀插值之间相辅相成的作用。

本小节的组织结构如下。首先，给出变元-命题-互模拟的定义及证明变元-命题-互模拟对μ-公式是不变的；根据变元-命题-互模拟定义μ-演算遗忘；然后，探讨遗忘的性质，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。

3.2.1　变元-命题-互模拟

与CTL遗忘相似，这里先给出V-互模拟的定义。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 3.4 (V-互模拟). 给定原子命题集$V\subseteq\mathscr{A}$和两个*Kripke*结构$\mathscr{M}_1$和$\mathscr{M}_2$。若$\mathscr{B}\subseteq S_1\times S_2$满足下面几个条件：

- $r_1\mathscr{B}r_2$，
- 对任意$s\in S_1$和$t\in S_2$，若$s\mathscr{B}t$，则对任意$p\in\mathscr{A}-V$，有$p\in L_1(s)$当且仅当$p\in L_2(t)$，
- 若$(s,s')\in R_1$和$s\mathscr{B}t$，则存在一个t'，使得$s'\mathscr{B}t'$和$(t,t')\in R_2$，且
- 若$s\mathscr{B}t$和$(t,t')\in R_2$，则存在一个s'，使得$(s,s')\in R_1$和$t'\mathscr{B}s'$。

则称$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_2$的V-互模拟关系。

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，那么$\mathscr{B}$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[103]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$\mathscr{M}'\models\varphi$[101,103]。因此，若$\mathrm{IR}(\varphi,V)$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 3.4. 令$\varphi=\nu X.(ch\wedge j\wedge\mathrm{EXEX}X)$，易证下面两个赋值$(\mathscr{M},v)$和$(\mathscr{M}',v')$（图 3.2中的两个Kripke结构）是$\varphi$的模型，其中

- $\mathscr{M}=(S,r,R,L)$，其中$S=\{s_0,s_1,s_2\}$，$r=s_0$，$L(s_0)=\{ch,j\}$，$L(s_1)=\{ch\}$，$L(s_2)=\emptyset$，$R=\{(s_0,s_1),(s_1,s_0),(s_0,s_2),(s_2,s_0)\}$，$v(X)=\{s_0\}$，
- $\mathscr{M}'=(S',r',R',L')$，其中$S'=\{t_0,t_1\}$,$r'=t_0$，$L'(t_0)=\{j\}$，$L'(t_1)=\emptyset$，$R'=\{(t_0,t_1),(t_1,t_0)\}$且$v'(X)=\{t_0\}$。

由于$\mathscr{M}$和$\mathscr{M}'$之间存在一个$\{ch\}$-互模拟关系$\mathscr{B}=\{(s_0,t_0),(s_1,t_1),(s_2,t_1)\}$，所以，$\mathscr{M}\leftrightarrow_{\{ch\}}\mathscr{M}'$。

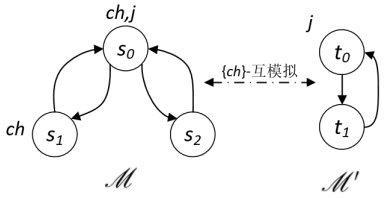


图 3.2: 两个$\{ch\}$-互模拟的Kripke结构示意图

μ-句子互模拟是不变的，μ-公式则不是[116]。这里定义一种对μ-公式是不变的互模拟。

定义 3.5 (变元-命题-互模拟)**.** 给定$V \subseteq \mathscr{A}$、$\mathscr{V}_1 \subseteq \mathscr{V}$、$\mathscr{M}_i = (S_i, r_i, R_i, L_i)$为*Kripke*结构、$s_i \in S_i$且$v_i : \mathscr{V} \to 2^{S_i}$，其中$i \in \{1,2\}$。若关系$\mathscr{B} \subseteq S_1 \times S_2$满足：

- $(s_1, s_2) \in \mathscr{B}$，
- $\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_2$之间的V-互模拟，且
- 对任意$(t_1, t_2) \in \mathscr{B}$和$X \in \mathscr{V} - \mathscr{V}_1$，$t_2 \in v_2(X)$当且仅当$t_1 \in v_1(X)$。

则称$\mathscr{B}$是$(\mathscr{M}_1, s_1, v_1)$和$(\mathscr{M}_2, s_2, v_2)$之间的一个$\langle \mathscr{V}_1, V\rangle$-互模拟。

若$(\mathscr{M}, s, v)$和$(\mathscr{M}', s', v')$之间存在一个$\langle \mathscr{V}_1, V\rangle$-互模拟关系$\mathscr{B}$，则称$(\mathscr{M}, s, v)$和$(\mathscr{M}', s', v')$是$\langle \mathscr{V}_1, V\rangle$-互模拟的，记为$(\mathscr{M}, s, v) \leftrightarrow_{\langle \mathscr{V}_1, V\rangle} (\mathscr{M}', s', v')$。若$(\mathscr{M}, s, v)$和$(\mathscr{M}', s', v')$之间存在一个$\langle \emptyset, V_1\rangle$-互模拟关系，则称$(\mathscr{M}, s, v)$和$(\mathscr{M}', s', v')$是*Var*-$V_1$-互模拟的，记为$(\mathscr{M}, s, v) \leftrightarrow_{V_1} (\mathscr{M}', s', v')$；称$\langle \emptyset, V_1\rangle$-互模拟为Var-$V_1$-互模拟。此外，若$s = r$且$s' = r'$，则$(\mathscr{M}, s, v) \leftrightarrow_{\langle \mathscr{V}_1, V\rangle} (\mathscr{M}', s', v')$简写为$(\mathscr{M}, v) \leftrightarrow_{\langle \mathscr{V}_1, V\rangle} (\mathscr{M}', v')$。

显然，若$(\mathscr{M}, s, v) \leftrightarrow_V (\mathscr{M}', s', v')$，则$\mathscr{M} \leftrightarrow_V \mathscr{M}'$。$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，但$(\mathscr{M}, v) \not\leftrightarrow_V (\mathscr{M}', v')$有时是成立的。例：例 3.4中的两个Kripke结构$\mathscr{M}$和$\mathscr{M}'$是$\emptyset$-互模拟的。然而，由于$r_1 \in v(X)$和$(r_1, r_2') \in \mathscr{B}$，但是$r_2' \notin v'(X)$，所以$(\mathscr{M}, v) \not\leftrightarrow_\emptyset (\mathscr{M}', v')$。对$S_1 \subseteq S$和二元关系$\mathscr{B} \subseteq S \times S'$，记$\mathscr{B}(S_1) = \{s' \mid (s, s') \in \mathscr{B},\ s \in S_1\}$。

再者，对任意$X \in \mathscr{V}$，令$v(X) = S$且$v'(X) = S'$，Kripke结构$\mathscr{M}$和$\mathscr{M}'$之间的任意V-互模拟可以容易地扩展为$(\mathscr{M}, r, v)$和$(\mathscr{M}', r', v')$之间的Var-V-互模拟，其中S和S'分别为$\mathscr{M}$和$\mathscr{M}'$中的状态集合。

例 3.5 (例 3.4的延续)**.** 令$\mathscr{M}$和$\mathscr{M}'$为图 3.2中的Kripke结构，$v : \mathscr{V} \to 2^S$和$v' : \mathscr{V} \to 2^{S'}$为将$\mathscr{V}$中的变元分别赋值到$\mathscr{M}$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$X \in \mathscr{V}$，$v(X) = \{s_0, s_1, s_2\}$且$v'(X) = \{t_0, t_1\}$，则$(\mathscr{M}, v) \leftrightarrow_{\{ch\}} (\mathscr{M}', v')$；
- 若对任意$X \in \mathscr{V} - \{X_1\}$，$v(X_1) = \{s_0\}$、$v'(X_1) = \{t_1\}$、$v(X) = \{s_0, s_1, s_2\}$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$t_0 \notin v'(X_1)$。

可以证明$\leftrightarrow_V$有如下性质。

命题 3.6. 令$\mathscr{V}_1,\mathscr{V}_2\subseteq\mathscr{V}$、$V,V_1\subseteq\mathscr{A}$且$\mathscr{M}_i$为*Kripke*结构（$i=1,2,3$），若$v_i:\mathscr{V}\to 2^{S_i}$，则：

(i) $\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}$为赋值间的等价关系；

(ii) 若$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_2,s_2,v_2)$、$\mathscr{V}_1\subseteq\mathscr{V}_2$且$V\subseteq V_1$，
则$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_2,V_1\rangle}(\mathscr{M}_2,s_2,v_2)$；

(iii) 若$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_2,s_2,v_2)$且$(\mathscr{M}_2,s_2,v_2)\leftrightarrow_{\langle\mathscr{V}_2,V_1\rangle}(\mathscr{M}_3,s_3,v_3)$，
则$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1\cup\mathscr{V}_2,V\cup V_1\rangle}(\mathscr{M}_3,s_3,v_3)$。

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。

(a) $\leftrightarrow_V$是自反的。显然，对任意的赋值$(\mathscr{M},s,v)$都有$(\mathscr{M},s,v)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M},s,v)$。

(b) 对称性：下证对任意$(\mathscr{M}_1,s_1,v_1)$和$(\mathscr{M}_2,s_2,v_2)$，若$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_2,s_2,v_2)$，则$(\mathscr{M}_2,s_2,v_2)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_1,s_1,v_1)$。假定$(\mathscr{M}_1,s_1,v_1)$和$(\mathscr{M}_2,s_2,v_2)$有$\langle\mathscr{V}_1,V\rangle$-互模拟关系$\mathscr{B}$，如下构造关系$\mathscr{B}_1$：$\mathscr{B}_1=\{(s,t)\mid(t,s)\in\mathscr{B}$，其中$t\in S_1$和$s\in S_2\}$。下证$\mathscr{B}_1$是$\mathscr{M}_2\mathscr{M}_1$之间的$V$-互模拟关系。

- 因为$(r_1,r_2)\in\mathscr{B}$，所以，$(r_2,r_1)\in\mathscr{B}_1$；
- 对任意$s\in S_2$和$t\in S_1$，若$(s,t)\in\mathscr{B}_1$，则$(t,s)\in\mathscr{B}$。因此，对任意$p\in\mathscr{A}-V$，$p\in L_1(t)$当且仅当$p\in L_2(s)$；
- 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$(t,s)\in\mathscr{B}$，则对任意$X\in\mathscr{V}-\mathscr{V}_1$，$s\in v_1(X)$当且仅当$t\in v_2(X)$。因此，若$(s,t)\in\mathscr{B}_1$，则对任意$X\in\mathscr{V}-\mathscr{V}_1$，$t\in v_2(X)$当且仅当$s\in v_1(X)$。所以，$(\mathscr{M}_2,s_2,v_2)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_1,s_1,v_1)$。

(c) 传递性：下证若$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_2,s_2,v_2)$和$(\mathscr{M}_2,s_2,v_2)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_3,s_3,v_3)$，则$(\mathscr{M}_1,s_1,v_1)\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}(\mathscr{M}_3,s_3,v_3)$。$(\mathscr{M}_1,s_1,v_1)$和$(\mathscr{M}_2,s_2,v_2)$有$\langle\mathscr{V}_1,V\rangle$-互模拟关系$\mathscr{B}_1$，$(\mathscr{M}_2,s_2,v_2)$ $(\mathscr{M}_3,s_3,v_3)$有$\langle\mathscr{V}_1,V\rangle$-互模拟关系$\mathscr{B}_2$，如下构造$\mathscr{B}$：$\mathscr{B}=\{(s,z)\mid(s,t)\in\mathscr{B}_1$ 和 $(t,z)\in\mathscr{B}_2$，其中$s\in S_1,t\in S_2,$和$z\in S_3\}$。可以类似(b)证明$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_3$之间的V-互模拟关系。因此，$\mathscr{M}_1\leftrightarrow_{\langle\mathscr{V}_1,V\rangle}\mathscr{M}_3$。

此外，若$(s,t)\in\mathscr{B}$，则存在$t\in S_2$使得$(s,t)\in\mathscr{B}_1$和$(t,z)\in\mathscr{B}_2$。因此，对任意$X\in\mathscr{V}-\mathscr{V}_1$，$s\in v_1(X)$当且仅当$t\in v_2(X)$当且仅当$z\in V_3(X)$。所以，$(\mathscr{M}_2,v_2)\leftrightarrow_V(\mathscr{M}_1,v_1)$。

(ii) 假定$\mathscr{B}_V$是$(\mathscr{M}_1,s_1,v_1)$和$(\mathscr{M}_2,s_2,v_2)$之间的$\langle\mathscr{V}_1,V\rangle$-互模拟关系。下证$\mathscr{B}_V$也是$(\mathscr{M}_1,s_1,v_1)$和$(\mathscr{M}_2,s_2,v_2)$之间的$\langle\mathscr{V}_2,V_1\rangle$-互模拟关系。显然：

- $(r_1,r_2)\in\mathscr{B}_V$；

- 对任意$w_1 \in S_1$和$w_2 \in S_2$，若$(w_1,w_2) \in \mathscr{B}_V$，则对任意$p \in \mathscr{A} - V_1$，$p \in L_1(w_1)$当且仅$p \in L_2(w_2)$（因为$V \subseteq V_1$且对任意$p \in \mathscr{A} - V$，$p \in L_1(w_1)$当且仅当$p \in L_2(w_2)$）；
- 因为$\mathscr{B}_V$ 是$\mathscr{M}_1$和$\mathscr{M}_2$之间的V-互模拟，所以V-互模拟的第三和第四个点很容易能够证明。

因此，$\mathscr{B}_V$ 是$\mathscr{M}_1$ 和$\mathscr{M}_2$之间的V_1-互模拟。

此外，若$(s,t) \in \mathscr{B}_V$，则对任意$X \in \mathscr{V} - \mathscr{V}_2$，$s \in v_1(X)$ 当且仅当$t \in v_2(X)$。因此，$(\mathscr{M}_1,s_1,v_1) \leftrightarrow_{\langle \mathscr{V}_2,V_1\rangle} (\mathscr{M}_2,s_2,v_2)$。

(iii) 由(ii)可知，若$(\mathscr{M}_1,s_1,v_1) \leftrightarrow_{\langle \mathscr{V}_1,V\rangle} (\mathscr{M}_2,s_2,v_2)$，则$(\mathscr{M}_1,s_1,v_1) \leftrightarrow_{\langle \mathscr{V}_1\cup\mathscr{V}_2,V\cup V_1\rangle} (\mathscr{M}_2,s_2,v_2)$。因此，由(i)可知$(\mathscr{M}_1,s_1,v_1) \leftrightarrow_{\langle \mathscr{V}_1\cup\mathscr{V}_2,V\cup V_1\rangle} (\mathscr{M}_3,s_3,v_3)$。 □

不难看出，命题 3.6对于Kripke结构之间的$\leftrightarrow_V$关系也是成立的。直观地说，(i)表示$\leftrightarrow_{\langle \mathscr{V}_1,V\rangle}$是赋值之间的自反、对称和传递关系。(ii)指明若两个赋值是$\langle \mathscr{V}_1,V\rangle$-互模拟的，则对于任意$V$的超集$V_1$和$\mathscr{V}_1$的超集$\mathscr{V}_2$，这两个赋值是$\langle \mathscr{V}_2,V_1\rangle$-互模拟的。(iii)表示如果一个Kripke结构和其它两个Kripke结构$\langle \mathscr{V}_1,V\rangle$和$\langle \mathscr{V}_2,V_1\rangle$互模拟，则这两个Kripke结构$\langle \mathscr{V}_1\cup\mathscr{V}_2,V\cup V_1\rangle$-互模拟。

引理 3.3. 令$\mathscr{M} = (S,r,R,L)$、φ为μ-公式且$S_1 \subseteq S_2 \subseteq S$。若$X$正出现在$\varphi$中，则$\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} \subseteq \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}$.

证明. 通过归纳φ的结构来证明。

基始. (a) $\varphi = p$，其中$p \in \mathscr{A}$。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \{p \mid p \in L(s)\}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(b) $\varphi = Y$，其中$Y \in \mathscr{V} - \{X\}$。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= v[X := S_1](Y)\\ &= v[X := S_2](Y)\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(c) $\varphi = X$。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= v[X:=S_1](X)\\ &= S_1\\ &\subseteq S_2\\ &= v[X=S_2](X)\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

归纳步. (a) $\varphi=\varphi_1*\varphi_2$，其中$*\in\{\vee,\wedge\}$且$X$正出现在$\varphi_i$中（$i=1,2$）。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]} * \|\varphi_2\|^{\mathscr{M}}_{v[X:=S_1]}\\ &\subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]} * \|\varphi_2\|^{\mathscr{M}}_{v[X:=S_2]} && \text{(归纳假设)}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(b) $\varphi=\mathrm{AX}\varphi_1$，且$X$正出现在$\varphi_1$中。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \{s \mid \forall s'.(s,s')\in R \Rightarrow s'\in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]}\}\\ &\subseteq \{s \mid \forall s'.(s,s')\in R \Rightarrow s'\in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]}\} && \text{(归纳假设)}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(c) $\varphi=\mathrm{EX}\varphi_1$，且$X$正出现在$\varphi_1$中。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \{s \mid \exists s'.(s,s')\in R \wedge s'\in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]}\}\\ &\subseteq \{s \mid \exists s'.(s,s')\in R \wedge s'\in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]}\} && \text{(归纳假设)}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(d) $\varphi=\nu Y.\varphi_1$，且X正出现在φ_1中。

(d.1) $Y=X$.

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \bigcup\{S'\subseteq S \mid S'\subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1][X:=S']}\}\\ &= \bigcup\{S'\subseteq S \mid S'\subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S']}\}\end{aligned}$$

$$= \bigcup\{S' \subseteq S \mid S' \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2][X:=S']}\}$$
$$= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.$$

(d.2) $Y \neq X$.

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \bigcup\{S' \subseteq S \mid S' \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1][Y:=S']}\}\\ &\subseteq \bigcup\{S' \subseteq S \mid S' \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2][Y:=S']}\} && \text{(归纳假设)}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(e) $\varphi = \mu X.\varphi_1$，且$X$正出现在$\varphi_1$中。

(e.1) $Y = X$.

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \bigcap\{S' \subseteq S \mid \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1][X:=S']} \subseteq S'\}\\ &= \bigcap\{S' \subseteq S \mid \|\varphi_1\|^{\mathscr{M}}_{v[X:=S']} \subseteq S'\}\\ &= \bigcap\{S' \subseteq S \mid \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2][X:=S']} \subseteq S'\}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

(e.2) $Y \neq X$. 注意，对任意μ-公式ψ，$\|\psi\|^{\mathscr{M}}_{v[X:=S]} \subseteq S$。

$$\begin{aligned}\|\varphi\|^{\mathscr{M}}_{v[X:=S_1]} &= \bigcap\{S' \subseteq S \mid \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1][Y:=S']} \subseteq S'\}\\ &\subseteq \bigcap\{S' \subseteq S \mid \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2][Y:=S']} \subseteq S'\} && \text{(归纳假设)}\\ &= \|\varphi\|^{\mathscr{M}}_{v[X:=S_2]}.\end{aligned}$$

□

命题 3.7 (不变性)**.** 令φ为μ-公式、$\mathscr{V}_1 \subseteq \mathscr{V}$且$V \subseteq \mathscr{A}$。若$(\mathscr{M},s,v) \leftrightarrow_{\langle \mathscr{V}_1,V\rangle} (\mathscr{M}',s',v')$且$IR(\varphi, V \cup \mathscr{V}_1)$，则$(\mathscr{M},s,v) \models \varphi$当且仅当$(\mathscr{M}',s',v') \models \varphi$。

证明. 令$\mathscr{M} = (S,r,R,L)$且$\mathscr{M}' = (S',r',R',L')$。假设$Var(\varphi) \cap (V \cup \mathscr{V}_1) = \emptyset$且$(\mathscr{M},s,v)$和$(\mathscr{M}', s',v')$之间的$\langle \mathscr{V}_1,V\rangle$-互模拟关系为$\mathscr{B}$。

基始. (a) $\varphi = p$，其中$p \notin V$。

$(\mathscr{M},s,v) \models \varphi$
$\Leftrightarrow p \in L(s)$
$\Leftrightarrow p \in L(s')$ $\quad ((\mathscr{M},s,v) \leftrightarrow_{\langle \mathscr{V}_1,V\rangle} (\mathscr{M}',s',v'),\ p \notin V)$
$\Leftrightarrow (\mathscr{M}',s',v') \models \varphi$。

(b) $\varphi = X$，其中$X \notin \mathscr{V}_1$。

$(\mathscr{M}, s, v) \models \varphi$
$\Leftrightarrow s \in v(X)$
$\Leftrightarrow s' \in v'(X)$　　$((s,s') \in \mathscr{B}, X \notin \mathscr{V}_1)$
$\Leftrightarrow (\mathscr{M}', s', v') \models \varphi$。

归纳步. (a) $\varphi = \neg\varphi_1$。

$(\mathscr{M}, s, v) \models \varphi$
$\Leftrightarrow (\mathscr{M}, s, v) \not\models \varphi_1$
$\Leftrightarrow (\mathscr{M}', s', v') \not\models \varphi_1$　　(归纳假设)
$\Leftrightarrow (\mathscr{M}', s', v') \models \varphi$。

(b) $\varphi = \varphi_1 \vee \varphi_2$。

$(\mathscr{M}, s, v) \models \varphi$
$\Leftrightarrow (\mathscr{M}, s, v) \models \varphi_1$或$(\mathscr{M}, s, v) \models \varphi_2$
$\Leftrightarrow (\mathscr{M}', s', v') \models \varphi_1$或$(\mathscr{M}', s', v') \models \varphi_2$　　(归纳假设)
$\Leftrightarrow (\mathscr{M}', s', v') \models \varphi$。

(c) $\varphi = \mathrm{AX}\varphi_1$。

$(\mathscr{M}, s, v) \models \varphi$
$\Rightarrow$ 对任意$(s, s_1) \in R$，有$(\mathscr{M}, s_1, v) \models \varphi_1$
$\Rightarrow$ 对任意$(s', s_1') \in R'$，存在$(s, s_1) \in R$，使得$(s_1, s_1') \in \mathscr{B}$且$(\mathscr{M}, s_1, v) \models \varphi_1$　　$((\mathscr{M}, s, v) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}', s', v'))$
$\Rightarrow$ 对任意$(s', s_1') \in R'$，有$(\mathscr{M}', s_1', v') \models \varphi_1$　　(归纳假设)
$\Rightarrow (\mathscr{M}', s', v') \models \varphi$。

另一个方向可以类似地证明。

(d) $\varphi = \nu X.\varphi_1$。

$(\mathscr{M}, s, v) \models \varphi$
$\Rightarrow s \in \bigcup\{S_1 \subseteq S \mid S_1 \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]}\}$
$\Rightarrow$ 存在$S_1 \subseteq S$，使得$S_1 \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]}$和$s \in S_1$
$\Rightarrow$ 对任意$w \in S_2 = S_1 \cup \{s_2 \mid (t, s_i) \in \mathscr{B},\ s_2 \in S$和$s_1 \in S_1$，其中$i = 1,2\}$，存在$w' \in S_1$使得$(\mathscr{M}, w, v) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}, w', v)$　　(命题 3.6)
$\Rightarrow s \in S_1$，且对任意$w \in S_2 = S_1 \cup \{s_2 \mid (t, s_i) \in \mathscr{B},\ s_2 \in S$和$s_1 \in S_1$，其中$i = 1,2\}$，存在$w' \in S_1$使得$(\mathscr{M}, w, v[X := S_2]) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}, w', v[X := S_2])$
$\Rightarrow s \in S_2$，且对任意$w \in S_2$，有$w \in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]}$　　($S_1 \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_1]}$, 归纳假设, 引理 3.3)
$\Rightarrow s \in S_2$，且存在$S_2 \subseteq S$，使得$S_2 \subseteq \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]}$
$\Rightarrow s \in S_2$，且对任意$u' \in \mathscr{B}(S_2)$，存在$u \in S_2$使得$(\mathscr{M}', u', v'[X := \mathscr{B}(S_2)]) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}, u,$

$v[X:=S_2])$ 且$u \in \|\varphi_1\|^{\mathscr{M}}_{v[X:=S_2]}$
$\Rightarrow s \in S_2$，且对任意$u' \in \mathscr{B}(S_2)$，有$u' \in \|\varphi_1\|^{\mathscr{M}'}_{v'[X:=\mathscr{B}(S_2)]}$ (归纳假设)
$\Rightarrow$ 存在$\mathscr{B}(S_2) \subseteq S'$，使得$\mathscr{B}(S_2) \subseteq \|\varphi_1\|^{\mathscr{M}'}_{v'[X:=\mathscr{B}(S_2)]}$ 且$s' \in \mathscr{B}(S_2)$ $((s,s') \in \mathscr{B})$
$\Rightarrow s' \in \bigcup\{S'_1 \subseteq S' \mid S'_1 \subseteq \|\varphi_1\|^{\mathscr{M}'}_{v'[X:=S'_1]}\}$
$\Rightarrow (\mathscr{M}',s',v') \models \varphi$。

另一个方向可类似地证明。 □

命题 3.7表明，若两个赋值是$\langle \mathscr{V}_1, V\rangle$-互模拟的，则这两个赋值同时满足（或不满足）同一个与$V \cup \mathscr{V}_1$无关的公式。例如，$(\mathscr{M}_i, v_i)$ $i = 1,2$为两个赋值，$\varphi = X$，$V = \{p\}$；若$(\mathscr{M}_1, v_1) \leftrightarrow_V (\mathscr{M}_2, v_2)$，则可以证明$(\mathscr{M}_1, v_1) \models \varphi$ 当且仅当$(\mathscr{M}_2, v_2) \models \varphi$。

此时，μ-演算遗忘如下定义：

定义 3.6 (μ-演算遗忘)**.** 令$V \subseteq \mathscr{A}$和φ为μ-公式。若$Var(\psi) \cap V = \emptyset$且下面等式成立，则称$\psi$是从$\varphi$中遗忘$V$后得到的结果：

$$Mod(\psi) = \{(\mathscr{M}, v) \mid \exists (\mathscr{M}', v') \in Mod(\varphi) \text{ 且} (\mathscr{M}', v') \leftrightarrow_V (\mathscr{M}, v)\}。$$

显然，当讨论的公式为μ-句子的时候，定义 3.6中的模型变成$\mathscr{M}$和$\mathscr{M}'$，即去掉赋值函数v和v'。

定义 3.6表明如果ψ和ψ'都是从φ中遗忘V中的原子命题得到的结果，则$Mod(\psi) = Mod(\psi')$，即：遗忘的结果之间是语义等价的（即有相同的模型）。此时，将遗忘的结果记为$\mathrm{F}_\mu(\varphi, V)$，除非另有说明，否则$\mathrm{F}_\mu(\varphi, V)$表示从$\varphi$中遗忘$V$的结果是$\mu$-公式。

3.2.2 遗忘算子及其性质

这部分展示μ-演算遗忘的语义性质。特别地，这里将证明上述μ-演算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-句子中遗忘任意原子命题集的结果总是一个μ-句子。此外，研究了遗忘算子的代数性质，包括分解性（decomposition）、切片性（slice）和同质性（homogeneity）。

定理 3.4. 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$Var(\psi) \cap \{q\} = \emptyset$且$\psi \equiv \mathrm{F}_\mu(\varphi, \{q\})$。

证明. 已有结果表明，对任意μ-句子φ和原子命题p，存在一个$\{p\}$-无关的μ-句子φ'（即$\mathrm{IR}(\varphi', \{p\})$）使得[103]：

$$\mathscr{M} \models \varphi' \text{ 当且仅当} \exists \mathscr{M}' \in Mod(\varphi) \text{ 使得} \mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}'.$$

这与本文遗忘的定义一致，因此上述结论成立。 □

令$V \subseteq \mathscr{A}$、φ为μ-公式、φ'是从φ中遗忘V后得到的结果。则μ-演算下遗忘的基本公设如下：

(W) 削弱：$\varphi \models \varphi'$;
(PP) 正支持：对任意μ-句子η，若$\mathrm{IR}(\eta,V)$和$\varphi \models \eta$则$\varphi' \models \eta$；
(NP) 负支持：对任意μ-句子η，若$\mathrm{IR}(\eta,V)$和$\varphi \not\models \eta$则$\varphi' \not\models \eta$；
(IR) 无关性：$\mathrm{IR}(\varphi',V)$。

定理 3.5 (表示性定理)**.** 给定μ-公式φ、φ'和ϕ，$V \subseteq \mathscr{A}$为原子命题集。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集，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$(\mathscr{M}',v')$
$\Rightarrow \exists(\mathscr{M},v)$使得$(\mathscr{M},v) \models \varphi$和$(\mathscr{M},v) \leftrightarrow_V (\mathscr{M}',v')$ (定义3.6)
$\Rightarrow$ 对于任意与V-无关且$\varphi \models \phi$的ϕ都有$(\mathscr{M}',v') \models \phi$
$\Rightarrow (\mathscr{M}',v') \models \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$

$(\Leftarrow)$ 由于$\mathrm{IR}(\mathrm{F}_\mu(\varphi,V),V)$和$\varphi \models \mathrm{F}_\mu(\varphi,V)$，由定义3.6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$A$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$\phi' \in A$，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3.6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3.5 表明基本公设对μ-演算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。

D'Agostino 研究了μ-演算均匀插值，并指出μ-算具有均匀插值性质[83,103-104]。换句话说，这意味着对任意μ-句子φ和有限的原子命题集$V \subseteq Var(\varphi)$，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，对于$\mu$-句子$\varphi$，上述定义的遗忘$\mathrm{F}_\mu(\varphi,V)$与$\widetilde{\exists}V\varphi$[83] 语义等价。

推论 3.4. 令$V\subseteq\mathscr{A}$且φ为μ-句子，则$\widetilde{\exists}V\varphi\equiv\mathrm{F}_\mu(\varphi,V)$。

证明. 显然$\varphi\models\mathrm{F}_\mu(\varphi,V)$且$\mathrm{F}_\mu(\varphi,V)\cap V=\emptyset$。此外，对任意$\phi$，若$\varphi\models\phi$且$\mathrm{IR}(\phi,V)$，由定理 3.5可知$\mathrm{F}_\mu(\varphi,V)\models\phi$。

因此，由均匀插值的定义可知$\mathrm{F}_\mu(\varphi,V)$是$\varphi$ 关于$Var(\varphi)-V$的均匀插值。 □

以下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 3.4. 令φ和α为两个μ-公式，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$(\mathscr{M},v)$，存在一个赋值$(\mathscr{M}',v')$ 使得$(\mathscr{M},v)\leftrightarrow_{\{q\}}(\mathscr{M}',v')$且$(\mathscr{M}',v')\models\varphi'$。显然，$(\mathscr{M}',v')\models\varphi$，又因为$\mathrm{IR}(\varphi,\{q\})$且$(\mathscr{M},v)\leftrightarrow_{\{q\}}(\mathscr{M}',v')$，所以$(\mathscr{M},v)\models\varphi$。

令$(\mathscr{M},v)\in Mod(\varphi)$，其中$\mathscr{M}=(S,s,R,L)$。如下构造$(\mathscr{M}',v')$，其中$\mathscr{M}'=(S,s,R,L')$且：

$$
\begin{aligned}
&L':S\to 2^{\mathscr{A}}\text{且}\forall s^*\in S,\text{若}(\mathscr{M},s^*)\not\models\alpha,\text{则}L'(s^*)=L(s^*)-\{q\},\\
&\text{否则}L'(s^*)=L(s^*)\cup\{q\},\\
&L'(s)=L(s)\cup\{q\}\text{若}(\mathscr{M},s)\models\alpha,\text{否则}L'(s)=L(s)-\{q\}。
\end{aligned}
$$

显然，$\mathscr{M}'\leftrightarrow_{\{q\}}\mathscr{M}$，所以$\mathscr{M}$和$\mathscr{M}'$之间存在一个$\{q\}$-互模拟关系$\mathscr{B}$。再者，对任意$X\in\mathscr{V}$，令$v'(X)=\mathscr{B}(v(X))$。

所以，$(\mathscr{M}',v')\models\varphi$、$(\mathscr{M}',v')\models q\leftrightarrow\alpha$且$(\mathscr{M},v)\leftrightarrow_{\{q\}}(\mathscr{M}',v')$。因此，$(\mathscr{M}',v')\models\varphi\wedge(q\leftrightarrow\alpha)$。又因为$(\mathscr{M},v)\leftrightarrow_{\{q\}}(\mathscr{M}',v')$且$\mathrm{IR}(\mathrm{F}_\mu(\varphi\wedge(q\leftrightarrow\alpha),q),\{q\})$，从而$(\mathscr{M},v)\models\mathrm{F}_\mu(\varphi\wedge(q\leftrightarrow\alpha),\{q\})$。 □

正如在第一章中所说，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遗忘与μ-演算遗忘之间的联系。对于给定的Kripke结构$\mathscr{M}=(S,R,L,r)$和命题公式ψ，$\mathscr{M}\models\psi$当且仅当$L(r)\models\psi$。经典命题逻辑与μ-演算下的遗忘之间的联系如下：

定理 3.6. 令φ为命题公式，$V\subseteq\mathscr{A}$为原子命题集，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$\mathscr{M}\in Mod(\mathrm{F}_\mu(\varphi,V))$

$\Rightarrow$ 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$\mathscr{B}$

$\Rightarrow r\mathscr{B}r'$

$\Rightarrow \mathscr{M}\models Forget(\varphi,V)$　　(IR($Forget(\varphi,V),V$), V-无关性)

($\Leftarrow$) 对任意$\mathscr{M}\in Mod(Forget(\varphi,V))$

$\Rightarrow \exists\mathscr{M}'\in Mod(\varphi)$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($\mathrm{F}_\mu(\varphi,V),V$), V-无关性）　　□

定理3.6表明μ-演算遗忘是命题逻辑遗忘的扩展。下面的性质在命题逻辑、S5[48]和CTL中都成立，这里证明其在μ-演算中也成立。

命题 3.8. *给定μ-公式φ、φ_i和ψ_i（$i=1,2$），$V\subseteq\mathscr{A}$为原子命题集。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) $\mathrm{F}_\mu(\psi_1\wedge\psi_2,V)\models\mathrm{F}_\mu(\psi_1,V)\wedge\mathrm{F}_\mu(\psi_2,V)$；
(vi) $\mathrm{F}_\mu(\varphi_1\wedge\varphi_2,V)\equiv\mathrm{F}_\mu(\varphi_1,V)\wedge\varphi_2$ *若*$IR(\varphi_2,V)$。

证明. (i) $(\mathscr{M},v)\in Mod(\mathrm{F}_\mu(\varphi,V))$

$\Leftrightarrow\exists(\mathscr{M}',v')\in Mod(\varphi)$ 使得$(\mathscr{M},v)\leftrightarrow_V(\mathscr{M}',v')$。

因为$\varphi\models\mathrm{F}_\mu(\varphi,V)$，所以$(\mathscr{M},v)\in Mod(\varphi)$ 蕴涵$(\mathscr{M},v)\models\mathrm{F}_\mu(\varphi,V)$。

(ii) 和(iii) 可以类似地证明。

(iv) $(\mathscr{M},v)\in Mod(\mathrm{F}_\mu(\varphi_1\vee\varphi_2,V))$

$\Leftrightarrow$ 存在$(\mathscr{M}',v')\in Mod(\varphi_1\vee\varphi_2)$使得$(\mathscr{M},v)\leftrightarrow_V(\mathscr{M}',v')$ 和$(\mathscr{M}',v')\models\varphi_1$ 或$(\mathscr{M}',v')\models\varphi_2$

$\Leftrightarrow$ 存在$(\mathscr{M},v) \leftrightarrow_V (\mathscr{M}',v')$ 使得$(\mathscr{M}',v') \models \varphi_1$ 或$(\mathscr{M}',v') \models \varphi_2$，则存在$(\mathscr{M}_1,v_1) \in Mod(\mathrm{F}_\mu(\varphi_1,V))$ 使得$(\mathscr{M}',v') \leftrightarrow_V (\mathscr{M}_1,v_1)$，或$\exists(\mathscr{M}_2,v_2) \in Mod(\mathrm{F}_\mu(\varphi_2,V))$ 使得$(\mathscr{M}',v') \leftrightarrow_V (\mathscr{M}_2,v_2)$

$\Leftrightarrow (\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_1,V)$或$(\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_2,V)$

$\Leftrightarrow (\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_1,V) \vee \mathrm{F}_\mu(\varphi_2,V)$。

(v)可以类似于(iv)一样证明。

(vi) $(\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_1 \wedge \varphi_2,V)$

$\Leftrightarrow$ 存在$(\mathscr{M},v) \leftrightarrow_V (\mathscr{M}',v')$ 使得$(\mathscr{M}',v') \models \varphi_1 \wedge \varphi_2$

$\Leftrightarrow$ 存在$(\mathscr{M},v) \leftrightarrow_V (\mathscr{M}',v')$ 使得$(\mathscr{M}',v') \models \varphi_1$ 和$(\mathscr{M}',v') \models \varphi_2$

$\Leftrightarrow (\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_1,V)$ 且$(\mathscr{M},v) \models \varphi_2$ $(\mathrm{IR}(\varphi_2,V))$

$\Leftrightarrow (\mathscr{M},v) \models \mathrm{F}_\mu(\varphi_1,V) \wedge \varphi_2$。 □

命题3.8(i) 表明从一个μ-句子中遗忘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，则它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再析取而得到；而(v)指出，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不成立。例：令$\varphi = p \wedge (q \vee \neg p)$，从$\varphi$中遗忘$p$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。

命题 3.9 (分解性). *给定μ-句子φ、原子命题集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$(\mathscr{M}_1,v_1)$（$\mathscr{M}_1 = (S_1,R_1,L_1,s_1)$）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$(\mathscr{M},v)$（$\mathscr{M} = (S,R,L,s)$）使得$(\mathscr{M}_1,v_1) \leftrightarrow_{\{p\} \cup V} (\mathscr{M},v)$。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$q \in V$，$q \in L_2(s_2)$当且仅当$q \in L(s)$，
- 对于其它的原子命题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) 其它情形：假定$\mathscr{M}_1$和$\mathscr{M}$有$\{p\} \cup V$-互模拟关系$\mathscr{B}$。

(i) 对任意$w \in S$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$q \in V$，$q \in L_2(w_2)$当且仅当$q \in L(w)$，
* 对其它原子命题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$(w_2', w_2) \in R_2$。

(3) 删除S_2和R_2中重复的元素。

显然，$\mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}_2$和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$。此外，对所有的$X \in \mathscr{V}$，令$v_2(X) = \mathscr{B}_1(v(X))$。因此，$(\mathscr{M}, v) \leftrightarrow_{\{p\}} (\mathscr{M}_2, v_2)$ 且$(\mathscr{M}_2, v_2) \leftrightarrow_V (\mathscr{M}_1, v_1)$，从而$(\mathscr{M}_2, s_2) \models \mathrm{F}_\mu(\varphi, p)$。所以，$(\mathscr{M}_1, v_1) \models \mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, p), V)$。

另一方面，假设$(\mathscr{M}_1, v_1)$是$\mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, p), V)$的一个模型

$\Rightarrow$ 存在$(\mathscr{M}_2, v_2)$使得$(\mathscr{M}_2, v_2) \models \mathrm{F}_\mu(\varphi, p)$ 且$(\mathscr{M}_2, v_2) \leftrightarrow_V (\mathscr{M}_1, v_1)$　　(定义3.6)

$\Rightarrow$ 存在$(\mathscr{M}, v)$ 使得$(\mathscr{M}, v) \models \varphi$ 且$(\mathscr{M}, v) \leftrightarrow_{\{p\}} (\mathscr{M}_2, v_2)$　　(定义3.6)

因此，由命题3.6 (iii)可知$(\mathscr{M}, v) \leftrightarrow_{\{p\} \cup V} (\mathscr{M}_1, v_1)$，所以$(\mathscr{M}_1, v_1) \models \mathrm{F}_\mu(\varphi, \{p\} \cup V)$。

□

下面这一性质为上述命题的推论。

推论 3.5 (切片性)**.** *给定μ-句子φ和原子命题集$V_i \subseteq \mathscr{A}$ $(i = 1, 2)$，有：*

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

F_μ的另一个性质是关于时序算子AX（EX）和不动点操作μX（νX）的：形如AXφ（EXφ）和$\mu X.\varphi$（$\nu X.\varphi$）的μ-句子的遗忘可以提到AX（EX）和μX（νX）后面计算。

引理 3.5. *令$(\mathscr{M}, s, v) \leftrightarrow_{\langle\{X\}, V\rangle} (\mathscr{M}', s', v')$且$\varphi$是一个$\mu$-公式。若$IR(\varphi, V)$且$X$是约束变元，则$(\mathscr{M}, s, v) \models \varphi$ 当且仅当$(\mathscr{M}', s', v') \models \varphi$。*

证明. 假设$(\mathscr{M}, s, v)$和$(\mathscr{M}', s', v')$ 之间的$\langle\{X\}, V\rangle$-互模拟关系为$\mathscr{B}_X$。

基始. (a) $\varphi = p$，其中$p \in \mathscr{A}$。

$(\mathscr{M}, s, v) \models \varphi$

$\Leftrightarrow p \in L(s)$

$\Leftrightarrow p \in L'(s)$　　$((\mathscr{M}, s, v) \leftrightarrow_{\langle\{X\}, V\rangle} (\mathscr{M}', s', v'))$

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

(b) $\varphi = Y$，其中$Y \in \mathscr{V} - \{X\}$。

$(\mathscr{M}, s, v) \models \varphi$

$\Leftrightarrow s \in v(Y)$

$\Leftrightarrow s' \in v'(Y)$ $((\mathscr{M},s,v) \leftrightarrow_{\langle\{X\},V\rangle} (\mathscr{M}',s',v'))$

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

归纳步. (a) $\varphi = \neg\varphi_1$。

$(\mathscr{M},s,v) \models \varphi$

$\Leftrightarrow (\mathscr{M},s,v) \not\models \varphi_1$

$\Leftrightarrow (\mathscr{M}',s',v') \not\models \varphi_1$

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

(b) $\varphi = \varphi_1 \vee \varphi_2$容易证明。

(c) $\varphi = \mathrm{AX}\varphi_1$。

$(\mathscr{M},s,v) \models \varphi$

$\Rightarrow$ 对任意$(s,s_1) \in R$，$(\mathscr{M},s_1,v) \models \varphi_1$

$\Rightarrow$ 对任意$(s',s_1') \in R'$，存在$(s,s_1) \in R$使得$(\mathscr{M},s_1,v) \leftrightarrow_{\langle\{X\},V\rangle} (\mathscr{M}',s_1',v')$和$(\mathscr{M},s_1,v) \models \varphi_1$

$\Rightarrow$ 对任意$(s',s_1') \in R'$，$(\mathscr{M}',s_1',v') \models \varphi_1$ (归纳假设)

$\Rightarrow (\mathscr{M}',s',v') \models \varphi$。

另一个方向可以类似证明。

(d) $\varphi = \nu Y.\varphi_1$。

(d.1) $Y = X$。

$(\mathscr{M},s,v) \models \nu X.\varphi_1$

$\Rightarrow s \in \|\varphi\|_v^{\mathscr{M}}$

$\Rightarrow s \in \bigcup\{S_1 \subseteq S \mid S_1 \subseteq \|\varphi_1\|_{v[X:=S_1]}^{\mathscr{M}}\}$

$\Rightarrow$ 存在$S_1 \subseteq S$，使得$S_1 \subseteq \|\varphi_1\|_{v[X:=S_1]}^{\mathscr{M}}$和$s \in S_1$

$\Rightarrow$ 存在$S_2 = S_1 \cup \{s_2 \mid (s_i,t) \in \mathscr{B}_X,\ s_2 \in S$和$s_1 \in S_1$，其中$i = 1,2\}$，使得$S_2 \subseteq \|\varphi_1\|_{v[X:=S_2]}^{\mathscr{M}}$且$s \in S_2$ (看命题 3.7的证明)

$\Rightarrow$ 存在$\mathscr{B}_X(S_2) \subseteq S'$，使得$\mathscr{B}_X(S_2) \subseteq \|\varphi_1\|_{v'[X:=\mathscr{B}_X(S_2)]}^{\mathscr{M}'}$、$s' \in \mathscr{B}_X(S_2)$且$(\mathscr{M},v[X := S_2]) \leftrightarrow_V (\mathscr{M}',v'[X := \mathscr{B}_X(S_2)])$ $((s,s') \in \mathscr{B}_X$, 不变性)

$\Rightarrow s' \in \bigcup\{S_1' \subseteq S' \mid S_1' \subseteq \|\varphi_1\|_{v'[X:=S_1']}^{\mathscr{M}'}\}$

$\Rightarrow (\mathscr{M}',s',v') \models \nu X.\varphi_1$。

(d.2) $Y \neq X$。

$(\mathscr{M},s,v) \models \nu Y.\varphi_1$

$\Rightarrow s \in \|\varphi\|_v^{\mathscr{M}}$

$\Rightarrow s \in \bigcup\{S_1 \subseteq S \mid S_1 \subseteq \|\varphi_1\|_{v[Y:=S_1]}^{\mathscr{M}}\}$

$\Rightarrow$ 存在$S_1 \subseteq S$，使得$S_1 \subseteq \|\varphi_1\|_{v[Y:=S_1]}^{\mathscr{M}}$且$s \in S_1$

$\Rightarrow$ 存在$S_2 = S_1 \cup \{s_2 \mid (s_i,t) \in \mathscr{B}_X,\ s_2 \in S$和$s_1 \in S_1$，其中$i = 1,2\}$，使得$S_2 \subseteq \|\varphi_1\|_{v[Y:=S_2]}^{\mathscr{M}}$

且$s \in S_2$　　(看命题 3.7的证明)
$\Rightarrow$ 存在$\mathscr{B}_X(S_2) \subseteq S'$，使得$\mathscr{B}_X(S_2) \subseteq \|\varphi_1\|^{\mathscr{M}'}_{v'[Y:=\mathscr{B}_X(S_2)]}$、$s' \in \mathscr{B}_X(S_2)$且$(\mathscr{M}, v[Y := S_2]) \leftrightarrow_{\langle\{X\},V\rangle}$ $(\mathscr{M}', v'[Y := \mathscr{B}_X(S_2)])$　　($(s,s') \in \mathscr{B}_X$, 归纳假设)
$\Rightarrow s' \in \bigcup\{S'_1 \subseteq S' \mid S'_1 \subseteq \|\varphi_1\|^{\mathscr{M}'}_{v'[Y:=S'_1]}\}$
$\Rightarrow (\mathscr{M}', s', v') \models \nu X.\varphi_1$。

另一个方向可以类似证明。　□

命题 3.10 (同质性). 给定原子命题集$V \subseteq \mathscr{A}$和μ-公式φ，则:

(i) $\mathrm{F}_\mu(\mathrm{AX}\varphi, V) \equiv \mathrm{AXF}_\mu(\varphi, V)$;
(ii) $\mathrm{F}_\mu(\mathrm{EX}\varphi, V) \equiv \mathrm{EXF}_\mu(\varphi, V)$;
(iii) 如果$\nu X.\varphi$为μ-句子，$\mathrm{F}_\mu(\nu X.\varphi, V) \equiv \nu X.\mathrm{F}_\mu(\varphi, V)$;
(iv) 如果$\mu X.\varphi$为μ-句子，$\mathrm{F}_\mu(\mu X.\varphi, V) \equiv \mu X.\mathrm{F}_\mu(\varphi, V)$。

证明. 令$(\mathscr{M}, v)$（$\mathscr{M} = (S, R, L, r)$）、$(\mathscr{M}_i, v_i)$（$\mathscr{M}_i = (S_i, R_i, L_i, r_i)$）和$(\mathscr{M}', v')$（$\mathscr{M}' = (S', R', L', r')$）为赋值，其中$i \in \mathbb{N}$。

(i) ($\Rightarrow$) $(\mathscr{M}, v) \models \mathrm{F}_\mu(\mathrm{AX}\varphi, V)$
$\Rightarrow$ 存在$(\mathscr{M}, v) \leftrightarrow_V (\mathscr{M}', v')$ 且$(\mathscr{M}', v') \models \mathrm{AX}\varphi$
$\Rightarrow$ 存在$(\mathscr{M}, v) \leftrightarrow_V (\mathscr{M}', v')$ 且对任意$(r', r'') \in R'$，$(\mathscr{M}', r'', v') \models \varphi$
$\Rightarrow$ 对任意$(r, r_1) \in R$，存在$(r', r'_1) \in R'$ 使得$(\mathscr{M}, r_1, v) \leftrightarrow_V (\mathscr{M}', r'_1, v')$ 且对任意$(r', r'') \in R'$，$(\mathscr{M}', r'', v') \models \varphi$
$\Rightarrow$ 对任意$(r, r_1) \in R$，存在$(\mathscr{M}, r_1, v) \leftrightarrow_V (\mathscr{M}', r'_1, v')$ 且$(\mathscr{M}', r'_1, v') \models \varphi$
$\Rightarrow$ 对任意$(r, r_1) \in R$，$(\mathscr{M}, r_1, v) \models \mathrm{F}_\mu(\varphi, V)$　　(IR$(\mathrm{F}_\mu(\varphi, V), V)$)
$\Rightarrow (\mathscr{M}, v) \models \mathrm{AX}(\mathrm{F}_\mu(\varphi, V))$。

($\Leftarrow$) $(\mathscr{M}, v) \models \mathrm{AXF}_\mu(\varphi, V)$
$\Rightarrow$ 对任意$(r, r_1) \in R$，$(\mathscr{M}, r_1, v) \models \mathrm{F}_\mu(\varphi, V)$
$\Rightarrow$ 对任意$(r, r_1) \in R$，$(\mathscr{M}_{r_1}, r_1, v) \models \mathrm{F}_\mu(\varphi, V)$
$\Rightarrow$ 对任意$(r, r_1) \in R$，存在$(\mathscr{M}_{r_1}, r_1, v) \leftrightarrow_V (\mathscr{M}', r', v')$ 且$(\mathscr{M}', r', v') \models \varphi$
$\Rightarrow$ 对任意$i \geq 0$，存在$(\mathscr{M}_{r_i}, r_i, v) \leftrightarrow_V (\mathscr{M}'_i, r'_i, v'_i)$ 且$(\mathscr{M}'_i, r'_i, v'_i) \models \varphi$，其中$\{r_0, r_1, \dots\} = \{s \mid (r,s) \in R\}$，$\mathscr{M}'_i = (S'_i, R'_i, L'_i, r'_i)$（假定当$i \neq j$时，$S'_i \cap S'_j = \emptyset$），$\mathscr{B}_i$是$(\mathscr{M}_{r_i}, r_i, v)$ 和$(\mathscr{M}'_i, r'_i, v'_i)$之间的Var-$V$-互模拟关系
$\Rightarrow (\mathscr{M}^*, v^*) \leftrightarrow_V (\mathscr{M}, v)$ 且$(\mathscr{M}^*, v^*) \models \mathrm{AX}\varphi$，其中$\mathscr{M}^* = (S^*, R^*, L^*, r)$且

- $S^* = \{r\} \cup \bigcup_{i \geq 0} S'_i$,
- $R^* = \{(r, r'_i) \mid i \geq 0\} \cup \bigcup_{i \geq 0} R'_i$,
- 对任意$s \in S^*$，$L^*(s) = \bigcup_{i \geq 0} L'_i(s)$，且$L^*(r) = L(r)$,

- 对任意$X \in \mathscr{V}$，

$$v^*(X) = \begin{cases} \bigcup_{i\geq 0} v_i(X), & \text{若} r \notin v(X); \\ \bigcup_{i\geq 0} v_i(X) \cup \{r\}, & \text{否则。} \end{cases}$$

$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\mathrm{AX}\varphi, V)$。

(ii)可类似于(i)证明。

(iii) $(\Rightarrow)$ $(\mathscr{M}, v) \models \mathrm{F}_\mu(\nu X.\varphi, V)$

$\Rightarrow$ 存在$(\mathscr{M}', v') \leftrightarrow_V (\mathscr{M}, v)$ 且$(\mathscr{M}', v') \models \nu X.\varphi$

$\Rightarrow r' \in \bigcup\{S_1 \subseteq S' \mid S_1 \subseteq \|\varphi\|^{\mathscr{M}'}_{v'[X:=S_1]}\}$

$\Rightarrow$ 存在$S_1 \subseteq S'$ 使得$S_1 \subseteq \|\varphi\|^{\mathscr{M}'}_{v'[X:=S_1]}$、 $r' \in S_1$和$r' \in \|\varphi\|^{\mathscr{M}'}_{v'[X:=S_1]}$

$\Rightarrow r' \in \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}'}_{v'[X:=S_1]}$ 且$S_1 \subseteq \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}'}_{v'[X:=S_1]}$ $(\varphi \models \mathrm{F}_\mu(\varphi, V))$

$\Rightarrow r' \in \bigcup\{S_1 \subseteq S' \mid S_1 \subseteq \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}'}_{v'[X:=S_1]}\}$

$\Rightarrow (\mathscr{M}', v') \models \nu X.\mathrm{F}_\mu(\varphi, V)$

$\Rightarrow (\mathscr{M}, v) \models \nu X.\mathrm{F}_\mu(\varphi, V)$ $(\mathrm{IR}(\nu X.\mathrm{F}_\mu(\varphi, V), V))$

$(\Leftarrow)$ $(\mathscr{M}, v) \models \nu X.\mathrm{F}_\mu(\varphi, V)$

$\Rightarrow r \in \bigcup\{S_1 \subseteq S \mid S_1 \subseteq \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}}_{v[X:=S_1]}\}$

$\Rightarrow$ 存在$S_1 \subseteq S$，使得$S_1 \subseteq \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}}_{v[X:=S_1]}$ 且$r \in S_1$

$\Rightarrow$ 存在$S_1 \subseteq S_2 \subseteq S$且$(\mathscr{M}', v') \leftrightarrow_V (\mathscr{M}, v[X := S_2])$（有Var-$V$-互模拟关系$\mathscr{B}$），使得$(\mathscr{M}', v')$ $\models \varphi$且$S_2 \subseteq \|\mathrm{F}_\mu(\varphi, V)\|^{\mathscr{M}}_{v[X:=S_2]}$

$\Rightarrow$ 存在$(\mathscr{M}', v') \leftrightarrow_V (\mathscr{M}, v[X := S_2])$，使得$\mathscr{B}(S_2) \subseteq \|\varphi\|^{\mathscr{M}'}_{v'[X:=\mathscr{B}(S_2)]}$

$\Rightarrow$ 存在$\mathscr{B}(S_2) \subseteq S'$，使得$\mathscr{B}(S_2) \subseteq \|\varphi\|^{\mathscr{M}'}_{v'[X:=\mathscr{B}(S_2)]}$且$r' \in \mathscr{B}(S_2)$ $((r, r') \in \mathscr{B}, r \in S_2)$

$\Rightarrow r' \in \bigcup\{S'_1 \subseteq S' \mid S'_1 \subseteq \|\varphi\|^{\mathscr{M}'}_{v'[X:=S'_1]}\}$

$\Rightarrow (\mathscr{M}', v') \models \nu X.\varphi$

$\Rightarrow (\mathscr{M}', v') \models \mathrm{F}_\mu(\nu X.\varphi, V)$且$(\mathscr{M}', v') \leftrightarrow_{\langle\{X\}, V\rangle} (\mathscr{M}, v)$

$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\nu X.\varphi, V)$ (引理 3.5)

(iv) 可类似地证明。 □

当命题3.10中的公式ϕ为命题公式时，从$\mathrm{AX}\phi$ $(\mathrm{EX}\phi)$ 或$\mu X.\varphi$ 中遗忘原子命题可以使用命题逻辑遗忘的计算方法来计算。

前文证明了μ-句子遗忘任意原子命题集的结果总是μ-句子。这里讨论一种μ-公式的子类：不含有不定点操作的μ-公式集，记为**X-类**。通过等值式：$\mathrm{AX}\varphi_1 \wedge \mathrm{AX}\varphi_2 \equiv \mathrm{AX}(\varphi_1 \wedge \varphi_2)$ 和$\mathrm{EX}\varphi_1 \vee \mathrm{EX}\varphi_2 \equiv \mathrm{EX}(\varphi_1 \vee \varphi_2)$，可以将X-类中的任意公式转换为具有下面形

式的公式的析取：

$$\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n, \tag{3.1}$$

其中φ_0是不含有时序算子的X-类中的公式，φ_i $(1 \leq i \leq n)$为X-类中的公式，且任意φ_i $(0 \leq i \leq n)$ 都有可能缺失。

引理 3.6. *令$V \subseteq \mathscr{A}$ 为原子命题集，$\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n$为具有形式(3.1)的可满足公式，则*

$$\begin{aligned}
&\mathrm{F}_\mu(\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n, V) \\
&\quad \equiv \mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{F}_\mu(\mathrm{AX}\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V) \\
&\quad \equiv \mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{AXF}_\mu(\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{EXF}_\mu(\varphi_i \wedge \varphi_1, V).
\end{aligned}$$

证明. 由命题 3.10的(i)和(ii)可知，

$$\begin{aligned}
&\mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{F}_\mu(\mathrm{AX}\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V) \\
\equiv\ &\mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{AXF}_\mu(\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{EXF}_\mu(\varphi_i \wedge \varphi_1, V).
\end{aligned}$$

$(\Rightarrow)$ $(\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n, V)$
$\Rightarrow$ 存在$(\mathscr{M}, v) \leftrightarrow_V (\mathscr{M}', v')$ 使得$(\mathscr{M}', v') \models \varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n$
$\Rightarrow (\mathscr{M}', v') \models \varphi_0$、$(\mathscr{M}', v') \models \mathrm{AX}\varphi_1$和$(\mathscr{M}', v') \models \mathrm{EX}\varphi_i$，其中$2 \leq i \leq n$
$\Rightarrow (\mathscr{M}', v') \models \varphi_0$、$(\mathscr{M}', v') \models \mathrm{AX}\varphi_1$和$(\mathscr{M}', v') \models \mathrm{EX}(\varphi_i \wedge \varphi_1)$，其中$2 \leq i \leq n$　$((\mathscr{M}, v) \models \mathrm{AX}\varphi_1)$
$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0, V)$、$(\mathscr{M}, v) \models \mathrm{F}_\mu(\mathrm{AX}\varphi_1, V)$和$(\mathscr{M}, v) \models \mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V)$，其中$2 \leq i \leq n$　$((\mathscr{M}, v) \leftrightarrow_V (\mathscr{M}', v'), \mathrm{IR}(\mathrm{F}_\mu(\varphi_0, V), V), \mathrm{IR}(\mathrm{F}_\mu(\mathrm{AX}\varphi_1, V), V), \mathrm{IR}(\mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V), V))$
$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{F}_\mu(\mathrm{AX}\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V)$
$(\Leftarrow)$ $(\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{F}_\mu(\mathrm{AX}\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{F}_\mu(\mathrm{EX}(\varphi_i \wedge \varphi_1), V)$，其中$\mathscr{M} = (S, r, R, L)$
$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0, V) \wedge \mathrm{AXF}_\mu(\varphi_1, V) \wedge \bigwedge_{2 \leq i \leq n} \mathrm{EXF}_\mu(\varphi_i \wedge \varphi_1, V)$
$\Rightarrow (\mathscr{M}, r, v) \models \mathrm{F}_\mu(\varphi_0, V)$，对任意$(r, r') \in R$，$(\mathscr{M}, r', v) \models \mathrm{F}_\mu(\varphi_1, V)$，且对任意$2 \leq i \leq n$，存在$(r, r'') \in R$ 使得$(\mathscr{M}, r'', v) \models \mathrm{F}_\mu(\varphi_i \wedge \varphi_1, V)$
$\Rightarrow$ 存在$(\mathscr{M}', v) \models \varphi_0$ 且$(\mathscr{M}', v) \leftrightarrow_V (\mathscr{M}, v)$，其中$\mathscr{M}' = (S, r, R, L')$；对任意$(r, r_j) \in R$，存在$(\mathscr{M}'_j, r'_j, v'_j)$ 使得$(\mathscr{M}'_j, r'_j, v'_j) \leftrightarrow_V (\mathscr{M}_{r_j}, r_j, v)$ 且$(\mathscr{M}'_j, r'_j, v'_j) \models \varphi_1$；且对任意$(r, r_j) \in R$

且$(\mathscr{M}_{r_j}, r_j, v) \models \mathrm{F}_\mu(\varphi_i \wedge \varphi_1, V)$ $(2 \leq i \leq n)$，存在$(\mathscr{M}'_j, r'_j, v'_j) \leftrightarrow_V (\mathscr{M}_{r_j}, r_j, v)$ 使得$(\mathscr{M}'_j, r'_j, v'_j)$ $\models \varphi_i \wedge \varphi_1$ $(j \geq 0$，当$x \neq y$时，假设$S_x \cap S_y = \emptyset)$，其中$\{r_0, r_1, \ldots\} = \{r' \mid (r, r') \in R\}$，$\mathscr{M}'_j = (S'_j, R'_j, L'_j, r'_j)$

$\Rightarrow$ 存在$(\mathscr{M}^*, v^*) \leftrightarrow_V (\mathscr{M}, v)$，使得$(\mathscr{M}^*, v^*) \models \varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \bigwedge_{2 \leq i \leq n} \mathrm{EX}(\varphi_i \wedge \varphi_1)$，其中$\mathscr{M}^* = (S^*, R^*, L^*, r)$且

- $S^* = \{r\} \cup \bigcup_{j \geq 0} S'_j$，
- $R^* = \{(r, r'_j) \mid j \geq 0\} \cup \bigcup_{j \geq 0} R'_j$，
- $L^*(r) = L'(r)$，且对任意$s \in S'_j$，$L^*(s) = L'_j(s)$，
- 对任意$X \in \mathscr{V}$，

$$v^*(X) = \begin{cases} \bigcup_{j \geq 0} v'_j(X), & \text{若} r \notin v(X); \\ \bigcup_{j \geq 0} v'_j(X) \cup \{r\}, & \text{否则。} \end{cases}$$

$\Rightarrow (\mathscr{M}, v) \models \mathrm{F}_\mu(\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n, V)$。 □

给定X-类中的公式φ，公式φ的度（记为$degree(\varphi)$）定义如下：

$$\begin{aligned} &degree(X) = degree(p) = degree(\neg p) = 0, \\ &degree(\mathrm{AX}\psi) = degree(\psi) + 1, \\ &degree(\mathrm{EX}\psi) = degree(\psi) + 1, \\ &degree(\psi_1 * \psi_2) = \max\{degree(\psi_1), degree(\psi_2)\}, \end{aligned}$$

其中$X \in \mathscr{V}$、$p \in \mathscr{A}$和$* \in \{\vee, \wedge\}$。

命题 3.11. *若$V \subseteq \mathscr{A}$为原子命题集、φ为X-类中的公式，则存在X-类中的公式ψ使得$\psi \equiv \mathrm{F}_\mu(\varphi, V)$。*

证明. 假定$degree(\varphi) = n$；通过归纳公式的度来证明这一结论。

基始. $n = 0$. 令φ是一个可能含有自由变元的度为0的μ-公式。由定理 3.6和命题 3.8 (vi)可知$\mathrm{F}_\mu(\varphi, V) = Forget(\varphi, V)$。

归纳步. 假设对所有$degree(\varphi_1) \leq k$的X-类中公式φ_1，$\mathrm{F}_\mu(\varphi_1, V)$在X-类中。证明对任意公式$\varphi_2$且$degree(\varphi_2) = k+1$，$\mathrm{F}_\mu(\varphi_2, V)$ 在X-类中。

φ_2可以被转换为形如公式(3.1)的公式的析取φ'，且$degree(\varphi') \leq k+1$（因为在转换过程中不会引入新的嵌套模态词）。

不失一般性地，假设

$$\varphi' = \bigvee_{1\le i\le x} \varphi_{i,0} \wedge \mathrm{AX}\varphi_{i,1} \wedge \mathrm{EX}\varphi_{i,2} \wedge \cdots \wedge \mathrm{EX}\varphi_{i,n_i},$$

且$\varphi_{i,j}$ $(1 \le i \le x, 1 \le j \le n_i)$是$degree(\varphi_{i,j}) \le k$的公式。

由引理 3.6和命题 3.8(iv)可知：

$$\begin{aligned}\mathrm{F}_\mu(\varphi', V) &\equiv \bigvee_{1\le i\le x} \mathrm{F}_\mu(\varphi_{i,0} \wedge \mathrm{AX}\varphi_{i,1} \wedge \mathrm{EX}\varphi_{i,2} \wedge \cdots \wedge \mathrm{EX}\varphi_{i,n_i}, V)\\ &\equiv \bigvee_{1\le i\le x} (\mathrm{F}_\mu(\varphi_{i,0}, V) \wedge \mathrm{AXF}_\mu(\varphi_{i,1}, V) \wedge \bigwedge_{2\le j\le n_i} \mathrm{EXF}_\mu(\varphi_{i,1} \wedge \varphi_{i,j}, V))。\end{aligned}$$

因此，由归纳假设可知$\mathrm{F}_\mu(\varphi', V)$在X-类中。 □

例 3.6. 令$\varphi_1 = X \wedge p$、$\varphi_2 = \mathrm{AX}(c \wedge \mathrm{EX}d) \wedge \mathrm{AX}e$、$\varphi_3 = \mathrm{EX}\neg d \wedge (\mathrm{EX}\neg p \vee \mathrm{EX}p)$、$\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$且$V = \{e, d\}$，其中$X \in \mathscr{V}$且$p, c, d, e$为原子命题。

如下计算公式φ的度：

$$\begin{aligned}degree(\varphi) &= \max\{degree(\varphi_1), degree(\varphi_2 \wedge \varphi_3)\}\\ &= \max\{0, \max\{degree(\varphi_2), degree(\varphi_3)\}\\ &= 2,\\ degree(\varphi_1) &= 0,\\ degree(\varphi_2) &= \max\{degree(\mathrm{AX}(c \wedge \mathrm{EX}d)), degree(\mathrm{AX}e)\}\\ &= \max\{\max\{0,1\}+1, 1\}\\ &= 2,\\ degree(\varphi_3) &= \max\{degree(\mathrm{EX}\neg d), degree(\mathrm{EX}\neg p \vee \mathrm{EX}p)\}\\ &= \max\{1, \max\{1,1\}\}\\ &= 1.\end{aligned}$$

此外，公式φ可如下转换为具有形式(3.1)的公式的析取：

$$\begin{aligned}\varphi &= \varphi_1 \wedge \varphi_2 \wedge \varphi_3\\ &\equiv X \wedge p \wedge \mathrm{AX}(c \wedge e \wedge \mathrm{EX}d) \wedge \mathrm{EX}\neg d \wedge (\mathrm{EX}\neg p \vee \mathrm{EX}p)\\ &\equiv (X \wedge p \wedge \mathrm{AX}(c \wedge e \wedge \mathrm{EX}d) \wedge \mathrm{EX}\neg d \wedge \mathrm{EX}\neg p) \vee\\ &\quad (X \wedge p \wedge \mathrm{AX}(c \wedge e \wedge \mathrm{EX}d) \wedge \mathrm{EX}\neg d \wedge \mathrm{EX}p).\end{aligned}$$

则从φ中遗忘V的结果为：

$$\begin{aligned}
\mathrm{F}_\mu(\varphi,V) \equiv\ & \mathrm{F}_\mu(X\wedge p\wedge \mathrm{AX}(c\wedge e\wedge \mathrm{EX}d)\wedge \mathrm{EX}\neg d\wedge \mathrm{EX}\neg p,V)\vee\\
& \mathrm{F}_\mu(X\wedge p\wedge \mathrm{AX}(c\wedge e\wedge \mathrm{EX}d)\wedge \mathrm{EX}\neg d\wedge \mathrm{EX}p,V)\\
\equiv\ & (X\wedge p\wedge \mathrm{AX}\mathrm{F}_\mu(c\wedge e\wedge \mathrm{EX}d,V)\wedge\\
& \mathrm{EX}\mathrm{F}_\mu(\neg d\wedge c\wedge e\wedge \mathrm{EX}d,V)\wedge \mathrm{EX}\mathrm{F}_\mu(\neg p\wedge c\wedge e\wedge \mathrm{EX}d,V))\vee\\
& (X\wedge p\wedge \mathrm{AX}\mathrm{F}_\mu(c\wedge e\wedge \mathrm{EX}d,V)\wedge\\
& \mathrm{EX}\mathrm{F}_\mu(\neg d\wedge c\wedge e\wedge \mathrm{EX}d,V)\wedge \mathrm{EX}\mathrm{F}_\mu(p\wedge c\wedge e\wedge \mathrm{EX}d,V))\\
\equiv\ & (X\wedge p\wedge \mathrm{AX}c\wedge \mathrm{EX}c\wedge \mathrm{EX}(\neg p\wedge c))\vee(X\wedge p\wedge \mathrm{AX}c\wedge \mathrm{EX}c\wedge \mathrm{EX}(p\wedge c))\\
\equiv\ & X\wedge p\wedge \mathrm{AX}c\wedge \mathrm{EX}c\wedge(\mathrm{EX}(\neg p\wedge c)\vee \mathrm{EX}(p\wedge c)).
\end{aligned}$$

3.2.3 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[83]，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[83]。因此，下面的命题成立。

推论 3.6. *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. 当φ为吸取公式时，$\mathrm{F}_\mu(\varphi,\{p\})$可通过将$\varphi$中的文字$p$和$\neg p$同时替换为$\top$得到。这只需扫描一遍公式$\varphi$，因此只需要$\varphi$大小的线性时间。 □

因此，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 3.7. 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$V=\{ch\}$，如下计算从这些公式里遗忘$V$：

(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-句子中遗忘某个原子命题集的模型）也是不容易的。

命题 3.12 (模型检测). *给定一个有限的Kripke结构$\mathscr{M}$、一个μ-句子φ和原子命题集$V\subseteq\mathscr{A}$。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∩*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[101]）A_φ，否则需要指数时间构造其对应的μ-自动机[2]. 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，假设$V \subseteq Var(\varphi)$且$V = \{p\}$。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$q \in Q$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$\mathscr{N} \leftrightarrow_{\{p\}} \mathscr{N}'$[103]，即$\mathscr{B}$和与$\mathrm{F}_\mu(\varphi, V)$等价的$\mu$-句子对应。

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∩co-NP[101]中。 □

给定μ-句子φ和ψ，V为原子命题集。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[117]）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$\mathrm{F}_\mu(\varphi, V) \models \mathrm{F}_\mu(\psi, V)$,

定理 3.7 (Entailment)**.** 给定μ-句子φ和ψ，V为原子命题集，则：

(i) 判定$\mathrm{F}_\mu(\varphi, V) \models^? \psi$是EXPTIME-完全的，

(ii) 判定$\psi \models^? \mathrm{F}_\mu(\varphi, V)$在EXPTIME里，

(iii) 判定$\mathrm{F}_\mu(\varphi, V) \models^? \mathrm{F}_\mu(\psi, V)$在EXPTIME里。

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3.12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[118]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[118])。

[2] 个人通信: Giovanna D'Agostino, 2020.

因此，判定是否$\mathrm{F}_\mu(\varphi,V)\models^?\psi$是EXPTIME的。

EXPTIME-难. $\mathrm{F}_\mu(\varphi,V)\models\psi$ 当且仅当$\neg\mathrm{F}_\mu(\varphi,V)\vee\psi$是有效的。令$\psi\equiv\bot$，$\mathrm{F}_\mu(\varphi,V)\models\bot$ 当且仅当$\varphi\models\bot$，判定$\varphi\models^?\bot$是EXPTIME-完全的[101]。因此原问题是EXPTIME-完全的。

(ii) 在EXPTIME中. 可类似于(i)来证明。

(iii) 因为$\mathrm{F}_\mu(\varphi,V)\models\mathrm{F}_\mu(\psi,V)$当且仅当$\varphi\models\mathrm{F}_\mu(\psi,V)$，所以这可类似(ii)来证明。

□

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[119]，及“Var-match”问题:

(i) [Var-independence] 公式φ是否独立于原子命题集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$\mathrm{F}_\mu(\varphi,V)\equiv\mathrm{F}_\mu(\psi,V)$。

推论 3.7. *给定μ-句子φ和ψ，V为原子命题集。则下面的判定问题在EXPTIME里。*

(i) 判定$\psi\equiv^?\mathrm{F}_\mu(\varphi,V)$，
(ii) 判定$\mathrm{F}_\mu(\varphi,V)\equiv^?\varphi$，
(iii) 判定$\mathrm{F}_\mu(\varphi,V)\equiv^?\mathrm{F}_\mu(\psi,V)$。

3.3 本章小结

本章定义了CTL和μ-公式的遗忘，并探索了遗忘的一般性质，这些性质包括：分解性、切片性、同质性及命题逻辑也满足的一些性质。这些基本性质为遗忘的计算提供了基本准则。此外，证明了本文所给出的遗忘是命题逻辑下遗忘的扩展；表示性定理表明了遗忘与四条基本公设之间的充分必要关系。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。

为了定义CTL公式的遗忘给出了Ind-结构之间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。这与遗忘所描述的“遗忘不想考虑的原子命题应不影响剩余原子命题上的结论”一致。因此，使用V-互模拟刻画原公式与遗忘结果的模型之间的关系，从而定义CTL遗忘。

与CTL不同，μ-演算是一种带变元的比CTL表达能力强的逻辑系统。本章给出了包含变元的互模拟定义——变元-命题-互模拟，并证明了变元-命题-互模拟对μ-公式的不变性，即：若两个赋值是$\mathscr{V}_1$-V-互模拟的（$\mathscr{V}_1\subseteq\mathscr{V}$且$V\subseteq\mathscr{A}$），则对任意$(V\cup\mathscr{V}_1)$-无关的公式这两个赋值同时满足（或不满足）该公式。证明了遗忘对μ-句子和不含有不动

点操作的μ-公式总是存在的，且μ-句子遗忘与均匀插值是等价的。最后，μ-析取公式和μ-句子遗忘的模型检测分别在NP$\cap$*co*-NP和EXPTIME里；而μ-句子遗忘的蕴涵问题在EXPTIME里，且部分蕴含问题是EXPTIME-完全的。

第四章　遗忘理论在反应式系统中的应用

遗忘可以用于从本体中抽取摘要、隐蔽敏感信息、计算逻辑差等；*WSC*和*SNC*在智能规划和形式化验证里有重要作用，例：在*2003*年，*Lin*使用*WSC*和*SNC*计算规划问题中的后继状态公理。本章针对第一章提出的问题*(2)*，研究*CTL*和μ-演算中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$V \subseteq Var(\varphi)$，若满足$q \in Var(\varphi) - V$，则q在φ和V上的*SNC*等价于公式$\varphi \wedge q$遗忘不在V中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）在本章看作是一个有限的初始结构，任意有限的初始结构（$\mathscr{K}$）都能用其特征公式来表示（**给定有限初始结构，如何计算其特征公式将在第五章中的介绍，本章只需认为其是一个*CTL*公式**），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极小改变现有知识的模型来使其适应新信息，这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*4.1*节给出最强必要条件和最弱充分条件的定义，探索这两个概念的相关性质，并给出基于遗忘的计算方法；其次，第*4.2*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等价；最后，总结本章的研究工作。

4.1　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 4.1 (充分和必要条件)**.** 给定两个公式φ和ψ，$V \subseteq Var(\varphi)$，$q \in Var(\varphi) - V$ 和$Var(\psi) \subseteq V$。

- 若$\varphi \models q \to \psi$，则称$\psi$是$q$在$V$和$\varphi$上的必要条件（necessary condition，NC）；
- 若$\varphi \models \psi \to q$，则称$\psi$是$q$在$V$和$\varphi$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q在V和φ上的必要条件ψ'，都有$\varphi \models \psi \to \psi'$，则称$\psi$是$q$在$V$和$\varphi$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q在V和φ上的充分条件ψ'，都有$\varphi \models \psi' \to \psi$，则称$\psi$是$q$在$V$和$\varphi$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。此外，如果公式ψ和ψ'都是q在V和φ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 4.1 (对偶性)**.** 令V、q、φ和ψ为定义*4.1*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) 假定ψ是q在V和φ上的SNC，ψ'是$\neg q$在V和φ上SC。

$\varphi \models q \to \psi$

$\Rightarrow \varphi \models \neg\psi \to \neg q$。

这表明$\neg\psi$是$\neg q$在V和φ的SC。

$\varphi \models \psi' \to \neg q$

$\Rightarrow \varphi \models q \to \neg\psi'$，即：$\neg\psi'$是$q$在$V$和$\varphi$上的NC

$\Rightarrow \varphi \models \psi \to \neg\psi'$　　(假设)

$\Rightarrow \varphi \models \psi' \to \neg\psi$。

另一个方向可以类似地证明。

(ii) 可以类似SNC的情形证明。　□

在定义4.1中，用公式α替换q，$V \subseteq Var(\alpha) \cup Var(\phi)$，则为公式的最强必要条件和最弱充分条件的定义。下面的命题表明了原子命题的充分（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 4.2. 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。$\varphi$是集合$V$上的公式，则$\varphi$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. 这里证明SNC的情形，WSC的情形可以类似地证明。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。

($\Rightarrow$) 下证$SNC(\varphi,\alpha,V,\Gamma) \Rightarrow SNC(\varphi,q,V,\Gamma')$。假设$\varphi'$是$q$在$V$和$\Gamma'$上的NC。

$\Gamma \models \alpha \to \varphi$ (已知)
$\Rightarrow \Gamma \wedge (q \leftrightarrow \alpha) \models \alpha \to \varphi$
$\Rightarrow \Gamma' \models q \to \varphi$ ($q \equiv \alpha$)
$\Gamma' \models q \to \varphi'$ (假设)
$\Rightarrow \Gamma' \models \alpha \to \varphi'$ ($q \equiv \alpha$)
$\Rightarrow \mathrm{F}_{\mu}(\Gamma', \{q\}) \models \alpha \to \varphi'$ (IR$(\alpha \to \varphi', \{q\})$, (**PP**))
$\Rightarrow \Gamma \models \alpha \to \varphi'$ (引理 3.4)
$\Rightarrow \Gamma \models \varphi \to \varphi'$ (已知: $SNC(\varphi, \alpha, V, \Gamma)$)
$\Rightarrow \Gamma' \models \varphi \to \varphi'$.

总之，φ是q在V和Γ'上的NC，且对任意q在V和Γ'上的NCφ'，有$\Gamma' \models \varphi \to \varphi'$。因此，$SNC(\varphi, q, V, \Gamma')$。

($\Leftarrow$) 下证$SNC(\varphi, q, V, \Gamma') \Rightarrow SNC(\varphi, \alpha, V, \Gamma)$。假定$\varphi'$是$\alpha$在$V$和$\Gamma$上的NC。

$\Gamma' \models q \to \varphi$ (已知)
$\Rightarrow \Gamma' \models \alpha \to \varphi$ ($q \equiv \alpha$)
$\Rightarrow \mathrm{F}_{\mu}(\Gamma', \{q\}) \models \alpha \to \varphi$ (IR$(\alpha \to \varphi, \{q\})$, (**PP**))
$\Rightarrow \Gamma \models \alpha \to \varphi$ (引理 3.4)
$\Gamma \models \alpha \to \varphi'$ (假设)
$\Rightarrow \Gamma' \models \alpha \to \varphi'$
$\Rightarrow \Gamma' \models q \to \varphi'$ ($q \equiv \alpha$)
$\Rightarrow \Gamma' \models \varphi \to \varphi'$ (已知：$SNC(\varphi, q, V, \Gamma')$)
$\Rightarrow \mathrm{F}_{\mu}(\Gamma', \{q\}) \models \varphi \to \varphi'$ (IR$(\varphi \to \varphi', \{q\})$, (**PP**))
$\Rightarrow \Gamma \models \varphi \to \varphi'$ (引理 3.4)

总之，φ是α在V和Γ上的NC，且对任意α在V和Γ上的NCφ'，有$\Gamma \models \varphi \to \varphi'$。因此，$SNC(\varphi, \alpha, V, \Gamma)$。 □

例 4.1 (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$\mathscr{K}_2 \models \varphi \to \psi$，即：$\varphi$是$\psi$在$V$和$\mathscr{K}_2$上的SC;

(ii) 下证“对任意ψ在V和$\mathscr{K}_2$上的SC φ'都有$\mathscr{K}_2 \models \varphi' \to \varphi$”。易知若$\mathscr{K}_2 \not\models \varphi'$，则$\mathscr{K}_2 \models \varphi' \to \varphi$。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IR$(\varphi', \overline{V})$，所以$\phi \models \mathrm{EX}\neg d$。因此，$\varphi' \models \varphi$且$\mathscr{K}_2 \models \varphi' \to \varphi$。

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 4.1. 给定公式φ、原子命题集$V \subseteq Var(\varphi)$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4.1可知，对任意命题$q'$，$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q', (Var(\varphi) \cup \{q'\}) - V)$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$\varphi \wedge q \models \mathscr{F}$，这说明$\mathscr{F}$是$q$在$V$和$\varphi$上的NC；对任意与$(Var(\varphi) \cup \{q\}) - V$无关的公式$\psi$，若$\varphi \wedge q \models \psi$，则由表达性定理可知$\mathscr{F} \models \psi$。

值得注意的是，定理 4.1中，当考虑的公式为μ-公式时，要求出现在φ中的变元都包含在V中。

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 4.1. 令$\mathscr{K} = (\mathscr{M}, s_0)$为初始结构，其中$\mathscr{M} = (S, R, L, s_0)$为有限原子命题集$\mathscr{A}$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 4.2 (例1的延续). 令$\mathscr{A} = \{d, se, sp, s\}$和$V = \{d, se\}$，求$s$在$V$和初始结构$\mathscr{K} = (\mathscr{M}, s_0)$上的WSC，其中$\mathscr{M}$为例 1.1中初始状态为$s_0$的汽车制造企业模型结构。

由定理 4.1可知，s在V和初始结构$\mathscr{K} = (\mathscr{M}, s_0)$上的WSC为$\neg\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}) \wedge \neg s, \{s\} \cup \{sp\})$。

由于涉及到后文中遗忘的计算方法，本例的详细计算过程放到第 5.1。

4.2 知识更新

本小节介绍遗忘的另一个应用：知识更新（Knowledge update）。具体说来，本节将使用遗忘定义知识更新，使得用这种方式定义的知识更新满足下面由Katsuno和Mendelzon提出的基本条件[1]：

- (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) $(\Gamma_1 \vee \Gamma_2) \diamond \varphi \equiv (\Gamma_1 \diamond \varphi) \vee (\Gamma_2 \diamond \varphi)$。

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。

本小节假设所有初始结构都是有限的，即：状态来源于有限状态空间且$\mathscr{A}$为有限原子命题集。

由第五章可知，任意$\mathscr{A}$上的有限初始结构$\mathscr{M}$（为了简化符号，本节用初始Kripke结构$\mathscr{M}$代替初始结构$(\mathscr{M}, s_0)$）都能用一个CTL公式——特征公式$\mathscr{F}_{\mathscr{A}}(\mathscr{M})$来表示。给定公式$\varphi$和$\psi$，$V_{min} \subseteq \mathscr{A}$为使得$\mathrm{F}_{\mathrm{CTL}}(\varphi, V_{min}) \wedge \psi$可满足的极小子集。此外，记

$$\bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \psi)$$

为所有$\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \psi$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 4.2. 给定公式Γ和φ。知识更新操作$\diamond_{CTL}$定义如下：

$$Mod(\Gamma \diamond_{CTL} \varphi) = \bigcup_{\mathscr{M} \in Mod(\Gamma)} \bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \varphi),$$

其中，$\mathscr{F}_{\mathscr{A}}(\mathscr{M})$是$\mathscr{M}$在$\mathscr{A}$上的特征公式，$V_{min} \subseteq \mathscr{A}$是使得$\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min})$可满足的极小子集。

直观上，$\Gamma \diamond_{\mathrm{CTL}} \varphi$表示：通过极小化改变$\Gamma$的模型，使其与$\varphi$的模型一致来更新$\Gamma$。所以，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，CTL知识更新也可以用类似于命题逻辑里基于偏序关系的更新来定义：令I，J_1和J_2为三个解释，即：原子命题集；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$\leq_{I,pma}$的极小模型集。

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 4.3. 给定三个有限初始结构$\mathscr{M}$、$\mathscr{M}_1$和$\mathscr{M}_2$，$\mathscr{M}_1$比$\mathscr{M}_2$更接近$\mathscr{M}$（记为$\mathscr{M}_1\leq_{\mathscr{M}}\mathscr{M}_2$），当且仅当对任意$V_2\subseteq\mathscr{A}$，若$\mathscr{M}_2\leftrightarrow_{V_2}\mathscr{M}$，则存在$V_1\subseteq V_2$使得$\mathscr{M}_1\leftrightarrow_{V_1}\mathscr{M}$。$\mathscr{M}_1<_{\mathscr{M}}\mathscr{M}_2$当且仅当$\mathscr{M}_1\leq_{\mathscr{M}}\mathscr{M}_2$且$\mathscr{M}_2\not\leq_{\mathscr{M}}\mathscr{M}_1$。

在这种情形下，若$\mathscr{M}_1\leq_{\mathscr{M}}\mathscr{M}_2$，则存在一个原子命题的极小集$V_{min}$使得$\mathscr{M}_1\leftrightarrow_{V_{min}}\mathscr{M}$，且对任意$V\subseteq\mathscr{A}$，若$\mathscr{M}_2\leftrightarrow_V\mathscr{M}$则$V_{min}\subseteq V$且$\mathscr{M}_1\leftrightarrow_V\mathscr{M}$。

例 4.3. 令$\mathscr{M}=(S,R,L,r)$、$\mathscr{M}_1=(S_1,R_1,L_1,r_1)$、$\mathscr{M}_2=(S_2,R_2,L_2,r_2)$为三个初始结构（如图 4.1），其中$S=S_1=S_2=\{s_0,s_1\}$，$r=r_1=r_2=s_0$，$R=R_1=R_2=\{(s_0,s_1),(s_1,s_1)\}$，$L(s_0)=\{ch,j\}$，$L_1(s_0)=L_2(s_0)=\{ch\}$，$L(s_1)=L_1(s_1)=\emptyset$，$L_2(s_1)=\{j\}$。

可以检查$\mathscr{M}\leftrightarrow_{\{j\}}\mathscr{M}_1$，$\mathscr{M}\leftrightarrow_{\{j,ch\}}\mathscr{M}_2$，$\{j\}\subseteq\{j,ch\}$，且对任意原子命题集$V'\subset\{j\}$（或$V'\subset\{j,ch\}$），有$\mathscr{M}\not\leftrightarrow_{V'}\mathscr{M}_1$（或$\mathscr{M}\not\leftrightarrow_{V'}\mathscr{M}_2$）。因此，$\mathscr{M}_1\leq_{\mathscr{M}}\mathscr{M}_2$。

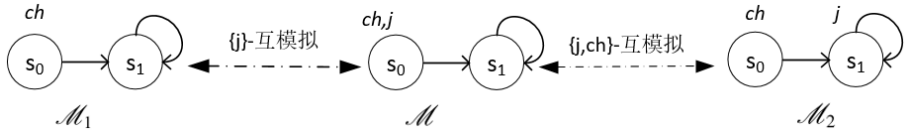


图 4.1: 初始结构间的$\leq_{\mathscr{M}}$关系。

给定有限初始结构集M和有限初始结构$\mathscr{M}$，用$Min(M,\leq_{\mathscr{M}})$表示$M$中关于偏序关系$\leq_{\mathscr{M}}$的极小有限初始结构集。则$\leq_{\mathscr{M}}$与知识更新操作$\diamond_{\mathrm{CTL}}$有如下关系。

定理 4.2. 给定μ-句子Γ和φ，则：

$$Mod(\Gamma\diamond_{CTL}\varphi)=\bigcup_{\mathscr{M}\in Mod(\Gamma)}Min(Mod(\varphi),\leq_{\mathscr{M}}).$$

证明. $\mathscr{M}'\in Mod(\Gamma\diamond_{\mathrm{CTL}}\varphi)$
$\Leftrightarrow$ 存在$\mathscr{M}\models\Gamma$和极小子集$V_{min}\subseteq\mathscr{A}$，使得$\mathscr{M}'\in Mod(\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}),V_{min})\wedge\varphi)$
$\Leftrightarrow$ 存在$\mathscr{M}\models\Gamma$和$V_{min}\subseteq\mathscr{A}$，使得$\mathscr{M}\leftrightarrow_{V_{min}}\mathscr{M}'$和$\mathscr{M}'\models\varphi$，且$V_{min}$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}),V_{min})\wedge\varphi$可满足的$\mathscr{A}$极小子集

$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$，使得对任意$\mathscr{M}'' \models \varphi$，$\mathscr{M}' \leq_{\mathscr{M}} \mathscr{M}''$
$\Leftrightarrow \mathscr{M}' \in Min(Mod(\varphi), \leq_{\mathscr{M}})$。 □

从定理4.2可以看出，通过遗忘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 4.3. *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 4.2可知，$Mod(\Gamma \diamond_{\mu} \varphi) \subseteq Mod(\varphi)$，因此$\Gamma \diamond_{\mu} \varphi \models \varphi$。

(U2). $\mathscr{M}' \in Mod(\Gamma \diamond_{\mu} \varphi)$
$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$，使得对任意$\mathscr{M}'' \models \varphi$ 和$V \subseteq \mathscr{A}$，$\mathscr{M}'' \leftrightarrow_V \mathscr{M}$ 蕴涵$\mathscr{M}' \leftrightarrow_V \mathscr{M}$
$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$和$V_{min} = \emptyset$，使得$\mathscr{M}' \leftrightarrow_{V_{min}} \mathscr{M}$ $(\Gamma \models \varphi)$
$\Leftrightarrow \mathscr{M}' \models \Gamma$。

容易证明$\diamond_{\mu}$满足(U3)和(U4)。

(U5). $\mathscr{M}' \models (\Gamma \diamond_{\mu} \varphi) \wedge \psi$
$\Rightarrow$ 存在$\mathscr{M} \models \Gamma$，使得对任意$\mathscr{M}'' \models \varphi \wedge \psi$ 和$V \subseteq \mathscr{A}$，$\mathscr{M}'' \leftrightarrow_V \mathscr{M}$蕴涵$\mathscr{M}' \leftrightarrow_V \mathscr{M}$
$\Rightarrow$ 存在$\mathscr{M} \models \Gamma$和$V_{min} \subseteq \mathscr{A}$，使得$\mathscr{M}' \leftrightarrow_{V_{min}} \mathscr{M}$和$\mathscr{M}' \models \varphi \wedge \psi$，其中$V_{min}$是使得$\mathscr{M}' \leftrightarrow_{V_{min}} \mathscr{M}$的$\mathscr{A}$的极小子集
$\Rightarrow \mathscr{M}' \models \Gamma \diamond_{\mu} (\varphi \wedge \psi)$。

(U6). $\mathscr{M}' \models \Gamma \diamond_{\text{CTL}} \varphi$
$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$，使得对任意$\mathscr{M}'' \models \varphi$ 和$V \subseteq \mathscr{A}$，$\mathscr{M}'' \leftrightarrow_V \mathscr{M}$ 蕴涵$\mathscr{M}' \leftrightarrow_V \mathscr{M}$
$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$和$V_{min} \subseteq \mathscr{A}$，使得$\mathscr{M}' \leftrightarrow_{V_{min}} \mathscr{M}$，其中$V_{min}$是使得$\text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \varphi$和$\text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \psi$是一致的 $(\Gamma \diamond_{\text{CTL}} \varphi \models \psi, \Gamma \diamond_{\text{CTL}} \psi \models \varphi)$
$\Leftrightarrow$ 存在$\mathscr{M} \models \Gamma$，使得对任意$\mathscr{M}'' \models \psi$ 和$V \subseteq \mathscr{A}$，$\mathscr{M}'' \leftrightarrow_V \mathscr{M}$ 蕴涵$\mathscr{M}' \leftrightarrow_V \mathscr{M}$
$\Leftrightarrow \mathscr{M}' \models \Gamma \diamond_{\text{CTL}} \psi$。

(U7). $\mathscr{M}' \models (\Gamma \diamond_{\text{CTL}} \varphi) \wedge (\Gamma \diamond_{\text{CTL}} \psi)$，且设$\mathscr{M}$是$\Gamma$的唯一模型
$\Rightarrow$ 存在两个极小子集$V_1, V_2 \subseteq \mathscr{A}$，使得$\mathscr{M} \leftrightarrow_{V_1} \mathscr{M}'$ 和$\mathscr{M} \leftrightarrow_{V_2} \mathscr{M}'$
$\Rightarrow$ 存在两个极小子集$V_1, V_2 \subseteq \mathscr{A}$，使得$\mathscr{M}' \models \text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1) \wedge \varphi$ 和$\mathscr{M}' \models \text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_2) \wedge \psi$是一致的
$\Rightarrow \mathscr{M}' \models \text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1 \cap V_2)$，$\mathscr{M}' \leftrightarrow_{V_1 \cap V_2} \mathscr{M}$，$V_1 = V_2$
$\Rightarrow V_1$是使得$\text{F}_{\mu}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1) \wedge (\varphi \vee \psi)$可满足的极小子集
$\Rightarrow \mathscr{M}' \models \Gamma \diamond_{\text{CTL}} (\varphi \vee \psi)$。

(U8). $\mathscr{M} \models (\Gamma_1 \vee \Gamma_2) \diamond_{\text{CTL}} \varphi$
$\Leftrightarrow$ 存在$\mathscr{M}_1 \models \Gamma_1$ (或$\mathscr{M}_1 \models \Gamma_2$) 和一个极小子集$V_{min}$，使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$
$\Leftrightarrow \mathscr{M} \models (\Gamma_1 \diamond_{\text{CTL}} \varphi) \vee (\Gamma_2 \diamond_{\text{CTL}} \varphi)$。 □

例 4.4. 令$\mathscr{A}=\{ch,j\}$、$\varphi=\nu X.j\wedge ch\wedge \text{EXEX}X$、$\psi=\nu X.\neg j\wedge ch\wedge \text{EXEX}X$且Kripke结构的状态空间为$\{s_0,s_1\}$，则用$\psi$更新$\varphi$计算如下：

$$\begin{aligned}
Mod(\varphi)=&\{((1),r=s_0,L(s_0)=\{ch,j\},L(s_1)=\{ch,j\}),\\
&((2),r=s_1,L(s_1)=\{ch,j\},L(s_0)=\{ch,j\}),\\
&((3),r=s_0,L(s_0)=\{ch,j\},L(s_1)=\mathscr{C}),\\
&((4),r=s_1,L(s_1)=\{ch,j\},L(s_0)=\mathscr{C}),\\
&((5),r=s_0,L(s_0)=\{ch,j\},L(s_1)=\mathscr{C}),\\
&((6),r=s_1,L(s_1)=\{ch,j\},L(s_0)=\mathscr{C}),\dots\}\\
Mod(\psi)=&\{((1),r=s_0,L(s_0)=\{ch\},L(s_1)=\{ch\}),\\
&((2),r=s_1,L(s_1)=\{ch\},L(s_0)=\{ch\}),\\
&((3),r=s_0,L(s_0)=\{ch\},L(s_1)=\mathscr{C}),\\
&((4),r=s_1,L(s_1)=\{ch\},L(s_0)=\mathscr{C}),\\
&((5),r=s_0,L(s_0)=\{ch\},L(s_1)=\mathscr{C}),\\
&((6),r=s_1,L(s_1)=\{ch\},L(s_0)=\mathscr{C}),\dots\}
\end{aligned}$$

其中，四元组$((i),r=s_k,L(s_0)=V_1,L(s_1)=V_1)$表示Kripke结构$(S,r,R,L)$，其中$S=\{s_0,s_1\}$、$r=s_k$ $(r\in\{0,1\})$、转换关系如图 4.2中的(i) $(i\in\{1,2,3,4,5,6\}$[1]$)$、s_0 和s_1分别被$V_1\subseteq\{ch,j\}$ 和$V_2\subseteq\{ch,j\}$标记且$\mathscr{C}\in\{\emptyset,\{j\},\{ch\},\{j,ch\}\}$。

$Mod(\varphi\diamond_\mu\psi)=\bigcup_{\mathscr{M}\in Mod(\varphi)}Min(Mod(\psi),\leq_{\mathscr{M}})$，根据定义 4.3容易检查$Mod(\varphi\diamond_\mu\psi)=Mod(\psi)$。直观地说，由于在$\psi$中$j$在偶数状态不再为真、$ch$保持为真且$\psi$和$\varphi$都不知道模型偶数状态的信息，因而用$\psi$更新$\varphi$得到的结果为$\psi$自身。

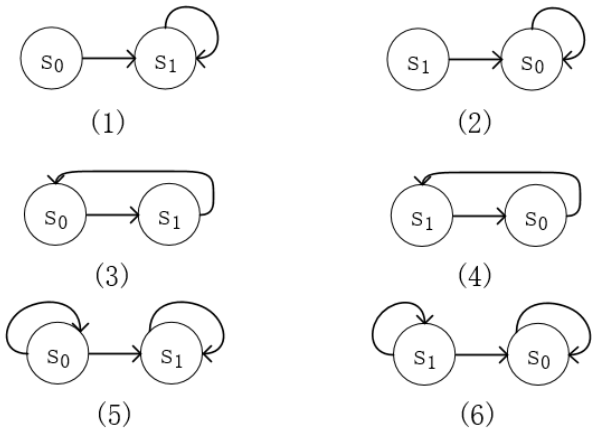


图 4.2: 状态空间为$\{s_0,s_1\}$的六个Kripke结构示意图

[1]这里只列出部分转换关系，其余转换关系可以容易地枚举出来。

4.3 本章小结

本章讨论了如何使用遗忘计算最强必要条件（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，并给出使用遗忘计算原子命题在给定条件下的SNC（WSC）的方法。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。

第五章　CTL遗忘计算方法

第三章探索了*CTL*遗忘的定义及其基本性质，本章给出两种计算*CTL*遗忘的算法：基于模型的算法和基于归结的算法*CTL-forget*。

基于模型的计算考虑固定有限状态空间下的初始结构。本章证明这样的初始结构可由一个*CTL*公式来表示，因而其遗忘结果可以由*CTL*公式表示。

基于归结的计算方法首先将*CTL*公式转化为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句，然后使用归结系统$R^{\succ,S}_{CTL}$中的规则对要遗忘的原子命题归结，最后将得到的结果等价转化为*CTL*公式。基于此，本章详细介绍了基于*Prolog*的*CTL-forget*算法实现，并给出部分实验结果及结果分析。

最后总结本章的主要工作。

5.1　基于模型的有界CTL遗忘计算

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[55]，这表明CTL遗忘不是封闭的。此时，探索CTL下遗忘封闭的情形对深入应用遗忘有重要意义。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析取；最后，提出一种基于模型的方法计算遗忘。

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，则存在一个状态空间是$\mathscr{S}$的子集的初始结构$(\mathscr{M},s_0)$，使得$(\mathscr{M},s_0)\models\varphi$。

由此可见，在这种情形下只有有限个初始结构会被考虑。下文表明在这一约束条件下CTL中的遗忘是封闭的。

本节其余部分组织如下：首先，在第5.1.1节定义有界V-互模拟，然后定义初始结构的特征公式；其次，在第5.1.2节证明约束CTL下的遗忘是封闭的，并给出计算方法；最后，在第5.1.3节给出基于模型的计算遗忘的算法，并分析算法的复杂性。

5.1.1　描述初始结构

本小节介绍与一个初始结构相关的CTL公式——特征公式。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。因此，本小节首先介绍计算树之间的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。

5.1.1.1 计算树V-互模拟

第三章中的定义3.1给出了Ind-结构之间的V-互模拟定义及其相关性质，结构之间的V-互模拟有相同的定义，即：Ind-结构之间V-互模拟具有的性质都可以平凡的迁移到结构之间V-互模拟。这里给出有限结构间有界V-互模拟（与深度n关联的V-互模拟）的定义，并证明V-互模拟与有界V-互模拟在有限结构下等价。因此，第三章遗忘有的性质在本章约束情形下基本也有[120]，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$V \subseteq \mathscr{A}$是原子命题集，$i \in \{1,2\}$，$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$是初始Kripke结构，$\mathscr{K}_i = (\mathscr{M}_i, s_i)$是结构。$\mathscr{B}_n^V$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集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$((\mathscr{M}_1, s_1), (\mathscr{M}_2, s_2)) \in \mathscr{B}_n$。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$n \geq 0$，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$\pi_i = (s_{i,1}, s_{i,2}, \ldots)$，若对于任意$j \in \mathbb{N}_{\geq 1}$[1]，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$\mathscr{B}_{k+1} = \mathscr{B}_k$（$k \in \mathbb{N}$）。

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。

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

证明. 令$V \subseteq \mathscr{A}$、$\mathscr{K}_i = (\mathscr{M}_i, s_i)$ $(i \in \{1,2\})$且$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$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$n \geq 0$，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）。

下证对任意$r_1 \in S_1$和$r_2 \in S_2$，若$(r_1, r_2) \in \mathscr{B}$，则

(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$(w_1, w_2) \in \mathscr{B}$；
(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$n \geq 0$，都有$(r_1, r_2) \in \mathscr{B}_n$，可知$(r_1, r_2) \in \mathscr{B}_0$，这表明$L_1(r_1) - V = L_2(r_2) - V$。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$\mathscr{B}_{k+1} = \mathscr{B}$可知$(r_1, r_2) \in \mathscr{B}_{k+1}$、$w_1 \in S_1$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，下面的推论显然成立。

推论 5.1. *令$V \subseteq \mathscr{A}$和$\mathscr{K}_i = (\mathscr{M}_i, s_i)$（$i \in \{1,2\}$）。若$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$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$V \subseteq \mathscr{A}$和初始Kripke结构$\mathscr{M}_i$（$i = 1,2$）。如果下面条件同时满足：

- $L_1(s_1) - V = L_2(s_2) - V$,
- 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$\mathrm{Tr}_n(s_1)$的子树$\mathrm{Tr}_{n-1}(s_1')$，使得$\mathrm{Tr}_{n-1}(s_1') \leftrightarrow_V \mathrm{Tr}_{n-1}(s_2')$；

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$\mathrm{Tr}_n(s_1) \leftrightarrow_V \mathrm{Tr}_n(s_2)$）。

特别地，当$n = 0$时，只需考虑第一个条件即可。

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

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

证明. 下面从两个方面来证明这一结论。

($\Rightarrow$) 下证“如果$(s_1, s_2) \in \mathscr{B}_n$，则对于任意$0 \leq j \leq n$，有$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$”成立。$(s_1, s_2) \in \mathscr{B}_n$表明$Tr_n(s_1)$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$(s_1, s_{1,1}) \in R_1$，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$(s_2, s_{2,1}) \in R_2$，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$0 \leq j \leq n$，都有$Tr_j(s_1) \leftrightarrow_V Tr_j(s_2)$。

($\Leftarrow$)下证“如果对于任意$0 \leq j \leq n$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，$(s_1, s_2) \in \mathscr{B}_1$。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，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$V \subseteq \mathscr{A}$、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。 □

5.1.1.2 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区别开[2]。下面讨论如何使用CTL公式描述计算树，且表明具有（或没有）V-互模拟关系的计算树的特征公式之间的关系。为此，首先给出计算树特征公式的定义。

[2]相似的方法在Mycielski等人的文章中已被使用[121]，在这篇文章中一元公式的结构通过等价类$\equiv_{\overline{k}}$被描述为Hintikka公式[122]。另一个类似的工作是Yankov-Fine构造[123]。

定义 5.2. 给定原子命题集$V \subseteq \mathscr{A}$、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$\mathscr{F}_V(Tr_n(s))$，$n \geq 0$）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。

通俗一点，$\mathscr{F}_V(\mathrm{Tr}_0(s))$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，则它们在V上的特征公式是逻辑等价的。

引理 5.2. 给定原子命题集$V \subseteq \mathscr{A}$、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$s_x \in S$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$0 \leq m \leq n$，若$\mathrm{Tr}_m(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_m(s')$，则$\mathscr{F}_V(\mathrm{Tr}_m(s)) \equiv \mathscr{F}_V(\mathrm{Tr}_m(s'))$。下证若$\mathrm{Tr}_{n+1}(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n+1}(s')$，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$k = m$、$s_k \in S$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$\mathrm{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k+1}(s'_{k-1})$，所以，对任意$(s_{k-1},s_k)\in R$，存在$(s'_{k-1},s'_k)\in R'$，使得$\mathrm{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k}(s'_k)$，且对任意$(s'_{k-1},s'_k)\in R'$，存在$(s_{k-1},s_k)\in R$，使得$\mathrm{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k}(s'_k)$。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、根为s'、深度为n的计算树特征公式的模型，则s和s'至少属于$\mathscr{B}_n$，即：s和s'能相互模拟至少到第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$Tr_n(s)\leftrightarrow_{\overline{V}} Tr_n(s')$。

证明. (i) **基始**（$n=0$）：从树的特征公式定义可知$L(s)\models\mathscr{F}_V(\mathrm{Tr}_0(s))$，所以，$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s))$。

归纳步（$n>0$）：假设对任意$k\geq 0$，$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_k(s))$，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$(s,s')\in R$，有$(\mathscr{M},s')\models\mathscr{F}_V(\mathrm{Tr}_k(s'))$。因此，$(\mathscr{M},s)\models\mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_k(s')$，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$(s,s')\in R$，都有$(\mathscr{M},s')\models\bigvee_{(s,s')\in R}\mathscr{F}_V(\mathrm{Tr}_k(s'))$。因此，

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

所以，对任意$n\geq 0$，$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s))$。

(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$\mathrm{Tr}_{n-1}(s)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s')$。下证若$(\mathscr{M},s)\models\mathscr{F}_V(\mathrm{Tr}_n(s'))$，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，$(\mathscr{M},s)\models\mathrm{AX}\left(\bigvee_{(s',s'_1)\in R}\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1))\right)$。所以，对于任意$(s,s_1)\in R$，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，$(\mathscr{M},s)\models\bigwedge_{(s',s'_1)\in R'}\mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_{n-1}(s'_1))$。对任意$(s',s'_1)\in R'$，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$\mathrm{Tr}_{n-1}(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_{n-1}(s'_1)$。

□

5.1.1.3　初始结构的特征公式

由V-互模拟的定义和命题5.2自然地可以衍生出一个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。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。

此外，对于给定的初始Kripke结构$\mathscr{M}$和原子命题集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$\mathscr{M}$，存在一个最小数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$\mathscr{M}$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$ch(\mathscr{M},V)=0$。

直观上，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$(s,s') \notin \mathscr{B}_c$；第二类中的状态都是$V$-不可区分的。

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$s' \in S$，$(\mathscr{M},s) \leftrightarrow_{\overline{V}} (\mathscr{M},s')$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)可知$(\mathscr{M},s) \models \phi$，从而对任意$s' \in S$，若$s \leftrightarrow_{\overline{V}} s'$，由定理3.1和$\mathrm{IR}(\phi,\mathscr{A}-V)$可知$(\mathscr{M},s') \models \phi$。

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，可如下定义初始结构的特征公式。

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

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

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$\mathscr{F}_V(\mathrm{Tr}_c(s)) \neq \mathscr{F}_V(\mathrm{Tr}_c(s'))$。特别地，$T(s)$确保了状态$s \in S$被CTL公式描述，而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$\mathscr{K}_2 = (\mathscr{M},s_0)$（图3.1）。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$\overline{V} = \{s,se\}$。

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$(s_2,s') \in R$，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$\mathrm{dis}_V(\mathscr{M},s_1,s_2,1)$。

同理可得：$\mathrm{dis}_V(\mathscr{M},s_0,s_1,0)$、$\mathrm{dis}_V(\mathscr{M},s_1,s_3',1)$、$\mathrm{dis}_V(\mathscr{M},s_0,s_2,0)$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：

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

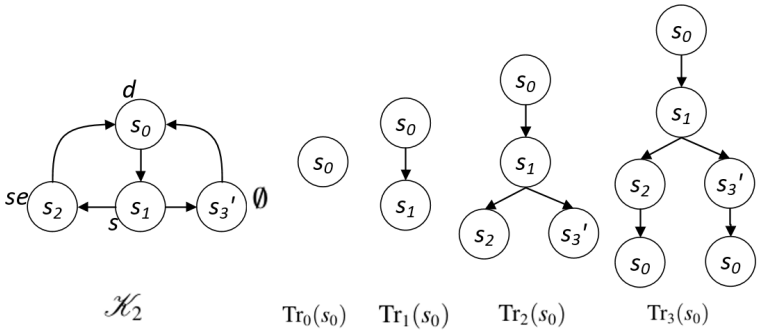


图 5.1: 初始结构$\mathscr{K}_2$（源于图3.1）及其计算树示意图

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，特征公式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：下文中给定系统下的最弱充分条件计算。直观上，特征公式保持了给定初始结构在原子命题集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，$\mathrm{IR}(\mathscr{F}_V(\mathscr{M}, s_0), \overline{V})$。为了证明上述结论成立，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$(\mathscr{M}, s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s_0))$。下证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)

  $\forall (s, s_1) \in R$：

  $(\mathscr{M}, s_1) \models \mathscr{F}_V(\mathrm{Tr}_c(s_1))$ $(s_1 \leftrightarrow_{\overline{V}} s_1)$

  $\Rightarrow (\mathscr{M}, s) \models \bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))$

  $\Rightarrow (\mathscr{M}, s_0) \models \bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1))$ $(\mathrm{IR}(\bigwedge_{(s,s_1)\in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_c(s_1)), \overline{V}), s_0 \leftrightarrow_{\overline{V}} s)$

  $\forall (s, s_1) \in R$：

  $(\mathscr{M}, s_1) \models \bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2))$

  $\Rightarrow (\mathscr{M}, s) \models \mathrm{AX} \left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2)) \right)$

  $\Rightarrow (\mathscr{M}, s_0) \models \mathrm{AX} \left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2)) \right)$ $(\mathrm{IR}(\mathrm{AX} \left(\bigvee_{(s,s_2)\in R} \mathscr{F}_V(\mathrm{Tr}_c(s_2)) \right), \overline{V}), s_0 \leftrightarrow_{\overline{V}} s)$

  $\Rightarrow (\mathscr{M}, s_0) \models G(\mathscr{M}, s)$。

对任意其它能从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$s \in S$，$(\mathscr{M}, s_0) \models \mathrm{AG}\ G(\mathscr{M}, s)$，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$n \geq 0$，$Tr_n(s_0) \leftrightarrow_{\overline{V}} Tr_n(s'_0)$。

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$k > 0$，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$0 \leq i \leq k-1$，$s_{i+1}$（$s'_{i+1}$）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$L(s_k) - \overline{V} = L'(s'_k) - \overline{V}$。

在讨论其它点时，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) 下证$\forall(s_k,s_{k+1}) \in R$，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) 下证$\forall(s'_k,s'_{k+1}) \in R'$，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易得。

□

上文描述了初始结构特征公式的计算，下面给出例4.2的详细计算过程。

例 5.2 (例 4.2的延续)**.** 令$\mathscr{A} = \{d, se, sp, s\}$和$V = \{d, se\}$，求$s$在$V$和初始结构$\mathscr{K} = (\mathscr{M}, s_0)$上的WSC，其中$\mathscr{M}$为例 1.1中初始状态为$s_0$的汽车制造企业模型结构。

由定理 4.1可知，s在V和初始结构$\mathscr{K} = (\mathscr{M}, s_0)$上的WSC为$\neg \mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}) \wedge \neg s, \{s\}$

$\cup\{sp\}$)。

$$\begin{aligned}&\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K})\wedge\neg s,\{s\}\cup\{sp\})\\&\equiv\mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K})\wedge\neg s,\{sp\}),\{s\})\\&\equiv\mathrm{F}_{\mathrm{CTL}}(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}),\{sp\})\wedge\neg s,\{s\}).\end{aligned}$$

$\mathscr{K}_1\models\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}),\{sp\})$
$\Leftrightarrow$ 存在$\mathscr{K}_1'\leftrightarrow_{\{sp\}}\mathscr{K}_1$，使得$\mathscr{K}_1'\models\mathscr{F}_{\mathscr{A}}(\mathscr{K})$和
$\Leftrightarrow$ 存在$\mathscr{K}_1'\leftrightarrow_{\{sp\}}\mathscr{K}_1$，使得$\mathscr{K}_1'\leftrightarrow_{\emptyset}\mathscr{K}$
$\Leftrightarrow$ 存在$\mathscr{K}_1'\leftrightarrow_{\{sp\}}\mathscr{K}$和$\mathscr{K}_1'\models\mathscr{F}_{V\cup\{s\}}(\mathscr{K})$ (定理 5.1)
$\Leftrightarrow\mathscr{K}_1\models\mathscr{F}_{V\cup\{s\}}(\mathscr{K})$。

由此可知，$\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}),\{sp\})\equiv\mathscr{F}_{V\cup\{s\}}(\mathscr{K})$。所以要计算$\neg\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K})\wedge\neg s,\{s\}\cup\{sp\})$，只需计算$\neg\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{V\cup\{s\}}(\mathscr{K})\wedge\neg s,\{s\})$。令$V'=V\cup\{s\}=\{d,se,s\}$，则$\overline{V'}=\{sp\}$。

易知，除了s_2和s_4是$\overline{V'}$-互模拟的以外，其余都是V'-可区分的，且$\mathrm{dis}_{V'}(\mathscr{M},s_i,s_j,0)$，其中$i\neq j$且$i$和$j$不能同时分别取值2和4。因此，可以计算$\mathscr{M}$关于$V'$的特征数为：

$$ch(\mathscr{M},V')=\max\{k\mid s,s'\in S\text{ 且}\mathrm{dis}_{V'}(\mathscr{M},s,s',k)\}=0。$$

所以，可以由以下步骤计算$\mathscr{K}_2$关于V的特征公式：

$$\begin{aligned}&\mathscr{F}_{V'}(\mathrm{Tr}_0(s_0))=d\wedge\neg se\wedge\neg s,\qquad\mathscr{F}_{V'}(\mathrm{Tr}_0(s_1))=\neg d\wedge\neg se\wedge s,\\&\mathscr{F}_{V'}(\mathrm{Tr}_0(s_2))=\mathscr{F}_{V'}(\mathrm{Tr}_0(s_4))=\neg d\wedge se\wedge\neg s,\qquad\mathscr{F}_{V'}(\mathrm{Tr}_0(s_3)=\neg d\wedge\neg se\wedge\neg s,\\&\mathscr{F}_{V'}(\mathscr{M},s_0)\equiv d\wedge\neg se\wedge\neg s\wedge\\&\qquad\mathrm{AG}(d\wedge\neg se\wedge\neg s\rightarrow\mathrm{AX}(\neg d\wedge\neg se\wedge s))\wedge\\&\qquad\mathrm{AG}(\neg d\wedge\neg se\wedge s\rightarrow[\mathrm{AX}((\neg d\wedge se\wedge\neg s)\vee(\neg d\wedge\neg se\wedge\neg s))\wedge\\&\qquad\mathrm{EX}(\neg d\wedge se\wedge\neg s)\wedge\mathrm{EX}(\neg d\wedge\neg se\wedge\neg s)])\wedge\\&\qquad\mathrm{AG}(\neg d\wedge\neg se\wedge\neg s\rightarrow\mathrm{AX}(d\wedge\neg se\wedge\neg s))\wedge\\&\qquad\mathrm{AG}(\neg d\wedge se\wedge\neg s\rightarrow\mathrm{AX}(d\wedge\neg se\wedge\neg s)).\end{aligned}$$

令：

$$\begin{aligned}&\varphi_1=d\wedge\neg se\wedge\neg s\rightarrow\mathrm{AX}(\neg d\wedge\neg se\wedge s)\equiv\neg d\vee se\vee s\vee\mathrm{AX}(\neg d\wedge\neg se\wedge s),\\&\varphi_2=\neg d\wedge\neg se\wedge s\rightarrow\mathrm{AX}(\neg d\wedge se\wedge\neg s)\equiv d\vee se\vee\neg s\vee\end{aligned}$$

$$[\mathrm{AX}((\neg d \wedge se \wedge \neg s) \vee (\neg d \wedge \neg se \wedge \neg s)) \wedge \mathrm{EX}(\neg d \wedge se \wedge \neg s) \wedge \mathrm{EX}(\neg d \wedge \neg se \wedge \neg s)],$$

$$\varphi_3 = \neg d \wedge \neg se \wedge \neg s \rightarrow \mathrm{AX}(d \wedge \neg se \wedge \neg s) \equiv d \vee se \vee s \vee \mathrm{AX}(d \wedge \neg se \wedge \neg s),$$

$$\varphi_4 = \neg d \wedge se \wedge \neg s \rightarrow \mathrm{AX}(d \wedge \neg se \wedge \neg s) \equiv d \vee \neg se \vee s \vee \mathrm{AX}(d \wedge \neg se \wedge \neg s).$$

此外，令

$$\psi_1 = \neg d \vee se \vee s \qquad \psi_2 = d \vee se \vee \neg s$$

$$\psi_3 = d \vee se \vee s \qquad \psi_4 = d \vee \neg se \vee s$$

$$\alpha_1 = \mathrm{AX}(\neg d \wedge \neg se \wedge s) \qquad \alpha_2 = \mathrm{AX}((\neg d \wedge se \wedge \neg s) \vee (\neg d \wedge \neg se \wedge \neg s))$$

$$\alpha_3 = \mathrm{EX}(\neg d \wedge se \wedge \neg s) \wedge \mathrm{EX}(\neg d \wedge \neg se \wedge \neg s) \qquad \alpha_4 = \mathrm{AX}(d \wedge \neg se \wedge \neg s)$$

$$\alpha_5 = \mathrm{AX}(d \wedge \neg se \wedge \neg s) \qquad \psi = d \wedge \neg se \wedge \neg s.$$

则，

$$\begin{aligned}
&\psi \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \equiv (\psi \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4) \vee (\psi \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\psi \wedge \psi_1 \wedge \psi_2 \wedge \alpha_4 \wedge \psi_4) \vee (\psi \wedge \psi_1 \wedge \psi_2 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\psi \wedge \psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \psi_4) \vee (\psi \wedge \psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \alpha_4) \vee \\
&(\psi \wedge \psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \psi_4) \vee (\psi \wedge \psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\psi \wedge \alpha_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4) \vee (\psi \wedge \alpha_1 \wedge \psi_2 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\psi \wedge \alpha_1 \wedge \psi_2 \wedge \alpha_4 \wedge \psi_4) \vee (\psi \wedge \alpha_1 \wedge \psi_2 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\psi \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \psi_4) \vee (\psi \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\psi \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \psi_4) \vee (\psi \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5).
\end{aligned}$$

$$\begin{aligned}
&\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \equiv (\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4) \vee (\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\psi_1 \wedge \psi_2 \wedge \alpha_4 \wedge \psi_4) \vee (\psi_1 \wedge \psi_2 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \psi_4) \vee (\psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \alpha_4) \vee \\
&(\psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \psi_4) \vee (\psi_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\alpha_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4) \vee (\alpha_1 \wedge \psi_2 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\alpha_1 \wedge \psi_2 \wedge \alpha_4 \wedge \psi_4) \vee (\alpha_1 \wedge \psi_2 \wedge \alpha_4 \wedge \alpha_5) \vee \\
&(\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \psi_4) \vee (\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \psi_3 \wedge \alpha_5) \vee \\
&(\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \psi_4) \vee (\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5).
\end{aligned}$$

所以，

$$
\begin{aligned}
&\neg \mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{V\cup\{s\}}(\mathscr{K})\wedge\neg s,\{s\})\\
&\equiv \neg \mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{V\cup\{s\}}(\mathscr{K}),\{s\})\\
&\equiv \neg \mathrm{F}_{\mathrm{CTL}}(\psi\wedge \mathrm{AG}(\varphi_1\wedge\varphi_2\wedge\varphi_3\wedge\varphi_4),\{s\})\\
&\equiv \neg(\mathrm{F}_{\mathrm{CTL}}(\psi\wedge\varphi_1\wedge\varphi_2\wedge\varphi_3\wedge\varphi_4,\{s\})\wedge \mathrm{AG}\mathrm{F}_{\mathrm{CTL}}(\varphi_1\wedge\varphi_2\wedge\varphi_3\wedge\varphi_4,\{s\}))\\
&\equiv \neg\Big(\Big(d\vee\big((d\vee se)\wedge\mathrm{AX}(d\wedge\neg se)\big)\vee\big((d\vee se\vee\mathrm{AX}\neg d)\wedge\mathrm{EX}(\neg d\wedge se)\wedge\mathrm{EX}(\neg d\wedge\neg se)\big)\\
&\vee(d\wedge\mathrm{AX}(\neg d\wedge\neg se))\vee\big((d\vee\neg se)\wedge\mathrm{EX}\neg d\big)\Big)\wedge\\
&\mathrm{AG}\Big(\big((d\vee se)\wedge\mathrm{AX}(d\wedge\neg se)\big)\vee\big(\mathrm{AX}\neg d\wedge\mathrm{EX}(\neg d\wedge se)\wedge\\
&\mathrm{EX}(\neg d\wedge\neg se)\big)\vee\big((d\vee se)\wedge\mathrm{AX}(\neg d\wedge\neg se)\big)\vee\mathrm{EX}\neg d\Big)\Big).
\end{aligned}
$$

5.1.2 遗忘封闭性及复杂性

当CTL公式的长度（符号的个数）为n时，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结构能保证公式的可满足性[96]。对于其它拥有同样大小状态空间上的任意初始结构，都能在状态空间$\mathscr{S}$上找到一个初始结构与之互模拟，且由定理5.1可知它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。 □

该结论表明：任意CTL公式都与其模型的特征公式的析取逻辑等价。这对遗忘理论的封闭性提供了重要的理论支撑，即：从公式里遗忘原子命题集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V，从φ中遗忘V中的元素得到的结果为：

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

下面分析遗忘在CTL段$\mathrm{CTL}_{\mathrm{AF}}$下推理问题的复杂性结果，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[24]。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此类公式遗忘结果的模型是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$V\subseteq\mathscr{A}$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[33]；条件(b)也可用Baier等人的推论7.45[24]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[49]。因此，上述问题是NP-难的。 □

下面讨论$\mathrm{CTL}_{\mathrm{AF}}$段下关于遗忘的逻辑蕴涵问题的复杂性。下文用记号$\alpha\models^?\beta$表示“$\alpha$是否逻辑蕴涵$\beta$”。

定理 5.2 (Entailment)**.** *令φ和ψ为CTL_{AF}中的两个公式，V为原子命题集。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) co-NP-难. 已有结论表明，判定公式φ是否是可满足的是NP-完全的[124]。令$\psi\equiv\bot$，只需证明φ是不可满足的，该问题是co-NP-难的。所以，原问题是co-NP-难的。

在co-NP中. 定理3.4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[124]。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$(\mathscr{M}',s_0')\not\models\psi$；(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) 在Π_2^{P}中. 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$(\mathscr{M},s_0)\not\models \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$是否成立。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。

Π_2^{P}-难. 令$\psi\equiv\top$。则这一问题被归约为判定$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$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文献[119]。

(iii) 在Π_2^{P}中. 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$(\mathscr{M}_1,s_1)\models\varphi$但是对任意$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}_2,s_2)$，$(\mathscr{M}_2,s_2)\not\models\psi$。由命题5.3的证明可知，$(\mathscr{M},s)$和$(\mathscr{M}_1,s_1)$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$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}_2,s_2)$，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。

Π_2^{P}-难. 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原子公式集。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。*

5.1.3 基于模型的遗忘算法

这里提出基于模型的遗忘算法，该算法与第5.2.1.3节中基于语法的算法不同。本小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。

这里给出如何计算从规范中遗忘一些原子命题。

例 5.3. 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$sp$，即从$\alpha$中遗忘$sp$，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。

如下所述，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。

算法 5.1 基于模型的CTL遗忘过程

Input: CTL公式φ和原子命题集V
Output: $\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$
$\psi \leftarrow \bot$ **foreach** $\mathscr{A}$ 和$\mathscr{S}$上的初始结构$\mathscr{K}$ **do**
  if $\mathscr{K} \not\models \varphi$ **then continue**
  foreach 满足$\mathscr{K} \leftrightarrow_V \mathscr{K}'$的初始结构$\mathscr{K}'$ **do**
    $\psi \leftarrow \psi \vee \mathscr{F}_{\overline{V}}(\mathscr{K}')$
  end
end
return ψ

命题 5.4. 令φ为*CTL*公式，$V \subseteq \mathscr{A}$为原子命题集，状态空间大小为$|\mathscr{S}| = m$，$|\mathscr{A}| = n$，$|V| = x$。使用算法*5.1*计算从φ中遗忘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$B \subseteq \mathscr{S}$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$B$，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），需要$N(i) = P_i(s_0) + i \times (P_i(s) + i \times P_i(s'))$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

$$
P_i(y) = k + i \times k + i^2 \times k + \cdots + i^i \times k = \frac{i^i - 1}{i - 1} k,
$$

$$\begin{aligned} N(i) &= P_i(s_0) + i \times (P_i(s) + i \times P_i(s')) \\ &= (i^2 + i + 1)P_i(y) \\ &= (i^2 + i + 1)\frac{i^i - 1}{i - 1}k. \end{aligned}$$

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$O((n-x)m^{2(m+2)}2^{nm} \times \log m)$。 □

5.2 基于归结的遗忘计算方法

已有结果显示，任意CTL公式可以转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集。归结是一种以子句为计算对象的判断可满足性的方法。本节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$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集（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5.2所示。CTL遗忘结果不总是存在的，因此，基于归结的方法在有的情况下不能计算出遗忘结果。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。

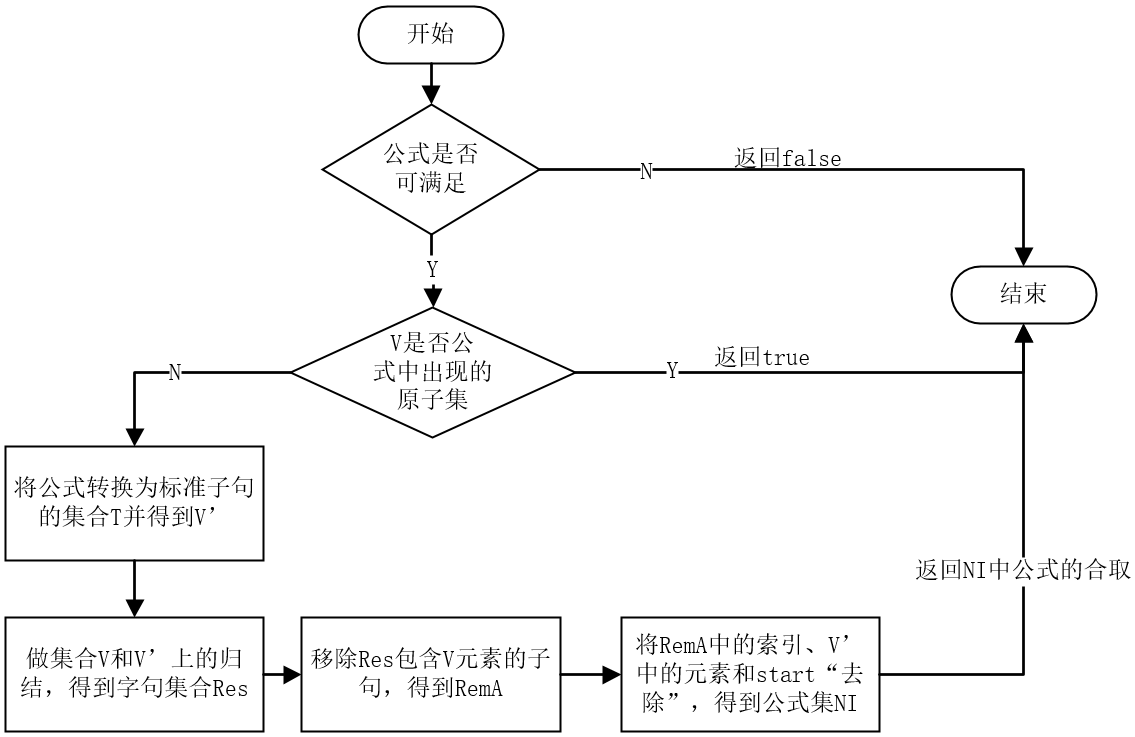


图 5.2: 基于归结的遗忘的主要流程图

本章展示如何使用表2.3中的归结规则来计算CTL遗忘。为了计算从CTL公式φ中

遗忘集合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？如在第二章中所说，将一个CTL公式转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集会引入新的原子命题和索引。虽然已有的研究说明，CTL公式可以转换为带索引的公式集并保证其可满足性，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。

本章其余部分组织如下：首先，在第5.2.1节详细介绍CTL-forget各个过程及算法复杂性；其次，在第5.2.2节介绍基于Prolog的CTL-算法的实现；最后，在第5.2.3节给出并分析实验结果。

5.2.1　基于归结的算法CTL-forget

通过上述分析，本节讲述如何使用$R^{\succ,S}_{\mathrm{CTL}}$归结系统对要遗忘的原子命题做归结，然后如何将归结后得到的结果等价转换为CTL公式。

5.2.1.1　CTL归结UF

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中的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包括：需要遗忘的原子命题和转换过程中引入的新原子命题。

令T为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集，$p$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：

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

对于原子命题集V，定义$\mathrm{UF}(T,\emptyset) = T$且$\mathrm{UF}(T,\{p\} \cup V) = \mathrm{UF}(\mathrm{UF}(T,p),V)$。直观上，$T$在$p$上的展开是对$p$做穷尽的归结，即：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CTL公式φ，从$\mathrm{UF}(T_\varphi,V)$ 中移除含有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与T_φ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 5.5. 令φ为一个*CTL*公式，$V \subseteq \mathscr{A}$为原子命题集。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)** $\mathrm{UF}(T_\varphi, V) \equiv_U ERes(\varphi, V)$。为此，定义如下由$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集构成的序列：$T_0 = T_\varphi, T_1, T_2, \ldots, T_n = \mathrm{UF}(T_\varphi, V)$，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$T_\varphi \equiv_U \mathrm{UF}(T_\varphi, V)$，只需证明对任意$0 \leq i < n$，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) 下证若$\Pi \rightarrow_r R_i$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$\neg p$）。

已有结果表明$\Pi \models R_i$[125]，因此，$T_{i+1} = T_i \cup \Lambda^{\mathrm{A}}_{\neg l}$，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$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集[126]。显然，对所有$(\mathscr{M}_1, s_1) \in Mod(T_i = X \cup \Pi)$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$(\mathscr{M}_2, s_2) \in Mod(T_{i+1} = T_i \cup \Lambda^{\mathrm{A}}_{\neg l})$，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$0 \leq i < n$，有$T_i \equiv_U T_{i+1}$。又因为$\equiv_U$为等价关系，所以$T_\varphi \equiv_U \mathrm{UF}(T_\varphi, V)$。

(F2) 假设$V = \{p\}$，C_i $(i = 1, 2)$为经典子句，$l = p$或$l = \neg p$。显然$\mathrm{UF}(T_\varphi, V) \models ERes(\varphi, V)$。下证对任意$\mathscr{K} = (\mathscr{M}, s) \in Mod(ERes(\varphi, V))$（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，从以下几点证明上述结论成立。

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

(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$Pt \in \{\mathrm{A}, \mathrm{E}\}$。

若对任意其它子句C'，$p \notin Var(C')$，则对任意$\mathscr{K} = (\mathscr{M}, s) \in Mod(ERes(\varphi, V))$，如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S, R, L', [_], s)$（即$s' = s$），其中$L'$与$L$一样，除了对任意$s_1 \in S$，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，$(\mathscr{M}, s) \leftrightarrow_{\{p\}} (\mathscr{M}', s')$且$(\mathscr{M}', s') \models C' \wedge C$。

若$C' = Q \rightarrow Pt\mathrm{F}\neg l$，不失一般性地，令$l = p$。对任意$\mathscr{K} = (\mathscr{M}, s) \in Mod(ERes(\varphi, V))$，如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S, R, L', [_], s)$，其中$L'$与$L$相同，除了

(1) $L'(s)=L(s)\cup\{p\}$；

(2) 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；

(3) 对任意其它的非初始状态$s''\in S$，$L'(s'')=(L(s'')-\{Q\})\cup\{p\}$ （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$C'=Q\to Pt\mathrm{X}(C_2\vee\neg l)$ （令$Pt=\mathrm{A}$，$Pt=\mathrm{E}$可类似地证明），则有$Q\to\mathrm{AX}(C_1\vee C_2)\in\mathrm{UF}(T_\varphi,V)$。因此，对任意$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，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$s_1\in S$，若$(\mathscr{M},s_1)\not\models Q$，则对任意$(s_1,s_2)\in R$，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$L'(s_2)=L(s_2)-\{p\}$”。显然，$\mathscr{K}\leftrightarrow_{\{p\}}\mathscr{K}'$ 且$\mathscr{K}'\models C'\wedge C$。

(ii) 若$C'=Q\to Pt\mathrm{F}\neg l$。不失一般性地，假设$l=p$。$C$和$C'$在$p$上是可归结的，则一定存在子句集$\{P_1^1\to *C_1^1,\ldots,P_{m_1}^1\to *C_{m_1}^1,P_1^n\to *C_1^n,\ldots,P_{m_n}^1\to *C_{m_n}^1\}$，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（$\neg C_1\to l$为子句集中的一个），使得$\bigvee_{i=1}^n\bigwedge_{j=1}^{m_i}P_j^i\to\mathrm{EXEG}l$。因此，通过使用规则ERES1 （ERES2规则类似）可以得到一个子句$C''=\top\to\neg Q\vee\neg p\vee C_1$。所以，在使用规则SRES8在$C$和$C''$上后得到子句$\top\to\neg Q\vee C_1$。在这种情况下，对任意$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，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$s_1\in S$，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) 可以类似地证明其它类型的子句，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$Q\to\mathrm{AX}(C_1\vee l)$。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$C'=Q_1\to Pt\mathrm{X}(C_2\vee\neg l)$ （令$Pt=\mathrm{E}_{ind}$，$Pt=\mathrm{A}$的情形可以类似地证明），则$Q\wedge Q_1\to\mathrm{E}_{ind}\mathrm{X}(C_1\vee C_2)\in\mathrm{UF}(T_\varphi,V)$。所以，对任意$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，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$s_1\in S$

(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$(s_1,s_2)\in R$，（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) 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$L'(s'_2)=L(s'_2)$。对其它满足$(s_1,s_2)\in R$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$L'(s'_2)=L(s'_2)$。

显然，$\mathscr{K}\leftrightarrow_{\{p\}}\mathscr{K}'$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$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集$\{P_1^1\rightarrow *C_1^1,\ldots,P_{m_1}^1\rightarrow *C_{m_1}^1,P_1^n\rightarrow *C_1^n,\ldots,P_{m_n}^1\rightarrow *C_{m_n}^1\}$，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，对任意$\mathscr{K}=(\mathscr{M},s)\in Mod(ERes(\varphi,V))$，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$(\mathscr{M}',s')\models C'\wedge C$。

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

例 5.4 (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** $\rightarrow r$	$(1,2,$**SRES5**$)$	(2) **start** $\rightarrow x\vee y$	$(1,4,$**SRES5**$)$
(3) $\top\rightarrow\neg z\vee y\vee f\vee m$	$(3,4,$**SRES8**$)$	(4) $y\rightarrow\mathrm{AX}(f\vee m\vee y)$	$(3,8,$**SRES6**$)$
(5) $\top\rightarrow\neg z\vee x\vee p$	$(4,5,$**SRES8**$)$	(6) $\top\rightarrow\neg z\vee x\vee q$	$(4,6,$**SRES8**$)$
(7) $y\rightarrow\mathrm{AX}(x\vee p)$	$(5,8,$**SRES6**$)$	(8) $y\rightarrow\mathrm{AX}(x\vee q)$	$(6,8,$**SRES6**$)$
(9) **start** $\rightarrow f\vee m\vee y$	$(3,(2),$**SRES5**$)$	(10) **start** $\rightarrow x\vee p$	$(5,(2),$**SRES5**$)$
(11) **start** $\rightarrow x\vee q$	$(6,(2),$**SRES5**$)$	(12) $\top\rightarrow p\vee\neg z\vee f\vee m$	$(5,(3),$**SRES8**$)$
(13) $\top\rightarrow q\vee\neg z\vee f\vee m$	$(6,(3),$**SRES8**$)$	(14) $y\rightarrow\mathrm{AX}(p\vee f\vee m)$	$(5,(4),$**SRES6**$)$
(15) $y\rightarrow\mathrm{AX}(q\vee f\vee m)$	$(6,(4),$**SRES6**$)$	(16) **start** $\rightarrow f\vee m\vee p$	$(5,(9),$**SRES5**$)$
(17) **start** $\rightarrow f\vee m\vee q$	$(6,(9),$**SRES5**$)$		

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

start $\rightarrow z$, **start** $\rightarrow f\vee m\vee q$, **start** $\rightarrow x\vee y$, **start** $\rightarrow q\vee x$, **start** $\rightarrow f\vee m\vee y$,
$\top\rightarrow f\vee m\vee\neg x$, $\top\rightarrow q\vee f\vee m\vee\neg z$, $\top\rightarrow f\vee m\vee\neg z\vee y$,

$$\top \to q \vee x \vee \neg z, \quad \top \to x \vee y \vee \neg z, \quad \top \to q \vee \neg y, \quad z \to \mathrm{AF}x,$$
$$y \to \mathrm{AX}(q \vee f \vee m), \quad y \to \mathrm{AX}(x \vee q), \quad y \to \mathrm{AX}(x \vee y), \quad y \to \mathrm{AX}(f \vee m \vee y).$$

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

5.2.1.2　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，在转换和归结过程中都会引入索引和新的原子命题。本节介绍如何移除这些新引入的元素。

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

引理 5.6. *如果$j \in \mathscr{I}$，ψ_i, φ_i $(1 \le i \le n)$为CTL公式，那么：*

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

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

(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$(\mathscr{M}, s_0) \models \bigwedge_{i \in S} \psi_i$（$S \subseteq \{1, \ldots, n\}$），则存在$s_0$的下一个状态$s_1$，使得$(s_0, s_1) \in [j]$和$(\mathscr{M}, s_1) \models \bigwedge_{i \in S} \varphi_i$。由$[j]$的定义可知$(s_0, s_1) \in R$，因此$(\mathscr{M}, s_0) \models (\bigwedge_{i \in S} \psi_i) \to \mathrm{E}_{\langle j \rangle}\mathrm{X}(\bigwedge_{i \in S} \varphi_i)$。

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

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

($\Leftarrow$) 对等式右边公式的任意模型$(\mathscr{M}, s_0)$，若$(\mathscr{M}, s_0) \models \bigwedge_{i \in S} \psi_i$（$S \subseteq \{1, \ldots, n\}$），则存在$s_0$的下一个状态$s_1$，使得$(\mathscr{M}, s_1) \models \bigwedge_{i \in S} \varphi_i$。容易构造一个初始Ind-结构$(\mathscr{M}', s_0)$（$\mathscr{M}' = (S, R, L, [_]', s_0)$），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$(\mathscr{M}', s_0) \models \{\psi_i \to \mathrm{E}_{\langle j \rangle}\mathrm{X}\varphi_i \mid 1 \le i \le n\}$。

(iii)的证明与(ii)的证明类似。 □

本文将引理5.6中(i)、(ii)、(iii)等号$\equiv_*$（$* \in \{$空字符串，$\emptyset\}$）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$\alpha_i = \psi_i \to \mathrm{E}_{\langle j \rangle}\mathrm{X}\varphi_i$ $(1 \le i \le n)$和$\beta_1 = \psi_1 \to \mathrm{E}_{\langle j \rangle}\mathrm{F}\varphi_1$。

算法 5.2 RM-index(Σ)

Input: 有限$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集$\Sigma$
Output: CTL公式集
foreach Σ中拥有相同索引$\langle i\rangle$的E-子句构成的极大子集Δ **do**
  if 存在索引为$\langle i\rangle$的E-某时子句$\alpha\in\Sigma$ **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

因为$\mathrm{EX}\varphi_1\wedge\mathrm{EX}\varphi_2\not\models\mathrm{EX}(\varphi_1\wedge\varphi_2)$，引理5.6(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$\psi_i$ $(1\leq i\leq n)$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5.2实现，其目的是消除索引。下面的推论表明，当移除索引之后Σ和RM-index(Σ)是互模拟等价的。

推论 5.3. *如果φ为一个CTL公式、$U=Var(T_\varphi)-Var(\varphi)$，$V\subseteq\mathscr{A}$为原子命题集、$\Sigma=\mathrm{ERes}(\mathrm{UF}(\varphi,V\cup U),V)$，那么RM-index($\Sigma$)$\equiv_\emptyset\Sigma$。*

通过下面的定理，可以移除一些新引入的原子命题而保持互模拟等价。

引理 5.7 (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公式集（$n,m\geq 1$）、$\Gamma$为$x$正出现在其中的有限个CTL公式集。下面式子成立：*

$$\Gamma\cup\Delta\equiv_{\{x\}}\Gamma\left[x/\bigwedge(\{C_i\mid 1\leq i\leq n\}\cup\{B_i\mid 1\leq i\leq m\})\right]. \tag{5.1}$$

证明. 令$\varphi=\bigwedge(\{C_i\mid 1\leq i\leq n\}\cup\{B_i\mid 1\leq i\leq m\})$。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$(\mathscr{M},s_0)\models x$，显然$(\mathscr{M},s_0)\models\varphi$成立。

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

(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$(\mathscr{M},s_1)\models\bigwedge\Delta$
$\Rightarrow$ 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$(s_0,s_1)\in R$，有$(\mathscr{M},s_1)\models\psi_1(x)$和$(\mathscr{M},s_1)\models\bigwedge\Delta$
$\Rightarrow$ 对任意$(s_0,s_1)\in R$，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$\pi=(s_0,s_1,\dots)\in R$，使得对某个$j\geq 0$，有$(\mathscr{M},s_j)\models\psi_2(x)$，对任意$0\leq i<j$，有$(\mathscr{M},s_i)\models\psi_1(x)$，且对所有$x\geq 0$，有$(\mathscr{M},s_x)\models\bigwedge\Delta$
$\Rightarrow$ 由归纳假设可知，存在一条路径$\pi=(s_0,s_1,\dots)\in R$，使得对某个$j\geq 0$，有$(\mathscr{M},s_j)\models\psi_2(x)[x/\varphi]$，且对任意$0\leq i<j$，有$(\mathscr{M},s_i)\models\psi_1(x)[x/\varphi]$
$\Rightarrow (\mathscr{M},s_0)\models \mathrm{E}((\psi_1(x)[x/\varphi])\,\mathrm{U}\,(\psi_2(x)[x/\varphi]))$
$\Rightarrow (\mathscr{M},s_0)\models (\mathrm{E}(\psi_1(x)\mathrm{U}\psi_2(x)))[x/\varphi]$。

可以类似证明其它情况。

($\Leftarrow$) 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$(\mathscr{M}',s_0)\models x\leftrightarrow\varphi$）。

显然易证，$(\mathscr{M},s_0)\leftrightarrow_{\{x\}}(\mathscr{M}',s_0')$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公式集$\Sigma$，用$\mathrm{GAL}(\Sigma,\{x\})$表示$\mathrm{GAL}(\Gamma\cup\Delta,\{x\})$，其中$\Delta\subseteq\Sigma$为与引理5.7中$\Delta$有相同性质的出现在$\Sigma$中有唯一负出现$x$的公式集，$\Gamma\subseteq\Sigma$是$x$正出现在其中的公式集。对于原子命题集$V$，定义

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

例 5.5 (例5.4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：

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

第二步考虑原子命题z、$\Delta'=\{\top\rightarrow q\vee f\vee m\vee\neg z,\top\rightarrow f\vee m\vee y\vee\neg z,z\rightarrow \mathrm{AF}(f\vee m)\}$

算法 5.3 CTL-forget(φ, V)

Input: CTL公式φ和原子命题集V
Output: 公式集
if $\varphi \equiv \bot$ **then return** $\bot$; // 若公式不可满足，则遗忘结果为$\bot$
if $V = Var(\varphi)$ **then return** $\top$; // 若遗忘所有原子命题，则结果为$\top$
$T_\varphi \leftarrow \mathrm{SNF}^g_{\mathrm{CTL}}(\varphi)$; // 将φ转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句
$\Sigma \leftarrow \mathrm{UF}(T_\varphi, V \cup U)$，其中$U = Var(T_\varphi) - Var(\varphi)$; // 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
用φ替换Σ中的初始子句“$\mathrm{AG}(\mathbf{start} \rightarrow \varphi)$” ; // 去除**start**
return Σ

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：

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

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。

5.2.1.3 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5.3。该算法的输入为一个CTL公式φ和一个原子命题集，输出为一个与φ互模拟等价的CTL公式。

定理 5.3. *若φ为一个CTL公式、$V \subseteq \mathscr{A}$、Σ =CTL-forget(φ, V)且$U = Var(\Sigma) - Var(\varphi)$，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5.5，引理5.6和5.7。

(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$\Sigma \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$。 □

由此可知，如果引入的新原子命题能全部移除，那么得到的结果即为遗忘结果。

例 5.6 (例5.5的延续). 容易看出CTL-forget$(\varphi, \{p, r\})$包含下面的公式

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

$$\mathrm{AG}(y \to \mathrm{AX}(q \vee f \vee m)), \quad \mathrm{AG}(y \to \mathrm{AX}(f \vee m \vee y)).$$

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 5.6. *给定CTL公式φ，若φ满足下面约束：（1）φ中不包括操作符$Pt\mathscr{T}$（其中$Pt \in \{\mathrm{A},\mathrm{E}\}$且$\mathscr{T} \in \{\mathrm{U},\mathrm{G}\}$）；（2）对于任意原子命题$p \in V$，若$p$和$\neg p$出现在同一时序算子的范围内。那么，*CTL-forget$(\varphi, V) \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$。

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$\Pi = \{\top \to \neg x \vee p \vee \psi_1, \top \to \neg x \vee \neg p \vee \psi_2\}$的子句集$\Sigma$，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，$\Sigma$中不包含其它含有$p$的公式。

由归结过程可产生子句$\top \to \neg x \vee \psi_1 \vee \psi_2$，由定理5.7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$\to$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5.7替换。因此，由定理5.3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[99]。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5.3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 5.7. *给定CTL公式φ和原子命题集$V \subseteq \mathscr{A}$。算法5.3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5.3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。

5.2.2 基于Prolog的CTL-forget算法实现

基于Prolog的CTL-forget算法实现系统以CTL公式和原子命题集为输入，CTL公式为输出。其所识别的CTL公式的符号与第二章中CTL的语言符号对应关系如下：

- x_i和其余小写字母开头的字符串构成原子命题集，其中$i \geq 0$为自然数，且x_i和z被设定为只能是在如下描述的转换过程中引入的原子命题；
- “$false$”和“$true$”分别与常量符号“⊥”和“⊤”对应；
- “$start$”与命题常量“**start**”对应；
- “&”、“\/”、“−”和“−>”分别与联结符号“∧”、“∨”、“¬”和“→”对应；
- “~”和“^”分别与路径量词“A”和“E”对应；
- “@”、“∗”、“?”和“$”分别与时序操作符“G”、“X”、“F”和“U”对应。

例 5.7. 字符串(~∗((−$y1$\/ − $y2$\/ − $y4$)&(−$y1$\/$y2$\/$y4$)&($y1$\/$y2$\/ − $y3$)&($y1$\/$y3$\/ − $y4$)&(−$y1$\/$y2$\/ − $y3$)))为CTL 公式。

此系统主要包括五个模块：转换模块（transCTL2SNF/6）、归结模块（两个过程：step_resolution/3和temp_resolution/8）、“移除”原子命题模块（removeAtom/3），“移除”索引（pro6/3）和“移除”新引入的原子命题（ackerM/3）。下面就这几个模块做详细的介绍。

转换模块：用二元谓词transCTL2SNF/6来实现将CTL公式转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的过程。在使用该谓词前，谓词ctl2NNF/2将输入的CTL公式φ转换为其NNF形式并使用表2.2中的等式化简公式得到公式$simp(nnf(\varphi))$，并产生如下公式列表（list）：

$$[start -> z, z -> simp(nnf(\varphi))].$$

transCTL2SNF/6谓词通过迭代表 2.1中的每条规则将CTL公式转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集，每一个子句用一个四元谓词snf_clause(Parm1,Parm2,Parm3,Parm4)表示，其中：

- Parm1表示子句类型，即第 2.3.2节中的六种子句类型，用“start”、“true”、“ef”、“af”、“ex”和“ax”分别表示初始子句、全局子句、E-某时子句、A-某时子句、E-步子句和A-步子句；
- Parm2为原子命题构成的列表，表示$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的头；
- Parm3为原子命题构成的列表，表示$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的尾；
- Parm4表示索引号，且如果Parm4=nil，则该子句不是带索引的子句。

在transCTL2SNF(Parm1,Parm2,Parm3,Parm4,Parm5,Parm6)中：参数Parm1为公式形成的列表，即：$[start -> z, z -> simp(nnf(\varphi))]$；Parm2和Parm3初始化为0，分别表示

索引号和新引入的原子命题下标；Parm4和Parm5分别表示转换过程结束后返回的索引号和原子命题下标的最大值；Parm6为经过转换Parm1后得到的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集。

归结模块：主要有两个过程step_resolution/3和temp_resolution/8。step_resolution(List1, V, List2)从List1中选择能使用表 2.3中的归结规则对V中的元素做归结的公式，并将得到结果存在列表List2中。这一过程循环执行，直到不再产生新的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句为止。temp_resolution/8过程主要用于寻找归结规则**ERES1**和**ERES2**的前提（一组$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集，参照第 2.4节）。

“移除”原子命题模块：主要用于将含有要遗忘的原子命题的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句去掉，这一过程比较简单，这里不详细介绍。

*“移除”索引模块*过程使用引理 5.6中的互模拟等价公式将$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句中的索引去掉，因为CTL公式中不包含索引，所以这一过程是必须的。

*“移除”新引入的原子命题*使用一般化的Ackermann引理（引理 5.7）将转换模块中引入的新原子命题去除。

5.2.3　实验

本节给出上一小节中prolog实现的CTL-forget计算系统的实验结果，并分析实验结果。本实验在Linux服务器上进行，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在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。所有实验数据和实验结果都可以从网上获取[3]。

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，$|V|$为原子命题集V中的原子个数。

5.2.3.1　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，令一组是随机生成的数据。标准数据集来源于CTL-RP [4]。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$\bot$。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5.1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。

[3] https://github.com/fengrenyan/forgetting-in-CTL/tree/main/Appendix
[4] https://sourceforge.net/projects/ctlrp/

表 5.1: 计算CTL-forget(φ, V)所使用的CPU时间（单位：秒(s)）

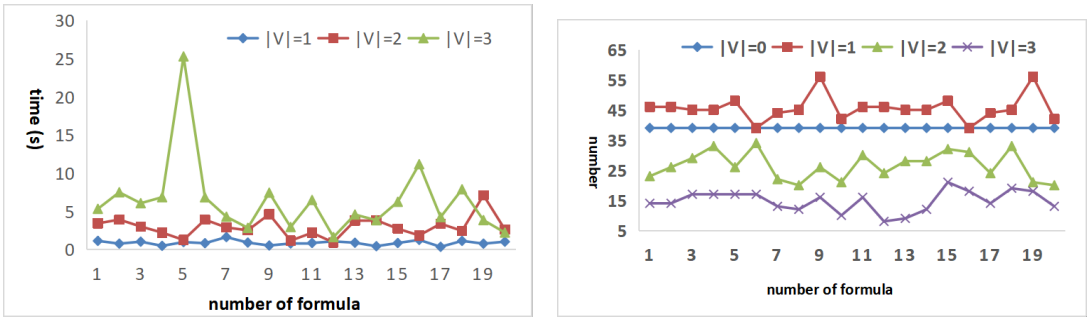
φ \ $\lvert V\rvert$	1	2	3	4
s001	0.0505	0.1053	0.2259	0.3680
s002	0.3645	1.0416	5.6372	10.0184
s003	97.5341	71.5396	190.1157	423.5793
s004	77.5086	77.4246	101.1284	118.7461
s001-3	681.2883	613.1859	1617.047	2356.949

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$\mathscr{A}$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5.3所示，图5.3(a) 展示了计算遗忘所需要的时间，图5.3(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5.3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5.4所示，其与$|\varphi_i| = 12$时有相似的结果。

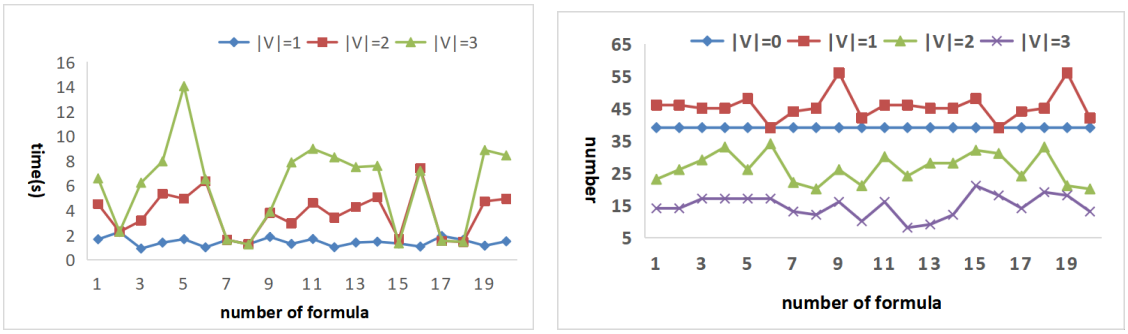


(a) 计算遗忘需要的CUP时间　　(b) $\mathrm{SNF}^g_{\mathrm{CTL}}$子句的个数

图 5.3: 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$\varphi_i = 12$。

5.2.3.2 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：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

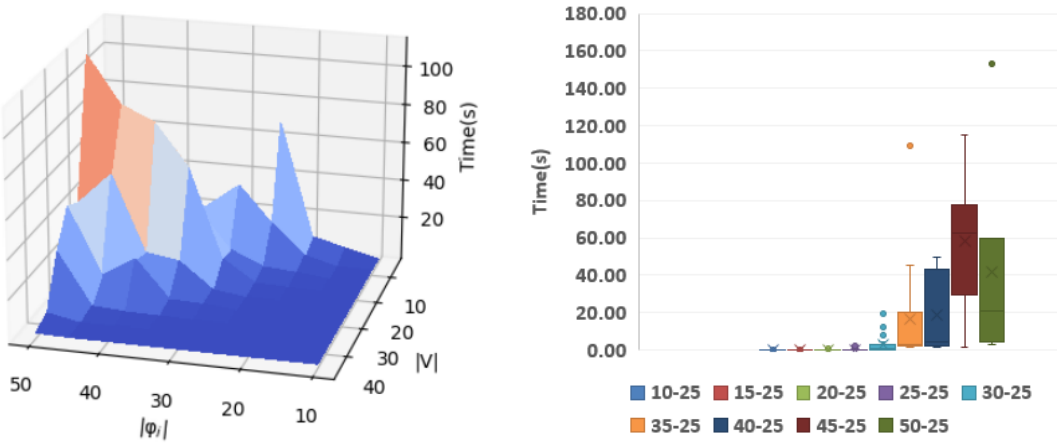


(a) 计算遗忘需要的CUP时间　　(b) $\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的个数

图 5.4: 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$\varphi_i = 16$。

子命题集$\mathscr{A}$上的公式、V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5.5所示。



(a) 平均CPU时间(s)　　(b) $|V| = 25$时所使用CPU时间箱线图

图 5.5: 计算3-CNF公式SNC的CPU时间

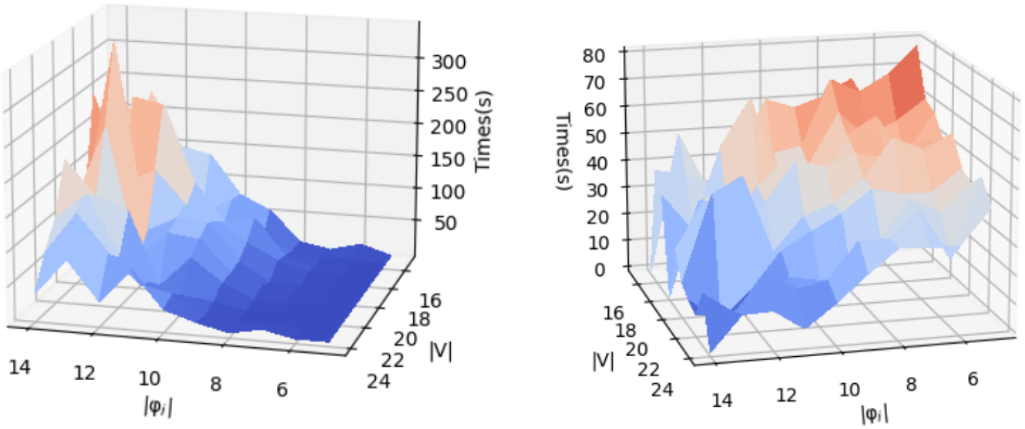
从图5.5(a)可看出，随着$|\varphi|$增大或$|V|$的减小时间消耗越大。直观上，若$|\varphi|$越大或者$|V|$越小，则$\mathrm{F}_{\mathrm{CTL}}(\varphi, \overline{V})$越难计算。这与上一小节中的结论相符合。图5.5(b)展示了当$|V| = 25$、$nc \in \{10, 15, \ldots, 45, 50\}$时20个随机实例的箱线图。这同样证明了$nc$越大SNC越难计算。

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5.6(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，若$|\varphi|$越大或者$|V|$越小，则$\mathrm{F}_{\mathrm{CTL}}(\varphi,\overline{V})$越难计算。此外，图5.6(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 5.6: 计算CTLSNC的平均时间和存在SNC的公式占比.

综上所述，算法5.3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时计算效率较高。

5.3 本章小结

本章第 5.1节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析取，因而可以计算遗忘的结果。最后，给出了基于模型的计算遗忘算法，并分析该算法关于公式的大小是指数空间的。

本章第 5.2节探索了如何使用Zhang等人提出的归结系统计算CTL遗忘。为此，首先，使用Zhang提出的转换规则将CTL公式转换成$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集，并使用归结规则在要遗忘的原子命题集上做穷尽的归结；然后，给出几个互模拟等价的等式消除索引，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，给出计算遗忘的算法，并分析该算法的复杂性，分析表明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个数是指数时间的。

随后，第 5.2.2节给出了基于Proplog的算法CTL-forget实现介绍。第 5.2.3节从两个角度来评估算法CTL-forget的系统实现。实验结果表明，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个数和公式长度都很敏感：要遗忘的原子命题越多或公式越长，计算效率越低。此外，关于计算SNC的实验表明，算法CTL-forget在用于实验的数据集中，大部分情况是能计算出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。

第六章　总结与展望

本章首先总结本文的研究工作，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。

6.1　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发系统的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。

本文以使用遗忘计算反应式系统在给定条件下的WSC（SNC）和知识更新为主线，解决了第一章提出的问题，并取得了以下成果：

(1) 本文通过互模拟定义了CTL遗忘，并给出CTL遗忘具有分解性、切片性和同质性等基本属性。此外，提出了一种基于归结的算法计算CTL遗忘。为此，提出一般化的ACkermann引理和消除索引的方法，用于将归结得到的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句转换为CTL公式。最后，使用Prolog实现了该算法，并做了相关实验评估了该系统。

特别地，本文证明了CTL遗忘不总是存在的。因此，探索了约束CTL遗忘。在这种情形中，构成CTL公式原子命题是有限的，公式长度为n且公式模型是有限的。提出了一种有限互模拟，并证明这种有限互模拟与上述的互模拟在有限结构下是等价的。本文证明，有限结构可以用其特征公式——CTL公式表达。从而证明任意公式的遗忘是存在的，即：与该公式的模型互模拟的所有结构特征公式的吸取。

(2) μ-演算是一种表达能力较强的逻辑语言。本文给出了μ-演算遗忘的定义，并证明了其满足基本的遗忘性质。为此，本文提出了一种新的互模拟，证明了μ-公式对这一互模拟是不变的，即：若两个结构是$\langle \mathcal{V}_1, V\rangle$-互模拟的（$\mathcal{V}_1$是变元集，$V$是原子命题集），则对于任意与$V \cup \mathcal{V}_1$无关的公式，这两个结构同时满足（或不满足）该公式。

此外，本文探索了两个遗忘存在的子类：μ-句子和x-类。

(3) 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，若遗忘除这些原子命题之外的原子命题结果可用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结果总是存在的。此外，当给定的反应式系统为有限系统的情形下，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间偏序关系的定

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。

(4) 实现了基于归结的算法，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。

6.2　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中遗忘封闭且计算简单的子类。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更新系统，并得到新系统$\mathscr{M}'$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，但是更多关于遗忘问题的复杂性应该被探索，如：整个CTL下的遗忘的模型检测和蕴涵问题的复杂性。

(4) 研究算法 5.3的完备性。尽管在第五给出了算法正确性分析，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 5.3的完备性，并探索其他可靠且完备的算法。

(5) μ-演算遗忘变元的研究。μ-公式中含有约束变元和自由变元，如何遗忘变元将成为下一步研究工作。

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] 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.

[21] 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.

[22] 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.

[23] 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.

[24] BAIER C, KATOEN J. Principles of model checking[M]. [S.l.]: MIT Press, 2008.

[25] LEGATO W J. A weakest precondition model for assembly language programs[M]. [S.l.]: April, 2002.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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 Cur-

rent Trends in Concurrency, Overviews and Tutorials. Springer, 1986: 510-584. https://doi.org/10.1007/BFb0027047.

[32] 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):1-14.

[33] CLARKE E M, GRUMBERG O, PELED D A. Model checking[M/OL]. MIT Press, 2001. http://books.google.de/books?id=Nmc4wEaLXFEC.

[34] CLARKE E, GRUMBERG O, LONG D. Model checking[C]//NATO ASI DPD. [S.l.: s.n.], 1996: 305-349.

[35] 王政. 嵌入式周期控制系统的建模与分析[D]. 上海: 华东师范大学, 2012.

[36] 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.

[37] LIN F, REITER R. Forget it[C]//Working Notes of AAAI Fall Symposium on Relevance. [S.l.: s.n.], 1994: 154-159.

[38] 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.

[39] 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.

[40] ACKERMANN W. Untersuchungen über das eliminationsproblem der mathematischen logik[J]. Mathematische Annalen, 1935, 110(1):390-413.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] ZHANG Y, ZHOU Y. Knowledge forgetting: Properties and applications[J]. Artificial Intelligence, 2009, 173(16-17):1525-1537.

[49] ZHANG Y, ZHOU Y. Properties of knowledge forgetting[C]//PAGNUCCO M, THIELSCHER M. Proceedings of NMR 2008. Sydney, Australia: [s.n.], 2008: 68-75.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] MAKSIMOVA L. Temporal logics of “the next” do not have the beth property[J]. Journal of Applied Non-Classical Logics, 1991, 1:73-76.

[56] D’AGOSTINO G. Interpolation in non-classical logics[J]. Synthese, 2008, 164(3): 421-435.

[57] 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.

[58] 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.

[59] 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.

[60] ZHANG Y, FOO N Y. Solving logic program conflict through strong and weak forgettings[J]. Artificial Intelligence, 2006, 170(8-9):739-778.

[61] 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.

[62] WONG K S. Forgetting in logic programs[D]. [S.l.]: The University of New South Wales, 2009.

[63] 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.

[64] 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.

[65] WANG Y, WANG K, WANG Z, et al. Knowledge forgetting in circumscription: A preliminary report[C/OL]//BONET B, KOENIG S. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. AAAI Press, 2015: 1649-1655. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9866.

[66] 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.

[67] GONÇALVES R, KNORR M, LEITE J, et al. On the limits of forgetting in answer set programming[J]. Artificial Intelligence, 2020, 286(0):103307.

[68] 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.

[69] GONÇALVES R, KNORR M, LEITE J. Forgetting in answer set programming–a survey [J]. arXiv preprint arXiv:2107.07016, 2021.

[70] 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.

[71] 朱嘉玮. 从“被遗忘权”看个人信息保护[J]. 现代商贸工业, 2020, 8.

[72] 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.

[73] 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.

[74] 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.

[75] KONIECZNY S, LANG J, MARQUIS P. Da2 merging operators[J]. Artificial Intelligence, 2004, 157(1-2):49-79.

[76] 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.

[77] 徐岱. 遗忘及应用[D]. 北京: 北京大学, 2011.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020: 1569-1576. https://aaai.org/ojs/index.php/AAAI/article/view/5517.

[93] KRIPKE S A. Semantical analysis of modal logic i normal modal propositional calculi [J]. Mathematical Logic Quarterly, 1963, 9(5-6):67-96.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] D'AGOSTINO G, HOLLENBERG M. Uniform interpolation, automata and the modal μ-calculus[J]. Logic Group Preprint Series, 1996, 165.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

[112] 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.

[113] 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.

[114] 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.

[115] 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.

[116] 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.

[117] WANG Y. On forgetting in tractable propositional fragments[J/OL]. CoRR, 2015, abs/1502.02799. http://arxiv.org/abs/1502.02799.

[118] COMON H. Tree automata techniques and applications[M]. [S.l.: s.n.], 1997.

[119] 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.

[120] 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.

[121] 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.

[122] HINTIKKA J. Distributive normal forms in the calculus of predicates[J]. Cambridge University Press, 1953, 20(2).

[123] 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.

[124] 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.

[125] BOLOTOV A. Clausal resolution for branching-time temporal logic.[D]. [S.l.]: Manchester Metropolitan University, 2000.

[126] ZHANG L, HUSTADT U, DIXON C. A refined resolution calculus for CTL[C]// International Conference on Automated Deduction. [S.l.]: Springer, 2009: 245-260.

攻读博士学位期间科研和论文情况

参与国家自然科学基金三项，发表CCF B类会议论文一篇，SCI论文两篇在审。

致　　谢

如果世界上有幸运儿，我想我应该是其中一个。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。

在论文即将完成之际，谨向我的导师致以最诚挚的谢意和最崇高的敬意。导师严谨的科研态度让我获益良多，能在知识表示与推理这个要求严谨的方向有点点成果，导师功不可没。感谢导师多年来给予的辛勤培养、无私教育和悉心关怀。导师在本文选题、研究、以及撰写过程中自始至终给予的悉心指导，使我能够在宽松的学术氛围中及时跟踪国际最前沿的学术动态并且全心全力地投入到感兴趣的科学研究中。导师崇高的师德风范、渊博的知识储备、严谨的治学态度、敏锐的学术直觉、忘我的工作精神、积极的生活态度使我深受教益。

感谢参与该博士学位论文的各位评审，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。

感谢我的家人付出的一切。感谢他们在生活上的悉心照顾和学业上的鼓励支持，博士论文的顺利完成同时凝结着他们的心血、汗水、理解和牺牲。

最后，向所有曾经给予我关心和帮助的人们致以最真诚的谢意和祝福。