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

随着计算机系统越来越复杂，系统正确性和系统及其系统描述（规范，specification）之间的一致性越来越难以得到保证。模型检测是一个保证系统正确性行之有效的方法之一。然而在模型检测中，若系统不满足给定的规范（即与规范不一致），如何更新系统使其能够与规范一致是长时间以来的一个重要问题。与这个问题密切相关的两个概念是最强必要条件（the strongest necessary condition, SNC）和最弱充分条件（the weakest sufficient condition, WSC），其分别对应于形式化验证中的最强后件（the strongest post-condition, SP）和最弱前件（the weakest precondition, WP）。此外，随着对系统信息越来越清晰，现有的规范不可避免地会与新的知识有冲突。此时，如何将之前融入的元素在不影响其他信息的情况下“移除”也是个亟待解决的问题。

系统规范的描述语言以时序逻辑为主。其中CTL（Computation tree logic）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表达能力更强的逻辑语言。。。。。

遗忘是一种知识抽取的技术，其被应用于信息隐藏、冲突解决和计算逻辑差等领域。本文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. 给出CTL下的遗忘的概念及其相关性质。首先，本文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，本文探讨了遗忘算子的代数属性，包括模块性、交换性和同质性；第三，表达定理表明遗忘和Zhang等人提出的四条准则具有当且仅当的关系，即：遗忘的结果满足四条准则，且满足那四条准则的公式为遗忘的结果。

2. 提出一种基于归结的方法计算CTL下的遗忘。该方法使用Zhang等人提出的归结系统，在这个过程中需要将CTL公式转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$（separate 。。。 global clauses），最后再将具有索引的$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句转换为CTL。在这一过程中需要计算的遗忘的公式总是和各个过程的输出保持互模拟等价关系。

3. 给出了CTL下遗忘封闭的情形——约束的CTL。在这种情形下限制了公式的长度为n、公式所依赖的模型个数为有限个及构成公式的原子命题是有限的。此时，公式的模型可以用其特征公式——CTL公式来表示。因此，遗忘的结果可以由其所有模型在给定原子命题集合上的特征公式的吸取来表示，显然该公式是一个CTL公式（即：遗忘在这种情形下是封闭的）。

4. 研究了μ-演算下的遗忘。μ-演算是一种具有均匀插值（uniform interpolation）性质，本文说明了μ-演算下的遗忘与均匀插值是等价的，这意味着μ-演算下的遗忘是封

闭的，这是其与CTL的不同。此外，研究了μ-演算下遗忘的基本属性和复杂性，为均匀插值的研究提供了新的角度。

5. 给出了遗忘与WSC（SNC）和知识更新（Knowledge update）的关系。WSC对模型的验证和修改具有重要作用，现有方法只能计算可终止模型的WSC，而像反应式系统这类不可终止的系统的WSC如何计算没有有效的方法。本文通过遗忘的方法给出了计算WSC（SNC），并用遗忘定义了知识更新使得其满足。。等人提出的知识更新应满足的八条准则。

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

其意义主要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。

信息化的快速发展和深度应用所引发的隐私安全挑战，成为了制约数据开放、共享、交换和应用的瓶颈，并引起了法律界和学术界的高度关注。从技术的角度，差分隐私保护算法作为一种重要的隐私保护技术，在面向多维及其复杂关联的数据隐私保护方面的研究还不够成熟。首先，由于数据类型混合、稀疏性、域值空间大等原因，差分隐私的多维数据处理面临隐私脆弱性、计算效率低等方面的挑战；其次，数据融合的关联性、背景知识攻击和策略型敌手攻击，数据的隐私性和可用性的矛盾成为了突出问题。针对上述问题，从博弈的角度探讨隐私与效用的均衡及优化，不失为一种较好的解决方案。本文重点围绕隐私与效用均衡这一核心问题，基于信息熵、优化理论和博弈均衡等相关理论和方法，以构建均衡、优化模型为主线，在隐私量化方法设计、隐私与效用博弈模型构建及均衡求解、优化模型建立及求解等方面，取得了一系列的成果，为从技术和管理相结合的视角解决隐私保护问题提供了借鉴。本文所取得的主要研究成果包括：

1. 提出了差分隐私的信息熵度量模型及方法。针对隐私量化问题，基于Shannon基本通信模型，结合差分隐私的随机扰动原理，给出了有噪声信道的差分隐私通信模型及其形式化描述；进一步通过定义差分隐私保护模型中的信息熵、条件熵、联合熵、互信息量以及条件互信息量等概念，设计了以信息熵为核心的隐私度量模型；针对多维属性及其关联问题，基于图模型和马尔可夫模型等提出了面向多关联属性的差分隐私信息熵度量模型及方法，并基于数据处理不等式和Fano 不等式给出了信息泄露量的上下界。理论分析与实验结果表明，所提出的量化模型和方法能够有效地实现差分隐私量化目标，为隐私泄露风险评估和隐私保护机制设计提供了基础支撑。

2. 提出了含背景知识攻击的差分隐私优化模型。在所建立的差分隐私通信模型的基础上，结合所提出的隐私度量模型与方法和损失压缩理论，建立含背景知识的敌手

模型，以此为基础提出了含背景知识攻击的差分隐私通信模型；在基于条件互信息量设计的隐私率失真函数基础上，提出了含背景知识攻击的最优化模型；进一步针对所提出优化模型的求解问题，利用Blahut-Arimoto交替最小化方法设计和实现了权衡隐私与效用的迭代最小化算法，并给出了其计算复杂度分析。理论分析和实验仿真结果表明，所提出的相关方法相对于对称信道隐私保护机制具有明显优势。

3. 提出了有序随机响应扰动方案(Orderly Randomized Response Perturbation，ORRP)。针对多维数据差分隐私保护所面临的隐私脆弱性和效率低问题，面向数据收集场景的隐私保护需求，提出了一种有序随机响应扰动方案，有效解决了现有隐私保护机制忽视数据分布的影响、处理域值空间大和数据稀疏导致计算效率低的问题。该方案以所提出的隐私度量模型为基础，通过对隐私保护与数据可用性之间矛盾的分析和量化，给出了满足数据质量约束下最小化隐私泄露的互信息优化模型的形式化描述，以此实现最优隐私机制的概率密度函数计算并实现随机扰动。同时，参考独立并联信道模型，将上述结果推广到了多维数据情形。最后，从隐私泄露、数据可用性质量、关联度损失等方面给予了理论分析和实验仿真。结果表明，所提出的ORRP方案在数据语义完整性、隐私性和数据可用质量等方面比现有的结果更具有优势。

4. 提出了隐私保护的攻防博弈(Privacy-Preserving Attack Defense，PPAD)模型。针对差分隐私系统中存在消息灵通的策略型敌手问题，围绕数据收集场景设计了差分隐私保护的选择策略，以此为基础提出了隐私保护的攻防博弈模型，并通过均衡求解实现了差分隐私保护中隐私与效用之间的平衡。该方案基于所建立的差分隐私基本通信模型，在分析隐私保护方和策略型敌手方各自目标的基础上，构建了隐私极小极大优化模型，给出了包含参与者集合、策略空间、收益函数等的隐私攻防博弈模型的形式化描述。论文巧妙地利用了隐私互信息量的内涵与外延，构造了隐私保护的效用函数，最终具体实现了一个两方零和对策博弈模型的构建。论文利用极小极大定理和凹凸博弈理论给出了该攻防博弈模型的均衡分析，并进一步基于最优策略响应设计了均衡求解的策略优化选择算法。理论分析与数值实验结果表明，所提出的模型及方法能有效解决等价隐私保护机制之间的比较问题，同时可用于隐私保护程度最差情况下的隐私泄露风险评估。

关键词： 遗忘理论（forgetting），最强必要条件（SNC），最弱充分条件（WSC），知识更新（knowledge update）

Abstract

The challenges of privacy and security caused by the rapid development of informatization and in-depth applications, have become a bottleneck restricting data opening, sharing, exchange, and application, and have attracted great attention from the legal and academic communities. From the perspective of technology, the differential privacy (DP) protection algorithm, as an important privacy protection technology, is not mature enough in the research of data privacy protection for multi-dimensional and complex associations. Firstly, due to the mixed data types, sparseness, and large domain value space etc, the multi-dimensional data processing of DP is faced with the challenges such as privacy vulnerability and low computational efficiency. Secondly, the relevance of data fusion, background knowledge attacks and strategic adversary attacks, and the contradiction between data privacy and usability have become prominent issues. For the problems mentioned above, it is a better solution to investigate the trade-off and optimization of privacy and utility from the perspective of the game theory. Thus, this article mainly focuses on the crucial problem of the trade-off between privacy and utility. Based on information entropy, optimization theory and game equilibrium and other related theories and methods, the equilibrium and optimization models are constructed as the main line of this research. A series of results have been achieved in designing of privacy quantification methods, constructing and solving game model between privacy and utility, optimization model establishment and solving, etc., which provide a reference for solving privacy protection issues from the perspective of combining technology and management. The major contributions can be summarized as follows.

1. The information entropy metric models and methods of DP are proposed. For the quantitative problem of privacy, the noisy DP communication model and formalization statement are defined based on the Shannon's fundamental communication model and the randomized perturbation principle of DP. Further, the notions of information entropy, conditional entropy, joint entropy, mutual information and conditional mutual information, etc., are defined under the differential privacy model, and then, the privacy metric models with information entropy as the core are designed. For the problem of multi-dimensional and correlated attributes, based on the graph and Markov model, etc., a privacy metric model and method for multi-dimensional and correlated attributes is proposed. Then, the upper and lower bounds of privacy leakage are quantified by using data processing inequality and Fano's inequality. Theoretic analysis and experimental results are demonstrating the proposed metric model and method can effectively

achieve the goal of DP quantification, and further provide basic support for privacy leakage risk assessment and privacy protection mechanism design.

2. The differential privacy optimization model with background knowledge attacks is proposed. Based on the established fundamental communication model of the DP, lossy compression theory and the proposed privacy metric model, the adversary model which has relevant background knowledge is established, and further the DP communication model with background knowledge attacks is proposed. By using conditional mutual information measures privacy, this paper updates the form of the well-known rate distortion function, and proposes the differential privacy optimization model with background knowledge attacks. Further, the alternating minimization iteration algorithm solving the proposed optimization model is designed and implemented based on the Blahut-Arimoto alternating minimization method, and the computation complexity analysis is provided. Theoretic analysis and experimental results are demonstrating the proposed method have significant advantages in data quality and privacy leakage when compared with the existing symmetrical channel mechanism.

3. The orderly randomized response perturbation (ORRP) scheme is proposed. For the problem of low efficiency and privacy vulnerability when deal with multi-dimensional data using local differential privacy, and facing the privacy protection requirements of data collection scenarios, this paper proposes an orderly randomized response perturbation scheme. The proposed ORRP scheme effectively solves the impact of the existing privacy protection mechanisms ignoring data distribution, and the problem of low computing efficiency caused by the large processing domain value space and sparse data. To be specific, the proposed ORRP scheme based on the prior proposed privacy metric model. A mutual information optimization model subjects to a given data quality loss constraint to minimize privacy leakage, is proposed by analyzing and quantifying the requirements of privacy and data quality. Further, the probability density function (PDF) of the optimal privacy mechanism is computed by the means above, and it is used to achieve randomized perturbation. Meanwhile, referring to the independent parallel channel model, the above methods are extended to the case of multi-dimensional data. Finally, theoretical analysis and experimental simulations are given in terms of privacy leakage, data usability quality, and correlation loss. The results demonstrate that the proposed ORRP has more advantages than the existing methods in terms of data semantic integrity, privacy and data availability quality.

4. The privacy-preserving attack and defense (PPAD) game model is proposed. For the problem of informed and strategic adversary in the differential privacy system, the selection strategy of differential privacy protection is designed around the data collection scenarios. On the basis of the above, the PPAD game model is proposed, and the trade-off between privacy

and utility in the protection of differential privacy is achieved by solving the equilibrium. The proposed scheme is based on the established differential privacy basic communication model. The privacy minimax optimization model is established by analyzing the privacy goals of defender and strategic attacker, and further the formalization statement of PPAD is provided, which includes players' sets, strategic spaces and payoff functions etc. This paper cleverly uses the connotation and extension of private mutual information to construct the utility function of privacy protection, and finally realized the construction of a two-person zero-sum (TPZS) game model. Then, this paper provides the game analysis by using von Neumann's minimax theorem and concave-convex game, and further designs a strategy optimization selection algorithm to calculate saddle point based on the optimal strategy response. Theoretic analysis and numeric simulation results show that the proposed model and method can effectively solve the problem of comparison between equivalent privacy mechanisms, and also can be used for privacy leakage risk assessment in the worst case of privacy protection.

Keywords: Privacy metric, differential privacy protection, rate-distortion function, game equilibrium, optimization model

第一章　绪论

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

1.1　研究背景与意义

1.1.1　研究背景

形式化验证是一种广泛应用在硬件[1-3]和软件系统中[4-5]的有别于测试的、采用数学方法证明系统满足给定特性的验证（verification）技术。软件和硬件的缺陷会导致严重的后果，如表1.1中列出的几个重大事件。近年来，为了减少系统（尤其是像火箭发射系统和卫星发射系统等关键领域的系统）错误带来的损失，形式化方法的研究与应用越来越受到人们的重视。包括INTEL、AMD、IBM、NVIDIA、CADENCE、Motorola、西门子和微软等在内的大型公司纷纷引入了形式化验证方法。与此同时，学术界也在形式化验证领域取得了突破性的成果，比如：剑桥大学进行了ARM6处理器的验证[6]，德国的Verisoft项目验证了一个一万行的操作系统内核[7]。

表 1.1: 由系统故障引起的重大事件概览

发生时间	事故事件（系统错误）	造成的损失
1991年	美国爱国者导弹系统舍入错误	28名士兵死亡、100人受伤等
1996年	阿丽亚娜5型运载火箭因软件不同飞行条件下代码重用	导致其与其他卫星在瞬间灰飞烟灭
1999年	火星探测器用错度量单位	引起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导致	导致动车脱节脱轨、多人失去生命

形式化验证有两种主要的验证方法：自动定理证明（Automated theorem proving）和模型检测（Model checking)。在自动定理证明中，系统模型和规范（specification）被同一形式化语言分别描述为φ_{imp}和φ_{spec}，剩下的任务就是证明性质，如:$\varphi_{imp} \rightarrow \varphi_{spec}$或$\varphi_{imp} \leftrightarrow \varphi_{spec}$。常用的自动定理证明方式有基于规约（Resolution）[8]和常用于模态逻辑的基于表推理（tableau）[9]的方法。然而，在自动定理证明中寻找不变式

（invariant）是一个相当困难的问题。因此，为了避免像Hoare逻辑[10]、动态逻辑[11]和分离逻辑[12]在形式化验证中寻找不变式，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[13]。

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

近年来，模型检测问题在知识表示与推理（KRR）领域的推进下取得了丰富的科研成果，例如：基于SAT的有界（bounded）模型检测[16]和基于OBDD的符号模型检测[17]已经使得模型检测问题在时间和空间效率上取得了很大的进步，在一定程度上缓解了其固有的状态空间爆炸问题。此外，大量优质的模型验证器（如：NuSMV[1]、SPIN[2]、Uppaal[3] 等）也相继发展起来，并且大部分的验证器都可以用来验证多种时态逻辑描述的公式。

时态逻辑作为描述系统规范的形式化语言，它研究状态随时间变化的系统的逻辑特性。由于软件和硬件的运行的本质是状态变化的过程，所以时态逻辑在软件程序验证和硬件验证中应用得相当广泛。计算树逻辑（Computation Tree Logic, CTL）是分支时态逻辑的一种，其模型检测是多项式时间可行的。然而，CTL表达系统性质的表达能力不如μ-演算（μ-calculus），如：“某给定的系统中存在一条路径使得该路径上的第偶数个状态满足特定的性质”这一规范是不能用其他时态逻辑表示的[18]。充分考虑这两种逻辑语言自身的特性，本文将要研究的描述规范的逻辑语言限制到CTL和μ-演算下。因此，本文所说的公式指CTL（或μ-演算）公式，即用来描述一个规范（或性质）的公式是CTL（或μ-演算）公式。

从知识抽取（或“剪掉”）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掉而只抽取关注的部分信息。此外，随着时间的推移，由于某些原因使得系统的部分信息过时，此时就需要将这样的过时的信息在不影响其他信息的情况下“剪掉”。考虑如下示例：

例 1.1 (汽车制造企业模型)**.** 一个汽车制造企业能够生产两种汽车：小轿车(se)和跑车(sp)。每隔一段时间，该企业都会做一个生产决策(d)，即：合理的生产计划。刚开始的时候，该企业做出了具有三个选择和方案：（1）先生产足够的se，然后在生

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

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

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

产sp；（2）先生产足够的sp，然后在生产se；（3）同时生产se和sp。这一过程可以有图1.1中的Kripke结构（带标志的状态转换图）$\mathscr{M} = (S, R, L)$形式化地展现出来，其中:

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

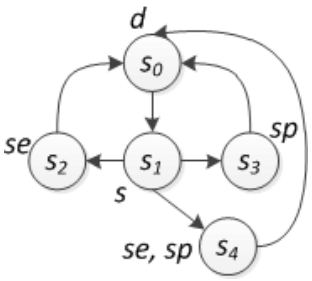


图 1.1: 汽车制造企业模型

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[19]。但是对于给定原子命题的集合，从这些大型系统中“移除”掉这些原子而保持与这些原子无关的性质是一个复杂的问题。此外，在这种情形下，两个重要的概念：最强必要条件（SNC）和最弱充分条件（WSC）问题也随之产生，其中SNC是指最一般的结论，WSC指最特殊的诱因。

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

1.1.2　研究意义

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

最强后件（SP）和最弱前件（WP）（分别对应于上文提到的SNC和WSC）是形式化验证中的两个重要的概念，其不仅被用于汇编语言程序推理[20]和制定验证条件[21]，还被应用于形式化验证过程中的负例生成[22]和系统精化（refinement）[23]。当$\mathscr{M} \not\models \varphi$时，若知道某个性质$\psi$使得若$\mathscr{M}$按照此性质进行修改后得到的新模型$\mathscr{M}'$能

满足φ，即ψ为使得$\mathscr{M} \models \varphi$成立的充分条件（$\mathscr{M} \models \psi \rightarrow \varphi$）。然而，现有的方法不能直接应用于计算当$\mathscr{M}$为不终止系统（如：反应式系统（reactive system））时使得“$\mathscr{M} \models \varphi$”为真的最强必要条件（SNC）和最弱充分条件（WSC）（其详细原因将在下文指出）。此时，探索在$\mathscr{M}$下使得φ满足的定义在某个符号集合上的SNC和WSC将更进一步完善模型检测问题，同时也为基于WSC的负例生成和精化提供了理论依据和新的计算方法。

此外，在上文中提到现有的方法不能直接应用于计算当$\mathscr{M}$为不终止系统时使得“$\mathscr{M} \models \varphi$”为真的最强必要条件（SNC）和最弱充分条件（WSC），在本文中将探索一种叫做*遗忘理论*（forgetting）的方法来计算充分（必要）条件。正如下文将要说到的，遗忘理论作为知识表示与推理（KRR）中重要理论，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，作为本文一个重要的研究意义，本文的研究将为时序逻辑下的遗忘理论的研究提供一个理论框架。与此同时，借助遗忘理论计算上述形式化验证问题中的充分（必要）条件，这架起了KRR与形式化验证的桥梁。

1.2 相关研究工作回顾

遗忘（forgetting）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）两种称呼。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（logic differences）。此外，其也被用于包括信念更新（belief update）、修改（repair）、规划（planning）和知识独立性的其他领域。

在规划中，遗忘主要用来计算其对应的后继状态公里所需要的SNC和WSC。下面就与本文密切相关的遗忘理论和SNC（WSC）进行详细的回顾。

1.2.1 遗忘理论

遗忘这一词源于Lin等人关于一阶逻辑（first-order logic，FOL）工作[24]，在此之前的研究中多提到的是均匀插值[25-26]和SOQE[27]。

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

$$
\begin{aligned}
&Forget(\varphi, \emptyset) = \varphi, \\
&Forget(\varphi, P \cup \{q\}) = Forget(Forget(\varphi, \{q\}), P).
\end{aligned}
$$

在FOL中，遗忘通常被看为SOQE问题的一个实例。特别地，从FOL公式φ中遗忘掉一个n-元为此P得到结果为一个二阶公式$\exists R\varphi[P/R]$[24]。从这个角度看来，遗忘就是找到一个与二阶公式$\exists R\varphi[P/R]$等价的一阶公式。然而，二阶逻辑的表达能力是严格大于一阶逻辑的，因而可以容易得出FOL下的遗忘不是封闭的，也就是说从有的一阶公式中遗忘掉某些谓词得到的结果不可以用一阶公式来表示。作为FOL的一个子类，描述逻辑公式的遗忘也不总是存在的[28]，甚至对最基本的描述逻辑ALC而言，遗忘的存在性问题都是不可判定的。尽管如此，描述逻辑作为一种在语义网领域很重要的语言，其子类（包括ALCOHI和ALCOIH）中的遗忘通常被用来抽取视图（review）[29-33]。

现有的研究一阶逻辑和描述逻辑下的遗忘的方法有基于消解（resolution）和基于Ackermann引理的方法[34]。其中基于消解的方法是一种基于子句的归结反驳方法，消解规则是其基础。通常在这种方法中首先要把公式转换为其子句形式，然后再使用消解规则，最后将得到的子句集合中包含有要遗忘的谓词（原子命题）“移除”掉后得到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消解规则和转换规则）。基于Ackermman引理的方法主要是直接或间接（扩展）下面的Ackermann引理得到的。

引理 1.1 (Lemma 6.1 of[34]). *给定关系变元X和一阶公式$\alpha(\bar{x},\bar{z})$和$\beta(X)$，其中$\bar{x}$和$\bar{z}$为普通变元构成的多元组、$\bar{x}$中变元的个数与X的参数个数相同、且α中不包括X。*

- *若$\beta(X)$关于X是正的，即：X在$\beta(X)$中的每次出现前面都有偶数个“$\neg$”符号，则：*

$$\exists X\{\forall\bar{x}[X(\bar{x})\rightarrow\alpha(\bar{x},\bar{z})]\wedge\beta(X)\}\equiv\beta(X)^{X(\bar{x})}_{\alpha(\bar{x},\bar{z})}。$$

- *若$\beta(X)$关于X是负的，即：X在$\beta(X)$中的每次出现前面都有奇数个“$\neg$”符号，则：*

$$\exists X\{\forall\bar{x}[\alpha(\bar{x},\bar{z})\rightarrow X(\bar{x})]\wedge\beta(X)\}\equiv\beta(X)^{X(\bar{x})}_{\alpha(\bar{x},\bar{z})}。$$

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）[35]。模态逻辑中的遗忘与经典逻辑下的遗忘不同，因为模态逻辑系统中引入了模态词，此时就不能以简单的谓词（命题）替换的方式获取遗忘的结果，如：

例 1.2. [36] 令S5公式$\varphi=\mathrm{K}p\wedge\neg\mathrm{K}q\wedge\neg\mathrm{K}\neg q$，则如果使用命题逻辑下的计算方法得到的结果为$\varphi[q/\top]\vee\varphi[q/\bot]$。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条件被给出，值得注意的是这四个条件与知识遗忘形成了“当且仅当”的关系。换句话说，当知道某一个公式

满足那四个条件则该公式为遗忘的结果，当知道某一结果为遗忘结果时它一定满足那四个基本条件。

均匀插值作为遗忘的一个对偶概念，这里有必要介绍一下模态逻辑系统的这一性质的研究现状。S5、K和KD模态逻辑系统具有均匀插值性质[37]，而模态逻辑系统没有均匀插值性质，如：模态逻辑量化的S5[38]和K4、和S4及其扩展都没有均匀插值性质[39]，因此其遗忘也不是封闭的。因此，在研究这些具有均匀插值性质的模态逻辑下的遗忘时可以借鉴S5系统下的遗忘方法，也可以参考K系统下的基于消解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[34]。

在非单调推理（non-monotonic reasoning）环境中，科研工作者们也从遗忘的基本条件的角度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和JAIR上的文章[40-46]，Eiter、Gonccalves等人的综述[47-48]。

在现实生活中，遗忘有很多应用。下面列出几点：

- 计算后继状态公理：在规划问题中，根据最强必要条件和最弱充分条件有利于求出后继状态公理[49]。在该文章中，最强必要条件和最弱充分条件都用遗忘来计算；
- 信息隐藏：在有的关键领域，为实现隐私保护，敏感信息必须被隐藏。而有的系统现在都基于本体，要做到隐私保护，只需要将那些敏感的概念（concept）和角色（role）符号隐藏（遗忘）就行了；
- 计算逻辑差：
- 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$(U1)-(U8)$较为常用，本文也使用这几个基本条件；
- 提取本体的概要：当一个本体工程师想要快速了解并测试一个本体的内容时，能事先快速地提取出该本体的概要是非常有用的。如果一个本体含有很多无关信息时，这将使得事半功倍。

1.2.2 SNC和WSC

正如前面所说，WSC（SNC）对于软件工程中系统的形式化验证有着及其重要的作用。一般说来，最强必要条件（SNC）是最一般的推论（the most general consequence），即：命题成立时能推出的最强的后件（the strongest post-condition, SP），

SNC能够蕴涵所有的必要条件；最弱充分条件（WSC）是最特殊的诱因（the most specific abduction），即：使得命题成立的最弱的前提条件（the weakest precondition，WP）。

特别地，给定一个程序（program）S和某一状态（state）的规范（specification）Q，则S关于Q的WSC是一个能够描述S初始状态的规范，其中S满足以下两个条件：（i）S必须终止，（ii）S执行完成后必须到达能满足Q的状态。Dijkstra提出了四条规则来计算这样的后件，程序语言里的四种语句（即：赋值语句（assignment rule）、顺序语句（sequence rule）、条件语句（conditional rule）和循环语句（loop rule））分别对应了这四种规则[50]。此外，这两个概念还可用于系统精化[23]、模型检测中的负例产生[22]、汇编语言程序的推理[20]和制定验证条件[21]。

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[51]，SNC和WSC都可以用遗忘来计算[52-53]。之后，SNC和WSC被扩展到FOL下，且用SOQE实现了SNC和WSC的计算[53]。

这里对SNC和WSC的发展和应用只做了简单的概述，在背景知识部分再对其在命题逻辑和一阶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很好的解决反应式系统下的WSC（SNC）的求解。然而，WSC是一种进行系统修改的重要知识，寻求一种有效的求解方法有利于系统正确性的确保。在知识表示与推理中，一种叫做遗忘的技术可以用于求解给定理论（公式）的WSC（SNC）。但是，就如上面所述时序逻辑下的遗忘理论尚处于不成熟阶段，没有一个统一的理论框架。此外，如何用遗忘来计算给定系统模型和性质的WSC（SNC）也是一个重要问题。

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

(1) *CTL*和μ-演算的遗忘理论

本文研究了CTL中遗忘理论的方法和性质，特别是其遗忘结果的存在性、复杂性等，为探索用遗忘理论计算SNC和WSC提供理论基础。具体说来，遗忘理论具有削弱（Weakening，(W)）、正维持（Positive Persistence，(PP)）、负维持（Negative Persistence，(NP)）、无关性（Irrelevance，(IR)）等基本性质[35]。本文探索CTL和μ-演算的遗忘理论的以上性质，并探讨其与存在性之间的关系。此外，本文深入研究了CTL和μ-演算遗忘理论的基本准则、发现计算CTL和μ-演算遗忘结果的算法以及探讨CTL和μ-演算子

类与遗忘相关问题的计算复杂度，为研究计算SNC和WSC的性质、算法以及基本准则等作好铺垫。具体说来，有以下两点：

- **CTL的遗忘理论：** CTL不具有均匀插值（uniform interpolation）性质[54]，即：不存在一个算法使得对于任意的CTL公式，其遗忘掉任意原子命题的集合得到的结果仍然是CTL公式。在这种情况下，本文除了探讨了上述CTL中遗忘的性质，还研究了CTL子类的遗忘理论，特别是能保证其遗忘结果仍然是CTL可表达的子类。在这些子类中，一个特殊的子类为约束CTL（bunded CTL）。在这种情形下，每个公式的模型是有限个数个，且每一个模型都能用一个CTL表示。因此，其遗忘理论是封闭的。

- **μ-演算的遗忘理论：** 与CTL不同，μ-演算虽然表达能力比CTL强，其可满足性问题也比CTL的复杂，但是μ-演算具有均匀插值性质[55]。这意味着对于在任意的μ-演算句子中遗忘掉任意的原子命题的集合得到的结果仍然是μ-演算公式。本文给出了μ-演算下遗忘理论的主要框架：包括上述遗忘理论的性质和计算遗忘的方法。特别地，表明了本文提出的μ-演算下的遗忘定义与均匀插值是一对对偶概念，即本文中的遗忘理论的性质也是均匀插值的性质，这位研究均匀插值提供了另一种途径。此外，本文还说明了当μ-公式为吸取μ-公式时，计算遗忘理论可以在多项式时间内完成，这为μ-演算下的遗忘的计算提供了一种有效的方法。

(2) *CTL下遗忘理论的算法的研究及实现*

基于上述研究结果，设计并实现一个计算CTL和μ-演算遗忘结果的原型系统，并从实验角度研究其计算代价以启发快速的计算遗忘结果的算法。具体说来，本文给出了一种基于消解的计算CTL遗忘的算法，并使用Prolog实现了该算法。

(3) *遗忘理论在形式化验证和知识更新中的应用*

给出了使用遗忘理论计算给定有限系统模型的SNC和WSC。如上所述，一个有限的系统模型能够被一个CTL公式描述，因而可以使用遗忘理论计算其SNC和WSC。此外，知识更新是一种使用新发现的性质更新已有理论的技术，本文探讨了如何使用遗忘理论更新CTL和μ-演算下的理论。表明了使用遗忘理论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4个**关键问题**：

(1) *CTL的遗忘什么情形下存在？如何计算遗忘？*

CTL是一种分支时序逻辑，已有文献表明CTL不具有均匀插值性质。与此同时，CTL还引入了时态算子（temporal operator）。在此情形下，研究CTL的遗忘就不能像已有的经典命题逻辑和模态逻辑S5那样，因为遗忘与均匀插值是一对对偶概念，即：它

们可以互相证明彼此的存在性。为此，本文深度刨析现存的消解规则，提出了一种基于消解的计算遗忘的方法。该方法表明，当所有在转换为CTL标准形式过程中引入的新的原子命题都被“消除”掉时，使用这一方法得到的结果即为遗忘的结果，即：这一方法是可靠的。尽管CTL的遗忘理论不是封闭的，但是本文给出：当被消解的原子命题只能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，考虑到现实生活中有的情形下只需要考虑有限模型，所以本文研究了约束CTL下的遗忘理论。研究表明，这种情况下的遗忘结果可以由有限个模型的特征公式（一种CTL公式）的吸取来表示，从而说明了这种情形下的遗忘是封闭（存在）的。

(2) 遗忘理论与反应式系统的*SNC*和*WSC*的关系

在经典命题逻辑和一阶逻辑中，遗忘理论与SNC和WSC的关系分别已经被Lin和Doherty等人分别提出[22][23]。特别地，经典命题逻辑中的SNC和WSC被用于规划问题中的后继状态公理的计算。这里所说的SNC和WSC都是在给定的命题公式或一阶公式下的。本文给出，当给定一个有限反应式系统（Kripke structure）时，将该系统表示为其特征公式时就可使用上述所说的CTL和μ-演算的遗忘理论计算SNC和WSC。

(3) *CTL*和μ-演算的遗忘在推理问题上的复杂性

计算复杂性理论致力于将可计算问题根据它们本身的复杂性分类。研究表明，在经典命题逻辑中：CNF（Conjunctive normal form）公式的遗忘的推理问题最难是Π_2^p完全的，DNF（Disjunctive normal form）公式的遗忘的推理问题是co-NP-C的；在模态逻辑S5中，遗忘的模型检测问题是NP-C的，对应的推理问题是Π_2^p完全的。基于此，本文从现有复杂性结果和自动机理论研究CTL和μ-演算的遗忘在模型检测和推理问题上的复杂性。研究表明μ-演算下的复杂性。。。。。

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。全文共分为七章，组织结构如图1.2所示，各章节内容的具体安排如下：

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，介绍了本文研究所设计的语言的语法语义即相关技术。首先，给出了经典命题逻辑下解释（赋值）的定义，并基于此描述了解释本文研究的语言的模型结构——Kripke结构。在解释清楚语言所依赖的模型结构之后，本章给出CTL和μ-演算的语法和语义。最后介绍与上述逻辑语言密切相关的两个技术：遗忘和消解。

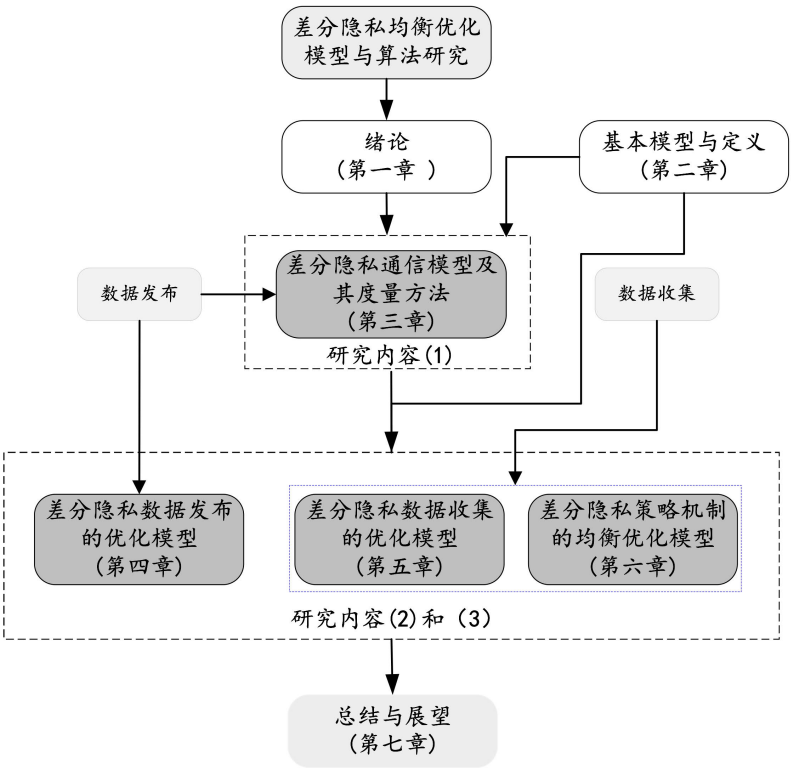


图 1.2: 本文的章节内容组织结构图

第三章为给出*CTL*下的遗忘的定义及其基本性质。首先，给出原子命题集合上的模型间互模拟的定义，并介绍这一定义的基本属性；其次，根据互模拟来定义CTL下的遗忘，并研究遗忘的属性，包括表达性属性（representation theory）和复杂性结果等。

建立差分隐私与信息论的基本联系，奠定本文的研究基础。首先，基于Shannon基本通信模型，介绍了差分隐私基本通信模型，并给出了形式化的模型表达。其次，以差分隐私的基本通信模型为基础，引入信息熵、互信息、失真的概念对隐私与效用进行度量，提出信息熵度量模型及方法。进一步，在基本的度量基础上，针对多维关联属性的情景，提出面向关联属性的差分隐私信息熵度量方法，本章中的度量模型及方法为开展后续章节的研究奠定了基础。

第四章为基于消解的*CTL*遗忘计算。首先，给出

研究了隐私与数据效用权衡的最优差分隐私机制。首先，基于第二章基础、第三章差分隐私通信模型及度量方法，分析了隐私系统的目标，借鉴率失真理论构建了面向差分隐私数据发布的优化模型。其次，在基本的通信模型基础上引入敌手模型，并考虑了敌手拥有关联辅助背景知识对隐私泄露的影响，提出了基于联合事件的互信息隐私度量。随后，修改率失真表述形式，提出了最小化隐私泄露的优化模型，用于

获得隐私机制的概率分布函数。此外，设计了互信息隐私最优信道机制的近似迭代求解算法，并给出具体的验证分析。

第五章为约束*CTL*下的遗忘，围绕隐私保护机制设计的两个阶段，研究了面向数据收集的多维数据最优隐私机制。首先，以第三章的度量为基础，形式化互信息隐私(MI-privacy)最优化模型，设计模型求解算法寻找最优机制的概率密度函数。随后，将其作为基本构建模块应用到多维数据，提出有序随机响应扰动(ORRP)方案。最后，针对提出的ORRP方案，介绍了隐私、数据效用以及相关度损失的评估与分析，并利用真实数据集给出了所提方案的实验分析。

第六章为μ-演算下的遗忘，应用博弈均衡理论研究了差分隐私策略机制选择问题。通过分析隐私保护系统参与者的隐私目标，基于信息熵度量模型及方法将隐私目标形式化为隐私泄露的极大极小问题。然后，分析系统参与者的可行策略，构建了隐私保护的攻防博弈(PPAD)模型。针对本文关注的应用，实例化PPAD为两方零和对策博弈模型，并提供了均衡的理论分析以及算法实现。最后，阐述了均衡在隐私保护中的内涵及意义。

第八章为应用与实验，首先讲述如何将遗忘应用于计算SNC（WSC）和知识更新，其次给出基于消解的算法实现模型的实验结果。

第**??**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　Kripke结构、时序逻辑、模型检测以及遗忘理论

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：*Kripke*结构、时序逻辑（尤其是计算树逻辑（*CTL*）和μ-演算）、模型检测和遗忘理论。首先，介绍解释时序逻辑语言所需的模型结构，即*Kripke*结构。其次，主要介绍时序逻辑中本文探讨的计算树逻辑和μ-演算。为了更加明确本文的研究动机，本章将详细介绍模型检测的基本概念和一些主要的性质。此外，遗忘理论是本文的研究重点，其概念、性质及在各个研究领域的研究和应用情况将会被当作本章的重点详细介绍。

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

2.1　Kripke结构

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

2.1.1　真假赋值和K-解释

经典命题语言$\mathscr{L}^p$由以下三类符号构成：

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

$\mathscr{L}^p$的原子公式的集合和公式的集合分别记作$Atom(\mathscr{L}^p)$和$\mathscr{F}(\mathscr{L}^p)$。其中，$Atom(\mathscr{L}^p)$是命题符号的集合，且$\varphi \in \mathscr{F}(\mathscr{L}^p)$当且仅当它能由（有限次使用）以下的三条规则生成[?]：

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

此外，也称“ture”和“false”为原子公式，分别记为“$\top$”和“$\bot$”。原子命题或其否定称为文字，有限个文字的吸取称为子句。

例 2.1. 下面几个字符串为$\mathscr{L}^p$的公式：

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

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

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

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

此时，例 2.1中的公式$(((\neg p) \leftrightarrow (q \vee r)) \rightarrow (r \wedge p))$就可写为$(\neg p \leftrightarrow q \vee r) \rightarrow r \wedge p$。当然，为了看起来方便，有的括号可以不必省略。

在讨论了命题公式的语法结构之后，接下来将讨论其语义解释。

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

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

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

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

- $(\varphi \leftrightarrow \psi)^v = \begin{cases} \top, & \text{如果} \varphi^v = \psi^v; \\ \bot, & \text{否则。} \end{cases}$

对于任意的命题公式φ和真假赋值v，当$\varphi^v = \top$时，称v是公式φ的一个模型，也可以记为$v \models \varphi$，读作v满足φ。一般地，当存在一个真假赋值v使得$v \models \varphi$，则称公式φ是可满足的。如果φ是可满足的，且$\neg\varphi$是不可满足的，则称φ是有效的。

值得注意的是，命题逻辑的语义也可定义在“解释（interpretation）”上。一个解释I是$\mathscr{A}$的子集。除了对原子命题$p \in \mathscr{A}$，I对公式的解释如真假赋值一样。在解释原子命题$p \in \mathscr{A}$上，p^I为真当且仅当$p \in I$。模型和可满足的定义与真假赋值的类似。

模态逻辑是经典逻辑的扩充，它是经典逻辑中引进“必然”和“可能”这两种模态词得到的。如上所述，命题的真假值只有两种，命题是真的（1）或是假的（0）。而在模态逻辑中，把命题区分为必然真的命题和并非必然真的命题，把假命题区分为必然假的和并非必然假的命题。对于任何命题φ，可以有两种模态命题：“φ是必然的”和“φ是可能的”。值得注意的是，时序逻辑也是模态逻辑的一种[56]。尽管如此，本文在说模态逻辑的时候通常指不带有时序操作符的情况，说时序逻辑时指带有时序操作符的情况。

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

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

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

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

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

- $\mathscr{M} \not\models \bot$，$\mathscr{M} \models \top$；
- $\mathscr{M} \models p$当且仅当$p \in w$，其中$p \in \mathscr{A}$；

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

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

2.1.2　Kripke结构的定义及相关术语

通常一个转换系统（transition system）能够被抽象为一个Kripke结构[19]。如上文所说，一个Kripke结构是一个三元组$\mathscr{M} = (S, R, L)$，其中：

- S是状态的非空集合；
- $R \subseteq S \times S$是状态转换函数；
- $L : S \to 2^{\mathscr{A}}$是一个标签函数。

在本文中，要求R是一个串行关系（serial relation），也即是对于S中的任意元素s，都存在S中的一个元素s'使得$(s, s') \in R$。

给定一个Kripke结构$\mathscr{M} = (S, R, L)$，$\mathscr{M}$上的一条路径是一个无限的状态序列$\pi = (s_0, s_1, \ldots)$且满足对于任意的$j \geq 0$都有$(s_j, s_{j+1}) \in R$，路径上的状态s被记为$s \in \pi$。当给路径π引入一个状态s作为下标，记为π_s，则称该路径是起点为该状态s的一条路径。如果对于$\mathscr{M}$中的任意状态s'，都有一条以s为起点的路径π_s使得$s' \in \pi_s$，那么称状态s为一个*初始状态*。给定s_0为$\mathscr{M}$中的一个初始状态，为了容易看出该初始状态，将该Kripke结构写为四元组(S, R, L, s_0)，并称该结构为*初始结构*以区分于原来的三元组。

*树*是一种只有一个根节点（没有其他节点指向且可达于其他节点的节点）无环图。给定一个初始结构$\mathscr{M} = (S, R, L, s_0)$和一个状态$s \in S$，定义在$\mathscr{M}$上以$s$为根节点的深度为为$n$（$n \geq 0$）的*计算树*$\mathrm{Tr}_n^{\mathscr{M}}(s)$被递归定义如下[57]：

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

一个初始结构$\mathscr{M} = (S, R, L, s_0)$和一个状态$s \in S$构成一个K-*结构*（或K-解释），写作$\mathscr{K} = (\mathscr{M}, s)$。在K-结构$\mathscr{K} = (\mathscr{M}, s)$中，若$s = s_0$，则称该K-结构为*初始K-结构*，此时有$\mathscr{K} = (\mathscr{M}, s_0)$。

2.2 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的系统的逻辑特性。由于软件和硬件的运行的本质是状态变化的过程，所以时态逻辑在软件程序验证和硬件验证中应用得相当广泛。计算树逻辑（Computation Tree Logic, CTL）是分支时态逻辑的一种，其模型检测是多项式时间可行的。然而，CTL表达系统性质的表达能力不如μ-演算（μ-calculus），如：“某给定的系统中存在一条路径使得该路径上的第偶数个状态满足特定的性质”这一规范是不能用其他时态逻辑表示的[18]。充分考虑这两种逻辑语言自身的特性，本节主要介绍CTL和μ-演算。因此，本文所说的公式指CTL（或μ-演算）公式，即用来描述一个规范（或性质）的公式是CTL（或μ-演算）公式。

2.2.1 计算树逻辑（CTL）

CTL由Clark和Emerson等人于1986年提出[58]。CTL的语言$\mathscr{L}$由下面的几类符号构成：

- 原子命题的集合$\mathscr{A}$；
- 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“吸取”和“否定”；
- 路径量词：A和E，分别表示“所有”和“存在”；
- 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“)”。

CTL的时序算子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，AF等。与经典命题逻辑一样，给联结符号规定优先级，有时候会带来意想不到的方便。CTL中的联结符号的优先级如下序列所示，每个左边的联结符号优先于右边的联结符号：

$$\neg \quad \mathrm{EF} \quad \mathrm{EG} \quad \mathrm{AX} \quad \mathrm{AF} \quad \mathrm{AG} \quad \wedge \quad \vee \quad \mathrm{EU} \quad \mathrm{EW} \quad \mathrm{AW} \quad \rightarrow$$

因此，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：

$$\phi ::= \bot \mid \top \mid p \mid \neg\phi \mid \phi \vee \phi \mid \mathrm{EX}\phi \mid \mathrm{EG}\phi \mid \mathrm{E}(\phi \ \mathrm{U} \ \phi) \tag{2.1}$$

其中，$p \in \mathscr{A}$。$\mathscr{L}$中其他形式的公式可以通过下面的定义（使用上述定义中(2.1)的形式）得到：

$$\varphi \wedge \psi \stackrel{def}{=} \neg(\neg\varphi \vee \neg\psi) \tag{2.2}$$

$$\varphi \rightarrow \psi \stackrel{def}{=} \neg\varphi \vee \psi \tag{2.3}$$

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

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

$$\mathrm{E}(\varphi \mathrm{W} \psi) \stackrel{def}{=} \neg\mathrm{A}((\varphi \wedge \neg\psi)\mathrm{U}(\neg\varphi \wedge \neg\psi)) \tag{2.6}$$

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

$$\mathrm{EF}\varphi \stackrel{def}{=} \mathrm{E}(\top \mathrm{U} \psi) \tag{2.8}$$

$$\mathrm{AX}\varphi \stackrel{def}{=} \neg\mathrm{EX}\neg\varphi \tag{2.9}$$

$$\mathrm{AG}\varphi \stackrel{def}{=} \neg\mathrm{EF}\neg\varphi \tag{2.10}$$

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

CTL的语义定义在Kripke结构上，可以严格地描述如下。

定义 2.4 (CTL的语义). 给定*CTL*公式φ，初始结构$\mathscr{M} = (S, R, L, s_0)$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：

- $(\mathscr{M}, s) \not\models \bot$且$(\mathscr{M}, s) \models \top$；
- $(\mathscr{M}, s) \models p$ 当且仅当$p \in L(s)$；
- $(\mathscr{M}, s) \models \varphi_1 \vee \varphi_2$当且仅当$(\mathscr{M}, s) \models \varphi_1$或$(\mathscr{M}, s) \models \varphi_2$；
- $(\mathscr{M}, s) \models \neg\varphi$当且仅当$(\mathscr{M}, s) \not\models \varphi$；
- $(\mathscr{M}, s) \models \mathrm{EX}\varphi$当且仅当存在$S$中的一个状态$s_1$，使得$(s, s_1) \in R$且$(\mathscr{M}, s_1) \models \varphi$；
- $(\mathscr{M}, s) \models \mathrm{EG}\varphi$当且仅当存在$\mathscr{M}$上的一条路径$\pi_s = (s_1 = s, s_2, \ldots)$，使得对每一个$i \geq 1$都有$(\mathscr{M}, s_i) \models \varphi$；
- $(\mathscr{M}, s) \models \mathrm{E}(\varphi \mathrm{U} \psi)$当且仅当存在$\mathscr{M}$上的一条路径$\pi_s = (s_1 = s, s_2, \ldots)$，使得对某一个$i \geq 1$有$(\mathscr{M}, s_i) \models \psi$，同时对任意的$1 \leq j < i$有$(\mathscr{M}, s_j) \models \varphi$。

与Browne和Bolotov等人的工作类似，本文只将初始K-结构作为模型的候选项[57,59]。换句话说，对于给定的K-结构$(\mathscr{M}, s)$和CTL公式φ，如果$(\mathscr{M}, s) \models \varphi$且$s = s_0$，则称$(\mathscr{M}, s)$为

公式φ的一个模型。更清楚地说，对于给定的初始K-结构$\mathscr{K}=(\mathscr{M},s_0)$，如果$\mathscr{K}\models\varphi$，则称$\mathscr{K}$是$\varphi$的一个模型。

为了符号的统一，这里列出文中出现的一些记号的含义。给定公式φ，公式的所有模型构成的集合记为$Mod(\varphi)$。此时就很容易定义公式的可满足性，即：如果$Mod(\varphi)\neq\emptyset$，则称$\varphi$是可满足的。给定两个公式$\varphi_1$和$\varphi_2$，若$Mod(\varphi_1)\subseteq Mod(\varphi_2)$，则称$\varphi_1$逻辑地蕴涵$\varphi_2$，记为$\varphi_1\models\varphi_2$。特别地，当$\varphi_1\models\varphi_2$且$\varphi_2\models\varphi_1$时，即$Mod(\varphi_1)=Mod(\varphi_2)$，则称$\varphi_1$和$\varphi_2$为逻辑等值公式（简称为等值公式），记作$\varphi_1\equiv\varphi_2$。值得注意的是，上述的记号也适用于讨论的对象为公式的集合的情形。此外，给定一个公式的集合Π和一个初始K-结构$\mathscr{K}$，若对于Π中的任意一个公式φ都有$\mathscr{K}\models\varphi$，则$\mathscr{K}\models\Pi$。

对于给定的公式φ，将出现在φ中的原子命题的集合记为$Var(\varphi)$。此外，给定公式φ和原子命题的集合V，如果存在一个公式ψ使得$Var(\psi)\cap V=\emptyset$且$\varphi\equiv\psi$，那么说$\varphi$与$V$中的原子命题无关，简称为$V$-无关（$V$-*irrelevant*），写作$\mathrm{IR}(\varphi,V)$。一种特殊的形式是$Var(\varphi)\subseteq V$，此时称$\varphi$为集合$V$上的公式。可以类似定义公式的集合与原子命题集合的无关性，也即是：如果对于公式的集合Π中的任意一个公式φ，$\mathrm{IR}(\varphi,V)$都成立，则Π与V中的原子命题无关，记为$\mathrm{IR}(\Pi,V)$。

2.2.2 CTL的标准形式

在讲述CTL的标准形式之前，先引入一种带有索引的CTL，记为CTL_{ind}。这种语言是在CTL的已有符号下加入下面几种符号得到：

- 命题常量符号**start**；
- 一个可数无限的索引集Ind；
- 带有索引ind（$ind\in\mathrm{Ind}$）的时序算子：$\mathrm{E}_{\langle ind\rangle}\mathrm{X}$, $\mathrm{E}_{\langle ind\rangle}\mathrm{F}$, $\mathrm{E}_{\langle ind\rangle}\mathrm{G}$, $\mathrm{E}_{\langle ind\rangle}\mathrm{U}$, 和$\mathrm{E}_{\langle ind\rangle}\mathrm{W}$。

与CTL公式的定义类似，其公式可以递归地定义如下：

$$\phi ::= \bot \mid \top \mid p \mid \neg\phi \mid \phi\vee\phi \mid \mathrm{EX}\phi \mid \mathrm{EG}\phi \mid \mathrm{E}(\phi\ \mathrm{U}\ \phi) \mid \mathrm{E}_{ind}\mathrm{X}\phi \mid \mathrm{E}_{ind}\mathrm{G}\phi \mid \mathrm{E}_{ind}(\phi\ \mathrm{U}\ \phi) \quad (2.11)$$

与CTL不同的是，CTL_{ind}的语义定义在一种扩展的初始-Kripke结构上，该结构被称为Ind-Kripke结构。一个Ind-Kripke结构是一个五元组$\mathscr{M}=(S,R,L,[_],s_0)$，且除了$[_]$，其余元素都跟初始结构中的元素对应。该五元组中的$[_]$是一个以Ind为定义域，$2^{S\times S}$为值域的后继函数，即$[_]:\mathrm{Ind}\to 2^{S\times S}$，且满足对于任意的$s\in S$都存在唯一一个$s'\in S$使得$(s,s')\in[ind]\cap R$。记$\pi_{s_i}^{ind}$是$\mathscr{M}$上的一条路径$(s_i,s_{i+1},s_{i+2},\ldots)$，且对于任意的$j\geq i$都有：

$$(s_j,s_{j+1})\in[ind]$$

一个Ind-Kripke结构$\mathscr{M}$和其上的一个状态s构成一个*ind*-结构，记为$(\mathscr{M},s)$。同理，如果s是初始状态s_0，则称$(\mathscr{M},s_0)$为初始*ind*-结构。

令φ是一个CTL$_{ind}$公式、$\mathscr{K}=(\mathscr{M},s_0)$是一个初始*ind*-结构、且$s$是$\mathscr{M}$上的一个状态，则$\varphi$和$(\mathscr{M},s_0)$的可满足关系$(\mathscr{M},s_0)\models\varphi$被定义如下（这里只列出带有索引的公式的可满足关系，其余公式的可参加CTL部分的定义）：

- $(\mathscr{M},s)\models$ **start**当且仅当$s=s_0$；
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}\mathrm{X}\psi$当且仅当对于路径$\pi_s^{\langle ind\rangle}$，有$(\mathscr{M},s')\models\psi$且$(s,s')\in[ind]$;
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}\mathrm{G}\psi$当且仅当对于任意的$s'\in\pi_s^{\langle ind\rangle}$，$(\mathscr{M},s')\models\psi$;
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}(\psi_1\mathrm{U}\psi_2)$ 当且仅当存在路径$\pi_s^{\langle ind\rangle}=(s=s_1,s_2,\ldots)$上的状态$s_j$（$j\geq 1$），使得$(\mathscr{M},s_j)\models\psi_2$，且对于任意的$s_k\in\pi_s^{\langle ind\rangle}$，若$1\leq k<j$，则$(\mathscr{M},s_k)\models\psi_1$；
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}\mathrm{F}\psi$当且仅当$(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}(\top\mathrm{U}\psi)$；
- $(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}(\varphi\mathrm{W}\psi)$当且仅当$(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}\mathrm{G}\varphi$或$(\mathscr{M},s)\models \mathrm{E}_{\langle ind\rangle}(\varphi\mathrm{U}\psi)$。

对于给定的公式φ和初始*ind*-结构$\mathscr{K}=(\mathscr{M},s_0)$，如果$\mathscr{K}\models\varphi$，则称$\mathscr{K}$是$\varphi$的一个*Ind*-模型，也称$\mathscr{K}$满足$\varphi$。其他的术语与CTL部分的类似，这里不再赘述。

已有结果表明，任意的CTL公式能够在多项式时间内被转换为CTL的全局子句分离的范式（separated normal form with global clauses for CTL，SNF$^g_{\mathrm{CTL}}$子句）[60-61]。SNF$^g_{\mathrm{CTL}}$子句是具有下面几种形式的公式：

$\mathrm{A}\mathrm{G}(\textbf{start}\rightarrow\bigvee_{j=1}^{k}m_j)$	(初始句，initial clause)
$\mathrm{A}\mathrm{G}(\top\rightarrow\bigvee_{j=1}^{k}m_j)$	(全局子句，global clause)
$\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n}l_i\rightarrow\mathrm{A}\mathrm{X}\bigvee_{j=1}^{k}m_j)$	(A-步子句，A-step clause)
$\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n}l_i\rightarrow\mathrm{E}_{\langle ind\rangle}\mathrm{X}\bigvee_{j=1}^{k}m_j)$	(E-步子句，E-step clause)
$\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n}l_i\rightarrow\mathrm{A}\mathrm{F}l)$	(A-某时子句，A-sometime clause)
$\mathrm{A}\mathrm{G}(\bigwedge_{i=1}^{n}l_i\rightarrow\mathrm{E}_{\langle ind\rangle}\mathrm{F}l)$	(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（$1\leq i\leq n$）、m_j（$1\leq j\leq k$）和l都是文字，且$ind\in$ Ind。从上述标准形式中，可以看到每个SNF$^g_{\mathrm{CTL}}$子句都是$\mathrm{A}\mathrm{G}(P\rightarrow G)$形式。因此在没有歧义的情况下，下文中将使用$P\rightarrow G$指代这些子句。此外，除了额外说明，本文通常讲SNF$^g_{\mathrm{CTL}}$子句和子句统称为子句。

对于给定的公式φ（其中的$\rightarrow$符号都用$\vee$和$\neg$表示），如果φ中所有原子命题p的出现都有偶数个否定符号在其之前，则称φ关于p是正的，否则称φ关于p是负的。此外，

对于给定的公式集合，如果该集合中的所有公式关于p都是正的，则说该集合关于p是正的，否则该集合关于p是负的。

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

表 2.1: 转换规则

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

在表 2.1中，$T \in \{\mathrm{X}, \mathrm{G}, \mathrm{F}\}$，$ind$是规则中引入的新的索引且$Q \in \{\mathrm{A}, \mathrm{E}_{\langle ind\rangle}\}$；$q$是一个原子命题, l是一个文字, D是文字的吸取（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)**通过引入新的替换规则将复杂的公式用新的原子命题替换；规则**Trans(6)**到规则**Trans(12)**用于移除掉那些不能出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$中的时序操作符[62]。

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

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

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

(3) 使用表 2.1中的规则将$\{\mathrm{AG}(\mathbf{start} \to z), \mathrm{AG}(z \to simp(nnf(\varphi)))\}$化简为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合。

下面通过一个简单的例子[61]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$\varphi$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：$\mathrm{EG}\neg p \wedge \mathrm{AF}p$；

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

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

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

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

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

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

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

2.2.3 μ-演算

μ-演算是一种表达能力与S2S[2]相同的逻辑语言，LTL（线性时序逻辑，linear temporal logic）、CLT和CTL*能表达的属性都能用μ-演算来表示。μ-演算是模态逻辑的扩展，本文讨论Kozen提出的命题μ-演算[63]。构成μ-演算语言的符号有：

- 原子命题符号的集合：$\mathscr{A}$；

[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。

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

通常认为AX和EX的优先级比布尔连接符高[64]，为了保证文章的统一性，本文规定各类符号之间的如下优先级：

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

此时可如下定义μ-演算的公式：

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

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

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

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

与CTL公式类似，μ-演算公式（简称为μ-公式或公式）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）。

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

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

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

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

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

在下文中，若$s\in\|\varphi\|_v$，则称s“满足”φ，记为$(\mathscr{M},s,v)\models\varphi$。此时，若$(\mathscr{M},r,v)\models\varphi$，则称$(\mathscr{M},r,v)$为$\varphi$的一个模型。当公式$\varphi$为$\mu$-句子时，可以将赋值函数$v$省略，记为$(\mathscr{M},s)\models\varphi$。记$Mod(\varphi)$为$\varphi$的模型的集合。其他记号与CTL情形类似，这里不再赘述。

2.2.4　μ-公式的析取范式

Janin等人首先提出了μ-演算的析取范式[65]，后来被逐步完善，本文使用文章[66]中的析取μ-公式的定义。

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

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

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

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

尽管覆盖-语法在形式上与上一小节中μ-公式的定义大相径庭，但是已经证明这两种定义是等价的[66]。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[66]). 析取μ-公式的集合$\mathscr{F}_d$是包含$\top$、$\bot$和不矛盾的文字的合取且封闭于下面几条规则的最小集合:

(1) 吸取式（*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) 固定点操作（*fixpoint operators*）：若$\varphi\in\mathscr{F}_d$，且对任意的公式ψ，φ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。

2.3 CTL下的归结

归结是一种用于判定给定的命题公式（或一阶公式）是否可满足的规则，该技术可以追溯到1960年Davis等的工作[67]，之后被Robinson加以完善[8]。对于给定的公式，归结给出一个反驳定理证明过程。

在看见了归结在在命题逻辑和一阶逻辑中取得成就之后，科研工作者们开始将精力致力于其他非经典逻辑中，并取得了相当显著的理论成果，如：模态逻辑（K系统，Q系统，T系统，S4和S5系统）中的归结[68]和时态逻辑（尤其是线性时序逻辑（LTL）和CTL）中的归结[69-70]。

这里主要介绍与本文直接相关的CTL下的归结。CTL下的归结起源于BolotovF的研究[70]，之后被Zhang等人完善[61]。不论是在BolotovF的工作还是在Zhang等人的工作中，其关键点都是将CTL公式转换为一个$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合。本文使用Zhang等人在[61]中的规则，如表所示。

在表 2.3中P和Q是文字的合取，C和D是文字的吸取，l是一个文字。此外，$\Lambda=\bigvee_{i=1}^{n}\bigwedge_{i=1}^{m_i}P_j^i$和$P_j^i$是文字的吸取，且$1\leq i\leq n$和$1\leq j\leq m$。

规则**SRES1-8**被叫做步-归结规则（step resolution rule）、**RW1-2**被叫做重写规则（rewrite rule）、**ERES1-2**被叫做可能归结规则（eventuality resolution rule）。值得注意的

表 2.3: 归结规则

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

是，规则**ERES1** 的前提“$\Lambda \to \mathrm{EXEG}l$”表示如下子句的集合$\Lambda_{\mathrm{EG}}$：

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

其中，对任意的i $(1 \le i \le n)$，

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

上面的最后两个条件确保了子句集合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-步子句的集合，**ERES2**的结果则通过表 2.1中的规则转换成等价可满足的全局和E-步子句的集合。值得注意的是在转换**ERES1-2**的结果为子句集合的过程中会引入一个新的原子命题[61]。从这个角度来看，每个归结规则的前件和结果都是子句形式。

对于给定的CTL公式，使用上述的归结规则可以到处一个子句的集合。形式化地说，源于$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集合$S$的一个推导（derivation）是一个满足如下条件的$\mathrm{SNF}_{\mathrm{CTL}}^{g}$子句集合的序列$S_0, S_1, S_2, \ldots,$：

- $S_0 = S$，和

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

$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集合$S$的一个反驳是一个源于$S$的推导$S_0, S_1, S_2, \ldots, S_i$，且$S_i$（$i \geq 0$）中包含一个矛盾——公式$\top \to \bot$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[61]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$\varphi \models \psi$当且仅当当且仅当$T_{\varphi \wedge \neg \psi}$有一个反驳。*

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

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

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

2.4 遗忘理论基础和SNC（WSC）

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

2.4.1 经典逻辑下的遗忘

遗忘一词起源于经典逻辑（包括命题逻辑和一阶逻辑）[24]，给定一个命题公式φ和一个原子命题p，下面将介绍一下在φ中遗忘（forget）掉p。在1.2.1中，从φ中遗忘掉p 得到的结果为$Forget(\varphi, \{p\}) \equiv \varphi[p/\top] \vee \varphi[p/\bot]$。

例 2.4. 某学校有a和b两个食堂，学生要么去a食堂吃饭要么去b食堂吃饭，如果想吃烤鱼（fish，f）就去a食堂吃饭，如果想吃炒饭（rice，r）就去b食堂吃饭。这一知识可

表示为命题公式$\varphi = (a \vee b) \wedge (f \to a) \wedge (r \to b)$。如果此时不考虑鱼，即：由于某种原因$a$食堂就不在卖烤鱼了，此时就应该“遗忘”烤鱼。这一计算过程表示如下：

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

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

在一阶逻辑中，一阶逻辑语言$\mathscr{L}_f$的解释有两种：$\mathscr{L}_f$和结构有联系或没有联系，互模拟的定义就要困难一些。这里考虑和结构有联系的情形，一个一阶结构由论域（domain）、指定的个体、关系和函数构成。此时，$\mathscr{L}_f$中的个体符合、n-元关系符号和m-元函数符号分别被解释为这个结构中指定的论域中的个体、论域上的n-元关系和m-元全函数（即处处有定义的函数）。对于给定的一阶结构M和$X \in$ {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（论域（domain）、指定的个体、关系和函数构成一个一阶结构），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$\vec{d} \in M_2[P]$。

现在给出如下一阶逻辑中遗忘实例化原子的形式化定义：

定义 2.7 (定义1[24]). *给定一个句子（sentence）φ和实例化原子p，φ'是从φ中遗忘掉p的结果当且仅当对任意的结构M，M是φ'的模型当且仅当存在一个φ的模型M'使得$M \sim_p M'$。*

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

例 2.5. 令$\varphi = J(mo) \vee J(fa) \vee B(sm)$、$p = J(mo)$，则：

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

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

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

(i) M_1和M_2有相同的论域，且每个函数符号被解释成相同的函数；

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

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

由定理 8[24]可知，从句子φ中遗忘掉谓词P的结果为$Forget(\varphi, P) = (\exists R)\varphi[P/R]$，其中$P$是$n$-元谓词，$R$是$n$-元谓词变量。正如前面所说的，一阶逻辑下的遗忘不是封闭的，此时不一定能找到一个与$(\exists R)\varphi[P/R]$等价的一阶公式。

本文采用了基于归结的方法来计算CTL中的遗忘，因此这里给出命题逻辑下基于归结的遗忘定义[71]。

定义 2.8. 给定命题公式φ和原子命题p，

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

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

从定义2.8不难看出计算从φ中遗忘p的结果可以分为三个步骤：（1）计算φ的合取范式，并得到$CNF(\varphi)$；（2）计算$Res(CNF(\varphi), p)$；（3）去除$CNF(\varphi)$包含p的子句。遗忘的定义种类很多，本文的定义采用的是上述所说的互模拟方式，因此这里不再赘述其他定义，感兴趣的读者可以参考Eiter的文章[47]。

在描述逻辑中，如果遗忘的结果是可以用当前讨论的描述逻辑来表示的，则该结果就是一个均匀插值。而判定均匀插值的存在性通常是很费时的，如：在$\mathscr{ALC}$和$\mathscr{EL}$中是双指数时间的。因此，不难看出描述逻辑中的遗忘通常也是很困难的，尽管如此也有很多方法克服这些问题，其中扩展描述语言（如：从$\mathscr{ALC}$到$\mathscr{ALC}_v$[72]）或引入新的辅助符号[73]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[74]和基于Ackermann引理的FAME[75]。

2.4.2　模态逻辑S5里的遗忘

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

原子命题的集合w_1和w_2是V-互模拟的，当且仅当$w_1 - V = w_2 - V$，记为$w_1 \sim_V w_2$，其中$w_1, w_2, V \subseteq \mathscr{A}$。给定原子命题的集合$V \subseteq \mathscr{A}$、两个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$\mathscr{M} \leftrightarrow_V \mathscr{M}'$）[36]，当且仅当存在一个二元关系$\sigma \subseteq W \times W'$使得$(w, w') \in \sigma$，且：

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

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

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

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

S5关于V-互模拟是不变的（invariant）：如果两个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。

定义 2.9 (定义1[36], knowledge forgetting). 给定模态$S5$公司φ和$V \subseteq \mathscr{A}$。如果下面的等式成立，则称知识集$KForget(\varphi, V)$是从φ遗忘掉V得到的结果：

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

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

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

(W) 弱性质（Weaking）：$\varphi \models \varphi'$；

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

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

(IR) 无关性（Irrelevance）：IR(φ', V)。

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

这四个条件与知识遗忘的关系为：

定理 2.1 (定理1[36]). *给定CTL公式φ和φ'，$V \subseteq \mathscr{A}$为原子命题的集合。下面的陈述是等价的：*

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

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

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

在本文中也将说明CTL和μ-演算的遗忘也有上述性质。值得注意的是任意的S5公式都能转换为与之等价的模态合取范式（MCNF）[76]，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吸取——模态析取范式（MDNF）[36,77]:

$$\varphi_0 \wedge \mathrm{K}\varphi_1 \wedge \mathrm{B}\varphi_2 \wedge \cdots \wedge \mathrm{B}\varphi_n \tag{2.12}$$

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都可能缺失。从子句2.12遗忘掉原子命题p可以转换成命题逻辑中的遗忘，即：

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

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

当然，模态一阶逻辑S5中的遗忘也得到了研究[77]，由于本文只考虑命题情形下时序逻辑的遗忘，就不再赘述。此外，Fang等人也讨论了关于多模态（multi-modal）K_n、D_n、T_n、$K45_n$、$KD45_n$情形下的遗忘的存在性[78]——这些逻辑里的遗忘总是存在的，其中n为智能体的个数。

2.4.3　遗忘的计算方法

在第 2.4.1和2.4.2小节中详细介绍了经典逻辑和模态逻辑S5下遗忘的定义和一些直接的计算方法。总的来说，这些方法分为两类：“代替”的方法和归结的方法。其中“代替”法是将要遗忘的原子命题在公式里用“$\top$”或“$\bot$”代替，归结的方法主要使用归结规则来“消除”需要遗忘的原子命题。然而，在上文中并没有对归结方法进行详细的介绍，所以这部分给出命题情形下归结方法的详细描述。

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

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

其中C_1和C_2是子句，p是原子命题。在这种情况下，基于归结的方法就如定义 2.8那样简单。在一阶逻辑中，将公式转换为子句形式的过程比较复杂，而归结规则也相对复杂一些。但是在一阶情形下归结，归结系统R[56]是可靠的且归结反驳是完备的。

在上一节中已经说明任意的S5公式能够转化成模态子句$C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n$的合取，因此S5下的归结系统RS5[68]如下：

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

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

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

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

$$(\mathrm{K}\bot)\frac{C \vee \mathrm{K}\bot}{C};$$

$$(\mathrm{B}\bot)\frac{C \vee \mathrm{B}(\bot, E)}{C};$$

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

$$(Fact)\frac{E[D \vee D \vee C])}{E[D \vee C]}.$$

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

Feng等人提出了基于上述归结系统RS5的计算遗忘的算法[79]，为了更清楚地描述该算法，这里还需要介绍两个概念：模态子句的包蕴（subsume）和清除

（suppressing）。给定两个模态子句$C = C_0 \vee \mathrm{K}C_1 \vee \cdots \vee \mathrm{K}C_{n-1} \vee \mathrm{B}C_n$和$C' = C'_0 \vee \mathrm{K}C'_1 \vee \cdots \vee \mathrm{K}C'_{m-1} \vee \mathrm{B}C'_m$，如果下面三个条件满足，则说$C$包蕴$C'$：

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

其中$Lit(X)$为出现在X中文字的集合。

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

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

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$V$上的清楚操作也记为$Supp(V,\phi)$，且：

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

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。

2.4.4 基于遗忘的SNC（WSC）计算

SNC和WSC的定义最先由Lin提出[80]，这部分给出其在命题逻辑和一阶逻辑下的形式化定义和计算方法。

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

由定命题5和3[80]分别可知SNC和WSC是一个对偶概念，且任意公式的SNC（WSC）都能转换称原子命题的形式计算，因此这里只讨论原子命题情形下的SNC的定义及其计算。

算法 2.1 S5下基于归结的遗忘计算[79]

输入:

Γ, V：　S5公式，原子命题的集合

输出:

$KForget(\Gamma, V)$：从Γ中遗忘掉V中原子的结果

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

定理 2.2 (定理2[80]). *给定命题公式φ、原子命题集合$V \subseteq Var(\varphi)$和原子命题$q \in (Var(\varphi) - V)$。令$V' = Var(\varphi) - (V \cup \{q\})$，则*

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

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

- 所有的单元子句都被替换为true;
- 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每个公式中X的出现全都替换成Y，即$S[X/Y] = \{\varphi[X/Y] \mid \varphi \in S\}$。

在一阶逻辑中，SNC和WSC的定义和命题逻辑下相似，且也可用遗忘来计算。但是一阶逻辑中的遗忘计算比较麻烦，且遗忘的结果不一定能用一阶语言表示出来。

定理 2.3 (引理4.1[53]). *对任意一阶公式α关系符号的集合P和句子Th:*

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

算法 2.2 命题逻辑下基于遗忘的SNC计算[80]

输入:

Γ, V, q：子句集合，原子命题的集合，出现在Γ且不出现在V中的原子命题

输出:

ϕ： V上的公式（ϕ是q在集合V和Γ上的最强必要条件）

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

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

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

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

在模态逻辑S5中，SNC和WSC也可以通过遗忘来计算。

定理 2.4 (定理4.1[79]). 给定$S5$公式Γ和原子命题集合$V \subseteq \mathscr{A}$，$q \in Var(\Gamma) - V$，则：

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

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

此时，由算法 2.1不难得出计算SNC和WSC的算法，这里就不在赘述。

2.5 本章小结

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 围绕本文的研究工作，本章首先介绍了隐私以及隐私泄露的定义，明确了本文中隐私保护的研究范畴，随后，介绍了差分隐私模型并给出标准形式的定义。其次，介绍了本文研究需要的Shannon信息论知识，包括基本通信模型、信息熵、条件熵、联合熵、互信息量等概

念。在此基础上，对信息论中两个重要的不等式和率失真理论进行了简要的叙述。进一步，介绍了本文将使用的优化理论知识以及在对策博弈模型中的应用。最后，结合本章内容，给定了本文中所研究的差分隐私均衡优化的定义。针对差分隐私模型中隐私保护与数据可用性之间的矛盾问题，利用均衡优化思想研究差分隐私最优化机制是本文研究的核心。本章中所介绍内容为后续章节提供了基本模型与定义，是开展后续研究工作的理论出发点。

第三章　遗忘理论的定义及其语义属性

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

3.1　引言

从一个公式中“遗忘”掉一些原子命题得到的结果应该不违背定义在剩余原子命题集合上的公式，也就是说对于剩余原子命题集合上的公式，原始公式能够逻辑蕴涵它当且仅当遗忘得到的结果能过逻辑蕴涵它。从模型的角度来讲，遗忘得到的结果的模型与原始公式的模型在除去被遗忘的那些原子命题之后是能够想互模拟的。互模拟描述的是两个在行为上能够相互替代的转换系统[19]。在本文中，转换系统被描述成为Kripke结构。因此为了描述遗忘理论，这部分给出在给定原子命题集合上的Kripke结构（或Ind-结构）上的互模拟的定义及其性质。

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

3.2　V-互模拟

这部分给出定义在给定原子命题集合V上的互模拟的概念，本文称之为V-互模拟。尽管在文章[35]中给出了相似的概念，但是如在基础知识部分所述，$S5$的语义是定义在一种特殊的Kripke结构（K-解释）下的，因而不具有一般性。接下来探讨一种更加一般的V-互模拟。

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

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

– 对任意$(s_2,s_2')\in R_2$，存在$(s_1,s_1')\in R_1$使得$(\mathscr{K}_1',\mathscr{K}_2')\in\mathscr{B}_n^V$。

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

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

定义 3.1 (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K-结构（或*Ind*-结构）。

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

上述V-互模拟的定义是现有互模拟定义的一般化，这可以从下面几个方面来体现[2]。首先，当给定的V为空集且谈论指定的初始状态时，本文的V-互模拟与定义在Baier等文章里的互模拟等价（定义7.1[19]）的概念一致。其次，在同一文章里的基于状态的互模拟（定义7.7[19]）是定义在给定结构的状态上的，因此与本文的V-互模拟（定义在结构的集合上）也不同。最后，本文的$\mathscr{B}_n$的定义与Browne的论文中的状态等价E_n类似，只是后者是定义在状态上[57]而本文的定义在K-结构（或Ind-结构）上。

下面例子呈现结构之间的V-互模拟。

例 3.1. 令$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$为三个K-结构，其图表示分别如图中的$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$所示。它们之间的互模拟关系也如图中标记所示，即$\mathscr{K}_1\leftrightarrow_{\{sp\}}\mathscr{K}_2$，$\mathscr{K}_2\leftrightarrow_{\{se\}}\mathscr{K}_3$和$\mathscr{K}_1\leftrightarrow_{\{sp,se\}}\mathscr{K}_3$。此外，可以看出$\mathscr{K}_1$，$\mathscr{K}_2$和$\mathscr{K}_3$之间是互不互模拟[19]的，即不$\emptyset$-互模拟。

为了简化书写和看起来简洁，当从上下文中能明确知道所指的初始结构和Ind-Kripke结构（为了方便，将这两种结构通称为结构），则可以使用状态来表示这些结构之间的V-互模拟，即使用$s_1\leftrightarrow_V s_2$替代$(\mathscr{M}_1,s_1)\leftrightarrow(\mathscr{M}_2,s_2)$。

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

命题 3.1. 令i是属于集合$\{1,2\}$的变量，V_1和V_2是$\mathscr{A}$的子集，s_1'和s_2'是两个状态，π_1'和π_2'是两条路径，$\mathscr{K}_j=(\mathscr{M}_j,s_j)(j=1,2,3)$是K-结构（或*Ind*-结构）。如果$(\mathscr{K}_1\leftrightarrow_{V_1}\mathscr{K}_2)$且$(\mathscr{K}_2\leftrightarrow_{V_2}\mathscr{K}_3)$，则：

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

[2] 在其他领域也有类似的定义，如：定义在数据库相关文献中的概念k-互模拟[81]。k-互模拟概念中涉及与本文$\mathscr{B}_n$类似的定义，只是其关系是从相反的方向（即：从孩子到父节点的方向）来说明的。此外，值得一提的是，本文的V-互模拟的概念是定义在K-结构上的。

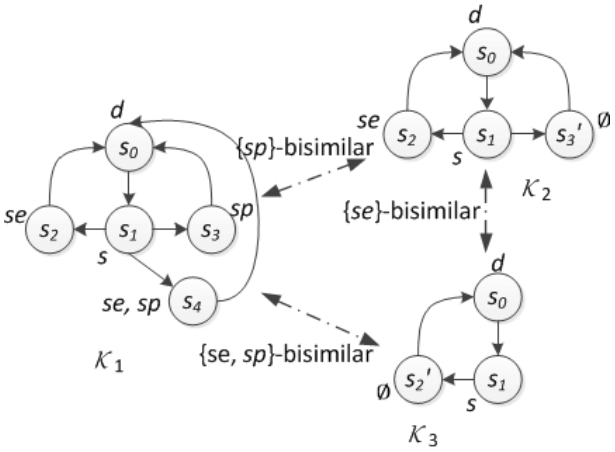


图 3.1: κ-结构之间的V-互模拟关系

(i) 若$s_1' \leftrightarrow_{V_i} s_2'$，则$s_1' \leftrightarrow_{V_1 \cup V_2} s_2'$；

(ii) 若$\pi_1' \leftrightarrow_{V_i} \pi_2'$，则$\pi_1' \leftrightarrow_{V_1 \cup V_2} \pi_2'$；

(iii) 对$\mathscr{M}_1$上的任意一条路径π_{s_1}，存在$\mathscr{M}_2$上的一条路径π_{s_2}使得$\pi_{s_1} \leftrightarrow_{V_1} \pi_{s_2}$；反之亦然；

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

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

证明. 这里给出结构为κ-结构的证明，结构为Ind-结构的情形可以类似地证明。

(i) 假设s_1'和s_2'分别来源于$\mathscr{M}_1$和$\mathscr{M}_2$，$\mathscr{K}_1 = (\mathscr{M}_1, s_1)$，$\mathscr{K}_2 = (\mathscr{M}_2, s_2)$。为了证明$s_1' \leftrightarrow_{V_1 \cup V_2} s_2'$，只需证明对于任意的$n \geq 0$，$(s_1', s_2') \in \mathscr{B}_n^{V_1 \cup V_2}$。

基始. 当$n = 0$时显然是成立的。

归纳步骤. 假设对于任意的$0 \leq k \leq n$，若$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_k^{V_1}$且$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_k^{V_2}$，则$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_k^{V_1 \cup V_2}$。这里根据定义 3.1证明对于若$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_{n+1}^{V_1}$且$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_{n+1}^{V_2}$，则$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_{n+1}^{V_1 \cup V_2}$。

（a）$L_1(s_1) - (V_1 \cup V_2) = L_2(s_2) - (V_1 \cup V_2)$是显然成立的。

（b）给定$i \in \{1,2\}$，$0 < k \leq n+1$，$\mathscr{K}_i^k = (\mathscr{M}_i, s_i^k)$。这里将证明对于任意的$(s_1, s_1^1) \in R_1$，存在一个$(s_2, s_2^1) \in R_2$使得$(s_1^1, s_2^1) \in \mathscr{B}_n^{V_1 \cup V_2}$。有归纳假设可知$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_n^{V_1 \cup V_2}$，因而有$(\mathscr{K}_1^1, \mathscr{K}_2^1) \in \mathscr{B}_{n-1}^{V_1 \cup V_2}$。因此，只需证明对于任意的$(s_1^n, s_1^{n+1}) \in R_1$，存在$(s_2^n, s_2^{n+1}) \in R_2$使得$(s_1^{n+1}, s_2^{n+1}) \in \mathscr{B}_0^{V_1 \cup V_2}$。因为$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_{n+1}^{V_1}$且$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_{n+1}^{V_2}$，所以有$L_1(s_1^{n+1}) - (V_1 \cup V_2) = L_1(s_2^{n+1}) - (V_1 \cup V_2)$。

（c）定义中的第三点可以类似（b）证明。

因此，当令s_1和s_2分别为s_1'和s_2'，就可得到$s_1' \leftrightarrow_{V_1\cup V_2} s_2'$。

(ii) 根据定义和(i)很容易证明。

(iii) 为了证明该结论成立，下面给出一个等价的V-互模拟的定义。令V是$\mathscr{A}$的一个子集，$\mathscr{K}_i = (\mathscr{M}_i, s_i)$ $(i = 1,2)$是κ(Ind)-结构，则$\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$（也记为$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}$）当且仅当

(a) $L_1(s_1) - V = L_2(s_2) - V$；

(b) 对任意的$(s_1, s_1') \in R_1$，存在一个$(s_2, s_2') \in R_2$使得$\mathscr{K}_1' \leftrightarrow_V \mathscr{K}_2'$；

(c) 对任意的$(s_2, s_2') \in R_2$，存在一个$(s_1, s_1') \in R_1$使得$\mathscr{K}_1' \leftrightarrow_V \mathscr{K}_2'$。

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

下面从两个方面证明上述结论成立。

($\Rightarrow$) (a) 显然$L_1(s_1) - V = L_2(s_2) - V$成立。

(b) $\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$当且仅当对于所有的$n \geq 0$都有$(\mathscr{K}_1, \mathscr{K}_2) \in \mathscr{B}_n$。因此，对任意的$(s_1, s_1') \in R_1$，存在一个$(s_2, s_2') \in R_2$使得对于任意的$n > 0$都有$(\mathscr{K}_1', \mathscr{K}_2') \in \mathscr{B}_n$且$L_1(s_1') - V = L_2(s_2') - V$。因此，$\mathscr{K}_1' \leftrightarrow_V \mathscr{K}_2'$。

(c) 这点的证明与(b)的证明类似。

($\Leftarrow$) 显然，$L_1(s_1) - V = L_2(s_2) - V$蕴涵$(s_1, s_2) \in \mathscr{B}_0$。(b)蕴涵对于任意的$(s_1, s_1') \in R_1$，存在一个$(s_2, s_2') \in R_2$使得对于任意的$n \geq 0$都有$(s_1', s_2') \in \mathscr{B}_n$。(c) 蕴涵对于任意的$(s_2, s_2') \in R_2$，存在一个$(s_1, s_1') \in R_1$使得对于任意的$n \geq 0$都有$(s_1', s_2') \in \mathscr{B}_n$。因此，对于任意的$n \geq 0$都有$(s_1, s_2) \in \mathscr{B}_n$。如此就可以知道$\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$。

(iv) 令$\mathscr{M}_j = (S_j, R_j, L_j, s_j)$ $(j = 1,2,3)$，$\mathscr{B}$是有V_1-互模拟关系的结构的集合（即：$s_1 \leftrightarrow_{V_1} s_2$当且仅当$(s_1, s_2) \in \mathscr{B}$），$\mathscr{B}''$是有$V_2$-互模拟关系的结构的集合。

集合$\mathscr{B}'$是由$\mathscr{B}$和$\mathscr{B}''$共同约束下得到的，即$\mathscr{B}' = \{(w_1, w_3) \mid (w_1, w_2) \in \mathscr{B}$且$(w_2, w_3) \in \mathscr{B}''\}$。显然$(s_1, s_3) \in \mathscr{B}'$成立。为了证明命题中的结论成立，下面从(iii)中的(a)，(b)和(c)三个方面来证明$\mathscr{B}'$是有$(V_1 \cup V_2)$-互模拟关系的结构的集合。

对于所有的$(w_1, w_3) \in \mathscr{B}'$：

(a) 存在S_2中的一个状态w_2使得$(w_1, w_2) \in \mathscr{B}$且$(w_2, w_3) \in \mathscr{B}''$。此外，由$(w_1, w_2) \in \mathscr{B}$可知对于任何不在$V_1$中的原子命题$q$，有$q \in L_1(w_1)$当且仅当$q \in L_2(w_2)$；由$(w_2, w_3) \in \mathscr{B}''$可知对于任何不在$V_2$中的原子命题$q'$，有$q' \in L_2(w_2)$当且仅当$q' \in L_3(w_3)$。因此可以得知，对于任意不在$V_1 \cup V_2$中的原子命题$r$，有$r \in L_1(w_1)$当且仅当$r \in L_3(w_3)$，即：$L_1(s_1) - (V_1 \cup V_2) = L_3(s_3) - (V_1 \cup V_2)$。

(b) 若$(w_1,u_1)\in R_1$，根据$\mathscr{B}'$的定义可知存在$w_2\in S_2$使得$(w_1,w_2)\in\mathscr{B}$和$(w_2,w_3)\in\mathscr{B}''$，则存在$u_2\in S_2$使得$(w_2,u_2)\in R_2$和$(u_1,u_2)\in\mathscr{B}$成立。因此，存在一个$u_3\in S_3$使得$(w_3,u_3)\in R_3$和$(u_2,u_3)\in\mathscr{B}''$成立。因此，有$(u_1,u_3)\in\mathscr{B}'$成立。

(c) 与(b)的证明类似，可以证明对于任意$(w_3,u_3)\in R_3$，存在$(w_1,u_1)\in R_1$使得$(u_1,u_3)\in\mathscr{B}'$。

因此有$\mathscr{K}_1\leftrightarrow_{V_1\cup V_2}\mathscr{K}_3$成立。

(v) 为了证明此结论，这里假定x和k是大于等于0的整数，$\mathscr{K}_{i,x}=(\mathscr{M}_i,s_{i,x})$是结构。为了方便，用$(s_{i,k},s_{i,k+1})\in R_i$表示路径$(s_i=s_{i,0},s_{i,1},s_{i,2},\ldots,s_{i,k},s_{i,k+1},\ldots)$上的第$k+2$个节点。接下来将展示对任意的$n\geq 0$都有$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_n^{V_2}$。

基始. $L_1(s_1)-V_1=L_2(s_2)-V_1$ （已知）
$\Rightarrow$ 对任意的$q\in(\mathscr{A}-V_1)$，有$q\in L_1(s_1)$当且仅当$q\in L_2(s_2)$
$\Rightarrow$ 对任意的$q\in(\mathscr{A}-V_2)$，有$q\in L_1(s_1)$当且仅当$q\in L_2(s_2)$ （$V_1\subseteq V_2$）
$\Rightarrow L_1(s_1)-V_2=L_2(s_2)-V_2$，即：$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_0^{V_2}$。

归纳步骤. 假定对于任意的$0\leq n\leq k$ $(k>0)$，若$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_n^{V_1}$则$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_n^{V_2}$成立，接下来证明若$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_k^{V_1}$则$(\mathscr{K}_1,\mathscr{K}_2)\in\mathscr{B}_{k+1}^{V_2}$。

(a) 显然$L_1(s_1)-V_2=L_2(s_2)-V_2$成立。

(b) 对于任意的$(s_1,s_{1,1})\in R_1$，下面证明存在一个$(s_2,s_{2,1})\in R_2$使得$(\mathscr{K}_{1,1},\mathscr{K}_{2,1})\in\mathscr{B}_k^{V_2}$。由归纳假设可知$(\mathscr{K}_{1,1},\mathscr{K}_{2,1})\in\mathscr{B}_{k-1}^{V_2}$，因而只需要证明：① 对于所有的$(s_{1,k},s_{1,k+1})\in R_1$存在一个$(s_{2,k},s_{2,k+1})\in R_2$使得$(\mathscr{K}_{1,k+1},\mathscr{K}_{2,k+1})\in\mathscr{B}_0^{V_2}$由于$(\mathscr{K}_{1,1},\mathscr{K}_{2,1})\in\mathscr{B}_k^{V_1}$。因此，类似与基始里的证明方法，可得$L_1(s_{1,k+1})-V_2=L_2(s_{2,k+1})-V_2$成立，即：$(\mathscr{K}_{1,k+1},\mathscr{K}_{2,k+1})\in\mathscr{B}_0^{V_2}$。② 可以类似①证明对任意的$(s_{2,k},s_{2,k+1})\in R_1$存在一个$(s_{1,k},s_{1,k+1})\in R_2$使得$(\mathscr{K}_{1,k+1},\mathscr{K}_{2,k+1})\in\mathscr{B}_0^{V_2}$。

(c) 对于任意的$(s_2,s_{2,1})\in R_2$，类似(b)可以证明存在$(s_1,s_{1,1})\in R_1$使得$(\mathscr{K}_{1,1},\mathscr{K}_{2,1})\in\mathscr{B}_k^{V_2}$。 □

在命题 3.1中，性质(i)-(iii)是V-互模拟的标准属性，含义比较直观。性质(iv)表示如果一个结构分别与另外的两个结构具有V_1和V_2-互模拟关系，则这两个结构是$V_1\cup V_2$-互模拟的（如图 4.1所示）。如后文所示，这一性质对遗忘理论性质的探索至关重要。性质(v)表示若两个结构是V_1-互模拟的，则对于任意的V_2，若$V_1\subseteq V_2$则这两个结构是V_2-互模拟的。

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

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

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

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

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

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

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

情形2：$\Phi=\neg\psi$.

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

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

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

情形3：$\Phi=\psi_1\vee\psi_2$.

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

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

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

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

情形4：$\Phi=\mathrm{EX}\psi$.

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

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

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

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

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

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

情形5：$\Phi=\mathrm{EG}\psi$.

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

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

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

$\Leftrightarrow$ 对于任意的$i\geq 0$都有$s_i\leftrightarrow_V s'_i$ ($\pi\leftrightarrow_V\pi'$)

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

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

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

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

$\Leftrightarrow$ 存在一条路径$\pi=(s=s_0,s_1,\ldots)$和$i\geq 0$使得$(\mathscr{M},s_i)\models\psi_2$，且对所有的$0\leq j<i$都

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

例 3.2. 令$\varphi_1=d\wedge\mathrm{EF}se\wedge\mathrm{AG}(se\rightarrow\mathrm{AX}d)$和$\varphi_2=d\wedge\mathrm{AX}se$是两个CTL公式，且$\mathrm{IR}(\varphi_1,\{sp\})$和$\mathrm{IR}(\varphi_2,\{sp\})$成立。因此可以验证图 4.1中的$\mathscr{K}_1$和$\mathscr{K}_2$都满足$\varphi_1$，但是都不满足$\varphi_2$。

3.3 遗忘理论及其语义属性

这部分将给出CTL下的遗忘理论的定义及其相关属性。

定义 3.2 (遗忘理论). 令V是$\mathscr{A}$的一个子集，Φ是一个公式。如果一个公式ψ满足下面条件，则称ψ为从Φ中遗忘掉V后得到结果，记为$\mathrm{F}_{\mathrm{CTL}}(\Phi,V)$：

- ψ与V中原子命题无关（即：$Var(\psi)\cap V=\emptyset$）；
- $Mod(\psi)=\{\mathscr{K}\mid\mathscr{K}$是一个初始K-结构$,\exists\mathscr{K}'\in Mod(\phi)\ \ s.t.\ \ \mathscr{K}'\leftrightarrow_V\mathscr{K}\}$

从定义 3.2可以看出，如果有两个公式ψ和ψ'都是从Φ中遗忘掉V中元素后得到的结果，则有$\psi\equiv\psi'$。从这个角度来看，可以说从Φ中遗忘掉V中元素后得到的结果之间是语义等价的。此外，当V中只包含一个元素的时候，可以省略掉集合符号，即：$\mathrm{F}_{\mathrm{CTL}}(\Phi,\{p\})\equiv\mathrm{F}_{\mathrm{CTL}}(\Phi,p)$。值得指出的是，遗忘理论的定义与均匀插值的语义定义等价，也即是遗忘理论与均匀插值是一对对偶概念，这与其他逻辑（包括模态逻辑S4、S5和经典命题逻辑）中的说法一致[?]。

从命题公式φ中遗忘掉原子命题p得到的结果记为：$Forget(\varphi,\{p\})\equiv\varphi[p/\bot]\vee\varphi[p/\top]$。值得注意的是，本文的遗忘理论的定义与Lin等人于1994提出命题逻辑下的遗忘理论一致。换句话说，本文将命题逻辑下的遗忘理论扩展到了CTL下。下面命题展示了上述结论：

定理 3.2. 给定一个命题公式φ和原子命题的集合$V\subseteq\mathscr{A}$，则下面逻辑等式成立。

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

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

一方面，对于$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的任意一个模型$(\mathscr{M},s)$，由遗忘理论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$(s,s')\in\mathscr{B}_0$，这意味着$(\mathscr{M},s)\models Forget(\varphi,V)$。

另一方面，对于$Forget(\varphi,V)$的任意一个模型$(\mathscr{M},s)$（$\mathscr{M}=(S,R,L,s)$），存在一个φ的模型$(\mathscr{M}',s')$（$\mathscr{M}'=(S',R',L',s')$）使得$(s,s')\in\mathscr{B}_0$。此时可以构建一个初始K-结构$(\mathscr{M}_1,s_1)$使得$\mathscr{M}_1=(S_1,R_1,L_1,s_1)$，其中：

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

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

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

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

引理 3.1. 给定两个公式φ和α，且$q\in\overline{Var(\varphi)\cup Var(\alpha)}$。则$\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge(q\leftrightarrow\alpha),q)\equiv\varphi$。

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

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

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

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

除了上述性质，遗忘理论还有其他一些一般属性。下面将详细介绍这些属性。

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

命题 3.2 (Modularity). 对于给定的公式φ，原子命题集合V，和原子命题p（$p \notin V$），下面的结论成立：

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

证明. 要证明上述结论成立，只需证明等式左右两边的公式有相同的模型。

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

(1) 对于s_2情形：令s_2是满足下面条件的状态：

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

(2) 其他情形：

- 对于所有的满足$w \in S$，$w_1 \in S_1$且$w \leftrightarrow_{\{p\} \cup V} w_1$的状态对$(w, w_1)$，如下构造$w_2 \in S_2$：
  - $p \in L_2(w_2)$当且仅当$p \in L_1(w_1)$，
  - 对于任意的$q \in V$，$q \in L_2(w_2)$当且仅当$q \in L(w)$，
  - 对于其他的原子命题q'，$q' \in L_2(w_2)$当且仅当$q' \in L_1(w_1)$当且仅当$q' \in L(w)$。
- 对于$(w_1', w_1) \in R_1$，若w_2是基于w_1构造而成，且w_2'是基于w_1'构造而成，则令$(w_2', w_2) \in R_2$。

(3) 删除掉S_2和R_2中重复的元素。

□

不难看出，从公式中遗忘掉原子命题的集合中的元素，可以将该集合拆成两个集合后遗忘。

推论 3.1. 对于给定的公式φ，原子命题集合V_1和V_2，下面的结论成立：

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

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

命题 3.3. 令$V \subseteq \mathscr{A}$为原子命题的集合，ϕ为*CTL*公式，则下面等式成立：

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

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

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

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

证明. 为了证明上述结论成立，这里新引入一个叫做“子结构”的概念。对于给定的初始结构$\mathscr{M} = (S, R, L, s_0)$，称满足下面约束的初始K-结构$(\mathscr{M}', s_0')$为$(\mathscr{M}, s_0)$的一个子结构：

- $S' \subseteq S$且$S' = \{s' \mid s'$ 从s_0'是可达的$\}$，
- $R' = \{(s_1, s_2) \mid s_1, s_2 \in S'$ 和$(s_1, s_2) \in R\}$，
- $L' : S' \to 2^{\mathscr{A}}$且对于所有的$s_1 \in S'$有$L'(s_1) = L(s_1)$，且
- s_0'要么是s_0本身，要么是从s_0可达的某一个状态。

(i) 要证明$\mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V) \equiv \mathrm{AXF}_{\mathrm{CTL}}(\phi, V)$，只需证明：

$$Mod(\mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V)) = Mod(\mathrm{AXF}_{\mathrm{CTL}}(\phi, V)).$$

($\Rightarrow$) 对于任意$\mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi, V)$的模型$(\mathscr{M}', s')$，存在一个初始K-结构$(\mathscr{M}, s)$使得$(\mathscr{M}, s) \models \mathrm{AX}\phi$且$(\mathscr{M}, s) \leftrightarrow_V (\mathscr{M}', s')$
$\Rightarrow$ 对任意$(\mathscr{M}, s)$的子结构$(\mathscr{M}_1, s_1)$（其中$(s, s_1) \in R$），$(\mathscr{M}_1, s_1) \models \phi$
$\Rightarrow$ 存在一个初始K-结构$(\mathscr{M}_2, s_2)$使得$(\mathscr{M}_2, s_2) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$且$(\mathscr{M}_2, s_2) \leftrightarrow_V (\mathscr{M}_1, s_1)$
$\Rightarrow$ 由这些$(\mathscr{M}_2, s_2)$容易构造出一个初始K-结构$(\mathscr{M}_3, s_3)$，使得s_2是s_3的直接后继状态，$(\mathscr{M}_2, s_2)$是$(\mathscr{M}_3, s_3)$的子结构且$(\mathscr{M}_3, s_3) \leftrightarrow_V (\mathscr{M}, s)$
$\Rightarrow (\mathscr{M}_3, s_3) \models \mathrm{AX}(\mathrm{F}_{\mathrm{CTL}}(\phi, V))$且$(\mathscr{M}_3, s_3) \leftrightarrow_V (\mathscr{M}', s')$
$\Rightarrow (\mathscr{M}', s') \models \mathrm{AX}(\mathrm{F}_{\mathrm{CTL}}(\phi, V))$（由定理 3.1）

($\Leftarrow$) 令$(\mathscr{M}_3, s_3)$是$\mathrm{AX}(\mathrm{F}_{\mathrm{CTL}}(\phi, V))$的模型，则对于任意$(\mathscr{M}_3, s_3)$的子结构$(\mathscr{M}_2, s_2)$（其中$(s_3, s_2) \in R_3$）有$(\mathscr{M}_2, s_2) \models \mathrm{F}_{\mathrm{CTL}}(\phi, V)$

$\Rightarrow$ 对任意上述的$(\mathscr{M}_2,s_2)$，存在一个初始K-结构$(\mathscr{M}_1,s_1)$使得$(\mathscr{M}_1,s_1)\models\phi$且$(\mathscr{M}_1,s_1)\leftrightarrow_V(\mathscr{M}_2,s_2)$

$\Rightarrow$ 由这些$(\mathscr{M}_1,s_1)$容易构造出一个初始K-结构$(\mathscr{M},s)$，使得s_1是s的直接后继状态，$(\mathscr{M}_1,s_1)$是$(\mathscr{M},s)$的子结构且$(\mathscr{M}_3,s_3)\leftrightarrow_V(\mathscr{M},s)$

$\Rightarrow(\mathscr{M},s)\models \mathrm{AX}\phi$且$(\mathscr{M}_3,s_3)\models \mathrm{F}_{\mathrm{CTL}}(\mathrm{AX}\phi,V)$。

(ii) 为了证明$\mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi,V)\equiv\mathrm{EXF}_{\mathrm{CTL}}(\phi,V)$，只需证明：

$$Mod(\mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi,V))=Mod(\mathrm{EXF}_{\mathrm{CTL}}(\phi,V)).$$

($\Rightarrow$) 对于任意$\mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi,V)$的模型$(\mathscr{M}',s')$，存在一个初始K-结构$(\mathscr{M},s)$使得$(\mathscr{M},s)\models\mathrm{AX}\phi$且$(\mathscr{M},s)\leftrightarrow_V(\mathscr{M}',s')$

$\Rightarrow$ 存在$(\mathscr{M},s)$的子结构$(\mathscr{M}_1,s_1)$（其中$(s,s_1)\in R$），$(\mathscr{M}_1,s_1)\models\phi$

$\Rightarrow$ 存在一个初始K-结构$(\mathscr{M}_2,s_2)$使得$(\mathscr{M}_2,s_2)\models\mathrm{F}_{\mathrm{CTL}}(\phi,V)$且$(\mathscr{M}_2,s_2)\leftrightarrow_V(\mathscr{M}_1,s_1)$

$\Rightarrow$ 由$(\mathscr{M}_2,s_2)$和$(\mathscr{M},s)$的其他子结构容易构$(\mathscr{M},s_x)$（$(s,s_x)\in S$）造出一个初始K-结构$(\mathscr{M}_3,s_3)$，使得s_2是s_3的直接后继状态，$(\mathscr{M}_2,s_2)$是$(\mathscr{M}_3,s_3)$的子结构且$(\mathscr{M}_3,s_3)\leftrightarrow_V(\mathscr{M},s)$

$\Rightarrow(\mathscr{M}_3,s_3)\models\mathrm{EX}(\mathrm{F}_{\mathrm{CTL}}(\phi,V))$且$(\mathscr{M}_3,s_3)\leftrightarrow_V(\mathscr{M}',s')$

$\Rightarrow(\mathscr{M}',s')\models\mathrm{EX}(\mathrm{F}_{\mathrm{CTL}}(\phi,V))$（由定理 3.1）

($\Leftarrow$) 令$(\mathscr{M}_3,s_3)$是$\mathrm{EX}(\mathrm{F}_{\mathrm{CTL}}(\phi,V))$的模型，则存在一个$(\mathscr{M}_3,s_3)$的子结构$(\mathscr{M}_2,s_2)$（其中$(s_3,s_2)\in R_3$）有$(\mathscr{M}_2,s_2)\models\mathrm{F}_{\mathrm{CTL}}(\phi,V)$

$\Rightarrow$ 对任意上述的$(\mathscr{M}_2,s_2)$，存在一个初始K-结构$(\mathscr{M}_1,s_1)$使得$(\mathscr{M}_1,s_1)\models\phi$且$(\mathscr{M}_1,s_1)\leftrightarrow_V(\mathscr{M}_2,s_2)$

$\Rightarrow$ 由$(\mathscr{M}_1,s_1)$和$(\mathscr{M}_3,s_3)$的其他子结构$(\mathscr{M}_3,s_x)$（$(s_3,s_x)\in S$）容易构造出一个初始K-结构$(\mathscr{M},s)$，使得s_1是s的直接后继状态，$(\mathscr{M}_1,s_1)$是$(\mathscr{M},s)$的子结构且$(\mathscr{M}_3,s_3)\leftrightarrow_V(\mathscr{M},s)$

$\Rightarrow(\mathscr{M},s)\models\mathrm{EX}\phi$且$(\mathscr{M}_3,s_3)\models\mathrm{F}_{\mathrm{CTL}}(\mathrm{EX}\phi,V)$。

同理可证(iii)和(iv)成立。 □

3.4 本章小结

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

章还说明了本文所给出的遗忘理论的定义是命题逻辑下遗忘理论定义的扩展。这些都为后文探索如何使用遗忘理论计算最强必要条件和最弱充分条件奠定了坚实的基础。

第四章　计算CTL下的遗忘：基于归结的方法

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

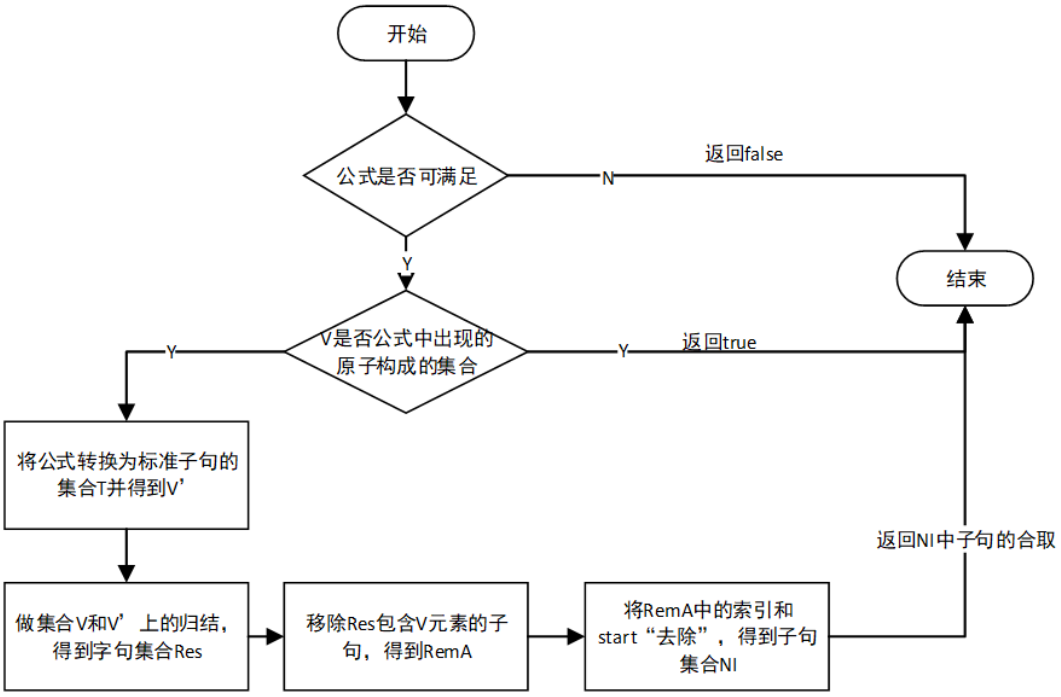


图 4.1: 基于归结的遗忘的主要流程图

4.1　引言

虽然在第二章详细介绍了命题逻辑和模态逻辑S5下的基于归结的计算遗忘的方法，但是值得注意的是CTL公式没有像这两种逻辑一样有标准的自身语言子句形式，而CTL的子句是有索引的。这就注定了CTL下的基于归结的遗忘与上述两者的不同，而且也应该要复杂一些（虽有不同，但也可以借鉴）。本章展示如何使用第 2.3节中表 2.3中的归结规则来计算CTL下的遗忘理论。在本章给定如下约定的记号。令$V \subseteq \mathscr{A}$是

要遗忘的原子命题的集合，$I \subseteq \text{Ind}$是索引的集合，V'表示计算过程所引入的原子命题的集合且满足$V' \cap V = \emptyset$。此外，φ是CTL公式，且T_φ是在φ上使用表 2.1中的转换规则得到的$\text{SNF}^g_{\text{CTL}}$子句的集合。显然可以知道，$V' = Var(T_\varphi) - Var(\varphi)$。在不另加说明的情况下$\mathscr{M}$表示五元组$(S,R,L,[_],s_0)$。此时，本章所设计的算法的伪代码如算法 4.1所示。

算法 4.1 $ERes(\varphi,V)$

输入:

φ　　:CTL公式

V　　:需要遗忘的原子命题的集合

输出:

$ERes(\varphi,V)$　:公式的合取

1: **if** φ 是不可满足的 **then**
2:　**return** $\bot$;
3: **end if**
4: **if** $V = Var(\varphi)$ **then**
5:　**return** $\bot$;
6: **end if**
7: $T_\varphi, V', I \leftarrow Transform(\varphi)$;
8: $Res \leftarrow Resolution(T_\varphi, V \cup V')$;
9: $RemA \leftarrow Removing\_atoms(Res, V)$;
10: $\text{NI} \leftarrow Removing\_index(RemA)$;
11: **return** $\bigwedge_{\psi \in \text{NI}_{\text{CTL}}} \psi$;

算法 4.1对于输入φ和V，输出结果记为$ERes(\varphi,V)$。为了实现这一目标，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？如在第三章中所展示的那样，将一个CTL公式转换为$\text{SNF}^g_{\text{CTL}}$子句的集合会引入新的原子命题和索引。虽然已有的研究说明了CTL公式可以转换为带索引的公式的集合并保证其可满足性，然而并没有表明这两种形式的公式之间的模型具有怎样的联系。本章给出一种扩展的互模拟定义，以描述两种公式的模型之间的关系。

(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，对应算法 4.1中的$Removing_atoms(Res,V)$过程，并提出一种一般化的Ackermann引理。为了“消除”索引，探索几个几个逻辑等价关系，对应算法 4.1中的$Removing_index(RemA)$过程。

本章其余部分组织如下：首先，第4.2节给出二元互模拟的定义及其相关性质。其次，第节从分节详细地介绍算法 4.1如何使用基于归结的方法计算遗忘。第三，第4.4节分析算法 4.1的可终止性及其时间和空间复杂性。最后总结本章的主要工作。

4.2 二元互模拟

对于给定的初始Ind-结构，这里定义一种$\langle V,I\rangle$-互模拟关系。为了与一元的（只考虑原子命题的集合）V-互模拟对应，称$\langle V,I\rangle$-互模拟为二元互模拟。其在V-互模拟的基础上又考虑了索引的集合在结构间关系。

定义 4.1 (二元互模拟). 令$V \subseteq \mathscr{A}$、$I \subseteq Ind$分别是原子命题和索引的集合，$\mathscr{K}_i = (\mathscr{M}_i, s^i)$是初始*Ind*-结构，其中$\mathscr{M}_i = (S_i, R_i, L_i, [_], s_0^i)$ $(i = 1,2)$。称$\mathscr{K}_1$和$\mathscr{K}_2$是$\langle V,I\rangle$-互模拟的（记为$\mathscr{K}_1 \leftrightarrow_{\langle V,I\rangle} \mathscr{K}_2$），当且仅当$\mathscr{K}_1 \leftrightarrow_V \mathscr{K}_2$且$\forall j \in (Ind - I)$有：

- 对任意的$(s,s_1) \in [j]_1$，存在$(s',s_1') \in [j]_2$使得$s \leftrightarrow_V s'$且$s_1 \leftrightarrow_V s_1'$；
- 对任意的$(s',s_1') \in [j]_2$，存在$(s,s_1) \in [j]_1$使得$s \leftrightarrow_V s'$且$s_1 \leftrightarrow_V s_1'$。

由定义 4.1可知，当探讨的公式为CTL公式时，因为不用考虑索引，$\leftrightarrow_{\langle V,I\rangle}$“降维”为$\leftrightarrow_V$。与$\leftrightarrow_V$类似，$\leftrightarrow_{\langle V,I\rangle}$在本文中至关重要的两个性质如下。

命题 4.1. 令$V_1, V_2 \subseteq \mathscr{A}$为原子命题的集合，$I_1, I_2 \subseteq Ind$为索引的集合，$\mathscr{K}_i = (\mathscr{M}_i, s_0^i)$ $(i = 1,2,3)$为初始*Ind*-结构。若$\mathscr{K}_1 \leftrightarrow_{\langle V_1,I_1\rangle} \mathscr{K}_2$、$\mathscr{K}_2 \leftrightarrow_{\langle V_2,I_2\rangle} \mathscr{K}_3$，则：

(i) $\mathscr{K}_1 \leftrightarrow_{\langle V_1 \cup V_2, I_1 \cup I_2\rangle} \mathscr{K}_3$；

(ii) 如果$V_1 \subseteq V_2$且$I_1 \subseteq I_2$，则$\mathscr{K}_1 \leftrightarrow_{\langle V_2,I_2\rangle} \mathscr{K}_2$。

证明. (i) 由定义 4.1可知$\mathscr{K}_1 \leftrightarrow_{V_1} \mathscr{K}_2$、$\mathscr{K}_2 \leftrightarrow_{V_2} \mathscr{K}_3$，因此由命题 3.1可知$\mathscr{K}_1 \leftrightarrow_{V_1 \cup V_2} \mathscr{K}_3$。

$\mathscr{K}_1 \leftrightarrow_{\langle V_1,I_1\rangle} \mathscr{K}_2$

$\Rightarrow \forall j \in (\mathrm{Ind} - (I_1 \cup I_2))$有：$\forall (s,s_1) \in [j]_1$，$\exists (s',s_1') \in [j]_2$使得$s \leftrightarrow_{V_1} s'$，$s_1 \leftrightarrow_{V_1} s_1'$

$\Rightarrow$ 又因为$\mathscr{K}_2 \leftrightarrow_{\langle V_2,I_2\rangle} \mathscr{K}_3$，所以$\exists (s'',s_1'') \in [j]_3$使得$s' \leftrightarrow_{V_2} s''$，$s_1' \leftrightarrow_{V_2} s_1''$

$\Rightarrow s \leftrightarrow_{V_1 \cup V_2} s''$且$s_1 \leftrightarrow_{V_1 \cup V_2} s_1''$

同理可证，$\forall (s'',s_1'') \in [j]_3$，$\exists (s,s_1) \in [j]_1$使得$s \leftrightarrow_{V_1 \cup V_2} s''$且$s_1 \leftrightarrow_{V_1 \cup V_2} s_1''$。因此，由定义 4.1可知$\mathscr{K}_1 \leftrightarrow_{\langle V_1 \cup V_2, I_1 \cup I_2\rangle} \mathscr{K}_3$。

(ii)可以由命题 3.1中的(ii)可得。 □

对于给定的Ind-Kripke结构$\mathscr{M} = (S,R,L,[_],s_0)$和$\mathscr{M}' = (S',R',L',[_],s_0')$，$\mathscr{M}$和$\mathscr{M}'$之间的二元互模拟关系$\leftrightarrow_{\langle V,I\rangle}$给描述公式间在二元组$\langle V,I\rangle$上的等价关系提供了前提条件。这同时也为解决本章引言部分提出的问题(1)奠定了基础。

定义 4.2. 给定两个公式（或公式的集合）T_1和T_2，$I \subseteq Ind$是索引的集合，$V'' \subseteq \mathscr{A}$是原子命题的集合。如果下面条件被满足，则称$T_1$和$T_2$在二元组$\langle V,I\rangle$上逻辑等价（记为$T_1 \equiv_{\langle V,I\rangle} T_2$）：

- $\forall (\mathscr{M},s_0) \in Mod(T_1)$，$\exists (\mathscr{M}',s_0') \in Mod(T_2)$使得$(\mathscr{M},s_0) \leftrightarrow_{\langle V,I\rangle} (\mathscr{M}',s_0')$，且
- $\forall (\mathscr{M}',s_0') \in Mod(T_2)$，$\exists (\mathscr{M},s_0) \in Mod(T_1)$使得$(\mathscr{M},s_0) \leftrightarrow_{\langle V,I\rangle} (\mathscr{M}',s_0')$。

4.3　基于归结的方法计算遗忘

在本节中，将围绕之前提出的两个问题和算法 4.1中的几个关键步骤（第7到第11行）来证明$ERes(\varphi,V) \equiv_{\langle V',\emptyset\rangle} \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。在这个等价关系中$\varphi$为CTL公式，$V$为需要遗忘的原子命题的集合，$V'$是将$\varphi$转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句集合过程中引入的新的原子命题的集合。这个结论表明，如果$ERes(\varphi,V)$公式中不包含V'中的元素，或者$\mathrm{IR}(ERes(\varphi,V),V')$，则$ERes(\varphi,V)$就是从$\varphi$中遗忘掉$V$中的原子命题之后得到的结果。

4.3.1　将CTL公式转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合

将CTL公式φ转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合这一过程（记为$Trangsform(\varphi)$）对应算法 4.1中的第7行所代表的过程。对于输入φ，该过程事先将φ转换为其否定范式（记为$\mathbf{nnf}(\varphi)$），然后通过下面的等价关系[61,82]将出现在公式中的$\top$和$\bot$“去掉”（记为$\mathbf{simp}(\mathbf{nnf}(\varphi))$）。

$$
\begin{array}{llll}
(\varphi \wedge \top) \equiv \varphi & (\varphi \wedge \bot) \equiv \bot & (\varphi \vee \top) \equiv \top & (\varphi \vee \bot) \equiv \varphi \\
\neg\top \equiv \bot & \neg\bot \equiv \top & QT\bot \equiv \bot & QT\top \equiv \top \\
Q(\varphi \mathrm{U} \bot) \equiv \bot & Q(\varphi \mathrm{U} \top) \equiv \top & Q(\bot \mathrm{U} \varphi) \equiv \varphi & Q(\top \mathrm{U} \varphi) \equiv Q\mathrm{F}\varphi \\
Q(\varphi \mathrm{W} \bot) \equiv Q\mathrm{G}\varphi & Q(\varphi \mathrm{W} \top) \equiv \top & Q(\bot \mathrm{W} \varphi) \equiv \varphi & Q(\top \mathrm{W} \varphi) \equiv \top
\end{array}
$$

在上述等价关系中，$Q \in \{\mathrm{A},\mathrm{E}\}$是路径量词，$T \in \{\mathrm{F},\mathrm{G},\mathrm{X}\}$是时序操作符。

在得到$\mathbf{simp}(\mathbf{nnf}(\varphi))$后，将$T_\varphi$初始化为$\{\mathrm{AG}(\mathbf{start} \rightarrow p), \mathrm{AG}(p \rightarrow \mathbf{simp}(\mathbf{nnf}(\varphi)))\}$，然后转换过程从表 2.1中找到匹配的规则来将$T_\varphi$中的非$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句形式的公式转换为$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句，并将得到的结果更新$T_\varphi$直到$T_\varphi$中不存在非$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句形式的公式。这个转换过程被写为算法 4.2。

在算法 4.2中，$Trans(\psi)$表示使用表 2.1中的某一条规则来转换ψ（规则中长横线上面的部分），并返回规则的结果（规则中长横线下面的部分）、引入的新原子命题和索引。这里使用规则**Trans(6)**为例来描述这一过程。令$\psi = q \rightarrow \mathrm{AX}\phi$，可以看出当$\phi$不是经典子句的时候$\psi$不是$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句，且规则**Trans(6)**是能使用的（匹配的）规则，则$Trans(\psi)$对ψ使用规则**Trans(6)**并返回结果$\{q \rightarrow \mathrm{AX}q', q' \rightarrow \phi\}$（在这里，引入了新的原子命题$q'$，但是没有引入新的索引）。

命题 4.2. *令φ是一个CTL公式，$(T_\varphi, V', I) = Transform(\varphi)$，则$\varphi \equiv_{\langle V',I\rangle} T_\varphi$。*

证明. 为了讨论方便，令$Transform(\varphi)$过程产生了一个公式集合的序列，$T_0, T_1, \ldots, T_n = T_\varphi$，其中$p$是不出现在$\varphi$中的原子命题，$T_0 = \{\mathrm{AG}(\mathbf{start} \rightarrow p), \mathrm{AG}(p \rightarrow \mathbf{simp}(\mathbf{nnf}(\varphi)))\}$且

算法 4.2 *Transform*(φ)

输入:

φ :CTL公式

输出:

T_φ : $\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合

V' : 新引入的原子命题的集合

I : 引入的索引的集合

1: $T_\varphi \leftarrow \{\mathrm{AG}(\mathbf{start} \rightarrow p), \mathrm{AG}(p \rightarrow \mathbf{simp}(\mathbf{nnf}(\varphi)))\}$;（其中$p \in \mathscr{A} - var(\varphi)$）
2: $V' \leftarrow \{p\}$;
3: $I \leftarrow \emptyset$;
4: **while** $\exists \psi \in T_\varphi$使得$\psi$不是$\mathrm{SNF}^g_{\mathrm{CTL}}$子句 **do**
5: $T_\varphi \leftarrow T_\varphi - \{\psi\}$;
6: $T_\varphi \leftarrow Trans(\psi) \cup T_\varphi$;
7: **if** $Trans(\psi)$引入了一个新原子命题q **then**
8: $V' \leftarrow V' \cup \{q\}$;
9: **end if**
10: **if** $Trans(\psi)$引入了一个新的索引ind **then**
11: $I \leftarrow I \cup \{ind\}$;
12: **end if**
13: **end while**
14: **return** T_φ, V', I;

对任意的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，所有的公式都是其否定范式的形式。

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

（1）这里将证明$\varphi \equiv_{\langle \{p\}, \emptyset\rangle} T_0$。

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

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

（2）这里将证明对任意的i（$0 \leq i < n$）有$T_i \equiv_{\langle V', I\rangle} T_{i+1}$成立，其中$T_{i+1} = (T_i - \{\psi\}) \cup R_i$。为了方便，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，$T_{i+1} = X \cup R_i$）。下面证明规则$t \in \{$**Trans(1), Trans(4), Trans(6)**$\}$的情形，其他情形可以类似地证明。

（a）$t =$ **Trans(1)**。

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

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

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

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

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

（b）$t=$**Trans(4)**。

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

可以如下构造Ind-Kripke结构$\mathscr{M}_2$：

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

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

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

（c）t=**Trans(6)**。

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

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

可以如下构造Ind-Kripke结构$\mathscr{M}_2$：

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

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

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

命题 4.2表示，对于给定的CTL公式φ，通过上述的转换过程得到的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合$T_\varphi$与$\varphi$在二元组$\langle V',I\rangle$上逻辑等价。下面给出本章的运行示例来展示每一个过程。

例 4.1 (运行示例)**.** 给定公式$\varphi=\mathrm{A}((p\wedge q)\mathrm{U}(f\vee m))\wedge r$和原子命题的集合$V=\{p,r\}$。$(T_\varphi,V',I)=Transform(\varphi)$，其中$V'=\{x,y,z,w\}$（$w$是与子句$z\rightarrow\mathrm{AF}x$对应的新引入的原子命题[1]），$I=\emptyset$且$T_\varphi$中的元素如下：

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

4.3.2 归结过程

本节的归结过程在转换过程之后执行。给定的公式φ和原子命题的集合V，令$(T_\varphi,V',I)=Transform(\varphi)$。在$T_\varphi$和$V\cup V'$上的归结过程（记为$Resolution(T_\varphi,V\cup V')$）产生一个子句集合的序列$T_0=T_\varphi,T_1,T_2,\ldots,T_n=Res$并返回$Res$。在这个序列中，对所有的$0\le i\le n$都有$T_{i+1}=T_i\cup R_i$（$R_i$是由表 2.3中的某条归结规则作用到$T_i$中的某些子句上得到的结果），且在$Res$中不能再由任何的归结规则产生新的子句（或者产生了矛盾，即子句$\mathbf{start}\rightarrow\bot$和子句$\top\rightarrow\bot$，因为在此情况下可以得出$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)\equiv\bot$）。值得注意的是，在这一过程中产生的$T_i$（$0\le i\le n$）是$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合。

下面的命题表示T_φ与归结过程得到的结果在二元组$\langle V\cup V',\emptyset\rangle$上逻辑等价。

[1]注意：本文中对每个Q-某时子句（$Q\in\{\mathrm{E},\mathrm{A}\}$）都产生一个新的原子变量$w$与之对应。

命题 4.3. 给定公式φ和原子命题的集合V。若$(T_\varphi, V', I) = Transform(\varphi)$，则$T_\varphi \equiv_{\langle V\cup V',\emptyset\rangle} Resolution(T_\varphi, V\cup V')$。

证明. 这一结论可以通过证明$T_i \equiv_{\langle V\cup V',\emptyset\rangle} T_{i+1}$（$0 \le i < n$），其中$T_{i+1} = T_i \cup R_i$。记$\Pi \to_r R_i$为通过对$\Pi \subseteq T_i$和原子命题$p \in (V \cap Var(\Pi))$使用表 2.3中的归结规则$r$得到$R_i$。

（1）这里证明若$r \in \{$**(SRES1)**, …, **(SRES8)**, (**RW1**), (**RW2**)$\}$，则$T_i \equiv_{\langle\{p\},\emptyset\rangle} T_{i+1}$。

一方面可以证明$\Pi \models R_i$，因而$T_{i+1} \models T_i$。另一方面，由于$T_i \subseteq T_{i+1}$，所以$T_{i+1} \models T_i$。显然$T_i \equiv T_{i+1}$，又$\emptyset \subseteq (V\cup V')$且$\emptyset \subseteq \emptyset$，由命题 4.1可知$T_i \equiv_{\langle V\cup V',\emptyset\rangle} T_{i+1}$。

（2）这里证明若$r =$**(ERES1)**，则$T_i \equiv_{\langle\{l, w^{\mathrm{A}}_{\neg l}\},\emptyset\rangle} T_{i+1}$，其中$w^{\mathrm{A}}_{\neg l} \in V'$是与子句$Q \to \mathrm{AF}\neg l$相关的新的原子命题，$l$是文字（即：$p$或者$\neg p$）。

在文章[83]中已经证明$\Pi \models R_i$，因此有$T_{i+1} = T_i \cup \Lambda^{\mathrm{A}}_{\neg l}$，其中$\Lambda^{\mathrm{A}}_{\neg l}$是通过使用表 2.1中的转换规则作用到$R_i$上得到的$\mathrm{SNF}^g_{\mathrm{CTL}}$子句的集合（请查看文章[84]获取更加详细的描述）。显然，对所有的$(\mathscr{M}_1, s_1) \in Mod(T_i = X \cup \Pi)$都存在一个$(\mathscr{M}_2, s_2) \in Mod(T_{i+1} = T_i \cup \Lambda^{\mathrm{A}}_{\neg l})$使得$(\mathscr{M}_1, s_1) \leftrightarrow_{\langle\{p, w^{\mathrm{A}}_{\neg l}\},\emptyset\rangle} (\mathscr{M}_2, s_2)$，且对任意的$(\mathscr{M}_2, s_2) \in Mod(T_{i+1} = T_i \cup \Lambda^{\mathrm{A}}_{\neg l})$也存在一个$(\mathscr{M}_1, s_1) \in Mod(T_i = X \cup \Pi)$使得$(\mathscr{M}_1, s_1) \leftrightarrow_{\langle\{p, w^{\mathrm{A}}_{\neg l}\},\emptyset\rangle} (\mathscr{M}_2, s_2)$。又$\{p, w^{\mathrm{A}}_{\neg l}\} \subseteq (V \cup V')$且$\emptyset \subseteq \emptyset$，由命题 4.1可知$T_i \equiv_{\langle V\cup V',\emptyset\rangle} T_{i+1}$。

当规则为**(ERES2)**时可以类似地证明。 □

命题4.3和命题4.2表明$\varphi \equiv_{\langle V'',\emptyset\rangle} Resolution(T_\varphi, V'')$，即对任意的公式$\psi$，$\mathrm{IR}(\psi, V'')$蕴涵“$\varphi \models \psi$当且仅当$Resolution(T_\varphi, V'') \models \psi$”，其中$V'' = V \cup V'$。

例 4.2 (例 4.1的延续)**.** 对例 4.1中的T_φ和$V \cup V'$使用上述归结过程，除了例 4.1中的子句，还得到如下子句:

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

4.3.3 “移除”过程

本节意在“移除”如何将归结过程得到的结果中含有V中原子命题的子句，而保证其在二元组$\langle V\cup V,I\rangle$上的逻辑等价关系，这一过程对应算法 4.1中的第9行。给定$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句$C$和原子命题的集合$V$：

$$Removing\_atoms(C,V)\equiv\begin{cases}\top, & Var(C)\cap V\neq\emptyset;\\ C, & \text{otherwise.}\end{cases}$$

此外，对于$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合$\Pi$，

$$Removing\_atoms(\Pi,V)=\{Removing\_atoms(r,V)\mid r\in\Pi\}.$$

可以看出，通过这一过程后得到的子句集合里的子句不再包含V中的原子命题的集合且与原集合满足下面关系。

命题 4.4. *对于给定的公式φ和原子命题的集合，有：*

$$Resolution(T_\varphi,V\cup V')\equiv_{\langle V\cup V',\emptyset\rangle} Removing\_atoms(Resolution(T_\varphi,V\cup V'),V).$$

证明. 因为要考虑的索引集合为空集，所以只需证明：

$$Resolution(T_\varphi,V\cup V')\equiv_{V\cup V'} Removing\_atoms(Resolution(T_\varphi,V\cup V'),V)$$

其中$\equiv_V$与$\equiv_{\langle V,I\rangle}$的定义类似，只是不考虑索引部分。为了证明这一过程，下面给出两个事实：

- **(GNA)** 对任意的$p\in Var(\varphi)$，p都不出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的左边；
- **(PI)** 对任意的$p\in V'$，如果p出现在$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的左边，那么$p$为负出现，即该子句关于$p$是负的。

不失一般性地，假设$V=\{p\}$，且规定$Res=Resolution(T_\varphi,V\cup V')$、$V''=V\cup V'$、$C_i$是经典子句、$l$是$p$或者$\neg p$。显然，$Res\models Removing\_atoms(Res,V)$，这里将证明对任意的$\mathscr{K}=(\mathscr{M},s)$（$\mathscr{K}\in Mod(Removing_atoms(Res,V))$且$\mathscr{M}=(S,R,L,s)$），存在一个初始结构$\mathscr{K}'=(\mathscr{M}',s')$使得$\mathscr{K}\leftrightarrow_{V''}\mathscr{K}'$且$\mathscr{K}'\models Res$。接下来从两个大点证明这一结论。

（1）假设全局子句$C=\top\rightarrow C_1\vee l\in Res$，其中$Var(l)=\{p\}$。

(a) 如果不存在子句$C'\in Res$使得C和C'在p上是可归结的[2]，这意味着在Res中不存

[2]如果能在表 2.3中找到一条规则使得子句C和C'为该规则的横线上面部分，且相应的文字是p，则称C和C'在p上是可归结。

在其他非Pt-某时子句包含有文字$\neg l$，其中$Pt \in \{\mathrm{A}, \mathrm{E}\}$。则有两种情况需要讨论：

(i) $p \notin Var(C')$。$\forall \mathscr{K} = (\mathscr{M}, s) \in Mod(Removing_atoms(Res, V))$可以如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S, R, L', s)$（即$s' = s$），其中$L'$与$L$一样，除了对于$s_1 \in S$，如果$(\mathscr{M}, s_1) \not\models C_1 \vee l$则“若$l = p$，则令$L'(s_1) = L(s_1) \cup \{p\}$，否则令$L'(s_1) = L(s_1) - \{p\}$”。

(ii) 如果$C' = Q \rightarrow Pt\mathrm{F}\neg l$。不失一般性地，假设$l = p$且$Q$为文字（$Q$为文字的合取的情形可以类似证明）。$\forall \mathscr{K} = (\mathscr{M}, s) \in Mod(Removing\_atoms(Res, V))$，如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S', R', L', s')$，其中$S' = S$，$R' = R$，$s' = s$，且$L' = L$，除了$\forall s \in S'$，“如果$Q$为正文字，则令$L'(s) = L(s) - \{Q\}$，且若$(\mathscr{M}, s) \not\models C_1$，则$L'(s) = L(s) \cup \{p\}$”，否则令$L'(s) = L(s)$。

因此，有$\mathscr{K} \leftrightarrow_{V''} \mathscr{K}'$且$\mathscr{K}' \models Res$。

(b) 如果存在子句$C' \in Res$使得C和C'在p上是可归结的。

(i) 若$C' = Q \rightarrow Pt\mathrm{X}(C_2 \vee \neg l)$（这里令$Pt = \mathrm{G}$，当$Pt = \mathrm{E}$时可以类似地证明），因此有$Q \rightarrow \mathrm{GX}(C_1 \vee C_2) \in Res$。$\forall \mathscr{K} = (\mathscr{M}, s) \in Mod(Removing\_atoms(Res, V))$，如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S, R, L', s)$（即：$s' = s$），其中$L'$与$L$类似，除了$\forall s_1 \in S$，若$(\mathscr{M}, s_1) \not\models Q$，则“$\forall (s_1, s_2) \in R$，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$L'(s_2) = L(s_2) - \{p\}$””，否则，若$(\mathscr{M}, s_2) \models C_1 \wedge \neg C_2$，则“若$l = p$，则$L'(s_2) = L(s_2) - \{p\}$，否则$L'(s_2) = L(s_2) \cup \{p\}$””；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，$\mathscr{K} \leftrightarrow_{V''} \mathscr{K}'$且$\mathscr{K}' \models C' \wedge C$。

(ii) 若$C' = Q \rightarrow Pt\mathrm{F}\neg l$。不失一般性地，假设$l = p$。由于$C$和$C'$在$p$上可归结，则存在$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集合$\Pi = \{P^1_1 \rightarrow *C^1_1, \ldots, P^1_{m_1} \rightarrow *C^1_{m_1}, P^n_1 \rightarrow *C^n_1, \ldots, P^1_{m_n} \rightarrow *C^1_{m_n}\}$使得$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} P^i_j \rightarrow \mathrm{EXEG}l$，其中$*$要么为空字符要么为集合$\{\mathrm{GX}, \mathrm{E}\langle ind \rangle \mathrm{X}\}$中的某个元素。此外，$\neg C_1 \rightarrow l \in \Pi$。因此，通过规则**ERES1**可以得到子句$C'' = \top \rightarrow \neg Q \vee \neg p \vee C_1$（对规则**ERES2**类似）。因此，在子句$C$和$C''$上使用规则**SRES8**可得$\top \rightarrow \neg Q \vee C_1$。$\forall \mathscr{K} = (\mathscr{M}, s) \in Mod(Removing\_atoms(Res, V))$，可以如下构造$(\mathscr{M}', s')$：令$\mathscr{M}' = (S, R, L', s)$（即：$s' = s$），其中$L'$与$L$类似，除了$\forall s_1 \in S$，“若$(\mathscr{M}, s_1) \models Q$，则令$L'(s_1) = L(s_1) - \{p\}$，否则令$L'(s_1) = L(s_1) \cup \{p\}$”。显然，$\mathscr{K} \leftrightarrow_{V''} \mathscr{K}'$且$\mathscr{K}' \models C' \wedge C$。

（2）这里考虑Pt-步子句，令$C \in Res$为子句$Q \rightarrow \mathrm{AX}(C_1 \vee \neg l)$。不失一般性地，假设存在子句$C' \in Res$使得$C$和$C'$在$p$上可归结，且$l = p$.

若$C' = Q_1 \to Pt\mathrm{X}(C_2 \vee l)$（$Pt = \mathrm{E}_{ind}$，$Pt = \mathrm{A}$可以类似地证明），则有$Q \wedge Q_1 \to \mathrm{E}_{ind}\mathrm{X}(C_1 \vee C_2) \in Res$。因此，$\forall \mathscr{K} = (\mathscr{M}, s) \in Mod(Removing\_atoms(Res, V))$，构造$(\mathscr{M}', s')$：$\mathscr{M}' = (S, R, L', s)$（令$s' = s$），其中$L'$与$L$类似，除了$\forall s_1 \in S$：

(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（对于$(s_1, s_2') \in \pi_s^{\langle ind \rangle}$，若$(\mathscr{M}, s_2') \not\models C_2$，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$\forall (s_1, s_2) \in R$（若$(\mathscr{M}, s_2) \not\models C_1$，则令$L'(s_2) = L(s_2) \cup \{p\}$，否则令$L'(s_2') = L(s_2')$），否则令$L'(s_2') = L(s_2')$”。

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

显然，$\mathscr{K} \leftrightarrow_{V''} \mathscr{K}'$且$\mathscr{K}' \models C' \wedge C$，其中$\mathscr{K}' = (\mathscr{M}', s')$。

其他情形的组合都可以从上述描述中找到或者类似，因此不在这里赘述。 □

例 4.3 (例 4.2的延续). 在移除掉$Resolution(T_\varphi, V'')$中包含$V$中元素的子句后，得到如下子句的集合：

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

4.3.4 索引和“start”的“消除”

算法 4.1中的一个关键的步骤就是将转换过程中引入的索引和**start**“消除”。通过表 2.1里的规则可知，没有两个E-某时子句有相同的索引，且在归结过程中没有E-某时子句产生。因而可以通过等价关系“$\mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi \equiv \varphi \vee \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{F}\varphi$”事先将“$\mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi$”转换为“$\varphi \vee \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{F}\varphi$”。

引理 4.1. 给定$\mathrm{SNF}^g_{\mathrm{CTL}}$公式$\mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi$，下面等价关系成立：

$$\mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi \equiv \varphi \vee \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{F}\varphi.$$

证明. ($\Rightarrow$) 令$(\mathscr{M}, s_0) \in Mod(\mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi)$，存在一条路径$\pi_{s_0}^{\langle ind \rangle} = (s_0, s_1, \ldots)$使得对于某些$s_j \in \pi_s^{\langle ind \rangle}$（$j \geq 0$）有$(\mathscr{M}, s_j) \models \varphi$。因此，$j = 0$或者$j > 0$，所以有$(\mathscr{M}, s_0) \models \varphi \vee \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{F}\varphi$。

($\Leftarrow$) 令$(\mathscr{M}, s_0) \in Mod(\varphi \vee \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{F}\varphi)$，因此有$(\mathscr{M}, s_0) \models \varphi$或者存在一条索引号为$ind$的路径$\pi_{s_0}^{\langle ind \rangle} = (s_0, s_1, \ldots)$使得$(\mathscr{M}, s_1) \models \mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi$。因此，有$(\mathscr{M}, s_0) \models \mathrm{E}_{\langle ind \rangle}\mathrm{F}\varphi$。 □

此外，要消除索引，还需要引入以下等价关系。

引理 4.2. 令P，P_i和φ_i为*CTL*公式，则：

(i) $\bigwedge_{i=1}^{n}(P \to \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_i) \equiv_{\langle\emptyset,\{ind\}\rangle} P \to \mathrm{EX}\bigwedge_{i=1}^{n}\varphi_i$；

(ii) $\bigwedge_{i=1}^{n}(P_i \to \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_i) \equiv_{\langle\emptyset,\{ind\}\rangle} \bigwedge_{e\in 2^{\{1,\ldots,n\}}\setminus\{\emptyset\}}(\bigwedge_{i\in e}P_i \to \mathrm{EX}(\bigwedge_{i\in e}\varphi_i))$；

(iii) $\bigwedge_{i=1}^{n}(P \to \mathrm{E}_{\langle ind\rangle}\mathrm{F}\varphi_i) \equiv_{\langle\emptyset,\{ind\}\rangle} P \to \bigvee \mathrm{EF}(\varphi_{j_1}\wedge\mathrm{EF}(\varphi_{j_2}\wedge\mathrm{EF}(\cdots\wedge\mathrm{EF}\varphi_{j_n})))$，其中$(j_1,\ldots,j_n)$为集合$\{1,\ldots,n\}$中的所有元素构成的序列；

(iv) $(P \to (C\vee \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_1))\wedge(P\to \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_2) \equiv_{\langle\emptyset,\{ind\}\rangle} P \to ((C\wedge\mathrm{EX}\varphi_2)\vee\mathrm{EX}(\varphi_1\wedge\varphi_2))$；

(v) $(P_1 \to \varphi\vee \mathrm{E}_{\langle ind\rangle}\mathrm{XE}_{\langle ind\rangle}\mathrm{F}\varphi_1)\wedge(P_2\to \mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_2) \equiv_{\langle\emptyset,\{ind\}\rangle} (P_1 \to \varphi\vee\mathrm{EXEF}\varphi_1)\wedge(P_2\to\mathrm{EX}\varphi_2)$ $\wedge(P_1\wedge P_2 \to ((\varphi\wedge\mathrm{EX}\varphi_2)\vee\mathrm{EX}(\varphi_2\wedge\mathrm{EF}\varphi_1)))$。

证明. (i) $\forall(\mathscr{M},s_0)\in Mod(\bigwedge_{i=1}^{n}(P\to\mathrm{E}_{\langle ind\rangle}\mathrm{X}\varphi_i))$，若$(\mathscr{M},s_0)\models P$，则存在$(s_0,s_1)\in[ind]$使得$(\mathscr{M},s_1)\models\varphi_1,\ldots,(\mathscr{M},s_1)\models\varphi_n$。因此存在$(s_0,s_1)\in R$使得$(\mathscr{M},s_1)\models\bigwedge_{i=1}^{n}\varphi_i$，即$(\mathscr{M},s_0)\models P\to\mathrm{EX}\bigwedge_{i=1}^{n}\varphi_i$。

$\forall(\mathscr{M},s_0)\in Mod(P\to\mathrm{EX}\bigwedge_{i=1}^{n}\varphi_i)$，假定存在$(s_0,s_1)\in R$使得$(\mathscr{M},s_1)\models\bigwedge_{i=1}^{n}\varphi_i$。容易构造一个初始Ind-结构$(\mathscr{M}',s_0)$使得$(\mathscr{M}',s_0)$与$(\mathscr{M},s_0)$相同，除了$(s_0,s_1)\in[ind]$，即：$(\mathscr{M},s_0)\leftrightarrow_{\langle\emptyset,\{ind\}\rangle}(\mathscr{M}',s_0)$。

(ii) ($\Rightarrow$) 对等式左手边公式的任意模型$(\mathscr{M},s_0)$，若$(\mathscr{M},s_0)\models\bigwedge_{i=1}^{m}P_{j_i}$（$j_i\in\{1,\ldots,n\}$且$1\le m\le n$），则存在$s_0$的下一个状态$s_1$（即：$(s_0,s_1)\in[ind]$）使得$(\mathscr{M},s_1)\models\bigwedge_{i=1}^{m}\varphi_{j_i}$。通过$[ind]$的定义，有$(s_0,s_1)\in R$，因此$(\mathscr{M},s_0)\models\bigwedge_{i=1}^{m}P_{j_i}\to\mathrm{EX}(\bigwedge_{i=1}^{m}\varphi_{j_i})$。另一边类似(i)的证明。

(iii) ($\Leftarrow$) 对等式右手边公式的任意模型$(\mathscr{M},s_0)$，如果$(\mathscr{M},s_0)\models P$，则存在一条路径$\pi_{s_0}=(s_0,s_1,\ldots)$使得$(\mathscr{M},s_{j_i})\models\varphi_{j_i}$（$1\le i\le n$）。构造一个初始Ind-结构$(\mathscr{M}',s_0)$使得$(\mathscr{M}',s_0)$与$(\mathscr{M},s_0)$相同，除了对$\pi_{s_0}$上的任意$(s_j,s_{j+1})$，存在$(s_j,s_{j+1})\in[ind]$ $(0\le j)$。显然，$(\mathscr{M}',s_0)\models\bigwedge_{i=1}^{n}(P\to\mathrm{E}_{\langle ind\rangle}\mathrm{F}\varphi_i)$且$(\mathscr{M},s_0)\leftrightarrow_{\langle\emptyset,\{ind\}\rangle}(\mathscr{M}',s_0)$。另一个方向与(ii)中的证明类似。

其他结果显然是成立的，类似上面的证明。 □

给定子句集合Π，*Removing_index*(Π)这一过程使用引理 4.2中的等价关系将出现在Π中的子句中的索引“去掉”，并返回不具有索引的公式的集合。

命题 4.5. 令Π是$\mathrm{SNF}^{g}_{\mathrm{CTL}}$子句的集合，且对于任意的索引$i$，至多包含一个索引为$i$的E-某时子句。因此，

$$\Pi \equiv_{\langle\emptyset,I\rangle} Removing\_index(\Pi)$$

其中I是出现在Π中的索引的集合.

证明. 由于Π没有两个E-某时子句有相同的索引，所以事先使用引理 4.1中的等价关系将显示出$\mathrm{E}_{ind}\mathrm{X}$时序算子，然后使用引理 4.2中的等价关系“消除”掉$\mathrm{E}_{ind}\mathrm{X}$中的索引。□

需要指出的是，例 4.3中没有索引需要消除。

给定公式φ，下面定义从φ中“消除”**start**的操作，结果记为φ_{CTL}：

$$\varphi_{\mathrm{CTL}} \equiv \begin{cases} D, & \text{如果}\varphi\text{是}\mathrm{AG}(\mathbf{start} \rightarrow D)\text{这种形式;} \\ \varphi, & \text{否则。} \end{cases}$$

此外，对于给定的公式的集合Π，$\Pi_{\mathrm{CTL}} = \{\varphi_{CTL} \mid \varphi \in \Pi\}$。因此，由$\varphi \equiv \mathrm{AG}(\mathbf{start} \rightarrow \varphi)$[83]可知$\Pi \equiv \Pi_{\mathrm{CTL}}$。

到此为止，已经介绍完了算法 4.1中的每个步骤。综上所述，通过算法 4.1得到的结果$ERes(\varphi, V)$与原公式φ在二元组$\langle V \cup V', \emptyset \rangle$上逻辑等价。在接下来的小节中将着重探讨本章开头提出的算法的一些相关计算属性，并提出一种一般的Ackermann引理来尽可能地使$ERes(\varphi, V)$逼近从φ中遗忘掉原子命题集合V的结果。

4.3.5 替换V'中的原子

尽管在 4.3.3节中已经定义了“移除”与V中元素相关的子句的操作，但是由转换过程可知新的原子命题（即V'中的原子命题）被引入而还没被处理过。为了尽可能多地替换这些原子命题，这部分扩展Szalas提出的Ackermann引理[85]如下。在此之前，先约定一些记号：对于给定的公式（或公式的集合）Γ，用$\Gamma[x/\varphi]$表示将Γ中的所有x的出现用φ替换得到的结果；在公式$Qt\mathscr{T}C$中，Qt（$\mathscr{T}$）为空字符或者$Qt \in \{\mathrm{A}, \mathrm{E}\}$且$\mathscr{T} \in \{\mathrm{X}, \mathrm{F}\}$。

定理 4.1 (一般化的Ackermann引理)**.** 令$\Delta = \{\top \rightarrow \neg x \vee C_1, \ldots, \top \rightarrow \neg x \vee C_n, x \rightarrow B_1, \ldots, x \rightarrow B_m\}$、$\Gamma'$是算法 *4.1*中*NI*的子集，$\Gamma = \Delta \cup \Gamma'$。其中，$x \in V'$、$C_i$ $(1 \leq i \leq n)$是不包含原子命题x的经典子句、B_j $(1 \leq j \leq m)$是形如$Qt\mathscr{T}C$形式的公式的吸取或者合取（其中C为不包含原子命题x的*CTL*公式）。令$\varphi = \bigwedge_{i=1}^{n} C_i \wedge \bigwedge_{j=1}^{m} B_j$，那么如果$\Gamma'$关于$x$是正的，则$\Gamma'[x/\varphi] \equiv_{\langle \{x\}, \emptyset \rangle} \Gamma$。

证明. ($\Rightarrow$) 对任意Γ的模型$(\mathscr{M}, s_0)$，由于Γ'关于x是正的（即：Γ'关于x是单调的）且$x \rightarrow \varphi$，因而有$(\mathscr{M}, s_0) \models \Gamma'[x/\varphi]$。

($\Leftarrow$) 对任意$\Gamma'[x/\varphi]$的模型$(\mathscr{M}, s_0)$（$\mathscr{M} = (S, R, L, [_], s_0)$），构造一个Ind-初始结构$\mathscr{M}' = (S', R', L', [_]', s_0')$使得$S' = S$、$R' = R$、$s_0' = s_0$、$[\_] = [\_]'$、且$L'$与$L$相同，除了对任意的$s' \in S'$，若$(\mathscr{M}', s') \models \varphi$，则令$L'(s') = L(s) \cup \{x\}$，否则令$L'(s') = L(s) - \{x\}$（即：$x \leftrightarrow \varphi$）。

显然，$(\mathscr{M},s_0)\leftrightarrow_{\langle\{x\},\emptyset\rangle}(\mathscr{M}',s_0')$且$(\mathscr{M}',s_0')\models\Gamma$。 □

这一结论尽量“去掉”了V'中的原子命题。在算法 4.1中并没有把这一过程写入，但是为了使算法的结果更加接近遗忘的结果，这一过程被加到消除**start**之前，消除索引之后。下面的例子展示了这一过程。

例 4.4 (例 4.3的延续)**.** 假定例 4.3中的子句的集合为Γ。显然，$\Delta=\{\top\rightarrow\neg x\vee f\vee m\}$且$\Gamma'=\Gamma-\Delta$。则使用定理 4.1在$\Delta$上（关于$x$），然后再使用定理 4.1在$\{\top\rightarrow\neg z\vee f\vee m\vee y,\top\rightarrow\neg z\vee f\vee m\vee q,\top\rightarrow\neg z\vee \mathrm{AF}(f\vee m)\}$上（关于$z$）得到下面的公式的集合：

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

此外，在上面例子的结果中使用“消除”**start**过程得到下面的公式的集合:

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

从上面的几个结论可以得出，对对给定的CTL公式φ和原子命题集合V，当包含V'（转换构成引入的原子命题的集合）中的原子命题的子句能够从NI中移除，则得到的结果为从φ中遗忘掉V后得到的结果。

定理 4.2. *给定CTL公式φ和原子命题集合V。则$\mathrm{ERes}(\varphi,V)\equiv_{\langle V',\emptyset\rangle}\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$，其中$V'$是由$Transform$过程引入并存留在$\mathrm{ERes}(\varphi,V)$的原子命题的集合。*

证明. 由命题4.2-4.5和定理4.1可知$\varphi\equiv_{\langle V'\cup V,\emptyset\rangle}ERes(\varphi,V)$。又$\varphi\equiv_{\langle V,\emptyset\rangle}\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$（由遗忘理论的定义）、$\mathrm{IR}(ERes(\varphi,V),V)$ 和$\mathrm{IR}(\mathrm{F}_{\mathrm{CTL}}(\varphi,V),V)$，因此有$ERes(\varphi,V)\equiv_{\langle V',\emptyset\rangle}\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。 □

这里需要说明的是，有的情况下算法 4.1不能完全替换掉V'中的原子命题，这与CTL不具有均匀插值性[54]一致。否则，若对于任意的CTL公式φ和原子命题集合V，存在一个CTL公式ψ使得$\mathrm{IR}(\psi,V\cup V')$且$\psi\equiv ERes(\varphi,V)$，则通过定理 4.2可知$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$总是存在。这与CTL不具有均匀插值性形成矛盾。

尽管如此，有的CTL公式的遗忘结果总是存在的，如下面的结论所示。

命题 4.6. *给定CTL公式φ，若φ满足下面约束：φ中不包括操作符$Pt\mathscr{T}$（其中$Pt\in\{\mathrm{A},\mathrm{E}\}$且$\mathscr{T}\in\{\mathrm{U},\mathrm{G}\}$），且对于任意的原子命题$p\in V$，若$p$和$\neg p$出现在同一时序算子的范围内；则$\mathrm{ERes}(\varphi,V)\equiv\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。*

证明. 不失一般性地假设$V=\{p\}$。对任意上述所说形式的CTL公式φ，假定$\varphi=\varphi_1\wedge\mathrm{AXEF}\varphi_2$，其中$p\notin Var(\varphi_1)$且$\varphi_2$是一个包含子句$C_1=\neg p\vee\psi_1$和$C_2=p\vee\psi_2$的CNF(conjunctive normal form)公式。φ可以被转换为包含集合$\Pi=\{\top\rightarrow\neg x\vee p\vee\psi_1,\top\rightarrow\neg x\vee\neg p\vee\psi_2\}$的

子句的集合Σ，其中x为新引入的原子命题，ψ_i（$i=1,2$）为经典子句。除此之外，Σ中不包含其他含有p的公式。

由归结过程可产生子句$\top \rightarrow \neg \vee \psi_1 \vee \psi_2$，由定理 4.1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，因而不会产生引入嵌套原子命题（同时出现在$\rightarrow$两边的原子命题），此时对新引入的其余的原子命题都可使用定理 4.1。因此，由定理 4.2可知$ERes(\varphi,V) \equiv \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。 □

4.4 算法的可终止性和计算复杂性

已有结果表明，转换过程和归结过程会终止[61]。此外，*Remove_atoms*、*Remove_index*、替换V'中的原子命题和T_{CTL}过程都会终止，因此算法 4.1会终止。其具体的时间和空间复杂性如下面的结论所示。

命题 4.7. *给定CTL公式φ和原子命题集合$V \subseteq \mathscr{A}$，令$(T_\varphi,V',I)=Transform(\varphi)$. 算法 4.1的时间和空间复杂性为$O((m+1)2^{4(n+n')})$，其中$n=|Var(\varphi)|$、$n'=|V'|$且$m=|I|$。*

证明. 由于*Transform*过程在多项时间内完成，*Remove_atoms*、*Remove_index*、T_{CTL}过程和替换V'中的原子命题最多都只需要扫描*Resolution*(φ)集合就能完成。因此，算法的复杂性主要依赖于归结过程。

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

在上述结论中值得注意的是m的大小不会大于公式φ中出现的时序算子的个数，因此可以得出算法 4.1的计算复杂性仅与出现在φ的原子命题个数和时序算子的个数相关。

4.5 实验与分析

本节给出所提出的基于归结的遗忘计算的实验结果，并分析实验结果。本章提出的算法 4.1用Prolog语言实现，并在Linux服务器上进行了实验，该服务器是具有8个Intel核和32GB内存的i7CPU，其锁频和主频分别为4770 K，3.50 GHz。每次计算的时间限制到1200秒以内。实验分析有两个部分：(1) 在随机数据集和标准数据集上的遗忘；(2) 在随机数据集上命题逻辑公式和CTL公式的SNC计算。所有的实验数据和实验结果都可以从网上获取[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。

[3] https://github.com/fengrenyan/forgetting-in-CTL/tree/main/Appendix

表 4.1: 计算$ERes(\varphi,V)$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

4.5.1　遗忘实验分析

这部的分实验数据分为两组：一组是来源于标准数据集，一组是随机生成的数据。标准数据集来源于CTL-RP [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，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$ERes(\varphi,V)$所使用的CPU时间（单位：秒(s)，不指出时也默认为秒）如表4.1所示，其中$\varphi \in \{s001, s002, s003, s004, s001-3\}$，$|V| \in \{1,2,3,4\}$。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。

除了上述标准数据集中的公式，我们也做了具有以下形式的公式的遗忘实验:

$$\varphi = \varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$$

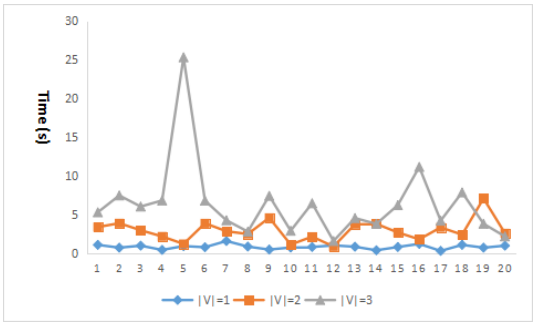
其中φ_i（$i = 1,2,3$）是随机产生的定义在原子命题集合$\mathscr{A}$上的3-CNF公式，且$|\varphi_1| = |\varphi_2| = |\varphi_3|$、$|\mathscr{A}| = 4$。这里做了六组计算$\mathrm{F}_{\mathrm{CTL}}(\varphi,V)$的实验，即$|\varphi_i| \in \{12,16\}$和$|V| \in \{1,2,3\}$的组合，每一组有二十个公式。

$|\varphi_i| = 12$时的实验结果如图 4.2所示，图 4.2(a) 展示了计算遗忘所需要的时间，图 4.2(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 4.2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 4.3所示，其与$|\varphi_i| = 12$时有相似的结果。

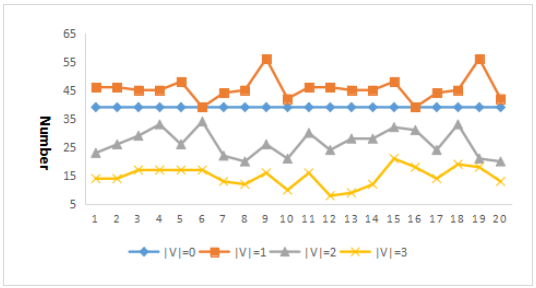
4.5.2　SNC计算结果分析

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：分

[4] https://sourceforge.net/projects/ctlrp/

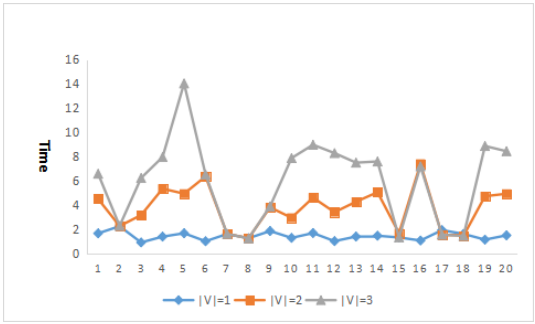


(a) 计算遗忘需要的CUP时间

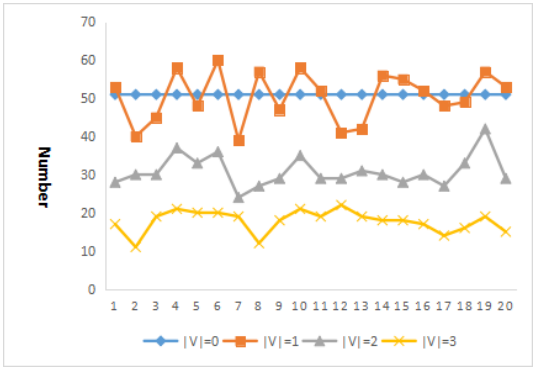


(b) SNF^{g}_{CTL}子句的个数

图 4.2: $\varphi_i = 12$时的计算结果



(a) 计算遗忘需要的CUP时间



(b) SNF^{g}_{CTL}子句的个数

图 4.3: $\varphi_i = 16$时的计算结果

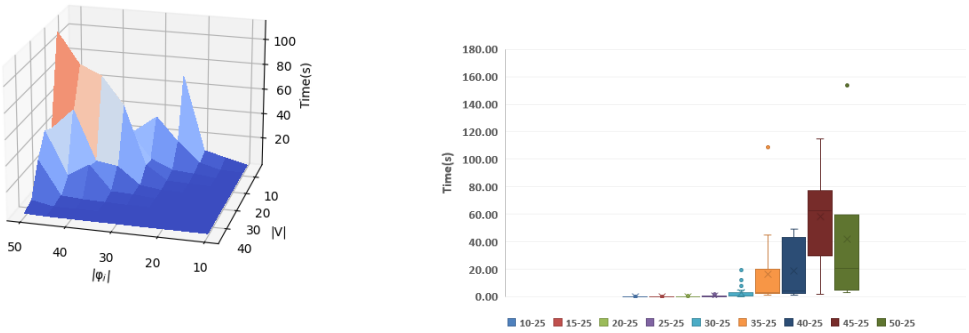
别计算经典命题公式和CTL公式的SNC，即：计算q在V和$\varphi \wedge q$上的SNC（$F_{CTL}(\varphi \wedge q, Var(\varphi) - V \cup \{q\})$），其中$V \subseteq Var(\varphi)$、$q \in Var(\varphi \wedge q) - V$。这些公式$\varphi$都是随机生成的定义在原子命题集合$\mathscr{A}$上的公式、$V$也是在计算过程中随机生成的、$q \notin Var(\varphi)$是一个固定的原子命题且$|\mathscr{A}| = 50$。

首先测试随机3-CNF命题公式。令$|V|$的取值范围为$\{5, 10, \ldots, 40, 45\}$，3-CNF公式的子句个数$nc$范围为$\{10, 15, \ldots, 45, 50\}$。在每种情形当中，计算20个随机实例$(\varphi, q, V)$：$\varphi$为$\mathscr{A}$上的公式，且$V \subseteq Var(\varphi)$。计算SNC的平均CPU时间如图 4.4所示。

从图 4.4(a)可看出，随着$|\varphi|$增大或$|V|$的减小时间消耗越大。直观地说，越大的$|\varphi|$或者越小的$|V|$意味着$F_{CTL}(\varphi, \overline{V})$更难计算。这与上一小节中的结论相符合。图 4.4(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：

$$\varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$$

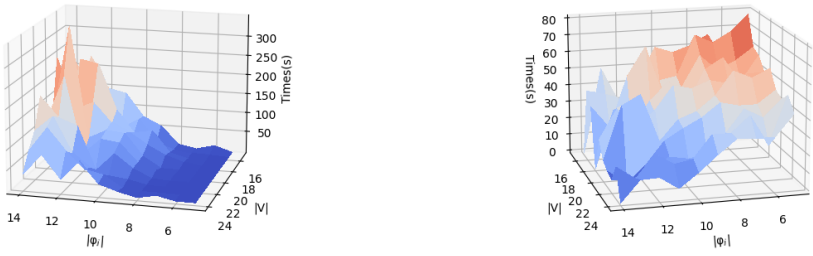


(a) 平均CPU时间(s)　　(b) $|V|=25$时所使用CPU时间箱线图

图 4.4: 计算3-CNF公式的SNC所需的CPU时间情况

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 4.5(a)展示了每种情形计算40个实例的SNC的平均CPU时间。与命题公式的情形相似，越大的$|\varphi|$或者越小的$|V|$意味着$\mathrm{F}_{\mathrm{CTL}}(\varphi,\overline{V})$更难计算。此外，图 4.5(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) SNC存在的占比(%)

图 4.5: CTL下计算SNC的性能情况.

综上所述，算法 4.1大多数情况下能计算出SNC（WSC），且当需要遗忘的原子个数很少或公式长度较小的时候计算效率很高。

4.6 本章小结

kdjfskdfjwiequroewqrjklsef

本章针对差分隐私数据发布中的隐私保护问题，借鉴Shannon信息论基本通信模型，在隐私与数据效用的度量基础上，利用率失真理论构建了隐私与失真的最优化模型，研究了满足数据质量损失约束条件的互信息隐私最优化机制问题，给出差分隐私数据发布的互信息隐私优化模型。随后，在数据发布的隐私通信模型中，考虑了隐私攻击者可能具有的关联辅助背景知识对互信息隐私泄露的影响，提出了基于联合事件的最小化互信息隐私泄露的优化模型。对于模型求解计算信道条件概率分布的问题，利用拉格朗日乘子法和凸问题的KKT最优性条件，给出解的参量表达式。在迭代算法计算方面，基于率失真函数求解的Blahut-Arimoto算法设计了最优化信道条件概率的迭代求解算法。最后，通过真实数据集上的实验结果，阐述了本章理论部分的研究成果，分析了所提出模型及算法的有效性及优势。

第五章　基于模型的方法计算CTL下的遗忘

遗忘理论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[54]，这就表明*CTL*中的遗忘理论不是封闭的[1]。此时，探索*CTL*下遗忘理论封闭的情形对深入引用遗忘理论有重要意义。为此，本章首先提出有限初始K-结构的特征公式；其次，表明*CTL*公式的遗忘结果在此情形下可以表示成其模型的特征公式的吸取；最后，提出一种基于模型的方法计算遗忘，且探索了如何使用遗忘计算最弱充分条件和知识更新。

5.1　引言

计算树逻辑是由Clarke和Emerson提出的一种分支时间时序逻辑，它能很好的描述并发系统的一些性质。Emerson和Halpern证明CTL具有小模型属性：如果一个公式是可满足的，那么它在一个小的有限模型下是可满足的[86]。具体说来，对于给定的CTL公式φ，如果公式的长度[2]为n（记为：$|\varphi| = n$），则存在一个状态数为$n8^n$的初始K-结构$(\mathscr{M}, s_0)$使得$(\mathscr{M}, s_0) \models \varphi$。

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

下文将表明在这一约束条件下CTL中的遗忘是封闭的。

本章其余部分组织如下：首先，第**??**节介绍本章的基本定义、系统模型，提出研究问题。其次，第**??**节阐述本章中提出的优化模型和ORRP方案。进一步，第**??**节给出所提方案在理论上的性能分析。随后，第**??**节给出真实数据集上的实验结果。最后，在第5.5节中进行本章工作总结。

5.2　描述初始K-结构

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

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

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

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K-结构的特征公式。

5.2.1 计算树的V-互模拟

****

引理 5.1. 给定原子命题集合$V \subseteq \mathscr{A}$和K-结构，其中$i = 1,2$且$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$为有限初始结构。$(s_1, s_2) \in \mathscr{B}$当且仅当$s_1 \leftrightarrow_V s_2$。

证明. 值得注意的是对任意的$n \geq 0$，都有$\mathscr{B}_{n+1} \subseteq \mathscr{B}_n$。又因为$\mathscr{B}_0 \subseteq S_1 \times S_2$ 是有限的，因而存在一个数k使得$\mathscr{B}_{k+1} = \mathscr{B}_k = \mathscr{B}$。

（1）证明若$(s_1, s_2) \in \mathscr{B}$则$s_1 \leftrightarrow_V s_2$。显然，$(s_1, s_2) \in \mathscr{B}$。所以，只需要证明$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_1$之间的一个$V$-互模拟关系。下面证明对任意的$r_1 \in S_1$和$r_2 \in S_2$，$(r_1, r_2) \in \mathscr{B}$当且仅当

(a) $L_1(r_1) - V = L_2(r_2) - V$;

(b) $\forall w_1 \in S_1$, if $(r_1, w_1) \in R_1$, then $\exists w_2 \in S_2$ s.t. $(r_2, w_2) \in R_2$ and $(w_1, w_2) \in \mathscr{B}$; and

(c) $\forall w_2 \in S_2$, if $(r_2, w_2) \in R_2$, then $\exists w_1 \in S_1$ s.t. $(r_1, w_1) \in R_1$ and $(w_1, w_2) \in \mathscr{B}$.

$(\Rightarrow)$ $(r_1, r_2) \in \mathscr{B}$
$\Rightarrow \forall n \geq 0$，$(r_1, r_2) \in \mathscr{B}_n$
$\Rightarrow (r_1, r_2) \in \mathscr{B}_0$且$\forall n > 0$，$(r_1, r_2) \in \mathscr{B}_n$
$\Rightarrow L_1(r_1) - V = L_2(r_2) - V$ （因此，(a)成立），

且$\forall n \geq 0$，$(r_1, r_2) \in \mathscr{B}_{n+1}$意味着下面几点成立：

- $L_1(r_1) - V = L_2(r_2) - V$;
- $\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$(w_1, w_2) \in \mathscr{B}_n$；且
- $\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$(w_1, w_2) \in \mathscr{B}_n$。

因为存在一个数k使得$\mathscr{B}_{k+1} = \mathscr{B}_k = \mathscr{B}$，所以对这样的$k$有$(r_1, r_2) \in \mathscr{B}_{k+1}$使得：

- $L_1(r_1) - V = L_2(r_2) - V$;
- $\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$(w_1, w_2) \in \mathscr{B}_k$
  $\Rightarrow \forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$(w_1, w_2) \in \mathscr{B}$ （因此，(b)成立）。

- $\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$(w_1, w_2) \in \mathscr{B}_k$
  $\Rightarrow \forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$(w_1, w_2) \in \mathscr{B}$ （因此，(c)成立）。

因此，$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_1$之间的一个V-互模拟关系。又因为$(s_1, s_2) \in \mathscr{B}$，所以$s_1 \leftrightarrow_V s_2$。

($\Leftarrow$) 假定(a)、(b)和(c)成立，这里证明$(r_1, r_2) \in \mathscr{B}$，即：对于任意的$n \geq 0$都有$(r_1, r_2) \in \mathscr{B}_n$。

(1) 由(a)可知$(r_1, r_2) \in \mathscr{B}_0$，即：$L_1(r_1) - V = L_2(r_2) - V$。

(2) 由(b)可知：$\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$\forall n \geq 0$都有$(w_1, w_2) \in \mathscr{B}_n$
$\Rightarrow \forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$\forall n > 0$都有$(w_1, w_2) \in \mathscr{B}_{n-1}$
$\Rightarrow \forall n > 0$，$\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$(w_1, w_2) \in \mathscr{B}_{n-1}$。

(3) 由(c)可知：$\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$\forall n \geq 0$都有$(w_1, w_2) \in \mathscr{B}_n$
$\Rightarrow \forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$\forall n > 0$都有$(w_1, w_2) \in \mathscr{B}_{n-1}$
$\Rightarrow \forall n > 0$，$\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$(w_1, w_2) \in \mathscr{B}_{n-1}$。

因此，$\forall n > 0$有：

- $(r_1, r_2) \in \mathscr{B}_0$;
- $\forall w_1 \in S_1$，若$(r_1, w_1) \in R_1$，则$\exists w_2 \in S_2$使得$(r_2, w_2) \in R_2$且$(w_1, w_2) \in \mathscr{B}_{n-1}$；且
- $\forall w_2 \in S_2$，若$(r_2, w_2) \in R_2$，则$\exists w_1 \in S_1$使得$(r_1, w_1) \in R_1$且$(w_1, w_2) \in \mathscr{B}_{n-1}$。

所以对于任意的$n \geq 0$都有$(r_1, r_2) \in \mathscr{B}$，即：$(r_1, r_2) \in \mathscr{B}$。

（ii）由$s_1 \leftrightarrow_V s_2$可知存在$\mathscr{M}_1$和$\mathscr{M}_1$之间的一个$V$-互模拟关系$\mathscr{R}$使得$(s_1, s_2) \in \mathscr{R}$。令$\mathscr{B} = \mathscr{R}$，显然对任意的$n \geq 0$有$(s_1, s_2) \in \mathscr{B}_n$。

□

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

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

在上述定义中，当$n=0$时，只需考虑第一个条件。

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

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

证明. 这里从下面两个方面来证明这一结论：

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

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

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

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

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

5.2.2　计算树的特征公式

由上面小节的讨论可知，V-互模拟可以将计算树分别开来[3]。本节讨论如何使用CTL公式描述一棵计算树，且表明具有（或没有）V-互模拟关系之间的计算树的特征公式又有怎么样的关系。为此，首先给出计算树的特征公式的定义。

定义 5.1. 给定原子命题集合$V \subseteq \mathscr{A}$、初始结构$\mathscr{M} = (S,R,L,s_0)$和状态$s \in S$。定义在$V$上的计算树$Tr_n(s)$的特征公式（记为$\mathscr{F}_V(Tr_n(s))$，$n \geq 0$）被递归定义如下：

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

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

由定义 5.1可知，计算树的特征公式从三个方面展示了计算树的信息：（1）只考虑V中的原子命题；（2）突出了树节点的内容，即：对于任意原子命题$p \in V$，若p在节点的标签中，则其正出现在特征公式中，否则负出现在特征公式中；（3）公式中的时序算子表示了状态之间的转换关系。通俗地讲，$\mathscr{F}_V(\mathrm{Tr}_0(s))$表明了节点$s$的在$V$上的内容；EX的合取部分和AX部分保证以$s$的每个直接后继状态$s'$为根深度为$k$的计算树都有一个CTL公式来描述。

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

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

证明. 通过归纳计算树的深度n来证明。

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

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

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

[3]Similar approaches has been taken in the literature e.g., in [87], a class (namely, $\equiv_{\overline{k}}$-class) of structures of monadic formulas has been characterized by Hintikka formulas [88]. Another example is Yankov-Fine construction in [89].

R'。显然，由计算树的特征公式可知：

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

and

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

又因为$\mathrm{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k+1}(s'_{k-1})$，所以对任意的$(s_{k-1},s_k)\in R$存在$(s'_{k-1},s'_k)\in R'$使得$\mathrm{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k}(s'_k)$，且对任意的$(s'_{k-1},s'_k)\in R'$存在$(s_{k-1},s_k)\in R$使得$\mathrm{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \mathrm{Tr}_{n-k}(s'_k)$。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，对于初始结构$\mathscr{M}$上的状态s和s'，若$(\mathscr{M},s)$是定义在V上的根为s'深度为n的计算树的特征公式，则s和s'至少属于$\mathscr{B}_n$，即：s和s'能想互模拟至少到第n层深度。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

□

5.2.3　初始κ-结构的特征公式

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

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

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

由$ch(\mathscr{M},V)$定义可知，对于任意的$\mathscr{M}$和V，$ch(\mathscr{M},V)$总是存在的，这体现在两个方面：（1）若$\mathscr{M}$是V-可区分的，存在两个状态s和s'是V-可区分的，由命题 5.2可知，存在一个数k使得$\mathrm{dis}_V(\mathscr{M},s,s',k)$成立；（2）若对于任意$k \geq 0$和$\mathscr{M}$上的两个状态$s$和$s'$都有$(s,s') \in \mathscr{B}_k$且$\mathscr{B}_k = \mathscr{B}_{k+1}$，则$ch(\mathscr{M},V) = 0$。

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

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

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

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

证明. (i) 这由引理 5.3易知。

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

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

由此，可定义初始K-结构的特征公式如下。

定义 5.2 (特征公式). 给定原子命题集合$V \subseteq \mathscr{A}he$和初始K-结构$\mathscr{K} = (\mathscr{M},s_0)$，其中$c = ch(\mathscr{M},V)$。对任意$\mathscr{M}$上得状态$s' \in S$，记$T(s') = \mathscr{F}_V(Tr_c(s'))$。则$\mathscr{K}$关于$V$的特征公式$\mathscr{F}_V(\mathscr{K})$为：

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

有时为了凸显出初始结构及其初始状态，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，$T(s_0)$确保了初始K-结构的初始状态被CTL公式描述；其余部分表明了结构$\mathscr{M}$上状态之间的转换关系。下面的例子给出了计算特征公式的一般步骤：

例 5.1 (Continued from Example **??**). 考虑图 5.1中左边的初始K-结构$\mathscr{K}_2 = (\mathscr{M},s_0)$（其最初出现在图 **??**中）。左边的为$\mathscr{M}$上的四棵计算树：从左到右表示以$s_0$为根、深度分别

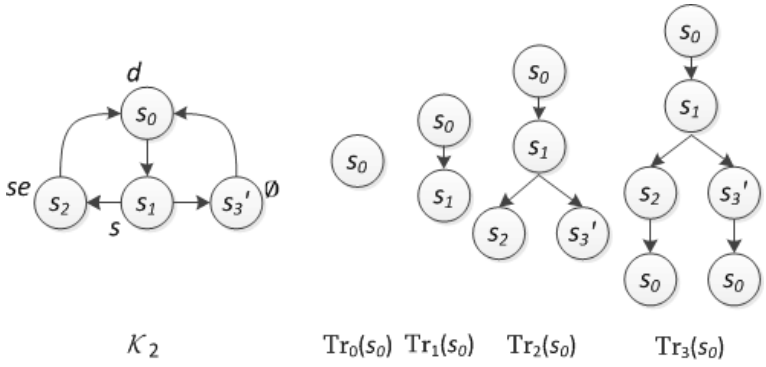


图 5.1: 左图为初始K-结构$\mathscr{K}_2$ (源于图 **??**)；右图：从左到右表示以s_0为根、深度分别为0、1、2和3的计算树（为简化图，计算树的标签没有给出，但是每个树节点的标签可从$\mathscr{K}_2$找到。）

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

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

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

所以，可以由以下步骤计算$\mathscr{K}_2$关于V的特征公式：

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

下面的定理表示上述定义的特征公式确实描述了一个初始K-结构，此时对系统结构的操作就可转换为对其特征公式的操作，如下文将要讲的给定系统下的最弱充分条件计算。更直观地说，特征公式保持了给定初始K-结构在原子命题集合V上的所有特性，即：具有$\overline{V}$-互模拟的两个初始K-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，下面证明首先证明$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathscr{M},s_0)$。

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

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

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

- 若$(\mathscr{M},s_0) \not\models \mathscr{F}_V(\mathrm{Tr}_c(s))$，显然$(\mathscr{M},s_0) \models G(\mathscr{M},s)$；
- 若$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s))$：

  $(\mathscr{M},s_0) \models \mathscr{F}_V(\mathrm{Tr}_c(s))$

  $\Rightarrow s_0 \leftrightarrow_{\overline{V}} s$ （引理 5.4）

  $\forall (s,s_1) \in R$：

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

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

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

  $\forall (s,s_1) \in R$：

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

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

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

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

对任意其他能从s_0可达的状态s'，都可以类似地证明$(\mathscr{M},s') \models G(\mathscr{M},s)$。因此，对任意的$s \in S$，$(\mathscr{M},s_0) \models \text{AG}\, G(\mathscr{M},s)$，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$(\mathscr{M}',s'_0) \models \mathscr{F}_V(M,s_0)$。因为$(\mathscr{M},s_0) \models \mathscr{F}_V(\mathscr{M},s_0)$且$\text{IR}(\mathscr{F}_V(\mathscr{M},s_0),\overline{V})$，由定理 3.1可知$(\mathscr{M}',s'_0) \models \mathscr{F}_V(M,s_0)$。

($\Rightarrow$) 证明：若$(\mathscr{M}',s'_0) \models \mathscr{F}_V(M,s_0)$，则$s_0 \leftrightarrow_{\overline{V}} s'_0$。为此，下面证明对任意的$n \geq 0$，$Tr_n(s_0) \leftrightarrow_{\overline{V}} Tr_n(s'_0)$。

基始（$n=0$）：由特征公式的定义，显然$Tr_0(s_0) \equiv Tr_0(s'_0)$成立。

归纳步骤（$n>0$）：假定对任意的$k>0$都有$\text{Tr}_k(s_0) \leftrightarrow_{\overline{V}} \text{Tr}_k(s'_0)$，下面证明$\text{Tr}_{k+1}(s_0) \leftrightarrow_{\overline{V}} \text{Tr}_{k+1}(s'_0)$。令$(s_0,s_1),(s_1,s_2),\ldots,(s_{k-1},s_k) \in R$且$(s'_0,s'_1),(s'_1,s'_2),\ldots,(s'_{k-1},s'_k) \in R'$，即对于任意的$0 \leq i \leq k-1$，$s_{i+1}$（$s'_{i+1}$）是$s_i$ (s'_i)的直接后继状态。由归纳假设可知，只需证明$\text{Tr}_1(s_k) \leftrightarrow_{\overline{V}} \text{Tr}_1(s'_k)$。

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

在讨论其他点时，首先考虑下面事实（**fact**）：

$(\mathscr{M}',s'_0) \models \mathscr{F}_V(\mathscr{M},s_0)$

$\Rightarrow \forall s' \in S'$，$\forall s \in S$，

$(\mathscr{M}',s') \models \mathscr{F}_V(\text{Tr}_c(s)) \rightarrow \left(\bigwedge_{(s,s_1)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_1))\right) \wedge \text{AX}\left(\bigvee_{(s,s_1)\in R} \mathscr{F}_V(\text{Tr}_c(s_1))\right)$

(I) $(\mathscr{M}',s'_0) \models \mathscr{F}_V(\text{Tr}_c(s_0)) \rightarrow \left(\bigwedge_{(s_0,s_1)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_1))\right) \wedge \text{AX}\left(\bigvee_{(s_0,s_1)\in R} \mathscr{F}_V(\text{Tr}_c(s_1))\right)$

(II) $(\mathscr{M}',s'_0) \models \mathscr{F}_V(\text{Tr}_c(s_0)))$ (已知)

(III) $(\mathscr{M}',s'_0) \models \left(\bigwedge_{(s_0,s_1)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_1))\right) \wedge \text{AX}\left(\bigvee_{(s_0,s_1)\in R} \mathscr{F}_V(\text{Tr}_c(s_1))\right)$ ((I),(II))

(b) 这里证明$\forall(s_k,s_{k+1}) \in R$，存在$(s'_k,s'_{k+1}) \in R'$使得$L(s_{k+1})-\overline{V} = L'(s'_{k+1})-\overline{V}$。

(1) $(\mathscr{M}',s'_0) \models \bigwedge_{(s_0,s_1)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_1))$ (III)

(2) $\forall(s_0,s_1) \in R$，$\exists(s'_0,s'_1) \in R'$ 使得$(\mathscr{M}',s'_1) \models \mathscr{F}_V(\text{Tr}_c(s_1))$ (1)

(3) $\text{Tr}_c(s_1) \leftrightarrow_{\overline{V}} \text{Tr}_c(s'_1)$ ((2), 引理 5.3)

(4) $L(s_1)-\overline{V} = L'(s'_1)-\overline{V}$ ((3), $c \geq 0$)

(5) $(\mathscr{M}',s'_1) \models \mathscr{F}_V(\text{Tr}_c(s_1)) \rightarrow \left(\bigwedge_{(s_1,s_2)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_2))\right) \wedge \text{AX}\left(\bigvee_{(s_1,s_2)\in R} \mathscr{F}_V(\text{Tr}_c(s_2))\right)$ **(fact)**

(6) $(\mathscr{M}',s'_1) \models \left(\bigwedge_{(s_1,s_2)\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_2))\right) \wedge \text{AX}\left(\bigvee_{(s_1,s_2)\in R} \mathscr{F}_V(\text{Tr}_c(s_2))\right)$ ((2), (5))

(7) ……

(8) $(\mathscr{M}',s'_k) \models \left(\bigwedge_{(s_k,s_{k+1})\in R} \text{EX}\mathscr{F}_V(\text{Tr}_c(s_{k+1}))\right) \wedge \text{AX}\left(\bigvee_{(s_k,s_{k+1})\in R} \mathscr{F}_V(\text{Tr}_c(s_{k+1}))\right)$ (与(6)类似)

(9) $\forall(s_k,s_{k+1}) \in R$，$\exists(s'_k,s'_{k+1}) \in R'$ 使得$(\mathscr{M}',s'_{k+1}) \models \mathscr{F}_V(\text{Tr}_c(s_{k+1}))$ (8)

(10) $\text{Tr}_c(s_{k+1}) \leftrightarrow_{\overline{V}} \text{Tr}_c(s'_{k+1})$ ((9), 引理 5.3)

(11) $L(s_{k+1})-\overline{V} = L'(s'_{k+1})-\overline{V}$ ((10), $c \geq 0$)

(c) 这里证明$\forall(s'_k,s'_{k+1})\in R'$，存在$(s_k,s_{k+1})\in R$使得$L(s_{k+1})-\overline{V}=L'(s'_{k+1})-\overline{V}$。

(1) $(\mathscr{M}',s'_k)\models \mathrm{AX}\left(\bigvee_{(s_k,s_{k+1})\in R}\mathscr{F}_V(\mathrm{Tr}_c(s_{k+1}))\right)$ (上面的(8))

(2) $\forall(s'_k,s'_{k+1})\in R'$，$\exists(s_k,s_{k+1})\in R$使得$(\mathscr{M}',s'_{k+1})\models\mathscr{F}_V(\mathrm{Tr}_c(s'_{k+1}))$ (1)

(3) $\mathrm{Tr}_c(s_{k+1})\leftrightarrow_{\overline{V}}\mathrm{Tr}_c(s'_{k+1})$ ((2), 引理 5.3)

(4) $L(s_{k+1})-\overline{V}=L'(s'_{k+1})-\overline{V}$ ((3), $c\geq 0$)

(ii) 由引理 5.2和5.4易知。

□

5.3 遗忘理论的封闭性

当给定的CTL公式的长度（字符的个数）为n，由小模型理论可知定义在状态个数为$k=n8^n$的状态空间$\mathscr{S}=\{s_1,s_2,\ldots,s_k\}$上的初始结构就能保证公式的可满足性[86]。对于其他拥有同样大小的状态空间上的任意初始K-结构，都能在$\mathscr{S}$状态空间上找到一个初始K-结构与之互模拟，且由定理 5.1可知他们有相同的特征公式。因此，只有有限个初始K-结构作为该公式的候选模型。因此下面结论成立。

引理 5.5. *给定CTL公式φ，下面等式成立：*

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

证明. 令$(\mathscr{M}',s'_0)$为φ的模型。由定理 5.1可知$(\mathscr{M}',s'_0)\models\mathscr{F}_{\mathscr{A}}(\mathscr{M}',s'_0)$，则：

$$(\mathscr{M}',s'_0)\models\bigvee_{(\mathscr{M},s_0)\in Mod(\varphi)}\mathscr{F}_{\mathscr{A}}(\mathscr{M},s_0).$$

另一方面，假定$(\mathscr{M}',s'_0)$为$\bigvee_{(\mathscr{M},s_0)\in Mod(\varphi)}\mathscr{F}_{\mathscr{A}}(\mathscr{M},s_0)$的模型。则存在$(\mathscr{M},s_0)\in Mod(\varphi)$使得$(\mathscr{M}',s'_0)\models\mathscr{F}_{\mathscr{A}}(\mathscr{M},s_0)$。由定理 5.1可知$(\mathscr{M},s_0)\leftrightarrow_{\emptyset}(\mathscr{M}',s'_0)$，从而由定理 3.1可知$(\mathscr{M},s_0)$是$\varphi$的一个模型。 □

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

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

在上述的遗忘理论的定义中说明了如果公式ψ的任意一个模型$\mathscr{K}$都能找到φ的一个模型$\mathscr{K}'$使得$\mathscr{K} \leftrightarrow_V \mathscr{K}'$，则称$\psi$为从$\varphi$中遗忘掉$V$中原子命题后得到的结果。为刻画S5逻辑下该概念的直观含义，Zhang等人提出了如下遗忘理论特性——这些特性被称为*遗忘理论公设*（forgetting postulates）[35]。给定CTL公式φ、$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$和原子命题集合$V \subseteq \mathscr{A}$和$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$，遗忘理论公设如下：

(**W**) 弱（Weakening）属性：$\varphi \models \varphi'$；

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

(**NP**) 负支持性（Negative Persistence）：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（Irrelevance）：$\mathrm{IR}(\varphi', V)$

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

定理 5.2 (Representation Theorem)**.** *给定CTL公式φ和φ'，$V \subseteq \mathscr{A}$为原子命题的集合。下面的陈述是等价的：*

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

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

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

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

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

($\Rightarrow$) 对任意$\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$的模型$\mathscr{K}'$
$\Rightarrow \exists \mathscr{K}$使得$\mathscr{K} \models \varphi$且$\mathscr{K} \leftrightarrow_V \mathscr{K}'$ (定义 3.2)
$\Rightarrow \forall \phi$，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 3.2可知$\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\} \models \mathrm{F}_{\mathrm{CTL}}(\varphi,V)$。

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

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

5.4 基于模型的遗忘理论计算方法

5.5 本章小结

本章针对差分隐私数据收集中多维数据处理的隐私脆弱性和有效性问题，为了权衡隐私泄露与数据质量损失，利用信息论的方法提出了有序随机响应扰动(ORRP)方案。首先，对于元组的多维属性使用分治策略思想分解元组属性，构建本地扰动的独立并联信道模型。其次，针对单属性分量的隐私保护问题，基于信息论的度量方法形式化数据质量损失约束前提下最小化隐私信息泄露的最优化模型，用于计算最优概率密度函数(PDF)。然后，以B-A为基本构建模块设计ORRP，利用上述PDF实现有序随机响应，并给出算法描述。最后，给出理论分析和真实数据集上的实验结果。

第六章　μ-演算中的遗忘理论

本章探索μ-演算中的遗忘理论。μ-演算是描述转换系统性质重要逻辑语言，其具有表达能力强的优点：μ-演算是一种表达能力与$S2S$[1]相同的逻辑语言，LTL（线性时序逻辑，*linear temporal logic*）、CLT和CTL^*能表达的属性都能用μ-演算来表示。

已有研究表明μ-演算具有均匀插值性质，这体现了μ-演算下的遗忘理论研究本质上与CTL下的不同。本章首先给出μ-演算下的遗忘理论的定义。其次，表明μ-演算下的遗忘理论是封闭的，这是其与CTL下的遗忘理论的最大的不同。最后，模型检测问题作为形式化验证的重要方法，本章给出μ-演算下遗忘理论的模型检测和推理问题的复杂性结果。

6.1　引言

μ-演算是一种表达能力较强的语言，它能表达CTL不能表示的性质，例如：Kripke结构中有一条路径，在这条路径上的基数状态满足公式$\neg q \wedge \neg p$，但是偶数状态满足$q \wedge p$。这一性质是不能用CTL公式来表示，但是可以用μ-演算公式表示如下：

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

这种情形在日常生活中是很常见的，如：偏序关系$(N, \leq)$（自然数集上的小于等于关系）构成的Kripke结构，其基数节点为基数、偶数节点为偶数。

均匀插值有以下含义：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。均匀插值的定义[66]如下（注意：这里的$Var(\varphi)$表示出现在φ中的原子命题和变元的集合）：

定义 6.1. 给定一个μ-句子φ和原子命题的集合V，φ关于V的均匀插值是满足下列条件的μ-句子θ：

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

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

从直观上来说，φ关于$Var(\varphi)\cap Var(\psi)$的均匀插值是从$\varphi$中“移除”掉不在$Var(\varphi)\cap Var(\psi)$中的原子命题而保留其在$Var(\varphi)\cap Var(\psi)$上的结论得到的结果。这与遗忘有着密切的关系。

再者，在背景知识部分已经指出，任意的μ-演算公式都能转换成吸取μ-公式，在这种形式下的公式的均匀插值是容易计算的，即：

定理 6.1 (定理3.6[66]). *吸取μ-公式φ的均匀插值$\exists p\varphi$与μ-公式$\varphi[p/\top,\neg p/\bot]$等价，其中$\varphi[p/\top,\neg p/\bot]$是同时将$p$及其否定$\neg p$分别用$\top$和$\bot$替换得到。*

尽管上面的定义中使用的$Var(\varphi)$表示出现在φ中的原子命题和变元的集合，但是均匀插值指的是与原子命题相关的部分，即：对任意的μ-句子φ和原子命题p，存在一个μ-句子$\widetilde{\exists}p.\varphi$使得$\widetilde{\exists}p.\varphi$是$\varphi$关于$\{p\}$的一个均匀插值。这表明在讨论均匀插值时，不考虑变元，即：不是φ关于某个变元的均匀插值。

本章将要说明本文所定义的μ-演算下的遗忘与[66]中定义的均匀插值是一对对偶概念，此时本文给出的遗忘的性质无疑也是均匀插值所具有的性质，这为μ-演算的均匀插值的探索提供了另一种思路。此外，借助于均匀插值的计算方法，本文也给出了计算遗忘的方法。这形成了遗忘和均匀插值之间相辅相成的作用。

本章的组织结构如下。首先，给出μ-演算下遗忘的定义；然后，探讨遗忘的一般性质，并给出其与均匀插值的关系；最后给出与遗忘相关的问题的复杂性。

6.2 遗忘的定义

与CTL情形下的遗忘相似，这里先给出V互模拟的定义。不失一般性地，令$\mathscr{M}_i=(S_i,r_i,R_i,L_i)$，$i$为自然数集$\mathbb{N}$中的元素。

定义 6.2 (V-互模拟). *给定原子命题集合$V\subseteq\mathscr{A}$和两个Kripke结构$\mathscr{M}_1$和$\mathscr{M}_2$。若下面几个条件满足，则称$\mathscr{B}\subseteq S_1\times S_2$是$\mathscr{M}_1$和$\mathscr{M}_2$的$V$-互模拟关系：*

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

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$\mathscr{M}_1$ and $\mathscr{M}_2$之间存在一个V-互模拟关系$\mathscr{B}$则称这两个Kripke结构$\mathscr{M}_1$和$\mathscr{M}_2$是V-互模拟的，记为$\mathscr{M}_1\leftrightarrow_V\mathscr{M}_2$，

例 6.1. 如图 6.1中的两个Kripke结构$\mathscr{M}=(S,r,R,L)$ 和$\mathscr{M}'=(S',r',R',L')$，其中:

- $S=\{s_0,s_1,s_2\}$，$S'=\{t_0,t_1\}$，$r=s_0$，$r'=t_0$；
- $R=\{(s_0,s_1),(s_1,s_0),(s_0,s_2),(s_2,s_0)\}$，$R'=\{t_0,t_1\}$；
- $L(s_0)=\{ch,j\}$，$L(s_1)=L(s_2)=\emptyset$，$L'(t_0)=\{j\}$，$L'(t_1)=\emptyset$。

由于$\mathscr{M}$ 和$\mathscr{M}'$之间存在一个二元$\{ch\}$-互模拟$\mathscr{B}=\{(s_0,t_0),(s_1,t_1),(s_2,t_1)\}$ $\mathscr{M}\leftrightarrow_{\{ch\}}\mathscr{M}'$，所以$\mathscr{M}\leftrightarrow\mathscr{M}'$。

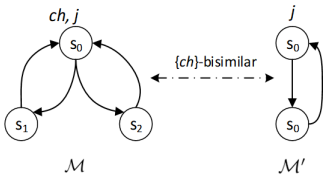


图 6.1: 两个$\{ch\}$-互模拟的Kripke结构

可以容易证明$\leftrightarrow_V$为一个等价关系，此外还有如下性质。

命题 6.1. *令$V,V_1\subseteq\mathscr{A}$为原子命题的集合，$\mathscr{M}_1$、$\mathscr{M}_2$、$\mathscr{M}_3$为Kripke 结构，则：*

(i) *$\leftrightarrow_V$ 是Kripke 结构之间的等价关系;*

(ii) *若$\mathscr{M}_1\leftrightarrow_V\mathscr{M}_2$ 且$\mathscr{M}_2\leftrightarrow_{V_1}\mathscr{M}_3$，则$\mathscr{M}_1\leftrightarrow_{V\cup V_1}\mathscr{M}_3$。*

证明. (i) 这里从自反性、对称性和传递性来证明该关系是一个等价关系。

(1) $\leftrightarrow_V$是自反的。容易检查对任意的Kripke结构$\mathscr{M}$都有$\mathscr{M}\leftrightarrow_V\mathscr{M}$。

(2) $\leftrightarrow_V$ 是对称的。这里证明对任意的$\mathscr{M}_1$ 和$\mathscr{M}_2$，若$\mathscr{M}_1\leftrightarrow_V\mathscr{M}_2$ 则$\mathscr{M}_2\leftrightarrow_V\mathscr{M}_1$。$\mathscr{M}_1$和$\mathscr{M}_2$之间存在$V$-互模拟关系$\mathscr{B}$，构造如下二元关系$\mathscr{B}_1=\{(s,t)\mid(t,s)\in\mathscr{B}\}$，现在从下面几点证明$\mathscr{B}_1$是$\mathscr{M}_2$ 和$\mathscr{M}_1$ 之间的一个V-关系：

- 由于$r_1\mathscr{B}r_2$，所以$r_2\mathscr{B}_1r_1$，
- 对任意的$s\in S_1$ 和$t\in S_2$，若$t\mathscr{B}_1s$，则$s\mathscr{B}t$，因此对于任意的$p\in\mathscr{A}-V$，$p\in L_1(s)$当且仅当$p\in L_2(t)$，且
- 因为$\mathscr{B}$是$\mathscr{M}_1$和$\mathscr{M}_2$之间存在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。

(3) $\leftrightarrow_V$是传递的。这里证明对任意的$\mathscr{M}_1$、$\mathscr{M}_2$、和$\mathscr{M}_3$，若$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$ 和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_3$则$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_3$。假定$\mathscr{M}_1$和$\mathscr{M}_2$之间存在$V$-互模拟关系为$\mathscr{B}_1$，$\mathscr{M}_2$和$\mathscr{M}_3$之间存在$V$-互模拟关系$\mathscr{B}_2$，构造二元关系: $\mathscr{B} = \{(s,z) \mid (s,t) \in \mathscr{B}_1 \text{ and } (t,z) \in \mathscr{B}_2\}$。此时，可以像(2)一样证明$\mathscr{B}$ 是$\mathscr{M}_1$ 和$\mathscr{M}_3$之间的一个V-互模拟关系。因此，$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_3$。

(ii) 为了证明$\mathscr{M}_1 \leftrightarrow_{V\cup V_1} \mathscr{M}_3$，只需找到一个$\mathscr{M}_1$ 和$\mathscr{M}_3$之间的二元$(V \cup V_1)$-互模拟关系$\mathscr{B}$。假定$\mathscr{M}_1$和$\mathscr{M}_2$之间存在V-互模拟关系为$\mathscr{B}_1$，$\mathscr{M}_2$和$\mathscr{M}_3$之间存在V_1-互模拟关系$\mathscr{B}_2$。令$\mathscr{B} = \{(s_1,s_3) \mid (s_1,s_2) \in \mathscr{B}_1 \text{ and } (s_2,s_3) \in \mathscr{B}_2\}$，容易证明$\mathscr{B}$是$\mathscr{M}_1$ 和$\mathscr{M}_3$之间的二元$(V \cup V_1)$-互模拟关系。

□

直观地说，(i)表示$\leftrightarrow_V$是Kripke结构的集合上的自反、对称和传递关系。(ii)表示如果一个Kripke结构和其他两个Kripke结构互相V和V_1互模拟，则这两个Kripke结构$V \cup V_1$-互模拟。这跟CTL情形下的互模拟类似，正如下文将要说到的那样，这一性质有助于证明μ-演算下遗忘的模块属性。

此时，μ-演算下的遗忘如下定义：

定义 6.3 (μ-演算下的遗忘)**.** 令$V \subseteq \mathscr{A}$ 和ϕ 为μ-句子。若下面等式成立，则称V上的μ-句子ψ是从ϕ中遗忘掉V后得到的结果，记为$\mathrm{F}_\mu(\phi,V)$：

$$Mod(\psi) = \{\mathscr{M} \mid \exists \mathscr{M}' \in Mod(\phi) \ \&\ \mathscr{M}' \leftrightarrow_V \mathscr{M}\}.$$

定义6.3表明如果ψ 和ψ' 都是从ϕ中遗忘掉V 中的原子命题得到的结果，则$Mod(\psi) = Mod(\psi')$，也就是说遗忘的结果之间是语义等价的（即有相同的模型）。

D'Agostino 研究了μ-演算下的均匀插值，并指出μ-算具有均匀插值性质[66,90?]。换句话说，这意味着对任意的μ-句子φ 和有限的原子命题的集合$V \subseteq Var(\varphi)$，都存在一个V-无关且与φ最接近的μ-句子$\widetilde{\exists} V\varphi$。值得注意的是，上述定义的遗忘$\mathrm{F}_\mu(\phi,V)$与$\widetilde{\exists} V\varphi$[66]语义等价.

6.3 遗忘的一般属性

这部分展示μ-演算下遗忘的语义属性。特别地，这里将证明上述μ-演算下的遗忘的定义与遗忘的那几条规则具有“当且仅当的关系”，且从任意μ-句子中遗忘掉任意原子命题的集合的结果总是一个μ-句子。此外，也研究了遗忘算子的代数属性，包括模块性（modularity）、交换性（commutativity）和同质性（homogeneity）。

定理 6.2. 给定原子命题$q \in \mathscr{A}$ 和μ-句子ϕ，则存在一个μ-句子ψ 使得$IR(\psi,\{q\})$ 且$\psi \equiv \mathrm{F}_\mu(\phi,\{q\})$。

证明. 已有结果表明，对任意的μ-句子ϕ和和原子命题p，存在一个$\{p\}$-无关的μ-句子ϕ'（即IR$(\phi',\{p\})$）使得（定理3.1[90]）：

$$\mathscr{M} \models \phi' \text{ 当且仅当} \exists \mathscr{M}' \in \phi \text{ 使得} \mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}'.$$

这与本文遗忘的定义一致，因此上述结论成立。 □

与模态S5和CTL情形类似，下面给出μ-演算下的遗忘的基本准则：

(W) 削弱（Weakening）：$\varphi \models \varphi'$;

(PP) 正支持（Positive Persistence）：对任意的μ-句子η，若IR(η,V) 和$\varphi \models \eta$ 则$\varphi' \models \eta$；

(NP) 负支持（Negative Persistence）：对任意的μ-句子η，若IR(η,V) 和$\varphi \not\models \eta$ 则$\varphi' \not\models \eta$；

(IR) 无关性（Irrelevance）：IR(φ',V)。

其中$V \subseteq \mathscr{A}$、φ 为μ-句子、φ' 是从φ中遗忘掉V后得到的结果。

定理 6.3 (表达性定理)**.** 给定μ-句子φ、φ' 和ϕ，$V \subseteq \mathscr{A}$为原子命题的集合。下面的几个陈述是等价的：

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

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

(iii) 若φ、φ' 及V和*(i)*、*(ii)*中的符号表示相同公式和原子命题的集合，则***(W)***、***(PP)***、***(NP)*** 和***(IR)*** 成立。

证明. $(i) \Leftrightarrow (ii)$. 为了证明这一结论成立，只需证明：

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

$(\Rightarrow)$ 对$\mathrm{F}_\mu(\varphi,V)$的任意模型$\mathscr{M}'$
$\Rightarrow \exists \mathscr{M}$ 使得$\mathscr{M} \models \varphi$ 和$\mathscr{M} \leftrightarrow_V \mathscr{M}'$ (定义 6.3)
$\Rightarrow$ 对于任意与V-无关且$\varphi \models \phi$的ϕ 都有$\mathscr{M}' \models \phi$
$\Rightarrow \mathscr{M}' \models \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\}$

$(\Leftarrow)$ 由于IR$(\mathrm{F}_\mu(\varphi,V),V)$和$\varphi \models \mathrm{F}_\mu(\varphi,V)$，由定义 6.3可知$\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi,V)\} \models \mathrm{F}_\mu(\varphi,V)$。

$(ii) \Rightarrow (iii)$. 为了方便，令$A = \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi, V)\}$。首先，对任意的$A$中的公式$\phi'$都有$\mathrm{IR}(\phi', V)$，所以有$\mathrm{IR}(A, V)$。因此，$\mathrm{IR}(\varphi', V)$。其次，对任意的$\phi' \in A$，都有$\varphi \models \phi'$，所以$\varphi \models \varphi'$。第三，$\forall \phi$ 且$\mathrm{IR}(\phi, V)$，若$\varphi \models \phi$ 则$\phi \in A$，因而$\varphi' \models \phi$。最后，$\forall \phi$ 且$\mathrm{IR}(\phi, V)$，若$\varphi \not\models \phi$ 则$\phi \notin A$。因此，由定义 6.3和V-无关性可知$\varphi' \not\models \phi$。

$(iii) \Rightarrow (ii)$. (1) $\varphi' \models \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi, V)\}$ ((**PP**))

(2) $\{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi, V)\} \models \varphi'$ ((**W**)，(**IR**))

$\Rightarrow \varphi' \equiv \{\phi \mid \varphi \models \phi, \mathrm{IR}(\phi, V)\}$ ((1)，(2)). □

定理 6.3 表明μ-演算下的遗忘跟其基本准则形成了一个“当且仅当”的关系：基本准则能描述遗忘的结果，遗忘的结果具有基本准则里的性质。这与S5和CTL下的情形相同。

除了上述的表达性定理，后文将说明准则(**IR**)对计算SNC 和WSC是重要的。对于μ-句子$\psi = \varphi \wedge (q \leftrightarrow \alpha)$，$\varphi \wedge \alpha$ 是$\{q\}$-无关的，则从ψ中遗忘掉q得到的结果是φ。正如将在第 八中展示，这一性质有助于将任意公式的SNC（WSC）转换为命题下的SNC（WSC）。这一性质可形式化如下：

引理 6.1. *令φ 和α为两个μ-句子，q为原子命题且$q \notin Var(\varphi) \cup Var(\alpha)$。则$\mathrm{F}_\mu(\varphi \wedge (q \leftrightarrow \alpha), q) \equiv \varphi$。*

证明. 令$\varphi' = \varphi \wedge (q \leftrightarrow \alpha)$。对$\mathrm{F}_\mu(\varphi', q)$的任意一个模型$\mathscr{M}$，存在一个Kripke结构$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_{\{q\}} \mathscr{M}'$ 和$\mathscr{M}' \models \varphi'$。显然有$\mathscr{M}' \models \varphi$，又因为$\mathrm{IR}(\varphi, \{q\})$ 和$\mathscr{M} \leftrightarrow_{\{q\}} \mathscr{M}'$，所以$\mathscr{M} \models \varphi$。

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

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

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

正如在第 一中所说的，遗忘在经典命题逻辑中首先被提出，并应用于各种领域。这里给出经典命题逻辑与μ-演算下的遗忘之间的联系。

首先回忆一下从命题公式φ中遗忘原子命题p得到的结果为$Forget(\varphi, \{p\}) \equiv \varphi[p/\bot] \vee \varphi[p/\top]$，且$Forget(\varphi, V \cup \{p\})$ 被递归地定义为：$Forget(Forget(\varphi, \{p\}), V)$，其中$Forget(\varphi, \emptyset) = \varphi$。此外，对于给定的Kripke结构$\mathscr{M} = (S, r, R, L)$和命题公式$\psi$，$\mathscr{M} \models \psi$当且仅当$L(r) \models \psi$。经典命题逻辑与$\mu$-演算下的遗忘之间的联系如下：

定理 6.4. *Let φ be a PL formula and $V \subseteq \mathscr{A}$, then*

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

证明. 令$\mathscr{M} = (S, r, R, L)$ 和$\mathscr{M}' = (S', r', R', L')$为Kripke结构。

($\Rightarrow$) 对任意的$\mathscr{M} \in Mod(\mathrm{F}_\mu(\varphi, V))$

$\Rightarrow$ 由定义 3.2可知$\exists \mathscr{M}' \in Mod(\varphi)$ 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)

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

$\Rightarrow \exists \mathscr{M}' \in Mod(\varphi)$ 使得$\forall p \in \mathscr{A} - V$，$p \in L(r)$ 当且仅当$p \in L'(r')$　　(*Forget*的定义)

如下构造Kripke 结构$\mathscr{M}_1 = (S_1, r_1, R_1, L_1)$：

* $S_1 = (S - \{r\}) \cup \{r_1\}$,

* 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 6.4表明μ-演算下的遗忘是命题逻辑的遗忘的扩展，这提示我们是否命题情形下的遗忘拥有的性质μ-演算下的遗忘也具有。下面的性质在命题逻辑、S5[35] 和CTL中都成立，接下来也证明其在μ-演算中也成立。

命题 6.2. 给定μ-句子φ、φ_i和ψ_i $(i = 1, 2)$，$V \subseteq \mathscr{A}$为原子命题的集合。则：

(i) $\mathrm{F}_\mu(\varphi, V)$ 是可满足的当且仅当φ是可满足的；

(ii) 若$\varphi_1 \equiv \varphi_2$，则$\mathrm{F}_\mu(\varphi_1, V) \equiv \mathrm{F}_\mu(\varphi_2, V)$；

(iii) 若$\varphi_1 \models \varphi_2$，则$\mathrm{F}_\mu(\varphi_1, V) \models \mathrm{F}_\mu(\varphi_2, V)$；

(iv) $\mathrm{F}_\mu(\psi_1 \vee \psi_2, V) \equiv \mathrm{F}_\mu(\psi_1, V) \vee \mathrm{F}_\mu(\psi_2, V)$,

(v) $\mathrm{F}_\mu(\psi_1 \wedge \psi_2, V) \models \mathrm{F}_\mu(\psi_1, V) \wedge \mathrm{F}_\mu(\psi_2, V)$。

证明. (i) ($\Rightarrow$) 假设$\mathscr{M}$ 是$\mathrm{F}_\mu(\varphi,V)$的模型，由$\mathrm{F}_\mu$的定义可知，存在$\varphi$的一个模型$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$。

($\Leftarrow$) 假定$\mathscr{M}$是φ的模型，则存在一个Kripke 结构$\mathscr{M}'$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，因此由$\mathrm{F}_\mu$的定义可知$\mathscr{M}' \models \mathrm{F}_\mu(\varphi,V)$。

(ii) 和(iii) 可以类似地证明。

(iv) ($\Rightarrow$) $\forall \mathscr{M} \in Mod(\mathrm{F}_\mu(\psi_1 \vee \psi_2,V))$
$\Rightarrow \exists \mathscr{M}' \in Mod(\psi_1 \vee \psi_2)$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$和$\mathscr{M}' \models \psi_1$（或$\mathscr{M}' \models \psi_2$）
$\Rightarrow \exists \mathscr{M}_1 \in Mod(\mathrm{F}_\mu(\psi_1,V))$ 使得$\mathscr{M}' \leftrightarrow_V \mathscr{M}_1$，或者$\exists \mathscr{M}_2 \in Mod(\mathrm{F}_\mu(\psi_2,V))$ 使得$\mathscr{M}' \leftrightarrow_V \mathscr{M}_2$
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1,V) \vee \mathrm{F}_\mu(\psi_2,V)$。

($\Leftarrow$) $\forall \mathscr{M} \in Mod(\mathrm{F}_\mu(\psi_1,V) \vee \mathrm{F}_\mu(\psi_2,V))$
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1,V)$ 或$\mathscr{M} \models \mathrm{F}_\mu(\psi_2,V)$
$\Rightarrow \exists \mathscr{M}_1$ 使得$\mathscr{M} \leftrightarrow_V \mathscr{M}_1$，且$\mathscr{M}_1 \models \psi_1$ 或者$\mathscr{M}_1 \models \psi_2$ （定义 6.3）
$\Rightarrow \mathscr{M}_1 \models \psi_1 \vee \psi_2$
$\Rightarrow \exists \mathscr{M}_2$使得$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$和$\mathscr{M}_2 \models \mathrm{F}_\mu(\psi_1 \vee \psi_2,V)$
$\Rightarrow \mathscr{M} \leftrightarrow_V \mathscr{M}_2$ （命题 6.1）
$\Rightarrow \mathscr{M} \models \mathrm{F}_\mu(\psi_1 \vee \psi_2,V)$ （定义 6.3）。

(v)可以像(iv)一样证明。 □

命题 6.2(i) 表明从一个μ-句子中遗忘掉一些原子命题不影响该句子的可满足性；从(ii)可以看出，如果两个句子是等价的，则他们遗忘相同原子命题得到的结果是等价的；(iv)指出吸取公式$\varphi_1 \vee \varphi_2$ 的遗忘可以由分开计算遗忘后在吸取而得到；而正如(v) 中指出的那样，合取公式$\psi_1 \wedge \psi_2$ 的遗忘不能分别计算再合取，这甚至对命题公式都是不成立。例：令$\varphi = p \wedge (q \vee \neg p)$，从$\varphi$中遗忘掉$p$的结果为$q$，但是$Forget(p,\{p\}) \wedge Forget(q \vee \neg p,\{p\}) \equiv \top$。显然二者不等价。

下面是关于μ-演算的遗忘算子的其他性质。

命题 6.3 (Modularity). *给定μ-句子φ、原子命题的集合V 和原子命题p且$p \notin V$，则：*

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

证明. 令$\mathscr{M}_1 = (S_1,s_1,R_1,L_1)$为$\mathrm{F}_\mu(\varphi,\{p\} \cup V)$的模型。由遗忘的定义可知，存在$\varphi$的一个模型$\mathscr{M}$ ($\mathscr{M} = (S,s,R,L)$) 使得$\mathscr{M}_1 \leftrightarrow_{\{p\} \cup V} \mathscr{M}$。如下构造Kripke结构$\mathscr{M}_2 = (S_2,s_2,R_2,L_2)$：

(1) 对与s_2，令s_2为满足下列条件的状态：

- $p \in L_2(s_2)$当且仅当$p \in L_1(s_1)$，
- 对任意的$q \in V$，$q \in L_2(s_2)$ 当且仅当$q \in L(s)$，

– 对于其他的原子命题q'，$q' \in L_2(s_2)$ 当且仅当$q' \in L_1(s_1)$ 当且仅当$q' \in L(s)$。

(2) 其他情形：假定$\mathscr{M}_1$和$\mathscr{M}$有$\{p\} \cup V$-互模拟关系$\mathscr{B}$。

(i) 对任意的$w \in S$ 和$w_1 \in S_1$ 且$(w, w_1) \in \mathscr{B}$，令$w_2 \in S_2$和

* $p \in L_2(w_2)$ 当且仅当$p \in L_1(w_1)$，
* 对任意的$q \in V$，$q \in L_2(w_2)$ 当且仅当$q \in L(w)$，
* 对其他原子命题q'，$q' \in L_2(w_2)$ 当且仅当$q' \in L_1(w_1)$ 当且仅当$q' \in L(w)$。

(ii) 若$(w_1', w_1) \in R_1$，且w_2 是由w_1构造，$w_2' \in S_2$ 由w_1'构造，则$(w_2', w_2) \in R_2$。

- 删除S_2和R_2中重复的元素。

可以容易检查$\mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}_2$ 和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$。因此，$(\mathscr{M}_2, s_2) \models \mathrm{F}_\mu(\varphi, p)$，所以$(\mathscr{M}_1, s_1) \models \mathrm{F}_\mu(\mathrm{F}_\mu(\varphi, p), V)$。

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

$\Rightarrow \exists \mathscr{M}_2$ 使得$\mathscr{M}_2 \models \mathrm{F}_\mu(\varphi, p)$ 和$\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$ (定义 6.3)

$\Rightarrow \exists \mathscr{M}$ 使得$\mathscr{M} \models \varphi$ 和$\mathscr{M} \leftrightarrow_{\{p\}} \mathscr{M}_2$ (定义 6.3)。

因此，由命题 6.1可知$\mathscr{M} \leftrightarrow_{\{p\} \cup V} \mathscr{M}_1$，因此$\mathscr{M}_1 \models \mathrm{F}_\mu(\varphi, \{p\} \cup V)$。 □

下面这一性质为上述命题的推论。

推论 6.1 (交换性)**.** *给定μ-句子φ和原子命题的集合$V_i \subseteq \mathscr{A}$ $(i = 1, 2)$，有：*

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

F_μ 的另一个属性是关于AX 和EX模态词的：形如AXφ 或EXφ的μ-句子的遗忘可以提到AX 和EX后面计算。而对于$\mu X.\varphi$和$\nu X.\varphi$就没有这样的性质，因为φ显然不是一个μ-句子。

命题 6.4 (同质性)**.** *给定原子命题集合$V \subseteq \mathscr{A}$ 和μ-句子ϕ，则:*

(i) $\mathrm{F}_\mu(\mathrm{AX}\phi, V) \equiv \mathrm{AX}\mathrm{F}_\mu(\phi, V)$.

(ii) $\mathrm{F}_\mu(\mathrm{EX}\phi, V) \equiv \mathrm{EX}\mathrm{F}_\mu(\phi, V)$.

证明. 令$\mathscr{M} = (S, s, R, L)$、$M_i = (S_i, s_i, R_i, L_i)$ （$i \in \mathrm{N}$）且$\mathscr{M}' = (S', s', R', L')$，若下面条件满足，则称$\mathscr{M}' = (S', s', R', L')$为$\mathscr{M}$的一个子结构：

- $S' \subseteq S$ 和$S' = \{s' \mid s' \text{ is reachable from } s'\} \cup \{s'' \mid s'' \text{ can not be reached from both } s \text{ and } s'\}$,

- $R' = \{(s_1, s_2) \mid s_1, s_2 \in S' \text{ and } (s_1, s_2) \in R\}$,
- $L' : S' \to 2^{\mathscr{A}}$ and $\forall s_5 1 \in S'$ there is $L'(s_1) = L(s_1)$, and
- s' is s or a state reachable from s.

We denote $\exists_{sub}$ as "there exists a sub-structure" and $\forall_{sub}$ as "for each sub-structure".

(i) To prove $\mathrm{F}_\mu(\mathrm{AX}\phi, V) \equiv \mathrm{AX}(\mathrm{F}_\mu(\phi, V))$, we only need to prove $Mod(\mathrm{F}_\mu(\mathrm{AX}\phi, V)) = Mod(\mathrm{AXF}_\mu(\phi, V))$:

$(\Rightarrow)$ $\forall \mathscr{M}' \in Mod(\mathrm{F}_\mu(\mathrm{AX}\phi, V))$, $\exists \mathscr{M}$ s.t. $\mathscr{M} \models \mathrm{AX}\phi$ and $\mathscr{M} \leftrightarrow_V \mathscr{M}'$ by Def. 3.2
$\Rightarrow \forall_{sub} \mathscr{M}_1$ of $\mathscr{M}$, there is $\mathscr{M}_1 \models \phi$, where s_1 is a directed successor of s
$\Rightarrow \exists \mathscr{M}_2$ s.t. $\mathscr{M}_2 \models \mathrm{F}_\mu(\phi, V)$ and $\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$ by Def. 3.2

It is easy to construct a Kripke structure $\mathscr{M}_3$ by $\mathscr{M}_2$ s.t. $\mathscr{M}_2$ is a sub-structure of $\mathscr{M}_3$, in which s_2 is a direct successor of s_3 and $\mathscr{M}_3 \leftrightarrow_V \mathscr{M}$.
$\Rightarrow \mathscr{M}_3 \models \mathrm{AX}(\mathrm{F}_\mu(\phi, V))$ and $\mathscr{M}_3 \leftrightarrow_V \mathscr{M}'$ by Pro. 6.1
$\Rightarrow \mathscr{M}' \models \mathrm{AX}(\mathrm{F}_\mu(\phi, V))$ by Def. 3.2.

$(\Leftarrow)$ $\forall\, \mathscr{M}_3 \in Mod(\mathrm{AX}(\mathrm{F}_\mu(\phi, V)))$, and $\forall_{sub} \mathscr{M}_2$ of $\mathscr{M}_3$, in which s_2 is a directed successor of s_3, there is $\mathscr{M}_2 \models \mathrm{F}_\mu(\phi, V)$
$\Rightarrow \forall \mathscr{M}_2$, $\exists \mathscr{M}_1$ s.t. $\mathscr{M}_1 \models \phi$ and $\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$ by Def. 3.2

It is easy to construct a Kripke structure $\mathscr{M}$ by $\mathscr{M}_1$ s.t. $\mathscr{M}_1$ is a sub-structure of $\mathscr{M}$, in which s_1 is a direct successor of s, and $\mathscr{M} \leftrightarrow_V \mathscr{M}_3$
$\Rightarrow \mathscr{M} \models \mathrm{AX}\phi$, and then $\mathscr{M}_3 \models \mathrm{F}_\mu(\mathrm{AX}\phi, V)$ by Def. 3.2.

(ii) In order to prove $\mathrm{F}_\mu(\mathrm{EX}\phi, V) \equiv \mathrm{EXF}_\mu(\phi, V)$, we only need to prove $Mod\ (\mathrm{F}_\mu(\mathrm{EX}\phi, V)) = Mod(\mathrm{EXF}_\mu(\phi, V))$.

$(\Rightarrow)$ $\forall \mathscr{M}' \in Mod(\mathrm{F}_\mu(\mathrm{EX}\phi, V))$, $\exists \mathscr{M}$ s.t. $\mathscr{M} \models \mathrm{EX}\phi$ and $\mathscr{M} \leftrightarrow_V \mathscr{M}'$ by Def. 3.2
$\Rightarrow \exists_{sub} \mathscr{M}_1$ of $\mathscr{M}$ s.t. $\mathscr{M}_1 \models \phi$, where s_1 is a directed successor of s
$\Rightarrow \exists \mathscr{M}_2$ s.t. $\mathscr{M}_2 \models \mathrm{F}_\mu(\phi, V)$ and $\mathscr{M}_2 \leftrightarrow_V \mathscr{M}_1$ by Def. 3.2

It is easy to construct a Kripke structure $\mathscr{M}_3$ by $\mathscr{M}_2$ s.t. $\mathscr{M}_2$ is a sub-structure of $\mathscr{M}_3$, in which s_2 is a direct successor of s_3, and $\mathscr{M}_3 \leftrightarrow_V \mathscr{M}$
$\Rightarrow \mathscr{M}_3 \models \mathrm{EX}(\mathrm{F}_\mu(\phi, V))$ and $\mathscr{M}_3 \leftrightarrow_V \mathscr{M}'$ by Pro. 6.1
$\Rightarrow \mathscr{M}' \models \mathrm{EX}(\mathrm{F}_\mu(\phi, V))$ by Def. 3.2.

$(\Leftarrow)$ $\forall\, \mathscr{M}_3 \in Mod(\mathrm{EX}(\mathrm{F}_\mu(\phi, V)))$, $\exists_{sub} \mathscr{M}_2$ of $\mathscr{M}_3$ s.t. $\mathscr{M}_2 \models \mathrm{F}_\mu(\phi, V)$
$\Rightarrow \exists \mathscr{M}_1$ s.t. $\mathscr{M}_1 \models \phi$ and $\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$ by Def. 3.2

It is easy to construct a Kripke structure $\mathscr{M}$ by $\mathscr{M}_1$ s.t. $\mathscr{M}_1$ is a sub-structure of $\mathscr{M}$, in which s_1 is a direct successor of s, and $\mathscr{M} \leftrightarrow_V \mathscr{M}_3$
$\Rightarrow \mathscr{M} \models \mathrm{EX}\phi$, and then $\mathscr{M}_3 \models \mathrm{F}_\mu(\mathrm{EX}\phi, V)$ by Def. 3.2.

□

AX （或EX）在F_μ上的同质性表明，在从AXφ （或EXφ）遗忘掉V中的原子命题等价于将F_μ提到AX 和EX后面计算的结果。特别地，当命题 6.4中的公式ϕ为命题公式时，从$Q\mathrm{X}\phi$ ($Q \in \{\mathrm{E}, \mathrm{A}\}$) 中遗忘原子命题可以使用命题逻辑的遗忘计算方法来计算。

6.4　计算复杂性

吸取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[66]。正如之前说过的$\mathrm{F}_\mu(\varphi, V)$与均匀插值$\widetilde{\exists} V\varphi$等价[66]。因此，下面的命题容易证明。

命题 6.5. *给定μ-句子φ 和原子命题$p \in \mathscr{A}$。若φ 是一个μ-句子，$\mathrm{F}_\mu(\varphi, \{p\})$能在线性时间内计算。*

这种情况下，可以首先将一个μ-句子转化为吸取μ-公式，再去计算遗忘。下面的例子给出如何计算从吸取μ-公式中遗忘“ch”。

例 6.2. 令$\varphi_1 = j \wedge ch \wedge Cover(\neg j \wedge \neg ch, \top)$、$\varphi_2 = \mu X.(j \wedge ch) \wedge Cover(X, \top)$、$\varphi_3 = \nu X.(j \wedge ch) \wedge Cover(Cover(X, \top), \top)$。令$V = \{ch\}$，我们能容易地计算从这些公式里遗忘掉$V$。

(1) $\mathrm{F}_\mu(\varphi_1, V) \equiv j \wedge Cover(\neg j, \top) \equiv j \wedge \mathrm{EX}(\neg j)$；

(2) $\mathrm{F}_\mu(\varphi_2, V) \equiv \mu X.j \wedge Cover(X, \top) \equiv \mu X.j \wedge \mathrm{EX}X$；

(3) $\mathrm{F}_\mu(\varphi_2, V) \equiv \nu X.j \wedge Cover(Cover(X, \top), \top) \equiv \nu X.j \wedge \mathrm{EX}(\mathrm{EX}X)$。

尽管如此，关于遗忘的模型检测（即：检查一个结构是否为从μ-句子中遗忘掉某个原子命题的集合的模型）也是困难的。

命题 6.6 (Model Checking). *给定一个有限的Kripke structure $\mathscr{M}$、一个μ-句子φ和原子命题的集合$V \subseteq \mathscr{A}$。有：*

*(i) 判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$ 在*EXPTIME*中；*

*(ii) 若φ 是一个吸取μ-公式，则判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$在*NP$\cap$*co*-NP*中。*

证明. 对于一个μ-公式φ，如果该公式为一个吸取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[64]）A_φ，否则需要指数时间构造其对应的μ-自动机EXPTIME[2]. 这里证明(ii)，(i)可以类似地证明。

[2]Personal communication: Giovanna D'Agostino, 2020.

令A_φ为一个μ-自动机且满足对任意的Kripke $\mathscr{N}$，A_φ 接受$\mathscr{N}$ 当且仅当$\mathscr{N} \models \varphi$，其中$A_\varphi = (Q, \Sigma_p, \Sigma_r, q_0, \delta, \Omega)$、$\Sigma_p = Var(\varphi)$。不是一般性地，假定$V \subseteq Var(\varphi)$ 和$V = \{p\}$。因此，构造一个μ-自动机$\mathscr{B} = (Q, \Sigma_p - V, \Sigma_r, q_0, \delta', \Omega)$ 使得对任意的$q \in Q$ 和$L \subseteq \Sigma_p - V$，

$$\delta'(q, L) = \delta(q, L) \cup \delta(q, L \cup \{p\}).$$

已有结论表明，对任意的Kripke结构$\mathscr{N}$，$\mathscr{B}$ 接受$\mathscr{N}$当且仅当存在φ的一个模型$\mathscr{N}'$使得$\mathscr{N} \leftrightarrow_{\{p\}} \mathscr{N}'$[90]，即$\mathscr{B}$对应于和$\mathrm{F}_\mu(\varphi, V)$等价的$\mu$-句子。

在这种情况下，判定$\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$问题被归约到判定是否$\mathscr{B}$ 接受$\mathscr{M}$的问题。而$\mathscr{B}$ 从根r接受一个Kripke结构$\mathscr{M} = (S, r, R, L)$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[64]中。 □

给定μ-句子φ和ψ，V 为原子命题的集合。从知识是进化的角度来看，以下推理问题（在命题逻辑里也有研究[?]）是值得探索的：

(i) [Var-weak] φ在ψ的原子命题上的约束至多有ψ强，即$\psi \models \mathrm{F}_\mu(\varphi, V)$；

(ii) [Var-strong] φ在ψ的原子命题上的约束至少有ψ强，即$\mathrm{F}_\mu(\varphi, V) \models \psi$；

(iii) [Var-entailment] φ在$Var(\varphi) \cap Var(\psi)$上的约束比$\psi$在$Var(\varphi) \cap Var(\psi)$上的约束强，即$\mathrm{F}_\mu(\varphi, V) \models \mathrm{F}_\mu(\psi, V)$,

值得注意的是，在(i) 和(ii)中，$Var(\varphi) - V = Var(\psi)$，在(iii)中，$V \subseteq (Var(\varphi) \cap Var(\psi))$。

定理 6.5 (Entailment). *给定μ-句子φ 和ψ，V为原子命题的集合，则下面的判定问题为*EXPTIME*-完全的。*

(i) 判定$\mathrm{F}_\mu(\varphi, V) \models^? \psi$,

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

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

证明. 这里给出(i)的证明，其他的结论能够类似地证明。

隶属性: 令A_φ 和A_ψ 分别为φ 和ψ的μ-自动机，由命题**??**的证明可从A_φ构造$\mathrm{F}_\mu(\varphi, V)$的$\mu$-自动机we can construct the μ-automaton $\mathscr{B}$ of $\mathrm{F}_\mu(\varphi, V)$ from A_φ。由命题7.3.2[?]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[?])。

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

Hardness: 对任意的μ-句子存在一个等价的μ-自动机，且对任意的μ-自动机存在一个等价的μ-句子[64]。因此，判定问题$\mathrm{F}_\mu(\varphi,V) \models^? \psi$规约成其对应的$\mu$-自动机是否为空的问题。因此，hareness直接来源于定义6.3[?]。 □

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[?]，及“Var-match”问题:

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

(ii) [Var-match] φ在ψ的原子命题上的约束与ψ等价，即$\mathrm{F}_\mu(\varphi,V) \equiv \psi$，

(iii) [Var-equivalence] φ和ψ在原子命题上V的约束是否等价，即$\mathrm{F}_\mu(\varphi,V) \equiv \mathrm{F}_\mu(\psi,V)$。

对于φ 和ψ，其原子命题上的约束是一样的。

推论 6.2. *给定μ-句子φ 和ψ，V为原子命题的集合。则下面的判定问题为*EXPTIME*-完全的。*

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

(ii) *判定$\mathrm{F}_\mu(\varphi,V) \equiv^? \varphi$，*

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

6.5 本章小结

本章针对差分隐私数据收集应用中存在的策略型攻击问题，利用信息论、博弈均衡理论研究了隐私防护者与隐私攻击者的理性策略选择，提出了隐私保护的攻防博弈(PPAD)模型，以实现隐私与数据效用均衡。首先，基于信息论度量方法分析差分隐私保护系统中隐私保护者和攻击者的隐私目标，形式化表述为互信息隐私的极大极小问题。其次，针对上述提出的问题，考虑策略型的隐私攻击者和防护者，提出隐私保护的攻防博弈模型，并具体为二人的零和博弈模型。随后，给出博弈的凹凸性以及均衡分析。进一步，为了求解博弈模型鞍点，设计了策略优化选择算法。最后，通过实验阐述了所提出的方案可以用于比较等价的隐私机制，并阐述了隐私量化是最坏情况下的隐私泄露，也即是，隐私防护者的最大隐私泄露。

第七章　遗忘理论的应用

本章针对差分隐私存在策略型攻击问题，基于差分隐私通信模型，提出隐私保护的攻防博弈模型，以实现隐私保护的隐私与数据效用均衡。首先，定义差分隐私保护系统中隐私保护者与攻击者(敌手)的隐私目标，并将其表述为隐私泄露的极大极小问题。针对该问题，以隐私度量为效用函数，构建两方零和对策博弈模型，并基于极大极小定理、凹凸博弈给出相应的博弈均衡分析。理论分析表明鞍点的存在，并进一步给出鞍点的内涵。其次，对于等价的ε-隐私机制，提出等价类隐私机制可比较的方法，解决ε-隐私度量存在的不足。最后，基于交替最优响应设计鞍点计算的策略优化选择算法。理论分析及实验结果表明提出的方法可辅助隐私保护者评估隐私泄露风险。

7.1　引言

近年来，私有敏感信息泄露问题引起了社会和学术研究领域的广泛关注，正在成为大数据时代的一个主要挑战。如医疗数据、在线社交活动、基于位置的服务等网络应用中对个人数据的使用，使得个人的隐私遭受到了潜在的风险，由此产生了用户隐私泄露问题。隐私泄露逐渐成为数据收集、发布、分析、感知等隐私计算[?]任务中迫切需要解决的问题，技术层面上亟需有效的隐私保护模型与算法。围绕隐私保护的核心任务，学术研究已提出诸多的隐私保护模型及解决方案。其中，差分隐私[???]是广泛被接受的隐私保护模型。为了克服基本假设中存在可信实体的局限性，本地模型的差分隐私[??](Local Differential Privacy，LDP)被提出，并主要应用于解决数据收集阶段的隐私保护问题。在差分隐私的本地模型中，每一个用户独立的扰动自己的原始数据，然后报告扰动后的数据给数据聚合者(收集者)。由于本地模型的显著特性，一经提出就受到学术研究和产业应用的关注。学术界围绕本地模型的应用，先后提出诸如RAPPOR[??]、k-RR[?]、OUE[?]等众多著名先进的隐私机制。产业界如Google Chrome 浏览器[?]、Apple公司操作系统[?]等将其应用于隐私保护数据收集、分析场景。纵观研究工作，数据聚合者通常是半诚实的敌手模型，隐私性与数据质量依然是核心的关注问题，隐私保护难以实现完美无泄露，相对的寻找隐私保护策略均衡成为较为理想的权衡折中解决方案。

实际的应用中，随机化响应[?]技术是有效实现LDP的方法[????]，其已成为LDP方案设计的基本构建模块。本质上，随机化响应是从原始数据到扰动输出数据的一个概率性映射。基于此，隐私机制的随机性与隐私保护的隐私和数据质量密切相关，这就是权衡隐私与效用课题的研究内容。目前，这仍然是差分隐私保护中学术研究的重点。在差分隐私本地模型的数据收集应用中，数据聚合者收集、存储、分析用户报告的扰

动数据[?]，扰动后的数据与原始数据之间的关联决定了隐私保护的隐私性与数据的可用性。为了解决权衡的问题，在寻找有效的折中方案过程中，隐私与数据质量的度量是基本的前提工作。当前，隐私预算参数ε是一个量化差分隐私不可区分等级的事实标准。但是，这个度量是分布独立的，其存在着一些不足之处。例如，一个确定性的隐私协议$Q(x) = x \bmod 2$提供$\varepsilon = \infty$的隐私保障，但是该隐私协议仍然可以阻止部分的隐私泄露[?]。除了上面提到的，这样的隐私度量无法在等价的ε-隐私机制集合中区分那个隐私机制的性能更好，因为集合中的隐私机制都提供相同的ε-不可区分等级。受这些问题的激励，度量也亟需新的评价方法。

针对上述问题，从隐私信息流的角度，基于信息论的方法可以得到有效的解决[?]。首先，上述有关LDP机制的数据处理过程，可以被建模为一个原始数据与扰动数据之间的噪声信道模型[??](参见**??**节内容)。然后，利用熵与互信息量定义隐私泄露度量，且已在诸多研究工作中得到了应用[????]。重要的，信息论的模型中考虑了数据分布和隐私机制对隐私泄露的影响，互信息隐私测量扰动数据包含原始数据的信息量，它捕捉住了隐私攻击者有关数据分布的先验知识。此外，隐私保护系统中仅有两方的参与者[?]，用户本地执行隐私协议旨在减少隐私泄露，其类似于隐私防护者。相似的，聚合者试图最大化隐私泄露，以至于推断用户的个人信息，类似于隐私攻击者。鉴于上述分析，本章中关注的问题演变为了有关隐私的攻防对抗问题。自然的，以博弈均衡的思想解决这个问题不失为一个理想的选择。现有存在的工作中，二人零和对策博弈[????]、斯坦伯格博弈[?]、贝叶斯博弈[?]等在差分隐私框架下都有一定的应用。重要的，从量化信息流的角度构建的信息泄露博弈[??]、量化信息流博弈[?]是有效的隐私分析方法。

鉴于上述的分析，本章中考虑在理性的框架下使用信息论的方法解决隐私与效用的均衡问题，通过分析隐私保护者与攻击者的隐私目标，首先将其形式化表述为隐私的极大极小问题。然后，基于差分隐私通信模型(**??**节)，提出隐私保护的攻防博弈模型，也即是一个二人的零和博弈模型。进一步，提出一个交替最优化算法计算提出的攻防博弈的鞍点，利用鞍点策略实现差分隐私的均衡优化。理论上的均衡分析和实验结果表明，提出的均衡思想是一种稳定的状态，可用于预测评估隐私泄露风险。本章的主要贡献可以总结如下：

(1) 通过使用信息论的方法量化隐私攻击者的信息增益，提出了隐私保护的攻防博弈模型(PPAD)，用于分析用户和聚合者的理性策略行为。

(2) 在隐私与效用的原则下，分析隐私防护者与攻击者的隐私目标，形式化表述互信息隐私的极大极小问题，构建二人零和对策博弈求解形式化表述的极大极小问题，并利用交替最优响应策略，设计交替最优的策略优化选择算法。

(3) 对于等价的ε-隐私机制，提出一种有效的比较分析方法，并进一步验证了互信息隐私泄露在最差情况下可以达到隐私泄露的上界，为隐私泄露风险评估提供了量化

分析的方法。

本章其余部分组织如下：首先，第6.2节阐述本章的系统模型、敌手模型，提出研究问题。其次，第**??**节提出隐私保护的攻防博弈模型(PPAD)，并给出均衡分析。进一步，第**??**节介绍均衡求解的策略优化选择算法。最后，第**??**节给出实验与分析，并在第6.5节总结本章的研究工作。

7.2 最强必要条件和最弱充分条件

这部分介绍如何使用遗忘理论计算最强必要条件和最弱充分条件。直观地说，最强必要条件指最一般的结果（the most general consequence），最弱充分条件指最特殊的诱因（the most specific abduction）。下面给出其形式化定义，本章所说的公式指的是μ-句子或CTL公式。

定义 7.1 (充分和必要条件)**.** 给定两个公式ϕ和ψ，$V \subseteq Var(\phi)$，$q \in Var(\phi) - V$ 和$Var(\psi) \subseteq V$。

- 若$\phi \models q \to \psi$，则称ψ是q在V和ϕ上的必要条件（necessary condition，NC）；
- 若$\phi \models \psi \to q$，则称ψ是q在V和ϕ上的充分条件（sufficient condition，SC）；
- 若ψ是q在V和ϕ上的必要条件，且对于任意的q在V和ϕ上的必要条件ψ'都有$\phi \models \psi \to \psi'$，则称$\psi$是$q$在$V$和$\phi$上的最强必要条件（strongest necessary condition，SNC）；
- 若ψ是q在V和ϕ上的充分条件，且对于任意的q在V和ϕ上的充分条件ψ'都有$\phi \models \psi' \to \psi$，则称$\psi$是$q$在$V$和$\phi$上的最弱充分条件（weakest sufficient condition，WSC）；

从上述定义可以看出，SNC（WSC）是q在V和ϕ上的NC（SC）中最强（最弱）的一个，即：对任意的（或SC）ψ'，$\phi \models \text{SNC} \to \psi'$（$\phi \models \psi' \to \text{WSC}$）。此外，如果公式$\psi$和$\psi'$都是$q$在$V$和$\phi$上的SNC（WSC），则$\psi \equiv \psi'$。下面的命题表明SNC和WSC是一对对偶概念。

命题 7.1 (对偶性)**.** 令V、q、φ和ψ为定义 *7.1*出现的符号。则ψ是q在V和ϕ上的*SNC*（*WSC*）当且仅当$\neg\psi$是$\neg q$在V和ϕ上的*WSC*（*SNC*）。

证明. （i）假设ψ是q在V和ϕ上的SNC。则$\varphi \models q \to \psi$，因而$\varphi \models \neg\psi \to \neg q$，即$\neg\psi$是$\neg q$ 在V 和ϕ上的SC. 设ψ' 是$\neg q$ 在V 和ϕ 上的SC：$\varphi \models \psi' \to \neg q$。则$\varphi \models q \to \neg\psi'$，即$\neg\psi'$是$q$ 在V 和ϕ 上NC。因此，由假设可知$\varphi \models \psi \to \neg\psi'$，所以$\varphi \models \psi' \to \neg\psi$。这证明了$\neg\psi$是$\neg q$ 在V 和ϕ上的WSC。可以类似地证明另一部分。

（ii）WSC的情形可以类似SNC的情形给出证明。 □

在定义 7.1中将q替换为任意的公式α，$V \subseteq Var(\alpha) \cup Var(\phi)$，则定义 7.1被推广到任意公式的最强必要条件和最弱充分条件的定义。下面的命题表示了原子命题的充分（必要）条件与公式的充分（必要）条件之间的关系：通过计算原子命题的充分（必要）条件来计算公式的充分（必要）条件。

命题 7.2. *给定公式Γ和α，$V \subseteq Var(\alpha) \cup Var(\Gamma)$，$q$是不出现在$\Gamma$和$\alpha$中的原子命题。集合$V$上的公式$\varphi$是$\alpha$在$V$和$\Gamma$上的SNC（WSC）当且仅当$\varphi$是$q$在$V$和$\Gamma'$上的SNC（WSC），其中$\Gamma' = \Gamma \cup \{q \leftrightarrow \alpha\}$。*

证明. 这里给出SNC部分的证明，WSC部分的证明与其类似。

对于任意的公式β，记$SNC(\varphi,\beta,V,\Gamma)$为“$\varphi$是$\beta$在$V$和$\Gamma$上的SNC”，$NC(\varphi,\beta,V,\Gamma)$为“$\varphi$是$\beta$在$V$和$\Gamma$上的NC”。

($\Rightarrow$) 证明“若$SNC(\varphi,\alpha,V,\Gamma)$，则$SNC(\varphi,q,V,\Gamma')$”。由$SNC(\varphi,\alpha,V,\Gamma)$和$\alpha \equiv q$可知$\Gamma' \models q \rightarrow \varphi$，即：$\varphi$是$q$在$V$和$\Gamma'$上的NC。假设$\varphi'$是$q$在$V$和$\Gamma'$上的任意NC，由于$\alpha \equiv q$和$IR(\alpha \rightarrow \varphi', \{q\})$，因此，$\mathrm{F}_{\mathrm{CTL}}(\Gamma', q) \models \alpha \rightarrow \varphi'$。由引理6.1可知$\Gamma \models \alpha \rightarrow \varphi'$，即：$NC(\varphi',\alpha,V,\Gamma)$。

($\Leftarrow$) 证明“若$SNC(\varphi,q,V,\Gamma')$，则$SNC(\varphi,\alpha,V,\Gamma)$”。由$SNC(\varphi,q,V,\Gamma')$、$IR(\alpha \rightarrow \varphi, \{q\})$和(**PP**) 可知$\mathrm{F}_{\mathrm{CTL}}(\Gamma', \{q\}) \models \alpha \rightarrow \varphi$，又由引理6.1可知$\Gamma \models \alpha \rightarrow \varphi$，即：$NC(\varphi,\alpha,V,\Gamma)$。设$\varphi'$是$\alpha$在$V$和$\Gamma$上的任意NC。由$\alpha \equiv q$和$\Gamma' = \Gamma \cup \{q \equiv \alpha\}$可知$\Gamma' \models q \rightarrow \varphi'$，即：$NC(\varphi',q,V,\Gamma')$。又因为$SNC(\varphi,q,V,\Gamma')$、$IR(\varphi \rightarrow \varphi', \{q\})$和(**PP**)，所以$\mathrm{F}_{\mathrm{CTL}}(\Gamma', \{q\}) \models \varphi \rightarrow \varphi'$。由引理6.1可知$\Gamma \models \varphi \rightarrow \varphi'$，因此$SNC(\varphi,\alpha,V,\Gamma)$成立。 □

为了对给定原子命题集合下的公式的最弱充分条件有个直观的认识，下面给出一个简单的例子。

例 7.1 (Continued from Example **??**). 本例来源于图 **??**中的初始结构$\mathscr{K}_2$。令$\psi = \mathrm{EX}(s \wedge (\mathrm{EX}se \vee \mathrm{EX}\neg d))$、$\varphi = \mathrm{EX}(s \wedge \mathrm{EX}\neg d)$、$\mathscr{A} = \{d,s,se\}$和$V = \{s,d\}$。下面证明$\varphi$是$\psi$在$V$和$\mathscr{K}_2$上的WSC：

(i) 由已知有$\varphi \models \psi$和$Var(\varphi) \subseteq V$。此外，$(\mathscr{M}, s_0) \models \varphi \wedge \psi$，因此$\mathscr{K}_2 \models \varphi \rightarrow \psi$，即：$\varphi$是$\psi$在$V$和$\mathscr{K}_2$上的SC;

(ii) 这里证明“对任意的ψ在V和$\mathscr{K}_2$上的SC φ'都有$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$”。易知若$\mathscr{K}_2 \not\models \varphi'$，则$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$。假设$\mathscr{K}_2 \models \varphi'$。由$\varphi'$是$\psi$在$V$和$\mathscr{K}_2$上的SC可知$\varphi' \models \mathrm{EX}(s \wedge \phi)$，其中$\phi$是使得$\phi \models \mathrm{EX}se \vee \mathrm{EX}\neg d$成立的公式。又$\mathrm{IR}(\varphi', \overline{V})$，所以$\phi \models \mathrm{EX}\neg d$。因此，$\varphi' \models \varphi$且$\mathscr{K}_2 \models \varphi' \rightarrow \varphi$。

如何使用遗忘理论计算SNC（WSC）是本章讨论的关键问题。下面首先给出其理论基础，然后再做直观的解释。

定理 7.1. 给定公式φ、原子命题的集合$V \subseteq Var(\varphi)$和原子命题$q \in Var(\varphi) - V$。

(i) $\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$ 是q在V和φ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：由遗忘理论的定义可知$\varphi \wedge q \models \mathscr{F}$。因此，$\varphi \models q \rightarrow \mathscr{F}$，即：$\mathscr{F}$ 是q在V 和φ上的NC。

“SNC”部分：假设ψ'为q在V和φ上的任意NC，即：$\varphi \models q \rightarrow \psi'$。由定理 3.1和$IR(\psi', (Var(\varphi) \cup \{q\}) - V)$可知，若$\varphi \wedge q \models \psi'$，则$\mathscr{F} \models \psi'$。由假设可知$\varphi \models q \rightarrow \psi'$，所以$\varphi \wedge \mathscr{F} \models \psi'$，因此$\varphi \models \mathscr{F} \rightarrow \psi'$。

由上面两部分可知，$\mathscr{F}$是q在V和φ上的SNC。

(ii) 令$\mathscr{F} = \neg\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \neg q, (Var(\varphi) \cup \{q\}) - V)$。由命题 7.1可知，对任意的命题$q'$，$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q', (Var(\varphi) \cup \{q'\}) - V)$是$q'$在$V$和$\varphi$上的SNC，当且仅当$\neg\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q', (Var(\varphi) \cup \{q'\}) - V)$是$\neg q'$在$V$和$\varphi$上的WSC。由(i)可知$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q', (Var(\varphi) \cup \{q'\}) - V)$是$q'$在$V$和$\varphi$上的SNC，所以$\neg\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q', (Var(\varphi) \cup \{q'\}) - V)$是$\neg q'$在$V$和$\varphi$上的WSC。令$q = \neg q'$，可得$\mathscr{F}$ 是q 在V 和φ上的WSC。 □

令$\mathscr{F} = \mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$。上面的定理可以直观地解释如下：由遗忘理论的定义可知$\varphi \wedge q \models \beta$，这说明$\mathscr{F}$ 是q 在V 和φ上的NC；对任意的与$(Var(\varphi) \cup \{q\}) - V$无关的公式$\psi$，若$\varphi \wedge q \models \psi$，则由定理 **??**可知$\beta \models \psi$。

由第五章可知，任意的有限的初始K-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K-结构下的SNC和WSC。

推论 7.1. 令$\mathscr{K} = (\mathscr{M}, s)$为初始K-结构，其中$\mathscr{M} = (S, R, L, s_0)$ 为有限原子命题集合$\mathscr{A}$上的初始-*Kripke*结构，$V \subseteq \mathscr{A}$且$q \in V' = \mathscr{A} - V$。则：

(i) $\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}) \wedge q, V')$是$q'$在$V$和$\mathscr{K}$上的*SNC*;

(ii) $\neg\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{K}) \wedge \neg q, V')$是$q'$在$V$和$\mathscr{K}$上的*WSC*。

7.3 μ-演算下的知识更新

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

- (U1) $\Gamma \diamond \phi \models \phi$;

- (U2) 若$\Gamma \models \phi$，则$\Gamma \diamond \phi \equiv \Gamma$；
- (U3) 若Γ和ϕ都是可满足的，则$\Gamma \diamond \phi$是可满足的；
- (U4) 若$\Gamma_1 \equiv \Gamma_2$且$\phi_1 \equiv \phi_2$，则$\Gamma_1 \diamond \phi_1 \equiv \Gamma_2 \diamond \phi_2$；
- (U5) $(\Gamma \diamond \phi) \wedge \psi \models \Gamma \diamond (\phi \wedge \psi)$；
- (U6) 若$\Gamma \diamond \phi \models \psi$且$\Gamma \diamond \psi \models \phi$，则$\Gamma \diamond \phi \equiv \Gamma \diamond \psi$；
- (U7) 若Γ有唯一一个模型，则$(\Gamma \diamond \phi) \wedge (\Gamma \diamond \psi) \models \Gamma \diamond (\phi \vee \psi)$；
- (U8) $(\Gamma_1 \vee \Gamma_2) \diamond \phi / (\Gamma_1 \diamond \phi) \vee (\Gamma_2 \diamond \phi)$。

其中，$\diamond$为知识更新操作，$\varphi \diamond \psi$表示用ψ更新φ得到的结果。

本小节假设所有的初始K-结构都是有限的，即：状态来源于有限的状态空间且$\mathscr{A}$为有限的原子命题的集合。下面定理显然成立：

定理 7.2. *给定μ-句子ϕ和原子命题的集合$V \subseteq \mathscr{A}$。存在一个μ-句子ψ使得：*

$$\mathscr{M} \models \psi \text{ 当且仅当存在} \mathscr{M}' \in Mod(\phi) \text{ 使得} \mathscr{M} \leftrightarrow_V \mathscr{M}'$$

其中$\mathscr{M}$和$\mathscr{M}'$都是有限的初始结构。

证明. 令$\psi = \mathrm{F}_\mu(\phi, V)$。由定理 6.2和遗忘理论的定义可知，对任意的$\mathscr{M} \models \psi$存在一个$\mathscr{M}' \models \phi$使得$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，且对每一个$\mathscr{M}' \in Mod(\phi)$都有$\mathscr{M}' \models \psi$。此时，容易证明对任意的有限初始结构$\mathscr{M}$，若$\mathscr{M} \models \psi$，则存在一个$\mathscr{M}'$使得$\mathscr{M}' \models \phi$且$\mathscr{M} \leftrightarrow_V \mathscr{M}'$。

此外，对任意的$\mathscr{M}' \in Mod(\phi)$，由(**W**)可知存在$\mathscr{M}' \models \psi$。又$\mathscr{M} \leftrightarrow_V \mathscr{M}'$，所以由定理 **??**可知$\mathscr{M} \models \psi$。 □

定理 7.2表明模型结构被限制到有限初始结构下的μ-演算下遗忘理论也是封闭的。此外，由?? 可知，任意$\mathscr{A}$上的有限初始K-结构$\mathscr{K}$都能用一个CTL公式——特征公式$\mathscr{F}_{\mathscr{A}}(\mathscr{K})$来表示，此公式也是$\mu$-句子。

对于给给定的$\mathscr{A}$和$V_{min} \subseteq \mathscr{A}$，记$\varphi = \mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min})$，其中$V_{min} \subseteq \mathscr{A}$是使得$\varphi$可满足的极小子集。此外，公式

$$\bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min}))$$

表示所有$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{K}), V_{min})$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$V_{min}$：

定义 7.2. 给定μ-句子Γ和ϕ。知识更新操作$\diamond_\mu$定义如下：

$$Mod(\Gamma \diamond_\mu \phi) = \bigcup_{\mathscr{K} \in Mod(\Gamma)} \bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_\mu(\mathscr{F}_\mathscr{A}(\mathscr{K}), V_{min}) \wedge \phi),$$

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

从直观上来说，$\Gamma \diamond_\mu \phi$表示通过极小化改变Γ的模型到ϕ的模型来更新Γ。换句话说，定义 7.2通过极小化改变Γ的每个模型，使得该模型能够满足ϕ来更新原有的知识Γ。从这个角度看，这样定义的知识更新是一种基于模型的知识更新方法。

此外，μ-演算下的知识更新也可以通过像命题逻辑里的那样来定义：令I，J_1和J_2为三个赋值，即：原子命题的集合；则J_1比J_2更接近I（记为：$J_1 \leq_{I,pam} J_2$）当且仅当$\mathrm{Diff}(I,J_1) \subseteq \mathrm{Diff}(I,J_2)$，其中$\mathrm{Diff}(X,Y) = \{p \in \mathscr{A} \mid X(p) \neq Y(p)\}$。那么命题逻辑里的知识更新——用$\psi$更新$\Gamma$，即为$\psi$的关于偏序关系$\leq_{I,pam}$的所有极小模型的集合（$I$是$\Gamma$的模型），即：

$$Mod(\Gamma \diamond_{pam} \psi) = \bigcup_{I \in Mod(\Gamma)} Min(Mod(\psi), \leq_{I,pam}).$$

其中，$Min(Mod(\psi), \leq_{I,pam})$是$\psi$的关于偏序关系$\leq_{I,pam}$的极小模型的集合。

类似地，这里定理有限初始结构之间关于另一个初始结构的偏序关系。

定义 7.3. 给定三个有限初始结构$\mathscr{M}$、$\mathscr{M}_1$和$\mathscr{M}_2$，$\mathscr{M}_1$比$\mathscr{M}_2$更接近$\mathscr{M}$（记为$\mathscr{M}_1 \leq_\mathscr{M} \mathscr{M}_2$）当且仅当对任意使得$\mathscr{M}_2 \leftrightarrow_{V_2} \mathscr{M}$成立的$V_2 \subseteq \mathscr{A}$都存在一个$V_1 \subseteq V_2$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$\mathscr{M}_2 \not\leq_\mathscr{M} \mathscr{M}_1$。

给定有限初始结构的集合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的集合。则$\leq_\mathscr{M}$与知识更新操作$\diamond_\mu$有如下关系。

定理 7.3. 给定μ-句子Γ和ϕ，则：

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

证明. 对每一个初始结构$\mathscr{M}' \in Mod(\Gamma \diamond_\mu \phi)$，这里证明存在一个$\mathscr{M} \in Mod(\Gamma)$使得$\mathscr{M}' \in Min(Mod(\phi), \leq_\mathscr{M})$。由定义 7.2可知，存在$\mathscr{M} \in Mod(\Gamma)$使得$\mathscr{M}' \in Mod(\mathrm{F}_\mu(\mathscr{F}_\mathscr{A}(\mathscr{M}), V_{min}) \wedge \phi)$。此外，存在一个特殊的$V' \subseteq \mathscr{A}$（即：$V' = V_{min}$）使得$\mathscr{M}' \leftrightarrow_{V'} \mathscr{M}$和$\mathscr{M}' \in Mod(\phi)$。因为$V'$是使得$\mathscr{M}' \leftrightarrow_{V'} \mathscr{M}$成立的极小子集，因此对任意使得$\mathscr{M}'' \leftrightarrow_{V_{min}} \mathscr{M}$成

立的ϕ的模型$\mathscr{M}''$，由遗忘理和特征公式论的定义可知$\mathscr{M}' \leq_{\mathscr{M}} \mathscr{M}''$。因此，$\mathscr{M}' \in Min(Mod(\phi), \leq_{\mathscr{M}})$。

对每一个初始结构$\mathscr{M}' \in \bigcup_{\mathscr{M} \in Mod(\Gamma)} Min(Mod(\phi), \leq_{\mathscr{M}})$，存在$\mathscr{M} \in Mod(\Gamma)$使得$\mathscr{M}' \in Min(Mod(\phi), \leq_{\mathscr{M}})$。设$V_{min}$是使得$\mathscr{M}' \leftrightarrow_{V_{min}} \mathscr{M}$成立的极小子集。根据$\leq_{\mathscr{M}}$的定义可知，不存在其他$\mathscr{M}'' \in Mod(\phi)$使得$\mathscr{M}'' \leftrightarrow_{V'} \mathscr{M}$且$V' \subset V_{min}$。因而$\mathscr{M}' \in Mod(\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \phi)$，所以$\mathscr{M}' \in Mod(\Gamma \diamond_\mu \phi)$。 □

从定理 7.3可以看出，通过遗忘理论定义的知识更新操作与通过有限初始结构间的偏序关系定义的知识更新一致，且通过遗忘理论定义的知识更新操作满足Katsuno和Mendelzon提出的八条基本条件。

定理 7.4. *知识更新操作$\diamond_\mu$满足Katsuno和Mendelzon提出的基本条件(U1)-(U8)。*

证明. (U1). 由定理 7.3可知$Mod(\Gamma \diamond_\mu \phi) \subseteq Mod(\phi)$，因此$\Gamma \diamond_\mu \phi \models \phi$。

(U2). 首先证明$\Gamma \diamond_\mu \phi \models \Gamma$。对$\Gamma \diamond_\mu \phi$的任意一个模型$\mathscr{M}$，存在一个$\mathscr{M}_1 \in Mod(\Gamma)$和$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。又$\Gamma \models \phi$，因此$V_{min} = \emptyset$，即$\mathscr{M} \models \Gamma$。类似地，对$\Gamma$的每一个模型$\mathscr{M}$，存在一个$\mathscr{M}_1 \in Mod(\Gamma \diamond_\mu \phi)$和一个$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。又$\Gamma \models \phi$，因此$V_{min} = \emptyset$。所以，$\Gamma \models \Gamma \diamond_\mu \phi$。

容易证明$\diamond_\mu$满足(U3)和(U4)。

(U5). 对$(\Gamma \diamond_\mu \phi) \wedge \psi$的每一个模型$\mathscr{M}$，存在一个$\mathscr{M}_1 \in Mod(\Gamma)$和一个$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。此外，可知$\mathscr{M} \models \phi \wedge \psi$，因此，$\mathscr{M} \models \Gamma \diamond_\mu (\phi \wedge \psi)$。

(U6). 这里给出$\Gamma \diamond_\mu \phi \models \Gamma \diamond_\mu \psi$的证明，另一个方向可以类似地证明。对$\Gamma \diamond_\mu \phi$的每一个模型$\mathscr{M}$，$\mathscr{M}$也是$\psi$的模型，且存在$\mathscr{M}_1 \in Mod(\Gamma)$和$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。因此，$\mathscr{M}$是$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \psi$的模型，也即是$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \psi$可满足的。设$V \subset V_{min}$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V) \wedge \psi$可满足的原子命题的集合，由$\Gamma \diamond_\mu \psi \models \phi$可知$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V) \wedge \phi$是可满足的，这与$V_{min}$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \phi$可满足的极小子集。因此，$V_{min}$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}_1), V_{min}) \wedge \psi$可满足的极小子集，所以，$\mathscr{M}$是$\Gamma \diamond_\mu \psi$的模型。

(U7). 设Γ有唯一的模型$\mathscr{M}$。对每一个$\mathscr{M}_1 \in Mod((\Gamma \diamond_\mu \phi) \wedge (\Gamma \diamond_\mu \psi))$，存在两个关于$\leq_{\mathscr{M}_1}$的极小子集$V_1$和$V_2$使得$\mathscr{M} \leftrightarrow_{V_1} \mathscr{M}_1$和$\mathscr{M} \leftrightarrow_{V_2} \mathscr{M}_1$成立，即：$\mathscr{M}_1$是$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1) \wedge \phi$和$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_2) \wedge \psi$的模型。因此，$\mathscr{M}_1 \leftrightarrow_{V_1 \cap V_2} \mathscr{M}$，即$\mathscr{M}_1$是$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1 \cap V_2)$的模型。所以，$V_1 = V_2$，否则$V_1$（或$V_2$）不是关于$\leq_{\mathscr{M}_1}$的极小子集。此外，$\mathscr{M}_1$也是$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1) \wedge (\phi \vee \psi)$的模型。

设$V_3 \subset V_1$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_3) \wedge (\phi \vee \psi)$可满足的原子命题集合，则有$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_3) \wedge \phi$ 或$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_3) \wedge \psi$是可满足的。不失一般性地，设$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_3) \wedge \phi$是可满足的，则$V_1$不是关于$\leq_{\mathscr{M}_1}$的极小子集，与前面的描述矛盾。因此，$V_1$是使得$\mathrm{F}_\mu(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_1) \wedge (\phi \vee \psi)$可满足的极小子集。所以，$\mathscr{M}_1$是$\Gamma \diamond_\mu (\phi \vee \psi)$的模型。

(U8) 对每一个$\mathscr{M} \in Mod((\Gamma_1 \vee \Gamma_2) \diamond_\mu \phi)$都存在一个$\mathscr{M}_1 \in Mod(\Gamma_1)$（或$\mathscr{M}_1 \in Mod(\Gamma_2)$）和一个$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。因此，$\mathscr{M} \models (\Gamma_1 \diamond_\mu \phi) \vee (\Gamma_2 \diamond_\mu \phi)$成立。

类似地，对每一个$\mathscr{M} \in Mod((\Gamma_1 \diamond_\mu \phi) \vee (\Gamma_2 \diamond_\mu \phi))$都存在一个$\mathscr{M}_1 \in Mod(\Gamma_1)$（或$\mathscr{M}_1 \in Mod(\Gamma_2)$）和一个$V_{min}$使得$\mathscr{M} \leftrightarrow_{V_{min}} \mathscr{M}_1$。因此，$\mathscr{M} \models (\Gamma_1 \vee \Gamma_2) \diamond_\mu \phi$成立。 □

7.4 本章小结

本章针对差分隐私数据收集应用中存在的策略型攻击问题，利用信息论、博弈均衡理论研究了隐私防护者与隐私攻击者的理性策略选择，提出了隐私保护的攻防博弈(PPAD)模型，以实现隐私与数据效用均衡。首先，基于信息论度量方法分析差分隐私保护系统中隐私保护者和攻击者的隐私目标，形式化表述为互信息隐私的极大极小问题。其次，针对上述提出的问题，考虑策略型的隐私攻击者和防护者，提出隐私保护的攻防博弈模型，并具体为二人的零和博弈模型。随后，给出博弈的凹凸性以及均衡分析。进一步，为了求解博弈模型鞍点，设计了策略优化选择算法。最后，通过实验阐述了所提出的方案可以用于比较等价的隐私机制，并阐述了隐私量化是最坏情况下的隐私泄露，也即是，隐私防护者的最大隐私泄露。

第八章　总结与展望

本章首先总结文中针对差分隐私保护模型的隐私与效用权衡问题做出的研究工作，概括文中使用的研究方法以及取得的研究成果。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。

8.1　工作总结

大数据时代的在线网络数据(如在线社交网络数据、医疗数据、移动轨迹数据等)促使个人隐私遭受潜在的隐私泄露风险，使得隐私泄露成为数据科学与工程的一个主要关注问题，迫切需要有效的数据隐私保护模型及方法。在诸多的隐私保护模型中，差分隐私逐渐成为数据隐私保护研究与应用中的一个事实上的隐私标准，在隐私保护数据发布、隐私保护数据收集、隐私保护数据分析等场景中得到了广泛的应用。差分隐私主要是利用随机性遏制个人隐私推断问题，其随机性涉及的隐私性与数据效用是研究差分隐私机制设计的核心。依据隐私与效用原则，隐私与效用仅能达到较为理想的平衡折中，这就是学术研究中备受关注的隐私与效用权衡问题。当前的研究工作在面向多维数据处理、属性关联以及关联数据隐私攻击等方面还存在一些不足之处，尚需要深入的研究。为此，本文围绕差分隐私应用中存在的隐私与数据效用的权衡问题，提炼出隐私与效用的度量、权衡隐私与效用的优化模型、隐私保护机制的设计及隐私保护机制的评价方法四个关键的问题。针对此，本文利用信息论、优化理论、对策博弈论方法从均衡优化的角度，研究了差分隐私通信模型及其度量方法、差分隐私的均衡优化模型和差分隐私均衡优化模型的算法，提出了面向关联属性的信息熵度量模型、数据关联的差分隐私优化模型、多维数据有序随机响应扰动方案(ORRP)和隐私保护的攻防博弈模型(PPAD)及其对应的算法。旨在借助信息论的基础方法，通过最优化和均衡的手段，探讨差分隐私的均衡优化方法，实现保护个人隐私的同时维持数据质量。具体的，本文的主要工作总结如下：

(1) 基于Shannon基本通信模型，结合差分隐私随机扰动，构建了差分隐私的基本通信模型，并给出形式化的描述。以此为基础，首先抽象差分隐私数据扰动为有损压缩信道机制。进一步，考虑含关联背景知识的敌手模型，提出了差分隐私含敌手背景知识的通信模型。其次，在通信模型的基础上，引入信息熵、联合熵、条件熵、互信息量以及失真等概念，建立了以信息论方法为核心的差分隐私度量模型，逐步形成差分隐私的信息熵度量体系。随后，以基本的度量为基础，针对多维关联属性的隐私度量问题，利用关联分析、图模型以及马尔可夫隐私链，提出了面向关联属性的差分隐私信息熵度量模型及方法。最后，利用数据处理和费诺不等式提供了相应的分析。

(2) 针对差分隐私数据发布中存在的隐私泄露问题，以所建立的差分隐私通信模型为基础，基于隐私与效用的度量方法，形式化表述了隐私与效用权衡问题，给出互信息隐私优化模型。进一步，针对差分隐私发布中存在先验知识的数据关联问题，考虑了含背景知识的敌手模型。通过引入条件互信息量，针对隐私攻击者完全背景知识、数据管理者拥有统计知识的情景，提出了条件互信息优化模型，用于求解最小化隐私泄露的最优隐私机制。最后，针对所提出的优化模型求解问题，设计了最小化的迭代算法，实验结果表明所提出的方法有效提高了数据质量。

(3) 针对差分隐私在处理多维数据时面临的隐私脆弱性和效率低的问题，利用信息论方法，研究了面向多维数据收集的最优机制问题，提出了有序随机响应扰动方案(ORRP)，有效弥补现有隐私机制忽略考虑先验分布的影响，提供相同属性级隐私保护强度的不足。首先，基于独立并联信道模型，使用分治策略思想分解元组分量。其次，基于隐私与效用度量为基础，针对单属性分量，将满足数据质量损失约束最小化规避隐私风险的隐私机制，形式化表述为一个计算单属性的最优输出概率密度函数的优化问题。然后，将上述推广到多维数据情景，提出了ORRP方案，利用模型计算的概率密度函数实现随机扰动，并给出了对应算法。最后，分析了所提出方案的隐私、效用及相关度损失，并在真实数据集上进行实验，分析所提出方案的优势。

(4) 针对差分隐私中存在策略型的敌手模型，在已构建的差分隐私基本通信模型和基本的度量基础上，分析隐私保护系统参与者的隐私目标，提出了隐私保护的攻防博弈模型(PPAD)，旨在于利用博弈均衡理论实现隐私保护系统中隐私与数据效用的均衡。首先，基于所建立的差分隐私度量模型，定义了隐私保护系统中隐私保护者和隐私攻击者(敌手)的隐私目标，形式化表述为有关隐私泄露的极大极小问题。其次，从参与者、策略空间、效用函数的角度给出了隐私攻防博弈的标准形式描述，构建了两方零和对策博弈模型。进一步，利用极大极小定理、凹凸博弈的理论提供了所建立博弈模型的均衡分析，即鞍点的分析。理论上的分析表明鞍点的存在性，并解释了鞍点在隐私保护中的涵义。对于等价的ε-隐私机制，提出了等价类隐私机制可比较的方法，解决了ε-隐私度量存在的不足。此外，基于交替最优响应策略设计了博弈均衡计算的策略优化选择算法，并给出了实验分析。

8.2 研究展望

当前，差分隐私在数据隐私保护中发挥重要的作用，应用范围涉及数据发布、数据收集、数据分析、机器学习等领域，对其应用的研究仍需要积极的推进。虽然本文基于信息论和对策博弈论的基础理论方法，从均衡优化的角度做出了一些有意义的探索工作，但是本文的研究中尚存在一些值得深入研究的问题。具体包括有：

(1) 在隐私度量方面，研究表达用户隐私敏感偏好强度的度量方法，建立差分隐私ε-度量、用户个性化隐私需求和信息熵度量的联系，为个性化的差分隐私研究奠定

基础。进一步，在面向多维关联数据情景研究并提出个性化的差分隐私方案是一个值得深入研究的重要方向。

(2) 在权衡隐私与效用的优化模型研究方面，研究最大化数据效用的优化模型，并设计具体的隐私机制是非常有价值的方向。其次，基于所建立的差分隐私通信模型，从信道容量的角度考虑差分隐私的最大信息传输率，对差分隐私保护系统中隐私信息率的定量化研究也是一个值得探索的方向。

(3) 在隐私与效用的均衡优化方面，基于博弈均衡理论的指导，利用非完全信息的动态博弈、静态博弈，建立两方或多方的攻防博弈模型探讨差分隐私保护的最优策略问题仍然是值得研究的方向。此外，基于量化信息流思想，构建差分隐私的信息泄露博弈仍然是值得关注的研究点。

参考文献

[1] LAM W K. 硬件设计验证基于模拟与形式的方法[M]. [出版地不详]: 北京: 机械工业出版社, 2007.

[2] 吕毅. 时序逻辑电路的形式验证方法研究[D]. [出版地不详]: 中国科学院研究生院(计算技术研究所), 2000.

[3] JANICK B夏宇闻. System Verilog验证方法学[M]. [出版地不详]: 北京航空航天大学出版社, 2007.

[4] 袁志斌. 软件开发的形式化工程方法[M]. [出版地不详]: 清华大学出版社, 2008.

[5] 古天龙. 软件开发的形式化方法[M]. [出版地不详]: 高等教育出版社, 2005.

[6] 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.

[7] 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.

[8] ROBINSON J A. A machine-oriented logic based on the resolution principle[J/OL]. J. ACM, 1965, 12(1):23-41. http://doi.acm.org/10.1145/321250.321253.

[9] HUGHES G E, CRESSWELL M J, CRESSWELL M M. A new introduction to modal logic[M]. [S.l.]: Psychology Press, 1996.

[10] 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.

[11] HAREL D, et al. First-order dynamic logic[J]. 1979.

[12] 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.

[13] LIN F. A formalization of programs in first-order logic with a discrete linear order[J/OL]. Artif. Intell., 2016, 235:1-25. https://doi.org/10.1016/j.artint.2016.01.014.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] LEINO K R M. Efficient weakest preconditions[J/OL]. Inf. Process. Lett., 2005, 93(6): 281-288. https://doi.org/10.1016/j.ipl.2004.10.015.

[22] 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.

[23] 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.

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

[25] VISSER A. Uniform interpolation and layered bisimulation[M]//Gödel'96 (Brno, 1996). [S.l.: s.n.], 1996: 139-164.

[26] 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.

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

[28] KONEV B, LUTZ C, WALTHER D, et al. Model-theoretic inseparability and modularity of description logic ontologies[J/OL]. Artif. Intell., 2013, 203:66-103. https://doi.org/10.1016/j.artint.2013.07.004.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

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

[37] IEMHOFF R. Uniform interpolation and sequent calculi in modal logic[J/OL]. Arch. Math. Log., 2019, 58(1-2):155-181. https://doi.org/10.1007/s00153-018-0629-0.

[38] FINE K. Failures of the interpolation lemma in quantified modal logic[J/OL]. J. Symb. Log., 1979, 44(2):201-206. https://doi.org/10.2307/2273727.

[39] SCHUMM G F. Some failures of interpolation in modal logic[J/OL]. Notre Dame J. Formal Log., 1986, 27(1):108-110. https://doi.org/10.1305/ndjfl/1093636529.

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

[41] 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.

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

[43] WANG Y, ZHANG Y, ZHOU Y, et al. Knowledge forgetting in answer set programming [J/OL]. J. Artif. Intell. Res., 2014, 50:31-70. https://doi.org/10.1613/jair.4297.

[44] 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.

[45] 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.

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

[47] 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.

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

[49] LIN F. Compiling causal theories to successor state axioms and strips-like systems[J/OL]. J. Artif. Intell. Res., 2003, 19:279-314. https://doi.org/10.1613/jair.1135.

[50] 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.

[51] LIN F. Compiling causal theories to successor state axioms and strips-like systems[J]. Journal of Artificial Intelligence Research, 2003, 19:279-314.

[52] LIN F. On strongest necessary and weakest sufficient conditions[J]. Artificial Intelligence, 2001, 128(1-2):143-159.

[53] 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.

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

[55] D'AGOSTINO G, LENZI G. On modal mu-calculus with explicit interpolants[J]. Journal of Applied Logic, 2006, 4(3):256-278.

[56] GABBAY D M, SCHMIDT R, SZALAS A. Second order quantifier elimination: Foundations, computational aspects and applications[M]. [S.l.]: College Publications, 2008.

[57] 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.

[58] 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.

[59] BOLOTOV A. A clausal resolution method for CTL branching-time temporal logic [J/OL]. Journal of Experimental & Theoretical Artificial Intelligence, 1999, 11(1):77-93. DOI: 10.1080/095281399146625.

[60] ZHANG L, HUSTADT U, DIXON C. First-order resolution for CTL[R]. [S.l.]: Citeseer, 2008.

[61] 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.

[62] 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.

[63] KOZEN D. Results on the propositional mu-calculus[J/OL]. Theor. Comput. Sci., 1983, 27:333-354. https://doi.org/10.1016/0304-3975(82)90125-6.

[64] 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.

[65] 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.

[66] D'AGOSTINO G, LENZI G. On modal mu-calculus with explicit interpolants[J/OL]. J. Appl. Log., 2006, 4(3):256-278. https://doi.org/10.1016/j.jal.2005.06.008.

[67] DAVIS M, PUTNAM H. A computing procedure for quantification theory[J/OL]. J. ACM, 1960, 7(3):201-215. http://doi.acm.org/10.1145/321033.321034.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] ZHAO Y. Automated semantic forgetting for expressive description logics[D/OL]. 2018. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.740373.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] FANG L, LIU Y, VAN DITMARSCH H. Forgetting in multi-agent modal logics[J/OL]. Artif. Intell., 2019, 266:51-80. https://doi.org/10.1016/j.artint.2018.08.003.

[79] FENG R, WANG Y, CHEN P, et al. Strongest necessary and weakest sufficient conditions in s5[C]//Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018). [S.l.]: World Scientific, 2018: 832-839.

[80] 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.

[81] KAUSHIK R, NAUGHTON J F, BOHANNON P, et al. Updates for structure indexes [C]//Proceedings of VLDB'02. [S.l.]: Elsevier, 2002: 239-250.

[82] EMERSON E A. Temporal and modal logic[M/OL]//VAN LEEUWEN J. Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics. Elsevier and MIT Press, 1990: 995-1072. https://doi.org/10.1016/b978-0-444-88074-1.50021-4.

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

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

[85] SZALAS A. Second-order quantifier elimination in modal contexts[C]//European Workshop on Logics in Artificial Intelligence. [S.l.]: Springer, 2002: 223-232.

[86] 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.

[87] 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.

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

[89] 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.

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

致　　谢

“立身以立学为先，立学以读书为本。”读书是韶华之年提高修养、塑造人格、提升能力的基石，厚积薄发的源泉，时至而立之年当以立身、立业、立家。求学之路，始于黄口，而立之年，学有所成。即将毕业之时，学位论文完稿之际，我衷心的感谢在贵州大学计算机科学与技术学院攻读博士学位期间对我关心、支持、鼓励和帮助的老师、同学和家人。

“古之学者必有师。师者，所以传道受业解惑也。”导师彭长根教授在我攻读博士学位期间，给予我了学术科研的悉心指导和帮助，带我走入了密码学与信息安全、数据安全与隐私保护的研究领域。在论文研究的选题、研究过程、论文写作等环节提出了诸多建设性意见和建议，使我很受启发。惑之不解时，彭老师学识渊博给出了启发式的建议，帮助我解决了研究过程中所面临的关键性困难问题。几年来，受彭老师严谨的学术熏陶、谆谆教诲，日渐培养和提高了我的科研学术能力。生活中，彭老师无微不至的关怀和关爱，以及彭老师温馨的学术团队使我感到了幸福和温暖。在此，我要感谢彭老师，感恩彭老师的指导和帮助才有了本文的学术研究成果。在未来的学习和工作中，我会进一步深入的从事该方向的研究。

感谢团队田有亮教授，感谢田老师在论文研究、撰写的过程中所提出的意见，以及生活上所给予的帮助。此外，感谢实验室团队谭伟杰博士、刘惠篮博士、丁红发博士、刘海博士等在日常科研及生活中所给予的帮助。同时，感谢实验室的师兄弟姐妹在日常生活中给予我的帮助，读书期间的温馨和睦相处和茶余饭后时的谈笑风生都让我感受到家的温暖。

感谢贵州大学计算机学院的老师为本文工作所提供的支持和帮助；感谢秦永彬教授、王以松教授等为本文的选题和研究方案的设计所提出的宝贵意见和建议。

感谢我的父母和妻子在学习和生活上给予我的支持和鼓励。在我攻读博士学位期间，父母对我无私地爱，默默的付出和承担着家庭生活的压力；面对学习困境和压力时，父母给我支持和关心；妻子在我他乡求学期间独自抚养、教育年幼的儿子，给我创造一个安心求学的条件，陪伴我这一段人生路走来，付出和努力了很多。

最后，感谢参与该博士学位论文评审、答辩的诸位专家学者，感谢您们为提高我的博士学位论文质量所提出的宝贵修改意见和建议。

攻读博士学位期间科研和论文情况

一、主持或参与科研项目

主持科研项目

1. 贵州省研究生科研基金立项课题：开放数据发布的隐私保护关键技术及隐私量化评估，合同编号KYJJ2017005

参与科研项目

1. 国家自然科学基金重点项目：数据共享应用的块数据融合分析理论与安全管控模型研究，项目基金号U1836205
2. 国家自然科学基金地区项目：理性隐私计算及隐私风险可控技术研究，项目基金号61662009
3. 贵州省科技计划重大专项：大数据安全与隐私保护关键技术研究，合同编号黔科合重大专项字[2018]3001)

二、发表论文

[1] **Ningbo Wu**, Changgen Peng (Corresponding author). An information theoretic approach to local differential privacy data collection [J] IEEE Transaction on Knowledge and Data Engineering (TKDE) SCI 2区，CCF推荐数据挖掘A类期刊，IF 3.856 (Major Revision, Under Review)

[2] **Ningbo Wu**, Changgen Peng (Corresponding author), Kun Niu. A privacy-preserving game model for local differential privacy by using information-theoretic approach[J]. IEEE ACCESS,2020,8:216741-216751. DOI:10.1109/ACCESS.2020.3041854. SCI 2区，IF 3.8

[3] **吴宁博**,彭长根(通信作者),田有亮,牛坤,丁红发.基于率失真的差分隐私效用优化模型[J]. 计算机学报,2020,43(8):1463-1478. DOI:10.11897/SP.J.1016.2020.01463, CCF推荐中文科技期刊A类，贵州大学(一级学术期刊)

[4] **吴宁博**,彭长根(通信作者),牟其林. 面向关联属性的差分隐私信息熵度量方法[J]. 电子学报,2019,47(11):2337-2343. DOI:10.3969/j.issn.0372-2112.2019.11.015, CCF推荐中文科技期刊A类，贵州大学(一级学术期刊)

附：学位论文原创性声明和关于学位论文使用授权的声明

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究在做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

论文作者签名：_____ 日 期：_____年 月

关于学位论文使用授权的声明

本人完全了解贵州大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权贵州大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密论文在解密后应遵守此规定）

论文作者签名：_____导师签名：_____日 期：_____年 月