分类号 TP301 学号 04069050

UDC 密级 公 开

工学博士学位论文

扩展时序逻辑的推理及符号化模型检验技术

博士生姓名 刘 万 伟

学 科 专 业 计算机科学与技术

研 究 方 向 计算机软件与理论

指 导 教 师 王 戟 教授

国防科学技术大学研究生院

二〇〇九年四月

Reasoning and Symbolic Model Checking of Extended Temporal Logics

Candidate: Liu Wanwei

Supervisor: Prof. Wang Ji

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Engineering

in Computer Science and Technology

Graduate School of National University of Defense Technology

Changsha, Hunan, P.R.China

April, 2009

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：扩展时序逻辑的推理及符号化模型检验技术

学位论文作者签名：_____ 日期： 年 月 日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将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、缩印或扫描等复制手段保存、汇编学位论文。

（保密学位论文在解密后适用本授权书。）

学位论文题目：扩展时序逻辑的推理及符号化模型检验技术

学位论文作者签名：_____ 日期： 年 月 日

作者指导教师签名：_____ 日期： 年 月 日

目 录

表 目 录

图 目 录

摘　要

随着计算机软、硬件系统复杂性的日益增长，系统设计和实现的正确性越来越难以得到保证。因此，用以检验系统正确性的形式化方法亟待出现。上个世纪 80 年代提出的模型检验方法被证明是行之有效的系统正确性验证手段。

执行模型检验的算法，对所采用规约语言的类型十分敏感。由于线性框架下的时序逻辑（如 LTL），具有表达能力（相对）较强、直观、兼容性好等特点，使得这类时序逻辑在实际应用中被使用的相对广泛。

但是，在工业界应用中，许多重要的时序性质无法采用 LTL 表达。因此，若干 LTL 的扩展被陆续提出，它们大致分为两类。

1. 一种方法是向时序逻辑中添加无穷多个时序连接子或者正规表达式，以期获得等价于 ω-正规语言的表达能力。这类逻辑如 ETL、FTL、PSL 等。
2. 另一种方法是向时序逻辑中添加二阶量词或者不动点算子。这类时序逻辑诸如 MSOL、QTL、线性 μ-演算等。

这些 LTL 的扩展，都与 ω-正规语言等价。对于这些逻辑本身，人们比较关心下列问题：

1. **判定问题**。即：逻辑的（可满足）判定性及其复杂度。这是在这些逻辑被提出时首先要解决的问题。
2. **公理化问题**。即：能否给出一套针对该种逻辑的可靠完备的公理系统。公理系统往往由若干公理和推理规则构成。这些公理/规则刻画了该种逻辑的实质。
3. **模型检验问题**。对于某种特定的时序逻辑，开发其高效的模型检验算法是人们追求的核心目标之一。同时，有无高效的检验算法也直接影响该种逻辑能否得到广泛应用。

对于时序逻辑的各类 ω-扩展，其公理化以及符号化模型检验算法的研究还具有另外的特殊意义。在线性结构上，等价于 ω-正规语言的时序逻辑具有足够强的能力表达工业界实际应用中用到的各种性质。各种线性时间的时序逻辑，可以看作它们的逻辑片断或者子逻辑。因此，这些逻辑的公理化以及符号化模型检验算法可以派生出它的子逻辑或者实例的的公理系统和符号化模型检验算法。本文主要工作如下：

1. 给出了三类 ETL（ETL_l、ETL_f、ETL_r）的可靠完备的公理系统，同时给出了基于时序算子编码的 ETL 逻辑片段可靠完备公理化方法。

2. 基于博弈方法，给出了μ-演算的公理系统的新的完备性证明。同已有的方法相比，该证明相对直观、简洁。
3. 基于 tableau 方法，给出了三类采用交错自动机作为连接子的 ETL（ATL_f、ATL_l、ATL_r）以及 PSL 的某个变种（APSL）的基于 BDD 的符号化模型检验算法。
4. 分别给出了具有一般形式和特定形式（ν-范式）的线性 μ-演算的基于 BDD 的符号化模型检验算法。
5. 基于上述算法，在模型检验工具 NuSMV 的基础上，实现了支持扩展时序逻辑的符号化模型检验工具 ENuSMV。它允许用户通过描述自动机的方式自定义时序连接子，能够检验全部的 ω-正规性质。

关键词：扩展时序逻辑，μ-演算，公理系统，符号化模型检验，ENuSMV

ABSTRACT

With the never-ending increase of the complexity of software and hardware, correctness of design and implementation becomes more and more difficult to guarantee. Therefore, verification technologies are urgently required in such fields. The model checking approach, presented in the 80s last century, has been proven an applicable technique in verifying the correctness of software and hardware systems.

Verification algorithms is highly sensitive to the type of specification language. Temporal logics under linear framework, such as LTL, benefit from several aspects such as (relative) stronger expressiveness, better intuitiveness, and better compatibility. As a result, linear time temporal logics receive a wide application in practice.

Nevertheless, LTL is still not powerful enough to express all the ω-regular properties used in industrial verification. Hence, many extensions of LTL are suggested, and in a broad sense, these logics can be obtained by the following two approaches.

1. One approach is to add infinitely many operators or regular expressions to obtain logics as expressive as ω-regular languages. Such extensions like ETLs, ForSpec, PSL etc.
2. The other approach is to cooperate temporal logics with second order quantifiers or fixpoint operators. For example, MSOL, QTL, or linear μ-calculus.

All these extensions are known to be as expressive as ω-regular expressions. For these logics, the following issues are mostly concerned.

1. **The decision problem**. i.e., the decidability of this logic and its inherit complexity. Those problems are usually solved at the first moment when the logics are proposed.
2. **The axiomatization problem**. i.e., to give a sound and complete axiom system for the logic. An axiom system usually consists of a set of axioms and deductive rules, which reflect the essence of the logic.
3. **The model checking problem**. For a given temporal logic, finding its effective verification algorithm is always the most important aim in model checking. In fact, a logic is doomed to be rarely used in model checking if its effective verification algorithm does not exist.

For various ω-extensions of linear temporal logic, the study of their axiomatization and symbolic model checking has special significance: These languages are powerful enough to express various specifications used in industrial community. Other temporal logics on linear structures can be viewed as fragments or sub-logics of these extensions. Hence, one can derive their axiom system or symbolic model checking algorithm by a systematic encoding or instantiation.

Major contributions of this thesis are listed as follows.

1. Sound and Complete axiom systems for three kinds of ETLs (i.e., ETL_l, ETL_f, ETL_r) are presented, and it is shown how to obtain sound and complete deductive systems for ETL fragments via instantiating temporal connectives.
2. Based on game theory, new proofs of completeness of μ-calculus axiom systems (both branching and linear) are given. In comparison to the existing approaches, these proofs are relative succinct.
3. BDD based symbolic model checking algorithms for three kinds of ETLs with alternating automata connectives (i.e., ATL_f, ATL_l, ATL_r) and a variant of PSL are given.
4. BDD based symbolic model checking approaches for linear μ-calculus formulas in general and specific forms (namely, ν-form) are respectively presented.
5. An extension of NuSMV, namely ENuSMV, is designed and implemented. ENuSMV is also a symbolic model checker, it allows users to customize their own temporal connectives by writing automata. With which, all the ω-regular temporal properties can be verified.

Key Words: Extended temporal logic, μ-calculus, Axiom system, Symbolic model checking, ENuSMV

第一章　绪论

1.1　引言

随着计算机技术的不断进步，当代软硬件技术的发展日新月异。同时，新的技术使得软、硬件系统复杂性不断提升：规模上百万行的软件屡见不鲜，大型芯片中的基本逻辑门的数量也是以十万为单位计数。这使得系统设计、实现的困难性愈来愈大，从而其正确性的保障问题也变得越来越突出。在安全攸关系统以及嵌入式系统的设计和实现中，正确性保障显得尤为重要— 系统在算法设计和实现上的错误有时会导致十分严重的后果。这些错误多数不是发生在语法层面，不可能完全在编译阶段排除；同时，靠人工的方式进行错误排除也不具有可行性。因而，提供软硬件设计正确性保障的形式化方法的需求也变得越来越紧迫。

对电路设计、程序代码以及网络安全协议的正确性进行检验，一直是计算机科学中的重要研究内容。早在 60 年代晚期，以 Floyd、Hoare[1] 为代表的一批计算机科学家最先研究了顺序程序的正确性证明技术。这类技术所关注的程序语法成分包括：赋值语句（当时的表达式中并未涉及指针操作）、顺序语句、条件分支语句、循环语句等。Hoare 推理系统的基本元素是形如 $\varphi\ \{P\}\ \psi$ 三元组，其中： P 是由一组语句构成的程序块；φ 和 ψ 都是一阶逻辑公式，分别称为 P 的**前置条件** 和**后置条件**。三元组 $\varphi\ \{P\}\ \psi$ 指明了“如果性质 φ 在 P 被执行之前成立，那么 ψ 必然在 P 被执行后成立”。比如，下面的“赋值公理”（Axiom of Assignment）

$$\frac{}{\varphi_t^x\ \{\mathtt{x} := \mathtt{t}\}\ \varphi}$$

就指明了：若要证明 φ 在语句 $\mathtt{x} := \mathtt{t}$ 之后成立，只需证明在该语句执行之前有 φ_t^x 成立即可（φ_t^x 是将 φ 中每个 x 替换为 t 得到的表达式）。Hoare 逻辑能够证明某些程序的正确性或者部分正确性，但其最大的障碍在于证明的过程不能完全自动化的进行。比如，其推理规则中包含如下的“循环规则”（Rule of Iteration）

$$\frac{\psi \wedge \varphi\ \{P\}\ \psi}{\psi\ \{\textbf{while}\ \varphi\ \textbf{do}\ P\}\ \psi \wedge \neg\varphi}$$

其中的 ψ 称为**循环不变式**。能够充当循环不变式的性质与人们期望证明的性质往往并不一致。同时，寻找循环不变式本身也是一件非常困难的任务，它需要对程序

本身有深刻的理解。因此，完全机械的依靠这种推理的方式对程序非平凡的性质进行正确性证明是不现实的。

上世纪 70 年代中期，一类以静态分析[2, 3, 4]技术为代表，针对程序通用错误（比如，内存泄漏错误，空指针引用，数组访问越界等）的检测技术被提出。这类技术技术中包括诸如“数据流分析技术”[5]（Data-Flow Analysis）、“控制流分析”（Control-Flow Analysis）、“基于约束的分析”（Constraint based Approach）以及“抽象解释”[4]（Abstract Interpretation）等等。

静态分析技术仍然是针对源程序（比如 C 代码，或者其语言子集）进行的。目前，工业界应用的静态分析工具在执行程序验证时，存在两个问题：

漏报：某些实际存在错误不能发现；

误报：某些被报告的错误并不是在程序中实际存在的。

一般说来，漏报和误报是一对平衡的矛盾，很难在实际中做到误报率和漏报率都低。此外，静态分析技术需要应对程序指针、数组等语法元素，这些往往难于处理。

到了 80 年代初期，随着并发程序在业界的广泛应用，人们关心的重点转移到算法、协议的逻辑正确性（比如：系统有无死锁，响应性、活性等）。换句话说，人们越来越关心系统设计的正确性。一般说来，能够在越高层面，越早的设计阶段发现逻辑性错误，修正错误和漏洞所需要的代价越小。在这种背景下，新的形式化验证方法的出现就变得非常必要了。

1.1.1 模型检验技术及其特点

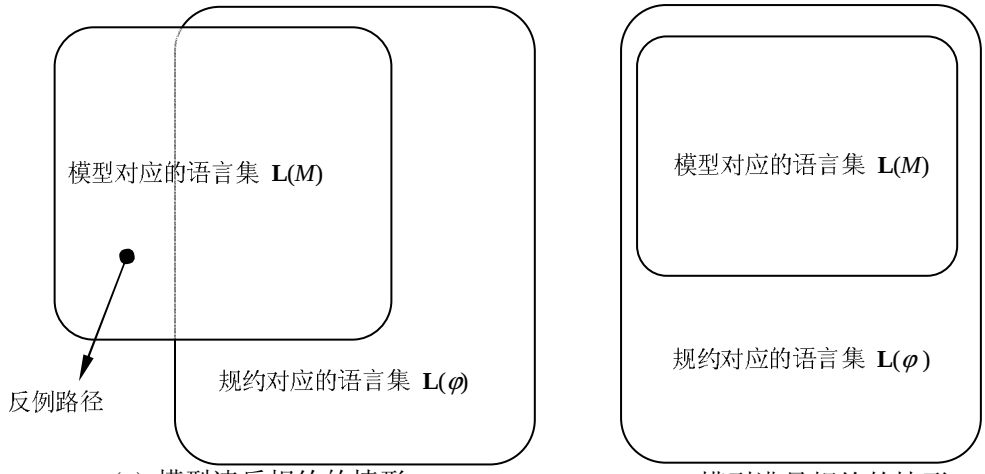
上个世纪 80 年代初，Clarke、Emerson[6]、Queille、Sifakis[7] 等人共同提出了“模型检验”技术。其验证的对象是程序（或者切片）的数学抽象，称为**模型**（或者称为**迁移系统**）。在 “The birth of model checking”[8] 一文中，Clarke 这样对该种技术给出定义：

> “We used the term *Model Checking* because we wanted to determine if the temporal formula φ was true in the Kripke structure M, i.e., whether the structure M was a *model* for the formula φ. ”

同静态分析相比，模型检验主要关注设计的“时序正确性”。它允许验证者精确的指定待检验的性质。

从本质上讲，模型检验问题是一个特殊的证明问题，它要证明的目标是“语言包含问题”，即：模型 M 对应的“语言”是否包含在规约 φ 所对应的“语言”中。如果成立，则可断言 $M \models \varphi$；否则，$\mathbf{L}(M)\backslash\mathbf{L}(\varphi)$ 中的任意执行序列都对应一条违反规

约的**反例路径**。如图 1.1 所示。



(a) 模型违反规约的情形　　(b) 模型满足规约的情形

图 1.1 模型检验目标示意图

自上个世纪 80 年代以来，模型检验技术得到了非常广泛的应用，已被证明是一种行之有效的自动化软、硬件验证技术。它主要应用在“并发系统”（Concurrent System）的正确性验证上。同基于证明的验证方式相比，模型检验具有下列优势。

1. **自动化程度高**：模型检验能够完全自动化的执行，而不需要任何的人工交互。多数的模型检验工具能够以批处理的方式一次性检验多条规约。
2. **可检验非平凡规约**：运行模型检验一般需要两个输入：一个是待检验的模型，另一个是待验证的性质。待验证的性质往往采用时序逻辑公式书写。时序逻辑能够非常灵活的表述诸如“安全性”、“活性”[9] 等时序性质。
3. **算法的终止**：由于模型检验本身要处理的是一个的语言包含问题，对有穷状态系统而言，该问题本身是一个可判定的问题。因而，从理论上讲，在“足够大”的时空开销内，模型检验算法总会停机。
4. **诊断信息**：当发现规约不被满足时，会生成一条反例路径。反例路径是模型中的一条执行序列，该序列能够揭示规约性质为何被违反。反例路径可以作为诊断信息，帮助设计者修正系统的错误。

模型检验技术的本质是“搜索式证明”。其瓶颈在于：问题状态空间的爆炸性膨胀往往会导致验证无法在期望的时空开销内完成。同时，主流的模型检验问题处理的对象是有穷的。虽然某些方法可以应用在诸如参数化系统（Parameterized System）、下推系统（Pushing-down System）的检验中（分别见 [10]、[11]、[12]）。但通常而言，无穷系统的模型检验问题是不可判定的[13, 14]。

1.1.2 时序逻辑的推理及检验

时序逻辑（Temporal Logic）在模型检验中处于核心地位，这类逻辑主要关心事件发生的顺序及时刻，用以充当模型检验中的规约语言。评价一种时序逻辑可用性的标准包含以下几个方面：

1. **直观性**：用该种时序逻辑描述规约时是否直观易懂。
2. **表达能力**：采用该种时序逻辑作为规约语言时，是否有足够的能力来刻画待验证的性质。
3. **易实现性**：采用该种逻辑作为规约语言的检验算法是否易于实现。

因此，对于特定的时序逻辑而言，人们主要研究三个方面的问题。

1. **判定问题**。即：逻辑的（可满足）判定性及其复杂度。这是在这些逻辑被提出时首先要解决的问题。
2. **公理化问题**。即：能否给出一套针对该种逻辑的可靠完备的公理系统。公理系统往往由若干公理和推理规则构成。这些公理/规则刻画了该种逻辑的实质。
3. **模型检验问题**。对于某种特定的时序逻辑，开发其高效的模型检验算法是人们追求的核心目标之一。同时，有无高效的检验算法也直接影响该种逻辑能否得到广泛应用。

时序逻辑的公理化是一个语法范畴的问题。它侧重于不依赖任何语义模型的推理证明。

从公理系统中的公理（实例）出发，经过一系列推理规则重写得到的公式称为公理系统的**定理**。一般而言，逻辑的公理系统的元性质包含如下几个方面：

可靠性 ：该公理系统中可证的定理都是该种逻辑中的效公式。

完备性 ：该种逻辑中的每个有效公式都能在该公理系统中被证出。

独立性 ：该公理系统中的每条公理和推理规则都无法由其他的公理和规则证出。

时序逻辑的推理问题之所以重要，是因为它研究的核心在于时序连接子或者二阶算子的性质。它能够反映这些连接子原始语义定义之外的其他性质。比如，在 LTL（Linear Temporal Logic, [15]）中，公理

$$\varphi_1 \mathsf{U} \varphi_2 \leftrightarrow (\varphi_2 \vee \varphi_1 \wedge \mathsf{X}(\varphi_1 \mathsf{U} \varphi_2))$$

是在符号化模型检验技术中构建公式 tableau 的关键（参见文 [16]）。

此外，时序逻辑的推理系统也是模型检验重要辅助手段，在某些针对特殊无穷系统的模型检验技术中，结合定理证明的技术能够有效的应对系统状态空间的膨胀。

与之相比，模型检验问题是一个语义层面的问题，它实质上是要检查模型的派生语义结构是否满足规约。

对于特定的时序逻辑而言，开发其高效的模型检验算法是非常重要的研究问题之一。大体上，模型检验算法可分为两大类。

显式模型检验 ：这类方法较早应用于线性框架下时序逻辑的模型检验算法之中。它的主要思想是构建规约之非的语言模型（比如，自动机），而后通过（on-the-fly）搜索反例路径的方式执行检验。

符号化模型检验 ：这种检验方式首先通过编码的方式将模型以及规约进行转换，使用布尔公式或者其他形式来对状态空间进行抽象，以避免显式的状态搜索。

虽然，从本质上讲，采用何种检验算法并不会改变问题固有的复杂度，但是，实验表明符号化的确能够有效的缓解状态空间的膨胀问题。同时，符号化方法是从无穷状态空间抽象出有穷表示的有力手段。正如 Wolper[17] 所指出的那样：

> "At the heart of all the techniques that have been proposed for exploring infinite state spaces, is a symbolic representation that can finitely represent infinite sets of states."

事实上，符号化模型检验算法是当前主流的验证方法。

符号化模型检验算法主要分为两大类：一类是基于 BDD（Binary Decision Diagram）等决策图[18]的检验算法，这类方法最终归结为 CTL（Computation Tree Logic, [19]）的符号化模型检验过程；另外一类是"限界模型检验"（Bounded Model Checking）[20]，这类方法最终归结为布尔公式的 SAT 问题。一般而言，前者对性质是否被满足不敏感；后者在性质被违反时，具有较高的检验效率，而在性质被满足时可能会产生非常高的检验代价。本文主要研究基于 BDD 的扩展时序逻辑的符号化模型检验算法，这一方面是出于对主流工具支持的考虑，另一方面是因为限界模型检验技术中 ω-正规性质 SAT 编码的复杂性。

1.2 研究目标及主要结果

较早的作为模型检验规约语言的时序逻辑有 LTL[15, 21]，CTL[19] 等。正如 Vardi[22] 所指出：线性框架下的时序逻辑具有表达能力（相对）较强、直观、兼容性好等特点，使得这类时序逻辑在实际应用中被使用的相对广泛。

Wolper 最早发现 LTL 并不能表达全部的 ω-正规性质[23]。事实上，它的表达能力恰好等价于 star-free 的 ω-正规语言[24, 25]（即：不使用"$*$"-操作子的 ω-正规语

言)。Pnueli 在文 [26] 和 [27] 中指出了规约语言具备完整的 ω-正规语言表达能力的重要性。比如：在组合模型验证（Modular Model Checking）技术中，要求规约语言必须能够描述全部的 ω-正规性质。甚至，有的学者这样评价针对 ω-正规性质推理和检验的重要性 [28]：

> "If one surveys much of the recent work devoted to the algorithmic verification of infinite-state systems, it quickly appears that regular languages have emerged as a unifying representation formalism for the sets of states of such systems.
>
> ... ...
>
> Indeed, regular languages described by finite automata are a convenient to manipulate, and already quite expressive formalism that can naturally capture infinite sets."

许多与 ω-正规语言等价的线性时序逻辑的扩展被提出。它们大致分为两类：

- 一类是将时序逻辑构筑于二阶量词或不动点算子。这类逻辑一般只包含有穷个算子，语法成分相对简洁。这类逻辑如 MSOLMonadic Second Order Logic，[25]、线性 μ-演算[29] 等。
- 另外一类是向时序逻辑中添加无穷多的时序算子（如 ω-文法、ω-自动机）或者直接使用正规表达式作为公式元素，以期获得等价于 ω-正规语言的能力。这类逻辑如 ETL（Extended Temporal Logic，[30]）、PSL（Property Specification Language，[31]）等等。

在本文中，将这两类逻辑统称为**扩展时序逻辑**。

事实上，对第二类时序逻辑扩展而言，可以根据实际的需要添加各种连接子，这样，就有机会在复杂性和表达能力之间取一个折衷。同时，这些时序逻辑构成了一个逻辑"谱系"[32] — 该谱系中的任何一种逻辑都可以看作是 LTL 逻辑扩展的"片断"（Fragment）或者"实例"（Instance）。

就时序逻辑的公理化问题而言，往往会针对采用连接子的不同而采取不同的公理化方法。对模型检验问题来说，其具体的实现算法对规约所采用的时序逻辑种类非常敏感。人们往往针对具体的逻辑开发单独的模型检验技术。

本文的主要研究目标在于：建立关于各类扩展时序逻辑及其逻辑片段的公理化以及符号化模型检验的统一框架。

本文主要工作及结果如下：

- 基于图删除技术，分别给出了三类 ETL（即：ETL_l、ETL_f 以及 ETL_r）的可靠

完备公理系统（$\mathscr{L}$、$\mathscr{F}$ 以及 $\mathscr{R}$）。此外，给出了基于时序连接子编码的 ETL 逻辑片段的时序连接子到对应基逻辑的实例公理化方法，从而可以派生出这些逻辑片段的可靠完备公理系统。

- 给出了新的针对 μ-演算公式的博弈系统。证明了该种博弈的取胜策略与公式可满足性之间的关系。基于博弈方法，给出了模态 μ-演算和线性 μ-演算公理系统完备性的简洁证明。
- 扩展了 LTL 的 tableau 方法。给出了三类使用交错自动机连接子作为连接子的 ETL（即：ATL_f、ATL_l 以及 ATL_r）的符号化模型检验算法。此外，还给出了 PSL 的某个变种（APSL）的符号化模型检验算法。这些算法，都是将线性框架下的时序逻辑的模型检验问题转化成为分支框架下时序逻辑（确切地说，CTL）的模型检验问题。在本文给出的符号化模型检验算法中，除编码 ATL_r 公式 tableau 所需的位变元数目正比于公式长度的平方外，编码其余几种逻辑公式的 tableau 引入的位变元数目均与公式长度成线性关系。
- 基于可编码的自动机转换过程和扩展的博弈系统，分别给出了具有一般形式和特定形式（ν-范式）线性 μ-演算公式的符号化模型检验算法。对于一般形式的线性 μ-演算公式而言，对其执行符号化模型所引入位变元数目正比于公式中 $\bigcirc$ 算子和公式中约束变元数目之积的平方。而对于具有 ν-范式形式的公式而言，额外引入的位变元数目正比于公式长度。
- 在模型检验工具 NuSMV 的基础上，实现了对 ETL_f 和 APSL 符号化模型检验的支持。它允许用户通过描述自动机的方式来自定义时序连接子。实验结果表明，这两种时序逻辑都能够较为高效的采用基于 BDD 的符号化方法检验。

该项工作的意义在于：

1. 对于 ETL 的公理化问题，针对三种 ETL 公式，分别定义了公式迁移图的“局部一致性” 和“全局一致性” 的概念。证明了可满足公式的迁移图中必然存在满足这两种一致性的路径。“全局一致性” 实质上反映了反应了 finite、looping 以及 repeating 自动机连接子的公共性质。同时，这些逻辑的公理系统中提供了刻画全局一致性的推理规则。对于不可满足的公式 φ 而言，其对应的公式迁移图中必然有某条**踪迹**（是一个公式序列）。于是，可以根据这条踪迹构造 $\neg\varphi$ 在相应公理系统中的证明。这样的方法具有一般性，它是 Wolper 公理化方法 [23] 的扩展。因而，它是 ETL 逻辑片段公理化方法的更加通用的手段。
2. 作为对比，另外一类构建于二阶量词或者不动点算子的时序逻辑（如 QTL、MSOL 以及 μ-演算等）具有简洁的语法结构。由于 QTL 和 MSOL 等不是初等可判

定的 [33, 25]，它们较少在实际的模型检验中被使用，所以本文主要研究了 μ-演算的公理化及符号化模型检验问题。事实上，Kozen 已经给出了关于不动点算子的核心公理和规则，但其完备性证明较为困难[34]。本文扩展了文 [35] 中的方法，基于博弈理论给出了其完备性较为简洁的证明。这种方法对分支时间和线性时间的 μ-演算均有效。特别的，线性时间的博弈系统是 ν-范式公式符号化模型检验的基础。

3. 本文的一个研究重点是各类 ETL 的符号化模型检验算法。ETL 中具有丰富的时序连接子，其中采用交错 finite、looping 以及 repeating 自动机编码的时序连接子分别反映了活性、安全性以及一般 ω-正规性质。绝大多数线性框架下的时序逻辑均可以通过编码时序连接子的方式嵌入到 ETL 中。因此，该种方法提供了第一类扩展方式中的时序逻辑及其片段符号化模型检验算法的统一框架。从方法的角度讲，它是 LTL 基于 tableau 符号化模型检验算法的扩展。这种符号化模型检验算法可以在已有模型检验工具 NuSMV 的基础上进行较少的修改便可实现。
4. 线性 μ-演算的符号化模型检验算法相对被研究的较少，它可以看作是对模态 μ-演算符号化模型检验方法的补充。本文提供的针对一般形式公式的模型检验算法是基于可编码的自动机转换进行的。作为对比，针对具有 ν-范式公式的模型检验算法是基于本文给出的博弈系统（的变形）给出的，它具有较低的复杂度。任何 ω-正规性质均能以该种形式的公式表示，特别的，一些常用的时序逻辑，如 LTL，可以以线性代价转化为该种公式。
5. 模型检验工具的设计和实现也是一个重要的研究内容。通过扩充语法成分而获得的 ENuSMV 工具在语法上兼容 NuSMV。它允许用户自定义时序连接子。此外，该工具还支持 APSL 的符号化模型检验，它是 PSL 的一种变种。目前，工业界使用的支持 PSL 的模型检验工具，如 RuleBase[36]，ZeroIn 等并没有完全开放其算法。因此，该开源工具可以看作是一种实现。

1.3 相关研究工作回顾

本节简要回顾与本文工作相关的若干工作。主要包括：模型检验技术的主要发展过程；时序逻辑公理化及推理问题的主要结果；以及若干已有的针对 ω-正规性质检验的方法。

1.3.1 从显式模型检验到符号化模型检验

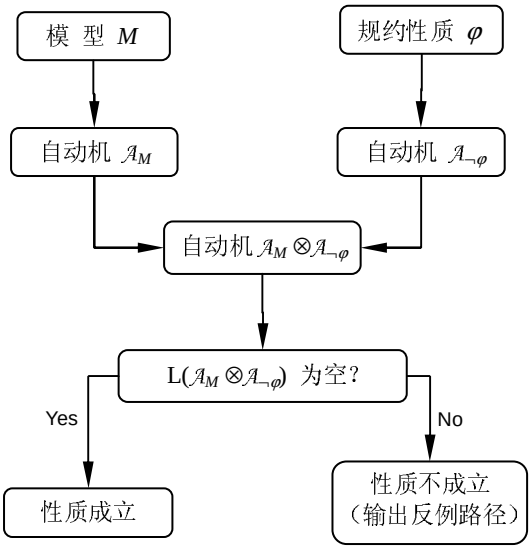


图 1.2 基于自动机的 LTL 模型检验框架

最早的模型检验框架由 Wolper、Vardi、Sistla 等人提出（见文 [37]）。该框架最早是针对 LTL 的模型检验而提出的，也可以扩展到其他线性框架下时序逻辑的模型检验问题中。

如图 1.2 所示，给定模型 M 以及 LTL 公式 φ，在检验 $M \models \varphi$ 是否成立时，需要按下列步骤进行：

- 将模型 M 转化为 Büchi 自动机（见定义 2.1.9） $\mathcal{A}_M$，使得 $\mathbf{L}(\mathcal{A}_M) = \mathbf{L}(M)$（见定义 2.1.12 以及定义 2.3.2）。
- 由 LTL 公式 φ 得到 Büchi 自动机 $\mathcal{A}_{\neg\varphi}$，使得 $\mathcal{A}_{\neg\varphi}$ 恰好接收违反 φ 的线性结构（这里，一个线性结构可以视为一个程序执行序列）。
- 接下来，构造 $\mathcal{A}_M$ 与 $\mathcal{A}_{\neg\varphi}$ 的乘积 $\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi}$，它满足：$\mathbf{L}(\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi}) = \mathbf{L}(\mathcal{A}_M) \cap \mathbf{L}(\mathcal{A}_{\neg\varphi})$。
- 这样，$M \models \varphi$ 当且仅当 $\mathbf{L}(\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi}) = \emptyset$。因此，只需要对自动机 $\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi}$ 接收的语言进行判空即可— 若为空，则表示 $M \models \varphi$；否则，$M \not\models \varphi$，并且 $\mathbf{L}(\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi})$ 中的任意一个线性结构都是 M 中违反 φ 的执行序列（也即，反例路径）。

该技术路线为模型检验工具 SPIN[38] 所采用。在该过程中，由 M 到 Büchi 自动机 $\mathcal{A}_M$ 的复杂度是线性的；而构造自动机 $\mathcal{A}_{\neg\varphi}$ 的代价却为 $2^{\mathcal{O}(|\varphi|)}$，其中 $|\varphi|$ 为 φ 的公式长度。于是 $\mathcal{A}_M \otimes \mathcal{A}_{\neg\varphi}$ 的状态数目为 $\mathcal{O}(|M| \times 2^{|\varphi|})$。同时，（非确定）Büchi

自动机的判空问题实质上是 *PATH* 问题，是 NLSPACE-complete 的。这样，采用上述过程执行 LTL 模型检验的时间复杂度为 $\mathcal{O}(|M| \times 2^{|\varphi|})$[39]。这样，随着模型的规模的增大以及规约长度的增加，执行检验的复杂度会迅速增加，往往会遇到状态空间膨胀的问题。

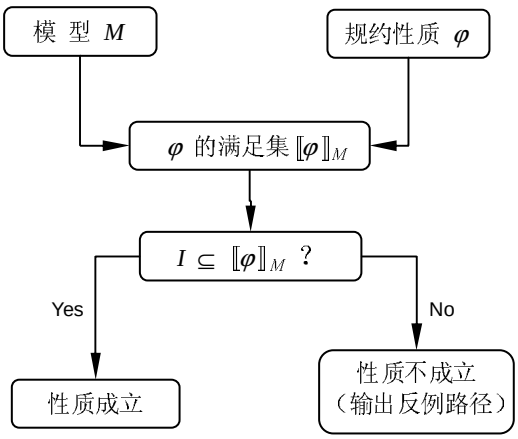


图 1.3 基于满足集的 CTL 模型检验框架

作为对比，早期的 CTL 模型检验算法相对高效。这是因为 CTL 可以嵌入到模态 μ-演算中，其语义可以基于状态集给出（见定义 2.3.17）。即：可以将公式 φ 映射成为模型中的状态集 $[\![\varphi]\!]_M$。因此，对于任给的规约 φ，可以采用“自底向上”的方式，从原子命题开始，逐个计算其子公式的“满足集”（也就是其语义函数对应的状态集）。CTL 的模型检验框架如图 1.3 所示。

对 CTL 而言，$M \models \varphi$ 当且仅当 $I \subseteq [\![\varphi]\!]_M$，而求解 $[\![\varphi]\!]_M$ 的复杂度为 $\mathcal{O}(|M| \times |\varphi|)$。更重要的是，此过程可以非常自然的采用基于 BDD 的符号化方式实现（见 2.3.3 节）。因此，早期的 CTL 模型检验工具，如 SMV[40]等，能够非常高效的执行模型检验[41]。

为提高 LTL 的模型检验效率，在 90 年代中期，Clarke 等人给出了基于 BDD 的 LTL 符号化模型检验算法[16]。如图 1.4 所示，该算法的核心思想是：

- 对于给定模型 M 以及 LTL 公式 φ 而言，$M \not\models \varphi$ 当且仅当 M 满足 CTL* 公式 $\mathsf{E}\neg\varphi$（见定义 2.2.7 以及定义 2.2.8）。
- 构造关于 $\neg\varphi$ 的 tableau $\mathcal{T}_{\neg\varphi}$，它恰好生成所有违反 φ 的路径。与 Büchi 自动机不同的是，tableau 可以采用一组位变元进行状态空间的表示；同时，它的迁移关系也可以用一组布尔公式统一表示。因而，它可以非常方便的采用 BDD 进行编码。

- 因此，$M \not\models \varphi$ 当且仅当 $M \| \mathcal{T}_{\neg\varphi}$ 中存在一条公平路径。这样，LTL 的模型检验问题就转换成了 CTL 模型检验问题：$M \not\models \varphi$ 当且仅当 $M \| \mathcal{T}_{\neg\varphi} \models \mathsf{EG}\ true$。

上面的几个模型检验框架，同样适用于带有公平性约束的迁移系统— 对于在 M 上增加公平性约束 $\mathcal{C}$ 后的公平迁移系统 $\mathcal{M}$（见定义 2.3.7），其模型检验过程相似，只是将相关操作（比如模型到 Büchi 自动机的转换，迁移系统合成等）换为针对公平迁移系统的版本即可。

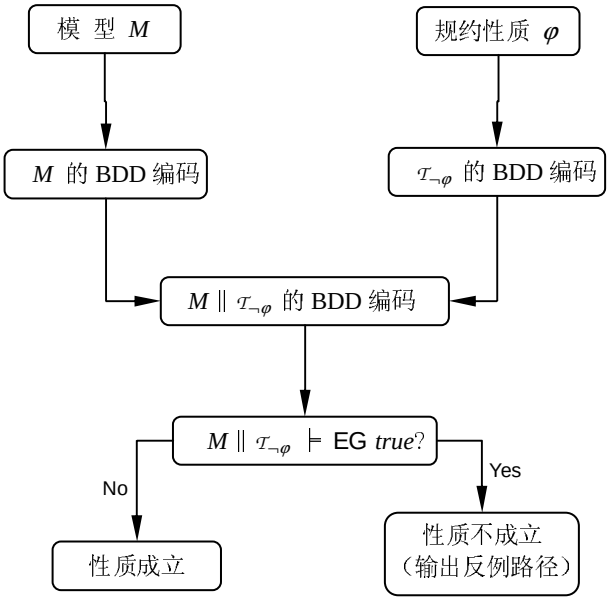


图 1.4 基于 BDD 的 LTL 符号化模型检验框架

在 Clarke、Grumberg 以及 Peled 编著的“Model Checking” 一书[42]的第六章中，LTL 的符号化模型检验算法被扩展至 CTL* 的模型检验中。苏（开乐）、吕（关锋）、洛（翔宇）等人给出了该算法的实现，同时开发出支持 CTL* 符号化模型检验的工具 MCTK（见文[43]）。

采用基于 BDD 的方式进行符号化模型检验时，检验的效率受 BDD 中位变元排序的影响很大。同时，由于 LTL 的模型检验问题的固有复杂度是 PSPACE-complete 的（见文[44]），因而基于 BDD 的方法仍然会遇到状态空间膨胀的瓶颈。为进一步提高模型检验的效率，Biere、Cimatti、Clarke 和 Zhu 等人提出了“限界模型检验”（Bounded Model Checking）的方法[20]。

限界模型检验的主要思想是：对于模型 M 和规约 φ，首先限定一个阈值 k，只检查 M 中是否有长度不超过 k 的执行序列违反规约 φ：若有，则检验过程停止，报告错误；否则，增加 k 的值，重复此过程。当 k 大于某个特定值（称为**完备阈值**）

后，若尚未发现反例路径，则停止检验，报告性质 φ 被模型 M 满足。

在对每个长度不超过 k 的路径执行性质检验时，需要将 M 的 k 步展开以及 $\neg\varphi$ 在长度为 k 的路径上（可能带有回路）的可满足条件进行布尔编码。该布尔编码是基于 k 个系统变元向量 $\vec{Z}_0, \ldots, \vec{Z}_{k-1}$ 的布尔公式。$M \models \mathsf{E}\neg\varphi$ 当且仅当该布尔编码是可满足的。这样，LTL 的模型检验问题就转化成了一个 SAT 问题。同时，当 k 的值取定后，将模型和（取非后）的规约编码的（时间）代价为 $\mathcal{O}(|\varphi| \times k \times |M|)$。因而，对于给定阈值的限界模型检验问题是 NP-complete 的[20]。

当违反性质的路径存在，且长度相对较短时，限界模型检验技术的确能够更有效的执行验证。然而，判定完备阈值的取值，仍是一个 PSPACE-complete 的问题。

此外，限界模型检验也可以应用于针对分支结构时序逻辑的检验中。在文 [45, 46] 中，Zhang（张文辉）等人给出了关于 ACTL 限界模型检验的若干有趣结果。

1.3.2 时序逻辑的推理与公理化

关于时序逻辑的公理化工作最早可以追溯到 Gabbay、Pnueli、Shelah 以及 Stavi 等人关于 LTL 公理化的研究（见文 [21]）。当时的公理系统中包括 8 条公理以及 3 条推理规则。其中，公理

$$\varphi_1 \mathsf{U} \varphi_2 \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \mathsf{X}(\varphi_1 \mathsf{U} \varphi_2))) \tag{1.1}$$

实质上刻画了 U 的不动点性质；而推理规则

$$\frac{\varphi}{\mathsf{G}\varphi} \tag{1.2}$$

刻画了“永真性被时序连接子 G 保持”这样的性质，它非常类似于一阶逻辑公理系统中的 (Gen) 规则。

在文 [47] 中，Emerson 和 Harpern 给出了CTL 的公理系统，该系统与上述 LTL 的公理系统最大的区别在于将 (1.1) 式中 U 算子的不动点刻画分成了 AU 和 EU 版本，即：

$$\mathsf{A}(\varphi_1 \mathsf{U} \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \mathsf{AXA}(\varphi_1 \mathsf{U} \varphi_2))) \tag{1.3}$$

$$\mathsf{E}(\varphi_1 \mathsf{U} \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \mathsf{EXE}(\varphi_1 \mathsf{U} \varphi_2))) \tag{1.4}$$

它们提供了 CTL 到模态 μ-演算嵌入（见 37 页的归纳定义）的依据。值得一提的是，在该系统中给出了一条类似于“极大不动点归纳”的规则，即：

$$\frac{\psi \rightarrow (\varphi_2 \wedge \mathsf{AX}(\psi \vee \mathsf{A}(\psi \mathsf{R} \varphi_2)))}{\psi \rightarrow \mathsf{A}(\varphi_1 \mathsf{R} \varphi_2)} \tag{1.5}$$

将该规则取对偶后，立即可以得到 U 算子的可满足性约束。

关于 CTL* 的完备推理系统出现的较晚。直到 2002 年，才由 Pnueli 和 Kesten 给出一个包含 10 条推理规则的 CTL* 推理系统（见文 [48]）。但这个推理系统中的某些推理规则并不完全独立于模型— 某些推理规则前件中需要关心所在模型的迁移关系。此外，Reynolds 在 2005 年给出了包含“过去时态连接子” 的 CTL* 公理系统（见文 [49]）。

Pnueli 等人给出的 LTL 公理系统是根据极大协调公式集给出的。Lange 和 Stirling 在 2001 年给出了一种基于焦点博弈（Focus Game）的 LTL 和 CTL 的可靠完备公理系统[35]。该系统包含 7 条公理（实际上只需 6 条既可，原文中的第三条公理可经第二条与第七条证出）和 3 条推理规则。其中，规则 (Rel)

$$\frac{\psi \to (\varphi_2 \wedge (\varphi_1 \vee \mathsf{X}((\varphi_1 \vee \psi)\mathsf{R}(\varphi_2 \vee \psi))))}{\psi \to (\varphi_1 \mathsf{R} \varphi_2)} \tag{1.6}$$

就刻画了 LTL 焦点博弈的取胜条件。对于 CTL，该规则分成了 (ARel) 和 (ERel) 两个版本— 只需将 (1.6) 式的 X 和 R 分别替换为 AX 和 AR 以及 EX 和 ER 即可。

使用博弈的方法证明公理系统的完备性，其过程相对简洁。在本文第 4 章，会将该技术推广至模态 μ-演算和线性 μ-演算。

另外一类非常重要的时序逻辑是等价于 ω-正规性质的线性框架下的时序逻辑：比如，ETL[23, 30]、QTL[50, 24, 25]、线性 μ-演算[29]等等。这类逻辑的推理系统的研究具有特殊意义：ω-正规性质具有良好的操作封闭性（比如：并、交、补）。此外，绝大多数线性框架下的时序逻辑的表达能力都是 ω-正规集的子集，从而能够“嵌入”到这些逻辑中去。

在文 [23] 中，Wolper 通过使用 ω-正规文法作为连接子，首先将线性时序逻辑的表达能力扩展至完全的 ω-正规集。在该文中，他还同时给出了关于该种逻辑的公理系统。该系统包括 2 条公理和 4 条规则（但实际上还应该包括 (MP) 规则）。这两条公理实际上是两条元公理（Meta Axiom）：

$$\mathcal{R}_i(\varphi_1, \ldots, \varphi_n) \to \bigvee \varphi_{(i,j)} \wedge \bigcirc \mathcal{R}_{(i,j)}(\varphi_1, \ldots, \varphi_n) \tag{1.7}$$

$$\psi_i \wedge \bigwedge_k \Box(\psi_k \to \bigvee_j (\varphi_{(k,j)} \wedge \bigcirc \psi_{(k,j)})) \to \mathcal{R}_i(\varphi_1, \ldots, \varphi_n) \tag{1.8}$$

其中，$\mathcal{R}$ 的终结符集合为 $\{a_1, \ldots, a_n\}$，非终结符集合为 $\{N_1, \ldots, N_k\}$，每个文法推导规则形如 $N_i \Rightarrow a_{(i,1)}N_{(i,1)} \mid \ldots \mid a_{(i,m)}N_{(i,m)}$。$\mathcal{R}_i$ 是将 $\mathcal{R}$ 的起始符号设为 N_i 得

到的新的文法。在公式 (1.7) 和公式 (1.8) 中，时序连接子 $\bigcirc$ 和 $\Box$ 分别与 LTL 中的 X 和 G 等同。在本文第 3 章中，会给出其他类型 ETL 的公理化系统。

Kesten 和 Pnueli 在文 [51] 中给出了关于 QTL 的公理系统。该系统讨论的 QTL 中包含过去时态连接子，共有 14 条公理（其中有关过去时态连接子的公理有 6 条）以及 3 条推理规则。其中，最为核心的是关于二阶量词的两条公理 (Q1) 和 (Q2)，分别描述为：

$$\mathsf{G}(\forall Z.\mathsf{X}\varphi \leftrightarrow \mathsf{X}\forall Z.\varphi) \tag{1.9}$$

$$\mathsf{G}(\forall Z.\varphi \rightarrow \varphi^{Z}_{\psi}) \tag{1.10}$$

其中，在公式 (1.10) 中，要求 ψ 对 φ 中的 Z 可代入。

关于线性 μ-演算的推理系统被研究的相对较少。Kaivola 在文 [52] 以及他的博士论文[53] 中给出了一个非常简洁的线性 μ-演算公理系统，该系统仅包含 4 条公理和 3 条推理规则。但是，其完备性证明非常复杂。Dax、Hofmann、Lange 在文 [54] 中给出了一个关于线性 μ-演算公式的“重写系统”，它仅包含 4 条重写规则：

$$\frac{\vdash \varphi,\psi,\Gamma}{\vdash \varphi\vee\psi,\Gamma} \qquad \frac{\vdash \varphi,\Gamma \quad \vdash \psi,\Gamma}{\vdash \varphi\wedge\psi\Gamma} \qquad \frac{\psi^{X}_{\mu X.\psi},\Gamma}{\mu X.\psi} \left(\frac{\psi^{X}_{\nu X.\psi},\Gamma}{\nu X.\psi}\right) \qquad \frac{\Gamma}{\bigcirc\Gamma,\Delta}$$

但是，该系统的“证明” 并非通常意义的由公理和推理规则得到目标公式的过程，而是以目标公式为根，通过重写规则，构造满足特定约束的无穷树的过程。与普通的重写系统不同的是，在 Dax、Hofmann、Lange 的系统中，如果根节点的公式是永真式，那么每个节点的公式集会保持析取有效。更确切的说，这些规则提供的实际是一组公式可满足性测试手段。

事实上，μ-演算的分支时间版本—模态 μ-演算，有着更为广泛的应用。早在 1983 年，Kozen 就给出了一套关于该种时序逻辑的公理系统（见文[55]）。但其完备性直到 1995 年才由 Walukiewicz 通过一种非常复杂的技术证明[56, 34]。

时序逻辑的推理技术可以作为模型检验的重要辅助手段。比如，Berezin 在其博士论文[11]中给出了一种将模型检验和定理证明相结合的手段，可以处理某些无穷状态系统的验证。Berezin 的系统中包含一系列形如 $M;\Sigma;\Gamma\vdash\Delta$ 的规则。其中，M 是一个迁移系统，Σ、Γ、Δ 均为一阶模态 μ-演算公式集。$M;\Sigma;\Gamma\vdash\Delta$ 成立当且仅当 $M|_{\Sigma}\models\bigwedge\Gamma\rightarrow\bigvee\Delta$ 成立。其中，$M|_{\Sigma}$ 是将 M 压缩在满足 $\bigwedge\Sigma$ 的状态集上得到的迁移系统。在 Berezin 的系统中，多数推导规则类似于 Gentzen 的相继式推演系

统中的规则。但其中较为关键的仍然是关于不动点算子的，比如：

$$\frac{M;\Sigma,\psi_{\alpha}^{X}\rightarrow\alpha;\Gamma\ \vdash\ \Delta,\alpha}{M;\Sigma;\Gamma\ \vdash\ \Delta,\mu X.\psi} \tag{1.11}$$

其中 α 是未经解释的新的 Skolem 谓词。这条规则实际上是由 Kozen 系统中的规则

$$\frac{\psi_{\varphi}^{X}\rightarrow\varphi}{\mu X.\psi\rightarrow\varphi} \tag{1.12}$$

变形得到的。

1.3.3 ω-正规性质的检验方法

目前工业界主要应用的一类规约语言，是通过将正规表达式、正规文法或者自动机作为语法成分，结合布尔算子构成的时序逻辑。这类逻辑经过不断演化，已经逐渐发展为工业标准（如 PSL）。使用这些等价于 ω-正规语言的逻辑做规约时，不宜采用先将其转化为 ω-自动机，而后进行模型检验的算法— 这是因为从时序逻辑到（非确定）自动机转化过程通常伴随高昂的时空代价，从而会进一步加剧状态空间膨胀的问题。

本文一个研究的重点是具有等价于 ω-正规语言描述能力的时序逻辑的符号化模型检验算法。在本节，回顾各种针对这类语言已有检验方法。

较早的一 ω-自动机作为规约进行检验的工作可以追溯到 Grumberg 和 Long 在组合模型检验（Modular Model Checking）中用到的技术（见文 [57]）。组合模型检验的主要思想是将迁移系统进行分解，它的验证规则是一系列形如 $\langle\varphi\rangle M\langle\psi\rangle$ 的三元组，其意义为：对迁移系统 M 而言，在性质 φ 成立的前提下，性质 ψ 也成立。即：若 $M\models\varphi$，则 $M\models\psi$。在组合模型检验中，存在如下的核心规则：

$$\frac{M\preceq\mathcal{A};\qquad\langle\mathcal{A}\rangle M'\langle\psi\rangle;\qquad\langle\psi\rangle M\langle\varphi\rangle}{\langle true\rangle M\|M'\langle\varphi\rangle} \tag{1.13}$$

其中，$\mathcal{A}$ 代表某 Büchi 自动机（所对应的性质），$M\preceq\mathcal{A}$ 是指 M 能够被 $\mathcal{A}$ **模拟**，它蕴涵 $\mathbf{L}(M)\subseteq\mathbf{L}(\mathcal{A})$。在检验 $M\models\mathcal{A}$ 是否成立时，[57] 中采用的做法是采用“符号化迭代”，并且当时并没有考虑公平性条件以及 $\mathcal{A}$ 的接收条件。

另外一种针对 ω-自动机的符号化模型检验算法由 Legay 和 Wolper 等人给出[28, 17, 58]，并且提供了模型检验工具 T(O)RMC[59]。在该工具中，采用确定的弱 Büchi 自动机作为规约。虽然从严格意义上来说，该种自动机的表达能力弱于 ω-正

规语言，但是采用其作为规约时，能够采用符号化增量迭代的方法高效实现。另外一些针对 C 程序进行模型检验的工具，诸如 SLAM，也采用自动机作为规约，但该自动机是运行于有穷字上的。

PSL 是目前工业界规约语言的标准，它采用扩展的正规表达式作为语法成分。商用的 PSL 的模型检验工具，比如 RULEBASE 等允许使用正规表达式作为公式语法成分。但这些工具目前大部分尚未公开其算法，同时，目前 RULEBASE 并不支持全部 FL 公式的模型检验。在最近的一篇论文 [60] 中，介绍了 RULEBASE 中如何实现针对形如 $\neg r$ 的 FL 公式的过程。该过程将该公式的模型检验转化为形如 $\mathsf{AG}\varphi$ 的 CTL 公式进行检验，其中 φ 是一个表示接收条件的布尔公式。2006 年，Pnueli 和 Zaks[61] 介绍了一种基于 tester 的 PSL 模型检验算法。一个 tester 实际上是一个 ω-自动机的符号化表示。并且，该文还给出了若干针对 abort 算子的重写规则。

此外，文 [62] 和文 [63] 中分别给出了基于规则重写和可达闭包计算的 ω-正规语言的符号化模型检验算法。特别的，文 [62] 还将讨论扩展至树语言。

相对于直接使用 ω-自动机/正规表达式作为规约的时序逻辑而言，各类 ETL[30] 是一种更加具有一般性的规约语言。这类逻辑将自动机、文法或者表达式作为时序连接子引入，并且这些连接子之间允许嵌套。因此，一个 ω-正规表达式可以直接看作没有嵌套的 ETL 公式。在本文第 5 章中，将给出三种使用交错自动机作为时序连接子的 ETL 的符号化模型检验算法。

μ-演算是另外一类广泛应用的时序逻辑。其中，线性时间的 μ-演算具有等价于 ω-正规语言的表达能力。分支时间的 μ-演算是在符号化模型检验中最早被研究的逻辑。该种逻辑的检验过程主要基于 Kleen 迭代，即计算不动点的过程。CTL 的符号化模型检验，也是通过将其嵌入到 μ-演算中进行的。给定迁移系统 M 以及模态 μ-演算公式 φ，检验 $M \models \varphi$ 是否成立的具体复杂度[64, 65]为

$$\mathcal{O}\left(|M| \times |E| \times \left(\frac{2 \times |M| \times |\varphi|}{\mathbf{Ad}(\varphi)}\right)^{\lfloor \frac{\mathbf{Ad}(\varphi)}{2} \rfloor}\right) \tag{1.14}$$

其中，$|E|$ 表示 M 的可达关系数目（即：迁移系统的边数）；$\mathbf{Ad}(\varphi)$ 表示 φ 的交换深度（可以理解为公式中为算子 μ 和 ν 算子带有依赖的交错嵌套深度，具体定义见 [66]）。最近，Schewe 在其博士论文[67] 中，将式 (1.14) 中的指数由 $\lfloor \frac{\mathbf{Ad}(\varphi)}{2} \rfloor$ 降为 $\lfloor \frac{\mathbf{Ad}(\varphi)}{3} \rfloor$。另外，文 [68] 和 [69] 中还给出了两种比较经典的显式 μ-演算模型检验算法。

相比而言，线性 μ-演算中由于缺少分支量词，其符号化模型检验的算法较难开发。在本文第 6 章中，将会给出线性 μ-演算的符号化模型检验算法。

1.4 本文组织结构

本文主要工作可分为两个部分：扩展时序逻辑的公理化部分和符号化模型检验部分。前者包括 3～4 章；后者包括 5～7 章。其中：

- 第 2 章回顾如干基本数学定义，包括：自动机、各类时序逻辑、以及基于 BDD 的符号化模型检验技术。
- 第 3 章给出关于三类扩展时序逻辑— ETL_l、ETL_f、ETL_r 的可靠完备公理系统；并介绍如何使用“实例化”的方法获得其逻辑片断的可靠完备公理系统。
- 第 4 章给出一种关于 μ-演算的新的博弈系统，以及基于博弈的模态/线性 μ-演算的公理系统的完备性证明。
- 第 5 章将基于 tableau 的 LTL 的符号化模型检验算法推广至三类采用交错自动机作为连接子的时序逻辑（ATL_l、ATL_f、ATL_r）以及 PSL 的某个变种（APSL）。由于 APSL 的分支部分对应于 CTL，所以主要讨论其线性部分（AFL）的符号化模型检验算法。
- 第 6 章给出线性 μ-演算的符号化模型检验算法。包括针对一般形式的公式和针对特殊形式公式的检验。
- 第 7 章介绍在开源工具 NuSMV 基础上实现的支持扩展时序逻辑符号化模型检验的工具 ENuSMV，以及采用该工具进行模型检验的若干实验结果。

各个章节之间的关系如下：第 2 章中介绍的自动机、时序逻辑以及符号化模型检验等概念是 3～7 章的基础。第 3 章和第 4 章分别介绍 ETL 和 μ-演算的公理化方法。这两类方法既相联系，又相区别：二者本质上都是基于公式重写进行的，并且都具有“踪迹”的概念；但是这两类踪迹却分别体现了时序连接子和不动点算子的特点。第 6 章符号化模型检验算法中使用的 tableau 实际上是将第 4 章中公式迁移图中非模态节点消除后得到的（即：去迟滞过程）。同时， ETL 公理系统中的 (Expand) 公理是定义公式 tableau 迁移函数的基础。第 6 章中针对 ν-范式线性 μ-演算公式的符号化模型检验算法是由第 4 章中博弈系统扩展得到的。第 7 章中的工具是第 5 章中算法的实现。

第二章　自动机、时序逻辑以及符号化模型检验

本章主要介绍本文用到的数学基础，包括：时序逻辑、自动机和模型检验。主要介绍这些基础概念的形式定义和与之相关的若干基本性质。本章组织如下：

- 2.1 节着重介绍有关各类 ω-自动机的基本概念。介绍诸如字、树、布尔公式等基本定义；ω-自动机的分类方式及其基本运算。
- 2.2 节着重介绍各类经典的时序逻辑的语法、语义。该节从 LTL、CTL 以及 CTL* 开始，进而给出两类时序逻辑扩展：通过添加二阶量词获得的 MSOL、QTL、μ-演算；以及通过添加时序连接子获得的 ETL、PSL 等扩展时序逻辑等。
- 2.3 节回顾基于 BDD（二叉决策图）的符号化模型检验技术的细节。该节首先介绍 BDD 的基本定义以及相关基本操作，而后给出基于 BDD 的 CTL 符号化模型检验的过程，该算法是本文第 5 章和第 6 章的基础。

2.1　ω-自动机

自动机作为一种广泛应用的计算模型，在理论计算机科学领域得到了十分广泛的应用。上世纪 60 年代，Büchi[33]、Trakhtenbrot[70]、McNaughton[71]、Rabin[72] 等人提出了一套完整的无穷语言（包括无穷字语言和无穷树语言）上的有穷自动机框架。这些自动机，最初是用来解决二阶逻辑的判定问题。目前，无穷语言上的自动机主要应用于以下几个方面：

1. **建模**. 自动机本身具有迁移结构，因而可以用来描述变迁系统的迁移行为。比如，在 SPIN[38]中，系统就被转化为一个非确定的 Büchi 自动机。
2. **判定**. 许多时序逻辑的可判定性往往转化为自动机的判空问题— 这时自动机视为时序逻辑公式的一个有穷模型。此外，从逻辑到自动机的转化建立了模型和规约表示之间的一致性。
3. **规约**. 自动机本身也可以作为规约（如 T(O)RMC[59]，Cadence SMV 中），同时，自动机也可以作为规约的连接子。

无穷语言上的自动机分为“**字自动机**” 和“**树自动机**” 两大类。在本文中，采用各类 ω-字自动机作为时序连接子的时序逻辑（ETL）是一个非常重要的研究对象。因此，本节主要介绍各种 ω-字上的自动机。

2.1.1 字、树、布尔公式

定义 2.1.1 (字) 对于任意的(有穷非空)字母表 Σ，其元素称为 Σ 的**字母**。Σ 上的一个**字** w 是由 Σ 的字母构成有穷或者无穷序列。

用 $|w|$ 表示该序列的长度。若 $|w| = \infty$，则称 w 为**无穷字**（或者 ω-字）；否则，称 w 为**有穷字**。分别令 Σ^* 和 Σ^ω 为 Σ 上有穷字和无穷字的集合；并记 $\Sigma^* \cup \Sigma^\omega$ 为 Σ^∞。

对任意的 $i \leq j < |w|$，用 $w(i)$ 表示 w 中的第 i 个字母（i 从 0 开始）；用 $w[i,j]$ 表示 w 的起始于第 i 个字母结束于第 j 个字母的子字。 □

定义 2.1.2 (树) 一棵**树** T 是 $\mathbb{N}^*$ 的前缀封闭子集。即，对于任意的 $x \in \mathbb{N}^*$ 以及 $c \in \mathbb{N}$ 有:

- 若 $x \cdot c \in T$，则 $x \in T$。
- 若 $x \cdot (c+1) \in T$，则 $x \cdot c \in T$。

若 T 是有穷集合，则称 T 为**有穷树**；否则，T 为**无穷树**。

每个 $x \in T$ 称为 T 的**节点**。节点 x 称为 $x \cdot c$ 的**父节点**；$x \cdot c$ 称为 x 的**子节点**。同一父节点的子节点之间互为**兄弟节点**。**祖先节点** 可归纳定义如下:

- 若 x 是 y 的父节点，则 x 是 y 的祖先节点；
- 若 x 是 y 的祖先节点，y 是 z 的祖先节点，则 x 是 z 的祖先节点。

y 是 x 的**后代节点** 当且仅当 x 是 y 的祖先节点。节点 ϵ 称为**根节点**，没有子节点的节点称为**叶节点**。

若节点序列 $\sigma = x_0, x_1, \ldots$ 满足: 每个 x_i 都是 x_{i+1} 的父节点，则称 σ 是 T 中的一条**路径**。称 σ 是一条**极大路径**，如果 σ 中最后一个节点是叶节点，或者 σ 是无穷序列。从根节点 ϵ 到节点 x 的路径长度称为 x 的**深度**，记作 $|x|$。即:

- $|\epsilon| = 0$;
- $|x \cdot c| = |x| + 1$。

若 T 是有穷树，则记 $\max\{|x| \mid x \in T\} + 1$ 为 $|T|$，称为 T 的**树高**。 □

例 2.1.1 采用这种节点编码方式，能够唯一的表示任意节点在树中的位置，显式的表明父子节点关系，以及任一节点在其兄弟节点之间的排序。如图 *2.1* 所示的树中，节点 $0 \cdot 1$ 有三个子节点: $0 \cdot 1 \cdot 0$、$0 \cdot 1 \cdot 1$ 和 $0 \cdot 1 \cdot 2$。节点 $0 \cdot 1$ 的深度为 2，节点 $0 \cdot 1 \cdot 2$ 的深度为 3。节点序列 $0, 0 \cdot 1, 0 \cdot 1 \cdot 1$ 是一条极大路径。 □

定义 2.1.3 (标记树) 任给非空集合 Q，一棵 Q-**标记树** 是一个序偶 $\langle T, \rho \rangle$，其中:

- T 是一棵树;

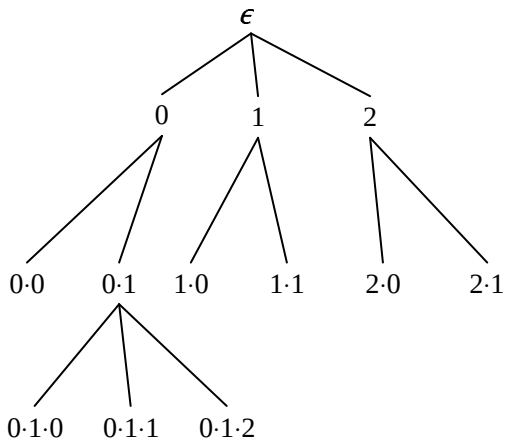


图 2.1 树的例子

- $\rho : T \to Q$, 是一个**标记函数**。

对于 T 中的任意路径 $\sigma = x_1, x_2, \ldots$, 将其对应的标记序列 $\rho(x_1), \rho(x_2), \ldots$ 记为 $\rho(\sigma)$。 □

定义 2.1.4 (布尔公式) 任给非空集合 Q, 其上的**布尔公式**集合 $\mathbf{B}(Q)$ 是满足下列条件的最小集合。

- $true \in \mathbf{B}(Q)$; $false \in \mathbf{B}(Q)$。
- 若 $q \in Q$, 则 $q \in \mathbf{B}(Q)$。
- 若 $\psi \in \mathbf{B}(Q)$, 则 $\neg\psi \in \mathbf{B}(Q)$。
- 若 $\varphi_1, \varphi_2 \in \mathbf{B}(Q)$, 则 $\varphi_1 \wedge \varphi_2 \in \mathbf{B}(Q)$。 □

习惯上, 还引入下列派生的布尔联结词。

$$\varphi_1 \vee \varphi_2 \stackrel{\text{def}}{=} \neg(\neg\varphi_1 \wedge \neg\varphi_2) \tag{2.1}$$

$$\varphi_1 \to \varphi_2 \stackrel{\text{def}}{=} \neg\varphi_1 \vee \varphi_2 \tag{2.2}$$

$$\varphi_1 \leftrightarrow \varphi_2 \stackrel{\text{def}}{=} (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1) \tag{2.3}$$

定义 2.1.5 (布尔指派及布尔公式的真值) 任给非空集合 Q, 其上的一个**真值指派**是一个函数 $e : Q \to \{0, 1\}$。在指派 e 下, 任何布尔公式 $\varphi \in \mathbf{B}(Q)$ 都对应一个布尔真值 $e^*(\varphi) \in \{0, 1\}$, 归纳定义如下。

- $e^*(true) = 1$; $e^*(false) = 0$。
- 若 $q \in Q$, 则 $e^*(q) = e(q)$。
- $e^*(\neg\psi) = \overline{e^*(\psi)}$; $e^*(\varphi_1 \wedge \varphi_2) = e^*(\varphi_1) \times e^*(\varphi_2)$; $e^*(\varphi_1 \vee \varphi_2) = e^*(\varphi_1) + e^*(\varphi_2)$; 。

其中, 布尔运算符 $-$、$\times$、$+$ 定义同常, 即:

1. $-0=1;\ -1=0$。
2. $1\times 1=1;\ 1\times 0=0\times 1=0\times 0=0$。
3. $0+0=0;\ 0+1=1+0=1+1=1$。 □

定义 2.1.6 (布尔公式的满足性) 任给非空集合 Q，布尔公式 $\varphi\in\mathbf{B}(Q)$，以及 $Q'\subseteq Q$，归纳定义**满足关系** $\models$ 如下。

- $Q'\models true;\ Q'\not\models false$。
- 若 $\varphi=q\in Q$，则 $Q'\models\varphi$ 当且仅当 $q\in Q'$。
- 若 $\varphi=\neg\psi$，则 $Q'\models\varphi$ 当且仅当 $Q'\not\models\psi$。
- 若 $\varphi=\varphi_1\wedge\varphi_2$，则 $Q'\models\varphi$ 当且仅当 $Q'\models\varphi_1$ 且 $Q'\models\varphi_2$。

若 $Q'\models\varphi$ 成立，则称集合 Q' **满足** φ。 □

事实上，Q 的任何一个子集 Q' 都能唯一确定一个真值指派 $e_{Q'}$，即：对于任意的 $q\in Q$，$e_{Q'}(q)=1$ 当且仅当 $q\in Q'$。下面的引理非常容易证明：

引理 2.1 对于任意的 $Q'\subseteq Q$ 以及 $\varphi\in\mathbf{B}(Q)$，$Q'\models\varphi$ 当且仅当 $e^*_{Q'}(\varphi)=1$。

因此，Q 的任何一个子集都可以视为 Q 上的一个真值指派。在定义自动机迁移时，需要用到一类特殊的布尔公式，称为“**正布尔公式**”，形式定义如下。

定义 2.1.7 (正布尔公式) 对于任意的非空集合 Q，其上的**正布尔公式**集合 $\mathbf{B}^+(Q)$，是满足下列条件的最小集合。

- $true\in\mathbf{B}^+(Q);\ false\in\mathbf{B}^+(Q)$。
- 若 $q\in Q$，则 $q\in\mathbf{B}^+(Q)$。
- 若 $\varphi_1,\varphi_2\in\mathbf{B}^+(Q)$，则 $\varphi_1\wedge\varphi_2\in\mathbf{B}^+(Q)$，$\varphi_1\vee\varphi_2\in\mathbf{B}^+(Q)$。 □

定义 2.1.8 (正布尔公式的对偶) 若 $\varphi\in\mathbf{B}^+(Q)$，则可以归纳定义其**对偶** $\overline{\varphi}$ 如下。

- $\overline{true}=false;\ \overline{false}=true$。
- 若 $\varphi=q\in Q$，则 $\overline{\varphi}=q$。
- 若 $\varphi=\varphi_1\wedge\varphi_2$，则 $\overline{\varphi}=\overline{\varphi_1}\vee\overline{\varphi_2}$；若 $\varphi=\varphi_1\vee\varphi_2$，则 $\overline{\varphi}=\overline{\varphi_1}\wedge\overline{\varphi_2}$。 □

例 2.1.2 由定义，若 $\varphi\in\mathbf{B}^+(Q)$ 则 $\overline{\varphi}$ 也在 $\mathbf{B}^+(Q)$ 中。设 $Q=\{q_1,q_2,q_3\}$ 以及 Q 上的正布尔公式 $\varphi=q_1\vee(q_2\wedge q_3)$，则 $\overline{\varphi}=q_1\wedge(q_2\vee q_3)$。 □

此外，正布尔公式具有单调性，即：若 $\varphi\in\mathbf{B}^+(Q)$，$Q'\subseteq Q$，且 $Q'\models\varphi$，则对于任意的 $Q'\subseteq Q''\subseteq Q$ 必然有 $Q''\models\varphi$。关于正布尔公式，容易证明下面的引理：

引理 2.2 设 Q 为非空集合，$\varphi\in\mathbf{B}^+(Q)$ 以及 $Q'\subseteq Q$，那么 $Q'\models\overline{\varphi}$ 当且仅当对任意的 $Q''\subseteq Q$ 有：若 $Q''\models\varphi$ 则 $Q'\cap Q''\neq\emptyset$。

2.1.2 ω-自动机的定义及其分类

本节介绍各类 ω-（字）自动机的定义、接收问题以及分类。

定义 2.1.9 (ω-自动机) 一个 **ω-自动机** 是一个五元组 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$，其中:

- Σ 是一个有穷**字母表**。
- Q 是一个有穷**状态集**。
- $\delta : Q \times \Sigma \to \mathbf{B}^+(Q)$。
- $q \in Q$，是一个**初始状态**。
- Ω 是一个**接收条件**，在稍后定义。 □

设自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$，则对于任意的 $q' \in Q$，用 $\mathcal{A}^{q'}$ 表示自动机 $\langle \Sigma, Q, \delta, q', \Omega \rangle$。于是，$\mathcal{A}$ 和 $\mathcal{A}^q$ 所指相同。

定义 2.1.10 (确定、非确定、全局、交错自动机) 设 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$ 是一个 ω-自动机，则可以按照迁移函数的特征对其进行分类:

- 对迁移函数的形式未加任何限制时，该自动机的迁移类型为**交错**（*Alternating*）的。
- 若对于任意的 $q' \in Q$ 以及 $a \in \Sigma$，表达式 $\delta(q, a)$ 中不出现 $\wedge$，则称该自动机的迁移类型是**非确定**（*Nondeterministic*）的。
- 若对于任意的 $q' \in Q$ 以及 $a \in \Sigma$，表达式 $\delta(q, a)$ 中不出现 $\vee$，则称该自动机的迁移类型是**全局**（*Universal*）的。
- 若该自动机的迁移类型既是非确定的又是全局的，则称其迁移类型是**确定**（*Deterministic*）的。 □

设 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Sigma \rangle$。若其迁移类型是非确定的或者全局的，则可以将迁移函数 δ 写成 $Q \times \Sigma \to 2^Q$ 的形式。若其迁移类型是确定的，则直接可以将 δ 写成 $Q \times \Sigma \to Q$ 的形式。

比如: $\delta(q_1, a_1) = \{q_1, q_2\}$，在非确定自动机中表示 $\delta(q_1, a_1) = q_1 \vee q_2$，而在全局自动机中表示 $\delta(q_1, a_1) = q_1 \wedge q_2$。

下面定义 ω-字自动机的**接收问题**。

定义 2.1.11 (自动机的运行及接收运行)
给定 ω-自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$ 以及无穷字 $w \in \Sigma^\omega$, $\mathcal{A}$ 在 w 上的一个**运行** 是一棵 Q-标记树 $\langle T, \rho \rangle$，其中:

- $\rho(\epsilon) = q$。
- 对每个 $x \in T$，若 x 不是叶节点，则集合 $\{\rho(x \cdot c) \mid c \in \mathbb{N}, x \cdot c \in T\} \models$

$\delta(\rho(x), w(|x|))$。

称 T 中的路径 σ 是一条**自然接收路径**，如果 σ 是一条终结于某叶节点 x 的极大有穷路径，且满足 $\delta(\rho(x), w(|x|)) = true$。这时，$x$ 称为**接收叶节点**。

对任意路径 σ，用 $\mathbf{Inf}(\rho(\sigma))$ 表示集合 $\{q \in Q \mid \sigma$ 中有无穷多个 x使得 $\rho(x) = q\}$；同时，用 $\mathbf{Occ}(\rho(\sigma))$ 表示集合 $\{q \in Q \mid$ 存在 σ 中的某个节点 x 使得 $\rho(x) = q\}$。常用的接收条件 Ω 有如下几种:

finite：接收条件 Ω 是 Q 的一个子集 F（称作**接收状态集**）。路径 σ 满足 Ω 当且仅当 σ 有穷，并且 σ 终结于某叶节点 x，且满足 $\rho(x) \in F$。

terminating：路径 σ 满足 Ω 当且仅当 σ 是自然接收路径。

looping：路径 σ 满足 Ω 当且仅当 σ 是无穷序列。

repeating：（或***Büchi***[33]）接收条件是 Q 的一个子集 F（称作**接收状态集**）。路径 σ 满足 Ω 当且仅当 $\mathbf{Inf}(\rho(\sigma)) \cap F \neq \emptyset$。

co-repeating：（或***co-Büchi***）接收条件是 Q 的一个子集 F。路径 σ 满足 Ω 当且仅当 $\mathbf{Inf}(\rho(\sigma)) \cap F = \emptyset$。

Rabin：接收条件 Ω 是一个子集对集合 $\{(E_1, F_1), \ldots, (E_m, F_m)\}$，路径 σ 满足 Ω 当且仅当存在某个 $1 \leq i \leq m$，使得 $\mathbf{Inf}(\rho(\sigma)) \cap E_i = \emptyset$ 且 $\mathbf{Inf}(\rho(\sigma)) \cap F_i \neq \emptyset$。

Streett：接收条件 Ω 是一个子集对集合 $\{(E_1, F_1), \ldots, (E_m, F_m)\}$，路径 σ 满足 Ω 当且仅当对每个 $1 \leq i \leq m$ 都有: 或者 $\mathbf{Inf}(\rho(\sigma)) \cap E_i \neq \emptyset$ 或者 $\mathbf{Inf}(\rho(\sigma)) \cap F_i = \emptyset$。

parity：接收条件 Ω 是一个从 Q 到 $\mathbb{N}$ 的部分函数。路径 σ 满足 Ω 当且仅当 $\max\{\Omega(q) \mid q \in \mathbf{Inf}(\rho(\sigma))\}$ 是偶数。

Muller：接收条件 Ω 是由 Q 的若干子集构成的集合 $\{F_1, \ldots, F_m\}$。路径 σ 满足 Ω 当且仅当存在 i 使得 $\mathbf{Inf}(\rho(\sigma)) = F_i$。

称 $\langle T, \rho \rangle$ 是 $\mathcal{A}$ 的一个**可接收运行**，如果对 T 中的任何一条极大路径 σ 而言，或者 σ 是自然接收路径，或者 σ 满足 $\mathcal{A}$ 的接收条件。 □

例 2.1.3 设自动机 $\mathcal{A} = \langle \{a\}, \{q_1, q_2\}, \delta, q_1, \Omega \rangle$，其中，$\delta(q_1, a) = q_1 \vee q_2$，$\delta(q_2, a) = q_1 \wedge q_2$，$\Omega$ 是 *looping* 接收条件。则图 *2.2* 中的 $\{q_1, q_2\}$-标记树就是 $\mathcal{A}$ 在无穷字 a^ω 上的一个可接收运行。 □

定义 2.1.12 (自动机识别的语言) 若存在自动机 $\mathcal{A}$ 在无穷字 w 上的一个可接收运行，则称 w 被 $\mathcal{A}$ **接收**。用 $\mathbf{L}(\mathcal{A})$ 表示 $\mathcal{A}$ 所接收的无穷字集合，称为 $\mathcal{A}$ 所识别的**语言**。 □

事实上，除 finite 接收条件之外，任何一种接收条件 Ω 都可以表示为一个布

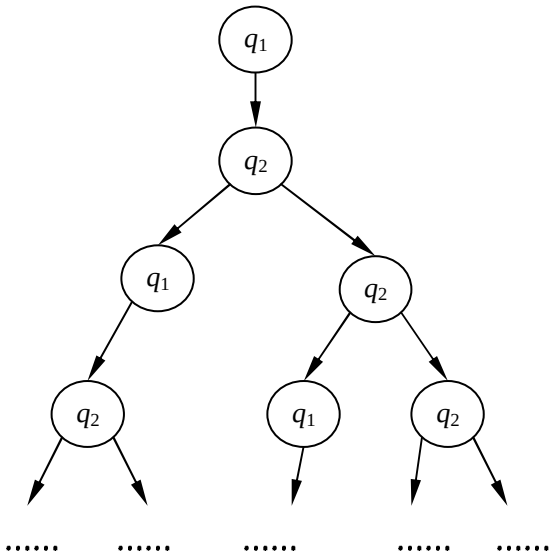


图 2.2 自动机运行示例

尔公式 φ_Ω（见文 [73]）。它满足：运行 $\langle T, \rho\rangle$ 中的极大路径 σ 满足 Ω，当且仅当 $\mathbf{Inf}(\rho(\sigma)) \models \varphi_\Omega$。具体的表示方式如表 2.1 所示。

表 2.1 ω-自动机接收条件（不含 finite）的布尔编码

接收条件类型	Ω	φ_Ω
looping	-	$\bigvee_{q\in Q} q$
terminating	-	$\bigwedge_{q\in Q} \neg q$
repeating	$F \subseteq Q$	$\bigvee_{q\in F} q$
co-Büchi	$F \subseteq Q$	$\bigwedge_{q\in F} \neg q$
Rabin	$\{(E_1, F_1), \ldots, (E_m, F_m)\}$	$\bigvee_{1\le i\le m}(\bigvee_{q\in F_i} q \wedge \bigwedge_{q\in E_i} \neg q)$
Streett	$\{(E_1, F_1), \ldots, (E_m, F_m)\}$	$\bigwedge_{1\le i\le m}(\bigwedge_{q\in F_i} \neg q \vee \bigvee_{q\in E_i} q)$
parity	$\Omega : Q \rightsquigarrow \mathbb{N}$	$\bigvee_{2i\le \mathrm{ran}(\Omega)}(\bigvee_{\Omega(q)=2i} q \wedge \bigwedge_{\Omega(q)>2i} \neg q)$
Muller	$\{F_1, \ldots, F_m\}$ （$F_i \subseteq Q$）	$\bigvee_{1\le i\le m}(\bigwedge_{q\in F_i} \wedge \bigwedge_{q\notin F_i} \neg q)$

采用这种接收条件编码方式，可以非常方便的对自动机进行求补操作。对任意 ω-自动机，可以如下定义其对偶自动机。

定义 2.1.13 (对偶自动机) 给定自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega\rangle$，且其接收条件不是 *finite*，则定义其**对偶自动机** $\overline{\mathcal{A}} = \langle \Sigma, Q, \overline{\delta}, q, \widetilde{\Omega}\rangle$。其中，对任意的 $q \in Q$ 以及 $a \in \Sigma$，$\overline{\delta}(q, a)$ 是 $\delta(q, a)$ 的对偶，即 $\overline{\delta(q, a)}$；$\widetilde{\Omega}$ 是 Ω 的取反，即：$\varphi_{\widetilde{\Omega}} = \neg\varphi_\Omega$。

□

显然，$\overline{\overline{\mathcal{A}}} = \mathcal{A}$。并且由上述定义知：$\mathbf{Inf}(\rho(\sigma))$ 满足 Ω 当且仅当 $\mathbf{Inf}(\rho(\sigma))$ 违反 $\widetilde{\Omega}$。同时，由表 2.1 知：当 Ω 分别是 looping、terminating、Büchi、co-Büchi、Rabin、Streett、parity 和 Muller 接收条件时，$\widetilde{\Omega}$ 则分别是 terminating、looping、co-Büchi、Büchi、Streett、Rabin、parity 和 Muller 接收条件。值得注意的是：parity 接收条件和 Muller 接收条件关于取反操作是封闭的：因为 parity 接收条件 Ω 的取反可以看作是 parity 接收条件 $\Omega+1$ — 即在每个 Ω 有定义的 q 处 $\Omega+1$ 也有定义，并且 $(\Omega+1)(q) = \Omega(q)+1$；Muller 接收条件 $\mathcal{F}$ 的取反可以看作是 Muller 接收条件 $2^Q \backslash \mathcal{F}$。

定理 2.3 ([74]) *设自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$，且其接收条件不是 finite，则 $\mathbf{L}(\overline{\mathcal{A}}) = \Sigma^\omega \backslash \mathbf{L}(\mathcal{A})$。*

而 finite 接收条件的自动机实质上可以等价的转化成 terminating 自动机：给定 finite 自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, F \rangle$，其中，$F = \{q_1, \ldots, q_m\} \subseteq Q$，则有 terminating 自动机 $\mathcal{A}' = \langle \Sigma, Q \backslash F \cup \{q'\}, \delta', q', - \rangle$ 使得 $\mathbf{L}(\mathcal{A}) = \mathbf{L}(\mathcal{A}')$。其中，$q' \notin Q$；且对于任意的 $q'' \in Q \backslash F \cup \{q'\}$ 以及 $a \in \Sigma$ 有：

$$\delta(q'', a) = \begin{cases} \delta(q'', a)^{q_1,\ldots,q_m}_{true,\ldots,true} & , q'' \in Q \backslash F \\ \delta(q, a)^{q_1,\ldots,q_m}_{true,\ldots,true} & , q'' = q', q \notin F \\ true & , q'' = q', q \in F \end{cases} 。$$

其中，$\varphi^{q_1,\ldots,q_m}_{true,\ldots,true}$ 表示将 φ 中的每个 q_i 的出现替换为 $true$ 得到的正布尔公式。

为了便于描述自动机的类型，本文采用一个三字母的缩写对其进行刻画：第一个字母是“A”（Alternating）、“N”（Nondeterministic）、“U”（Universal）或“D”（Deterministic）中的一个，指明其迁移类型；第二个字母是“L”、“T”、“F”、“B”、“C”、“R”、“S”、“P” 或“M” 中的一个，指明其接收条件；第三个字母，在本章中，仅限于“W”（Word），指明其接收语言的类型（与“W” 并列的还有“T”（Tree）和“G”（Graph），关于树自动机的例子可见第 4 章）。

例 2.1.4 *比如：ARW、NBW、USW、DFW 分别表示交错 Rabin 自动机、非确定 Büchi 自动机、全局 Streett 自动机、确定 finite 自动机。* □

有别于有穷字上的自动机，w-自动机的表达能力对迁移类型和接收条件非常敏感。比如：NBW 和 DBW 的表达能力不相同，前者严格强于后者— 考虑字母表 $\{a_1, a_2\}$ 上的无穷字 $(a_1|a_2)^*; a_2^\omega$，它能够被某 NBW 识别，却不能被任何 DBW 识别[75]。本节所讨论的各类自动机的表达能力之间的关系如图 2.3 所示。

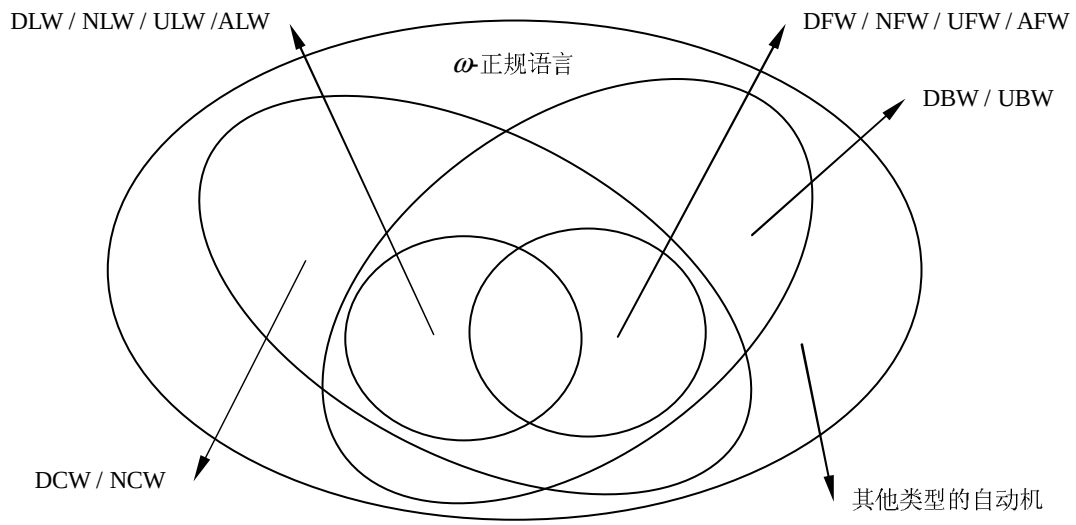


图 2.3 各类 ω-自动机表达能力之间的关系

2.2 时序逻辑

本节主要介绍各种时序逻辑。对一种逻辑而言，除给出必需的语法成份外，一般还需要至少一种模型以及关于该种模型的语义。比如，一阶逻辑（First Order Logic），是较为经典的逻辑之一，它是在命题逻辑的基础上引入量词和项获得的。传统一阶逻辑的语义定义在 Tarski 结构上，它具有足够强的表达能力来描述所需的规约性质。但是，由于一阶逻辑本身的不可判定性，在实际的模型检验中，并未得到广泛应用。

在实际应用中，需要一些定义在有"结构"的模型上的可判定的逻辑。**时序逻辑**在上世纪 70 年代提出 [76, 77, 15]，它主要用以描述系统的时序行为，即："事件"发生的（相对或绝对）顺序。可以按照下列方式对命题时序逻辑进行分类[78]。

1. **离散** v.s. **连续**。即：其语义结构中的状态是离散的还是连续的。
2. **线性** v.s. **分支**。即：其语义结构是定义于线性序列上还是定义在树型结构上。
3. **二值** v.s. **多值**。即：公式的语义取值是否仅限于真/假二值。

在本文涉及的逻辑中，主要关心离散时间的二值逻辑。

2.2.1 线性结构及分支结构

设 AP 是原子命题的集合，令 $\overline{AP} = \{\neg p \mid p \in AP\}$。现在定义描述时序逻辑语义的线性结构和分支结构（在本文中，自然数均从 0 开始）。

定义 2.2.1 (线性结构) 一个（无穷）**线性结构**是一个函数 $\pi : \mathbb{N} \to 2^{AP}$，它为每个时刻 i 赋予一个真值指派 $\pi(i)$。同时，一个线性结构也可以看作是以 2^{AP} 为字母表

的 ω-字，$\pi(i)$ 是其第 i 个字母。另外，对任意的 $j \in \mathbb{N}$，定义函数 $\pi^j : \mathbb{N} \to 2^{AP}$ 为 $\pi^j(i) = \pi(i+j)$。 □

定义 2.2.2 (分支结构) 一个（无穷）**分支结构**是一个无穷 2^{AP}-标记树 $\langle T, \rho \rangle$。分支结构也叫做**计算树**。 □

在本文中，对每个"计算树" $\langle T, \rho \rangle$ 而言，要求 T 中没有叶节点（即：T 中的每条极大路径都是无穷路径）。加入这样的限制是为了某些定义在分支结构上的时序逻辑（如模态 μ-演算）中的某些公理（如 $\Box\varphi \to \Diamond\varphi$）的可靠性。

2.2.2 LTL、CTL、CTL*

LTL [79, 15, 21] 是在命题逻辑的基础上通过添加时序算子 X 和 U 所获得的时序逻辑，其语法、语义定义如下。

定义 2.2.3 (LTL 语法) ***LTL* 合式公式**可以归纳定义如下:

- 命题常元 $true$ 是 LTL 公式。
- 若原子命题 $p \in AP$，则 p 是 LTL 公式。
- 若 ψ 是 LTL 公式，则 $\neg\psi$ 是 LTL 公式。
- 若 φ_1 和 φ_2 是 LTL 公式，则 $\varphi_1 \wedge \varphi_2$ 是 LTL 公式。
- 若 ψ 是 LTL 公式，则 $\mathsf{X}\psi$ 是 LTL 公式。
- 若 φ_1 和 φ_2 是 LTL 公式，则 $\varphi_1 \mathsf{U} \varphi_2$ 是 LTL 公式。 □

在本文提及的任意一种时序逻辑中，都存在如下缩写

$$false \stackrel{\text{def}}{=} \neg true \tag{2.4}$$

同时，为使用方便，在 LTL 中还会定义下列派生时序连接子：

$$\mathsf{F}\psi \stackrel{\text{def}}{=} true\mathsf{U}\psi \tag{2.5}$$

$$\mathsf{G}\psi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\psi \tag{2.6}$$

$$\varphi_1\mathsf{R}\varphi_2 \stackrel{\text{def}}{=} \neg(\neg\varphi_1\mathsf{U}\neg\varphi_2) \tag{2.7}$$

定义 2.2.4 (LTL 语义) 给定线性结构 π，LTL 公式 φ，以及位置 $i \in \mathbb{N}$，可以归纳定义**满足关系** $\models$ 如下:

- 若 $\varphi = true$，则 $\pi, i \models \varphi$ 一定成立。
- 若 $\varphi = p \in AP$，则 $\pi, i \models \varphi$ 当且仅当 $p \in \pi(i)$。
- 若 $\varphi = \neg\psi$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \models \varphi_1$ 且 $\pi, i \models \varphi_2$。

- 若 $\varphi = \mathsf{X}\psi$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i+1 \models \psi$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$，则 $\pi, i \models \varphi$ 当且仅当存在 $k \geq i$，使得 $\pi, k \models \varphi_2$，并且对于任意的 $i \leq j < k$ 都有 $\pi, j \models \varphi_1$。

特别的，当 $i = 0$ 时，直接将 $\pi, i \models \varphi$ 记作 $\pi \models \varphi$。 □

容易验证：在 LTL 中有

$$\varphi_1 \mathsf{R} \varphi_2 \leftrightarrow (\mathsf{G}\varphi_2) \vee (\varphi_2 \mathsf{U} (\varphi_1 \wedge \varphi_2)) \tag{2.8}$$

成立。

CTL 由 Clarke 和 Emerson[19] 等人提出。它是在命题逻辑的基础上添加时序算子 EX、EU 和 AU 等获得的时序逻辑。

定义 2.2.5 (CTL 语法) *CTL* **合式公式**可归纳定义如下：

- 命题常元 $true$ 是 *CTL* 公式。
- 若原子命题 $p \in AP$，则 p 是 *CTL* 公式。
- 若 ψ 是 *CTL* 公式，则 $\neg\psi$ 是 *CTL* 公式。
- 若 φ_1 和 φ_2 是 *CTL* 公式，则 $\varphi_1 \wedge \varphi_2$ 是 *CTL* 公式。
- 若 ψ 是 *CTL* 公式，则 $\mathsf{EX}\psi$ 是 *CTL* 公式。
- 若 φ_1 和 φ_2 是 *CTL* 公式，则 $\mathsf{A}(\varphi_1 \mathsf{U} \varphi_2)$ 和 $\mathsf{E}(\varphi_1 \mathsf{U} \varphi_2)$ 都是 *CTL* 公式。 □

同样，为方便起见，在 CTL 中引入下列派生时序连接子。

$$\mathsf{AF}\psi \stackrel{\text{def}}{=} \mathsf{A}(true \mathsf{U} \psi) \tag{2.9}$$

$$\mathsf{EF}\psi \stackrel{\text{def}}{=} \mathsf{E}(true \mathsf{U} \psi) \tag{2.10}$$

$$\mathsf{AX}\psi \stackrel{\text{def}}{=} \neg\mathsf{EX}\neg\psi \tag{2.11}$$

$$\mathsf{AG}\psi \stackrel{\text{def}}{=} \neg\mathsf{EF}\neg\psi \tag{2.12}$$

$$\mathsf{EG}\psi \stackrel{\text{def}}{=} \neg\mathsf{AF}\neg\psi \tag{2.13}$$

$$\mathsf{A}(\varphi_1 \mathsf{R} \varphi_2) \stackrel{\text{def}}{=} \neg\mathsf{E}(\neg\varphi_1 \mathsf{U} \neg\varphi_2) \tag{2.14}$$

$$\mathsf{E}(\varphi_1 \mathsf{R} \varphi_2) \stackrel{\text{def}}{=} \neg\mathsf{A}(\neg\varphi_1 \mathsf{U} \neg\varphi_2) \tag{2.15}$$

CTL 语义定义在分支结构（或者说计算树）上，可严格的描述如下。

定义 2.2.6 (CTL 语义) 给定分支结构 $\langle T, \rho \rangle$，*CTL* 公式 φ 以及节点 $x \in T$，可以归纳定义**满足关系** $\models$ 如下：

- 若 $\varphi = true$，则一定有 $T, \rho \models \varphi$ 成立。
- $T, \rho, x \models p$ 当且仅当 $p \in \rho(x)$。

- $T,\rho,x \models \neg\psi$ 当且仅当 $T,\rho,x \not\models \psi$。
- $T,\rho,x \models \varphi_1 \wedge \varphi_2$ 当且仅当 $T,\rho,x \models \varphi_1$ 且 $T,\rho,x \models \varphi_2$。
- $T,\rho,x \models \mathsf{EX}\psi$ 当且仅当存在 x 在 T 中的子节点 $x\cdot c$ 使得 $T,\rho,x\cdot c \models \psi$。
- $T,\rho,x \models \mathsf{A}(\varphi_1\mathsf{U}\varphi_2)$ 当且仅当对 T 中任意的路径 σ，若 $\sigma(0)=x$，则存在 $i\in\mathbb{N}$，使得 $T,\rho,\sigma(i) \models \varphi_2$，且对于任意的 $0\le j<i$，都有 $T,\rho,\sigma(j)\models\varphi_1$。
- $T,\rho,x \models \mathsf{E}(\varphi_1\mathsf{U}\varphi_2)$ 当且仅当存在 T 中的路径 σ，满足 $\sigma(0)=x$，且存在 $i\in\mathbb{N}$ 使得 $T,\rho,\sigma(i) \models \varphi_2$，同时对于任意的 $0\le j<i$ 都有 $T,\rho,\sigma(j)\models\varphi_1$。

特别的，当 $x=\epsilon$ 时，$T,\rho,x\models\varphi$ 也记作 $T,\rho\models\varphi$。 □

CTL* 是 LTL 和 CTL 的公共扩展[80]。它将公式分为“**路径公式**”和“**状态公式**”两类，具体定义如下。

定义 2.2.7 (CTL* 语法)

*CTL** 中的**状态公式**归纳定义如下:

- 命题常元 $true$ 是 *CTL** 状态公式。
- 若 $p\in AP$，则 p 是 *CTL** 状态公式。
- 若 ψ 是 *CTL** 状态公式，则 $\neg\psi$ 是 *CTL** 状态公式。
- 若 φ_1 和 φ_2 都是 *CTL** 状态公式，则 $\varphi_1\wedge\varphi_2$ 是 *CTL** 状态公式。
- 若 ψ 是 *CTL** 路径公式，则 $\mathsf{A}\psi$ 是 *CTL** 状态公式。

*CTL** 中的**路径公式**归纳定义如下:

- 若 φ 是 *CTL** 状态公式，则 φ 是 *CTL** 路径公式。
- 若 ψ 是 *CTL** 路径公式，则 $\neg\psi$ 是 *CTL** 路径公式。
- 若 φ_1 和 φ_2 都是 *CTL** 路径公式，则 $\varphi_1\wedge\varphi_2$ 也是 *CTL** 路径公式。
- 若 ψ 是 *CTL** 路径公式，则 $\mathsf{X}\psi$ 是 *CTL** 路径公式。
- 若 φ_1 和 φ_2 都是 *CTL** 路径公式，则 $\varphi_1\mathsf{U}\varphi_2$ 是 *CTL** 路径公式。 □

对 CTL* 公式，除可以引入在 LTL 中定义的派生连接子（见公式 (2.5) ～(2.7)）之外，还可以定义派生路径量词 E 如下。

$$\mathsf{E}\psi \stackrel{\text{def}}{=} \neg\mathsf{A}\neg\psi \tag{2.16}$$

CTL* 的语义也定义在分支结构上，但对状态公式和路径公式需要区别对待。

定义 2.2.8 (CTL* 语义)

给定计算树 $\langle T,\rho\rangle$ 以及 T 中的节点 x，对于*CTL** 状态公式，可以归纳定义**满足关系** $\models$ 如下。

- 若 $\varphi=true$，则一定有 $T,\rho,x\models\varphi$ 成立。

- 若 $\varphi = p \in AP$，则 $T, \rho, x \models \varphi$ 当且仅当 $p \in \rho(x)$。
- 若 $\varphi = \neg\psi$，则 $T, \rho, x \models \varphi$ 当且仅当 $T, \rho, x \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $T, \rho, x \models \varphi$ 当且仅当 $T, \rho, x \models \varphi_1$ 且 $T, \rho, x \models \varphi_2$。
- 若 $\varphi = \mathsf{A}\psi$，则 $T, \rho, x \models \varphi$ 当且仅当对任意的 T 中的路径 σ 以及 $i \in \mathbb{N}$ 都有：若 $\sigma(i) = x$ 则 $T, \rho, \sigma, i \models \psi$。

同样，当 $x = \epsilon$ 时，直接记作 $T, \rho \models \varphi$。

对于 T 中任意的路径 σ 以及 $i \in \mathbb{N}$，当 φ 是路径公式时，也可归纳定义**满足**关系 $\models$ 如下。

- 若 φ 是状态公式，则 $T, \rho, \sigma, i \models \varphi$ 当且仅当 $T, \rho, \sigma(i) \models \varphi$。
- 若 $\varphi = \neg\psi$，则 $T, \rho, \sigma, i \models \varphi$ 当且仅当 $T, \rho, \sigma, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $T, \rho, \sigma, i \models \varphi$ 当且仅当 $T, \rho, \sigma, i \models \varphi_1$ 且 $T, \rho, \sigma, i \models \varphi_2$。
- 若 $\varphi = \mathsf{X}\psi$，则 $T, \rho, \sigma, i \models \varphi$ 当且仅当 $T, \rho, \sigma, i+1 \models \psi$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$，则 $T, \rho, \sigma, i \models \varphi$ 当且仅当存在 $k \geq i$，使得 $T, \rho, \sigma, k \models \varphi_2$ 且对任意的 $i \leq j < k$，都有 $T, \rho, \sigma, j \models \varphi_1$。

当 $i = 0$ 时，直接记作 $T, \rho, \sigma \models \varphi$。 □

定义 2.2.9 (否定范式、ACTL、ECTL、ACTL*、ECTL*)

利用派生的连接子和路径量词，可以将任一 *CTL** 公式中的 "¬" 内移，直至原子命题之前。这样获得的公式称为原公式的**否定范式**。对于某 *CTL* 公式，若该公式的否定范式中的路径量词全为 A（*resp.* E），则称该公式为 *ACTL*（*resp. ECTL*）公式。同样，对于某 *CTL** 公式，若该公式的否定范式中的路径量词 A（*resp.* E），则称该公式为 *ACTL**（*resp. ECTL**）公式。 □

由于 LTL 和 CTL 的语义分别定义在线性结构和分支结构上，二者并不能直接进行比较。在定义了 CTL* 后，就可以在该逻辑内比较二者的表达能力了。因为，任意的 CTL 公式都是 CTL* 公式；而对于 LTL 公式 φ 而言，可以将其看作是 CTL* 公式 $\mathsf{A}\varphi$。

定义 2.2.10 给定 *LTL* 公式 φ 和 *CTL** 公式 ψ。称 φ 和 ψ 是**等价**的当且仅当：对于任意的分支结构 $\langle T, \rho \rangle$ 有 $(T, \rho \models \mathsf{A}\varphi) \Leftrightarrow (T, \rho \models \psi)$。 □

文 [81, 47, 80] 等比较了 LTL 和 CTL 的表达能力，指出了二者在表达能力上是不可比较的。即：既存在只能被 LTL 描述的时序性质，也存在只能被 CTL 描述的性质。

例 2.2.1 (LTL、CTL、CTL* 表达能力比较)

- *LTL* 公式 $\mathsf{GF}p$ 等价于 *CTL* 公式 $\mathsf{AGAF}p$。

- *LTL* 公式 $\mathsf{F}(p \wedge \mathsf{X}p)$ 无法被任何 *CTL* 公式表达。
- *CTL* 公式 $\mathsf{AGEF}p$ 无法被任何 *LTL* 公式表达。
- *CTL** 公式 $\mathsf{A}(\mathsf{F}(p_1 \wedge \mathsf{X}p_1)) \vee \mathsf{AGEF}p_2$ 既不能被任何 *LTL* 公式表达，也不能被任何 *CTL* 公式表达。

三者之间的关系如图 *2.4* 所示。 □

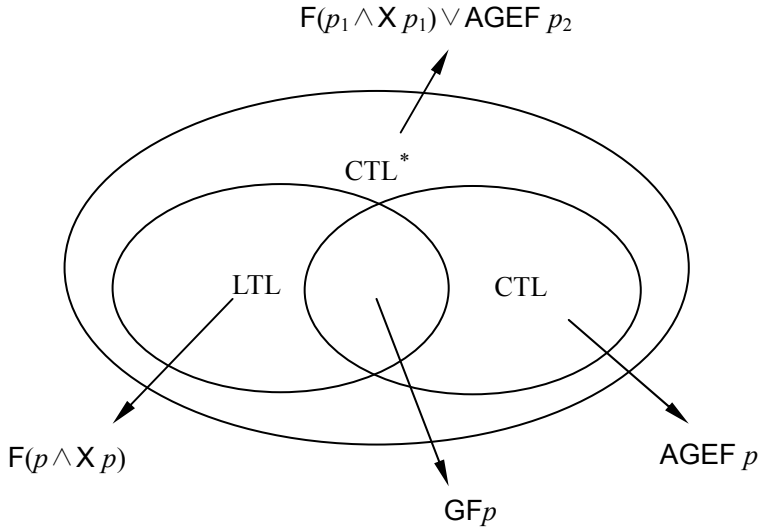


图 2.4 LTL、CTL、CTL* 表达能力关系

关于 LTL、CTL 之间的等价转化问题，有以下结论。

定理 2.4 ([80]) *若 CTL* 公式 φ 具有与之等价的 LTL 公式 ψ，则 ψ 必然是（或者等价于）由 φ 删除所有的路径量词所得到的公式。*

定理 2.5 ([82]) *对于给定的 LTL 公式，是否存在与之等价的 CTL 公式是可判定的。*

一个有趣的问题是："如果 *LTL* 公式 φ 存在与之等价的 *CTL* 公式 ψ，那么 ψ 是否一定是 *ACTL* 公式"？直到 2008 年，这个问题才由 Bojańczyk 所解决。他给出了的如下结论：

定理 2.6 ([83]) *对于给定的 LTL 公式，是否存在与之等价的 ACTL 公式是可判定的。并且，存在 LTL 公式 φ，它能够写成等价的 CTL 公式，但是不存在与之等价的 ACTL 公式。*

例 2.2.2 *在文 [83] 中，Bojańczyk 证明了：时序性质 $(a;b)^*;a;(a;b)^*c^\omega$ 既能够被 LTL 描述，也能够被 CTL 描述，但不能被 ACTL 描述。* □

Vardi 在文 [22] 中深入的比较了 LTL 和 CTL 的优缺点— 指出，LTL 无论是在易用性、简洁性还是在兼容性等方面都略胜一筹。因此，LTL 在模型检验中得到了较为广泛的使用。

然而，在工业应用中，制约 LTL 应用的一个瓶颈在于其表达能力— Wolper 指出：类似于 "p 在每个偶数时刻都成立" 等一类 ω-正规性质都不能被 LTL 表达[23]。后来，人们发现 LTL 的表达能力恰好等价于 star-free 的 ω-正规语言。在下面两个小节中，将会介绍几种扩展的时序逻辑。其中，语义定义在线性结构上的，表达能力全部等价于 ω-正规语言。

2.2.3 MSOL、QTL、μ-演算

MSOL（Monadic Second Order Logics）是一种较早的专用于描述时序的二阶逻辑。它是由 Prior[84, 85] 在经典二阶逻辑的基础上限制其语法成分和语义结构得到的。严格来讲，MSOL 既能描述分支语义，也能描述线性语义（称为 S1S）。本节主要针对线性语义介绍该种逻辑。在此之前，首先介绍 MSOL 的一阶逻辑片断 MFOL（Monadic First Order Logic）。

MFOL 的语言成分包括一个**一阶个体变元集合** $FV = \{t_0, t_1, \ldots\}$，但没有引入函词；语言中仅包括一个二元谓词 $<$，此外，还将原来命题逻辑中的每个原子命题 p 视为一个一元谓词（这也是 Monadic 的含义所在）。

定义 2.2.11 (MFOL语法) *MFOL* 的**合式公式**可归纳定义如下:

- 命题常元 $true$ 是 *MFOL* 公式。
- 若 $p \in AP$, $t \in FV$, 则 $p(t)$ 是 *MFOL* 公式。
- 若 $t_1, t_2 \in FV$, 则 $t_1 < t_2$ 是 *MFOL* 公式。
- 若 ψ、φ_1、φ_2 都是 *MFOL* 公式, 则 $\neg\psi$、$\varphi_1 \wedge \varphi_2$ 都是 *MFOL* 公式。
- 若 ψ 是 *MFOL* 公式, $t \in FV$, 则 $\exists t.\psi$ 是 *MFOL* 公式。 □

MFOL 中常用的派生谓词、联结词以及量词如下。

$$t_1 = t_2 \stackrel{\text{def}}{=} \neg(t_1 < t_2) \wedge \neg(t_2 < t_1) \tag{2.17}$$

$$t_1 \prec t_2 \stackrel{\text{def}}{=} (t_1 < t_2) \wedge \neg\exists t.((t_1 < t) \wedge (t < t_2)) \tag{2.18}$$

$$\forall t.\psi \stackrel{\text{def}}{=} \neg\exists t.\neg\psi \tag{2.19}$$

对 MFOL 而言，**约束/自由变元**、变元的**约束/自由出现** 以及**句子** 的定义如经典一阶逻辑。

定义 2.2.12 (MFOL语义) 一个**个体变元指派**是一个函数 $e : FV \to \mathbb{N}$。对于某个体变元 $t \in FV$ 以及 $i \in \mathbb{N}$，则 $e[t/i]$ 也是一个个体变元指派，它在 t 处的函数值为 i，在其余处的函数值与 e 的函数值相同。

给定线性结构 π，个体变元指派 e，以及 *MFOL* 公式 φ，可以归纳定义 φ 在个体变元指派 e 下的**满足关系** $\models$ 如下。

- 若 $\varphi = true$，则一定有 $\pi, e \models \varphi$ 成立。
- 若 $\varphi = p(t)$，则 $\pi, e \models \varphi$ 当且仅当 $p \in \pi(e(t))$。
- 若 $\varphi = (t_1 < t_2)$，则 $\pi, e \models \varphi$ 当且仅当 $e(t_1)$ 小于 $e(t_2)$。
- 若 $\varphi = \neg\psi$，则 $\pi, e \models \varphi$ 当且仅当 $\pi, e \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\pi, e \models \varphi$ 当且仅当 $\pi, e \models \varphi_1$ 且 $\pi, e \models \varphi_2$。
- 若 $\varphi = \exists t.\psi$，则 $\pi, e \models \varphi$ 当且仅当存在 $i \in \mathbb{N}$，使得 $\pi, e[t/i] \models \psi$。

特别的，当 φ 是句子时，直接将 $\pi, e \models \varphi$ 写作 $\pi \models \varphi$。 □

MFOL 是一种非常重要的一阶逻辑片断，Kamp 和 Gabbay 证明了它在表达能力上恰好等价于 LTL。该结论可表述如下：

定理 2.7 (LTL 和（单自由变元）MFOL 的等价性 [79, 86])

对于任意的 *LTL* 公式 φ，存在只有一个自由个体变元 t 的 *MFOL* 公式 ψ，使得对于任意的线性结构 π 以及个体变元指派 e 有：$\pi, e(t) \models \varphi$ 当且仅当 $\pi, e \models \psi$。

反之，对于仅含一个自由个体变元 t 的*MFOL* 公式ψ，存在一个 *LTL* 公式 φ，使得对于任意的线性结构 π 以及个体变元指派 e 有 $\pi, e(t) \models \varphi$ 当且仅当 $\pi, e \models \psi$。

MSOL 是在 MFOL 的基础上扩充了一个**集合变元集** $SV = \{X_0, X_1, \ldots, \}$ 得到的二阶逻辑片断。MSOL 的合式公式可归纳定义如下。

定义 2.2.13 (MSOL 语法)

- 所有的 *MFOL* 公式都是 *MSOL* 公式。
- 若 $X \in SV$，$t \in FV$，则 $X(t)$ 是 *MSOL* 公式。
- 若 $X \in SV$，ψ 是 *MSOL* 公式，则 $\exists X.\psi$ 是 *MSOL* 公式。 □

类似于 MFOL 中的情况，也可以引入如下的派生量词定义。

$$\forall X.\psi \stackrel{\text{def}}{=} \neg\exists X.\neg\psi \tag{2.20}$$

定义 2.2.14 (MSOL（线性）语义) 一个**集合变元指派**是一个函数 $E : SV \to 2^{\mathbb{N}}$。对于某集合变元 $X \in SV$ 以及 $N \subseteq \mathbb{N}$，则 $E[X/N]$ 也是一个集合变元指派，它在 X 处的函数值为 N，在其余处的函数值与 E 的函数值相同。

给定线性结构 π，个体变元指派 e，集合变元指派 E 以及 *MSOL* 公式 φ，可以归

纳定义 φ 在 e 和 E 下的**满足关系** $\models$ 如下。

- 若 φ 是纯 *MFOL* 公式，则 $\pi, e, E \models \varphi$ 当且仅当 $\pi, e \models \varphi$。
- 若 $\varphi = X(t)$，则 $\pi, e, E \models \varphi$ 当且仅当 $e(t) \in E(X)$。
- 若 $\varphi = \exists X.\psi$，则 $\pi, e, E \models \varphi$ 当且仅当存在$N \subseteq \mathbb{N}$，使得 $\pi, e, E[X/N] \models \psi$。

同样，若 *MSOL* 公式 φ 是句子，则直接将 $\pi, e, E \models \varphi$ 写作 $\pi \models \varphi$。 □

下面的定理说明了在线性结构上 MSOL 和 ω-正规语言的等价性。

定理 2.8 ([33]) 设$\Pi \subseteq (2^{AP})^{\omega}$，则 Π 是 ω- 正规语言当且仅当存在 *MSOL* 句子 φ，使得 $\pi \in \Pi$ 当且仅当 $\pi \models \varphi$。

另外一种基于二阶量词构建的时序逻辑是 QTL (Quantified Temporal Logic)（见 [50, 24, 25]）。该种逻辑同时使用时序算子和二阶谓词。

定义 2.2.15 (QTL 语法) 仍令 SV 为集合变元集，则 QTL 的**合式公式**集合可以归纳定义如下。

- 命题常元 $true$ 是 QTL 公式。
- 若 $p \in AP$，则 p 是 QTL 公式。
- 若 $X \in SV$，则 X 是 QTL 公式。
- 若 ψ、φ_1、φ_2 都是 QTL 公式，则 $\neg\psi$ 和 $\varphi_1 \wedge \varphi_2$ 都是 QTL 公式。
- 若 φ_1 和 φ_2 都是 QTL 公式，则 $\varphi_1 \mathsf{U} \varphi_2$ 是 QTL 公式。
- 若 ψ 是 QTL 公式，$X \in SV$，则 $\exists X.\psi$ 是 QTL 公式。 □

在 QTL 中，派生时序连接子 F、G、R 以及派生量词 $\exists$ 的定义分别如公式 (2.5)、(2.6)、
(2.7) 以及 (2.20) 所示。

定义 2.2.16 (QTL 语义) 给定线性结构 π，集合变元指派 $E : SV \to 2^{\mathbb{N}}$，位置 $i \in \mathbb{N}$，以及 *MSOL* 公式 φ，可以归纳定义**满足关系** $\models$ 如下。

- 若 $\varphi = true$，则 $\pi, E, i \models \varphi$ 一定成立。
- 若 $\varphi = p \in AP$，则 $\pi, E, i \models \varphi$ 当且仅当 $p \in \pi(i)$。
- 若 $\varphi = X \in SV$，则 $\pi, E, i \models \varphi$ 当且仅当 $i \in E(X)$。
- 若 $\varphi = \neg\psi$，则 $\pi, E, i \models \varphi$ 当且仅当 $\pi, E, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\pi, E, i \models \varphi$ 当且仅当 $\pi, E, i \models \varphi_1$ 且 $\pi, E, i \models \varphi_2$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$，则 $\pi, E, i \models \varphi$ 当且仅当存在 $k \geq i$，使得 $\pi, E, k \models \varphi_2$，且对于任意的 $i \leq j < k$ 有 $\pi, E, j \models \varphi_1$ 成立。
- 若 $\varphi = \exists X.\psi$，则 $\pi, E, i \models \varphi$ 当且仅当存在 $N \subseteq \mathbb{N}$ 使得 $\pi, E[X/N], i \models \psi$。

特别的，当 φ 是句子时，表达式中的 E 可以省略；当 $i = 0$ 时，表达式中的 i 也可

以省略。 □

文 [78] 证明了 QTL 和线性结构下 MSOL 的等价性。指出：二者都等价于以 2^{AP} 为字母表的 ω-正规语言。标准的 QTL 中使用的（唯一）时序算子是 U，Wu（吴志林）在文 [87] 中给出了采用其他时序算子 QTL 变种及其表达能力的详细讨论。

QTL 和 MSOL 的可满足性都不是初等可判定的（分别见文 [33] 及文 [25]）。即：对于长度为 n 的 QTL/MSOL 公式 φ，其可满足性是可判定的，但其判定的复杂度为$2^{\left.2^{\cdot^{\cdot^{\cdot^{2^n}}}}\right\}k}$。其中，指数深度 k 没有固定的上界。

作为比较，Emerson 和 Clarke[19]，Pratt[88]，Kozen[55] 等人通过的向动态逻辑（Dynamic Logic）中添加不动点算子而获得逻辑是初等可判定的[89]。这种逻辑称为 μ-演算，它构建于原子命题集合 AP 以及公式变元集合 VAR，其**合式公式定义**如下。

定义 2.2.17 (模态μ-演算语法)

- 命题常元 $true$ 是模态 μ-演算公式。
- 若 $p \in AP$，则 p 是模态 μ-演算公式。
- 若 $X \in VAR$，则 X 是模态 μ-演算公式。
- 若 ψ 是模态 μ-演算公式，则 $\neg\psi$ 也是模态 μ-演算公式。
- 若 φ_1、φ_2 都是模态 μ-演算公式，则 $\varphi_1 \wedge \varphi_2$ 也是模态 μ-演算公式。
- 若 ψ 是模态 μ-演算公式，则 $\Box\psi$ 也是模态 μ-演算公式。
- 若 ψ 是模态 μ-演算公式，且 X 在 ψ 中的所有出现都是**正出现**（即：出现在偶数个“¬”的辖域内），则 $\mu X.\psi$ 也是模态 μ-演算公式。 □

习惯上，模态 μ-演算中还引入如下的派生算子。

$$\Diamond\psi \stackrel{\text{def}}{=} \neg\Box\neg\psi \tag{2.21}$$

$$\nu X.\psi \stackrel{\text{def}}{=} \neg\mu X.\neg\psi^X_{\neg X} \tag{2.22}$$

其中，$\psi^X_{\neg X}$ 表示将 ψ 中所有自由出现的 X 替换为 $\neg X$ 得到的公式。

定义 2.2.18 (模态 μ-演算语义) 给定分支结构 $\langle T, \rho\rangle$，公式变元指派 $E : VAR \to 2^T$，以及模态 μ-演算公式 φ，则可以定义**满足集函数** $[\![\bullet]\!]_{\langle T,\rho\rangle}E$。该函数将每个公式映射为 T 的一个子集，归纳定义如下。

- 若 $\varphi = true$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = T$。
- 若 $\varphi = p \in AP$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = \{x \in T \mid p \in \rho(x)\}$。
- 若 $\varphi = X \in VAR$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = E(X)$。
- 若 $\varphi = \neg\psi$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = T\backslash([\![\psi]\!]_{\langle T,\rho\rangle}E)$。

- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = ([\![\varphi_1]\!]_{\langle T,\rho\rangle}E) \cap ([\![\varphi_2]\!]_{\langle T,\rho\rangle}E)$。
- 若 $\varphi = \Box\psi$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = \{x \in T \mid$ 对 x 在 T 中的每个子节点 $x{\cdot}c$，都有 $x{\cdot}c \in [\![\psi]\!]_{\langle T,\rho\rangle}E\}$。
- 若 $\varphi = \mu X.\psi$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = \bigcap\{T' \subseteq T \mid [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T'] \subseteq T'\}$。

为方便起见，将 $x \in [\![\varphi]\!]_{\langle T,\rho\rangle}E$ 简记为 $T, \rho, E, x \models \varphi$。并且，当 φ 是句子时，可将变元指派 E 省略；当 $x = \epsilon$ 时，也可将 x 省略。 □

CTL 可以嵌入至模态 μ-演算中。即：对于任何一个长度为 n 的 CTL 公式 φ，存在一个长度为 $\mathcal{O}(n)$ 的模态 μ-演算句子 $\hat{\varphi}$，使得对于任意的分支结构 $\langle T, \rho\rangle$ 以及 $x \in T$，有：$T, \rho, x \models \varphi$ 当且仅当 $T, \rho, x \models \hat{\varphi}$。具体而言，$\hat{\varphi}$ 可归纳构造如下：

- 若 $\varphi = true$，则 $\hat{\varphi} = true$。
- 若 $\varphi = p \in AP$，则 $\hat{\varphi} = p$。
- 若 $\varphi = \neg\psi$，则 $\hat{\varphi} = \neg\hat{\psi}$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\hat{\varphi} = \hat{\varphi_1} \wedge \hat{\varphi_2}$。
- 若 $\varphi = \mathsf{EX}\psi$，则 $\hat{\varphi} = \Diamond\hat{\psi}$。
- 若 $\varphi = \mathsf{A}(\varphi_1\mathsf{U}\varphi_2)$，则 $\hat{\varphi} = \mu X.(\hat{\varphi_2} \vee (\hat{\varphi_1} \wedge \Box X))$。
- 若 $\varphi = \mathsf{E}(\varphi_1\mathsf{U}\varphi_2)$，则 $\hat{\varphi} = \mu X.(\hat{\varphi_2} \vee (\hat{\varphi_1} \wedge \Diamond X))$。

正是由于存在上述变换过程，实际中的 CTL 检验会往往采用基于不动点的迭代计算进行（见 2.3 节）。

模态 μ-演算可以派生出各种变种。比如：在模态词上标注动作（Actions），添加“标称词”（Nominals），添加双向模态词，量度模态词（Graded Modalities）等。Vardi 证明了：只要不是将这些语法成分全部引入，模态 μ-演算仍是初等可判定的，并且是 EXPTIME-complete 的（见文 [89, 90]）。

Banieqbal 和 Barringer 提出了线性时间的 μ-演算[29]。同分支时间的版本相比，它只是将模态算子 $\Box$ 和 $\Diamond$ 合并成为一个算子 $\bigcirc$。其**合式公式**集归纳定义如下：

定义 2.2.19 (线性μ-演算语法)

- 命题常元 $true$ 是线性 μ-演算公式。
- 若 $p \in AP$，则 p 是线性 μ-演算公式。
- 若 $X \in VAR$，则 X 是线性 μ-演算公式。
- 若 ψ 是线性 μ-演算公式，则 $\neg\psi$ 也是线性 μ-演算公式。
- 若 φ_1、φ_2 都是线性 μ-演算公式，则 $\varphi_1 \wedge \varphi_2$ 也是线性 μ-演算公式。
- 若 ψ 是线性 μ-演算公式，则 $\bigcirc\psi$ 也是线性 μ-演算公式。
- 若 ψ 是模态 μ-演算公式，且 X 在 ψ 中的所有出现都是正出现，则 $\mu X.\psi$ 也

是线性 μ-演算公式。 □

线性 μ-演算的语义定义在线性结构上。

定义 2.2.20 (线性 μ-演算语义) 给定线性结构 π，变元指派 $E : VAR \to 2^{\mathbb{N}}$，以及线性 μ-演算公式 φ，则可以定义**满足集函数** $[\![\bullet]\!]_\pi E$。该函数将每个公式映射为 $\mathbb{N}$ 的一个子集，归纳定义如下。

- 若 $\varphi = true$，则 $[\![\varphi]\!]_\pi E = \mathbb{N}$。
- 若 $\varphi = p \in AP$，则 $[\![\varphi]\!]_\pi E = \{i \mid p \in \pi(i)\}$。
- 若 $\varphi = X \in VAR$，则 $[\![\varphi]\!]_\pi E = E(X)$。
- 若 $\varphi = \neg\psi$，则 $[\![\varphi]\!]_\pi E = \mathbb{N} \backslash ([\![\psi]\!]_\pi E)$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $[\![\varphi]\!]_\pi E = ([\![\varphi_1]\!]_\pi E) \cap ([\![\varphi_2]\!]_\pi E)$。
- 若 $\varphi = \bigcirc\psi$，则 $[\![\varphi]\!]_\pi E = \{i+1 \mid i \in [\![\psi]\!]_\pi E\}$。
- 若 $\varphi = \mu X.\psi$，则 $[\![\varphi]\!]_\pi E = \bigcap\{N \subseteq \mathbb{N} \mid [\![\psi]\!]_\pi E[X/N] \subseteq N\}$。

为方便起见，将 $i \in [\![\varphi]\!]_\pi E$ 简记为 $\pi, E, i \models \varphi$。并且，当 φ 是句子时，可将变元指派 E 省略；当 $i = 0$ 时，也可将 i 省略。 □

类似的，也可以将 LTL 嵌入到线性 μ-演算中。对于给定的 LTL 公式 φ，可以按下列过程归纳的得到等价的线性 μ-演算公式 $\hat{\varphi}$：

- 当 φ 为原子公式，或形如 $\neg\psi$、$\varphi_1 \wedge \varphi_2$ 时，$\hat{\varphi}$ 的定义同前。
- 若 $\varphi = \mathsf{X}\psi$，则 $\hat{\varphi} = \bigcirc\hat{\psi}$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$，则 $\hat{\varphi} = \mu X.(\hat{\varphi_2} \vee (\hat{\varphi_1} \wedge \bigcirc X))$。

线性 μ-演算的表达能力也等价于ω-正规语言，其判定的复杂度是 PSPACE-complete 的（见文 [91]）。

2.2.4 从 ETL 到 PSL

在 2.2.3 节中介绍的线性时序逻辑的表达能力均等价于 ω-正规语言。这些逻辑中，算子（包括连接子、模态/时序算子、量词、不动点算子）的数目都是有限的，语法成分相对简洁。但使用 MSOL、QTL 或者 μ-演算描述时序性质时，往往不太直观。比如，下面的模态 μ-演算公式

$$\nu X.(\Box X \wedge \mu Y.(p \vee \Diamond Y))$$

就等价于 CTL 公式 $\mathsf{AGEF}p$。但是，这样的公式并不是非常容易被理解。并且，当公式的不动点算子嵌套深度超过 2 时，公式几乎不具有可读性。

在线性框架下，另外一种获得等价于 ω-正规语言的方式是向时序逻辑中添加无穷多的时序算子。这类逻辑以 Wolper 提出的 ETL 为代表[23]。最初的 ETL 构建于

ω-正规文法，后来，发展为采用各类 ω-自动机作为时序连接子（见 [24, 25, 30, 92] 等）。本文中所采用的 ETL 也是以自动机为连接子的时序逻辑。

定义 2.2.21 (ETL 语法) *ETL* 的**合式公式** 归纳定义如下:

- 命题常元 $true$ 是 *ETL* 公式。
- 若 $p \in AP$, 则 p 是 *ETL* 公式。
- 若 ψ 是 *ETL* 公式, 则 $\neg\psi$ 是 *ETL* 公式。
- 若 ψ 是 *ETL* 公式, 则 $\bigcirc\psi$ 是 *ETL* 公式。
- 若 φ_1, φ_2 是 *ETL* 公式, 则 $\varphi_1 \wedge \varphi_2$ 是 *ETL* 公式。
- 若 $\mathcal{A}$ 是以 $\{a_1, \ldots, a_n\}$ 为字母表的 ω-自动机, $\varphi_1, \ldots, \varphi_n$ 是 *ETL* 公式, 则 $\mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 是 *ETL* 公式。 □

在 ETL 的原始语法中，并没有显式声明连接子“$\bigcirc$”，它可以通过自动机连接子定义。但在本文中，无论是在 ETL 的公理化还是符号化模型算法中，该算子都具有十分重要的意义，因此，将其显式添加至语法中。

在 ETL 中，派生连接子 $\vee$ 的定义同前。形如 $\mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 的公式称为**自动机公式**。同自动机中对应定义类似，若自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi)$，则将公式 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 记作 φ^q。于是，若 q 是 $\mathcal{A}$ 的初始状态，则 φ 与 φ^q 所指相同。

定义 2.2.22 (ETL 语义) 给定线性结构 π, 位置 $i \in \mathbb{N}$, 以及 *ETL* 公式 φ, 可以归纳定义**满足关系** $\models$ 如下。

- 若 $\varphi = true$, 则 $\pi, i \models \varphi$ 一定成立。
- 若 $\varphi = p \in AP$, 则 $\pi, i \models \varphi$ 当且仅当 $p \in \pi(i)$。
- 若 $\varphi = \neg\psi$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \models \varphi_1$ 且 $\pi, i \models \varphi_2$。
- 若 $\varphi = \bigcirc\psi$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i+1 \models \psi$。
- 若 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$, 则 $\pi, i \models \varphi$ 当且仅当 φ 存在一个在 π 上开始于 i 的可接收运行（见下面的定义 *2.2.23*）。 □

在上述定义中，用到了“**自动机公式运行**”的概念，定义如下。

定义 2.2.23 (自动机公式运行) 设 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$ 是一个 ω-自动机, 其中 $\Sigma = \{a_1, \ldots, a_n\}$。设 $\varphi_1, \ldots, \varphi_n$ 都是 *ETL* 公式, $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$。则 φ 在线性结构 π 上开始于位置 i 的一个**运行**是一个 Q-标记树 $\langle T, \rho \rangle$。其中:

- $\rho(\epsilon) = q$。
- 对每个节点 $x \in T$, 存在某个 $k \in \{1, \ldots, n\}$, 使得 $\pi, i + |x| \models \varphi_k$, 且集合 $\{\rho(x \cdot c) \mid c \in \mathbb{N},\ x \cdot c \in T\}$ 满足 $\delta(\rho(x), a_k)$。

若某节点 $x \in T$ 满足: 存在 $k \in \{1,\ldots,n\}$, 使得 $\pi, i+|x| \models \varphi_k$, 且 $\delta(\rho(x), a_k) = true$, 则 x 就可以是 T 的叶节点。这样的叶节点称为**接收叶节点**。终止于接收叶节点的路径称为**自然接收路径**。

称 $\langle T, \rho\rangle$ 是 φ 在 π 上开始于位置 i 的一个**可接收运行**, 当且仅当对于 T 中的任意一条极大路径 σ 有:

1. 或者 σ 为自然接收路径。
2. 或者 $\rho(\sigma)$ 满足 $\mathcal{A}$ 的接收条件 Ω (同定义*2.1.9*)。 □

根据采用的自动机连接子的类型, 可以将 ETL 进行分类。传统上, 主要关心下列几类 ETL:

- 采用 NFW、NLW、NBW 作为时序连接子的 ETL 分别称为 ETL_f、ETL_l、ETL_r。
- 采用 AFW、ALW、ABW 作为时序连接子的 ETL 分别称为 ATL_f、ATL_l、ATL_r。

由定义 2.2.23, 设自动机公式 $\varphi = \mathcal{A}(\varphi_1,\ldots,\varphi_n)$, 以及$\langle T, \rho\rangle$ 是 φ 在线性结构 π 起始于位置 i 的一个可接收运行, 则对于 T 的任意一条极大路径 σ:

- 若 φ 是 ETL_f 或 ATL_f 公式, 且 $\mathcal{A}$ 的接收状态集为 F, 则 σ 或者是 T 中的自然接收路径, 或者终止于某节点 x, 且 $\rho(x) \in F$。— 因而, T 一定是有穷树。
- 若 φ 是 ETL_l 或 ATL_l 公式, 则 σ 或者是 T 中的自然接收路径, 或者是一条无穷路径。
- 若 φ 是 ETL_r 或 ATL_r 公式, 且 $\mathcal{A}$ 的接收状态集为 F, 则 σ 或者是 T 中的自然接收路径, 或者 $\mathbf{inf}(\rho(\sigma)) \cap F \neq \emptyset$。

同"可接收运行"对偶的一个概念是"**拒绝例证**", 形式定义如下。

定义 2.2.24 (自动机公式拒绝例证) 设 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega\rangle$ 是一个 ω-自动机, 其中 $\Sigma = \{a_1,\ldots,a_n\}$。设 $\varphi_1,\ldots,\varphi_n$ 都是 *ETL* 公式, $\varphi = \mathcal{A}(\varphi_1,\ldots,\varphi_n)$。则 φ 在线性结构 π 上开始于位置 i 的一个**拒绝例证** 是一个 Q-标记树 $\langle T, \rho\rangle$。其中:

- $\rho(\epsilon) = q$。
- 对 T 中的任一节点 x, 以及 $k \in \{1,\ldots,n\}$, 若 $\pi, i+|x| \models \varphi_k$, 则集合 $\{\rho(x\cdot c) \mid c \in \mathbf{N}, x\cdot c \in T\}$ 满足 $\overline{\delta(\rho(x), a_k)}$。
- 对 T 中的任意极大路径 σ 而言, $\rho(\sigma)$ 都违反 $\mathcal{A}$ 的接收条件 Ω。

特别的, 若 $\mathcal{A}$ 是 *finite* 接收条件的自动机, 则还应增加限制: 对任意的 $x \in T$, $\rho(x) \notin F$。这里, F 是 $\mathcal{A}$ 的接收状态集。 □

由上述定义, 设 $\langle T, \rho\rangle$ 是 φ 在 π 上开始于 i 的拒绝例证。若 x 是 T 的一个叶节点, 则一定有: 对于任意的 $1 \le k \le n$, 若 $\pi, i+|x| \models \varphi_k$, 则 $\delta(\rho(x), a_k) = false$。

定理 2.9 给定 *ETL* 自动机公式 $\varphi = \mathcal{A}(\varphi_1,\ldots,\varphi_n)$, 线性结构 π, 以及位置 i。则

$\pi, i \models \neg\varphi$ 当且仅当 φ 存在一个在 π 上开始于 i 的拒绝例证。

该定理在 ETL 公理化以及 ATL_l、ATL_r 的符号化模型检验中起着非常重要的作用。它可以看作是定理 2.3 在 ETL 上的扩展（见文 [74]）。

随后，ETL 产生了许多变种，比如：Intel 的 ForSpec[93]，IBM 的 Sugar[94] 等等。这些规约语言在工业界得到了广泛的应用。这些语言在 2004 年由 Accellera 统一，形成了新的规约语言 PSL [31]，并成为工业标准（IEEE-1850）。

PSL 语言既包含线性成分— FL（Foundamental Language）公式，也包含分支成分— OBE（Optimal Branching Extensions）公式。其中，OBE 与 CTL 严格等同。FL 公式的构建基于 SERE（Sequential Extended Regular Expresssions）和若干内置时序连接子。SERE 是标准正规表达式的扩展，它的语法归纳定义如下。

定义 2.2.25 (SERE 语法)

- $0[*]$ 是 *SERE* 表达式。
- 若 $b \in \mathbf{B}(AP)$，则 b 是 *SERE* 表达式。
- 若 r 是 *SERE* 公式，则 r^* 是 *SERE* 表达式。
- 若 r_1，r_2 是*SERE* 表达式，则 $r_1|r_2$，$r_1\&r_2$，$r_1 : r_2$，$r_1; r_2$ 都是 *SERE* 表达式。

□

注意串拼接符号 $\cdot$ 与 ; 的区别：前者应用于串操作中，而后者专用于正则表达式中。

定义 2.2.26 (SERE 派生语言集) 每个*SERE* 表达式 r 都会派生出一个语言集 $\mathbf{L}(r) \subseteq (2^{AP})^*$，归纳定义如下:

- $\mathbf{L}(0[*]) = \{\epsilon\}$;
- 若 $r = b \in \mathbf{B}(AP)$，则 $\mathbf{L}(r) = \{w \mid |w| = 1, w(0) \models b\}$。
- 若 $r = (r')^*$，则 $\mathbf{L}(r) = \{w \mid$ 存在 $m \geq 0$ 使得 $w = w_1 \cdot \ldots \cdot w_m$ 其中 $w_i \in \mathbf{L}(r')\}$。
- 若 $r = r_1|r_2$，则 $\mathbf{L}(r) = \mathbf{L}(r_1) \cup \mathbf{L}(r_2)$。
- 若 $r = r_1\&r_2$，则 $\mathbf{L}(r) = \mathbf{L}(r_1) \cap \mathbf{L}(r_2)$。
- 若 $r = r_1 : r_2$，则 $\mathbf{L}(r) = \{w_1 \cdot l \cdot w_2 \mid l \in 2^{AP}, w_1 \cdot l \in \mathbf{L}(r_1), l \cdot w_2 \in \mathbf{L}(r_2)\}$。
- 若 $r = r_1; r_2$，则 $\mathbf{L}(r) = \{w_1 \cdot w_2 \mid w_1 \in \mathbf{L}(r_1), w_2 \in \mathbf{L}(r_2)\}$。 □

FL 公式是构建于 SERE 表达式以及时序算子 X、U、abort 以及 trigger 的时序逻辑，其语法定义如下。

定义 2.2.27 (FL 语法) *FL* 的合式公式归纳定义如下:

- 若 $\psi = b \in \mathbf{B}(AP)$，则 ψ 是 *FL* 公式。

- 若 ψ 是 FL 公式，则 $\neg\psi$ 是 FL 公式。
- 若 φ_1，φ_2 都是 FL 公式，则 $\varphi_1 \wedge \varphi_2$ 是 FL 公式。
- 若 ψ 是 FL 公式，则 $\mathsf{X}\psi$ 是 FL 公式。
- 若 φ_1，φ_2 是 FL 公式，则 $\varphi_1 \mathsf{U} \varphi_2$ 是 FL 公式。
- 若 ψ 是 FL 公式，$b \in \mathbf{B}(AP)$，则 ψ abort b 是 FL 公式。
- 若 ψ 是 FL 公式，r 是 $SERE$ 表达式，则 r trigger ψ 是 FL 公式。 □

FL 的语义既可以定义在无穷线性结构上，也可以定义在有穷线性结构上。在本文中，只讨论其在无穷线性结构上的语义。

定义 2.2.28 (FL 语义) 给定线性结构 π，位置 $i \in \mathbb{N}$，以及 FL 公式 φ，则可以定义**满足关系** $\models$ 如下。

- 若 $\varphi = b \in \mathbf{B}(AP)$，则 $\pi, i \models \varphi$ 当且仅当 $\pi(i)$ 满足 b。
- 若 $\varphi = \neg\psi$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \models \varphi_1$ 且 $\pi, i \models \varphi_2$。
- 若 $\varphi = \mathsf{X}\psi$，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i+1 \models \psi$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$，则 $\pi, i \models \varphi$ 当且仅当存在 $k \geq i$ 使得 $\pi, k \models \varphi_2$，且对于任意的 $i \leq j < k$，都有 $\pi, j \models \varphi_1$。
- 若 $\varphi = \psi$ abort b，则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \models \psi$ 或者存在 $k \geq i$ 以及线性结构 π' 使得 $\pi[0,k] = \pi'[0,k]$，$\pi(k)$ 满足 b，且 $\pi', i \models \psi$。
- 若 $\varphi = r$ trigger ψ，则 $\pi, i \models \varphi$ 当且仅当对于任意的 $j \geq i$，若 $\pi[i,j] \in \mathbf{L}(r)$，则 $\pi, j \models \psi$。

特别的，当 $i = 0$ 时，直接将 $\pi, i \models \varphi$ 写作 $\pi \models \varphi$。 □

注：本文中 FL 的语法定义与 [31] 中不尽相同，由于在模型检验中只关心 FL 在无穷线性结构上的语义，这里只选取了部分语法成分作为基本算子。在第 5 章中将会证明：只使用这些语法成分，就可以表达全部的 ω-正规性质。此外，原始定义中的其他语法成分可以派生得到，比如：

$$\varphi_1 \mathsf{R} \varphi_2 \stackrel{\text{def}}{=} \neg(\neg\varphi_1 \mathsf{U} \neg\varphi_2) \tag{2.23}$$

$$\psi \text{ monitor } b \stackrel{\text{def}}{=} \neg(\psi \text{ abort } \neg b) \tag{2.24}$$

$$r \text{ leads } \psi \stackrel{\text{def}}{=} \neg(r \text{ trigger} \neg\psi) \tag{2.25}$$

$$r! \stackrel{\text{def}}{=} r \text{ leads } true \tag{2.26}$$

上述派生算子/连接词将在 5.6 节中使用。

2.2.5 公式的可满足性及有效性

前面三小节中定义了各类常用的时序逻辑。现在给出其“可满足性”（Satisfiability）与“有效性”（Validity）的统一定义。

定义 2.2.29 (时序逻辑公式的可满足性)

对于线性框架下的时序逻辑公式 φ，则称 φ 是**可满足的**，是指:

- 当 φ 是 *LTL*、*ETL* 或 *PSL* 公式时，当且仅当存在线性结构 π 使得 $\pi \models \varphi$。
- 当 φ 是 *QTL* 或者线性 μ-演算公式时，当且仅当存在线性结构 π 以及（集合或公式）变元指派 E 使得 $\pi, E \models \varphi$。
- 当 φ 是 *MSOL* 公式时，当且仅当存在线性结构 π，个体变元指派 e，以及集合变元指派 E 使得 $\pi, e, E \models \varphi$。

对于分支框架下的时序逻辑公式 φ，则称 φ 是**可满足的**，是指:

- 当 φ 是 *CTL* 或 *CTL** 公式时，当且仅当存在分支结构 $\langle T, \rho\rangle$，使得 $T, \rho \models \varphi$。
- 当 φ 是模态 μ-演算公式时，当且仅当存在分支结构 $\langle T, \rho\rangle$ 以及变元指派 E，使得 $T, \rho, E \models \varphi$。

特别的，当 φ 是句子时，线性时序逻辑的可满足性可以统一定义为: φ 是**可满足的**且仅当存在线性结构 π，使得 $\pi \models \varphi$。分支时序逻辑的可满足性可以统一定义为: φ 是**可满足的** 且仅当存在分支结构 $\langle T, \rho\rangle$，使得 $T, \rho \models \varphi$。 □

定义 2.2.30 (时序逻辑公式的有效性)

对于线性框架下的时序逻辑公式 φ，则称 φ 是**有效的**，是指:

- 当 φ 是 *LTL*、*ETL* 或 *PSL* 公式时，当且仅当对任意线性结构 π，都有 $\pi \models \varphi$。
- 当 φ 是 *QTL* 或者线性 μ-演算公式时，当且仅当对任意线性结构 π 以及（集合或公式）变元指派 E 都有 $\pi, E \models \varphi$。
- 当 φ 是 *MSOL* 公式时，当且仅当对任意线性结构 π，个体变元指派 e，以及集合变元指派 E 都有 $\pi, e, E \models \varphi$。

对于分支框架下的时序逻辑公式 φ，则称 φ 是**有效的**,（记作 $\models \varphi$）是指:

- 当 φ 是 *CTL* 或 *CTL** 公式时，当且仅当对任意的分支结构 $\langle T, \rho\rangle$，都有 $T, \rho \models \varphi$。
- 当 φ 是模态 μ-演算公式时，当且仅当对任意分支结构 $\langle T, \rho\rangle$ 以及变元指派 E，都有 $T, \rho, E \models \varphi$。

特别的，当 φ 是句子时，线性时序逻辑的有效性可以统一定义为: φ 是**有效的** 且仅当对任意的线性结构 π，都有 $\pi \models \varphi$。这种情况下，分支时序逻辑的有效性可以

统一定义为: φ 是**有效的** 且仅当对任意分支结构 $\langle T, \rho\rangle$，都有 $T, \rho \models \varphi$。 □

显而易见，φ 是可满足的，当且仅当 $\neg\varphi$ 不是有效的；或者，φ 是有效的，当且仅当 $\neg\varphi$ 是不可满足的。

2.3 符号化模型检验

在本节，将会对"模型检验" 问题给出形式化定义，同时介绍基于 BDD 的符号化模型检验技术的若干细节。

2.3.1 模型检验问题

定义 2.3.1 (迁移系统) 一个（有穷）**迁移系统**（或称 *Kripke* 结构）是一个序偶 $M = \langle S, \Delta, I, \lambda\rangle$，其中:

- S 是一个有穷的**状态集**。
- $\Delta \subseteq S \times S$，是一组**迁移关系**（在本文中，除非特别声明，均要求该迁移关系是**连续的**，即: 对于任意的 $s \in S$，存在 $s' \in S$ 使得 $(s, s') \in \Delta$）。
- $I \subseteq S$，是一组**初始状态**集合。
- $\lambda : S \to 2^{AP}$，是一个**命题标记函数**。 □

从线性和分支角度，可以将每个迁移系统进行展开，从而可以派生出线性结构和分支结构。形式定义如下。

定义 2.3.2 (展开迹及派生线性结构) 设迁移系统 $M = \langle S, \Delta, I, \lambda\rangle$，称 $\hat{\pi} : \mathbb{N} \to S$ 是 M 的一个**展开迹**，如果 $\hat{\pi}(0) \in I$，且对每个 $i \in \mathbb{N}$ 有 $(\hat{\pi}(i), \hat{\pi}(i+1)) \in \Delta$。称 $\pi : \mathbb{N} \to 2^{AP}$ 是 $\hat{\pi}$ 对应的**派生线性结构**，如果 $\pi = \lambda \circ \hat{\pi}$。即: 对每个 $i \in \mathbb{N}$ 有 $\pi(i) = \lambda(\hat{\pi}(i))$。记 $\mathbf{L}(M)$ 为 M 所有展开迹对应的派生线性结构的集合。 □

定义 2.3.3 (展开树及派生分支结构) 设迁移系统 $M = \langle S, \Delta, I, \lambda\rangle$，称 S-标记树 $\langle T, \hat{\rho}\rangle$ 是 M 的一棵**展开树**，如果: $\hat{\rho}(\epsilon) \in I$，并且对于任意的 $x \in T$ 满足

- 对 x 在 T 中的每个子节点 $x \cdot c$，都有 $(\hat{\rho}(x), \hat{\rho}(x \cdot c)) \in \Delta$;
- 对每个 $s \in S$，若 $(\hat{\rho}(x), s) \in \Delta$，则存在 x 的某个子节点 $x \cdot c$，使得 $\hat{\rho}(x \cdot c) = s$。

称分支结构 $\langle T, \rho\rangle$ 是 $\langle T, \hat{\rho}\rangle$ 对应的**派生分支结构**，如果 $\rho = \lambda \circ \hat{\rho}$。记 $\mathbf{T}(M)$ 为 M 所有展开树对应的派生分支结构的集合。 □

在定义了迁移系统的派生线性结构和派生分支结构后，就可以对模型检验问题进行统一的定义了。

定义 2.3.4 (线性时序逻辑的模型检验问题) 给定迁移系统 $M = \langle S, \Delta, I, \lambda\rangle$，以及

线性时间的时序逻辑（LTL、ETL、线性 μ-演算等）公式 φ，称 M **满足** φ（记作 $M \models \varphi$），是指对任意的 $\pi \in \mathbf{L}(M)$ 都有 $\pi \models \varphi$。□

定义 2.3.5 (分支时序逻辑的模型检验问题) 给定迁移系统 $M = \langle S, \Delta, I, \lambda \rangle$，以及分支时间的时序逻辑（$CTL$、模态 μ-演算等）公式 φ，称 M **满足** φ（记作 $M \models \varphi$），是指对任意的 $\langle T, \rho \rangle \in \mathbf{T}(M)$ 都有 $T, \rho \models \varphi$。□

注意：CTL* 中既包括线性时间公式（路径公式），也包括分支时间的公式（状态公式）。现在约定：将 CTL* 看作是分支时间的时序逻辑。并且，若待检验的 CTL* 公式 φ 是一个路径公式，则认为是对 $\mathsf{A}\varphi$ 的检验。

定义 2.3.6 (迁移系统的合成) 给定迁移系统 $M_i = \langle S_i, \Delta_i, I_i, \lambda_i \rangle$（$i = 1, 2$），则 M_1 与 M_2 的**合成**，记作 $M_1 \| M_2$，是一个迁移系统 $\langle S, \Delta, I, \lambda \rangle$，其中：

- $S = \{(s_1, s_2) \mid s_1 \in S_1, s_2 \in S_2, \lambda_1(s_1) = \lambda_2(s_2)\}$。
- $((s_1, s_2), (s_1', s_2')) \in \Delta$ 当且仅当 $(s_1, s_1') \in \Delta_1$ 并且 $(s_2, s_2') \in \Delta_2$。
- $I = (I_1 \times I_2) \cap S$。
- $\lambda((s_1, s_2)) = \lambda_1(s_1) = \lambda_2(s_2)$。□

但是，迁移系统的描述能力并不等价于 ω-正规集。为此，在实际应用中，往往需要添加一系列的公平性约束。

定义 2.3.7 (公平迁移系统) 一个**公平迁移系统**是一个序偶 $\mathcal{M} = \langle M, \mathcal{C} \rangle$。其中，$M = \langle S, \Delta, I, \lambda \rangle$ 是一个迁移系统，称为 $\mathcal{M}$ 的**基迁移系统**；$\mathcal{C} = \{C_1, \ldots, C_m\}$，是一组**公平性约束**，其中每个公平性约束 $C_i \subseteq S$。为方便起见，在多数情况下，将 $\mathcal{M}$ 直接记作序偶 $\langle S, \Delta, I, \lambda, \mathcal{C} \rangle$。□

在公平迁移系统中，有下列重要概念。

定义 2.3.8 (公平展开迹) 设公平迁移系统 $\mathcal{M} = \langle M, \mathcal{C} \rangle$，其中 $M = \langle S, \Delta, I, \lambda \rangle$，$\mathcal{C} = \{C_1, \ldots, C_m\}$，且 $\hat{\pi}$ 是 M 的一个展开迹。令 $\mathbf{Inf}(\hat{\pi}) = \{s \in S \mid$ 有无穷多个 $i \in \mathbb{N}$ 使得 $\hat{\pi}(i) = s\}$。则称 $\hat{\pi}$ 是 $\mathcal{M}$ 中的**一条公平展开迹**，当且仅当对每个 $1 \leq i \leq m$ 有 $\mathbf{Inf}(\hat{\pi}) \cap C_i \neq \emptyset$。记 $\mathbf{L}(\mathcal{M})$ 为 M 所有的公平展开迹对应的派生线性结构。□

设 $\mathcal{M} = \langle M, \mathcal{C} \rangle$，注意区分 $\mathbf{L}(\mathcal{M})$ 和 $\mathbf{L}(M)$ 的区别。易知，$\mathbf{L}(\mathcal{M}) \subseteq \mathbf{L}(M)$ 是成立的，但反之不真。

定义 2.3.9 (展开树中的公平路径) 设 $\langle T, \hat{\rho} \rangle$ 是 M 的一个展开树，σ 是 T 中的一条（无穷）路径。令 $\mathbf{Inf}(\hat{\rho}(\sigma)) = \{s \in S \mid$ 有无穷多个 σ 中的节点 x 使得 $\hat{\pi}(x) = s\}$。则称 σ 是 T 中的一条（在公平性约束 $\mathcal{C}$ 下的）**公平路径**，当且仅当对每个 $1 \leq i \leq m$ 有 $\mathbf{Inf}(\hat{\rho}(\sigma)) \cap C_i \neq \emptyset$。□

同样，可以定义两个公平迁移系统的“合成”的概念如下。

定义 2.3.10 给定公平迁移系统 $\mathcal{M}_i = \langle M_i, \mathcal{C}_i \rangle$，且 M_i 的状态集为 S_i（$i = 1, 2$），则 $\mathcal{M}_1$ 与 $\mathcal{M}_2$ 的**合成** $\mathcal{M}_1 \| \mathcal{M}_2 = \langle M_1 \| M_2, \mathcal{C} \rangle$。其中，$\mathcal{C} = \{(C \times S_2) \cap S \mid C \in \mathcal{C}_1\} \cup \{(S_1 \times C) \cap S \mid C \in \mathcal{C}_2\}$。 □

对于迁移系统的合成，容易证明下面的定理。

定理 2.10 $\mathbf{L}(M_1 \| M_2) = \mathbf{L}(M_1) \cap \mathbf{L}(M_2)$; $\mathbf{L}(\mathcal{M}_1 \| \mathcal{M}_2) = \mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2)$。

对公平迁移系统而言，所关心的只是那些满足公平性约束的路径。现在，就可以定义“**带公平性约束的模型检验**”的概念了。

定义 2.3.11 (带公平性约束的线性时序逻辑模型检验) 给定公平迁移系统 $\mathcal{M} = \langle M, \mathcal{C} \rangle$，其中 $M = \langle S, \Delta, I, \lambda \rangle$。称 $\mathcal{M}$ **满足** 线性时序逻辑公式 φ（记作 $\mathcal{M} \models \varphi$），是指对任意的 $\pi \in \mathbf{L}(\mathcal{M})$，都有 $\pi \models \varphi$。 □

注意，线性时间的时序逻辑是可以表示公平性的：设 $M = \langle S, \Delta, I, \lambda \rangle$ 以及 $\mathcal{M} = \langle M, \mathcal{C} \rangle$，其中 $\mathcal{C} = \{C_1, \ldots, C_m\}$。则对每个 $C_i \in \mathcal{C}$，添加一个新的原子命题 p_i，使得 $p_i \in \lambda(s)$ 当且仅当 $s \in C_i$，则令

$$\Phi = \bigwedge_{i=1}^{m} \mathsf{GF} p_i$$

则容易验证：$\mathcal{M} \models \varphi$ 当且仅当 $M \models \Phi \to \varphi$。这样，就可以转化成为不带公平性约束的模型检验问题。

因此，公平性主要应用于分支时序逻辑（如 CTL）的模型检验之中。在给出其定义之前，首先介绍分支时序逻辑的**公平语义**。

定义 2.3.12 (分支时序逻辑的公平语义) 设公平迁移系统 $\mathcal{M} = \langle M, \mathcal{C} \rangle$，其中 $M = \langle S, \Delta, I, \lambda \rangle$。设 $\langle T, \hat{\rho} \rangle$ 是 M 的一棵展开树，再令 $\rho = \lambda \circ \hat{\rho}$，则对各种分支时序逻辑公式，可以定义其相对于 $\mathcal{C}$ 的**公平语义**。它可在普通语义基础上（见 *2.2* 节）通过下列修改得到。

（**CTL**） 对 *CTL*，在定义 *2.2.6* 的基础上，修改 EX、AU 以及 EU 的语义如下。

- $T, \rho, x \models \mathsf{EX}\psi$ 当且仅当 T 中存在公平性约束 $\mathcal{C}$ 下的公平路径 σ，使得 $\sigma(0) = x$，且 $T, \rho, \sigma(1) \models \psi$。
- $T, \rho, x \models \mathsf{A}(\varphi_1 \mathsf{U} \varphi_2)$ 当且仅当对 T 中每个在公平性约束 $\mathcal{C}$ 下的公平路径 σ，若 $\sigma(0) = x$，则存在 $k \in \mathbb{N}$，使得 $T, \rho, \sigma(k) \models \varphi_2$，且对于任意的 $0 \le j < k$ 都有 $T, \rho, \sigma(j) \models \varphi_1$。
- $T, \rho, x \models \mathsf{E}(\varphi_1 \mathsf{U} \varphi_2)$ 当且仅当 T 中存在公平性约束 $\mathcal{C}$ 下的公平路径 σ 以及 $k \in \mathbb{N}$，使得 $\sigma(0) = x$，且 $T, \rho, \sigma(k) \models \varphi_2$ 以及对于任意的 $0 \le j < k$，都有 $T, \rho, \sigma(j) \models \varphi_1$。

（**模态μ-演算**） 对模态 μ-演算，当定义满足集合时，在定义 *2.2.18* 的基础上修改 $\Box$ 的语义如下。

- 若 $\varphi = \Box\psi$，则 $[\![\varphi]\!]_{\langle T,\rho\rangle}E = \{x \in T \mid$ 对 T 中每个在约束 $\mathcal{C}$下的公平路径 σ, 若 $\sigma(0) = x$ 则 $\sigma(1) \in [\![\psi]\!]_{\langle T,\rho\rangle}E\}$。

（**CTL***） 对CTL^*，在定义 *2.2.8* 的基础上，修改 A 的语义如下。

- 若 $\varphi = \mathsf{A}\psi$，则 $T, \rho, x \models \varphi$ 当且仅当对 T 中任意在 $\mathcal{C}$ 下的公平路径 σ 都有: 若 $\sigma(i) = x$，则 $T, \rho, \sigma, i \models \psi$。

除上述算子外，其余算子的语义定义同前。 □

定义 2.3.13 (带公平性约束的分支时序逻辑模型检验) 设公平迁移系统 $\mathcal{M} = \langle M, \mathcal{C}\rangle$，其中 $M = \langle S, \Delta, I, \lambda\rangle$。称 $\mathcal{M}$ **满足** 分支时序逻辑公式 φ（记作 $\mathcal{M} \models \varphi$），是指对 M 的每个展开树 $\langle T, \hat{\rho}\rangle$，在相对于 $\mathcal{C}$ 的公平语义下，有 $T, \lambda \circ \hat{\rho} \models \varphi$ 成立。 □

2.3.2 二叉决策图: BDD

BDD（Binary Decision Diagram）是由 Bryant[18] 引入的用以表示布尔公式的数据结构，在该数据结构上可以高效的实现各种布尔操作。在上个世纪 90 年代，McMillan 等人将 BDD 技术引入了模型检验领域，建立了符号化模型检验技术[40]。该技术极大的提高了模型检验能够处理的问题规模。

定义 2.3.14 (二叉决策图) 一个**二叉决策图**（BDD）是一个具有唯一根节点（即: 入度为 0 的节点）的分层有向无环图 $\mathcal{D}$。其节点集中包括一组终结节点和一组非终结节点。

在每个非终结节点 t 上都标记一个布尔变元 $Lvar(t)$，且同一层的非终止节点上标记的布尔变元相同。在每个终结节点 t 上都标记一个布尔常量 $Lcons(t)$。此外，每个非终结节点 t 都有两个直接子节点 $High(t)$ 和 $Low(t)$; 终结节点的出度为 0。记 $\mathcal{D}$ 的根节点为 $Root(\mathcal{D})$。同时，对 $\mathcal{D}$ 中的每个节点 t，将 $\mathcal{D}$ 中以 t 为根节点的子图记为 $SubTr_{\mathcal{D}}(t)$。 □

定义 2.3.15 (布尔公式的 BDD 表示) 给定BDD $\mathcal{D}$ 以及原子命题指派 $e : AP \to \{0, 1\}$，则从任意一个节点 t 都可获得一个布尔值，记作 $Val(e, t)$，归纳定义如下:

- 若 t 是终结节点，则 $Val(e, t) = Lcons(t)$。
- 若 t 是非终结节点，且 $e(Lvar(t)) = 1$，则 $Val(e, t) = Val(e, High(t))$。
- 若 t 是非终结节点，且 $e(Lvar(t)) = 0$，则 $Val(e, t) = Val(e, Low(t))$。

称BDD $\mathcal{D}_\varphi$ 是布尔公式 φ 的***BDD* 表示**，如果对于任意的原子命题指派 e 有: $e \models \varphi$ 当且仅当 $Val(e, Root(\mathcal{D}_\varphi)) = 1$。 □

例 2.3.1 考虑包含布尔变元 z_0，z_1，z_2 的布尔公式 $z_0 \leftrightarrow (z_1 \wedge \neg z_2)$，其 *BDD* 表示如图 *2.5* 表示。 □

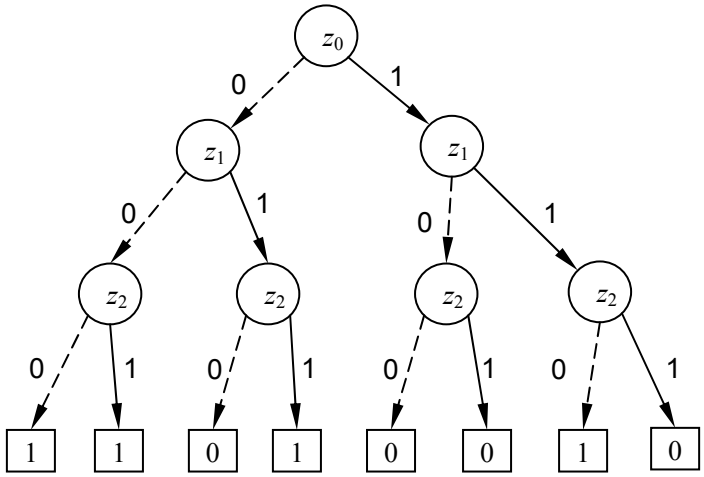


图 2.5 布尔公式 BDD 表示示例

对于任意一个 DBB，可以按照下列过程对其进行自底向上的约简：

合并终结节点：将标记相同的终结节点合并，并将合并前指向终结节点的边重定向至合并后的相应终结节点。

合并等价节点：若存在两个非终结节点 t_1 和 t_2，使得 $Lvar(t_1) = Lvar(t_2)$，$High(t_1) = High(t_2)$，且 $Low(t_1) = Low(t_2)$，则将 t_1 和 t_2 合并，然后将合并前指向 t_1 和 t_2 的边重定向至合并后的节点。

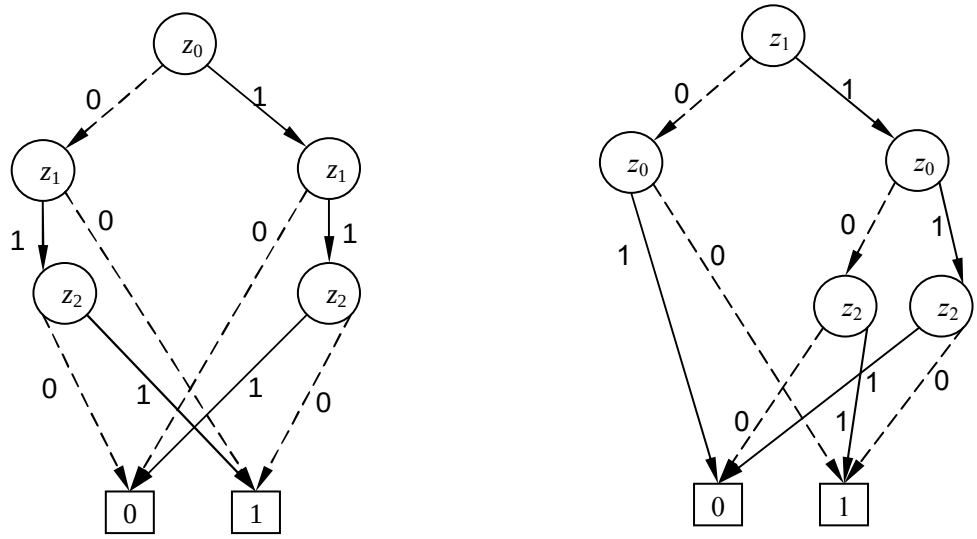
删除冗余节点：若某节点 t 满足 $High(t) = Low(t)$，则将 t 删除，并将原来指向 t 的边重定向到 $High(t)$（或 $Low(t)$）。

例 2.3.2 由图 *2.5* 中的 *BDD* 得到最终约简 *BDD* 如图 *2.6(a)* 所示。 □

应当注意的是：采取不同的命题排列顺序，最终得到的约简 BDD 会不同。比如，图 2.6(b) 中所示的 BDD 是由布尔公式 $z_0 \leftrightarrow (z_1 \wedge \neg z_2)$ 对应的 BDD 按照命题排序 $z_1 > z_0 > z_2$ 得到的约简 BDD。通常，最终得到的约简 BDD 的复杂程度受命题排序的影响很大。但是，寻找一个使得约简 BDD 较小的命题排序本身是一个 NP-hard 的问题[18]。比较经典的启发式排序方法可以见文 [95, 96] 等。

定义 2.3.16 (ROBDD) 按照某给定的命题顺序约简后得到的 *BDD* 称为***ROBDD***（*Reduced Ordered Binary Decision Diagram*）。 □

当命题顺序给定后，常用的布尔操作（比如："与"、"或"、"非" 等布尔操作）都能够在 ROBDD 上得以高效实现。在执行一元布尔操作"取非"时，只需将终结节点



(a) 按原始命题排序得到的约简 BDD　　(b) 按其他命题排序得到约简 BDD

图 2.6 按两种命题排序得到的约简 BDD

的标记翻转即可。设 $\mathcal{D}_{\varphi_1}$ 和 $\mathcal{D}_{\varphi_2}$ 分别是布尔公式 φ_1 和 φ_2 对应的 ROBDD，则对任一种二元布尔操作 $\star$（注：二元布尔操作共16 种），可以按照下列过程得到布尔公式 $\varphi_1 \star \varphi_2$ 对应的 ROBDD $\mathcal{D}_{\varphi_1 \star \varphi_2}$。

1. 令 t_1 和 t_2 分别是 $Root(\mathcal{D}_{\varphi_1})$ 和 $Root(\mathcal{D}_{\varphi_2})$。
2. 若 t_1 和 t_2 都是终结节点，则创建新的终结节点 t，将 $Lcons(t)$ 设为 $Lcons(t_1) \star Lcons(t_2)$。
3. 若 t_i 是非终结节点（$i \in \{1,2\}$），t_{3-i} 也是非终结节点且 $Lvar(t_i) > Lvar(t_{3-i})$ 或者 t_{3-i} 是终结节点，则创建新的非终结节点 t，将 $Lvar(t)$ 设为 $Lvat(t_i)$；同时，将 $Low(t)$ 和 $High(t)$ 分别设为 $Root(\mathcal{D}_{low})$ 和 $Root(\mathcal{D}_{high})$。其中，$\mathcal{D}_{low}$ 和 $\mathcal{D}_{high}$ 分别是以 $SubTr_{\mathcal{D}_{\varphi_i}}(Low(t_i))$ 和 $\mathcal{D}_{\varphi_{3-i}}$ 以及以 $SubTr_{\mathcal{D}_{\varphi_i}}(High(t_i))$ 和 $\mathcal{D}_{\varphi_{3-i}}$ 为参数递归调用此过程得到的 ROBDD。
4. 若 t_1 和 t_2 都是非终结节点，且 $Lvar(t_1) = Lvar(t_2)$，则创建新的非终结节点 t，将 $Lvar(t)$ 设为 $Lvat(t_1)$；同时，将 $Low(t)$ 和 $High(t)$ 分别设为 $Root(\mathcal{D}_{low})$ 和 $Root(\mathcal{D}_{high})$。其中，$\mathcal{D}_{low}$ 和 $\mathcal{D}_{high}$ 分别是以 $SubTr_{\mathcal{D}_{\varphi_1}}(Low(t_1))$ 和 $SubTr_{\mathcal{D}_{\varphi_2}}(Low(t_2))$ 以及以 $SubTr_{\mathcal{D}_{\varphi_1}}(High(t_1))$ 和 $SubTr_{\mathcal{D}_{\varphi_2}}(High(t_2))$ 为参数递归调用此过程得到的 ROBDD。
5. 最后，返回以 t 为根的 ROBDD 即可。

2.3.3 基于 BDD 的 CTL 符号化模型检验

在模型检验技术中，最先采用基于 BDD 的符号化技术的是针对采用 CTL 为规约语言的模型检验[40]。同时，CTL 的符号化模型检验也是其他类型时序逻辑的符号化模型检验的基础。在本文的第 5 章和第 6 章中将会分别讨论如何将 ETL 和线性 μ-演算的模型检验问题转化为 CTL 的符号化模型检验问题。因此，有必要对 CTL 符号模型检验技术进行简单的回顾。

带公平性约束的 CTL 符号化模型检验主要包括两个过程：

1. 将带公平性约束的 CTL 模型检验问题转化成为不带公平性约束的模态 μ-演算（逻辑片段）的模型检验问题。
2. 利用 BDD 技术，执行符号化模型检验过程。

下面，就这两个步骤分别加以阐述。

2.3.3.1 从带公平性的 CTL 检验到不带公平性的模态 μ-演算检验

在定义 2.2.18 中，曾给出了模态 μ-演算在分支模型上的语义（满足集）。在模态 μ-演算中，μ 和 ν 实际上分别是极小、极大不动点算子。通常而言，不动点的计算需要迭代[97]，而在无穷模型上，该迭代往往不能在有穷步内终止。为此，需要重新定义模态 μ-演算在有穷迁移系统上的语义。

定义 2.3.17 (模态 μ-演算在有穷迁移系统上的语义)
给定有穷迁移系统 $M = \langle S, \Delta, I, \lambda \rangle$，模态 μ-演算公式 φ，以及变元指派 $E : VAR \to 2^S$，则可以定义 φ 的**满足集** $[\![\varphi]\!]_M E$，它将公式 φ 映射为 S 的某个子集，归纳定义如下。

- 若 $\varphi = p \in AP$，则 $[\![\varphi]\!]_M E = \{s \in S \mid p \in \lambda(s)\}$。
- 若 $\varphi = X \in VAR$，则 $[\![\varphi]\!]_M E = E(X)$。
- 若 $\varphi = \neg\psi$，则 $[\![\varphi]\!]_M E = S \backslash [\![\psi]\!]_M E$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $[\![\varphi]\!]_M E = [\![\varphi_1]\!]_M E \cap [\![\varphi_2]\!]_M E$。
- 若 $\varphi = \Box\psi$，则 $[\![\varphi]\!]_M E = \{s \in S \mid$ 对每个$s' \in S$，若 $(s, s') \in \Delta$，则 $s' \in [\![\psi]\!]_M E\}$。
- 若 $\varphi = \mu X.\psi$，则 $[\![\varphi]\!]_M E = \bigcap\{S' \subseteq S \mid [\![\psi]\!]_M E[X/S'] \subseteq S'\}$。

特别的，当 φ 是句子时，直接将 $[\![\varphi]\!]_M E$ 写作 $[\![\varphi]\!]_M$。 □

由上述定义以派生连接子（或算子）$\vee$、$\Diamond$ 及 ν 的定义（分别见定义式（2.1）、（2.21）及（2.22））不难得到下面的推论：

- 若 $\varphi = \varphi_1 \vee \varphi_2$，则 $[\![\varphi]\!]_M E = [\![\varphi_1]\!]_M E \cup [\![\varphi_2]\!]_M E$。

- 若 $\varphi = \Diamond\psi$，则 $[\![\varphi]\!]_M E = \{s \in S \mid$ 存在$s' \in S$，使得 $(s, s') \in \Delta$，且 $s' \in [\![\psi]\!]_M E\}$。
- 若 $\varphi = \nu X.\psi$，则 $[\![\varphi]\!]_M E = \bigcup\{S' \subseteq S \mid S' \subseteq [\![\psi]\!]_M E[X/S']\}$。

下面的定理说明了这两种语义及（无公平性约束）模型检验之间的关系。

定理 2.11 给定迁移系统 $M = \langle S, \Delta, I, \lambda\rangle$，以及 μ-演算句子 φ，则在非公平语义下，下面三个条件等价:

- $M \models \varphi$。 ⎫不带公平性模型检验⎭
- $I \subseteq [\![\varphi]\!]_M$。 ⎫有穷语义⎭
- 对 M 的任意一展开树 $\langle T, \hat{\rho}\rangle$，都有 $T, \lambda \circ \hat{\rho} \models \varphi$。 ⎫原始不带公平性语义⎭

这样，在检验 $M \models \varphi$ 是否成立时，只需检验是否有 $I \subseteq [\![\varphi]\!]_M E$ 成立即可。定义 2.3.17 中给出了形如 $\mu X.\psi$ 和 $\nu X.\psi$ 公式满足集的定义式，但在实际计算中，可以利用如下的迭代过程计算 $[\![\mu X.\psi]\!]_M E$（resp. $[\![\nu X.\psi]\!]_M E$）：

1. 令 $S_0 = \emptyset$（resp. $S_0 = S$）。
2. 令 $S_{i+1} = [\![\psi]\!]_M E[X/S_i]$。
3. 若发现 $S_{k+1} = S_k$，则计算终止，遂令 $[\![\mu X.\psi]\!]_M E$（resp. $[\![\nu X.\psi]\!]_M E$）为 S_k。

由于 X 在 ψ 中的所有出现均为正出现，不难证明：上述迭代序列 $S_0, S_1, \ldots, S_k$ 是一个单调递增（resp. 单调递降）的序列。又因为 M 的状态集 S 有穷，所以上述迭代一定终止。

给定公平迁移系统 $\mathcal{M} = \langle M, \mathcal{C}\rangle$ 以及 CTL 公式 φ，在检验是否有 $\mathcal{M} \models \varphi$ 时，需要对模型和规约做下列变换。

首先，不妨设 $M = \langle S, \Delta, I, \lambda\rangle$，$\mathcal{C} = \{C_1, \ldots, C_m\}$，则为每个 C_i 引入一个新的命题 p_i，定义新的标记函数 λ'，它满足：

- 对每个 $p \in AP\backslash\{p_1, \ldots, p_m\}$ 以及 $s \in S$，$p \in \lambda'(s)$ 当且仅当 $p \in \lambda(s)$。
- 对每个 $p_i \in \{p_1, \ldots, p_m\}$ 以及 $s \in S$ 有：$p_i \in \lambda'(s)$ 当且仅当 $s \in C_i$。

将迁移系统 $\langle S, \Delta, I, \lambda'\rangle$ 记为 M'。

同时，应当注意到，在 CTL 中，使用 EX、EU 以及 EG 这三个时序连接子以及布尔连接子能够表示出其他的时序算子。这是因为：

$$\mathsf{E}(\varphi_1 \mathsf{R} \varphi_2) \quad \leftrightarrow \quad \mathsf{E}(\varphi_2 \mathsf{U}(\varphi_1 \wedge \varphi_2)) \vee \mathsf{EG}\varphi_2 \tag{2.27}$$

$$\mathsf{A}(\varphi_1 \mathsf{U} \varphi_2) \quad \leftrightarrow \quad \neg\mathsf{E}(\neg\varphi_1 \mathsf{R} \neg\varphi_2) \tag{2.28}$$

于是，对于任意的 CTL 公式 φ，存在模态 μ-演算公式 $\hat{\varphi}$，使得 $\mathcal{M}$ 在公平性约束 $\mathcal{C}$ 下满足 φ 当且仅当 M' 满足 $\hat{\varphi}$。这里，$\hat{\varphi}$ 归纳构造如下（[98, 40]）：

- 若 $\varphi = p \in AP$，则 $\hat{\varphi} = p$。
- 若 $\varphi = \neg\psi$，则 $\hat{\varphi} = \neg\hat{\psi}$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\hat{\varphi} = \hat{\varphi_1} \wedge \hat{\varphi_2}$。
- 若 $\varphi = \mathsf{EX}\psi$，则 $\hat{\varphi} = \Diamond(\hat{\psi} \wedge \Psi_{\mathcal{C}})$，其中 $\Psi_{\mathcal{C}} = \nu Z.(\bigwedge_{i=1}^{m} \mu Y.((p_i \wedge \Diamond Z) \vee \Diamond Y))$。
- 若 $\varphi = \mathsf{E}(\varphi_1 \mathsf{U} \varphi_2)$，则 $\hat{\varphi} = \mu X.((\hat{\varphi_2} \wedge \Psi_{\mathcal{C}}) \vee (\hat{\varphi_1} \wedge \Diamond X))$。
- 若 $\varphi = \mathsf{EG}\psi$，则 $\hat{\varphi} = \nu Z.(\hat{\psi} \wedge \bigwedge_{i=1}^{m} \mu Y.((p_i \wedge \Diamond Z) \vee \Diamond(Y \wedge \hat{\psi})))$。

上述过程中，由 CTL 公式消去公平性约束得到的都是**交换深度** [66]不超过 2 的模态 μ-演算公式。对这类 μ-演算公式断进行检验的复杂度是非常低的（见 [65]）。于是，对 $\mathcal{M} \models \varphi$ 的检验问题就转化为对 $M' \models \hat{\varphi}$ 的检验问题。

2.3.3.2 基于 BDD 的模态 μ-演算符号化模型检验

现在说明如何使用基于 BDD 的技术对不带公平性的迁移系统执行针对模态 μ-演算的符号化模型检验。

首先，介绍如何使用布尔公式对迁移系统进行编码。

定义 2.3.18 (迁移系统的布尔编码) 给定迁移系统 $M = \langle S, \Delta, I, \lambda \rangle$，可以采用一组位变元 $\vec{Z} = \{z_1, \ldots, z_n\}$ 对其进行**编码**。具体过程如下:

状态集合 每个状态 $s \in S$ 都唯一对应 $\vec{Z}$ 中位变元的一个真值指派。若 Z 的每个真值指派用一个 n 位 $\{0,1\}$-向量表示，则可以用一个映射函数 $f : S \to \{0,1\}^n$ 表示该种对应。

在编码时，要求 $2^n \geq \#S$，所以并不是每个指派都有 S 中的状态与之对应。因此，需定义**合法状态约束** $\Phi_S(\vec{Z})$。它是一个布尔函数，满足: 若 $(v_1, \ldots, v_n) \in \{0,1\}^n$ 对应于 S 中的一个合法状态，当且仅当将每个 z_i 的值赋为 v_i 后，Φ_s 的值为 1。

迁移关系 令 $\vec{Z'} = \{z'_1, \ldots, z'_n\}$。则对迁移关系 Δ，定义其布尔编码函数 $\Phi_\Delta(\vec{Z}, \vec{Z'})$，它满足: 对于任意的 $s, s' \in S$，若 $f(s) = (v_1, \ldots, v_n)$，$f(s') = (v'_1, \ldots, v'_n)$，则 $(s, s') \in \Delta$ 当且仅当将每个 z_i 的值赋为 v_i，将每个 z'_i 的值赋为 v'_i 后，Φ_Δ 的值为 1。

初始状态集 为初始状态集 I 定义布尔函数 $\Phi_I(\vec{Z})$，它满足: 对任意的 $s \in S$，若 $f(s) = (v_1, \ldots, v_n)$，则 $s \in I$ 当且仅当将每个 z_i 的值赋为 v_i 后，Φ_I 的值为 1。

命题标记函数 在迁移系统的原始定义中，λ 是一个从 S 到 2^{AP} 的映射。由于一个迁移系统中所涉及的命题数目是有限的，因此可以为 M 中涉及的每个原子命题 p 建立一个布尔编码函数 $\Phi^p_\lambda(\vec{Z})$，它满足: 对于任意的 $s \in S$，若 $f(s) =$

$(v_1,\ldots,v_n)$，则 $p\in\lambda(s)$ 当且仅当将每个 z_i 的值赋为 v_i 后，Φ^p_λ 的值为 1。更一般的情形，对于任意的 $S'\subseteq S$，都存在一个布尔公式 $\Phi_{S'}(\vec{Z})$，它满足：对任意的 $s\in S$，若 $f(s)=(v_1,\ldots,v_n)$，则 $s\in S'$ 当且仅当将每个 z_i 的值赋为 v_i 后 $\Phi_{S'}$ 的值为 1。

此外，对于带公平限制的迁移系统 $\mathcal{M}=\langle M,\{C_1,\ldots,C_m\}\rangle$，除引入上述布尔编码之外，再为每个 C_i 引入布尔编码函数 $\Phi_{C_i}(\vec{Z})$，该函数满足：对任意的 $s\in S$，若 $f(s)=(v_1,\ldots,v_n)$，则 $s\in C_i$ 当且仅当 z_i 的值赋为 v_i 后 Φ_{C_i} 的值为 1。为方便起见，记布尔公式集合 $\{\Phi_{C_1}\ldots,\Phi_{C_m}\}$ 为 $\Phi_{\mathcal{C}}$。 □

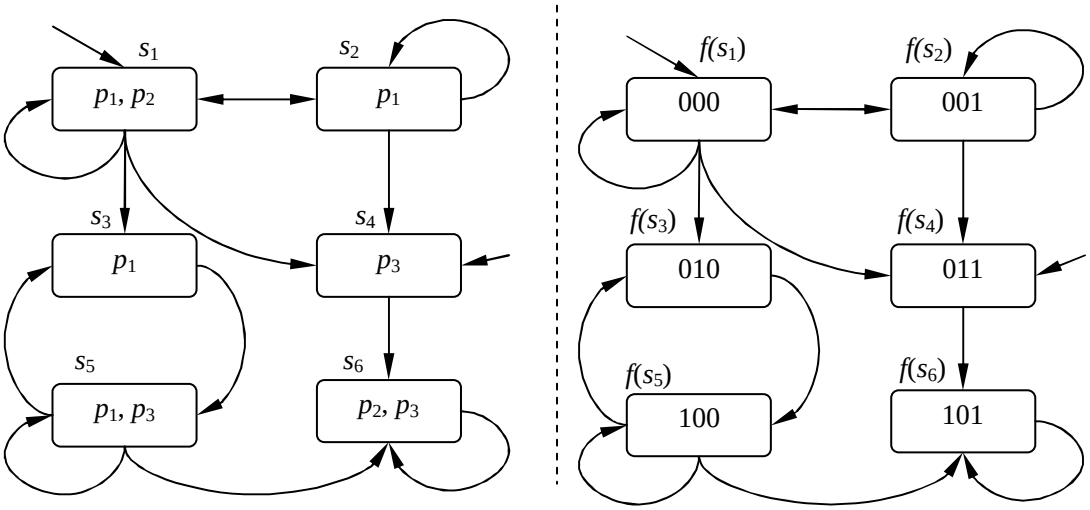


图 2.7 迁移系统布尔编码的示例

例 2.3.3 如图 *2.7* 所示的 6 状态迁移系统 $\langle S=\{s_1,\ldots,s_6\},\Delta,I=\{s_1,s_4\},\lambda\rangle$，可以采用三个布尔变元 z_0，z_1，z_2 对其编码。设状态映射关系为 f（例如：$f(s_5)$ 表示将 z_0，z_1，z_2 的值分别赋为 1，0，0），则其编码情况如下：

- $\Phi_S=\neg z_0\vee(z_0\wedge\neg z_1)$。
- 迁移系统中每一条边，可以编码为一个关于 $\vec{Z}$ 和 $\vec{Z'}$ 的布尔公式。比如，s_6 上的自圈可以表示为 $(z_0\wedge\neg z_1\wedge z_2)\wedge(z'_0\wedge\neg z'_1\wedge z'_2)$。于是，令 Φ_Δ 为这些边编码的析取即可。
- $\Phi_I=\neg z_0\wedge((\neg z_1\wedge\neg z_2)\vee(z_1\wedge z_2))$（或者化简为 $\neg z_0\wedge(z_1\leftrightarrow z_2)$）。
- M 中涉及的命题共有3 个：p_1，p_2 及 p_3。标记函数分别如下：
  1. $\Phi^{p_1}_\lambda=\neg z_0\wedge(\neg z_1\vee\neg z_2)\vee\neg z_2\wedge(\neg z_0\leftrightarrow z_1)$;
  2. $\Phi^{p_2}_\lambda=\neg z_1\wedge(z_0\leftrightarrow z_2)$;
  3. $\Phi^{p_3}_\lambda=(z_0\wedge\neg z_1)\vee(z_2\wedge(\neg z_0\leftrightarrow z_1))$。 □

两个迁移系统的合成也能够高效的基于布尔编码表示实现。设迁移系统 $M_i = \langle S_i, \Delta_i, I, \lambda_i \rangle$（$i \in \{1,2\}$），则对于 $M_1 \| M_2$ 而言：

- 设 M_1 和 M_2 中共同涉及的原子命题集合为 CP，则 $M_1 || M_2$ 的合法状态约束为：$\Phi_{S_1} \wedge \Phi_{S_2} \wedge \bigwedge_{p \in CP}(\Phi^p_{\lambda_1} \leftrightarrow \Phi^p_{\lambda_2})$。
- $M_1 \| M_2$ 的迁移关系的布尔编码为 $\Phi_{\Delta_1} \wedge \Phi_{\Delta_2}$。
- $M_1 \| M_2$ 的初始状态集的布尔编码为 $\Phi_{I_1} \wedge \Phi_{I_2}$。
- 对于每个原子命题 p，若 p 只在 M_i 中出现（$i \in \{1,2\}$），则其标记函数对应的布尔编码为 $\Phi^p_{\lambda_i}$；若 $p \in CP$，则其标记函数对应的布尔编码为 $\Phi^p_{\lambda_1} \wedge \Phi^p_{\lambda_2}$。

此外，对于两个公平迁移系统 $\mathcal{M}_i = \langle M_i, \mathcal{C}_i \rangle$（$i \in \{1,2\}$）而言，在获得 $M_1 \| M_2$ 的编码后，只需添加 $\Phi_{\mathcal{C}_1} \cup \Phi_{\mathcal{C}_2}$ 作为公平性约束编码，即可获得 $\mathcal{M}_1 \| \mathcal{M}_2$ 的布尔编码。

现在说明基于 BDD 的模态 μ-演算符号化模型检验过程。

给定迁移系统 $M = \langle S, \Delta, I, \lambda \rangle$，设其布尔编码为 $\langle \Phi_S, \Phi_\Delta, \Phi_I, \{\Phi^p_\lambda\}_p \rangle$。由于对任意的 $S' \subseteq S$，都有布尔编码 $\Phi_{S'}$ 与之对应（见定义 2.3.18），进而可以对应一个 ROBDD。于是任何一个变元指派 E 都可以等价的看作是将每个公式变元映射为一个 ROBDD 的函数。

这样，对于任意的模态 μ-演算公式 φ，以及任意的变元指派 E，可以归纳的生成 $\Phi_{[\![\varphi]\!]_M E}$（即：$[\![\varphi]\!]_M E$ 对应的布尔编码）对应的ROBDD $\mathcal{D}^E_\varphi$。

- 若 $\varphi = p \in AP$，则 $\mathcal{D}^E_\varphi$ 为 Φ^p_λ 对应的 ROBDD。
- 若 $\varphi = X \in VAR$，则 $\mathcal{D}^E_\varphi$ 为 $E(X)$ 对应的 ROBDD。
- 若 $\varphi = \neg\psi$，则 $\mathcal{D}^E_\varphi$ 为 $\mathcal{D}^E_\psi$ 经取非操作得到的 ROBDD。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\mathcal{D}^E_\varphi$ 为由 $\mathcal{D}^E_{\varphi_1}$ 和 $\mathcal{D}^E_{\varphi_2}$ 经合取操作得到的 ROBDD。
- 若 $\varphi = \Diamond\psi$，设编码 M 的位变元集合为 $\{z_1, \ldots, z_n\}$，则 $\mathcal{D}^E_\varphi$ 为布尔公式 $\exists z'_1, \ldots, z'_n.(\Phi_\Delta \wedge \Phi'_{[\![\psi]\!]_M E})$ 对应的 ROBDD。其中：
  1. $\Phi'_{[\![\psi]\!]_M E}$ 是将 $\Phi_{[\![\psi]\!]_M E}$ 中的每个 z_i 替换为 z'_i 得到的公式。
  2. $\exists z'_n \Psi$ 是 $\Psi^{z'_n}_{true} \vee \Psi^{z'_n}_{false}$ 的简写。
  3. $\exists z'_i, z'_{i+1}, \ldots, z'_n.\Psi$ 是 $\exists z'_i.(\exists z'_{i+1}, \ldots, z'_n.\Psi)$ 的简写。
- 若 $\varphi = \mu X.\psi$，则 $\mathcal{D}^E_\varphi$ 按如下过程迭代得到：
  1. 令 $\mathcal{D}_0 = \mathcal{D}^E_{\psi^X_{false}}$。
  2. 令 $\mathcal{D}_{i+1} = \mathcal{D}^{E[X/\mathcal{D}_i]}_\psi$。
  3. 若发现 $\mathcal{D}_k$ 与 $\mathcal{D}_{k+1}$ 等价，则停止迭代，令 $\mathcal{D}^E_\varphi = \mathcal{D}_k$ 即可。

最后，令 $\mathcal{D}$ 是由 $\mathcal{D}^E_{\neg\varphi}$ 和 Φ_I 对应的 ROBDD 经合取操作得到的 ROBDD（事实上，由于规约 φ 要求是句子，E 可以任取）。则由定理 2.11 有：$M \models \varphi$ 当且仅

当 $\mathcal{D}$ 对应的布尔公式为 $false$。

关于“当发现性质不被满足时，如何基于 BDD 给出反例路径” 这一问题可以参考文 [98] 中的第 8 节。

第三章　ETL 的公理化及其逻辑片断的实例公理化方法

3.1　引言

本章讨论 ETL 的公理化问题。ETL 是通过使用诸如正规文法[23]、自动机[30]、正规表达式等作为时序连接子得到的线性时序逻辑的统称。采用 ω-正规文法作为时序连接子的 ETL 最早由 Wolper 提出，其初衷在于增强 LTL 的表达能力。由于 ω-正规文法不能显式的表示接收条件，Vardi 和 Wolper 等人改用 ω-自动机作为 ETL 的时序连接子[30]。当时，着重讨论了三类自动机接收条件：finite、looping 和 repeating，并分别将以相应接收条件自动机作为连接子的 ETL 命名为 ETL_f、ETL_l 和 ETL_r。在文 [23] 中，Wolper 给出了一套 ETL 的公理系统。在本章，将 ETL 的公理化方法推广至 ETL_l、ETL_f 和 ETL_r。

Looping、finite、repeating 是三类最基本的自动机接收条件。同时，以 looping 自动机作为最外层连接子的 ETL 公式刻画了**安全性**（Safety）性质；以 finite 自动机作为最外层连接子的 ETL 公式刻画了**活性**性质。此外，repeating（Büchi）自动机具有比 finite 和 looping 自动机更强的表达能力（见下面的例子），因而能够同时刻画安全性和活性。

例 3.1.1 ([75, 99, 100, 101, 102, 103, 104, 30])

- 在字母表 $\Sigma = \{a_1, a_2\}$ 上，ω-字 a_1^ω 只能被 *looping* 自动机识别，而无法被任何 *finite* 自动机识别。
- 在字母表 $\Sigma = \{a_1, a_2\}$ 上，ω-字 $a_1^*; a_2; (a_1|a_2)^\omega$ 只能被 *finite* 自动机识别，而无法被任何 *looping* 自动机所识别。
- 在字母表 $\Sigma = \{a_1, a_2\}$ 上，ω-字 $(a_1^*; a_2)^\omega$ 既不能被任何 *finite* 自动机识别，也不能被任何 *looping* 自动机识别；它只能被 *repeating* 自动机所识别。 □

因此，finite 自动机与 looping 自动机的表达能力无法比较，且二者均严格弱于 repeating 自动机（见图 2.3）。但是，对于分别采用 NLW、NFW、NBW 作为连接子的扩展时序逻辑 ETL_l、ETL_f、ETL_r 而言，却有如下结论：

定理 3.1 ([30]) *ETL_l、ETL_f、ETL_r 的表达能力相同，且都等价于 ω-正规语言。*

采用自动机作为连接子的 ETL 的语法、语义的定义分别见定义 2.2.21 及定义 2.2.22。由于本章主要考虑的是 ETL_l、ETL_f 和 ETL_r 这三类时序逻辑，所以只关心 NLW、NFW 和 NBW 这三种自动机。因此，有以下几点特殊约定：

1. 对于 NLW 而言，由于其接收条件仅依赖于状态集，因而将 NLW 写作 $\langle \Sigma, Q, \delta, q, - \rangle$ 的形式；对于 NFW 和 NBW 而言，由于其接收条件是一个接收状态集合，因而将 NBW 写成 $\langle \Sigma, Q, \delta, q, F \rangle$ 的形式。
2. 由于本章中涉及的自动机都是非确定自动机，因而将迁移函数 δ 写成 $Q \times \Sigma \to 2^Q$ 的形式。即：$\delta(q,a) = \{q_1, \ldots, q_k\}$ 表示 $\delta(q,a) = \bigvee_{1 \le i \le k} q_i$（见 23 页约定）。
3. 在使用自动机作为 ETL 中的连接子时，字母表内的字母起到占位符的作用。这使得字母之间的相对顺序非常重要。因而，在 ETL 中，自动机的字母表不再视为一个无序的集合；而应该视做一个有序的向量。这也是 ETL 中的一般约定。
4. 由于本章中所讨论自动机都是非确定的，所以自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$ 在某 ω-字上的运行 $\langle T, \rho \rangle$ 就退化成了一条 Q-标记路径 $\rho(\sigma)$。在此约定：将 $\rho(\sigma)$ 视为 Q 上的有穷或无穷字（即：$\rho(\sigma) \in Q^\infty$）。其中，对于任意的 $i < |\sigma|$ 而言，$\rho(\sigma)$ 的第 i 个字母就是 $\rho(\sigma(i))$。
5. 同时，沿用在 2.2.4 节中的约定（见 39 页）：若自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi)$，则将公式 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 记作 φ^q。其中，$\mathcal{A}^q$ 与 $\mathcal{A}$ 唯一的区别在于前者的初始状态为 q。

根据上面的约定，ETL_l、ETL_f、ETL_r 对自动机公式的**满足关系**（见定义 2.2.22）的概念可以简化如下。

设自动机公式 $\varphi = \mathcal{A}(\varphi_1 \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, \Omega \rangle$，对每个线性结构 π 以及位置 $i \in \mathbf{N}$，则：

- 若 φ 是 ETL_f 公式，则 $\pi, i \models \varphi$ 当且仅当：将 $\mathcal{A}$ 视为有穷字上的自动机后，存在有穷字 $w \in \mathbf{L}(\mathcal{A})$，使得对于任意的 $0 \le j < |w|$ 有：若 $w(j) = a_k$，则 $\pi, i+j \models \varphi_k$。
- 若 φ 是 ETL_l 或 ETL_r 公式，则 $\pi, i \models \varphi$ 当且仅当：存在无穷字 $w \in \mathbf{L}(\mathcal{A})$，且对于任意的 $j \in \mathbf{N}$ 有：若 $w(j) = a_k$，则 $\pi, i+j \models \varphi_k$。 □

本章内容组织如下：

1. 3.2 节、3.3 节和 3.4 节分别给出关于 ETL_l、ETL_f 和 ETL_r 的可靠完备的公理系统 $\mathscr{L}$、$\mathscr{F}$ 和 $\mathscr{R}$。将分别定义三种 ETL 的“**重写规则**” 以及“**迁移图**” 的概念。进而给出基于“迁移图删除” 的完备性证明技术。
2. 3.5 节给出一种 ETL 逻辑片断（或者子逻辑）的实例化公理方法。将证明：对于 ETL 的某种逻辑片断，只需将其中的时序连接子编码为自动机，而后实例化相应 ETL（称为“基逻辑”）的公理系统中关于自动机连接子的公理以及规则，即可获得其可靠完备的公理系统。从而，该节提供了一种关于 ETL 逻辑片断的统

一的公理化框架。

3.2 $\mathbf{ETL}_l$ 的公理系统 $\mathscr{L}$

3.2.1 $\mathbf{ETL}_l$ 重写系统及迁移图

本章给出的公理化方法，都是基于“迁移图删除” 技术给出的。这里，首先给出 ETL 的**重写规则** 和 ETL 公式**迁移图** 的概念。这里，首先介绍 ETL_l 的迁移图的定义以及若干基本性质。

定义 3.2.1 (ETL 重写规则) *ETL* 的**重写规则** 是一类形如

$$\frac{\Gamma}{\Gamma'} \qquad \text{(name)}$$

的规则。其中，Γ 和 Γ' 都是有穷 *ETL* 公式集合; (name) 为该规则的**规则名**。此时，称“Γ' 可由 Γ 经规则 (name) 重写得到”。 □

为方便起见，以下将 $\Gamma \cup \{\varphi\}$ 简写为 Γ, φ；将 $\bigwedge_{\phi\in\Gamma}\phi$ 简写为 $\bigwedge\Gamma$；将 $\bigvee_{\phi\in\Gamma}\phi$ 简写为 $\bigvee\Gamma$。同时规定：$\bigwedge\emptyset = true$，$\bigvee\emptyset = false$。

各类 ETL 的公共重写规则如下。

$$\frac{\Gamma, \neg(\varphi_1 \vee \varphi_2)}{\Gamma, \neg\varphi_1 \wedge \neg\varphi_2} \quad \text{(nor)} \qquad \frac{\Gamma, \neg(\varphi_1 \wedge \varphi_2)}{\Gamma, \neg\varphi_1 \vee \neg\varphi_2} \quad \text{(nand)}$$

$$\frac{\Gamma, \neg\bigcirc\psi}{\Gamma, \bigcirc\neg\psi} \quad \text{(nnext)} \qquad \frac{\Gamma, \neg\neg\psi}{\Gamma, \psi} \quad \text{(nneg)}$$

$$\frac{\Gamma, \varphi_1 \vee \varphi_2}{\Gamma, \varphi_i} \quad \text{(or)} \qquad \frac{\Gamma, \varphi_1 \wedge \varphi_2}{\Gamma, \varphi_1, \varphi_2} \quad \text{(and)}$$

$$\frac{\phi_1, \ldots, \phi_m, \bigcirc\psi_1, \ldots, \bigcirc\psi_k}{\psi_1, \ldots, \psi_k} \qquad \text{(modal)}$$

其中，在规则 (or) 中，$i \in \{1, 2\}$。在规则 (modal) 中，要求每个 ϕ_i 都是**文字**。即：$\phi_i \in AP \cup \overline{AP} \cup \{true, false\}$，这里 $\overline{AP} = \{\neg p \mid p \in AP\}$。注意，在上述列出的重写规则中，除 (modal) 外，对重写前的公式集 Γ, φ 而言，要求 $\varphi \notin \Gamma$（下同）。

在 ETL_l 中，还有两条关于自动机公式的特殊的重写规则 (pexp) 和 (nexp)，分别描述如下。

$$\frac{\Gamma, \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, \bigvee_{1\le k\le n}(\varphi_k \wedge \bigvee_{q'\in\delta(q,a_k)} \bigcirc \mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n))} \quad \text{(pexp)}$$

$$\frac{\Gamma, \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, \bigwedge_{1\le k\le n}(\neg\varphi_k \vee (\varphi_k \wedge \bigwedge_{q'\in\delta(q,a_k)} \bigcirc \neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)))} \quad \text{(nexp)}$$

其中，$\mathcal{A}^q$ 为 NLW $\langle\{a_1, \ldots, a_n\}, Q, \delta, q, -\rangle$。注意，重写规则 (pexp) 和 (nexp) 并不是完全的对偶。

定义 3.2.2 (消解公式、生成公式) 在使用 *ETL* 重写规则由 Γ 获得 Γ' 时，Γ 中被选择重写的公式称为**消解公式**; Γ' 中由重写规则新得到的公式称为**生成公式**。 □

例 3.2.1 在前面列出的规则 (or) 中，消解公式为 $\varphi_1 \vee \varphi_2$，生成公式为 φ_1 或者 φ_2。而在规则 (and) 中，消解公式为 $\varphi_1 \wedge \varphi_2$，生成公式有两个— φ_1 以及 φ_2。而在规则 (modal) 中，Γ 中每个形如 $\bigcirc\psi_i$ 的公式都是消解公式，Γ' 中的 ψ_i 即为 $\bigcirc\psi_i$ 对应的生成公式。 □

下面定义 ETL_l "公式迁移图" 的概念。

定义 3.2.3 ($\mathbf{ETL}_l$ 公式迁移图)

任给 ETL_l 公式 φ，其**公式迁移图** 是一个三元组 $\mathcal{G}_\varphi = \langle\mathcal{V}_\varphi, \mathcal{E}_\varphi, \Gamma_0\rangle$，其中:

- $\mathcal{V}_\varphi$ 是该迁移图**节点集**。$\mathcal{V}_\varphi$ 中的每个节点 Γ 是一个有穷的 ETL_l 公式集。
- $\mathcal{E}_\varphi \subseteq \mathcal{V}_\varphi \times \mathcal{V}_\varphi$ 是该迁移图的**边集**。其中，$(\Gamma, \Gamma') \in \mathcal{E}_\varphi$ 当且仅当存在某重写规则，使得 Γ' 可由 Γ 经该重写规则得到。
- $\Gamma_0 = \{\varphi\} \in \mathcal{V}_\varphi$，是该迁移图的**初始节点**。

形如 $\{\phi_1, \ldots, \phi_m, \bigcirc\psi_1, \ldots, \bigcirc\psi_k\}$ 的节点称为**模态节点**，其中每个 ϕ_i 都是文字。对于节点 $\Gamma_1, \Gamma_2 \in \mathcal{V}_\varphi$，称 Γ_1 **可到达** Γ_2，当且仅当:

1. $(\Gamma_1, \Gamma_2) \in \mathcal{E}_\varphi$; 或者

2. 存在 $\Gamma_3 \in \mathcal{V}_\varphi$，使得 $(\Gamma_1, \Gamma_3) \in \mathcal{E}_\varphi$ 且 Γ_3 可到达 Γ_2。

特别的，当 $(\Gamma, \Gamma') \in \mathcal{E}_\varphi$ 时，则称 Γ **可直接到达** Γ'。在公式迁移图中，要求节点互不相同，并且任意节点都是"初始可达" 的。即: 对任意的 $\Gamma \in \mathcal{V}_\varphi$，$\Gamma_0$ 可到达 Γ。模态节点 (唯一的) 可直接到达的节点称为**状态节点**。 □

例 3.2.2 设 $NLW\ \mathcal{A}^{q_1} = \langle\{a_1, a_2\}, \{q_1, q_2\}, \delta, q_1, -\rangle$。其中，$\delta(q_1, a_1) = \{q_1, q_2\}$，$\delta(q_2, a_1) = \{q_1\}$，$\delta(q_2, a_2) = \{q_2\}$，$\delta(q_1, a_2) = \emptyset$。令 $\varphi = \mathcal{A}^{q_1}(p_1 \wedge p_2, \neg p_3)$，其中，$p_1, p_2, p_3 \in AP$。则 φ 的公式迁移图如图 *3.1* 所示。 □

与通常的图一样，公式迁移图也可以定义“路径”、“回路” 的概念。

定义 3.2.4 (公式迁移图中的路径和回路)

给定 ETL_l 公式 φ 及其公式迁移图 $\mathcal{G}_\varphi = \langle \mathcal{V}_\varphi, \mathcal{E}_\varphi, \{\varphi\}\rangle$，则 $\mathcal{G}_\varphi$ 中的**一条路径** P 是一个节点序列 $\Gamma_1, \Gamma_2, \ldots$，其中每个 $(\Gamma_i, \Gamma_{i+1}) \in \mathcal{E}_\varphi$。

称路径 $\Gamma_1, \Gamma_2, \ldots$ 是**完全的**，如果:

1. $\Gamma_1 = \{\varphi\}$；并且
2. 该路径或者为无穷节点序列或者以节点 $\emptyset$ 结尾。

如果 $\Gamma_1 = \Gamma_m$，则称路径 $\Gamma_1, \ldots, \Gamma_m$ 是**一条回路**。 □

对于模态节点 $\{\phi_1, \ldots, \phi_m, \bigcirc\psi_1, \ldots, \bigcirc\psi_k\}$，当 $k = 0$ 时，则可由 (modal) 规则生成节点 $\emptyset$。

定义 3.2.5 (极大连通子图) 给定 ETL_l 公式 φ，其公式迁移图 $\langle \mathcal{V}_\varphi, \mathcal{E}_\varphi, \{\varphi\}\rangle$ 中的一个**连通子图** 是一个节点集 $\mathcal{S} \subseteq \mathcal{V}_\varphi$。其中，对任意的 $\Gamma_1, \Gamma_2 \in \mathcal{S}$，$\Gamma_1$ 和 Γ_2 之间相互可到达。特别的，如果 $\mathcal{G}_\varphi$ 中不存在连通子图 $\mathcal{S}'$，使得 $\mathcal{S} \subset \mathcal{S}'$，则称 $\mathcal{S}$ 是 $\mathcal{G}_\varphi$ 的**极大连通子图**。 □

注意，这里定义“子图” 时，只涉及节点，而未涉及边。这是因为边集（即节点间的可达关系）可以由其所在的迁移图的边集压缩至该子图的所在节点集上唯一确定。

引理 3.2 公式迁移图的任何回路中必包含模态节点。

证明. 对每个公式 ψ，按如下方式对其赋予一个自然数 $\mathbf{Rk}(\psi)$:

- 令 $\mathbf{Rk}(true) = \mathbf{Rk}(false) = 0$。
- 若 $p \in AP$ 则 $\mathbf{Rk}(p) = \mathbf{Rk}(\neg p) = 0$。
- $\mathbf{Rk}(\bigcirc\psi') = 0$; $\mathbf{Rk}(\neg\bigcirc\psi') = 1$。
- $\mathbf{Rk}(\varphi_1 \wedge \varphi_2) = \mathbf{Rk}(\varphi_1 \vee \varphi_2) = \mathbf{Rk}(\varphi_1) + \mathbf{Rk}(\varphi_2) + 1$。
- $\mathbf{Rk}(\neg(\varphi_1 \wedge \varphi_2)) = \mathbf{Rk}(\neg\varphi_1 \vee \neg\varphi_2) + 1$; $\mathbf{Rk}(\neg(\varphi_1 \vee \varphi_2)) = \mathbf{Rk}(\neg\varphi_1 \wedge \neg\varphi_2) + 1$。
- $\mathbf{Rk}(\neg\neg\psi') = \mathbf{Rk}(\psi') + 1$。
- 若 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, -\rangle$，则

$$\mathbf{Rk}(\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)) = \mathbf{Rk}(\bigvee_{1\le k\le n} (\varphi_k \wedge \bigvee_{q'\in\delta(q,a_k)} \bigcirc\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n))) + 1;$$

$$\mathbf{Rk}(\neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)) = \mathbf{Rk}(\bigwedge_{1\le k\le n} (\neg\varphi_k \vee (\varphi_k \wedge \bigwedge_{q'\in\delta(q,a_k)} \bigcirc\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)))) + 1。$$

容易证明，$\mathbf{Rk}$ 是一个良定义的函数。现将该函数按如下提升，将其定义域扩展为

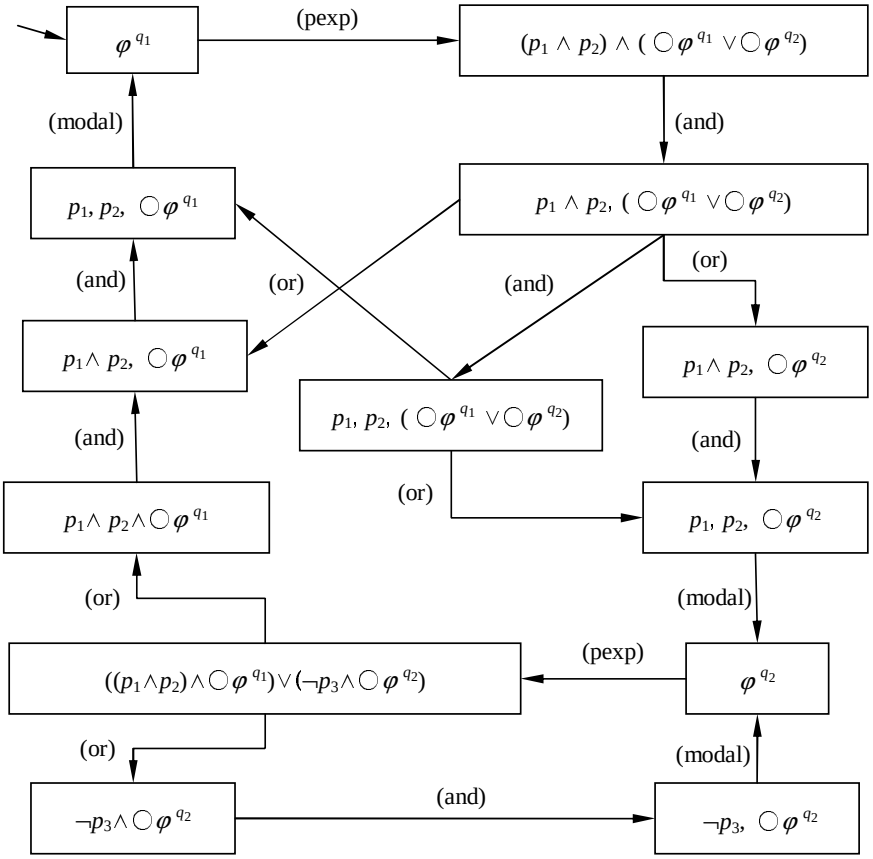


图 3.1 ETL_l 公式迁移图示例

公式集，即：

$$\mathbf{Rk}(\Gamma) \stackrel{\text{def}}{=} \sum_{\psi \in \Gamma} \mathbf{Rk}(\psi) \tag{3.1}$$

那么，在关于 φ 的公式迁移图 $\langle \mathcal{V}_\varphi, \mathcal{E}_\varphi, \{\varphi\} \rangle$ 中，若 $(\Gamma, \Gamma') \in \mathcal{E}_\varphi$，且由 Γ 获得 Γ' 的规则不是 (modal)，则必有 $\mathbf{Rk}(\Gamma) > \mathbf{Rk}(\Gamma')$ 成立。因此，对任意回路 $P = \Gamma_1, \ldots, \Gamma_m$，若其中不包含模态节点，则 $\mathbf{Rk}(\Gamma_1) > \mathbf{Rk}(\Gamma_m)$，但这与 $\Gamma_1 = \Gamma_m$ 矛盾。 □

为了捕捉路径的内部结构，现在引入“踪迹” 的概念。

定义 3.2.6 (路径中的踪迹) 设 $P = \Gamma_1, \Gamma_2, \ldots,$ 是公式迁移图 $\mathcal{G}_\varphi$ 中的一条路径。称公式序列 $\tau = \phi_1 \phi_2 \ldots$ 是 P 中的一条**踪迹**，如果：$\phi_i \in \Gamma_i$，并且当 Γ_{i+1} 在 P 中存在时，ϕ_{i+1} 在 τ 中存在，同时：

- 若 ϕ_i 是 Γ_i 中的消解公式，则 ϕ_{i+1} 是由 Γ 得到 Γ' 时 ϕ_i 对应的生成公式。
- 若 ϕ_i 不是 Γ_i 中的消解公式，则 $\phi_{i+1} = \phi_i$。 □

现在，就可以定义 ETL_l 公式迁移图路径的“一致性” 了。

定义 3.2.7 ($\mathbf{ETL}_l$ 公式迁移图路径一致性) 给定 ETL_l 公式 φ，设 $\mathcal{G}_\varphi$ 为其公式迁

移图，P 是 $\mathcal{G}_\varphi$ 中的路径。如果 P 满足:

局部一致性：P 的任何一个节点中不含 $false$ 或者互补对（比如，公式 ψ 和 $\neg\psi$ 就是一对互补对）;

全局一致性：对任意的自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 以及 $\mathcal{A}$ 中的状态 q，公式 $\neg\varphi^q$ 在 P 中的任意踪迹 τ 中只出现有穷多次

则称 P 满足**一致性**。

称公式迁移图 $\mathcal{G}_\varphi$ 是**一致**的，当且仅当 $\mathcal{G}_\varphi$ 中存在一条满足一致性的完全路径。□

下面的两条定理刻画了 ETL_l 公式可满足性和其迁移图一致性之间的关系。

定理 3.3 若 ETL_l 公式 φ 的公式迁移图 $\mathcal{G}_\varphi$ 是一致的，则 φ 是可满足的。

证明. 由定义，$\mathcal{G}_\varphi$ 中必存在满足一致性的完全路径 $P = \Gamma_0, \Gamma_1, \ldots$，其中 $\Gamma_0 = \{\varphi\}$。

令 Γ_{M_i} 为 P 中第 i 个模态节点（注意 i 从 0 开始）。同时，令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目。于是，$\Gamma_{M_{\gamma(i)}}$ 恰为出现在 Γ_i 后（包含 Γ_i 在内）的第一个模态节点。显然，当 Γ_i 本身为模态节点时，二者相同。同时，若 $i \le j \le M_{\gamma(i)}$，则 $\gamma(i) = \gamma(j)$。

设 π 是某个满足约束“ 对于任意的 $i \in \mathbb{N}$，若 Γ_{M_i} 存在，则 $\pi(i) \models \bigwedge(\Gamma_{M_i} \cap (AP \cup \overline{AP}))$ ” 线性结构。（这里的 “$\models$” 是布尔公式的满足关系，见定义 2.1.6）— 由于 P 满足局部一致性，因而每个 $\pi(i)$ 都存在。现在归纳证明：对于任意的 $\psi \in \Gamma_i$，都有 $\pi, \gamma(i) \models \psi$。

- 由于 P 满足局部一致性，所以 $\psi \neq false$；此外，当 $\psi = true$ 时结论显然。
- 当 $\psi \in AP \cup \overline{AP}$ 时，ψ 会保留至 $\Gamma_{M_{\gamma(i)}}$ 中。由 π 的构造知，$\pi, \gamma(i) \models \psi$。
- 当 $\psi = \varphi_1 \wedge \varphi_2$ 时，存在 $i < j < M_{\gamma(i)}$，使得 $\varphi_1 \in \Gamma_j$，$\varphi_2 \in \Gamma_j$（针对 ψ 使用 (and) 规则后）。由归纳假设知，$\pi, \gamma(j) \models \varphi_1$ 且 $\pi, \gamma(j) \models \varphi_2$。再由 $\gamma(j) = \gamma(i)$ 知 $\pi, \gamma(i) \models \psi$。
- 类似的，也可以证明当 $\psi = \varphi_1 \vee \varphi_2$、$\psi = \neg(\varphi_1 \wedge \varphi_2)$、$\psi = \neg(\varphi_1 \vee \varphi_2)$ 或者 $\psi = \neg\neg\psi'$ 时的情况。
- 若 $\psi = \bigcirc\psi'$，则 ψ 会保留至 $\Gamma_{M_{\gamma(i)}}$ 中。使用 (modal) 规则后，必然有 $\psi' \in \Gamma_{M_{\gamma(i)}+1}$。易证：$\gamma(M_{\gamma(i)} + 1) = \gamma(i) + 1$。于是，由归纳假设有 $\pi, \gamma(i) + 1 \models \psi'$。从而 $\pi, \gamma(i) \models \psi$。
- 若 $\psi = \neg\bigcirc\psi'$，则必然存在 $i < j \le M_{\gamma(i)}$，使得 $\bigcirc\neg\psi' \in \Gamma_j$。于是，使用 (modal) 规则后，必然有 $\neg\psi' \in \Gamma_{M_{\gamma(i)}+1}$。同前所述，依归纳假设有 $\pi, \gamma(i) + 1 \models \neg\psi'$。从而 $\pi, \gamma(i) \models \psi$。
- 若 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，且 $\mathcal{A} = \langle\{a_1, \ldots, a_n\}, Q, \delta, q, -\rangle$，则经使用 (pexp)、(or)、

以及 (and) 后，存在 $1 \leq k_0 \leq n$ 及 $i < j \leq M_{\gamma(i)}$ 以及 $q_1 \in Q$，满足：

1. $q_1 \in \delta(q, a_{k_0})$；
2. $\varphi_{k_0} \in \Gamma_j$；
3. $\bigcirc\psi^{q_1} = \bigcirc\mathcal{A}^{q_1}(\varphi_1, \ldots, \varphi_n) \in \Gamma_{M_{\gamma(i)}}$。

由归纳假设，$\pi, \gamma(i) \models \varphi_{k_0}$ 成立。同时，在对 $\Gamma_{M_{\gamma(i)}}$ 施加规则 (modal) 后，必然有 $\psi^{q_1} \in \Gamma_{M_{\gamma(i)}+1}$。重复讨论此过程，便可以得到一个无穷字 $a_{k_0}, a_{k_1}, \ldots \in \mathbf{L}(\mathcal{A}^q)$，并且对每个 $l \in \mathbb{N}$，都有 $\pi, \gamma(i) + l \models \varphi_{k_l}$。由定义有 $\pi, \gamma(i) \models \psi$。

- 若 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 且 $\mathcal{A}^q = \langle\{a_1, \ldots, a_n\}, Q, \delta, q, -\rangle$，则经使用 (nexp)、(and) 以及 (or) 后，对每个 $1 \leq k \leq n$ 而言：
  1. 或者存在某个 $i < j \leq M_{\gamma(i)}$，使得 $\neg\varphi_k \in \Gamma_j$；
  2. 或者存在某个 $i < j \leq M_{\gamma(i)}$，使得 $\varphi_k \in \Gamma_j$，并且对每个 $q' \in \delta(q, a_k)$ 而言，有 $\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n) \in \Gamma_{M_{\gamma(i)}+1}$。

  对前一种情况，由归纳假设得 $\pi, \gamma(i) \models \neg\varphi_k$。对后一种情况，有 $\pi, \gamma(i) \models \varphi_k$ 成立，且对于每个 $q' \in \delta(q, a_k)$，在路径 $\Gamma_i, \ldots, \Gamma_{M_{\gamma(i)}}$ 中存在踪迹 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$,
  $\ldots, \bigcirc\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$。同时，在使用 (modal) 规则后，每个 $\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$ 都会出现在 $\Gamma_{M_{\gamma(i)}+1}$ 中。重复上述过程，便可以得到一棵 Q-标记树 $\langle T, \rho\rangle$，它满足：
  1. $\rho(\epsilon) = q$；
  2. 对每个 $x \in T$ 及 $1 \leq k \leq n$ 而言，若 $\pi, \gamma(i) + |x| \models \varphi_k$，则对每个 $q' \in \delta(\rho(x), a_k)$，存在 $c \in \mathbb{N}$，使得 $x \cdot c \in T$ 且 $\rho(x \cdot c) = q'$。

  此外，T 中的每条路径都唯一对应于 P 中的一条踪迹。由于 P 满足全局一致性，所以 T 必然是一棵有穷树。于是，$\langle T, \rho\rangle$ 必然是 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在 π 上起始于 $\gamma(i)$ 的一个拒绝例证（见定义 2.2.24）。由定理 2.9 知 $\pi, \gamma(i) \models \psi$。

易知，上述归纳是完全的。特别的，当取 $i = 0$ 时，因为 $\Gamma_0 = \{\varphi\}$，$\gamma(0) = 0$。所以，$\pi \models \varphi$ 成立。 □

定理 3.4 *若 ETL_l 公式 φ 是可满足的，则 φ 的公式迁移图 $\mathcal{G}_\varphi$ 必然满足一致性。*

证明. 由于 φ 是可满足的，所以存在线性结构 π 使得 $\pi \models \varphi$。现在，根据 π 构造一个 $\mathcal{G}_\varphi$ 中的节点序列 $\Gamma_0, \Gamma_1, \ldots$ 如下。

- 令 $\Gamma_0 = \{\varphi\}$。同时，对每个 Γ_i，仍然令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目。由于 $\gamma(0) = 0$，于是，由假设知 $\pi, \gamma(0) \models \varphi = \bigwedge\Gamma_0$。
- 对当前节点 Γ_i，假设 $\pi, \gamma(i) \models \bigwedge\Gamma(i)$ 成立。则按照下列方式获得下一个节点

Γ_{i+1}。

1. 若 Γ_i 是模态节点，则 Γ_{i+1} 是由 Γ_i 经规则 (modal) 得到的节点。
2. 若 Γ_i 不是模态节点，则任从 Γ_i 选取不是文字的公式 ψ 作为消解公式。当消解公式选定后，能够采用的重写规则随之确定。当 ψ 不是形如 $\varphi_1 \vee \varphi_2$ 的公式时，令 Γ_{i+1} 是由 Γ_i 经该重写规则得到的节点。当 $\psi = \varphi_1 \vee \varphi_2$ 时，则必有某个 φ_k（其中，$k \in \{1,2\}$）满足 $\pi, \gamma(i) \models \varphi_k$。这时，令 $\Gamma_{i+1} = \Gamma_i \backslash \{\psi\} \cup \{\varphi_k\}$，则 Γ_{i+1} 可由 Γ_i 经规则 (or) 得到。

在第一种情况下，$\gamma(i+1) = \gamma(i)+1$。由 $\pi, \gamma(i) \models \bigwedge \Gamma_i$ 可得 $\pi, \gamma(i)+1 \models \bigwedge \Gamma_{i+1}$。也就是 $\pi, \gamma(i+1) \models \bigwedge \Gamma_{i+1}$。在第二种情况下，$\gamma(i+1) = \gamma(i)$，并且由重写规则以及 Γ_i 的构造过程知 $\pi, \gamma(i) \models \bigwedge \Gamma_i$ 成立蕴含 $\pi, \gamma(i+1) \models \bigwedge \Gamma_{i+1}$ 成立。

- 若当前的节点 $\Gamma_i = \emptyset$，则停止构造；否则，按照上一步骤构造下一节点 Γ_{i+1}。这样，该序列或者终止于 $\emptyset$，或者长度无穷。

由于 Γ_0 是 $\mathcal{G}_\varphi$ 中的初始节点，并且每个 $(\Gamma_i, \Gamma_{i+1}) \in \mathcal{E}_\varphi$。因而，该节点序列必然是 $\mathcal{G}_\varphi$ 中的一条完全路径。

由于对每个 Γ_i 都有 $\pi, \gamma(i) \models \bigwedge \Gamma_i$ 成立，所以 Γ_i 中不可能含有 $false$ 或者互补对。因此该路径必然满足局部一致性。现在证明该路径同时满足全局一致性。

用反证法：假设该路径中存在某条无穷踪迹 τ，使得公式 $\neg\psi^q = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在其中出现无穷多次。其中，NLW $\mathcal{A}^q = \langle\{a_1, \ldots, a_n\}, Q, \delta, q, -\rangle$。不妨设 $\tau = \phi_i\phi_{i+1}\ldots$，其中 $\phi_i = \neg\varphi^q \in \Gamma_i$。令 $q_0 = q$，则在使用规则 (nexp)、(and)、(or) 后，第一次使用 (modal) 规则前，必然有某个 $1 \leq k_0 \leq n$ 以及 $q_1 \in \delta(q_0, a_0)$ 使得 $\varphi_{k_0} \wedge \bigcirc\neg\psi^{q_1}$ 出现在 τ 中，由路径的构造过程知：$\pi, \gamma(i) \models \varphi_{k_0}$。并且，在使用(modal) 后，保留在 τ 中的公式必然是 $\neg\psi^{q_1}$（否则，$\neg\psi^q$ 将不会再次出现在该踪迹中）。重复此讨论，便可以得到一个无穷字 $w = a_{k_0}, a_{k_1}, \ldots \in \mathbf{L}(\mathcal{A}^q)$ 使得对每个 $j \in \mathbb{N}$，有 $\pi, \gamma(i)+j \models \varphi_{k_j}$。按照定义，有 $\pi, \gamma(i) \models \psi^q$。但是，由于 $\neg\psi^q \in \Gamma_i$，而路径的构造过程保证 $\pi, \gamma(i) \models \bigwedge \Gamma_i$ 成立。这样就产生了矛盾!

因此，该路径是满足一致性的路径，从而 $\mathcal{G}_\varphi$ 是一致的。 □

推论 3.5 *ETL_l 公式 φ 是可满足的，当且仅当其公式迁移图 $\mathcal{G}_\varphi$ 是一致的。*

3.2.2 $\mathbf{ETL}_l$ 公理系统及可靠性、完备性

ETL_l 的公理系统 $\mathscr{L}$ 由 4 条公理和 3 条推理规则构成，如表 3.1 所示。

在公理 (Expand) 和推理规则 (Loop) 中，$\mathcal{A}^{q_i} = \langle\Sigma, Q, \delta, q_i, -\rangle$，其中 $\Sigma = \{a_1, \ldots, a_n\}$，$Q = \{q_1, \ldots, q_m\}$。注意，公理 (Expand) 和规则 (Loop) 能够匹配任

表 3.1 ETL_l 的公理系统 $\mathscr{L}$

公 理	
所有的重言式	(Tau)
$\neg\bigcirc\varphi \leftrightarrow \bigcirc\neg\varphi$	(Next)
$\bigcirc(\varphi_1 \to \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \to \bigcirc\varphi_2)$	(Kri)
$\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \leftrightarrow \bigvee\limits_{1\le k\le n}(\varphi_k \wedge \bigvee\limits_{q_j\in\delta(q_i,a_k)} \bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n))$	(Expand)
推 理 规 则	
$\dfrac{\varphi_1;\quad \varphi_1 \to \varphi_2}{\varphi_2}$	(MP)
$\dfrac{\varphi}{\bigcirc\varphi}$	(XGen)
$\dfrac{\bigwedge_{1\le i\le m}(\psi_i \to \bigvee_{1\le k\le n}(\varphi_k \wedge \bigvee_{q_j\in\delta(q_i,a_k)} \bigcirc\psi_j))}{\bigwedge_{1\le i\le m}(\psi_i \to \mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n))}$	(Loop)

意的 NLW 自动机连接子。

例 3.2.3 设 NLW $\mathcal{A}_1 = \langle\{a_1,a_2\},\{q_1,q_2\},\delta_1,q_1,-\rangle$, $\mathcal{A}_2 = \langle\{a_1\},\{q_1\},\delta_2,q_1,-\rangle$, 其中, $\delta_1(q_1,a_1) = \{q_1,q_2\}$; $\delta_1(q_1,a_2) = \{q_2\}$; $\delta_1(q_2,a_1) = \emptyset$; $\delta_1(q_2,a_2) = \{q_1\}$; $\delta_2(q_1,a_1) = \{q_1\}$。$p_1$、$p_2$、$p_3$ 都是原子命题。令 $\varphi_2^{q_1} = \mathcal{A}_2^{q_1}(p_1)$, $\varphi_1^{q_1} = \mathcal{A}_1^{q_1}(\varphi_2^{q_1}, q_2 \wedge q_3)$, 则由 (Expand) 可以得到下列公理:

$$\varphi_2^{q_1} \leftrightarrow (p_1 \wedge \bigcirc\varphi_2^{q_1}) \tag{3.2}$$

$$\varphi_1^{q_1} \leftrightarrow ((\varphi_2^{q_1} \wedge (\bigcirc\varphi_1^{q_1} \vee \bigcirc\varphi_1^{q_2})) \vee ((p_2 \wedge p_3) \wedge \bigcirc\varphi_1^{q_2})) \tag{3.3}$$

$$\varphi_1^{q_2} \leftrightarrow ((p_2 \wedge p_3) \wedge \bigcirc\varphi_1^{q_1}) \tag{3.4}$$

由 (Loop) 则可以得到

$$\frac{\psi_1 \to (p_1 \wedge \bigcirc\psi_1)}{\psi_1 \to \varphi_2^{q_1}} \tag{3.5}$$

等推理规则。 □

定义 3.2.8 (证明) 系统 $\mathscr{L}$ 关于 φ 的一个**证明**是一个有穷公式序列 $\varphi_0, \varphi_1, \ldots, \varphi_k = \varphi$。其中，对每个 φ_i 而言:

- 或者 φ_i 是某公理的实例;
- 或者存在 $i_1 < \ldots < i_j < i$，使得 φ_i 可由 φ_{i_1}，…，φ_{i_j} 及某推理规则得到。

若 $\mathscr{L}$ 中存在关于 φ 的证明，则称 φ 在 $\mathscr{L}$ 中**可证**，记作 $\vdash_{\mathscr{L}} \varphi$。当上下文无歧义时，可以忽略该下标，直接将其记作 $\vdash \varphi$。 □

定理 3.6 ($\mathscr{L}$ 的可靠性) 对 $\mathscr{L}$ 系统而言，若 $\vdash \varphi$，则 $\models \varphi$。

证明. 由 ETL_l 的语义定义（见定义 2.2.22）容易检验每条公理都是有效公式，同时，$\mathscr{L}$ 中的推理规则保持有效性。这里，唯一比较有趣的是 (Loop) 规则的可靠性：假设 NLW $\mathcal{A}^{q_i} = \langle \{a_1, \ldots, a_n\}, \{q_1, \ldots, q_m\}, \delta, q_i, - \rangle$，且前件可证出，即

$$\bigwedge_{1 \le i \le m} (\psi_i \to \bigvee_{1 \le k \le n} (\varphi_k \wedge \bigvee_{q_j \in \delta(q_i, a_k)} \bigcirc \psi_j))$$

是永真式，那么，对每个 ψ^i 以及线性结构 π，若 $\pi \models \psi$ 成立，则

1. 令 $i_0 = i$。
2. 必存在某个 $k_0 \in \{1, \ldots, n\}$ 以及 $q_{i_1} \in \delta(q_{i_0}, a_{k_0})$，使得 $\pi, 0 \models \varphi_{k_0}$，且 $\pi, 1 \models \psi_{i_1}$。
3. 同理，也存在某个 $k_1 \in \{1, \ldots, n\}$ 以及 $q_{i_2} \in \delta(q_{i_1}, a_{k_1})$，使得 $\pi, 1 \models \varphi_{k_1}$，且 $\pi, 2 \models \psi_{i_2}$。
4. $\cdots$ $\cdots$（重复此讨论）。

这样，便存在无穷字 $a_{k_0}, a_{k_1}, \ldots \in \mathbf{L}(\mathcal{A}^{q_i})$，其中对每个 $j \in \mathbb{N}$ 都有 $\pi, j \models \varphi_{k_j}$（这里，$k_j \in \{1, \ldots, n\}$）。由定义，$\pi \models \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$。于是，后件

$$\bigwedge_{1 \le i \le m} (\psi_i \to \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n))$$

是永真的。因而，若 $\vdash \varphi$，则 $\models \varphi$。 □

设 $\mathcal{G}_\varphi = \langle \mathcal{V}_\varphi, \mathcal{E}_\varphi, \{\varphi\} \rangle$ 是 φ 对应的迁移图。由公理 (Tau)、(Kri)、(Next) (Expand)，推理规则 (MP)、(XGen) 以及公式迁移图的重写规则，很容易验证下面的引理成立。

引理 3.7 (公式迁移图的推理性质) 对任意的 $\Gamma \in \mathcal{V}_\varphi$，有:

- 若 Γ 是模态节点，则有 $\vdash \bigwedge \Gamma \to \bigvee_{(\Gamma, \Gamma') \in \mathcal{E}_\varphi} \bigcirc \bigwedge \Gamma'$ 成立（事实上，这种情况下 Γ 只有唯一的直接可到达的节点）。

- 若 Γ 不是模态节点，则有 $\vdash \bigwedge\Gamma \leftrightarrow \bigvee_{(\Gamma,\Gamma')\in\mathcal{E}_\varphi} \bigwedge\Gamma'$ 成立。

证明. 对每个 $(\Gamma,\Gamma')\in\mathcal{E}_\varphi$，按由 Γ 得的 $\mathcal{G}'$ 所采用的规则讨论即可得结论。 □

上述性质称为公式迁移图的“**推理性质**”。该性质具有“**删除不变性**”，即：

引理 3.8 (推理性质的删除不变性) *对于任意的 $\Gamma'\in\mathcal{V}_\varphi$，若 $\vdash \neg\bigwedge\Gamma'$，则将 Γ' 在 $\mathcal{G}_\varphi$ 中删除后，“推理性质”仍然成立。*

证明. 对任意的 $\Gamma\in\mathcal{V}_\varphi$，分三种情况讨论。

首先，若 $(\Gamma,\Gamma')\notin\mathcal{E}_\varphi$，则对于 Γ 而言，“推理性质”必然成立。

其次，当 $(\Gamma,\Gamma')\in\mathcal{E}_\varphi$，则当 Γ 是模态节点时，Γ' 是 Γ 唯一可直接到达的节点。因此，在删除 Γ' 之前有：

1. $\vdash \bigwedge\Gamma \to \bigcirc\bigwedge\Gamma'$ ⟅推理性质⟆
2. $\vdash \neg\bigwedge\Gamma'$ ⟅已知⟆
3. $\vdash \bigcirc\neg\bigwedge\Gamma'$ ⟅2 及 (XGen)⟆
4. $\vdash \neg\bigcirc\bigwedge\Gamma'$ ⟅(Next)、3 以及 (MP)⟆
5. $\vdash ((\bigwedge\Gamma \to \bigcirc\Gamma') \wedge \neg\bigcirc\bigwedge\Gamma') \to (\bigwedge\Gamma \to false)$ ⟅(Tau)⟆
6. $\vdash \bigwedge\Gamma \to false$ ⟅ 1、4、5 及 (MP)⟆

在删除 Γ' 后，由于 $\bigvee_{(\Gamma,\Gamma'')\in\mathcal{E}_\varphi} \bigwedge\Gamma'' = \bigvee\emptyset = false$，所以，在这种情况下推理性质对节点 Γ 仍然成立。

最后，就是 $(\Gamma,\Gamma')\in\mathcal{E}_\varphi$，且 Γ 不是模态节点的情况。这时，在 Γ' 被删除前：

1. $\vdash \bigwedge\Gamma \leftrightarrow \bigvee_{(\Gamma,\Gamma'')\in\mathcal{E}_\varphi} \bigwedge\Gamma''$ ⟅推理性质⟆
2. $\vdash \neg\bigwedge\Gamma'$ ⟅已知⟆
3. $\vdash (\bigwedge\Gamma \leftrightarrow \bigvee_{(\Gamma,\Gamma'')\in\mathcal{E}_\varphi} \bigwedge\Gamma'') \wedge (\neg\bigwedge\Gamma') \to (\bigwedge\Gamma \leftrightarrow \bigvee_{\substack{(\Gamma,\Gamma'')\in\mathcal{E}_\varphi \\ \Gamma''\neq\Gamma'}} \bigwedge\Gamma'')$ ⟅(Tau)⟆
4. $\vdash \bigwedge\Gamma \leftrightarrow \bigvee_{\substack{(\Gamma,\Gamma'')\in\mathcal{E}_\varphi \\ \Gamma''\neq\Gamma'}} \bigwedge\Gamma''$ ⟅ 1、2、3 以及 (MP) ⟆

因此，在这种情况下，删除 Γ' 后，节点 Γ 处的“推理性质” 仍然成立。 □

定理 3.9 *对于 $\mathscr{L}$ 系统而言，若 ETL_l 公式 φ 是不可满足的，则 $\vdash\neg\varphi$。*

证明. 设 $\mathcal{G}_\varphi = \langle\mathcal{V}_\varphi,\mathcal{E}_\varphi,\{\varphi\}\rangle$ 是 φ 对应的迁移图。由于 φ 不可满足，由定理 3.3 可知，$\mathcal{G}_\varphi$ 必不满足一致性。

现在，准备采用这样的策略：对于每个 $\Gamma\in\mathcal{V}_\varphi$，若当前 $\neg\bigwedge\Gamma$ 已被证出，则将 Γ 从图中删除。这样，只需证明初始节点 $\{\varphi\}$ 会被删除即可。

在 $\mathcal{G}_\varphi$ 中的极大连通子图构成集合上，可以定义偏序关系 $\preceq_\varphi$ 如下：$\mathcal{S}_1\preceq_\varphi\mathcal{S}_2$ 当且仅当存在 $\Gamma_i\in\mathcal{S}_1$，$\Gamma_2\in\mathcal{S}_2$，并且由 Γ_1 可以到达 Γ_2。关于 $\preceq_\varphi$ 的极大元称为 $\mathcal{G}_\varphi$ 的**末端极大连通子图**。同时，若节点 $\Gamma\in\mathcal{G}_\varphi$ 不在任何连通子图中，则称 Γ 为 $\mathcal{G}_\varphi$ 中的**独立节点**。显然，独立节点一定不带自圈（即：$(\Gamma,\Gamma)\notin\mathcal{E}_\varphi$）。不能到达任何节

点的独立节点称为**末端独立节点**。

对于 $\mathcal{G}_\varphi$ 中的任意一个末端极大连通子图 $\mathcal{S}$，假设存在一条从初始节点 $\{\varphi\}$ 到 $\mathcal{S}$ 中某节点的路径且该路径中的任意节点中不含 $false$ 或互补对（若这样的路径不存在，可先将 $\mathcal{G}_\varphi$ 中含有 $false$ 或互补对的节点删除，这样 $\mathcal{S}$ 将变得非初始可达，从而可以作为一个整体删除）则按照下列方式将 $\mathcal{S}$ 中的所有节点删除。

- 对 $\mathcal{S}$ 当前剩余的节点集中每个包含 $false$ 或者互补对的节点 Γ 而言，由于 $\neg\bigwedge\Gamma$ 直接可由 (Tau) 证得，所以可以将 Γ 直接从当前节点集合中删除。
- 对于 $\mathcal{S}$ 中的任意一个末端孤立节点 Γ，因为 $\mathcal{G}_\varphi$ 不一致，可以断言 $\Gamma\neq\emptyset$。并且，原来由 Γ 可直接到达的节点都已被删除。
  1. 若 Γ 是模态节点，则由引理 3.8 知：$\bigwedge\Gamma\rightarrow false$ 可证，这等价于 $\neg\bigwedge\Gamma$。
  2. 若 Γ 不是模态节点，设 $\Gamma_1,\ldots,\Gamma_m$ 是最初在 $\mathcal{G}_\varphi$ 中由 Γ 直接可达的节点，则由 $\vdash\bigwedge\Gamma\leftrightarrow\bigvee_{1\leq i\leq m}\bigwedge\Gamma_i$，及对每个 $1\leq i\leq m$ 有 $\vdash\neg\bigwedge\Gamma_i$（因为每个 Γ_i 已被删除）。由 (Tau) 公理和 (MP) 规则可得 $\vdash\neg\bigwedge\Gamma$。

  这两种情况都意味着 Γ 也可删除。
- 同时，删除当前非初始可达的节点，也不会破坏剩余部分的“推理性质”的成立。
- 若 $\mathcal{S}$ 的当前剩余节点集中无末端独立节点，并且所有节点均满足局部一致性，则必然存在一条无穷路径 P。由于任何节点都是初始可达的，因此 $\mathcal{G}_\varphi$ 中必然存在一条以 P 为后缀的无穷路径，所以 P 必不满足全局一致性。这样， P中一定存在一条无穷踪迹 τ，同时存在某个形如 $\neg\mathcal{A}^{q_l}(\varphi_1,\ldots,\varphi_n)$ 的公式在 τ 出现无穷多次。这里，不妨设 $\mathcal{A}^{q_l}=\langle\{a_1,\ldots,a_n\},\{q_1,\ldots,q_m\},\delta,q_l,-\rangle$。

  对 $\mathcal{S}$ 中的每个节点 Γ，由定义，τ 在每次经过 Γ 时，都会取其中的某个公式（注意：τ 的不同次经过 Γ 所取的公式可能会不同）。这里，用 $\Gamma_{[\tau]}$ 表示由 τ 在各次经过 Γ 时所取的公式构成的集合。

  由于 P 为无穷路径，其中必然包含回路。由引理 3.2，P 中一定包含模态节点，从而也包含状态节点。于是，对每个 $1\leq i\leq m$，令

$$\mathcal{S}_i=\{\Gamma\mid\Gamma\text{ 为}\mathcal{S}\text{ 中的状态节点，且 }\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\in\Gamma_{[\tau]}\}$$

以及

$$\psi_i=\bigvee_{\Gamma\in\mathcal{S}_i}\bigwedge\Gamma \tag{3.6}$$

则容易证明至少有一个 $\mathcal{S}_i$ 非空，并且 $\mathcal{S}$ 中的任一状态节点必属于某个 $\mathcal{S}_i$。由推理性质以及 (nexp) 重写规则，对每个 $1 \leq i \leq m$ 可以得到

$$\vdash \psi_i \rightarrow \bigvee_{1 \leq k \leq n} (\varphi_k \wedge \bigvee_{q_j \in \delta(q_k, a_i)} \bigcirc \psi_j) \tag{3.7}$$

这是由于 $\neg \mathcal{A}^{q_l}(\varphi_1, \ldots, \varphi_n)$ 在 τ 中出现无穷多次，所以每次对公式 $\neg \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$ 施加 (nexp) 规则时，必然有某个 $1 \leq k \leq n$ 使得析取支

$$\varphi_k \wedge \bigwedge\nolimits_{q_j \in \delta(q_i, a_k)} \bigcirc \neg \mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$$

保留在 τ 中。由 (3.7) 以及 (Loop) 规则，可以得到

$$\vdash \psi_i \rightarrow \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n) \tag{3.8}$$

另一方面，由于$\neg \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$ 出现于 $\mathcal{S}_i$ 中的每个节点中，由 (Tau) 有

$$\vdash \psi_i \rightarrow \neg \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n) \tag{3.9}$$

成立。这样，由 (3.8)、(3.9)、(Tau) 公理以及 (MP) 规则，立即可以得到 $\vdash \neg \psi_i$。换言之，对 $\mathcal{S}_i$ 中的每个节点 Γ，都有 $\vdash \neg \bigwedge \Gamma$。由于 $\bigcup_{1 \leq i \leq m} \mathcal{S}_i$ 覆盖了 P 中所有的状态节点，因而 P 中的每个状态节点都可以删除。这意味着 P 中的所有模态节点都会被删除。

交替利用上面的过程，可以将 $\mathcal{S}$ 中全部节点删除。类似的，可以将其他极大连通子图以及末端独立节点/不可达节点删除，直至整个图的节点集变为空。这时，由于初始节点 $\{\varphi\}$ 也被删除，于是有 $\vdash \neg \varphi$。 □

定理 3.10 ($\mathscr{L}$ 的完备性) 对 $\mathscr{L}$ 系统而言，若 $\models \varphi$，则 $\vdash \varphi$。

证明. 若 $\models \varphi$，则 $\neg \varphi$ 是不可满足的。于是：

1. $\vdash \neg\neg \varphi$ ⟅ 定理 3.9⟆
2. $\vdash \neg\neg \varphi \rightarrow \varphi$ ⟅(Tau)⟆
3. $\vdash \varphi$ ⟅1、2 以及 (MP)⟆

□

3.3 $\mathbf{ETL}_f$ 的公理系统 $\mathscr{F}$

3.3.1 $\mathbf{ETL}_f$ 重写系统及迁移图

本节介绍 ETL_f 的重写规则、公式迁移图及其性质。

同 ETL_l 相比，ETL_f 重写规则中除包括 59 页中给出的 7 条公共规则外，还包括如下 4 条关于自动机连接子的重写规则（以下，假设 NFW $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, F \rangle$，并且设 $\Sigma = \{a_1, \ldots, a_n\}$，$Q = \{q_1, \ldots, , q_m\}$，$q \in Q$）。

$$\frac{\Gamma, \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, \bigvee_{1 \le k \le n}(\varphi_k \wedge \bigvee_{q' \in \delta(q,a_k)} \bigcirc \mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n))} \quad \text{(pexp\_1)}$$

$$\frac{\Gamma, \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, \bigwedge_{1 \le k \le n}(\neg\varphi_k \vee \bigwedge_{q' \in \delta(q,a_k)} \bigcirc \neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n))} \quad \text{(nexp\_1)}$$

$$\frac{\Gamma, \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, true} \quad \text{(pexp\_2)}$$

$$\frac{\Gamma, \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)}{\Gamma, false} \quad \text{(nexp\_2)}$$

其中，重写规则 (pexp_1)、(nexp_1) 的使用条件是 $q \notin F$；而 (pexp_2)、(nexp_2) 的使用条件是 $q \in F$。

这是因为，当 $q \in F$ 时，如果将 $\mathcal{A}^q$ 看做有穷字上的自动机，则空字 $\epsilon \in \mathbf{L}(\mathcal{A}^q)$。因此，对任意的线性结构 π，都有 $\pi \models \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 成立（见 58 页说明）。

关于 ETL_f **公式迁移图** 及其绝大部分相关概念的定义与 ETL_l 的对应概念定义相同。唯一不同的是“路径一致性” 的概念需要修改如下。

定义 3.3.1 ($\mathbf{ETL}_f$ 公式迁移图路径一致性) 给定 ETL_f 公式 φ，设 $\mathcal{G}_\varphi$ 为其公式迁移图，P 是 $\mathcal{G}_\varphi$ 中的路径。如果 P 满足:

局部一致性：P 的任何一个节点中不含 $false$ 或者互补对;

全局一致性：对任意的自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 以及 $\mathcal{A}$ 中的状态 q，公式 φ^q 在 P 中的任意踪迹 τ 中只出现有穷多次

则称 P 满足**一致性**。

称公式迁移图 $\mathcal{G}_\varphi$ 是**一致**的，当且仅当 $\mathcal{G}_\varphi$ 中存在一条满足一致性的完全路径。 □

同样，对于 ETL_f 公式的迁移图而言，也有如下两个性质成立。

定理 3.11 若 ETL_f 公式 φ 的公式迁移图 $\mathcal{G}_\varphi$ 是一致的，则 φ 是可满足的。

证明. 类似于定理 3.3 的证明。按照定义，$\mathcal{G}_\varphi$ 中必存在满足一致性的完全路径 $P = \Gamma_0, \Gamma_1, \ldots$，并且 $\Gamma_0 = \{\varphi\}$。

仍令 Γ_{M_i} 为 P 中第 i 个模态节点，令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目。以及 π 是某个满足约束：“对于任意的 $i \in \mathbb{N}$ 而言，若 Γ_{M_i} 存在，则 $\pi(i) \models$

$\bigwedge(\Gamma_{M_i} \cap (AP \cup \overline{AP}))$ " 的线性结构。类似的，也可以归纳证明：对于任意的 $\psi \in \Gamma_i$，都有 $\pi, \gamma(i) \models \psi$。

- 当 $\psi \in AP \cup \overline{AP} \cup \{true, false\}$ 或者形如 $\varphi_1 \wedge \varphi_2$、$\varphi_1 \vee \varphi_2$、$\neg(\varphi_1 \wedge \varphi_2)$、$\neg(\varphi_1 \vee \varphi_2)$ 或者 $\bigcirc\psi'$、$\neg\neg\psi'$，$\neg\bigcirc\psi'$ 时，与定理 3.3 中的证明完全相同。
- 当 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 时（其中，NFW $\mathcal{A}^q = \langle\{a_1, \ldots, a_n\}, \{q_1, \ldots, q_m\}, \delta, q, F\rangle$）。若 $q \in F$，则 ψ 等价于 $true$，这时当然有 $\pi, \gamma(i) \models \psi$。否则，可以断言，$\psi$ 经过若干次应用 (pexp_1)、(or)、(and)、(modal) 后，必然存在某个状态序列 $q_0, \ldots, q_{l+1} \in Q^*$ 以及 $k_0, \ldots, k_l \in \{1, \ldots, n\}^*$，其中
  1. $q_0 = q$，$q_{l+1} \in F$，并且每个 $q_{j+1} \in \delta(q_j, a_{k_j})$。
  2. φ_{k_j} 出现在某个介于 $\Gamma_{M_{\gamma(i)+j-1}}$ 和 $\Gamma_{M_{\gamma(i)+j}}$ 之间的节点中。
  3. 对每个 $0 \leq j \leq l+1$，公式 $\bigcirc\psi^{q_{j+1}}$ 出现在节点 $\Gamma_{M_{\gamma(i)+j}}$ 中。

  — 若上述条件不成立，则必有某个 q_j 使得 ψ^{q_j} 无穷次出现在 P 中的某条踪迹中，这会违反全局一致性。

  若将 $\mathcal{A}^q$ 视为有穷字上的自动机，则有 $a_{k_0}, \ldots, a_{k_l} \in \mathbf{L}(\mathcal{A}^q)$。同时，依归纳假设，对每个 $0 \leq j \leq l$ 都有 $\pi, \gamma(i) + j \models \varphi_{k_j}$ 成立。于是，$\pi, \gamma(i) \models \psi$ 在这种情况下也成立。
- 若 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，则仍然按照定理 3.3 中的方式构造 Q-标记树 $\langle T, \rho\rangle$。注意到形如 $\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$ 的公式在 $q' \in F$ 时会被重写为 $false$。而 $\langle T, \rho\rangle$ 的构造过程保证对每个 $x \in T$ 而言，$\neg\mathcal{A}^{\rho(x)}(\varphi_1, \ldots, \varphi_n)$ 必然在 P 中的某个节点内出现。由 P 满足局部一致性可以断言：对任意的 $x \in T$，$\rho(x) \notin F$。因而，$\langle T, \rho\rangle$ 是 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在 π 上起始于位置 $\gamma(i)$ 的一个拒绝例证。于是，由定理 2.9 有 $\pi, \gamma(i) \models \psi$。

于是，$\pi, \gamma(0) \models \varphi$，即：$\pi \models \varphi$。 □

定理 3.12 *若 ETL_f 公式 φ 是可满足的，则 φ 的公式迁移图 $\mathcal{G}_\varphi$ 必然满足一致性。*

证明. 不妨设 $\pi \models \varphi$。同定理 3.4 的证明类似，该证明也是通过证实 $\mathcal{G}$ 中的满足一致性的完全路径的存在性获得，但方法却有很大不同— 在该过程中，对形如 $\varphi_1 \vee \varphi_2$ 的公式应用重写规则 (or) 时析取支的选择策略不再仅仅根据 π 来判断。而且需要在某些公式上添加"**标注**"（Annotation），用以指导某些特殊的析取支的选择，以保证全局一致性。现在，节点序列 $\Gamma_0, \Gamma_1, \ldots$ 构造如下。

首先，令 $\Gamma_0 = \{\varphi\}$。同时，对每个 Γ_i，仍令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目。由于 $\gamma(0) = 0$，于是由已知条件有 $\pi, \gamma(0) \models \varphi = \bigwedge\Gamma_0$。其次，对当前节点 Γ_i，归纳假设 $\pi, \gamma(i) \models \bigwedge\Gamma_i$ 成立。则按照下列步骤构造 Γ_{i+1}。

- 若 Γ_i 是模态节点，则 Γ_{i+1} 是由 Γ_i 经规则 (modal) 得到的节点。
- 对 Γ_i 中的每个形如 $\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$ 的新的生成公式 ψ，(特别的，这里也将 Γ_0 中的 φ 视为新的生成公式，并且 $\mathcal{A}^{q_i}=\langle\Sigma,Q,\delta,q,F\rangle$，其中 Σ 与 Q 分别为 $=\{a_1,\ldots,a_n\}$ 和 $\{q_1,\ldots,q_m\}$)，若 ψ 目前尚无标注，则按照如下方式为其添加一个标注：

  由于 $\pi,\gamma(i)\models\psi$，则必然存在 $a_{k_0},\ldots,a_{k_l}\in\Sigma^*$ 以及 $q_0,q_1,\ldots,q_{l+1}$ 满足：$q_0=q$，$q_{l+1}\in F$，且 $q_{j+1}\in\delta(q_j,a_{k_j})$；同时，对每个 $0\le j\le l$，有 $\pi,\gamma(i)+j\models\varphi_{k_j}$。于是，在 Γ_i 中，将 ψ 的标注设为 $q_0,a_{k_0},\ldots,q_l,a_{k_l},q_{l+1}$。

  同时，不难证明：对每个 $0\le j\le l$，有 $\pi,\gamma(i)+j\models\bigcirc\psi^{q_{j+1}}$ 成立。
- 任从 Γ_i 选取公式 ψ（其中 ψ 不是文字）作为得到 Γ_{i+1} 的消解公式，同时，能够使用的重写规则也唯一确定。

  若 ψ 不含有标注，且 ψ 不是形如 $\varphi_1\vee\varphi_2$ 的公式时，直接令 Γ_{i+1} 为 Γ_i 由该消解公式对应的规则得到的节点。

  若 $\psi=\varphi_1\vee\varphi_2$，但 ψ 在 Γ_i 中无标注，则必然存在 $j\in\{1,2\}$ 使得 $\pi,\gamma(i)\models\varphi_j$。这时，将 φ_j 作为 (or) 规则的生成公式。

  若 ψ 在 Γ_i 中有标注，则可由归纳断言 ψ 必是具有 $\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$、$\bigcirc\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$、$\bigvee_k(\varphi_k\wedge\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n))$、$\varphi_k\wedge\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$ 或者 $\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$ 这几种形式之一的公式。

  1. 若 $\psi=\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$：当 $q\in F$ 时，直接利用重写规则 (pexp_2) 将 ψ 重写为 $true$，不设标注；否则，利用规则 (pexp_1) 将 ψ 展开，并将展开后公式的标注设为 σ。
  2. 若 $\psi=\bigcirc\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$ 则所选用的重写规则必为 (modal)，并且，Γ_{i+1} 中对应的生成公式必为 $\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$。这时，将该公式在 Γ_{i+1} 的标注设为 ψ 在 Γ_i 中的标注。
  3. 设 $\psi=\psi_1\vee\psi_2$，其中 $\psi_1=\varphi_{k_s}\wedge\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$，$\psi_2=\bigvee_{\substack{k\\k\neq k_s}}(\varphi_k\wedge\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n))$，且 ψ 在 Γ_i 中的标注为 $\sigma=q_1,a_{k_1},q_2,\ldots$。则：当 $k_s=k_1$ 时，将 ψ_1 作为 (or) 规则的生成公式，且将 ψ_1 在 Γ_1 中的标注设为 $q_2,a_{k_2},\ldots$；否则，将 ψ_2 作为生成公式，且其在 Γ_{i+1} 中的标注仍为 σ。
  4. 若 $\psi=\varphi_k\wedge\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$，且其在 Γ_i 中的标注为 σ，则采用 (and) 对 ψ 重写后，将新生成的 $\bigvee_j\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$ 在 Γ_{i+1} 中的标注设置为 σ。
  5. 设 $\psi=\psi_1'\vee\psi_2'$，其中 $\psi_1'=\bigcirc\mathcal{A}^{q_s}(\varphi_1,\ldots,\varphi_n)$，$\psi_2'=\bigvee_{\substack{j\\j\neq s}}\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)$，且 ψ 在 Γ_i 中的标注为 σ。则：当$q_1=q_s$ 时，将 ψ_1' 作为 (or) 规则的生成

公式；否则，将 ψ_2' 作为生成公式。这两种情况下，生成公式在 Γ_{i+1} 的标注都设为 σ。

- 在上述过程中，对每个 $\phi \in \Gamma_i \cap \Gamma_{i+1}$，若 ϕ 在 Γ_i 中存在标注 σ，则将 ϕ 在 Γ_{i+1} 中的标注设为（或者重新赋值为）σ。

由重写规则以及标注的构造过程知：$\pi, \gamma(i+1) \models \bigwedge \Gamma_{i+1}$ 仍然成立。此外，若当前节点 $\Gamma_i = \emptyset$，则停止构造。否则，重复上述过程将得到一个 $\mathcal{G}_\varphi$ 中的无穷节点序列。同定理 3.4，可以证明该节点序列必为 $\mathcal{G}_\varphi$ 中的完全路径。同时，由于对每个 i 都有 $\pi, \gamma(i) \models \bigwedge \Gamma_i$，所以每个节点中不可能含有 $false$ 或者互补对。于是，该路经满足局部一致性。此外，标注函数保证了：任何正自动机公式在该路径中的任何踪迹中只有有穷次出现，从而该路经满足全局一致性。由定义，该路径必为满足一致性约束的路径，从而 $\mathcal{G}_\varphi$ 是一致的。 □

由定理 3.11 和定理 3.12，立即可以得到下面的推论。

推论 3.13 *ETL_f 公式 φ 是可满足的，当且仅当公式迁移图 $\mathcal{G}_\varphi$ 是一致的。*

3.3.2 $\mathbf{ETL}_f$ 公理系统及可靠性、完备性

ETL_f 的公理系统 $\mathscr{F}$ 如表 3.2 所示。其中，公理 (Tau)、(Next)、(Kri) 与 $\mathscr{L}$ 中的对应公理相同；规则 (MP) 及 (XGen) 也分别与 $\mathscr{L}$ 中对应的规则相同。

在公理 (Expand)、(Acc) 以及规则 (Fin) 中的 NFW $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, F \rangle$，其中 $\Sigma = \{a_1, \ldots, a_n\}$，$Q = \{q_1, \ldots, q_m\}$。此外，使用公理 (Expand) 时，要求 $q \notin F$；使用公理 (Acc) 时，要求 $q \in F$。

对于公理系统 $\mathscr{F}$ 而言，其“**证明**” 的概念与 $\mathscr{L}$ 中的定义相似。类似的，用 $\vdash_{\mathscr{F}} \varphi$ 表示 φ 可在 $\mathscr{F}$ 中证出。在本节，由于只讨论 $\mathscr{F}$ 系统，故而也将其从下标处忽略，直接记作 $\vdash \varphi$。

定理 3.14 ($\mathscr{F}$ 的可靠性) *对 $\mathscr{F}$ 系统而言，若 $\vdash \varphi$，则 $\models \varphi$。*

证明. 同定理 3.6 的证明过程：只需要检验 $\mathscr{F}$ 中每条公理都是有效公式，并且每条推理规则都保持有效性即可。大多数公理、规则是容易检验的，这里只简要的说明规则 (Fin) 对有效性的保持。假设该规则的前件成立，即：对每个 $1 \le i \le m$，公式

$$\psi_i \to \bigwedge_{1 \le k \le n} (\varphi_k \to \bigwedge_{q_j \in \delta(q_i, a_k)} \bigcirc \psi_j) \tag{3.10}$$

以及

$$\psi_i \to \bigwedge_{\delta(q_i, a_l) \cap F \neq \emptyset} \neg \varphi_l \tag{3.11}$$

表 3.2 ETL_f 的公理系统 $\mathscr{F}$

公 理

(Tau)、(Next)、(Kri) 同 $\mathscr{L}$ 中的对应公理。

$$\mathcal{A}^q(\varphi_1,\ldots,\varphi_n) \tag{Acc}$$

$$\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \leftrightarrow \bigvee_{1\le k\le n}(\varphi_k \wedge \bigvee_{q_j\in\delta(q_i,a_k)} \bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)) \tag{Expand}$$

推 理 规 则

(MP)、(XGen) 同 $\mathscr{L}$ 中对应的规则。

$$\frac{\bigwedge_{\substack{1\le i\le m\\ q_i\notin F}}(\psi_i \to (\bigwedge_{1\le k\le n}(\varphi_k \to \bigwedge_{q_j\in\delta(q_i,a_k)}\bigcirc\psi_j) \wedge \bigwedge_{\delta(q_i,a_l)\cap F\neq\emptyset}\neg\varphi_l))}{\bigwedge_{\substack{1\le i\le m\\ q_i\notin F}}(\psi_i \to \neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n))} \tag{Fin}$$

是有效的。那么，对每个 $1\le i\le m$，以及任意的线性结构 π 而言，若 $q_i\notin F$，且 $\pi\models\psi_i$，则按照下列方式构造一棵 Q-标记树 $\langle T,\rho\rangle$：

- 将 $\rho(\epsilon)$ 设为 q_i。由假设有 $\pi,|\epsilon|\models\psi_i$。
- 对于任意的 $x\in T$，归纳假设“若 $\rho(x)=q_j$，则 $\pi,|x|\models\psi_j$” 成立。

  令 $K_x=\{k\mid \pi,|x|\models\varphi_k\}$；$Q_x=\{q_l\in Q\mid$ 存在 $k\in K_x$ 使得 $q_l\in\delta(q_j,a_k)\}$。现在，对每个 $q_l\in Q_x$，为 x 添加一个子节点 y，且将 $\rho(y)$ 设为 q_l。因为 $|y|=|x|+1$，所以由 (3.10) 得 $\pi,|y|\models\psi_l$。同时，由 (3.11) 得 $q_l\notin F$。

于是，$\langle T,\rho\rangle$ 必然是 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 在 π 上起始于位置 0 的一个拒绝例证。因而 $\pi\models\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$。这说明，对每个 $1\le i\le m$ 而言，后件

$$\psi_i \to \neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \tag{3.12}$$

也是有效的。□

同样，对于 ETL_f 公式的迁移图，也可以证明其 “推理性质”（见引理 3.7）以及该性质的 “删除不变性”（见引理 3.8）。

此外，对于$\mathscr{F}$ 而言，采用“迁移图删除” 技术证明其完备性时，还需要用到下列定义和定理。

定义 3.3.2 (自动机连接子嵌套深度) 对任意的 ETL_f 公式 φ，归纳定义其**自动机连接子嵌套深度** $\mathbf{Dep}(\varphi)$ 如下:

- 若 $\varphi \in AP$，则 $\mathbf{Dep}(\varphi) = 0$。
- 若 $\varphi = \neg\psi$，则 $\mathbf{Dep}(\varphi) = \mathbf{Dep}(\psi)$。
- 若 $\varphi = \bigcirc\psi$，则 $\mathbf{Dep}(\varphi) = \mathbf{Dep}(\psi)$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$，则 $\mathbf{Dep}(\varphi) = \max\{\mathbf{Dep}(\varphi_1), \mathbf{Dep}(\varphi_2)\}$。
- 若 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，则 $\mathbf{Dep}(\varphi) = \max\{\mathbf{Dep}(\varphi_1), \ldots, \mathbf{Dep}(\varphi_n)\} + 1$。 □

以下，为方便起见，对路径和踪迹沿用字符串中的记号：

1. 设路径 $P = \Gamma_0, \Gamma_1, \ldots$，则令 $P(i) = \Gamma_i$，$P[i,j] = \Gamma_i, \ldots, \Gamma_j$，$P[i..] = \Gamma_i, \Gamma_{i+1}, \ldots$。若有穷路径 $P' = \Gamma'_0, \ldots, \Gamma'_k$，则用 $P' \cdot P$ 表示路径 $\Gamma'_0, \ldots, \Gamma'_k, \Gamma_0, \Gamma_1, \ldots$。
2. 设踪迹 $\tau = \phi_0, \phi_1, \ldots$，则令 $\tau(i) = \phi_i$，$\tau[i,j] = \phi_i, \ldots, \phi_j$，$\tau[i..] = \phi_i, \phi_{i+1}, \ldots$。若有穷踪迹 $\tau' = \phi'_0, \ldots, \phi'_k$，则用 $\tau' \cdot \tau$ 表示踪迹 $\phi'_0, \ldots, \phi'_k, \phi_0, \phi_1, \ldots$。
3. 对于路径 P 以及踪迹 τ，分别用 $|P|$ 和 $|\tau|$ 表示 P 和 τ 的长度。

定义 3.3.3 (踪迹的拓延) 设 $\tau = \phi_0, \ldots, \phi_k$ 和 $\tau' = \phi'_0, \phi'_1, \ldots$ 都是路径 P 中的踪迹。其中，$|\tau|$ 有穷且 $|\tau'| \geq |\tau|$。称 τ' 是 τ 在 P 中的**拓延**（*Expansion*），如果 ϕ_0 与 ϕ'_0 位于 P 的同一节点中，且对于任意的 $0 \leq i < |\tau|$ 都有 $\tau(i) = \tau'(i)$。 □

定义 3.3.4 (基本迁移路径、迁移公平路径) 设 $\mathcal{G}_\varphi = \langle \mathcal{V}_\varphi, \mathcal{E}_\varphi, \{\varphi\}\rangle$，$\mathcal{S}$ 是 $\mathcal{G}_\varphi$ 的连通子图。称 $P = \Gamma_1, \ldots, \Gamma_m$ 是 $\mathcal{S}$ 中的一条**基本迁移路径**，如果每个 $\Gamma_i \in \mathcal{S}$，Γ_1 是状态节点，Γ_m 是模态节点，并且对于任意 $1 \leq j < m$，Γ_j 都不是模态节点。

称无穷路径 P' 是 $\mathcal{S}$ 中的**迁移公平路径**，如果对 $\mathcal{S}$ 中每个基本迁移路径 P，都有 P 在 P' 出现无穷多次。即: 有无穷多个 $i \in \mathbb{N}$，使得对于任意的 $j < |P|$，有 $P(j) = P'(i+j)$。 □

定理 3.15 若 ETL_f 公式 φ 不可满足，$\mathcal{S}$ 是其公式迁移图中的某个末端极大连通子图。并且:

- $\mathcal{S}$ 中所有含有 $false$ 或者互补对的节点已被删除。
- 对于任意的 $\Gamma \in \mathcal{S}$，若其所有直接可达节点被删除，则 Γ 也已被删除。
- 存在从初始节点 $\{\varphi\}$ 到 $\mathcal{S}$ 中某节点的路径，并且该路径中的任何节点不含 $false$ 或者互补对。

那么，一定存在无穷路径 P 以及 P 中的踪迹 τ，以及某个 NFW $\mathcal{A}^q = \langle\{a_1, \ldots, a_n\}, \{q_1, \ldots, q_m\}, \delta, q, F\rangle$，以及 ETL_f 公式 φ_1，…，φ_n，使得:

- P 是 $\mathcal{S}$ 中的迁移公平路径，并且 P 覆盖 $\mathcal{S}'$ 中的每个节点。
- $\psi^{q_i} = \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$ 在 τ 中出现无穷多次。

- 对于每个 $1 \leq j \leq m$，以及 $\mathcal{S}$ 中的每个状态节点 Γ，若 $\psi^{q_j} \in \Gamma_{[\tau]}$（该记号的定义在定理 *3.9* 的证明中给出），则 $\vdash \bigwedge \Gamma \backslash \{\psi^{q_j}\} \rightarrow \bigwedge_{\delta(q_j,a_l)\cap F \neq \emptyset} \neg \varphi_l$。

证明. 用反证法。假设该命题不成立，则在该前提条件下，将结论取反后就有：

“对于任意覆盖 $\mathcal{S}$ 的无穷路径 P，以及 P 中任意的无穷踪迹 τ 以及 NFW $\mathcal{A} = \langle \{a_1, \ldots, a_m\}, Q, \delta, q, F \rangle$，若 $\psi^{q_i} = \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi)$ 在 τ 中无穷多次出现，则存在 $q_j \in Q$，状态节点 $\Gamma \in \mathcal{S}'$，以及 $l \in \{1, \ldots, n\}$，使得 $\psi^{q_j} \in \Gamma_{[\tau]}$，$\delta(q_j, a_l) \cap F \neq \emptyset$，并且 $\nvdash \bigwedge \Gamma \backslash \{\psi^{q_j}\} \rightarrow \neg \varphi_l$。”

下面给出该种假设的反驳。其主要思想是要证明若取反后的结论成立，则 φ 的公式迁移图中必然存在一个满足一致性的完全路径，从而与 φ 不可满足矛盾。

该路径采用分段构造。在给出构造过程之前，首先观察下列性质。

1. 对于任意的踪迹 $\tau = \phi_0, \phi_1, \ldots$，以及对每个 $i < |\tau| - 1$ 都有 $\mathbf{Dep}(\phi_i) \geq \mathbf{Dep}(\phi_{i+1})$。
2. 若存在某正自动机公式 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在踪迹 $\tau = \phi_0, \phi_1, \ldots$ 中无穷多次出现，且设 ϕ_i 是 τ 中第一个等于 ψ 的公式，则对于任意的 $j \geq i$，都有 $\mathbf{Dep}(\phi_j) = \mathbf{Dep}(\psi)$。
3. 给定路径 P，以及 $P[i,j]$ 中始于某正自动机公式 ψ 的有穷踪迹 τ。若 τ 中的最后一个公式位于 $P(j)$ 中，且对于任意的 $l < |\tau|$，都有 $\mathbf{Dep}(\tau(l)) = \mathbf{Dep}(\psi)$，则称 τ 是 $P[i,j]$ 中的**极大自动机踪迹**。对于任意的 $k > j$，可以证明：τ 在 $P[i,k]$ 中的所有拓延中，最多只有一条位于 $P[i,k]$ 中极大自动机踪迹。

上述所有性质均可依 ETL_f 公式重写规则的特点证得。比如，对性质 (1)，只需对每个重写规则检验：“若 $\Gamma \cup \{\phi'\}$ 可由 $\Gamma \cup \{\phi\}$ 经某重写规则获得，则 $\mathbf{Dep}(\phi') \leq \mathbf{Dep}(\phi)$” 即可。性质 (2) 可直接由性质 (1) 推得。对性质 (3)，只需依 $k - j$ 的值进行归纳即可。

令 P_0 是任意一条由若干 $\mathcal{S}'$ 中的节点构成的，且包含极大自动机踪迹的有穷路径— 这样的 P_0 一定存在，这是因为 $\mathcal{S}'$ 中的任何一条无穷路径均不满足全局一致性（注意 φ 不可满足，而 $\mathcal{S}'$ 中的每个节点都不含 $false$ 或互补对），所以必然有某条踪迹中存在某无穷多次出现的正自动机公式。于是 P_0 可以由此无穷路径截取。

令 W_0 为由 P_0 中所有极大自动机踪迹构成的集合。由于 P_0 是有穷路径，所以 W_0 必然是有穷集合。

对每个 $i \in \mathbb{N}$，假设 P_i 与 W_i 已经获得，不妨设 P_i 中的最后一个节点为 Γ，则按照如下步骤获得 P_{i+1} 与 W_{i+1}。

- 若 $W_i = \emptyset$，则令 P 是任意一条由 $\mathcal{S}'$ 中节点构成的起始于节点 Γ 且至少包含一条极大自动机踪迹的有穷路径（这样的路径必然存在）。令 $P_{i+1} = P[1..]$，$W_i + 1$ 为 $P_0 \cdot P_1 \cdot \ldots \cdot P_i \cdot P_{i+1}$ 中所有的极大自动机踪迹。
- 若 $W_i \neq \emptyset$，则任取以 Γ 为起点的且覆盖 $\mathcal{S}$ 中所有节点的迁移公平路径 P（由于 $\mathcal{S}$ 是 $\mathcal{G}_\varphi$ 的连通子图，这样的路径必然存在）。此时，应区分以下两种情况。
- 第一种情况：如果对于 W_i 中的每个踪迹 τ，在路径 $P_0 \cdot P_1 \cdot \ldots \cdot P_i \cdot P[1..]$ 中不存在长度无穷的拓延 τ'，使得
  - 存在某正自动机公式 ψ 在 τ' 中无穷多次出现；
  - 对 τ' 中出现的每个公式 ϕ，有 $\mathbf{Dep}(\psi) = \mathbf{Dep}(\phi)$

  则任取某 $k \geq 1$，令 $P_{i+1} = P[1, k]$，且 $W_{i+1} = \emptyset$ 即可。
- 第二种情况，存在 W_i 中的某个踪迹 τ，以及其在 $P_0 \cdot P_1 \cdot \ldots \cdot P_i \cdot P[1..]$ 中的无穷拓延 τ' 满足上述两个要求。这时，P_{i+1} 的构造过程如下。

  不失一般性，设正自动机公式 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，（其中 $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, F \rangle$，$\Sigma = \{a_1, \ldots, a_n\}$）在踪迹 τ' 中无穷多次出现。因为 P 是一条覆盖 $\mathcal{S}$ 中全部节点的迁移公平路径，由取反后的结论，必然存在 P 中的某个状态节点 Γ'（这里，不妨设 $\Gamma' = P(m)$），以及 $q_j \in Q$，$1 \leq l \leq n$，使得 $\psi^{q_j} \in \Gamma'_{[\tau']}$。同时，$\delta(q_j, a_l) \cap F \neq \emptyset$，且 $\not\vdash \bigwedge(\Gamma' \backslash \{\psi^{q_j}\}) \rightarrow \neg\varphi_l$。不妨设 $q_f \in \delta(q_j, a_l) \cap F$，同时令 $\Gamma'' = \Gamma' \backslash \{\psi^{q_j}\}$。注意：由于 τ' 中含有无穷多次的正自动机公式出现，因而 $q_j \notin F$。否则，ψ^{q_j} 会被重写为 $true$，其任何拓延中都不可能再包含自动机公式。考虑如下的公式序列 $\theta_0 \sim \theta_3$，其中

$$\theta_0 = \theta_1 \vee (\bigvee_{l' \neq l} (\varphi_{l'} \wedge \bigvee_{j' \in \delta(q_j, a_{l'})} \bigcirc \psi^{q_{j'}})) \tag{3.13}$$

$$\theta_1 = \varphi_l \wedge \theta_2 \tag{3.14}$$

$$\theta_2 = \theta_3 \vee \bigvee_{\substack{j' \neq f \\ j' \in \delta(q_j, a_l)}} \bigcirc \psi^{q_{j'}} \tag{3.15}$$

$$\theta_3 = \bigcirc \psi^{q_f} \tag{3.16}$$

再令 $\Gamma_{i,0} \sim \Gamma_{i,3}$ 分别为 $\Gamma'' \cup \{\theta_0\}$、$\Gamma'' \cup \{\theta_1\}$、$\Gamma'' \cup \{\varphi_l, \theta_2\}$ 以及 $\Gamma'' \cup \{\varphi_l, \theta_3\}$。于是，$\Gamma', \Gamma_{i,0}, \Gamma_{i,1}, \Gamma_{i,2}, \Gamma_{i,3}$ 是 $\mathcal{S}$ 中的一条路径。并且，由前一个节点获得后一个节点的重写规则依次是 (pexp_1)、(or)、(and) 以及 (or)。现在说明 $\Gamma_{i,0} \sim \Gamma_{i,3}$ 中的每个节点都保留在 $\mathcal{S}$ 中。

若 $\Gamma_{i,k}$ 被删除，则有 $\vdash \neg\bigwedge\Gamma_{i,k}$。同时，由 (Tau) 得，对每个 $0 \leq k \leq 3$ 都有

$$\vdash \neg\bigwedge\Gamma_{i,k} \rightarrow \neg\bigwedge\Gamma_{i,3} \tag{3.17}$$

成立。因此，若存在某个 $\Gamma_{i,k}$ 已被删除，则皆可由 (MP) 规则得到 $\vdash \neg\bigwedge\Gamma_{i,3}$。现在只需要说明 $\nvdash \neg\bigwedge\Gamma_{i,3}$（也就是 $\nvdash \neg(\bigwedge\Gamma'' \wedge \varphi_l \wedge \bigcirc\psi^{q_f})$）即可。若不然，立即可以得到如下的证明序列：

1. $\vdash \neg(\bigwedge\Gamma'' \wedge \varphi_l \wedge \bigcirc\psi^{q_f})$ ⟅假设⟆
2. $\vdash \neg(\bigwedge\Gamma'' \wedge \varphi_l \wedge \bigcirc\psi^{q_f}) \rightarrow (\bigcirc\psi^{q_f} \rightarrow (\bigwedge\Gamma'' \rightarrow \neg\varphi_l))$ ⟅(Tau)⟆
3. $\vdash \bigcirc\psi^{q_f} \rightarrow (\bigwedge\Gamma'' \rightarrow \neg\varphi_l)$ ⟅1、2 以及(MP)⟆
4. $\vdash \psi^{q_f}$ ⟅$q_f \in F$，(Acc)⟆
5. $\vdash \bigcirc\psi^{q_f}$ ⟅5 以及(XGen)⟆
6. $\vdash \bigwedge\Gamma'' \rightarrow \neg\varphi_l$ ⟅3、5 以及(MP)⟆

而 $\vdash \bigwedge\Gamma'' \rightarrow \neg\varphi_l$ 实际上就是 $\vdash \bigwedge\Gamma'\backslash\{\psi^{q_j}\} \rightarrow \neg\varphi_l$，但这与取反后的结论矛盾。因此，$\Gamma_{i,0}$～$\Gamma_{i,3}$ 都是保留在 $\mathcal{S}$ 中的节点。

再令 P' 是 $\mathcal{S}'$ 中某个起使于 $\Gamma_{i,3}$，结束于某模态节点 Γ_h 的有穷路径，且 P' 中除 Γ_h 外再无其他模态节点（这样的路径一定存在）。易知，$\bigcirc\psi^{q_f}$ 一定出现于 P' 的每个节点中。不妨设 $\Gamma_h = \{\phi_1, \ldots, \phi_k\} \cup \{\bigcirc\psi_1, \ldots, \bigcirc\psi_t\} \cup \{\bigcirc\psi^{q_f}\}$，则令 $\Gamma'_{i,0} = \{\psi_1, \ldots, \psi_t\} \cup \{\psi^{q_j}\}$、$\Gamma'_{i,1} = \{\psi_1, \ldots, \psi_t\} \cup \{true\}$。于是，$\Gamma_h, \Gamma'_{i,0}, \Gamma'_{i,1}$ 必然是 $\mathcal{S}'$ 中的路径（注意，由 Γ_h 未被删除可以推出 $\Gamma'_{i,0}$、$\Gamma'_{i,1}$ 均未被删除）。并且，由前一节点得到后一节点的重写规则分别为 (modal)、(pexp_2)（因为 $q_f \in F$）。

令 $P_{i+1} = P[1,m]\cdot(\Gamma_{i,0}, \Gamma_{i,1}, \Gamma_{i,2}, \Gamma_{i,3})\cdot P'[1..]\cdot(\Gamma'_{i,0}, \Gamma'_{i,1})$。于是，在 $P_0\cdot\ldots\cdot P_i\cdot P_{i+1}$ 中不存在任何极大自动机踪迹能够成为 τ 的拓延（因为 $\mathbf{Dep}(\varphi_l) < \mathbf{Dep}(\psi)$，以及 $\mathbf{Dep}(true) = 0 < \mathbf{Dep}(\psi)$）。再令 $W_{i+1} = \{\tau'_i \mid \tau'_i$ 是 $P_0 \cdot \ldots \cdot P_i \cdot P_{i+1}$ 中的极大自动机踪迹，并且存在 W_i 中的某条踪迹 τ_i，使得 τ'_i 是 τ_i 在 $P_0\cdot\ldots\cdot P_i \cdot P_{i+1}$ 中的拓延$\}$。由前所述的性质(3)，以及 “W_i 中的踪迹 τ 没有在 $P_0 \cdot \ldots \cdot P_{i+1}$ 中的拓延能够成为该路径中的极大自动机踪迹” 这个事实，立即可以得到 $\#W_{i+1} < \#W_i$。

上述构造过程得到的路径 $P_0\cdot\ldots\cdot P_i\cdot\ldots$ 必为无穷路径（这时因为节点 $\emptyset$ 必不在 $\mathcal{S}'$ 中— 否则将存在一条从初始节点 $\{\varphi_0\}$ 到 $\emptyset$ 且满足局部一致性的有穷路径这与 φ 不可满足矛盾。从而该构造可以无穷延续）。以下，记其为 $\widetilde{P}$。该构造过程同时保证了：有无穷多个 $i \in \mathbb{N}$，使得 $W_i = \emptyset$。

现在证明：$\widetilde{P}$ 中的任意踪迹中不会无穷多次出现某个正自动机公式。若不然，设 $\widetilde{P}$ 中的某踪迹 $\widetilde{\tau}$ 以及某个正自动机公式 $\psi^q = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在 $\widetilde{\tau}$ 中出现无穷多次。不失一般性，设 $\widetilde{\tau}(i)$ 是 $\widetilde{\tau}$ 中第一个等于 ψ^q 的公式。于是对于任意的 $i' \geq i$，都有 $\mathbf{Dep}(\widetilde{\tau}(i')) = \mathbf{Dep}(\psi^q)$。同时设 $\psi^q(i)$ 所处的节点为 $\widetilde{P}(j)$ （这里 $j \geq i$）。则，必然存在某个最小的自然数 k 使得 $W_k = \emptyset$ 且 $\sum_{l \leq k} |P_l| > j$。由构造知，$\widetilde{\tau}[i, L_{k+1}] \in W_{k+1}$，其中 $L_{k+1} = \sum_{l \leq k+1} |P_l| - (j - i) - 1$。同时，对每个 $k' > k + 1$，令 $L_{k'} = \sum_{l \leq k'} |P_l| - (j - i) - 1$。注意到 $\widetilde{\tau}[i, L_{k'}]$ 必然是 $\widetilde{\tau}[i, L_{k+1}]$ 在 $P_0 \cdot \ldots \cdot P_{k'}$ 中的拓延，并且 $\widetilde{\tau}[i, L_{k'}]$ 也是 $P_0 \cdot \ldots \cdot P_{k'}$ 中的极大自动机踪迹，因此必然有 $\widetilde{\tau}[i, L_{k'}] \in W_{k'}$。然而，这意味对于任意的 $k' > k$ 都有 $W_{k'} \neq \emptyset$，矛盾!

从而 $\widetilde{P}$ 必然满足全局一致性。

由于 $\mathcal{S}'$ 是一个 $\mathcal{G}_\varphi$ 的连通子图，并且其中的一个节点可由 $\{\varphi\}$ 经过某满足局部一致性的路径到达，因而，$\mathcal{G}_\varphi$ 中必然存在有穷路径 $\widehat{\Gamma_0}, \ldots, \widehat{\Gamma_m}, \widehat{\Gamma_{m+1}}$。其中 $\widehat{\Gamma_0} = \{\varphi\}$，$\widehat{\Gamma_{m+1}} = P_0(0)$，并且任意的 $\widehat{\Gamma_i}$ 中不含 $false$ 或者互补对。令 $\widehat{P} = (\widehat{\Gamma_0}, \ldots, \widehat{\Gamma_m})$。于是，$\widehat{P} \cdot \widetilde{P}$ 就是 $\mathcal{G}_\varphi$ 中满足一致性的路径（局部一致性显然。同时由于 $\widehat{P} \cdot \widetilde{P}$ 中的任意一条无穷踪迹必然有某个无穷后缀在 $\widetilde{P}$ 中，因此其中不可能包含无穷多次出现的正自动机公式，从而满足全局一致性）。

但是，由定理 3.11，这意味着 φ 是可满足的。于是，将结论取反后会得到矛盾，从而该定理成立。 □

定理 3.16 *对于 $\mathscr{F}$ 系统而言，若 ETL_f 公式 φ 是不可满足的，则 $\vdash \neg\varphi$。*

证明. 由于 ETL_f 公式迁移图关于“推理性质”、以及“推理性质删除不变性” 仍然成立。这里仍然采用“公式迁移图删除” 技术对该定理进行证明。

对 $\mathcal{G}$ 中的每个末端极大连通子图 $\mathcal{S}$，假设存在一条从初始节点 $\{\varphi\}$ 到 $\mathcal{S}$ 中某个节点的满足局部一致性的有穷路径（否则，将 $\mathcal{G}_\varphi$ 中所有含 $false$ 或者互补对的节点删除， $\mathcal{S}$ 将变得初始不可达，这样就可将其作为一个整体删除），则采用如下步骤对其进行删除。

同 ETL_l 的公式迁移图一样，首先删除 $\mathcal{S}$ 中所有的含 $false$ 或互补对的节点；其次递归的删除 $\mathcal{S}$ 中的（那些因为删除了含 $false$ 或互补对的节点而成为的）末端独立节点。同样，由推理性质可以证明：对每个被删除的节点 Γ，都有 $\vdash \neg \bigwedge \Gamma$ 成立。

在删除了这些节点以后，$\mathcal{S}$ 剩余的节点就有被分为若干个极大连通子图。设 $\mathcal{S}'$ 是其中的一个末端极大连通子图。

由定理 3.15，一定存在无穷路径 P 以及 P 中的踪迹 τ，以及某个 NFW $\mathcal{A}^q = \langle\{a_1,\ldots,a_n\},\{q_1,\ldots,q_m\},\delta,q,F\rangle$（$q\in\{q_1,\ldots,q_m\}$），以及 ETL_f 公式 φ_1，…，φ_n，使得：

- P 是 $\mathcal{S}'$ 中的迁移公平路径，并且 P 覆盖 $\mathcal{S}'$ 中的每个节点。
- 存在某个 q_i 使得 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 在 τ 中出现无穷多次。
- 对于每个 $1\le j\le m$ 以及 $\mathcal{S}'$ 中的每个状态节点 Γ，若 $\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)\in\Gamma_{[\tau]}$，则 $\vdash\bigwedge(\Gamma\backslash\{\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)\})\to\bigwedge_{\delta(q_j,a_l)\cap F\neq\emptyset}\neg\varphi_l$。

其中，$\Gamma_{[\tau]}$ 表示由 τ 各次经过 Γ 时所取的公式构成的集合。不失一般性，设 τ 中的每个公式均在其中出现无穷多次（若不然，必存在某个 i 使得每个在 τ 中有穷次出现的公式在 $\tau[i..]$ 中不再出现。这时，用 $\tau[i..]$ 替换 τ 即可）。

注意到 $\mathcal{S}'$ 中的任何一个节点 Γ 都位于某个基本迁移路径中（因为 Γ 能到达自身，而该回路中至少有一个模态节点，见引理 3.2），并且 P 是一个迁移公平路径，它会经过每个基本迁移路径无穷多次，所以 τ 必然经过 $\mathcal{S}'$ 中的每个节点。

对每个 $1\le i\le m$，若 $q_i\notin F$，则令 $\mathcal{S}_i=\{\Gamma\mid\Gamma$ 是 $\mathcal{S}'$ 中的状态节点，且 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\in\Gamma_{[\tau]}\}$。同时，对每个 $1\le i\le m$，令

$$\psi_i=\bigvee_{\Gamma\in\mathcal{S}_i}\bigwedge(\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\})\vee\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\tag{3.18}$$

由 τ 的性质，已经保证对每个 $\Gamma\in\mathcal{S}_i$ 有

$$\vdash\bigwedge(\Gamma_i\backslash\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n))\to\bigwedge_{\delta(q_i,a_l)\neq\emptyset}\neg\varphi_l\tag{3.19}$$

同时，对每个 $1\le l\le n$，若 $\delta(q_i,a_l)\cap F\neq\emptyset$，则由 (Expand)、(Tau)、(MP) 以及 (Kri) 可得

$$\vdash\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\to(\neg\varphi_l\vee\bigwedge_{q_j\in\delta(q_i,a_l)}\neg\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n))\tag{3.20}$$

由于 $\delta(q_i,a_l)\cap F\neq\emptyset$，由 (Acc)、(Tau)、(XGen) 以及(MP) 可得

$$\vdash\bigvee_{q_j\in\delta(q_i,a_l)}\bigcirc\mathcal{A}^{q_j}(\varphi_1,\ldots,\varphi_n)\tag{3.21}$$

于是，由 (3.20)、(3.21)、(Tau) 以及(MP) 立即可以得到

$$\vdash\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\to\neg\varphi_l\tag{3.22}$$

于是

$$\vdash \neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \to \bigwedge_{\delta(q_i,a_l)\cap F\neq\emptyset} \neg\varphi_l \tag{3.23}$$

由 (3.19)、(3.23) 以及 ψ_i 的定义式(3.18) 立即得到

$$\vdash \psi_i \to \bigwedge_{\delta(q_i,a_l)\cap F\neq\emptyset} \neg\varphi \tag{3.24}$$

另一方面，对每个 $\Gamma\in\mathcal{S}_i$，由定义有 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\in\Gamma_{[\tau]}$。同时，对于 $\mathcal{S}'$ 中的每个起始于 Γ 的有穷路径 P' 而言，若 P' 是 P 的子路径（即：存在某个 $d\in\mathbb{N}$，使得对于任意的 $0\le d'<|P'|$，都有 $P(d+d')=P'(d')$）则必存在唯一一个起始于 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 的 τ 的“子踪迹” — 该子踪迹保持每个公式的自动机连接子嵌套深度不变（见定理 3.15 证明中的性质 (3)），将其记为 $\tau|_{P',q_i}$。对每个 $1\le j\le n$，令

$$\mathcal{P}_{\Gamma,j}=\left\{P'\;\middle|\;\begin{array}{l}P'\text{ 是 }\mathcal{S}'\text{ 中起始于 }\Gamma\text{，且 }\varphi_j\wedge\bigvee_{q_k\in\delta(q_i,a_j)}\bigcirc\mathcal{A}^{q_k}(\varphi_1,\ldots,\varphi_n)\\ \text{在 }\tau|_{P',q_i}\text{ 中出现的基本迁移路径}\end{array}\right\}。$$

若 $\mathcal{P}_{\Gamma,q_i}=\emptyset$，则表示这些路径在 $\mathcal{S}$ 中已经被删除。这时有

$$\vdash \bigwedge\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\}\to(\neg\varphi_j\vee\bigwedge_{q_k\in\delta(q_i,a_j)}\neg\bigcirc\mathcal{A}^{q_k}(\varphi_1,\ldots,\varphi_n)) \tag{3.25}$$

再由 (3.18)，立即得到

$$\vdash \bigwedge\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\}\to(\varphi_j\to\bigwedge_{q_k\in\delta(q_i,a_j)}\bigcirc\psi_k) \tag{3.26}$$

若 $\mathcal{P}_{\Gamma,q_i}\neq\emptyset$，则不妨设 $\mathcal{P}_{\Gamma,q_i}=\{P'_1,\ldots,P'_s\}$。由于 P 是迁移公平路径，所以每个 P'_l 都会无穷多次作为 P 的子路径出现。对每个 $1\le l\le s$，令 Γ_l 是 P'_l 中最后一个节点。由定义 3.3.4，Γ_l 必然是模态节点。因而，必然存在某个 $q_{i_l}\in\delta(q_i,a_j)$，使得 $\tau|_{\Gamma,q_i}$ 在 Γ_l 中所取的公式为 $\bigcirc\mathcal{A}^{q_{i_l}}(\varphi_1,\ldots,\varphi_n)$。由推理性质，有

$$\vdash(\bigwedge\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\}\wedge\varphi_j\wedge\bigvee_{q_k\in\delta(q_i,a_j)}\bigcirc\mathcal{A}^{q_k}(\varphi_1,\ldots,\varphi_n))\to\bigvee_{1\le l\le s}\bigwedge\Gamma_l \tag{3.27}$$

对 Γ_l 中的每个公式而言：除 $\bigcirc\mathcal{A}^{q_{i_l}}$ 这一个公式之来源于 $\bigvee_{q_k\in\delta(q_i,a_j)}\bigcirc\mathcal{A}^{q_k}(\varphi_1,\ldots,\varphi_n)$ 之外，其余每个公式都来源于 $\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\}\cup\{\varphi_j\}$ 中的某个公式（也就是说，存在由其中的某个公式到该公式的踪迹）。于是，又有

$$\vdash\bigwedge\Gamma\backslash\{\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\}\to(\varphi_j\to\bigvee_{1\le l\le s}\bigwedge\Gamma\backslash\{\bigcirc\mathcal{A}^{q_{i_l}}(\varphi_1,\ldots,\varphi_n)\}) \tag{3.28}$$

对每个 $1 \leq l \leq m$，设 Γ'_l 为 Γ_l 在 $\mathcal{G}_\varphi$ 中（唯一）的直接可达节点。由于 Γ_l 是模态节点，从而 Γ'_l 是状态节点。由于 $\mathcal{S}'$ 是连通图，所以 Γ'_l 一定未被删除。同时，由推理性质，有

$$\vdash \bigwedge \Gamma_l \backslash \{\bigcirc \mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \rightarrow \bigcirc \bigwedge \Gamma'_l \backslash \{\mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \tag{3.29}$$

同时，对每个 Γ'_l 以及 $\mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n)$（其中，$q_k \in \delta(q_i, a_j)$），令

$$\widehat{\Gamma_k} = \Gamma'_l \backslash \{\mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \cup \{\mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n)\}。$$

若 $\widehat{\Gamma_k} \in \mathcal{S}'$，则必然有 $\widehat{\Gamma_k} \in \mathcal{S}_k$。由于

$$\bigwedge \Gamma'_l \backslash \{\mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \leftrightarrow \bigwedge \widehat{\Gamma_k} \backslash \{\mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n)\},$$

所以在这种情况下有

$$\vdash \bigwedge \Gamma'_l \backslash \{\mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \rightarrow \bigvee_{\widehat{\Gamma} \in \mathcal{S}_k} \bigwedge \widehat{\Gamma} \backslash \{\mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n)\} \tag{3.30}$$

成立。若 $\widehat{\Gamma_k} \notin \mathcal{S}'$，则其在 $\mathcal{S}$ 中被删除，在这种情况下有

$$\vdash \bigwedge \Gamma'_l \backslash \{\mathcal{A}^{q_{i_l}}(\varphi_1, \ldots, \varphi_n)\} \rightarrow \neg \mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n) \tag{3.31}$$

成立。由 (3.29)、(3.28)、(3.30)、(3.31)、(3.18)、公理 (Tau)、规则 (MP) 以及 q_k 是任意的 $\delta(q_i, a_j)$ 中的状态，故而

$$\vdash \Gamma \backslash \{\mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)\} \rightarrow (\varphi_j \rightarrow \bigcirc \psi_k) \tag{3.32}$$

再由 j、k 的任意性，有

$$\vdash \Gamma \backslash \{\mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)\} \rightarrow \bigwedge_{1 \leq j \leq n} (\varphi_j \rightarrow \bigwedge_{q_k \in \delta(q_i, a_j)} \bigcirc \psi_k) \tag{3.33}$$

同时，因为 $q_i \notin F$，由 (Expand)、(Tau) 以及(MP) 有

$$\vdash \neg \mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n) \rightarrow \bigwedge (\varphi_j \rightarrow \bigwedge_{q_k \in \delta(q_i, a_j)} \neg \bigcirc \mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n)) \tag{3.34}$$

成立。而由 (3.18) 可得 $\vdash \neg \mathcal{A}^{q_k} \rightarrow \psi_k$。于是就有

$$\vdash \neg \mathcal{A}^{q_k}(\varphi_1, \ldots, \varphi_n) \rightarrow \bigwedge_{1 \leq j \leq n} (\varphi_j \rightarrow \bigwedge_{q_k \in \delta(q_i, a_j)} \bigcirc \psi_k) \tag{3.35}$$

进而，根据 (3.33)、(3.35) 以及 (3.18) 有

$$\vdash \psi_i \to \bigwedge_{1\le j\le n} (\varphi_j \to \bigwedge_{q_k\in\delta(q_i,a_j)} \bigcirc\psi_k) \tag{3.36}$$

最后，由 (3.24)、(3.36)、以及 (Fin) 规则，可以得到

$$\vdash \psi_i \to \neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \tag{3.37}$$

因而，对每个 $\Gamma\in\mathcal{S}_i$，都有

$$\vdash \bigwedge\Gamma \to \neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \tag{3.38}$$

但是，由定义，对每个 $\Gamma\in\mathcal{S}_i$，有 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)\in\Gamma$。所以又有

$$\vdash \bigwedge\Gamma \to \mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \tag{3.39}$$

成立。这样由 (3.38)、(3.39)、(Tau) 公理以及 (MP) 规则，立即可以得到

$$\vdash \neg\bigwedge\Gamma \tag{3.40}$$

这个结论。由路径 P 的性质，$\mathcal{S}'$ 中的每个状态节点都属于某个 $\mathcal{S}_i$。因此，$\mathcal{S}'$ 中的每个状态节点都可被删除。进而，每个模态节点也可删除。这样，$\mathcal{S}'$ 中不再存在回路（因为引理 3.2 中的结论对 ETL_f 公式的迁移图仍然成立）。从而，可以由推理性质将 $\mathcal{S}'$ 中所有的节点删除。

重复此过程，删除 $\mathcal{G}_\varphi$ 中其余极大连通子图和独立节点，直至初始节点 $\{\varphi\}$ 被删除。这时，就可得到 $\vdash\neg\varphi$。 □

例 3.3.1 设 $NFW\,\mathcal{A}^{q_1}=\langle\{a_1,a_2\},\{q_1,q_2\},\delta,q_1,\emptyset\rangle$，其中 $\delta(q_1,a_1)=\{q_2\}$，$\delta(q_2,a_2)=\{q_1\}$、$\delta(q_1,a_2)=\delta(q_2,a_1)=\emptyset$。再设 $\varphi^{q_1}=\mathcal{A}^{q_1}(p_1,p_2)$、$\varphi^{q_2}=\mathcal{A}^{q_2}(p_1,p_2)$，其中 p_1、p_2 是原子命题。由于 $\mathcal{A}^{q_1}$ 的接收状态集为 $\emptyset$，所以 $\varphi^{q_1}\wedge\varphi^{q_2}$ 不可满足。该公式对应的迁移图如图 *3.2* 所示。其中，对 φ^{q_1} 和 φ^{q_2} 使用 (pexp_1) 规则时进行了化简。同时，为简单起见，除 (Γ_0,Γ_1) 之外的其余边（实心线）未画出。这样，$\Gamma_1\sim\Gamma_9$ 便构成了一个极大连通子图 $\mathcal{S}$。

在使用“迁移图删除”技术证明 $\vdash\neg(\varphi^{q_1}\wedge\varphi^{q_2})$ 过程中，$\mathcal{S}$ 中对应于定理 *3.15* 的踪迹 τ 以点划线标出。从图中可以看出：对每个 $1\le i\le 9$，$\#\Gamma_{i[\tau]}=2$。

在该迁移图中，唯一的状态节点是 Γ_1。由于 $\Gamma_{1[\tau]}=\{\varphi^{q_1},\varphi^{q_2}\}$，所以可令 $\mathcal{S}_1=\mathcal{S}_2=\{\Gamma_1\}$。于是，$\psi_1=\varphi^{q_2}\vee\neg\varphi^{q_1}$，$\psi_2=\varphi^{q_1}\vee\neg\varphi^{q_2}$。注意到 $\delta(q_1,a_2)=\delta(q_2,a_1)=\emptyset$，

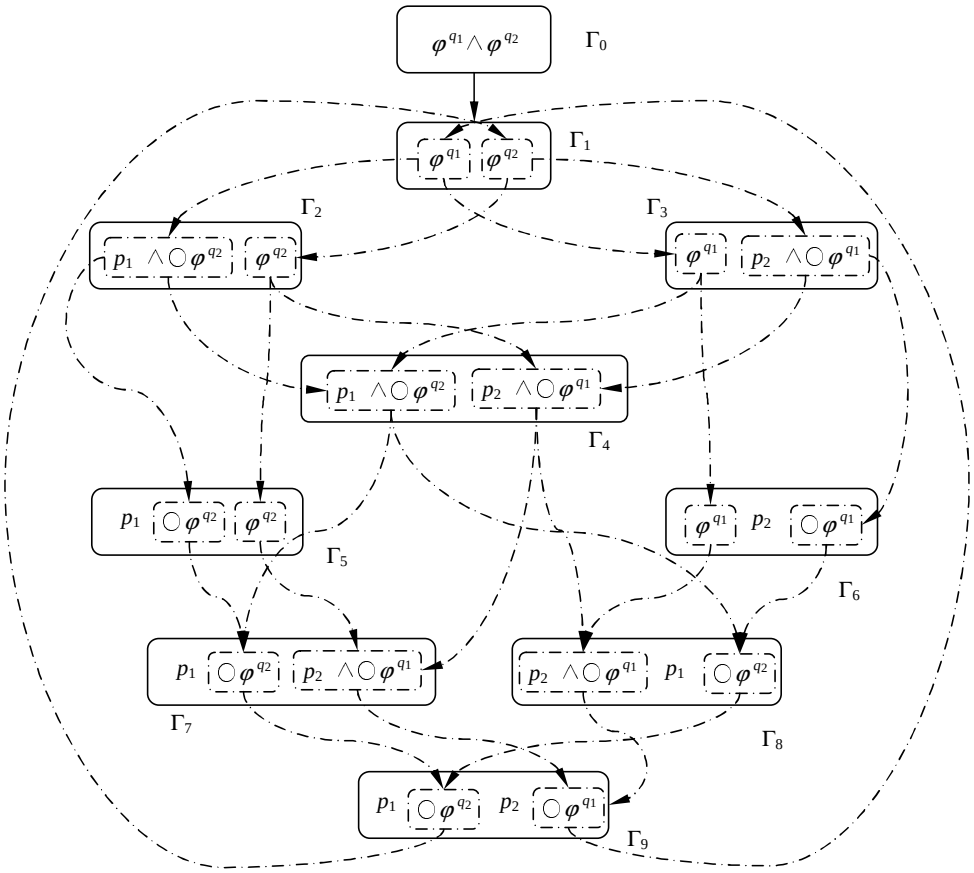


图 3.2 基于"迁移图删除" 之 ETL_f 完备性证明示例

而 $\bigwedge \emptyset = true$，于是

$$\vdash \quad \psi_1 \rightarrow ((p_1 \rightarrow \bigcirc \psi_2) \wedge (p_2 \rightarrow true))$$
$$\vdash \quad \psi_2 \rightarrow ((p_1 \rightarrow true) \wedge (p_2 \rightarrow \bigcirc \psi_1))$$

同时注意到 $\mathcal{A}^{q_1}$ 的接收状态集为 $\emptyset$，故而由上式以及 (Fin) 规则立即可以得到

$$\vdash \quad \psi_1 \rightarrow \neg \varphi^{q_1}$$
$$\vdash \quad \psi_2 \rightarrow \neg \varphi^{q_2}$$

也就是

$$\vdash \quad (\neg \varphi^{q_1} \vee \varphi^{q_2}) \rightarrow \neg \varphi^{q_1}$$
$$\vdash \quad (\varphi^{q_1} \vee \neg \varphi^{q_2}) \rightarrow \neg \varphi^{q_2} \quad 。$$

于是，从上式不难得到 $\vdash \varphi^{q_1} \rightarrow \neg \varphi^{q_2}$（或者 $\vdash \varphi^{q_2} \rightarrow \neg \varphi^{q_1}$），即: $\vdash \neg(\varphi^{q_1} \wedge \varphi^{q_2})$。这样，$\Gamma_1$ 可以删除。进而，$\mathcal{S}$ 中其余节点也可以删除（因为 Γ_1删除后不再有回路），Γ_0 也可删除（因为 Γ_1 是其唯一后继）— 事实上，在本例中，在删除 Γ_1 之前目标公式就已经被证出。 □

定理 3.17 ($\mathscr{F}$ 的完备性) 若 ETL_f 公式 φ 是有效的，则 $\vdash \varphi$

证明. 同定理 3.10 的证明。 □

3.4 $\mathbf{ETL}_r$ 的公理系统 $\mathscr{R}$

3.4.1 $\mathbf{ETL}_r$ 重写系统及迁移图

对于 ETL_r 而言，其重写规则与 ETL_l 的重写规则完全相同。除“**路径一致性**”这个概念之外，其余与公式迁移图相关的概念定义同前。

ETL_r 中的时序连接子是非确定的 repeating（或者 Büchi 自动机）。因而它比 ETL_l、ETL_f 更具有一般性。在 ETL_r 中，正/负自动机公式都会产生一致性约束。

定义 3.4.1 ($\mathbf{ETL}_r$ 公式迁移图路径一致性) 给定 ETL_r 公式 φ，设 $\mathcal{G}_\varphi$ 为其公式迁移图，P 是 $\mathcal{G}_\varphi$ 中的路径。如果 P 满足:

局部一致性：P 的任何一个节点中不含 $false$ 或者互补对;

正全局一致性：对 P 中任意的踪迹 τ，若某正自动机公式 $\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 在 τ 中出现无穷多次，则存在 $\mathcal{A}^{q_i}$ 的某个接收状态 q_f，使得 $\mathcal{A}^{q_f}(\varphi_1,\ldots,\varphi_n)$ 也在 τ 中出现无穷多次;

负全局一致性：对 P 中任意的踪迹 τ，若某负自动机公式 $\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 在 τ 中出现无穷多次，则对 $\mathcal{A}^{q_i}$ 中的每个接收状态 q_f 而言，$\neg\mathcal{A}^{q_f}(\varphi_1,\ldots,\varphi_n)$ 在 τ 中只出现有穷次

则称 P 满足**一致性**。同样，若公式迁移图 $\mathcal{G}_\varphi$ 中存在一条满足一致性的完全路径，则称 $\mathcal{G}_\varphi$ 是**一致**的。 □

定理 3.18 给定 ETL_r 公式 φ，若 $\mathcal{G}_\varphi$ 是一致的，则 φ 是可满足的。

证明. 类似于定理 3.3 以及定理 3.11 的证明。由定义，$\mathcal{G}_\varphi$ 中必存在满足一致性的完全路径 $P=\Gamma_0,\Gamma_1,\ldots$，并且 $\Gamma_0=\{\varphi\}$。

在此，仍令 Γ_{M_i} 为 P 中第 i 个模态节点，令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目，以及 π 是某个满足约束 “对于任意的 $i\in\mathbb{N}$，若 Γ_{M_i} 存在，则 $\pi(i)\models\bigwedge(\Gamma_{M_i}\cap(AP\cup\overline{AP}))$ ” 的线性结构。现在，利用公式结构归纳法证明：对于任意的 $\psi\in\Gamma_i$，都有 $\pi,\gamma(i)\models\psi$。

- 当 ψ 是文字，或者是形如 $\varphi_1\wedge\varphi_2$、$\varphi_1\vee\varphi_2$、$\neg(\varphi_1\wedge\varphi_2)$、$\neg(\varphi_1\vee\varphi_2)$ 或者 $\bigcirc\psi'$、$\neg\neg\psi'$，$\neg\bigcirc\psi'$ 的公式时，与定理 3.3 中的证明完全相同。
- 若 $\psi=\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$，且 $\mathcal{A}=\langle\{a_1,\ldots,a_n\},Q,\delta,q,F\rangle$，则经使用 (pexp)、(or)、以及 (and) 后，存在 $1\le k_0\le n$ 及 $i<j\le M_{\gamma(i)}$ 和 $q_1\in Q$，满足：

1. $q_1 \in \delta(q, a_{k_0})$；

2. $\varphi_{k_0} \in \Gamma_j$；

3. $\bigcirc\psi^{q_1} = \bigcirc\mathcal{A}^{q_1}(\varphi_1, \ldots, \varphi_n) \in \Gamma_{M_{\gamma(i)}}$。

重复讨论此过程，便可以得到一个无穷字 $a_{k_0}, a_{k_1}, \ldots$，其在 $\mathcal{A}^q$ 上的运行为 $q_0, q_1, \ldots$。同时，由于 P 满足正全局一致性，所以一定有无穷多个 $i \in \mathbb{N}$ 使得 $q_i \in F$，因此 $a_{k_0}, a_{k_1}, \ldots \in \mathbf{L}(\mathcal{A}^q)$。并且，对每个 $l \in \mathbb{N}$ 都有 $\pi, \gamma(i) + l \models \varphi_{k_l}$。于是，由定义有 $\pi, \gamma(i) \models \psi$。

- 若 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，则仍然按照定理 3.3 中的方式构造 Q-标记树 $\langle T, \rho\rangle$。注意 T 中的任意一条路径 $\sigma = x_0, x_1, \ldots$ 都对应于 P 中的一条踪迹 τ，其中 $\neg\mathcal{A}^{\rho(x_i)}(\varphi_1, \ldots, \varphi_n)$ 依次从 τ 中出现。由于 P 满足负全局一致性，所以 $\mathbf{Inf}(\rho(\sigma)) \cap F = \emptyset$。于是，$\langle T, \rho\rangle$ 一定是 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 在 π 上开始于位置 $\gamma(i)$ 的一个拒绝例证。因此，$\pi, \gamma(i) \models \psi$。

由上述归纳，得 $\pi, \gamma(0) \models \varphi$，即：$\pi \models \varphi$。 □

定理 3.19 给定 ETL_r 公式 φ，若 φ 是可满足的，则 $\mathcal{G}_\varphi$ 是一致的。

证明. 由于 φ 是可满足的，不妨设线性结构 $\pi \models \varphi$。同定理 3.4 以及 3.12 的证明类似，也需要证实 $\mathcal{G}_\varphi$ 中存在满足一致性的完全路径 $P = \Gamma_0, \Gamma_1, \ldots$。

但是，由于ETL_r 中同时要求正全局一致性和负全局一致性，因此其中正、负自动机公式的衍生公式都会添加标注。其中，正自动机公式及其衍生公式（包括形如 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigcirc\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigvee_k(\varphi_k \wedge \bigvee_j \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n))$、$\varphi_k \wedge \bigvee_j \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 或者 $\bigvee_j \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 的公式）的标注是一个无穷“状态-字母” 交错序列；而对于负自动机公式及其衍生公式（包括形如 $\neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigcirc\neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigwedge(\neg\varphi_k \vee (\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)))$、$\neg\varphi_k \vee (\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n))$、$\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 或者 $\bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 的公式）的标注是一棵标记树。

首先，令$\Gamma_0 = \{\varphi\}$。此时，对每个 Γ_i，仍令 $\gamma(i)$ 为出现在 Γ_i 之前的模态节点数目。由于 $\gamma(0) = 0$，所以，由假设知 $\pi, \gamma(0) \models \varphi = \bigwedge\Gamma_0$。

对当前节点 Γ_i，归纳假设 $\pi, \gamma(i) \models \bigwedge\Gamma_i$ 成立，则按照下列方式获得 P 中的下一个节点 Γ_{i+1} 并为其中的公式设置标注。以下，称 ψ 在节点 Γ_{i+1} 中的某生成公式 ψ' 是**新生成的**，是指 ψ' *在* Γ *中不存在*。特别的，这里也将 Γ_0 中的 φ 视为新的生成公式。

若 Γ_i 是模态节点，则直接令 Γ_{i+1} 为由 Γ_i 经使用 (modal) 规则得到的节点。同时，对 Γ_{i+1} 中的每个公式 ψ，若 $\bigcirc\psi$ 在 Γ_i 中存在标注 A，则将 ψ 在 Γ_{i+1} 中的标注设为 A。这时，由于 $\gamma(i+1) = \gamma(i) + 1$，所以 $\pi, \gamma(i+1) \models \bigwedge\Gamma_{i+1}$ 成立。

若 Γ_i 不是模态节点，则在 Γ_i 中任取（不是文字的）公式 ψ 作为消解公式得到下一节点 Γ_{i+1}。对每个 $\phi \in \Gamma_i \cap \Gamma_{i+1}$，若 ϕ 在 Γ_i 中存在且在其中有标注，则将其在 Γ_{i+1} 中的标注设为其在 Γ_i 中的标注。若 ψ 在 Γ_i 中没有标注，则 Γ_{i+1} 的获取方式同见定理 3.4 中的构造。若 ψ 在 Γ_i 中存在标注，则在生成 Γ_{i+1} 时，为新生成的公式或者在 Γ_i 中不存在标注的生成公式 ψ' 添加标注的过程如下。

- 若 Γ_{i+1} 中的新生成公式 $\psi' = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，且 $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, F \rangle$，其中 Σ 与 Q 分别为 $= \{a_1, \ldots, a_n\}$ 和 $\{q_1, \ldots, q_m\}$：则由于 $\pi, \gamma(i) \models \psi$，则必然存在 $w = a_{k_0}, a_{k_1}, \ldots \in \mathbf{L}(\mathcal{A}^q)$，以及 w 在 $\mathcal{A}^q$ 上的可接收运行 $q_0, q_1, \ldots \in Q^\omega$，其中 $q_0 = q$，以及有无穷多个 t 使得 $q_t \in F$。同时，对于每个 $j \in \mathbb{N}$，有 $\pi, \gamma(i) + j \models \varphi_{k_j}$。这样，将 ψ' 在 Γ_{i+1} 的标注设为 $q_0, a_{k_0}, q_1, a_{k_1}, \ldots$。
- 若 Γ_{i+1} 中的新生成公式 $\psi' = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，则由于此时 $\gamma(i+1) = \gamma(i)$，故而 ψ' 存在某个在 π 上起始于 $\gamma(i)$ 的拒绝例证 $\langle T, \rho \rangle$。将 ψ' 在 Γ_{i+1} 中的标注设为 $\langle T, \rho \rangle$ 即可。
- 当 ψ 形如 $\bigvee_k(\varphi_k \wedge \bigvee_j \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n))$、或者 $\varphi_k \wedge \bigvee_j \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 时，处理方式与定理 3.4 中相同。
- 当 ψ 形如 $\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$ 时，将 Γ_{i+1} 中由 ψ 经 (pexp) 规则得到的生成公式的标注赋为 ψ 在 Γ_i 中的标注。
- 当 ψ 形如 $\neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigcirc\neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$、$\bigwedge(\neg\varphi_k \vee (\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)))$ 或者 $\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 时，将 Γ_{i+1} 中每个与 ψ 具有相同自动机连接子嵌套深度的新生成公式的标注赋为 ψ 在 Γ_i 中的标注。
- 当 $\psi = \neg\varphi_k \vee (\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n))$ 时，若 $\pi, \gamma(i) \models \neg\varphi_k$ 则将 $\neg\varphi_k$ 作为 (or) 规则的生成公式；否则，将 $\varphi_k \wedge \bigwedge_j \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 作为生成公式，并将其在 Γ_{i+1} 中的标注赋为 ψ 在 Γ_i 中的标注。
- 当 $\psi = \psi_1 \wedge \psi_2$, 其中 $\psi_1 = \bigcirc\neg\mathcal{A}^{q_l}(\varphi_1, \ldots, \varphi_2)$，$\psi_2 = \bigwedge_{j \neq l} \bigcirc\neg\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$ 时，在采用 (and) 规则后，ϕ_1 和 ϕ_2 都会作为生成公式出现在 Γ_{i+1} 中。假设 ψ 在 Γ_i 中的标注为 $\langle T, \rho \rangle$。若 ψ_1 是 Γ_{i+1} 中新的生成公式，则由标注的构造保证：在 T 中必然存在深度为 1 的节点 c 使得 $\rho(c) = q_l$。这样，将 ψ_1 在 Γ_{i+1} 的标注设为 $\langle T_c, \rho_c \rangle$，其中 $T_c = \{x \in \mathbb{N}^* \mid c \cdot x \in T\}$，$\rho_c(x) = \rho(c \cdot x)$。若 ψ_2 是 Γ_{i+1} 中新的生成公式，当其含有多个合取项时，将 ψ_2 在 Γ_{i+1} 中的标注赋为 $\langle T, \rho \rangle$。若 ψ_2 只含一个合取项时，不妨设 $\psi_2 = \bigcirc\neg\mathcal{A}^{q'_l}$。同样，在 T 中必然存在深度为 1 的节点 c' 使得 $\rho(c') = q'_l$。这样，将 ψ_2 在 Γ_{i+1} 的标注设为 $\langle T_{c'}, \rho_{c'} \rangle$，其中 $T_{c'} = \{x \in \mathbb{N}^* \mid c' \cdot x \in T\}$，$\rho_{c'}(x) = \rho(c' \cdot x)$。

上述过程的每种情形都保证 $\pi, \gamma(i+1) \models \bigwedge \Gamma_{i+1}$，成立。

若遇到某个 $\Gamma_i = \emptyset$，则停止构造，否则，会得到 $\mathcal{G}_\varphi$ 中的一条无穷路径。由于对每个 Γ_i 都有 $\pi, \gamma(i) \models \bigwedge \Gamma_i$ 成立，因此该路径必然满足局部一致性。由标注过程，保证了该路径必然还满足正 / 负全局一致性。这是因为：对于该路径中任意一条踪迹 τ，若某正（resp. 负）自动机公式 ψ（resp. $\neg\psi$）在 τ 中出现无穷多次，且设 ψ^{q_i}（resp. $\neg\psi^{q_i}$）是 τ 中依次出现的第 i 个与 ψ 具有相同自动机连接子深度的正（resp. 负）自动机公式，则由构造保证：$q_0, q_1, q_2 \ldots$ 是 ψ^{q_0}（resp. $\neg\psi^{q_0}$）在某个节点内的标注（resp. 在某个节点内的标注树中的某条路径）。于是，有无穷多个（resp. 有穷个）i 使得 q_i 为接收状态。

所以，该路径是满足一致性的路径。从而 $\mathcal{G}_\varphi$ 是一致的。 □

推论 3.20 *ETL_r 公式 φ 是可满足的，当且仅当其公式迁移图 $\mathcal{G}_\varphi$ 是一致的。*

3.4.2 $\mathbf{ETL}_r$ 公理系统及可靠性、完备性

本节给出 ETL_r 的公理系统。在此之前，首先考察一类特殊的非确定 Büchi 自动机。

定义 3.4.2 (极大 NBW) *设 NBW（即：非确定 repeating/Büchi 自动机）$\mathcal{A} = \langle \Sigma, Q, \delta, q, F \rangle$。称状态序列 $q_1, q_2, \ldots, q_m \in Q^*$ 是 $\mathcal{A}$ 中的**迁移环路**，如果 $q_1 = q_m$，并且存在 $a_1, a_2, \ldots, a_{m-1} \in \Sigma^*$，使得对每个 $1 \leq i < m$，都有 $q_{i+1} \in \delta(q_i, a_i)$。称 $\mathcal{A}$ 是**极大**的，当且仅当对 $\mathcal{A}$ 中的每个迁移环路 $q_1, \ldots, q_m$ 而言，都存在某个 $1 \leq i < m$ 使得 $q_i \in F$。* □

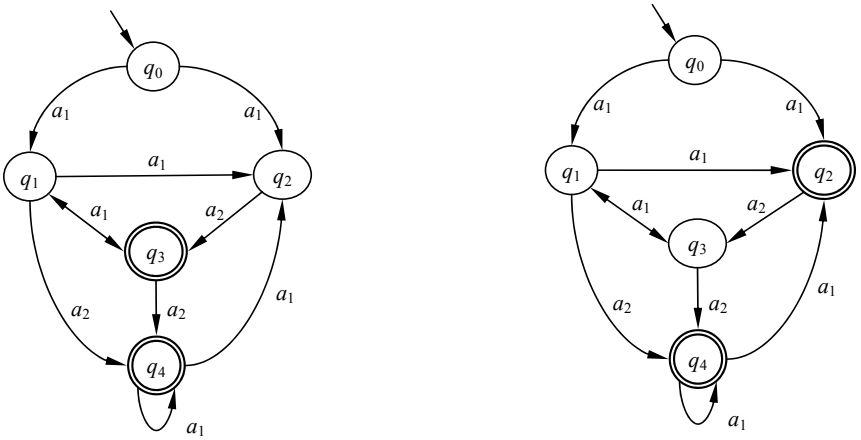
例 3.4.1 *设 NBW $\mathcal{A}_1$、$\mathcal{A}_2$ 分别如图 3.3(a) 及图 3.3(b) 所示。其中，$\mathcal{A}_1$ 的接收状态集为 $\{q_3, q_4\}$，$\mathcal{A}_2$ 的接收状态集为 $\{q_2, q_4\}$。于是，$\mathcal{A}_1$ 是极大 NBW，而 $\mathcal{A}_2$ 不是极大 NBW。这是因为 $\mathcal{A}_2$ 的迁移环路 q_1, q_3, q_1 中不含接收状态。* □

关于极大的 NBW 而言，下面的引理显然成立。

引理 3.21 *若 $\mathcal{A} = \langle \Sigma, Q, \delta, q, F \rangle$ 是极大的 NBW，则 $\mathcal{A}$ 的每个运行都是可接收的。*

ETL_r 的公理系统 $\mathscr{R}$ 由 5 条公理以及 4 条规则组成，如表 3.3 所示。对于该公理系统，有如下几点需要说明：

1. 规则 (PRep) 及规则 (NRep) 中的自动机连接子 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, \{q_1, \ldots, q_m\}, \delta, q, F \rangle$，并且 $q \in \{q_1, \ldots, q_m\}$。
2. 在公理 (Mono) 中，$\mathcal{A}_1$ 和 $\mathcal{A}_2$ 是具有相同的（n-字母）字母表的 NBW，并且，使用该公理的前提条件为 $\mathbf{L}(\mathcal{A}_1) \subseteq \mathbf{L}(\mathcal{A}_2)$。
3. 在规则 (PRep) 中，要求每个连接子 $\mathcal{A}^{q_i}$ 都是极大的 NBW。



(a) NBW $\mathcal{A}_1$ (b) NBW $\mathcal{A}_2$

图 3.3 极大/非极大 NBW 的示例

同样，为方便起见，在本节中将 $\vdash_{\mathscr{R}}$ 直接简写为 $\vdash$。

定理 3.22 ($\mathscr{R}$ 的可靠性) 对 $\mathscr{R}$ 系统而言，若 $\vdash \varphi$，则 $\models \varphi$。

证明. 这里，只说明新的公理/规则的有效性即可。

- 在新的公理中，(Mono) 的有效性显然。这是因为对于任意的线性结构 π 而言，若 $\pi \models \mathcal{A}_1(\varphi_1, \ldots, \varphi_n)$ 成立，则存在 $w \in \mathbf{L}(\mathcal{A}_1)$，使得对于任意 $i \in \mathbb{N}$ 有：若 $w(i) = a_k$，则 $\pi, i \models \varphi_k$。由于 $\mathbf{L}(\mathcal{A}_1) \subseteq \mathbf{L}(\mathcal{A}_2)$，所以 $w \in \mathbf{L}(\mathcal{A}_2)$。由定义有 $\pi \models \mathcal{A}_2(\varphi_1, \ldots, \varphi_n)$ 成立。
- 对于规则 (PRep)，其有效性的保持如定理 3.6 中关于 (Loop) 之证明：当前件有效时，必然对每个 ψ_i 以及线性结构 π 有：若$\pi \models \psi_i$ 成立，则存在无穷字 $w \in \mathbf{L}(\mathcal{A}^{q_i})$，使得对于任意的 $j \in \mathbb{N}$ 而言，若 $w(j) = a_k$，则 $\pi, j \models \varphi_k$。由于 $\mathcal{A}^{q_i}$ 是极大的，由引理 3.21 知 $\mathcal{A}^{q_i}$ 在 w 上的对应的运行必为可接收的。于是，$\pi \models \mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$。因而，后件也是有效的。
- 对于规则 (NRep)，其证明完全类似于 $\mathscr{F}$ 系统中关于 (Fin) 规则有效性保持的证明。在前件为永真的前提下，同样可以对每个 ψ_i 构造一棵标记树 $\langle T_i, \rho_i \rangle$，该标记树实质为 $\mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$ 在 π 上（起始于 0）的拒绝例证。从而，$\pi \models \neg\mathcal{A}^{q_i}(\varphi_1, \ldots, \varphi_n)$。因此，后件也是有效公式。

这样，$\mathscr{R}$ 系统是可靠的公理系统。 □

注意，在规则 (PRep) 中，限制自动机连接子的极大性是必要的，否则，该规则会失效。考虑下例中的情形：

例 3.4.2 设 $NBW\, \mathcal{A}_1 = \langle \{a_1, a_2\}, \{q_1, q_2\}, \delta_1, q_1, \{q_2\} \rangle$，其中 $\delta_1(q_1, a_1) = \{q_1\}$，$\delta_1(q_1,$

表 3.3 ETL_r 的公理系统 $\mathscr{R}$

公 理
(Tau)、(Next)、(Kri)、(Expand) 同 $\mathscr{L}$ 中对应的公理。
$\mathcal{A}_1(\varphi_1,\ldots,\varphi_n)\rightarrow\mathcal{A}_2(\varphi_1,\ldots,\varphi_n)$ (Mono)
规 则
(MP)、(XGen) 同 $\mathscr{L}$ 中对应的规则。
$\dfrac{\bigwedge_{1\leq i\leq m}(\psi_i\rightarrow\bigvee_{1\leq k\leq n}(\varphi_k\wedge\bigvee_{q_j\in\delta(q_i,a_k)}\bigcirc\psi_j))}{\bigwedge_{1\leq i\leq m}(\psi_i\rightarrow\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n))}$ (PRep)
$\dfrac{\bigwedge_{\substack{1\leq i\leq m\\ q_i\notin F}}(\psi_i\rightarrow(\bigwedge_{1\leq k\leq n}(\varphi_k\rightarrow\bigwedge_{q_j\in\delta(q_i,a_k)}\bigcirc\psi_j)\wedge\bigwedge_{\delta(q_i,a_l)\cap F\neq\emptyset}\neg\varphi_l))}{\bigwedge_{\substack{1\leq i\leq m\\ q_i\notin F}}(\psi_i\rightarrow\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n))}$ (NRep)

$a_2)=\delta_1(q_2,a_1)=\delta_1(q_2,a_2)=\{q_2\}$。设 NBW $\mathcal{A}_2=\langle\{a_1\},\{q_1\},\delta_2,q_1,\{q_1\}\rangle$，其中 $\delta_2(q_1,a_1)=\{q_1\}$。再设 p_1，p_2 为原子命题。容易证明：$\mathcal{A}_1^{q_1}(p_1,p_2)$ 与 LTL 公式 $p_1\mathsf{U}p_2$ 等价；$\mathcal{A}_2^{q_1}(p_1)$ 与 LTL 公式 $\mathsf{G}p_1$ 等价。令 $\psi_1=\mathcal{A}_2^{q_1}(p_1)$，$\psi_2=\mathcal{A}_1^{q_2}(p_1,p_2)$，于是：

1. $\vdash\psi_1\leftrightarrow(p_1\wedge\bigcirc\psi_1)$ ⟅(Expand)⟆
2. $\vdash\psi_2\leftrightarrow((p_1\wedge\bigcirc\psi_2)\vee(p_2\wedge\bigcirc\psi_2))$ ⟅(Expand)⟆
3. $\vdash\psi_1\rightarrow((p_1\wedge\bigcirc\psi_1)\vee(p_2\wedge\bigcirc\psi_2))$ ⟅1, (Tau) 以及 (MP)⟆

但是，由于 $\mathcal{A}_1$ 不是极大的 NBW，所以并不能由 2、3 及 (PRep) 得到

$$\psi_1\rightarrow\mathcal{A}^{q_1}(p_1,p_2)$$

这样的结论。否则，将会证出 $\mathsf{G}p_1\rightarrow p_1\mathsf{U}p_2$ 这样的非有效公式。 □

定理 3.23 若 ETL_r 公式 φ 不可满足，$\mathcal{S}$ 是其公式迁移图中的某个末端极大连通子图。并且：

- $\mathcal{S}$ 中的任意无穷路径都满足负全局一致性。
- $\mathcal{S}$ 中所有含有 $false$ 或者互补对的节点已被删除。对于任意的 $\Gamma\in\mathcal{S}$，若其所有直接可达节点已被删除，则 Γ 也已被删除。

- 存在从初始节点 $\{\varphi\}$ 到 $\mathcal{S}'$ 中某节点的路径，并且该路径中的任何节点不含 $false$ 或者互补对。

那么，一定存在无穷路径 P，和 P 中的某条无穷踪迹 τ，以及某个 NBW $\mathcal{A}^{q_i} = \langle\{a_1,\ldots,a_n\},\{q_1,\ldots,q_m\},\delta,q_i,F\rangle$，以及 ETL_r 公式 φ_1，...，φ_n，使得:

- P 是 $\mathcal{S}$ 中的迁移公平路径，并且 P 覆盖 $\mathcal{S}$ 中的每个节点。
- $\psi^{q_i} = \mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n)$ 在 τ 中出现无穷多次。
- 对于每个 $1 \leq j \leq m$ 以及 $\mathcal{S}$ 中的每个状态节点 Γ，若 $\psi^{q_j} \in \Gamma_{[\tau]}$，则 $\vdash \bigwedge\Gamma\backslash\{\psi^{q_j}\} \rightarrow \bigwedge_{\delta(q_j,a_l)\cap F\neq\emptyset} \neg\varphi_l$。

证明. 该定理的证明完全类似于定理 3.15 之证明— 同样利用反证法，将结论取反之后，通过构造满足一致性的路径导致矛盾得出。注意：同定理 3.15 相比，现在增加了一个条件：$\mathcal{S}$ 中的任意无穷路径都满足负全局一致性。

为方便证明，先给出几个定义。

1. 起始于某个正自动机公式，且保持自动机连接子嵌套深度不变的踪迹称为**正自动机踪迹**。
2. 设 τ' 为某个（有穷或无穷）正自动机踪迹，且 τ' 中公式中的最外层自动机连接子为 $\mathcal{B}$，且不妨设该连接子的字母表 $\Sigma' = \{a'_1,\ldots,a'_t\}$，接收状态集为 F'。则称 τ' 的后缀 $\tau[i..]$ 是**接收无关**的，当且仅当对于任意的 $j \geq i$，若 $\tau(j)$ 是某个形如 $\mathcal{B}^{q'}(\varphi'_1,\ldots,\varphi'_t)$ 的公式，则 $q' \notin F'$。
3. 同时，由重写规则，易证：对于 $\mathcal{S}$ 中的任意有穷路径 P，以及 P 中的有穷正自动机踪迹 τ 而言，τ 在 P 中的所有拓延（见定义 3.3.3）之中，有且只有一个极大正自动机踪迹。将该极大自动机踪迹称为 τ 在 P 中的**保持拓延**。

同定理 3.15，下面给出在将结论取反的前提下，满足一致性的无穷路径 $\widehat{P}$。该路径仍是无穷多个有穷路径 P_0，P_1，... 的连接。

令 P_0 是任意一条由若干 $\mathcal{S}$ 中的节点构成的，且包含极大自动机踪迹的有穷路径。同时，令 $W_0 = \{\tau[i..] \mid \tau$ 是 P_0 中的极大自动机踪迹，$\tau[i..]$ 是 τ 中的接收无关后缀，并且对任意的 $i' < i$ 而言，$\tau[i'..]$ 都不是接收无关的$\}$。

对每个 $i \in \mathbb{N}$，假设 P_i 与 W_i 已经获得，不妨设 P_i 中的最后一个节点为 Γ，则按照如下步骤获得 P_{i+1} 与 W_{i+1}。

- 若 $W_i = \emptyset$，则令 P 是任意一条由 $\mathcal{S}$ 中节点构成的起始于节点 Γ 且至少包含一条极大自动机踪迹的有穷路径。令 $P_{i+1} = P[1..]$，$W_{i+1} = \{\tau[i..] \mid \tau$ 是 $P_0\cdot\ldots\cdot P_i\cdot P_{i+1}$ 中的极大自动机踪迹，$\tau[i..]$ 是 τ 中的接收无关后缀，并且对任意的 $i' < i$ 而言，$\tau[i'..]$ 都不是接收无关的$\}$。

- 若 $W_i \neq \emptyset$，则任取以 Γ 为起点的且覆盖 $\mathcal{S}$ 中所有节点的迁移公平路径 P。易知，对每个 $\tau \in W_i$，在 $P_0 \cdot P_1 \cdot \ldots \cdot P_i \cdot P[1..]$ 中均存在其唯一的保持拓延 τ'。不失一般性，设 τ' 中无穷出现的最外层自动机连接子为 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, F \rangle$；设公式 $\psi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 在 τ' 中出现无穷多次。

  由取反后的结论，必然存在 P 中的某个状态节点 Γ'（这里，不妨设 $\Gamma' = P(m)$），以及 $q_j \in Q$，$1 \leq l \leq n$，使得 $\psi^{q_j} \in \Gamma'_{[\tau']}$；同时，$\delta(q_j, a_l) \cap F \neq \emptyset$，且 $\not\vdash \bigwedge(\Gamma' \backslash \{\psi^{q_j}\}) \rightarrow \neg\varphi_l$。

  这时，同定理 3.15 过程，构造节点 $\Gamma_{i,0}$～$\Gamma_{i,3}$。同样可以证明：$\Gamma_{i,0}$～$\Gamma_{i,3}$ 中的每个节点都保留在 $\mathcal{S}$ 中。再令 P' 是 $\mathcal{S}$ 中任意起使于 $\Gamma_{i,3}$，结束于某模态节点 Γ'' 的有穷路径，且 P' 中除 Γ'' 外再无其他模态节点，于是，必然存在某个 $q_f \in \delta(q_j, a_l) \cap F$，使得 $\psi^{q_f} \in \Gamma''_{[\tau']}$。

  令 $P_{i+1} = P[1, m] \cdot (\Gamma_{i,0}, \Gamma_{i,1}, \Gamma_{i,2}, \Gamma_{i,3}) \cdot P'[1..] \cdot (\Gamma'_{i,0}, \Gamma'_{i,1})$，于是，$\tau$ 在 $P_0 \cdot P_1 \cdot \ldots \cdot P_{i+1}$ 的保持拓延必然不是接收无关的。再令 $W_{i+1} = \{\tau'' \mid$ 存在某个 $\tau''' \in W_i$，使得 τ'' 是 τ''' 在 $P_0 \cdot P_1 \cdot \ldots \cdot P_{i+1}$ 的保持拓延$\}$。这样就有 $\#W_{i+1} < \#W_i$。

上述构造保证了：有无穷多个 $i \in \mathbb{N}$，使得 $W_i = \emptyset$。于是，$\widehat{P}$ 中任意的正自动机踪迹 τ 必然满足正一致性。否则，必然存在某个 $j \in \mathbb{N}$，使得 $\tau[j..]$ 是接收无关的，同时必然存在某个 $k \in \mathbb{N}$，使得对于任意的 $l > k$，存在 $\tau[i..]$ 的某个片段保留在 W_l 中。这样，就不会有无穷多个 W_i 为空集。

同时，由于 $\mathcal{S}$ 是连通子图，于是必然存在某个从初始节点 $\{\varphi\}$ 到达 $P(0)$ 的且满足局部一致性的有穷路径 $\widetilde{P}$。这样，就构造出了 $\mathcal{G}_\varphi$ 中满足一致性的完全路径，这与 φ 不可满足矛盾。所以，若将待证结论否定，则会导致矛盾，从而待证的结论成立。 □

定理 3.24 对于 $\mathscr{R}$ 系统而言，若 ETL_r 公式 φ 是不可满足的，则 $\vdash \neg\varphi$。

证明. 仍然采用“迁移图删除” 技术对其进行证明。同 $\mathscr{L}$ 及 $\mathscr{F}$ 中的过程类似，下面只说明如何对末端极大连通子图进行删除。

对于 $\mathcal{G}_\varphi$ 中任意一个由当前剩余节点构成的末端极大连通子图 $\mathcal{S}$，假设存在从初始节点 $\{\varphi\}$ 到 $\mathcal{G}_\varphi$ 中某节点满足局部一致性的有穷路径，同时，$\mathcal{S}$ 中所有含有互补对或者 $false$ 的节点都已经被删除，并且新生成的末端独立节点也已经被删除。

对于 $\mathcal{S}$ 中任意一条无穷路径 P，若 P 不满足负全局一致性约束，则 P 中必然存在一条无穷踪迹 τ，以及某个负自动机公式$\neg\phi$（不妨设 $\neg\phi = \neg\mathcal{A}(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, F \rangle$），以及某个 $q' \in F$，使得 $\neg\phi^{q'}$ 在 τ 中有无穷多次出现。对路径 P 做如下修改：

1. 必然存在某个 $i \in \mathbb{N}$，使得对于任意 $j \geq i$，$\tau(j)$ 在 τ 中都无穷多次出现。不妨设 $\tau(i)$ 所在的节点为 $P(k)$，那么将 P 修改为 $P[k..]$；同时，将 τ 修改为 $\tau[i..]$。在做了这样的修改后，对于每个 $j \in \mathbb{N}$，可以断言 $\tau(j)$ 所在的节点恰为 $P(j)$。
2. 对于任意的 $l, l' \in \mathbb{N}$（不妨设 $l < l'$）在当前所获得的路径 P 中，如果 $P[l] = P[l']$，$\tau(l) = \tau(l')$，并且对于任意的 $q' \in F$ 以及 $l \leq l'' < l'$ 都有 $\tau(l'') \neq \neg\phi^{q'}$，那么将当前的 P 修改为 $P[0..l] \cdot P[l'+1..]$；将当前的 τ 修改为 $\tau[0..l] \cdot \tau[l'+1..]$。

容易验证，最终得到的路径 P 仍是 $\mathcal{G}_\varphi$ 中一条违反负全局一致性的路径— 这是因为：对于负自动机公式 $\neg\phi^q$，若 $q \in F$，则该式在原来 τ 中的每次出现均得以保留。

于是，对（最后得到的）路径 P 中的每个状态节点 Γ 而言，$\Gamma_{[\tau]}$ 都是形如 $\neg\phi^q$ 的公式（$q \in Q$）。同时，对每个 $q_i \in Q$ 以及节点 Γ 而言，若存在某个 $k \in \mathbb{N}$，使得 $P(k) = \Gamma$ 且 $\tau(k) = \psi^{q_i}$，则按照如下方式构造一个 NBW $\widehat{\mathcal{A}}^{(\Gamma,q_i)} = \langle \{a_1, \ldots, a_n\}, \widehat{Q}, \widehat{\delta}, (\Gamma, q_i), \widehat{F} \rangle$，其中：

- $\widehat{Q} = \{(\Gamma', q_j) \mid \Gamma'$ 是 $\mathcal{S}$ 中的状态节点，$q_j \in Q$，并且存在 $k \in \mathbb{N}$，使得 $P(k) = \Gamma'$，$\tau(k) = \neg\phi^{q_j}\}$。
- 对于任意的 a_i，$(\Gamma_1, q_{k_1}) \in \widehat{Q}$，$(\Gamma_2, q_{k_2}) \in \widehat{Q}$，$(\Gamma_2, q_{k_2}) \in \widehat{\delta}((\Gamma_1, q_{k_1}), a_i)$ 当且仅当
  - 存在 $l_j \in \mathbb{N}$（$j = 1, 2$）其中 $l_1 < l_2$，并且 $P(l_j) = \Gamma_j$，$\tau(l_j) = \neg\phi^{q_{k_j}}$。
  - 在路径 $P[l_1..l_2]$ 中仅存在一个模态节点（换言之，$P[l_1..l_2]$ 是 $\mathcal{G}_\varphi$ 中的基本迁移路径）。
  - 存在某个 $l_1 < l' < l_2$，使得 $\tau(l') = \varphi_i \wedge \bigcirc\neg\phi^{q_{k_2}}$。
- $\widehat{F} = (2^{\mathcal{V}_\varphi} \times F) \cap \widehat{Q}$。

由 P 的构造过程知，$\widehat{\mathcal{A}}^{(\Gamma,q_i)}$ 的任一迁移环路中都包含某个接收状态，从而是极大的。同时，由 (nexp) 规则知，$(\Gamma_2, q_{k_2}) \in \widehat{\delta}((\Gamma_1, q_{k_1}), a_i)$ 蕴含 $q_{k_2} \in \delta(q_{k_1}, a_i)$。于是，若无穷字 w 能够被 $\widehat{\mathcal{A}}^{(\Gamma,q_i)}$ 以运行 $(\Gamma_0, q_{l_0}), (\Gamma_1, l_1), \ldots$ 接收，则 w 必然能够被 $\mathcal{A}^{q_i}$ 以运行 $q_{l_0}, q_{l_1}, \ldots$ 接收（注意，$(\Gamma_i, q_{l_i}) \in \widehat{F}$当且仅当 $q_{l_i} \in F$）。所以，必然有

$$\mathbf{L}(\widehat{\mathcal{A}}^{(\Gamma,q_i)}) \subseteq \mathbf{L}(\mathcal{A}^{q_i}) \tag{3.41}$$

成立。接下来，对每个 $(\Gamma, q_i) \in \widehat{Q}$，令

$$\psi_{\Gamma,i} = \bigwedge \Gamma \tag{3.42}$$

由推理性质以及重写规则 (nexp)，容易得到

$$\vdash \psi_{\Gamma,i} \rightarrow \bigvee\nolimits_{1 \leq k \leq n} (\varphi_k \wedge \bigvee\nolimits_{(\Gamma', q_j) \in \widehat{\delta}((\Gamma, q_i), a_k)} \bigcirc\psi_{\Gamma', q_j}) \tag{3.43}$$

因为每个 $\widehat{\mathcal{A}}^{(\Gamma,q_i)}$ 都是极大的，由 (PRep) 规则得

$$\vdash \psi_{\Gamma,i} \to \widehat{\mathcal{A}}^{(\Gamma,q_i)}(\varphi_1,\ldots,\varphi_n) \tag{3.44}$$

进而，由 (3.41) 式及 (Mono) 规则得

$$\vdash \psi_{\Gamma,i} \to \mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) \tag{3.45}$$

由 $\widehat{\mathcal{A}}^{(\Gamma,q_i)}$ 的构造知 $\neg\mathcal{A}^{q_i}(\varphi_1,\ldots,\varphi_n) = \neg\phi^{q_i} \in \Gamma$（注意：$\tau(k) \in P(k)$）。于是，由 (Tau) 公理和 (MP) 规则立即可以到 $\vdash \neg\bigwedge\Gamma$。同理，$P$ 中其他所有的状态节点都可以删除。接下来，删除新得到的末端独立节点以及不可达节点。

重复使用上述过程，直至 $\mathcal{S}$ 当前剩余的节点集合中不存在违反负全局一致性的路径。若此时 $\mathcal{S}$ 中尚未为空，则对于当前剩余的任意一个极大连通子图 $\mathcal{S}'$，定理 3.23 的前提条件成立。于是，存在某条迁移公平路径 P' 以及 P' 中的踪迹 τ'，NBW $\mathcal{A}' = \langle\{a'_1,\ldots,a'_{n'}\},\{q'_1,\ldots,q'_m\},\delta',q'_i,F'\rangle$，以及 ETL_r 公式 $\varphi'_1,\ldots,\varphi'_{n'}$ 使得

- $\theta^{q'_i} = \mathcal{A}'^{q'_i}(\varphi'_1,\ldots,\varphi'_m)$ 在 τ' 中出现无穷多次。
- 对于每个 $1 \le j \le m'$，以及 $\mathcal{S}'$ 中的每个状态节点 Γ，若 $\theta^{q'_j} \in \Gamma_{[\tau']}$，则 $\vdash \bigwedge\Gamma\backslash\{\theta^{q'_j}\} \to \bigwedge_{\delta'(q'_j,a'_l)\cap F'\neq\emptyset}\neg\varphi'_l$。

不失一般性，设 τ' 中每个公式都在其中无穷多次出现（否则，可以取 τ' 的某个后缀作为满足条件的踪迹）。对每个 $1 \le j \le m'$，若 $q'_j \notin F'$，则令 $\mathcal{S}'_j = \{\Gamma \mid \Gamma$ 是 $\mathcal{S}'$ 中的状态节点，且 $\theta^{q'_j} \in \Gamma_{[\tau']}\}$。同时，对每个 $1 \le j \le m'$，令

$$\psi'_j = \bigvee_{\Gamma\in\mathcal{S}'_j}\bigwedge(\Gamma\backslash\{\theta^{q'_j}\}) \vee \neg\theta^{q'_j} \tag{3.46}$$

同定理 3.16 中的证明过程完全类似，在 $\mathscr{R}$ 系统中，可以得到 $\vdash \psi'_j \to \neg\theta^{q'_j}$，即：

$$\vdash \psi'_j \to \neg\mathcal{A}'^{q'_j}(\varphi'_1,\ldots,\varphi'_{n'}) \tag{3.47}$$

（只是需要将原来使用 (Fin) 规则的地方替换为 (NRep)）。这样，对每个 $\Gamma \in \mathcal{S}'_j$ 而言，同时有 $\vdash \bigwedge\Gamma \to \theta^{q'_j}$ 以及 $\vdash \bigwedge\Gamma \to \neg\theta^{q'_j}$ 成立。于是，可以将整个 $\mathcal{S}'$ 删除。重复此过程，直至原来的极大连通子图 $\mathcal{S}$ 的所有节点均被删除。 □

讨论. 在定理 3.24 的证明过程中，用到了 (Mono) 公理。事实上，这条公理并不是可实例化的（见下节）。因此，该条公理的存在对于 ETL_r 逻辑片断的实例公理化是不利的。然而，回顾定理 3.24 的证明过程，在对 P 中的节点进行删除时，当

$\mathcal{A}(\varphi_1,\ldots,\varphi_n)$ 是极大的 NBW 时，事实上可以不使用 (Mono) 公理：同定理 3.9，对每个 $1\leq i\leq m$，令

$$\mathcal{S}_i=\{\Gamma\mid \Gamma\text{是 } P \text{ 中的状态节点，且 } \neg\phi^{q_i}\in\Gamma_{[\tau]}\}$$

以及

$$\psi_i=\bigvee_{\Gamma\in\mathcal{S}_i}\bigwedge\Gamma$$

则同样可以得到

$$\vdash\psi_i\rightarrow\bigvee_{1\leq k\leq n}\varphi_k\wedge\bigvee_{q_j\in\delta(q_i,a_k)}\bigcirc\phi^{q_j}\tag{3.48}$$

由于每个 $\mathcal{A}^{q_i}$ 都是极大的，由 (PRep) 有 $\vdash\psi_i\rightarrow\phi^{q_i}$。这样，对于每个 $\Gamma\in\mathcal{S}_i$，同时有 $\vdash\bigwedge\Gamma\rightarrow\phi^{q_i}$ 以及 $\vdash\bigwedge\Gamma\rightarrow\neg\phi^{q_i}$。于是，有 $\vdash\neg\bigwedge\Gamma$。这样 P 中所有的状态节点都可删除。

定理 3.25 ($\mathscr{R}$ 的完备性) 对 $\mathscr{T}$ 系统而言，若 $\models\varphi$，则 $\vdash\varphi$。

推论 3.26 设 φ 是有效的 ETL_r 公式，且 φ 中的所有连接子都是极大的 NBW，则 $\mathscr{R}$ 中存在关于 φ 的且不使用 (Mono) 公理的证明序列。

3.5 ETL逻辑片断的实例化公理化

前三节分别给出了 ETL_l、ETL_f 和 ETL_r 的可靠完备公理系统。这三类 ETL 都具有等价于 ω-正规语言的表达能力。这些逻辑中都包含无穷多个时序连接子，因此它们具有丰富的**逻辑片段**（或称**子逻辑**）。在本节，将介绍如何从 ETL 的公理系统（$\mathscr{L}$、$\mathscr{T}$、$\mathscr{R}$）中获得其逻辑片段可靠完备公理系统的方法。

定义 3.5.1 (时序连接子的自动机编码) 对于某（线性）时序逻辑中 n 元时序操作子 K，若存在某自动机连接子 $\mathcal{A}_{\mathsf{K}}$ 使得对于任意的 $\varphi_1,\ldots,\varphi_n$ 公式 $\mathsf{K}(\varphi_1,\ldots,\varphi_n)$ 和 $\mathcal{A}_{\mathsf{K}}(\varphi_1,\ldots,\varphi_n)$ 都逻辑等价，则称 $\mathcal{A}_{\mathsf{K}}$ 是 K 的一个**自动机编码**。 □

例 3.5.1 在 LTL 中，时序连接子 U 可以编码为 NFW $\mathcal{A}_{\mathsf{U}}=\langle\{a_1,a_2\},\{q_1,q_2\},\delta_{\mathsf{U}},q_1,\{q_2\}\rangle$，其中 $\delta_{\mathsf{U}}(q_1,a_1)=\{q_1\}$，$\delta_{\mathsf{U}}(q_1,a_2)=\delta_{\mathsf{U}}(q_2,a_1)=\delta_{\mathsf{U}}(q_2,a_2)=\{q_2\}$（如图 *3.4(a)* 所示）。这样，$\varphi_1\mathsf{U}\varphi_2$ 就与 ETL_f 公式 $\mathcal{A}_{\mathsf{U}}(\varphi_1,\varphi_2)$ 等价。同时，LTL 中的时序连接子 R（其定义式见公式 *(2.7)*）可以被 NLW $\mathcal{A}_{\mathsf{R}}=\langle\{a_1,a_2,a_3\},\{q_1,q_2\},\delta_{\mathsf{R}},q_1,-\rangle$ 编码。其中，$\delta_{\mathsf{R}}(q_1,a_1)=\{q_1\}$，$\delta_{\mathsf{R}}(q_1,a_2)=\delta_{\mathsf{R}}(q_2,a_3)=\{q_2\}$，$\delta_{\mathsf{R}}$ 在其余处的函数值为 $\emptyset$（如图 *3.4(b)* 所示）。容易验证，LTL 公式 $\varphi_1\mathsf{R}\varphi_2$ 等价于 ETL_l 公式 $\mathcal{A}_{\mathsf{R}}(\varphi_2,\varphi_1\wedge\varphi_2,true)$。 □

(a) 连接子 U 的 finite 自动机编码　　(b) 连接子 R 的 looping 自动机编码

图 3.4 时序连接子自动机编码示例

定义 3.5.2 (实例化公理系统) *在公理系统 $\mathscr{F}$、$\mathscr{L}$ 以及 $\mathscr{R}$ 中，存在若干适用于某类自动机（NFW、NLW、NBW）连接子的公理（如 (Expand) 公理）以及规则（如 (Fin) 规则）。对于任一 ETL 逻辑片段，可以通过编码其连接子的方式例化出针对该连接子的公理/规则。其中，被例化的 ETL 称为*基逻辑*。这样，就会得到关于该逻辑片段的公理系统，称为（相对于基逻辑的）*实例化公理系统*。* □

注意，由于 $\mathscr{F}$、$\mathscr{L}$、$\mathscr{R}$ 中均存在关于 ○ 的公理或者规则，因此本节主要关心时序连接子中至少包含有等价于 ○ 连接子（比如 LTL 中的 X 连接子）的 ETL 逻辑片段。并且，在生成实例化公理系统时，该连接子直接对应于基逻辑中的 ○。

同时，采用 ETL_r 作为基逻辑获取其逻辑片段的实例化公理系统时，又有如下要求：

1. (Mono) 不派生例化公理。
2. 该逻辑片段中连接子的编码都是极大的 NBW。

引入该限制的原因如 95 页的讨论：在使用“图删除” 技术构造 ETL_r 公式的证明序列时（见定理 3.24），如果该公式中存在非极大的自动机连接子，则需要根据迁移图构造一个新的自动机连接子。而这个连接子未必在该逻辑片段中存在，因而可能没有针对该连接子的例化公理/规则。同时，下面的引理保证该了引入这种限制后，逻辑片段的编码仍能够进行。

引理 3.27 *对于任意的 ETL_r 公式 φ，存在与之等价的 ETL_r 公式 φ'，并且 φ' 中所有的自动机连接子都是极大的NBW。*

证明. 由文 [30] 中结论知：存在 ETL_l 公式 φ'' 与 φ 等价。而对于 φ'' 中的每个 NLW 连接子 $\langle \Sigma, Q, \delta, q, - \rangle$，在 ETL_r 中都可替换为 NBW 连接子 $\langle \Sigma, Q, \delta, q, Q \rangle$，并且显然该连接子是极大的。因此，只需令 φ' 是在 φ'' 上完成上述替换所得到 ETL_r 公式即可。 □

例 3.5.2 *对于 LTL 而言，可以将算子 U 编码为 $\mathcal{A}_{\mathsf{U}}$（见例 3.5.1）。而由 (Acc) 知，公式 $\mathcal{A}_{\mathsf{U}}^{q_2}(\varphi_1, \varphi_2)$ 等价于 $true$。于是，可以由 (Expand) 例化出公理*

$$(\varphi_1 \mathsf{U} \varphi_2) \leftrightarrow (\varphi_1 \wedge \mathsf{X}(\varphi_1 \mathsf{U} \varphi_2) \vee \varphi_2) \qquad \text{(Until-Expand)}$$

这里，将第二个析取支 $\varphi_2 \wedge \mathsf{X}\mathcal{A}^{q_2}(\varphi_1, \varphi_2)$ 直接替换成 φ_2。由 (Fin) 可以例化出规则

$$\frac{\psi_1 \rightarrow ((\varphi_1 \rightarrow \mathsf{X}\psi_1) \wedge \neg\varphi_2)}{\psi_1 \rightarrow \neg(\varphi_1 \mathsf{U} \varphi_2)} \quad \text{(Until-Finite)}$$

注意，为了同 *LTL* 中的表示一致，$\bigcirc$ 被替换称为 X。同样，为一致起见，公理 (Next) 和 (Kri) 对应的 *LTL* 版本分别为 $\neg\mathsf{X}\varphi \leftrightarrow \mathsf{X}\neg\varphi$ 和 $\mathsf{X}(\varphi_1 \rightarrow \varphi_2) \leftrightarrow (\mathsf{X}\varphi_1 \rightarrow \mathsf{X}\varphi_2)$。此外，(XGen) 规则的后件写作 $\mathsf{X}\varphi$。 □

在上述示例中，公理 (Tau)、(Next)、(Kri)、(Until-Expand)、规则 (MP)、(XGen)、(Until-Finite) 构成了一个由 $\mathscr{F}$ 实例化得到的 LTL 公理系统。注意：由于 $\mathcal{A}^{q_2}_{\mathsf{U}}$ 不是必需的连接子，因此 (Acc) 没有例化公理。但是，在 (Until-Expand) 化简过程中却暗含使用了该公理。

现在，总结一下关于 ETL 片段的实例化公理方法的步骤:

1. 首先，选取一个适当的 ETL 作为基逻辑。
2. 将逻辑片段中的时序算子用合适的自动机连接子进行编码。
3. 实例化与自动及相关的公理以及规则，获得实例化公理系统。
4. 对例化出的公理/规则做适当的变形、化简。

关于实例化公理方法，有如下结论。

定理 3.28 给定某 *ETL* 逻辑片段，若其某个表达完备的连接子集合，除与 $\bigcirc$ 对应的连接子外，都能够用其基逻辑（ETL_l、ETL_f、ETL_r）中包含的自动机连接子编码。则由该基逻辑公理系统（$\mathscr{L}$、$\mathscr{F}$、$\mathscr{R}$）实例化得到的，必然是关于该片段的可靠完备公理系统。

证明. 由实例化公理方法所得系统的可靠性显然：因为由 ETL 公理系统例化出的公理和规则都是可靠的。现在只证明该公理系统的完备性。

为叙述方便，对采用自动机编码的连接子和原连接子不加区分。进而对编码后的公式和逻辑片段中原公式（比如 $\mathcal{A}_{\mathsf{U}}(\varphi_1, \varphi_2)$ 和 $\varphi_1 \mathsf{U} \varphi_2$）不加区分。同时，将基逻辑的公理系统称为“原系统”，将逻辑片段对应的实例化公理系统称为“派生系统”。

设 φ 是该逻辑片段中的有效公式，从而是其基逻辑中的有效公式。由原系统的完备性，存在 φ 在基逻辑中的证明序列 $\psi_0, \psi_1, \ldots, \psi_m = \varphi$。假设该证明序列也是经“图删除” 过程获得，从而派生系统中必然包含针对该连接子的例化公理/规则。由实例化编码的限制保证，该证明过程出现的连接子都是在 φ 中出现的。同时假定，上述证明序列中的公式同时做了与得到实例化公理系统时同样的化简调整。比如：若派生系统如例 3.5.2 所述，则每个 ψ_i 中形如 $\mathcal{A}^{q_2}_{\mathsf{U}}(\varphi_1, \varphi_2)$ 的子公式都被替换成了 $true$，以保证每次使用 (Until-Expand) 时都与派生系统中的形式相一致。

下面证明，对每个 $1 \leq i \leq m$ 而言，ψ_i 均可在派生系统中证出。

1. 若 ψ_i 在原系统中可由公理 (Tau)、(Next)、(Kri) 得到，则由于这些公理都保留在派生系统中，所以 ψ_i 在派生系统中仍可由相应的公理得到。
2. 若 ψ_i 在原系统中可由某关于自动机连接子的公理得到，则在派生系统中必然存在关于该连接子的例化公理。于是，ψ_i 在派生系统中可以由该例化出的公理得到。
3. 若存在某个 $i' < i'' < i$，使得 ψ_i 在原系统中可由 $\psi_{i'}$、$\psi_{i''}$ 及 (MP) 规则得到，则在派生系统中，ψ_i 仍能由 $\psi_{i'}$、$\psi_{i''}$ 及 (MP) 规则得到。
4. 若存在 $i' < i$，使得 ψ_i 在原系统中可由 $\psi_{i'}$ 以及 (XGen) 规则得到。则在派生系统中，ψ_i 仍能由 $\psi_{i'}$ 以及 (XGen) 规则得到。
5. 若存在 $i_1 < i_2 < \ldots < i_k < i$，使得 ψ_i 是由 $\psi_{i_1}, \psi_{i_2}, \ldots, \psi_{i_k}$ 以及原系统中的某关于自动机连接子的规则获得。则在派生系统中该规则必然存在关于该连接子的例化规则。于是，在派生系统中，ψ_i 必然可以由 $\psi_{i_1}, \psi_{i_2}, \ldots, \psi_{i_k}$ 以及该例化规则得到。

注意到 $\psi_m = \varphi$，于是 φ 可证。这意味着派生系统的完备性。 □

应当指出的是：采用实例化公理方法时，最终得到的公理系统的形式对基逻辑以及参与编码时序连接子的十分敏感。

例 3.5.3 例如，*LTL* 同时也可以看作是 ETL_l 的片段。这时，为了能够采用 *looping* 自动机对时序连接子进行编码，需要将 *LTL* 看作是包含算子 X 和 R 的 ETL_l 逻辑片段，而要将 U 看作派生连接子。设 $\mathcal{A}_{\mathsf{R}}$ 是例 *3.5.1* 所示的 *NLW*（如图 *3.4(b)* 所示），则由 $\mathscr{L}$ 中的 (Expand) 公理可得

$$\begin{aligned}\mathcal{A}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true) \leftrightarrow \quad & (\varphi_2 \wedge \mathsf{X}\mathcal{A}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)) \\ & \vee(\varphi_1 \wedge \varphi_2 \wedge \mathsf{X}\mathcal{A}^{q_2}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true))\end{aligned} \tag{3.49}$$

以及

$$\mathcal{A}^{q_2}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true) \leftrightarrow (true \wedge \mathsf{X}\mathcal{A}^{q_2}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)) \tag{3.50}$$

由 (Loop) 规则可得

$$\frac{\begin{array}{ll} & (\psi_1 \rightarrow ((\varphi_2 \wedge \mathsf{X}\psi_1) \vee (\varphi_1 \wedge \varphi_2 \wedge \mathsf{X}\psi_2))) \\ \wedge & (\psi_2 \rightarrow (true \wedge \mathsf{X}\psi_2)) \end{array}}{\begin{array}{ll} & (\psi_1 \rightarrow \mathcal{A}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)) \\ \wedge & (\psi_2 \rightarrow \mathcal{A}^{q_2}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)) \end{array}} \tag{3.51}$$

令 $\phi_1 = false$，$\phi_2 = \neg\mathcal{A}^{q_2}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$，于是:

1. $\vdash \phi_1 \rightarrow ((\varphi_2 \wedge \mathsf{X}\phi_1) \vee (\varphi_1 \wedge \varphi_2 \wedge \mathsf{X}\phi_2))$ ⟅(Tau)⟆

2. $\vdash \phi_2 \rightarrow \mathsf{X}\phi_2$ 即:

$\vdash \phi_2 \rightarrow (true \wedge \mathsf{X}\phi_2)$ ⟅(3.50)、(Tau)、(Next)⟆

3. $\vdash \phi_2 \rightarrow \mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ 即:

$\vdash \neg\mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true) \rightarrow \mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ ⟅*1*、*2*、(Loop)⟆

4. $\vdash (\neg\mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true) \rightarrow \mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)) \rightarrow \mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ ⟅(Tau)⟆

5. $\vdash \mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ ⟅*3*、*4*、(MP)⟆

于是，可以将公式$\mathcal{A}_{\mathsf{R}}^{q_2}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ 替换为 $true$，再 *(3.49)* 中的 $\mathcal{A}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ 替换为 $\varphi_1\mathsf{R}\varphi_2$ 并化简，即可得到 (Expand) 关于 R 的例化公理:

$$(\varphi_1\mathsf{R}\varphi_2) \leftrightarrow ((\varphi_2 \wedge \mathsf{X}(\varphi_1\mathsf{R}\varphi_2)) \vee (\varphi_1 \wedge \varphi_2)) \qquad \text{(Releases-Expand)}$$

同样，将公式 *(3.51)* 的中的 $\mathcal{A}_{\mathsf{R}}(\varphi_2, \varphi_1 \wedge \varphi_2, true)$ 替换为 $\varphi_1\mathsf{R}\varphi_2$ 后，可以将 ψ_2 放松为 $true$。于是，化简得到 (Loop) 关于 R 的例化规则:

$$\frac{\psi_1 \rightarrow ((\varphi_2 \wedge \mathsf{X}\psi_1) \vee (\varphi_1 \wedge \varphi_2))}{\psi_1 \rightarrow (\varphi_1\mathsf{R}\varphi_2)} \qquad \text{(Releases-Loop)}$$

这样，就可以得到从 $\mathscr{L}$ 例化而来的 *LTL* 公理系统。 □

下面给出一个使用 ETL_r 作为基逻辑进行进行实例化的示例。

例 3.5.4 若希望从 $\mathscr{R}$ 实例化得到 *LTL* 的公理系统，必须选用能够被极大 *NBW* 编码的时序连接子。这里采用 W(*Weak Until*)和 G 作为其起初连接子。其中

$$\varphi_1\mathsf{W}\varphi_2 \stackrel{\text{def}}{=} (\varphi_1\mathsf{U}\varphi_2) \vee \mathsf{G}\varphi_1 \qquad (3.52)$$

这时，U 可以看作派生连接子，因为很容易检验

$$(\varphi_1\mathsf{U}\varphi_2) \leftrightarrow ((\varphi_1\mathsf{W}\varphi_2) \wedge \neg\mathsf{G}\neg\varphi_2) \qquad (3.53)$$

是有效的。

考虑 *NBW* $\mathcal{A}_{\mathsf{W}} = (\{a_1, a_2, a_3\}, \{q_1, q_2\}, \delta_{\mathsf{W}}, q_1, \{q_1, q_2\})$ 以及 $\mathcal{A}_{\mathsf{G}} = (\{q_1\}, \{a_1\}, \delta_{\mathsf{G}}, q_1, \{q_1\})$，其中 $\delta_{\mathsf{W}}(q_1, a_1) = \{q_1\}$，$\delta_{\mathsf{W}}(q_1, a_2) = \{q_2\}$，$\delta_{\mathsf{W}}(q_2, a_3) = \{q_2\}$，$\delta_{\mathsf{W}}(q_1, a_3) = \delta_{\mathsf{W}}(q_2, a_1) = \delta_{\mathsf{W}}(q_2, a_2) = \emptyset$，$\delta_{\mathsf{G}}(q_1, a_1) = \{q_1\}$。这两个 *NBW* 都是极大的，并且容易检验 $\varphi_1\mathsf{W}\varphi_2$ 和 $\mathsf{G}\varphi$ 分别等价于 $\mathcal{A}_{\mathsf{W}}(\varphi_1, \varphi_2, true)$ 和 $\mathcal{A}_{\mathsf{G}}(\varphi)$。

这样，(Expand) 关于 W 和 G 例化出的公理分别为

$$(\varphi_1 \mathsf{W} \varphi_2) \leftrightarrow ((\varphi_1 \wedge \mathsf{X}(\varphi_1 \mathsf{W} \varphi_2)) \vee \varphi_2) \qquad \text{(WUntil-Expand)}$$

$$\mathsf{G}\varphi \leftrightarrow (\varphi \wedge \mathsf{X}\mathsf{G}\varphi) \qquad \text{(Global-Expand)}$$

(PRep) 关于 W 和 G 的例化规则分别为

$$\frac{\psi_1 \rightarrow (\varphi_1 \wedge \mathsf{X}\psi_1 \vee \varphi_2)}{\psi_1 \rightarrow (\varphi_1 \mathsf{W} \varphi_2)} \qquad \text{(WUntil-PRep)}$$

$$\frac{\psi_1 \rightarrow (\varphi \wedge \mathsf{X}\psi_1)}{\psi_1 \rightarrow \mathsf{G}\varphi} \qquad \text{(Global-PRep)}$$

注意：由于 $\mathcal{A}_{\mathsf{W}}$ 和 $\mathcal{A}_{\mathsf{G}}$ 中的每个状态都是终止状态，故而 (NRep) 没有关于 W 和 G 的例化规则（获得上述公理、规则的中间过程略）。□

最后，再以一个具有无穷多个时序算子的逻辑片段为例，介绍如何使用实例化方法获取其公理系统。

例 3.5.5 考虑这样一种时序逻辑：除 $\bigcirc$ 之外，对每个 $k > 1$ 引入了一个时序连接子 P^k。其中，对任意的线性结构 π 以及 $i \in \mathbb{N}$:

- $\pi, i \models \mathsf{P}^k\varphi$ 当且仅当对于任意的 $j \in \mathbb{N}$ 有 $\pi, i + j \times k \models \varphi$。

这样的时序逻辑可以用来规约“周期性质”。要得到该逻辑的公理系统，可以将其视为 ETL_l 的逻辑片段。

对每个 $k > 1$，可以构造一个 NLW $\mathcal{A}_k = \langle \{a_1, a_2\}, \{q_1, \ldots, q_k\}, \delta_k, q_1, - \rangle$，其中:

$$\delta_k(q_i, a_1) = \begin{cases} \{q_2\} & \text{，若 } i = 1 \\ \emptyset & \text{，否则} \end{cases} \quad ; \qquad \delta_k(q_i, a_2) = \begin{cases} \emptyset & \text{，若 } i = 1 \\ \{q_{i+1}\} & \text{，若 } 1 < i < k \\ \{q_1\} & \text{，若 } i = k \end{cases} \quad 。$$

于是，$\mathsf{P}^k\varphi$ 等价于 $\mathcal{A}_k^{q_0}(\varphi, true)$。于是，由 (Expand)，对每个 $1 \leq i < k$ 有：$\mathcal{A}_k^{q_i}(\varphi, true) \leftrightarrow \bigcirc\mathcal{A}_k^{q_{i+1}}(\varphi, true)$，并且 $\mathcal{A}_k^{q_{k-1}}(\varphi, true) \leftrightarrow \bigcirc\mathsf{P}^k\varphi$。这样，就可以将

$$\mathsf{P}^k\varphi \leftrightarrow (\varphi \wedge \bigcirc^k \mathsf{P}^k\varphi) \qquad \text{(P-Expand)}$$

作为 (Expand) 实例的最终形式。其中 $\bigcirc^k$ 为 k 个连续 $\bigcirc$ 连接子的缩写。同样的，规则 (Loop) 的实例为

$$\frac{\psi \rightarrow (\varphi \wedge \bigcirc^k \psi)}{\psi \rightarrow \mathsf{P}^k\varphi} \qquad \text{(P-Loop)}$$

相比较而言，如果采用 ETL_f 作为该逻辑的基逻辑，实例化过程将会复杂得多。□

3.6 本章小结

在本章中，分别给出了关于三类扩展时序逻辑（ETL_l、ETL_f、ETL_r）的公理系统（$\mathscr{L}$、$\mathscr{F}$、$\mathscr{R}$），并证明了这三个系统的可靠性和完备性。这些系统中包含若干刻画自动机时序连接子的公理和规则，而这些公理和规则分别刻画了 looping、finite、repeating 自动机连接子的公共时序性质。

此外，本章还给出了一种针对各种 ETL 逻辑片段的“实例化公理方法”。采用该种方法，可以获得各类 ETL 逻辑片段的可靠、完备公理系统。虽然运用该过程得到的公理系统对所采用的基逻辑和用来编码的连接子十分敏感，并且有时需要做一定的变换（消除额外的连接子）才能得到较为简洁的公理系统。但是相对于完全从头开发一套公理系统，其难度和代价要小的多。

本章工作主要针对采用非确定自动机连接子的 ETL 进行，同时主要考虑了 looping、finite、repeating 这三类接收条件。虽然 ETL_l、ETL_f、ETL_r 这三类 ETL 具有完备的 ω-正规表达能力，但是研究更为一般的 ETL 的公理化问题有可能得到其他有价值的、刻画更一般性质的公理、规则。因此，将来的工作包括两个方面：一是研究具有交错迁移结构以及更一般接收条件自动机连接子的 ETL 的公理化问题；二是研究其逻辑片段的实例公理化技术。

第四章　基于博弈的 μ-演算公理化

4.1　引言

在前一章，给出了关于三类 ETL（ETL_l、ETL_f、ETL_r）的公理系统。这三类 ETL 中包含无穷多个时序连接子，能够表达全部的 ω-正规性质。另外一类重要的时序逻辑是 μ-演算，它包括分支时间（即：模态 μ-演算）和线性时间（即：线性 μ-演算）两个版本。

关于 μ-演算公理化问题，在文 [55, 105, 56, 34, 52, 53, 54] 等文献中已经给出了深入的研究和讨论。然而，对于公理化问题而言，采用不同的方法往往可以得到不同的公理系统。它能够从不同的侧面刻画逻辑的本质。此外，即使对于同一个公理系统，利用不同的方法证明其完备性时，往往能够反应人们对该系统的不同理解。

比如，在 2001 年，Lange 和 Stirling 基于博弈理论给出了新的 CTL 和 LTL 的公理系统[35]。这两个系统分别描述了 Release 时序连接子（R）的可满足特征。Lange-Stirling 的 LTL 公理系统中，包含一条推理规则 (Rel)（见公式(1.6)），该公式反映了当公式可满足（resp. 不可满足）时参与者 0（resp. 参与者 1）所执行的取胜策略（见文 [35]）。

采用博弈方法证明公理系统的完备性时过程相对直观、简洁。本章介绍如何将基于 Game 的公理系统完备性证明的方法推广至模态 μ-演算和线性 μ-演算。即：

- 定义关于 μ-演算的博弈格局迁移规则以及对应的取胜条件。证明博弈规则的可靠性以及完备性。即："句子 φ 是可满足的（resp.不可满足的）当且仅当参与者 0（resp. 参与者 1）持有在 $\mathcal{G}_\varphi$ 上的取胜策略"。从而给出了一种关于 μ-演算公式可满足性的测试方法。
- 将给出 μ-演算博弈求解的复杂度。即：模态 μ-演算博弈的求解复杂度是 EXPTIME-complete 的；线性 μ-演算博弈的求解复杂度是 PSPACE-complete 的。
- 模态 μ-演算的公理系统（$\mathscr{G}$）早在上个世纪 80 年代初就由 Kozen 给出，但是其完备性直至上世纪 90 年代中期才被 Walukiewicz 用一种十分复杂的技术证明。线性 μ-演算的可靠完备公理系统（$\mathscr{H}$）由 Kaivola 于上世纪 90 年代给出，其完备性证明是基于推演表方法进行的，过程也十分复杂。作为博弈方法的应用，本章分别利用针对模态 μ-演算和线性 μ-演算的博弈方法给出 $\mathscr{G}$ 系统和 $\mathscr{H}$ 系统完备性的相对简洁的证明。

本章内容组织如下：

1. 4.2 节介绍模态 μ-演算的博弈以及基于博弈方法给出 Kozen-Walukiewicz 的模态 μ-演算公理系统完备性证明。主要包括对模态 μ-演算若干相关概念的定义（如：良命名公式、公式的“Fischer-Ladner” 闭包、受卫公式）；介绍模态 μ-演算的判定模型— 交错 parity 树自动机。定义模态 μ-演算的博弈格局迁移规则、取胜条件，并证明其可靠性及完备性；给出模态 μ-演算的博弈的确定性问题以及求解复杂度；最后给出基于模态 μ-演算博弈方法证明了公理系统 $\mathscr{G}$ 的完备性。
2. 4.3 节介绍线性 μ-演算的博弈的公理化问题。主要介绍线性 μ-演算博弈格局迁移规则、取胜条件、可靠性、完备性，给出其求解复杂度。最后，利用该理论给出线性 μ-演算公理系统完备性的证明。该过程可以看作是模态 μ-演算相应结论的特例。

4.2 模态 μ-演算的博弈系统及公理系统

4.2.1 模态 μ-演算的相关概念

模态（分支时间）μ-演算基本语法与语义的形式定义分别见第 2 章中的定义 2.2.17 及定义 2.2.18。本节进一步介绍几个有关 μ-演算公式的概念，这些概念将会在模态 μ-演算的博弈系统中使用。同时，将介绍有关模态 μ-演算的判定模型— parity 树自动机的相关概念。

4.2.1.1 公式范式及迭代封闭式

定义 4.2.1 (良命名公式) 称模态 μ-演算公式 φ 是**良命名**的，如果 φ 的不同约束变元之间不重名；自由变元和约束变元之间不同名。 □

例 4.2.1 考虑下面三个模态 μ-演算公式:

$$\begin{aligned}\varphi_1 &= (p_1 \wedge \neg X) \vee \mu X.(X \wedge \Box Y \wedge \nu X.(X \vee \neg p_2))\\ \varphi_2 &= (p_1 \wedge \neg Y) \vee \mu X.(X \wedge \Box Y \wedge \nu X.(X \vee \neg p_2))\\ \varphi_3 &= (p_1 \wedge \neg Y) \vee \mu X.(X \wedge \Box Y \wedge \nu Z.(Z \vee \neg p_2))\end{aligned}$$

由于 φ_1 中同时存在自由公式变元 X 和约束公式变元 X，所以 φ_1 不是良命名的；由于 φ_2 中有两个重名的约束变元 X，因而它也不是良命名的；按照定义，φ_3 是一个良命名的公式。 □

定义 4.2.2 (约束变元绑定式) 设 φ 是一个良命名的模态 μ-演算公式，则对于 φ 中的任一约束变元 X 而言，在 φ 中都存在唯一的形如 $\mu X.\psi$ 或者 $\nu X.\psi$ 子公式将其绑定，该子公式称为 X 在 φ 中的**绑定式**，记作 $\mathbf{D}_\varphi(X)$。 □

例 4.2.2 设公式 $\varphi = (p_1 \wedge \neg Y) \vee \mu X.(X \wedge \Box Y \wedge \nu Z.(Z \vee \neg p_2))$，则 $\mathbf{D}_\varphi(X) = \mu X.(X \wedge \Box Y \wedge \nu Z.(Z \vee \neg p_2))$，$\mathbf{D}_\varphi(Z) = \nu Z.(Z \vee \neg p_2)$。 □

定义 4.2.3 (约束变元层次关系) 给定良命名公式 φ，以及 φ 中的约束变元 X、Y，若 $\mathbf{D}_\varphi(Y)$ 是 $\mathbf{D}_\varphi(X)$ 的子公式，则称（在公式 φ 中） X **位于** Y **的外层**，记作 $Y \lhd_\varphi X$。 □

例 4.2.3 在公式 $\varphi = \mu X.(p_1 \vee \Box \nu Y.(p_2 \wedge X \wedge \Diamond Y))$ 中，有 $Y \lhd_\varphi X$ 成立。 □

引理 4.1 设 φ 是良命名的公式，则 $\lhd_\varphi$ 是“传递的”和“向上可比较”的。即：

- 若 $X \lhd_\varphi Y$，且 $Y \lhd_\varphi Z$ 则 $X \lhd_\varphi Z$。
- 若 $X \lhd_\varphi Y$，且 $X \lhd_\varphi Z$，则 Y 和 Z 关于 $\lhd_\varphi$ 是可比较的（即：$Y \lhd_\varphi Z$ 和 $Z \lhd_\varphi Y$ 二者之中必有一个成立）。

证明. 传递性的证明很容易：由定义可得 $\mathbf{D}_\varphi(X)$ 是 $\mathbf{D}_\varphi(Y)$ 的子公式，$\mathbf{D}_\varphi(Y)$ 是 $\mathbf{D}_\varphi(Z)$ 的子公式。于是，$\mathbf{D}_\varphi(X)$ 必是 $\mathbf{D}_\varphi(Z)$ 的子公式。对于向上传递性，假设 $\mathbf{D}_\varphi(Y)$ 和 $\mathbf{D}_\varphi(Z)$ 互不为子公式，由于 φ 是良命名的，它们便不可能有公共的子公式 $\mathbf{D}_\varphi(X)$，从而导致矛盾。事实上，“向上传递性” 说明了 $\lhd_\varphi$ 的关系图一定是一棵树。 □

定义 4.2.4 (受卫公式) 称良命名模态 μ-演算公式 φ 是**受卫**的，如果 φ 中的每个约束变元 X 都满足：“X 在 $\mathbf{D}_\varphi(X)$ 中的每个出现都在 $\Diamond$ 或者 $\Box$ 的辖域中”，则称 φ 是**受卫公式**。 □

例 4.2.4 公式 $\mu X.(p \wedge X \vee \Box \nu Y.(Y \wedge X))$ 不是受卫的；而公式 $\nu X.(\mu Y.(p \vee \Diamond Y) \wedge \Box X)$ 是受卫的。 □

引理 4.2 (Kozen-Walukiewicz) 每个模态 μ-演算公式都存在与之等价的受卫公式。

这里使用 Walukiewicz 在文 [34] 中给出的方法。具体过程见 4.2.3 节中定理 4.22。

定义 4.2.5 (公式的否定范式) 给定公式 φ、反复利用德摩根律、$\neg\Box\psi \leftrightarrow \Diamond\neg\psi$、$\neg\Diamond\psi \leftrightarrow \Box\neg\psi$、$\neg\mu Y.\psi \leftrightarrow \nu Y.\neg\psi^Y_{\neg Y}$、$\neg\nu Y.\psi \leftrightarrow \mu Y.\neg\psi^Y_{\neg Y}$ 以及 $\neg\neg\psi \leftrightarrow \psi$ 等模式对 φ 做等价变形，使得 $\neg$ 仅出现在原子命题以及自由变元之前。最终形式称为 φ 的**否定范式**。 □

例 4.2.5 公式 $\neg\mu X.(\Diamond\neg p_1 \vee \mu Y.(p_2 \vee \Diamond Y) \vee X)$ 的否定范式是 $\nu X.(\Box p_1 \wedge \nu Y.(\neg p_2 \wedge$

$\Box Y) \wedge X)$。 □

定义 4.2.6 (约束变元类型) 设良命名公式 φ 对应的否定范式为 φ'，X 是 φ 中的一个约束变元。则，称 X 为 φ 的 **μ-型约束变元**（*resp.* **ν-型约束变元**），当且仅当 $\mathbf{D}_{\varphi'}(X)$ 是形如 $\mu X.\psi$（*resp.* $\nu X.\psi$）的公式。 □

例 4.2.6 注意，约束变元的类型一定是针对约束变元的绑定式在原公式否定范式中的形式而定的。比如：若 $\varphi = \neg\mu X.(p \vee \Diamond X)$，则 X 实际上是 φ 的一个 ν-型约束变元。 □

定义 4.2.7 (博弈范式) 如果公式 φ 是良命名的、受卫的、且是否定范式，则称 φ 是一个**博弈范式**。

引理 4.3 任意的模态 μ-演算公式 φ 都能化为等价的博弈范式。

证明. 首先，根据引理 4.2，将其化为受卫公式。其次，对于 φ 中的任一子公式 $\mu Y.\psi$（resp. $\nu Y.\psi$），容易证明："若 Z 是不在 $\mu X.\psi$（resp. $\nu X.\psi$）中出现的变元，则 $\mu Y.\psi$（resp. $\nu Y.\psi$）等价于 $\mu Z.\psi_Z^Y$（resp. $\nu Z.\psi_Z^Y$）"。于是，φ 与 $\varphi_{\mu Z.\psi_Z^Y}^{\mu Y.\psi}$（resp. $\varphi_{\nu Z.\psi_Z^Y}^{\nu Y.\psi}$）等价。反复利用此过程，重命名 φ 中的约束变元，直至其中没有重名的约束变元以及没有与自由变元重名的约束变元。最后，将其转化为否定范式即可。 □

定义 4.2.8 (Fischer-Ladner 闭包 [106]) 设 φ 是某写成博弈范式的公式，则 φ 的 *Fischer-Ladner* 闭包 $\mathbf{Cl}(\varphi)$ 是满足下列约束的最小集合：

- $\varphi \in \mathbf{Cl}(\varphi)$;
- 若 $\varphi_1 \wedge \varphi_2 \in \mathbf{Cl}(\varphi)$ 或者 $\varphi_1 \vee \varphi_2 \in \mathbf{Cl}(\varphi)$，则 $\varphi_1 \in \mathbf{Cl}(\varphi)$，且 $\varphi_2 \in \mathbf{Cl}(\varphi)$;
- 若 $\Box\psi \in \mathbf{Cl}(\varphi)$ 或者 $\Diamond\psi \in \mathbf{Cl}(\varphi)$ 则 $\psi \in \mathbf{Cl}(\varphi)$;
- 若 $\mu X.\psi \in \mathbf{Cl}(\varphi)$，则 $\psi_{\mu X.\psi}^X \in \mathbf{Cl}(\varphi)$；若 $\nu X.\psi \in \mathbf{Cl}(\varphi)$，则 $\psi_{\nu X.\psi}^X \in \mathbf{Cl}(\varphi)$。 □

例 4.2.7 设公式 $\varphi = \mu X.(p \vee \Box\nu Y.(X \wedge \Diamond Y))$，则 $\mathbf{Cl}(\varphi) = \{\varphi, p \vee \Box\nu Y.(\varphi \wedge \Diamond Y), p, \Box\nu Y.(\varphi \wedge \Diamond Y), \nu Y.(\varphi \wedge \Diamond Y), \varphi \wedge \Diamond\nu Y.(\varphi \wedge \Diamond Y), \Diamond\nu Y.(\varphi \wedge \Diamond Y)\}$。 □

易证，$\#\mathbf{Cl}(\varphi) \in \mathcal{O}(|\varphi|)$，其中 $|\varphi|$ 为 φ 的长度（即：φ 的子公式数目）。若 φ 是句子，则 $\mathbf{Cl}(\varphi)$ 中的每个公式都是句子。同时注意：$\mathbf{Cl}(\varphi)$ 中的公式可能并不是良命名的（如上例中的公式 $\nu Y.(\varphi \wedge \Diamond Y)$，约束变元 Y 在 φ 中也有出现）。因此，有时会根据需要将 $\mathbf{Cl}(\varphi)$ 中公式的约束变元重命名。

定义 4.2.9 (公式的迭代封闭式) 设 φ 是句子（并假设其已写成博弈范式），则对于 φ 的任意一子公式 ψ，都存在 $\psi' \in \mathbf{Cl}(\varphi)$ 与之对应。其中 ψ' 按如下过程得到：

1. 令 $\psi'_0 = \psi$。
2. 若 ψ'_i 是句子，则可断言 $\psi'_i \in \mathbf{Cl}(\varphi)$，这时，令 $\psi' = \psi'_i$ 停止迭代；否则，任取 ψ'_i 中的自由变元 X，令 $\psi'_{i+1} = (\psi'_i)^X_{\mathbf{D}_\varphi(X)}$。

称 ψ' 为 ψ（关于 φ）的**迭代封闭式**，记作 $\psi' = \mathbf{IC}_\varphi(\psi)$。 □

例 4.2.8 在例 *4.2.7* 中，$\mathbf{IC}_\varphi(p) = p$, $\mathbf{IC}_\varphi(X) = \varphi$, $\mathbf{IC}_\varphi(Y) = \nu Y.(\varphi \wedge \Diamond Y)$, $\mathbf{IC}_\varphi(p \vee \Box\nu Y.(X \wedge \Diamond Y)) = p \vee \Box\nu Y.(\varphi \wedge \Diamond Y)$。 □

同样，有时也需要对 $\mathbf{IC}_\varphi(\psi)$ 进行约束变元重命名。需要注意的是：一般而言，函数 $\mathbf{IC}_\varphi$ 并不是双射。比如，在例 4.2.7 中有 $\mathbf{IC}_\varphi(X) = \mathbf{IC}_\varphi(\mathbf{D}_\varphi(X))$。进一步，对该函数，有如下性质成立：

1. 若 $p \in AP$，则 $\mathbf{IC}_\varphi(p) = p$； $\mathbf{IC}_\varphi(\neg p) = \neg p$。
2. 若 X 是 φ 中的自由变元，则 $\mathbf{IC}_\varphi(X) = X$；若 X 是 φ 中的约束变元，则 $\mathbf{IC}_\varphi(X) = \mathbf{IC}_\varphi(\mathbf{D}_\varphi(X))$。
3. $\mathbf{IC}_\varphi(\varphi_1 \vee \varphi_2) = \mathbf{IC}_\varphi(\varphi_1) \vee \mathbf{IC}_\varphi(\varphi_2)$； $\mathbf{IC}_\varphi(\varphi_1 \wedge \varphi_2) = \mathbf{IC}_\varphi(\varphi_1) \wedge \mathbf{IC}_\varphi(\varphi_2)$。
4. $\mathbf{IC}_\varphi(\Box\psi) = \Box\mathbf{IC}_\varphi(\psi)$； $\mathbf{IC}_\varphi(\Diamond\psi) = \Diamond\mathbf{IC}_\varphi(\psi)$。
5. 设 p 是不在 φ 中出现的原子命题，则 $\mathbf{IC}_\varphi(\mu X.\psi) = \mu X.(\mathbf{IC}_\varphi(\psi_p^X)_X^p)$； $\mathbf{IC}_\varphi(\nu X.\psi) = \nu X.(\mathbf{IC}_\varphi(\psi_p^X)_X^p)$。

4.2.1.2 模态 μ-演算的判定模型：parity 树自动机

在 2.1 中介绍了无穷字上的 ω-自动机。为研究模态 μ-演算的判定问题以及为模态 μ-演算博弈的可靠性、完备性证明做铺垫，需要介绍一类特殊的树自动机，称为“**parity 树自动机**”。同字自动机相比，树自动机识别的语言是树的集合，而不再是字的集合。该种类型自动机由 Wilke 在文 [65] 中提出，其具体定义如下。

定义 4.2.10 (交错 parity 树自动机) 一个**交错 *parity* 树自动机**（*Alternating Parity Tree Automaton*，简写为 *APT*）是一个四元组 $\mathcal{A} = \langle Q, \delta, q, \Omega\rangle$，其中：

- Q 是一个有穷**状态集合**。
- $\delta : Q \to \chi(Q)$，是一个**迁移函数**。其中：$\chi(Q)$ 是满足下列条件的最小集合：
  - $true \in \chi(Q)$； $false \in \chi(Q)$；
  - 若 $p \in AP$，则 $p \in \chi(Q)$，$\neg p \in \chi(Q)$；
  - 若 $q', q_1, q_2 \in Q$，则 $q', \Box q', \Diamond q', q_1 \wedge q_2, q_1 \vee q_2 \in \chi(Q)$。
- $q \in Q$，是一个**初始状态**。
- 接收条件 Ω 一个部分函数，即：$\Omega : Q \rightsquigarrow \mathbb{N}$。 □

注意，该种自动机不强调字母表，原因在于其具有一个默认的字母表 2^{AP}。此外，应当注意无穷字上的自动机与该类树自动机迁移函数形式的区别。

定义 4.2.11 (计算树在 APT 上的运行) 设 $\langle T, \rho\rangle$ 是一个计算树，其中 $\rho : T \to 2^{AP}$，$\mathcal{A} = \langle Q, \delta, q_0, \Omega\rangle$ 是一个交错 *parity* 树自动机。则 $\mathcal{A}$ 在 $\langle T, \rho\rangle$ 上的一个**运行**

是一个 $Q \times T$-标记树 $\langle \hat{T}, \hat{\rho} \rangle$，其中：

- $\hat{\rho}(\epsilon) = (q_0, \epsilon)$。
- 对于任意的 $x \in \hat{T}$，设 $\hat{\rho}(x) = (q, y)$，则：
  - $\delta(q) \neq false$。
  - 若 $\delta(q) = true$，则不对 q 和 y 做任何限制。
  - 若 $\delta(q) = p$，则 $p \in \rho(y)$；若 $\delta(q) = \neg p$，则 $p \notin \rho(y)$。
  - 若 $\delta(q) = q'$，则 x 在 $\hat{T}$ 中有唯一的子节点 $x \cdot 0$，且 $\hat{\rho}(x \cdot 0) = (q', y)$。
  - 若 $\delta(q) = \Box q'$，则对于每个 $c \in \mathbb{N}$，如果 $y \cdot c \in T$，那么就有 $x \cdot c \in \hat{T}$，且 $\hat{\rho}(x \cdot c) = (q', y \cdot c)$。
  - 若 $\delta(q) = \Diamond q'$，则 x 在 $\hat{T}$ 中有唯一的子节点 $x \cdot 0$，并且 y 在 T 中存在某个子节点 $y \cdot c$，使得 $\hat{\rho}(x \cdot 0) = (q', y \cdot c)$。
  - 若 $\delta(q) = q_1 \wedge q_2$，则 x 在 $\hat{T}$ 有两个子节点 $x \cdot 0$ 与 $x \cdot 1$，其中 $\hat{\rho}(x \cdot 0) = (q_1, y)$，$\hat{\rho}(x \cdot 1) = (q_2, y)$。
  - 若 $\delta(q) = q_1 \vee q_2$，则 x 在 $\hat{T}$ 中有唯一的子节点 $x \cdot 0$，其中 $\hat{\rho}(x \cdot 0) = (q_1, y)$ 或者 $\hat{\rho}(x \cdot 0) = (q_2, y)$。

对于 $\hat{T}$ 中的任意一条无穷路径 $\sigma = x_0, x_1, \ldots$，令 $\mathbf{Inf}(\Omega(\sigma)) = \{n \in \mathbb{N} \mid$ 有无穷多个 $i \in \mathbb{N}$ 使得 $\Omega(\hat{\rho}_1(x_i)) = n\}$。其中 $\hat{\rho}_1$ 是 $\hat{\rho}$ 的第一投影函数，即：若 $\hat{\rho}(x) = (q, y)$，则 $\hat{\rho}_1(x) = q$。称 $\langle \hat{T}, \hat{\rho} \rangle$ 是一个**可接收运行**，当且仅当对 $\hat{T}$ 中的每条无穷路径 σ，都有 $\max(\mathbf{Inf}(\Omega(\sigma)))$ 为偶数。

称计算树 $\langle T, \rho \rangle$ 可被 $\mathcal{A}$ **接收**（同样记作 $\langle T, \rho \rangle \in \mathbf{L}(\mathcal{A})$），当且仅当 $\mathcal{A}$ 在 $\langle T, \rho \rangle$ 上存在某个可接收运行。 □

对每个良命名的句子 φ（假设 φ 已写为否定范式），都可以构造一个 APT $\mathcal{A}_\varphi = \langle Q_\varphi, \delta_\varphi, q_\varphi, \Omega_\varphi \rangle$，其中：

- $Q_\varphi = \{q_\psi \mid \psi$ 是 φ 的子公式$\}$（这样，初始状态 q_φ 自然属于 Q_φ）。
- δ_φ 定义如下：
  - $\delta_\varphi(q_{true}) = true$；$\delta_\varphi(q_{false}) = false$。
  - 对 φ 中的原子命题 p 而言，$\delta_\varphi(q_p) = p$；$\delta_\varphi(q_{\neg p}) = \neg p$。
  - 对 φ 中的（约束）变元 X 而言，$\delta_\varphi(q_X) = q_{\mathbf{D}_\varphi(X)}$。
  - $\delta_\varphi(q_{\varphi_1 \vee \varphi_2}) = q_{\varphi_1} \vee q_{\varphi_2}$；$\delta_\varphi(q_{\varphi_1 \wedge \varphi_2}) = q_{\varphi_1} \wedge q_{\varphi_2}$。
  - $\delta_\varphi(q_{\Box\psi}) = \Box q_\psi$；$\delta_\varphi(q_{\Diamond\psi}) = \Diamond q_\psi$。
  - $\delta_\varphi(q_{\mu X.\psi}) = q_\psi$；$\delta_\varphi(q_{\nu X.\psi}) = q_\psi$。
- 接收条件 Ω_φ 是任意一个满足如下约束的部分函数：

– Ω_φ 在状态 q_ψ 处有定义当且仅当 ψ 是 φ 中的（约束）变元；
– 若 X 是 φ 中的 μ-型约束变元，则 $\Omega_\varphi(q_X)$ 为奇数；若 X 是 φ 中的 ν-型约束变元，则 $\Omega_\varphi(q_X)$ 为偶数。
– 若 $Y \lhd_\varphi X$，则 $\Omega_\varphi(q_Y) < \Omega_\varphi(q_X)$。

定理 4.4 ([65]) 设 φ 是良命名的模态 μ－演算句子（设 φ 已经被写为否定范式），则对于任意计算树 $\langle T, \rho\rangle$ 有：$T, \rho \models \varphi$ 当且仅当 $\langle T, \rho\rangle \in \mathbf{L}(\mathcal{A}_\varphi)$。

注意：本文在构造自动机时，接收条件 Ω_φ 的定义与文 [65] 中稍有不同。由于 Wilke 的定义中需要用到公式“**交换深度**” 的概念，故而本文对其定义做了简化。但这并不影响定理 4.4 的证明— 二者唯一区别在于：按这种方法得到的 Ω_φ 的值域中具有较多的元素数目。

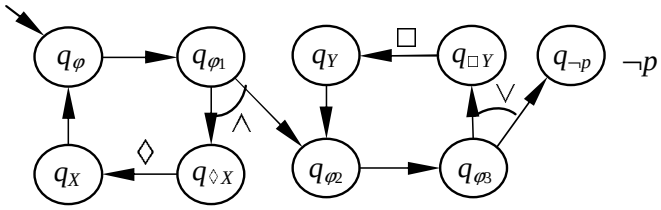


图 4.1 模态 μ-演算公式 APT 示例

例 4.2.9 设句子 $\varphi = \nu X.(\Diamond X \wedge \mu Y.(\neg p \vee \Box Y))$。则 $\mathcal{A}_\varphi$ 对应的 *APT* 如图 *4.1* 所示。其中，φ_1、φ_2、φ_3 分别为 $\Diamond X \wedge \mu Y.(\neg p \vee \Box Y)$、$\mu Y.(\neg p \vee \Box Y)$、$\neg p \vee \Box Y$。 □

4.2.2 模态 μ-演算的博弈系统

在文 [35] 中，Lange 和 Stirling 提出了一种基于博弈方法的时序逻辑公式可满足性测试方法，并在其基础上给出了新的关于 LTL 和 CTL 的公理系统。在本节，将该种方法推广至模态 μ-演算。由于模态 μ-演算的语义定义在分支结构（或者说，“计算树”）上，因此可以考虑采用博弈（Game）的方法对公式的可满足性进行测试。

对于给定的模态 μ-演算句子 φ（在本节，均认为其已被写为博弈范式），可以想像系统中有两个参与者（参与者 0 和参与者 1）在进行“比赛” — 前者试图证明 “φ 是可满足的”；而后者试图证明 “φ 是不可满足的”。每个参与者选取某条格规则将当前的公式集（称为当前的**格局**）进行重写，得到新的格局，直到当前格局序列满足某参与者的取胜条件。模态 μ－演算的博弈系统形式定义如下。

定义 4.2.12 (模态 μ-演算公式的博弈系统) 设 φ 是某个写为博弈范式的句子，则关于 φ 的**博弈系统** 是一个三元组 $\mathcal{G}_\varphi = \langle \mathcal{L}_\varphi, \mathcal{W}_\varphi, \Gamma_0\rangle$，其中：

- $\mathcal{L}_\varphi \subseteq 2^{\mathbf{Sub}(\varphi)}$，其中，$\mathbf{Sub}(\varphi)$ 为 φ 的子公式集合，每个 $\Gamma \in \mathcal{L}_\varphi$ 称为一个**格局**。特别的，仅包含文字以及形如 $\Box\psi$、$\Diamond\psi$ 公式的格局称为**模态格局**。
- $\mathcal{W}_\varphi \subseteq \mathcal{L}_\varphi \times \mathcal{L}_\varphi$，是 $\mathcal{G}_\varphi$ 的**格局迁移关系**。其中，$(\Gamma, \Gamma') \in \mathcal{W}_\varphi$ 当且仅当某个参与者（参与者 0 或参与者 1）能够采用某条格局迁移规则由 Γ 得到 Γ'。格局迁移规则稍后定义。
- $\Gamma_0 = \{\varphi\}$，为 $\mathcal{G}_\varphi$ 的**初始格局**。 □

（在第 3 章中，曾经用符号 $\mathcal{G}_\varphi$ 表示 ETL 公式 φ 的公式迁移图。在本章，用 $\mathcal{G}_\varphi$ 表示 μ-演算公式 φ 的博弈系统。）

定义 4.2.13 (格局迁移规则) 模态 μ-演算公式博弈系统中**格局迁移规则**（简称**规则**）的一般形式为:

$$\frac{\Gamma}{\Gamma'} \qquad \text{(name)}$$

其中，(name) 称为**规则名**，Γ' 称为**重写前格局**，Γ' 称为**重写后格局**。模态 μ－演算博弈系统中的迁移规则可以分为三类:

0-型规则：该类规则只能被参与者 0 使用;

1-型规则：该类规则只能被参与者 1 使用;

公共规则：该类规则既能被参与者 0 使用，也能被参与者 1 使用。 □

- 0-型格局迁移规则包括如下两条（以下，为简便起见，将 $\Gamma \cup \{\psi\}$ 简写为 Γ, ψ。同样，对于每个形如 Γ, φ 的重写前格局而言，要求$\varphi \notin \Gamma$）:

$$\frac{\Gamma, \psi_1 \vee \psi_2}{\Gamma, \psi_1} \qquad \text{(or-1)} \qquad\qquad \frac{\Gamma, \psi_1 \vee \psi_2}{\Gamma, \psi_2} \qquad \text{(or-2)}$$

- 1-型格局迁移规则仅有一条：

$$\frac{\{l_1, \ldots, l_k, \Diamond\varphi_1, \ldots, \Diamond\varphi_m, \Box\psi_1, \ldots, \Box\psi_n\}}{\{\varphi_j, \psi_1, \ldots, \psi_n\}} \qquad \text{(modal)}$$

其中，$l_1, \ldots, l_k \in AP \cup \overline{AP} \cup \{true, false\}$。上面给出的是 $m > 0$ 时的情形，其中 $j \in \{1, \ldots, m\}$; 特别的，当 $m = 0$ 时，重写后的格局为 $\{\psi_1, \ldots, \psi_n\}$。

- 公共格局迁移规则有如下四条。

$$\frac{\Gamma, \psi_1 \wedge \psi_2}{\Gamma, \psi_1, \psi_2} \qquad \text{(and)} \qquad\qquad \frac{\Gamma, X}{\Gamma, \mathbf{D}_\varphi(X)} \qquad \text{(fix)}$$

$$\frac{\Gamma, \mu X.\psi}{\Gamma, \psi} \quad \text{(mu-rmv)} \qquad \frac{\Gamma, \nu X.\psi}{\Gamma, \psi} \quad \text{(nu-rmv)}$$

例 4.2.10 设 $\varphi = \nu Y.(p \wedge \Diamond Y) \wedge \mu X.(\Box X \vee p)$，则 $\mathcal{G}_\varphi$ 如图 *4.2* 所示。 □

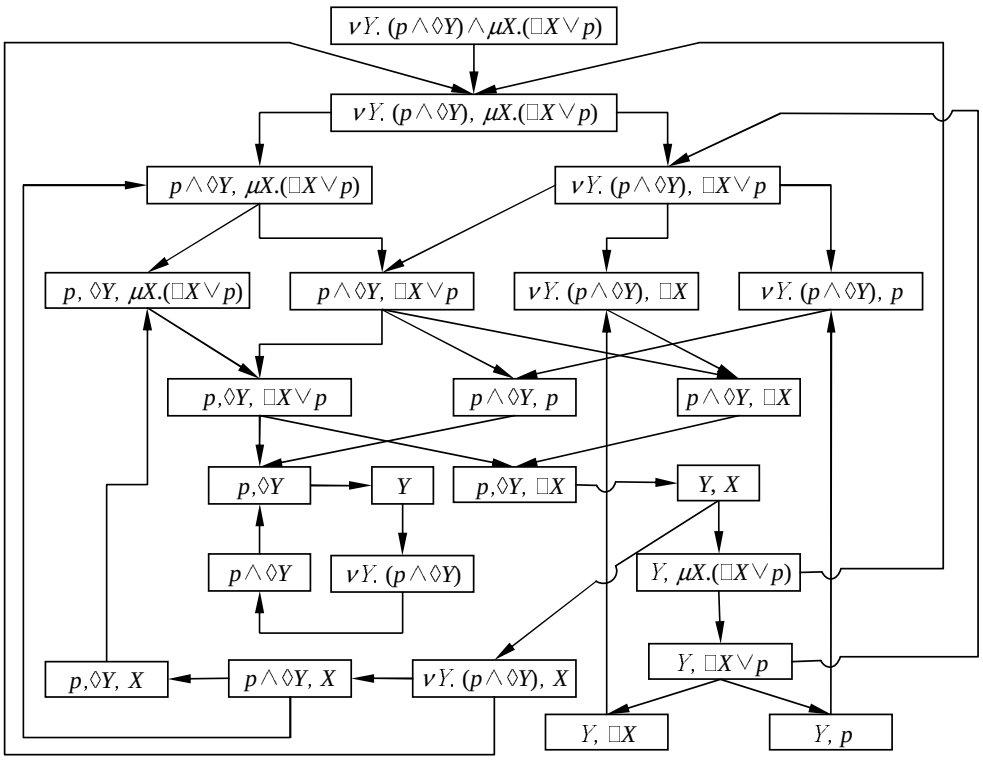


图 4.2 模态 μ-演算博弈系统示例

定义 4.2.14 (对决) 设 $\mathcal{G}_\varphi$ 是句子 φ 的博弈系统。则 $\mathcal{G}_\varphi$ 中的一个**对决**（*Match*）是一个格局序列 $\Gamma_0, \Gamma_1, \ldots$。其中，每个 $(\Gamma_i, \Gamma_{i+1}) \in \mathcal{W}_\varphi$。特别的，当 Γ_0 是初始格局 $\{\varphi\}$ 时，称该对决为一个**初始对决**。

称对决 $\Gamma_0, \Gamma_1, \ldots$ 是**终结的**，如果满足下列条件之一:

1. 该对决是有穷序列 $\Gamma_0, \Gamma_1, \ldots, \Gamma_m$，其中 Γ_m 中含有 $false$ 或者互补的文字。
2. 该对决是有穷序列 $\Gamma_0, \Gamma_1, \ldots, \Gamma_m$，其中 $\Gamma_m \subseteq AP \cup \overline{AP} \cup \{true\}$，且其中不含互补对。
3. 该对决为无穷序列。

此外，该对决的任意真前缀都不是终结的。 □

同 ETL 公式迁移图中的重写规则，μ-演算博弈系统的格局迁移规则中也有对应的消解公式、生成公式的概念

定义 4.2.15 (消解公式、生成公式) 在使用格局迁移规则由 Γ 获得 Γ' 时，Γ 中被选择重写的公式称为**消解公式**; Γ' 中由重写规则新得到的公式称为**生成公式**。 □

例 4.2.11 规则 $(\mathrm{or}-1)$ 中的消解公式是 $\psi_1 \vee \psi_2$，生成公式是 ψ_1。规则 (and) 中的消解公式是 $\psi_1 \wedge \psi_2$，生成公式有两个— ψ_1 和 ψ_2。规则 (modal) 中 $\Diamond\psi_j$ 和每个 $\Box\varphi_i$ 都是消解公式，它们对应的生成公式分别是 ψ_j 和 φ_i。 □

定义 4.2.16 (对决中的踪迹) 设 $\Gamma_0, \Gamma_1, \ldots$ 是 $\mathcal{G}_\varphi$ 中的某个对决。则称公式序列 $\psi_0, \psi_1, \ldots$ 是该对决中的一条**踪迹** 如果:

1. $\psi_i \in \Gamma_i$。
2. 若 ψ_i 是由 Γ_i 得到 Γ_{i+1} 所用规则时 Γ_i 中的消解公式，则 ψ_{i+1} 是对应的生成公式; 否则，$\psi_{i+1} = \psi_i$。 □

定义 4.2.17 (对决取胜条件) 设 $\Gamma_0, \Gamma_1, \ldots$ 是 $\mathcal{G}_\varphi$ 中一个终结的对决，若该次对决满足下列条件之一，则称参与者 0 是该次对决的**获胜者**:

1. 该对决是有穷序列 $\Gamma_0, \Gamma_1, \ldots, \Gamma_m$，其中 $\Gamma_m \subseteq AP \cup \overline{AP} \cup \{true\}$，且 Γ_m 中不含互补的文字对。
2. 该序列无穷，且每一条无穷踪迹中无穷多次出现约束变元中位于最外层是 φ 中的某个 ν-型约束变元。

若该次对决满足下列条件之一，则称参与者 1 是该次对决的**获胜者**:

1. 该对决是有穷序列 $\Gamma_0, \Gamma_1, \ldots, \Gamma_m$，其中 Γ_m 中包含 $false$ 或者互补的文字对。
2. 该序列无穷，且某条无穷踪迹中无穷多次出现的最外层约束变元是 φ 中的某个 μ-型约束变元。 □

引理 4.5 设 $\Gamma_0, \Gamma_1, \ldots$ 是 $\mathcal{G}_\varphi$ 中一个对决。若 $\Gamma_i = \Gamma_j$ ($j > i$)，则 Γ_i 和 Γ_j 之间一定有某个格局是模态格局。

证明. 同引理 3.2 的思路类似，仍对 φ 的每个子公式 ψ 赋一个非负整数 $\mathbf{Rk}(\psi)$，归纳定义如下：

- 若 $\psi \in AP \cup \overline{AP} \cup \{true, false\}$，则 $\mathbf{Rk}(\psi) = 0$。
- 若 ψ 形如 $\Box\psi'$ 或者 $\Diamond\psi'$ 时，$\mathbf{Rk}(\psi) = 0$。
- 若 ψ 是 φ 中的（约束）变元，则 $\mathbf{Rk}(\varphi) = \mathbf{Rk}(\mathbf{D}_\varphi(\psi)) + 1$。
- 若 $\psi = \psi_1 \vee \psi_2$ 或者 $\psi = \psi_1 \wedge \psi_2$，则 $\mathbf{Rk}(\psi) = \mathbf{Rk}(\psi_1) + \mathbf{Rk}(\psi_2) + 1$。
- 若 $\psi = \mu X.\psi'$ 或者 $\psi = \nu X.\psi'$，则 $\mathbf{Rk}(\psi) = \mathbf{Rk}(\psi') + 1$。

由于 φ 是受卫公式，容易证明 $\mathbf{Rk}$ 是良定义的（注意：对于非受卫公式，该函数可

能存在定义循环依赖)。同时，对每个 $i \leq k \leq j$，将 $\mathbf{Rk}$ 提升为

$$\mathbf{Rk}(\Gamma_k) \stackrel{\text{def}}{=} \sum_{\psi \in \Gamma_k} \mathbf{Rk}(\psi) \tag{4.1}$$

容易验证：若由 Γ_k 到 Γ_{k+1} 使用的格局迁移规则不是 (modal)，那么一定有 $\mathbf{Rk}(\Gamma_k) > \mathbf{Rk}(\Gamma_{k+1})$，于是 $\mathbf{Rk}(\Gamma_i) > \mathbf{Rk}(\Gamma_{i+1}) > \ldots > \mathbf{Rk}(\Gamma_j)$。而这与 $\mathbf{Rk}(\Gamma_i) = \mathbf{Rk}(\Gamma_j)$ 矛盾（因为 $\Gamma_i = \Gamma_j$)。而能够应用 (modal) 规则的格局只能是模态格局。因此，Γ_i 和 Γ_j 之间必然存在某个模态格局。 □

引理 4.6 *$\mathcal{G}_\varphi$ 中任意一个终结的对决都对应于一个唯一的获胜者。*

证明. 任取 $\mathcal{G}_\varphi$ 中的终结对决 $\Gamma_0, \Gamma_1, \ldots$。如果该对决为有穷序列，不妨设最后一个格局为 Γ_m，那么：或者 Γ 中含有互补的文字对，这种情况下参与者 1 是获胜者；或者 $\Gamma \subseteq AP \cup \overline{AP} \cup \{true\}$ 且不含互补文字对，这种情况下参与者 0 是获胜者。下面考虑 $\Gamma_0, \Gamma_1, \ldots$ 是无穷对决的情况。首先证明：“*该对决中的任意一条无穷踪迹中必然无穷多次出现约束变元，且这些无穷多次出现的约束变元中存在位于最外层者*”。

设 $\psi_0, \psi_1, \ldots$ 是该对决中的一条无穷踪迹。因为该对决中必然有某个格局出现无穷多次，由引理 4.5，必然有无穷多个模态格局。因而，该踪迹中有无穷多个公式是所在格局的消解公式— 因为如果 Γ_i 是模态节点，ψ_i 必然是形如 $\Box\psi'$ 或者 $\Diamond\psi'$ 的公式（若 ψ_i 是文字，则 ψ_{i+1} 将不存在)，这样 ψ_i 就是消解公式。观察所有的格局迁移规则，只有 (fix) 规则能够使消解公式长度增加。因而，该规则必然被无限次作用在该踪迹中的公式上，这意味着该踪迹中有无穷多次约束变元出现。令 $\mathcal{V}$ 是由在该踪迹中无穷多次出现的约束变元构成的集合，显然 $\mathcal{V} \neq \emptyset$。同时，必然存在某个 $i \in \mathbb{N}$，使得该踪迹中出现在 ψ_i 之后的变元都在 $\mathcal{V}$ 中（因为 $\mathcal{V}$ 是由该踪迹中无穷多次出现的变元构成的集合)。现在说明 $\mathcal{V}$ 中必然有某个变元位于最外层。设 $\psi_j = X$，$\psi_k = Y$，其中 $k > j > i$ 且 ψ_j 和 ψ_k 之间没有其他的公式是 φ 中的变元，则可以断言 $\psi_{i+1} = \mathbf{D}_\varphi(X)$。这时，还可以断言一定有下列两种情况之一成立：

1. 存在某个 $j < l < k$ 使得 $\varphi_l = \mathbf{D}_\varphi(Y)$。这说明 $\mathbf{D}_\varphi(Y)$ 一定是 $\mathbf{D}_\varphi(X)$ 的子公式，这种情况下有 $Y \lhd_\varphi X$ 成立。
2. $\mathbf{D}_\varphi(Y)$ 不在 ψ_i 和 ψ_j 之间出现。这说明 Y 在 $\mathbf{D}_\varphi(X)$ 中自由，因而一定有 $X \lhd_\varphi Y$ 成立。

因此，该踪迹中任何两个相邻出现的约束变元关于 $\lhd_\varphi$ 可比较。注意到 $\mathcal{V}$ 是有穷集，$\lhd_\varphi$ 是“传递的” 和“向上可比较的”（见引理 4.1)，因此 $\mathcal{V}$ 中必然存在某个位于最外层的约束变元。显然，它或者是 μ-型约束变元，或者是 ν-型约束变元。

因而，若该对决中的每条无穷踪迹中无穷多次出现的约束变元中最外层者都是 ν-型约束变元，则获胜者为参与者 0；否则，获胜者为参与者 1。 □

定义 4.2.18 (博弈策略，取胜博弈策略) 设 $\mathcal{G}_\varphi = \langle \mathcal{L}_\varphi, \mathcal{W}_\varphi, \{\varphi\}\rangle$ 是 φ 的博弈系统。参与者 i（$i \in \{0,1\}$）的一个**博弈策略**（简称**策略**）是一个函数 $f : \mathcal{L}_\varphi^* \to \mathcal{L}_\varphi$。称取胜策略 f 是**历史无关**（*History-Free*）的，是指对于任意的有穷博弈 $\Gamma_0, \ldots, \Gamma_m$ 和 $\Gamma'_0, \ldots, \Gamma'_n$ 而言，若 $\Gamma_m = \Gamma'_n$ 则 $f(\Gamma_0 \cdot \ldots \cdot \Gamma_m) = f(\Gamma'_0 \cdot \ldots \cdot \Gamma'_n)$。称参与者 i 在某次对决 $\Gamma_0, \Gamma_1, \ldots$ 中**遵循** 策略 f，如果对该序列中每个 k 而言，若 Γ_{k+1} 是由参与者 i 从 Γ_k 得到，则 $\Gamma_{k+1} = f(\Gamma_0 \cdot \ldots \cdot \Gamma_k)$。称 f 是参与者 i 的一个**取胜博弈策略**（简称**取胜策略**），是指：对于 $\mathcal{G}_\varphi$ 中的任何一个起始于 $\{\varphi\}$ 的终结对决，若参与者 i 在该对决中遵循 f，则参与者 i 是该对决的获胜者。 □

定理 4.7 $\mathcal{G}_\varphi$ 中最多有 1 个参与者有关于该博弈的取胜策略。

证明. 反设两个参与者 0 和 1 分别有取胜策略 f_0 和 f_1。则从初始格局 $\{\varphi\}$ 开始，二者分别遵循这两个策略构建某个对决，直至其终止。（注意：任何一个尚未终止的对决，一定能够继续进行下去。因此，它将来或者会满足有穷终止条件，或者成为无穷的格局序列）由引理 4.6，该终止对决具有唯一的获胜者。但由取胜策略的定义，两个参与者都是该对决的获胜者，从而产生了矛盾。 □

下面介绍由 Gurevich、Harrington、Emerson、Jutla 给出的博弈理论中的重要结论（见文 [107, 108]）。

定理 4.8 若博弈系统中的格局数目有穷，并且取胜条件互斥，则参与者 i 有关于该博弈的取胜策略，当且仅当该参与者具有关于该博弈历史无关的取胜策略。

“取胜条件互斥” 是指每个终结对决都对应于唯一的获胜者（这一点可由引理 4.6 保证）。事实上，上述定理适用于更加一般的博弈获胜条件以及参与者数目多于两个的情形。

接下来，将通过建立公式可满足性与该种博弈取胜者之间的关系。下面的两个定理会证明：“参与者 0 有关于 $\mathcal{G}_\varphi$ 的取胜策略当且仅当 φ 是可满足的”。

定理 4.9 若模态 μ-演算句子 φ 是可满足的，则参与者 0 有关于 $\mathcal{G}_\varphi$ 的取胜策略。

证明. 因为 φ 可满足，故存在计算树 $\langle T, \rho\rangle$ 使得 $T, \rho \models \varphi$。由定理 4.4 知 APT $\mathcal{A}_\varphi = \langle Q_\varphi, \delta_\varphi, q_\varphi, \Omega_\varphi\rangle$ 在 $\langle T, \rho\rangle$ 上必存在某个可接收运行 $\langle \widehat{T}, \hat{\rho}\rangle$。注意到 $\hat{\rho}$ 的值域为 $Q_\varphi \times T$，为方便起见，分别用 $\hat{\rho}_1$ 和 $\hat{\rho}_2$ 来表示 $\hat{\rho}$ 的两个投影。即：若 $\hat{\rho}(x) = (q_\psi, y)$，则 $\hat{\rho}_1(x) = q_\psi$，$\hat{\rho}_2(x) = y$。同时，对每个 $x \in \widehat{T}$，记 $||x||$ 为 x 在 $\widehat{T}$ 中的**模态深度**，它描述了从根节点 ϵ 到 x 父节点之间的路径上 $\hat{\rho}_1$ 的值形如 $q_{\Diamond\psi'}$ 或者 $q_{\Box\psi'}$ 的节点数目。其形式定义归纳如下给出：

- $||\epsilon|| = 0$。
- 设 $\hat{\rho}_1(x) = q_\psi$，若 ψ 形如 $\Diamond\psi'$ 或者 $\Box\psi'$，则 $||x{\cdot}c|| = ||x||{+}1$；否则，$||x{\cdot}c|| = ||x||$。

下面，将根据该可接收运行构造参与者 0 的取胜策略 f。在构造 f 的同时，将伴随证明下面的性质：*"在 f 的控制下，对 $\mathcal{G}_\varphi$ 中任何一个起始于格局 $\{\varphi\}$ 的尚未终结的对决 $\Gamma_0, \ldots, \Gamma_m$，以及其中的每个格局 Γ_i（$0 \leq i \leq m$）而言，存在 $\widehat{T}$ 中的节点集合 $\{x_1, \ldots, x_n\}$（称为 Γ_i 的**对应节点集**），满足:*

1. $||x_1|| = \ldots = ||x_n||$;
2. $\hat{\rho}_2(x_1) = \ldots = \hat{\rho}_2(x_n)$;
3. *对每个 $\psi \in \Gamma_i$，存在某个 $1 \leq j \leq n$，使得 $\hat{\rho}_1(x_j) = q_\psi$。"*

首先，该对决必然起始于初始格局 $\{\varphi\}$，这时，可以选取 $\{\epsilon\}$ 为该格局的对应节点集。这是由于 $\{\epsilon\}$ 为单元素集合，且 $\hat{\rho}_1(\epsilon) = q_\varphi$，因而满足上述要求。

其次，归纳假设当前的（未终结）对决为 $\Gamma_0, \ldots, \Gamma_m$，其中 $\Gamma_0 = \{\varphi\}$。现在证明，对 Γ_m 施加任意的 1-型规则或者公共规则后所得到的新格局 Γ_{m+1} 的对应节点集仍然存在。

- 当使用的格局迁移规则是 (and) 时，不妨设 Γ_m 的对应节点集为 $\{x_1, \ldots, x_n\}$，消解公式 $\psi_m = \psi'_m \wedge \psi''_m$，并且 $\hat{\rho}_1(x_j) = q_{\psi_m}$。由 $\mathcal{A}_\varphi$ 的构造知 $\delta_\varphi(q_{\psi_m}) = q_{\psi'_m} \wedge q_{\psi''_m}$，再由定义 4.2.11 知，$x_j$ 在 $\widehat{T}$ 中必然有两个子节点 x'_j 和 x''_j，使得 $\hat{\rho}_1(x'_j) = q_{\psi'_m}$，$\hat{\rho}_2(x''_j) = q_{\psi''_m}$，并且 $\hat{\rho}_2(x'_j) = \hat{\rho}_2(x''_j) = \hat{\rho}_2(x_j)$，同时依定义有 $||x'_j|| = ||x''_j|| = ||x_j||$。于是，可令 $\{x_1, \ldots, x_{j-1}, x'_j, x''_j, x_{j+1}, \ldots, x_n\}$ 作为 Γ_{m+1} 的对应节点集。
- 当使用的格局迁移规则是 (mu-rmv) 时，不妨设 Γ_m 的对应节点集为 $\{x_1, \ldots, x_n\}$，消解公式 $\psi_m = \mu X.\psi'_m$，并且 $\hat{\rho}_1(x_j) = q_{\psi_m}$。由 $\mathcal{A}_\varphi$ 的构造知 $\delta_\varphi(q_{\psi_m}) = q_{\psi'_m}$。再由定义 4.2.11 知，$x_j$ 在 $\widehat{T}$ 中必然存在子节点 x'_j，使得 $\hat{\rho}_1(x'_j) = q_{\psi'_m}$，并且 $\hat{\rho}_2(x'_j) = \hat{\rho}_2(x_j)$，同时按照定义有 $||x'_j|| = ||x_j||$。于是，可令 $\{x_1, \ldots, x_{j-1}, x'_j, x_{j+1}, \ldots, x_n\}$ 作为 Γ_{m+1} 的对应节点集。类似的，也可以证明使用 (nu-rmv) 规则时的情形。
- 当使用的格局迁移规则是 (fix) 时，不妨设 Γ_m 的对应节点集为 $\{x_1, \ldots, x_n\}$，消解公式 $\psi_m = X$（其中 X 是 φ 中的约束变元），并且 $\hat{\rho}_1(x_j) = q_{\psi_m}$。由 $\mathcal{A}_\varphi$ 的构造知 $\delta_\varphi(q_{\psi_m}) = q_{\mathbf{D}_\varphi(X)}$。再由定义 4.2.11 知，$x_j$ 在 $\widehat{T}$ 中必然存在子节点 x'_j 使得 $\hat{\rho}_1(x'_j) = q_{\mathbf{D}_\varphi(X)}$，并且 $\hat{\rho}_2(x'_j) = \hat{\rho}_2(x_j)$，同时按照定义 $||x'_j|| = ||x_j||$。于是，可令 $\{x_1, \ldots, x_{j-1}, x'_j, x_{j+1}, \ldots, x_n\}$ 作为 Γ_{m+1} 的对应节点集。
- 当使用的规则是 (modal) 时，不妨设 $\Gamma_m = \{l_1, \ldots, l_k, \Box\varphi_1, \ldots, \Box\varphi_n, \Diamond\psi_1, \ldots,$

$\Diamond\psi_t\}$，并且 $\Gamma_{m+1}=\{\varphi_1,\ldots,\varphi_n,\psi_j\}$。同时，设在 Γ_m 的对应节点集中 $\hat{\rho}_1(x_1)=q_{\Box\varphi_1}$，…，$\hat{\rho}_1(x_n)=q_{\Box\varphi_n}$，$\hat{\rho}_1(y_1)=q_{\Diamond\psi_1}$，…，$\hat{\rho}_1(y_t)=q_{\Diamond\psi_t}$。以及 $\hat{\rho}_2(x_1)=\ldots=\hat{\rho}_2(x_n)=\hat{\rho}_2(y_1)=\ldots=\hat{\rho}_2(y_t)=z$（其中，$z\in T$）。由 $\mathcal{A}_\varphi$ 的构造知，$\delta_\varphi(q_{\Diamond\psi_j})=\Diamond q_{\psi_j}$，$\delta_\varphi(q_{\Box\varphi_i})=\Box q_{\varphi_i}$（$i\in\{1,\ldots,n\}$）。同时，由定义 4.2.11 知，存在 z 的子节点 $z'\in T$，以及 y_j 在 $\widehat{T}$ 中的子节点 y'_j 使得 $\hat{\rho}(y'_j)=(q_{\psi_j},z')$。由于 z' 是 z 在 T 中的子节点，所以，x_i 在 $\widehat{T}$ 中存在子节点 x'_i，使得 $\hat{\rho}(x_i)=(q_{\varphi_i},z')$。注意到 $||x'_i||=||x_i||+1$，$||y'_j||=||y_j||+1$，于是 $||x'_1||=\ldots=||x'_n||=||y'_j||$。因此，可以取 $\{x'_1,\ldots,x'_n,y'_j\}$ 作为 Γ_{m+1} 的对应节点集。

接下来证明：如果当前格局 Γ_m 可以施加 0-型规则，那么参与者 0 有策略保证得到的新格局 Γ_{m+1} 也存在对应节点集。

- 不妨设 Γ_m 的对应节点集为 $\{x_1,\ldots,x_n\}$，消解公式 $\psi_m=\psi'_m\vee\psi''_m$，并且 $\hat{\rho}_2(x_j)=q_{\psi_m}$。由 $\mathcal{A}_\varphi$ 的构造知，$\delta_\varphi(q_{\psi_m})=q_{\psi'_m}\vee q_{\psi''_m}$。由定义 4.2.11，必然存在 x_j 的某个子节点 x'_j 使得 $\hat{\rho}_2(x'_j)=\hat{\rho}_2(x_j)$，$\hat{\rho}_1(x'_j)\in\{q_{\psi'_m},q_{\psi''_m}\}$。由定义，$||x_j||=||x'_j||$ 成立。如果 $\hat{\rho}_1(x'_j)=q_{\psi'_m}$，则参与者使用 (or-1) 规则，这时，得到的新格局 Γ_{i+1} 为 $\Gamma_i\backslash\{\psi_m\}\cup\{\psi'_m\}$；如果 $\hat{\rho}_1(x'_j)=q_{\psi''_m}$，则参与者使用 (or-2) 规则，这时，得到的新格局 Γ_{i+1} 为 $\Gamma_i\backslash\{\psi_m\}\cup\{\psi''_m\}$。在这两种情况下，均可令 $\{x_1,\ldots,x_{j-1},x'_j,x_{j+1},\ldots,x_n\}$ 为 Γ_{m+1} 的对应节点集。

于是，参与者 0 的取胜策略 f 可按如下方式确定：首先，将所有公共规则和 0-型规则按照任意特定顺序规定优先级。对任意未终结（且在参与者上述策略控制下）的对决 $\Gamma_0,\ldots,\Gamma_m$，若 Γ_m 可施加 0-型或公共规则，则定义 $f(\Gamma_0,\ldots,\Gamma_m)$ 为可对 Γ_m 施加的优先级最高的规则后所获得的格局。并且，当采用 0-型规则时，新格局按照所讨论的参与者 0 的策略获得。在其他情况下（包括：1. $\Gamma_0,\ldots,\Gamma_m$ 不是合法的对决序列或者不可能在上述策略下产生；2. $\Gamma_0,\ldots,\Gamma_m$ 中已经包括一个终结对决真前缀；3. 或者 Γ_m 上无法施加 0-型规则和公共规则），则可令 $f(\Gamma_0,\ldots,\Gamma_m)$ 为 $\mathcal{G}_\varphi$ 中的任一格局。

再证明，当参与者 0 遵循 f 时，任一终结对决的获胜者不可能是参与者 1。

1. 若该终结对决是有穷序列 $\Gamma_0,\ldots,\Gamma_m$，由前面的讨论知，Γ_m 必存在某个对应节点集 $\{x_1,\ldots,x_n\}$。假设 Γ_m 中含有一对互补的文字 p 和 $\neg p$，那么不妨设 $\hat{\rho}_1(x_i)=q_p$，$\hat{\rho}_1(x_j)=q_{\neg p}$ 并且 $\hat{\rho}_2(x_i)=\hat{\rho}_2(x_j)=y$。这时，由 $\mathcal{A}_\varphi$ 的构造知，$\delta_\varphi(q_p)=p$；$\delta_\varphi(q_{\neg p})=\neg p$。由定义 4.2.11 知 $p\in\rho(y)$ 且 $p\notin\rho(y)$，这不可能。同样，也可以证明 Γ_m 中不含 $false$。
2. 若该终结对决是无穷序列 $\Gamma_0,\Gamma_1,\ldots$，则由 f 的构造知，对于其中的任意一条无

穷踪迹 $\psi_0, \psi_1, \ldots$，$\widehat{T}$ 中必然存在某个无穷路径 $x_0, x_1, \ldots$ 以及数列 $i_0 < i_1 < \ldots$ 其中 $i_0 = 0$，$x_0 = \epsilon$，且对于任意的 $i_j \leq l < i_{j+1}$ 都有 $\hat{\rho}_1(x_j) = q_{\psi_l}$。由于 $\langle \widehat{T}, \hat{\rho} \rangle$ 是 $\mathcal{A}_\varphi$ 的一个可接收运行，由 Ω_φ 的定义知：踪迹 $\psi_0, \psi_1, \ldots$ 无穷多次出现的约束变元中位于最外层者必定是某个 ν-型约束变元。

由取胜条件的确定性（见引理 4.6），参与者 0 必然是该终结对决的获胜者。这样，f 就是参与者 0 的一个取胜策略。 □

当证明“若参与者 0 有关于 $\mathcal{G}_\varphi$ 的取胜策略，则 φ 可满足”时，希望证明其逆否命题。这里，需要引入如下的概念以及引理。

定义 4.2.19 (博弈树) 关于 $\mathcal{G}_\varphi$ 的一棵**博弈树**，是一个特殊的标记树 $\langle \bar{T}, \bar{\rho} \rangle$，其中每个节点 x 上的标注 $\bar{\rho}(x)$ 是 $\mathcal{G}_\varphi$ 中的某个有穷对决。同时，满足如下约束：

- $\bar{\rho}(\epsilon)$ 为某个初始对决。
- 对每个 $x \in \bar{T}$ 而言，$\bar{\rho}(x)$ 中最多只含有一个模态格局。如果含有，该模态格局只能是 $\bar{\rho}(x)$ 中最后一个格局。
- x 是 $\bar{T}$ 中的叶节点当且仅当 $\bar{\rho}$ 是某个终结对决。
- 若 x 是非叶节点，则 $\bar{\rho}(x)$ 的最后一个格局是模态格局，并且 x 和它的子节点之间满足“**拼接条件**”。即：假设 $\bar{\rho}(x)$ 的最后一个格局为 $\{l_1, \ldots, l_k, \Box\varphi_1, \ldots, \Box\varphi_n, \Diamond\psi_1, \ldots, \Diamond\psi_m\}$，则：当 $m > 0$ 时，x 有 m 个子节点，且对于每个 $1 \leq j \leq m$，存在 x 的某个子节点 $x \cdot c$，使得 $\bar{\rho}(x \cdot c)$ 中的第一个格局是 $\{\varphi_1, \ldots, \varphi_n, \psi_j\}$；当 $m = 0$ 时，x 有一个子节点 $x \cdot 0$，且 $\bar{\rho}(x \cdot 0)$ 中的第一个格局是 $\{\varphi_1, \ldots, \varphi_n\}$。

对于 $\bar{T}$ 中的任意一条路径 $\sigma = x_0, x_1, \ldots$ 而言，$\bar{\rho}(x_0) \cdot \bar{\rho}(x_1) \cdot \ldots$ 必然是 $\mathcal{G}$ 中的一个对决。该对决称为 σ 在 $\bar{\rho}$ 下的**导出对决**，记作 $\bar{\rho}(\sigma)$。 □

定理 4.10 若句子 φ 不可满足，则对于 $\mathcal{G}_\varphi$ 中的任意一棵博弈树 $\langle \bar{T}, \bar{\rho} \rangle$ 而言，其中必然存在某个起始于根节点的极大路径 σ，使得 $\bar{\rho}(\sigma)$ 的获胜者是参与者 1。

证明. 用反证法。反设存在某个关于 $\mathcal{G}_\varphi$ 的博弈树 $\langle \bar{T}, \bar{\rho} \rangle$ 满足：

- 对 $\bar{T}$ 中的每个叶节点 x 而言，$\hat{\rho}(x)$ 的最后一个格局是 $AP \cup \overline{AP} \cup \{true\}$ 的子集，且其中不含互补文字对。
- 对 $\bar{T}$ 中的每个起始于根节点的无穷路径 σ 的导出对决 $\bar{\rho}(\sigma)$ 中的每条无穷踪迹 τ 而言，无穷多次出现在 τ 中的最外层约束变元是 ν-型约束变元。

由引理 4.5，易知上述反设恰好是结论的否定。

现在，构造一棵计算树 $\langle T, \rho \rangle$，其中：

- $\bar{T} \subseteq T$。
- 对于每个 $x \in \bar{T}$，不妨设 Γ_x 是 $\bar{\rho}(x)$ 中的最后一个格局，则 $\rho(x)$ 为任意某个满

足约束 $(\Gamma_x \cap AP) \subseteq \rho(x)$ 且 $\{p \in AP \mid \neg p \in \Gamma_x\} \cap \rho(x) = \emptyset$ 的原子命题集合。

- 对每个 $x \in T \backslash \bar{T}$，一定存在某个祖先节点 x'，并且 x' 是 $\bar{T}$ 中的叶节点。
- 对每个 $x \in T \backslash \bar{T}$，令 $\rho(x)$ 是 AP 的任意某个子集即可。

事实上，T 是通过向 $\bar{T}$ 的叶节点添加无穷子树获得的。之所以做这样的修改，是因为要求每个计算树中不存在叶节点（见 28 页约定）。由前面的假设，当 $x \in \bar{T}$ 时，无论 x 是否为叶节点，Γ_x 中一定不含互补的文字，因此满足上述约束的 $\rho(x)$ 一定存在。

下面证明：$T, \rho \models \varphi$。由定理 4.4，只需构造 $\mathcal{A}_\varphi$ 在 $\langle T, \rho \rangle$ 上的一个可接收运行 $\langle T', \rho' \rangle$ 即可。

首先，令 $\rho'(\epsilon) = (q_\varphi, \epsilon)$。在此过程中，还将伴随构造一系列函数 $g_{x,i}$，其中 $x \in \bar{T}$，$i \in \mathbb{N}$。该函数将 $\bar{\rho}(x)$ 中的第 i 个格局中的每个公式 ψ 映射为 T' 中的某个节点，并且满足 $\rho'(g_{x,i}(\psi)) = (q_\psi, x)$。首先，令 $g_{\epsilon,0} = \epsilon$，并且假设函数 $g_{x,i}$ 已构造完毕，同时设 $\bar{\rho}(x)$ 中的第 i 个格局为 $\Gamma_{x,i}$。

当 $\Gamma_{x,i}$ 不是 $\bar{\rho}(x)$ 中最后一个格局时，$\Gamma_{x,i}$ 不是模态格局。并且，第 $i+1$ 个格局 Γ_{i+1} 必然也在其中。于是：

- 若 $\Gamma_{x,i+1}$ 由 $\Gamma_{x,i}$ 经规则 (and) 规则获得，则不妨设 $\Gamma_{x,i}$ 中的消解公式为 $\psi_1 \wedge \psi_2$，且 $g_{x,i}(\psi_1 \wedge \psi_2) = y$。易知，$\psi_1$、$\psi_2$ 是 $\Gamma_{x,i+1}$ 中的生成公式，并且由归纳知 $\rho'(y) = (q_{\psi_1 \wedge \psi_2}, x)$。于是，在 T' 中为 y 生成两个子节点 $y \cdot 0$ 和 $y \cdot 1$，并令 $\rho'(y{\cdot}0) = (q_{\psi_1}, x)$，$\rho'(y{\cdot}1) = (q_{\psi_2,x})$。同时，令 $g_{x,i+1}(\psi_1) = y{\cdot}0$，$g_{x,i+1}(\psi_2) = y{\cdot}1$。
- 若 $\Gamma_{x,i+1}$ 由 $\Gamma_{x,i}$ 经规则 (or-1)（resp.(or-2)）得到，则不妨设 $\Gamma_{x,i}$ 中的消解公式为 $\psi_1 \vee \psi_2$，且 $g_{x,i}(\psi_1 \vee \psi_2) = y$。易知，$\psi_1$（resp. ψ_2）是 $\Gamma_{x,i+1}$ 中的生成公式，并且由归纳知 $\rho'(y) = (q_{\psi_1 \wedge \psi_2}, x)$。于是，在 T' 中为 y 生成一个子节点 $y{\cdot}0$，并令 $\rho'(y \cdot 0) = (q_{\psi_1}, x)$（resp. $\rho'(y \cdot 0) = (q_{\psi_2}, x)$）。同时，令 $g_{x,i+1}(\psi_1) = y \cdot 0$（resp. $g_{x,i+1}(\psi_2) = y \cdot 1$）。
- 若 $\Gamma_{x,i+1}$ 由 $\Gamma_{x,i}$ 经规则 (fix) 规则获得，则不妨设 $\Gamma_{x,i}$ 中的消解公式为 Y，且 $g_{x,i}(Y) = y$。易知，$\mathbf{D}_\varphi(Y)$ 是 $\Gamma_{x,i+1}$ 中的生成公式，并且由归纳知 $\rho'(y) = (q_Y, x)$。于是，在 T' 中为 y 生成一个子节点 $y{\cdot}0$，并令 $\rho'(y \cdot 0) = (q_{\mathbf{D}_\varphi(Y)}, x)$。同时，令 $g_{x,i+1}(\mathbf{D}_\varphi(Y)) = y \cdot 0$。
- 若 $\Gamma_{x,i+1}$ 由 $\Gamma_{x,i}$ 经规则 (mu-rmv)（resp. (nu-rmv)）规则获得，则不妨设 $\Gamma_{x,i}$ 中的消解公式为 $\mu X.\psi'$（resp. $\nu X.\psi'$）且 $g_{x,i}(\mu X.\psi') = y$（resp. $g_{x,i}(\nu X.\psi') = y$）。易知，ψ' 是 $\Gamma_{x,i+1}$ 中的生成公式，并且由归纳知 $\rho'(y) = (q_{\mu X.\psi'}, x)$（resp. $\rho'(y) = (q_{\nu X.\psi'}, x)$）。于是，在 T' 中为 y 生成一个子节点 $y{\cdot}0$，并令 $\rho'(y \cdot 0) =$

$(q_{\psi'}, x)$。同时，令 $g_{x,i+1}(\psi') = y \cdot 0$。

同时，在上述情况中，对于 $\Gamma_{x,i+1}$ 中的每个非生成公式 ϕ，均令 $g_{x,i+1}(\phi) = g_{x,i}(\phi)$。

当 $\Gamma_{x,i}$ 是 $\bar{\rho}(x)$ 中最后一个格局时，分两种情况讨论。

- x 是 $\bar{T}$ 中的叶节点。由假设，$\Gamma_{x,i} \subseteq AP \cup \overline{AP} \cup \{true\}$。于是对于每个原子命题 $p \in AP$ 而言，若 $p \in \Gamma_{x,i}$，(resp. $\neg p \in \Gamma_{x,i}$) 不妨设 $g_{x,i}(p) = y$ (resp. $g_{x,i}(\neg p) = y$)，于是 $\rho'(y) = (q_p, x)$ ($\rho'(y) = (q_{\neg p}, x)$)。由于 x 是 $\bar{T}$ 中的叶节点，由 ρ 的构造有 $p \in \rho(x)$ ($p \notin \rho(x)$)。
- x 不是 $\bar{T}$ 中的叶节点，则 $\Gamma_{x,i}$ 是模态格局。不妨设 $\Gamma_{x,i} = \{l_1, \ldots, l_k, \Box\varphi_1, \ldots, \Box\varphi_n, \Diamond\psi_1, \ldots, \Diamond\psi_m\}$，且 x 在 $\bar{T}$ 中的 m 个子节点为 x_1，...，x_m。同时，对每个 $1 \leq j \leq m$，设 $\bar{\rho}(x_j)$ 中的第一个格局（即 $\Gamma_{x_j,0}$）为 $\{\varphi_1, \ldots, \varphi_n, \psi_j\}$。再假设 $g_{x,i}(\Box\varphi_l) = y_l$，$g_{x,i}(\Diamond\psi_j) = z_j$。由归纳得 $\rho'(y_l) = (q_{\Box\varphi_l}, x)$，$\rho'(z_j) = (q_{\Diamond\psi_j}, x)$。于是，为每个 y_l 添加 m 个子节点 $y_l \cdot 0$，...，$y_l \cdot (m-1)$，且令 $\rho'(y_l \cdot (j-1)) = (q_{\varphi_l}, x_j)$；为每个 z_j 添加一个子节点 $z_j \cdot 0$，且令 $\rho'(z_j \cdot 0) = (q_{\psi_j}, x_j)$。（以上，$l = 1, \ldots, n$，$j = 1, \ldots, m$）最后，对每个 $1 \leq j \leq m$，令 $g_{x_j,0}(\varphi_l) = y_l \cdot (j-1)$，$g_{x_j,0}(\psi_j) = z_j \cdot 0$。这样，对每个 x_j 可以递归调用此构造过程。（以上，讨论的是 $m > 0$ 时的过程。事实上，当 $m = 0$ 时，由于 x 在仅有一个子节点 $x \cdot 0$，且 $\Gamma_{x \cdot 0,0} = \{\varphi_1, \ldots, \varphi_n\}$。这时，只要为每个 y_1 添加一个子节点 $y_1 \cdot 0$，令 $\rho'(y_l \cdot 0) = x \cdot 0$，再令 $g_{x \cdot 0,0}(\varphi_l) = y_l \cdot 0$ 即可。）

在上述构造过程中，T 中的每个节点 x 都会生成 T' 中的若干节点（虽然实际构造过程中只用到了 $\bar{T}$ 中的节点）。由定义 4.2.11，容易检验 $\langle T', \rho' \rangle$ 是 $\langle T, \rho \rangle$ 在 $\mathcal{A}_\varphi$ 上的一个运行— 对于 T' 中的每个节点 y，不妨设 $\rho'(y) = (q_\psi, x)$：当 ψ 不是文字时，直接可以验证其满足定义 4.2.11 中的运行条件。当 ψ 是文字时，不妨设 $g_{x,i} = y$，那么 ψ 一定会出现在 $\bar{\rho}(x)$ 的最后一个格局中。由 ρ 的定义可知：若 ψ 是正文字，则 $\psi \in \rho(x)$，若 ψ 是负文字，则 $\psi \notin \rho(x)$。

此外，由构造过程知：对于 T' 中的每条无穷路径 σ'，必然存在 $\bar{T}$ 中的某条路径 $\bar{\rho}$ 的导出对决中的踪迹 $\bar{\tau}$，使得对 φ 中的每个约束变元 X 而言，X 在 $\bar{\tau}$ 中出现无穷多次当且仅当 $\{q_X\} \times T$ 中的元素在 σ' 中无穷多次出现。由假设，在 $\bar{\tau}$ 中无穷多次出现的约束变元中位于最外层者为 ν-型约束变元。于是，$\mathbf{Inf}(\Omega_\varphi(\sigma'))$ 必为偶数。从而 $\langle T', \rho' \rangle$ 是可接收的运行。由定理 4.4 有 $T, \rho \models \varphi$，这与 φ 不可满足矛盾。于是，假设不成立，从而定理中结论得证。 □

定理 4.11 *若句子 φ 是不可满足的，则参与者 0 不可能有关于 $\mathcal{G}_\varphi$ 的取胜策略。*

证明. 用反证法。假设参与者 0 存在某个关于 $\mathcal{G}_\varphi$ 的取胜策略 f，则通过如下方式

构建一棵关于 $\mathcal{G}_\varphi$ 的博弈树 $\langle \bar{T}, \bar{\rho} \rangle$。

首先，建立根节点 ϵ，并令 $\Gamma_{\epsilon,0} = \{\varphi\}$。其次，对于每个已经构建的节点 x，按照如下方式确定 $\bar{\rho}(x)$。设 $\bar{\sigma}$ 是从根节点 ϵ 到 x 父节点的路径，则对每个 i，令 $\Gamma_{x,i} = f(\bar{\rho}(\sigma) \cdot (\Gamma_{x,0}, \ldots, \Gamma_{x,i-1}))$，直至出现以下两种情况之一：

1. $\bar{\rho}(\sigma) \cdot (\Gamma_{x,0}, \ldots, \Gamma_{x,i})$ 是一个终结对决。这时，令 $\bar{\rho}(x) = \Gamma_{x,0}, \ldots, \Gamma_{x_i}$。同时，$x$ 成为 $\bar{T}$ 中的叶节点。
2. $\Gamma_{x,i}$ 是模态格局 $\{l_1, \ldots, l_k, \Box\varphi_1, \ldots, \Box\varphi_n, \Diamond\psi_1, \ldots, \Diamond\psi_m\}$。这时，令 $\bar{\rho}(x) = \Gamma_{x,0}, \ldots, \Gamma_{x_i}$。若 $m > 0$，则为 x 添加 m 个子节点 $x \cdot 0$，…，$x \cdot (m-1)$，并对每个 $0 \leq j < m$，令 $\Gamma_{x \cdot j,0} = \{\varphi_1, \ldots, \varphi_n, \psi_{j+1}\}$。若 $m = 0$，则为 x 添加一个子节点 $x \cdot 0$，并令 $\Gamma_{x \cdot 0,0} = \{\varphi_1, \ldots, \varphi_n\}$。继而，对 x 的每个子节点递归调用此过程构造。

由定理 4.10 知，$\langle \bar{T}, \bar{\rho} \rangle$ 中必然存在一个起始于根节点的极大路径 $\bar{\sigma}$，使得 $\bar{\sigma}$ 的导出对决 $\bar{\rho}(\bar{\sigma}) = \Gamma_0, \Gamma_1, \ldots$（其中 $\Gamma_0 = \Gamma_{\epsilon,0} = \{\varphi\}$）是一个以参与者 1 为获胜者的终结对决。由构造知，对于任意一个 i，只要 Γ_i 不是模态格局或终结格局，则必然有 $\Gamma_{i+1} = f(\Gamma_0 \cdot \ldots \cdot \Gamma_i)$（由于 (modal) 是 1-型规则，故参与者 0 无法对模态格局施加格局迁移规则），故而该对决是遵循 f 的。这与 f 是参与者 0 的取胜策略相矛盾。于是，参与者 0 关于 $\mathcal{G}_\varphi$ 的“取胜策略” 事实上并不存在。 □

定理 4.12 *若模态 μ-演算句子 φ 是不可满足的，则参与者 1 有关于 $\mathcal{G}_\varphi$ 的取胜策略。*

证明. 这里，不加证明的引用 Martin 关于时序逻辑博弈理论中的一个一般性的结论 [73]: 若博弈 $\mathcal{G}_\varphi$（事实上，这里的取胜条件可以更一般）

1. 是良划分的；
2. 任意终结对决的获胜者是确定的

则 $\mathcal{G}_\varphi$ 中有且仅有一个参与者有取胜策略。条件 2 可由引理 4.6 保证。但本节给出的博弈系统中并不是良划分的（所谓**良划分**，是指对于任意一个格局，仅有一个参与者能够对其施加格局迁移规则）。为此，可以稍微调整一下博弈系统的定义— 将 (and)、(fix)、(mu-rmv) 以及 (nu-rmv) 这四条公共规则重设为 0-型规则。这样，参与者 0 和参与者 1 能够施加迁移规则的格局分别是非模态格局和模态格局。于是，每个博弈系统均满足良划分约束。在这种设置下，容易检验定理 4.10 及定理 4.11 的证明仍然有效。于是，由 Martin 定理，当句子 φ 不可满足时参与者 1 有关于 $\mathcal{G}_\varphi$ 的取胜策略 f。显然，f 也是 (and)、(fix)、(mu-rmv) 以及 (nu-rmv) 为公共规则时参与者 1 的取胜策略。 □

定理 4.9 和定理 4.12 建立了模态 μ-演算（句子）公式可满足性与其对应博弈的取胜者之间的联系。于是，可以得到如下的推论。

推论 4.13 (博弈的确定性) *参与者 0（resp. 参与者 1）有关于 $\mathcal{G}_\varphi$ 的取胜策略当且仅当 φ 是可满足（resp. 不可满足）的。*

推论 4.14 (博弈的求解复杂性) *模态 μ-演算博弈的求解（即: 确定具有取胜策略的参与者）问题是 EXPTIME-complete的。*

证明. 由推论 4.13 以及模态 μ-演算公式的可满足性是 EXPTIME-complete 的[89, 90]可立即得到。（注：在模态 μ-演算中，良命名受卫句子公式的可满足性问题也是 EXPTIME-complete 的。） □

4.2.3 模态 μ-演算公理系统：完备性的证明

本节将使用在 4.2.2 节中的博弈理论来证明 Kozen[55]的模态 μ-演算公理系统的完备性。这个公理系统的完备性最先由 Walukiewicz 证明[56, 34]，因此，该公理系统也称为 Kozen-Walukiewicz 系统。

Kozen 的模态 μ-演算系统实际上是“多模态系统”。即：模态算子 $\Box$ 和 $\Diamond$ 伴随“动作词”（Action）一起出现。这样，每个模态算子都形如 $[a]$ 或者 $\langle a\rangle$，其中 $a \in ACT$。而 ACT 是系统取定动作词集合。在迁移系统中，动作词标注在“边”上。这样，在多模态迁移系统中，迁移关系 Δ 就成了 $S \times ACT \times S$ 的子集。在多模态 μ-演算中，如果 $(s_1, a, s_2) \in \Delta$，就称 s_2 是 s_1 的一个 a-后继。于是，$[a]\varphi$ 和 $\langle a\rangle\varphi$ 的“语义” 可以非形式的描述如下：

- s “满足” $[a]\varphi$ 当且仅当 s 的每个 a-后继都“满足” φ；
- s “满足” $\langle a\rangle\varphi$ 当且仅当 s 的某个 a-后继能“满足” φ。

多模态 μ-演算能够更加精细的区分系统的迁移动作（从某种意义上，它保留了动态逻辑的特征）。但对于模态 μ-演算的公理化问题而言，动作词不是研究的重点— 单模态 μ-演算的公理化方法可以平凡的扩展至多模态系统中。因此，在本节会忽略动作词：将 Kozen 系统中的每个公式 $[a]\psi$ 替换为 $\Box\psi$；将每个公式 $\langle a\rangle\psi$ 替换为 $\Diamond\psi$。

同时，为使得公理系统尽量简单，只将 $\neg$、$\wedge$、$\Box$ 和 μ 视为基本算子，而将 $\vee$、$\Diamond$ 和 ν 等看作派生算子（见第 2 中的相关定义）。此外，为书写简便，公理系统中还会出现布尔连接子 $\rightarrow$、$\leftrightarrow$。Kozen 的模态 μ-演算公理系统 $\mathscr{G}$ 如表 4.1 所示（其中，公理 (Exp) 和规则 (Lfp) 要求 X 在 ψ 中所有的出现均为正出现）。

事实上，该公理系统并不与 Kozen 最初的公理系统相同。比如：这里引入了 (Suc) 公理，来保证语义模型的**连续性** （即：迁移系统中的每个状态都有后继状

表 4.1 模态 μ-演算的公理系统 $\mathscr{G}$

公 理	
所有的重言式	(Tau)
$\Box\neg\varphi \to \neg\Box\varphi$	(Suc)
$\Box(\varphi_1 \wedge \varphi_2) \leftrightarrow (\Box\varphi_1 \wedge \Box\varphi_2)$	(Next)
$\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$	(Kri)
$\psi^X_{\mu X.\psi} \to \mu X.\psi$	(Exp)
推 理 规 则	
$\dfrac{\varphi_1;\quad \varphi_1 \to \varphi_2}{\varphi_2}$	(MP)
$\dfrac{\varphi}{\Box\varphi}$	(XGen)
$\dfrac{\psi^X_\varphi \to \varphi}{\mu X.\psi \to \varphi}$	(Lfp)

态）— 因为它实际上是

$$\Box\varphi \to \Diamond\varphi \tag{4.2}$$

的变形。由于通过 (XGen) 规则可以得到 $\Box true$，于是进而可以得到 $\Diamond true$。公理 (Next)、(Kri)，规则 (MP)、(XGen) 也不是 Kozen 系统中所原有的。它们是某些公理规则的变形、化简及重组。此外，Walukiewicz 在证明 Kozen 系统的完备性时，并没有提及(MP) 规则。但是，该规则实际上是必须的，因此将其添加至公理系统。由于 $\mathscr{G}$ 系统中最核心的公理和规则— (Exp) 和 (Lfp) 是最先在 Kozen 系统中引入的，因此，仍将 $\mathscr{G}$ 系统看作是 Kozen 系统的变形。

同前面 ETL 的公理系统类似，仍用 $\vdash_{\mathscr{G}} \varphi$ 来表示“公式 φ 可在 $\mathscr{G}$ 系统中证出”。为方便起见，在本节将所有的 $\vdash_{\mathscr{G}} \varphi$ 简写为 $\vdash \varphi$。

定理 4.15 ($\mathscr{G}$ 系统的可靠性) *对于 $\mathscr{G}$ 系统而言，若 $\vdash \varphi$，则 $\models \varphi$。*

证明. 公理 (Tau) 以及关于模态算子的公理、规则的有效性可直接验证。现在只说

明公理 (Exp) 的有效性以及规则 (Lfp) 对有效性的保持。

由定义 2.2.18，容易证明性质

$$[\![\psi^X_{\mu X\psi}]\!]_{\langle T,\rho\rangle}E = [\![\psi]\!]_{\langle T,\rho\rangle}E[X/[\![\mu X.\psi]\!]_{\langle T,\rho\rangle}E] \tag{4.3}$$

以及当 X 在 ψ 中正出现时，$[\![\psi]\!]_{\langle T,\rho\rangle}E$ 关于 E 的单调性。即：对于任意的计算树 $\langle T,\rho\rangle$，若 $T'\subseteq T''\subseteq T$，则

$$[\![\psi]\!]_{\langle T,\rho\rangle}E[X/T'] \subseteq [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T''] \tag{4.4}$$

成立。由定义，$[\![\varphi]\!]_{\langle T,\rho\rangle}E = \bigcap\{T'\subseteq T \mid [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T']\subseteq T'\}$。所以对于每个满足 $[\![\psi]\!]_{\langle T,\rho\rangle}E[X/T']\subseteq T'$ 的 T' 有：

$$[\![\psi]\!]_{\langle T,\rho\rangle}E[X/[\![\mu X.\psi]\!]_{\langle T,\rho\rangle}E] \subseteq [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T'] \subseteq T'$$

从而有

$$[\![\psi^X_{\mu X.\psi}]\!]_{\langle T,\rho\rangle}E \subseteq \bigcap\{T'\subseteq T \mid [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T']\subseteq T'\} = [\![\mu X.\psi]\!]_{\langle T,\rho\rangle}E \tag{4.5}$$

上式意味着公理 (Exp) 的有效性。

对于 (Lfp) 规则，若前件 $\psi^X_\varphi \to \varphi$ 是有效的，则对任意的计算树 $\langle T,\rho\rangle$ 有

$$[\![\psi^X_\varphi]\!]_{\langle T,\rho\rangle}E = [\![\psi]\!]_{\langle T,\rho\rangle}E[X/[\![\varphi]\!]_{\langle T,\rho\rangle}E] \subseteq [\![\varphi]\!]_{\langle T,\rho\rangle}E \tag{4.6}$$

于是，若令 $T'' = [\![\psi^X_\varphi]\!]_{\langle T,\rho\rangle}E$，则上式保证了 $[\![\psi]\!]_{\langle T,\rho\rangle}E[X/T'']\subseteq T''$。这样

$$[\![\mu X.\psi]\!]_{\langle T,\rho\rangle}E = \bigcap\{T'\subseteq T \mid [\![\psi]\!]_{\langle T,\rho\rangle}E[X/T']\subseteq T'\} \subseteq [\![\varphi]\!]_{\langle T,\rho\rangle}E \tag{4.7}$$

从而后件也是有效的。 □

定义 4.2.20 称 ψ 对 φ 中的子公式 ϕ 是**可代入的**，如果:

- ϕ 中每个变元的自由出现的不是在 φ 中的约束出现;
- 对 ψ 中每个自由变元 X，ϕ（的指定出现）不在任何 $\mu X.$ 的辖域内。 □

同时，对于代入式 φ^ϕ_ψ，如非特别说明，本章默认遵循如下约定：

1. 若 ϕ 是原子命题 p，则用 ψ 替换 p 在 φ 中的每个出现。
2. 若 ϕ 是自由变元 X，则用 ψ 替换 X 在 φ 中的每个自由出现。
3. 若 ϕ 是其他形式的公式，则允许使用 ϕ 替换其在 φ 某些指定出现。

对于代入操作，由如下引理。

引理 4.16 (代入引理) *设 ϕ 是 φ 中的原子命题或者自由变元，ψ 对 φ 中的 ϕ 可代入，那么：若 $\vdash \varphi$ 则 $\vdash \varphi^{\phi}_{\psi}$。*

证明. 由于 φ 在 $\mathscr{G}$ 中可证，于是存在 φ 在 $\mathscr{G}$ 中的证明序列 $\varphi_0, \ldots, \varphi_n = \varphi$。现在，只需要说明 $(\varphi_0)^{\phi}_{\psi}, \ldots, (\varphi_n)^{\phi}_{\psi}$ 是 φ^{ϕ}_{ψ} 在 $\mathscr{G}$ 中的证明序列即可。

- 显然，若 φ_i 是公理，则 $(\varphi_i)^{\phi}_{\psi}$ 是公理。
- 若 φ_i 可由（前一个序列中）若干下标小于 i 的公式及某规则重写得到，则 $(\varphi_i)^{\phi}_{\psi}$ 必然也能够通过（后一个序列中）相同位置的公式以及该规则重写得到。

注意到 $(\varphi_n)^{\phi}_{\psi}$ 即是 φ^{ϕ}_{ψ}，因此 φ^{ϕ}_{ψ} 必然可在 $\mathscr{G}$ 中可证。 □

在证明 $\mathscr{G}$ 系统的完备性之前，首先证明该系统的若干导出定理及规则。第一组定理/导出规则是关于模态算子的。

引理 4.17 *在 $\mathscr{G}$ 系统中，以下定理/规则可证：*

- **（连续性定理）**$\Box\varphi \to \Diamond\varphi$;
- **（模态规则）**$\dfrac{\neg(\varphi \wedge \psi)}{\neg(\Box\varphi \wedge \Diamond\psi)}$。

证明. 该组定理的证明主要使用公理 (Suc)、(Next)、(Kri) 及 (Tau) 。

- $\Box\varphi \to \Diamond\varphi$。

1. $\vdash \Box\neg\varphi \to \neg\Box\varphi$ ⟅(Suc)⟆
2. $\vdash (\Box\neg\varphi \to \neg\Box\varphi) \to (\Box\varphi \to \neg\Box\neg\varphi)$ ⟅(Tau)⟆
3. $\vdash \Box\varphi \to \neg\Box\neg\varphi$ 即 $\vdash \Box\varphi \to \Diamond\varphi$ ⟅1、2 以及 (MP)⟆

- $\dfrac{\neg(\varphi \wedge \psi)}{\neg(\Box\varphi \wedge \Diamond\psi)}$

1. $\vdash \neg(\varphi \wedge \psi)$ 即 $\vdash \varphi \to \neg\psi$ ⟅前提条件⟆
2. $\vdash \Box(\varphi \to \neg\psi)$ ⟅1 及 (XGen)⟆
3. $\vdash \Box(\varphi \to \neg\psi) \to (\Box\varphi \to \Box\neg\psi)$ ⟅(Kri)⟆
4. $\vdash \Box\varphi \to \Box\neg\psi$ 即 $\vdash \neg(\Box\varphi \wedge \Diamond\psi)$ ⟅2、3 及(MP)⟆

□

第二组导出定理及规则是一组关于“**单调性**”的性质。

引理 4.18 *在 $\mathscr{G}$ 系统中，以下定理/规则可证：*

- **（性质一）**若 $\vdash \varphi_1 \to \varphi_2$，$\phi$ 是原子命题或自由变元，且 ψ 对 φ_1、φ_2 中的 ϕ 可代入，则 $\vdash (\varphi_1)^{\phi}_{\psi} \to (\varphi_2)^{\phi}_{\psi}$。
- **（性质二）**若 $\vdash \varphi_1 \to \varphi_2$，且 X 在 φ_1、φ_2 中的出现均为正出现，则 $\vdash \mu X.\varphi_1 \to \mu X.\varphi_2$。

- （**性质三**）设 $\vdash \psi_1 \to \psi_2$，ϕ 是 φ 中的原子命题或自由变元，且 ψ_1、ψ_2 均对 φ 中的 ϕ 可代入。若 ϕ 在 φ 中所有的出现都是正出现（*resp.* 负出现），则 $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$（*resp.* $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$）。

证明. 本组定理的证明主要采用归纳法。

- **性质一**：由于 $\vdash \varphi_1 \to \varphi_2$，由引理 4.16，有 $\vdash (\varphi_1 \to \varphi_2)^{\phi}_{\psi}$，即：$\vdash (\varphi_1)^{\phi}_{\psi} \to (\varphi_2)^{\phi}_{\psi}$。
- **性质二**：

1. $\vdash \varphi_1 \to \varphi_2$ ⟅前提条件⟆
2. $\vdash (\varphi_1)^{X}_{\mu X.\varphi_2} \to (\varphi_2)^{X}_{\mu X.\varphi_2}$ ⟅1 以及性质一⟆
3. $\vdash (\varphi_2)^{X}_{\mu X.\varphi_2} \to \mu X.\varphi_2$ ⟅(Exp)⟆
4. $\vdash (\varphi_1)^{X}_{\mu X.\varphi_2} \to \mu X.\varphi_2$ ⟅2、3、(Tau) 及(MP)⟆
5. $\vdash \mu X.\varphi_1 \to \mu X.\varphi_2$ ⟅4 及(Lfp)⟆

- **性质三**：对 φ 使用结构归纳法。

1. 当 φ 是原子命题或变元时，结论显然（注意区分 $\varphi = \phi$ 和 $\varphi \neq \phi$ 两种情况）。
2. 若 $\varphi = \neg\varphi'$，且 ϕ 在 φ' 中正出现（resp. 负出现），则 ϕ 在 φ 中负出现（resp. 正出现）。由归纳假设得 $\vdash (\varphi')^{\phi}_{\psi_1} \to (\varphi')^{\phi}_{\psi_2}$（resp. $\vdash (\varphi')^{\phi}_{\psi_2} \to (\varphi')^{\phi}_{\psi_1}$）。于是，由 (Tau) 公理及 (MP) 规则得 $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$（resp. $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$）。
3. 若 $\varphi = \varphi_1 \wedge \varphi_2$，且 ϕ 在 φ_1、φ_2 中正出现（resp. 负出现），则 ϕ 在 φ 中正出现（resp. 负出现）。据归纳假设，对 $i \in \{1,2\}$ 都有 $\vdash (\varphi_i)^{\phi}_{\psi_1} \to (\varphi_i)^{\phi}_{\psi_2}$（resp. $\vdash (\varphi_i)^{\phi}_{\psi_2} \to (\varphi_i)^{\phi}_{\psi_1}$）。于是，由 (Tau) 公理及 (MP) 规则得 $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$（resp. $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$）。
4. 若 $\varphi = \Box\varphi'$，且 ϕ 在 φ' 中正出现（resp. 负出现），则 ϕ 在 φ 中正出现（resp. 负出现）。据归纳假设有 $\vdash (\varphi')^{\phi}_{\psi_1} \to (\varphi')^{\phi}_{\psi_2}$（resp. $\vdash (\varphi')^{\phi}_{\psi_2} \to (\varphi')^{\phi}_{\psi_1}$）。于是，由 (XGen) 规则、(Kri) 公理得 $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$（resp. $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$）。
5. 若 $\varphi = \mu X.\varphi'$，且 ϕ 在 φ' 中正出现（resp. 负出现），则 ϕ 在 φ 中正出现（resp. 负出现）。据归纳假设有：$\vdash (\varphi')^{\phi}_{\psi_1} \to (\varphi')^{\phi}_{\psi_2}$（resp. $\vdash (\varphi')^{\phi}_{\psi_2} \to (\varphi')^{\phi}_{\psi_1}$）。注意到 X 在 φ' 中一定是正出现，于是由性质二得 $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$（resp. $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$）。

□

由于 $\nu X.\psi$ 是 $\neg\mu X.\neg\psi^{X}_{\neg X}$ 的缩写，同样也可以证明性质二关于 ν-算子的形式。即：若 $\vdash \varphi_1 \to \varphi_2$，且 X 在 φ_1、φ_2 中的出现均为正出现，则 $\vdash \nu X.\varphi_1 \to \nu X.\varphi_2$。

定理 4.19 (替换定理)

- 设 $\vdash \psi_1 \to \psi_2$。若 ψ_1 在 φ 中的（指定出现）为正出现（*resp.* 负出现）则 $\vdash \varphi \to \varphi^{\psi_1}_{\psi_2}$（*resp.* $\vdash \varphi^{\psi_1}_{\psi_2} \to \varphi$）。

- 设 $\vdash \psi_1 \leftrightarrow \psi_2$。则 $\vdash \varphi \leftrightarrow \varphi^{\psi_1}_{\psi_2}$

该定理的证明非常简单— 设 p 是某个不在 ψ_1、ψ_2 以及 φ 中出现的原子命题，并令 $\psi = \varphi^{\psi_1}_p$ （即 ψ 是将 ψ_1 在φ 中的指定出现替换为 p 后得到的公式）。于是，$\varphi = \psi^p_{\psi_1}$，$\varphi^{\psi_1}_{\psi_2} = \psi^p_{\psi_2}$。由引理 4.18 中的性质三立即可得结论。对于第二个性质，可以设 ψ 是将 ψ_1 在 φ 中的（指定）正出现替换为新原子命题 p_1，（指定）负出现替换为新原子命题 p_2 得到的公式。于是，$\varphi = \psi^{p_1,p_2}_{\psi_1,\psi_1}$，$\varphi^{\psi_1}_{\psi_2} = \psi^{p_1,p_2}_{\psi_2,\psi_2}$。类似于引理 4.18 中的性质三中的证明，仍对 ψ 按公式结构进行归纳即可得结论。

第三组导出定理及规则是关于不动点算子的。

引理 4.20 *在 $\mathscr{G}$ 系统中，以下定理/规则可证:*

- （**不动点定理**）$\mu X.\psi \leftrightarrow \psi^X_{\mu X.\psi}$;
- （**约束变元换名规则**）*若 Y 不是 ψ 中的自由变元，且对 ψ 中的 X 可代入，则* $\mu X.\psi \leftrightarrow \mu Y.\psi^X_Y$;
- （**强化规则**）$\dfrac{\varphi^X_{\mu X.(\psi\wedge\varphi)} \rightarrow \psi}{\mu X.\varphi \rightarrow \psi}$（*其中 X 在 ψ 中不自由*）

证明. 该组导出定理/规则的证明主要依靠 (Exp) 公理以及 (Lfp) 规则。

- **不动点定理**：$\vdash \mu X.\psi \leftrightarrow \psi^X_{\mu X.\psi}$。

1. $\vdash \psi^X_{\mu X.\psi} \rightarrow \mu X.\psi$ ⟅(Exp)⟆
2. $\vdash \psi^X_{\psi^X_{\mu X.\psi}} \rightarrow \psi^X_{\mu X.\psi}$ ⟅1，X 在 ψ 中正出现，引理 4.18 中的性质三⟆
3. $\vdash \psi^X_{\mu X.\psi} \rightarrow \mu X.\psi$ ⟅2 及 (Lfp)⟆
4. $\vdash \mu X.\psi \leftrightarrow \psi^X_{\mu X.\psi}$ ⟅1、3、(Tau) 及(MP)⟆

- **约束变元换名规则**：

1. $\vdash (\psi^X_Y)^Y_{\mu Y\psi^X_Y} \rightarrow \mu Y.\psi^X_Y$ 即 $\vdash \psi^X_{\mu Y.\psi^X_Y} \rightarrow \mu Y.\psi^X_Y$ ⟅(Exp)⟆
2. $\vdash \mu X.\psi \rightarrow \mu Y.\psi^X_Y$ ⟅1 及(Lfp)⟆
3. $\vdash \psi^X_{\mu X.\psi} \rightarrow \mu X.\psi$ 即 $\vdash (\psi^X_Y)^Y_{\mu X.\psi} \rightarrow \mu X.\psi$ ⟅(Exp)⟆
4. $\vdash \mu Y.\psi^X_Y \rightarrow \mu X.\psi$ ⟅3 及(Lfp)⟆
5. $\vdash \mu X.\psi \leftrightarrow \mu Y.\psi^X_Y$ ⟅2、4、(Tau) 及(MP)⟆

- （**强化规则**[55, 53]）

1. $\vdash \varphi^X_{\mu X.(\psi\wedge\varphi)} \rightarrow \psi$ ⟅前提条件⟆
2. $\vdash \varphi^X_{\mu X.(\psi\wedge\varphi)} \rightarrow \psi \wedge \varphi^X_{\mu X.(\psi\wedge\varphi)}$ ⟅1、(Tau)、(MP)⟆
3. $\vdash \psi \wedge \varphi^X_{\mu X.(\psi\wedge\varphi)} \leftrightarrow \mu X.(\psi \wedge \varphi)$ ⟅不动点定理，以及 X 在 ψ 中不自由⟆
4. $\vdash \varphi^X_{\mu X.(\psi\wedge\varphi)} \rightarrow \mu X.(\psi \wedge \varphi)$ ⟅2、3、(Tau)、(MP)⟆
5. $\vdash \mu X.\varphi \rightarrow \mu X.(\psi \wedge \varphi)$ ⟅4 及 (Lfp)⟆

6. $\vdash \mu X.\varphi \to \psi \wedge \varphi^X_{\mu X.(\psi\wedge\varphi)}$ ⟅3 及替换定理⟆
7. $\vdash \mu X.\varphi \to \psi$ ⟅6、(Tau)、(MP)⟆

□

同样也可以证明不动点定理以及约束变元换名规则关于 ν-算子的形式，即：

- $\nu X.\psi \leftrightarrow \psi^X_{\nu X.\psi}$；
- 若 Y 不是 ψ 中的自由变元，且对 ψ 中的 X 可代入，则 $\nu X.\psi \leftrightarrow \nu Y.\psi^X_Y$。

于是，由约束变元换名规则以及替换定理（定理 4.19），立即可以得到如下结论。

定理 4.21 *对于任意的模态 μ-演算公式 φ, 都存在一个良命名的公式 φ', 使得 $\varphi \leftrightarrow \varphi'$ 在 $\mathscr{G}$ 中可证。*

下面的定理说明了在 $\mathscr{G}$ 中可证明每个公式都于某受卫公式等价。

定理 4.22 (Walukiewicz[56]) *对于任意的模态 μ-演算公式 φ, 都存在一个受卫的公式 φ', 使得 $\varphi \leftrightarrow \varphi'$ 在 $\mathscr{G}$ 中可证。*

证明. 采用公式结构归纳法进行证明。当 φ 是文字、变元或者形如 $\neg\psi$、$\varphi_1 \wedge \varphi_2$、$\Box\psi$ 的公式时证明是非常容易的，下面只说明当 φ 是不动点公式 $\mu X.\psi$ 时的情况。由定理 4.21，不妨设 φ 是良命名的。

设 $\varphi = \mu X.\psi$，并归纳的假设 ψ 中的每个子公式已经全部等价的转化为受卫公式（并已根据定理 4.19 进行了替换）。对于 ψ 中每个非受卫出现的 X，首先做如下等价变换：若某个非受卫的 X 出现在某不动点子公式 $\mu Y.\phi$ 中，由不动点性质有

$$\vdash \mu Y.\phi \leftrightarrow \phi^Y_{\mu Y.\phi}。$$

于是，由替换定理，可以用 $\phi^Y_{\mu Y.\phi}$ 替换 ψ 中的 $\mu Y.\phi$。注意到 Y 在 ϕ 中的出现都是受卫出现，所以 X 在 $\phi^Y_{\mu Y.\phi}$ 中的每个非受卫出现均不在 μY 的辖域内。这样，通过有限次转化，可以将 ψ 等价变形，使得 X 在的每个非受卫出现均不在不动点算子的辖域内。接下来，通过 (Tau) 公理和 (MP) 规则，必将 ψ 等价的变形为形如 $(X \vee \psi_1) \wedge \psi_2$ 的公式，其中 X 在 ψ_1 和 ψ_2 内的出现均为受卫出现。这样，就有 $\vdash \mu X.\psi \leftrightarrow \mu X.((X \vee \psi_1) \wedge \psi_2)$。现在，希望证明有

$$\vdash \mu X.((X \vee \psi_1) \wedge \psi_2) \leftrightarrow \mu X.(\psi_1 \wedge \psi_2)$$

成立，过程如下。

1. $\vdash \psi_1 \wedge \psi_2 \to (X \vee \psi_1) \wedge \psi_2$ ⟅(Tau)⟆
2. $\vdash \mu X.(\psi_1 \wedge \psi_2) \to \mu X.((X \vee \psi_1) \wedge \psi_2)$ ⟅1 及引理 4.18 性质二⟆
3. $\vdash (\psi_1 \wedge \psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)} \leftrightarrow \mu X.(\psi_1 \wedge \psi_2)$ ⟅引理 4.20 性质一⟆

4. $\vdash (((\psi_1 \wedge \psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)} \vee (\psi_1)^X_{\mu X.(\psi_1\wedge\psi_2)}) \wedge (\psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)}) \rightarrow (\psi_1 \wedge \psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)}$ ⟅(Tau)⟆
5. $\vdash (\mu X.(\psi_1 \wedge \psi_2) \vee (\psi_1)^X_{\mu X.(\psi_1\wedge\psi_2)} \wedge (\psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)}) \rightarrow \mu X.(\psi_1 \wedge \psi_2)$ 即：
   $\vdash ((X \vee \psi_1) \wedge \psi_2)^X_{\mu X.(\psi_1\wedge\psi_2)} \rightarrow \mu X.(\psi_1 \wedge \psi_2)$ ⟅3、4 及替换定理⟆
6. $\vdash \mu X.((X \vee \psi_1) \wedge \psi_2) \rightarrow \mu X.(\psi_1 \wedge \psi_2)$ ⟅5 及 (Lfp)⟆
7. $\vdash \mu X.((X \vee \psi_1) \wedge \psi_2) \leftrightarrow \mu X.(\psi_1 \wedge \psi_2)$ ⟅2、6、(Tau) 及(MP)⟆

因此，在 $\mathscr{G}$ 中可证 $\mu X.\psi \leftrightarrow \mu X.(\psi_1 \wedge \psi_2)$，而 $\mu X.(\psi_1 \wedge \psi_2)$ 是一个受卫公式。□

推论 4.23 对每个模态 μ-演算公式 φ，存在良命名的受卫公式 ψ 使得 $\vdash \varphi \leftrightarrow \psi$。并且，对每个模态 μ-演算句子公式 φ，若 ψ 是 φ 的博弈范式，则 $\vdash \varphi \leftrightarrow \psi$。

定义 4.2.21 (析取活跃变元) 给定公式 φ（设 φ 被写为否定范式）及变元 X。称 X 在 φ 中是**析取活跃**的，如果满足下列条件之一：

- $\varphi = X$;
- $\varphi = \varphi_1 \wedge \varphi_2$，$X$ 在 φ_1 中析取活跃，或者 X 在 φ_2 中析取活跃;
- $\varphi = \varphi_1 \vee \varphi_2$，$X$ 在 φ_1 中析取活跃，并且 X 在 φ_2 中析取活跃;
- $\varphi = \Box\psi$ 或者 $\varphi = \Diamond\psi$，同时 X 在 ψ 中析取活跃;
- $\varphi = \mu Y.\psi$ 或者 $\varphi = \nu Y.\psi$，同时 $X \neq Y$，并且 X 在 ψ 中析取活跃。□

引理 4.24 若 X 在 φ 中是析取活跃的，则 $\vdash \mu X.\varphi \rightarrow false$。

证明. 首先，按照公式结构归纳法，证明 $\vdash \varphi^X_{false} \rightarrow false$。

- 当 $\varphi = X$ 时结论显然。
- 当 $\varphi = \varphi_1 \wedge \varphi_2$，且 X 在某个 φ_i（$i = 1, 2$）中析取活跃；或者 $\varphi = \varphi_1 \vee \varphi_2$ 并且 X 在 φ_1、φ_2 中都是析取活跃时，由归纳假设及 (Tau) 公理、(MP) 规则很容易得到 $\vdash \varphi^X_{false} \rightarrow false$。
- 若 $\vdash \psi^X_{false} \rightarrow false$，即 $\vdash \neg\psi^X_{false}$，则由 (XGen) 规则，立即得到 $\vdash \Box\neg\psi^X_{false}$，即 $\vdash \neg\Diamond^X_{flase}$，也就是 $\Diamond\psi^X_{false} \rightarrow false$。再由“连续性定理”（见引理 4.17）有 $\vdash \Box\psi^X_{false} \rightarrow \Diamond\psi^X_{false}$，从而 $\vdash \Box\psi^X_{false} \rightarrow false$。于是，当 $\varphi = \Diamond\psi$ 或者 $\varphi = \Box\psi$ 时结论成立。
- 若 $\vdash \psi^X_{false} \rightarrow false$ 则对于任意的 $Y \neq X$，由单调性定理有 $\vdash \mu Y.\psi^X_{false} \rightarrow \mu Y.false$，$\vdash \nu Y.\psi^X_{false} \rightarrow \nu Y.false$。由不动点定理，可得 $\vdash \mu Y.false \leftrightarrow false$ 以及 $\vdash \nu Y.false \rightarrow false$。于是，当 $\varphi = \mu Y.\psi$ 或者 $\varphi = \nu Y.\psi$ 时结论也成立。

这样，$\vdash \varphi^X_{false} \leftrightarrow false$ 对每个 X 在其中活跃的公式 φ 都成立。由 (Lfp) 规则，立即可以得到 $\vdash \mu X.\varphi \rightarrow false$。□

下面，将利用本节前面的结论以及模态 μ-演算的博弈理论来证明 $\mathscr{G}$ 系统的若

干性质。

定理 4.25 *若句子 φ（写为博弈范式）不可满足，则 $\neg\varphi$ 在 $\mathscr{G}$ 中可证。*

证明. 设 φ 的博弈系统 $\mathcal{G}_\varphi = \langle \mathcal{L}_\varphi, \mathcal{W}_\varphi, \{\varphi\}\rangle$。同时，增加如下限制：

- 对于模态格局，只允许参与者 1 施加格局迁移规则。
- 对于非模态格局，只允许参与者 0 施加格局迁移规则。

显然，这样的博弈系统是“良划分”的和确定的。（相关概念见定理 4.12 的证明过程）由定理 4.12，参与者 1 必然存在关于 $\mathcal{G}_\varphi$ 的取胜策略 f（这是因为，在这种情况下，定理 4.11 的证明仍然有效）。

由于 φ 不可满足，由定理 4.12，参与者 1 必然存在关于 $\mathcal{G}_\varphi$ 的取胜策略 f。由定理 4.8，不妨设 f 是历史无关的。由于历史无关的博弈策略仅取决于当前对决中的最后一个格局，因此不妨将 f 看作从 $\mathcal{L}_\varphi$ 到 $\mathcal{L}_\varphi$ 的函数，并且对每个不含 $false$ 及互补文字的模态格局 Γ 而言，$f(\Gamma)$ 都能由 Γ 经 (modal) 规则得到。

构造 $\mathcal{G}_\varphi$ 的子图 $\mathcal{G}^f_\varphi = \langle \mathcal{L}^f_\varphi, \mathcal{W}^f_\varphi, \{\varphi\}\rangle$，其中：

- $\mathcal{L}^f_\varphi = \mathcal{L}_\varphi$。
- $\mathcal{W}^f_\varphi \subseteq \mathcal{W}_\varphi$，且满足：
  1. 若 Γ 中含有 $false$ 或者互补的文字对，则对于任意的 $\Gamma' \in \mathcal{L}^f_\varphi$，$(\Gamma, \Gamma') \notin \mathcal{W}^f_\varphi$。即：含有互补文字对的格局在 $\mathcal{G}^f_\varphi$ 中没有后继。
  2. 若 Γ 是不含互补的文字对模态格局，则 $(\Gamma, \Gamma') \in \mathcal{W}^f_\varphi$ 当且仅当 $\Gamma' = f(\Gamma)$。即；不含互补的文字对的模态格局在 $\mathcal{G}^f_\varphi$ 中有唯一的后继。
  3. 若 Γ 是不含互补文字对的非模态格局，则对于任意的 $\Gamma' \in \mathcal{L}^f_\varphi$，$(\Gamma, \Gamma') \in \mathcal{W}^f_\varphi$ 当且仅当 $(\Gamma, \Gamma') \in \mathcal{W}_\varphi$。

  以下，对 $\mathcal{G}^f_\varphi$ 中的每个格局 Γ，记

$$\mathbf{IC}_\varphi(\Gamma) \stackrel{\text{def}}{=} \{\mathbf{IC}_\varphi(\psi) \mid \psi \in \Gamma\} \tag{4.8}$$

$$\bigwedge \mathbf{IC}_\varphi(\Gamma) \stackrel{\text{def}}{=} \bigwedge_{\psi \in \mathbf{IC}_\varphi(\Gamma)} \psi \tag{4.9}$$

（关于公式“迭代闭包” $\mathbf{IC}_\varphi(\psi)$，见定义4.2.9）现证明 $\mathcal{G}^f_\varphi$ 满足如下重要性质：

1. 若 $\Gamma \in \mathcal{L}^f_\varphi$ 不是模态格局，则 $\vdash \bigwedge \mathbf{IC}_\varphi(\Gamma) \leftrightarrow \bigvee_{(\Gamma,\Gamma') \in \mathcal{W}^f_\varphi} \bigwedge \mathbf{IC}_\varphi(\Gamma')$。
2. 若 $\Gamma \in \mathcal{L}^f_\varphi$ 是（不含互补文字对的）模态格局，则 $\vdash \neg \bigwedge \mathbf{IC}_\varphi(f(\Gamma))$ 蕴含 $\vdash \neg \bigwedge \mathbf{IC}_\varphi(\Gamma)$。

这两条性质称为 $\mathcal{G}^f_\varphi$ 的“**推理性质**”。性质 2 可直接由引理 4.17 中的“模态规则” 得到。性质 1 可以根据格局 Γ 所使用的格局迁移规则获得。事实上，若 $(\Gamma, \Gamma') \in \mathcal{W}^f_\varphi$，

当从 Γ 获得 Γ' 所使用的规则是公共规则时，必然有 $\vdash \bigwedge \mathbf{IC}_\varphi(\Gamma) \leftrightarrow \bigwedge \mathbf{IC}_\varphi(\Gamma')$；而若能够对 Γ 使用 0-型规则，则会获得两个格局 Γ_1、Γ_2，且必然有 $\vdash \bigwedge \mathbf{IC}_\varphi(\Gamma) \leftrightarrow \bigwedge \mathbf{IC}_\varphi(\Gamma_1) \vee \bigwedge \mathbf{IC}_\varphi(\Gamma_2)$。这里，唯一复杂的情况是使用 (mu-rmv) 或者 (nu-rmv) 规则的情形。当使用 (mu-rmv) 规则时，不妨设 Γ 中的消解公式为 $\mu X.\psi$，则消解后格局的生成公式必然是 ψ。由定义 4.2.9，容易证明 $\vdash \mathbf{IC}_\varphi(\mu X.\psi) \leftrightarrow \mathbf{IC}_\varphi(\psi)$（类似的，也有 $\vdash \mathbf{IC}_\varphi(\nu X.\psi) \leftrightarrow \mathbf{IC}_\varphi(\psi)$）。

注意到 $\mathcal{G}_\varphi^f$ 中的每一条路径 $\Gamma_0, \Gamma_1, \ldots$（这里，“路径” 的定义同常，即要求每个 $(\Gamma_i, \Gamma_{i+1}) \in \mathcal{W}_\varphi^f$）都对应于 $\mathcal{G}_\varphi$ 中的一个对决，同时，在此过程中参与者 1 遵循 f。由于 f 是参与者 1 关于 $\mathcal{G}_\varphi$ 的取胜策略，而 $\mathcal{G}_\varphi^f$ 中的每条初始可达的极大路径（即：无穷路径或者终结于没有后继格局的路径）都对应于 $\mathcal{G}_\varphi$ 中的一个终结对决，因此：

- 若该极大路径有穷，则该路径终结于某个含有互补文字对的格局；
- 若该极大路径无穷，则其中必然包含某无穷踪迹，使得该踪迹中无穷多次出现的（约束）变元中位于最外层者是某 μ-型约束变元。

现在，对 $\mathcal{G}_\varphi^f$ 中的格局进行“删除”。即：对格局 Γ 而言，若 $\bigwedge \mathbf{IC}_\varphi(\Gamma) \to false$ 当前能够在 $\mathscr{G}$ 中证出，则将 Γ 从 $\mathcal{G}_\varphi^f$ 中删除。显然，若某些格局被删除，“推理性质” 在剩余的图中仍然成立。

首先，删除 $\mathcal{G}_\varphi^f$ 中所有非初始可达的格局。其次，删除所有含有 $false$ 或者互补文字对的格局。对于原子命题 p 而言，$\mathbf{IC}_\varphi(p) = p$，$\mathbf{IC}_\varphi(\neg p) = \neg p$ 因此若 Γ 中含有互补文字对，$\mathbf{IC}_\varphi(\Gamma)$ 中也含有互补文字对。所以有 $\vdash \bigwedge \mathbf{IC}_\varphi(\Gamma) \to false$。此外，若某格局 Γ 的所有后继格局均被删除，则此格局也可被删除。

对于剩余图中的每个末端极大连通子图 $\mathcal{S}$，其中必然存在某个覆盖 $\mathcal{S}$ 中所有格局的无穷路径，以及该路径（对应的对决）中的某条无穷踪迹 τ，使得某 μ-型约束变元 X 在 τ 中无穷多次出现，同时对每个在 τ 中无穷多次出现的变元 Y，都有 $Y \lhd_\varphi X$ 成立。这里，不妨设 τ 中的每个公式都出现无穷多次（否则，可截取 τ 的某个后缀作为 τ）。于是，必然存在 $\mathcal{S}$ 中的某个格局 Γ，使得 $\mathbf{D}_\varphi(X) \in \Gamma$，且 τ 有无穷多次经过 Γ 时所取的公式为 $\mathbf{D}_\varphi(X) \in \Gamma$。不妨设 $\mathbf{D}_\varphi(X) = \mu X.\psi$。若 X 在 ψ 中不是析取活跃的，则必然存在 ψ 的某个子公式 ϕ，使得 X 在 ϕ 中不自由，同时 ϕ 出现在 ψ 的某个子公式 $\phi \vee \phi'$（或者 $\phi' \vee \phi$，这里不失一般性）中，并且任何包含 $\phi \vee \phi'$ 的公式 ψ' 不出现在形如 $\phi'' \wedge \psi'$ 的上下文中，其中 X 在 ϕ'' 中是析取活跃的。这里，不妨设 ϕ 是满足上述性质的最外层公式。由于 X 是无穷多次出现在 τ 中的最外层约束变元，因此对于 τ 中的每个公式 ψ'，或者该公式为 $\mathbf{D}_\varphi(X)$，或者 X 在

该公式中自由。由上述假设以及格局迁移规则，$\mathcal{S}$ 中必然存在某个格局 Γ'，使得 τ 在经过该格局时，所取的公式为 $\phi\vee\phi'$，同时，该格局的后继 $\Gamma''=\Gamma'\backslash\{\phi\vee\phi'\}\cup\{\phi\}$ 必然已被删除— 若不然，τ 在 Γ'' 中取的公式必然为 ϕ（因为并且任何包含 $\phi\vee\phi'$ 的公式 ψ' 不出现在形如 $\phi''\wedge\psi'$ 的上下文中），而无穷多次出现 τ 中的最外层约束变元 X 在 ϕ 中不自由，这不可能。于是，由归纳有

$$\vdash\bigwedge\mathbf{IC}_\varphi(\Gamma'\backslash\{\phi\vee\phi'\}\cup\{\phi\})\rightarrow false \tag{4.10}$$

由 (Tau) 及 (MP) 知：

$$\vdash\bigwedge\mathbf{IC}_\varphi(\Gamma')\rightarrow(\mathbf{IC}_\varphi(\phi\vee\phi')\leftrightarrow\mathbf{IC}_\varphi(\phi')) \tag{4.11}$$

由于上式对每个这样的 ϕ 都成立，由“推理性质” 以及替换定理，以及（每个这样的）Γ' 均与 Γ 在同一个连通子图中，从而必然存在某个 $\mu X.\hat{\psi}$，使得 X 在 $\mathbf{IC}_\varphi(\hat{\psi}_p^X)_X^p$ 中析取活跃（其中 p 是不在 $\hat{\psi}$ 中出现的原子命题）— 事实上，$\hat{\psi}$ 可由 ψ 经上述过程删除 X 不在其中析取活跃的析取支后等价变形得到。显然，

$$\vdash\bigwedge\mathbf{IC}_\varphi(\Gamma)\leftrightarrow\bigwedge\mathbf{IC}_\varphi(\Gamma\backslash\{\mu X.\psi\}\cup\{\mu X.\hat{\psi}\}) \tag{4.12}$$

由于 $\mathbf{IC}_\varphi(\mu X.\psi)\in\mathbf{IC}_\varphi(\Gamma)$，立即可得

$$\vdash\bigwedge\mathbf{IC}_\varphi(\Gamma)\rightarrow\mathbf{IC}_\varphi(\mu X.\hat{\psi}) \tag{4.13}$$

由于 X 在 $\mathbf{IC}_\varphi(\hat{\psi}_p^X)_X^p$ 中析取活跃，由引理 4.24 得 $\vdash\mu X.(\mathbf{IC}_\varphi(\hat{\psi}_p^X)_X^p)\rightarrow false$，即：$\vdash\mathbf{IC}_\varphi(\mu X.\hat{\psi})\rightarrow false$（见 107 页给出的性质）。于是有 $\vdash\bigwedge\mathbf{IC}_\varphi(\Gamma)\rightarrow false$。

这样，Γ 就可以从 $\mathcal{S}$ 中删除。接下来，递归删除不可达格局以及无后继的格局，$\mathcal{S}$ 就被分称若干个新的极大连通子图。重复此过程，可以删除 $\mathcal{G}_\varphi^f$ 中的其余末端极大连通子图和末端节点，直至 $\mathcal{G}_\varphi^f$ 变为空。

当格局 $\{\varphi\}$ 被删除时，有 $\vdash\mathbf{IC}_\varphi(\varphi)\rightarrow false$ 成立。由于 φ 是句子，因而 $\mathbf{IC}_\varphi(\varphi)=\varphi$。于是 $\vdash\neg\varphi$。 □

定理 4.26 ($\mathscr{G}$ 的完备性) 对 $\mathscr{G}$ 系统而言，若 $\models\varphi$，则 $\vdash\varphi$。

证明. 设 $X_1,\ldots,X_n$ 是所有在 φ 中自由出现的变元，则任取 n 个互不相同且不在 φ 中出现的原子命题 $p_1,\ldots,p_n$，并令 $\varphi'=\varphi_{p_1,\ldots,p_n}^{X_1,\ldots,X_n}$。显然，$\varphi'$ 是句子，并且容易证明：对任意的计算树 $\langle T,\rho\rangle$ 以及变元指派 E，有 $T,\rho,E\models\varphi$ 当且仅当 $T,\rho'\models\varphi'$，其中 E 满足“对任意的 $x\in T$ 以及 $1\leq i\leq n$，$x\in E(X_i)$ 当且仅当 $p_i\in\rho'(x)$”。由于 φ 是有效的，所以 φ' 也是有效的，于是 $\neg\varphi'$ 是不可满足的。设 φ'' 是 $\neg\varphi'$ 的博弈范式，于是：

1. $\vdash \neg\varphi''$ ⟅定理 4.25⟆
2. $\vdash \neg\varphi'' \leftrightarrow \varphi'$ ⟅推论 4.23⟆
3. $\vdash \varphi'$ ⟅1、2 及替换定理⟆

再注意到 $\varphi = (\varphi')^{p_1,\cdots,p_n}_{X_1,\ldots,X_n}$，于是由引理 4.16 有 $\vdash \varphi$ 成立。 □

4.2.4 相关工作

在本节，给出了一类新的关于模态 μ-演算的博弈系统。该种博弈可以作为模态 μ-演算公式可满足性的测试手段。从取胜条件的角度而言，它是 Emerson-Jutla 的 parity 博弈（parity Game[108]）的一种变种。Emerson-Jutla 博弈的格局也是构建于 parity 树自动机的运行— 但在这种博弈系统中，某个参与者（原文中称为参与者 I，对应于本文中的参与者 0），选择析取分支（Nondeterministic Choice，对应于原文中的 OR-Node）；而另一个参与者（原文中称为参与者 II，对应于本文中的参与者 1）选择合取分支（Universal Choice，对应于原文中的 AAND-Node）。因此，Emerson-Jutla 博弈中的一个"对决" 实质上是本节定义博弈对决中的一条踪迹。而后者中的一个对决实际上对应于前者的一棵博弈树。但是，这种博弈在验证局部可满足性时，需要测试在某策略下另一参与者所有的选择（原文中的"Bundle"），即该博弈树中同一层节点的标记中是否含有互补文字。

最早使用博弈方法对时序逻辑（LTL、CTL）进行公理化的工作由 Lange 和 Stirling 在文 [35] 中给出。该种博弈称为"**焦点博弈**"（Focus Game）。在这种博弈中，一个格局是一个序偶 $\langle\Gamma,\psi\rangle$，其中 $\psi\in\Gamma$，称为该格局中的**焦点公式**。

在焦点博弈的一次对决中，当遇到形如 $\langle\Gamma\cup\{\varphi_1\cup\varphi_2\},\psi\rangle$ 或者形如 $\langle\Gamma,\varphi_1\vee\varphi_2\rangle$ 的格局时，参与者 0 决定将哪个析取支（φ_1 或 φ_2）保留在公式集合中；当遇到形如 $\langle\Gamma,\varphi_1\wedge\varphi\rangle$ 的格局时，参与者 1 决定将哪个合取支设为焦点公式；此外，当遇到"模态格局"，即形如 $\langle\{l_1,\ldots,l_n,\mathsf{AX}\varphi_1,\ldots,\mathsf{AX}\varphi_m,\mathsf{EX}\psi_1,\ldots,\mathsf{EX}\psi_k\},\mathsf{AX}\varphi_i\rangle$ 的格局时（其中 l_1～l_k 都是文字），参与者 1 决定将 ψ_1～ψ_k 中的哪个公式保留在重写后格局中。

焦点博弈的每个终结对决都是有穷序列— 当对决序列中重复出现某一格局时对决一定会终止。之所以能够这样，是因为 CTL 和 LTL 中，只有 next（X、AX、EX）和 until（U、AU、EU）这两类连接子。而 until 算子的编码形式（见 3.5 节）是对应于正规语言 $(a_1)^*;a_2$ 的有穷自动机。而 LTL 和 CTL 公式仅对应于交换深度不超过 2 的 μ-演算公式— 换言之，由 LTL 或 CTL 转化成的 μ-演算公式的每个形如 $\mu X.\psi$ 的子公式中，都没有自由变元出现。

本节将 Martin 和 Stirling 的博弈方法扩展至模态 μ-演算。在本节介绍的博弈系统中，格局是公式集合。当遇到包含 $\varphi_1 \vee \varphi_2$ 的格局时，参与者 0 决定将哪个析取支保留在格局中；当遇到模态格局 $\{l_1, \ldots, l_n, \Box\varphi_1, \ldots, \Box\varphi_m, \Diamond\psi_1, \ldots, \Diamond\psi_k\}$ 时，参与者 1 决定将哪个 ψ_j 保留至下一格局。本节的博弈系统中取消了焦点公式，换之以踪迹的概念（二者实质等价）；并且，对决也不再是有穷格局序列。

系统 $\mathscr{G}$ 最早由 Kozen 给出。公理 (Exp) 和规则 (Lfp) 是该公理系统的核心，它们精确的刻画了不动点算子的本质。然而，在普通的模态逻辑的公理系统上增加公理 (Exp) 和规则 (Lfp) 是就是完备的模态 μ-演算的公理系统，这个问题一直到该系统提出十余年后才由 Walukiewicz 解决（在此之前，Walukiewicz 曾给出过另外一套关于 μ-演算的公理系统，见文 [105]）。在文 [56] 中，Walukiewicz 证明公理系统完备性的方法是 “tableau 互模拟” 以及 “tableau 演绎” 技术。Tableau 方法是另一种公式可满足性测试手段。通过这种技术，Walukiewicz 证明了任何一个模态 μ-演算公式都等价于某个“合取分离式”（Aconjunctive Formula）。进而，利用文 [109] 中的结论：“合取分离公式 φ 是可满足的，当且仅当 φ 的**置换式** 是可满足的” 来证明 Kozen 系统的完备性。其中，φ 的置换式是将 φ 中的不动点算子删除，μ-型约束变元替换为 $false$，ν-型约束变元替换为 $true$ 而得到的公式。同基于 $tableau$ 的完备性证明相比，本节给出的方法相对简洁。

4.3 线性 μ-演算的博弈系统及公理系统

4.3.1 线性 μ-演算的相关概念及博弈系统

本节给出线性 μ-演算的博弈系统定义。在线性 μ-演算中，“良命名公式”、“约束变元绑定式” 以及“约束变元层次关系” 等概念的定义与 4.2.1 节完全相同。此外，“受卫公式”、“公式的否定范式”、“公式的博弈范式” 以及“迭代闭包” 的概念只需将模态 μ-演算的相应定义中 $\Box$ 和 $\Diamond$ 两个算子合并为一个算子 $\bigcirc$ 即可得到线性 μ-演算中的对应概念。

这里，需要给出 parity 树自动机的对应概念— “迟滞 parity（字）-自动机”。这种自动机是本节线性 μ-演算公式的一种判定模型，同时是线性 μ-演算符号化模型检验算法的基础（见第 6 章）。

定义 4.3.1 (迟滞交错 parity 自动机) 一个迟滞交错 ***parity*** 自动机（*Stuttering Alternating Parity Word Automaton*, 简写为 $SAPW$）是一个四元组 $\mathcal{A} = \langle Q, \delta, q, \Omega \rangle$，其中:

- Q 是一个有穷**状态集合**。
- $\delta : Q \to \chi'(Q)$，是一个**迁移函数**。其中：$\chi'(Q)$ 是满足下列条件的最小集合：
  - $true \in \chi(Q)$；$false \in \chi(Q)$；
  - 若 $p \in AP$，则 $p \in \chi'(Q)$，$\neg p \in \chi'(Q)$；
  - 若 $q', q_1, q_2 \in Q$，则 $q', \bigcirc q', q_1 \wedge q_2, q_1 \vee q_2 \in \chi'(Q)$。
- $q \in Q$，是一个**初始状态**。
- 接收条件 Ω 一个部分函数，即：$\Omega : Q \rightsquigarrow \mathbb{N}$。 □

同样，这里也没有强调字母表，这是因为该种自动机识别的是语言是线性结构。因此，其暗含的字母表是 2^{AP}。

定义 4.3.2 (线性结构在 SAPW 上的运行) 给定线性结构 $\pi \in (2^{AP})^{\omega}$，以及 $SAPW$ $\mathcal{A} = \langle Q, \delta, q_0, \Omega \rangle$，则 $\mathcal{A}$ 在 π 上的一个**运行** 是一棵 $(Q \times \mathbb{N})$-标记树 $\langle T, \rho \rangle$，其中：

- $\rho(\epsilon) = (q_0, 0)$。
- 对于任意的 $x \in \hat{T}$，设 $\hat{\rho}(x) = (q, i)$，则
  - $\delta(q) \neq false$。
  - 若 $\delta(q) = true$，则不对 q 和 y 做任何限制。
  - 若 $\delta(q) = p$，则 $p \in \pi(i)$；若 $\delta(q) = \neg p$，则 $p \notin \pi(i)$。
  - 若 $\delta(q) = q'$，则 x 在 T 中有唯一的子节点 $x \cdot 0$，且 $\rho(x \cdot 0) = (q', i)$。
  - 若 $\delta(q) = \bigcirc q'$，则 x 在 T 中有唯一的子节点 $x \cdot 0$，且 $\rho(x \cdot 0) = (q', i+1)$。
  - 若 $\delta(q) = q_1 \wedge q_2$，则 x 在 T 有两个子节点 $x \cdot 0$ 与 $x \cdot 1$，其中 $\rho(x \cdot 0) = (q_1, i)$，$\rho(x \cdot 1) = (q_2, i)$。
  - 若 $\delta(q) = q_1 \vee q_2$，则 x 在 T 中有唯一的子节点 $x \cdot 0$，其中 $\rho(x \cdot 0) = (q_1, i)$ 或者 $\rho(x \cdot 0) = (q_2, i)$。

对于 T 中的任意一条无穷路径 $\sigma = x_0, x_1, \ldots$，令 $\mathbf{Inf}(\Omega(\sigma)) = \{n \in \mathbb{N} \mid$ 有无穷多个 $i \in \mathbb{N}$ 使得 $\Omega(\rho_1(x_i)) = n\}$。其中 ρ_1 是 ρ 的第一投影函数，即：若 $\rho(x) = (q, i)$，则 $\rho_1(x) = q$。称 $\langle T, \rho \rangle$ 是一个**可接收运行**，当且仅当对 T 中的每条无穷路径 σ，都有 $\max(\mathbf{Inf}(\Omega(\sigma)))$ 为偶数。

称线性结构 π 可被 $\mathcal{A}$ **接收**（同样记作 $\pi \in \mathbf{L}(\mathcal{A})$），当且仅当存在其 $\mathcal{A}$ 在 π 上存在某个可接收运行。 □

之所以将该种自动机称为“迟滞的”（Stuttering），是因为对该种自动机而言，当读取当前字母后，输入字上的“读头” 不是每次都“后移” 一个位置，还有可能停留在原位置。

对每个良命名的句子 φ（假设 φ 已写为否定范式），都可以构造一个 SAPW

$\mathcal{A}_\varphi = \langle Q_\varphi, \delta_\varphi, q_\varphi, \Omega_\varphi \rangle$，其中：

- $Q_\varphi = \{q_\psi \mid \psi$ 是 φ 的子公式$\}$（这样，初始状态 q_φ 自然属于 Q_φ）。
- δ_φ 定义如下：
  - $\delta_\varphi(q_{true}) = true$；$\delta_\varphi(q_{false}) = false$。
  - 对 φ 中的原子命题 p，$\delta_\varphi(q_p) = p$；$\delta_\varphi(q_{\neg p}) = \neg p$。
  - 对 φ 中的（约束）变元 X，$\delta_\varphi(q_X) = q_{\mathbf{D}_\varphi(X)}$。
  - $\delta_\varphi(q_{\varphi_1 \vee \varphi_2}) = q_{\varphi_1} \vee q_{\varphi_2}$；$\delta_\varphi(q_{\varphi_1 \wedge \varphi_2}) = q_{\varphi_1} \wedge q_{\varphi_2}$。
  - $\delta_\varphi(q_{\bigcirc\psi}) = \bigcirc q_\psi$。
  - $\delta_\varphi(q_{\mu X.\psi}) = q_\psi$；$\delta_\varphi(q_{\nu X.\psi}) = q_\psi$。
- 接收条件 Ω_φ 是任意一个满足如下约束的部分函数：
  - Ω_φ 在状态 q_ψ 处有定义当且仅当 ψ 是 φ 中的（约束）变元；
  - 若 X 是 φ 中的 μ-型约束变元，则 $\Omega_\varphi(q_X)$ 为奇数；若 X 是 φ 中的 ν-型约束变元，则 $\Omega_\varphi(q_X)$ 为偶数。
  - 若 $Y \lhd_\varphi X$，则 $\Omega_\varphi(q_Y) < \Omega_\varphi(q_X)$。

同定理 4.4，下面的定理说明了 SAPW 可以作为线性 μ-演算的判定模型。

定理 4.27 *设 φ 是良命名的线性 μ-演算句子（设 φ 已经被写为否定范式），则对于任意线性结构 π 有：$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$。*

之所以使用“迟滞” 形式，是因为这种自动机与线性 μ-演算的博弈迁移规则存在一一对应，它是 4.2.1 节中介绍的 APT 的线性版本。在第 6 章中，将会看到每个 $\mathcal{A}_\varphi$ 都能够等价转化为一个的以 2^{AP} 为字母表的标准 APW。

线性 μ-演算句子公式的博弈系统的定义同模态 μ-演算，只是格局迁移规则稍微不同。线性 μ-演算的格局迁移规则分类同前，但是由于在线性 μ-演算中算子 $\Box$ 和 $\Diamond$ 被合并为一个 $\bigcirc$，因此 (modal) 规则修改如下：

$$\frac{\{l_1, \ldots, l_k, \bigcirc\varphi_1, \ldots, \bigcirc\varphi_n\}}{\{\varphi_1, \ldots, \varphi_n\}} \qquad \text{(modal)}$$

其中 $l_1, \ldots, l_k$ 都是文字。

线性 μ-演算中“对决”、“踪迹”、“取胜策略”、“博弈树” 以及对决的取胜条件严格与模态 μ-演算中的情形相同。同样，也可以证明下列结论，其证明过程只需将 4.2.2 节中的证明修改为相应的线性版本即可。

定理 4.28 *若线性 μ-演算句子 φ 是可满足的，则参与者 0 有关于 $\mathcal{G}_\varphi$ 的（历史无关的）取胜策略。*

定理 4.29 若线性 μ-演算句子 φ 是不可满足的，则参与者 1 有关于 $\mathcal{G}_\varphi$ 的取胜策略。

推论 4.30 (博弈的确定性) 参与者 0（*resp.* 参与者 1）有关于 $\mathcal{G}_\varphi$ 的取胜策略当且仅当 φ 是可满足（*resp.* 不可满足）的。

推论 4.31 (博弈的求解复杂性) 线性 μ-演算博弈的求解（即：确定具有取胜策略的参与者）问题是*PSPACE-complete*的。

证明. 由推论 4.30 以及线性 μ-演算公式的可满足性是 PSPACE-complete 的[91] 可立即得到。 □

4.3.2 线性 μ-演算公理系统：完备性的证明

在本节，将使用 4.3.1 中给出的线性 μ-演算的博弈理论来证明 Kaivola 公理系统 $\mathscr{H}$ 的完备性（见文 [52, 53]）。Kaivola 的线性 μ-演算公理系统见表 4.2。

表 4.2 线性 μ-演算的公理系统 $\mathscr{H}$

公 理	
所有的重言式	(Tau)
$\bigcirc\neg\varphi \leftrightarrow \neg\bigcirc\varphi$	(Next)
$\bigcirc(\varphi_1 \rightarrow \varphi_2) \rightarrow (\bigcirc\varphi_1 \rightarrow \bigcirc\varphi_2)$	(Kri)
$\psi^X_{\mu X.\psi} \rightarrow \mu X.\psi$	(Exp)
推 理 规 则	
$\dfrac{\varphi_1;\quad \varphi_1 \rightarrow \varphi_2}{\varphi_2}$	(MP)
$\dfrac{\varphi}{\bigcirc\varphi}$	(XGen)
$\dfrac{\psi^X_\varphi \rightarrow \varphi}{\mu X.\psi \rightarrow \varphi}$	(Lfp)

注意，同 $\mathscr{G}$ 系统相比，$\mathscr{H}$ 系统中关于时序算子的公理只有两条。因为 $\mathscr{G}$ 系统

中公理

$$\Box(\varphi_1 \wedge \varphi_2) \leftrightarrow (\Box\varphi_1 \wedge \Box\varphi_2)$$

对应的线性形式

$$\bigcirc(\varphi_1 \wedge \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$$

在系统 $\mathscr{H}$ 是可证的（见引理 4.33）。

定理 4.32 ($\mathscr{H}$ 的可靠性) *在 $\mathscr{H}$ 系统中，若 $\vdash \varphi$，则 $\models \varphi$。*

证明. 公理 (Kri)、(Next) 的有效性，以及规则 (XGen) 对有效性的保持可直接验证。公理 (Exp) 的有效性以及规则 (Lfp) 对有效性的保持只需将定理 4.15 的相应证明换为线性 μ-演算版本即可。 □

在 $\mathscr{H}$ 系统中，同样可证明代入引理（同引理 4.16）和替换定理（同定理 4.19）。同时，还可以证明如下的导出定理/规则。

引理 4.33 *在 $\mathscr{H}$ 系统中，以下定理/规则可证:*

- $\bigcirc(\varphi_1 \wedge \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$;
- $\bigcirc(\varphi_1 \vee \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \vee \bigcirc\varphi_2)$。

证明. 该组导出定理的证明主要使用 (Next) 和 (Kri) 公理。

- $\bigcirc(\varphi_1 \wedge \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$
  1. $\vdash \bigcirc(\varphi_1 \to \neg\varphi_2) \leftrightarrow (\bigcirc\varphi_1 \to \bigcirc\neg\varphi_2)$ 即
     $\vdash \bigcirc\neg(\varphi_1 \wedge \varphi_2) \leftrightarrow (\neg\bigcirc\varphi_1 \vee \bigcirc\neg\varphi_2)$ ⟅(Kri)⟆
  2. $\vdash \bigcirc\neg\varphi_2 \leftrightarrow \neg\bigcirc\varphi_2$ ⟅(Next)⟆
  3. $\vdash \bigcirc\neg(\varphi_1 \wedge \varphi_2) \leftrightarrow (\neg\bigcirc\varphi_1 \vee \neg\bigcirc\varphi_2)$ 即
     $\vdash \bigcirc\neg(\varphi_1 \wedge \varphi_2) \leftrightarrow \neg(\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$ ⟅1、2、以及替换定理⟆
  4. $\vdash \bigcirc\neg(\varphi_1 \wedge \varphi_2) \leftrightarrow \neg\bigcirc(\varphi_1 \wedge \varphi_2)$ ⟅(Next)⟆
  5. $\vdash \neg\bigcirc(\varphi_1 \wedge \varphi_2) \leftrightarrow \neg(\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$ ⟅3、4、以及替换定理⟆
  6. $\vdash \bigcirc(\varphi_1 \wedge \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \wedge \bigcirc\varphi_2)$ ⟅5、(Tau) 及 (MP)⟆
- $\bigcirc(\varphi_1 \vee \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \vee \bigcirc\varphi_2)$
  1. $\vdash \bigcirc(\neg\varphi_1 \to \varphi_2) \leftrightarrow (\bigcirc\neg\varphi_1 \to \bigcirc\varphi_2)$ 即
     $\vdash \bigcirc(\varphi_1 \vee \varphi_2) \leftrightarrow (\bigcirc\neg\varphi_1 \to \bigcirc\varphi_2)$ ⟅(Kri)⟆
  2. $\vdash \bigcirc\neg\varphi_1 \leftrightarrow \neg\bigcirc\varphi_1$ ⟅(Next)⟆
  3. $\vdash \bigcirc(\varphi_1 \vee \varphi_2) \leftrightarrow (\neg\bigcirc\varphi \to \bigcirc\varphi_2)$ 即
     $\vdash \bigcirc(\varphi_1 \vee \varphi_2) \leftrightarrow (\bigcirc\varphi_1 \vee \bigcirc\varphi_2)$ ⟅1、2 及替换定理⟆

□

上述引理从语法角度阐明了：在线性 μ-演算中，$\bigcirc$ 算子关于 $\wedge$ 和 $\vee$ 都是分配的。而在模态 μ-演算中，$\Box$ 算子只关于 $\wedge$ 分配，$\Diamond$ 算子只关于 $\vee$ 分配。

引理 4.34 *在 $\mathscr{H}$ 系统中，以下定理/规则可证:*

- （**性质一**）若 $\vdash \varphi_1 \to \varphi_2$，且 ψ 对 φ_1、φ_2 中的 ϕ 可代入（ϕ 是原子命题或自由变元），则 $\vdash (\varphi_1)^{\phi}_{\psi} \to (\varphi_2)^{\phi}_{\psi}$。
- （**性质二**）若 $\vdash \varphi_1 \to \varphi_2$，且 X 在 φ_1、φ_2 中的出现均为正出现，则 $\vdash \mu X.\varphi_1 \to \mu X.\varphi_2$。
- （**性质三**）设 $\vdash \psi_1 \to \psi_2$，ϕ 是 φ 中的原子命题或自由变元，且 ψ_1、ψ_2 对 φ 中的 ϕ 可代入。若 ϕ 在 φ 中所有的出现都是正出现（*resp.* 负出现），则 $\vdash \varphi^{\phi}_{\psi_1} \to \varphi^{\phi}_{\psi_2}$（*resp.* $\vdash \varphi^{\phi}_{\psi_2} \to \varphi^{\phi}_{\psi_1}$）。

证明. 同引理 4.18。 □

引理 4.35 *在 $\mathscr{H}$ 系统中，以下定理/规则可证:*

- （**不动点定理**）$\mu X.\psi \leftrightarrow \psi^{X}_{\mu X.\psi}$;
- （**约束变元换名规则**）若 Y 不是 ψ 中的自由变元，且对 ψ 中的 X 可代入，则 $\mu X.\psi \leftrightarrow \mu Y.\psi^{X}_{Y}$;
- （**强化规则**）$\dfrac{\varphi^{X}_{\mu X.(\psi \wedge \varphi)} \to \psi}{\mu X.\varphi \to \psi}$（其中 X 在 ψ 中不自由）。

证明. 同引理 4.20。 □

于是，下面的定理同样在 $\mathscr{H}$ 中成立。

定理 4.36 *对于任意的线性 μ-演算公式 φ，都存在一个良命名的公式 φ'，使得 $\varphi \leftrightarrow \varphi'$ 在 $\mathscr{H}$ 中可证。*

定理 4.37 *对于任意的模态 μ-演算公式 φ，都存在一个受卫的公式 φ'，使得 $\varphi \leftrightarrow \varphi'$ 在 $\mathscr{G}$ 中可证。*

证明. 同定理 4.22。 □

推论 4.38 *对每个线性 μ-演算公式 φ 而言，都存在良命名的受卫公式 ψ 使得 $\vdash \varphi \leftrightarrow \psi$。同时，对每个线性 μ-演算句子公式 φ，若 ψ 是 φ 的博弈范式，则 $\vdash \varphi \leftrightarrow \psi$。*

接下来，定义**析取活跃变元**的线性 μ-演算版本。

定义 4.3.3 (析取活跃变元) *给定公式 φ（设 φ 被写为否定范式）及变元 X。称 X 在 φ 中是**析取活跃**的，如果满足下列条件之一:*

- $\varphi = X$;
- $\varphi = \varphi_1 \wedge \varphi_2$，$X$ 在 φ_1 中析取活跃，或者 X 在 φ_2 中析取活跃;
- $\varphi = \varphi_1 \vee \varphi_2$，$X$ 在 φ_1 中析取活跃，并且 X 在 φ_2 中析取活跃;

- $\varphi = \bigcirc\psi$, 同时 X 在 ψ 中析取活跃;
- $\varphi = \mu Y.\psi$ 或者 $\varphi = \nu Y.\psi$, 同时$X \neq Y$, 并且 X 在 ψ 中析取活跃。 □

引理 4.39 若 X 在 φ 中是析取活跃的, 则 $\vdash \mu X.\varphi \to false$。

证明. 同引理 4.24。 □

于是，由前面得到的关于线性 μ-演算的博弈结果，可以证明关于 $\mathscr{H}$ 系统的如下定理。

定理 4.40 若句子 φ（写为博弈范式）不可满足, 则 $\neg\varphi$ 在 $\mathscr{G}$ 中可证。

证明. 同定理 4.25。 □

定理 4.41 ($\mathscr{H}$ 的完备性) 在 $\mathscr{H}$ 中, 对任意的线性 μ-演算公式 φ 而言, 若 $\models \varphi$, 则 $\vdash \varphi$。

证明. 同定理 4.26。 □

4.3.3 相关工作

线性 μ-演算的博弈系统是模态 μ-演算博弈系统的特例。这里，唯一发生变化的格局迁移规则是 (modal) 规则。事实上，针对线性 μ-演算的模态格局时，参与者 1 并没有做实际的“选择” — 模态格局的对应的重写后格局是确定的。在有些博弈系统中（如 [34]），可能存在如下的 1-型规则：

$$\frac{\Gamma}{\Gamma'} \qquad \text{(weaken)}$$

其中 $\Gamma' \subseteq \Gamma$。但是，对于本节中所涉及的取胜条件而言，参与者 1 使用这条规则不会产生对其更加有利的格局，因而未将其引入。因此，在线性 μ-演算的博弈系统中，能够实际影响对决的只有参与者 0 所采取的策略。

但是，这种博弈系统同样可以证明线性 μ-演算系统的完备性，事实上，$\mathscr{H}$ 系统也可以看作是 $\mathscr{G}$ 系统的特例，它是将 $\mathscr{G}$ 系统中关于算子 $\Box$ 和 $\Diamond$ 的公理/规则替换为关于 $\bigcirc$ 的公理/规则所获得的公理系统；而公理系统的核心— (Exp) 公理以及 (Lfp) 规则则得以保留。

Kaivola 的完备性证明是基于特殊的 tableau 方法证明的。这种 tableau 的每个格局是一个三元组 $\langle i, \Gamma, \mathbf{d}\rangle$，其中 $i \in \mathbb{N}$，Γ 是一个公式集合，$\mathbf{d}$ 是一个重命名列表，它记录了每个不动点公式的别名列表。在这种 tableau 系统中，形如 $\langle i, \Gamma \cup \{\mu X.\psi\}, \mathbf{d}\rangle$ 或者 $\langle i, \Gamma \cup \{\nu X.\psi\}, \mathbf{d}\rangle$ 的格局被重写为 $\langle i, \Gamma \cup \{\psi_Z^X\}, \mathbf{d} \cup \{(Z, \mu X.\psi)\}$，其中 Z 是不在 Γ 和 $\mathbf{d}$ 的别名列表中出现的新变元；而形如 $\langle i, \Gamma \cup \{Z\}, \mathbf{d}\rangle$ 的格局被重写为

$\langle i, \Gamma \cup \{\psi_Z^X\}, \mathbf{d}\rangle$，其中 $(Z, \mu X.\psi) \in \mathbf{d}$。

在这种 tableau 规则下，一个无穷格局序列的踪迹中只有一个变元会无穷多次出现（即无穷多次在该踪迹中出现的变元中的最外层者），而出现在其内部的变元会被无穷多次重命名。基于该种技术，Kaivola 证明了："任何一个线性 μ-演算公式 φ，都存在一个与之等价的，且写成 *'Banan-*范式*'* 的公式 ψ，使得 $\vdash_{\mathscr{H}} \varphi \leftrightarrow \psi$。" 其中，Banan-范式是一种比"合取分离式" 更强的形式。而对于不可满足的（Banan-范式）公式 ψ，在 $\mathscr{H}$ 中证明$\neg\psi$ 是容易的。

本节的 $\mathscr{H}$ 系统的完备性证明，是基于线性 μ-演算博弈系统给出的。线性 μ-演算博弈系统可以看作是模态 μ-演算博弈系统的特例。同时，它在线性 μ-演算的符号化模型检验算法中也起着重要的作用（见 6.3 节）。

4.4 本章小结

就模态 μ-演算和线性 μ-演算，本章给出了 parity 博弈的一种变种，并利用该种博弈理论给出了 $\mathscr{G}$ 系统和 $\mathscr{H}$ 系统新的完备性证明。该种博弈的判定问题可以规约为公式的可满足性问题，因此它可以看做是 μ-演算的可满足性测试手段— 事实上，从公式的博弈系统以及参与者 0（历史无关）的取胜策略中即可提取出公式的模型。

μ-演算是从模态逻辑中增加 μ-算子后获得的时序逻辑。$\mathscr{G}$ 系统和 $\mathscr{H}$ 系统的核心是 (Exp) 公理和 (Lfp) 规则。它们简洁、直观的刻画了 μ 算子的"极小不动点" 性质。但是，它们是否完全的刻画了 μ-算子的性质（即：$\mathscr{G}$ 和 $\mathscr{H}$ 系统是否完备），在文 [56] 之前，一直是一个开放的问题。

Walukiewicz 和 Kaivola 给出的关于 $\mathscr{G}$、$\mathscr{H}$ 系统的完备性证明都是基于 tableau 方法进行的。通过 tableau 互模拟等技术来演绎证明 μ-演算公式与某些特定形式公式的等价性，从而可以给出有效公式演绎序列的构造过程。

在本章，μ-演算公理系统完备性是基于博弈方法给出的。相比而言，该证明过程相对简洁。

第五章　基于 tableau 的交错 ETL 和 APSL 符号化模型检验

5.1　引言

本文第 3 章介绍了若干采用（非确定）ω-自动机作为连接子的扩展时序逻辑（诸如 ETL_l、ETL_f、ETL_r）的公理化问题。同时，介绍了如何通过编码时序连接子而获得其他逻辑的公理系统的方法。

对于各类时序逻辑而言，除推理问题之外，另一个重要的研究内容是时序逻辑的模型检验问题。甚至，后者的重要性要超过前者。

首先尝试对线性时序逻辑（LTL）采用基于 BDD 的符号化方法进行模型检验的工作由 Grumberg、Clarke 和 Hamaguchi 等人给出 [110]。同显式的 LTL 模型检验算法相比（见图 1.2），该种方法不是将待检验的规约公式（取非后）转化为 Büchi 自动机，而是转化为一个 tableau。LTL 公式 φ 的 tableau $\mathcal{T}_\varphi$ 可以看做是一个特殊的公平迁移系统，并且也满足性质（称为**语言性质**）：“对任意的线性结构 π 而言，$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$ ”。但同 Büchi 自动机相比，tableau 中的每个“状态” 由 φ 中的“**基础公式**”（elementary formula）的真值决定；同时，其“迁移关系”也可以由一个关于基础公式的布尔公式统一刻画。因此，$\mathcal{T}_\varphi$ 能够非常直接的采用基于 BDD 的符号化表示。于是，$\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。这样，LTL 的模型检验问题便转化成了 CTL 的（符号化）模型检验问题（过程见 2.3.3 节）。

如前所述，许多重要的时序性质并不能被 LTL 所描述。事实上，为弥补 LTL 表达能力的缺陷，工业界使用了诸多具有等价与 ω-正规语言表达能力的时序逻辑描述规约（如 PSL[31]）。本章讨论使用交错 ω-自动机作为连接子的扩展时序逻辑及其某些特例的符号化模型检验问题。这里，主要关心三类扩展时序逻辑— ATL_f、ATL_l 以及 ATL_r，即：分别使用 AFW、ALW 以及 ABW 作为时序连接子的交错 ETL。

研究以交错自动机作为连接子的扩展时序逻辑（ATL_f、ATL_l、ATL_r）的符号化模型检验技术，是出于下列考虑：

1. 交错自动机的迁移结构比非确定自动机的迁移结构更加灵活、方便，能够更加简洁的（即：使用更少的状态）描述连接子。而非确定自动机可以看做其特例。
2. 事实上，工业界中的规约语言中，其语法成分中的确包含对应与交错自动机连接子的成分。比如，在 PSL 的 SERE 中，& 和 | 操作子实际对应于自动机的全

局迁移和非确定迁移（见定义 2.2.25）。

3. 虽然交错自动机的迁移结构比非确定自动机的迁移结构更具一般性，但是采用交错自动机作为连接子的扩展时序逻辑的模型检验算法的复杂度并不比采用非确定自动机作为连接子的扩展时序逻辑高。

作为特例，本章还将给出 PSL 的某变种— APSL 的符号化模型检验算法。在该种时序逻辑中，辅助公式构造子不再是 SERE，而是有穷字上的自动机。此外，APSL 中还包含 PSL 中所特有的时序连接子，如 abort、trigger 等（见定义 2.2.27）。这些时序连接子不能被自动机编码。

研究扩展时序逻辑的符号化模型检验技术还具有如下意义。

- 如前所述，finite、looping 以及 repeating 接收条件的自动机连接子分别描述了活性性质、安全性质以及一般的 ω-正规性质。研究采用以这些自动机为时序连接子的时序逻辑的符号化模型检验问题，可以给出活性时序连接子、安全时序连接子以及一般 ω-时序连接子的符号化验证方式。
- ATL_f、ATL_l 以及 ATL_r 都具有等价于 ω-正规语言的描述能力。具有等价于 ω-正规语言的时序逻辑在工业界有着十分广泛的应用。某些 ETL 的变种，如 PSL，已经被接纳为工业标准。多数构建于时序连接子的时序逻辑（如 LTL）都能看做是这些逻辑的片段。这些逻辑的符号化模型检验技术，可以看做是 LTL 符号化模型检验算法的扩展。它能够提供一个统一的符号化模型检验框架。
- 这些符号化模型检验算法均是基于 tableau 实现。因此，这些算法可以在对现有模型检验工具 、NuSMV 的基础上扩展实现。在本文的第 7 章将会介绍某些扩展时序逻辑符号模型检验算法的工具实现以及实验结果。

本章内容结构组织如下。

1. 5.2 节回顾扩展时序逻辑模型检验中所需的两个基础概念：“自动机公式在无穷字上的运行”、“自动机公式关于无穷字的拒绝例证”。同时介绍“自动机公式相似”、“公式否定范式”、“自动机公式关于布尔公式代入” 等概念。最后证明自动机公式的展开定理。
2. 5.3 节～5.5 节依次介绍三种交错 ETL（ATL_f、ATL_l、ATL_r）的符号化模型检验技术。介绍其基础公式（Elementary Formulas）、满足集以及 tableau 的定义。证明其 tableau 的语言性质。最后，给出这些算法基于 BDD 的符号化实现。
3. 5.6 节给出 PSL 的某个变种— APSL 的符号化模型检验算法。该种逻辑的分支部分等价于 CTL，其线性部分（称为 AFL）等价于 FL（即：PSL 的线性部分）。由于 CTL 的模型检验技术已被充分研究，因此本节只介绍针对 AFL 的符号化

模型检验算法。

5.2 公共概念及性质

在本章的 扩展时序逻辑符号化模型检验技术中，需要使用的基础概念有：“**自动机公式在无穷字上的运行**”、“**自动机公式关于无穷字的拒绝例证**”。这两个概念的具体见定义 2.2.23 及定义 2.2.24。此外，定理 2.9 阐明了“拒绝例证” 性质。即：对于自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，线性结构 π，以及位置 i 而言：$\pi, i \models \neg\varphi$ 当且仅当 φ 存在一个在 π 上开始于 i 的拒绝例证。

为方便起见，本节还需引入下列概念。这些概念在 ATL_f、ATL_l、ATL_r 的符号化模型检验中有着非常重要的作用。

定义 5.2.1 (公式的否定范式) 给定公式 φ，可以通过德摩根律以及模式 $\neg\bigcirc\varphi \leftrightarrow \bigcirc\neg\varphi$、$\neg\neg\varphi \leftrightarrow \varphi$、$\neg true \leftrightarrow false$，$\neg false \leftrightarrow true$ 将其等价变形，使得 $\neg$ 仅出现在原子命题和自动机连接子之前，这样得到的最终形式称为 φ 的**否定范式**。 □

定义 5.2.2 (自动机公式相似) 设自动机公式 $\varphi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, \Omega \rangle$，则对任意的 $q' \in Q$，称公式 $\varphi^{q'} = \mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$ 与 φ **相似**，记作 $\varphi \sim \varphi^{q'}$。 □

显然，$\sim$ 是一个等价关系。若 $\varphi \sim \psi$，则 φ 和 ψ 最多只相差自动机连接子的初始状态。

定义 5.2.3 (相似等价类) 给定公式 φ（否定范式），则用 $[\psi]_\varphi^\sim$ 表示 φ 的自动机子公式 ψ 关于 $\sim$ 所在的等价类。 □

例 5.2.1 设 $\varphi = (\mathcal{A}_1^{q_1}(p_1, p_2) \vee \bigcirc\neg\mathcal{A}_2^q(p_2, \bigcirc p_1, p_1 \wedge \neg p_2)) \wedge (\neg\mathcal{A}_2^{q'}(p_2, \bigcirc p_1, p_1 \wedge \neg p_2) \wedge \bigcirc\mathcal{A}_1^{q_2}(p_1, p_2))$，并令 $\psi = \mathcal{A}_1^{q_1}(p_1, p_2)$，$\phi = \mathcal{A}_2^q(p_2, \bigcirc p_1, p_1 \wedge \neg p_2)$，则 φ 中的自动机子公式关于 $\sim$ 被划分为两个等价类 $[\psi]_\varphi^\sim$ 和 $[\phi]_\varphi^\sim$，其中 $[\psi]_\varphi^\sim = \{\psi^{q_1}, \psi^{q_2}\}$，$[\phi]_\varphi^\sim = \{\phi^q, \phi^{q'}\}$。（注意：$\psi$ 和 ϕ 分别与 ψ^{q_1} 和 ϕ^q 相同）。 □

定义 5.2.4 (自动机公式对布尔公式的代入) 给定正自动机公式 $\varphi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A}^q = \langle \Sigma, Q, \delta, q, \Omega \rangle$。则对任意的 $\theta \in \mathbf{B}^+(Q)$，用 $\mathscr{I}_\varphi^+(\theta)$ 表示将 θ 中的每个 $q' \in Q$ 替换为 $\bigcirc\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$ 所得的公式；用 $\mathscr{I}_\varphi^-(\theta)$ 表示将 θ 中的每个 $q' \in Q$ 替换为 $\bigcirc\neg\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$ 所得的公式。 □

例 5.2.2 设 $\varphi = \mathcal{A}^{q_1}(p_1, p_1 \wedge \neg p_2)$，则 $\mathscr{I}_\varphi^+(q_1 \wedge (q_2 \vee q_3)) = \bigcirc\mathcal{A}^{q_1}(p_1, p_1 \wedge \neg p_2) \wedge (\bigcirc\mathcal{A}^{q_2}(p_1, p_1 \wedge \neg p_2) \vee \bigcirc\mathcal{A}^{q_3}(p_1, p_1 \wedge \neg p_2))$；$\mathscr{I}_\varphi^-(q_1 \vee (q_2 \wedge q_3)) = \bigcirc\neg\mathcal{A}^{q_1}(p_1, p_1 \wedge \neg p_2) \vee (\bigcirc\neg\mathcal{A}^{q_2}(p_1, p_1 \wedge \neg p_2) \wedge \bigcirc\neg\mathcal{A}^{q_3}(p_1, p_1 \wedge \neg p_2))$。 □

引理 5.1 对于自动机公式对布尔公式的代入，有如下性质成立:

- 若 $\varphi_1 \sim \varphi_2$，则 $\mathscr{I}^+_{\varphi_1}(\theta) = \mathscr{I}^+_{\varphi_2}(\theta)$，$\mathscr{I}^-_{\varphi_1}(\theta) = \mathscr{I}^-_{\varphi_2}(\theta)$。
- $\neg\mathscr{I}^+_{\varphi}(\theta) \leftrightarrow \mathscr{I}^-_{\varphi}(\overline{\theta})$。其中，$\overline{\theta}$ 是 θ 的对偶布尔公式（即：将 θ 中的 $\wedge$ 和 $\vee$ 互换，$true$ 和 $false$ 互换后得到的公式，见定义 *2.1.8*）。
- 设 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$，则对于线性结构 π 以及位置 $i \in \mathbb{N}$ 有:
  - $\pi, i \models \mathscr{I}^+_{\varphi}(\theta)$ 当且仅当存在 $Q' \subseteq Q$ 使得 $Q' \models \theta$ 并且对于任意 $q' \in Q'$ 有$\pi, i \models \bigcirc\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$。
  - $\pi, i \not\models \mathscr{I}^+_{\varphi}(\theta)$ 当且仅当存在 $Q' \subseteq Q$ 使得 $Q' \models \overline{\theta}$ 并且对于任意 $q' \in Q'$ 有$\pi, i \models \neg\bigcirc\mathcal{A}^{q'}(\varphi_1, \ldots, \varphi_n)$。

注意：上面的每个 θ 都是正布尔公式，在上下文“$Q' \models \theta$” 中，$\models$ 是布尔公式的满足关系。以上定理的证明，只需对 θ 使用公式结构归纳法即可获得。

定理 5.2 (展开定理) 设 $\varphi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \Sigma, Q, \delta, q, \Omega \rangle$，则:

- 若 φ 是 ATL_f 公式，且 q 是 $\mathcal{A}$ 中的接收状态，则 φ 等价于 $true$。
- 若 φ 是 ATL_l 公式或 ATL_r 公式，或者 φ 是 ATL_f 公式，但 q 不是 $\mathcal{A}$ 中的接收状态，则 φ 等价于 $\bigvee_{1\le k\le n}(\varphi_k \wedge \mathscr{I}^+_{\varphi}(\delta(q, a_k)))$。

证明. 当 φ 是 ATL_f 公式，且 q 是 $\mathcal{A}$ 中的接收状态时，由于 $\mathcal{A}$（即：$\mathcal{A}^q$）接收 Σ 上的空字，因此 φ 在任意的线性结构（以及任意位置）上都存在可接收运行 $\langle \{\epsilon\}, \rho \rangle$，其中 $\rho(\epsilon) = q$。于是，在这种情况下 φ 等价于 $true$。

当 φ 是 ATL_l 公式或 ATL_r 公式，或者 φ 是 ATL_f 公式，但 q 不是 $\mathcal{A}$ 中的接收状态时，对于任意的线性结构 π 以及位置 $i \in \mathbb{N}$：

- 若 $\pi, i \models \varphi$，则 φ 必然有某个在 π 上开始于位置 i 的可接收运行 $\langle T, \rho \rangle$。不妨设根节点的所有子节点为 $0, \ldots, m$，同时设 $\rho(c) = q_c$ （$0 \le c \le m$）。由定义，必然存在某个 $1 \le k \le n$，使得 $\pi, i \models \varphi_k$，并且 $\{q_0, \ldots, q_m\} \models \delta(q, a_k)$。对每个 $0 \le c \le m$，令 $\langle T_c, \rho_c \rangle$ 是 $\langle T, \rho \rangle$ 以节点 c 为根的子树，即 $T_c = \{x \in \mathbb{N}^* \mid c \cdot x \in T\}$，$\rho_c(x) = \rho(c \cdot x)$。则由定义可验证 $\langle T_c, \rho_c \rangle$ 必是 φ^{q_c} 在 π 上起始于 $i+1$ 的可接收运行，从而 $\pi, i+1 \models \varphi^{q_c}$，即 $\pi, i \models \bigcirc\mathcal{A}^{q_c}(\varphi_1, \ldots, \varphi_n)$。由引理 5.1 有 $\pi, i \models \mathscr{I}^+_{\varphi}(\delta(q, a_k))$。于是，$\pi, i \models \bigvee_{1\le k\le n}(\varphi_k \wedge \mathscr{I}^+_{\varphi}(\delta(q, a_k)))$。
- 反之，若 $\pi, i \models \bigvee_{1\le k\le n}(\varphi_k \wedge \mathscr{I}^+_{\varphi}(\delta(q, a_k)))$，则必然存在某个 $1 \le k \le n$，使得 $\pi, i \models \varphi_k$ 且 $\pi, i \models \mathscr{I}^+_{\varphi}(\delta(q, a_k))$。由引理 5.1，必然存在某个 $\{q_0, \ldots, q_m\} \subseteq Q$，使得 $\{q_0, \ldots, q_m\} \models \delta(q, a_k)$，并且对每个 $0 \le c \le m$，有 $\pi, i \models \bigcirc\mathcal{A}^{q_c}(\varphi_1, \ldots, \varphi_n)$，即 $\pi, i+1 \models \varphi^{q_c}$。于是，对每个 $0 \le c \le m$ 而言，φ^{q_c} 有在 π 上开始于位置

$i+1$ 的可接收运行 $\langle T_c, \rho_c\rangle$。这样，可以构造 Q-标记树 $\langle T, \rho\rangle$，其中 T 的根节点的第 c 个直接子树为 T_c，即：$T = \{\epsilon\} \cup \bigcup_{0\le c\le m}\{c\cdot x \mid x \in T_c\}$；并且 $\rho(\epsilon) = q$，$\rho(c\cdot x) = \rho_c(x)$。于是，$\langle T, \rho\rangle$ 是 φ 在 π 上开始于 i 的一个可接收运行，从而 $\pi, i \models \varphi$。

综上所述，在这种情况下 φ 等价于 $\bigvee_{1\le k\le n}(\varphi_k \wedge \mathscr{I}^+_\varphi(\delta(q, a_k)))$。 □

事实上，展开定理的第二种情况同样适用于其他类型的扩展时序逻辑。在第 3 章中给出的 ETL 公理系统中，曾经介绍过公理 (Expand)。该公理实际上是展开定理的一个特例：当采用的时序连接子是非确定自动机时，$\mathscr{I}^+_\varphi(\delta(q, a_k))$ 就变成了 $\bigvee_{q_j\in\delta(q,a_k)} \bigcirc\mathcal{A}^{q_j}(\varphi_1, \ldots, \varphi_n)$。同样，(Acc) 公理所刻画的正是展开定理第一种情况的特例。

在本章给出的符号化模型检验技术中，需要为待验证的逻辑公式（取非之后的）构建一个 tableau，它实际上是一个特殊的公平迁移结构。这里，首先回忆在 2.3.1 节中曾给出的若干定义。

给定公平迁移系统 $\mathcal{M} = \langle S, \Delta, I, \lambda, \mathcal{C}\rangle$，其中 $\mathcal{C} = \{C_1, \ldots, C_m\}$，则：

- $\mathcal{M}$ 的一个**展开迹** 是一个无穷状态序列 $\sigma = s_0, s_1, \ldots$，其中 $s_0 \in I$，$(s_i, s_{i+1}) \in \Delta$（注意：在定义 2.3.8 中，将其定义为一个从 $\mathbb{N}$ 到 S 的函数，二者实质等价）。进一步，如果对每个 $1 \le j \le m$，有无穷多个 $i \in \mathbb{N}$，使得 $s_i \in C_j$，则称 σ 是 $\mathcal{M}$ 中的一条**公平展开迹**。
- 称线性结构 π 是 $\mathcal{M}$ 的某个展开迹 $s_0, s_1, \ldots$ 的**派生线性结构** 使得对任意的 $i \in \mathbb{N}$ 有 $\pi(i) = \lambda(s_i)$。

然而，在实际的应用中，一个（公平）迁移系统往往只关心部分原子命题的真值—这样能够有效的压缩迁移系统的状态空间。具体而言，设某个公平迁移系统 $\mathcal{M} = \langle S, \Delta, I, \lambda, \mathcal{C}\rangle$ 所关心的原子命题集合为 CP（这里 $CP \subseteq AP$），则 λ 实际上是从 S 到 2^{CP} 的函数。因此，需要将迁移系统展开迹的“派生线性结构”的概念扩展如下。

定义 5.2.5 设公平迁移系统 $\mathcal{M} = \langle S, \Delta, I, \lambda, \mathcal{C}\rangle$ 所关心的原子命题集合为 CP, $\sigma = s_0, s_1, \ldots$ 是 $\mathcal{M}$ 中的一条展开迹。则称线性结构 π 是 σ 的**派生线性结构**，如果对于任意的 $i \in \mathbb{N}$，有 $\pi(i) \cap CP = \lambda(s_i)$。 □

以后，若某迁移系统只关心部分原子命题的真值，会给出声明。此外，仍用 $\mathbf{L}(\mathcal{M})$ 表示 $\mathcal{M}$ 中的公平展开迹对应的派生线性结构集合。

对于两个（公平）迁移系统，二者所关心的原子命题集合可能不同。因此，需要将公平迁移系统“合成” 的概念（定义 2.3.10）扩展如下。

定义 5.2.6 给定两个公平迁移系统 $\mathcal{M}_i = \langle S_i, \Delta_i, I_i, \lambda_i, \mathcal{C}_i\rangle$ 且二者关心的原子命题

集合为 CP_i ($i=1,2$)。则 $\mathcal{M}_1$ 与 $\mathcal{M}_2$ 的**合成**，记作 $\mathcal{M}_1\|\mathcal{M}_2$，是一个公平迁移系统 $\langle S,\Delta,I,\lambda,\mathcal{C}\rangle$。其中， $\mathcal{M}_1\|\mathcal{M}_2$ 所关心的原子命题集合为 $CP_1\cup CP_2$，并且:

- $\mathcal{S}=\{(s_1,s_2)\mid s_1\in S_1,s_2\in S_2,$并且对任意的 $p\in CP_1\cap CP_2,p\in\lambda_1(s_1)$ 当且仅当 $p\in\lambda_2(s_2)\}$。
- $((s_1,s_2),(s_1',s_2'))\in\Delta$ 当且仅当 $(s_1,s_1')\in\Delta_1$，$(s_2,s_2')\in\Delta_2$。
- $I=(I_1\times I_2)\cap S$。
- 对每个 $(s_1,s_2)\in S$，$\lambda((s_1,s_2))=\lambda_1(s_1)\cup\lambda_2(s_2)$。
- $\mathcal{C}=\{(C\times S_2)\cap S\mid C\in\mathcal{C}_1\}\cup\{(S_1\times C)\cap S\mid C\in\mathcal{C}_2\}$。 □

经过上述修改之后，定理 2.10 中的结论仍然成立。即：$\mathbf{L}(\mathcal{M}_1\|\mathcal{M}_2)=\mathbf{L}(\mathcal{M}_1)\cap\mathbf{L}(\mathcal{M}_2)$。而事实上，2.3.3.2 节给出的符号化模型检验算法恰好适用于这种更加一般的情况。

5.3 $\mathbf{ATL}_f$ 符号化模型检验

5.3.1 $\mathbf{ATL}_f$ 公式的 tableau

本节介绍 ATL_f 公式的符号化模型检验算法。由于 ATL_f 中使用的自动机连接子为 AFW，而 finite 接收条件是一个接收状态集。因此，若 AFW $\mathcal{A}=\langle\Sigma,Q,\delta,q,\Omega\rangle$ 且 Ω 为接收状态集 F，则在本节，直接将 $\mathcal{A}$ 写为 $\langle\Sigma,Q,\delta,q,F\rangle$。在介绍 ATL_f 的 tableau 的构造及其语言性质（language-property）之前，首先定义若干相关定义。这些定义，均是文 [110] 中相应概念的扩展。在本节，假设所有的 ATL_f 公式都已写为否定范式。

定义 5.3.1 ($\mathbf{ATL}_f$ 的基础公式集) 给定 ATL_f 公式 φ，归纳定义 φ 的**基础公式集** $\mathbf{El}(\varphi)$ 如下:

- 若 $\varphi=true$ 或者 $\varphi=false$，则 $\mathbf{El}(\varphi)=\emptyset$。
- 若 $\varphi=p$ 或者 $\varphi=\neg p$ (其中 $p\in AP$)，则 $\mathbf{El}(\varphi)=\{p\}$。
- 若 $\varphi=\varphi_1\wedge\varphi_2$，或者 $\varphi=\varphi_1\vee\varphi_2$ 则 $\mathbf{El}(\varphi)=\mathbf{El}(\varphi_1)\cup\mathbf{El}(\varphi_2)$。
- 若 $\varphi=\bigcirc\psi$，则 $\mathbf{El}(\varphi)=\{\varphi\}\cup\mathbf{El}(\psi)$。
- 若 $\varphi=\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$ 或者 $\varphi=\neg\mathcal{A}^q(\varphi_1,\ldots,\varphi_n)$，且 $\mathcal{A}$ 的状态集为 Q，则 $\mathbf{El}(\varphi)=\bigcup_{1\le k\le n}\mathbf{El}(\varphi_k)\cup\{\bigcirc\mathcal{A}^{q'}(\varphi_1,\ldots,\varphi_n)\mid q'\in Q\}$。 □

注意，$\mathbf{El}(\varphi)$ 并不一定是 $\mathbf{Sub}(\varphi)$ 的子集（这里 $\mathbf{Sub}(\varphi)$ 是 φ 的子公式集）。此外，$\mathbf{El}(\varphi)$ 中只包括原子命题和形如 $\bigcirc\psi$ 的公式。

例 5.3.1 设 $\varphi=\mathcal{A}^{q_1}(\mathcal{A}_2^{q_2'}(p_2),p_1\wedge\neg p_2)$，且 $\mathcal{A}_1$ 和$\mathcal{A}_2$ 的状态集分别为 $\{q_1,q_2\}$ 和

$\{q_1', q_2'\}$，则 $\mathbf{El}(\varphi) = \{\bigcirc\mathcal{A}_1^{q_1}(\mathcal{A}_2^{r_2}(p_2), p_1 \wedge \neg p_2), \bigcirc\mathcal{A}_1^{q_2}(\mathcal{A}_2^{r_2}(p_2), p_1 \wedge \neg p_2), \bigcirc\mathcal{A}_2^{q_1'}(p_2), \bigcirc\mathcal{A}_2^{q_2'}(p_2), p_1, p_2\}$。 □

定义 5.3.2 (公式的满足集) 对于任意的 ATL_f 公式 φ，可以定义函数 $\mathbf{Sat}_\varphi : \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi) \to 2^{2^{\mathbf{El}(\varphi)}}$ 如下。

- $\mathbf{Sat}_\varphi(true) = 2^{\mathbf{El}(\varphi)}$；$\mathbf{Sat}_\varphi(false) = \emptyset$。
- $\mathbf{Sat}_\varphi(p) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid p \in \Gamma\}$；$\mathbf{Sat}_\varphi(\neg p) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid p \notin \Gamma\}$，其中 $p \in AP$。
- $\mathbf{Sat}_\varphi(\bigcirc\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \bigcirc\psi \in \Gamma\}$。
- $\mathbf{Sat}_\varphi(\psi_1 \wedge \psi_2) = \mathbf{Sat}_\varphi(\psi_1) \cap \mathbf{Sat}_\varphi(\psi_2)$；$\mathbf{Sat}_\varphi(\psi_1 \vee \psi_2) = \mathbf{Sat}_\varphi(\psi_1) \cup \mathbf{Sat}_\varphi(\psi_2)$。
- 若 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，且 $\mathcal{A} = \langle\{a_1, \ldots, a_n\}, Q, \delta, q, F\rangle$，则当 $q \in F$ 时，$\mathbf{Sat}_\varphi(\psi) = 2^{\mathbf{El}(\varphi)}$；当 $q \notin F$ 时，$\mathbf{Sat}_\varphi(\psi) = \bigcup_{1\le k\le n}(\mathbf{Sat}_\varphi(\varphi_k) \cap \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\delta(q, a_k))))$。
- 若 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，则 $\mathbf{Sat}_\varphi(\psi) = 2^{\mathbf{El}(\varphi)} \backslash \mathbf{Sat}_\varphi(\mathcal{A}^q(\varphi_1, \ldots, \varphi_n))$。 □

关于自动机公式对正布尔公式的代入式，**Sat** 函数存在如下性质。

引理 5.3 设 $\mathcal{A} = \langle\{a_1, \ldots, a_n\}, Q, \delta, q, F\rangle$，$\psi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，$\theta \in \mathbf{B}^+(Q)$。则：

- $\Gamma \in \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\theta))$，当且仅当存在 $Q' \subseteq Q$ 使得 $Q' \models \theta$ 并且对于每个 $q' \in Q'$ 都有 $\Gamma \in \mathbf{Sat}_\varphi(\bigcirc\psi^{q'})$。
- $\Gamma \notin \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\theta))$，当且仅当存在 $Q' \subseteq Q$ 使得 $Q' \models \overline{\theta}$ 并且对于每个 $q' \in Q'$ 都有 $\Gamma \notin \mathbf{Sat}_\varphi(\bigcirc\psi^{q'})$。

同引理 5.1 一样，只需对 θ 使用公式结构归纳法即可获得上述引理的证明。

定义 5.3.3 给定 ATL_f 公式 φ，设 $\psi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 是 φ 中的自动机子公式，且 $\mathcal{A} = \langle\{a_1, \ldots, a_n\}, Q, \delta, q_0, F\rangle$。定义迁移关系 $\Delta_\psi^f \subseteq (2^{\mathbf{El}(\varphi)} \times 2^Q) \times (2^{\mathbf{El}(\varphi)} \times 2^Q)$ 如下：对于任意的 $\Gamma, \Gamma' \subseteq \mathbf{El}(\varphi)$ 以及 $P, P' \subseteq Q$，$((\Gamma, P), (\Gamma', P')) \in \Delta_\psi^f$ 当且仅当：

- 如果 $P \neq \emptyset$，那么对于任意的 $q \in Q\backslash F$ 有：存在某个 $1 \le k \le n$，使得 $\Gamma \in \mathbf{Sat}_\varphi(\varphi_k)$ 并且 $P' \models \delta(q, a_k)$。
- 如果 $P = \emptyset$，那么对于任意的 $q \in Q\backslash F$ 有：$q \in P'$ 当且仅当 $\Gamma' \in \mathbf{Sat}_\varphi(\psi^q)$。

显然，若 $\psi \sim \psi'$，则 Δ_ψ^f 与 $\Delta_{\psi'}^f$ 相同。于是，该关系的下标可以用 ψ 关于 $\sim$ 所在的等价类 $[\psi]_\varphi^\sim$ 替换。以后，也将 Δ_ψ^f 写作 $\Delta_{[\psi]_\varphi^\sim}^f$。 □

例 5.3.2 设 $\varphi = \mathcal{A}(p_1, p_2)$，其中 $\mathcal{A} = \langle\{a_1, a_2\}, \{q_1, q_2\}, \delta, q_1, \{q_2\}\rangle$，并且 $\delta(q_1, a_1) = q_1$，$\delta(q_1, a_2) = q_2$，$\delta(q_2, a_1) = \delta(q_2, a_2) = true$。显然 $\mathbf{El}(\varphi) = \{p_1, p_2, \bigcirc\varphi^{q_1}, \bigcirc\varphi^{q_2}\}$。于是，由定义有 $((\{p_1, \bigcirc\varphi^{q_1}\}, \{q_1, q_2\}), (\{p_1, p_2\}, \{q_1, q_2\})) \in \Delta_\varphi^f$。 □

定义 5.3.4 ($\mathbf{ATL}_f$ 公式的 tableau) 给定 ATL_f 公式 φ，设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]_\varphi^\sim$，…，$[\psi_m]_\varphi^\sim$，且 ψ_i 自动机连接子的状

态集为 Q_i。则可为 φ 构建一个 *tableau* $\mathcal{T}_\varphi$，它是一个特殊的（公平）迁移系统 $\langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi\rangle$。其中:

- $S_\varphi = \{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \subseteq \mathbf{El}(\varphi), P_i \subseteq Q_i\}$。
- $(\langle \Gamma, (P_1, \ldots, P_m)\rangle, \langle \Gamma', (P'_1, \ldots, P'_m)\rangle) \in \Delta_\varphi$ 当且仅当
  - 对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 有: $\Gamma \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\Gamma' \in \mathbf{Sat}_\varphi(\psi)$;
  - 对每个 $1 \leq i \leq m$ 有: $((\Gamma, P_i), (\Gamma', P'_i)) \in \Delta^f_{[\psi_i]^\sim_\varphi}$。
- $I_\varphi = \{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\varphi)\}$。
- 对每个 $\langle \Gamma, (P_1, \ldots, P_m)\rangle \in S_\varphi$, $\lambda_\varphi(\langle \Gamma, (P_1, \ldots, P_m)\rangle) = \Gamma \cap AP$。
- $\mathcal{C}_\varphi = \{C_1, \ldots, C_m\}$, 其中 $C_i = \{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid P_i = \emptyset\}$。 □

对于 ATL_f 公式 φ 而言，其 tableau $\mathcal{T}_\varphi$ 所关心的原子命题集合为 $\mathbf{El}(\varphi) \cap AP$。

例 5.3.3 设 $\varphi = \mathcal{A}(p)$, 其中 $\mathcal{A} = \langle\{a\}, \{q\}, \delta, q, \{q\}\rangle$, $\delta(q, a) = q$, 则 $\mathcal{T}_\varphi$ 的 *tableau* 如图 *5.1* 所示。由于 $\mathbf{Sat}_\varphi(\varphi) = \mathbf{El}(\varphi)$, 因此 $\mathcal{T}_\varphi$ 中的每个状态都是初始状态。此外，在 $\mathcal{T}_\varphi$ 中，$\langle\emptyset, (\{q\})\rangle$、$\langle\{p\}, (\{q\})\rangle$、$\langle\{p\}, (\emptyset)\rangle$, $\langle\emptyset, (\emptyset)\rangle$ 没有后继状态。 □

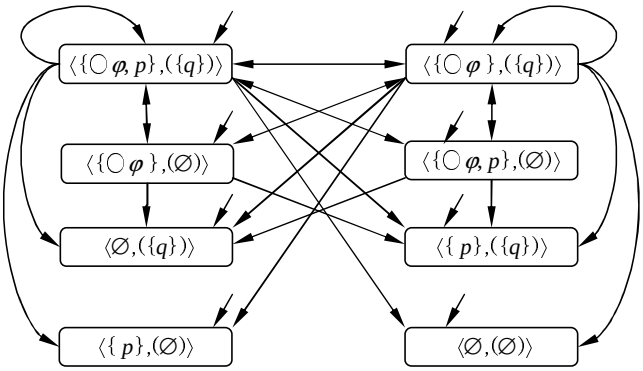


图 5.1 ATL_f tableau 示例

在一个 tableau 中，可能存在没有后继的状态，因此可以将它们删除。由于这里只关心 $\mathcal{T}_\varphi$ 的无穷（字）语言，而在任何无穷展开迹中都不会包含这些状态，因此即使将这些状态保留，也不会对 $\mathbf{L}(\mathcal{T}_\varphi)$ 产生实质影响。

下面通过两个定理（定理 5.4 和定理 5.5）来说明 ATL_f 公式的“语言性质”。即：对于任意的线性结构 π 而言，$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

定理 5.4 对于任意 *ATL*$_f$ 公式以及线性结构 π, 若 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$, 则 $\pi \models \varphi$。

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]^\sim_\varphi, \ldots, [\psi_m]^\sim_\varphi$。同时，设 ψ_j 的自动机连接子 $\mathcal{A}^{q_j}_j = \langle\Sigma_j, Q_j, \delta_j, q_j, F_j\rangle$，以及 $\psi_j = \mathcal{A}^{q_j}_j(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$，其中，$\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$。

设 π 是 $\mathcal{T}_\varphi$ 中公平展开迹 $s_0, s_1, \ldots$ 所对应的派生线性结构，其中 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i}) \rangle$。下面用结构归纳法证明：对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$ 以及任意的 $i \in \mathbb{N}$ 都有

1. 若 $\Gamma_i \in \mathbf{Sat}(\psi)$，则 $\pi, i \models \psi$；
2. 若 $\Gamma_i \notin \mathbf{Sat}(\psi)$，则 $\pi, i \not\models \psi$。

- 基本情形证明过程较为简单：
  - 当 $\psi = true$ 或者 $\psi = false$ 时结论显然。
  - 当 $\psi = p$（resp. $\psi = \neg p$）时，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $p \in \Gamma_i$（resp. $p \notin \Gamma_i$），当且仅当 $p \in \pi(i)$（$p \notin \pi(i)$）当且仅当 $\pi, i \models p$（resp. $\pi, i \models \neg p$）。于是，当 $\psi = p$ 或 $\psi = \neg p$ 时，结论 1、2 都成立。
  - 当 $\psi = \psi' \wedge \psi''$ 时，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi') \cap \mathbf{Sat}_\varphi(\psi'')$。由归纳假设，当且仅当 $\pi, i \models \psi'$ 且 $\pi, i \models \psi''$，当且仅当 $\pi, i \models \psi$。于是，这种情况下结论 1、2 都成立。
  - 当 $\psi = \psi' \vee \psi''$ 时，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi') \cup \mathbf{Sat}_\varphi(\psi'')$。由归纳假设，当且仅当 $\pi, i \models \psi'$ 或 $\pi, i \models \psi''$，当且仅当 $\pi, i \models \psi$。于是，这种情况下结论 1、2 都成立。
  - 当 $\psi = \bigcirc \psi'$ 时，则 $\psi \in \mathbf{El}(\varphi)$。于是，由 Δ_φ 的定义，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi')$。由归纳假设，这当且仅当 $\pi, i+1 \models \psi'$，当且仅当 $\pi, i \models \psi$。于是，这种情况下结论 1、2 都成立。
- 下面证明当 ψ 是正自动机公式时结论 1 成立。

不妨设 $\psi = \psi_j^{q_0}$（这里，$1 \leq j \leq m$，并且 $q_0 \in Q_0$）。如果 $q_0 \in F_j$，则由展开定理知 ψ 等价于 $true$，显然 $\pi, i \models \psi$ 成立。以下考虑 $q_0 \notin F$ 的情况。这时，要证明 $\pi, i \models \psi$，只需构造一棵 ψ 在 π 上开始于位置 i 的可接收运行 $\langle T, \rho \rangle$ 即可。由于 $s_0, s_1, \ldots$ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹，于是由公平限制 C_j 知，必定存在 i_1、i_2 使得 $i < i_1 < i_2$ 且 $P_{j,i_1} = P_{j,i_2} = \emptyset$。同时，对于任意的 $i_1 < l < i_2$，$P_{j,l} \neq \emptyset$。于是，$\langle T, \rho \rangle$ 之构造可分为两个阶段进行。

第一阶段：构造 $\langle T, \rho \rangle$ 的第 0 至 $i_1 - i + 1$ 层。

- 为 T 添加根节点 ϵ，并令 $\rho(\epsilon) = q_0$。由已知条件，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$。换言之，有 $\Gamma_{i+|\epsilon|} \in \mathbf{Sat}_\varphi(\psi^{\rho(\epsilon)})$ 成立。
- 对于 T 中每个新添加的节点 x（其中 $|x| \leq i_1 - i$），归纳假设 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\psi^{\rho(x)})$

成立。若 $\rho(x) \in F_j$，则令 x 为 T 的一个叶节点。否则，由定义有

$$\Gamma_{i+|x|} \in \bigcup_{1 \le k \le \#\Sigma_j} (\mathbf{Sat}_\varphi(\varphi_{j,k}) \cap \mathbf{Sat}_\varphi(\mathscr{I}^+_\psi(\delta_j(\rho(x), a_{j,k})))) \tag{5.1}$$

于是，必然存在某个 $1 \le k \le \#\Sigma_j$，使得 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\mathscr{I}^+_\psi(\delta_j(\rho(x), a_{j,k})))$。由归纳假设有 $\pi, i+|x| \models \varphi_{j,k}$。同时，由引理 5.3 知必然存在某个 $Q_{j,x} \subseteq Q_j$ 使得 $Q_{j,x} \models \delta_j(\rho(x), a_{j,k})$，并且对于任意的 $q_{x,c} \in Q_{j,x}$，有 $\Gamma_{i+|x|} \in \mathbf{Sat}(\bigcirc\psi^q_{x,c})$ 成立。不妨设 $Q_{j,x} = \{q_{x,0}, \ldots, q_{x,t}\}$，于是对每个 $0 \le c \le t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。注意，当 $\delta_j(\rho(x), a_{j_k}) = true$ 时，$Q_{j,x}$ 可以为 $\emptyset$，这时 x 就成为 T 中的一个接收叶节点。对于每个新加入的节点 $x \cdot c$，由于 $|x \cdot c| = |x| + 1$，所以由迁移关系 Δ_φ 知：$\Gamma_{i+|x|+1} \in \mathbf{Sat}_\varphi(\psi^{q_{x,c}})$，换言之，$\Gamma_{i+|x \cdot c|} \in \mathbf{Sat}_\varphi(\psi^{\rho(x \cdot c)})$。

若在该过程中得到的 $\langle T, \rho \rangle$ 中的每个节点都已处理完毕（即：对每个 $x \in T$，若 $\rho(x) \notin F_j$，则已按上述过程为 x 添加子节点），并且每个叶节点 x 或者是接收叶节点或者 $\rho(x) \in F_j$，则显然得到的有穷树是 ψ 在 π 上起始于位置 i 的一个可接收运行。否则，只可能是 T 中深度为 $i_1 - i + 1$ 的节点尚未处理完毕，转入下一阶段构造。

第二阶段：构造 $\langle T, \rho \rangle$ 的剩余层。

- 对于每个 $x \in T$，若 $|x| = i_1 - i + 1$，则 x 是在第一阶段内创建的节点，于是有 $\Gamma_{i+|x|} = \Gamma_{i_1+1} \in \mathbf{Sat}_\varphi(\psi^{\rho(x)})$。注意到 $\Gamma_{i_1} = \emptyset$，而由迁移关系 Δ_φ 知 $((\Gamma_{i_1}, P_{j,i_1}), (\Gamma_{i_1+1}, P_{j,i_1+1})) \in \Delta^f_{[\psi_j]^\sim_\varphi}$，所以 $\rho(x) \in P_{j,i_1+1}$。于是，$\{\rho(x) \mid x \in T, |x| = i_1 - i + 1\} \subseteq P_{j,i_1+1}$。
- 对每个 $l \in \{i_2 - i, \ldots, i_1 - i\}$，归纳假设 $\{\rho(x) \mid x \in T, |x| = l\} \subseteq P_{j,i+l}$ 成立。对每个节点 $x \in T$，若 $|x| = l$ 且 x 尚未处理完毕，则按照如下方式对其进行处理：如果 $\rho(x) \in F_j$，那么令 x 为 T 中的叶节点。如果 $\rho(x) \notin F_j$，则由 $((\Gamma_{j,i+l}, P_{j,i+l}), (\Gamma_{j,i+l+1}, P_{j,i+l+1})) \in \Delta^f_{[\psi_j]^\sim_\varphi}$ 并且 $P_{j,i+l} \neq \emptyset$ 知，存在某个 $1 \le k \le \#\Sigma_j$ 使得 $\Gamma_{j,i+l} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $P_{j,i+l+1} \models \delta_j(\rho(x), a_{j,k})$。由归纳假设，$\pi, i+l \models \varphi_{j,k}$。同时，由于 $\delta_j(\rho(x), a_{j,k}) \in \mathbf{B}^+(Q_j)$，所以存在 P_{i+l+l} 的某个子集 $Q_{j,x}$ 满足 $\delta_j(\rho(x), a_{j,k})$。不妨设 $Q_{j,x} = \{q_{x,0}, \ldots, q_{x,t}\}$，于是对每个 $0 \le c \le t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。特别的，当 $\delta_j(\rho(x), a_{j,k}) = true$ 时，x 可以成为 T 的接收叶节点。因此，对于 T 中每个深度为 $l+1$ 的节点 $x \cdot c$ 都有 $\rho(x \cdot c) \in P_{i+l+1}$。于是，$\{\rho(x) \mid x \in T, |x| = l + 1\} \subseteq P_{j,i+l+1}$ 成立。

注意到 $P_{j,i_2} = \emptyset$，因此 T 一定是一棵有穷树。进一步，对于 T 的每个叶节点 x，由

构造过程知：x 或者是 $\langle T,\rho\rangle$ 中的一个接收叶节点，或者 $\rho(x)\in F_j$。所以 $\langle T,\rho\rangle$ 是 ψ 在 π 上开始于 i 的一个可接收运行。由定义 $\pi,i\models\psi$ 成立。

- 下面证明当 ψ 是负自动机公式时结论 1 成立。

不妨设 $\psi=\neg\phi$，其中 $\phi=\psi_j^{q_0}$，$1\le j\le m$，$q_0\in Q_j$。反设 $\pi,i\not\models\psi$，于是 $\pi,i\models\phi$。由定义，存在 ϕ 在 π 上起始于 i 的可接收运行 $\langle T,\rho\rangle$。下面自底向上证明："对于任意的 $x\in T$，有 $\Gamma_{i+|x|}\in\mathbf{Sat}_\varphi(\phi^{\rho(x)})$ 成立"。

当 x 是 T 中的叶节点时，分两种情况证明：

- 若 $\rho(x)\in F_j$，则 $\mathbf{Sat}_\varphi(\phi^{\rho(x)})=2^{\mathbf{El}(\varphi)}$，显然结论成立。
- 若 x 是 T 中的接收节点，则存在某个 $1\le k\le\#\Sigma_j$ 使得 $\pi,i+|x|\models\varphi_{j,k}$ 并且 $\delta_j(\rho(x),a_{j,k})=true$。由归纳假设知 $\Gamma_{i+|x|}\in\mathbf{Sat}_\varphi(\varphi_{j,k})$，同时 $\mathscr{I}_\phi^+(\delta_j(\rho(x),a_{j,k}))=true$。于是，$\Gamma_{i+|x|}\in(\mathbf{Sat}_\varphi(\varphi_{j,k})\cap\mathbf{Sat}_\varphi(\mathscr{I}_\phi^+(\delta_j(\rho(x),a_{j,k}))))$，进而有

$$\Gamma_{i+|x|}\in\bigcup_{1\le k\le\#\Sigma_j}(\mathbf{Sat}_\varphi(\varphi_{j,k})\cap\mathbf{Sat}_\varphi(\mathscr{I}_\phi^+(\delta_j(\rho(x),a_{j,k}))))=\mathbf{Sat}_\varphi(\phi^{\rho(x)})\quad(5.2)$$

成立。

再证明 x 是非叶节点时的情况。

- 若 $\rho(x)\in F_j$，则 $\mathbf{Sat}_\varphi(\phi^{\rho(x)})=2^{\mathbf{El}(\varphi)}$，显然结论成立。
- 否则，归纳假设 x 的每个子节点 $x\cdot c$ 都有 $\Gamma_{i+|x\cdot c|}=\Gamma_{i+|x|+1}\in\mathbf{Sat}_\varphi(\phi^{\rho(x\cdot c)})$ 成立。由于 $\langle T,\rho\rangle$ 是 ϕ 在 π 上起始于位置 i 的可接收运行，于是必然存在某个 $1\le k\le\#\Sigma_j$ 使得 $\pi,i+|x|\models\varphi_{j,k}$ 并且 $\{\rho(x\cdot c)\mid x\cdot c\in T\}\models\delta_j(\rho(x),a_{j,k})$。由（公式结构）归纳假设，$\Gamma_{i+|x|}\in\mathbf{Sat}_\varphi(\varphi_{j,k})$。同时，存在 x 的某个子节点集合 $\{x\cdot c_0,\ldots,x\cdot c_m\}\models\delta_j(\rho(x),a_{j,k})$。由于对每个 $x\cdot c_l$ 都有 $\Gamma_{i+|x|+1}\in\mathbf{Sat}(\phi^{\rho(x\cdot c_l)})$，因此由迁移关系 Δ_φ 得 $\Gamma_{i+|x|}\in\mathbf{Sat}(\bigcirc\phi^{\rho(x\cdot c_l)})$。由引理 5.3 得 $\Gamma_{i+|x|}\in\mathbf{Sat}_\varphi(\mathscr{I}_\phi^+(\delta_j(\rho(x),a_{j,k})))$。于是有

$$\Gamma_{i+|x|}\in\bigcup_{1\le k\le\#\Sigma_j}(\mathbf{Sat}_\varphi(\varphi_{j,k})\cap\mathbf{Sat}_\varphi(\mathscr{I}_\phi^+(\delta_j(\rho(x),a_{j,k}))))=\mathbf{Sat}_\varphi(\phi^{\rho(x)})\quad(5.3)$$

成立。

显然，上述归纳对于 T 中的根节点 ϵ 仍然成立。于是有 $\Gamma_i=\Gamma_{i+|\epsilon|}\in\mathbf{Sat}_\varphi(\phi^{\rho(\epsilon)})=\mathbf{Sat}_\varphi(\phi)$ 成立（因为 $\langle T,\rho\rangle$ 是 ϕ 在 ψ 上起始于 i 的可接收运行，所以 $\rho(\epsilon)=q_0$，而 $\phi^{q_0}=\phi$）。

考虑到 $\psi=\neg\phi$，而 ϕ 是自动机公式，因而 $\mathbf{Sat}_\varphi(\psi)=2^{\mathbf{El}(\varphi)}\backslash\mathbf{Sat}_\varphi(\phi)$。于是 $\Gamma_i\in\mathbf{Sat}_\varphi(\phi)$。这与前提条件 $\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 矛盾。所以，假设 $\pi,i\models\phi$ 是不成立的，于是有 $\pi,i\models\psi$。

- 下面说明当 ψ 是正/负自动机公式时，结论 2 成立。
  - 当 ψ 是正自动机公式且 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi)$ 时，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\neg\psi)$。由前面的证明，$\pi, i \models \neg\psi$，所以 $\pi, i \not\models \psi$。
  - 当 ψ 是负自动机公式且 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi)$ 时，不放设 $\psi = \neg\phi$，于是 ϕ 是正自动机公式并且 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi)$。由前面的证明，$\pi, i \models \phi$，因此 $\pi, i \not\models \psi$。

显然，上述归纳是完全的。由于 $s_0 \in I_\varphi$，所以 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$。由所证结论，有 $\pi, 0 \models \varphi$，即：$\pi \models \varphi$ 成立。 □

定理 5.5 对于任意的 ATL_f 公式 φ 以及线性结构 π，若 $\pi \models \varphi$ 则 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]_\varphi^\sim, \ldots, [\psi_m]_\varphi^\sim$。同时，设 ψ_j 的自动机连接子$\mathcal{A}^{q_j} = \langle \Sigma_j, Q_j, \delta_j, q_j, F_j \rangle$，以及 $\psi_j = \mathcal{A}_j^{q_j}(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$，其中，$\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$。

对任意的性结构 π，若 $\pi \models \varphi$ 成立，则采用如下方案证明 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$：

1. 首先，根据 π 构造 $\mathcal{T}_\varphi$ 中的无穷状态序列 $\sigma = s_0, s_1, \ldots$。其中 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i}) \rangle$，并且该构造保证对每个 $1 \le j \le m$，有无穷多个 i 使得 $P_{j,i} = \emptyset$。
2. 先证明 $s_0 \in I_\varphi$，即：$\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$；其次证明 $(s_i, s_{i+1}) \in \Delta_\varphi$。于是 σ 是 $\mathcal{T}_\varphi$ 中的一条展开迹，并且是公平展开迹。
3. 证明 π 是 σ 的派生线性结构，从而 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

- 首先给出状态序列 σ 的构造过程。

对于每个 $i \in \mathbb{N}$，令 $\Gamma_i = \{\psi \in \mathbf{El}(\varphi) \mid \pi, i \models \psi\}$。

对于每个 $1 \le j \le m$ 以及每个 $i \in \mathbb{N}$，集合 $P_{j,i}$ 按照如下方式确定。

首先，令 $C_0^j = 0$，并且 $P_{j,C_0^j} = \emptyset$。其次，对于每个 $k \in \mathbb{N}$，归纳假设 $P_{j,C_k^j} = \emptyset$，则使用下列步骤确定 C_{k+1}^j 的值，同时对每个 $C_k^j < l \le C_{k+1}^j$ 构建 $P_{j,l}$：

- 令 $P_{j,C_k^j+1} = \{q \in Q_j \mid \pi, C_k^j + 1 \models \psi_j^q\}$。
- 于是，对每个 $q \in P_{j,C_k^j+1}$，存在 ψ_j^q 在 π 上起始于位置 $C_k^j + 1$ 的某个可接收运行 $\langle T_{q,k}, \rho_{q,k} \rangle$。注意到每个 $T_{q,k}$ 一定是有穷树，于是 $|T_{q,k}| < +\infty$。令

$$C_{k+1}^j = C_k^j + 1 + \max\{|T_{q,k}| \mid q \in P_{j,C_k^j+1}\}。$$

显然，若 $P_{j,C_k^j+1} = \emptyset$，则有 $C_{k+1}^j = C_k^j + 1$ 成立。

- 对于每个 $C_k^j < l \le C_{k+1}^j$，令

$$P_{j,l} = \bigcup_{q \in P_{j,C_k^j}} \{\rho_{q,k}(x) \mid x \in T_{q,k}, \text{并且 } |x| = l - C_k^j - 1\}。$$

- 容易验证 $P_{j,C_{k+1}^j} = \emptyset$。

- 现在证明 σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

  首先证明：对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\pi, i \models \psi$。

  - 当 $\psi = true$ 或者 $\psi = false$ 时结论显然。
  - 对于任意 $p \in AP$，当 $\psi = p$ 时由于 $p \in \mathbf{El}(\varphi)$，则由 Γ_i 的构造保证 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $p \in \Gamma_i$ 当且仅当 $\pi, i \models \psi$。
  - 由上述结论，当 $\psi = \neg p$ 时，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \notin \mathbf{Sat}_\varphi(p)$，这当且仅当 $\pi, i \not\models p$，当且仅当 $\pi, i \models \psi$。
  - 若 $\psi = \bigcirc\psi'$，由于 $\psi \in \mathbf{El}(\varphi)$，所以由 Γ_i 的构造保证 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$。而这当且仅当 $\psi \in \Gamma_i$，当且仅当 $\pi, i \models \psi$。
  - 若 $\psi = \psi_1 \wedge \psi_2$（resp. $\psi = \psi_1 \vee \psi_2$），则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi_1)$ 且（resp. 或）$\Gamma_i \in \mathbf{Sat}_\varphi(\psi_2)$。由归纳假设，当且仅当 $\pi, i \models \psi_1$ 且（resp. 或）$\pi, i \models \psi_2$，当且仅当 $\pi, i \models \psi$。
  - 若 ψ 是正自动机公式，不妨设 $\psi = \psi_0^q$（其中 $q_0 \in Q_j$）。如果 $q_0 \in F_j$ 则相当于 $\psi = true$ 的情形，下面假设 $q_0 \notin F_j$。在这种情况下，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \bigcup_{1 \le k \le \#\Sigma_j} (\mathbf{Sat}_\varphi(\varphi_{j,k}) \cap \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+ \delta_j(q_0, a_{j,k})))$，当且仅当存在某个 $1 \le k \le \#\Sigma_j$，使得 $\Gamma_i \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $\Gamma_i \in \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\delta_j(q_0, a_{j,k})))$。于是：

    1. $\pi, i \models \varphi_{j,k}$ ⟅$\Gamma_i \in \mathbf{Sat}_\varphi(\varphi_{j_k})$ 以及归纳假设⟆
    2. 存在某个 $Q' \subseteq Q_j$ 使得 $Q' \models \delta_j(q_0, a_{j,k})$ 并且对于任意的 $q' \in Q'$，$\Gamma_i \in \mathbf{Sat}_\varphi(\bigcirc\psi_j^{q'})$。 ⟅引理 5.3⟆
    3. 对每个 $q' \in Q'$，$\pi, i \models \bigcirc\psi_j^{q'}$ ⟅$\bigcirc\psi_j^{q'} \in \mathbf{El}(\varphi)$，该情形已证⟆
    4. $\pi, i \models \mathscr{I}_\psi^+(\delta_j(q, a_{j,k}))$ ⟅2、3 以及引理 5.1⟆

    于是，当且仅当存在 $1 \le k \le \#\Sigma_j$ 使得 $\pi, i \models \varphi_{j,k} \wedge \mathscr{I}_\psi^+(\delta_j(q_0, a_{j,k}))$，当且仅当 $\pi, i \models \bigvee_{1 \le k \le \#\Sigma_j} (\varphi_{j,k} \wedge \mathscr{I}_\psi^+(\delta_j(q_0, a_{j,k})))$，（由展开定理）当且仅当 $\pi, i \models \psi$。
  - 若 ψ 是负自动机公式，不妨设 $\psi = \neg\phi$，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\phi \notin \mathbf{Sat}_\varphi(\phi)$，（由归纳）当且仅当 $\pi, i \not\models \phi$，当且仅当 $\pi, i \models \psi$。

  接下来证明：对于每个 $i \in \mathbb{N}$，都有 $(s_i, s_{i+1}) \in \Delta_\varphi$。

  一方面，对于每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 以及每个 $i \in \mathbb{N}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\pi, i \models \bigcirc\psi$，当且仅当 $\pi, i+1 \models \psi$，当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi)$。因此，接下来只需证明对每个 $1 \le j \le m$ 以及 $i \in \mathbb{N}$ 都有 $((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta^f_{[\psi_j]_\varphi^\sim}$ 即可。

  - 若 $P_{j,i} \neq \emptyset$，不妨设 $C_k^j < i < C_{k+1}^j$。对于每个 $q \in P_{j,i} \backslash F_j$，由 σ 的构造知，必然存在某个 $q' \in Q_j$ 以及 $\psi_j^{q'}$ 在 π 上开始于 $C_k^j + 1$ 的可接收运行 $\langle T_{q',k}, \rho(q', k) \rangle$，同时存在某个 $x \in T_{q',k}$ 使得 $|x| = i - C_k^j - 1$，且 $\rho_{q',k}(x) = q$。

由于 $\langle T_{q',k}, \rho(q',k)\rangle$ 是 $\psi_j^{q'}$ 在 π 上开始于 C_k^j+1 的一个可接收运行，因此存在某个 $1 \le l < \#\Sigma_j$ 使得 $\pi, |x| + C_k^j + 1 \models \varphi_{j,l}$（即 $\pi, i \models \varphi_{j,l}$）以及 $Q_{j,x} = \{\rho_{q',k}(x \cdot c) \mid c \in \mathbb{N}, x \cdot c \in T_{q',k}\} \models \delta_j(q, a_{j,l})$。

由前面结论，$\Gamma_i \in \mathbf{Sat}_\varphi(\varphi_{j,l})$。同时，由 σ 的构造知 $Q_{j,x} \subseteq P_{j,i+1}$。由正布尔公式的单调性知 $P_{j,i+1} \models \delta_j(q, a_{j,l})$。

- 若 $P_{j,i} = \emptyset$，则由 σ 的构造知，必然存在某个 $k \in \mathbb{N}$ 使得 $i = C_k^j$。于是，$P_{j,i+1} = P_{j,C_k^j+1} = \{q \in Q_j \mid \pi, i+1 \models \psi_j^q\}$。由刚证得的结论，立即有 $P_{j,i+1} = \{q \in Q_j \mid \Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi_j^q)\}$。

综上所述，对于任意 $i \in \mathbb{N}$，都有 $(s_i, s_{i+1}) \in \Delta_\varphi$ 成立。

现在说明 σ 是 $\mathcal{T}_\varphi$ 中的公平展开迹。

1. 由于 $\pi \models \varphi$（即 $\pi, 0 \models \varphi$），所以有 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$。于是，$s_0 \in I_\varphi$。
2. 对于任意的 $i \in \mathbb{N}$，已经证明 $(s_i, s_{i+1}) \in \Delta_\varphi$。
3. 由 σ 的构造，对于每个 $1 \le j \le m$，都有无穷多个 $i \in \mathbb{N}$，使得 $P_{j,i} = \emptyset$。事实上，对每个 $k \in \mathbb{N}$，$P_{j,C_k^j} = \emptyset$。

因此，按照定义 σ 确为 $\mathcal{T}_\varphi$ 中的公平展开迹。

- 最后说明 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

事实上，由于已经证明 σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹，只需证明 π 是 σ 在 $\mathcal{T}_\varphi$ 中的派生线性结构即可。换言之，只需证明对任意的 $i \in \mathbb{N}$ 有$\pi(i) \cap (AP \cap \mathbf{El}(\varphi)) = \lambda_\varphi(s_i)$ 即 $\pi(i) \cap \mathbf{El}(\varphi) = \Gamma_i \cap AP$ 即可。

由 σ 的构造：$\psi \in \Gamma_i \cap AP$ 当且仅当 $\psi \in \mathbf{El}(\varphi)$ 且 $\pi, i \models \psi$，当且仅当 $\psi \in \pi(i) \cap \mathbf{El}(\varphi)$。于是，对于每个 $i \in \mathbb{N}$ 有 $\pi(i) \cap \mathbf{El}(\varphi) = \Gamma_i \cap AP$ 成立。 □

由定理 5.4 及定理 5.5，立即可以得到如下的推论。

推论 5.6 ($\mathbf{ATL}_f$ 公式 tableau 的语言性质) *对于任意 ATL_f 公式以及线性结构 π 有：$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。*

下面的定理将 ATL_f 公式的模型检验问题转化为 CTL 公式的模型检验问题。

定理 5.7 *对于任意的公平迁移系统 $\mathcal{M}$ 以及 ATL_f 公式 φ 有：$\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。*

证明. 一方面，由推论 5.6，对于任意的线性结构 π 有：$\pi \in \mathbf{L}(\mathcal{T}_{\neg\varphi})$ 当且仅当 $\pi \models \neg\varphi$，即 $\pi \not\models \varphi$。另一方面，由定义 2.3.11，$\mathcal{M} \models \varphi$ 当且仅当对于任意的 $\pi \in \mathbf{L}(\mathcal{M})$，都有 $\pi \models \varphi$ 成立。于是，必然有 $\mathbf{L}(\mathcal{M}) \cap \mathbf{L}(\mathcal{T}_{\neg\varphi}) = \emptyset$，即：$\mathbf{L}(\mathcal{M} \| \mathcal{T}_{\neg\varphi}) = \emptyset$（见定理 2.10）。这当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi}$ 中不存在公平展开迹，即 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。 □

5.3.2 基于 BDD 的 ATL_f tableau 编码

在 2.3 节，曾经介绍过 CTL 模型检验算法基于 BDD 的符号化实现方法。5.3.1 中的定理 5.7 将 ATL_f 的模型检验问题转化成为了 CTL 的模型检验问题。

对于 ATL_f 公式 φ，由于 $\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\, true$。因此，在执行 CTL 模型检验之前，需要构建 $\mathcal{M} \| \mathcal{T}_{\neg\varphi}$ 的 BDD 表示。由于模型 $\mathcal{M}$ 的符号化表示往往可以直接从输入获得（类似于 SMV[40]、NuSMV[111] 中的做法）；同时，2.3.3.2 节介绍了如何从两个（公平）迁移系统的符号化表示得到其合成的符号化表示的方法。因此，这里介绍如何如何获得任意 ATL_f 公式 φ 的 tableau 的符号化表示的方法（对于 $\mathcal{T}_{\neg\varphi}$，同样只需对 $\neg\varphi$ 的否定范式使用此过程即可）。

假设 φ 已经写为否定范式，并且 $[\psi_1]_{\varphi}^{\sim}$，…，$[\psi_m]_{\varphi}^{\sim}$ 是其所有自动机子公式由关系 $\sim$ 所划分的等价类，其中 ψ_j 的自动机连接子为 $\mathcal{A}_j = \langle \Sigma_j, Q_j, \delta_j, q_j, F_j \rangle$，并且 $\psi_j = \mathcal{A}_j(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$。于是，$\mathcal{T}_\varphi = \langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi \rangle$ 的各要素编码过程如下。

位变元集合：对每个 $\psi \in \mathbf{El}(\varphi)$，引入一个位变元 z_ψ；同时，对每个 $1 \le j \le m$ 以及 $q \in Q_j$ 引入一个位变元 $u_{j,q}$。

状态约束：由于待检验的 CTL 性质为 $\mathsf{EG}\, true$，因此可以不必删除 $\mathcal{T}_\varphi$ 中无后继的状态。故而可以认为每个状态都是合法的。于是，这里令 $\Phi_{S_\varphi} = true$。

迁移关系：迁移关系 Φ_{Δ_φ} 按照如下方式获得。

首先，对于每个 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，归纳构建 $\mathbf{Sat}_\varphi(\psi)$ 之布尔表示 ϑ_ψ 如下：

- $\vartheta_{true} = true$，$\vartheta_{false} = false$。
- 对于每个 $p \in AP$，若 $\psi = p$，则 $\vartheta_\psi = z_p$；若 $\psi = \neg p$，则 $\vartheta_\psi = \neg z_p$。
- 若 $\psi = \bigcirc\psi'$，则令 $\vartheta_\psi = z_\psi$。（因为 $\psi \in \mathbf{El}(\varphi)$，所以位变元 z_ψ 在编码时被引入。）
- 若 $\psi = \psi_1 \wedge \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \wedge \vartheta_{\psi_2}$；若 $\psi = \psi_1 \vee \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \vee \vartheta_{\psi_2}$。
- 若 $\psi = \psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$）：当 $q \in F_j$ 时，令 $\vartheta_\psi = true$；否则，令 $\vartheta_\psi = \bigvee_{1 \le k \le \#\Sigma_j} (\vartheta_{\varphi_{j,k}} \wedge \vartheta_{\mathscr{I}_\psi^+(\delta_j(q, a_{j,k}))})$。
- 若 $\psi = \neg\psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$）：当 $q \in F_j$，则令 $\vartheta_\psi = false$；否则，令 $\vartheta_\psi = \neg\vartheta_{\psi_j^q}$。

由于 $\mathscr{I}_\psi^+(\delta_j(q, a_{j,k}))$ 是一些形如 $\bigcirc\psi$ 的公式的布尔组合，因此该递归计算过程必然终止。容易用归纳法证明，布尔公式 ϑ_ψ 有如下性质：对于任意的 $\Gamma \subseteq \mathbf{El}(\varphi)$，$\Gamma \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 ϑ_ψ 在变元指派 e_Γ 下真值为 1。其中 e_Γ 满足：对于每个 $\phi \in \mathbf{El}(\varphi)$，$e_\Gamma(z_\phi) = 1$ 当且仅当 $\phi \in \Gamma$。

接下来，令 ϑ'_ψ 是将 ϑ_ψ 中的每个位变元换成其次态版本（即：将每个 z_ϕ 替换为 z'_ϕ，将每个 $u_{j,q}$ 替换为 $u'_{j,q}$）所得之布尔公式。同时，对每个 $\theta \in \mathbf{B}^+(Q_j)$，令 $\mathbf{U}_j(\theta)$ 为将 θ 中的每个 q 替换为 $u_{j,q}$ 所得之布尔公式；令 $\mathbf{U}'_j(\theta)$ 为将 θ 中的每个 q 替换为 $u'_{j,q}$ 所得之布尔公式。于是，迁移关系 Φ_{Δ_φ} 就是下列各项之合取。

$$\bigwedge_{\bigcirc\psi\in\mathbf{El}(\varphi)} z_{\bigcirc\psi} \leftrightarrow \vartheta'_\psi \tag{5.4}$$

$$\bigwedge_{1\le j\le m}((\bigvee_{q\in Q_j} u_{j,q}) \rightarrow \bigwedge_{q\in Q_j\setminus F_j}(u_{j,q} \rightarrow \bigvee_{1\le k\le \#\Sigma_j}(\vartheta_{\varphi_{j,k}} \wedge \mathbf{U}'_j(\delta_j(q,a_{j,k})))))) \tag{5.5}$$

$$\bigwedge_{1\le j\le m}((\bigwedge_{q\in Q_j} \neg u_{j,q}) \rightarrow \bigwedge_{q\in Q_j}(u'_{j,q} \leftrightarrow \vartheta'_{\psi_j^q})) \tag{5.6}$$

对于任意的 $s = \langle \Gamma, (P_1, \ldots, P_m)\rangle \in S_\varphi$ 以及任意的 $s' = \langle \Gamma', (P'_1, \ldots, P'_m)\rangle \in S_\varphi$，令 e_s、$e_{s'}$ 是满足如下约束的两个变元指派。

- 对于每个 $\phi \in \mathbf{El}(\varphi)$：$e_s(z_\phi) = 1$ 当且仅当 $\phi \in \Gamma$；$e_{s'}(z'_\phi) = 1$ 当且仅当 $\phi \in \Gamma'$。
- 对于每个 $1 \le j \le m$ 以及每个 $q \in Q_j$：$e_s(u_{j,q}) = 1$ 当且仅当 $q \in P_j$；$e_{s'}(u'_{j,q}) = 1$ 当且仅当 $q \in P'_j$。

于是，e_s 和 $e_{s'}$ 就构成了一个**联合指派**。即：对于每个 $\psi \in \mathbf{El}(\varphi)$，$z_\psi$ 和 z'_ψ 的真值分别由 e_s 和 $e_{s'}$ 给出；对于每个 $q \in Q_j$，$u_{j,q}$ 和 $u'_{j,q}$ 的真值分别由 e_s 和 $e_{s'}$ 给出。这时，很容易验证下列性质：

1. 公式 (5.4) 在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1 当且仅当对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 有：$\Gamma \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\Gamma' \in \mathbf{Sat}_\varphi(\psi)$。
2. 公式 (5.5) 与公式 (5.6) 的合取在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1 当且仅当对每个 $1 \le j \le m$ 有 $((\Gamma, P_j), (\Gamma', P'_j)) \in \Delta^f_{[\psi_j]_{\widetilde{\varphi}}}$。

于是，这样获得的 Φ_{Δ_φ} 是关于 Δ_φ 一个符合定义的布尔编码。

初始状态集：由于在 $\mathcal{T}_\varphi$ 的定义中，$s = \langle \Gamma, (P_1, \ldots, P_m)\rangle \in I_\varphi$ 当且仅当 $\Gamma \in \mathbf{Sat}_\varphi(\varphi)$。因此，可令初始状态集的编码 $\Phi_{I_\varphi} = \vartheta_\varphi$。

标记函数：由于在 $\mathcal{T}_\varphi$ 的定义中，$\lambda_\varphi(\langle \Gamma, (P_1, \ldots, P_m)\rangle) = \Gamma \cap AP$。于是，对每个 $p \in AP \cap \mathbf{El}(\varphi)$，令 $\Phi^p_{\lambda_\varphi} = z_p$ 即可。

公平性约束：由于 $\mathcal{C}_\varphi = \{C_1, \ldots, C_m\}$，其中 $C_j = \{\langle \Gamma, (P_1, \ldots, P_m) \mid P_j = \emptyset\rangle \mid P_j = \emptyset\}$。于是，$\Phi_{\mathcal{C}_\varphi} = \{\Phi_{C_1}, \ldots, \Phi_{C_m}\}$，其中 $\Phi_{C_j} = \bigwedge_{q\in Q_j} \neg u_{j,q}$。

容易看出：对于上述的 ATL_f 公式 φ，对其进行符号化编码所需的位变元数目

为

$$\#\mathbf{El}(\varphi) + \sum_{1\leq j\leq m} \#Q_j \tag{5.7}$$

而显然这个数目与公式的长度成线性关系。

5.4 $\mathbf{ATL}_l$ 符号化模型检验

5.4.1 $\mathbf{ATL}_l$ 公式的 tableau

现在，将符号化模型检验的研究目标转向 ATL_l 公式。从某种意义上说，ATL_l 可以看作是 ATL_f 的"对偶"。因此，其符号化模型检验算法与 ATL_f 大体类似，只是公式 tableau 的构建方式稍有不同。

ATL_l 中使用的时序连接子是 ALW。由于 ALW 采用 looping 接收条件，所以对于一个 ALW $\langle\Sigma, Q, \delta, q, \Omega\rangle$ 而言，可以将 Ω 忽略。所以为方便起见，在本节将其简写为 $\langle\Sigma, Q, \delta, q, -\rangle$。同样，缺省情况下，本节假设公式都写为否定范式。

对于 ATL_l 公式而言，其**基础公式集** 的定义与 ATL_f 公式中的定义严格相同。即：对于任意的 ATL_l，可按定义 5.3.1 相同的计算过程归纳获得 φ 的基础公式集 $\mathbf{El}(\varphi)$。但是，由于 ALW 中不关心接收状态集，所以公式的**满足集**的定义需要修改如下。

定义 5.4.1 (公式的满足集) 对于任意的 ATL_l 公式 φ, 可以定义函数 $\mathbf{Sat}_\varphi : \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi) \to 2^{2^{\mathbf{El}(\varphi)}}$ 如下。

- $\mathbf{Sat}_\varphi(true) = 2^{\mathbf{El}(\varphi)}$; $\mathbf{Sat}_\varphi(false) = \emptyset$。
- $\mathbf{Sat}_\varphi(p) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid p \in \Gamma\}$; $\mathbf{Sat}_\varphi(\neg p) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid p \notin \Gamma\}$。
- $\mathbf{Sat}_\varphi(\bigcirc\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \bigcirc\psi \in \Gamma\}$。
- $\mathbf{Sat}_\varphi(\psi_1 \wedge \psi_2) = \mathbf{Sat}_\varphi(\psi_1) \cap \mathbf{Sat}_\varphi(\psi_2)$; $\mathbf{Sat}_\varphi(\psi_1 \vee \psi_2) = \mathbf{Sat}_\varphi(\psi_1) \cup \mathbf{Sat}_\varphi(\psi_2)$。
- 若 $\psi = \mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，且 $\{a_1, \ldots, a_n\}$ 和 δ 分别为 $\mathcal{A}$ 的字母表和迁移函数，则 $\mathbf{Sat}_\varphi(\psi) = \bigcup_{1\leq k\leq n}(\mathbf{Sat}_\varphi(\varphi_k) \cap \mathbf{Sat}_\varphi(\mathscr{S}^+_\psi(\delta(q, a_k))))$。
- 若 $\psi = \neg\mathcal{A}^q(\varphi_1, \ldots, \varphi_n)$，则 $\mathbf{Sat}_\varphi(\psi) = 2^{\mathbf{El}(\varphi)} \backslash \mathbf{Sat}_\varphi(\mathcal{A}^q(\varphi_1, \ldots, \varphi_n))$。 □

容易证明，对于 ATL_l 公式而言，引理 5.3 中的结论仍然成立。对称的，对于 ATL_l 公式而言，可以定义如下的迁移关系。

定义 5.4.2 给定 ATL_l 公式 φ, 设 $\psi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$ 是 φ 中的自动机子公式，且 $\mathcal{A} = \langle\{a_1, \ldots, a_n\}, Q, \delta, q_0, -\rangle$。定义迁移关系 $\Delta^l_\psi \subseteq (2^{\mathbf{El}(\varphi)} \times 2^Q) \times (2^{\mathbf{El}(\varphi)} \times 2^Q)$ 如下: 对于任意的 $\Gamma, \Gamma' \subseteq \mathbf{El}(\varphi)$ 以及 $P, P' \subseteq Q$, $((\Gamma, P), (\Gamma', P')) \in \Delta^l_\psi$ 当且仅当

- 如果 $P \neq \emptyset$, 那么对于任意的 $q \in Q$ 以及每个 $1 \leq k \leq n$ 有: 若 $\Gamma \in \mathbf{Sat}_\varphi(\varphi_k)$, 则 $P' \models \overline{\delta(q, a_k)}$。
- 如果 $P = \emptyset$, 那么对于任意的 $q \in Q$ 有: $q \in P'$ 当且仅当 $\Gamma' \notin \mathbf{Sat}_\varphi(\psi^q)$。

显然, 若 $\psi \sim \psi'$, 则 Δ^l_ψ 与 $\Delta^l_{\psi'}$ 相同。于是, 该关系的下标可以用 ψ 关于 $\sim$ 所在的等价类 $[\psi]^\sim_\varphi$ 替换。以后, 也将 Δ^l_ψ 写作 $\Delta^l_{[\psi]^\sim_\varphi}$。 □

例 5.4.1 设 $\varphi = \mathcal{A}(p_1, p_2)$, 其中 $\mathcal{A} = \langle\{a_1, a_2\}, \{q_1, q_2\}, \delta, q_1, -\rangle$, 并且 $\delta(q_1, a_1) = q_1 \vee q_2$, $\delta(q_1, a_2) = q_2$, $\delta(q_2, a_1) = \delta(q_2, a_2) = q_2$。显然 $\mathbf{El}(\varphi) = \{p_1, p_2, \bigcirc\varphi^{q_1}, \bigcirc\varphi^{q_2}\}$。于是, 由定义有 $((\{p_1, \bigcirc\varphi^{q_1}\}, \emptyset), (\{p_1, p_2\}, \{q_1, q_2\})) \in \Delta^l_\varphi$ — 这是因为 $\{p_1, p_2\} \notin \mathbf{Sat}_\varphi(\varphi^{q_1}) \cup \mathbf{Sat}_\varphi(\varphi^{q_2})$。 □

定义 5.4.3 ($\mathbf{ATL}_l$ 公式的 tableau) 给定 ATL_f 公式 φ, 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]^\sim_\varphi$, ..., $[\psi_m]^\sim_\varphi$, 且 ψ_i 的自动机连接子的状态集为 Q_i。那么, 可以为 φ 构建一个 *tableau* $\mathcal{T}_\varphi$, 它是一个公平迁移系统 $\langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi\rangle$。其中:

- $S_\varphi = \{\langle\Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \subseteq \mathbf{El}(\varphi), P_i \subseteq Q_i\}$。
- $(\langle\Gamma, (P_1, \ldots, P_m)\rangle, \langle\Gamma', (P'_1, \ldots, P'_m)\rangle) \in \Delta_\varphi$ 当且仅当:
  - 对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 有: $\Gamma \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当$\Gamma' \in \mathbf{Sat}_\varphi(\psi)$;
  - 对每个 $1 \leq i \leq m$ 有: $((\Gamma, P_i), (\Gamma', P'_i)) \in \Delta^l_{[\psi_i]^\sim_\varphi}$。
- $I_\varphi = \{\langle\Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\varphi)\}$。
- 对每个 $\langle\Gamma, (P_1, \ldots, P_m)\rangle \in S_\varphi$ 有: $\lambda_\varphi(\langle\Gamma, (P_1, \ldots, P_m)\rangle) = \Gamma \cap AP$。
- $\mathcal{C}_\varphi = \{C_1, \ldots, C_m\}$, 其中 $C_i = \{\langle\Gamma, (P_1, \ldots, P_m)\rangle \mid P_i = \emptyset\}$。 □

对于 ATL_l 公式 φ 而言, $\mathcal{T}_\varphi$ 所关心的原子命题集合也是 $\mathbf{El}(\varphi) \cap AP$。

注意, 定义 5.3.4 和定义 5.4.3 唯一的区别在于迁移关系 Δ_φ 的构造。即: 前者要求 $((\Gamma, P_i), (\Gamma', P'_i)) \in \Delta^f_{[\psi_i]^\sim_\varphi}$, 而后者要求 $((\Gamma, P_i), (\Gamma', P'_i)) \in \Delta^l_{[\psi_i]^\sim_\varphi}$。

例 5.4.2 设 $\varphi = \mathcal{A}(p)$, 其中 $\mathcal{A} = \langle\{a\}, \{q\}, \delta, q, -\rangle$, $\delta(q, a) = q$, 则 $\mathcal{T}_\varphi$ 的 *tableau* 如图 *5.2* 所示。其中, 初始状态集合 $I_\varphi = \{\langle\{p, \bigcirc\varphi\}, (\{q\})\rangle, \langle\{p, \bigcirc\varphi\}, \emptyset\rangle\}$。同时应当注意: 其他 6 个状态并不是初始可达的。 □

下面说明 ATL_l 公式 tableau 的"语言性质"。虽然 ATL_f 公式和 ATL_l 公式的 tableau 在构造上仅有细微的不同, 但二者语言性质的证明过程却有很大区别。因此, 这里给出其具体证明过程。

定理 5.8 对于任意 ATL_l 公式以及线性结构 π 有: 若 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$, 则 $\pi \models \varphi$。

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]^\sim_\varphi, \ldots, [\psi_m]^\sim_\varphi$。同时, 设 ψ_j 的自动机连接子$\mathcal{A}^{q_j}_j = \langle\Sigma_j, Q_j, \delta_j, q_j, -\rangle$, 以及 $\psi_j = \mathcal{A}^{q_j}_j(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$,

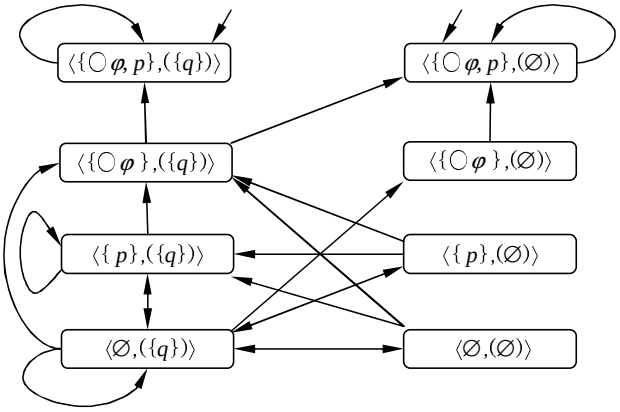


图 5.2 ATL_l tableau 示例

其中，$\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$。

设 π 是 $\mathcal{T}_\varphi$ 中公平展开迹 $s_0, s_1, \ldots$ 所对应的派生线性结构，其中 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i})\rangle$。与定理 5.4 的证明思路相同，用结构归纳法证明：对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$ 以及任意的 $i \in \mathbb{N}$ 都有：

1. 若 $\Gamma_i \in \mathbf{Sat}(\psi)$，则 $\pi, i \models \psi$；
2. 若 $\Gamma_i \notin \mathbf{Sat}(\psi)$，则 $\pi, i \not\models \psi$。

- 对于基本情形（包括：$\psi \in \{true, false\} \cup AP \cup \overline{AP}$、$\psi = \psi_1 \wedge \psi$、$\psi = \psi_1 \vee \psi_2$ 以及 $\psi = \bigcirc\psi'$ 等情况）的证明，与定理 5.4 中的过程相同。
- 现在证明：当 ψ 是正自动机公式时，结论 1 成立。

不妨设 $\psi = \psi_j^{q_0}$（这里，$1 \le j \le m$，并且 $q_0 \in Q_j$）。要证明 $\pi, i \models \psi$，只需为 ψ 构建一个在 π 上开始于位置 i 的可接收运行 $\langle T, \rho\rangle$ 即可。

- 添加根节点 ϵ，并令 $\rho(\epsilon) = q_0$。由已知条件，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$，即 $\Gamma_{i+|\epsilon|} \in \mathbf{Sat}_\varphi(\psi_j^{\rho(\epsilon)})$。
- 对每个新添加的节点 x，归纳假设 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\psi_j^{\rho(x)})$ 成立。于是，存在某个 $1 \le k \le \#\Sigma_j$ 使得 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\delta_j(\rho(x), a_{j,k})))$。由归纳假设知 $\pi, i + |x| \models \varphi_{j,k}$。同时，由引理 5.3 知存在 $Q_{j,x} \subseteq Q_j$ 使得 $Q_{j,x} \models \delta_j(\rho(x), a_{j,k})$ 并且对于任意的 $q_{x,c} \in Q_{j,x}$，有 $\Gamma_{i+|x|} \in \mathbf{Sat}(\bigcirc\psi_{x,c}^q)$ 成立。不妨设 $Q_{j,x} = \{q_{x,0}, \ldots, q_{x,t}\}$。于是对每个 $0 \le c \le t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。（当 $\delta_j(\rho(x), a_{j,k}) = true$ 时，$Q_{j,x}$ 可以为 $\emptyset$，这时 x 就成为 T 中的一个接收叶节点。）由于 $|x \cdot c| = |x| + 1$，而由迁移关系 Δ_φ 知，$\Gamma_{i+|x|+1} \in \mathbf{Sat}_\varphi(\psi^{q_{x,c}})$，换言之，$\Gamma_{i+|x\cdot c|} \in \mathbf{Sat}_\varphi(\psi^{\rho(x\cdot c)})$。

由上述构造过程，可知 $\langle T, \rho\rangle$ 中的任何一条极大路径而言，该路径或者结束于某个

接收叶节点，或者是无穷路径。因此，$\langle T,\rho\rangle$ 是 ψ 在 π 上开始于 i 的一个可接收运行。于是，$\pi,i\models\psi$。

- 现在证明：当 ψ 是负自动机公式时，结论 1 成立。

不妨设 $\psi=\neg\psi_j^{q_0}$（这里，$1\le j\le m$，并且 $q_0\in Q_j$）。要证明 $\pi,i\models\psi$，只需构造一个 $\psi_j^{q_0}$ 在 π 上开始于 i 的一个拒绝例证 $\langle T,\rho\rangle$ 即可。由于 $s_0,s_1,\ldots$ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹，于是由公平限制 C_j 知，必定存在 i_1、i_2 使得 $i<i_1<i_2$ 且 $P_{j,i_1}=P_{j,i_2}=\emptyset$。同时，对于任意的 $i_1<l<i_2$，$P_{j,l}\neq\emptyset$。于是，该构造可分为两个阶段进行。

第一阶段：构造 $\langle T,\rho\rangle$ 的第 0 至 i_1-i+1 层。

- 为 T 添加根节点 ϵ，并令 $\rho(\epsilon)=q_0$。由已知条件，$\Gamma_i\notin\mathbf{Sat}_\varphi(\psi_j^{q_0})$。换言之，$\Gamma_{i+|\epsilon|}\notin\mathbf{Sat}_\varphi(\psi^{\rho(\epsilon)})$。
- 对于 T 中每个新添加的节点 x（其中 $|x|\le i_1-i$），归纳假设 $\Gamma_{i+|x|}\notin\mathbf{Sat}_\varphi(\psi_j^{\rho(x)})$ 成立。于是，由定义有

$$\Gamma_{i+|x|}\in\bigcap_{1\le k\le\#\Sigma_j}(\overline{\mathbf{Sat}_\varphi(\varphi_{j,k})}\cup\overline{\mathbf{Sat}_\varphi(\mathscr{I}^+_{\psi_j}(\delta_j(\rho(x),a_{j,k})))}) \tag{5.8}$$

（在上面的公式中，对每个 $\Gamma\subseteq\mathbf{El}(\varphi)$ 而言，$\overline{\Gamma}$ 是 $\mathbf{El}(\varphi)\backslash\Gamma$ 的简写）。于是，对于任意的 $1\le k\le\#\Sigma_j$，或者 $\Gamma_{i+|x|}\notin\mathbf{Sat}_\varphi(\varphi_{j,k})$，或者 $\Gamma_{i+|x|}\notin\mathbf{Sat}_\varphi(\mathscr{I}^+_{\psi_j}(\delta_j(\rho(x),a_{j,k})))$。由归纳假设，$\Gamma_{i+|x|}\notin\mathbf{Sat}_\varphi(\varphi_{j,k})$ 当且仅当$\pi,i+|x|\models\varphi_{j,k}$。因此，若 $\pi,i\models\varphi_{j,k}$，则必然有 $\Gamma_{i+|x|}\notin\mathbf{Sat}_\varphi(\varphi_{j,k})$。再由引理 5.3 知必然存在某个 $Q_{j,k,x}\subseteq Q_j$ 使得 $Q_{j,k,x}\models\overline{\delta_j(\rho(x),a_{j,k})}$，并且对于任意的 $q\in Q_{j,k,x}$，有 $\Gamma_{i+|x|}\notin\mathbf{Sat}(\bigcirc\psi^q)$ 成立。这时，令 $Q_{j,x}=\bigcup_{\pi,i+|x|\models\varphi_{j_k}}Q_{j,k,x}$ 并不妨设 $Q_{j,x}=\{q_{x,0},\ldots q_{x,t}\}$。于是，对每个 $0\le c\le t$ 为 x 添加子节点 $x\cdot c$，并令 $\rho(x\cdot c)=q_{x,c}$。显然，对于每个 $1\le k\le\#\Sigma_j$，若 $\pi,i+|x|\models\varphi_{j,k}$ 则必然有 $Q_{j,x}\models\overline{\delta_j(\rho(x),q_{j,k})}$。同时，对每个 $q_{x,c}\in Q_{j,x}$ 一定有 $\Gamma_{i+|x|}\in\mathbf{Sat}_\varphi(\bigcirc\psi_j^{q_{x,c}})$。注意到 $|x\cdot c|=|x|+1$，并且 $(s_{i+|x|},s_{i+|x|+1})\in\Delta_\varphi$，于是对每个 $x\cdot c$，一定有 $\Gamma_{i+|x\cdot c|}\in\mathbf{Sat}_\varphi(q_j^{\rho(x\cdot c)})$。

若在该过程中得到的 $\langle T,\rho\rangle$ 中的每个节点都已处理完毕（x 处理完毕是指已经按照上述过程为 x 添加子节点）则所得的有穷树必然是 $\psi_j^{q_0}$ 在 π 上开始于 i 的一个拒绝例证— 这是因为，若 x 是 T 中的一个叶节点，则一定对于每个 $1\le k\le\#\Sigma_j$ 有：$\pi,i+|x|\models\varphi_{j,k}$ 蕴含 $\delta_j(\rho(x),a_{j,k})=false$。若在该过程中未将所有节点处理完毕，则这些节点只可能是 T 中深度为 i_1-i+1 的节点。此时，转入下一阶段构造。

第二阶段：构造 T 的剩余层。

- 对于每个 $x \in T$，若 $|x| = i_1 - i + 1$，则 x 是在第一阶段内创建的节点，于是有 $\Gamma_{i+|x|} = \Gamma_{i_1+1} \notin \mathbf{Sat}_\varphi(\psi^{\rho(x)})$。注意到 $\Gamma_{i_1} = \emptyset$，则由迁移关系 Δ_φ 知 $((\Gamma_{i_1}, P_{j,i_1}), (\Gamma_{i_1+1}, P_{j,i_1+1})) \in \Delta^l_{[\psi_j]^\sim_\varphi}$，所以 $\rho(x) \in P_{j,i_1+1}$。于是，$\{\rho(x) \mid x \in T, |x| = i_1 - i + 1\} \subseteq P_{j,i_1+1}$。
- 对每个 $i_2 - i > l > i_1 - i$，归纳假设 $\{\rho(x) \mid x \in T, |x| = l\} \subseteq P_{j,i+l}$ 成立。对每个节点 $x \in T$，若 $|x| = l$ 且 x 尚未处理完毕，则按如下方式对其进行处理：

  由于 $((\Gamma_{j,i+l}, P_{j,i+l}), (\Gamma_{j,i+l+1}, P_{j,i+l+1})) \in \Delta^l_{[\psi_j]^\sim_\varphi}$ 并且 $P_{j,i+l} \neq \emptyset$ 知，对每个 $1 \leq k \leq \#\Sigma$ 若 $\Gamma_{j,i+l} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，则有 $P_{j,i+l+1} \models \overline{\delta_j(\rho(x), a_{j,k})}$。由归纳假设，$\Gamma_{j,i+l} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$ 当且仅当 $\pi, i + l \models \varphi_{j,k}$。

  由于每个 $\delta_j(\rho(x), a_{j,k})$ 都是正布尔公式。由于 $\delta_j(\rho(x), a_{j,k}) \in \mathbf{B}^+(Q_j)$，所以存在 P_{i+l+l} 的（极小）子集 $Q_{j,x}$ 使得对每个 $1 \leq k \leq n$ 有：$\pi, i + |x| \models \varphi_{j,k}$ 蕴含 $Q_{j,x} \models \delta_j(\rho(x), a_{j,k})$。不妨设 $Q_{j,x} = \{q_{x,0}, \ldots, q_{x,t}\}$，于是对每个 $0 \leq c \leq t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。于是，$\{\rho(x) \mid x \in T, |x| = l+1\} \subseteq P_{j,i+l+1}$ 成立。

注意到 $P_{j,i_2} = \emptyset$，所以 T 一定是一棵有穷树。并且，若 x 是 T 中的一个叶节点，则一定是对于每个 $1 \leq k \leq n$，若 $\pi, i + |x| \models$ 则 $\delta_j(\rho(x), a_{j_k}) = false$。由定义，$\langle T, \rho\rangle$ 是 $\psi_j^{q_0}$ 在 π 上开始于 i 的一个拒绝例证，故而 $\pi, i \models \neg\psi_j^{q_0}$。即：$\pi, i \models \psi$。

- 现在证明：当 ψ 是正/负自动机公式时，结论 2 成立。
  - 当 ψ 是正自动机公式且 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi)$ 时，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\neg\psi)$。由前面的证明，$\pi, i \models \neg\psi$，所以 $\pi, i \not\models \psi$。
  - 当 ψ 是负自动机公式且 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi)$ 时，不放设 $\psi = \neg\phi$，于是 ϕ 是正自动机公式并且 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi)$。由前面的证明，$\pi, i \models \phi$，因此 $\pi, i \not\models \psi$。

显然，上述归纳是完全的。由于 $s_0 \in I_\varphi$，所以 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$。由所证结论，有 $\pi, 0 \models \varphi$，即：$\pi \models \varphi$ 成立。 □

定理 5.9 *对于任意的 ATL_l 公式 φ 以及线性结构 π，若 $\pi \models \varphi$ 则 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。*

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]^\sim_\varphi, \ldots, [\psi_m]^\sim_\varphi$。同时，设 ψ_j 的自动机连接子$\mathcal{A}^{q_j} = \langle \Sigma_j, Q_j, \delta_j, q_j, -\rangle$，以及 $\psi_j = \mathcal{A}_j^{q_j}(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$，其中，$\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$。

设线性结构 π 满足 $\pi \models \varphi$，证明 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$ 的过程同定理 5.5 的证明过程类似：只需证明 π 是 $\mathcal{T}_\varphi$ 中某个公平展开迹 $\sigma = s_1, s_1, \ldots$ 的派生线性结构即可。其中，每个 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i})\rangle$。

- σ 的构造过程如下：

对于每个 $i \in \mathbb{N}$，令 $\Gamma_i = \{\psi \in \mathbf{El}(\varphi) \mid \pi, i \models \psi\}$。

对于每个 $1 \leq j \leq m$ 以及每个 $i \in \mathbb{N}$，$P_{j,i}$ 的值按照如下方式确定。

首先，令 $C_0^j = 0$ 以及 $P_{j,C_0^j} = \emptyset$。其次，对于每个 $k \in \mathbb{N}$，归纳假设 $P_{j,C_k^j} = \emptyset$，则使用下列步骤确定 C_{k+1}^j 的值，并且构建每个 $P_{j,l}$，（其中 $C_k^j < l \leq C_{k+1}^j$）：

- 令 $P_{j,C_k^j+1} = \{q \in Q_j \mid \pi, C_k^j + 1 \models \psi_j^q\}$。
- 于是，对每个 $q \in P_{j,C_k^j+1}$，存在 ψ_j^q 在 π 上起始于位置 $C_k^j + 1$ 的某个拒绝例证 $\langle T_{q,k}, \rho_{q,k}\rangle$。注意到每个 $T_{q,k}$ 一定是有穷树，于是 $|T_{q,k}| < +\infty$。令

$$C_{k+1}^j = C_k^j + 1 + \max\{|T_{q,k}| \mid q \in P_{j,C_k^j+1}\}。$$

- 对于每个 $C_k^j < l \leq C_{k+1}^j$，令

$$P_{j,l} = \bigcup\nolimits_{q \in P_{j,C_k^j}} \{\rho_{q,k}(x) \mid x \in T_{q,k}, \text{并且 } |x| = l - C_k^j - 1\}。$$

- 容易验证 $P_{j,C_{k+1}^j} = \emptyset$。

- 再证明 σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

事实上，与定理 5.5 中的构造相比，二者唯一的区别在于：前者基于可接收运行，而后者基于拒绝例证。因此，同样可以证明：对于任意的 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 以及每个 $i \in \mathbb{N}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi)$。接下来只需证明对每个 $1 \leq j \leq m$ 以及 $i \in \mathbb{N}$，都有 $((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta^l_{[\psi_j]^\sim_\varphi}$ 即可。

- 若 $P_{j,i} \neq \emptyset$，不妨设 $C_k^j < i < C_{i+1}^j$。对于每个 $q \in P_{j,i}$，由 σ 的构造知，必然存在某个 $q' \in Q_j$ 以及 $\psi_j^{q'}$ 在 π 上开始于 $C_k^j + 1$ 的拒绝例证 $\langle T_{q',k}, \rho(q', k)\rangle$，以及某个 $x \in T_{q',k}$ 使得 $|x| = i - C_k^j - 1$，且 $\rho_{q',k}(x) = q$。

  由于 $\langle T_{q',k}, \rho(q', k)\rangle$ 是 $\psi_j^{q'}$ 在 π 上开始于 $C_k^j + 1$ 的拒绝例证，所以，对于任意的 $1 \leq l \leq \#\Sigma_j$ 若 $\pi, |x| + C_k^j + 1 \models \varphi_{j,l}$（即 $\pi, i \models \varphi_{j,l}$）则必有 $Q_{j,x} = \{\rho_{q',k}(x \cdot c) \mid c \in \mathbb{N}, x \cdot c \in T_{q',k}\} \models \overline{\delta_j(q, a_{j,l})}$。

  由 σ 的构造知 $Q_{j,x} \subseteq P_{j,i+1}$，再由正布尔公式的单调性知 $P_{j,i+1} \models \delta_j(q, a_{j,l})$。
- 若 $P_{j,i} = \emptyset$，则由 σ 的构造知，必然存在某个 $k \in \mathbb{N}$ 使得 $i = C_k^j$。于是，$P_{j,i+1} = P_{j,C_k^j+1} = \{q \in Q_j \mid \pi, i + 1 \models \psi_j^q\}$。换言之，$P_{j,i+1} = \{q \in Q_j \mid \Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi_j^q)\}$。

综上所述，对于任意 $i \in \mathbb{N}$，都有 $(s_i, s_{i+1}) \in \Delta_\varphi$ 成立。由于 $\pi \models \varphi$（即 $\pi, 0 \models \varphi$），所以有 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$。于是，$s_0 \in I_\varphi$。再由 σ 的构造，对于每个 $1 \leq j \leq m$，都有无穷多个 $i \in \mathbb{N}$，使得 $P_{j,i} = \emptyset$。因此，σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

- 最后，同定理 5.5，可证明 π 是 σ 的派生线性结构。即：对于任意的 $i \in \mathbb{N}$，有 $\pi(i) \cap \mathbf{El}(\varphi) = \Gamma_i \cap AP$ 成立。

综上所述，π 是 $\mathcal{T}_\varphi$ 中某个公平展开迹的派生线性结构，因此 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。 □

推论 5.10 ($\mathbf{ATL}_l$ 公式 tableau 的语言性质) *对于任意 ATL_l 公式以及线性结构 π 有: $\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。*

于是，同定理 5.7，ATL_l 的模型检验问题也可以转化为 CTL 模型检验问题。

定理 5.11 *对于任意的公平迁移系统 $\mathcal{M}$ 以及 ATL_l 公式 φ 而言，$\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。*

5.4.2 基于 BDD 的 $\mathbf{ATL}_l$ tableau 编码

现在介绍 ATL_l 模型检验算法基于 BDD 的实现。与 5.3.2 节类似，这里主要说明公式 tableau 的编码方法。

给定 ATL_l 公式 φ，设 $[\psi_1]_\varphi^\sim$，…，$[\psi_m]_\varphi^\sim$ 是其所有自动机子公式由关系 $\sim$ 所划分的等价类，其中 ψ_j 的自动机连接子为 $\mathcal{A}_j = \langle \Sigma_j, Q_j, \delta_j, q_j, - \rangle$，并且 $\psi_j = \mathcal{A}_j(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$。于是，$\mathcal{T}_\varphi = \langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi \rangle$ 的各要素编码过程如下。

位变元集合：同 ATL_f 公式 tableau，对每个 $\psi \in \mathbf{El}(\varphi)$，引入一个位变元 z_ψ；同时，对每个 $1 \le j \le m$ 以及 $q \in Q_j$ 引入一个位变元 $u_{j,q}$。

状态约束：由于 $\mathcal{T}_\varphi$ 中没有后继的状态的存在对于 CTL 公式 $\mathsf{EG}\ true$ 的成立与否没有影响，因此仍令 $\Phi_{S_\varphi} = true$。

迁移关系：同 ATL_f tableau 中的做法类似，首先对于每个 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，构建 $\mathbf{Sat}_\varphi(\psi)$ 之布尔表示 ϑ_ψ。对于 $\psi \in AP \cup \overline{AP} \cup \{true, false\}$、$\psi = \psi_1 \wedge \psi_2$、$\psi = \psi_1 \vee \psi_2$、$\psi = \bigcirc\psi'$ 的情况与 ATL_f 中的定义相同。但是，由于 ATL_l 的连接子不关心接收状态，因此需要对 ψ 是正、负自动机公式的情况做如下修改。

- 若 $\psi = \psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$），不妨设 $\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$，则令 $\vartheta_\psi = \bigvee_{1 \le k \le \#\Sigma_j} (\vartheta_{\varphi_{j,k}} \wedge \vartheta_{\mathscr{I}_\psi^+(\delta_j(q,a_{j,k}))})$。
- 若 $\psi = \neg\psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$），则令 $\vartheta_\psi = \neg\vartheta_{\psi_j^q}$。

同样容易证明：对于任意的 $\Gamma \subseteq \mathbf{El}(\varphi)$，$\Gamma \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 ϑ_ψ 在变元指派 e_Γ 下真值为 1。其中 e_Γ 满足：对于每个 $\phi \in \mathbf{El}(\varphi)$，$e_\Gamma(z_\phi) = 1$ 当且仅当 $\phi \in \Gamma$。

于是，Φ_{Δ_φ} 就是下列各项之合取。

$$\bigwedge_{\bigcirc\psi \in \mathbf{El}(\varphi)} z_{\bigcirc\psi} \leftrightarrow \vartheta'_\psi \tag{5.9}$$

$$\bigwedge_{1\le j\le m}((\bigvee_{q\in Q_j} u_{j,q})\to \bigwedge_{q\in Q_j}(u_{j,q}\to \bigwedge_{1\le k\le \#\Sigma_j}(\vartheta_{\varphi_{j,k}}\to \mathbf{U}'_j(\overline{\delta_j(q,a_{j,k})}))))\tag{5.10}$$

$$\bigwedge_{1\le j\le m}((\bigwedge_{q\in Q_j}\neg u_{j,q})\to \bigwedge_{q\in Q_j}(u'_{j,q}\leftrightarrow \neg\vartheta'^{q}_{\psi_j}))\tag{5.11}$$

在上述各式中，ϑ'_ψ 以及 $\mathbf{U}'_j$ 的定义同 5.3.2 节。同样，容易验证这样得到的 Φ_{Δ_φ} 确为 Δ_φ 的布尔编码。

初始状态集：初始状态集的布尔编码 Φ_{I_φ} 仍然为 ϑ_φ。

标记函数：对每个 $p\in AP\cap\mathbf{El}(\varphi)$，仍令 $\Phi^p_{\lambda_\varphi}=z_p$。

公平性约束：公平性约束的编码同前：$\Phi_{\mathcal{C}_\varphi}=\{\Phi_{C_1},\ldots,\Phi_{C_m}\}$，其中 $\Phi_{C_j}=\bigwedge_{q\in Q_j}\neg u_{j,q}$。

同样，对于上述的 ATL_l 公式 φ，对其进行符号化编码所需的位变元数目仍为

$$\#\mathbf{El}(\varphi)+\sum_{1\le j\le m}\#Q_j\tag{5.12}$$

所以，对其编码的（空间）复杂度仍然正比于公式长度。

5.5 ATL_r 符号化模型检验

5.5.1 ATL_r 公式带 rank 的拒绝例证

本节将研究 ATL_r 的符号化模型检验技术。同 ATL_f、ATL_l 的模型检验技术相比，ATL_r 的模型检验技术的空间复杂度要高许多— 虽然这三种逻辑的模型检验问题都是 PSPASCE-complete 的[30]，但在 5.5.3 节中会看到：对 ATL_r 公式 φ 而言，编码其 tableau 所需的位变元数目为 $\mathcal{O}(|\varphi|^2)$。当 φ 中含有自动机公式时，tableau 的构建是基于 NBW 的求补算法进行的。而 Yan（严奇琦）证明了该算法的下界为 $(0.76n)^n$（见文 [112, 113]）。

自动机公式的可接收运行以及拒绝例证仍然是构建 ATL_r 公式 tableau 的关键。在本节，将介绍若干与之相关的概念以及某些重要性质。在本节，若 ABW $\mathcal{A}=\langle\Sigma,Q,\delta,q,\Omega\rangle$ 的接收条件 Ω 是一个接收状态集合 F，则直接将其记作 $\langle\Sigma,Q,\delta,q,F\rangle$。

定义 5.5.1 (标记树中的子树同构节点) 设 $\langle T,\rho\rangle$ 是一棵标记树，$x_1,x_2\in T$。称 x_1 和 x_2 是 $\langle T,\rho\rangle$ 中的**子树同构节点**，（记作 $x_1\approx_{\langle T,\rho\rangle}x_2$）当且仅当

- 对于任意的 $y\in\mathbb{N}^*$，$x_1\cdot y\in T$ 当且仅当 $x_2\cdot y\in T$；
- 若 $x_1\cdot y\in T$，则 $\rho(x_1\cdot y)=\rho(x_2\cdot y)$。 □

定义 5.5.2 (层次子树同构标记树) 称标记树 $\langle T,\rho\rangle$ 是 i-**层子树同构的**（这里，$l\in\mathbb{N}$），当且仅当对于任意的 $x_1,x_2\in T$，若 $|x_1|=|x_2|\leq i$ 且 $\rho(x_1)=\rho(x_2)$，则 $x_1\approx_{\langle T,\rho\rangle}x_2$。

称标记树 $\langle T,\rho\rangle$ 是**层次子树同构的**，如果对于任意的 $x_1,x_2\in T$，若 $|x_1|=|x_2|$ 且 $\rho(x_1)=\rho(x_2)$，则 $x_1\approx_{\langle T,\rho\rangle}x_2$。 □

下面，将 Büchi 自动机理论中的一些基本结论扩展至 ATL_r。在自动机理论中，任意的 ABW 在某无穷字 w 上存在（可接收）运行，当且仅当其在 w 上存在层次子树同构的（可接收）运行。类似的，该结论对于 ATL_r 自动机公式仍然成立。

引理 5.12 *自动机公式 φ 在 π 上有开始于 i 的可接收运行当且仅当 φ 在 π 上有开始于 i 的层次子树同构的可接收运行。*

证明. 该定理的证明完全类似于自动机中的相应证明。充分性显然，往证必要性。

设 $\langle T,\rho\rangle$ 是 φ 在 π 上开始于 i 的一个可接收运行，则自顶向下的对 $\langle T,\rho\rangle$ 做如下修改：

- 令 $\langle T_0,\rho_0\rangle=\langle T,\rho\rangle$，则显然 $\langle T_0,\rho_0\rangle$ 是 0-层子树同构的。
- 假设 $\langle T_i,\rho_i\rangle$ 是 i-层子树同构。若其不是 i+1-层子树同构的，则必然存在 $x_1,x_2\in T_i$（不妨设 x_1 是 x_2 的兄长节点），使得 $|x_1|=|x_2|=i+1$ 并且 $x_1\not\approx_{\langle T_i,\rho_i\rangle}x_2$。于是，可以将 $\langle T_i,\rho_i\rangle$ 中以 x_2 为根的子标记树替换为以 x_1 为根的子标记树。即：可以暂时得到这样一棵标记树 $\langle T_i',\rho_i'\rangle$，其中
  - $T_i'=T_i\backslash\{x_2\cdot y\mid x_2\cdot y\in T_i\}\cup\{x_2\cdot y\mid x_1\cdot y\in T_i\}$；
  - 对 T_i' 中的每个节点 x，若存在某个 $y\in\mathbb{N}^*$ 使得 $x=x_2\cdot y$，则 $\rho_i'(x)=\rho_i(x_1\cdot y)$；否则 $\rho_i'(x)=\rho_i(x)$。

  由于 T_i 中深度为 $i+1$ 的节点数目有穷，所以经过若干次上述替换必将得到一个 $i+1$-层子树同构的标记树，令其为 $\langle T_{i+1},\rho_{i+1}\rangle$。
- 令上述过程的极限为 $\langle\bar{T},\bar{\rho}\rangle$，即 $\langle\bar{T},\bar{\rho}\rangle=\lim\limits_{i\to\infty}\langle T_i,\rho_i\rangle$。

由于对于每个 $i\in\mathbb{N}$ 而言，$\langle T_i,\rho_i\rangle$ 都是 i-层子树同构的，所以 $\langle\bar{T},\bar{\rho}\rangle$ 一定是层次子树同构的。

容易看出：对于上述过程的每次修改，若修改前是 φ 在 π 上开始于 i 的运行，则修改后必是 φ 在 π 上开始于 i 的运行。— 以从 $\langle T_i,\rho_i\rangle$ 到 $\langle T_i',\rho_i'\rangle$ 为例，只需说明对于每个 $x\in T_i'$（分三种情况：x 是 x_2 的祖先节点；$x=x_2$，$x=x_2\cdot y$ 且 $x_1\cdot y\in T_i'$），其子节点均满足定义中的条件即可。所以 $\langle\bar{T},\bar{\rho}\rangle$ 是 φ 在 π 上开始于 i 的运行。

此外，对于 $\bar{T}$ 中的每条极大路径 $\bar{\sigma}$，必然存在 T 中的一条极大路径 σ 使得

$\bar{\rho}(\bar{\sigma}) = \rho(\sigma)$。因此，$\langle \bar{T}, \bar{\rho} \rangle$ 是 φ 在 π 上开始于 i 的可接收运行。 □

同样，也可以证明下面的引理。

引理 5.13 *自动机公式 φ 在 π 上开始于 i 的拒绝例证当且仅当 φ 有在 π 上开始于 i 上层次子树同构的拒绝例证。*

接下来，介绍带 rank 的 ATL_r 自动机公式拒绝例证。以下，分别用 $\mathbb{E}$ 和 $\mathbb{O}$ 表示**偶数集** 和**奇数集**。

定义 5.5.3 (带 rank 的 ATL_r 自动机公式拒绝例证) *设 ATL_r 自动机公式 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, F \rangle$。则 φ 在线性结构 π 上起始于位置 i 的一个带 rank 的拒绝例证 是一个三元组 $\langle T, \rho, \kappa \rangle$。其中:*

- $\rho : T \to Q$，*并且满足:*
  - $\rho(\epsilon) = q$;
  - *对每个* $x \in T$ *以及* $1 \leq k \leq n$，*若* $\pi, i + |x| \models \varphi_k$，*则* $\{\rho(x \cdot c) \mid c \in \mathbb{N}, x \cdot c \in T\} \models \overline{\delta(\rho(x), a_k)}$。
- $\kappa : T \to \mathbb{N}$，*并且满足:*
  - *对每个* $x, y \in T$，*若* x *是* y *的祖先节点则* $\kappa(x) \geq \kappa(y)$;
  - *对每个* $x \in T$，*若* $\rho(x) \in F$，*则* $\kappa(x) \in \mathbb{E}$;
  - *对* T *中的每条无穷路径* $\sigma = x_0, x_1, \ldots$，*有无穷多个* $j \in \mathbb{N}$ *使得* $\kappa(x_j) \in \mathbb{O}$。

有时，也将 $\langle T, \rho, \kappa \rangle$ 写为 $\langle T, (\rho, \kappa) \rangle$。其中，$(\rho, \kappa)$ 作为一个整体看作是从 T 到 $Q \times \mathbb{N}$ 的函数。 □

事实上，一个带 rank 的拒绝例证实际上是带有两个标记函数的标记数。该概念是受 Kupferman 和 Vardi 的 NBW 求补算法中的“带 rank 的运行图”（Ranked Run Diagram）概念[114, 115] 启发而提出。这里，将其推广至 ATL_r 的自动机公式。接下来，会说明普通的公式拒绝例证与带 rank 的公式拒绝例证之间的关系。

定理 5.14 *设 ATL_r 自动机公式 φ 存在一个在 π 上开始于位置 i 的拒绝例证当且仅当 φ 存在一个在 π 上开始于位置 i 的带 rank 的拒绝例证。*

证明. 不妨设 $\varphi = \mathcal{A}(\varphi_1, \ldots, \varphi_n)$，其中 $\mathcal{A} = \langle \{a_1, \ldots, a_n\}, Q, \delta, q, F \rangle$。现在，分别证明该命题的充分性和必要性（该证明的必要性部分是受 Kupferman 和 Vardi 相关工作启发得到）。

- 充分性的证明非常容易。

设 $\langle T, \rho, \kappa \rangle$ 是 φ 在 π 上开始于位置 i 的 rank 的拒绝例证，则现在证明 $\langle T, \rho \rangle$ 就是 φ 在 π 上开始于 i 的一个拒绝例证。

由定义 2.2.24，只需证明对于 T 中的任意一条无穷路径 σ 都有 $\rho(\sigma)$ 不满足接

收条件即可。也就是说，若 $\sigma = x_0, x_1, \ldots$，则仅有有穷多个 $j \in \mathbb{N}$，使得 $\rho(x_j) \in F$。

若不然，则存在无穷多个 j 使得 $\rho(x_j) \in F$。首先，任取其中一个使得 $\rho(x_{j_0}) \in F$ 的 j_0。由定义，$\kappa(x_{j_0}) \in \mathbb{E}$。于此同时，由于 σ 中有无穷多个 j 使得 $\kappa(x_j) \in \mathbb{O}$，所以必然存在某个 $j_0' > j_0$ 使得 $\kappa(x_{j_0'}) \in \mathbb{O}$。由于 x_{j_0} 是 $x_{j_0'}$ 的祖先节点，由 κ 的限制，必然有 $\kappa(x_{j_0}) > \kappa(x_{j_0'})$（注意：由于二者奇偶性不同，不可能相等）。再一次，由于存在无穷多个 j 使得 $\rho(x_j) \in F$，所以必然有某个 $j_1 > j_0'$ 使得 $\kappa(x_{j_1}) \in F$，于是 $\kappa(x_{j_1}) \in \mathbb{E}$。同样，因为 $x_{j_0'}$ 是 x_{j_1} 的祖先节点，所以 $\kappa(x_{j_0'}) > \kappa(x_{j_1})$。重复此讨论，可以得到一条无穷序列 $j_0, j_0', j_1, j_1', j_2, \ldots$，使得 $\kappa(x_{j_0}), \kappa(x_{j_0'}), \kappa(x_{j_1}), \kappa(x_{j_1'}), \kappa(x_{j_2}), \ldots$ 是一个无穷递降序列。但是，由于 κ 的值域是自然数，所以 $\kappa(x_{j_0})$ 是一个有穷非负整数，因此这样的无穷递降序列不可能存在。

- 再证必要性。在此之前，先定义几个将要用到的概念：

1. 对于任意一棵 Q-标记树 $\langle T, \rho\rangle$，以及 $l \in \mathbb{N}$，用 $\mathbf{Lb}_{\langle T,\rho\rangle}(l)$ 表示所有出现在 T 中深度为 l 节点的标记。即：$\mathbf{Lb}_{\langle T,\rho\rangle}(l) = \{\rho(x) \mid x \in T, |x| = l\}$。
2. 称 T 中的节点 x **在 T 中有穷**，是指 T 中以 x 为根的子树是有穷树。即：$\{y \in \mathbb{N}^* \mid x \cdot y \in T\}$ 是有穷集。
3. 称 T 中的节点 x 是 F-**无关**的，是指对于 x 的每个子孙节点 y，都有 $\rho(y) \notin F$。

设 $\langle T, \rho\rangle$ 是 φ 在 π 上开始于 i 的一个拒绝例证，由引理 5.13，不妨设 $\langle T, \rho\rangle$ 是层次子树同构的。现在，按照如下的方式构造一系列的 Q-标记树 $\langle T_i, \rho_i\rangle$，其中：

- $T_0 = T$；$\rho_0 = \rho$。
- $T_{2k+1} = T_{2k} \backslash \{x \in T_{2k} \mid x \text{ 在 } T \text{ 中有穷}\}$；$\rho_{2k+1}$ 是 ρ 在 T_{2k+1} 上的压缩（即：对于任意的 $x \in T_{2_k+1}$，$\rho_{2k+1}(x) = \rho(x)$）。
- $T_{2k+2} = T_{2k+1} \backslash \{x \in T_{2k+1} \mid x \text{ 在 } T \text{ 中是 } F\text{-无关的}\}$；$\rho_{2k+2}$ 是 ρ 在 T_{2k+2} 上的压缩。

现在证明：存在某个 $k \in \mathbb{N}$，使得 $T_k = \emptyset$。为此，需要归纳证明如下结论：对于每个 $k \leq \#Q$，存在某个 $l_k \in \mathbb{N}$，使得对于任意的 $l \geq l_k$，有 $\#\mathbf{Lb}_{\langle T_{2k},\rho_{2k}\rangle}(l) \leq \#Q - k$。

- 由于 $\langle T_0, \rho_0\rangle = \langle T, \rho\rangle$ 是一棵 Q-标记树，于是对于每个 $l \in \mathbb{N}$，有 $\#\mathbf{Lb}_{\langle T_0,\rho_0\rangle}(l) \leq \#Q$。因此，当 $k = 0$ 时，直接令 $l_0 = 0$ 即可。
- 假设当 $k = m$ 时 l_k 存在，现在说明当 $k = m+1$（这里，$m+1 \leq \#Q$）时 l_k 也存在。如果 $T_{2m+1} = \emptyset$，则结论显然；否则，T_{2m+1} 中没有有穷节点。现在证明在从 T_{2m+1} 得到 T_{2m+2} 的过程中至少有一棵无穷子树被删除。换言之，T_{2m+1} 中必然存在某个 F-无关的节点。

  若不然，假设 T_{2m+1} 中没有 F-无关的节点。于是，必然存在某个节点 $x_0 \in$

T_{2m+1}，使得 $\rho_{2m+1}(x) \in F$。同样，在 T_{2m+1} 中，以 x_0 为根的子树中也必不存在 F-无关的节点。因此，在该子树中必然存在某个节点 x_1，使得 $\rho_{2m+1}(x_1) \in F$。由于以 x_1 为根的子树中也没有 F-无关的节点，所以此讨论可无穷重复进行。这样，就可以得到 T_{2m+1} 中的一条无穷路径 σ，使得每个 x_0，x_1，x_2，... 均在 σ 中出现。而对于每个 $i \in \mathbb{N}$，都有 $\rho_{2m+1}(x_i) \in F$。注意到每个 x_i 又均是 T 中的节点，所以 σ 必是 T 中的无穷路径；而 ρ_{2m+1} 是 ρ 在 T_{2m+1} 上的压缩，所以每个 $\rho(x_i) \in F$，这与 $\langle T, \rho\rangle$ 是 φ 在 π 上开始于 i 的一个拒绝例证矛盾。

于是，T_{2m+1} 中存在某个 F-无关的节点 x，其所在的（无穷）子树被删除。在 T_{2m+1} 中任取一条开始于 x 的无穷路径 $\sigma = x'_0, x'_1, x'_2, \ldots$（其中 $x'_0 = x$），则现在证明对于每个 $j \in \mathbb{N}$，有

$$\rho_{2m+1}(x'_j) \in \mathbf{Lb}_{\langle T_{2m+1}, \rho_{2m+1}\rangle}(|x'_j|) \backslash \mathbf{Lb}_{\langle T_{2m+2}, \rho_{2m+2}\rangle}(|x'_j|)$$

成立。显然，由于 $x'_j \in T_{2m+1}$，所以 $\rho_{2m+1}(x'_j) \in \mathbf{Lb}_{\langle T_{2m+1}, \rho_{2m+1}\rangle}(|x'_j|)$，现在只要证明 $\rho_{2m+1}(x'_j) \notin \mathbf{Lb}_{\langle T_{2m+2}, \rho_{2m+1}\rangle}(|x'_j|)$ 即可。注意到 x'_j 是 x 的子孙节点，由定义易知 x'_j 必然也是 T_{2m+1} 中的 F-无关节点，因此 x'_j 必然会被删除。此外，对于任意的 $x''_j \in T_{2m+1}$，如果 $|x'_j| = |x''_j|$ 且 $\rho_{2m+1}(x''_j) = \rho_{2m+1}(x'_j)$，那么由于 T 是层次子树同构的，所以对于任意的 $y \in \mathbb{N}^*$ 有："若 $x''_j \cdot y \in T_{2m+1}$，则 $x'_j \cdot y \in T_{2m+1}$，且 $\rho_{2m+1}(x'_j \cdot y) = \rho_{2m+1}(x''_j \cdot y)$"。这说明 x''_j 也是 T_{2m+1} 中的 F-无关节点，从而以其为根的子树也会被删除。于是，$\rho_{2m+1}(x'_j) \notin \mathbf{Lb}_{\langle T_{2m+2}, \rho_{2m+1}\rangle}(|x'_j|)$。因此，对于任意的 $l \geq |x|$，必然有

$$\#\mathbf{Lb}_{\langle T_{2m+2}, \rho_{2m+2}\rangle}(l) \leq \#\mathbf{Lb}_{\langle T_{2m+1}, \rho_{2m+1}\rangle}(l) - 1$$

成立。于是，当 $k = m + 1$ 时，只需取 $l_{m+1} = \max\{l_m, |x|\}$ 即可。

以上，证明了"对于任意的 $l \geq l_k$，有 $\#\mathbf{Lb}_{\langle T_{2k}, \rho_{2k}\rangle}(l) \leq \#Q - k$"。于是，当 $k = \#Q$ 时，$\langle T_{2k}, \rho_{2k}\rangle$ 必然是一棵有穷树，从而 $T_{2\times(\#Q)+1} = \emptyset$。接下来，对每个 $x \in T$，令

$$\kappa(x) = \max\{i \mid 0 \leq i \leq 2 \times (\#Q) + 1, x \in T_i\},$$

现在证明 $\langle T, \rho, \kappa\rangle$ 是 φ 在 π 上开始于位置 i 的一个带 rank 的拒绝例证。显然，ρ 满足定义 5.5.3 中的约束，下面只对 κ 的约束进行逐条验证即可。

- 对于任意的 $x, y \in T$，若 y 是 x 的子孙节点，则 x 不会在 y 被删除之前删除，因而 $\kappa(x) \geq \kappa(y)$。

- 对于每个 $x \in T$，若 $\rho(x) \in F$，则 x 只可能是作为某个 T_{2k} 中的有穷节点删除。于是 $\kappa(x) \in \mathbb{E}$。
- 对 T 中的每条无穷路径 $\sigma = x_0, x_1, \ldots$，已经证明有 $\kappa(x_0) \geq \kappa(x_1) \geq \ldots$ 成立。但由于 $\kappa(x_0)$ 是一个有穷的自然数，所以必然存在某个 $k \in \mathbb{N}$，使得对于任意的 $k' \geq k$，$\kappa(x_{k'})$ 为常数。不妨设 $n = \kappa(x_k) = \kappa(x_{k+1}) = \ldots$，则 n 必然是奇数。若不然，有 $\kappa(x_k) = \kappa(x_{k+1}) = \ldots \in \mathbb{E}$。这意味着 $x_k, x_{k+1}, \ldots$ 将在某个 T_{2m} 中作为有穷节点被删除。但是，$x_k, x_{k+1}, \ldots$ 是开始于 x_k 的一条无穷路径，这不可能! 故而，$n \in \mathbb{O}$。所以 σ 中无穷多个 j 使得 $\kappa(x_j) \in \mathbb{O}$。

综上所述，$\langle T, \rho, \kappa \rangle$ 确为 φ 在 π 上开始于 i 的一个带 rank 的拒绝例证。 □

由定理 2.9 以及定理 5.14 的证明过程，立即得到如下推论。

推论 5.15 *设 ATL_r 自动机公式 φ 的连接子的状态集为 Q，则 $\pi, i \models \neg\varphi$ 当且仅当 φ 在 π 上存在某个开始于 i 的带 rank 的某个拒绝例证 $\langle T, \rho, \kappa \rangle$，并且 $\kappa(\epsilon) \leq (2 \times (\#Q) + 1)$。*

5.5.2 $\mathbf{ATL}_r$ 公式的 tableau

本节给出 ATL_r 公式 tableau 的定义及其语言性质的证明。

对于 ATL_r 公式 φ 而言，其**基础公式集合 El**(φ) 的定义与 ATL_f 和 ATL_l 中的定义完全相同（见定义 5.3.1）。对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，公式 ψ 相对于 φ 的**满足集 Sat**$_\varphi(\psi)$ 的定义与 ATL_l 中的定义完全相同（见定义 5.4.1）。此外，为构建 ATL_r 公式的 tableau，还需引入如下概念。

首先，对于任意的 $n \in \mathbb{N}$，引入如下的缩写：

$$\mathbb{N}[n] \stackrel{\text{def}}{=} \{0, 1, \ldots, n\} \tag{5.13}$$

$$\mathbb{E}[n] \stackrel{\text{def}}{=} \mathbb{N}[n] \cap \mathbb{E} \tag{5.14}$$

$$\mathbb{O}[n] \stackrel{\text{def}}{=} \mathbb{N}[n] \cap \mathbb{O} \tag{5.15}$$

定义 5.5.4 (正布尔公式的拉伸) *给定集合 Q，则对于 Q 的任意子集 F，可以归纳定义函数 $\Lambda_F : \mathbf{B}^+(Q) \times \mathbb{N} \times \{+, -\} \to \mathbf{B}^+(Q \times \mathbb{N} \times \{+, -\})$，如下。*

- *基本情况有如下两种:*

$$\Lambda_F(q, n, -) = \begin{cases} \bigvee_{m \in \mathbb{N}[n]}(q, m, -) \vee \bigvee_{k \in \mathbb{O}[n]}(q, k, +) & , q \notin F \\ \bigvee_{m \in \mathbb{E}[n]}(q, m, -) & , q \in F, n \in \mathbb{E} \\ false & , q \in F, n \in \mathbb{O} \end{cases};$$

$$\Lambda_F(q,n,+)=\begin{cases}\bigvee_{m\in\mathbb{N}[n]}(q,m,-)\vee\bigvee_{k\in\mathbb{N}[n]}(q,k,+) & ,q\notin F\\ \bigvee_{m\in\mathbb{E}[n]}(q,m,-)\vee\bigvee_{k\in\mathbb{E}[n]}(q,k,+) & ,q\in F,n\in\mathbb{E}\\ false & ,q\in F,n\in\mathbb{O}\end{cases}。$$

- $\Lambda_F(\theta_1\vee\theta_2,n,\pm)=\Lambda_F(\theta_1,n,\pm)\vee\Lambda_F(\theta_2,n,\pm)$（这里，$\pm\in\{+,-\}$）。
- $\Lambda_F(\theta_1\wedge\theta_2,n,\pm)=\Lambda_F(\theta_1,n,\pm)\wedge\Lambda_F(\theta_2,n,\pm)$（这里，$\pm\in\{+,-\}$）。 □

例 5.5.1 设 $Q=\{q_1,q_2\}$，$F=\{q_1\}$，则 $\Lambda_F(q_1\wedge q_2,2,-)=((q_1,2,-)\vee(q_1,0,-))\wedge((q_2,1,+)\vee\bigvee_{0\le k\le 2}(q_2,k,-))$。 □

引理 5.16 设 $\theta\in\mathbf{B}^+(Q)$，$F\subseteq Q$，$n\in\mathbb{N}$，$\Upsilon\subseteq Q\times\mathbb{N}\times\{+,-\}$。则 $\Upsilon\models\Lambda_F(\theta,n,+)$ 当且仅当 Υ 存在一个子集 Υ' 使得:

- $\{q\mid$ 存在 $(q,m,\pm)\in\Upsilon'\}\models\theta$;
- 若 $(q,m,\pm)\in\Upsilon'$ 且 $q\in F$ 则 $m\in\mathbb{E}$;
- 若 $(q,m,\pm)\in\Upsilon'$ 则 $m\le n$。

$\Upsilon\models\Lambda_F(\theta,n,-)$ 当且仅当 Υ 存在一个子集 Υ'' 使得:

- $\{q\mid$ 存在 $(q,m,\pm)\in\Upsilon''\}\models\theta$;
- 若 $(q,m,\pm)\in\Upsilon'$ 且 $q\in F$ 则 $m\in\mathbb{E}$;
- 若 $(q,m,\pm)\in\Upsilon''$ 则 $m\le n$;
- 若 $(q,m,+)\in\Upsilon''$ 则 $m\in\mathbb{O}$。

证明. 对正布尔公式 θ 使用公式结构归纳法即可。 □

定义 5.5.5 给定 ATL_r 公式 φ，设 $\psi=\mathcal{A}(\varphi_1,\ldots,\varphi_n)$ 是 φ 中的自动机子公式，且 $\mathcal{A}=\langle\{a_1,\ldots,a_n\},Q,\delta,q_0,F\rangle$。定义迁移关系 $\Delta_\psi^{r+}\subseteq(2^{\mathbf{El}(\varphi)}\times 2^Q)\times(2^{\mathbf{El}(\varphi)}\times 2^Q)$ 如下: 对于任意的 $\Gamma,\Gamma'\subseteq\mathbf{El}(\varphi)$ 以及 $P,P'\subseteq Q$，$((\Gamma,P),(\Gamma',P'))\in\Delta_\psi^{r+}$ 当且仅当:

- 对每个 $q\in P$ 有: $\Gamma\in\mathbf{Sat}_\varphi(\psi^q)$; 对每个 $q\in P'$ 有: $\Gamma'\in\mathbf{Sat}_\varphi(\psi^q)$。
- 若 $P\neq\emptyset$，则对于任意的 $q\in Q\backslash F$，存在某个 $1\le k\le n$，使得 $\Gamma\in\mathbf{Sat}_\varphi(\varphi_k)$，且 $P'\models\delta(q,a_k)$。
- 若 $P=\emptyset$，则对于任意的 $q\in Q\backslash F$，$q\in P'$ 当且仅当 $\Gamma'\in\mathbf{Sat}_\varphi(\psi^q)$。

显然，若 $\psi\sim\psi'$，则 Δ_ψ^{r+} 与 $\Delta_{\psi'}^{r+}$ 相同。于是，该关系的下标可以用 ψ 关于 $\sim$ 所在的等价类 $[\psi]_\varphi^\sim$ 替换。以后，也将 Δ_ψ^{r+} 写作 $\Delta_{[\psi]_\varphi^\sim}^{r+}$。 □

注意本节中 Δ_ψ^{r+} 的定义与 5.3.1 节中 Δ_ψ^f 的定义之间的区别：前者比后者多了一个额外的约束。此外，在定义 ATL_r 公式的 tableau 时，还用到了另外一个派生迁移关系，定义如下。

定义 5.5.6 给定 ATL_r 公式 φ，设 $\psi=\mathcal{A}(\varphi_1,\ldots,\varphi_n)$ 是 φ 中的自动机子公式，且 $\mathcal{A}=\langle\{a_1,\ldots,a_n\},Q,\delta,q,F\rangle$。定义迁移关系 $\Delta_\psi^{r-}\subseteq(2^{\mathbf{El}(\varphi)}\times 2^{Q\times\mathbb{N}\times\{+,-\}})\times$

$(2^{\mathbf{El}(\varphi)} \times 2^{Q\times\mathbb{N}\times\{+,-\}})$ 如下：对于任意的 $\Gamma,\Gamma' \subseteq \mathbf{El}(\varphi)$ 以及 $\Upsilon,\Upsilon' \subseteq Q\times\mathbb{N}\times\{+,-\}$，$((\Gamma,\Upsilon),(\Gamma',\Upsilon')) \in \Delta_{\psi}^{r-}$ 当且仅当：

- 对于每个 $(q,m,\pm) \in \Upsilon \cup \Upsilon'$，若 $q \in F$ 则 $n \in \mathbb{E}$。
- 对于每个 $(q,m,\pm) \in \Upsilon$ 有：以及每个 $1 \le k \le n$ 有：若 $\Gamma \in \mathbf{Sat}_{\varphi}(\varphi_k)$，则 $\Upsilon' \models \Lambda_F(\overline{\delta(q,a_k)},m,\pm)$。
- 如果 $\Upsilon \subseteq (Q\times\mathbb{N}\times\{+\})$，那么 $\Upsilon' \subseteq (Q\times\mathbb{N}\times\{-\})$，并且对每个 $q \in Q$ 有：若 $\Gamma' \notin \mathbf{Sat}_{\varphi}(\psi^q)$ 则存在某个 $m \in \mathbb{N}$ 使得 $(q,m,-) \in \Upsilon'$。

以上，$\pm \in \{+,-\}$。显然，若 $\psi \sim \psi'$，则 Δ_{ψ}^{r-} 与 $\Delta_{\psi'}^{r-}$ 相同。于是，该关系的下标可以用 ψ 关于 $\sim$ 所在的等价类 $[\psi]_{\varphi}^{\sim}$ 替换。以后，也将 Δ_{ψ}^{r-} 写作 $\Delta_{[\psi]_{\varphi}^{\sim}}^{r-}$。 □

定义 5.5.7 (ATL$_r$ 公式的 tableau) 给定 ATL_r 公式 φ，设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]_{\varphi}^{\sim}$，…，$[\psi_m]_{\varphi}^{\sim}$，且 ψ_i 自动机连接子的状态集为 Q_i。则可为 φ 构建一个 *tableau* $\mathcal{T}_{\varphi}$，它是一个特殊的（公平）迁移系统 $\langle S_{\varphi},\Delta_{\varphi},I_{\varphi},\lambda_{\varphi},\mathcal{C}_{\varphi}\rangle$。其中：

- $S_{\varphi} = \{\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle \mid \Gamma \subseteq \mathbf{El}(\varphi), P_i \subseteq Q_i, \Upsilon_i \subseteq Q_i\times\mathbb{N}[2\times(\#Q_i)+1]\times\{+,-\}\}$。
- $(\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle,\langle\Gamma',(P'_1,\ldots,P'_m),(\Upsilon'_1,\ldots,\Upsilon'_m)\rangle) \in \Delta_{\varphi}$ 当且仅当下面条件被满足：
  - 对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 有：$\Gamma \in \mathbf{Sat}_{\varphi}(\bigcirc\psi)$ 当且仅当 $\Gamma' \in \mathbf{Sat}_{\varphi}(\psi)$；
  - 对每个 $1 \le i \le m$ 有：$((\Gamma,P_i),(\Gamma',P'_i)) \in \Delta_{[\psi_i]_{\varphi}^{\sim}}^{r+}$。
  - 对每个 $1 \le i \le m$ 有：$((\Gamma,\Upsilon_i),(\Gamma,\Upsilon'_i)) \in \Delta_{[\psi_i]_{\varphi}^{\sim}}^{r-}$。
- $I_{\varphi} = \{\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle \mid \Gamma \in \mathbf{Sat}_{\varphi}(\varphi)\}$。
- 对每个 $\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle \in S_{\varphi}$，$\lambda_{\varphi}(\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle) = \Gamma \cap AP$。
- $\mathcal{C}_{\varphi} = \{C_1^+,\ldots,C_m^+\}\cup\{C_1^-,\ldots,C_m^-\}$，其中 $C_i^+ = \{\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle \in S_{\varphi} \mid P_i = \emptyset\}$，$C_i^- = \{\langle\Gamma,(P_1,\ldots,P_m),(\Upsilon_1,\ldots,\Upsilon_m)\rangle \in S_{\varphi} \mid \Upsilon_i \subseteq Q_i\times\mathbb{N}\times\{+\}\}$。 □

对于 ATL$_l$ 公式 φ 而言，$\mathcal{T}_{\varphi}$ 所关心的原子命题集合也是 $\mathbf{El}(\varphi)\cap AP$。下面证明 ATL$_r$ 公式 tableau 的语言性质。

定理 5.17 对于任意 ATL_r 公式以及线性结构 π，若 $\pi \in \mathbf{L}(\mathcal{T}_{\varphi})$，则 $\pi \models \varphi$。

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]_{\varphi}^{\sim}$，…，$[\psi_m]_{\varphi}^{\sim}$。同时，设 ψ_j 的自动机连接子 $\mathcal{A}_j^{q_j} = \langle\Sigma_j,Q_j,\delta_j,q_j,F_j\rangle$，以及 $\psi_j = \mathcal{A}_j^{q_j}(\varphi_{j,1},\ldots,\varphi_{j,\#\Sigma_j})$，其中，$\Sigma_j = \{a_{j,1},\ldots,a_{j,\#\Sigma_j}\}$。

同定理 5.4 以及定理 5.8 的证明思路相同：设 π 是 $\mathcal{T}_\varphi$ 中公平展开迹 $s_0, s_1, \ldots$ 所对应的派生线性结构，其中 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i}), (\Upsilon_{1,i}, \ldots, \Upsilon_{m,i}) \rangle$。下面用结构归纳法证明：对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$ 以及每个 $i \in \mathbb{N}$ 有：

1. 若 $\Gamma_i \in \mathbf{Sat}(\psi)$，则 $\pi, i \models \psi$；
2. 若 $\Gamma_i \notin \mathbf{Sat}(\psi)$，则 $\pi, i \not\models \psi$。

- 对于基本情形（包括：$\psi \in \{true, false\} \cup AP \cup \overline{AP}$、$\psi = \psi_1 \wedge \psi$、$\psi = \psi_1 \vee \psi_2$ 以及 $\psi = \bigcirc\psi'$ 等情况）的证明，与定理 5.4 中的过程相同。
- 现在证明：当 ψ 是正自动机公式时，结论 1 成立。

不妨设 $\psi = \psi_j^{q_0}$（这里，$1 \leq j \leq m$，并且 $q_0 \in Q_0$）。这时，要证明 $\pi, i \models \psi$，只需为 ψ 构造一个在 π 上开始于位置 i 的可接收运行 $\langle T, \rho \rangle$ 即可。由于 $s_0, s_1, \ldots$ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹，于是，由公平限制 C_j^+ 知必定存在无穷序列 $i_0, i_1, i_2, \ldots$ 使得 $i < i_0 < i_1 < i_2 < \ldots$ 并且对于每个 i_k 有 $P_{j,i_k} = \emptyset$，同时对于每个 $i_k < l < i_{k+1}$ 有 $P_{j,l} \neq \emptyset$。于是，$\langle T, \rho \rangle$ 的构造过程可以按如下方式分阶段进行。

首先，构造 $\langle T, \rho \rangle$ 的第 0 层到 $i_0 - i + 1$ 层。

- 为 T 添加根节点 ϵ，并令 $\rho(\epsilon) = q_0$。由已知条件，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$。换言之，有 $\Gamma_{i+|\epsilon|} \in \mathbf{Sat}_\varphi(\psi^{\rho(\epsilon)})$ 成立。
- 对于 T 中每个新添加的节点 x（其中 $|x| \leq i_0 - i$），归纳假设 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\psi^{\rho(x)})$ 成立。由定义有

$$\Gamma_{i+|x|} \in \bigcup_{1 \leq k \leq \#\Sigma_j} (\mathbf{Sat}_\varphi(\varphi_{j,k}) \cap \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\delta_j(\rho(x), a_{j,k})))) \tag{5.16}$$

于是，必然存在某个 $1 \leq k \leq \#\Sigma$，使得 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\mathscr{I}_\psi^+(\delta_j(\rho(x), a_{j,k})))$。由归纳假设，有 $\pi, i+|x| \models \varphi_{j,k}$。同时，由引理 5.3 知必然存在某个 $Q_{j,x} \subseteq Q_j$ 使得 $Q_{j,x} \models \delta_j(\rho(x), a_{j,k})$，并且对于任意的 $q_{x,c} \in Q_{j,x}$ 有 $\Gamma_{i+|x|} \in \mathbf{Sat}(\bigcirc\psi_{x,c}^q)$ 成立。不妨设 $Q_{j,x} = \{q_{x,0}, \ldots, q_{x,t}\}$，于是对每个 $0 \leq c \leq t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。注意，当 $\delta_j(\rho(x), a_{j_k}) = true$ 时，$Q_{j,x}$ 可以为 $\emptyset$，这时 x 就成为 T 中的一个接收叶节点。对于每个新加入的节点 $x \cdot c$，由于 $|x \cdot c| = |x| + 1$，而由迁移关系 Δ_φ 知：$\Gamma_{i+|x|+1} \in \mathbf{Sat}_\varphi(\psi^{q_{x,c}})$，换言之，$\Gamma_{i+|x \cdot c|} \in \mathbf{Sat}_\varphi(\psi^{\rho(x \cdot c)})$。

若此过程中所有节点均处理完毕，则显然所得的有穷标记树是 ψ 在 π 上开始于 i 的一个可接收运行。否则，只可能是 T 中深度为 $i_0 - i + 1$ 的节点尚未处理完毕。由构造过程知，对于任意的 $x \in T$，若 $|x| = i_0 - i + 1$，则 $\Gamma_{i_0+1} \in \mathbf{Sat}_\varphi(\psi^{\rho(x)})$。这时，令 $l = 0$，转入下一个构造过程。

参数过程：给定 $l \in \mathbb{N}$，构造 $\langle T, \rho\rangle$ 的 $i_l - i + 2$ 至 $i_{l+1} - i + 1$ 层。

在使用这个构造过程时，需使用下面两个归纳不变式：假设 h 层节点已经构造完毕（ $i_l - i + 2 \leq h \leq i_{l+1} - i$），则对每个在 T 中深度为 h 的节点 x 有

- **归纳不变式一**：$\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\psi^\rho(x))$。
- **归纳不变式二**：设 σ_x 是 T 中从某深度为 $i_1 - i + 1$ 的节点到 x 的路径（该路径唯一），若对 σ_x 中每个不同于 x 的节点 y 都有 $\rho(y) \notin F$，则有 $\rho(x) \in P_{j,i+|x|}$ 成立。

容易验证：当 $h = i_l - i + 1$ 时这两个归纳不变式为真— 因为 $P_{j,i_l} = \emptyset$，所以由限制 $(s_{i_l}, s_{i_l+1}) \in \Delta_\varphi$ 知 $P_{j,i_l+1} = \{q \in Q_j \mid \Gamma_{j,i_l+1} \in \mathbf{Sat}_\varphi(\psi^q)\}$。故而由前提条件知：对每个 $x \in T$，若 $|x| = i_l - i + 1$，则 $\rho(x) \in P_{j,i_l}$。

假设 h 层节点已经构造完毕（ $i_l - i + 2 \leq h \leq i_{l+1} - i$），则按照如下方式分情况构造 $h + 1$ 层的节点。

- 情况一：当 $\rho(x) \in P_{j,i+|x|}$ 且 $\rho(x) \notin F$ 时。这时，由 $\Delta^{r+}_{[\psi]^\sim_\varphi}$ 中的限制知：存在某个 $1 \leq k \leq \#\Sigma_j$，使得 $\Gamma_{j,i+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $P_{j,i+|x|+1} \models \delta_j(\rho(x), a_{j,k})$。由归纳假设，$\pi, i+|x| \models \varphi_{j,k}$。此外，必然存在 $P_{j,i+|x|+1}$ 的某个子集 $\{q_{x,0}, \ldots, q_{x,t}\}$ 满足 $\delta_j(\rho(x), a_{j,k})$。这时，为每个 $0 \leq c \leq t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。
- 情况二：当 $\rho(x) \notin P_{j,i+|x|}$ 或 $\rho(x) \in F$ 时。这时，由归纳不变式一以及 $\mathbf{Sat}_\varphi$ 的定义知：存在某个 $1 \leq k \leq \#\Sigma_j$，使得 $\Gamma_{j,i+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$，并且 $\Gamma_{j,i+|x|} \in \mathbf{Sat}_\varphi(\mathscr{I}^+_\psi(\delta_j(\rho(x), a_{j,k})))$。由引理 5.3，存在某个 $\{q_{x,0}, \ldots, q_{x,t}\} \subseteq Q_j$，使得对每个 $0 \leq c \leq t$ 都有 $\Gamma_{i+|x|} \in \mathbf{Sat}_\varphi(\bigcirc\psi^{q_{x,c}})$。由 Δ_φ 的约束知 $\Gamma_{i+|x|+1} \in \mathbf{Sat}_\varphi(\psi^{q_{x,c}})$。这时，为每个 $0 \leq c \leq t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$。

容易证明：按照上述方式添加 $h+1$ 层节点后，两个归纳不变式仍然成立。此外，注意到 $P_{j,i_{l+1}} = \emptyset$，所以由归纳不变式二立即可得结论：对于 T 中任一开始于某深度为 $i_l - i + 1$ 的节点而终结于某深度为 $i_{l+1} - i$ 节点的路径 σ 中，必然包含某个节点 x 使得 $\rho(x) \in F$。

如果上述过程终止于某个 l，即：能够得到一个每条路径都终止于接收叶节点的有穷树，则显然已经得到 ψ 在 π 上开始于 i 的一个可接收运行。否则，以 $l + 1$ 为参数重复调用以上过程。

若上述过程最终得到的 $\langle T, \rho\rangle$一棵无穷 Q-标记树，则容易验证其为 ψ 在 π 上开始于 i 的一个运行。为说明其是可接收的，任取 T 的无穷路径 $\sigma = x_0, x_1, \ldots$，不妨设 $|x_0| = h$，则由重复构造过程知：对于每个 $l \in \mathbb{N}$，若 $i_l - i + 1 \geq h$，则在 σ 中

必然存在某个深度介于 $i_l - i + 1$ 和 $i_{l+1} - i$ 的节点 x，使得 $\rho(x) \in F$。于是有无穷多个 $k \in \mathbb{N}$，使得 $\rho(x_k) \in F$。由定义，$\langle T, \rho\rangle$ 必是 ψ 在 π 上开始于 i 的一个可接收运行。于是，$\pi, i \models \psi$。

- 下面证明当 ψ 是负自动机公式时结论 1 成立。

不妨设 $\psi = \neg\psi_j^{q_0}$，其中 $1 \leq j \leq m$，$q_0 \in Q_j$，则 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi_j^q)$。由公平性约束 C_j^- 知存在序列 $i_0, i_1, \ldots$ 其中 $i < i_0 < i_1 < \ldots$，并且对于每个 $l \in \mathbb{N}$ 有 $\Upsilon_{j,i_l} \subseteq Q_j \times \mathbb{N} \times \{+\}$。现在，分两个阶段证明 $\pi, i \models \psi$。

首先，证明对于任意的 $q' \in Q_j$，若 $\Gamma_{i_0+1} \notin \mathbf{Sat}_\varphi(\psi_j^{q'})$，则 $\pi, i_0 + 1 \not\models \psi_j^{q'}$。

在该前提下，要证明 $\pi, i_0 + 1 \not\models$ 只需要构造一个 ψ_j^q 在 π 上开始于 $i_0 + 1$ 的带 rank 的拒绝例证 $\langle T, \rho, \kappa\rangle$ 即可。为此，需要借助一个额外的标记 $\tau : T \to \{+, -\}$，并且暂时需构造四元组 $\langle T, \rho, \kappa, \tau\rangle$ 如下：

- 构造 T 的根节点 ϵ。由于 $\Upsilon_{i_0} \subseteq (Q_j \times \mathbb{N} \times \{+\})$，$((\Gamma_{i_0}, \Upsilon_{j,i_0}), (\Gamma_{i_0+1}, \Upsilon_{j,i_0+1})) \in \Delta^{r-}_{[\psi_j]^\sim_\varphi}$，以及 $\Gamma_{i_0+1} \notin \mathbf{Sat}_\varphi(\psi_j^{q'})$ 所以存在某个 $n \in \mathbb{N}$，使得 $(q', n, -) \in \Upsilon_{j,i_0+1}$（由 $\mathcal{T}_\varphi$ 的定义，$n \leq 2 \times (\#Q_j) + 1$）。于是，令 $\rho(\epsilon) = q'$，$\kappa(\epsilon) = n$，$\tau(\epsilon) = -$。这样，就有 $(\rho(\epsilon), \kappa(\epsilon), \tau(\epsilon)) \in \Upsilon_{j,i_0+1+|\epsilon|}$ 成立。
- 对每个已经构造的节点 x，归纳假设 $(\rho(x), \kappa(x), \tau(x)) \in \Upsilon_{j,i_0+1+|x|}$ 成立。由 $((\Gamma_{i_0+1+|x|}, \Upsilon_{j,i_0+1+|x|}), (\Gamma_{i_0+2+|x|}, \Gamma_{j,i_0+2+|x|})) \in \Delta^{r-}_{[\psi_j]^\sim_\varphi}$ 知，对于任意的 $1 \leq k \leq \#\Sigma_j$ 有：若 $\Gamma_{i_0+1+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$（由归纳假设，这等价于 $\pi, i_0+1+|x| \models \varphi_{j,k}$），则 $\Upsilon_{i_0+2+|x|} \models \Lambda_{F_j}(\overline{\delta_j(\rho(x), a_{j,k})}, \kappa(x), \tau(x))$。设 $\Upsilon_{i_0+2+|x|}$ 的满足上述性质的某个极小子集为 $\Upsilon_{j,x} = \{(q_{x,0}, n_{x,0}, \pm_{x,0}), \ldots, (q_{x,t}, n_{x,t}, \pm_{x,t})\}$，则对每个 $0 \leq c \leq t$，为 x 添加子节点 $x \cdot c$，并令 $\rho(x \cdot c) = q_{x,c}$，$\kappa(x \cdot c) = n_{x,c}$，$\tau(x \cdot c) = \pm_{x,c}$。这时，有如下性质成立：
  1. 注意到 $\Upsilon_{j,x}$ 的极小性，由定义 5.5.4 可知：对每个 $0 \leq c \leq t$ 都有 $n_{x \cdot c} \leq \kappa(x)$。
  2. 同时，由于对每个 $1 \leq k \leq \#\Sigma_j$ 有 $\Gamma_{i_0+1+|x|} \in \mathbf{Sat}_\varphi(\varphi_{j,k})$ 蕴含 $\Upsilon_{j,x} \models \Lambda_{F_j}(\overline{\delta_j(\rho(x), a_{j,k})}, \kappa(x), \tau(x))$，这时对 $\overline{\delta_j(\rho(x), a_{j,k})}$ 使用结构归纳法可以得到 $\{\rho(x \cdot c) \mid 0 \leq c \leq t\} \models \delta_j(\rho(x), a_{j,k})$。

  此外，显然对每个 $x \cdot c$ 都有 $(\rho(x \cdot c), \kappa(x \cdot c), \tau(x \cdot c)) \in \Upsilon_{j,i_0+2+|x|}$。因而归纳不变式对 x 的每个子节点都成立。

现在证明 $\langle T, \rho, \kappa\rangle$ 是 $\psi_j^{q'}$ 上开始于 $i_0 + 1$ 的一个带 rank 的拒绝例证。在此，只需要验证 ρ 和 κ 满足定义 5.5.3 即可。

首先说明 ρ 满足定义中的约束。

- 由构造知 $\rho(\epsilon)=q'$；
- 此外，由添加子节点的过程知，对每个 $x\in T$ 以及每个 $1\le k\le \#\Sigma_j$，若 $\pi,i_0+1+|x|\models\varphi_{j,k}$ 则 $\{\rho(x\cdot c)\mid x\cdot c\in T\}\models\delta_j(\rho(x),a_{j,k})$。

再说明 κ 满足定义中的约束。

- 由添加子节点的过程，对 x 的每个子节点 $x\cdot c$，都有 $\kappa(x)\ge\kappa(x\cdot c)$。于是由 $\ge$ 关系的传递性知若 x 是 y 的祖先节点，则必有 $\kappa(x)\ge\kappa(y)$。
- 由构造过程的归纳不变式知：对于每个 $x\in T$，必然存在某个 $(q,n,\pm)\in\Upsilon_{j,i_0+|x|}$ 使得 $q=\rho(x)$，$n=\kappa(x)$。由于 $((\Gamma_{i_0+1+|x|},\Upsilon_{i_0+1+|x|}),(\Gamma_{i_0+2+|x|},\Upsilon_{i_0+2+|x|}))\in\Delta^{r-}_{[\psi_j]^\sim_\varphi}$，所以由定义 5.5.7 知：若 $q\in F$ 则 $n\in\mathbb{E}$。
- 对于 T 中的每一条无穷路径 $\sigma=x_0,x_1,\ldots$，现在说明有无穷多个 $k\in\mathbb{N}$ 使得 $\kappa(x_k)\in\mathbb{O}$。对于每个 $l\ge1$，显然有 $(\rho(x_{i_l-i_0}),\kappa(x_{i_l-i_0}),\tau(x_{i_1-i_0}))\in\Upsilon_{j,i_l+1}$，以及 $(\rho(x_{i_{l+1}-i_0-1}),\kappa(x_{i_{l+1}-i_0-1}),\tau(x_{i_{1+1}-i_0-1}))\in\Upsilon_{j,i_{l+1}}$。要证明该结论，只需说明在 $x_{i_l-i_0}$ 和 $x_{i_{l+1}-i_0-1}$ 之间必然存在某个节点 x' 使得 $\kappa(x')\in\mathbb{O}$ 即可。考虑到 $\Upsilon_{j,i_l+1}\subseteq Q_j\times\mathbb{N}\times\{-\}$（见定义 5.5.7）以及 $\Upsilon_{j,i_{l+1}}\subseteq Q_j\times\mathbb{N}\times\{+\}$，所以 $\tau(x_{i_l-i_0})=-$，$\tau(x_{i_{l+1}-i_0-1})=+$。于是，$x_{i_l-i_0}$ 和 $x_{i_{l+1}-i_0-1}$ 之间必然存在某个节点 x_t 使得 $\tau(x_t)=-$ 而 $\tau(x_{t+1})=+$。由构造过程知 $\Upsilon_{j,x_t}=\{(\rho(x_t\cdot c),\kappa(x_t\cdot c),\tau(x_t\cdot c))\mid x_t\cdot c\in T\}$ 是满足 “对于任意 $1\le k\le\#\Sigma_j$，若 $\Gamma_{i_0+1+|x_t|}\in\mathbf{Sat}_\varphi(\varphi_{j,k})$，则 $\Upsilon_{j,x_t}\models\Lambda_{F_j}(\overline{\delta_j(\rho(x_t),a_{j,k})},\kappa(x_t),\tau(x_t))$” 的最小集合，因此 Υ_{j,x_t} 中的每个元素都是集合 $Q_j\times\mathbb{N}\times\{+,-\}$ 中在表达式

$$\bigwedge_{\Gamma_{i_0+1+|x_t|}\in\mathbf{Sat}_\varphi(\varphi_{j,k})}\Lambda_{F_j}(\overline{\delta_j(\rho(x_t),a_{j,k})},\kappa(x_t),\tau(x_t))$$

中出现的三元组。因为 $\tau(x_t)=-$，所以由定义 5.5.4 知若 $(q,n,+)\in\Upsilon_{j,x_t}$ 则 $q\in\mathbb{O}$。注意到 x_{t+1} 是 x_t 的子节点，所以 $(\rho(x_{t+1}),\kappa(x_{t+1}),\tau(x_{t+1}))\in\Upsilon_{j,x_t}$。又因为 $\tau(x_{t+1})=+$，所以 $\kappa(x_{t+1})\in\mathbb{O}$，从而结论得证。

综上所述，$\langle T,\rho,\kappa\rangle$ 确为 $\psi_j^{q'}$ 在 π 上开始于 i_0+1 的带 rank 的拒绝例证。由定义 5.5.7 中的约束，$\kappa(\epsilon)\le2\times(\#Q_j)+1$。于是，由推论 5.15 得 $\pi,i_0+1\not\models\psi_j^{q'}$。

再证对每个 $i\le l\le i_0+1$ 及 $q\in Q_j$ 有：$\Gamma_l\notin\mathbf{Sat}_\varphi(\psi_j^q)$ 蕴含 $\pi,l\not\models\psi_j^q$。

该结论自后向前逆向归纳证明。

- 当 $l=i_0+1$ 时，已经证明该结论成立。
- 假设该结论对于 $l+1$ 成立。于是，对每个 $q\in Q_j$ 有：

$$\begin{aligned}&\Gamma_l\notin\mathbf{Sat}_\varphi(\psi_j^q)\\ \Leftrightarrow\ &\Gamma_l\in\textstyle\bigcup_{1\le k\le\#\Sigma_j}(\overline{\mathbf{Sat}_\varphi(\varphi_{j,k})}\cup\overline{\mathbf{Sat}_\varphi(\mathscr{I}^+_{\psi_j}(\delta_j(q,a_{j,k})))}) && \text{⟅}\mathbf{Sat}_\varphi\text{ 的定义⟆}\end{aligned}$$

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$，$\Gamma_l \in \mathbf{Sat}_\varphi(\varphi_{j,k})$ 蕴含 $\Gamma_l \notin \mathbf{Sat}_\varphi(\mathscr{I}^+_{\psi_j}(\delta_j(q, a_{j,k})))$。

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：$\pi, l \models \varphi_{j,k}$ 蕴含 $\Gamma_l \notin \mathbf{Sat}_\varphi(\mathscr{I}^+_{\psi_j}(\delta_j(q, a_{j,k})))$。⟅关于子公式的归纳假设⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则存在 $Q'_j \subseteq Q_j$ 使得 $Q'_j \models \overline{\delta_j(q, a_{j,k})}$ 且对每个 $q' \in Q'_j$ 有 $\Gamma_l \notin \mathbf{Sat}_\varphi(\bigcirc\psi_j^{q'})$。⟅引理 5.3⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则存在 $Q'_j \subseteq Q_j$ 使得 $Q'_j \models \overline{\delta_j(q, a_{j,k})}$ 且对每个 $q' \in Q'_j$ 有 $\Gamma_{l+1} \notin \mathbf{Sat}_\varphi(\psi_j^{q'})$。⟅$(s_l, s_{l+1}) \in \Delta_\varphi$⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则存在 $Q'_j \subseteq Q_j$ 使得 $Q'_j \models \overline{\delta_j(q, a_{j,k})}$ 且对每个 $q' \in Q'_j$ 有 $\pi, l+1 \not\models \psi_j^{q'}$（即 $\pi, l+1 \models \neg\psi_j^{q'}$）。⟅该结论在 $l+1$ 处的归纳假设⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则存在 $Q'_j \subseteq Q_j$ 使得 $Q'_j \models \overline{\delta_j(q, a_{j,k})}$ 且对每个 $q' \in Q'_j$ 有 $\pi, l \models \bigcirc\neg\psi_j^{q'}$。⟅$\bigcirc$ 算子语义定义⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则 $\pi, l \models \mathscr{I}^-_{\psi_j}(\overline{\delta_j(q, a_{j,k})})$。⟅引理 5.1⟆

$\Leftrightarrow$ 对每个 $1 \le k \le \#\Sigma_j$ 有：若 $\pi, l \models \varphi_{j,k}$ 则 $\pi, l \models \neg\mathscr{I}^+_{\psi_j}(\delta_j(q, a_{j,k}))$。⟅引理 5.1⟆

$\Leftrightarrow$ $\pi, i \models \bigwedge_{1 \le k \le \#\Sigma_j}(\neg\varphi_{j,k} \vee \neg\mathscr{I}^+_{\psi_j}(\delta_j(q, a_{j,k})))$（换言之，$\pi, i \not\models \bigvee_{1 \le k \le \#\Sigma_j}(\varphi_{j,k} \wedge \mathscr{I}^+_{\psi_j}(\delta_j(q, a_{j,k})))$）。

$\Leftrightarrow$ $\pi, i \not\models \psi_j^q$。⟅展开定理（定理 5.2）⟆

于是，若该结论对于 $l+1$ 成立，则其对于 l 也成立。

特别的，当 $l = i$，$q = q_0$ 时，结论也成立。而 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi_j^{q_0})$ 正是前提条件。所以这时有 $\pi, i \not\models \psi_j^{q_0}$。换言之，$\pi, i \models \psi$。所以当 ψ 是负自动机公式时结论 1 也成立。

- 同定理 5.4 以及定理 5.8 中讨论的情形相同："ψ 是正/负自动机公式时结论 1 成立" 将直接蕴含"ψ 是负/正自动机公式时结论 2 成立"。

显然，上述归纳是完全的。由于 $s_0 \in I_\varphi$，所以 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$。由上述结论有 $\pi, 0 \models \varphi$，即：$\pi \models \varphi$ 成立。 □

定理 5.18 *对于任意的 ATL_r 公式 φ 以及线性结构 π，若 $\pi \models \varphi$ 则 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。*

证明. 设 φ 中所有的自动机子公式被关系 $\sim$ 划分成的等价类为 $[\psi_1]^\sim_\varphi, \ldots, [\psi_m]^\sim_\varphi$。同时，设 ψ_j 的自动机连接子 $\mathcal{A}^{q_j} = \langle \Sigma_j, Q_j, \delta_j, q_j, F_j \rangle$，以及 $\psi_j = \mathcal{A}_j^{q_j}(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$，其中，$\Sigma_j = \{a_{j,1}, \ldots, a_{j,\#\Sigma_j}\}$。

设线性结构 π 满足 $\pi \models \varphi$，同定理 5.5 及定理 5.9 的证明思路相同：采用如下策略证明 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$：

1. 根据 π 构造 $\mathcal{T}_\varphi$ 中的无穷状态序列 $\sigma = s_0, s_1, \ldots$。其中，$s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i}),$

$(\Upsilon_{1,i},\ldots,\Upsilon_{m,i})\rangle$，并且该构造保证对每个 $1 \leq j \leq m$ 都有无穷多个 i 使得 $P_{j,i} = \emptyset$ 以及无穷多个 i 使得 $\Upsilon_{j,i} \subseteq Q_j \times \mathbb{N} \times \{+\}$。

2. 证明 σ 是 $\mathcal{T}_\varphi$ 中的一条展开迹。也即需要说明：
   - $s_0 \in I_\varphi$。
   - 对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$，以及每个 $i \in \mathbb{N}$ 有：$\Gamma_i \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi)$。
   - 对每个 $1 \leq j \leq m$ 以及每个 $i \in \mathbb{N}$ 有：$((\Gamma_i, P_{j,i}),(\Gamma_{i+1}, P_{j,i+1})) \in \Delta^{r+}_{[\psi_j]^\sim_\varphi}$ 以及 $((\Gamma_i, \Upsilon_{j,i}),(\Gamma_{i+1}, \Upsilon_{j,i+1})) \in \Delta^{r-}_{[\psi_j]^\sim_\varphi}$

   从而，再由构造保证 σ 是 $\mathcal{T}_\varphi$ 中的公平展开迹。
3. 最后证明 π 是 σ 的派生线性结构。于是就有 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

• 首先给出状态序列 σ 的构造过程。

对每个 $i \in \mathbb{N}$，Γ_i 的构造同定理 5.5 及定理 5.9 中的过程相同，即：对每个 $i \in \mathbb{N}$，令 $\Gamma_i = \{\psi \in \mathbf{El}(\varphi) \mid \pi, i \models \psi\}$。

对每个 $1 \leq j \leq m$ 以及 $i \in \mathbb{N}$，$P_{j,i}$ 的构造过程如下。首先，令 $C^j_0 = 0$，并且 $P_{j,C^j_0} = \emptyset$。对每个 $k \in \mathbb{N}$，归纳假设 $P_{j,C^j_k} = \emptyset$，则使用下列步骤确定 C^j_{k+1} 的值，同时对每个 $C^j_k < l \leq C^j_{k+1}$ 构建 $P_{j,l}$：

- 令 $P_{j,C^j_k+1} = \{q \in Q_j \mid \pi, C^j_k + 1 \models \psi^q_j\}$。
- 于是，对每个 $q \in P_{j,C^j_k+1}$，存在 ψ^q_j 在 π 上起始于位置 $C^j_k + 1$ 的某个可接收运行 $\langle T_{q,k}, \rho_{q,k}\rangle$。对每个 $T_{q,k}$，令

$$T'_{q,k} = T_{q,k} \backslash \{x \in T_{q,k} \mid x \text{ 存在祖先节点 } y \text{ 使得 } \rho_{q,k}(y) \in F_j\}。$$

  于是，每个 $T'_{q,k}$ 必然是有穷树— 否则 $T_{q,k}$ 中必然存在某条无穷路径 $x_0, x_1, \ldots$，使得每个 $\rho_{q,k}(x_l) \notin F_j$，矛盾!
- 令 $C^j_{k+1} = C^j_k + 1 + \max\{|T'_{q,k}| \mid q \in P_{j,C^j_k+1}\}$。显然，如果 $P_{j,C^j_k+1} = \emptyset$，则有 $C^j_{k+1} = C^j_k + 1$ 成立。
- 对每个 $C^j_k < l \leq C^j_{k+1}$，令 $P_{j,l} = \bigcup_{q \in P_{j,C^j_k}} \{\rho_{q,k}(x) \mid x \in T'_{q,k}, \text{且 } |x| = l - C^j_k - 1\}$。
- 容易验证 $P_{j,C^j_{k+1}} = \emptyset$。

对每个 $1 \leq j \leq m$ 以及 $i \in \mathbb{N}$，集合 $\Upsilon_{j,i}$ 的构造过程如下。首先，令 $D^j_0 = 0$，$\Upsilon_{j,D^j_0} = \emptyset$。在确定 D^j_{k+1} 的值以及构造 Υ_{j,D^j_k+1} 和 $\Upsilon_{j,D^j_{k+1}}$ 之间的三元组集合时，还需借助一个额外的数据结构 Θ_k。Θ_k 中的每个元素都是形如 $\langle T_{q,h}, \rho_{q,r}, \kappa_{q,h}\rangle$ 的三元组。其中 $q \in Q_j$，$r \leq k$。直观的讲：$\langle T_{q,r}, \rho_{q,r}, \kappa_{q,r}\rangle$ 是 ψ^q_j 在 π 上开始于 $D^j_r + 1$ 的一个带 rank 的拒绝例证。特别的，令 $\Theta_{-1} = \emptyset$。同时，对每个 $k \in \mathbb{N}$，归纳假设

$\Upsilon_{j,D_k^j} \subseteq Q_j \times \mathbb{N} \times \{+\}$，同时，$\Theta_{k-1}$ 已经存在。于是，Υ_{j,D_k^j+1} 和 Υ_{j,D_{k+1}^j} 之间的三元组集合采用如下过程构建：

- 对每个 $q \in Q_j$，若 $\pi, D_k^j + 1 \models \neg\psi_j^q$，则由推论 5.15，存在 ψ_j^q 在 π 上开始于 $D_k^j + 1$ 的带 rank 的拒绝例证 $\langle T_{q,k}, \rho_{q,k}, \kappa_{q,k}\rangle$，同时 $\rho_{q,k}(\epsilon) \leq 2 \times (\#Q_j) + 1$。令

$$\Upsilon_{j,k} = \Upsilon_{j,k-1} \cup \{\langle T_{q,k}, \rho_{q,k}, \kappa_{q,k}\rangle \mid \pi, D_k^j + 1 \models \neg\psi_j^q\}。$$

- 对每个 $\langle T_{q,r}, \rho_{q,r}, \kappa_{q,r}\rangle \in \Theta_k$，定义函数$\tau_{q,r,k} : T_{q,r} \to \{+, -\}$ 如下：对每个 $x \in T_{q,h}$，若 x 满足：
  1. $|x| > D_k^j - D_r^j$；
  2. $T_{q,k}$ 中存在路径 $x_1, \ldots, x_n$，其中 $x_n = x$，$|x_1| = D_k^j - D_r^j + 1$，并且存在 $1 \leq m \leq n$ 使得 $\kappa_{q,r}(x_m) \in \mathbb{O}$

  则令 $\tau_{q,r,k}(x) = +$；否则，令 $\tau_{q,r,k}(x) = -$。显然，对每个 $T_{q,h} \in \Theta_k$，必存在某个最小的深度 $h_{q,r}$ 使得对任意的 $x \in T_{q,r}$ 有：若 $|x| \geq h_{q,r}$ 则 $\tau_{q,r,k}(x) = +$。令

$$D_{k+1}^j = \max\{D_r^j + h_{q,r} + 1 \mid \langle T_{q,r}, \rho_{q,r}, \kappa_{q,r}\rangle \in \Theta_k\}。$$

  容易验证：$D_{k+1}^j > D_k^j$。
- 对于每个 $D_k^j < l \leq D_{k+1}^j$，令

$$\Upsilon_{j,l} = \bigcup_{\langle T_{q,r}, \rho_{q,r}, \kappa_{q,r}\rangle \in \Theta_k} \{(\rho_{q,r}(x), \kappa_{q,r}(x), \tau_{q,r,k}(x)) \mid x \in T_{q,r}, |x| = l - D_r^j - 1\}。$$

  则容易验证：$\Upsilon_{j,D_k^j+1} \subseteq Q_j \times \mathbb{N} \times \{-\}$；以及 $\Upsilon_{j,D_{k+1}^j} \subseteq Q_j \times \mathbb{N} \times \{+\}$。

同时注意到：对于任意的 $1 \leq j \leq m$ 以及 $i \in \mathbb{N}$，若 $(q, n, \pm) \in \Upsilon_{j,i}$ 则 $n \leq 2 \times (\#Q_j) + 1$。于是，每个 s_i 都是 $\mathcal{T}_\varphi$ 中的状态。

- 现在说明 σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

类似于定理 5.9 中的过程，可以证明：“对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$ 以及任意 $i \in \mathbb{N}$ 而言，$\pi, i \models \psi$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$” 成立。由于 $\pi, 0 \models \varphi$，所以 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$，于是 $s_0 \in I_\varphi$。同时，对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 以及任意的 $i \in \mathbb{N}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\pi, i \models \bigcirc\psi$，当且仅当 $\pi, i+1 \models \psi$，当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi)$。

接下来证明：对任意的 $i \in \mathbb{N}$ 及 $1 \leq j \leq m$，$((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta^{r+}_{[\psi_j]^\sim_\varphi}$。

- 对于任意的 $q \in P_{j,i}$，不妨设 $C_k^j + 1 \leq i < C_{k+1}^j$（注意每个 $P_{j,C_k^j} = \emptyset$，因此不必讨论 $i = C_k^j$ 的情况）。由构造过程知必然存在某个 $q' \in Q_j$ 以及某个 $x \in T_{q',k}$ 使得 $|x| = i - C_k^j - 1$ 并且 $\rho_{q',k}(x) = q$。容易看出：$\langle T_{q',k}, \rho_{q',k}\rangle$ 中以 x 为根的子标

记树恰好是 ψ_j^q 在 π 上开始于 i 的一个可接收运行，于是 $\pi, i \models \psi_j^q$。由前面证得的结论可知：$\Gamma_i \in \mathbf{Sat}_\varphi(\psi_j^q)$。同理，对于任意的 $q \in P_{j,i+1}$ 有 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi_j^q)$。

- 若 $P_{j,i} \neq \emptyset$，则对于任意的 $q \in P_{j,i} \backslash F_j$，则必然存在某个 $q' \in Q_j$ 以及某个 $x \in T'_{q',k}$ 使得 $|x| = i - C_k^j - 1$ 并且 $\rho_{q',k}(x) = q$。由 $T'_{q',k}$ 和 $T_{q',k}$ 之间的关系以及 $\rho_{q',k}(x) \notin F$，容易证明 x 在 $T_{q',k}$ 中的每个子节点必然同时存在于 $T'_{q'_k}$ 中。同时，由构造知：对于 x 的任意一个子节点 $x \cdot c$，有 $\rho_{q',k}(x \cdot c) \in P_{j,i+1}$。而由 $T_{q',k}$ 的定义知，存在某个 $1 \leq l \leq \#\Sigma_j$，使得 $\pi, i \models \varphi_{j,l}$（即：$\Gamma_i \in \mathbf{Sat}_\varphi(\varphi_{j,l})$），并且 $\{\rho_{q',k}(x \cdot c) \mid x \cdot c \in T_{q',k}\} \models \delta_j(q, a)$。所以 $P_{j,i+1} \models \delta_j(q, a_{j,l})$。
- 若 $P_{j,i} = \emptyset$，则必然是 i 等于某个 C_k^j。由构造知 $P_{j,i+1}$（也就是 P_{j,C_k^j+1}）为集合 $\{q \in Q_j \mid \pi, i+1 \models \psi_j^q\}$，即：$\{q \in Q_j \mid \Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi_j^q)\}$。

于是，由照定义有 $((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta^{r+}_{[\psi_j]_{\widetilde{\varphi}}}$。

再证明：对每个 $1 \leq j \leq m$ 及 $i \in \mathbb{N}$ 有 $((\Gamma_i \Upsilon_{j,i}), (\Gamma_{i+1}, \Upsilon_{j,i+1})) \in \Delta^{r-}_{[\psi_j]_{\widetilde{\varphi}}}$。

- 对于每个 $(q, n, \pm) \in \Upsilon_{j,i} \cup \Upsilon_{j,i+1}$，不妨设 $D_k^j + 1 \leq i \leq D_{k+1}^j$（由于 $\Upsilon_{j,D_0^j} = \Upsilon_{j,0} = \emptyset$，因此不必考虑 $i = 0$ 的情况），由构造知必然存在某个 $\langle T_{q',r}, \rho_{q',r}, \kappa_{q',r}\rangle \in \Theta_k$ 以及 $x \in T_{q',r}$ 使得 $\rho_{q',r}(x) = q$，$\kappa_{q',r}(x) = n$。由于 $\langle T_{q',r}, \rho_{q',r}, \kappa_{q',r}\rangle$ 是 $\psi_j^{q'}$ 在 π 开始于 $D_r^j + 1$ 上的一个带 rank 的拒绝例证，因此若 $q \in F$ 则 $n \in \mathbb{E}$。
- 对于每个 $(q, n, \pm) \in \Upsilon_{j,i}$（不妨设 $D_k^j + 1 \leq i \leq D_{k+1}^j$），则必然存在某个 $\langle T_{q',r}, \rho_{q',r}, \kappa_{q',r}\rangle \in \Theta_k$ 以及 $x \in T_{q',r}$ 使得 $\rho_{q',r}(x) = q$，$\kappa_{q',r}(x) = n$ 以及 $\tau_{q',r,k}(x) = \pm$。由构造在 $\langle T_{q',r}, \rho_{q',r}, \kappa_{q',r}\rangle$ 中由定义，对于每个 $1 \leq l \leq \#\Sigma_j$，若 $\pi, i \models \varphi_{j,l}$，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\varphi_{j,l})$。于是，$\{\rho_{q',r}(x \cdot c) \mid x \cdot c \in T_{q',r}\} \models \overline{\delta_j(q, a_{j,l})}$。并且，由构造知：对于每个 $x \cdot c \in T_{q',r}$，必然存在某个 $\pm_c \in \{+, -\}$ 使得 $(\rho_{q',r}(x \cdot c), \kappa_{q',r}(x \cdot c), \pm_c) \in \Upsilon_{j,i+1}$（事实上，若 $i < D_{k+1}^j$ 则 $\pm_c = \tau_{q',r,k}(x \cdot c)$；若 $i = D_{k+1}^j$，则 $\pm_c = -$）。显然，$\kappa_{q',r}(x \cdot c) \leq n$；同时，若 $\pm = -$，则 $\pm_c = +$ 仅当 $\kappa_{q',r}(x \cdot c) \in \mathbb{O}$。于是，由引理 5.16 知 $\Upsilon_{j,i+1} \models \overline{\delta_j(q, a_{j,l})}$。
- 若 $\Upsilon_{j,i} \subseteq Q_j \times \mathbb{N} \times \{+\}$，则意味着 i 等于某个 D_k^j。于是由构造知 $\Upsilon_{j,i+1} \in Q_j \times \mathbb{N} \times \{-\}$；同时，对于任意的 $q \in Q_j$，若 $\Gamma_i \notin \mathbf{Sat}_\varphi(\psi_j^q)$（即：$\pi, i \not\models \psi_j^q$），则必然存在某个 $n \leq 2 \times (\#Q_j) + 1$，使得 $(q, n, -) \in \Upsilon_{j,i+1}$。

于是，由定义 $((\Gamma_i \Upsilon_{j,i}), (\Gamma_{i+1}, \Upsilon_{j,i+1})) \in \Delta^{r-}_{[\psi_j]_{\widetilde{\varphi}}}$。

这样，对于每个 $i \in \mathbb{N}$ 都有 $(s_i, s_{i+1}) \in \Delta_\varphi$。此外，由构造知：对于每个 $1 \leq j \leq m$ 而言，σ 满足公平性约束 C_j^+（即：有无穷多个 i 使得 $P_{j,i} = \emptyset$）以及公平性约束 C_j^-（即：有无穷多个 i 使得 $\Upsilon_{j,i} \subseteq Q_j \times \mathbb{N} \times \{+\}$），所以 σ 必为 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

- 最后，同定理 5.5，可证明 π 是 σ 的派生线性结构。即：对于任意的 $i \in \mathbb{N}$ 都有 $\pi(i) \cap \mathbf{El}(\varphi) = \Gamma_i \cap AP$ 成立。

综上所述，π 是 $\mathcal{T}_\varphi$ 中某个公平展开迹的派生线性结构，因此有 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。□

推论 5.19 ($\mathbf{ATL}_r$ 公式 tableau 的语言性质) 对于任意 ATL_r 公式以及线性结构 π 有: $\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

于是，同定理 5.7 以及定理 5.11，ATL_l 的模型检验问题也可以转化为 CTL 模型检验问题。

定理 5.20 对于任意的公平迁移系统 $\mathcal{M}$ 以及 ATL_r 公式 φ，则 $\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。

5.5.3 基于 BDD 的 $\mathbf{ATL}_r$ tableau 编码

在上一小节，介绍了从 ATL_r 的模型检验到 CTL 的模型检验转化过程。现在介绍 ATL_r 公式 tableau 的 BDD 编码方法。

给定 ATL_r 公式 φ，设 $[\psi_1]_\varphi^\sim$，...，$[\psi_m]_\varphi^\sim$ 是其所有自动机子公式由关系 $\sim$ 所划分的等价类，其中 ψ_j 的自动机连接子为 $\mathcal{A}_j = \langle \Sigma_j, Q_j, \delta_j, q_j, F_j \rangle$，并且 $\psi_j = \mathcal{A}_j(\varphi_{j,1}, \ldots, \varphi_{j,\#\Sigma_j})$。于是，$\mathcal{T}_\varphi = \langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi \rangle$ 的各要素编码过程如下。

位变元集合：参与编码的位变元包括如下各项：

1. 对每个 $\psi \in \mathbf{El}(\varphi)$，引入一个位变元 z_ψ；
2. 对每个 $1 \le j \le m$ 以及 $q \in Q_j$ 引入一个位变元 $u_{j,q}$；
3. 对每个 $1 \le j \le m$，$q \in Q_j$，$n \in \mathbb{N}[2 \times (\#Q_j) + 1]$ 以及 $\pm \in \{+, -\}$，引入一个位变元 $v_{j,q,n,\pm}$。

迁移关系：迁移关系 Φ_{Δ_φ} 按照如下方式获得。

首先对于每个 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，构建 $\mathbf{Sat}_\varphi(\psi)$ 之布尔表示 ϑ_ψ。

- $\vartheta_{true} = true$，$\vartheta_{false} = false$。
- 对于每个 $p \in AP$，若 $\psi = p$，则 $\vartheta_\psi = z_p$；若 $\psi = \neg p$，则 $\vartheta_\psi = \neg z_p$。
- 若 $\psi = \bigcirc\psi'$，则令 $\vartheta_\psi = z_\psi$。
- 若 $\psi = \psi_1 \wedge \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \wedge \vartheta_{\psi_2}$；若 $\psi = \psi_1 \vee \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \vee \vartheta_{\psi_2}$。
- 若 $\psi = \psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$）：则令 $\vartheta_\psi = \bigvee_{1 \le k \le \#\Sigma_j} (\vartheta_{\varphi_{j,k}} \wedge \vartheta_{\mathscr{I}_\psi^+(\delta_j(q,a_{j,k}))})$。
- 若 $\psi = \neg\psi_j^q$（其中 $1 \le j \le m$，$q \in Q_j$）：则令 $\vartheta_\psi = \neg\vartheta_{\psi_j^q}$。

事实上，上述过程与 ATL_l tableau 中的做法完全相同。同样容易证明：对于任意的 $\Gamma \subseteq \mathbf{El}(\varphi)$，$\Gamma \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 ϑ_ψ 在变元指派 e_Γ 下真值为 1。其中

e_Γ 满足：对于每个 $\phi \in \mathbf{El}(\varphi)$，$e_\Gamma(z_\phi) = 1$ 当且仅当 $\phi \in \Gamma$。

接下来，令 ϑ'_ψ 是将 ϑ_ψ 中的每个位变元换成其次态版本所得之布尔公式。同时，对每个 $\theta \in \mathbf{B}^+(Q_j)$，令 $\mathbf{U}_j(\theta)$（resp. $\mathbf{U}_j(\theta)$）为将 θ 中的每个 q 替换为 $u_{j,q}$（resp. $u'_{j,q}$）所得之布尔公式。对每个 $\theta \in \mathbf{B}^+(Q_j \times \mathbb{N} \times \{+,-\})$，令 $\mathbf{V}_j(\theta)$（resp. $\mathbf{V}'_j(\theta)$）为将 θ 中的每个 $(q,n,\pm)$ 替换为 $v_{j,q,n,\pm}$（resp. $v'_{j,q,n,\pm}$）所得之布尔公式。于是，Φ_{Δ_φ} 就是下列各项的合取。

$$\bigwedge_{\bigcirc\psi \in \mathbf{El}(\varphi)} z_{\bigcirc\psi} \leftrightarrow \vartheta'_\psi \tag{5.17}$$

$$\bigwedge_{1\le j\le m} \Big(\bigwedge_{q\in Q_j} (u_{j,q} \to \vartheta_{\psi_j^q}) \Big) \tag{5.18}$$

$$\bigwedge_{1\le j\le m} \Big(\big(\bigvee_{q\in Q_j} u_{j,q} \big) \to \bigwedge_{q\in Q_j\setminus F_j} \big(u_{j,q} \to \bigvee_{1\le k\le \#\Sigma_j} (\vartheta_{\varphi_{j,k}} \wedge \mathbf{U}'_j(\delta_j(q,a_{j,k})))\big)\Big) \tag{5.19}$$

$$\bigwedge_{1\le j\le m} \Big(\big(\bigwedge_{q\in Q_j} \neg u_{j,q} \big) \to \bigwedge_{q\in Q_j} (u'_{j,q} \leftrightarrow \vartheta'_{\psi_j^q}) \Big) \tag{5.20}$$

$$\bigwedge_{1\le j\le m} \bigwedge_{\substack{q\in F_j,\\ n\in\mathbb{O}[2\times(\#Q_j)+1]}} (\neg v_{j,q,n,+} \wedge \neg v_{j,q,n,-}) \tag{5.21}$$

$$\bigwedge_{j,q,n,\pm} \big(v_{j,q,n,\pm} \to \bigwedge_{1\le k\le \#\Sigma_j} (\vartheta_{j,k} \to \mathbf{V}'_j(\Lambda_{F_j}(\overline{\delta_j(q,a_{j,k})}, n, \pm)))\big) \tag{5.22}$$

$$\Big(\bigwedge_{j,q,n} \neg v_{j,q,n,-} \Big) \to \Big(\bigwedge_{j,q,n} \neg v'_{j,q,n,+} \wedge \bigwedge_{q\in Q_j} \big(\vartheta'_{\psi_j^q} \to \bigvee_{n\in\mathbb{N}[2\times(\#Q_j)+1]} v'_{j,q,n,-}\big)\Big) \tag{5.23}$$

对 $\mathcal{T}_\varphi$ 中的状态 $s = \langle \Gamma, (P_1,\ldots,P_m), (\Upsilon_1,\ldots,\Upsilon_m)\rangle$ 及 $s' = \langle \Gamma', (P'_1,\ldots,P'_m), (\Upsilon'_1, \ldots, \Upsilon'_m)\rangle$，令 e_s、$e_{s'}$ 是满足如下约束的两个变元指派。

- 对每个 $\phi \in \mathbf{El}(\varphi)$ 有：$e_s(z_\phi) = 1$ 当且仅当 $\phi \in \Gamma$；$e_{s'}(z'_\phi) = 1$ 当且仅当 $\phi \in \Gamma'$。
- 对每个 $1 \le j \le m$ 以及每个 $q \in Q_j$ 有：$e_s(u_{j,q}) = 1$ 当且仅当 $q \in P_j$；$e_{s'}(u'_{j,q}) = 1$ 当且仅当 $q \in P'_j$。
- 对每个 $1 \le j \le m$，$q \in Q_j$，$n \in \mathbb{N}[2\times(\#Q_j)+1]$ 以及每个 $\pm \in \{+,-\}$ 有：$e_s(v_{j,q,n,\pm}) = 1$ 当且仅当 $(q,n,\pm) \in \Upsilon_j$；$e_{s'}(v_{j,q,n,\pm}) = 1$ 当且仅当 $(q,n,\pm) \in \Upsilon'_j$。

这时，很容易验证下列性质：

1. 公式 (5.17) 在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1 当且仅当对每个 $\bigcirc\psi \in \mathbf{El}(\varphi)$ 有：$\Gamma \in \mathbf{Sat}_\varphi(\bigcirc\psi)$ 当且仅当 $\Gamma' \in \mathbf{Sat}_\varphi(\psi)$。

2. 公式 (5.18) 至公式 (5.20) 的合取在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1 当且仅当对每个 $1 \leq j \leq m$ 有 $((\Gamma, P_j), (\Gamma', P'_j)) \in \Delta^{r+}_{[\psi_j]^{\sim}_{\varphi}}$。
3. 公式 (5.21) 至公式 (5.23) 的合取在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1 当且仅当对每个 $1 \leq j \leq m$ 有 $((\Gamma, \Upsilon_j), (\Gamma', \Upsilon'_j)) \in \Delta^{r-}_{[\psi_j]^{\sim}_{\varphi}}$。

换言之，$(s, s') \in \Delta_\varphi$ 当且仅当上述各式在 e_s 和 $e_{s'}$ 的联合指派下的真值为 1。因此，Φ_{Δ_φ} 是 Δ_φ 的一个布尔编码。

初始状态集：初始状态集的布尔编码 Φ_{I_φ} 仍然为 ϑ_φ。

标记函数：对每个 $p \in AP \cap \mathbf{El}(\varphi)$，仍令 $\Phi^p_{\lambda_\varphi} = z_p$。

公平性约束：由于 $\mathcal{C}_\varphi = \{C^+_1, \ldots, C^+_m\} \cup \{C^-_1, \ldots, C^-_m\}$，其中 $C^+_j = \{\langle \Gamma, (P_1, \ldots, P_m), (\Upsilon_1, \ldots, \Upsilon_m)\rangle \mid P_j = \emptyset\}$，而 $C^-_j = \{\langle \Gamma, (P_1, \ldots, P_m), (\Upsilon_1, \ldots, \Upsilon_m)\rangle \mid \Upsilon_j \subseteq Q_j \times \mathbb{N} \times \{+\}\}$。所以 $\Phi_{\mathcal{C}_\varphi} = \{\Phi_{C^+_1}, \ldots, \Phi_{C^+_m}\} \cup \{\Phi_{C^-_1}, \ldots, \Phi_{C^-_m}\}$，其中：$\Phi_{C^+_j} = \bigwedge_{q \in Q_j} \neg u_{j,q}$，$\Phi_{C^-_j} = \bigwedge_{q \in Q_j} \bigwedge_{n \in \mathbb{N}[2\times(\#Q_j)+1]} \neg v_{j,q,n-}$。

容易看出：对于上述的 ATL_r 公式 φ，对其进行符号化编码所需的位变元数目为

$$\#\mathbf{El}(\varphi) + \sum_{1 \leq j \leq m} (4 \times (\#Q_j)^2 + 5 \times (\#Q_j)) \tag{5.24}$$

而这个数目是属于 $\mathcal{O}(|\varphi|^2)$ 量级的。

5.6 APSL 符号化模型检验

5.6.1 PSL 的变种：APSL

PSL[31] 是 ETL 的一个特例，目前已经成为工业界的规约语言标准（IEEE-1850）。该种逻辑既包含分支时间的语法成分（OBE 公式），又包含线性时间的语法成分（FL 公式）。OBE 公式本质上是 CTL 公式，而 FL 公式的表达能力等价于 ω-正规语言。除内置的时序连接子外，FL 公式中还采用 SERE 作为公式的构造子（FL 公式及 SERE 语法语义定义见 2.2.4 节）。

在 2.2.4 节中，曾经介绍过 PSL 的线性部分— FL 的语法、语义。现在说明任意的 ω-正规性质都能被 FL 公式表达。

引理 5.21 *对于每个以 2^{AP} 为字母表的 ω-正规语言 R，都存在某个不含 abort 连接子的 FL 公式 φ_R，使得对每个线性结构 π 都有：$\pi \in \mathbf{L}(R)$ 当且仅当 $\pi \models \varphi_R$。*

证明. 首先证明：存在 $m \in \mathbb{N}$ 以及（以 2^{AP} 为字母表的）有穷正规语言 $r_1, r'_1, \ldots, r_m, r'_m$ 使得

$$\mathbf{L}(R) = \bigcup_{1 \le k \le m} \mathbf{L}(r_k; (r'_k)^\omega) \tag{5.25}$$

成立。该证明非常容易：由于 R 是（字母表 2^{AP} 上的）ω-正规语言，所以存在非确定 Büchi 自动机 $\mathcal{A}_R = \langle 2^{AP}, Q, \delta, q_0, \{q_1, \ldots, q_m\}\rangle$ 使得 $\mathbf{L}(R) = \mathbf{L}(\mathcal{A}_R)$。因此，只需令 r_k 为有穷字上的自动机 $\mathcal{A}_i = \langle 2^{AP}, Q, \delta, q_0, \{q_k\}\rangle$ 所对应的有穷正规语言，而令 r'_k 为有穷字上的自动机 $\mathcal{A}_k = \langle 2^{AP}, Q, \delta, q_k, \{q_k\}\rangle$ 所对应的有穷正规语言即可。

由定义式 (2.25) 以及定义式 (2.26) 可得："对于任意的线性结构 π 以及 $i \in \mathbb{N}$，$\pi, i \models r!$ 当且仅当存在 $j \ge i$ 使得 $\pi[i, j] \in \mathbf{L}(r)$"。于是，令

$$\varphi_R = \bigvee_{1 \le k \le m} (r_k! \wedge ((r_k; (r'_k)^*) \ \mathbf{trigger}\ (true; r'_k!)))$$

即可。这是因为：$\pi \models \varphi_R$ 当且仅当存在 $1 \le k \le m$ 以及 $i_0 \le i_1 \le i_2 \le \ldots$ 使得 $\pi[0, i_0] \in \mathbf{L}(r_k)$ 以及对每个 l 有 $\pi[i_j + 1, i_{j+1}] \in \mathbf{L}(r'_k)$。这当且仅当 $\pi \in \mathbf{L}(R)$。 □

使 SERE 作为公式构造子，虽然非常灵活简洁，但是却不利于创建公式的 tableau。此外，abort 连接子的存在也为构建公式 tableau 带来了一定的障碍。为了更好的使用符号化算法对这种规约语言进行检验，文 [116, 117]中提出了 PSL 的一种变种，称之为 APSL。APSL 的线性成分对应的称为 AFL，其分支部分仍为 OBE。同标准的 FL 相比，AFL 使用有穷字上的自动机来代替 SERE 作为公式构造子。这样，便可以相对容易的给出公式 tableau 的定义。此外，AFL 中严格限制了 abort 算子的使用— 该算子的第一个操作子必须是有穷自动机。

为方便起见，在本节，认为 AP 是一个有穷集（事实上，可以将公式和模型中出现的原子命题取作 AP）。同时，将遵循 PSL 中的书写习惯：用 b、b_1、b_2、... 等表示 AP 上的布尔公式；用 r、r_1、r_2、... 等表示 SERE 表达式。在本节中所使用的有穷字上的自动机的迁移结构都是非确定的，且均只具有一个初始状态（事实上，任何多初始状态的有穷字上的自动机均可化为只有一个初始状态的自动机）。因此，有穷字自动机的运行可以看做是一个状态序列。此外，如非特别说明，有穷字上的自动机的字母表均为 2^{AP}。此外，假设每个有穷字上的自动机 $\mathcal{A}$ 都是**精化的**。即：$\mathcal{A}$ 中的每个状态都能到达某个接收状态。这种假设是合理的— 因为若 $\mathcal{A}$ 中某状态不能到达任何接收状态，则将其删除后 $\mathbf{L}(\mathcal{A})$ 值不发生改变。容易证明：对于精化的自动机 $\mathcal{A}$ 而言，一定有 $\mathbf{L}(\mathcal{A}) \neq \emptyset$。

下面给出 APSL 的线性部分— AFL 的语法、语义定义。

定义 5.6.1 (AFL 语法) *AFL* 的**合式公式**归纳定义如下:

- 若 $\psi = b \in \mathbf{B}(AP)$, 则 ψ 是 *AFL* 公式。
- 若 ψ 是 *AFL* 公式, 则 $\neg\psi$ 是 *AFL* 公式。
- 若 φ_1、φ_2 都是 *AFL* 公式, 则 $\varphi_1 \wedge \varphi_2$ 是 *AFL* 公式。
- 若 ψ 是 *AFL* 公式, 则 $\mathsf{X}\psi$ 是 *AFL* 公式。
- 若 φ_1、φ_2 都是 *AFL* 公式, 则 $\varphi_1 \mathsf{U} \varphi_2$ 是 *AFL* 公式。
- 若 $\mathcal{A}$ 是有穷字上的自动机, $b \in \mathbf{B}(AP)$, 则 $\mathcal{A}$ abort! b 是 *AFL* 公式。
- 若 ψ 是 *AFL* 公式, $\mathcal{A}$ 是有穷字上的自动机, 则 $\mathcal{A}$ trigger ψ 是 *AFL* 公式。□

应当注意的是：在 AFL 中使用的是 abort 算子的"加强版本" abort!。在标准 FL 中，ψ abort b 等价于 $\psi \vee (\psi$ abort! $b)$。

定义 5.6.2 (AFL 语义) 给定线性结构 π, 位置 $i \in \mathbb{N}$, 以及 *FL* 公式 φ, 则可归纳定义**满足关系** $\models$ 如下。

- 若 $\varphi = b \in \mathbf{B}(AP)$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi(i)$ 满足 b。
- 若 $\varphi = \neg\psi$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \not\models \psi$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i \models \varphi_1$ 且 $\pi, i \models \varphi_2$。
- 若 $\varphi = \mathsf{X}\psi$, 则 $\pi, i \models \varphi$ 当且仅当 $\pi, i+1 \models \psi$。
- 若 $\varphi = \varphi_1 \mathsf{U} \varphi_2$, 则 $\pi, i \models \varphi$ 当且仅当存在 $k \geq i$, 使得 $\pi, k \models \varphi_2$, 且对于任意的 $i \leq j < k$, 都有 $\pi, j \models \varphi_1$。
- 若 $\varphi = \mathcal{A}$ abort! b, 则 $\pi, i \models \varphi$ 当且仅当存在 $k \geq j \geq i$ 以及线性结构 π' 使得 $\pi[0, j-1] = \pi'[0, j-1]$, $\pi(j)$ 满足 b, 并且 $\pi'[i, k] \in \mathbf{L}(\mathcal{A})$。
- 若 $\varphi = \mathcal{A}$ trigger ψ, 则 $\pi, i \models \varphi$ 当且仅当对于任意的 $j \geq i$, 若 $\pi[i, j] \in \mathbf{L}(\mathcal{A})$, 则 $\pi, j \models \psi$。

特别的, 当 $i = 0$ 时, 直接将 $\pi, i \models \varphi$ 写作 $\pi \models \varphi$。 □

引理 5.22 对每个 *SERE* 表达式 r 而言, 若 $\mathbf{L}(r) \neq \emptyset$, 则存在某个有穷字上的自动机 $\mathcal{A}_r$, 使得 $\mathbf{L}(\mathcal{A}_r) = \mathbf{L}(r)$。

证明. 根据 r 的结构归纳构造即可。

- 若 $r = 0[*]$，则令 $\mathcal{A}_r = \langle 2^{AP}, \{q\}, \delta_\epsilon, q, \{q\}\rangle$。其中，对每个 $a \subseteq AP$，$\delta_\epsilon(q, a) = \emptyset$。
- 若 $r = b \in \mathbf{B}(AP)$，则令 $\mathcal{A}_r = \langle 2^{AP}, \{q_1, q_2\}, \delta_b, q_1, \{q_2\}\rangle$。其中，对每个 $a \subseteq AP$ 有

$$\delta_b(q_1, a) = \begin{cases} q_2 & , a \models b \\ \emptyset & , a \not\models b \end{cases},$$

以及 $\delta(q_2, a) = \emptyset$。

- 若 $r = (r')^*$，且 $\mathcal{A}_{r'} = \langle 2^{AP}, Q, \delta', q_0, F\rangle$，则令 $\mathcal{A}_r = \langle 2^{AP}, Q, \delta, q_0, F \cup \{q_0\}\rangle$。其中，对每个 $q \in Q$ 以及 $a \subseteq AP$ 有

$$\delta(q,a) = \begin{cases} \delta'(q,a) & ,q \notin F \\ \delta'(q,a) \cup \delta'(q_0,a) & ,q \in F \end{cases} 。$$

- 若 $r = r_1|r_2$，且 $\mathcal{A}_{r_i} = \langle 2^{AP}, Q_i, \delta_i, q_i, F_i\rangle$（这里 $i = 1,2$，$Q_1 \cap Q_2 = \emptyset$，下同），则令 $\mathcal{A}_r = \langle 2^{AP}, Q_1 \cup Q_2, \delta, q_0, F_1 \cup F_2\rangle$。其中 $q_0 \notin Q_1 \cup Q_2$，对每个 $q \in Q_1 \cup Q_2 \cup \{q_0\}$ 以及 $a \subseteq AP$ 有

$$\delta(q,a) = \begin{cases} \delta_1(q_1,a) \cup \delta_2(q_2,a) & ,q = q_0 \\ \delta_1(q,a) & ,q \in Q_1 \\ \delta_2(q,a) & ,q \in Q_2 \end{cases} 。$$

- 若 $r = r_1 \& r_2$，且 $\mathcal{A}_{r_i} = \langle 2^{AP}, Q_i, \delta_i, q_i, F_i\rangle$，则令 $\mathcal{A}_r = \langle 2^{AP}, Q_1 \times Q_2, \delta, (q_1,q_2), F_1 \times F_2\rangle$。其中对每个 $(q,q') \in Q_1 \times Q_2$ 以及 $a \subseteq AP$ 有

$$\delta((q,q'),a) = \{(q'',q''') \mid q'' \in \delta_1(q,a), q''' \in \delta_2(q',a)\}。$$

- 若 $r = r_1; r_2$，且 $\mathcal{A}_{r_i} = \langle 2^{AP}, Q_i, \delta_i, q_i, F_i\rangle$，则令 $\mathcal{A}_r = \langle 2^{AP}, Q_1 \cup Q_2, \delta, q_1, F_2\rangle$。其中对每个 $q \in Q_1 \cup Q_2$ 以及 $a \subseteq AP$ 有

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & ,q \in Q_1 \backslash F_1 \\ \delta_1(q,a) \cup \delta_2(q_2,a) & ,q \in F_1 \\ \delta_2(q,a) & ,q \in Q_2 \end{cases} 。$$

- 若 $r = r_1 : r_2$，且 $\mathcal{A}_{r_i} = \langle 2^{AP}, Q_i, \delta_i, q_i, F_i\rangle$，则令 $\mathcal{A}_r = \langle 2^{AP}, Q_1 \cup Q_2, \delta, q_1, F_2\rangle$。其中对每个 $q \in Q_1 \cup Q_2$ 以及 $a \subseteq AP$ 有

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & ,q \in Q_1, \delta_1(q,a) \cap F_1 = \emptyset \\ \delta_1(q,a) \cup \delta_2(q_2,a) & ,q \in Q_1, \delta_1(q,a) \cap F_1 \neq \emptyset \\ \delta_2(q,a) & ,q \in Q_2 \end{cases} 。$$

□

由引理 5.21，引理 5.22，以及 FL 公式 $r!$ 和 r trigger ψ 分别等价于 AFL 公式 $\mathcal{A}_r$ leads $true$ 和 $\mathcal{A}_r$ trigger ψ（其中 $\mathcal{A}_r$ 是由 r 得到的有穷字上的自动机）这个事实，立即可以得到如下定理。

定理 5.23 对于每个以 2^{AP} 为字母表的 ω-正规语言 R，都存在某个 AFL 公式 φ_R，使得对每个线性结构 π 都有：$\pi \in \mathbf{L}(R)$ 当且仅当 $\pi \models \varphi_R$。

在文 [118] 以及 [61] 中证明了加入 abort 算子（包括其限制性版本 abort!）后不会增强 FL 的表达能力。因此，立即有如下推论。

推论 5.24 (FL 和 AFL 的等价性) *FL 公式和 AFL 公式（在无穷模型上）的表达能力相同，都等价于 ω-正规语言。*

同样，类似与公式 (2.24) 及公式 (2.25)，在 AFL 中存在如下派生算子。

$$\varphi_1 \mathsf{R} \varphi_2 \stackrel{\text{def}}{=} \neg(\neg\varphi_1 \mathsf{U} \neg\varphi_2) \tag{5.26}$$

$$\mathcal{A} \text{ monitor } b \stackrel{\text{def}}{=} \neg(\mathcal{A} \text{ abort!} \neg b) \tag{5.27}$$

$$\mathcal{A} \text{ leads } \psi \stackrel{\text{def}}{=} \neg(\mathcal{A} \text{ trigger} \neg\psi) \tag{5.28}$$

对这些派生连接子，根据定义容易证明有如下的直观语义。

引理 5.25 *对于任意的线性结构 π 以及 $i \in \mathbb{N}$，则：*

- $\pi, i \models \varphi_1 \ \mathsf{R}\ \varphi_2$ 当且仅当：或者对于每个 $j \geq i$ 有 $\pi, i \models \varphi_2$；或者存在某个 $k \geq i$ 使得 $\pi, k \models \varphi_1$，并且对每个 $i \leq j \leq k$ 有 $\pi, j \models \varphi_2$。
- $\pi, i \models \mathcal{A}$ monitor b 当且仅当对于任意的 $k \geq j \geq i$ 以及线性结构 π'，若 $\pi[0, j-1] = \pi'[0, j-1]$ 且 $\pi'[i, k] \in \mathbf{L}(\mathcal{A})$，则 $\pi(j)$ 满足 b。
- $\pi, i \models \mathcal{A}$ leads ψ 当且仅当存在 $j \geq i$ 使得 $\pi[i, j] \in \mathbf{L}(\mathcal{A})$ 且 $\pi, j \models \psi$。

定义 5.6.3 (AFL 公式否定范式) 对于任意的 AFL 公式 φ，利用德摩根律、模式 $\neg\neg\varphi \leftrightarrow \varphi$、$\neg\mathsf{X}\varphi \leftrightarrow \mathsf{X}\neg\varphi$ 以及上述派生算子的定义，可以将布尔连接子 $\neg$ 内移，使之仅出现在原子命题之前。这样得到的公式称为公式的**否定范式**。 □

例 5.6.1 设 $\varphi = \neg(\mathsf{X}(\mathcal{A}_1\mathsf{abort!}(p_1 \wedge \neg p_2))) \vee \neg(\mathcal{A}_2\mathsf{trigger}(p_2 \mathsf{U} \neg p_3))$，则 φ 的否定范式为 $\mathsf{X}(\mathcal{A}_1\mathsf{monitor}(\neg p_1 \wedge p_2)) \vee \mathcal{A}_2\mathsf{leads}(\neg p_2 \mathsf{R} p_3)$。 □

AFL 中的连接子 abort!、monitor、trigger 和 leads 的第一个操作子要求是自动机。关于这类连接子，有如下定理成立。

引理 5.26 *设有穷字上的自动机 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$，则对于任意的线性结构 π 以及 $i \in \mathbb{N}$：*

1. $\pi, i \models \mathcal{A}$ abort! b 当且仅当 $\pi, i \models b \vee \bigvee_{q' \in \delta(q, \pi(i))} \mathsf{X}(\mathcal{A}^{q'}\text{abort! } b)$。
2. $\pi, i \models \mathcal{A}$ monitor b 当且仅当 $\pi, i \models b \wedge \bigwedge_{q' \in \delta(q, \pi(i))} \mathsf{X}(\mathcal{A}^{q'}\text{monitor } b)$。
3. $\pi, i \models \mathcal{A}$ trigger ψ 当且仅当
   - 若 $\delta(q, \pi(i)) \cap F \neq \emptyset$ 则 $\pi, i \models \psi$；并且
   - $\pi, i \models \bigwedge_{q' \in \delta(q, \pi(i))} \mathsf{X}(\mathcal{A}^{q'}\text{trigger } \psi)$。

4. $\pi, i \models \mathcal{A}$ leads ψ 当且仅当
   - $\delta(q, \pi(i)) \cap F \neq \emptyset$ 且 $\pi, i \models \psi$; 或者
   - $\pi, i \models \bigvee_{q' \in \delta(q,\pi(i))} \mathsf{X}(\mathcal{A}^{q'}\text{leads } \psi)$。

上述定理是 5.6.2 节中确定公式满足集函数（见定义 5.6.5）以及构建 AFL 公式 tableau 的基础。

5.6.2 AFL 公式的 tableau

由于在 APSL 公式中，OBE 公式实质上就是 CTL 公式，因而在研究 APSL 公式的符号化模型检验算法时，主要关注 AFL 公式的符号化模型检验算法。本节将定义 AFL 公式的 tableau 的定义，研究其语言性质，以及给出从 AFL 模型检验到 CTL 模型检验的转化过程。同时，假设所有的 AFL 公式已都写为否定范式。

定义 5.6.4 (AFL 公式的基础公式集合) 给定 *AFL* 公式 φ，归纳定义 φ 的**基础公式集** $\mathbf{El}(\varphi)$ 如下:

- 若 $\varphi = b \in \mathbf{B}(AP)$，则 $\mathbf{El}(\varphi) = AP$。
- 若 $\varphi = \varphi_1 \wedge \varphi_2$ 或者 $\varphi = \varphi_1 \vee \varphi_2$，则 $\mathbf{El}(\varphi) = \mathbf{El}(\varphi_1) \cup \mathbf{El}(\varphi_2)$。
- 若 $\varphi = \mathsf{X}\varphi'$，则 $\mathbf{El}(\varphi') \cup \{\varphi\}$。
- 若 $\varphi = \varphi_1 \ \mathsf{U}\ \varphi_2$ 或 $\varphi = \varphi_1 \ \mathsf{R}\ \varphi_2$，则 $\mathbf{El}(\varphi) = \mathbf{El}(\varphi_1) \cup \mathbf{El}(\varphi_2) \cup \{\mathsf{X}\varphi\}$。
- 设 $\mathcal{A}$ 是以 Q 为状态集的有穷字上的自动机，b 是布尔公式。若 $\varphi = \mathcal{A}$ abort! b，则 $\mathbf{El}(\varphi) = AP \cup \{\mathsf{X}(\mathcal{A}^q\text{abort! } b) \mid q \in Q\}$；若 $\varphi = \mathcal{A}$ monitor b，则 $\mathbf{El}(\varphi) = AP \cup \{\mathsf{X}(\mathcal{A}^q\text{monitor } b) \mid q \in Q\}$。
- 设 $\mathcal{A}$ 是以 Q 为状态集的有穷字上的自动机。若 $\varphi = \mathcal{A}$ trigger ψ，则 $\mathbf{El}(\varphi) = \mathbf{El}(\psi) \cup \{\mathsf{X}(\mathcal{A}^q \text{ trigger } \psi) \mid q \in Q\}$；若 $\varphi = \mathcal{A}$ leads ψ，则 $\mathbf{El}(\varphi) = \mathbf{El}(\psi) \cup \{\mathsf{X}(\mathcal{A}^q \text{ leads } \psi) \mid q \in Q\}$。 □

容易看出，$\mathbf{El}(\varphi)$ 中的公式或者是原子命题，或者形如 $\mathsf{X}\psi$。对每个 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，也可以定义其**满足集**函数 $\mathbf{Sat}_\varphi$。

定义 5.6.5 (AFL 公式的满足集函数) 对于任意的 *AFL* 公式 φ，可以定义函数 $\mathbf{Sat}_\varphi : \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi) \to 2^{2^{\mathbf{El}(\varphi)}}$ 如下。

- 若 $\psi \in AP$ 或者 ψ 形如 $\mathsf{X}\psi'$，则 $\mathbf{Sat}_\varphi(\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \psi \in \Gamma\}$;
  若 $\psi \in \overline{AP}$，则 $\mathbf{Sat}_\varphi(\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \psi \notin \Gamma\}$。
- 若 $\psi = \psi_1 \wedge \psi_2$，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(\psi_1) \cap \mathbf{Sat}_\varphi(\psi_2)$;
  若 $\psi = \psi_1 \vee \psi_2$，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(\psi_1) \cup \mathbf{Sat}_\varphi(\psi_2)$。

- 若 $\psi = \psi_1 \mathsf{U} \psi_2$，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(\psi_2) \cup (\mathbf{Sat}_\varphi(\psi_1) \cap \mathbf{Sat}_\varphi(\mathsf{X}(\psi_1 \mathsf{U} \psi_2)))$;
  若 $\psi = \psi_1 \mathsf{R} \psi_2$，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(\psi_2) \cap (\mathbf{Sat}_\varphi(\psi_1) \cup \mathbf{Sat}_\varphi(\mathsf{X}(\psi_1 \mathsf{R} \psi_2)))$。
- 设 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F \rangle$。
  若 $\psi = \mathcal{A}$ abort! b，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(b) \cup \bigcup_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'} \text{abort! } b))$;
  若 $\psi = \mathcal{A}$ monitor b，则 $\mathbf{Sat}_\varphi(\psi) = \mathbf{Sat}_\varphi(b) \cap \bigcap_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'} \text{monitor } b))$。
- 设 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F \rangle$。
  若 $\psi = \mathcal{A}$ trigger ψ'，则 $\mathbf{Sat}_\varphi(\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid$ 若 $\delta(q, \Gamma \cap AP) \neq \emptyset$, 则 $\Gamma \in \mathbf{Sat}_\varphi(\psi')\} \cap \bigcap_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'} \text{trigger } \psi'))$;
  若 $\psi = \mathcal{A}$ leads ψ'，则 $\mathbf{Sat}_\varphi(\psi) = \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \delta(q, \Gamma \cap AP) \neq \emptyset$, 且 $\Gamma \in \mathbf{Sat}_\varphi(\psi')\} \cup \bigcup_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'} \text{leads } \psi'))$。 □

定义 5.6.6 (牵引对及其派生迁移关系) 给定 AFL 公式 φ（否定范式），称 $\langle \mathcal{A}, \psi \rangle$（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F \rangle$）是 φ 中的一个**牵引对**，如果存在 $q \in Q$，使得 $\mathcal{A}^q \text{leads } \psi$ 是出现在 φ 中的子公式。

对 φ 中的牵引对 $\langle \mathcal{A}, \psi \rangle$，可以派生一个迁移关系 $\Delta_{\langle \mathcal{A}, \psi \rangle} \subseteq (2^{\mathbf{El}(\varphi)} \times 2^Q) \times (2^{\mathbf{El}(\varphi)} \times 2^Q)$。其中，$((\Gamma, P), (\Gamma', P')) \in \Delta_{\langle \mathcal{A}, \psi \rangle}$ 当且仅当:

- 若 $P \neq \emptyset$，则对于任意的 $q \in P$ 而言，或者 $\Gamma \in \mathbf{Sat}_\varphi(\psi)$ 且 $\delta(q, \Gamma \cap AP) \neq \emptyset$; 或者有某个 $q' \in P'$，且使得 $q' \in \delta(q, \Gamma \cap AP)$。
- 若 $p = \emptyset$，则 $P' = \{q \in Q \mid \Gamma' \in \mathbf{Sat}_\varphi(\mathcal{A}^q \text{leads } \psi)\}$。 □

定义 5.6.7 (AFL 公式的 tableau) 给定 AFL 公式 φ，设 $\langle \mathcal{A}_1, \psi_1 \rangle$，…，$\langle \mathcal{A}_m, \psi_m \rangle$ 是 φ 中出现的所有牵引对，且 $\mathcal{A}_i$ 的状态集合为 Q_i。则 φ 的 *tableau* $\mathcal{T}_\varphi$ 是公平迁移系统$\langle S_\varphi, \Delta_\varphi, I_\varphi, \lambda_\varphi, \mathcal{C}_\varphi \rangle$。其中:

- $S_\varphi = \{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid \Gamma \subseteq \mathbf{El}(\varphi), P_i \subseteq Q_i\}$。
- $(\langle \Gamma, (P_1, \ldots, P_m) \rangle, \langle \Gamma', (P_1', \ldots, P_m') \rangle) \in \Delta_\varphi$ 当且仅当
  - 对每个 $\mathsf{X}\psi \in \mathbf{El}(\varphi)$，$\Gamma \in \mathbf{Sat}_\varphi(\mathsf{X}\psi)$ 当且仅当 $\Gamma' \in \mathsf{Sat}_\varphi(\psi)$。
  - 对每个 $1 \leq i \leq m$，$((\Gamma, P_i), (\Gamma', P_i')) \in \Delta_{\langle \mathcal{A}_i, \psi_i \rangle}$。
- $I_\varphi = \{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\varphi)\}$。
- 对每个 $\langle \Gamma, (P_1, \ldots, P_m) \rangle \in S_\varphi$，$\lambda_\varphi(\langle \Gamma, (P_1, \ldots, P_m) \rangle) = \Gamma \cap AP$。
- $\mathcal{C}_\varphi$ 包括如下的公平约束:
  - 对 φ 中每个子公式 $\phi_1 \mathsf{U} \phi_2$ 添加一个公平性约束 $\{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\phi_2)$ 或 $\Gamma \notin \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)\}$。
  - 对 φ 中的每个子公式 $\mathcal{A}$ abort!b 添加一个公平性约束 $\{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid \Gamma \in \mathbf{Sat}_\varphi(b)$ 或 $\Gamma \notin \bigcup_{q \in Q} \mathbf{Sat}_\varphi(\mathcal{A}^q \text{abort!} b)\}$。其中，$Q$ 是 $\mathcal{A}$ 的状态集。

– 对 φ 中的每个牵引对 $\langle \mathcal{A}_i, \psi_i \rangle$，添加一个公平性约束 $\{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid P_i = \emptyset\}$。 □

AFL 公式 tableau 关心的原子命题集合为 AP（如上节约定，这里认为 AP 是一个有穷集合）下面证明 AFL 公式 tableau 的语言性质。

定理 5.27 对于任意 *AFL* 公式以及线性结构 π，若 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$，则 $\pi \models \varphi$。

证明. 设 $\langle \mathcal{A}_1, \psi_1 \rangle, \ldots, \langle \mathcal{A}_m, \psi_m \rangle$ 是 φ 中所有牵引对，其中 $\mathcal{A}_i = \langle 2^{AP}, Q_i, \delta_i, q_i, F_i \rangle$。设 π 是 $\mathcal{T}_\varphi$ 中公平展开迹 $\sigma = s_0, s_1, \ldots$ 的派生线性结构，其中 $s_i = \langle \Gamma_i, (P_{1,i}, \ldots, P_{m,i}) \rangle$。现在，用公式结构归纳法证明：对于任意的 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$ 以及 $i \in \mathbb{N}$，若 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 则 $\pi, i \models \psi$。

- 若 $\psi = p$（resp. $\psi = \neg p$），则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $p \in \pi(i)$（resp. $p \notin \pi(i)$）于是有 $\pi, i \models \psi$。
- 若 $\psi = \psi' \wedge \psi''$，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi') \cap \mathbf{Sat}_\varphi(\psi'')$。由归纳假设有 $\pi, i \models \psi'$ 且 $\pi, i \models \psi''$，因此 $\pi, i \models \psi$。
- 若 $\psi = \psi' \vee \psi''$，则 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi') \cup \mathbf{Sat}_\varphi(\psi'')$。由归纳假设有 $\pi, i \models \psi'$ 或 $\pi, i \models \psi''$，因此 $\pi, i \models \psi$。
- 若 $\psi = \mathsf{X}\psi'$，则 $\psi \in \mathbf{El}(\varphi)$。由 Δ_φ 的定义，$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi')$。由归纳假设有 $\pi, i+1 \models \psi'$，因此 $\pi, i \models \psi$。
- 若 $\psi = \phi_1 \mathsf{U} \phi_2$，则对于任意的 $j \geq i$ 有 $\Gamma_j \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_2)$ 或者 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_1) \cap \mathbf{Sat}_\varphi(\mathsf{X}(\phi_1 \mathsf{U} \phi_2))$。由 Δ_φ 的限制：$\Gamma_j \in \mathbf{Sat}_\varphi(\mathsf{X}(\phi_1 \mathsf{U} \phi_2))$ 当且仅当 $\Gamma_{j+1} \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)$。换言之，对于任意的 $j \geq i$，若 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)$ 且 $\Gamma_j \notin \mathbf{Sat}_\varphi(\phi_2)$ 则 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_1)$ 且 $\Gamma_{j+1} \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)$。
  由公平性约束 $\{\langle \Gamma, (P_1, \ldots, P_m) \rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\phi_2) \text{ 或 } \Gamma \notin \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)\}$ 以及 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)$ 可知，必然有某个 $k \geq i$ 使得 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi_2)$ 并且对每个 $i \leq j < k$ 有 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_1)$。根据归纳假设，有 $\pi, k \models \phi_2$ 且对每个 $i \leq j < k$ 有 $\pi, j \models \phi_1$。于是，$\pi, i \models \psi$。
- 若 $\psi = \phi_1 \mathsf{R} \phi_2$，则对于任意的 $j \geq i$，$\Gamma_j \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_2)$，并且 $\Gamma_j \in (\mathbf{Sat}_\varphi(\phi_1) \cup \mathbf{Sat}_\varphi(\mathsf{X}(\phi_1 \mathsf{R} \phi_2)))$。同时，由 Δ_φ 的限制知 $\Gamma_j \in (\mathbf{Sat}_\varphi(\mathsf{X}(\phi_1 \mathsf{R} \phi_2)))$ 当且仅当 $\Gamma_{j+1} \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{R} \phi_2)$。
  由于 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{R} \phi_2)$，重复上述讨论，可知必然有以下两种情形之一成立：或者对任意的 $j \geq i$ 有 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_2)$；或者存在某个 $k \geq i$ 使得 $\Gamma_k \in \mathbf{Sat}_\varphi(\phi_1)$ 并且对任意的 $i \leq j \leq k$ 有 $\Gamma_j \in \mathbf{Sat}_\varphi(\phi_1)$。于是，由归纳假设，或者对于每个 $j \geq i$ 有 $\pi, i \models \phi_2$；或者存在某个 $k \geq i$ 使得 $\pi, k \models \phi_1$，并且对每个 $i \leq j \leq k$

有 $\pi, j \models \phi_2$。由引理 5.25 有 $\pi, i \models \psi$。

- 若 $\psi = \mathcal{A}$ abort! b（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q_0, F\rangle$），假设对于任意的 $j \geq i$ 有 $\Gamma_j \notin \mathbf{Sat}_\varphi(b)$。那么，在此前提下，对任意的 $j \geq i$ 以及 $q \in Q$ 有：若 $\Gamma_j \in \mathbf{Sat}_\varphi(\mathcal{A}^q\text{abort! } b)$ 则存在某个 $q' \in \delta(q, \Gamma_j \cap AP)$ 使得 $\Gamma_j \in \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'}\text{abort! } b))$ 即 $\Gamma_{j+1} \in \mathbf{Sat}_\varphi(\mathcal{A}^{q'}\text{abort! } b)$。于是，对任意的 $j \geq i$ 有 $\Gamma_j \in \bigcup_{q\in Q} \mathbf{Sat}_\varphi(\mathcal{A}^q\text{abort! } b)$。因此，在这种前提下 σ 违反公平性约束 $\{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(b)$ 或 $\Gamma \notin \bigvee_{q\in Q} \mathbf{Sat}_\varphi(\mathcal{A}^{q'}\text{abort! } b)\}$。这与 σ 是 $\mathcal{T}_\varphi$ 中的公平展开迹相矛盾!

  于是，必然存在某个 $j \geq i$，使得 $\Gamma_j \in \mathbf{Sat}_\varphi(b)$（即：对每个 $i \leq l < j$，$\Gamma_l \notin \mathbf{Sat}_\varphi(b)$），并且存在 $q_0, q_1, \ldots, q_{j-i+1} \in Q^*$ 使得对每个 $0 \leq l \leq j - i$ 有 $q_{l+1} \in \delta(q_l, \Gamma_{i+l} \cap AP)$，以及 $\Gamma_{i+l} \in \mathbf{Sat}_\varphi(\mathcal{A}^{q_l}\text{abort!}b)$。由于 $\mathcal{A}^{q_{j-i}}$ 是精化的，因此存在 $w \in (2^{AP})^*$ 使得 $w \in \mathbf{L}(\mathcal{A}^{q_{j-i}})$。同时注意到 $\pi(i + l) = \Gamma_{i+l} \cap AP$，因此 $\pi[i, j - 1] \cdot w \in \mathbf{L}(\mathcal{A})$。

  令 $k = j + |w| - 1$，再令 π' 是任意一个满足："$\pi'[0, j - 1] = \pi[0, j - 1]$ 且对每个 $j \leq l \leq k$ 有 $\pi'(l) = w(l - j)$" 的线性结构。于是 $\pi'[i, k] = \pi[i, j - 1] \cdot w \in \mathbf{L}(\mathcal{A})$ 且 $\pi(j) \models b$（因为 $\Gamma_j \in \mathbf{Sat}_\varphi(b)$）。由定义，$\pi, i \models \psi$。

- 若 $\psi = \mathcal{A}$ monitor b（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$），则构造序列 $Q_0, Q_1, \ldots$ 如下：令 $Q_0 = \{q\}$，$Q_{l+1} = \bigcup_{q'\in Q_l} \delta(q', \Gamma_{i+l} \cap AP)$。由于 $Q_0 = \{q\}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\mathcal{A}^q$ monitor $b)$，$\Gamma_{i+1} \cap AP = \pi(i + 1)$ 以及 $\mathbf{Sat}_\varphi$ 函数的定义，不难证明：对于任意的 $l \in \mathbb{N}$ 以及 $q' \in Q_l$，有 $\Gamma_{i+l} \in \mathbf{Sat}_\varphi(\mathcal{A}^{q'}$ monitor $b)$ 成立。

  因此，对于任意的 $k \geq j \geq i$ 以及线性结构 π'，若 $\pi[0, j - 1] = \pi'[0, j - 1]$，且 $\pi[i, k] \in \mathbf{L}(\mathcal{A})$，则必然存在 $q_0, q_1, \ldots, q_{j-i}$，使得 $q_0 = q$，$q_l \in Q_{i+l}$ 并且 $q_{l+1} \in \delta(q_l, \pi(i + l))$（事实上，$q_0, q_1, \ldots, q_{j-i}$ 可看作是 $\pi[i, j - 1]$ 在 $\mathcal{A}$ 上的运行）。由前面的结论，$\Gamma_j \in \mathbf{Sat}_\varphi(\mathcal{A}^{q_{j-i}}\text{monitor } b) \subseteq \mathbf{Sat}_\varphi(b)$。因此，$\Gamma_j \in \mathbf{Sat}_\varphi(b)$，从而 $\pi, j \models b$。所以，由引理 5.25 可得 $\pi, i \models \psi$。

- 若 $\psi = \mathcal{A}$ trigger ψ'（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$），则同样构造序列 $Q_0, Q_1, \ldots$ 如下：令 $Q_0 = \{q\}$，$Q_{l+1} = \bigcup_{q'\in Q_l} \delta(q', \Gamma_{i+l} \cap AP)$。由 $Q_0 = \{q\}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\mathcal{A}^{q_0}$ trigger $\psi')$ 以及 $\mathbf{Sat}_\varphi$ 函数的定义，不难证明：对于任意的 $l \in \mathbb{N}$ 以及 $q' \in Q_l$，有 $\Gamma_{i+l} \in \mathbf{Sat}_\varphi(\mathcal{A}^{q'}$ trigger $\psi)$ 成立。并且，若 $Q_{l+1} \cap F \neq \emptyset$ 则 $\Gamma_{i+l} \in \mathbf{Sat}_\varphi(\psi')$。

  因此，对于任意的 $j \geq i$，若 $\pi[i, j] \in \mathbf{L}(\mathcal{A})$ 则必然存在 $\pi[i, j]$ 在 $\mathcal{A}$ 上的运行 $q_0, \ldots, q_{j-i+1}$，其中 $q_0 = q$，$q_{j-i+1} \in F$。注意到 $q_{l+1} \in \delta(q_l, \pi(i + 1)) = \delta(q_l, \Gamma_{i+l} \cap AP)$，所以对每个 $0 \leq l \leq j - i + 1$ 有 $q_l \in Q_l$。由于 $q_{j-i+1} \in F$，所

以 $Q_{j-i+1}\cap F\neq\emptyset$。由前面所得的结论 $\Gamma_j\in\mathbf{Sat}_\varphi(\psi')$。由归纳假设有 $\pi,j\models\psi'$。于是，由定义有 $\pi,i\models\psi$。

- 若 $\psi=\mathcal{A}$ leads ψ'（其中 $\mathcal{A}=\langle 2^{AP},Q,\delta,q,F\rangle$），此时，如果存在 Q 中的状态序列 $q_0,\ldots,q_{k-1},q_k$ 满足
  - $q_0=q$，$q_k\in F$，并且对每个 $0\leq l<k$ 有 $q_{l+1}\in\delta(q_l,\pi(i+l))$；
  - $\Gamma_{i+k-1}\in\mathbf{Sat}_\varphi(\psi')$

  那么必有 $\pi,i\models\psi$ 成立。这是因为，$\pi[i,i+k-1]\in\mathbf{L}(\mathcal{A})$，且由归纳假设有 $\pi,i+k-1\models\psi'$ 成立。

  反设这样的序列不存在，则构造如下的序列 $Q_0,Q_1,\ldots$，其中：$Q_0=\{q\}$，$Q_{l+1}=\bigcup_{q'\in Q_l}\delta(q',\pi(i+l))$。于是，在这种假设下，对任意的 $l\in\mathbb{N}$，若 $\bigcup_{q'\in Q_l}\delta(q',\pi(i+l))\cap F\neq\emptyset$ 则 $\Gamma_{i+l}\notin\mathbf{Sat}_\varphi(\psi')$。同时注意到 $\pi(i+l)=\Gamma_{i+l}\cap AP$，于是由 $\mathbf{Sat}_\varphi$ 的定义归纳可得：对每个 $l\geq 0$，必然存在某个 $q'\in Q_l$ 使得 $\Gamma_{i+l}\in\mathbf{Sat}_\varphi(\mathcal{A}^{q'}\text{ leads }\psi')$。

  不妨设 $\langle\mathcal{A},\psi'\rangle$ 是 φ 中的第 j 个牵引对（即：$\langle\mathcal{A},\psi'\rangle=\langle\mathcal{A}_j,\psi_j\rangle$），则由公平性约束 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle\mid P_j=\emptyset\}$ 知必然存在某个 $t>i$ 使得 $P_{j,t}=\emptyset$。于是，由 $\Delta_{\langle\mathcal{A}_j,\psi_j\rangle}$ 的定义知 $P_{j,t+1}=\{q'\in Q\mid\Gamma_{t+1}\in\mathbf{Sat}_\varphi(\mathcal{A}^{q'}\text{trigger }\psi')\}$。所以，$P_{j,t+1}\cap Q_{t-i+1}\neq\emptyset$，从而 $P_{j,t+1}\neq\emptyset$。此外，在这种假设下，对每个 $l\geq 1$ 有 $P_{j,t+l}\cap Q_{j,t-i+l}\neq\emptyset$ 蕴含 $P_{j,t+l+1}\cap Q_{j,t-i+l+1}\neq\emptyset$。因此，对任意 $l\geq 1$ 有 $P_{j,t-i+l}\neq\emptyset$。但是，这违反公平性约束 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle\mid P_j=\emptyset\}$，与 σ 是 $\mathcal{T}_\varphi$ 中的公平展开迹矛盾！因此这种假设不成立。于是，前面提到的序列 $q_0,\ldots,q_k$ 必然存在，从而 $\pi,i\models\psi$。

容易验证，上述归纳是完全的。由于 $s_0\in I_\varphi$，所以 $\Gamma_0\in\mathbf{Sat}_\varphi(\varphi)$。由前面所证结论，有 $\pi,0\models\varphi$ 成立。换言之，$\pi\models\varphi$。 □

定理 5.28 *对于任意 AFL 公式以及线性结构 π 有：若 $\pi\models\varphi$，则 $\pi\in\mathbf{L}(\mathcal{T}_\varphi)$。*

证明. 设 $\langle\mathcal{A}_1,\psi_1\rangle$，…，$\langle\mathcal{A}_m,\psi_m\rangle$ 是 φ 中所有牵引对，其中 $\mathcal{A}_i=\langle 2^{AP},Q_i,\delta_i,q_i,F_i\rangle$。假设 $\pi\models\varphi$，要证明 $\pi\in\mathbf{L}(\mathcal{T}_\varphi)$，只需根据 π 构造 $\mathcal{T}_\varphi$ 中的某条公平展开迹 $\sigma=s_0,s_1,\ldots$，且使得 σ 的派生线性结构为 π 即可。其中，$s_i=\langle\Gamma_i,(P_{1,i},\ldots,P_{m,i})\rangle$。

- σ 的构造过程如下。

  对每个 $i\in\mathbb{N}$，令 $\Gamma_i=\{\psi\in\mathbf{El}(\varphi)\mid\pi,i\models\psi\}$。

  对于每个 $1\leq j\leq m$ 以及每个 $i\in\mathbb{N}$，$P_{j,i}$ 的值按照如下方式确定。

  首先，令 $C_0^j=0$，并且 $P_{j,C_0^j}=\emptyset$。其次，对于每个 $k\in\mathbb{N}$，归纳假设 $P_{j,C_k^j}=\emptyset$，则使用下列步骤确定 C_{k+1}^j 的值，同时对每个 $C_k^j<l\leq C_{k+1}^j$ 构建 $P_{j,l}$：

- 令 $P_{j,C_k^j+1}=\{q\in Q_j \mid \pi, C_k^j+1 \models \mathcal{A}_j^q \text{ leads } \psi_j\}$。
- 于是，对每个 $q\in P_{j,C_k^j+1}$，存在某个 $D_{k,q}^j \geq C_k^j+1$，使得 $\pi[C_k^j+1, D_k^j]\in \mathbf{L}(\mathcal{A}_j^q)$，并且 $\pi, D_k^j \models \psi_j$。以下设 $\sigma_{k,q}^j \in (Q_j)^*$ 为 $\pi[C_k^j+1, D_k^j]$ 在 $\mathcal{A}_j^q$ 上的的某个可接收运行，则 $\sigma_{k,q}^j$ 中的最后一个状态必为 F_j 中的状态。于是，令 $C_{k+1}^j=\max\{D_{k,q}^j \mid q\in P_{j,C_k^j+1}\}+1$。特别的，如果 $P_{j,C_k^j+1}=\emptyset$，则有 $C_{k+1}^j=C_k^j+1$ 成立。
- 对于每个 $C_k^j < l \leq C_{k+1}^j$，令

$$P_{j,l}=\bigcup\nolimits_{q\in P_{j,C_k^j}}\{\sigma_{k,q}^j(l-C_k^j-1) \mid D_{k,q}^j \geq l\}。$$

  简而言之，若 $\sigma_{k,q}^j=q_0,q_1,\ldots,q_t$（其中 $q_0=q$，$q_t\in F_j$），则 q_0 至 q_{t-1} 依次会被加入至 P_{j,C_k^j+1} 至 P_{j,C_k^j+t}（但 q_t 不会因为 $\sigma_{k,q}^j$ 被添加至 P_{j,C_k^j+t+1}）。
- 容易验证 $P_{j,C_{k+1}^j}=\emptyset$。

- 现在证明 σ 是 $\mathcal{T}_\varphi$ 中的一条展开迹。

  首先证明：对于任意的 $\psi\in \mathbf{Sub}(\varphi)\cup\mathbf{El}(\varphi)$，$\Gamma_i\in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=true$ 或者 $\psi=false$ 时结论显然。
  - 若 $\psi=p$（resp. $\psi=\neg p$）且 $p\in AP$，则 $\Gamma\in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $p\in\Gamma_i$（resp. $p\notin\Gamma_i$）当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=\mathsf{X}\psi'$ 时，一定有 $\psi\in\mathbf{El}(\varphi)$，因此由 Γ_i 的构造知：$\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=\psi'\wedge\psi''$（resp. $\psi=\psi'\vee\psi''$）时，$\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i\in\mathbf{Sat}_\varphi(\psi')$ 且（resp. 或） $\Gamma_i\in\mathbf{Sat}_\varphi(\psi'')$，（由归纳假设）当且仅当 $\pi, i\models\psi'$ 且（resp. 或）$\pi, i\models\psi''$，当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=\psi'\mathsf{U}\psi''$ 时，$\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i\in\mathbf{Sat}_\varphi(\psi'')\cup(\mathbf{Sat}_\varphi(\psi')\cap\mathbf{Sat}_\varphi(\mathsf{X}(\psi'\mathsf{U}\psi'')))$，（由归纳假设）当且仅当 $\pi, i\models\psi''\vee(\psi'\wedge\mathsf{X}(\psi'\mathsf{U}\psi''))$，当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=\psi'\mathsf{R}\psi''$ 时，$\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i\in\mathbf{Sat}_\varphi(\psi'')\cap(\mathbf{Sat}_\varphi(\psi')\cup\mathbf{Sat}_\varphi(\mathsf{X}(\psi'\mathsf{R}\psi'')))$，（由归纳假设）当且仅当 $\pi, i\models\psi''\wedge(\psi'\vee\mathsf{X}(\psi'\mathsf{R}\psi''))$，当且仅当 $\pi, i\models\psi$。
  - 当 $\psi=\mathcal{A}\ \mathsf{abort!}\ b$ 时（其中 $\mathcal{A}=\langle 2^{AP},Q,\delta,q,F\rangle$），$\Gamma_i\in\mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i\in\mathbf{Sat}_\varphi(b)\cup\bigcup_{q'\in\delta(q,\Gamma_i\cap AP)}\mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'}\mathsf{abort!}\ b))$。注意到 $\Gamma_i\cap AP=\pi(i)$，由归纳假设，这当且仅当 $\pi, i\models b\vee\bigvee_{q'\in\delta(q,\pi(i))}\mathsf{X}(\mathcal{A}^{q'}\mathsf{abort!}\ b)$。由引理 5.26，这当且仅当 $\pi, i\models\psi$。

- 当 $\psi = \mathcal{A}$ monitor b 时（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$），$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \mathbf{Sat}_\varphi(b) \cap \bigcap_{q' \in \delta(q, \Gamma_i \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'}\text{monitor } b))$。由归纳假设，这当且仅当 $\pi, i \models \mathcal{A}$ monitor b，当且仅当 $\pi, i \models b \wedge \bigwedge_{q' \in \delta(q, \pi(i))} \mathsf{X}(\mathcal{A}^{q'}\text{monitor } b)$。由引理 5.26，这当且仅当 $\pi, i \models \psi$。
- 当 $\psi = \mathcal{A}$ trigger ψ' 时（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$），$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \text{若 } \delta(q, \Gamma \cap AP) \neq \emptyset, \text{则 } \Gamma \in \mathbf{Sat}_\varphi(\psi')\}$ 并且 $\Gamma_i \in \bigcap_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'}\text{trigger } \psi'))$。由归纳假设以及 $\pi(i) = \Gamma_i \cap AP$，这当且仅当 $\delta(q, \Gamma_i \cap AP) \neq \emptyset$ 蕴含 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi')$（由归纳假设即 $\pi, i \models \psi'$）以及 $\pi, i \models \bigwedge_{q' \in Q} \mathsf{X}(\mathcal{A}^{q'}\text{trigger } \psi')$。由引理 5.26，这当且仅当 $\pi, i \models \psi$。
- 当 $\psi = \mathcal{A}$ leads ψ' 时（其中 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F\rangle$），$\Gamma_i \in \mathbf{Sat}_\varphi(\psi)$ 当且仅当 $\Gamma_i \in \{\Gamma \subseteq \mathbf{El}(\varphi) \mid \delta(q, \Gamma \cap AP) \neq \emptyset, \text{且 } \Gamma \in \mathbf{Sat}_\varphi(\psi')\}$ 或者 $\Gamma_i \in \bigcup_{q' \in \delta(q, \Gamma \cap AP)} \mathbf{Sat}_\varphi(\mathsf{X}(\mathcal{A}^{q'}\text{leads } \psi'))$。由归纳假设以及 $\pi(i) = \Gamma_i \cap AP$，这当且仅当或者 $\delta(q, \Gamma_i \cap AP) \neq \emptyset$ 且 $\Gamma_i \in \mathbf{Sat}_\varphi(\psi')$（即：$\pi, i \models \psi'$）；或者 $\pi, i \models \bigvee_{q' \in Q} \mathsf{X}(\mathcal{A}^{q'}\text{leads } \psi')$。由引理 5.26，这当且仅当 $\pi, i \models \psi$。

于是，对于任意的 $\mathsf{X}\psi \in \mathbf{El}(\varphi)$ 以及任意的 $i \in \mathbb{N}$，$\Gamma_i \in \mathbf{Sat}_\varphi(\mathsf{X}\psi)$ 当且仅当 $\pi, i \models \mathsf{X}\psi$，当且仅当 $\pi, i+1 \models \psi$，当且仅当 $\Gamma_{i+1} \in \mathbf{Sat}_\varphi(\psi)$。

其次证明：对每个 $i \in \mathbb{N}$ 以及 $1 \leq j \leq m$，$((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta_{\langle \mathcal{A}_j, \psi_j\rangle}$。

- 当 $P_{j,i} \neq \emptyset$ 时，不妨设 $C_k^j > i > C_{k+1}^j$。于是，由 $P_{j,i}$ 的构造知：对每个 $q \in P_{j,i}$，存在某个 $q' \in Q_j$ 以及 $\pi[C_k^j{+}1, D_k^j]$ 上的可接收运行 $\sigma_{k,q'}^j$ 使得 $\sigma_{k,q'}^j(i - C_k^j - 1) = q$。如果 $i = D_k^j$，则 $\delta(q, \pi(i)) = \delta(q, \Gamma_i \cap AP) \neq \emptyset$，且 $\pi, i \models \psi_j$（即 $\Gamma_i \in \mathbf{Sat}_\varphi)(\psi_j)$）。若果 $i < D_k^j$，则存在 $q'' \in \delta(q, \pi(i)) = \delta(q, \Gamma_i \cap AP)$ 使得 $q'' \in P_{j,i+1}$。
- 当 $P_{j,i} = \emptyset$ 时，必然有 i 等于某个 C_k^j。于是 $P_{j,i+1} = P_{j,C_k^j+1} = \{q' \in Q_j \mid \pi, i+1 \models \mathcal{A}_j^{q'}\text{ leads } \psi_j\} = \{q' \in Q_j \mid \Gamma_i \in \mathbf{Sat}_\varphi(\mathcal{A}_j^{q'}\text{leads}\psi_j)\}$。

综上所述，对每个 $i \in \mathbb{N}$ 以及 $1 \leq j \leq m$ 都有 $((\Gamma_i, P_{j,i}), (\Gamma_{i+1}, P_{j,i+1})) \in \Delta_{\langle \mathcal{A}_j, \psi_j\rangle}$。再由前面证明的结论可知，对每个 $i \in \mathbb{N}$ 都有 $(s_i, s_{i+1}) \in \Delta_\varphi$。同时，由于 $\pi, 0 \models \varphi$，于是 $\Gamma_0 \in \mathbf{Sat}_\varphi(\varphi)$，进而有 $s_0 \in I_\varphi$。于是，σ 确为 $\mathcal{T}_\varphi$ 中的一条展开迹。

- 现在证明 σ 满足 $\mathcal{C}_\varphi$ 中的每条公平性约束。
  - 对 φ 中的每个子公式 $\phi_1 \mathsf{U} \phi_2$：若只有有穷多个 i 使得 $\pi, i \models \phi_1 \mathsf{U} \phi_2$，由前面所证结论，只有有穷多个 i 使得 $\Gamma_i \in \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)$。在这种情况下，$\sigma$ 中有无穷多个状态在集合 $\{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \notin \mathbf{Sat}_\varphi(\phi_1 \mathsf{U} \phi_2)\}$ 中。若有无穷多个 i 使得 $\pi, i \models \phi_1 \mathsf{U} \phi_2$，则（由 U 语义）有无穷多个 i' 使得 $\pi, i' \models \phi_2$。由前面所证结论，在这种情况下，σ 中有无穷多个状态在集合 $\{\langle \Gamma, (P_1, \ldots, P_m)\rangle \mid \Gamma \in$

$\mathbf{Sat}_\varphi(\phi_2)\}$ 中。因而，无论哪种情况， σ 均满足由 $\phi_1\mathsf{U}\phi_2$ 对应的公平性约束 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \notin \mathbf{Sat}_\varphi(\phi_1\mathsf{U}\phi_2)\} \cup \{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(\phi_2)\}$。

- 对 φ 中的每个子公式 $\mathcal{A}$ abort! b （不妨设 $\mathcal{A}$ 的状态集为 Q）：若只有有穷多个 i 使得 $\pi,i \models \bigvee_{q'\in Q}\mathcal{A}^{q'}$abort! b，由前面所证结论，只有有穷多个 i 使得 $\Gamma_i \in \mathbf{Sat}_\varphi(\bigvee_{q'\in Q}\mathcal{A}^{q'}$abort! $b)$。在这种情况下，σ 中有无穷多个状态在集合 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \notin \mathbf{Sat}_\varphi(\bigvee_{q'\in Q}\mathcal{A}^{q'}$abort! $b\}$ 中。若有无穷多个 i 使得 $\pi,i \models \bigvee_{q'\in Q}\mathcal{A}^{q'}$abort! b，则（由 abort 语义）有无穷多个 i' 使得 $\pi,i' \models b$。由前面所证结论，在这种情况下，σ 中有无穷多个状态在集合 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(b)\}$ 中。因而，无论哪种情况，σ 均满足由 $\mathcal{A}$ abort! b 对应的公平性约束 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \notin \mathbf{Sat}_\varphi(\bigvee_{q'\in Q}\mathcal{A}^{q'}$abort! $b)\} \cup \{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid \Gamma \in \mathbf{Sat}_\varphi(b)\}$。
- 对 φ 中的每个牵引对 $\langle\mathcal{A}_j,\psi_j\rangle$，由 σ 的构造知必然存在无穷多个 $i \in \mathbb{N}$ 使得$P_{j,i} = \emptyset$。因此，σ 必然满足 $\langle\mathcal{A}_j,\psi_j\rangle$ 对应的公平性约束 $\{\langle\Gamma,(P_1,\ldots,P_m)\rangle \mid P_j = \emptyset\}$。

所以，σ 满足 $\mathcal{C}_\varphi$ 中的每个公平性约束，因而 σ 是 $\mathcal{T}_\varphi$ 中的一条公平展开迹。

- 最后说明 π 是 σ 对应的派生线性结构，从而推出 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。

由于 $\mathcal{T}_\varphi$ 所关心的原子命题集合为 AP，而对于每个 $s = \langle\Gamma,(P_1,\ldots,P_m)\rangle$ 有 $\lambda_\varphi(s) = \Gamma \cap AP$。因此，对每个 $i \in \mathbb{N}$，要证明 $\pi(i) = \lambda_\varphi(s_i)$ 成立，只需对每个 $p \in AP$ 有："$p \in \Gamma_i$ 当且仅当 $p \in \pi(i)$" 即可。由 σ 的构造，对每个 $p \in AP$，由于 $p \in \mathbf{El}(\varphi)$，所以 $p \in \Gamma_i$ 当且仅当 $\pi,i \models p$ 当且仅当 $p \in \pi(i)$。于是，由定义有 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。 □

推论 5.29 (AFL 公式 tableau 的语言性质) *对于任意 AFL 公式以及线性结构 π 有: $\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{T}_\varphi)$。*

因此，AFL 的模型检验问题也可以转化为 CTL 的模型检验问题。

定理 5.30 *对于任意的公平迁移系统 $\mathcal{M}$ 以及 AFL 公式 φ 有: $\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M}\|\mathcal{T}_{\neg\varphi} \not\models \mathsf{EG}\ true$。*

5.6.3 基于 BDD 的 AFL tableau 编码

现在，介绍如何以 BDD 的形式获得 AFL 公式 tableau 的编码。任给 AFL 公式 φ（假设 φ 已经写为否定范式），设 $\langle\mathcal{A}_1,\psi_1\rangle$，…，$\langle\mathcal{A}_m,\psi_m\rangle$ 是 φ 中所有牵引对，其中 $\mathcal{A}_i = \langle 2^{AP},Q_i,\delta_i,q_i,F_i\rangle$。则对 $\mathcal{T}_\varphi$ 编码的各要素描述如下。

位变元集合：对每个 $\psi \in \mathbf{El}(\varphi)$，引入一个位变元 z_ψ；此外对每个 $1 \leq j \leq m$ 以及 $q \in Q_j$，引入一个位变元 $u_{j,q}$。

状态约束：状态约束 $\Phi_{S_\varphi} = true$。

迁移关系：迁移关系 Φ_{Δ_φ} 按照如下方式获得。

首先，对于每个 $\psi \in \mathbf{Sub}(\varphi) \cup \mathbf{El}(\varphi)$，归纳构建 $\mathbf{Sat}_\varphi(\psi)$ 之布尔表示 ϑ_ψ 如下：

- $\vartheta_{true} = true$，$\vartheta_{false} = false$。
- 对于每个 $p \in AP$，若 $\psi = p$，则 $\vartheta_\psi = z_p$；若 $\psi = \neg p$，则 $\vartheta_\psi = \neg z_p$。
- 若 $\psi = \mathsf{X}\psi'$，则令 $\vartheta_\psi = z_\psi$。（因为 $\psi \in \mathbf{El}(\varphi)$，所以位变元 z_ψ 在编码时被引入。）
- 若 $\psi = \psi_1 \wedge \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \wedge \vartheta_{\psi_2}$；若 $\psi = \psi_1 \vee \psi_2$，则 $\vartheta_\psi = \vartheta_{\psi_1} \vee \vartheta_{\psi_2}$。
- 若 $\psi = \phi_1 \mathsf{U} \phi_2$，则 $\vartheta_\psi = \vartheta_{\phi_2} \vee (\vartheta_{\phi_1} \wedge \vartheta_{\mathsf{X}(\phi_1 \mathsf{U} \phi_2)})$；若 $\psi = \phi_1 \mathsf{R} \phi_2$，则 $\vartheta_\psi = \vartheta_{\phi_2} \wedge (\vartheta_{\phi_1} \vee \vartheta_{\mathsf{X}(\phi_1 \mathsf{R} \phi_2)})$。
- 设 $\mathcal{A}$ 的状态集合为 Q。若 $\psi = \mathcal{A}$ abort! b，则 $\vartheta_\psi = \vartheta_b \vee \bigvee_{q' \in Q} \vartheta_{\mathsf{X}(\mathcal{A}^{q'} \text{abort! } b)}$；若 $\psi = \mathcal{A}$ monitor b，则 $\vartheta_\psi = \vartheta_b \wedge \bigwedge_{q' \in Q} \vartheta_{\mathsf{X}(\mathcal{A}^{q'} \text{monitor } b)}$。
- 设 $\mathcal{A} = \langle 2^{AP}, Q, \delta, q, F \rangle$。若 $\psi = \mathcal{A}$ trigger ψ'，则 $\vartheta_\psi = \bigwedge_{\substack{\Gamma \subseteq AP \\ \delta(q,\Gamma) \cap F \neq \emptyset}} (\vartheta_\Gamma \rightarrow \vartheta_{\psi'}) \wedge \bigwedge_{q' \in Q} \vartheta_{\mathsf{X}(\mathcal{A}^{q'} \text{trigger } \psi')}$；若 $\psi = \mathcal{A}$ leads ψ'，则 $\vartheta_\psi = \bigvee_{\substack{\Gamma \subseteq AP \\ \delta(q,\Gamma) \cap F \neq \emptyset}} (\vartheta_\Gamma \wedge \vartheta_{\psi'}) \vee \bigvee_{q' \in Q} \vartheta_{\mathsf{X}(\mathcal{A}^{q'} \text{leads } \psi')}$。其中，对任意的 $\Gamma \subseteq AP$，ϑ_Γ 是 $\bigwedge_{p \in \Gamma} z_p \wedge \bigwedge_{p \notin \Gamma} \neg z_p$ 的简写。当然，遍历 AP 的所有子集是困难的。因此，在第 6 章的工具实现中，将给出一种变通的实现方式— 用布尔表达式代替原子命题集合的子集。

接下来，令 ϑ'_ψ 为将 ϑ_ψ 中的每个位变元替换为其次态版本所得的公式。则 Φ_{Δ_φ} 就是以下布尔公式的合取。

$$\bigwedge_{\mathsf{X}\psi \in \mathbf{El}(\varphi)} z_{\mathsf{X}\psi} \leftrightarrow \vartheta'_\psi \tag{5.29}$$

$$\bigwedge_{1 \leq j \leq m} ((\bigvee_{q \in Q_j} u_{j,q}) \rightarrow \bigwedge_{q \in Q_j} (u_{j,q} \rightarrow (\vartheta_{\psi_j} \wedge \bigvee_{\substack{\Gamma \subseteq AP \\ \delta(q,\Gamma) \cap F_j \neq \emptyset}} \vartheta_\Gamma) \vee \bigvee_{\substack{\Gamma \subseteq AP \\ q' \in \delta(q,\Gamma)}} u'_{j,q'}) \tag{5.30}$$

$$(\bigwedge_{q \in Q_j} \neg u_{j,q}) \rightarrow \bigwedge_{q \in Q_j} (u'_{j,q} \leftrightarrow \vartheta'_{\mathcal{A}_j^{q'} \text{leads } \psi_j}) \tag{5.31}$$

容易验证，(5.29) 至 (5.31) 的合取确为 Φ_{Δ_φ} 的布尔编码。

初始状态集：初始状态集的编码 $\Phi_{I_\varphi} = \vartheta_\varphi$。

标记函数：对每个 $p \in AP$，$\Phi^p_{\lambda_\varphi} = z_p$。

公平性约束：根据 $\mathcal{C}_\varphi$ 的定义，$\Phi_{\mathcal{C}_\varphi}$ 的编码包含如下各项。

- 对 φ 中的每个子公式 $\phi_1 \mathsf{U} \phi_2$，其对应的公平性约束的布尔编码为 $\vartheta_{\phi_2} \vee \neg\vartheta_{\phi_1 \mathsf{U} \phi_2}$。
- 对 φ 中的每个子公式 $\mathcal{A}$ abort! b（其中 $\mathcal{A}$ 的状态集为 Q），其对应的公平性约束的布尔编码为 $\vartheta_b \vee \bigwedge_{q' \in Q} \neg\vartheta_{\mathcal{A}\text{abort! } b}$。
- 对 φ 中的每个牵引对 $\langle \mathcal{A}_j, \psi_j \rangle$，其对应的公平性约束的布尔编码为 $\bigwedge_{q \in Q_j} \neg u_{j,q}$。

容易看出：对于上述的 AFL 公式 φ，对其进行符号化编码所需的位变元数目为

$$\#\mathbf{El}(\varphi) + \sum_{1 \le j \le m} \#Q_j \tag{5.32}$$

而显然这个数目与公式的长度成线性关系。

5.7 本章小结

在本章，给出了四种时序逻辑— ATL_f、ATL_l、ATL_r 以及 AFL 的符号化模型检验算法。这四种时序逻辑均使用显式的时序算子，且其表达能力均等价于 ω-正规语言。这些逻辑的模型检验算法均是基于 tableau 方法进行— 通过构建（取非后）待验证规约的 tableau，可以将线性框架下时序逻辑的模型检验问题转化为 CTL 的符号化模型检验问题，从而可以采用基于 BDD 的符号化方法实现。

上述方法，是在 LTL 的符号化模型检验算法的基础上扩展而来的。每种扩展时序逻辑的 tableau 中都有一个额外的部件体现公式的“公平性约束”。在 ATL_f 和 ATL_l 的 tableau 中，附加部件分别体现正/负自动机公式的公平性约束；在 ATL_r 的 tableau 中同时存在这两种约束，而对于 AFL 公式，该种约束体现在公式中的牵引对中。

该种扩展，给出了此类扩展时序逻辑符号化模型检验的一个统一的框架。这些扩展时序逻辑的模型检验问题，都是 PSPACE-complete 的 [92]。实现这些算法所需的位变元数目，除 ATL_r 之外，都与规约性质成线性关系。因此，上述算法可以在 NuSMV 的基础上通过简单扩展得到高效实现（见第 7 章）。

ATL_r 公式符号化模型检验复杂度的主要来源在于为满足其负自动机子公式的公平性约束时所引入附加结构（Υ）。进一步讲，其中的主要瓶颈在于 NBW 的求补所带来的复杂度。如 5.5.2 节中所讨论的那样，受限于该问题复杂度的下界[112, 113] $\mathbf{Tig}(n-1)$（约为 $(0.76n)^n$），不可能存在线性代价的 ATL_r 公式的 tableau 编码方案。最近，Schewe[119] 将 Büchi 自动机求补问题的上界推进至 $\mathbf{Tig}(n+1)$，而

该值与已知下界仅相差一个 $\mathcal{O}(n^2)$ 的系数。因此，有可能存在更加高效的 ATL_r 公式 tableau 的构建过程，使其具有更加高效的编码方案（比如：位变元数目为 $\mathcal{O}(n \times \log(n))$ 的编码，其中 $n = |\varphi|$）。这是一个将来要研究的问题。

第六章　线性 μ-演算的符号化模型检验

6.1　引言

在第 5 章中，给出了三类扩展时序逻辑（ATL_f、ATL_l、ATL_r）以及 PSL 的变种（APSL）的线性部分（AFL）的符号化模型检验算法。在本章，将给出另外一类等价于 ω-正规语言的时序逻辑— 线性 μ-演算的符号化模型检验算法。

在针对 ATL_f、ATL_l、ATL_r 以及 AFL 的符号化模型检验算法中，都是基于 tableau 方法进行的。对采用显式时序连接子的时序逻辑而言，公式的 tableau 可以看作是在公式迁移图（见第 3 章）中截取出模态节点后附加上公平性约束而得到的迁移系统。在使用显式显式时序时序逻辑的公式迁移图中，任何一条踪迹中公式的时序连接子的嵌套深度（不包括 $\bigcirc$ 算子）是单调不增的。以 LTL 为例，对于公式 $\varphi_1\mathsf{U}\varphi_2$ 而言，将其展开成 $\varphi_2 \vee (\varphi_1 \wedge \mathsf{X}(\varphi_1\mathsf{U}\varphi_2))$ 后，踪迹中出现 φ_1、φ_2 或者 $\mathsf{X}(\varphi_1\mathsf{U}\varphi_2)$ 三者之一。其中，$\mathsf{X}(\varphi_1\mathsf{U}\varphi_2)$ 会被保持至下一个模态节点，而后重新生成 $\varphi_1\mathsf{U}\varphi_2$。但是，踪迹中一旦生成其子公式 φ_1 或 φ_2 后，$\varphi_1\mathsf{U}\varphi_2$ 便再不会出现在踪迹中。这样，对每个踪迹添加特定的公平性约束（比如 $\varphi_2 \vee \neg(\varphi_1\mathsf{U}\varphi_2)$）后，便可得到满足特定语言性质的公式 tableau。

然而，上述性质对于使用二阶量词或者不动点算子的时序逻辑公式却不成立。考虑线性 μ-演算公式

$$\varphi = \mu X.(p_1 \vee \nu Y.(\bigcirc X \vee (p_2 \wedge \bigcirc Y)))$$

在其重写序列中（可以认为是 φ 的博弈中的一个对决），公式 $\mathbf{D}_\varphi(X)$ 和 $\mathbf{D}_\varphi(Y)$ 有可能在其某条踪迹中无穷多次交替出现。换言之，由于在线性 μ-演算的公式中存在着变元嵌套关系，所以在其公式迁移图的踪迹中，有可能存在子公式生成父公式的情况。但是，仅将模态节点（或者说博弈系统中的模态格局）保留时，公式之间的相互生成信息便被丢弃。因此，第 5 章中介绍的 tableau 方法在线性 μ-演算的符号化模型检验中难以奏效，所以必须寻求新的检验算法。

尽管线性 μ-演算公式的难理解性和其固有的复杂度使得该种时序逻辑在实际应用中相对较少（主要见 [29, 91, 53, 11] 等文献），但是研究该种时序逻辑的符号化模型检验算法仍然具有一定的理论意义。

1. 首先，ATL_f 公式和 ATL_l 公式可以以线性复杂度转化为等价的线性 μ-演算公

式。因此，从某种意义上讲，线性 μ-演算的符号化模型检验问题是一个更加具有一般性的问题。

2. 其次，研究线性 μ-演算的符号化模型检验问题，以促使人们寻求该种逻辑易于被 BDD 编码实现的语言模型。
3. 线性 μ-演算公式能够写为具有特定形式的范式。针对这些特定形式的公式的模型检验算法可能是高效的。
4. 线性 μ-演算的分支时间版本— 模态 μ-演算的模型检验算法，得到了较为广泛的研究（见在本文的 2.3.3.2节）；但是线性版本的 μ-演算的模型检验算法被研究的相对较少。因此，研究线性 μ-演算的符号化模型检验算法是对前面算法的补充。

关于线性 μ-演算的符号化模型检验算法，本章主要研究如下两方面内容：

- 研究一般形式的线性 μ-演算公式的符号化模型检验算法。该算法主要基于自动机转化的符号化表示实现。
- 研究一类具备特定形式的线性 μ-演算的符号化模型检验算法。该种形式的线性 μ-演算公式是所有 ω-正规性质的一种范式。该算法由第 4 章中的线性 μ-演算公式的博弈系统的性质得到。

本章组织如下：

1. 6.2 节研究一般形式的线性 μ-演算公式的符号化模型检验算法。该节介绍如何将一个具有博弈范式的线性 μ-演算公式转化为若干个可以采用 BDD 表示的自动机的乘积，从而得到其符号化模型检验算法。
2. 6.3 节首先证明任何一个 ω-正规性质可以转化成为 ν-范式的线性 μ-演算公式。这样的公式可以非常容易的转化为一个迟滞迁移模型。通过建立模型间的迟滞合成，将该类线性 μ-演算的模型检验问题转化为 CTL 的模型检验问题。

6.2 一般形式的线性 μ-演算的符号化模型检验

本节给出一般形式的线性 μ-演算公式的符号化模型检验问题。由于含有自由变元的公式的语义依赖于变元指派，因此以线性 μ-公式作为规约时，只使用句子。此外，由于任何线性 μ-演算公式都能以多项式代价写为博弈范式，（见定义 4.2.7）因此本节假设所有的线性 μ-演算公式都已写为博弈范式形式。

6.2.1 线性 μ-演算公式的自动机表示

在本节，将会给出线性 μ-演算的一种自动机表示，将会证明：任何一个线性 μ-演算公式 φ 都能构造一个 NLW 和若干个 NBW 的乘积。将这些自动机的乘积投影至字母表 2^{AP} 上之后能够恰好获得该自动机的语言模型。

在本文 4.3.1 节中介绍过线性 μ-演算公式的一种判定模型— **迟滞 parity（字）-自动机**（SAPW），并给出了从（良命名）线性 μ-演算句子到 SAPW 的如下构造过程，它可以看作是 Wilke 的模态 μ-演算自动机构造[65]的特例（这里，复述 134 页的部分内容）：

对每个良命名的句子 φ，可以构造一个 SAPW $\mathcal{A}_\varphi = \langle Q_\varphi, \delta_\varphi, q_\varphi, \Omega_\varphi \rangle$，其中：

- $Q_\varphi = \{q_\psi \mid \psi \in \mathbf{Sub}(\varphi)\}$。
- δ_φ 定义如下：
  - $\delta_\varphi(q_{true}) = true$；$\delta_\varphi(q_{false}) = false$。
  - 对 φ 中的原子命题 p 而言，$\delta_\varphi(q_p) = p$；$\delta_\varphi(q_{\neg p}) = \neg p$。
  - 对 φ 中的（约束）变元 X 而言，$\delta_\varphi(q_X) = q_{\mathbf{D}_\varphi(X)}$。
  - $\delta_\varphi(q_{\varphi_1 \vee \varphi_2}) = q_{\varphi_1} \vee q_{\varphi_2}$；$\delta_\varphi(q_{\varphi_1 \wedge \varphi_2}) = q_{\varphi_1} \wedge q_{\varphi_2}$。
  - $\delta_\varphi(q_{\bigcirc\psi}) = \bigcirc q_\psi$。
  - $\delta_\varphi(q_{\mu X.\psi}) = q_\psi$；$\delta_\varphi(q_{\nu X.\psi}) = q_\psi$。
- 接收条件 Ω_φ 是任意一个满足如下约束的部分函数：
  - Ω_φ 在状态 q_ψ 处有定义当且仅当 ψ 是 φ 中的（约束）变元；
  - 若 X 是 φ 中的 μ-型约束变元，则 $\Omega_\varphi(q_X)$ 为奇数；若 X 是 φ 中的 ν-型约束变元，则 $\Omega_\varphi(q_X)$ 为偶数。
  - 若 $Y \lhd_\varphi X$，则 $\Omega_\varphi(q_Y) < \Omega_\varphi(q_X)$。

同时，在定理 4.27 中证明了：“对任意的线性结构 π 而言，$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$”，从而 $\mathcal{A}_\varphi$ 是 φ 的语言模型。当时，使用迟滞自动机的目的在于构造线性 μ-演算公式的博弈迁移规则。现在要给出从线性 μ-演算句子到标准 APW 的转化，而后证明由同一公式所得的 APW 和 SAPW 的等价性。

对每个（良命名的）句子 φ，可以构造一个 APW $\mathcal{A}'_\varphi = \langle 2^{AP}, Q'_\varphi, \delta', q'_{(\varphi,\perp)}, \Omega'_\varphi \rangle$，其中：

- $Q' = \{q'_{(\psi,\perp)} \mid \psi \in \mathbf{Sub}(\varphi)\} \cup \{q'_{(\psi,X)} \mid \psi \in \mathbf{Sub}(\varphi), X \in \mathbf{Sub}(\varphi) \cap VAR\}$。这样，初始状态 $q'_{(\varphi,\perp)} \in Q'$。
- 迁移函数 δ' 的计算过程需要用到一个辅助函数 θ。对每个 $a \subseteq AP$ 以及任意的

$Y \in \mathbf{Sub}(\varphi) \cap VAR$, $\delta'(q'_{(\psi,\perp)}, a) = \delta'(q'_{(\psi,Y)}, a) = \theta(q'_{(\psi,\perp)}, a)$。其中 θ 函数的计算过程递归定义如下：

- $\theta(q'_{(true,\perp)}, a) = true$; $\theta(q'_{(false,\perp)}, a) = false$。
- 对每个 $p \in AP$：若 $p \in a$ 则 $\theta(q'_{(p,\perp)}, a) = \theta(q'_{(p,Y)}, a) = true$; $\theta(q'_{(\neg p,\perp)}, a) = \theta(q'_{(\neg p,Y)}, a) = false$。若 $p \notin a$，则 $\theta(q'_{(p,\perp)}, a) = \theta(q'_{(p,Y)}, a) = false$; $\theta(q'_{(\neg p,\perp)}, a) = \theta(q'_{(\neg p,Y)}, a) = true$。
- 对每个 $X \in VAR$, $\theta(q'_{(X,\perp)}, a) = \theta(q'_{(\mathbf{D}_\varphi(X),X)}, a)$;
  若 $Y \lhd_\varphi X$，则 $\theta(q'_{(X,Y)}, a) = \theta(q'_{(\mathbf{D}_\varphi(X),X)}, a)$;
  若 $X \lhd_\varphi Y$ 或 X 与 Y 的依赖关系不可比较，则 $\theta(q'_{(X,Y)}, a) = \theta(q'_{(\mathbf{D}_\varphi(X),Y)}, a)$。
- $\theta(q'_{(\bigcirc\psi,\perp)}, a) = q'_{(\psi,\perp)}$; $\theta(q'_{(\bigcirc\psi,Y)}, a) = q'_{(\psi,Y)}$。
- $\theta(q'_{(\psi_1\vee\psi_2,\perp)}, a) = \theta(q'_{(\psi_1,\perp)}, a) \vee \theta(q'_{(\psi_2,\perp)}, a)$; $\theta(q'_{(\psi_1\vee\psi_2,Y)}, a) = \theta(q'_{(\psi_1,Y)}, a) \vee \theta(q'_{(\psi_2,Y)}, a)$。$\theta(q'_{(\psi_1\wedge\psi_2,\perp)}, a) = \theta(q'_{(\psi_1,\perp)}, a) \wedge \theta(q'_{(\psi_2,\perp)}, a)$; $\theta(q'_{(\psi_1\wedge\psi_2,Y)}, a) = \theta(q'_{(\psi_1,Y)}, a) \wedge \theta(q'_{(\psi_2,Y)}, a)$。
- $\theta(q'_{(\mu X.\psi,\perp)}, a) = \theta(q'_{(\psi,\perp)}, a)$; $\theta(q'_{(\mu X.\psi,Y)}, a) = \theta(q'_{(\psi,Y)}, a)$。$\theta(q'_{(\nu X.\psi,\perp)}, a) = \theta(q'_{(\psi,\perp)}, a)$; $\theta(q'_{(\nu X.\psi,Y)}, a) = \theta(q'_{(\psi,Y)}, a)$。

- 接收条件 Ω_φ 是任意一个满足如下约束的部分函数：
  - Ω'_φ 在每个 $q'_{(\psi,Y)}$ 处有定义；在每个 $q'_{(\psi,\perp)}$ 处无定义。
  - 若 Y 是 μ-型（resp. ν-型）约束变元，则 $\Omega'_\varphi(q'_{(\psi,Y)})$ 是奇数（resp. 偶数）。
  - 若 $Y \lhd_\varphi X$，则 $\Omega'_\varphi(q'_{(\psi,Y)}) \leq \Omega'_\varphi(q'_{(\psi',X)})$。这里，$\psi$ 和 ψ' 是 φ 的任意两个子公式。

例 6.2.1 设 $\varphi = \mu X.(p_1 \vee \nu Y.(\bigcirc X \vee (p_2 \wedge \bigcirc Y)))$, $AP = \{p_1, p_2\}$。令 $\psi_1 = \nu Y.(\bigcirc X \vee (p_2 \wedge \bigcirc Y))$, $\psi_2 = \bigcirc X \vee (p_2 \wedge \bigcirc Y)$, $\psi_3 = p_2 \wedge \bigcirc Y$, 再令 $a_0 = \emptyset$, $a_1 = \{p_1\}$, $a_2 = \{p_2\}$ 以及 $a_3 = \{p_1, p_2\}$, 现在计算 $\delta'(q'_{(\psi_0,\perp)}, a_2)$ 即 $\theta(q'_{(\psi_0,\perp)}, a_2)$ 的值。按照定义:

$$\begin{aligned}
\theta(q'_{(\psi_0,\perp)}, a_2) &= \theta(q'_{(p_1,\perp)}, a_2) \vee \theta(q'_{(\psi_1,\perp)}, a_2) \\
&= \theta(q'_{(\psi_2,\perp)}, a_2) \\
&= \theta(q'_{(\bigcirc X,\perp)}, a_2) \vee \theta(q'_{(\psi_2,\perp)}, a_2) \\
&= q'_{(X,\perp)} \vee \theta(q'_{(\psi_3,\perp)}, a_2) \\
&= q'_{(X,\perp)} \vee \theta(q'_{(p_2,\perp)}, a_2) \wedge \theta(q'_{(\bigcirc Y,\perp)}, a_2) \\
&= q'_{(X,\perp)} \vee q'_{(Y,\perp)}
\end{aligned}$$

注意，上式中 $\theta(q'_{(p_1,\perp)}, a_2) = false$, $\theta(q'_{(p_2,\perp)}, a_2) = true$。类似的，可以得到: $\delta'(q'_{(\psi_0,\perp)}, a_0) = q'_{(X,\perp)}$; $\delta'(q'_{(\psi_0,\perp)}, a_1) = \delta'(q'_{(\psi_0,\perp)}, a_3) = true$; $\delta'(q'_{(X,\perp)}, a_0) = q'_{(X,X)}$;

$\delta'(q'_{(X,\perp)},a_1)=\delta'(q'_{(X,\perp)},a_3)=true$; $\delta'(q'_{(X,\perp)},a_2)=q'_{(X,X)}\vee q'_{(Y,X)}$; $\delta'(q'_{(Y,\perp)},a_0)=\theta'(q'_{(Y,\perp)},a_1)=false$; $\delta'(q'_{(Y,\perp)},a_2)=\delta'(q'_{(Y,\perp)},a_3)=q'_{(X,\perp)}\vee q'_{(Y,\perp)}$。此外，对于每个 $i\in\{0,1,2,3\}$，$\delta'(q'_{(X,X)},a_i)=\delta'(q'_{(X,\perp)},a_i)$; $\delta'(q'_{(Y,X)},a_i)=\delta'(q'_{(Y,\perp)},a_i)$。 □

计算 θ 的递归过程有两个出口。即：在计算 $\theta(q'_{(\psi,\perp)},a)$ 或 $\theta(q'_{(\psi,Y)},a)$ 时，当 $\psi\in AP\cup\overline{AP}\cup\{true,false\}$ 或者 $\psi=\bigcirc\psi'$ 时计算过程终止。因此，若 φ 是受卫公式，上述计算过程总会终止。事实上，由于 $\lhd_\varphi$ 是向上可比较的，因此从初始状态开始递归计算 θ 时，并不会遇到“X 与 Y 的依赖关系不可比较” 的情况。事实上，Q'_φ 中的许多状态并不是“初始可达”的。它们仅仅是为了计算 θ 方便而引入的中间状态。比如在例 6.2.1 中，实际的初始可达状态只有 $q'_{(\varphi,\perp)}$、$q'_{(X,\perp)}$，$q'_{(Y,\perp)}$，$q'_{(X,X)}$ 以及 $q'_{(Y,X)}$ 这五个。关于 $\mathcal{A}'_\varphi$ 中的初始可达状态数目，容易证明如下引理。

引理 6.1 设 K 和 N 分别是 φ 中形如 $\bigcirc\psi$ 的公式数目和约束变元的数目，则 $\mathcal{A}'_\varphi$ 中初始可达的状态数目不超过 $(K+1)\times(N+1)$。

现在证明 $\mathcal{A}'_\varphi$ 也是 φ 的语言模型。为此，只需证明 $\mathbf{L}(\mathcal{A}_\varphi)=\mathbf{L}(\mathcal{A}'_\varphi)$ 即可。

定理 6.2 设 $\mathcal{A}_\varphi$ 和 $\mathcal{A}'_\varphi$ 分别是由 φ 得到的 $SAPW$ 和 APW。对于任意的线性结构 π，若 $\pi\in\mathbf{L}(A)_\varphi$，则 $\pi\in\mathbf{L}(\mathcal{A}'_\varphi)$。

证明. 设 $\langle T,\rho\rangle$ 是 π 在 $\mathcal{A}_\varphi$ 上的可接收运行。由于 ρ 是从 T 到 $Q_\varphi\times\mathbb{N}$ 的函数，用 ρ_1 和 ρ_2 分别表示 ρ 的两个分量。即：若 $\rho(x)=(q_\psi,i)$，则 $\rho_1(x)=q_\psi$，$\rho_2(x)=i$。

称 $x\in T$ 是 T 中的**状态节点**，如果 $x=\epsilon$ 或者对 x 的父节点 x' 有 $\rho_2(x)=\rho_2(x')+1$ 成立。显然，若 x 是状态节点且 x' 是 x 的父节点，则 $\rho_1(x')$ 必然形如 $q_{\bigcirc\psi}$。对 T 中的任意节点 x 及其子孙节点 y，称 y 是 x 的**直接状态子孙节点**，如果 y 是 T 中的状态节点，并且 $\rho_2(y)=\rho_2(x)+1$。以下，用 $\mathbf{K}(x)$ 表示 x 在 T 中的直接状态子孙节点集合。于是，若 $y\in\mathbf{K}(x)$，则必然存在从 x 到 y 的有穷路径 $\sigma_{x,y}$。对每个 $y\in\mathbf{K}(x)$，定义 $V_{x,y}\in VAR\cup\{\perp\}$ 如下：

$$V_{x,y}=\begin{cases}X & \text{，存在 } \sigma_{x,y} \text{ 中的某节点 } x'(x'\neq y) \text{ 使得 } \rho_1(x')=q_X, \text{且对任意的}\\ & Y\in VAR, \text{若存在 } \sigma_{x,y} \text{ 中的节点 } x''(x''\neq y) \text{ 使得 } \rho_1(x'')=q_Y,\\ & \text{则有 } Y\lhd_\varphi X\\ \perp & \text{，对任意的 } Y\in VAR \text{ 以及 } \sigma_{x,y}\text{中每个不为 } y \text{ 的节点 } x', \text{有 } \rho_1(x')\neq q_Y\end{cases}。$$

注意 $V_{x,y}$ 是良定义的— 这是因为 $\lhd_\varphi$ 是向上可比较的，所以对于任意的 $X,Y\in VAR$，如果存在 $x',x''\in T$ 使得 $\rho_1(x')=q_X$，$\rho_1(x'')=q_Y$，且 x' 是 x'' 的祖先，那么必然有 $X\lhd_\varphi Y$ 或者 $Y\lhd_\varphi X$ 二者之一成立。

首先证明：对于任意的 $x\in T$，若 $\rho(x)=(q_\psi,i)$，则集合 $\mathbf{H}(x)=\{q'_{(\phi,Y)}\mid$

存在 $y \in \mathbf{K}(x)$ 使得 $\rho_1(y) = \phi, V_{x,y} = Y\}$ 满足 $\theta(q'_{(\psi,\perp)}, \pi(i))$。

采用引理 4.5 中的做法，对每个 $\psi \in \mathbf{Sub}(\varphi)$，赋于一个自然数 $\mathbf{Rk}(\psi)$（只是将基础项改为 "若 $\psi \in AP \cup \overline{AP} \cup \{true, false\}$ 或者 $\psi = \bigcirc\psi'$ 则 $\mathbf{Rk}(\psi') = 0$"）。以下，按照 $\mathbf{Rk}(\psi)$ 的值利用第二归纳法证明该结论。

- 当 $\psi \in AP$（resp. $\psi \in \overline{AP}$）时，在 $\mathcal{A}_\varphi$ 中 $\delta_\varphi(q_\rho(x)) = \rho(x)$。由于 $\rho(x) = (q_\psi, i)$，由定义 4.3.2，必然有 $\psi \in \pi(i)$（resp. $\psi \notin \pi(i)$）。而此时，在 $\mathcal{A}'_\varphi$ 中必然有 $\theta(q'_{(\psi,\perp)}, i) = true$。于是命题成立。
- 当 $\psi = \bigcirc\psi'$ 时，由定义 4.3.2，x 在 T 中必然存在子节点 y 使得 $\rho(y) = (q_{\psi'}, i+1)$，由定义有 $y \in \mathbf{K}(x)$。这时，路径 $\sigma_{x,y}$ 中不为 y 的节点只有 x，而 $\rho_1(x) = q_{\bigcirc\psi'}$，因此 $V_{x,y} = \perp$。由于 $\theta(q'_{(\psi,\perp)}, \pi(i)) = q'_{(\psi',\perp)}$，所以此时命题成立。
- 当 $\psi = X \in VAR$ 时，由定义 4.3.2，x 存在 T 中存在子节点 y，使得 $\rho(y) = (q_{\mathbf{D}_\varphi(X)}, i)$。由归纳假设，$\mathbf{H}(y)$ 满足 $\theta(q'_{(\mathbf{D}_\varphi(X),\perp)}, \pi(i))$。由 θ 的定义，容易证明：$\theta(q'_{(X,\perp)}, \pi(i))$ 是将 $\theta(q'_{(\mathbf{D}_\varphi(X),\perp)}, \pi(i))$ 中的每个 $q'_{(\phi,Y)}$ 及 $q'_{(\phi,\perp)}$ 替换为 $q'_{(\phi,X)}$ 所得的公式，其中 $Y \lhd_\varphi X$。与此同时，$\mathbf{H}(x)$ 是将 $\mathbf{H}(y)$ 中的每个 $q'_{(\phi,Y)}$ 及 $q'_{(\phi,\perp)}$ 替换为 $q'_{(\phi,X)}$ 所得的集合。因此，当 $\psi = X$ 时结论成立。
- 当 $\psi = \psi_1 \vee \psi_2$ 时，由定义 4.3.2 知 x 必然在 T 中存在子节点 y 使得 $\rho(y) = (q_{\psi_1}, i)$ 或者 $\rho(y) = (q_{\psi_2}, i)$。由归纳假设，$\mathbf{H}(y)$ 满足 $\theta(q'_{(\psi_1,\perp)}, \pi(i))$ 或者 $\theta(q'_{(\psi_2,\perp)}, \pi(i))$。容易证明：$\mathbf{H}(y) \subseteq \mathbf{H}(x)$，同时 $\theta(q'_{(\psi,\perp)}, \pi(i)) = \theta(q'_{(\psi_1,\perp)}, \pi(i)) \vee \theta(q'_{(\psi_2,\perp)}, \pi(i))$。因此 $\mathbf{H}(x)$ 满足 $\theta(q'_{(\psi,\perp)}, \pi(i))$。同样可以证明 $\psi = \psi_1 \wedge \psi_2$ 的情形。
- 当 $\psi = \mu X.\psi'$ 是，由定义 4.3.2，x 必然在 T 中存在子节点 y 使得 $\rho(y) = (q_{\psi'}, i)$。由归纳假设知 $\mathbf{H}(y)$ 满足 $\theta(q'_{(\psi',\perp)}, \pi(i))$。容易证明：$\mathbf{H}(y) \subseteq \mathbf{H}(x)$，同时 $\theta(q'_{(\psi,\perp)}, \pi(i)) = \theta(q'_{(\psi',\perp)}, \pi(i))$。因此 $\mathbf{H}(x)$ 满足 $\theta(q_{(\psi,\perp)}, \pi(i))$。同样可以证明 $\psi = \nu X.\psi$ 的情形。

综上所述，该结论对于 T 中的任意节点 x 成立。

现在构造 Q'_φ-标记树 $\langle T', \rho' \rangle$ 如下（该构造伴随建立一个从 T' 到 T 的映射 f）：

- 令 $\rho'(\epsilon) = q'_{(\varphi,\perp)}$，并令 $f(\epsilon) = \epsilon$。
- 对 T 中的每个节点 x，设 $f(x) = y$ 以及 $\mathbf{K}(y) = \{y_0, \ldots, y_t\}$，则对每个 $0 \le c \le t$ 而言，为 x 添加一个子节点 $x \cdot c$，并令 $\rho'(x \cdot c) = q'_{(\phi, V_{y,y_c})}$，其中 $\rho(y) = q_\phi$；再令 $f(x \cdot c) = y_c$。

  容易归纳证明：对每个 $x \in T'$，$\rho'(x) = q'_{(\psi,\star)}$ 当且仅当 $\rho_1(f(x)) = q_\psi$，其中

$\star \in VAR \cup \{\bot\}$；并且 $|x| = \rho = \rho_2(f(x))$。同时，集合 $\{\rho'(x \cdot c) \mid x \cdot c \in T'\}$ 恰好为 $\mathbf{H}(f(x))$。不妨设 $\rho'(x) = q'_{(\psi,\star)}$，$|x| = i$，注意到 $\delta(q'_{(\psi,\star)}, \pi(i)) = \theta(q'_{(\psi,\bot)}, \pi(i))$，于是由上述结论有 $\{\rho'(x \cdot c) \mid x \cdot c \in T'\}$ 满足 $\delta(q'_{(\psi,\star)}, \pi(i))$。因此，$\langle T', \rho' \rangle$ 是 $\mathcal{A}'_\varphi$ 在 π 上的一个运行。

任取 T' 中的一条无穷路径 $\sigma' = x_0, x_1, \ldots$，令 $\max(\mathbf{Inf}(\Omega'_\varphi(\sigma'))) = n$（这里 $\mathbf{Inf}(\Omega'_\varphi(\sigma)) = \{m \in \mathbb{N} \mid$ 有无穷多个 $i \in \mathbb{N}$ 使得 $\Omega'_\varphi(\rho(x_i)) = m\}$），则必然存在某个 $q'_{(\phi,X)} \in Q'_\varphi$，使得 $\Omega'_\varphi(q'_{(\phi,X)}) = n$ 并且有无穷多个 $i \in \mathbb{N}$ 使得 $\rho'(x_i) = q'_{(\phi,X)}$。由构造，在 T 中包含节点 $f(x_0), f(x_1), \ldots$ 的无穷路径（记为 σ）中必然有无穷多个节点 x 使得 $\rho(x) = q_X$。与此同时，对于任意的 $Y \in VAR$，若该无穷路径包含无穷多个节点 x' 使得 $\rho(x') = q_Y$。于是，由 θ 的定义有 $Y \lhd_\varphi X$。根据 Ω_φ 的定义以及 $\langle T, \rho \rangle$ 是 π 在 $\mathcal{A}_\varphi$ 上的可接收运行可知 X 必然是 φ 中的 ν-型约束变元，所以 n 是偶数。于是，$\langle T', \rho' \rangle$ 是 π 在 $\mathcal{A}'_\varphi$ 上的一个可接收运行，所以 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$。 □

在证明定理 6.2 的逆定理之前，首先引入以下概念。

对任意给定的标记树 $\langle \hat{T}, \hat{\rho} \rangle$ 以及 $x \in \hat{T}$，用 $\langle \hat{T}^x, \hat{\rho}^x \rangle$ 表示 $\langle \hat{T}, \hat{\rho} \rangle$ 中以 x 为根的子标记树。即：$\hat{T}^x = \{y \in \mathbb{N} \mid x \cdot y \in \hat{T}\}$，$\hat{\rho}^x(y) = \hat{\rho}(x \cdot y)$。

定义 6.2.1 (单步展开树) 给定 $q_\psi \in Q_\varphi$（这里，Q_φ 是 φ 所对应 APT 的状态集）以及 $a \subseteq AP$，则 q_ψ 关于 a 的一棵**单步展开树** 是一棵有穷 Q_φ-标记树 $\langle \hat{T}, \hat{\rho} \rangle$。其中，$\hat{\rho}(\epsilon) = q_\psi$，并且:

- 若 $\psi \in AP \cup \overline{AP} \cup \{true\}$，则 $\hat{T}$ 中仅有根节点 ϵ，并且 $a \models \psi$。
- 若 $\psi = X \in VAR$，则 $\hat{T}$ 的根节点仅有一个子节点 0，并且 $\langle \hat{T}^0, \hat{\rho}^0 \rangle$ 是 $q_{\mathbf{D}_\varphi(X)}$ 在 a 上的一个单步展开树。
- 若 $\psi = \bigcirc\psi'$，则 $\hat{T}$ 中仅有两个节点 ϵ 和 0，并且 $\hat{\rho}(0) = q_{\psi'}$。
- 若 $\psi = \psi_1 \wedge \psi_2$，则根节点 ϵ 有两个子节点 0 和 1，并且 $\langle \hat{T}^0, \hat{\rho}^0 \rangle$ 和 $\langle \hat{T}^1, \hat{\rho}^1 \rangle$ 中一个是 q_{ψ_1} 在 a 上的单步展开树，另一个是 q_{ψ_2} 在 a 上的单步展开树。
- 若 $\psi = \psi_1 \vee \psi_2$，则根节点 ϵ 仅有一个子节点 0，并且 $\langle \hat{T}^0, \hat{\rho}^0 \rangle$ 是 q_{ψ_1} 或者 q_{ψ_2} 在 a 上的单步展开树。
- 若 $\psi = \mu X.\psi'$ 或者 $\psi = \nu X.\psi'$，则根节点 ϵ 仅有一个子节点 0，并且 $\langle \hat{T}^0, \hat{\rho}^0 \rangle$ 是 $q_{\psi'}$ 在 a 上的单步展开树。 □

定理 6.3 设 $\mathcal{A}_\varphi$ 和 $\mathcal{A}'_\varphi$ 分别是由 φ 得到的 $SAPW$ 和 APW。对于任意的线性结构 π 而言，若 $\pi \in \mathbf{L}(\mathcal{A}'_\varphi)$，则 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$。

证明. 设 $\langle T', \rho' \rangle$ 是 $\mathcal{A}'_\varphi$ 在 π 上的一个可接收运行。要证明 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$，只需根据 $\langle T', \rho' \rangle$ 构造 $\mathcal{A}_\varphi$ 在 π 上的一个可接收运行 $\langle T, \rho \rangle$ 即可。该运行的构造需要借助 Q_φ

中状态的单步展开树。在此之前，需要证明单步展开树的如下性质。

设 $\psi \in \mathbf{Sub}(\varphi)$，$P' \subseteq Q'_\varphi$，$a \subseteq AP$。若 $P' \models \theta(q'_{(\psi,\perp)}, a)$，则必然存在 q_ψ 在 a 上的某个单步展开树 $\langle \hat{T}, \hat{\rho} \rangle$ 使得对 $\hat{T}$ 中的每个叶节点 x，都存在某个 $q'_{(\hat{\rho}(x),\star)} \in P'$ 并且：

$$\star = \begin{cases} X & ,\text{存在 } x \text{ 的某祖先节点 } y, \text{使得 } \hat{\rho}(y) = q_X, \text{并且对从 } \epsilon \text{ 到 } x \text{路径上} \\ & \text{任一非叶节点 } y' \text{ 以及任意的 } Y \in VAR, \text{若 } \hat{\rho}(y) = q_Y \text{ 则 } Y \lhd_\varphi X \\ \perp & ,\text{对于 } x \text{ 的任一祖先节点 } y \text{ 以及任意的 } Y \in VAR \text{ 有 } \hat{\rho}(y) \neq q_Y \end{cases} 。$$

以下，若 $\langle \hat{T}, \hat{\rho} \rangle$ 与 P' 之间满足上述关系，则记作 $P' \Vdash \langle \hat{T}, \hat{\rho} \rangle$。

仍然根据 $\mathbf{Rk}(\psi)$ 的值进行归纳。

- 若 $\psi \in AP \cup \overline{AP} \cup \{true, false\}$，则 P' 满足 $\theta(q'_{(\psi,\perp)}, a)$ 当且仅当 $a \models \psi$，这时结论成立。
- 若 $\psi = \bigcirc\psi'$，则 $\theta(q'_{(\psi,\perp)}, a) = q'_{(\psi'\perp)}$，所以此时必然有 $q'_{(\psi'\perp)} \in P'$。而在 $\langle \hat{T}, \hat{\rho} \rangle$ 中，有 $\hat{\rho}(0) = q'_{(\psi'\perp)}$，并且 0 是 $\hat{T}$ 中的叶节点，故而此时结论成立。
- 若 $\psi = X \in VAR$，则由 θ 的定义，必然存在 $P'' \subseteq Q'_\varphi$，使得 P'' 满足 $\theta(q'_{(\mathbf{D}_\varphi(X),\perp)}, a)$，并且$P''$ 是将 P' 中每个 $q'_{(\phi,X)}$ 替换为 $q'_{(\phi,\perp)}$ 或者替换为某些 $q'_{(\phi,Y)}$ 得到（这里 $Y \lhd_\varphi X$）的集合。由归纳假设，必然存在 $q'_{(\mathbf{D}_\varphi,\perp)}$ 在 a 上的某个单步展开树 $\langle \hat{T}', \hat{\rho}' \rangle$ 使得 $P'' \Vdash \langle \hat{T}', \hat{\rho}' \rangle$。令 $\langle \hat{T}, \hat{\rho} \rangle$ 是这样一棵标记树：
  - $\hat{T} = \{\epsilon\} \cup \{0 \cdot x \mid x \in \hat{T}'\}$；
  - $\hat{\rho}(\epsilon) = q_X$，$\hat{\rho}(0 \cdot x) = \hat{\rho}'(x)$。

  于是，$\langle \hat{T}, \hat{\rho} \rangle$ 是 q_ψ 在 a 上的一棵展开树。同时容易验证：对于 $\hat{T}$ 中的每个叶节点 $0 \cdot x$，由于存在 $q'_{(\hat{\rho}'(x),\star)} \in P''$ 满足结论中的条件。所以由 P' 和 P'' 的关系知：若 $\star = \perp$ 或 $\star = Y \lhd_\varphi X$，则 $q'_{(\hat{\rho}'(x),X)} \in P'$；否则 $q'_{(\hat{\rho}'(x),\star)} \in P'$。因此，$P' \Vdash \langle \hat{T}, \hat{\rho} \rangle$。
- 若 $\psi = \psi_1 \vee \psi_2$，则由于 $P'' \models \theta(q'_{(\psi,\perp)}, a)$，因此存在 $k \in \{1, 2\}$ 使得 $P'' \models \theta(q'_{(\psi_k,\perp)}, a)$。根据归纳假设，存在 ψ_k 在 a 上的一棵展开树 $\langle \hat{T}', \hat{\rho}' \rangle$，使得 $P' \Vdash \langle \hat{T}', \hat{\rho}' \rangle$。令 $\langle \hat{T}, \hat{\rho} \rangle$ 是这样一棵标记树：
  - $\hat{T} = \{\epsilon\} \cup \{0 \cdot x \mid x \in \hat{T}'\}$；
  - $\hat{\rho}(\epsilon) = q_X$，$\hat{\rho}(0 \cdot x) = \hat{\rho}'(x)$。

  于是，$\langle \hat{T}, \hat{\rho} \rangle$ 是 q_ψ 在 a 上的一棵展开树。同时，由定义容易验证 $P' \Vdash \langle \hat{T}, \hat{\rho} \rangle$。
- 若 $\psi = \psi_1 \wedge \psi_2$，则由于 $P'' \models \theta(q'_{(\psi,\perp)}, a)$，所以 $P'' \models \theta(q'_{(\psi_1,\perp)}, a)$，$P'' \models \theta(q'_{(\psi_2,\perp)}, a)$。由归纳假设，存在 q_{ψ_1} 和 q_{ψ_2} 在 a 上的展开树 $\langle \hat{T}', \hat{\rho}' \rangle$ 以及 $\langle \hat{T}'', \hat{\rho}'' \rangle$ 使得 $P' \Vdash \langle \hat{T}', \hat{\rho}' \rangle$ 以及 $P' \Vdash \langle \hat{T}'', \hat{\rho}'' \rangle$。令 $\langle \hat{T}, \hat{\rho} \rangle$ 是这样一棵标记树：

– $\hat{T}=\{\epsilon\}\cup\{0\cdot x\mid x\in\hat{T}'\}\cup\{1\cdot x\mid x\in\hat{T}''\}$；

– $\hat{\rho}(\epsilon)=q_X$，$\hat{\rho}(0\cdot x)=\hat{\rho}'(x)$，$\hat{\rho}(1\cdot x)=\hat{\rho}''(x)$。

于是，$\langle\hat{T},\hat{\rho}\rangle$ 是 q_ψ 在 a 上的一棵展开树。同时，由定义容易验证 $P'\Vdash\langle\hat{T},\hat{\rho}\rangle$。

- 若 $\psi=\mu X.\psi'$ 或者 $\psi=\nu X.\psi'$，则由于 $P''\models\theta(q'_{(\psi,\perp)},a)$，所以 $P''\models\theta(q'_{(\psi',\perp)},a)$。由归纳假设，存在 ψ' 在 a 上的一棵展开树 $\langle\hat{T}',\hat{\rho}'\rangle$，使得 $P'\Vdash\langle\hat{T}',\hat{\rho}'\rangle$。令 $\langle\hat{T},\hat{\rho}\rangle$ 是这样一棵标记树：

  – $\hat{T}=\{\epsilon\}\cup\{0\cdot x\mid x\in\hat{T}'\}$；

  – $\hat{\rho}(\epsilon)=q_X$，$\hat{\rho}(0\cdot x)=\hat{\rho}'(x)$。

  于是，$\langle\hat{T},\hat{\rho}\rangle$ 是 q_ψ 在 a 上的一棵展开树。同时，由定义容易验证 $P'\Vdash\langle\hat{T},\hat{\rho}\rangle$。

因此，上述结论成立。

现在给出 $\langle T,\rho\rangle$ 的构造过程，该过程伴随从 T 到 T' 的部分函数 g 的构造。

- 首先，构造根节点 ϵ，令 $\rho_1(\epsilon)=q_\varphi$，$\rho_2(\epsilon)=0$（这里，$\rho_1$ 和 ρ_2 分别是 ρ 的两个分量，即若 $\rho(x)=(q_\psi,i)$，则 $\rho_1(x)=q_\psi$，$\rho_2(x)=i$），再令 $g(\epsilon)=\epsilon$。于是，$g(\epsilon)$ 存在，并且 $\rho'(g(\epsilon))=q'_{(\varphi,\perp)}$。$\rho_2(\epsilon)=0=|\epsilon|=|g(\epsilon)|$。
- 其次，对 T 中每个目前无子节点的节点 x，归纳假设 $g(x)$ 存在，并且若 $\rho_1(x)=q_\psi$ 则有 $\rho'(g(x))=q'_{(\psi,\star)}$ 以及 $\rho_2(x)=|g(x)|$ 成立。由于 $\langle T',\rho'\rangle$ 是 π 在 $\mathcal{A}'_\varphi$ 上的一个运行，所以有 $\{\rho'(g(x)\cdot c)\mid g(x)\cdot c\in T'\}\models\delta'_\varphi(\rho'(g(x)),\pi(|g(x)|))=\delta'_\varphi(\rho'(g(x)),\pi(\rho_2(x)))$。

  若 $\rho_1(x)=q_p$（resp. $\rho_1(x)=q_{\neg p}$），则 $\delta'_\varphi(\rho'(g(x)),\pi(\rho_2(x)))$ 的值只能为 $true$ 或 $false$。由于其是可满足的，因此必为 $true$。这说明 $\pi(\rho_2(x))\models p$（resp. $\pi(\rho_2(x))\models\neg p$）。这时，令 x 为 T 中的一个叶节点。

  否则，不妨设 $\rho_1(x)=q_\psi$（其中 $\psi\notin AP\cup\overline{AP}$）。由于 $P'_x=\{\rho'(g(x)\cdot c)\mid g(x)\cdot c\in T'\}\models\delta'_\varphi(\rho'(g(x)),\pi(\rho_2(x)))=\theta(q'_{(\psi,\perp)},\pi(\rho_2(x)))$，所以由前面所证结论，必然存在 q_ψ 在 $\pi(\rho_2(x))$ 上的单步展开树 $\langle\hat{T}_x,\hat{\rho}_x\rangle$，使得 $P'_x\Vdash\langle\hat{T}_x,\hat{\rho}_x\rangle$。现在，向 T 中添加节点集合 $\{x\cdot y\mid y\in\hat{T}_x\}$。并且，对每个 $y\in\hat{T}_x$ 令 $\rho_1(x\cdot y)=\hat{\rho}_x(y)$；同时，若 y 是 $\hat{T}_x$ 中的非叶节点，则令 $\rho_2(x\cdot y)=\rho_2(x)$，若 y 是 $\hat{T}_x$ 中的叶节点，则令 $\rho_2(x\cdot y)=\rho_2(x)+1$。

  对于 $\hat{T}_x$ 中的每个叶节点 y，不妨设 $\hat{\rho}_x(y)=q_\phi$，由于 $P'_x\Vdash\theta(q'_{(\psi,\perp)},\pi(\rho_2(x)))$，所以由定义知存在 $\star\in VAR\cup\{\perp\}$ 使得 $q'_{(\phi,\star)}\in P'_x$。换言之，存在 $g(x)\cdot c\in T'$，使得 $\rho'(g(x)\cdot c)=q'_{(\phi,\star)}$。现令 $g(x\cdot y)=g(x)\cdot c$。则容易验证归纳不变式对每个 $x\cdot y$ 仍然成立。

根据上述构造过程，可以验证 $\langle T,\rho\rangle$ 中的每个节点均符合定义 4.3.2 中的要求，因

而 $\langle T,\rho\rangle$ 是 $\mathcal{A}_\varphi$ 在 π 上的一个运行。

对于 T 中的任意一条无穷路径 $\sigma = x_0, x_1, \ldots$，设 $\max(\mathbf{Inf}(\Omega_\varphi(\sigma))) = n$（这里 $\mathbf{Inf}(\Omega_\varphi(\sigma)) = \{m \in \mathbb{N} \mid$ 有无穷多个 $i \in \mathbb{N}$ 使得 $\Omega_\varphi(\rho(x_i)) = m\}$）。由构造，必然存在某个 $X \in VAR$ 以及无穷多个 $i \in \mathbb{N}$ 使得 $\rho_1(x_i) = q_X$，并且对于任意的 $Y \in VAR$，若存在无穷多个 $j \in \mathbb{N}$ 使得 $\rho_1(x_i) = q_Y$ 则 $Y \lhd_\varphi X$。由 $\langle T,\rho\rangle$ 的构造，在 σ 中必然包含无穷多个节点 $x'_0, x'_1, \ldots$ 使得 g 在每个 x'_i 上有定义。令 T' 中包含 $g(x'_0), g(x'_1), \ldots$ 的路径为 σ'，则必然存在某个 $\phi \in \mathbf{Sub}(\varphi)$ 以及无穷多个 $i \in \mathbb{N}$ 使得 $\rho'(x'_i) = q'_{(\phi,X)}$；同时，对于任意的 $Y \in VAR$ 以及 $\phi' \in \mathbf{Sub}(\varphi)$，若存在无穷多个 $j \in \mathbb{N}$ 使得 $\rho'(x'_j) = q'_{(\phi',Y)}$ 则必然有 $Y \lhd_\varphi X$。由于 $\langle T',\rho'\rangle$ 是 $\mathcal{A}'_\varphi$ 在 π 上的可接收运行，由 Ω'_φ 的定义知 X 必然是 φ 中的 ν-型约束变元。因此 n 是偶数。所以 $\langle T,\rho\rangle$ 是 $\mathcal{A}_\varphi$ 在 π 上的一个可接收运行，故而 $\pi \in \mathbf{L}(\mathcal{A}'_\varphi)$。 □

由定理 6.2 及定理 6.3，立即可以得到如下推论。

推论 6.4 设 $\mathcal{A}_\varphi$ 和 $\mathcal{A}'_\varphi$ 分别是由 φ 得到的 $SAPW$ 和 APW，则 $\mathbf{L}(\mathcal{A}_\varphi) = \mathbf{L}(\mathcal{A}'_\varphi)$。

6.2.2 从交错 parity 自动机到非确定 Büchi 自动机

上节，给出了从线性 μ-演算公式（博弈范式）到 APW 的转化过程。接下来，需要将 APW 转化至（广义的）NBW 。由于公平迁移系统与 NBW之间本质相同，这样执行线性 μ-演算的模型检验过程便可实现。在本小节，为了使过程更加通用，不再将自动机的字母表限定在 2^{AP} 上。以下，会用到 5.5.1 节中**层次子树同构** 的概念（见定义 5.5.2）。在本节，为简便起见，将非确定自动机在无穷字上的运行看作是状态序列。并将非确定自动机的迁移函数写成 $Q \times \Sigma \to 2^Q$ 的形式（其中 Q 和 Σ 分别是自动机的状态集和字母表）。

6.2.2.1 辅助概念和操作定义

在本节，仍用 $\mathbb{O}$ 和 $\mathbb{E}$ 表示奇数集和偶数集。并沿用 5.5.2 节引入的记法：对每个自然数 m，分别用 $\mathbb{N}[m]$、$\mathbb{O}[m]$、$\mathbb{E}[m]$ 表示由不大于 m 的自然数、奇数、偶数构成的集合（见 169 页公式 (5.13)～(5.15)）。

接下来，引入如下定义和操作，这些定义和操作会在后面的转化过程中使用。

定义 6.2.2 (NGBW) 一个**广义非确定 *Büchi* 自动机**（简记为$NGBW$）是一个序偶 $\langle \Sigma, Q, \delta, q, \Omega\rangle$。其中:

- Σ、Q、δ、q 定义同前。
- $\Omega = \mathcal{F} \subseteq 2^Q$，是该自动机的接收条件。

不妨设 $\mathcal{F}=\{F_1,\ldots,F_m\}$，其中 $F_i\subseteq Q$。则运行 σ 是可接收的当且仅当对每个 $1\leq i\leq m$ 都有 $\mathbf{Inf}(\sigma)\cap F_i\neq\emptyset$。 □

于是，任何一个 NBW 都可以看作是只有一个接收状态集的 NGBW。而 NGBW 之间可以方便的定义**乘积** 操作如下。

定义 6.2.3 给定两个 $NGBW$ $\mathcal{A}=\langle\Sigma,Q,\delta,q_0,\mathcal{F}\rangle$ 和 $\mathcal{A}=\langle\Sigma,Q',\delta',q_0',\mathcal{F}'\rangle$。则 $\mathcal{A}$ 和 $\mathcal{A}'$ 之间的**乘积**，记作 $\mathcal{A}\otimes\mathcal{A}'$，是一个 $NGBW$ $\langle\Sigma,Q\times Q',\delta'',(q_0,q_0'),\mathcal{F}''\rangle$ 其中：

- $\delta''((q,q'),a)=\{(q'',q''')\mid q''\in\delta(q,a),q'''\in\delta'(q',a)\}$;
- 不妨设 $\mathcal{F}=\{F_1,\ldots,F_m\}$，$\mathcal{F}'=\{F_1',\ldots,F_n'\}$，则 $\mathcal{F}''=\{F_i\times Q'\mid 1\leq i\leq m\}\cup\{Q\times F_j'\mid 1\leq j\leq n\}$。 □

容易验证，$\mathbf{L}(\mathcal{A}\otimes\mathcal{A}')=\mathbf{L}(\mathcal{A})\cap\mathbf{L}(\mathcal{A}')$。

文 [120] 中给出了从 ABW 到 NBW 的标准转化方法，称为**断点构造法**（Break Point Construction），具体过程如下。

给定 ABW $\mathcal{A}=\langle\Sigma,Q,\delta,q_0,F\rangle$，则可以构造等价的 NBW $\mathcal{A}'=\langle\Sigma,Q',\delta',q_0',F'\rangle$ 使得 $\mathbf{L}(\mathcal{A})=\mathbf{L}(\mathcal{A}')$。其中：

- $Q'=\{(P,P')\mid P\subseteq Q,P'\subseteq P\}$。
- 若 $P'\neq\emptyset$，则 $\delta'((P,P'),a)=\{(R,R'\backslash F)\mid$ 对每个 $q\in P$，有 $R\models\delta(q,a)$; 对每个 $q\in P',R'\models\delta(q,a)\}$；若 $P'=\emptyset$，则 $\delta'((P,P'),a)=\{(R,R\backslash F)\mid$ 对每个 $q\in P$, 有 $R\models\delta(q,a)\}$。
- $q_0'=(\{q_0\},\emptyset)$。
- $F'=2^Q\times\{\emptyset\}$。

为了编码方便起见，将 $\mathcal{A}'$ 的构造等价变形如下。令 $\mathcal{A}''=\langle\Sigma,Q'',\delta'',q'',F''\rangle$ 其中：

- $Q''=2^{Q\times\{+,-\}}$。
- 若 $P\not\subseteq Q\times\{+\}$，则 $\delta(P,a)=\{R\mid$ 对每个 $(q,+)\in P,\{q'\mid(q',+)\in R\}\models\delta(q,a)$; 对每个 $(q,-)\in P$ 有 $(\{q'\in Q\backslash F\mid(q',-)\in R\}\cup\{q'\in F\mid(q',+)\in R\})\models\delta(q,a)\}$；

  若 $P\subseteq Q\times\{+\}$，则 $\delta(P,a)=\{R\subseteq Q\times\{-\}\mid$ 对每个 $(q,+)\in P$, 有 $\{q'\mid(q',-)\in R\}\models\delta(q,a)\}$。
- $q''=\{(q_0,-)\}$。
- $F''=2^{Q\times\{+\}}$。

容易证明：$\mathbf{L}(\mathcal{A}')=\mathbf{L}(\mathcal{A}'')$。

定义 6.2.4 (自动机投影) 给定字母表 $\Sigma\times\Sigma'$ 上的非确定自动机 $\mathcal{A}=\langle\Sigma\times\Sigma',Q,\delta,q_0,$

$\Omega\rangle$，则 $\mathcal{A}$ 在 Σ 上的**投影**，记作 $\mathcal{A}|_\Sigma$，是一个非确定自动机 $\mathcal{A}' = \langle \Sigma, Q, \delta', q_0, \Omega\rangle$，其中 $\delta'(q,a) = \{q' \mid$ 存在 $a' \in \Sigma'$，**使得** $q' \in \delta(q,(a,a'))\}$。□

对称的，也可以定义 $\mathcal{A}$ 在 Σ' 上的投影 $\mathcal{A}|_{\Sigma'}$。一方面，对于任意的 $W = (a_0, a'_0), (a_1, a'_1), \ldots \in (\Sigma \times \Sigma')^*$，若 $\mathcal{A}$ 能够以运行 $q_0, q_1, \ldots$ 接收 W，则 $\mathcal{A}'$ 必然也能以该运行接收 $a_0, a_1, \ldots$。另一方面，若 $\mathcal{A}$ 能够以运行 $q'_0, q'_1, \ldots$ 接收 $w = a_0, a_1, \ldots$（这里 $q'_0 = q_0$），则由于 $q_{i+1} \in \delta'(q'_i, a_i)$，所以对每个 $i \in \mathbb{N}$，存在 a'_i 使得 $q'_{i+1} \in \delta(q'_i, (a_i, a'_i))$。于是，$\mathcal{A}$ 能够以运行 $q'_0, q'_1, \ldots$ 接收 $(a_0, a'_0), (a_1, a'_1), \ldots$。因此，有以下定理成立。

定理 6.5 设 $\mathcal{A}$ 是字母表 $\Sigma \times \Sigma'$ 上的非确定自动机，$\mathcal{A}' = \mathcal{A}|_\Sigma$，则对于任意的 $w = a_0, a_1, \ldots \in \Sigma^\omega$ 而言，$w \in \mathbf{L}(\mathcal{A}')$ 当且仅当存在 $w' = a'_0, a'_1, \ldots \in (\Sigma')^\omega$ 使得 $(a_0, a'_0), (a_1, a'_1), \ldots \in \mathbf{L}(\mathcal{A})$。

最后，引入**标记树编码**的定义。在文 [92] 中，Kupferman、Piterman 和 Vardi 等人利用标记树编码的思想给出了从双向交错自动机到单项非确定自动机的转化算法。（该文中输入自动机的接收条件为 hesitant 是一种综合了 Büchi 和 co-Büchi 的接收条件）。在下节，将会利用该手段给出从 APW 到 NGBW 的转化过程。

定义 6.2.5 (标记树编码) 任给一棵层次子树同构的 Q-标记树 $\langle T, \rho\rangle$，则 $\langle T, \rho\rangle$ 的一个**编码** 是一个 $2^{Q\times Q}$ 上的序列 $L_0, L_1, \ldots$。其中，对每个 $i \in \mathbb{N}$:

- 对每个 $x \in T$，若 $|x| = i$，则对于 x 的每个子节点 $x\cdot c$，有 $(\rho(x), \rho(x\cdot c)) \in L_i$。
- 对每个 $(q, q') \in L_i$，存在 $x \in T$ 以及 x 的某个子节点 $x\cdot c$ 使得 $|x| = i$，$\rho(x) = q$ 以及 $\rho(x \cdot c) = q'$。

称 $q_0, q_1, \ldots$ 是 $L_0, L_1, \ldots$ 中的一条**踪迹** 如果对每个 $i \in \mathbb{N}$ 都有 $(q_i, q_{i+1}) \in L_i$。□

例 6.2.2 若 Q 是有穷集合，则使用上述方法便可将一棵无穷树编码至有穷的字母表上。设 $Q = \{q_0, q_1, q_2, q_3\}$，则图 *6.1(a)* 中所示的有穷 Q-标记树（容易验证该标记树是层次子树同构的）的编码如图 *6.1(b)* 所示。□

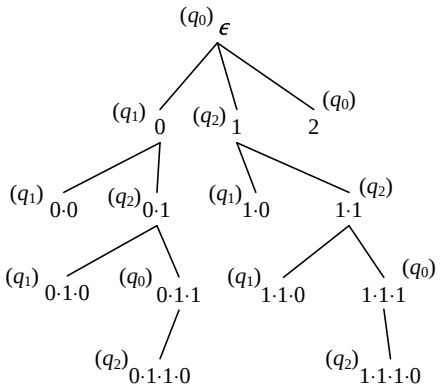
类似于定理 5.14，可证明如下结论。

引理 6.6 对于任意的 *APW* $\mathcal{A}$ 以及线性结构 π 而言，π 在 $\mathcal{A}$ 上有一个可接收运行当且仅当 π 在 $\mathcal{A}$ 上有一个层次子树同构的可接收运行。

6.2.2.2 转化过程

现在，给出从 APW 到 NBW 的转化过程。

步骤一. 首先，从输入的 APW $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, \Omega\rangle$ 按照如下方式得到两个以 $\Sigma \times 2^{Q\times Q}$ 为字母表的自动机 $\mathcal{A}_1$ 和 $\mathcal{A}_2$。



(a) 编码前的标记树

$L_0 = \{(q_0, q_1), (q_0, q_2), (q_0, q_0)\}$

$L_1 = \{(q_1, q_1), (q_1, q_2), (q_2, q_1), (q_2, q_2)\}$

$L_2 = \{(q_2, q_1), (q_2, q_0)\}$

$L_3 = \{(q_0, q_2)\}$

(b) 标记树编码

图 6.1 标记树编码示例

$\mathcal{A}_1$ 是一个 DLW $\langle \Sigma \times 2^{Q\times Q}, 2^Q, \delta_1, \{q_0\}, \Omega_1\rangle$。其中，对任意的 $P \subseteq Q$ 以及任意的 $(a, L) \in \Sigma \times 2^{Q\times Q}$，如果：

- $\{q \in Q \mid$ 存在 $q' \in Q$ 使得 $(q, q') \in L\} \subseteq P$；
- 对每个 $q \in P$，集合 $\{q' \mid (q, q') \in L\}$ 满足 $\delta(q, a)$

则令 $\delta_1(P, (a, L)) = \{q' \in Q \mid$ 存在 $q \in Q$ 使得 $(q, q') \in L\}$；否则，令 $\delta_1(P, (a, L)) = \emptyset$。

$\mathcal{A}_2$ 是一个 NPW $\langle \Sigma \times 2^{Q\times Q}, Q, \delta_2, q_0, \Omega_2\rangle$，其中 Ω_2 在 q 处有定义当且仅当 Ω 在 q 处有定义，并且 $\Omega_2(q) = \Omega(q) + 1$。此外，对每个 $q \in Q$ 以及 $(a, L) \in \Sigma \times 2^{Q\times Q}$ 有：$\delta(q, (a, L)) = \bigvee_{(q,q')\in L} q'$。

于是，对于任意的 $\sigma \in Q^\omega$ 而言，$\mathbf{Inf}(\sigma)$ 满足 Ω 当且仅当 $\mathbf{Inf}(\sigma)$ 违反 Ω_2。由构造，很容易证明下面引理。

引理 6.7 *对每个 $W = (a_0, L_0), (a_1, L_1), \ldots \in (\Sigma \times 2^{Q\times Q})^\omega$ 而言，$W \in \mathbf{L}(\mathcal{A}_1)$ 当且仅当 $L_0, L_1, \ldots$ 是 $a_0, a_1, \ldots$ 在 $\mathcal{A}$ 上一个(层次子树同构)运行的编码。$W \in \mathbf{L}(\mathcal{A}_2)$ 当且仅当 $L_0, L_1, \ldots$ 中包含一条无穷踪迹 σ 使得 $\mathbf{Inf}(\sigma)$ 违反 Ω。*

推论 6.8 *对每个 $W = (a_0, L_0), (a_1, L_1), \ldots \in (\Sigma \times 2^{Q\times Q})^\omega$ 而言，$W \in \mathbf{L}(\mathcal{A}_1)\backslash\mathbf{L}(\mathcal{A}_2)$ 当且仅当 $L_0, L_1, \ldots$ 是 $\mathcal{A}$ 在 $a_0, a_1, \ldots$ 上一个(层次子树同构的)可接收运行的编码。*

步骤二. 将由步骤一中得到的 $\mathcal{A}_2$ 转化为若干个 NBW。设 $\max\{\Omega_2(q) \mid q \in Q\} = n$，则为每个 $m \in \mathbb{E}[n]\backslash\{0\}$ 创建一个 NBW $\mathcal{A}_{3,m} = \langle \Sigma \times 2^{Q\times Q}, Q_{3,m}, \delta_{3,m}, (q_0, 0), F_{3,m}\rangle$。其中：

- $Q_{3,m} = (Q \times \{0\}) \cup (Q\backslash\{q \mid \Omega_2(q) > m\}) \times \{m\}$。

- 对于每个 $q \in Q$ 以及 $(a, L) \in \Sigma \times 2^{Q \times Q}$ 有：$\delta_{3,m}((q,0),(a,L)) = \{(q',0) \mid q' \in \delta_2(q,(a,L))\} \cup \{(q',m) \in Q_{3,m} \mid q' \in \delta_2(q,(a,L))\}$；
  对每个 $(q,m) \in Q_{3,m}$ 以及 $(a,L) \in \Sigma \times 2^{Q \times Q}$ 有：$\delta_{3,m}((q,m),(a,L)) = \{(q',m) \in Q_m \mid q' \in \delta_2(q,(a,L))\}$。
- $F_{3,m} = \{(q,m) \mid \Omega_2(q) = m\}$。

引理 6.9 $\mathbf{L}(\mathcal{A}_2) = \bigcup_{m \in \mathbb{E}[n] \setminus \{0\}} \mathbf{L}(\mathcal{A}_{3,m})$。

证明. 假设 $\mathcal{A}_2$ 以运行 $q_0, q_1, \ldots$ 接收 W，则必然存在某个 $m \in \mathbb{E}[n] \setminus \{0\}$ 使得有无穷多个 $i \in \mathbb{N}$（注意，Ω_2 值域中的最小值大于 0），满足 $\Omega_2(q_i) = m$，以及存在某个 $j \in \mathbb{N}$ 使得对任意的 $k \geq j$ 有 $\Omega(q_k) \leq m$。因此，$\mathcal{A}_{3,m}$ 能够以接收运行 $(q_0,0), \ldots, (q_j,0), (q_{j+1},m), (q_{j+2},m), \ldots$ 接收 W。

反之，若 $\mathcal{A}_{3,m}$ 以运行 $(q_0,0), \ldots, (q_j,0), (q_{j+1},m), (q_{j+2},m), \ldots$ 接收 W，则对于任意的 $k \geq j$ 有 $\Omega_2(q_k) \leq m$；并且，有无穷多个 $i \in \mathbb{N}$ 使得 $\Omega_2(q_i) = m$。因此，$\mathcal{A}_2$ 能以运行 $q_0, q_1, \ldots$ 接收 W。 □

第三步. 将每个 $\mathcal{A}_{3,m}$ 求补。即构造 NBW $\mathcal{A}_{4,m}$ 使得 $\mathbf{L}(\mathcal{A}_{4,m}) = (\Sigma \times 2^{Q \times Q})^\omega \setminus \mathbf{L}(\mathcal{A}_{3,m})$。

首先，由 $\mathcal{A}_{3,m}$ 构造 ABW $\mathcal{A}'_{4,m}$，使得 $\mathbf{L}(\mathcal{A}'_{4,m}) = (\Sigma \times 2^{Q \times Q})^\omega \setminus \mathbf{L}(\mathcal{A}_{3,m})$。这里，为了方便下一节迁移系统的构造以及符号化编码，采用文 [121] 中的构造（其他构造方法见 [114, 122, 115, 119] 等）。

设 $\#Q_{3,m} = k$，则 $\mathcal{A}'_{4,m} = \langle \Sigma \times 2^{Q \times Q}, Q'_{4,m}, \delta'_{4,m}, q'_{4,m}, F'_{4,m} \rangle$。其中：

- $Q'_{4,m} = Q_{3,m} \times \mathbb{N}[2k+1]$。以下，为简单起见，将 $Q'_{4,m}$ 中的状态 $((q,0),t)$ 直接写作 $(q,0,t)$，将 $((q,m),t)$ 直接写作 (q,m,t)。
- 对每个 $(q,0,t) \in Q'_{4,m}$ 以及 $(a,L) \in \Sigma \times 2^{Q \times Q}$，

$$\delta'_{4,m}((q,0,t),(a,L)) = \left(\bigwedge_{q' \in \delta_2(q,(a,L))} \bigvee_{t' \leq t} (q',0,t')\right) \wedge \left(\bigwedge_{\substack{q' \in \delta_2(q,(a,L)) \\ (q',m) \in Q_{3,m}}} \bigvee_{t' \leq t} (q',m,t')\right)$$

对每个 $(q,m,t) \in Q'_{4,m}$ 以及 $(a,L) \in \Sigma \times 2^{Q \times Q}$

$$\delta'_{4,m}((q,m,t),(a,L)) = \begin{cases} false & ,\omega_2(q) = m \text{ 且 } t \in \mathbb{O} \\ \bigwedge_{\substack{q' \in \delta_2(q,(a,L)) \\ (q',m) \in Q_{3,m}}} \bigvee_{t' \leq t} (q',m,t') & ,\text{否则} \end{cases}。$$

- $q'_{4,m} = (q_0, 0, 2k+1)$。
- $F'_{4,m} = Q_{3,m} \times \mathbb{O}[2k+1]$。

接下来，再令 $\mathcal{A}_{4,m}$ 是由 $\mathcal{A}'_{4,m}$ 经（变形的）断点构造法得到的 NBW 即可。

步骤四，得到与 $\mathcal{A}$ 等价的 NGBW $\mathcal{A}_5$。事实上，只要令

$$\mathcal{A}_5 = (\mathcal{A}_1 \otimes \bigotimes\nolimits_{m\in\mathbb{E}[n]\setminus\{0\}} \mathcal{A}_{4,m})|_{\Sigma}$$

即可。下面证明 $\mathcal{A}_5$ 与 $\mathcal{A}$ 等价。

定理 6.10 $\mathbf{L}(\mathcal{A}) = \mathbf{L}(\mathcal{A}_5)$。

证明. 首先说明：$\mathbf{L}(\mathcal{A}_2) = (\Sigma \times 2^{Q\times Q})^{\omega}\backslash\mathbf{L}(\bigotimes_{m\in\mathbb{E}[n]\setminus\{0\}} \mathcal{A}_{4,m})$。

由构造步骤三知，对于每个 $m \in \mathbb{E}[n]\setminus\{0\}$ 都有 $\mathcal{A}_{4,m} = (\Sigma \times 2^{Q\times Q})^{\omega}\backslash\mathbf{L}(\mathcal{A}_{3,m})$。因此，$\mathbf{L}(\bigotimes_{m\in\mathbb{E}[n]\setminus\{0\}}) = (\Sigma \times 2^{Q\times Q})^{\omega}\backslash \bigcup_{m\in\mathbb{E}[n]\setminus\{0\}} \mathbf{L}(\mathcal{A}_{3,m})$。于是，由引理 6.9，立即有 $\mathbf{L}(\mathcal{A}_2) = (\Sigma \times 2^{Q\times Q})^{\omega}\backslash\mathbf{L}(\bigotimes_{m\in\mathbb{E}[n]\setminus\{0\}} \mathcal{A}_{4,m})$ 成立。

换言之，$\mathbf{L}(\bigotimes_{m\in\mathbb{E}[n]\setminus\{0\}} \mathcal{A}_{4,m}) = (\Sigma \times 2^{Q\times Q})^{\omega}\backslash\mathbf{L}(\mathcal{A}_2)$。于是，由自动机乘积的性质，立即有 $\mathbf{L}(\mathcal{A}_1 \otimes \bigotimes_{m\in\mathbb{E}[n]\setminus\{0\}} \mathcal{A}_{4,m}) = \mathbf{L}(\mathcal{A}_1)\backslash\mathbf{L}(\mathcal{A}_2)$。

由定理 6.5，对于任意的 $w = a_0, a_1, \ldots \in \Sigma^{\omega}$ 而言，$w \in \mathbf{L}(\mathcal{A}_5)$ 当且仅当存在 $W = L_0, L_1, \ldots \in (2^{Q\times Q})^{\omega}$ 使得 $(a_0, L_0), (a_1, L_1), \ldots \in \mathbf{L}(\mathcal{A}_1)\backslash\mathbf{L}(\mathcal{A}_2)$。由引理 6.7，这当且仅当 W 是 $\mathcal{A}$ 在 w 上一个可接收运行的编码。

因此，$w \in \mathbf{L}(\mathcal{A}_5)$ 当且仅当 $\mathcal{A}$ 在 w 上存在一个可接收运行，当且仅当 $w \in \mathbf{L}(\mathcal{A})$。于是，$\mathbf{L}(\mathcal{A}) = \mathbf{L}(\mathcal{A}_5)$。 □

6.2.3 检验算法及符号化实现

在 6.2.1 节及 6.2.2 节分别给出了从线性 μ-演算公式到 APW 以及从 APW 到 NGBW 的转化过程。在本节，将会结合上述两个过程给出线性 μ-演算的符号化模型检验算法。

给定写成博弈范式的线性 μ-演算公式 φ，由推论 6.4，可以得到一个 APW $\mathcal{A}'_{\varphi}$，使得对任意的线性结构 π 而言，$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{A}'_{\varphi})$ 成立。此外，以 $\mathcal{A}'_{\varphi}$ 作为输入，由 6.2.2 节构造，可以得到一个与之等价的 NBGW。

此外，给定以 2^{AP} 为字母表的 NGBW（本节假设 AP 为有穷集）$\langle 2^{AP}, Q, \delta, q_0, \{F_1, \ldots, F_m\}\rangle$，可以按照如下标准的方法获得一个公平迁移系统 $\mathcal{M} = \langle S, \Delta, I, \lambda, \mathcal{C}\rangle$。其中：

- $S = Q \times 2^{AP}$。
- $((q_1, a_1), (q_2, a_2)) \in \Delta$ 当且仅当 $q_2 \in \delta(q_1, a_1)$。
- $I = \{q_0\} \times 2^{AP}$。
- $\lambda((q, a)) = a$。
- $\mathcal{C} = \{(F_1 \times 2^{AP}) \cap S, \ldots, (F_m \times 2^{AP}) \cap S\}$。

容易验证 $\mathbf{L}(\mathcal{A}) = \mathbf{L}(\mathcal{M})$。

对于线性 μ-演算公式 φ，对任意的 $L \subseteq Q'_\varphi \times Q'_\varphi$，令

$$\mathbf{Src}(L) \stackrel{\text{def}}{=} \{q'_{(\psi,\star)} \in Q'_\varphi \mid \text{存在 } q'_{(\psi',\star')} \text{使得 } (q'_{(\psi,\star)}, q'_{(\psi',\star')}) \in L\} \tag{6.1}$$

$$\mathbf{Tar}(L) \stackrel{\text{def}}{=} \{q'_{(\psi,\star)} \in Q'_\varphi \mid \text{存在 } q'_{(\psi',\star')} \text{使得 } (q'_{(\psi',\star')}, q'_{(\psi,\star)}) \in L\} \tag{6.2}$$

此外，对每个 $q'_{(\psi,\star)} \in Q'_\varphi$，令

$$L(q'_{(\psi,\star)}) \stackrel{\text{def}}{=} \{q'_{(\psi',\star')} \mid (q'_{(\psi,\star)}, q'_{(\psi',\star')}) \in L\} \tag{6.3}$$

$$L^{-1}(q'_{(\psi,\star)}) \stackrel{\text{def}}{=} \{q'_{(\psi',\star')} \mid (q'_{(\psi',\star')}, q'_{(\psi,\star)}) \in L\} \tag{6.4}$$

以上，$\star, \star' \in VAR \cup \{\perp\}$。

设 φ 是写成博弈范式的线性 μ-演算公式，并设 $X_1, \ldots, X_n$ 为 φ 中所有的 μ-型约束变元。回忆 $\mathcal{A}'_\varphi$ 中 Ω'_φ 的定义：对每个 $\phi \in \mathbf{Sub}(\varphi)$，$\Omega'_\varphi(q'_{(\phi,X_i)})$ 都赋予一个奇数。为方便起见，假设 $\Omega_\varphi(q_{X_i}) = \Omega'(q'_{(\phi,X_i)})$（其中 Ω_φ 是 φ 对应的 SAPW $\mathcal{A}_\varphi$ 的接收条件）。由 Ω'_φ 的定义，对任意的 $\psi_1, \psi_2 \in \mathbf{Sub}(\varphi)$ 有 $\Omega'_\varphi(q_{(\psi_1,X_i)}) = \Omega'_\varphi(q_{(\psi_2,X_i)})$ 于是，由 6.2.2 节中的构造步骤二所得的 Ω_2（即 $\mathcal{A}_2$ 的接收条件）的值域中大于 0 的偶数的数目恰好为 Ω'_φ 值域中奇数的数目，也即 n。于是，步骤三中由 $\mathcal{A}_2$ 得到的 NBW 恰好有 n 个。并且，由转化过程知，这些 NBW 的状态数目均不超过 $2 \times (\#Q'_\varphi)$。而对于公式 φ 中的每个 μ-型约束变元，又可以得到如下的派生迁移关系。

定义 6.2.6 设 X 是线性 μ-演算公式 φ 中的 μ-型约束变元，$K = 4 \times (\#Q'_\varphi) + 1$，$k = \Omega_\varphi(q_X) + 1$，则可以定义派生迁移关系：$\Delta_{X,\varphi} \subseteq (2^{Q'_\varphi \times \{0,k\} \times \mathbb{N}[K] \times \{+,-\}}) \times (2^{Q'_\varphi \times Q'_\varphi}) \times (2^{AP}) \times (2^{Q'_\varphi \times \{0,k\} \times \mathbb{N}[K] \times \{+,-\}})$。设 $L \subseteq Q'_\varphi \times Q'_\varphi$，$a \subseteq AP$，$P, P' \subseteq Q'_\varphi \times \{0,k\} \times \mathbb{N}[K] \times \{+,-\}$，则 $(P, a, L, P') \subseteq \Delta_{X,\varphi}$ 当且仅当：

1. 对每个 $(q'_{(\psi,\star)}, l, t, \pm) \in P$，或者 $q'_{(\psi,\star)} \in \mathbf{Src}(L)$；或者 $\psi \in AP \cup \overline{AP} \cup \{true\}$，且 $a \models \psi$。
2. 对每个 $(q'_{(\psi,\star)}, l, t, \pm) \in P'$，有 $q'_{(\psi,\star)} \in \mathbf{Tar}(L)$。
3. 若 $(q'_{(\psi,X)}, \Omega_\varphi(q_X) + 1, t, \pm) \in P \cup P'$，则 t 必须是偶数。
4. 若 $P \subseteq Q'_\varphi \times \{0,k\} \times \mathbb{N}[K] \times \{+\}$，则 $P' \subseteq Q'_\varphi \times \{0,k\} \times \mathbb{N}[K] \times \{-\}$。
5. 对每个 $(q'_{(\psi,\star)}, 0, t, \pm) \in P$，要求：若 $q'_{(\psi',\star')} \in L(q'_{(\psi,\star)})$ 则存在 $t', t'' \leq t$，$\pm', \pm'' \in \{+,-\}$ 使得$(q'_{(\psi',\star')}, 0, t', \pm') \in P'$ 以及$(q'_{(\psi',\star')}, k, t'', \pm'') \in P'$。同时，当 $\pm = -$ 时，要求$\pm' = +$（*resp.* $\pm'' = +$）仅当 t'（*resp.* t''）是奇数。

6. 对每个 $(q'_{(\psi,\star)},k,t,\pm)\in P$，要求：若 $q'_{(\psi',\star')}\in L(q'_{(\psi,\star)})$，则存在 $t'\leq t$，$\pm'\in\{+,-\}$ 使得$(q'_{(\psi',\star')},k,t',\pm')\in P'$。同时，当 $\pm=-$ 时，要求$\pm'=+$ 仅当 t' 是奇数。 □

假设 $X=X_i$。事实上，上述定义模拟了 (a,L) 在步骤三中得到的 NBW $\mathcal{A}_{4,i}$ 上的一步运行。于是，从 $\mathcal{A}'_\varphi$ 可以得到如下的公平迁移系统 $\mathcal{M}_\varphi=\langle S_\varphi,\Delta_\varphi,I_\varphi,\lambda_\varphi,\mathcal{C}_\varphi\rangle$。其中

- S_φ 由所有满足如下约束的序偶 $\langle a,L,(P_1,\ldots,P_n)\rangle$ 构成：
  - $a\subseteq AP$。
  - $L\subseteq Q'_\varphi\times Q'_\varphi$，且满足约束：对于任意的 $q'_{(\psi,\star)}\in\mathbf{Src}(L)$，集合 $\{q'_{(\psi',\star')}\mid(q'_{(\psi,\star)},q'_{(\psi',\star')})\}$ 满足 $\theta(q'_{(\psi,\perp)},a)$。
  - 令 $K=4\times(\#Q'_\varphi)+1$，则对每个 $1\leq i\leq n$，$P_i\subseteq Q'_\varphi\times\{0,\Omega_\varphi(q_{X_i})+1\}\times\mathbb{N}[K]\times\{+,-\}$。
- $(\langle a,L,(P_1,\ldots,P_n)\rangle,\langle a',L',(P'_1,\ldots,P'_n)\rangle)\in\Delta_\varphi$ 当且仅当：
  - $\mathbf{Src}(L')\subseteq\mathbf{Tar}(L)$，对于任意的 $q'_{(\psi,\star)}\in\mathbf{Tar}(L)\backslash\mathbf{Scr}(L')$，一定有 $\psi\in AP\cup\overline{AP}$，以及 $a'\models\psi$。
  - 对每个 $1\leq i\leq n$，有 $(P_i,a,L,P'_i)\in\Delta_{X_i,\varphi}$。
- I_φ 是所有满足如下约束的状态 $\langle a,L,(P_1,\ldots,P_n)\rangle$ 构成的集合：
  - $q'_{(\varphi,\perp)}\in\mathbf{Scr}(L)$；
  - 对每个$1\leq i\leq n$ 都有 $(P_{i,0},a,L,P_i)\in\Delta_{X_i,\varphi}$。其中 $P_{i,0}=\{(q'_{(\varphi,\perp)},0,\Omega_\varphi(q_{X_i})+1,-)\}$。
- $\lambda_\varphi(\langle a,L,(P_1,\ldots,P_n)\rangle)=a$。
- $\mathcal{C}_\varphi=\{C_1,\ldots,C_n\}$，其中 $C_i=\{\langle a,L,(P_1,\ldots,P_n)\rangle\mid P_i\subseteq Q'_\varphi\times\{0,\Omega_\varphi(q_{X_i})+1\}\times\mathbb{N}[K]\times\{+\}\}$。

之所以令 $K=4\times(\#Q'_\varphi)+1$，是因为每个 $\mathcal{A}_{3,m}$ 的状态数不超过 $2\times(\#Q'_\varphi)$（见步骤三）。事实上，在状态 $\langle a,L,(P_1,\ldots,P_n)\rangle$ 中：L 描述了 (a,L) 在步骤一中得到的 DLW 上的一步运行（注意，读入 (a,L) 后到达的状态恰为 $\mathbf{Tar}(L)$，因此没有对其状态进行单独记录）。如前所述，P_i 恰好描述了 (a,L) 在步骤三中得到的 NBW 上的一步运行；公平约束 C_i 刻画了该 NBW 的接收条件。最后，λ_φ 完成了步骤四中的投影操作。因此，$\mathcal{M}_\varphi$ 恰好是由 $\mathcal{A}'_\varphi$ 得到的 NGBW 对应的公平迁移模型。于是，立即有如下结论。

定理 6.11 对于任意的线性结构 π 而言，$\pi\models\varphi$ 当且仅当 $\pi\in\mathbf{L}(\mathcal{M}_\varphi)$。

于是，对于任意的公平迁移系统 $\mathcal{M}$ 以及线性 μ-演算公式 φ，$\mathcal{M}\models\varphi$ 当且仅

当 $\mathbf{L}(\mathcal{M}) \subseteq \mathbf{L}(\mathcal{M}_\varphi)$，当且仅当 $\mathbf{L}(\mathcal{M} \| \mathcal{M}_{\neg\varphi}) = \emptyset$。因此，线性 μ-演算的模型检验问题可以转化为 CTL 的模型问题。

定理 6.12 给定公平迁移系统 $\mathcal{M}$ 以及线性 μ-演算公式 φ，$\mathcal{M} \models \varphi$ 当且仅当 $\mathcal{M} \| \mathcal{M}_{\neg\varphi} \not\models \mathsf{EG}\ true$。

任给线性 μ-演算公式 φ，现在给出 $\mathcal{M}_\varphi$ 基于 BDD 的编码方法。仍设 $X_1, \ldots, X_n$ 是 φ 中所有的 μ-型约束变元，$K = 4 \times (\#Q'_\varphi) + 1$。则编码迁移系统 $\mathcal{M}_\varphi$ 所需的要素如下。

位变元集合：由于 S_φ 中的每个状态是一个序偶 $\langle a, L, (P_1, \ldots, P_n)\rangle$，于是，需要引入下列位变元。

- 对每个 $p \in AP$，引入一个位变元 z_p。
- 对每对 $(q'_{(\psi,\star)}, q'_{(\psi',\star')}) \in Q'_\varphi \times Q'_\varphi$，引入一个位变元 $y_{(\psi,\star,\psi',\star')}$，（其中，$\star, \star' \in VAR \cup \{\bot\}$）。
- 对每个 $1 \le i \le n$，$q'_{(\psi,\star)} \in Q'_\varphi$，$l \in \{0, \Omega_\varphi(q_{X_i})+1\}$，$t \le K$ 以及 $\pm \in \{+, -\}$，引入一个位变元 $u_{(i,\psi,\star,l,t,\pm)}$。

状态约束：首先，为 Q'_φ 中的每对状态 $q'_{(\psi,\star)}$ 以及 $q'_{(\psi',\star')}$，递归定义函数 $\vartheta(\psi, \star, \psi', \star')$ 如下。

- $\vartheta(\psi, \star, true, \star') = true$；$\vartheta(\psi, \star, false, \star') = false$。
- 若 $\psi' = p \in AP$，则 $\vartheta(\psi, \star, \psi', \star') = z_p$；若 $\psi' = \neg p \in \overline{AP}$，则 $\vartheta(\psi, \star, \psi', \star') = \neg z_p$；若 $\psi' = \bigcirc\psi''$，则 $\vartheta_i(\psi, \star, \psi', \star') = y_{(\psi,\star,\psi'',\star')}$。
- 若 $\psi' = X \in VAR$，则分两种情况：如果 $\star = \bot$，或者 $\star = Y$ 并且 $Y \lhd_\varphi X$，那么 $\vartheta(\psi, \star, \psi', \star') = \vartheta(\psi, \star, \mathbf{D}_\varphi(X), X)$；如果 $\star = Y$ 并且 $X \lhd_\varphi Y$，那么 $\vartheta(\psi, \star, \psi', \star') = \vartheta(\psi, \star, \mathbf{D}_\varphi(X), Y)$。
- 若 $\psi' = \psi'_1 \vee \psi'_2$，则 $\vartheta(\psi, \star, \psi', \star') = \vartheta(\psi, \star, \psi'_1, \star') \vee \vartheta(\psi, \star, \psi'_2, \star')$。若 $\psi' = \psi'_1 \wedge \psi'_2$，则 $\vartheta(\psi, \star, \psi', \star') = \vartheta(\psi, \star, \psi'_1, \star') \wedge \vartheta(\psi, \star, \psi'_2, \star')$。
- 若 $\psi' = \mu X.\psi''$ 或者 $\psi' = \nu X.\psi''$，则 $\vartheta(\psi, \star, \psi', \star') = \vartheta(\psi, \star, \psi'', \star')$。

于是，状态约束编码 Φ_{S_φ} 为：

$$\bigwedge_{\psi,\star}((\bigvee_{\psi',\star'} y_{(\psi,\star,\psi',\star')}) \rightarrow \vartheta(\psi, \star, \psi, \bot)) \tag{6.5}$$

这里，$\bigwedge_{\psi,\star}$ 和 $\bigvee_{\psi',\star'}$ 分别是 $\bigwedge_{\psi\in\mathbf{Sub}(\varphi)} \bigwedge_{\star\in VAR(\varphi)\cup\{\bot\}}$ 和 $\bigvee_{\psi'\in\mathbf{Sub}(\varphi)} \bigvee_{\star'\in VAR(\varphi)\cup\{\bot\}}$ 的缩写（其中 $VAR(\varphi)$ 是在 φ 中出现的变元集合）。以下类似。

迁移关系：迁移关系的编码 Φ_{Δ_φ} 是下列布尔公式的合取。

$$\bigwedge_{\psi,\star}(\bigvee_{\psi',\star'} y'_{(\psi,\star,\psi',\star')} \rightarrow \bigvee_{\psi'',\star''} y_{(\psi'',\star'',\psi,\star)}) \tag{6.6}$$

$$\bigwedge_{\psi\notin AP\cup\overline{AP},\star}(\bigvee_{\psi'',\star''} y_{(\psi'',\star'',\psi,\star)} \rightarrow \bigvee_{\psi',\star'} y'_{(\psi,\star,\psi',\star')}) \tag{6.7}$$

$$\bigwedge_{\psi\in AP\cup\overline{AP},\star}((\bigvee_{\psi',\star'} y_{(\psi',\star',\psi,\star)} \wedge \bigwedge_{\psi'',\star''} \neg y'_{(\psi,\star,\psi'',\star'')}) \rightarrow \beta_\psi) \tag{6.8}$$

其中，若 $\psi = p \in AP$，则 $\beta_\psi = z'_p$；若 $\psi = \neg p \in \overline{AP}$，则 $\beta_\psi = \neg z'_p$。

$$\bigwedge_{1\le i\le n}((\bigwedge_{\substack{\psi\notin AP\cup\overline{AP}\\ l\in\{0,k_i\},t\in\mathbb{N}[K]}}\bigwedge_{\star,\pm} u_{(i,\psi,\star,l,t,\pm)}) \rightarrow \bigvee_{\psi',\star'} y'_{(\psi,\star,\psi',\star')}) \tag{6.9}$$

$$\bigwedge_{1\le i\le n}((\bigwedge_{\substack{\psi\in AP\cup\overline{AP},\\ l\in\{0,k_i\},t\in\mathbb{N}[K]}}\bigwedge_{\star,\pm} u_{(i,\psi,\star,l,t,\pm)}) \rightarrow \beta_\psi) \tag{6.10}$$

$$\bigwedge_{1\le i\le n}((\bigwedge_{\substack{l\in\{0,k_i\}\\ t\in\mathbb{N}[K]}}\bigwedge_{\psi,\star,\pm} u'_{(i,\psi,\star,l,t,\pm)}) \rightarrow \bigvee_{\psi',\star'} y'_{(\psi',\star',\psi,\star)}) \tag{6.11}$$

其中，$K = 4\times(\#Q'_\varphi)+1$，$k_i = \Omega_\varphi(q_{X_i})+1$。

$$\bigwedge_{1\le i\le n}\bigwedge_{\substack{\psi,\pm,k_i\\ t\in\mathbb{O}[K]}}(\neg u_{(i,\psi,\star,l,k_i,\pm)} \wedge \neg u'_{(i,\psi,\star,l,k_i,\pm)}) \tag{6.12}$$

$$\bigwedge_{1\le i\le n}(\bigwedge_{\psi,\star,l,t}\neg u_{(i,\psi,\star,l,t,-)} \rightarrow \bigwedge_{\psi,\star,l,t}\neg u'_{(i,\psi,\star,l,t,+)}) \tag{6.13}$$

$$\begin{aligned}\bigwedge_{1\le i\le n}(\bigwedge_{\psi,\star,t} u_{(i,\psi,\star,0,t,-)} \rightarrow \bigwedge_{\psi',\star'}(y'_{(\psi,\star,\psi',\star')} \rightarrow \\ \bigwedge_{l\in\{0,k_i\}}(\bigvee_{t'\le t} u'_{(i,\psi',\star',l,t',-)} \vee \bigvee_{t''\in\mathbb{O}[t]} u'_{(i,\psi',\star',l,t'',+)})))\end{aligned} \tag{6.14}$$

$$\bigwedge_{1\le i\le n}(\bigwedge_{\psi,\star,t} u_{(i,\psi,\star,0,t,+)} \rightarrow \bigwedge_{\psi',\star'}(y'_{(\psi,\star,\psi',\star')} \rightarrow \bigwedge_{l\in\{0,k_i\}}(\bigvee_{t'\le t,\pm} u'_{(i,\psi',\star',l,t',\pm)}))) \tag{6.15}$$

$$\begin{aligned}\bigwedge_{1\le i\le n}(\bigwedge_{\psi,\star,t} u_{(i,\psi,\star,k_i,t,-)} \rightarrow \bigwedge_{\psi',\star'}(y'_{(\psi,\star,\psi',\star')} \rightarrow \\ (\bigvee_{t'\le t} u'_{(i,\psi',\star',k_i,t',-)} \vee \bigvee_{t''\in\mathbb{O}[t]} u'_{(i,\psi',\star',k_i,t'',+)})))\end{aligned} \tag{6.16}$$

$$\bigwedge_{1\le i\le n}(\bigwedge_{\psi,\star,t} u_{(i,\psi,\star,k_i,t,+)} \rightarrow \bigwedge_{\psi',\star'}(y'_{(\psi,\star,\psi',\star')} \rightarrow (\bigvee_{t'\le t,\pm} u'_{(i,\psi',\star',k_i,t',\pm)}))) \tag{6.17}$$

初始状态集：根据 I_φ 的定义，其布尔编码 Φ_{I_φ} 应为 $\bigvee_{\psi,\star} y_{(\varphi,\perp,\psi,\star)}$ 与

$$\bigwedge_{1\le i\le n}\Big(\bigwedge_{\psi,\star}(y_{(\varphi,\perp,\psi,\star)}\rightarrow \bigwedge_{l\in\{0,k_i\}}\bigvee_{t\le K}(u_{(i,\psi,\star,l,t,-)}\vee\bigvee_{t'\in\mathbb{O}[K]}u_{(i,\psi,\star,l,t,+)}))\Big) \tag{6.18}$$

的合取。

标记函数：对每个 $p\in AP$，$\Phi^p_{\lambda_\varphi}=z_p$。

公平性约束：公平性约束的编码为 $\Phi_{\mathcal{C}_\varphi}=\{\Phi_{C_1},\dots\Phi_{C_n}\}$。其中，对每个 $1\le i\le n$，Φ_{C_i} 表示为 $\bigwedge_{\psi,\star,l,t}\neg u_{(i,\psi,\star,l,t,-)}$。

可以验证，上述过程得到的恰为 $\mathcal{M}_\varphi$ 的 BDD 编码。其中，所需位变元的数目为 $\mathcal{O}(|\varphi|^4)$。确切的说：设 φ 中形如 $\bigcirc\psi$ 的子公式数目为 K，约束变元的数目为 N，则对 $\mathcal{M}_\varphi$ 编码所需的位变元数目为 $\mathcal{O}(K^2\times N^2)$。

6.3 特定形式的线性 μ-演算的符号化模型检验

在上一节，给出了对具有一般形式的线性 μ-演算公式的符号化模型检验算法，该算法主要基于可编码的自动机转化过程，其所需的位变元数目为 $\mathcal{O}(|\varphi|^4)$。在本节，将给出一种针对特定形式线性 μ-演算公式的符号化模型检验算法。该种形式是线性 μ-演算公式的一种范式，任何 ω-正规性质均可写为该种形式的公式。对该种公式执行符号化模型检验的代价非常低，只需要引入 $\mathcal{O}(|\varphi|)$ 个额外的位变元即可。

6.3.1 线性 μ-演算范式及迟滞迁移系统

在开始本节讨论前，首先定义线性 μ-演算公式的一种范式。而后，将证明任何 ω-正规语言都能化为该种形式的线性 μ-演算公式。

定义 6.3.1 *称（良命名）线性 μ-演算公式 φ 是一个 ν-范式，如果对 φ 中的任意 μ-型约束变元 X 以及 ν-型约束变元 Y 都有：若 $Y\lhd_\varphi X$ 则 X 在 $\mathcal{D}_\varphi(Y)$ 中不出现。* □

下面说明任何 ω-正规性质都能写成 ν-范式。由于对字母表 2^{AP} 上的任何一个 ω-正规语言 R，都可以写为范式

$$r_1;(r'_1)^\omega \mid r_2;(r'_2)^\omega \mid \dots \mid r_{m-1};(r'_{m-1})^\omega \mid r_m;(r'_m)^\omega$$

的形式，其中 r_i、r'_i 都是 2^{AP} 上的有穷正规语言（简要说明见引理 5.21）。同时，在文 [103]中，介绍了由 Büchi、Wolper、Sistla 等给出如下的从 ω-正规语言到线性 μ-演算公式的转化算法：固定一个公式变元 V，则对于字母表 2^{AP} 上的正规语言 R，可以归纳构造一个线性 μ-演算公式 $\mathbf{TL}_V(R)$，其中

- 对每个 $a \subseteq AP$，$\mathbf{TL}_V(a) = \bigwedge_{p\in a} p \wedge \bigwedge_{p\notin a} \neg p \wedge V$。
- $\mathbf{TL}_V(r_1; r_2) = \mathbf{TL}_V(r_1)^V_{\bigcirc \mathbf{TL}_V(r_2)}$。
- $\mathbf{TL}_V(r_1 \mid r_2) = \mathbf{TL}_V(r_1) \vee \mathbf{TL}_V(r_2)$。
- $\mathbf{TL}_V(r^*) = \mu X.(\mathbf{TL}_V(r)^V_{V\vee\bigcirc X})$（其中，$X$ 是某个不在 $\mathbf{TL}_V(r)$ 中出现的变元）。
- $\mathbf{TL}_V(r^\omega) = \nu X.\mathbf{TL}_V(r)^V_{\bigcirc X}$（其中，$X$ 是某个不在 $\mathbf{TL}_V(r)$ 中出现的变元）。

以上，r、r_1、r_2 都是 2^{AP} 上的有穷正规语言。于是，对于每个正规表达式 $r_i;(r'_i)^\omega$ 而言，执行上述操作后得到的线性 μ-演算公式必为句子，且 V 不在 $\mathbf{TL}_V(r_i;(r'_i)^\omega)$ 中出现。

例 6.3.1 设 $AP = \{p_1, p_2\}$，$a_1 = \{p_1\}$，$a_2 = \{p_2\}$，$a_3 = \{p_1, p_2\}$。则对于 $R = (a_1; a_2)^*; (a_3)^\omega$，$\mathbf{TL}_V(R)$ 的计算过程如下:

1. $\mathbf{TL}_V(a_1) = p_1 \wedge \neg p_2 \wedge V$;
2. $\mathbf{TL}_V(a_2) = \neg p_1 \wedge p_2 \wedge V$;
3. $\mathbf{TL}_V(a_1; a_2) = (p_1 \wedge \neg p_2) \wedge \bigcirc(\neg p_1 \wedge p_2 \wedge V)$;
4. $\mathbf{TL}_V((a_1; a_2)^*) = \mu X.((p_1 \wedge \neg p_2) \wedge \bigcirc(\neg p_1 \wedge p_2 \wedge (V \vee \bigcirc X)))$;
5. $\mathbf{TL}_V(a_3) = p_1 \wedge p_2 \wedge V$;
6. $\mathbf{TL}_V(a_3^\omega) = \nu Y.(p_1 \wedge p_2 \wedge \bigcirc Y)$;
7. $\mathbf{TL}_V((a_1; a_2)^*; (a_3)^\omega) = \mu X.((p_1 \wedge \neg p_2) \wedge \bigcirc(\neg p_1 \wedge p_2 \wedge (\nu Y.(p_1 \wedge p_2 \wedge \bigcirc Y) \vee \bigcirc X)))$。

□

假设上述步骤所获得的公式是良命名的（这点在每次引入约束变元时选用新的变元名既可保证）。那么对每个 $r_i;(r'_i)^\omega$，由上述转化过程可以看出：

- $\mathbf{TL}_V(r_i)$ 和 $\mathbf{TL}_V(r'_i)$ 中不会出现 ν-型约束变元。
- 计算 $\mathbf{TL}_V((r'_i)^\omega)$ 时会引入唯一的 ν-型约束变元（不妨设其为 Y_i）。
- $\mathbf{TL}_V(r_i;(r'_i)^\omega)$ 中位于 Y_i 外层的约束变元（见定义 4.2.3）仅出现在 $\mathbf{TL}_V(r_i)$ 中。换言之，位于 Y_i 外层的约束变元不会出现在 $\nu Y_i.\mathbf{TL}_V(r'_i)^V_{\bigcirc Y_i}$ 中。于是，$\mathbf{TL}_V(R)$ 一定是 ν-范式。

由因为线性 μ-演算具有与 ω-正规语言等价的表达能力。于是，立即可以得到如下的定理。

定理 6.13 对于任一线性 μ-演算句子 φ，都能写成 ν-范式。

此外，容易根据公式结构归纳法证明：按 2.2.3 节中给出的从 LTL 公式到线性 μ-演算公式的转换过程得到的一定是 ν-范式。于是，立即有如下结论。

定理 6.14 对于任一 *LTL* 公式 φ，存在与之等价的长度为 $\mathcal{O}(|\varphi|)$ 的线性 μ-演算公

式，且其为 ν-范式。

本节引入如下记号：给定线性 μ-演算公式 φ，用 $\mathbf{NV}(\varphi)$ 表示 φ 中所有的 ν-型约束变元构成的集合。

设 φ 写为 ν-范式的线性 μ-演算公式，$Y \in \mathbf{NV}(\varphi)$。容易证明：在 φ 的博弈系统 $\mathcal{G}_\varphi$（见第 4 章中定义）中的任一条无穷踪迹 τ 中，位于 Y 外层的 μ-型约束变元不会出现在 Y 之后。因此，若 Y 在 τ 中无穷多次出现，则无穷多次出现在 τ 中的约束变元集合中的最外层者必为某个 ν-型约束变元。所以，参与者 0 是 $\mathcal{G}_\varphi$ 中的某个无穷对决的取胜者当且仅当某个 ν-型约束变元 Y 在该对决的每条无穷踪迹中都出现无穷多次。于是，如果能够从参与者 0 的取胜者策略中提取出一个满足 φ 的线性结构，就能通过博弈系统（的变形）来构造 φ 的语言模型。

为达到上述目的，这里对线性 μ-演算的博弈规则稍加修改。在原来关于 $\mathcal{G}_\varphi$ 的定义中，每个格局都是 $\mathbf{Sub}(\varphi)$ 的子集，现在将每个格局其扩展成 $\mathbf{Sub}(\varphi)\times\{+,-\}$ 的子集。其中格局中的第二个分量记录了踪迹目前是否遇到过某 ν-型约束变元。在任何一条踪迹 σ 中，对序偶 $(\psi,-)$ 而言，只有当 $\psi \in \mathbf{NV}(\varphi)$ 时才有可能将其第二个分量由 “-” 变为 “+”。此外，若在当前格局中所有序偶的第二个分量都是 “+”，将会使用某条规则将所有序偶的第二个分量变为 “−”。于是，如果某对决无穷多次到达 $\mathbf{Sub}(\varphi)\times\{+\}$ 的某个子集，也无穷多次到达 $\mathbf{Sub}(\varphi)\times\{-\}$ 的某个子集，那么其中任意一条无穷踪迹中必然无穷多次出现某 ν-型约束变元。从而可以从该对决中提取出满足 φ 的线性结构。

扩展的博弈规则如下所示。其中，序偶中出现的每个 $\pm$、$\pm_i$ 以及 $\pm_i'$ 均是 $\{+,-\}$ 中的元素。在规则 (or)、(and)、(fix)、(mu-rmv)、(nu-rmv) 中，要求 $\pm$ 的值在重写前后一致；在规则 (modal) 中，要求每个 $\pm_i'$ 的值在重写前后一致。在规则 (or) 中，要求 $i \in \{1,2\}$。在规则 (fix-tog) 中，要求 $Y \in \mathbf{NV}(\varphi)$。在规则 (modal) 中，要求每个 l_i 都是文字；能够使用 (modal) 的格局称为 **模态格局**。在使用 (sign-tog) 时，要求 $\{(\psi_1,+),\ldots,(\psi_m,+)\}$ 不能是模态格局。此外，为保证博弈系统的连续性，将 $\emptyset$ 也视为特殊的模态格局。

$$\frac{\Gamma,(\psi_1\vee\psi_2,\pm)}{\Gamma,(\psi_i,\pm)}\ \text{(or)} \qquad \frac{\Gamma,(\psi_1\wedge\psi_2,\pm)}{\Gamma,(\psi_1,\pm),(\psi_2,\pm)}\ \text{(and)}$$

$$\frac{\Gamma,(Y,\pm)}{\Gamma,(\mathbf{D}_\varphi(Y),+)}\ \text{(fix-tog)} \qquad \frac{\Gamma,(X,\pm)}{\Gamma,(\mathbf{D}_\varphi(X),\pm)}\ \text{(fix)}$$

$$\frac{\Gamma,(\mu X.\psi,\pm)}{\Gamma,(\psi,\pm)}\quad\text{(mu-rmv)}\qquad\frac{\Gamma,(\nu X.\psi,\pm)}{\Gamma,(\psi,\pm)}\quad\text{(nu-rmv)}$$

$$\frac{\{(\psi_1,+),\ldots,(\psi_m,+)\}}{\{(\psi_1,-),\ldots,(\psi_m,-)\}}\quad\text{(sign-tog)}$$

$$\frac{\{(l_1,\pm_1),\ldots,(l_k,\pm_k),(\bigcirc\psi_1,\pm'_1),\ldots,(\bigcirc\psi_m,\pm'_m)\}}{\{(\psi_1,\pm'_1),\ldots,(\psi_m,\pm'_m)\}}\quad\text{(modal)}$$

为使用尽可能低的空间开销实现上述算法，需要引入**迟滞迁移系统**的概念。

定义 6.3.2 (迟滞公平迁移系统) 一个迟滞(公平)迁移系统是一个序偶 $\mathcal{N}=\langle S,N,\Delta,I,\lambda,\mathcal{C}\rangle$。其中:

- S 是一个有穷**状态集**。
- $N\subseteq S$，是一组**模态状态**集合。
- $\Delta\subseteq S\times S$，是一个**迁移函数**。并且，若 $(s,s')\in\Delta$，则或者 $s\in N$，或者 $\lambda(s)=\lambda(s')$。
- $I\subseteq S$，是一组**初始状态**集合。
- $\lambda:S\to 2^{AP}$，是一个**命题标记函数**。
- $\mathcal{C}=\{C_1,\ldots,C_m\}$ 是一组**公平约束**集合。每个公平约束 C_i 均为 S 的子集。□

定义 6.3.3 设公平迁移系统 $\mathcal{N}=\langle S,N,\Delta,I,\lambda,\mathcal{C}\rangle$，则 $\mathcal{N}$ 中的一条**展开迹** σ 是一个无穷状态序列 $\sigma=s_0,s_1,\ldots$，其中 $s_0\in I$，$(s_i,s_{i+1})\in\Delta$。称 σ 是 $\mathcal{N}$ 中的一条**公平展开迹**，如果

- $\mathbf{Inf}(\sigma)\cap N\neq\emptyset$;
- 对每个 $1\leq i\leq m$，$\mathbf{Inf}(\sigma)\cap C_i\neq\emptyset$。

其中 $\mathbf{Inf}(\sigma)$ 表示 σ 中无穷多次出现的状态集合。

对于 $\mathcal{N}$ 中的公平展开迹 σ，设 $s_{k_0},s_{k_1},\ldots$ 是由 σ 中的所有模态状态 (按原来在 σ 中的顺序) 构成的子序列。则称线性结构 π 是 σ 的**派生线性结构**，如果对每个 $i\in\mathbb{N}$ 都有 $\pi(i)=\lambda(s_{k_i})$。

用 $\mathbf{L}(\mathcal{N})$ 表示 $\mathcal{N}$ 中所有的公平展开迹对应的派生线性结构集合。□

事实上，迟滞公平迁移系统是在通常的公平迁移系统上添加了“模态状态集” 以及对迁移函数增加约束后得到的。在公平迁移系统的一条公平展开迹 $s_0,s_1,\ldots$ 中，状态上的命题标注只有每次经过模态后才有可能发生改变。直观的讲，若 s_i 不是模

态状态，则 s_i 与 s_{i+1} 实际上对应同一个“模态时刻”。因此，将这样的迁移系统称为“迟滞的”。反之，一个普通的公平迁移系统也可以看作是特殊的迟滞公平迁移系统— 只需将其中的每个状态均看作是模态状态即可。

下面，定义迟滞公平迁移系统之间的合成。根据需要，这里只说明如何定义一个（非迟滞）的公平迁移系统与一个迟滞公平迁移系统之间的合成。

定义 6.3.4 (迟滞公平迁移系统的合成) 给定迟滞公平迁移系统 $\mathcal{N}=\langle S,N,\Delta,I,\lambda,\mathcal{C}\rangle$ 与（非迟滞）公平迁移系统 $\mathcal{M}=\langle S',\Delta',I',\lambda',\mathcal{C}'\rangle$ 则 $\mathcal{M}$ 与 $\mathcal{N}$ 的**合成**（记作 $\mathcal{M}\|\mathcal{N}$），是一个迟滞公平迁移系统 $\langle\hat{S},\hat{N},\hat{\Delta},\hat{\lambda},\hat{\mathcal{C}}\rangle$ 其中：

- $\hat{S}=\{(s',s)\mid s'\in S',s\in S,\lambda(s)=\lambda'(s')\}$。
- $\hat{N}=(S'\times N)\cap\hat{S}$。
- 对任意的$((s'_1,s_1),(s'_2,s_2))\in\hat{S}\times\hat{S}$，$(s'_1,s_1),(s'_2,s_2)\in\hat{\Delta}$ 当且仅当：
  - 若 $s_1\in S\backslash N$，则要求 $s'_2=s'_1$，$(s_1,s_2)\in\Delta$；
  - 若 $s_1\in N$，则要求 $(s'_1,s'_2)\in\Delta'$，$(s_1,s_2)\in\Delta$。
- 对于任意的 $(s',s)\in\hat{S}$，$\hat{\lambda}((s',s))=\lambda(s)$。
- 设 $\mathcal{C}=\{C_1,\ldots,C_m\}$，$\mathcal{C}'=\{C'_1,\ldots,C'_n\}$，则 $\hat{\mathcal{C}}=\{(C'_k\times S)\cap\hat{S}\mid 1\leq k\leq n\}\cup\{(S'\times C_k)\cap\hat{S}\mid 1\leq k\leq m\}$。 □

与普通的公平迁移系统合成类似，容易证明迟滞公平迁移系统的合成具有如下的性质。

引理 6.15 给定迟滞公平迁移系统 $\mathcal{N}$ 与（非迟滞）公平迁移系统 $\mathcal{M}$，则 $\mathbf{L}(\mathcal{M}\|\mathcal{N})=\mathbf{L}(\mathcal{M})\cap\mathbf{L}(\mathcal{N})$。

设 φ 是某写为 ν-范式的线性 μ-演算公式。于是，可以为 φ 构建一个迟滞公平迁移系统 $\mathcal{N}_\varphi=\langle S_\varphi,N_\varphi,\Delta_\varphi,I_\varphi,\lambda_\varphi,\mathcal{C}_\varphi\rangle$，其中：

- $S_\varphi=\{(a,\Gamma)\mid a\subseteq AP,\Gamma\subseteq\mathbf{Sub}(\varphi)\times\{+,-\}\}$，且满足：对于任意的 $p\in AP$，$\pm\in\{+,-\}$，若 $(p,\pm)\in\Gamma$ 则 $p\in a$；若 $(\neg p,\pm)\in\Gamma$ 则 $p\notin a$。
- $N_\varphi=\{(a,\Gamma)\mid\Gamma$ 是模态格局$\}$。
- $((a,\Gamma),(a',\Gamma'))\in\Delta_\varphi$ 当且仅当满足下列条件之一：
  - Γ' 可由 Γ 经规则 (modal) 得到。
  - 存在状态序列 $(a_1,\Gamma_1),\ldots,(a_m,\Gamma_m)$ 使得：$a=a_1=\ldots=a_m=a'$，$\Gamma=\Gamma_1$，$\Gamma'=\Gamma_m$，且每个 Γ_{k+1} 可由 Γ_k 经除 (modal) 之外的某条规则得到。并且，如果该过程中使用到了 (sign-tog) 规则，则 $m=2$。
- $I_\varphi=\{(a,\Gamma)\mid(\psi,-)\in\Gamma\}$。
- $\mathcal{C}_\varphi=\{C^+_\varphi,C^-_\varphi\}$，其中 $C^+_\varphi=\{(a,\Gamma)\in S_\varphi\mid\Gamma\subseteq\mathbf{Sub}(\varphi)\times\{+\}\}$；$C^-_\varphi=\{(a,\Gamma)\in$

$S_\varphi \mid \Gamma \subseteq \mathbf{Sub}(\varphi) \times \{-\}\}$。

特别的，在 $\mathcal{N}_\varphi$ 中，还存在着**基本展开迹** 的概念。

定义 6.3.5 (迟滞公平迁移系统中的基本展开迹) 称 $\sigma = (a_0, \Gamma_0), (a_1, \Gamma_1), \ldots$ 是 $\mathcal{N}_\varphi$ 中的一条**基本展开迹** 如果:

- σ 是 $\mathcal{N}_\varphi$ 中的一条展开迹;
- 对每个 $k \in \mathbb{N}$，或者 Γ_k 是模态格局，或者 $a_k = a_{k+1}$ 并且 Γ_{k+1} 可由 Γ_k 经除 (modal) 之外的某条规则得到。 □

下面的引理容易证明。

引理 6.16 若 σ 是 $\mathcal{N}_\varphi$ 中的一条展开迹，则必存在 $\mathcal{N}_\varphi$ 中的一条基本展开迹 σ'，使得: 若 σ 是公平展开迹则 σ' 也是公平展开迹，并且此时 σ 和 σ' 具有相同的派生线性结构。

下面将会证明 $\mathcal{N}_\varphi$ 的"语言性质"，即：对于任意的线性结构 π 而言，$\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$。事实上，只需证明 $\mathbf{L}(\mathcal{A}_\varphi) = \mathbf{L}(\mathcal{N}_\varphi)$ 即可。这里，$\mathcal{A}_\varphi$ 是由 φ 得到的 SAPW。

定理 6.17 对于任意的线性结构 π，若 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$ 则 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$。

证明. 设 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$，并且 $\langle T, \rho\rangle$ 是 $\mathcal{A}_\varphi$ 在 π 上的一个可接收运行。同前面的记法，令 ρ_1 和 ρ_2 分别是 ρ 的两个分量，即：若 $\rho(x) = (q_\psi, i)$，则 $\rho_1(x) = q_\psi$，$\rho_2(x) = i$。要证明 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$，只需根据 $\langle T, \rho\rangle$ 构造 $\mathcal{N}_\varphi$ 中的一条公平展开迹 $\sigma = s_0, s_1, \ldots$，其中 $s_k = (a_k, \Gamma_k)$ 而后证明 π 是 σ 的派生线性结构即可。

σ 的构造过程如下。该过程同时伴随一系列函数 $f_k : \Gamma_k \to T$ 的构造。

首先，令 $s_0 = (\pi(0), (\varphi, -))$，$f_0((\varphi, -)) = \epsilon$。于是，$s_0 \in I_\varphi$。由定义 4.3.2 有 $\rho_1(\epsilon) = q_\varphi$，$\rho_2(\epsilon) = 0$。接下来，对每个 $k \in \mathbb{N}$，归纳假设 f_k 满足：

1. 对每个 $(\psi, \pm) \in \Gamma_k$ 有 $\rho_1(f_k((\psi, \pm))) = q_\psi$，$a_k = \pi(\rho_2(f_k((\psi, \pm))))$;
2. 若 $(\psi, \pm) \in \Gamma_k$，$(\psi', \pm') \in \Gamma_k$，则 $\rho_2(f_k((\psi, \pm))) = \rho_2(f_k((\psi', \pm')))$。

接下来，按照如下方式由 s_k 获得 s_{k+1}。

- 若 Γ_k 是模态格局 $\{(l_1, \pm_1), \ldots, (l_m, \pm_m), (\bigcirc\psi_1, \pm'_1), \ldots, (\bigcirc\psi_n, \pm'_n)\}$，则按照 (modal) 规则得到 $\Gamma_{k+1} = \{(\psi_1, \pm'), \ldots, (\psi_n, \pm'_n)\}$。

  由归纳假设，对每个 $(\bigcirc\psi_t, \pm'_t)$，有 $\rho_1(f_k((\bigcirc\psi_t, \pm'_t))) = q_{\bigcirc\psi_t}$。由于 $\delta_\varphi(q_{\bigcirc\psi_t}) = \bigcirc q_{\psi_t}$，所以若 $f_k((\bigcirc\psi_t, \pm'_t)) = x_t$，则存在 x_t 的子节点 $x_t \cdot c \in T$ 使得 $\rho(x_t \cdot c) = (q_{\psi_t}, j+1)$。于是，令 $f_{k+1}((\psi_t, \pm'_t)) = x_t \cdot c$。

  接下来，令 $a_{k+1} = \pi(j+1)$。于是容易验证 f_{k+1} 仍然满足所有归纳不变式。
- 若 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{+\}$，且 Γ_k 不是模态格局，则令 Γ_{k+1} 是将 Γ_k 中每个 $(\psi, +)$

替换为 $(\psi,-)$ 所得的格局（从而 Γ_{k+1} 可从 Γ_k 经规则 (sign-tog) 得到）。再令 $a_{k+1}=a_k$。对每个 $(\psi,-)\in\Gamma_{k+1}$，令 $f_{k+1}((\psi,-))=f_k((\psi,+))$。于是，容易验证 f_{k+1} 仍然满足所有归纳不变式。

- （以下，设 Γ_k 不能应用 (modal) 规则或者 (sign-tog) 规则）
  若存在 $(\psi_1\vee\psi_2,\pm)\in\Gamma_k$，不妨设 $f_k((\psi_1\vee\psi_2,\pm))=x$，则由归纳假设，$\rho_1(x)=q_{\psi_1\vee\psi_2}$。由定义 4.3.2，$\delta_\varphi(q_{\psi_1\vee\psi_2})=q_{\psi_1}\vee q_{\psi_2}$。于是 x 必然存在子节点 $x\cdot c$ 使得 $\rho_1(x\cdot c)=q_{\psi_1}$ 或者 $\rho_1(x\cdot c)=q_{\psi_2}$，同时 $\rho_2(x)=\rho_2(x\cdot c)$。这里，不失一般性，设 $\rho_1(x\cdot c)=q_{\psi_1}$。
  令 $\Gamma_{k+1}=\Gamma_k\backslash\{(\psi_1\vee\psi_2,\pm)\}\cup\{(\psi_1,\pm)\}$，这样，$\Gamma_{k+1}$ 可由 Γ_k 经规则 (or) 得到。同时，令 $a_{k+1}=a_k$。此外，令 $f_{k+1}((\psi_1,\pm))=x\cdot c$；对于其余的每个 $(\phi,\pm')\in\Gamma_k$，令 $f_{k+1}((\phi,\pm'))=f_k((\phi,\pm'))$。
- 若存在 $(\psi_1\wedge\psi_2,\pm)\in\Gamma_k$，不妨设 $f_k((\psi_1\wedge\psi_2,\pm))=x$，则由归纳假设 $\rho_1(x)=q_{\psi_1\wedge\psi_2}$。由定义 4.3.2，$\delta_\varphi(q_{\psi_1\wedge\psi_2})=q_{\psi_1}\wedge q_{\psi_2}$。于是 x 必然存在子节点 $x\cdot c_1$ 和 $x\cdot c_2$ 使得 $\rho_1(x\cdot c_i)=q_{\psi_i}$ 以及 $\rho_2(x)=\rho_2(x\cdot c_i)$（这里 $i=\{1,2\}$）。
  令 $\Gamma_{k+1}=\Gamma_k\backslash\{(\psi_1\wedge\psi_2,\pm)\}\cup\{(\psi_1,\pm),(\psi_2,\pm)\}$，这样，$\Gamma_{k+1}$ 可由 Γ_k 经规则 (and) 得到。同时，令 $a_{k+1}=a_k$。此外，令 $f_{k+1}((\psi_1,\pm))=x\cdot c_1$，$f_{k+1}((\psi_2,\pm))=x\cdot c_2$；对于其余的每个 $(\phi,\pm')\in\Gamma_k$，令 $f_{k+1}((\phi,\pm'))=f_k((\phi,\pm'))$。
- 若存在 $(X,\pm)\in\Gamma_k$，不妨设 $f_k((X,\pm))=x$，则由归纳假设 $\rho_1(x)=q_X$。由定义 4.3.2，$\delta_\varphi(q_X)=q_{\mathbf{D}_\varphi(X)}$。于是 x 必然存在子节点 $x\cdot c$ 使得 $\rho_1(x\cdot c)=(q_{\mathbf{D}_\varphi(X)})$ 以及 $\rho_2(x)=\rho_2(x\cdot c)$。
  令 $\Gamma_{k+1}=\Gamma_k\backslash\{(X,\pm)\}\cup\{(\mathbf{D}_\varphi(X),\pm')\}$，这时 Γ_{k+1} 可由 Γ_k 经规则 (fix) 或者规则 (fix-tog) 得到。同时，令 $a_{k+1}=a_k$。接下来，令 $f_{k+1}((X,\pm))=x\cdot c$；对于其余的每个 $(\phi,\pm'')\in\Gamma_k$，令 $f_{k+1}((\phi,\pm'))=f_k((\phi,\pm''))$。
- 若存在 $(\mu X.\psi,\pm)\in\Gamma_k$（resp. $(\nu X.\psi,\pm)\in\Gamma_k$），不妨设 $f_k((\mu X.\psi,\pm))=x$（resp. $f_k((\nu X.\psi,\pm))=x$），则由归纳假设 $\rho_1(x)=q_\psi$。于是 x 必然存在子节点 $x\cdot c$ 使得 $\rho_1(x\cdot c)=(q_\psi)$ 以及 $\rho_2(x)=\rho_2(x\cdot c)$。
  令 $\Gamma_{k+1}=\Gamma_k\backslash\{(\mu X.\psi,\pm)\}\cup\{(\psi,\pm))\}$（resp. $\Gamma_{k+1}=\Gamma_k\backslash\{(\nu X.\psi,\pm)\}\cup\{(\psi,\pm))\}$），这样，$\Gamma_{k+1}$ 可由 Γ_k 经规则 (mu-rmv)（resp. (nu-rmv)）得到。同时，令 $a_{k+1}=a_k$。此外，令 $f_{k+1}((\psi,\pm))=x\cdot c$；对于其余的每个 $(\phi,\pm')\in\Gamma_k$，令 $f_{k+1}((\phi,\pm'))=f_k((\phi,\pm'))$。

现在证明 σ 确为 $\mathcal{N}_\varphi$ 中的一条公平展开迹。首先验证每个 s_k 均为 $\mathcal{N}_\varphi$ 中的合法状态，即：对于每个 $p\in AP$，若 $(p,\pm)\in\Gamma_k$ 则 $p\in a_k$，若 $(\neg p,\pm)\in\Gamma_k$ 则

$p \notin a_k$。

利用类似于第 4 章中技术（如引理 4.5 中的定义过程），通过对每个 $\psi \in \mathbf{Sub}(\varphi)$ 赋一个自然数 $\mathbf{Rk}(\psi)$ 的方式可以证明 σ 中有无穷多个模态状态。设 s_h 是 s_k 后（包括 s_k）的第一个模态状态。对于每个 $p \in AP$，若 $(p, \pm) \in \Gamma_k$（resp. $(\neg p, \pm) \in \Gamma_k$）则 $(p, \pm) \in \Gamma_h$（resp. $(\neg p, \pm) \in \Gamma_h$）。不妨设 $f_h((p, \pm)) = x$（resp. $f_h(\neg, \pm) = x$），则 $\rho_1(x) = q_p$（resp. $\rho_1(x) = q_{\neg p}$），$\pi(\rho_2(x)) = a_h$。由定义 4.3.2，$\delta(q_p) = p$（resp. $\delta(q_{\neg p}) = \neg p$），于是要求 $\pi, \rho_2(x) \models p$（resp. $\pi, \rho_2(x) \models \neg p$）。因此，$p \in a_k$（resp. $p \notin a_k$）。

此外，对于任意的 s_k 和 s_{k+1}，或者 Γ_{k+1} 可由 Γ_k 经 (modal) 规则得到；或者 $a_k = a_{k+1}$ 且 Γ_{k+1} 可由 Γ_k 经除 (modal) 外的某条规则得到。因此 $(s_k, s_{k+1}) \in \Delta_\varphi$，于是 σ 是 $\mathcal{N}_\varphi$ 中的一条展开迹，并且是基本展开迹。

现在证明有无穷多个 k 使得 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{+\}$。用反证法，假设存在某个 $t \in \mathbb{N}$，使得对于任意的 $k \geq t$，都有 $\Gamma_k \not\subseteq \mathbf{Sub}(\varphi) \times \{+\}$。于是，对于任意的 $k \geq t$，由 Γ_k 获得 Γ_{k+1} 的规则不可能是 (sign-tog)。注意到 (sign-tog) 是唯一能够使序偶中第二个分量由 “+” 变为 “−” 的规则，而在不使用该规则时，任何踪迹中出现在 $(\psi, +)$ 后面序偶的第二个元素必然也是 “+”。这样，必然存在一条无穷踪迹 τ 使得 τ 中第二个分量为 “+” 的序偶只出现有穷次。否则，假设每个包含 $\mathbf{Sub}(\varphi) \times \{-\}$ 中元素的极大踪迹长度都有穷，则这些踪迹必然有前继在 Γ_t 中或者经过 Γ_t。于是，由 König 引理，存在一个 $d \in \mathbb{N}$，使得这些踪迹在经过 Γ_t 后延伸的长度不超过 d。这样 $\Gamma_{t+d+1} \subseteq \mathbf{Sub}(\varphi) \times \{+\}$。不妨设 τ 开始于 Γ_h 中的某个序偶，则由构造知 $f_h(\tau(0)), f_{h+1}(\tau(1)), \ldots$ 必然对应与 T 中的一条无穷路径 $\sigma' = x_0, x_1, \ldots$。并且对每个 $Y \in \mathbf{NV}(\varphi)$ 而言，σ' 中只有有穷多个节点 x_k 使得 $\rho_1(x_k) = q_Y$（否则，有无穷多个 k 使得 $\tau(k) = (Y, \pm)$，在下一次使用 (modal) 规则前必然会使用 (fix-tog)，从而使该踪迹中序偶的第二个元素变为 “+”）。但是，这样 $\mathbf{Inf}(\rho(\sigma'))$ 就违反了 Ω_φ。这与 $\langle T, \rho \rangle$ 是 $\mathcal{A}_\varphi$ 在 π 的可接收运行相矛盾。因此，假设不成立，从而公平限制 $\mathbf{Sub}(\varphi) \times \{+\}$ 被满足。

再证明有无穷多个 k 使得 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{-\}$。如果在满足 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{+\}$ 的格局中有无穷多个是非模态格局，则由构造知将会无穷多次应用规则 (sign-tog)，这时必然满足条件。否则，必然有某个 $\Gamma_h \subseteq \mathbf{Sub}(\varphi) \times \{+\}$，使得对于任意的 $k \geq h$，若 $\Gamma_h \subseteq \mathbf{Sub}(\varphi) \times \{+\}$ 则 Γ_k 是模态节点。注意到对 Γ_k 应用 (modal) 规则后 Γ_{k+1} 仍是 $\mathbf{Sub}(\varphi) \times \{+\}$ 的子集。因此 Γ_h 中的每个元素必定形如 $(\bigcirc^t l, \pm)$，其中 $l \in AP \cup \overline{AP} \cup \{true, false\}$，$\bigcirc^t$ 是 t 个连续 $\bigcirc$ 算子的缩写。于是，自 Γ_h 有穷步

后，每个格局都变为 $\emptyset$，而 $\emptyset \subseteq \mathbf{Sub}(\varphi) \times \{-\}$。

最后说明 π 是 σ 在 $\mathcal{N}_\varphi$ 中的派生线性结构。设 σ 中的模态状态依次为 $s_{l_0}, s_{l_1}, \ldots$，则由构造归纳可得：

1. 对于任意的 $0 \leq k \leq l_0$，有 $a_k = a_0$，并且对于任意的 $(\psi, \pm) \in \Gamma_k$ 有 $f_k((\psi, \pm)) = 0$。于是 $a_{l_0} = \pi(0)$。
2. 假设 $\pi(t) = a_{l_t}$ 成立，且对于任意的 $l_{t-1} + 1 \leq k \leq l_t$ 以及 $(\psi, \pm) \in \Gamma_k$ 有 $f_k((\psi, \pm)) = t$，则由构造知 $a_{l_t+1} = \pi(t+1)$。同样可得：对于任意的 $l_t + 1 \leq k \leq l_{t+1}$ 以及 $(\psi, \pm) \in \Gamma_k$，有 $f_k((\psi, \pm)) = t + 1$。于是，归纳不变式被保持。

因此，π 是 σ 在 $\mathcal{N}_\varphi$ 中的派生线性结构，所以有 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$。 □

定理 6.18 *对于任意的线性结构 π 有：若 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$ 则 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$。*

证明. 设 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$，且 π 是 $\mathcal{N}_\varphi$ 中公平展开迹 $\sigma = s_0, s_1, s_2, \ldots$。由引理 6.16，不妨设 σ 是 $\mathcal{N}_\varphi$ 中的基本公平展开迹。同时，对每个 $i \in \mathbb{N}$，设 $s_i = (a_i, \Gamma_i)$。

以下，对每个 $k \in \mathbb{N}$，令 $\eta(k)$ 为 σ 中出现在 s_k 之前（不包括 s_k）模态状态的数目。则由定义有 $a_k = \pi(\eta(k))$ 成立。

要证明 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$，只需构造 $\mathcal{A}_\varphi$ 在 π 上的一个可接收运行 $\langle T, \rho \rangle$ 即可。该运行的构造过程伴随一些列函数 $g_k : \Gamma_k \to T$ 的构造。

首先，添加根节点 ϵ，令 $\rho(\epsilon) = (\varphi, 0)$ $g_0((\varphi, -)) = \epsilon$。

其次，对每个 $k \in \mathbb{N}$，归纳假设"对任意的 $(\psi, \pm) \in \Gamma_k$，$\rho(g_k((\psi, \pm))) = (q_\psi, \eta(k))$" 成立。于是，可根据 Γ_k 获得 Γ_{k+1} 所使用的规则按如下方式为 T 添加新的节点。

- 当使用的规则是 (modal) 时，则 Γ_k 必是模态格局。不妨设 $\Gamma_k = \{(l_1, \pm_1), \ldots, (l_n, \pm_n), (\bigcirc\psi_1, \pm'_1), \ldots, (\bigcirc\psi_m, \pm'_m)\}$，那么，$\Gamma_{k+1} = \{(\psi_1, \pm'_1), \ldots, (\psi_m, \pm'_m)\}$。

  对于每个 $(l_t, \pm_t)$，设 $g_k((l_t, \pm_t)) = x_t$，则由归纳假设有 $\rho(x_t) = (q_{l_t}, \eta(k))$。由定义 4.3.2 有 $\delta_\varphi(q_{l_t}) = l_t$。而由 $a_k = \pi(\eta(k))$ 以及 (a_k, Γ_k) 是 S_φ 中的状态知 $\pi(\eta(k)) \models l_t$。

  对每个 $(\bigcirc\psi_t, \pm'_t)$，不妨设 $g_k(\bigcirc\psi_t, \pm'_t) = x_t$，则由归纳假设 $\rho(x_t) = (q_{\bigcirc\psi_t}, \eta(k))$。于是，为每个 x_t 添加一个子节点 $x_t \cdot 0$，并令 $\rho(x_t \cdot 0) = (q_{\psi_t}, \eta(k+1))$。注意到 Γ_k 是模态格局，从而 s_k 是模态状态，于是 $\eta(k+1) = \eta(k) + 1$。再令 $g_{k+1}((\psi_t, \pm'_t)) = x_t \cdot 0$。这样，归纳不变式对 Γ_{k+1} 中的每个元素均成立。
- 当使用的规则是 (sign-tog) 时，不妨设 $\Gamma_k = \{(\psi_1, +), \ldots, (\psi_m, +)\}$，则 $\Gamma_{k+1} = \{(\psi_1, -), \ldots, (\psi_m, -)\}$。这时，不向 T 中添加任何新的节点。对每个 $(\psi_t, -) \in \Gamma_{k+1}$，令 $g_{k+1}((\psi_t, -)) = g_k((\psi_t, +))$ 即可。

- 当使用的规则是 (and) 时，不妨设 Γ_k 和 Γ_{k+1} 分别为 $\Gamma' \cup \{(\psi_1 \wedge \psi_2, \pm)\}$ 和 $\Gamma' \cup \{(\psi_1, \pm), (\psi_2, \pm)\}$。设 $g_k((\psi_1 \wedge \psi_2, \pm)) = x$，则由归纳假设，$\rho(x) = (q_{\psi_1 \wedge \psi_2}, \eta(k))$。这时，为 x 添加两个子节点 $x \cdot 0$ 和 $x \cdot 1$，并令 $\rho(x \cdot 0) = (q_{\psi_1}, \eta(k))$，$\rho(x \cdot 1) = (q_{\psi_2}, \eta(k))$。令 $g_{k+1}((\psi_1, \pm)) = x \cdot 0$，$g_{k+1}((\psi_2, \pm)) = x \cdot 1$；对其余每个 $(\phi, \pm') \in \Gamma'$，令 $g_{k+1}((\phi, \pm')) = g_k((\phi, \pm'))$。
- 当使用的规则是 (or) 时，不妨设 Γ_k 和 Γ_{k+1} 分别为 $\Gamma' \cup \{(\psi_1 \vee \psi_2, \pm)\}$ 和 $\Gamma' \cup \{(\psi_l, \pm)\}$（这里，$l \in \{1, 2\}$）。设 $g_k((\psi_1 \vee \psi_2, \pm)) = x$，则由归纳假设，$\rho(x) = (q_{\psi_1 \vee \psi_2}, \eta(k))$。这时，为 x 添加子节点 $x \cdot 0$，并令 $\rho(x \cdot 0) = (q_{\psi_l}, \eta(k))$。令 $g_{k+1}((\psi_l, \pm)) = x \cdot 0$；对其余每个 $(\phi, \pm') \in \Gamma'$，令 $g_{k+1}((\phi, \pm')) = g_k((\phi, \pm'))$。
- 当使用的规则是 (fix) 或者 (fix-tog) 时，不妨设 Γ_k 和 Γ_{k+1} 分别为 $\Gamma' \cup \{(X, \pm)\}$ 和 $\Gamma' \cup \{(\mathbf{D}_\varphi(X), \pm')\}$。设 $g_k((X, \pm)) = x$，则由归纳假设，$\rho(x) = (q_X, \eta(k))$。这时，为 x 添加子节点 $x \cdot 0$，并令 $\rho(x \cdot 0) = (q_{\mathbf{D}_\varphi(X)}, \eta(k))$。令 $g_{k+1}((q_{\mathbf{D}_\varphi(X)}, \pm)) = x \cdot 0$；对其余每个 $(\phi, \pm') \in \Gamma'$，令 $g_{k+1}((\phi, \pm')) = g_k((\phi, \pm'))$。
- 当使用的规则是 (mu-rmv)（resp. (nu-rmv)）时，不妨设 Γ_k 和 Γ_{k+1} 分别为 $\Gamma' \cup \{(\mu X.\psi, \pm)\}$（resp. $\Gamma' \cup \{(\nu X.\psi, \pm)\}$）和 $\Gamma' \cup \{(\psi, \pm)\}$。设 $g_k((\mu X.\psi, \pm)) = x$（resp. $g_k((\nu X.\psi, \pm)) = x$），则由归纳假设，$\rho(x) = (q_{\mu X.\psi}, \eta(k))$（resp. $\rho(x) = (q_{\nu X.\psi}, \eta(k))$）。这时，为 x 添加子节点 $x \cdot 0$，并令 $\rho(x \cdot 0) = (q_\psi, \eta(k))$。令 $g_{k+1}((\psi, \pm)) = x \cdot 0$；对其余每个 $(\phi, \pm') \in \Gamma'$，令 $g_{k+1}((\phi, \pm')) = g_k((\phi, \pm'))$。

由定义，可以验证 $\langle T, \rho \rangle$ 是 $\mathcal{A}_\varphi$ 在 π 上的一个运行。接下来证明 $\langle T, \rho \rangle$ 是一个可接收运行。

对于 T 中任意一条无穷路径 $x_0, x_1, \ldots$，必然存在某个 $h \geq 0$ 以及踪迹 $(\psi_0, \pm_0)$, $(\psi_1, \pm_1), \ldots$ 使得 $(\psi_k, \pm_k) \in \Gamma_{k+h}$，并且 $g_{h+k}((\psi_k, \pm_k)) = x_k$。由构造知，$\rho_1(x_k) = q_{\psi_k}$（这里，$\rho_1$ 是 ρ 在第一个分量上的投影）。

由于 σ 是 $\mathcal{N}_\varphi$ 中的一条基本公平展开迹，因此有无穷多个 k 使得 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{+\}$，同时又有无穷多个 k 使得 $\Gamma_k \subseteq \mathbf{Sub}(\varphi) \times \{-\}$。注意到 $(\psi_0, \pm_0)$, $(\psi_1, \pm_1), \ldots$ 是 $\Gamma_0, \Gamma_1, \ldots$ 中的踪迹，因此必然由无穷多个 k 使得 $\pm_k = -$ 并且 $\pm_{k+1} = +$。而能够使第二个分量由 “–” 变为 “+” 的规则只有 (fix-tog)，这说明有无穷多个 k 使得 $\rho_1(x_k) \in \mathbf{NV}(\varphi)$。

由于 φ 是 ν-范式，所以必然存在某个 $Y \in \mathbf{NV}(\varphi)$ 使得有无穷多个 $k \in \mathbb{N}$ 使得 $\rho_1(x_k) = q_Y$，并且对于任意的 μ-型约束变元 X，若 $Y \lhd_\varphi X$ 则只有有穷多个 t 使得 $\rho_1(x_t) = q_X$。于是，$\max\{\Omega_\varphi(q_X) \mid X \text{ 是 } \varphi \text{ 中的约束变元，且存在无穷多个 } k \text{ 使得 } \rho_1(x_k) = q_X\} = \Omega_\varphi(q_Y) \in \mathbb{E}$。

因此，$\langle T, \rho\rangle$ 是 $\mathcal{A}_\varphi$ 在 π 上的一个可接收运行。于是，有 $\pi \in \mathbf{L}(\mathcal{A}_\varphi)$。 □

由定理 4.27、定理 6.17 以及定理 6.18，立即可以得到如下的推论。

推论 6.19 设 $\mathcal{N}_\varphi$ 是 φ 对应的迟滞公平迁移系统，则对任意的线性结构 π 有: $\pi \models \varphi$ 当且仅当 $\pi \in \mathbf{L}(\mathcal{N}_\varphi)$。

一般情况下，会采用 ν-范式的公式的取非做规约。也就是说，这类规约一般描述“安全性质”。

定理 6.20 设 φ 是具有 ν-范式形式的线性 μ-演算句子，则 $\mathcal{M} \models \neg\varphi$ 当且仅当 $\mathcal{M}\|\mathcal{N}_\varphi \not\models \mathsf{EG}\ true$。

6.3.2 检验算法的符号化实现

在本节，将给出 6.3.1 节中针对 ν-范式线性 μ-演算公式的符号化实现。

这里，首先介绍针对迟滞公平迁移系统的 BDD 编码方式。事实上，对于迟滞公平迁移系统 $\mathcal{N} = \langle S, N, \Delta, I, \lambda, \mathcal{C}\rangle$，只需在 2.3.3.2 节所定义编码的基础上增加如下部分即可：

模态状态集 ：为初始状态集 I 定义布尔函数 $\Phi_N(\vec{Z})$，它满足：对任意的 $s \in S$，若 $f(s) = (v_1, \ldots, v_n)$，则 $s \in I$ 当且仅当将 z_i 的值赋为 v_i 后 Φ_N 的值为 1。

迁移函数 ：新的迁移函数在原来迁移函数编码的基础上增加一个合取项

$$\Phi_N \vee \bigwedge_{p\in AP} (\Phi^p_\lambda \leftrightarrow \Phi'^p_\lambda) \tag{6.19}$$

后得到的。其中 Φ'^p_λ 是将 Φ^p_λ 中的每个 z_i 替换为 z'_i 获得的布尔公式。

给定迟滞公平迁移系统 $\mathcal{N} = \langle S, N, \Delta, I, \lambda, \mathcal{C}\rangle$ 以及迁移系统 $\mathcal{M} = \langle S', \Delta', I', \lambda', \mathcal{C}'\rangle$，编码 $\mathcal{M}\|\mathcal{N}$ 时，其模态状态 $S' \times N$ 的编码仍为 Φ_N，其迁移函数的编码为

$$(\Phi_N \rightarrow \Phi_\Delta \wedge \Phi_{\Delta'})) \wedge (\neg\Phi_N \rightarrow \bigwedge_{z_i\in\vec{Z}} (z_i \leftrightarrow z'_i) \wedge \Phi_\Delta) \tag{6.20}$$

其中，$\vec{Z}$ 是编码 $\mathcal{M}$ 的位变元集合。其余部分的编码同普通公平迁移系统间的合成。

在定理 6.20 中，给出了（写为 ν-范式的）线性 μ-演算公式的模型检验问题到 CTL 模型检验问题的转化方法。现在说明对每个 $\mathcal{N}_\varphi$ 的编码过程（注意，这里 φ 已经写为否定范式）。

位变元集合 ：对每个 $p \in AP$，引入一个位变元 z_p；对每个 $\psi \in \mathbf{Sub}(\varphi)$，引入两个位变元 $y_{(\psi,+)}$ 和 $y_{(\psi,-)}$。

状态约束 ： $\mathcal{N}_\varphi$ 状态约束的布尔编码为 $\bigwedge_{p\in AP,\pm\in\{+,-\}}((y_{p,\pm} \rightarrow z_p)) \wedge (y_{(\neg p,\pm)} \rightarrow \neg z_p)$。

模态状态 ：令 $\mathbf{Bas}(\varphi) = \{\psi \in \mathbf{Sub}(\varphi) \mid \psi \in AP \cup \overline{AP} \text{ 或 } \psi = \bigcirc\psi'\}$。则 Φ_{N_φ} 的编码为 $\bigwedge_{\psi \notin \mathbf{Bas}(\varphi)}(\neg y_{(\psi,+)} \wedge \neg y_{(\psi,-)})$。

迁移函数 ：首先，令 Ψ_{Δ_φ} 为如下各式的合取。

$$\bigwedge_{\psi_1 \wedge \psi_2 \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(\psi_1 \wedge \psi_2,\pm)} \leftrightarrow (y'_{(\psi_1 \wedge \psi_2,\pm)} \vee (y'_{(\psi_1,\pm)} \wedge y'_{(\psi_2,\pm)}))) \tag{6.21}$$

$$\bigwedge_{\psi_1 \vee \psi_2 \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(\psi_1 \vee \psi_2,\pm)} \leftrightarrow (y'_{(\psi_1 \vee \psi_2,\pm)} \vee (y'_{(\psi_1,\pm)} \vee y'_{(\psi_2,\pm)}))) \tag{6.22}$$

$$\bigwedge_{X \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(X,\pm)} \leftrightarrow (y'_{(X,\pm)} \vee y'_{(\mathbf{D}_\varphi(X),\pm)})) \tag{6.23}$$

$$\bigwedge_{Y \in \mathbf{NV}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(Y,\pm)} \leftrightarrow (y'_{(Y,\pm)} \vee y'_{(\mathbf{D}_\varphi(X),+)})) \tag{6.24}$$

$$\bigwedge_{\mu.X\psi \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(\mu.X\psi,\pm)} \leftrightarrow (y'_{(\mu X.\psi,\pm)} \vee y'_{(\psi,\pm)})) \tag{6.25}$$

$$\bigwedge_{\nu.X\psi \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(\nu.X\psi,\pm)} \leftrightarrow (y'_{(\nu X.\psi,\pm)} \vee y'_{(\psi,\pm)})) \tag{6.26}$$

则迁移函数的编码 Φ_{Δ_φ} 就是以下三项的合取。

$$(\neg\Phi_{N_\varphi} \wedge \bigvee_{\psi \in \mathbf{Sub}(\varphi)} y_{(\psi,-)}) \rightarrow \Psi_{\Delta_\varphi} \tag{6.27}$$

$$(\neg\Phi_{N_\varphi} \wedge \bigwedge_{\psi \in \mathbf{Sub}(\varphi)} \neg y_{(\psi,-)}) \rightarrow \bigwedge_{\psi \in \mathbf{Sub}(\varphi)} (y_{(\psi,+)} \leftrightarrow y'_{(\psi,-)}) \tag{6.28}$$

$$\Phi_{N_\varphi} \rightarrow \bigwedge_{\bigcirc\psi \in \mathbf{Sub}(\varphi)} \bigwedge_{\pm \in \{+,-\}} (y_{(\bigcirc\psi,\pm)} \leftrightarrow y'_{(\psi,\pm)}) \tag{6.29}$$

初始状态集合 ：初始状态的布尔编码 $\Phi_{I_\varphi} = y_{(\varphi,-)}$。

标记函数 ：对于每个 $p \in AP$，$\Phi^p_\lambda = z_p$。

公平限制 ：公平性限制 $\Phi_{\mathcal{C}_\varphi} = \{\Phi_{C^+_\varphi}, \Phi_{C^-_\varphi}\}$。其中，$\Phi_{C^+_\varphi} = \bigwedge_{\psi \in \mathbf{Sub}(\varphi)} \neg y_{(\phi,-)}$；$\Phi_{C^-_\varphi} = \bigwedge_{\psi \in \mathbf{Sub}(\varphi)} \neg y_{(\phi,+)}$。

容易看出，编码 φ 所额外引入的位变元通常情况下为 $2 \times |\varphi|$ 个（因为，形如 z_p 的位变元在编码 $\mathcal{M}$ 时已经被引入）。

在实际的实现中，现有的模型检验工具并没有在底层实现对迟滞公平迁移系统的 CTL 符号化模型检验算法的支持。但是，注意到对于迟滞公平迁移系统 $\mathcal{N} = \langle S, N, \Delta, I, \lambda, \mathcal{C}\rangle$ 而言，$\mathcal{N} \models \mathsf{EG}\ true$ 当且仅当 $\mathcal{N}$ 中有一条公平展开迹。而这当且仅当非迟滞公平迁移系统 $\mathcal{N}' = \langle S, \Delta, I, \lambda, \mathcal{C} \cup \{N\}\rangle$ 中有一条公平展开迹。换言之，当且仅当 $\mathcal{N}' \models \mathsf{EG}\ true$。

因此，在定理 6.20 的实际实现中，当获得 $\mathcal{M}\|\mathcal{N}_{\varphi}$ 的 BDD 编码后，忽略模态状态编码而增加公平性约束 $\Phi_{N_{\varphi}}$，直接将其视为非迟滞的公平迁移系统，而后在其上执行标准的 CTL 符号化模型检验算法验证 $\mathsf{EG}\ true$。这样，就不需要对算法的底层做任何修改。

6.4 本章小结

在本节，给出了线性 μ-演算的符号化模型检验算法，主要研究了两方面的内容：

- 针对一般形式的线性 μ-演算公式的符号化模型检验算法。
- 针对具有某种特定形式— ν-范式的线性 μ-演算公式的符号化模型检验算法。

在第一个问题中，是通过一系列的自动机转化，构造公式的语言模型实现的。该转化要求每一步都能够实现符号化编码。在这种方法中，编码规约性质所需布尔变元的数目与是规约长度成多项式关系。其中主要的复杂度来源在于非确定 Büchi 自动机的求补。

虽然在 [115]、[119] 等文献中描述了更加紧致的 Büchi 自动机求补算法，但是这些算法的实现中，目标自动机的状态中都包含某个 Büchi 的 rank 函数。采用布尔公式对这种 rank 函数进行编码是困难的。此外，在文 [91] 中，Vardi 给出了从线性 μ-演算到 NBW 的直接转化算法。甚至，该种逻辑中带有过去时序算子。然而，采用该种算法得到的编码并不会降低布尔变元的数目。

因此，寻找更加高效的线性 μ-演算公式的符号化模型算法仍是将来需要研究的问题之一。

相对而言，针对写为 ν-范式的线性 μ-演算公式的模型检验的复杂性要低的多。但是它适用的场合有限—它往往用来检查“模型中是否有路径违反指定性质”。因而这种公式只有在作为规约的取非出现时算法才能有效采用。

因此，研究比 ν-范式限制条件宽松的线性 μ-演算公式的高效符号化模型检验算法也是需要继续研究的一个问题。

第七章　扩展的符号化模型检验工具：ENuSMV

7.1　引言

本文的第 5 章和第 6 章中分别研究了两类等价于 ω-正规语言的时序逻辑（交错 ETL 和线性 μ-演算）的符号化模型检验算法。这些算法，在理论上为 ω-正规性质的验证提供了一个框架。同时，为这些逻辑符号化模型检验的工具实现提供了支持。

在模型检验领域，工具的地位丝毫不亚于理论本身。一方面，工具是验证理论有效性和可行性的重要依据；另一方面，工具本身在实践中有着重要的实用价值。

根据工具建模语言的抽象层次不同，可以将它们分为“**模型级检验工具**” 和“**代码级检验工具**” 两类。前者如 SPIN[38]、SMV[40]，它们的输入是诸如 PROMELA、SMV script 等对算法、协议的描述语言。后者如 SLAM[123]、Java-PATHFINDER[124]、BLAST[125]、MAGIC[126]、CBMC[127] 等，它们的输入是类 C 的高级命令式语言源程序。

相比而言，模型级检验工具关注算法、协议的设计正确性，而代码级检验工具可以检测实现的正确性。此外，前者比较适合于硬件电路的时序正确性检查（比如，VIS 以及 Cadence SMV 中直接提供了从硬件描述语言 Verilog 到 SMV Script 的转化工具），而后者适合于软件程序的正确性检查。由于本文主要关注 ω-正规性质的符号化模型检验方法，而扩展部分在表达能力上的优势在“强时序” 的模型中（即对 “next” 算子有严格定义比如同步时序电路）中更能体现。因此，这里这要关注模型级检验工具。

最早出现的符号化模型检验工具是由 McMillian 等人设计的 SMV（Symbolic Model Verifier）。它支持 CTL 的符号化检验（检验算法具体过程见本文 2.3.3.2 节）。在 SMV 中，建模语言（SMV script）以“模块” 为单位进行组织，而模块的核心是变元的赋值语句。这样，模型在组织上就非常贴近于 VHDL、Verilog 等硬件设计/描述语言。

后来，Grumberg、Clarke、Hamaguchi 等人给出了基于 tableau 的 LTL 模型检验算法[16, 110]，从而将 LTL 的模型检验问题转化为 CTL 的模型检验问题。基于该算法，CMU/ICT-irst 设计实现了同时支持 CTL 和 LTL 的符号化模型检验工具 NUSMV（见文 [111]），并在建模语言上实现了一定的语法扩展。随后，该工具成为

一个开源项目，增加了许多新的语言特性以及对其他时序逻辑符号化验证的支持。目前，NuSMV 的最新版本为 Ver 2.4.3，它支持 CTL、LTL、RTCTL 以及部分 PSL（能够转化为 CTL 或者 LTL 的部分）的符号化模型检验。

在 NuSMV 的基础上，我们对其进行了进一步的扩展。目前已释放的最新版本增加了对 ETL_f 和 APSL 的符号化模型检验的支持（该工具遵循 GPL2 标准，可在 http://enusmv.sourceforge.net 处获得）。其中，ETL_f 可以看作是 ATL_f 的特例，将 5.3 节中算法例化即可直接获得。之所以优先提供对这两种语言的符号化模型检验算法的支持，是出于如下考虑：

1. ETL_f 中使用非确定有穷自动机作为时序连接子，而这种自动机是人们最常接触的计算模型之一。因而，使用该种逻辑书写的规约易于理解。此外，多数的 ω-正规性质可以直接被 ETL_f 公式简洁的描述。而且，LTL 公式可以直接编码为该种逻辑公式。
2. APSL 是 PSL 的变种（见本文 5.6 节）。而 PSL 目前已经成为规约语言的工业标准。此外，ETL_f 同 APSL 的线性部分— AFL 有很好的互补性：一方面，AFL 中采用字母表 2^{AP} 上的有穷自动机作为辅助构造子，这种构造子可以看作是 ETL_f 公式 $\mathcal{A}(\varphi_1,\ldots,\varphi_n)$，其中每个 φ_i 只能是布尔公式，而 ETL 中却允许自动机公式的嵌套；另一方面，AFL 中提供了丰富的时序连接子，如 trigger、abort 等，这些连接子无法直接用自动机编码。
3. 在 5、6 章中研究的 5 种等价于 ω-正规语言的时序逻辑中，这两种语言是最为直观的。就目前已经开发出的符号化模型检验算法而言，实现对这两种语言的符号化模型检验算法的代价也是最低的（编码 ETL_f 公式 tableau 的复杂度与编码 ATL_f 公式 tableau 的复杂度完全相同）。

本章组织结构如下。

1. 7.2 节介绍 ENuSMV 的语法的成分。其中 7.2.1 节介绍NuSMV 的部分原有语法成分，包括建模语言和规约语言；7.2.2 节介绍 ENuSMV 相对于NuSMV的扩展语法成分；7.2.3 节将介绍如何使用 ENuSMV 进行性质验证。
2. 7.3 节介绍使用 ENuSMV 进行模型检验的若干实验结果。其中 7.3.1 节将介绍针对 ETL_f 性质模型检验的实验结果；7.3.2 节将介绍针对 AFL 性质模型检验的实验结果。

7.2 ENuSMV 基础语法及语法扩展

7.2.1 NuSMV 原有语法

ENuSMV 是在 NuSMV 2.4.3 的基础上扩充了若干支持 ω-正规性质验证的语法成分后得到的模型检验工具。ENuSMV 在语法上完全兼容 NuSMV。本节将简要介绍如何使用 SMV Script 对系统进行建模，以及如何书写规约（这里只介绍 7.3 节中将要用到的语法部分）。关于 NuSMV 的详细语法定义，可参见 NuSMV 手册 [128]（事实上，在编译 NuSMV 的过程中，该参考手册会伴随生成）。

NuSMV 2.4.3 中定义了丰富的数据类型。包括布尔（0、1 或者 TRUE、FALSE）、整型、子界、枚举、字、数组、集合等类型。此外，还内置了 &、|、!、->、<-> 等基本的逻辑运算符（分别对应于数学符号 $\wedge$、$\vee$、$\neg$、$\rightarrow$、$\leftrightarrow$）；+、−、∗、/、mod 等整数运算符；union、in 等集合运算符（分别对应于数学符号 $\cup$、$\in$）；以及 next 表达式（稍后定义）。

在 NuSMV 中，通过关键字 VAR 来声明一个变量；通过关键字 DEFINE 来定义一个变量（即该变量是通过描述其与某些已经定义的变量之间的关系给出的）。比如，下面的语句

```
VAR     v_1 : boolean;
        v_2 : boolean;
DEFINE  v_3 := ! v_1 & v_2;
```

就新声明了两个布尔型变量 v_1 和 v_2，定义了一个新的布尔类型的变量 v_3。

在一个模块内部，可以分别通过使用 INIT、TRANS 等关键字来声明初始条件约束和迁移关系约束。比如，初始条件约束

```
INIT   v_2 <-> ! v_1;
```

就说明了“所有初始状态中布尔变量 v_1 和 v_2 的取值不同” 的特征；而迁移约束

```
TRANS   next(v_2) = v_1 & v_2;
```

就要求在“所有迁移中，下一状态中变量 v_2 的取值要与前一状态中 v_1 与 v_2 的合取值相同” 这样的约束。这里，next(v) 实际上是变量 v 的“次态版本” （直观的讲，v 与 next(v) 的关系如同位变元 z 与 z' 的关系）。

通过 INIT 和 TRANS 当然可以定义一个模块的初始条件和迁移关系。事实上，NuSMV 中式用的更加广泛的是以 ASSIGN 关键词引导的变量赋值语句为核心的模块定义方式。在 ASSIGN 语句中，有两个附加的关键字— init 和 next。前者用于给变量赋初始值，后者用于给变量赋“下一时刻” 的值。比如，语句

```
VAR  con : 0 .. 5;
ASSIGN
    init(con) := 0;
    next(con) := (con +1) mod 6;
```

就描述了一个模 6 循环的变量 con。其中，第一句的变量声明指明了 con 是一个取值介于 0 和 5 之间的子界类型（是整数类型的子集）。

通常，赋值语句会配合条件分支语句使用。条件分支语句分别以关键字 case 和 esac 结尾。这样的语句如

```
next(v) :=
    case
        cond_1 : expr_1;
        cond_2 : expr_2;
            .  .  .
        cond_n : expr_n;
    esac;
```

其中，每个 expr_i 都是布尔表达式。注意，在执行条件分支赋值语句时，目标变元总是被赋予（自上而下）最先被匹配的条件所引导表达式的值。因此，上述语句命令式语言的等价描述为

```
    if (cond_1)   then v := expr_1;
elseif (cond_2)   then v := expr_2;
        .  .  .
elseif (cond_n-1) then v := expr_n-1;
else    v:= expr_n;
```

在 NuSMV 中，要求分支赋值语句要匹配所有的可能。即：上述所有的 cond_i 的逻辑析取为 1。因此，在这种语句中最后一个分支条件往往是 1。

此外，每个 expr_i 可以是一个集合表达式。这时，当 cond_i 是第一个满足的布尔条件时，v 可以从 expr_i 中非确定的选择一个值。这样，就实现了非确定迁移。

在 NuSMV 中，支持两种公平性声明：一种以 JUSTICE 关键字引导，另外一种以 COMPASSION 引导。为同 CMU SMV 兼容，JUSTICE 也可以写作 FAIRNESS。

JUSTICE（或者 FAIRNESS）后面跟一个（关于位变元的）布尔表达式 expr。一条路径满足该公平性约束当且仅当 expr 在该路径中的无穷多个状态上被满足。因

此，该关键字实际上刻画的是 Büchi 接收条件。

COMPASSION 后面跟一对布尔表达式 expr_1 和 expr_2。一条路径满足该公平性约束当且仅当：若 expr_1 被该路径无穷多次满足则 expr_2 也被该路径无穷多次满足。因此，该关键字刻画的是 Streett 接收条件。目前，COMPASSION 关键字只能用在线性时序逻辑的验证过程中。

除基本布尔表达式外，除主模块外的每个模块实例中都内置了一个 running 变量。该变量为真当且仅当该模块实例被选择运行。因此，声明

```
FAIRNESS running;
```

后，就可以保证该模块实例被系统无穷多次调度。

在当前版本的 NuSMV 中，允许声明 CTL 和 LTL 规约。这两种规约分别以关键字 CTLSPEC 和 LTLSPEC 引导。为与 CMU SMV 兼容，CTLSPEC 也可以写作 SPEC。

NuSMV 中与时序连接子 X、U、R、F、G 相对应的语法记号分别为 X、U、V、F、G。与路径量词 A、E 相对应的语法记号分别为 A、E。在 LTL 规约声明中，还允许使用“过去时态” 时序连接子。此外，NuSMV 中还允许书写“不变式规约”（Invariant Specifications）、PSL 规约以及 RTCTL 规约。其中，不变式规约以关键字 INVARSPEC 引导，后面跟一个布尔表达式 expr。事实上，规约声明 INVARSPEC expr 和 CTLSPEC AG expr 等价。在 NuSMV 2.4.3 中，允许书写标准的 PSL 规约（以关键字 PSLSPEC 引导）。但是却只能验证那些能够用 LTL 或者 CTL 公式表示的性质。RTCTL 是带有实时约束的计算树逻辑，该种规约以关键字 SPEC 引导。关于这种逻辑的声明方法这里不做介绍。

NuSMV 中的文件是以模块（module）为单位进行组织的。一个模块内部可以包括变量声明/定义、变量赋值、公平性声明以及规约声明等（在 NuSMV 中，规约属于模块的一部分）。模块声明以关键字 MODULE 引导；模块声明可以带参数（不需指定参数类型）。模块可以以两种方式实例化— 同步方式（Synchronous）和异步方式（Asynchronous）。同步的模块实例在统一的时钟调度下运行，即每次进程调度时都会同时执行一步；而异步模块实例之间以交叠方式并发执行。在声明异步模块实例时，需要以关键字 process 指明。此外，文件中必须包含一个主模块 main，它没有参数，是模型执行的入口。

下面，给出一个 NuSMV 模块的书写实例（在本章中，如果是来源于他处的 SMV script 代码，或者使用第三方工具得到的代码，均会在相应地方声明。此外，本章中大部分代码可以在 ENuSMV 安装包的/nusmv/example/etl 目录下找到）。

例 7.2.1 考虑经典的“杯子倒水”问题：假设有一大一小两个杯子，容量分别是 5 升和 3 升。有一个水池，有充分多的水。假设这两个杯子上没有刻度，问能否倒出 4 升水（即：在某个时刻大杯中有 4 升水）？

要对该问题进行建模，需要定义两个变量：big 和 small，分别用以记录大、小两个杯子中当前的水量。为压缩状态空间，将这两个变量分别声明为 0..5 和 0..3 的子界类型。此外，还引入一个枚举型变量 act，它有六个可能的取值：big_small、small_big、big_well、well_big、well_small、small_well。比如，act = big_small 就表示当前动作是“从大杯向小杯倒水”；act = small_well 表示当前的动作是“将水从小杯倒向池中”。该变量的一个作用在于保证 big 和 small 变量的同步。比如：当大杯子向小杯倒入 w 升水后，就有 next(big) = big − w 以及 next(small) = small + w。由于杯子上都没有刻度，因此：每次从池中取水必须将杯子取满；每次向池中倒水必须将杯子倒空；从甲杯向乙杯倒水时，或者将甲杯倒空，或者将乙杯倒满。该问题的 SMV script 描述如下，待检验的性质声明为 CTLSPEC EF (big = 4)。 □

```
MODULE main
VAR
big   : 0..5;
small : 0..3;
act   : {well_big, well_small, big_small,
         small_big,big_well, small_well};
ASSIGN
init(big)   := 0;
init(small) := 0;
next(big)   :=
case
   act = well_big   : 5;
   act = big_well   : 0;
   act = well_small : big;
   act = small_well : big;
   (act = big_small) & (big >= (3-small)) : (3+big+small) mod 6;
   (act = big_small) & (big < (3-small))  : 0;
   (act = small_big) & (small >= (5-big)) : 5;
```

```
    (act = small_big) & (small <(5-big))   : (big+small) mod 6;
    1                 :  0;
esac;
next(small)  :=
case
    act = well_big   : small;
    act = big_well   : small;
    act = well_small : 3;
    act = small_well : 0;
    (act = big_small) & (big >= (3-small)) :  3;
    (act = big_small) & (big < (3-small))  : (small + big) mod 4;
    (act = small_big) & (small >= (5-big)) : (small+ big +3) mod 4;
    (act = small_big) & (small <(5-big))   :  0;
    1                 : 0;
esac;
next(act) :=
{well_big, well_small, big_small, small_big, big_well, small_well};

CTLSPEC EF big =4
```

这里需要说明的是，当满足 $(\texttt{act} = \texttt{big\_small})$ 及 $(\texttt{big} >= (3 - \texttt{small}))$ 时，为何要将 big 的值赋为 $(3 + \texttt{big} + \texttt{small}) \bmod 6$。事实上，$(3 + \texttt{big} + \texttt{small}) \bmod 6$ 与 $(\texttt{big} - (3 - \texttt{small}))$ 所指相同。但是 NuSMV 解析器只能推断这个表达式的值介于 -3 .. 5 之间。当将这样的值赋给介于 0 .. 5 之间的子界类型变量 big 时，就会报错。同样原因，当 $(\texttt{act} = \texttt{small\_big})$ 及 $(\texttt{small} < (5 - \texttt{big}))$ 时，big 的值赋为 $(\texttt{big} + \texttt{small}) \bmod 6$；当 $(\texttt{act} = \texttt{big\_small})$ 及 $(\texttt{big} < (3 - \texttt{small}))$ 时，small 的值赋为 $(\texttt{small} + \texttt{big}) \bmod 4$；当 $(\texttt{act} = \texttt{small\_big})$ 及 $(\texttt{small} >= (5 - \texttt{big}))$ 时，small 的值赋为 $(\texttt{small} + \texttt{big} + 3) \bmod 4$（事实上，它等同于 $\texttt{small} - (5 - \texttt{big})$）。这也是使用子界类型的变量时特别需要注意的地方。

7.2.2 ENuSMV 扩展语法

ENuSMV 在 NuSMV 的基础上增加了对 ETL_f 和 AFL 的符号化模型检验的支持。为此，在语法上进行了如下的扩充。

在 ENuSMV 中，允许通过描述自动机（NFW）的方式自定义时序连接子。为此，新增了三个关键字 CONNECTIVE、STATES 和 TRANSITIONS 用以定义自动机连接子。

自动机的状态集合的定义以关键字 STATES 引导。比如

```
STATES: st_1, >st_2, st_3, st_4<
```

就定义了一个含有 4 个状态的状态集合。其中，前面冠以 > 的状态是初始状态，比如 st_2；后面缀以 < 的状态是接收状态（或者称终止状态），比如 st_4。需要注意的是，> 或者 < 并不是状态名的一部分。此外，在 ENuSMV 中，状态集合要求包含唯一的初始状态（否则解析器会报错），并且要有至少有一个的终止状态（否则会报出一个警告）。一个状态既是初始状态又是终止状态是允许的（虽然包含这种状态的连接子只会产生出等价于 $true$ 的公式）。

自动机的迁移关系以关键字 TRANSITIONS 引导，并且需要配合 case 语句使用。比如，下面的语句

```
TRANSITIONS (q_1)
case
    a_1 : q_2;
    a_2 : {q_2,q_3};
esac;
```

就描述了状态 q_1 的（部分）迁移关系：读入字母 a_1 后，会迁移到状态 q_2，读入字母 a_2 后，会迁移到 q_2 或者 q_3。当右边的状态集中仅包含一个状态时，花括号可以省略不写。注意，一个状态可以具有多个迁移声明。比如，下面的两个声明

```
TRANSITIONS(q_1)
case
    a_1 :  q_1;
    a_2 :  {q_2, q_3};
    a_1 :  q_2;
esac;
```

```
TRANSITIONS(q_1)
    case
        a_2 :  q_1;
        a_3 :  q_2;
        a_1 :  {q_1, q_4};
    esac;
```

可以等价的合并为如下的单个迁移声明。

```
TRANSITIONS(q_1)
case
    a_1 :  {q_1, q_2, q_4};
```

```
        a_2 :  {q_1, q_2, q_3};
        a_3 :  q_2;
    esac;
```

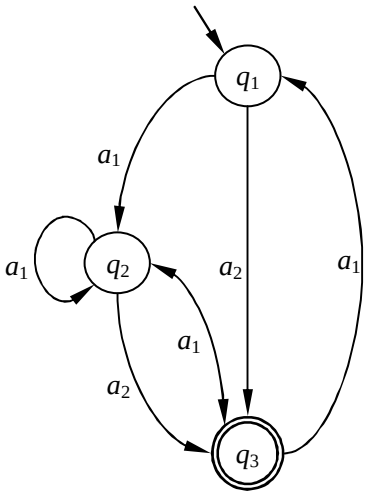
在自动机的状态集中，接收状态可以没有迁移声明。但是对于每个非终止状态，必须至少包含一个迁移声明。因为不产生任何迁移的非终止状态是冗余的。

关键字 CONNECTIVE 用以引导连接子的声明。其语法为

```
CONNECTIVE [conn_name] ([alphabet])
[connective_body]
```

其中，[alphabet] 是自动机的字母表，该部分必须非空；[connective_body] 中包含一个状态集合声明和若干个迁移声明。

例 7.2.2 图 *7.1* 给出了某自动机连接子 conn 的定义示例。其中，标志符 q_1、q_2、q_3 分别对应状态 q_1、q_2，q_3；标志符 a_1、a_2 分别对应字母 a_1、a_2。 □




```
CONNECTIVE  conn (a_1, a_2)
STATES: >q_1, q_2, q_3<
TRANSITIONS(q_1)
case a_1 : q_2; a_2 : q_3; esac;
TRANSITIONS(q_2)
case
    a_1 :  {q_2, q_3};
    a_2 :  q_3;
esac;
TRANSITIONS(q_3)
case  a_1 : {q_1, q_2}; esac;
```

图 7.1 ENuSMV 中自动机连接子定义示例

在定义了自动机连接子之后，就可以定义 ETL_f 公式了。这时，除了标准的布尔连接子之外，还可以使用 X 连接子（对应于 ETL_f 中的 $\bigcirc$ 连接子）和已经声明的自动机连接子。比如，conn(expr_1, expr_2) 就是一个合法的 ETL_f 公式声明。其中 expr_1 和 expr_2 也是 ETL_f 公式声明。ETL_f 规约以关键字 ETLSPEC 引导，后面跟一个 ETL_f 公式声明。

ENuSMV 在版本 1.1 中增加了对 AFL 公式符号化模型检验的支持。在第 5 章介绍 AFL 公式时，曾将自动机构造子的字母表固定为 2^{AP}。然而，这对于书写规约造

成很大的不便。为此，在 ENuSMV 1.1 中，将 AFL 自动机构造子的字母表看作是布尔公式。于是，可以用声明 ETL_f 自动机公式的办法来声明 AFL 的自动机构造子，只是被限制为只能连接布尔公式。此外，还引入了 U、V、abort、monitor、|->、T 这几个时序连接子语法记号，它们分别对应于 AFL 时序连接子 U、R、abort!、monitor、trigger、leads。比如

```
conn(b_1,b_2) abort b
conn(b_1,b_2) trigger expr
```

都是合法的 AFL 公式声明。其中，b_1、b_2 以及 b 都是布尔表达式。AFL 规约以关键字 AFLSPEC 引导。

7.2.3 ENuSMV 验证执行过程

ENuSMV 提供了两种检验执行方式：交互方式（Interactive Manner）以及批处理方式（Batch Manner）。

以交互方式执行 ENuSMV 只需在命令行中键入./NuSMV -int 即可。NuSMV 2.4.3 提供了丰富的交互式命令，比如建立模型、验证规约、统计可达状态数目等等（详见文献 [128] 的第三章）。这些交互式命令在 ENuSMV 1.0 和 1.1 中仍然有效。

在命令行中执行 ENuSMV 时，如果命令行参数中不加入选项 -int 即以批处理方式执行。事实上，多数的交互式命令均能够以命令行选项的形式在批处理方式下加入。目前，对 ETL_f 以及 AFL 公式检验只能以批处理方式执行。其原因在于：用户定义自动机连接子时，很难在一行命令行内写下（在后续版本，可能考虑增加从文件读入自动机连接子的交互式命令）。验证需要的统计信息，可以通过命令行参数获得。此外，ENuSMV 1.1 在检验 LTL、ETL 以及 AFL 时，特别增加了对状态统计的支持— 执行检验前会打印出由规约构建的 tableau 的 SMV 表示，并且统计出执行检验时搜索的状态数目以及可达状态数目。

当执行检验完毕，ENuSMV 会报告模型检验的结果。当规约性质被违反时，还会给出一条反例路径。ENuSMV 会打印出该路径上每个状态中取值（较上一状态）发生变化的变量值。比如，在例 7.2.1 中，执行检验的结果是 *true*。为获得取得 4 升水的操作过程，可以将规约性质取反，即检验性质 AG ! (big = 4)。这样，从 ENuSMV 给出的反例路径中即可得到规划。

7.3 实验结果

ENuSMV 的两个版本，分别增加了对 ETL_f 和 AFL 的符号化模型检验的支持。这里，给出使用 ENuSMV 执行模型检验的若干实验结果。如非特别声明，本章中的实验结果均在如下平台上获得：

处理器 ：Intel Core Duo2 (2.66 GHZ)；

内存 ：2048 M 内存；

操作系统 ：Fedora Linux 7.0；

编译器 ：GCC 4.1.2；

ENuSMV 版本 ：7.3.1 节实验所采用版本均为 1.0 (2008-06-12 released)；7.3.2 节实验所采用版本均为 1.1 (2008-11-15 released)。

7.3.1 ETL_f 模型检验结果

本小节给出使用 ENuSMV 检验 ETL_f 的若干实验结果。这里，主要考虑以下几个方面：

1. 比较 ETL_f 同 LTL、CTL 之间的模型检验效率。这时需要检验若干 star-free 的 ω-正规性质。
2. 测试 ETL_f 对非 star-free 的 ω-正规性质的模型检验效率。
3. 测试 ETL_f 所能够处理的模型规模。

这里，“规模” 的意义有两种：一种是模型文件本身的代码长度；一种是模型检验过程中生成的可达状态的数目。在本节，主要针对前者实行测试；下一节，将针对后者进行测试。事实上，在固定的实验平台上，当规约的复杂度远小于模型时，这两种规模反应的是工具（或者说算法实现）的属性，与性质基本无关。

本节及下节实验结果中的时间均是取 5 次检验的平均值，该结果通过使用 Linux 中的 `time` 命令取得（以用户时间和内核时间之和计算，包括输出反例的时间）。表中的状态数目，均是指模型检验过程中生成的可达状态数目。这个数目按照如下方式取得：对于 CTL 性质，可以在检验时通过增加 `-r` 参数获得；对于 LTL、ETL_f 以及后面的 AFL 性质，是通过调用 CUDD 库中的状态统计函数得到的。如果希望重现下列实验的结果，可以采取如下做法：首先，在 ENuSMV 源码文件 /nusmv/src/ltl/ltl.c 的函数 `Ltl_CheckLtlSpec` 中找到语句

```
s0 = eval_ctl_spec(fsm, bdd_enc, spec_formula, Nil);
```

在此之前添加语句

```
BddFsm_print_reachable_states_info(fsm, false, nusmv_stdout);
```

然后重新编译工程即可。此外，如果将上面第二个参数改为 `true`，还会打印出每个可达状态的详细信息。

7.3.1.1 令牌环网

首先，以一个同步模型为例，比较 CTL、LTL 以及 ETL_f 三者之间的模型检验效率。考虑如下的“令牌环网” 问题。

一个令牌环网络由 n 个网络节点构成。这些网络节点首尾相连，构成环状拓扑结构。这些处理单元以同步方式工作。在每个时刻，每个网络节点都有可能进行 I/O 操作 (即: 网络报文的收发)。为了使网络上的数据保持一致，这些网络节点间需要维系一个“令牌” — 拥有令牌的网络节点将数据发送 (且必须发送) 到网络上，没有令牌的节点如果需要进行 I/O，则只能进行报文收取。同时，拥有令牌的节点将令牌传递给下一个网络节点。

在这里，对令牌环网关心两个问题：

安全性 ：任何时刻网络上最多有一个网络节点拥有令牌。

活性 ：任何网络节点都能够无穷多次得到令牌。

单个网络节点的 SMV script 描述如下。其中，参数 `init_tk` 表示该网络节点最初是否拥有令牌；布尔变量 `token` 和 `io` 则分别指明了该网络节点当前是否拥有令牌以及是否正在进行I/O 操作。

```
MODULE node(init_tk, input)
VAR
    token   : boolean;
    io      : boolean;
ASSIGN
    init(token) := init_tk;
    next(token) := input;
    init(io)    :=
    case
        token : 1;
        1     : {0,1};
    esac;
    next(io)    :=
```

```
case
    input : 1;
    1     : {0,1};
esac;
```

接下来，在主模块内以同步方式声明 n 个 node 的实例 node_0，...，node_(n-1)。其中，node_0 初始拥有令牌，第 $(i+1) \mod n$ 个网络节点的 input 与第 i 个网络节点的 token 相连接。即：在主模块中声明

```
node_0      : node(1, cell_(n-1).token);
node_1      : node(0, cell_0.token);
. . .
node_(n-1)  : node(0, cell_(n-2).token);
```

安全性性质和活性性质的 LTL（resp. CTL）描述分别为

$$\bigwedge_{0\le i<j<n} \mathsf{G}(\neg p_i \vee \neg p_j) \quad (\text{resp.} \bigwedge_{0\le i<j<n} \mathsf{AG}(\neg p_i \vee \neg p_j))$$

以及

$$\bigwedge_{0\le i<n} \mathsf{G\ F}\ p_i \quad (\text{resp.} \bigwedge_{0\le i<n} \mathsf{AG\ AF}\ p_i)$$

其中，命题 p_i 对应于 node_i.token。以下，为一致起见，在行文中均用数学形式书写规约，并且会指明命题变元与程序变量的对应关系。

为了书写 ETL_f 规约，引入 NFW $\mathcal{A}_{\mathsf{F}} = \langle\{a_1, a_2\}, \{q_1, q_2\}, \delta_{\mathsf{F}}, q_1, \{q_2\}\rangle$。其中 $\delta_{\mathsf{F}}(q_1, a_1) = \{q_1\}$，$\delta_{\mathsf{F}}(q_1, a_2) = q_2$，$\delta_{\mathsf{F}}(q_2, a_1) = \delta_{\mathsf{F}}(q_2, a_2) = \emptyset$。容易检验，安全性性质和活性性质可以分别用 ETL_f 公式

$$\bigwedge_{0\le i<j<n} \neg\mathcal{A}_{\mathsf{F}}(true, p_i \wedge p_j)$$

以及

$$\bigwedge_{0\le i<n} \neg\mathcal{A}_{\mathsf{F}}(true, \neg\mathcal{A}_{\mathsf{F}}(true, p_i))$$

表示。

“令牌环网” 的安全性性质和活性性质的模型检验结果如表 7.1 所示。由于各模块的对称性，检验安全性质和活性性质时只取规定的 i、j 即可。在表 7.1 中，#states 所指的为模型状态检验过程中生成的可达状态数（以下实验相同）。对于 CTL 而言，

表 7.1 令牌环网模型检验结果

n	安全性质						活性性质				
	#states			**#time**(sec)			**#states**		**#time**(sec)		
	CTL	LTL	ETL_f	CTL	LTL	ETL_f	LTL	ETL_f	CTL	LTL	ETL_f
3	12	24	192	0.006	0.008	0.009	48	768	0.007	0.008	0.008
4	32	64	512	0.007	0.008	0.008	128	2048	0.007	0.008	0.008
5	80	160	1280	0.007	0.009	0.009	320	5120	0.007	0.008	0.008
6	192	384	3072	0.008	0.009	0.009	768	12288	0.007	0.008	0.009
7	448	892	7136	0.008	0.009	0.009	1784	28544	0.008	0.009	0.009
8	1024	2048	16384	0.008	0.009	0.009	4096	65536	0.008	0.009	0.011

当取相同的 n 时，检验安全性质和活性性质时产生的可达状态数相同（均为模型中的可达状态数目）。因此，未将关于活性性质 CTL 模型检验的可达状态数列出。

从上述实验结果中可以看出：对于同一个参数 n、执行 CTL、LTL 以及 ETL_f 符号化模型检验时生成的可达状态数目存在一个固定的比值。对于安全性质，该比值为 $1:2:16$，对于公平性质，该比值为 $1:4:64$。这种现象是必然的— 对于 LTL 和 ETL_f 而言，最终要转化为 CTL 的模型检验。该模型就是系统模型与该公式之非对应的 tableau 的合成。由于构建 tableau 需要引入额外的位变元，因此三者最终模型（原模型与 tableau 的合成）的规模之间存在一个固定的比值。从表中数据还可以看出，CTL、LTL 以及 ETL_f 执行该模型检验的时间开销基本相同。

7.3.1.2 异步非门环电路

现在考虑一个异步模型：非门环电路（该模型是 NuSMV 2.4.3. 一个自带的示例，其模型文件为 /nusmv/examples/ctl-ltl/ring.smv）。

一个非门环由若干个锁存器构成，这些锁存器首尾相连，构成一个环。每个锁存器在每次的输出值为上次输入值的取反。

该问题的 SMV script 的描述如下。

```
MODULE main
VAR
    gate_1 : process inverter(gate_n.output);
    gate_2 : process inverter(gate_1.output);
```

```
    . . .
    gate_n : process inverter(gate_(n-1).output);
MODULE inverter(input)
VAR
    output : boolean;
ASSIGN
    init(output) := 0;
    next(output) := !input;
FAIRNESS   running
```

在这个实验中，主要关心系统能否达到“稳态”。即：能否经过在有穷次迁移后，系统中每个锁存器的输出不再发生改变。换言之，是否每个锁存器都会无穷多次输出 0 和 1。该性质用 LTL（resp. CTL）描述为

$$\bigwedge_{1\le i\le n}(\mathsf{G}\ \mathsf{F}\ p_i \wedge \mathsf{G}\ \mathsf{F}\ \neg p_i) \quad (\text{resp.} \bigwedge_{1\le i\le n}(\mathsf{AG}\ \mathsf{AF}\ p_i \wedge \mathsf{AG}\ \mathsf{AF}\ \neg p_i))$$

其中，命题 p_i 对应于 `gate_i.output`。这样，该性质的 ETL_f 描述为

$$\bigwedge_{1\le i\le n} \neg\mathcal{A}_{\mathsf{F}}(true, \neg\mathcal{A}_{\mathsf{F}}(true, p_i)) \wedge \neg\mathcal{A}_{\mathsf{F}}(true, \neg\mathcal{A}_{\mathsf{F}}(true, \neg p_i))$$

其中，$\mathcal{A}_{\mathsf{F}}$ 与 7.3.1.1 节中定义相同。

表 7.2 异步非门环电路模型检验结果

n	**#reachable states**			**#time**(sec)		
	CTL	LTL	ETL_f	CTL	LTL	ETL_f
6	63	252	4032	0.011	0.012	0.016
9	511	2044	32704	0.013	0.040	0.040
12	4095	16380	262080	0.024	0.050	0.052
15	32767	131068	2.09709e+09	0.046	0.307	0.357

异步非门环的模型检验结果如表 7.2 所示。在本实验中，当 n 取奇数的时候结论成立；当 n 取偶数时存在反例路径。由于电路的对称性，这里只检验某个固定的 i；同时，由于 p_i 和 $\neg p_i$ 的对称性，这里只检验等价于 $\mathsf{G}\ \mathsf{F}\ p_i$ 的性质。在这个例子中，检验 CTL、LTL 以及 ETL_f 的空间开销比为 $1:4:64$。因此，构建 LTL 及

ETL_f 引入的位变元分别为 2 和 6。从表中可以看出，执行 CTL 模型检验的时间开销最小，而执行 LTL 和 ETL_f 模型检验的时间开销基本相同。

7.3.1.3 非 star-free 性质的检验

在前面的实验中，检验的都是能够用 LTL 描述的性质，这类性质等价于 star-free 的 ω-正规性质。事实上，许多 ω-正规性质是不能被 LTL 表述的，下面说明如何寻找这样的性质。

取定 $p \in AP$，考虑这样的一系列线性结构 π_0，π_i，...。其中，对于每个 $i \in \mathbb{N}$ 以及 $j \in \mathbb{N}$ 有：$p \notin \pi_i(j)$ 当且仅当 $j = i$。于是，可以证明下面的定理。

定理 7.1 ([23]) *设 LTL 公式中 φ 中含有 k 个 X 连接子，则对于任意的 $m \geq n \geq k$，$\pi_m \models \varphi$ 当且仅当 $\pi, n \models \varphi$。*

上面的定理实际上说明了：当 m 和 n 超过 φ 中 X 算子的数目时，φ 将无法区分 π_m 和 π_n。于是，立即可以证明：

1. “命题 p 在每个偶数时刻被保持” 无法被 LTL 公式表达[23]；
2. 设布尔表达式 b_1 与 b_2 不等价，且 $b_1 \wedge b_2$ 是可满足的，则 FL 公式 $(b_1; b_2)^*\mathsf{leads}\, b$ 无法被 LTL 公式表达。

注意：LTL 公式 $\eta = \psi \wedge \mathsf{G}(\psi \rightarrow \mathsf{XX}\psi)$ 并不是性质 1 的一个表述。比如，设 π 是仅在 $\mathbb{E} \cup \{3\}$ 处使得 p 成立的线性结构，则 π 是性质 1 的模型，而不是 η 的模型。更一般的，对其他类似的周期性质（或者说“采样性质”）也有相同结论。

对于性质 2 而言，如果 b_1 与 b_2 等价，则退化成了 $b_1\mathsf{U}(b_1 \wedge p)$。但一般而言，只要 r 不是 star-free 的，那么形如 r leads ψ、r $\mathsf{trigger}$ ψ、r abort ψ 的性质都不存在等价的 LTL 表述。而这些性质是在硬件电路验证中被广泛使用的。

现在，给出使用 ENuSMV 检验非 star-free 的 ω-正规性质的例子。这里，仍以 7.3.1.1 节中令牌环网络的模型为例，这检验如下的性质：

周期性质 ：设网络中有 n 个节点，则节点 0 在每个 $k \times n$ 时刻（$k \in \mathbb{N}$）都进行 I/O 操作。

刻画规约所需的时序连接子可以用 NFW $\mathcal{A}_n = \langle \{a_1, a_2\}, \{q_0, \ldots, q_n\}, \delta_n, q_0, \{q_n\} \rangle$ 描述。其中，对每个 $0 \leq i < n-1$，$\delta_n(q_i, a_1) = \{q_{i+1}\}$；$\delta_n(q_{n-1}, a_1) = \{q_0\}$；$\delta_n(q_0, a_2) = \{q_n\}$；对每个 $0 < i \leq n$，$\delta_n(q_i, a_2) = \emptyset$。这样，待检验的性质可以用 ETL_f 表示为 $\neg\mathcal{A}_n(true, \neg p)$。其中，命题 p 对应于 `node_0.io`。

本实验的结果如表 7.3 所示。在该实验中，网络节点每增加 1，执行检验生成的可达状态数大约增加 10 倍，但是检验所需的时间开销并未达到空间开销的增加

表 7.3 令牌环网周期性质检验结果

n	# reachable states	# time(sec)
3	3072	0.009
4	32768	0.011
5	327680	0.018
6	3.14573e+06	0.025
7	2.93601e+07	0.034
8	2.68435e+08	0.048
9	2.41592e+09	0.058

幅度。

7.3.1.4 SELinux-安全策略配置

最后，为测试 ENuSMV 能够处理的模型规模，重现了文 [129] 中提到的 SELinux 安全策略配置文件的安全性检查。依照该文中的做法，使用 Slat 工具[130] 将 SELinux 的安全策略配置文件转化成了 SMV script。得到的代码长度（模型部分）为 120147 行，模型文件大约为 6 M（文件名：/nusmv/examples/etl/flow.smv）。

这里，检验了文 [129] 中关心的两条“事件断言”（Event Assertion）和一条“顺序断言”（Order Assertion）。实验结果如表 7.4 所示。特别需要说明的是，本实验结果是在 512 M 内存以及 3 GHZ 处理器平台上取得的（软件环境未发生变化）。此外，这里还列出了执行模型检验时生成的状态总数。通过与可达状态数目的比较，可以看出这个由工具生成的模型并不低效。

表 7.4 SELinux 安全策略配置文件模型检验结果

Property	# reachable states/# total states	# time (sec)
Event assertion 1	1.41717 e+ 10/2.39638 e+ 10	37.969
Event assertion 2	9.06992 e+ 11/1.53368 e+ 12	37.969
Order assertion	2.38435 e+ 10/4.79275 e+ 10	38.066

7.3.2 AFL 模型检验结果

同上一小节，在使用 ENuSMV 对 AFL 执行模型检验时，主要考虑如下几个方面：

1. 比较 AFL 同 LTL、CTL 之间的模型检验效率。
2. 测试 AFL 对非 star-free 的 ω-正规性质的模型检验效率。
3. 测试 AFL 所能够处理的模型规模。本节主要关注模型检验过程中可达状态的规模。

7.3.2.1 哲学家就餐问题

首先，考虑经典的“哲学家就餐”问题，该问题描述如下。

有 n 个哲学家（$n \geq 2$）围坐在一张餐桌旁。每两个毗邻的哲学家之间放有一根筷子。哲学家们除了吃饭就是思考 — 每当某个哲学家感到饥饿时，他就准备吃饭，但在此之前，必须取得他身边的两根筷子。吃完饭后，放下筷子，继续思考。

这里，希望检验“可调度性”，即：存在一条执行路径，使得每个哲学家都有机会吃饭。

为对该问题进行建模，首先引入 n 个枚举变量 fork_0，...，fork_(n − 1)。它们可以取三个值：Idle、L_Occ、R_Occ，分别表示这根筷子当前处于空闲状态、被其左边的思考着拿起、被其右边的思考着拿起。于是，单个思考者的行为可以用如下的 SMV script 代码描述。

```
MODULE thinker(left,right)
VAR
    eating : boolean;
ASSIGN
    init(eating):=0;
    next(eating) :=
    case
        !eating & left=R_Occ & right=L_Occ : 1;
        1: 0;
    esac;
    next(left) :=
    case
```

```
            !eating & left=Idle : R_Occ;
            !eating & right=R_Occ & left =R_Occ: Idle;
            eating : Idle;
            1      : left;
        esac;
        next(right):=
        case
            !eating & right= Idle : L_Occ;
            !eating & left=L_Occ & right = L_Occ : Idle;
            eating : Idle;
            1      : right;
        esac;
    FAIRNESS running;
```

接下来，在主模块内声明 n 个 thinker 的实例

```
thinker_0       :   process thinker(fork_0,fork_1);
thinker_1       :   process thinker(fork_1,fork_2);
. . .
thinker_(n-1)   :   process thinker(fork_n-1,fork_0);
```

即可完成对本问题的建模。由于可调度性是针对单条路径的性质，因此只能直接由 CTL 规约描述。令 p_i 对应的命题为 thinker_i.eating。则可调度性的 CTL 描述为

$$\mathsf{EF}(p_0 \wedge (\mathsf{EF}p_1 \wedge \ldots (\mathsf{EF}p_{n-1})))$$

（这里，由于程序的对称性，只需对某个固定的 i 进行检验即可）。为了使用 LTL 规约以及 AFL 规约，这里准备检验上述规约的取非。这样，模型满足 CTL 规约当且仅当违反 LTL 及 AFL 规约。取反后的可调度性用 LTL 描述为

$$\mathsf{G}(\neg p_0 \vee (\mathsf{G}\neg p_1 \vee \ldots (\mathsf{G}\neg p_{n-1})))$$

在本例中，用同样的公式表示 AFL 规约。

“哲学家就餐问题” 的 ENuSMV 模型检验结果如表 7.5 所示。由于 AFL 和 LTL 规约采用相同的描述，它们会得到相同的 tableau，因此可达状态数目严格相同，这样执行模型检验所需的时间也大致相等。从表中可以看出：检验 LTL/AFL 性质的空间开销（可达状态数）约为检验 CTL 性质的 2 倍。

表 7.5 “哲学家就餐”问题模型检验结果

#thinkers	#states		#time(sec)		
	CTL	AFL	CTL	LTL	AFL
5	343	686	0.013	0.029	0.030
6	1135	2270	0.020	0.046	0.046
7	3545	7090	0.037	0.122	0.124
8	11395	22790	0.070	0.310	0.311

7.3.2.2 DME 电路问题

下面考虑一个工业界的实际用例— Martin 的 DME（Distributed Mutual Exclusion）电路问题。一个 DME 电路由若干个仲裁器构成，这些仲裁器构成了一个环形网络。(关于该电路的详细描述见文 [131])。

本实验的 SMV 代码是 NuSMV 中自带的。它可以在NuSMV 2.4.3 的安装目录下的子目录 /nusmv/examples/bmc/dme 下找到。在文 [16] 中，提到了关于这个模型的如下两个重要性质（但这里的模型文件与文 [16] 中提到模型文件相差很大）：

安全性 ：在任何时刻，不可能有两个仲裁器同时被响应；

响应性 ：任何一个仲裁器发出的请求将来必将被响应。

假设电路中有 n 个仲裁器，与第 i 个仲裁器请求/响应信号量相关的命题为 req_i/ack_i（分别对应代码中的 `req_i`/`ack_i`），则“安全性” 和“响应性” 对应的 LTL 规约分别为$\mathsf{G}\bigwedge_{1\leq i<j\leq n}(\neg ack_i \vee \neg ack_j)$ 和$\mathsf{G}(req_i \rightarrow ack_i)$。

这一次，不再将 LTL 规约和 AFL 规约采用同样的表示。这里，引入两个 NFW $\mathcal{A}_{\mathsf{L}}$ 和 $\mathcal{A}_{\mathsf{F}}$：

- $\mathcal{A}_{\mathsf{L}} = \langle\{a\}, \{q\}, \delta_{\mathsf{L}}, \{q\}, \{q\}\rangle$，其中 $\delta_{\mathsf{L}}(q, a) = \{q\}$；
- $\mathcal{A}_{\mathsf{F}} = \langle\{a_1, a_2\}, \{q_1, q_2\}, q_1, \{q_2\}\rangle$，其中 $\delta_{\mathsf{F}}(q_1, a_1) = \{q_1\}$，$\delta_{\mathsf{F}}(q_1, a_2) = \{q_2\}$，$\delta_{\mathsf{F}}(q_2, a_1) = \delta_{\mathsf{F}}(q_2, a_2) = \emptyset$。

再令 $b = \bigvee_{1\leq i<j\leq n}(ack_i \wedge ack_j)$，于是，安全性和响应性可以分别表示为 AFL 公式 $\neg\mathcal{A}_{\mathsf{L}}(true)$ $\mathsf{abort!}$ b 和 $\mathcal{A}_{\mathsf{F}}(true, req_i)$ $\mathsf{trigger}(\mathcal{A}(true, ack))$ leads $true$。

关于该实验安全性和活性的 ENuSMV 模型检验结果分别见表 7.6 和表 7.7。该实验同时也测试了 ENuSMV 在该平台上能够处理的问题规模。在本实验中，当仲裁器的数目超过 4 时，检验无法完成（即使是使用 NuSMV的基础 CTL检验算法）。因此，在这个平台上，ENuSMV 能处理的状态规模介于 $5.6\mathrm{e}+21$ （56 万

表 7.6 DME 模型安全性性质模型检验结果

# cells	#reachable states			#time (sec)		
	CTL	LTL	AFL	CTL	LTL	AFL
2	4.74164e+12	9.48329e+12	7.58663e+13	0.904	4.067	2.160
3	1.10428e+19	2.20857e+19	1.76685e+20	4.027	8.099	12.788
4	3.5499e+26	7.09979e+26	5.85376e+27	-	-	-

表 7.7 DME 模型响应性性质模型检验结果

# cells	#reachable states			#time (sec)		
	CTL	LTL	AFL	CTL	LTL	AFL
2	4.74164e+12	1.89666e+13	2.42772e+15	0.0962	2.123	3.395
3	1.10428e+19	4.41713e+19	5.65393e+21	8.046	15.102	22.317
4	3.5499e+26	1.41996e+27	1.81755e+29	-	-	-

亿亿）和 $3.5e+26$ 之间（这里没有使用- `bmc` 选项）。对于安全性性质而言，执行 LTL 检验和 AFL 检验时的空间代价约为 $1:8$；对于活性性质而言，这个比值约为 $1:128$。这里，也有一个有趣的现象— 在检验含有两个仲裁器的 DME 线路的安全性质时，AFL 性质反而能在更短的时间内结束检验。

7.3.2.3 二进制累加器实验

最后，采用 AFL 检验一组非 star-free 的 ω-正规性质。考虑如下的“二进制累加器” 电路问题：一个“二进制累加器” 由若干个模二的计数器单元 `bit_0`，...，`bit_n` 构成。每个计数器单元的 SMV script 代码描述如下。

```
MODULE counter_cell(carry_in)
VAR
    pre_value : boolean;
    value     : boolean;
ASSIGN
    init(pre_value) := 0;
    init(value)     := 0;
```

```
        next(pre_value) := value;
        next(value)     := (value + carry_in) mod 2;
    DEFINE
        carry_out       : = pre_value & carry_in;
```

（注：本例是在 NuSMV 的 /nusmv/examples/ctl-ltl/counter.smv 的基础上修改得到的— 在每个二进制累加器中增加了一个布尔变量 pre_value。）

该电路由这些计数器单元以串联方式构成— bit_0 的 carry_in 置为 1；bit_(i+1) 的 carry_in 与 bit_i 的 carry_out 连接；整个电路以同步方式运转。即：在主模块中声明

```
bit_0 : counter_cell(1);
bit_1 : counter_cell(bit_0.carry_out);
. . .
bit_n : counter_cell(bit_(n-1).carry_out);
```

该电路的示意图如图 7.2 所示。

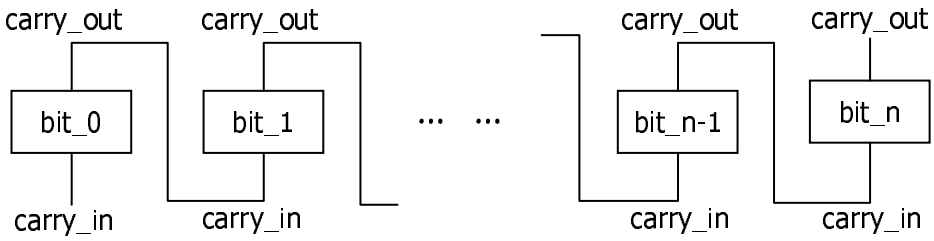


图 7.2 二进制累加器示意图

这里，主要检验以下两个性质：

周期性：在每个 $2k+2$ 时刻，bit_0 都会产生进位（$k \in \mathbb{N}$）。

监测性：从时刻 1 起，一旦波形匹配 $true;(true;p_0)^*$，那么 p_1 就会在下两个时刻发生。其中，命题 p_0 和 p_1 分别表示 bit_0.carry_out 和 bit_1.carry_out。

由前一节中的结论可知，这两个性质都不是 star-free 的，因而不能被 LTL 表示。周期性性质和监测性性质用 AFL 分别描述为 $\neg \mathsf{X}\mathcal{A}_{\mathsf{E}}(true, \neg p_0)$ leads $true$ 和 $\mathsf{X}(\mathcal{A}_{\mathsf{E}}(true, p_0)$ trigger $(\mathsf{X}p_1))$。这里，$\mathcal{A}_{\mathsf{E}} = \langle\{a_1, a_2\}, \{q_1, q_2, q_3\}, \delta_{\mathsf{E}}, q_1, \{q_3\}\rangle$，其中，$\delta_{\mathsf{E}}(q_1, a_1) = \{q_2\}$，$\delta_{\mathsf{E}}(q_2, a_1) = \{q_1\}$，$\delta_{\mathsf{E}}(q_2, a_2) = \{q_3\}$ 以及 $\delta_{\mathsf{E}}(q_1, a_2) = \delta_{\mathsf{E}}(q_3, a_1) = \delta_{\mathsf{E}}(q_3, a_2) = \emptyset$。

“二进制累加器电路” 模型检验结果如表 7.8 所示。在当前版本的 ENuSMV 实现中，为了直观起见，将 trigger 的语义进行了稍微的修改：“$\pi, i \models \mathcal{A}$ trigger ψ

表 7.8 “二进制累加器电路” 模型检验结果

n	周期性		监测性	
	#reachable states	**# time**(sec)	**#reachable states**	**# time**(sec)
5	17408	0.010	2176	0.008
6	33792	0.010	4224	0.008
7	66560	0.011	8320	0.009
8	132096	0.012	16512	0.011
9	263168	0.019	32896	0.018
10	525312	0.024	65664	0.021

当且仅当对每个 $j \geq i$，若 $\pi[i,j] \in \mathbf{L}(\mathcal{A})$ 则 $\pi^{j+1} \models \psi$”。这是监测性性质规约中 trigger 连接子子第二个参数中只有一个 X 的原因。

7.4 本章小结

本章介绍了符号化模型检验工具 ENuSMV 的使用以及若干实验结果。这些实验结果显示：ETL_f 和 AFL 规约都能较为高效的被检验。检验这些规约所花的空间代价是 CTL 的常数倍。目前，它能处理 12 万行规模以及可达状态数目超过 10^{21} 的模型。

目前的 ENuSMV 支持 ETL_f 和 AFL 两种扩展。同 LTL 的符号化模型检验算法类似，这两种时序逻辑的符号化模型检验算法都是通过构建公式的 tableau 将线性框架下的时序逻辑的模型检验问题转化为 CTL 的模型检验问题。这种做法的特点在于：一方面，相比于等价的 CTL 公式的检验过程增加了时间/空间的开销；另一方面，它增加了对用户自定义时序连接子的支持，能够检验全部的 ω-正规性质。这样，用户在使用模型检验工具时，可以有多种选择— 对于能够用 CTL/LTL 描述的性质，尽量使用 CTL/LTL 书写规约；对于非 star-free 的 ω-正规性质，应当选用 ETL_f 或者 AFL 书写规约。

下几个版本的 ENuSMV 准备实现如下扩展。

- 增加对 ν-范式线性 μ-演算的符号化模型检验的支持。
- 基于 ATL_r 模型检验算法，增加 SMV 模型间语言等价性检查的支持（允许变量别名映射）。
- 完成从 FL 到 AFL 的前端转换器，使得标准 PSL 的模型检验能够被支持。
- 增加对这些逻辑交互式检验的命令行支持。

第八章　结束语

8.1　本文的主要工作

本文主要关注两大类等价于 ω-正规语言的扩展时序逻辑的推理问题以及符号化模型检验问题。本文主要工作总结如下：

1. 首先，给出了三种使用非确定自动机作为时序连接子的扩展时序逻辑（即：ETL_l、ETL_f、ETL_r）的可靠完备的公理系统（$\mathscr{L}$、$\mathscr{F}$、$\mathscr{R}$）；进而，给出了这些时序逻辑片段的实例公理化方法。
2. 针对模态 μ-演算和线性 μ-演算，给出了新的博弈系统。该种博弈可以看作是 parity game 的一种变种。建立了该种博弈与 μ-演算公式可满足性之间的联系。进而，利用该种博弈理论给出了 $\mathscr{G}$ 系统和 $\mathscr{H}$ 系统完备性的相对简洁的证明。
3. 给出了三种使用交错自动机的扩展时序逻辑（即：ATL_f、ATL_l、ATL_r）的基于 BDD 的符号化模型检验算法；提出了 PSL 的变种— APSL，并给出了其线性部分（AFL）的符号化模型检验算法。
4. 基于可编码的自动机转换以及线性 μ-演算博弈方法的变形，分别给出了具有一般形式的线性 μ-演算公式和具有特殊形式的线性 μ-演算公式（ν-范式）的符号化模型检验算法。
5. 在 NuSMV 的基础上，实现了支持 ω-正规性质符号化模型检验工具 ENuSMV。它目前支持 ETL_f 和 AFL 的模型检验。

8.2　进一步的工作

进一步的工作包括如下几个方面。

- **关于 ETL 公理化**：第 3 章中，主要给出的是针对使用非确定自动机连接子的 ETL 的公理逻辑系统。关于这类逻辑的公理化问题，包括如下几个方面的将来工作：
  1. 研究使用交错自动机做连接子的 ETL 的公理化。一个猜想是：将关于自动机连接子的公理/规则替换为交错自动机的迁移函数即可。但目前尚未得到证明。
  2. 尽管 finite、looping、repeating 是 ETL 中使用最多的自动机接收条件，但

是研究使用其他接收条件自动机做连接子的 ETL 的公理化问题有可能认识这些连接子新的性质。因此这是一个需要研究的内容。

3. 研究使用双向自动机做连接子的 ETL 的公理化问题。

- **关于 ETL 的符号化模型检验：**在第 5 章中，研究了 ATL_f、ATL_l 以及 ATL_r 的符号化模型检验系统。将来需要研究的工作有如下几个方面：
  1. 降低 ATL_r 的符号化模型检验算法的编码复杂度，开发需要位变元数目更少的的 tableau 构建方法（目前，一个猜测是存在位变元数目为 $\mathcal{O}(|\varphi| \times \log(|\varphi|))$ 的编码方式）。
  2. 开发使用双向自动机作为连接子的 ETL 的符号化模型检验算法。
- **关于线性 μ-演算的符号化模型检验：**关于这个问题需要改进的地方主要包扩以下两个方面:
  1. 降低符号化编码的复杂度。其主要的着眼点也是在于优化 NBW 求补算法的符号化表示。
  2. 寻找比 ν-范式更具一般的线性 μ-演算公式的高效符号化模型检验算法。
- **关于 ENuSMV 工具：**对于该符号化模型检验工具，希望在其后继版本中提供如下的扩展：
  1. 增加对 ν-范式线性 μ-演算的符号化模型检验的支持。
  2. 基于 ATL_r 模型检验算法，增加 SMV 模型间语言等价性检查的支持（允许变量别名映射）。
  3. 完成从 FL 到 AFL 的前端转换器，使得标准 PSL 的模型检验能够被支持。
  4. 增加对这些逻辑交互式检验的命令行支持。
- **其他工作：**另外一个可能的将来工作是研究一阶 ETL 的推理以及（受限的）模型检验问题，以期给出一种揉合 ETL 模型检验和定理证明的框架，以处理部分无穷系统。

致　谢

在本文即将完成之际，首先深切缅怀我的导师陈火旺 教授。从硕士阶段到博士阶段，陈老师一直是我的研究生导师。陈老师一生治学严谨、襟怀坦荡、造诣深厚、克己奉公。在我的印象里，陈老师是一位“严厉”的导师，他从做人、做事、做学问上，都对我们严格要求。但同时，他又对我们在工作、生活上关怀无微不至。我仍清楚的记得 2007 年上半年时，陈老师在病榻上耐心地为师兄修改论文的情形；我仍清楚的记得他用打点滴的手握笔的样子；我也仍然记得在他病危前一个月，当我向他报告最近的突破时，陈老师欣慰的笑容。今天，虽然陈老师永远的离开了我们，但是老师孜孜不倦、严肃认真的研究态度，将是我一生中永远珍藏的宝贵财富。同时，感谢我的师母祖沛南女士。祖老师乐观坚毅的品格，积极向上的生活态度，将永远值得我学习。

感谢我的导师王戟教授。是王老师手把手的教会了我逻辑、自动机以及模型检验的基础理论。王老师对问题有着独到深刻的见解，常常能够一针见血的指出问题的实质。感谢王老师让我自由的选择研究方向。在整个博士阶段，不论我做哪个具体的问题，王老师总是支持我，并且和我进行认真细致的讨论。读博的第三年，是我最艰苦的日子。我曾经因为做不出结果而一度沮丧，甚至产生过想要放弃的念头。而王老师从未对我丧失过信心，是他的鼓励使我坚持到了最后。我不知如何感谢王老师。我想，我现在能做的就是把研究做的更深入。

感谢王兵山教授。王老师深厚的理论功底，使我敬仰。我有幸聆听他讲授的《形式语言》、《计算机逻辑》、《范畴论》等课程，这些为我以后的研究打下了较为坚实的基础。王老师为人坦然、淡泊，待人热情、诚恳，工作认真、负责。这些，无不值得我学习。

感谢毛晓光教授。从本科起，我就在毛老师的带领下做课题。毛老师平易近人、待人随和、关心学生、治学严谨。我对软件理论的兴趣，是与毛老师最初的指导分不开的。

感谢李舟军教授。李老师在计算理论，尤其是在进程代数方面有着深厚的造诣。从硕士入学的第一年，我便有幸接受李老师的熏陶。从李老师那里，我学到了《形式语义》、《进程代数》等课程的基础。这些对我以后的研究起到了很大的帮助作用。

感谢 602 教研室的毛新军、谭庆平、王挺、罗宇、周会平、董威、李暾、文艳军等各位老师。感谢他们对我的耐心指导和热情的帮助。我在学业上的每一点进步都

是与他们的帮助分不开的。

感谢荔建琦、李梦君两位师兄。从入师门的第一天起，荔师兄严谨执着的科研态度就深深影响感染了我们。荔师兄在科研上脚踏实地、精益求精、一丝不苟，从来不幻想去走所谓的“捷径”。梦君师兄刻苦、努力，学术上涉猎广泛，是我的良师益友。我仍记得 03、04 两个暑假和梦君师兄、周倜师弟一起编写 SPVT 工具的情形。那段日子过的忙碌而充实，使我在学术上得到许多收获。

感谢颜炯、邓欣、颜跃进、于洋、陈波、王涛、张丽娟等师兄、师姐对我的帮助；感谢陈立前、张晓艳、陈耀东、魏登萍、余杰、李莎莎等同学对我的支持。

感谢马晓东、陈振邦两位室友。他们同时是我的同门师兄弟，感谢他们陪我一起走过这段求学历程。感谢一同入学的沈锐、徐建军、张明、吴彤、吴宏、王进、罗军、马卓、唐勇、赵宝康、王大伟、徐金波、董孟高、王攀峰、杜云飞、温俊、齐星云、李瑞、方兴等同学，多年来的朝夕相处，使我们建立了深厚的友谊。

感谢王昭飞同学。昭飞学习踏实认真，刻苦细致。是他与我一起编写、调试、改进 ENuSMV，没有他的帮助，这个工具是不可能顺利完成的。

感谢张圣栋、李仁见、陈杰、樊沛、董龙明、张羽丰、樊华、徐厚峰、黄海、沈思淇、侯苏宁等各位师兄弟。是他们陪伴我一起度过了过去的学习、工作的时光。

感谢胡慎信、汪长江、吉炳安、郑光辉、鲍资富、王新国等历任博士生队队干部的关怀。是他们默默的奉献，为我们营造了一个良好的学习氛围。

感谢我的祖父母，他们对我殷切的希望，一直是我工作、学习的动力；他们的慈爱，使我感受到温馨。感谢我的父母，他们含辛茹苦的养育，使我长大成人；他们的宽容、付出，教会了我做人的道理。感谢我的姑父、姑妈，他们对我的期望，一直是我前进的动力。

感谢我的女友郭欣女士。是她的宽容体贴，才使我的研究顺利完成。这么多年，我陪她上街的次数基本上可以数的过来。而她对此没有太多的怨言，对我抱以极大的支持和理解。我想，这么多年，我愧欠她的有太多太多。

感谢 Nir Piterman 和 Sven Schewe。感谢他们在 Büchi 自动机确定化问题上与我进行的探讨，特别是 Nir 对我文章逐字逐句所做的修改。感谢 Pierre Wolper 教授等人对 ENuSMV 的反馈以及对我的鼓励。

感谢毛紫阳老师提供的LaTeX模板。本文模板是在其基础上修改得到的。很难想象如果没有TeX帮助，这篇长达 200 多页的博士论文将如何完成。

参考文献

[1] C. A. R. Hoare. An Axiomatic Basis for Computer Programming. Communications of the ACM. Oct. 1969, **12**(10):576–583.

[2] G.A. Kildall. A Unified Approach to Global Program Optimization. ACM Annual Symposium on Prociple of Programming Languages. The ACM Press, 194–206.

[3] J.B. Kam, J.D. Ullman. Monotone Data Flow Analysis Frameworks. Acta Information. 1977, **7**:305–317.

[4] P. Cousot, R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. ACM Annual Symposium on Principles of Programming Languages. ACM Press, 1977, 238–252.

[5] F. E. Allen, J. A. Cocke. A Program Data-Flow Analysis Procedure. Communic Ations of the ACM. 1976, **19**(3):137–174.

[6] E.M. Clarke, E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic. D. Kozen, (Editor) Logic of Programs. Springer-Verlag, 1982, vol. 131 of Lecture Notes in Computer Science, 52–71.

[7] J.P. Queille, J. Sifakis. Specification and Verification of Concurrent Systems in CESAR. M. Dezani-Ciancaglini, U. Montanari, (Editors) International Symposium on Programming. Springer-Verlag, 1982, vol. 137 of Lecture Notes in Computer Science, 216–230.

[8] E.M. Clarke. The Birth of Model Checking. Springer-Verlag, 2008, vol. 5000 of Lecture Notes in Computer Science, 1–26.

[9] L. Lamport. Proving the Correctness of Multiprocess Programs. IEEE Trans on Soft Eng. 1977, **SE–7**.

[10] M.C. Browne, E.M. Clarke, O. Grumberg. Reasoing about Networks with Many Identical Finite-state Process. Information and Computation. 1989, **81**(1):13–31.

[11] S. Berezin. Model Checking and Theorem Proving: A Unified Framework. Phd thesis, Carnegie Mellon University, Pittsburgh, PA, USA, Jan. 2002.

[12] N. Piterman. Verification of Infinite-State Systems. Phd. thesis, The Weizmann Institute of Science, Rehovot, Israel, Oct 2004.

[13] K. Apt, D. Kozen. Limits for Automatic Verification of Finite-state Systems. Information Processing Letters. 1986, **15**:307–309.

[14] I. Suzuki. Proving Properties of A Ring of Finite-State Machines. Information

Processing Letters. 1999, **28**:213–214.

[15] A. Pnueli. The Temporal Logic of Programs. Proc. of 18th IEEE Symposium on Foundation of Computer Science (FOCS' 77). IEEE Computer Society, 1977, 46–57.

[16] E.M. Clarke, O. Grumberg, K. Hamaguchi. Another Look at LTL Model Checking. CAV'94. Springer-Verlag, 1994, vol. 818 of Lecture Notes in Computer Science, 415–427.

[17] B. Boigelot, A. Legay, P. Wolper. Omega-Regular Model Checking. Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, TACAS 2004. Barcelona, Spain: Springer-Verlag, 2004, vol. 2988 of Lecture Notes in Computer Science, 561–575.

[18] R. E. Bryant. Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers. 1986, **C-35**(8):677–691.

[19] E. A. Emerson, E. M. Clarke. Characterizing Correctness Properties of Parallel Programs Using Fixpoints. Proc. of the 7th Int. Colloquium on Automata, Languages and Programming (ICALP'80). Springer-Verlag, 1980, vol. 85 of Lecture Notes in Computer Science, 169–181.

[20] Alessandro Cimatti Armin Biere, E. M. Clarke, Y. Zhu. Symbolic Model Checking without BDDs. Proceedings of the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Springer-Verlag, 1999, vol. 1579 of Lecture Notes in Computer Science, 193–207.

[21] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi. On the Temporal Analysis of Fairness. Proc, 7th ACM Symp. on Principles of Programming Languages (POPL'80). 1880, 163–173.

[22] M. Y. Vardi. Branching vs. Linear Time: Final Showdown. TACAS'01. Springer, 2001, vol. 2031 of Lecture Notes in Computer Science, 1–22.

[23] P. Wolper. Temporal Logic Can Be More Expressive. Information and Control. 1983, **56**(1–2):72–99.

[24] A. P. Sistla, M. Y. Vardi, P. Wolper. The Complementation Problem for Büchi Automata with Applications to Temporal Logic. Proc. 10th Int. Colloquium on Automata, Languages and Programming. Nafplion: Springer-Verlag, 1985, vol. 194 of Lecture Notes in Computer Science, 465–474.

[25] A. P. Sistla, M. Y. Vardi, P. Wolper. The Complementation Problem for Büchi Automata with Applications to Temporal Logic. Theoretical Computer Science. 1987, **49**:217–237.

[26] A. Pnueli. Linear and Branching Structures in the Semantics and Logics of Reactive Systems. W. Brauer, (Editor) International Colloquium on Automata, Language and Programming. Springer-Verlag, 1985, vol. 194 of Lecture Notes in Computer Science, 15–32.

[27] O. Lichtenstein, A. Pnueli, L. Zuck. The Glory of the Past. Workshop on Logics of Programs. Brooklyn: Springer-Verlag, 1985, vol. 193 of Lecture Notes in Computer Science, 97–107.

[28] B. Boigelot, A. Legay, P. Wolper. Iterating Transducers in the Large. Proc. 15th Int. Conf. on Computer Aided Verification. Boulder: Springer-Verlag, 2003, vol. 2725 of Lecture Notes in Computer Science, 223–235.

[29] B. Banieqbal, H. Barringer. Temporal logic with fixed points. B. Banieqbal, H. Barringer, and A. Pnueli, editors, Temporal Logic in Specification. Springer-Verlag, 1987, vol. 398 of Lecture Notes in Computer Science, 62–74.

[30] M. Y. Vardi, P. Wolper. Reasoning about Infinite Computations. Information and Computation. November 1994, **115**(1):1–37.

[31] Accellera. Accellera Property Languages Reference Manual. http://www.eda.org/vfv/docs/PSL-v1.1.pdf, June 2004.

[32] M. Lange. Linear Time Logics Around PSL: Complexity, Expressiveness, and A Little Bit of Succinctness. L. Caires, V.T. Vasconcelos, (Editors) CONCUR'07. Springer-Verlag, 2007, vol. 4703 of Lecture Notes in Computer Science, 90–104.

[33] J.R. Büchi. On A Decision Method in Restricted Second Order Arithmetic. Proc. Int. Congr. Method and Philosophy of Science 1960. Palo Alto, CA, USA: Stanford University Press, 1962, 1–12.

[34] I. Walukiewicz. Completeness of Kozen's Axiomatization of the Propositional μ-Calculus. Information and Computation. 2000, **157**:142–182.

[35] M. Lange, C. Stirling. Focus Games for Satisfiability and Completeness of Temporal Logic. Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, LICS'01. Boston, MA, USA: IEEE Computer Society Press, 2001.

[36] I. Beer, S. Ben-David, C. Eisner, A. Landver. RuleBase: an Industry-Oriented Formal Verification Tool. 33rd Design Automation Conference. ACM Press, 1996, 655–660.

[37] P. Wolper, M. Y. Vardi, A. P. Sistla. Reasoning about Infinite Computation Paths. Proc. 24th IEEE Symposium on Foundations of Computer Science. Tucson, 1983, 185–194.

[38] G.J. Holzmann. The Model Checker SPIN. IEEE Transactions on Software Engineering. 1997, **23**(5):279–295. Special Issue on Methods in Software Practice.

[39] J. Katoen. Concepts, Algorithms, and Tools for Model Checking. Lehrstuhl für Informatik VII Friedrich-Alexander Universität Erlangen-Nürnberg: FAU, 1998, 2 edn. Lecture Notes of the Course "Mechanised Validation of Parallel Systems".

[40] K. L. McMillan. Symbolic Model Checking, An Approach to the State Explosion Problem. Ph.D. thesis, Carnegie Mellon University, Kluwer Academic Publishers, 1993.

[41] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, L.J. Hwang. Symbolic Model Checking 10^20 States and Beyond. Information and Computation. 1992, **98**(2):142–170.

[42] E. M. Clarke, O. Grumberg, D. A. Peled. Model Checking. Longdon, England: The MIT Press, 1999.

[43] 苏开乐, 吕关锋, 洛翔宇. CTL* 的符号化模型检测. 计算机学报. Nov. 2005, **28**(11):1798–1806.

[44] A. P. Sistla, E. M. Clarke. The complexity of Propositional Linear Temporal Logics. Journal of Assoc Comput Mach. 1985, **32**(3):733–749.

[45] W. Zhang. Model Checking with SAT-based Characterization of ACTL formulas. ICFEM'07. Lecture Notes in Computer Science, Springer-Verlag, 2007, 191–211.

[46] W. Zhang. Verification of ACTL Properties by Bounded Model Checking. EUROCAST'07. 2007, 556–563.

[47] E. A. Emerson, J. Y. Harpern. Decision Procedures and Expressiveness in The Temporal Logics of Branching Time. Journal of the ACM. 1986, **33**(1):151–178.

[48] A. Pnueli, Y. Kesten. A Deductive Proof System for CTL*. L. Brim, P. Jancar, M. Kretínský, A. Kucera, (Editors) CONCUR02. Springer, 2002, vol. 2421 of Lecture Notes in Computer Science, 24–40.

[49] M. Reynolds. An axiomatization of PCTL*. Information and Computation. July 2005, **201**:72–119.

[50] A.P. Sistla. Theoretical Issues in The Design and Verification of Distributed Systems. Ph.d. thesis, CMU Dept. of Computer Science, 1983.

[51] Y. Kesten, A. Pnueli. Complete Proof System for QPTL. J Log Comput. 2002, **12**(5):701–745.

[52] R. Kaivola. Axiomatising Linear Time Mu-calculus. I. Lee, S. A. Smolka, (Editors) CONCUR '95. Springer, 1995, vol. 962 of Lecture Notes in Computer Science, 423–

437.

[53] R. Kaivola. Using Automata to Characterise Fixed Point Temporal Logics. Ph.D. thesis, University of Edinburgh, 1997.

[54] C. Dax, M. Hofmann, M. Lange. A Proof System for the Linear Time μ-Calculus. S. Arun-Kumar, N. Garg, (Editors) FSTTCS06. Springer-Verlag, 2006, vol. 4337 of Lecture Notes in Computer Science, 273 – 284.

[55] D. Kozen. Results on the Propositional μ-calculus. Theoretical Computer Science. 1983, **27**:333–354.

[56] I. Walukiewicz. Completeness of Kozen's Axiomatization of the Propositional μ-Calculus. Proceedings 10th Annual IEEE Symp. on Logic in Computer Science, LICS'95, Los Alamitos, CA: IEEE Computer Society Press, 1995. 14–24.

[57] O. Grumberg, D. E. Long. Model Checking and Modular Verification. 1996, **16**:843–872.

[58] A. Legay, P. Wolper. On the Use of Automata-based Techniques in Symbolic Model Checking. Electronic Notes in Theoretical Computer Science. 2006, **150**(1).

[59] A. Legay. T(O)RMC: A Tool for (ω)-Regular Model Checking. A. Jupta, S. Malik, (Editors) The 20th International Conference on Computer Aided Verification (CAV'08). Princeton, N,J, USA: Springer, 2008, vol. 5123 of Lecture Notes in Computer Science, 548–551.

[60] S. Ben-David, D. Fisman, S. Ruah. Embedding Finite Automata within Regular Expressions. Theoretical Computer Science. Sep 2008, **404**:202–218.

[61] A. Pnueli, A. Zaks. PSL Model Checking and Run-time Verification via Testers. J. Misra T. Nipkow, E. Sekerinski, (Editors) Formal Methods (2006). Springer-Verlag, vol. 4085 of Lecture Notes in Computer Science, 573–586.

[62] Y. Kesten, O. Maler, M. Marcus, A. Pnueli, E. Shahar. Symbolic Model Checking with Rich Assertional Languages. Theoretical Computer Science. 2001, **256**(1-2):93–112.

[63] A. Bouajjani, B. Jonsson, M. Nilsson, T. Touili1. Regular Model Checking. E. A. Emerson, A. P. Sistla, (Editors) Computer Aided Verification (CAV'00). 2000, vol. 1855 of Lecture Notes in Computer Science, 403–418.

[64] D. E. Long, A. Browne, E. M. Clarke, S. Jha, W. R. Marrero. An Imprived Algorithm for the Evaluation of Fixpoint Expressions. D. L. Dill, (Editor) CAV'94. Springer, 1994, vol. 818 of Lecture Notes in Computer Science, 338–350.

[65] T. Wilke. Alternating Tree Automata, Parity Games, and Modal μ-Calculus. Bull,

Belg, Math, Soc. 2002, **8**(2):359–391.

[66] E. A. Emerson, C. L. Lei. Efficient Model Checking in Fragments of the Propositional Mu-Calculus. First IEEE Symposium on Logic in Computer Science. Los Alamitos: IEEE Computer Society, 1986, 267–278.

[67] S. Schewe. Synthesis of Distributed Systems. Phd thesis, Saarbrücken, 2008.

[68] R. Cleaveland. Tableau-based Model Checking in the Propositional Mu-Calculus. Acta Informatica. 1990, **27**(8):725–748.

[69] E. A. Emerson, C. S. Jutla, A. P. Sistla. On Model-checking for Fragments of μ-calculus. International Conference on Computer-Aided Verification (CAV'93). Springer-Verlag, 1993, vol. 697 of Lecture Notes in Computer Science, 385–396.

[70] B.A. Trakhtenbrot. Finite Automata and The Logic of One-Place Predication. Sibirian Mathnatical Journal. 1962, **3**(2):102–131.

[71] R. McNaughton. Testing and Generating Infinite Sequences by A Finite Automaton. Information and Computation. 1966, **9**:521–530.

[72] M.O. Rabin. Decidability of Second Order Theories and Automata on Infinite Trees. Transaction of the AMS. 1969, **141**:1–35.

[73] C. Löding. Methods for the Transformation of ω-Automata: Complexity and Connection to Second-order Logic. Master's thesis, Christian-Albrechts-University of Kiel, 1998.

[74] D.E. Muller, P.E. Schupp. Alternating Automata on Infinite Trees. Theoretical Computer Science. 1987, **54**:267–276.

[75] Y. Choueka. Theories of automata on ω-tapes. Journal of Computer and System Sciences. 1974, **8**:117–141.

[76] R. M. Burstall. Program Proving as Hand Simulation with A Little Induction. IFIP Congress. 1974, 308–312.

[77] F. Kröger. Lar: A Logic of Algorithmic Reasoning. Acta Inf. 1977, **8**:243–266.

[78] E. M. Clarke, B. Schlingloff. Model Checking. A. Robinson, A. Voronkov, (Editors) Handbook of Automated Reasoning, MIT and Elsevier Science Publishers, 2001, vol. 2. 1369–1522.

[79] H. W. Kamp. Tense Logic and the Theory of Linear Order. Ph.d. thesis, Univ. of Calif, Los Angeles, 1968.

[80] E.M. Clarke, I.A. Draghicescu. Expressibility Results for Linear-time and Branching Time Logics. J.W. de Bakker, W.P. de Roever, G. Rozenberg, (Editors) Proc.Workshop on Linear Time, Branching Time, and Partial Order in Logics and

Models for Concurrency. Springer-Verlag, 1988, vol. 354 of Lecture Notes in Computer Science, 428–437.

[81] L. Lamport. Sometimes is Sometimes "Not Never"– on the Temporal Logic of Programs. Proc. 7th ACM Symp. on Principle of Programming Languages. ACM Press, 1980, 174–185.

[82] M. Maidl. The common fragment of CTL and LTL. Foundations of Computer Science. 2000, 643–652.

[83] M. Bojańczyk. The Common Fragment of ACTL and LTL. R. Amadio, (Editor) FOSSACS'08. Springer-Verlag, 2008, vol. 4962 of Lecture Notes in Computer Science, 172–185.

[84] A. N. Prior. Time and Modality. Oxford University Press, 1957.

[85] A. N. Prior. Past, Present and Future. Clarendon Press, 1967.

[86] D. Gabbay. The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems. B. Banieqbal, (Editor) Temporal Logic in Specification. Springer-Verlag, 1989, vol. 398 of Lecture Notes in Computer Science, 431–448.

[87] Z. Wu. On the Expressive Power of QLTL. Proc. 4th International Colloquium on Theoretical Aspects of Computing (ICTAC'07). Macau, China: Springer-Verlag, 2007, vol. 4711 of Lecture Notes in Computer Science, 337–341.

[88] V. Pratt. A Decidable μ-Calculus. ACM Symposium on Foundations of Computer Science (FOCS81). ACM Press, 1981, 75–84.

[89] The Propositional Mu-Calculus is Elementary. J. Paredaens, (Editor) 11th International Colloquium on Automata, Languages and Program. Springer-Verlag, 1984, vol. 172 of Lecture Notes in Computer Science, 465–472.

[90] P. A. Bonatti, C. Lutz, A. Murano, M. Y. Vardi. The Complexity of Enriched μ-Calculi. ICALP 2006. Springer, 2006, vol. 4052 of Lecture Notes in Computer Science, 540–551.

[91] M.Y. Vardi. A Temporal Fixpoint Calculus. Proc. of 15th ACM symposium on Principles of Programming Languages (FOCS'88). San Diego, California, 1988, 250–259.

[92] O. Kupferman, N. Piterman, M. Y. Vardi. Extended Temporal Logic Revisited. Proc. 12th International Conference on Concurrency Theory. Denmark: Springer, 2001, vol. 2154 of Lecture Notes in Computer Science, 519–535.

[93] I. Beer, S. Ben-David, C. Eisner, D. Fisman, A. Gringauze, Y. Rodeh. The Temporal Logic Sugar. G. Berry, H. Comon, A. Frinkel, (Editors) 13th International

Conference on Computer Aided Verification. London, UK: Springer-Verlag, 2001, vol. 2102 of Lecture Notes in Computer Science, 363–367.

[94] R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M. Y. Vardi, Y. Zbar. The ForSpec Temporal Logic: A New Temporal Property-Specification Language. TACAS'02. Springer, 2002, vol. 2280 of Lecture Notes in Computer Science, 296–311.

[95] M. Fujita, H. Fujisawa, N. Kawato. Evaluation and Improveness of Boolean Comparison Method Based on Binary Decision Diagrams. IEEE International Conference on Computer Aided Design. IEEE Society Press, 1988, 10–13.

[96] S. Malik, A. Wang, R. Brayton, A. Saniovanni-Vincenteli. Logic Verification using Binary Decision Diagrams in A Logic Synthesis Environment. International Conference on Computer-Aided Design. IEEE Society Press, 1988, 6–9.

[97] A. Tarski. A Lattice-Theoretical Fixpoint Theorem and Its Applications. Pacific Journal of Mathematics. 1955, **5**(2):285–309.

[98] E.M. Clarke, O. Grumberg, D. Long. Verification Tools for Finite-State Concurrent Systems. REX School/Symposium Proceedings. Springer-Verlag, 1993, vol. 803 of Lecture Notes in Computer Science, 124–175.

[99] M. Kaminski. A Classification of ω-Regular Languages. Theoretical Computer Science. 1985, **36**:217–229.

[100] L. Landweber. Decision Problems for ω-Automata. MST. 1969, **3**:374–385.

[101] T. Moriya, H. Yamusaki. Accepting Conditions for Automata on ω-Languages. Theoretical Computer Science. 1988, **61**:137–147.

[102] L. Staiger. Research in the Theory of ω-Languages. Electron Inf Verarbeit Kybernetic. 1987, **23**:415–439.

[103] W. Thomas. Handbook of Theoretical Computer Science, Elsevier, 1990, vol. B. 135–191.

[104] K. Wagner. On ω-Regular Sets. Information and Control. 1979, **43**:123–177.

[105] I. Walukiewicz. On Completeness of the μ-calculus. LICS'93. 1993, 136–146.

[106] M. J. Fischer, R. E. Ladner. Propositional Dynamic Logic of Regular Programs. International Journal of Computer and System Science. 1979, **18**(7):194–211.

[107] Y. Gurevich, L. Harrington. Trees, Automata, and Games. Proceeding of 14th ACM Symposium on the Theory of Computing. San Francisco, California, 1982, 60–65.

[108] E. A. Emerson, C. S. Jutla. Tree Automata, Mu-calculus, and Determinacy. 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'91). 1991,

368–377.

[109] D. Janin, I. Walukiewicz. Automata for the Modal μ-Calculus and related Results. 20th International Symposium of Mathematical Foundations of Computer Science (MFCS'95). Prague: Springer, 1995, vol. 969 of Lecture Notes in Computer Science, 552–562.

[110] O. Grumberg, E. Clarke, K. Hamaguchi. Another Look at LTL Model Checking. Formal Methods in System Design. February 1997, **10**(1):57–71.

[111] A. Cimatti, E. Clarke, F. Giunchiglia1, M. Roveri. NuSMV: A New Symbolic Model Verifier. CAV'99. Springer-Verlag, 1999, vol. 1633 of Lecture Notes in Computer Science, 495–499.

[112] Q. Yan. Lower Bounds for Complementation of ω-Automata via the Full Automata Technique. ICALP'06. Springer-Verlag, 2006, vol. 4052 of LNCS, 589–600.

[113] Q. Yan. Lower Bounds for Complementation of ω-Automata via the Full Automata Technique. Journal of Logical Methods in Computer Science. 2008, **4**(1:5).

[114] O. Kupferman, M.Y. Vardi. Weak Alternating Automata Are Not That Weak. ACM TCL. 2001, **2**(2):408–429.

[115] E. Friedgut, O. Kupferman, M.Y. Vardi. Büchi Complementation Made Tighter. ATVA' 06. Springer-Verlag, 2004, vol. 3299 of LNCS, 64–78.

[116] W. Liu, J. Wang, H. Chen, X. Ma. Symbolic Model Checking APSL. The 2nd IEEE Symposium on Theoretical Aspects of Software Engineering. IEEE Soc., 2008, 39–46.

[117] W.Liu, J. Wang, H.Chen, X. Ma, Z. Wang. Symbolic Model Checking APSL. Frontiers of Computer Science in China. 2009, **3**(1):130–141.

[118] D. Bustan, D. Fisman, J. Havlicek. Automata Construction for PSL. Tech. Rep. MCS05-04, IBM Haifa Research Lab, May 2005.

[119] S. Schewe. Büchi Complementation Made Tight. STACS 2009. IBFI, 2009, 661–672.

[120] S. Miyano, T. Hayashi. Alternating Finite Automata on ω-Words. Theoretical Computer Science. 1984, **32**:321–330.

[121] S. Gurumurthy, O. Kupferman, F. Somenzi, M.Y. Vardi. On Complementing Nondeterministic Büchi Automata. 12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods. Lecture Notes in Computer Science, Springer-Verlag, 2003.

[122] O. Kupferman, M.Y. Vardi. Complementation Constructions for Nondeterministic Automata on Infinite Words. 11th International Conference on Tools and algorithms

for the construction and analysis of systems. Lecture Notes in Computer Science, Springer-Verlag, 2004.

[123] T. Ball, S.K. Rajamami. The SLAM Project: Debugging System Software via Static Analysis. ACM Press, 2002, 1–3.

[124] K. Havelund, T. Pressburger. Model Checking Java Programs using Java PathFinder. STTT. 2000, **2**(4):366–381.

[125] D. Beyer, T. A. Henzinger, R. Jhala. The Software Model Checker BLAST. Int Journal of Software Tools Technol Transfer. 2007, **9**:505–525.

[126] E.M. Clarke, A. Groce, S. Jha, H. Veith. Modular Verification of Software Components in C. IEEE Soc., 2003, 385–395.

[127] E.M. Clarke, D. Kroening, F. Lerda. A Tool for Checking ANSI-C Programs. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04). Springer-Verlag, 2004, vol. 2988 of Lecture Notes in Computer Science, 168–176.

[128] R. Cavada, A. Cimatti, C. A. Jochim, G. Keighren, E. Olivetti, M. Pistore, M. Roveri, A. Tchaltsev. NuSMV 2.4 User Manual. http://nusmv.fbk.eu/NuSMV/userman/v24/nusmv.pdf, Apr. 2007.

[129] J. Guttman, A. Herzog, J. D. Ramsdell. Slat: Information Flow Analysis in Security Enhanced Linux. http://nusmv.fbk.eu/NuSMV/userman/v24/nusmv.pdf, April 2005.

[130] J. Guttman, A. Herzog, J. D. Ramsdell. Information Flow in Operating Systems: Eager Formal Methods. IFIPWG 1.7Workshop on Issues in the Theory of Security. 2003.

[131] A. J. Martin. The Design of A Self-Timed Circuit for Distributed Mutual Exclusion. H. Fuchs, (Editor) Proceedings of the 1985 Chapel Hill Conference on Very Large Scale Integration. 1985.

作者在学期间取得的学术成果

[1] Wanwei Liu, Ji Wang, Huowang Chen, Xiaodong Ma, Zhaofei Wang. Symbolic Model Checking APSL. Frontiers of Computer Science in China, Springer-Higher Education Press, 2009, **3**(1):130–141.

[2] Wanwei Liu, Ji Wang. A Tighter Analysis of Piterman's Büchi Determinization. Information Processing Letters, Elsevier, 2009,**109**(16): 941–945.

[3] Wanwei Liu, Ji Wang, Wei Dong, Huowang Chen. Axiomatizing Extended Temporal Logic Fragments via Instantiation. In Proceedings of International Colloquium on Theoretical Aspects of Computing 2007, Springer-Verlag, 2007, vol. 4711 of Lecture Notes in Computer Science, 322–336.

[4] Wanwei Liu, Ji Wang, Huowang Chen, Xiaodong Ma. Symbolic Model Checking APSL. In Proceedings of IEEE and IFIP Symposium of Theoretical Aspects of Software Engineering 2008, IEEE Press, 2008, 39–46.

[5] Ji Wang, Xiaodong Ma, Wei Dong, Huofeng Xu, Wanwei Liu. Demand-Driven Memory Leak Detection Based on Flow and Context Sensitive Pointer Analysis. Journal of Computer Science and Technology, 2009, **24**(2): 347–356.

[6] 刘万伟, 王戟, 王昭飞. ETL 的符号化模型检验. 软件学报, 2009, **20**(8): 2015–2025.

[7] 刘万伟, 王戟, 陈火旺. 基于 Game 理论的 μ-演算公理化. 计算机研究与发展, 2007, **44**(11): 1896–1902.

[8] 刘万伟, 王戟, 陈火旺. 线性 μ-演算交换深度的可判定性及其复杂度. 计算机研究与发展(增刊), 2008, **45**(Suppl): 1–6.

[9] 刘万伟, 王戟, 陈火旺. 基于 LTL Tableau 的自动机构造. 吉林大学学报（工学版）, 2007, **37**(1): 132–135.

[10] 刘万伟, 周倜, 李梦君, 李舟军. 一种基于进程代数的安全协议验证消解算法. 计算机工程与科学, 2006. **28**(7): 14–17.

[11] 周倜, 李梦君, 刘万伟, 李舟军. 安全协议的进程代数规约到逻辑程序的自动转换. 计算机工程与科学. 2006. **28**(1): 22–25.

[12] 卓琳, 刘万伟, 谭庆平. 一种基于场景的性质验证方法. 计算机工程与科学, 2006. **28**(4): 56–59.

[13] 陈立前, 王戟, 刘万伟. 基于约束的多面体抽象域的弱结合. 软件学报（已录用）.

注：[1] 是 [4] 的 journal 扩展。