基于遗忘的反应式系统最弱充分条件研究

二〇二二年六月

姓名：冯仁艳
导师：王以松
联合导师：Erman Acar[1]
研究方向：软件工程技术与人工智能

[1]LIACS, Leiden University, The Netherlands

目录



图 1: 系统故障引起的系列灾难现场

表 1: 由系统故障引起的重大事件概览

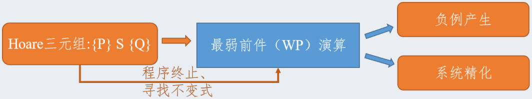
时间	事故原因	损失
1991 年	美国爱国者导弹系统舍入错误	28 名士兵死亡、100 人受伤等
1996 年	阿丽亚娜 5 火箭代码重用	火箭与其它卫星毁灭
1999 年	火星探测器用错度量单位	探测器坠毁并造成了 3.27 亿美元的损失
2011 年	温州 7.23 动车信号设备 在设计上存在严重的缺陷	动车脱节脱轨、多人失去生命

研究背景和意义：形式化验证为系统的正确提供了有力依据

自动定理证明（Automated theorem proving）

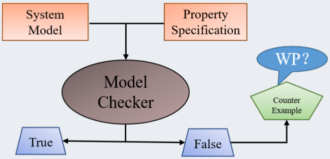
令 ϕ_{imp} 和 ϕ_{spec} 分别表示系统模型和规范对应的时序逻辑公式：

- $\phi_{imp} \rightarrow \phi_{spec}$， 或
- $\phi_{imp} \leftrightarrow \phi_{spec}$。



模型检测（Model Checking）

- $\mathscr{M} \models^{?} \phi_{spec}$.
- 反应式系统 (reactive system)：是指与环境有着持续不断交互的系统。
- 如何计算反应式系统的 WP?



例 1 (汽车制造企业模型)

一个汽车制造企业能够生产两种汽车：小轿车 (se) 和跑车 (sp)。每隔一段时间，该企业都会做一个生产决策 (d)，即：合理的生产计划。刚开始的时候，该企业做出了具有三个选择 (s) 的方案：

(1) 先生产足够的 se，然后在再生产 sp;

(2) 先生产足够的 sp，然后再生产 se;

(3) 同时生产 se 和 sp。

这一过程可以由图 2中的 Kripke 结构（带标签的状态转换图）$\mathscr{M} = (S, R, L)$ 形式化地展现出来，其中:

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

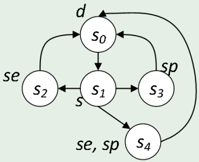


图 2: 汽车制造企业模型

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

研究背景和意义：知识表示与推理（KR）中的 SNC 和 WSC

最强必要条件（SNC）和最弱充分条件（WSC）

SNC 和 WSC 分别用于描述给定理论下的最一般的结果（consequence）和最一般的诱因（abduction）[8]。满足下面两个条件的 φ 称为 q 在理论 Σ 下的 SNC：

(1) $\Sigma \models q \rightarrow \varphi$;

(2) 对任意 φ' 且 $\Sigma \models q \rightarrow \varphi'$，有 $\Sigma \models \varphi \rightarrow \varphi'$。

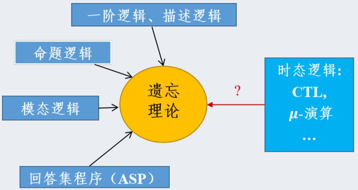
满足下面两个条件的 ψ 称为 q 在理论 Σ 下的（WSC）：

(1) $\Sigma \models \psi \rightarrow q$;

(2) 对任意 ψ' 且 $\Sigma \models \psi' \rightarrow q$，有 $\Sigma \models \psi' \rightarrow \psi$。

遗忘理论（Forgetting）

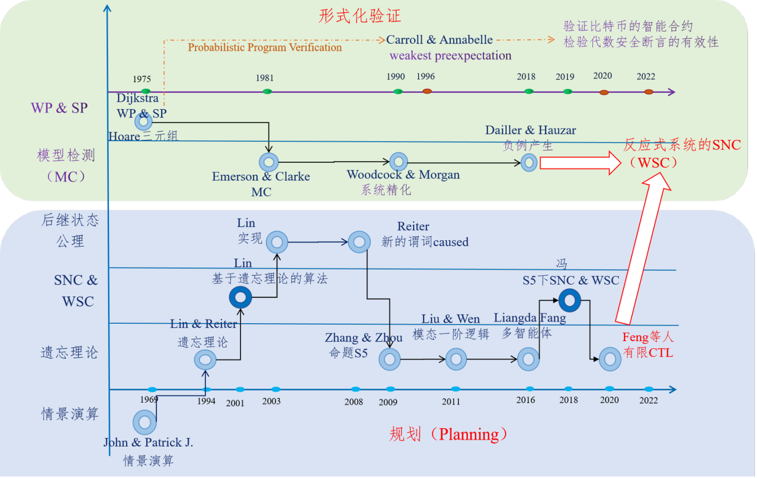
遗忘是一种从理论中抽取知识的技术 [9]，被用于规划[5, 7] 和知识更新 中 [13]。非形式化地，对于逻辑语言 L 中的任意公式和原子集合，如果从该公式中遗忘掉该原子集合后得到的结果仍然在 L 中，则称遗忘存在，同时也称该公式和原子集合的遗忘存在。

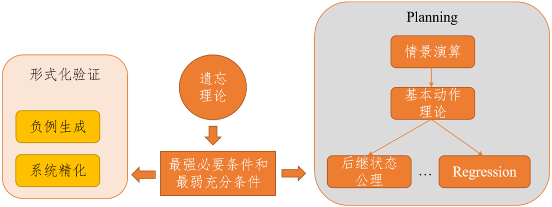


- CTL（Computation tree logic）：计算树逻辑，是一种分支时序逻辑
  - 其模型检测（MC）问题能在多项时间内完成；
  - 能很好的表达系统要求的安全属性（Safety properties）、活性属性（Liveness properties)、持续属性（Persistence properties）和公平属性（Fairness properties）。
- μ-演算（μ-calculus）：是其他形式体系的机械基础
  - LTL、CTL、L_w 等时态逻辑都能用 μ-演算表示；
  - S1S 表达能力严格不如 μ-演算；
  - μ-演算与 S2S 有相同的表达能力；
  - ……

形成时序逻辑系统遗忘理论的框架（verification），架起形式化验证和知识表示与推理（KR）的桥梁。

国内外研究现状


形式化验证
Probabilistic Program Verification
Carroll & Annabelle
weakest preexpectation
验证比特币的智能合约
检验代数安全断言的有效性
1975
1981
1990
1996
2018
2019
2020
2022
WP & SP
Dijkstra
WP & SP
Hoare三元组
模型检测
（MC）
Emerson & Clarke
MC
Woodcock & Morgan
系统精化
Dailler & Hauzar
负例产生
反应式系统的SNC
（WSC）
后继状态
公理
Lin
实现
Reiter
新的谓词caused
SNC &
WSC
Lin
基于遗忘理论的算法
冯
S5下SNC & WSC
Lin & Reiter
遗忘理论
遗忘理论
Zhang & Zhou
命题S5
Liu & Wen
模态一阶逻辑
Liangda Fang
多智能体
Feng等人
有限CTL
1969
1994
2001
2003
2008
2009
2011
2016
2018
2020
2022
情景演算
John & Patrick J.
情景演算
规划（Planning）



通过人工智能的知识表示与推理 (KR) 技术，从遗忘理论出发，研究反应式系统(在某个符号集上)SNC 和 WSC 的表示与计算，提高反应式系统的可靠性 (或辅助证明反应式系统的正确性)。

- CTL 和 μ-演算的遗忘理论
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
- 计算 *CTL* 遗忘的计算方法

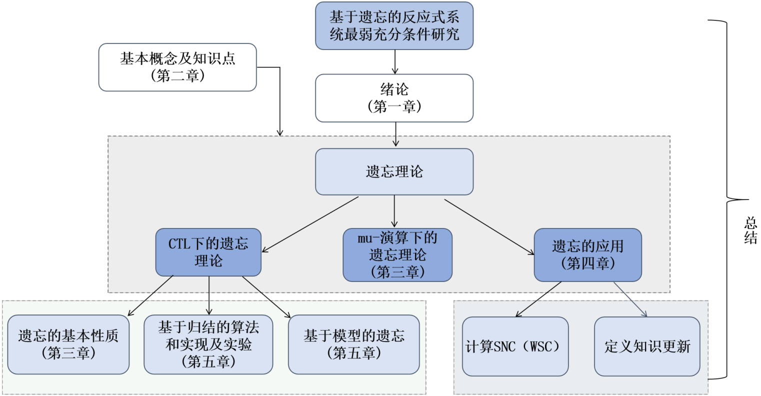


图 3: 文章组织结构示意图

拟解决的关键科学问题

- CTL 的遗忘什么情形下存在？（CTL 不具有 uniform interpolation 性质）
- 遗忘理论与反应式系统的 SNC 和 WSC 的关系
  - 反应式系统不终止
  - 遗忘理论的作用对象是公式
- *CTL* 和 μ-演算的遗忘在推理问题上的复杂性

目录

Kripke 结构

$\mathscr{A}$：原子命题的集合　　　　Ind：索引的集合

定义 2 (初始 Ind-Kripke 结构)

一个初始 Ind-Kripke 结构是一个五元组 $\mathscr{M}=(S,R,L,[\_],s_0)$，其中：

- S 是状态的非空集合，s_0 是 $\mathscr{M}$ 的初始状态（参见下文）；
- $R\subseteq S\times S$ 是状态转换函数，且对任意 $s\in S$，存在 $s'\in S$ 使得 $(s,s')\in R$；
- $L:S\to 2^{\mathscr{A}}$ 是一个标签函数；
- $[\_]:\mathrm{Ind}\to 2^{S\times S}$ 是一个函数，其使得对任意 $ind\in\mathrm{Ind}$，若 $s\in S$，则存在唯一一个 $s\in S$ 使得 $(s,s')\in[ind]\cap R$。

相关概念

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

Kripke 结构

$\mathscr{A}$：原子命题的集合　　　Ind：索引的集合

定义 2 (初始 Ind-Kripke 结构)

一个初始 Ind-Kripke 结构是一个五元组 $\mathscr{M}=(S,R,L,[\_],s_0)$，其中：

- S 是状态的非空集合，s_0 是 $\mathscr{M}$ 的初始状态（参见下文）；
- $R\subseteq S\times S$ 是状态转换函数，且对任意 $s\in S$，存在 $s'\in S$ 使得 $(s,s')\in R$；
- $L:S\to 2^{\mathscr{A}}$ 是一个标签函数；
- $[\_]:\mathrm{Ind}\to 2^{S\times S}$ 是一个函数，其使得对任意 $ind\in\mathrm{Ind}$，若 $s\in S$，则存在唯一一个 $s\in S$ 使得 $(s,s')\in[ind]\cap R$。

相关概念

令 $\mathscr{M}=(S,R,L)$ 为 Kripke 结构，$\mathscr{M}'=(S,R,L,[\_])$ 为 Ind-Kripke 结构：

- 路径：$\mathscr{M}$ 上的路径是 $\mathscr{M}$ 上的状态构成的无限序列 $\pi=(s_0,s_1,s_2,\dots)$，且满足对任意 $j\geq 0$，$(s_j,s_{j+1})\in R$；
- $s'\in\pi$：表示 s' 是路径 π 上的一个状态；π_s：表示以 s 为起点的 $\mathscr{M}$ 上的一条路径；
- 初始状态：如果对任意 $s'\in S$，都存在路径 π_s 使得 $s'\in\pi_s$，那么称 s 为初始状态；
- 索引路径：$\mathscr{M}'$ 上的一条索引路径 $\pi_s^{\langle ind\rangle}$（$ind\in\mathrm{Ind}$）是一条路径 $(s_0(=s),\ s_1,\ s_2,\ \dots)$，且对任意 $j\geq 0$，有 $(s_j,s_{j+1})\in[ind]$。

$\mathscr{A}$：原子命题的集合　　　　Ind：索引的集合

定义 2 (初始 Ind-Kripke 结构)

一个初始 Ind-Kripke 结构是一个五元组 $\mathscr{M} = (S, R, L, [\_], s_0)$，其中：

- S 是状态的非空集合，s_0 是 $\mathscr{M}$ 的初始状态（参见下文）；
- $R \subseteq S \times S$ 是状态转换函数，且对任意 $s \in S$，存在 $s' \in S$ 使得 $(s, s') \in R$；
- $L : S \to 2^{\mathscr{A}}$ 是一个标签函数；
- $[\_] : \mathrm{Ind} \to 2^{S \times S}$ 是一个函数，其使得对任意 $ind \in \mathrm{Ind}$，若 $s \in S$，则存在唯一一个 $s \in S$ 使得 $(s, s') \in [ind] \cap R$。

相关概念

- (Ind-) 结构：初始 (Ind-)Kripke 结构 $\mathscr{M}$ 和是 $\mathscr{M}$ 中的状态 s 构成的二元组 $\mathscr{K} = (\mathscr{M}, s)$；
- 初始 (Ind-) 结构：(Ind-) 结构 $\mathscr{K} = (\mathscr{M}, s)$ 中 s 为初始状态的情形。

CTL 的语法

CTL 的语言符号

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

定义 3 (带索引的 CTL)

带索引的 CTL 公式的存在范式 (existential normal form, ENF)可以用巴科斯范式递归定义如下：

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

其中，$p \in \mathscr{A}$，$ind \in \mathrm{Ind}$。

没有索引和 **start** 的公式称为 CTL 公式。

定义 3 (带索引的 CTL)

带索引的 CTL 公式的存在范式 (existential normal form, ENF)可以用巴科斯范式递归定义如下：

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

其中，$p \in \mathscr{A}$，$ind \in \mathrm{Ind}$。
没有索引和 **start** 的公式称为 CTL 公式。

CTL 中其它形式的公式可以通过如下定义（使用上述定义中的形式）得到：

$$\varphi \wedge \psi \stackrel{def}{=} \neg(\neg\varphi \vee \neg\psi) \tag{1}$$

$$\varphi \to \psi \stackrel{def}{=} \neg\varphi \vee \psi \tag{2}$$

$$\mathrm{A}(\varphi \mathrm{U} \psi) \stackrel{def}{=} \neg\mathrm{E}(\neg\psi \mathrm{U}(\neg\varphi \wedge \neg\psi)) \wedge \neg\mathrm{EG}\neg\psi \tag{3}$$

$$\mathrm{A}(\varphi \mathrm{W} \psi) \stackrel{def}{=} \neg\mathrm{E}((\varphi \wedge \neg\psi)\mathrm{U}(\neg\varphi \wedge \neg\psi)) \tag{4}$$

$$\mathrm{E}(\varphi \mathrm{W} \psi) \stackrel{def}{=} \neg\mathrm{A}((\varphi \wedge \neg\psi)\mathrm{U}(\neg\varphi \wedge \neg\psi)) \tag{5}$$

$$\mathrm{AF}\varphi \stackrel{def}{=} \mathrm{A}(\top \mathrm{U} \psi) \tag{6}$$

$$\mathrm{EF}\varphi \stackrel{def}{=} \mathrm{E}(\top \mathrm{U} \psi) \tag{7}$$

$$\mathrm{AX}\varphi \stackrel{def}{=} \neg\mathrm{EX}\neg\varphi \tag{8}$$

$$\mathrm{AG}\varphi \stackrel{def}{=} \neg\mathrm{EF}\neg\varphi \tag{9}$$

定义 3 (带索引的 CTL)

带索引的 CTL 公式的存在范式 (existential normal form, ENF)可以用巴科斯范式递归定义如下：

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

其中，$p\in\mathscr{A}$，$ind\in\mathrm{Ind}$。
没有索引和 **start** 的公式称为 CTL 公式。

符号优先级

带索引的 CTL 中各类符号的优先级如下，且从左到右优先级逐渐降低：

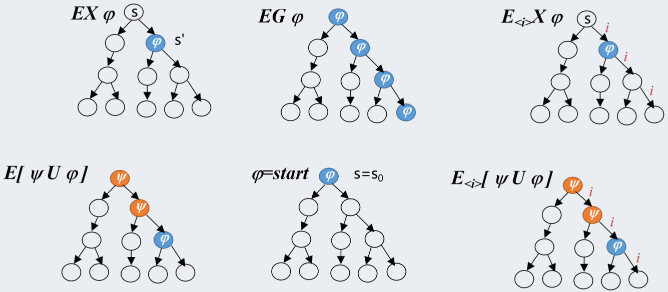
$$\neg, \mathrm{EX}, \mathrm{EF}, \mathrm{EG}, \mathrm{AX}, \mathrm{AF}, \mathrm{AG}, \mathrm{E}_{\langle ind\rangle}\mathrm{X}, \mathrm{E}_{\langle ind\rangle}\mathrm{F}, \mathrm{E}_{\langle ind\rangle}\mathrm{G}, \wedge, \vee, \mathrm{EU}, \mathrm{AU}, \mathrm{EW}, \mathrm{AW}, \mathrm{E}_{\langle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

此外，给定一个不包含“$\rightarrow$”的公式 φ 和原子命题 p，

- 若 p 的前面有偶数个否定 $\neg$，则称 p 在 φ 中的出现为正出现，否则为负出现。
- 若 φ 中所有 p 的出现都为正出现（或负出现），则称 φ 关于 p 是正的（或负的）。

定义 4 (带索引的 CTL 的语义)

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



CTL 的语义

记号

令 φ、φ_1 和 φ_2 为公式，这里列出文中出现的一些记号及其含义。

- 模型：满足公式 φ 的初始 Ind-结构称为 φ 的一个模型；
- $Mod(\varphi)$：公式 φ 的所有模型构成的集合；
- 可满足：如果 $Mod(\varphi) \neq \emptyset$，则称 φ 是可满足的；
- 逻辑蕴涵：若 $Mod(\varphi_1) \subseteq Mod(\varphi_2)$，则称 φ_1逻辑地蕴涵φ_2，记为 $\varphi_1 \models \varphi_2$；
- 逻辑等值：当 $\varphi_1 \models \varphi_2$ 且 $\varphi_2 \models \varphi_1$ 时，即 $Mod(\varphi_1) = Mod(\varphi_2)$，则称 φ_1 和 φ_2 为逻辑等值公式（简称为等值公式），记作 $\varphi_1 \equiv \varphi_2$；
- $Var(\varphi)$：出现在 φ 中的原子命题集；
- $\bigvee \Pi$ 和 $\bigwedge \Pi$ 分别表示有限集 Π 中公式的析取和合取；
- V-无关（V-irrelevant）：给定公式 φ 和原子命题集 V，如果存在一个公式 ψ 使得 $Var(\psi) \cap V = \emptyset$ 且 $\varphi \equiv \psi$，那么说 φ 与 V 中的原子命题无关，简称为V-无关（V-irrelevant），写作 $\mathrm{IR}(\varphi, V)$。
- 文字（literal）、子句（clause）、析取范式等跟经典命题情形中的定义一样。

$\mathrm{SNF}^g_{\mathrm{CTL}}$ 子句

具有下面几种形式的公式称为 CTL 全局子句分离范式（separated normal form with global clauses for CTL，$\mathrm{SNF}^g_{\mathrm{CTL}}$ 子句）[12, 11]：

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

其中 k 和 n 都是大于 0 的常量，l_i（$1 \leq i \leq n$）、m_j（$1 \leq j \leq k$）和 l 都是文字且 $ind \in \mathrm{Ind}$。

CTL 的标准形式

转换规则

一个 CTL 公式 φ 可以通过下表中的规则转换为一个 $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ 子句集，记为 T_{φ}。

表 2: 转换规则

$$\textbf{Trans(1)}\ \frac{q\to \mathrm{E}T\varphi}{q\to \mathrm{E}_{\langle ind\rangle}T\varphi};\qquad \textbf{Trans(2)}\ \frac{q\to \mathrm{E}(\varphi_1 \mathrm{U} \varphi_2)}{q\to \mathrm{E}_{\langle ind\rangle}(\varphi_1 \mathrm{U}\varphi_2)};\qquad \textbf{Trans(3)}\ \frac{q\to\varphi_1\wedge\varphi_2}{q\to\varphi_1, q\to\varphi_2};$$

$$\textbf{Trans(4)}\ \frac{q\to\varphi_1\vee\varphi_2\ (\text{如果 } \varphi_2 \text{ 不是子句})}{q\to\varphi_1\vee p, p\to\varphi_2};\qquad \textbf{Trans(5)}\ \frac{q\to D}{\top\to\neg q\vee D};\ \frac{q\to\bot}{\top\to\neg q};\ \frac{q\to\top}{\{\}}$$

$$\textbf{Trans(6)}\ \frac{q\to Q\mathrm{X}\varphi\ (\text{如果 } \varphi \text{ 不是子句})}{q\to Q\mathrm{X}p, p\to\varphi};\qquad \textbf{Trans(7)}\ \frac{q\to Q\mathrm{F}\varphi\ (\text{如果 } \varphi \text{ 不是文字})}{q\to Q\mathrm{F}p, p\to\varphi};$$

$$\textbf{Trans(8)}\ \frac{q\to Q(\varphi_1 \mathrm{U}\varphi_2)\ (\text{如果 } \varphi_2 \text{ 不是文字})}{q\to Q(\varphi_1 \mathrm{U} p), p\to\varphi_2};\qquad \textbf{Trans(10)}\ \frac{q\to Q\mathrm{G}\varphi}{q\to p, p\to\varphi, p\to Q\mathrm{X}p};$$

$$\textbf{Trans(9)}\ \frac{q\to Q(\varphi_1 \mathrm{W}\varphi_2)\ (\text{如果 } \varphi_2 \text{ 不是文字})}{q\to Q(\varphi_1 \mathrm{W} p), p\to\varphi_2};$$

$$\textbf{Trans(11)}\ \frac{q\to Q(\varphi \mathrm{U} l)}{q\to l\vee p, p\to\varphi, p\to Q\mathrm{X}(l\vee p), q\to Q\mathrm{F}l};\qquad \textbf{Trans(12)}\ \frac{q\to Q(\varphi \mathrm{W} l)}{q\to l\vee p, p\to\varphi, p\to Q\mathrm{X}(l\vee p)}.$$

其中，$T\in\{\mathrm{X},\mathrm{G},\mathrm{F}\}$，$ind$ 是规则中引入的新索引且 $Q\in\{\mathrm{A},\mathrm{E}_{\langle ind\rangle}\}$；$q$ 是一个原子命题，l 是一个文字，D 是文字的析取（即子句），p 是新的原子命题；φ，φ_1，和 φ_2 都是 CTL 公式。

例 4

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

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

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

(3) 使用转换规则转换 $\{\mathrm{AG}(\mathbf{start} \to z), \mathrm{AG}(z \to (\mathrm{EG}\neg p \wedge \mathrm{AF} p))\}$，详细步骤如下：

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

因此，得到的 φ 对应的 $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ 子句集为：

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

μ-演算的语法

不动点符号：μ 和 ν，分别表示“最小不动点”和“最大不动点”。

$\mathscr{V}$：变元符号的可数集。

各类符号之间的优先级如下（从左到右优先级逐渐变低）：

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

定义 5 (μ-演算公式)

μ-演算公式（简称为 μ-公式或公式）递归定义如下：

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

其中 $p\in\mathscr{A}$ 且 $X\in\mathscr{V}$。

约定

- 公式 $\nu X.\varphi$ 中的 X 总是正出现在 φ 中，即：φ 中 X 的每一次出现之前都有偶数个否定符号“¬”；
- 称出现在 $\mu X.\varphi$ 和 $\nu X.\varphi$ 中的变元 X 是受约束的（bound），且受约束的变元称为约束变元，不受约束的变元称为自由变元；
- 文字（literal）：原子命题和变元符号及其各自的否定；
- 这里所谈到的公式指的是取名恰当的（well-named）、受保护（guarded）的 μ-公式。

μ-演算的语法

不动点符号：μ 和 ν，分别表示“最小不动点”和“最大不动点”。
$\mathscr{V}$：变元符号的可数集。
各类符号之间的优先级如下（从左到右优先级逐渐变低）：

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

定义 5 (μ-演算公式)

μ-演算公式（简称为 μ-公式或公式）递归定义如下：

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

其中 $p \in \mathscr{A}$ 且 $X \in \mathscr{V}$。

约定

- 公式 $\nu X.\varphi$ 中的 X 总是正出现在 φ 中，即：φ 中 X 的每一次出现之前都有偶数个否定符号“$\neg$”；
- 称出现在 $\mu X.\varphi$ 和 $\nu X.\varphi$ 中的变元 X 是受约束的（bound），且受约束的变元称为约束变元，不受约束的变元称为自由变元；
- 文字（literal）：原子命题和变元符号及其各自的否定；
- 这里所谈到的公式指的是取名恰当的（well-named）、受保护（guarded）的 μ-公式。

μ-演算的语义

定义 6

给定 μ-演算公式 φ、Kripke 结构 $\mathscr{M}=(S,R,L,r)$ 和一个从 $\mathscr{V}$ 中的变量到 $\mathscr{M}$ 中状态的赋值函数 $v:\mathscr{V}\to 2^S$。公式在 $\mathscr{M}$ 和 v 上的解释是 S 的一个子集 $\|\varphi\|_v^{\mathscr{M}}$（如果在上下文中 $\mathscr{M}$ 是明确的，则可以省去上标）：

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

其中，$v[X:=S']$ 是一个赋值函数，它除了 $v[X:=S'](X)=S'$ 之外，和 v 完全相同。

注意： 虽然这里的 Kripke 结构不要求其二元关系是完全的，但是这里的情况更加一般化，其结论也能推广到二元关系是完全的情形。

定义 6

给定 μ-演算公式 φ、Kripke 结构 $\mathscr{M}=(S,R,L,r)$ 和一个从 $\mathscr{V}$ 中的变量到 $\mathscr{M}$ 中状态的赋值函数 $v:\mathscr{V}\to 2^S$。公式在 $\mathscr{M}$ 和 v 上的解释是 S 的一个子集 $\|\varphi\|_v^{\mathscr{M}}$（如果在上下文中 $\mathscr{M}$ 是明确的，则可以省去上标）：

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

其中，$v[X:=S']$ 是一个赋值函数，它除了 $v[X:=S'](X)=S'$ 之外，和 v 完全相同。

记号和约定

- 赋值：由 $\mathscr{M}$、其赋值函数 v 和 $\mathscr{M}$ 上的状态 s 构成的三元组 $(\mathscr{M},s,v)$ 称为为赋值（当 s 为 $\mathscr{M}$ 的根时，$(\mathscr{M},s,v)$ 简写为 $(\mathscr{M},v)$，也称其为一个赋值）；
- 若 $s\in\|\varphi\|_v$，则称 s “满足” φ，记为 $(\mathscr{M},s,v)\models\varphi$；
- $Mod(\varphi)$：φ 的模型的集合，即 $Mod(\varphi)=\{(\mathscr{M},v)\mid(\mathscr{M},r,v)\models\varphi\}$（当 φ 为 μ-句子时，也可简写为 $Mod(\varphi)=\{\mathscr{M}\mid(\mathscr{M},r,v)\models\varphi\}$）；
- 当公式 φ 为 μ-句子时，可以将赋值函数 v 省略。

μ-公式的析取范式

μ-演算的覆盖 -语法

在覆盖 -语法语法中，用覆盖操作（cover operator）集替换上述 μ-公式的定义中的 EX，且满足

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

定义 7 (析取 μ-公式 [4])

析取 μ-公式集 $\mathscr{F}_d$ 是包含 $\top$、$\bot$ 和不矛盾的文字的合取且封闭于下面几条规则的最小集合:

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

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

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

目录

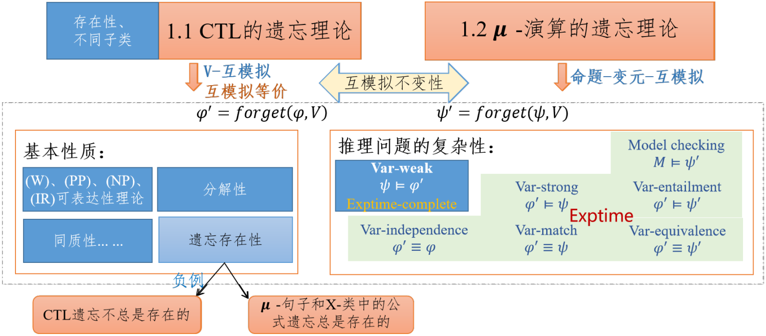


图 4: CTL 和 μ 遗忘理论

定义 8 (V-互模拟)

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

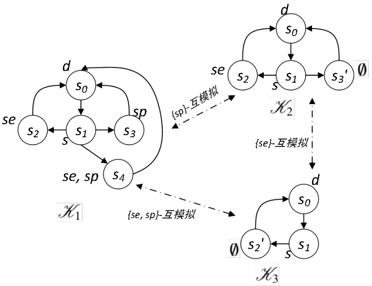
(i) $L_1(s_1) - V = L_2(s_2) - V$;

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

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

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

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



定义 8 (V-互模拟)

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

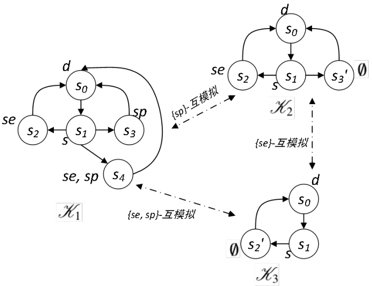
(i) $L_1(s_1) - V = L_2(s_2) - V$;

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

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

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

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



定义 8 (V-互模拟)

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

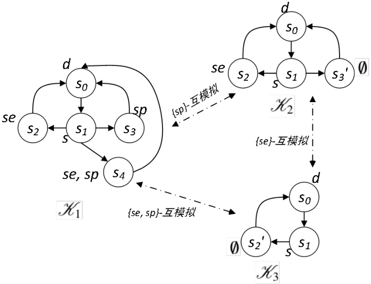
(i) $L_1(s_1) - V = L_2(s_2) - V$;

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

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

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

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



定义 9 (互模拟等价，bisimilar equivalence)

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

命题 1

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

定义 9 (互模拟等价，bisimilar equivalence)

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

命题 1

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

定义 10 (遗忘，forgetting)

令 V 是 $\mathscr{A}$ 的子集，Φ 是公式。如果公式 ψ 满足下面条件：

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

那么，称 ψ 为从 Φ 中遗忘 V 后得到的结果，记为 $\mathrm{F}_{\mathrm{CTL}}(\phi, V)$。

遗忘理论公设

给定 CTL 公式 φ、$\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$、原子命题集 $V \subseteq \mathscr{A}$ 和 $\varphi' = \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$，CTL 下遗忘理论公设如下：

(W) 削弱：$\varphi \models \varphi'$；

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

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

(IR) 无关性：$\mathrm{IR}(\varphi', V)$。

定理 11 (表达性定理，Representation Theorem)

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

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

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

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

例 11

令 p 和 x 为两个不同的原子命题，$\varphi(p,x)$[a]为下面公式合取 [10]：

$$\mathrm{AG}(\neg x \wedge \neg \mathrm{AG}p \to \neg \mathrm{AX}\neg x), \qquad \mathrm{AG}(\neg \mathrm{AX}\neg x \to \mathrm{AX}x),$$

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

Maksimova 证明了 $\varphi(p,x) \wedge \varphi(p,y) \models x \leftrightarrow y$，且不存在 CTL 公式 ψ 使得 $Var(\psi) = \{p\}$ 且 $\varphi(p,x) \models x \leftrightarrow \psi$，即 CTL 不具有 Beth 性质。

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

命题 2

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

定理 12

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

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

命题 2 (分解性，Decomposition)

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

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

命题 3 (同质性)

令 $\mathscr{T} \in \{\mathrm{X}, \mathrm{F}, \mathrm{G}\}$、$\mathscr{P} \in \{\mathrm{A}, \mathrm{E}\}$，$\phi$ 为 CTL 公式，且 $P \subseteq \mathscr{A}$ 为原子命题集，则：

$$\mathrm{F}_{\mathrm{CTL}}(\mathscr{P}\mathscr{T}\phi, P) \equiv \mathscr{P}\mathscr{T}\mathrm{F}_{\mathrm{CTL}}(\phi, P).$$

定义 11 (V-互模拟)

给定原子命题集 $V \subseteq \mathscr{A}$ 和两个 Kripke 结构 $\mathscr{M}_1$ 和 $\mathscr{M}_2$，其中 $\mathscr{M}_i = (S_i, R_i, L_i, r_i)$ $(i = 1, 2)$。若 $\mathscr{B} \subseteq S_1 \times S_2$ 满足下面几个条件：

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

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

$\mathscr{M}_1 \leftrightarrow_V \mathscr{M}_2$、$(\mathscr{M}_1, r_1) \leftrightarrow_V (\mathscr{M}_2, r_2)$：如果 $\mathscr{M}_1$ 和 $\mathscr{M}_2$ 之间存在一个 V-互模拟关系。

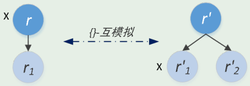
例 11 (不变性反例)

令 $\varphi = \mathrm{AX}\neg X \vee \mathrm{AX}X$，$(\mathscr{M}, v)$ 和 $(\mathscr{M}', v')$ 为赋值，其中 $\mathscr{M} = (S, r, R, L)$、$\mathscr{M}' = (S', r', R', L')$ 且

$$S = \{r, r_1\}, R = \{(r, r_1)\}, L(r) = L(r_1) = \emptyset, v(X) = \{r_1\},$$

$$S' = \{r', r'_1, r'_2\}, R' = \{(r', r'_1), (r', r'_2)\}, L(r') = L(r'_1) = L(r'_2) = \emptyset, v'(X) = \{r'_1\}.$$

$\mathscr{B} = \{(r, r'), (r_1, r'_1), (r_1, r'_2)\}$ 是 $\mathscr{M}$ 和 $\mathscr{M}'$ 之间的一个 $\emptyset$-互模拟。
但是，$(\mathscr{M}, v) \models \varphi$ 而 $(\mathscr{M}', v') \not\models \varphi$。



定义 11 (变元 -命题 -互模拟)

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

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

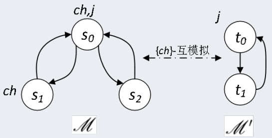
则称 $\mathscr{B}$ 是 $(\mathscr{M}_1, s_1, v_1)$ 和 $(\mathscr{M}_2, s_2, v_2)$ 之间的一个$\underline{\langle \mathscr{V}_1, V \rangle\text{-互模拟}}$。

- $\underline{(\mathscr{M}, s, v) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}', s', v')}$：若 $(\mathscr{M}, s, v)$ 和 $(\mathscr{M}', s', v')$ 之间存在一个 $\langle \mathscr{V}_1, V \rangle$-互模拟关系 $\mathscr{B}$，则称 $(\mathscr{M}, s, v)$ 和 $(\mathscr{M}', s', v')$ 是 $\langle \mathscr{V}_1, V \rangle$-互模拟的；
- 若 $s = r$ 且 $s' = r'$，则 $(\mathscr{M}, s, v) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}', s', v')$ 简写为 $(\mathscr{M}, v) \leftrightarrow_{\langle \mathscr{V}_1, V \rangle} (\mathscr{M}', v')$；
- $\langle \mathscr{V}_1, V \rangle$ 是一个等价关系。

例子

令 $\mathscr{M}$ 和 $\mathscr{M}'$ 为图中的 Kripke 结构，$v:\mathscr{V}\to 2^S$ 和 $v':\mathscr{V}\to 2^{S'}$ 为将 $\mathscr{V}$ 中的变元分别赋值到 $\mathscr{M}$ 和 $\mathscr{M}'$ 的状态集上的赋值函数。可以检查下面的结论成立：

- 若对任意 $X\in\mathscr{V}$，$v(X)=\{s_0,s_1,s_2\}$ 且 $v'(X)=\{t_0,t_1\}$，则 $(\mathscr{M},v)\leftrightarrow_{\{ch\}}(\mathscr{M}',v')$；
- 若对任意 $X\in\mathscr{V}-\{X_1\}$，$v(X_1)=\{s_0\}$、$v'(X_1)=\{t_1\}$、$v(X)=\{s_0,s_1,s_2\}$ 且 $v'(X)=\{t_0,t_1\}$，则 $(\mathscr{M},v)\not\leftrightarrow_{\{ch\}}(\mathscr{M}',v')$；这是因为 $(s_0,t_0)\in\mathscr{B}$ 且 $s_0\in v(X_1)$，但是 $t_0\notin v'(X_1)$。



命题 4 (不变性)

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

定义 11 (μ-演算遗忘)

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

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

与 CTL 共同性质

表达性定理、分解性、同质性等。

定理 12 (存在性)

给定原子命题 $q \in \mathscr{A}$ 和 μ-句子 φ，则存在一个 μ-句子 ψ 使得 $Var(\psi) \cap \{q\} = \emptyset$ 且 $\psi \equiv \mathrm{F}_\mu(\varphi, \{q\})$。

命题 5 (同质性)

给定原子命题集 $V \subseteq \mathscr{A}$ 和 μ-公式 φ，则:

(iii) 如果 $\nu X.\varphi$ 为 μ-句子，$\mathrm{F}_\mu(\nu X.\varphi, V) \equiv \nu X.\mathrm{F}_\mu(\varphi, V)$;

(iv) 如果 $\mu X.\varphi$ 为 μ-句子，$\mathrm{F}_\mu(\mu X.\varphi, V) \equiv \mu X.\mathrm{F}_\mu(\varphi, V)$。

定义 11 (μ-演算遗忘)

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

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

与 CTL 共同性质

表达性定理、分解性、同质性等。

定理 12 (存在性)

给定原子命题 $q \in \mathcal{A}$ 和 μ-句子 φ，则存在一个 μ-句子 ψ 使得 $Var(\psi) \cap \{q\} = \emptyset$ 且 $\psi \equiv \mathrm{F}_\mu(\varphi, \{q\})$。

命题 5 (同质性)

给定原子命题集 $V \subseteq \mathcal{A}$ 和 μ-公式 φ，则:

(iii) 如果 $\nu X.\varphi$ 为 μ-句子，$\mathrm{F}_\mu(\nu X.\varphi, V) \equiv \nu X.\mathrm{F}_\mu(\varphi, V)$;

(iv) 如果 $\mu X.\varphi$ 为 μ-句子，$\mathrm{F}_\mu(\mu X.\varphi, V) \equiv \mu X.\mathrm{F}_\mu(\varphi, V)$。

基于遗忘的反应式系统最弱充分条件研究

绪论
研究背景和意义
国内外研究现状
研究目标
研究内容及拟解决的关键科学问题
背景知识
Kripke 结构
CTL 的语法和语义
μ-演算
CTL 和 μ-演算遗忘理论
CTL 遗忘理论
μ-演算遗忘理论
遗忘理论在反应式系统中的应用
简介
最弱充分条件
知识更新
CTL 遗忘计算方法
简介
基于模型的有界 CTL 遗忘计算
基于归结的遗忘计算方法
基于归结的算法
CTL-forget 实现及实验
总结与展望
总结
展望
参考文献

x-类

不含有不定点操作的 μ-公式集，记为**x-类**。通过等值式：$\mathrm{AX}\varphi_1 \wedge \mathrm{AX}\varphi_2 \equiv \mathrm{AX}(\varphi_1 \wedge \varphi_2)$ 和 $\mathrm{EX}\varphi_1 \vee \mathrm{EX}\varphi_2 \equiv \mathrm{EX}(\varphi_1 \vee \varphi_2)$，可以将 x-类中的任意公式转换为具有下面形式的公式的析取：

$$\varphi_0 \wedge \mathrm{AX}\varphi_1 \wedge \mathrm{EX}\varphi_2 \wedge \cdots \wedge \mathrm{EX}\varphi_n, \tag{1}$$

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

命题 6

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

例 13

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

如下计算公式 φ 的度：

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

此外，公式 φ 可如下转换为具有形式 (1) 的公式的析取：

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

则从 φ 中遗忘 V 的结果为：

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

命题 7 (模型检测)

给定一个有限的 *Kripke* 结构 $\mathscr{M}$、一个 μ-句子 φ 和原子命题集 $V \subseteq \mathscr{A}$。有：

(i) 判定 $\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$ 在 EXPTIME 中；

(ii) 若 φ 是一个析取 μ-公式，则判定 $\mathscr{M} \models^? \mathrm{F}_\mu(\varphi, V)$ 在 NP$\cap$*co*-NP 中。

定理 14 (Entailment)

给定 μ-句子 φ 和 ψ，V 为原子命题集，则：

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

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

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

目录

基于遗忘的反应式系统最弱充分条件研究

绪论
研究背景和意义
国内外研究现状
研究目标
研究内容及拟解决的关键科学问题
背景知识
Kripke 结构
CTL 的语法和语义
μ-演算
CTL 和 μ-演算遗忘理论
CTL 遗忘理论
μ-演算遗忘理论
遗忘理论在反应式系统中的应用
简介
最弱充分条件
知识更新
CTL 遗忘计算方法
简介
基于模型的有界 CTL 遗忘计算
基于归结的遗忘计算方法
基于归结的算法 CTL-forget 实现及实验
总结与展望
总结
展望
参考文献

- 反应式系统被表示成 Kripke 结构；
- 初始 Kripke 结构的特征公式看作 CTL 公式；

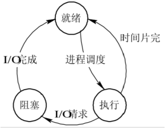
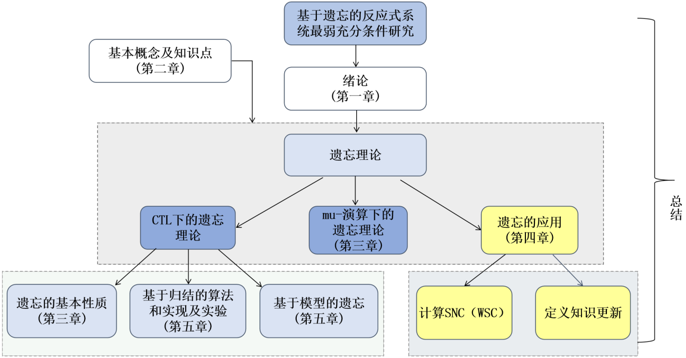


图 5: 进程的三种基本状态及其转换



定义 15 (充分和必要条件)

给定两个公式 φ 和 ψ，$V \subseteq Var(\varphi)$，$q \in Var(\varphi) - V$ 和 $Var(\psi) \subseteq V$。

- 若 $\varphi \models q \rightarrow \psi$，则称 ψ 是 q 在 V 和 φ 上的必要条件（necessary condition，NC）；
- 若 $\varphi \models \psi \rightarrow q$，则称 ψ 是 q 在 V 和 φ 上的充分条件（sufficient condition，SC）；
- 若 ψ 是 q 在 V 和 φ 上的必要条件，且对于任意 q 在 V 和 φ 上的必要条件 ψ'，都有 $\varphi \models \psi \rightarrow \psi'$，则称 ψ 是 q 在 V 和 φ 上的最强必要条件（strongest necessary condition，SNC）；
- 若 ψ 是 q 在 V 和 φ 上的充分条件，且对于任意 q 在 V 和 φ 上的充分条件 ψ'，都有 $\varphi \models \psi' \rightarrow \psi$，则称 ψ 是 q 在 V 和 φ 上的最弱充分条件（weakest sufficient condition，WSC）。

命题 8 (对偶性)

令 V、q、φ 和 ψ 为定义15出现的符号。则 ψ 是 q 在 V 和 φ 上的 SNC（WSC）当且仅当 $\neg\psi$ 是 $\neg q$ 在 V 和 φ 上的 WSC（SNC）。

命题 9

给定公式 Γ 和 α，$V \subseteq Var(\alpha) \cup Var(\Gamma)$，$q$ 是不出现在 Γ 和 α 中的原子命题。φ 是集合 V 上的公式，则 φ 是 α 在 V 和 Γ 上的 SNC（WSC）当且仅当 φ 是 q 在 V 和 Γ' 上的 SNC（WSC），其中 $\Gamma' = \Gamma \cup \{q \leftrightarrow \alpha\}$。

定理 16

给定公式 φ、原子命题集 $V \subseteq Var(\varphi)$ 和原子命题 $q \in Var(\varphi) - V$。

(i) $\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$ 是 q 在 V 和 φ 上的 SNC;

(ii) $\neg\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge \neg q, (Var(\varphi) \cup \{q\}) - V)$ 是 q 在 V 和 φ 上的 WSC。

例 17 (例 1的延续)

令 $\mathscr{A} = \{d, se, sp, s\}$ 和 $V = \{d, se\}$，求 s 在 V 和初始结构 $\mathscr{K} = (\mathscr{M}, s_0)$ 上的 WSC，其中 $\mathscr{M}$ 为例 1中初始状态为 s_0 的汽车制造企业模型结构。

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

由于涉及到后文中遗忘的计算方法，本例的详细计算过程放到后面。

约定

- 本小节假设所有初始结构都是有限的，即：状态来源于有限状态空间且 $\mathscr{A}$ 为有限原子命题集；
- 任意 $\mathscr{A}$ 上的有限初始结构 $\mathscr{M}$（为了简化符号，本节用初始 Kripke 结构 $\mathscr{M}$ 代替初始结构 $(\mathscr{M}, s_0)$）都能用一个 CTL 公式——特征公式 $\mathscr{F}_{\mathscr{A}}(\mathscr{M})$ 来表示；
- 给定公式 φ 和 ψ，$V_{min} \subseteq \mathscr{A}$ 为使得 $\mathrm{F}_{\mathrm{CTL}}(\varphi, V_{min}) \wedge \psi$ 可满足的极小子集。
- 记
$$\bigcup_{V_{min} \subseteq \mathscr{A}} Mod(\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \psi)$$
为所有 $\mathrm{F}_{\mathrm{CTL}}(\mathscr{F}_{\mathscr{A}}(\mathscr{M}), V_{min}) \wedge \psi$ 的模型集合的并集。

定义 18

给定公式 Γ 和 φ。知识更新操作 $\diamond_{\mathrm{CTL}}$ 定义如下：

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

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

定义 19

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

例 20

令 $\mathscr{M} = (S, R, L, r)$、$\mathscr{M}_1 = (S_1, R_1, L_1, r_1)$、$\mathscr{M}_2 = (S_2, R_2, L_2, r_2)$ 为三个初始结构（如图 6），其中 $S = S_1 = S_2 = \{s_0, s_1\}$，$r = r_1 = r_2 = s_0$，$R = R_1 = R_2 = \{(s_0, s_1), (s_1, s_1)\}$，$L(s_0) = \{ch, j\}$，$L_1(s_0) = L_2(s_0) = \{ch\}$，$L(s_1) = L_1(s_1) = \emptyset$，$L_2(s_1) = \{j\}$。
可以检查 $\mathscr{M} \leftrightarrow_{\{j\}} \mathscr{M}_1$，$\mathscr{M} \leftrightarrow_{\{j,ch\}} \mathscr{M}_2$，$\{j\} \subseteq \{j, ch\}$，且对任意原子命题集 $V' \subset \{j\}$（或 $V' \subset \{j, ch\}$），有 $\mathscr{M} \not\leftrightarrow_{V'} \mathscr{M}_1$（或 $\mathscr{M} \not\leftrightarrow_{V'} \mathscr{M}_2$）。因此，$\mathscr{M}_1 \leq_{\mathscr{M}} \mathscr{M}_2$。

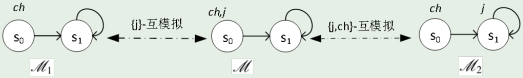


图 6: 初始结构间的 $\leq_{\mathscr{M}}$ 关系。

定义 19

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

定理 20

给定 μ-句子 Γ 和 φ，则：

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

其中，$Min(Mod(\varphi), \leq_{\mathcal{M}})$ 是 φ 的关于偏序关系 $\leq_{\mathcal{M}}$ 的极小模型集。

定理 21

知识更新操作 $\diamond_{\mathrm{CTL}}$ 满足 *Katsuno* 和 *Mendelzon* 提出的基本条件 *(U1)-(U8)*。

例 22

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

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

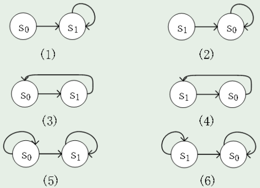


图 6: 状态空间为 $\{s_0,s_1\}$ 的六个 Kripke 结构示意图[a]

[a]这里只列出部分转换关系，其余转换关系可以容易地枚举出来。

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

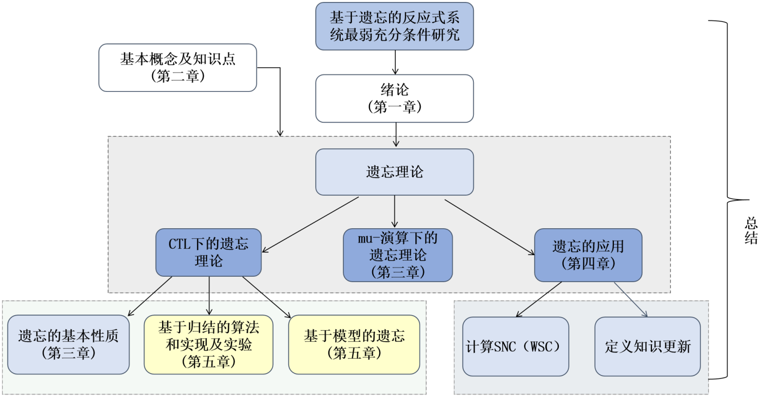
$Mod(\varphi\diamond_\mu\psi)=\bigcup_{\mathscr{M}\in Mod(\varphi)}Min(Mod(\psi),\leq_{\mathscr{M}})$，根据定义 19容易检查 $Mod(\varphi\diamond_\mu\psi)=Mod(\psi)$。直观地说，由于在 ψ 中 j 在偶数状态不再为真、ch 保持为真且 ψ 和 φ 都不知道模型偶数状态的信息，因而用 ψ 更新 φ 得到的结果为 ψ 自身。

绪论
研究背景和意义
国内外研究现状
研究目标
研究内容及拟解决的关键科学问题
背景知识
Kripke 结构
CTL 的语法和语义
μ-演算
CTL 和 μ-演算遗忘理论
CTL 遗忘理论
μ-演算遗忘理论
遗忘理论在反应式系统中的应用
简介
最弱充分条件
知识更新
CTL 遗忘计算方法
简介
基于模型的有界 CTL 遗忘计算
基于归结的遗忘计算方法
基于归结的算法 CTL-forget 实现及实验
总结与展望
总结
展望
参考文献

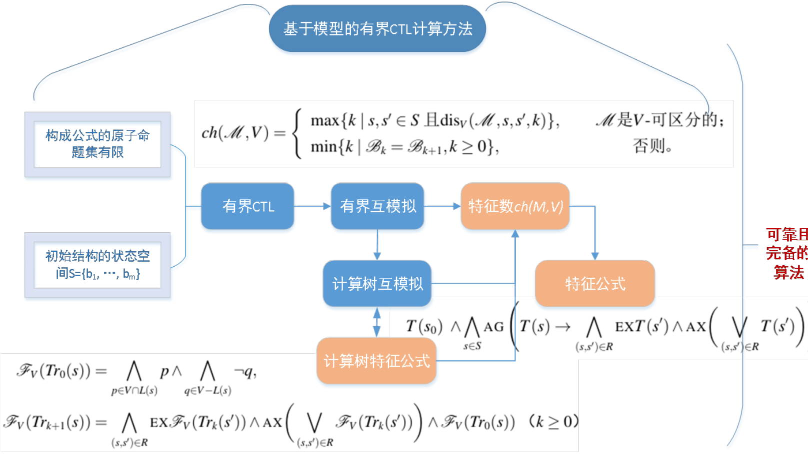
目录

简介

- 基于模型的计算方法；
- 基于归结的计算方法（CTL-forget 算法）；
- 基于 Prolog 的 CTL-forget 算法实现。



基于模型的计算方法总体框架



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

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

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

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

图 7: 基于模型的有界 CTL 遗忘方法

$\mathscr{B}_n^V$

令 $V\subseteq\mathscr{A}$ 是原子命题集，$i\in\{1,2\}$，$\mathscr{M}_i=(S_i,R_i,L_i,s_0^i)$ 是初始 Kripke 结构，$\mathscr{K}_i=(\mathscr{M}_i,s_i)$ 是结构。$\mathscr{B}_n^V$ 递归定义如下：

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

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

定义 23 (有界 V-互模拟)

令 V 是 $\mathscr{A}$ 的一个子集，$i\in\{1,2\}$，$\mathscr{K}_1$ 和 $\mathscr{K}_2$ 是结构。

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

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

$\mathscr{B}_n^V$

令 $V \subseteq \mathscr{A}$ 是原子命题集，$i \in \{1,2\}$，$\mathscr{M}_i = (S_i, R_i, L_i, s_0^i)$ 是初始 Kripke 结构，$\mathscr{K}_i = (\mathscr{M}_i, s_i)$ 是结构。$\mathscr{B}_n^V$ 递归定义如下：

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

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

定理 23

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

计算树

给定一个初始 Kripke 结构 $\mathscr{M}=(S,R,L,s_0)$ 和一个状态 $s\in S$，$\mathscr{M}$ 上以 s 为根节点、深度为 n（$n\geq 0$）的计算树$\mathrm{Tr}_n^{\mathscr{M}}(s)$ 递归定义如下 [1]：

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

计算树互模拟

给定原子命题集 $V\subseteq\mathscr{A}$ 和初始 Kripke 结构 $\mathscr{M}_i$（$i=1,2$）。如果下面条件同时满足：

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

则称 $\mathscr{M}_1$ 的计算树 $\mathrm{Tr}_n(s_1)$ 和 $\mathscr{M}_2$ 的计算树 $\mathrm{Tr}_n(s_2)$ 是 V-互模拟的（记为 $(\mathscr{M}_1,\mathrm{Tr}_n(s_1))\leftrightarrow_V(\mathscr{M}_2,\mathrm{Tr}_n(s_2))$，简写为 $\mathrm{Tr}_n(s_1)\leftrightarrow_V\mathrm{Tr}_n(s_2)$）。

计算树

给定一个初始 Kripke 结构 $\mathscr{M}=(S,R,L,s_0)$ 和一个状态 $s\in S$，$\mathscr{M}$ 上以 s 为根节点、深度为 n（$n\geq 0$）的计算树$\mathrm{Tr}_n^{\mathscr{M}}(s)$ 递归定义如下 [1]:

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

命题 8

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

定义 24

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

$$\mathscr{F}_V(\mathrm{Tr}_0(s)) = \bigwedge_{p \in V \cap L(s)} p \wedge \bigwedge_{q \in V - L(s)} \neg q,$$

$$\mathscr{F}_V(\mathrm{Tr}_{k+1}(s)) = \bigwedge_{(s,s') \in R} \mathrm{EX}\mathscr{F}_V(\mathrm{Tr}_k(s')) \wedge \mathrm{AX}\left(\bigvee_{(s,s') \in R} \mathscr{F}_V(\mathrm{Tr}_k(s'))\right) \wedge \mathscr{F}_V(\mathrm{Tr}_0(s)) \quad (k \geq 0)。$$

含义

由定义24可知，计算树的特征公式从三个方面展示了计算树的信息：

(1) 只考虑 V 中的原子命题；

(2) 突出了树节点的内容，即：对于任意原子命题 $p \in V$，若 p 在节点的标签中，则其正出现在特征公式中，否则负出现在特征公式中；

(3) 公式中的时序算子表示了状态之间的转换关系。

引理 24

给定原子命题集 $V \subseteq \mathcal{A}$、初始 Kripke 结构 $\mathcal{M} = (S, R, L, s_0)$ 和 $\mathcal{M}' = (S', R', L', s_0')$、$s \in S$、$s' \in S'$ 且 $n \geq 0$。若 $\mathrm{Tr}_n(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_n(s')$，则 $\mathcal{F}_V(\mathrm{Tr}_n(s)) \equiv \mathcal{F}_V(\mathrm{Tr}_n(s'))$。

引理 25

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

(i) $(\mathcal{M}, s) \models \mathcal{F}_V(\mathrm{Tr}_n(s))$；

(ii) 若 $(\mathcal{M}, s) \models \mathcal{F}_V(\mathrm{Tr}_n(s'))$，则 $\mathrm{Tr}_n(s) \leftrightarrow_{\overline{V}} \mathrm{Tr}_n(s')$。

V-可区分

若初始 Kripke 结构 $\mathcal{M}$ 的两个状态 s 和 s' 不是 $\overline{V}$-互模拟的（即：$s \not\leftrightarrow_{\overline{V}} s'$），则称 s 和 s' 是V-可区分的。用 $\mathrm{dis}_V(\mathcal{M}, s, s', k)$ 表示状态 s 和 s' 在命题**??**中所说的最小数 k 下是 V-可区分的。

特征数

$\mathcal{M}$ 关于原子命题集 V 的特征数，记为 $ch(\mathcal{M}, V)$ 定义如下：

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

定义 26 (特征公式)

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

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

定理 27

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

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

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

例 26

考虑右下图中左边的初始结构 $\mathscr{K}_2=(\mathscr{M},s_0)$。左边的为 $\mathscr{M}$ 上的四棵计算树：从左到右表示以 s_0 为根、深度分别为 0、1、2 和 3 的计算树（为简化图，计算树的标签没有给出，但是每个树节点的标签可从 $\mathscr{K}_2$ 找到）。令 $V=\{d\}$，则 $\overline{V}=\{s,se\}$。
因为 $L(s_1)-\overline{V}=L(s_2)-\overline{V}$，所以有 $\mathrm{Tr}_0(s_1)\leftrightarrow_{\overline{V}}\mathrm{Tr}_0(s_2)$。由于存在 $(s_1,s_2)\in R$，使得对任意 $(s_2,s')\in R$，都有 $L(s_2)-\overline{V}\neq L(s')-\overline{V}$，所以，$\mathrm{Tr}_1(s_1)\not\leftrightarrow_{\overline{V}}\mathrm{Tr}_1(s_2)$。由此可知，$s_1$ 和 s_2 是 V-可区分的，且 $\mathrm{dis}_V(\mathscr{M},s_1,s_2,1)$。
同理可得：$\mathrm{dis}_V(\mathscr{M},s_0,s_1,0)$、$\mathrm{dis}_V(\mathscr{M},s_1,s_3',1)$、$\mathrm{dis}_V(\mathscr{M},s_0,s_2,0)$ 和 $\mathrm{dis}_V(\mathscr{M},s_0,s_3',0)$。此外，$s_2\leftrightarrow_{\overline{V}}s_3'$。因此，可以计算 $\mathscr{M}$ 关于 V 的特征数为：

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

$\mathrm{Tr}_2(s_0)$ $\mathrm{Tr}_3(s_0)$
所以，可以由以下步骤计算 $\mathscr{K}_2$ 关于 V 的特征公式：

$\mathscr{F}_V(\mathrm{Tr}_0(s_0))=d,\qquad \mathscr{F}_V(\mathrm{Tr}_0(s_1))=\neg d,$

$\mathscr{F}_V(\mathrm{Tr}_0(s_2))=\neg d,\qquad \mathscr{F}_V(\mathrm{Tr}_0(s_3'))=\neg d,$

$\mathscr{F}_V(\mathrm{Tr}_1(s_0))=\mathrm{EX}\neg d\wedge \mathrm{AX}\neg d\wedge d\equiv \mathrm{AX}\neg d\wedge d,$

$\mathscr{F}_V(\mathrm{Tr}_1(s_1))=\mathrm{EX}\neg d\wedge \mathrm{EX}\neg d\wedge \mathrm{AX}(\neg d\vee\neg d)\wedge\neg d\equiv \mathrm{AX}\neg d\wedge\neg d,$

$\mathscr{F}_V(\mathrm{Tr}_1(s_2))=\mathrm{EX}d\wedge \mathrm{AX}d\wedge\neg d\equiv \mathrm{AX}d\wedge\neg d,$

$\mathscr{F}_V(\mathrm{Tr}_1(s_3'))\equiv\mathscr{F}_V(\mathrm{Tr}_1(s_2)),$

$$\begin{aligned}\mathscr{F}_V(\mathscr{M},s_0)\equiv{}& \mathrm{AX}\neg d\wedge d\wedge\\ & \mathrm{AG}(\mathrm{AX}\neg d\wedge d\rightarrow \mathrm{AX}(\mathrm{AX}\neg d\wedge\neg d))\wedge\\ & \mathrm{AG}(\mathrm{AX}\neg d\wedge\neg d\rightarrow \mathrm{AX}(\mathrm{AX}d\wedge\neg d))\wedge\\ & \mathrm{AG}(\mathrm{AX}d\wedge\neg d\rightarrow \mathrm{AX}(\mathrm{AX}\neg d\wedge d)).\end{aligned}$$

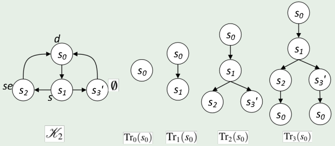


图 8: 初始结构 $\mathscr{K}_2$ 及其计算树示意图

引理 27

给定 CTL 公式 φ，下面等式成立：

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

遗忘封闭性

从 φ 中遗忘 V 中的元素得到的结果为：

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

$\mathrm{CTL}_{\mathrm{AF}}$：表示 CTL 公式只包含时序算子 AF 的子类。

命题 9 (模型检测)

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

定理 27 (Entailment)

令 φ 和 ψ 为 $\mathrm{CTL}_{\mathrm{AF}}$ 中的两个公式，V 为原子命题集。则：

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

(ii) 判定 $\psi \models^? \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$ 是 Π_2^{P}-完全的，

(iii) 判定 $\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \models^? \mathrm{F}_{\mathrm{CTL}}(\psi, V)$ 是 Π_2^{P}-完全的。

推论 28

令 φ 和 ψ 为 $\mathrm{CTL}_{\mathrm{AF}}$ 中的两个公式，V 原子公式集。则

(i) 判定 $\psi \equiv^? \mathrm{F}_{\mathrm{CTL}}(\varphi, V)$ 是 Π_2^{P}-完全的，

(ii) 判定 $\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \equiv^? \varphi$ 是 co-NP-完全的，

(iii) 判定 $\mathrm{F}_{\mathrm{CTL}}(\varphi, V) \equiv^? \mathrm{F}_{\mathrm{CTL}}(\psi, V)$ 是 Π_2^{P}-完全的。

算法 5.1 基于模型的CTL遗忘过程

Input: CTL公式φ和原子命题集V

Output: $\mathrm{F}_{\mathrm{CTL}}(\varphi, V)$

$\psi \leftarrow \bot$ **foreach** $\mathscr{A}$ 和$\mathscr{S}$上的初始结构$\mathscr{K}$ **do**

    if $\mathscr{K} \not\models \varphi$ **then continue**

    foreach 满足$\mathscr{K} \leftrightarrow_V \mathscr{K}'$的初始结构$\mathscr{K}'$ **do**

        $\psi \leftarrow \psi \vee \mathscr{F}_{\overline{V}}(\mathscr{K}')$

    end

end

return ψ

命题 10

令 φ 为 CTL 公式，$V \subseteq \mathscr{A}$ 为原子命题集，状态空间大小为 $|\mathscr{S}| = m$，$|\mathscr{A}| = n$，$|V| = x$。使用算法 5.1 计算从 φ 中遗忘 V 中原子的空间复杂度为 $O((n-x)m^{2(m+2)}2^{nm}\log m)$，且时间复杂性至少与空间复杂性相同。

基于归结的算法 CTL-forget——总体框架

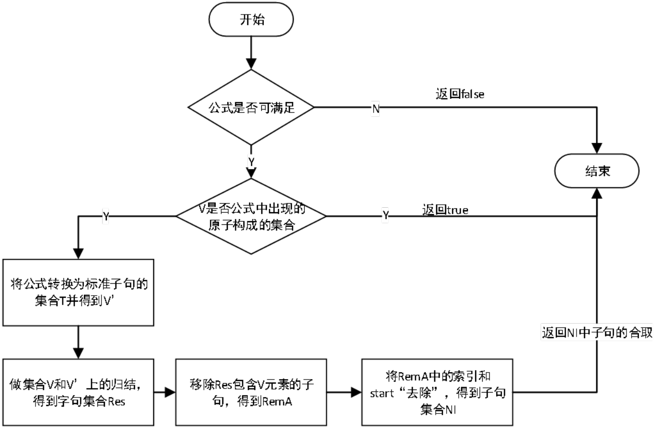


图 9: 基于归结的遗忘的主要流程图

- 如何表示 CTL 公式和带索引的 CTL 公式之间的关系？
- 如何“移除”无关的原子命题（包括需要遗忘的原子命题和转换过程中引入的新的原子命题），以及如何“消除”索引？

基于归结的算法 CTL-forget——总体框架

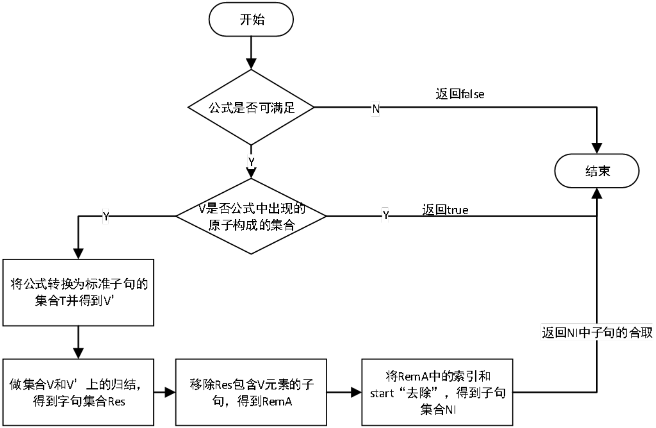


图 9: 基于归结的遗忘的主要流程图

- 如何表示 CTL 公式和带索引的 CTL 公式之间的关系？
- 如何“移除”无关的原子命题（包括需要遗忘的原子命题和转换过程中引入的新的原子命题），以及如何“消除”索引？

表 3: $R^{\succ,S}_{\mathrm{CTL}}$ 归结系统

$$\textbf{(SRES1)}\ \frac{P \to \mathrm{AX}(C \vee l),\ Q \to \mathrm{AX}(D \vee \neg l)}{P \wedge Q \to \mathrm{AX}(C \vee D)};$$

$$\textbf{(SRES2)}\ \frac{P \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee l),\ Q \to \mathrm{AX}(D \vee \neg l)}{P \wedge Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)};$$

$$\textbf{(SRES3)}\ \frac{P \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee l),\ Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(D \vee \neg l)}{P \wedge Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)};$$

$$\textbf{(SRES4)}\ \frac{\textbf{start} \to C \vee l,\ \textbf{start} \to D \vee \neg l}{\textbf{start} \to C \vee D};$$

$$\textbf{(SRES5)}\ \frac{\top \to C \vee l,\ \textbf{start} \to D \vee \neg l}{\textbf{start} \to C \vee D};$$

$$\textbf{(SRES6)}\ \frac{\top \to C \vee l,\ Q \to \mathrm{AX}(D \vee \neg l)}{Q \to \mathrm{AX}(C \vee D)};$$

$$\textbf{(SRES7)}\ \frac{\top \to C \vee l,\ Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(D \vee \neg l)}{Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}(C \vee D)};$$

$$\textbf{(SRES8)}\ \frac{\top \to C \vee l,\ \top \to D \vee \neg l}{\top \to C \vee D};$$

$$\textbf{(RW1)}\ \frac{\bigwedge_{i=1}^{n} m_i \to \mathrm{AX}\bot}{\top \to \bigvee_{i=1}^{n} \neg m};$$

$$\textbf{(RW2)}\ \frac{\bigwedge_{i=1}^{n} m_i \to \mathrm{E}_{\langle ind \rangle}\mathrm{X}\bot}{\top \to \bigvee_{i=1}^{n} \neg m};$$

$$\textbf{(ERES1)}\ \frac{\Lambda \to \mathrm{EXEG}l,\ Q \to \mathrm{AF}\neg l}{Q \to \mathrm{A}(\neg \Lambda \mathrm{W} \neg l)};$$

$$\textbf{(ERES2)}\ \frac{\Lambda \to \mathrm{E}_{\langle ind \rangle}\mathrm{XE}_{\langle ind \rangle}\mathrm{G}l,\ Q \to \mathrm{E}_{\langle ind \rangle}\mathrm{F}\neg l}{Q \to \mathrm{E}_{\langle ind \rangle}(\neg \Lambda \mathrm{W} \neg l)}.$$

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

记号

- 令 T 为 $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ 子句集，p 为原子命题。T 在 p 上的展开（记为 $\mathrm{UF}(T,p)$）是集合 T 和如下集合的并集：

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

- $\mathrm{UF}(T,\emptyset) = T$ 且 $\mathrm{UF}(T,\{p\}\cup V) = \mathrm{UF}(\mathrm{UF}(T,p),V)$;
- $\underline{ERes}(\varphi,V) = \{\alpha \in \mathrm{UF}(T_\varphi,V) \mid Var(\alpha)\cap V = \emptyset\}$。

命题 11

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

例 29 (例??的延续)

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

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

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

$$\textbf{start}\to z,\quad \textbf{start}\to f\vee m\vee q,\quad \textbf{start}\to x\vee y,\quad \textbf{start}\to q\vee x,\quad \textbf{start}\to f\vee m\vee y,$$

$$\top\to f\vee m\vee\neg x,\quad \top\to q\vee f\vee m\vee\neg z,\quad \top\to f\vee m\vee\neg z\vee y,$$

$$\top\to q\vee x\vee\neg z,\quad \top\to x\vee y\vee\neg z,\quad \top\to q\vee\neg y,\quad z\to \mathrm{AF}x,$$

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

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

两个主要过程

- 消除索引；
- 移除新引入的原子命题。

引理 30

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

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

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

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

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

算法 5.2 RM-index(Σ)

Input: 有限$\mathrm{SNF}^g_{\mathrm{CTL}}$子句集Σ
Output: CTL公式集
foreach Σ中拥有相同索引$\langle i\rangle$的E-子句构成的极大子集Δ **do**
  if 存在索引为$\langle i\rangle$的E-某时子句$\alpha\in\Sigma$ **then**
    foreach $\beta\in rei(\Delta)$ **do** $\Sigma\leftarrow\Sigma\cup rfi(\alpha,\beta)$ $\Sigma\leftarrow\Sigma-\{\alpha\}$
  end
  $\Sigma\leftarrow\Sigma-\Delta\cup rxi(\Delta)$
end
return Σ

其中，$rei(\{\alpha_i\mid 1\le i\le n\})$、$rxi(\{\alpha_i\mid 1\le i\le n\})$、$rfi(\{\beta_1,\alpha_2\})$ 分别表示引理 33 中 (i)、(ii)、(iii) 等号 $\equiv_*$（$*\in\{$ 空字符串，$\emptyset\}$）的右边，$\alpha_i=\psi_i\rightarrow \mathrm{E}_{\langle j\rangle}\mathrm{X}\varphi_i$ $(1\le i\le n)$ 且 $\beta_1=\psi_1\rightarrow \mathrm{E}_{\langle j\rangle}\mathrm{F}\varphi_1$。

推论 30

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

引理 30 (一般化的 Ackermann 引理，Generalised Ackermann's Lemma)

令 x 为一个原子命题、$\Delta=\{\mathrm{AG}(\top\to\neg x\vee C_1),\ldots,\mathrm{AG}(\top\to\neg x\vee C_n),\mathrm{AG}(x\to B_1),\ldots,\mathrm{AG}(x\to B_m)\}$ 为只包含一个 x 的 CTL 公式集（$n,m\geq 1$）、Γ 为 x 正出现在其中的有限个 CTL 公式集。下面式子成立：

$$\Gamma\cup\Delta\equiv_{\{x\}}\Gamma\left[x/\bigwedge(\{C_i\mid 1\leq i\leq n\}\cup\{B_i\mid 1\leq i\leq m\})\right]. \tag{2}$$

例 30 (例??的延续)

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

$$\mathbf{start} \to z, \quad \mathbf{start} \to f \vee m \vee q, \quad \mathbf{start} \to f \vee m \vee y,$$

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

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

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

$$\mathbf{start} \to (q \vee f \vee m) \wedge (f \vee m \vee y) \wedge \mathrm{AF}(f \vee m), \quad \mathbf{start} \to f \vee m \vee q, \quad \mathbf{start} \to f \vee m \vee y,$$

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

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

算法 5.3 CTL-forget(φ, V)

Input: CTL公式φ和原子命题集V
Output: 公式集
if $\varphi \equiv \perp$ **then return** $\perp$; // 若公式不可满足，则遗忘结果为$\perp$
if $V = Var(\varphi)$ **then return** $\top$; // 若遗忘所有原子命题，则结果为$\top$
$T_\varphi \leftarrow \mathrm{SNF}^g_{\mathrm{CTL}}(\varphi)$; // 将φ转换为$\mathrm{SNF}^g_{\mathrm{CTL}}$子句
$\Sigma \leftarrow \mathrm{UF}(T_\varphi, V \cup U)$，其中$U = Var(T_\varphi) - Var(\varphi)$; // 展开
$\Sigma \leftarrow ERes(\Sigma, V)$; // 移除包含V中元素的子句
$\Sigma \leftarrow$ RM-index(Σ) ; // 从Σ移除索引
$\Sigma \leftarrow \mathrm{GAL}(\Sigma, Var(\Sigma) - Var(\varphi))$; // 移除留存的新的原子命题
用$\mathrm{AG}\varphi$替换Σ中的初始子句“$\mathrm{AG}(\mathbf{start} \to \varphi)$” ; // 去除**start**
return Σ

定理 31 (可靠性)

若 φ 为一个 CTL 公式、$V \subseteq \mathscr{A}$、$\Sigma =$CTL-*forget*(φ, V) 且 $U = Var(\Sigma) - Var(\varphi)$，则：

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

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

命题 11

给定 CTL 公式 φ 和原子命题集 $V \subseteq \mathscr{A}$。算法 5.3 的时间和空间复杂性为$\underline{O((m+1)2^{4(n+n')})}$，其中 $n = |Var(\varphi)|$、$n' = |V'|$ 为新引入的原子命题的个数、m 为引入的索引个数。

例 31 (例??的延续)

容易看出 CTL-forget$(\varphi,\{p,r\})$ 包含下面的公式

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

命题 11 (遗忘存在的子类)

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

基于归结的算法 CTL-forget 实现

系统描述

- 输入输出：基于 Prolog 的 CTL-forget 算法实现系统以 CTL 公式和原子命题集为输入，CTL 公式为输出；
- 系统识别的 CTL 公式的符号与第??章中 CTL 的语言符号对应关系如下：
  - x_i 和其余小写字母开头的字符串构成原子命题集，其中 $i \geq 0$ 为自然数，且 x_i 和 z 被设定为只能是在如下描述的转换过程中引入的原子命题；
  - “*false*”和“*true*”分别与常量符号“$\bot$”和“$\top$”对应；
  - “*start*”与命题常量“**start**”对应；
  - “&”、“\/”、“$-$”和“$->$”分别与联结符号“$\wedge$”、“$\vee$”、“$\neg$”和“$\rightarrow$”对应；
  - “~”和“^”分别与路径量词“A”和“E”对应；
  - “@”、“*”、“?”和“\$”分别与时序操作符“G”、“X”、“F”和“U”对应。

例 32

字符串 (~*((−y1\/−y2\/−y4)&(−y1\/y2\/y4)&(y1\/y2\/−y3)&(y1\/y3\/−y4)&(−y1\/y2\/−y3))) 为 CTL 公式。

基于归结的算法 CTL-forget 实现

系统主要模块

此系统主要包括五个模块[a]：

- 转换模块（transCTL2SNF/6）：
- 归结模块（两个过程：step_resolution/3 和 temp_resolution/8）
- “移除”原子命题模块（removeAtom/3）
- “移除”索引（pro6/3）
- “移除”新引入的原子命题（ackerM/3）

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

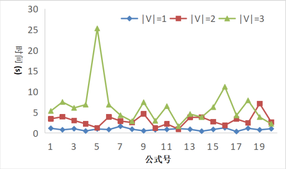
基于归结的算法 CTL-forget 实验——实验 1：计算遗忘

(1) 标准数据集来源于 CTL-RP：`https://sourceforge.net/projects/ctlrp/`

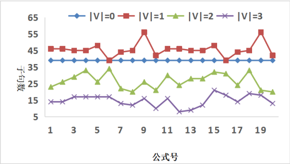
表 5.1: 计算CTL-forget(φ, V)所使用的CPU时间（单位：秒(s)）

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

(2) 计算 CTL-forget(φ, V) 使用的时间和在“移除原子命题”步骤后 $\mathrm{SNF}^g_{\mathrm{CTL}}$ 子句的个数，其中 $\varphi = \varphi_1 \wedge \mathrm{AX}\varphi_2 \wedge \mathrm{EX}\varphi_3$，$\varphi_i = 12$ $(i = 1, 2, 3)$。



(a) 计算遗忘需要的 CUP 时间

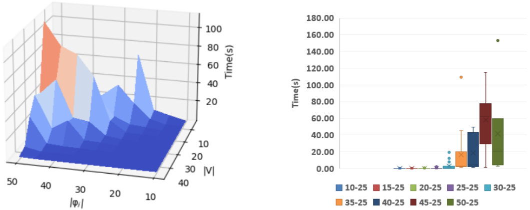


(b) $\mathrm{SNF}^g_{\mathrm{CTL}}$ 子句的个数

基于归结的算法 CTL-forget 实验——实验 2：计算 SNC

计算 q 在 V 和 $\varphi \wedge q$ 上的 SNC（$\mathrm{F}_{\mathrm{CTL}}(\varphi \wedge q, Var(\varphi) - V \cup \{q\})$），其中 $V \subseteq Var(\varphi)$、$q \in Var(\varphi \wedge q) - V$。

(1) 随机 3-CNF，$|\mathscr{A}| = 50$，每组 20 个公式。



(c) 平均 CPU 时间 (s)

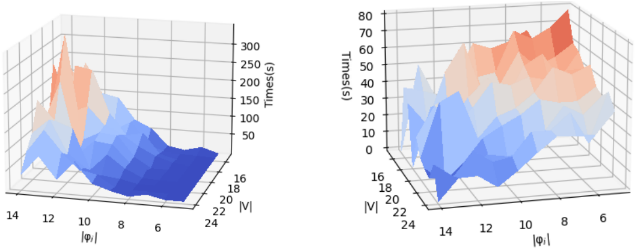
(d) $|V| = 25$ 时所使用 CPU 时间箱线图

图 10: 计算 3-CNF 公式 SNC 的 CPU 时间

总结：基于归结的算法大多数情况下能计算出 SNC（WSC），且当需要遗忘的原子个数很少或公式长度较小时计算效率较高。

计算 q 在 V 和 $\varphi\wedge q$ 上的 SNC（$\mathrm{F}_{\mathrm{CTL}}(\varphi\wedge q, Var(\varphi)-V\cup\{q\})$），其中 $V\subseteq Var(\varphi)$、$q\in Var(\varphi\wedge q)-V$。

(2) CTL 公式 $\varphi=\varphi_1\wedge \mathrm{AX}\varphi_2\wedge \mathrm{EX}\varphi_3$，$\varphi_i=12$ ($i=1,2,3$) 为 $|\mathscr{A}|=50$ 上的 3-CNF 且 $|\varphi_1|=|\varphi_2|=|\varphi_3|$，每组 40 个公式。



(a) 计算遗忘需要的 CUP 时间

(b) $\mathrm{SNF}^{g}_{\mathrm{CTL}}$ 子句的个数

图 10: 计算 CTLSNC 的平均时间和存在 SNC 的公式占比

总结： 基于归结的算法大多数情况下能计算出 SNC（WSC），且当需要遗忘的原子个数很少或公式长度较小时计算效率较高。

目录

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

总结

- CTL 和 μ-演算的遗忘理论
  - CTL 的遗忘理论：基本性质（表达性理论、代数属性和封闭性等）
  - μ-演算的遗忘理论：基本性质、复杂性和互模拟不变性等
- 遗忘理论在反应式系统的形式化验证和知识更新中的应用
  - 计算 WSC 和 SNC：定义、基本性质和基于遗忘的计算方法等
  - 定义知识更新：两种知识更新定义和基本性质
- 计算 CTL 遗忘的算法
  - 基于模型的计算方法：有界互模拟、特征值、特征公式、封闭性、复杂性和算法等
  - 基于消解（resolution）的计算方法：算法及其可靠性、遗忘存在的子类
  - 实现与实验分析

展望

- CTL 和 -演算的遗忘
  - 遗忘结果总是存在的子类；
  - 遗忘相关问题复杂性分析；
  - CTL 和 μ-演算遗忘之间的关系。
- “CTL 和 -演算公式的遗忘结果是否分别是 CTL 和 -演算可表示”这一问题的可判定性研究
- 遗忘与 WSC（SNC）之间的相互关系与应用

基于遗忘的反应式系统最弱充分条件研究

绪论
研究背景和意义
国内外研究现状
研究目标
研究内容及拟解决的关键科学问题
背景知识
Kripke 结构
CTL 的语法和语义
μ-演算
CTL 和 μ-演算遗忘理论
CTL 遗忘理论
μ-演算遗忘理论
遗忘理论在反应式系统中的应用
简介
最弱充分条件
知识更新
CTL 遗忘计算方法
简介
基于模型的有界 CTL 遗忘计算
基于归结的遗忘计算方法
基于归结的算法
CTL-forget 实现及实验
总结与展望
总结
展望
参考文献

作者在攻读博士学位期间参与项目及成果

- dd
- de
- 3e

敬请各位老师批评指正
谢谢!

参考文献 I

[1] Michael C. Browne, Edmund M. Clarke, and Orna Grümberg. "Characterizing finite Kripke structures in propositional temporal logic". In: Theoretical Computer Science 59.1-2 (1988), pp. 115–131.

[2] Giovanna D'Agostino and Marco Hollenberg. "Logical Questions Concerning The μ-Calculus: Interpolation, Lyndon and Los-Tarski". In: The Journal of Symbolic Logic 65.1 (2000), pp. 310–332. DOI: 10.2307/2586539. URL: https://doi.org/10.2307/2586539.

[3] Giovanna D'Agostino and Marco Hollenberg. "Uniform interpolation, automata and the modal μ-calculus". In: Logic Group Preprint Series 165 (1996).

参考文献 II

[4] Giovanna D'Agostino and Giacomo Lenzi. "On modal μ-calculus with explicit interpolants". In: Journal of Applied Logic 4.3 (2006), pp. 256–278. DOI: 10.1016/j.jal.2005.06.008. URL: https://doi.org/10.1016/j.jal.2005.06.008.

[5] Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas. "Computing Strongest Necessary and Weakest Sufficient Conditions of First-Order Formulas". In: Proceedings of IJCAI'01. Ed. by Bernhard Nebel. Morgan Kaufmann, 2001, pp. 145–154. ISBN: 1-55860-777-3.

[6] Dexter Kozen. "Results on the Propositional μ-Calculus". In: Theoretical Computer Science 27 (1983), pp. 333–354. DOI: 10.1016/0304-3975(82)90125-6. URL: https://doi.org/10.1016/0304-3975(82)90125-6.

参考文献 III

[7] Fangzhen Lin. "Compiling causal theories to successor state axioms and STRIPS-like systems". In: Journal of Artificial Intelligence Research 19 (2003), pp. 279–314.

[8] Fangzhen Lin. "On strongest necessary and weakest sufficient conditions". In: Artificial Intelligence 128.1-2 (2001), pp. 143–159. DOI: 10.1016/S0004-3702(01)00070-4. URL: https://doi.org/10.1016/S0004-3702(01)00070-4.

[9] Fangzhen Lin and Ray Reiter. "Forget It!" In: In Proceedings of the AAAI Fall Symposium on Relevance. New Orleans, US, 1994, pp. 154–159.

[10] Larisa Maksimova. "Temporal logics of "the next" do not have the beth property". In: Journal of Applied Non-Classical Logics 1 (1991), pp. 73–76.

参考文献 IV

[11] Lan Zhang, Ullrich Hustadt, and Clare Dixon. "A resolution calculus for the branching-time temporal logic CTL". In: ACM Transactions on Computational Logic (TOCL) 15.1 (2014), pp. 1–38.

[12] Lan Zhang, Ullrich Hustadt, and Clare Dixon. First-order Resolution for CTL. Tech. rep. Technical Report ULCS-08-010, Department of Computer Science, University of Liverpool, 2008.

[13] Yan Zhang and Yi Zhou. "Knowledge forgetting: Properties and applications". In: Artificial Intelligence 173.16-17 (2009), pp. 1525–1537.