用于信息物理系统可判定验证的同步信号时序逻辑

Partha Roop Sobhan Chatterjee Avinash Malik 奥克兰大学 UniversityofAuckland
UniversityofAuckland 新西兰奥克兰 Auckland,NewZealand Auckland,NewZealand
p.roop@auckland.ac.nz sobhan.chatterjee@auckland.ac.nz avinash.malik@auckland.ac.nz

Nathan Allen 奥克兰理工大学 Auckland, NewZealand
nathan.allen@aut.ac.nz

Logan Kenwright
奥克兰大 f Auckland
新西兰奥克兰
logan.kenwright@auckland.ac.nz

arXiv:2603.25531v1 [cs.FL] 26 Mar 2026

摘要
许多信息物理系统（CPS）运行于安全关键环境中，在此环境下，正确执行、可靠性和可信性至关重要。信号时序逻辑（STL）为检查安全关键信息物理系统提供了形式化框架。然而，一般而言，STL 的静态验证是不可判定的；只有采用基于运行时的方法进行验证时才可行，但这类方法存在局限性。我们提出同步信号时序逻辑（SSTL），这是 STL 的一个可判定片段，支持静态安全性和活性性质验证。在 SSTL 中，我们假设信号以固定的离散步长进行采样，这些步长称为时钟刻；随后提出信号不变性假设（SIH），该假设受到同步程序中类似假设的启发。我们定义了 SSTL 的语法和语义，并证明 SIH 是 STL 公式与其对应 SSTL 公式之间等价的充要条件。通过将 SSTL 转换为 $\mathrm{LTL}_{\mathcal{P}}$（定义在谓词上的 LTL），我们能够使用 SPIN模型检查器进行可判定的模型检测。我们在一个 33节点人体心脏模型及其他案例研究中展示了该方法。

关键词
SSTL、时序逻辑、静态验证、模型检测、活性性质、安全性质

1 引言

农业机器人、自动驾驶车辆、制造单元等自主系统，是信息物理系统（CPS） [2], 的一个子类，其中（通常）离散且分布式的控制器控制着连续的被控对象。此类系统运行在人类存在的环境中。因此，确保其安全运行至关重要。人们已提出许多不同的自主系统安全保障技术。这些技术可分为① 静态验证 [1], , 它确保系统在部署前能够安全运行。然而，众所周知，即使在简单情况下，信息物理系统的静态验证也是不可判定的 [14]。② 信息物理系统的运行时验证 [9, 12], , 即验证引擎在已部署的系统上检查所需属性。运行时验证是静态验证的补充，因为它不会枚举系统所有可能的执行路径。 ③ 控制器综合 [6], 则是在满足形式化属性的同时，为给定的被控对象生成控制器。控制器综合问题，该问题被表述为一个约束优化问题，但在大型系统中存在可扩展性问题。

无论选择哪种方案，信息物理系统的形式化属性都使用适当的时序逻辑来规约，其中谓词定义在连续变量之上。信号时序逻辑（STL） [10] 是自主信息物理系统属性规约中应用最广泛的时序逻辑。本文研究了静态 [17] 和运行时验证 [3, 22] ，以及控制器综合 [18, 20] 。STL属性的静态验证是不可判定的 [4]。对系统轨迹进行过近似可以在一定程度上缓解这一问题。然而，这种过近似仅适用于安全性质（坏事永远不会发生）。据我们所知，以STL表达的活性（即好事最终会发生）性质从未在自主系统中经过静态验证。本文旨在应对STL安全性和活性属性验证中不可判定性的挑战。这些方法对于验证环境约束至关重要，能够使安全自主成为现实。

我们关注环境输入和被控对象输出，它们以时间序列数据的形式被感知，也称为信号。例如，考虑一个33节点心脏模型场景，我们关注心脏传导系统中的两个关键节点：窦房结（SA结）和房室结（AV结）。在任意时刻，心脏的状态都可以用一个向量表示，该向量包含这两个节点处的电势。形式上，状态向量 $\vec{x}$ 是一个映射，$\vec{x}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^2$，其定义域表示任意时刻，值域则给出该时刻SA结和AV结处的电势对。此处，时间连续演进，其定义域为$\mathbb{R}_{\geq 0}$，也称为实时系统。在实时场景中，信号验证是不可判定的。此处，我们考虑一种替代方案，即离散时间 场景，在其中时间按离散步进演进，其定义域为 $\mathbb{N}_{\geq 0}$。这种方法用于许多实际系统，包括所有在离散计算机上实现的系统，例如通过对某个输入进行采样来实现。称为同步编程 [7], 的一种编程范式，基于在称为时钟刻的离散时刻对环境进行采样。采用同步形式来离散化信号时序逻辑的理由如下：

（1）来自物理源的任何信号都具有与其质量成正比的惯性。例如，心电图传感器就是如此。这里，在表示心房去极化（心跳）的 P 波之后，会有一段间隔

然后才出现下一个事件 Q，它表示心室去极化的开始。

（2）在信号处理领域，除了对连续信号进行变换外，也有针对离散版本的变换。以傅里叶变换（FT）和离散傅里叶变换（DFT）为例。FT 适用于对连续时间信号进行理论分析，而 DFT 则适用于针对采样信号设计计算算法。因此，我们认为需要一种 STL 的同步变体，将逻辑本身离散化。

（3）在广泛用于信息物理系统的同步编程中，会使用一种假设来约束环境。这称为同步假设，其含义是相对于被控对象，控制系统的响应速度无限快。换言之，它要求传感器的数据到达速率不超过控制器处理数据的速率。在我们的研究中，可以利用这一假设来形式化 STL 与其抽象之间的关系，我们将其称为同步信号时序逻辑（*SSTL*），尤其用于形式化两个采样点之间信号的状态。

基于上述内容，我们的前提是：一种 STL 的离散变体 SSTL，将 STL 逻辑离散化并专为同步系统设计，可为信息物理系统的可扩展验证带来方法。我们还认为，在适当的鲁棒性 [13], 假设下，SSTL 是 STL 的可靠且完备的抽象。本文的主要贡献如下：

(1) 我们提出了首个 STL 逻辑的同步（离散）抽象，称为 SSTL。我们引入了一种可判定的信号时序逻辑（STL）编码，称为同步信号时序逻辑（SSTL）。(2) 我们定义了 SSTL 的语法和语义，并证明在对信号特性作出适当假设的情况下，它可作为 STL 的可靠且完备的抽象。(3) 我们给出了 SSTL 的 $\mathrm{LTL}_{\mathcal{P}}$ 翻译，并利用它通过 SPIN模型检查器对 SSTL 性质进行模型检测。这使 SSTL 性质的模型检测问题变得可判定且实用。(4) 另一项重要贡献是使用形式时序逻辑对可靠的人类心脏模型进行形式化，并使用 SPIN 对其进行验证。

本文其余部分安排如下。第 2 节介绍一个心脏模型示例，以说明离散时序逻辑规约面临的挑战和要求。第 3 节对 SSTL 进行形式化定义，包括其语法、语义，以及能够基于 STL 实现可靠且完备抽象的信号不变性假设（SIH）。第 4 节介绍我们对 SSTL 性质的验证。第5节介绍我们对 SSTL 性质的评估。第 6 节讨论时序逻辑变体，以及信息物理系统的验证方法。最后，第 7 节总结我们的贡献，并探讨未来的研究方向。

2 动机示例

在本节中，我们将介绍一个基于33节点人体心脏模型的案例研究。该模型具有代表性，可用于数字医疗系统，其中安全性和活性性质至关重要。

本案例研究突显了连续时间时序逻辑规约（如信号时序逻辑）的局限性，并说明了为何需要一种适用于数字化实现的离散同步变体。

2.1 心脏模型案例研究

经过充分测试和验证的心脏模型，对于开发安全可靠的数字医疗系统至关重要，例如心脏起搏器和除颤器。通常，需要一个能够准确模拟人体心脏及其可能患有的疾病的心脏模型。然而，心脏模型及其电学特性的复杂性，使得使用传统模型检验技术验证其性质颇具挑战。在此，我们采用一个含 33 个节点的心脏模型，说明如何使用 STL 验证连续时间信号上的性质所面临的挑战，并展示如何利用 SSTL 框架验证该心脏模型的性质。

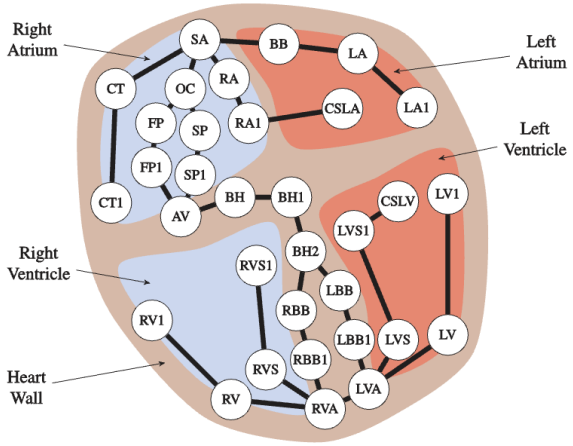


图1：33节点心脏模型。（改编自 [21]）

33节点心脏模型 [21] 通过表示心房、房室结和心室区域的互连节点来模拟心脏的电传导（见图1）。电脉冲从窦房结（SA结）传至房室结（AV结），再沿束支传至心室。主要关注的信号是心房和心室电图（A_心电图和V_心电图），以及各节点的动作电位，这些信号均随时间进行跟踪。

为描述并验证这些信号的时序要求，我们使用信号时序逻辑，它允许定义形式化属性，例如要求心室激活（V_心电图）在给定时间窗口内跟随心房激活（A_心电图），从而确保心律符合生理安全要求。

2.2 HeartModel 的信号时序逻辑规范

为展示信号时序逻辑在心脏模型中的应用，我们设计了四项关键策略，用以体现心脏电活动的基本安全性和性能要求。这些规约展示了数字医疗系统中必须验证的时序属性类型。

属性 *1*：房室传导安全性。第一项策略确保房室传导时序恰当，这对于维持协调的心律至关重要：

$$\varphi_{AV} := \square\left(A_{EGM} > V_{a,th} \rightarrow \Diamond_{[0.180,0.240]}(V_{EGM} > V_{v,th})\right) \quad (1)$$

该规约要求，每当发生心房激活（A_{EGM}超过阈值$V_{a,th}$）时，必须在 180ms 至 240ms 内随后发生心室激活（V_{EGM}超过阈值 $V_{v,th}$）。此性质可确保房室结将电脉冲从心房正确传导至心室，避免危险的延迟。

性质 *2*：心室不应期。第二项策略强制执行心室不应期，以防止可能导致心律失常的心室过早收缩：

$$\varphi_{refractory} := \square\left(V_{EGM} > V_{v,th} \rightarrow \Diamond_{[0.600,1.00]}(V_{EGM} \leq V_{v,th})\right) \quad (2)$$

该规约确保每次心室激活后，心室心肌在心室激活后的 600ms 至 1000ms 内进入不应区（V_{EGM}低于阈值）。此性质可防止发生危险的快速心室节律，并确保心室充盈有充足时间。

性质 *3*：活性性质。第四项性质确保心脏模型永远不会陷入死锁。它最终会发出心房激活和心室激活。

$$\varphi_{liveness_A} := \Diamond(A_{EGM} > V_{a,th}) \quad (3)$$

$$\varphi_{liveness_V} := \Diamond(V_{EGM} > V_{v,th}) \quad (4)$$

示例 *2.1.* 考虑以下用于描述房室传导特性的信号时序逻辑公式：

$$\varphi_{AV} = \square\left(A_{EGM} > V_{a,th} \rightarrow \Diamond_{[0.180,0.240]} V_{EGM} > V_{v,th}\right) \quad (5)$$

该公式表示：“每当发生心房激活时” ($A_{EGM} > V_{a,\ th}$)，“接下来的 *180ms* 到 *240ms* 内必须发生心室激活” ($V_{EGM} > V_{v,\ th}$)。

现在，考虑一个连续信号，其中 A_{EGM}和 V_{EGM}对每个时间 $t \in [0,\infty)$均有定义。要在 $t = 0.625$秒时评估 φ_{AV}，此时 $A_{EGM} > V_{a,\ th}$变为真，我们必须验证之后的某个时刻（在 180ms 到 240ms 内，即 $[805, 865]$ 毫秒）$V_{EGM} > V_{v,\ th}$为真。图2a展示了这一过程。假设 $V_{a,\ th}$和 $V_{v,\ th}$均为 80 mV。如图所示，心房激活发生在 $t = 0.625$ 秒，而心室激活发生在约 835 毫秒处，处于 180ms 到 240ms 的范围内。因此，该信号时序逻辑公式在 $t = 0.625$ 秒时满足。

2.3 STL面临的挑战

尽管信号时序逻辑（STL）提供了一种用于描述时序性质的形式化语言，但在数字医疗系统中实际应用时仍存在若干局限。

连续时间与离散时间：信号时序逻辑（STL）基于连续时间定义，而实际应用中，生理信号是在离散区间上采样得到的。

难以处理的验证：在信号时序逻辑中检查公式 1 这样的性质，需要评估实数区间内的所有点（见图2a），这使得一般验证变得不可判定且计算上难以实现 [4]。

采样问题：离散采样会带来一些挑战：*(i)* 关键事件遗漏：违反安全性质的事件可能发生在采样点之间，从而未被检测到。*(ii)* 时序算子的含义不明确：“在 80ms 内”($\Diamond_{[0,0.08]}$) 这类区间的含义取决于采样率，可能导致评估结果不一致。

3 同步信号时序逻辑（SSTL）

本节将介绍 SSTL，这是一种专为在离散时间下运行的信息物理系统（CPS）设计的同步版 STL。我们的方法是扩展 STL，以处理通过物理系统同步执行获得的信号；我们将其称为 SSTL。在 SSTL 中，时间以离散时刻或时钟周期推进，轨迹则在时钟刻边界处获得。由于大多数物理信号在 $\mathbb{R}_{\geq 0}$ 上是连续且实值的，要使用 SSTL 进行验证，我们需要将实时时间变量 t 映射为离散时间变量 $[t]$。这种映射与我们的 SIH 相结合，确保我们能够推理离散时间行为，同时保证相对于底层连续系统的可靠性。

定义 *3.1*。（时间离散化）设 $t \in \mathbb{R}_{\geq 0}$为一个实时变量，$\Delta t \in \mathbb{R}_{\geq 0}$表示实时环境中的时钟周期长度。于是，我们可以定义 $[t]: \mathbb{R}_{\geq 0} \rightarrow \mathbb{N}_{\geq 0}$如下：

对于任意 $k \geq 0$ $k \in \mathbb{N}_{\geq 0}$ $t \in [k\Delta t,\ k+1\ \Delta t \implies [t] = k = \lfloor t\ \Delta t \rfloor$ 和，我们有，（）（）/。因此，对于任意区间，其离散时间区间为 $[[a],[b]] \in \mathbb{N}_{\geq 0}$，其中 $[a] = \lfloor a/\Delta t \rfloor$ 且 $[b] = \lfloor b/\Delta t \rfloor$。 $[a,b] \in \mathbb{R}_{\geq 0}$

□

定义 *3.2*。（SSTL 轨迹）SSTL 轨迹 w_d是一个函数 w_d：$\mathbb{N}_{\geq 0} \rightarrow \mathbb{R}^n$，将离散时间点映射到信号值。对于给定的轨迹 w_d和离散时间 $[t] \in \mathbb{N}_{\geq 0}$，$x_i^{w_d}([t])$表示轨迹 w_d中信号 x_i在时间 $[t]$处的值，其中 $i = 1,\ 2,\ \ldots\ldots,\ n$。 □

定义 *3.3.*（信号不变性假设（SIH））时间离散化操作的结果是，信号在相邻两个离散时刻之间保持不变。具体而言，信号的值在任意单个时刻区间内均不会发生变化。形式上，对于所有 $t, t' \in [k\Delta t, (k+1)\Delta t)$，其中 $t, t' \in \mathbb{R}_{\geq 0}$和 $k \in \mathbb{N}_{\geq 0}$，均有：

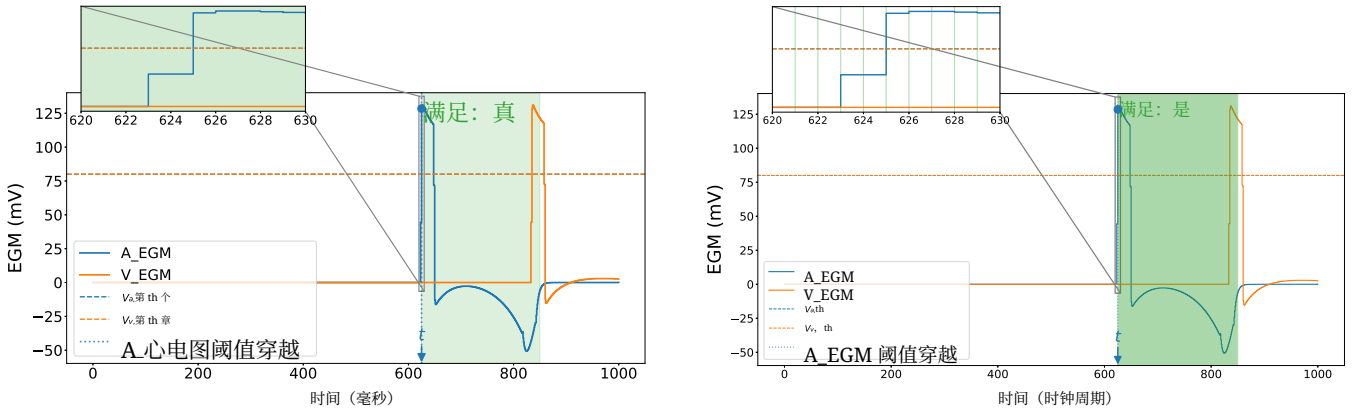
$$x_i(t) = x_i(t') \quad \text{and} \quad [t] = k. \quad (6)$$

因此，对于任意 $t \in [k\Delta t, (k+1)\Delta t)$，我们有

$$x_i^{w_d}([t]) = x_i^{w}(t). \quad (7)$$

SIH 对可靠性至关重要，因为它保证了底层连续信号的变化不会导致离散时刻之间的信息丢失。 □

注释 1. 更严格的 *SIH* 形式要求输入信号（来自传感器或连续过程）的采样频率满足奈奎斯特准则，从而使离散迹成为连续信号的可靠抽象。数学上，$F_s \geq \kappa F_m$，其中$F_s = 1/\Delta t$是采样频率，F_m是输入信号的最高频率分量（或带宽），且 $\kappa \in \mathbb{N}_{\geq 2}$。以该频率或更高频率进行采样，可以保留信号的关键特征（避免混叠），并确保连续信号中不会发生任何相关变化



(a) 在 秒处对信号时序逻辑公式 ($\varphi = \square\, A_{EGM} > V_{a\,th} \rightarrow (,(,)))$ **求值。** $\Diamond_{[0.180,0.240]}\ V_{EGM} > V_{v\,th}$ $t = 0.625$

(b) 在 $t = 625$ **个时钟周期处对 SSTL 公式 (**$[\varphi] = \square(A_{EGM} > V_{a,\,th} \rightarrow \Diamond_{[[0.180],[0.240]]}(V_{EGM} > V_{v,th}))$**) 求值。**

图 2：心脏模型场景中信号时序逻辑与 SSTL 的比较。

在 两个连续采样之间，而仅在时钟周期时刻评估谓词会漏掉这些变化。因此，分段常值表示（在每个时刻区间内保持恒定）不会丢失满足 *SSTL* 公式所需的信息，*SIH* 也因此得以成立。因此，维持奈奎斯特率对于连续时间性质的可靠离散时间验证至关重要。

通过适当选择 κ，*SSTL* 性质中的区间边界可以与连续时间信号边界对齐，使采样时刻与信号的自然时序一致，且离散表示与连续表示保持一致。对于每个时钟刻输出单个值的离散 过程（例如周期为 T 的时钟或事件生成器），信号仅在这些时刻有定义；选择使 Δt 等于或为 T 的整数倍，可确保过程以其自然频率采样，并且 *SIH* 按定义得到满足，因为时钟刻之间不存在底层连续信号。 □

定义 *3.4*。 如果 φ 是任意一个 STL 公式，则相应的 SSTL 公式 $[\varphi]$ 递归定义如下：

$$[\varphi] := \top \mid [x_i^{w_d}(t) \geq 0] \mid \neg[\varphi] \mid [\varphi_1] \wedge [\varphi_2] \mid [\varphi_1]\mathcal{U}_{[[a],[b]]}[\varphi_2] \tag{8}$$

其中， $a \in \mathbb{R}_{\geq 0}$，且 $b \in \mathbb{R}_{\geq 0}$。 □

定义 *3.5.*（SSTL 布尔语义）对于 SSTL 轨迹 w_d和离散时间 $[t]$，布尔语义递归定义如下：

$$\begin{aligned}
&(w_d, [t]) \models \top \\
&(w_d, [t]) \models [x_i^{w_d}(t) \geq 0] \quad &\iff\ & x_i^{w_d}([t]) \geq 0 \\
&(w_d, [t]) \models \neg[\varphi] \quad &\iff\ & (w_d, [t]) \not\models [\varphi] \\
&(w_d, [t]) \models [\varphi_1] \wedge [\varphi_2] \quad &\iff\ & (w_d, [t]) \models [\varphi_1] \wedge (w_d, [t]) \models [\varphi_2]
\end{aligned}$$

$$\begin{aligned}
(w_d, [t]) \models [\varphi_1]\mathcal{U}_{[[a],[b]]}[\varphi_2] \iff\ & \exists\, [t'] \in [[t+a], [t+b]] : \\
& (w_d, [t']) \models [\varphi_2] \\
& \text{and } \forall\, [t''] \in [[t], [t']) : (w_d, [t'']) \models [\varphi_1]
\end{aligned} \tag{9}$$

我们可以将其他常用算子重新定义为语法缩写：

$$[\varphi_1] \vee [\varphi_2] := \neg(\neg[\varphi_1] \wedge \neg[\varphi_2]) \tag{10}$$

$$[\varphi_1] \implies [\varphi_2] := \neg[\varphi_1] \vee [\varphi_2] \tag{11}$$

$$\Diamond_{[[a],[b]]}[\varphi] := \top\mathcal{U}_{[[a],[b]]}[\varphi] \tag{12}$$

$$\square_{[[a],[b]]}[\varphi] := \neg\Diamond_{[[a],[b]]}\neg[\varphi] \tag{13}$$

无时间算子（如 U、 ◊ 和 □）只是区间为 $[0, +\infty)$ 的有界算子的简写。因此，它们与有界算子具有相同的语义。

定理 3.6。 如果轨迹 w^1 包含满足 *SIH* 的信号，且该轨迹也满足 *STL* 公式 φ，那么对于任意 $t \in \mathbb{R}_{\geq 0}$和 $[t] \in \mathbb{N}_{\geq 0}$，有：

$$(w, t) \models \varphi \iff (w_d, [t]) \models [\varphi]$$

该定理的证明见附录 *B*。

例 *3.7*。 考虑公式 1 中给出的心脏模型性质。这里， A_{EGM}和 V_{EGM}信号均满足 SIH，因为它们以固定频率 $F_s = 1\ \Delta t = /\ \text{Hz}$ 生成和采样。图 2 中展示了针对该性质，STL 与 SSTL 的求值差异。假设我们在 $t_0 = 0.625$ 秒时刻（蓝色点线所示）评估 STL 公式是否满足，此时$A_{EGM} > V_{a,\ th}$ 变为真。对于 STL，我们必须检查：如果在时间 t_0 处， $A_{EGM}(t_0) > V_{a,\ th}$，那么必须存在$t' \in [t_0 + 0.180, t_0 + 0.240]$，使得 $V_{EGM}(t') > V_{v,\ th}$。满足性检查涉及该实值区间中的每一个点。可以看出，该 STL 公式在 $t_0 = 0.625$ 秒时满足。

对于 SSTL 情况，假设信号以固定间隔 $\Delta t = 0.001$ 秒采样（例如每个时钟刻为 1 毫秒），并且我们正在评估

[1]有关信号时序逻辑轨迹、语法和语义的定义，请参见附录 A。

时钟刻 $\lceil t_0 \rceil = \lfloor 0.625/0.001 \rfloor = 625$ 处。SSTL 公式的满足性检查

$$[\varphi_{AV}] := \square \left(A_{EGM} > V_{a,th} \rightarrow \Diamond_{[[0.180],[0.240]]} (V_{EGM} > V_{v,th}) \right)$$

需要将公式的时间区间投影到离散域。我们需要检查区间 $[625+\lfloor 0.180/0.001 \rfloor, 625+\lfloor 0.240/0.001 \rfloor] = [625+180, 625+240] = [805, 865]$ V_{EGM} / 中的每个时钟刻，看其是否超过 $V_{v,th}$，前提是 A_{EGM} 在 $\lceil t_0 \rceil$ 超过 $V_{a,th}$。我们可以看到，SSTL 公式在 $t_0 = 0.625$ 秒或 $\lceil t_0 \rceil = 625$ 个时钟周期处满足。

如果 SIH 性质成立（即信号在相邻两个离散时刻之间保持不变），那么 SSTL 公式的满足与 STL 公式的满足等价。这是因为每个离散时刻上的满足检查都与对应实值区间上的满足检查相匹配，反之亦然。此外，虽然 STL 的求值需要检查实值区间中的每个点，但 SSTL 的求值只需检查每个离散时刻（图中以绿色竖线表示）。

4 SSTL 性质验证

为了通过模型检测实现对 SSTL 公式的自动验证，我们提出将 SSTL 翻译为 LTL$_{\mathcal{P}}$，然后使用 SPIN 模型检查器在基于状态的模型和运动学模型上进行验证。我们还展示了 LTL$_{\mathcal{P}}$ 翻译如何构成 SSTL 的可靠且完备的抽象。

4.1 带谓词的线性时序逻辑（LTL）

为了将 SSTL 公式翻译为适用于模型检测的形式，我们采用带谓词的线性时序逻辑（LTL$_{\mathcal{P}}$），其灵感来自 [16], 中的工作；该工作通过整合定义在实值变量上的谓词扩展了传统 LTL。LTL$_{\mathcal{P}}$ 逻辑使我们能够表达时序属性，同时保留定义在连续值或离散值信号上的谓词。

定义 *4.1.*（LTL$_{\mathcal{P}}$ 语法）LTL$_{\mathcal{P}}$ 公式 ϕ 的语法定义如下：

$$\phi ::= \top \mid P \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2 \tag{14}$$

$$P ::= a_1 y_1 + a_2 y_2 + \ldots + a_n y_n \bowtie b \tag{15}$$

其中 P: $\mathbb{R}^n \rightarrow \mathbb{B}$ 是一个谓词符号，表示变量 $y_1, y_2, \ldots, y_n$ 上的关系条件。具体而言，P 的形式为 $a_1 y_1 + a_2 y_2 + \ldots + a_n y_n \bowtie b$，其中 $a_1, \ldots, a_n$ 是实值系数，b 是偏移量，$\bowtie$ 是从 $\{<, \leq, =, \geq, >\}$ 中选取的关系运算符。

时序算子：

⦿ϕ（下一步）：ϕ 在下一个状态成立

- $\phi_1 \mathrm{U} \phi_2$（直到）：ϕ_1 一直成立，直到 ϕ_2 变为真

运算符 $\square$ 和 $\Diamond$ 可以定义为语法缩写：

$$\Diamond\phi := \top \mathcal{U} \phi \tag{16}$$

$$\square\phi := \neg\Diamond\neg\phi \tag{17}$$

定义 *4.2*。（LTL$_{\mathcal{P}}$ 语义）：LTL$_{\mathcal{P}}$ 轨迹 σ 是一个状态的无限序列，其中每个状态 s_j，$\forall j \in \mathbb{N}_{\geq 0}$ 都为变量 $y_1, \ldots, y_n$ 赋值。对于状态 s_j，我们用 $s_j(y_1), \ldots, s_j(y_n)$ 表示状态 s_i 中变量 $y_1, y_2, \ldots, y_n$ 的取值。满足关系 $(\sigma, j) \models \phi$（读作“σ 在位置 j 满足 ϕ”）递归定义如下：

$$(\sigma, j) \models P \iff a_1 s_j(y_1) + a_2 s_j(y_2) + \ldots + a_n s_j(y_n) \bowtie b \tag{18}$$

$$(\sigma, j) \models \neg\phi \iff (\sigma, j) \not\models \phi \tag{19}$$

$$(\sigma, j) \models \phi_1 \wedge \phi_2 \iff (\sigma, j) \models \phi_1 \text{ and } (\sigma, j) \models \phi_2 \tag{20}$$

(21)

$$(\sigma, j) \models \phi_1 \vee \phi_2 \iff (\sigma, j) \models \phi_1 \text{ or } (\sigma, j) \models \phi_2 \tag{22}$$

$$(\sigma, j) \models \bigcirc\phi \iff (\sigma, j+1) \models \phi \tag{23}$$

$$(\sigma, j) \models \phi_1 \mathcal{U} \phi_2 \iff \exists k \geq j \text{ such that } (\sigma, k) \models \phi_2 \tag{24}$$

$$\text{and } \forall l \in [j, k), \sigma, l \models \phi_1 \tag{25}$$

$$(\sigma, j) \models \square\phi \iff \forall k \geq j, (\sigma, k) \models \phi \tag{26}$$

$$(\sigma, j) \models \Diamond\phi \iff \exists k \geq j, (\sigma, k) \models \phi \tag{27}$$

如果序列 σ 满足 LTL$_{\mathcal{P}}$ 公式 ϕ（记作 $\sigma \models \phi$），则称之为如此，当且仅当 $\sigma, 0 \models \phi$。

示例 *4.3*。考虑以下涉及两个变量 y_1 和 y_2 的 LTL$_{\mathcal{P}}$ 公式（其中 y_1 是以摄氏度为单位的温度，x，而 y_2 是以伏特为单位的电压 v）：

$$\phi := 2y_1 - y_2 \geq 10$$

也就是说，如果温度的 2 倍减去电压大于或等于 10，则 ϕ 在某个状态下为真。

现在，假设我们有一条 LTL$_{\mathcal{P}}$ 轨迹 $\sigma = s_0, s_1, s_2, \ldots$，其中每个状态都为这些变量赋值。

以状态 s_5 为例，其中 $s_5(y_1) = 15$ 和 $s_5(y_2) = 18$。将这些值代入公式，得到：

$$2 \cdot s_5(y_1) - s_5(y_2) = 2 \cdot 15 - 18 = 30 - 18 = 12$$

Si因此 $12 \geq 10$，公式 ϕ 在状态 s_5 处成立，即 $(\sigma, 5) \models \phi$。

4.2 SSTL 到 LTL 的翻译

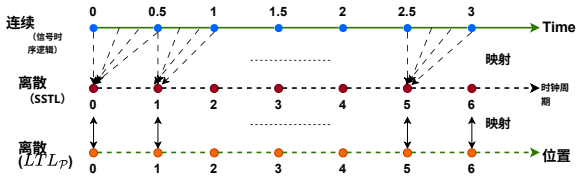


图 3：连续时间与离散时间之间映射的示意图，假设信号以固定间隔 $\Delta t = 0.5$ 秒采样

定义 *4.4*。（SSTL 到 LTL$_{\mathcal{P}}$ 时间双射）我们将 SSTL 轨迹中的位置与 LTL$_{\mathcal{P}}$ 轨迹之间的对应关系形式化如下。设 $w_d = (x_0^{w_d}, x_1^{w_d}, \ldots\ldots)$ 是一条离散时间信号轨迹，其中 $x_k^{w_d}$ 是离散时刻 $k \in \mathbb{N}_{\geq 0}$ 的取值。设 $\sigma = (s_0, s_1, \ldots\ldots)$ 是对应的 LTL$_{\mathcal{P}}$ 轨迹，其中每个 s_j 给出位置 j 处的状态。根据定义 3.1，我们定义一个映射 π: $\mathbb{N}_{\geq 0} \rightarrow \mathbb{N}_{\geq 0}$，使得

$$j = \pi(\lceil t \rceil) := \lceil t \rceil \tag{28}$$

对于每个时钟刻 $[t]$。也就是说， w_d中的第 $x_{[t]}^{w_d}$个取值对应于 σ中的第j个位置，从而在每个时钟刻建立两条轨迹之间的双射。形式化地，

$$\forall [t] \in \mathbb{N}_{\geq 0}, \quad x[t]^{w_d} \leftrightarrow s_j$$

因此，SSTL 公式 $[\varphi]$ 在时间 $[t]$ 上的满足，等价于 SSTL 公式 $[\varphi]$ 在位置 j上的满足。□

定义 *4.5*。（SSTL 到 $\text{LTL}_{\mathcal{P}}$ 函数）设 w_d 为一条 SSTL 轨迹，并设 σ 为将同一轨迹视为 $\text{LTL}_{\mathcal{P}}$ 轨迹。将 SSTL 公式的集合记为 $[\![\varphi]\!]$，将 $\text{LTL}_{\mathcal{P}}$ 公式的集合记为 $[\![\phi]\!]$。我们记作 $\tau: [\![\varphi]\!] \to [\![\phi]\!]$，表示通过翻译 SSTL 公式 $[\varphi]$得到的 $\text{LTL}_{\mathcal{P}}$ 公式 ϕ。

$$\phi = \tau([\varphi]) \tag{29}$$

□

定义*4.6*。（SSTL 到 $\text{LTL}_{\mathcal{P}}$ 函数翻译的无界区间情形）SSTL 语法各情形的翻译可分为两部分：一部分处理具有无界区间的公式（$[0, \infty]$），另一部分处理具有有界区间的公式（$[a, b]$）。根据定义 4.4，SSTL 轨迹与 $\text{LTL}_{\mathcal{P}}$ 轨迹之间存在一一对应关系。利用函数 τ 翻译无界情形十分直接，具体如下：

$$\tau\big(x_i^{w_d}([t]) \geq 0\big) := s_j(x_i^{w_d}) \geq 0 \tag{30}$$

$$\tau(\neg[\varphi]) := \neg\tau([\varphi]), \tag{31}$$

$$\tau([\varphi_1] \wedge [\varphi_2]) := \tau([\varphi_1]) \wedge \tau([\varphi_2]), \tag{32}$$

$$\tau\big([\varphi_1]\ \mathcal{U}\ [\varphi_2]\big) := \tau([\varphi_1])\ \mathcal{U}\ \tau([\varphi_2]) \tag{33}$$

$$\tau(\Box[\varphi]) := \Box\tau([\varphi]) \tag{34}$$

$$\tau(\Diamond[\varphi]) := \Diamond\tau([\varphi]) \tag{35}$$

为了处理 $\text{LTL}_{\mathcal{P}}$ 公式中的有界时间区间，我们引入位置索引 j，表示当前轨迹位置（即当前离散时钟刻）。索引 j在每一步递增 1。当一项有时间界限的义务开始时（例如，在某个时钟刻 A 成立时计算 $\Box(A \to \Diamond_{[a,b]}V)$，或在有界直到的进入时钟刻），我们将进入位置记录为 j_0，并在该项义务持续期间保持不变。如果多项义务在时间上重叠，则每项义务都有自己的（概念性） j_0 副本。这样，任意满足 $a,\ b \in \mathbb{N}_{\geq 0}$ 的区间 $[a, b]$ 都可以表示为关于 j和 j_0的谓词。我们定义守卫：

$$\text{within}[a, b] \ := \ (j_0 + a \leq j \leq j_0 + b), \tag{36}$$

这是一个在当前状态 s_j [2]上求值的普通 $\text{LTL}_{\mathcal{P}}$ 原子谓词。

引理 4.7。（j_0 副本数量的有界性）设 *SSTL* 公式在（可能无限的）离散迹上进行解释，并通过将其翻译为带时间有界算子的 $LTL_{\mathcal{P}}$ 来实现。那么，对于任意位置 j，不同活跃义务的数量（即必须在 j 跟踪的不同概念性 j_0 副本）仅取决于 j 处的活跃义务数量，而该数量由截至该位置的公式结构和轨迹决定。

[2]请参阅附录 E，了解更多详情。

证明概要见附录 D。□

基于这一思路，在时钟刻 $[t]$ 和位置 j 处求值时，有界算子的推导如下。其他算子的翻译方式与无界情形相同。

$$\tau\big([\varphi_1]\ \mathcal{U}_{[[a],[b]]}\ [\varphi_2]\big) := \tau([\varphi_1])\ \mathcal{U}\ \big(\tau([\varphi_2]) \wedge \text{within}[a, b]\big), \tag{37}$$

$$\tau\big(\Diamond_{[[a],[b]]}[\varphi]\big) := \Diamond\big(\tau([\varphi]) \wedge \text{within}[a, b]\big), \tag{38}$$

$$\tau\big(\Box_{[[a],[b]]}[\varphi]\big) := \Box\big(\text{within}[a, b] \ \to \ \tau([\varphi])\big) \tag{39}$$

示例 *4.8.* 我们通过无界和时间有界两种情形，展示 SSTL 中 Until 算子到 $\text{LTL}_{\mathcal{P}}$ 的翻译。

公式：

- *Unbounded Until:*

$$[\varphi] := x_1^{w_d}([t]) \geq 0\ \mathcal{U}\ x_2^{w_d}([t]) \geq 0$$

- *Bounded Until (with interval* $[5, 10]$*):*

$$[\varphi] := x_1^{w_d}([t]) \geq 0\ \mathcal{U}_{[[5],[10]]}\ x_2^{w_d}([t]) \geq 0$$

信号表：

$x_1^{w_d}$和 $x_2^{w_d}$在位置 $j = [t]$ 处的具体值如下：

$[t]$	0	1	2	3	4	5	6	7	8	9	10
$x_1^{w_d}$	1	1	1	0.5	0.8	0.2	1	0.5	0.2	-1	-0.7
$x_2^{w_d}$	-1	-1	-0.8	-0.6	-0.5	-0.1	-0.15	0.6	1	1	0.8

翻译：

- *Unbounded Until:*

$$\begin{aligned}\tau([\varphi]) &= \tau\big(x_1^{w_d}([t]) \geq 0\big)\ \mathcal{U}\ \tau\big(x_2^{w_d}([t]) \geq 0\big)\\ &= (s_j(x_1^{w_d}) \geq 0)\ \mathcal{U}\ (s_j(x_2^{w_d}) \geq 0)\end{aligned}$$

- 有界 *Until*:

$$\begin{aligned}\tau([\varphi]) &= \tau\big(x_1^{w_d}([t]) \geq 0\big)\ \mathcal{U}\ \big(\tau\big(x_2^{w_d}([t]) \geq 0\big) \wedge \text{within}[5, 10]\big)\\ &= (s_j(x_1^{w_d}) \geq 0)\ \mathcal{U}\ \big((s_j(x_2^{w_d}) \geq 0) \wedge \text{within}[5, 10]\big)\end{aligned}$$

where $\text{within}[5, 10] := (j_0 + 5 \leq j \leq j_0 + 10)$, ensuring that $x_2^{w_d} \geq 0$ must become true between ticks 5 and 10, counted from the start of the Until obligation.

真值表及说明：

下表给出了每个位置 j 上所有子公式、时间窗口谓词以及无界和有界直到公式的真值。

$j = [t]$	0	1	2	3	4	5	6	7	8	9	10
$\varphi_1 = x_1^{w_d}[t] \geq 0$	1	1	1	1	1	1	1	1	1	0	0
$\varphi_2 = x_2^{w_d}[t] \geq 0$	0	0	0	0	0	0	0	1	1	1	1
$\varphi_1\ \mathrm{U}\ \varphi_2$	1	1	1	1	1	1	1	1	1	0	0
在$[5, 10]$内	0	0	0	0	0	1	1	1	1	1	1
$\tau(\varphi_2)$ ∧在$[5, 10]$内	0	0	0	0	0	0	0	1	1	1	1
$\tau(\varphi_1)\ \ \mathcal{U}\ \ (\tau(\varphi_2)\ \wedge$ 在$[5, 10]$内)	0	0	1	1	1	1	0	0	0	0	0
φ_1直到$_{[5,10]}$ φ_2	0	0	1	1	1	1	0	0	0	0	0

解释：

- 前两行记录了 φ_1 ($x_1^{w_d}[t] \geq 0$) 和 φ_2 ($x_2^{w_d}[t] \geq 0$) 为真的位置。
- 第三行计算轨迹上的无界直到：一旦 φ_1 在 $j = 9$处失效，直到就变为假。
- within$[5, 10]$ 中的谓词仅在时钟周期 5 到 10 期间为真，标记出有界直到的时间窗口。
- $\tau(\varphi_2) \wedge$within$[5, 10]$ 显示了在哪些时钟周期内，第二个信号为非负且我们处于该窗口内。
- 行 $\tau(\varphi_1)$ U $(\tau(\varphi_2) \wedge$within$[5, 10])$ 给出了有界直到 LTL$_\mathcal{P}$ 翻译的满足情况。
- 最后一行 $(\varphi_1 \, \mathrm{U}_{[5,10]} \, \varphi_2)$ 与该翻译相符，表明 SSTL 语义与 LTL$_\mathcal{P}$ 翻译一致。

有界时序算子要求 $x_2^{w_d} \geq 0$ 在时钟周期 $j_0 + 5$ 和 $j_0 + 10$ 之间成立，并且在此前的每个时钟周期都满足 $x_1^{w_d} \geq 0$；否则，有界直到条件不成立。

4.2.1 公式嵌套。翻译函数 τ 以递归方式应用，以处理嵌套公式。当公式包含子公式时，会分别翻译每个子公式，并根据父公式的结构组合翻译结果。这种递归方法确保复杂的嵌套时序公式能够正确翻译为 LTL$_\mathcal{P}$。

例如，考虑一个嵌套公式，其中一个有界算子包含另一个时序算子：

$$[\varphi] := \square_{[[0],[20]]}\left(x_1^{w_d}([t]) \geq 0 \; \mathcal{U} \; x_2^{w_d}([t]) \geq 0\right)$$

翻译过程采用递归方式进行：

$$\begin{aligned}
\tau([\phi]) &= \tau\left(\square_{[[0],[20]]}\left(x_1^{w_d}([t]) \geq 0 \; \mathcal{U} \; x_2^{w_d}([t]) \geq 0\right)\right) \\
&= \square\left(\text{within}[0, 20] \rightarrow \tau\left(x_1^{w_d}([t]) \geq 0 \; \mathcal{U} \; x_2^{w_d}([t]) \geq 0\right)\right) \\
&= \square\left(\text{within}[0, 20] \rightarrow \left(\tau(x_1^{w_d}([t]) \geq 0) \; \mathcal{U} \; \tau(x_2^{w_d}([t]) \geq 0)\right)\right) \\
&= \square\left(\text{within}[0, 20] \rightarrow \left((s_j(x_1^{w_d}) \geq 0) \; \mathcal{U} \; (s_j(x_2^{w_d}) \geq 0)\right)\right)
\end{aligned}$$

当多个有界算子嵌套时，每个算子都会引入自己的时间约束，并使用单独的 j_0 值。例如，在类似 $\Diamond_{[[a],[b]]}\square_{[[c],[d]]}\phi$ 的公式中，外层 $\Diamond$ 算子会捕获其激活时的 j_0，而内层 $\square$ 算子则会在外层算子的作用范围内激活时捕获自己的 j_0'。这样可确保每个有界时序算子都能相对于其激活时刻，正确跟踪自身的时间区间。

定理 4.9。 *若我们有一条 SSTL 轨迹 w_d 和一条 LTL$_\mathcal{P}$ 轨迹 σ，且 SSTL 公式 $[\varphi]$ 通过定义 4.5 中的翻译函数 τ 被翻译为 LTL$_\mathcal{P}$ 公式 $\tau([\varphi])$，则：*

$$(w_d, [t]) \models [\varphi] \quad \Longleftrightarrow \quad (\sigma, j) \models \tau([\varphi])$$

该定理的证明见附录 *C*。

引理 4.10（SSTL 验证的可判定性）。 *SSTL 的验证问题是可判定的。这直接由定理 4.9 和 LTL 模型检测的可判定性得出。由于每个 SSTL 公式 $[\varphi]$ 都可以翻译为等价的 LTL$_\mathcal{P}$ 公式$\tau([\varphi])$，并且有限状态系统上的 LTL$_\mathcal{P}$ 模型检测是可判定的，[15], 因此 SSTL 验证也是可判定的。*

证明概要。根据定理 4.9，该结果可通过归约为 LTL$_\mathcal{P}$ 模型检测得出。对于每个 SSTL 公式 $[\varphi]$，我们都可以构造一个等价的 LTL$_\mathcal{P}$ 公式 $\tau([\varphi])$。模型检测有限状态系统上的 LTL 可判定：给定一个转移系统 M 和一个 LTL$_\mathcal{P}$ 公式 ϕ，构造一个 Büchi 自动机 A$_{\neg\phi}$，用于表示规约的否定，并检查$L(M) \cap L(\text{A}_{\neg\phi}) = \emptyset$；相应的空性测试是可判定的，且可在线性于乘积自动机 [15]大小的时间内完成。

另外两个论证确保该规约确实产生一个有限 乘积自动机：

(1) LTL$_\mathcal{P}$ 通过对程序变量使用算术谓词扩展了 LTL。在我们的设定中，每个变量的取值范围都是固定的有限域（有界整数或定精度实数），因此赋值字母表仍然是有限的，翻译生成的自动机也都是有限状态的。

(2) SSTL $\rightarrow$ LTL$_\mathcal{P}$ 翻译会引入形如 within$[a, b] := ($ $j_0 + a \leq j \leq j_0 + b)$ 的守卫，这些守卫依赖于一个记账索引 j_0。根据引理 4.7，在任意位置 j，处于活动状态的不同 j_0 副本只有有限个，其数量受公式结构及其时间区间宽度的限制。因此，计算此类守卫所需的状态空间也是有限的。

基于这些观察，模型检测的乘积仍然是有限状态的，因此 SSTL 验证具有可判定性。□

SSTL 翻译能够统一处理安全性质和活性性质：

- **安全性质**（“坏事永远不会发生”）使用 $\square$ 和否定来表达，对应于拒绝违反安全条件的轨迹的 Büchi 自动机。
- **活性性质**（“好事最终会发生”）使用 $\Diamond$来表达，对应于带有接受条件、确保最终满足性质的 Büchi 自动机。

此外，SSTL 模型检测的复杂度继承了 LTL 模型检测的复杂度界：相对于公式大小为 PSPACE 完全，相对于模型大小为多项式时间。实际上，SPIN 等现代模型检查器采用了复杂的优化技术（偏序约简、对称性约简、状态压缩），使得对现实系统模型的验证切实可行。

4.3 SSTL 验证过程

给定 SSTL 公式 $[\varphi]$，我们递归地应用翻译函数 τ，以获得等价的 LTL$_\mathcal{P}$ 公式 $\phi = \tau([\varphi])$，然后使用 SPIN模型检查器 [15] 在系统模型上验证性质 ϕ。

4.3.1 SPIN 对 LTL 的支持$_\mathcal{P}$。 标准 LTL 与 LTL$_\mathcal{P}$ 之间的一个关键区别在于，LTL$_\mathcal{P}$ 原子命题可以是关于程序状态变量的任意谓词，包括算术表达式和关系表达式。经典线性时序逻辑是在固定集合 AP中的命题原子之上定义的，而 LTL$_\mathcal{P}$ 允许使用形如 $s_j(x_i) \geq 0$、$(j_0 + a \leq j \leq j_0 + b)$ 的谓词，以及涉及对整数和浮点变量执行算术运算的复合表达式。

SPIN 原生支持这种扩展形式的 LTL：它允许将任何有效的 Promela（SPIN 的编程语言）布尔表达式作为 LTL 公式中的原子命题。当

表 1：在三个案例研究中验证的 SSTL 性质。使用 SPIN 对这些性质进行验证，并显示了验证时间。

系统	性质	SSTL 公式	结果	Time
交通 轻量级	$[\varphi_{mutex}]$	$\square\neg(NS_{green}=1 \wedge EW_{green}=1)$	满足	1.713秒
	$[\varphi_{NS_safe}]$	$\square(NS_{green}=1 \rightarrow EW_{red}=1)$	满足	1.502秒
	$[\varphi_{EW_safe}]$	$\square(EW_{green}=1 \rightarrow NS_{red}=1)$	满足	1.624秒
	$[\varphi_{fairness}]$	$\square\Diamond(NS_{green}=1)$	不满足	1.559秒
	$[\varphi_{liveness_NS}]$	$\Diamond(NS_{green}=1)$	满足	1.475秒
	$[\varphi_{response_NS_yellow}]$	$\square(NS_{green}=1 \rightarrow \Diamond(NS_{yellow}=1))$	满足	1.539秒
	$[\varphi_{bounded_NS_yellow}]$	$\square(NS_{green}=1 \rightarrow \Diamond_{[[3],[5]]}(NS_{yellow}=1))$	满足	2.093秒
行人 过街	$[\varphi_{no_conflict}]$	$\square\neg(cars_{green}=1 \wedge walk_{signal}=1)$	满足	275.288秒
	$[\varphi_{queue_bounded}]$	$\square(waiting_{peds} \leq 5 \wedge waiting_{peds} \geq 0)$	满足	309.154s
	$[\varphi_{threshold_walk}]$	$\square(waiting_{peds} \geq 2 \rightarrow \Diamond(walk_{signal}=1))$	不满足	2.293s
	$[\varphi_{bounded_wait}]$	$\square((cars_{green}=1 \wedge waiting_{peds} \geq 2) \rightarrow \Diamond_{[[2],[5]]}(walk_{signal}=1))$	不满足	1.627秒
健康 心脏 模型	$[\varphi_{AV}]$	$\square(A_{EGM} \geq V_{a,\ th} \rightarrow \Diamond_{[[0.180],[0.240]]}(V_{EGM} \geq V_{v,\ th}))$	满足	65.879s
	$[\varphi_{VV}]$	$\square(V_{EGM} \geq V_{v,\ th} \rightarrow \Diamond_{[[0.6],[1.00]]}(V_{EGM} > V_{v,\ th}))$	满足	57.497s
	$[\varphi_{liveness_A}]$	$\Diamond(A_{EGM} > V_{a,th})$	满足	6.757秒
	$[\varphi_{liveness_V}]$	$\Diamond(V_{EGM} > V_{v,\ th})$	满足	6.437秒

SPIN 在模型检测期间计算 LTL 公式时，会在当前全局状态的上下文中计算每个谓词，此时所有 Promela变量都具有具体值。例如，谓词 $(A_{EGM} > A_{EGM,\ th})$ 会根据当前状态中 A_{EGM} 和 $A_{EGM,\ th}$ 的值计算为真或假。同样，公式 1 所示的时序约束会直接作为关于状态变量的算术表达式进行计算。

需要注意的是，尽管 SPIN 会对包含算术表达式的谓词进行求值，但底层模型检查器仍在有限状态空间上运行。因此，LTL谓词中出现的所有变量都必须具有有限定义域（例如，有界整数或具有有限精度的浮点值）。对于浮点值，我们保留三位小数，并将其乘以 1000 后四舍五入转换为整数。这种简单的方法确保状态空间是有限的。在 SIH 条件下，离散时间轨迹能够忠实地表示连续时间轨迹（时钟周期之间没有信息损失），因此，针对连续时间轨迹的 SSTL 语义和 STL–SSTL等价性是精确的。舍入步骤是进行有限状态模型检测的实现要求。因此，SPIN 验证相对于量化后的轨迹（按所选精度舍入）是精确的，而非相对于原始实数值轨迹。增加小数位数（例如，使用 10000 而非 1000 进行缩放）可以提高近似精度，但代价是状态空间更大。

*4.3.2 Promela*中的系统建模。我们在 Promela 中将系统建模为有限状态机（FSM）与 C 代码块的组合：前者用于编码系统的控制逻辑，后者用于封装连续动态。模型中的主要时间参考是离散时钟刻计数变量，它在每个 Promela 执行步递增，确保所有采样变量和事件都与这一离散时间线对齐。有限状态机状态表示运行阶段（例如心脏模型中的“等待心房”或“等待心室”），而转换则由信号值或特定时钟刻／时间的流逝触发。

如有需要，物理信号计算通过嵌入式 C 代码实现，Promela 则通过 *c_decl*、*c_state* 和 *c_code* 块集成这些代码。Promela 在每个时钟刻与这些 C 例程同步：仿真向前推进，关键状态变量（例如心电图信号或事件时间戳）随之更新并提供给性质检查使用，有限状态机逻辑则根据新值作出响应。为确保验证所用的状态空间有限，将执行边界（例如最大时钟刻数或事件阈值）设为约束。根据 $\text{LTL}_{\mathcal{P}}$ 性质，我们还定义并跟踪 j_0，将其作为标记有时间界限的义务起始时刻的时钟刻计数变量。

4.3.3 使用 *SPIN* 进行验证。SPIN 通过构造系统模型与对应于给定 $\text{LTL}_{\mathcal{P}}$ 性质的否定的 Büchi 自动机的乘积（交集）来执行验证。系统模型以 Promela 中定义的有限状态机（FSM）表示。工作流程如下：用户将系统编码为 Promela FSM（包括用于描述动态特性的任何嵌入式 C 代码），并使用 ltl 声明指定 $\text{LTL}_{\mathcal{P}}$ 性质。SPIN 在内部将该 LTL 公式的否定转换为 Büchi 自动机。

验证期间，SPIN 会探索系统 FSM 与该性质自动机的乘积，搜索反例——也就是说，寻找系统行为与违反该性质的行为集合相交的执行。通过这种方法，SPIN 能够识别乘积自动机中可作为性质违反见证的无限执行（循环）（例如，错误状态无限次出现，或未能满足活性保证）。

对有限状态空间内所有可达行为进行检查。如果发现交集——即系统模型与否定属性对应自动机的乘积存在一条接受轨迹——SPIN 就会生成一条反例轨迹，供用户检查。否则，如果不存在这样的交集，则可验证该属性对模型的所有可能执行均成立，从而确保被分析系统的安全性和活性。

5 评估

本节通过三个复杂程度递增的案例研究，评估 SSTL 框架的有效性：交通灯控制器、行人过街系统和心脏模型。每个案例研究都展示了 SSTL 如何表达并验证不同系统中的关键时序属性，涵盖基本的安全性和活性约束，以及涉及整数值信号的复杂生理动力学。所有案例研究均同步执行。

表 1 列出了针对每个案例研究验证的关键 SSTL 性质。所有性质均使用 SPIN模型检查器，按照第 4.2 节所述的翻译流程进行验证。翻译过程由人工完成。表中展示了 SSTL 性质、验证结果以及验证该性质所花费的时间。

验证结果表明，SSTL 在复杂度不断增加的多种应用领域中均有效。交通灯控制器是一个有限状态离散系统，展示了对简单布尔信号的验证，包括规定绝不可发生情形的安全性质（互斥、安全联锁），以及保证最终取得进展的活性性质 。该系统的采样频率与心脏模型的数据生成速率相同。在验证的七项性质中，有六项得到满足；公平性性质 $[\varphi_{fairness}]$ 未得到满足，这表明模型或规约存在问题。

行人过街系统是一个离散系统，其信号取值为整数，展示了 SSTL 处理整数值信号的能力，并验证了涉及布尔控制信号和整数队列大小的属性。该系统的采样速率与模型执行速率相同。系统成功验证了安全性质（无流量冲突、队列有界性），验证时间为 2 至 309 秒，展示了该框架对混合信号类型进行推理的能力。从表 1 可以看出，所有属性均满足，但有界响应属性（$[\varphi_{threshold_walk}]$和 $[\varphi_{bounded_wait}]$）未满足，这表明模型或规约可能存在问题。此外，在故障模型中发现故障所需的时间明显更短（以秒计，而非数百秒），这展示了当我们的目标是确定模型是否存在故障时，SSTL 框架的效率。

心脏模型以 0.001 秒的固定间隔采样，展示了 SSTL 对具有精确时序约束的复杂生理系统的表达能力。该心脏模型用 C语言实现，随后在 Promela 中使用，并通过心脏模型的顶层抽象验证属性。由健康心脏模型满足的四个属性均以连续时间表示（如信号时序逻辑），并采用 SSTL 语法，其中包括时间有界响应属性 $[\varphi_{AV}]$和 $[\varphi_{VV}]$，用于描述关键的房室（AV）传导延迟和心室不应期。验证时间为 6.4 至 65.9 秒；有界时序属性耗时较长，反映了针对具有毫秒精度时序约束的整数值心电图信号进行推理的复杂性。

我们还通过在模型中引入疾病，测试了相同的心脏模型属性。我们测试了房室传导阻滞、左束支传导阻滞和右束支传导阻滞 [4]。我们发现，在这三种疾病中，患病心脏模型均不满足属性 $[\varphi_{AV}]$，而满足属性 $[\varphi_{VV}]$、$[\varphi_{liveness_A}]$和 $[\varphi_{liveness_V}]$。因此，我们可以推断，这些疾病会影响房室传导延迟，但不会影响心脏模型的活性。

然而，对于状态空间较大的复杂系统（例如行人过街系统和心脏模型），我们发现其内存需求更高；这两个系统使用的都是整数值信号（或转换为整数值的信号）。行人过街系统和心脏模型的内存需求分别达到 7000 MB 和 9000 MB。不过，可以使用 SPIN 的内存缩减功能（这可能会增加验证时间），并通过模型抽象缩减状态空间，从而降低内存需求。还可以使用其他更高效的 LTL 模型检测技术来降低内存需求。

总的来说，这些案例研究的一项关键发现是 SSTL框架具有普适性：任何可用信号时序逻辑表达且满足信号不变性假设（*SIH*）的性质，都可以系统地转换为 *SSTL*，随后再转换为 $LTL_{\mathcal{P}}$ 以进行验证。此外，SSTL 还能够直接验证离散系统。所验证的性质涵盖了时序逻辑表达能力的各个方面：使用恒真算子的安全性质（□）、使用最终算子的活性性质（◇）、使用时间约束算子的有界响应性质（$\Diamond_{[[a],[b]]}$, $\Box_{[[a],[b]]}$），以及这些性质的复杂组合。

6 相关工作

6.1 离散化概况

表 2 展示了以往的离散化方法。如表所示，Fainekos 和 Pappas [13]提出利用定时状态序列 $\mu = (\sigma, \tau)$，通过离散时间分析来检验连续时间信号，其中采样函数 τ和界定函数 $E(\cdot)$可用于插值。他们的保证依赖于步长有界的可变或固定采样，以及利普希茨连续性或有界导数等正则性条件。这为在离散时间内监测连续属性提供了一种有理论依据的方法，但代价是对信号平滑性作出较强假设。Deshmukh 等人 [5] 则主张在离散时刻 $\{t_0, \ldots\ldots, t_N\}$采用分段常数插值和鲁棒区间语义（RoSI），且不要求均匀采样；这对于在线滑动窗口监测很有吸引力。然而，对于需要确保采样点之间保真度的应用而言，分段常数重构可能过于粗糙。

另一类研究则对时间本身进行离散化。Chen 等人 [8]定义了一种采用均匀采样的离散时间 STL，其中采样间隔为 $\mathbb{N} \geq 0$，并添加了累积时间算子 $C^I\tau$，提供了一个简洁的框架，但切断了与稠密时间语义之间的明确联系，而这种联系对物理推理至关重要。Ferrère 等人 [11]提出了混合时间 STL，使用接口算子 $@cd$ 和 $@dc$，在周期为 T 的周期采样和右连续性条件下实现连续域与离散域之间的转换，使时间转换显式化，但需要谨慎遵循接口规约以保持语义不变。Pant 等人 [20]

[3]由于内存限制，我们使用 SPIN 将验证深度设为 200,000，未发现任何错误。

[4]这些阻滞会阻止电信号从心脏的一个区域传导至另一个区域。为简洁起见，此处省略结果

表 2：离散化方法与我们的 SSTL 对比（简明版）

Work	时间处理	关键假设/工具	相较于 SSTL 的局限性
Fainekos&Pappas [13]	计算树逻辑/度量时序逻辑的离散检查	鲁棒性、有界采样	需要鲁棒性裕度；还需额外假设
Deshmukh 等人 [5]	分段常数插值	RoSI、部分轨迹	解释选择影响语义
Chen&Francis [8]	均匀离散时间信号时序逻辑	对 $\mathbb{N}_{\geq 0}$ 进行固定采样	打破与稠密时间信号时序逻辑的明确联系
Donzé 等人 [11]	带接口的混合时间	$@_{cd}/@_{dc}$ 运算符	接口规约增加了复杂性
Yang 等人 [20]	用于规划的均匀离散化	鲁棒语义、混合整数二次规划	面向优化；不保证等价性

表 3：信号时序逻辑验证方法与我们的 SSTL 的比较

Work	核心思想	优势	相较于 SSTL 的局限性
Roehm 等人 [19]	可达集抽象	对采样点可靠且完备	需要进行可达性分析；抽象代价高昂
Bae&Lee [3]	语法分离 + 可满足性模理论	反驳完备性（有界）	实数上的复杂可满足性模理论；范围受限
Yu 等人 [22]	ϵ-强化 + 可满足性模理论	鲁棒模型检测	仅能保证达到阈值
Lercher 和 Althoff [17]	集合上的四值信号时序逻辑	增量验证	取决于集合细化策略
Belta 和 Sadraddini [6]	差分凸规划（鲁棒性树）	凸子问题	受限的动力学／规范类别

采用均匀网格 $[0{:}dt{:}T]$ 和鲁棒信号时序逻辑语义进行轨迹优化，并假设 $dt \in \mathbb{R}^+$ 以及有限控制输入序列；他们关注的是优化，而非验证。

与这些方法不同，SSTL 提供了一种新的离散时间时序逻辑，它通过信号不变性假设保持与连续时间信号时序逻辑语义已获证明的等价性，而无需插值方案或接口算子。这种等价性确保，使用 SSTL 得到的验证结果可直接对应于底层连续系统的性质。

6.2 信号时序逻辑验证

由于存在连续时间信号，对信息物理系统上的信号时序逻辑进行验证颇具挑战性，而且通常不可判定 [3]。表 3 根据各方法相对于我们方法的核心思路，总结了具有代表性的验证方法。如表所示，Roehm 等人 [19]将系统抽象为可达集，从而对采样时间信号时序逻辑得出可靠且完备的结果，但代价是可达性计算开销巨大；Bae 和 Lee [3] 使用语法分离（STL-GT）将有界问题化简为可满足性模理论问题，但仍需要复杂的实数算术编码；Yu 等人的 STLmc [22] 将 ϵ-强化与可满足性模理论相结合，以获得阈值以内的鲁棒保证，但仍局限于实数上的无量词片段。

其他研究方向也类似地以牺牲通用性来换取可处理性。如表 3 所示，Sato 等人 [20] 将可变区间信号时序逻辑编码为混合整数线性规划，使可判定性取决于线性/矩形动力学；Lercher 和 Althoff [17] 采用基于可达集的四值信号时序逻辑，并进行增量精化，其精度取决于底层的可达性引擎；Takayama 等人 [6] 则通过 DC 编程处理离散时间 STL，以限制系统类别和性质类别为代价，获得凸子问题。

我们的 SSTL 框架采用了不同的方法：通过直接在离散时间域中运算，并证明其与连续时间 STL 等价，我们能够使用离散时间模型检测技术进行验证，同时确保语义上与连续时间性质保持一致。这避免了对复杂的可满足性模理论编码或可达性抽象的需求，同时提供适用于底层连续系统的保证。

7 结论

本文提出了 SSTL，这是首个面向信息物理系统安全性和活性性质可判定验证的 STL 同步离散抽象。我们利用离散时间解决 STL 验证中不可判定性的根本挑战，同时在信号不变性假设（SIH）下保持可靠性和完备性。只要 SIH 性质成立，就可以将 STL 性质转换为 SSTL 性质，并使用 SPIN 进行验证。

我们的方法包含三个关键步骤：首先，通过应用时间投影算子$\lfloor t \rfloor$： $\mathbb{R}_{\geq 0} \to \mathbb{N}_{\geq 0}$，将连续时间信号离散化，其中 $\lfloor t \rfloor = \lfloor t/\Delta t \rfloor$ 将实时时间映射为离散时刻。SIH 确保信号在每个时刻区间内保持不变，从而保留采样点之间的信息。其次，我们提供理论度量方法，以针对给定的应用领域和信号属性计算采样率 Δt。第三，我们使用翻译函数 τ将 SSTL 公式翻译为 $\text{LTL}_{\mathcal{P}}$，其中有界时序算子 $\mathbf{U}_{[a],[b]}$、 $\Box_{[[a],[b]}$、 $\Diamond_{[[a],[b]}$通过 $\text{within}[a,b]$ 谓词进行编码，该谓词用于捕捉相对于义务起始点的时间窗口。

我们使用 SPIN 的穷尽式状态空间探索来验证翻译后的公式，并利用其对状态变量上的算术谓词的原生支持。验证过程通过在 Promela 模型中嵌入 C 代码来整合连续动力学，其中物理仿真（例如心脏模型中的离子通道动力学）会在每次状态转移时执行。SPIN 会在状态空间探索过程中即时评估实值信号（离散化为有限精度，以构成有限状态空间）、布尔状态指示器和整数计时变量上的谓词。模型检查器从 $\text{LTL}_{\mathcal{P}}$公式构造 Büchi 自动机，并将其与系统模型的状态图求交，通过接受环检测发现违规情况。

因此，SSTL 为解决这一关键需求提供了理论基础和实用的验证方法，使下一代可证明安全的自主信息物理系统得以开发。未来的研究方向包括扩展 SSTL 以纳入鲁棒性度量、在更复杂的模型上测试该框架并尝试不同的模型检查器、与运行时监控集成以提供互补保障，以及探索自动属性综合和属性翻译。

致谢

来自奥克兰大学和奥克兰理工大学的作者感谢 Google 资助项目“基于 Google bittide 设计可扩展的同步应用”和“利用 bittide 协议实现经过形式化验证的自动驾驶车辆”提供的支持。他们还感谢与 Google bittide 团队每两周举行的会议，团队成员包括 Calin Cascaval、Tammo Spalink、Sanjay Lall 和 Martin Izzard。他们也感谢 Supratim Gupta 对本文信号处理方面提出的宝贵反馈和建议。

参考文献

[1] Matthias Althoff、Goran Frehse 和 Antoine Girard。2021。用于可达性分析的集合传播技术。《控制、机器人与自主系统年度评论》4, 1（2021），369–395。[2] Rajeev Alur。2015。《信息物理系统原理》。MIT出版社。[3] Kyungmin Bae 和 Jia Lee。2019。使用语法分离对信号时序逻辑性质进行有界模型检测。*stlMC*：信号时序逻辑的有界模型检测3，POPL（2019年1月），51:1–51:30。https://doi.org/10.1145/3290364[4] Kyungmin Bae 和 Jia Lee。2019。使用语法分离对信号时序逻辑性质进行有界模型检测。《*ACM*编程语言汇刊》3，POPL（2019），1–30。[5] Ezio Bartocci、Jyotirmoy Deshmukh、Alexandre Donzé、Georgios Fainekos、Oded Maler、Dejan Ničković 和 Sriram Sankaranarayanan。2018。信息物理系统的规约式监控：理论、工具与应用综述。载于《运行时验证讲义》，Ezio Bartocci 和 Yliès Falcone（编）。第10457卷。施普林格国际出版，Cham，135–175。https://doi.org/10.1007/978-3-319-75632-5_5 丛书名称：计算机科学讲义。[6] Calin Belta 和 Sadra Sadraddini。2019。控制综合的形式化方法：优化视角。《控制、机器人与自主系统年度评论》2, 1（2019），115–140。[7] A. Benveniste、P. Caspi、S.A. Edwards、N. Halbwachs、P. Le Guernic 和 R. de Simone。2003。十二年后的同步语言。《*IEEE*学报》91, 1（2003），64 83 10 1109 2002 805826 – 。https://doi.org/ 。/JPROC. 。[8] Tongwen Chen 和 Bruce Allen Francis。1995。。《最优采样数据控制系统》。施普林格，伦敦。https://doi.org/10.1007/978-1-4471-3037-6[9] David "davidad" Dalrymple、Joar Skalse、Yoshua Bengio、Stuart Russell、Max Tegmark、Sanjit Seshia、Steve Omohundro、Christian Szegedy、Ben Goldhaber、Nora Ammann、Alessandro Abate、Joe Halpern、Clark Barrett、Ding Zhao、Tan Zhi-Xuan、Jeannette Wing 和 Joshua Tenenbaum。2024。迈向有保障的安全AI：确保AI系统稳健可靠的框架。arXiv: 2405 06624 [ ] 2405 06624 . cs.AI https://arxiv.org/abs/ 。[10] Alexandre Donzé。2013。。关于信号时序逻辑。载于《运行时验证》，Axel Legay 和 Saddek Bensalem（编）。施普林格柏林海德堡，柏林，海德堡，382–383。[11] Alexandre Donzé、Thomas Ferrère 和 Oded Maler。2013。STL的高效鲁棒监控。载于《计算机辅助验证》，Natasha Sharygina 和 Helmut Veith（编）。施普林格，柏林，海德堡，264–279。https://doi.org/10.1007/978-3-642-39799-8 19[12] Tommaso Dreossi、Alexandre Donzé 和 Sanjit A Seshia。2019。利用机器学习组件对信息物理系统进行组合式证伪。《自动推理杂志》63, 4（2019），1031–1053。[13] Georgios E. Fainekos 和 George J. Pappas。2009。连续时间信号时序逻辑规约的鲁棒性。《理论计算机科学》410, 42（2009年9月 2009 4262 4291 10 1016 2009 06 021 ），– 。[14] https://doi.org/ 。/j.tcs. 。Thomas A Henzinger、Peter W Kopke、Anuj Puri 和 Pravin Varaiya。1995。混合自动机有哪些性质是可判定的？载于《第二十七届*ACM*计算理论年度研讨会论文集》。373–382。

[15] Gerard Holzmann。2011。《*SPIN*模型检查器：入门与参考手册》（第1版）。Addison-Wesley Professional。[16] YoungMin Kwon 和 Gul Agha。2008。LTLC：控制线性时序逻辑。载于《混合系统：计算与控制》。施普林格，316–329。https://osl.cs. illinois.edu/media/papers/kwon-2008-hybrid-ltlc.pdf[17] Florian Lercher 和 Matthias Althoff。2024。使用四值信号时序逻辑对混合系统进行增量验证。载于《国际计算机辅助验证会议》。施普林格，259–281。[18] Lars Lindemann 和 Dimos V. Dimarogonas。2019。信号时序逻辑任务的控制屏障函数。《*IEEE*控制系统快报》3, 1（2019年1月），96 101 10 1109 2018 2853182 – 。https://doi.org/ 。/LCSYS. 。[19] Hendrik Roehm、Jens Oehlerking、Thomas Heinz 和 Matthias Althoff。2016。连续系统与混合系统的STL模型检查。载于《验证与分析自动化技术》，Cyrille Artho、Axel Legay 和 Doron Peled（编）。施普林格国际出版，Cham，412–427。https://doi.org/10.1007/978- 3-319-46520-3 26[20] Guang Yang、Calin Belta 和 Roberto Tron。2020。结合控制屏障函数的连续时间信号时序逻辑规划。载于《*2020*年美国控制会议（*ACC*）》。4612–4618。https://doi.org/10.23919/ACC45564.2020. 9147387 ISSN：2378-5861。[21] Eugene Yip、Sidharta Andalam、Partha S. Roop、Avinash Malik、Mark L. Trew、Weiwei Ai 和 Nitish Patel。2018。迈向用于起搏器验证的心脏传导系统仿真。《*ACM*信息物理系统汇刊》2, 4，文章 32 2018 26 10 1145 3134845（7月），页数。[22] 2022 https://doi.org/ 。/ Geunyeol Yu、Jia Lee 和 Kyungmin Bae。 。STLmc：使用可满足性模理论对混合系统进行鲁棒STL模型检查。载于《计算机辅助验证》，Sharon Shoham 和 Yakir Vizel（编）。施普林格国际出版，Cham，524–537。https://doi.org/10.1007/978-3-031-13185-1_26

信号时序逻辑的语法与语义

定义 *A.1.*（STL轨迹）STL轨迹 w 是一个函数 $w\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^n$，它将实时时间映射为信号值。对于给定的轨迹 w和实时时间 $t \in \mathbb{R}_{\geq 0}$，$x_i^w(t)$表示轨迹 w中信号 x_i在时间 t的值，其中 $i = 1,\ 2,\ \ldots\ldots,\ n$。

定义 *A.2*。（STL 语法）STL 公式的语法 φ 定义如下：

$$\varphi ::= \top \mid x_i^w(t) \geq 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U}_{[a,b]} \varphi_2 \tag{40}$$

定义 *A.3*。（STL 语义）STL 公式的语义φ 定义如下：

$$\begin{aligned}
(w,t) &\models \top \iff \top \text{ is true} \\
(w,t) &\models x_i^w(t) \geq 0 \iff x_i^w(t) \geq 0 \\
(w,t) &\models \neg\varphi \iff (w,t) \not\models \varphi \\
(w,t) &\models \varphi_1 \wedge \varphi_2 \iff (w,t) \models \varphi_1 \text{ and } (w,t) \models \varphi_2
\end{aligned}$$

$$\begin{aligned}
(w,t) \models \varphi_1 \mathcal{U}_{[a,b]} \varphi_2 \iff \exists t_1 \in [t+a, t+b] : (w,t_1) \models \varphi_2 \text{ and} \\
\forall t_2 \in [t, t_1) : (w,t_2) \models \varphi_1
\end{aligned}$$

B 定理 3.6 的证明

证明。正向：给定 $(w,t) \models \varphi$，我们需要证明 $(w_d, [t]) \models [\varphi]$。令 $t \in [k\Delta t, (k+1)\Delta t)$，其中某个 $k \in \mathbb{N}_{\geq 0}$。那么，$[t] = k$并且根据 SIH，$\vec{x}(t) = \vec{x}([t])$。接下来，证明将基于信号时序逻辑公式 φ的四种情况以及 SSTL 语义。

- **Case 1:** $\varphi = \top$.
  Then $[\varphi] = \top$, and trivially
  $$(w,t) \models \top \implies (w_d, [t]) \models \top.$$
- **Case 2:** $\varphi = x_i^w(t) \geq 0$.
  By definition $[\varphi] = x_i^{w_d}([t]) \geq 0$. By SIH we have $x_i^w(t) = x_i^{w_d}([t])$, hence
  $$(w,t) \models x_i^w(t) \geq 0 \implies (w_d, [t]) \models x_i^{w_d}([t]) \geq 0.$$

- **Case 3:** $\varphi = \neg\psi$.
  Then $[\varphi] = \neg[\psi]$.

$$\begin{aligned}(w,t) \models \neg\psi &\iff (w,t) \not\models \psi \\ &\implies (w_d,[t]) \not\models [\psi] \quad \text{(SIH)} \\ &\implies (w_d,[t]) \models \neg[\psi] \\ \therefore (w,t) \models \neg\psi &\implies (w_d,[t]) \models \neg[\psi].\end{aligned}$$

- **情况4：** $\varphi = \varphi_1 \wedge \varphi_2$。
  然后 $[\varphi] = [\varphi_1] \wedge [\varphi_2]$。

$$\begin{aligned}\because (w,t) \models \varphi_1 \wedge \varphi_2 &\iff (w,t) \models \varphi_1 \text{ and } (w,t) \models \varphi_2 \\ &\implies (w_d,[t]) \models [\varphi_1] \text{ and } (w_d,[t]) \models [\varphi_2] \quad \text{(SIH)} \\ &\implies (w_d,[t]) \models [\varphi_1] \wedge [\varphi_2] \\ \therefore (w,t) \models \varphi_1 \wedge \varphi_2 &\implies (w_d,[t]) \models [\varphi_1] \wedge [\varphi_2].\end{aligned}$$

- **情况5：** $\varphi = \varphi_1 \, \mathrm{U}_{[a,b]} \, \varphi_2$。根据定义
  $[\varphi] = [\varphi_1] \, \mathrm{U}_{[a,b]} \, [\varphi_2]$。我们必须证明

$$(w,t) \models \varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2 \implies (w,[t]) \models [\varphi_1] \, \mathcal{U}_{[a,b]} \, [\varphi_2].$$

  假设 $(w,t) \models \varphi_1 \mathrm{U}_{[a,b]} \varphi_2$。那么，根据密集时间 STL 语义，存在一个实数时刻 $t_1 \in [t+a, t+b]$，使得

$$(w,t_1) \models \varphi_2 \quad \text{and} \quad \forall t_2 \in [t,t_1) : (w,t_2) \models \varphi_1.$$

  根据 直到 的 SSTL 语义，$[t_1] \in [[t+a],[t+b]]$和 $[t_2] \in [[t],[t_1])$，因此，根据 SIH，我们有，$x_i^w(t_1) = x_i^{w_d}([t_1])$ 和 $x_i^w(t_2) = x_i^{w_d}([t_2])$，因此，

$$(w_d,[t_1]) \models [\varphi_2] \quad \text{and} \quad (w_d,[t_2]) \models [\varphi_1].$$

  因此，

$$(w_d,[t]) \models [\varphi_1] \, \mathcal{U}_{[a,b]} \, [\varphi_2].$$

$$\therefore (w,t) \models \varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2 \implies (w_d,[t]) \models [\varphi_1] \, \mathcal{U}_{[a,b]} \, [\varphi_2].$$

  反向：给定 $(w_d,[t]) \models [\varphi]$，我们必须证明 $(w,t) \models \varphi$。我们通过反证法证明这一点，假设 $(w,t) \not\models \varphi$ 但 $(w_d,[t]) \models [\varphi]$。令 $t \in [k\Delta t, (k+1)\Delta t)$ 对某个 $k \in \mathbb{N}_{\geq 0}$ 成立，因此 $[t] = k$。根据 SIH，$x_i^w(t) = x_i^{w_d}([t])$。

- **情况 1：** $[\varphi] = \top$。于是 $\varphi = \top$。假设 $(w,t) \not\models \top$ 是错误的，因为 $\top$ 全局为真。$\therefore (w_d,[t]) \models \top \implies (w,t) \models \top$。

- 情况 2：$[\varphi] = x_i^{w_d}([t]) \geq 0$。– 根据假设，$(w_d,[t]) \models x_i^{w_d}([t]) \geq 0$，这意味着 $x_i^{w_d}([t]) \geq 0$。– 根据 SIH，$x_i^w(t) = x_i^{w_d}([t])$。– 因此，

$$x_i^w(t) \geq 0 \qquad w,t \models x_i^w(t) \geq 0,$$

  这意味着 $(w,t) \not\models \varphi$ 的假设相矛盾。$\therefore (w_d,[t]) \models x_i^{w_d}([t]) \geq 0 \implies (w,t) \models x_i^w(t) \geq 0$。

- **C况3：** $[\varphi] = [\varphi_1] \wedge [\varphi_2]$.
  – 根据 SSTL 语义，$(w_d,[t]) \models [\varphi_1] \wedge [\varphi_2]$ 意味着 $(w_d,[t]) \models [\varphi_1]$ 以及 $(w_d,[t]) \models [\varphi_2]$。– 根据 SIH，由于 $w_d,[t] \models [\varphi_1]$ 和 $w,t \models \varphi_1$，$w_d,[t] \models [\varphi_2]$ 和 $w,t \models \varphi_2$，必然有 $(w,t) \models \psi$ 且 $(w,t) \models \theta$。– 根据信号时序逻辑语义，$(w,t) \models \psi \wedge \theta$。– 这与假设 $(w,t) \not\models \varphi_1 \wedge \varphi_2$ 矛盾。$\therefore (w_d,[t]) \models [\varphi_1] \wedge [\varphi_2] \implies (w,t) \models \varphi_1 \wedge \varphi_2$。

- **情况4：** $[\varphi] = \neg[\psi]$.
  – 根据假设，$(w_d,[t]) \models \neg[\psi]$；根据 SSTL 语义，这意味着 $(w_d,[t]) \not\models [\psi]$。– 根据 SIH，若 $(w_d,[t]) \not\models [\psi]$，则必然有 $(w,t) \not\models \psi$。– 根据信号时序逻辑语义，$(w,t) \not\models \psi$意味着 $(w,t) \models \neg\psi$。– 这与假设 $(w,t) \not\models \neg\psi$矛盾。$\therefore (w_d,[t]) \models \neg[\psi] \implies (w,t) \models \neg\psi$。

- **情况5：** $[\varphi] = [\varphi_1]\mathcal{U}_{[a,b]}[\varphi_2]$.
  – 根据假设，$(w_d,[t]) \models [\varphi_1]\mathrm{U}_{[a,b]}[\varphi_2]$。– 根据 SSTL 语义，存在一个离散时间 $[t'] \in [[t+a],[t+b]]$，使得 $(w_d,[t']) \models [\varphi_2]$，且对于所有离散时间 $[t''] \in [[t],[t'])$，$(w_d,[t'']) \models [\varphi_1]$。– 令 $t_{real} \in [t+a, t+b]$为一个映射到 $[t']$的实时时刻，即 $[t_{real}] = [t']$。– 根据 SIH，由于 $(w_d,[t']) \models [\varphi_2]$，必有 $(w,t_{real}) \models \varphi_2$。– 类似地，对于所有满足 $[t''_{real}] \in [[t],[t'])$的 $t''_{real} \in [t,t_{real}]$，由于 $(w_d,[t'']) \models [\varphi_1]$，必有 $(w,t''_{real}) \models \varphi_1$。– 根据稠密时间信号时序逻辑的语义，存在这样的 t_{real}，以及 $[t,t_{real}]$中满足这些条件的一个连续时间区间，这意味着 $(w,t) \models \varphi_1 \mathrm{U}_{[a,b]} \varphi_2$

  – 这与假设 $(w,t) \not\models \varphi_1 \mathrm{U}_{[a,b]} \varphi_2$ 矛盾。$\therefore (w_d,[t]) \models [\varphi_1]\mathrm{U}_{[a,b]}[\varphi_2] \implies (w,t) \models \varphi_1 \mathrm{U}_{[a,b]} \varphi_2$。

由于两个方向都已证明，我们得出 $\varphi \implies w_d,[t] \models [\varphi] \iff (w,t) \models \varphi$。 □

C语言 定理 4.9 的证明

证明。设 w_d 是一条 SSTL 轨迹，并设 σ 是将同一状态序列视为 $\mathrm{LTL}_{\mathcal{P}}$ 轨迹。根据定义 4.4，σ 中的离散位置 j 与 w_d的逻辑时钟刻 $[t]$ 相对应，即 $j = [t]$。我们通过对 $[\varphi]$进行结构归纳法证明双向蕴含。

正向：给定 $(w_d,[t]) \models [\varphi]$，我们需要证明 $(\sigma,j) \models \tau([\varphi])$。

- **情况 1：** $[\varphi] = \top$。根据定义 3.5，$(w_d,[t]) \models \top$ 显然成立。同样，根据定义 4.2，$(\sigma,j) \models \top$ 也显然成立。因此，

$$(w_d,[t]) \models \top \implies (\sigma,j) \models \tau(\top).$$

- **情况 2：** $[\varphi] = x_i^{w_d}([t]) \geq 0$。
  根据定义 4.4，$\tau(x_i^{w_d}[t] \geq 0) = s_j \, x_i^{w_d} \geq 0$。$w_d,[t] \models x_i^{w_d}[t] \geq 0$ 翻译为 (σ,j)。给定 (σ,j)，根据 SSTL 语义（定义 3.5），我们有 $x_i^{w_d}([t]) \geq 0$。由于 $\mathrm{LTL}_{\mathcal{P}}$ 轨迹中的状态s_j赋予相同的值（其中 $j = [t]$），$s_j(x_i^{w_d}) = x_i^{w_d}([t])$，因此得到 $s_j(x_i^{w_d}) \geq 0$。所以，根据 $\mathrm{LTL}_{\mathcal{P}}$ 4.2 $\sigma,j \models s_j \, x_i^{w_d} \geq 0$语义（定义 3.5），$(\sigma,j)$。

- **情形 3：** $[\varphi] = \neg[\psi]$。
  假设 $(w_d,[t]) \models \neg[\psi]$。
  – 根据 SSTL 语义（定义 3.5），这意味着 $(w_d,[t]) \not\models [\psi]$。
  – 根据子公式上的结构归纳假设 $[\psi]$，我们有 $(w_d,[t]) \not\models [\psi]$ 蕴含 $(\sigma,j) \not\models \tau([\psi])$。

– 根据 $\mathrm{LTL}_{\mathcal{P}}$ 语义（定义 4.2），$(\sigma,\ j) \not\models \tau([\psi])$ 表示 $(\sigma,\ j) \models \neg\tau([\psi])$。– 根据翻译（定义 4.4），$\tau(\neg[\psi]) = \neg\tau([\psi])$。因此，$(w_d,\ [t]) \models \neg[\psi] \Longrightarrow (\sigma,\ j) \models \tau(\neg[\psi])$。

• 情况4：$[\varphi] = [\psi_1] \wedge [\psi_2]$。假设 $(w_d,\ [t]) \models [\psi_1] \wedge [\psi_2]$。– 根据 SSTL 语义（定义 3.5），这意味着 $w_d\ [t] \models [\psi_1]$ 和 $w_d\ [t] \models [\psi_2]$。– 根据结构归纳假设，$(w_d,\ [t]) \models [\psi_1]$ 意味着 $(\sigma,\ j) \models \tau([\psi_1])$，且 $(w_d,\ [t]) \models [\psi_2]$ 意味着 $(\sigma,\ j) \models \tau([\psi_2])$。– 根据 LTLP 语义（定义 4.2），$(\sigma,\ j) \models \tau([\psi_1])$ 和 $(\sigma,\ j) \models \tau([\psi_2])$ 意味着 $(\sigma,\ j) \models \tau([\psi_1]) \wedge \tau([\psi_2])$。– 根据翻译（定义 4.4），$\tau([\psi_1] \wedge [\psi_2]) = \tau([\psi_1]) \wedge \tau([\psi_2])$。因此，$w_d\ [t] \models [\psi_1] \wedge [\psi_2] \Longrightarrow \sigma\ j \models \tau\ [\psi_1] \wedge [\psi_2]$。

• 情况5：$[\varphi] = [\psi_1] \mathrm{U}\ [\psi_2]$（无界直到）。假设 $(w_d, [t]) \models [\psi_1]\ \mathrm{U}\ [\psi_2]$。– 根据 SSTL 语义（定义 3.5），存在一个离散时间 $[t'] \geq [t]$，使得 $(w_d, [t']) \models [\psi_2]$，并且对于所有 $[t''] \in [[t], [t'])$，$(w_d, [t'']) \models [\psi_1]$。– 根据结构归纳假设，$(w_d, [t']) \models [\psi_2]$ $\sigma\ [t\ ] \models \tau\ [\psi_2]$ $[t\ ] \in$ 蕴含 $(,')$ $(),$ $[[t], [t\ ]$ $w_d\ [t\ ] \models [\psi_1]$ $\sigma\ [t\ ] \models \tau\ [\psi_1]$ 并且对于所有 $''$），$(,'')$ 蕴含 $(,'')$ $()$。– 由于 $j = [t]$，且根据定义 4.4 中确立的对应关系，存在 $k = [t'] \geq j$，使得 $(\sigma, k) \models \tau([\psi_2])$，并且对于所有 $l \in [j, k]$，$(\sigma, l) \models \tau([\psi_1])$。– 这正是 LTLP 中 Until 的语义（定义 4.2）：$(\sigma, j) \models \tau([\psi_1]) \mathrm{U} \tau([\psi_2])$。因此，$(w_d, [t]) \models [\psi_1]\ \mathrm{U}\ [\psi_2] \Longrightarrow (\sigma, j) \models \tau([\psi_1]\ \mathrm{U}\ [\psi_2])$。

• 情况 6：$[\varphi] = [\psi_1] \mathrm{U}_{[[a],[b]]}\ [\psi_2]$（有界直到）。假设 $(w_d,\ [t]) \models [\psi_1] \mathrm{U}_{[[a],[b]]}\ [\psi_2]$。– 根据 SSTL 语义（定义 3.5），存在一个离散时间 $[t\ ] \in [[t+a], [t+b]]$ $w_d\ [t\ ] \models'$，使得 $[\psi_2]$ $[t\ ] \in [[t], [t\ ]$ $w_d\ [t\ ] \models [\psi_1]$ $(,')$ 并且对于所有 $''$），$(,'')$。– 根据结构归纳假设，$(w_d,\ [t'])\ \models \psi_2]$ $\sigma\ [t\ ] \models \tau\ [\psi_2]$ $[t\ ] \in [[t], [t\ ]$ $w_d\ [t\ ] \models [\psi_1]$ $\sigma\ [t\ ] \models \tau\ [\psi_1]$ t帽子 蕴含 $(,')$ $()$ 并且对于所有 $''$），$(,'')$ 蕴含 $(,'')$ $()$。– 令 $j_0 = [t]$ 为义务起始索引。由于 $[t'] \in [t + a], [t + b]]$，我们有 $[t + a] \leq [t'] \leq [t + b]$，这意味着 $j_0 + a \leq j = [t'] \leq j_0 + b$。– 根据 within$[a, b]$ 守卫的定义，这意味着 within$[a, b]$ 在位置 $j = [t']$ 处为真。– 因此，$(\sigma,\ [t'])\ \models \tau([\psi_2]) \wedge$ within$[a, b]$。– 结合对于所有 $l \in [j_0, [t'])$，$(\sigma,\ l) \models \tau([\psi_1])$ 这一事实，根据直到的 LTLP 语义，我们有：$(\sigma,\ j_0) \models \tau([\psi_1])\ \mathrm{U}\ (\tau([\psi_2]) \wedge$within$[a, b])$。– 根据翻译，$\tau([\psi_1] \mathrm{U}_{[[a],[b]]}[\psi_2]) = \tau([\psi_1])\ \mathrm{U}\ (\tau([\psi_2])$ 在 within$[a, b])$ 内。因此，$w_d\ [t] \models [\psi_1]\ _{[[a],[b]]}\ [\psi_2] \Longrightarrow \sigma\ j \models ,)\ \mathrm{U}\ (,\ )([\psi_1]$ $\tau \mathrm{U}_{[[a],[b]]}[\psi_2])$。

• **情形 7：** $[\varphi] = \Diamond_{[[a],[b]]}[\psi]$**（有界最终）**。假设 $(w_d, [t]) \models \Diamond_{[[a],[b]]}[\psi]$.– 根据 SSTL 中有界最终的定义（定义 3.5），这等价于 $\top \mathrm{U}_{[[a],[b]]}[\psi]$。– 根据 SSTL 语义，存在一个离散时间 $[t'] \in [[t + a], [t + b]]$ $w_d\ [t'] \models [\psi]$，使得 $(,)$。– 根据结构归纳假设，$(w_d\ [t'] \models ,)$ 蕴含 $(,)$ $()$。$[\psi]$ $\sigma\ [t'] \models \tau\ [\psi]$ – 令 $j_0 = [t]$。由于 $[t'] \in [[t + a], [t + b]]$，我们有 $j_0 + a \leq [t'] \leq j_0 + b$，这意味着 within$[a, b]$ 在位置 $[t']$ 成立。– 因此，$(\sigma, [t']) \models \tau([\psi]) \wedge$within$[a, b]$。– 由于存在这样一个位置 $[t'] \geq [t]$，根据 LTL 中钻石算子的语义：$(\sigma, j) \models \Diamond(\tau([\psi]) \wedge$within$[a, b])$。– 根据翻译（定义 4.4），$\Diamond\ \tau\ [\psi] \wedge$ $[a, b]$ $(\Diamond_{[[a],[b]]}[\psi])$ $(\ )$ within $)$。

因此，$(w_d,\ [t]) \models \Diamond_{[[a],[b]]}[\psi] \Longrightarrow (\sigma,\ j) \models \tau(\Diamond_{[[a],[b]]}[\psi])$。

• 情况 8：$[\varphi] = \Box_{[[a],[b]]}[\psi]$（有界始终）。假设 $(w_d, [t])$ $\models \Box_{[[a],[b]]}[\psi]$。– 根据 SSTL 中有界始终的定义（定义 3.5），这等价于 $\neg\Diamond_{[[a],[b]]}\neg[\psi]$。– 根据 SSTL 语义，对于所有离散时间 $[t\ ] \in [[t +', 我们有 (,')$。– 根据结构归纳假设，对于所有 $[t'] \in [[t + a], [t + b]]$ $w_d\ [t\ ] \models [\psi]$ $\sigma\ [t\ ] \models,$ $(,')$ 蕴含 $(,')$ $(\tau[\psi])$。– 令 $j_0 = [t]$。对于任意位置 $k \geq j_0$，如果 within$[a, b]$ 在 k 处为真，那么根据定义 $j_0 + a \leq k \leq j_0 + b$，这意味着对于某个 $[t'] \in [[t + a], [t + b]]$，$k = [t']$。– 对于这样的 k，我们有 $(\sigma, k) \models \tau([\psi])$。– 因此，对于所有 $k \geq j_0$，如果 within$[a, b]$ 在 k 处成立，那么 $\tau\ [\psi]$ k $[a, b] \rightarrow ()$ 在 处成立。这就是 within $([\psi])$ 在每个位置处的蕴含关系。– 根据 LTL 的全称框算子语义：$(\sigma\ j_0 \models \Box$ $[a, b] \rightarrow,)$ $(\text{within } (\tau[\psi]))$。– 根据翻译（定义 4.4），$= \Box$ $(\Box_{[[a],[b]]}[\psi])$ $(\text{within}[a, b] \rightarrow \tau([\psi]))$。因此，$(w_d, [t]) \models \Box_{[[a],[b]]}[\psi] \Longrightarrow (\sigma, j) \models \tau(\Box_{[[a],[b]]}[\psi])$。由此，对所有 SSTL 公式，正向方向均成立。反向：给定 $(\sigma, j) \models \tau([\varphi])$，我们需要证明 $(w_d,\ [t]) \models [\varphi]$。

• **情况 1：** $[\varphi] = \top$。根据定义 4.2，$(\sigma,\ j) \models \top$ 显然成立。同样，根据定义 3.5，$(w_d,\ [t]) \models \top$ 也显然成立。因此，

$$(\sigma, j) \models \tau(\top) \implies (w_d, [t]) \models \top.$$

• **情况 2：** $[\varphi] = x_i^{w_d}([t]) \geq 0$。假设 $(\sigma,\ j) \models s_j(x_i^{w_d}) \geq 0$。– 根据 $\mathrm{LTL}_{\mathrm{P}}$ 波语义（定义 4.2 $s_j\ x_i^{w_d} \geq .$），这意味着 $()$。– 根据定义 4.4 中的对应关系，$s_j = x_i^{w_d}\ [t]$ $(x_i^{w_d})$ $()$。– 因此，$x_i^{w_d}([t]) \geq 0$。– 根据 SSTL 语义（定义 3.5 $w_d\ [t] \models x_i^{w_d}\ [t] \geq .$），$(,)$ $()$。因此，$(\sigma\ j \models \tau\ x_i^{w_d}\ [t] \geq 0 \implies w_d\ [t] \models ,)$ $(())$ $(,)$ $()$。

• **情形 3：** $[\varphi] = \neg[\psi]$。假设 $(\sigma,\ j) \models \tau(\neg[\psi])$。

– 根据翻译， $\tau(\neg[\psi]) = \neg\tau([\psi])$.
– 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），$(\sigma, j) \models \neg\tau([\psi])$ 表示 $(\sigma, j) \not\models \tau([\psi])$。– 根据对子公式 $[\psi]$ 的结构归纳假设，$(\sigma, j) \not\models \tau([\psi])$ 蕴含 $(w_d, [t]) \not\models [\psi]$。 – 根据 SSTL 语义（定义 3.5），$(w_d, [t]) \not\models [\psi]$ 表示 $(w_d, [t]) \models \neg[\psi]$。

因此，$(\sigma, j) \models \tau(\neg[\psi]) \implies (w_d, [t]) \models \neg[\psi]$.

- **情况4：** $[\varphi] = [\psi_1] \wedge [\psi_2]$。 假设 $(\sigma, j) \models \tau([\psi_1] \wedge [\psi_2])$。
– 根据翻译， $\tau([\psi_1] \wedge [\psi_2]) = \tau([\psi_1]) \wedge \tau([\psi_2])$。– 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），$\sigma, j \models \tau [\psi_1] \wedge$ 语义（定义 .），τ，$(\sigma, j) \models \tau([\psi_1])$ 且 $(\sigma, j) \models \tau([\psi_2])$ 表示 $(\sigma, j) \models \tau([\psi_1] \wedge [\psi_2])$。– 根据结构归纳假设，$(\sigma, j) \models \tau([\psi_1])$ 蕴含 $(w_d, [t]) \models [\psi_1]$，且 $(\sigma, j) \models \tau([\psi_2])$ 蕴含 $(w_d, [t]) \models [\psi_2]$。– 根据 SSTL 语义（定义 3.5），$(w_d, [t]) \models [\psi_1]$ 且 $(w_d, [t]) \models [\psi_2]$ 表示 $(w_d, [t]) \models [\psi_1] \wedge [\psi_2]$。

因此，$(\sigma, j) \models \tau [\psi_1] \wedge [\psi_2] \implies w_d [t] \models [\psi_1] \wedge$ 。

- **情况5：** $[\varphi] = [\psi_1] \mathcal{U} [\psi_2]$ **（无界直到）**。假设 $(\sigma, j) \models \tau([\psi_1] \mathcal{U} [\psi_2])$。
设 1 – 根据翻译， $\tau([\psi_1] \mathrm{U} [\psi_2]) = \tau([\psi_1]) \mathrm{U} \tau([\psi_2])$。– 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），存在 $k \geq [t]$ 使得 $(\sigma, k) \models \tau([\psi_2])$，且对所有 $l \in [[t], k)$，$(\sigma, l) \models \tau([\psi_1])$。– 令 $[t'] := k$。根据结构归纳假设，$(\sigma, k) \models \tau([\psi_2])$ 蕴含 $(w_d, [t']) \models [\psi_2]$，且对所有 $l \in [[t], k)$，$(\sigma, l) \models \tau([\psi_1])$ 蕴含 $(w_d, l) \models [\psi_1]$。– 由于 $[t'] = k \geq [t]$，我们有 $[t'] \geq [t]$。– 因此，存在 $[t'] \geq [t]$ $w_d [t'] \models$ 使得 $[\psi_2]$ $[t''] \in [[t], [t']$ $w_d [t''] \models [\psi_1]$ 得 (,)，且对所有)，(,)。– 根据 SSTL 语义（定义 3.5），这恰好是 $(w_d, [t]) \models [\psi_1] \mathrm{U} [\psi_2]$。

因此，$(\sigma, j) \models \tau([\psi_1] \mathrm{U} [\psi_2]) \implies (w_d, [t]) \models [\psi_1] \mathrm{U} [\psi_2]$.

- **情形 6：** $[\varphi] = [\psi_1] \mathcal{U}_{[[a],[b]]} [\psi_2]$ **（有界直到）**。
假设 $(\sigma, j) \models \tau([\psi_1] \mathrm{U}_{[[a],[b]]} [\psi_2])$。– 根据翻译， $\tau([\psi_1] \mathrm{U}_{[[a],[b]]} [\psi_2]) = \tau([\psi_1]) \mathrm{U} (\tau([\psi_2]) \wedge$ within$[a, b])$. – 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），存在 $k \geq [t]$，使得 $(\sigma, k) \models \tau([\psi_2]) \wedge$within$[a, b]$，并且对于所有 $l \in [[t], k)$，$(\sigma, l) \models \tau([\psi_1])$。– 由 ($\sigma\ k \models \tau\ [\psi_2] \wedge [a, b]$ $\sigma\ k \models$,) ( ) τ within , 可得 σ, k (,) $([\psi_2])$ 且 (,) $\models$within$[a, b]$。– 令 $j_0 = [t]$ 和 $[t'] := k$。由于 $(\sigma, k) \models$within$[a, b]$，根据守卫的定义可得 $j_0 + a \leq k \leq j_0 + b$ $[t + a] \leq [t'] \leq [t + b]$ $[t'] \in$，这意味着，因此 $[[t + a], [t + b]]$。– 根据结构归纳假设，$(\sigma, k) \models \tau([\psi_2])$ 意味着 ($w_d [t'] \models [\psi_2]$ $l \in [[t], k$ $\sigma\ l \models$,), 并且对于所有)，(,) $([\psi_1])$ 意味着 $(w_d,$) $\models [\psi_1]$。– 因此，存在 $[t'] \in [[t + a], [t + b]]$，使得 $(w_d, [t']) \models [\psi_2]$，并且对于所有 $[t''] \in [[t], [t'])$，$(w_d, [t'']) \models [\psi_1]$。– 根据 SSTL 语义（定义 3.5），这恰好就是 $(w_d, [t]) \models [\psi_1] \mathrm{U}_{[[a],[b]]} [\psi_2]$。

因此，($\sigma\ j \models \tau\ [\psi_1]$ $[\psi_1]$ $_{[[a],[b]]}[\psi_2] \implies w_d [t] \models$,) ( ) (,) U。

- **情形 7：** $[\varphi] = \Diamond_{[[a],[b]]} [\psi]$ **（有界最终）**。假设 $(\sigma, j) \models \tau(\Diamond_{[[a],[b]]} [\psi])$。– 根据翻译， $\tau(\Diamond_{[[a],[b]]} [\psi]) = \Diamond(\tau([\psi]) \wedge$在.....范围内 $[a, b])$。
– 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），存在使得 $k \geq [t]$ $\sigma\ k \models \tau\ [\psi] \wedge [a, b]$ (,) ( ) 在.....范围内。– 由此，我们有 $(\sigma, k) \models \tau([\psi])$，且 $(\sigma, k) \models$在.....范围内$[a, b]$。– 令 $j_0 = [t]$ 和 $[t'] := k$。由于 $(\sigma, k) \models$在.....范围内$[a, b]$，根据守卫的定义，我们有 $j_0 + a \leq k \leq j_0 + b$ $[t + a] \leq [t'] \leq [t + b]$ $[t'] \in$，这意味着，因此 $[[t + a], [t + b]]$。– 根据结构归纳假设，$(\sigma, k) \models \tau([\psi])$ 蕴含 $(w_d, [t']) \models [\psi]$。– 因此，存在 $[t'] \in [[t + a], [t + b]]$ 使得 $(w_d, [t']) \models [\psi]$。– 根据 SSTL 中有界最终的定义（定义 3.5），这恰好是 $(w_d, [t]) \models \Diamond_{[[a],[b]]} [\psi]$。

因此，($\sigma\ j \models \tau \Diamond_{[[a],[b]]} [\psi] \implies w_d [t] \models$,) ( ) (,)。 $\Diamond_{[[a],[b]]} [\psi]$

- **情况 8：** $[\varphi] = \Box_{[[a],[b]]} [\psi]$ **（有界始终）**。假设 $(\sigma, j) \models \tau(\Box_{[[a]}$ $_{'[b]]} [\psi])$。
– 根据翻译， $\tau\ \Box_{[[a],[b]]} [\psi] = \Box$ $[a, b] \to$ ( ) （在 τ $([\psi])$ 内）。– 根据 LTL$_{\mathcal{P}}$ 语义（定义 4.2），对于所有 $k \geq [t]$，我们有 $(\sigma, k) \models$在$[a, b] \to \tau([\psi])$ $k \geq [t]$内。– 这意味着，对于所有 $k \geq [t]$，如果 $(\sigma, k) \models$在$[a, b]$内，那么 $(\sigma, k) \models \tau([\psi])$。– 任取一个 $[t'] \in [[t + a], [t + b]]$。令 $k := [t']$。于是 $[t + a] \leq k \leq [t + b]$，这意味着 k满足 $j_0 + a \leq k \leq j_0 + b$，其中 $j_0 = [t]$。– 因此，$(\sigma, k) \models$在$[a, b]$ 内。– 根据上述蕴含关系，$(\sigma, k) \models \tau([\psi])$。– 根据结构归纳假设，$(\sigma, k) \models \tau([\psi])$ 蕴含 $(w_d, [t']) \models [\psi]$。– 由于 $[t']$ 在 $[[t + a], [t + b]]$ 中是任意的，对于所有 $[t'] \in [[t + a], [t + b]]$，我们有 $(w_d, [t']) \models [\psi]$。– 根据 SSTL 中有界始终的定义（定义 3.5），这正是 $(w_d, [t]) \models \Box_{[[a],[b]]} [\psi]$。

因此，($\sigma\ j \models \tau \Box_{[[a],[b]]} [\psi] \implies w_d [t] \models$,) ( ) (,)。 $\Box_{[[a],[b]]} [\psi]$

因此，反向对所有 SSTL 公式均成立 .

结合两个方向，我们已证明，对于每个 SSTL 公式 $[\varphi]$ 和每个离散时间 $[t]$，

$$(w_d, [t]) \models [\varphi] \iff (\sigma, j) \models \tau([\varphi]),$$

由此完成了定理 4 的证明 .9.

□

D 引理 4.7 的证明

证明概要 证明的核心在于，SSTL 公式中的每个有界时间时序算子 β 都会产生义务，而这些义务的生命周期受其区间宽度的限制。令 $w_\beta = b_\beta - a_\beta + 1$ 表示该宽度，并令 $W = \sum_\beta w_\beta$（由于公式是有限的，因此该值有限）。

在我们跟踪的任意离散位置 j，对于每项活动义务，我们都会记录算子启用时捕获的入口索引 j_0。有两点值得注意：

(1) **生命周期上界**。为算子β在 j_0 创建的义务，只有在 j满足 $j_0 + a_\beta \le j \le j_0 + b_\beta$时才能保持有效；因此，其生命周期至多为$w_\beta$步。(2) **创建速率**。对于每个算子，在每个位置至多可以开始一项新义务。

结合这两个事实，在长度为w_β的任意滑动窗口内，对于 β，同时有效的义务至多有w_β项。对所有算子求和，可得必须在任意位置跟踪的不同 j_0 值的数量具有统一上界W。这种有限性是引理 4.10 中可判定性结果的基础。 □

E 实现细节

将有界时间区间处理纳入 SSTL 到 LTL_φ的翻译并非只有一种方式，也可以采用其他等价翻译。这里，我们介绍一种可实现且验证速度更快的翻译。

翻译中使用的 $\mathrm{within}[a,b]$ 条件在概念和逻辑上都是正确的。然而，由于 SPIN 在 LTL 公式中处理状态空间探索的方式，这种条件并不容易用于验证。为此，我们可以将翻译修改为等价的翻译，使其可实现且（可能）更快地完成验证，如下所示：

$$\tau\big([\varphi_1]\,\mathrm{U}_{[[a],[b]]}\,[\varphi_2]\big) := \tau\big([\varphi_1]\big)\ \wedge (j \le j_0 + b)\,\mathrm{U}\ \ \tau\big([\varphi_2]\big)\ \wedge (j \ge j_0 + a) \tag{41}$$

$$\tau\big(\Diamond_{[[a],[b]]}[\varphi]\big) := \tau\big(\top\ \mathcal{U}_{[[a],[b]]}\ [\varphi]\big) \tag{42}$$

$$\tau\big(\Box_{[[a],[b]]}[\varphi]\big) := \neg\Diamond_{[[a],[b]]}\neg[\varphi] \tag{43}$$

我们在 SSTL 到 LTL_φ 的翻译实现中使用了这种翻译。

等价性的证明概要。我们比较原始守卫 $\mathrm{within}[a,b] := (j_0 + a \le j \le j_0 + b)$ 与三个转换情形中使用的可实现编码。对于每个运算符，我们证明，SPIN 友好公式接受的轨迹集合，与使用 $\mathrm{within}[a,b]$ 的规约接受的轨迹集合完全相同。

(1) **有界直到**。在任意位置 j，只要某个未来的 j' 满足 $j_0+a \le j' \le j_0+b$，且 φ_2 成立，同时 φ_1 在 j' 之前的所有时钟周期都成立，该义务就应当得到满足。将其改写为普通的直到运算符后，恰好得到 LTL 公式 $\tau([\varphi_1])\ \wedge (j \le j_0 + b)\,\mathrm{U}\ \ \tau([\varphi_2])\ \wedge (j \ge j_0 + a)$。左侧合取项确保路径尚未越过 $j_0 + b$（与窗口的上界相符），而右侧合取项则确保见证时钟刻不早于 $j_0 + a$。因此，时间窗口得到了等价的约束。(2) **有界最终与全局**。这是有界直到情形的直接推论。

在所有情况下，可实现的编码都对 j 施加与原始 $\mathrm{within}[a,b]$ 谓词相同的下界和上界，从而确立了语义等价性。□