

化路径片段 ω 的长度为 $|\omega| = N \times (1 + \lceil e/\epsilon \rceil)$,其中, N 为 \mathcal{A} 中状态(位置)个数($|S|$), ϵ 为假设的一个可控循环消耗的最小时间(不考虑兹诺行为).根据鸽笼原理,某个状态最多在 ω 中出现 $1 + \lceil e/\epsilon \rceil$;也就意味着 ω 中最多有 $\lceil e/\epsilon \rceil$ 个可控循环.因此,从 s 开始的所有满足观测时长约束 $[b,e]$ 的符号化路径片段长度最大为 $N \times (1 + \lceil e/\epsilon \rceil)$.那么,找到所有满足观测时长约束的符号化路径片段的循环次数最多为 $N^{N \times (1 + \lceil e/\epsilon \rceil)}$.加入到栈中的状态数有上限,而且每次循环都会去掉栈中一个元素,因此栈一定会为空,即循环一定终止.

对于可靠性,假设 ω 是集合 Θ 中的一条符号化执行路径,显然,根据算法 1 找到的执行片段一定是自动机 \mathcal{A} 可以产生的,并且根据第 9 行、第 10 行、第 19 行的判断条件,该符号化执行片段的长度与区间 $[b,e]$ 相交.

对于完备性,假设 $\omega: (s_i, \mu(s_i, \sigma_{i+1}, s_{i+1})) \dots (s_j, \mu(s_j, \sigma_{j+1}, s_{j+1}))$ 是自动机 \mathcal{A} 能够产生的一条符号化路径的一个片段,并且长度区间与约束区间 $[b,e]$ 相交.那么,可以构造一个 $\omega' \in \Theta$.首先,我们让 ω' 从 $(s_i, \mu(s_i, \sigma_{i+1}, s_{i+1}))$ 开始,显然,算法 1 中第 3 行可以办到.由于 ω 是自动机 \mathcal{A} 能够产生的一条符号化路径的一个片段,那么, ω 中的第 2 个位置 $(s_{i+1}, \mu(s_{i+1}, \sigma_{i+2}, s_{i+2}))$ 一定是 $(s_i, \mu(s_i, \sigma_{i+1}, s_{i+1}))$ 的一个后继,并且 $(s_i, \mu(s_i, \sigma_{i+1}, s_{i+1}))(s_{i+1}, \mu(s_{i+1}, \sigma_{i+2}, s_{i+2}))$ 的时间长度的最大值一定小于 e ,那么,根据第 9 行~第 18 行, ω' 会等于 $(s_i, \mu(s_i, \sigma_{i+1}, s_{i+1}))(s_{i+1}, \mu(s_{i+1}, \sigma_{i+2}, s_{i+2}))$.以此推导,直到 $(s_j, \mu(s_j, \sigma_{j+1}, s_{j+1}))$ 被加到 ω' 的最后.那么, ω' 会在执行第 10 行、第 11 行或者执行第 19 行、第 20 行后被加入到 Θ 中,即 $\omega' \in \Theta$. □

3 构造量词线性算术表达式

本节我们介绍如何将一条满足观测时长约束的路径片段转化为一个量词线性算术表达式.给定一个观测区间 $[t,t']$,那么它一定被自动机的状态(位置)所占据,如图 3 所示,在该段时间区间上,系统依次处于状态(位置) l_1, l_2, \dots, l_n (可以有重复状态,这里为了描述简洁,重新给出下标号).如果区间长度 $t'-t$ 在观测区间长度约束 $[b,e]$ 内,即 $b \leq t'-t \leq e$,那么,其对应的路径片段即为一条满足观测时长约束的路径片段.

对于每一个位置 $l_i (1 \leq i \leq n)$,我们引入一个变量 $\delta_i \in \mathbb{R}_{\geq 0}$,表示系统在该位置停留的时长.对于 ELDI 子式,如果含有模态词切变,例如子式 $\phi_1; \phi_2; \dots; \phi_n$,那么,每一个切变点必然位于某个位置 l_j 内.对于某个切变点 $chop_i$,我们引入一个变量 $\tau_i \in \mathbb{R}_{\geq 0}$.如图 3 所示,如果切变点 $chop_i$ 位于位置 l_j 内,那么, τ_i 表示从系统进入位置 l_j 到切变点 $chop_i$ 的时间长度.运用引入的这些变量,我们可以将一条满足观测时长约束的符号化路径片段表达为 $(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$.下面我们给出将其转化为一种量词线性算术表达式的方法.

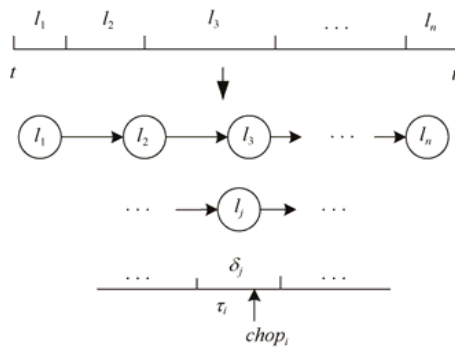


Fig.3 Interpretation for introduced variables δ, τ

图 3 对于引入变量 δ, τ 的解释

3.1 构造符号化路径片段的时限约束

首先,我们需要将这条符号化路径片段的时限约束表示出来,即刻画实时自动机产生这条符号化路径片段的所有时限约束.这些时限约束包括 3 种类型:非负约束、加和有界约束和 δ 约束.

定义 7(非负约束). 给定一个满足观测时长约束 $[b,e]$ 的路径片段 $(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$,那么,系统在每个位置

的停留时长 δ_i 是一个非负实数:

$$\bigcap_{i=1}^n (\delta_i \geq 0).$$

定义 8(加和有界约束). 给定一个满足观测时长约束 $[b,e]$ 的路径片段 $(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$, 那么, 系统在所有位置停留的总时长满足观测时长约束:

$$b \leq \sum_{i=1}^n \delta_i \leq e.$$

加和有界约束的意义在于, 第 2 节中求取的每个符号化路径片段的时间长度均是一个区间, 只是与时长约束 $[b,e]$ 有交集. 那么, 加和有界约束则进一步限定, 无论每个 δ_i 如何取值, 加和一定需要满足时长约束 $[b,e]$, 去除掉路径片段的时间长度在 $[b,e]$ 约束之外时的 δ_i 取值情况.

定义 9(δ 约束). 给定一个满足观测时长约束 $[b,e]$ 的路径片段 $(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$, 那么, 系统在每个位置停留的时长 δ_i 应满足实时自动机在相应位置停留的时限约束, 即保证该路径片段确实为一条实时自动机能够产生的路径的一个片段.

对于实时自动机, δ 约束求取的基本思路如下, 分两种情况加以讨论.

第 1 种情况, 如果路径片段只含有一个位置 (l_1, δ_1) , 那么, 获取实时自动机中所有以位置 l_1 作为源节点的迁移, 可以用迁移上的约束求取系统在该位置停留的时限约束. 由于不能够知晓开始观测时系统已经在该位置停留了多长时间, 因此需要对迁移上的约束做出修改, 即将约束下界修改为 0 即可. 例如, 如图 1 中, 假设有一个满足时长约束的路径片段为 (s_1, δ_1) , 有两条迁移 $s_1 \xrightarrow{b, [1,3]} s_1$ 和 $s_1 \xrightarrow{c, [2,4]} s_2$ 以 s_1 作为源节点, 那么, 对于某条执行路径, 系统在 s_1 上停留的时限约束为 $[1,3]$ 或 $[2,4]$. 由于不知道开始观察时系统已在 s_1 上停留了多长时间, 因此, 观察到系统在 s_1 上停留的时限约束为 $[0,3]$ 或 $[0,4]$. 那么, 对于这条路径片段来说, δ 约束为 $0 \leq \delta_1 \leq 3$ 或 $0 \leq \delta_1 \leq 4$.

第 2 种情况, 如果执行片段长度大于 1, 假设路径片段为 $(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$, 那么, 对于路径片段中相邻的两个位置 (l_i, δ_i) 和 (l_{i+1}, δ_{i+1}) , 先找到 l_i 到 l_{i+1} 的所有迁移, 将迁移上的时限约束作为对于 (l_i, δ_i) 的 δ 约束. 同样地, 对于 (l_1, δ_1) 的 δ 约束需要处理下界.

由于第 2 节查找满足观测时限的符号化执行路径时已采用该思想, 因此具体实现时只需将第 2 节中对于该条符号化执行路径片段所求的每个位置时间长度区间变为相应的 δ 约束. 表 2 中算法为生成 3 种时限约束的算法. 该算法输入为某个符号化路径片段 $\omega(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$, 输出为这个符号化路径片段所满足的时限约束 Γ . 该算法比较简单, 行 2 到行 3 是对每个位置生成其需要满足的 δ 约束, 运用函数 $\delta_constraint()$ 求取, 具体方法如前所述. 第 4 行将 3 种类型的约束取交集, 即为时限约束 Γ .

Table 2 The algorithm for generating the timing constraints

表 2 生成时限约束的算法

<p>算法 2. <i>Constarints</i>(ω).</p> <p>输入: $\omega: (l_1, \delta_1), \dots, (l_n, \delta_n)$;</p> <p>输出: Γ.</p> <p>1 begin</p> <p>2 foreach $i \in \{1, \dots, n\}$ do</p> <p>3 $\Gamma_i := \delta_constraint(l_i, \delta_i)$;</p> <p>4 $\Gamma := \bigwedge_{i=1}^n (\Gamma_i \wedge \delta_i \geq 0) \wedge b \leq \sum_{i=1}^n \delta_i \leq e$;</p> <p>5 return Γ;</p>	
--	--

3.2 构造语义不等式

为了能够用量词线性算术表达式表达每一个线性时段子式 \mathcal{D} 的语义信息, 我们需要构造 3 种不等式: 线性时段子式不等式、 τ - δ 不等式以及 τ 附加不等式.

定义 10(线性时段子式不等式). 给定一个 ELDI 子式 ϕ , 对于其中每一个线性时段子式 \mathcal{D} , 将 \mathcal{D} 中每一个状

态(位置)积分 $\int l_j$ 用引入的变量 δ 和 τ 表示,得到的不等式 \mathcal{D}' 即为一个线性时段子式不等式.

定义 11(τ - δ 不等式). 如果一个切变点 $chop_i$ 落在位置 l_j 内,那么对应的变量 τ_i 和 δ_j 应满足的不等关系为 $0 \leq \tau_i \leq \delta_j$,称为一个 τ - δ 不等式.

定义 12(τ 附加不等式). 如果存在两个相邻的切变点 $chop_i$ 和 $chop_{i+1}$ 位于同一个位置 l_j 内,如果 $chop_{i+1}$ 在 $chop_i$ 左边,那么对应的变量 τ_i 和 τ_{i+1} 应满足的不等关系为 $0 \leq \tau_{i+1} \leq \tau_i$,称为一个 τ 附加不等式.

给定一个符号化路径片段 $\omega(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$ 和 EDLI 子式 ϕ ,将其翻译为一组不等式的思路是,根据 EDLI 子式 ϕ 的逻辑结构,分解到每个线性时段子式,将其翻译为线性时段子式不等式、 τ - δ 不等式、 τ 附加不等式;然后再用相应的逻辑结构组合起来.

- 当 ϕ 是一个线性时段子式,即 $\phi = \mathcal{D} = \sum_{i \in \Omega} c_i \int S_i \leq c$ 时,假设其含有 k 个状态积分子 $\int S_1, \dots, \int S_k$, 对于 $\omega(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$, $\int S_i = \sum_{S_j=l_i} \delta_j$, 即将 ω 中所有与 S_i 相同的位置 l_j 对应的停留时长 δ_j 求和,用该加和表达式代替 \mathcal{D} 中 $\int S_i$;如果没有与 S_i 相同的 l_j ,则用 0 代替 $\int S_i$. 因此得到的线性时段子式不等式为 $\mathcal{D}' = \mathcal{D} \left[\sum_{S_j=l_i} \delta_j / \int S_i \right]$, 表示将线性时段子式中的状态积分用相应的变量 δ 的加和表达式替换.

- 当 $\phi = \neg \phi_1 \vee \phi_2, \phi_1 \wedge \phi_2$ 时,我们将它们递归分解为线性时段子式.
- 当 $\phi = \phi_1; \phi_2$ 时,将 ω 分为两个子片段 ω_1 和 ω_2 , 分别对应到公式 ϕ_1 和 ϕ_2 上,从而将其转化为与关系(\wedge)上的两个子问题.

3.3 构造量词线性算术表达式

与文献[17]相比,本文构造的时限约束和不等式更加明确,同时,构造算法中不需要引入额外的 δ 变量,即在遇到切变时,通过已有 δ 变量和 τ 变量的算术运算形式来表示片段中被切割的位置分属两个子片段的时间长度 $(\tau, \delta_i - \tau)$. 由于求解量词线性算术表达式的复杂度与变量个数相关,因此,本文采用的方法减少了变量个数,有效地缩短了求解所需时间.具体算法参考表 3 中算法.

Table 3 The algorithm for generating a QLRA formula
表 3 生成一个 QLRA 公式的算法

算法 3. $QLRA(\phi, \omega)$.	
输入: $\omega = (l_1, \delta_1), \dots, (l_n, \delta_n); \phi$;	
输出: ξ .	
1	begin
2	case $\phi := \mathcal{D}$ do
3	$\xi := \left(\mathcal{D} \left[\sum_{S_j=l_i} \delta_j / S_i, \dots, \sum_{S_j=l_i} \delta_j / \int S_i \right] \right)$;
4	case $\phi := \neg \phi_1$ do
5	$\xi := \neg QLRA(\phi_1, \omega)$;
6	case $\phi := \phi_1 \wedge \phi_2$ do
7	$\xi := QLRA(\phi_1, \omega) \wedge QLRA(\phi_2, \omega)$;
8	case $\phi := \phi_1 \vee \phi_2$ do
9	$\xi := QLRA(\phi_1, \omega) \vee QLRA(\phi_2, \omega)$;
10	case $\phi := \phi_1; \phi_2$ do
11	$\xi := \bigvee_{\tau=0}^{\delta_1} (QLRA(\phi_1, ((l_1, \delta_1), \dots, (l_i, \tau))) \wedge QLRA(\phi_2, ((l_i, \delta_i - \tau), \dots, (l_n, \delta_n)))) \wedge 0 \leq \tau \leq \delta_1$;
12	$Q := Q, \exists \tau$;
13	return ξ ;

表 3 中生成一种量词线性算术表达式的算法,其输入为一个扩展线性时段表达式 ϕ 和一个符号化路径片段 $\omega(l_1, \delta_1)(l_2, \delta_2) \dots (l_n, \delta_n)$; 输出为一个 EDLI 子式表达式 ξ . 整个算法是一种递归算法,根据扩展线性时段表达式 ϕ 的逻辑结构进行递归.当递归到某个线性时段子式时,运用引入的变量的加和形式替换掉式子中的状态积分,得到

线性时段子式不等式(行 3).当碰到逻辑连接词时,根据逻辑连接词的语义,将问题分解为子问题,递归地调用本算法.每当碰到一个切变时,引入一个变量 τ ,将 ω 分为两部分(行 11),同时需要添加相应的 τ - δ 不等式(行 11 中最后一个不等式).当遇到两个切变点落在同一个状态时, τ - δ 不等式会变成一个 τ 附加不等式.例如,当 ϕ_1 仍然含有切变,且切变点落在 $\omega_1:(l_1, \delta_1) \dots (l_i, \tau)$ 的最后一个位置时,新引入的变量 τ 需要满足的 τ - δ 不等式为 $0 \leq \tau \leq \delta$,那么,实际上是一个 τ 附加不等式.当片段有 n 个位置时,当前切变点的落位会有 $n+2$ 种情况,即在这 n 个位置之前、在这 n 个位置中的某个、在这 n 个位置之后.集合 Q 记录引入的(多个) τ 变量.最后,我们将引入的 δ 变量、算法 2 求出的时限约束表达式 Γ 和得到的表达式 ξ 构造一个 QLRA 表达式 $\pi = \forall \delta_1, \dots, \delta_n, Q\Gamma \Rightarrow \xi$.

3.4 构造算法的正确性证明

同样地,我们需要证明构造算法的正确性,正确性证明主要依赖递归正确性.

定理 2(算法 3 的正确性). 给定一个满足观测时长约束 $[b, e]$ 的符号化路径片段 ω 和 ELDI 公式 ϕ ,算法 3 是正确的,即:

- 终止性:算法会终止;
- 可靠性:如果 $\omega \models \phi$,那么,QLRA(ϕ, ω)得到的 QLRA 表达式是满足的(satisfiable);
- 完备性:如果 QLRA(ϕ, ω)得到的 QLRA 表达式是满足的,则 $\omega \models \phi$.

证明:对于终止性来说,由于算法是根据 ELDI 公式 ϕ 的结构进行递归的,因此可以看出算法是终止的.对于可靠性和完备性,下面分析每种递归结构.

• 当 ϕ 是一个线性时段子式,即 $\phi = D = \sum_{i \in \Omega} c_i \int S_i \leq c$ 时, $I_{\omega}([t_1, t_2]) \models b \leq \ell \leq e \Rightarrow \sum_{i \in \Omega} c_i \int S_i \leq c$ 当且仅当 $b \leq t_2 - t_1 \leq e \Rightarrow \sum_{i \in \Omega} c_i I_{\omega}(\int S_i)([t_1, t_2]) \leq c$,其中, t_1, t_2 分别为路径片段的起始时刻和终止时刻.显然,当按照语义替换后,仍然有这一关系.因此,如果 $\omega \models \phi$,那么,QLRA(ϕ, ω)得到的 QLRA 表达式是满足的;如果 QLRA(ϕ, ω)得到的 QLRA 表达式是满足的,则 $\omega \models \phi$.

• 当 $\phi = \neg \phi_1$ 时,对于可靠性,假设 QLRA($\neg \phi_1, \omega$)不满足,那么,根据算法 3 有 \neg QLRA(ϕ_1, ω)不满足,也就意味着 QLRA(ϕ_1, ω)正确.根据归纳假设条件,我们有 $\omega \models \phi_1$,可以得到 ω 不满足 $\neg \phi_1$.对于完备性,假设 QLRA($\neg \phi_1, \omega$)正确,根据算法 3 有 QLRA(ϕ_1, ω)不满足.根据归纳假设,我们有 ω 不满足 $\neg \phi_1$,因此得到 $\omega \models \phi$.

• 当 $\phi = \phi_1 \vee \phi_2$ 时,对于可靠性,假设 QLRA($\phi_1 \vee \phi_2, \omega$)不满足,根据算法 3 有 QLRA(ϕ_1, ω) \vee QLRA(ϕ_2, ω)不满足.根据归纳假设,我们有 ω 不满足 ϕ_1 并且 ω 不满足 ϕ_2 ,因此, ω 不满足 $\phi_1 \vee \phi_2$.对于完备性,假设 QLRA($\phi_1 \vee \phi_2, \omega$)正确,根据算法 3 有 \neg QLRA(ϕ_1, ω)或者 \neg QLRA(ϕ_2, ω)不满足.根据归纳假设,我们有 ω 不满足 ϕ_1 或者 ω 不满足 ϕ_2 ,因此得到 $\omega \models \phi_1 \vee \phi_2$.

• 当 $\phi = \phi_1 \wedge \phi_2$ 时,证明与 $\phi = \phi_1 \vee \phi_2$ 类似.

• 当 $\phi = \phi_1; \phi_2$ 时,显然 $\omega \models \phi_1; \phi_2$ 当且仅当 ω 被分为两个部分 ω_1 和 ω_2 且 $\omega_1 \models \phi_1 \wedge \omega_2 \models \phi_2$.根据算法 3,我们选取了所有可能的切变点位置,并按语义分解为两个子问题 $\omega_1 \models \phi_1$ 和 $\omega_2 \models \phi_2$.因此,根据归纳假设,我们得出:如果 $\omega \models \phi_1; \phi_2$,那么,QLRA($\phi_1; \phi_2, \omega$)得到的 QLRA 表达式是满足的;如果 QLRA($\phi_1; \phi_2, \omega$)得到的 QLRA 表达式是满足的,则 $\omega \models \phi_1; \phi_2$. \square

3.5 构建量词线性算术表达式举例

下面举例说明如何构建一个量词线性算术表达式.

例 2:我们考虑图 1 所示的实时自动机以及 ELDI 公式: $3 \leq \ell \leq 4 \Rightarrow \int s_0 + \int s_1 \leq 1; 2 \int s_1 - \int s_2 \leq 2$,令 $D_1 = \int s_0 + \int s_1 \leq 1, D_2 = 2 \int s_1 - \int s_2 \leq 2$.现有一条满足观测时长约束的符号化路径片段 $\omega:(s_0, \delta_0)(s_1, \delta_1)(s_2, \delta_2)$,可见我们引入了 3 个 δ 变量.由于有一个切变,因此我们引入一个 τ 变量 τ_1 .

首先,我们生成符号化路径片段的所有时限约束.对于 δ 约束,由于 (s_0, δ_0) 是第 1 个位置,后继位置为 (s_1, δ_1) ,因此, δ_0 应先取迁移 (s_0, a, s_1) 的时限约束,即 $1 \leq \delta_0 \leq 2$.由于 (s_0, δ_0) 是第 1 个位置,开始观测时并不知道系统在 s_0 已停留多长时间,因此,将 δ_0 的下界改为 0,得到 δ_0 的最后约束为 $0 \leq \delta_0 \leq 2$.用同样的方法得到 (s_1, δ_1) 的时限约束为 $2 \leq \delta_1 \leq 4, (s_2, \delta_2)$ 的时限约束为 $\delta_2 \geq 0$.非负约束为 $(\delta_0 \geq 0 \wedge \delta_1 \geq 0 \wedge \delta_2 \geq 0)$,加和有界约束为 $3 \leq \delta_0 + \delta_1 + \delta_2 \leq 4$.所有时限约束为 $3 \leq \delta_0 + \delta_1 + \delta_2 \leq 4 \wedge (0 \leq \delta_0 \leq 2 \wedge 2 \leq \delta_1 \leq 4 \wedge \delta_2 \geq 0) \wedge (\delta_0 \geq 0 \wedge \delta_1 \geq 0 \wedge \delta_2 \geq 0)$.

然后,我们根据算法 3 来构造不等式和 QLRA 表达式.由于当前 ELDI 公式中是一个切变形式,那么会匹配到第 11 行,并有 5 种可能情况:(1) 当切变点落在 s_0 左侧时, ω_1 为空, ω_2 为 $(s_0, \delta_0)(s_1, \delta_1)(s_2, \delta_2)$,那么, $\mathcal{D}_1 = 0 + 0 \leq 1$, $\mathcal{D}_2 = 2\delta_1 - \delta_2 \leq 2$, 即得到线性时段子式的不等式 $0 + 0 \leq 1 \wedge 2\delta_1 - \delta_2 \leq 2$. (2) 当切变点落在 s_0 时, ω_1 为 (s_0, τ_1) , ω_2 为 $(s_0, \delta_0 - \tau_1)(s_1, \delta_1)(s_2, \delta_2)$,那么, $\mathcal{D}_1 = \tau_1 + 0 \leq 1$, $\mathcal{D}_2 = 2\delta_1 - \delta_2 \leq 2$, 即得到线性时段子式的不等式 $\tau_1 + 0 \leq 1 \wedge 2\delta_1 - \delta_2 \leq 2 \wedge 0 \leq \tau_1 \leq \delta_0$. (3) 当切变点落在 s_1 时, ω_1 为 $(s_0, \delta_0)(s_1 - \tau_1)$, ω_2 为 $(s_1, \delta_1 - \tau_1)(s_2, \delta_2)$,那么, $\mathcal{D}_1 = \delta_0 + \tau_1 \leq 1$, $\mathcal{D}_2 = 2(\delta_1 - \tau_1) - \delta_2 \leq 2$, 即得到线性时段子式的不等式 $\delta_0 + \tau_1 \leq 1 \wedge 2(\delta_1 - \tau_1) - \delta_2 \leq 2 \wedge 0 \leq \tau_1 \leq \delta_1$. (4) 当切变点落在 s_2 时,用同样的方法得到线性时段子式的不等式为 $\delta_0 + \delta_1 \leq 1 \wedge 0 - (\delta_2 - \tau_1) \leq 2 \wedge 0 \leq \tau_1 \leq \delta_2$. (5) 当切变点落在 s_2 的右侧时,用上述方法得到线性时段子式的不等式为 $\delta_0 + \delta_1 \leq 1 \wedge 0 - 0 \leq 2$. 因此,根据算法 3 构造的 QLRA 表达式为

$$\begin{aligned} \pi &= \forall \delta_0, \forall \delta_1, \forall \delta_2, \exists \tau_1. 3 \leq \delta_0 + \delta_1 + \delta_2 \leq 4 \wedge (0 \leq \delta_0 \leq 2 \wedge 2 \leq \delta_1 \leq 4 \wedge \delta_2 \geq 0) \wedge (\delta_0 \geq 0 \wedge \delta_1 \geq 0 \wedge \delta_2 \geq 0) \\ &\Rightarrow (0 + 0 \leq 1 \wedge 2\delta_1 - \delta_2 \leq 2) \vee (\tau_1 + 0 \leq 1 \wedge 2\delta_1 - \delta_2 \leq 2 \wedge 0 \leq \tau_1 \leq \delta_0) \vee (\delta_0 + \tau_1 \leq 1 \wedge 2(\delta_1 - \tau_1) - \delta_2 \leq 2 \wedge 0 \leq \tau_1 \leq \delta_1) \\ &\vee (\delta_0 + \delta_1 \leq 1 \wedge 0 - (\delta_2 - \tau_1) \leq 2 \wedge 0 \leq \tau_1 \leq \delta_2) \vee (\delta_0 + \delta_1 \leq 1 \wedge 0 - 0 \leq 2). \end{aligned}$$

4 求解 QLRA 表达式及判定定理

根据定理 1 和定理 2,给定一个实时自动机 \mathcal{A} 和一个观测时长约束有上界的 ELDI 公式 Φ ,我们可以将模型检验问题 $\mathcal{A} \models \Phi$ 转化为 QLRA 表达式的正确性(validity)问题.

定理 3. 给定一个实时自动机 \mathcal{A} 和一个观测时长约束有上界的 ELDI 公式 Φ , $\mathcal{A} \models \Phi$ 当且仅当 $\bigcap_{\omega \in \text{Search}(\mathcal{A}, [b, e])} \text{QLRA}(\Phi, \omega)$ 是正确的(valid).

定理 3 可以从定理 1 和定理 2 得出,这里不再证明.根据 Tarski 理论,量词线性算术表达式的可满足性和正确性问题是可判定的^[20],因此有下面的结论.

定理 4. 给定一个实时自动机 \mathcal{A} 和一个观测时长约束有上界的 ELDI 公式 Φ , $\mathcal{A}, [b, e] \models \Phi$ 是可判定的.

QLRA 表示式可以运用量词消去技术求解,量词消去技术基于柱形代数分解理论(cylindrical algebraic decomposition,简称 CAD)^[21],该方法的最坏复杂度与变量规模呈二阶指数关系,但是,由于 QLRA 表达式中均为线性不等式,因此,求解比较迅速.目前,量词消去技术在许多求解器中都有实现,例如,REDLOG、QEPCAD^[22]、Z3 等.

5 工具实现与实验

目前,我们在 Linux 平台上实现了一个原型工具,如图 4 所示为工具的架构图.工具接收两个输入:一是,实时自动机(\mathcal{A})模型文件,格式参考 UPPAAL 模型文件格式;二是,扩展线性时段不变式(ϕ),存放在一个文本文件中.工具会接收这两个输入,首先寻找所有满足观测时长约束 ℓ 的符号化路径片段,并得到其集合 Θ ,然后构造 QLRA 表达式,生成所有 QLRA 表达式的集合 Π ,并保存在符合 REDLOG 输入格式的文本文件中;然后将该文件传递给量词消去工具 REDLOG,REDLOG 会依次求解 QLRA 表达式 π_i ,并将每个表达式的结果(true or false)记录在一个文本文件中.

在实验部分,我们讨论文献[14,16]中给出的一个实验.一个实时自动机 \mathcal{A} 由 N 个简单的实时自动机 M 迭代组成. M_i 如图 5 所示,由 4 个状态组成 A_i, B_i, C_i, D_i ,系统在每个状态上停留一个时间单位.特别地,对于 $1 \leq i \leq N, D_i$ 有一条迁移前往 A_{i+1} ;对于 $1 \leq j \leq i \leq N, D_i$ 有一条迁移前往 A_j .文献[14,16]在离散时间语义下进行了实验,本文则在标准连续时间语义下进行了实验,迭代次数 N 从 3 到 6.表 4 是我们的实验结果, t_{qlra} 表示生成所有 QLRA 表达式所用时间, t_{qe} 表示运用量词消去工具求解所有 QLRA 表达式所用时间, t_{total} 表示验证的总时间,而 t'_{total} 表示采用文献[17]中算法的总时间,时间单位均为 s.我们同样对含有不同逻辑运算符和切变的 ELDI 式子进行了实验,工具和实验可以参考 <https://github.com/Leslieaj/VCELDI>.

由于本文采用标准连续时间语义,而文献[16]采用离散时间语义,所以模型检验的总时间比文献[16]中相同问题所用时间要长.这是由于采用方法、基于的模型以及讨论的时间语义不同所致,我们的优势是可以解决连

续时间语义下的模型检验问题.与文献[17]中算法的实验结果相比,我们的验证方法所用时间明显低于文献[17]验证所需时间,可见本文方法有效地降低了验证算法的复杂度,提高了实际验证速度.

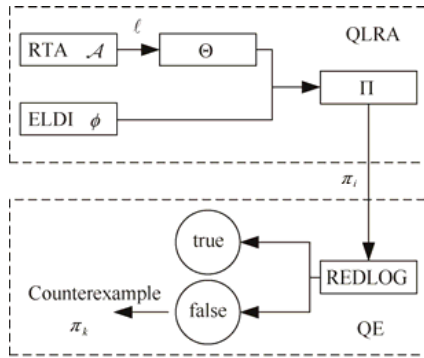


Fig.4 The tool structure
图4 工具架构图

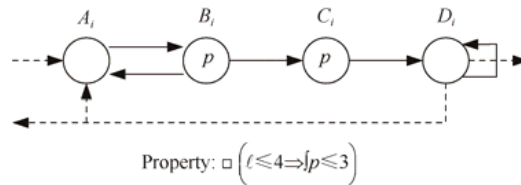


Fig.5 The model and property of the experiment
图5 实验的模型与性质

Table 4 The results of the experiment

表4 实验结果

N	t_{qlar}	t_{qe}	t_{total}	t'_{total}
3	0.7	0.7	1.4	1.8
4	1.1	1.0	2.1	3.1
5	1.7	1.5	3.2	4.5
6	2.4	1.9	4.3	6.1

6 总结

本文讨论了在连续时间语义下,对于一个观测时长约束有上界的扩展线性时段不变式(ELDI)在实时自动机上的模型检验问题.本文证明了该问题是可以判定的,并且给出了判定方法.首先,我们采用符号化的思想,运用深度优先搜索得到所有满足观测时长约束的符号化路径片段;然后,将每一个符号化路径片段转化为一个量词线性算术表达式;最后,运用量词消去工具 REDLOG 求解这些量词线性算术表达式.

时间复杂度方面,对于第1步,我们可以看到算法1与自动机中状态(位置)数目呈一阶指数关系.第2步中,生成 QLRA 表达式的算法复杂度是多项式复杂度.最后一步运用量词消去工具求解,量词消去工具的最坏时间复杂度与变量个数呈二阶指数关系.但是,根据前文分析,QLRA 表示式中均为线性不等式,所以求解比较快,近似为多项式和一阶指数,这一点可以从实验 t_{qlar} 和 t_{qe} 的比较中看出.所以,对于时间复杂度来说,与实时自动机的规模和扩展线性时段不变式的切变个数呈一阶指数关系,最坏是二阶指数关系.而文献[17]中的验证算法复杂度一般为二阶指数,最坏情况为三阶指数.与文献[17]相比,复杂度和运行速度方面的优化有以下两点:第一,缩短了搜索空间,降低了搜索算法的复杂度,并减少了所产生的 QLRA 公式数量;第二,对于单个 QLRA 表达式,减少了量词的引入个数.从而,整体上降低了验证算法的复杂度,并加快了量词消去工具求解时的实际运行速度.

References:

- [1] Zhou CC, Hoare CAR, Ravn AP. A calculus of durations. Information Processing Letters, 1991,40(5):269–276.
- [2] Li XS, Zhou CC. Duration calculi: An overview. Chinese Journal of Computers, 1994,(11):842–851 (in Chinese with English abstract).
- [3] Halpern JY, Manna Z, Moszkowski BC. A hardware semantics based on temporal intervals. In: Proc. of the 10th Int'l Colloquium on Automata, Languages, and Programming (ICALP). Berlin: Springer-Verlag, 1983. 278–291. [doi: 10.1007/BFb0036915]
- [4] Meyer R, Faber J, Hoenicke J, Rybalchenko A. Model checking duration calculus: A practical approach. Formal Aspects of Computing, 2008,20(4):481–505. [doi: 10.1007/s00165-008-0082-7]
- [5] Zhou CC, Hansen MR. Duration Calculus: A Formal Approach to Real-Time Systems. Berlin, Heidelberg: Springer-Verlag, 2004. [doi: 10.1007/978-3-662-06784-0]

- [6] Zhou CC, Hansen MR, Sestoft P. Decidability and undecidability results for duration calculus. In: Proc. of the 10th Annual Symp. on Theoretical Aspects of Computer Science (STACS). Berlin, Heidelberg: Springer-Verlag, 1993. 58–68. [doi: 10.1007/3-540-56503-5_8]
- [7] Zhou CC, Zhang J, Yang L, Li XS. Linear duration invariants. In: Proc. of the 3rd Int'l Symp. on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT). Berlin, Heidelberg: Springer-Verlag, 1994. 86–109. [doi: 10.1007/3-540-58468-4_161]
- [8] Braberman VA, Huang DV. On checking timed automata for linear duration invariants. In: Proc. of the 19th IEEE Real-Time Systems Symp. (RTSS). Washington: IEEE Computer Society, 1998. 264–273. [doi: 10.1109/REAL.1998.739752]
- [9] Li XD, Huang DV. Checking linear duration invariants by linear programming. In: Proc. of the Concurrency and Parallelism, Programming, Networking, and Security: The 2nd Asian Computing Science Conf. (ASIAN). Berlin, Heidelberg: Springer-Verlag, 1996. 321–332. [doi: 10.1007/BFb0027804]
- [10] Li XD, Huang DV, Zheng T. Checking hybrid automata for linear duration invariants. In: Proc. of the Advances in Computing Science: The 3rd Asian Computing Science Conf. (ASIAN). Berlin, Heidelberg: Springer-Verlag, 1997. 166–180. [doi: 10.1007/3-540-63875-X_51]
- [11] Thai P, Huang DV. Verifying linear duration constraints of timed automata. In: Proc. of the 1st Int'l Colloquium on Theoretical Aspects of Computing (ICTAC). Berlin, Heidelberg: Springer-Verlag, 2004. 295–309. [doi: 10.1007/978-3-540-31862-0_22]
- [12] Zhang MM, Huang DV, Liu ZM. Verification of linear duration invariants by model checking CTL properties. In: Proc. of the 5th Int'l Colloquium on Theoretical Aspects of Computing (ICTAC). Berlin, Heidelberg: Springer-Verlag, 2008. 395–409. [doi: 10.1007/978-3-540-85762-4_27]
- [13] Zhang MM, Liu ZM, Zhan NJ. Model checking linear duration invariants of networks of automata. In: Proc. of the 3rd Int'l IPM Conf. on Fundamentals of Software Engineering (FSEN). Berlin, Heidelberg: Springer-Verlag, 2009. 244–259. [doi: 10.1007/978-3-642-11623-0_14]
- [14] Fränzle M, Hansen MR. Efficient model checking for duration calculus based on branching-time approximations. In: Proc. of the 6th IEEE Int'l Conf. on Software Engineering and Formal Methods (SEFM). Washington: IEEE Computer Society, 2008. 63–72. [doi: 10.1109/SEFM.2008.26]
- [15] Fränzle M, Hansen MR. Efficient model checking for duration calculus. Int'l Journal of Software and Informatics, 2009,3(2-3): 171–196.
- [16] Zu Q, Zhang MM, Zhu JQ, Zhan NJ. Bounded model-checking of discrete duration calculus. In: Proc. of the 16th Int'l Conf. on Hybrid Systems: Computation and Control (HSCC). New York: ACM, 2013. 213–222. [doi: 10.1145/2461328.2461362]
- [17] An J, Zhan NJ, Li XS, Zhang MM, Wang Y. Model checking bounded continuous-time extended linear duration invariants. In: Proc. of the 21st Int'l Conf. on Hybrid Systems: Computation and Control (HSCC). New York: ACM, 2018. 81–90. [doi: 10.1145/3178126.3178147]
- [18] Dima C. Real-time automata. Journal of Automata, Languages and Combinatorics, 2001,6(1):3–24. [doi: 10.25596/jalc-2001-003]
- [19] Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183–235.
- [20] Tarski A. A Decision Method for Elementary Algebra and Geometry. Berkeley: University of California Press, 1951.
- [21] Collins G. Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Proc. of the 2nd GI Conf. on Automata Theory and Formal Languages. Berlin, Heidelberg: Springer-Verlag, 1975. 134–183. [doi: 10.1007/3-540-07407-4_17]
- [22] Brown CW. QEPCAD B: A program for computing with semi-algebraic sets using CADs. ACM SIGSAM Bulletin, 2003,37(4): 97–108. [doi: 10.1145/968708.968710]

附中文参考文献:

- [2] 李晓山,周巢尘.时段演算综述.计算机学报,1994,(11):842–851.



安杰(1990—),男,湖北京山人,博士,主要研究领域为形式化方法,混成系统,模型检测.



张苗苗(1971—),女,博士,研究员,博士生导师,CCF 专业会员,主要研究领域为形式化方法.