带数据约束实时系统的模型检测

倪水妹[1,2] 曹子宁[1,2] 李心磊[3]
(民用飞机模拟飞行国家重点实验室 上海 201210)[1]
(南京航空航天大学计算机科学与技术学院 南京 210016)[2] (安徽工业大学计算机学院 马鞍山 243032)[3]

摘 要 带数据约束的实时系统是指一种既带有时间约束又带有数据变量约束的计算系统。目前将离散数据约束和连续时间约束统一在一个模型中的规范及验证研究较少。文中提出了一种既带有连续数据约束又带有离散数据约束的规范——基于连续时间的ZIA规范,并给出它的时序逻辑。MARTE是UML在嵌入式实时系统领域的建模规范,在工业界的应用非常广泛,但是目前对其模型检测的研究较少。在MARTE的基础上扩展Z,提出了Z-MARTE,并将Z-MARTE转换为基于连续时间的ZIA模型,在实现对连续时间ZIA模型检测的同时,也实现了对Z-MARTE的模型检测。最后通过一个实例进行验证,说明此方法可行有效。

关键词 数据约束,实时系统,连续时间,MARTE,ZIA,模型检测
中图法分类号 TP301 **文献标识码** A

Model Checking for Real-time Systems with Data Constraints

NI Shui-mei[1,2] CAO Zi-ning[1,2] LI Xin-lei[3]
(National Key Laboratory of Civil Aircraft Flight Simulation, Shanghai 201210, China)[1]
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[2]
(College of Computer Science, Anhui University of Technology, Maanshan 243032, China)[3]

Abstract Real-time systems with data constraints refer to computing systems both with time-bound and data variables constraints. Now there are few studies about the specification and verification of a unified model which connects discrete data constraints and continuous-time together. The article presented a ZIA specification based on continuous-time which not only has continuous-time constraints, but also has discrete data constraints and then gave its temporal logic. MARTE is a modeling specification of a UML profile in the field of embedded real-time systems. MARTE is widely used in industry, but studies about its model checking are few. The article proposed Z-MARTE by expanding Z on MARTE, and converted Z-MARTE to a ZIA model based on continuous-time. When model checking for ZIA model based on continuous-time is realized, model checking for Z-MARTE is realized. Finally, an example was given to indicate that this method is feasible and effective.

Keywords Data constraints, Real-time system, Continuous-time, MARTE, ZIA, Model checking

1 引言

带数据约束的实时系统[1]是指一种既带有时间约束,又带有数据变量约束的计算系统。飞行控制、核反应堆控制以及铁路调度控制等计算机控制系统都属于带数据约束的实时系统。这些系统中许多动作的完成都与时间相关,即要满足一定的时间限制,如某个动作要在一秒钟内完成;同时这些系统中数据变量之间也有一定的约束关系,如飞机的飞行速度可能跟气压、高度等有一定的约束关系。对这类系统而言,确保其正确性和可靠性是至关重要的。

模型检测是一种用于并发系统性质验证的算法技术[2,3],但它只能对有穷状态系统进行性质验证。其基本思想是用状态迁移系统S表示系统的行为,用时序逻辑公式F来描述系统的性质。这样,"系统是否满足所期望的性质"就转化为数学问题"状态迁移系统S是否是公式F的一个模型"[4]。

为了对带数据约束的实时系统进行研究,文献[5,6]中提出了ZIA、HZIA规范并给出了ZIA、HZIA上的精化关系,但是并没有给出对应的时序逻辑以及模型检测算法。文献[7]中提出了一种基于离散时间的ZIA[8]规范,但是基于离散时间的模型[9]较适用于同步系统,如果要利用离散时间来模拟异步系统,就必须选择一些固定的时间量子来离散化时间,使得系统中任意两个事件之间的延迟时间都是这个时间量子的倍数,这点是很难做到的,并可能会限制系统可以模拟的准确

到稿日期:2013-05-17 返修日期:2014-01-22 本文受航空科学基金(20128052064),中央高校基本科研业务费专项资金(NZ2013306),国家重点基础研究发展计划(973计划)(2014CB744900),江苏省计算机信息处理技术重点实验室开放课题(209035)资助。
倪水妹(1988—),女,硕士生,主要研究方向为形式化方法,E-mail:nsm1227@126.com;曹子宁(1972—),男,教授,博士生导师,主要研究方向为形式化方法、人工智能。

性。例如，图1模拟一个恒温器。变量 x 表示温度，在控制模式 Off 下，加热器是关闭的(off)，并且温度以速率 $x'=-0.1x$ 下降；在控制模式 On 下，加热器是打开的(on)，并且温度以速率 $x'=5-0.1x$ 上升。初始条件下，加热器处于关闭状态并且温度是 20°。根据跳转条件 $x<19$，一旦温度低于 19°，加热器就跳转到 on 状态开始加热；根据结点不变条件 $x\geqslant 18$，当温度下降至 18°时加热器跳转至 on 状态开始加热。其中 x' 表示温度随时间变化的一阶导数，温度是随时间连续变化的。显然，这样的系统就无法用离散时间 ZIA 来模拟。因此，本文提出了一种基于连续时间的 ZIA 规范。

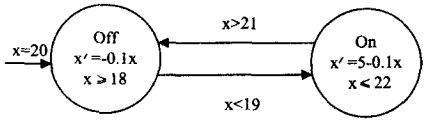

图1 恒温器自动机

MARTE(Modeling and Analysis of Real Time and Embedded systems)是 UML 在嵌入式实时系统领域的建模规范[10]，弥补了 UML 对嵌入式实时领域非功能属性表达能力的不足。MARTE 规范采用图形化的方式描述系统，缺乏精确的语义，因此难以直接进行形式化分析与验证。而形式化方法作为一种以数学为基础的方法，能够清晰、精确、抽象、简明地规范和验证软件系统及其性质，极大地提高软件的安全性和可靠性。因此需要将 MARTE 模型转换到形式化模型进行验证，以保证软件的可信度。Ahmed M. Mostafa 等提出使用 Z 形式化 MARTE 中用例图、类图、状态图等[11]，但并未解决对 MARTE 的模型检测问题。

我们在 MARTE 的基础上扩展 Z，提出了 Z-MARTE，并将 Z-MARTE 转换为基于连续时间的 ZIA 模型，这样在实现对连续时间的 ZIA 模型的模型检测时，也就实现了对 Z-MARTE 的模型检测。

本文第2节主要介绍了基于连续时间的 ZIA 规范并给出了其对应的时序逻辑公式；第3节主要介绍了 Z-MARTE，并实现了从 Z-MARTE 到基于连续时间 ZIA 的转换，最后给出了转换正确性的证明；第4节主要介绍了等价域的划分方法并给出了模型检测算法；第5节给出一个实例进行验证；最后是结束语。

2 基于连续时间的 ZIA(CT-ZIA)

ZIA 可以同时刻画系统的行为和状态，却没有刻画实时方面的能力；而时间自动机是针对实时系统提出的一种模型。因此，通过将 ZIA 与时间自动机相结合，提出一种针对实时系统的能够同时刻画行为和状态的规范 CT-ZIA。

2.1 CT-ZIA 的定义

定义1 一个基于连续时间的 ZIA(CT-ZIA) P：

$P=\langle S_P, S_P^i, A_P^I, A_P^O, A_P^H, V_P^I, V_P^O, V_P^H, F_P^Y, F_P^A, X, I, T_P\rangle$

其中，(1)S_P 是状态的集合。(2)$S_P^i\subseteq S_P$ 是初始状态的集合。如果 $S_P^i=\varnothing$，P 就为空。(3)A_P^I，A_P^O 和 A_P^H 分别是不相交的输入动作集合、输出动作集合和内部动作集合。我们记所有动作集合 $A_P=A_P^I\cup A_P^O\cup A_P^H$。(4)$V_P^I$，$V_P^O$ 和 V_P^H 分别是不相交的输入变量集合、输出变量集合和内部变量集合。我们记所有变量集合 $V_P=V_P^I\cup V_P^O\cup V_P^H$。(5)$F_P^Y$ 是一个映射，把 S_P 中的任意一个状态映射到用 Z 语言描述的状态模式。(6)F_P^A 是一个映射，把 A_P^I 中的任意一个输入动作映射到用 Z 语言描述的输入操作模式，把 A_P^O 中的任意一个输出动作映射到用 Z 语言描述的输出操作模式，把 A_P^H 中的任意一个内部动作映射到用 Z 语言描述的内部操作模式。(7)X 为时钟变量的非负实数有限集合，$C(X)$ 为 X 上时钟约束的集合，其语法定义如下：$\Phi::=x\prec c\mid c\prec x\mid\phi_1\wedge\phi_2$，其中 $x\in X$，$\prec\in\{<,\leqslant\}$，c 为非负有理数。(8)映射 $I:S_P\rightarrow C(X)$ 为每个状态赋以一时间约束，此约束称为结点不变量。(9)T_P 是状态之间转换关系的集合，$T_P\subseteq S_P\times A_P\times C(X)\times 2^X\times S_P$。如果 $(s,a,\varphi,\lambda,s')\in T_P$，那么表示在满足转换约束条件 φ 的前提下，通过动作 $a\in A_P(s)$，状态 s 可以迁移到新的状态 s'，同时 $\lambda\subseteq X$ 中的时钟被重置为 0。并且，$\models((F_P^Y(s)\wedge F_P^A(a))\backslash(x_1,\cdots,x_m)\Leftrightarrow F_P^Y(t)[y_1'/y_1,\cdots,y_n'/y_n]$，其中 $\{x_1,\cdots,x_m\}$ 是 $F_P^Y(s)$ 上的变量集合，$\{y_1,\cdots,y_n\}$ 是 $F_P^Y(t)$ 上的变量集合，$F_P^A(a)$ 上的变量集合是 $\{x_1,\cdots,x_m\}\cup\{y_1',\cdots,y_n'\}$ 的子集。

2.2 CT-ZIA 对应的时序逻辑 RT-DCL

为了能够对基于连续时间的 ZIA 进行模型检测，有必要给出其对应的时序逻辑。由于目前广泛使用的时序逻辑公式(如 CTL、LTL)没有涉及对数据变量约束的描述，并且只能表达性质“事件 P 最终总会发生”，事件响应时间或者事件出现的次数这样的数量信息都不能够从这些技术之中直接获得。因此下面提出 CT-ZIA 规范对应的时序逻辑 RT-DCL (Real Timed-Data Constraints Logic)，通过在时序操作符上加入下标来约束时间范围[12,13]。

在给出 CT-ZIA 规范对应的逻辑公式之前，有一点需要说明：由于 CT-ZIA 规范可以对数据变量进行约束，因此 CT-ZIA 中刻画的性质不仅有命题逻辑公式，还有一阶逻辑公式和变量，仅使用¬、∧、E、A、U 等算子并不能够对数据进行约束，所以又引入了全称量词∀和存在量词∃，来描述在一定取值范围内变量的一些性质。下面给出 RT-DCL 的语法和语义描述。

2.2.1 语法

本小节将给出 RT-DCL 的语法定义，即 RT-DCL 的逻辑合法公式。

定义2 RT-DCL 公式集合的定义如下：

(1)如果 ϕ 形如 $p(x_1,\cdots,x_n)$，那么 $\phi\in$ RT-DCL。其中 $x_1,\cdots,x_n$ 是变量，p 是一个 n 元谓词；

(2)如果 $\phi_1,\phi_2\in$ RT-DCL，那么 $\phi_1\vee\phi_2\in$ RT-DCL；

(3)如果 $\phi_1,\phi_2\in$ RT-DCL，那么 $\phi_1\wedge\phi_2\in$ RT-DCL；

(4)如果 $\phi_1\in$ RT-DCL，那么 $(\forall x:T)\phi_1\in$ RT-DCL；

(5)如果 $\phi_1\in$ RT-DCL，那么 $(\exists x:T)\phi_1\in$ RT-DCL；

(6)如果 $\phi_1,\phi_2\in$ RT-DCL，那么 $E\phi_1U_{\sim c}\phi_2\in$ RT-DCL，其中 $\sim\in\{<,\leqslant,=,>,\geqslant\}$，$c\in\mathbf{N}$；

(7)如果 $\phi_1,\phi_2\in$ RT-DCL，那么 $A\phi_1U_{\sim c}\phi_2\in$ RT-DCL，其中 $\sim\in\{<,\leqslant,=,>,\geqslant\}$，$c\in\mathbf{N}$。

2.2.2 语义

在给出 RT-DCL 的语义定义之前，先给出 CT-ZIA 上计算路径的定义 。

定义3(计算路径) 一个基于连续时间的 ZIA P 中的计算路径有可能是一个无限的状态序列 $\pi'=(s_0,s_1,\cdots)$，在状态 s_i 上有转移 $(s_i,a,\varphi,\lambda,s_i')\in T_P$，其中每一个 $i\in\mathbf{N}$。对于一条计算路径 $\pi'=(s_0,s_1,\cdots)$，令 $\pi'[k]=s_k$，$\pi_k'=(s_0,\cdots,$

$s_k)$,$k\in N$。用$\Pi'(s)$表示 P 中所有以 s 为起点的无限计算路径的集合。

下面给出 RT-DCL 的语义定义,以判断一个给定公式的真假。

定义 4(RT-DCL 的语义) 假设带数据约束实时系统的 CT-ZIA 模型:

$P=\langle S_P, S_P^s, A_P^I, A_P^Q, A_P^H, V_P^I, V_P^Q, V_P^H, F_P^V, F_P^A, X, I, T_P\rangle$

ϕ 是一个 RC-DCL 公式,$s\in S_P$,满足关系$(P,s)\models\phi$,归纳定义如下:

(1)$(P,s)\models p(x_1,\cdots,x_n)$当且仅当 $F_P^V(s)\rightarrow p(x_1\cdots x_n)$ 对每一个 $s\in S_P$;

(2)$(P,s)\models\phi_1\vee\phi_2$ 当且仅当$(P,s)\models\phi_1$ 或者$(P,s)\models\phi_2$;

(3)$(P,s)\models\phi_1\wedge\phi_2$ 当且仅当$(P,s)\models\phi_1$ 并且$(P,s)\models\phi_2$;

(4)$(P,s)\models(\forall x:T)\phi_1$ 当且仅当$(P,s)\models\bigcap_{v\in T}\phi_1\{v/x\}$;

(5)$(P,s)\models(\exists x:T)\phi_1$ 当且仅当$(P,s)\models\bigcup_{v\in T}\phi_1\{v/x\}$;

(6)$(P,s)\models E\phi_1 U_{\sim_c}\phi_2$ 当且仅当对某些计算路径 $\pi'\in\Pi'(q)$,$t\sim c$,$\pi'[t]\models\phi_2$,并且对每一个 $0<t'<t$,$\pi'[t']\models\phi_1$;

(7)$(P,s)\models A\phi_1 U_{\sim_c}\phi_2$ 当且仅当对每一条计算路径 $\pi'\in\Pi'(q)$,$t\sim c$,$\pi'[t]\models\phi_2$,并且对每一个 $0<t'<t$,$\pi'[t']\models\phi_1$。

3 带数据约束的 MARTE(Z-MARTE)

MARTE 是 UML 在嵌入式实时系统领域的建模规范[10],弥补了 UML 对嵌入式实时领域的非功能属性的表达能力的不足,在工业界的应用非常广泛,但是 MARTE 规范采用图形化的方式描述系统,缺乏精确的语义信息,因此难以直接进行一致性验证。本文在 MARTE 基础上提出 Z-MARTE,用 Z 来形式化描述 MARTE 规范。

3.1 Z-MARTE 介绍

Z-MARTE 是将 MARTE 的核心概念转换为用 Z 语言的模式进行表示。因为 MARTE 是在 UML 上的扩充,所以只要给出用 Z 模式对类图的形式化描述,就可以同样用 Z 把 MARTE 进行形式化描述。根据类图的构成,下面将给出 UML 中类型、关联、聚合、泛化的 Z 模式的形式化描述。

同样地,在 Z 中类型可以分为基本类型和组合类型,这可以对应到 MARTE 中的类型。类似 UML 中的类,类型可以有自己的名字,由一组属性和操作组成。这可以对应到 Z 模式中的 3 个组成部分:名字、说明部分和谓词部分。下面给出一个示例,如图 2 所示。

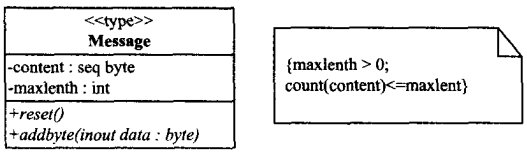

图 2 Message 类型

将一个类型的属性映射到一个 Z 模式,称作属性模式,类型的不变量作为模式的谓词部分。图中的属性如下所示。

Message_Attributes

$content : seq\ byte$

$maxlength : int$

$maxlength > 0$

$\#content \leqslant maxlength$

图 2 中的操作可以转换到 Z 模式,称作操作模式,m 代表操作前的模式,m'代表操作后的模式。

Reset

$m, m' : Message\_Attributes$

$m'.content = m.content\ \emptyset$

$m'.maxlength = m.maxlength$

AddByte

$m, m' : Message\_Attributes$

$data? : byte$

$\#(m.content) < m.maxlength$

$m'.content = m.conteng\langle data?\rangle$

$m'.maxlength = m.maxlength$

最后给出 Message 的类型模式。

Message

$instance : MESSAGE$

$attributes : MESSAGE\ Message\_Attributes$

$reset : MESSAGE\ Reset$

$addbyte: MESSAGE\ AddByte$

$dom\ attributes = instances$

$dom\ reset = instances$

$dom\ addbyte = instances$

$\forall p : instance;\ att1 : Reset;\ att2: AddByte$

$att1 \in reset(p)$

$att2 \in addbyte(p) \cdot att1.m = attributes(p)$

$att2.m = attributes(p)$

下面同样用一个例子给出关联的形式化描述,如图 3 所示。

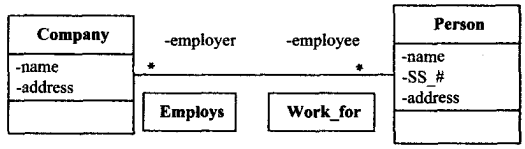

图 3 公司-员工关联图

从 Company 和 Person 两个类的角度来看,这两个类之间有两个关联:Employs,Work_for。

Employs

$employer : Company$

$employee : Person$

$employs : Company \leftrightarrow Person$

$dom\ employs = employer$

$ran\ employs = employee$

$\forall i, j : employer \cdot (i.ident = j.ident \Leftrightarrow i = j)$

$\forall i, j : employee \cdot (i.ident = j.ident \Leftrightarrow i = j)$

Works_for

$company : Company$

$person : Person$

$work\_for : Person \leftrightarrow Company$

$work\_for \in * \leftrightarrow * [person, company]$

$\forall i, j : company \cdot (i.ident = j.ident \Leftrightarrow i = j)$

$\forall i, j : person \cdot (i.ident = j.ident \Leftrightarrow i = j)$

接着给出聚合的形式化描述。聚合是一种特别的关联类型,它指明了相关部件之间的生命期的依赖关系。

图 4 表示一个 Division 实例可以在多个 Company 实例之间共享。

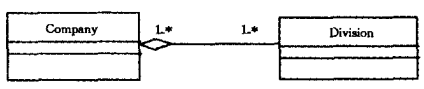

图 4 Company-Division 聚合关系

```
Company
ident : Company_ID
divisions : Division
# divisions ≥ 1

Division
ident : Division_ID

Struct
companies : Company
divisions : Division

∀ d: divisions · (∀ c: companies · d ∈ c.divisions)
∀c1,c2: companies· c1.ident = c2.ident⇔c1 = c2
∀b1,b2: divisions· b1.ident = b2.ident⇔b1 = b2
```

泛化和继承关系是子类和父类之间的一种关系，是用一个带三角形的箭头从子类出发指向父类。用Z模式描述的泛化继承关系包含了在子类型属性模式中定义共享属性。

3.2 Z-MARTE到CT-ZIA的转换

虽然目前MARTE在工业界的应用非常广泛，但是直接对其进行模型检测较为困难。本文通过将其转换为基于连续时间的ZIA模型并提供一个针对连续时间的ZIA的模型检测算法，实现了对MARTE的模型检测。下面将带数据约束的实时系统用Z-MARTE规范刻画为一个六元组 $M=\langle EO, NFP, C, TB, A_M, EX\rangle$，其中

(1) $EO=\{eo_i : i\in \mathbf{N}\}$ 表示系统状态的集合。

(2) NFP 表示系统中对数据性质进行描述的集合。性质是指描述状态的数据属性，可以转换为Z模式进行描述。例如系统执行过程中的总延迟(包括排队延迟)、资源利用率和一个服务的响应时间以及吞吐量等都是系统的 NFP(即非功能属性)。

(3) C 表示系统中时钟变量的非负实数集合，TB 为 C 上的时钟基，时钟基是时钟变量的有序可数集合。定义如下：$TB=\{[c_i,c_j]\mid c_i<c_j\}$，其中 $c_i,c_j\in C$。

(4) A_M 表示系统中的动作集合，包括系统内部动作 A_M^{in}、周围环境中的输入动作 A_M^{I} 和输出动作 A_M^{O}，记作 $A_M=A_M^{in}\cup A_M^{I}\cup A_M^{O}$。

(5) EX 表示系统中转换行为执行的集合，直观意义上来讲就是状态转换的集合，$EX\subseteq EO\times A_M\times 2^C\times TB\times EO$。$(eo,e,\gamma,\delta,eo')\in EX$，表示在满足转换约束条件 $\delta\in[c_i,c_j]$ 的前提下，状态 eo 可以通过动作 $a_i\in A_M$ 迁移到新的状态 eo'，同时 $\gamma\subseteq C$ 中的时钟被重置为0。

从Z-MARTE到基于连续时间ZIA的转化过程如下：

假设 $M=\langle EO, NFP, C, TB, A_M, EX\rangle$ 是由Z-MARTE规范所建模的带数据约束的实时系统，将它转换到基于连续时间的ZIA(CT-ZIA)模型 $P=\langle S_P, S_P^{i}, A_P^{I}, A_P^{O}, A_P^{in}, V_P^{I}, V_P^{O}, V_P^{in}, F_P^{V}, F_P^{A}, X, I, T_P\rangle$，其中：

(1) $S_P=\{s_i : eo_i\in EO, i\in \mathbf{N}\}$，其中 $s_i\in S_P$ 是CT-ZIA模型 P 中的状态，$eo_i\in EO$ 是Z-MARTE模型 M 中的状态。

(2) $A_P=\{a_i : a_j\in A_M, a_j\rightarrow a_i, i,j\in \mathbf{N}\}$，其中 $a_j\in A_M$ 是Z-MARTE模型 M 中的动作，$a_j\rightarrow a_i$ 表示将模型 M 中的每一个动作 a_j 映射为模型CT-ZIA中的一个动作 a_i，根据动作执行者的不同，将它们分别归入内部动作 A_P^{in}、输入动作 A_P^{I} 和输出动作 A_P^{O} 集合中。

(3) $V_P=\{v_i : nfp_i\in NFP, nfp_i\rightarrow v_i, i\in \mathbf{N}\}$，其中 $nfp_i\in NFP$ 是Z-MARTE模型 M 中对数据性质进行描述的非功能属性，$nfp_i\rightarrow v_i$ 表示将每一个非功能性属性 nfp_i 描述中用到的变量 v_i 根据变量的拥有者分别把它们归入到输入变量 V_P^{I}、输出变量 V_P^{O} 和中间变量 V_P^{in} 的集合中。

(4) F_P^{V} 就是将Z-MARTE模型 M 中对数据性质进行描述的每一个非功能属性 $nfp_i\in NFP(i\in \mathbf{N})$ 映射到Z模式的集合。

(5) F_P^{A} 是对于CT-ZIA模型 P 中的动作 $A_P=A_P^{I}\cup A_P^{O}\cup A_P^{in}$，把 A_P^{I} 中的任意一个输入动作映射到用Z语言描述的输入操作模式，把 A_P^{O} 中的任意一个输出动作映射到用Z语言描述的输出操作模式，把 A_P^{in} 中的任意一个内部动作映射到用Z语言描述的内部操作模式。

(6) $X=\{x_i : c_i\in C, i\in \mathbf{N}\}$，$C(X)=\{c_x : [c_i,c_j]\in TB, i,j\in \mathbf{N}, [c_i,c_j]\rightarrow x_i, x_i\in X, i\in \mathbf{N}\}$，其中 $c_i\in C$ 是Z-MARTE模型 M 中的一个时钟变量，映射为CT-ZIA模型 P 中的一个时钟变量 $x_i\in X$，$[c_i,c_j]\in TB$ 是Z-MARTE模型 M 中的一个时钟基，$[c_i,c_j]\rightarrow x_i$ 表示将时钟基 $[c_i,c_j]\in TB$ 映射为CT-ZIA模型 P 中对于时钟 x_i 的时钟约束。

(7) $I=\{s_i\mid\rightarrow[c_i,c_j] : s_i\rightarrow x_i, [c_i,c_j]\rightarrow x_i, s_i\in S, [c_i,c_j]\in TB\}$，其中 $s_i\rightarrow x_i$ 表示将CT-ZIA模型 P 中状态 s_i 上的时钟约束与Z-MARTE模型 M 中时钟基 TB 中的时钟对 $[c_i,c_j]$ 联系起来。

(8) $T_P\subseteq S_P\times A_P\times C(X)\times 2^X\times S_P$，其中 $s_i\in S_P, s_i'\in S_P$ 对应 M 中的状态，$eo_i\in EO, eo_i'\in EO$ 分别表示状态的开始和结束。每一个状态转换 $(s_i\xrightarrow{a_i}s_i')$，其中 $a_i\in A_P$ 对应 M 中的状态转换 $ex_i\in EX$，$ex_i\rightarrow(eo_i\xrightarrow{a_j}eo_i')$，其中 $a_j\in A_M$，时钟约束 $C(X)$ 对应 M 中的一个时钟基 TB。

在系统内部状态变迁过程中，CT-ZIA中的状态转换是根据所要转换系统中的迁移映射过来的，这就保证了在转换过程中系统内部行为的一致性；对于时钟的刻画，把实时系统中的时钟变量提取出来，CT-ZIA中的时钟变量是根据实时系统中的时钟变量以及时间事件来进行时钟不变量的描述，这就确保了实时性的一致性描述；对于数据的约束能力，因为实时系统中非功能属性可以映射到Z语言模型，而CT-ZIA就是把数据用Z模式来进行约束的，这样在数据约束能力方面它们采用的是一样的Z模式，所以从实时系统转换到CT-ZIA之后，它的行为和性质与转换之前保持了一致。

3.3 模型转换正确性证明

3.2节中介绍了Z-MARTE到基于连续时间的ZIA的转换过程，那么当有一个针对连续时间的ZIA进行模型检测的模型检测算法时，也就实现了对带数据约束的实时系统的模型检测。在给出模型转化正确性证明之前，先给出Z-MARTE规范所建模的实时系统中计算路径的定义。

定义5(计算路径) 一个由Z-MARTE规范建模的实时系统中的计算路径有可能是一个无限的状态序列 $\pi=(eo_0, eo_1, \cdots)$，在状态 eo_i 上有转移 $ex_i\in EX$，$ex_i\rightarrow(eo_i\xrightarrow{a_j}eo_i')$，其中每一个 $i\in \mathbf{N}, a_j\in A_M$。对于一条计算路径 $\pi=(eo_0, eo_1, \cdots)$，令 $\pi[k]=eo_k$，$\pi_k=(eo_0,\cdots,eo_k)$，$k\in \mathbf{N}$。用 $\prod(eo)$ 表示 P 中所有以 eo 为起点的无限计算路径的集合。

下面给出RT-DCL在Z-MARTE规范所建模的实时系统 M 中的语义定义，来判断一个给定公式在 M 中的真假。

定义 6 假设 $M=\langle EO, NFP, C, TB, A_M, EX\rangle$ 是由 Z-MARTE规范所建模的实时系统，ϕ 是一个 RC-DCL 公式，$eo\in EO$，满足关系 $(M,eo)\models\phi$ 归纳定义如下：

(1) $(M,eo)\models p(x_1,\cdots,x_n)$ 当且仅当 $F_P^V(nfp_i)\rightarrow p(x_1,\cdots,x_n)$ 对每一个 $nfp_i\in NFP, i\in\mathbf{N}$；

(2) $(M,eo)\models\phi_1\vee\phi_2$ 当且仅当 $(M,eo)\models\phi_1$ 或者 $(M,eo)\models\phi_2$；

(3) $(M,eo)\models\phi_1\wedge\phi_2$ 当且仅当 $(M,eo)\models\phi_1$ 并且 $(M,eo)\models\phi_2$；

(4) $(M,eo)\models(\forall x:T)\phi_1$ 当且仅当 $(M,eo)\models\bigcup\limits_{v\in T}\phi_1\{v/x\}$；

(5) $(M,eo)\models(\exists x:T)\phi_1$ 当且仅当 $(M,eo)\models\bigcup\limits_{v\in T}\phi_1\{v/x\}$；

(6) $(M,eo)\models E\phi_1U_{\sim_c}\phi_2$ 当且仅当对某些计算路径 $\pi\in\Pi(q), t\sim c, \pi[t]\models\phi_2$，并且对每一个 $0<t'<t, \pi[t']\models\phi_1$；

(7) $(M,eo)\models A\phi_1U_{\sim_c}\phi_2$ 当且仅当对每一条计算路径 $\pi\in\Pi(q), t\sim c, \pi[t]\models\phi_2$，并且对每一个 $0<t'<t, \pi[t']\models\phi_1$。

下面介绍从带数据约束的实时系统到基于连续时间的 ZIA 转换的正确性。如果 $M=\langle EO,NFP,C,TB,A_M,EX\rangle$ 是由 Z-MARTE 规范建模的带数据约束的实时系统，$P=\langle S_P, S_P^i, A_P^\flat, A_P^\sharp, A_P^\natural, V_P^\flat, V_P^\sharp, V_P^\natural, F_P^V, F_P^A, X, I, T_P\rangle$ 是由 Z-MARTE 规范模型转换得到的 CT-ZIA 模型，ϕ 是描述系统所要满足性质的 RT-DCL 公式。若要证明从 Z-MARTE M 到 CT-ZIA P 的转换是正确的，则只需要证明 M 满足这个 RT-DCL 公式当且仅当 P 也满足这个 RT-DCL 公式。因此有下面的引理。

引理 1 令 M 是由 Z-MARTE 规范所建模的带数据约束的实时系统，P 是由 M 转换得到的 CT-ZIA 模型，ϕ 是描述系统性质的 RT-DCL 公式，有 $(M,eo)\models\phi$ 当且仅当 $(P,s)\models\phi$。

证明：从 3.2 节的转换过程我们很容易看出 M 中每一个状态 eo_i 与 P 中每一个状态 s_i 是一一对应的，其中 $i\in\mathbf{N}$；M 中每一个状态转换 $ex_i\in EX, ex_i\rightarrow(eo_i\xrightarrow{a_i}eo_i')$，对应 P 中的一个状态转换 $(s_i\xrightarrow{a_i}s_i')$，其中 $a_i\in A_P$；M 中每一个时钟 $c_i\in C$ 映射为 P 中的时钟 x_i，M 中时钟基中的每一个有序对映射为 P 中的一个时钟约束。利用数学归纳法对 ϕ 结构复杂度进行归纳证明，证明过程如下：

假设 $(M,eo)\models\phi$ 当且仅当 $(P,s)\models\phi$。

(1) 若 ϕ 形如 $p(x_1,\cdots,x_n)$，其中 $x_1,\cdots,x_n$ 是变量，p 是一个 n 元谓词，$(M,eo)\models\phi$，则对每一个 $nfp_i\in NFP$ 有 $F_P^V(nfp_i)\rightarrow p(v_1,\cdots,v_n)$，因为 M 中每一个 $nfp_i\in NFP$ 描述中用到的变量 v_i 根据变量的拥有者分别把它们归到 P 中输入、输出和中间变量的集合中，所以 $F_P^V(s)\rightarrow p(x_1,\cdots,x_n)$，即 $(P,s)\models\phi$；

(2) 若 $\phi=\phi_1\vee\phi_2$，$(M,eo)\models\phi$ 即 $(M,eo)\models\phi_1\vee\phi_2$，则 $(M,eo)\models\phi_1$ 或者 $(M,eo)\models\phi_2$，表示 ϕ_1 和 ϕ_2 在 Z-MARTE 模型 M 中的状态 eo 上有且仅有一个为真，根据归纳假设可知 ϕ_1 和 ϕ_2 在 CT-ZIA 模型 P 中对应的状态 s 上有且仅有一个为真即 $(P,s)\models\phi_1$ 或者 $(P,s)\models\phi_2$，所以 $(P,s)\models\phi_1\vee\phi_2$ 即 $(P,s)\models\phi$；

(3) 若 $\phi=\phi_1\wedge\phi_2$，$(M,eo)\models\phi$ 即 $(M,eo)\models\phi_1\wedge\phi_2$，则 $(M,eo)\models\phi_1$ 并且 $(M,eo)\models\phi_2$，表示 ϕ_1 和 ϕ_2 在 Z-MARTE 模型 M 中的状态 eo 上同时为真，根据归纳假设可知 ϕ_1 和 ϕ_2 在 CT-ZIA 模型 P 中对应的状态 s 上同时为真即 $(P,s)\models\phi_1$ 并且 $(P,s)\models\phi_2$，所以 $(P,s)\models\phi_1\wedge\phi_2$ 即 $(P,s)\models\phi$；

(4) 若 $\phi=(\forall x:T)\phi_1$，$(M,eo)\models\phi$，即 $(M,eo)\models(\forall x:T)\phi_1$，则 $(M,eo)\models\bigcap\limits_{v\in T}\phi_1\{v/x\}$，对每一个 $v\in T$，$(M,eo)\models\phi_1\{v/x\}$ 表示对每一个 $v\in T$，$\phi_1\{v/x\}$（表示将公式 ϕ_1 中的 x 用 v 来替换）在 Z-MARTE 模型 M 中的状态 eo 上为真，根据归纳假设可知对每一个 $v\in T$，$\phi_1\{v/x\}$（表示将公式 ϕ_1 中的 x 用 v 来替换）在 CT-ZIA 模型 P 中的状态 s 上为真，即对每一个 $v\in T$，$(P,s)\models\phi_1\{v/x\}$，则 $(P,s)\models\bigcap\limits_{v\in T}\phi_1\{v/x\}$，所以 $(P,s)\models(\forall x:T)\phi_1$，即 $(P,s)\models\phi$；

(5) 若 $\phi=(\exists x:T)\phi_1$，$(M,eo)\models\phi$，即 $(M,eo)\models(\exists x:T)\phi_1$，则 $(M,eo)\models\bigcup\limits_{v\in T}\phi_1\{v/x\}$，对每一个 $v\in T$，$(M,eo)\models\phi_1\{v/x\}$ 表示有一个 $v\in T$，$\phi_1\{v/x\}$（表示将公式 ϕ_1 中的 x 用 v 来替换）在 Z-MARTE 模型 M 中的状态 eo 上为真，根据归纳假设可知有一个 $v\in T$，$\phi_1\{v/x\}$（表示将公式 ϕ_1 中的 x 用 v 来替换）在 CT-ZIA 模型 P 中的状态 s 上为真，即有一个 $v\in T$，$(P,s)\models\phi_1\{v/x\}$，则 $(P,s)\models\bigcup\limits_{v\in T}\phi_1\{v/x\}$，所以 $(P,s)\models(\exists x:T)\phi_1$，即 $(P,s)\models\phi$；

(6) 若 $\phi=E\phi_1U_{\sim_c}\phi_2$，$(M,eo)\models\phi$，即 $(M,eo)\models E\phi_1U_{\sim_c}\phi_2$，则存在某些计算路径 $\pi\in\Pi(q)$，$\pi=(eo_0,eo_1,\cdots)$，$t\sim c, \pi[t]\models\phi_2$（$\phi_2$ 在 M 中状态 eo_t 上为真），并且对每一个 $0<t'<t, \pi[t']\models\phi_1$（$\phi_1$ 在 M 中状态 $eo_{t'}$ 上为真），根据归纳假设，P 中存在对应的某些计算路径 $\pi'\in\Pi'(q)$，$\pi'=(s_0,s_1,\cdots)$，$t\sim c, \pi'[t]\models\phi_2$（$\phi_2$ 在 P 中状态 s_t 上为真），并且对每一个 $0<t'<t$，$\pi'[t']\models\phi_1$（ϕ_1 在 P 中状态 $s_{t'}$ 上为真），所以 $(P,s)\models E\phi_1U_{\sim_c}\phi_2$，即 $(P,s)\models\phi$；

(7) 若 $\phi=A\phi_1U_{\sim_c}\phi_2$，$(M,eo)\models\phi$，即 $(M,eo)\models A\phi_1U_{\sim_c}\phi_2$，则对每一条计算路径 $\pi\in\Pi(q)$，$\pi=(eo_0,eo_1,\cdots)$，$t\sim c, \pi[t]\models\phi_2$（$\phi_2$ 在 M 中状态 eo_t 上为真），并且对每一个 $0<t'<t, \pi[t']\models\phi_1$（$\phi_1$ 在 M 中状态 $eo_{t'}$ 上为真），根据归纳假设，P 中存在对应的每一计算路径 $\pi'\in\Pi'(q)$，$\pi'=(s_0,s_1,\cdots)$，$t\sim c, \pi'[t]\models\phi_2$（$\phi_2$ 在 P 中状态 s_t 上为真），并且对每一个 $0<t'<t$，$\pi'[t']\models\phi_1$（ϕ_1 在 P 中状态 $s_{t'}$ 上为真），所以 $(P,s)\models A\phi_1U_{\sim_c}\phi_2$，即 $(P,s)\models\phi$。

4 模型检测

在 2.1 节 CT-ZIA 的定义中，定义时钟约束（用于结点的不变量和转换约束条件）包含任意有理常数，将每一个时钟约束中的常量与出现在所有时钟约束中的常量的分母的最小公倍数 m 相乘[14]，这个转换将所有的常量转变为整数，这样所遇到的时钟约束中的所有常量都是整数值了，而时钟的值依然可以是任意非负实数值。

对于模型检测 CT-ZIA P 来说，需要得到等价的转换系统 $S(P)$。让 $S(P)$ 的初始状态 $s_0=(s_i,v_0)$，$s_i\in S_P^i$ 为 P 的初始结点，v_0 为初始时钟，其中 $v(x)$ 表示对时钟 $x\in X$ 取当前值。这也就意味着可以形式化 CT-ZIA 去满足 RT-DCL 公式 ϕ。

定义 7 已知 P 为一个 CT-ZIA 模型，S 是模型 P 中的状态，ϕ 为对应的时序逻辑 RT-DCL 公式，w 为模型 P 上的时钟，则 $(P,s)|=\phi$ 当且仅当 $(S(P),(s_0,w_0))|=\phi$，对所有时钟 y 有 $w_0(y)=0$。

通过这样的等价，实际上对 CT-ZIA P 的模型检测可以转换为对 $S(P)$ 变迁系统的模型检测。但存在一个主要问题：对于变迁系统 $S(P)$ 而言，状态空间 $S\times V(X)$ 集合是无限的，因为 $V(X)$ 是一个无限集合，表示对时钟集合 X 中所有时钟取所有值的集合。

4.1 等价域构造方法

由于对于变迁系统 $S(P)$ 而言，状态空间 $S\times V(X)$ 集合是无限的，而现在的模型检测技术只能处理有穷状态系统，因此必须把无穷状态转化为有穷状态[15]。为了获得 $S(P)$ 的无穷状态空间的有穷表示，Alur，Courcoubetis 和 Dill 在文献[16,17]中提出一种时钟等价方法，把时间自动机等价为域自动机，但是按照 Alur 时钟等价方法构造出的域自动机，存在状态空间迅速膨胀爆炸的问题。本文采用一种优化的时钟等价方法，并且为了满足优化的时钟等价方法，修改了域自动机定义，使等价后的域自动机状态数尽量少[18]。这样我们也可以将 CT-ZIA 等价为一种带 Z 的域自动机，从而使 CT-ZIA 的模型检测问题成为可能。

定义 8 设 c 为时钟的约束常量值。

对时钟 x 而言，c_x 表示时钟 x 的约束常量值。若从 s 结点到 s' 结点的边上有一时钟约束 $x\geqslant 2$，即 $v(x)\geqslant c_x$ 时，变迁转换允许发生，则 $c_x=2$。

定义 9 域 r 是状态结点和时钟区域的结合，可以描述为一个二元对 (s,v_p)，其中，$s\in S$，v_p 为时钟关键点，且 $v_p\in V(X)$。$s_{\approx}$ 是 $s=(s,v)$ 相对应于 (s,v_p) 的缩写。

定义 10 设 r 是一个无界域，表示对所有的时钟 $v(x)\geqslant c_x$，$x\in X$。

定义 11 设 r 的后继域为 $succ(r)$，r 是一个无界域，则 $r=succ(r)$；r 和 r' 为两个不同的域，即 $r\neq r'$，$r=succ(r)$ 表示为存在正实数 $d_p\in\mathbb{R}^+$，即 $d_p:v_{p_2}-v_{p_1},v_{p_2}>v_{p_1}$ 使 $r=s_{\approx}$，$s\xrightarrow{d_p}s'$，$r=(s+d_p)_{\approx}$ 成立。

定义 12 域自动机 $R(P,\phi)$ 是一个转换系统 $(R,r,\rightarrow)$，定义如下：

$R=S_{/\approx}=\{s_{\approx}\mid s\in S\}$；

$r_0=s_{0\approx}$；

$r\rightarrow r'$ 当且仅当 $\exists s,s'(r=s_{\approx}\wedge r'=s_{\approx}{}'s\xrightarrow{*}s')$；

$r\rightarrow r'$ 当且仅当 $r'=succ(r)$。

对状态 s 来说，约束条件 $x\geqslant 2$ 中的约束常量 2 是一个关键点，因为当 $v(x)>2$ 或 $v(x)<2$ 时，无论取何值都是不相关的，所以仅需要考虑约束常量这个关键点就可以了。综上所述，对时钟 x 而言，可分为 3 种情况：

(1) $v(x)<c_x$ 则 $v'(x)<c_x$；

(2) $v(x)=c_x$ 则 $v'(x)=c_x$；

(3) $v(x)>c_x$ 则 $v'(x)>c_x$。

考虑时钟存在复位操作，则 0 也是一个特定的关键点。如果一个时钟有 n 个不同约束常量，那么就存在 $n+1$ 个时钟关键点。假定时钟关键点集合 C_x 有 $c_{x_0}<c_{x_1}<\cdots<c_{x_k}$，且 $c_{x_0}=0,c_{x_i}\in C_x$，其中 $i=0,1,\cdots,k$，则优化时钟等价方法(1)如下：

(1) $c_{x_{i-1}}<v(x)<c_{x_i}$，则 $c_{x_{i-1}}<v'(x)<c_{x_i}$；

(2) $v(x)>c_{x_k}$，则 $v'(x)>c_{x_k}$；

(3) $v(x)=c_{x_k}$，则 $v'(x)=c_{x_k}$。

通过以上的等价，时钟就变为有限集合。

如果存在两个时钟 x,y，以及相应时钟关键点集合 C_x 和 C_y，则可以用一个二元对 (c_x,c_y) 来描述时钟关键点，且 $(c_x,c_y)\in C_x\times C_y$。然而时钟等价不仅需要考虑每个时钟的取值范围，而且还需要考虑 $y-x$ 的取值范围。为了确定 $y-x$ 的取值范围，首先要确定 $y-x$ 的分界线，根据时钟关键点来定义分界线 $y-x=h$，其中 $h=c_y-c_x,c_y\in C_y$ 且 $0<y<c_y,c_x\in C_x$，且 $0<x<c_x$。假定两个时钟关键点集合 C_x 和 $C_y,c_{x_i}\in C_x,c_{y_i}\in C_y$，并满足 $c_{x_0}<c_{x_1}<\cdots<c_{x_k},c_{y_0}<c_{y_1}<\cdots<c_{y_l}$，且 $h_1<h_2<h_m,m=k\times l$。那么时钟等价方法(2)如下：

(1)如果 $h_{s-1}<v(y)-v(x)<h_s$，且 $c_{x_{i-1}}<v(x)<c_{x_i}$，$c_{y_{j-1}}<v(y)<c_{y_j}$，则 $h_{s-1}<v'(y)-v'(x)<h_s$，且 $c_{x_{i-1}}<v'(x)<c_{x_i},c_{y_{j-1}}<v'(y)<c_{y_j}$，其中 $c_{y_{j-1}}-c_{x_i}<h_{s-1},h_s<c_{y_j}-c_{x_{i-1}}$，$i=1,\cdots k,j=1,\cdots,l,s=2,\cdots,m$。

(2)如果 $v(x)-v(y)=h_s$，且 $c_{x_{i-1}}<v(x)<c_{x_i},c_{y_{j-1}}<v(y)<c_{y_j}$，则 $v'(x)-v'(y)=h_s$ 且 $c_{x_{i-1}}<v(x)<c_{x_i},c_{y_{j-1}}<v'(y)<c_{y_j}$，其中 $c_{y_{j-1}}-c_{x_i}<h_s<c_{y_j}-c_{x_{i-1}}$，$i=1,\cdots,k,j=1,\cdots,l,s=1,\cdots,m$。

当考虑两个时钟时，还可以对它进行一些限制，因为仅当 $v(x)<c_{x_{\max}},v(y)<c_{y_{\max}}$ 时，才需考虑两个时钟之间的大小关系，否则就无需考虑两个时钟之间的大小关系。

4.2 等价域自动机构造

如图 5 所示，CT-ZIA 有一个时钟集合 $X=\{x\}$ 且 $c_x=3$，它仅有一个时钟 x 和时间约束 $x\geqslant 3$。根据上节所给出的等价方法(1)，就可以得到 CT-ZIA 的等价域：$[x=0]$，$[0<x<3]$，$[x=3]$，$[x>3]$，从而可以得出一个域自动机，如图 6 所示。图中所示域自动机有两种类型迁移，即实线和虚线。实线表示时间增加，虚线表示等价的 CT-ZIA 状态迁移。此外图中的灰色框表示为无界域(表示对所有的时钟 $v(x)>c_x$，$x\in Xv(x)$)。

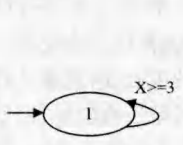

图 5 CT-ZIA

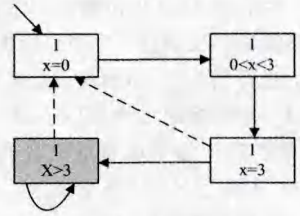

图 6 域自动机

由于时钟属于非负实数集 $\mathbb{R}^+$，故 n 个时钟将构造成 n 维 $\mathbb{R}^+$ 空间。对于图 7，CT-ZIA 的时钟集合 $X=\{x,y\}$，并且时钟 x 有两个时钟约束常量 2 和 3，则时钟 x 关键点集合 $C_x=\{0,2,3\}$；时钟 y 有一个时钟约束常量 4，则时钟 y 的关键点集 $C_y=\{0,4\}$。这时等价时间域将是二维空间 $\mathbb{R}^+\times\mathbb{R}^+$。根据方法(1)，将得到如下的等价时间域：

$[(x=0,y=0)][(x=0,0<y<4)];[(x=0,y=4)][(x=0,y>4)]$；

$[(0<x<2,y=0)][(0<x<2,0<y<4)];[(0<x<2,y=4)][(0<x<2,y>4)];$

$[(x=2,y=0)][(x=2,0<y<4)];[(x=2,y=4)][(x=2,y>4)];$

$[(2<x<3,y=0)][(2<x<3,0<y<4)];[(2<x<3,y=4)][(2<x<3,y>4)];$

$[(x=3,y=0)][(x=3,0<y<4)];[(x=3,y=4)][(x=3,y>4)];$

$[(x>3,y=0)][(x>3,0<y<4)];[(x>3,y=4)][(x>3,y>4)];$

根据方法(2),首先来确定 $y-x$ 的分界线,根据关键点(2,4)和(3,4)可确定两条分界线 $y-x=4-2=2$, $0<x<2,0<y<4$和 $y-x=4-3=1,0<x<3,0<y<4$,并可得 $h_1=1,h_2=2$。通过分界线可把区域$[(0<x<2,0<y<4)]$和$[(2<x<3,0<y<4)]$再进行细分,考虑$[(0<x<2,0<y<4)]$,如果需要分界线在此区域起到作用,分界线 $y-x=h$ 需要满足以下条件:$-2<h<4,0<x<2,0<y<4$,故分界线$y-x=1$和 $y-x=2$ 满足条件。区域$[(0<x<2,0<y<4)]$再细分为$[(0<x<2,0<y<4),-2<y-x<1]$,$[(0<x<2,0<y<4),y-x=1]$,$[(0<x<2,0<y<4),1<y-x<2]$,$[(0<x<2,0<y<4),y-x=2]$,$[(0<x<2,0<y<4),2<y-x<4]$。类似地,对于$[(2<x<3,0<y<4)]$也做同样处理。图 7 所示的 CT-ZIA 通过优化等价规则后得到等价时间域,如图 8 所示。

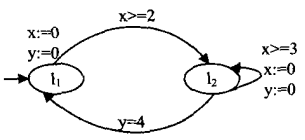

图 7 CT-ZIA

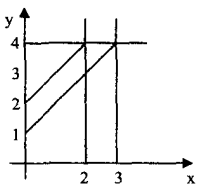

图 8 时间域

4.3 模型检测算法

下面给出基于连续时间的 ZIA 的模型检测算法,当我们有了一个针对基于连续时间的 ZIA 的模型检测算法时,也就有了一个针对 Z-MARTE 的模型检测算法,从而实现了对带数据约束实时系统的模型检测。

Rajeev Alur 在文献[19]中提出了一种标记算法并证明了该算法的正确性。本节介绍的模型检测算法是在此算法基础上进行的改进,即加入了数据变量的约束方面,基本思路是通过下面的标记过程,用 ϕ 的子公式来标记所有的域,根据前文的定义 7,如果域自动机的初始状态最后用 ϕ 标记,那么 CT-ZIA 满足 RT-DCL 公式 ϕ,下面介绍模型检测算法。

模型检测算法:初始时,用如下的专门命题来标记域。每一个顶点用公式 true 来标记。对于出现在 ϕ 中的每一个下标$\sim c$,假设有一新命题 $p_{\sim c}$,如果 $v|=x_{\sim c}$,则用 $p_{\sim c}$ 标记顶点。此外,假设 p_b 是一新命题公式,它在顶点 r 为真当且仅当 r 不是一无界域。

假设 Ψ 是 ϕ 的一个子公式。假设顶点已经用 Ψ 的每一个子公式标记。设 r 是任何一个域。

(1)若 Ψ 是一个原子命题 $p(x_1,\cdots,x_n)$,如果 $F_P^V(s)\rightarrow p(x_1,\cdots,x_n)$,则用 Ψ 标记 r。

(2)若 $\Psi=\phi_1\vee\phi_2$,如果 r 用 ϕ_1 或者 ϕ_2 来标记,那么用 Ψ 标记它。

(3)若 $\Psi=\phi_1\wedge\phi_2$,如果 r 用 ϕ_1 和 ϕ_2 来标记,那么用 Ψ 标记它。

(4)若 $\Psi=(\forall x:T)\phi$,如果 r 用$(\forall u\in T)\phi\{u/x\}$来标记,那么就用 Ψ 标记它。

(5)若 $\Psi=(\exists x:T)\phi$,如果 r 用$(\exists u\in T)\phi\{u/x\}$来标记,那么就用 Ψ 标记它。

(6)假设 Ψ 是时序公式 $Q(\phi_1U_{\sim c}\phi_2)$,其中 Q 或者是存在量词或者是全称量词,如果以 r 开始的通过 $R(P,\phi)$ 的某一路径,有一前缀 $r_1,r_2,\cdots,r_n,1\leqslant i<n,r_i$ 用 ϕ_1 标记,且 r_n 用 ϕ_2 标记,$p_{\sim c}$ 用 p_b 或 ϕ_1 标记,那么用 Ψ 标记 r。

函数 Sub 的功能是,当给定一个公式 ϕ,它将返回一个 ϕ 的子公式队列,其中如果 ϕ_1 是 ϕ 的子公式,ϕ_2 是 ϕ_1 的子公式,那么在队列 $Sub(\phi)$ 中 ϕ_2 出现在 ϕ_1 的前面。给出模型检测算法如下所示:

```
Procedure set Check(P,φ)
φ=p_{~c} if v|=x_{~c}then φ∈label(r);
{for each Ψ in Sub(φ) do;
case Ψ=p(x1,⋯,xn);r=(s,v)
  if s∈{s|F_P^V(s)→p(x1,⋯,xn)} then
  label(r)=label(r)∪Ψ;
//其中 Z 模式 F_P^V(s)看作是一阶逻辑公式
  break;
case Ψ=φ1∨φ2:
  if φ1∈label(r) then label(r)=label(r)∪Ψ
  else if φ2∈label(r) then label(r)=label(r)∪Ψ;
  break;
case Ψ=φ1∧φ2:
  if φ1∈label(r){
    if φ2∈label(r) then label(r)=label(r)∪Ψ;}
    break;
case Ψ=(∃x:T)φ;r=(s,v);
  if s∈ ⋃_{u∈T} {s|φ({u/x})∈label(r) then label(r)=label(r)∪Ψ};
  //其中 x 的类型是 T
  break;
case Ψ=(∀x:T)φ;r=(s,v);
  if s∈ ⋂_{u∈T} {s|φ({u/x})∈label(r) then label(r)=label(r)∪Ψ};
  //其中 x 的类型是 T
  break;
case Ψ=E(φ1U_{~c}φ2) T:={s|φ2∈label(r)}
  for one s∈T do label(r)=label(r)∪Ψ;
    while T≠∅ do
      choose s∈T;T=T\{s};
      for all t such that R(t,s) do
  if Ψ∉label(ri) and φ1∈label(ri) and (pb∈label(φ) or φ1∈label
    (φ)) then label(ri)=label(ri)∪Ψ;
      T=T∪{t};
      break;
case Ψ=A(φ1U_{~c}φ2) T:={s|φ2∈label(r)}
  for all s∈T do label(r)=label(r)∪Ψ;
    while T≠∅ do
      choose s∈T;T=T\{s};
      for all t such that R(t,s) do
  if Ψ∉label(ri) and φ1∈label(ri) and (pb∈label(φ) or φ1∈label
    (φ)) then label(ri)=label(ri)∪Ψ;
```

T=T∪{t};
break;
}//end

Rajeev Alur 在文献[19]中对标记算法的正确性给出了证明，本文证明方法与此类似，下面的引理表明了上述标记过程的正确性。

引理 2(改进的标记算法正确性) 令 Ψ 是一个公式，如果上面介绍的标记算法用 Ψ 来标记域 $r=(s,v)$ 当且仅当 $(S(P),(s,v))|=\Psi$。

证明：下面利用数学归纳法对 Ψ 的结构进行归纳证明，证明过程如下：

假设 r 用 Ψ 标记当且仅当 $(S(P),(s,v))|=\Psi$

(1)若 Ψ 是一个原子命题 $p(x_1,\cdots,x_n)$，如果 r 用 Ψ 来标记，则 $F_p^{\forall}(s)\rightarrow p(x_1,\cdots,x_n)$，所以 $(S(P),(s,v))|=\Psi$；

(2)若 $\Psi=\phi_1\vee\phi_2$，如果 r 用 Ψ 来标记，则 r 用 ϕ_1 或者 ϕ_2 来标记，根据归纳假设可知 $(S(P),(s,v))|=\phi_1$ 或者 $(S(P),(s,v))|=\phi_2$，所以 $(S(P),(s,v))|=\phi_1\vee\phi_2$，即 $(S(P),(s,v))|=\Psi$；

(3)若 $\Psi=\phi_1\wedge\phi_2$，如果 r 用 Ψ 来标记，则 r 用 ϕ_1 和 ϕ_2 来标记，根据归纳假设可知 $(S(P),(s,v))|=\phi_1$ 并且 $(S(P),(s,v))|=\phi_2$，所以 $(S(P),(s,v))|=\phi_1\wedge\phi_2$，即 $(S(P),(s,v))|=\Psi$；

(4)若 $\Psi=(\forall x:T)\phi_1$，如果 r 用 Ψ 来标记，则 r 用 $\phi_1\{u/x\}$ 来标记对每一个 $u\in T$，根据归纳假设可知 $(S(P),(s,v))|=\phi_1\{u/x\}$ 对每一个 $u\in T$，即 $(S(P),(s,v))|=\Psi$；

(5)若 $\Psi=(\exists x:T)\phi_1$，如果 r 用 Ψ 来标记，则 r 用 $\phi_1\{u/x\}$ 来标记对存在一个 $u\in T$，根据归纳假设可知 $(S(P),(s,v))|=\phi_1\{u/x\}$ 对存在一个 $u\in T$，即 $(S(P),(s,v))|=\Psi$；

(6)若 Ψ 是时序公式 $Q(\phi_1U_{\sim c}\phi_2)$，其中 Q 或者是存在量词或者是全称量词，如果 r 用 Ψ 来标记，则以 r 开始的通过 $R(P,\phi)$ 的某一路径，有一前缀 $r_1,r_2,\cdots,r_n,1\leqslant i<n,r_i$ 用 ϕ_1 标记，且 r_n 用 ϕ_2 标记，$p_{\sim c}$ 用 p_b 或 ϕ_1 标记，根据归纳假设可知，以 r 开始的通过 $R(P,\phi)$ 的某一路径，有一前缀 $r_1,r_2,\cdots,r_n,1\leqslant i<n,(S(P),r_i)|=\phi_1$ 并且 $(S(P),r_n)|=\phi_2$，并且 $p_{\sim c}|=p_b$ 或者 $p_{\sim c}|=\phi_1$，所以 $(S(P),(s,v))|=Q(\phi_1U_{\sim c}\phi_2)$ 即 $(S(P),(s,v))|=\Psi$。

这表明了模型检测的判定过程：给定一个 CT-ZIA 和一个 RT-DCL 公式 ϕ，首先构造一个区域图，然后根据上述标记过程用 ϕ 的子公式来标记所有的域，根据上述引理，$S(P)|=\phi$ 当且仅当 (s_0,v_0) 用 ϕ 标记。

4.4 模型检测算法在现有工具基础上的实现问题

现有常见模型检测工具大多针对离散时间，也有少部分工具针对连续时间，但不能直接用这些工具来实现本文算法，原因如下：

1. 本文所使用的模型 CT-ZIA 是带有数据变量约束的，现有常见工具(如 UPPAAL)所基于的模型(如时间自动机)没有数据约束；

2. 本文的模型中包含的变量既包括连续变量也包括离散变量，现有常见的工具所基于的模型中通常只包含有离散变量。

如果要在现有工具的基础上实现本文的算法，需要把本文所用的模型转换为现有工具所基于的模型。但是将本文模型转换为现有工具所基于的模型存在以下两个方面的问题：

1. 将本文的模型转换为现有工具所基于的模型会导致转换的模型比较大，例如，本文 4.2 节图 6 中的一个状态 $0<x<3$ 就变成 $x=1,x=2$ 两个状态，状态 $x>3$ 就变成 $x=4,x=5,\cdots$ 等多个状态，这样会导致转换的模型(如图 9 右)状态非常多，使得模型检测复杂度高；

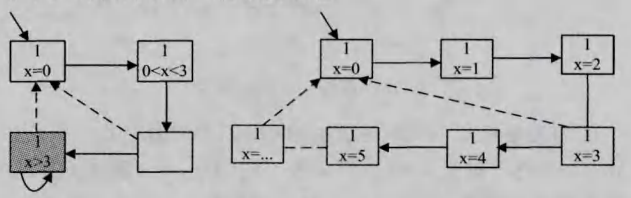

图 9 域自动机

2. 因为本文的逻辑公式 RT-DCL 与现有工具所基于的模型对应的时序逻辑公式(如 TCTL)不同，所以也要将模型对应的时序逻辑公式进行转换，同理会导致逻辑公式长度变大，模型检测复杂度高。

5 实例应用

本节以一个机场飞机着陆控制的例子来说明模型检测带数据约束的实时系统的具体过程，以验证方法可行有效。

5.1 问题描述

图 10 描述了一架飞机、机场跑道和机场塔台，机场跑道是指飞机场内用来供应航空飞行器起飞或降落的超长条形区域；机场塔台或称控制塔，是一种设置于机场中的航空运输管制设施，用来监看以及控制飞机起降的地方。

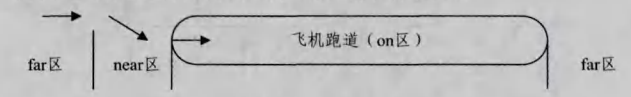

图 10 飞机跑道

飞机从 far 区到达 near 区接近机场时，控制塔发出 App 信号，跑道上的信号灯接收到信号后从绿灯状态变为红灯状态，表示跑道忙碌。飞机从 near 区到达 on 区，飞机 near 区到达 on 区需要 1～2 分钟，信号灯从绿灯状态变为红灯状态不超过 1 分钟。当飞机离开跑道进入停机坪时，塔台发出 Exit 信号，跑道信号灯接受到信号后从红灯状态变为绿灯状态，表示跑道空闲。飞机在跑道(on 区)上滑行的时间需要 1～2 分钟，跑道信号灯从红灯状态变为绿灯状态不超过 1 分钟。我们定义距离机场超过 20km 为 far 区，小于等于 20km 为 near 区，假设飞机在 far 区的速度不大于 700km/h，在 near 区的速度不大于 400km/h，在 on 区滑行速度不大于 200km/h。

5.2 建立模型

分别用 f,n,o,fr,s,b,r 表示状态 far,near,on,free,send,busy,receive。飞机和跑道的自动机模型如图 11、图 12 所示。

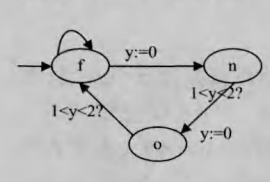

图 11 飞机自动机模型

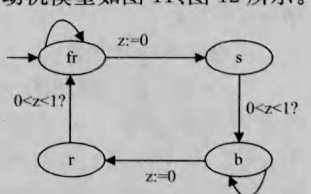

图 12 跑道自动机模型

假设系统初始状态时飞机在 far 状态且跑道在 free 状态，根据自动机的状态转换关系，得到整个系统的 CT-ZIA 的

转换关系，在此过程中去除一些无效状态。整个系统的 CT-ZIA 上的时间约束是自动机中相应状态转换关系上的时间约束的合取。根据图 11、图 12 得到图 13 的整个飞机降落机场的 CT-ZIA 模型。

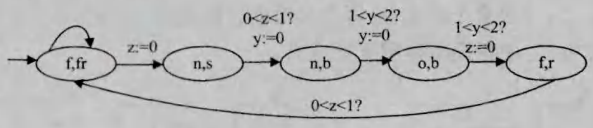

图 13　系统 CT-ZIA 模型

因为在整个飞机降落机场的 CT-ZIA 模型中 $c_y=2$，所以图 14 中的 y 轴标 1 和 2 两个值。同时 $c_z=1$，而在 CT-ZIA 转换到区域图的过程中，引入了一个额外的时钟 x，所以用 x 轴表示 z 的取值，在图 14 中 x 轴标 1。

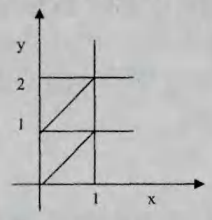

图 14　时间域

根据图 14 中的划分域的坐标，因为时钟的初始值都设置为 0，所以从原点出发，再根据求取等价域和构造域自动机的方法，得到如图 15 所示的整个系统的域自动机模型。

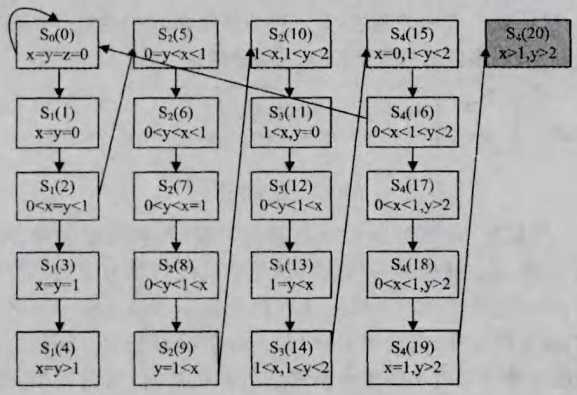

图 15　域自动机模型

5.3　性质验证

先给出飞机模型各个状态的 Z 模式：

$F_P^V(f)=S_f\triangleq[distance?:N;speed!:N\mid distance?>20;400<speed!\leqslant 700]$

$F_P^V(n)=S_n\triangleq[distance?:N;speed!:N\mid distance?\leqslant 20;200<speed!\leqslant 400]$

$F_P^V(o)=S_o\triangleq[distance?:N;speed!:N\mid distance?=0;0<speed!\leqslant 200]$

来验证下面的性质：

(1)无论飞机何时到达 on 区，跑道都是处于 busy 状态，跑道空闲状态用 free 来表示。用 RT-DCL 公式表示为 $A(trueU(on\wedge free))$。考虑 $A(trueU(on\wedge free))$，因为在 S2 和 S3 是用 busy 标记的，故我们用 $A(trueU(on\wedge free))$ 标记所有顶点，根据算法最后用 $A(trueU(on\wedge free))$ 标记顶点 0，所以性质成立。

(2)跑道处于 busy 状态不超过 10 分钟。用 RT-DCL 公式表示为 $E(busyU_{\leqslant 10}free)$。考虑 $E(busyU_{\leqslant 10}free)$，顶点 0，1,3,5,7,9,11,13,15,17,19 用 P_b 标记。0,1,2,5,6,7,8,9,10,11,12,13,14,15,16,0,…是一条无空 Fp-公平路径。busy 和 $P_{\leqslant 10}$，P_b，P_b 在顶点 5 成立，free 在顶点 0,1,2 成立，根据算法用 $E(busyU_{\leqslant 10}free)$ 标记顶点 0，所以性质成立。

(3)当飞机与机场距离小于 20km 时，飞机速度就变为介于 200km/h 与 400km/h 之间 。用 RT-DCL 公式表示为 $(distance?\leqslant 0\vee distance?>20)\vee(speed!>200\wedge speed!\leqslant 400)$成立，根据算法，状态 n 对应的 Z 模式有以上 RT-DCL 公式成立，所以性质成立。

结束语　本文为了对带有数据约束的实时系统进行研究，将带有数据约束的实时系统用带有数据约束的 MARTE 来模拟并提出了一种基于连续时间的 ZIA 规范，其既可以描述系统的行为属性，也可以刻画系统数据方面的属性，通过将带数据约束的 MARTE 转化为基于连续时间的 ZIA，并给出基于连续时间 ZIA 的模型检测算法，实现了对带数据约束实时系统的模型检测。本文在原理和方法上对带数据约束的实时系统进行了研究，下一步的研究工作是将这些原理和方法运用到具体的实际实时系统中，这是未来研究工作的方向和重点。

参 考 文 献

[1] 李广元，唐稚松. 带有时钟变量的线性时序逻辑与实时系统验证[J]. 软件学报，2002，13(1)：33-41

[2] 戎玫，张广泉. 模型检测新技术研究[J]. 计算机科学，2003，30(5)：102-104

[3] Bérard B，Bidoit M，Finkel A，et al. Systems and software verification：model-checking techniques and tools[M]. Springer Publishing Company，Incorporated，2010

[4] 王晶，戎玫，张广泉，等. 基于概率模型检测的 Web 服务组合验证[J]. 计算机科学，2012，39(1)：120-123

[5] Cao Z，Wang H. Extending interface automata with z notation[M]// Fundamentals of Software Engineering. Berlin Heidelberg：Springer，2012：359-367

[6] Cao Z，Wang H. Hybrid ZIA and its Approximated Refinement Relation[C]// Proceeding of：ENASE 2011-Proceedings of the 6th International Conference on Evaluation of Novel Approaches to Software Engineering. Beijing，China，June 2011：260-265

[7] 狄杨思. 形式规范的自动验证算法的研究[D]. 南京：南京航空航天大学，2012

[8] 狄杨思，曹子宁，王辉. 一种基于形式规范 ZIA 的自动验证方法[C]//火力控制技术/航空电子系统综合技术 2011 年学术年会. 2011：113-120

[9] Clarke E M，Grumberg O，Peled D A. Model checking[M]. MIT press，1999

[10] OMG. UML Profile for MARTE：Modeling and Analysis of Real-time Embedded Systems[OL]. http：//www. omg. org /spec /MARTE/1. 1，2011

[11] Mostafa A M，Ismail M A，El-Bolok H，et al. Toward a formalization of uml2. 0 metamodel using z specifications[C]// Eighth ACIS International Conference on Software Engineering，Artificial Intelligence，Networking，and Parallel/Distributed Computing，2007. SNPD 2007. IEEE，2007，1：694-701

[12] Koymans R. Specifying message passing and time-critical systems with temporal logic[M]. Springer，1992

（下转第 269 页）

对比3个混淆矩阵，可以看出，与两种人工特征方法相比，本文所采用的无监督特征学习方法对3种情感具有最高的识别率，HOG/HOF方法次之，HOG3D方法的识别率最低。其中，本文方法对厌倦的识别率较其他两种方法的提高较为显著。对疑惑的识别率，HOG/HOF方法接近于本文方法，而HOG3D方法的识别率相对较低。

对比3种情感的识别率，可以发现其中愉快的识别率最高，3种方法都达到或超过了80%，本文方法达到了90%；疑惑的识别率次之；厌倦的识别率最低，除本文方法外，其他两种方法的识别率都低于70%，容易和另外两种情感混淆。通过分析视频片段，发现愉快情感的面部表情相对单一，主要是笑脸，较容易识别；疑惑的面部动作主要是眉毛的靠近和眼睑的缩紧，部分受试者的表现非常微小，识别起来有一定难度；而厌倦的面部动作较为多样，包括眨眼、动嘴、上睑下降和打哈欠，还有部分表现也非常微小，因此给识别增加了很大难度。

表4是3种方法的平均识别率和特征向量维度，从中可以看到，本文方法的特征向量维度最低，而识别率最高。相比之下，HOG/HOF方法的特征向量维度高达17万，但没有产生较高的识别率。

表4 3种方法的平均识别率和特征向量维度

方法	平均识别率(%)	特征向量维度(万)
本文方法	82.3	3.2
Dense+HOG/HOF	79.0	17.0
Dense+HOG3D	73.7	6.0

同时还进行了算法耗时实验，在配置为AMD Athlon IIX255，3.1 GHz/8 GB的计算机上用Matlab进行编程测试。从预处理、提取特征到分类识别，本文的方法平均耗时0.004秒/帧，合250帧/秒。这表明，该方法完全能满足实时视频中情感识别的要求。

结束语 本文提出了一种基于无监督提取表情时空特征的情感识别方法，来对愉快、疑惑和厌倦3种情感进行识别。该方法首先对表情视频片段进行规范化预处理，然后采用SISA模型从中学习和提取表情的时空特征，最后用线性SVM进行情感分类。通过实验验证，本文的方法不仅识别速度快，而且相比其他两种人工特征方法，在相对低的特征空间维度下也能有效地提取出人脸表情的时空特征，从而获得较高的识别率。

参考文献

[1] McDaniel B T, D'Mello S K, King B G, et al. Facial features for affective state detection in learning environments[C]// Proceedings of the 29th Annual Cognitive Science Society Conference, 2007. Nashville, TX, USA, Cognitive Science Society, 2007: 467-472

[2] Dahmane M, Meunier J. Emotion recognition using dynamic grid-based hog features[C]// Proceedings of IEEE International Conference and Workshop on Automatic Face and Gesture Recognition, 2011. IEEE, Santa Barbara, CA, USA, 2011: 884-888

[3] Song Y, Morency L P, Davis R. Learning a sparse codebook of facial and body microexpressions for emotion recognition[C]// Proceedings of the 15th ACM on International conference on multimodal interaction, 2013. Sydney, Australia, ACM, 2013: 237-244

[4] Hayat M, Bennamoun M, El-Sallam A. Evaluation of spatiotemporal detectors and descriptors for facial expression recognition [C]// Proceedings of IEEE 5th International Conference on Human System Interactions, 2012. IEEE, Perth, West Australia, 2012: 43-47

[5] Schmidt E M, Kim Y E. Learning emotion-based acoustic features with deep belief networks [C] // Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2011. IEEE, New Paltz, NY, USA, 2011: 65-68

[6] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion [J]. The Journal of Machine Learning Research, 2010, 11: 3371-3408

[7] Le Q V, Zou W Y, Yeung S Y, et al. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011. IEEE, Colorado Springs, CO, USA, 2011: 3361-3368

[8] O' Toole A J, Harms J, Snow S L, et al. A video database of moving faces and people [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 812-816

[9] Lucey P, Cohn J F, Kanade T, et al. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression[C]// Proceedings of IEEE Workshops on Computer Vision and Pattern Recognition, 2010. IEEE, San Francisco, CA, USA, 2010: 94-101

[10] Fan R E, Chang K W, Hsieh C J, et al. LIBLINEAR: A library for large linear classification [J]. The Journal of Machine Learning Research, 2008, 9: 1871-1874

(上接第262页)

[13] Emerson E A, Mok A K, Sistla A P, et al. Quantitative temporal reasoning [C] // Computer-Aided Verification. Berlin Heidelberg: Springer, 1991: 136-145

[14] Alur R, Dill D L. A theory of timed automata[J]. Theoretical computer science, 1994, 126(2): 183-235

[15] Alur R. Timed automata[C]// Computer Aided Verification. Berlin Heidelberg: Springer, 1999: 8-22

[16] Alur R. Techniques for automatic verification of real-time systems[D]. Stanford University, 1991

[17] Alur R, Courcoubetis C, Dill D. Model-checking for real-time systems[C]// Logic in Computer Science, 1990. LICS'90, Proceedings. , Fifth Annual IEEE Symposium on e. IEEE, 1990: 414-425

[18] 钱俊彦，赵岭忠，古天龙. 一种基于时间自动机的时钟等价性优化方法[J]. 计算机工程，2005，9(18)：71-73

[19] Alur R, Courcoubetis C, Dill D. Model-checking in dense real-time[J]. Information and Computation, 1993, 104(1): 2-34