文章编号: 1001-0920 (2014) 01-0083-06 **DOI:** 10.13195/j.kzyjc.2012.1470

嵌入式控制状态转移的确定性实时语义

王剑平, 张云生, 张 果, 张 晶

(昆明理工大学 信息工程与自动化学院，昆明 650500)

摘 要: 嵌入式控制是在实时事件驱动下物理系统状态转移的执行过程, 采用超致密时间 (SDT) 标签表示事件的实时性, 描述一种自然表达时间的实时状态转移语义模型; 证明事件实时标签是时间值上的偏序函数, 实时事件的状态转移顺序与对应的非实时过程一样; 针对状态转移轨迹中的实时约束条件, 把非实时过程上的操作扩展到实时过程, 得到确定性的操作执行顺序. 一个控制横毛织机编织运动的实例表明了语义表达的有效性.

关键词: 嵌入式控制；状态转移；超致密时间；实时约束；确定性操作

中图分类号: TP393 **文献标志码:** A

Real-time semantics of state transition for embedded control systems

WANG Jian-ping, *ZHANG Yun-sheng*, *ZHANG Guo*, *ZHANG Jing*

(Faculty of Information Engineering and Automation，Kunming University of Science and Technology，Kunming 650500，China. Correspondent：WANG Jian-ping，E-mail：kmustwjp@126.com)

Abstract: Embedded control is the procedure execution of the physical system state transition driven by the real-time event. It uses super dense time(SDT) label to indicate the real-time nature of the event, and describes a real-time state transition semantic model which naturally expresses time. It is proved that the real-time event tag is a partial order function in time value, and the state transition sequence of real-time events are same as the corresponding non-real-time process. For the real-time constraints on the trajectory of the state transition, the operation on the non-real-time process is extended to the real-time process to obtain the execution order of determinacy operation. An example of controling flat machine weaving movement shows the effectiveness of the semantic expression.

Key words: embedded control；state transition；super dense time；real-time constraints；determinacy operation

0 引 言

嵌入式控制把一系列计算进程和物理系统紧密组合, 通过计算进程控制物理实体, 物理实体又借助于网络和计算组件实现对环境的感知和控制[1]. 设计具有实时性质的语义模型, 推导系统暂态性能的模态逻辑, 考虑在广域时空条件下程序执行轨迹的计时能力和耗时代价, 以及在不牺牲系统性能和资源的前提下提高系统状态控制的确定性和实时精度是嵌入式控制技术的难点.

在形式语言模型中, 系统执行过程是一组事件驱动的状态转移轨迹. 如果事件间隔是有限的, 则可以用事件发生顺序表达执行过程. 但是这样的语义在表达与物理过程相互作用时, 有许多与实际不相符合的地方. 关键是要在连续时间的某个确定时刻选择发生某个确定行为, 因此必须考虑无限事件序列. 最直接的方法是在执行序列中增加与之配对的时间序列, 其中第 i 个元素给出第 i 个事件发生的时间.

嵌入式控制状态转移的确定实时语义的研究主要集中于语义一致性、时间和并发行为描述、系统中的动态行为描述等方面[2]. 文献 [3-4] 提出一种基于标签概念的模型用于系统信号事件的标记, 支持时间进化、信号间协调和因果关系的描述, 但时间标签不能提供任何的实时确定性执行语义. 文献 [5] 提出一种基于 Kahn 过程理论的确定性语义系统功能理论, 但对于离散连续混合系统仍然存在局限性. 在嵌入式控制的实时时间约束的研究中, 引入时间信息的常用方法是把状态转移的上限时间和下限时间联系起来, 如时间 Petri 网[6]、时间转移系统[7-8]、实时自动机

收稿日期: 2012-10-05；**修回日期:** 2013-03-24.

基金项目: 国家自然科学基金项目(61364008, 61263017)；云南省自然科学基金项目(2009CD041, 2010CD038)；云南省教育厅重点基金项目(2013Z127)；昆明理工大学人才培养项目(14118596).

作者简介: 王剑平(1975−), 男, 副教授, 博士, 从事信息物理系统、实时嵌入式系统的研究；张云生(1948−), 男, 教授, 博士生导师, 从事实时系统、复杂系统等研究.

系统[9]和时序图方法[10]. 但考虑状态转移不仅要表达转移之间的时间间隔, 而且必须直接表示出转移时刻, 硬实时系统调度算法[11]用矩阵描述每条状态转移路径延迟间隔的约束条件, 给出了一个常数k, 使得在单位长度的时间间隔中至多可能发生k次转移. 此外, 还有其他一些扩展的定量时间暂态逻辑方法[12], 这些方法大多借用了离散时间或虚拟时钟语义逻辑.

离散时间模型要求时间序列是单增的整数序列, 这种模型适应于一定种类的同步程序语义, 例如同步数字电路, 其中认为状态变化在时钟信号到达时准确发生. 它的优点之一是能容易地转换成一种普通的形式语言, 每一步执行轨迹增加一时间步长. 在实际发生的事件之间插入占用时间的空闲事件, 一旦程序执行, 每个事件的时间与其位置相同, 这样可以略去时间序列, 只留下普通字符顺序, 因此离散时间行为可以采用普通计算操作语义控制. 但物理过程中事件并非总是发生在整数值时间, 而离散时间模型又要求先验地选择某些固定值近似连续时间, 这就限制了控制精确性.

除了要求时间序列是整数, 虚拟时间模型类似于离散时间模型. 在此模型中事件以实数值时间按指定顺序发生, 但在轨迹中仅记录与数字时钟对应的时间, 仍然是整数. 这种模型也容易转换成传统的形式语言. 首先增加一时间片tick到事件集, 将包含有时间轨迹的事件按同样的顺序对应到无时间轨迹的事件集; 然后, 在第i个和第$i+1$个事件之间插入$t_{i+1}-t_i$的tick数, 该数可以是0. 这样控制操作简单, 但缺点是仅以近似的意义表达时间.

本文基于超致密时间(SDT)模型[13], 研究建立连续时间控制的程序符号语义模型, 讨论在超致密时间约束条件下有限状态转移的确定性. 实际例子表明了该语义模型的有效性.

1 状态转移语义模型

在连续时间领域操作的物理过程, 描述事件时间的是致密的非负实数集R_+, 无界地单调增加, 反映出牛顿物理时间范畴. 实数值时间变量导致了把无限的状态转移控制转换成形式语言的困难.

1.1 状态转移轨迹

事件可以记为对一变量赋值, 或者发出一条指令, 或者接收到一条外部消息等. 通过状态转移语义模型关联一组可观察的事件和状态转移过程.

定义 1 元组$\langle A,S,s_0,E\rangle$表示事件驱动状态转移的语义符号. 其中: $a\in A$为输入事件字符集, $s\in S(T,V)$为状态集, $s_0\subseteq S$为初始状态集, $E\subseteq S\times S\times A$为状态转移路径[13].

系统从初始状态开始, 如果$(s,s',a)\in E$, 则表示系统在发生事件a后, 状态从s改变到s'. 事件集A上的$a=a_1a_2a_3\cdots$, 状态转移轨迹$r:s_0(a_1)\to s_1(a_2)\to s_2(a_3)\to\cdots$是在$a$作用下的状态转移过程. 对于所有$i\geqslant 1$, 由$\langle s_{i-1},s_i,a_i\rangle$组成的无限多状态转移路径记作$\inf(r)$. 如果对状态转移增加约束条件, 即增加可行状态集$F\subseteq S$, 则元组$\langle A,S,s_0,E,F\rangle$表示在事件$a\in A$上的运行$r$是可行运行, 即当且仅当$\inf(r)\bigcap F\neq\varnothing$(即非空), r是可行的.

上述定义的状态转移与时间无关, 轨迹仅记录状态转移顺序. 如果两个事件a和b同时发生, 则对应的事件集合是$\{a,b\}$. 如图1描述的两个事件$\{a,b\}$上的状态转移可表述为如下语句:

$$L_1=\{(a+b)a\}. \tag{1}$$

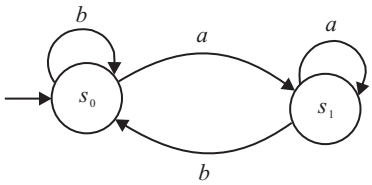


图 1 非实时状态转移

1.2 事件实时标签

实际上状态转移的执行过程不仅要记录事件序列, 而且要指明事件发生的时间, 系统的行为与对事件响应的时间密切相关. 因此可在事件字符上耦合一实数时间值来定义事件实时标签.

定义 2 (a,t)是事件A上的实时标签. 其中: $a=a_1a_2a_3\cdots$是A上的无限事件序列; $t=t_1t_2t_3\cdots$是实数时间值$t_i\in R_+$的无限序列, 且满足如下约束:

1) t严格单调递增, 即对于所有$i\geqslant 1, t_i<t_{i+1}$;

2) 在执行过程中, 对每一时间值$t\in R_+$, 总有$i\geqslant 1$, 使得$t_i>t$;

3) 特别地, 同一时间值可能与连续事件相关, 因此t仅单调增加, 即对于所有$i\geqslant 1, t_i\leqslant t_{i+1}$.

2 实时状态转移

一般状态转移根据事件输入字符选择下一状态, 在实时情况下, 状态转移还必须考虑与事件关联的时间要求.

2.1 实时状态转移语义

扩展状态转移语义到实时情况, 使执行轨迹能够准确地记录转移时间.

定义 3 在状态转移语义符号中增加2进制时钟集C, 则元组$\langle A,S,s_0,C,E\rangle$给出了实时状态转移语义. 其中: $E\subseteq S\times S\times A\times 2^C\times\Delta(C)$给出一组转移路径; $\langle s,s',a,\lambda,\delta\rangle\in E$表示系统在输入符号$a$后, 状态从$s$改变到$s'$; $\delta\subseteq\Delta(C)$表示时间约束; $\lambda\subseteq C$表

示在转移时重设时钟[13].

实时状态转移轨迹 $r:(s_0,v_0)(a_1,\tau_1)\to(s_1,v_1)(a_2,\tau_3)\to\cdots$ 表示路径 $\langle s_i,s_{i+1},a_i,\lambda_i,\delta_i\rangle$ 从状态 $((s_0,v_0)=0)$ 开始, 要求在任意状态 (s_i,v_i), 时钟读取的值等于从它被设置的上一时间开始消逝的时间, 同时可以重置时钟为 0. $v_i=[\lambda_i\mapsto 0](v_{i-1}+\tau_i-\tau_{i-1})$ 满足约束 δ_i.

同样, 对状态转移增加约束条件 F, 当且仅当 $\inf(r)\bigcap F\neq\varnothing$, 元组 $\langle A,S,s_0,C,E,F\rangle$ 表示 $r=(s,v)$ 的运行是可行的.

图 2 给出了实时状态转移, 其执行语句是

$$L_2=\{(ab,t)|\forall i.(t_{2i}<t_{2i-1}+2)\}. \tag{2}$$

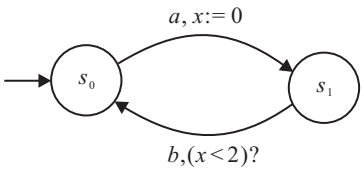


图 2　实时状态转移

图 2 中, s_0 为初始状态, x 为状态转移的时钟记录, $x:=0$ 表示状态转移时使时钟 x 重置为 0, $(x<2)?$ 表示判定时间是否满足小于 2 的约束条件.

如果采用两个时钟 x 和 y (见图 3), 则执行语句为

$$L_3=\{(abcd,t)|\forall i.((t_{4i+3}<t_{4i+1}+1)\bigwedge (t_{4i+4}>t_{4i+2}+2))\}. \tag{3}$$

该语句表示状态在 s_0,s_1,s_2 和 s_3 之间循环, 时钟每次重置为 0. 读取 a 状态从 s_0 转移到 s_1, 读取 c 从 s_2 转移到 s_3, 但要检查 $(x<1)?$ 以保证在上次发生 a 的时间值 1 之内发生 c. 类似地, 设置另一时钟 y 的约束条件, 以保证 b 和随后 d 之间的延迟总是大于 2.

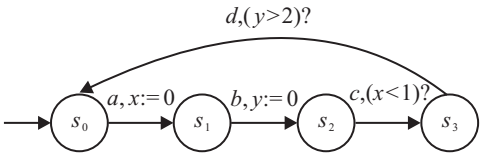


图 3　两个时钟记录的状态转移

注意在上述例子中, 明确约束了 a、c 之间和 b、d 之间的延迟, 并不须在 a 和随后 b 之间, 以及 c 和随后 d 之间直接设置任何时间约束, 这是使用了两个时钟 x 和 y 的优点. 实际上 L_4 可以定义为两个语句, 即

$$L_3^1=\{(abcd,x)|\forall i.(t_{4i+3}<t_{4i+1}+1\}, \tag{4}$$

$$L_3^2=\{(abcd,y)|\forall i.(t_{4i+4}<t_{4i+2}+2\}. \tag{5}$$

注意这里两个时钟对应于同一个固定的总体时间, 并以同一速率计数时间增加.

2.2　时间约束条件

时间约束表示仅当时钟当前值满足约束条件时才发生转移. 最简单的约束可以是时钟值与一时间常数比较, 也可以是这种简单约束的布尔组合, 为便于设置, 时间常数可以是任何非负有理数 Q.

定义 4　对于时钟变量集 X, 时钟约束

$$\Delta(X):=x\leqslant c|x\geqslant c|\neg\delta(x)|\delta_1(x)\wedge\delta_2(x).$$

其中: $x\in X$ 为时钟值, $c\in Q$ 为常数.

对时钟 x 计算一实数时间值 $v(x)$, 它是 X 到 Q 的映射, 当且仅当判定 v 给出的值对 Δ 是真时, 称状态 (s,v) 满足约束条件. 显然诸如布尔值 true、$(x=c)$、$x\in[2,6)$ 都是简单约束条件.

2.3　超致密时间模型

在实数值表示的致密时间下, 任意多的事件可能在有限时间间隔发生, 而且事件可能相互任意接近. 例如考虑图 4 表达的语句

$$L_4=\{(ab,t)|\forall i.((t_{2i-1}=i)\bigwedge (t_{2i}-t_{2i-1}>t_{2i+2}-t_{2i+1}))\}, \tag{6}$$

其时间 a 和随后 b 之间的时间间隔严格递减. 其中一个实时标签样本序列为

$$(a,1)\to(b,1.5)\to(a,2)\to(b,2.25)\to$$
$$(a,3)\to(b,3.125)\to\cdots,$$

可以看出不断扩展的无限状态是不可计数的. 如果要求所有时间值 t_i 是任意小的固定常数的倍数, 则语义表达将是空的, 所以不能以时钟计数判定状态转移的实际时间约束条件. 但如果考虑采用具有偏序性质的超致密时间模型对状态扩展进行排序, 则可以将描述事件的时间轨迹表示为执行顺序.

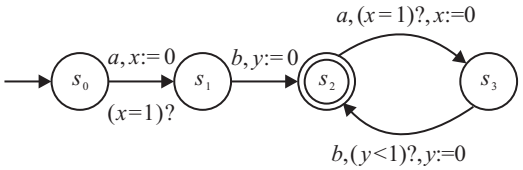


图 4　事件时间间隔变化的状态转移

定义 5　时间序列 $(\tau,n)\in(T=R_+\times N)$ 表示超致密时间 (SDT) [14]. 其中: $\tau\in R_+$ 为步长, $n\in N$ 为步数. 如果 $(\tau,n)\geqslant(\tau',n')$, 则 $\tau\geqslant\tau'$, 或 $\tau=\tau'$ 和 $n\geqslant n'$.

SDT 模型对物理系统的控制提供了在严格意义上比实数更好的时间模型.

定义 6　给定 $(\tau,n)\in T$, 集合 T 上的

$$D[\tau,n]=\{(\tau',n')\in T|(\tau',n')\leqslant(\tau,n)\},$$
$$D(\tau,n)=\{(\tau',n')\in T|(\tau',n')<(\tau,n)\}$$

是 (τ,n) 产生的闭的和开的下降集; 而

$$D(\tau,\infty)=\{(\tau',n')\in T|\tau'\leqslant\tau\}$$

是 τ 产生的无限下降集.

下面的定理将表明(τ,n)是T上的偏序集$(T,\leqslant)$[14], 具有字典一样的前缀顺序.

定理 1 T的下降集有如下4种形式之一:

1) 对于某$(\tau,n)\in T$, T的下降集是$D[\tau,n]$;

2) 对于某$(\tau,n)\in T$, T的下降集是$D(\tau,n)$;

3) 对于某$\tau\in R_+$, T的下降集是$D(\tau,\infty)$;

4) T的下降集是T本身.

证明 假设D是一T的下降集, 如果$D=\varnothing$, 则$D=D(0,0)$.

由定义6, 如果D非空, 若D中有一最大元素(τ,n), 则$D=D[\tau,n]$. 若D的上界是$(\tau,n)\in T$, 但D中没有最大元素, 则$T\backslash D$是D的上集. 即若$(\tau,n)\in T\backslash D$、$(\tau',n')\in T$和$(\tau,n)\leqslant(\tau',n')$, 则$(\tau',n')\in T\backslash D$. 当$(\tau,n)$是$T\backslash D$的最小元素时, $D=D(\tau,n)$.

如果D有上界而没有最大元素, 同时$T\backslash D$没有最小元素, 注意$T\backslash D$下有界和D非空, 则假设

$$T=\{\tau\in R_+|\exists n\in N,(\tau,n)\in D\}.$$

若T中有最大元素$\tau\in R_+$, 由于D在T中, 而D没有最大元素, 则有$D=D(\tau,\infty)$; 若T没有最大元素, 则$R_+\backslash T$有最小元素τ. 因为$(\tau,0)$不可能属于D, 所以$(\tau,0)$是D的最小上界. 而$(\tau,0)$是$T\backslash D$的最小元素, 这与$T\backslash D$没有最小元素相矛盾, 所以$D=D(\tau,\infty)$.

如果D没有上界, 则D一定是T本身. □

通过对(τ,n)排序, 事件实时标签(a,t)是定义在T的下降集D上的偏序函数.

下面的引理证明将用超致密时间表达时间序列, 实时状态转移与非实时转移的顺序一样.

引理 1 设L是实时规则语言, 对于每个事件字符a, 当且仅当存在时间序列$t\in(\tau,n)$, 使得$(a,t)\in L$, 则有$a\in\mathrm{Untime}(L)$, 这里$\mathrm{Untime}(L)$是非实时语言表达.

证明 必要性. 如果存在时间序列$t\in(\tau,n)$, 使得$(a,t)\in L$, 则根据定理1, 事件的时间序列是排序的, 可以略去, 所以$a\in\mathrm{Untime}(L)$.

充分性. 假设t是任意实数值时间序列, $(a,t)\in L$. 如果有$\varepsilon\in R_+$, 使状态转移的时钟约束常数是ε的整数倍, 即对于某些$0\leqslant j\leqslant i,n\in N,t_i=t_j+n\varepsilon$, 则可以选择用超致密时间$(\tau,n)$表示$t_i=n_i\tau$; 否则$\tau_i\in R_+$, 对于所有$0\leqslant j\leqslant i$, $n\in N$, 总有$\tau_i-\tau_j=n\varepsilon'<n\varepsilon$, 由于$R_+$的致密性, 可以选择$(\tau_i-\tau_j)$任意接近$n\varepsilon$. 显然, 在第$i$次状态转移点沿着两个时间序列$(t_i-t_j)$和$(\tau_i-\tau_j)$的值满足同一组约束, 因此可以构造$(a,(\tau,n))$作用的运行$r'=(s',v'(\tau,n))$, 服从与$r=(s,v(t))$相同的运行路径. 特别地, 令$v_0{}'=v_0$, 如果沿$r$的第$i$次转移是$(s_{i-1},s_i,a_i,\lambda_i,\phi_i)$, 则令$v_i{}'=[\lambda_i\mapsto0](v_{i-1}{}'+\tau_i-\tau_{i-1})$, 超密时间序列表达了同样的作用. □

3 确定操作语义

在状态转移轨迹中, 由于事件a的数量没有限制, 不断扩展的状态数量是无限的(实质上是不可计数的), 将需要无限制的时钟计数, 可以证明其时间复杂度呈指数增长. 考虑时钟约束的可表达性, 定义$t=(\tau,n),\tau\in Q,n\in N$是实时约束.

3.1 时钟约束选择

对于每一$x\in C$, 设实时状态转移约束n_x是满足时钟约束

$$\{x=n|n=0,1,\cdots,n_x\}\bigcup$$

$$\{(n-1)\tau<x<n\tau|n=1,\cdots,n_x\}\bigcup\{x>n_x\}$$

的最大整数.

引理 2 实时状态转移$\langle A,S,s_0,C,E\rangle$中, (s,v)是实时标签(a,t)上的运行, 当且仅当$(s,v(\tau,n))$是$(a,(\tau,n))$上的运行.

证明 如果τ是所有出现在状态转移表中时钟约束常数的最小公倍数, 则对于时钟的约束仅是τ的整数倍n, 即$n\tau=v(t),n\leqslant n_x$. 对于任意时间约束常数$v(t)$而言, 总可以找到一个$\tau$, 使$(n-1)\tau\leqslant v(t)\leqslant n\tau$无限地接近于$v(t)$, 因此可以选择$(s,v(\tau,n))$在$(a,(\tau,n))$上的运行非常接近地替代$(s,v(t))$在$(a,t)$上的运行. □

引理2表明可以选择仅包括整系数n的时钟约束, 由$n\tau$排列获得状态转移顺序.

3.2 状态转移操作

由引理1和引理2, 下面的定理表明在超致密时间模型下, 事件实时标签$(a,(\tau,n))$集A的实时执行过程$\mathrm{time}((A,L))$, 与事件序列$a\in\mathrm{Untime}(L)$组成的非实时过程$\mathrm{Untime}((A,L))$顺序相同.

定理 2 如果r是事件集a上的顺序运行, 则存在一时间序列$t=(\tau,n)$和(a,t)上的运行r', 使得r等于$[r']$, 这里$[r']$是(a,t)上的顺序运行.

证明 首先, a的顺序运行r从初始状态s_0开始, 在第i步r_i, a_i使s_{i-1}转移到s_i. 考虑(a,t)上的运行r', 时间序列是超致密表达, $t=(\tau,n)$, 则从(s_0,v_0)开始, 一步步地在时间$t_i=(\tau,i)$时, a_i使(s_{i-1},v_{i-1})转移到(s_i,v_i), $\langle s_{i-1},s_i,a_i,\lambda_i,\delta_i\rangle\in E$使得$v_i$满足$\delta_i$且$v_i=[\lambda_i\mapsto0](v_{i-1}+t_i-t_{i-1})$. 定理1表明, 事件实时标签$(a,t)$是定义在$(\tau,n)\in T$的下降集$D$上的偏序函数, 通过对$(\tau,n)$排序, (a_i,t_i)与a_i的顺序一致, 所以在(a_i,t_i)上的运行顺序与r_i一致, 即$[r']=r$.

另外, 如果有另一时间序列$t'=(\tau',n)$的转移

过程 r'', 虽然 $t_i' = (\tau', i) \neq (\tau, i) = t_i$, 选择整系数的时钟约束, 沿 r'' 使 (s'_{i-1}, v'_{i-1}) 到 (s_i', v_i') 的过程与 r' 一样, 同时使 $v_i' = [\lambda_i' \mapsto 0](v'_{i-1} + t_i' - t'_{i-1})$, 则在时间 $t_i' = (\tau', i)$ 发生的事件一定是 a_i, 显然在 (a_i, t_i') 上的顺序也与 r_i 一致, 所以 (a, t) 上的运行顺序是唯一的, 即 $[r''] = [r'] = r$. □

至此, 将非实时过程上的操作顺序扩展到实时过程, 与事件相关的时间值可由操作顺序排除.

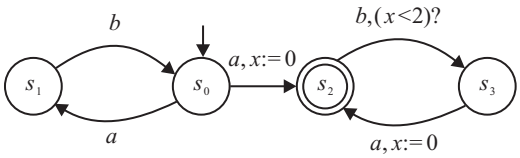
3.3　确定性实时状态转移

在非实时情况下, 确定性的状态转移有单一的初始状态, 给定一输入事件, 则下一状态是唯一确定的. 对于实时过程希望有类似的确定原则.

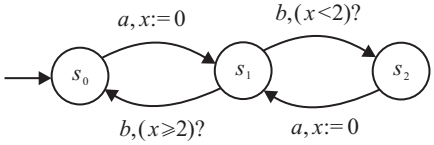
定义 7　实时状态转移 $\langle A, S, s_0, C, E\rangle$ 是确定的, 当且仅当:

1) 它仅有一个初始状态 $|s_0| = 1$;

2) 对于所有的 $s \in S$, $a \in A$, 对 $\langle s, -, a, -, \delta_1\rangle$ 和 $\langle s, -, a, -, \delta_2\rangle$ 形式地每对转移, 时钟约束 δ_1 和 δ_2 是相互排斥的, 即 $\delta_1 \bigwedge \delta_2 = \varnothing$ 不存在.



(a)　非确定的状态转移



(b)　确定的状态转移

图 5　确定及非确定的状态转移

比较图 5(a) 和图 5(b), 有同一状态转移语句

$$L_5 = \{(ab, \tau) | \exists i, \forall j \geqslant i(\tau_{2j} < (\tau_{2j-1} + 2))\}. \quad (7)$$

s_0 是初始状态, s_2 是允许状态. 图 5(a) 中状态 s_0 和 s_1 之间由 a、b 交替输入, 循环一段时间后非确定地转移到 s_2, 设 $x = 0$; 然后每 2 个时间单位内 s_2 和 s_3 之间循环一次. 图 5(b) 中状态从 s_0 转移到 s_1, s_1 上的两个约束 $x < 2$ 和 $x \geqslant 2$ 相互排斥, 所以状态转移 $\{s_1, s_2\}$ 是确定的.

定理 3　一个确定实时状态转移在给定的实时标签上至多只有一个运行.

证明　考虑在实时标签 (a, t) 上的运行从唯一初始状态 (s_0, v_0) 开始, 假设在时间 t_{i-1} 已运行到第 $i-1$ 步. 下一步的状态是 (s_i, v_i), 由转移的确定性知, 在时间 t_i 至多只有一种转移 $\langle s_{i-1}, s_i, a_i, \lambda_i, \delta_i\rangle$ 使 $v_i = v_{i-1} + t_i - t_{i-1}$ 满足 δ_i. 如果这样的转移不存在, 则没有 (a, t) 上的运行. □

3.4　实时控制实例

考虑一自动毛织横机的例子. 机头在机架上来回运行, 每当向左或向右到位时, 停留等待加针操作. 由于机头惯性较大, 采用定位开关控制其位置不精准. 如果采用确定性计时语义控制策略, 则可以准确地控制机头定位和加针操作. 过程由 3 部分执行: 机头、加针器和控制器. 如图 6 所示, 指令集是 {Right, Left, Adding, Back}, $n_m\tau$ 记为运动时间, 其中 τ 是与运动速度对应的有理数单位时长, n_m 是与编织宽度对应的步长参数, 加针循环次数由参数 n_a 给定. 两个时钟 x 和 y 分别计机头和加针运行时间. 语句表达为

$$L = \{(\mathrm{RALAB}, t) | \forall i.(t_{i+1} = t_i + n_m) \bigwedge (t_{i+1} = n_a)\}. \quad (8)$$

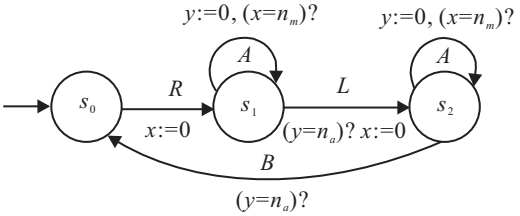


图 6　自动毛织横机控制实例

控制器发送指令 Right 给机头, 置 $x := 0$, 机头向右到状态 s_1; 判定延迟时间 $(x = n_m)$, 发送指令 Adding 给加针器, 置 $y := 0$, 循环加针; 在状态 s_1 判定加针计数 $(y = n_a)$? 同时置 $x := 0$, 机头向左到 s_2; 再判定 $(x = n_m)$? 置 $y := 0$, 循环加针; 再判定加针计数 $(y = n_a)$? 然后返回状态 s_0 完成一次编织运动.

根据上述建模中可变位置的实时控制实例, 通过确定性语义建模方法, 克服了传统建模中实数值时间变量导致把无限的状态转移转换为形式语言的困难, 同时实例应用证实了用超致密时间表达时间序列, 其状态转移与非实时状态转移的顺序一致.

4　结　论

本文用超致密时间表达嵌入式控制的实时执行过程, 通过定义事件的实时标签, 在实时状态转移轨迹上建立了一种简单的状态转移语义模型. 选择时钟与时间常数及其布尔组合比较作为约束条件, 用有限多的时钟值注释了具有时间约束的状态转移. 由超致密时间的偏序性质, 证明了事件实时标签是实数值时间上的偏序函数, 时钟约束可以仅包括整系数的排列获得状态转移顺序. 实时状态转移与非实时转移的顺序一样, 可以把非实时过程上的操作扩展到实时过程. 通过在给定的实时标签上至多只有一个运行的证明, 得到了确定性实时状态转移操作语义. 一个具体运动控制例子表明了将执行过程变换为一组关联指令执

行顺序的有效性.

如果考虑事件序列本身的随机性, 则可以扩展本文讨论的语义模型表达形式, 通过插入统计信息确定具有最大最小延迟时间的分布系统状态转移约束, 这样对与物理过程交互作用的嵌入式控制系统将更为有用.

参考文献(References)

[1] Lee E A. Computing foundations and practice for cyber-physical systems: A preliminary report[R]. Berkeley: University of California, 2007.

[2] Derler P, Vincentelli A S. Modeling cyber-physical systems[J]. Proc of the IEEE, 2012, 100(1): 13-28.

[3] Eidson J C, Lee E A, Matic S, et al. Time-centric models for designing embedded cyber-physical systems[R]. Berkeley: University of California, 2009.

[4] Benveniste A, Bourke T, Caillaud B, et al. Non-standard semantics of hybrid systems modelers[J]. J of Computer and System Sciences, 2012, 78(3): 877-910.

[5] Caspi P, Benveniste A, Lublinerman R, et al. Actors without directors: A Kahnian view of heterogeneous systems[J]. Hybrid Systems: Computation and Control, 2009, 5469(1): 46-60.

[6] Ramchandani C. Analysis of asynchronous concurrent systems by Petri nets[R]. Massachusetts: Massachusetts Institute of Technology Cambridge Project MAC, 1974.

[7] Ostroff J S. Temporal logic for real-time systems[M]. Cambridge: Research Studies Press, 1989: 260-262.

[8] Henzinger T, Manna Z, Pnueli A. Temporal proof methodologies for real-time systems[C]. Proc of the 18th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. New York: ACM, 1991: 353-366.

[9] Bengtsson J, Yi W. Timed automata: Semantics, algorithms and tools[J]. Lectures on Concurrency and Petri Nets, 2004, 3098(1): 87-124.

[10] 王剑平, 张云生, 张果, 等. 并行分布控制网络的实时信号时序流图分析[J]. 控制与决策, 2010, 25(11): 1727-1731.
(Wang J P, Zhang Y S, Zhang G, et al. Analysis of signal timing sequence flow chart on parallel and distribute control network[J]. Control and Decision, 2010, 25(11): 1727-1731.)

[11] Thiele L, Chakraborty S, Naedele M. Real-time calculus for scheduling hard real-time systems[C]. The 2000 IEEE Int Symposium on Circuits and Systems. Geneva: IEEE, 2000, 4: 101-104.

[12] Alur R, Henzinger T A. A really temporal logic[C]. The 30th Annual Symposium on Foundations of Computer Science. New York: IEEE, 1989: 164-169.

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

[14] Manna Z, Pnueli A. Verifying hybrid systems[J]. Hybrid Systems, 1993, 736(1): 4-35.

[15] Munkres J R. Topology[M]. 2nd ed. Englewood Cliffs: Prentice Hall, 2000: 4-25.

(责任编辑: 李君玲)