授予单位代码：10459

研究生学号：98134

密　　级：

郑州大学
硕士学位论文

论文题目：关于时间自动机及其构造区域自动机的算法

姓　　名：宋　煌

学科门类：工　学

一级学科：计算机科学与技术

专　　业：计算机软件与理论

研究方向：形式语言与自动机

导　　师：庄雷　教授

日　　期：2001年5月

关于时间自动机及其构造区域自动机的算法

摘要

论文首先给出了时间自动机的定义及相关符号，然后对使用等价类方法构造区域自动机的描述进行了算法实现。该算法的空间复杂度是状态数目的几何级扩张。对上述算法做进一步改进，使得平均时间复杂度有所改善。通过分析时间自动机的时间约束条件，又提出了一种新的构造区域自动机的算法，该算法的时间复杂度与空间复杂度都有了较大的改善。接着介绍了时间自动机的最新进展，即有关事件记录自动机，事件预测自动机以及事件时钟自动机。对上述自动机的有关性质及它们所识别的语言类之间的关系进行了研究，得出一些有益的结论。论文最后给出了一个使用时间自动机的验证实例。

关键字：时间自动机，时间后继，区域自动机，位移值，位移空间，相容值，相容空间

ABSTRACT

First of all, the paper presents the basic definition and the symbols of timed automaton. Second realizes the construction of region automata through equivalence class, but the space complexity is the exponential blowup in the number of state in the construction algorithm of region automata. To get a better average space complexity, improves on it And then analyzes the time constraint of region automata, gets a new method to construct region automaton. Ameliorates both the time complexity and the space complexity. Third, introduces the new development of timed automaton that is event-recording automata, event-predicting and event-clock automata. And then researches the property and the relation of language it given, proposes some interesting conclusion. At last gives an instance of verification.

Key words: timed automaton, time successive, region automaton, displacement value, displacement space, coherent value, coherent space,

目 录

引 言

自动机作为一种有效的形式化工具无论是在理论还是在实践中都有着很重要的作用。在计算机科学领域内，有限自动机是模拟分析许多现象的工具，特别是在并行有限状态系统中，自动机理论有很重要的地位。人们一直在使用ω-自动机进行并行系统的推导。在并行计算的模型计算中，系统是由其行为来区别的。设由一系列状态或事件来代表行为，系统的行为集合就是形式语言。人们已经研究出各种各样的计算形式化，例如非确定 Büchi 自动机，确定与非确定的 Muller 自动机，ω-正则表达式，时序逻辑的形式方程等等。需要指出，各类计算形式化具有相同的表达能力。已经证明一个系统就可用产生该语言的自动机来模拟。复合或复杂系统是通过产生模拟其子系统的自动机而来的。由于系统接受的行为同样组成一形式语言。所以要求说明能通过另一自动机给出。通过诸如线形临时逻辑之类的计算形式化的事实，自动机作为形式化说明的充分性已经得到验证。对于一个系统满足其要求的检查验证问题，就转变为检查俩个自动机的语言包含问题。语言包含的确定过程一般包括自动机表示说明的补。其依次依赖于确定性。

有限状态自动机能够很方便的描述有限状态系统，但是对于状态变换还与时间相关的系统的描述就显得无能为力。此类形式化仅保留事件序列。尽管撇开时间的定量因素来考虑具有许多优点，但是当系统推论与物理进程相互影响时，比如飞机场的控制系统，就要求精确考虑实时因素。时间有限自动机理论是在有限自动机基础上发展而来的，在本质上与ω-正则语言相同。为捕捉实时系统的行为，计算模型需要用时间概念扩展。为了研究诸如并行系统、实时系统之类与时间相关的系统，在自动机中引入时间，使用时间自动机来表述、验证此类系统。以该理论为基础的解决方案，已经应用于一些自动工具，例如 COSPAN、KRONOS 和 UPPAAL，但对于时间自动机的一般验证问题即语言包含问题还未确定。因为，与非时间化问题不同，时间自动机的非确定多样性比确定多样性有着更强的表达能力。

时间自动机理论中，区域自动机有着很重要的地位。在系统设计中存在这么一些事件，与其相关的延迟在约束的范围外，这类事件称为时间不一致行为。而区域自动机就是排除时间不一致行为的重要工具。但是，区域自动机的几何级空间复杂度是进行验证、排除时间不一致行为的重要障碍。本文实现了区域自动机的构造算法，并改进了该算法。通过分析时间自动机的约束，提出了一种新的区域自动机构造算法，改进了其几何级的空间复杂度。由于一个转换能够发生，首先，这个转换必然是发生于某一段时间区间内的，其次在这段时间区域内，所有的约束必然同时满足（否则转换就不可能发生）。那么就可以通过判断某个时钟值是否在这个区间内，最终判断与此时间相联系的事件是否时间一致。

第一章 时间自动机的定义

1.1 ω-自动机

先简要介绍一下相关的ω-正则语言理论。正则语言的形式化定义是给定有限字母表上的有限字集。而ω-语言则是由无限字的集合组成的。如前所述，本文研究的时间自动机所接受的就是时间序列上的无限事件序列。Σ^{ω}表示有限字母表Σ上的无限字的集合，则有限字母表Σ上的ω-语言是Σ^{ω}的子集。ω-自动机同非确定有限自动机等价，但是为了处理无限的输入字，其接受条件作了相应的改变。本文主要讨论两种类型的ω-自动机：Büchi 自动机与 Muller 自动机。

转换表A是元组$\langle \Sigma, S, S_0, E\rangle$，其中$\Sigma$是输入字母表，S 是自动机状态的有限状态集合，$S_0 \subseteq S$是初始状态集合，$E \subseteq S \times S \times \Sigma$是边集。自动机起始于某一初始状态，如果$\langle s, s', a\rangle \in E$，那么自动机就可以读入输入符 a，状态从 s 转为 s′ 。

对字母表上的字$\sigma = \sigma_1\sigma_2\ldots$

$$r: s_0 \xrightarrow{\sigma_1} s_1 \xrightarrow{\sigma_2} s_2 \xrightarrow{\sigma_3} \ldots$$

如果$s_0 \in S$并且对任意的$i \geq 1$，$\langle s_{i-1}, s_i, \sigma_i\rangle \in E$，我们就说 r 是 A 在$\sigma$上的运行。对此运行，集合 inf（r）由状态 s 组成$s \in S$，使得$s = s_i$，其中 s 出现无限多次，$i \geq 0$。

对转换表加入不同的接受条件，就可以定义不同类型的ω-自动机。Büchi 自动机 A 是具有接受状态集合$F \subseteq S$的转换表$\langle \Sigma, S, S_0, E\rangle$。A 在$\Sigma$上的时间字$\sigma \in \Sigma^{\omega}$的运行是可接受的，当且仅当 inf（r）$\cap F \neq \varnothing$。换句话说，运行 r 是可接受的当且仅当 F 中的某些状态在 r 中重复无限多次。由 A 所 接受的语言，是 A 接受的由Σ^{ω}上的字σ的集合。

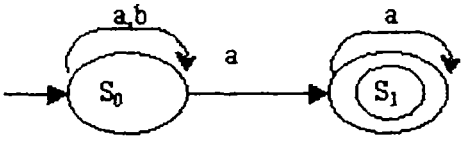


图 1.1 接受 $(a+b)^*a^{\omega}$ 的 Büchi 自动机

例 1.1 ：如图 1.1 所示，其所对应转换表为$\langle \{a,b\}, \{s_0,s_1\}, \{s_0\}, E\rangle$，$E=\{\langle s_0, s_0, a\rangle, \langle s_0, s_0, b\rangle, \langle s_0, s_1, a\rangle, \langle s_1, s_0, a\rangle,\}$，每一个自动机可接受的运行具有形式：

$$r: s_0 \xrightarrow{\sigma_1} s_0 \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} s_0 \xrightarrow{a} s_1 \xrightarrow{a} s_1 \xrightarrow{a} \ldots$$

$\sigma_i \in \{a,b\}$，$1 \leq i \leq n$，$n \geq 0$。自动机接受具有有限个 b 的语言。即语言$L_0=(a+b)^*a^{\omega}$。

ω语言是ω-正则的当且仅当其被某个 Büchi 自动机所接受。因此，例 1.1 所示自动机是ω-正则语言。ω-正则语言在布尔操作下封闭，通过求出 Buhi 自动机的乘积可以求出其交。人们已经找到求 Büchi 自动机补的构造方法。当 Buhi 自动机用来模拟有限状态并行进程时，验证问题就转变为求语言的包含问题。而ω-正则语言的包含问题是可判定的。为检查一个自动机所接受的语言是否包含于另一个自动机所产生的语言，需要判断前者与后者补的交集是否为空。而判空过程相对容易，仅需要检查是否有一个可达的循环，其至少含有应该和可接受的状态。一般而言，Büchi 自动机的求补过程是个状态数目成几何级扩展的过程，而语言包含问题是 PSPACE 完全的。但是检查确定自动机的语言包含问题却可以在多项式时间内完成。转换表〈Σ，S，S_0，E〉是确定的，当且仅当（1）仅有一个起始状态，即$|S_0|=1$；（2）起始于某一状态 s（s∈S），以 a（a∈Σ）为标志的边至多有一条。因此，对一个确定的转换表而言，当前状态和下一个输入符唯一确定下一个状态。相应的，对一个给定的字，确定的 Büchi 自动机至多有一个运行。与关于有限字的时间自动机不同，确定的 Büchi 自动机接受的语言类比ω-正则语言类小的多。例如，不存在确定的 Büchi 自动机接受例 1.1 所示的语言 L_0。而 Muller 自动机以一个更强的接受条件避免了此问题。

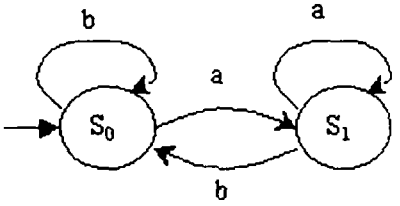


图 1.2 接受(a+b)*a$^\omega$的确定 Muller 自动机

Muller 自动机是一转换表〈Σ，S，S_0，E〉，其具有一个接受簇 $F\subseteq 2^S$。A 在字 $\sigma\in\Sigma^\omega$上的一行是可接受的运行，当且仅当 inf（r）∈F。即运行 r 是可接受的，当且仅当沿 r 的状态集合，危险次重复出现的状态是 F 中的某个元素。A 定义语言的方式与 Büchi 自动机的定义方式相同。Muller 自动机接受的语言类同 Büchi 自动机接受的语言类相同，等于由确定的 Muller 自动机接受的语言类。

例 1.2：如图 1.2 所示，确定的 Muller 自动机接受语言 L_0。由{a,b}上的仅含有限个 b 的字组成。其接受语言簇为$\{\{s_1\}\}$。因此，每一可接受运行仅可访问状态 s_0 有限次。

因此，确定的 Muller 自动机示有效的ω-正则语言表达工具。因为（1）其与所对应的非确定部分的表达能力相同；（2）可在多项式时间内求补。存在算法构造 Muller 自动机的交，检查语言包含问题。

1.2 时间转换表

时间性质的模拟，从不同的角度出发有不同的描述，但是一般来说有以下三种。第一种是离散的时间模型(discrete-time model)。其要求时间序列是单调递增的整数。

其适用于某些同步数字回路，当时钟符号到来时，其符号值已经改变。该模型的优点之一是能够比较容易的转换为一般正则语言。可以使用有限自动机处理离散的时间行为。当然，在物理过程中事件并不总是在整型值的时间内发生。因此离散的时间模型通过选择固定的先验，粗略估计连续时间。于是，就限制了所要模拟物理系统的精确性。第二种是虚拟时钟模型(fictitious-clock model)。除了要求整型序列是非递减的，它与离散时间模型相似。使用该模型分析问题时其时间执行序列的意义是，事件依据一定的顺序在实值时间点上发生。但对应于一数字时钟，仅有实际时间的整型读数才可以记录于序列。该模型同样可以较容易的转换为正则语言。同样可以使用有限自动机处理离散的时间行为。其缺点是只能近似的描述时间。我们运用第三种密集时间模型(dense-time model)，因为对物理过程在时间上的操作而言，密集时间模式是一个更为自然的模式。其时间是一个密集集合。使用该模式，事件的时间是单调递增的无界实数。但是，处理有限自动机框架内的密集时间比以上两种时间模式要复杂的多。因为这牵涉到一个向一般形式语言转换的问题。

为此，时间自动机提供了一套简单有效的方法。使用有限个变量来表示时间，称为时间变量。同时用一个条件来注释状态转换图，由于这个与时间有关的条件决定了状态的转换发生的时机，因此称之为时间限制。对于某个字符串，每个字符对应一个实型的时间标志我们称之为时间字。对某次转换，时间自动机可能检查其时钟值，并对一些时钟符以新值，那么时间自动机就可以接受时间字。自动机理论可以对实时系统用有限控制解决一些验证问题。

通过在一个时间字内对每一个符号与实型时间值相比较，来定义时间字，从而扩展ω-自动机的定义，使其接受时间字。从而得出类似于ω-正则语言的理论——时间正则语言类。

时间语言

定义时间字后就可使实时系统的时间对应于事件字母表上的时间字。和密集时间模型一样，使用非负实数R作为时间域，字σ上的时间序列定义如下

定义：时间序列 $\tau=\tau_1\tau_2\cdots$ 是时间值的无限序列，$\tau_i\in R$ 且 $\tau_i>0$，满足以下条件：

(1)单调性：τ单调递增，即 $\tau_i<\tau_{i+1}$, $i\geq 1$；

(2)递增性：对 $t\in R$，存在某个 $i\geq 1$. 使得 $\tau_i>t$。

字母表Σ上的时间字是一序偶（σ，τ），$\sigma=\sigma_1\sigma_2\cdots$ 是Σ上的无限字，而Σ上时间语言是Σ上的时间字集合。如果时间字看作自动机的输入，其代表在时间 τ_i 时，符号为 σ_i。若把每一符号 σ_i 解释为某一事件的发生，那么相应的元素 τ_i 解释为 σ_i 的发生时间。在某些环境下，某一序列中可允许同一时间值与多个连续事件相联系。处理此情况的定义仅有稍许不同，但是在此模型中，结论依然成立。

例 1.3：字母表 $\Sigma=\{a, b\}$，定义时间语言 L_1，L_1 是由时间字（σ，τ）组成，使得其在时间 5.6 后无 b 出现。由此可得语言

$$L_1=\{(\sigma,\ \tau)\mid \forall\ i\ (\tau_i>5.6)\ \rightarrow(\sigma_i=b)\}$$

例 1.4：时间语言的字母表Σ={a, b}，其中 a, b 交替出现，且 a, b 之间的时间间隔持续增加。可得语言

$$L_2=\{((ab)^{\omega},\ \tau)\mid \forall\ i\ (\tau_{2i}-\tau_{2i-1})<(\tau_{2i+2}-\tau_{2i+1})\}$$

时间语言的交、并、补操作同普通语言的操作相同。另外定义了与符号相联系的忽略时间值的非时间操作。即仅考虑时间字序偶到其第一个元素集的映射。

定义：对字母表Σ上的时间语言 L，Utime(L)是一个ω-语言，其由σ（$\sigma\in\Sigma^{\omega}$）组成，使得对某些时间序列τ，（σ，τ）∈L。

例如对例 1.3 Utime(L_1)是ω-语言$(a+b)^*a^{\omega}$。对例 1.4Utime(L_2)仅由字（ab）$^{\omega}$组成。

1.2.1 具有时间约束的转换表。

现在就可将转换表扩展为时间转换表，使得它读入时间字。当自动机状态转换时，下一状态的选择依赖于输入符。而在时间转换表，不仅依赖于输入符的时间，还依赖于上一输入符读入的时间。为此，与每一转换表相联系，设置一个实型的有限时钟集合。时钟能随任一转换复位。在任何情形下，时钟值总等于其上一次复位后的时间间隔。对每次转换设一时间约束，要求只有当前时钟值满足其约束时，转换才可能发生。如下例：

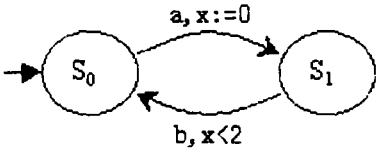


图 1.3 时间转换表的例子

例 1.5：时间转换表如图 1.3 所示，起始状态 s_0。仅有一个时钟 x 关于边有形如 x :=0 的注释，相当于当通过该边时时钟复位。关于边的形如（x<2）?的注释给出了与边相联系的时钟约束。自动机起始于状态 s_0，读入输入符 a，转为状态 s_1。此时时钟 x 赋 0。当处于状态 s_1时，时钟 x 表示从上一符号 a 发生后，时间的流逝值。当且仅当该值小于 2 时，从 s_1到 s_0的转换才可发生。当自动机返回状态 s_0时，循环重新开始。该转换表所表达的时间约束是 a 与其后跟随的 b 的延迟总小于 2，更形式的

$$\{((ab)^{\omega},\ \tau)\mid \forall\ i\ (\tau_{2i}<\tau_{2i-1}+2)\}$$

因此，为约束两个转换 e_1, e_2之间的延迟，要求时钟在 e_1复位，且与 e_2相联系。

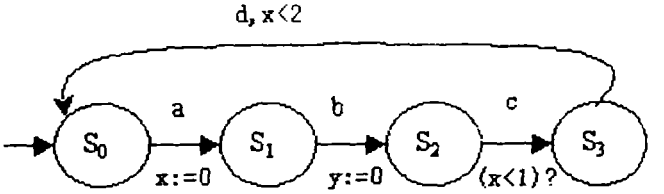


图 1.4 具有两个时钟的时间转换表

注意，时钟可以并行设置。这就意味着，不同的时钟可在不同的时间间隔内重新开始，并且在不同的读入时间间隔内无下界。

例 1.6：如图 1.4 使用两个时钟变量 x,y 的时间转换表所接受的语言

$$L_3=\{((abcd)^\omega,\ \tau)\mid \forall\ j\ (\tau_{4j+3}<\tau_{4j+1}+1)\wedge(\tau_{4j+34}>\tau_{4j+2}+2)\}$$

自动机在状态 s_0, s_1, s_2, s_3 之间循环。当输入 a，从 s_0 转换到 s_1 时时钟 x 赋 0。与从 s_2 到 s_3 以 c 标志的转换相联系，检查(x<1)?，确使 c 在前一个 a 发生的 1 个时间内发生。与此相似，独立时钟 y，当读入 b 时复位。读 d 时检查其值，确使 b 和其后跟随的 d 之间的延迟大于 2。注意例 1.6，为约束 a，c 与 b，d 之间的延迟，在 a 与其紧随的 b 之间或 c 与其紧随的 d 之间的时间差并没有给出明确的界限。这是多时钟的优势所在，各时钟可以彼此独立的设置。语言 L_3 可以看作如下所定义的语言 L_3^1 与 L_3^2 的交。

$$L_3^1=\{((abcd)^\omega,\ \tau)\mid \forall\ j\ (\tau_{4j+3}<\tau_{4j+1}+1)\}$$

$$L_3^2=\{((abcd)^\omega,\ \tau)\mid \forall\ j\ (\tau_{4j+34}>\tau_{4j+2}+2)\}$$

L_3^1 和 L_3^2 的每个语言均可以使用仅含一个时钟的自动机来表达，但是其交集必须用两个时钟。需要指出的是在分步系统中，所有自动机的时钟都相应于一个固定的全局时间，以同一固定的频率计算时间。各个时钟是为了表达系统的时间属性而虚设的。换句话说，可以认为自动机装配了若干个秒表，它们可以彼此独立的起始、检测，但是都指向同一时钟。

1.2.2 时钟约束与时钟解释

为形式定义时间自动机，需要判明在边中允许哪种类型的时钟约束。约束的最简单的形式是以时间常量与时钟值相比较，仅允许出现此类简单约束的布尔连接。非负有理数域内的任一值均可作为时间常量，当然，允许更复杂的约束。

定义：X 是时钟变量集，时钟约束 δ 的集合 Φ（X）定义如下：

$$\delta ::= x\leqslant c \mid c\leqslant x \mid \neg\ \delta \mid \delta_1\wedge\delta_2$$

$x\in X$，c 是有理数常量。

已知时钟集 X 的时钟解释 υ，表示对每一个时钟赋一个实值可以表示从 X 到 R 的映射。$t\in R$，$\upsilon+t$ 表示对 x 赋值 $\upsilon(x)+t$，$\upsilon\cdot t$ 表示对 x 赋值 $\upsilon(x)\cdot t$，对 $Y\subseteq X$，$[Y\rightarrow t]\ v$ 表示 X 的时钟解释，对任意的 $x\in Y$，赋值 t，而其余的时钟满足 v。

1.2.3 时间自动机

以下是时间转换表的精确定义。

定义：时间转换表A是一五元组$\langle \Sigma, S, S_0, C, E\rangle$

- Σ是有限字母表，
- S是有限状态集，
- $S_0 \in S$是起始状态，
- C是有限时钟集，
- 边集$E \subseteq S\times S\times \Sigma \times 2^C \times \Phi(C)$给出转换集。边$\langle s, s', a, \lambda, \delta\rangle$代表输入符号为a时从状态s到状态$s'$的转换。集合$\lambda \subseteq C$代表复位时钟，$\delta$是C上的时间约束。

给定一时间字（σ，τ）时间转换表在时间0起始于某一个起始状态，所有的时钟初始化为0。且都反应着时间的流逝。A在时间τ_i读入输入符号σ_i，如果当前时钟值满足δ_i则从s变为s'。更形式的表示为$\langle s, s', \sigma_i, \lambda, \delta_i\rangle$。$\lambda$内的时钟在此转换后被复位，所以它的元素所记录的是相对于此次转换后的时间。该行为被时间转换的运行所定义。运行记录了在各处转换点上的状态和所有的时钟值。对时间序列$\tau = \tau_1\tau_2\cdots$定义$\tau_0=0$。

定义：时间转换表$\langle \Sigma, S, S_0, C, E\rangle$在时间字（$\sigma$，$\tau$）上的运行r是一个无限序列，表示为（$\bar{s}$，$\bar{v}$）

$$r: \langle s_0, v_0\rangle \xrightarrow[\tau_1]{\sigma_1} \langle s_1, v_1\rangle \xrightarrow[\tau_2]{\sigma_2} \langle s_2, v_2\rangle \xrightarrow[\tau_3]{\sigma_3} \cdots$$

其中$s_i \in S$，$v_i \in [C \to R]$。对所有的$i \geq 0$，满足以下要求：

（1）$s_0 \in S_0$，对任意的时钟变量x，$x \in C$，$v_0(x)=0$；

（2）对任意的$i \geq 1$，在E内存在一边，形如$\langle s_{i-1}, s_i, \sigma_i, \lambda_i, \delta_i\rangle$使得$(v_{i-1} + \tau_i - \tau_{i-1})$满足$\delta_i$，$v_i$等于$[\lambda_i \to 0](v_{i-1} + \tau_i - \tau_{i-1})$

在这里，使用inf(r)表示自动机的一个运行r无限重复部分，则该运行在时间字上是可接受的，当且仅当$inf(r) \in f$。

例1.7：对例1.5所示的时间转换表，考虑一个时间字

$$(a, 2) \to (b, 2.7) \to (c, 2.8) \to (d, 5)$$

下面给出运行的初始部分，时钟解释由表[x, y]代替

$$\langle s_0, [0,0]\rangle \xrightarrow[2]{a} \langle s_1, [0, 2]\rangle \xrightarrow[2.7]{b} \langle s_2, [0.7, 0]\rangle \xrightarrow[2.8]{c} \langle s_3, [0.8, 0.1]\rangle \xrightarrow[5]{d} \langle s_0, [3, 2.3]\rangle \cdots$$

设（σ，τ）上的运行r=（$\bar{s}$，$\bar{v}$），在时间t时，τ_i与τ_{i+1}之间的时钟值

由时钟解释（$\upsilon_i + t - \tau_i$）给出。当从状态 s_i 到 s_{i+1} 之间的转换发生时，使用（$\upsilon_i + \tau_{i+1} - \tau_i$）检查时钟约束，在时间 τ_{i+1}，某个时钟被复位。

转换表 A=<Σ，S，S_0，E>可以看作一个时间转换表 A′。A′的时钟集合为空，转换表的边〈s，s′，a〉转化为〈s，s′，a，∅，true〉。A′的运行显然与 A 的运行相对应。

1.3 时间 Büchi 自动机

现在就可以使用时间转换表结合接受条件来定义时间 Büchi 自动机。

定义：时间 Büchi 自动机（TBA）是元组<Σ，S，S_0，C，E，F>，其中<Σ，S，S_0，C，E>是时间转换表，F⊆S 是可接受状态集合。

时间 Büchi 自动机在时间字（σ，τ）上的运行 r=（$\bar{s}$，$\bar{\upsilon}$）是可接受的，当且仅当 inf（r）∩F≠∅。对时间 Büchi 自动机，其接受的语言集合 L(A)为

{（σ，τ）| 在（σ，τ）上 A 有可接受的状态}

有了时间 Büchi 自动机后就可以很方便的定义时间正则语言了。

定义：时间语言 L 是时间正则的，当且仅当存在某个时间 Büchi 自动机 A 使得 L=L(A).

例 1. 8：如例 1.6 所示的语言 L_3是正则的，其接受集合由所有状态组成。

对每一个在Σ上的ω-正则语言 L，时间语言{（σ，τ）| σ ∈L}是正则的。如例 1.4 所示，语言 L_2是典型的非正则语言。其要求相邻的 a，b 之间的时间差组成一个递增序列。另一个非正则语言是{（a^ω，τ）| ∀ i（$\tau_{2i}=2^i$）}。在下例中，自动机联合 Büchi 自动机接受条件与时间属性来表示一个有趣的收敛反应属性。

例 1.9：如图 1.5 所示,自动机接受字母表{a，b}上的时间语言 L_{crt}

$$L_{crt}=\{((ab)^\omega,\ \tau) \mid \exists i,\ \forall\ j\geqslant i,\ \ \tau_{2i}\leqslant\tau_{2i-1}+2\ \}$$

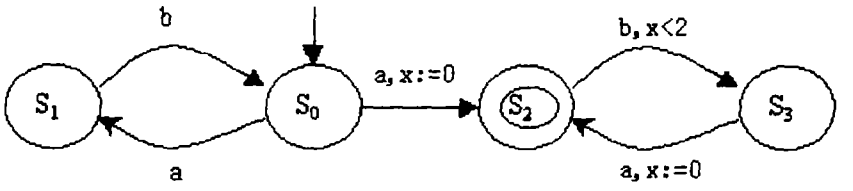


图 1.5 接受 L_{crt}的时间 Büchi 自动机

该自动机初始状态为 s0，接受状态为 s2，仅具有一个时钟 x，自动机起始于状态 s0，在 s0，s1 之间循环一会，然后非确定的移至状态 s2，x 设为 0。当在 s2，s3 之间循环时，读入 a 时钟复位，确保下个 b 在 2 个时间单位内到来。把符号 b 作为对符号 a 请求的响应，自动机就模拟了一个具有收敛性质的反应系统，反应总在

2 个单位时间内到达。下例形式，时间自动机同样可以表达周期形的行为。

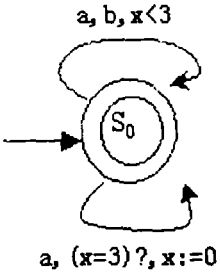


图 1.6 时间自动机表述先验行为

例 1.10：图 1.6 所示的自动机接受字母表{a，b}上语言

{（σ，τ）| 对任意的 i，存在 j，τ_j=3i 且 σ_j=a}

自动机仅有一个状态 s0，一个时钟 x，时钟 x 在 3 个时间单位的时间间隔内复位，要求在时钟值为 3 时，必须存在一个符号 a，其表达了在 3 的整数倍时间内 a 发生的时间属性。

1.4 时间 Muller 自动机

扩展了时间转换表就可以产生使用具有 Muller 接受条件的时间自动机，具有 Muller 接受条件的自动机的详细定义见如下

定义：时间 Muller 自动机（TMA）是一个元组<Σ，S，S_0，C，E，f>，<Σ，S，S_0，C，E>是一个时间转换表，f 属于 2^S 表述一接受簇。

在时间字(σ,τ)上的运行 r=（$\bar{s}$，$\bar{v}$）是可接受的当且仅当 inf（r）∈f。

对于时间 Muller 自动机 A 其接受的时间语言 L(A)定义为：{(σ，τ)| A 有(σ，τ)上的可接受运行}。

例 1.12：如图 1.7 所示，自动机的字母表为{a,b,c}，起始状态为 S_0。接受簇仅有一个元素{S_0,S_2}。所以可接受运行仅在 S_0,S_1 之间循环有限次，而在 S_0,S_2 之间无限次循环。自动机接受的字(σ，τ)满足（1）σ∈（a（b+c））*（ab）ω；（2）i≥1，若第 2i 个符号是 c，则 $\tau_{2i-1}-\tau_{2i-2}$ 小于 2，否则小于 5。

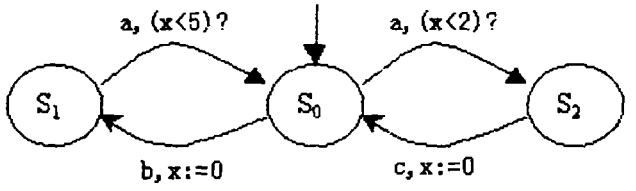


图 1.7 时间 Muller 自动机

非时间 Büchi 自动机与 Muller 自动机具有相同的表达能力，定理 1.1 表明，该结论对时间 Büchi 自动机与时间 Muller 自动机同样成立。由此可见时间 Muller 自动

机接受的语言类等同于时间正则语言。该定理同样表明ω-正则语言被 Büchi 自动机接受，当且仅当其被某个 Muller 自动机接受。

定理 1.1：时间语言被时间 Büchi 自动机接受，当且仅当其被某个时间 Muller 自动机接受。

证明：设时间 Büchi 自动机 $A=\langle \Sigma, S, S_0, C, E, F\rangle$，时间 Muller 自动机 A' 具有和 A 一样的时间转换表，其接受簇 $f=\{S' \subseteq S \mid S' \cap F \neq \varnothing\}$。显然 $L(A)=L(A')$。

给定一个时间 Muller 自动机 $A=\langle \Sigma, S, S_0, C, E, f\rangle$现在来构造一个时间 Büchi 自动机，使用 Büchi 自动机来模拟 Muller 自动机的接受条件，从而接受同样的语言。注意 $L(A)=\cup_{F\in f} L(A_F)$，$A_F'$ 接受语言 $L(A_F)$。设 $F=\{s_1, \cdots, s_k\}$，自动机 A_F' 使用使用非确定性来预测永远集合 F 的时机。用一个计数器来确定 F 被无限次访问。A_F' 的状态形如 (s, i)，$s\in S$，$i\in\{0, 1, \cdots, k\}$。初始集合为 $S_0\times\{0\}$。自动机模拟 A 的转换，在某个点上，非确定的将第二个元素设为 1，对 A 的每个转换$\langle s, s', \sigma, \lambda, \delta\rangle$，自动机 A_F' 有一个转换$\langle\langle s, 0\rangle, \langle s', 0\rangle, \sigma, \lambda, \delta\rangle$。另外，如果 $s'\in F$，则存在一个转换$\langle\langle s, 0\rangle, \langle s', 1\rangle, \sigma, \lambda, \delta\rangle$。

当第二个元素非 0，自动机要求停在集合 F 内。对 A 的转换$\langle s, s', a, \lambda, \delta\rangle$，s，s′ 都在 F 内 $(1\leqslant i\leqslant k)$，则存在 A_F' 的转换$\langle\langle s, i\rangle, \langle s', j\rangle, \sigma, \lambda, \delta\rangle$，如果 $s=s_i$ 则 $j=i+1 \bmod k$，否则 $j=i$。唯一的接受状态是$\langle s_k, k\rangle$。

第二章 区域自动机及新的构造算法

本文将实现自动机语言判空算法，然后对此改进，并提出新的处理方法。转换表存在一条可接受路径显然是自动机语言非空的必要条件。但是，自动机的时间约束排除了某些行为，由此可以构造一个 Büchi 自动机其非时间字集与被时间自动机接受的时间字是一致的。

限于整形常量

如前所述，时间自动机允许时钟约束与有理数常量比较，以下引理显示，为查空，需要时间自动机的时钟约束仅含整形常量。对时间序列 τ，$t\in Q$，使用 $t^*\tau$ 表示乘以时间序列的每一个 τ_i。

引理：对时间转换表 A，时间字（σ，τ），$t\in Q$，A_t 是对 A 的每一个含 d 的时钟约束以 t^*d 取代后所得的时间转换表。（$\bar{s}$，$\bar{\upsilon}$）是 A 在（σ，τ）的运行，当且仅当（$\bar{s}$，$t^*\bar{\upsilon}$）是 A_t 在（σ，τ^*d）上的运行。

该引理从定义中很容易得出。由此，在 A 与 A_t 之间的运行同构。如果 t 是出现在 A 时钟约束中常量分数部分，分母的最小公倍数，那么 A_t 的时钟约束部分仅含整数常量。在此转换内，每个常量的值最多以所有初始常量的积为增长量。对常量二进制编码，用 $|\delta(A)|$ 表示时钟约束的长度，易得 $|\delta(A_t)|$ 的上界为 $|\delta(A)|^2$。该结果依赖于对常数的二进制编码，一元编码似的 $|\delta(A_t)|$ 是 $|\delta(A)|$ 的几何级扩张。L(A) 为空当且仅当 $L(A_t)$ 为空。因此，可以使用 A_t 来判断 L(A) 是否为空。同样，Untime(L(A)) 等于 Untime($L(A_t)$)。因此，设时钟约束仅使用正型常量。

2.1 时钟区域

在时间的每一点上，时间转换表的约束行为决定于其状态和所有时钟的值。鉴于此给出以下定义

定义：对时间转换表$<\Sigma$，S，S_0，C，E>，扩展状态是一序偶<s，υ>，$s\in S$，υ 是 C 上的时钟解释。

由于此类扩展状态的数目是无限不可数的，不可能构造一个自动机其状态为扩展状态。但是具有相同状态的两个扩展状态，所有时钟整数部分相同，分数部分大小保持一定的次序，那么起始于二扩展状态的；两个运行非常相似。时钟值整数部分判断某个时钟约束是否满足，二分数部分判断哪个时钟最先改变其整数部分。如在扩展状态中，时钟 x,y 在 0，1 之间，那么时钟约束（x=1）的转换是否跟随（y=1）依赖于当前时钟值是否满足（x<y）。

时钟值的整数部分可以任意大，但是如果与时钟 x 比较的常量决不比 C 大，那么其实际值一旦超过 C 以后的路径将无相关序列。

在形式化该概念之前，先定义一个函数 fract：R➔ N。设 x 表示任意的自然数，则 fract(x)为对 x 求整。⌊t⌋表示 t 的整数部分，即 t=⌊t⌋ + fract(t)。设 C 内每一个时钟均出现在某个时钟约束中。

定义：时间转换表 A=<Σ，S，S_0，C，E>。对每一个 x∈C，设 c_x是出现在 E 中形如（x≤c）或（c≤x）的时间约束子式中最大的整数 c。

等价关系~定义于 C 上的所有时钟解释集：υ ~ υ′ 当且仅当以下成立：

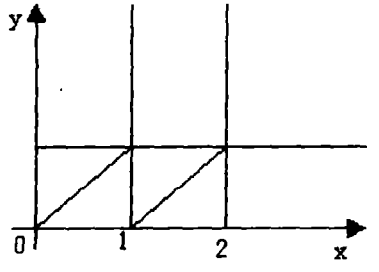
(1)对所有的 x∈C，或者⌊υ(x)⌋与⌊υ′(x)⌋相同或者二者都大于 c_x。

(2)对所有的 x，y∈C，且 υ(x) ≤c_x，υ(y) ≤c_y，fract(υ(x)) ≤fract(υ(y))当且仅当 fract(υ′(x)) ≤fract(υ′(y))成立。

(3)对任意 x∈C，υ(x) ≤c_x，fract(υ(x))=0 当且仅当 fract(υ′(x))=0。

A 的时钟区域是由~得到的时钟解释的等价类。使用[υ]来表示 υ 所属的等价类。每一区域可以唯一的由一有限时钟集合所满足的时钟约束所标志。例如，具有两个时钟的时钟解释 υ(x)=0.3，υ(y)=0.7。[υ]内的每个时钟解释满足约束（0<x<y<1）。用[0<x<y<1]取代此区域，通过下例对等价类的本质有更好的了解。

例 2. 1：考虑具有两个时钟 x，y 的时间转换表，C_x=2，C_y=1。时钟区域如图 2.1 所示：



6 个角点：例如[（0，1）]

14 个开放线区间：例如[0<x=y<1]

8 个开放区域：例如[0<x<y<1]

图 2.1 时钟区域

注意仅有有限个区域。对 A 的时钟约束 δ，若 υ ~ υ′，那么 υ 满足 δ 当且仅当 υ′ 满足 δ。时钟区域 α 满足时钟约束 δ 当且仅当对每个 υ ∈ α 满足 δ。每一个区域可以表示如下：

（1）对每个时钟 x，其时钟约束来自以下集合

{x=c|c=0，1，…，C_x}∪{c-1<x<c| c=1，…，C_x}∪{x> C_x}

(2) 每对时钟 x、y，使得 c、d 在（1）出现，使得 c-1<x<c，d-1<y<d，fact（x）小于、等于或是大于 fact（y）。

通过计算以上形式方程的可能联结数目，可得到上界。

引理：时钟区域的上界是[|C|！*$2^{|C|}$*∏x∈C[2 c_x+2]]。

由于|δ(A)|代表对 A 的时钟二进制编码后的时钟约束的长度，因此，可得乘积∏x∈C[2 c_x+2]的复杂度为 $O[2^{|\delta(A)|}]$。因为|C|的数目由|δ(A)|限制，从而使区域

数为 $O[2^{|\delta(A)|}]$。如果增加 $\delta(A)$，而不增加时钟数目或者与时钟相比较的最大常量，那么区域数并不随 $|\delta(A)|$ 增加。一个区域可以在 $|\delta(A)|$ 内以线形空间代替。

2.2 时间后继

查空的判定过程的第一步是构造一转换表，其路径以某种方式模拟 A 的运行。在这里，使用 A 的等价类区域自动机 R(A)，R(A)的状态记录时间转换表 A 的状态和当前时钟值的等价类。其形如$\langle s, \alpha\rangle$，$s\in S$，α 是时钟区域。当 A 的扩展状态是$\langle s, \upsilon\rangle$，则 R(A)的状态是$\langle s, [\upsilon]\rangle$。等价类区域自动机起始于某一状态$\langle s_0, [\upsilon_0]\rangle$，$s_0$ 是 A 的起始状态，而时钟解释 υ_0 则对每个时钟赋值 0。定义 R(A)的转换表，使其模拟所期望的转换。转换关系定义如下，有一个以 a 标记的从$\langle s, \alpha\rangle$到$\langle s', \alpha'\rangle$的边当仅当在状态 s 有一时钟值 $\upsilon\in\alpha$ 对某些 $\upsilon'\in\alpha'$ 能产生一个关于 a 的到扩展状态$\langle s', \alpha'\rangle$的转换。

使用时钟区域的时间后继关系可以较容易的转换边间的关系。时钟区域 α 的时间后继是随着时间流逝，可被一个时钟解释 $\upsilon\in\alpha$ 访问到的时钟区域。

定义：时钟区域 α' 是时钟区域 α 的时间后继当且仅当对每一个 $\upsilon\in\alpha$ 存在正数 $t\in R$ 使得 $\upsilon+t\in\alpha'$。

例 2.2：考虑上例，时钟区域 α 的时间后继在 α 内的某一点沿对角线方向某条线可达的区域，即平行于 线 x=y。例如，区域[(1<x<2)，(0<y<x-1)]的时间后继除了自身，还有[(x=2)，(0<y<1)]，[(x>2)，(0<y<1)]，[(x>2)，(y=1)]，[(x>2)，(y>1)]。

下面给出构造时间后继的方法。由上文可得时钟区域 α 由两个条件给出（1）对每个时钟 x，其约束形如 x=c，c-1<x<c 或是 $x>C_x$。（2）以对时钟变量出现在（1）内，使得 c-1<x<c，d-1<y<d，fract（x）与 fract（y）是可比较的。计算 α 的所有时间后继。显然，时间后继关系首先是一个传递关系。

首先，若对每一个时钟 x，α 满足约束 $x>C_x$，α 仅有一个时间后继，就是它本身。在例 2.1 中该情形所对应的区域是[(x>2)，(y>1)]。

其次，设非空集合 C_0 由时钟组成，任意的时钟 $x\in C_0$，使得对某个 $c\leqslant C_x$，α 满足约束 x=c。在此情形下，随着时间前进，x 的分数部分不再为 0，时钟区域立即改变。α 的时间后继同 β 相同，而 β 如下所示：

（1）对 $x\in C_0$，如果 α 满足（x= C_x）那么在 β 内（$x>C_x$），如果 α 满足（x= c）那么在 β 内（1<x<c+1）。$x\notin C_0$ 则 α、β 内关于 x 的时钟约束不变。

（2）时钟 x，y 使得 $x\leqslant C_x$，$y\leqslant C_y$ 在 α 内成立，它们 α、β 内的次序关系相同。

例如对例 2.1 中[(x=0)，(0<y<1)]的时间后继同[(0<x<y<1)]的时间后继相同。

最后，若以上都不成立，设时钟集合 C_0（$x\in C_0$）是 α 不满足 $x>C_x$ 的时钟的集合。

并且时钟具有最大的分数部分。即对任一个时钟 y，α 不满足 $y>C_y$。那么 fract(y) ≤fract(x)是α内的一个约束。在此情形下，随着时间前进，C_0内的时钟被赋整值。时钟区域β如下所述：

（1）对 $x\in C_0$，若α满足$(c-1<x<c)$，那么β满足 x=c。若 $x\notin C_0$，α、β内的约束相同；

（2）时钟 x, y 使得 $c-1<x<c$，$d-1<y<d$ 在（1）内出现，则它们在α、β内的分数部分关系相同。

此时α的时间后继包括α，β以及β的所有时间后继。例如对例 2.1[(0<x<y<1)]的时间后继包括其自身，[(0<x<1)，(y=1)]和[(0<x<1)，(y=1)]的所有时间后继。

2.3 等价类区域自动机

现在就可以定义等价类区域自动机了

定义：时间转换表 $A=<\Sigma, S, S_0, C, E>$，相应的等价类区域自动机 R(A)是Σ上的转换表

- R(A)的状态形如$<s, \alpha>$，$s\in S$，α是一时钟区域。
- 初始状态形如$<s_0, [\upsilon_0]>$，$s_0\in S_0$，对所有 $x\in C$，$\upsilon_0(x)=0$。
- R(A)有边$<<s, \alpha>, <s', \alpha'>, a>$当且仅当存在$<s, s', a, \lambda, \delta>\in E$和一时钟区域$\alpha''$，使得(1)时钟区域$\alpha''$是α的时间后继，(2) α'' 满足δ，(3) $\alpha'=[\lambda\mid\rightarrow 0]\ \alpha''$

Alur，和 Dill 给出了求等价类区域自动机的算法描述。构造等价类区域自动机的重点及难点在于求区域时钟（region）的时间后继(time_successor)。对某个区域时钟α，求其时间后继，设时钟 x 为其任一时钟，$x=\lfloor x\rfloor+fract(x)$将α分为三类

1. $x> c_x$；
2. $x<c_x$ 且 fract(x)=0；
3. $x<c_x$ 且 fract(x)≠0；

例 2.3：例如对图 2.2，$f=\{s_1, s_3\}$我们求其等价类区域自动机 A_0，显然其初始状态为$<s_0, [x=y=0]>$。因为[x=y=0]的时间后继除其自己外还有[0<x=y<1]，[x=y=1]，[x=y>1]，又根据等价类区域自动机所对应的转换表$<s_0, s_1, a, \{y=0\}, ->$，得下一状态为$<s_1, [0=y<x<1]>$，$<s_1, [x=1, y=0]>$，和$<s_1, [x>1, y=0]>$。我们求 [0=y<x<1]的时间后继，可得[0=y<x<1]，[0<y<x<1]，[0<y<x=1]，[0<y<1, x>1]，[y=1, x>1]，[y>1, x>1]，对应的转换表$<s_1, s_2, b, \{\}, \{y=1\}>$，其后继状态为$<s_2, [y=1, x>1]>$。依次类推，使用该算法我们可以得到以下等价类区域自动机。

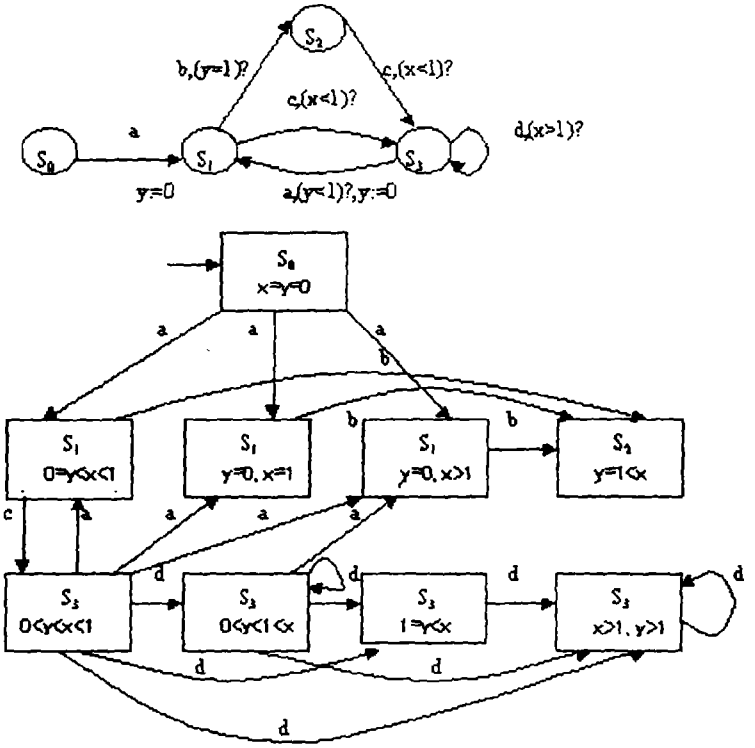


图 2.2 自动机及其等价类区域自动机

现在继续建立 A 与 R(A)之间对应的运行。

定义：对 A 的运行 r= （$\tilde{s}$ ， $\tilde{v}$ ）形如

$$r: \langle s_0, v_0\rangle \xrightarrow[\tau_1]{\sigma_1} \langle s_1, v_1\rangle \xrightarrow[\tau_2]{\sigma_2} \langle s_2, v_2\rangle \xrightarrow[\tau_3]{\sigma_3} \cdots$$

其投影[r]= （$\tilde{s}$ ，[$\tilde{v}$]）是序列

$$r: \langle s_0, [v_0]\rangle \xrightarrow{\sigma_1} \langle s_1, [v_1]\rangle \xrightarrow{\sigma_2} \langle s_2, [v_2]\rangle \xrightarrow{\sigma_3} \cdots$$

从 R(A)的边关系定义可以看出，[r]是 R(A)在 σ 的运行。由于时间沿 r 无限增长，任意的 x∈C 或是无限增长，或是无限次复位。因此，x∈C 对无限多个 i≥0， v_i 满足[（x=0）∨（x>c,）]。可得以下定义。

定义：等价类区域自动机 R(A)的运行 r 形如

$$r: \langle s_0, a_0\rangle \xrightarrow{\sigma_1} \langle s_1, a_1\rangle \xrightarrow{\sigma_2} \langle s_2, a_2\rangle \xrightarrow{\sigma_3} \cdots$$

是递增的当且仅当对每个 $x\in C$，有无限多个 $i\geq 0$，使得 α_i 满足[（x=0）∨（x>c_x）]。

因此对 A 在（σ，τ）上的运行 r，[r]是 R(A)在σ上的递增运行。以下引理隐含着 R(A)的运行对应于 A 的投影。在给出引理之前，先看一个例子。

例 2.4：对例 2.1 所示的等价类区域自动机 $R(A_0)$的每个运行的前缀具有以下三种形式之一：（1）自动机在区域<s_1,[0=y<x<1]>与<s_3,[0<y<x<1]>之间循环；（2）自动机在区域<s_3,[0<y<x<1]>上自循环；（3）自动机呆在区域内<s_3,[x>1，y>1]>。

只有（3）对应着递增运行。对类型 1，尽管 y 无限次复位，x 的值总小于 1。对类型 2，运行中尽管 x 的值无界，但时钟 y 有限次复位，其值有界。因此，A_0的每个递增运行对应于 $R(A_0)$的第 3 类运行。

引理：R(A)在σ上的运行 r 是递增的，那么存在一时间序列τ和 A 在（σ，τ）上的运行 r′ 使得 r 等于[r′]。

证明：对 R(A)在σ上的运行 $r=(\bar{s},\bar{\alpha})$，构造运行 r′ 和时间序列τ。如同以前，r′ 起始于<s_0, υ_0>。设 $\upsilon_i\in\alpha_i$，在时间 τ_i，A 的扩展状态是<s_i, υ_i>。R(A)存在以 σ_{i+1}标志的，从<s_i, α_i>到<s_{i+1}, α_{i+1}>的边。从等价类区域自动机的定义可得存在边<s_i, s_{i+1}, σ_{i+1}, λ_{i+1}, δ_{i+1}>，其属于 E，和 α_i的时间后继 α_{i+1}'，使得 $\alpha_{i+1}=[\lambda_{i+1}\rightarrow 0]\ \alpha_{i+1}'$。从时间后继的定义可得：存在时间 τ_{i+1} 使得 $(\upsilon_i+\tau_{i+1}-\tau_i)\in\alpha_{i+1}'$。显然，$\upsilon_{i+1}\in\alpha_{i+1}$，转换到 A 的下一扩展状态<$s_{i+1}$, υ_{i+1}>发生的时间是 τ_{i+1}。

重复使用这种构造方式，可得在（σ，τ）上的运行 r′，$r'=(\bar{s},\bar{\upsilon})$。[r′]=r。

此构造方式存在的唯一问题是τ可能不满足递增条件。设τ是收敛序列，因为 r 是递增运行，可以构造另一时间序列τ′ 满足递增要求。显然，自动机在时间 τ_i' 可以遵循同 r′ 一样的转换序列。

设 C_0是沿 r 无限次重复的时钟集合。由于τ是收敛的，向前某一位置后，C_0的每一时钟在到 1 之前复位。由于 r 的递增性，不在 C_0内的任一时钟，在某一位置后决不复位。并且满足 x>c_x，因此必定存在 $j\geq 0$，使得(1)第 j 个转换后每一时钟 $x\notin C_0$。一直满足 x>c_x，且 $x\in C_0$满足 x<1；（2）对每个 k>j，$\tau_k-\tau_j<0.5$。

设 $j<k_1<k_2<$ …是无限整数序列，使得 C_0内的任意时钟 x 沿 r 在第 k_i和 k_{i+1}转换之间至少复位一次。构造另一运行 $r''=(\bar{s},\bar{\upsilon}')$，时间转换序列τ′ 表示如下：沿 r′ 方向，r′′ 转换相同。若 $i\in\{k_1, k_2, \cdots\}$，那么要求 i+1 个转换在延迟（$\tau_{i+1}-\tau_i$）后发生，否则要求为 0.5。注意沿 r′′ 在第 k_i和 k_{i+1}转换之间的延迟小于 1。相应地，不管其他延迟，在第 j 个转换之后，C_0的时钟值总小于 1。所以在转换点，所有时钟约束的真值和时钟区域保持不变。由此可得 r′′ 是 A 的运行，[r′′]=[r′]=r。

由于τ无限次跳过间隔 0.5，满足递增要求。所以 r′′ 为引理所求的运行。□

2.4 非时间构造

对时间自动机 A 其等价类区域自动机可以用来识别 Untime[L(A)]。定理 2.1 是针对时间 Büchi 自动机而言的，但它对时间 Muller 自动机同样成立。

定理 2.1：给定一个时间 Büchi 自动机 $A=<\Sigma, S, S_0, C, E, F>$。存在一个在 Σ 上的 Büchi 自动机接受 Untime[L(A)]。

证明：构造 Büchi 自动机 A'，其转换表是 R(A)，等价类区域自动机对应的转换表为 $<\Sigma, S, S_0, C, E>$，A' 的可接受集合为 $F'=\{<s, \alpha>|s \in F\}$。

如果对 A 在 (σ, τ) 上的可接受运行 r，那么[r]是递增的且是 A' 在 σ 上的可接受运行。反之应用上述引理，给定 A' 在 σ 上的递增运行 r，引理给出了时间序列 τ 和 A 在 (σ, τ) 上的运行 r'，使得 r 等于 $[r']$。r 可接受，则 r' 也可接受。可得 $\sigma \in$ Untime[L(A)]当且仅当 A' 具有递增可接受运行。

对 $x \in C$ 设 $F_x=\{<s, \alpha>|\alpha \Rightarrow [(x=0) \vee (x>c_x)]\}$。因为 A' 的运行是可接受的当且仅当来自 F_x 的状态无限次重复。可直接构造另一个 Büchi 自动机 A''，使得 A' 具有在 σ 上的可接受运行当且仅当 A'' 在 σ 上有可接受运行。

A'' 是所求自动机，$L(A'')=$ Untime[L(A)]。□

例 2.5：对例 2.1 所示的等价类区域自动机 $R(A_0)$。由于 A_0 的所有状态都是可接受的。从递增运行的描述可以得到 $R(A_0)$ 的转换表，其可转变为 Büchi 自动机，仅仅需要将其接受状态转变为 $\{<s_3, [x>1, y>1]>\}$，于是

$$\text{Untime}[L(A_0)]=L[R(A_0)]=ac(ac)^*d^\omega$$

上一定理表明时间自动机内的时间信息是符号正则的。其一致性是由有限状态的自动机来检查。定理 2.1 的等价形式是：若时间语言 L 是时间正则的，那么 Untime(L) 是 ω 正则的。

进而，为了检查给定的语言时间 Büchi 自动机是否为空，用以上定理的证明方法构造一 Büchi 自动机，检查其相应的语言是否为空。

定理 2.2：给定一个时间 Büchi 自动机 $A=<\Sigma, S, S_0, C, E, F>$，L(A)是否为空可以时间 $O[(|E|+|S|)\cdot 2^{|\delta(A)|}]$ 内得到检验。

证明：设 Büchi 自动机 A' 是由定理 2.1 的证明方法构造而来。由于 A' 的状态数为 $O[|S|\cdot 2^{|\delta(A)|}]$，边数为 $O[|E|\cdot 2^{|\delta(A)|}]$。

语言 L(A)是非空的当且仅当在 A' 内存在循环 C，使得从 A' 的起始状态出发 C 是可以进入的。C 至少有一个 F' 的状态且包含 F_x 内的每一个状态。检查可以在 A' 的线形时间内完成。由此可得其复杂性为 $O[(|E|+|S|)\cdot 2^{|\delta(A)|}]$。□

若一自动机 A 的时钟约束包括引理常量，需要使用所有有理数的最小公倍数 t 构造 A_t。这是一个时钟约束数扩充。$\delta[A_t]=O[\delta(A)^2]$。

即使改变时间自动机的接受条件，以上方法仍然可以应用。特别是给定时间 Muller 自动机 A。可有效的构造出 Muller（或 Büchi）自动机接受 Untime[L(A)]。

就可以应用其对 L(A)判空。

等价类区域自动机的顶点是 $0[|A_S|*\Pi|A_i|*2^{|\delta(AS)|-\Sigma|\delta(Ai)|}]$，爆炸式的增长来源与三个因素：

（1）复杂性与实现[||P_i]的全局时间自动机的状态数成比例，是元素数目的扩展；

（2）任意时钟 x，与其比较的最大常量为 c_x，复杂性与常量 c_x 的乘积成比例；

（3）复杂性与所有时钟集上的后选方式成比例。

第一个原因是验证问题即使对非时间情形，也是最简单的。由于元素数一般很大，在实现模型检查时，该因素一直是主要障碍。第二个在推导它的形式化中是典型的，即使对简单离散模型，实际常量的扩展也是很明显的。若不同元素的延迟界限一般不同，该原因将是复杂性扩张的主要原因。最后，非时间构造中需要计算不同分数部分所有可能的大小关系。当转向一简单的模型，比如离散时间模型，模型将避免复杂度爆炸性的增长。但是由于时钟数与各自动机数成线形关系，其与第一个因素相同。

综合上文，可以看出等价类区域自动机是进行算法验证的关键。一个事件序列与延迟约束时间一致，当且仅当与事件相联系的递增实型时间值，使得每一延迟的间隔在允许的间隔内。为此，需要使用等价类区域自动机排除时间不一致序列。但是由于其几何级的时间复杂度，使得其应用受到了极大的限制。

对等价类区域自动机，文章仅给出了理论上的说明，并没有实现它。同时，从其分析可以看出其空间复杂度实现为 $0[|A_S|*\Pi|A_i|*2^{|\delta(AS)|-\Sigma|\delta(Ai)|}]$，是一个爆炸性的增长。当然，从以上分析可知，这主要由三个因素所引起的。本章主要讨论对等价类区域自动机的改进，期望找到较好的方法，来简化其空间复杂度。不失一般性，在以下对等价类区域自动机的讨论过程中，我们仅对单个自动机的构造进行考虑。

2.5 等价类区域自动机算法的初步实现

等价类区域自动机的算法构造主要在于时间后继的构造，决定等价类区域自动机空间爆炸式增长的三个因素，有两个在于时间后继的构造。因此，时间后继的分析是解决算法性能的关键。为了分析其算法实现，并提出相应的改进方法，在这里先根据其等价类关系给出算法实现。设初始时，某一个时钟区域 α，求其后继。

算法 1：

step 1.

$\beta=\alpha$，$T=T\cup\{\beta\}$

step 2.

设 x 为 α 内任一时钟，若 $x>c_x$ 则算法终止，否则转 3

step 3.

若存在不为空的集合 C_0 所包含的任一时钟 x，x 在 α 内满足 $x=c$ 且 fract(x)=0，

则在β内

①若 $x = c_x$ 则在β内用 $x > c_x$ 取代 $x = c_x$，否则用 $c < x < c + 1$ 取代，对于不属于 C_0 的 x，α，β时钟约束相同。

②对时钟 x, y 使得 $x < c_x, y < c_y$，在α中为真，fract(x), fract(y)的关系α，β内相同。

$\alpha = \alpha - C_0 - \{x, y\}$转 1。

若不存在该集合则转 4

step 4.

设任一属于时钟集合 C_0 的时钟 x，α满足 $x < c_x$ 且其具有最大的分数部分，即对所有的时钟 y，α不满足 $y > c_y$，fract(y)≤fract(x)是α的约束。在此情形下一旦时间前进,集合 C_0 内的时钟即赋整值。时钟区域β如下

① $x \in C_0$ 如 α 满足 $c - 1 < x < c + 1$ 则β满足 $x = c$。若 $x \notin C_0$，α，β内约束相同。

② 时钟 x, y 使得 $c - 1 < x < c + 1, d - 1 < y < d + 1$ 出现于①中 fract(x), fract(y) 的关系α，β内相同。

$\alpha = \alpha - C_0$ 转 1。

例 2.3 就是沿袭该算法求出时间后继的的例子。

2.6 等价类区域自动机算法的改进

在算法构造过程中，时钟区域α的时间后继对任一时钟 x，其构造都在区间 $0 \leqslant x < c_x$。为改进以上算法，首先引入一个概念：

定义：（区域可达）<s，α>是等价类区域自动机 R(A)的一个结点，时钟区域α的时间后继集合为 T=TS(α)。β∈T，若对某一可接受运行存在边<< s，α>，< s′，β>，a>则对α而言，β是区域可达的。

初始状态显然可达。

由于算法 1 所求结果并非都是可达的或是有意义的。例如对例 16，时钟区域<S_2，1=x<y>是不可达的。因此，没有必要再对其以及它的子孙求时钟区域。由此，即使某一个时钟区域本身是可达的，但是如果其子孙都是不可达的则其本身也可以看作不可达的。可以对所得结果进一步化简，通过剪枝除去不可达或是使时间约束无意义的时间后继，就可以从平均复杂度方面对算法进行化简。可得以下结论：

定理 2.3：设时钟区域α，及其时间后继β，若α使得时间约束δ为真，但β使得时间约束δ为假，则β的时间后继必然使得时间约束δ为假。

证明：设 x 为δ内某个时钟变量，则关于 x 的时间约束形如 x=c 或 x>c 或 x<c。我们对这三种情况分别证明。

① 当关于 x 的时间约束形如 x=c 时，因为α使得δ为真，所以在α内 x 的时间约束表示为 x=c，则对任意的β，$\beta \neq \alpha$，x 的时间约束表示为 $x = c \pm t$（$t \neq 0$）。

显然，β及其时间后继使得δ为假。

② 当关于x的时间约束形如x<c时，因为α使得δ为真，所以在α内x的时间约束表示为x<c，而其时间后继β使得δ为假，因为β为α的时间后继，所以在β及β的时间后继内x的取值范围为[c, c_x+1)。可得β及其时间后继使得δ为假。

③ 当关于x的时间约束形如x>c时，因为α使得δ为真，所以其任意得时间后继β关于x的时间约束均属于（c，c_x+1)。因此必使得δ为真。显然定理2.3也是成立的。

综合①②③定理2.3显然成立。□

由此，可得如下定理2.4：

定理2.4：对等价类区域自动机的状态S：〈s，α〉，若α是该状态祖先的时钟区域的不可达时间后继，若状态S′是S祖先，S′与S之间仅存在一条通路，S′的子树均无回路，则对S′进行剪枝而不影响系统性能。

证明：假设存在某个状态S′，其子树状态集为P，S∈P。对其剪枝后影响了系统性能。则一定存在某个可接受运行r，设r所遍历的状态集为Q，则Q ∩ P≠∅。S属于某个子树，因为S的时钟区域是不可达的，所以从S的祖先到S的转换的约束必然为假。及对S剪枝不会影响系统性能。定理2.4成立。□

实际上，凡是某个到叶子结点的无回路子树或是回路中不包含接受状态，都可进行剪枝而不影响系统性能，因为inf(r)与其交集必然为空。但是，我们这里主要考虑的是在区域改造中进行处理，所以对此类情况不予考虑。

算法2：

step 1.

若β是区域可达的则T=T∪{β}，否则若β是α的唯一后继或β的后继都是不可达的则T=T-{α}，算法终止。

若T的大小不变则算法终止，否则α=β，转2

step 2.

设x为α内任一时钟，若x> c_x 则算法终止，否则转3

step 3.

若存在不为空的集合C_0所包含的任一时钟x，x在α内满足x=c且fract(x)=0，则在β内

①若x=c_x则在β内用x>c_x取代x=c_x，否则用c<x<c+1取代，对于不属于C_0的x，α，β时钟约束相同。

②对时钟x, y使得x<c_x, y<c_y，在α中为真，fract(x), fract(y)的关系α，β内相同。

α=α-C_0-{x，y}转1。

若不存在该集合则转4

step 4.

设任一属于时钟集合 C_0 的时钟 x，α 满足 $x < c_x$ 且其具有最大的分数部分，即对所有的时钟 y，α 不满足 $y > c_y$，fract(y)≤fract(x)是 α 的约束。在此情形下一旦时间前进,集合 C_0 内的时钟即赋整值。时钟区域 β 如下

③ $x \in C_0$ 如 α 满足 $c-1<x<c+1$ 则 β 满足 $x=c$。若 $x \notin C_0$，α，β 内约束相同。

④ 时钟 x，y 使得 $c-1<x<c+1$，$d-1<y<d+1$ 出现于①中 fract(x), fract(y) 的关系 α，β 内相同。

$\alpha = \alpha - C_0$ 转 1。

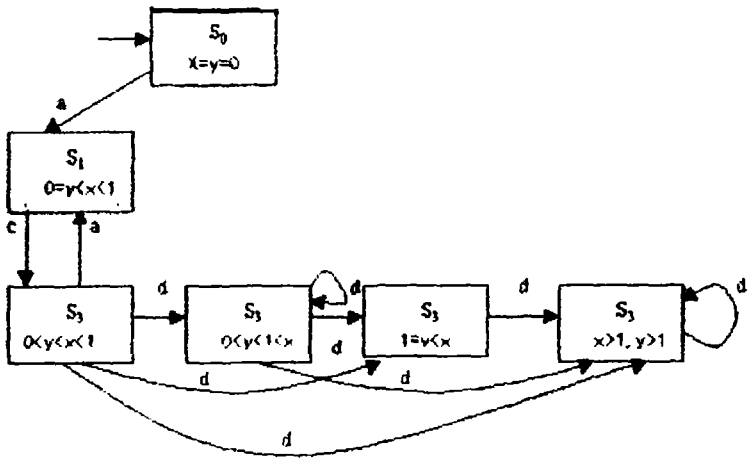


图 2.3 改进的构造算法构造出的等价类区域自动机

例 2.6：对例 2.3 所示的自动机使用该算法。显然，对任意运行 r，状态<S_2，α>∉inf（r）。所以〈S_2，α〉是不可达的，α 为任意区域。又因为〈S_2，α〉是〈S_1，[y=0，x=1]〉和〈S_1，[y=0，x>1]〉的唯一后继，所以〈S_1，[y=0，x=1]〉和〈S_1，[y=0，x>1]〉也是不可达的。可得等价类区域自动机如图 2.3。

2.7 一种新的区域自动机构造方法

2.7.1 位移值与位移空间

定义：关于时钟变量 x 的两个相邻约束 δ_1、δ_2，d_1、d_2 是 δ_1、δ_2 内使得关于 x 的约束为真的两个时间值，$d=d_1-d_2$，则 d 称为 x 在 δ_1、δ_2 之间的一个位移。

例 2.7：如图 2.4 所示的时间自动机，对时钟 x，相邻状态转换 S_0 到 S_1 以及 S_1 到 S_3 之间 1.5，2，5 都是相对于时钟 x 的位移。

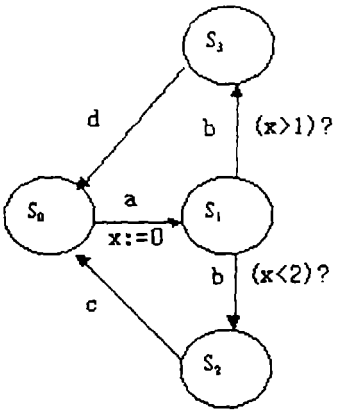


图 2.4　具有一个时钟的时间自动机

从定义可以看出，位移使得相邻的两个状态的转换发生成为可能，但是，转换发生往往不是仅仅发生于某一时刻，而很可能是发生于某一个时间范围内。例如对例 2.7，任何使得 x>2 的时钟值都可使得 S_1 到 S_3 的转换发生。因此，位移作为一个静态的概念，很难表示其动态变化，所以，引入位移空间。

定义 ：设 x 在二相邻约束 δ_1、δ_2 之间的任一位移为 d，集合 S=∪d，则 S 称为 x 在 δ_1、δ_2 之间的位移空间。

由此，对例 2.7，x>2 是使得的时钟值满足 S_1 到 S_3 的转换发生可能的位移空间。从以上定义可以看到，位移尽管使得某个时钟变量的约束为真，但是，并不一定使得转换发生成为可能。为了区别这二概念，我们引入下一概念。

2.7.2 相容、相容值与相容空间

在这里，使用~表示>，<，=，即两个数在数轴上的次序关系。

定义：时间转换表内，对于相邻约束 δ_1、δ_2，若 x~c∈ δ_1 可满足，且存在某个位移空间使得 x~c′ ∈ δ_2 也可满足，则在 δ_1、δ_2 内，关于 x 的时间约束相容。

由时间约束的定义可以得到，时间约束相容是个相容关系。显然，它满足自反性。不妨设相邻约束 δ_1、δ_2、δ_3，$\delta_1\delta_2$、$\delta_2\delta_3$ 之间的位移空间分别为 S_1、S_2，且都关于 x 的时间约束相容。

例 2.8：对图 2.4 所示的自动机对于循环 C=S_0 S_1 S_0，其关于 x 约束相容。使得相邻时钟约束相容的位移空间为 0≤x<5。

定义 ：设任意的时钟变量 x 及二相邻约束 δ_1、δ_2，x∈ δ_1，x+h 使得关于 x 的时间约束在 δ_2 内为真，则称 h 为时钟变量 x 在 δ_1、δ_2 间一个相容值。

可以看出，相容值是可以使转换发生的位移值。例如对图 2.4 所示的时间自动机，

由于只有一个时钟变量，其相容值等于位移值，但是，对含有多个时钟的时间自动机，相容值为位移值的交，即该值对所有的时钟而言，其约束都为真。

例 2.9：对图 2.2 所示的时间自动机，相邻转换 $S_0 \rightarrow S_1 \rightarrow S_3$。0，0.1，0.8，0.976...都是关于相邻约束的相容值。

与位移值相类似，对于相容值也存在一个时间范围问题。同样，为了描述这种动态行为，我们引入另一概念。

定义：(1)$H_0=\{0\}$是时间转换表中时钟变量的初始相容空间；

(2)设 x 是任意时钟变量，其相容值为 h_x，设 $H=\cup h_x$，则集合 H 是相容空间。

位移空间与相容空间显然尽管产生方式有些相似，尤其是当仅有一个时钟变量时，位移空间等于相容空间。但是二者存在本质上的区别。位移空间是使得某一个特定的时钟变量的约束为真的值的集合，但是，却可能使得另一时钟变量的约束不一定为真。而相容空间内的任意值均使得时钟变量的约束为真。凡是不为空的相容空间，必然与位移空间有交集，即相容空间是所有位移空间的子集，反之则不一定成立。对某一个转换，各时钟变量在相容空间内对时钟变量本身而言是同步的，即通过位移值及位移空间，实现了在全局时间的同步。

定义：对二时钟变量 x、y，以下操作称为对二时钟变量的裁齐，$a_1<x<a_2$，$b_1<y<b_2$，且存在 x<y，对实数 d，有 $\max(a_1,b_1)<d<\min(a_2,b_2)$，$\max(a_1,b_1)<x<d<\min(a_2,b_2)$，$\max(a_1,b_1)<d<y<\min(a_2,b_2)$。

实际上，时钟变量的裁齐是构造相容值的工具。当 x、y 取 d 时，必然满足对其的约束，同时，若 x、y 不可裁齐，必然不存在关于使得约束成立的时间增量。

定理 2.5：时间转换表中，若转换能够发生，则不为空的相容空间，必然与相对应位移空间有交集，即时钟变量必可裁齐。

证明：设相邻两个时间约束 δ_1、δ_2，与该转换对应的相容空间为 H_1、H_2，且 $H_1\neq\varnothing$，$H_2\neq\varnothing$。设时钟变量 x 在该转换下对应的位移空间为 D。若 $H_2\cap D=\varnothing$，由相容空间定义可知，对于 $h\in H_2$，使得 δ_2 为真，但是，不满足关于 x 的约束，显然矛盾，即 $H_2\cap D\neq\varnothing$，且由定义可得可以裁齐。□

以上工作，主要集中于相容空间的构造。我们所得相容空间使得时钟变量能在一个较大的范围内得到满足。我们还可以根据需要进一步细分相容空间，称之为分割相容空间。

定理 2.6：对于相容空间 H，$H'\subseteq H$，可以使用 H′，裁齐时间变量，将 H 分为三部分，H′，H″，H‴。若对任意值 $h'\in H'$，$h''\in H''$，$h'''\in H'''$，$h'\leqslant h''\leqslant h'''$均成立，则 H′，H″，H‴对 H 所处位置而言均是相容空间。称为 H′对相容空间的分割。

证明：由相容空间的定义可得，其对所有有关转换，都可使得其约束为真。又因为各个相容空间经过裁齐时间变量，所以各个时间变量的大小关系保持不变，可得所有集合仍然是所对应转换的相容空间。□

定义：时间转换表内，对于相邻约束 δ_1、δ_2，若 $x\sim c\in\delta_1$ 可满足，且存在某个相容空间使得 $x\sim c'\in\delta_2$ 也可满足，则在 δ_1、δ_2 内，关于 x 的时间约束相容。

根据以上定义，现在我们给出区域自动机的定义。

2.7.3 新的区域自动机构造方法

根据以上定义，现在我们对区域自动机重新定义定义。

定义：时间相容转换表 $A=<\Sigma, S, S_0, C, E>$，相应的区域自动机 R(A)是 Σ 上的转换表

- R(A)的状态形如<s，H>，$s\in S$，H 是一相容空间。
- 初始状态形如$<s_0, H_0>$，$s_0\in S_0$，对所有 $x\in C, x=0\in H_0$。
- R(A)有边<< s，H>，< s′，H′ >，a>当且仅当存在<s，s′，a，λ，δ >∈E,对于 h′ ∈H′ ,对任意的时钟变量 x，h′ 均使得 δ 为真。

由以上定义，我们得到一个定理 2.7

定理 2.7：对于区域自动机 R(A)，任意相邻二状态的时钟变量约束相容。

证明：区域自动机的任意状态形如<S，H>，其中 H 为相容空间，由相容空间及区域自动机的定义可得该定理 2.7 显然成立。□

2.7.4 新的区域自动机构造算法的实现

一． 位移值和位移空间的初步求法：

本文所提出的区域自动机的构造方法是基于相容空间的构造上的。而相容空间的构造是一个迭代过程。每一步的迭代一个相容值。要想得到相容值，必须先得到位移值。因此，在这里先给出位移值及位移空间的算法。

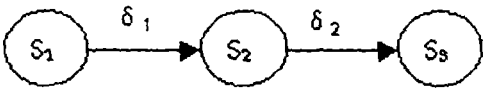


图 2.5 具有相邻约束的转换图

设一个转换如图 2.5，其时间约束为 δ_2。其相邻某一个（祖先）转换时间约束为 δ_1，时钟变量 $x\in\delta_1$、δ_2。可分以下情况求其位移空间(不失一般性，我们设 $c_1<c_2$)。

算法 3：

1．$x=c_1\in\delta_1$ 则

若 $x=c_2\in\delta_2$ 则位移空间为 D={d}（$d=c_2-c_1$）。

若 $x<c_2\in\delta_2$ 则位移空间为 D，若 $d\in D$ 则 $0<d<c_2-c_1$。

若 $x>c_2\in\delta_2$ 则位移空间为 D，若 $d\in D$ 则 $d>c_2-c_1$。

2．$x<c_1\in\delta_1$ 则

若 $x=c_2$ 或 $x<c_2\in\delta_2$ 则位移空间 D，若 $d\in D$ 则 $d<c_2$。

若 $x>c_2\in\delta_2$ 则位移空间为 D，若 $d\in D$ 则 $d>c_2$。

3．$x>c_1\in\delta_1$ 则

若 $x=c_2$ 或 $x<c_2\in\delta_2$ 则位移空间为 D，若 $d\in D$ 则 $0<d<c_2-c_1$。

若 $x>c_2\in\delta_2$ 则位移空间为 D，若 $d\in D$ 则 $d>c_2-c_1$

4.若 δ_1 内不含关于 x 的时钟约束，则沿逆边上溯，寻找第一个含 x 的时钟约束，若找到的不止一个，则裁齐后在根据 1，2，3 求出位移空间。

例 2.10：对图 2.2 所示的时间自动机状态转换 $S_1\rightarrow S_3\rightarrow S_1$ 对应的时钟约束分别为 δ_1：(x<1)，δ_2：[（y<1）?，（y：=0]），则对 y 求位移空间。因为，$S_1\rightarrow S_3$ 之间的约束没有关于 y 的时钟约束，所以前溯至状态 $S_0\rightarrow S_1$，和 $S_0\rightarrow S_1$。y 的位移空间为 0<y<1。

二．对特定的时钟变量 x 分割位移空间：

首先假设对时钟变量 y 的位移空间已经求出，现在要求根据 y 的位移空间求出 x 与 y 的相容空间。

算法 4：

假设某一个时钟变量 y 的位移空间 D 求出时钟变量 x，y 的相容空间 H。

取得所 x 对应转换的约束记为 δ

根据 H 及时钟变量 x 以及该转换源状态的相容空间求得集合 D′，若 D′ 为空则算法结束。

根据 D′ 求出该变量的位移空间记为 D″（$D''\subseteq D'$）。

三．新的区域自动机的构造方法

现在，就可以根据算法 3，4 构造新的区域自动机了。

算法 5：

step 1.

取得源状态的时间约束 δ_1

step 2.

目标状态的时间约束记为 δ_2

step 3.

对任一个时钟变量 x，从 δ_2 中取出其约束，根据算法 3 求其位移空间 D_1

step 4.

若 δ_2 为空，转 step 6，否则，对某一个时钟变量 y，从 δ_2 中取出其约束。

step 5.

根据算法 4，求得 y 的位移空间。转 step 4。

step 6.

对各时钟变量的相容空间，求其交，可得目标状态的相容空间。若目标状态已被访问过，则使用所得相容空间与原始相容空间的交集，分割原始相容空间。而所得相容空间与原始相容空间的差集，构造新的状态。算法结束。

例 2.11：再次考虑图 2.2 所示的时间自动机，使用该算法构造所得区域自动机如图 2.6 所示。

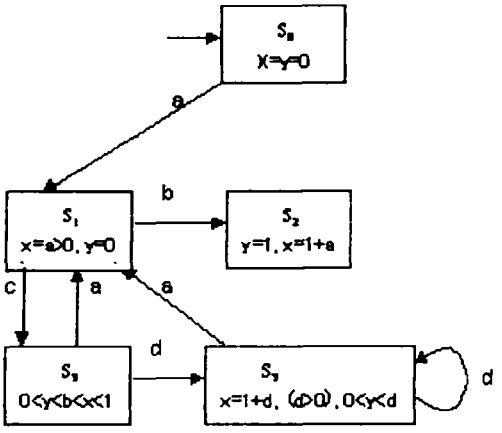


图 2.6 新的构造方法所得区域自动机

2.8 算法分析

由本章分析可以得出，等价类区域自动机几何级算法空间复杂度的产生由于三个因素。改进算法主要针对第一个原因。由于在构造过程中，某些不可达状态及其子孙的时钟区域不再计算，因此可以从平均复杂度方面改进算法性能。而对于新的构造方法，主要是针对因素二处理的。由于在区域构造过程中，考虑的是一个区间，在这个范围内，所有的时钟约束均可满足，因此，对第三个因素同时有影响。

对算法 1 进行分析。设 T 是 α 的时间后继集合，$x \in \alpha$，则可得在 T 内 x 所属区域至少为 $2\ c_x+2$。因此 α 的区域至多有 $\sum x \in C[2\ c_x+2]$ 时间后继。对时间约束进行二机制编码，可得等价类区域自动机 R(A) 的状态数为 $O[|S| \cdot 2^{|\delta(A)|}]$。自动机从 $\langle S, \alpha \rangle$ 出发也最多有一条边，由此可得等价类区域自动机 R(A) 的边数为 $O[|E| \cdot 2^{|\delta(A)|}]$。所以，区域图可以在时间 $O[(|E|+|S|) \cdot 2^{|\delta(A)|}]$ 内构造。但，对于使得 δ 为假的时钟区域的时间后继已经没有必要求出。事实上，对于时间自动机而言，状态转换与时间约束紧密联系。如果某一状态所对应区域本身是不可达的，那么通过剪枝该区域所产生的时间后继，及其所对应的状态就不必继续进行计算。

对于时间验证问题，Alur 认为使用区域自动机能够排除时间不一致序列，其优点我们这显而易见。模型的语言仅仅是组成元素自动机一起构成的区域自动机。但是，这种方法致命的弱点是其几何级的复杂度。对于所给出的算法改进，虽然对最坏时间复杂度而言并未有所改进，但是可以看出它对平均时间复杂度和空间复杂度都有较大的改进。

分析新的区域自动机构造算法，可以看出我们的计算工作主要集中在相容空间的计算过程中。而对所有时钟变量相容空间的计算时间花费为 O(|C|)。我们在区域自动机的构造过程中，对 Muler 自动机的每一个状态转换过程，都要计算每一个时钟变量的相容空间，而每一个状态转换都对应自动机的一条边。由算法 1，2，3，显然其时间复杂度为 O(|C||E|)。与经典等价类区域自动机的构造比较而言，我们的算法有了很大的进步。

第三章 其它类型的时间自动机

以上所描述的时间自动机，时间总是记录着事件的发生，确定事件发生发时机。针对实际工作中不同的工作环境，开发并应用着多种时间自动机。下面再介绍一种确定的时间自动机类。

3.1 确定时间自动机

时间自动机类的补操作不封闭，不可能自动比较两个自动机的语言。本文定义确定的时间自动机，显示确定的时间 Muller 自动机接受的语言类对所有的布尔操作封闭。

对非时间确定转换表而言仅有一个初始状态，从每一个状态出发，给定一输入符，下一状态唯一确定。对时间自动机的确定性需要一个标准：对某一扩展状态和一输入符号，沿时间进行方向，转换后下一状态唯一确定。其允许从某一状态出发有多个相同标志的转换，但是要求其时钟约束必须互斥，使得在任一时间仅有一个转换有效。

定义：时间转换表$<\Sigma, S, S_0, C, E>$是确定的当且仅当

（1）其仅有一个初始状态：$|S_0|=1$；

（2）设 $s\in S$，$a\in\Sigma$，对每一个形如$<s, —, a, —, \delta_1>$，$<s, —, a, —, \delta_2>$的时钟约束 δ_1，δ_2 互斥，即 $\delta_1\wedge\delta_2$ 永假。

时间自动机是确定的当且仅当其时间转换表是确定的。注意，不考虑时钟约束因素，以上确定性定义与转换表确定性定义匹配。因此，每个确定的转换表同样是确定的时间转换表。所以仅考虑确定时间 Muller 自动机（DTMS）。

例 3.1：确定时间 Muller 自动机如图 3.1 所示，接受语言 L_{crt}：

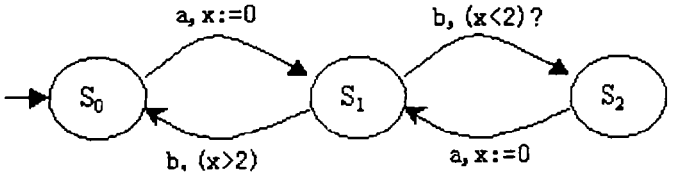


图 3.1 确定时间 Muller 自动机

$$L_{crt}=\{((ab)^{\omega}, \tau) \mid \exists_i, \forall j\geq i\ (\tau_{2j}<\tau_{2j-1}+2)\}$$

Muller 接受簇为$\{\{s_1, s_2\}\}$状态 s_1具有关于 b 的两个互斥输出。可接受条件要求具有时钟约束（$x>2$）的转换仅发生有限次。

由于确定自动机具有以下属性，所以易于求补。

引理：对给定的时间字，确定时间转换表至多有一个运行。

证明：对时间转换表 A 和时间字（σ,τ），运行起始于时间 0，具有扩展状态 $<s_0, \upsilon_0>$。s_0 是唯一的初始状态，设在时间 τ_{j-1} 上招展状态是$<s, \upsilon>$，构造到 j-1 步。由于 A 的确定性，在时间 τ_j 至多有一个转换$<s, s', \sigma_j, \delta, \lambda>$使得在时间 τ_j 的时钟解释 $\upsilon+\tau_j-\tau_{j-1}$ 满足 δ，若不存在此转换，则 A 无（σ,τ）上的运行，否则转换选择唯一的扩展运行到第 j 步，由此决定了在时间 τ_j 的招展状态。□

3.2 事件记录自动机

现在，约定时钟集合 C_Σ 与事件的关联方式。设任一 $a\in\Sigma$。使用 x_a 表示 a 的事件记录时钟。给定一个时间字 $w=(a_0,t_0)(a_1,t_1)\cdots$，w 的第 j 个位置的时钟 x_a 是 t_j-t_i，i 是 j 之前，$a_i=a$ 的最大 i。如果 j 之前事件 a 未发生，那么 x_a 未定义，表示为$\perp$。

重新定义时钟值取值范围为 $R_\perp=R^{\geq 0}\cup\{\perp\}$，即特殊位$\perp$与非负实数的并。现在定义一个时钟函数 $t_j^w(x_a)$，它表示在时间字 w 上，事件 a 的事件记录时钟在 j 点的时钟值即：

$$t_j^w(x_a)=\begin{cases} t_j-t_i & \exists i\ 0\leq i<j \wedge a_i=a \wedge (\forall\ k)\ i<k<j \rightarrow a_k\neq a \\ \perp & (\forall\ k)\ 0\leq k<j \rightarrow a_k\neq a \end{cases}$$

首先重定义时间转换表：

定义：时间转换表 A 是一五元组$<\Sigma, S, S_0, C_\Sigma, E>$

- Σ 是有限字母表；
- S 是有限状态集；
- $S_0\in S$ 是起始状态；
- C_Σ 是与事件相联系的有限时钟集；
- 边集 $E\subseteq S\times S\times\Sigma\times\Phi$ 给出了转换集。边$<s, s', a, \phi>$代表输入符号 a 时从状态 s 到状态 s′ 的转换。ϕ 是 C_Σ 上的时间约束。

现在，就可以使用重定义的时间转换表来定义此类自动机了。

定义：事件记录自动机（ECA）是元组$<\Sigma, S, S_0, C_\Sigma, E, F>$，其中$<\Sigma, S, S_0, C_\Sigma, E>$是其时间转换表，$C_\Sigma$ 内的任一时钟记录 Σ 内某事件发生后时间的流逝，$F\subseteq S$ 是可接受状态集合。

自动机时钟 x_a 当每次遇到事件（输入符号）a 时复位，事件记录自动机时钟 x_a 的值取决于输入字而非自动机。事件记录自动机的每一个事件包含一个时钟，用来记录该事件的上一次发生时间。因此，事件记录自动机的每一时钟是记录当前事件与相同前

一事件间时间的实型变量。正是该事件的最近一次发生，使得与其相联系的时钟被复位。所以，事件记录自动机的时钟与某些事件有一紧密、预定义的联系。在一实时系统模型中，对每一事件一般都能知道事件上一次发生后的时间间隔。例如，模拟一设备输入与输出间的 1-2 秒的延迟，使用时钟 z 记录上一事件后的时间值。要求输出事件发生时时钟约束位 $1<z<2$。

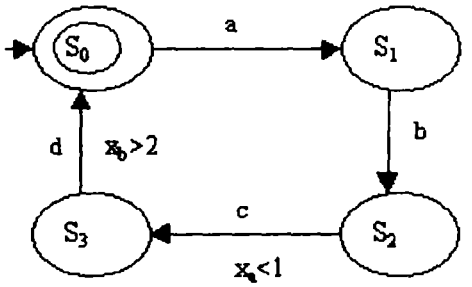


图 3.2 接受（abcd）*的事件记录自动机

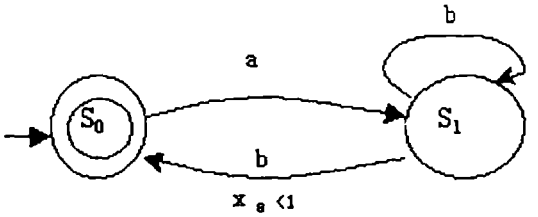


图 3.3 接受(ab*b)*的事件记录自动机

例 3.2：如图 3.2 所示的事件记录自动机其接受的语言是

$$L(A)=\{((abcd)^*,\ \tau)\mid \tau_1-\tau_3<1,\ \tau_4-\tau_2>2\}$$

时钟 x_a 确保在事件 a 发生的 1 个时间单位内，紧接其后的事件 c 发生。时钟 x_b 确保在事件 b 发生 2 个时间单位后，紧接其后的事件 d 发生。

例 3.3：如图 3.3 所示的事件记录自动机其接受的语言是(ab*b)*，时钟 x_a 确保在 a 发生一个时间单位后，事件 b 发生。

3.3 事件预测自动机

事件预测自动机中与事件相连的时钟预测该事件下一次发生的时间。如果说事件记录自动机提供了过去事件的时间进行消息，事件预测自动机则提供了关于未来事件的发生时间。设 $a\in\Sigma$。使用 y_a 表示 a 的事件预测时钟。在时间字的任一位置，y_a 的值指出了未来事件 a 的发生时间，使用特殊符号$\perp$表示事件未来不会发生。现

在重定义时钟函数 $t_j^w(y_a)$，它表示在时间字 w 上，在 j 点事件 a 下一次发生时，事件预测时钟的时钟值即：

$$t_j^w(y_a)=\begin{cases} t_j-t_i & \exists i\ j<i \wedge a_i=a \wedge (\forall\ k)\ j<k<i \rightarrow a_k\neq a \\ \perp & (\forall\ k)\ j<k \rightarrow a_k\neq a \end{cases}$$

定义：事件记录自动机（ECA）是元组$<\Sigma, S, S_0, C_\Sigma, E, F>$，其中$<\Sigma, S, S_0, C_\Sigma, E>$是其时间转换表，$C_\Sigma$内的任一时钟记录 Σ 内某一事件下次发生的时间，$F\subseteq S$是可接受状态集合。

例 3.4：图 3.4 所示的事件预测自动机接受的语言为(aa*b)"使得事件 a 发生一个时间单位后，事件 b 发生。

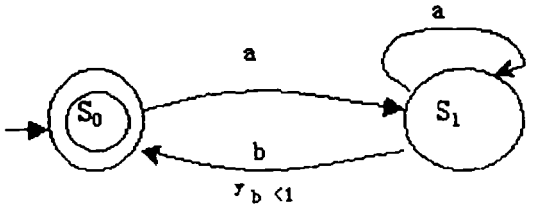


图 3.4 接受(aa*b)"的事件预测自动机

3.4 事件时钟自动机

从一、二节可以看出，所展示的这类时间自动机，其时钟变量与事件紧密相连。此类时间自动机称为事件时钟自动机(ECA)。需要指出，事件时钟自动机是通过限制时钟的使用，得到一时间自动机。自动机的时钟与事件（输入符号）有一个预定义的联系。事件记录时钟是一个历史变量，它的值等于与时间相关的事件的最后一次发生时间。事件预测变量是一预测变量，其值等于与当前时间相关的事件的下一次发生时间。

定义：事件记录自动机（ECA）是元组$<\Sigma, S, S_0, C_\Sigma, E, F>$，其中$<\Sigma, S, S_0, C_\Sigma, E>$是其时间转换表，$C_\Sigma$内的任一时钟均与 Σ 内的某一事件相联系，$F\subseteq S$是可接受状态集合。

因此事件时钟自动机并不控制其时钟的重赋值，并且对每一个转换，其所有时钟值仅取决于输入字。该属性决定了事件时钟自动机的确定性，进而导出求补过程。事实上，事件时钟自动机类、事件记录自动机类以及事件预测自动机类在所有的布尔操作下封闭，但时间自动机对补不封闭，对事件自动机而言其语言包含问题是可以判定的。在下一章中集中描述。

第四章 时间语言类的性质

本文考虑的模型其执行是无限的事件序列，而不是状态。但其与以状态为基础的模型理论仅在细节上有些差异。在此框架下，才能用一系列的时间与执行踪迹相比较，从而加入时间因素。在这里，时间序列的第 i 个时间元素给出了事件序列第 i 个时间元素的发生时间。

4.1 判空的复杂性

时间 Büchi 自动机判空算法的复杂性是时钟数和时间约束常量的几何级数，复杂性的爆炸式增长似乎是不可避免的。已经知道线形有界自动机的可接受问题是多项式完全问题（P 类问题）。可以将线形有界自动机的可接受问题转化为时间 Büchi 自动机的判空问题，从而证明判空问题是 P 下界。同样可以通过论证算法可在多项式空间内实现，来证明该问题是多项式完全的。

定理 4.1：给定时间自动机 A 的判空问题是个多项式完全问题。

证明：首先，处理多项式成员。由于区域自动机的状态数是 A 的时钟数目的几何级数，不可能构造整个转换表。但是，可以使用多项式空间预测期望路径，从而来检查区域自动机的非空性质。

现在回到多项式复杂性，判断一给定线形有界自动机是否接受一输入串是多项式完全的。线形有界自动机 M 是非确定图灵机，其带头不能超过输入标志的结尾。构造一个时间 Büchi 自动机 A 使其语言非空，当且仅当 M 对一输入停机。

Γ 使 M 的输入字母集合，Q 是其状态，$\Sigma=\Gamma\cup\{\Gamma\times Q\}$，$a_1$，$a_2$，…，$a_k$ 表示 Σ 的元素。在 M 的配置内带子读入 γ_1，γ_2，…，γ_n，机器在状态 q 读入带子第 i 个输入符号，被 Σ 的字符串 σ_1，σ_2，…，σ_n 取代，若 $i\neq j$ 且 $\sigma_i=\langle\gamma_j, q\rangle$ 那么 $\sigma_i=\gamma_j$。

在配置保持不变后，接受对应于一特别状态 q_f。A 的字母表包括 Σ，另外有一符号 a_0。M 的计算由字编码

$$\sigma_1^{\,1}\, a_0\cdots \sigma_n^{\,1}\, a_0\, \sigma_1^{\,2}\, a_0\cdots \sigma_n^{\,2}\, a_0\cdots \sigma_1^{\,j}\, a_0\cdots \sigma_n^{\,j}\, a_0\cdots$$

使得根据上表，对第 j 个配置编码 $\sigma_1^{\,j}\ldots\sigma_n^{\,j}$，同样编码计算与此字相联系的时间序列：要求相邻接的 a_0 间的时间差为 k+1。若 $\sigma_i^{\,j}=a_1$ 那么要求其时间比先前的 a_0 大 1，时间序列的编码用来实现连续条件。

根据以上方法，使用 M 的停机运算来编码时间字，构造 A，接受时间字。A 使

用 2n+1 个时钟。x 在每个 a_0 复位。当读入 a_0，要求（x=k+1）有效，读入 a_i，要求（x=i）有效，该条件确保时间序列的编码同字一致。对每一单元 i，有二时钟 x，y。对奇数的 j，x_i 使用 σ_i^j 复位。对偶数的 j，y_i 使用 σ_i^j 复位。设自动机第一次读到 j 配置时，j 是奇数。时钟 x_i 的值代表 j 配置的第 i 个单元。根据 M 的转换规则，检查 x_{i-1}，x_i，x_{i+1} 的值确定 σ_i^j 的值的选择。当读入第 j+1 个配置时，y 时钟设为相应的值。当读入第 j+2 个配置时，该值。得到检查。合适的初始化和停机，可以以非常直接的方式实现。A 的空间是 n 和 M 的空间的多项式级数。□

4.2 包含和等价

Büchi 自动机的语言包含问题可以在多项式时间内解决。但是，却不存在确定过程来检查两个时间 Büchi 自动机的语言是否存在包含关系。这是使用时间自动机作为有限实时系统的语言表述工具，进行自动验证的障碍。

公理：给定字母表 Σ 上的两个时间 Büchi 自动机 A_1、A_2，检查 $L(A_1) \subseteq L(A_2)$是 Π_1^1 复杂性的。

证明：将给定在 Σ 上的时间自动机的普遍性问题转化为语言包含问题。设 A_{univ} 是接受 Σ 上的每个时间字的自动机。已经证明，已经证明，给定字母表 Σ 上的时间自动机，判断其是否接受 Σ 上的所有时间字是 Π_1^1 复杂性的。由此，自动机 A 是普遍性的当且仅当 $L(A_{univ}) \subseteq L(A_2)$。□

现在考虑一下二自动机的检查等价性，自动机等价性的自然定义是使用其被接受语言的等价性，但是，存在其他定义方法。

定义：字母表 Σ 上的时间 Büchi 自动机 A_1、A_2，$A_1 \sim_1 A_2$ 当且仅当 $L(A_1)=L(A_2)$；$A_1 \sim_2 A_2$ 当且仅当 Σ 上的所有时间自动机 A，当 $L(A) \cap L(A_2)$为空时，$L(A) \cap L(A_1)$为空。

对在补下封闭的自动机类，上述二定义一致。由于其对补操作的不封闭性，二定义与时间正则语言类不同。事实上，$A_1 \sim_1 A_2$ 蕴涵着 $A_1 \sim_2 A_2$。反之，不成立。它的促进因素是，某一自动机由两个自动机复合而成，模拟有限状态系统，它给出了不同的行为，有一种情况是复合语言为空，另一种情况是可能有一个连接执行。

4.3 时间正则语言的性质

下一定理. 考虑了时间正则语言的封闭属性。

定理 4.2：时间正则语言类有限次并、交后仍然是封闭的。

证明：对时间 Büchi 自动机 $A_i=<\Sigma, S_i, S_{i0}, C_i, E_i, F_i>$，i=1, 2, …n。不失一般性，设时钟集合 C_i 不相交。构造接受 $L(A_i)$ 的并和交的时间 Büchi 自动机。

由于时间 Büchi 自动机是非确定的，所以对并的情况很容易处理。所求时间 Büchi

自动机仅是对标准 Büchi 自动机积的构造稍作修改，即可实现。自动机 A 的时钟集合为$\cup C_i$，A 的状态形如$\langle s_1, \cdots, s_n, k\rangle$，其中 $s_i \in S_i$，$1 \leqslant k \leqslant n$，$s_i$跟踪 A_i的状态。k 用来记录各个自动机循环通过接受条件的次数，初始值为 1，它的值从 k 增加至 k+1（mod n）当且仅当第 k 个自动机状态是可接受的状态。注意在这里 n （mod n）=n。

A 的初始状态形如$\langle s_1, \cdots, s_n, 1\rangle$，$s_i$是 A_i的初始状态。从状态的前 n 个元素出发，A 有一个以 a 标志的邻接转换，新状态为$\langle s_1', \cdots, s_n', 1\rangle$，若 $s_k \in F_k$，则 j=k+1（mod n），否则 j=k。此次转换的时钟复位集合为$\cup C_i$，与之相关的时钟约束为$\wedge \delta_i$。

记数值在 1…n 之间无限循环，当且仅当所有自动机的接受状态均满足。于是定义 A 的接受集合由形如$\langle s_1, \cdots, s_n, n\rangle$的状态组成，$s_n \in F_n$。□

在以上构造过程中，所求自动机状态数为 $n*\prod|S_i|$，时钟数目为 $\Sigma|C_i|$，边集合数目则为 $n*\prod|E_i|$。设对时钟常量二进制编码，则 $|E|$包含时钟约束的长度。

即使对时间正则语言，在有限的时间间隔内可以有任意多个事件发生，或者说符号可以彼此间任意接近。如下例所示：

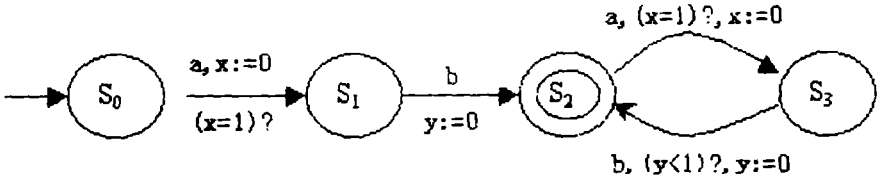


图 4.1 接受 $L_{converge}$的时间自动机

例 4.1：图 4.1 接受的语言为

$$L_{converge}=\{((ab)^{\omega}, \tau) \mid \forall i ((\tau_{2i-1}=i) \wedge (\tau_{2i}-\tau_{2i-1}>\tau_{2j-2}-\tau_{2i+1})\}$$

例 4.1 显示实型模型与离散模型截然不同。如果要求所有 τ_i的时间间隔为一个很小的固定常量 ε 的倍数，其所接受的语言将为空。即时间自动机并不区分实数集合 R 与有理数集合 Q。可间密集时间有极其重要的地位。下面给出一个定理，其显示若要求所有的序列内的时间值是有理数，非时间语言 Untime(L(A))保持不变。

定理 4.3：设 L 是时间正则语言，字 σ，$\sigma \in$Untime(L(A))当且仅当存在一时间序列 τ，使得对 i≥1，$\tau_i \in Q$ 且$(\sigma, \tau) \in L$。

证明：对时间自动机 A 与字 σ，若存在一个有理数时间序列，使得$(\sigma, \tau) \in L(A)$，显然 $\sigma \in$Untime(L(A))。

设任一时间序列 τ，$(\sigma, \tau) \in L(A)$。设 $\varepsilon \in Q$，使得每个出现在 A 内的时钟常量是 ε 的整数倍。设 $\tau_0'=0$，$\tau_0=0$。若 $\tau_j=\tau_j'+n\varepsilon$ $(0 \leqslant j<i, n \in N$，则 $\tau_i'=\tau_j'+n\varepsilon$，否则 $\tau_i' \in Q$。对 $0 \leqslant j<i, n \in N$，$\tau_i'-\tau_j'<n\varepsilon$，当且仅当 $\tau_i-\tau_j<n\varepsilon$。由于 Q 的密集性，$\tau_i'$ 的选择总是存在的。

对 A 在(σ, τ)上的可接受运行，$r=(\bar{s}, \bar{v})$，因为 τ' 的构造，若时钟在第 i

个转换点复位，那么眼时间序列 $\tau_{i'}-\tau_i$ 和 $\tau_{j'}-\tau_{i'}$，第 j 个转换点满足同样的时钟约束集。于是，就可以构造(σ,τ)上的可接受运行 $r'=(\bar{s},\bar{\upsilon})$，$r'$ 与 r 遵循同样的序列，特别是 $\upsilon_0'=\upsilon_0$。若 r 的第 i 个转换点有边$<s_{i-1}, s_i, \sigma_i, \lambda_i, \delta_i>$，那么 $\upsilon_i'=[\upsilon_i' \rightarrow 0](\upsilon_{i-1}'+\tau_i'-\tau_{i-1}')$。A 接受转换$(\sigma,\tau')$。□

4.4 补操作的非封闭性

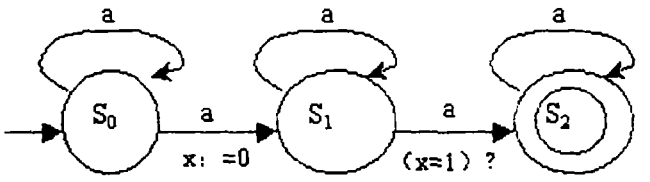
语言包含全体的Π_1^1复杂性意味着时间 Büchi 自动机对补操作不封闭。

公理：时间正则语言对补操作不封闭。

证明： 对在字母表 Σ 上的时间 Büchi 自动机 A_1、A_2，$L(A_1)\subseteq L(A_2)$当且仅当 $L(A_1)$与 $L(A_2)$的补的交为空。设时间 Büchi 自动机在补下封闭，相应的 $L(A_1)\subseteq L(A_2)$当且仅当存在时间 Büchi 自动机 A 使得 $L(A_1)\cap L(A)$非空，但是 $L(A_2)\cap L(A)$为空。即 $L(A_1)\not\subset L(A_2)$当且仅当 A_1、A_2 对$\sim_2$不等价。因此，包含问题的补是递归可枚举的，与包含问题的Π_1^1复杂性矛盾。□

例 4.2：如图 4.2 所示，自动机接受的语言是$\{(a^\omega,\tau)\mid(\exists i,j)\ \tau_j=\tau_{i+1}\}$

图 4.2 非求补自动机



该语言的补不能使用时间 Büchi 自动机符号化，其补要求确定不存在任意一对 a，其时间间隔为 1。由于在时间间隔为 1 内，关于 a 的数目无界限，在过去的一个时间间隔内，为记录 a 的时间将需要无数个时钟。

4.5 时钟约束的选择

在这里选用某些方式修改时钟约束，显示此方法怎么影响不同问题的表达能力与复杂性。时钟约束允许原子操作，时钟值与有理数常量比较，布尔连接。时间自动机仅表达关于转换间的常量的界限。

首先，扩展时钟约束的定义，允许公式包含两个时钟，例如 $x\leqslant y+c$，特别是时钟约束集合 $\Phi(X)$。对时钟 x，$y\in X$，$c\in Q$，允许使用公式 $x\leqslant y+c$ 和 $x+c\leqslant y$。非时间构造可以很容易的处理此扩展。仅需要重新定义关于时钟解释的等价关系。除去前述条件，需要两个等价的时钟解释，它们与在所有时钟约束出现的子公式一致。同样可以证明，时钟约束的扩展并不能对时间自动机的表达能力有所增加。

在时钟约束中允许加法存在，能够书写形如 $x+y \leqslant x'+y'$ 的时钟约束。这样，就可以允许自动机同不同的延迟比较，增加了形式化的表达能力（注意不是自动机的表达能力）。下例所给出的自动机语言不是时间正则的。

例 4.3：如图 4.3 所示的自动机有字母表{a，b，c}，其表达 a、b、c 循环出现。b、c 之间的时间延迟是与其相邻的上一对 a、b 间延迟的 2 倍。识别语言可以定义为

$$\{ ((abcd)^{\omega}, \tau) \mid \forall j [\tau_{3j} - \tau_{3j-1} = 2(\tau_{3j-1} - \tau_{3j-2})] \}$$

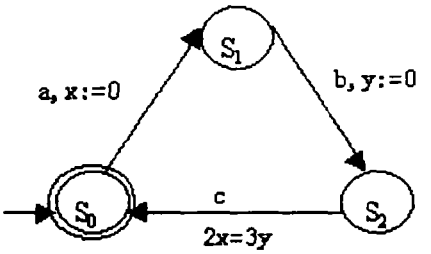


图 4.3 自动机时钟约束包含+

具有“+”的时钟约束太强了，不能被有限状态系统实现。即使要求所有事件都在整值时间内发生（即离散的时间模式），为了检查两个符号的间隔与下两个符号间隔是否相同，自动机必须有一个无限的存储空间。因此，使用有限资源，自动机仅能比较而不能记录延迟。事实上，在时间约束中引入加法，使得时间自动机的空问题不可判定。

4.6 封闭属性

考虑确定时间自动机的封闭属性，同非时间一样，被确定时间 Muller 自动机接受的语言类在布尔操作下封闭。

定理 4.4：确定的时间 Muller 自动机接受的语言类对并、交、补操作封闭。

证明：首先定义一个关于确定的时间 Muller 自动机的转换，设任意的确定的时间 Muller 自动机 $A=<\Sigma, S, S_0, C, E, f>$，对 A 求补，构造另一确定的时间 Muller 自动机 A*。先为自动机增加一空状态 q，从每一状态 s（包括 q）增加一从 s 到 q 的转换。对此边，时钟约束是起始于 s，以 a 为标志的边具有补相交时钟约束的障碍，保留接受条件不变。该构造保持了确定性和接受时间字集合。新的自动机 A*对每一状态 s 和输入符号 a，以 a 标志始于 s 的边的时钟约束的分离是有效的公式。注意，A*在任意时间字上仅有一个运行。称此类自动机是完全的。

设 $A_i=<\Sigma, S_i, s_{0i}, C_i, E_i, f_i>$（i=1、2）是两个补相交时钟集的完全确定时间 Muller 自动机。首先使用乘积构造时间转换表 A。A 的状态集合是 $S_1 \times S_2$，其出是状态是 $<s_{01}, s_{02}>$。时钟集是 $C_1 \cup C_2$。A 的转换通过连接具有相同标志的 2 个自动机的转来定义。对于 A_1 的 a-转换 $<s_1, t_1, a, \lambda_1, \delta_1>$，和 A_2 的 a-转换 $<s_2, t_2, a, \lambda_2, \delta_2>$，A 有以转换 $<<s_1, s_2>, <t_1, t_2>, a, \lambda_1 \cup \lambda_2, \delta_1 \wedge \delta_2>$。显然，A 同

样是确定的，A 在（σ，τ）上有唯一的运行，其通过将在（σ，τ）上 A_i 的唯一运行复合后得到。

设 f′ 由集合 $F\subseteq S_1\times S_2$ 组成，使得使得 F 到第一个元素的投影是 A_1 的可接受集。即：

$$f^1=\{F\subseteq S_1\times S_2 \mid \{s\in S_1 \mid \exists\, s'\in S_2,\ <s,\ s'>\in F\}\in f_1\}$$

因此，A 的运行 r 是 A_1 的运行当且仅当 inf（r）$\in f^1$。同理定义 f^2，使得 f^2 由集合 F 组成，使得$\{s' \mid \exists\, s\in S_1,\ <s,\ s'>\in F\}\in f_2$。现在，联合 A 与 Muller 接受簇 $f^1\cup f^2$，给出了接受 L（A_1）∪L（A_2）的确定 Muller 自动机，使用接受簇 $f^1\cap f^2$ 给出接受 L（A_1）∩L（A_2）的确定 Muller 自动机。

最后对自动机求补。设 A 是完全确定 Muller 自动机 A=<Σ，S，s_0，C，E，f>。A 在给定时间字上恰有一运行，因此，（σ，τ）是在 L（A）上的补当且仅当 A 在其上的运行不满足 A 的接受条件。因此，补语言是一确定时间 Muller 自动机所接受的具有和 A 一样的时间转换表，其接受条件是 2^S-f。□

考虑确定时间 Büchi 自动机的封闭性质，如前所述，确定的 Büchi 自动机对补不封闭。性质有无限个 a 可以被确定 Büchi 自动机所表述，但其补，仅有有限个 a 不能被确定 Büchi 自动机所表述。不能期望确定时间 Büchi 自动机类对补封闭，但是由于每个确定时间 Büchi 自动机可视为一个确定时间 Muller 自动机，确定时间 Büchi 自动机的补被一个确定时间 Muller 自动机接受。下一定理描述了封闭性质。

定理 4.5：被确定时间 Büchi 自动机接受的时间语言类的并、交操作封闭，但其补操作不封闭。确定时间 Büchi 自动机的补语言，被某个确定时间 Muller 自动机所接受。

证明：对并使用上述的定理 4.4 证明方法中的构造法，如同确定时间 Muller 自动机乘积构造转换表法。其接受条件是$\{<s,\ s'>\mid s\in F_1\vee s'\in F_2\}$。时间 Büchi 自动机的乘积构造显示该方法保持了确定性。而时间 Büchi 自动机的交的封闭性如下：

确定 Büchi 自动机补的非封闭性导致了确定时间 Büchi 自动机的补的非封闭性。$\{(\sigma,\tau)\mid \sigma\in (a+b)^*b^\omega\}$不能被一确定时间 Büchi 自动机所接受。$(a+b)^*b^\omega$不可能被被确定 Büchi 自动机所表述。

设 A=<Σ，S，s_0，C，E，F>是一个确定自动机的补，（σ，τ）属于 L（A）当且仅当 A 在其上唯一的运行不满足 A 的接受条件。因此被确定时间 Muller 自动机接受的补语言具有和 A 一样的时间转换表，接受簇为 2^S-F。□

事件时钟自动机作为时间自动机子集，下一定理. 给出了其对布尔操作的封闭性。

定理 4.6：被事件时钟自动机接受的语言类对布尔操作交、并、补封闭。

证明：由于事件时钟自动机允许多重起始，其并操作显然成立。

同样，可以以很简便的方式构造其交。对给定的事件自动机 A_1、A_2，其乘积 $A_1\times A_2$ 满足事件自动机。$A_1\times A_2$ 的状态是一个序偶，由 A_1、A_2 的状态组成，其每一个以

a 标志的边对应 A_1、A_2 以 a 标志的边。其时钟约束对应 A_1、A_2 以 a 标志边的约束的交。

它的补操作的封闭性依赖于确定性构造。给定一个事件时钟自动机 A，A 对应的确定时间自动机的补，来自于 A 对应的确定时间自动机接受条件的补。

4.7 表达能力

比较不同类型的时间自动机，仅仅通过重写接受条件，每一个确定时间 Büchi 自动机可以轻易地似的用确定时间 Muller 自动机表达。首先，每一个 ω-正则语言和确定 Muller 自动机的表达能力相同。因此，也同确定时间 Muller 自动机的表达能力相同。换句话说，确定的 Büchi 自动机比确定的 Muller 自动机的表达能力弱得多。某些ω-正则语言不能被确定 Büchi 自动机表述。下一引理显示，同样不能使用确定时间 Büchi 自动机表示此类语言。从而，确定时间 Büchi 自动机比确定时间 Muller 自动机的表达能力也要弱。事实上确定时间 Muller 自动机对补封闭，而确定时间 Büchi 自动机则不行。

定理 4.7：对 ω-正则语言，时间语言$\{(\sigma,\tau)\mid\sigma\in L\}$可以被某个确定时间 Büchi 自动机接受当且仅当 L 可被某些确定 Büchi 自动机接受。

证明：显然，若 L 被一确定 Büchi 自动机接受，那么$\{(\sigma,\tau)\mid\sigma\in L\}$可被视为时间自动机的同样的自动机接受。

设语言$\{(\sigma,\tau)\mid\sigma\in L\}$被某个确定时间 Büchi 自动机接受，构造另一个确定时间 Büchi 自动机 A′ 使得 $L(A')=\{(\sigma,\tau)\mid(\sigma\in L)\wedge\forall\ i\ (\tau_i=i)\}$，A′ 要求在每一个转换转换时间增 1。自动机 A′ 可通过 A 引入一特别时钟 x，获得 A 内的每一边增加时间约束 x=1，要求起在每一条边复位。A′ 同样是确定的。

接着构造 A′ 的非时间延迟。注意 $Untime(L(A'))=L$。构造 $R(A')$ 时，仅需要考虑那些时钟分数部分为 0 的时钟区域。由于在每一步时间的增加是预先确定的，A′ 也是确定的，可得 $R(A')$ 是确定转换表。同样需要检查递增条件。可得非时间过程构造的自动机是一个接受 L 的确定 Büchi 自动机。□

时间语言类	封闭操作
TMA=TBA	并、交
∪	
DTMA	并、交、补
∪	
DTBA	并、交

图 4.4 时间自动机类

ω-语言类	封闭操作
MA=BA=DMA ∪ DBA	并、交、补 并、交

图 4.5 ω-自动机类

从以上讨论可得对确定时间 Muller 自动机的语言 L 若 Untime（L）是确定 Büchi 自动机语言，则 L 是确定时间 Büchi 自动机语言。考虑使用确定时间 Muller 自动机实现收敛反应属性 L_{crt} 的可表达性。该语言还包含时间的联结。设有确定时间 Büchi 自动机可表述该属性（即是 Untime（L_{crt}）可被 Büchi 自动机描述）。

沿上述证明的思路，同样可以显示ω-正则语言 L，时间语言{（σ，τ）|σ∈L} 被某个确定时间 Muller 自动机（或时间 Muller 自动机或时间 Büchi 自动机）所接受，当且仅当 L 被某个确定 Muller 自动机（或 Muller 自动机或 Büchi 自动机）所接受。

由于确定 Muller 自动机与时间 Muller 自动机不同，其对补封闭，可得被确定时间 Muller 自动机接受的语言类比时间 Muller 自动机小。特别时有些语言，例如某些 a 对，时间间隔为 1，不能用确定 Muller 自动机表示，其依赖于非确定性。

图 4.4 显示了不同类之间的包含性及封闭性，与此相比较，ω-自动机不同类的自动机也于图 4.5 中给出了。

对于事件时钟自动机而言，由于证明相似所以集中给出。在给出之前，先规定一些符号。使用 DTA 表示确定时间自动机所识别的语言，ERA 表示事件记录自动机所识别的语言，EPA 表示事件预测自动机所识别的语言，ECA 表示事件时钟所识别的语言，NTA 表示非确定时间自动机所识别的语言。

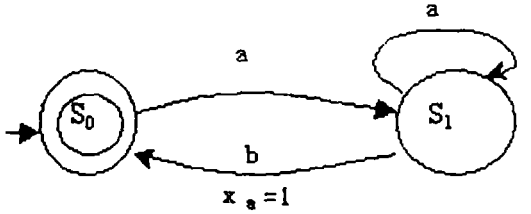


图 4.6 一个事件记录自动机

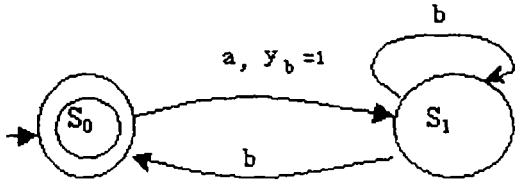


图 4.7 一个事件预测自动机

定理 4.8：事件记录自动机（ERA）与（EPA）不存在包含关系。

证明：（1）ERA⊄EPA

给出一个反例。如图 4.6 所示的事件记录自动机 A 就不能被事件预测自动机所定义。假设存在一个事件预测自动机 B 可以定义语言 L（A），不失一般性，设 B 的时钟约束仅含整型常量。对时间字 w_1=（a,0）(b,0.5)(b,1) …与 w_1=（a,0）(b,0.5)(b,0.9) …。事件预测自动机 B 使用时钟变量 y_a, y_b 上的时钟约束。尽管时钟函数 t_j^w（y_a）在 w_1 与 w_2 上取值不同，由于时钟约束仅含整型常量，不能检测到。对 B 的任意约束 φ 及任意位置 0≤j≤2，t_j^{w1}➔φ，当且仅当 t_j^{w2}➔φ。因此，B 接受 w_1 当且仅当 B 接受 w_2。但是，自动机 A 接受 w_1 却不接受 w_2。矛盾。

（2） EPA⊄ERA

如图 4.7 所示的事件预测自动机与（1）类似可证矛盾。

综合（1）、（2）结论显然成立。□

定理 4.9：事件记录自动机和事件预测自动机的并是事件时钟自动机的真子集。

证明：设自动机 A_1，A_2 的并为事件时钟 A，类似于定理 4.9 可证时间语言 L（A_1），L（A_2）不属于事件记录自动机也不属于事件预测自动机。即事件记录自动机和事件预测自动机的并是事件时钟自动机的真子集。□

定理 4.10：事件时钟自动机是非确定时间自动机的真子集。

证明：由于非确定时间自动机对补操作不封闭，显然成立。□

为了便于验证定理 4.11 与定理 4.12 仅在有限字中验证，但是结论很容易推广到 ω 字中。

定理 4.11：事件预测自动机不是确定时间自动机的子集。

证明：时间语言 L={（a^kb, τ）| $0≤j<k, t_k-t_j=1$}被事件预测自动机接受，该事件预测自动机仅要求在仅含 a 的字符串中，时钟约束 $y_b=1$ 满足。但是 L 不能被确定时间自动机接受。假设，存在某一个确定时间自动机 B 接受语言 L，设 B 仅使用整型常量，具有 m 个时钟。设时间字 w=（a^{m+2}, $\tau_0 \cdots \tau_{m+1}$）$0=\tau_0<\tau_1<\cdots<\tau_m<\tau_{m+1}=1$。由于 B 是确定的，对 B 读入时间字 w，B 最多只有一个计算。由于 B 最多有 m 个时钟，最少有一个位置 1≤j≤m，使得当自动机读入（a,1）时，B 没有时钟具有值 $1-t_j$。因此自动机 B 忽略了时间 t_j。在 w 后增加输入（b,t_j+1）形成 w_1，在 w 后增加输入（b,t' +1）形成 w_2，其中 $t' \neq t_j$，并且有 $\tau_{j-1}<t'<\tau_{j+1}$。对时间字 w_1 与 w_1，当读入第 m+2 个输入（新增的输入）时，B 的时钟满足相同的时钟约束。因此，自动机 B 接受 w_1 当且仅当其接受 w_2。但是 w_1 在 L 中，w_2 却不属于 L。□

定理 4.12：确定时间自动机不是事件记录自动机的子集。

证明：语言 L={（aaa，$t_0t_1t_2$）| $t_2-t_0=1$}被确定时间自动机接受，该自动机仅需一个时钟 x，当读入第一个 a 时复位，读入第三个 a 时检查时钟约束 x=1，但是不能被事件时钟自动机接受。证明与定理 4.9 相似。□

事件自动机与时间自动机类的关系图 4.8。

一方面，事件记录自动机具有足够的表达能力来模拟有限时间转换系统，另一

方面，它是确定的，对所有的布尔操作封闭，因此，语言包含问题对事件自动机而言是封闭的。它是检验两个时间转换系统是否具有相同的时间行为集合很好的工具。

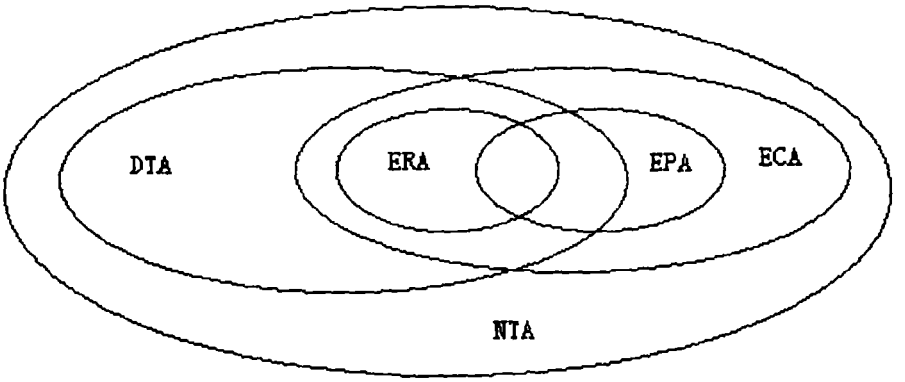


图 4.8 事件时钟自动机与时间自动机类的关系

第五章 自动验证

实时系统（例如进程控制系统，符号处理系统等等）之类的系统，要求持续对其环境做出反应，常常要求程序员指定时间约束，例如某一输入要求在某一段时间间隔内发生。时间概念在此类系统中特别重要。它要求一类很严格的时间约束。迄今，已经开发出一些形式化方法来处理此类系统，他们可归为以下三类：

1.时间进程代数。通过在经典的进程代数（例如 CCS,CSP,和 ACP）中加入时间概念和若干时间操作符。所得公式与其不含时间因素的祖先完全不同。由于它们允许模拟诸如暂停，例外，先验，中断等概念，所以可应用于系统的说明和验证。

2.以时序逻辑为基础的各类形式化。它们的主要工作在于系统的验证。最近作为有力的工具，可执行时序逻辑（EPL）被提了出来。其将逻辑观点与操作模式相结合而形成。该类形式化已经应用于某些领域内的应用。包括硬件仿真，时序数据库和时序平面图。已经存在诸如 Chronolog, FLIMETTE, Concurrent METATEM ，和 Tempura 之类的 ETL 系统。

3.包含专门为此类系统编程的语言。一般而言，此类语言主要应用确定自动机理论编制。由于大自动机很难设计、维护、修改。某些应用于此类系统的高级语言，在过去几年内一直在设计中。并行同步语言 ESTERL, LUSTRE, SINGNAL, 和 Statechars 已经在一些工业领域中得到应用。这类语言以实时反应为前提，即程序是由某些输入符号激活并立即反应，产生输出。若不存在语句确实消耗时间，计算就能立即执行。计算通过对系统的所有进程立即广播完成。一个符号的产生或消失可在任何时间侦测到。通过将程序（满足某些要求）编译为有限状态自动机，完美的同步设想就能在实际中实现。自动机的单步执行时间是受限度。

5.1 验证

下面给出了如何使用时间自动机理论来证实有限状态实时系统的正确性，并给出了一个验证问题的简单形式，但是足够演示时间自动机的验证问题的应用。首先介绍并行进程的线形语义学。

5.1.1 追踪语义

在追踪语义内，可以观察事件与进程的联系。使用所有追踪的集合来模拟进程。追踪时进程运行时可见的（线形）事件序列。例如，事件可表示对变量赋值，

或是在控制板上按下某一个按钮，或是消息的到达。设所有的事件立刻发生，具有间隔的行为使用事件标志行为的起止。Hoare 最先对 CSP 系统提出了此类模型。

在本文的模型中，追踪是一系列的事件集合，因此，若事件 a，b 同时发生，在该模型内相应的追踪有集合{a, b}。在一般模型内，该集合可被所有可能的序列取代，即 a 可以跟随 b 同样 b 也可以跟随 a。同样，仅使用无限序列来模拟实时系统的非终止交互作用。

形式地，给定一个事件集合 A，追踪 $\sigma=\sigma_1\sigma_2\cdots$是 $\rho^*(A)$（A 的非空子集合）上的无限字。非时间进程是序偶（A，X），由系统的可见事件集合 A 和可能的追踪集合 X 组成。

例 5.1：设信道 P 联结二元素，a 代表一个消息到达 P 的末端，b 代表在 P 的另一端消息的传送。当前一消息未达到另一端时，P 不能接受新消息。相应地事件 a, b 交替出现。设消息不断出现，唯一可能的追踪是：

$$\sigma_P:\ \{a\}\rightarrow\{b\}\rightarrow\{a\}\rightarrow\{b\}\rightarrow\cdots$$

使用符号 a 表示平凡集{a}，P 的进程可表示为（{a, b}，（ab）ω）。

对进程可定义不同的操作，但是使用简单的操作对描述复杂系统大有裨益。在这进考虑最重要的此类操作，即并行复合操作。进程集合并行复合描述所有并行进程的联结行为。

使用投影操作可以方便的定义并行复合操作，投影 $\sigma\in\rho^*(A)^\omega$到 $B\subseteq A$ 写作 $\sigma\lceil B$，交使用就是 σ 内的集合与 B 相交，从序列中删去所有的空集合。例如 $\sigma_P\lceil\{a\}$ 是追踪 aω。注意，投影操作可以导致有限序列，但对无限个 i 当 $\sigma_P\cap B$ 非空时足以表达追踪 σ 到 B 的投影。

对进程$\{P_i=(A_i, X_i)\mid i=1, 2, \cdots n\}$，其并行复合 $||P_i$ 是具有事件集合 $\cup A_i$ 的进程，其追踪是

$$\{\sigma\in\rho^*(\cup A_i)^\omega\mid\wedge\sigma\lceil A_i\in X_i\}$$

因此 σ 是$||P_i$的追踪当且仅当 $\sigma\lceil A_i$ 是 P_i 的追踪，i=1，2…n。当不存在公共元素时，上述定义相当与所有追踪的非约束插入。

例 5.2：对另一信道 Q 与上例中的 P 相连。Q 消息事件的到达和 b 相同。设 c 表示 Q 另一端消息的传递，进程 Q 由（{b，c}，（bc）ω）给出。P、Q 复合，要求它们在公共元素 b 上同步，在每一对 b 之间允许事件 a 发生在 c 之前，c 发生在 a 之前和二者同时发生。因此，[P||Q]具有事件集{a, b, c}，有无限个追踪。

在此框架下，验证问题表现为包含问题。实现和表述都以非时间进程给出，实现进程是若干较小元素进程的复合。对于表述（A，X_S），实现（A，X_I）是正确的当且仅当 $X_I\subseteq X_S$。

例 5.3：对上例所示信道，实现进程是[P||Q]。表述作为进程 S=（{a，b，c}，（abc）ω）给出。因此，表述要求另一消息到达 P 之前，消息到达 Q 的另一端。此时，由于[P||Q]具有太多的追踪，例如 ab（acb）ω，其不满足 S 的表述。

注意，根据以上验证的定义，$X_I=\varnothing$对每一个表述的实现都是正确的。为克服此

问题，需要区别系统控制的输出事件和环境控制的输入事件。要求实现不能阻止环境执行输入事件。

5.1.2 追踪中加入时间

非时间进程模拟事件序列，而非事件发生的确切时间。因此，上例仅给出了事件a,b 的序列而无事件间的延迟。当然，可以通过与一时间值序列相比较而加入时间，同样，在这里使用实数集来模拟时间。

时间序列 $\tau=\tau_1\tau_2\cdots$ 是时间值的无限序列，$\tau_i\in R$，满足严格单调递增条件。在事件集合 A 上的时间追踪是一序偶（σ，τ），其中 σ 是 A 上的追踪。τ 是时间序列。注意，在追踪中，不同事件在一个元素中可同时发生。

在时间字（σ，τ）内每个 τ_i 给出了事件 σ_i 的发生时间，特别指出，τ_1 给出了第一个事件的发生时间。设 $\tau_1>0$，$\tau_0=0$。递增条件蕴涵着在有界时间间隔内，仅有有限个事件可以发生。排除了诸如 1/2，3/4，7/8…之类的收敛的时间序列，在一个时间单位之前，系统有无限多个事件。

事件进程是一个序偶（A，L），A 是有限事件集，L 是 A 上的时间追踪字集。例如对信道 P 设第一个消息在时间 1 到达，后继消息以 3 个时间单位的固定间隔到达。进而，对每个消息以 1 个时间单位通过信道。进程仅有一个时间追踪：

$$\rho_P=(a,1)\rightarrow(b,2)\rightarrow(a,4)\rightarrow(b,5)\rightarrow\cdots$$

其所代表的时间进程为：$P^T=(\{a,b\},\{\rho_P\})$。

非时间进程的操作以很明显的方式扩展到时间进程。为得到（σ，τ）到 $B\subseteq A$ 的投影，首先以 B 交 σ 的每个事件集合，沿与之相关的时间值删去空集。并行复合操作除了应用于时间追踪的操作外，保持不变。因此，对两个进程的并行复合操作，要求二者在同一时间应该有共同的事件参与。就排除了插入的可能性，即二时间追踪的并行复合或是平凡的或时间追踪为空。

例 5.4：还看两信道 P、Q 相连的例子。对 Q 唯一可能的追踪是 $\sigma_Q=(bc)^\omega$。另外，Q 的时间表述表明一个消息川过信道的时间，即相邻 b,c 间的延迟是某个在 1，2 之间的实数。时间进程 Q^T 有无限个时间追踪，它们可形式的表示为：

$[\{b,c\},\{(\sigma_Q,\tau)\mid\forall\ i\ (\tau_{2i-1}+1<\tau_{2i}<\tau_{2i-1}+2)\}]$

$[P^T,Q^T]$的描述通过复合 ρ_P 和每一个时间追踪 Q^T 得到。复合进程有无数个时间追踪。例如：$(a,1)\rightarrow(b,2)\rightarrow(c,3.8)\rightarrow(a,4)\rightarrow(b,5)\rightarrow(c,6.02)\rightarrow\cdots$

与时钟相连的时间值通过 Untime 操作忽略。对时间追踪 P=（A，L），Untime（A，L）是非时间进程，其是由事件集合 A 和追踪 σ 组成的使得对某个时间序列 τ，（σ，τ）$\mid\in$L 的追踪集组成。注意

$$Untime(P_1\|P_2)\subseteq Untime(P_1)\|Untime(P_1\|P_2)$$

但是，如同下一例子所演示的那样，两边没有必要相等。换句话说，当二进程复

合时，时间追踪保留的信息约束了可能的追踪。

例 5.5：再次考虑信道 P，Q 相连的例子，Untime（P^T）=P，Untime（Q^T）=Q，即[P||Q]JYOU 仅有非时间追踪（abc）$^\omega$。即[P||Q]有无限个追踪。在每一对 b 之间，事件 a、c 可以任何次序进入。

验证问题再一次展示为包含问题。现在实现作为若干时间进程的复合给出，表述也以时间进程给出。

例 5.6：还看信道 P，Q 相连的例子。若以时间进程[P||Q]模拟实现，那么其满足表述 S。S 是一时间进程（{a, b, c}，{（abc）$^\omega$，τ}）。尽管表述 S 仅包含事件序列。[P^T||Q^T]对应于 S 的正确性依赖于两个信道的时间约束。

5.1.3 ω-自动机和验证

首先，简述一下使用 Büchi 自动机验证非时间进程。对非时间进程（A，X），X 是字母表 $\rho^*(A)^\omega$ 上的 ω-语言。若其是正则语言，则可被 Büchi 自动机表示。

使用字母表 $\rho^*(A)^\omega$ 上的 Büchi 自动机 A_P 模拟具有事件集 A 的（非时间的）有限状态进程 P。自动机的状态对应于进程的内部状态。若进程对 a 事件，自动机 A_P 可以从状态 s 转至 s′，则其具有一转换<s，s′，a>，a⊆A。自动机的接受条件答应于进程约束的有效性。自动机 A_P 接受或产生 P 的追踪，即进程 P 由（A，L（A_P））给出，此进程称为 ω-正则进程。

通过对每一个元素以 Büchi 自动机来描述，可以描述整个系统。设系统 I 由 n 个元素组成，每一个进程以 ω-正则进程 P_i=（A，L（A_P））描述。实现进程是[||P_i]。使用 Büchi 自动机的语言交的构造方式，可以自动构造 I 的自动机。由于不同元素的事件集不同，在应用乘积构造法之前，应使不同的自动机字母表不同。设 A=∪A_i，对每个 A_i 构造字母表 $\rho^*(A_i)$ 上的自动机 A_i'，使得 L（A_i'）={σ∈$\rho^*(A)^\omega$ | σ┌A_i∈L（A_i）}所求自动机 A_i 是自动机 A_i' 的乘积。

表述作为 $\rho^*(A)^\omega$ 上的 ω-正则语言 S 给出，实现满足要求当且仅当 L（A_I）⊆S。S 的属性使用 Büchi 自动机 A_S 给出，此时，验证问题周围 L（A_I）∩L（A_S）的判定。

验证问题是多项式完全的 A_I 的空间是各个元素空间的扩展。若 A_S 非确定，求补过程将是爆炸性的扩展。因此，验证的复杂性同样是表述的几何级数。但若 A_S 是确定的，复杂性仅是表述空间的多项式级数。

5.1.4 使用时间自动机验证

对时间进程（A，L），L 是 $\rho^*(A)$ 上的时间语言。时间进程的集合 L 是时间正则语言时，可用时间自动机表示。

有限状态系统被时间 Büchi 自动机模拟，转换表给出了系统的状态转换图。已经

给出了如何使用时钟取代不同元素的时间延迟。如前所述，接受条件对应于公平条件，因此，与时间 Büchi 自动机 A_P 相联系的有限状态进程 P，使得 $L(A_P)$ 由 P 的时间进程组成。

一般而言，实现被描述为 n 个元素的复合，每个元素应由一时间进程 $P_i=(A_i, L(A_i))$ 模拟。可以构造一时间 Büchi 自动机 A_I 来取代复合进程$[\|P_i]$。为此，首先使不同自动机的字母表不同，然后取其交。但在验证过程中需要指出，将不明确的构造实现自动机 A_I。

系统表述作为字母表 $\rho^*(A)$ 上的另一时间正则语言 S 给出，$A=\cup A_i$。系统是正确的当且仅当 $L(A_I)\subset S$。若 S 是以时间 Büchi 自动机给出，一般而言，正确性检查是不可判定的。但是，若 S 是以确定时间 Muller 自动机 A_S 给出，如前所述，就可解决此问题。

定理：对时间进程 $P_i=(A_i, L(A_i))$，i=1，…，n。被时间自动机 A_i 模拟。有一个确定的时间自动机 A_S。$L(A_S)$ 内$[\|P_i]$的语义集的包含性检查，可在多项式空间内完成。

证明：对时间 Büchi 自动机 $A_i=<\rho^*(A_i), S_i, S_{i0}, C_i, E_i, F_i>$，i=1，…，n。确定时间 Muller 自动机 $A_S=<\rho^*(A), S_0, S_{00}, C_0, E_0, f_0>$。不失一般性，设 C_i i=0，1，…，n 不相交。

构造区域自动机的转换表算法，对应于 A_S 与 A_i 的时间转换表乘积 A，A 的时钟集表示为 $C=\cup C_i$，A 的状态形如$<s_0, s_1, \cdots, s_n>$，每一个 $s_i\in S_i$，A 的初始状态形如$<s_0, s_1, \cdots, s_n>$，每一个 $s_i\in S_{i0}$。A 的转换由联结具有相同事件集标志的各自动机的转换而得。状态 $s=<s_0, \cdots, s_n>$有一到状态 $s'=<s_0', \cdots, s_n'>$的转换，具有事件标志集 $a\in\rho^*(A)$，时钟约束 $\wedge\delta_i$，和时钟集 $\cup\lambda_i$ 当且仅当对 $0\leqslant i\leqslant n$，或有一转换$<s_i, s_i', a\cap A_i, \lambda_i, \delta_i>\in E_i$，或自动机 A_i 不参与此转换，即 $s_i'=s_i$，$a\cap A_i=\varnothing$，$\lambda_i=\varnothing$；$\delta_i=$true。

区域自动机 R（A）使用 A 的乘积定义。为检查包含性，算法在区域自动机内寻求循环，使得（1）从区域许多即的初始状态该循环是可进入的；（2）满足递增条件，对每一个时钟 $x\in C$，循环至少包含一个区域，满足[（$x=0$）∨（$x>c_x$）]；（3）复合的定义要求仅考虑无限运行。在其内，每一个自动机都无限参与。要求对 $1\leqslant i\leqslant n$，自动机 A_i 参与一个转换；（4）公平性要求所有的实现自动机 A_i 都能满足对 $1\leqslant i\leqslant n$，循环包含某个状态，其第 I 个属于接受集 F_i；（5）表述的公平性条件不满足，即循环的状态到元素 A_S 的元素的投影不属于接受簇 f。包含性无效，当且仅当循环内以上条件都可满足。

区域自动机的每个状态都可在所求深入自动机的多项式空间内取代，由此，包含性检查可在多项式内完成。□

5.2 验证实例

考虑一个自动控制仪的例子，其控制铁路道口门的开关。系统由三个元素组成，火车，门和控制器。

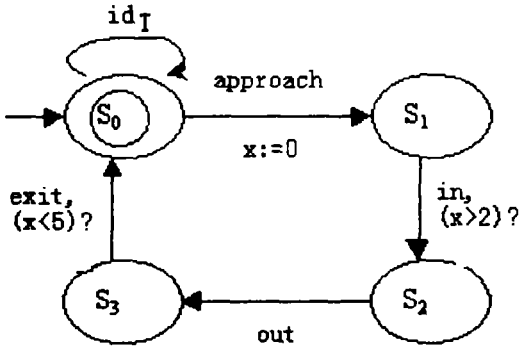


图 5.1 火车

自动机模拟火车如图 5.1 所示，事件集为{approach，exit，in，out，id_T}。火车起始于状态 s_0，事件 idT 代表火车不请求进入。火车使用事件“approach”、“exit”与控制器联络，事件“in”、“out”标志火车从道口进入、exit。要求火车进入道口，至少 2 分钟前发出 approach 符号，因此 approach、进入的最小延迟是 2 分钟。符号 approach、exit 的最大延迟是 5 分钟。这是火车要求的有效性，使用一个时钟 x 表达时间要求。

自动机模拟门如图 5.2 所示，事件集为{raise，lower，up，down，id_G }。状态 s_0 门开，s_2 门关。通过 raise、lower 与控制器联系。up、down 表示门的开放与闭合。门对符号 lower 在关闭 1 分内反应，1-2 分内响应 up。可在状态 s_0,s_2 永远执行 id_G 转换。

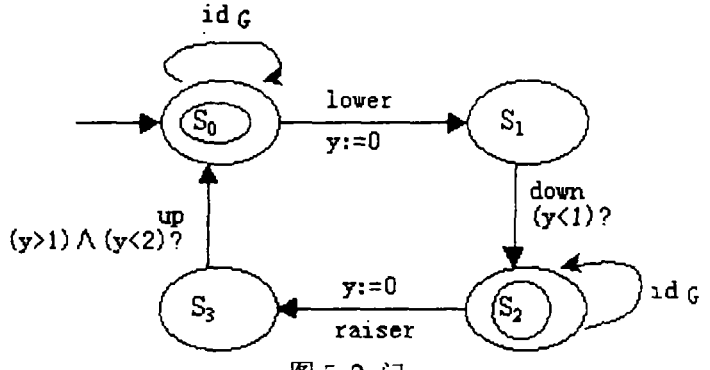


图 5.2 门

最后图 5.3 显示自动机模拟控制器，事件集是{approach，exit，up，lower，id_C }。Id_C 状态是 s_0。一旦其接受火车发送的 approach 符号，立即通过对门发出符号 lower 来反应，反应时间为 1 分钟。接受符号 exit，在一分钟内向门发出 up 符号。

整个系统是[火车||门||控制器]。事件集是三个元素的并。所有的自动机是确定的，

每个运行是可接受的。复合以上三个自动机，时间自动机 A_1 表述了整个系统。系统正确性要求如下：

安全性：一旦火车在道内，门必须关上；

实时有效性：10 分间隔内，门决不关。

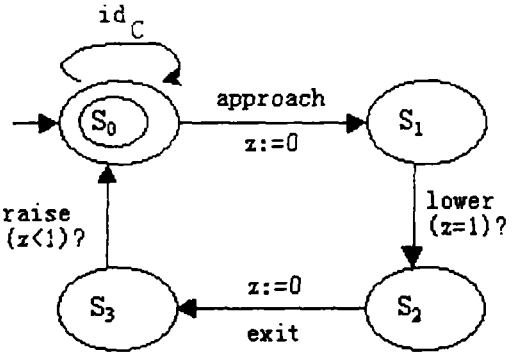


图 5.3 控制器

表述仅描述事件 in，out，up，down。自动机安全属性由图 5.4 表述。“in” 标志的边代表任何包含 “in” 的事件集。以 “in，¬ out” 标志的边意味着任何 in 但不包含 out 的事件集。自动机不允许 in 在 down 之前和 up 在 out 之前。所有的状态都是接受状态。实时有效性由如图 5.5 所示的时间自动机给出，自动机要求在 down 在发生 up 后 10 分内产生。

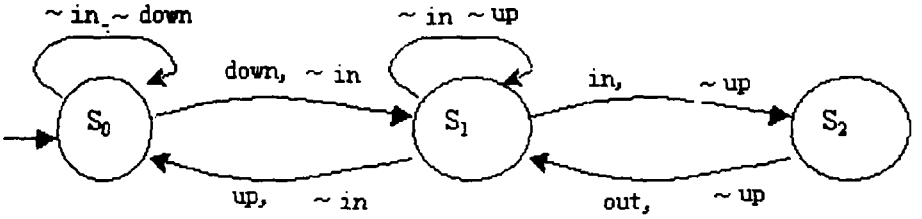


图 5.4 安全属性

自动机是确定的，因此可求补。显然，接受条件不是必要的，状态 s_1 同样可在接受集内。时间递推性确保不永远访问状态 s_0。状态 s_1 的自循环具有约束 $x<10$。可以分别查出 A_2 的正确性违反了表述。显然，尽管齐全属性是纯粹的定量属性，忽略时间属性则不成立。

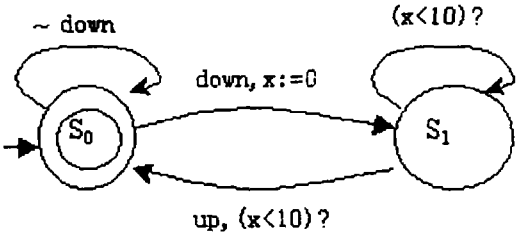


图 5.5 实时有效性

第六章 结 语

本文的主要工作是构造区域自动机。它是针对区域自动机的几何级空间复杂度而提出的。在时间自动机发生状态转换时，通过分析时间约束内不同时钟变量的取值空间，然后取各空间的交集而得到的。如果时间自动机当前状态是可达的，那么一个与事件相联系的时间值只要属于这个空间内，必然使得该约束为真。如果该空间不存在，则转换必然不存在。

时间自动机的行为预测包含两个方面，即时间与事件。对于事件的预测，通过寻找状态转换图的最大强连通分量，可以找到多项式级的复杂度判定方法。下一步工作将是使用区域自动机联系事件与时间进行事件的预测。时间自动机接受的是无限字，但是可接受字总满足一定的接受条件，具有一定的规律。通过事件预测将能更好的发现并研究其规律性。

致 谢

在这里我首先要衷心感谢我的导师庄雷教授。三年来正是庄老师的悉心指导与严格要求，才使我初步掌握了科学研究的方法与规律，才使得我三年来的学习生活丰富而又充实，领略了在知识的海洋中遨游的乐趣。尤其是在本文的写作过程中，从始至终都得到了庄老师的指导，并且给我提出了很多宝贵意见。还要感谢苏锦祥教授，对本文提出了许多建设意见。同时，还要感谢时间自动机讨论班的周清雷老师，周文俊老师和郑丽萍同学，对本文的形成，给予了很大的帮助。在此谨表示最诚挚的谢意，谢谢三年来对我的帮助与支持，使我在郑大度过了紧张而又美好的三年。

最后，还要感谢系资料室的庞军老师，程文扬老师为我提供了很多有价值的资料，为我工作的开展提供了很大的便利。

参考文献

[1] R. Alur, and D. L. Dill, (1994), A theory of timed automata, in Theoret. Comput. Sci.126,183-235.

[2] R. Alur, A. Itai, R. P. Kurshan, and M.Yannakakis (1995), Timing verification by successive approximation, in Information and Computation 118,142-157.

[3] Rina S. Cohen and Arie Y. Gold(1977), Theory of ω-Language, in Journal of Computer and System Science 15,168-184

[4] A.P Sistla, M. Vardi and P. Wolper, The complementation problem for Büchi automation with application to temporal logic(1987), in Theoret. Comput. Sci.49 217-237

[5] S. Safra On the complexity of ω-automata, in Proc. 29th IEEE Symp. on Foundations of computer science(1988) 319-327

[6] Aggarwal S. and Kurahan R.P. and Sharma D.(1983), A Language for Specification and Analysis of protocols, in IFIP Protocol Specification, Testing and verification 3 181-402

[7] Clarke E.M., Emerson E.A. and Sistla A.P.(1986),Automatic Verification of Finite-state Concurrent Systems Using Temporal-logic Specifications, ACM Trans. Programing Languages Systems8(2),244-263

[8] Browne M.C., Clarke E.M., Dill D.L., and Mishra B.(1986) Automatic Verification of Sequential Circuits using Temporal Logic, IEEE.

[9] R.Alur, C.Courcoubertis and D.Dill. Model-checking for real-time systems. In: *Proc. 5th IEEE Symp.on logic in computer Science(1990)414-425.*

[10] R.Alur, C.Courcoubertis and D.Dill. Model-checking in dense real time. Inform. and Comput.104(1993)2-34.

[11] R.Alur,L.Fix,T.A.Henzinger. Event-clock automata:a determinizable class of timed automata. In : Theoret. Comput. Sci.211(1999) 253-273.

[12]A.sistla,M.Vardi,P.Wolper. The complementation problem for Büchi automata with application to temporal logic. In : Theoret. Comput. Sci.49(1987) 217-237.

[13]G.Cattaneo,E.Formenti, G. Manzini and L. Margara. Ergodicity, transitivity and regularity for linear cellular automata over Z_m. In : Theoret. Comput. Sci.233(2000) 147-164.

[14]D. Giammarresi and R. Montalbano. Deterministic generalized automata.

In : Theoret. Comput. Sci.215(1999) 191-208.

[15] P. Caron and D. Ziazi. Characterization of Glushkov automata. In : Theoret. Comput.

Sci.233(2000) 75-90.

[16] K. Hashiguchi. New upper bounds to the limitedness of distance automata. in: Theoret. Comput. Sci.233(2000) 19-32.

[17] R. Alur, and, T.A.Henzinger, A really temporal logic. in: Proc.30th IEEE Symp. on Foundations of Comput.Sci.(1989)164-176

[18]. A.N.,Algorithms finding the order of local testability of deterministic finite automaton and estimations of the order. in: Theoret. Comput. Sci.235(2000) 183-204.