离散时间区间时序逻辑可满足性的判定

朱维军[1,2],张海宾[1],周清雷[2]
(1.西安电子科技大学计算机学院,陕西西安 710071;
2.郑州大学信息工程学院,河南郑州 450052)

摘 要: 目前还没有模型检查的方法自动检测模型是否满足时间区间时序逻辑描述的性质.我们约束时间域到离散时间,证明了离散时间区间时序逻辑的可满足性是可判定的,因而是可模型检查的.提出了时间正则图模型,通过从离散时间区间时序逻辑到时间正则图的构造,提出了基于该逻辑的判定算法,该算法可以推广到其它的时序逻辑模型检查,并优于现有的基于自动机的时序逻辑判定方法.

关键词: 模型检查; 离散时间区间时序逻辑; 时间正则图; 可满足性判定
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2010) 05-1039-07

On the Decidability of Satisfiability of Discrete TITL Formulae

ZHU Wei-jun[1,2], ZHANG Hai-bin[1], ZHOU Qing-lei[2]
(1. *School of Computer Science, Xidian University. Xi'an, Shaanxi* 710071, *China*;
2. *School of Information Engineering, Zhengzhou University, Zhengzhou, Henan* 450052, *China*)

Abstract: There is no method available for checking whether a model satisfies a property described by a timed interval temporal logic (TITL) formula. In this paper, we formalize a novel model named timed normal form graph (TNFG). In addition, we give a procedure to construct TNFG from discrete TITL formula, and prove that the satisfiability of discrete TITL is decidable. The new method can be also used for other types of temporal logic and it is superior to the existing approaches.

Key words: model checking; discrete timed interval temporal logic; timed normal form graph; satisfiability checking

1 引言

为了验证网络协议与硬件系统的性质,模型检查技术近年被广泛研究与使用.在实时领域,Alur 提出的时间自动机(Timed automata, TA)[1]已经成为建立模型的事实标准,各种实时逻辑则被提出并用来描述系统的实时性质[2,3].不幸的是,稠密时间域上的实时逻辑的可满足性是不可判定的,因此不可用来模型检查.解决的办法有多种,比如不允许逻辑描述准点性质[2,5],不允许时间约束在非算子下出现[6].然而,这样的限制降低了逻辑的表达能力,导致一些实时性质不能被描述.多位研究者寻求更优的解决,结果表明,离散时间域上的实时逻辑可以满足实时性质验证的需要[6,7].

经典实时逻辑是建立在基于连续点的语义基础上,公式只能在孤立的点上被满足,而基于区间的实时逻辑[9,10,12]的优点在于可以描述整个区间的性质.然而,已有的实时模型检查算法均遭遇到状态空间爆炸问题,成为困扰研究人员的难题.这也导致了寻找逻辑规范可执行子集的努力,时间区间时序逻辑[9](Timed interval temporal logic, TITL)的时序版 EITL(Extend interval temporal logic)具有一个规范可执行的子集 Extend tempura 语言[4,8,14],可通过执行规范的方法验证时序性质.目前的 tempura 语言已在验证领域取得巨大成功[8,14],我们的研究正在把 tempura 语言扩展到实时领域,这样,*TITL* 的子集将在实时区间领域成为第一个规范可执行的逻辑.这是用 *TITL* 逻辑实时区间验证的潜在优势.

2 离散时间区间时序逻辑 $TITL_N$ 语法

TITL 是对不含 *Chopstar* 操作的 *EITL*(本文称之为 *UTITL*)的实时扩展[9].

定义 1 状态 s 是一个二元组 (a,t),其中 a 是命题集 P 在 $\{true, false\}$ 上的映射,它的规范形式是一个集合 $a_P=\{x \mid x\in P \wedge x = true\}$, t 是一个映射 $t: T\rightarrow N$,表示当前状态的时间.

收稿日期:2009-06-09;修回日期:2009-09-24
基金项目:国家 863 高技术研究发展计划(No.2007AA010408);河南省重大科技攻关计划(No.092101220104);陕西省科技攻关计划(No.2009K01-36)

定义 2 时间状态序列定义为：$\sigma = < s_0, s_1, \cdots, s_i, \cdots >$，其中 s_i 为状态.

定义 3 $TITL_N$ 公式构造

(1) $t ::= T \mid T_f$

(2) $\delta ::= T_f \leqslant a \mid T_f < a \mid T_f > a \mid T_f \geqslant a \mid \delta_1 \wedge \delta_2$

(3) $\varphi ::= p \mid \delta \mid skip \mid \varphi_1 \vee \varphi_2 \mid \neg \varphi \mid \varphi_1 ; \varphi_2$

定义 4 $TITL_N$ 公式缩写

(1) $p \wedge q ::= \neg(\neg p \vee \neg q)$ (2) $p \rightarrow q ::= \neg p \vee q$

(3) $Op ::= skip ; p$ (4) $\Box p ::= \neg \Diamond \neg p$

(5) $more ::= Otrue$ (6) $empty ::= \neg more$

(7) $p ;_I q ::= (p \wedge T_f \in I) ; q$

定义 5 $TITL_N$ 公式的解释是 $I = (\sigma, i, k, j)$，其中 σ 是一个在 $< s_i, \cdots, s_k, \cdots, s_j >$ 上的时间状态序列，$i, k, j \in N$，s_k 为当前状态，$len(\sigma) = j - i$ 为区间的时序长度，$len_t(\sigma) = s_t(j) - s_t(i)$ 为区间的时间长度，其中 $s_t(i)$ 为状态 s_i 对应的时间.

3 $TITL_N$ 语义

定义 6 令 $c \in N$，令 $s_p(k)$ 表示 $p \in AP$ 在状态 s_k 下的真值，$TITL_N$ 语义定义为：

Term: (1) $T = s_t(k)$ (2) $T_f = s_t(j) - s_t(k)$

Timed formula:

(1) $I \mid = T_f \leqslant C$ 当且仅当 $s_t(j) - s_t(k) \leqslant C$

(2) $I \mid = T_f \geqslant C$ 当且仅当 $s_t(j) - s_t(k) \geqslant C$

(3) $I \mid = T_f < C$ 当且仅当 $s_t(j) - s_t(k) < C$

(4) $I \mid = T_f > C$ 当且仅当 $s_t(j) - s_t(k) > C$

(5) $I \mid = \delta_1 \wedge \delta_2$ 当且仅当 $I \mid = \delta_1$ 且 $I \mid = \delta_2$

State and temporal formula:

(1) $I \mid = p$ 当且仅当 $s_p(k) = true$

(2) $I \mid = \neg \varphi$ 当且仅当 $I \mid \neq \varphi$

(3) $I \mid = \varphi_1 \vee \varphi_2$ 当且仅当 $I \mid = \varphi_1$ 或 $I \mid = \varphi_2$

(4) $I \mid = skip$ 当且仅当 $len(\sigma) = 1$

(5) $I \mid = \varphi_1 ; \varphi_2$ 当且仅当 $\exists r, k \leqslant r \leqslant j$，使得 $(\sigma, i, k, r) \mid = \varphi_1$ 且 $(\sigma, i, k, r) \mid = \varphi_2$

4 正则形和正则图(Normal Form Graph, NFG)

我们用 $Untime(P)$ 表示 $TITL_N$ 公式 P 去掉时间约束所得到的区间时序公式 $UTITL$. 文献[13]给出了把 $UTITL$ 公式 P 转换成正则形公式的算法，以及生成公式 P 正则图的 $NFG(P)$ 算法、对正则图 G 化简的算法 $SIMPLIFY(G)$、在化简图上判定公式 P 的算法 $CHECK(P)$. 篇幅所限，这里只给出结论：

定理 1[13] 任意公式 $R \in L_{UTITL}$ 都可转化成正则形

定理 2[13] 对任意公式 $R \in UTITL$，$\exists k \in N$，使得 $\mid CL(R) \mid = k$

定理 3[13] 对公式 R 的正则图 G，R 有穷(无穷)可满足当且仅当 G 中存在有穷(无穷)路径

5 时间正则图(Timed Normal Form Graph, TNFG)

定义 7 对一个 $TITL_N$ 公式 P，$TNFG$ 是一个三元组 $G = (CL(P), EL(P), C)$，其中，C 为时钟集，顶点集合 $CL(P)$ 及边集合 $EL(P)$ 归纳定义如下：

(1) $Untime(P) \in CL(P)$

(2) 对任意 $Q \in CL(P) \setminus \{\varepsilon, false\}$，如果 $Q \equiv Q_e \wedge empty \vee V_{i=1}^{r}(Q_i \wedge OQ'_i)$，则 $\varepsilon \in CL(P)$，$(Q, Q_e, \varepsilon, \delta, \lambda) \in EL(P)$，对任意 $i, 1 \leqslant i \leqslant r$，$Q'_i \in CL(P)$，$(Q, Q_i, Q'_i, \delta, \lambda) \in EL(P)$，其中 λ 为清零时钟集，δ 为时钟约束.

(3) $CL(P)$ 和 $EL(P)$ 仅由(1)、(2)产生，$G = (CL(P), EL(P), C)$.

定义 8 对 $TITL_N$ 公式 P 定义如下：$Unext(P) \triangleq \{$所有不在 P 的 O 域中出现的 P 的子公式$\}$，$Sub(P)$ 定义为 P 的所有子公式，$Unchop(P)$ 是一布尔变量，$Unchop(P) = true$ 当且仅当 P 中不含 $chop$ 子区间 $Unot(P) \triangleq \{$所有不在 P 的 $\neg$ 域中出现的 P 的子公式$\}$

定义 9 对 $TNFG$ 中结点 N，$pre(N)$ 定义为 N 的所有前驱结点.

定义 10 设 I 为时钟约束，$\bar{I}$ 为 I 的补，对 $TITL_N$ 公式 P，$\bar{I}(Q)$ 定义为 Q 中主 $CHOP$ I 操作 $(M ;_I N)$ 由 $CHOP$ $\bar{I}$ 操作 $(M ;_{\bar{I}} N)$ 替代得到的时间公式，称之为 P 的时间补公式 $\bar{P}$. 显然，$\bar{P} \in TITL_N$

算法 1 由 $TITL_N$ 公式 P 生成 $TNFG$ 的算法

```
TNFG(P)
Begin
    SIMPLIFY(NFG(Untime(P)));
    CL(P):=CL(Untime(P)); EL(P):=EL(Untime(P));
            /* 生成不含时钟约束的正则图
    T:=φ/* 已经由 NFG 结点直接转换为 TNFG 结点的临时集合 T
    for all Qs of CL(P) in the order of building NFG's
nodes do
    T:=T∪{Q}
    for all M;N∈Unext(Q)∩Unot(Q) and Unchop(M) and M;N∉
Sub(t∈T)
    and M;_I N∈Sub(P),do /* 对满足条件的含时钟约束的结点
        的所有约束
    new(x);/* 每一约束分配一时钟
for all EL(Q)=(Q,Qe,ε),do EL(Q):=(Q,Qe,ε,I_{x=0},{x})
    /* I_{x=0}表 x=0 时满足约束 I,其中 I=δ
    for all EL(Q)=(Q,Qi,Q'i),do EL(Q):=(Q,Qi,Q'i,true,
        {x})
/* 开始计时,NFG 的边变成 TNFG 的边
    for all EL(pre(w∧N)) do /* 当前区间结束,其中 w 为状态公
        式,可为空
```

```
    if EL(pre(w ∧ N)) = (R, Ri, R'i) then EL(pre(w ∧ N)) := (R,
Ri, R'i, Ix, {x})
/* 转换满足时钟约束,NFG 的边变成 TNFG 的边
    end for
  end for
  for all ¬(M; N) ∈ Unext(Q) and Unchop(M) and ¬(M; IN) ∈ Sub
    (P) do
      /* 时钟约束在 ¬ 辖域内出现
      CL(P) := CL(P) ∪ CL(TNFG(Ī(Q)))
    /* ¬(M; N)的 TNFG 上添加 M; IN 结点
      EL(P) := EL(P) ∪ EL(TNFG(Ī(Q)))
    /* 此处添加 M; IN 的 TNFG 边
    end for
  end for
    for all of e in EL(P) do /* 对所有不含时钟约束的转换
    if      e = (Q, Qi, Q'i)                    then
e := (Q, Qi, Q'i, true, ϕ) /* NFG 的边变成 TNFG 的边
    end for
end TNFG
```

定理 4　对任意公式 $R \in L_{TITL_N}$, $\exists k \in N$,使得 $|CL(R)| = k$

证明　考查 *TNFG*(*R*),首先生成 *Untime*(*R*)的 *NFG*.由定理 2,令 $|CL_{NFG}(Untime(R))| = k_1$,其中 $k_1 \in N$.从 *TNFG* 算法可知:

(1)若 *R* 中 ¬ 操作辖域内不含 *CHOP I* 操作(;$_I$),则 $|CL_{TNFG}(R)| = |CL_{NFG}(Untime(R))| = k_1$,令 $k = k_1$,命题得证.

(2)若 *R* 中 ¬ 操作的辖域内含 *CHOP I* 操作,则有:$|CL_{TNFG}(R)| - |CL_{NFG}(Untime(R))| = |CL_{TNFG}(\bar{I}(Q_1))|$ 上式去掉了最外层的 ¬ 操作.若 Q_1 中仍有 ¬ 操作的辖域内含 *CHOP I* 操作,则有 $|CL_{TNFG}(\bar{I}(Q_1))| - |CL_{NFG}(Untime(\bar{I}(Q_1)))| = |CL_{TNFG}(\bar{I}_2(Q_2))|$

……

由于 *CHOP I* 操作的嵌套为有穷,因此必存在 $n \in N$,使得

$|CL_{TNFG}(\bar{I}_{n-1}(Q_{n-1}))| - |CL_{NFG}(Untime(\bar{I}_{n-1}(Q_{n-1})))| = |CL_{TNFG}(\bar{I}_n(Q_n))|$

且 $|CL_{TNFG}(\bar{I}_n(Q_n))| = |CL_{NFG}(\bar{I}_n(Q_n))| = k_n$

$\therefore |CL_{TNFG}(R)| = k_0 + k_1 + \cdots + k_n = k \in N$

□

6　$TITL_N$ 公式可满足性判定算法

在 *TNFG* 上判定 $TITL_N$ 公式 *P* 的可满足性.算法如下.(其中 N_x 表示 *TNFG* 中 *x* 开始计时的结点,即含 *x* 清零边的出结点,N_{I_x} 表示时钟约束 I_x 应被满足的结点集,即含 I_x 的边的出结点组成的集合.)

算法 2　基于 *TNFG* 的 $TITL_N$ 公式可满足性判定算法

```
function1 (x, a, Nx, N_Ix) /* 判定时钟约束 Ix = (x < a)是否可被满
足
begin
    A := Nx; B := N_Ix; /* 考查图中 A 与 B 中点之间有界可达性,
从 A 开始广度优先搜索
    p := A; S := {A}; flag := 0; n := 0; /* S 为当前搜索层所有
结点的集合
    while flag = 0 and n ≤ a - 1 do /* 在 a - 1 步之内寻找 B 中结
点
      T := ∅
      for all p ∈ S do /* 当前层所有结点 p
       if p ∈ B then flag := 1 /* 若找到,则标记
       S := S - {p} /* 从当前层去掉结点 p
       for all i do /* p 的所有子结点 q
        q := post_i{p}; T := T ∪ {q}; /* 添加下一层结点 q 到 T
      end for
        end for
        S := T
        n := n + 1; /* 步长加一,考查下一层结点
        end while
        if flag then return true else return false /* 若有标记,则找到
两点间距离 a - 1 步之内路径
end function1
function2 (x, a, Nx, N_Ix) /* 判定时钟约束 Ix = (x = a)是否可被满
足
begin
A := Nx; B := N_Ix; p := A; S := {A}; flag := 0; n := 0;
    while flag = 0 and n ≤ a do /* 在 a 步之内寻找 B 中结点
      T := ∅
      for all p ∈ S do /* 当前层所有结点 p
      if p ∈ B then flag := 1 /* 若找到,则标记
        S := S - {p} /* 从当前层去掉结点 p
        for all i do /* p 的所有子结点 q
        q := post_i{p}; T := T ∪ {q}; /* 添加下一层结点 q 到 T
    end for
  end for
   S := T
      n := n + 1; /* 步长加一,考查下一层结点
    end while
if flag then return true else return false /* 若有标记,则找到两点间距
离 a 步之内路径
end function2
function3 (x, a, Nx, N_Ix) /* 判定时钟约束 Ix = (x > a)是否可被满
足
begin
    return true
end function3
function4 (x, a, b, Nx, N_Ix) /* 判定时钟约束 Ix = (x ∈ (a, b))是
否可被满足
```

begin
return function1(x, b, N_x, N_{I_x})
end function4
function5 (x, a, b, N_x, N_{I_x}) /* 判定时钟约束 $I_x = (x \in (a, b])$是否可被满足
begin
return
function1(x, b, N_x, N_{I_x}) ∨ function2(x, b, N_x, N_{I_x})end function5
function6 (x, a, b, N_x, N_{I_x}) /* 判定时钟约束 $I_x = (x \in [a, b))$是否可被满足
begin
return function1(x, b, N_x, N_{I_x})
end function6
function7 (x, a, b, N_x, N_{I_x}) /* 判定时钟约束 $I_x = (x \in [a, b])$是否可被满足
begin
return
function1(x, b, N_x, N_{I_x}) ∨ function2(x, b, N_x, N_{I_x}) *end* function7
Decision Procedure *CHECKT*(P) /* 在 *TNFG* 上判定 $TITL_N$ 公式的可满足性
(1) *CHECK*(*Untime*(P));
(2)Build the *TNFG* of P, $G = (CL(P); EL(P), C)$, by algorithm *TNFG*;
(3) $b'' :=$ *false* /* 一个布尔变量描述是否图中时钟约束可满足
for all π in *TNFG*(P) do /* 考查所有路径 π
$b_\pi :=$ *true* /* 一个布尔变量描述路径 π 中是否所有时钟约束可满足
for all x in π do /* 考查路径 π 所有时钟约束
/* 以下 if 语句由时钟约束不同形式求是否可满足
if $I_x = (x < a)$ then
$b_x :=$ function1(x, a, N_x, N_{I_x})
if $I_x = (x = a)$ then
$b_x :=$ function2(x, a, N_x, N_{I_x})
if $I_x = (x > a)$ then
$b_x :=$ function3(x, a, N_x, N_{I_x})
if $I_x = (x \leqslant a)$ then
$b_x :=$ function1(x, a, N_x, N_{I_x})
∨ function2(x, a, N_x, N_{I_x})
if $I_x = (x \geqslant a)$ then
$b_x :=$ function2(x, a, N_x, N_{I_x})
∨ function3(x, a, N_x, N_{I_x})
if $I_x = (x \in (a, b))$ then
$b_x :=$ function4(x, a, b, N_x, N_{I_x})
if $I_x = (x \in (a, b])$ then
$b_x :=$ function5(x, a, b, N_x, N_{I_x})
if $I_x = (x \in [a, b))$ then
$b_x :=$ function6(x, a, b, N_x, N_{I_x})
if $I_x = (x \in [a, b])$ then
$b_x :=$ function7(x, a, b, N_x, N_{I_x})
$b_\pi := b_\pi \wedge b_x$
end for /* 若 b_π 真则 π 所有时钟约束可满足
for all π do
if b_π then $b'' :=$ *true*
end for /* 只要有一个路径的所有时钟约束可满足,即找到模型
end for
(4) if $b'' \wedge b'$ then P is satisfiable otherwise unsatisfiable. /* 若时钟约束可满足,且 *UTITL* 公式可满足,则 $TITL_N$ 公式可满足

分析 1 若 $TITL_N$ 公式 φ 可满足,则存在时间正则图 G,使得图中存在路径 σ,有 $\sigma \mid = \varphi$ 成立.

比较 $TITL_N$ 公式 φ 和 *UTITL* 公式 φ',可看出,φ 的时间约束只能存在于 $p\,;_I q$ 中.若 φ 可满足:

(1)如果 φ 中不含 $;_I$ 操作,则 $\varphi = \varphi'$.由定理 3 易知结论成立.

(2)如果 φ 中含 $;_I$ 操作,不失一般性,设 $(\sigma, i, k, j) \mid = p\,;_I q$.在正则图 G' 的 s_k 结点的出度边加 $x := 0$,表示 s_k 当前状态开始计时;若公式 p 在 s_k 的某子孙结点 s_r 满足 *empty*,则对 s_r 的所有出度边标上 δ,也就是说在 s_r 考查是否满足时钟约束.这样得到的即为时间正则图 G.

定理 5 对一个 $TITL_N$ 公式 φ,存在一个时间正则图 G,使得:若 φ 可满足,则至少存在图中一条路径 σ,有 $\sigma \mid = \varphi$ 成立;若 φ 不可满足,则对图中任意路径 σ,有 $\sigma \mid \neq \varphi$ 成立

证明 当前状态到下一时刻状态的 *reduction* 中,首先用正则形算法把 φ 的非时间公式化为正则形.

令 $(\sigma, i, k, j) \mid = \varphi$,采用结构归纳法证明:

(1)如果 $\varphi \in UTITL$,根据分析 1,命题得证.

(2)如果 $(\sigma, i, k, j) \mid = p\,;_I q$, $I \in [a, b]$, $a \in N$, $b \in N$,施归纳于位置点 l.

(a)令 $l = k$,构造图中结点 s_k,根据定理 1,把 $p\,;_I q$ 化成正则形,得 $(\sigma, i, k, j) \mid = p_e \wedge q_e \wedge empty \vee V_{j=1} p_e \wedge q_c^j \wedge Oq_n^j \vee V_{i=1} p_c^i \wedge O(p_n^i; q) \equiv p'_e \wedge empty \vee Vp'_{cj} \wedge Op'_{ni}$,构造图结点 ε,对每个 j,构造图结点 s_{k+1}^j,对每个 i,构造图结点 s_{k+1}^i,对正则形每一极小项,在 G 中添加转换规则(边),有 $EL := EL \cup \{(s_k, \varepsilon, p_e \wedge q_e, \delta(x = 0), x)\} \cup \{(s_k, s_{k+1}^j, p_e \wedge q_c^j, \delta(x = 0), x)\} \cup \{(s_k, s_{k+1}^i, p_c^i, true, x)\}$,其中,$s_k \mid = p\,;_I q$, $s_{k+1}^i \mid = p_n^i; q$, $s_{k+1}^j \mid = q_n^j$, $\delta(x = 0)$表示对 $x = 0$ 满足约束 δ.

(b)当前点为 l 时结构归纳.如果 $(\sigma, i, l, j) \mid = p\,;_I q = p_e \wedge q_e \wedge empty \vee V_{j=1} p_e \wedge q_c^j \wedge Oq_n^j \vee V_{i=1} p_c^i \wedge O(p_n^i; q)$,则 $(\sigma, i, l+1, j) \mid = empty$ 或 $(\sigma, i, l+1, j) \mid = q_n^j$ 或

$(\sigma,i,l+1,j)\models p_n^i;q$,构造图结点 ε,对每个 j,构造图结点 s_{l+1}^j,对每个 i,构造图结点 s_{l+1}^i,对正则形每一极小项,在 G 中添加转换规则,有 $EL:=EL\cup\{(s_l,\varepsilon,p_e\wedge q_e,\delta,x)\}\cup\{(s_l,s_{l+1}^j,p_e\wedge q_c^j,\delta,x)\}\cup\{(s_l,s_{l+1}^i,p_c^i,true,\varnothing)\}$. 如果,$(\sigma,i,l,j)\models q'\equiv p_e\wedge empty\vee V_{i=1}p_{ci}\wedge Op_{ni}$,其中 q'可由 q 做 *reduction* 得到,则构造方法如下:添加 G 中结点 ε, $EL:=EL\cup\{(s_l,\varepsilon,p_e,true,\varnothing)\}$,对 $\forall i$,添加 s_{l+1}^i, $EL:=EL\cup\{(s_l,s_{l+1}^i,p_{ci},true,\varnothing)\}$,其中 $s_l=\{\phi\}$, $s_{l+1}^i=\{p_{ni}\}$.

(c) $l=j$ 时或不能添加新结点时,构造停止.

(a)、(b)、(c)构造了$(\sigma,i,k,j)\models p;_I q$ 时间正则图.

(3)如果 $\varphi=Q\wedge p;_I q$, $Q\in UTITL$,只需把 $Q\wedge p;q$ 化成正则型,并按造(2)中方法对 $p;q$ 的极小项中包含 *empty* 的子孙结点添加时钟约束 δ,并对当前结点 x 清 0 即可.

(4)如果 $\varphi=p_1;_{I_1}q_1\wedge p_2;_{I_2}q_2=\varphi_1\wedge\varphi_2$,则把 $p_1;q_1\wedge p_2;q_2$ 化成正则形,考查 $p_1;q_1$ 的子孙结点,按照(2)中方法对 $p_1;q_1$ 的极小项包含 $p_1^e\wedge empty$ 的子孙结点添加时钟约束 δ_1,当前结点 $x_1:=0$, $x_1\in X$,同理对 $p_2;q_2$ 的极小项包含 $p_2^e\wedge empty$ 的子孙结点添加时钟约束 δ_2,当前结点 $x_2:=0$, $x_2\in X$.(此时极小项可能含两区间状态公式).

(5)如果 $\varphi=Q\vee p;_I q$,只需简单合并 Q 和 $p;_I q$ 的结点集和转换规则集即可.

(6)如果 $\varphi=\neg(p;_I q)$,则 $\varphi=\neg(p;q\wedge T_f\#c)=\neg(p;q)\vee T_f\overline{\#}c$,则 $\varphi=\neg(p;q)\vee((p;q)\wedge T_f\overline{\#}c\vee\neg(p;q)\wedge T_f\overline{\#}c)=\neg(p;q)\vee((p;q)\wedge T_f\overline{\#}c)$,其中对 $\#=\{<,>,=,\leqslant,\geqslant\}$, $\overline{\#}$ 表 $\#$ 的对偶,(即约束非成立).

(a)对 $\neg(p;q)$,可按照定理 3、构造正则图 G',使得 G'中任意路径π,有 $\pi\models\neg(p;q)$,按照分析 1 的方法,构造时间正则图 G,使得 G 中任意路径σ,有 $\sigma\models\neg(p;q)$.

(b)对$(p;q)\wedge T_f\overline{\#}c$,可按照本证明(1)~(5)的方法构造时间正则图 G.

(1)~(6)构造了时间正则图,针对结构它是完备的.显而易见 *TNFG* 在 *NFG* 上加上了时钟约束.在上述图中证明命题的结论.根据公式 φ 中 $\neg$ 辖域内是否包含 *CHOP I* 操作分成①、②两种情况讨论.

①如公式 φ 中 $\neg$ 辖域内不含 *CHOP I* 操作.

(Ⅰ)若 φ 可满足,则 $Untime(\varphi)$可满足.由定理 3,对 *NFG* 中路径σ',有 $\sigma'\models Untime(\varphi)$.在 *TNFG* 中对满足$\sigma\models\varphi$ 的所有σ,有,$\exists\sigma'$使得$\sigma'=Untime(\sigma)$且$\forall\delta_\sigma$有$\sigma\models\delta_\sigma$,其中 δ_σ 表示路径σ 上的时钟约束.满足这样条件的路径 σ 必存在,否则 φ 不可满足;

(Ⅱ)若 φ 不可满足,则有两种情况:(ⅰ) $Untime(\varphi)$不可满足.由定理 3,不存在 *NFG*,使得对其中某一路径 σ'有$\sigma'\models Untime(\varphi)$成立.由于 *TNFG* 是在 *NFG* 中添加时钟约束,因此不存在 *TNFG* 使得对其中某一路径 σ 有$\sigma\models\varphi$ 成立;(ⅱ) $Untime(\varphi)$可满足.由(Ⅰ)的证明,对 *NFG* 中路径σ'有$\sigma'\models Untime(\varphi)$.在 *TNFG* 中使得$\sigma'=Untime(\sigma)$成立的路径 σ 中,假设 σ 时钟约束 δ_σ 均可满足,则有 $\sigma\models\varphi$ 成立,这与 φ 不可满足矛盾.因此对 *TNFG* 任一路径σ,所有时钟约束 δ_σ 不可都满足.即对 *TNFG* 的任一路径σ 有$\sigma\models\neq\varphi$.

②如公式 φ 中 $\neg$ 辖域内含 *CHOP I* 操作对所有 $\neg(p;_I q)\in Sub(\varphi)$,则令 $\neg(p;q)\vee p;_I q$ 代替 $\neg(p;_I q)$在 φ 中所有出现,得到公式 φ'',显然有 $\varphi''=\varphi$.对 φ'',易用与①同样方法证明结论.

推论 1 任意公式 $P\in L_{TITL_N}$可满足当且仅当算法 2 判定其可满足.

7　一个例子

例 1　判定 $TITL_N$ 公式 $\varphi\equiv\bigcirc^2(\Box\bigcirc p;q)\vee(p;_\delta\Box\bigcirc q)$的可满足性,其中 $\delta\equiv 2<x<5$.判定过程如下:

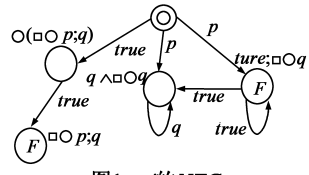


图1　φ'的*NFG*

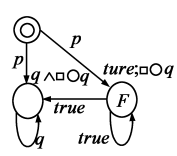


图2　φ'简化 *NFG*

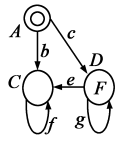


图3　φ的*TNFG*

a:　φ', C:　$q\wedge\Box\bigcirc q$, D:　$true;\Box\bigcirc q$,

b:　$(\varnothing,p,q\wedge\Box\bigcirc q,2<x<5,\{x\})$,

c:　$(\varnothing,p,true;\Box\bigcirc q,true,\{x\})$,

e:　$(true;\Box\bigcirc q,true,q\wedge\Box\bigcirc q,2<x<5,\varphi)$,

f:　$(q\wedge\Box\bigcirc q,q,q\wedge\Box\bigcirc q,true,\varnothing)$,

g:　$(true;\Box\bigcirc q,true,true;\Box\bigcirc q,true,\varnothing)$

(1)构造公式 $\varphi'\equiv\bigcirc^2(\Box\bigcirc p;q)\vee(p;\Box\bigcirc q)$的 *NFG*,如图 1.

(2)对 *NFG* 化简,如图 2.

(3)因为化简后不为空,所以 φ'可满足,由基于 *NFG* 的 *UTITL* 判定算法,$b':=true$.

(4)由算法 1,构造公式 φ 的 *TNFG*,如图 3,图中包括 A、C、D 三点和 b、c、e、f、g 五边.

(5)在 *TNFG* 中判定时钟约束是否可满足.由算法

2,有如下过程:对公式中唯一的时钟约束 φ,调用 function4,function4 又调用 function1(x, b, N_x, N_{I_x}),其中函数的参数为:$N_x = A$, $N_{I_x} = \{A, D\}$, $b = 5$. 当 $n = 0$ 时,有 $A \in N_{I_x}$,此时 $flag := 1$,找到两点间 4 步之内路径,找到满足约束 δ 的模型. 由算法 2,$b'' := true$,又因 b' 为真,所以 φ 可满足.

8 结论

我们提出了一种新模型——时间正则图,并提出 $TITL_N$ 公式转换为时间正则图的算法、$TITL_N$ 公式的判定算法. 已有的实时区间逻辑判定,需要从逻辑到自动机的转换,这些基于逻辑公式复杂结构的转换方法非常复杂,时间复杂度为非初等[2,11,12]. 而我们方法的时间复杂度虽然也是非初等(问题固有复杂度下限),但"逻辑非"算子包含区间中,新方法复杂度随原子命题出现个数 $|\varphi(AP)|$ 的指数增长,而自动机方法复杂度则随公式长度 $|\varphi|$ 的指数增长,$|\varphi(AP)| \ll |\varphi|$,且多次"逻辑非"算子包含区间导致了复杂度指数阶的递增,因此对区间逻辑判定算法非初等时间复杂度 $a^{n\cdot\cdot^{\cdot}}$ 的底数 a,新方法($a = O(2^{|\varphi(AP)|})$)远小于现有的自动机方法($a = O(2^{|\varphi|})$). 同时新方法避免了从图到自动机的转换,进一步提高了算法效率,它虽是基于区间实时逻辑 *TITL* 提出来,也可用于 *TPTL*[3]、*MITL*[2]等主流非区间实时逻辑或时段演算[10].

本工作提出了 *TITL* 逻辑判定方法,如用 $TITL_N$ 公式描述实时系统区间性质,用时间正则图或时间自动机建立模型,就可模型检查 $TITL_N$ 性质.

除时间复杂度之外,线性实时逻辑模型检查面临另一主要问题是状态空间爆炸. 新方法降低了生成图结点数和边数. 图 4 和图 5 对公式$\bigcirc\bigcirc_{x\leqslant 3}p$ 两种判定方法比较,*TABLEAUX* 生成时间图产生了 12 个结点 12 条边,*TNFG* 生成图仅产生 5 个结点 6 条边. 现有方法构造自动机只有用基于公式结构子集、公式分型或对逻辑进行强约束特殊限定的技术[2,11],而 *TNFG* 方法生成新结点只有 *next* 算子产生,这样就避免了大量冗余结点的产生,因而缓解了状态空间的指数增长,特别适合对一般长公式的判定.

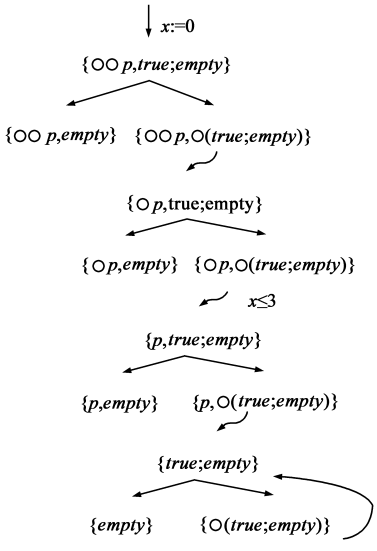


图4 TABLEAUX生成公式$\bigcirc\bigcirc_{x\leqslant 3}p$的时间图

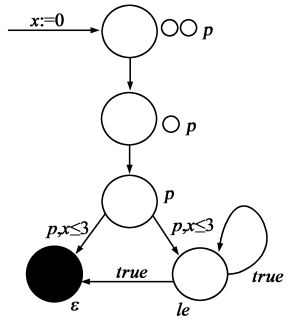


图5 公式$\bigcirc\bigcirc_{x\leqslant 3}p$的*TNFG*

参考文献:

[1] Alur R, Dill D L. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2): 183 – 236.

[2] Alur R, Feder T, Henzinger T A. The benefits of relaxing punctuality[J]. Journal of the ACM, 1996, 43(1): 116 – 146.

[3] Alur R, Henzinger T A. A really temporal logic[J]. Journal of the ACM, 1994, 41(1): 181 – 204.

[4] Duan Z, Koutny M. A framed temporal logic programming language[J]. Journal of Computer Science and Technology, 2004, 19(3): 314 – 351.

[5] Alur R, Henzinger T A. Logics and models of real time: A survey[A]. LNCS600[C]. Berlin: Springer-Verlag, 1992. 74 – 106.

[6] Wilke T. Specifying timed state sequences in powerful decidable logics and timed automata[A]. LNCS863[C]. Berlin: Springer-Verlag, 1994. 694 – 715.

[7] Thomas A, Henzinger T A, Manna Z, Pnueli A. What good are digital clocks? [A]. In Proc. ICALP'92, LNCS623[C]. Heidelberg: Springer, 1992. 545 – 558.

[8] Duan Z, Yang X, Koutny M. Framed temporal logic programming[J]. Science of Computer Programming, 2008, 70(1): 31 – 61.

[9] Duan Z. Modeling of Hybrid Systems[M]. Beijing: Science Press, 2004. 1 – 43, 105.

[10] Zhou C, Hoare C A, Ravn A P. A calculus of duration[J]. Information Processing Letters, 1991, 40(5): 269 – 276.

[11] Li G, Tang Z. Translating a continuous-time temporal logic into timed automata[A]. Proceedings of the first Asian Symposium on Programming Languages and Systems (APLAS

2003), LNCS2895[C]. Berlin: Springer-Verlag, 2003. 322 – 338.

[12] Hanson M R. Model – checking discrete duration calculus[J]. Formal Aspects of Computing, 1994, 6A: 826 – 845.

[13] Duan Z, Tian C, Zhang L. A decision procedure for propositional projection temporal logic with infinite models[J]. Acta Informatica, 2008, 45(1): 43 – 78.

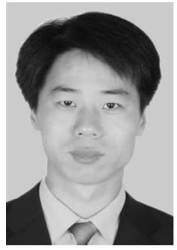
[14] Duan Z. Temporal logic and temporal logic programming [M]. Beijing: science press, 2005. 1 – 175.

[15] 张建民,沈胜宇,李思昆.最小布尔不可满足子式的求解算法[J].电子学报,2009,37(5):993 – 999.
Zhang Jian-min, Shen sheng-yu, Li Si-kun. Algorithms for deriving minimum unsatisfiable boolean subformulae[J]. Acta Electaonica Sinica, 2009, 37(5): 993 – 999. (in Chinese)

作者简介:

朱维军　男,1976 年生于河南郑州,博士研究生、讲师,研究方向:模型检查、时序逻辑与自动机、实时系统. E-mail: zhuweijun76@163.com

周清雷　男,1962 年生于河南新乡,教授、博士生导师,研究方向:实时系统、形式语言与自动机、模型检查、网络安全.



张海宾　男,1982 年生于山东菏泽,博士、讲师,研究方向:实时、混合系统,形式化方法.