普通のUntil演算子を持つ命題実時間論理について*

4J－4

鈴木康人† 米崎直樹‡
†北陸先端科学技術大学院大学 情報科学研究科
‡東京工業大学 情報理工学研究科

1 はじめに

実時間論理は時間論理の一種であり、リアルタイムシステムの検証や合成に用いるため、研究がなされている。時間論理が注目するのは、イベントの起こる順序であって具体的な時間ではない。それに対し、実時間論理は具体的な時刻に注目した論理である。ここでは、通常の弱いUntil演算子と実数時刻の付けられた連続時間領域と可算無限以下の時区間を持つ実時間論理の検証法について報告する。

2 これまでの実時間論理研究について

2.1 分類

R.AlurとT.A.Henzingerは、既存の様々な実時間論理の研究を時間モデルの構造と論理式の構文から分類している [AH92]。

モデル時間構造に注目してより詳細に分類すると以下のようになる。ただし、BT,FQ,*はそれぞれ、Bounded Temporal Oprator論理、Freeze Quantifier論理、当稿で紹介する論理であるものとする。

時間の定義域	時間構造			
	濃度	階層性	線型時間	分岐時間
整数	離散	有	MTL TPTL RTTL XCTL	RTCTL
実数	連続	有	MITL BT FQ *	
	連続	無		TCTL

濃度については離散、稠密、連続があり、それぞれについて区間の階層性を考えるか否かで更に分類できる。また区間の濃度や区間の階層性の深さで分類することが出来るが、これまでの論理の時間モデル構造は定義域が整数である場合、整数を離散の性質を持つものとして扱い、区間を可算無限以下で一階層だけ入れたもの(例えば、TPTL)であり、定義域が実数である場合、実数を連続の性質を持つものであるとして扱い、区間を可算無限以下で一階層を入れたもの(例えば、BT)であるか、もしくは無階層のもの(例えば、TCTL)の場合でしかない。

これらの論理の決定手続きは、時間の定義域が整数のものに分類されている論理については全て存在し、また、実数のものではMITL,TCTLに決定手続きが存在している。

2.2 論理の不自然さ

文献 [AH92] で紹介されているBounded Until Operator論理(以後、BT)を例にとって、その意味論の不自然さを説明しよう。BTの構文は、pを命題変数、Iを有理数を端点とする区間とするとき$\phi = p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_I \phi_2$である。$\mathcal{U}_I$の直感的な意味は現時点から見て、相対区間$I$の中で$\phi_2$が必ず存在し、その時点まで$\phi_1$が成立するというものである。この$\mathcal{U}_I$の形式的な定義は$[(\rho,t) \models \phi_1 \mathcal{U}_I \phi_2 \Leftrightarrow (\rho,t') \models \phi_2$ for some $t' \in t+I, (\rho,t'') \models \phi_1$ for all $t < t'' < t']$である。この定義によると現時点tから見て将来で$t+I$と共通部分を持つ左に開であるような$\phi_2, \neg\phi_1$が成立する区間の手前の区間までϕ_1が成立しているときでも$\phi_1 \mathcal{U}_I \phi_2$が真でないことになる。

3 普通のUntil演算子付き命題実時間論理

当研究においては、Until演算子に関しては区間で解釈を行なうようにして、以上の不自然さを解消した、BTと同等の表現力を有する論理を与える。

この論理の構文は、pを命題変数、rを実数時間とするとき$\phi = p|\neg\phi|\phi_1 \wedge \phi_2|\phi$ at $r|\phi$ for $r|\phi_1 \mathcal{U} \phi_2$である。これらの直感的な意味は、at演算子、for演算子については、それぞれ時点で解釈され、ϕが現時点からr時間の後に成立する、r時間の間成立することである。$\mathcal{U}$演算子は、ϕ_2が成立する区間までϕ_1が成立し続けることを意味する。$\bot, \top, \vee, \rightarrow, \leftrightarrow, \Box, \Diamond$は上の表現から組み合わされるものとする。

形式的には以下のような意味論を定義した。ρは一つ

*Propositional Real-time Logic with Normal Until-operator

†Yasuhito X-Suzuki. School of Information Science, Japan Advanced Institute of Science and Technology,Hokuriku.
‡Naoki Yonezaki. Graduate School of Information Science and Engineering, Tokyo Institute of Technology.

の時区間に属する時刻に対し同一の割当を行なう意味関数とする。V_t^sはρと区間列Iとの組で表される構造sと時間t、論理式を受けて、真理値と区間の列の組を返す関数であり、SVはそういった組の真理値の方をSIは区間列の方を選択する関数である。

$V_t^s(p) = (p \in \rho(t), \bar{I})$
$V_t^s(\neg A) = (not(SV(V_t^s(A))), SI(V_t^s(A)))$
$V_t^s(A \wedge B) = (SV(V_t^s(A)) \text{ and } SV(V_t^s(B)), \bar{I}')$
$V_t^s(A \text{ at } a) = (SV(V_{t+a}^s(A)), \bar{I}'')$
$V_t^s(A \text{ for } a)$
$= (\forall t'.(t < t' < t + a \Rightarrow SV(V_{t'}^s(A))), \bar{I}'')$
$V_t^s(A \,\mathcal{U}\, B)$
$= (\forall t'.(t \in I'_i \text{ and } t < t'$
$\Rightarrow (SV(V_{t'}^s(A)) \text{ and } SV(V_{t'}^s(\neg B))))$
$\text{or } \exists j.(t \in I'_i \text{ and } I'_i \leq I'_j \text{ and}$
$\forall t'.(t' \in I'_j \Rightarrow SV(V_{t'}^s(B))) \text{and} \forall k.(i \leq k < j \Rightarrow$
$\forall t''.(t < t'' \text{ and } t'' \in I'_k \Rightarrow SV(V_{t''}^s(A))))), \bar{I}')$

ただし、$\bar{I}'$は、二つの論理式A, Bの区間列を重ねて細分した区間からなる区間列、$\bar{I}''$は論理式Aの区間列を時間aだけスライドさせた区間列と論理式Aの区間列を重ねて細分した区間からなる区間列のことである。

4 証明手続き−タブロー

4.1 展開規則

ここで与えられているタブローは、全ての時間論理式を、at ならびに for 演算子と時間定数c、時間変数xを用いた時間項tに書き直して展開する。時間項をT、時間項を用いた論理式の表現の有限集合をF、時間項の満たすべき条件式の集合を Cond としたとき、その三組$< F, T, \text{Cond} >$、区間をIとするときには、組$< F, I >$をノードとする。以下にタブローのα-rule,β-rule を示す。

[α-Rule]
$\neg\neg A \Rightarrow A : A \wedge B \Rightarrow A, B : A \text{ at } 0 \Rightarrow A$
$\neg(A \text{ at } t) \Rightarrow (\neg A) \text{ at } t : A \text{ for } 0 \Rightarrow \top$
$\neg(A \text{ for } 0) \Rightarrow \bot : \neg(A \text{ for } t) \Rightarrow \neg A \text{ at } c[c < t]$
$(A \wedge B) \text{ for } t \Rightarrow A \text{ for } t, B \text{ for } t$
$(\neg\neg A) \text{ for } t \Rightarrow A \text{ for } t$
$(\neg(A \text{ at } t)) \text{ for } t' \Rightarrow ((\neg A) \text{ at } t) \text{ for } t'$
$(A \text{ for } t) \text{ for } t' \Rightarrow A \text{ for } (t + t')$
$(A \text{ at } t) \text{ for } t' \Rightarrow (A \text{ for } t') \text{ at } t$

[β-Rule]
$\neg(A \wedge B) \Rightarrow \neg A; \neg B$
$A \,\mathcal{U}\, B \Rightarrow B; \neg B, B \text{ for } c; \neg B, (A \wedge \neg B) \text{ for } \infty;$
$\neg B, (A \wedge \neg B) \text{ for } c', B \text{ at } c'$
$\neg(A \,\mathcal{U}\, B) \Rightarrow \neg B, (\neg A \wedge \neg B) \text{ for } c;$
$\neg B, (\neg B) \text{ for } c', (\neg A \wedge \neg B) \text{ at } c'$
$\neg(A \wedge B) \text{ for } t \Rightarrow \neg A \text{ at } x[x < t]; \neg B \text{ at } x[x < t]$
$(\neg(A \text{ for } t)) \text{ for } t' \Rightarrow (\neg A) \text{ for } t'; A \text{ for } c,$
$\neg A \text{ at } c[t' < c < t]; A \text{ for } c', \neg A \text{ at } c',$
$((\neg(A \text{ for } t)) \text{for}(t' - c')) \text{at } c'[c' < t', c' < t];$
$(\neg A) \text{for } c'', ((\neg(A \text{ for } t)) \text{for}(t' - c'')) \text{at } c''$
$[c'' < t']$
$(A \,\mathcal{U}\, B) \text{for } t \Rightarrow (\neg(\neg A \wedge \neg B)) \text{ at } c,$
$(\neg(\neg A \wedge \neg B)) \text{ for } c,$
$(B \text{ for } (t - c)) \text{at } c[c < t];$
$\neg(\neg A \wedge \neg B) \text{ at } c', (\neg(\neg A \wedge \neg B)) \text{ for } c',$
$(A \text{ for } (t - c')) \text{at } c'[c' < t], (A \,\mathcal{U}\, B) \text{ at } t$
$(\neg(A \,\mathcal{U}\, B)) \text{ for } t \Rightarrow ((\neg A) \text{ for } (t - c)) \text{ at } c[c < t],$
$(\neg B) \text{ for } t; (\neg B) \text{ for } t, \neg B \text{ at } t,$
$(\neg(A \,\mathcal{U}\, B)) \text{ at } t$

α, β-rule の適用はノード$< F, T, \text{Cond} >$に対して行なわれ、$\Rightarrow$の左辺にある論理表現XがFの中に現れたとき、右辺の; の個数+1 だけ現ノードをコピーし、それぞれからXを除いた後; で区切られている論理表現の列をコピーされたノードに加えるものとする。その際、[ ] で囲まれた条件式は矛盾しない限り、Cond に加える。もし、矛盾した場合はその論理表現の列をノードに加えない。可能な限りα, β-rule を適用したその後、$A, \neg A$なる表現のないノードに今度は時間展開を行なう。時間展開とはノードに含まれる一番外側の時間項の中で極小の値tを選択し、Cond にその条件を書き加えた後で、現時点Tと$T + t$を端点とする区間と時点$T + t$からなるノードを現ノードからつなぐ規則である。その際、Cond の条件内で時間変数の具体的な値を定める。このようにして得た木構造の葉は常に時点からなるノードである。α, β-rule を可能な限り適用したとき、現時点の値を for 論理式の時間項に加えて同じ式集合を持ったノードが親にあるとき、ループは形成されるものとする。以上の規則で形成された有限有向グラフをタブローとする。

4.2 判定規則

$A, \neg A$を含んだノードを閉ノードとする。ループ内部のノードに∞を時間項として持つ論理式がないとき、ループから出る全てのノードが閉ノードであるときに、閉ループであるものとする。全ての子ノードが閉ノードないし閉ループであるようなノードは、やはり閉ノードであるものとする。葉ノードからルートに向けて閉ノードのチェックを行なったとき、ルートが閉ノードとなった場合は、Fは充足不能である。

このタブローによれば、Cond 中の不等式を満たす解の存在問題が決定可能であるとき、充足不能性が判定できる。

参考文献

[AH92] R.Alur and T.A.Henzinger,Logics and models of real time : a survey , In: *Proceedings of the REX Workshop*,Realt-time:Theory in Practice , Lecture Notes in Computer Science 600(Springer,Berlin,1992).