高度交通システム 1－5
（2000．5．24）

実時間システムのパラメータ制約自動導出およびITSへの応用

中田 明夫† 新子 浩康‡ 大場 充‡ 東野 輝夫†

†大阪大学大学院基礎工学研究科情報数理系専攻
‡広島市立大学情報科学部情報数理学科

本論文では、あるクラスの時間オートマトンモデルで記述された実時間システムが与えられた時相論理式で書かれた性質を満足するために必要十分な時間パラメータに関する制約を自動導出する手法 [5] を道路交通に関する例題に適用し、その有効性を評価する。[5] で提案したモデルは Alur らが提案した時間オートマトンモデルと異なり、ある状態からある状態への到達可能であるためのパラメータに関する条件を遷移条件の論理積で表現できるという特徴を持つ。満たすべき性質は時相論理式の一種である TCTL で記述する。TCTL によってある時間以内にある動作ができる状態に到達可能であるなどといった性質を記述できる。例題として、信号付き交差点を車がある時間以内に通過するための車の速度 (実際には、車が道路のある領域を通過するのにかかる時間) と信号が変わるタイミングに関する制約を導出する。

Symbolic Model Checking for Parametric Real-Time Systems and its Application to ITS

Akio Nakata† Hiroyasu Atarashi‡ Mitsuru Ohba‡ Teruo Higashino†

† Dept. of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University

‡ Dept. of Computer Science, Faculty of Information Sciences, Hiroshima City University

In this paper, we examine how a symbolic model checking method can be applied to ITS. Firstly, we present a method (which is a kind of symbolic model checking) for deriving the weakest condition for parameters of a behavioral real-time system specification (written in a state transition model) to satisfy a required property written in a temporal logic formula. The behavioral specification is written in A-TSLTS, a kind of state transition models proposed in [1]. The required property is written in TCTL(Timed Computation Tree Logic[2]). Then we model a car and a traffic light example using A-TSLTS. Finally, we derive the weakest condition between the car speed (actually the elapsed time for passing the specified area of the road) and the turning timing of the traffic light in order to satisfy a property such as "the car can always pass through the intersection within time t".

1 まえがき

信頼性の高い実時間システムの設計技法の一つとしてモデル検査 (有限状態機械で記述したシステムの動作仕様が時相論理式で記述した要求仕様を満足するか否かの検査) がある。従来のモデル検査技法では動作仕様及び要求仕様における設計パラメータを完全に指定した後、動作仕様が要求仕様を満たしているか否かを Yes/No で判定するものであった。しかし、設計の段階ではパラメータの値を試行錯誤で決めるのではなく、むしろ動作仕様が要求仕様を満足するようなパラメータの値に関する条件を求められる方が望ましい。そこで我々は [5] にて、時相論理式で記述された実時間システムの要求仕様と状態遷移モデルで記述された実時間システムの動作仕様が与えられ、双方にパラメータが記述されているとき、動作仕様が要求仕様を満足するための双方のパラメータ群に関する必要十分条件を自動的に導出する方法を提案した。

[5] では動作仕様の記述モデルとして A-TSLTS (Alternating Timed Symbolic Labelled Transition System) [1] を用いている。A-TSLTS は実時間システムを記述できるオートマトンモデルの一種である。A-TSLTS は有限個の状態を持ち、各状態はいくつかの状態変数 (パラメータ) を持っている。また、各遷移は時間経過による遷移 (時間遷移) または (入出力) 動作による遷移 (動作遷移) のいずれかである。時間遷移は経過時間を代入する変数 (時間変数) および遷移条件からなり、動作遷移は動作名および遷移条件からなる。各遷移条件は時間変数およびパラメータ変数を用いた線形不等式の論理結合で記述する。各状態は時間遷移のみが可能な状態 (休止状態) か動作遷移のみが可能な状態 (活動状態) のいずれかであるとし、休止状態からは活動状態へ、活動状態からは休止状態に交互に遷移するとする。A-TSLTS は Alur らが提案した時間オートマトン [3] と異なり、状態 s から状態 s' へ到達するための状態 s のパラメータに関する条件が s から s' までの遷移条件の論理結合で表現できるという特徴がある。

要求仕様を記述する実時間時相論理としては TCTL(Timed Computation Tree Logic)[2] を採用する。TCTL は例えば「ある動作 a の実行後 t 秒

以内にある動作 b が実行可能にならなければならない」などといった性質(要求)を記述することができる論理である。文献 [5] ではモデル A-TSLTS の特徴を生かし、TCTL の各構文要素 f と閉路を持たない(=木状の) A-TSLTS の状態 s の組 (s, f) から s が性質 f を満たすための s の状態変数(パラメータ)に関する条件 $PT(s, f)$ を再帰的に求めるアルゴリズムを提案している。閉路をもつような(繰り返し動作を含む)例題に対しても、それが一定の動作後に常に初期状態に戻るような周期的な動きをするものであれば、初期状態へ戻る遷移を削除した、閉路を持たないモデルに変換することでパラメータ条件を導出できる。

本研究では [5] の手法を交通信号の例題に適用し、ITS の分野への応用の可能性を検討する。交通信号の例題は周期的であるため、1 周期分の動作を抽出した閉路を持たないモデルで表現すれば、[5] の手法が適用可能である。また、この例題は車の速度、信号の変化するタイミングなど多くの時間パラメータを持つため、指定した条件(最低何秒以内に交差点を通過できればよいかなど)を満足するパラメータ条件を導出できれば、交通量に応じた信号タイミングの動的変更や、カーナビゲーションシステムとの組合せによる信号タイミングに応じた最短経路選択などの応用に役立つと思われる。

本研究ではモデル化の対象となる道路を一定の領域に分割し、自動車の状態をその自動車が現在いる領域の名前で表すことにする。さらに、信号機も有限状態機械でモデル化し、自動車と信号機の状態の組を系全体の状態と見なすことで、道路交通をモデル化する。パラメータとしては自動車が各領域に進入してから離脱するまでの時間(=その領域の長さ/自動車の速度)および、信号機の状態(色)が変化する時間を持つとする。さらに、そのモデルが満たすべき性質を TCTL で記述し、[5] の手法を適用することにより、自動車の速度と信号機のタイミングパラメータとの関係式を自動導出する。得られた結果には、自動車と信号機のどちらが制御対象であるかによって 2 通りの利用方法がある。信号機が制御対象であれば、観測された自動車の速度から、与えられた性質を満足できる信号タイミングを求めるために利用できる。一方、自動車が制御対象であれば、無線などで信号タイミング情報を受信し、その情報に応じて速度や経路を制御するための条件として利用できる。

本稿の構成は以下の通りである。まず、2 章では動作仕様の記述モデル A-TSLTS について説明する。3 章では要求仕様の記述言語である TCTL について説明する。4 章にて動作仕様の状態 s が TCTL の式 f を満足するためのパラメータに関する条件 $PT(s, f)$ を求めるアルゴリズムを提案する。5 章では交通信号の例題への適用例を示す。6 章で結論と今後の課題を述べる。

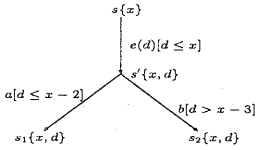

図 1: A-TSLTS の例

2 A-TSLTS モデル

動作仕様の記述モデル A-TSLTS[1] は以下のように定義される。

定義 2.1 $M = \langle S, Act, Var, DVar(), Pred, E, s_0 \rangle$ を **A-TSLTS** と定義する。ここで、S は状態名の有限集合、Act は動作名の集合、Var はパラメータ変数の集合、$DVar()$ は S から Var の部分集合への写像、$Pred$ は Var に属する変数を自由変数にもつ線形不等式の論理結合の集合、E は遷移関係で $S \times (Var \cup Act) \times Pred \times S$ の部分集合、$s_0 \in S$ は初期状態である。$d \in Var$ に対して $(s, d, P, s') \in E$ を**時間遷移**と呼び $s{-}e(d)[P]{\rightarrow}s'$ と記述する。$a \in Act$ に対して $(s, a, P, s') \in E$ を**動作遷移**と呼び $s{-}a[P]{\rightarrow}s'$ と記述する。A-TSLTS の各状態は**休止状態** (idle state) または**活動状態** (active state) のいずれか一方に属す。休止状態では時間遷移のみが実行可能で、活動状態に遷移する。活動状態では動作遷移のみが実行可能で、休止状態に遷移する。また、各状態にはパラメータ変数の集合 $DVar(s)$ が関連づけられている。以後、例えば $DVar(s) = \{x, y\}$ であるとき状態 s を $s\{x, y\}$ と書くことにする。$s{-}e(d)[P]{\rightarrow}s'$ は状態 s からある時間だけ経過したのち、状態 s' に遷移し、その経過時間を変数 d に代入するという遷移を表す。P は遷移条件で、d および s のパラメータ変数 $DVar(s)$ が満たすべき条件を線形不等式の論理結合で記述する。また、$s'{-}a[P]{\rightarrow}s_1$ は状態 s' のパラメータが遷移条件 P を満たすときに動作 a を実行可能で、状態 s_1 に遷移することを表す。休止状態からは時間遷移は一本のみ記述できるとする。活動状態からは複数の動作遷移で分岐することを許す。 □

例 2.1 図 1 に A-TSLTS の例を示す。図 1 において、遷移 $s\{x\}{-}e(d)[d \leq x]{\rightarrow}s'\{x, d\}$ は、s から x 以下の時間だけ経過して s' に遷移し、その経過時間が変数 d に代入されることを表す。また、遷移 $s'\{x, d\}{-}a[d \leq x-2]{\rightarrow}s_1\{x, d\}$ は s' のパラメータ d および x が $d \leq x-2$ を満たすとき、すなわち状態 s から $x-2$ 単位時間以内に a を実行可能であることを表す。遷移 $s'\{x, d\}{-}b[d > x-3]{\rightarrow}s_2\{x, d\}$ も同様である。 □

A-TSLTS で記述された動作仕様の具体的な動きは、初期状態に対してパラメータの値を具体的に決

めたときに初めて定まる。例えば、例 2.1 の状態 s を初期状態としたとき、この動作仕様は $x = 3.5$ のとき、$s(x = 3.5) -1.5\rightarrow s'(x = 3.5, d = 1.5)$ という動きが可能である。上の例では、$x = 3.5$ という値の割り当てにおいてある d の値 1.5 が存在し、遷移条件 $d \leq x$ を満たすので、1.5 という値の時間遷移を実行できて、状態 s' のパラメータ x, d をそれぞれ 3.5, 1.5 に割り当てた状態に遷移する。このような各パラメータ変数への値の割り当てを付値とよび ρ, ρ' などで表す。また、x に値 v を割り当て、それ以外の変数には ρ と同じ値を割り当てる付値を $\rho[x = v]$ と記述する。一階述語 P の各自由変数に付値 ρ の値を割り当てたときに真となることを $\rho \models P$ と記述する。s のパラメータの値を付値 ρ で決めたときの状態を状態 s と ρ の組 (s, ρ) で表し、s の ρ による具体的状態と呼ぶ。A-TSLTS 記述の具体的な動きは具体的状態の間の遷移関係として以下のように定義される。

定義 2.2 A-TSLTSM の各時間遷移 $s-e(d)[P]\rightarrow s'$ に対して具体的状態間の遷移関係を以下のように定める。

- $\rho \models \exists d[0 \leq d \wedge P]$ であるような任意の ρ および $\rho[d = t] \models [0 \leq d \wedge P]$ であるような任意の t に対して $(s, \rho)-t\rightarrow(s', \rho[d = t])$. さらに任意の t' $(0 \leq t' \leq t)$ に対して $(s, \rho)-t'\rightarrow(s', \rho[d = t'])-(t - t')\rightarrow(s', \rho[d = t])$. [時間の連続性より]

また、各動作遷移 $s-a[P]\rightarrow s'$ に対して具体的状態間の遷移関係を以下のように定める。

- $\rho \models P$ であるような任意の ρ に対して $(s, \rho)-a\rightarrow(s', \rho)$

このように定義された具体的状態における状態遷移システムを、M の**具体的遷移システム**と呼ぶ。 □

例 2.2 図 2 に単純な T 字路における交通信号と車の経路の関係をモデル化した例を示す。図において、A1～A4 は道路の領域に付けられた名前であり、車の状態は現在どの領域にいるかで表される。例えば西から東へ抜ける車の状態は A1 → A2 → A3 の順に遷移する。また、信号の状態は簡単のため西→東および西→南方向の交通に関係する信号の色のみを考慮し、B(青) → Y(黄) → A(右折矢印) → R(赤) の順で遷移するとする。系全体の状態を車の状態と信号の状態の対で表し、A-TSLTS モデルで記述したのが図 3 である。図において遷移条件および各状態のパラメータリストは省略している。このモデルの動きは以下の通りである。まず、初期状態 $(A1, B)$ から出発し、時間遷移によってある時間 (d_1 単位時間) 経過する。次に、信号の色が青から黄に変わるか (動作 by)、車が領域 A1 から A2 に移動するか (動作 1-2) のどちらかが起こる。前者であれば、状態 $(A1, Y)$ に遷移し、後者なら $(A2, B)$ へ遷移する。状態 $(A1, Y)$ 以降も同様に状態が遷移する。動作 ya は信号が黄から矢印へ変わる遷移を表し、動作 2-3 は車が領域 A2 から A3 へ進入する遷移を表す。動作 ar, 2-4 なども同様に定義される。これらの状態遷移には道路交通法による制約があり、例えば、状態 $(A1, R)$ からは車は領域 A2 へ進入できない (赤信号のため)。また、状態 $(A2, A)$ からは車は領域 A4 へのみ進入可能である (右折矢印のため)。さらに、右折にかかる最悪の場合の時間を考慮するため、右折矢印が出るまでは右折できないと仮定する。図 3 は木ではないため、さらに [5] が適用出来るように木に変換し、パラメータおよび遷移条件を記述したモデルを図 4 に示す。パラメータ p_1 は車が領域 A1 から A2 へ移動するのにかかる時間である。p_2 は同じく A2 から A3、p_3 は A2 から A4 へ移動するのにかかる時間である。p_4 は信号が青から黄へ変わるまでの時間、p_5 は黄から矢印、p_6 は矢印から赤に変わるまでの時間である。 □

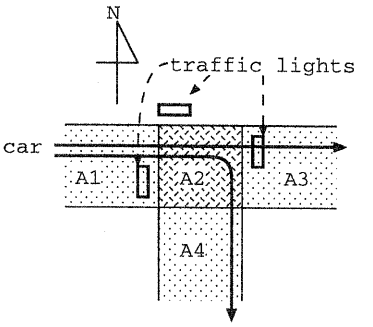

図 2: 単純な交通信号モデル

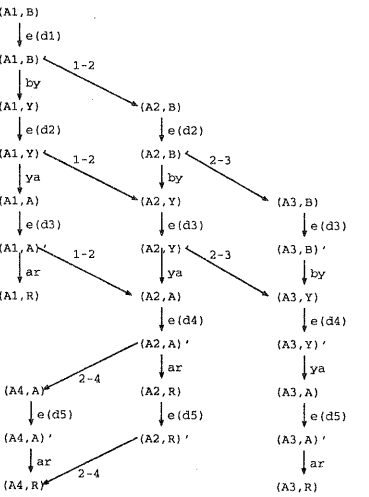

図 3: A-TSLTS によるモデル化の例

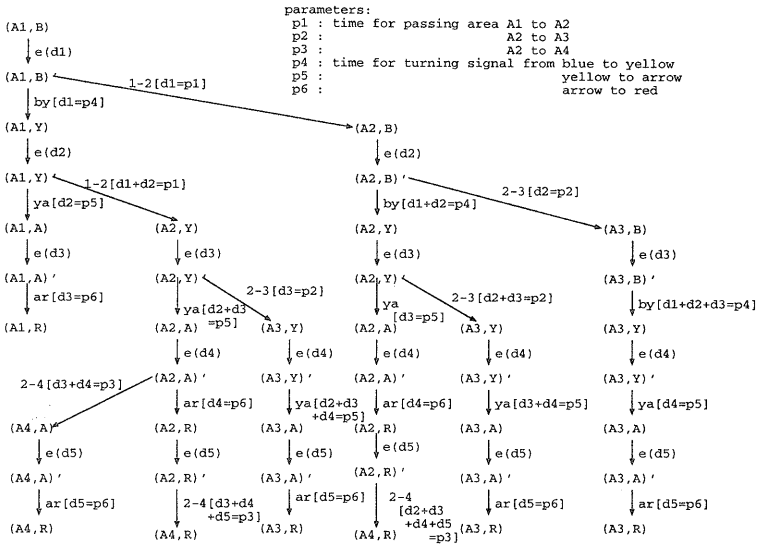

図 4: 図 3 を木に変換したモデル

$$
\begin{array}{rcl}
f & ::= & true \quad (\text{恒真}) \\
& | & false \quad (\text{恒偽}) \\
& | & \neg f \quad (\text{否定}) \\
& | & f \wedge f \quad (\text{論理積}) \\
& | & f \vee f \quad (\text{論理和}) \\
& | & f \Rightarrow f \quad (\text{含意}) \\
& | & \langle a \rangle_{\sim c} f \quad (\text{存在 next 演算子}) \\
& | & [a]_{\sim c} f \quad (\text{全称 next 演算子}) \\
& | & f E U_{\sim c} f \quad (\text{存在 until 演算子}) \\
& | & f A U_{\sim c} f \quad (\text{全称 until 演算子}) \\
& | & E G_{\sim c} f \quad (\text{存在 always 演算子}) \\
& | & A G_{\sim c} f \quad (\text{全称 always 演算子}) \\
& | & E F_{\sim c} f \quad (\text{存在 eventually 演算子}) \\
& | & A F_{\sim c} f \quad (\text{全称 eventually 演算子})
\end{array}
$$

図 5: TCTL の構文

$$
\begin{array}{rcl}
(s,\rho) & \models & true. \\
(s,\rho) \models \neg f & \stackrel{\text{def}}{=} & (s,\rho) \not\models f. \\
(s,\rho) \models f_1 \wedge f_2 & \stackrel{\text{def}}{=} & (s,\rho) \models f_1 \text{かつ} (s,\rho) \models f_2.
\end{array}
$$

$(s,\rho) \models \langle a \rangle_{\sim c} f \stackrel{\text{def}}{=}$
$(s,\rho) - t \rightarrow (s',\rho') - a \rightarrow (s'',\rho'')$
なるある遷移系列が存在して
$t \sim c$ かつ $(s'',\rho'') \models f$.

$(s,\rho) \models f_1 E U_{\sim c} f_2 \stackrel{\text{def}}{=}$
$(s,\rho) = (s_1,\rho_1) - t_1 \rightarrow (s'_1,\rho'_1) - a_1 \rightarrow$
$\cdots - t_{k-1} \rightarrow (s'_{k-1},\rho'_{k-1}) - a_{k-1} \rightarrow (s_k,\rho_k)$
なるある遷移系列が存在して、
$(s_k,\rho_k) \models f_2$かつ $t_1 + \cdots + t_{k-1} \sim c$
かつ任意の $i (1 \leq i \leq k-1)$ に対して $(s_i,\rho_i) \models f_1$.

図 6: TCTL 式の意味定義

3 TCTL

本章では本研究で要求仕様を記述する TCTL(Timed Computation Tree Logic) について述べる。

定義 3.1 TCTL 式 f の構文は図 5 に示す BNF で定義される。ただし、a は動作名、c は非負実数定数、$\sim$ は比較演算子 $<, \leq, >, \geq, =$ のいずれかである。$\sim c$ は省略可能とし、その場合は ≥ 0 が指定されたものとする。 □

TCTL 式は具体的状態、すなわち、状態とパラメータへの付値の対に対してその状態からの動作系列に対して成り立って欲しい性質を記述する論理である。直観的な意味は次の通りである。$true$ は任意の具体的状態に対して成り立つ式である。$\neg f$ は f が成り立たない具体的状態に対して成り立つ式である。$false$ は任意の具体的状態に対して成り立たない式であり、$\neg true$ と同じ意味である。$f_1 \wedge f_2$ は f_1 と f_2 が共に成り立つ具体的状態で成り立つ

(1) $EF_{<q_1}\langle 2\text{-}3\rangle true$
(2) $AG[1\text{-}2]EF_{\leq q_2}\langle 2\text{-}4\rangle true$

図 7: 交通信号モデルに対する条件

式である。$f_1 \vee f_2$ は f_1 と f_2 の少なくともいずれか一方が成り立つ具体的状態で成り立つ式であり、$\neg((\neg f_1) \wedge (\neg f_2))$ と同じ意味である。$f_1 \Rightarrow f_2$ は f_1 が成り立たないか f_2 が成り立つかどちらかの具体的状態で成り立つ式であり、$\neg(f_1 \wedge (\neg f_2))$ と同じ意味である。$\langle a\rangle_{\leq c} f$ は c 単位時間以内に動作 a を実行可能で、次の状態で f が成り立つような具体的状態で成り立つ式である。$\sim$ が $\geq, <, >, =$ の場合も同様に定義されるので、以下、$\sim$ が $\leq$ の場合についてのみ述べる。$[a]_{\leq c} f$ は c 単位時間以内に動作 a を実行したならば必ず次の状態で f が成り立つような具体的状態で成り立つ式で、TCTL 式 $\neg\langle a\rangle_{\leq c}\neg f$ と同じ意味である。$f_1 EU_{\leq c} f_2$ は f_2 が成り立つ状態へ到達するある動作系列が存在して、その状態へ到達するまでの任意の状態では f_1 が成り立っているような具体的状態で成り立つ式である。$f_1 AU_{\leq c} f_2$ は f_2 が成り立つ状態へ到達する任意の動作系列に対して、その状態へ到達するまでの任意の状態では f_1 が成り立っているような具体的状態で成り立つ式であり、$\neg(\neg f_2 EU_{\leq c}(\neg f_1 \wedge \neg f_2))$ と同じ意味である。$EF_{\leq c} f$ は $true EU_{\leq c} f$ と同じ意味、$AG_{\leq c} f$ は $\neg EF_{\leq c}\neg f$ と同じ意味、$AF_{\leq c} f$ は $true AU_{\leq c} f$ と同じ意味、$EG_{\leq c} f$ は $\neg AF_{\leq c}\neg f$ と同じ意味である。

一般に、$(s,\rho) \models f$ で TCTL 式 f が A-TSLTS の具体的状態 (s,ρ) において真となることを表す。

関係 $\models$ の形式的な定義を以下に示す。なお、以下の定義では $true$, $\neg f$, $f_1 \wedge f_2$, $\langle a\rangle_{\sim c} f$, $f_1 EU_{\sim c} f_2$ の 5 つの構文の意味定義のみ示す。他の構文はすべてこれら 5 種類の構文のみを用いて記述可能である。

定義 3.2 A-TSLTS の任意の具体的状態 (s,ρ) と TCTL 式 f の各構文に対して関係 $(s,\rho) \models f$ を図 6 のように定義する。 □

例 3.1 例 2.2 のモデル (の初期状態 $(A1, B)$) が満たすべき性質として、「直進車は q_1 単位時間以内に必ず A1 から A3 まで通過できる」および「右折車は q_2 単位時間内に必ず A1 から A4 へ右折できる」という 2 つの性質を TCTL で記述すると、図 7 のようになる。性質 (1) は車が初期状態から領域 A3 へ達するまで q_1 単位時間以内で到達できることを表し、また、性質 (2) は車が領域 A1 から A2 に移動したら必ず、q_2 単位時間以内で A4 へ抜けることが可能であることを表す。 □

4 パラメータ条件導出

本章では文献 [5] で提案した、ループを含まない A-TSLTS 仕様の状態 (正確には休止状態)s および TCTL 式 f から、s が f を満足するための s のパラメータに関する条件式を実数プレスブルガー文の形で出力するアルゴリズムを簡単に紹介する。その前にまず、本研究におけるパラメータ導出問題の正確な定義を以下に示す。

定義 4.1 パラメータ条件導出問題とは、A-TSLTS モデル M とその任意の休止状態 s および TCTL 式 f が与えられたとき、任意の付値 ρ に対して条件「$\rho \models P \Leftrightarrow (s,\rho) \models f$」を満足するような実数プレスブルガー式 P を導出する問題である。 □

パラメータ条件 P を具体的に求める関数 $PT(s,f)$ を以下に形式的に定義する。

定義 4.2 ループを持たない A-TSLTS モデルの休止状態 s および TCTL 式 f に対して実数プレスブルガー式を返す関数 $PT(s,f)$ を図 8 のように定義する。ただし、図 8 において $PT'(s,f,d)$ は $PT(s,f)$ の先頭の存在量化子 $\exists d$ を除去した論理式とし、$P\{e/x\}$ は論理式 P に含まれる自由変数 x を式 e で置き換えた論理式を表すものとする。 □

文献 [5] の結果は以下の定理で表される。

定理 4.1 ループを持たない A-TSLTS 仕様 M に対して、アルゴリズム $PT(s,f)$ はパラメータ条件導出問題の解を与える。すなわち、

$$\forall\rho[\rho \models PT(s,f) \Leftrightarrow (s,\rho) \models f]$$ □

5 導出例

例 2.2 の交通信号モデルと例 3.1 の性質 (1),(2) に対してアルゴリズム $PT()$ を適用した結果は図 9 のようになる。一般にアルゴリズム $PT()$ の適用結果は量化子を含む複雑な式になるが、これをプレスブルガー式簡約化ツール Omega[1] を用いて簡約化した結果、以下の式が得られた。

$$\begin{array}{lll} \text{性質 (1)} & : & p_1 < q_1 \wedge p_1 + p_2 < q_1 \\ \text{性質 (2)} & : & p_3 < q_2 \wedge p_4 < p_1 + q_2 \\ & & \wedge p_4 + p_5 < p_1 + q_2 \end{array}$$

導出にかかった時間は 0.2 秒程度であった。したがって、上記の式を満たすように各パラメータ $p_1 \sim p_6$ を設定すれば性質 (1),(2) を満足するようにできる。

6 あとがき

本論文では、文献 [5] で提案した、ループのない A-TSLTS モデルおよび TCTL 式で記述された性質からモデルが性質を満足するために必要十分なパラ

[1] Omega は `http://www.cs.umd.edu/projects/omega` で得られるフリーソフトである。本研究では Version1.1 の SUN SPARC Solaris2.5 用実行ファイルを用いた。

$$PT(s,true) \stackrel{\text{def}}{=} true \qquad PT'(s,true,d_s) \stackrel{\text{def}}{=} true$$

$$PT(s,\neg f) \stackrel{\text{def}}{=} \neg PT(s,f) \qquad PT'(s,\neg f,d_s) \stackrel{\text{def}}{=} \neg PT'(s,f,d_s)$$

$$PT(s,f_1 \wedge f_2) \stackrel{\text{def}}{=} PT(s,f_1) \wedge PT(s,f_2) \qquad PT'(s,f_1 \wedge f_2,d_s) \stackrel{\text{def}}{=} PT'(s,f_1,d_s) \wedge PT'(s,f_2,d_s)$$

$I(a,s) = \{i | s\text{-}e(d_s)[P_s] \rightarrow s'\text{-}a[P_i] \rightarrow s_i\}$ のとき、

$$PT(s,\langle a \rangle_{\sim c} f) \stackrel{\text{def}}{=} \exists d_s[[0 \leq d_s] \wedge [d_s \sim c] \wedge P_s \wedge \bigvee_{i \in I(a,s)} [P_i \wedge PT(s_i,f)]]$$

$$PT'(s,\langle a \rangle_{\sim c} f,d_s) \stackrel{\text{def}}{=} [[0 \leq d_s] \wedge [d_s \sim c] \wedge P_s \wedge \bigvee_{i \in I(a,s)} [P_i \wedge PT(s_i,f)]]$$

$I(s) = \{i | s\text{-}e(d_s)[P_s] \rightarrow s'\text{-}a_i[P_i] \rightarrow s_i\}$ のとき、

$$PT(s,f_1 EU_{\sim c} f_2) \stackrel{\text{def}}{=} \exists d_s[[0 \leq d_s] \wedge [d_s \sim c] \wedge [PT'(s,f_2,d_s) \vee [PT'(s,f_1,d_s) \wedge P_s \wedge \bigvee_{i \in I(s)} [P_i \wedge PT(s_i,f_1 EU_{\sim(c-d_s)} f_2)]]]]$$

$$PT'(s,f_1 EU_{\sim c} f_2,d_s) \stackrel{\text{def}}{=} [[0 \leq d_s] \wedge [d_s \sim c] \wedge [PT'(s,f_2,d_s) \vee [PT'(s,f_1,d_s) \wedge P_s \wedge \bigvee_{i \in I(s)} [P_i \wedge PT(s_i,f_1 EU_{\sim(c-d_s)} f_2)]]]]$$

図 8: アルゴリズム $PT(s,f)$

$$\begin{aligned}
&PT((A1,B), EF_{<q_1}\langle 2\text{-}3\rangle true) = \\
&\quad \exists d_1((d_1 < q_1) \wedge \\
&\quad\quad ((d_1 = p_1) \wedge \\
&\quad\quad\quad \exists d_2(d_2 < q_1 - d_1 \wedge (d_2 = p_2 \vee \\
&\quad\quad\quad\quad ((d_1 + d_2 = p_4) \wedge \\
&\quad\quad\quad\quad\quad \exists d_3(d_3 < q_1 - d_2 - d_1 \wedge d_2 + d_3 = p_2)))) \\
&\quad\quad \vee \\
&\quad\quad (d_1 = p_4) \wedge \\
&\quad\quad\quad \exists d_2(d_2 < q_1 - d_1 \wedge d_1 + d_2 = p_1 \wedge \\
&\quad\quad\quad\quad \exists d_3(d_3 < q_1 - d_2 - d_1 \wedge d_3 = p_2)))) \\
&PT((A1,B), AG[1\text{-}2] EF_{\leq q_2}\langle 2\text{-}4\rangle true) = \\
&\quad \forall d_1(\neg(d_1 = p_1) \vee \\
&\quad\quad \exists d_2(d_1 + d_2 = p_4 \wedge d_2 < q_2 \wedge \\
&\quad\quad\quad \exists d_3(d_3 = p_5 \wedge d_3 < q_2 - d_2 \wedge \\
&\quad\quad\quad\quad \exists d_4(d_4 < q_2 - d_2 - d_3 \wedge d_2 + d_3 + d_4 = p_3 \\
&\quad\quad\quad\quad\quad \vee d_4 = p_6 \wedge \\
&\quad\quad\quad\quad\quad\quad \exists d_5(d_5 < q_2 - d_2 - d_3 - d_4 \\
&\quad\quad\quad\quad\quad\quad\quad \wedge d_2 + d_3 + d_4 + d_5 = p_3))))) \\
&\quad \wedge \forall d_1(\neg(d_1 = p_4) \wedge \\
&\quad\quad \forall d_2(\neg(d_1 + d_2 = p_1) \vee \\
&\quad\quad\quad \exists d_3(d_2 + d_3 = p_5 \wedge d_3 < q_2 \wedge \\
&\quad\quad\quad\quad \exists d_4(d_4 < q_2 - d_3 \wedge d_3 + d_4 = p_3 \\
&\quad\quad\quad\quad\quad \vee d_4 = p_6 \wedge \\
&\quad\quad\quad\quad\quad\quad \exists d_5(d_5 < q_2 - d_3 - d_4 \\
&\quad\quad\quad\quad\quad\quad\quad \wedge d_3 + d_4 + d_5 = p_3)))))
\end{aligned}$$

図 9: アルゴリズム $PT()$ の適用結果

メータ値に対する条件式を自動導出するアルゴリズムを交通信号の例題に適用した。本例題程度の単純なものでも、状態の直積をとることによりモデルの状態数はやや多くなった。その結果、導出される式も複雑なものになったが、簡約化ツールを併用することにより、最終的な制約式としては簡潔なものを導出することが出来た。今後の課題としては、モデルの状態数が増えないようなモデル化手法および導出アルゴリズムを考案することが考えられる。

参考文献

[1] Nakata, A., Higashino, T. and Taniguchi, K.: Time-Action Alternating Model for Timed LOTOS and its Sympolic Verification of Bisimulation Equivalence, in *Proc. of FORTE/PSTV'96*, pp. 279–294, IFIP, Chapman & Hall (1996).

[2] Alur, R., Courcoubetis, C. and Dill, D.: Model-Checking in Dense Real-Time, *Information and Computation*, Vol. 104, pp. 2–34 (1993).

[3] Alur, R. and Dill, D.: Automata For Modelling Real-Time Systems, in Paterson, M. S. ed., *Proc. of ICALP'90*, Vol. 443 of *Lecture Notes in Computer Science*, pp. 322–335, Springer-Verlag (1990).

[4] Wang, F.: Parametric Timing Analysis for Real-Time Systems, *Information and Computation*, Vol. 130, No. 2, pp. 131–150 (1996).

[5] 新子浩康, 中田明夫, 大場充:時相論理式を満足する実時間プロトコル仕様のパラメータ条件導出, 情処研報 2000-DPS-97-9, 情報処理学会 (2000).