数理解析研究所講究録
第1744巻 2011年 25-34

2010年度冬のLAシンポジウム [4]

確率時間WiGARによるPTCTLサブクラスのモデル検査

高橋 正樹* 清水 隆也† 山根 智‡

1 導入

1.1 研究背景

近年，情報化社会が進むにつれてシステムの重要性が高くなってきており，誤作動を起こした場合に多大な影響を及ぼすシステムが存在する．その中には通信プロトコルのように確率的リアルタイム動作をする組込みシステムも存在する．そのため確率リアルタイムシステムに対しても網羅的な検証が可能である形式的手法による検証手法の確立が求められている．確率リアルタイムシステムのモデル化には確率時間オートマトン [8] を用いるのが一般的である．しかし実際のシステムを単純に適用しただけでは状態数が膨大になり計算不可能となる状態爆発という問題が生じる．そのため状態爆発を解決した形式的検証手法の確立が求められている

1.2 関連研究

確率リアルタイムシステムに対する形式的検証手法の研究としてはSymbolic model checking[8] が存在する．またCEGAR(Counter Example Guided Abstraction Refinement) の枠組みを用いたリアルタイムシステムに対するCTLのモデル検査 [11] の流れを受け継いだ研究として確率CEGARによる確率システムの確率到達性問題の検証 [5], 確率時間CEGARによる確率リアルタイムシステムに対する確率到達性問題の検証 [12] が挙げられる．Symbolic model checkingではPTCTL(Probabilistic Timed Computation Tree Logic) 式 [8] であらわされる検証項目全てに対して検証が可能である．しかし，この手法においても検証に必要なモデルの状態数が大きくなってしまう場合が存在する．一方，確率時間CEGARの手法は反例を用いて抽象モデルに対して精錬を繰り返すことで検証に必要なモデルの状態数を削減することができるCEGARの枠組みを確率リアルタイムシステムに対しも利用できるように拡張した研究である．そのため少ない状態数で確率リアルタイムシステムを検証できる．しかし，確率時間CEGARの手法ではPTCTL式で表現できる検証項目の一部である単純な確率到達可能性問題しか検証を行えないという欠点が存在する．

1.3 提案手法

本研究では上記で述べた関連研究の特徴からCEGARの枠組みを拡張することでPTCTL式で表現できる検証項目の全てではないが確率到達性問題以上の検証項目を検証することができる確率時間WiGAR(Witness Guided Abstraction Refinement)の手法を提案する．具体的には4章において述べるが, 単純にCEGARの枠組みをPTCTLの検証に適用した場合,「抽象モデルがPTCTL式を満たすならば必ず具体モデルもPTCTL式を満たす」というCEGARによる検証に必要となる抽象モデルを構築できないという問題が発生する．そこで本研究では,「抽象モデルがPTCTL式を満たさないならば必ず具体モデルもPTCTL式を満たさない」という性質を持つ抽象モデルを構築し，さらに反例ではなく実例を用いることでことでその問題を解決している．

以下では，2章において確率リアルタイムシステムを表現するモデルである確率時間オートマトン，その意味である確率時間システム，そして検証項目を

*金沢大学大学院自然科学研究科
†第1著者に同じ
‡金沢大学

表現する PTCTL の定義を行う．3 章では確率時間オートマトンの意味である確率時間システムの抽象モデルを定義し，その構築方法について述べる．4 章では，確率時間 WiGAR の流れについて説明する．5 章では実際に確率時間 WiGAR を用いて実験を行い既存手法と比較することで，その有効性を示す．最後に 6 章ではまとめと今後の課題について述べる．

2 具体モデルと PTCTL

確率リアルタイムシステムのモデルとして確率時間オートマトン [6] を定義する．

定義 2.1 (確率時間オートマトン) 確率時間オートマトン G は以下の組 $(L, l_0, C, Inv, prob)$ で定義される．

- ロケーションの有限集合 L
- 初期ロケーション $l_0 \in L$
- クロック変数の有限集合 C
- ロケーションに不変条件を割り付ける関数 $Inv : L \to Zones(C)$
- 確率遷移関係の有限集合 $prob \subseteq L \times Zones(C) \times \mathsf{Dist}(2^C \times L)$

□

G の状態 s はロケーションとクロック評価の対 (l, ν) で表現される．G の動作は，初期ロケーション l_0 と全てのクロック変数の値が 0 であるクロック評価 ν_0 の対である初期状態 (l_0, ν_0) から開始する．(l_0, ν_0) から状態間を時間遷移か離散遷移を行うことによって動作する．

時間遷移は同一ロケーション内で行い，状態 (l, ν) から $t \in \mathbb{R}^{>0}$ 時間遷移する場合は，確率 1 で状態 $(l, \nu + t)$ になる．ただしロケーションの不変条件により，t は $\nu + t \triangleright Inv(l)$ を満たさなければならない．

離散遷移は確率遷移関係 $(l, \zeta_g, p) \in prob$ を用いてロケーション間で行う遷移である．確率遷移関係は，遷移元ロケーション $l \in L$，遷移条件 $\zeta_g \in \mathrm{Zones}(C)$，

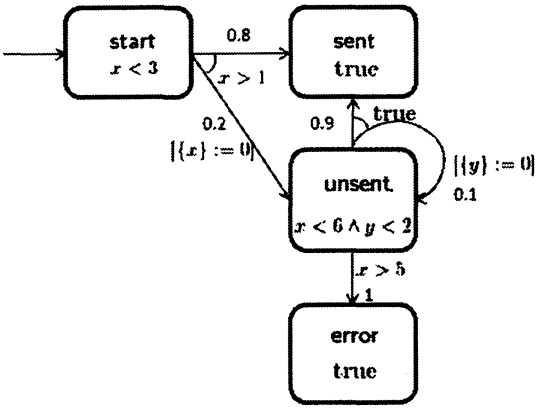

図 1: 確率時間オートマトン G_1

確率分布 $p \in \mathsf{Dist}(2^C \times L)$ の組である．確率分布 $p : 2^C \times L \to (0, 1]$ は，リセットするクロック変数の集合 $X \in 2^C$ と，遷移先ロケーション $l' \in L$ から遷移確率 $p(X, l') \in (0, 1]$ を与える写像である．

状態 (l, ν) からの離散遷移を考える．遷移関係が (l, ζ_g, p)，リセットするクロックの集合が X，遷移先ロケーションが l' であったとき，離散遷移先の状態は $(l', \nu[X := 0])$ となる．

以下の二つを満たすとき離散遷移可能である．

- 遷移元状態のクロック評価が遷移条件を満たしている．
- 遷移先状態のクロック評価が遷移先ロケーションの不変条件を満たしている．

つまり，$\nu \triangleright \zeta_g$ かつ $\nu[X := 0] \triangleright Inv(l')$ でなければならない．

例 2.1 確率時間オートマトンの動作例を，図 *1* のモデルを例にとり説明する．

$x, y \in C$ はクロック変数であり，start, sent, unsent, error $\in L$ はロケーションである．図 *1* における start から確率 *0.8* で sent に，確率 *0.2* で unsent に離散遷移する確率分布を μ_0，unsent から確率 *0.9* で sent に，確率 *0.1* で unsent に離散遷移する確率分布を μ_1 とする．

- 初期状態 $(\mathsf{start}, x = 0 \wedge y = 0)$

- *2*単位時間の時間経過 $(\mathsf{start}, x = 2 \wedge y = 2)$ ここでは不変条件 $x < 3$ より *3*単位時間以上時間経過できない.
- 離散遷移関係 $(\mathsf{start}, x > 1, \mu_0)$ により確率 *0.2* の離散遷移 $(\mathsf{unsent}, x = 0 \wedge y = 2)$*(*リセット $[\{x\} := 0]$ によりクロック xのみ *0*にリセットされる*)*
- *1*単位時間の時間経過 $(\mathsf{unsent}, x = 1 \wedge y = 3)$
- 離散遷移関係 $(\mathsf{unsent}, \mathsf{true}, \mu_1)$ により確率 *0.9* の離散遷移 $(\mathsf{sent}, x = 1 \wedge y = 3)$

以上のように時間遷移，離散遷移を繰り返すことによりシステムは動作する.

2.1 意味論

確率時間オートマトン G の意味を時間確率システム $\mathcal{M}$[6] とし，抽象化における具体モデルとして利用する．時間確率システムはマルコフ決定過程の形をとる．さらに与えられたパスの充足条件を満たすパスの合計確率である充足確率を定義するため，時間確率システムの非決定を解決するアドバサリを導入し，離散時間マルコフ連鎖の形に変換する.

定義 2.2 (時間確率システム) 確率時間オートマトン $G = (L, C, \mathit{Inv}, \mathit{prob}, \mathcal{L})$ の意味となる時間確率システム $\mathcal{M}_G$ を組 $(S, \mathit{Steps}, \mathcal{L}')$ とする.

- 状態集合 $S \subseteq L \times \mathcal{V}_C$
- 状態遷移関係 $\mathit{Steps} \subseteq S \times \mathbb{R}^{>0} \times \mathsf{Dist}(2^C \times S)$
- ラベリング関数 任意の $(l, \nu) \in S$ に対して $\mathcal{L}'(l, \nu) = \mathcal{L}(l)$

状態集合 S に含まれる状態 $(l, \nu) \in S$ は $\nu \triangleright \mathit{Inv}(l)$ を満たしていなければならない.

状態遷移関係 Steps は時間遷移と離散遷移からなる．状態遷移関係における確率分布を $\mu \in \mathsf{Dist}(2^C \times S)$ とし，特に時間遷移における確率分布を $\mu_\perp$ とする.

時間遷移における経過時間を $t \in \mathbb{R}^{>0}$ 単位時間，G における確率遷移関係を $(l, \zeta_g, p) \in \mathit{prob}$ として，$(l, \nu) \in S$ から $(l', \nu') \in S$ への遷移関係 $((l, \nu), t, \mu) \in \mathit{Steps}$ を以下のように定義する.

- t 単位時間の時間遷移
  $(l, \nu) \xrightarrow{t, \mu_\perp(\emptyset, (l', \nu'))} (l', \nu')$
  ただし，

$$\mu_\perp(\emptyset, (l', \nu')) = \begin{cases} 1 & \text{if } l' = l \wedge \nu' = \nu + t \wedge \\ & \quad \nu' \triangleright \mathit{Inv}(l) \wedge t > 0 \\ 0 & \text{otherwise} \end{cases}$$

- (l, ζ_g, p) による離散遷移
  $(l, \nu) \xrightarrow{0, \mu(X, (l', \nu'))} (l', \nu')$
  ただし，

$$\mu(X, (l', \nu')) = \begin{cases} p(X, l') & \text{if } \nu \triangleright \zeta_g \wedge \nu' = \nu[X := 0] \\ 0 & \text{otherwise} \end{cases}$$

□

$\mathcal{M}$ 上の状態遷移関係の確率 $\mu(X, (l', \nu'))$ による遷移は，クロック変数 X をリセットして状態 (l', ν') へ到達する遷移である．離散遷移ついて，G 上の確率分布 $p(X, l')$ には $\mathcal{M}$ 上の確率分布 $\mu(X, (l', \nu'))$ が対応しているため，これらの確率は等しい.

$\mathcal{M}$ 上のパス ω は以下のような $\mathcal{M}$ の状態 $s \in S$ から始まる以下のような非空の有限または無限列である.

$$\omega = s \xrightarrow{t, \mu(X, s')} s' \xrightarrow{t', \mu'(X', s'')} s'' \xrightarrow{t'', \mu''(X'', s''')} \cdots$$

パスが有限長である場合，ω_{fin} と表記し，その $|\omega_{\mathit{fin}}|$ を長さ（遷移の回数），$\mathit{last}(\omega_{\mathit{fin}})$ を最後の状態とする．無限長である場合，ω_{ful} と表記しする．また状態 s から始まる全ての無限長 (有限長) のパスの集合を $\mathit{Path}_{\mathit{ful}}(s)(\mathit{Path}_{\mathit{fin}}(s))$ とする．また，$\Sigma_{s \in S} \mathit{Path}_{\mathit{ful}}(s)$ であるすべての無限長のパスの集合を $\mathit{Path}_{\mathit{ful}}$ とし，同様に $\Sigma_{s \in S} \mathit{Path}_{\mathit{ful}}(s)$ であるすべての有限長のパスの集合を $\mathit{Path}_{\mathit{fin}}$ と記述する.

次に, 時間確率システムの非決定的選択を解決させるものとして, アドバサリを導入する.

定義 2.3 (時間確率システムのアドバサリ) $\mathcal{M}$ のアドバサリ A は, 有限長のパス ω_{fin} から, パスの最後の状態を遷移元とする状態遷移関係 $(last(\omega_{fin}), t, \mu) \in Steps$ の時間遷移量と確率分布を割り当てる関数 $A : Path_{fin} \to \mathbb{R}^{\geq 0} \times Dist(2^C \times S)$ である. 全てのアドバサリからなる集合を $A \in Adv$ とする.

2.2 PTCTL

ここでは確率時間オートマトンの検証に用いられる Probabilistic Timed Computation Tree Logic(PTCTL)[8] のサブクラスを紹介する. PTCTL は時間論理 (TCTL) と確率論理 (PCTL) を結合させたものであり, 時間的, 確率的なシステムの特性を表現することができる. PTCTL のサブクラスの構文は以下のように定義される.

定義 2.4 (制限された PTCTL の構文)
$\phi ::= a \mid \zeta \mid \phi \vee \phi | \phi \wedge \phi | {}^{\exists}\mathcal{P}_{>\lambda}[\phi \; \mathcal{U} \; \phi]$
ここで $a \in AP, \zeta \in Zones(C), \lambda \in [0,1]$ である. □

本論文では以後, 明示しない限り PTCTL は制限された PTCTL であるとする.

次に制限された PTCTL の意味論を定義する.

定義 2.5 (制限された PTCTL の意味論) $TPS = (S, TStep, \mathcal{L}')$ を確率時間オートマトン PTA に対しての時間確率システムとする. 任意の状態 $s \in S$, $PTCTL$ のフォーミュラ θ, に対して s が θ を充足することを $s \models \theta$ と表す. 関係 $\models$ は以下のように再帰的に定義される.

$$
\begin{aligned}
s \models a &\iff a \in \mathcal{L}'(s) \\
s \models \zeta &\iff s \triangleright \zeta \\
s \models \phi_1 \vee \phi_2 &\iff s \models \phi_1 \; or \; s \models \phi_2 \\
s \models \phi_1 \wedge \phi_2 &\iff s \models \phi_1 \; and \; s \models \phi_2
\end{aligned}
$$

$s \models^{\exists} \mathcal{P}_{>\lambda}[\phi_1 \mathcal{U} \phi_2] \iff P_s^A(\phi_1 \mathcal{U} \phi_2) > \lambda$ である $A \in Adv$ が存在する.

ここで
$P_s^A(\phi_1 \mathcal{U} \phi_2) = Prob_s^A\{\omega \in Path_{ful}^A(s) \mid \omega \models \phi_1 \mathcal{U} \phi_2\}$ ある.
また, 任意のパス $\omega \in Path_{ful}$ がパスの充足条件 $\phi_1 \mathcal{U} \phi_2$ を充足することを $\omega \models \phi_1 \mathcal{U} \phi_2$ と表している, 関係 $\models$ は以下のように定義される.
$\omega \models \phi_1 \mathcal{U} \phi_2 \iff \exists i \in \mathcal{N}.(\omega(i) \models \phi_2$ かつ $\forall j < i.\omega(j) \models \phi_1 \mathcal{U} \phi_2)$

□

ここである状態がパス論理式と呼ばれる ${}^{\exists}\mathcal{P}_{>\lambda}[\phi_1 \mathcal{U} \phi_2]$ である形の PTCTL 式を満たした場合, 必ず実例と呼ばれる $\sum_{\omega \in \Omega} Prob_{fin}^A(\omega) > \lambda$ を満たす状態 $s \in S$, アドバサリ $A \in Adv$,$\Omega \in \{Path_{ful}^A(s) \mid \omega \models \phi_1 \mathcal{U} \phi_2\}$ であるパスの有限集合 Ω の組み (s, A, Ω) が存在する.

上記のように PTCTL 式の意味論は状態に対して定義されている. ここでシステムが PTCTL 式 θ で表現された特性を満たすとはシステムの具体モデルの初期状態 s_0 が $s_0 \models \theta$ となる時であり, その時に限る. つまり, PTCTL のモデル検査とは具体モデルの初期状態が PTCTL 式を充足するかどうかの検証であると言える.

3 抽象モデルと PTCTL

本研究では具体モデルである時間確率システムに対して時間の述語抽象化を行うことで抽象モデルを構築する. 以下では述語抽象化について述べる.

定義 3.1 (抽象化述語) クロック変数の集合 C において, 述語 ψ は以下のように定義される.

$$\psi ::= x_1 \leq c | x_1 < c | x_1 - x_2 < d | \text{true}$$

ここで, $x_1, x_2 \in C$, $c \in \mathbb{N}$, $d \in \mathbb{Z}$ である. クロック評価 ν, 抽象化述語 ψ において, ν に関する述語 ψ の真偽値を $\psi\nu \in \{true, false\}$ とすると, ψ 中のクロック変数 $x \in C$ に対応する値 $\nu(x)$ を代入した

結果得られる式が真となるとき，かつそのときに限り ν は ψ をみたし，$\psi\nu = \mathit{true}$ であるという．ゾーンの定義にある $x > c$，$x \geq c$，$x_1 - x_2 < d$ は，それぞれ $x \leq c$，$x < c$，$x_2 - x_1 \geq -d$ の述語で真偽値が $\psi\nu = \mathit{false}$ の場合としてとらえることができる．また，すべてのクロック評価 $\nu \in \mathcal{V}_C$ において $\psi = \mathrm{true}$ は，$\psi\nu = \mathit{true}$ とする．

□

本研究ではロケーションごとに抽象化述語の集合 $\Psi^l = \{\psi_0^l, \cdots, \psi_{n-1}^l\}$ を与える．ここで，ψ_i^l は，ロケーション l における抽象化述語である．また，全てのロケーションにおける抽象化述語の族を $\mathbf{\Psi} = \{\Psi^{l_0}, \cdots, \Psi^{l_k}\}$ とする．

述語集合 Ψ^l に含まれる述語の数と等しい長さの，ビットベクトル b^l を考える．そのようなビットベクトル b^l を用いて，組 (l, b^l) を抽象状態 $s^\sharp$ とする．また，全てのロケーションにおけるビットベクトルの集合を $\mathcal{B}$，抽象状態の集合を $S^\sharp$ とする．

$\mathbf{\Psi}$ により $\mathcal{M}$ の状態 (l, ν) から抽象状態 (l, b^l) へのマッピングである抽象化関数 $\alpha : S \to S^\sharp$ が決定される．

この関数によって得られる抽象状態 $(l, b^l) = \alpha((l, \nu))$ における b^l について，その i 番目の要素を $b^l(i)$ とすると，$\psi_i^l\nu = b^l(i)$ である．例えば，$\Psi^l = \{x \leq 1, x - y < -1\}$ において状態 $s = (l, x = 1 \wedge y = 1)$ を抽象化した抽象状態は $\alpha(s) = (l, (\mathit{true}, \mathit{false}))$ となる．α の逆像を，具体化関数 $\gamma : S^\sharp \to 2^S$ とする．α，γ ともに，全射でも単射でもない．この α と γ を以下のように定義する．

定義 3.2 (抽象化・具体化) C はクロックの集合とし，$\mathcal{V}_C$ は対応するクロック評価の集合とする．

述語の有限集合 $\mathbf{\Psi} = \{\Psi^{l_0}, \cdots, \Psi^{l_k}\}$ が与えられたとき，抽象化関数 $\alpha : S \to S^\sharp$ は以下のように定義される．

$$\alpha((l, \nu)) = (l, b^l) \text{ s.t. } \forall i. b^l(i) = \psi_i^l\nu$$

また具体化関数 $\gamma : S^\sharp \to 2^S$ は以下のように定義される．

$$\gamma((l, b^l)) = \{(l, \nu) \in L \times \mathcal{V}_C | \mathit{Inv}(l) \wedge \bigwedge_{i=0}^{n-1} b^l(i) = \psi_i^l\nu\}$$

□

また，$b^l\Psi^l$ をその述語とビットベクトルが示すゾーンとして以下のように定義して用いる．

$$b^l\Psi^l = \zeta \text{ s.t. } \nu \triangleright \zeta \wedge \psi_i^l\nu \equiv b(i)$$

3.1 抽象モデル

抽象化述語と抽象化・具体化関数を用いて，ある述語集合 $\mathbf{\Psi}$ における時間確率システムの抽象モデルを構築する．抽象モデルは時間確率システムと同様にマルコフ決定過程の形をとる．抽象モデルは文献 [11] に従い確率分布の導入により拡張し，オーバー近似になるように定義する．オーバー近似とは，具体モデルがもつ遷移を抽象モデルに全て持たせることで，抽象化に健全性を持たせる近似である．

定義 3.3 (抽象モデルの形成) 確率時間オートマトン $G = (L, l_0, C, \mathit{Inv}, \mathit{Steps})$ から変換された時間確率システム $\mathcal{M} = (S, \mathit{Steps}, \mathcal{L}')$ における，述語集合 $\mathbf{\Psi}$ による抽象モデル $\mathcal{M}_{\mathbf{\Psi}}^\sharp = (S^\sharp, \mathit{Steps}^\sharp, \mathcal{L}')$ を以下のように構築する．

- 抽象状態集合 $S^\sharp = L \times \mathcal{B}$
- 抽象状態遷移関係 $\mathit{Steps}^\sharp \subseteq S^\sharp \times \mathsf{Dist}(2^C \times S^\sharp)$
- ラベリング関数 $\mathcal{L}'(l, b) = \mathcal{L}(l)$

$\exists (l, \nu) \in \gamma((l, b)).((l, \nu), \mu) \in \mathit{Steps}$ であるとき，遷移 $((l, b), \mu^\sharp) \in \mathit{Steps}^\sharp$ が存在する．

$((l, \nu), t, \mu)$ に対応する $((l, b), \mu^\sharp)$ における $\mu^\sharp$ は以下で定義される確率分布である．ただし $\alpha((l, \nu)) = (l, b), \alpha((l', \nu')) = (l', b')$ である．

$$\mu^\sharp(X, (l', b')) = \mu(X, (l', \nu'))$$

□

$\mathcal{M}^\sharp$ 上のアドバサリ $A^\sharp$ は，時間遷移量の t を省き，$A^\sharp : \mathit{Path}_{fin}^\sharp \to \mathit{Dist}(2^C \times S^\sharp)$ である．$\mathcal{M}^\sharp$ は $\mathcal{M}$ と同じくマルコフ決定過程であるため，$\mathcal{M}$ で用いていた表現に $^\sharp$ を付けることで，$\mathcal{M}^\sharp$ でも同様に表現する．

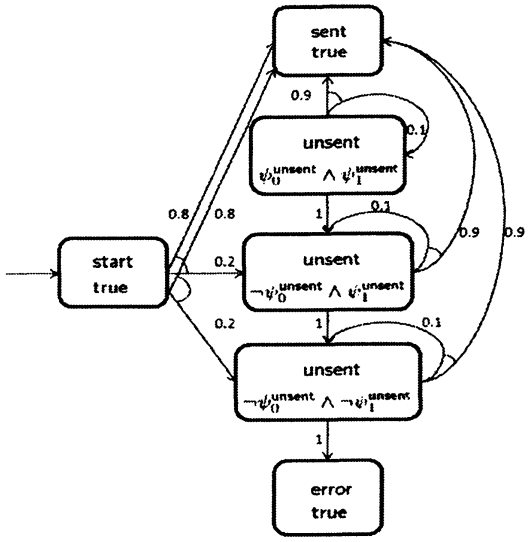

図 2: G_1 の抽象構造 $\mathcal{M}_{\Psi}^{\sharp}$

例 3.1 図 *1* の G_1 を抽象化した場合の抽象構造 $\mathcal{M}_{\Psi}^{\sharp}$ を図 *2* に示す．抽象化に伴う述語集合は $\psi_0^{un} = x \leq 1, \psi_1^{un} = x \leq 5$ である．

3.2 抽象モデル上での PTCTL の定義

本論文では PTCTL の充足性問題の解析を状態爆発を防ぐために抽象モデル上で行う．そのため抽象モデル上で PTCTL の意味論，PTCTL を充足する抽象状態の集合及び充足性問題を以下のように定義する．ただし状態 $s^{\sharp} = (l, b^l)$ とする．

定義 3.4 (抽象モデル上での PTCTL の意味論)

$$
\begin{aligned}
s^{\sharp} \models a &\iff a \in \mathcal{L}'(s^{\sharp}) \\
s^{\sharp} \models \zeta &\iff b^l \Psi^l \wedge \zeta \neq \emptyset \\
s^{\sharp} \models \phi \vee \psi &\iff s^{\sharp} \models \phi \ or \ s^{\sharp} \models \psi \\
s^{\sharp} \models \phi \wedge \psi &\iff s^{\sharp} \models \phi \ and \ s^{\sharp} \models \psi \\
s^{\sharp} \models^{\exists} \mathcal{P}_{>\lambda}[\phi_1 \mathcal{U} \phi_2] &\iff P_{s^{\sharp}}{}^{A^{\sharp}}(\phi_1 \mathcal{U} \phi_2) > \lambda \text{となる} \\
&\quad A \in Adv \text{が存在する．}
\end{aligned}
$$

ここで $P_{s^{\sharp}}^{A^{\sharp}}(\phi_1 \mathcal{U} \phi_2) = Prob_s^A\{\omega^{\sharp} \in Path_{ful}^{A^{\sharp}}(s^{\sharp}) \mid \omega^{\sharp} \models \phi_1 \mathcal{U} \phi_2\}$ ある．
また，任意のパス $\omega^{\sharp} \in Path_{ful}^{\sharp}$ がパスの充足条件 $\phi_1 \mathcal{U} \phi_2$ を充足することを $\omega^{\sharp} \models \phi_1 \mathcal{U} \phi_2$ と表している，関係 $\models$ は以下のように定義される．

$$
\begin{aligned}
\omega^{\sharp} \models \phi_1 \mathcal{U} \phi_2 \iff \exists i \in \mathcal{N}.(&\omega^{\sharp}(i) \models \phi_2 \text{かつ} \\
&\forall j < i.\omega^{\sharp}(j) \models \phi_1 \vee \phi_2)
\end{aligned}
$$

□

以上のように抽象モデルに対する PTCTL の意味論を定義することで次の定理がいえる．

定理 3.1 時間確率システム $\mathcal{M}$ の初期状態 s_0 と，そのオーバ近似である抽象モデル $\mathcal{M}^{\sharp}$ の初期抽象状態 $s_0^{\sharp}$ があった時，任意の形の *PTCTL* 式 θ に対して以下のことがいえる．$s_0 \models \theta \rightarrow s_0^{\sharp} \models \theta$

□

ここである状態がパス論理式と呼ばれる ${}^{\exists}\mathcal{P}_{>\lambda}[\phi_1 \mathcal{U} \phi_2]$ である形の PTCTL 式を満たした場合，必ず実例の候補と呼ばれる $\sum_{\omega^{\sharp} \in \Omega^{\sharp}} Prob_{fin}^{A^{\sharp}}(\omega^{\sharp}) > \lambda$ を満たす状態 $s^{\sharp} \in S^{\sharp}$，アドバサリ $A \in Adv^{\sharp}$ とパスの有限集合 $\Omega^{\sharp} \in \{\omega_{fin}^{\sharp} | \omega_{ful}^{\sharp} \in Path_{fin}^{A^{\sharp}}(s^{\sharp}) \wedge \omega^{\sharp} \models \phi_1 \mathcal{U} \phi_2\}$ の組み $(s^{\sharp}, A^{\sharp}, \Omega^{\sharp})$ が存在する．

4 確率時間 WiGAR

この章では確率時間 WiGAR の枠組みによる検証の流れについて解説する．ここで具体モデルと抽象モデルは定理 3.1 により PTCTL で表現される特性に対して健全性を保っている．つまり図 3 の関係になっている．

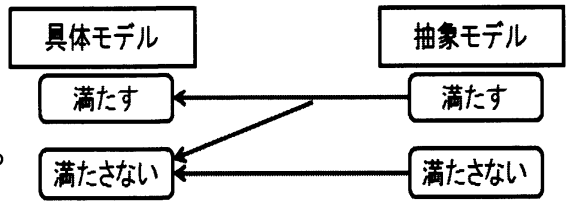

図 3: 具体モデルと抽象モデルの関係

つまり検証したい特性を抽象モデルが満たさなかったならば具体モデルでも満たさないことがわかる．本研究では具体モデルが抽象モデルに対して健全性をもつことを利用して図4の流れで検証を行う．

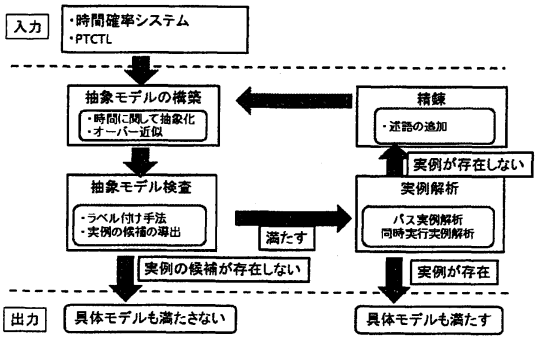

図4: 確率時間WiGARによる検証の流れ

以下では「抽象モデルの構築」,「抽象モデル検査」,「実例解析」,「精錬」の4つのそれぞれのフェーズについて簡単に述べる．

4.1 抽象モデルの構築

ここでは確率時間オートマトンの意味にあたる確率時間システムに対して文献手法 [11] により提案されている述語抽象化抽象により抽象化を行う．この抽象化により無限状態遷移系であった確率時間システムを有限状態遷移系に変換することができる．

4.2 抽象モデル検査

次に，述語抽象化により得られた抽象モデル上で検証したいPTCTL式のモデル検査を行う．検証方法としては文献 [13] で提案されているラベル付け手法によるPCTLの検証をPTCTLに拡張した手法を用いて行う．ここで抽象モデルは有限遷移系であるため有限時間内でラベル付け手法による検証を終えることができる．なお，この検証により抽象モデルがPTCTL式を満たさなければ，具体モデルである確率時間システムもまたPTCTL式を満たさないことがわかり確率時間WiGARによる検証を終えることができる．逆にもし満たした場合は，実例解析のフェーズに移行する．

4.3 実例解析

実例解析では抽象モデルではPTCTL式を満たしたが，具体モデルでも満たすのかということを解析する．具体的には古典的なモデル検査の手法によりPTCTL式を満たす状態集合を厳密に求めていくことになる．しかし，パス論理式と呼ばれる ${}^{\exists}\mathcal{P}_{>\lambda}[\phi_1 \mathcal{U} \phi_2]$ を満たす状態集合を正確に導出することは難しい．そこで本研究では，抽象モデル上でパス論理式を満たしたときに得られる実例 (Witness) の候補を解析することで，パス論理式を満たす状態集合の部分集合を求める方法を提案する．ここで，実例の候補は抽象状態 $s^\sharp$, アドバサリ $A^\sharp$, パス集合 $\Omega^\sharp$ の組である $(s^\sharp, A^\sharp, \Omega^\sharp)$ の形をとる．しかし，実例の候補の要素であるアドバサリ $A^\sharp$ とパス集合 $\Omega^\sharp$ の取り方として無限の組み合わせが存在する場合があるため，全ての実例の候補を解析した場合，解析が有限時間で終わらない．そこで本論文では解析する実例の候補のアドバサリとしては最大確率を与えるアドバサリを選択し，パス集合の選択方法として最小反例 [3] の手法を採用する．この最小反例の採用により，有限のパス集合から成る実例の候補をただ一つ構成することができる．そして，この実例の候補のみを解析することで実例解析は有限時間で終えることができる．ただし，この手法により得られる状態集合は実際にパス論理式を満たす状態集合の部分集合である．なお，この手法により求まったPTCTL式を満たす状態集合に具体モデルの初期状態が含まれていれば，時間確率システムはPTCTL式を満たすということがわかり，確率時間WiGARによる検証を終えることができる．逆に，初期状態を含んでいなかった場合は精錬のフェーズに移行する．

4.4 精錬

精錬では，抽象モデルはPTCTL式を満たしたが，その満たした実例の候補を解析したところ具体モデルの初期状態はPTCTL式を満たさなかった場合に行う．そのため，再び同じ誤った実例の候補により抽象モデルがPTCTL式を満たしてしまうことがないように抽象モデルを再構築する必要がある．つまり，精錬ではそのような抽象モデルの再構築に必要な述語の導出を行う．

以上が確率時間WiGARを構成する4つのフェーズである．以上のように確率時間WiGARでは4つのフェイズを検証結果が得られるまで繰り返すことになる．ここで必要に応じて抽象モデルを具体化し

ているため，できるだけ少ない状態数で検証を終えることができ，効率的な検証が可能になっている．

5 実験

この章では，これまで述べてきた確率時間 WiGAR の枠組みを用いて実際に検証を行い，文献 [8] による Symbolic model checking と状態数を比較することで確率時間 WiGAR の有効性を示す．なお，確率時間 WiGAR の実装においてゾーン演算は QE(Quantifier Elimination)[17] を利用して行う．

実験は IEEE1394 FireWire root contention プロトコルの確率時間オートマトンのモデル上において実験を行った．

ここで IEEE1394 FireWire root contention プロトコルとは 2 つのノードが接続されたネットワーク上でリーダを選出するためのプロトコルである．

このプロトコル動作を確率時間オートマトンにより表現したものが図 5 である．

今回の実験ではデットライン（表 5 での DLine）を設定した上で，1 以上の確率でそのデッドライン以内にリーダを選出できるか (*elect* に到達できるか) どうかを検証項目とする．この検証項目を PTCTL 式では ${}^{\exists}\mathcal{P}_{>0}[\text{true}\ \mathcal{U}\ (elect \wedge t > DLine)]$ と表現することができる．この実験において Symbolic model checking と確率時間 WiGAR とで結果を返すことができたモデルの状態数を比較したものが表 5 であり，それをグラフかしたものが図 6 である．なお，計算時間やメモリ使用量は実行環境が異なるため比較対象から除外した．

また，充足性問題の解ははデッドライン，検証手法にかかわらず'no' となった．

表 5 に示されるように確率時間 WiGAR ではどのようなデッドラインの値に対しても Symolic model checking よりも少ない状態数でのモデル上で検証を終えることができることを確認できた．これは，確率時間 WiGAR では最小実例を採用することにより，PTCTL 式を満たす上で最も大きな貢献をした（確率が最も大きくなる）パスから解析することにより効率的に抽象モデルを再構築できたためであると考えられる．また，状態数の減少は Symbolic model checking では検証に多くの状態数を持つモデルを構築する必要があったデッドラインを大きく設定場合により顕著に表れた．この場合確率時間 WiGAR においても検証には多くの WIGAR のループ回数が必要とされたが，そのそれぞれのループにおいて最小実例の採用により効率的に抽象モデルの構築が行われている．そのため 1 ループ毎の効率的な抽象モデルの構築が積み重なり，結果的に確率時間 WiGAR では Symbolic model checking と比べより良好な結果が得られたためであると考えられる．またこのことは状態数の増加によりメモリ制約が厳しい検証に対してより有効であるといえ，既存の検証手法では状態爆発の原因により検証できなかったものを検証できる可能性があるという点で有効性があると考えられる．

6 まとめ

本研究では確率的，リアルタイム的動作をするシステムのモデル検査において CEGAR の枠組みを改良した WiGAR の枠組みを導入することで空間計算量を大幅に減らす枠組みの提案を行った．また WiGAR の枠組みにおいて必要となる実例解析の手法，及び実例が存在しなかった場合における精錬の手法の提案を行った．

今後の課題としては，制限されていない PTCTL で表現される特性に対しても CEGAR の枠組みを用いて検証できるように理論を拡張することを目的としている．

参考文献

[1] Alfro,L,D,Model Checking of probabilistic and nondeterministic systems,Springer-Verlaf,LNCS 1026, (1995)499-513.

[2] Clarke. E. M, Grumberg. O, Peled. D. A, Model Checking,The MIT Press,(2000)

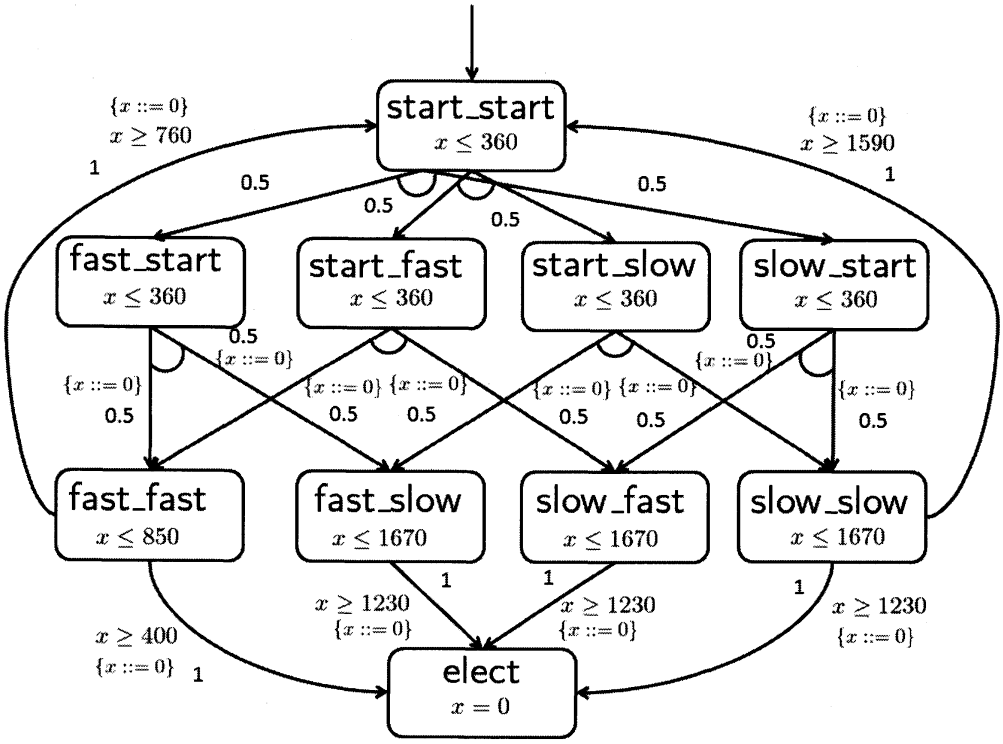

図 5: FireWire のプロトコルの確率時間オートマトン

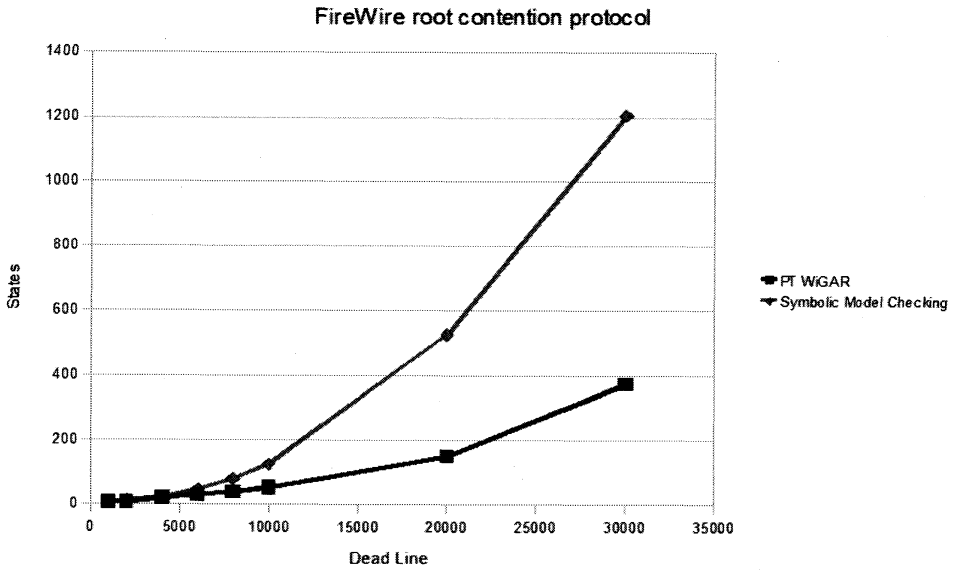

図 6: 状態数の比較のグラフ

表 1: 実験結果 3

Dline[$10^3 ns$]	Symbolic model checking	確率時間 WiGAR	WiGAR のループ回数
2	15	10	1
4	25	22	13
6	47	33	24
8	81	41	32
10	126	54	45
20	528	151	137
30	1206	374	328

[3] Han, T. and Katoen, J. P.: Counterexamples in probabilistic model checking, LNCS 4424,(2007)72-86.

[4] Henzinger. T. A, Nicollin. X, Sifakis. J, Yovine. S, Symbolic Model Checking for Real-Time Systems, Information and Computation 111 (1994)394-406.

[5] Hermanns. H, Wachter. B, Zhang. L, Probabilistic CEGAR, LNCS 5123, (2008)162-175.

[6] Kwiatkowska. M, Norman. G, Segala. R, Sproston. J, Automatic Verification ofReal-Time Systems with Discrete Probability Distributions, LNCS, 1601 (1999) 75-95.

[7] Kwiatkowska. M, Norman. G, Segala. R, Sproston. J, Automatic verification of real time systems with discrete probability distributions, Theor. Comput. Sci 282(1)(2002)101-150.

[8] M. Kwiatkowska, G. Norman, J. Sproston and F. Wang, Symbolic ModelChecking for Probabilistic Timed Automata, Information and Computation 205(7)(2007)1027-1077.

[9] Puterman. M, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley- Interscience, (1994).

[10] S. Graf, H. Saidi, Construction of Abstract State Graphs with PVS,LNCS 1254, (1997)72-83.

[11] Sorea. M, Oller. M. O. M, Rue. H, Predicate abstraction for dense real-time systems, Electronic Notes in Theoretical Computer Science 65(6)(2002).

[12] 森下 篤, 駒形 龍太, 山根 智, 確率時間 CEGAR, 電子情報通信学会技術研究報告. CST, コンカレント工学 109(73)(2009)25-30.

[13] Hansson. H, Jonsson. B,A Framework for Reasoning about Time and Reliability(2002)102-111.

[14] Bengtsson. J, Yi. W, Timed Automata:Semantics,Algorithms and Tools,LNCS 3098, (2004)87-124.

[15] T. Henzinger, X. Nicollin, J. Sifakis, S. Yovine, Symbolic model checking for real-time systems, Information and Computation 111(2) (1994)193-244.

[16] S. Tripakis, L ' analyse formelle des systemes temporises en pratique, Ph.D. thesis, Universite Joseph Fourier(1998).

[17] A. Tarski, A decision method for elementary algebra and geometry, University of California Press (Berkeley)(1951).