数理解析研究所講究録
第1799巻 2012年 29-36



2011年度冬のLAシンポジウム [6]

# 抽象化洗練を用いた時間確率システムに対する形式的検証手法

清水 隆也* 森下 篤† 山根 智‡

## 1 導入

### 1.1 背景

E.M.Clarkeらによって，リアクティブシステムの反例による抽象化精錬の枠組み（CEGAR）[6] のモデル検査 [4] が提案された．モデル検査ではシステムの状態数が大きくなる状態爆発の解決が課題になるが，述語抽象化 [7] を導入し，状態数を抑えながら検証可能な手法がCEGARである．本研究は，確率時間オートマトンに対してCEGARを適用し，モデル検査の特性の1つである到達可能性解析による安全性の検証を行う．

### 1.2 確率時間CEGAR

一般に，抽象化を行うとシステムは本来の性質を損なう．CEGAR[6] では，抽象モデル上での反例の候補の導出と，反例を用いた抽象モデルの精錬を，結論が得られるまで繰り返すことで正当性を保証するとともに，偽反例から得た情報により，検証に必要な部分のみを詳細化することで状態数を抑えることができる．CEGARによる枠組みを用いて検証を行うには以下の手法を確立しなければならない．

- 検証したい性質を抽象モデルが持っていれば，具体モデルも持っている健全性を保つ抽象化手法
- 抽象モデルから反例の候補を導出でき，その反例の候補が実動作可能な反例であるか解析する反例解析手法
- 反例解析の結果，反例の候補に対応する反例が存在しないとき，同じ反例の候補を生じさせないように抽象モデルを精錬する手法

本論文ではこれらを確立し，確率時間オートマトンを対象としてCEGARを導入することで，検証対象に応じて抑えられた状態空間の構築が可能であることを示す．上記手法の開発において，確率分岐による複数パスの同時の実行可能性を同時実行反例解析によって判定し，その偽反例による抽象モデルの精錬手法を実現する．また，本手法の実装実験を行い，既存手法 [10] と比較することで，より小さい状態空間での検証が可能であることを示す．

*金沢大学大学院自然科学研究科
†第1著者に同じ
‡金沢大学理工研究域電子情報学系

### 1.3 関連研究

これまでCEGARを適用した検証手法として，リアルタイムシステムを対象とした時間CEGAR[11] や，ハイブリッドシステムを対象とした研究 [5]，確率システムを対象とした確率CEGAR[9]，などが研究されてきた．

本研究では確率システム及びリアルタイムシステムの両方を併せ持つ性質を対象とするため，検証を行う上での課題として，同時実行可能性の解析手法が必要である．この手法はこれまで提案されておらず，本稿ではその手法を示す．

一方，確率リアルタイムシステムに対する検証の既存手法として，記号モデル検査 [10] がある．CEGARは，状態数を抑えながら検証可能な手法であり，導入することで効率的な検証を期待できる．本研究では確率時間CEGARを計算機上に実装して，上記手法との性能比較を行う．

## 2 確率時間オートマトン

### 2.1 準備

ここでは，クロック変数とゾーン [1]，確率時間オートマトン $G$[10]，時間確率システム $\mathcal{M}$[10]，$\mathcal{M}$ 上のパス $\omega$[10]，時間確率システムのアドバサリ $A$[10]，$A$ で $\mathcal{M}$ を解決した離散時間マルコフ連鎖 $\mathcal{MC}^A$[10] を準備する．これらの定義については，それぞれの文献に従うものとする．

### 2.2 確率到達可能性問題と反例

到達可能性問題はソフトウェアの検証におけるもっとも基本的な問題であり，様々な検証問題は到達可能性問題に帰着させることができる．本研究は，時間確率システムの安全性を到達可能性問題によって検証する．

**定義 2.1** (到達可能性問題). ロケーション集合 $L_e \subseteq L$ について，ロケーション $l_e \in L_e$ をもつ状態の集合を $S_e = \{(l_e, \nu) \in S | l_e \in L_e\}$ とする．また，$\lambda \in [0,1]$ を $s_0$ から状態集合 $S_e$ への到達確率とする．

$G$ の到達可能性問題を，到達確率 $\lambda \in [0,1]$ と目的ロケーションの集合 $L_e \subseteq L$ の組 $(\lambda, L_e)$ とする．

到達可能性問題 $(\lambda, L_e)$ の答えが *"yes"* であるとは，$\mathcal{M}$ において，$\forall A.Prob^A(S_e) \leq \lambda$ の場合に限る．それ以外の場合は *"no"* である． □

$L_e$ を到達することが望ましくないロケーションの集合，$S_e$ をそのような状態の集合，$Prob^A(S_e)$ を初期状態 $s_0$ か

ら，アドバサリ $A$ での $S_e$ への到達確率とする．次に，反例を以下のように定義する．

**定義 2.2** (到達可能性問題の反例)．あるアドバサリ $A$ で非決定を解決した離散時間マルコフ連鎖 $\mathcal{MC}^A$ 上で得られる有限長のパスの集合を $Path^A_{fin}$ とする．$Path^A_{fin}$ のうち，目的状態集合 $S_e$ に到達するパスからなる有限集合を $\Omega \subseteq \{\omega \in Path^A_{fin} | last(\omega) \in S_e\}$ とし，$Prob^A_{fin}(\omega)$ を $\omega \in Path^A_{fin}$ の確率とする．このとき，$\sum_{\omega\in\Omega} Prob^A_{fin}(\omega) > \lambda$ となる $A$ と $\Omega$ が存在するならば，それらを組 $(A, \Omega)$ とし，その組を反例とする． □

反例とは，到達可能性問題の解が "no" となる証拠である．本研究で対象とする到達可能性問題について，以下の定理が成り立つ．

**定理 2.1** (到達確率と到達パスの集合)．$last(\omega)$ を $\omega$ の最後の状態とする．$Prob^A(S_e) > \lambda$ ならば，そのときに限り $S_e$ に到達する有限パスの有限集合 $\Omega \subseteq \{\omega \in Path^A_{fin} | last(\omega) \in S_e\}$ で $\sum_{\omega\in\Omega} Prob^A_{fin}(\omega) > \lambda$ となる反例が存在する． □

# 3 述語抽象化

述語抽象化 [7] は無限状態遷移系の有限の近似を計算し，状態爆発を抑制するために用いられる．本章は文献 [11] のに従い，時間確率システムについて述語抽象化を拡張する．

## 3.1 抽象化述語と述語抽象化

抽象化の手法は文献 [7] と同様であるが，その述語には以下のものを用いる．

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

確率時間 CEGAR では，導入する述語を basis[11] に含まれる述語に限定することで，停止性を保証する．述語を追加する手法およびその定理については 6 章で示す．

## 3.2 抽象モデル

抽象モデルは，文献 [11] を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．

**定義 3.1** (抽象モデルの形成)．$\mathcal{B}$ をビットベクトルの集合，$\alpha$ を抽象化関数，$\gamma$ を具体化関数として，確率時間オートマトン $G = (L, l_0, C, Inv, prob)$ から変換された時間確率システム $\mathcal{M} = (S, s_0, Steps)$ における，述語集合 $\Psi$ による抽象モデル $\mathcal{M}^\sharp = (S^\sharp, s^\sharp_0, Steps^\sharp)$ を以下のように構築する．

- 抽象状態集合 $S^\sharp = L \times \mathcal{B}$
- 初期抽象状態 $s^\sharp_0 = \alpha(s_0)$
- 抽象状態遷移関係 $Steps^\sharp \subseteq S^\sharp \times \mathrm{Dist}(2^C \times S^\sharp)$

$\exists (l, \nu) \in \gamma((l, b)).((l, \nu), \mu) \in Steps$ であるとき，遷移 $((l, b), \mu^\sharp) \in Steps^\sharp$ 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．

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

□

$\mathcal{M}$ が時間の制約を伴うマルコフ決定過程であったのに対し，$\mathcal{M}^\sharp$ は時間のない一般的なマルコフ決定過程であるが，$\mathcal{M}$ 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．

## 3.3 抽象モデルと時間確率システムの関係

$\mu^\sharp_\perp$ を時間遷移を表す確率分布として，$\mathcal{M}^\sharp$ のパスは $\mathcal{M}$ のパスと同様に以下の様に示す．

$$\omega^\sharp = s^\sharp_0 \xrightarrow{\mu^\sharp_0(X_0, s^\sharp_1)} s^\sharp_1 \xrightarrow{\mu^\sharp_1(X_1, s^\sharp_2)} s^\sharp_2 \xrightarrow{\mu^\sharp_2(X_2, s^\sharp_3)} \cdots$$

$\mathcal{M}$ 上の 2 つの状態 $s_1, s_2$ 間の時間遷移は，$\mathcal{M}^\sharp$ では $\alpha(s_1) = \alpha(s_2) = s^\sharp$ であるとき 1 つの抽象状態 $s^\sharp$ 内で行われてしまう．よって反例解析にて $\omega^\sharp$ から $\omega$ を導く際，$\omega^\sharp$ 内の離散遷移 $s^\sharp \xrightarrow{\mu^\sharp}$ はその抽象状態で何らかの時間遷移後に離散遷移した $s_1 \xrightarrow{t,\mu_\perp} s_2 \xrightarrow{0,\mu}$ となる．$\omega$ に対応する $\omega^\sharp$ にはこの 1 つの抽象状態内の時間遷移を含めないことにする．

**定義 3.2** (パスの対応関係)．$\mathcal{M}$ と $\Psi$ によって構成された $\mathcal{M}^\sharp$ において，$\mathcal{M}$ のパス $\omega$ に対応した $\mathcal{M}^\sharp$ のパス $\omega^\sharp$ とは，$\omega$ の全ての遷移に対して，以下の手順によって構成される．

1. $\omega$ の遷移が離散遷移 $s \xrightarrow{0,\mu(X,s')} s'$ ならば，$\omega^\sharp$ の遷移は $\alpha(s) \xrightarrow{\mu^\sharp(X,\alpha(s'))} \alpha(s')$
2. $\omega$ の遷移が時間遷移 $s \xrightarrow{t,\mu_\perp(\emptyset,s')} s'$ かつ $\alpha(s) = \alpha(s')$ ならば，$\omega^\sharp$ の遷移は存在しない．
3. $\omega$ の遷移が時間遷移 $s \xrightarrow{t,\mu_\perp(\emptyset,s')} s'$ かつ $\alpha(s) \neq \alpha(s')$ ならば，$\omega^\sharp$ の遷移は $\alpha(s) \xrightarrow{\mu^\sharp_\perp(\emptyset,s')} \alpha(s')$

また，このような $\omega \in Path_{fin}$ と $\omega^\sharp \in Path^\sharp_{fin}$ の対応を，抽象化関数と同じ記号を用いて，$\alpha_{\mathrm{Path}} : Path_{fin} \to Path^\sharp_{fin}$ と定義する． □

**定義 3.3** (アドバサリの対応関係)．$\mathcal{M}^\sharp$ のアドバサリ $A^\sharp : Path^\sharp_{fin} \to Dist(2^C \times S^\sharp)$ が $\mathcal{M}$ のアドバサリ $A : Path_{fin} \to \mathbb{R}^{\geq 0} \times Dist(2^C \times S)$ に対応しているとは，以下を満たす場合である．

$\forall \omega \in Path^A_{fin}.\forall s \in S.\forall X \subseteq C.A(\omega) = (t, \mu) \wedge A^\sharp(\alpha_{Path}(\omega)) = \mu^\sharp$ に対して，$\mu(X, s) = \mu^\sharp(X, \alpha(s))$

また，このような $A \in Adv$ と $A^\sharp \in Adv^\sharp$ の対応を抽象化関数と同じ記号を用いて，$\alpha_{\mathrm{Adv}} : Adv \to Adv^\sharp$ と定義する． □

定義 3.2 および，定義 3.3 より，次の定理が成り立つ．

**定理 3.1** (パスの対応定理)．あらゆる $\omega \in Path^A_{fin}$ において，いかなる $\Psi$ による $\mathcal{M}^\sharp$ であっても，対応するパス $\alpha_{Path}(\omega) = \omega^\sharp \in Path^{A^\sharp}_{fin}$ および対応するアドバサリ $\alpha_{\mathrm{Adv}}(A) = A^\sharp \in Adv^\sharp$ が存在し，$Prob^A_{fin}(\omega) = Prob^{A^\sharp}_{fin}(\omega^\sharp)$ である．

**定理 3.2** (パスの抽象化). $\mathcal{M}$ において，アドバサリ $A$ によって導出されるパス集合 $Path^A_{fin}$ のうち，異なる 2 つのパス $\omega_1, \omega_2 \in Path^A_{fin}$ について，次の関係が成り立つ．

$$\alpha_{Path}(\omega_1) \neq \alpha_{Path}(\omega_2)$$

### 3.3.1 反例の候補

**定義 3.4** (反例の候補). $G$ と到達可能性問題 $(\lambda, L_e)$ において，$G$ の意味 $\mathcal{M}$ の抽象モデル $\mathcal{M}^\sharp$ の反例の候補とは，$l_e \in L_e$ をもつ $\mathcal{M}^\sharp$ 上の集合を $S^\sharp_e \subseteq S^\sharp$ として，$Prob^{A^\sharp}_{fin}(S^\sharp_e) > \lambda$ となる，$\mathcal{M}^\sharp$ のアドバサリ $A^\sharp$ とパス集合 $\Omega^\sharp$ の組である．反例 $(A, \Omega)$ が反例の候補 $(A^\sharp, \Omega^\sharp)$ に対応しているとは，以下の関係を満たしているときである．

$$\alpha_{Adv}(A) = A^\sharp \wedge \forall \omega \in \Omega. \exists \omega^\sharp \in \Omega^\sharp. \alpha_{Path}(\omega) = \omega^\sharp \qquad \square$$

$Prob^{A^\sharp}_{fin}(S^\sharp_e)$ は $S^\sharp_e$ へ到達可能なパス集合の合計到達確率である．次に，反例と反例の候補に関する 2 つの定理を示す．

**定理 3.3** (合計到達確率の関係). $\mathcal{M}$ 上のアドバサリ $A$ と $\mathcal{M}^\sharp$ 上のアドバサリ $A^\sharp$ が対応している場合，それぞれの到達確率について以下が成り立つ．

$$Prob^{A}_{fin}(S_e) \leq Prob^{A^\sharp}_{fin}(S^\sharp_e)$$

$Prob^{A}_{fin}(S_e)$ は，アドバサリ $A$ における $\mathcal{M}$ での目的状態集合 $S_e$ への到達確率である．

定理 3.3 は，$\mathcal{M}$ において到達確率が最大となるアドバサリを考えた時，$\mathcal{M}$ の最大到達確率が抽象モデル上の最大到達確率以下になることを示す．

**定理 3.4** (反例の存在). $\mathcal{M}$ で反例 $(A, \Omega)$ が存在するならば，必ず $\mathcal{M}^\sharp$ で反例の候補 $(A^\sharp, \Omega^\sharp)$ が存在する．

この定理から $\mathcal{M}^\sharp$ で反例の候補が存在しなければ反例が存在しないため，確率到達可能性問題の解は “yes” となる．

## 3.4 同時実行

ここで，確率時間オートマトンの同時実行性の問題について，図 1 の確率時間オートマトン $G_1$ と，図 2 の抽象化述語 $\Psi$ ($\Psi^{l_0} = \{\text{true}\}, \Psi^{l_1} = \{\text{true}\}, \Psi^{l_2} = \{\text{true}\}, \Psi^{l_e} = \{\text{true}\}$) による抽象モデル $\mathcal{M}^\sharp_0$ を用いて例を示す．

$\mathcal{M}^\sharp_1$ から反例の候補のパス集合 $\Omega^\sharp$ として，2 つのパス $\omega^\sharp_1 = s^\sharp_0 \to s^\sharp_1 \to s^\sharp_e$ と $\omega^\sharp_2 = s^\sharp_0 \to s^\sharp_2 \to s^\sharp_e$ が挙げられたとする．$G_1$ の $\mathcal{M}$ 上において，この 2 つのパスに対応するパスはそれぞれ存在するが $\omega^\sharp_1$ に対応するパス $\omega_1$ では，$l_0$ において 1 単位時間以上の時間遷移をしなければ $l_e$ に到達できないのに対し，$\omega^\sharp_2$ に対応するパス $\omega_2$ では，$l_0$ において時間遷移すると $l_e$ に到達する事はできない．つまり初期状態 $(l_0, \nu_0)$ にて $\omega_1$ でそれぞれ異なるアドバサリで動作している．よって，$(A^\sharp, \Omega^\sharp)$ から $\Omega$ だけでなく $A$ の対応も調べる必要がある．本研究では，後の反例解析にて，同時実行反例解析を提案することでこの問題を解決する．

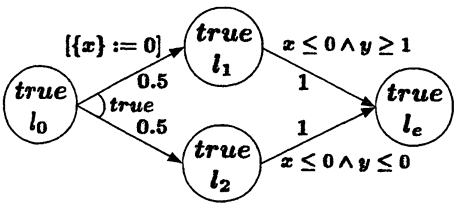


図 1: 確率時間オートマトン $G_1$

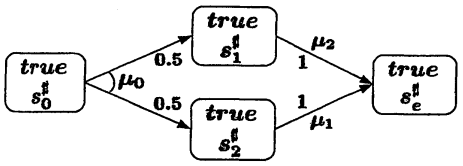


図 2: 抽象モデル $\mathcal{M}^\sharp_0$

# 4 確率時間 CEGAR

述語抽象化及び反例による精錬を自動的に検証に適用していくアプローチが CEGAR(反例による抽象化と精錬)[6] の枠組みである．確率時間 CEGAR の検証の流れを図 3 に示す．

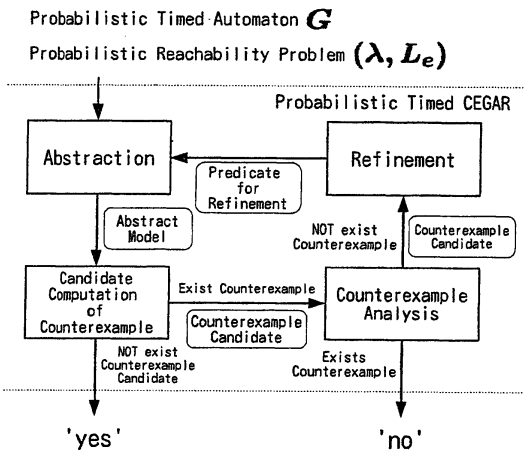


図 3: 確率時間 CEGAR による検証

1. 抽象化：述語集合 $\Psi$ から抽象モデル $\mathcal{M}^\sharp$ を計算する．述語集合 $\Psi$ は，$\forall l \in L. \Psi^l = \{\text{true}\}$ として開始する．

2. 反例の候補の導出：$\mathcal{M}^\sharp_\Psi$ 上で反例の候補 $(A^\sharp, \Omega^\sharp)$ を求める．反例の候補が存在しない場合，“yes” を出力し検証を終了する．

3. 反例解析：反例の候補 $(A^\sharp, \Omega^\sharp)$ に対応する反例 $(A, \Omega)$ が存在するかどうかを求める．存在すれば “no” を出力し，検証を終える．存在しなければ，$(A^\sharp, \Omega^\sharp)$ は具体モデルで実現できないと判断する．このような反例の候補を偽反例とする．
4. 精錬：反例解析の結果，偽反例であればその偽反例 $(A^\sharp, \Omega^\sharp)$ を $\mathcal{M}^\sharp$ から取り除くための新しい述語を導出し，新しい $\Psi$ を得る．
5. (1) に戻る．

これらのサイクルを繰り返していくことにより，最終的にシステムが確率到達可能性問題に対し “yes” か “no” かを判定する．

# 5 反例解析

本章では，確率時間 CEGAR における反例解析の手法を提案する．これは反例の候補 $(A^\sharp, \Omega^\sharp)$ の導出と反例解析によって行われる．

## 5.1 反例の候補の導出

反例の候補を導出する手順を以下に示す．

1. アドバサリを導出: 検証確率 $\lambda$ より大きい到達確率になるアドバサリ $A^\sharp$ を求める．この $A^\sharp$ が存在しなければ反例の候補は存在しない
2. パス集合を導出: $A^\sharp$ におけるパスのうち，パスの合計確率が，検証確率 $\lambda$ より大きくなるような，有限長のパスからなる有限集合 $\Omega^\sharp$ を求める

手順 1. によって検証確率を超える到達確率になるアドバサリ $A^\sharp$ が求まれば，定理 2.1 より必ずパスの集合 $\Omega^\sharp$ は存在し，手順 2. によりそのパス集合を求めることができる．この 2 つの手順により，反例の候補 $(A^\sharp, \Omega^\sharp)$ を導出する．また，ここでは $A^\sharp$ として，最大到達確率になるシンプルなアドバサリを用い，$\Omega^\sharp$ として，パスの数の少ない最小反例 [8] を用いる．

$\mathcal{M}^\sharp$ はマルコフ決定過程であるため，最大到達確率とアドバサリは線形計画法によって計算可能であり [2]．シンプルなアドバサリ [2] となる．

ここで求めた最大到達確率が検証確率以下であれば反例の候補は存在しないと言うことができる．さらに定理 3.4 より，具体モデルにおいて反例が存在しないといえる．従って，最大到達確率が検証確率以下であれば確率到達可能性問題に対して “yes” と答えることができる．

$A^\sharp$ によって $\mathcal{M}^\sharp$ の非決定が解決されると，その動作は離散時間マルコフ連鎖として記述される．そこで，確率 CEGAR[9] でも利用されている，最小反例 (smallest counterexample[8]) を用いる．このような $(A^\sharp, \Omega^\sharp)$ を反例の候補として利用する．

## 5.2 反例解析

### 5.2.1 反例解析の概要

反例解析はパス反例解析と同時実行反例解析の 2 つの手順からなる．

まずパス反例解析では，$\Omega^\sharp$ のそれぞれのパス $\omega^\sharp$ について，対応する $\mathcal{M}$ 上のパス $\omega$ が存在するかどうかを求める．もし，対応するパスがなければ，$\{\omega \in Path_{fin} | \alpha_{Path}(\omega) = \omega^\sharp\} = \emptyset$ となる．このようにして得られた各パス集合から，任意のパスをそれぞれ 1 つ選んで集合 $\Omega$ とし，これを反例のパスの集合とする．この $\Omega$ の全ての組み合わせからなる集合を $\mathbf{\Omega}$ とする．ここで $\Omega^\sharp$ は最小反例であるため，反例の候補のパスのうち，1 つでも $\{\omega \in Path_{fin} | \alpha_{Path}(\omega) = \omega^\sharp\} = \emptyset$ であれば，この反例の候補 $(A^\sharp, \Omega^\sharp)$ は偽反例となることに注意する．

次の同時実行解析では，求めたパスの集合 $\Omega \in \mathbf{\Omega}$ が 1 つのアドバサリ $A$ で構成可能かどうかを確かめる．つまり，$\exists \Omega \in \mathbf{\Omega}. \exists A \in Adv. \Omega \subseteq Path^A_{fin}$ を確かめる．このような $\Omega$ および $A$ が存在しない場合，この反例の候補は偽反例であることがわかる．

それぞれの反例解析によって，導出した $(A^\sharp, \Omega^\sharp)$ が偽反例であると判断された場合，同じ反例の候補を導出しないように，新しい述語を追加して抽象モデルの精錬を行う．

各手順で求められたパス集合 $\Omega$ とアドバサリ $A$ を反例とし，反例が求まれば確率到達可能性問題に対して，“no” という解を示すことができる．

### 5.2.2 準備（ゾーンによる解析）

この反例解析の手法では，$\omega^\sharp$ に対応するパス集合 $\{\omega | \alpha_{Path}(\omega) = \omega^\sharp\}$ を，$\omega^\sharp$ の各抽象状態に対応するゾーンとすることで取り扱う．つまり，ある $\omega^\sharp$ 中の抽象状態 $s^\sharp$ について，そのゾーンに着目し，以下のように取り扱う．

$$\{\nu | \alpha_{Path}(\omega) = \omega^\sharp \wedge \exists i. (\alpha(\omega(i)) = s^\sharp \wedge \omega(i) = (l, \nu))\}$$

定義 3.2 の (2) より，$\omega^\sharp$ に現れない時間遷移を考慮する必要があることに注意する．そのため，集める各抽象状態のゾーンを，時間遷移前のゾーンである到達条件と，時間遷移後のゾーンである出発条件の 2 つとする．$\omega^\sharp$ の $i$ 番目の状態 $(l, b)$ の出発条件を $\zeta^{dep}_{i,\omega^\sharp}$，到達条件を $\zeta^{rea}_{i,\omega^\sharp}$ とする．ここで，反例解析で用いるゾーン演算を定義する．

**定義 5.1** (ゾーン演算)．時間確率システムのゾーンを $\zeta \in Zones(C)$，確率遷移関係を $(l, \zeta_g, p) \in prob$，リセットクロック変数の集合を $X \subseteq C$ として，ゾーンを変形する以

下の演算を定義する.

$$\begin{aligned}
\mathsf{time_succ}[\zeta] &= \{\nu | \exists t \in \mathbb{R}^{\geq 0}.\nu - t \triangleright \zeta\} \\
\mathsf{time_pre}[\zeta] &= \{\nu | \exists t \in \mathbb{R}^{\geq 0}.\nu + t \triangleright \zeta\} \\
\mathsf{reset}[\zeta, X] &= \{\nu[X := 0] | \nu \triangleright \zeta\} \\
\mathsf{free}[\zeta, X] &= \{\nu | \nu[X := 0] \triangleright \zeta\} \\
\mathsf{discrete_succ}[\zeta, \zeta^g, X] &= \mathsf{reset}[\zeta \wedge \zeta^g, X] \\
\mathsf{discrete_pre}[\zeta, \zeta^g, X] &= \mathsf{free}[\zeta, X] \wedge \zeta^g
\end{aligned}$$

□

ここで $(\nu - t)$ は，全てのクロック $x \in C$ について $(\nu - t)(x) = \nu(x) - t$ となるクロック評価である.

time_succ[ζ], time_pre[ζ] はそれぞれ時間遷移の演算である. discrete_succ[$\zeta, \zeta^g, X$] および discrete_pre[$\zeta, \zeta^g, X$] は離散遷移の演算であり，discrete_succ[$\zeta, \zeta^g, X$] は，ガード条件が $\zeta^g$，リセットクロック集合が $X$ である離散遷移によって，あるゾーン $\zeta$ からガード条件およびリセットクロックを考慮して遷移可能なゾーンを返す. discrete_pre[$\zeta, \zeta^g, X$] は同様の離散遷移によって，あるゾーン $\zeta$ へ遷移可能なゾーンを返す.

### 5.2.3 パス反例解析

パス反例解析では，反例の候補の $\Omega^\sharp$ から $\mathbf{\Omega}$ を求めるために，各抽象パス $\omega^\sharp \in \Omega^\sharp$ 上の，各抽象状態の到達条件と出発条件を以下の手順で求める.

1. 目的状態に到達可能な到達条件と出発条件を，$\omega^\sharp$ の目的状態から time_pre または discrete_pre 演算を用いて後方から求める
2. 求めた到達条件と出発条件上で，初期状態から到達可能な到達条件と出発条件を，$\omega^\sharp$ の初期状態から time_succ または discrete_succ 演算を用いて前方から求める

$\omega^\sharp(i)$ を $\omega^\sharp$ の $i$ 番目の抽象状態 $(l_{\omega^\sharp,i}, b^{\omega^\sharp,i})$ とし，このロケーション $l_{i,\omega^\sharp}$ での $G$ での不変条件を $Inv(l_{i,\omega^\sharp})$ とする. また，$i$ 番目の遷移のクロック変数のリセットを $X_{i,\omega^\sharp}$ とし，この遷移に対応する $G$ でのガード条件を $\zeta^g_{i,\omega^\sharp}$ と表現する. 以降，$Inv(l_{i,\omega^\sharp}) \wedge b^{i,\omega^\sharp}\Psi^{l_{i,\omega^\sharp}}$ を単に $\zeta_{i,\omega^\sharp}$ と表記する.

パス反例解析の手順1. のアルゴリズムを Algorithm 1 に示す. 入力は反例の候補の $\Omega^\sharp$ である.

もし $\omega^\sharp(i)$ の出発条件 $\zeta^{dep}_{i,\omega^\sharp}$ が false になった場合 (line:6,11)，この $\omega^\sharp$ 上の $\omega^\sharp(i)$ に対応する状態集合からは目的状態に到達できないことを示しており，偽反例であるとして $\mathcal{M}^\sharp$ を精錬する. 具体的には，

$l$ *s.t.* $\omega^\sharp(i+1) = (l, b)$, $\zeta^{rea}_{i+1,\omega^\sharp}$, time_succ($b^{i,\omega^\sharp}\Psi^{l_{i,\omega^\sharp}}$) or discrete_succ($b^{i,\omega^\sharp}\Psi^{l_{i,\omega^\sharp}}, \zeta^g, X$)

の情報が必要になる.

このように，パスの最初の抽象状態 $\omega^\sharp(0)$ における到達条件 $\zeta^{rea}_{0,\omega^\sharp}$ を求めるまで，この手順を繰り返す.

ここで，$\zeta_0 = \{\nu_0\} \subseteq \zeta^{rea}_{0,\omega^\sharp}$ でなければならないことに注意する. もし，$\omega^\sharp(0)$ の到達条件が false になれば (line:17)，$\omega^\sharp$ によって初期状態 $s_0 = (l_0, \nu_0)$ から目的状態に到達できないことがわかるため，この反例の候補を偽反例とすることができる. 精錬では，

$l_0$ *s.t.* $\omega^\sharp(0) = (l_0, b)$, $\zeta^{rea}_{0,\omega^\sharp}$, $\zeta_0$

の情報が必要になる.

すべての $\omega^\sharp \in \Omega^\sharp$ について，具体モデル上に対応するパスが存在することが分かった場合，exist を出力し，パス反例解析の手順 2. を行う.

**Algorithm 1** パス反例解析 手順 1

```
1: for ω♯ ∈ Ω♯ do
2:   ζ^rea_{|ω♯|,ω♯} ← ζ_{|ω♯|,ω♯}
3:   for (i = |ω♯| − 1, ..., 0) do
4:     if ω♯(i) —μ♯→ is a time transition then
5:       ζ^dep_{i,ω♯} ← time_pre[ζ^rea_{i+1,ω♯}] ∧ ζ_{i,ω♯}
6:       if ζ^dep_{i,ω♯} = false then
7:         return spurious l_{i+1,ω♯}, ζ^rea_{i+1,ω♯},
           time_succ[ζ_{i,ω♯}] ∧ ζ_{i+1,ω♯}
8:       end if
9:     else
10:      ζ^dep_{i,ω♯} ←
           discrete_pre[ζ^rea_{i+1,ω♯}, ζ^g_{i,ω♯}, X_{i,ω♯}] ∧ ζ_{i,ω♯}
11:      if ζ^dep_{i,ω♯} = false then
12:        return spurious l_{i+1,ω♯}, ζ^rea_{i+1,ω♯},
           discrete_succ[ζ_{i,ω♯}, ζ^g_{i,ω♯}, X_{i,ω♯}]∧
           ζ_{i+1,ω♯}
13:      end if
14:    end if
15:    ζ^rea_{i,ω♯} ← time_pre[ζ^dep_{i,ω♯}] ∧ ζ_{i,ω♯}
16:  end for
17:  if ζ^rea_{0,ω♯} ∧ ζ_0 = false then
18:    return spurious l_0, ζ^rea_{0,ω♯}, ζ_0
19:  end if
20: end for
21: return exist
```

反例解析の手順 2. では，$\omega^\sharp$ に対応するパス集合 $\Omega$ の集合である $\mathbf{\Omega}$ を導出する. 反例解析の手順 2. を Algorithm 2 に示す.

Algorithm 1，Algorithm 2 によって，抽象パス $\omega^\sharp$ の各状態における到達条件および出発条件を求めることで，$\omega^\sharp$ に対応する具体モデル上でのパス集合を導出できた. この到達条件と出発条件を用いて，次の同時実行反例解析を行う.

### 5.2.4 同時実行反例解析

同時実行反例解析では，各 $\Omega$ から任意のパス $\omega \in \Omega$ を 1 つずつ選び出して新たな集合 $\Omega'$ とし，$\Omega'$ に含まれる全て

**Algorithm 2** パス反例解析 手順2

1: **for** $\omega^\sharp \in \Omega^\sharp$ **do**
2: $\quad \zeta^{rea}_{0,\omega^\sharp} \leftarrow \zeta_0$
3: $\quad$ **for** $(i = 0, \ldots, |\omega^\sharp| - 1)$ **do**
4: $\quad\quad \zeta^{dep}_{i,\omega^\sharp} \leftarrow \mathsf{time\_succ}[\zeta^{rea}_{i,\omega^\sharp}] \wedge \zeta^{dep}_{i,\omega^\sharp}$
5: $\quad\quad$ **if** $\omega^\sharp(i) \xrightarrow{\mu_\sharp}$ is a time transition **then**
6: $\quad\quad\quad \zeta^{rea}_{i+1,\omega^\sharp} \leftarrow \mathsf{time\_succ}[\zeta^{dep}_{i,\omega^\sharp}] \wedge \zeta^{rea}_{i+1,\omega^\sharp}$
7: $\quad\quad$ **else**
8: $\quad\quad\quad \zeta^{rea}_{i+1,\omega^\sharp} \leftarrow$
$\quad\quad\quad \mathsf{discrete\_succ}[\zeta^{dep}_{i,\omega^\sharp}, \zeta^{g}_{i,\omega^\sharp}, X_{i,\omega^\sharp}] \wedge \zeta^{rea}_{i+1,\omega^\sharp}$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: **end for**

のパスを導出可能なアドバサリ $A$ が存在するかを確かめる．

アドバサリはパスを入力することで，次の遷移における時間遷移量と確率分布を返すため，$A$ で導出可能なパス集合 $\Omega$ に含まれる任意の2つのパス $\omega_1, \omega_2 \in \Omega$ について，それぞれのプレフィックス $\omega_1', \omega_2'$ が $\omega_1' = \omega_2'$ であるとき，$\omega_1', \omega_2'$ からの時間遷移量と確率分布が等しいと言える．すなわち同時実行反例解析では，パス集合 $\Omega$ に含まれる各パス $\omega \in \Omega$ について，等しいプレフィックス $\omega'$ があるならば，その次の遷移は，同じ時間経過量による時間遷移，または同じ確率分布による離散遷移であることを確かめれば良い．

同時実行反例解析では，共通のプレフィックスを持つパスについて，各状態の出発条件，到達条件それぞれについて積を取ることで，そのプレフィックスの出発条件 $\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$ と到達条件 $\zeta^{rea}_{\omega^\sharp_{i\text{-}th}}$ をゾーンとして求める．

同時実行反例解析では，いずれかのプレフィックスの出発条件または到達条件が false となったとき，複数のパスで共通の時間経過量によって具体モデル上で実行できないとして，反例の候補を偽反例とし，精錬を行う．

同時実行反例解析のアルゴリズムを Algorithm 3 に示す．アルゴリズムに対する入力は，パス反例解析の結果得られた $\omega^\sharp \in \Omega^\sharp$ の各抽象状態における到達条件と出発条件である．出力は $(A, \Omega)$ が存在していれば exist，偽反例であると分かれば spurious を返す．このアルゴリズムは，Algorithm 2 で求めた，各パスの各抽象状態における到達条件および出発条件を利用し，前方からプレフィックスに対する出発条件と到達条件を time_succ または discrete_succ 演算によって計算する．

もし実行中にゾーンが false になれば (line:15-17) 偽反例と判断し，$l_{i,\omega}$, $\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$, $\zeta^{dep}_{i,\omega^\sharp}$ の情報を用いて精錬を行う (line:16)．

全てのプレフィックスの出発条件および到達条件が false でなければ，exists と出力され (line:27)，反例が存在する事が分かり，確率到達可能性問題に対して "yes" と答えることができる．

**Algorithm 3** 同時実行反例解析

1: **for** $\omega^\sharp \in Path^\sharp_{fin}$ **do**
2: $\quad \zeta^{rea}_{\omega^\sharp} \leftarrow \mathrm{true}$
3: $\quad \zeta^{dep}_{\omega^\sharp} \leftarrow \mathrm{true}$
4: **end for**
5: $\zeta^{dep}_{\omega^\sharp = s_0} \leftarrow \zeta_0$
6: **for** $i = 0, \cdots, C_{\Omega_{\max}}$ **do**
7: $\quad$ **for** $\omega^\sharp \in \Omega^\sharp$ **do**
8: $\quad\quad$ **if** $|\omega^\sharp| > i$ **then**
9: $\quad\quad\quad \zeta^{rea}_{\omega^\sharp_{i\text{-}th}} \leftarrow \zeta^{rea}_{\omega^\sharp_{i\text{-}th}} \wedge \zeta^{rea}_{i,\omega^\sharp}$
10: $\quad\quad\quad \zeta^{dep}_{\omega^\sharp_{i\text{-}th}} \leftarrow \mathsf{time\_succ}[\zeta^{rea}_{\omega^\sharp_{i\text{-}th}}] \wedge \zeta_{i+1,\omega^\sharp}$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad$ **for** $\omega^\sharp \in \Omega^\sharp$ **do**
14: $\quad\quad$ **if** $|\omega^\sharp| > i$ **then**
15: $\quad\quad\quad$ **if** $\zeta^{dep}_{\omega^\sharp_{i\text{-}th}} \wedge \zeta^{dep}_{i,\omega^\sharp} = \mathrm{false}$ **then**
16: $\quad\quad\quad\quad$ **return** spurious $l_{i,\omega^\sharp}, \zeta^{dep}_{\omega^\sharp_{i\text{-}th}}, \zeta^{dep}_{i,\omega^\sharp}$
17: $\quad\quad\quad$ **end if**
18: $\quad\quad\quad \zeta^{dep}_{\omega^\sharp_{i\text{-}th}} \leftarrow \zeta^{dep}_{\omega^\sharp_{i\text{-}th}} \wedge \zeta^{dep}_{i,\omega^\sharp}$
19: $\quad\quad\quad$ **if** $\omega^\sharp(i) \xrightarrow{\mu^\sharp}$ is a time transition **then**
20: $\quad\quad\quad\quad \zeta^{rea}_{\omega^\sharp_{(i+1)\text{-}th}} \leftarrow \mathsf{time\_succ}[\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}] \wedge \zeta_{i+1,\omega^\sharp}$
21: $\quad\quad\quad$ **else**
22: $\quad\quad\quad\quad \zeta^{rea}_{\omega^\sharp_{(i+1)\text{-}th}} \leftarrow$
$\quad\quad\quad\quad \mathsf{discrete\_succ}[\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}, \zeta^{g}_{i,\omega^\sharp}, X_{i,\omega^\sharp}] \wedge$
$\quad\quad\quad\quad \zeta_{i+1,\omega^\sharp}$
23: $\quad\quad\quad$ **end if**
24: $\quad\quad$ **end if**
25: $\quad$ **end for**
26: **end for**
27: **return** exist

次章では，これら反例解析および同時実行解析の情報を用いて精錬を行う手法を示す．

## 6 精錬

反例解析において，反例の候補が偽反例であることが分かった場合，精錬を行う．精錬では，その偽反例を導出する原因となった抽象パスを持たない抽象モデル $\mathcal{M}^\sharp$ を構築できる新しい述語 $\Psi$ を導出する．

具体的には，ある1つのロケーションと，そのロケーションに追加するための1つ以上の述語を導出する．述語は，ある2つのゾーンを分割可能な述語とする．

### 6.1 パス反例解析による精錬

#### 6.1.1 $\omega^\sharp(i)$ の出発条件 $\zeta^{dep}_{i,\omega^\sharp}$ が false になった場合

抽象モデル構築において $\omega^\sharp(i)$ から $\omega^\sharp(i+1)$ への遷移が構築されるのは，$\omega^\sharp(i)$ に含まれる状態から $\omega^\sharp(i+1)$ に含ま

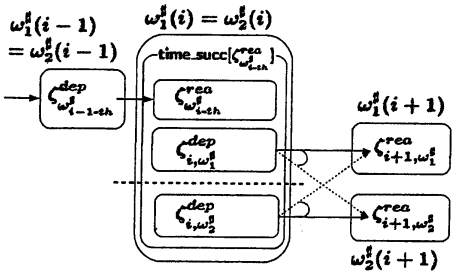


図 4: 同時実行反例解析における精錬

れる状態への遷移が存在するためであるが，このパスを実行するための状態間には遷移が存在しない事が原因である．

よって，この抽象状態のゾーン $\zeta_{i+1,\omega^\sharp}$ を $\zeta^{rea}_{i+1,\omega^\sharp}$ を含むゾーンと $\mathsf{discrete\_succ}(\zeta_{i,\omega^\sharp}, \zeta^g, X)$ または $\mathsf{time\_succ}(\zeta_{i,\omega^\sharp})$ を含むゾーンに分割することで，このパスを取り除くことができる．

#### 6.1.2 $\zeta^{rea}_{0,\omega^\sharp}$ と $\zeta_0$ の積が false になった場合

具体モデル上の全てのパスは $\nu_0$ から始まる．よって，全ての抽象パスは $\zeta_0 = \{\nu_0\}$ を含む抽象状態から始まるが，抽象パス $\omega^\sharp$ の最初の抽象状態 $\omega^\sharp(0)$ の到達条件 $\zeta^{rea}_{0,\omega^\sharp}$ に $\zeta_0$ 含まれていないため，具体モデルでは初期状態 $\nu_0$ からこのパスを実行できないことがわかる．

よって，この抽象状態のゾーン $\zeta_{0,\omega^\sharp}$ を $\zeta_0$ を含むゾーンと $\zeta^{rea}_{0,\omega^\sharp}$ を含むゾーンに分割することで，このパスを取り除くことができる．

### 6.2 同時実行反例解析による精錬

同時実行反例解析では，$\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$ が false になることと，反例の候補である全てのパスを同じアドバサリで実行できないことは同値である．図 4 を用いてこれを説明する．抽象パス $\omega^\sharp_1$ および $\omega^\sharp_2$ はそれぞれの $i$ 番目までの遷移において共通のプレフィックス $\omega^\sharp_{i\text{-}th}$ を持つとし，$i+1$ 番目の遷移において，初めて異なる抽象状態へ遷移するものとする．

パス反例解析の Algorithm 1 および Algorithm 2 によって，$\zeta^{rea}_{i+1,\omega^\sharp_1}$ に遷移可能な状態集合として，$\zeta^{dep}_{i,\omega^\sharp_1}$ を計算した．つまり $\zeta^{dep}_{i,\omega^\sharp_1}$ に含まれる状態からのみ $\zeta^{rea}_{i+1,\omega^\sharp_1}$ に含まれる状態へ遷移できる．$\omega^\sharp_2$ についても同様である．よって，$\zeta^{dep}_{i,\omega^\sharp_1}$ に含まれるが，$\zeta^{dep}_{i,\omega^\sharp_2}$ には含まれない状態からは $\zeta^{rea}_{i+1,\omega^\sharp_2}$ へ遷移できない．

アドバサリは，あるパスを入力することで次の遷移における時間遷移量と確率分布を返すものであった．すなわち，あるアドバサリが存在するとは，全ての状態においてその時間遷移量と確率分布が一意に決まる事を意味する．

ところが，$\zeta^{dep}_{i,\omega^\sharp_1}$ と $\zeta^{dep}_{i,\omega^\sharp_2}$ に共通部分が無い場合，$\zeta^{rea}_{i+1,\omega^\sharp_1}$ と $\zeta^{rea}_{i+1,\omega^\sharp_2}$ の両方へ遷移可能な状態は存在しない．そのため，図 4 では $\zeta^{dep}_{i,\omega^\sharp_1}$ に含まれる状態から $\omega^\sharp_1$ に対応するパスを導出するためには $\zeta^{rea}_{i+1,\omega^\sharp_1}$ に含まれる状態へ遷移する確率分布を選択しなければならず，$\omega^\sharp_2$ に対応するパスを導出するためには，$\zeta^{dep}_{i,\omega^\sharp_2}$ に含まれる状態へ時間遷移し，そこから $\zeta^{rea}_{i+1,\omega^\sharp_2}$ に含まれる状態へ遷移する確率分布を選択しなければならない．つまり，両方のパスを導出可能な確率分布は存在しないことになる．従って，これらのパスを導出可能なアドバサリが存在しないことが分かる．

このように，対応するパス集合を同一のアドバサリによって実行できない事が分かった場合，$\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$ と $\zeta^{dep}_{i,\omega^\sharp}$ を分割することで精錬を行う．なぜなら，これらを分割する述語を加えることで，これまで $\omega^\sharp(i)$ のゾーン内に隠れていた $\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$ と $\zeta^{dep}_{i,\omega^\sharp}$ 間の時間遷移が抽象モデル上に確率分布として現われるようになるためである (図 4 では，$\zeta^{dep}_{i,\omega^\sharp_1}$ から $\zeta^{dep}_{i,\omega^\sharp_2}$ への遷移が時間遷移としてモデル上に現われた)．よって，反例解析の $A^\sharp$ の導出において，$\zeta^{dep}_{\omega^\sharp_{i\text{-}th}}$ または $\zeta^{dep}_{i,\omega^\sharp}$ を含む抽象状態は，これまで存在した抽象確率分布に加え，この新たに現われた時間遷移の抽象確率分布を選択できるようになり，これらの確率分布からいずれか 1 つが選ばれるため，同様の同時実行の問題は発生しない．

### 6.3 新たな述語の導出と検証の停止性

**定理 6.1** (サイクル毎に新たな述語を導出可能)．確率時間 *CEGAR* は，到達可能性問題の解が得られるまで，サイクル毎に新たな述語を導出可能である．

**定理 6.2** (検証の停止性)．確率時間 *CEGAR* による検証は有限回数のサイクルで解を得ることができる．

## 7 実験

本章では，確率時間 CEGAR のプロトタイプを Scala によって実装し，実験を行って既存手法とその状態数について比較する．

### 7.1 述語の選択

確率時間 CEGAR を実装するにあたり，精錬における述語選択のアルゴリズムを決定する必要がある．ゾーンの実装として DBM を用いることで，ゾーンによって表される領域の辺を容易に取り扱うことができる．これを利用し，述語の選択は以下のアルゴリズムにより行う．

1. 分割したい 2 つのゾーンを構成する辺を元とする集合から冪集合を得る
2. その冪集合の元のうち，要素が少ないものから順に，2 つのゾーンが分割 できるものを探す

CEGAR における述語抽象化では，分割に用いる述語の増加に応じて抽象モデルの状態数が増える．そのため，このアルゴリズムは，2 つのゾーンを分割可能な述語の組み合

| $D$ | PT CEGAR | Symbolic MC | 状態数比 |
|---|---|---|---|
| 2000 | 10 | 15 | 0.6667 |
| 4000 | 14 | 25 | 0.5600 |
| 6000 | 18 | 47 | 0.3830 |
| 8000 | 22 | 81 | 0.2716 |
| 10000 | 26 | 126 | 0.2063 |
| 20000 | 46 | 528 | 0.0871 |
| 30000 | 63 | 1206 | 0.0522 |
| 40000 | 78 | 2168 | 0.0360 |
| 50000 | 93 | 3426 | 0.0271 |
| 60000 | 108 | 4964 | 0.0218 |

表 1: 状態数の比較: FireWire root contention protocol

わせのうち，より少ない数の述語で分割しようとするものである．

さらに，述語として $x_1 - x_2 \leq d$ や $x_1 - x_2 < d$ のような diagonal 制約を優先して用いる．これは，diagonal 制約を用いないと，分割した状態間に時間遷移が生じる場合があり，これに起因して必要のない状態が生成され，状態数が増加することが考えられるためである．

### 7.2 実験モデルの制限

DBM のように，モデルに表れる最大定数を用いてゾーンを表現する場合，前方解析が正しく行えない事が知られているため [3]，本実験では，クロック変数が 3 以下のモデルのみを対象とする．

### 7.3 FireWire root contention protocol

本実験では Symbolic model checking[10] で対象とされている IEEE 1394 FireWire root contention protocol に対し，デッドラインが 2000 から 60000 まで同様の性質，すなわち等価な到達可能性問題を $(\lambda, L_e) = (0, \{elect\})$ について検証することで，その状態数の比較を行う．

### 7.4 実験結果

表 1 に実験結果を示す．全ての Dead Line について，既存手法よりも少ない状態数で検証を終えている．また，状態数比の推移から，Dead Line の増加に伴い，状態数がより削減されていることがわかる．

## 8 まとめ

本論文では，確率時間オートマトンの到達可能性解析において，CEGAR による枠組みを適用した検証手法を確立した．本論文の主な貢献は以下の 3 点である．

- 確率時間オートマトンの到達可能性解析において，CEGAR を導入することにより，検証対象に応じた状態空間の構築を可能にした
- 確率時間オートマトンにおいて，確率分岐による複数パスの同時の実行可能性を同時実行反例解析として判定し，偽反例による抽象モデルの精錬手法を実現した
- 本手法を実装して実験を行い，既存手法と比較し，より少ない状態数での検証が可能であることを示した

## 参考文献


[1] Bengtsson, J. and Yi, W.: Timed Automata: Semantics, Algorithms and Tools, *LNCS*, Vol. 3098, pp. 87–124 (2004).

[2] Bianco, A. and de Alfaro, L.: Model checking of probabilistic and nondeterministic systems, *LNCS*, Vol. 1026, pp. 499–513 (1995).

[3] Bouyer, P.: Untameable Timed Automata!, *LNCS*, Vol. 2607, pp. 620–631 (2003).

[4] Clarke, E. M., Grumberg, O. and Peled, D.: Model Checking, MIT (2000).

[5] Clarke, E. M., Fehnker, A., Han, Z., Krogh, B. H., Ouaknine, J., Stursberg, O. and Theobald, M.: Abstraction and Counterexample-Guided Refinement in Model Checking of Hybrid Systems, *IJFCS*, Vol. 14, No. 4, pp. 583–604 (2003).

[6] Clarke, E. M., Grumberg, O., Jha, S., Lu, Y. and Veith, H.: Counterexample-Guided Abstraction Refinement, *LNCS*, Vol. 1855, pp. 154–169 (2000).

[7] Graf, S. and Saïdi, H.: Construction of Abstract State Graphs with PVS, *LNCS*, Vol. 1254, pp. 72–83 (1997).

[8] Han, T. and Katoen, J. P.: Counterexamples in probabilistic model checking, *LNCS*, Vol. 4424, pp. 72–86 (2007).

[9] Hermanns, H., Wachter, B. and Zhang, L.: Probabilistic CEGAR, *LNCS*, Vol. 5123, pp. 162–175 (2008).

[10] Kwiatkowska, M., Norman, G., Sproston, J. and Wang, F.: Symbolic model checking for probabilistic timed automata, *Information And Computation*, Vol. 205, No. 7, pp. 1027–1077 (2007).

[11] Moller, M. O., Rues, H. and Sorea, M.: Predicate Abstraction for Dense Real-Time Systems, *Electronic Notes in Theoretical Computer Science*, Vol. 65, No. 6, pp. 218–237 (2002).