数理解析研究所講究録 1426 巻 2005 年 133-139

確率時間オートマトンの確率時間強模倣検証アルゴリズム

Verification Algorithm of Probabilistic Timed Simulation for Probabilistic Timed Automata

小寺 広志* 山根 智†

Hiroshi Kodera Satoshi Yamane

金沢大学大学院 自然科学研究科‡

Graduate School of Natural Science and Technology, Kanazawa University

概要 2003年，*S. Yamane* により次の判定問題が定義され，その決定可能性が示された：「ソフトリアルタイムシステムを表現・解析する1つの有用な道具である確率時間オートマトンというモデル上において，時間強模倣関係と確率模倣関係の組合せとして定義される確率時間強模倣関係が，初期状態対を含んでいるか否か」．但し，判定アルゴリズムは与えられていない．そこで，本論文では，具体的な判定アルゴリズムを提案する．その基本的考え方は，注意深く選んだ初期集合から確率時間強模倣の条件を充たさない状態対（反例）を段階的に取除き，残った集合内に初期状態対が含まれるか否かを判定する，というものである．

1 はじめに

研究の位置付け

形式的検証：近年，コンピュータソフトウェアの巨大化（e.g. 携帯用 OS の多機能化），並列化（e.g. グリッドコンピューティング），分散化（e.g. インターネットの検索エンジンなどの Web サービス）；一言で言えば複雑化，及び，社会環境のコンピュータシステムへの依存が強まっていく傾向に伴い，ソフトウェアの信頼性の保証が強く要求されている．この要求に対する1つの解答の試みが形式的検証の理論であり，計算機科学におけるオートマトン理論の応用分野の1つでもある．形式的検証法とは，検証対象とするシステムを数学モデルで表現し（e.g. グラフやオートマトンによりシステムをモデル化し），その上で状態空間を網羅的に探索することでバグの取り残しがないように検証する，という方法である．その他にも，数理論理学における証明論を基礎とした型理論，シーケント計算などによりシステムの正しさを証明する，という検証の方向もある．形式的検証法は，"テスト" などの，人間の直観と経験と推理力に頼った有限パターンの検査とは異なり，バグの取り残しはないが，"モデル化" の問題と網羅的な探索を行うが故の "計算量" の問題がある．現実的にはテストと併用して小規模な部分システムに適用するのが効果的だと考えられる．

確率時間オートマトンというモデル：通信プロトコル，プロセス間通信などにおける時間制約（e.g. 受信待ち時間；送信開始から3単位時間未満なら待つが3単位時間以上は待たない）や動作に対する確率的条件（e.g. 送信エラー；送信データがロストする確率が 0.5%，ロストしない確率が 99.5%）を記述し，ロジック PCTL で記述される性質（e.g. $q_{send}, \mathcal{E} \models_{Adv} z.[true\ \exists \mathcal{U}\ a_{send} \wedge (z \leq 5)]_{\geq 0.8} \iff$ スケジューラ Adv により決められた動作において，初期状態から送信開始状態 q_{send} まで5単位時間内に確率 0.8 以上で到達可能）が成立つか否かを自動的に解析できる道具として確率時間オートマトンという数学モデルがある [7]．このモデルは，ω オートマトン [2] を拡張した時間オートマトン [2], [6] の更なる拡張として考案されていて，理論的バックボーンがある．本論文では，現実のシステムをモデル化する考え方や方法を開発するのではなく，確率時間オートマトンという形式でモデルが与えられたという前提の下で議論を行う．

模倣関係を使った検証：近年の複雑なコンピュータシステムの多くが 並列分散システム や リアルタイムシステム で構成されていることから，これらのシステムが検証対象となる．分散システムの記述には非決定性を用いることが有効であることが知られている（e.g. プロセス間の相対速度を無視できる．仕様記述の際の実装の自由度の範囲設定に使える）[11]．一方，リアルタイムシステムのオートマトンによるモデル化には，時間オートマトンが標準的である．これらより，オートマトン理論を使う限りにおいては，非決定性の時間オートマトン（或いは非決定性の確率時間オートマトン）でシステムをモデル化しそれを検証するという流れが自然である．一方で，非決定性の時間オートマトンは補集合に対し閉じていない（時間言語の補集合の計算は決定不能）[2]．従って，非決定性を含む時間オートマトンに対して，$L(\mathcal{A}_{imp}) \subseteq L(\mathcal{B}_{spec}) \iff L(\mathcal{A}_{imp}) \cap L(\mathcal{B}^c_{spec}) = \{\}$ という性質に基付く言語包含関係を用いた検証は出来ない．これに対し，模倣関係を用いた検証は決定可能である（模倣関係と言語包含関係は相関があり，"模倣関係が成立つならば言語包含関係も成立つ" という性質がある）．これより，模倣関係は上の意味で言語包含関係よりも優れた検証法であると言える．模倣関係と検証のつながりを直感的に述べると，仕様 $\mathcal{B}$ が実装 $\mathcal{A}$ の任意動作を真似ることが出来るならば，$\mathcal{B}$ には $\mathcal{A}$ に対応する構造が全て含まれていて，$\mathcal{B}$ 上で調べたい特性（e.g. 到達可能性）が成立っていれば，$\mathcal{A}$ 上でもその特性が成立つ．そして，$\mathcal{B}$ 上での検証コストが $\mathcal{A}$ 上のそれよりも小さければ，$\mathcal{A}$ 上での検証を $\mathcal{B}$ 上での検証で代替えすることで，検証項目が多いほど全体の検証コストの低減が図れる．模倣関係は言わば，検証の補助的な役割を担う．

以上述べたように，本研究は大枠としては形式的検証の研究に含まれ，確率時間オートマトンというモデル上での確率時間強模倣関係を用いた検証に関係している．より明確な位置付けは，以下の関連研究の紹介の中で述べる．

*E-mail: kodera@csl.ec.t.kanazawa-u.ac.jp

†E-mail: syamane@is.t.kanazawa-u.ac.jp

‡〒 920-1192 石川県金沢市角間町 金沢大学大学院 自然科学研究科

関連研究の紹介

1. 1996 年 ; S. Tasiran, R. Alur らにより，時間オートマトン上での時間強模倣関係が定義され，その決定可能性が示された [5].
2. 1999 年 ; M. Kwiatkowska らにより，モデル確率時間オートマトンが提案され，そのモデル上での Model-Checking アルゴリズムが提案された [7].
3. 1999 年 ; C. Baier らにより，確率オートマトン上での確率模倣関係を解くアルゴリズムが提案された [8].
4. 2003 年 ; S. Yamene により，確率時間オートマトン上で確率時間強模倣関係が定義された．またその関係を計算するアルゴリズムの決定可能性の証明も行われている [12]．但し，この論文では具体的なアルゴリズムは与えられていない．
5. 2003 年 ; R. Lanotte らにより "確率分布の非決定性のない" 確率時間オートマトン上で弱双模倣関係が定義され，その関係を計算するアルゴリズムが提案されている [13]．但し，彼らは双模倣関係に特化した Partition アルゴリズムを採用していて，そのアルゴリズムは一方向性の模倣関係の計算には適用できない．また，彼らの扱っているモデルは本論文で扱うモデルよりも表現力が低い．

本研究の直接の関連は 4. である．論文 4. と他の論文との関連を述べると，次の通りである．

- 論文 4. では，2. をベースにして確率時間オートマトンの定義が与えられている．
- また，1. の時間強模倣関係と 3. の確率模倣関係の組合せとして確率時間強模倣関係の定義が与えられている．
- 更に，決定可能性の証明には 1. の証明テクニックが用いられている．

研究の目的

本論文では，論文 4. で述べられていない確率時間強模倣関係の計算アルゴリズムを構成することを目的とする．

研究の価値

確率時間強模倣関係よりも緩い関係である，確率時間弱模倣関係は段階的詳細化検証に応用出来る．ここに，確率時間強模倣検証アルゴリズムは確率時間弱模倣検証アルゴリズムを考えるための基礎になる．一方，理論面においては，反例法に基く一方向性の模倣検証アルゴリズムは確率時間オートマトン上ではまだ開発されていないため，新規性がある．

本論文の構造

先ず 2 章で，確率時間オートマトンと確率時間強模倣関係，そして確率時間強模倣検証問題をそれぞれ定義する．次に 3 章で，確率時間強模倣検証アルゴリズムに関して，アルゴリズムの原理 (正当性) とその具体的構成とアルゴリズムの停止性を述べる．最後に 4 章で，まとめと今後の課題を述べる．

2 確率時間オートマトンと確率時間強模倣関係の定義

本章では，確率時間オートマトンと確率時間強模倣関係の定義を行う．そして，確率時間強模倣検証問題を定義する．

2.1 確率時間オートマトンの Syntax と Semantics の定義

オートマトンは "形式的な構造" と "その構造上での動作" の 2 つを規定すれば厳密に定義される．

確率時間オートマトン (Probabilistic Timed Automata ; PTA) は時間オートマトンに "離散確率分布" が付加され拡張されたモデルであり，時間オートマトンは有限オートマトンに "クロック変数という時間を測定する変数" が付加されたモデルである [7].

Definition 2.1 *(PTA の Syntax)*

PTA $\mathcal{A}$ は 7 つ組 $(S, \bar{s}, \Sigma, \mathcal{X}, inv, prob, \langle \tau_s \rangle_{s \in S})$ である．ここで，

1. S はロケーションの有限集合．
2. $\bar{s} \in S$ は初期ロケーション．
3. Σ はアクションラベルの有限集合．
4. $\mathcal{X}$ は非負実数値を取る変数の有限集合で，各変数をクロック変数と呼ぶ．
5. $inv : S \to \mathbf{Z}_{\mathcal{X}}$ は各ロケーションに時間制約 (不変条件) を与える関数．
6. $prob : S \to 2^{\Sigma \times 2^{\mathcal{X}} \times FDist\left(S \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\right)}$ は，各ロケーションに対し，そのロケーションから外向きに出る辺の有限集合を与える関数．
7. $\langle \tau_s \rangle_{s \in S}$ は，ロケーション s から外向きに出ている各辺に対し時間制約 (ガード条件) を与える関数 $\tau_s : prob(s) \to \mathbf{Z}_{\mathcal{X}}$ の有限集合．

但し，

$$\mathbf{Z}_{\mathcal{X}} \stackrel{def}{=} \left\{ \bigwedge_{0 \leq i \neq j \leq |\mathcal{X}|} x_i - x_j \prec c \;\middle|\; \begin{array}{l} x_0 = 0, \\ x_k \in \mathcal{X} (k \in \{1, 2, \cdots, |\mathcal{X}|\}), \\ \prec \in \{<, \leq\},\ c \in \mathbb{Z} \end{array} \right\}$$

また，$\mu \in FDist(\mathcal{Q}) \stackrel{def}{\Longleftrightarrow} \mu : \mathcal{Q} \to Dist(\mathcal{Q})$ と定義し，

$$Dist(\mathcal{Q}) \stackrel{def}{=} \left\{ p \;\middle|\; p : \mathcal{Q} \to [0, 1] \text{ s.t. } \sum_{q \in \mathcal{Q}} p(q) = 1 \right\}$$

は離散確率分布の集合である．$\mu \in FDist(\mathcal{Q})$ に $q \in \mathcal{Q}$ をパラメータとして渡すと，$\mathcal{Q}$ 上の離散確率分布 $\mu(q) \in Dist(\mathcal{Q})$ が得られる．上記において，記号 $\stackrel{def}{=}$ は左辺を右辺で定義するという意味で用い，記号 $\stackrel{def}{\Longleftrightarrow}$ は左辺と右辺を同値の条件として定義するという意味で用いてある．

このPTA は，動作に対する確率的条件と時間制約の両方を表現できる．但し，"動作に対する確率的条件" の表現は，各辺から "ロケーションの集合上の離散確率分布" への遷移という形で表現されるものに限る．また，"時間制約の表現" は，クロック変数間の線型不等式という形で表現されるものに限る．

ある時刻における次の動作範囲を規定する情報を状態と言う．PTA における状態はロケーションと各クロック変数の値の対で定義される．

Definition 2.2 *(PTA の状態)*

集合 $states(\mathcal{A}) \stackrel{def}{=} \bigcup_{s \in S} \{\langle s, \mathbf{a} \rangle \mid \mathbf{a} \in [\![inv(s)]\!]\} \subset S \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ の元を *PTA* の状態と言う．但し，$[\![\cdot]\!] : \mathbf{Z}_{\mathcal{X}} \to \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ は時間制約を表す文字列の解釈である．

Definition 2.3 *(PTA の初期状態)*

状態 $\bar{q} \stackrel{def}{=} \langle \bar{s}, (0, \cdots, 0) \rangle$ を初期状態と言い，任意の *PTA* $\mathcal{A}$ の状態集合 $states(\mathcal{A})$ は初期状態を含むものとする．

オートマトンの動作とは状態遷移の系列である．このため，1 回の状態遷移が定義されれば動作が定義される．

Definition 2.4 *(PTA の Semantics)*
PTA における *1* 回の状態遷移は，次の *2* 通りの遷移集合の和集合の元である．

1. 時間遷移 *(Timed Transitions)*

$$\langle s, \mathbf{a}\rangle \xrightarrow{\delta} \langle s, \mathbf{a}+(\delta,\cdots,\delta)\rangle$$
$$\stackrel{def}{\Longleftrightarrow}$$
$$\exists \delta \in \mathbb{R}_{>0} \text{ s.t. } \forall \delta' \in \mathbb{R}_{>0}\{ \quad \delta' \leq \delta \ \Rightarrow\ \mathbf{a}+(\delta',\cdots,\delta') \in [\![inv(s)]\!] \ \}$$

2. 離散遷移 *(Discrete Transitions)*

$$\langle s, \mathbf{a}\rangle \xrightarrow{(\sigma,\lambda,\mu)} \langle s', \mathbf{a}[\lambda := 0]\rangle$$
$$\stackrel{def}{\Longleftrightarrow}$$
$$\exists (\sigma,\lambda,\mu) \in prob(s) \text{ s.t. } \mathbf{a} \in [\![\tau_s(\sigma,\lambda,\mu)]\!] \wedge \exists s' \in S \text{ s.t. } \{ \mathbf{a}[\lambda := 0] \in [\![inv(s')]\!] \wedge \mu(\langle s, \mathbf{a}\rangle)(\langle s', \mathbf{a}[\lambda := 0]\rangle) > 0 \ \}$$

ここで，$\mathbf{a}[\lambda := 0]$ は $\lambda \subset \mathcal{X}$ で指定される各クロック変数の値を全て 0 にリセットしたものを表す．また，上記の記号 $\stackrel{def}{\Longleftrightarrow}$ の下の方の条件はそれぞれ，遷移前の事前条件と遷移後の事後条件の両方を含んでいることに注意．

2.2 確率時間強模倣関係の定義

確率時間強模倣関係を定義する．これは時間強模倣関係 [5] と確率模倣関係 [4] の組合せとして与えられる [12].

Definition 2.5 *(確率時間強模倣関係)*
$PTA\ \mathcal{A} = \left(S_A, \overline{s}_A, \Sigma_A, \mathcal{X}_A, inv_A, prob_A, \langle \tau_s^A \rangle_{s\in S_A}\right)$, $\mathcal{B} = \left(S_B, \overline{s}_B, \Sigma_B, \mathcal{X}_B, inv_B, prob_B, \langle \tau_s^B \rangle_{s\in S_B}\right)$ それぞれの状態集合上で次の条件を充たす二項関係 $R \subset states(\mathcal{A}) \times states(\mathcal{B})$ を確率時間強模倣関係と言う．また，確率時間強模倣関係 R の中で最大の集合を $R_{[\mathcal{A}\times\mathcal{B}]}$ で表す．

全ての $(\langle s_1, \mathbf{a}\rangle, \langle s_2, \mathbf{b}\rangle) \in R$ に対して，

1. 時間模倣条件：
$\forall \delta \in \mathbb{R}_{>0}$,
$\langle s_1, \mathbf{a}\rangle \xrightarrow{\delta} \langle s_1, \mathbf{a}+(\delta,\cdots,\delta)\rangle$ ならば，
$\langle s_2, \mathbf{b}\rangle \xrightarrow{\delta} \langle s_2, \mathbf{b}+(\delta,\cdots,\delta)\rangle$
$\wedge\ (\langle s_1, \mathbf{a}+(\delta,\cdots,\delta)\rangle, \langle s_2, \mathbf{b}+(\delta,\cdots,\delta)\rangle) \in R.$

2. 確率模倣条件：
全ての $(\sigma,\lambda_1,\mu_1) \in prob_{\mathcal{A}}(s_1)$ に対して，
$\langle s_1, \mathbf{a}\rangle \xrightarrow{(\sigma_1,\lambda_1,\mu_1)} \langle s_1', \mathbf{a}[\lambda_1 := 0]\rangle$ ならば，
$\langle s_2, \mathbf{b}\rangle \xrightarrow{(\sigma,\lambda_2,\mu_2)} \langle s_2', \mathbf{b}[\lambda_2 := 0]\rangle$
$\wedge\ \mu_1(\langle s_1, \mathbf{a}\rangle) \sqsubseteq_R \mu_2(\langle s_2, \mathbf{b}\rangle).$

但し，上の条件 *1.* と *2.* は *and* の関係である．また，$p_1 = \mu_1(\langle s_1, \mathbf{a}\rangle)$, $p_2 = \mu_2(\langle s_2, \mathbf{b}\rangle)$ とおいて，

$$p_1 \sqsubseteq_R p_2 \iff \exists w : states(\mathcal{A}) \times states(\mathcal{B}) \to [0,1] \text{ s.t.}$$

$$\begin{cases} i) & \forall q_1 \in states(\mathcal{A}),\ \sum_{q_2 \in states(\mathcal{B})} w(q_1,q_2) = p_1(q_1). \\ ii) & \forall q_2 \in states(\mathcal{B}),\ \sum_{q_1 \in states(\mathcal{A})} w(q_1,q_2) = p_2(q_2). \\ iii) & \text{全ての } (q_1,q_2) \in states(\mathcal{A}) \times states(\mathcal{B}) \text{ に対して，} \\ & \quad w(q_1,q_2) > 0 \text{ ならば } (q_1,q_2) \in R. \end{cases}$$

この関数 w を *weight function* と言う．

ここで，確率分布の値の設定の仕方に次の制約を与える．これにより確率分布は "離散確率分布" の意味を持つ．本論文で扱うのは離散確率分布のみである．このような回りくどい定義の仕方をする理由は，確率模倣関係を自然に定義するためである．その自然さを読者に理解してもらうために，$p_1 \sqsubseteq_R p_2$ **の定義を示した後で，**離散確率分布を導入するという順序を選んだ．

Definition 2.6 *(離散確率分布)*
$\forall \langle s, \mathbf{a}\rangle \in [\![\tau_s(\sigma,\lambda,\mu)]\!]$ に対し，

$$\mu(\langle s, \mathbf{a}\rangle)(\langle s', \mathbf{a}[\lambda := 0]\rangle) := \mathcal{P}_s(\sigma,\lambda,\mu)(s')$$

と設定する．ここで，$\mathcal{P}_s : prob(s) \to Dist(S)$ を離散確率分布と呼ぶことにする．これは遷移先のロケーション s' のみにより確率の値を割り付けるものである．

PTA 間の模倣関係を，確率時間強模倣関係を用いて次のように定義する．

Definition 2.7 *(PTA 間の模倣関係)*
初期状態対 $(\overline{q}_A, \overline{q}_B)$ が $R_{[\mathcal{A}\times\mathcal{B}]}$ に含まれているとき，$\mathcal{A} \preceq_{pt} \mathcal{B}$ と表し，*PTA* $\mathcal{A}$ は $\mathcal{B}$ に模倣されると言う．

注意 ; Definition 2.5 は，[12] における確率時間強模倣関係の定義とは次の点が異なる：「離散遷移後の状態対が R に含まれるという条件 ($(\langle s_1', \mathbf{a}[\lambda_1 := 0]\rangle, \langle s_2', \mathbf{b}[\lambda_2 := 0]\rangle) \in R$) を排除した点」．この変更により，2 つの PTA 間の構造の類似度がより小さくても $\mathcal{A} \preceq_{pt} \mathcal{B}$ が成立つように条件が緩和されている．

2.3 確率時間強模倣検証問題の定義

確率時間強模倣検証問題とは次のように定義される判定問題である [12].

Definition 2.8 *(確率時間強模倣検証問題)*
入力：*2* つの *PTA* $\mathcal{A}$, $\mathcal{B}$.
出力：$\mathcal{A} \preceq_{pt} \mathcal{B}$ ならば *yes* / そうでなければ *no*.

本論文ではこれ以降，確率時間強模倣検証アルゴリズム（確率時間強模倣検証問題を解く，停止性と正当性が保証された有限の手続き）を具体的に構成する．

3 確率時間強模倣検証アルゴリズム

本章では，確率時間強模倣検証アルゴリズムを与える．先ず，アルゴリズムの原理 (正当性) を説明し，次に，具体的構成を示す．そして最後にアルゴリズムの停止性を述べる．

3.1 アルゴリズムの原理

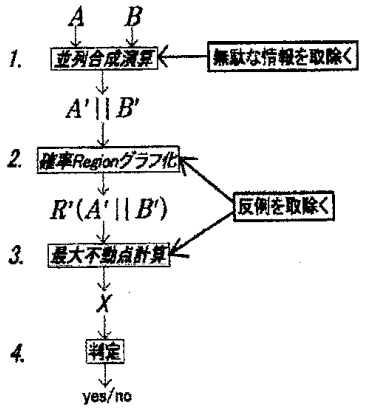

図 1；アルゴリズムの全体の流れとアイディアの概観

基本的考え方は，$R_{[A\times B]} \subset states(\mathcal{A}) \times states(\mathcal{B})$ より，$states(\mathcal{A}) \times states(\mathcal{B})$ の中から確率時間強模倣の条件 (時間模倣条件，確率模倣条件) を充たさない状態対 (反例) を全て取除いたものが $R_{[A\times B]}$ であり，反例を段階的に検出して全て取除こう，というものである．また，初期状態対 $(\overline{q}_A, \overline{q}_B)$ が $R_{[A\times B]}$ に含まれているか否かを判定することが目的であり，$R_{[A\times B]}$ そのものでなく，集合 $\mathcal{F} \stackrel{def}{=} \{X \mid X \subset R_{[A\times B]},\ (\overline{q}_A, \overline{q}_B) \in R_{[A\times B]} \iff (\overline{q}_A, \overline{q}_B) \in X\}$ の元を考えれば必要かつ充分である．但し，具体的な判定方法を構築するという立場から，$\mathcal{F}$ に含まれれば何を取ってきてもよい訳ではなく，$X \in \mathcal{F}$ は，次の 2 条件を充たす必要がある：

1. X は具体的に構成可能であること．
2. $(\overline{q}_A, \overline{q}_B)$ が X に含まれているか否かが判定可能であること．

Lemma 3.1 (計算対象の集合 X)

$X \stackrel{def}{=} h(Y) \cap R_{[A\times B]}$ と定めると，X は $X \in \mathcal{F}$ を充たす．但し，h は $h(((s_1,s_2),\mathbf{a}\circ\mathbf{b})) = (\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle)$ と定まる関数 $h : states(\mathcal{A} \parallel \mathcal{B}) \to states(\mathcal{A}) \times states(\mathcal{B})$．また，$Y$ は

$$Y \stackrel{def}{=} \{q \in Reach_{A\parallel B} \mid \exists r = \overline{q}_{A\parallel B} \stackrel{l_0,\mu_0}{\longrightarrow} \cdots \stackrel{l_{k-1},\mu_{k-1}}{\longrightarrow} q \in Path_{fin}^{A\parallel B} \text{ s.t. } r \text{ は } \mathcal{B} \text{ の非同期遷移を含まない}\}$$

と定め，$\overline{q}_{A\parallel B} \in Y$ である．

(証明)

$\mathcal{A} \preceq_{pt} \mathcal{B} \iff (\overline{q}_A, \overline{q}_B) \in R_{[A\times B]}$ かつ，$(\overline{q}_A, \overline{q}_B) \in h(Y)$ より，$\mathcal{A} \preceq_{pt} \mathcal{B} \Rightarrow (\overline{q}_A, \overline{q}_B) \in X$．

$\mathcal{A} \not\preceq_{pt} \mathcal{B} \iff (\overline{q}_A, \overline{q}_B) \notin R_{[A\times B]}$ かつ，$X \subset R_{[A\times B]}$ より，$\mathcal{A} \not\preceq_{pt} \mathcal{B} \Rightarrow (\overline{q}_A, \overline{q}_B) \notin X$．

(証明終)

以下，Lemma 3.1 で定義した X の具体的な構成可能性を示す．X は $h(Y)$ から確率時間強模倣の条件を充たさない状態対 (反例) を全て取除いた集合と解釈できることに注意しておく．

先ず，反例とは具体的には何か？ を明らかにしておく必要がある．$R = R_{[A\times B]}$ に対し，Definition 2.5 の論理的否定を取って，機械的に次の条件を得る．

次の 2 つの条件を充たす $(\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle) \in R_{[A\times B]}$ が存在する：

1. $\exists \delta \in \mathbb{R}_{>0}$ s.t.
   $\langle s_1,\mathbf{a}\rangle \stackrel{\delta}{\to} \langle s_1,\mathbf{a}+(\delta,\cdots,\delta)\rangle$
   $\wedge$ ($\langle s_2,\mathbf{b}\rangle \stackrel{\delta}{\not\to} \langle s_2,\mathbf{b}+(\delta,\cdots,\delta)\rangle$
   $\vee$ $(\langle s_1,\mathbf{a}+(\delta,\cdots,\delta)\rangle, \langle s_2,\mathbf{b}+(\delta,\cdots,\delta)\rangle) \notin R_{[A\times B]}$).
2. 次の条件を充たす $(\sigma_1,\lambda_1,\mu_1) \in prob_A(s_1)$ が存在する：
   $\langle s_1,\mathbf{a}\rangle \stackrel{(\sigma_1,\lambda_1,\mu_1)}{\longrightarrow} \langle s_1',\mathbf{a}'\rangle$
   $\wedge$ (全ての $(\sigma_2,\lambda_2,\mu_2) \in prob_B(s_2)$
   where $\sigma_1 = \sigma_2$ に対して {
   $\langle s_2,\mathbf{b}\rangle \stackrel{(\sigma_2,\lambda_2,\mu_2)}{\not\longrightarrow} \langle s_2',\mathbf{b}'\rangle \vee \mu_1(\langle s_1,\mathbf{a}\rangle) \not\sqsubseteq_{R_{[A\times B]}} \mu_2(\langle s_2,\mathbf{b}\rangle)$
   }).

但し，条件 1. と 2. は or の関係である．

今，状態対 (q_1,q_2) において，$a_{(q_1,q_2)}$ が真であることを (q_1,q_2) において時間非同期遷移が存在することに対応付け し，$b_{(q_1,q_2)}$ が真であることを (q_1,q_2) において離散非同期遷移が存在することに対応付けるならば，$a_{(q_1,q_2)} \vee b_{(q_1,q_2)}$ が真であることは (q_1,q_2) が反例であることを意味する．この反例を検出するには，

1) もし，$a_{(q_1,q_2)}$ が真であれば (q_1,q_2) は反例である．
2) もし，$b_{(q_1,q_2)}$ が真であれば (q_1,q_2) は反例である．

の 2 つの判定を用意すればよい．(q_1,q_2) が反例ならば少なくとも片方には引っ掛かる．

一方，先の条件 2. を "$\mathcal{A}$ の離散非同期が存在すること" の判定と "確率模倣しない離散非同期遷移が存在すること" の判定とに分割できる．つまり，

1) 先ず，状態対 $(\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle)$ において，$\langle s_1,\mathbf{a}\rangle$ が通れる辺 $(\sigma_1,\lambda_1,\mu_1) \in prob_A(s_1)$ がある場合，$\langle s_2,\mathbf{b}\rangle$ に対し，$\sigma_2 = \sigma_1$ なる同一ラベルの付いた通れる辺が存在するか，存在しないかのいずれかが成立つ．もし存在しなければ状態対は条件 2. を充たすため反例である (反例 *2*)．

2) 通れる辺が $n \geq 1$ 本存在したとして，その n 本の中に確率模倣する辺が存在するか，存在しないかのいずれかが成立つ．もし存在しなければ状態対は条件 2. を充たすため反例である (反例 *3*)．

最後に，条件 1. において，先ず，条件

$$\exists \delta \in \mathbb{R}_{>0} \text{ s.t. } \langle s_1,\mathbf{a}\rangle \stackrel{\delta}{\to} \langle s_1,\mathbf{a}+(\delta,\cdots,\delta)\rangle \ \wedge\ \langle s_2,\mathbf{b}\rangle \stackrel{\delta}{\not\to} \langle s_2,\mathbf{b}+(\delta,\cdots,\delta)\rangle$$

を充たす状態対 $(\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle)$ は明らかに $R_{[A\times B]}$ には含まれず反例である．そして，その状態対に 1 回の時間遷移で到達可能な全ての状態対もまた反例である，と判定する．このように判定すれば，条件 $(\langle s_1,\mathbf{a}+(\delta,\cdots,\delta)\rangle, \langle s_2,\mathbf{b}+(\delta,\cdots,\delta)\rangle) \notin R_{[A\times B]}$ は不要になる．

以上，反例の検出に用いる判定条件を形式的に表現すると次の 3 つの条件式が得られる．かつ，これらの条件式を用いた段階的な検出で反例として検出されなかった状態対は反例ではない．

Definition 3.1 (反例の検出パターン)

反例は次の *3* つのパターンに分けて検出可能である．

反例 *1*. $\mathcal{A}$ の時間非同期遷移が可能な状態対 $(\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle)$;

$$\exists \delta \in \mathbb{R}_{>0} \text{ s.t. } \langle s_1,\mathbf{a}\rangle \stackrel{\delta}{\to} \langle s_1,\mathbf{a}+(\delta,\cdots,\delta)\rangle \ \wedge\ \langle s_2,\mathbf{b}\rangle \stackrel{\delta}{\not\to} \langle s_2,\mathbf{b}+(\delta,\cdots,\delta)\rangle.$$

反例 *2*. $\mathcal{A}$ の離散非同期遷移が可能な状態対 $(\langle s_1,\mathbf{a}\rangle, \langle s_2,\mathbf{b}\rangle)$;

$$\exists (\sigma_1,\lambda_1,\mu_1) \in prob_A(s_1) \text{ s.t. } \langle s_1,\mathbf{a}\rangle \stackrel{(\sigma_1,\lambda_1,\mu_1)}{\longrightarrow} \langle s_1',\mathbf{a}[\lambda_1 := 0]\rangle$$
$$\wedge \text{ 全ての } (\sigma_2,\lambda_2,\mu_2) \in prob_B(s_2) \text{ where } \sigma_1 = \sigma_2 \text{ に対して } \{ \forall s_2' \in S_B, \langle s_2,\mathbf{b}\rangle \stackrel{(\sigma_2,\lambda_2,\mu_2)}{\not\longrightarrow} \langle s_2',\mathbf{b}[\lambda_2 := 0]\rangle\}.$$

反例 *3*. 確率模倣しない離散同期遷移が可能な状態対 $(\langle s_1,\mathbf{a}\rangle \langle s_2,\mathbf{b}\rangle)$;

$$\exists (\sigma_1,\lambda_1,\mu_1) \in prob_A(s_1) \text{ s.t. } \langle s_1,\mathbf{a}\rangle \stackrel{(\sigma_1,\lambda_1,\mu_1)}{\longrightarrow} \langle s_1',\mathbf{a}[\lambda_1 := 0]\rangle$$
$$\wedge \left(\langle s_2,\mathbf{b}\rangle \stackrel{(\sigma_2,\lambda_2,\mu_2)}{\longrightarrow} \langle s_2',\mathbf{b}'\rangle \text{ where } \sigma_1 = \sigma_2 \ \wedge\ \mu_1 \not\sqsubseteq_{R_{[A\times B]}} \mu_2 \right).$$

次に，反例における確率模倣の判定はどうやって行なうのか？ に解答する．その際，次に述べる Proposition 3.1 を基礎にする．

Proposition 3.1 *(確率模倣の判定法 (C. Baier [8]))*

$p_1 \sqsubseteq_R p_2 \iff$ ネットワーク $\mathcal{N}(p_1,p_2,R)$ の最大フローが 1．但し，これは確率オートマトン上で成立つ．また，ネットワーク $\mathcal{N}(p_1,p_2,R) = (V_U, E_U, \bot, \top, c_U)$ は，2 つの確率オートマトン $\mathcal{A}_i = (Q_i, \Sigma_i, \mathcal{E}_i)$ $(i = 0,1)$ に対し次のように構成される．

- $V_U \stackrel{def}{=} Q \cup \widehat{Q}$．ここで，$Q \stackrel{def}{=} Q_0 \cup Q_1$．また，$\widehat{Q} \stackrel{def}{=} \{\widehat{q} \mid q \in Q\}$ は，$Q \cap \widehat{Q} = \{\}$ を充たす，(Q とは異なる) 状態集合である．
- $E_U \stackrel{def}{=} \{(\bot, q) \mid q \in Q\} \cup \{(q_1, \widehat{q_2}) \mid (q_1,q_2) \in R\} \cup \{(\top, \widehat{q}) \mid \widehat{q} \in \widehat{Q}\}$.

- 容量関数 $c_U : E_U \to \mathbb{R}_{\geq 0}$ は以下のように定められる；
  - $\forall(\bot, q_1) \in E,\ c_U(\bot, q_1) \stackrel{def}{=} p_1(q_1)$.
  - $\forall(\widehat{q_2}, \top) \in E,\ c_U(\widehat{q_2}, \top) \stackrel{def}{=} p_2(q_2)$.
  - $\forall(q_1, \widehat{q_2}) \in E$ where $q_1 \neq \bot \wedge q_2 \neq \top$, $c_U(q_1, \widehat{q_2}) \stackrel{def}{=} 1$.

(証明略) [8] を参照のこと.

確率模倣判定は，確率 Region グラフ [7] 上で有限の方法で行なうことが出来る．その根拠となるのが次の Lemma 3.2 である．

Lemma 3.2 (確率模倣判定に用いる情報の局所性)
状態対 $(\langle s_1, \mathbf{a}\rangle, \langle s_2, \mathbf{b}\rangle)$ において，$\langle s_1, \mathbf{a}\rangle$ が通れる辺が存在しその辺を $(\sigma_1, \lambda_1, \mu_1)$ とし，$\langle s_2, \mathbf{b}\rangle$ が通れる辺が存在しその辺を $(\sigma_2, \lambda_2, \mu_2)$ とする．このとき，$p_1 = \mu_1(\langle s_1, \mathbf{a}\rangle)$, $p_2 = \mu_2(\langle s_2, \mathbf{b}\rangle)$ として，

$$p_1 \sqsubseteq_R p_2 \iff p_1 \sqsubseteq_{R \cap L} p_2$$

が成立つ．言い換えれば，確率模倣の判定には局所的な情報のみで必要かつ充分である．但し，$L \stackrel{def}{=} supp^{(\sigma_1,\lambda_1,\mu_1)}_{\langle s_1,\mathbf{a}\rangle} \times supp^{(\sigma_2,\lambda_2,\mu_2)}_{\langle s_2,\mathbf{b}\rangle}$．ここで，

$$supp^{(\sigma,\lambda,\mu)}_{\langle s,\mathbf{a}\rangle} \stackrel{def}{=} \left\{ \langle s', \mathbf{a}[\lambda := 0]\rangle \mid \mu(\langle s, \mathbf{a}\rangle)(\langle s', \mathbf{a}[\lambda := 0]\rangle) > 0 \right\}.$$

(証明)
PTA の動作を考えると，状態 $\langle s, \mathbf{a}\rangle$ と通る辺 (σ, λ, μ) が決定した場合，リセット後の各クロック変数値は固定され，ロケーションのみに自由度がある．従って，

$$\bigcup_{s' \in S} \mu(\langle s, \mathbf{a}\rangle)(\langle s', \mathbf{a}[\lambda := 0]\rangle) = 1 \quad (*)$$

が成立つ (逆に言えば，これが成立たない PTA はエラー動作を含んでいる．今，エラー動作を含む PTA は考えないことにする). 一方，[8] で与えられている Proposition 3.1 の証明は，ネットワークが "有限であること" に依存していないため，Proposition 3.1 は確率時間オートマトン上でも成立つと考えられる．

上の条件式 $(*)$ と Proposition 3.1 のネットワークの構成法を踏まえ，容量が 0 の辺は無視されるから，$p_1 = \mu_1(\langle s_1, \mathbf{a}\rangle)$, $p_2 = \mu_2(\langle s_2, \mathbf{b}\rangle)$ として，2 つのネットワーク $\mathcal{N}(p_1, p_2, R)$ と $\mathcal{N}(p_1, p_2, R \cap L)$ は等価なネットワークとなる．故に，各ネットワークの最大フローは等しくなり，$p_1 \sqsubseteq_R p_2 \iff p_1 \sqsubseteq_{R \cap L} p_2$.

(証明終)

次に，$h(Y)$ 上から反例を全て取除くことは出来るのか？という問いに解答する．$h(Y)$ は注意深く選んだ集合であり，反例をもれなく取除くための情報を全て含んでいる．このことを，反例を検出する方法を具体的に与えることで示す．以下の検出法のいずれも，必要充分条件として与えてあるため，反例の検出もれはない．

以下，PTA の並列合成演算 [12]，Clock Region [4], [6]，succ 遷移 [7]，確率 Region グラフ [7] はそれぞれ既知とする．

Lemma 3.3 (時間非同期の検出法)
次のように構成される集合 T を考える．

$$Stop \stackrel{def}{=} \{\, v \mid v \in Reach_{\mathcal{R}(\mathcal{A}' \| \mathcal{B}')},\ v = \langle (s_1, s_2), \mathbf{c}\rangle \notin V_{div} \wedge \nexists succ(v) \text{ at } inv_A(s_1) \wedge inv_B(s_2) \,\}.$$

$$T \stackrel{def}{=} \{\, v' \mid v' = \langle (s_1, s_2), \mathbf{c}\rangle \in Stop,\ \exists succ(v') \text{ at } (\, inv_A(s_1) \wedge \psi_\Omega \text{ where } [\![\psi_\Omega]\!] = \mathbb{R}^{|\mathcal{X}_B|}_{\geq 0} \,) \,\}$$

このとき，

$$T \neq \{\} \iff \exists\, q \in \langle (s_1, s_2), [\mathbf{c}]\rangle \in Stop \text{ s.t. } q \text{ からの } \mathcal{A} \text{ の時間非同期遷移が可能.}$$

が成立つ．但し，$[\mathbf{c}] \subset \mathbb{R}^{|\mathcal{X}|}_{\geq 0}$ は $\mathbf{c} \in \mathbb{R}^{|\mathcal{X}|}_{\geq 0}$ が属す *Clock Region*.

(証明)
$Stop \neq \{\}$ は常に成立つことに注意する．
$(\Rightarrow)$; $T \neq \{\}$ ならば，T の定義より，次の条件を充たす頂点 $v = \langle (s_1, s_2), [\![\psi_1 \wedge \psi_2]\!]\rangle \in Stop$ が存在する：

$$\exists q_1 \in \langle s_1, [\![\psi_1]\!]\rangle,\ \exists \delta \in \mathbb{R}_{>0} \text{ s.t. } q_1 \xrightarrow{\delta} q_1' \wedge q_1' \in succ(\langle s_1, [\![\psi_1]\!]\rangle).$$

$\langle s_1, [\![\psi_1]\!]\rangle \in Stop$ は時間発散的な頂点ではないため，$\langle s_1, [\![\psi_1]\!]\rangle \neq succ(\langle s_1, [\![\psi_1]\!]\rangle)$ が成立ち，$q_1' \in succ(\langle s_1, [\![\psi_1]\!]\rangle) \subseteq [\![inv_A(s_1) \wedge inv_B(s_2)]\!]$ であるため，上の遷移 $q_1 \xrightarrow{\delta} q_1'$ は時間非同期遷移であり，q_1 から時間非同期遷移が可能．
$(\Leftarrow)$; $q = \langle (s_1, s_2), \mathbf{a} \circ \mathbf{b}\rangle \in Stop$ を $\mathcal{A}$ の時間非同期遷移が可能な状態とすると，

$$\exists q',\ \exists \delta \in \mathbb{R}_{>0} \text{ s.t. } q \xrightarrow{\delta} q' \wedge [q] \neq [q'].$$

が成立つ (並列合成演算の定義より，$[\![inv_A(s_1) \wedge inv_B(s_2)]\!]$ 内には時間非同期遷移先は存在しないため，$q \in [\![inv_A(s_1) \wedge inv_B(s_2)]\!]$ と q' の属す Clock Region は異なると言える). $[q] \neq [q']$ だから，$q' = \langle (s_1', s_2'), \mathbf{a}' \circ \mathbf{b}'\rangle$ として，$succ(\langle s_1, [\mathbf{a}]\rangle) = \langle s_1', [\mathbf{a}']\rangle$ が成立ち，従って，$[q] \in T$ であり，$T \neq \{\}$.

(証明終)

Lemma 3.4 (離散非同期の検出法)
反例 *2.* は次の 2 つのパターンに分類される．
(1) $\mathcal{A}$ の非同期辺を通る遷移が可能な状態対．
(2) $\mathcal{A}$ 上では通れたが，$\mathcal{A} \| \mathcal{B}$ 上では通れなくなった同期辺．
そして，それぞれの場合の離散非同期の検出原理は以下である．
(1)' $\exists (\sigma, \lambda, \mu) \in prob_{\mathcal{A}' \| \mathcal{B}'}(s_1, s_2)$ s.t. $\rho \subset [\![\tau^{\mathcal{A}' \| \mathcal{B}'}_{(s_1,s_2)}(\sigma, \lambda, \mu)]\!] \wedge (\sigma, \lambda, \mu)$ は $\mathcal{A}$ の非同期辺 $\iff \forall q \in \langle (s_1, s_2), \rho\rangle$, q からの離散非同期遷移 (1) が可能.
(2)' $\exists (\sigma, \lambda_1 \uplus \lambda_2, \mu_1 \times \mu_2) \in prob_{\mathcal{A}' \| \mathcal{B}'}(s_1, s_2)$ s.t. $p_1 \subset [\![\tau^{\mathcal{A}}_{s_1'}(\sigma, \lambda_1, \mu_1)]\!] \wedge \rho_2 \not\subset [\![\tau^{\mathcal{B}}_{s_2'}(\sigma, \lambda_2, \mu_2)]\!] \wedge \langle \sigma, \lambda, \mu_1 \times \mu_2\rangle$ は $\mathcal{A}$ の同期辺 $\iff \forall q \in \langle (s_1, s_2), \rho_1 \cap \rho_2\rangle$, q からの離散非同期遷移 (2) が可能.

(証明略) 紙面の都合.

Lemma 3.5 (確率模倣しない同期遷移の検出法)
$\exists$ 同期辺 $(v, \sigma, \eta_1 \times \eta_2) \in E$ s.t. $\mathcal{N}(\eta_1, \eta_2, V')$ の最大フローが 1 でない $\iff \forall q \in v$, q からの確率非模倣遷移が可能.

(証明略) 紙面の都合．ネットワーク $\mathcal{N}(\eta_1, \eta_2, V')$ の構成法は後述のアルゴリズムを参照のこと．

以下，図 1 の流れに沿って，全体の計算の流れを箇条書きで示す．各番号は，図 1 の番号に対応付けてある．

1. 確率時間強模倣関係の時間模倣条件において，$\mathcal{A}$ の時間遷移と $\mathcal{B}$ の時間遷移の時間経過量 δ が同一であるため，時間同期が取れる状態集合を，並列合成演算 [5] で計算する．その際，$\mathcal{B}$ の非同期辺を取除く．取除いたものを $\mathcal{A}' \| \mathcal{B}'$ とおく．これは，Y を構成するための第 1 ステップである．
2. Y は確率 Region グラフ [7] $\mathcal{R}(\mathcal{A}' \| \mathcal{B}')$ と対応する．そして，$\mathcal{R}(\mathcal{A}' \| \mathcal{B}')$ の構成過程で時間非同期遷移を経由しないと到達できない状態 (反例 1) と離散非同期遷移を経由しないと到達できない状態 (反例 2) を Y から取除いたもの X_1 ($X \subset h(X_1) \subset h(Y)$) を構成する．この X_1 は $\mathcal{R}(\mathcal{A}' \| \mathcal{B}')$ の部分グラフ $\mathcal{R}'(\mathcal{A}' \| \mathcal{B}')$ に対応する．
3. X_1 上から確率模倣しない同期遷移を経由しないと到達できない状態 (反例 3) を取除いたもの X_2 ($X \subset h(X_2) \subset h(X_1)$) を最大不動点計算により構成する．すると，X_2 内には反例が存在せず，$X = h(X_2)$ が成立つ．
4. $X_2 \in \overline{q}_{\mathcal{A} \| \mathcal{B}} \iff X \in (\overline{q}_{\mathcal{A}}, \overline{q}_{\mathcal{B}})$ だから，X_2 内に $\overline{q}_{\mathcal{A} \| \mathcal{B}}$ が含まれているか否かを判定する．

3.2 確率時間強模倣検証アルゴリズムの構成

以下，アルゴリズムの全体の構成を先ず示し，次に，`detectAsynTransitions( )` と `detectNotProbSim( )` のみアルゴリズムを示す．残りは簡単であるため割愛する．

全体の構成

Algorithm **main**

0.0 **Input** : 2 つの PTA $\mathcal{A}, \mathcal{B}$.
1.0 **Compute** : 並列合成演算を行い，$\mathcal{B}$ の非同期辺を取除く．
1.1 $\mathcal{A} \parallel \mathcal{B} \leftarrow$ `parallelComposition`$(\mathcal{A}, \mathcal{B})$;
1.2 $\mathcal{A}' \parallel \mathcal{B}' \leftarrow$ `eraseAsynEdgesOfB`$(\mathcal{A} \parallel \mathcal{B})$;
2.0 **Compute** : 確率 Region グラフを構成し，その構成過程で反例 1 と反例 2 を取除く．
2.1 $\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}') \leftarrow$ `detectAsynTransitions`$(\mathcal{A}' \parallel \mathcal{B}')$;
3.0 **Compute** : $\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}')$ 上で到達可能集合を計算．
3.1 $V' \leftarrow$ `getReachabilitySet`$(\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}'))$;
4.0 **Compute** : 最大不動点計算により，V' から反例 3 を取除く
4.1 $V'' \leftarrow$ `detectNotProbSim`$(V', \mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}'))$;
5.0 **Output** : $\mathcal{A} \preceq_{pt} \mathcal{B}$ か否か．
5.1 **if** ($[\overline{q}_{\mathcal{A}\parallel\mathcal{B}}] \in V''$) **then** { **return** *yes*; }
5.2 **else** { **return** *no*; }

非同期遷移の検出アルゴリズム

Algorithm **detectAsynTransitions**

0.00 **Input** : PTA $\mathcal{A}' \parallel \mathcal{B}'$.
1.00 **Initialize** : 辺集合と頂点集合の初期化．
1.01 $E \leftarrow \{\}$;
1.02 $V_{div} \leftarrow \{\}$; $V \leftarrow \{v_0\}$; // $v_0 = \langle(\overline{s}_{\mathcal{A}}, \overline{s}_{\mathcal{B}}), [\mathbf{0}]\rangle$.
1.03 *enqueue*(v_0);
2.00 **Compute** : 確率 Region グラフを構成しつつ，反例 1 と反例 2 を取除く．
2.01 **while** (*queue* が空でない) {
2.02 $v \leftarrow$ *dequeue*(); // let $v = \langle(s_1, s_2), \rho\rangle$.
2.03 **if** (v からの *succ* 遷移は既に計算済み)
2.04 **then** { **goto** 2.*Z*; }
2.05 // (1) *succ* 遷移の構成：
2.06 $\tau_{succ} \leftarrow (v^{(i)})_{i\geq 1}$;
// $v^{(i)} = succ^i(v)$ **at** $inv(s_1, s_2)$.
2.07 **if** ($v^{(|\tau_{succ}|)}$が時間発散的なノードでない)
then {
2.08 // ■反例 1(時間非同期遷移) の検出
2.09 T を計算； // cf. Lemma 3.3
2.10 **if** ($T \neq \{\}$) **then** {
2.11 **if** ($v == v_0$) **then** { **exit**; }
2.12 **else** {
2.13 `rewriteEdge`(v);
2.14 **goto** 2.*Z*;
2.15 }
2.16 }
2.17 }
2.18 **else** { $V_{div} \leftarrow V_{div} \cup \{v^{(|\tau_{succ}|)}\}$; }
2.19 **for** $i = 1$ **to** $|\tau_{succ}|$ {
2.20 **if** ($v^{(i)} \notin V$) **then** { *enqueue*$(v^{(i)})$; }
2.21 $V \leftarrow V \cup \{v^{(i)}\}$;
2.22 $E \leftarrow E \cup \left\{\left(v^{(i-1)}, succ, \eta_{succ}^{\langle s, v^{(i)}\rangle}\right)\right\}$;
// $v^{(0)} = v$.
2.23 }
2.Z // (2) 離散遷移の構成：
2.25 **forall** $(\sigma, \lambda, \mu_1 \times \mu_2) \in prob_{\mathcal{A}'\parallel\mathcal{B}'}(s_1, s_2)$ {
2.26 **if** $\left(\rho \subset [\![\tau^{\mathcal{A}'\parallel\mathcal{B}'}_{(s_1,s_2)}(\sigma, \lambda, \mu_1 \times \mu_2)]\!]\right)$ {
2.27 // ■反例 2($\mathcal{A}$ の非同期辺) の検出
2.28 **if** ($ek((s_1, s_2), \sigma) == asyn_{\mathcal{A}}$)
2.29 **then**{
2.30 **if** ($v == v_0$) **then** { **exit**; }
2.31 **else** {
2.32 `rewriteEdge`(v);
2.33 **goto** 2.01;
2.34 }
2.35 }
2.36 // この時点で $ek((s_1, s_2), \sigma) == syn$.
2.37 $supp(\mu_1 \times \mu_2) \leftarrow \{(s_1', s_2') \in S_{\mathcal{A}} \times S_{\mathcal{B}} \mid \mathcal{P}^{\mathcal{A}}_{s_1}(\sigma, \lambda_1, \mu_1)(s_1') \times \mathcal{P}^{\mathcal{B}}_{s_2}(\sigma, \lambda_2, \mu_2)(s_2') > 0\}$;
// $\lambda = \lambda_1 \uplus \lambda_2$.
2.38 **forall** $(s_1', s_2') \in supp(\mu_1 \times \mu_2)$ {
2.39 $v' \leftarrow \langle(s_1', s_2'), \rho[\lambda := 0]\rangle$;
2.40 **if** ($v' \notin V$)
then { *enqueue*(v'); }
2.41 $V \leftarrow V \cup \{v'\}$;
2.42 }
2.43 $E \leftarrow E \cup \left\{\left(v, \sigma, \eta^{\langle s, v\rangle}_{\langle\sigma, \lambda, \mu_1\times\mu_2\rangle}\right)\right\}$;
2.44 }
2.45 **else if** ($\rho_1 \subset [\![\tau^{\mathcal{A}}_{s_1}(\sigma, \lambda_1, \mu_1)]\!]$) {
// $\rho = \rho_1 \cap \rho_2$.
2.46 // ■反例 2($\mathcal{A}$ の同期辺) の検出
2.47 **if** ($ek((s_1, s_2), \sigma) == syn$) **then** {
2.48 **forall** $(\sigma', \lambda_2', \mu_2') \in prob_{\mathcal{B}}(s_2)$
where $\sigma' == \sigma \neq succ$ {
2.49 **if** ($\rho_2 \not\subset [\![\tau^{\mathcal{B}}_{s_2}(\sigma', \lambda_2', \mu_2')]\!]$)
then {
2.50 `rewriteEdge`(v);
2.51 **goto** 2.01;
2.52 }
2.53 }
2.54 }
2.55 }
2.56 } // end forall
2.57 } // end while
3.00 **Output** : 部分 Region グラフ $\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}')$.

ここで，幾つかの注意を箇条書きで述べる．

- `rewriteEdge`(v) は，$v = \langle(s_1, s_2), \rho\rangle$ として，

  $$(v_{prev}, \sigma, \eta) \in E \ \textit{where}\ \eta(v) > 0$$

  を充たす辺を

  $$(v_{prev}, \sigma, \eta') \ \textit{where}\ \eta'(\langle(s_1, s_2), \textit{empty}\rangle) \leftarrow \eta(v), \eta'(v) \leftarrow 0$$

  に書き換える処理であり，反例を充たす状態を取除くことに相当する．ここで，後の確率模倣判定を行うため，(s_1, s_2) の情報は保持してある．
- $succ^i(v)$ **at** ζ は，v から i 回の *succ* 遷移でζ内で到達可能な *Clock Region* である．
- $ek((s_1, s_2), \sigma) \in \{asyn_{\mathcal{A}}, asyn_{\mathcal{B}}, syn\}$ はロケーション (s_1, s_2) におけるラベル σ が付いた辺の種類 (edge kind) を意味する．ここで，$\mathcal{A}' \parallel \mathcal{B}'$ 上には $\mathcal{B}$ の非同期辺が存在しないため，$ek((s_1, s_2), \sigma) == asyn_{\mathcal{B}}$ となることはない．

確率模倣しない同期遷移の検出アルゴリズム

Algorithm detectNotProbSim
0.00 **Input** : 頂点集合 V', 部分 Region グラフ $\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}')$.
// let $\mathcal{R}'(\mathcal{A}' \parallel \mathcal{B}') = (V, E)$.
1.00 **Compute** : 最大不動点計算により反例 3 を取除く.
1.01 **do** {
1.02 $V'' \leftarrow V'$;
1.03 **forall** $v \in V'$ {
1.04 **forall** $(v, \sigma, \eta_1 \times \eta_2) \in E$ *where* $\sigma \neq succ$ {
1.05 // ネットワークの構成.
$(V_U, E_U, \bot, \top, c_U) \leftarrow$ **getNet**(η_1, η_2, V');
1.06 最大フロー f_{max} を求める;
1.07 **if** ($\mathcal{F}(f_{max})$!= 1) **then** {
1.08 $V' \leftarrow V' \backslash \{v\}$;
1.09 **goto** 1.3; // try next vertex
1.10 }
1.11 }
1.12 }
1.13 } **while** (V'' != V') ; // $V'' == V'$ なら終了.
2.00 **Output** : 頂点集合 V''.

ここで, **getNet**(η_1, η_2, V') において, 次のようにネットワーク $(V_U, E_U, \bot, \top, c_U)$ における頂点集合 V_U, 辺集合 E_U, 容量関数 $c_U : E_U \to \mathbb{R}_{\geq 0}$ を構成する.

V_U の構成 :
$V_U \leftarrow V_1 \cup V_2$; *where*
$V_1 = \{v_1 \mid \eta_1(v_1) > 0\}$, $V_2 = \{v_2 \mid \eta_2(v_2) > 0\}$.

E_U の構成 :
$E_U \leftarrow E_\bot \cup E_1 \cup E_\top$; *where*
$E_1 = \{(v_1, v_2) \in V_1 \times V_2 \mid v_1 \circ v_2 \in V'\}$.
$E_\bot = \{(\bot, v_1) \mid v_1 \in V_1\}$, $E_\top = \{(v_2, \top) \mid v_2 \in V_2\}$.

c_U の構成 :
$\forall (v_1, v_2) \in E_1,\ c_U(v_1, v_2) \leftarrow (\eta_1 \times \eta_2)(v_1 \circ v_2)$;
$\forall (\bot, v_1) \in E_\bot,\ c_U(\bot, v_1) \leftarrow \eta_1(v_1)$;
$\forall (v_2, \top) \in E_\top,\ c_U(v_2, \top) \leftarrow \eta_2(v_2)$;

アルゴリズムの停止性

入力と出力は有限時間で停止する. また, 並列合成演算も有限時間で停止する. そして, Clock Region の数 $|\mathcal{V}_{\mathcal{X}_{\mathcal{A}\parallel\mathcal{B}}}|$ は有限で有限回の dequeue() により queue が空になるため, 確率 Region グラフの任意の部分グラフの構成は有限回の操作で可能である, かつ, 反例の検出は有限時間で可能より, 処理 **detectAsynTransitions**$(\mathcal{A}' \parallel \mathcal{B}')$ は有限時間で停止する. そして, 有限グラフとなる確率 Region グラフ上での到達可能集合の計算も有限時間で停止する. 最後に, **detectNotProbSim**() は, 最大不動点計算 ($f : 2^{S_{\mathcal{A}\parallel\mathcal{B}} \times \mathcal{V}_{\mathcal{X}_{\mathcal{A}\parallel\mathcal{B}}}} \to 2^{S_{\mathcal{A}\parallel\mathcal{B}} \times \mathcal{V}_{\mathcal{X}_{\mathcal{A}\parallel\mathcal{B}}}}$ なる $\bigcap$-連続関数 (**where** $f(V) \stackrel{def}{=} V$ から確率非模倣の頂点を取除く) と初期集合 $V' \subset S_{\mathcal{A}\parallel\mathcal{B}} \times \mathcal{V}_{\mathcal{X}_{\mathcal{A}\parallel\mathcal{B}}}$ に対し, 最大不動点 $fix(V') = \bigcap_{i=0}^{\infty} f^i(V')$ **where** $f(fix(V')) = \bigcap_{i=0}^{\infty} f(f^i(V')) = \bigcap_{i=0}^{\infty} f^{i+1}(V') = fix(V')$) として形式化でき, f の定義域・値域共に有限集合より, $\exists N \in \mathbb{N}$ s.t. $\bigcap_{i=0}^{N} f^i(V') = fix(V') \Rightarrow f^N(V') = f^{N+1}(V')$ が成立ち, 有限のステップで停止条件を充たし停止する. 以上より, 全ての手続きは有限時間で停止する.

4 おわりに

本論文では, *S. Yamane* により考案された "確率時間オートマトン上の確率時間強模倣関係に初期状態対が含まれているか否か" という問題に有限の手続きで解答する, 反例を検出し取除くという考え方に基くアルゴリズムを提案した. 反例の検出を有限の手続きで行える理由は主に "Clock Region による時間構造の有限化" と "確率 Region グラフによる到達可能性解析の有限化" に基いている.

今後の課題としては, 以下の項目が挙げられる.

- 実装によるアルゴリズムの性能評価.
- アルゴリズムの効率化・抽象化.
- 確率時間弱模倣関係の場合の検証アルゴリズムの考案.
- 実用化研究.

参考文献

[1] Bengt Jonsson, and Kim Guldstrand Larsen ; Specification and Refinement of Probabilistic Processes ; *In Proceedings of the 6th LICS*, July 1991.

[2] Rajeev Alur and David L. Dill ; A Theory of Timed Automata ; *Theoretical Computer Science*, Vol. 126, pages 183-235, 1994.

[3] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine ; Symbolic Model Checking for Real-time Systems ; *Information and Computation*, 111(2):193-244, 1994.

[4] Roberto Segala and Nancy Lynch ; Probabilistic Simulations for Probabilistic Processes ; *In CONCUR'94: Concurrency Theory. 5th Int. Conf., volume 836 of LNCS*, pages 481-496. Springer-Verlag, 1994.

[5] Serdar Tasiran, Rajeev Alur, Robert P. Kurshan, and Robert K. Brayton ; Verifying Abstractions of Timed Systems ; *LNCS 663*, pages 546-562, 1996.

[6] Sergio Yovine ; Model Checking Timed Automata ; In G. Rozenberg and F.Vaandrager, editors, *Embedded Systems*, volume 1494 of *LNCS*, pages 114-152. Springer, 1998.

[7] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston ; Automatic Verification of Real-Time Systems with Discrete Probability Distributions ; *Proc. of the 5th AMAST Workshop on Formal Methods for Real-Time and Probabilistic Systems, LNCS 1601*, 1999.

[8] Christel Baier, Bettina Engelen, and Mila Majster-Cederbaum ; Deciding Bisimilarity and Similarity for Probabilistic Processes ; *Journal of Computer and System Sciences 60*, pages 187-231, 1999.

[9] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled ; Model Checking ; The MIT Press, 1999.

[10] Robin Milner ; communicating and mobile systems : the π-calculus ; CAMBRIDGE UNIVERSITY Press, 1999.

[11] M. I. A. Stoelinga ; An Introduction to Probabilistic Automata ; *EATCS Bulletin*, No. 78, 2002.

[12] Satoshi Yamane ; Probabilistic Timed Simulation Verification and its Application to Stepwise Refinement of Real-Time Systems ; *Advances in Computing Science - ASIAN 2003 Programming Languages and Distributed Computation, 8th Asian Computing Science Conference*, pages 276-290, Mumbai, India, December 10-14, 2003.

[13] Ruggero Lanotte, Andrea Maggiolo-Schettini, and Angelo Troina ; Weak Bisimulation for Probabilistic Timed Automata and Applications to Security ; *Proc. 1st Int. Conference on Software Engineering and Formal Methods (SEFM'03)*, IEEE Computer Society Press, 34-43, September 2003.