Kleisli Simulation for Real-Weighted Automata and its Algorithm

卜部 夏木（蓮尾研究室）

Overview

- Simulation
- Kleisli Simulation
- Contribution
  1. Implementation
  2. Increasing the Chance of Simulation
- Experimental Results and Comparison

Overview

- Simulation
- Kleisli Simulation
- Contribution
  1. Implementation
  2. Increasing the Chance of Simulation
- Experimental Results and Comparison

Motivation

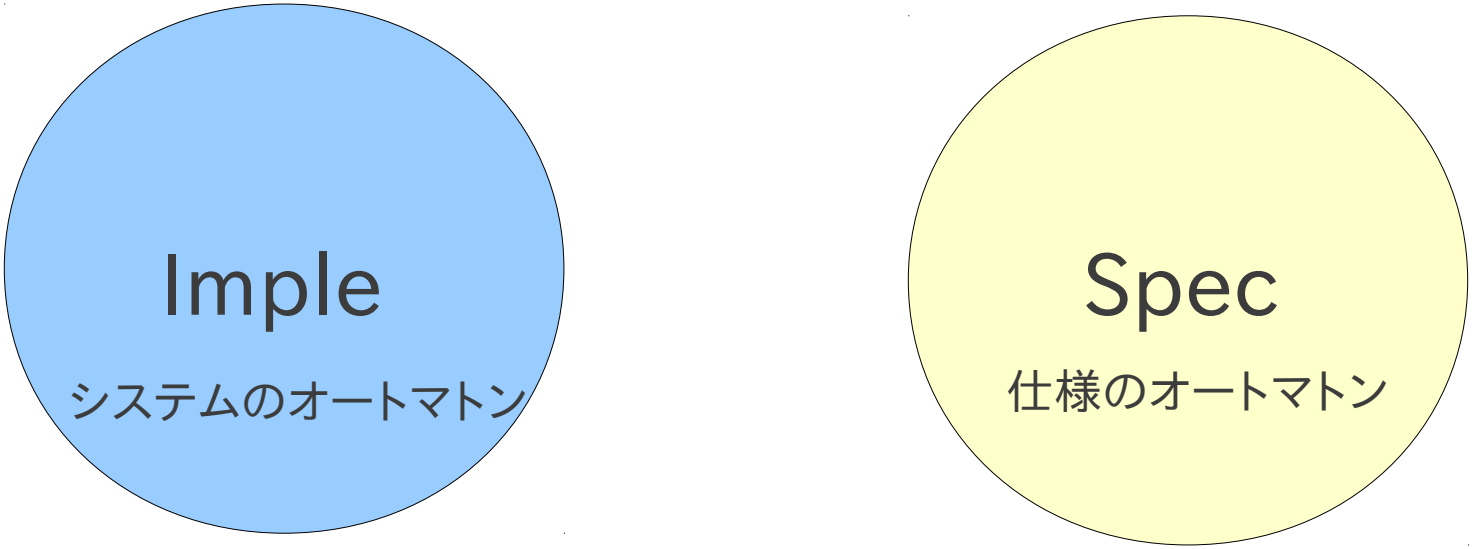
- 並列システムの形式検証

- 定量的な性質

  例：確率・空間・時間…

並列システムの形式検証

ここでの方法・・・

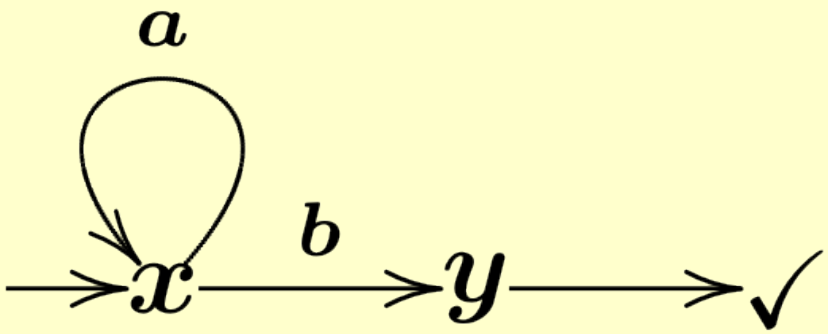


- システムの「ふるまい」が仕様の「ふるまい」に「含まれる」かどうか調べる

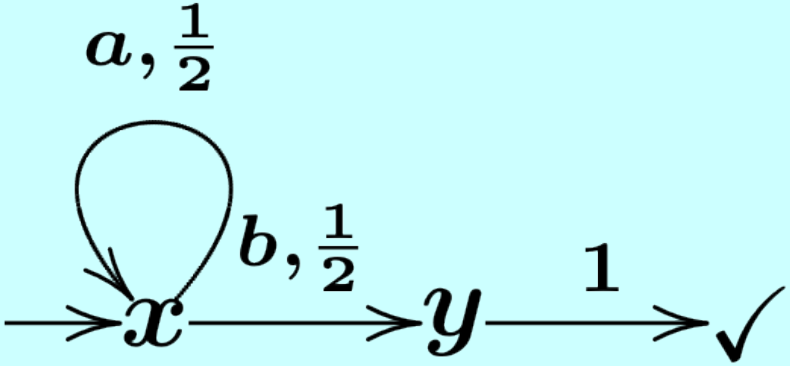
Trace Semantics

Trace semantics = 線形時間の全体的なふるまい

- 非決定性オートマトン



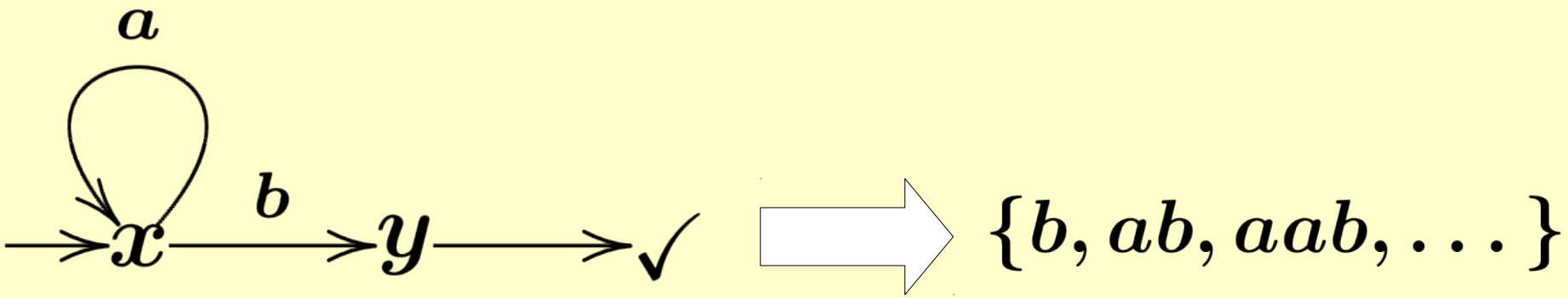
- 確率的オートマトン



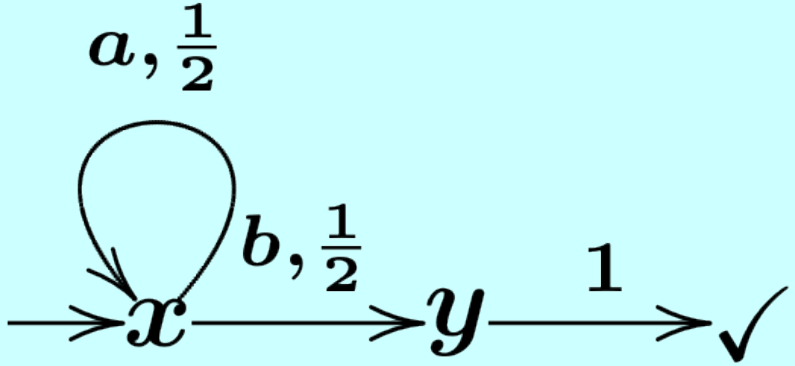
Trace Semantics

Trace semantics = 線形時間の全体的なふるまい

- 非決定性オートマトン



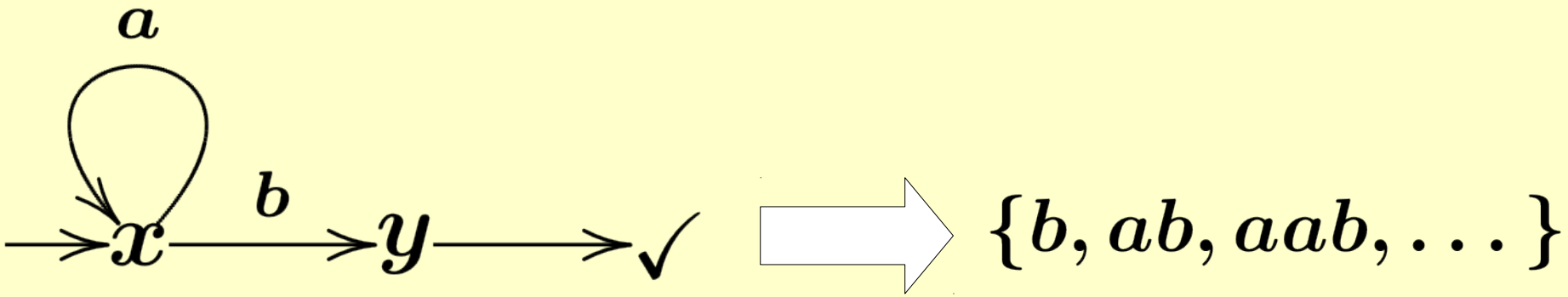
- 確率的オートマトン



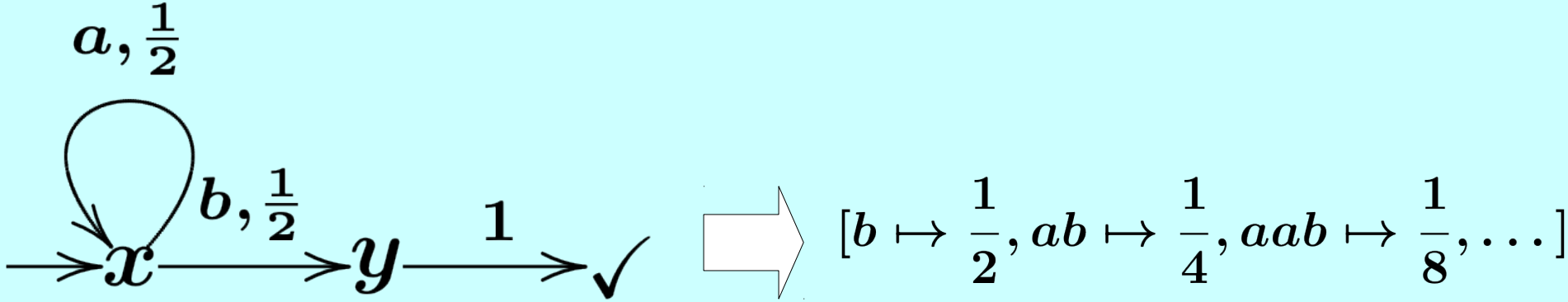
Trace Semantics

Trace semantics = 線形時間の全体的なふるまい

- 非決定性オートマトン



- 確率的オートマトン



Trace Inclusion

- Trace inclusion

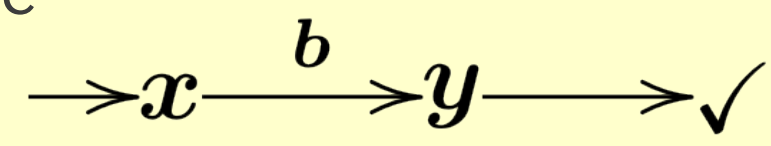
  = trace semantics 間の包含関係

Trace Inclusion

- Trace inclusion

  = trace semantics 間の包含関係
- 非決定性オートマトン → 集合の包含関係

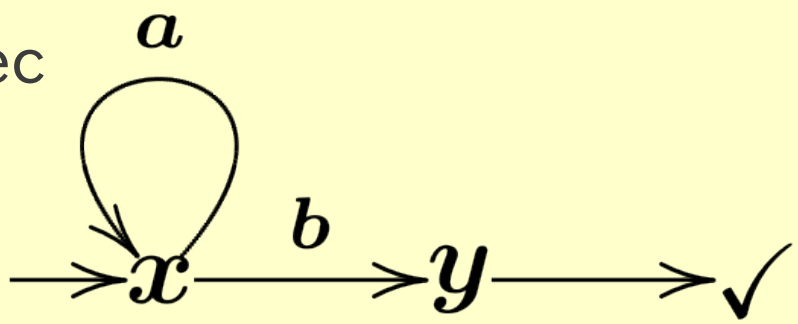
Imple



$\{b\}$

$\cap$

Spec



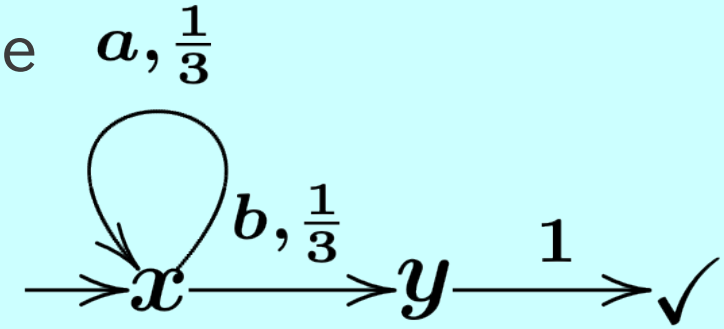
$\{b, ab, aab, \ldots\}$

Trace Inclusion

- Trace inclusion

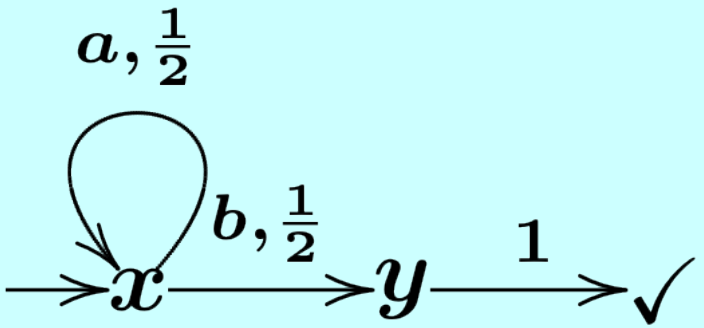
  = trace semantics 間の包含関係
- 確率的オートマトン → 文字列毎の確率

Imple



$$[b \mapsto \frac{1}{3}, ab \mapsto \frac{1}{9}, aab \mapsto \frac{1}{27}, \ldots]$$

Spec



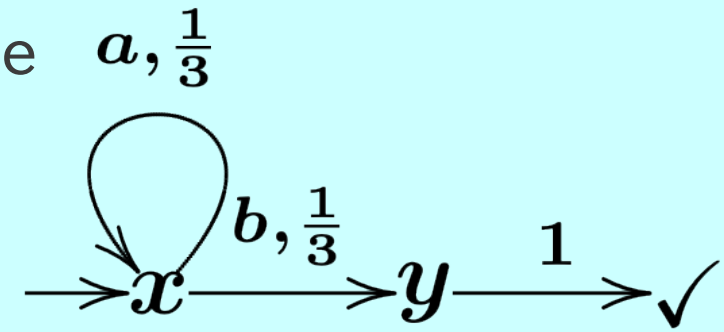
$$[b \mapsto \frac{1}{2}, ab \mapsto \frac{1}{4}, aab \mapsto \frac{1}{8}, \ldots]$$

Trace Inclusion

- Trace inclusion

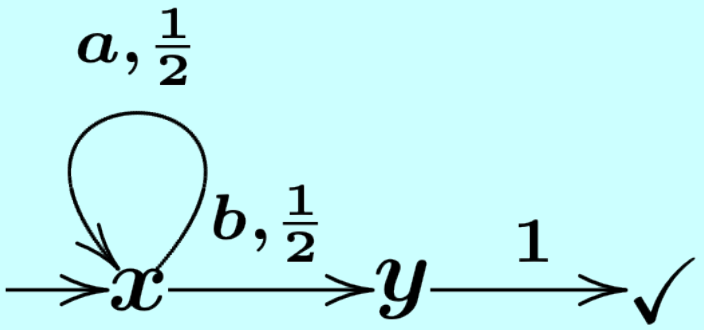
  = trace semantics 間の包含関係
- 確率的オートマトン → 文字列毎の確率

Imple



$$[b \mapsto \frac{1}{3}, ab \mapsto \frac{1}{9}, aab \mapsto \frac{1}{27}, \ldots]$$

Spec

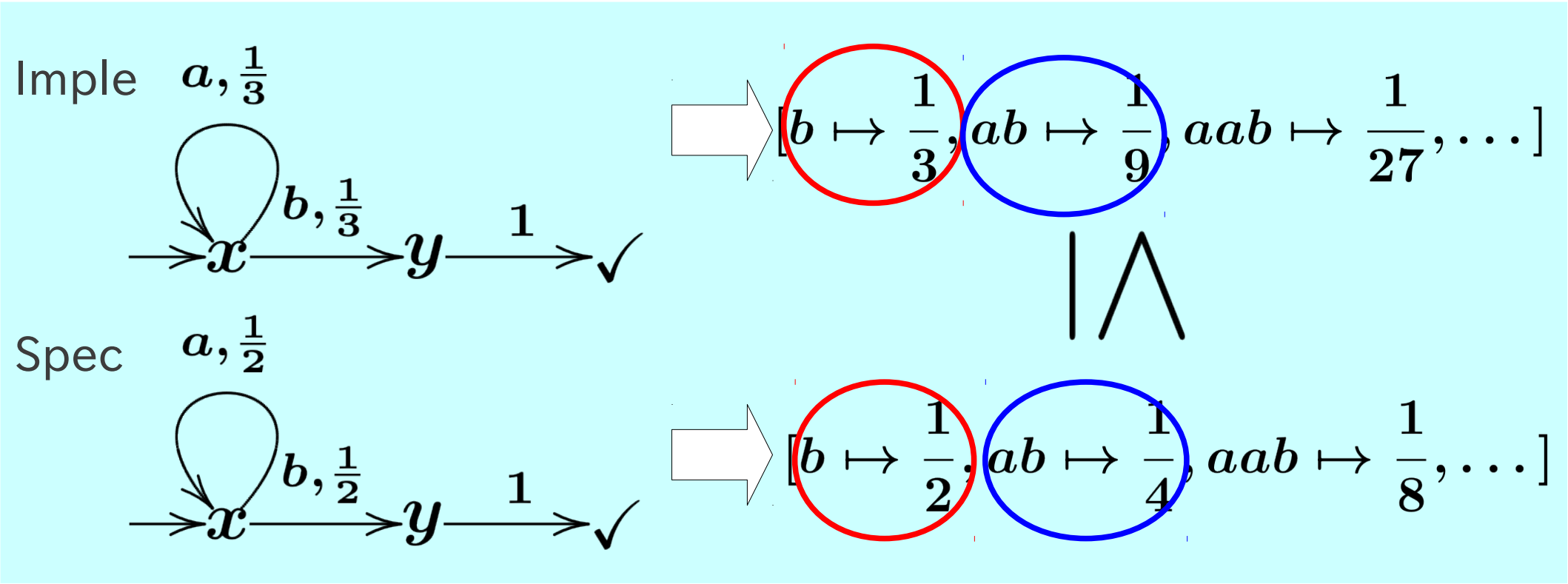


$$[b \mapsto \frac{1}{2}, ab \mapsto \frac{1}{4}, aab \mapsto \frac{1}{8}, \ldots]$$

Trace Inclusion

- Trace inclusion

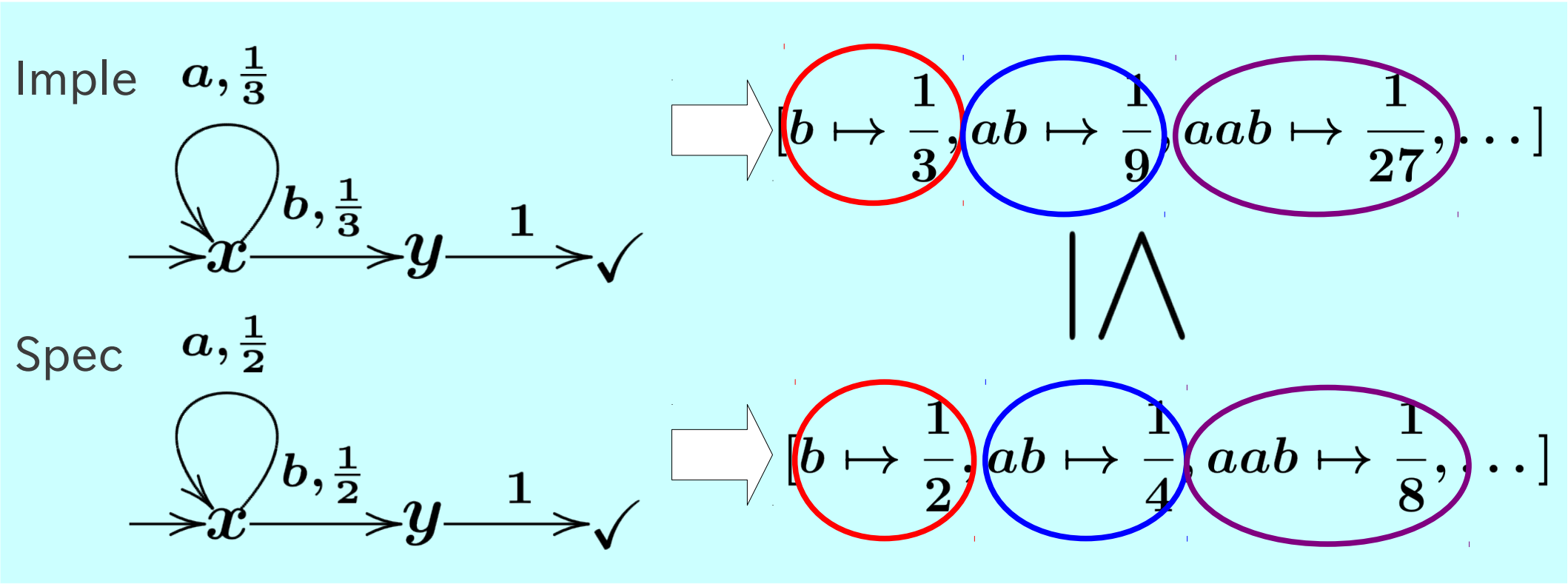
  = trace semantics 間の包含関係
- 確率的オートマトン → 文字列毎の確率



Trace Inclusion

- Trace inclusion

  = trace semantics 間の包含関係
- 確率的オートマトン → 文字列毎の確率



目標

- Trace inclusion を調べたい

- Trace semantics は普通は無限

$$\{b, ab, aab, \dots\} \quad [b \mapsto \frac{1}{2}, ab \mapsto \frac{1}{4}, aab \mapsto \frac{1}{8}, \dots]$$

目標

- Trace inclusion を調べたい
- Trace semantics は普通は無限

$$\{b, ab, aab, \dots\} \quad [b \mapsto \frac{1}{2}, ab \mapsto \frac{1}{4}, aab \mapsto \frac{1}{8}, \dots]$$

⇨ 自動で調べることは難しい

Simulation = Stepwise Trace Inclusion

- 全体の trace inclusion を調べる代わりに
  stepwise な trace inclusion を調べる

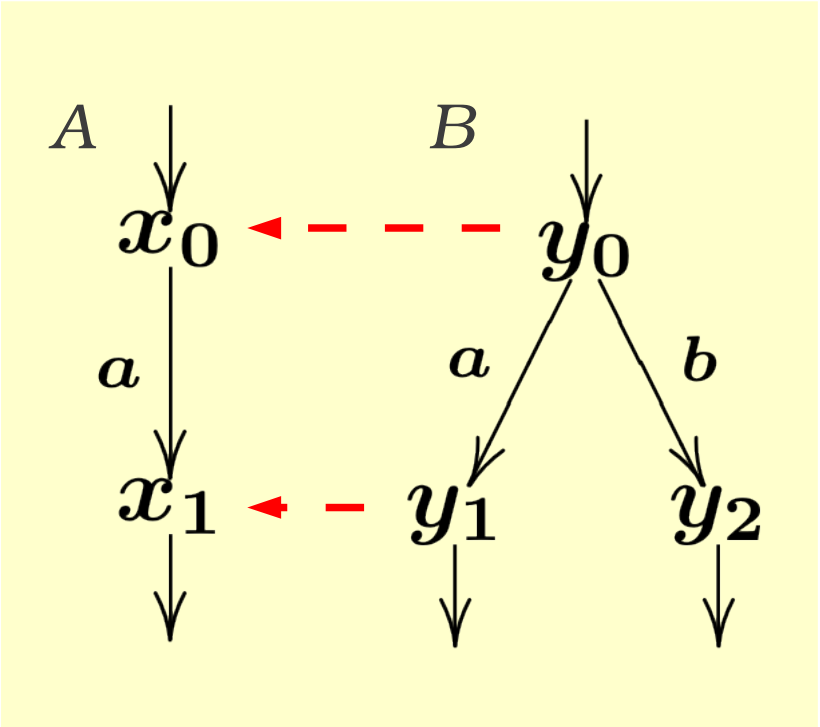
Simulation = Stepwise Trace Inclusion

- 全体の trace inclusion を調べる代わりに
  stepwise な trace inclusion を調べる
  ⇨ Simulation

Simulation = Stepwise Trace Inclusion

- 全体の trace inclusion を調べる代わりに
  stepwise な trace inclusion を調べる

  ⇨ Simulation

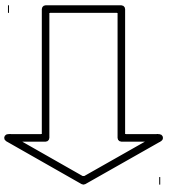


$$R \subseteq A \times B \quad \text{s.t.}$$
$$xRy,\ x \xrightarrow{a} x',\ y \xrightarrow{a} y' \ \Rightarrow\ x'Ry'$$

⇨ AからBへの simulationが存在

Simulation

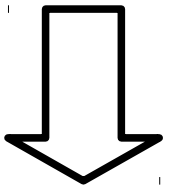
A から B への simulation が存在



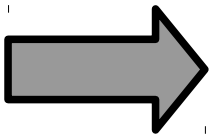
A の trace は B の trace に含まれる (trace inclusion)

Simulation

A から B への simulation が存在

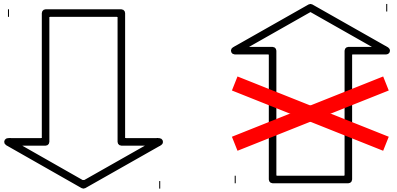


A の trace は B の trace に含まれる (trace inclusion)

 健全性

Simulation

A から B への simulation が存在



A の trace は B の trace に含まれる (trace inclusion)

⇒ 健全性

⇒ 完全性は必ずしも成り立たない

Overview

- Simulation
- Kleisli Simulation
- Contribution
  1. Implementation
  2. Increasing the Chance of Simulation
- Experimental Results and Comparison

Coalgebra

- State-based な遷移系の圏論的な理論
  - 例えば並列システム
- Jacobs, Rutten, Reichel in 90's

Kleisli 圏

- 圏 C とモナド T の Kleisli 圏 $Kl(T)$

  - 対象：Obj (C)
  - 射　：$f : X \nrightarrow Y$ in $Kl(T)$ is

    $f : X \to TY$ in C

Kleisli 圏

- Kleisli 圏で様々なシステムを表現できる
- モナド T ごとに様々なシステム

例:

Powerset monad

$T = P$ $PX = \{A \subseteq X\}$

→ 非決定性オートマトン

Subdistribution monad

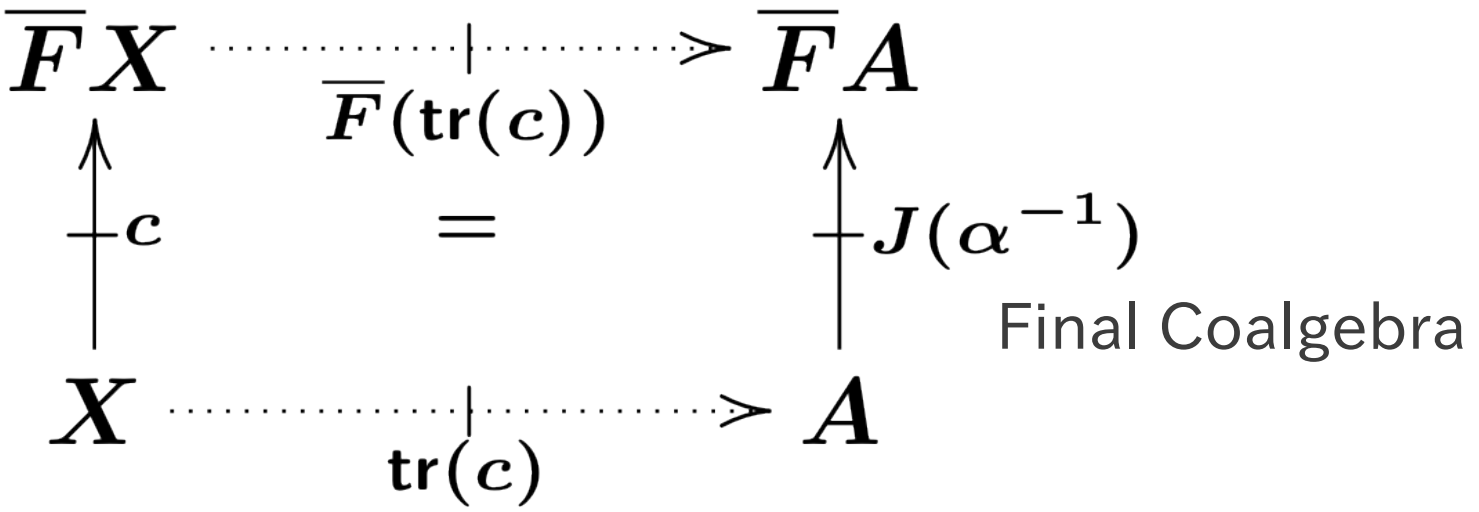
$T = D$ $DX = \{f : X \to [0,1] \mid \sum_x f(x) \leq 1\}$

→ 確率的オートマトン

Trace Semantics via Coinduction

(Hasuo, Jacobs, and Sokolova, 2006)

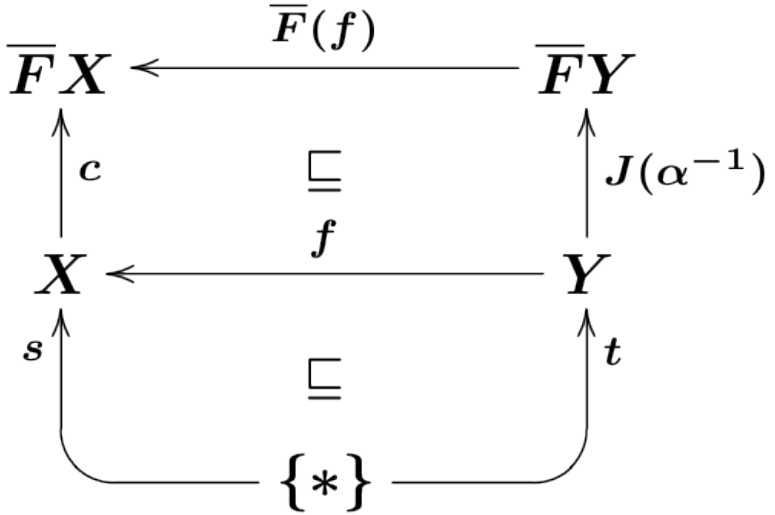
- Kleisli 圏を用いた
  trace semantics の一般的な定義
- 様々な遷移系に trace semantics を定義できる



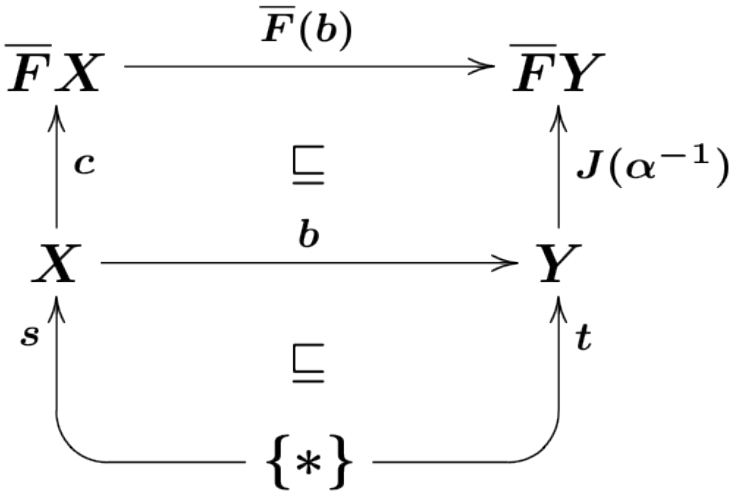
Kleisli Simulation (Hasuo, 2006)

- Kleisli 圏を用いた
  simulation の一般的な定義
- 様々な遷移系に simulation (Kleisli simulation)を定義できる

- Forward



Backward

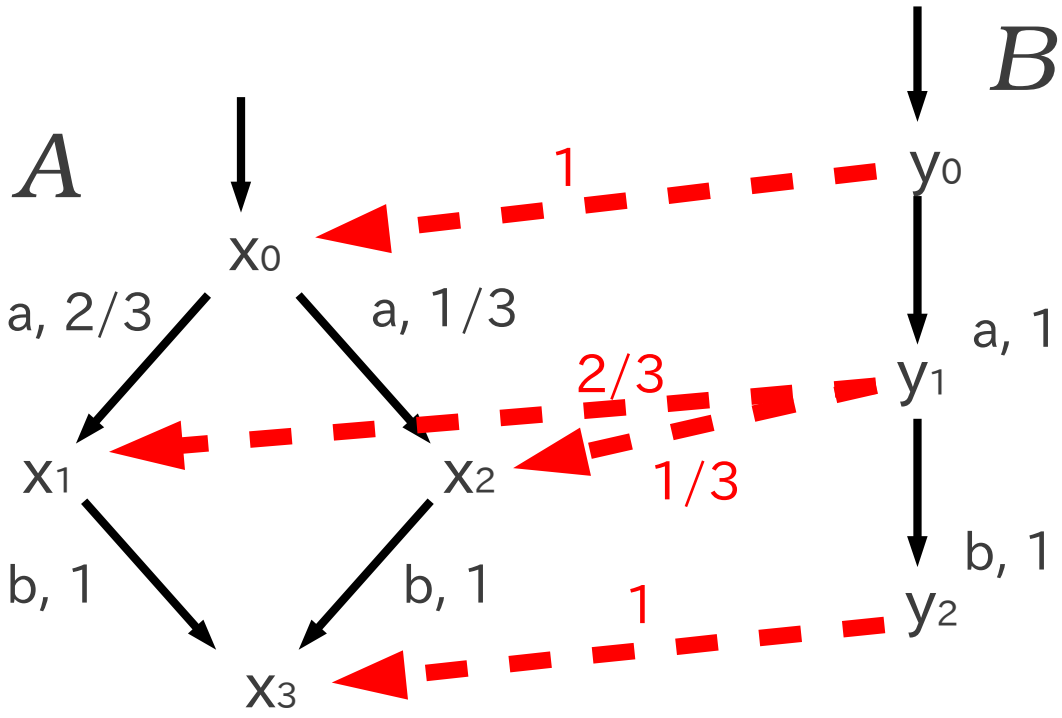


確率的オートマトンの Kleisli Simulation

定義:

A からB への forward simulation とは $f : B \to DA$ s.t.

$$\forall x \in A.\, \forall y \in B.\, \forall \sigma \in \Sigma.\ \mathbf{Pr}\left[\begin{array}{c} y \\ \sigma \downarrow \\ x \end{array}\right] \leq \mathbf{Pr}\left[\begin{array}{c} y \\ \sigma \downarrow \\ x \end{array}\right]$$



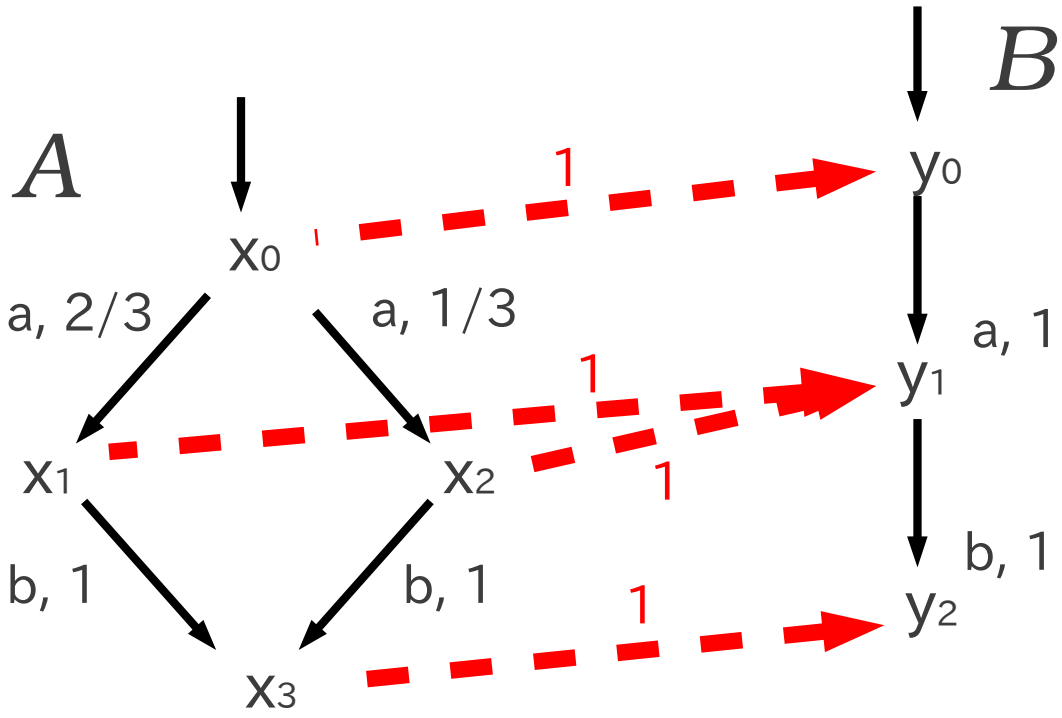
⇒ 線形不等式
(Hasuo, 2010)

確率的オートマトンの Kleisli Simulation

定義:

A からB への backward simulation とは $b : A \to DB$ s.t.

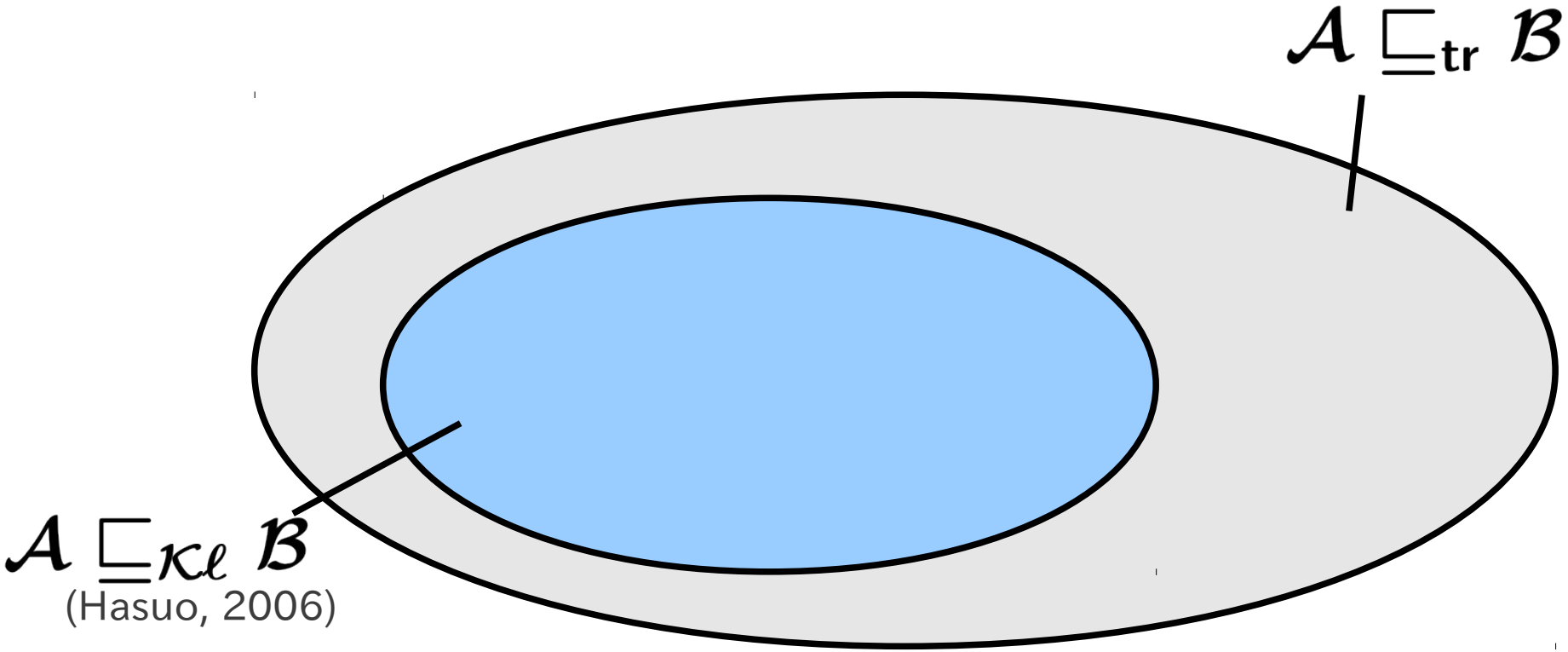
$$\forall x \in A.\, \forall y \in B.\, \forall \sigma \in \Sigma.\ \mathbf{Pr}\left[\begin{array}{c} x \\ \downarrow \sigma \\ \circ \longrightarrow y \end{array}\right] \leq \mathbf{Pr}\left[\begin{array}{c} x \longrightarrow \circ \\ \qquad\quad \downarrow \sigma \\ \qquad\quad y \end{array}\right]$$



⇨ 線形不等式
(Hasuo, 2010)

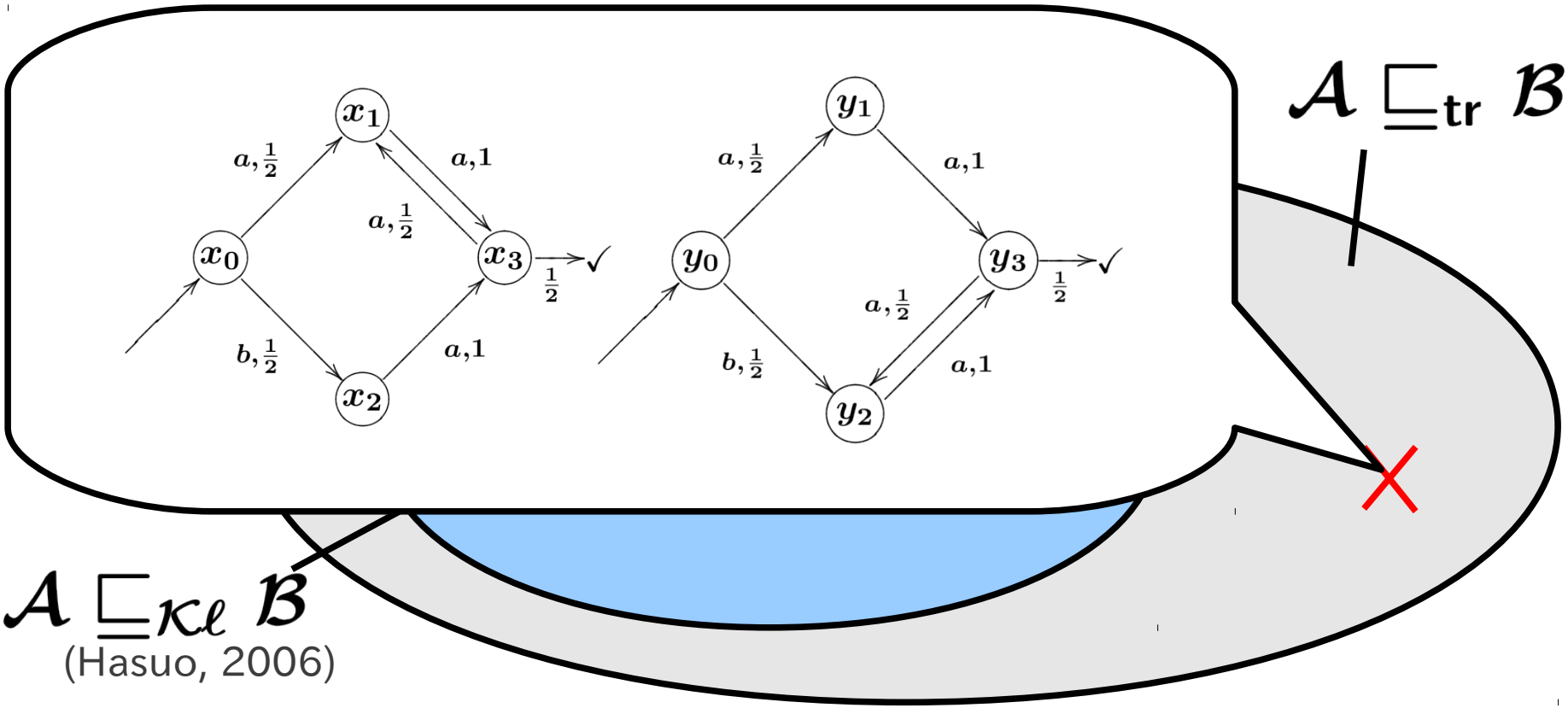
確率的オートマトンのKleisli Simulation

- 確率的オートマトンの trace inclusion は決定不能 (Blondel & Canterni, 2003)



確率的オートマトンのKleisli Simulation

- 確率的オートマトンの trace inclusion は決定不能 (Blondel & Canterni, 2003)



(Hasuo, 2006)

Overview

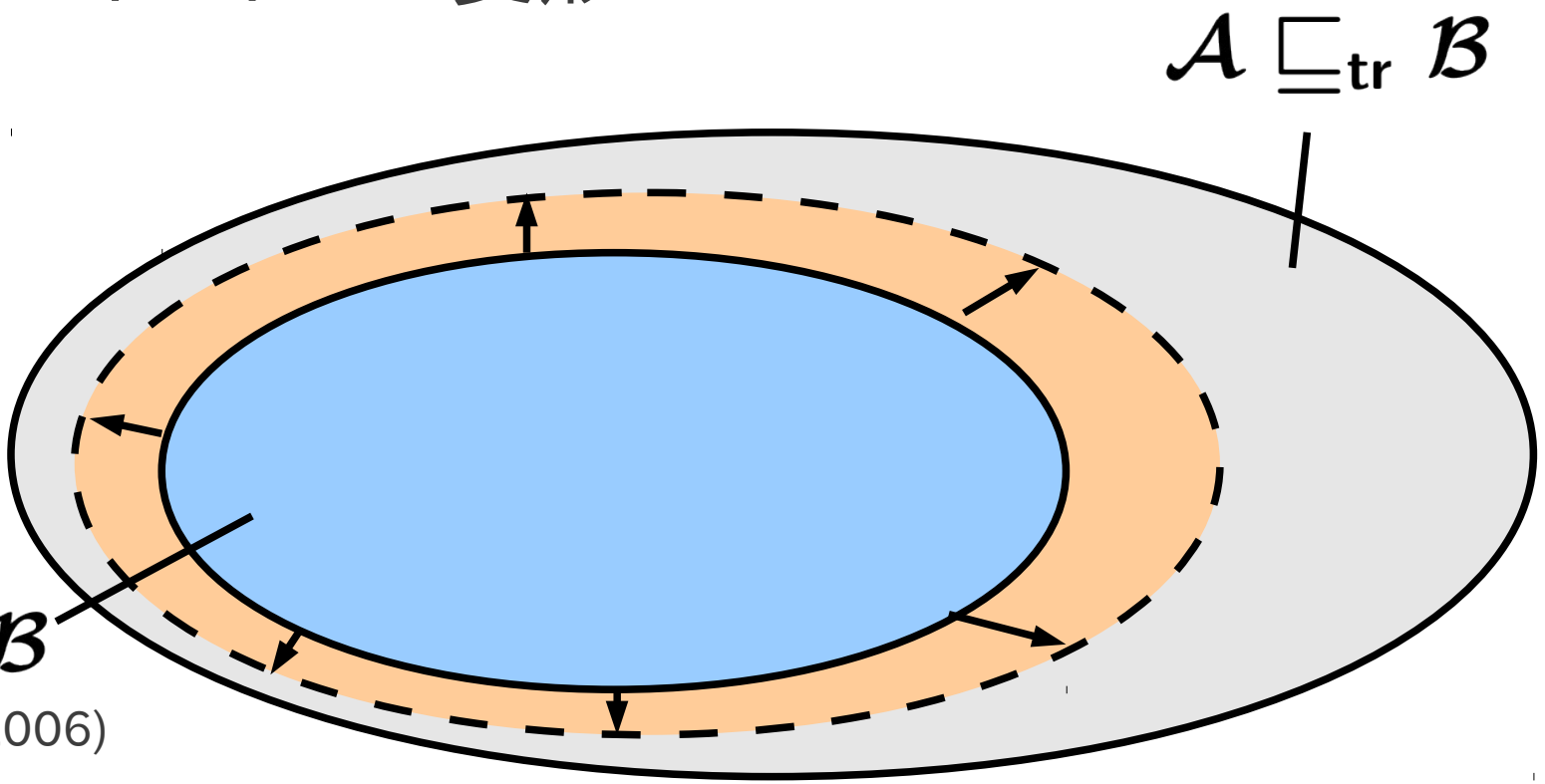
- Simulation
- Kleisli Simulation
- Contribution
  1. Implementation
  2. Increasing the Chance of Simulation
- Experimental Results and Comparison

Contributions

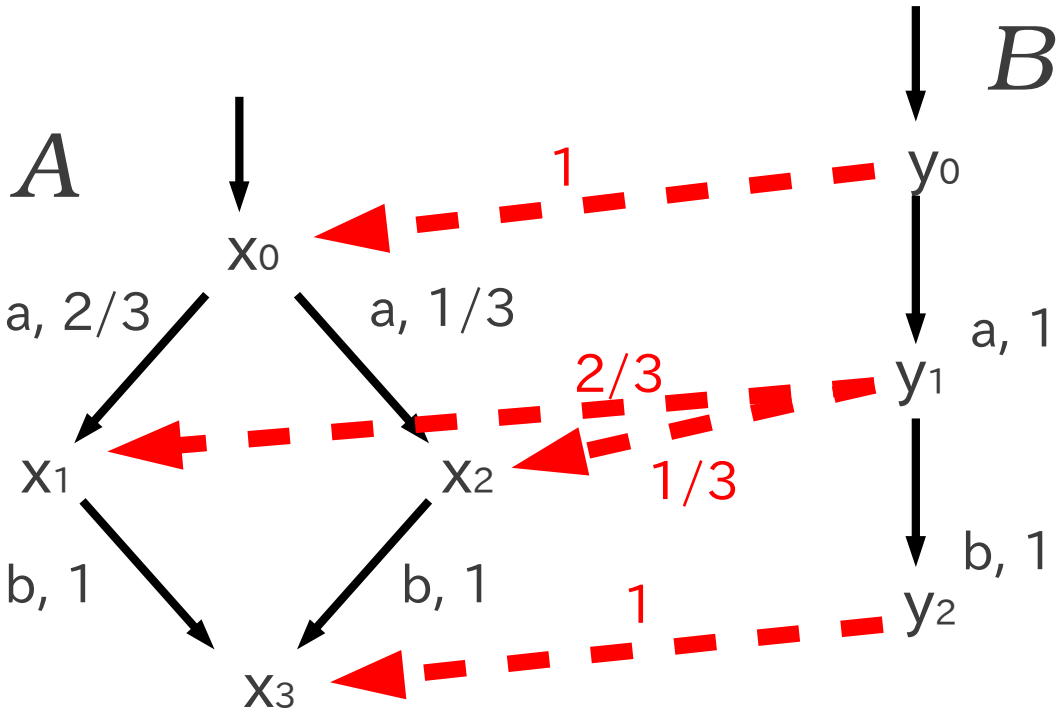
1. 線形不等式を解くことで確率的オートマトンの Kleisli simulation を調べるプログラムを実装

Contributions

2. Kleisli simulation を存在しやすくする工夫
   - 確率 → 実数
   - オートマトンの変形

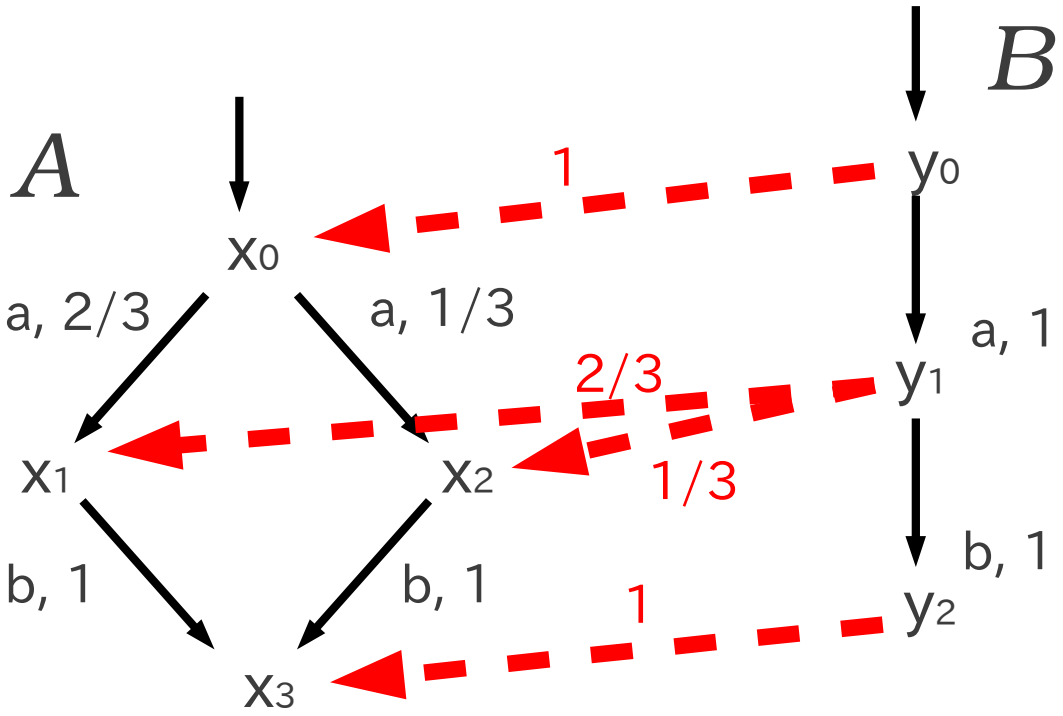


確率 → 実数重み



- 確率的オートマトンのKleisli sim. は確率分布 *[0,1]*

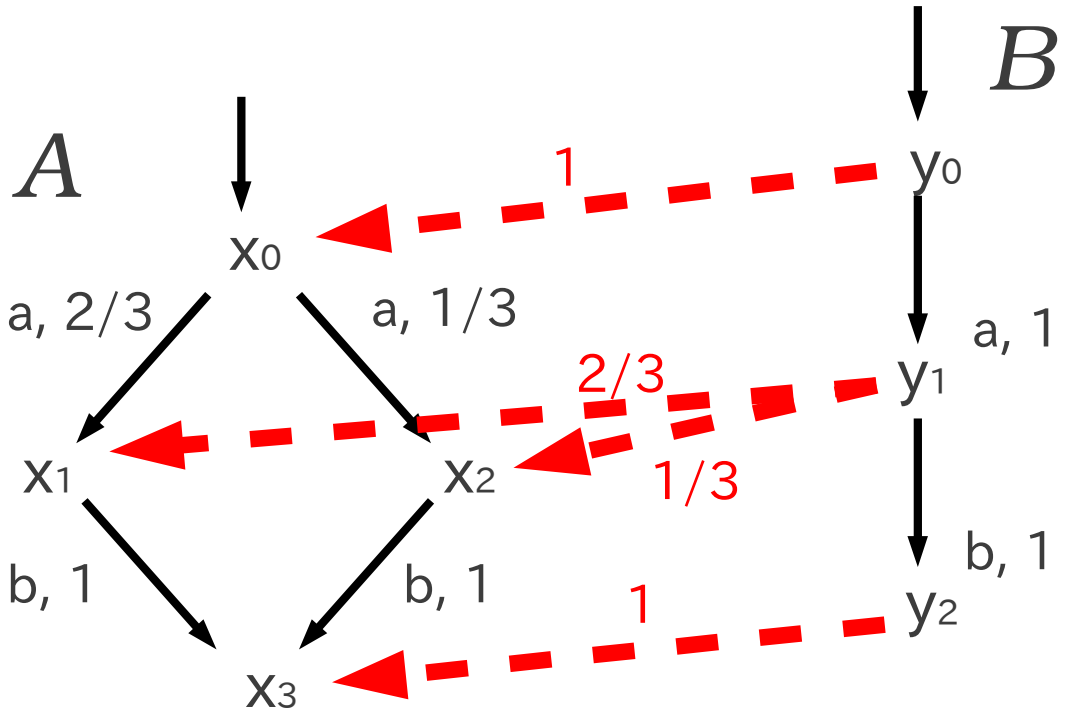
確率 → 実数重み



- 確率的オートマトンのKleisli sim. は確率分布 *[0,1]*

  ⇨ 実数分布 *[0, ∞)* に拡張

確率 → 実数重み



- 確率的オートマトンのKleisli sim. は確率分布 *[0,1]*

  ⇨ 実数分布 *[0, ∞)* に拡張

- モナド: $D \to M$

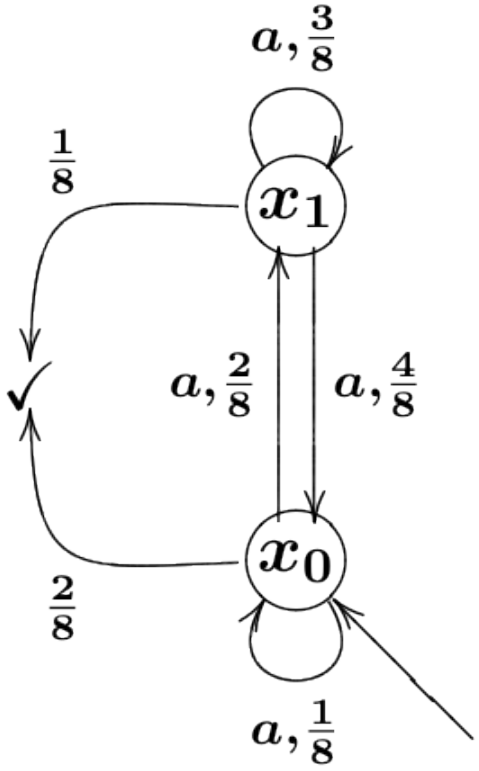
$$DX = \{f : X \to [0,1] \mid \sum_x f(x) \leq 1\}$$

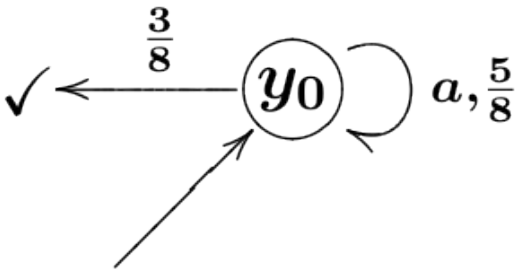
$$MX = \{f : X \to [0,\infty] \mid \mathrm{supp}(f) \text{ is countable}\}$$

システム: 確率的オートマトン→重み付きオートマトン

確率 → 実数重み

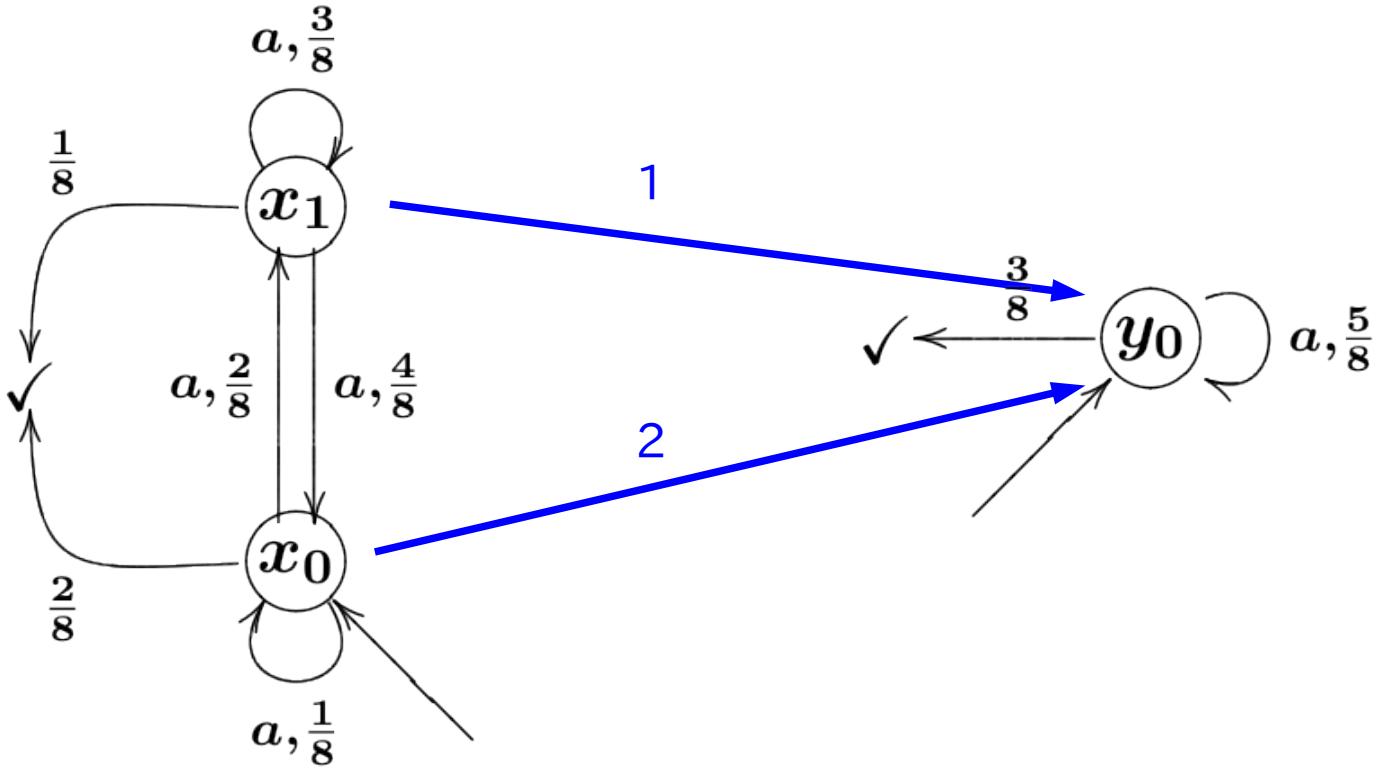
- 新しく Kleisli simulation が見つかることがある





確率 → 実数重み

- 新しく Kleisli simulation が見つかることがある

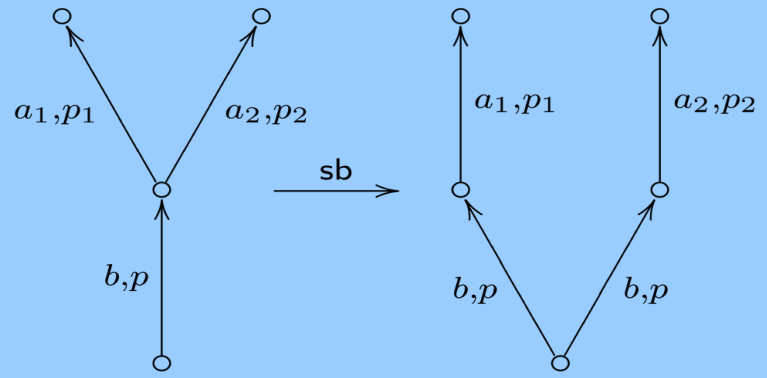


オートマトンの変形

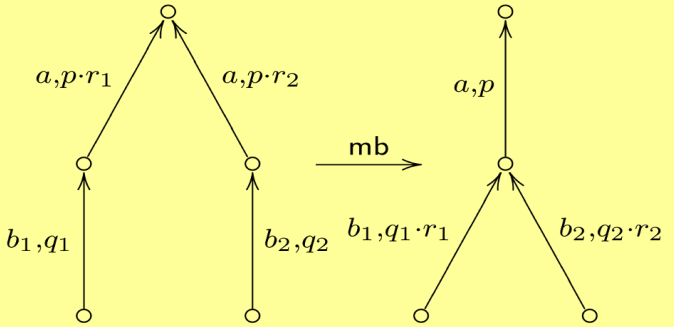
- オートマトンを変形し 、
  Kleisli simulation を存在しやすく

4種の変形

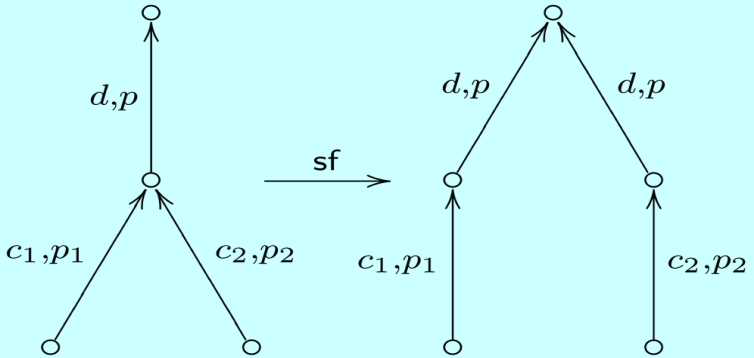
Split Backward



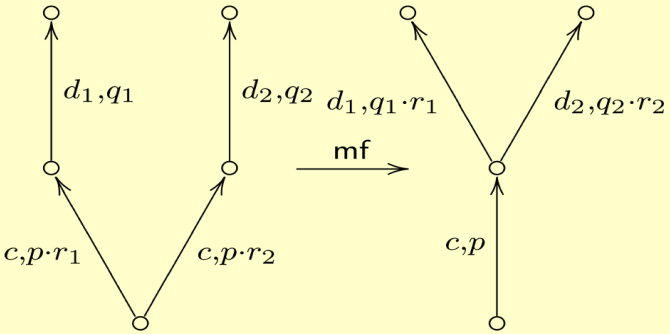
Merge Backward



Split Forward



Merge Forward



変形の性質

- 変形は Kleisli simulation の健全性を保つ

Lemma 3.1.5. *Each of the following implies* $\forall w \in \Sigma^*. \mathcal{A}(w) \leq \mathcal{B}(w)$.

1. $\mathcal{A} \xrightarrow{\mathsf{sb}} \mathcal{A}_{\mathsf{sb}}$ *and* $\mathcal{A}_{\mathsf{sb}} \sqsubseteq_{\mathbf{F}} \mathcal{B}$
2. $\mathcal{B} \xrightarrow{\mathsf{sb}} \mathcal{B}_{\mathsf{sb}}$ *and* $\mathcal{A} \sqsubseteq_{\mathbf{B}} \mathcal{B}_{\mathsf{sb}}$
3. $\mathcal{A} \xrightarrow{\mathsf{sf}} \mathcal{A}_{\mathsf{sf}}$ *and* $\mathcal{A}_{\mathsf{sf}} \sqsubseteq_{\mathbf{B}} \mathcal{B}$
4. $\mathcal{B} \xrightarrow{\mathsf{sf}} \mathcal{B}_{\mathsf{sf}}$ *and* $\mathcal{A} \sqsubseteq_{\mathbf{F}} \mathcal{B}_{\mathsf{sf}}$
5. $\mathcal{A} \xrightarrow{\mathsf{mb}} \mathcal{A}_{\mathsf{mb}}$ *and* $\mathcal{A}_{\mathsf{mb}} \sqsubseteq_{\mathbf{F}} \mathcal{B}$
6. $\mathcal{B} \xrightarrow{\mathsf{mb}} \mathcal{B}_{\mathsf{mb}}$ *and* $\mathcal{A} \sqsubseteq_{\mathbf{B}} \mathcal{B}_{\mathsf{mb}}$
7. $\mathcal{A} \xrightarrow{\mathsf{mf}} \mathcal{A}_{\mathsf{mf}}$ *and* $\mathcal{A}_{\mathsf{mf}} \sqsubseteq_{\mathbf{B}} \mathcal{B}$
8. $\mathcal{B} \xrightarrow{\mathsf{mf}} \mathcal{B}_{\mathsf{mf}}$ *and* $\mathcal{A} \sqsubseteq_{\mathbf{F}} \mathcal{B}_{\mathsf{mf}}$

- 変形は既存の Kleisli simulation を壊さない

Lemma 3.1.6. *Assume* $\mathcal{A} \sqsubseteq_{\mathbf{F}} \mathcal{B}$. *Each of the following holds.*

1. $\mathcal{A} \xrightarrow{\mathsf{sb}} \mathcal{A}_{\mathsf{sb}} \Rightarrow \mathcal{A}_{\mathsf{sb}} \sqsubseteq_{\mathbf{F}} \mathcal{B}$
2. $\mathcal{B} \xrightarrow{\mathsf{sf}} \mathcal{B}_{\mathsf{sf}} \Rightarrow \mathcal{A} \sqsubseteq_{\mathbf{F}} \mathcal{B}_{\mathsf{sf}}$
3. $\mathcal{A} \xrightarrow{\mathsf{mb}} \mathcal{A}_{\mathsf{mb}} \Rightarrow \mathcal{A}_{\mathsf{mb}} \sqsubseteq_{\mathbf{F}} \mathcal{B}$
4. $\mathcal{B} \xrightarrow{\mathsf{mf}} \mathcal{B}_{\mathsf{mf}} \Rightarrow \mathcal{A} \sqsubseteq_{\mathbf{F}} \mathcal{B}_{\mathsf{mf}}$

Assume $\mathcal{A} \sqsubseteq_{\mathbf{B}} \mathcal{B}$. *Each of the following holds.*

5. $\mathcal{B} \xrightarrow{\mathsf{sb}} \mathcal{B}_{\mathsf{sb}} \Rightarrow \mathcal{A} \sqsubseteq_{\mathbf{B}} \mathcal{B}_{\mathsf{sb}}$
6. $\mathcal{A} \xrightarrow{\mathsf{sf}} \mathcal{A}_{\mathsf{sf}} \Rightarrow \mathcal{A}_{\mathsf{sf}} \sqsubseteq_{\mathbf{B}} \mathcal{B}$
7. $\mathcal{B} \xrightarrow{\mathsf{mb}} \mathcal{B}_{\mathsf{mb}} \Rightarrow \mathcal{A} \sqsubseteq_{\mathbf{B}} \mathcal{B}_{\mathsf{mb}}$
8. $\mathcal{A} \xrightarrow{\mathsf{mf}} \mathcal{A}_{\mathsf{mf}} \Rightarrow \mathcal{A}_{\mathsf{mf}} \sqsubseteq_{\mathbf{B}} \mathcal{B}$

変形の性質

- これ以上変形できないオートマトンの間の Kleisli simulation は**完全**

Theorem 3.2.6. *Let $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma, M_{\mathcal{A}}, \alpha_{\mathcal{A}}, \eta_{\mathcal{A}})$ and $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, M_{\mathcal{B}}, \alpha_{\mathcal{B}}, \eta_{\mathcal{B}})$ be S-weighted automata. Assume $(\mathcal{M}_S, F)$-systems $\mathcal{X}_{\mathcal{A}}$ and $\mathcal{X}_{\mathcal{B}}$ be their representations.*

1. *Assume below.*
   - $\mathcal{A} \overset{\mathsf{sb}}{\not\to}$.
   - $\mathcal{A} \overset{\mathsf{mb}}{\not\to}$.
   - *Each transition weights of $\mathcal{A}$ has its inverse element for multiplication.*

   Then the following statements hold:

   (a) *If all states of $\mathcal{A}$ is reachable to the final state,* $\mathcal{X}_{\mathcal{A}} \sqsubseteq_{\mathbf{F}} \mathcal{X}_{\mathcal{B}} \Leftrightarrow \forall w \in \Sigma^*.\ \mathcal{A}(w) \leq \mathcal{B}(w)$.

   (b) *If all states of $\mathcal{B}$ is reachable from the initial state,* $\mathcal{X}_{\mathcal{B}} \sqsubseteq_{\mathbf{B}} \mathcal{X}_{\mathcal{A}} \Leftrightarrow \forall w \in \Sigma^*.\ \mathcal{B}(w) \leq \mathcal{A}(w)$.

2. *Assume below.*
   - $\mathcal{A} \overset{\mathsf{sf}}{\not\to}$.
   - $\mathcal{A} \overset{\mathsf{mf}}{\not\to}$.
   - *Each transition weights of $\mathcal{A}$ has its inverse element for multiplication.*
   - *The transposes of $\mathcal{A}$ and $\mathcal{B}$ can be defined.*

   Then the following statements hold:

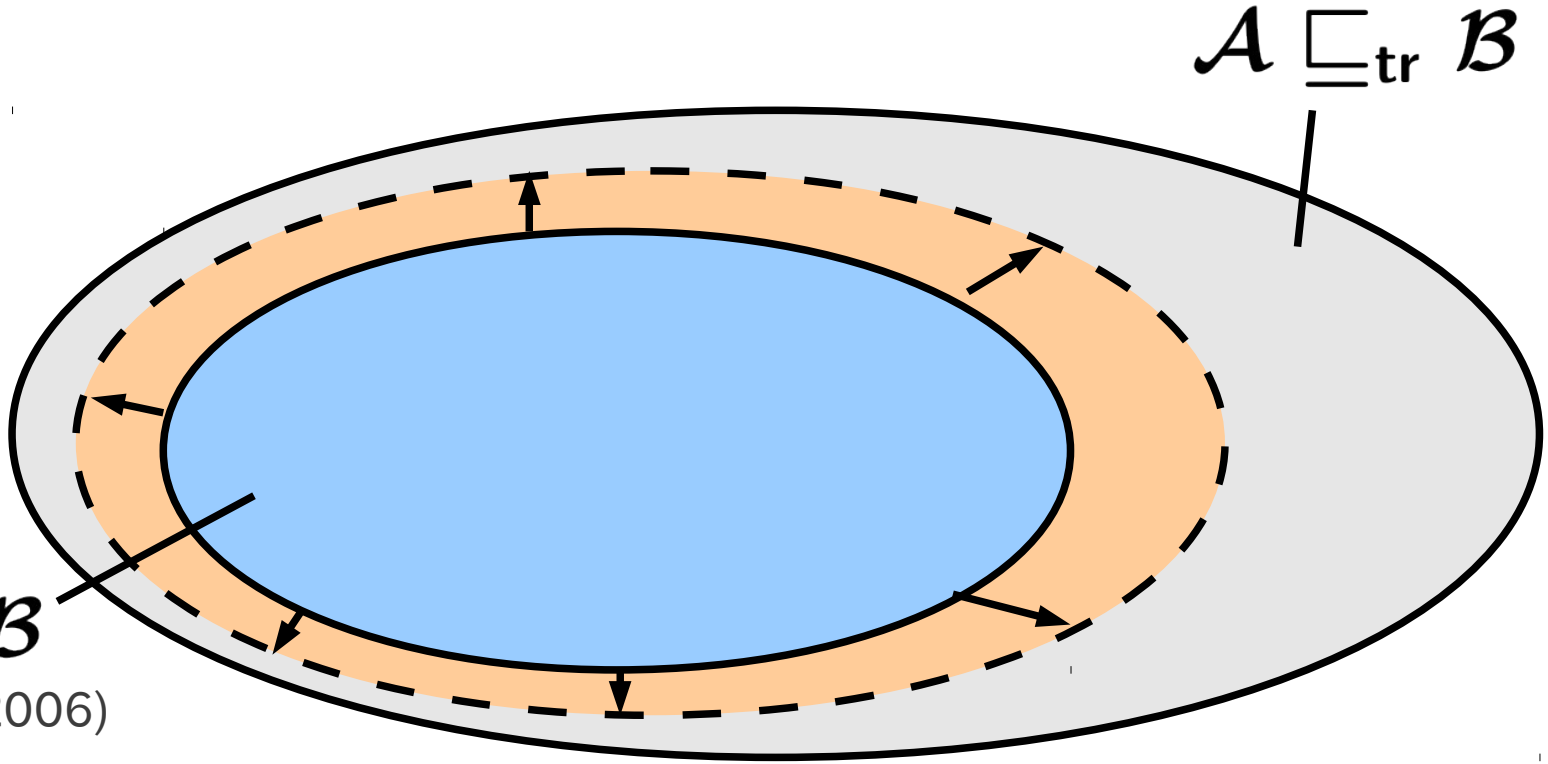
   (a) *If all states of $\mathcal{A}$ is reachable from the initial state,* $\mathcal{X}_{\mathcal{A}} \sqsubseteq_{\mathbf{B}} \mathcal{X}_{\mathcal{B}} \Leftrightarrow \forall w \in \Sigma^*.\ \mathcal{A}(w) \leq \mathcal{B}(w)$.

   (b) *If all states of $\mathcal{B}$ is reachable to the final state,* $\mathcal{X}_{\mathcal{B}} \sqsubseteq_{\mathbf{F}} \mathcal{X}_{\mathcal{A}} \Leftrightarrow \forall w \in \Sigma^*.\ \mathcal{B}(w) \leq \mathcal{A}(w)$.

- 普通は無限回変形できる

変形の性質

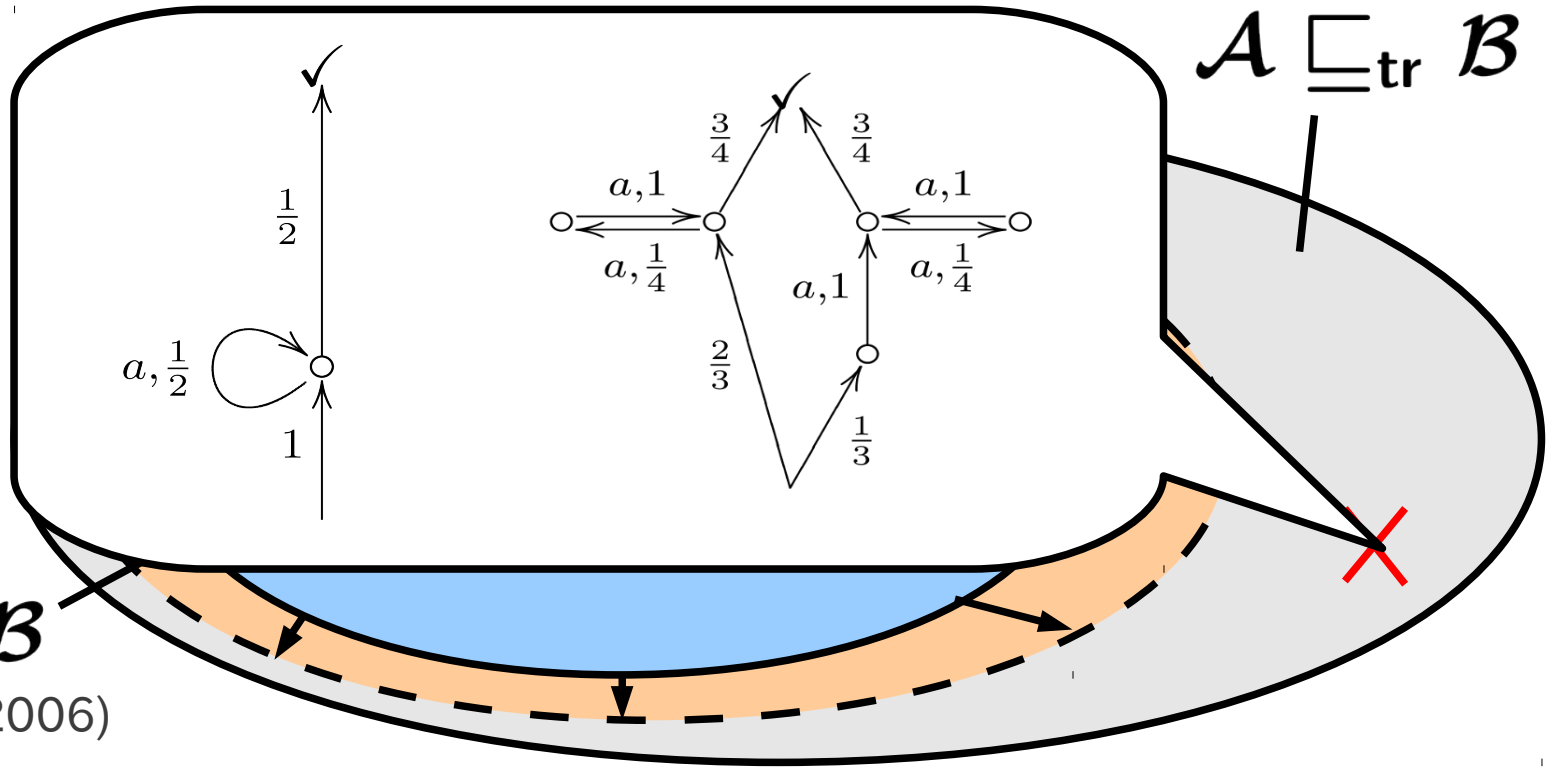
- 確率的オートマトンのtrace inclusion は半決定不能



(Hasuo, 2006)

変形の性質

- 確率的オートマトンのtrace inclusion は半決定不能
- 何度変形してもKleisli simulationが
  存在するようにならない



(Hasuo, 2006)

Overview

- Simulation
- Kleisli Simulation
- Contribution
  1. Implementation
  2. Increasing the Chance of Simulation
- Experimental Results and Comparison

実装

- Kleisli simulation を探すプログラム
  - 線形計画問題ソルバを用いて線形不等式を解く
  - glpk
  - 単体法
- オートマトンを変形するプログラム

実験結果

- Grade protocol の正しさの検証
- オートマトンの等価性を調べることで検証できる

parm		protocol		specific			type		time(sec)		space	ex
G	S	st	tr	st	tr	al	dir	f/b	user	real	(GB)	
2	8	578	1522	130	642	11	$p \to s$	fwd	1.77	2.00	1.21	E
								bwd	2.03	2.20	1.21	N
							$s \to p$	fwd	1.94	2.08	1.14	N
								bwd	1.72	1.88	1.22	E
2	10	1102	2982	202	1202	13	$p \to s$	fwd	9.42	9.95	4.05	E
								bwd	11.51	12.00	4.08	N
							$s \to p$	fwd	10.99	11.45	3.82	N
								bwd	9.25	9.81	4.09	E
2	12	1874	5162	290	2018	15	$p \to s$	fwd	38.60	40.16	11.51	E
								bwd	49.03	50.38	11.60	N
							$s \to p$	fwd	47.99	49.40	10.83	N
								bwd	38.34	39.78	11.63	E
3	8	1923	7107	243	2163	20	$p \to s$	fwd	44.43	62.04	12.26	E
								bwd	56.92	79.31	12.59	N
							$s \to p$	fwd	55.48	57.12	12.27	N
								bwd	44.11	67.60	12.64	E
4	6	1636	7468	196	1924	23	$p \to s$	fwd	30.28	31.64	10.39	E
								bwd	38.79	39.97	10.49	N
							$s \to p$	fwd	37.86	39.10	9.79	N
								bwd	29.94	31.25	10.49	E

(Kiefer et al. 2011)

- Space consuming
- Kieferらの実装より遅い
  - Trace inclusionは trace equivalenceより難しい

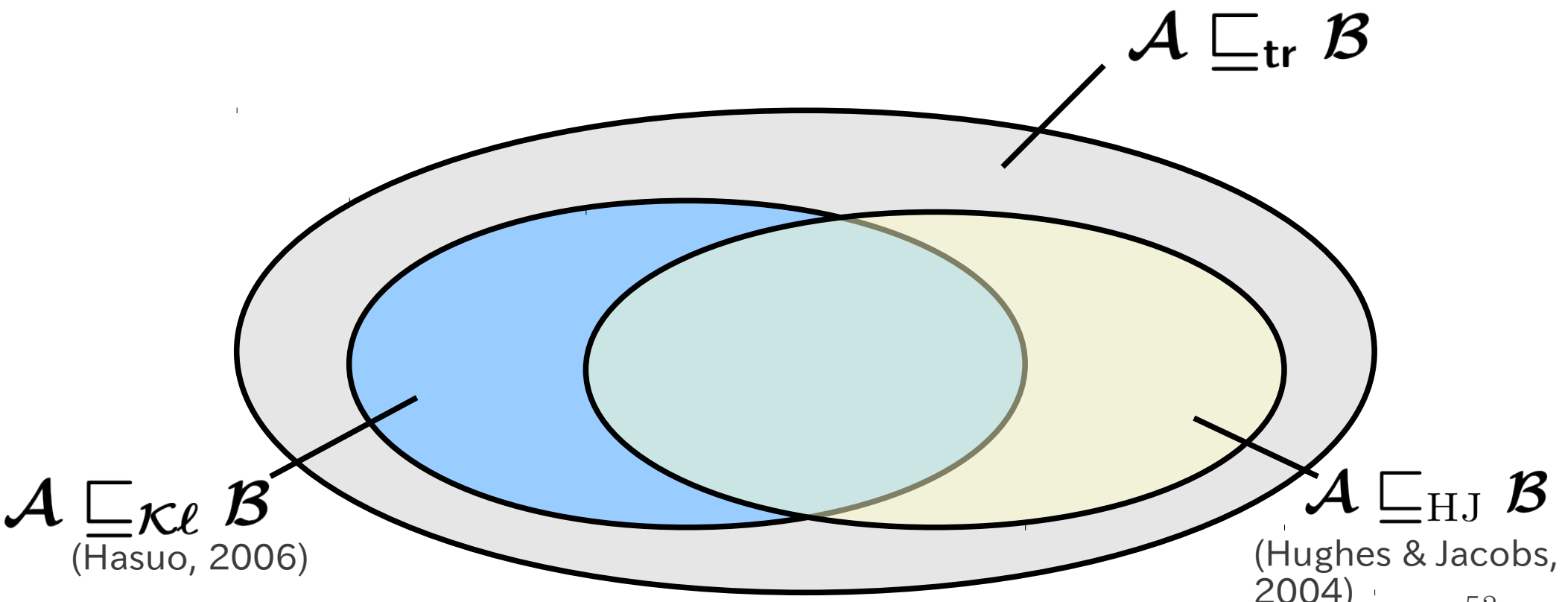
実験結果

- Crowds protocol の probable innocence の検証
- Trace inclusion を調べることで検証できる (Hasuo, Kawabe, Sakurada, 2010)
- 変形が必要になる場合が発生→変形・探索を繰り返す

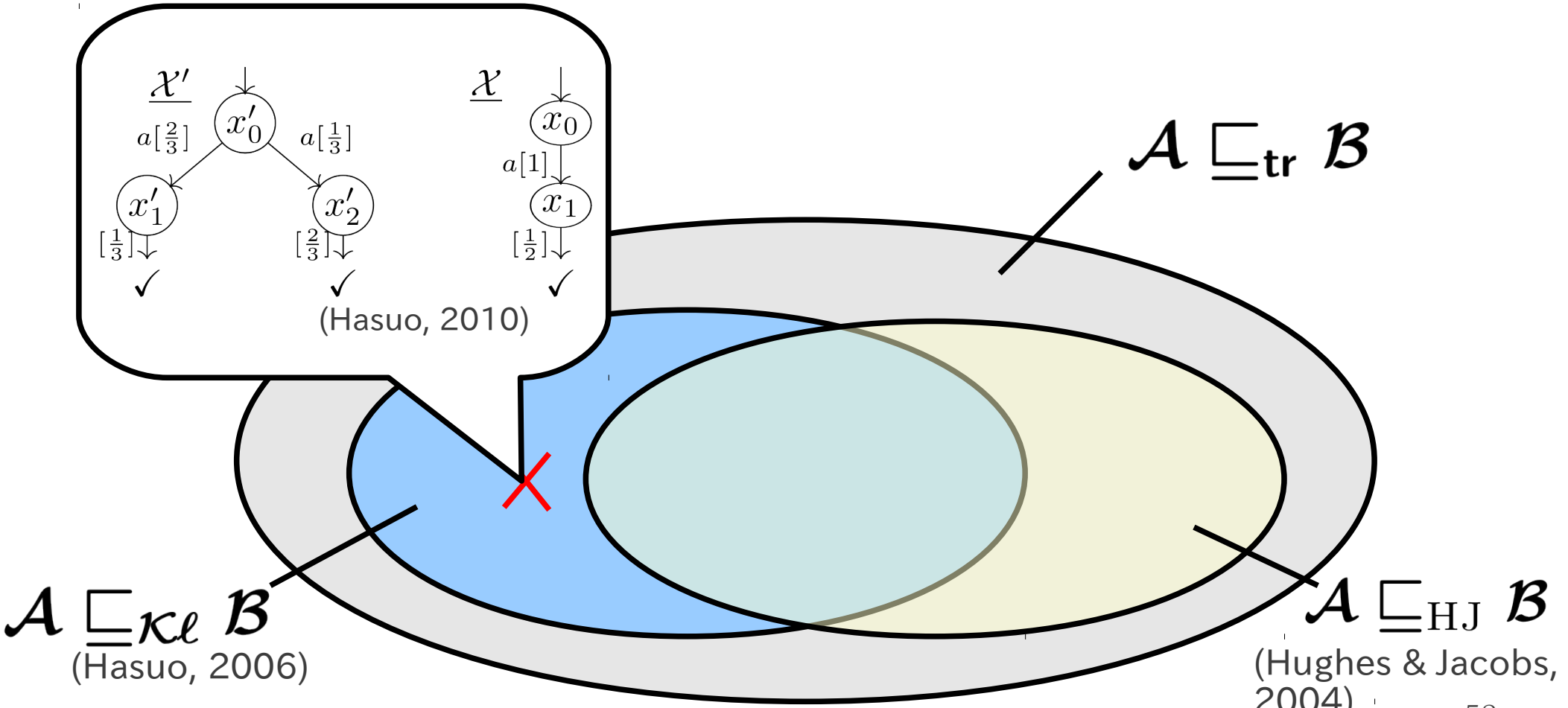
parm			orig		inno				time(sec)		space	iter
n	c	p_f	st	tr	st	tr	al	f/b	user	real	(GB)	
5	1	$\frac{9}{10}$	7	44	7	56	18	fwd	0.10	0.18	0.08	2R
								bwd	0.00	0.06	0.01	2L
7	1	$\frac{3}{4}$	9	88	9	118	26	fwd	0.90	1.06	0.53	2R
								bwd	0.01	0.07	0.01	2L
10	2	$\frac{4}{5}$	12	224	12	280	54	fwd	71.72	152.39	12.87	2R
								bwd	0.06	0.12	0.03	2L
20	6	$\frac{4}{5}$	22	1514	22	1696	238	fwd	OOM			
								bwd	1.29	1.49	0.78	2L
30	6	$\frac{4}{5}$	32	4732	32	5112	550	fwd	SF			
								bwd	11.80	13.11	5.99	2L

- Backward simulation is faster

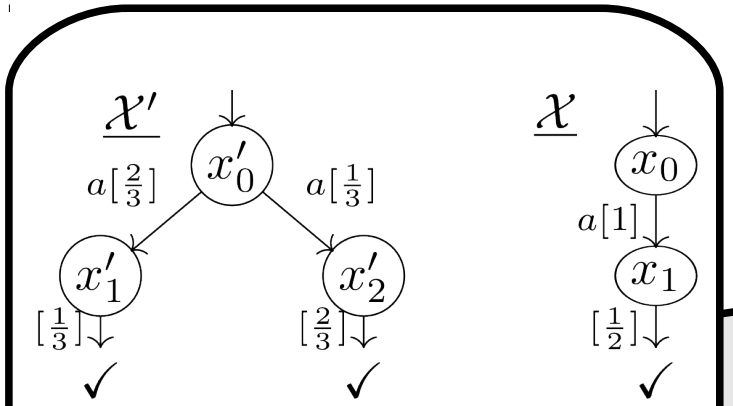
他のSimulationとの比較



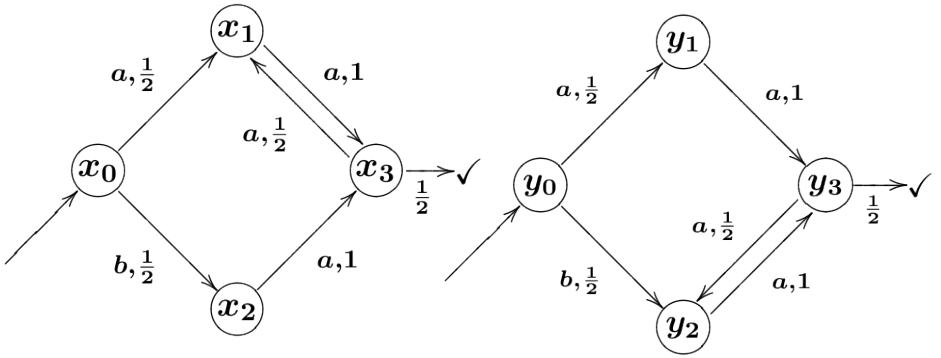
他のSimulationとの比較



他のSimulationとの比較



(Hasuo, 2010)



$\mathcal{A} \sqsubseteq_{\mathcal{KL}} \mathcal{B}$
(Hasuo, 2006)

$\mathcal{A} \sqsubseteq_{\mathrm{HJ}} \mathcal{B}$
(Hughes & Jacobs, 2004)

まとめ

- Kleisli simulationを探すプログラムを実装
- Simulation を存在しやすくする工夫

Future Work

- オートマトンの変形を圏論的に
- 他のシステムに対するKleisli simulationの実装