更新可能時間オートマトンの新たな拡張について

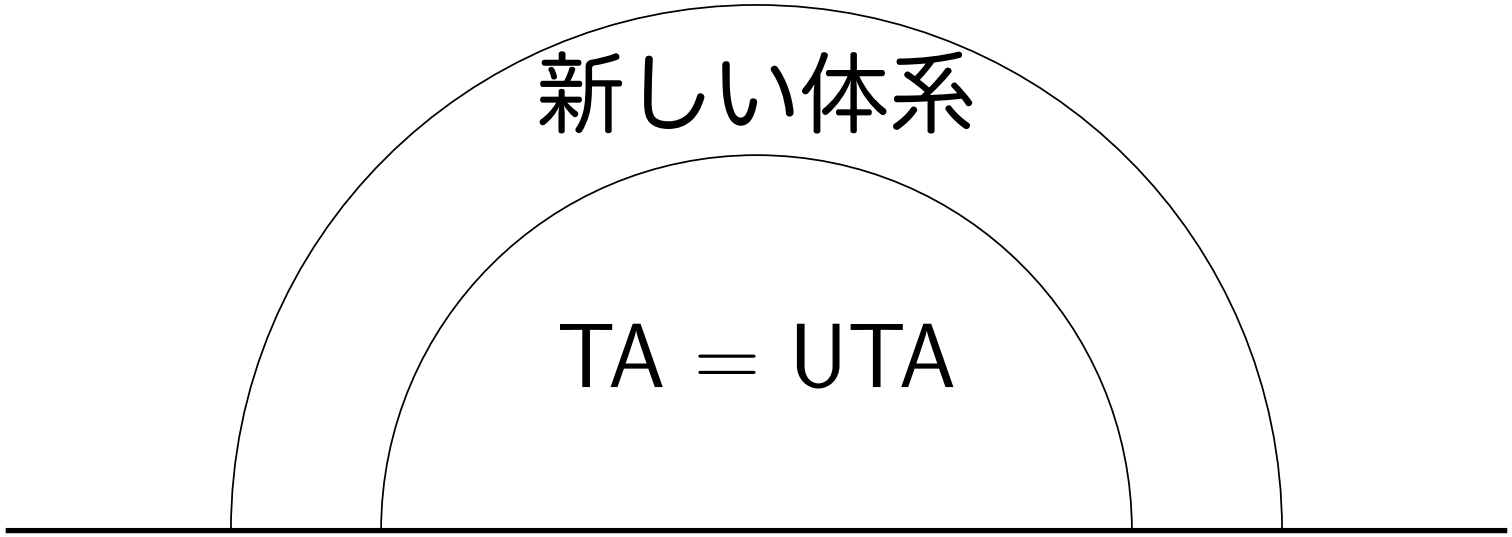
◎上里 友弥（筑波大学）
南出 靖彦（東京工業大学）

JSSST 2015，9 月 9 日 at 早稲田大学

2 つの結果

- 時間オートマトン（Timed Automaton），1994 年，Alur & Dill.
- 更新可能（Updatable）時間オートマトン，2000 年，Bouyer ら.

① 言語クラス（表現力）を拡大：

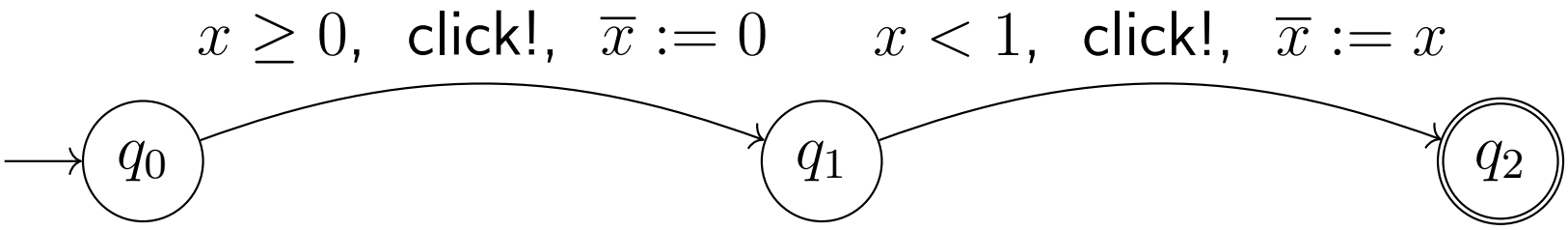


② 到達可能性問題 $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \Rightarrow^* q_{\mathsf{goal}}$ が決定可能 

- 時間オートマトンベースのモデル検査器（UPPAAL など）で基盤となる判定問題.

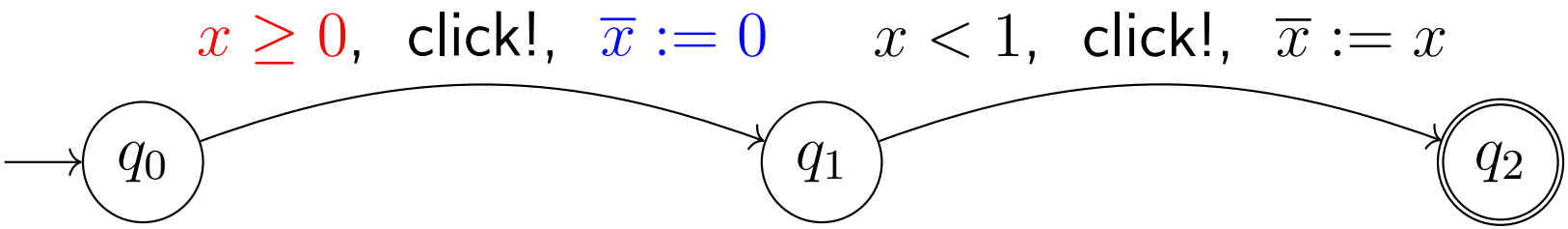
時間オートマトン = オートマトン + クロック

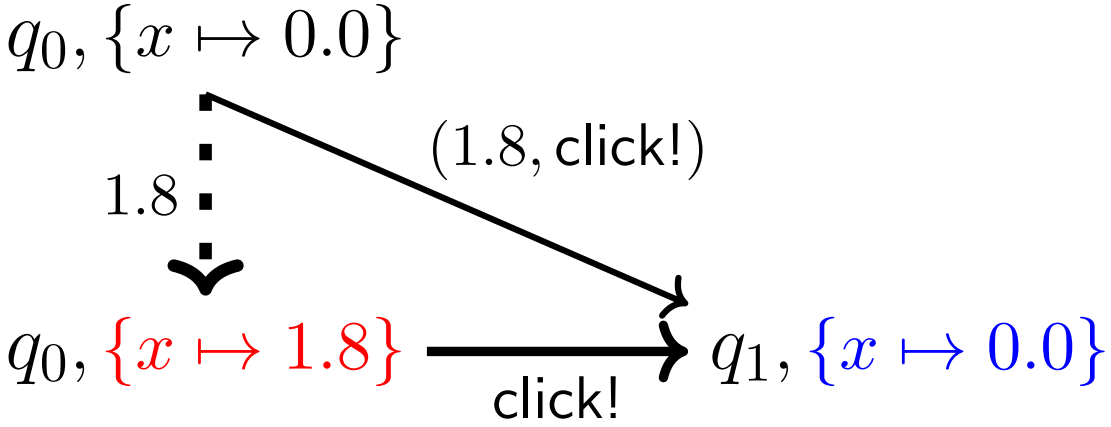
計算の基本単位：$\langle q, \{x_1 \mapsto r_1, \ldots, x_n \mapsto r_n\} \rangle \quad (r_i \in \mathbb{Q}_{\geq 0})$



時間オートマトン = オートマトン + クロック

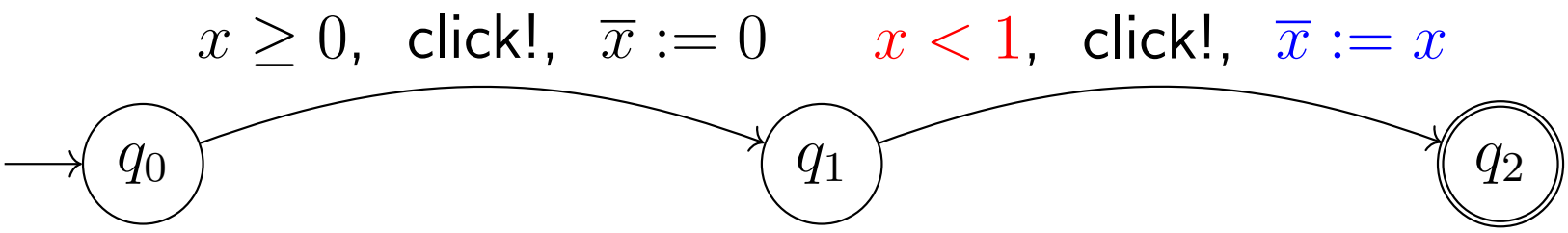
計算の基本単位：$\langle q, \{x_1 \mapsto r_1, \ldots, x_n \mapsto r_n\}\rangle \quad (r_i \in \mathbb{Q}_{\geq 0})$

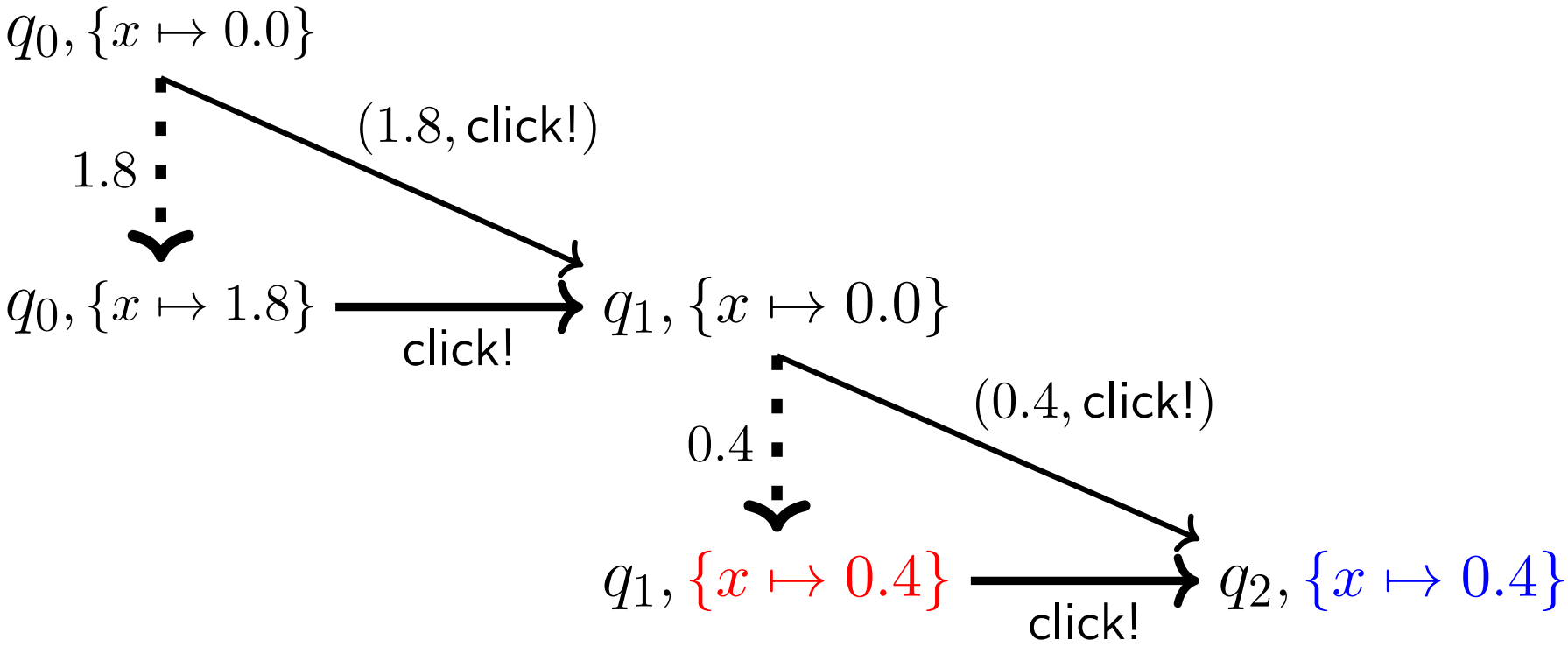




時間オートマトン = オートマトン + クロック

計算の基本単位：$\langle q, \{x_1 \mapsto r_1, \ldots, x_n \mapsto r_n\} \rangle \quad (r_i \in \mathbb{Q}_{\geq 0})$





$$\frac{\nu \models \varphi \qquad \nu' \in \mathbf{up}(\nu)}{\langle p, \nu \rangle \xrightarrow{\varphi,\ event,\ \mathbf{up}} \langle q, \nu' \rangle}$$

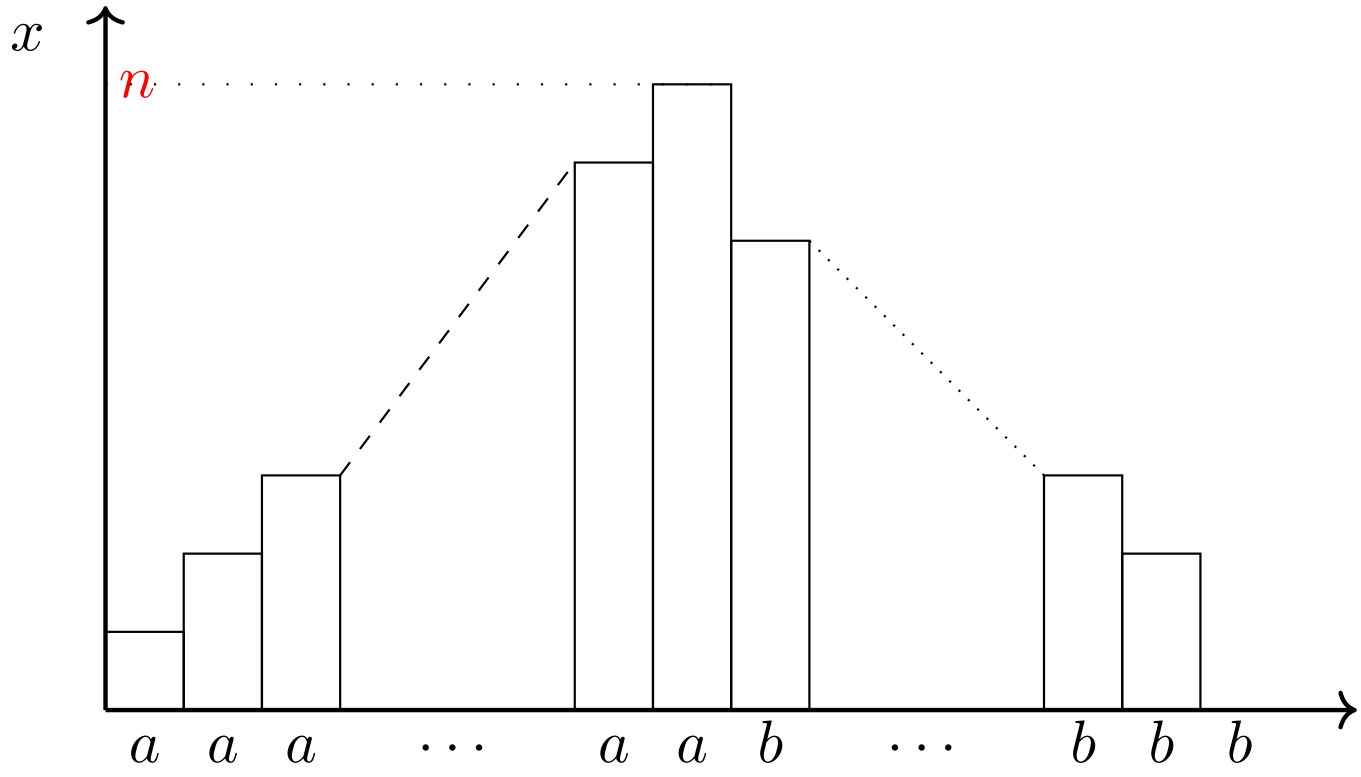
体系	制約式 φ	更新式 $\mathbf{up}$
時間オートマトン	$x \gtreqless k$	$\overline{x} := 0,\ \overline{x} := x$
我々の体系	$x \gtreqless k,$ $x \gtreqless y,$ $\mathrm{nat}(x)$	$\overline{x} \gtreqless k,$ $\overline{x} \gtreqless y,$ $\mathrm{nat}(\overline{x})$

ただし，$k \in \mathbb{N}$;　　x, y はクロック;

言語の例

$$L_{ab} \triangleq \{\overbrace{(1,a)(1,a)\ldots(1,a)}^{n}\overbrace{(0,b)\ldots(0,b)}^{m} : m \leq n\}.$$

一秒ごとに a　　ノータイムで b



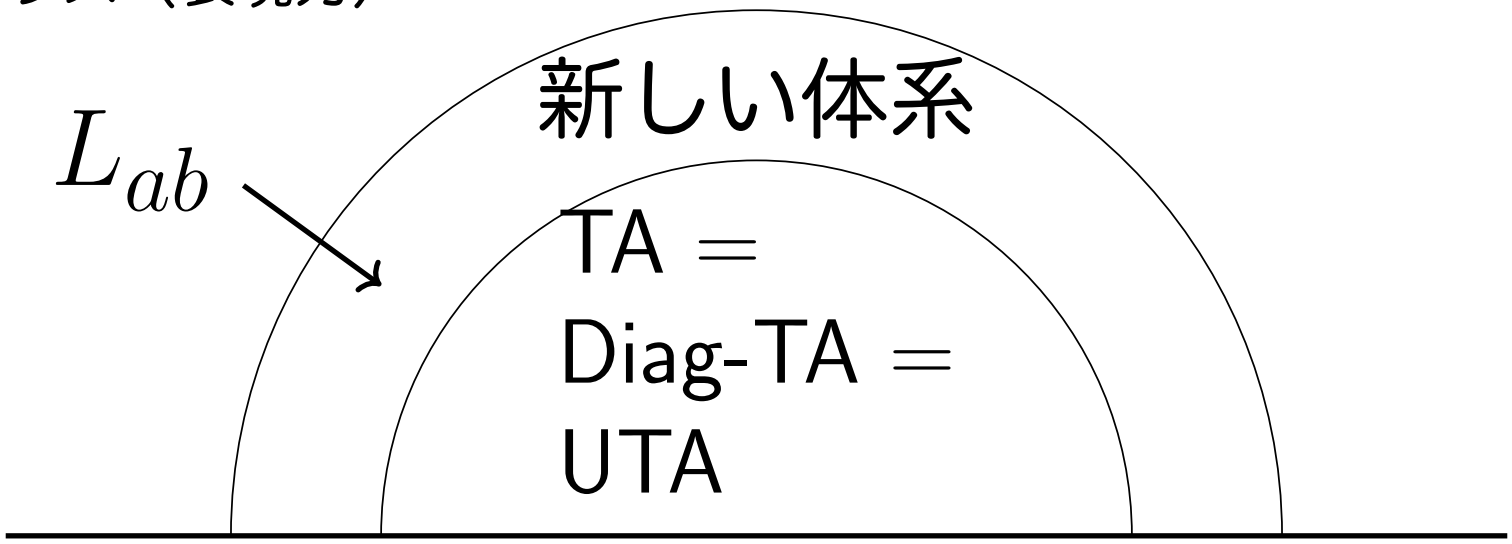
$\mathbf{up} \equiv \overline{x} < x \wedge \mathrm{nat}(\overline{x})$ で,

現在の x 未満 $(\overline{x} < x)$ の自然数値 $(\mathrm{nat}(x))$ になるよう更新.

我々の体系の位置づけ　もしくは　本研究の結果

- TA：時間オートマトン，1994 年，Alur & Dill；
- Diag-TA：対角線制約付き TA，1994 年，Alur & Dill；
- UTA：先行研究の更新可能時間オートマトン，2000 年，Bouyer ら.

言語クラス（表現力）



TA で受理できるならば、以下が正規言語でなければならない：

$$Untime(L_{ab}) = \{\overbrace{aa\ldots a}^{n}\overbrace{b\ldots b}^{m} : 0 < m \leq n\}.$$

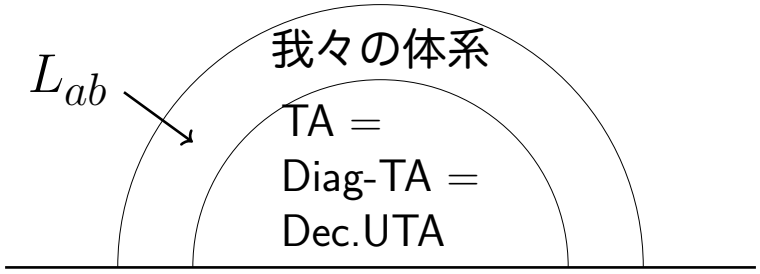
命令セットに基づく既存体系との比較

体系	制約式 φ	更新式 $\mathbf{up}$	到達可能性
TA	$x \approx k$	$\overline{x} = x,$ $\overline{x} = 0$	PSPACE 完全
Diag-TA	$x \approx k,$ $x - y \approx k$	$=$ TA	PSPACE 完全
Bouyer の UTA	$=$ Diag	TA $+ \overline{x} \leq k$	PSPACE 完全
	$=$ Diag	TA $+ \overline{x} \geq k$	決定不能
我々の UTA	$x \approx k,$ $x \approx y,$ $\mathrm{nat}(x)$	$\overline{x} \approx k,$ $\overline{x} \approx y,$ $\mathrm{nat}(\overline{x})$	決定可能

- Bouyer ら：制約「$x - y = 1$」と更新 $\overline{x} \geq 0$ でチューリング完全.
- 我々：$k = 0$ の形 $x \approx y$ に制限 + $\mathrm{nat}(\cdot)$ を導入.

我々の体系の位置づけ　もしくは　本研究の結果

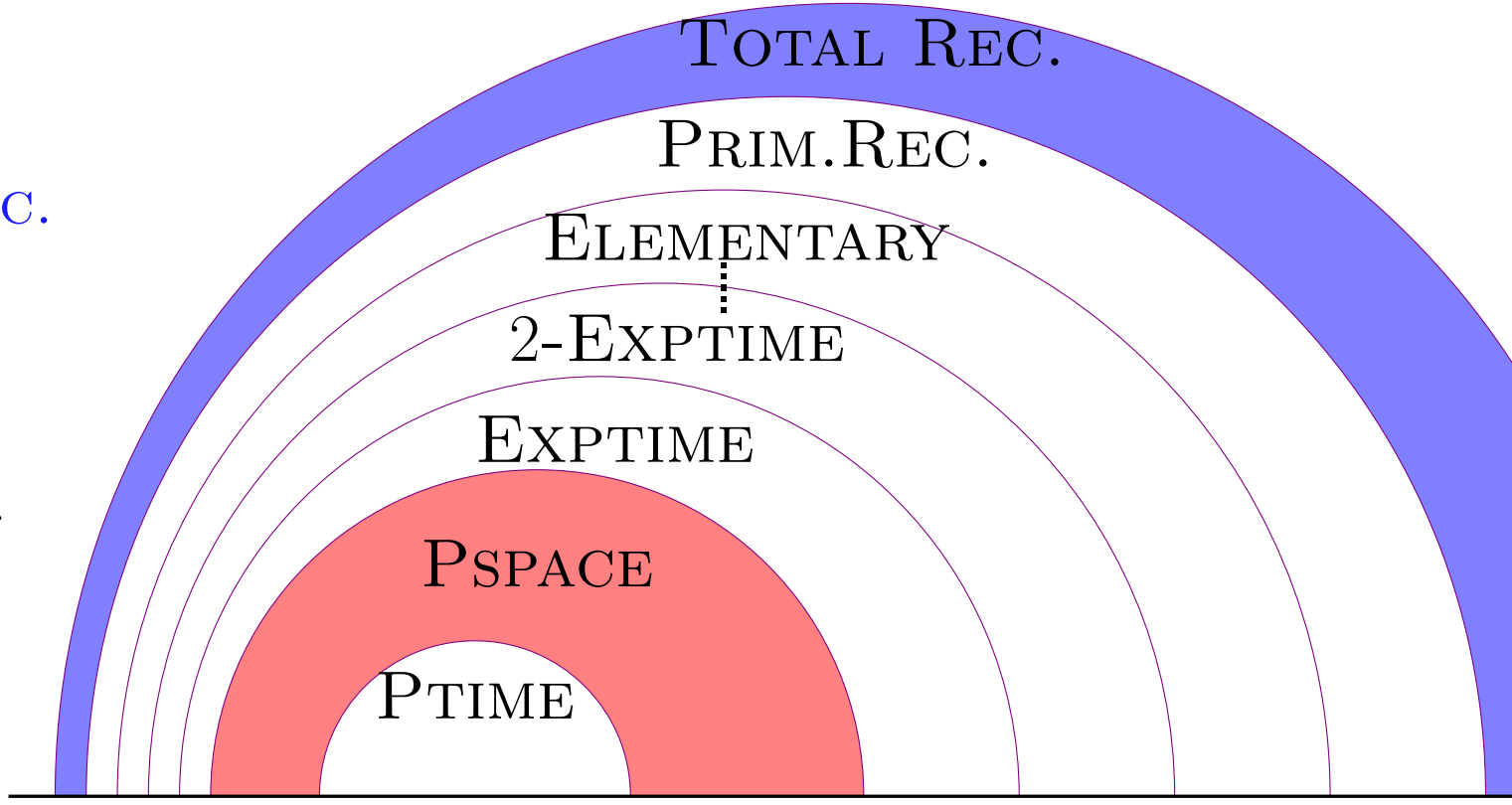
言語クラス（表現力）



到達可能性問題 $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \Rightarrow^* q_{\mathsf{goal}}$ の計算量

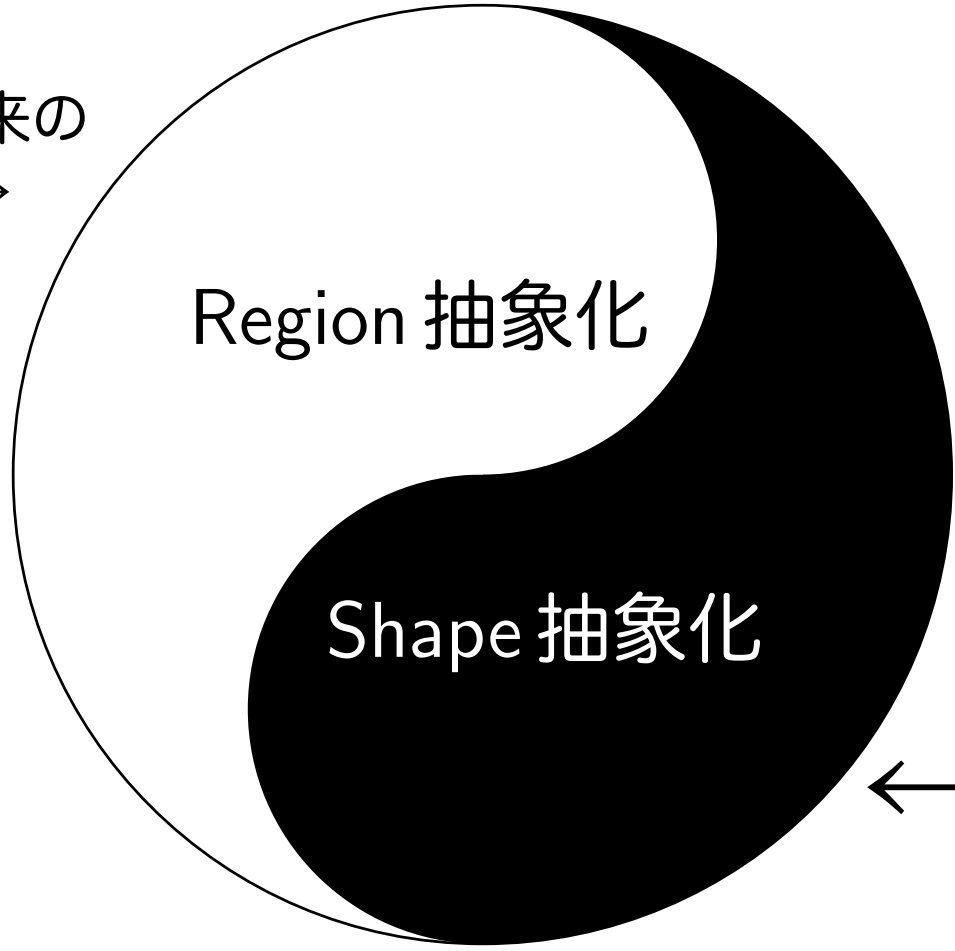
我々の体系：
NON PRIM.REC.

{ TA, Diag-TA, Dec.UTA }
PSPACE 完全



我々の体系での決定可能性の証明手法

Alur & Dill 以来の
伝統手法 →



← New!!

Region 抽象化は，クロックの小数部分を扱う．
Shape 抽象化は，クロックのカウンター的な振る舞いを扱う．
⇒ Shape 抽象化の説明のために，離散的な体系を考える．

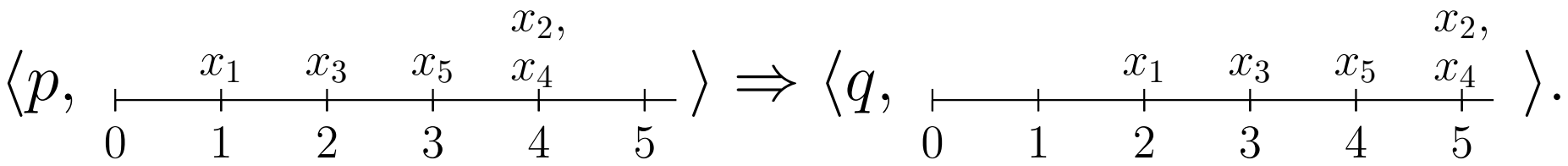
離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

計算状況 $\langle q, \mu \rangle$ は，状態 $q \in Q$ と，自然数上の割当 $\mu : \mathcal{X} \to \mathbb{N}$.

離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

計算状況 $\langle q, \mu \rangle$ は，状態 $q \in Q$ と，自然数上の割当 $\mu : \mathcal{X} \to \mathbb{N}$.

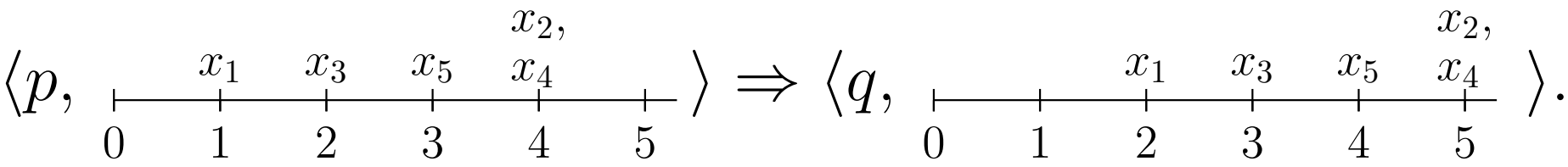
- $(p, q, \textsc{Time})$ は全てのクロック値を「1」増やす：



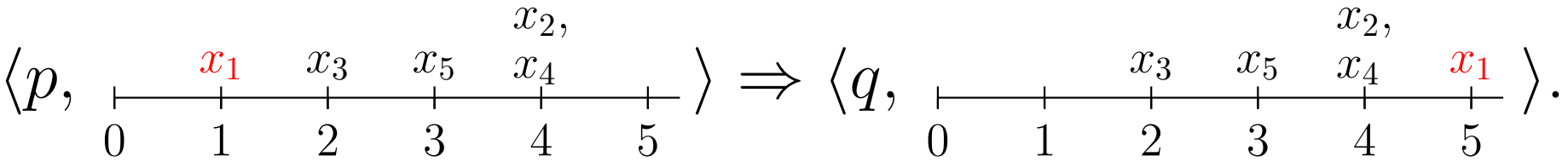
離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

計算状況 $\langle q, \mu \rangle$ は，状態 $q \in Q$ と，自然数上の割当 $\mu : \mathcal{X} \to \mathbb{N}$.

- $(p, q, \textsc{Time})$ は全てのクロック値を「1」増やす：



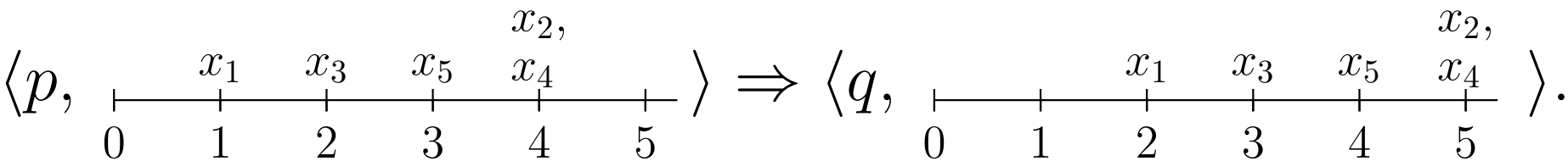
- $(p, q, \textsc{Upd}(x))$ は非決定的更新 $x \leftarrow \mathsf{rand}()$：



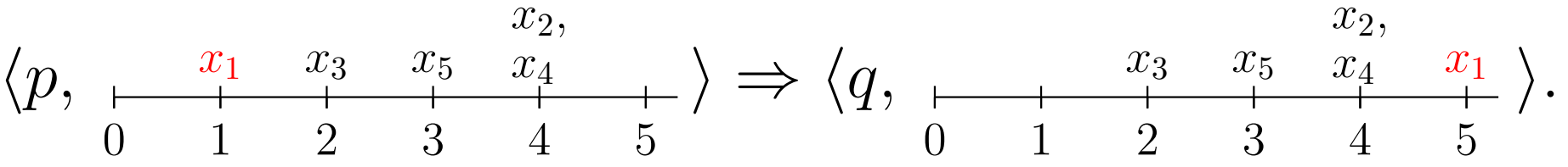
離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

計算状況 $\langle q, \mu \rangle$ は，状態 $q \in Q$ と，自然数上の割当 $\mu : \mathcal{X} \to \mathbb{N}$.

- $(p, q, \textsc{Time})$ は全てのクロック値を「1」増やす：



- $(p, q, \textsc{Upd}(x))$ は非決定的更新 $x \leftarrow \mathsf{rand}()$：



- $(p, q, \textsc{Test}(\varphi))$ は条件を満足するか検査：

$$\mu \models \varphi \text{ ならば } \langle p, \mu \rangle \Rightarrow \langle q, \mu \rangle.$$

離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

条件式は 2 種類：

- クロックと定数の比較：$\mu \models x \begin{matrix} \geqq \\ \leqq \end{matrix} k \quad (x \in \mathcal{X}, k \in \mathbb{N})$.
- クロック同士の比較：$\mu \models x \begin{matrix} \geqq \\ \leqq \end{matrix} y \quad (x, y \in \mathcal{X})$.

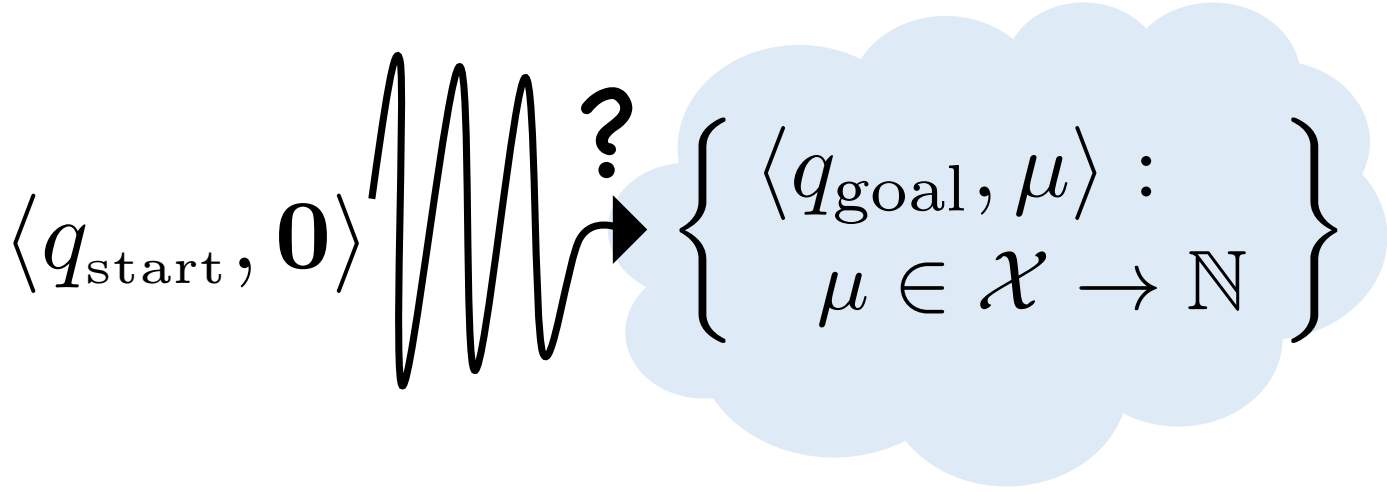
離散時間遷移系 $\mathcal{S} = (Q, \mathcal{X}, \Delta)$

条件式は 2 種類：

- クロックと定数の比較：$\mu \models x \begin{matrix} \geqq \\ \leqq \end{matrix} k \quad (x \in \mathcal{X}, k \in \mathbb{N})$.
- クロック同士の比較：$\mu \models x \begin{matrix} \geqq \\ \leqq \end{matrix} y \quad (x, y \in \mathcal{X})$.

主定理：到達可能性問題の決定可能性

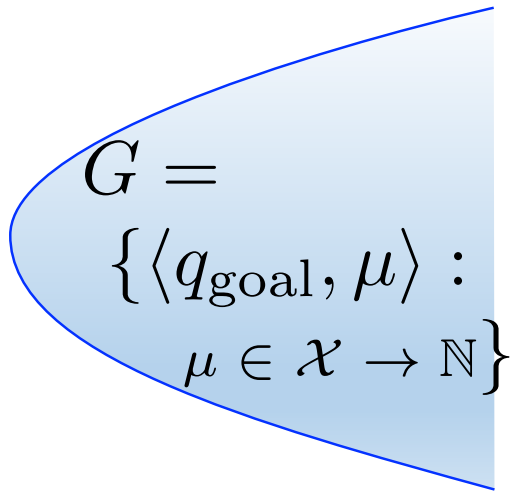
$\langle q_{\mathsf{start}}, \mathbf{0} \rangle$ から状態 q_{goal} に到達するかどうかは，決定可能.



ただし，$\mathbf{0} \triangleq \{x_1 \mapsto 0, \quad x_2 \mapsto 0, \quad \ldots, \quad x_n \mapsto 0\}$.

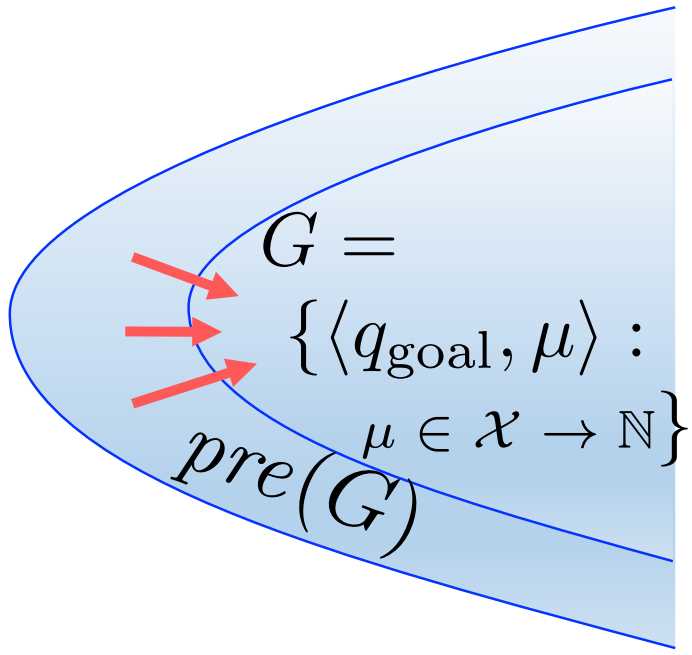
q_{goal} から $\langle q_{\mathsf{start}}, \mathbf{0} \rangle$ への後ろ向き到達可能性解析

1. $G = \{\langle q_{\mathsf{goal}}, \mu \rangle : \mu \in \mathcal{X} \to \mathbb{N}\}$.
2. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in G$ なら終了.



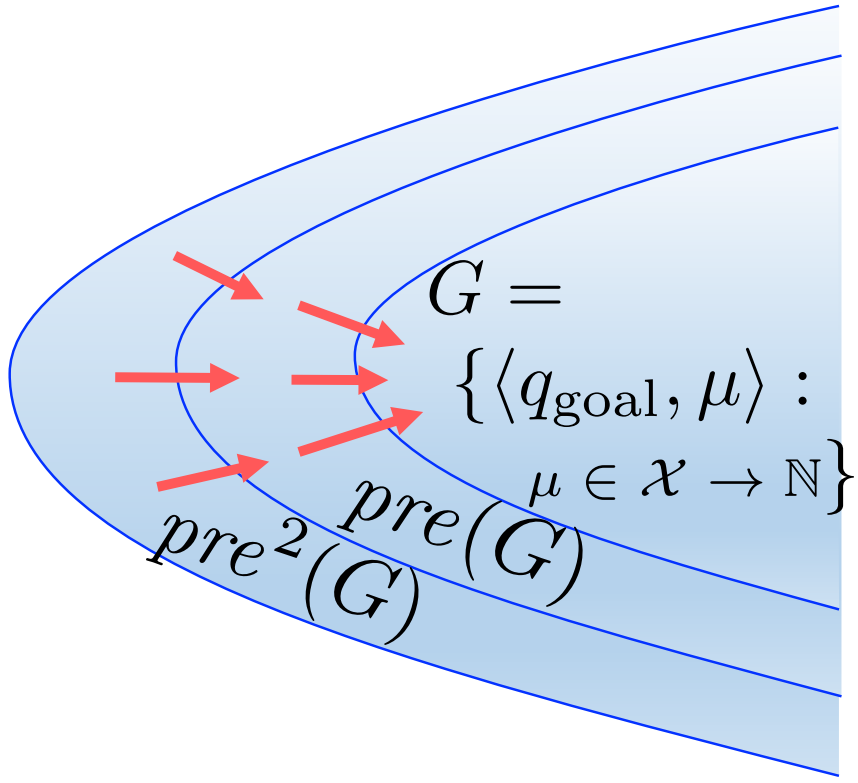
q_{goal} から $\langle q_{\mathsf{start}}, \mathbf{0} \rangle$ への後ろ向き到達可能性解析

1. $G = \{\langle q_{\mathsf{goal}}, \mu \rangle : \mu \in \mathcal{X} \to \mathbb{N}\}$.
2. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in G$ なら終了.
3. 入っていなければ，1 歩手前を計算：
   $pre(G) \triangleq \{c : \exists c' \in G.\ c \Rightarrow c'\}$.
4. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in pre(G)$ なら終了.



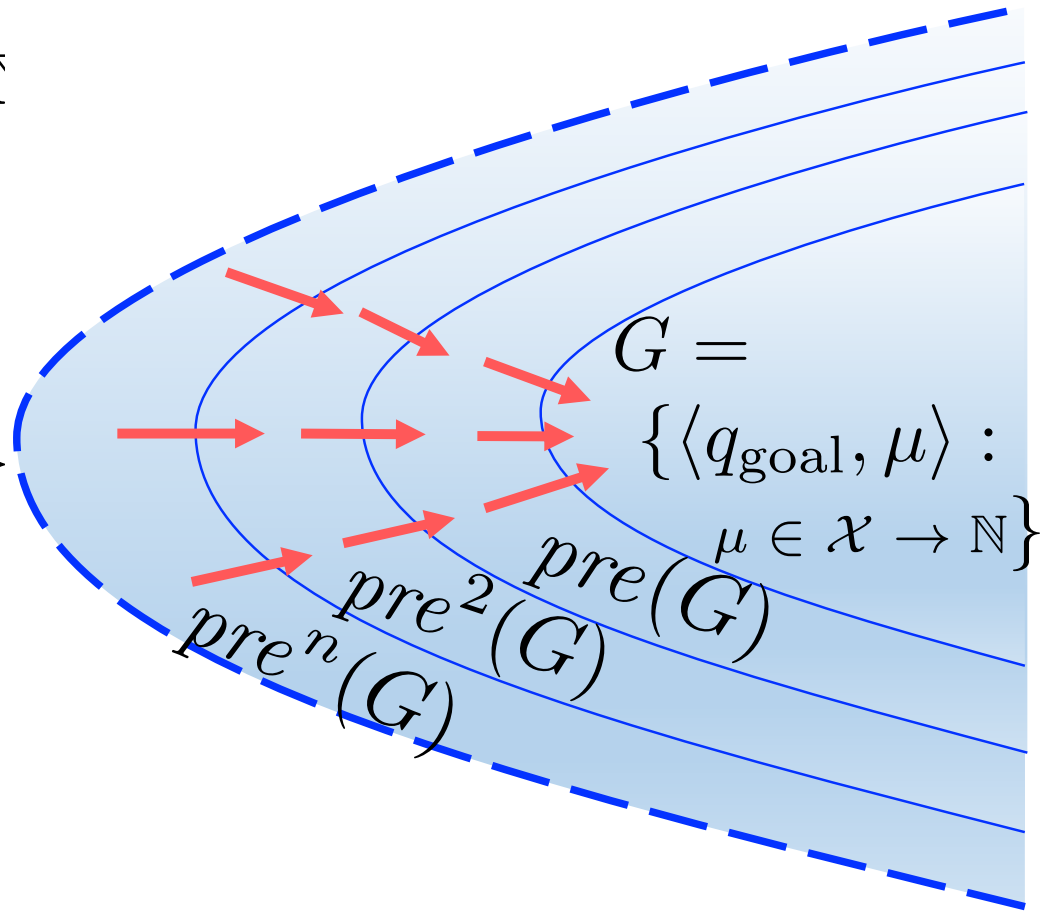
q_{goal} から $\langle q_{\mathsf{start}}, \mathbf{0} \rangle$ への後ろ向き到達可能性解析

1. $G = \{\langle q_{\mathsf{goal}}, \mu \rangle : \mu \in \mathcal{X} \to \mathbb{N}\}$.
2. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in G$ なら終了.
3. 入っていなければ，1 歩手前を計算：
   $pre(G) \triangleq \{c : \exists c' \in G.\ c \Rightarrow c'\}$.
4. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in pre(G)$ なら終了.
5. 入っていなければ，
   $pre^2(G)$ を計算して…



q_{goal} から $\langle q_{\mathsf{start}}, \mathbf{0} \rangle$ への後ろ向き到達可能性解析

1. $G = \{\langle q_{\mathsf{goal}}, \mu \rangle : \mu \in \mathcal{X} \to \mathbb{I}$
2. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in G$ なら終了.
3. 入っていなければ，1 歩手前を計算
   $pre(G) \triangleq \{c : \exists c' \in G.\ c \Rightarrow$
4. $\langle q_{\mathsf{start}}, \mathbf{0} \rangle \in pre(G)$ なら終了.
5. 入っていなければ，
   $pre^2(G)$ を計算して…



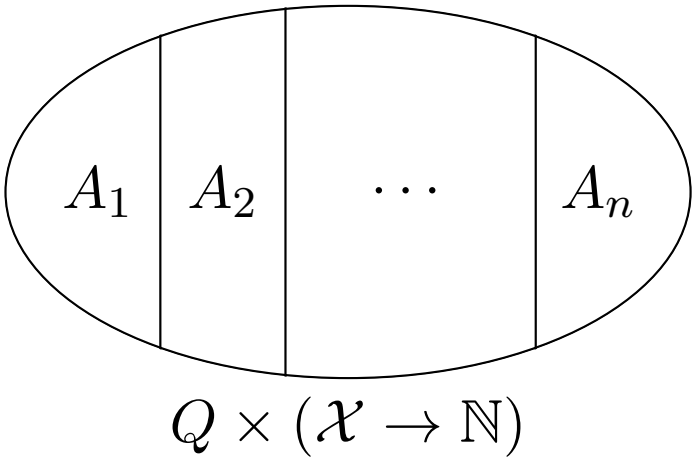
探索空間「$Q \times (\mathcal{X} \to \mathbb{N})$」が有限でないので

$$G \subsetneq (G \cup pre(G)) \subsetneq (G \cup pre(G) \cup pre^2(G)) \subsetneq \cdots$$

いつまでも拡大できる $\iff$ いつ打ち切れば良いのか分からない.

典型的戦略：有限の抽象化

計算状況全体を有限に抽象化



pre の正確な抽象化

$$\begin{array}{ccc} \text{Abs: } \langle p, A\rangle & \xrightarrow{\textsc{Pre}} & \langle q, A'\rangle \\ \llbracket\cdot\rrbracket \downarrow & \circlearrowleft & \llbracket\cdot\rrbracket \downarrow \\ \text{Conc: } \langle p, \mu\rangle & \xrightarrow[pre]{} & \langle q, \mu'\rangle \end{array}$$

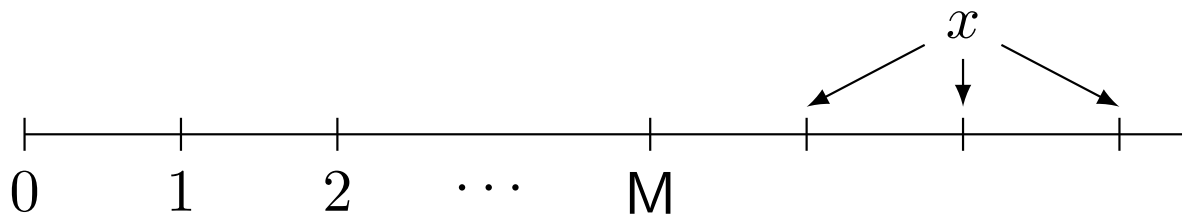
有限なのでいつかは停まる：

$$\begin{array}{rcl} \mathcal{A} & \subsetneq & \mathcal{A} \cup \textsc{Pre}(\mathcal{A}) \\ & \subsetneq & \mathcal{A} \cup \textsc{Pre}(\mathcal{A}) \cup \textsc{Pre}^2(\mathcal{A}) \\ & \subsetneq & \cdots \\ & \subsetneq & \mathcal{A} \cup \textsc{Pre}(\mathcal{A}) \cup \cdots \cup \textsc{Pre}^n(\mathcal{A}) \\ & = & \mathcal{A} \cup \textsc{Pre}(\mathcal{A}) \cup \cdots \cup \textsc{Pre}^n(\mathcal{A}) \cup \textsc{Pre}^{n+1}(\mathcal{A}). \end{array}$$

区別できないものをまとめる：PreShape

$\mathcal{S} = (Q, \mathcal{X}, \Delta)$ の制約 $x \gtreqless k$ で最大の k を M とする.

M を超えると区別不能：

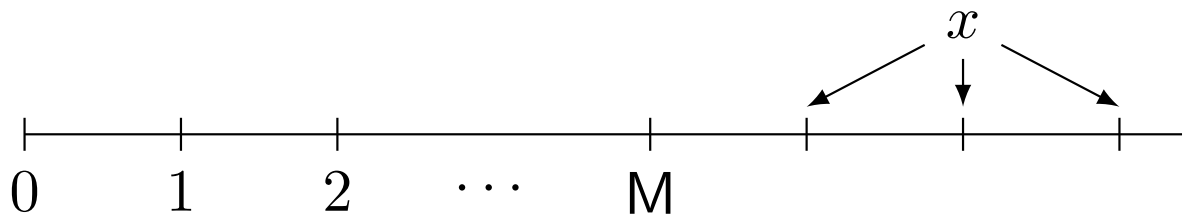


- $x >$ M という事実だけが重要.
- ただし，$x \gtreqless y$ のために順序だけは覚えておく.

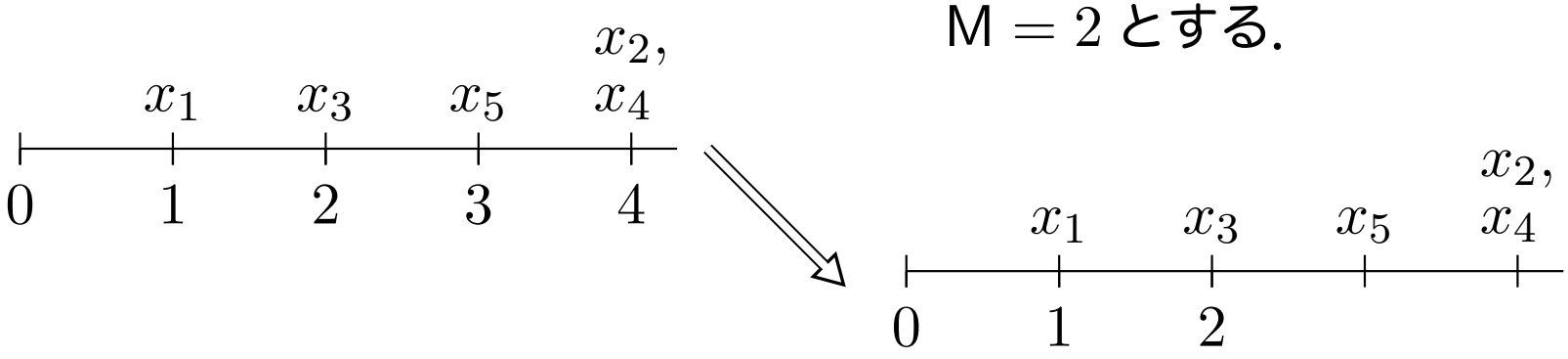
区別できないものをまとめる：PreShape

$\mathcal{S} = (Q, \mathcal{X}, \Delta)$ の制約 $x \gtreqless k$ で最大の k を M とする.

M を超えると区別不能：



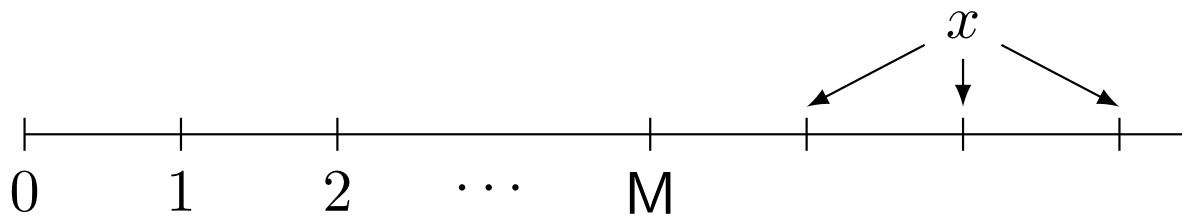
- $x >$ M という事実だけが重要.
- ただし，$x \gtreqless y$ のために順序だけは覚えておく.



区別できないものをまとめる：PreShape

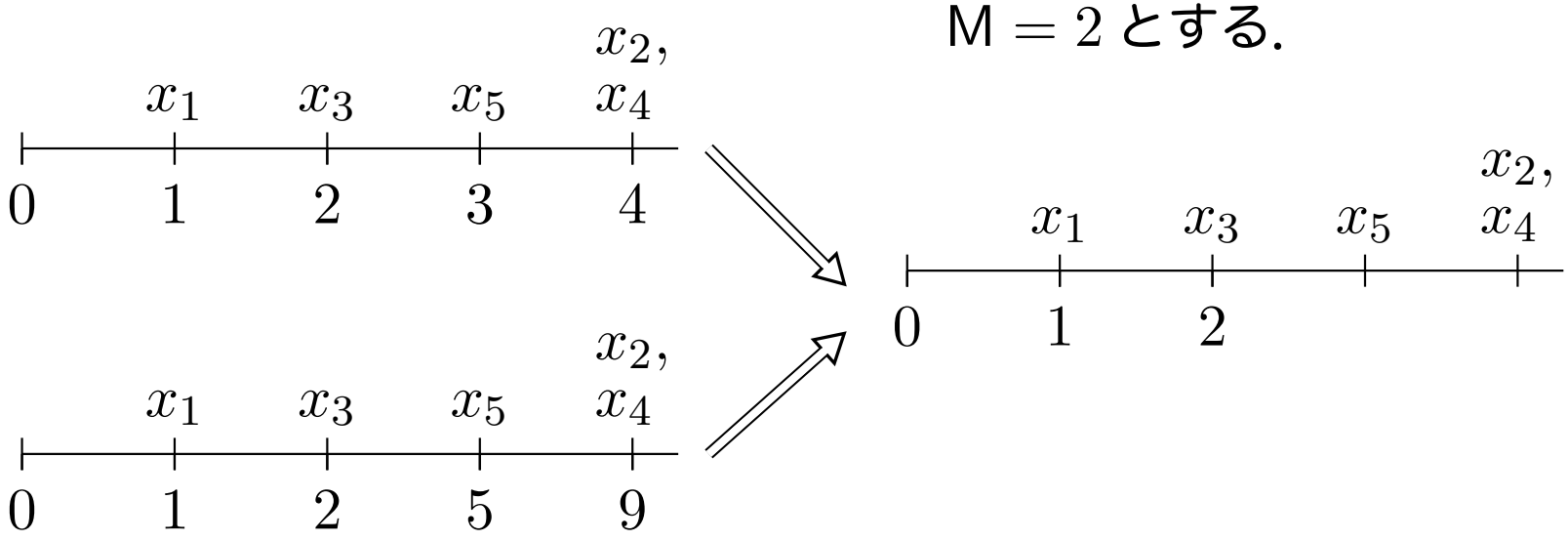
$\mathcal{S} = (Q, \mathcal{X}, \Delta)$ の制約 $x \gtreqless k$ で最大の k を M とする.

M を超えると区別不能：

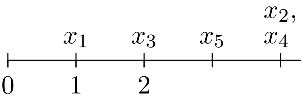


- $x >$ M という事実だけが重要.
- ただし，$x \gtreqless y$ のために順序だけは覚えておく.

M = 2 とする.



$$\llbracket$$



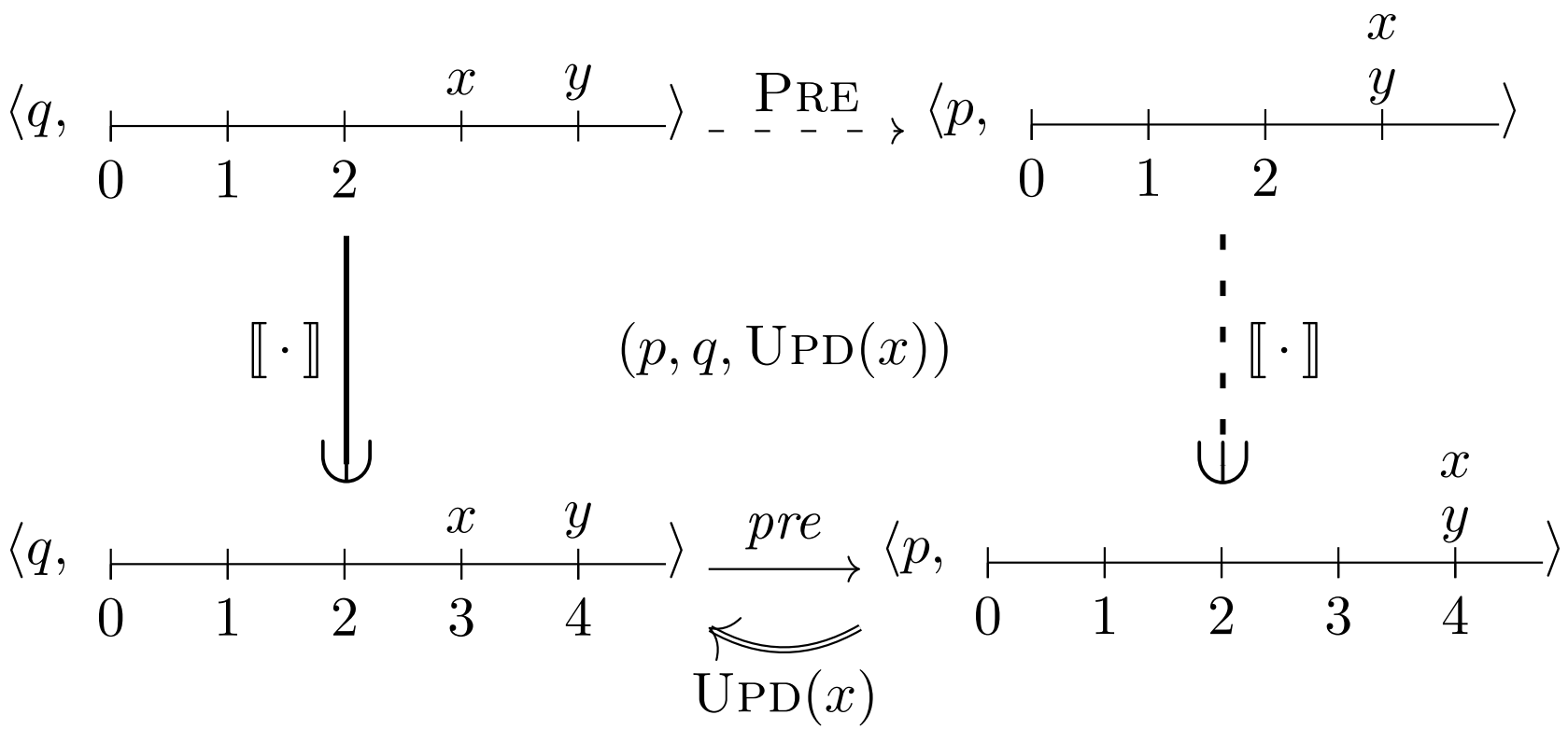
$$\rrbracket = \left\{ \mu : \begin{array}{l} \mu \models x_1 = 1 \land x_3 = 2 \land x_2 = x_4, \\ \mu \models 2 < x_5 < x_2 \end{array} \right\}.$$

状態 q と preshape P について, $\llbracket \langle q, P \rangle \rrbracket \triangleq \{ \langle q, \mu \rangle : \mu \in \llbracket P \rrbracket \}$.

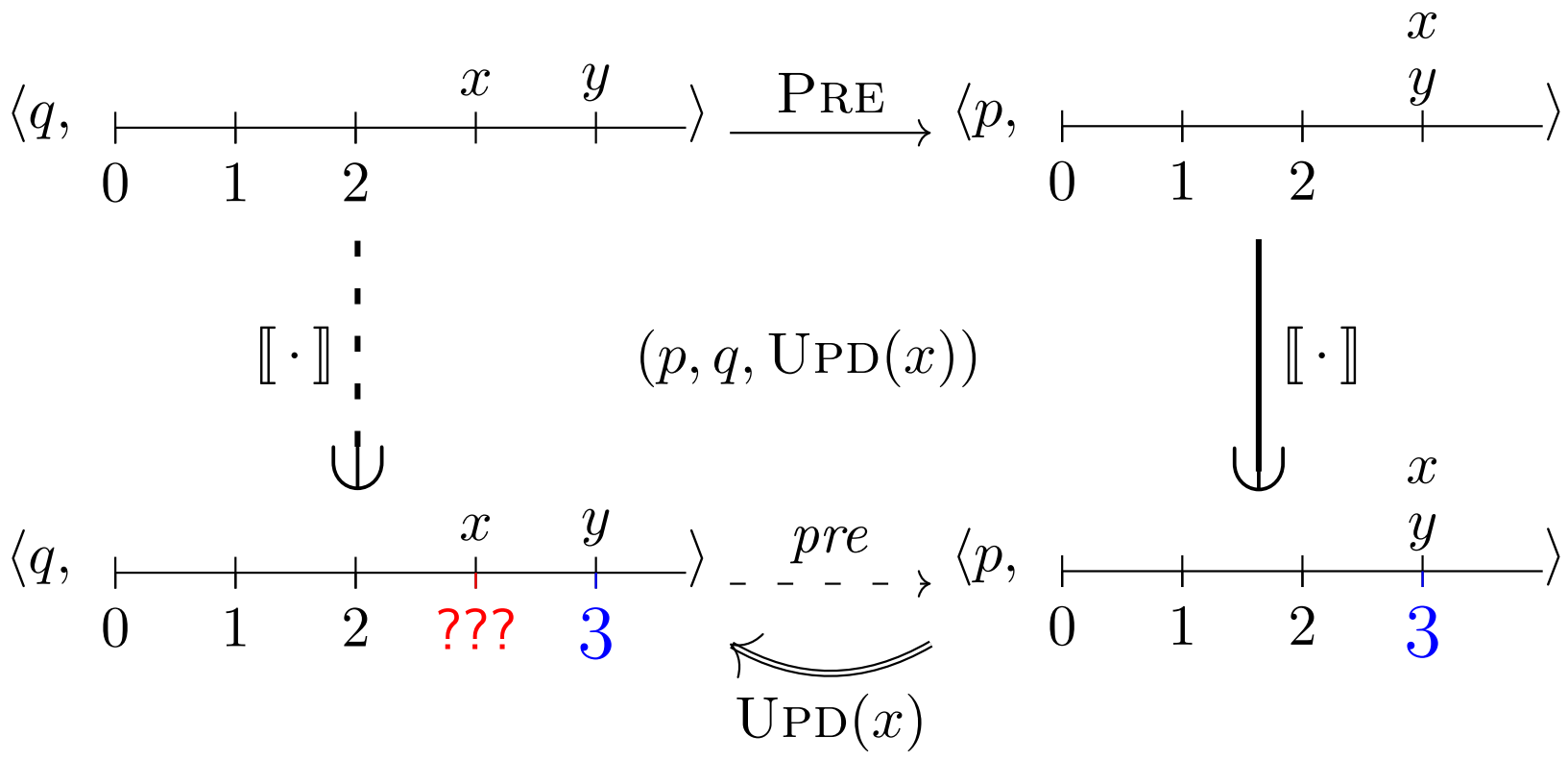
$$\left[\!\left[\begin{array}{c} \\ x_1 \quad x_3 \quad x_5 \quad x_2, x_4 \\ 0 \quad 1 \quad 2 \end{array} \right]\!\right] = \left\{ \mu : \begin{array}{l} \mu \models x_1 = 1 \wedge x_3 = 2 \wedge x_2 = x_4, \\ \mu \models 2 < x_5 < x_2 \end{array} \right\}.$$

状態 q と preshape P について，$[\![\langle q, P\rangle]\!] \triangleq \{\langle q, \mu\rangle : \mu \in [\![P]\!]\}$.

片側はうまくいく：$pre([\![\langle q, P\rangle]\!]) \subseteq [\![\textsc{Pre}(\langle q, P\rangle)]\!]$



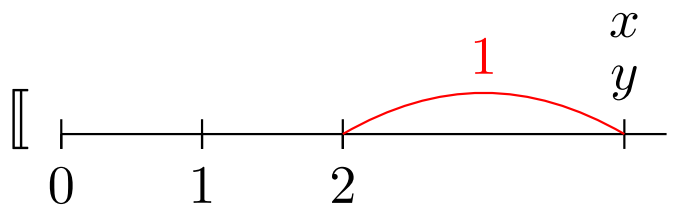
$pre(\llbracket\langle q, P\rangle\rrbracket) \supseteq \llbracket \textsc{Pre}(\langle q, P\rangle)\rrbracket$ ← 失敗



原因：Preshape では，y が 4 以上である，ことが表現できない．

Shape = PreShape + 距離情報

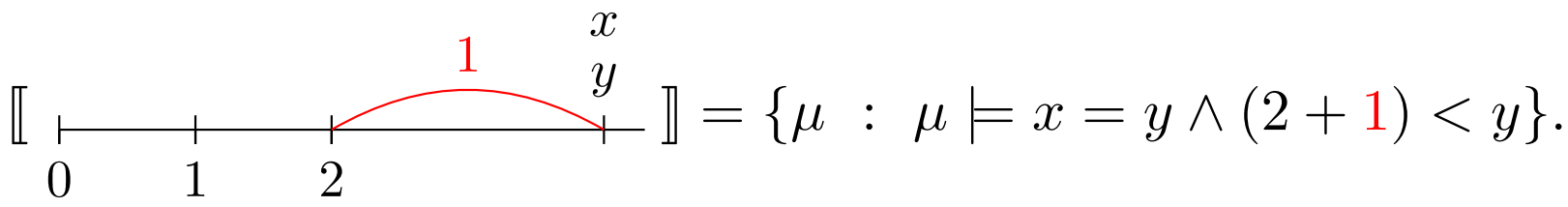
Preshape に，「距離情報」を加えることで，正確さを取り戻す．



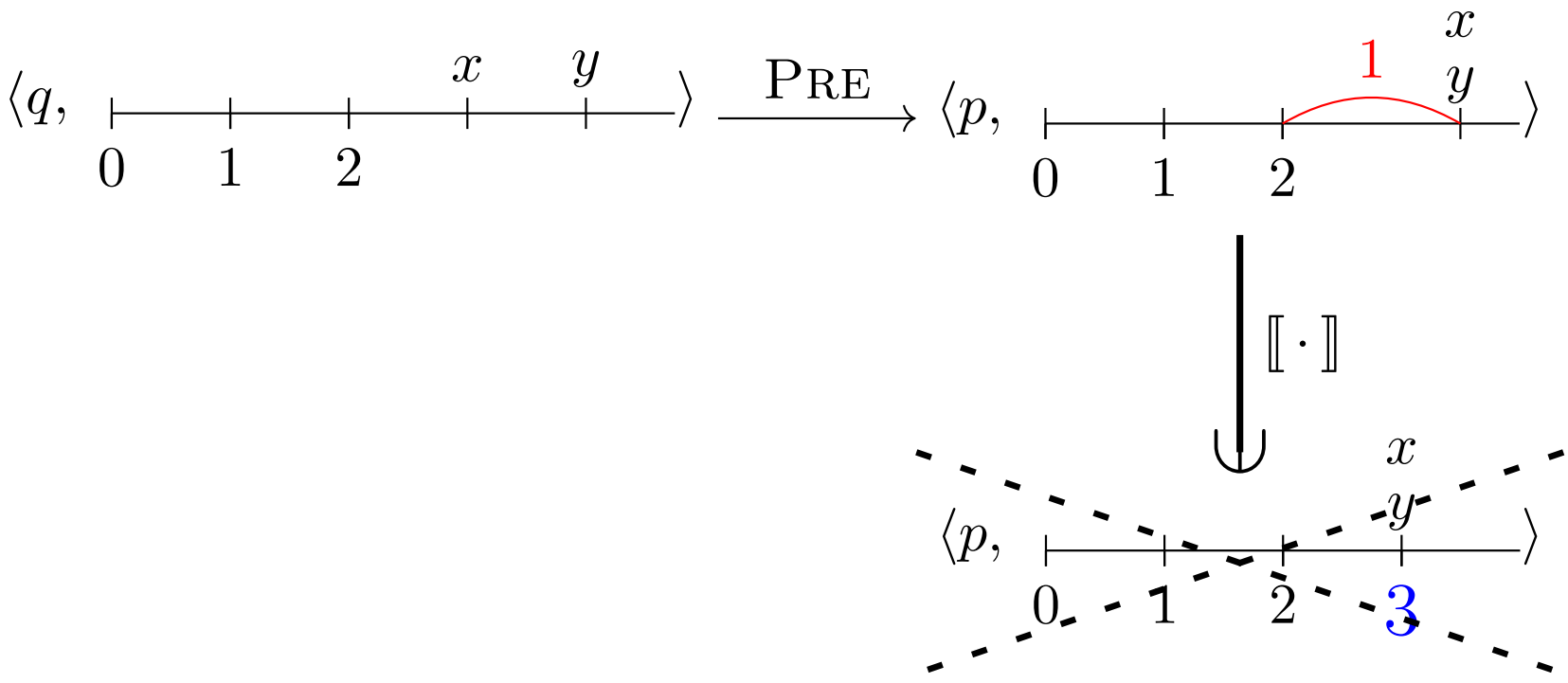
$$[\![\quad]\!] = \{\mu \ : \ \mu \models x = y \wedge (2 + 1) < y\}.$$

Shape = PreShape + 距離情報

Preshape に，「距離情報」を加えることで，正確さを取り戻す．

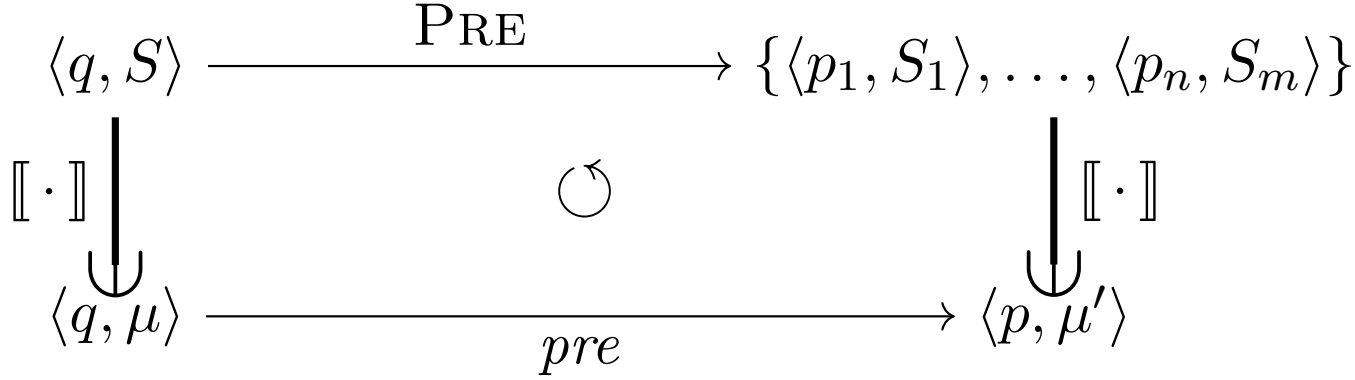


$$[\![\quad]\!] = \{\mu \ : \ \mu \models x = y \wedge (2+1) < y\}.$$

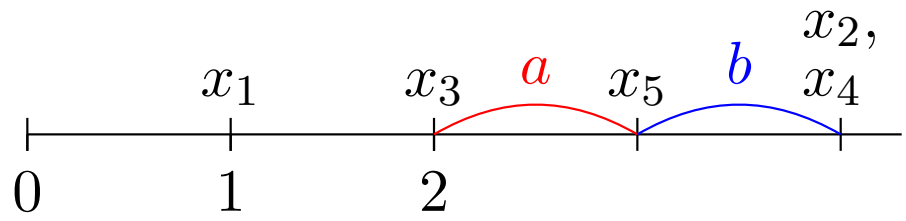


Shapeの性質（と状況の整理）

正確な表現：



無限の空間：

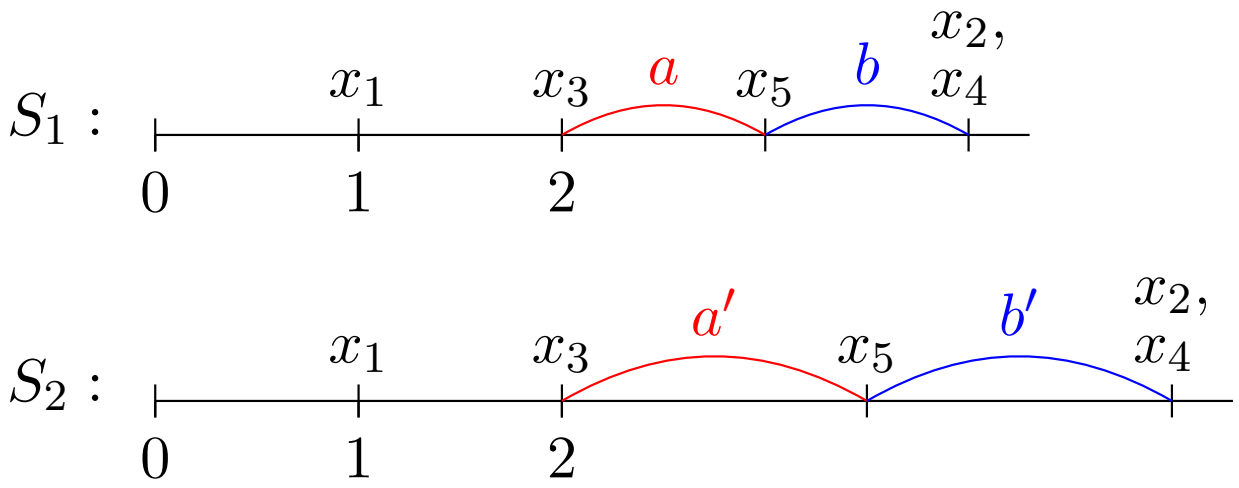


$a, b \in \mathbb{N}$.

構造	サイズ	pre	アルゴリズム
$Q \times (\mathcal{X} \to \mathbb{N})$	無限	正確な表現	✘
PreShape	有限	不正確な表現	✘
Shape	無限 WQO	正確な表現	✓

Shape 上の良い擬順序（Well Quasi Ordering）

順序の入れ方：同じ preshape を細分化した２つの shape について…



$$S_1 \preccurlyeq S_2 \overset{\text{def}}{\iff} a \leq a' \ \wedge \ b \leq b'.$$

「直積順序」から定まる順序 $\preccurlyeq$ について

- $\preccurlyeq$ は，Well Quasi Ordering（良い擬順序）をなす.
- $S_1 \preccurlyeq S_2$ ならば $[\![S_2]\!] \subseteq [\![S_1]\!]$.

Well-Quasi-Ordered Algorithm

少し賢い（？）$\textsc{Pre}$ 関数

function $\textsc{Expand}(\{C_1, C_2, \ldots, C_n\})$
  $\mathcal{R} := \{C_1, C_2, \ldots, C_n\}$
  for $i = 1$ to n **do**
    for each $D \in \textsc{Pre}(C_i)$ **do**
      if $\mathcal{R} \Subset \mathcal{R} \cup \{D\}$ **then**　　▷ D の追加は無駄でない
        $\mathcal{R} := \mathcal{R} \cup \{D\}$
  return $\mathcal{R}$

$\{C_1, \ldots, C_m\} \Subset \{C_1, \ldots, C_m, D\} \overset{\text{def}}{\iff} C_i \preccurlyeq D$ とする C_i がない.
$C_i \preccurlyeq D$ だとすると，$[\![D]\!] \subseteq [\![C_i]\!]$ から，D を加えるのは無駄.

$\preccurlyeq$ が w.q.o（良い擬順序）なので $\textsc{Expand}$ は必ず停止する：

$$\mathcal{C} \subsetneq \textsc{Expand}(\mathcal{C}) \subsetneq \textsc{Expand}^2(\mathcal{C}) \subsetneq \cdots \subsetneq \textsc{Expand}^n(\mathcal{C}) = \textsc{Expand}^{n+1}(\mathcal{C}).$$

Well-Quasi-Ordering

順序集合 $(X, \preceq)$ が well-quasi-ordering であるとは：

- 無限列 $x_1, x_2, \ldots, \overset{\preceq}{\frown} x_m, \ldots, x_n, \ldots$ に 2 点 $m < n$ が存在する.
- 無限に長い「無駄のない拡大」$Y_0 \Subset Y_1 \Subset \cdots$ が存在しない.
  $Y \Subset (Y \cup \{x\}) \overset{\text{def}}{\iff} y \preceq x$ とする $y \in Y$ が存在しない.

wqo の代表例：$(\mathbb{N}, \leq)$，wqo 上の直積順序 $(\mathbb{N}^d, \leq^d)$.

Well-Quasi-Ordering

順序集合 $(X, \preceq)$ が well-quasi-ordering であるとは：

- 無限列 $x_1, x_2, \ldots, \overset{\preceq}{x_m, \ldots, x_n}, \ldots$ に 2 点 $m < n$ が存在する.
- 無限に長い「無駄のない拡大」$Y_0 \Subset Y_1 \Subset \cdots$ が存在しない.
  $Y \Subset (Y \cup \{x\}) \overset{\text{def}}{\iff} y \preceq x$ とする $y \in Y$ が存在しない.

wqo の代表例：$(\mathbb{N}, \leq)$，wqo 上の直積順序 $(\mathbb{N}^d, \leq^d)$.

Shape 上の順序 $S_1 \preccurlyeq S_2$ は，自然数上の直積順序で定義した.

定理

Shape 上の順序 $\preccurlyeq$ は wqo である.
（証明には，任意の次元 d で，$(\mathbb{N}^d, \leq^d)$ が wqo であることを使う.）

系

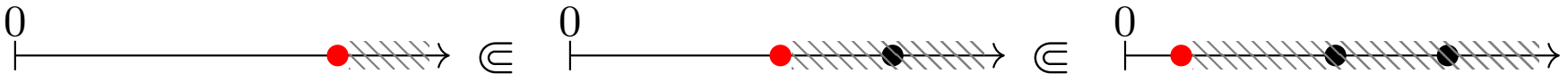
$\mathcal{C} \subsetneq \textsc{Expand}(\mathcal{C}) \subsetneq \textsc{Expand}^2(\mathcal{C}) \subsetneq \cdots \subsetneq \textsc{Expand}^n(\mathcal{C}) = \textsc{Expand}^{n+1}(\mathcal{C})$.

$\mathbb{N}^d$ 上の直積順序が wqo

命題

$(\mathbb{N}, \leq)$ は wqo である.

無駄のない拡大 ($Y \Subset Y \cup \{x\} \overset{\text{def}}{\iff} y \in Y$ に $y \leq x$ がない）は停まる.

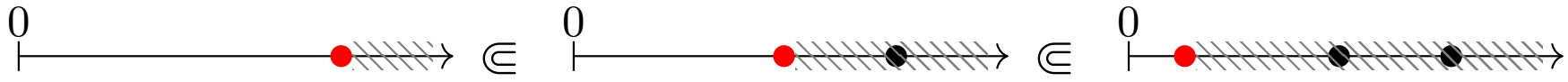


$\mathbb{N}^d$ 上の直積順序が wqo

命題

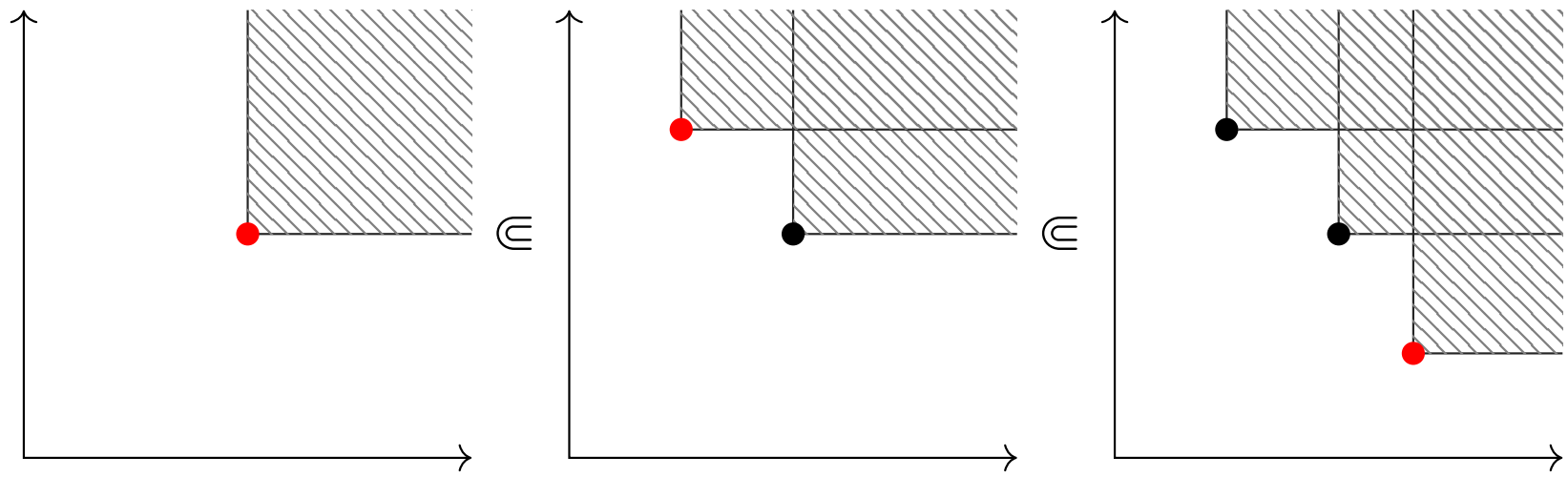
$(\mathbb{N}, \leq)$ は wqo である.

無駄のない拡大 ($Y \Subset Y \cup \{x\} \overset{\text{def}}{\iff} y \in Y$ に $y \leq x$ がない) は停まる.



命題

$(\mathbb{N}^2, \leq^2)$ は wqo である.

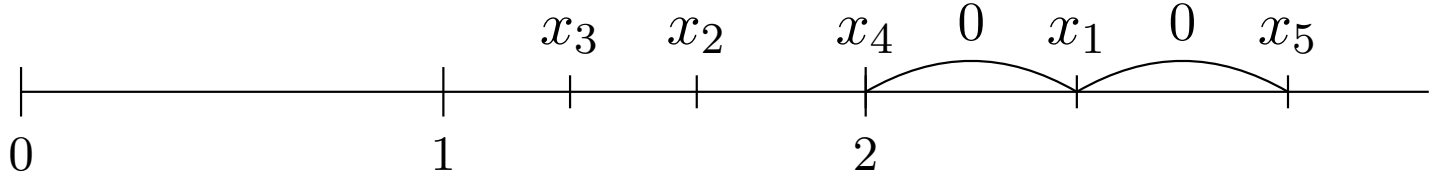


Shape + Region abstraction

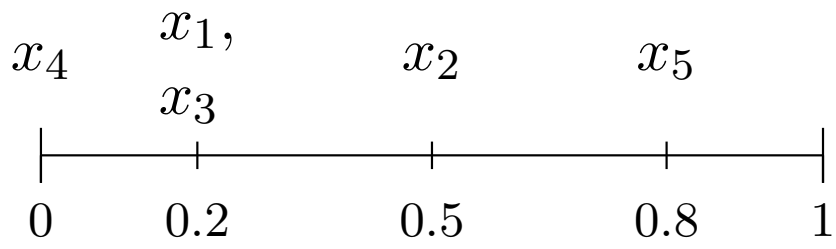
$$\nu = \{x_1 \mapsto 2.2, x_2 \mapsto 1.5, x_3 \mapsto 1.2, x_4 \mapsto 2.0, x_5 \mapsto 2.8\}$$

全ての割当は，shape S と region R の組 (S, R) に抽象化される.

S =



R =



$= \{x_4\}_0\{x_1, x_3\}\{x_2\}\{x_5\}.$

離散時間遷移系と同様の性質が (S, R) 上で成立する.

定理

更新可能時間オートマトンの到達可能性問題は決定可能.

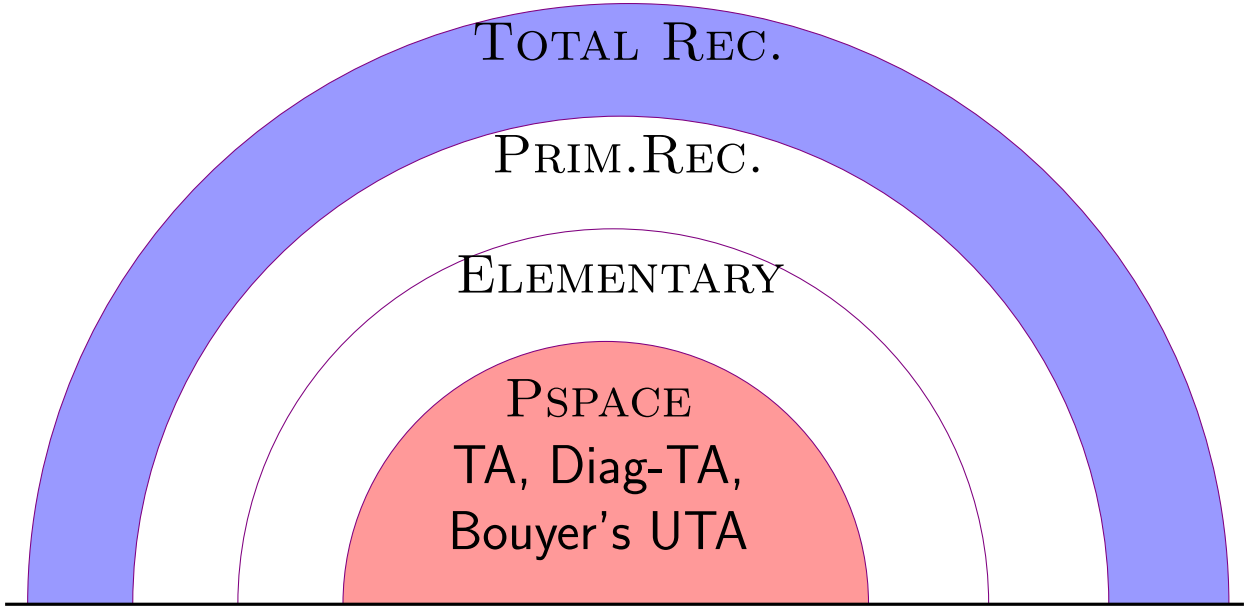
関連体系と到達可能性問題の計算量

Lossy Counter Machine [Mayr '03, Schnoebelen '10]

- LCM の到達可能性問題の決定可能性は，WQO を元にした議論で示される.
- LCM の到達可能性問題は Non-Primitive Recursive [Schnoebelen '10].

LCM の到達可能性問題 $\xrightarrow{\text{PTIME}}$ 離散時間遷移系の到達可能性問題

$\xrightarrow{\text{PTIME}}$ 更新可能時間オートマトンの到達可能性問題.



まとめ

- 時間オートマトンの新たな拡張を与えた.
  1. 表現力を拡大した；
  2. 到達可能性問題のための Shape 抽象を導入した；
  3. Shape 上の WQO が停止性証明の鍵となった.

今後の課題

- 言語 L_{ab} は，結局クロックというよりカウンター的な例．時間オートマトン「らしい」例を見つけたい.
- 到達可能性問題の「上界」は考えなかったので，これに取り組みたい.
- 更なる拡張はあり得るか？ Timed Pushdown Automaton などの最近の拡張を，更に推し進められないか？