外国語論文講読

"Communicating and Mobile Systems: the π-Calculus" by Robin Milner

Chap. 3 Sequential Processes and Bisimulation

岡本 吉央

(yoshio@klee.c.u-tokyo.ac.jp)

2000 年 11 月 15 日 (水)

この章の概要

- Nondeterministic sequential processes を定義する
  (labelled transition systems, sequential precess expressions)
- それらの間の等価性を考える
  (bisimulation, structural congruence)
- この章では concurrency は扱わない
  (以後の章で，concurrent processes について同様な議論をする)

3.1 Labelled transition systems

いくつか記法

$\mathcal{N}$	:	name の集合 (無限集合)
$\overline{\mathcal{N}} = \{\overline{a} : a \in \mathcal{N}\}$	:	co-name の集合 ($\mathcal{N} \cap \overline{\mathcal{N}} = \emptyset$ とする)
$\mathcal{L} = \mathcal{N} \cup \overline{\mathcal{N}}$	:	label の集合
$Act = \mathcal{L}$	:	action の集合

Name と Co-name

name a ⟵ 補完的な action ⟶ co-name $\overline{a}$

例

2 章の vending machine

customer の action 'tea' ⟷ vending machine の action '$\overline{\text{tea}}$'

Labelled transition systems

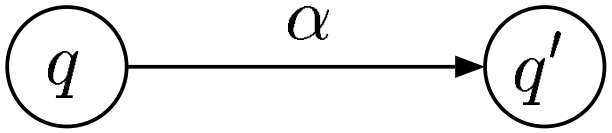
Def. 3.1 (labelled transition system, LTS)

Act 上の labelled transition system とは 2 つ組 $(\mathcal{Q}, \mathcal{T})$ で

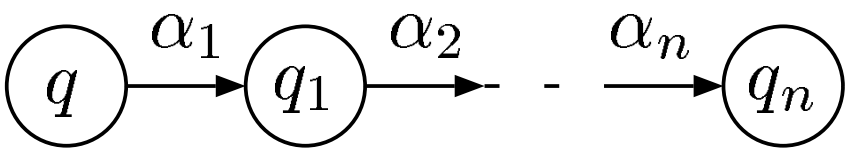
集合 $\mathcal{Q}$	:	state の集合
三項関係 $\mathcal{T} \subseteq (\mathcal{Q} \times Act \times \mathcal{Q})$	:	transition relation

矢印による記法

$(q, \alpha, q') \in \mathcal{T}$ のとき $q \xrightarrow{\alpha} q'$ とかく



source target



derivative

LTS とオートマトン

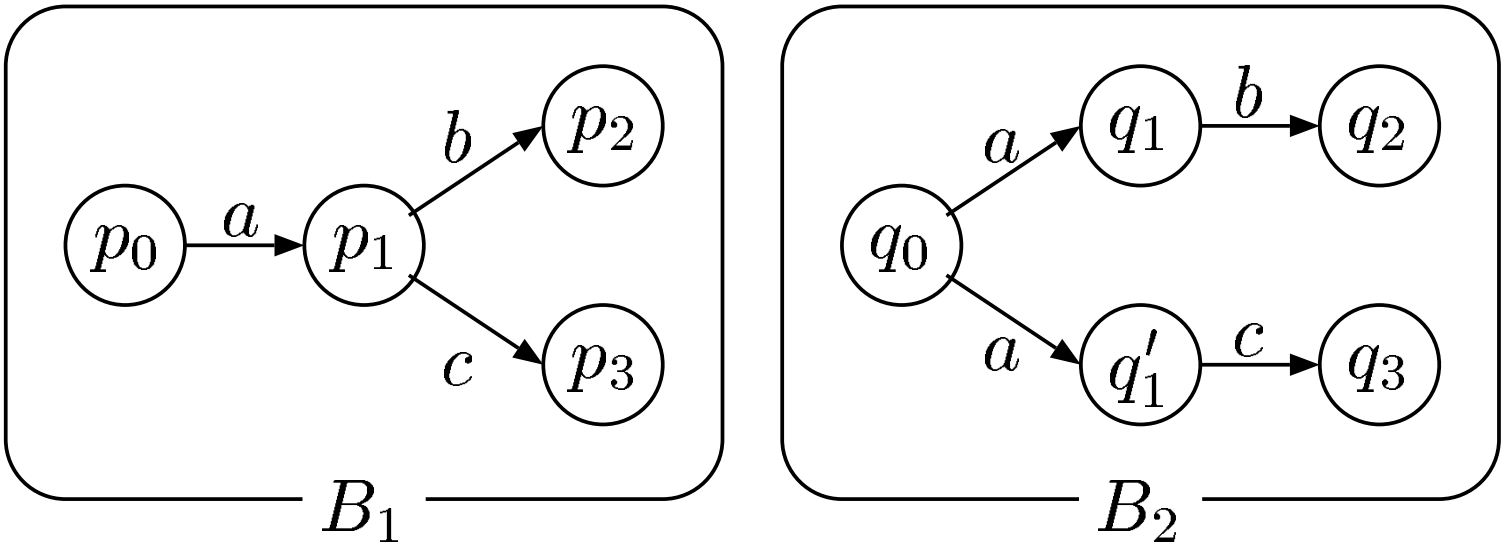
LTS は「初期状態と終了状態のないオートマトン」と見なせる

- 終了状態
  ⇒ すべての状態を終了状態と思う
- 初期状態
  ⇒ 任意の状態を初期状態とすると，異なるオートマトンが得られる

3.2 Strong simulation

LTS の 2 つの状態が等価とは?

例 3.2



前章の議論に従うと，B_1 と B_2 は等価ではない

‘simulation’ の考え方

例 3.2 の場合

「p_0 は q_0 を simulate(模倣) できる」

意味

p_0 の行動パターンの豊かさ $\geq$ q_0 の行動パターンの豊かさ

形式的に....

$\Rightarrow$ strong simulation

Strong simulation

Def. 3.3 (strong simulation)

$(\mathcal{Q}, \mathcal{T})$: labelled transition system

$\mathcal{S} \subseteq \mathcal{Q} \times \mathcal{Q}$: $\mathcal{Q}$ 上の二項関係

- $\mathcal{S}$ が $(\mathcal{Q}, \mathcal{T})$ 上の strong simulation $\overset{\text{def}}{\Longleftrightarrow}$

  if $(p, q) \in \mathcal{S}$ and $p \overset{\alpha}{\to} p'$,

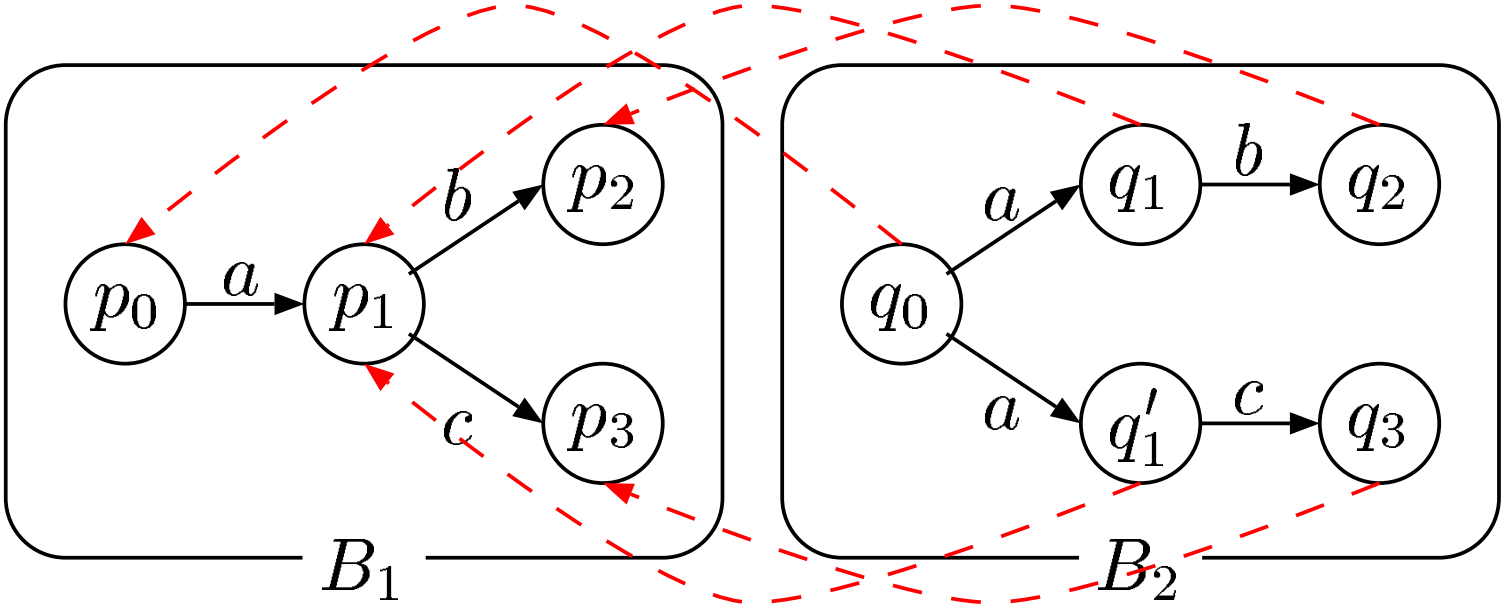
  then $\exists q' \in \mathcal{Q}$ s.t. $q \overset{\alpha}{\to} q'$ and $(p', q') \in \mathcal{S}$

- q が p を strongly simulate する $\overset{\text{def}}{\Longleftrightarrow}$

  $\exists$ strong simulation $\mathcal{S}$ s.t. $(p, q) \in \mathcal{S}$

strong simulation の例

Ex. 3.4 Ex. 3.2 の B_1 , B_2 について

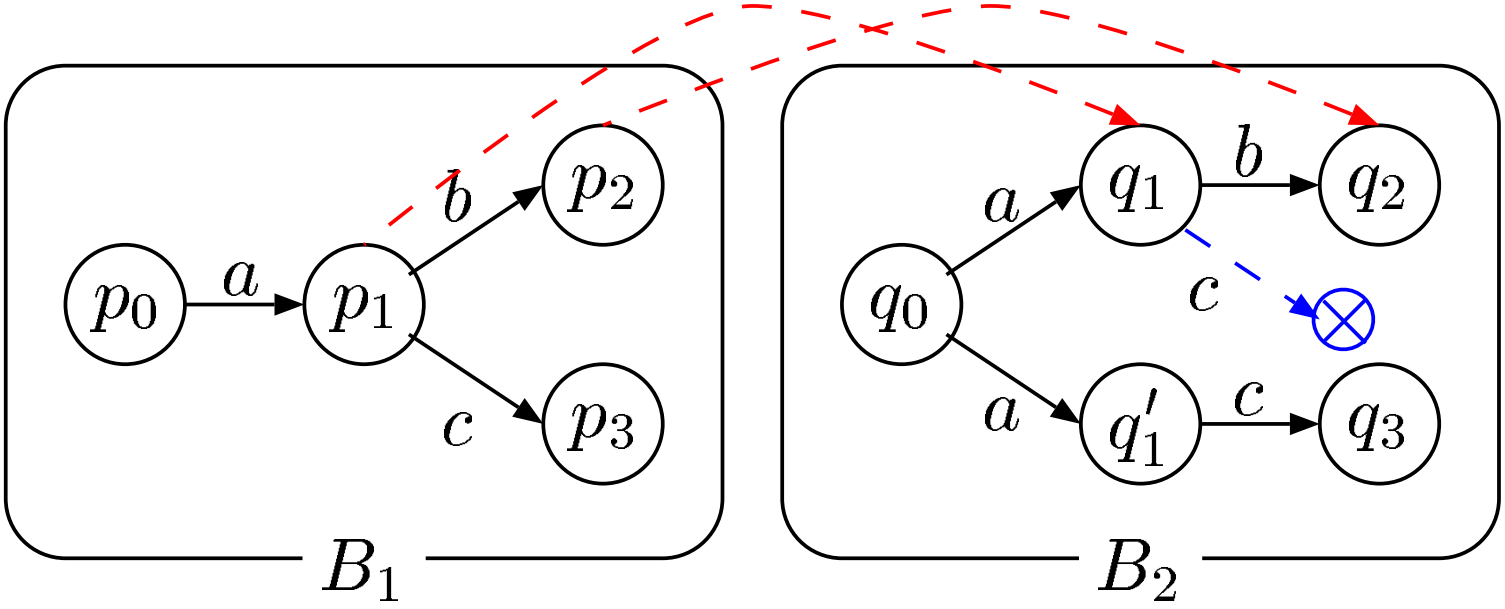


$\Rightarrow$ p_0 strongly simulates q_0.

strong simulation ではない例

Ex. 3.5 Ex. 3.2 の B_1 , B_2 について

(p_1, q_1) を含むような strong simulation は存在しない .



3.3 Strong bisimulation

strong bisimulation

逆二項関係 $\mathcal{R}^{-1}$: $(x, y) \in \mathcal{R} \Leftrightarrow (y, x) \in \mathcal{R}^{-1}$

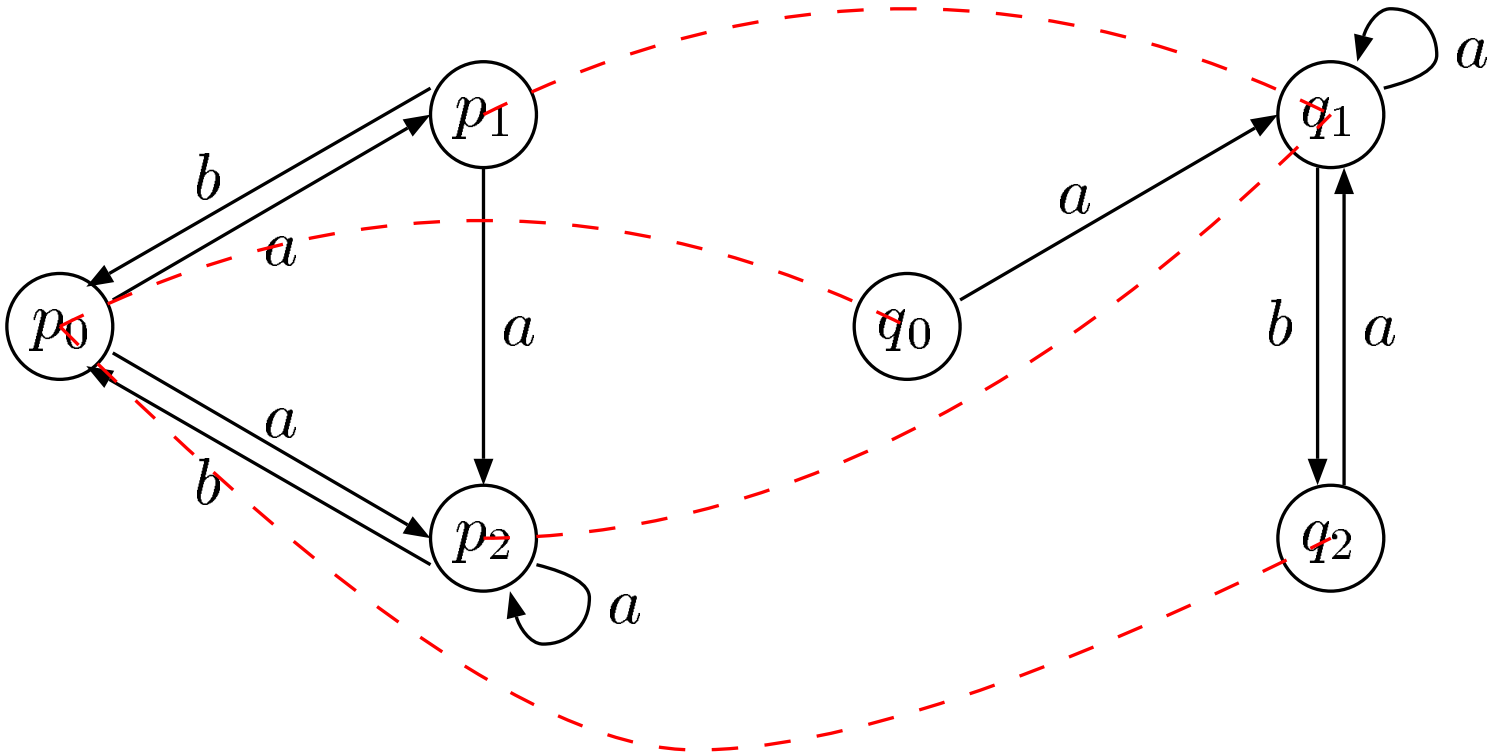
Def. 3.6 (strong bisimulation, strong equivalence)

- $\mathcal{Q}$ 上の二項関係 $\mathcal{S}$ が LTS$(\mathcal{Q}, \mathcal{T})$ 上の strong bisimulation $\overset{\text{def}}{\Longleftrightarrow}$
  $\mathcal{S}$ かつ $\mathcal{S}^{-1}$ が strong simulation
- p と q が strongly bisimilar (strongly equivalent) $\overset{\text{def}}{\Longleftrightarrow}$
  $\exists$ strong bisimulation $\mathcal{S}$ s.t. $(p, q) \in \mathcal{S}$

記法 $p \sim q$: p と q が strongly bisimilar (strongly equivalent)

strong bisimulation の例

Ex. 3.7 次の LTS において，$p_0 \sim q_0$



bisimilarity の性質

Prop. 3.9

(1) $\sim$ は同値関係である，すなわち，

- $p \sim p$ (反射性)
- $p \sim q \Longrightarrow q \sim p$ (対称性)
- $p \sim q$ and $q \sim r \Longrightarrow p \sim r$ (推移性)

(2) $\sim$ 自体も strong bisimulation である

Prop. 3.9.(1) の証明

反射性

1. $\mathcal{Q}$ 上の恒等関係 $\{(p,p) : p \in \mathcal{Q}\}$ を考える
2. 恒等関係の逆関係も恒等関係である
3. ゆえに，恒等関係が strong simulation であることを示せばよい
4. strong simulation の定義 (Def. 3.3) において，$q' = p'$ とすればよい

対称性

1. 関係 $\mathcal{S}$ が bisimulation のときに，$\mathcal{S}^{-1}$ も bisimulation であることを示せばよい

2. $(\mathcal{S}^{-1})^{-1} = \mathcal{S}$ と strong bisimulation の定義 (Def. 3.6) よりわかる

推移性

1. 示せばよいことは，$\mathcal{S}_1, \mathcal{S}_2$ が bisimulation であるときに，

$$\mathcal{S}_1\mathcal{S}_2 = \{(p, r) : (p, q) \in \mathcal{S}_1, (q, r) \in \mathcal{S}_2 \text{ for some } q\}$$

   も bisimulation であること

2. まず，$(\mathcal{S}_1\mathcal{S}_2)^{-1} = \mathcal{S}_2^{-1}\mathcal{S}_1^{-1}$ であることを示す

$$\begin{aligned}(\mathcal{S}_1\mathcal{S}_2)^{-1} &= \{(r, p) : (p, q) \in \mathcal{S}_1, (q, r) \in \mathcal{S}_2 \text{ for some } q\} \\ &= \{(r, p) : (q, p) \in \mathcal{S}_1^{-1}, (r, q) \in \mathcal{S}_2^{-1} \text{ for some } q\} \\ &= \mathcal{S}_2^{-1}\mathcal{S}_1^{-1}\end{aligned}$$

3. ゆえに，$\mathcal{S}_1\mathcal{S}_2$ が simulation であることを示せば十分

4. ここで，次の 2 つを仮定する：

   仮定 **A.** $(p, r) \in \mathcal{S}_1\mathcal{S}_2$

   仮定 **B.** $p \xrightarrow{\alpha} p'$

5. この仮定のもとで示すことは，ある r' が存在して，

   帰結 **A.** $r \xrightarrow{\alpha} r'$ かつ

   帰結 **B.** $(r, r') \in \mathcal{S}_1\mathcal{S}_2$

6. 仮定 A. より，ある q が存在して

   (a) $(p, q) \in \mathcal{S}_1$

   (b) $(q, r) \in \mathcal{S}_2$

7. $\mathcal{S}_1$ は simulation なので，6.(a) と 仮定 B. より，ある q' が存在して

   (a) $q \xrightarrow{\alpha} q'$ かつ

   (b) $(p', q') \in \mathcal{S}_1$

8. $\mathcal{S}_2$ は simulation なので，7.(a) と 6.(b) より，ある r' が存在して

   (a) $r \xrightarrow{\alpha} r'$ かつ

   (b) $(q', r') \in \mathcal{S}_2$

9. 8.(a) は，まさに帰結 A. である

10. 7.(b)，8.(b) および $\mathcal{S}_1\mathcal{S}_2$ の定義より，$(p', r') \in \mathcal{S}_1\mathcal{S}_2$
    これは，まさに帰結 B. である

Prop. 3.9.(2) の証明

1. 次を仮定する

   仮定 **A.** $p \sim q$

   仮定 **B.** $p \stackrel{\alpha}{\to} p'$

2. このときに示すことは，ある q' が存在して

   帰結 **A.** $p' \sim q'$

   帰結 **B.** $q \stackrel{\alpha}{\to} q'$

3. 仮定 A. より，ある bisimulation $\mathcal{S}$ に対して，$(p, q) \in \mathcal{S}$

4. 3. と仮定 B. より，ある q' が存在して，

   (a) $(p', q') \in \mathcal{S}$

(b) $q \stackrel{\alpha}{\to} q'$

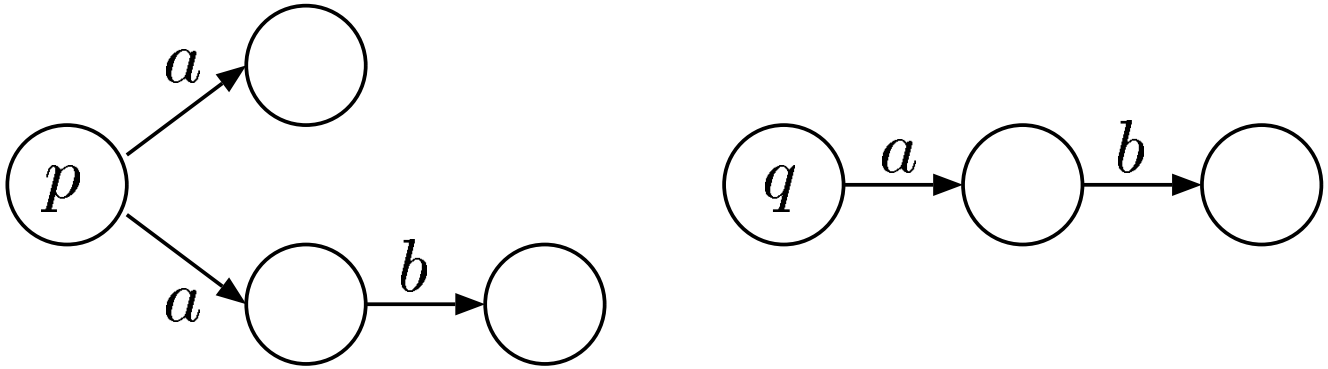
5. 4.(b) は，まさに帰結 B. である

6. 4.(a) より，$p' \sim q'$ であり，これはまさに帰結 A. である

補足

$p \sim q \Longleftrightarrow p$ strongly simulates q and q strongly simulates p ???

Ex. 3.10 次の 2 つの LTS を考える



この場合において，

- p strongly simulates q and q strongly simulates p
- $p \not\sim q$

3.4 Sequential process expressions

プロセス

システムの各状態を表すためにプロセス表現を用いる

記法

Id	:	プロセス (プロセス ID) の集合 (無限集合)
$A\langle a, b, c\rangle$	:	name a, b, c をパラメータとするプロセス A
$\vec{a}$	:	name の列

記法の続き

$\vec{a}, \vec{b}$ が同じ長さの異なる name-sequence，P がプロセスのとき，

$\{\vec{b}/\vec{a}\}P$: P の中の $\vec{a}$ を $\vec{b}$ で置換したプロセス

$\mathrm{fn}(P)$: P に表れる name (free name) の集合

Sequential process expression

Def. 3.11 (sequential process expression)

$$P ::= A\langle a_1, \ldots, a_n \rangle \mid \sum_{i \in I} \alpha_i.P_i$$

ここで，I は有限添字集合，P_i は process expression
(おそらく，α_i は name)

$\mathcal{P}^{\mathrm{seq}}$: sequential process expression の全体

S.p.e. の補足説明

- $I = \{1, 2, 3\}$ のとき

$$\sum_{i \in I} \alpha_i.P_i = \alpha_1.P_1 + \alpha_2.P_2 + \alpha_3.P_3$$

(順番は重要ではない)

- $I = \emptyset$ のとき

$$\sum_{i \in I} \alpha_i.P_i = 0$$

Defining equation

仮定 すべてのプロセスは defining equation を持つ

$$A(\vec{a}) \stackrel{\text{def}}{=} P_A$$

ここで，P_A は和であり，$\vec{a}$ は $\mathrm{fn}(P_A)$ を含むとする

(例は後程)

Structural congruence

気持ち

$A(\vec{a}) \stackrel{\text{def}}{=} P_A$ のとき，$A\langle\vec{b}\rangle$ と $\{\vec{b}/\vec{a}\}P_A$ の意味が同じであってほしい

Def. 3.12 (Structural congruence)

2 つの s.p.e. P, Q が structurally congruent (記法： $P \equiv Q$)

$\Updownarrow$ def

we can transorm one into the other by replacing occurrences of $A\langle\vec{b}\rangle$ by $\{\vec{b}/\vec{a}\}P_A$, or vice versa, for arbitrary A defined by $A(\vec{a}) \stackrel{\text{def}}{=} P_A$

Structural congruence の例

例 次の 2 つのプロセス定義を考える

$$A(a,b) \stackrel{\text{def}}{=} a.A\langle a,b\rangle + b.B\langle a,a\rangle$$

$$B(c,d) \stackrel{\text{def}}{=} c.d.0$$

このとき，

$$B\langle a,a\rangle \equiv a.a.0$$

$$A\langle a,b\rangle \equiv a.(a.A\langle a,b\rangle + b.B\langle a,a\rangle) + b.a.a.0$$

Sequential process の LTS

Def. 3.13 (The LTS of sequential processes)

LTS of sequential processes over $Act \overset{\text{def}}{\Longleftrightarrow}$

状態： $\mathcal{P}^{\text{seq}}$

遷移： $P \equiv \sum_{i \in I} \alpha_i.P_i \Longrightarrow P \xrightarrow{\alpha_i} P_j$ for each $j \in I$

この章の残りでは... LTS of sequential process の例を見る

- Boolean buffer
- Scheduler
- Counter

3.5 Boolean buffer

Boolean buffer とは

容量が 2 の buffer で，入力は $\{0, 1\}$

$Act = \{in_0, in_1, \overline{out_0}, \overline{out_1}\}$

in_0	:	0 を入れる	$\overline{out_0}$	:	0 を取り出す
in_1	:	1 を入れる	$\overline{out_1}$	:	1 を取り出す

buffer にある列 $s \in \{0, 1, 00, 01, 10, 11\}$ に対して，対応する状態 $Buff_s^{(2)}$ を考える

初期状態は何も入ってない状態 $Buff^{(2)}$

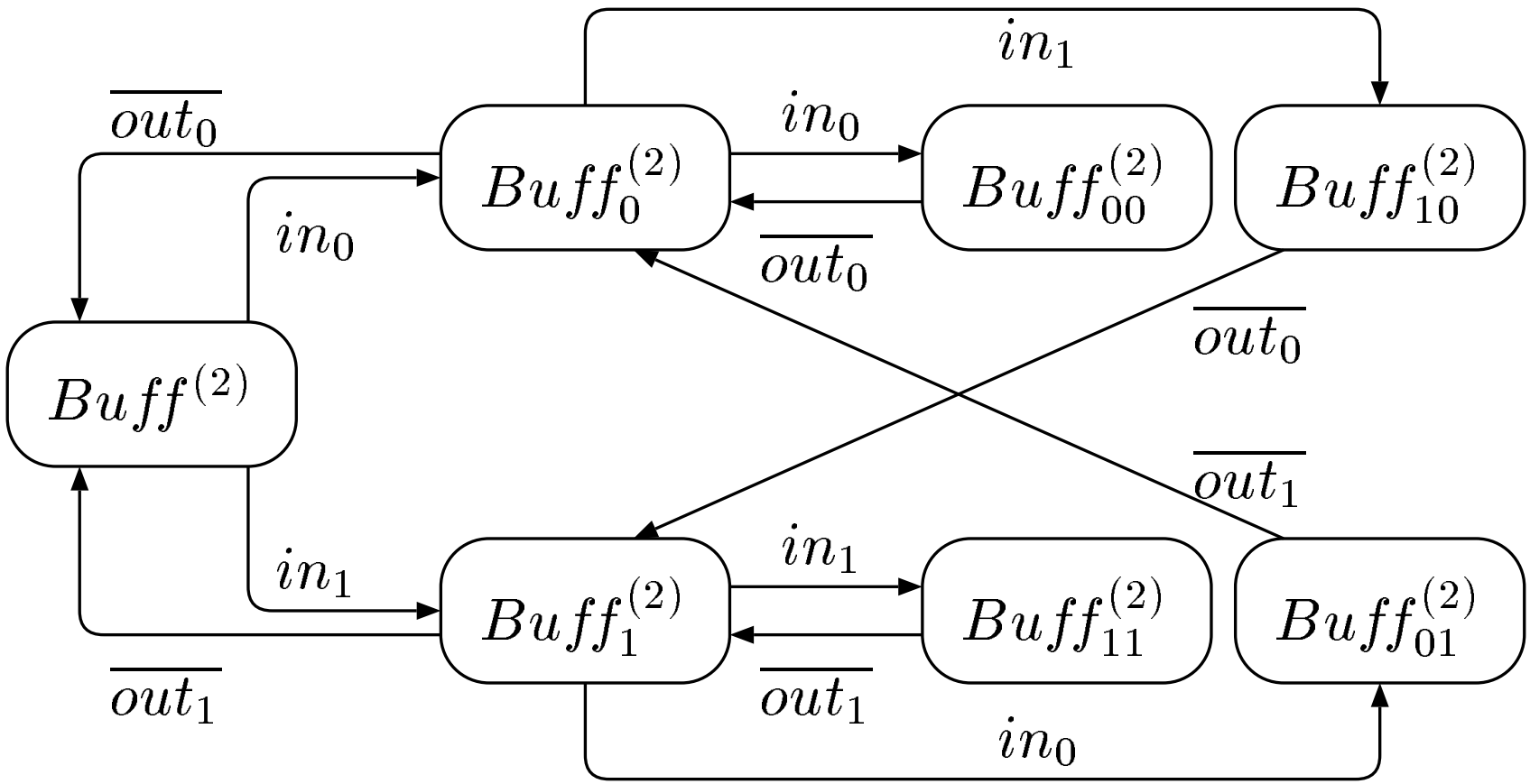
Boolean buffer の defining equations

$$Buff^{(2)} \stackrel{\text{def}}{=} \sum_{i\in\{0,1\}} in_i.Buff_i^{(2)}$$

$$Buff_i^{(2)} \stackrel{\text{def}}{=} \overline{out_i}.Buff^{(2)} + \sum_{j\in\{0,1\}} in_j.Buff_{ji}^{(2)}$$

$$Buff_{ij}^{(2)} \stackrel{\text{def}}{=} \overline{out_j}.Buff_i^{(2)}$$

Boolean buffer の 図



3.6 Scheduler

Scheduler とは... (1)

n 個の agent $\{P_1, \ldots, P_n\}$ が順番に仕事をしていく

$\longrightarrow$ scheduler はそれを調整する

Scheduler のボタン :

a_i : P_i が仕事を始めるときに押される $(i = 1 \ldots, n)$

b_i : P_i が仕事を終えたときに押される $(i = 1 \ldots, n)$

Scheduler とは... (2)

Scheduler の仕様 (informal)

(1) $a_1, \ldots, a_n$ は a_1 から順に押される

(2) それぞれの i に対して，a_i と b_i は a_i から始まって，交互に押される

(3) (1),(2) の条件を満たしていれば，どのボタンをいつ押しても良い

Scheduler とは... (3)

状態 $Sched_{i,X} \quad (i \in \{1, \ldots, n\}, X \subseteq \{1, \ldots, n\})$ では，

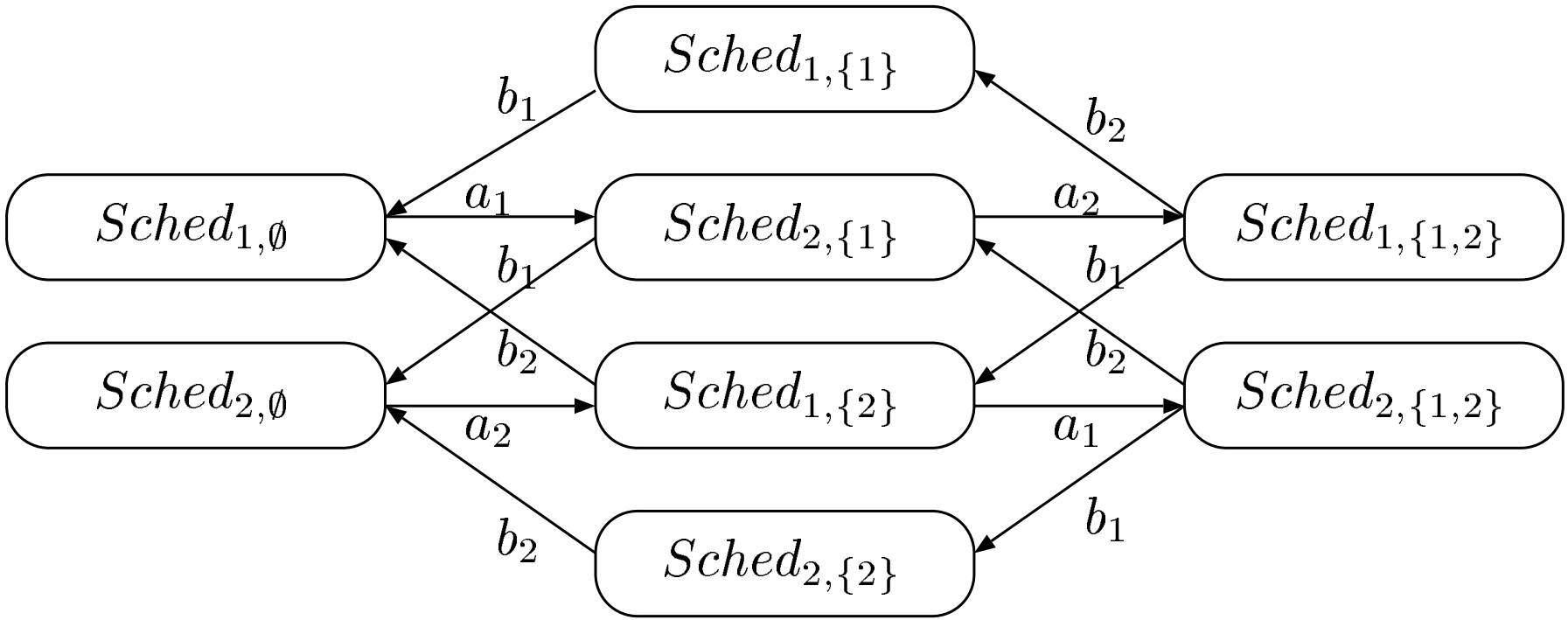
- 次に仕事できるのは P_i
- $P_j \ (j \in X)$ は仕事をしている最中

Defining equations は

$$Scheduler \stackrel{\text{def}}{=} Sched_{1,\emptyset}$$

$$Sched_{i,X} \stackrel{\text{def}}{=} \begin{cases} \sum_{j \in X} b_j.Sched_{i,X-j} & (i \in X) \\ \sum_{j \in X} b_j.Sched_{i,X-j} + a_i.Sched_{i+1,X \cup i} & (i \notin X) \end{cases}$$

Scheduler の図



3.7 Counter

Counter とは...

任意の自然数を保持できるカウンタ (無限の状態をもつ)

ボタン：

inc	:	押す度に increment する
$\overline{zero}$	:	0 のときに押すと，それを確認できる
$\overline{dec}$	:	0 でないときに押すと，decrement できる

Counter の LTS

状態： $Count_n\ (n \geq 0)$

$$Count_0 \stackrel{\text{def}}{=} inc.Count_1 + \overline{zero}.Count_0$$

$$Count_{n+1} \stackrel{\text{def}}{=} inc.Count_{n+2} + \overline{dec}.Count_n$$

$$Count \stackrel{\text{def}}{=} Count_0$$

3.8 Summary

まとめ

- labelled transition systems, bisimilarity の導入
- その例をいくつか観察