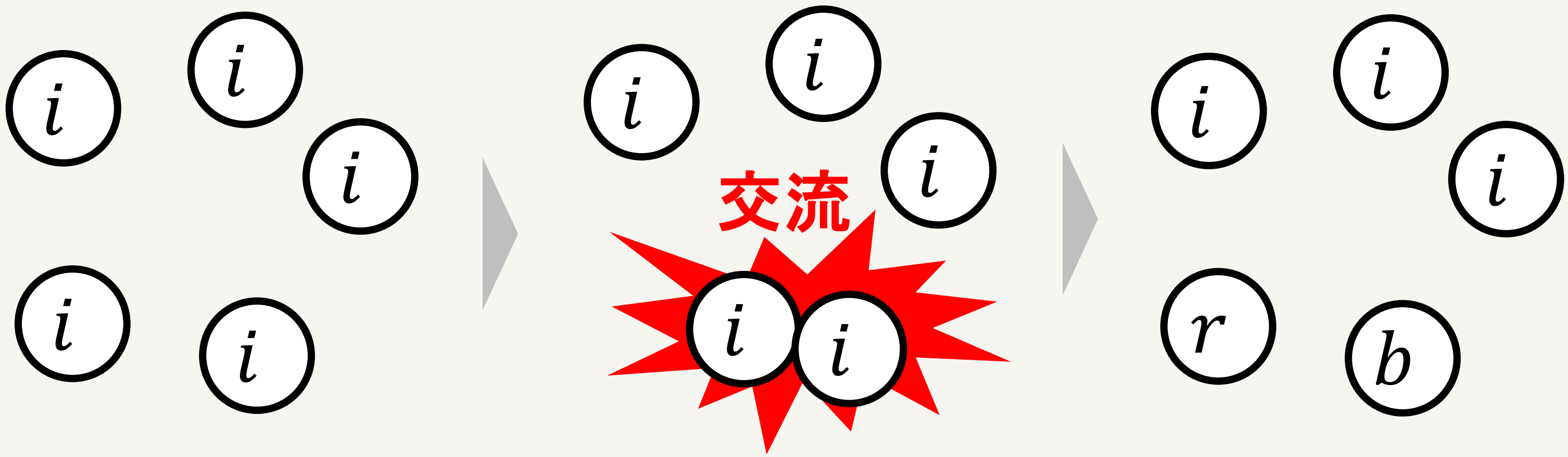
通信グラフの形状判定を行う個体群プロトコル

奈良先端科学技術大学院大学
安見 嘉人　大下 福仁　井上 美智子

個体群プロトコルモデル [1]

- 複数の受動的に移動する個体で構成
  - 各個体は同一の状態機械
  - 2つの個体が近づいたとき，交流により状態を更新

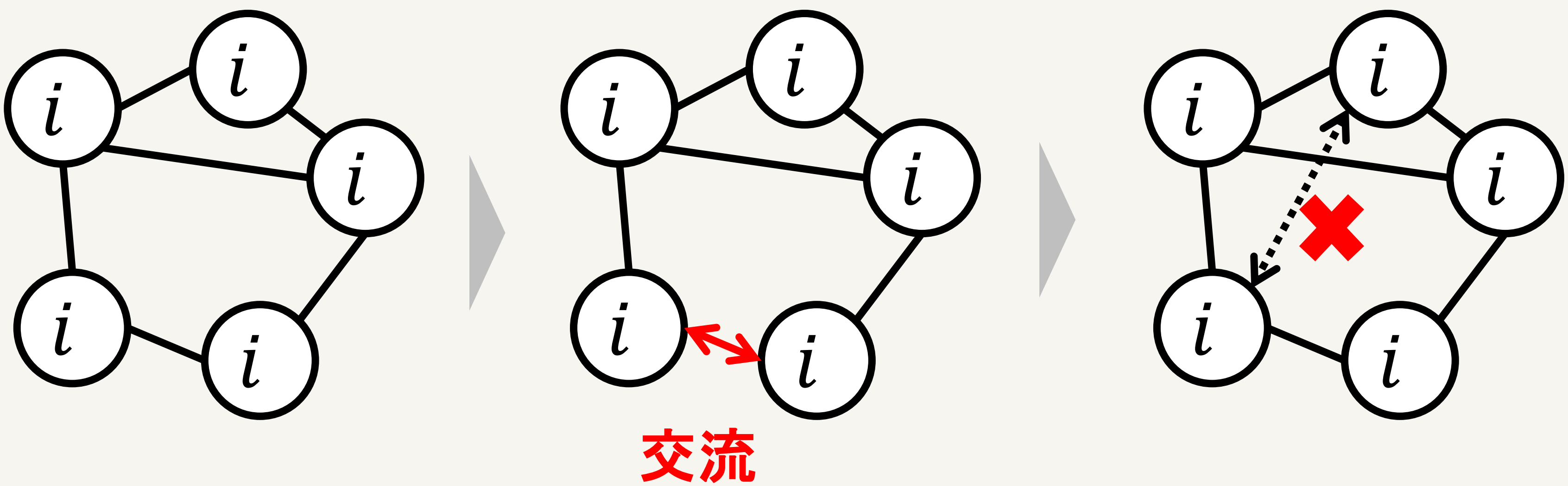


- 応用例
  - センサネットワークを用いた野鳥の観測
  - 分子ロボット

[1] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M. J., and Peralta, R., Distributed Computing(2006)

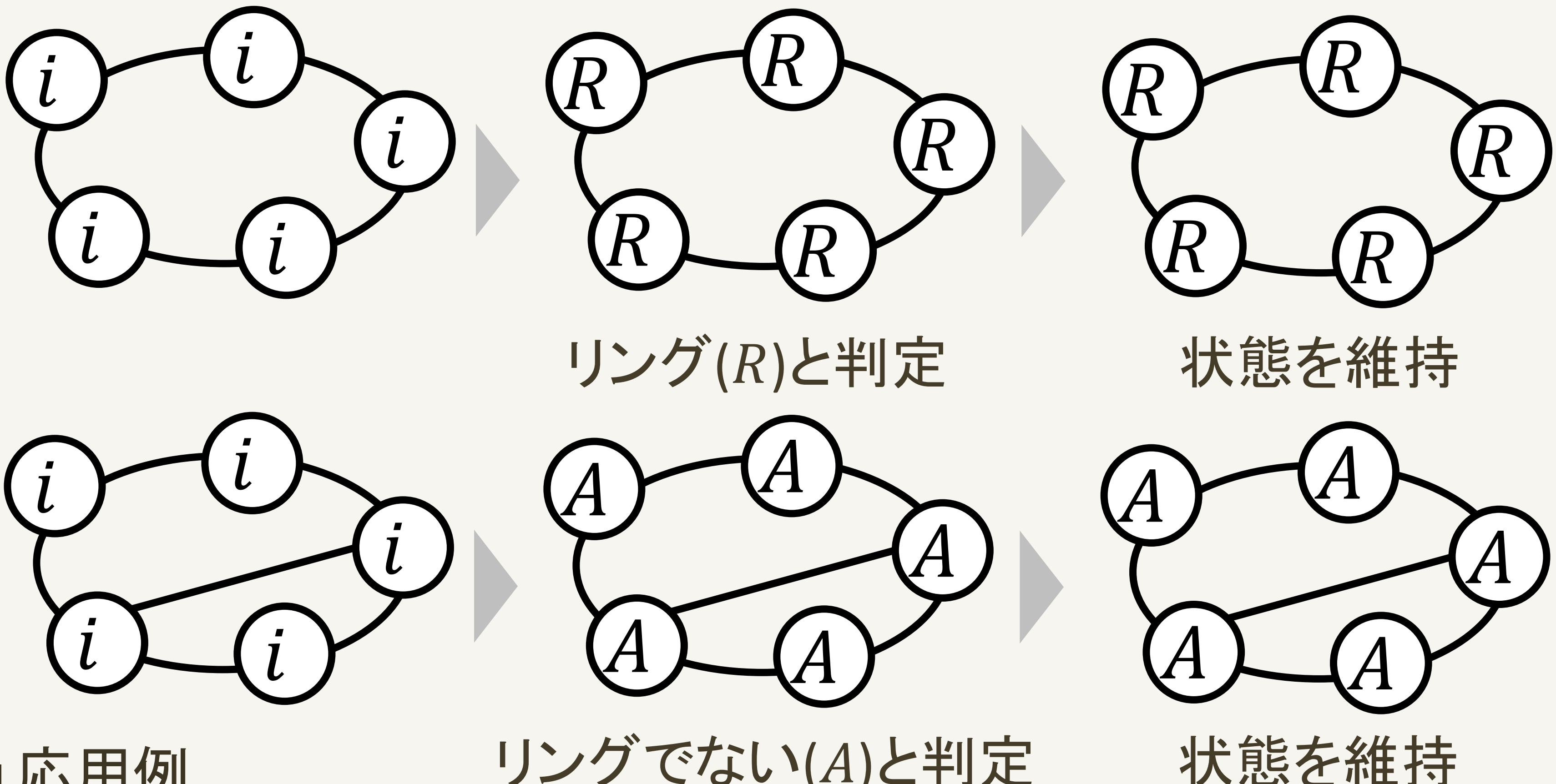
個体群プロトコルモデルの通信グラフ

- グラフのノードは個体
- グラフの辺はその個体間で交流可能かどうか



形状判定を行う個体群プロトコル

- 与えられた通信グラフが所望のグラフ形状であるか判定
  - ライン・リング・木・スター・二部グラフ・k−正則グラフ・完全グラフを判定



- 応用例
  - 特定のグラフ形状で動作するプロトコルが動作可能かわかる

研究成果

各判定に必要な状態数 (Pは個体数の上界, nは個体数)

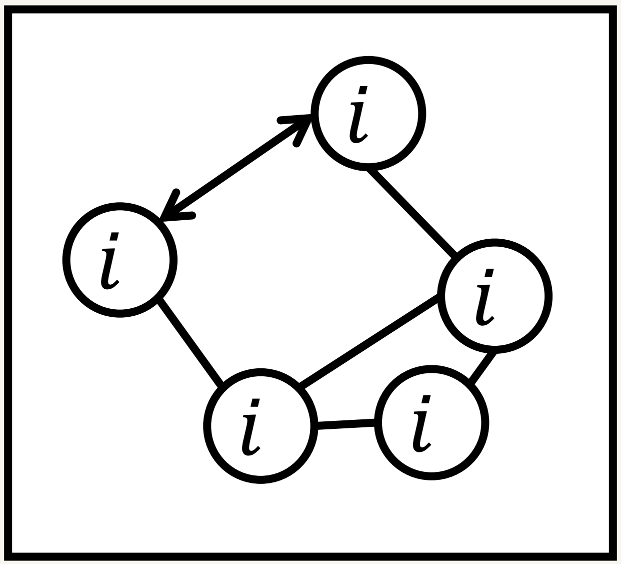
<table>
<tr><th colspan="3">モデル</th><th colspan="4">グラフ形状</th></tr>
<tr><th>初期値</th><th>公平性</th><th>入力</th><th>ライン・リング・木・二部グラフ</th><th>k-正則</th><th>スター</th><th>完全</th></tr>
<tr><td rowspan="6">一定</td><td rowspan="3">全体公平</td><td>n</td><td>$O(1)$</td><td>$O(k\log n)$</td><td>$O(1)$</td><td>$O(n)$</td></tr>
<tr><td>P</td><td>$O(1)$</td><td>$O(k\log P)$</td><td>$O(1)$</td><td>$O(P^2)$</td></tr>
<tr><td>なし</td><td>$O(1)$</td><td></td><td>$O(1)$</td><td></td></tr>
<tr><td rowspan="3">弱公平</td><td>n</td><td colspan="2">不可</td><td>$O(n)$</td><td>$O(n)$</td></tr>
<tr><td>P</td><td colspan="4" rowspan="2">不可</td></tr>
<tr><td>なし</td></tr>
<tr><td rowspan="3">任意</td><td rowspan="3">全体公平/弱公平</td><td>n</td><td colspan="4" rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
</table>

公平性(fairness)

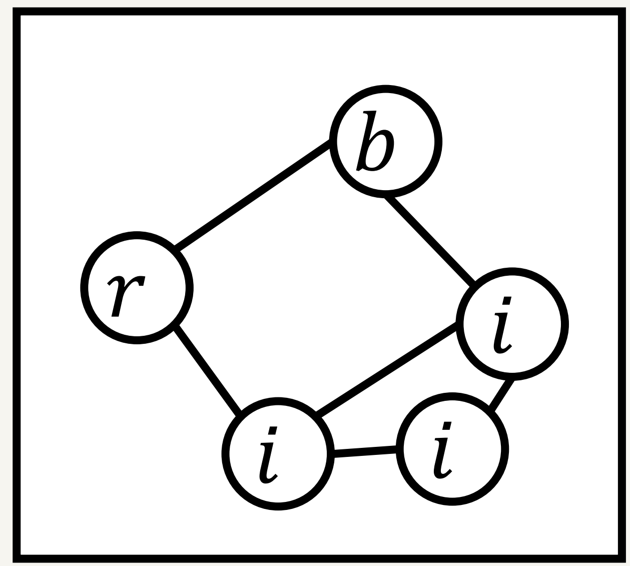
- 交流順序に関する仮定
- 公平性の種類
  - 弱公平性
  - 全体公平性

状況（configuration）

- 個体群の大域的な状態
  - 状態が遷移すると状況も遷移



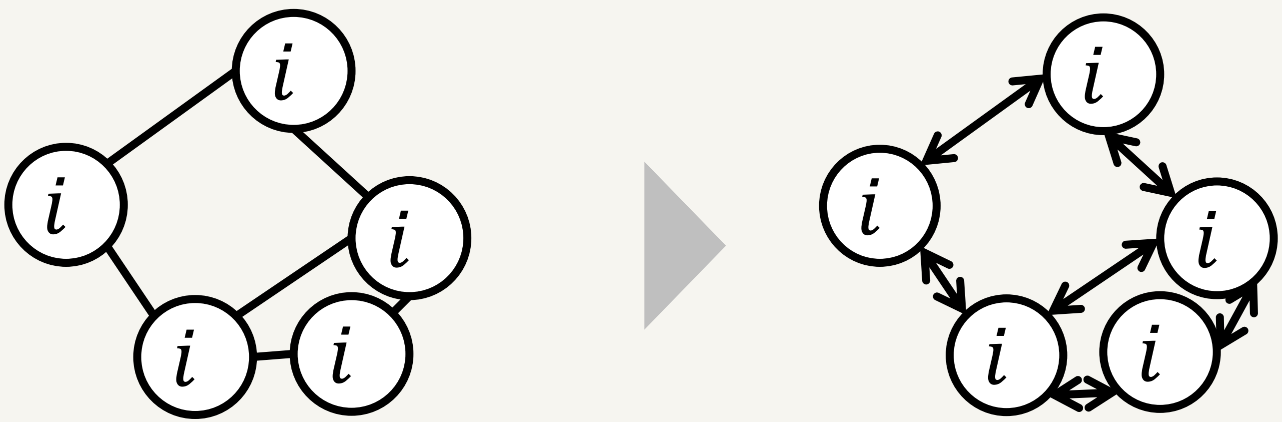
状況C



状況C’

弱公平性(weak fairness)

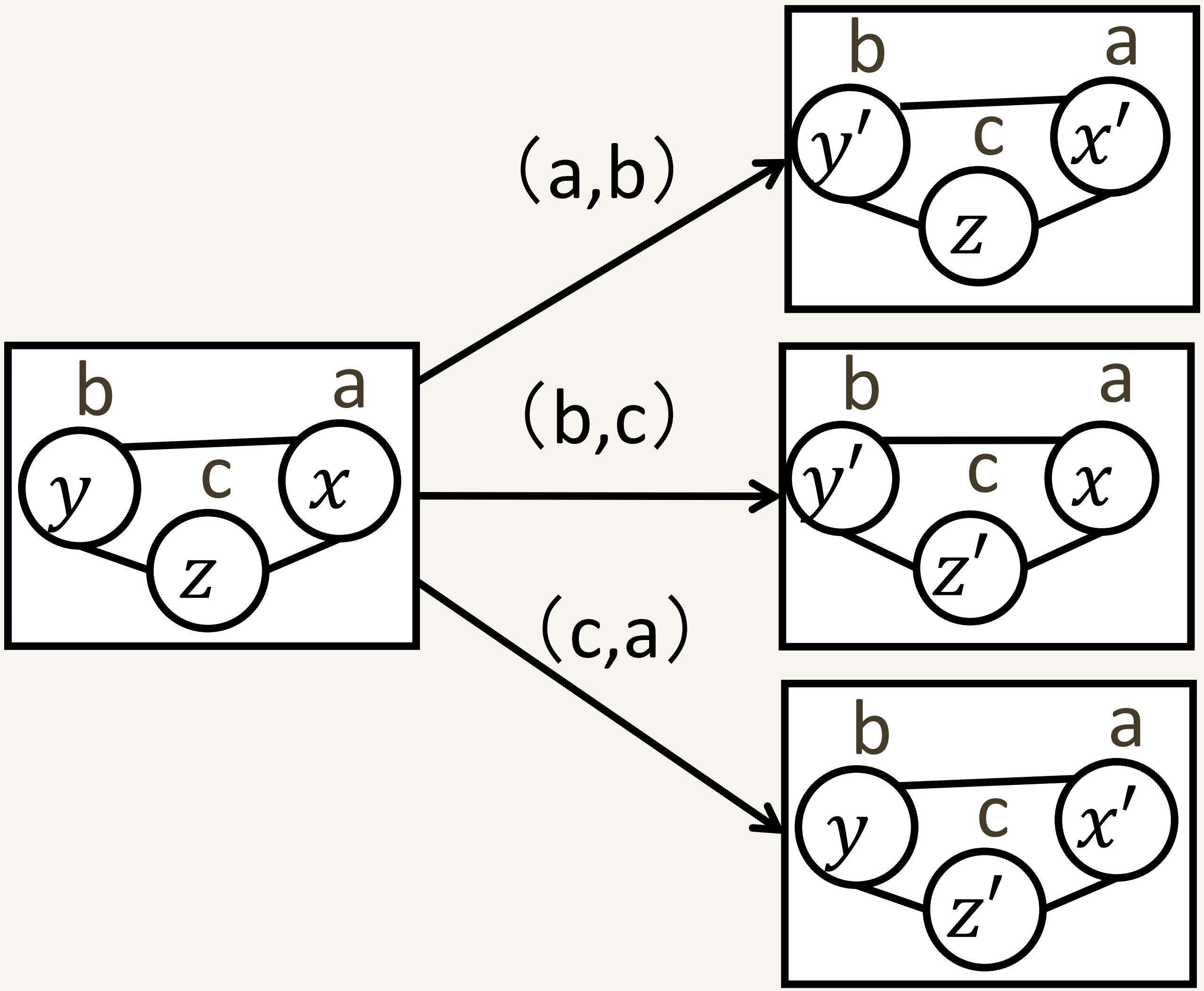
- 交流可能な各個体間で無限回交流が発生



それぞれ無限回交流

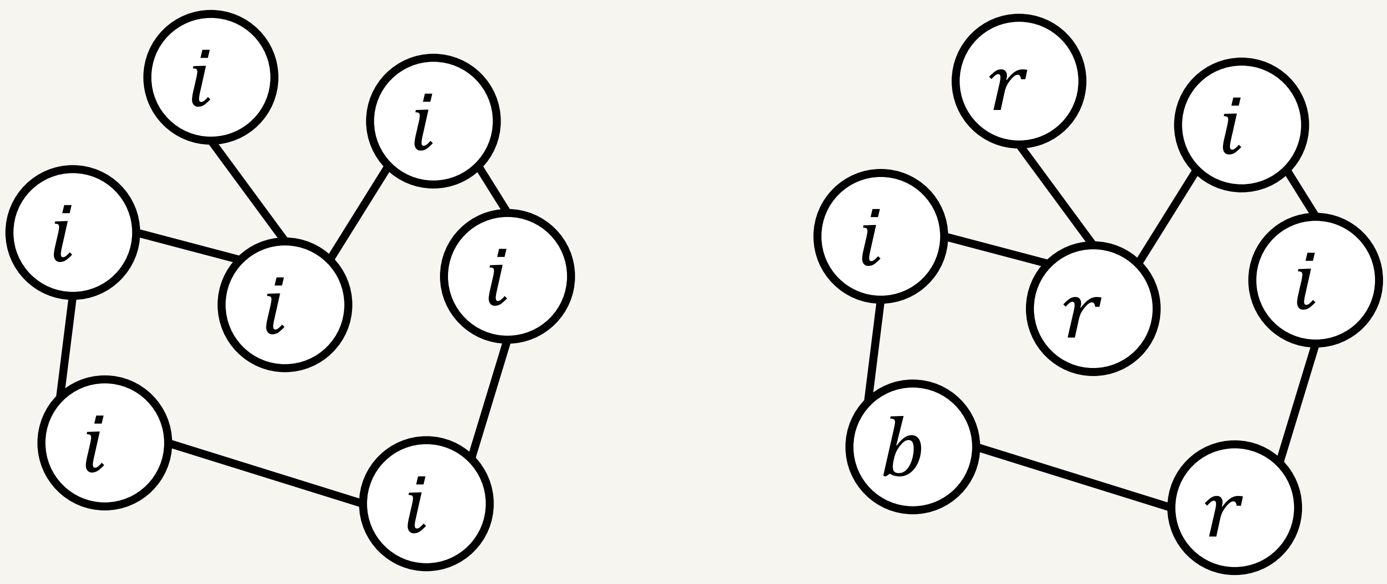
全体公平性(global fairness)

- 無限回現れる状況から起こりうる全ての状況に無限回遷移



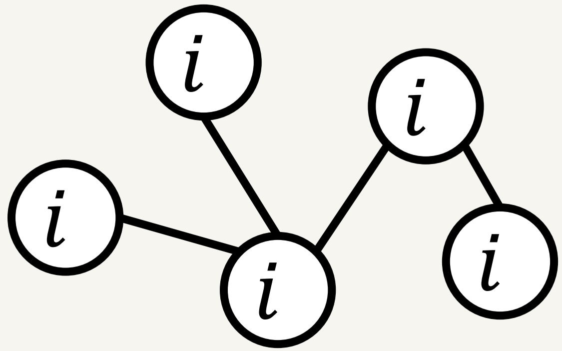
初期状態

- 初期値一定
  - 全ての個体が同じ初期状態を持つ
- 初期値任意
  - 個体は任意の初期状態を持つ



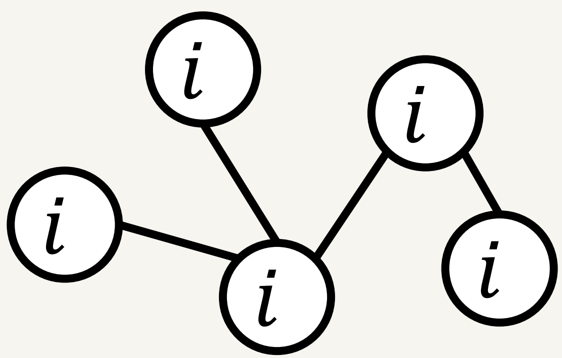
個体数に関する入力

■入力として個体数nが与えられる



$n = 5$が入力
個体数が5とわかる

■入力として個体数の上界Pが与えられる



$P \geq 5$が入力
個体数がP以下とわかる

■入力なし

研究成果

各判定に必要な状態数 (Pは個体数の上界, nは個体数)

<table>
<tr><th colspan="3">モデル</th><th colspan="4">グラフ形状</th></tr>
<tr><th>初期値</th><th>公平性</th><th>入力</th><th>ライン・リング・
木・二部グラフ</th><th>k-正則</th><th>スター</th><th>完全</th></tr>
<tr><td rowspan="6">一定</td><td rowspan="3">全体公平</td><td>n</td><td>$O(1)$</td><td>$O(k \log n)$</td><td>$O(1)$</td><td>$O(n)$</td></tr>
<tr><td>P</td><td>$O(1)$</td><td>$O(k \log P)$</td><td>$O(1)$</td><td>$O(P^2)$</td></tr>
<tr><td>なし</td><td>$O(1)$</td><td></td><td>$O(1)$</td><td></td></tr>
<tr><td rowspan="3">弱公平</td><td>n</td><td colspan="2">不可</td><td>$O(n)$</td><td>$O(n)$</td></tr>
<tr><td>P</td><td colspan="4" rowspan="2">不可</td></tr>
<tr><td>なし</td></tr>
<tr><td rowspan="3">任意</td><td rowspan="3">全体公平/
弱公平</td><td>n</td><td colspan="4" rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
</table>

研究成果

判定に必要な状態数 (Pは個体数の上界, nは個体数)

<table>
<tr><th colspan="3">モデル</th><th rowspan="2">ライン・リング・木・二部グラフ</th></tr>
<tr><th>初期値</th><th>公平性</th><th>入力</th></tr>
<tr><td rowspan="6">一定</td><td rowspan="3">全体公平</td><td>n</td><td>$O(1)$</td></tr>
<tr><td>P</td><td>$O(1)$</td></tr>
<tr><td>なし</td><td>$O(1)$</td></tr>
<tr><td rowspan="3">弱公平</td><td>n</td><td rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
<tr><td rowspan="3">任意</td><td rowspan="3">全体公平
/弱公平</td><td>n</td><td rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
</table>

<table>
<tr><th colspan="2">グラフ特性</th></tr>
<tr><th>閉路</th><th>次数k以上</th></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td colspan="2">不可</td></tr>
<tr><td colspan="2">不可</td></tr>
</table>

研究成果

判定に必要な状態数 (Pは個体数の上界, nは個体数)

<table>
<tr><th colspan="3">モデル</th><th rowspan="2">ライン・リング・木・二部グラフ</th></tr>
<tr><th>初期値</th><th>公平性</th><th>入力</th></tr>
<tr><td rowspan="6">一定</td><td rowspan="3">全体公平</td><td>n</td><td>$O(1)$</td></tr>
<tr><td>P</td><td>$O(1)$</td></tr>
<tr><td>なし</td><td>$O(1)$</td></tr>
<tr><td rowspan="3">弱公平</td><td>n</td><td rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
<tr><td rowspan="3">任意</td><td rowspan="3">全体公平
/弱公平</td><td>n</td><td rowspan="3">不可</td></tr>
<tr><td>P</td></tr>
<tr><td>なし</td></tr>
</table>

<table>
<tr><th colspan="2">グラフ特性</th></tr>
<tr><th>閉路</th><th>次数k以上</th></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td>$O(1)$</td><td>$O(k)$</td></tr>
<tr><td colspan="2">不可</td></tr>
<tr><td colspan="2">不可</td></tr>
</table>

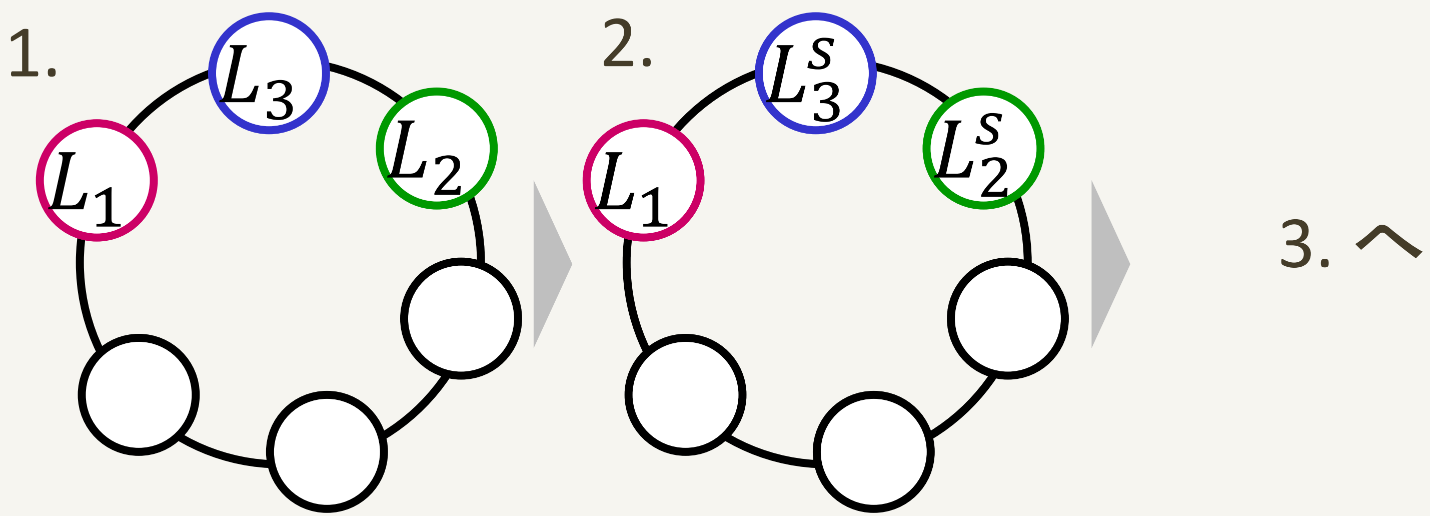
アイデア

初期値一定
全体公平

1. グラフ中を移動するトークンを3つ選出(L_1, L_2, L_3)

2. トークン2つ(L_2, L_3)を隣り合わせに停止させ(L_2^S, L_3^S),
残りの1つ(L_1)がそれらを目印に閉路探索

3. トークン2つを目印にするので左右がわかり，閉路を発見可能
(例：$L_3^S \rightarrow L_2^S$の順で交流したら閉路あり)



※初めは全員が閉路なしと判定

アイデア

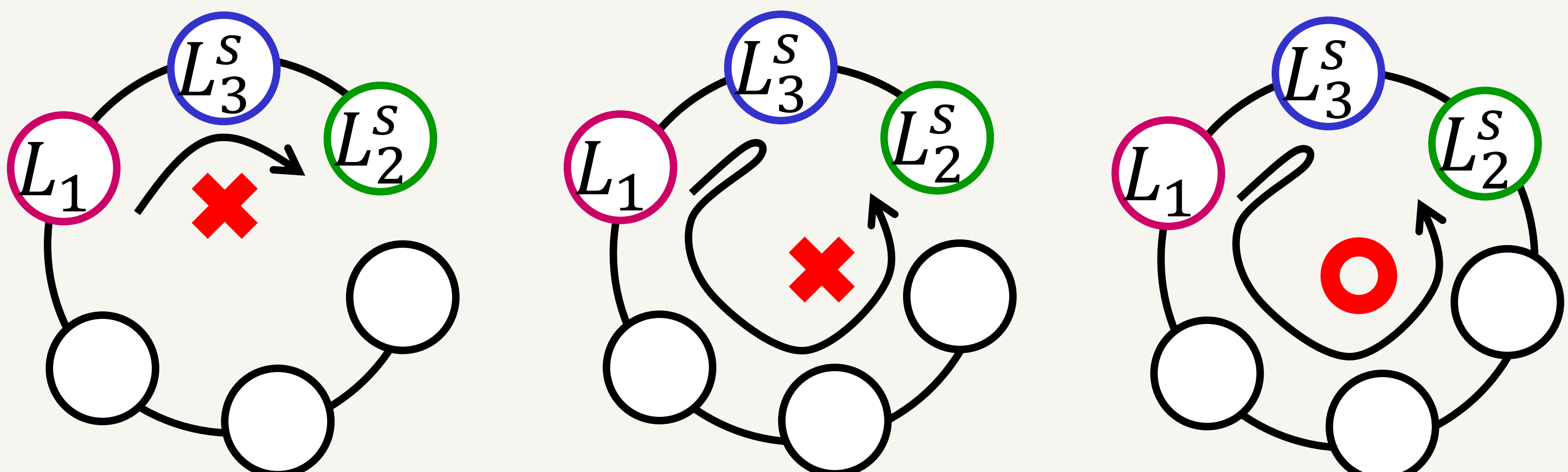
初期値一定
全体公平

1. グラフ中を移動するトークンを3つ選出(L_1, L_2, L_3)

2. トークン2つ(L_2, L_3)を隣り合わせに停止させ(L_2^s, L_3^s),
残りの1つ(L_1)がそれらを目印に閉路探索

3. トークン2つを目印にするので左右がわかり，閉路を発見可能
(例：$L_3^s \rightarrow L_2^s$の順で交流したら閉路あり)

3.

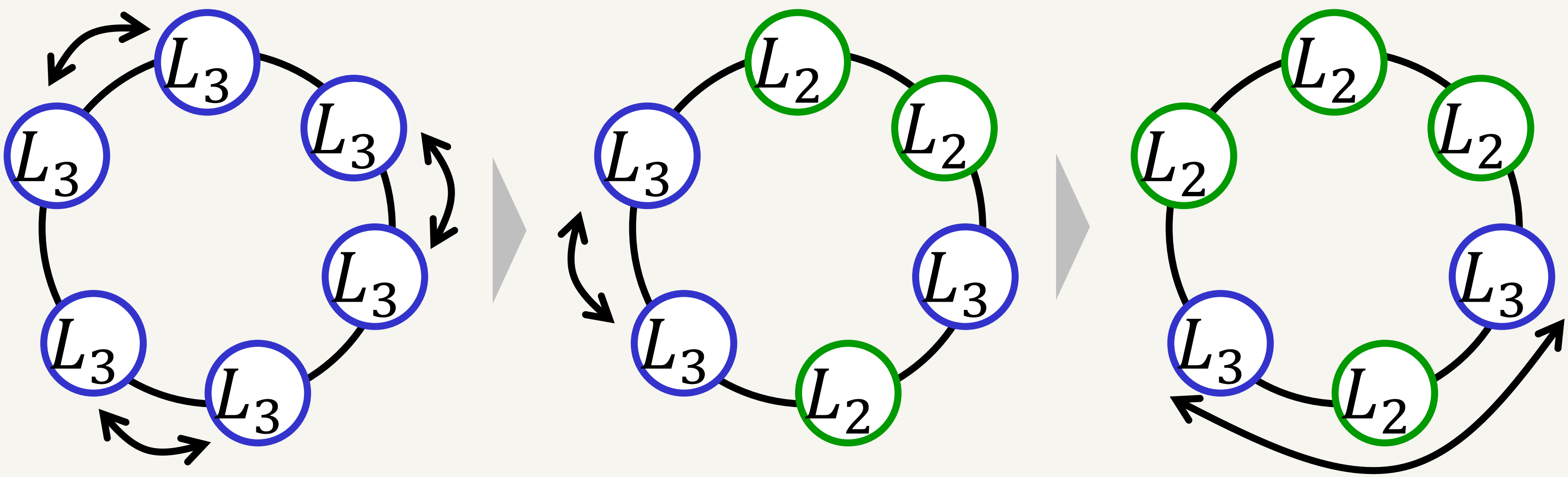


※初めは全員が閉路なしと判定

1: トークン選出

初期値一定
全体公平

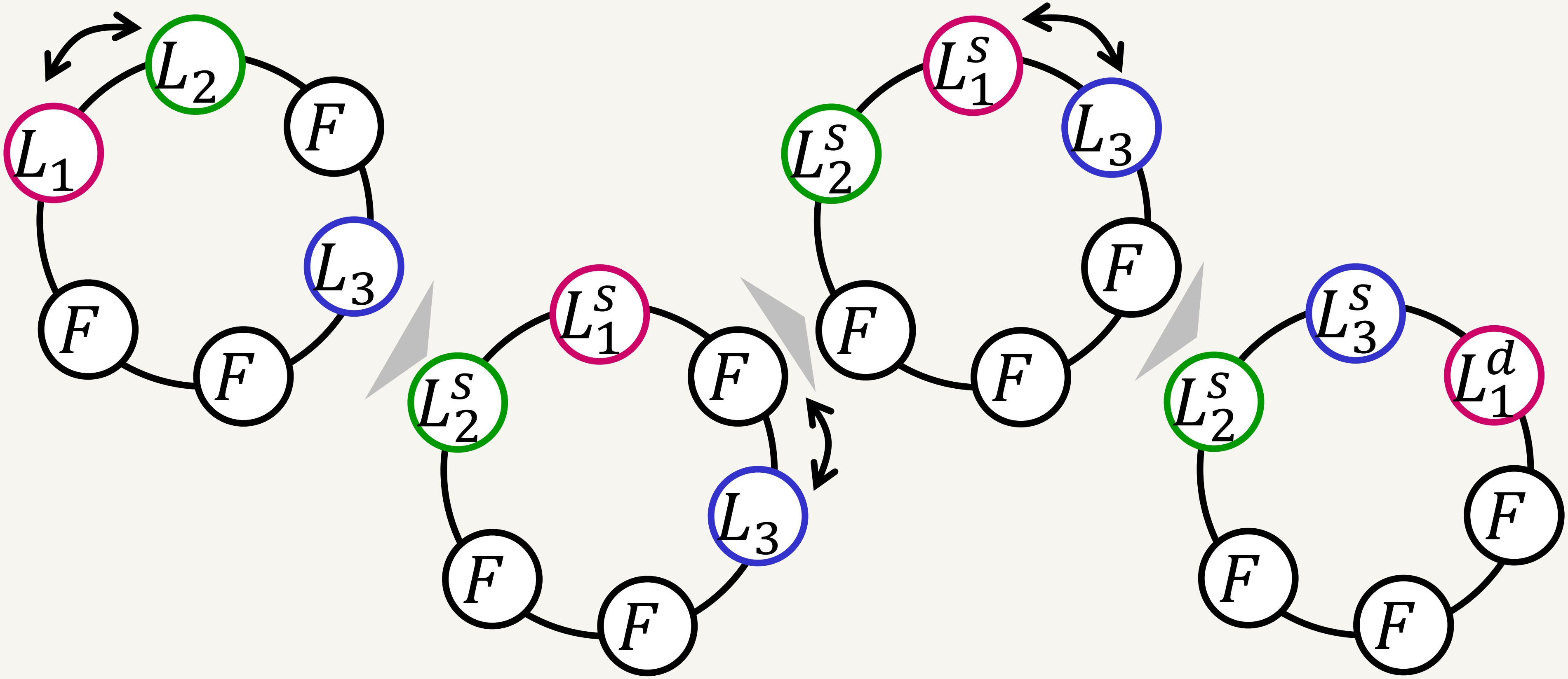
- 初期状態では，全個体がL_3を持つ
- 選出動作: $(L_3, L_3) \to (L_3, L_2)$,
  $(L_2, L_2) \to (L_2, L_1), (L_1, L_1) \to (L_1, F)$
- トークンはグラフ中を移動できるので，いつか選出が完了



2: トークン停止

初期値一定
全体公平

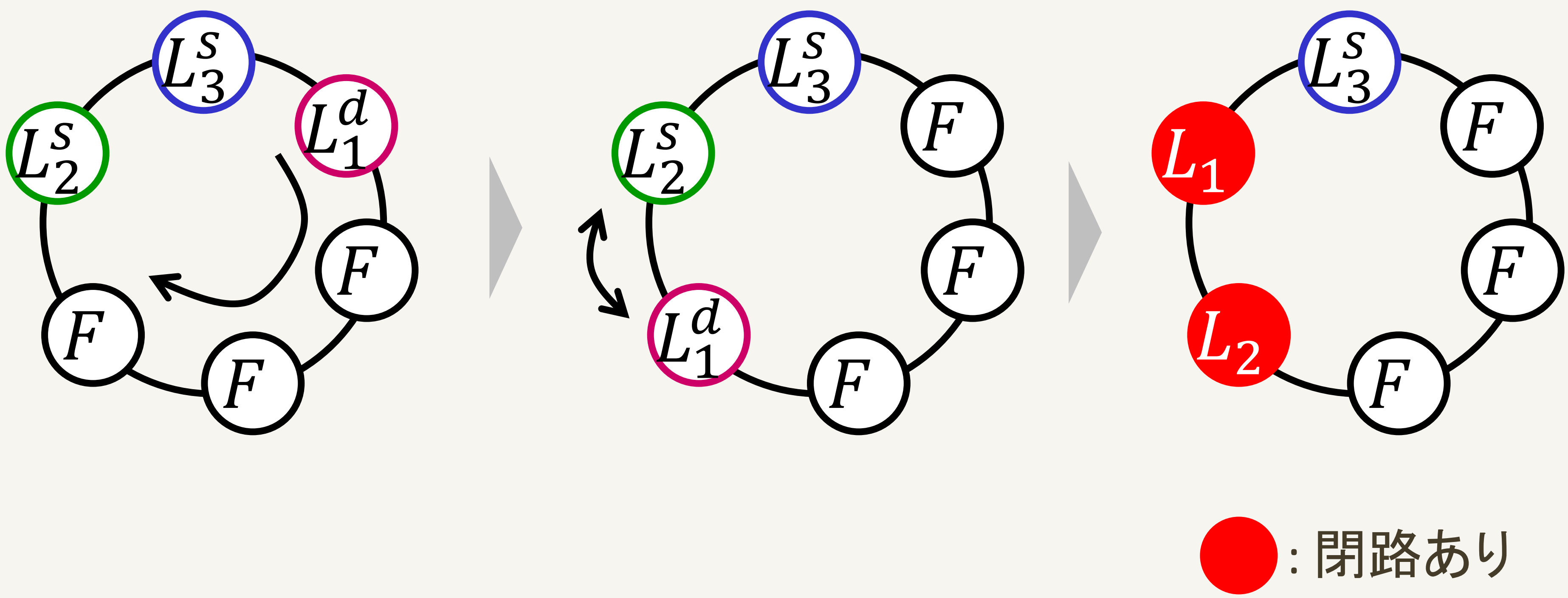
- L_1はL_2, L_3を順番に停止状態(L_2^s, L_3^s)にする
- L_2^sとL_3^sが隣り合うようにするため，L_1はL_2と交流したらL_3を待つ(L_1^s)



3: 閉路探索

- トークン2つが停止したら，L_1は閉路を探すモード（L_1^d）になる
- L_1^dが最後に交流したのはL_3で，L_2^sと交流するには別の道を通るしかない
  - $\left(L_1^d, L_2^s\right)$ →閉路あり

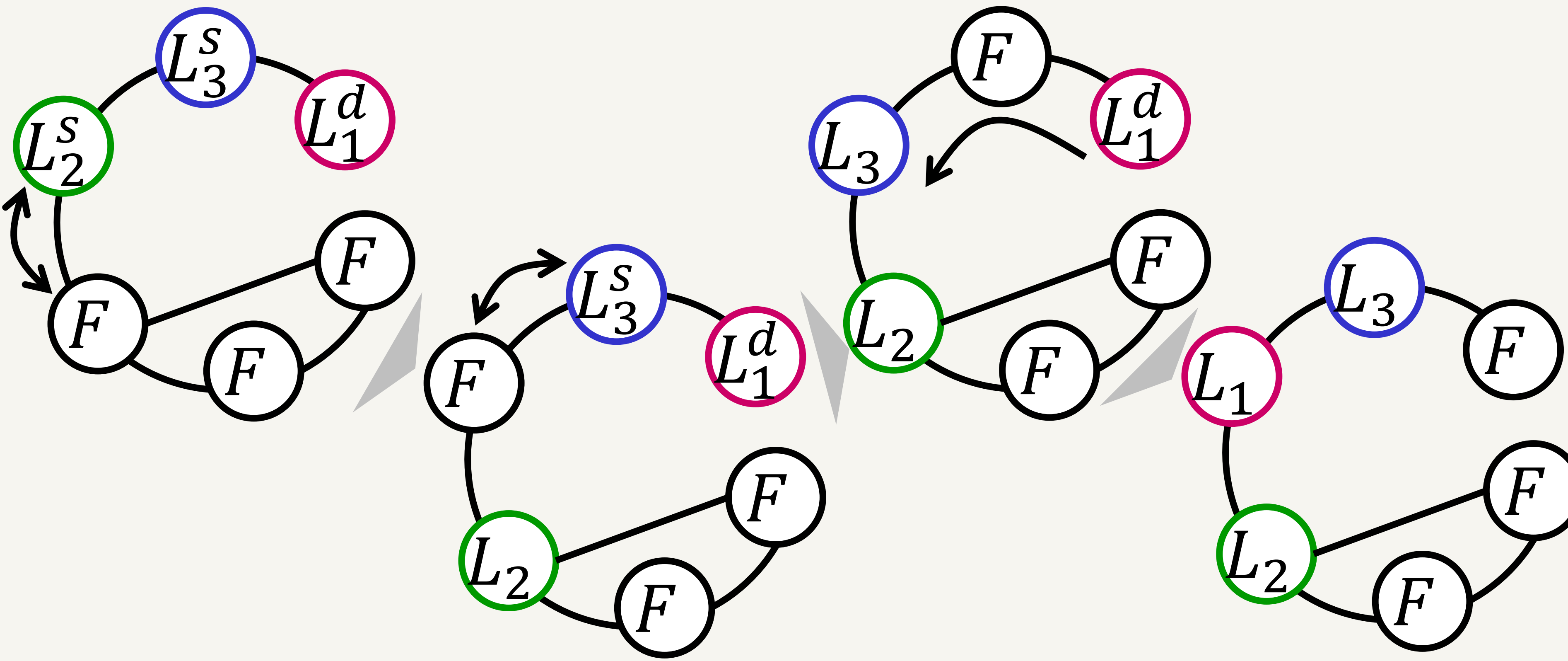
初期値一定
全体公平



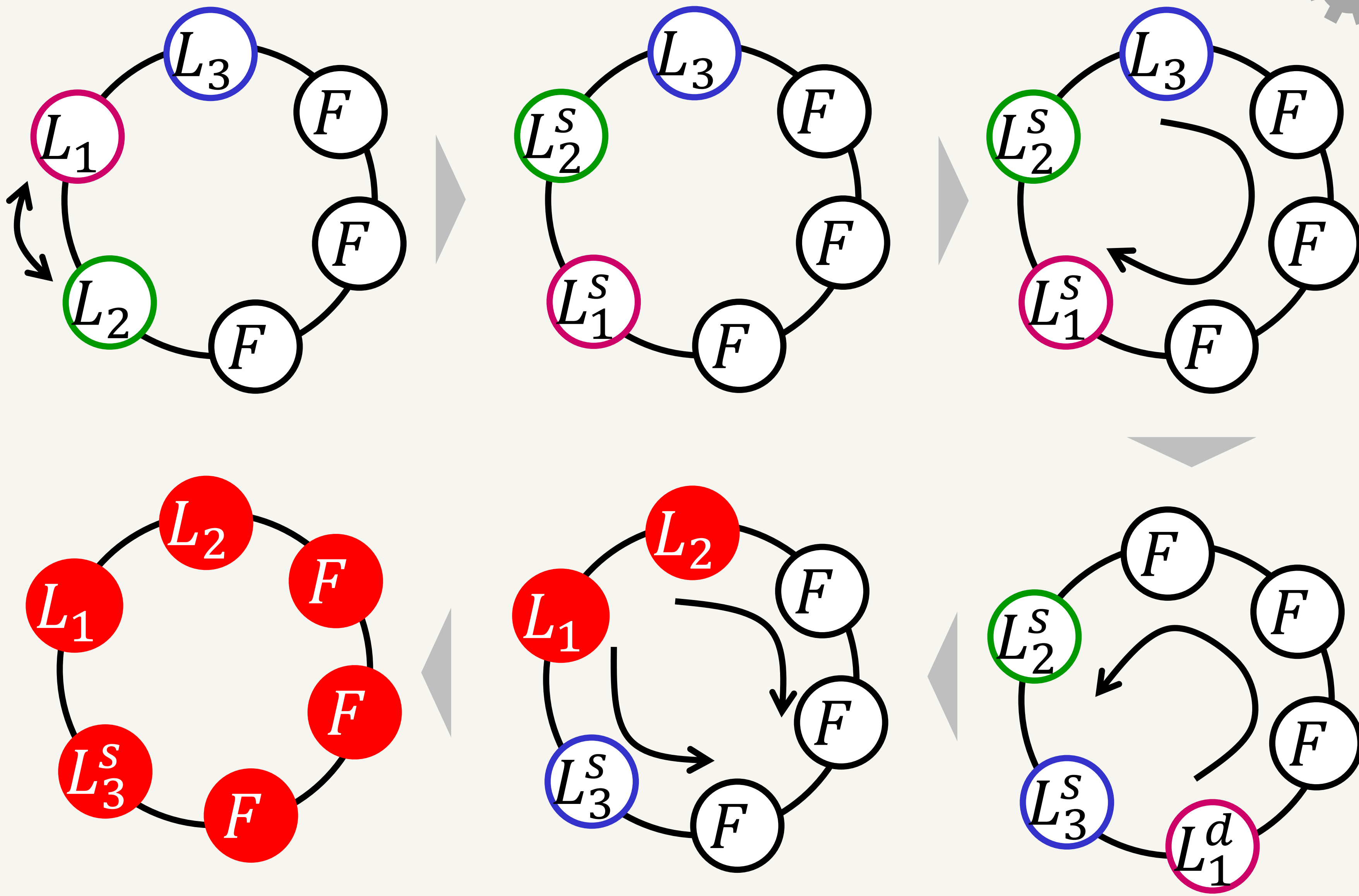
問題点

- 停止場所が悪いと閉路を発見できない
- 対処法
  - 他の個体と交流したら，停止状態から復帰
  - 全体公平から，いつか閉路でトークンが停止し，閉路を発見可能

初期値一定
全体公平



実行例


L_3
F
L_1
F
L_2
F
L_3
L_2^s
F
F
L_1^s
F
L_3
L_2^s
F
F
L_1^s
F
F
L_2^s
F
F
L_3^s
L_1^d
L_2
L_1
F
F
L_3^s
F
L_2
L_1
F
F
L_3^s
F

研究成果

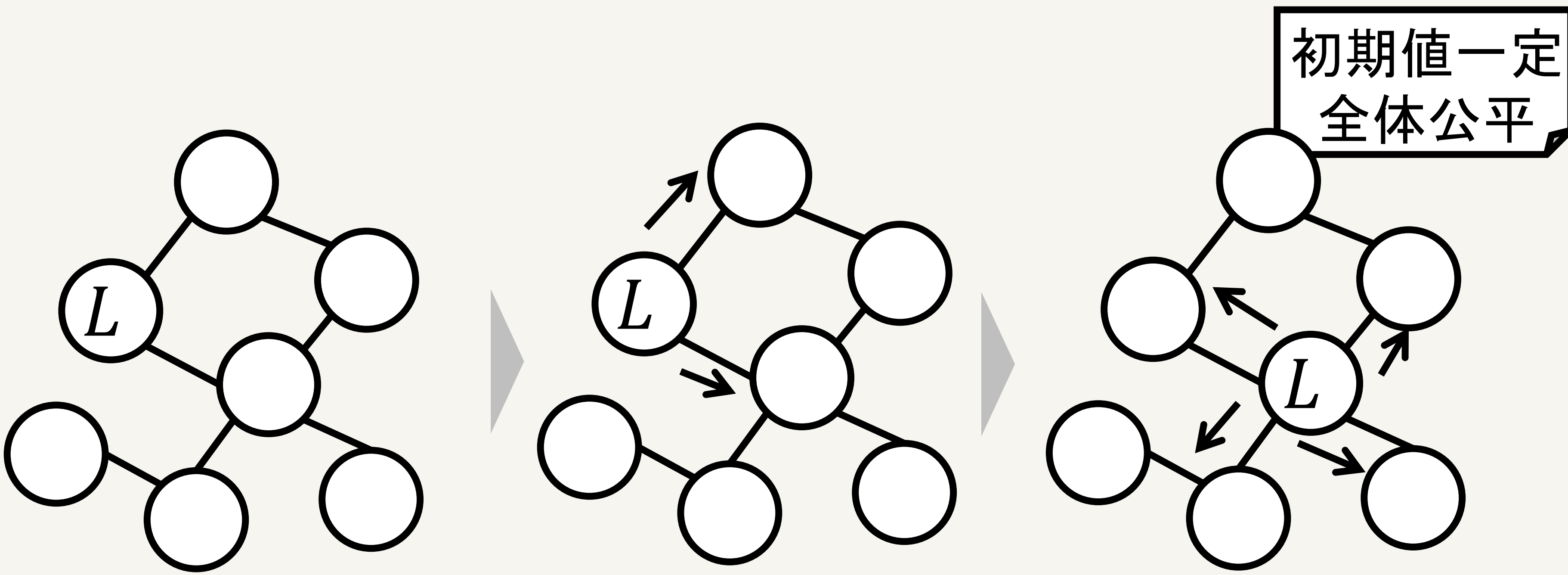
判定に必要な状態数 (Pは個体数の上界, nは個体数)

モデル			ライン・リング・木・二部グラフ
初期値	公平性	入力	
一定	全体公平	n	$O(1)$
		P	$O(1)$
		なし	$O(1)$
	弱公平	n	不可
		P	
		なし	
任意	全体公平/弱公平	n	不可
		P	
		なし	

グラフ特性	
閉路	次数k以上
$O(1)$	$O(k)$
$O(1)$	$O(k)$
$O(1)$	$O(k)$
不可	
不可	

アイデア

1. グラフ中を移動するリーダートークンを1つ選出(L)
2. リーダートークンは現在トークンを持つ個体の次数を調べながらグラフを移動
3. 全体公平性から, いつか全個体を調べ終わる

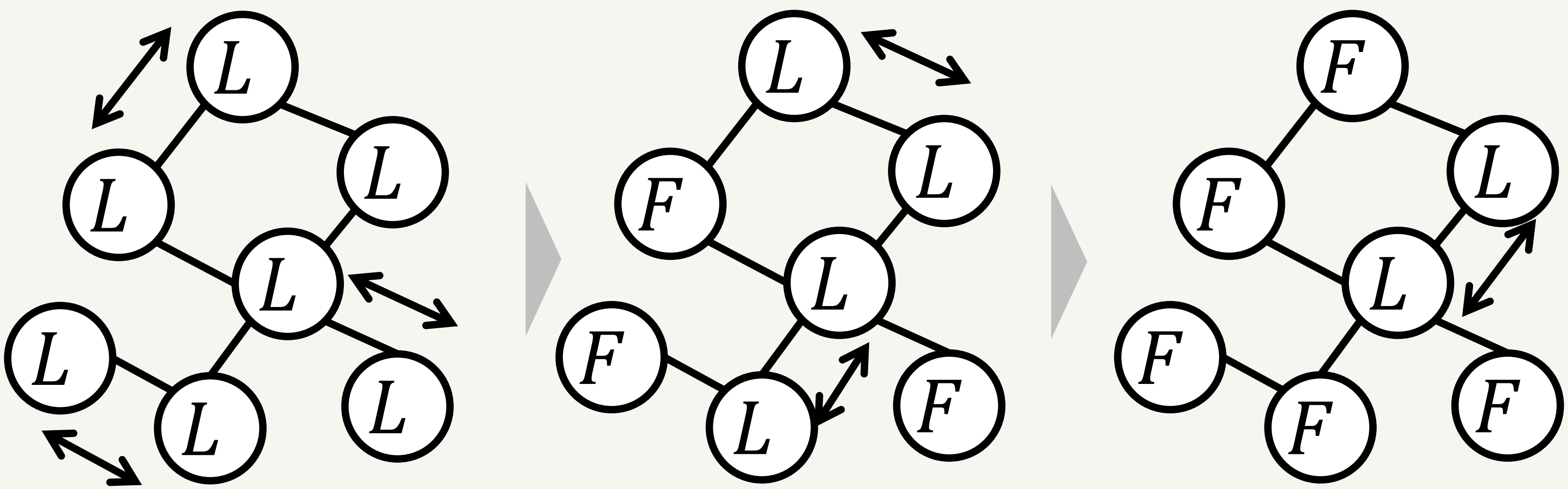


※初めは全員がk以上なしと判定

1: リーダートークンの選出

初期値一定
全体公平

- 初期状態では，全個体がLを持つ
- $(L, L) \rightarrow (L, F)$
- トークンはグラフ中を移動するため，いつか1つだけが残る



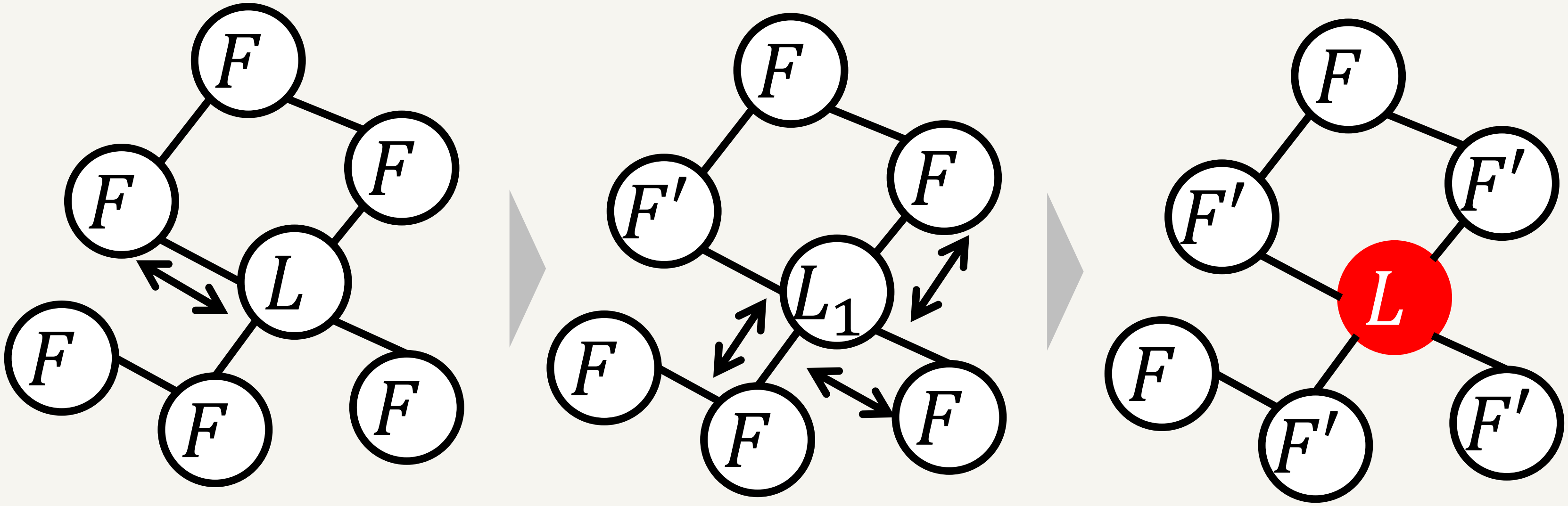
2: 次数を調べる

初期値一定
全体公平

- リーダートークンは交流した個体にマークする
  - このとき，カウントを行う

$(L_x, F) \rightarrow (L_{x+1}, F')$，$(L_{k-1}, F) \rightarrow k$以上ある

- カウントがkになれば，次数がk以上の個体があると判定



$k = 4$の例

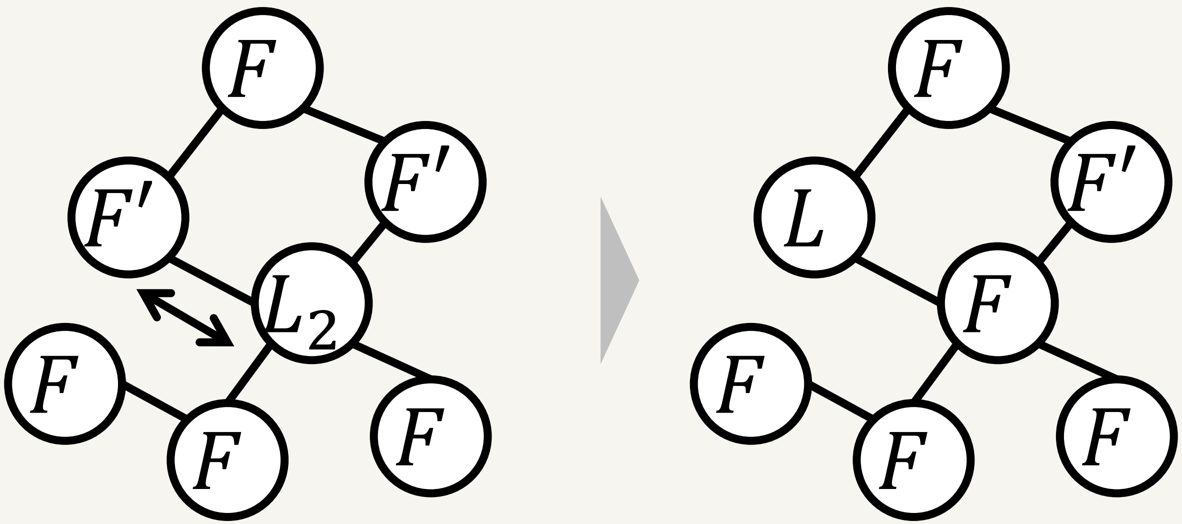
: k以上あり

2: グラフ移動

- リーダートークンはマークした個体と交流すると移動
  - このとき，マークを消し カウントもリセット

$$(L_x, F') \to (F, L)$$

初期値一定
全体公平



まとめと今後の課題

- 今後の課題
  - 時間計算量のよいアルゴリズムの提案
  - 複雑なグラフの判定（グリッド, トーラスなど）

モデル			グラフ形状			
初期値	**公平性**	**入力**	**ライン・リング・木・二部グラフ**	**k-正則**	**スター**	**完全**
一定	全体公平	n	$O(1)$	$O(k\log n)$	$O(1)$	$O(n)$
		P	$O(1)$	$O(k\log P)$	$O(1)$	$O(P^2)$
		なし	$O(1)$		$O(1)$	
	弱公平	n	不可		$O(n)$	$O(n)$
		P	不可			
		なし				
任意			不可			