ペトリネット理論

平石邦彦
北陸先端科学技術大学院大学情報科学研究科
hira@jaist.ac.jp

ペトリネットとは

- 1960年代にドイツ人の Carl Adam Petri によって考案された理論的計算モデル.
- 「並行動作」,「分散状態」,「資源の概念」を表現可能なオートマトン.
- 数学的モデル＋図的言語.
- 実行可能モデル：状態変化を図上に表現可能（トークンゲーム）.

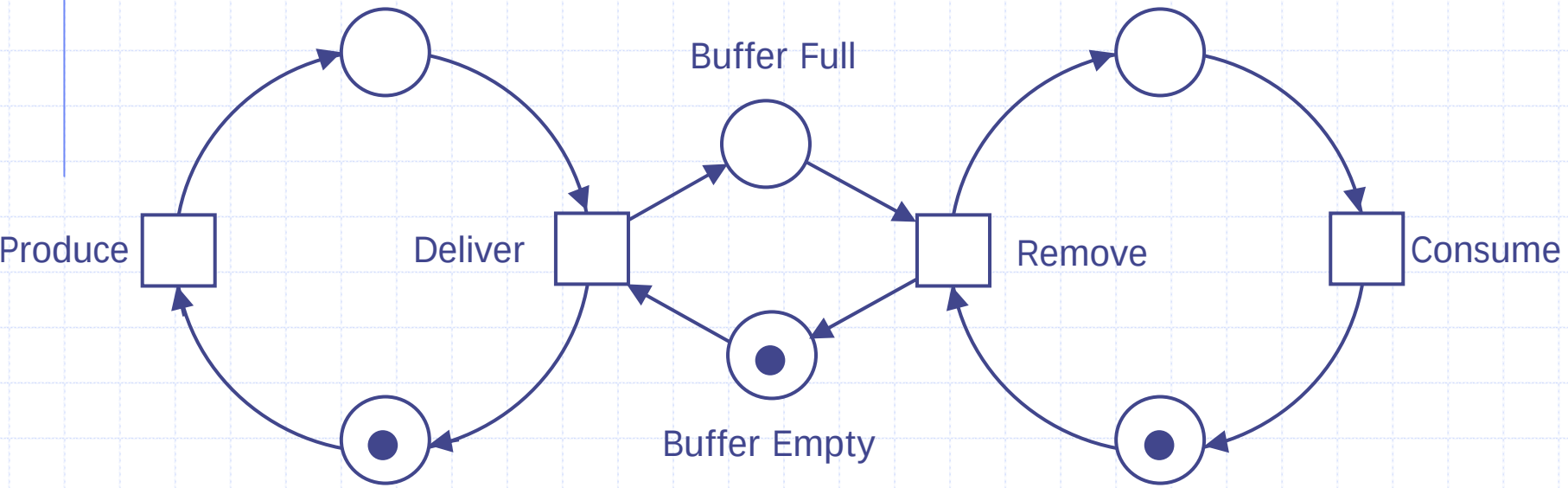
チュートリアルの内容

1. ペトリネットとは？
   - 例：プレース／トランジションネット（ペトリネット）
   - 有限オートマトンとの比較
   - プロセス代数との比較
2. 解析手法
   - 線形代数的手法
   - サブクラス
   - 縮約法
   - 到達可能木，被覆木
   - 時相論理とモデル検査
   - 効率化手法
3. 拡張モデル
   - 時間／確率ペトリネット
   - 高水準ペトリネット
   - ファジイペトリネット
   - ハイブリッドペトリネット
   - オブジェクトペトリネット

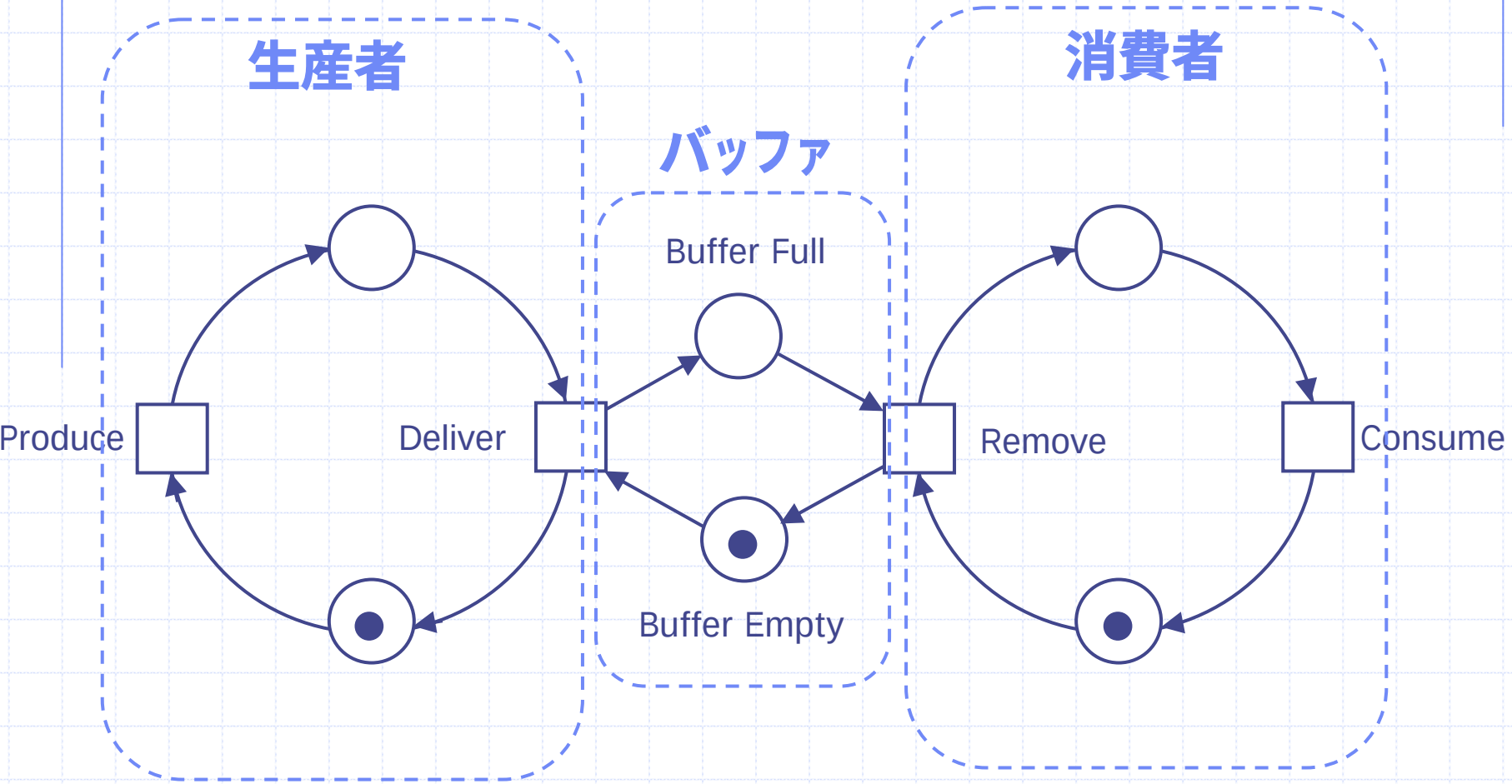
チュートリアルの内容

1. ペトリネットとは？
   - 例：プレース／トランジションネット（ペトリネット）
   - 有限オートマトンとの比較
   - プロセス代数との比較
2. 解析手法
   - 線形代数的手法
   - サブクラス
   - 縮約法
   - 到達可能木，被覆木
   - 時相論理とモデル検査
   - 効率化手法
3. 拡張モデル
   - 時間／確率ペトリネット
   - 高水準ペトリネット
   - ファジイペトリネット
   - ハイブリッドペトリネット
   - オブジェクトペトリネット

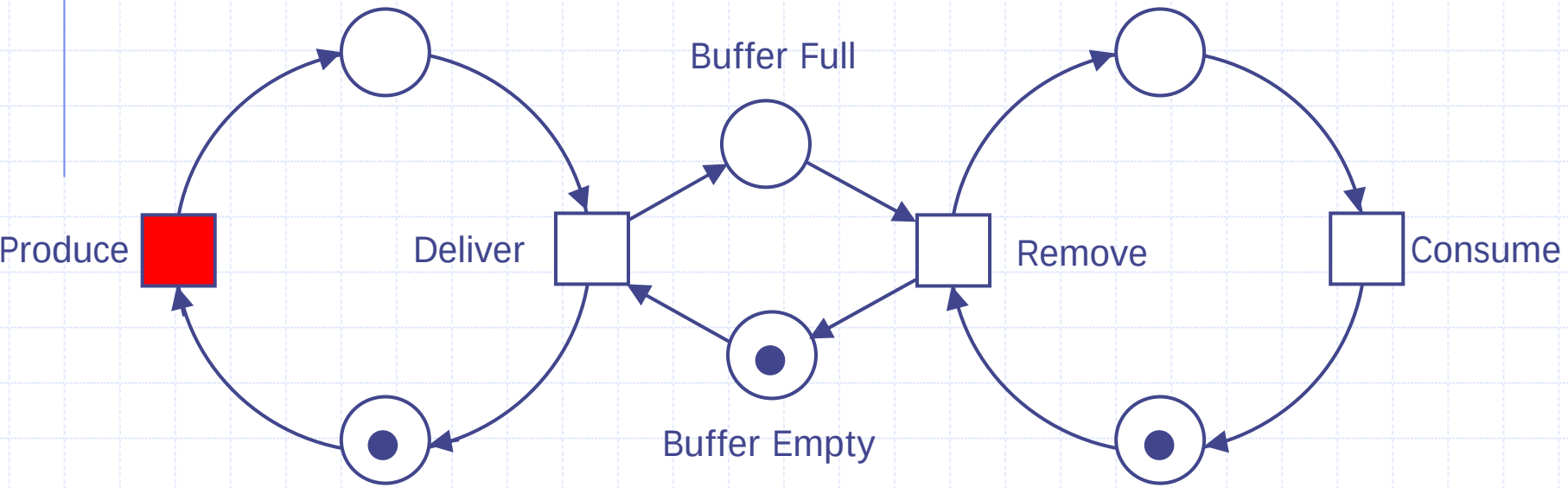
例:生産者・消費者システム



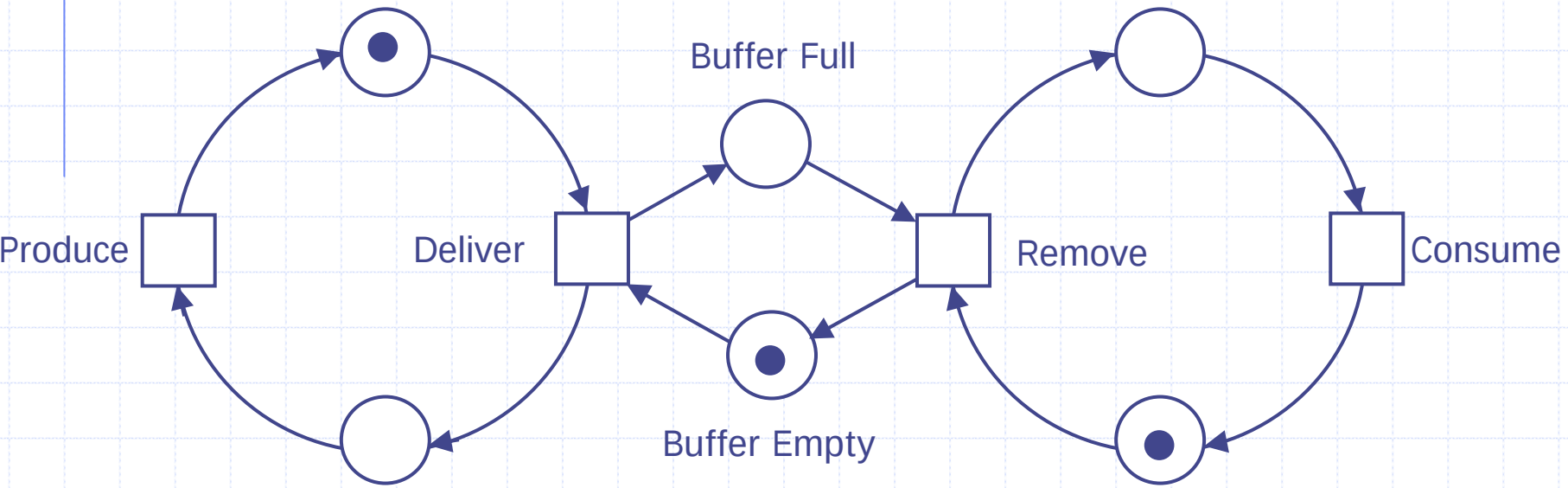
例:生産者・消費者システム



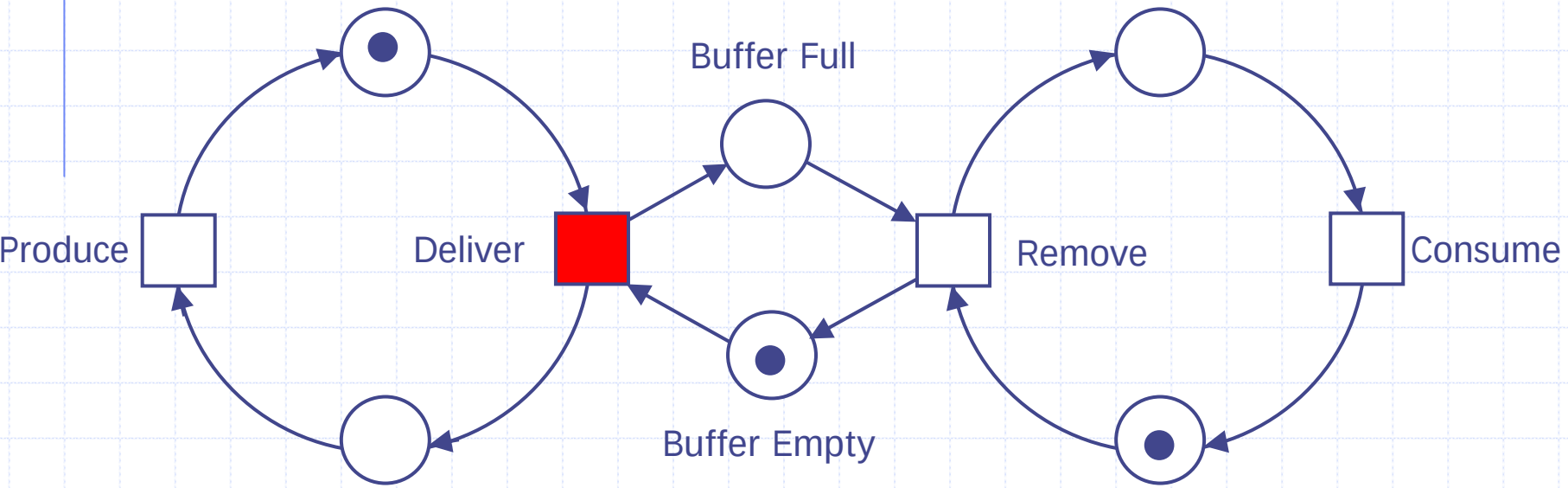
例：生産者・消費者システム



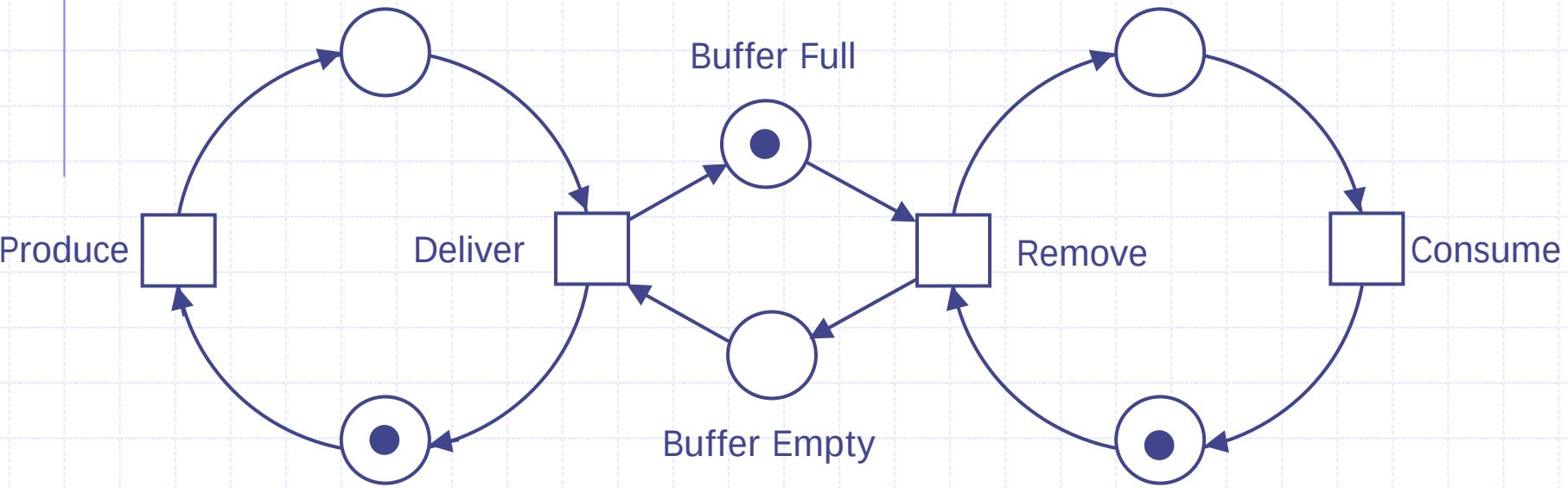
例：生産者・消費者システム



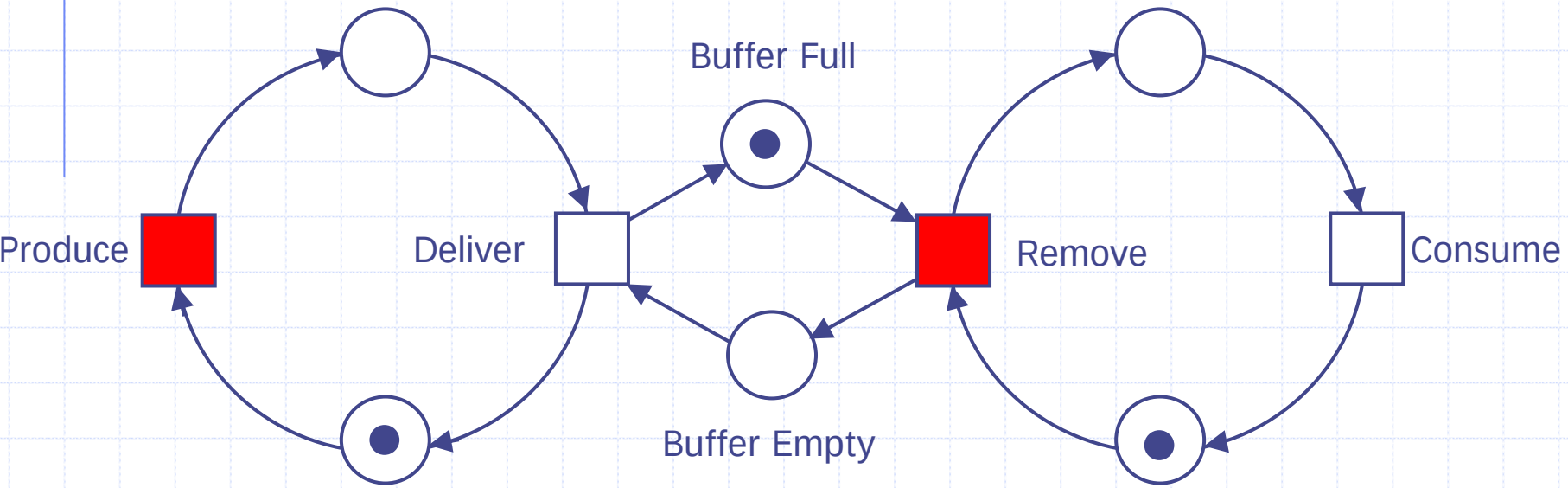
例：生産者・消費者システム



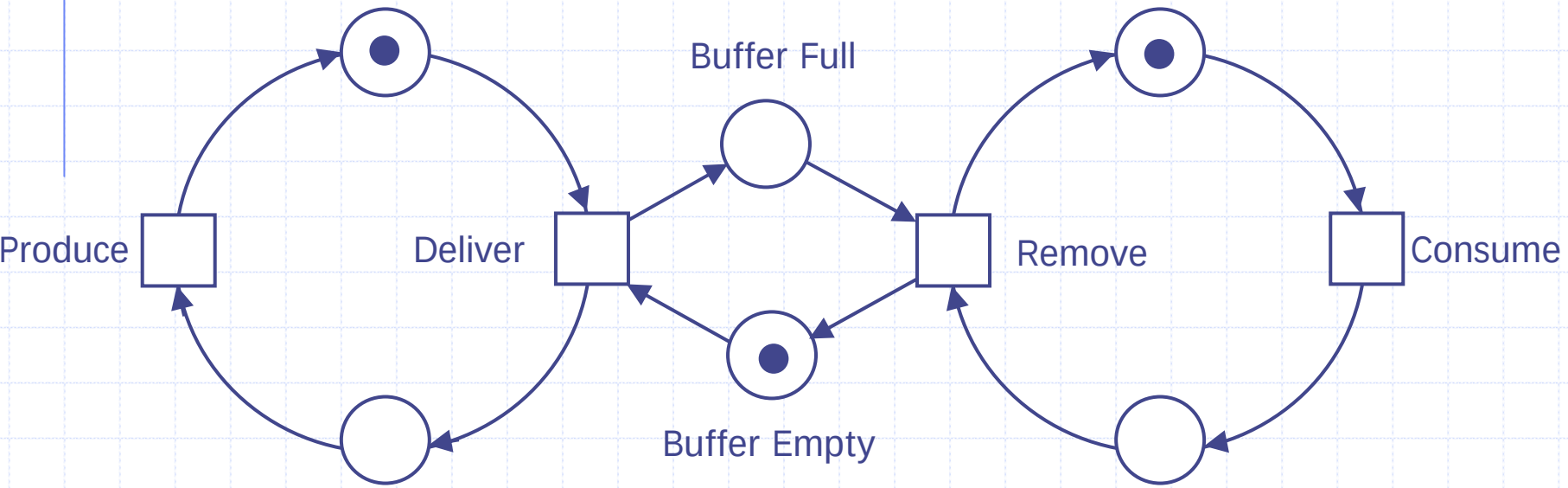
例：生産者・消費者システム



例：生産者・消費者システム



例：生産者・消費者システム



グラフ構造

- ペトリネットは2種類の頂点“□”（トランジション, transition）と“◯”（プレース, place）からなる有向2部グラフ．
- さらに，プレース“◯”の中に“•”（トークン, token）が置かれている．

状態表現：局所的状態

- プレースはそこに入るトークンの個数によりシステムの局所的状態を表現する．
- 1つのトークンは1つの資源の存在に対応する．あるプレースにトークンが k 個置かれていれば，k 個の資源がそこに存在することを意味する．
  - 生産者および消費者はそれぞれ1つのトークンで表されている．
  - バッファの空き領域も1つの資源である．
- 同じプレースに複数のトークンがあった場合，それらは区別されない．個数のみが意味をもつ．

状態表現：大域的状態

- ペトリネット全体でのトークンの配置をマーキング(marking)という．
- マーキングはプレース数の次元の非負整数ベクトル

$$m = [m(p_1), m(p_2), \ldots, m(p_n)]$$

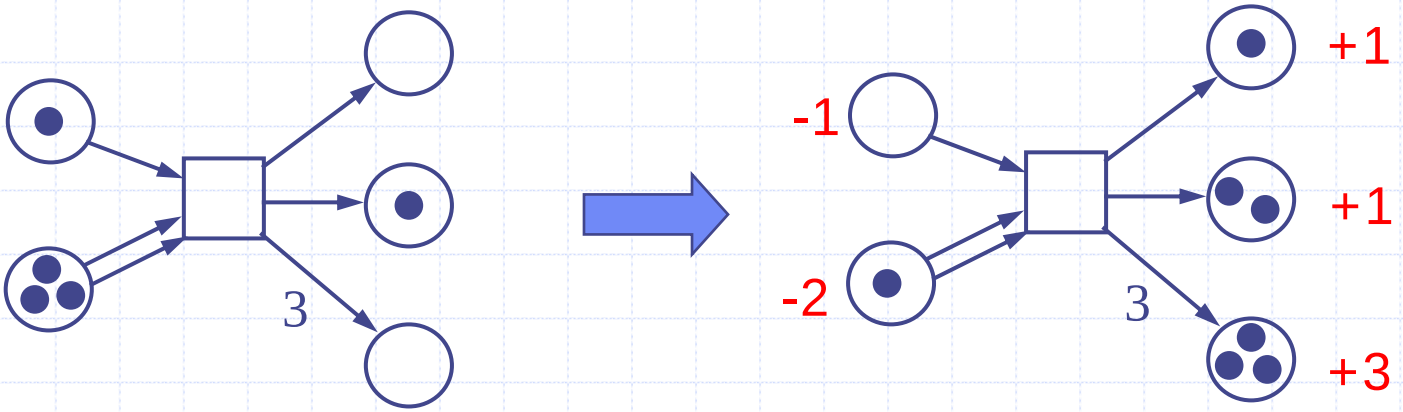
で表現される．ここで，$m(p_i)$ はプレース p_i のトークン数である．
- マーキングはシステムの大域的状態を表す．

状態遷移規則

- トランジションは局所的状態に基づく状態遷移規則を表現する．
- トランジションは各入力プレースに矢印（アーク，arc）の本数*以上のトークンが存在するとき発火可能であるという．

  *1本の矢印＋整数値で表すこともある．
- トランジションの発火により，各入力プレースから矢印の本数だけのトークンが失われ，出力プレースに矢印の本数だけのトークンが加えられる．

トランジションの発火



$$\begin{bmatrix}1\\3\\0\\1\\0\end{bmatrix} - \begin{bmatrix}1\\2\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\\1\\3\end{bmatrix} = \begin{bmatrix}0\\1\\1\\2\\3\end{bmatrix}$$

有限オートマトンとの違い

- 無限状態の表現.
- 合成の容易さ.
- 資源数に対する変更の容易さ.
- 並行性の表現.

無限状態の表現

$$[0] \rightarrow [1] \rightarrow [2] \rightarrow [3] \rightarrow \cdots$$

無限状態の表現

$$[0] \rightarrow [1] \rightarrow [2] \rightarrow [3] \rightarrow \cdots$$

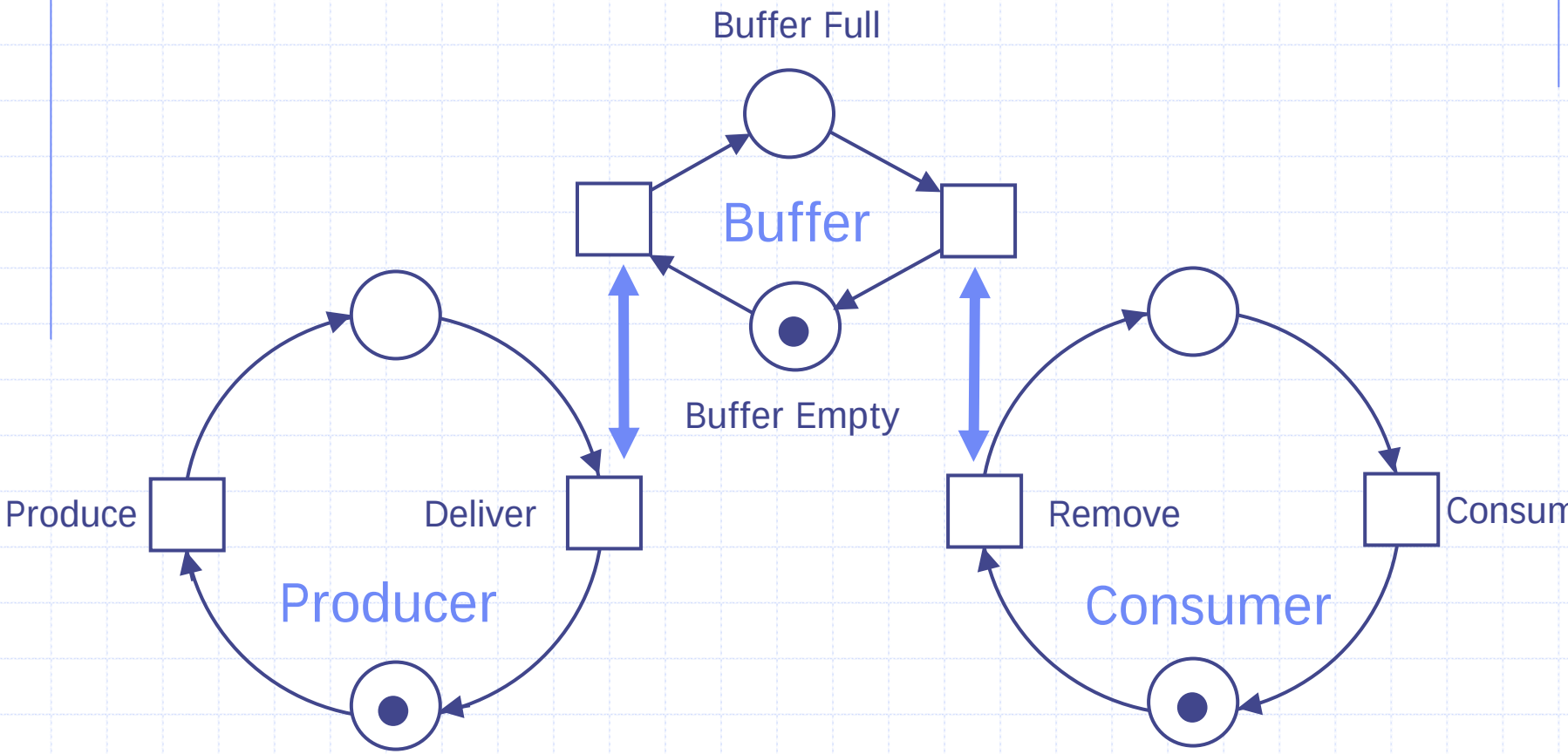
無限状態の表現

$$[0] \rightarrow [1] \rightarrow [2] \rightarrow [3] \rightarrow \cdots$$

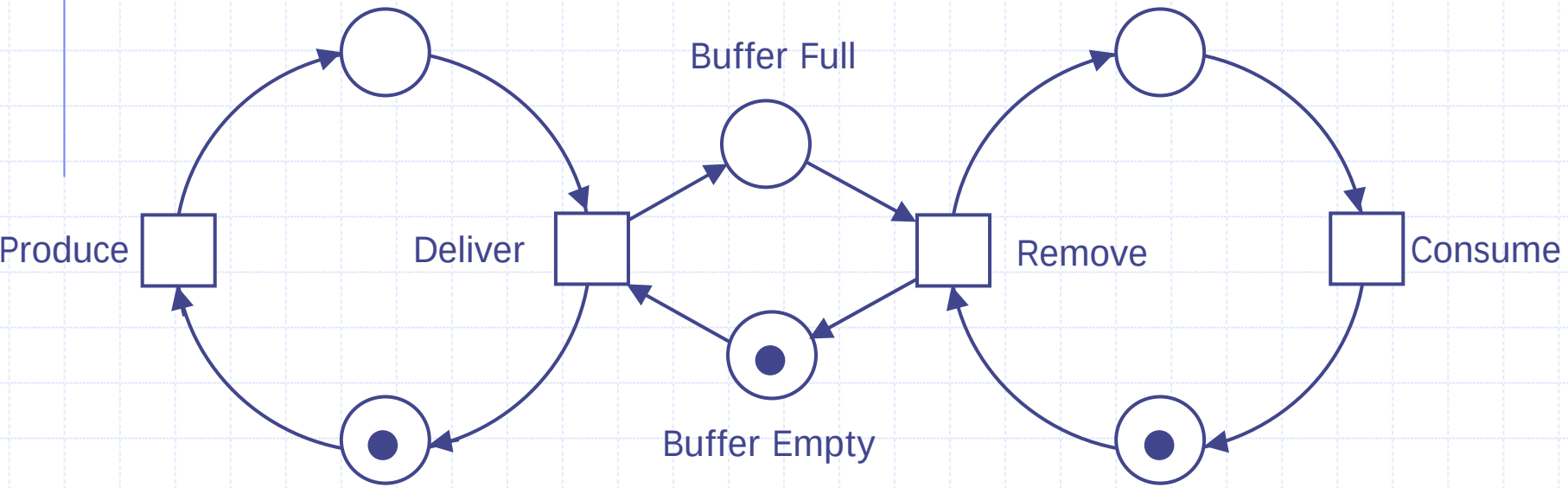
無限状態の表現

$$[0] \rightarrow [1] \rightarrow [2] \rightarrow [3] \rightarrow \cdots$$

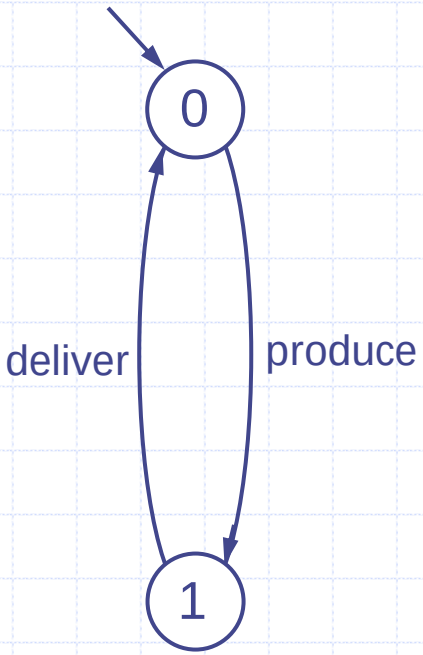
合成：ペトリネット


Buffer Full
Buffer
Buffer Empty
Produce
Deliver
Producer
Remove
Consum
Consumer

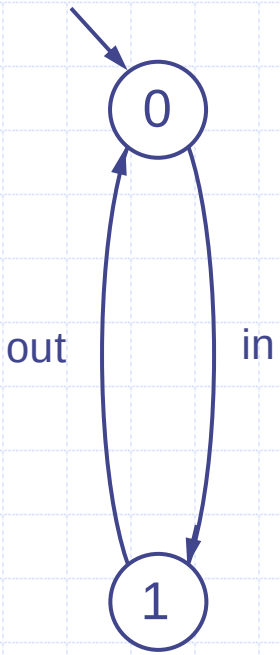
合成：ペトリネット



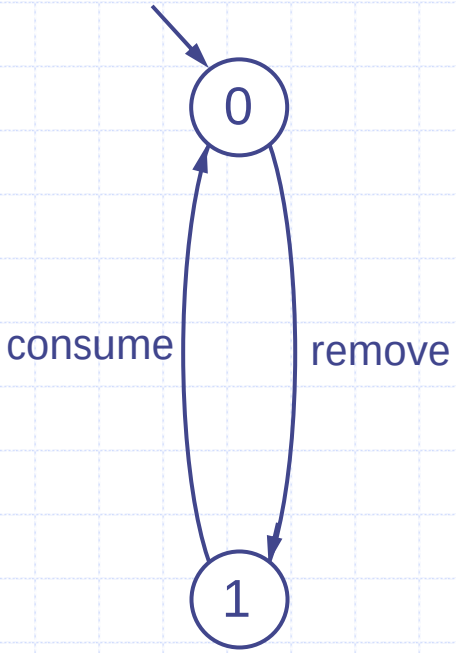
合成：有限オートマトン



Producer



Buffer



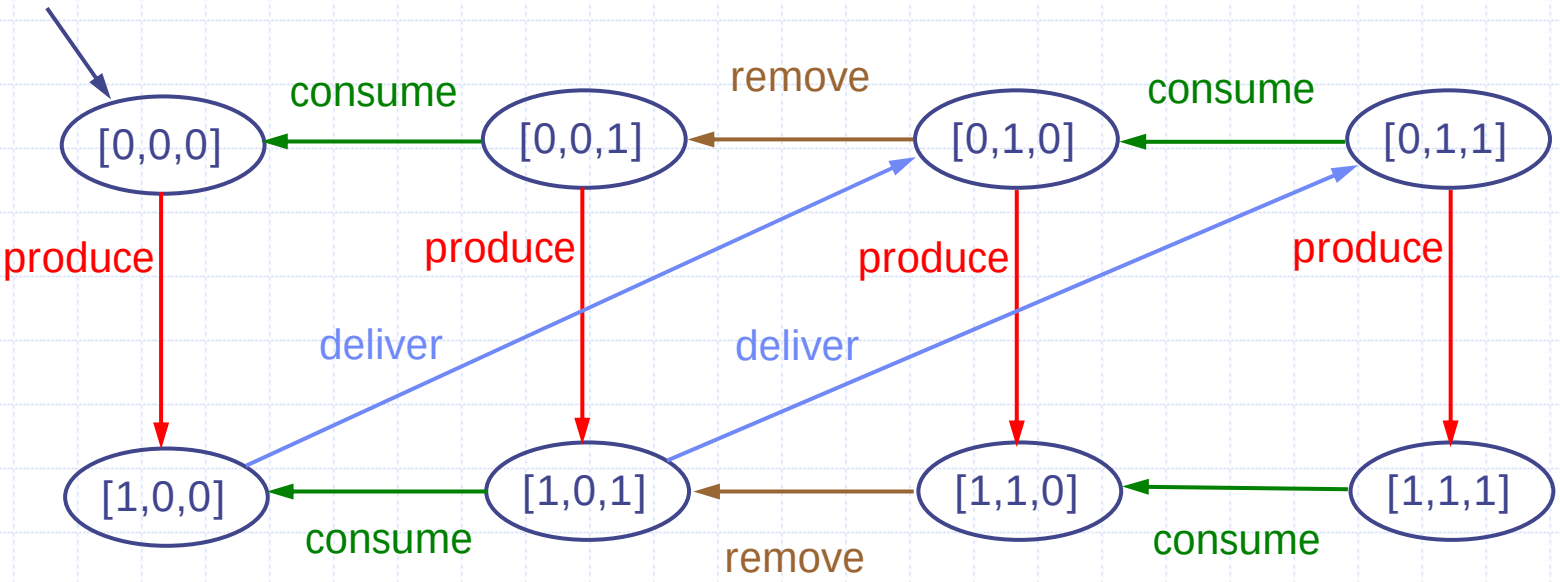
Consumer

合成：有限オートマトン

(Producer || Buffer || Consumer) / {produce/in, consume/out}

合成：有限オートマトン

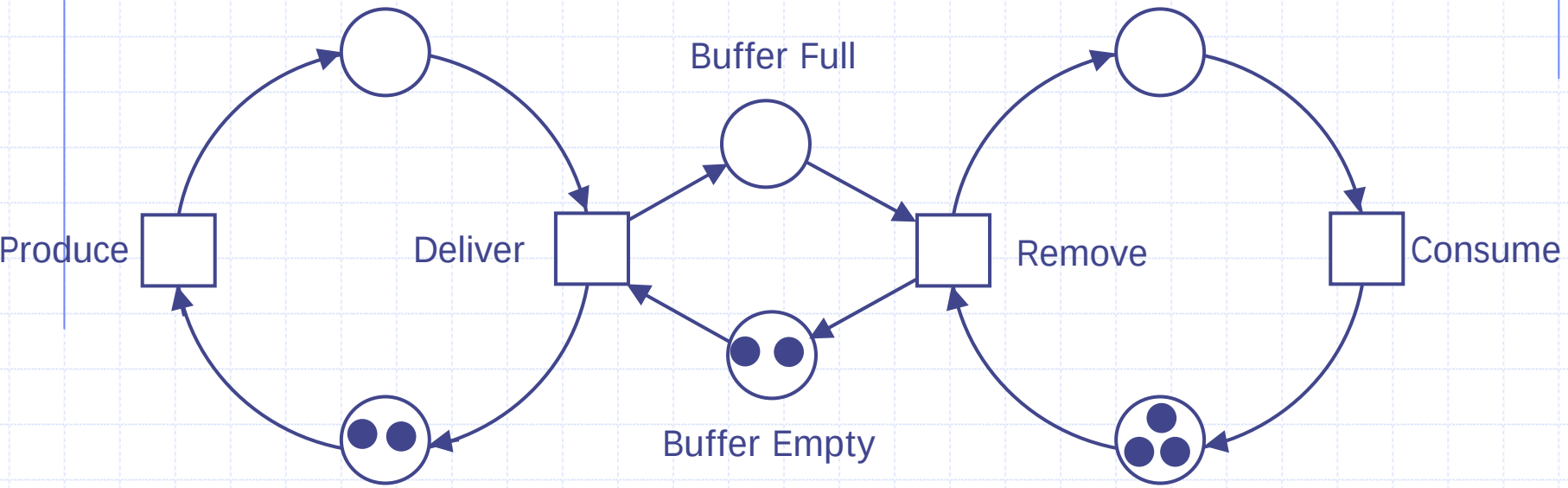
(Producer || Buffer || Consumer) / {produce/in, consume/out}



合成：有限オートマトン

- 合成されたオートマトンの状態数は$2 \times 2 \times 2 = 8$である．
- 合成されたオートマトンのサイズは最大で各オートマトンの状態数の積になるのに対し，ペトリネットの表現では各コンポーネントの和にしかならない．
- ペトリネットのサイズがオートマトンに比べて小さくてすむのは状態表現の方法のためである．
- 例では6個のプレースがあるが，それぞれたかだか1個のトークンしか入らないとしても，トークンの配置は2^6通りある（実際は，初期状態から到達可能なのは8状態のみである）．

資源数の変更



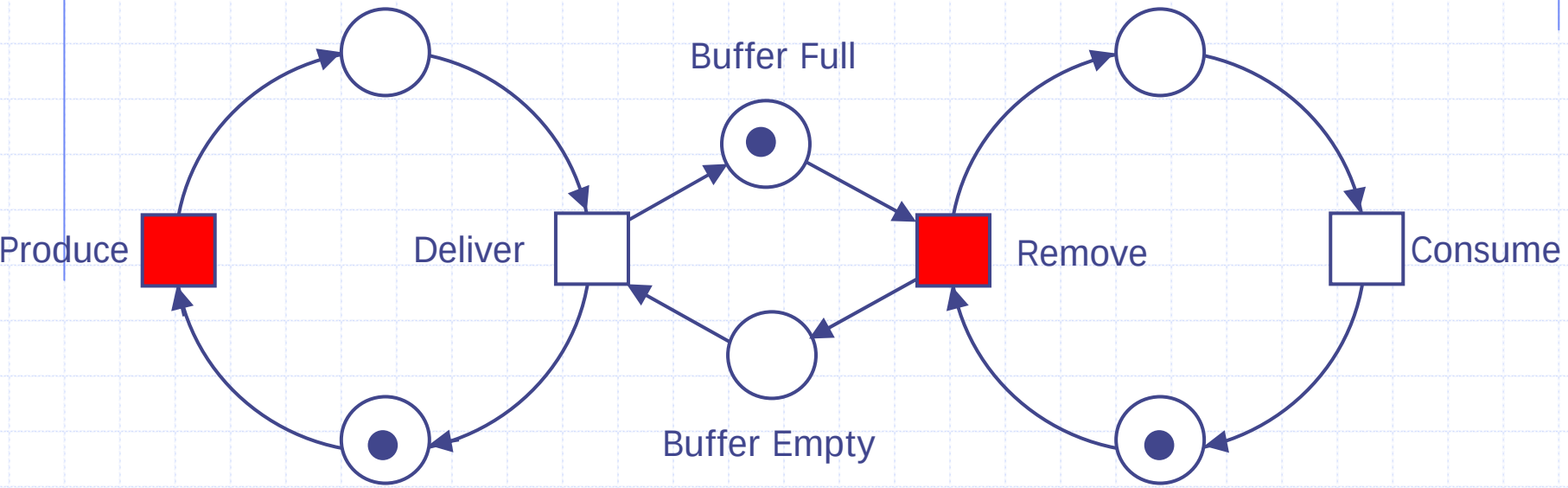
生産者2，消費者3，バッファ容量2の場合

同じ例をオートマトンで表現すると状態数は96になる．

並行性の表現：ペトリネット

- 生産者・消費者システムのペトリネットにおいて，2つの動作 produce と remove はそれらがそれぞれ実行可能ならば，独立して実行可能である．ここで“独立”の意味は，一方の動作の実行が他方の動作の実行可能性に影響を与えないことである．
- 独立な動作はそれらに対応するトランジションが**競合**しないことで表現される．
  - 競合しない条件：2つのトランジションが共通の入力プレースを持たない，あるいは，持ったとしても両方のトランジションの発火に必要な十分の数のトークンが存在する．

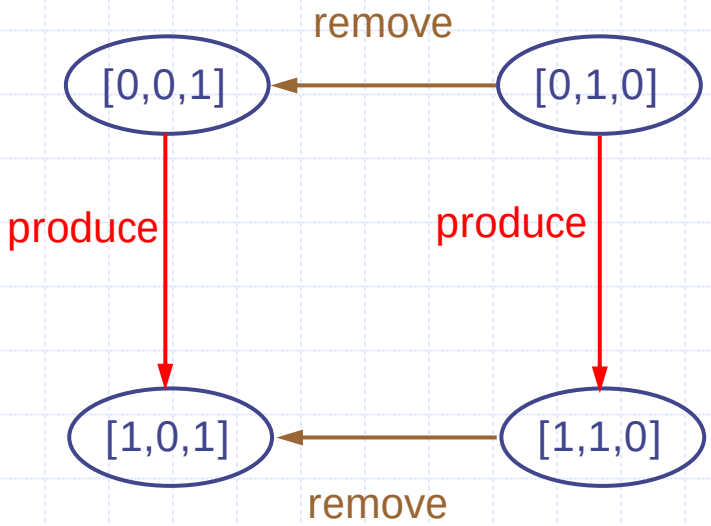
並行性の表現：ペトリネット



produce と remove の発火は独立

並行性の表現：オートマトン

状態[0,1,0]から produce remove, remove produce のどちらも起こり得る．



独立な系列のインターリービングにより並行性が表現される．

プロセス代数との違い

生産者･消費者システム

Producer := produce.deliver.Producer;
Buffer := in.out.Buffer;
Consumer := remove.consume.Consumer;
System := (Producer || Buffer || Consumer)/{deliver/in, remove/out}.

プロセス代数との違い

- プロセス代数の利点：
  - プログラミング言語との親和性．
  - コンポーネントを組み合わせていくような合成が容易．
  - 代数的な手法，あるいは，付随する論理系を用いた推論，証明が可能．
- ペトリネットの利点：
  - 状態（プレース）と遷移規則（トランジション）の分離．
  - 局所的状態および遷移規則のみの記述＝大域的状態および遷移規則は局所的状態および遷移規則から導かれる．
  - 図的言語．
  - グラフ理論，線形代数による解析．
- 両者を融合させた研究：Petri Net Algebra．

チュートリアルの内容

1. ペトリネットとは？
   - 例：プレース／トランジションネット（ペトリネット）
   - 有限オートマトンとの比較
   - プロセス代数との比較
2. 解析手法
   - 線形代数的手法
   - サブクラス
   - 縮約法
   - 到達可能木，被覆木
   - 時相論理とモデル検査
   - 効率化手法
3. 拡張モデル
   - 時間／確率ペトリネット
   - 高水準ペトリネット
   - ファジイペトリネット
   - ハイブリッドペトリネット
   - オブジェクトペトリネット

ペトリネットの理論的解析

- ペトリネットの解析はつぎの2種類に分類できる.
  - グラフ的構造を利用した静的な解析
  - どのようなふるまいをするかを調べる動的な解析
- 静的解析を行う様々な手法が存在することがペトリネットの特徴.
- 主に扱われるのは, 記述されたモデルがある性質を満たすかどうかを判定する, あるいはその性質を満たすようにペトリネットを合成する問題である.

解析される主な性質

- 到達可能性(reachability)：初期マーキングからある与えられたマーキングがトランジションの発火により到達できる．他の多くの問題が到達可能性を判定する問題に帰着可能であり，その意味でペトリネットの解析において最も基本的な問題の1つである．
- 活性(liveness)：初期マーキングから到達可能な任意のマーキングにおいて，そこから各トランジションが発火可能なマーキングに到達できる．
- 有界性(boundedness)：各プレースに入り得るトークン数に上限が存在する．上限を1とした有界性を特に安全性(safeness)という．
- 保存性(conservativity)：重み付きのトークン数の和が一定．
- 公平性(fairness)：継続的に発火可能なトランジションはいつかは必ず発火する．

線形代数的手法：行列表現

- P/Tネットの有向2部グラフとしての構造は，2つの行列

$$A^- = [a^-_{ij}],\ A^+ = [a^+_{ij}]$$

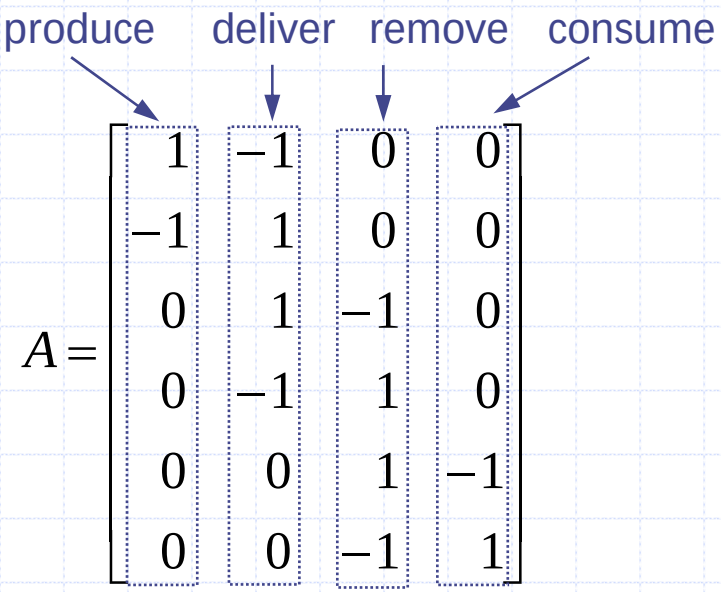
により記述できる．ここで，

  - a^-_{ij} はプレース p_i からトランジション t_j へのアークの本数，
  - a^+_{ij} はトランジション t_j からプレース p_i へのアークの本数

である．

- $A = A^+ - A^-$ をペトリネットの**接続行列**という．

行列表現：例



生産者・消費者システム

線形代数的手法：状態方程式

- トランジションの有限系列 $\sigma = t^{(1)}t^{(2)}\ldots t^{(k)}$ に対し，各トランジション t_i が σ に出現する回数を第i成分にもつ非負整数ベクトルを σ の発火回数ベクトルといい，$\psi(\sigma)$で表す．
- マーキング $m^{(0)}$ からトランジションの系列 σ の発火によりマーキング $m^{(k)}$ に到達するとき，つぎの等式が成り立つ．

$$m^{(k)} = m^{(0)} + A\psi(\sigma).$$

これを状態方程式という．

- マーキング m がマーキング m_0 から到達可能ならば，$m = m_0 + Ax$ を満たす非負整数ベクトル x が存在する（**到達可能性の必要条件**）．

状態方程式：例

初期マーキングから

σ = produce, deliver, remove, produce, consume, deliver

と発火させたときの状態方程式

$$\underset{m}{\begin{bmatrix}0\\1\\1\\0\\0\\1\end{bmatrix}} = \underset{m_0}{\begin{bmatrix}0\\1\\0\\1\\0\\1\end{bmatrix}} + \underbrace{\begin{bmatrix}1 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 0 & 1 & -1 & 0\\ 0 & -1 & 1 & 0\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix}2\\2\\1\\1\end{bmatrix}}_{\psi(\sigma)}$$

線形代数的手法：インバリアント

- インバリアント (invariant) とは，一般にシステムの動作に対し不変であるような量，値，性質のことである．
- インバリアントを用いてペトリネットの様々な性質を解析できる．
- ペトリネットでは2種類のインバリアントが用いられる．
  - $Ax = 0$ を満たす非負整数ベクトルを *T*ーインバリアントといい，マーキングを変化させないような発火系列の発火回数ベクトルに対応する．定常的動作をするような有限状態システムには必ず *T*ーインバリアントが存在する．
  - $A^t y = 0$ を満たす (非負) 整数ベクトルを *P*ーインバリアントという．*P*ーインバリアントにより重み付けされたトークン数の合計が任意のトランジションの発火により不変である (到達可能性の必要条件)．

インバリアント：例

$[1,1,1,1]^t$ は T-インバリアント.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 0.$$

発火系列 σ = produce, deliver, remove, consume は状態を変えない.

インバリアント：例

[1,1,0,0,0,0] は*P*-インバリアント

$$\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}1 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 0 & 1 & -1 & 0\\ 0 & -1 & 1 & 0\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}=0.$$

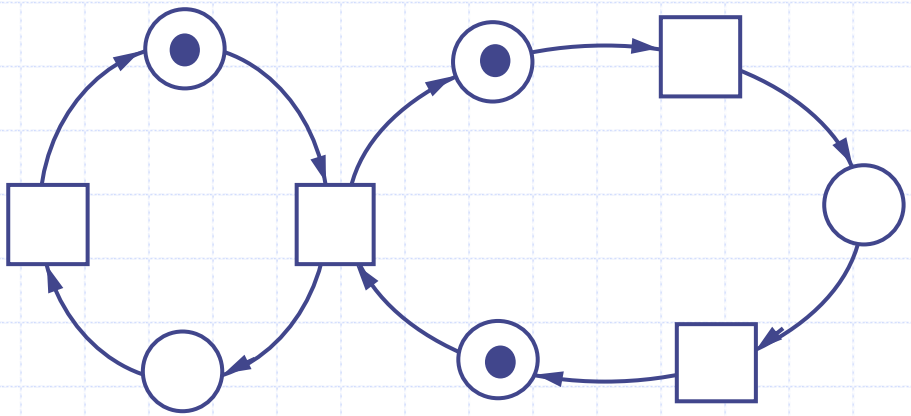
生産者数は不変.

サブクラス

- ペトリネットのグラフ的構造を制限したいくつかのサブクラスに対しては，到達可能性あるいは活性の必要十分条件が求められている．
- このようなクラス：
  - マークグラフ
  - 状態機械
  - 無競合ネット
  - 自由選択ネット
  - ・・・

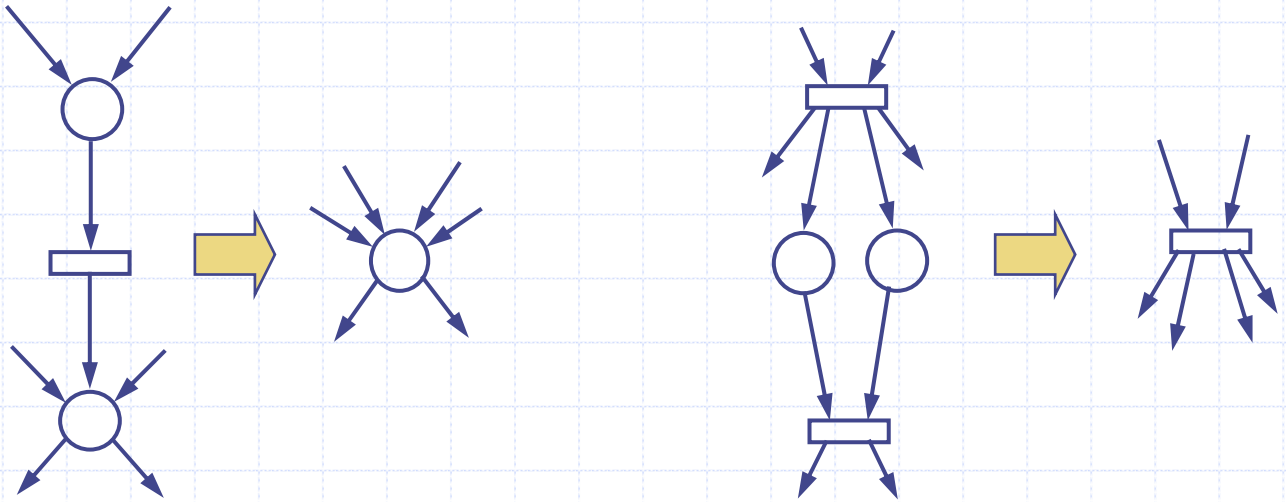
サブクラス：マークグラフ

- マークグラフ(Marked Graph)：各プレースの入出力アーク数がそれぞれ1．
- どのような発火を行っても有向閉路内のトークン数の総和が変化しない．これを利用して，到達可能性および活性の必要十分条件が求められる．



縮約

- ある性質を保存したまま，より小さいサイズのペトリネットに変換する手法がある．解析に要する計算量を減らすのに有効な方法である．
- 例：活性を保存する縮約方法（一部）：



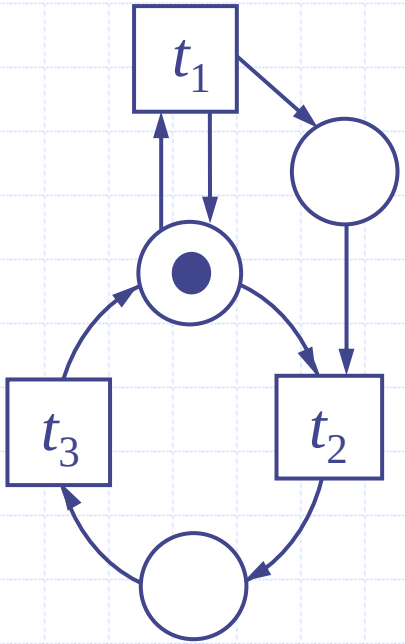
到達可能木

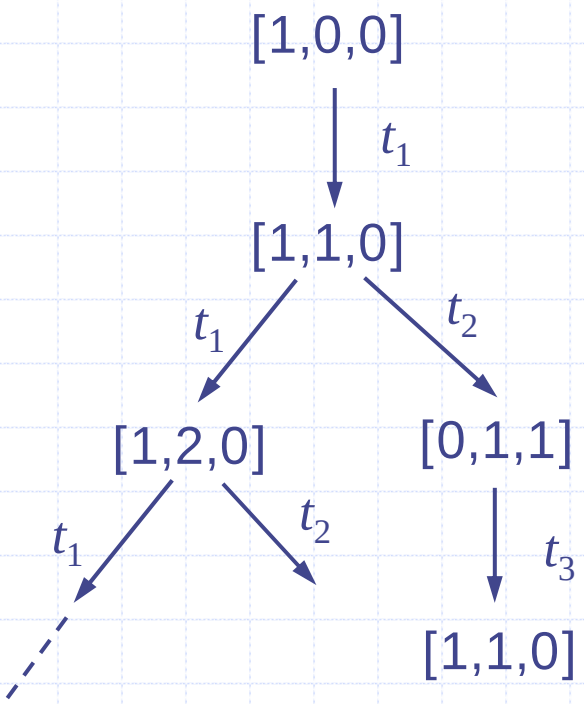
- 到達可能木 (reachability tree)：ペトリネットの状態空間を表現する一方法．初期マーキングを根とし，到達可能なマーキングをアークで結ばれた各節点に配置した木構造で状態空間を表現する．
- 同一のマーキングをもつ節点を1つにまとめることにより状態遷移図が得られる．
- 非有界なペトリネットの到達可能木は無限の大きさをもつ．非有界なペトリネットに対して到達可能木を実効的な解析方法とするためにはその大きさを有限にしなければならない．

被覆木

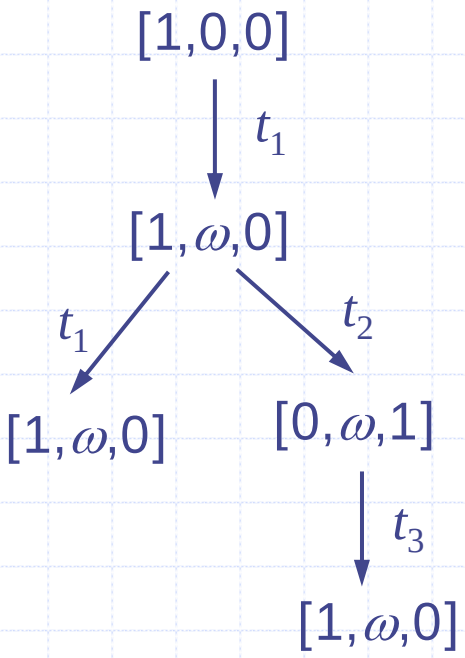
- 被覆木(coverability tree)：無限を表す記号 ω を用いることにより到達可能木を有限の大きさで表現する方法.
- 被覆木により有界性や保存性が確認できる.
- 到達可能性については記号 ω の導入による情報の欠落により調べることはできない.

到達可能木，被覆木：例





到達可能木



被覆木

モデル検査

- 有限状態システムに対し，満たすべき性質を時相論理(temporal logic)で記述し，それがモデル上で満たされるかどうかを検証アルゴリズムにより自動的に判定する方法をモデル検査(model checking)という．
- 特に，モデル上での状態探索をせずに，論理式の計算を2分決定グラフ(BDD)を用いて行うことで検証を行う記号モデル検査法(symbolic model checking method)の開発により，かなり大規模のシステムに対しても検証が可能になった．
- 記号モデル検査法はもともと順序回路の検証用に開発された手法であるが，有界なペトリネットに対しても適用可能である．

時相論理：例

- 活性を分岐時間時相論理の1つであるCTL (computation tree logic) で記述するとつぎのようになる．

$$\mathrm{AGEF}\, EN_t\,.$$

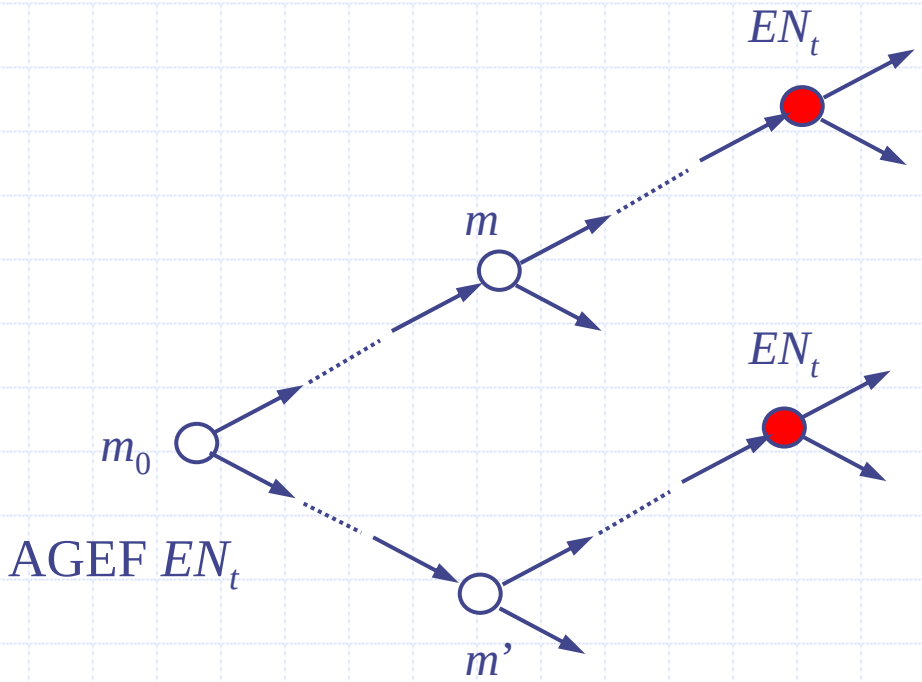
- EN_t を各マーキングにおいてトランジション t が発火可能であるかどうかを表す原子命題とする．
- 時相オペレータ A, E, G, F はつぎの意味をもつ．
  - A：その状態からのすべてのパスについて
  - E：その状態からのあるパスについて
  - G：そのパス上のすべての状態について
  - F：そのパス上のある状態について

時相論理：例

AGEF EN_t は，

「現在の状態からどのような状態遷移をしても，その状態から将来トランジション t を発火させることができる」

というトランジション t に関する活性を表現する．



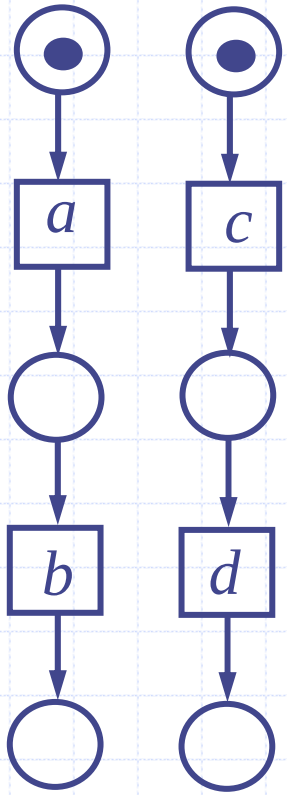
効率化手法：状態空間爆発

- ペトリネットのサイズに対して，初期マーキングから到達可能なマーキング数が指数関数オーダーで増加する現象を状態空間爆発(state spece explosion)という．
- 逆に言えば，ペトリネットは膨大な数の状態空間を非常に小さい記述でモデル化できることを意味する．
- 静的解析だけではシステムのふるまいに関するすべてのふるまいを解析することはできない．システムの厳密な検証を行うときには状態空間爆発は避けては通れない問題である．
- なお，この問題はペトリネット固有の問題ではなく，並行システムを記述する形式的モデルの解析すべてに共通する問題である．

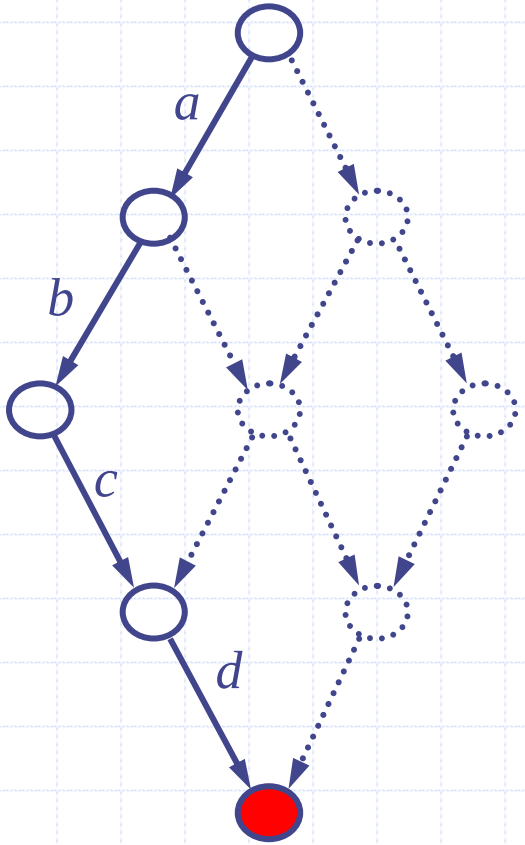
効率化手法：半順序法

- 状態空間爆発が起こる理由の1つとして，並行動作により多数の状態が生成されることが挙げられる．
  - 並行動作はどのような順序でも起こり得るため，それらの順列の個数だけ可能な動作系列および中間状態が存在する．
  - このような動作系列すべてを考慮しなくても，たとえばデッドな状態（どのトランジションも発火不能な状態）に到達するかどうかは判定できる．
- 時相論理により記述された性質に対し，その判定に必要ない並行動作を考慮しないことにより，状態空間の一部のみを生成し，検証を行う効率的に行う手法が開発されており，半順序法(partial order method)と呼ばれる．

半順序法：例



デッドな状態に到達するか？

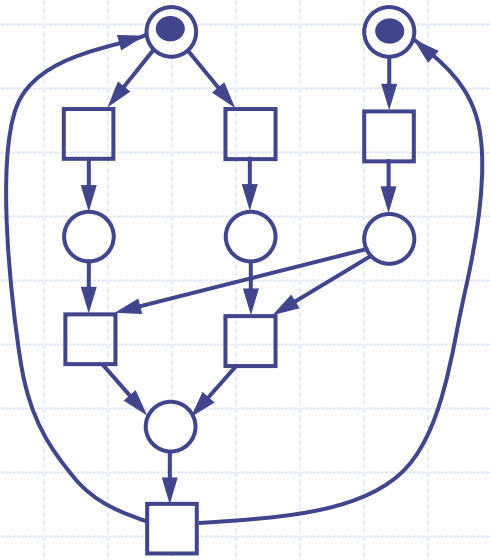


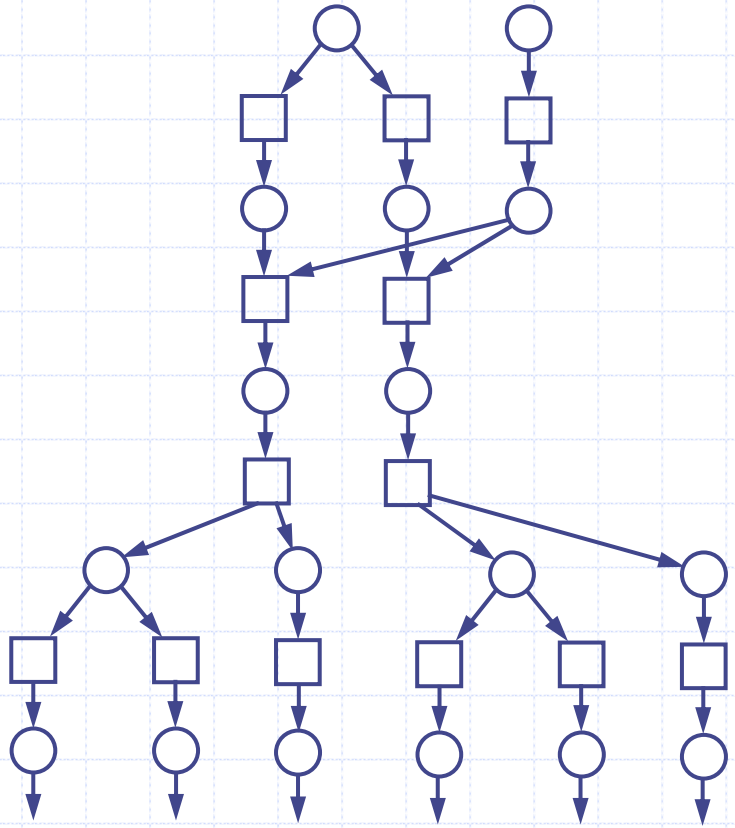
縮約状態空間

効率化手法：アンフォールディング

- 状態空間表現として状態遷移図のように逐次的ふるまいとして記述するのではなく，半順序関係として記述する方法としてアンフォールディング(unfolding)がある．
- アンフォールディングは，ペトリネットを初期マーキングから発火可能なトランジションに沿って展開していったものである．
- アンフォールディングは後ろ向きに無競合な非循環ペトリネットであり，無限の大きさをもつ．
- ある点（カットオフ点）で展開を停止する．このとき，そこまでの有限部分に元のペトリネットで到達可能なマーキングがすべて含まれるための，カットオフ点が満たすべき条件が知られている．

アンフォールディング：例





カットオフ点

チュートリアルの内容

1. ペトリネットとは？
   - 例：プレース／トランジションネット（ペトリネット）
   - 有限オートマトンとの比較
   - プロセス代数との比較
2. 解析手法
   - 線形代数的手法
   - サブクラス
   - 縮約法
   - 到達可能木，被覆木
   - 時相論理とモデル検査
   - 効率化手法
3. 拡張モデル
   - 時間／確率ペトリネット
   - 高水準ペトリネット
   - ファジイペトリネット
   - ハイブリッドペトリネット
   - オブジェクトペトリネット

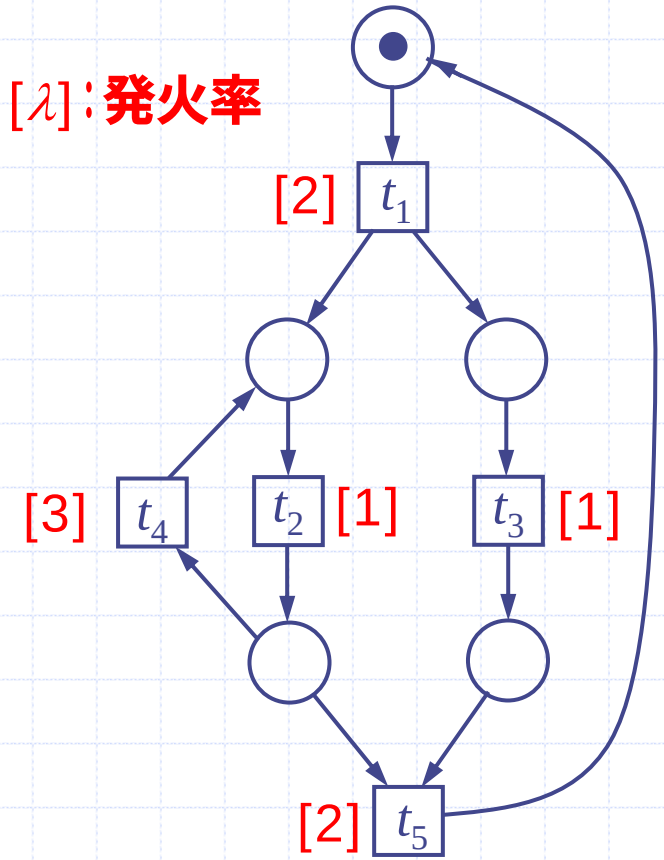
時間ペトリネット，タイムペトリネット

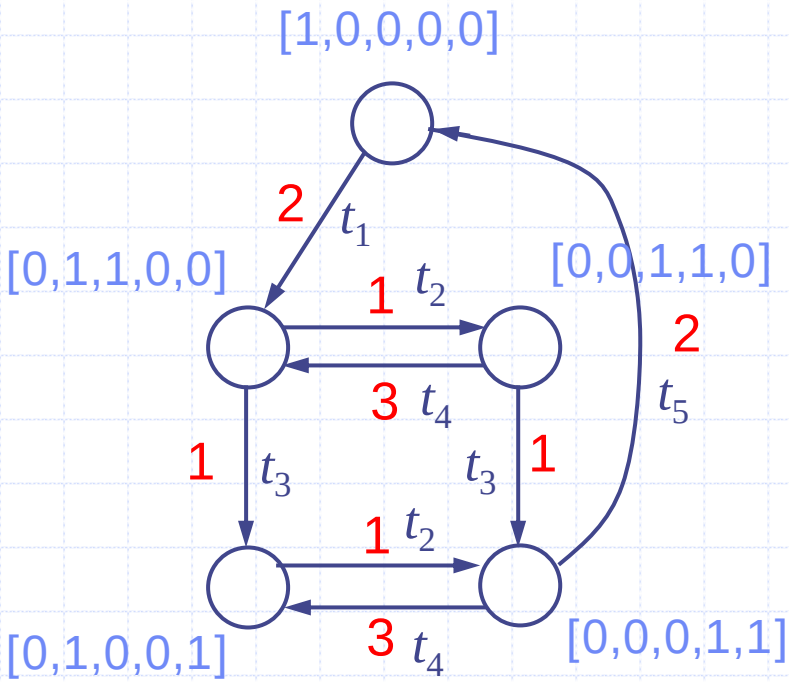
- P/Tネットに確定的な時間の概念を導入したのが時間ペトリネット(timed Petri net)である．時間の導入の方法により以下のように分類できる．
  - 発火遅延時間モデル：各トランジションに対し，そのトランジションが発火可能になってから実際に発火するまでの遅延時間を表す数値を付随させる．
  - 発火継続時間モデル：各トランジションに対し発火継続時間を付随させる．
  - プレース時間モデル：各プレースに対しそれが発火に利用できるようになるまでの遅延時間を付随させる．
- 各トランジションに対し，発火可能になってから実際に発火するまでの遅延時間の下限および上限を付随させるモデルをタイムペトリネット(time Petri net)という．

確率ペトリネット

- 時間ペトリネットで導入された時間を確率分布の形で置き換えたモデル.
  - 確率ペトリネット (stochastic Petri net)：各トランジションに対し発火可能となってから実際に発火するまでの遅延時間を指数分布の確率分布で与えたモデル.
  - 一般化確率ペトリネット(generalized stochastic Petri net)：指数分布に従うトランジションと遅延0で発火するトランジションの2種類のトランジションをもつモデルであり，論理的ふるまいと時間的ふるまいの両方を表現できる.
  - 決定性/確率ペトリネット(deterministic/ stochastic Petri net)：指数分布の遅延時間をもつトランジションと，一定値の遅延時間をもつトランジションの2種類のトランジションをもつ.
  - ・・・

確率ペトリネット：例





連続時間マルコフ連鎖

確率ペトリネット：解析

- 発火率が λ であるようなトランジションの発火遅延時間は指数分布 $d(x) = \lambda e^{-\lambda x}$ に従い，その平均値は $1/\lambda$ である．
- 対応する連続時間マルコフ連鎖の推移率行列：

$$Q = \begin{bmatrix} -2 & 2 & 0 & 0 & 0 \\ 0 & -2 & 1 & 1 & 0 \\ 0 & 3 & -4 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & 3 & -5 \end{bmatrix}$$

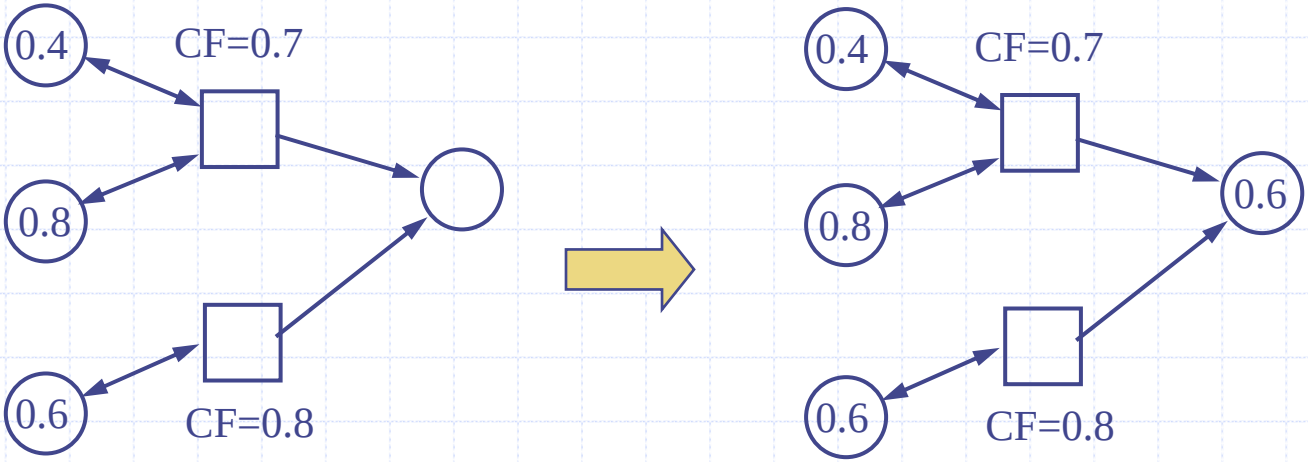
- 平衡状態における各状態の存在確率 $\pi = [\pi_1, \pi_2, \ldots, \pi_5]$ はつぎの方程式を解くことで求められる．

$$\pi Q = 0, \Sigma_i \pi_i = 1.$$

ファジイペトリネット

- ファジイペトリネット(fuzzy Petri net)：不確実性をもつ対象に対する推論を行うルールベース・システムの表現に用いられる．
- 状態遷移規則（一例）：
  - 各プレースは1つの条件に対応しており，区間 $[0, 1]$ の任意の実数値をもつトークンが入る．
  - 各トランジションには確信度を表すファジイ値が与えられる．
  - 発火に伴うマーキングの変化：入力プレースのトークンのファジイ値を $a_1, a_2, \ldots, a_k$，トランジションの確信度を c，出力プレースに既に存在するトークンのファジイ値を b としたとき，トランジションの発火により b は $max(b, min(c, min(a_1, \ldots, a_k)))$ に更新される．

ファジイペトリネット：例

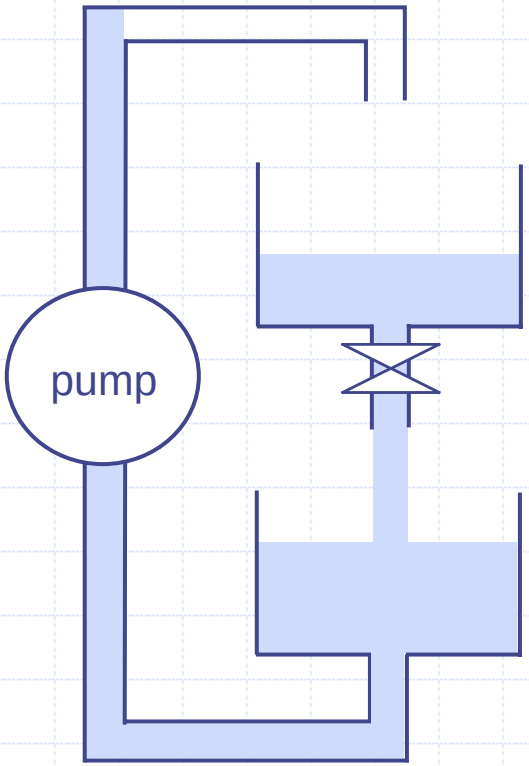


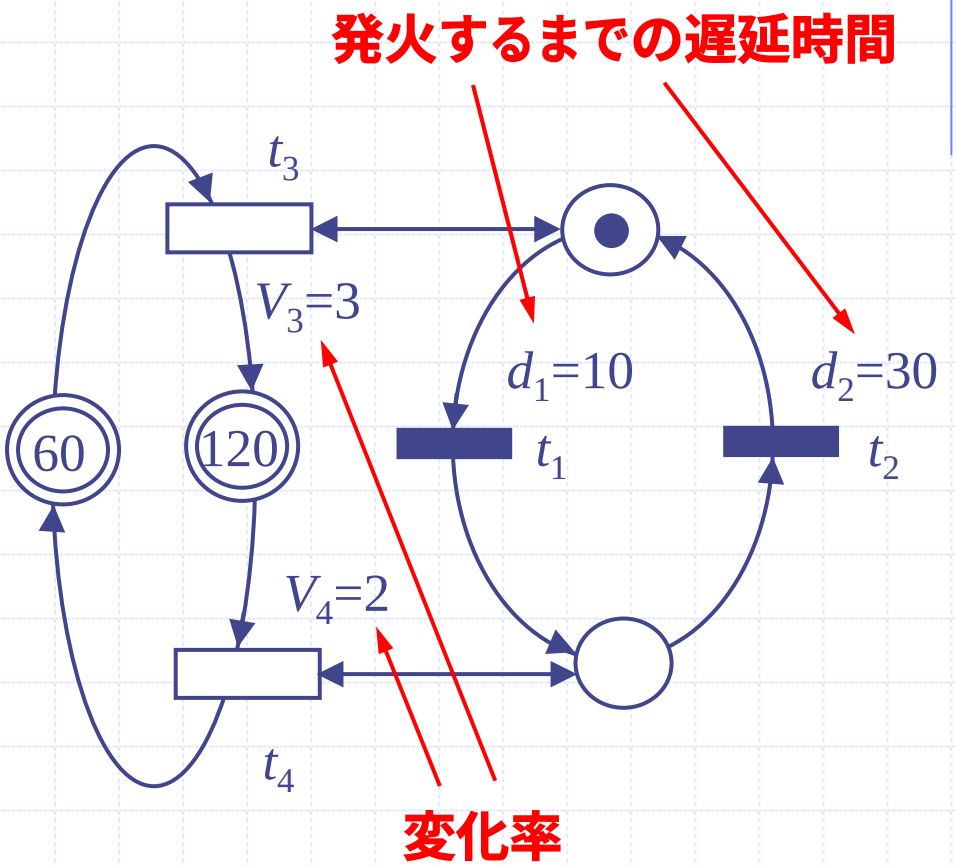
$$0.6 = \max(0, \min(0.7, \min(0.4, 0.8)), \min(0.8, 0.6)).$$

ハイブリッドペトリネット

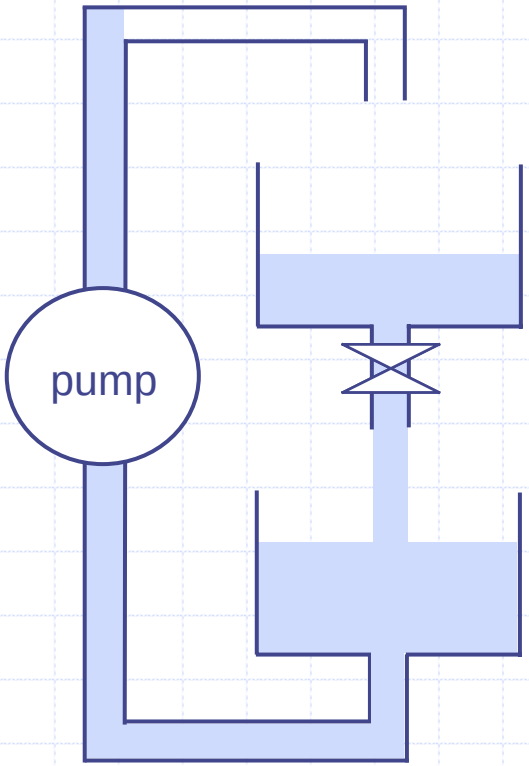
- 連続ペトリネット(continuous Petri net)：連続値をもつプレースとその値を連続的に変化させるトランジションをもつモデル．
- ハイブリッドペトリネット(hybrid Petri net)：連続的状態変化を表現した部分と離散値をもつ通常のペトリネットを融合させたモデル．

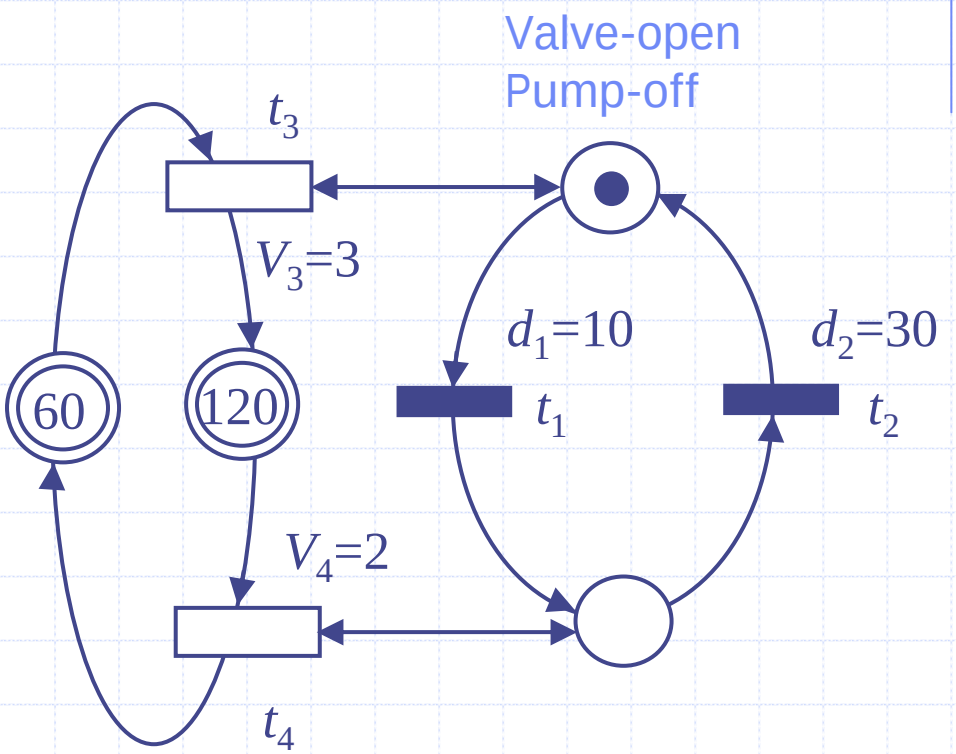
ハイブリッドペトリネット：例



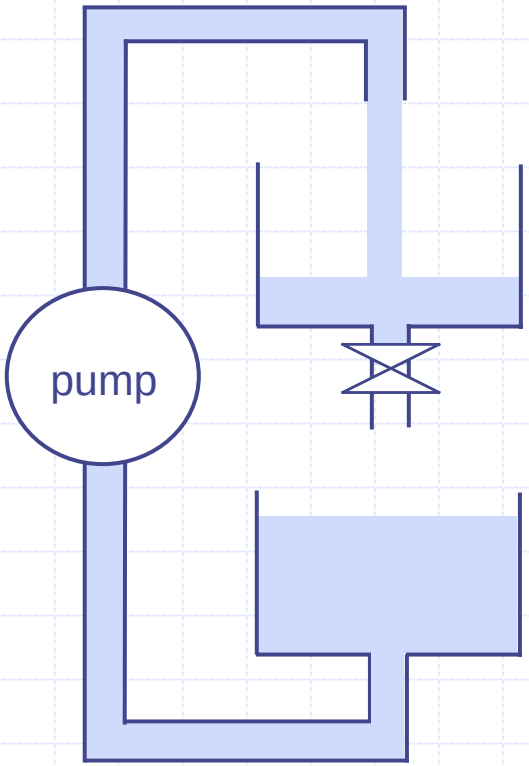


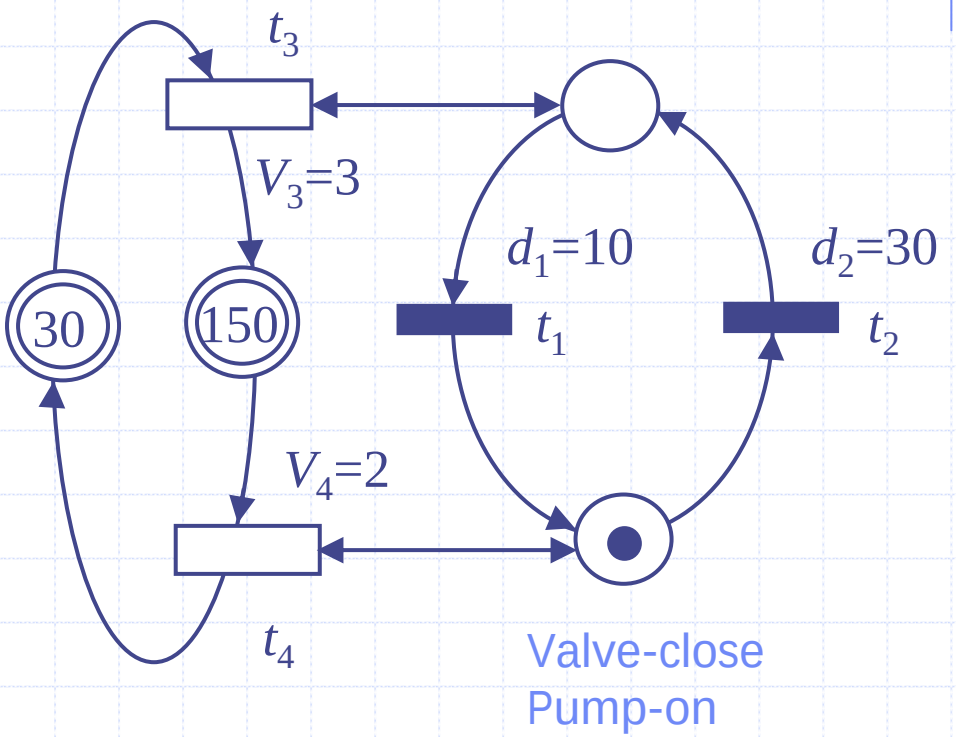
ハイブリッドペトリネット：例





ハイブリッドペトリネット：例

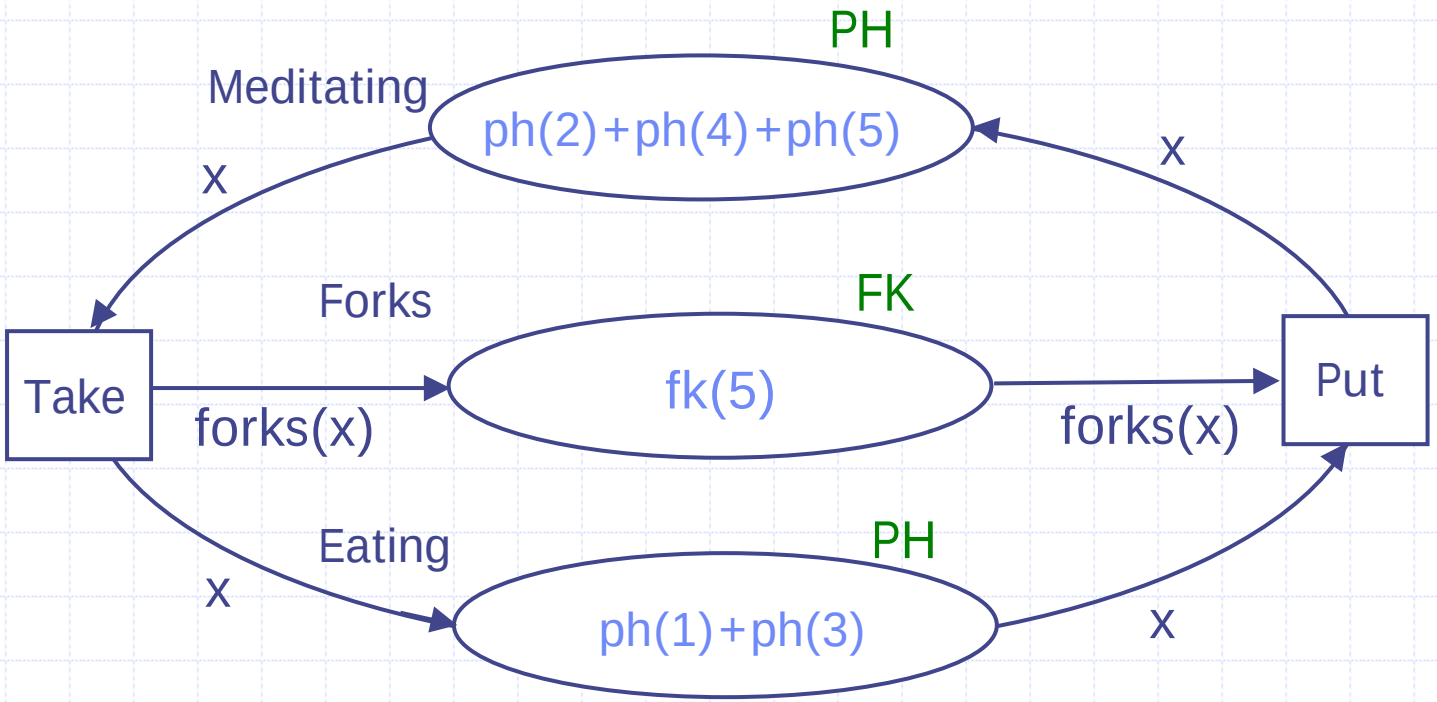




高水準ペトリネット

- 大規模なシステムをモデル化するとき，通常のペトリネットでは同型な構造をもつ部分ネットが数多く生成され，システムを記述する上で大きな負担となる．
- この問題を解消するために提案されたのが述語／トランジションネット(predicate/ transition net)やカラーペトリネット(coloured Petri net)などの高水準ペトリネット (high-level Petri net)である．
- 通常のペトリネットをアセンブラ言語に対応させて考えると，高水準ペトリネットは抽象データ型やオブジェクト指向を支援する高水準言語と言える．

カラーペトリネット：哲学者の食事問題

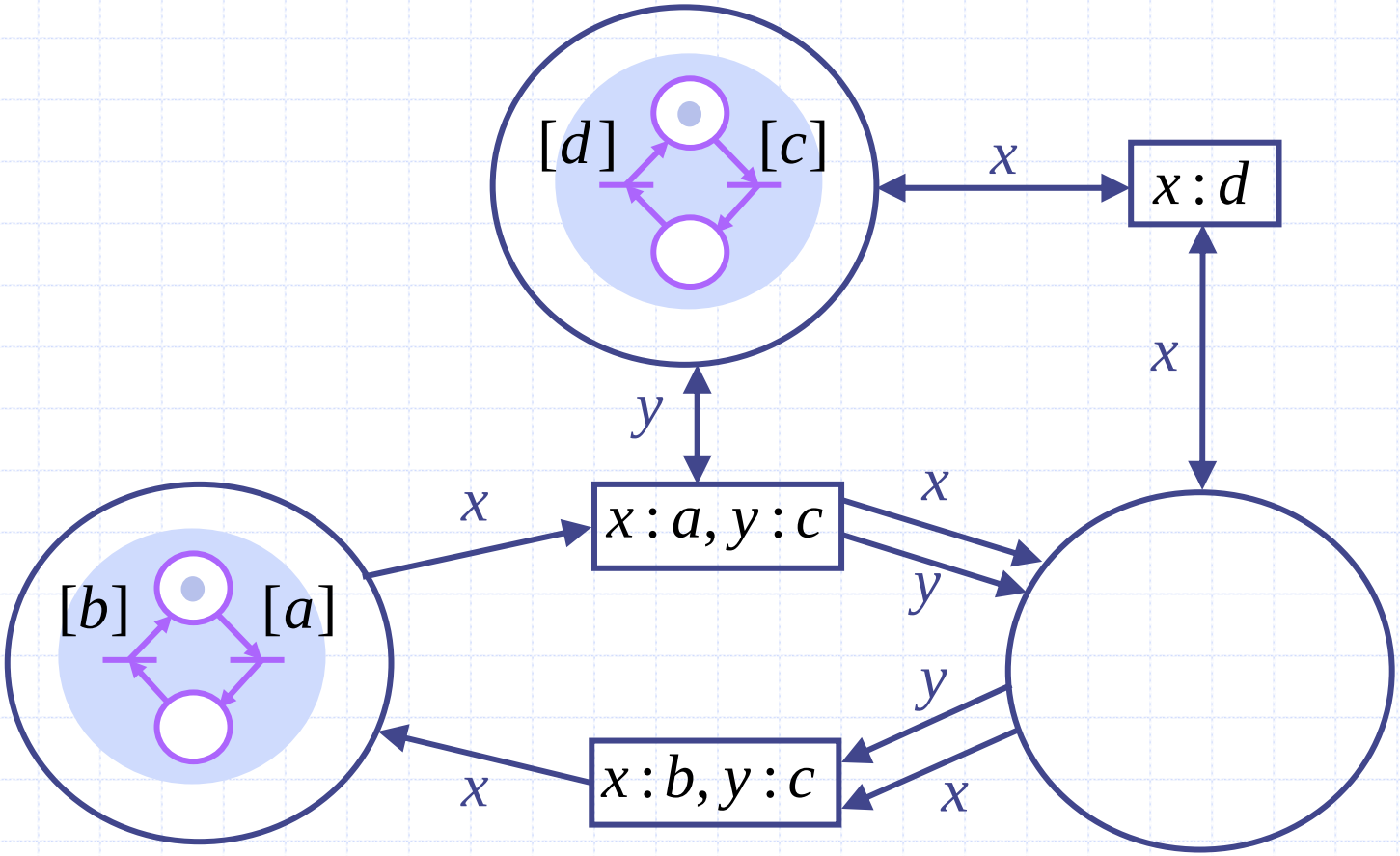



```
val n = 5;
color PH = index ph with 1..n;
color FK = index fk with 1..n;
var x : PH;
fun forks(ph(i)) = 1`fk(i) + 1`fk(if i = n then 1 else i + 1);
```

オブジェクトペトリネット

- **オブジェクトペトリネット**(object Petri net)：トークンがオブジェクト，すなわち，データとそれを操作するためのメソッドを合わせたもの．
- 様々な形式化が存在．

PN²：2階層ペトリネット



情報のソース

- ペトリネットにはモデルの作成，静的/動的解析，シミュレーションなどを行う様々なツールが存在する．
- http://www.daimi.au.dk/PetriNets/に一覧がまとめられている．多くのツールはインターネット上で入手可能である．

参考図書

- 村田忠夫：ペトリネットの解析と応用，近代科学社(1992)
- 椎塚久雄：実例ペトリネット，コロナ社 (1992).
- 離散事象システム研究専門委員会 (編) ：ペトリネットとその応用，計測自動制御学会 (1992).
- 青山幹雄, 平石邦彦, 内平直志:ペトリネットの理論と実践,朝倉書店 (1995).
- 熊谷貞俊，薦田憲久：ペトリネットによる離散事象システム論，コロナ社 (1995).