PETRI NETS

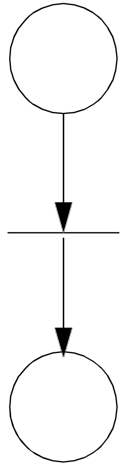
1. Basic Petri Net Model

2. Properties and Analysis of Petri Nets

3. Extended Petri Net Models

Petri Nets

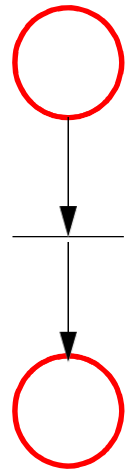
- **Systems are specified as a directed bipartite graph.**
  The two kinds of nodes in the graph:



Petri Nets

- **Systems are specified as a directed bipartite graph. The two kinds of nodes in the graph:**

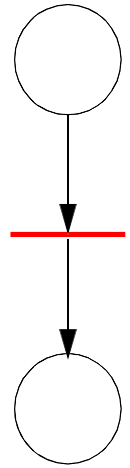
    1. ***Places*****: they hold the distributed state of the system expressed by the presence/absence of tokens in the places.**



Petri Nets

- **Systems are specified as a directed bipartite graph. The two kinds of nodes in the graph:**

  1. ***Places*: they hold the distributed state of the system expressed by the presence/absence of tokens in the places.**
  2. ***Transitions*: denote the activity in the system**

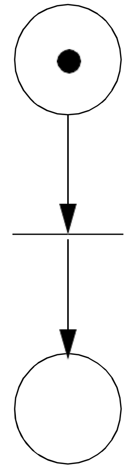


Petri Nets

- Systems are specified as a directed bipartite graph.
  The two kinds of nodes in the graph:

  1. ***Places***: they hold the distributed state of the system expressed by the presence/absence of tokens in the places.

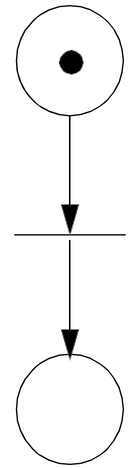
  2. ***Transitions***: denote the activity in the system



- ***The state of the system***: captured by the marking of the places (number of tokens in each place)

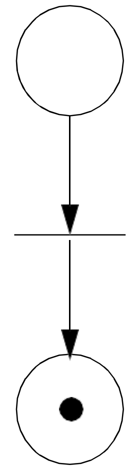
Petri Nets

- The dynamic evolution of the system: determined by the firing process of transitions.
    - A transition is enabled and may fire whenever all its predecessor places are marked.



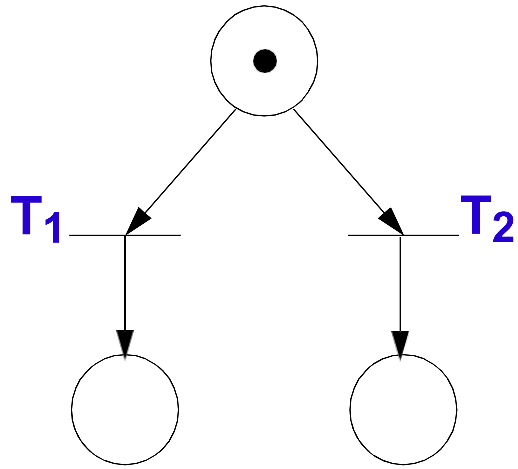
Petri Nets

- **The dynamic evolution of the system: determined by the firing process of transitions.**
  - **A transition is enabled and may fire whenever all its predecessor places are marked.**
  - **If a transition fires it removes a token from each predecessor place and adds a token to each successor place.**



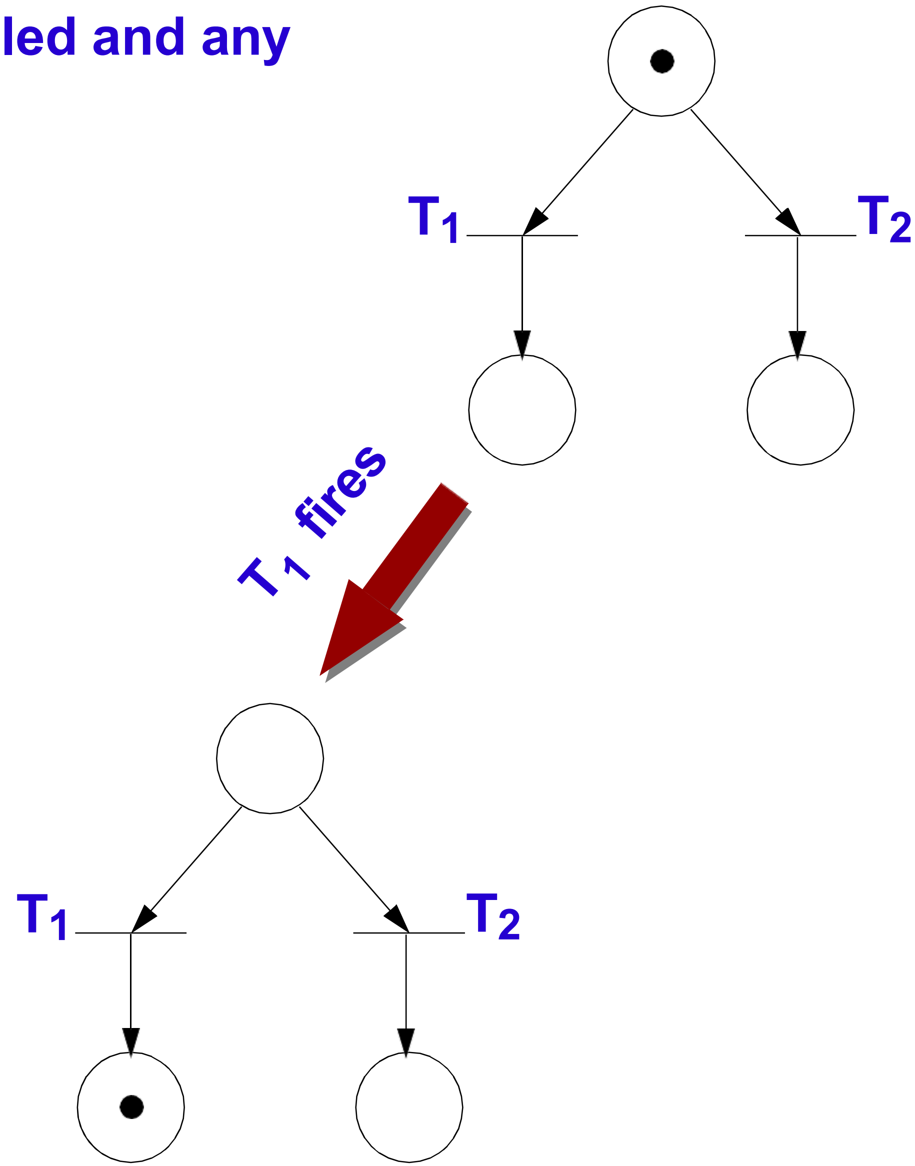
Nondeterminism

- **Both T1 and T2 are enabled and any of the two may fire.**



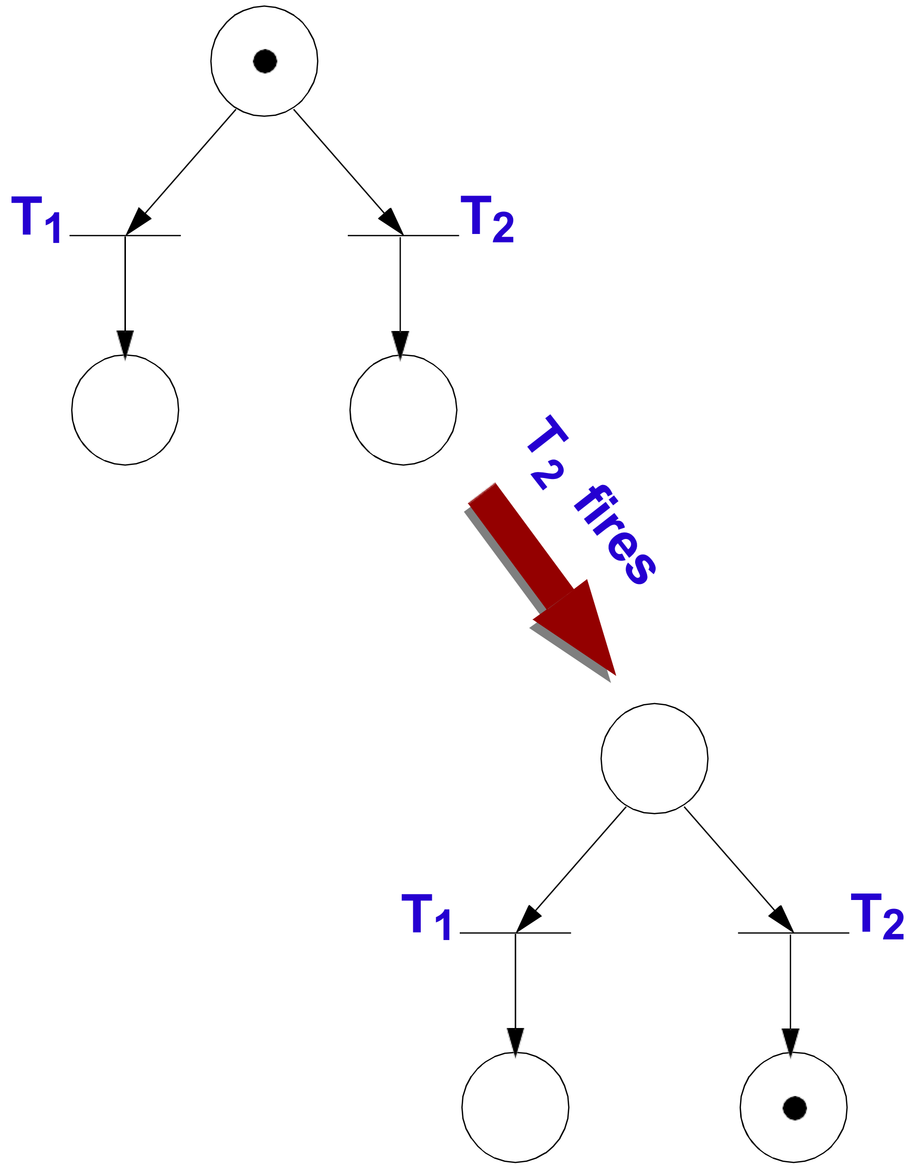
Nondeterminism

- Both T1 and T2 are enabled and any of the two may fire.



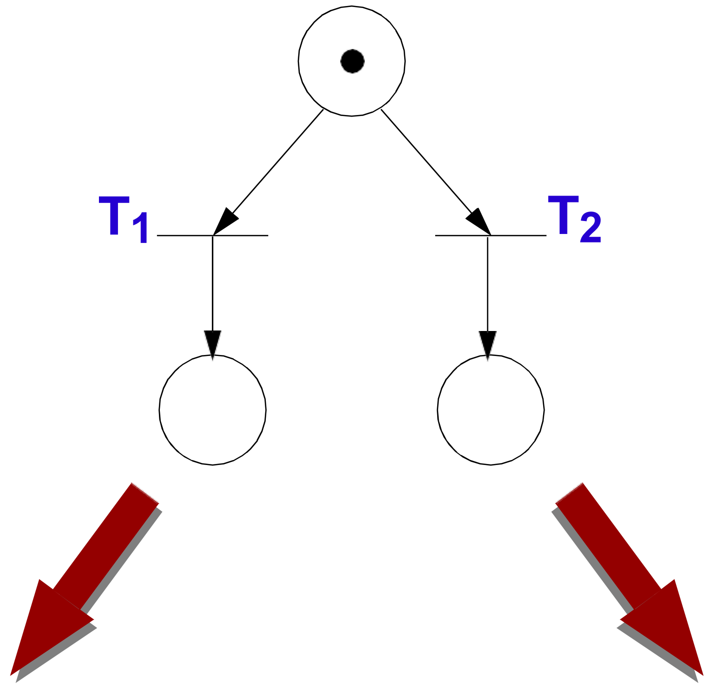
Nondeterminism

- Both T1 and T2 are enabled and any of the two may fire.

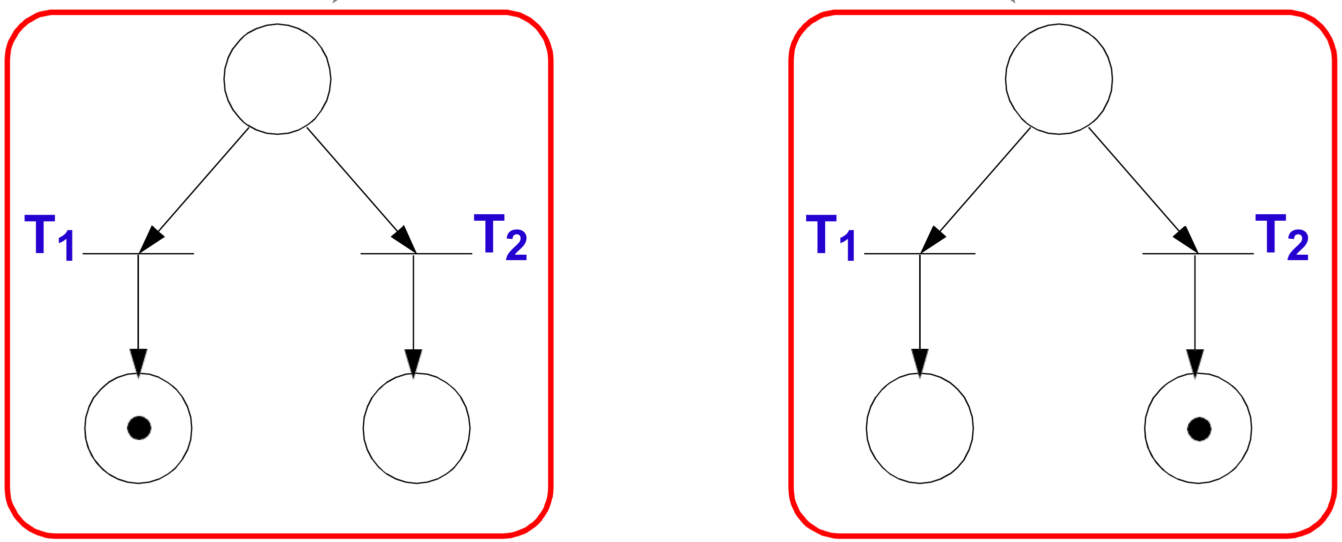


Nondeterminism

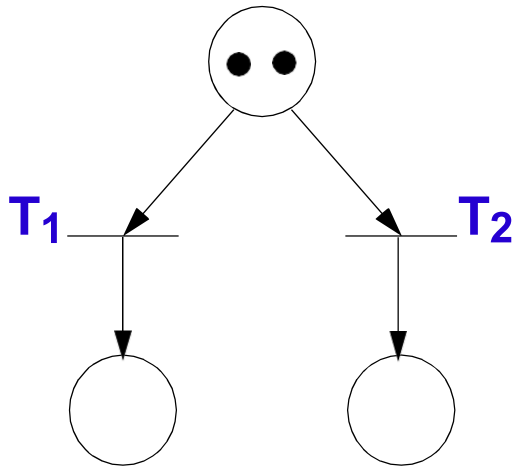
- Both T1 and T2 are enabled and any of the two may fire.



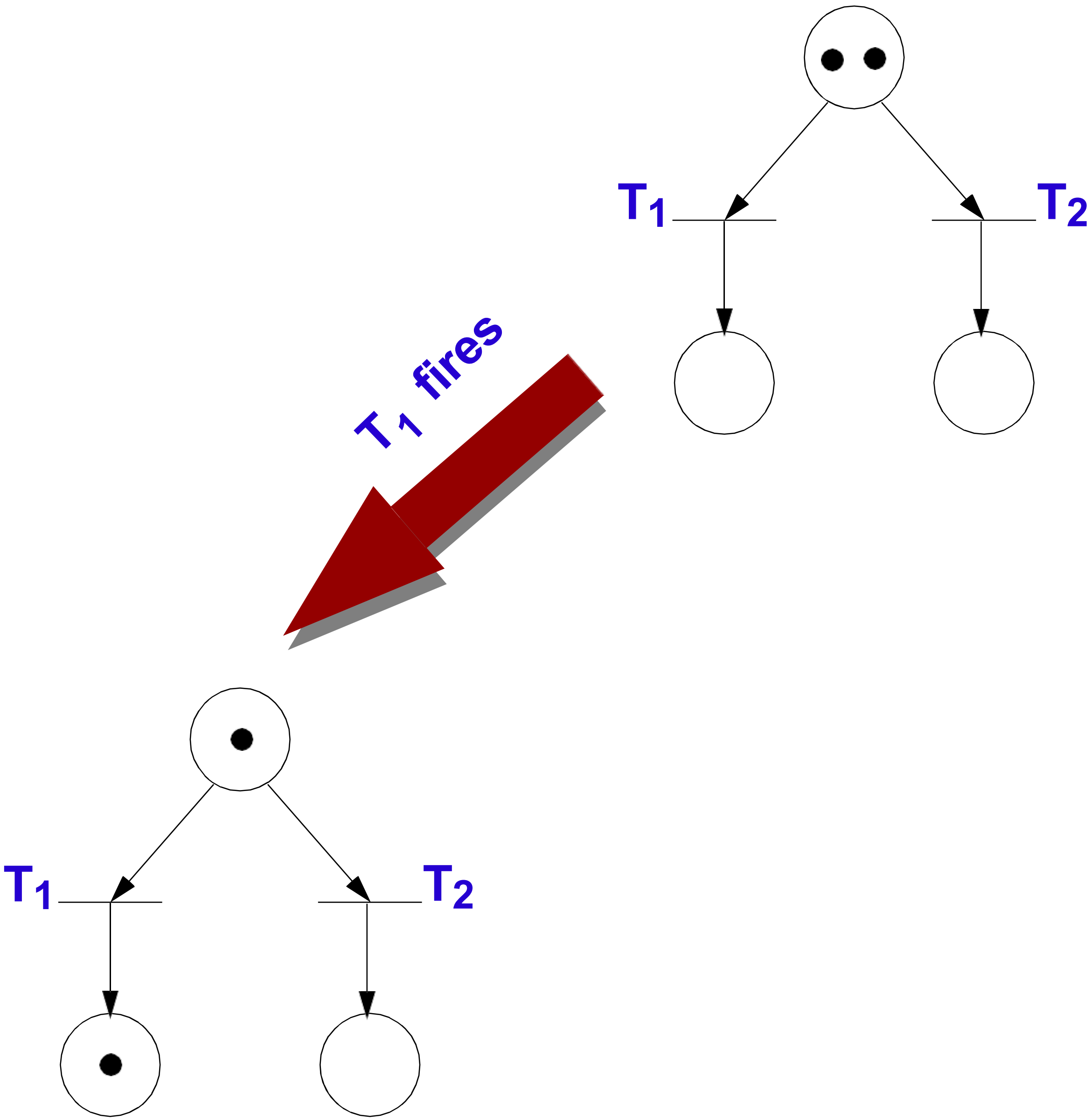
- Any of these two states might be the next state.



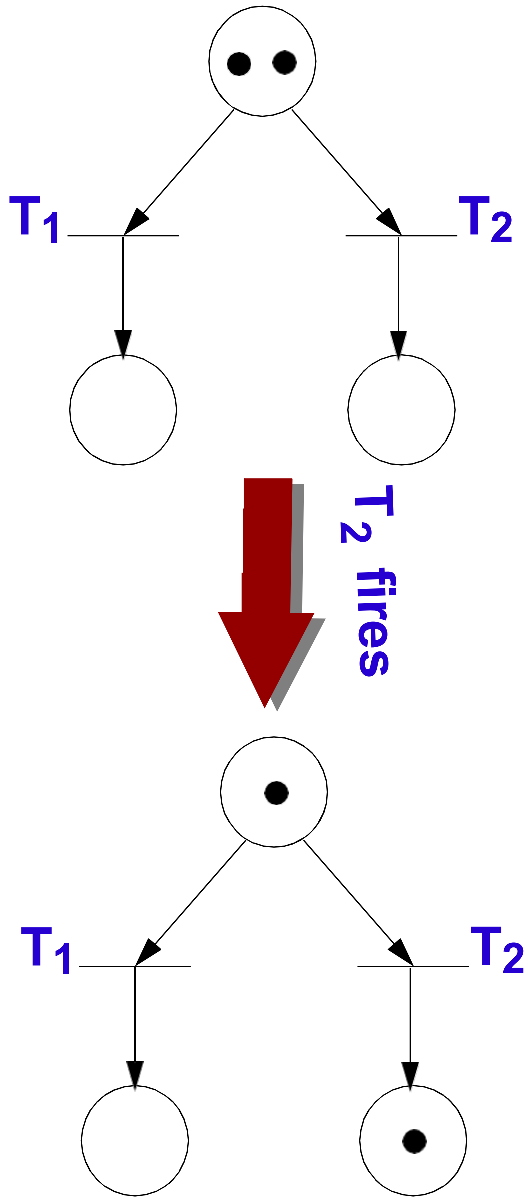
Nondeterminism



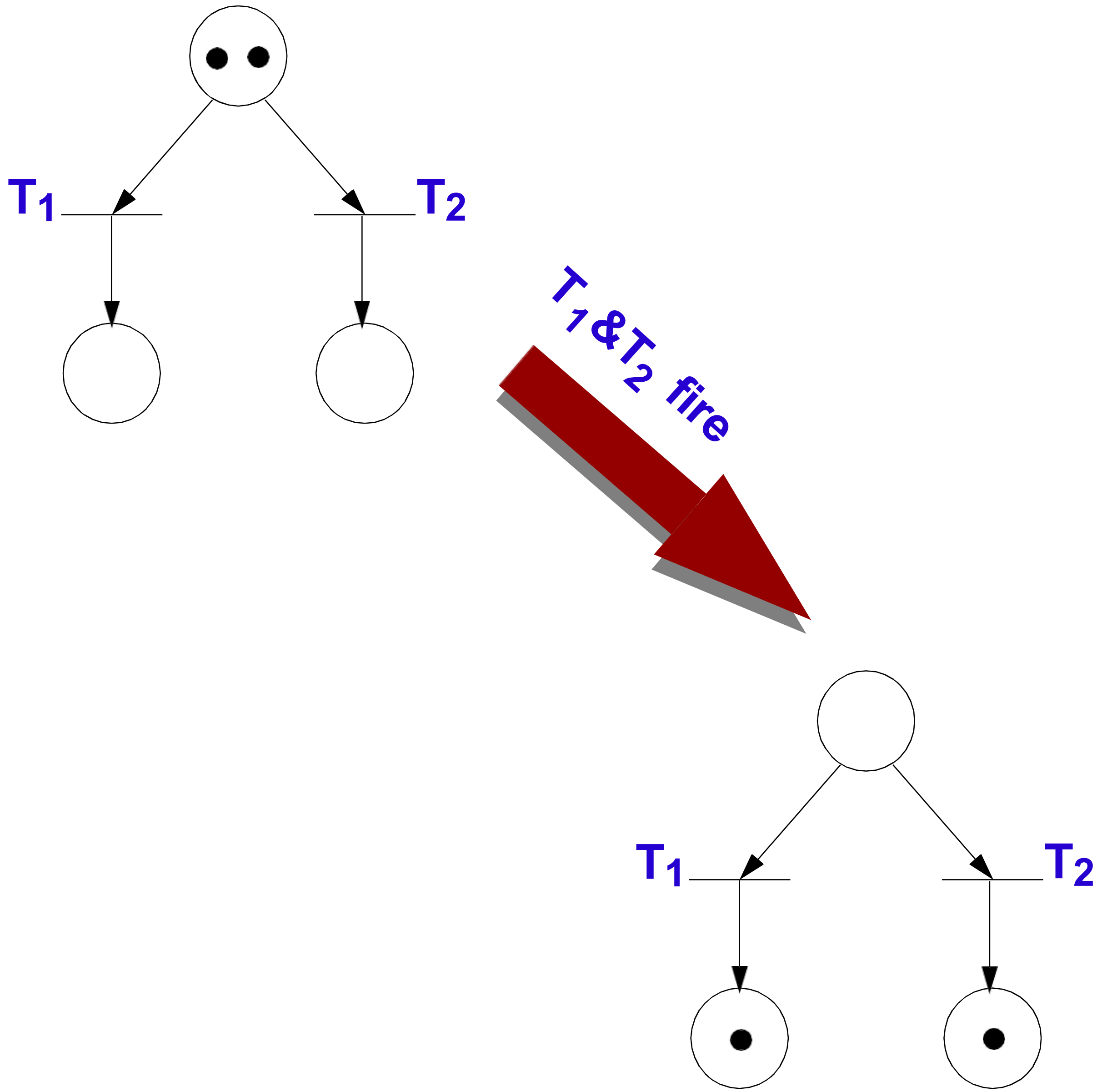
Nondeterminism



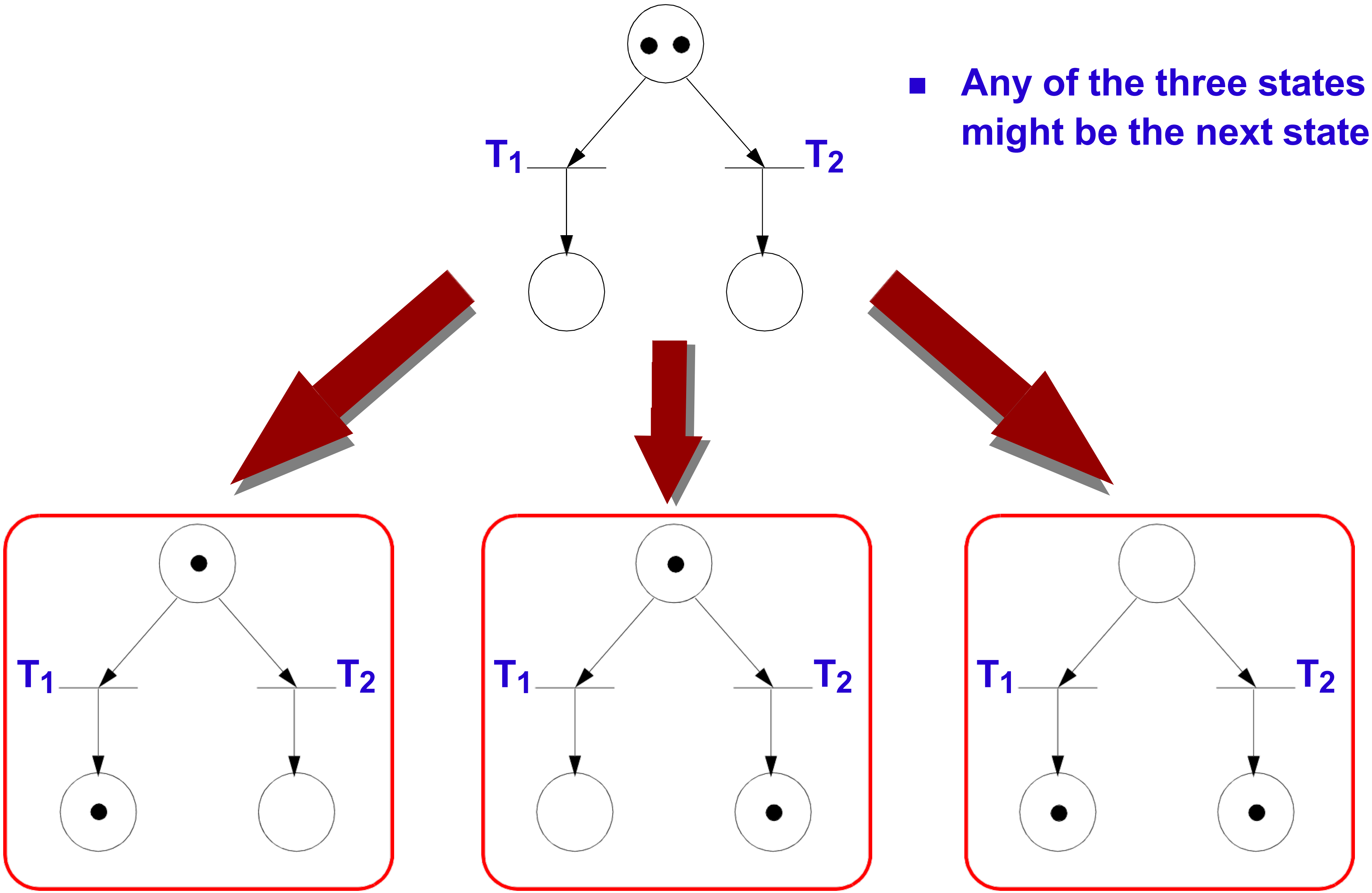
Nondeterminism



Nondeterminism

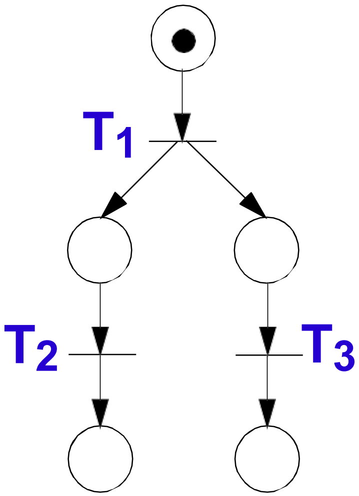


Nondeterminism



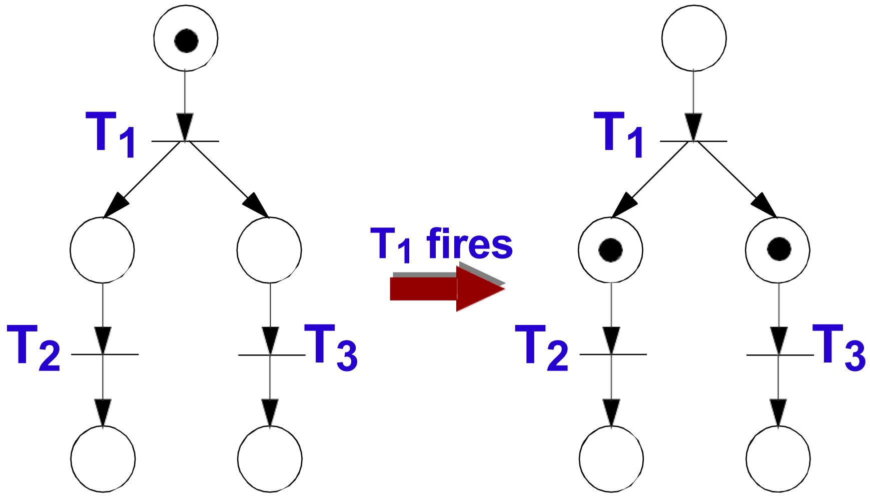
- Any of the three states might be the next state.

Nondeterminism

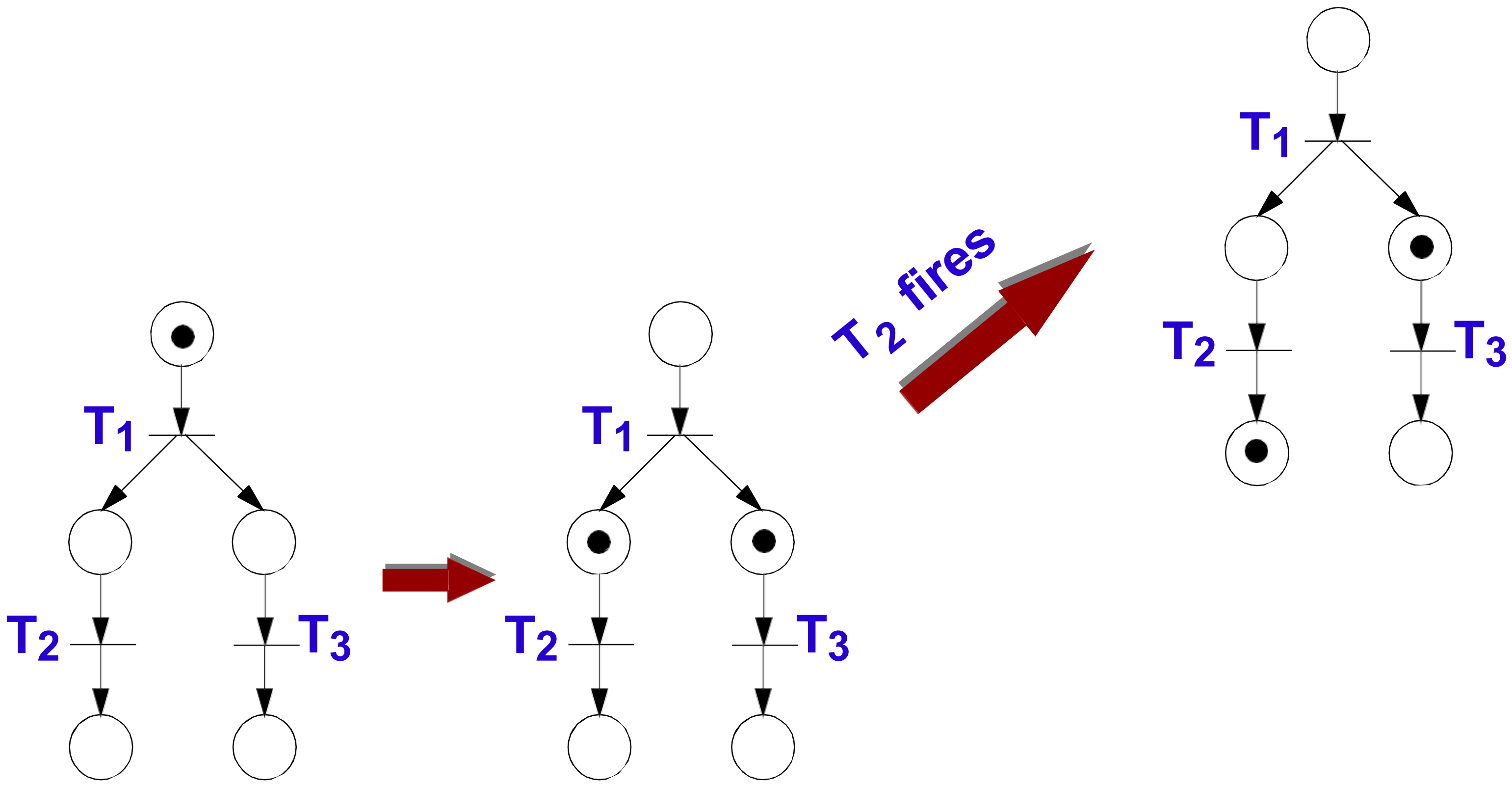


Nondeterminism

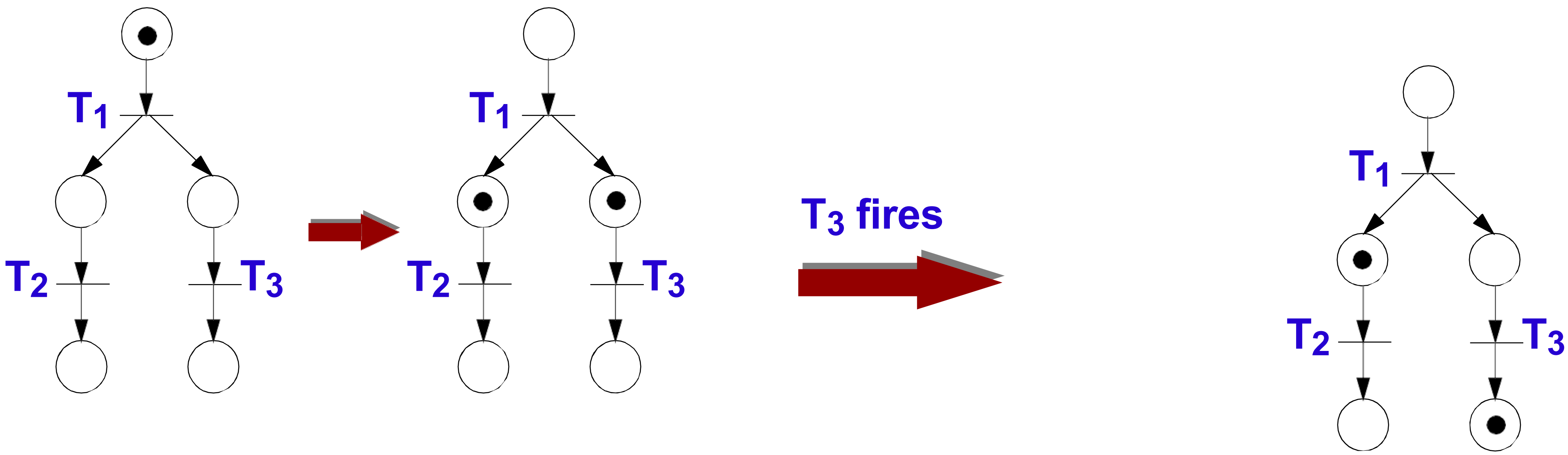
- **No nondeterminism here:**
  T_1 is the only enabled transition!



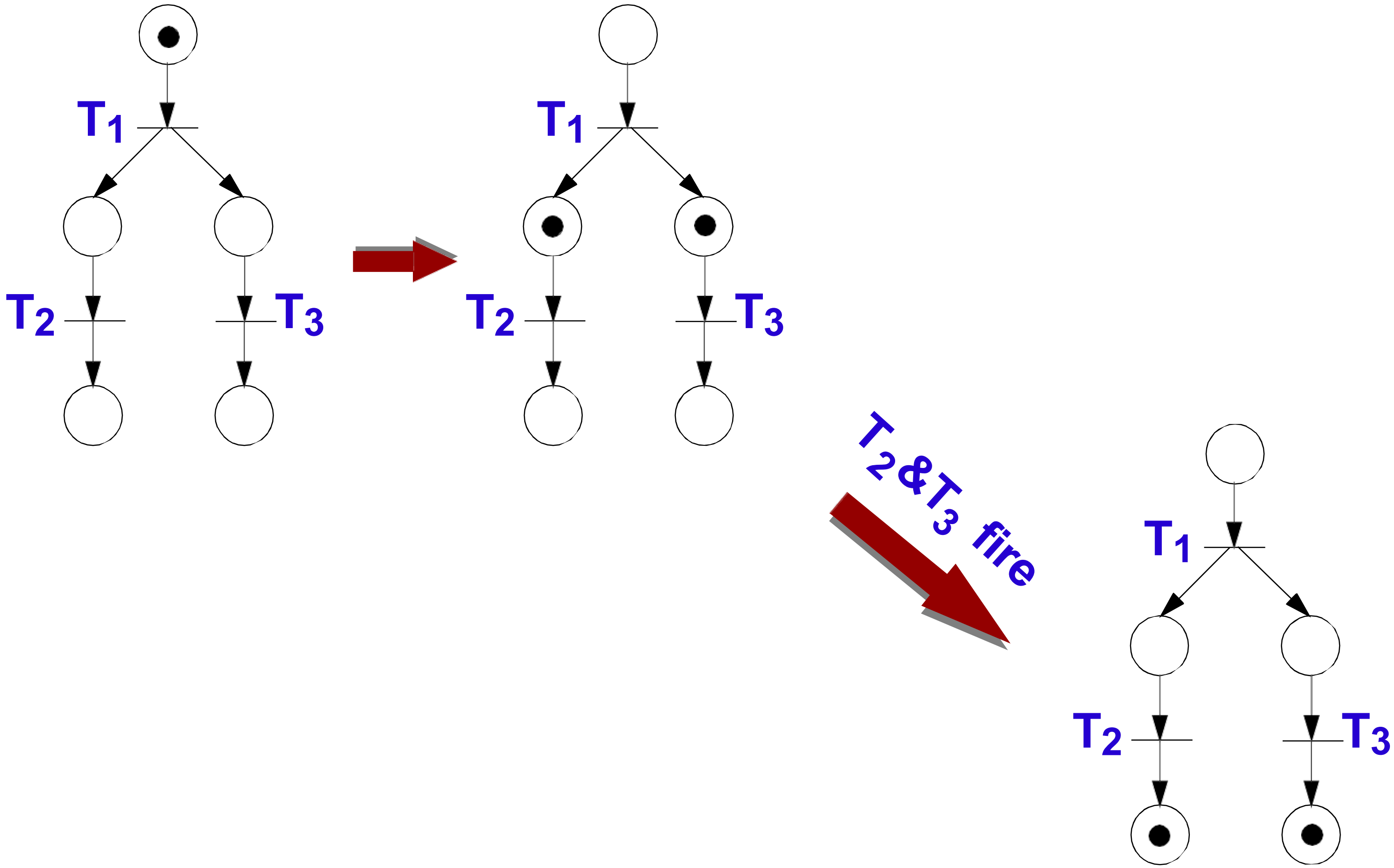
Nondeterminism



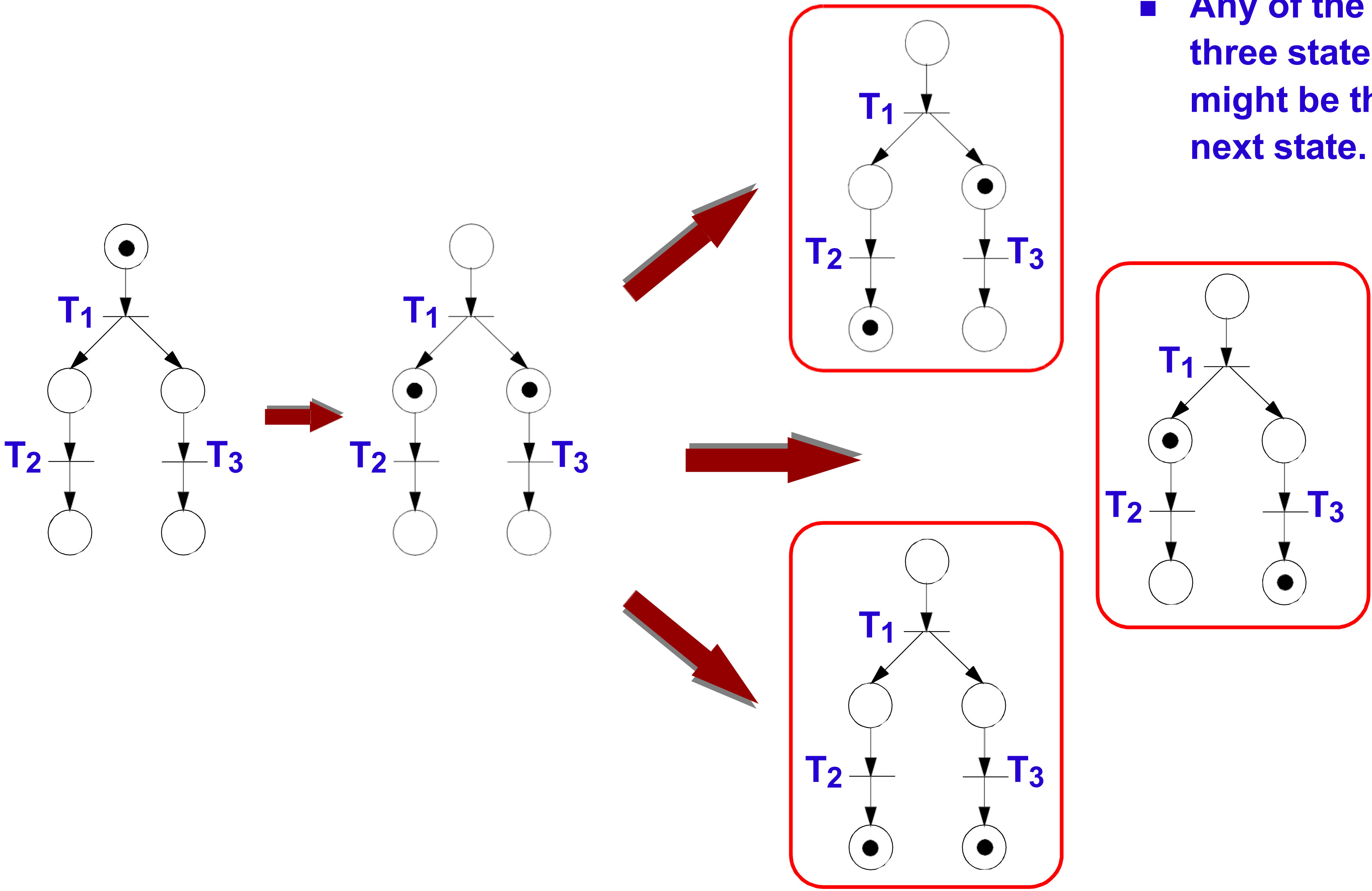
Nondeterminism



Nondeterminism



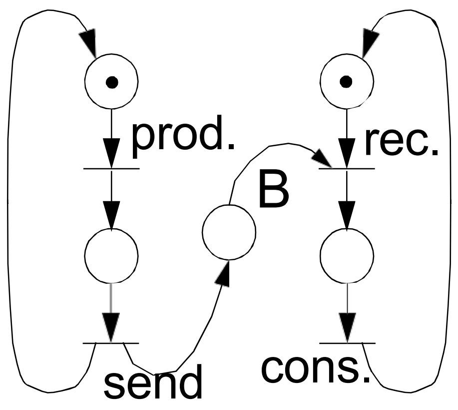
Nondeterminism



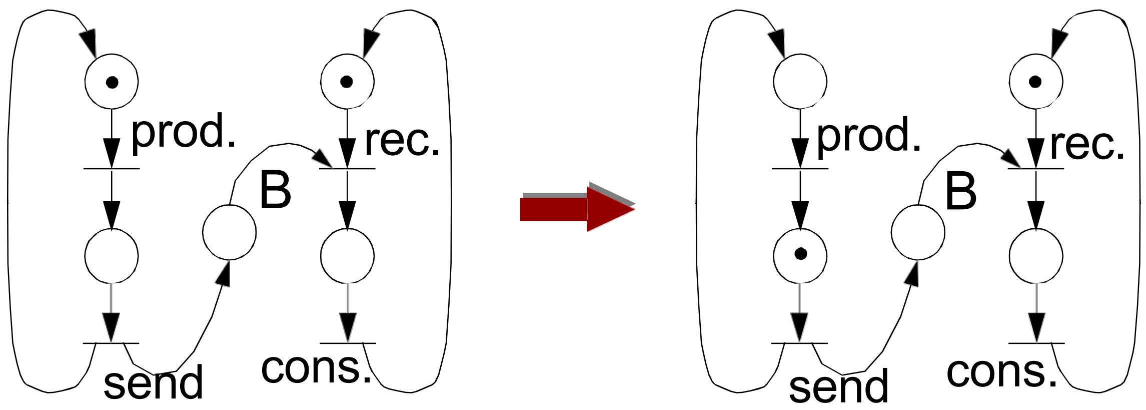
- Any of the three states might be the next state.

Petri Net Example

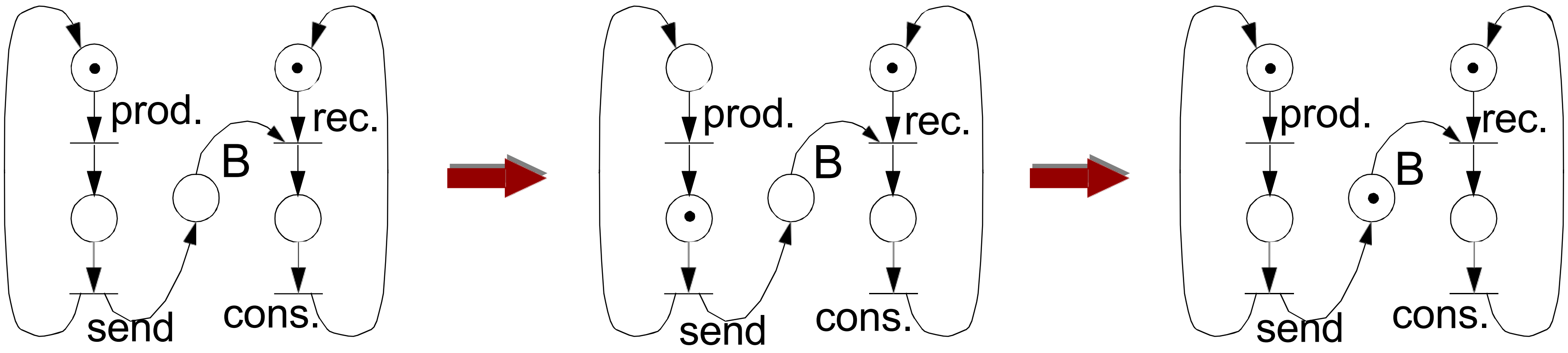
A producer and a consumer process communicating through a buffer:



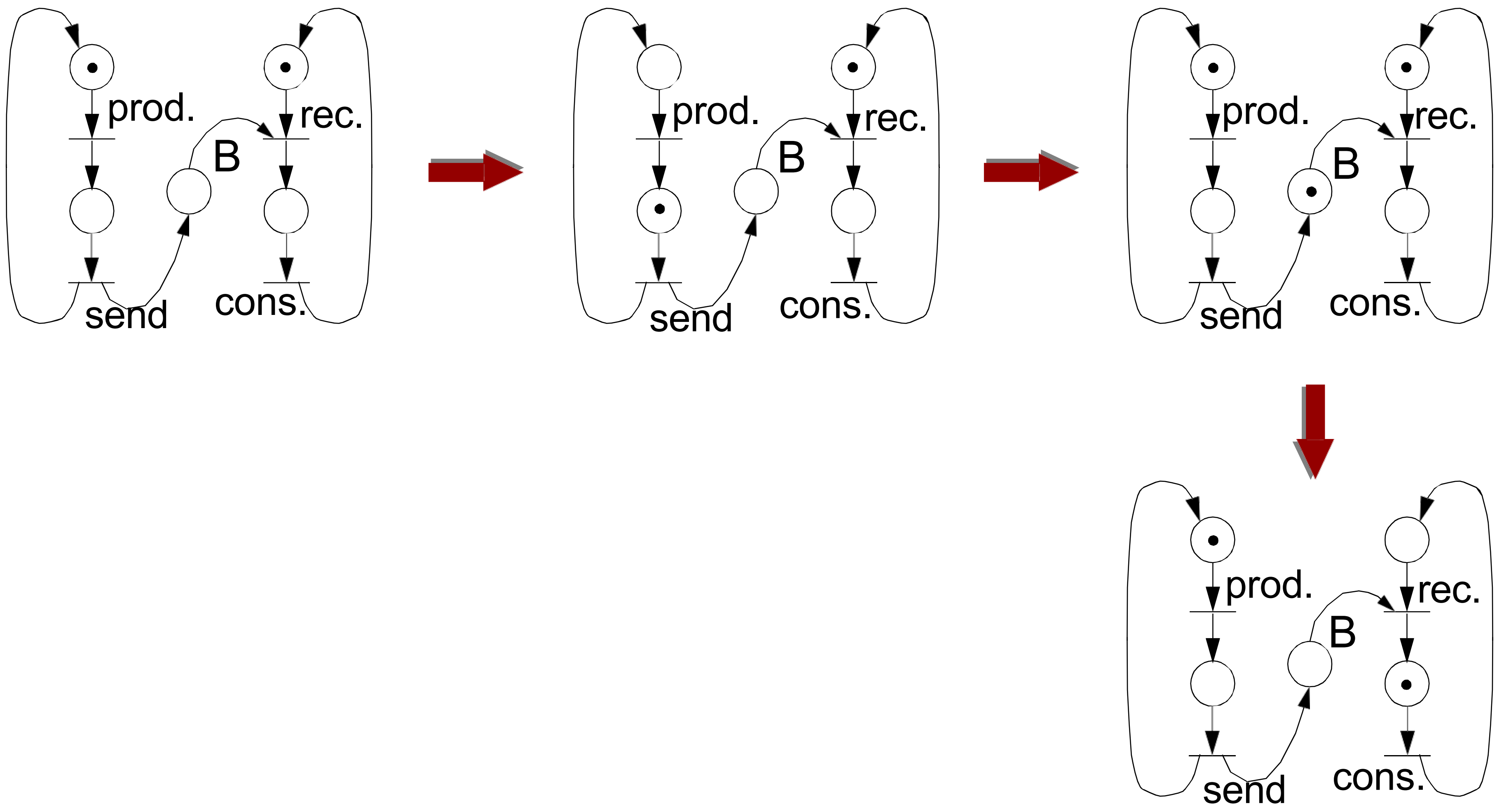
Petri Net Example



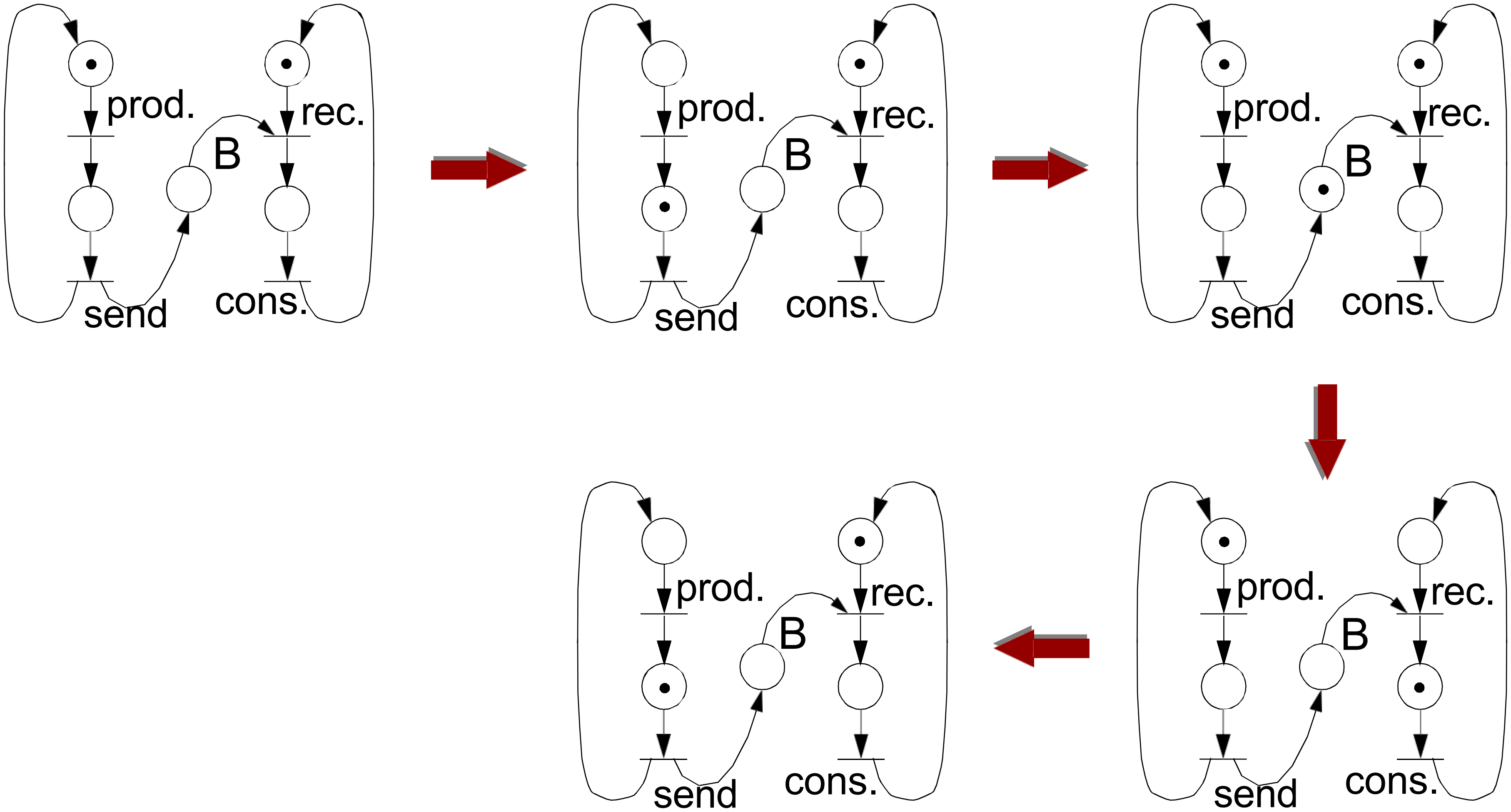
Petri Net Example



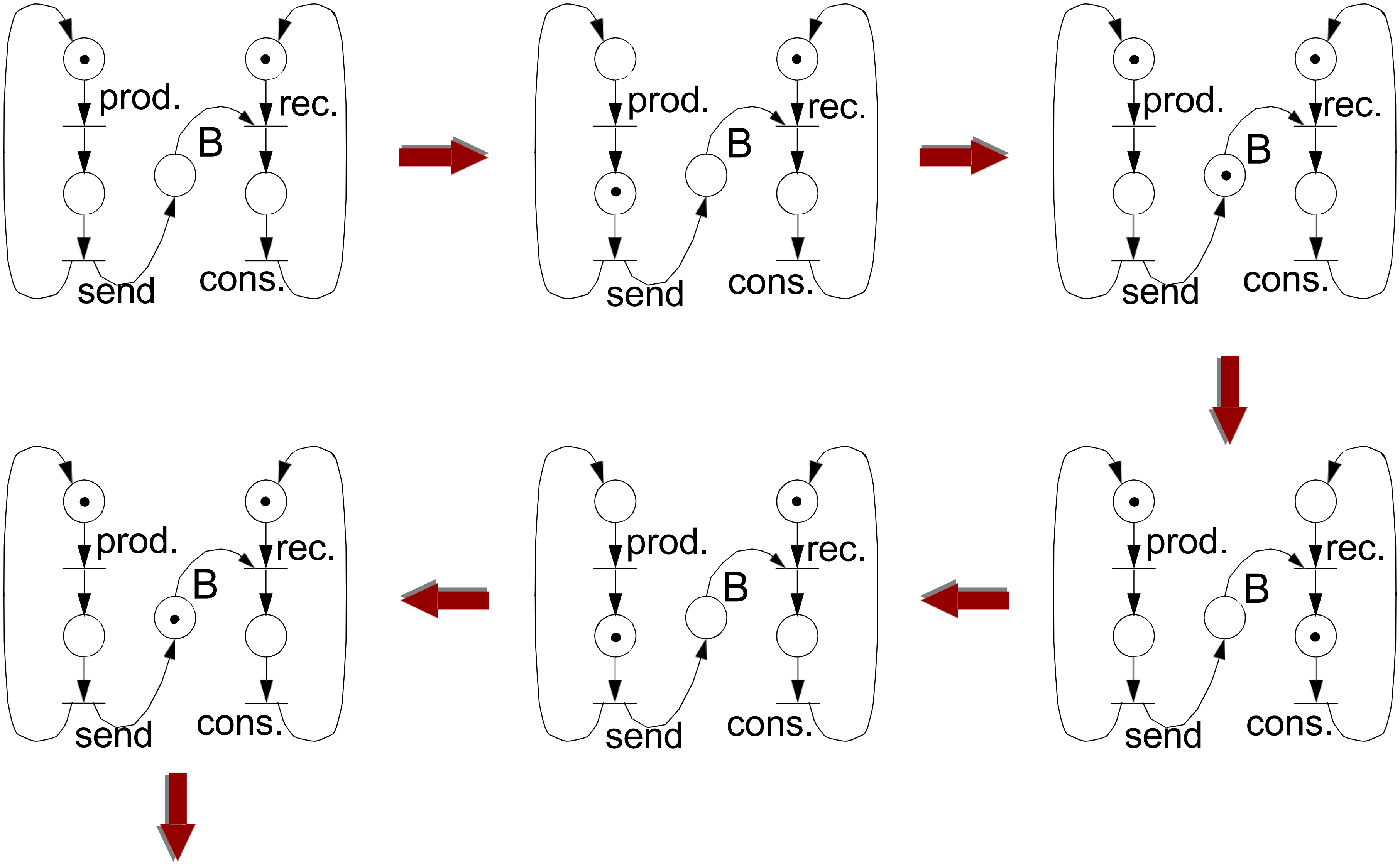
Petri Net Example



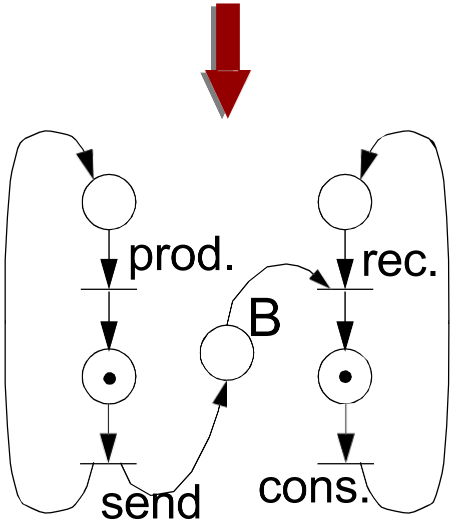
Petri Net Example



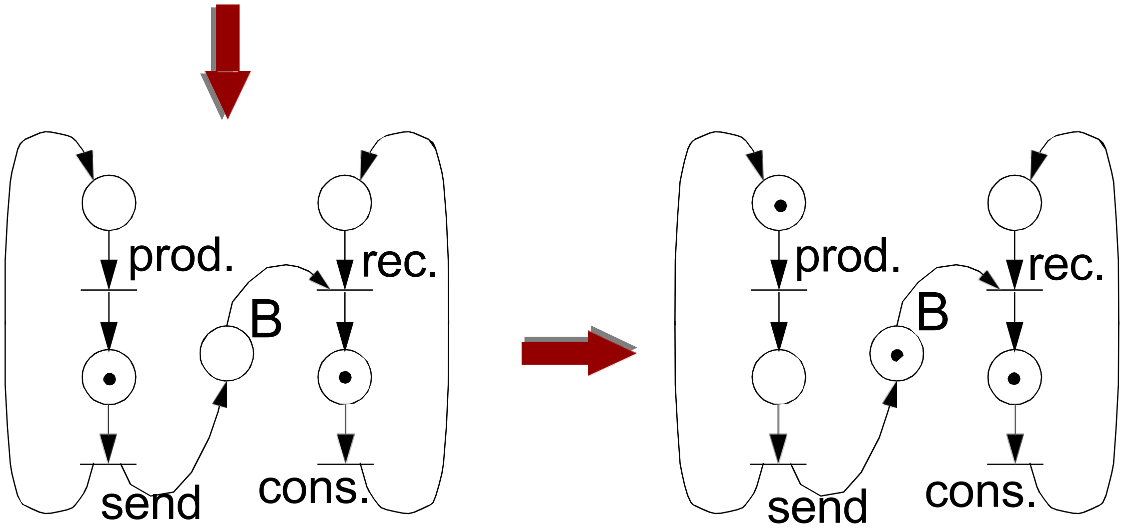
Petri Net Example



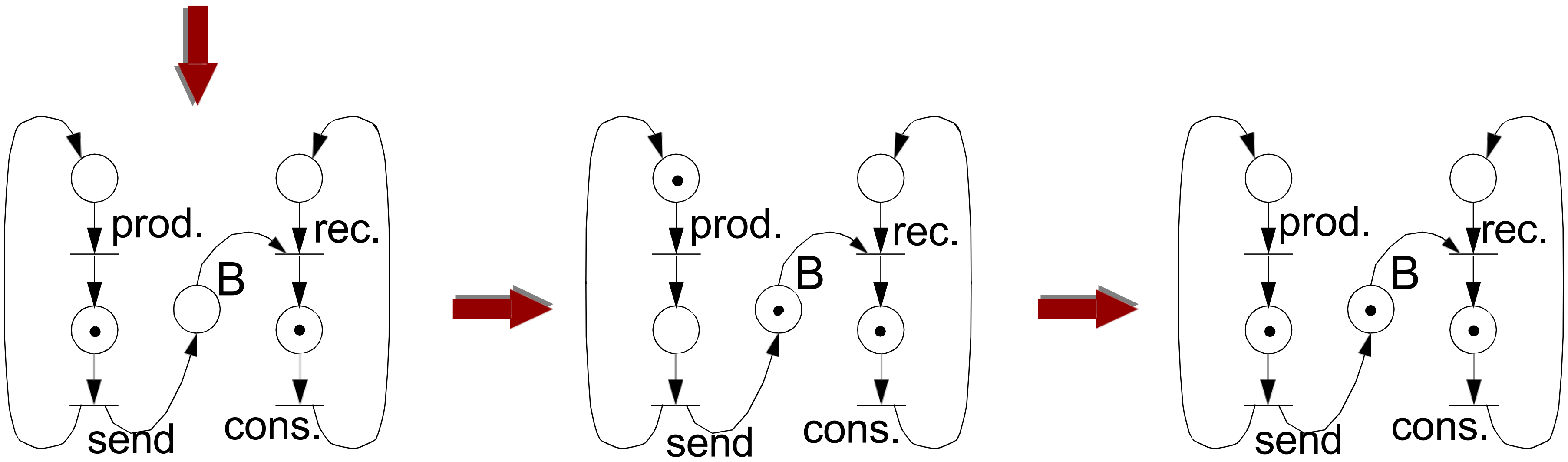
Petri Net Example



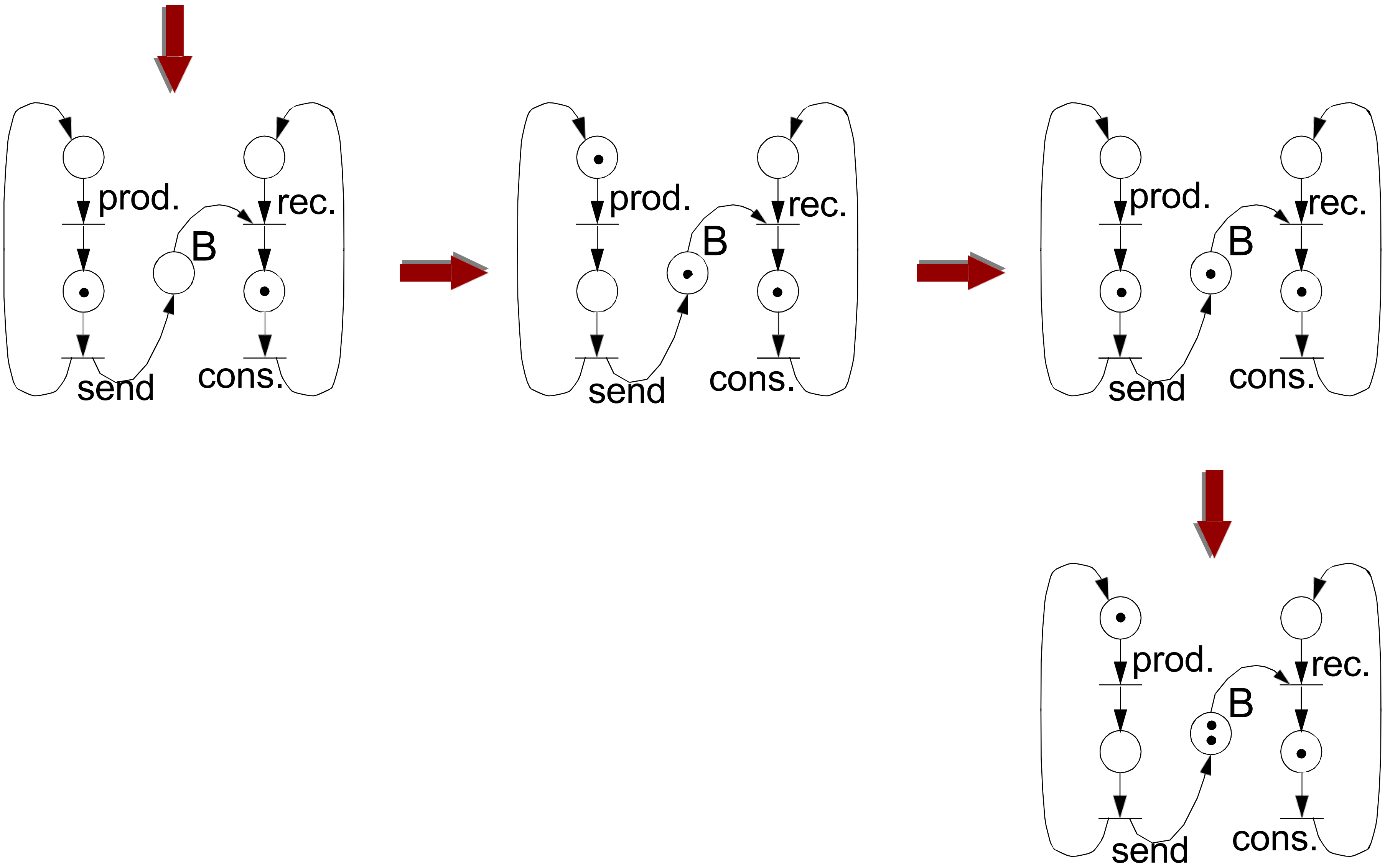
Petri Net Example



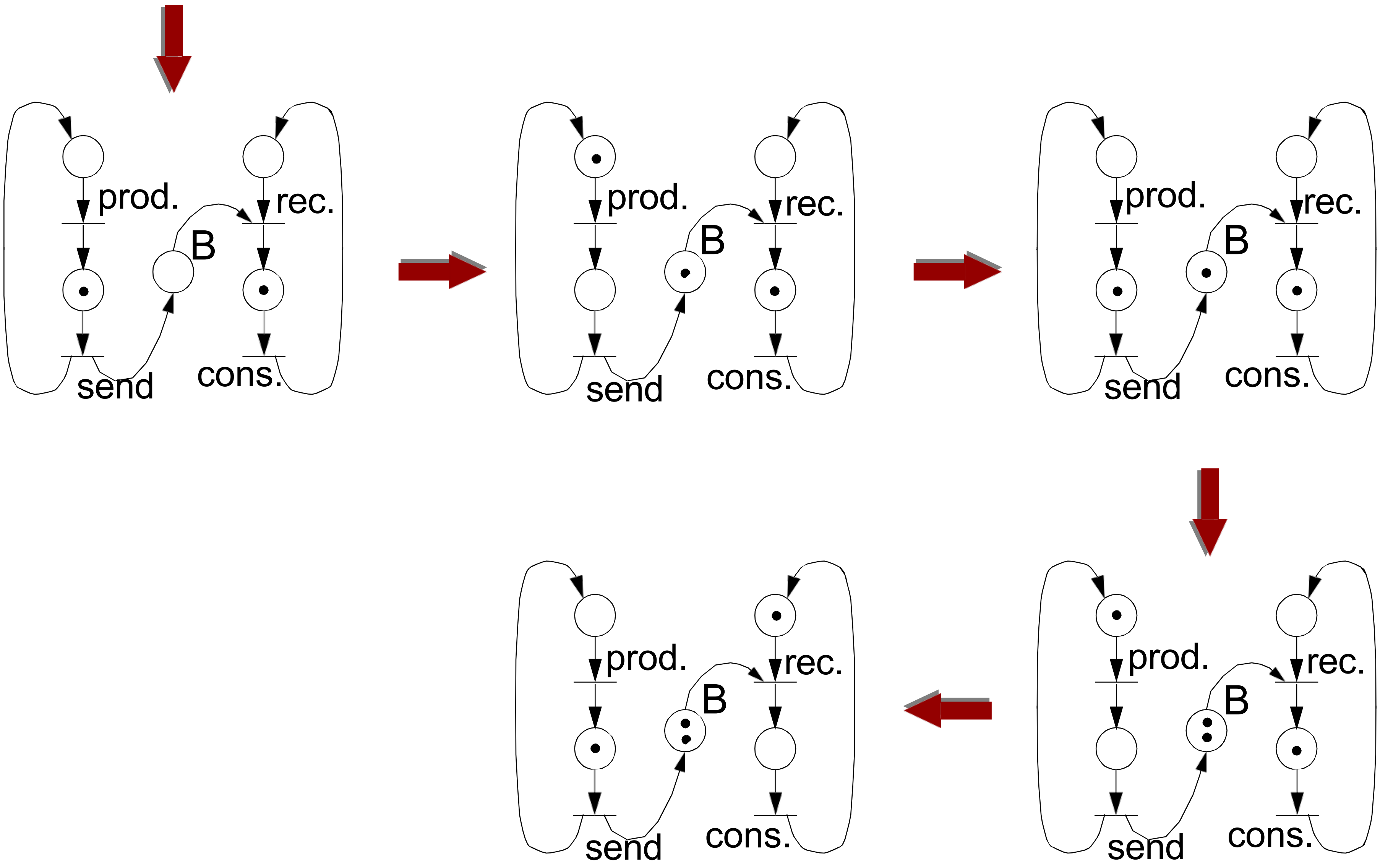
Petri Net Example



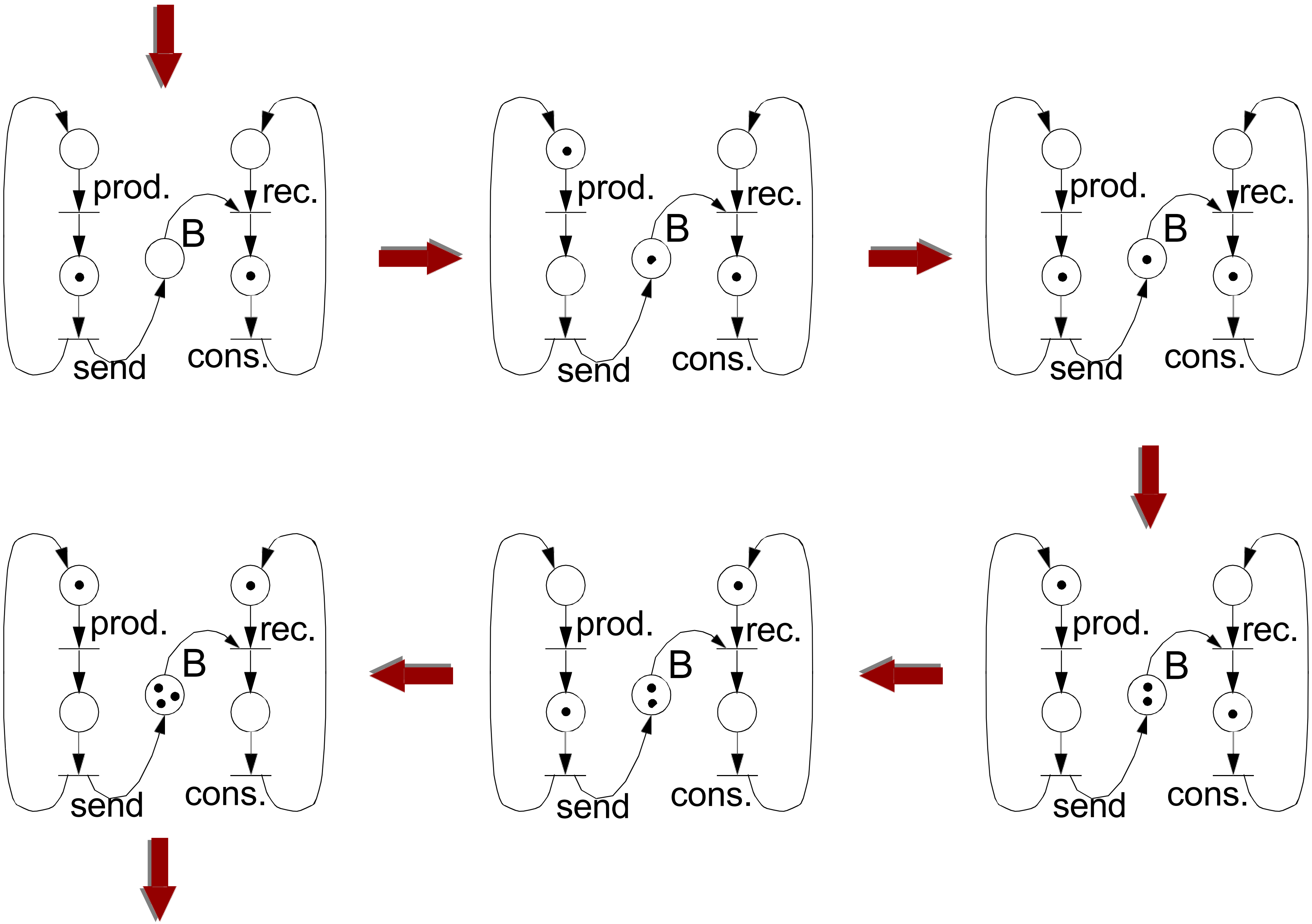
Petri Net Example



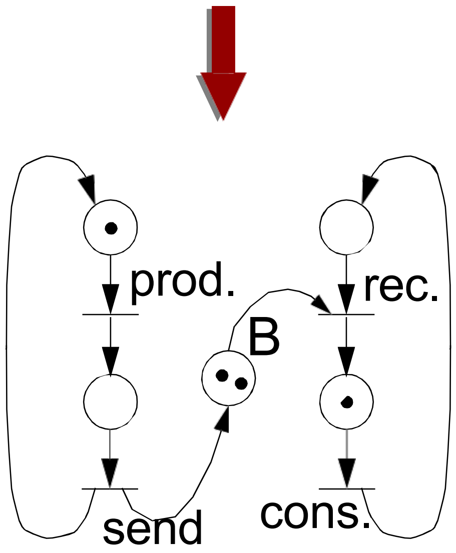
Petri Net Example



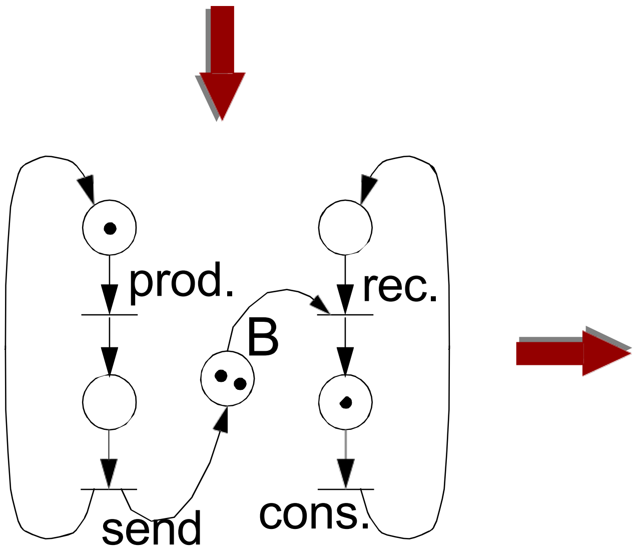
Petri Net Example



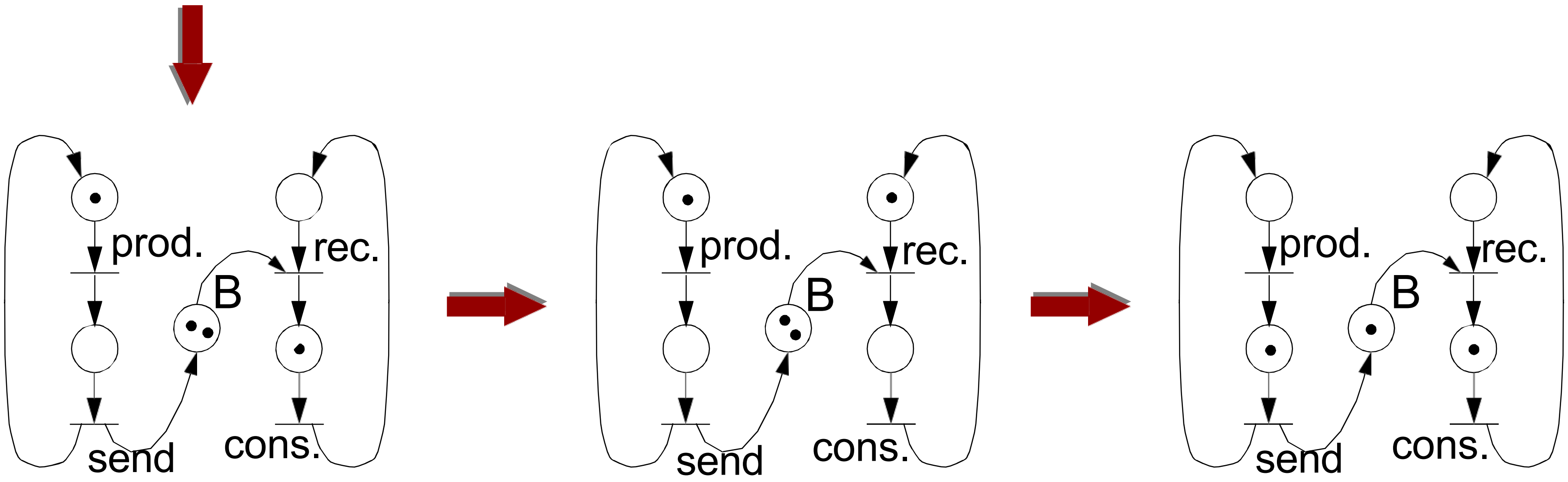
Petri Net Example



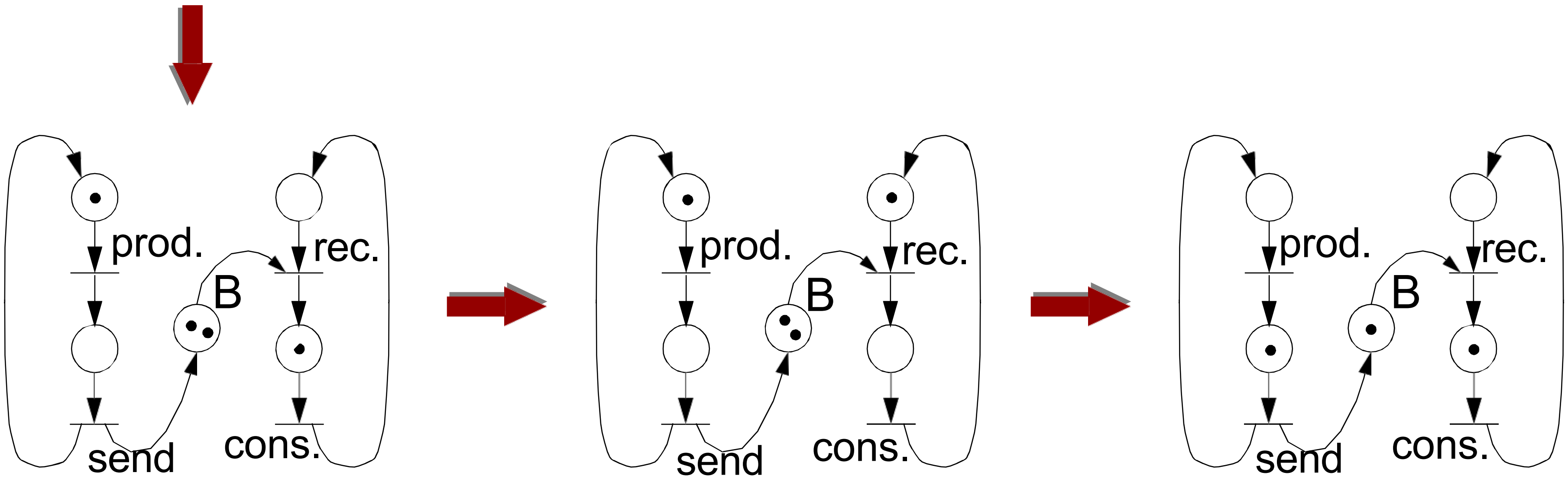
Petri Net Example



Petri Net Example



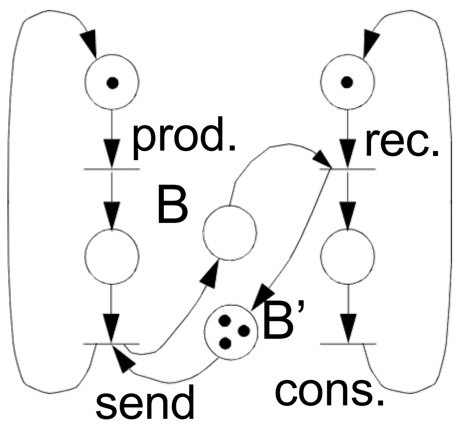
Petri Net Example



- **Notice that the buffer is considered to be infinite (tokens accumulate in *B*).**

Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')

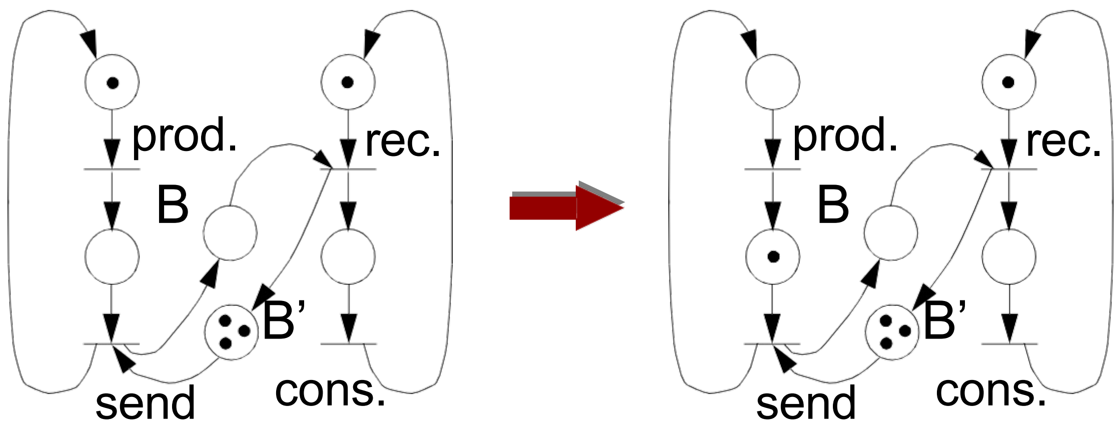


- **Nr. of tokens in B': how many free slots are available in the buffer;**
- **Nr. of tokens in B: how many messages (tokens) are in the buffer.**

Total number of tokens in *B* and *B*' is constant (= 3).

Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')

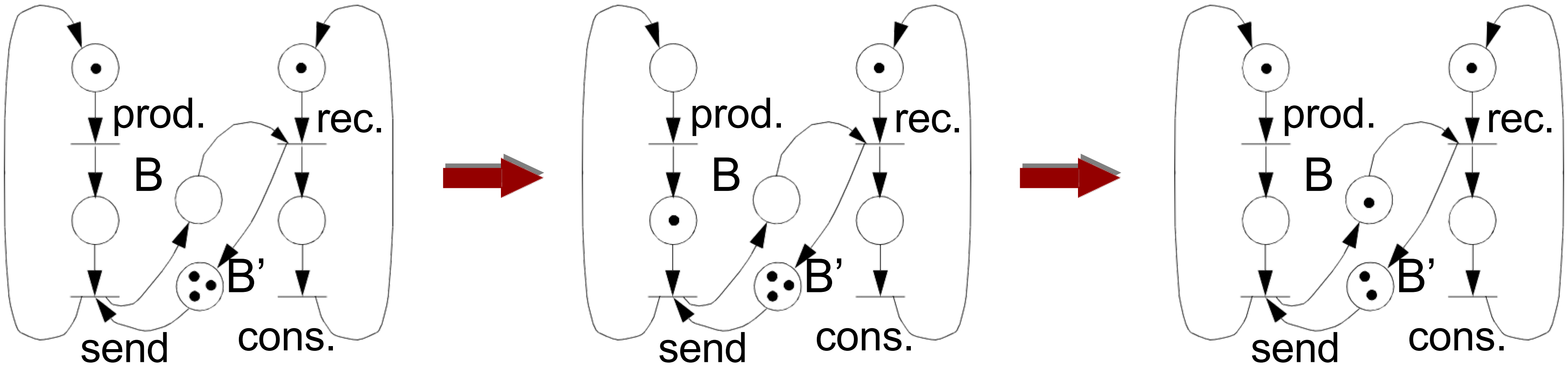


- **Nr. of tokens in B': how many free slots are available in the buffer;**
- **Nr. of tokens in B: how many messages (tokens) are in the buffer.**

Total number of tokens in *B* and *B*' is constant (= 3).

Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')

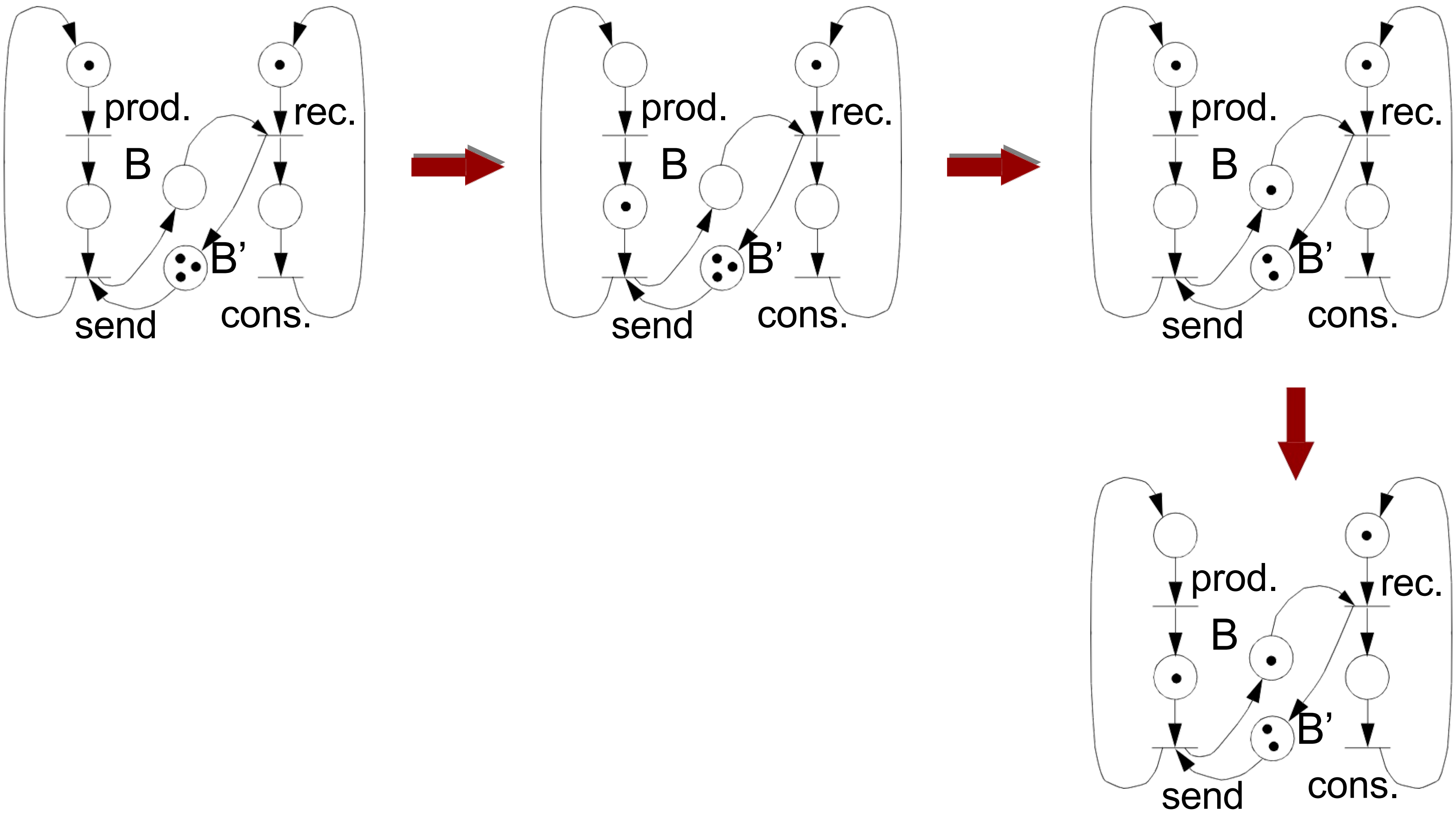


- **Nr. of tokens in B': how many free slots are available in the buffer;**
- **Nr. of tokens in B: how many messages (tokens) are in the buffer.**

Total number of tokens in *B* and *B*' is constant (= 3).

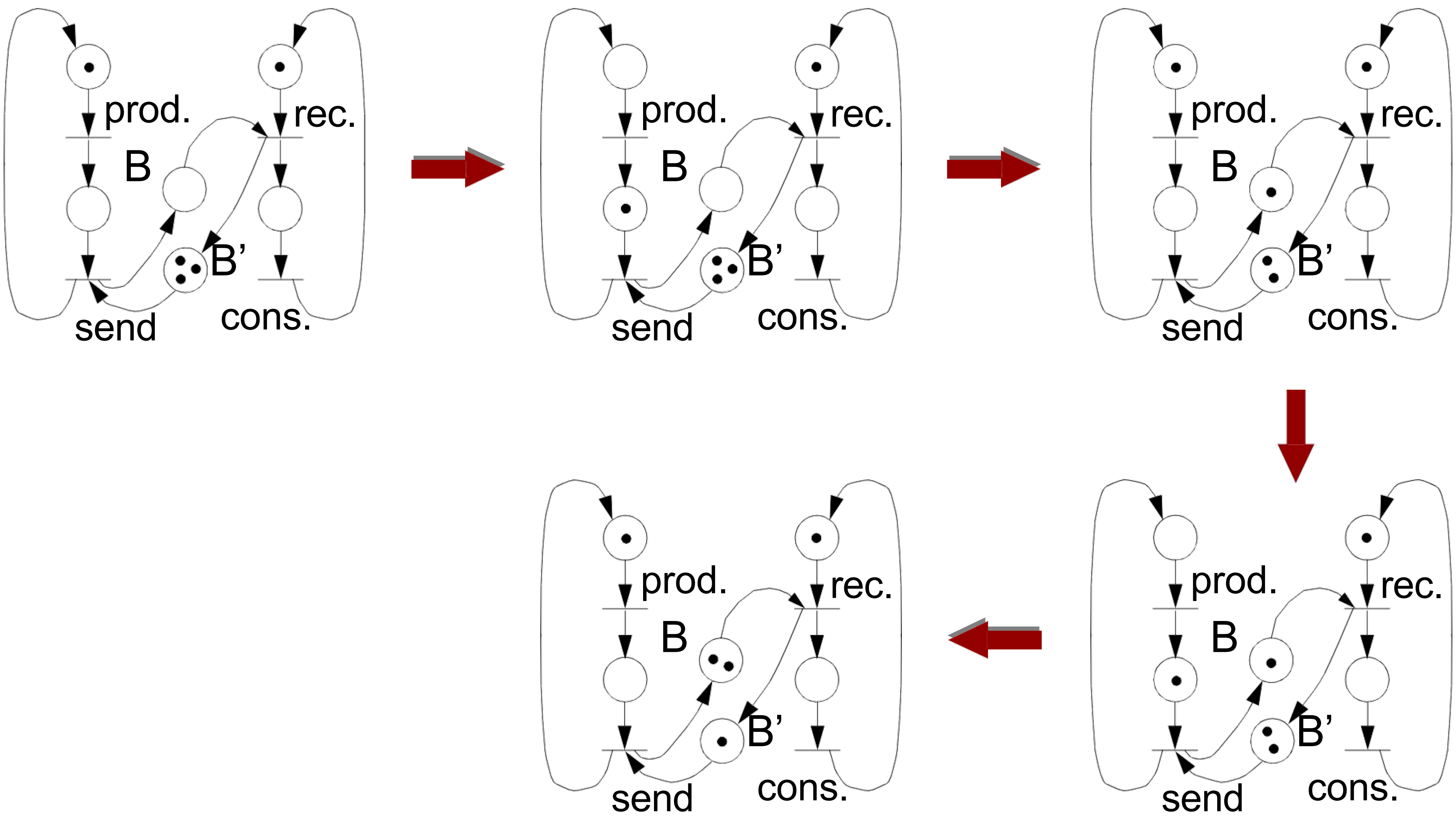
Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')



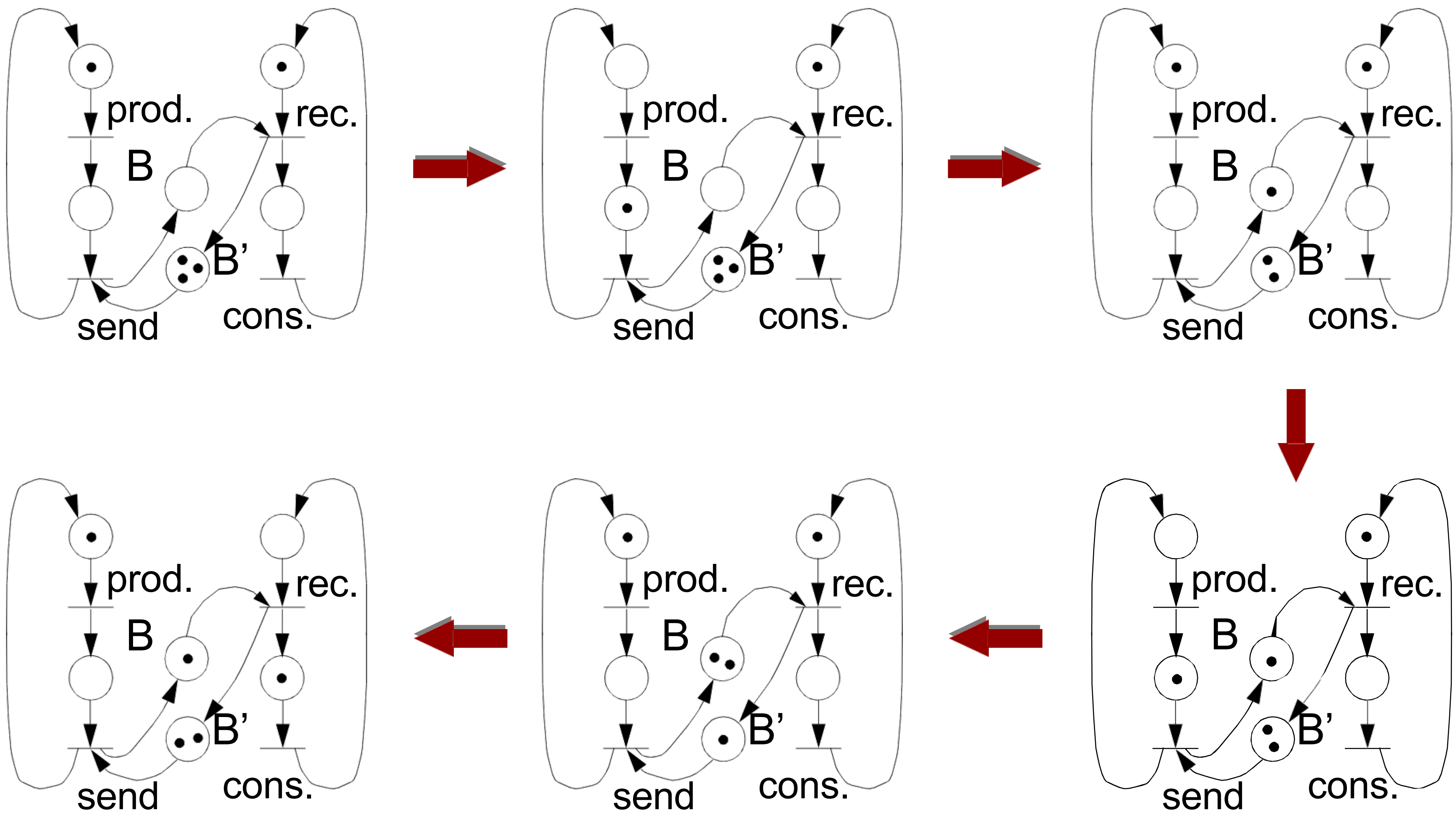
Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')



Petri Net Example

Here we have the same model as on the previous slides, but with limited buffer. The buffer size is three (number of initial tokens in *B*')



Some Features and Applications of Petri Nets

- **Intuitive.**

  Easy to express concurrency, synchronisation, nondeterminism.

  Nondeterminism is an important difference between Petri nets and dataflow!

- **As an uninterpreted model, Petri Nets can be used for several, very different classes of problems.**

  - ***Uninterpreted model*: nothing has to be specified related to the particular activities associated to the transitions.**

Some Features and Applications of Petri Nets

- **Petri Nets have been intensively used for modeling and analysis of industrial production systems, information systems, but also**
  - **Computer architectures**
  - **Operating systems**
  - **Concurrent programs**
  - **Distributed systems**
  - **Hardware systems**

Properties and Analysis of Petri Nets

- Several properties of the system can be analysed using Petri nets:
  - *Boundedness*: number of tokens in a place does not exceed a limit. If this limit is 1, the property is sometimes called *safeness*.
    - You can check that available resources are not exceeded.

Properties and Analysis of Petri Nets

- **Several properties of the system can be analysed using Petri nets:**
  - ***Boundedness*: number of tokens in a place does not exceed a limit.**

    If this limit is 1, the property is sometimes called *safeness*.
    - **You can check that available resources are not exceeded.**
  - ***Liveness*: A transition *t* is called live if for every possible marking there exists a chance for that transition to become enabled.**

    The whole net is live, if all its transitions are live.
    - **Important in order to check that the system is not deadlocked.**

Properties and Analysis of Petri Nets

- **Several properties of the system can be analysed using Petri nets:**
  - ***Boundedness*: number of tokens in a place does not exceed a limit.**

    If this limit is 1, the property is sometimes called *safeness*.
    - **You can check that available resources are not exceeded.**
  - ***Liveness*: A transition *t* is called live if for every possible marking there exists a chance for that transition to become enabled.**

    The whole net is live, if all its transitions are live.
    - **Important in order to check that the system is not deadlocked.**
  - ***Reachability*: given a current marking M and another marking M', does there exist a sequence of transitions by which M' can be obtained?**
    - **You can check that a certain desired state (marking) is reached.**
    - **You can check that a certain undesired state is never reached.**

Properties and Analysis of Petri Nets

Mathematical tools are available for analysis of Petri Nets.

↓

The properties discussed above can be formally verified.

- **Petri nets (like dataflow systems) are *asynchronous concurrent*.**
  - **Events can happen at any time.**
  - **There exists a partial order of events.**

Extended Petri Net Models

Basic Petri Net models have a limited expressive power.

Extended Petri Net Models

Basic Petri Net models have a limited expressive power.

- **Timed Petri Nets**
  - **Transitions have associated times (time intervals)**
  - **Tokens are carrying time stamps.**

  With timed Petri nets we can model the timing aspects

Extended Petri Net Models

Basic Petri Net models have a limited expressive power.

- Timed Petri Nets
  - Transitions have associated times (time intervals)
  - Tokens are carrying time stamps.

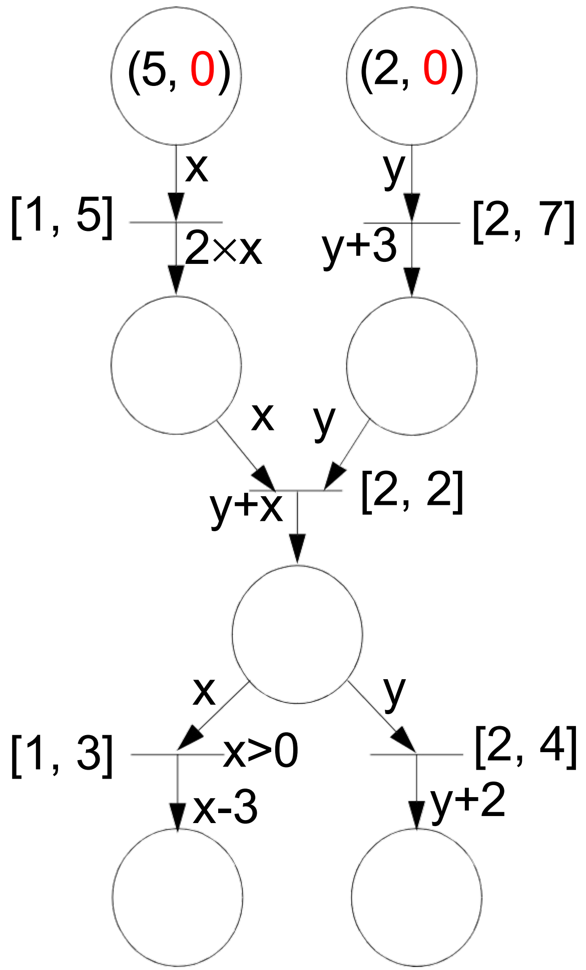
  With timed Petri nets we can model the timing aspects

- Coloured Petri Nets
  - Tokens have associated values
  - Transitions have associated functions

  Coloured Petri Nets are similar to dataflow models (but also capture nondeterminism!).

Extended Petri Net Models

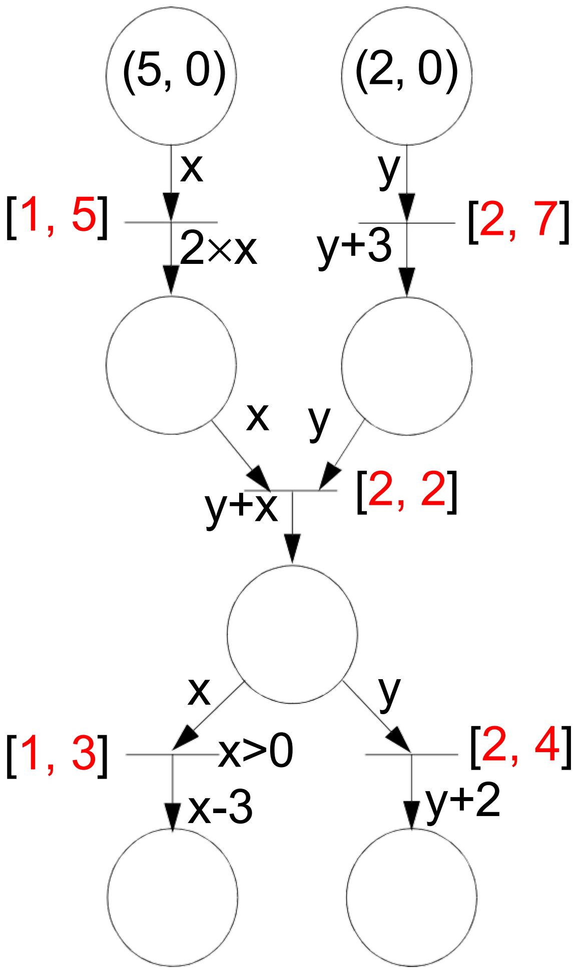
Coloured and Timed Petri net



- Tokens carry Time stamps

Extended Petri Net Models

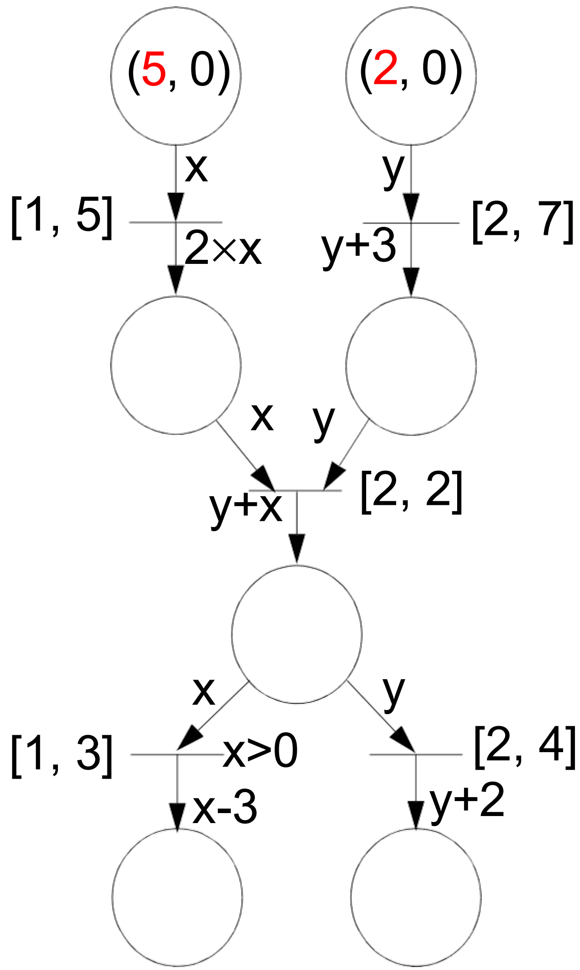
Coloured and Timed Petri net



- Tokens carry Time stamps
- Transitions have associated time (interval)

Extended Petri Net Models

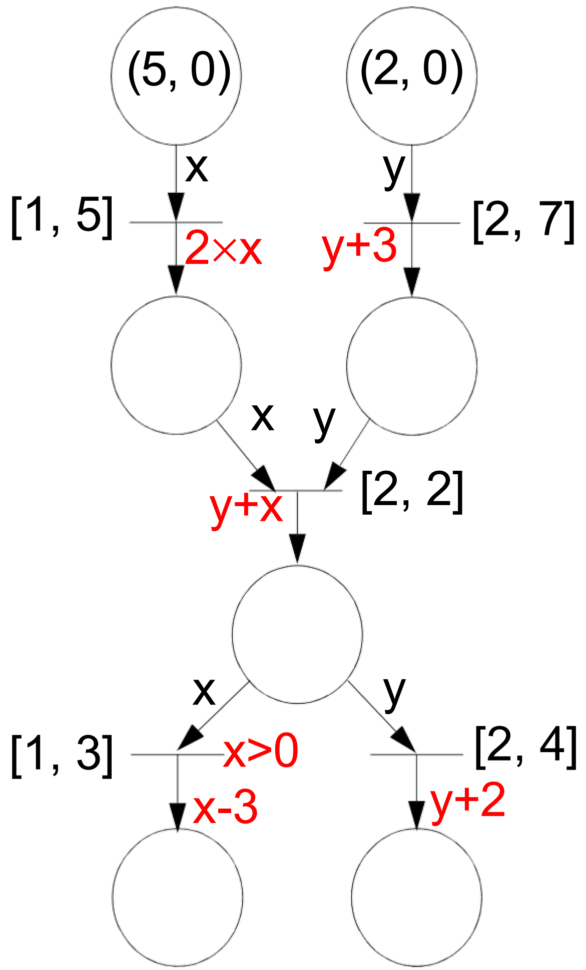
Coloured and Timed Petri net



- Tokens carry Time stamps
- Transitions have associated time (interval)
- Tokens have associated values

Extended Petri Net Models

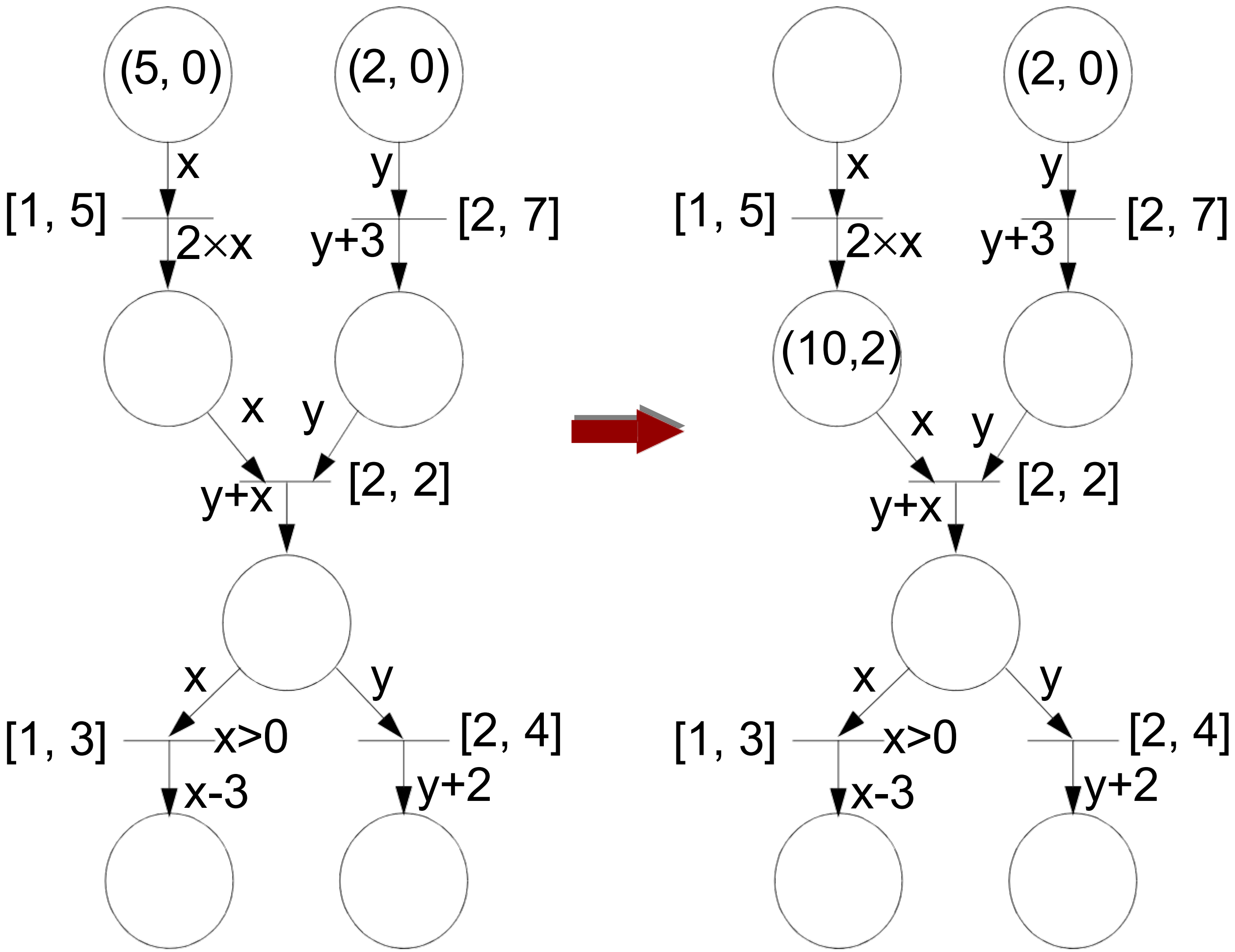
Coloured and Timed Petri net



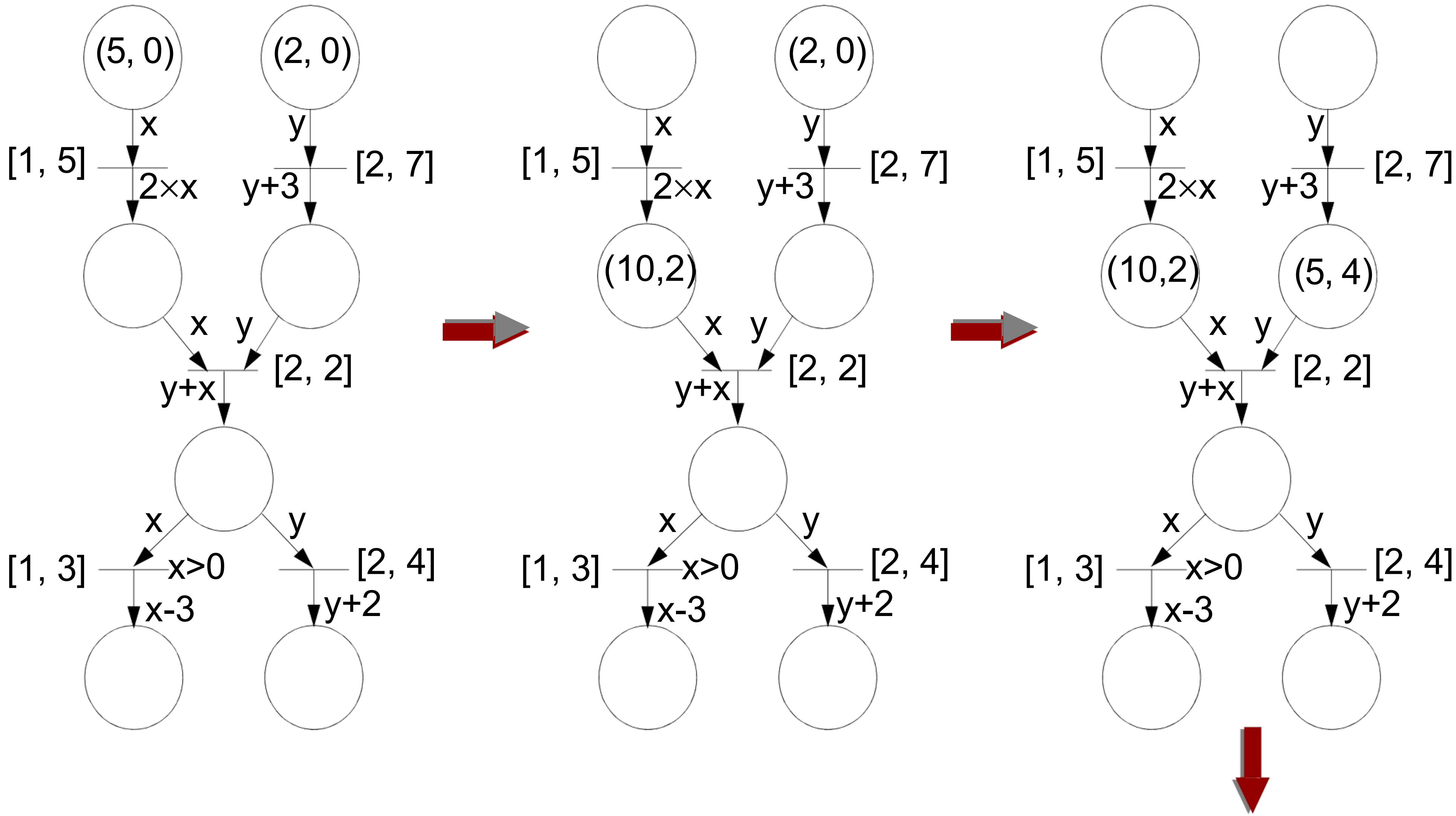
- Tokens carry Time stamps
- Transitions have associated time (interval)

- Tokens have associated values
- Transitions have associated functions and guards

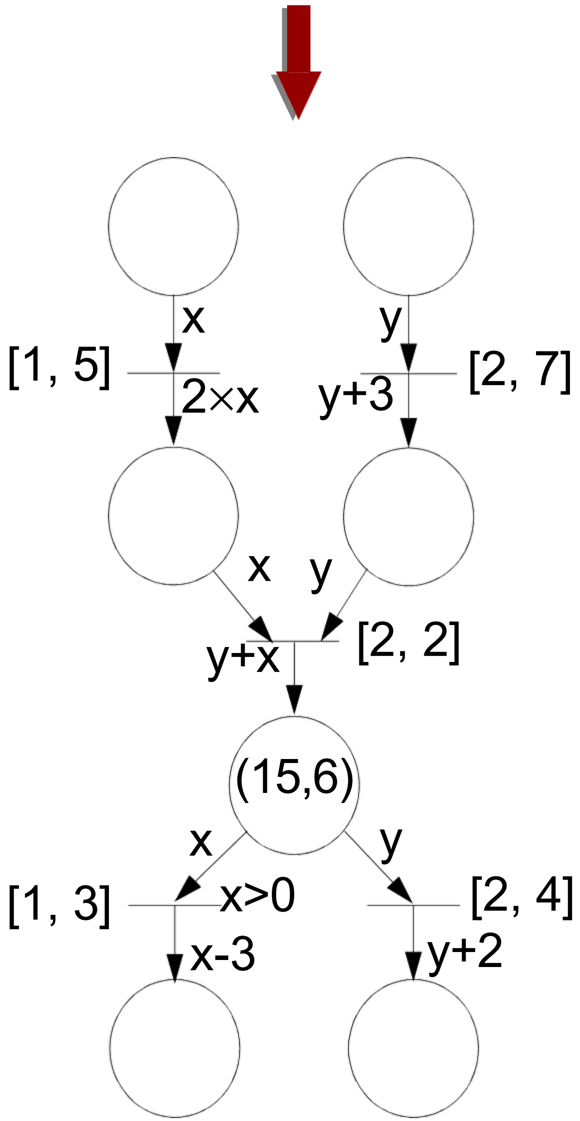
Extended Petri Net Models



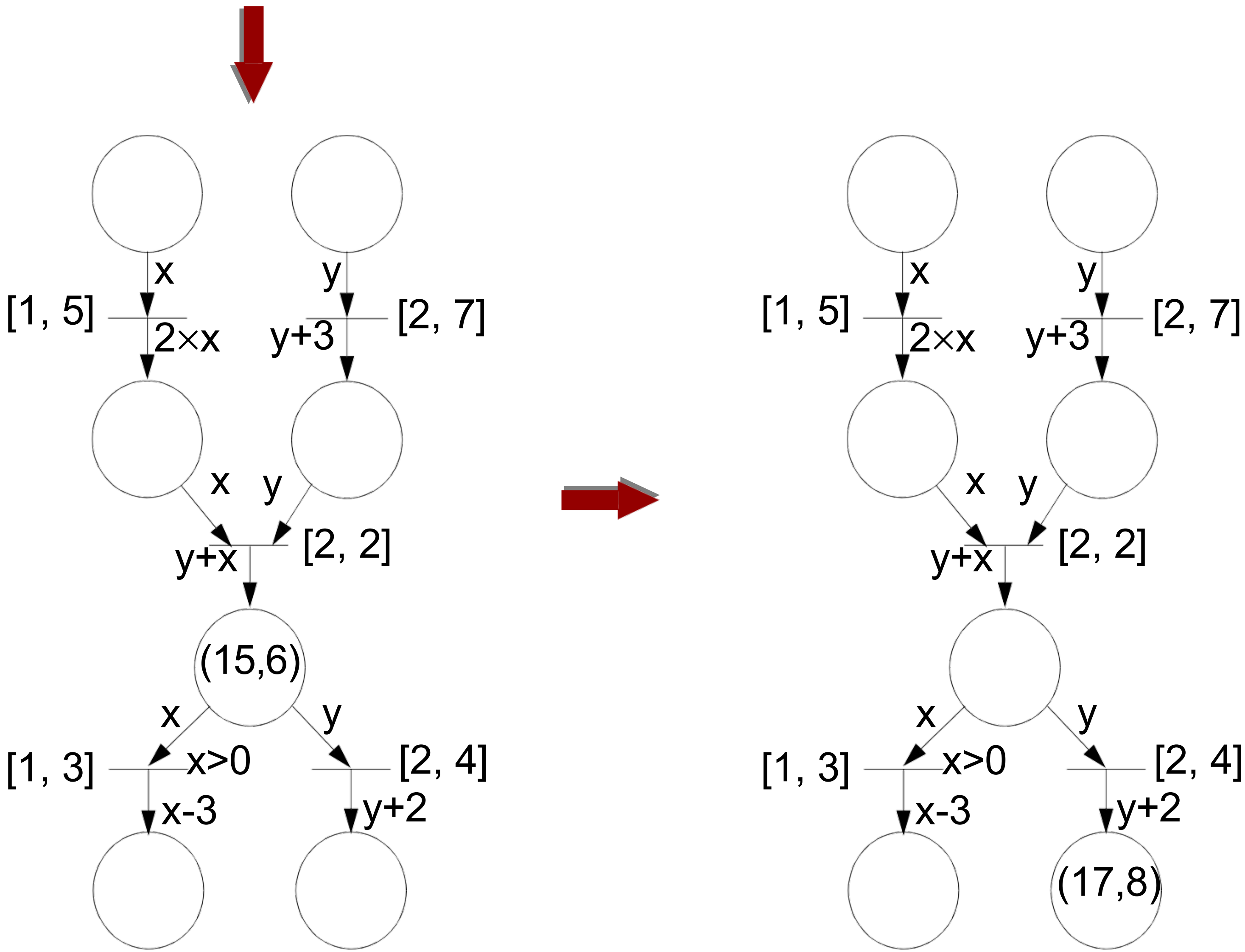
Extended Petri Net Models



Extended Petri Net Models



Extended Petri Net Models



Extended Petri Net Models

- Extended Petri Nets have a larger expressive power then classical Petri Nets.

  ⬇

  Analysis is more complex; the formal analysis of properties can take very large amounts of time (memory).

- Simulation of the Petri Net is very often used in order to verify the system and to estimate performance