

# Cyclic Communicating Processes: *Controller Synthesis*

P.S. Thiagarajan

National University of Singapore

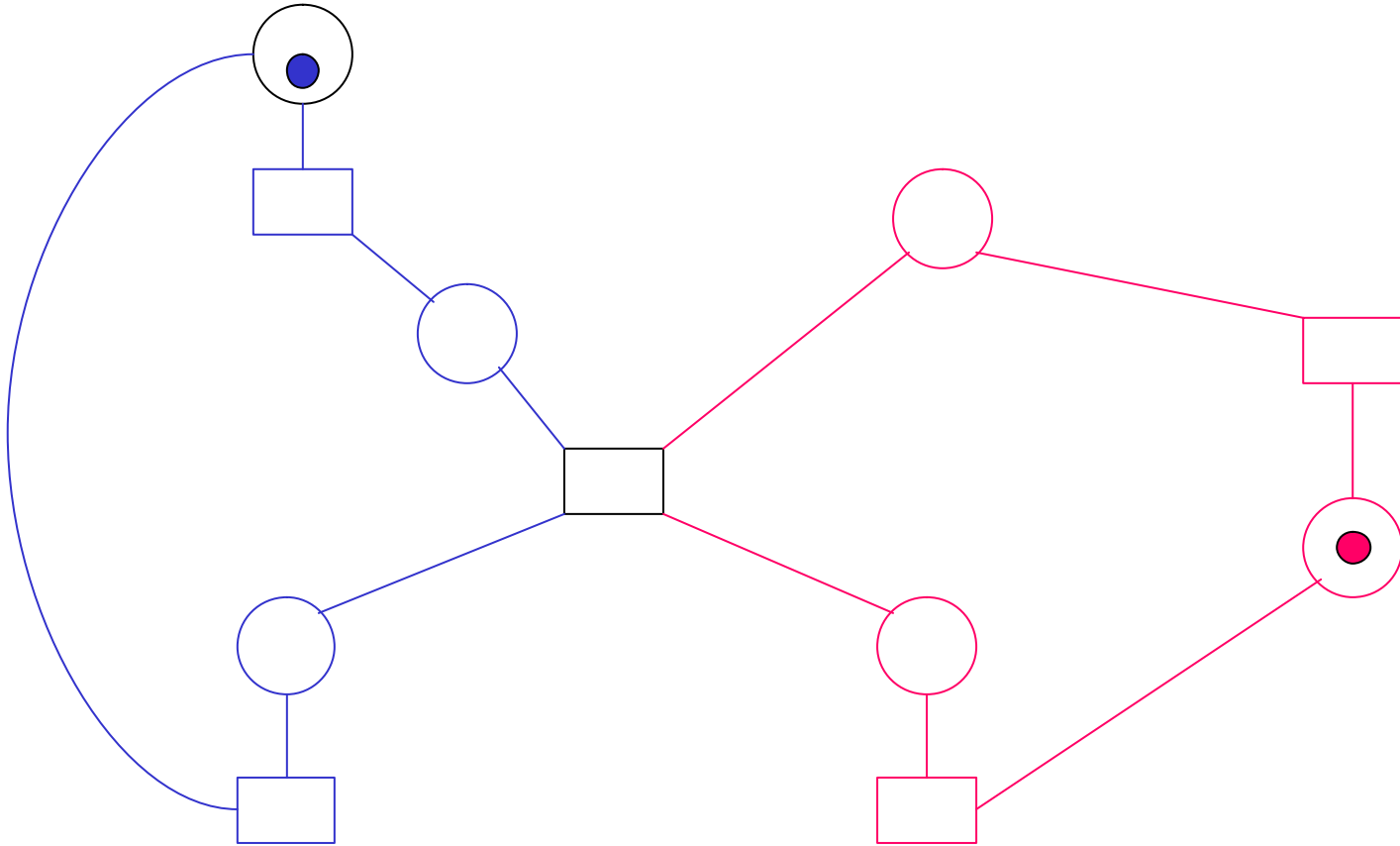
# CCPs

- A network of processes (**components, agents**).
- For each process, the control flow is *cyclic*.
- State of a process :
  - (control state, **data values**)
    - *finite abstraction* of data values.
- Interaction between processes:
  - Both control states and data values are changed.
- In net terms:
  - **1-safe colored marked graphs**.

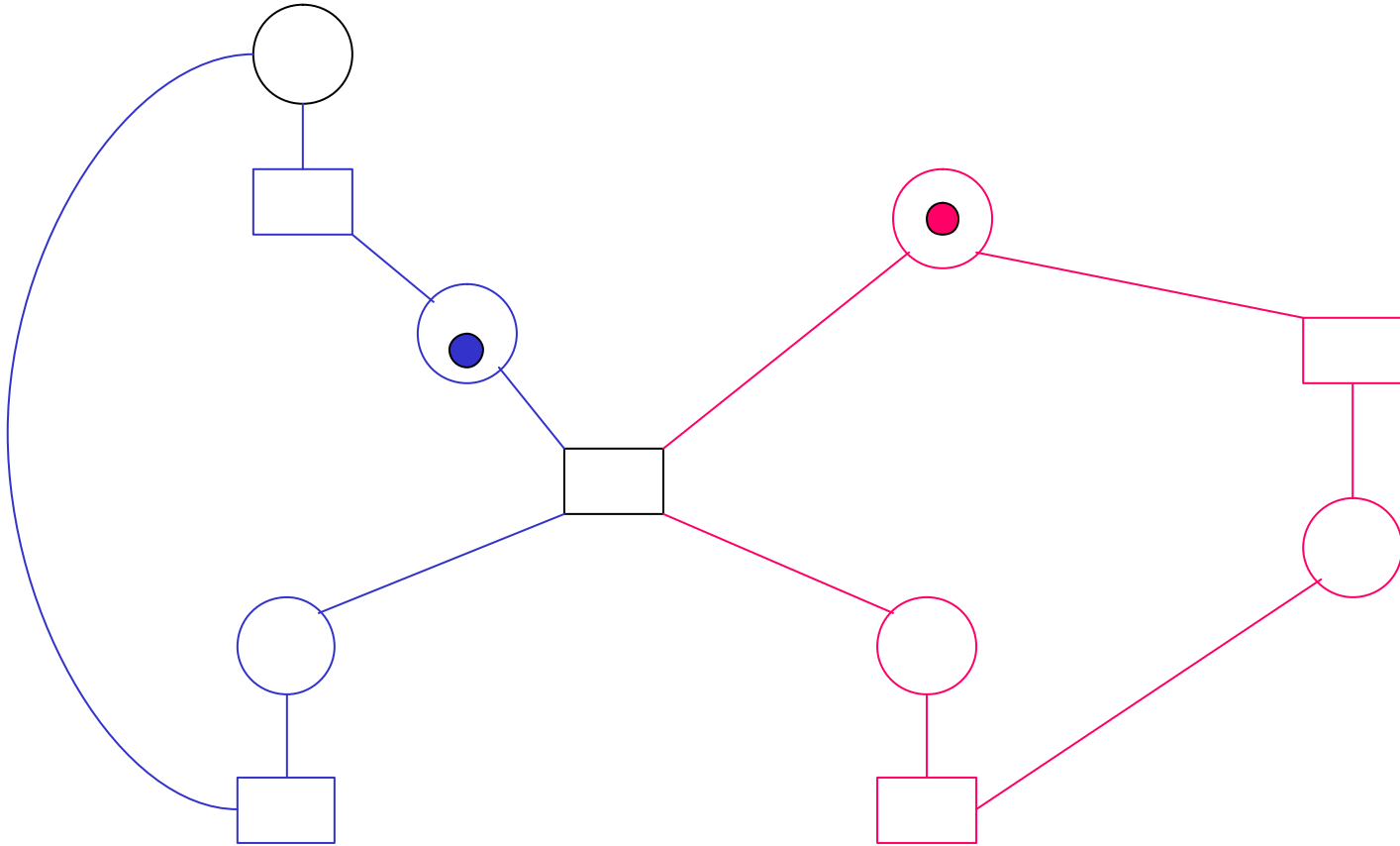
# Appealing Features

- A good deal of modeling power.
  - To be validated !
- Supports **hierarchy**.
- Powerful theory:
  - Formal verification
  - *Controller synthesis*.

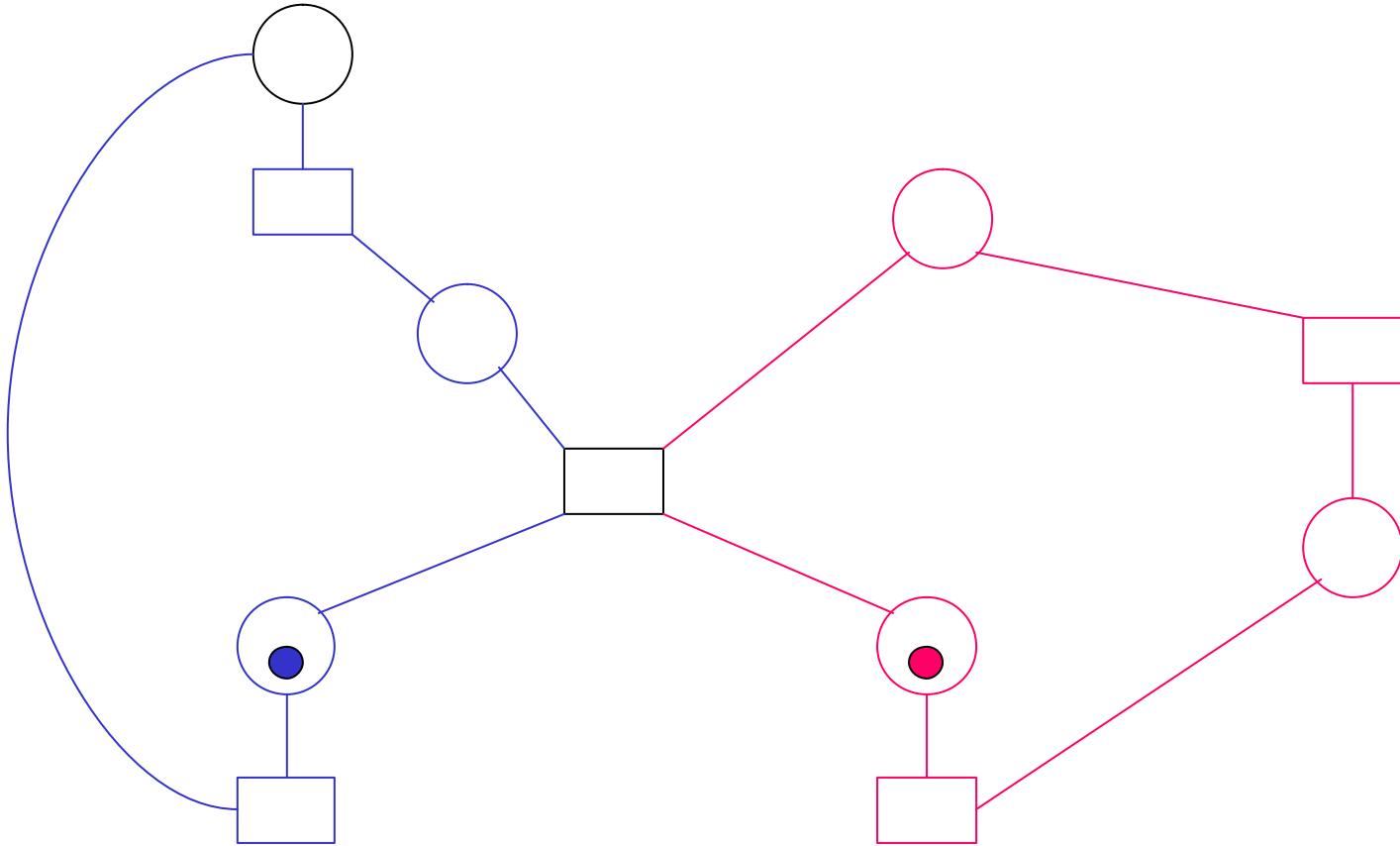
# The Model: Control Flow



# The Model: Control Flow



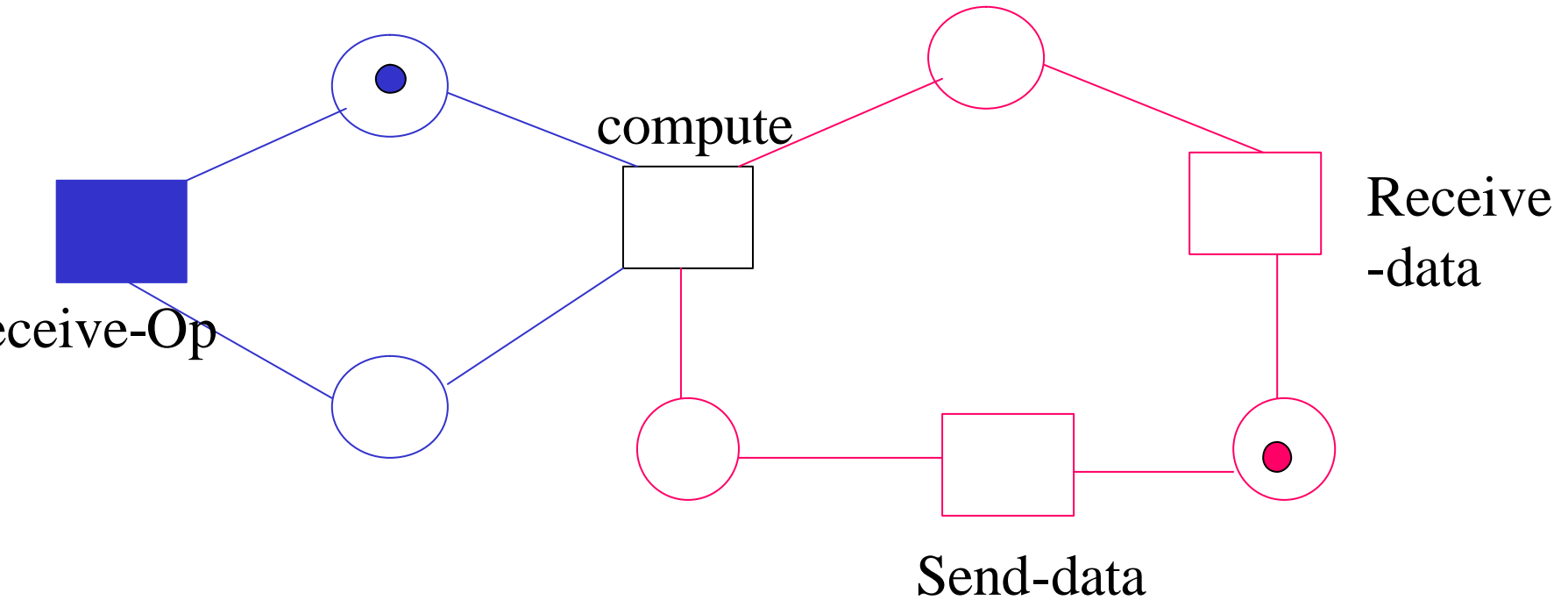
# The Model: Control Flow



# The Model: Data Transformation

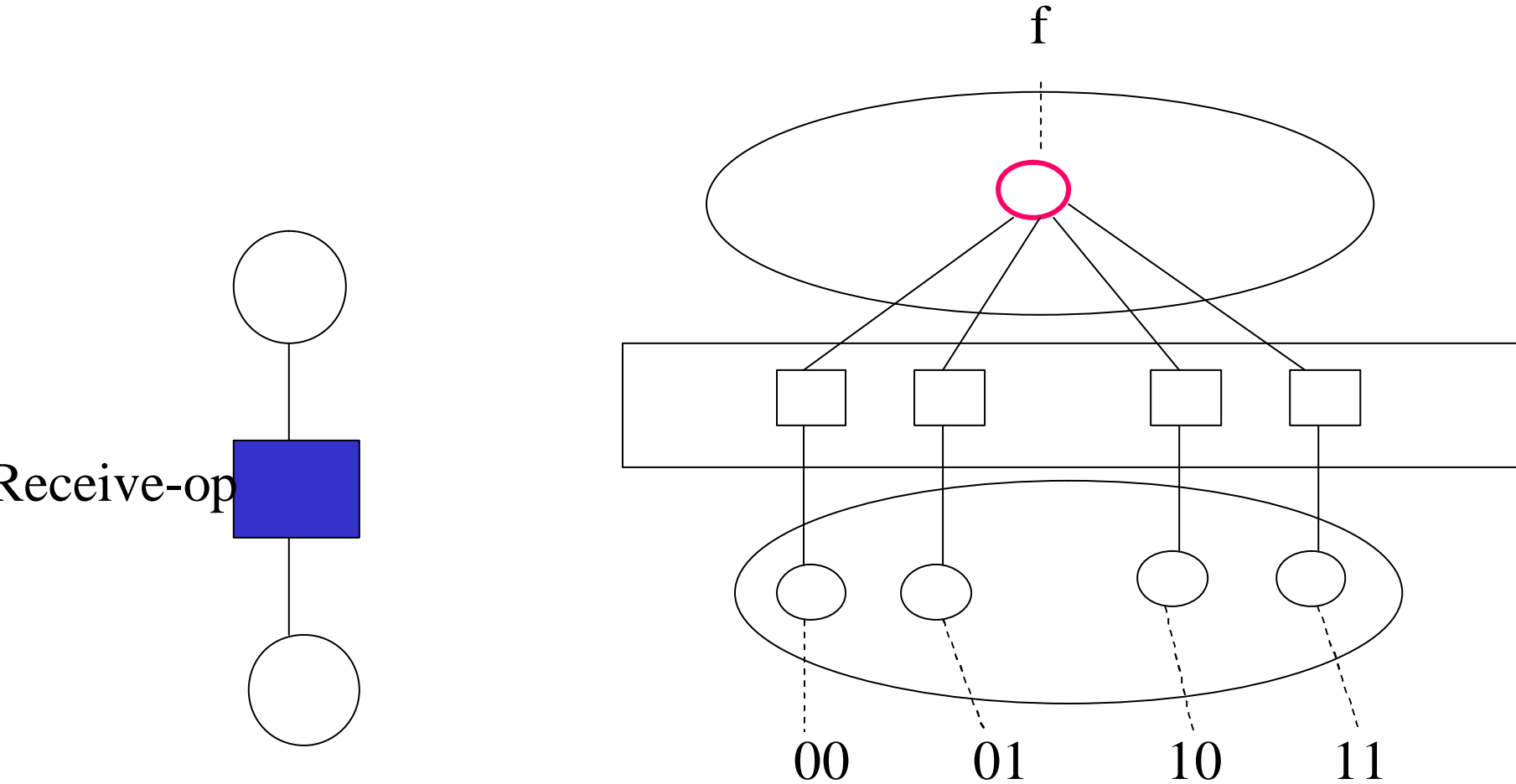
- $P = \{p, q, r, \dots\}$
- $p \dashv\vdash \{x_{p1}, x_{p2}, \dots, x_{pn}\}$ 
  - The data variables of  $p$ .
- $D1, D2, \dots, Dn$ 
  - The data domains (values)
    - $x_{p1} \in D1$
  - Each data domain is finite ! (infinite?)

# Example

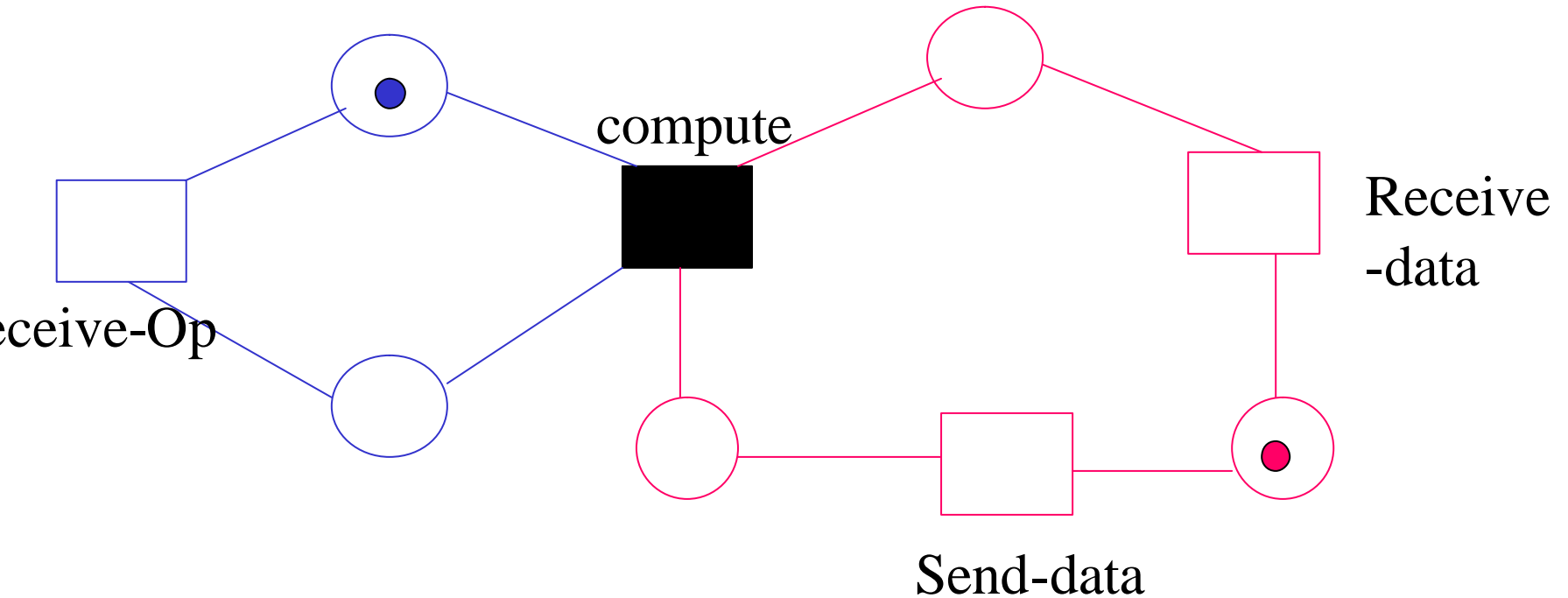




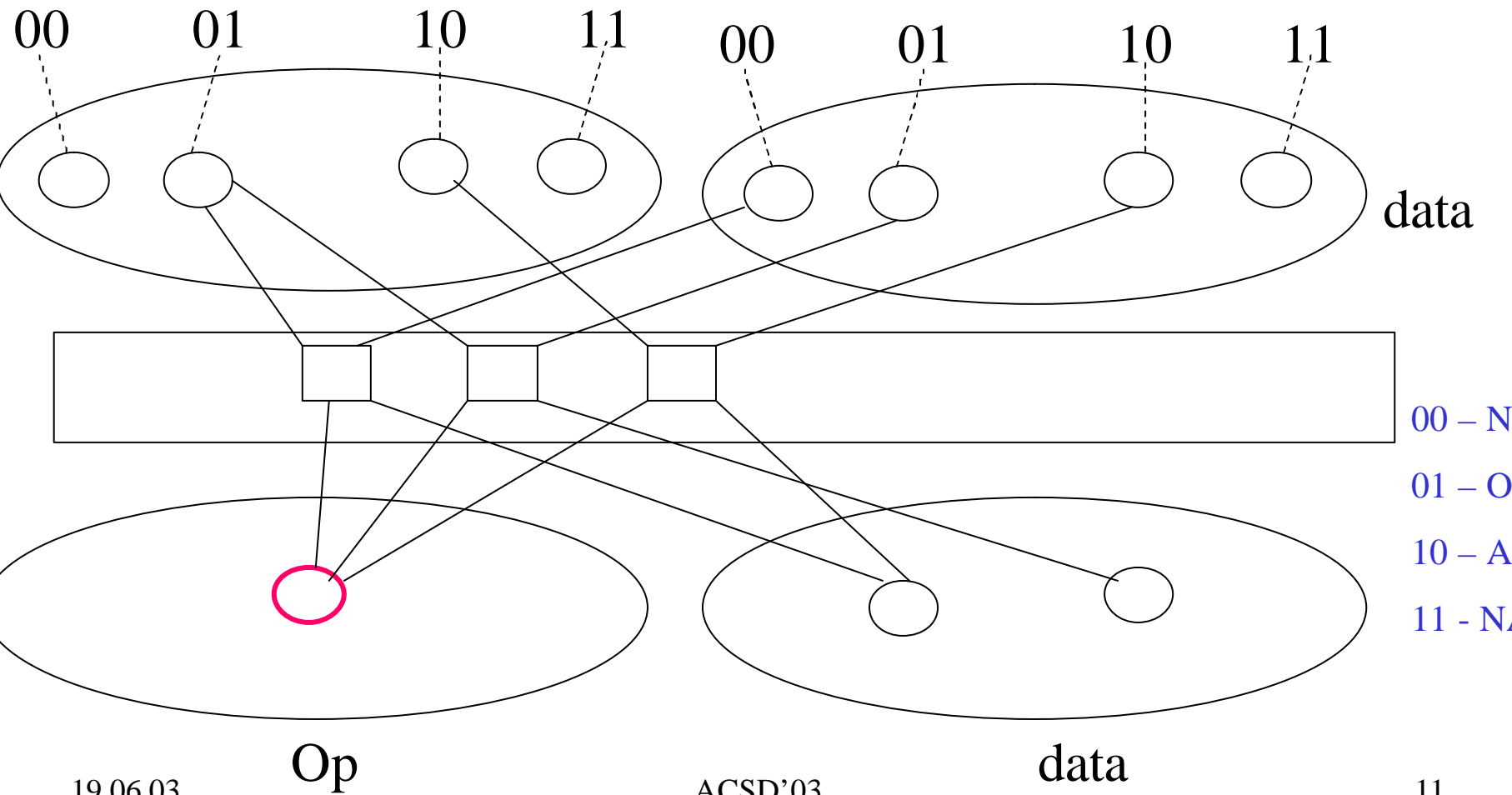
# Example



# Example



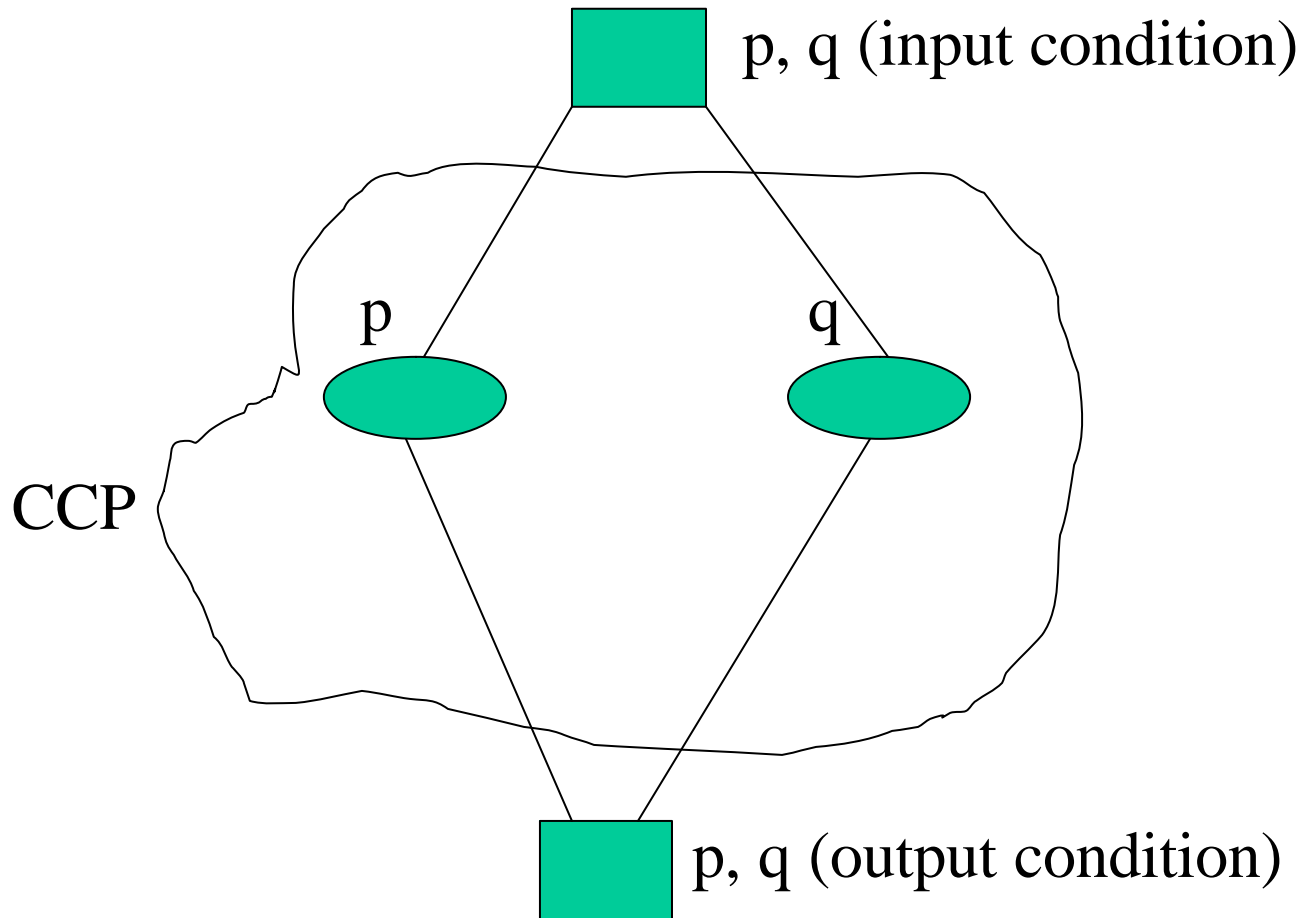
# The Model: Data Transformation



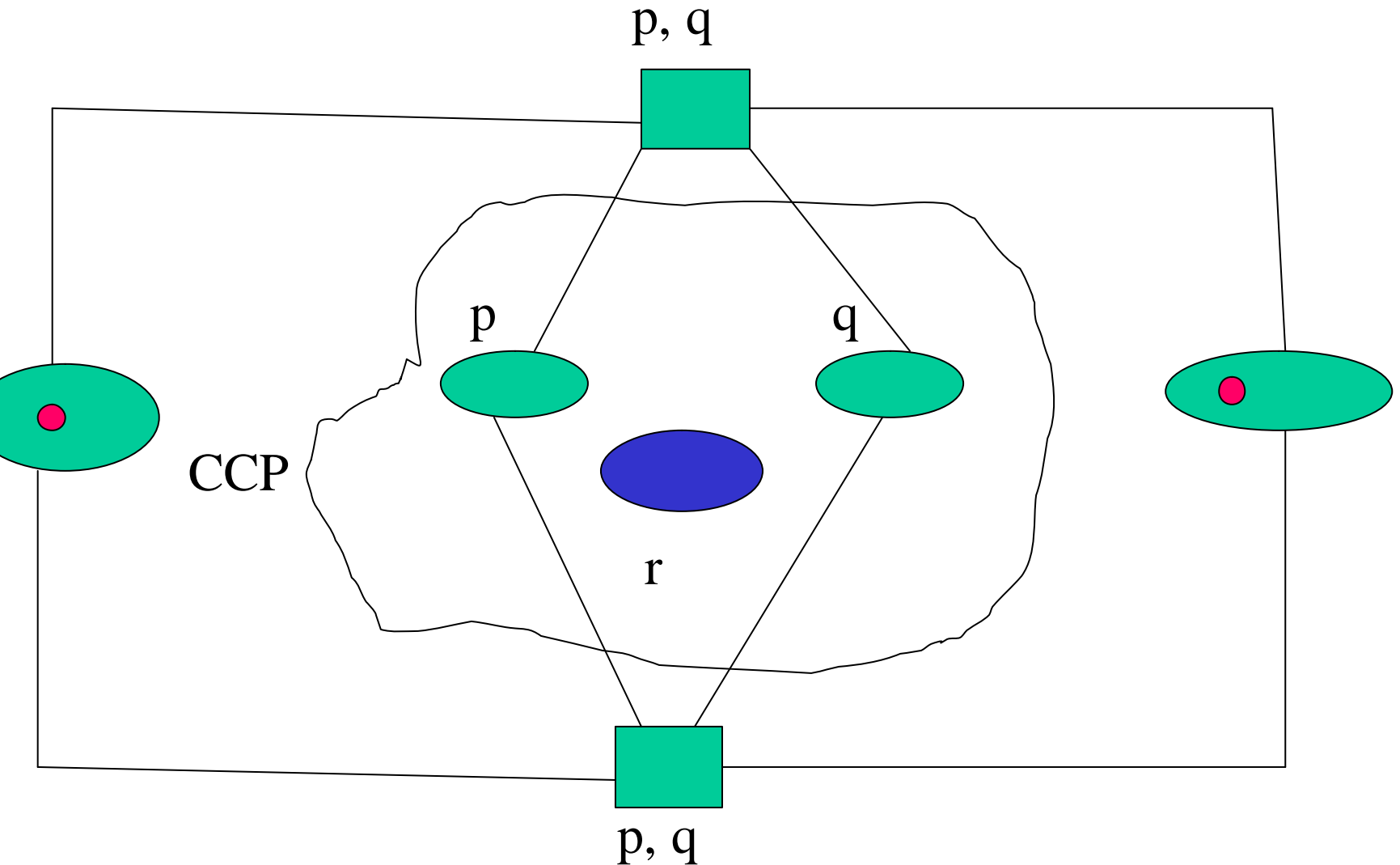
# The Model

- A *subclass* of Colored Petri nets.
- Well-defined notion of **components**.
- Two level description:
  - **Control flow**:
    - a 1-safe marked graph.
  - **Data flow**:
    - a 1-safe Petri net,
    - structurally very complex
    - behaviorally, a 1-safe free choice Petri net.

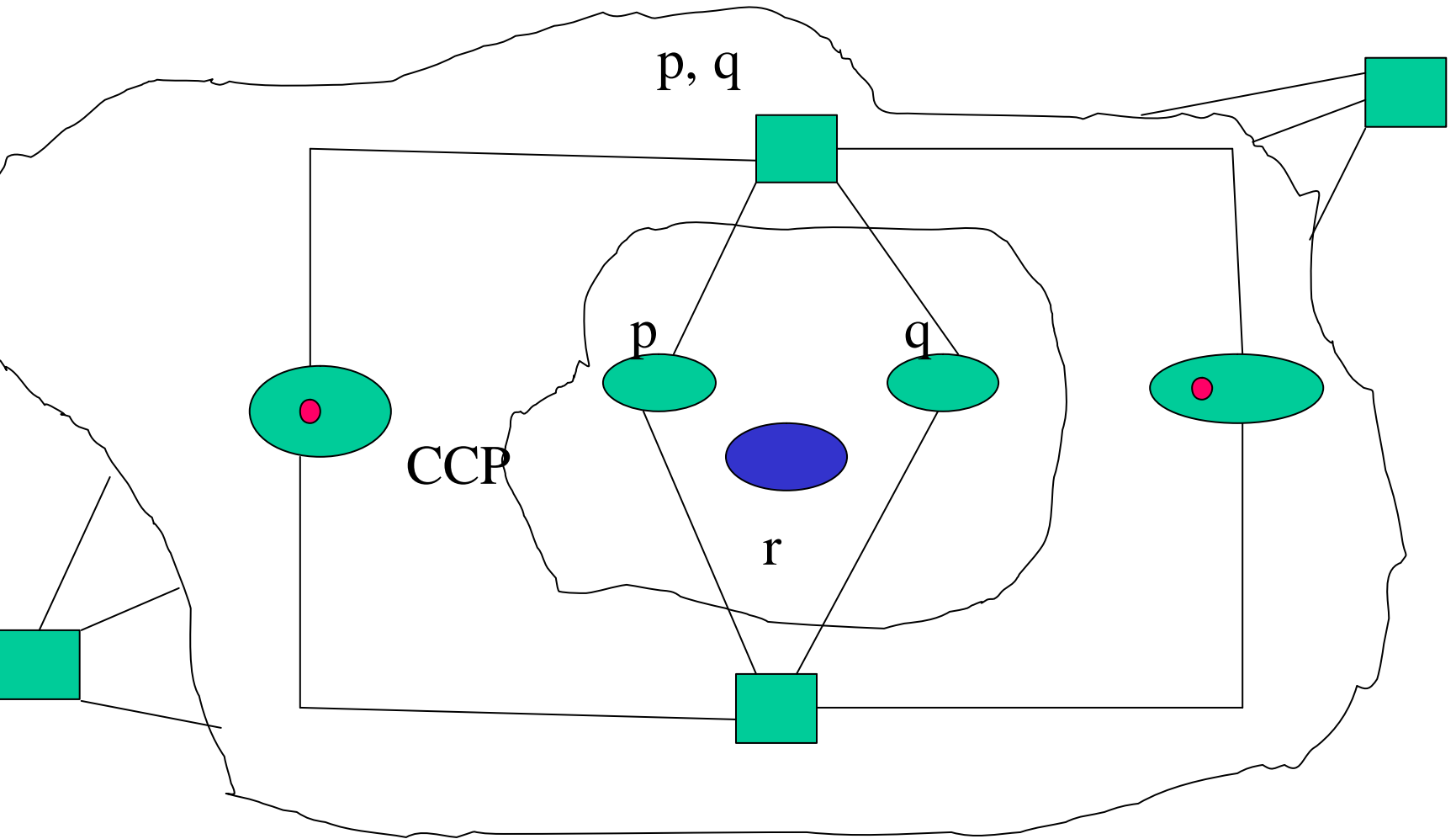
# Hierarchical Descriptions.

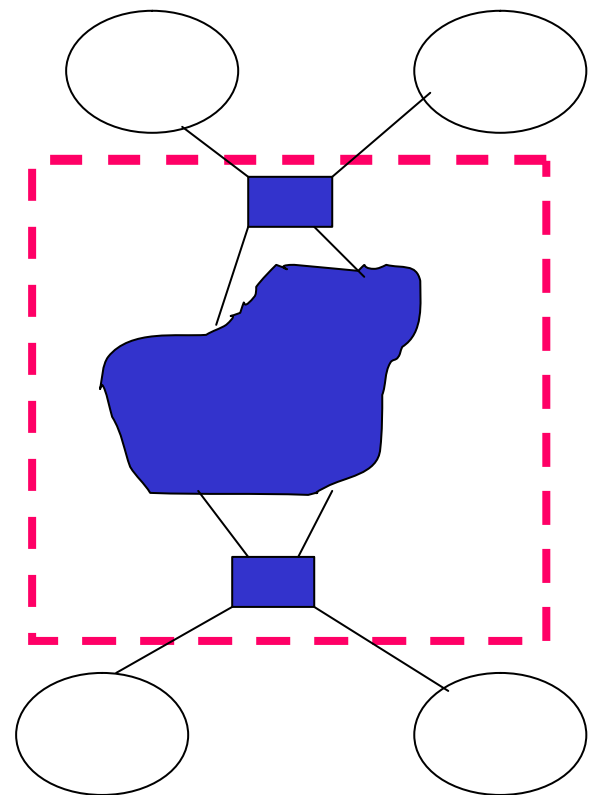
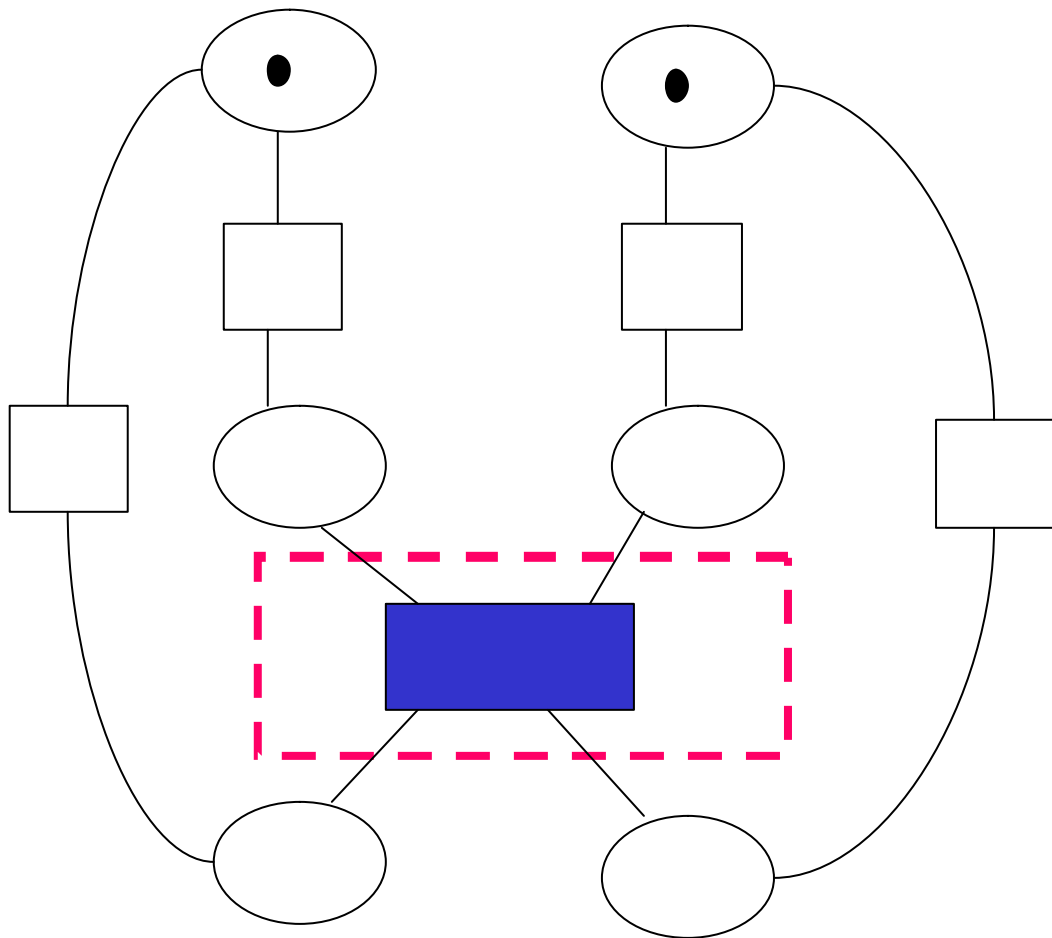


# Hierarchical Descriptions.



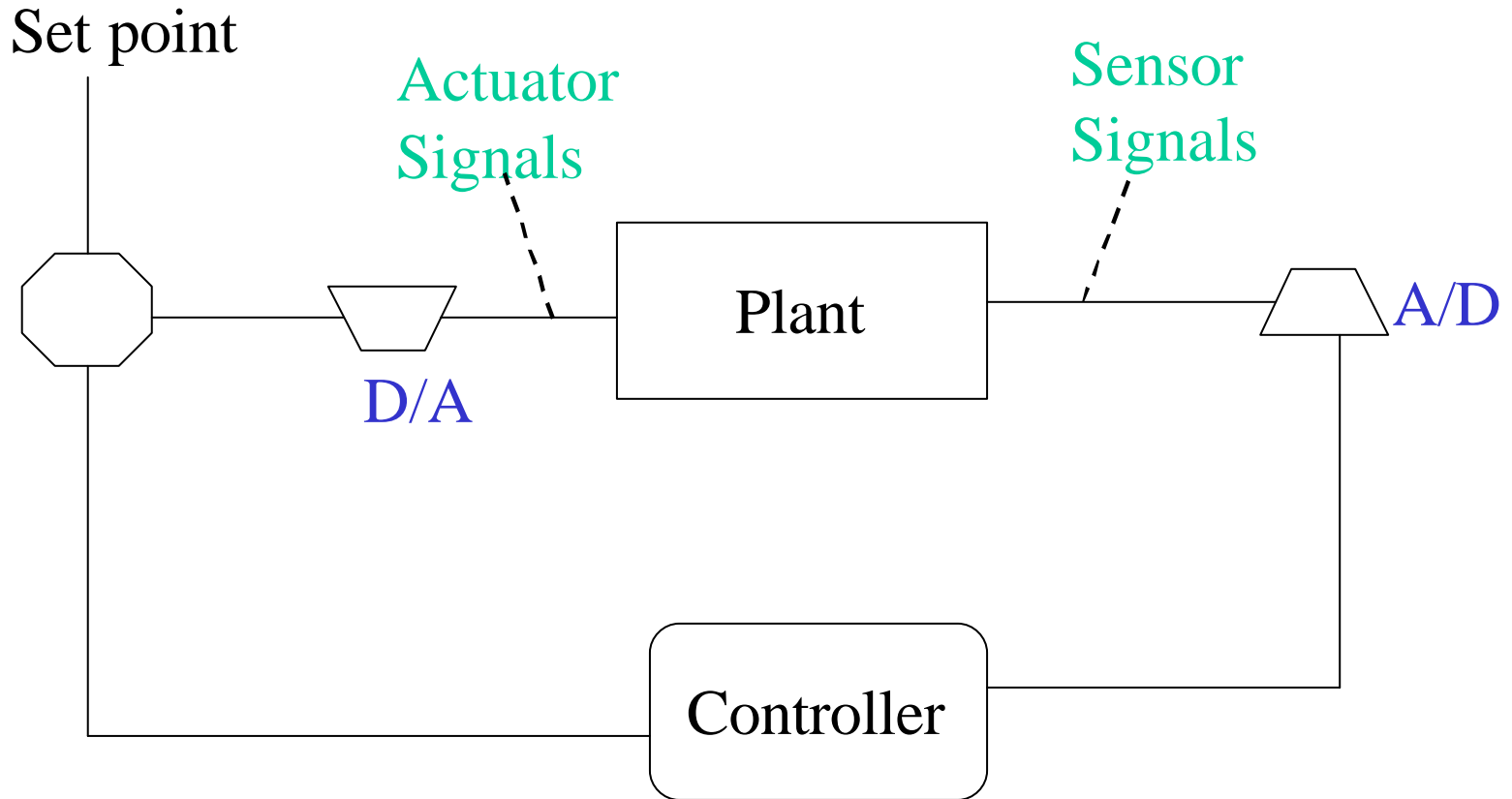
# Hierarchical Descriptions.

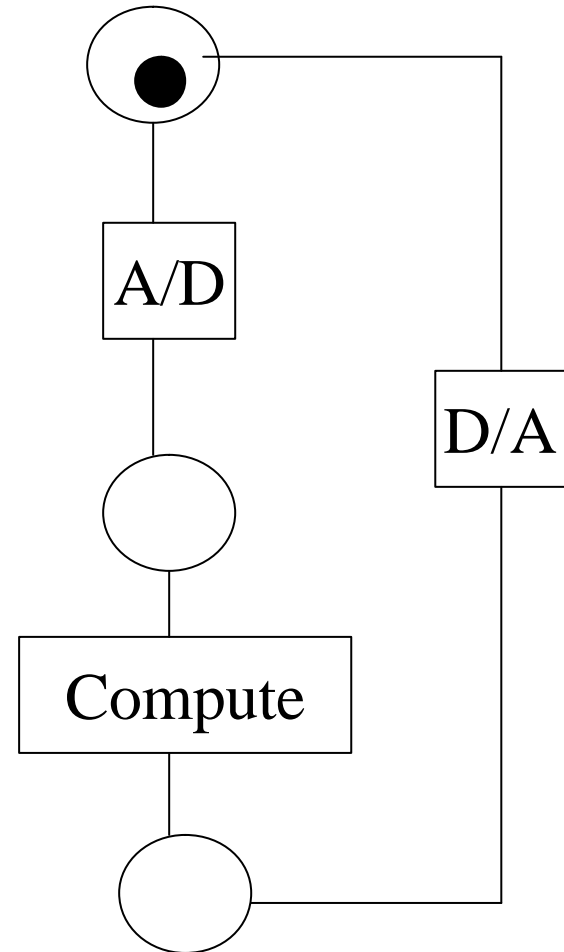
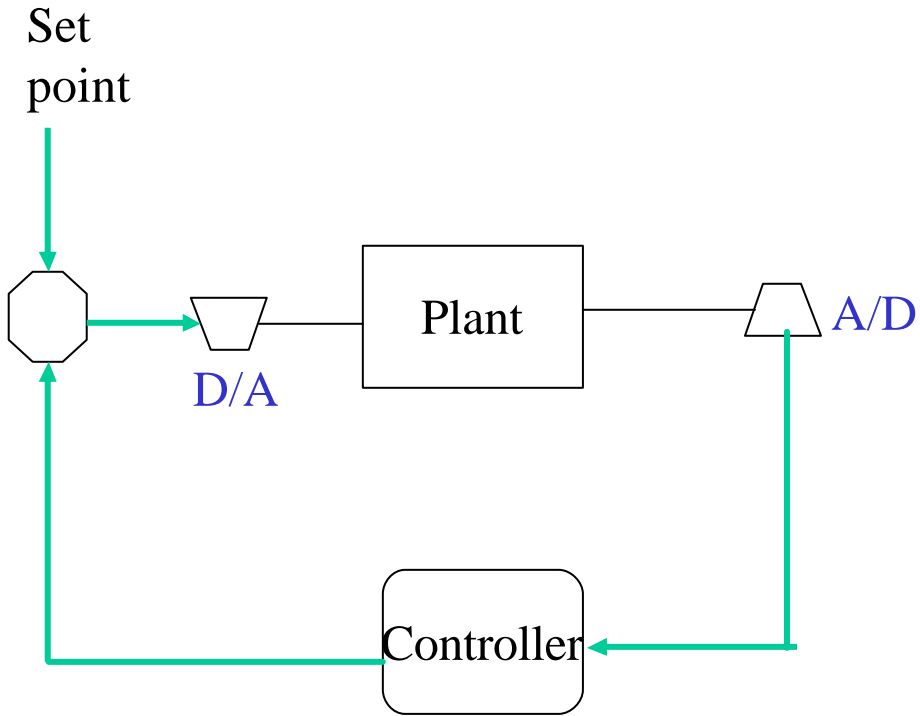




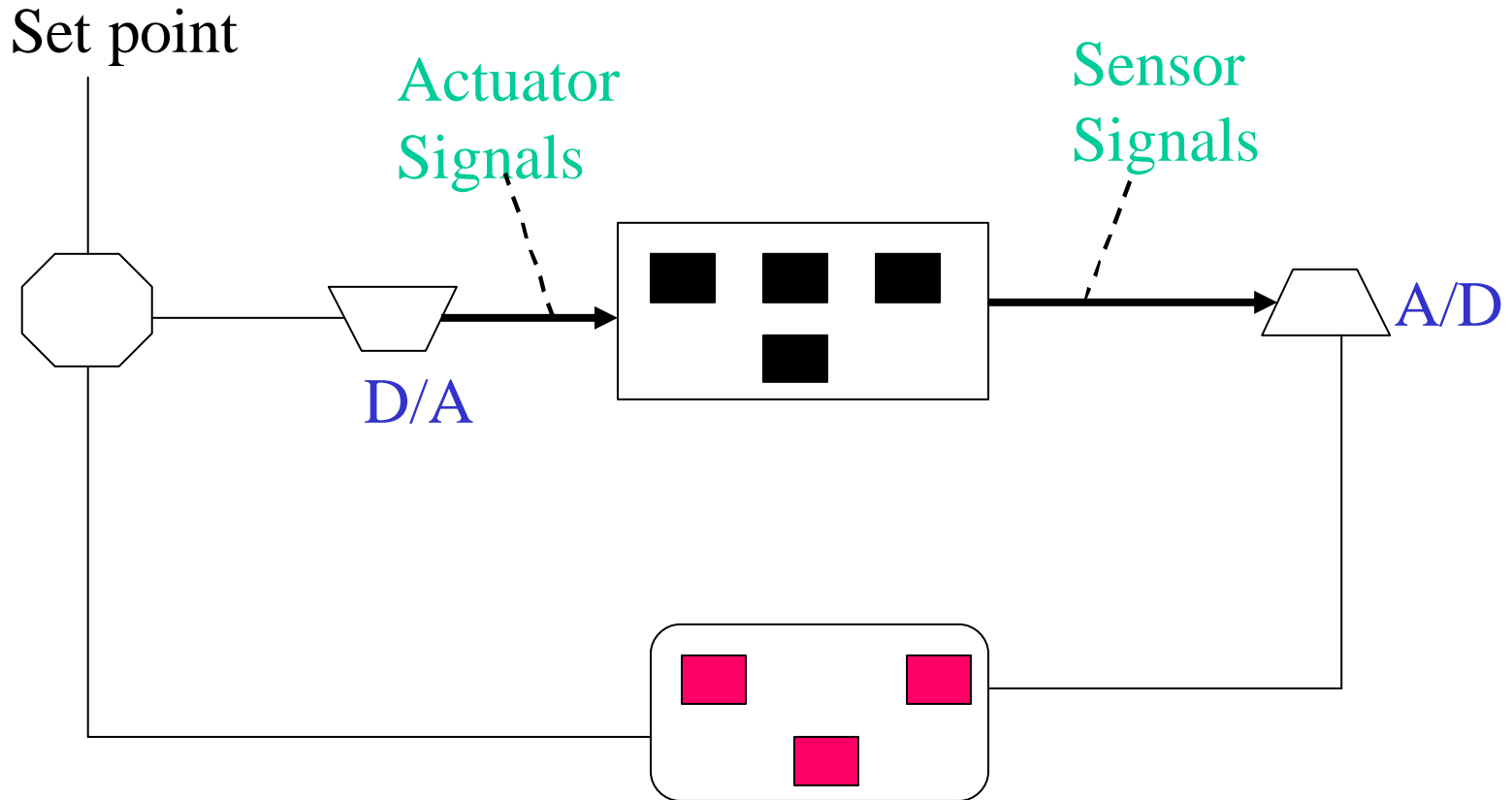


# Applications





# Distributed Plants and Controllers



# Multi-Level Networks

- Similar to boolean networks but gates are:
  - Non-deterministic (Relational)
  - Multi-valued
  - Succinct descriptions of boolean networks.
- **A Theory of Non-Deterministic Networks:**
  - Alan Mishchenko and Robert Brayton.
- MVSIS Group, UC Berkeley:
- <http://www-cad.eecs.berkeley/mvsi>

# Multi-Level Networks

- So far only combinational circuits.
- CCPs are a natural model for a subclass of “sequential” multi-level networks.
- Can model multi-level asynchronous circuits.
- Extension of MG-based Signal Transition Graphs ?

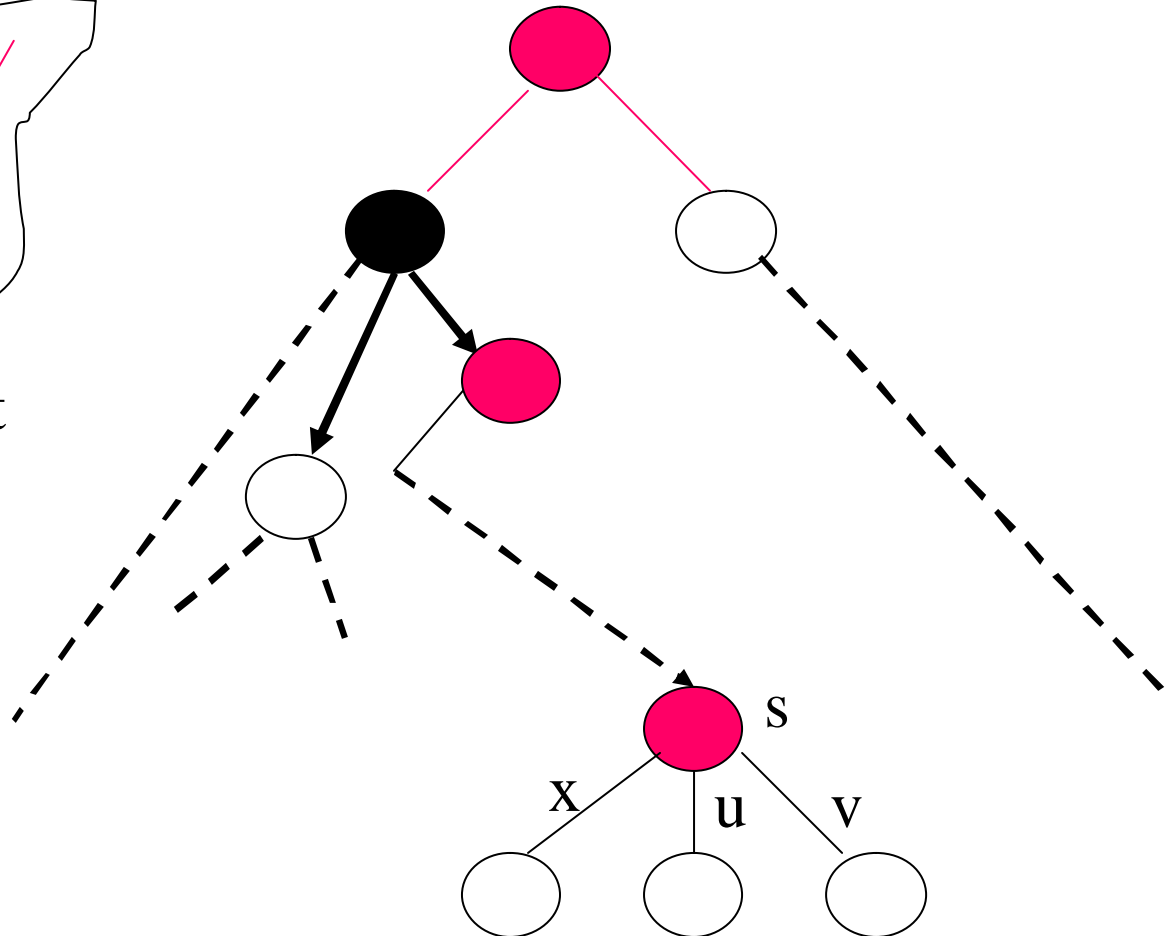
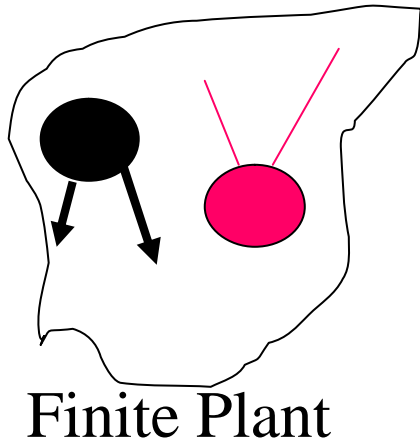
# Verification

- $CCP \neq ES_{CCP} \neq TS_{ES}$
- We can model-check  $TS_{ES}$  against the full MSO specifications.
- We can model-check TS against all branching time temporal logics:
  - CTL, CTL\* ....
- Efficient CCP-specific procedures?

# Controller Synthesis

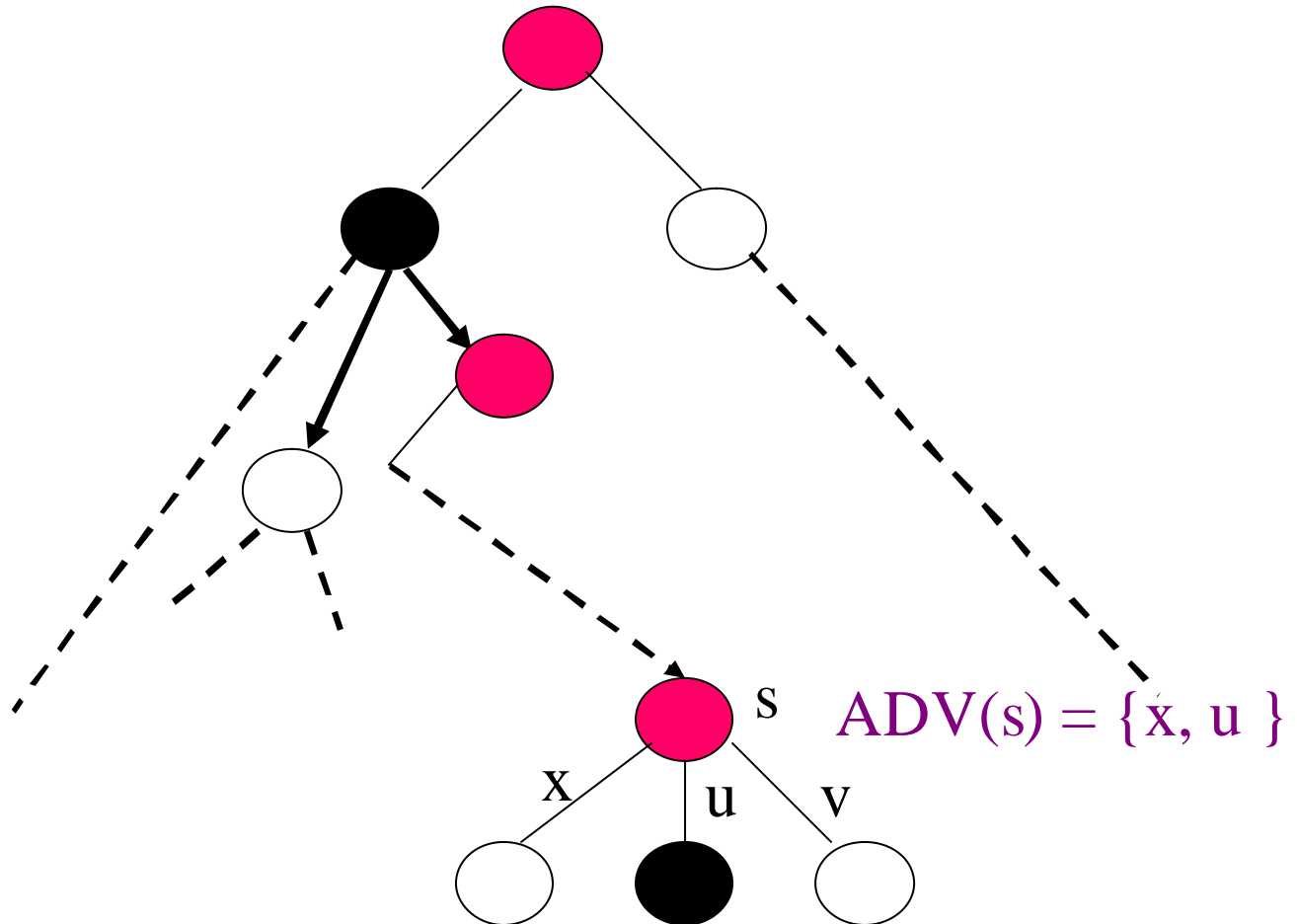
- We can solve powerful *distributed* controller synthesis problems for CCPs.
- Controller synthesis:
  - Well-established theory for *sequential* discrete event systems.
  - **Very hard** in distributed settings.
  - An approach to handle:
    - non-determinism, don't care conditions, program synthesis.

# Controller Synthesis

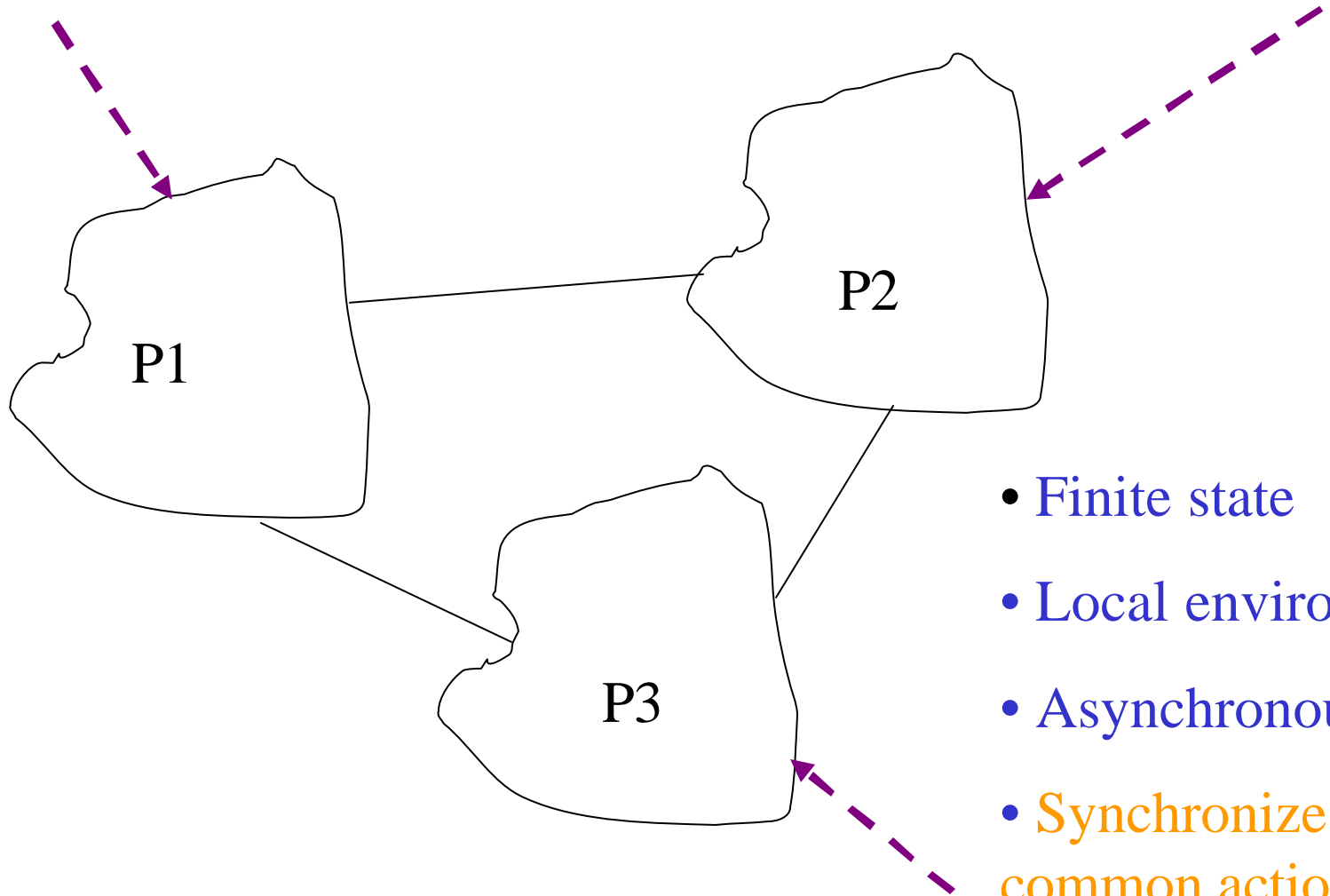




# Strategies

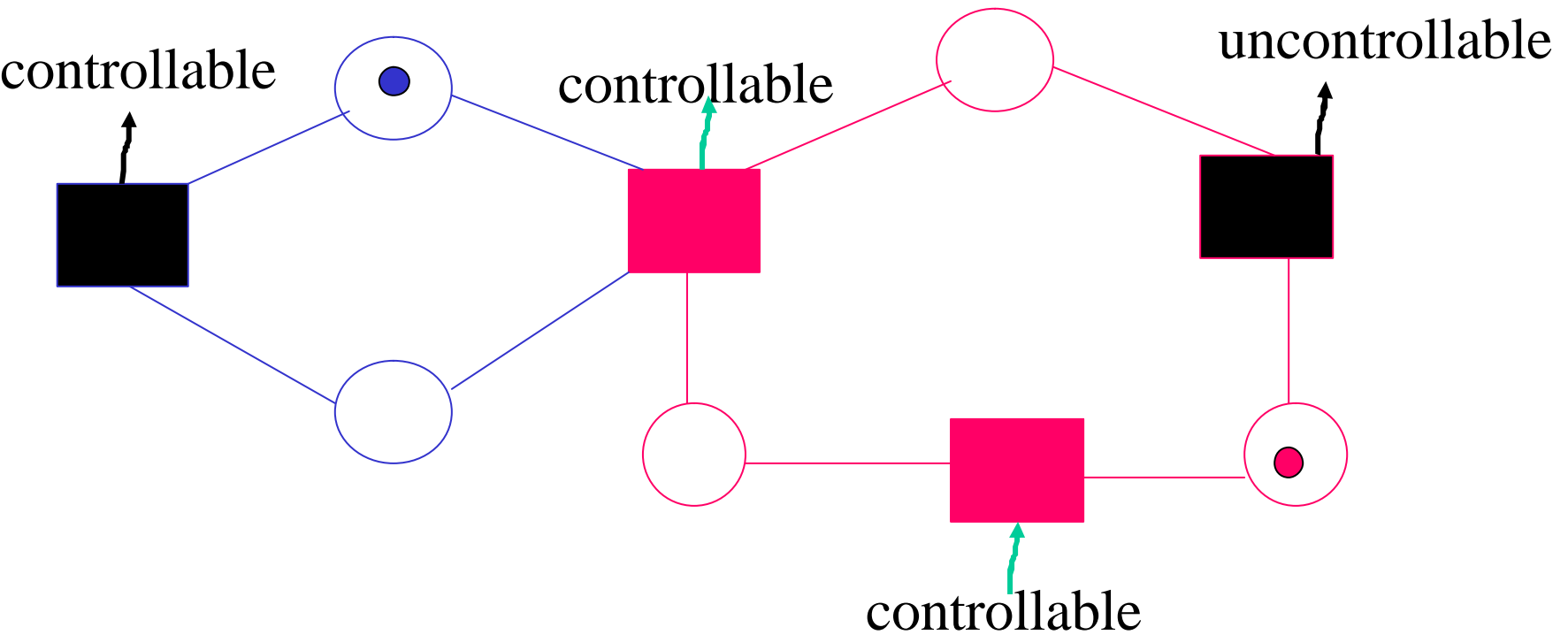


# Distributed Controller Synthesis



- Finite state
- Local environments
- Asynchronous
- Synchronize on common actions.

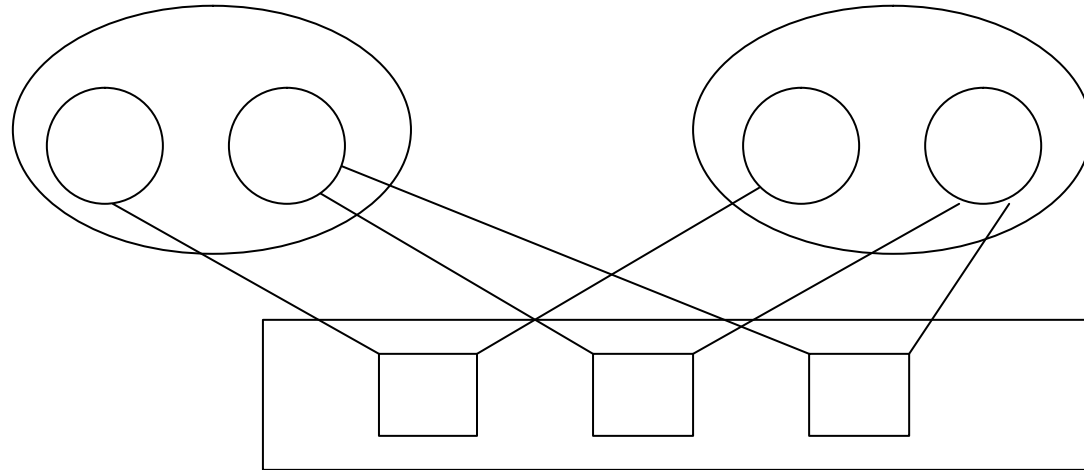
# Controller Synthesis



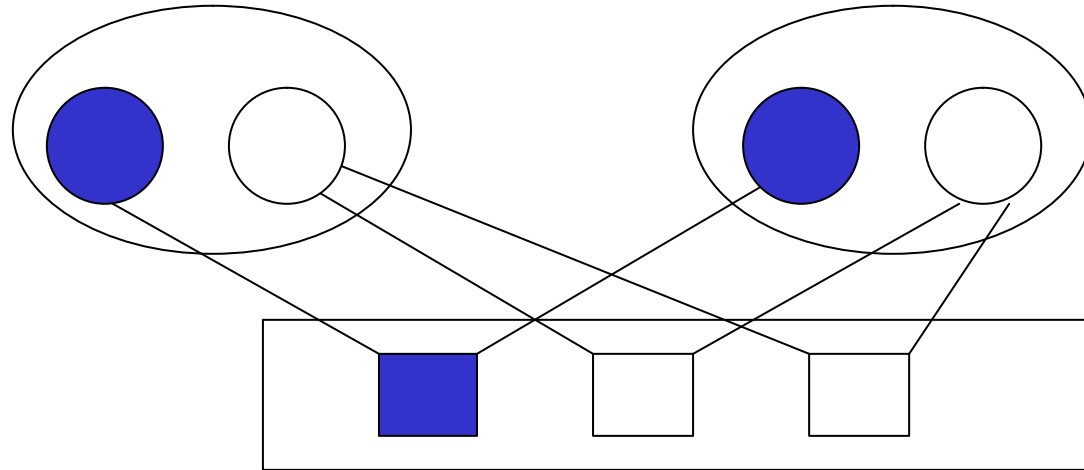
# Controller Synthesis

- Find *local strategies*, one for each component which, together, win against the environments.
  - Can remember only its history and knowledge gained through communications.
- Winning Local strategies ---- **Distributed controller**.

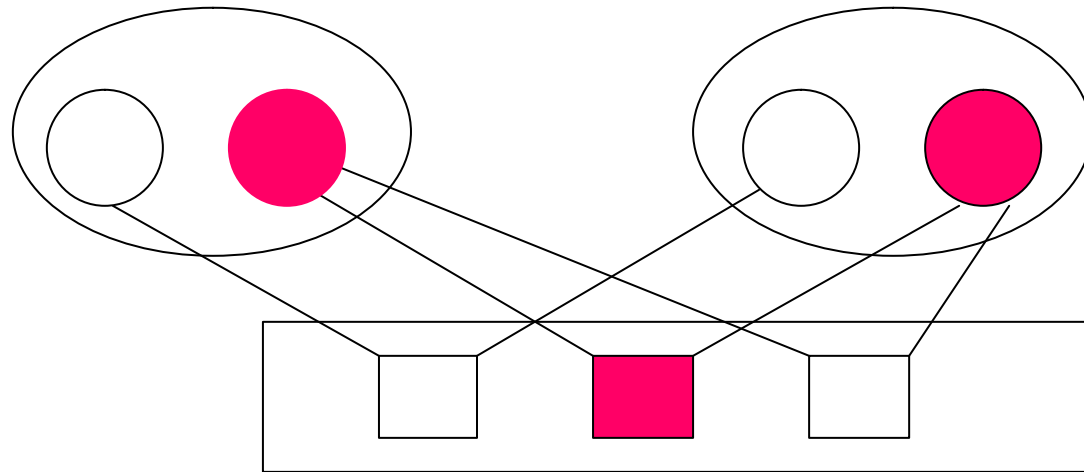
# Local Strategies



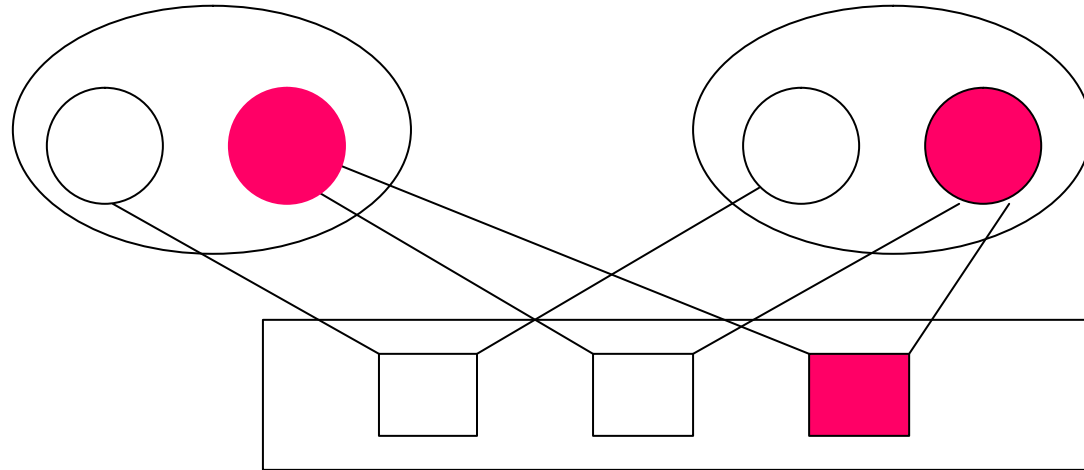
# Local Strategies



# Local Strategies

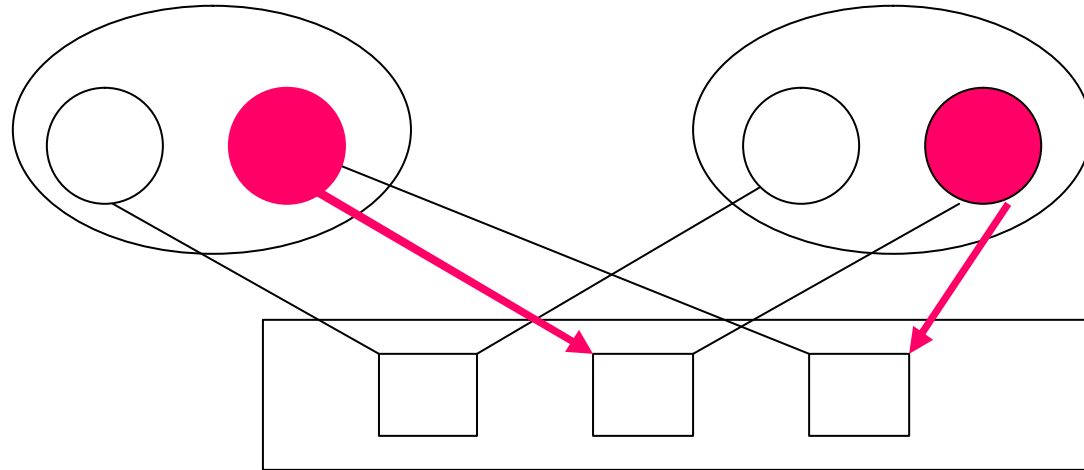


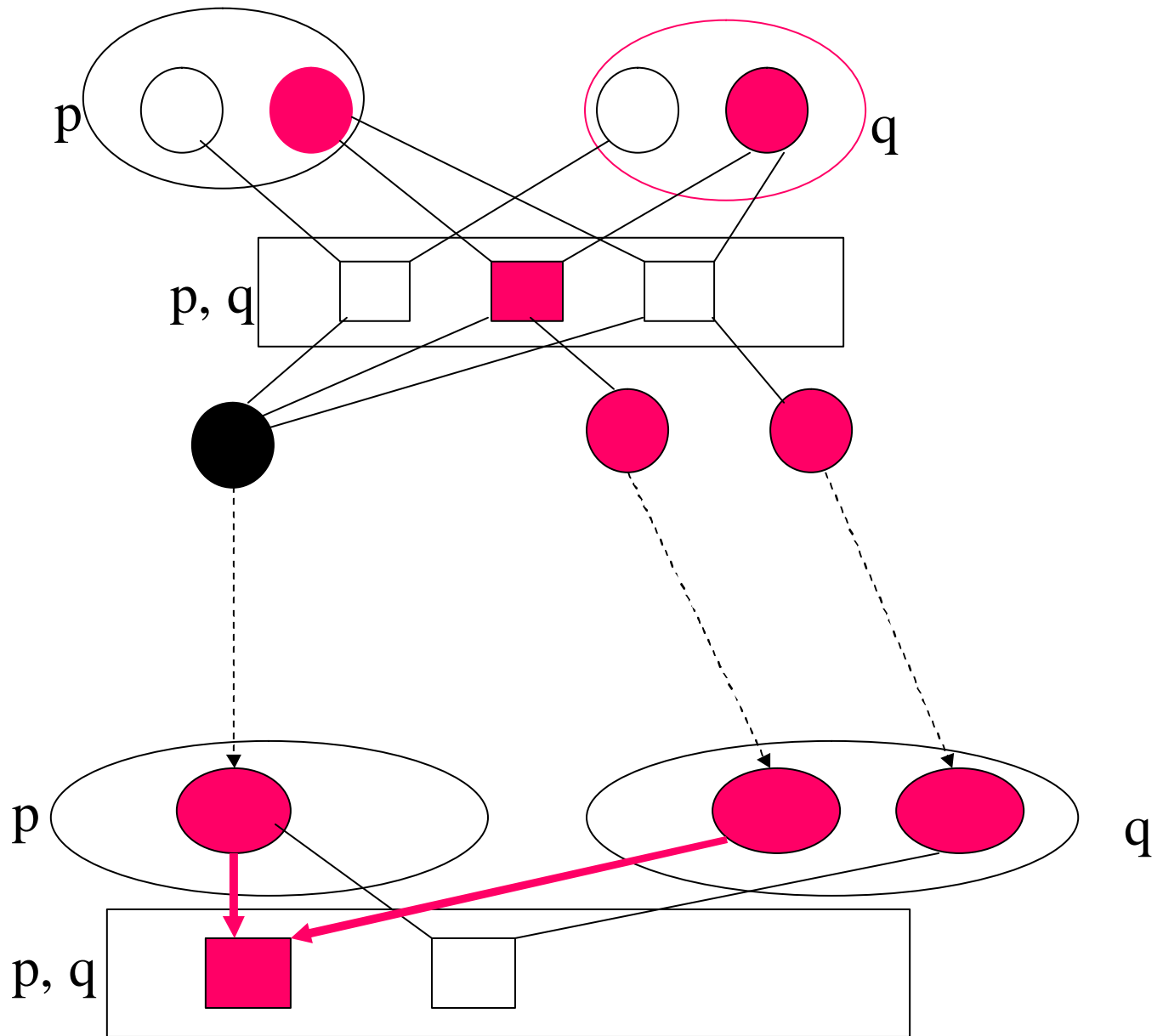
# Local Strategies

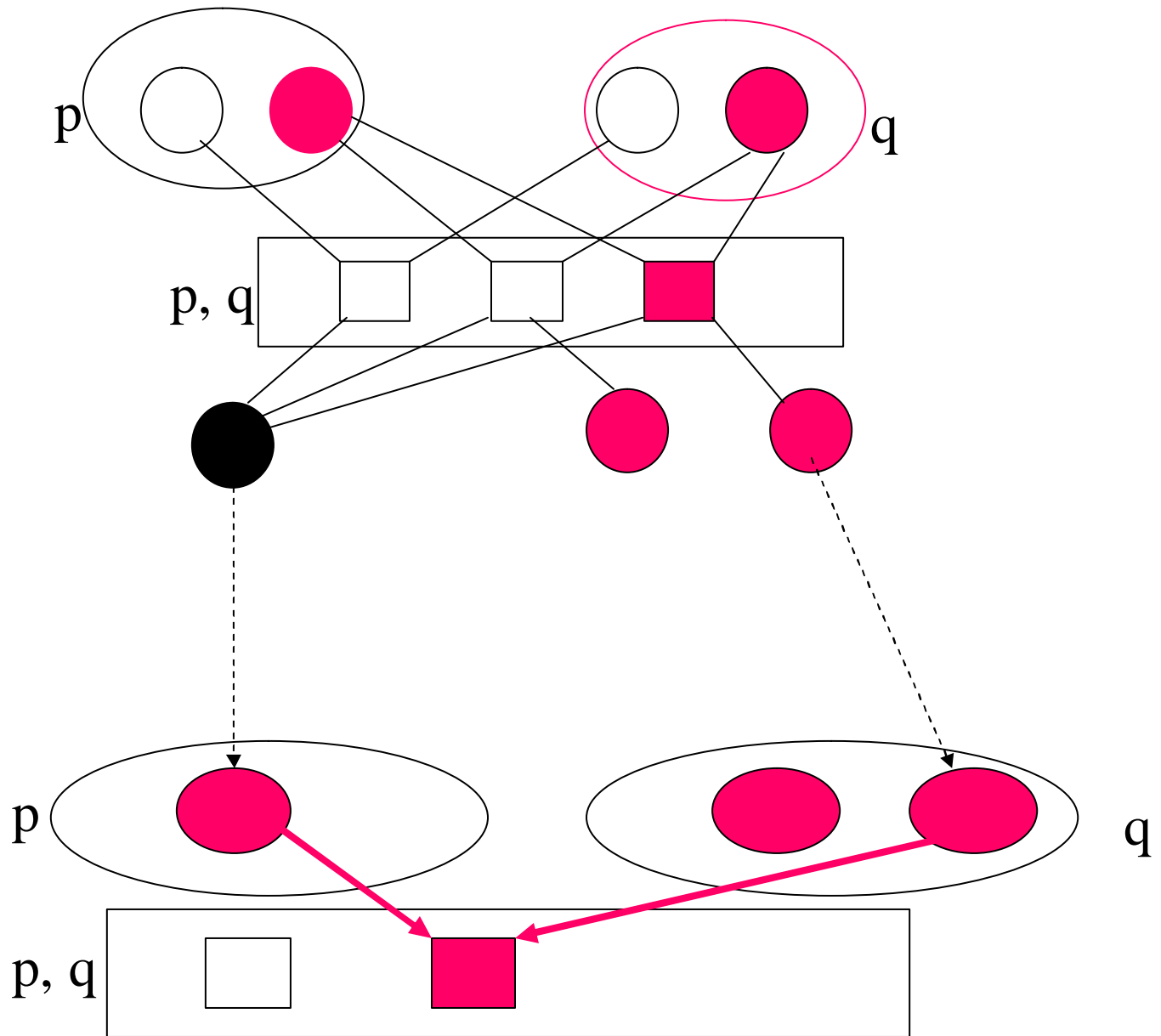




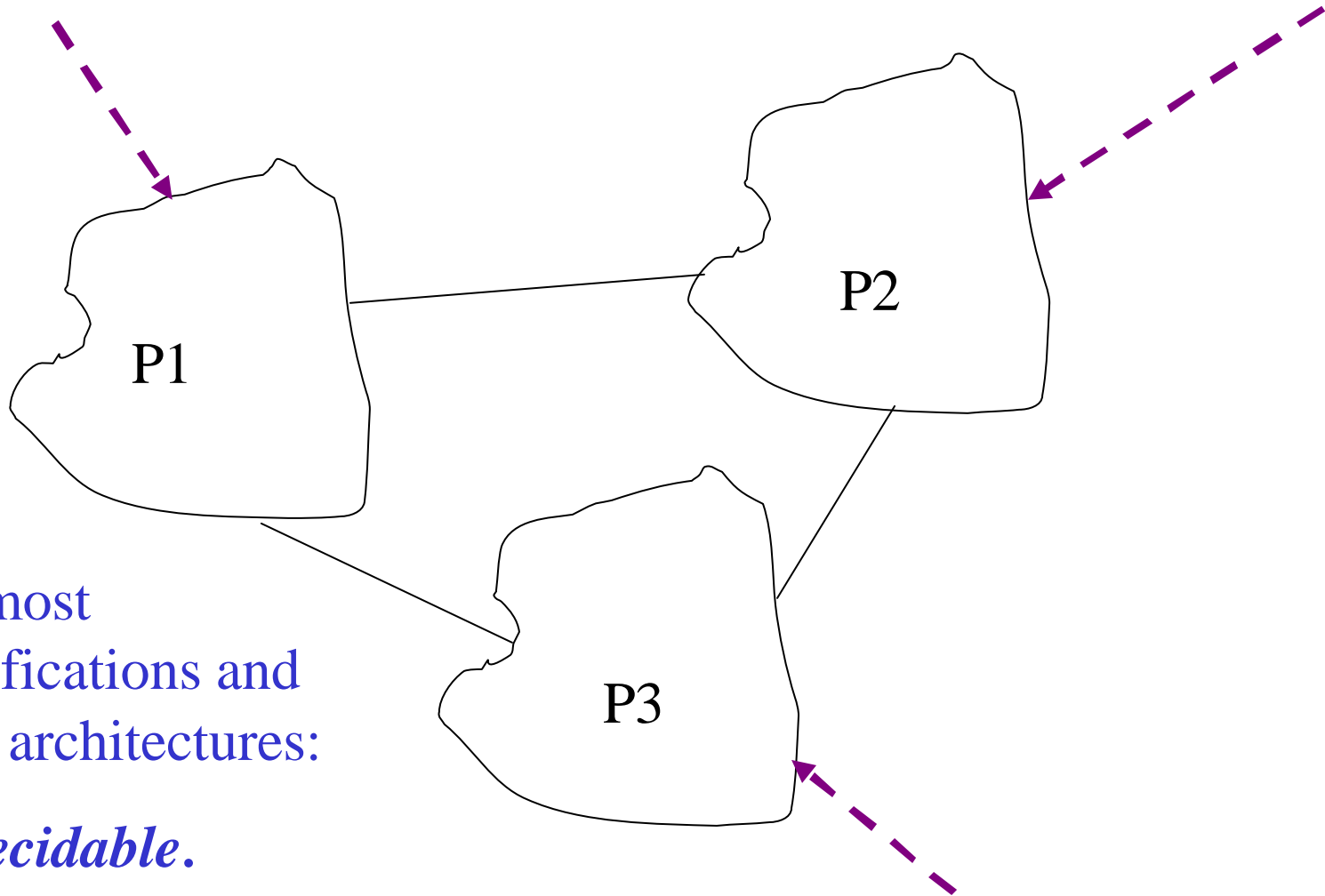
# Local Strategies







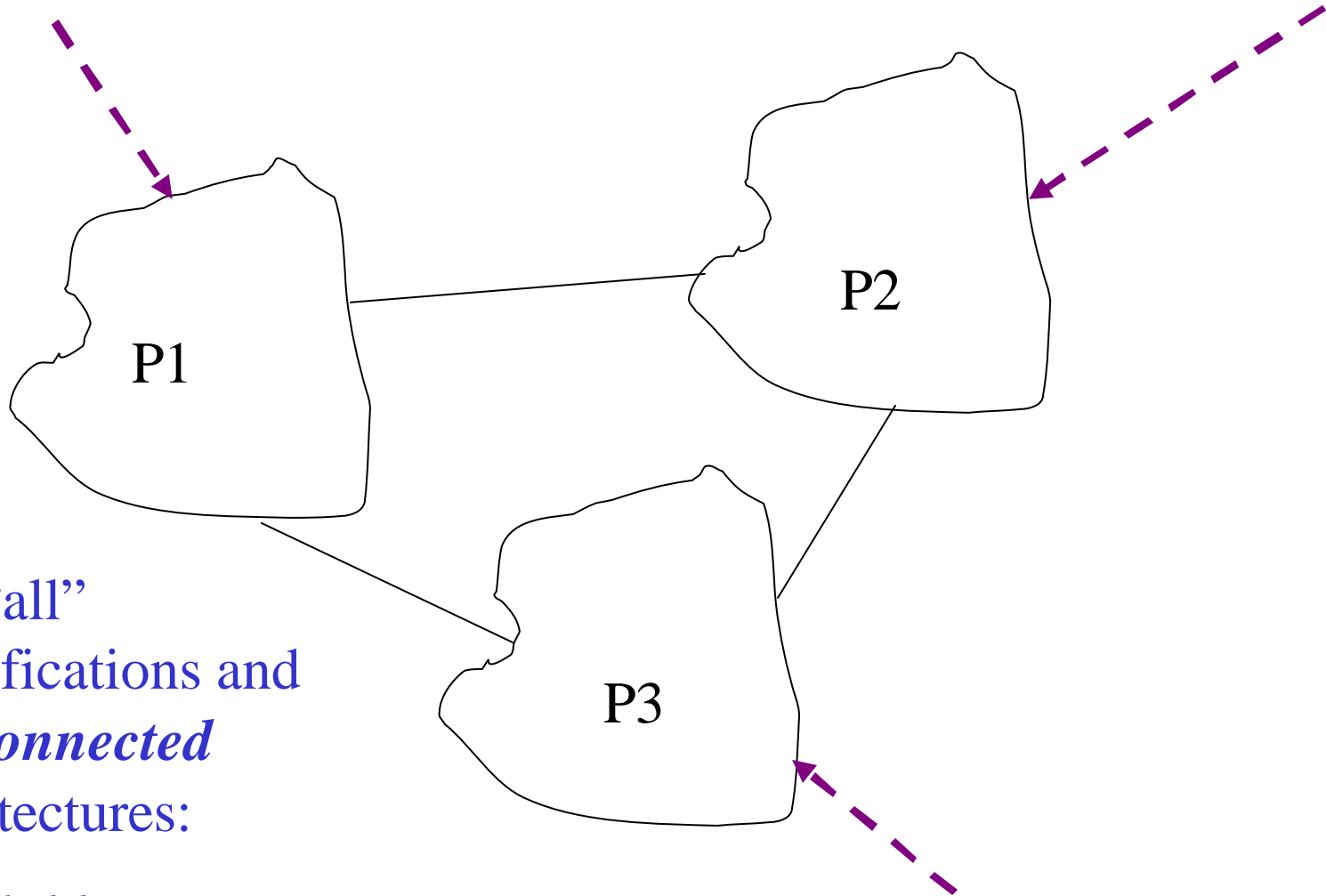
# General setting



For most specifications and most architectures:

*Undecidable.*

# The CCP setting



For “all”  
specifications and  
all *Connected*  
architectures:

*decidable.*

# Conclusion

- Cyclic Communicating Plants:
  - Restricted structured subclass of colored Petri nets.
  - Good balance of control flow and data abstractions.
  - Supports hierarchy, formal verification.
  - Admits a powerful theory:
    - Controller synthesis
- Applications?