



# Validating the Model

---

## Expressiveness 1

components of models, token flow etc.  
can easily be identified in process nets





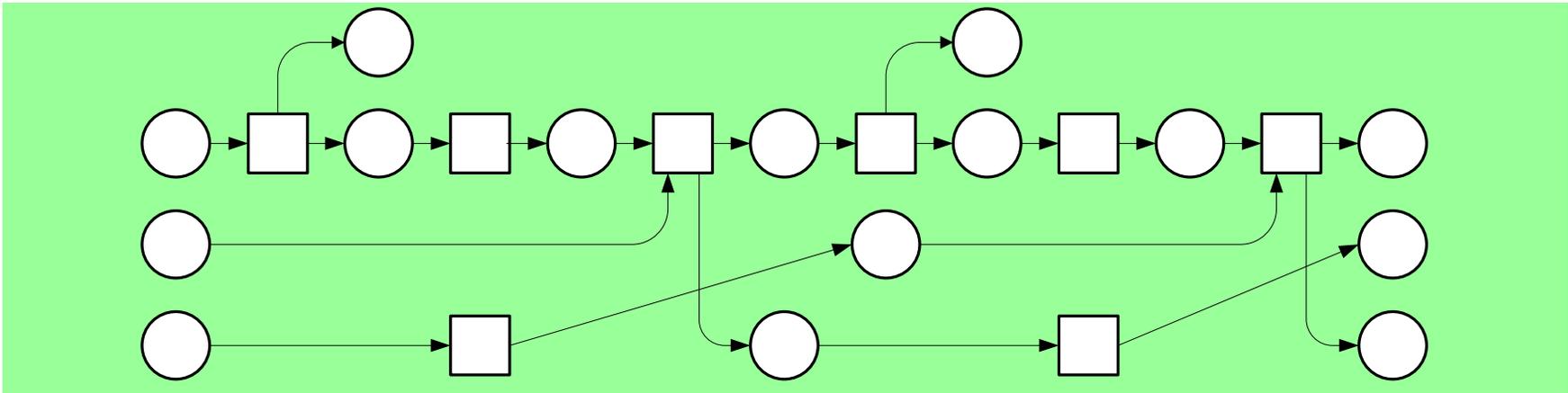
# Validating the Model

---

## Expressiveness 2

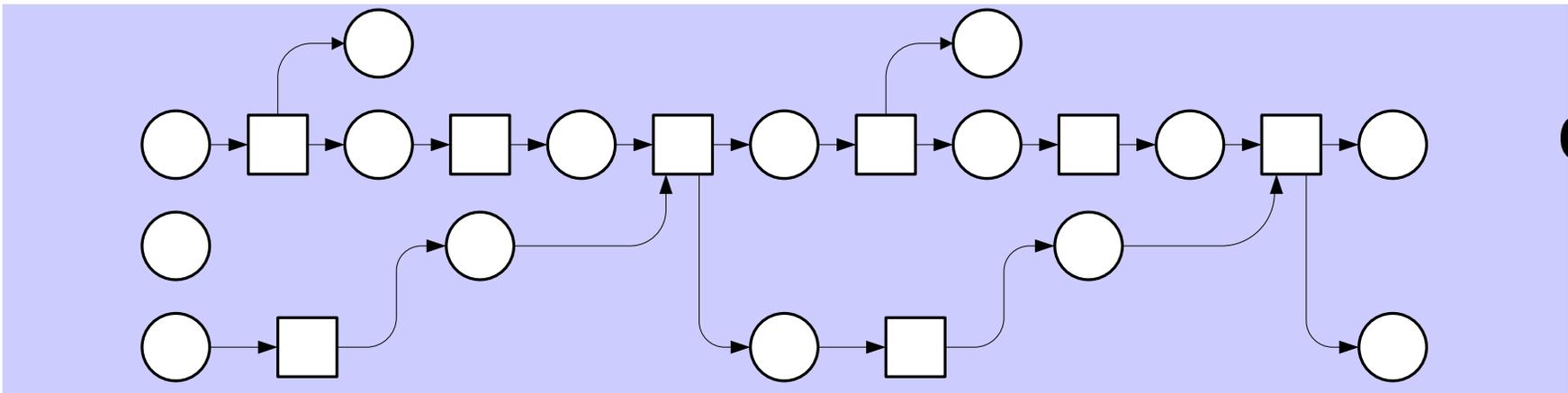
causal relationships are explicitly represented  
(which is not the case for occurrence sequences  
of non-safe place/transition nets)

# Advantage 2: Expressiveness



a common occurrence sequence of both process nets:

**insert, accept, brew, dispense, brew, insert, accept, dispense**



ready

insert



# Validating Requirements

---

## What type of requirements?

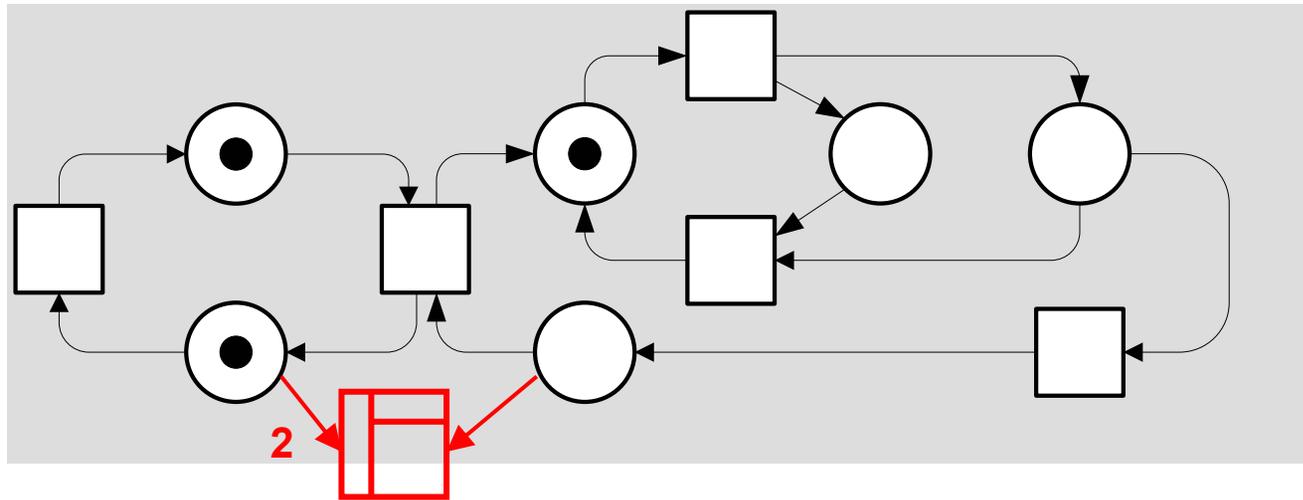
- linear-time properties that can be interpreted on single process nets

## Two examples:

- facts representing invariant properties
- goals representing “leads-to”-properties



# A Fact

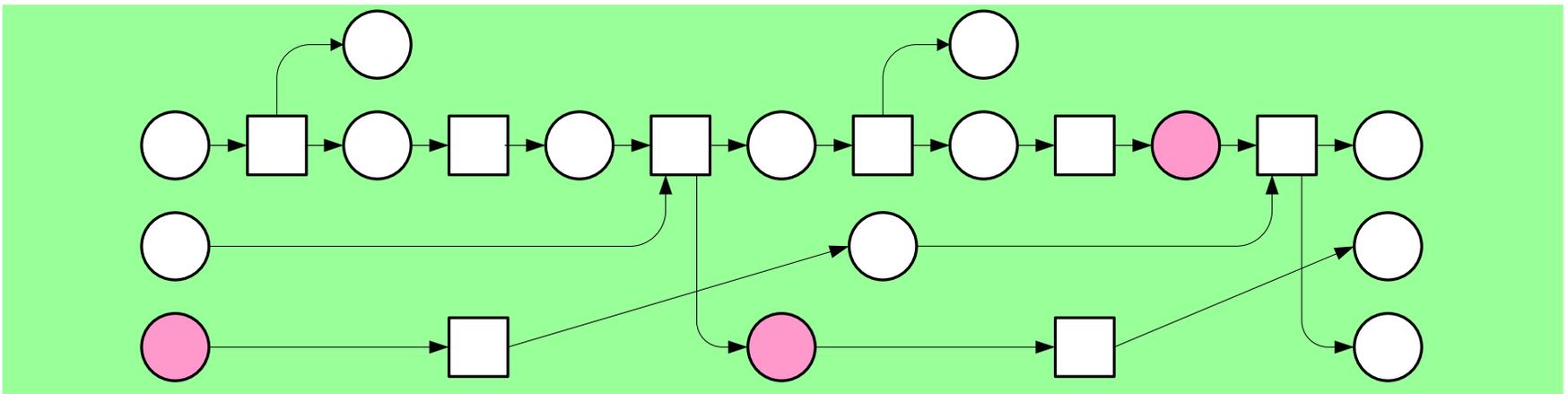
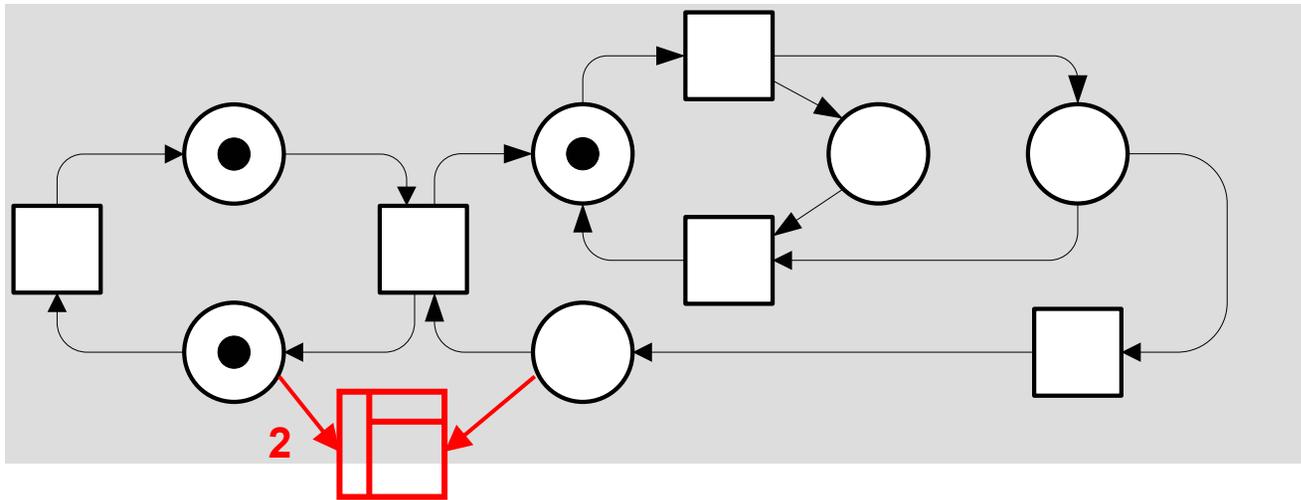


identify processes with a reachable marking  
marking **cold** twice and marking **accepted**

wa



# A Fact

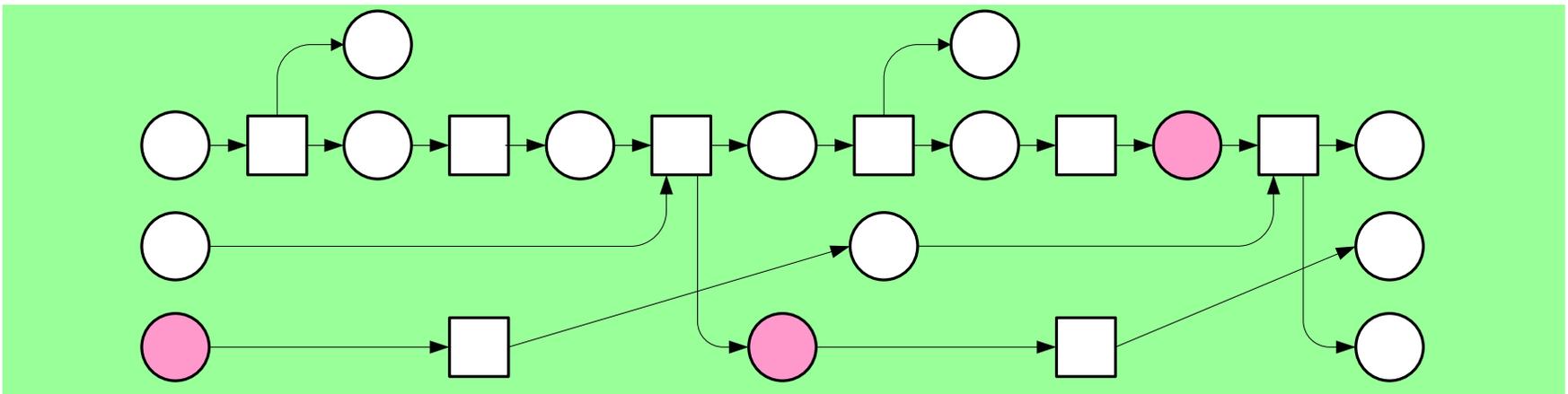


wa



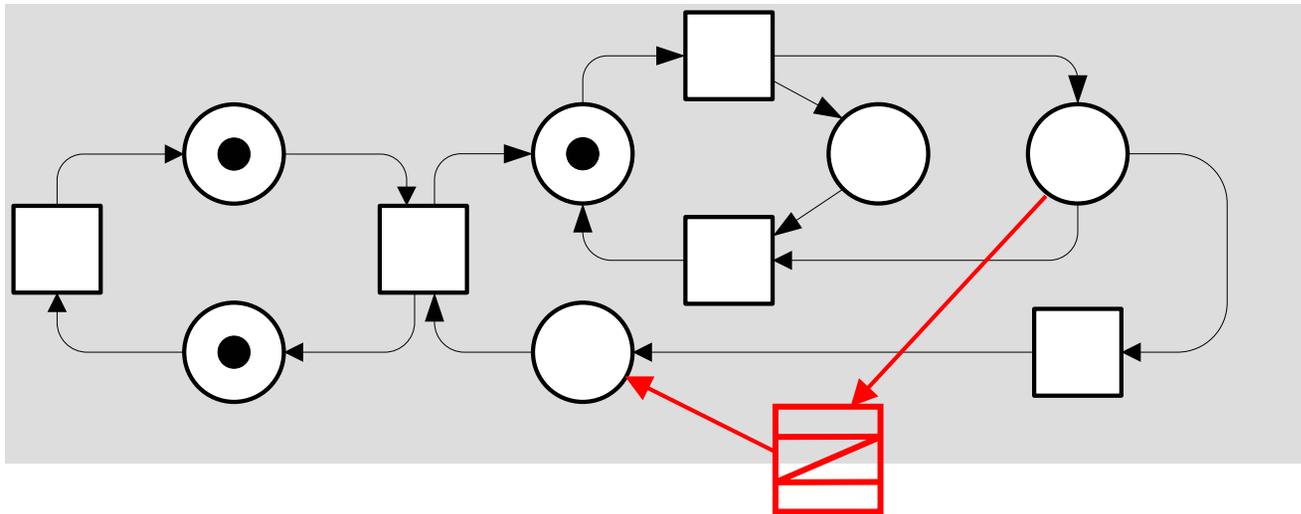
# A Fact

- A co-set of appropriately marked places of the process...
- ... is included in a cut
- ... which corresponds to a reachable marking of the process
- ... which corresponds to a reachable marking of the net.





# A Goal



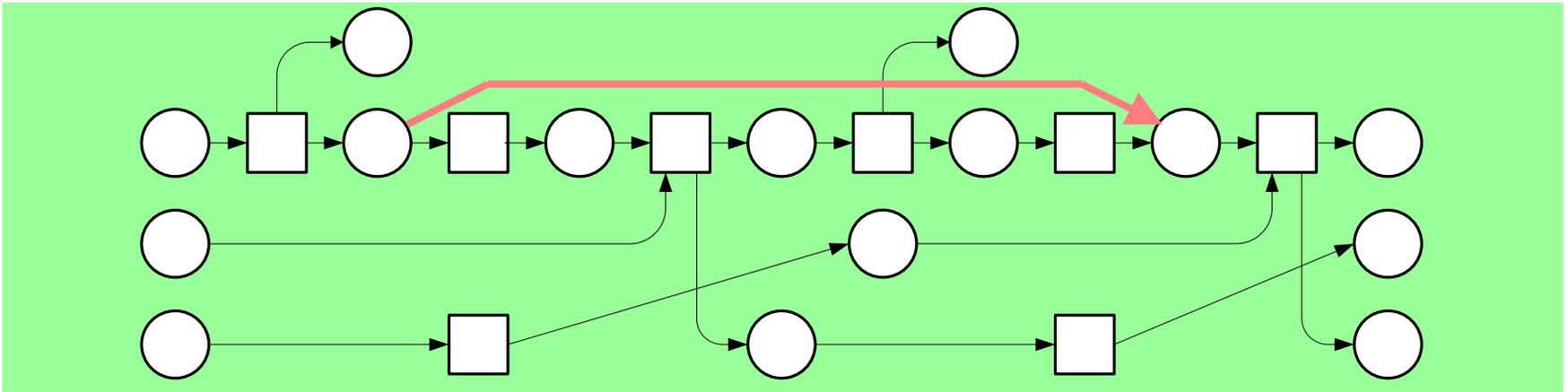
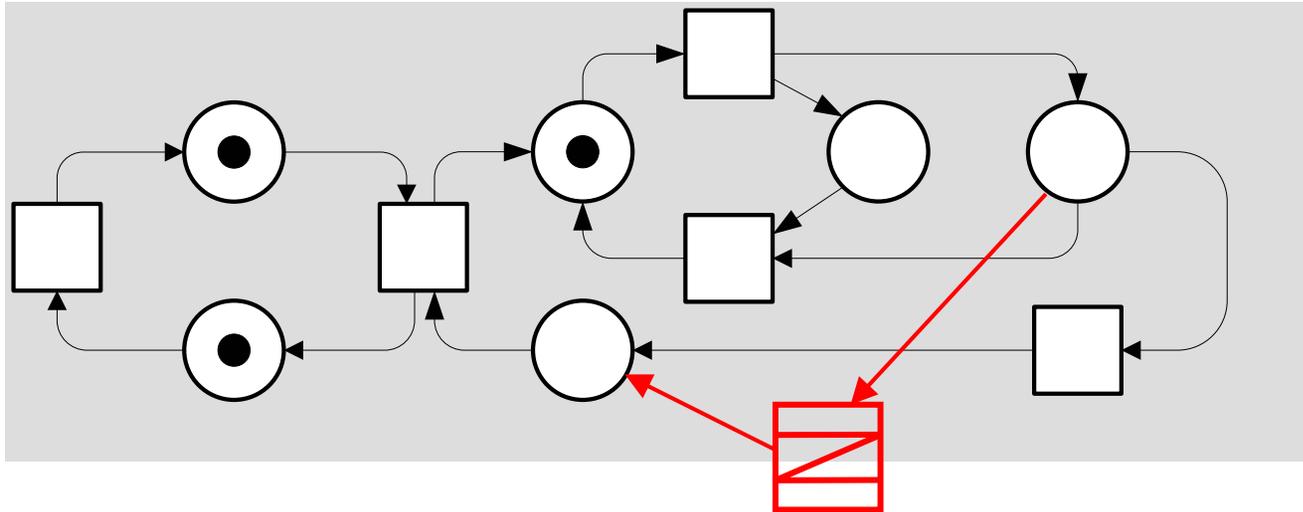
identify processes such that:

If there is a token on *inserted* then eventually there will be a token on *accepted*

wa



# A Goal



wa

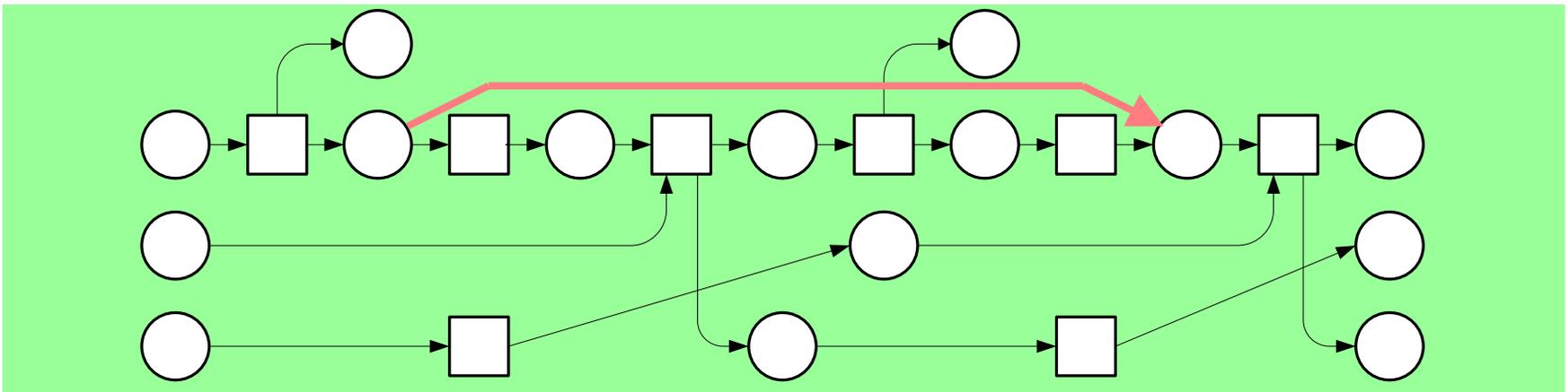


# A Goal

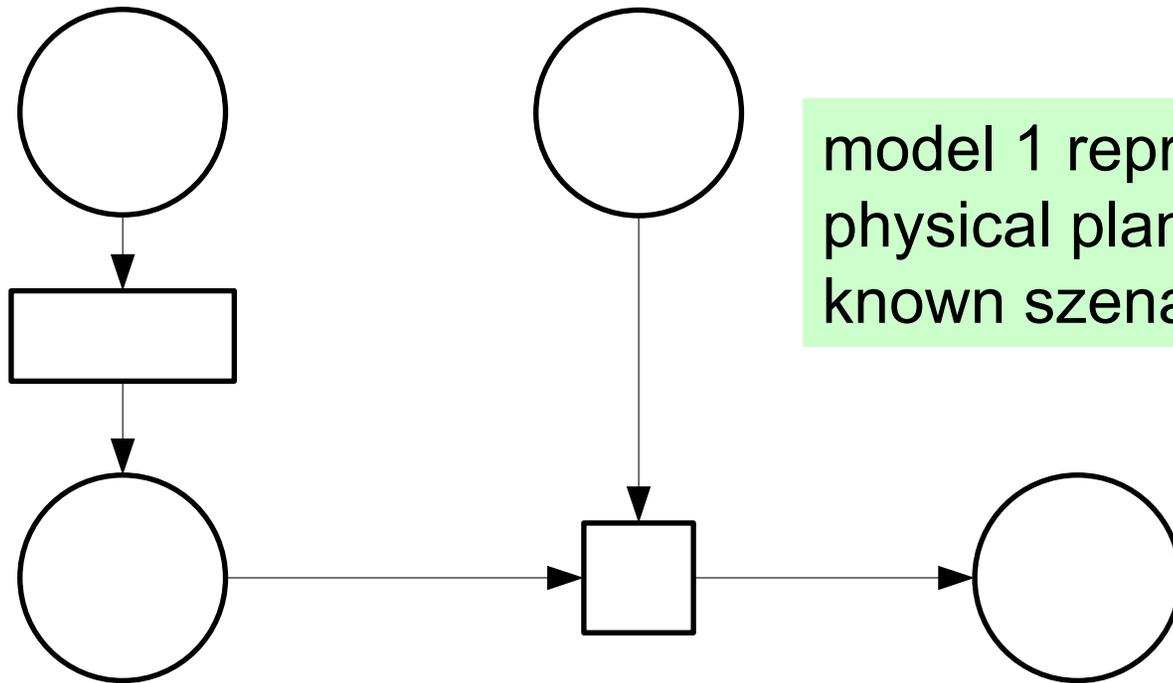
***accepted*** causally depends on ***inserted***

... hence from each marking of the process  
that marks the condition ***inserted***  
a marking will be reached that marks condition ***accepted***.

So for this run, the place  
***accepted*** will eventually be marked after ***inserted***.



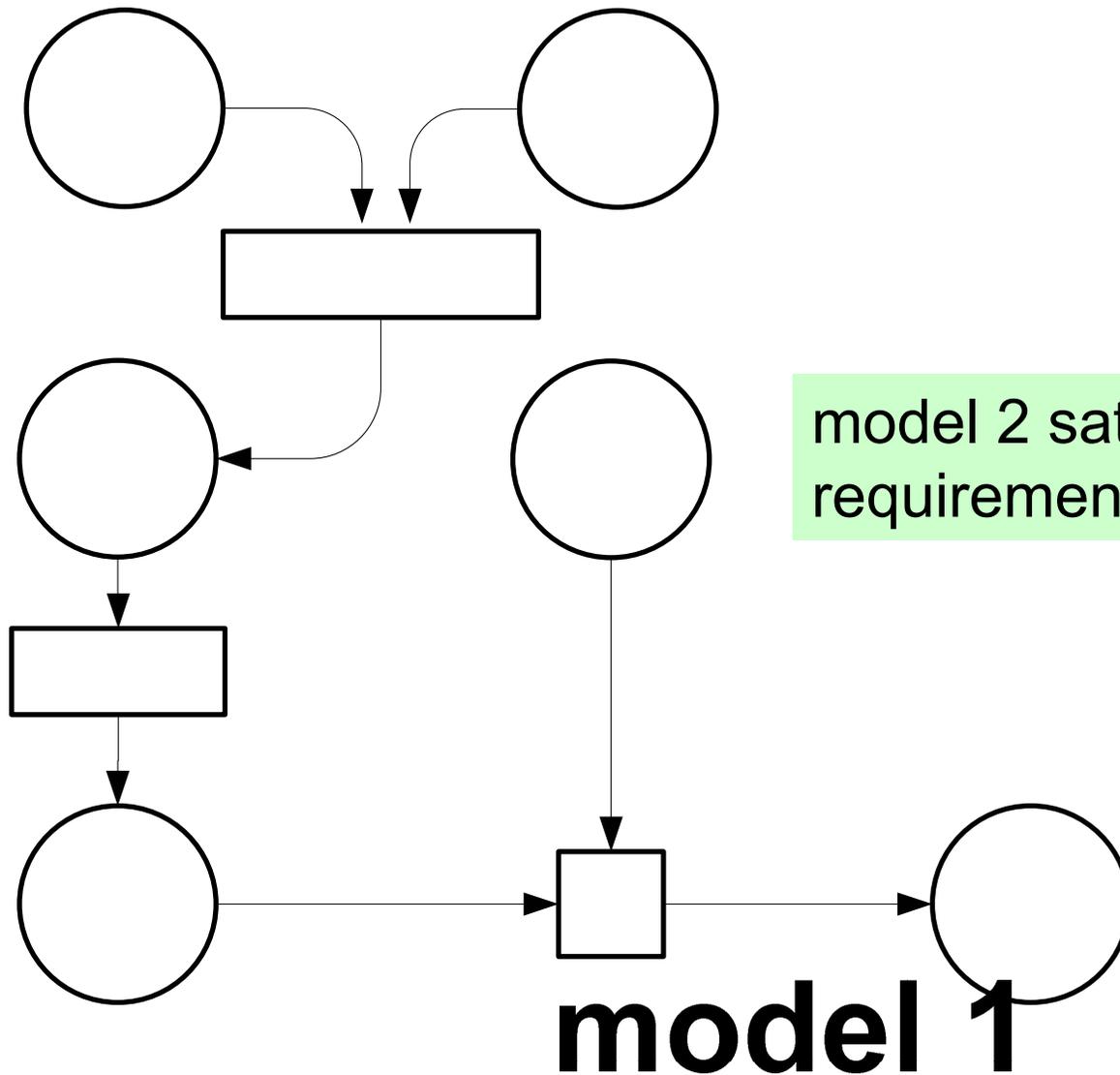
# Stepwise Validation of Requirements



model 1 represents the physical plant, known szenarios etc.

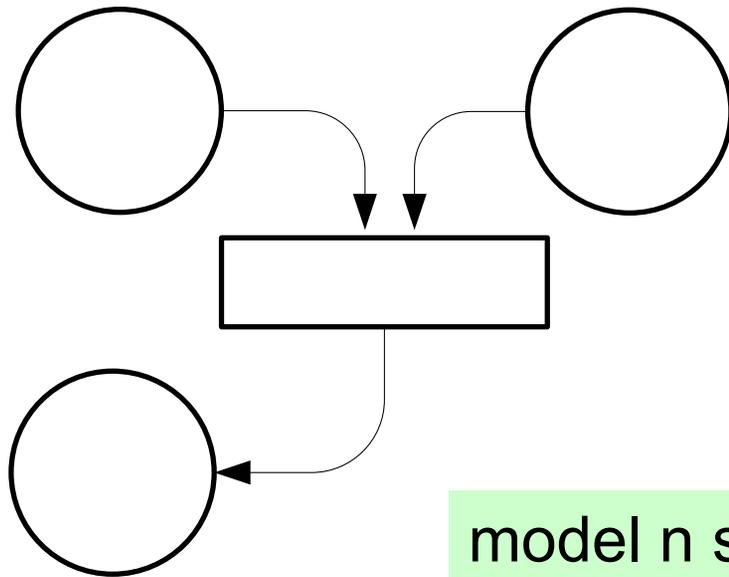
# model 1

# Stepwise Validation of Requirements



model 2 satisfies requirement 1  
requirement 2 is validated

# Stepwise Validation of Requirements



model n satisfies requirements 1, ... , n-1  
final model satisfies requirements 1, ... , n

**This approach only works in general  
if all requirements restrict behavior**

# model n



# An Industrial Case Study

---



Conducted together with engineers from Audi

**Goal:** modeling controlled car features + control

**Here:** control of fuel gauge

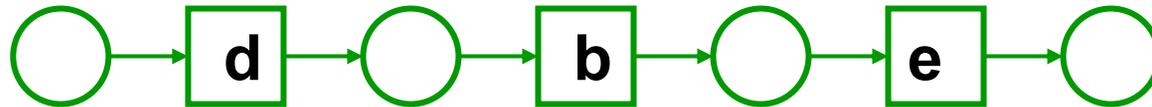
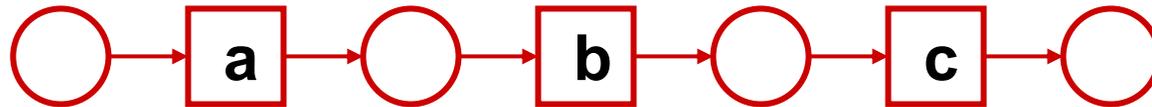
**Given:** not plant + control (hardware + algorithm)

but scenarios, which are translated into runs

# Synthesis of process models



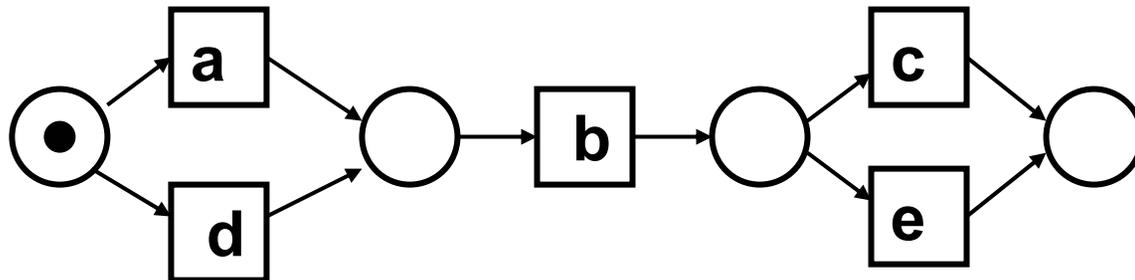
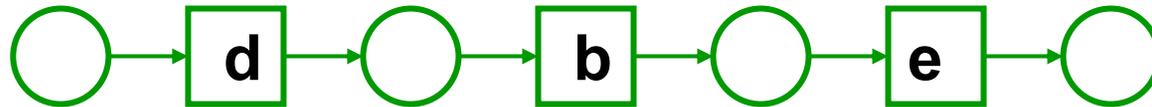
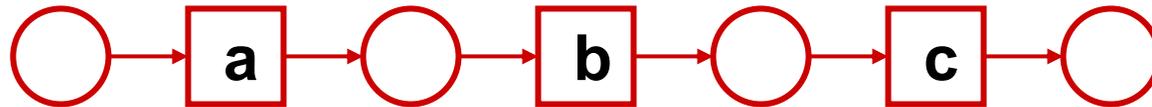
## Folding of all process nets



# Synthesis of process models



## Folding of all process nets

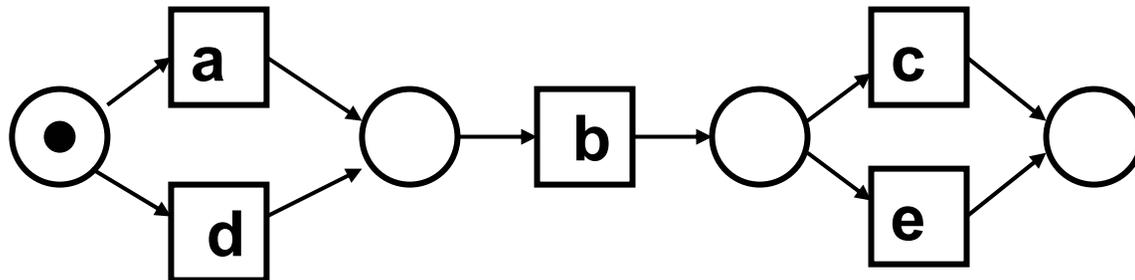
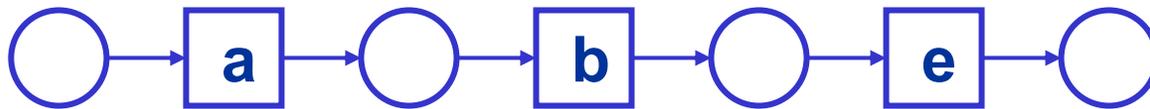


# Synthesis of process models



## Folding of all process nets

Problem: additional runs, e.g.



# Synthesis of process models



## Folding of all process nets

Solution: more precise models

