Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni

http://www.di.unipi.it/~bruni

18 - Workflow modules

Object

We study Workflow modules to model interaction between workflows

Ch.6 of Business Process Management: Concepts, Languages, Architectures

Problem

Not all tasks of a workflow net are automatic:

they can be triggered manually or by a message

they can be used to trigger other tasks

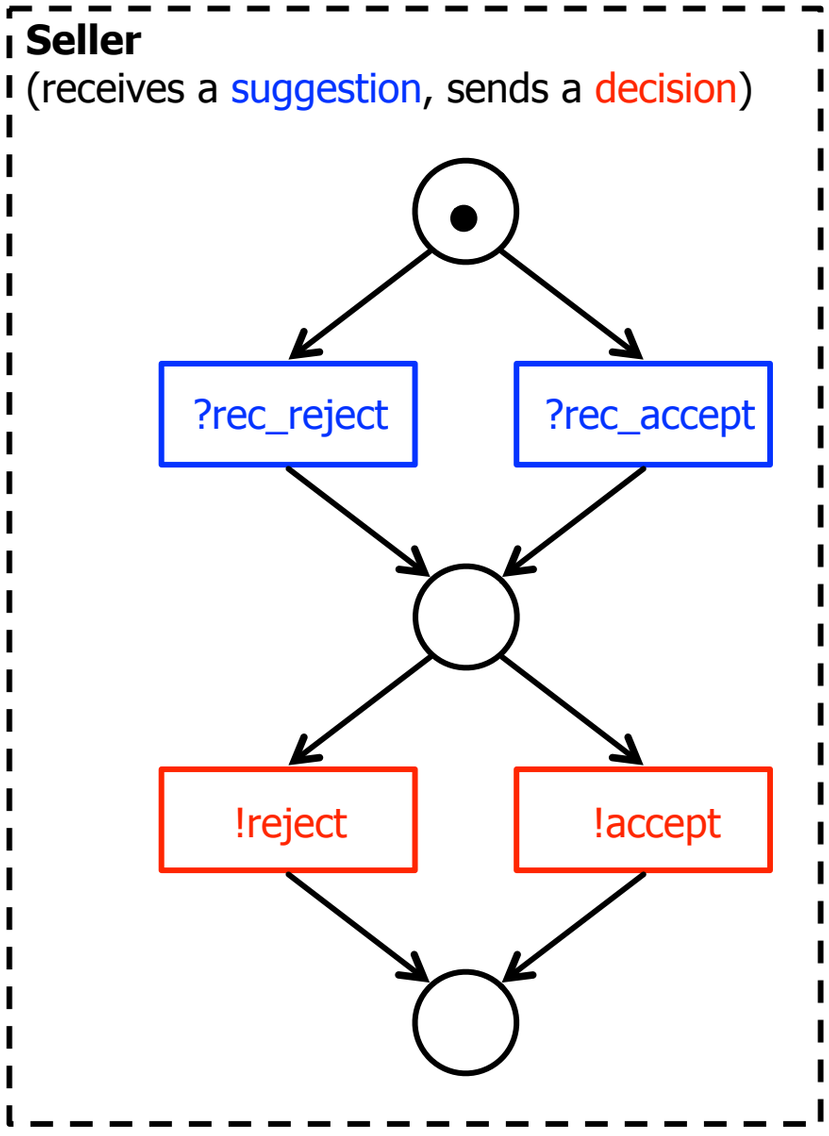
How do we represent this?

Implicit interaction

Separately developed processes

Some activities can input messages (symbol ?)

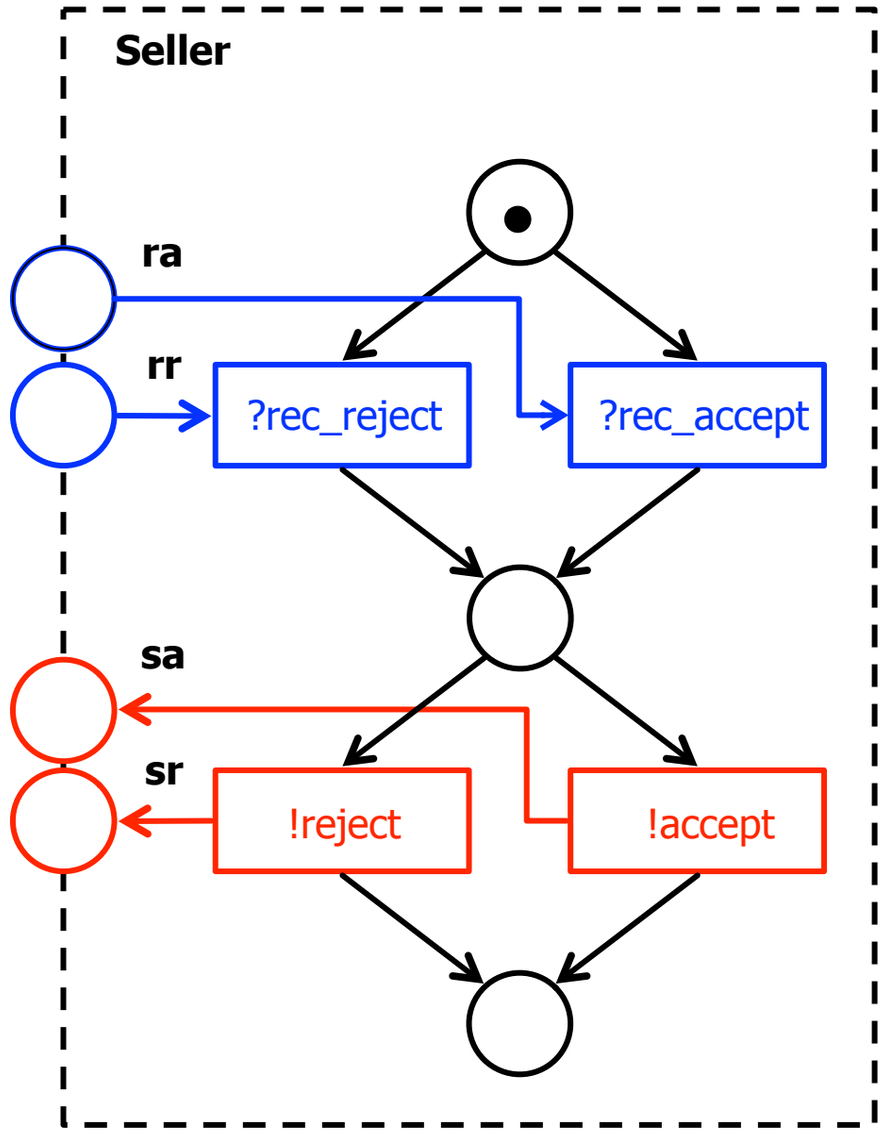
Some activities can output messages (symbol !)



Interface

Seller has an interface for interaction

It consists of some input places and some output places



From Workflow nets to Workflow modules

Assume the original workflow net has been validated:

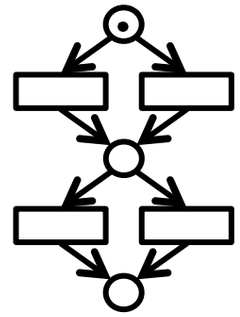
it is a sound (and maybe safe) workflow net

When we add the (places in the) interface
it is no longer a workflow net!
It becomes a **workflow module**

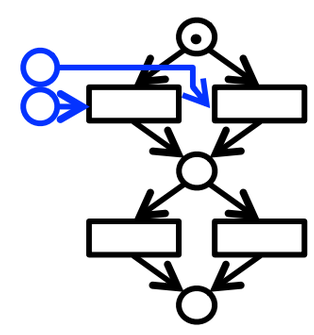
Workflow Modules

Definition: A **workflow module** consists of

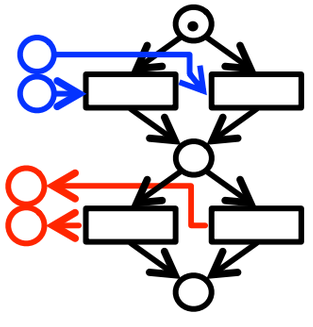
a (sound) workflow net (P,T,F)



plus a set P^I of incoming places
plus a set of incoming arcs $F^I \subseteq (P^I \times T)$

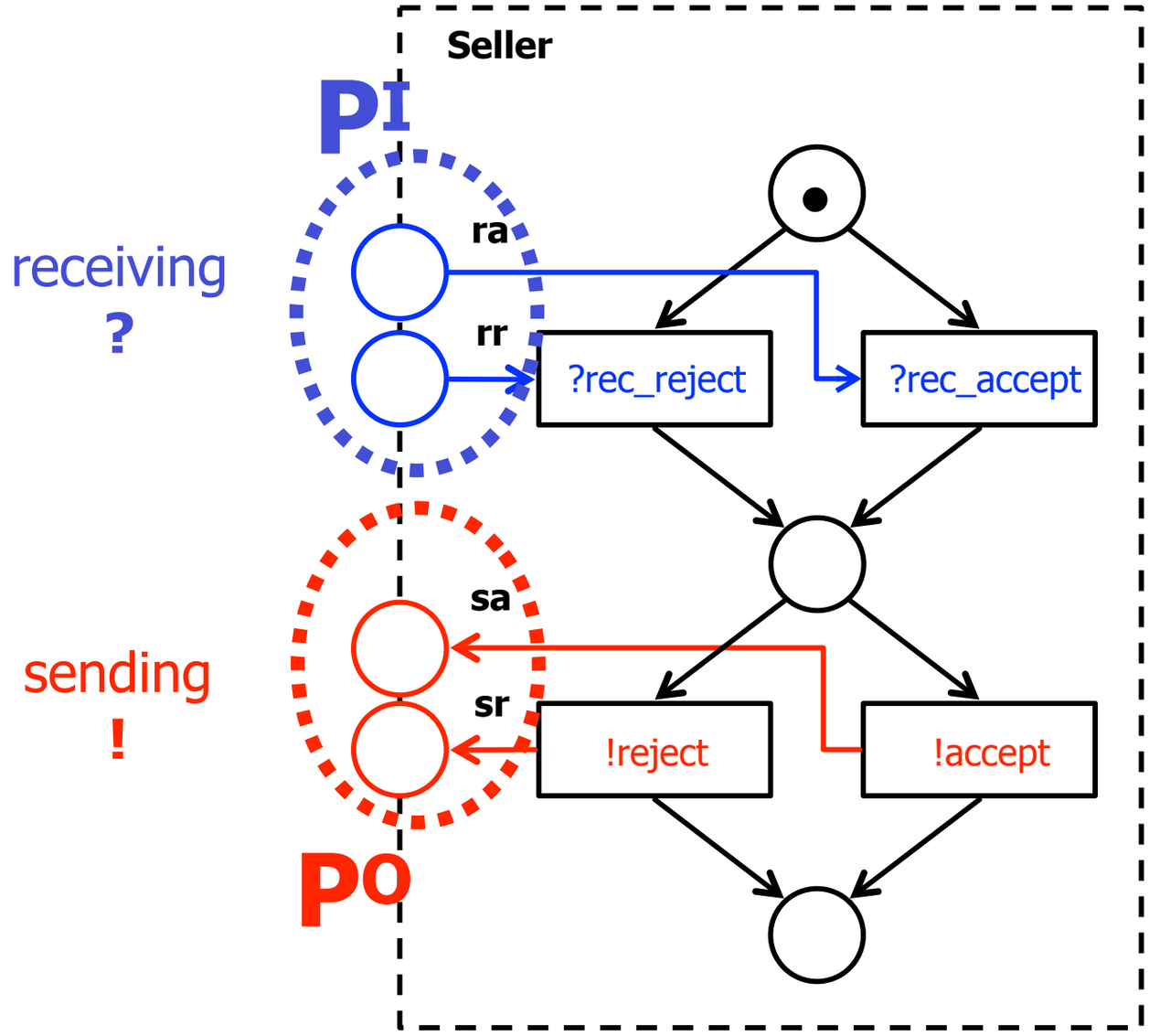


plus a set P^O of outgoing places
plus a set of outgoing arcs $F^O \subseteq (T \times P^O)$



such that each transition in T has
at most one arc to places in the interface $P^I \cup P^O$

Workflow module: example

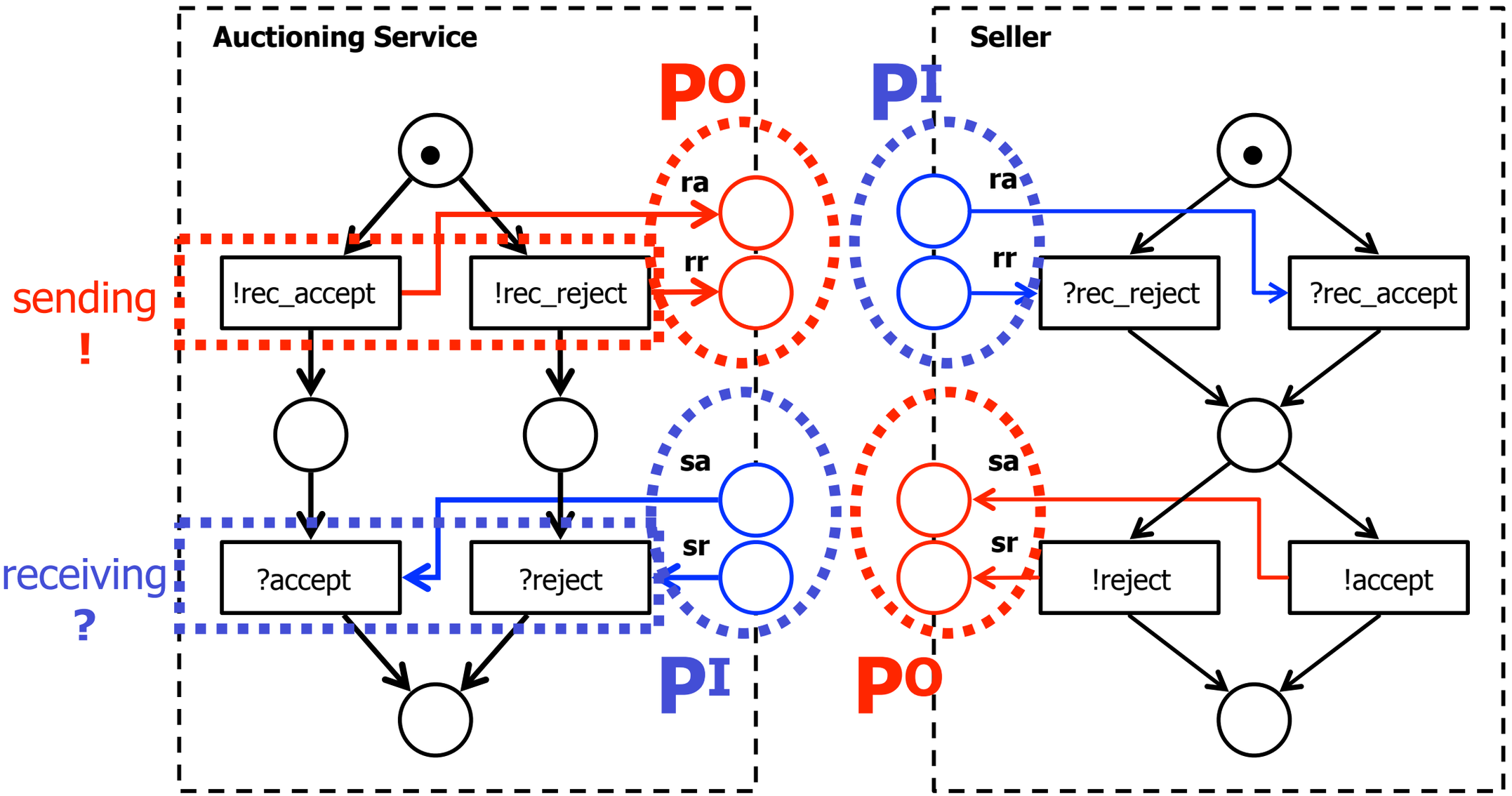


Structural compatibility

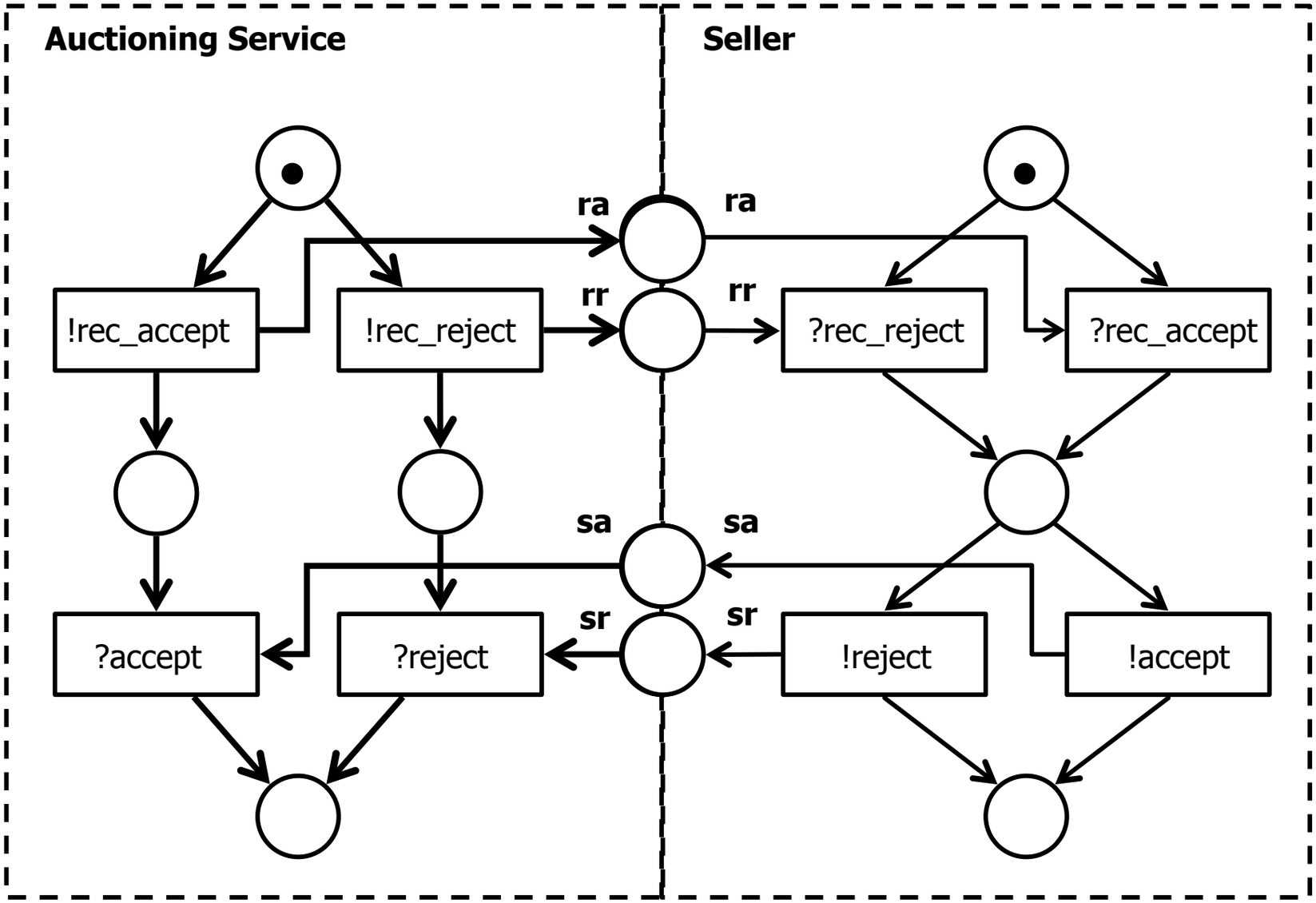
A set of workflow modules is called
structurally compatible
if
for every message that can be sent
there is exactly a module who can receive it,
and
for every message that can be received
there is exactly a module who can send it

(formats of message data are assumed to match)

Compatibility



Interaction



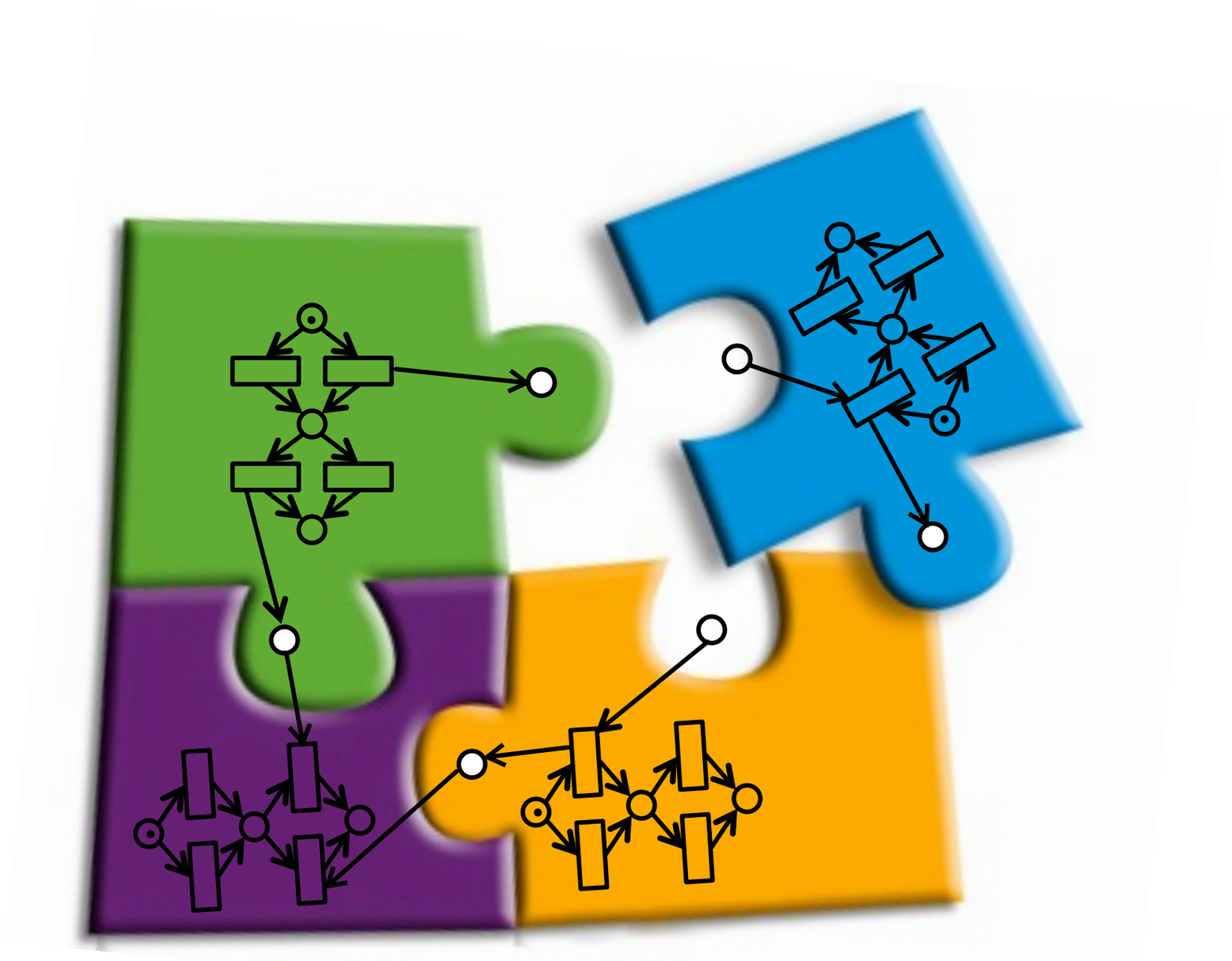
Problem

We have added places and arcs to single wf nets
We have joined places of different wf modules

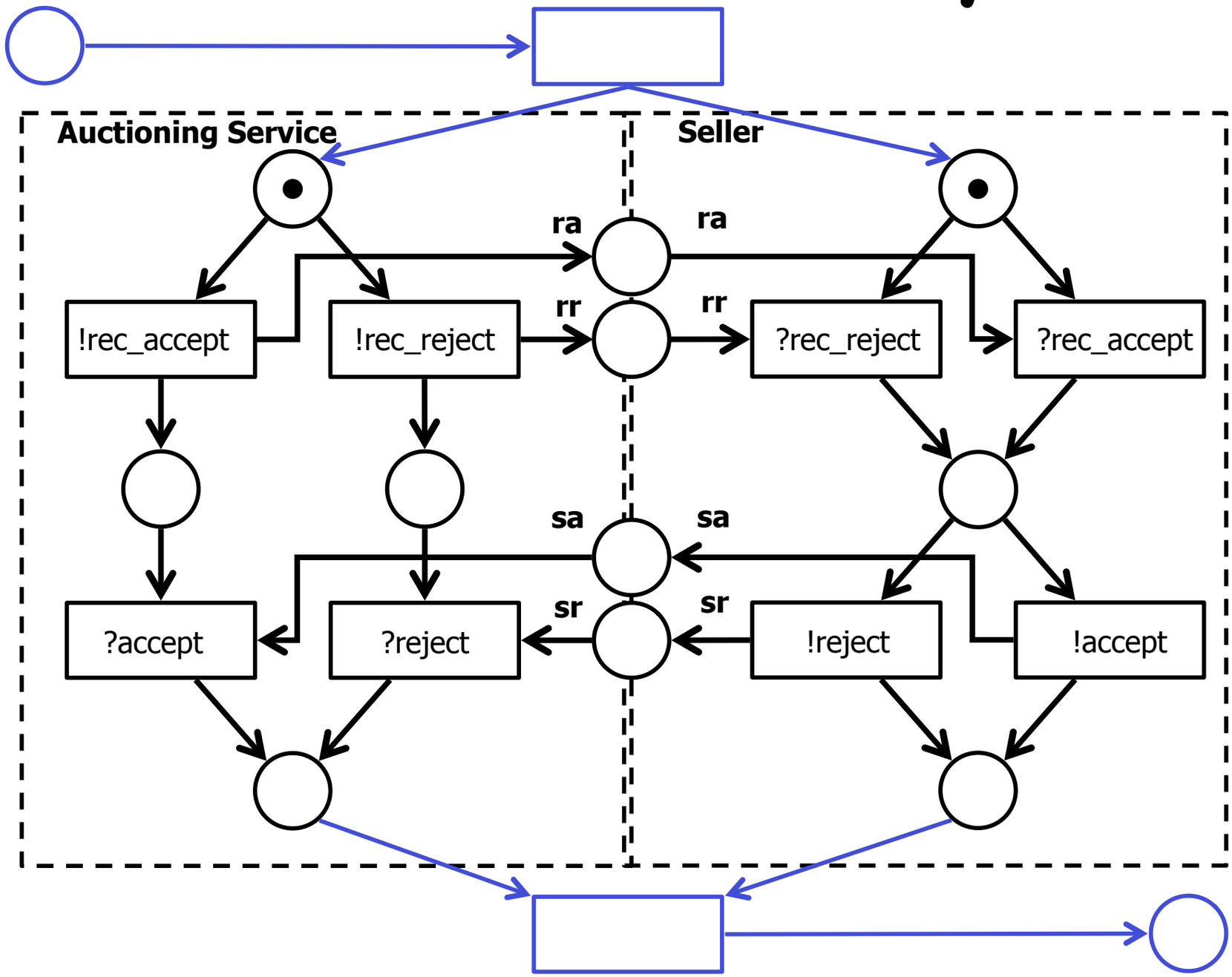
How do we check that the system behaves well?

What has this check to do with WF net soundness?

Workflow systems



Workflow system



Workflow system

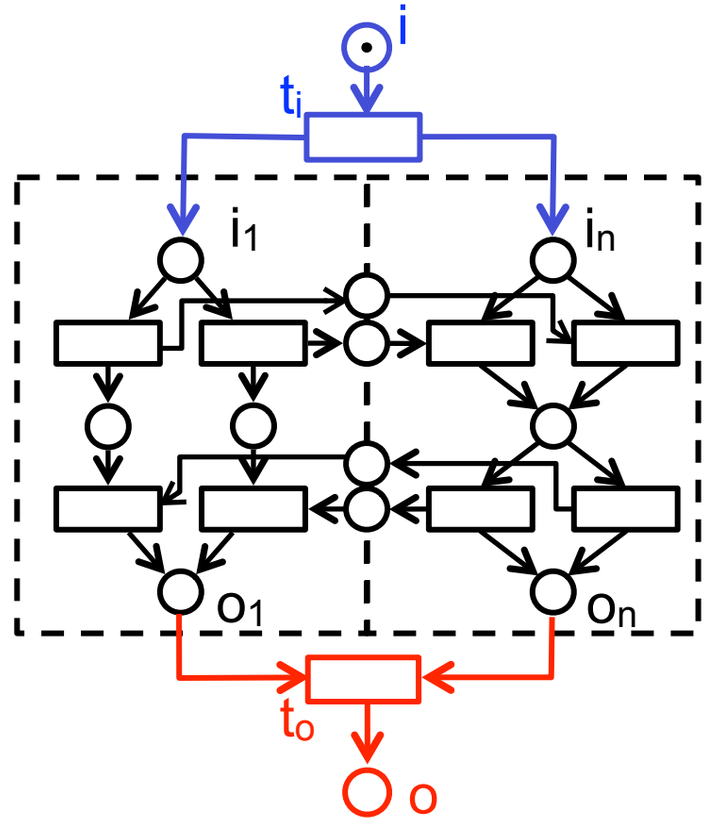
Definition: A **workflow system** is a wf net that consists of

a set of n structurally compatible wf modules
(initial places $i_1,\ldots,i_n$, final places $o_1,\ldots,o_n$)

plus an initial place **i**
and a transition $\mathbf{t_i}$ from **i** to $i_1,\ldots,i_n$

plus a final place **o**
and a transition $\mathbf{t_o}$ from $o_1,\ldots,o_n$ to **o**

whose initial marking is **i**



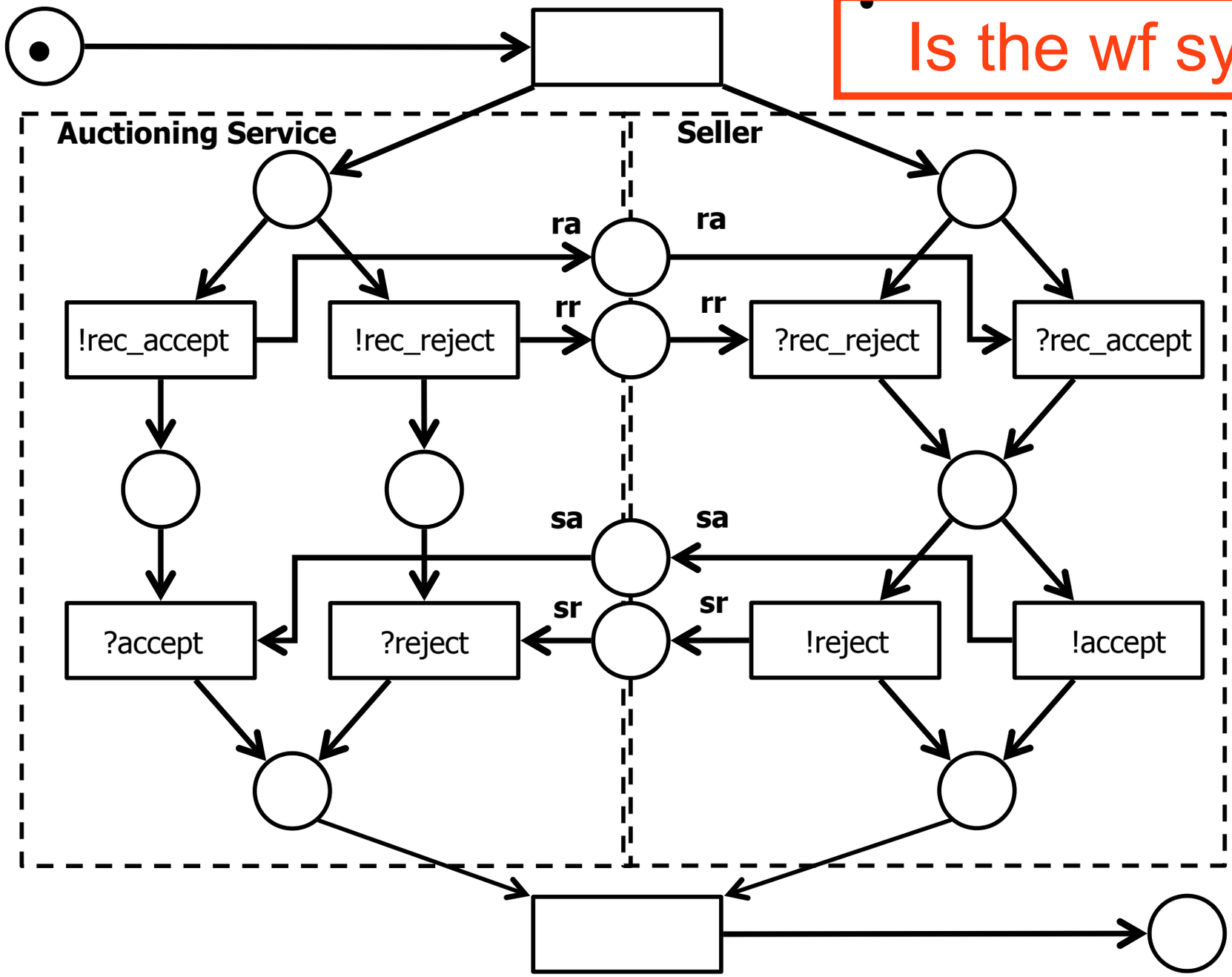
Soundness of workflow systems

A workflow system is just an ordinary workflow net

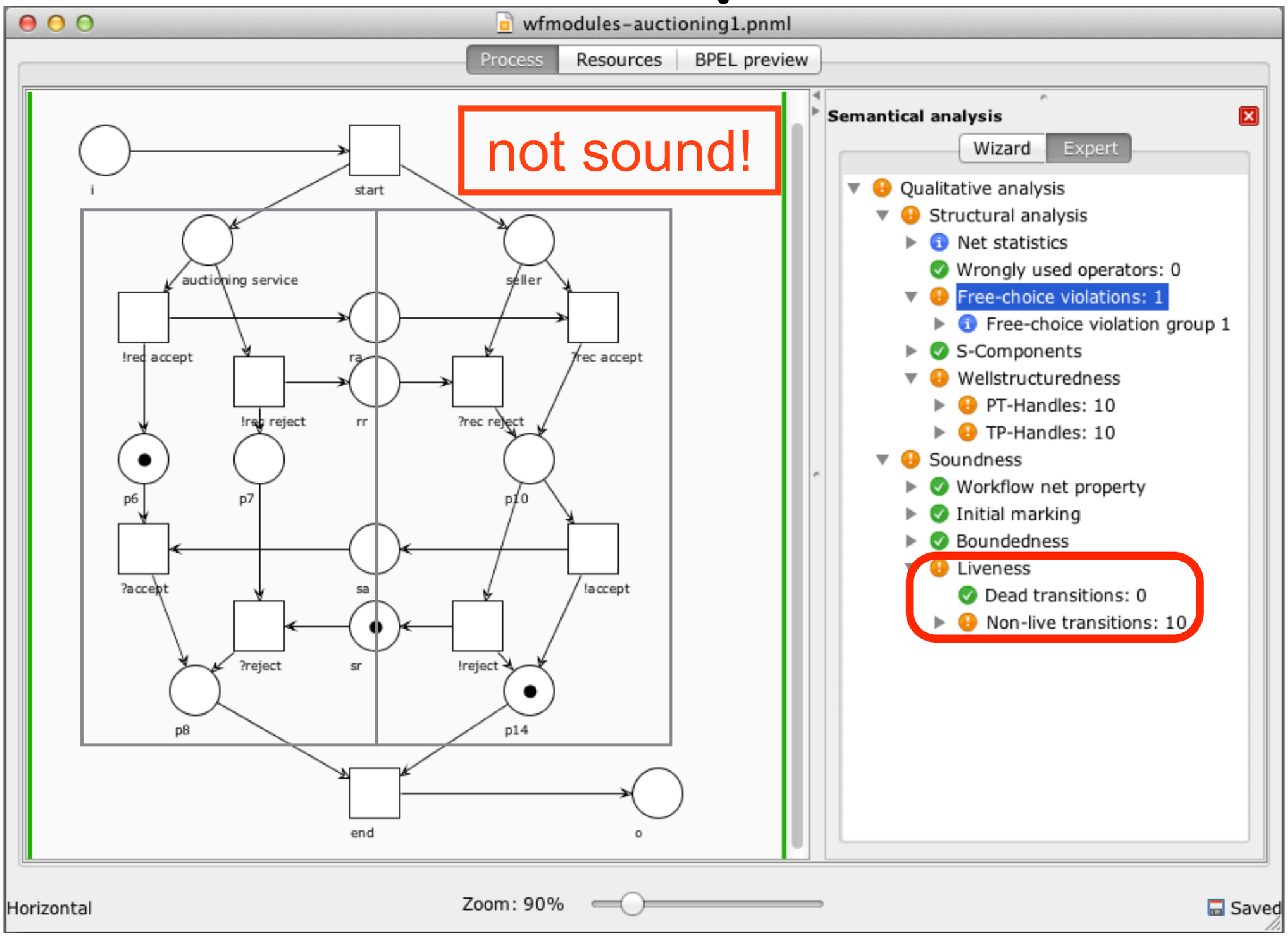
We can check its **soundness** as usual

Example

Is the wf system sound?

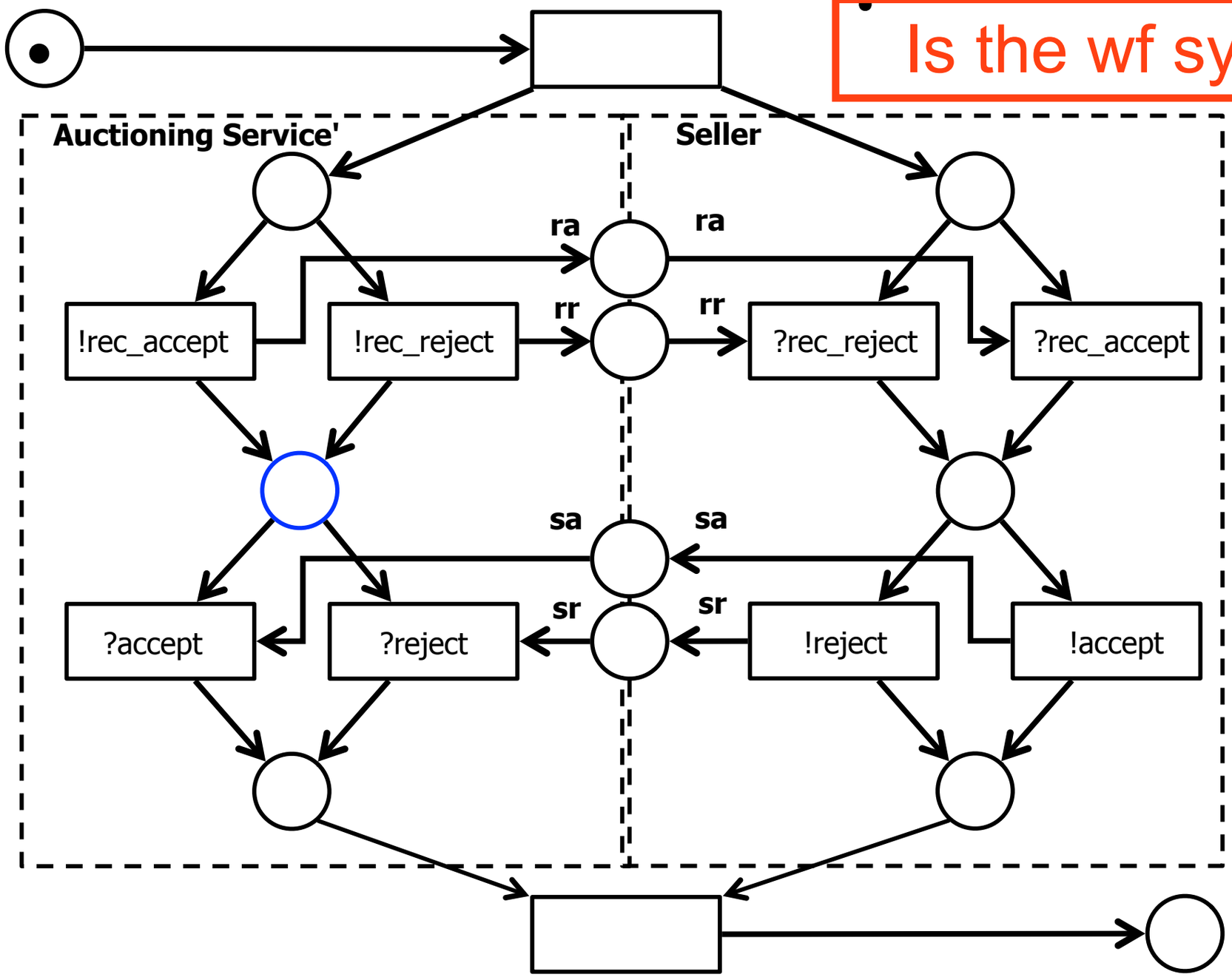


Example

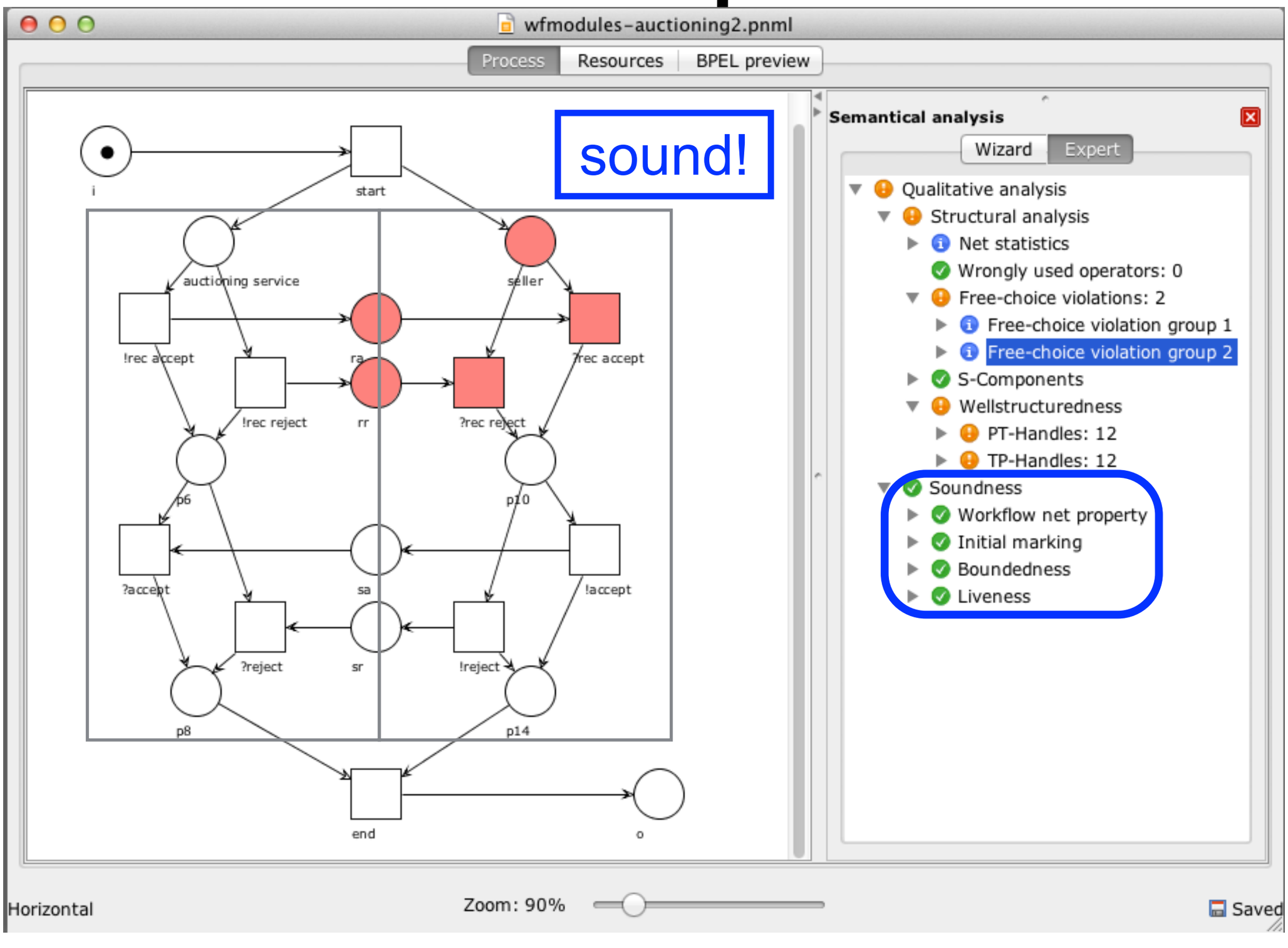

wfmodules-auctioning1.pnml
Process
Resources
BPEL preview
not sound!
i
start
auctioning service
seller
!rec accept
ra
?rec accept
!rec reject
rr
?rec reject
p6
p7
p10
?accept
sa
!accept
?reject
sr
!reject
p8
p14
end
o
Semantical analysis
Wizard
Expert
Qualitative analysis
Structural analysis
Net statistics
Wrongly used operators: 0
Free-choice violations: 1
Free-choice violation group 1
S-Components
Wellstructuredness
PT-Handles: 10
TP-Handles: 10
Soundness
Workflow net property
Initial marking
Boundedness
Liveness
Dead transitions: 0
Non-live transitions: 10
Horizontal
Zoom: 90%
Saved

Example

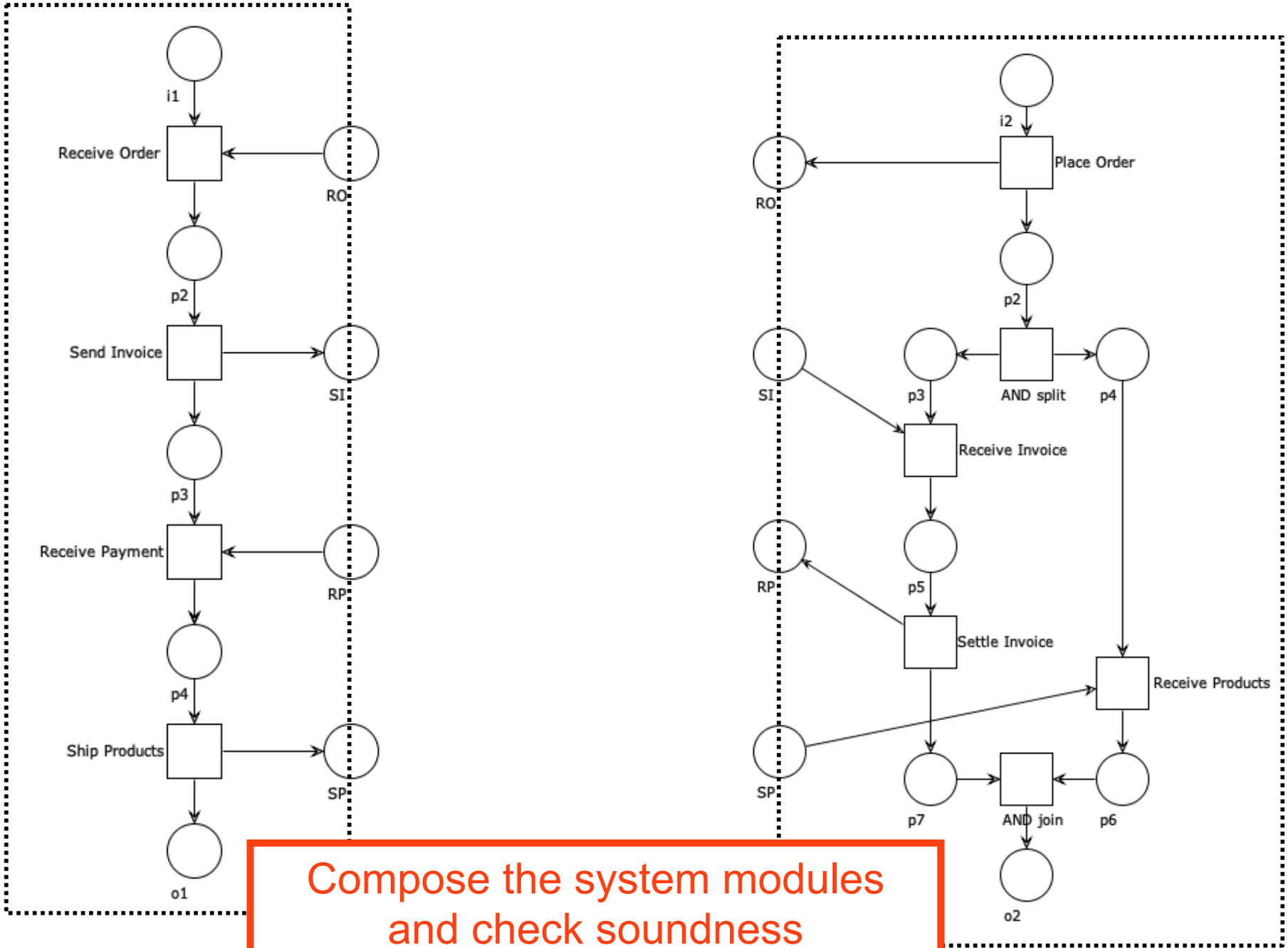
Is the wf system sound?



Example

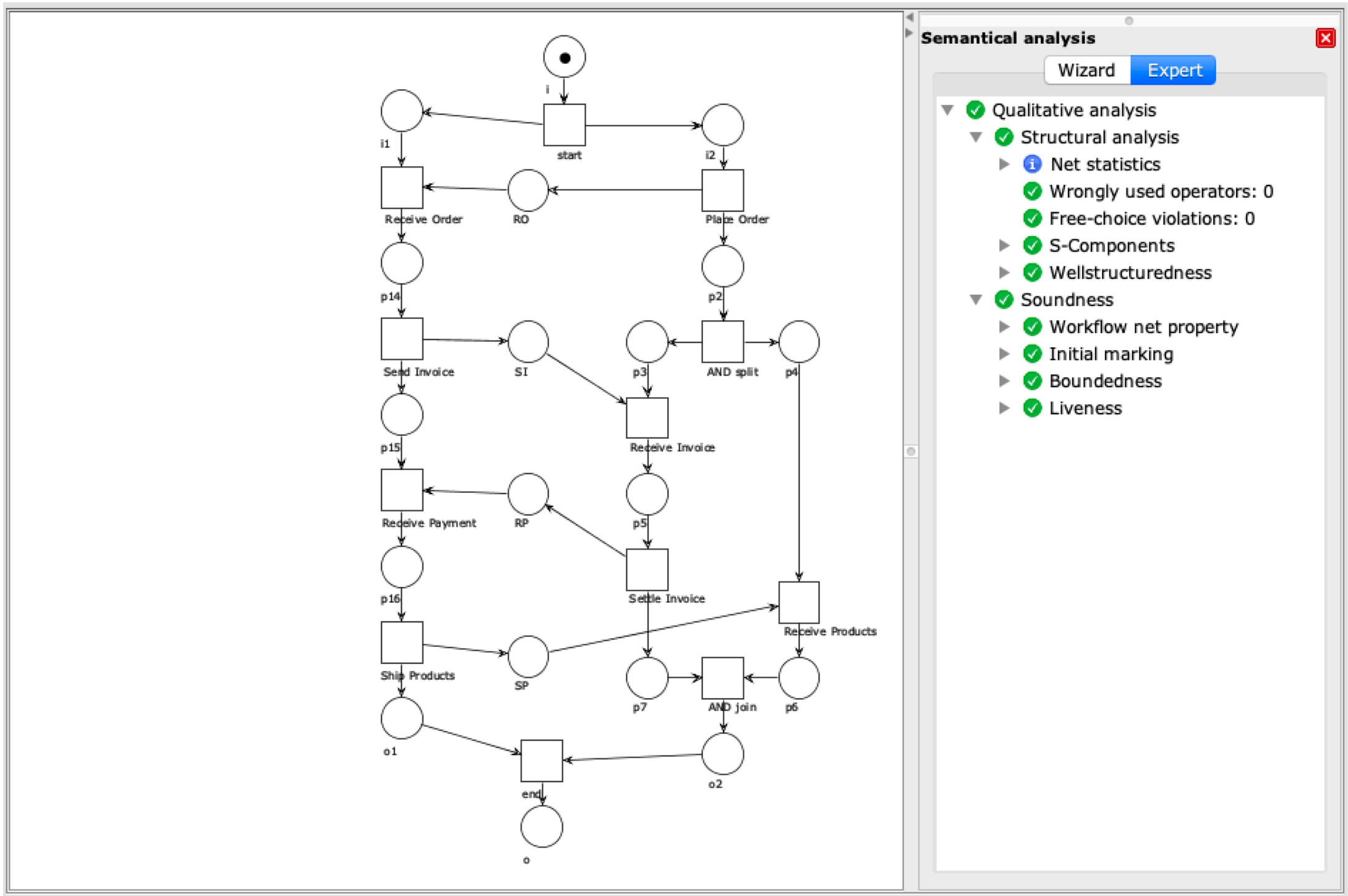

wfmodules-auctioning2.pnml
Process
Resources
BPEL preview
sound!
i
start
auctioning service
seller
!rec accept
ra
?rec accept
!rec reject
rr
?rec reject
p6
p10
?accept
sa
!accept
?reject
sr
!reject
p8
p14
end
o
Semantical analysis
Wizard
Expert
Qualitative analysis
Structural analysis
Net statistics
Wrongly used operators: 0
Free-choice violations: 2
Free-choice violation group 1
Free-choice violation group 2
S-Components
Wellstructuredness
PT-Handles: 12
TP-Handles: 12
Soundness
Workflow net property
Initial marking
Boundedness
Liveness
Horizontal
Zoom: 90%
Saved

Exercise

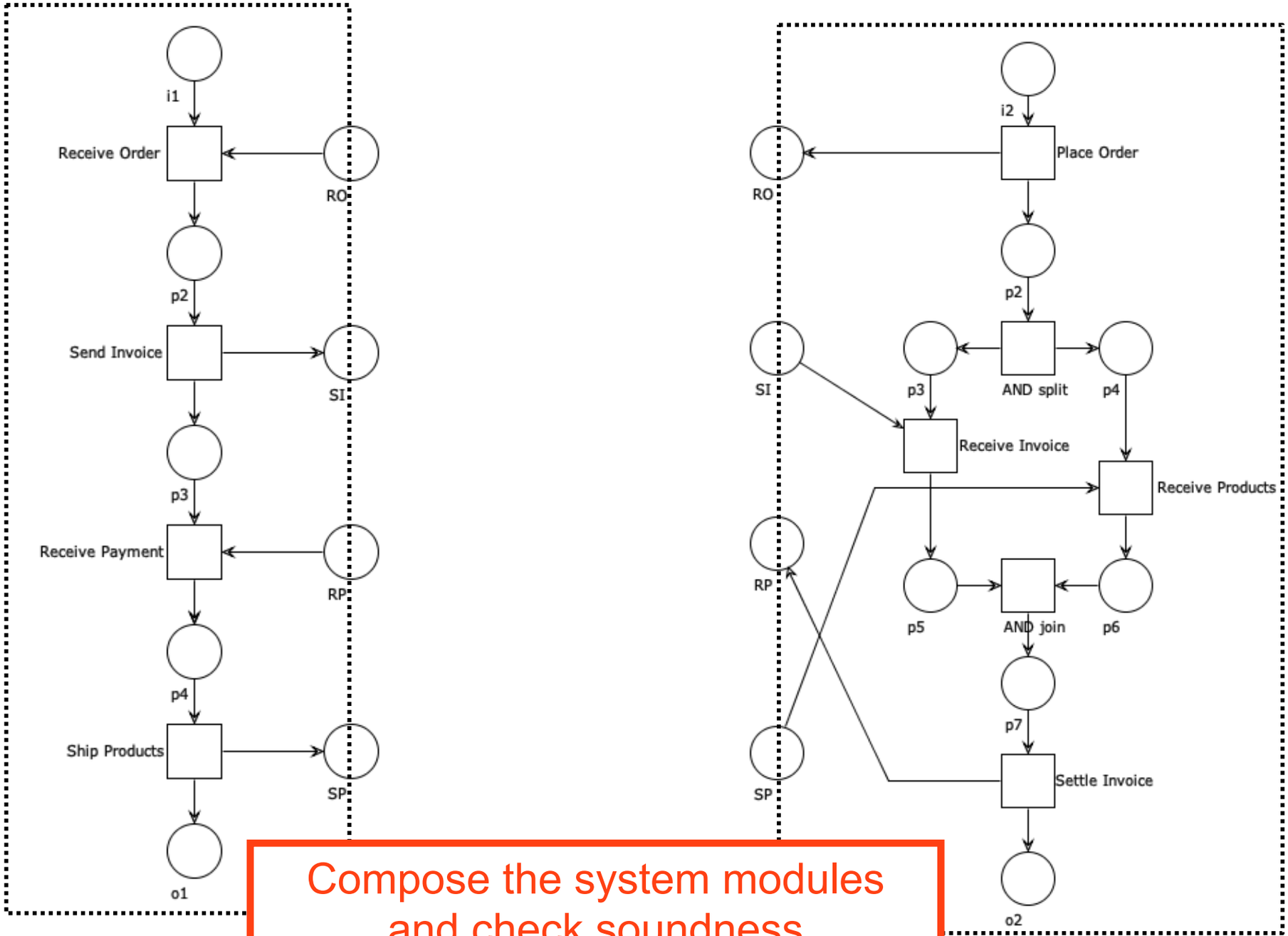


Compose the system modules and check soundness

Exercise

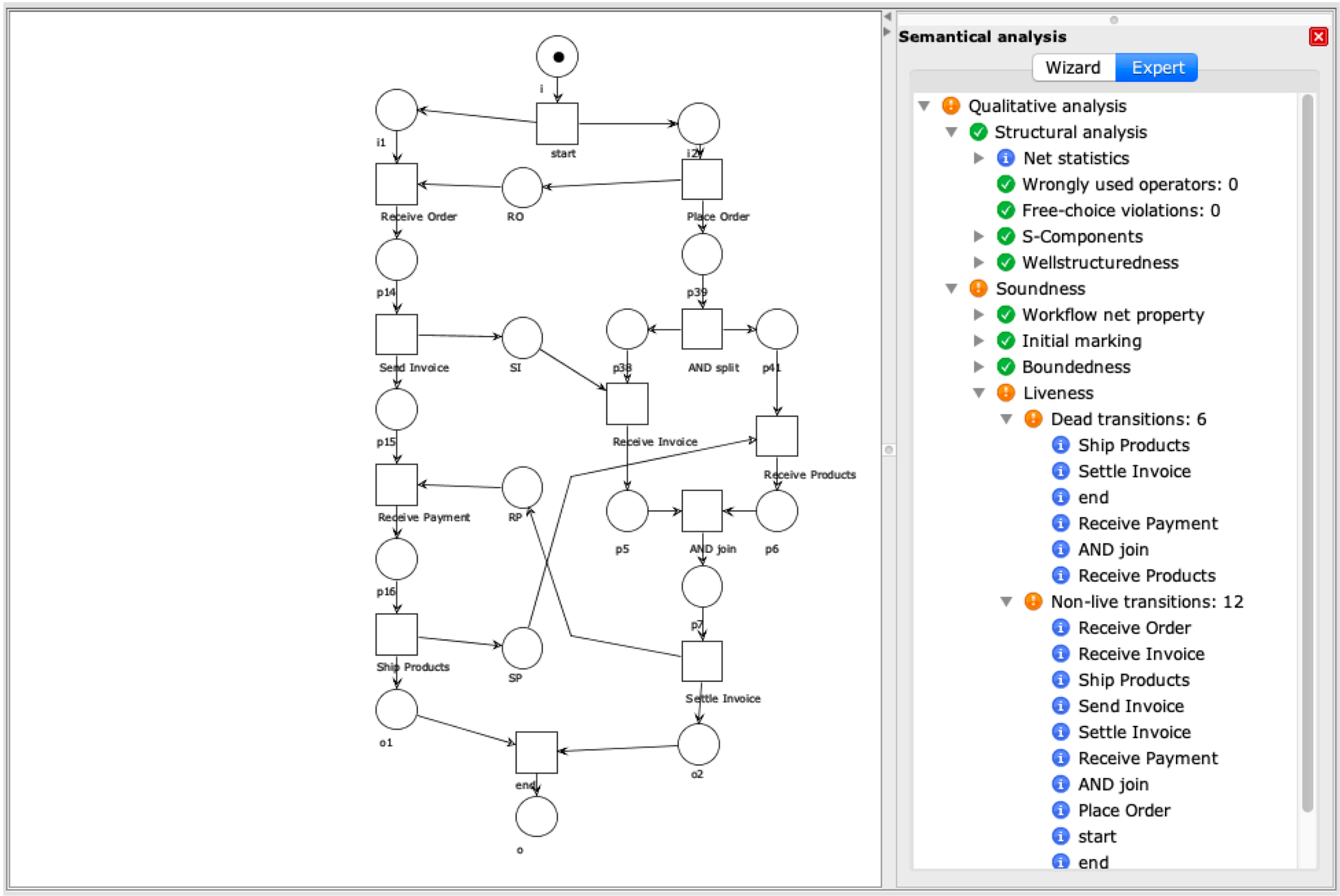

i
i1
start
i2
Receive Order
RO
Place Order
p14
p2
Send Invoice
SI
p3
AND split
p4
p15
Receive Invoice
Receive Payment
RP
p5
p16
Settle Invoice
Receive Products
Ship Products
SP
p7
AND join
p6
o1
o2
end
o
Semantical analysis
Wizard
Expert
Qualitative analysis
Structural analysis
Net statistics
Wrongly used operators: 0
Free-choice violations: 0
S-Components
Wellstructuredness
Soundness
Workflow net property
Initial marking
Boundedness
Liveness

Exercise



Compose the system modules and check soundness

Exercise


i
start
i1
i2
Receive Order
RO
Place Order
p14
p39
Send Invoice
SI
p38
AND split
p41
p15
Receive Invoice
Receive Products
Receive Payment
RP
p5
AND join
p6
p16
p7
Ship Products
SP
Settle Invoice
o1
o2
end
o
Semantical analysis
Wizard
Expert
Qualitative analysis
Structural analysis
Net statistics
Wrongly used operators: 0
Free-choice violations: 0
S-Components
Wellstructuredness
Soundness
Workflow net property
Initial marking
Boundedness
Liveness
Dead transitions: 6
Ship Products
Settle Invoice
end
Receive Payment
AND join
Receive Products
Non-live transitions: 12
Receive Order
Receive Invoice
Ship Products
Send Invoice
Settle Invoice
Receive Payment
AND join
Place Order
start
end

Weak soundness

Problem

When checking behavioural compatibility
the soundness of the overall net
is a too restrictive requirement

Workflow modules are designed separately,
possibly reused in several systems
It is unlikely that every functionality they offer is
involved in each system

Problem

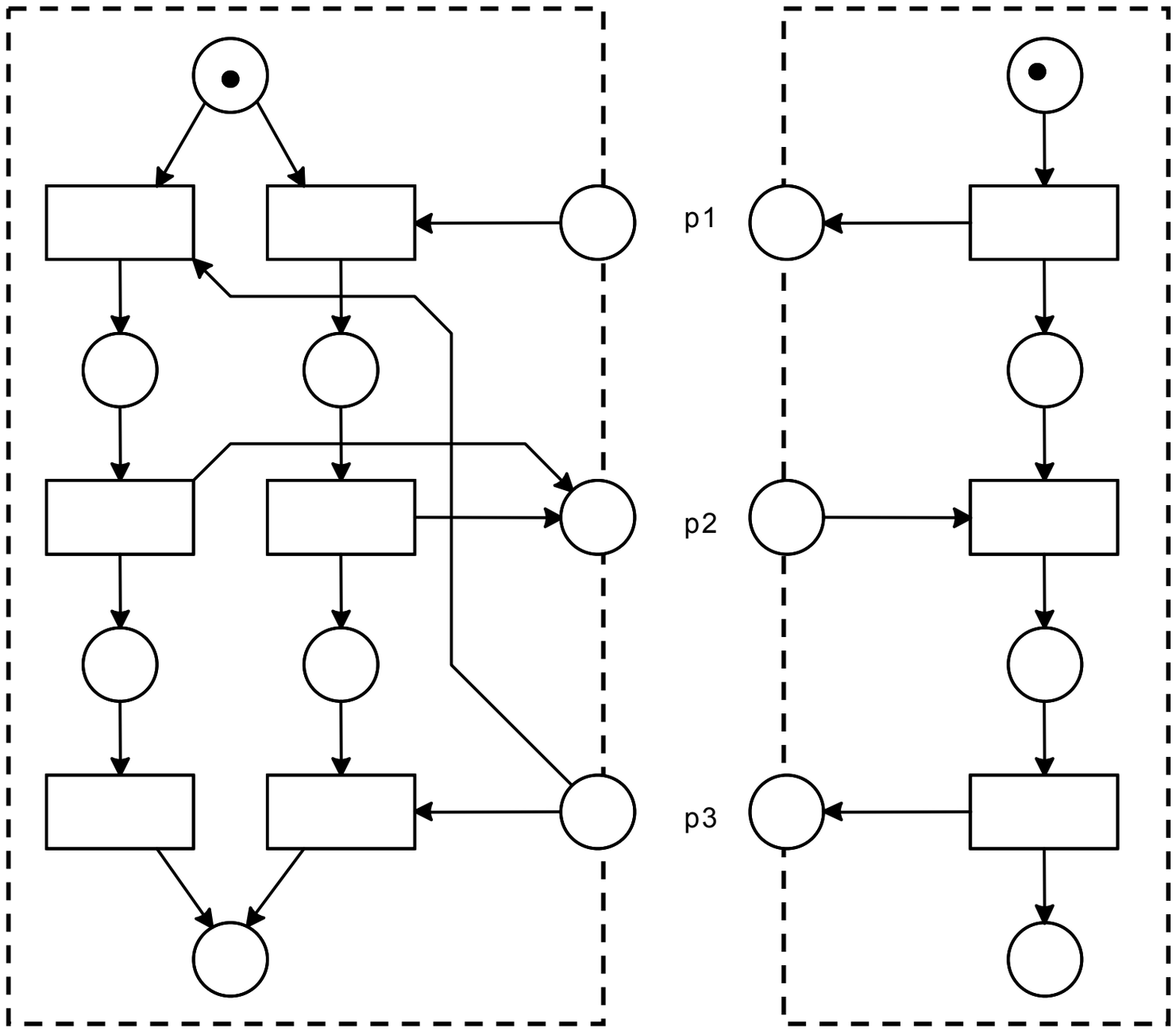
Definition: A workflow net is **weak sound** if it satisfies “option to complete” and “proper completion”

(dead tasks are allowed)

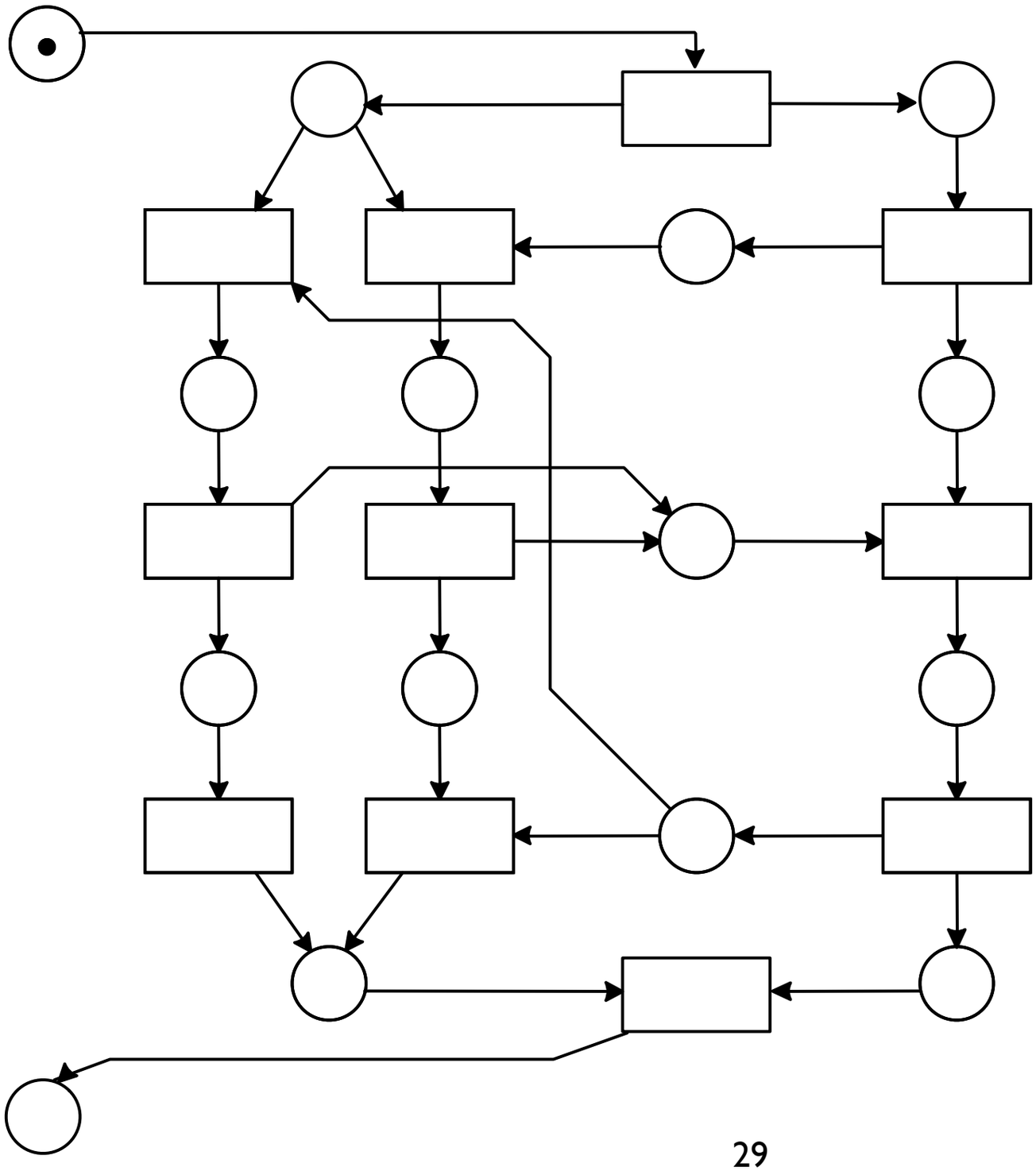
Weak soundness can be checked on the RG

It guarantees deadlock freedom and proper termination of all modules

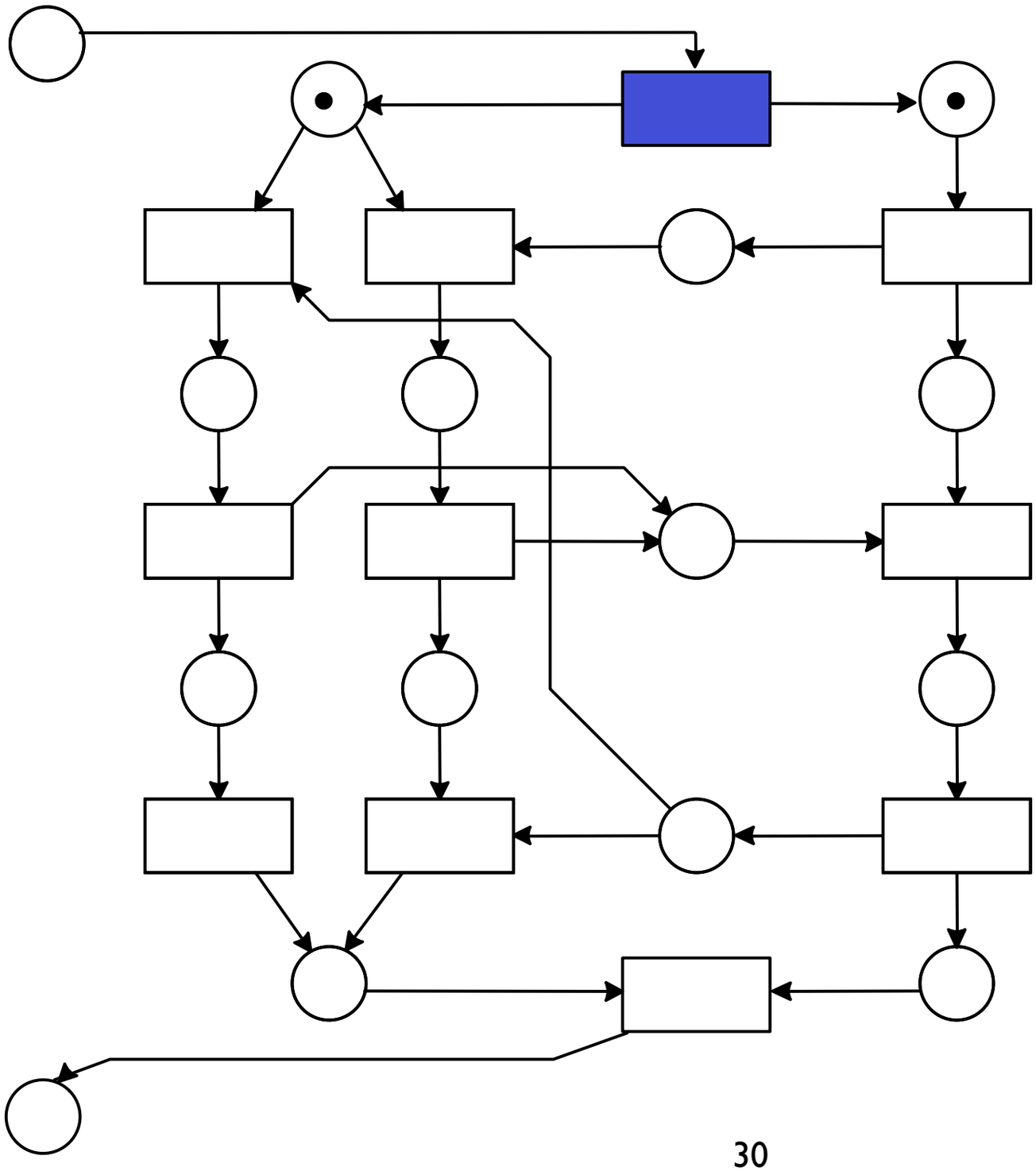
Sound + Sound = ?



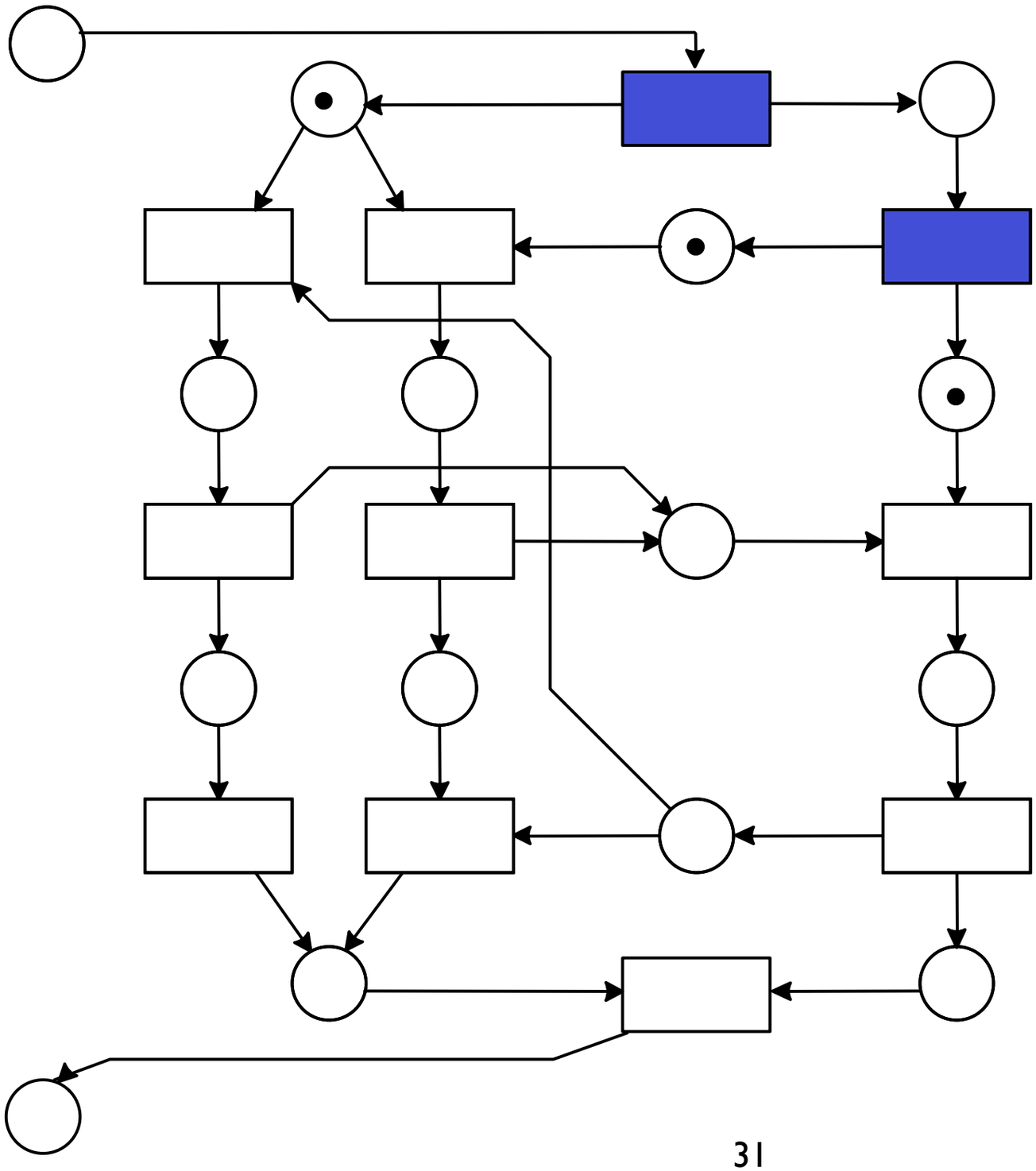
Sound + Sound = not sound



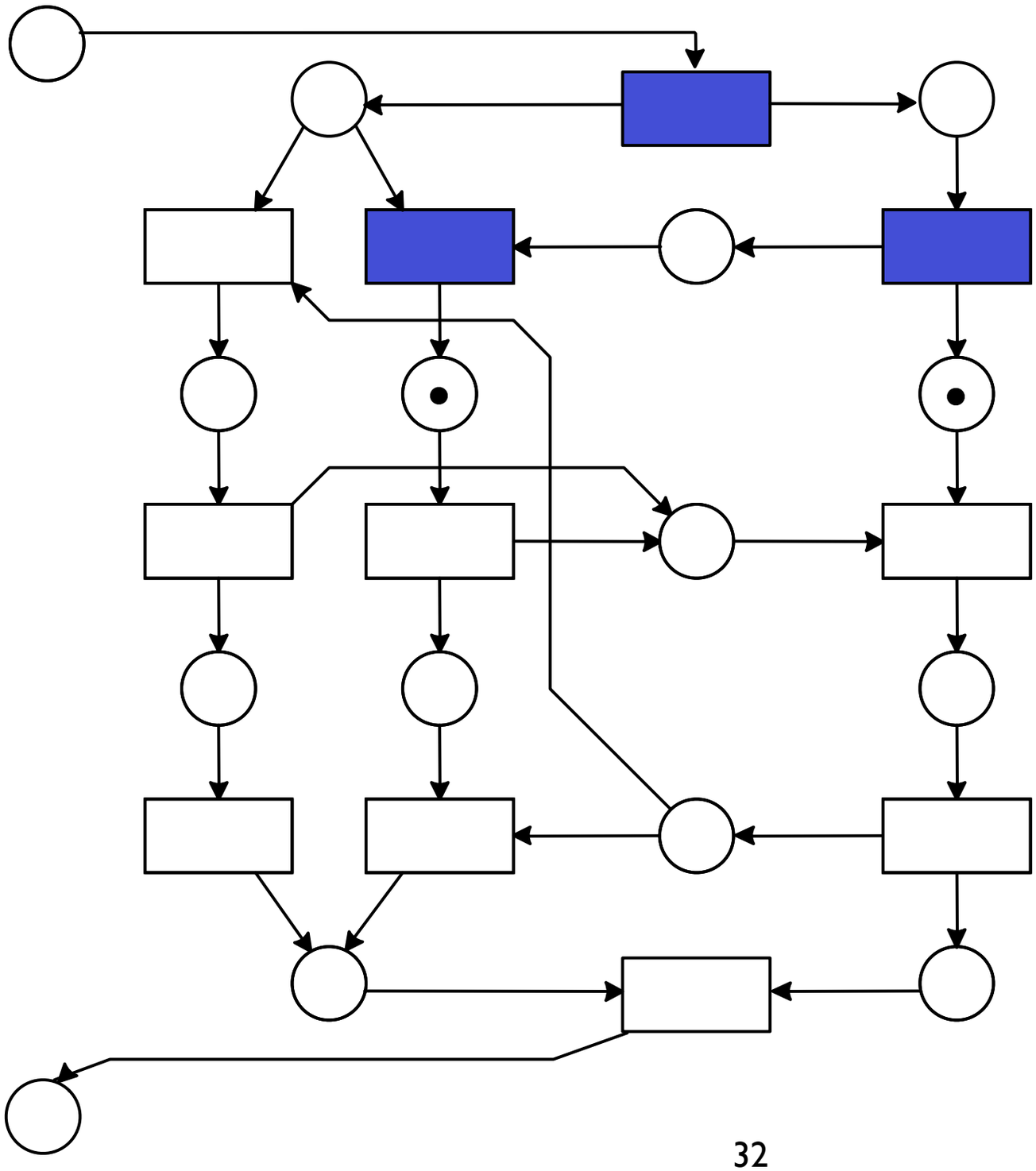
Sound + Sound = not sound



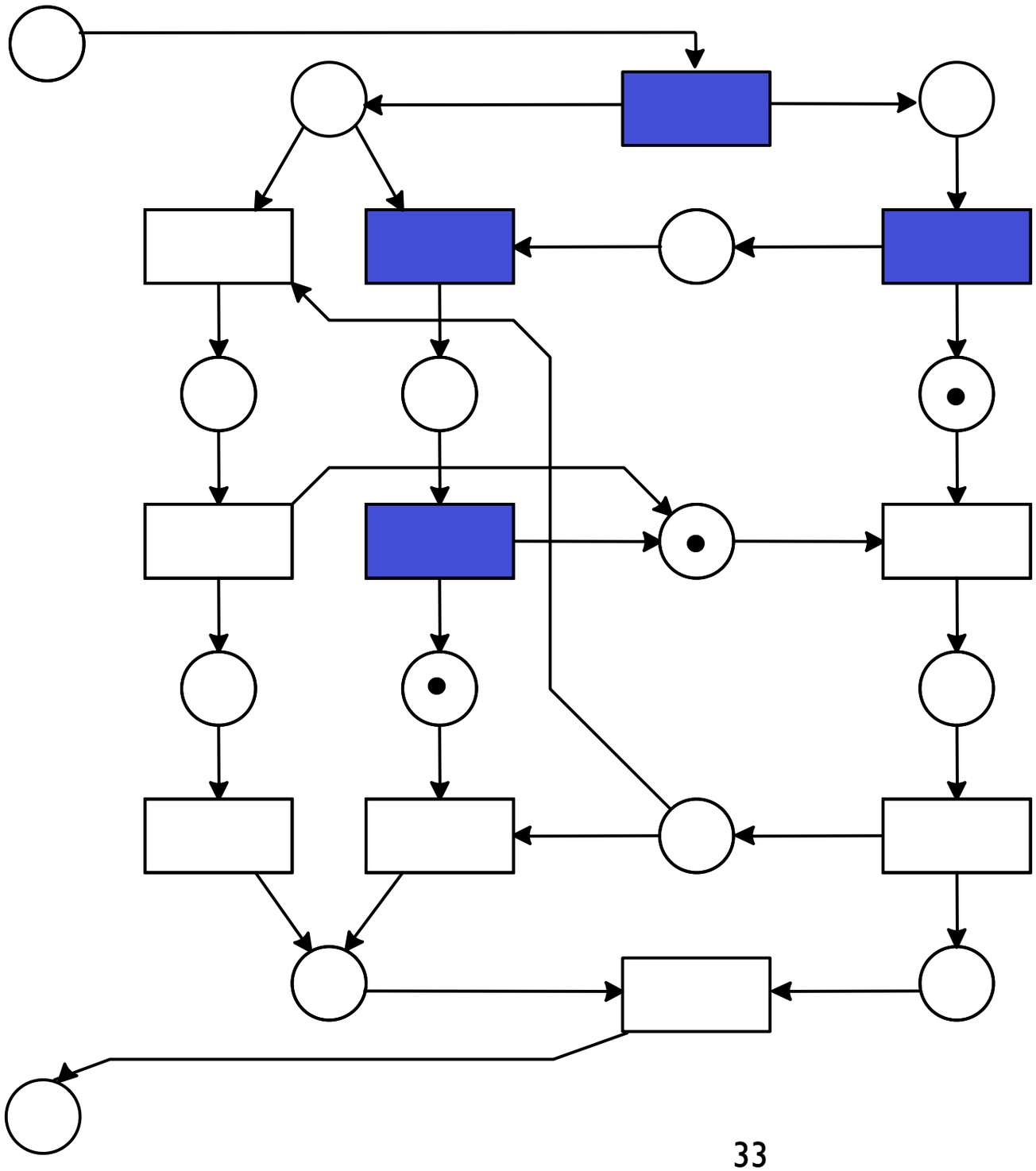
Sound + Sound = not sound



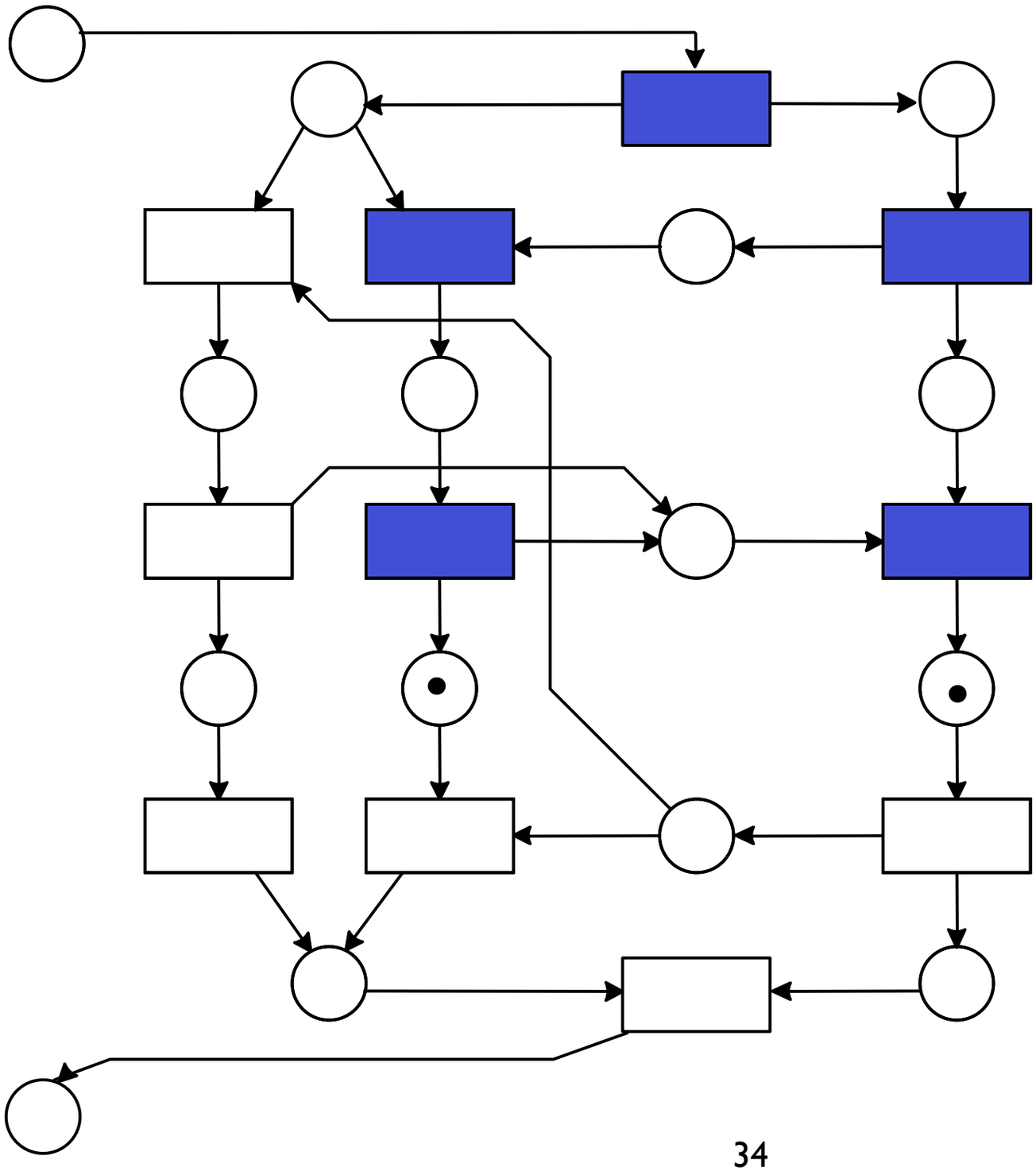
Sound + Sound = not sound



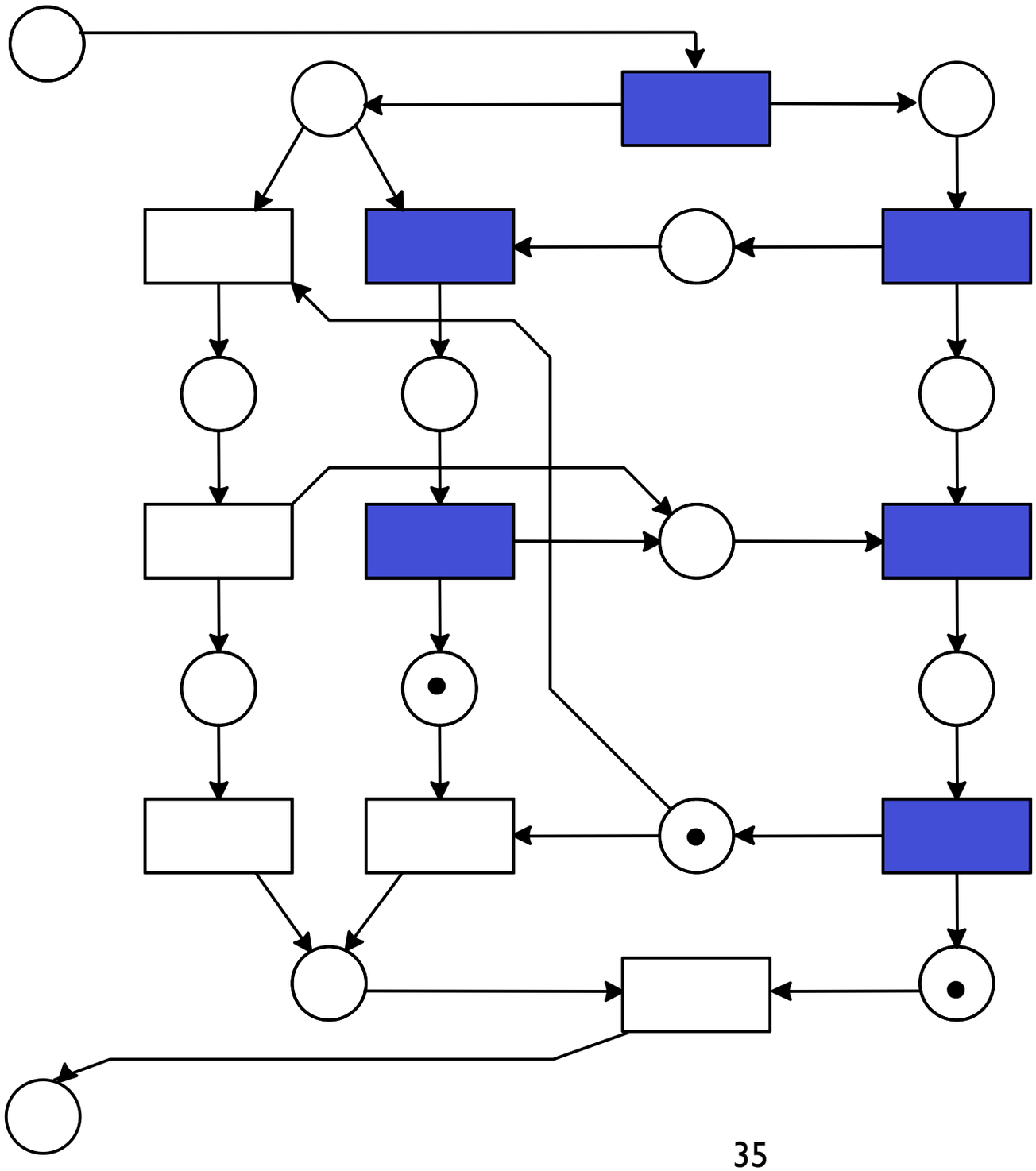
Sound + Sound = not sound



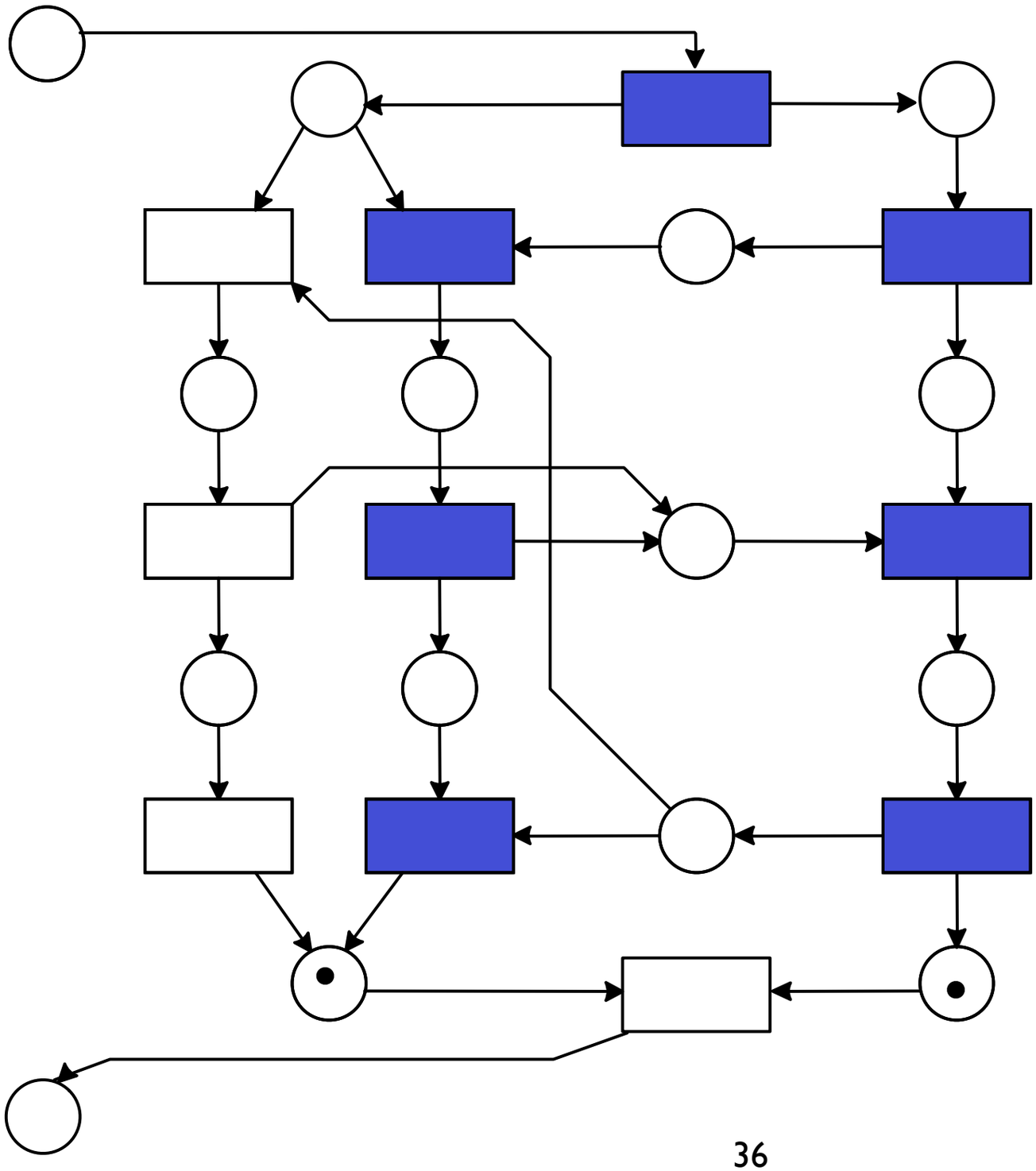
Sound + Sound = not sound



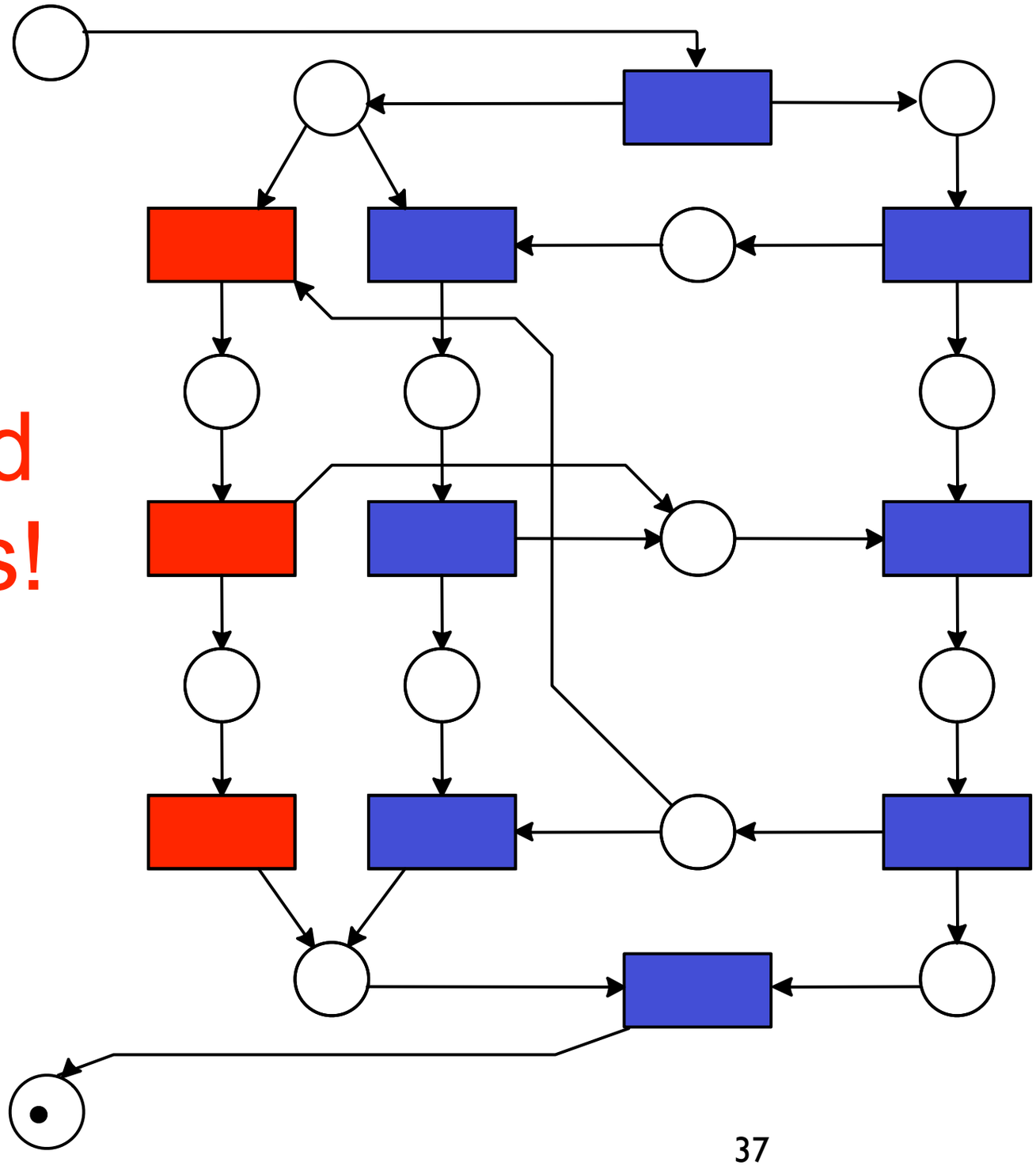
Sound + Sound = not sound



Sound + Sound = not sound

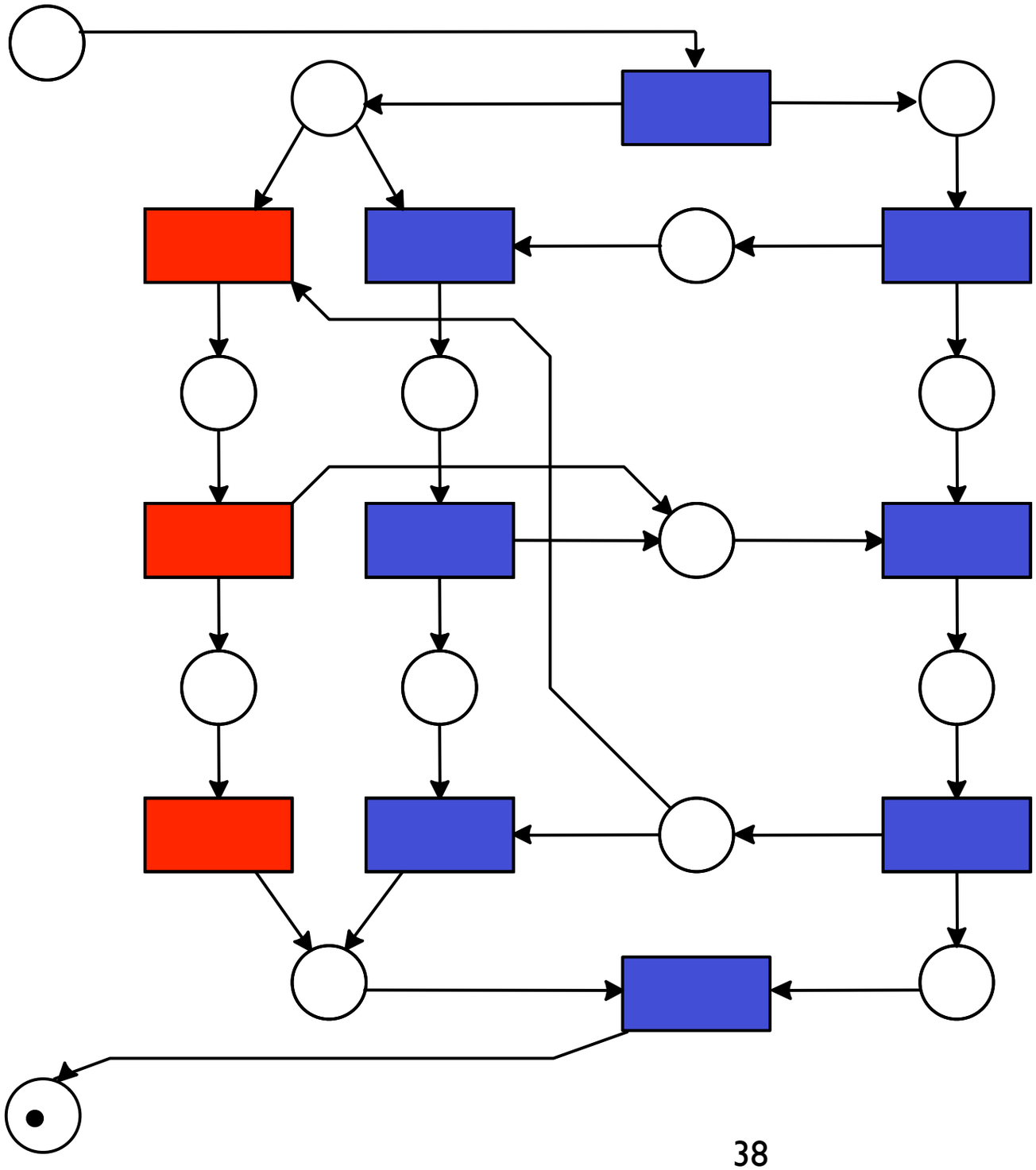


Sound + Sound = not sound



Dead tasks!

Sound + Sound = not sound



Weak Sound!