Refreshing an Old Problem: Relating Nets and Event Structures in the Reversible Setting

G. Michele Pinna

OPCT2023@Bertinoro

Starting point

Petri nets (PN) and event structures (ES) are two well known and studied models for concurrency

PN on the operational side, ES on the denotational one

Question: how PN and ES cope with reversibility and how to relate them?

Starting point

Petri nets (PN) and event structures (ES) are two well known and studied models for concurrency

PN on the operational side, ES on the denotational one

Question: how PN and ES cope with reversibility and how to relate them?

lot of work has been made by others, I'm working on this mainly with Claudio Mezzina and Hernán Melgratti focussing on the Petri Nets side

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\circ$ (enabling/disabling), ...

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\multimap$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\multimap$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

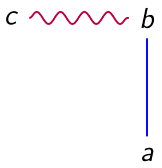
configurations furnish a denotational view of the system

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\circ$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

a prime ES

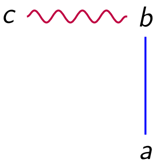


ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\circ$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

a prime ES and the configurations



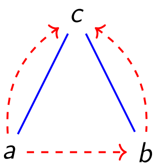
$$
\begin{tikzcd}
\emptyset \arrow[r] \arrow[d] \arrow[dr] & \{a\} \arrow[r] \arrow[d] & \{a, b\} \\
\{c\} \arrow[r] & \{a, c\} &
\end{tikzcd}
$$

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\!\circ$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

an asymmetric ES

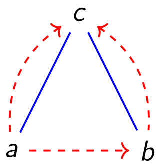


ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\multimap$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

an asymmetric ES and the configurations



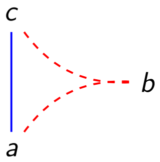
$$
\begin{tikzcd}
 & \{a\} \arrow[r] & \{a, b\} \arrow[dd] \\
\emptyset \arrow[ur] \arrow[dr] & & \\
 & \{b\} & \{a, b, c\}
\end{tikzcd}
$$

ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\circ$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

an inhibitor ES ($\vdash\!\!\circ(\{a\}, b, \{c\})$, $\vdash\!\!\circ(\emptyset, c, \{a\})$)

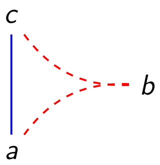


ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\!-\!\!\circ$ (enabling/disabling), ...

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

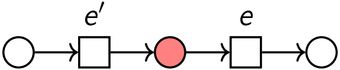
an inhibitor ES and the configurations



$$\begin{tikzcd}
\emptyset \arrow[r] \arrow[d] & \{b\} \arrow[r] & \{a, b\} \arrow[d] \\
\{a\} \arrow[r] & \{a, c\} \arrow[r] & \{a, b, c\}
\end{tikzcd}$$

A closer look to dependencies in nets

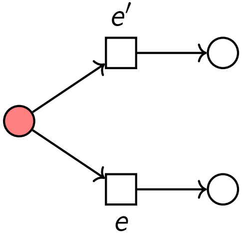
$e < e'$ can be represented as



the partial order is mimicked by the flow relation, there is a shared place between the two events

A closer look to dependencies in nets

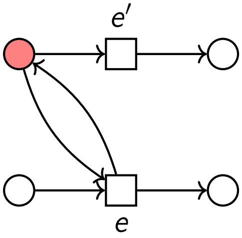
$e \# e'$ can be represented as



there is a shared place between the two events

A closer look to dependencies in nets

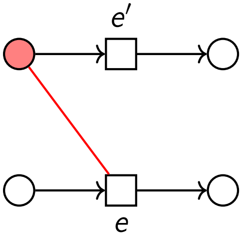
$e \nearrow e'$ can be represented as



dependencies are obtained still via a shared place

A closer look to dependencies in nets

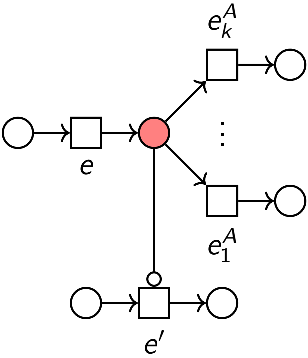
$e \nearrow e'$ can be represented as



dependencies are obtained still via a shared place and read arc can be used as well

A closer look to dependencies in nets

$\vdash\!\!\multimap(\{e\}, e', A)$ can be represented as



dependencies are obtained via a shared place, but some dependencies arise from inhibitor arcs

A closer look to dependencies in nets

and you can continue with the same machinery for other kind of dependencies....

Occurrence Nets

suitable classes of Petri nets are used as a model for concurrent systems (ON)

Occurrence Nets

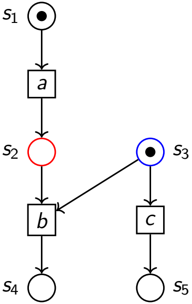
suitable classes of Petri nets are used as a model for concurrent systems (ON)

dependencies among transitions are modeled using shared places

Occurrence Nets

suitable classes of Petri nets are used as a model for concurrent systems (ON)

dependencies among transitions are modeled using shared places

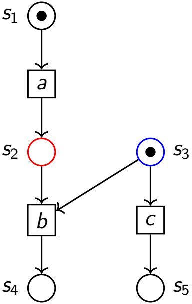


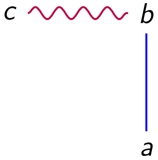
s_2 says that b depends on a and s_3 says that b and c are in conflict

Occurrence Nets

suitable classes of Petri nets are used as a model for concurrent systems (ON)

dependencies among transitions are modeled using shared places





A correspondence between the net and the prime event structures is established and well know

Occurrence Nets

It is possible to add some inhibitor or read arcs to ON to establish similar results for AES or IES

Back to ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\circ$ (enabling/disabling), ...

Back to ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\multimap$ (enabling/disabling), ...

some events $F \subseteq E$ are reversible and their removal is subject to some relations: $\prec$ (reverse causality) and $\triangleleft$ (prevention)

Back to ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\!\multimap$ (enabling/disabling), ...

some events $F \subseteq E$ are reversible and their removal is subject to some relations: $\prec$ (reverse causality) and $\triangleleft$ (prevention)

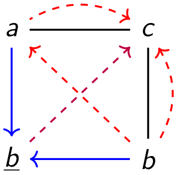
sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

Back to ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\multimap$ (enabling/disabling), ...

some events $F \subseteq E$ are reversible and their removal is subject to some relations: $\prec$ (reverse causality) and $\triangleleft$ (prevention)

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations

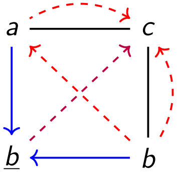


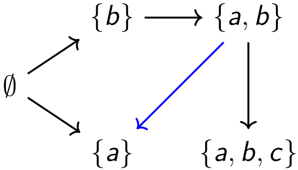
Back to ES

in an ES dependencies among events E are described using some relations (not necessarily binary ones), e.g. $<$ (causality), $\#$ (conflict), $\nearrow$ (weak causality), $\vdash\!\!\!\circ$ (enabling/disabling), ...

some events $F \subseteq E$ are reversible and their removal is subject to some relations: $\prec$ (reverse causality) and $\lhd$ (prevention)

sets of events ($X \subseteq E$) that can stay together represent the states of the systems (configurations), the computations





Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions

Reversing ON?

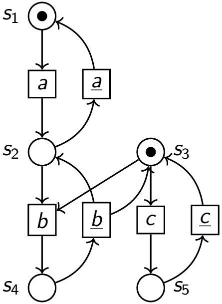
We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always

Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always

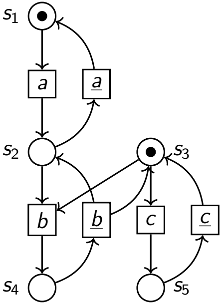


On occurrence nets add the reversing transitions

Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always

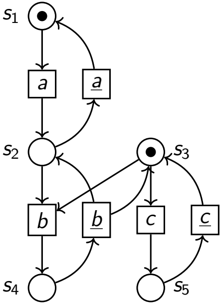


reverse causality and prevention are added using the structure of the net

Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always

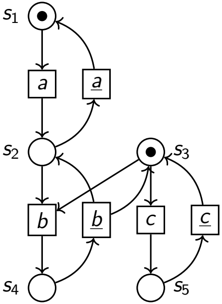


The structure of the net is the limitation: it works with some reversible ES with suitable conditions

Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always

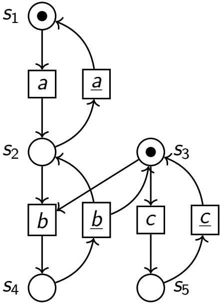


ES where to undo an event you should guarantee that no event causally depending on it is present

Reversing ON?

We do have an ON modeling the forward flow, can we add some reversing transitions

It would be nice to have ON (maybe enriched to take into account other kind of causality) and add the reversing transitions but it do not work always



ES where each event is reversible

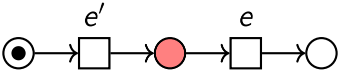
Back to dependencies in nets

Observe: dependencies via shared places can be modeled differently

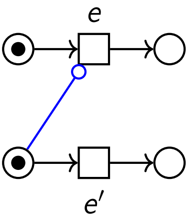
Back to dependencies in nets

Observe: dependencies via shared places can be modeled differently

from

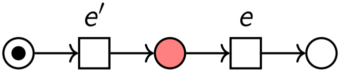


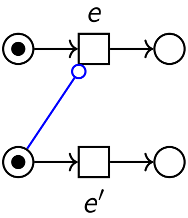
to

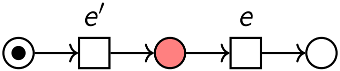


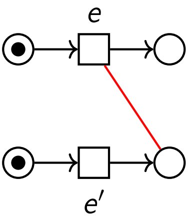
Back to dependencies in nets

Observe: dependencies via shared places can be modeled differently

from 

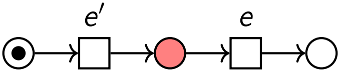
to 

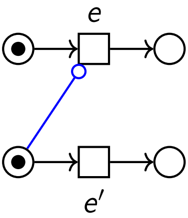
or from 

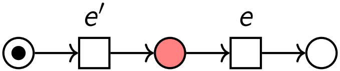
to 

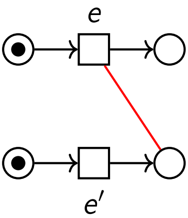
Back to dependencies in nets

Observe: dependencies via shared places can be modeled differently

from 

to 

or from 

to 

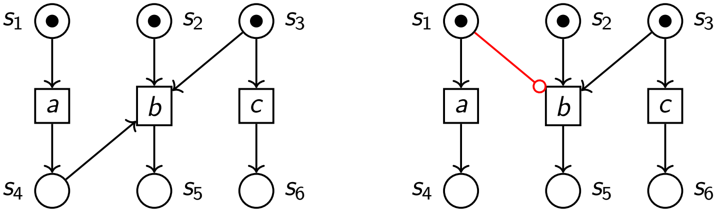
dependencies can be all modeled using contextual arcs

Pushing this idea further

Inhibitor arcs can be used to model causal dependencies

Pushing this idea further

Inhibitor arcs can be used to model causal dependencies



$a < b$ and $b \# c$

Pushing this idea further

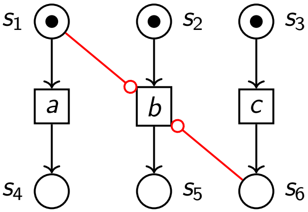
Inhibitor arcs can be used to model causal dependencies

they can be used as well to model asymmetric conflict

Pushing this idea further

Inhibitor arcs can be used to model causal dependencies

they can be used as well to model asymmetric conflict

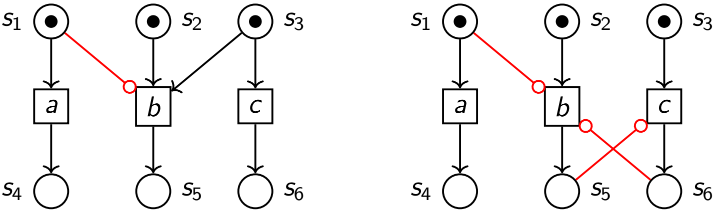


$$a < b \text{ and } b \nearrow c$$

Pushing this idea further

Inhibitor arcs can be used to model causal dependencies

they can be used as well to model asymmetric conflict



$$a < b,\ b \nearrow c \text{ and } c \nearrow b$$

$b \nearrow c$ and $c \nearrow b$ model $b \# c$

Reversing transitions again

Can we apply the same intuition for the reversing transitions?

Reversing transitions again

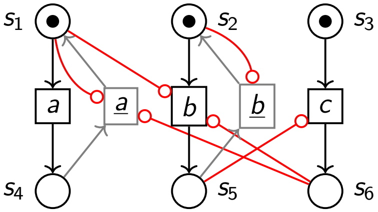
Can we apply the same intuition for the reversing transitions?

reverse causality and prevention may be modeled again with the same intuition: inhibitor arcs

Reversing transitions again

Can we apply the same intuition for the reversing transitions?

reverse causality and prevention may be modeled again with the same intuition: inhibitor arcs



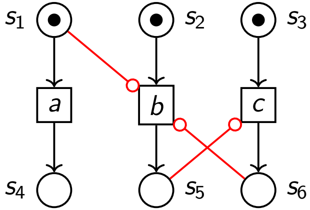
here just a and b are reversed

What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

What we have done

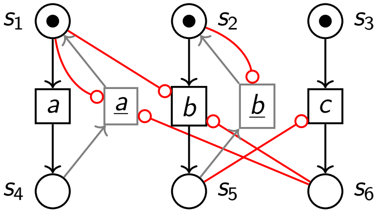
On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)



What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

also the ones on the reversing transitions

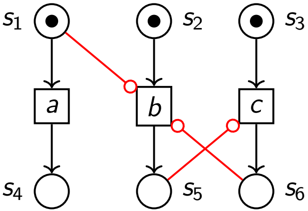


the reverse causality and prevention are on the reversing transitions

What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

forgetting the reversing transitions we get a



What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

To this kind of net we can associate a reversible ES

What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

To this kind of net we can associate a reversible ES and vice versa

What we have done

On the net side we have defined nets where all the dependencies among transitions are modeled using inhibitor arcs (called Causal Nets, CN)

To this kind of net we can associate a reversible ES and vice versa

We have done it for reversible asymmetric ES and reversible prime ES

Do we lose something?

No: each ON can be transformed into a CN

if the CN has suitable characteristics we can retrive an ON

Categorical view

reversible event structures can be turned into categories

Categorical view

reversible event structures can be turned into categories

we can do the same for reversible CN

Categorical view

reversible event structures can be turned into categories

we can do the same for reversible CN

and we have an adjuntion between these categories

Open problems (some of them)

reversible ES can be used to give semantics to a reversible process algebra, how could we use reversible CN?

Open problems (some of them)

reversible ES can be used to give semantics to a reversible process algebra, how could we use reversible CN?

with Claudio and Hernán we did but we used something that are almost ON!

Open problems (some of them)

reversible ES can be used to give semantics to a reversible process algebra, how could we use reversible CN?

with Claudio and Hernán we did but we used something that are almost ON!

On the net side it seems easy to add other kind of dependencies (arising from modifiers), how these do apply to the reversible ES side? The reverse causality and prevention relation have few constraints, can we reason a bit more on them?

Open problems (some of them)

reversible ES can be used to give semantics to a reversible process algebra, how could we use reversible CN?

with Claudio and Hernán we did but we used something that are almost ON!

On the net side it seems easy to add other kind of dependencies (arising from modifiers), how these do apply to the reversible ES side? The reverse causality and prevention relation have few constraints, can we reason a bit more on them?

How these do relate to Irek & Iain & Ivan axiomatic characterisation?

Open problems (some of them)

reversible ES can be used to give semantics to a reversible process algebra, how could we use reversible CN?

with Claudio and Hernán we did but we used something that are almost ON!

On the net side it seems easy to add other kind of dependencies (arising from modifiers), how these do apply to the reversible ES side? The reverse causality and prevention relation have few constraints, can we reason a bit more on them?

How these do relate to Irek & Iain & Ivan axiomatic characterisation?

...

Thank you!