Token Multiplicity in Reversing Petri Nets Under the Individual Token Interpretation

Anna Philippou and Kyriaki Psara

Department of Computer Science

University of Cyprus

Reversible computation

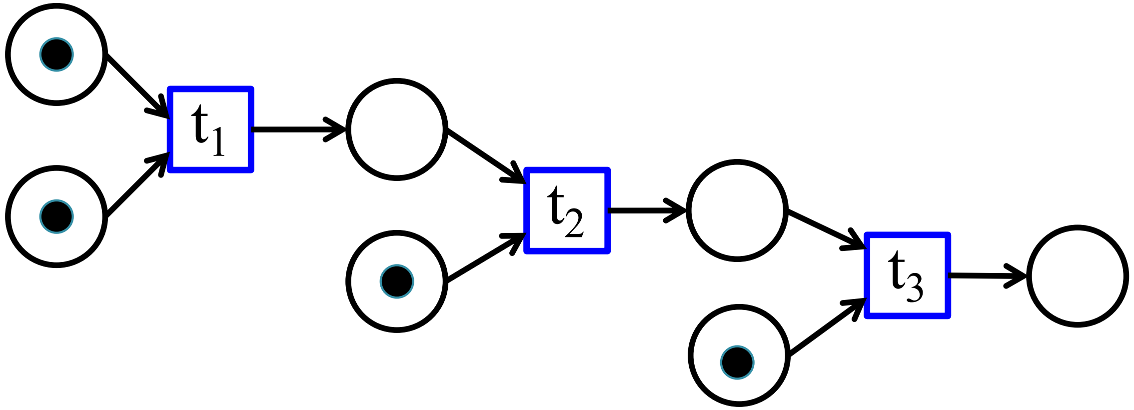
- Unconventional form of computing where computation can run backwards as naturally as it can go forwards
  - Every set of instructions has the ability to be carried out in reverse order
- Origins
  - Landauer 1960: logically irreversible operations result in bit erasure that causes heat dissipation, i.e. loss of energy
  - Reversible logic gates and circuits may lead to low-energy computing
- Applications
  - Program debugging and testing
  - Programming abstractions for fault-tolerant systems
    - transactions, system-recovery schemes, checkpoint-rollback protocols
  - Biological modelling, robotics, quantum computation, etc.

Formal models for reversible computation

- Investigate the theoretical foundations of reversibility
- Develop computation models for reversible systems and techniques for their analysis
  - Process calculi, event structures, Petri nets, …
- Forms of reversibility
  - Backtracking: executing past actions in the exact inverse order in which they occurred
  - Causal-order reversibility: an action can be undone if all its effects have already been reversed
  - Out-of-causal-order reversibility: actions can be undone in an out-of-causal order

Petri nets (PNs)

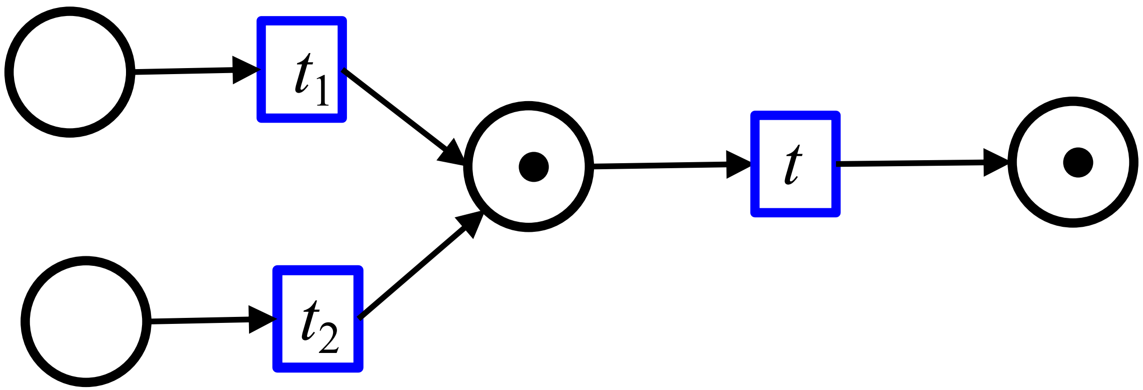
- A powerful graphical language for discrete event systems
- Rich mathematical theory
- Wide variety of tools
- A PN consists of
  - A set of places
  - A set of transitions
  - A set of edges
  - A set of tokens



A transition can fire if the places incoming to the transition have tokens. The effect of firing a transition is to transfer the tokens from the incoming places to the outgoing places.

Reversing computation in PNs – challenges

- What does one need to remember in order to reverse transitions in PNs?
- How do we identify legitimate backward moves?
- Key challenges: backward conflicts and causality



In our previous work…

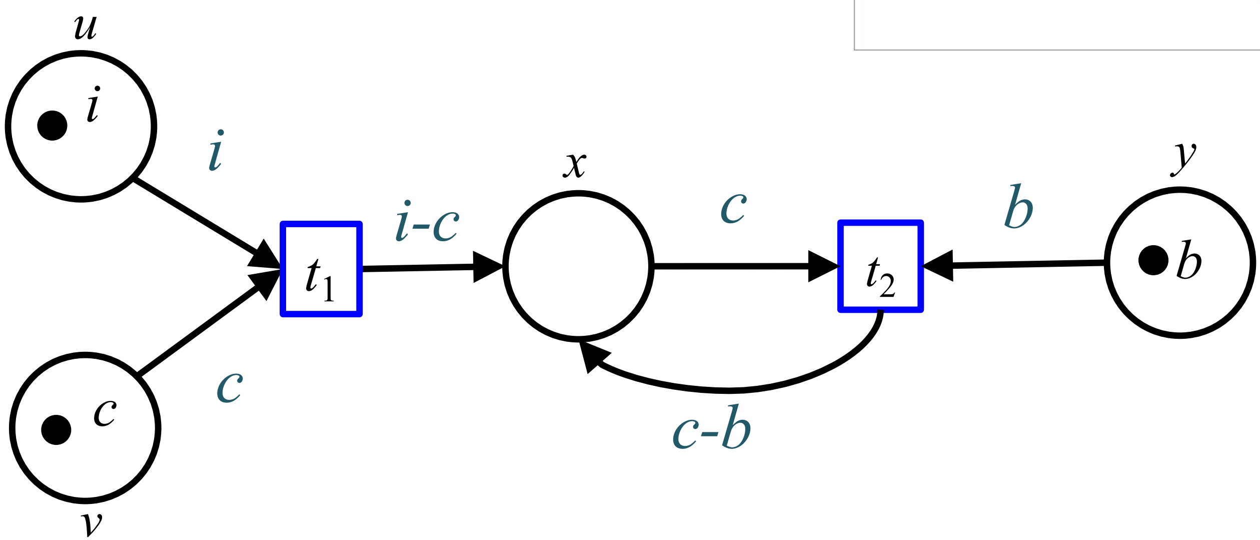
- Reversing Petri Nets (RPNs)
  - High-level Petri nets with individual tokens that satisfy the conservativeness property, where functions form bonds between tokens
  - Support backtracking, causal-order, and out-of-causal order reversibility
- Individual tokens
  - Each token is identified by its name/type: a, b, …
  - Tokens are preserved
  - Tokens can be bonded together: a-b, b-c, …
- Transitions
  - Move tokens from incoming to outgoing places
  - May form or break bonds between tokens
  - Reversing a transition involves destroying/creating the bonds created/destroyed by the transition and returning tokens from outgoing to incoming places
- Histories
  - Transitions are associated with keys, which capture the order in which transitions were executed
  - **Convey information to resolve backward nondeterminism and establish a causality relation**
  - A state is a pair $\langle M,H \rangle$ where M is the marking and H the history

Execution

- Forward execution: $\langle M, H\rangle \xrightarrow{t} \langle M', H'\rangle$
  - A transition may be executed if the required tokens are available
  - Tokens and their connected components are transferred from the incoming to the outgoing places of the transition
  - Bonds can be created/destroyed
- Causal-order reversing: $\langle M, H\rangle \overset{t}{\rightarrow}_c \langle M', H'\rangle$
  - A transition can be reversed if all transition occurrences it has caused have already been reversed
  - Tokens and their connected components are moved from the outgoing places of the transition to its incoming places
  - Bonds created/broken by the transition are broken/created

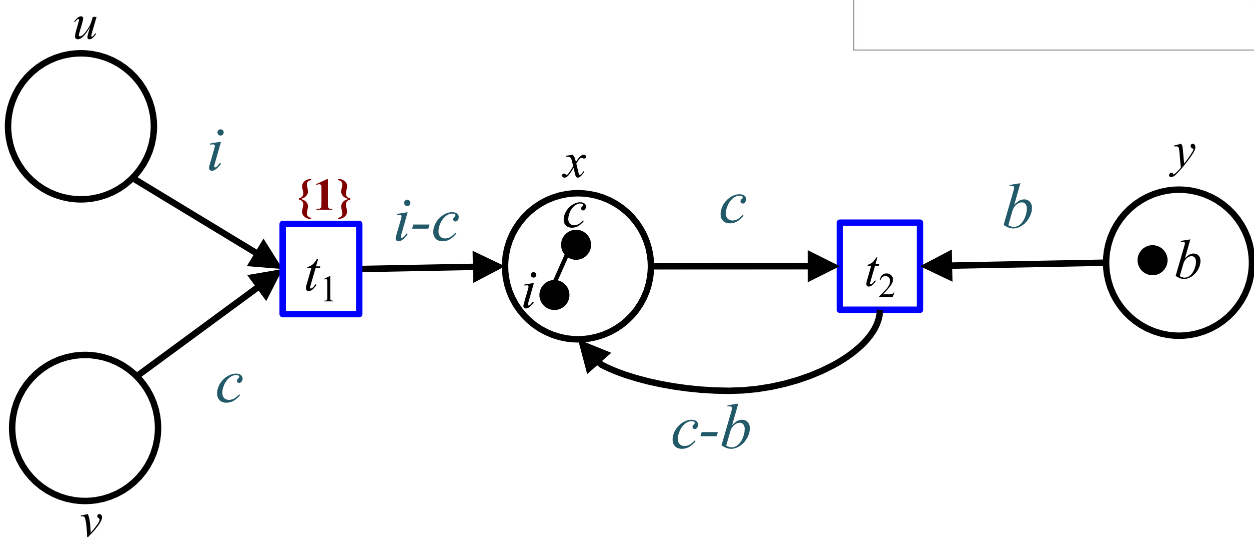
Example – Pen assembly/disassembly

- System with three components:
  - The ink, *i*
  - The cup, *c*
  - The button, *b*



Example – Pen assembly/disassembly

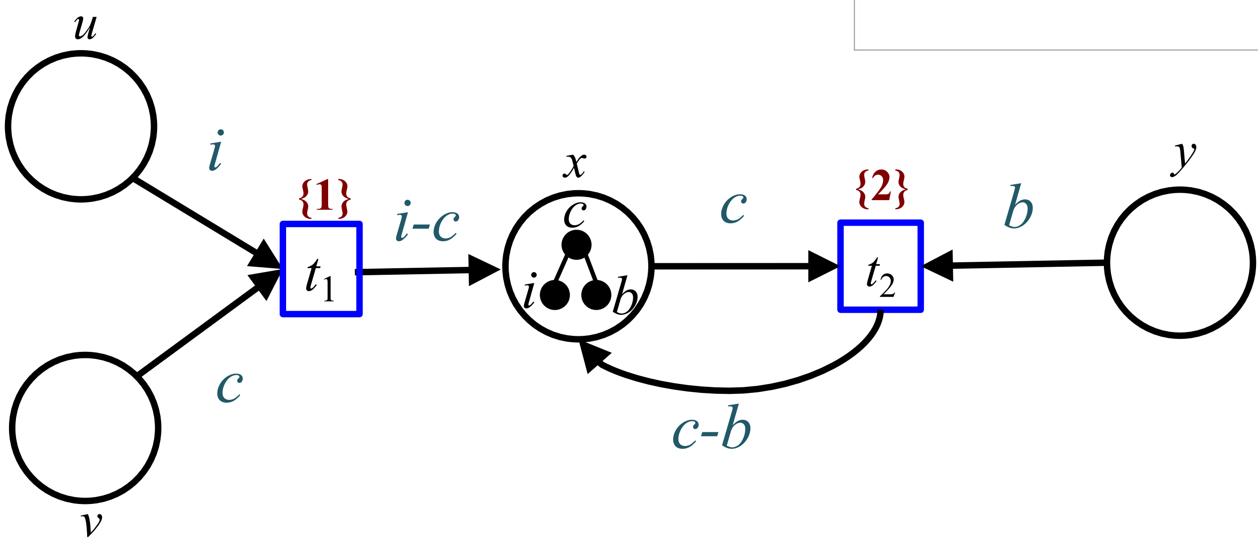
- System with three components:
  - The ink, i
  - The cup, c
  - The button, b



After executing t_1

Example – Pen assembly/disassembly

- System with three components:
  - The ink, i
  - The cup, c
  - The button, b

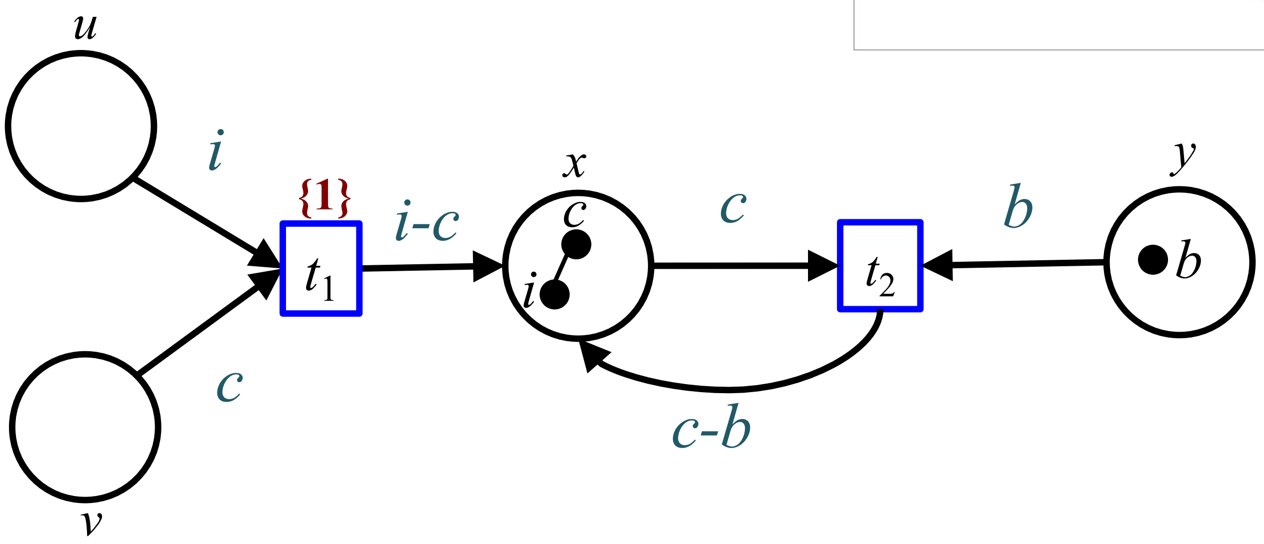


After executing t_2.

Transition t_1 has caused transition t_2.

Example – Pen assembly/disassembly

- System with three components:
  - The ink, i
  - The cup, c
  - The button, b

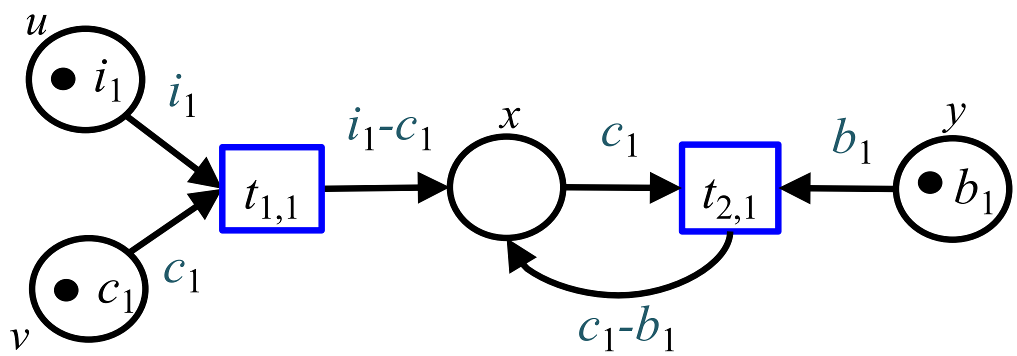


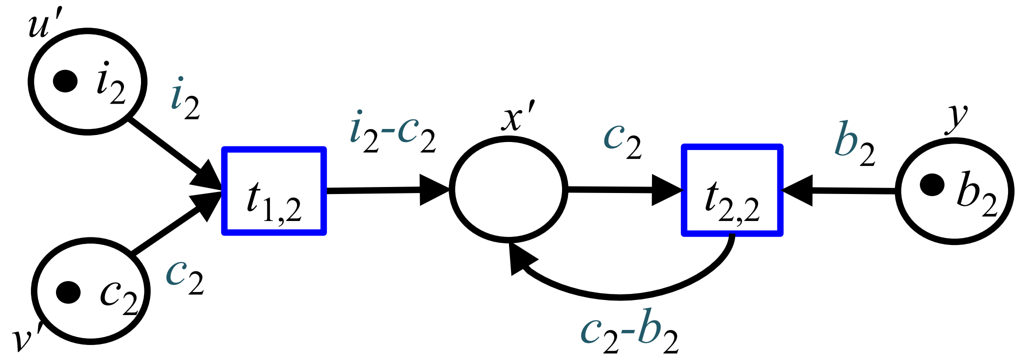
Only transition t_2 can be reversed

Two pens?

- RPNs feature named tokens that are pairwise distinct
  - There exists exactly one token of each “type”
- How can we model a system with two pens?
- RPN with six tokens:
  - The inks, i_1, i_2
  - The cups, c_1, c_2
  - The buttons, b_1, b_2

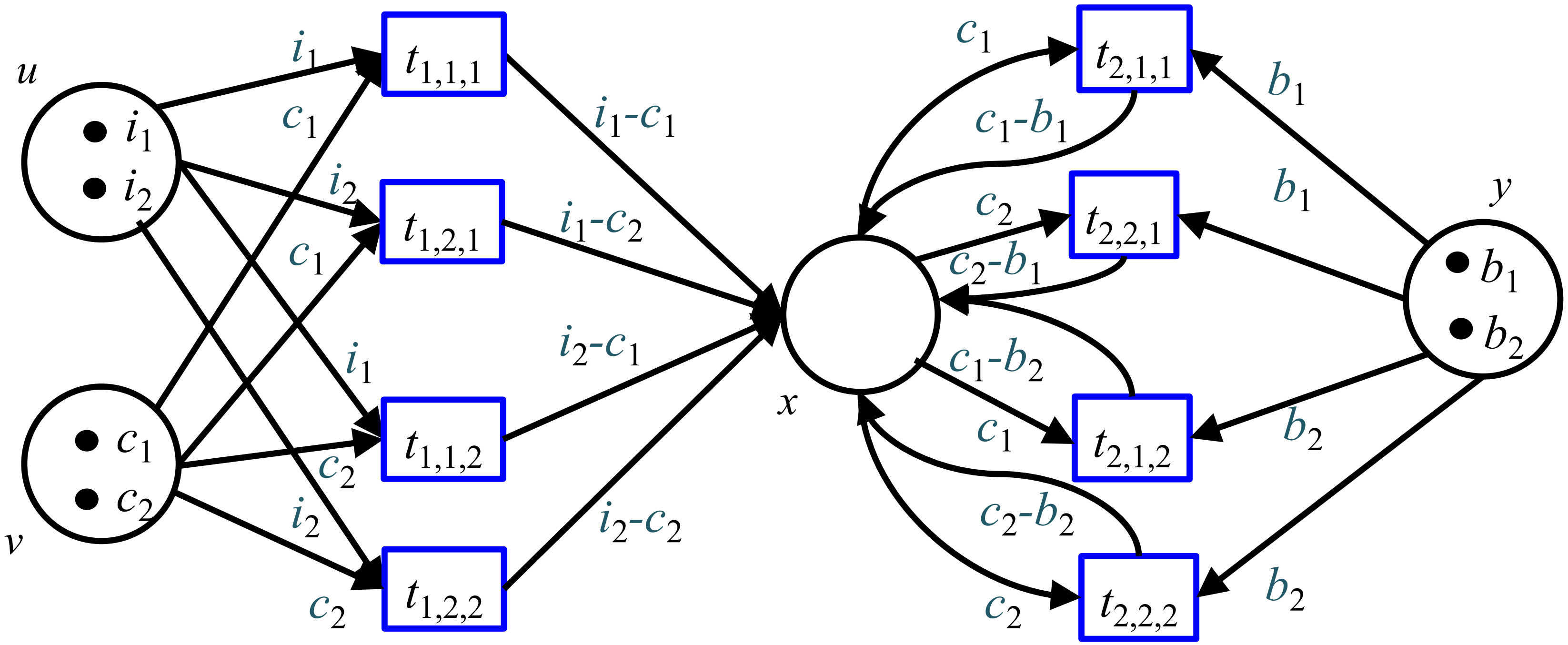
Two pens?





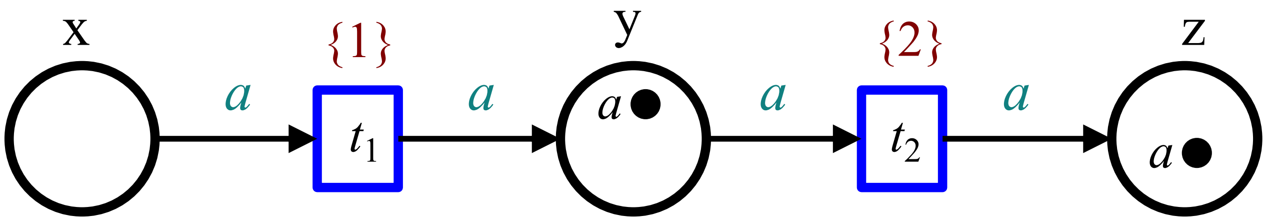
Tokens of the same type should be indistinguishable.

Two pens

u
v
x
y
i1
i2
c1
c2
b1
b2
t1,1,1
t1,2,1
t1,1,2
t1,2,2
t2,1,1
t2,2,1
t2,1,2
t2,2,2
i1-c1
i1-c2
i2-c1
i2-c2
c1-b1
c2-b1
c1-b2
c2-b2

Introducing multiple tokens



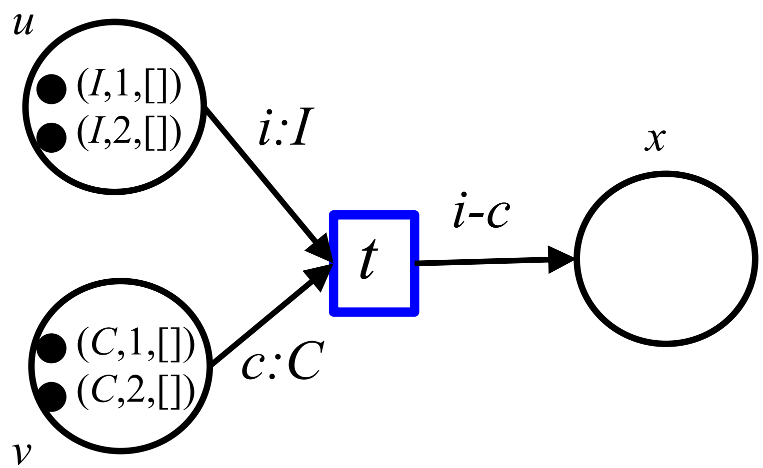
- Has t_1 caused t_2?
- Two approaches to token multiplicity
  - Individual-token interpretation: each token is considered unique and is identified by its causal path
  - Collective-token interpretation: tokens of the same type are considered as identical
- Is it possible to reverse t_1?
  - Individual-token interpretation: It depends on which a was used to fire t_2
  - Collective-token interpretation: Yes (as long as there is an available token)

Individual-token interpretation

- Token types
  - Multiple *instances* of a token type may exist in a net
  - Token instances of the same type have the same capabilities

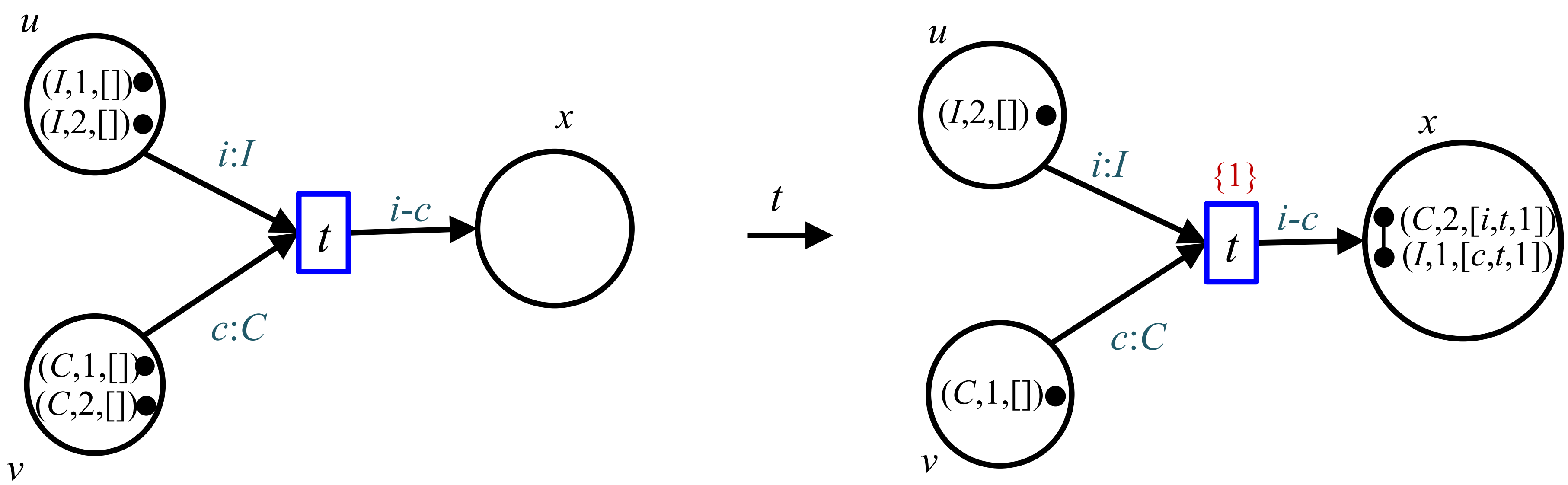
- Arcs are associated with typed variables
  - $u{:}A$: request for a token of type A

- Tokens instances carry their causal path
  - $(A, i, [(k_1,t_1,v_1),\ldots,(k_n,t_n v_n)])$: token of type A has participated in transitions $t_1,..,t_n$, with keys $k_1,..,k_n$ forming variable associations with $v_1,...,v_n$



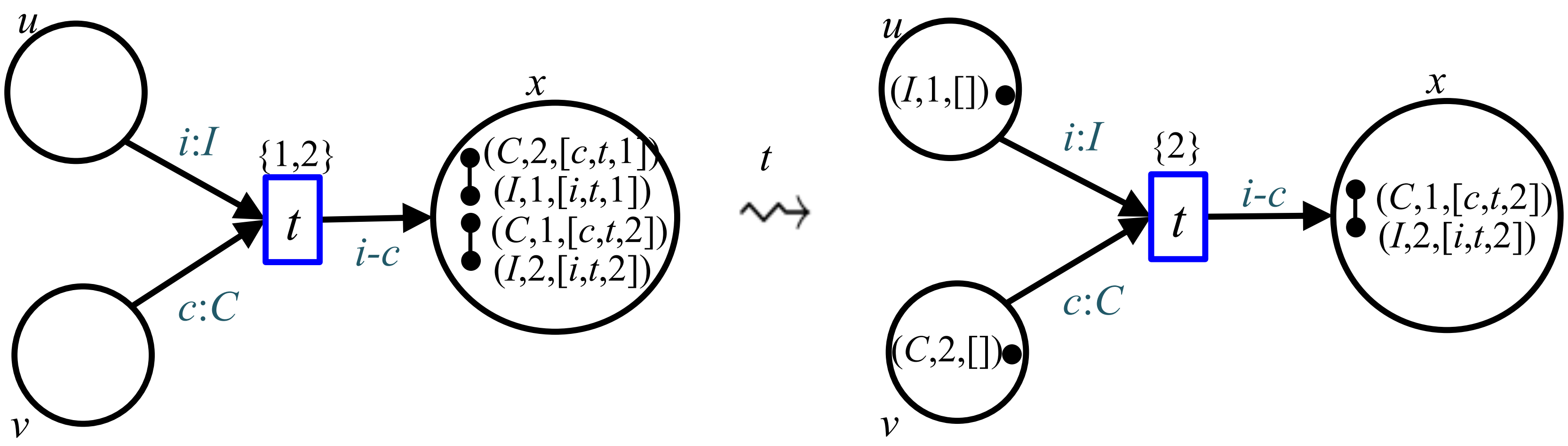
Forward execution

- A transition is enabled if:
  - There is a collection of token/bond instances in the incoming places of the transition that can be instantiated to the incoming variables of the transition.
- Firing a transition results in:
  - Transferring all relevant tokens from the incoming places to the outgoing places and creating/destroying bonds as specified by the transition
  - Extending the history of the transition with the next available key in ascending order
  - Updating the causal path of the tokens involved in the newly-executed transition



Causal-order reversing

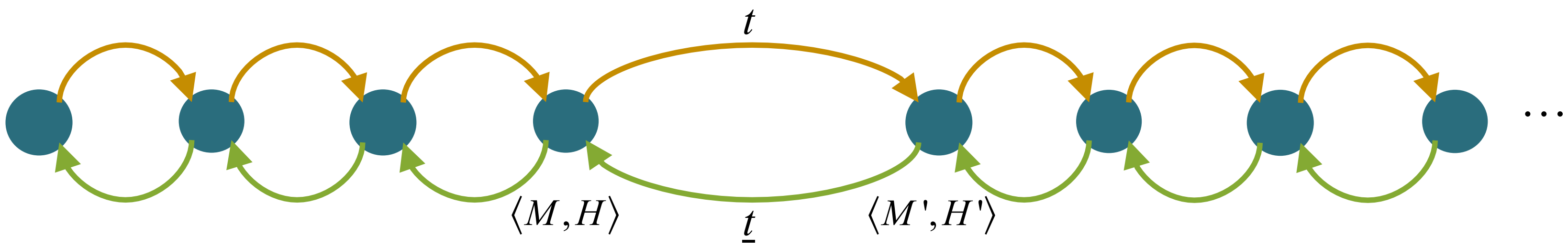
- A causal link exists between two transitions if one produces tokens used to fire the other
- A transition occurrence can be reversed if:
  - All the tokens/bond instances involved in firing the occurrence have not engaged in any further transitions or, if they did, these transitions have been reversed
- Reversing a transition results in:
  - Transferring all relevant token/bond instances from the outgoing places to the incoming places forming/breaking bonds as necessary
  - The history of the transition is updated by removing the key of the reversed transition occurrence
  - Updating the causal path of the tokens by removing the record of the reversed transition



Causal-order reversibility

Loop Lemma

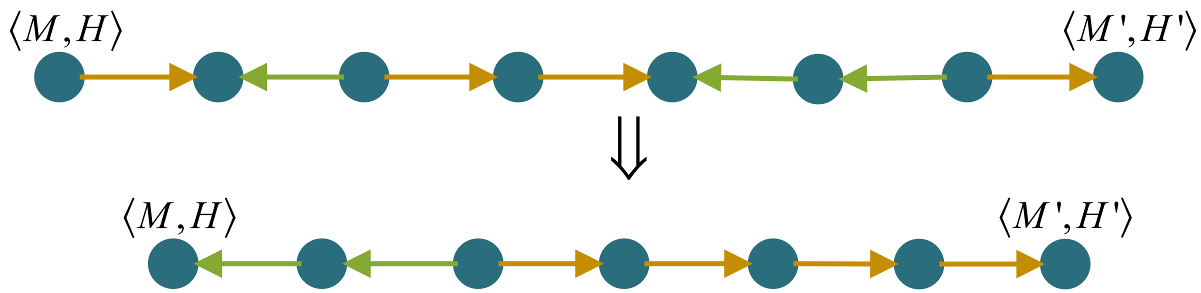
For any forward transition $\langle M, H\rangle \xrightarrow{t} \langle M', H'\rangle$ there is a causal-order reverse transition $\langle M', H'\rangle \xrightarrow{t}_c \langle M, H\rangle$, and vice versa.



Causal-order reversibility

Parabolic Lemma

For any execution $\langle M, H\rangle \xrightarrow{\sigma} \langle M', H'\rangle$ where σ is a sequence of both forward and reverse transitions, there exists $\langle M, H\rangle \xrightarrow{\underline{r}} \langle M'', H''\rangle \xrightarrow{r'} \langle M', H'\rangle$ where $\underline{r}$ is a sequence of reverse actions and *r'* a sequence of forward actions.



Causal-order reversibility

Causal Consistency Theorem

Two sequences of transitions lead to the "same" states

$$\langle M, H\rangle \xrightarrow{\sigma_1} \langle M', H'\rangle \ , \ \langle M, H\rangle \xrightarrow{\sigma_2} \langle M', H'\rangle$$

if and only if σ_1 and σ_2 differ only by reordering of *independent* transitions and inserting or removing pairs of *opposite* actions.

Multiple vs single tokens

- What is the expressiveness relation between MPRNs and RPNs?

A *Labelled Transition system* (LTS) is a tuple $(Q,E,\rightarrow,I)$ where

- Q is a countable set of states
- E is a countable set of actions
- $\rightarrow$ is the step transition relation
- I is the initial state

Two LTSs $(Q_1,E_1,\rightarrow_1,I_1)$ $(Q_2,E_2,\rightarrow_2,I_2)$ are *isomorphic* if they differ only in the names of their states and events, i.e. if there are bijections $\gamma{:}Q_1\rightarrow Q_2$ and $\eta{:}E_1\rightarrow E_2$ such that $\gamma(I_1) = I_2$ and $p \xrightarrow{u}_1 q$ *if and only if* $\gamma(p) \xrightarrow{\eta(u)}_2 \gamma(q)$.

Multiple vs single tokens

- An SRPN is an MPRN where each token type contains exactly one token instance.

Theorem 1

For each MRPN exists an *equivalent* SRPN and vice versa.

Theorem 2

For each SRPN exists an *equivalent* RPN and vice versa.

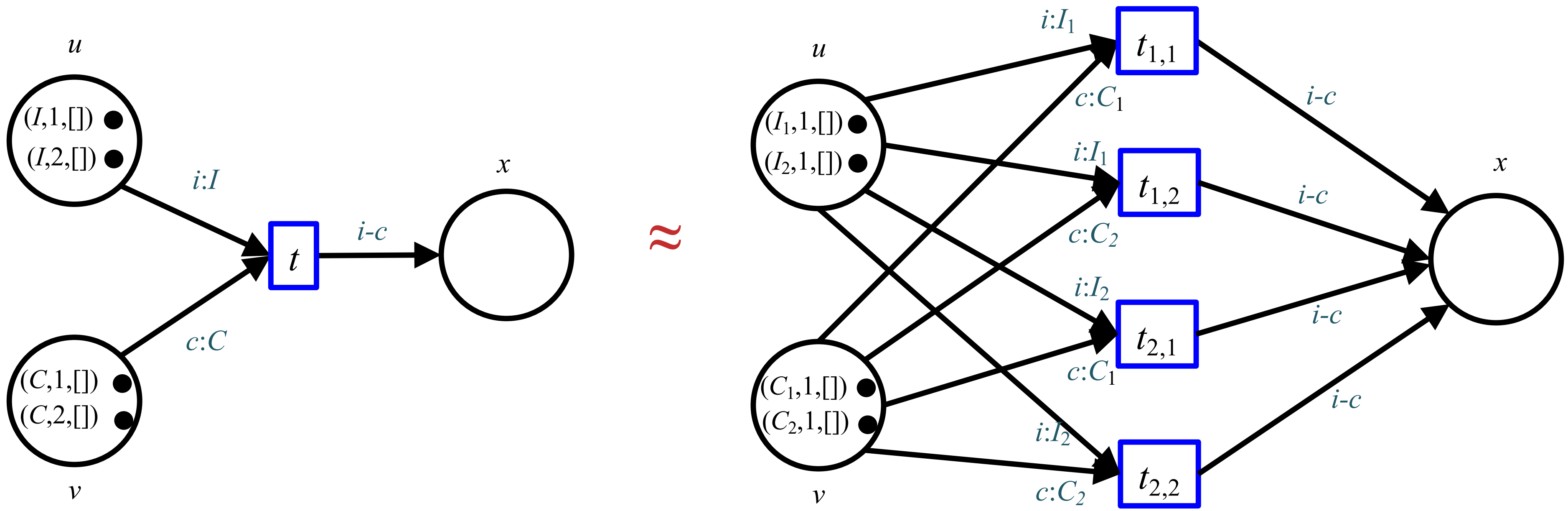
Two nets are equivalent to each other if they give rise to isomorphic LTSs.

Theorem 1 – Proof idea (1)

It is possible to construct a translation from MRPNs to SRPNs by

- creating a distinct SRPN token type for each MRPN token instance, and
- cloning transitions for each token type combination

Example:

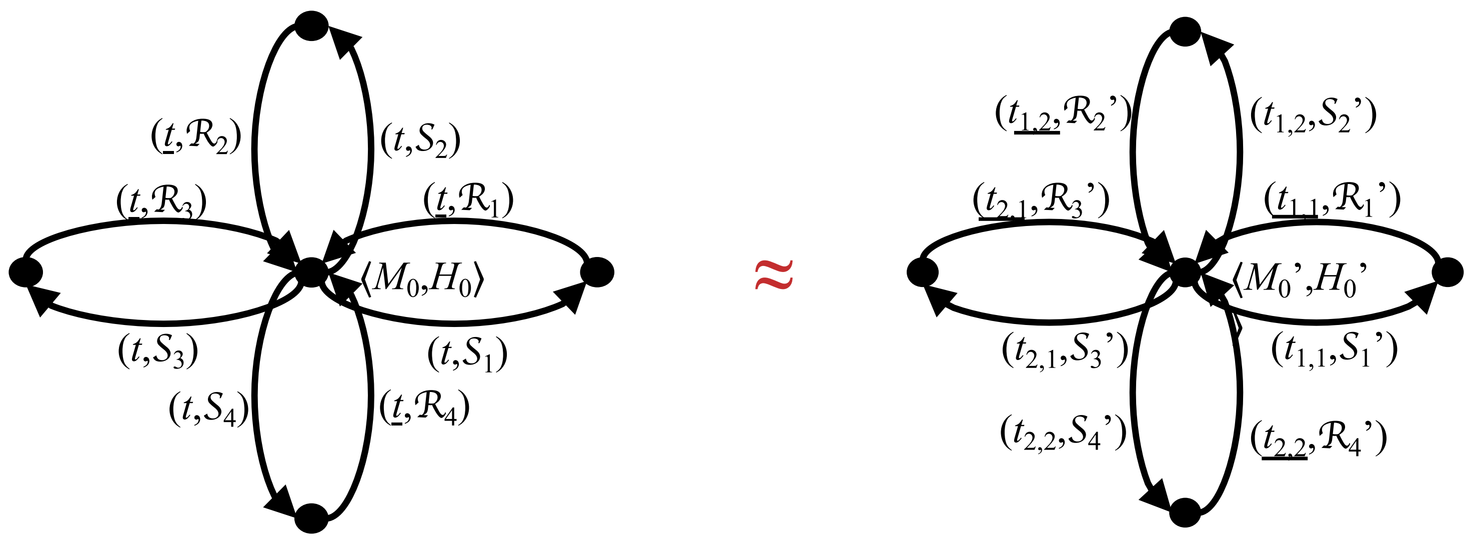


Theorem 1 – Proof idea (2)

Bijections to capture the isomorphism of the respective LTSs can be established by associating

- tokens instances of MPRNs to those of SRPNs, and
- transition occurrences of MRPNs with transitions of SRPNs

Example continued:



Conclusions

- An approach to reversing computation in Petri nets based on the concept of bonds
- RPNs with multiple tokens under the individual-token interpretation
  - Tokens of the same type can fire any eligible transition when going forward, but only the transitions they have fired when going backward
  - A transition occurrence can reverse in causal order if it was the last transition executed by all the tokens it has involved
  - All information needed for reversal is captured locally as histories in tokens - no need for global control
- Multiple tokens do not increase the expressiveness of the model
- Backtracking and out-of-causal-order reversibility also considered [Psara 2021]
- RPNs with multiple tokens under the collective-token interpretation [PP-TCS 2022]
  - local, out-of-causal-order type of reversibility

On-going and future work

- Relationship between RPNs and Colored Petri Nets [BGMPPP 2018, BGMPPP 2022]
- Tool development
  - Prototype simulator
  - Translation into ASP [DKPP 2020]
  - Model checking and analysis techniques
- Applications
  - Applications from biochemistry (the autoprotolysis of water, the ERK signaling pathway, the ammonium potassium pump) [KACPPU 2020, PP-JLAMP 2022, PP-TCS 2022]
  - Distributed algorithm for antenna selection in MIMO systems [PPS 2020, SPP 2020]
  - Transaction-processing systems [Psara 2021]

Thank you! Questions?