Realisability of Chemical Pathways

Jakob L. Andersen[1] Sissel Banke[1] Rolf Fagerberg[1]
Christoph Flamm[3] Daniel Merkle[2,1] Peter F. Stadler[3,4]

[1]Department of Mathematics and Computer Science,
University of Southern Denmark, Denmark

[2]Faculty of Technology, Bielefeld University, Germany

[3]Institute for Theoretical Chemistry, University of Vienna, Austria

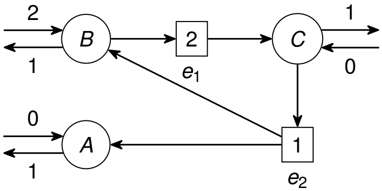
[4]Bioinformatics Group, Department of Computer Science,
University of Leipzig, Germany

TBI Winterseminar 2024

novo nordisk foundation
INDEPENDENT RESEARCH FUND DENMARK
SDU

Goal

Given a pathway, which is a steady state solution with integer constraints, we want to compute the sequence of reactions. This can later be used for atom tracing.

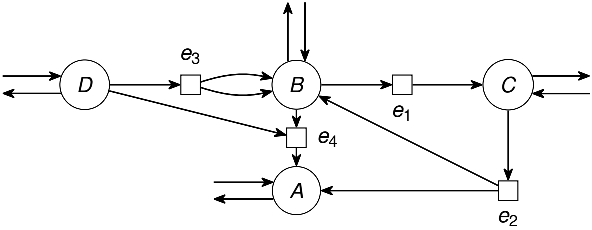


Preliminaries — Extended Hypergraph

We model chemical reaction networks (CRNs) as directed hypergraphs

- Vertices correspond to molecules.
- Hyperedges correspond to reactions.

We extend the directed hypergraph such that each vertex has input and output channels (Andersen et al. 2019).

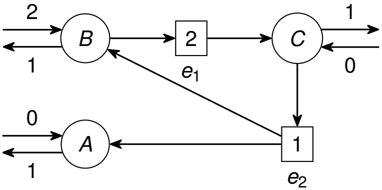


Preliminaries — Pathways as Integer Hyperflows

Pathways in CRNs can be modelled by integer hyperflows on the extended hypergraph (Andersen et al. 2019).

- The flow must uphold the *flow conservation constraint*.
- The flow specifies the number of times the reactions have to be used in the pathway.
- Gives a mechanistic understanding of pathway.

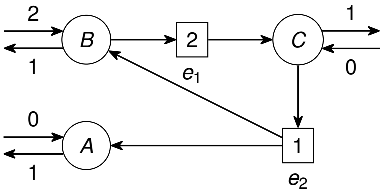
It allows one to specify input and output compounds and pathways that fulfills this specification are enumerated.

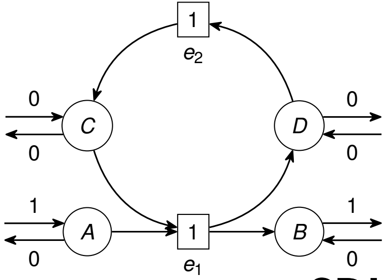


Extension of Integer Hyperflows

We want to extend the integer hyperflow framework, such that we can:

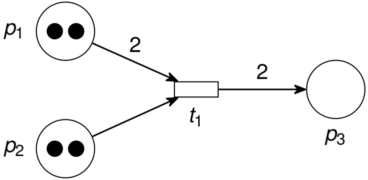
- computationally check if there exists a sequence of reactions that "realises" a pathway.
- distinguish between pathways that can be "realised" and those that cannot.
- compute the sequence of the reactions that realises the pathway.

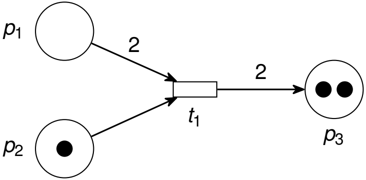




Petri Nets

- Places correspond to vertices/compounds.
- Transitions correspond to hyperedges/reactions.
- Tokens specify amount available of a compound.
- A marking specifies the amount of tokens on each place.



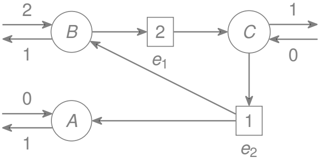


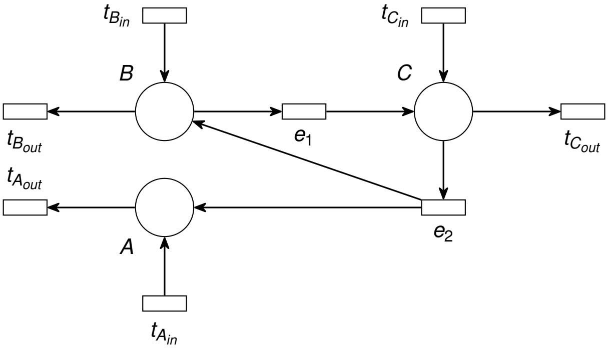
Is the Flow Realisable?

To answer the question, we will:

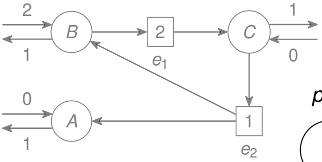
- Remodel the integer hyperflow as a Petri net.
- Computationally check (with the Petri net tool LoLA (Schmidt 2000)) if the Petri net can reach a “target” marking, making it realisable.
  - If so, a “realisability certificate” is computed.

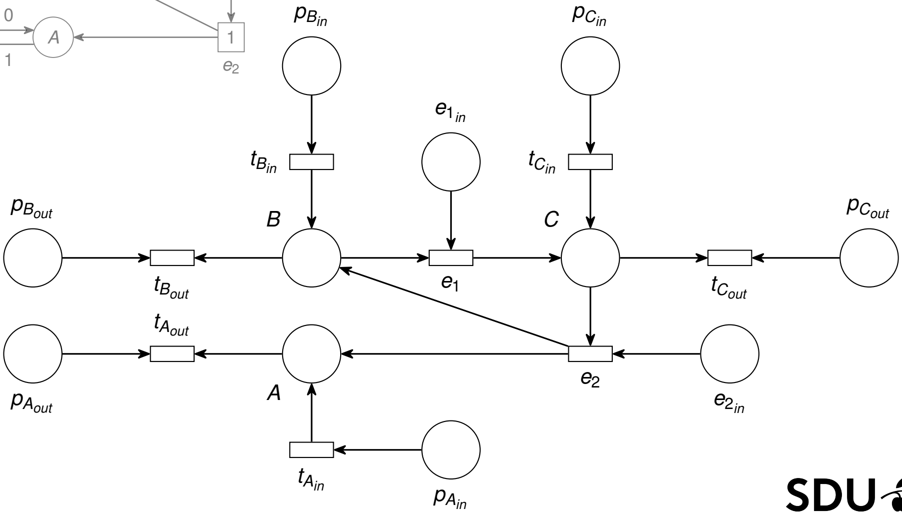
Convert the Integer Hyperflow to a Petri net



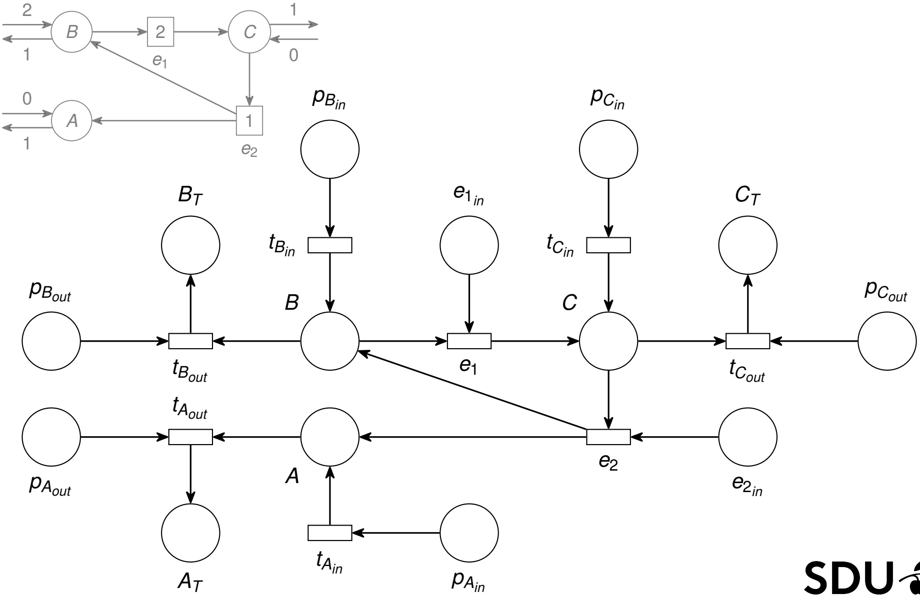


Convert the Integer Hyperflow to a Petri net

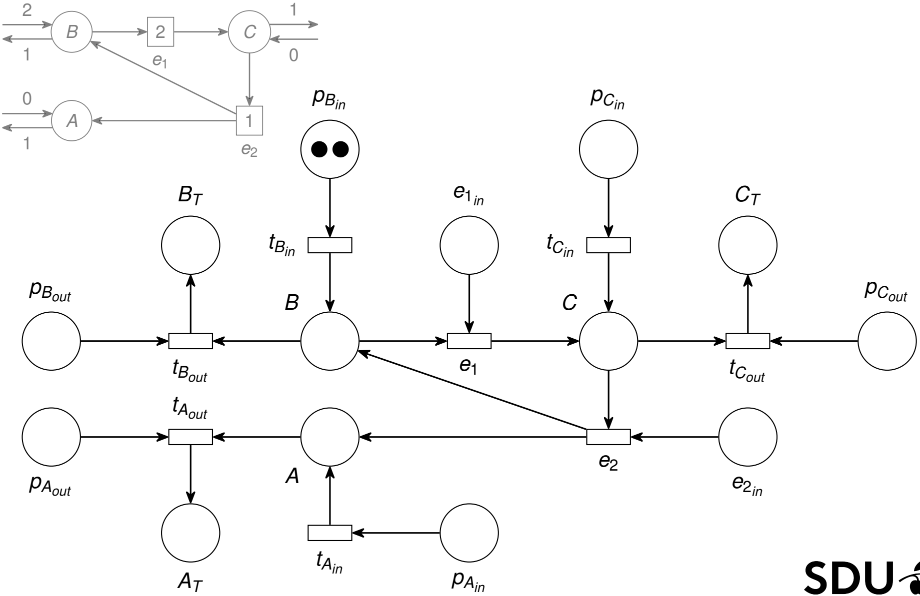




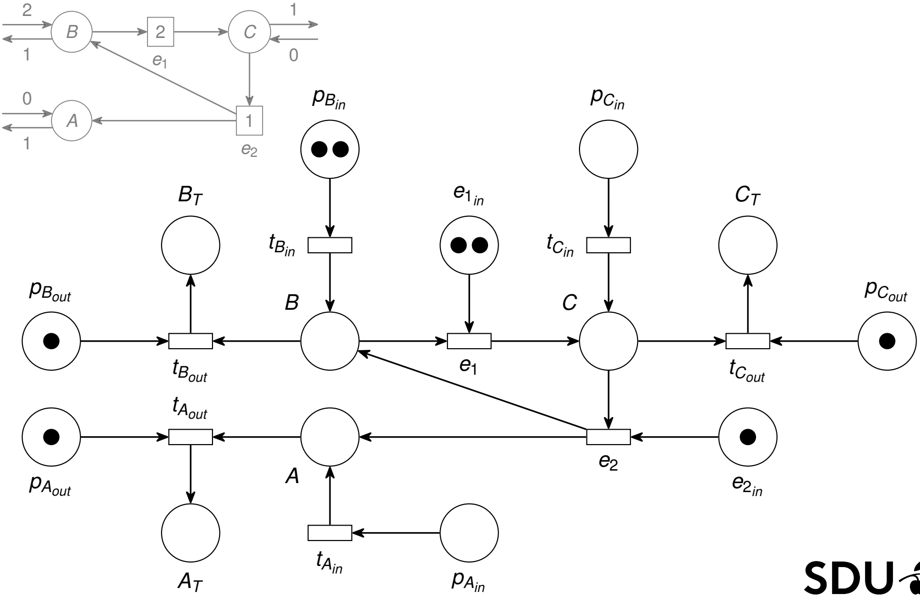
Convert the Integer Hyperflow to a Petri net



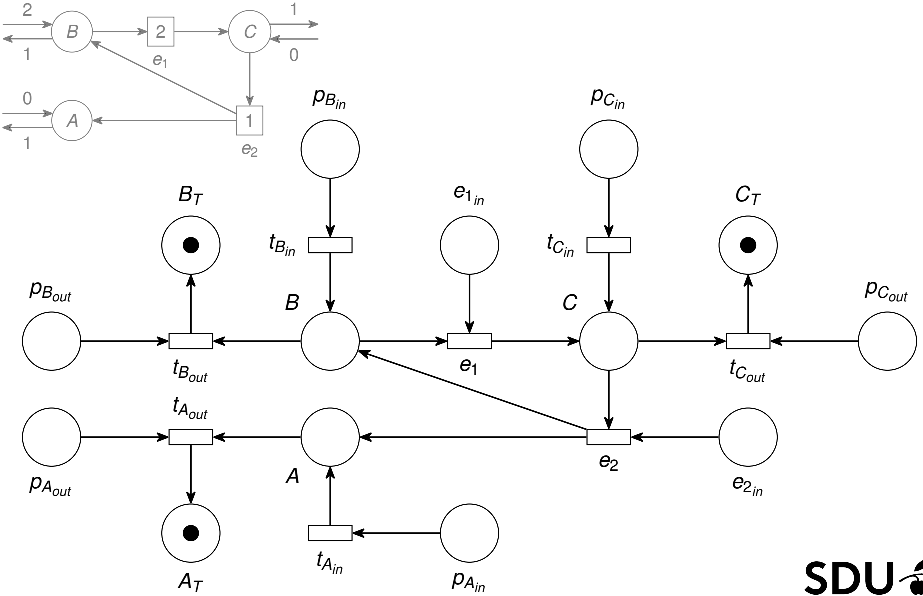
Convert the Integer Hyperflow to a Petri net



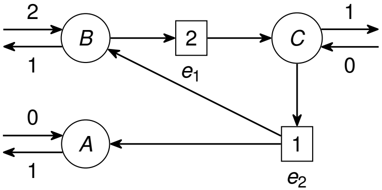
Convert the Integer Hyperflow to a Petri net

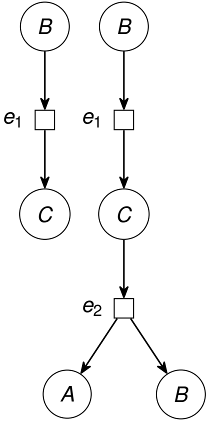


Convert the Integer Hyperflow to a Petri net



A Realisability Certificate





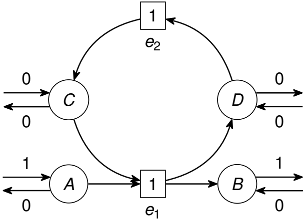
Some flows are not realisable. We want to understand why, so let us categorise them by how the can be made realisable. For that we propose two possibilities:

- borrowing one (or several) compounds — borrow-realisable flows.
- scaling the flow by some integer — scaled-realisable flows.

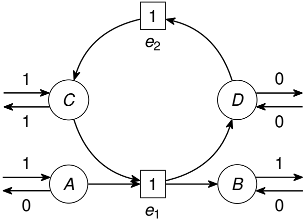
SDU

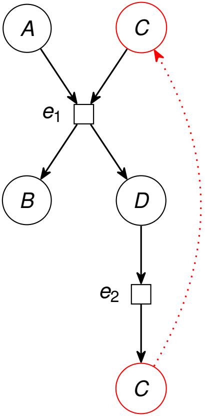
Borrow-Realisable Flows

This flow is not realisable as is.



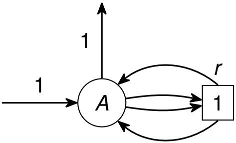
But if we borrow C it becomes realisable.



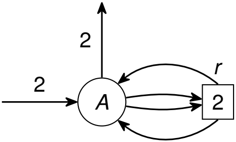


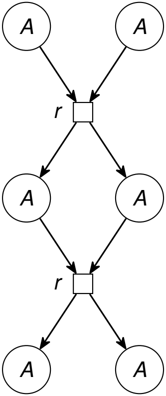
Scaled-Realisable Flows

This flow is not realisable as is.



But if we multiply the flow by 2 it becomes realisable.





Open Question on Scaled-Realisable Flows

Are all scaled-realisable flows "monotone" scaled-realisable?

Definition (Monotone Scaled-Realisable)

A flow f is monotone scaled-realisable iff it is scaled-realisable for all integers $j \geq k$, where k is the smallest factor for which it is scaled-realisable.

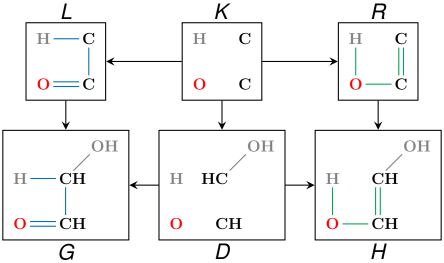
We believe so, but so far no proof exists (and neither does a counterexample).

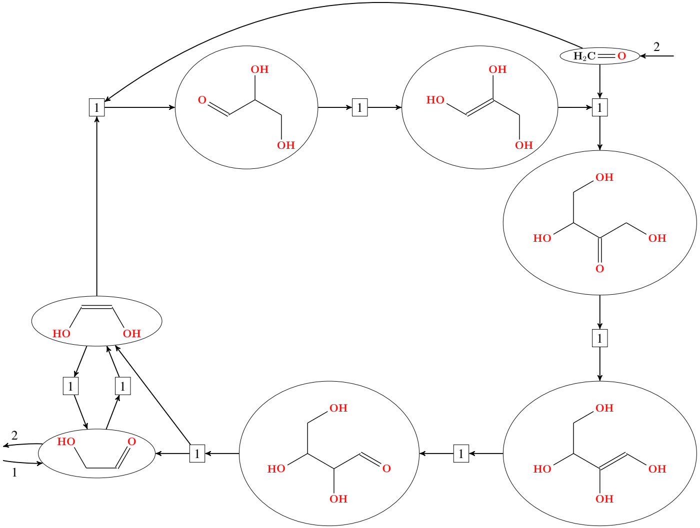
Thanks for your attention!

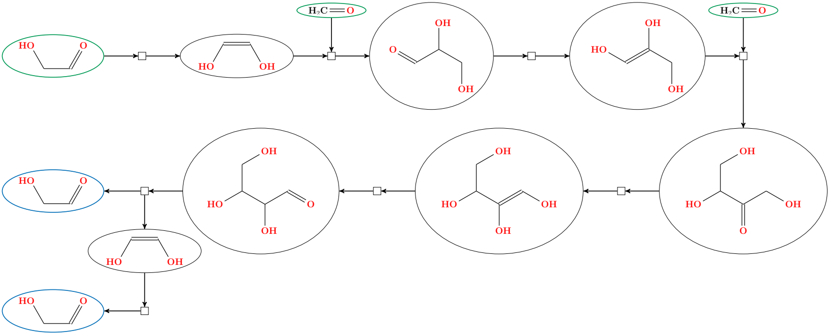
References I

- Andersen, Jakob L. et al. (2019). "Chemical Transformation Motifs — Modelling Pathways as Integer Hyperflows". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.2, pp. 510–523.
- Schmidt, Karsten (2000). "LoLA A Low Level Analyser". English. In: *Application and Theory of Petri Nets 2000*. Ed. by Mogens Nielsen and Dan Simpson. Vol. 1825. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 465–474. ISBN: 978-3-540-67693-5.

SDU


L
K
R
H
C
O
C
OH
CH
HC
G
D
H


$H_2C{=}O$
2
1
2
1
OH
O
OH
HO
OH
OH
HO
OH
O
OH
HO
OH
OH
OH
HO
O
OH
HO
OH
HO
O


HO
O
HO
OH
$H_2C=O$
OH
O
OH
HO
OH
OH
$H_2C=O$
HO
O
OH
HO
O
OH
HO
OH
OH
HO
OH
OH
HO
OH
O
HO
OH
HO
O