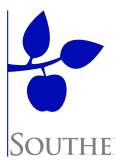
MedØlDatschgerl and Beyond

Jakob Lykke Andersen, Christoph Flamm,
Daniel Merkle, Peter F. Stadler

Department of Mathematics and Computer Science
University of Southern Denmark

Bled, February 2012



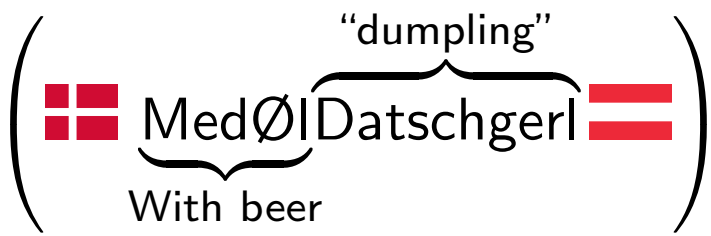
University of Southern Denmark

MedØlDatschgerl and Beyond

Jakob Lykke Andersen, Christoph Flamm,
Daniel Merkle, Peter F. Stadler

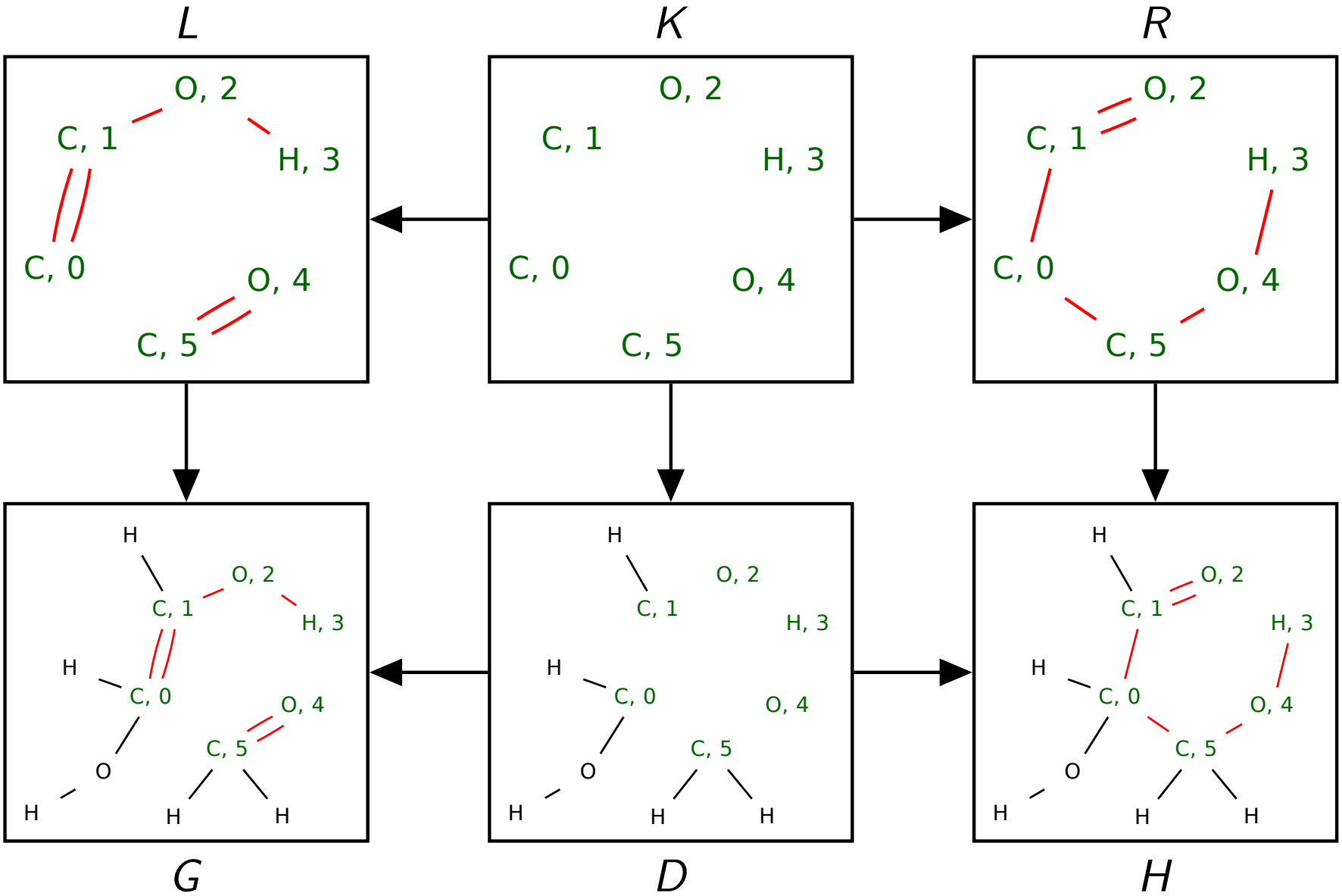
Department of Mathematics and Computer Science
University of Southern Denmark

Bled, February 2012



University of Southern Denmark

Graph Transformation - Double Pushout Approach

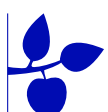


Graph Grammars

Grammar: $\mathcal{H} = (\mathcal{G}, \mathcal{R})$, starting graphs and transformation rules

Example: Formose

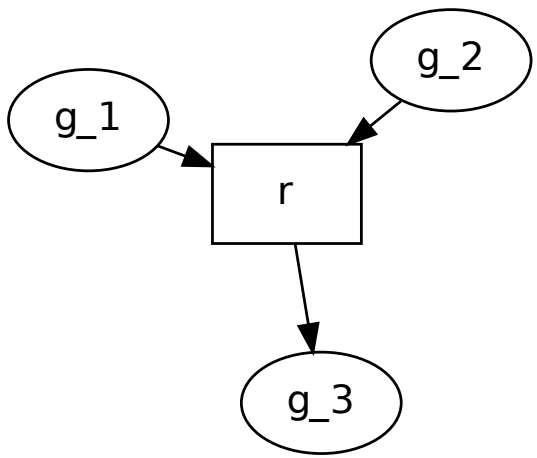
- Starting graphs:
  - g_0 formaldehyde
  - g_1 glycolaldehyde
- Transformation rules:
  - r_0 keto-enol-tautomerism, one direction
  - r_1 keto-enol-tautomerism, the other direction
  - r_2 aldol addition, one direction
  - r_3 aldol addition, the other direction



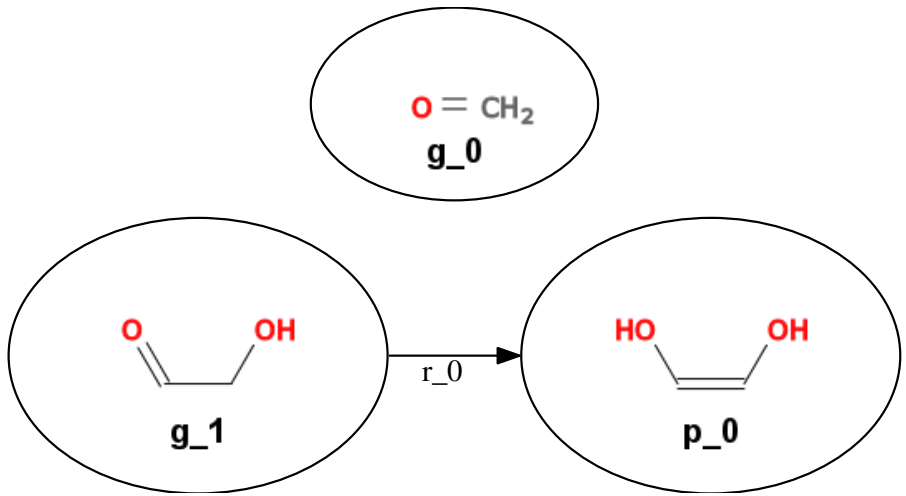
Derivation Graphs (Reaction Networks)

Input: a graph grammar (e.g., Formose)
Output: a directed hypergraph of (all) graph derivations

Visualization: $\{g_1, g_2\} \overset{r}{\Rightarrow} g_3$ is represented as

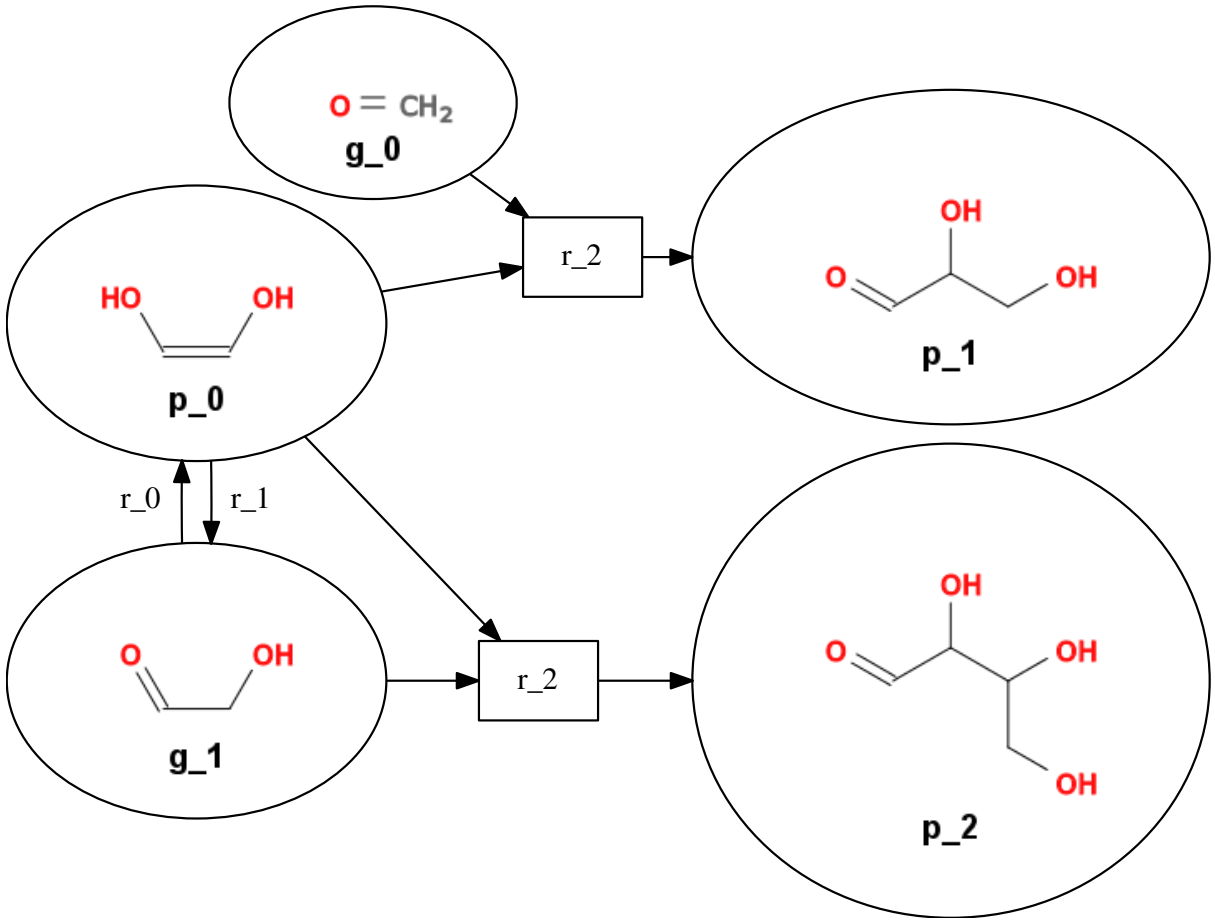


Reaction Network of Formose



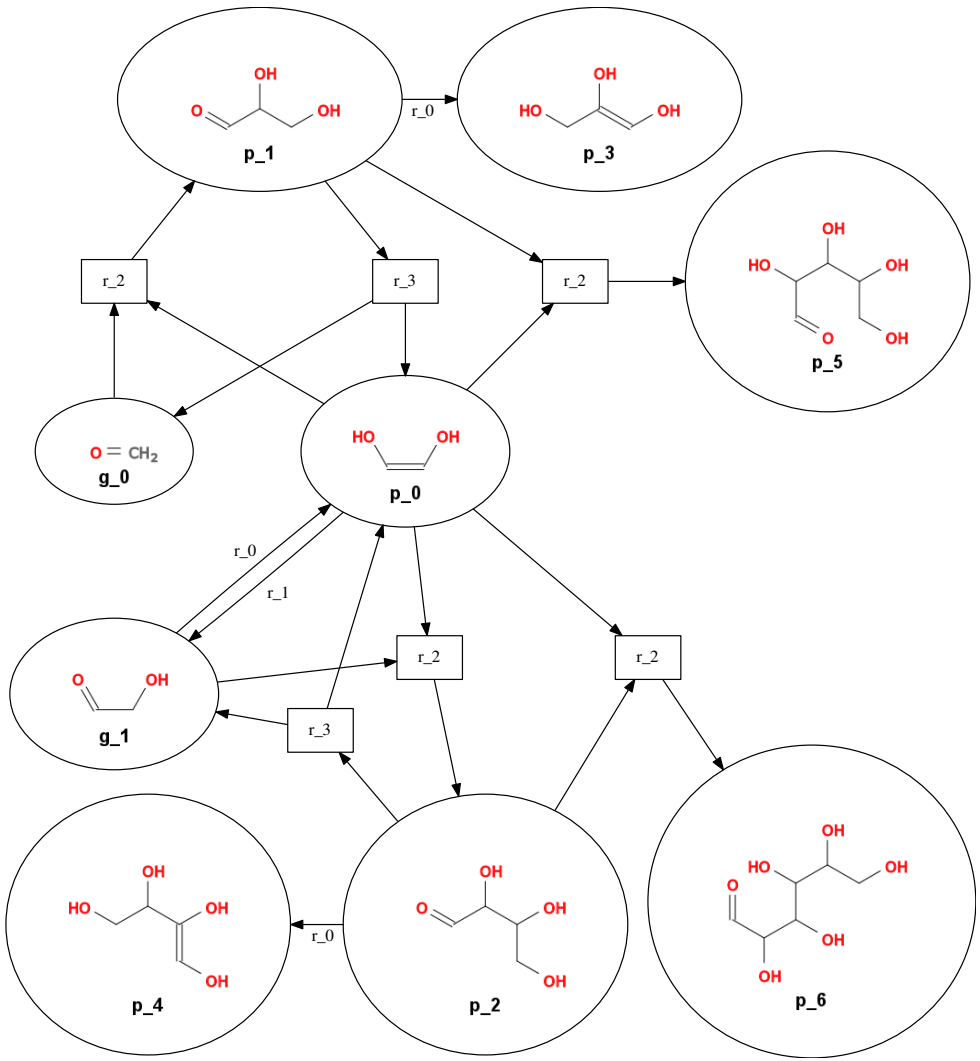
Generation 1

Reaction Network of Formose



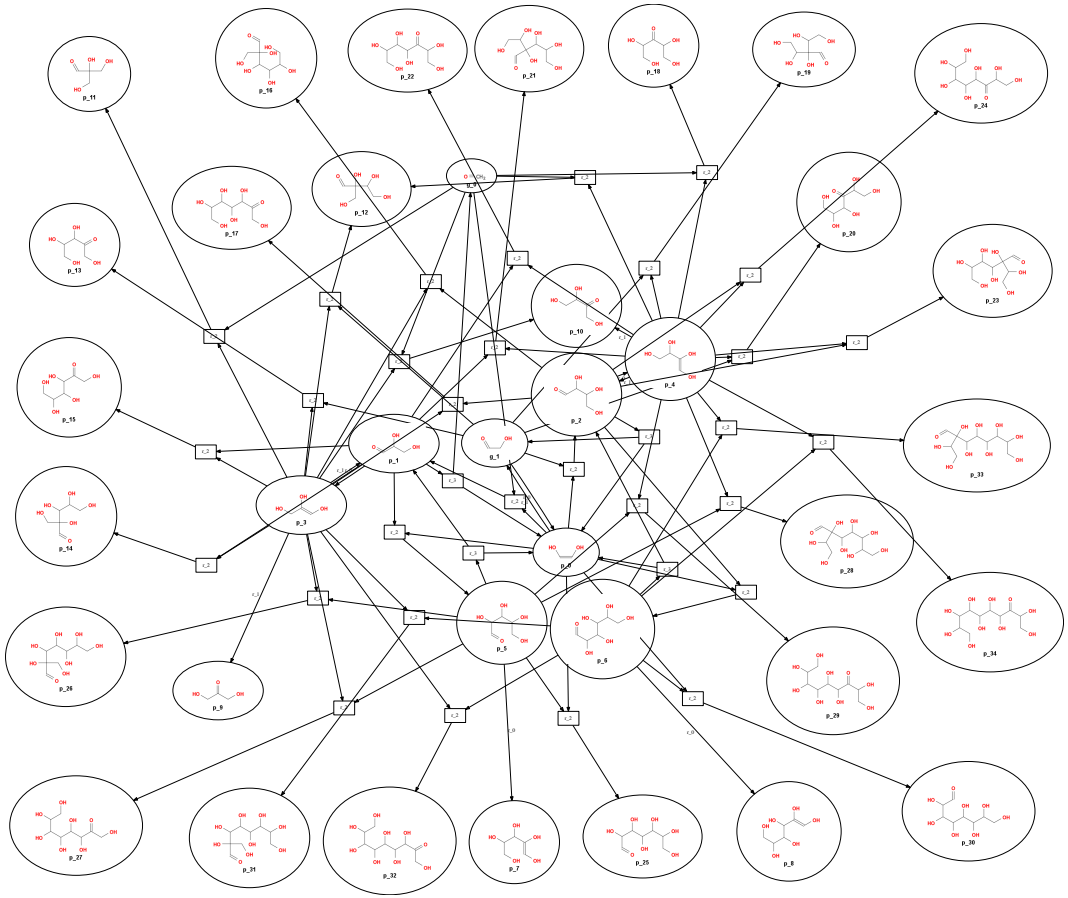
Generation 2

Reaction Network of Formose



Generation 3

Reaction Network of Formose



Generation 4

Pathways in Reaction Networks

Idea: use network flows as model for chemical pathways,
and find interesting flows

Problem: derivation graphs are hypergraphs

Solution: use integer linear programming (ILP)

Examples of interesting questions::

General pathway:

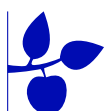
$$6 \text{ ribulose-5-phosphate} \rightarrow 5 \text{ fructose-6-phosphate}$$

Is it possible? and how?

Autocatalysis:

$$2 \text{ formaldehyde} + 1 \text{ glycolaldehyde} \rightarrow 2 \text{ glycolaldehyde}$$

Is it possible? and how?



Basis Model

Input: a derivation graph, (V, E)

Augment with input and output edges:

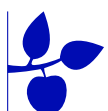
$$E_I = \{e_{in}^v = (\emptyset, \{v\}) | v \in V\} \qquad \bar{E} = E \cup E_I \cup E_O$$
$$E_O = \{e_{out}^v = (\{v\}, \emptyset) | v \in V\}$$

Variables:

$$x_e \in \mathbb{N}_0 \quad , \quad \forall e \in \bar{E}$$

Constraints:

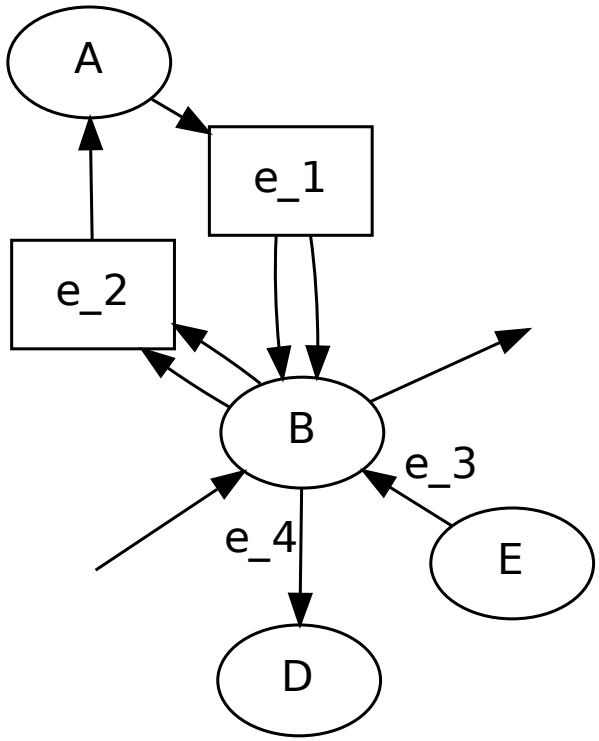
$$\sum_{e \in in(v)} x_e = \sum_{e \in out(v)} x_e \quad , \quad \forall v \in V$$



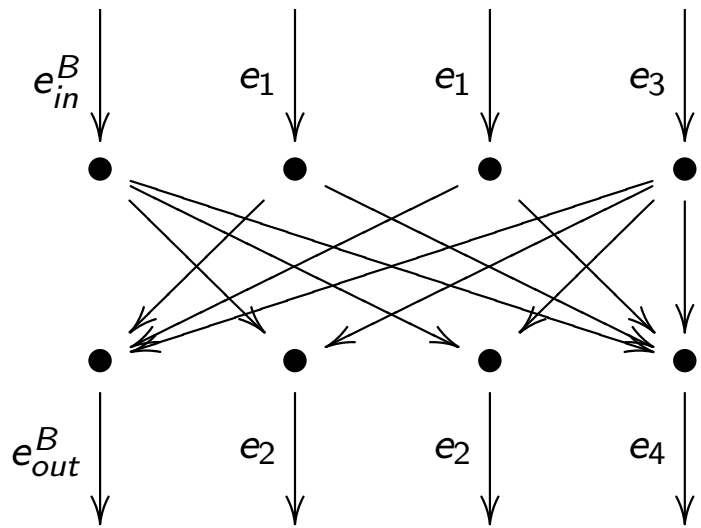
Change of the Basis Model

Replace each vertex with a bipartite graph

Part of a derivation graph



B



The extension Autocata

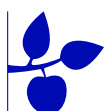
Variables:

$$Z_v \in \{0,1\} \quad , \quad \forall v \in V$$
$$Z_v^{in} \in \{0,1\} \quad , \quad \forall v \in V$$

Desired constraints:

$$Z_v = 1 \Leftrightarrow 0 < x_{in}^v < x_{out}^v$$
$$Z_v^{in} = 1 \Leftrightarrow 0 < x_{in}^v$$

$$\sum_{v \in V} Z_v \geq 1$$



The extension Autocata

Constraints:

$$Z_v^{in} \leq x_{in}^v \tag{1}$$

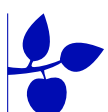
$$M \cdot Z_v^{in} \geq x_{in}^v \tag{2}$$

$$Z_v \leq x_{in}^v \tag{3}$$

$$x_{in}^v < x_{out}^v + M \cdot (1 - Z_v) \tag{4}$$

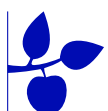
$$M \cdot Z_v \geq x_{out}^v - x_{in}^v - M \cdot (1 - Z_v^{in}) \tag{5}$$

x_{in}^v		x_{out}^v	${}^1Z_v^{in}$	${}^2Z_v^{in}$	${}^3Z_v$	${}^4Z_v$	${}^5Z_v$
0		0	0	–	0	0	–
0		42	0	–	0	–	–
42		0	–	1	–	0	–
	$=$		–	1	–	0	–
	$<$		–	1	–	–	1
	$>$		–	1	–	0	–

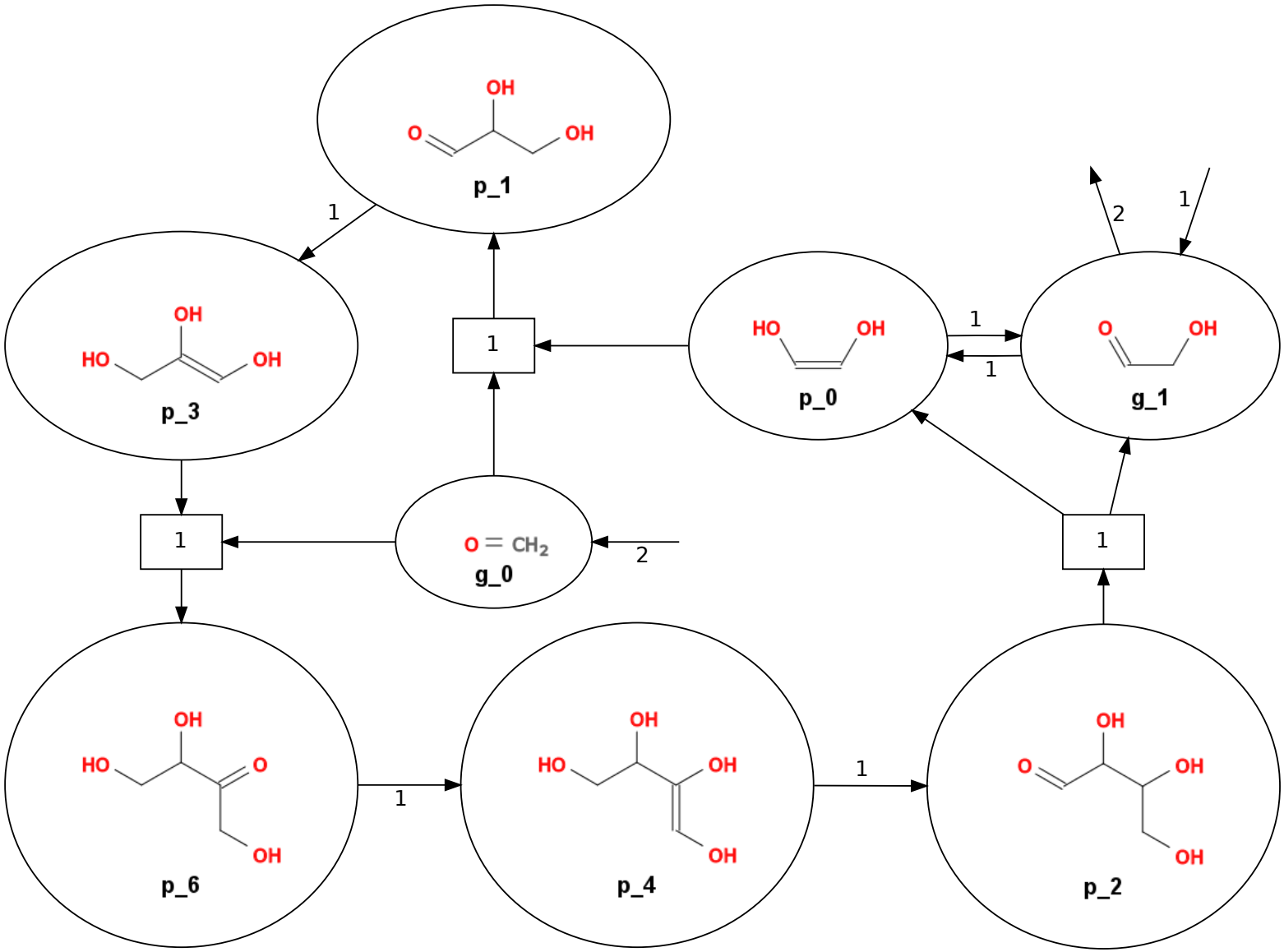


MedØlDatschgerl Code – Detection of Autocatalysis

```
load smiles    C=O
               OCC=O    ;
load r    keto_enol_forward.gml
          keto_enol_backward.gml
          aldol_addition_forward.gml
          aldol_addition_backward.gml    ;
dg size 16
autocata sources input; sinks sources;
list autocata
print dg
set DGFlow::printOnlyFiltered false
set DGFlow::printOnlyFlowLabels true
print autocata
```



Autocatalysis in Formose



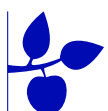
Autocatalysis in Metabolic Networks

Computational identification of obligatorily autocatalytic replicators embedded in metabolic networks
[Kun et al. Genome Biology 2008]

Networks:

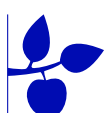
- *Escherichia coli*
- *Heliobacter pylori*
- *Staphylococcus aureus*
- *Mycobacterium tuberculosis*
- *Methanosarcina barkeri*

Runs: for each network, run the detection with each internal molecule added to the input and output sets, individually
2898 runs in total ($\approx$ 3 days with 17 computers (3 cores each))



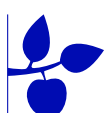
Autocatalysis in Metabolic Networks

Networks	Auto. molecules	Molecules	Percentage
Escherichia coli	226	625	36%
Heliobacter pylori	118	412	29%
Staphylococcus aureus	193	577	35%
Mycobacterium tuberculosis	255	740	34%
Methanosarcina barkeri	143	558	26%



Autocatalysis in Metabolic Networks

32 molecules, autocatalytic in all 5 networks					
13dpg	4pasp	gdp	nad	ppi	ump
23dhdp	adp	gln-L	nadh	prpp	utp
26dap-M	arg-L	gmp	nadp	pser-L	
2pg	aspsa	gtp	nadph	so3	
3pg	atp	hom-L	pep	thdp	
3php	coa	lac-L	phom	udp	



Model for Catalysis, Part 1 of 2

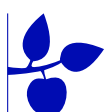
$$0 < x^v_{in} - x^v_{out} + M \cdot (1 - Z^{>}_v) \quad (1) \qquad 0 < x^v_{out} - x^v_{in} + M \cdot (1 - Z^{<}_v) \quad (3)$$

$$M \cdot Z^{>}_v \geq x^v_{in} - x^v_{out} \quad (2) \qquad M \cdot Z^{<}_v \geq x^v_{out} - x^v_{in} \quad (4)$$

$$1 - Z^0_v \leq x^v_{in} + x^v_{out} \quad (5)$$

$$M \cdot (1 - Z^0_v) \geq x^v_{in} + x^v_{out} \quad (6)$$

x^v_{in}		x^v_{out}	${}^1Z^{>}_v$	${}^2Z^{>}_v$	${}^3Z^{<}_v$	${}^4Z^{<}_v$	${}^5Z^0_v$	${}^6Z^0_v$
0		0	0	–	0	–	1	–
0		42	0	–	–	1	–	0
42		0	–	1	0	–	–	0
	$=$		0	–	0	–	–	0
	$<$		0	–	–	1	–	0
	$>$		–	1	0	–	–	0



Model for Catalysis, Part 2 of 2

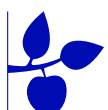
$$Z_v^c \geq 1 - Z_v^< - Z_v^> - Z_v^0 \tag{7}$$

$$0 \leq x_{in}^v - x_{out}^v + M \cdot (1 - Z_v^c) \tag{8}$$

$$0 \leq x_{out}^v - x_{in}^v + M \cdot (1 - Z_v^c) \tag{9}$$

$$Z_v^c \leq x_{in}^v + x_{out}^v \tag{10}$$

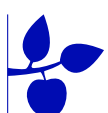
x_{in}^v	x_{out}^v	${}^7Z_v^c$	${}^8Z_v^c$	${}^9Z_v^c$	${}^{10}Z_v^c$
0	0	–	–	–	0
0	42	–	0	–	–
42	0	–	–	0	–
	$=$	1	–	–	–
	$<$	–	0	–	–
	$>$	–	–	0	–



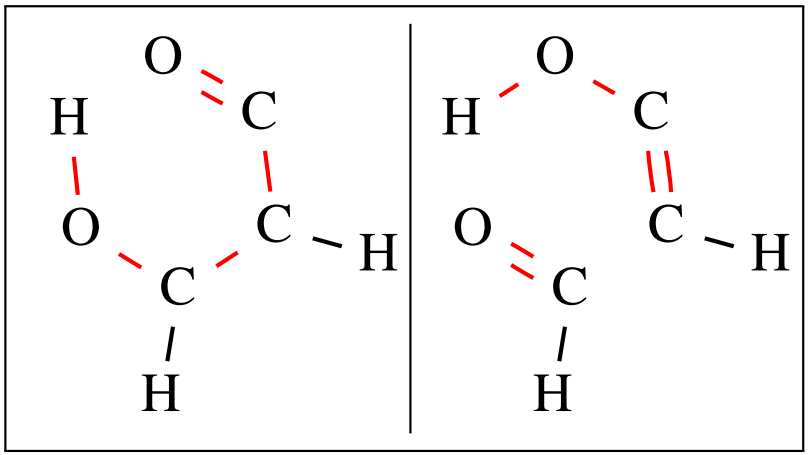
The Pentose-phosphate Pathway

$$6 \text{ ribulose-5-phosphate} \rightarrow 5 \text{ fructose-6-phosphate}$$

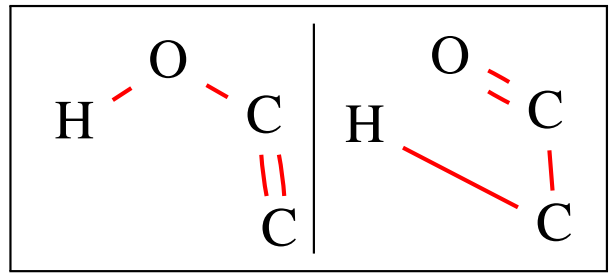
Is it possible? and how?



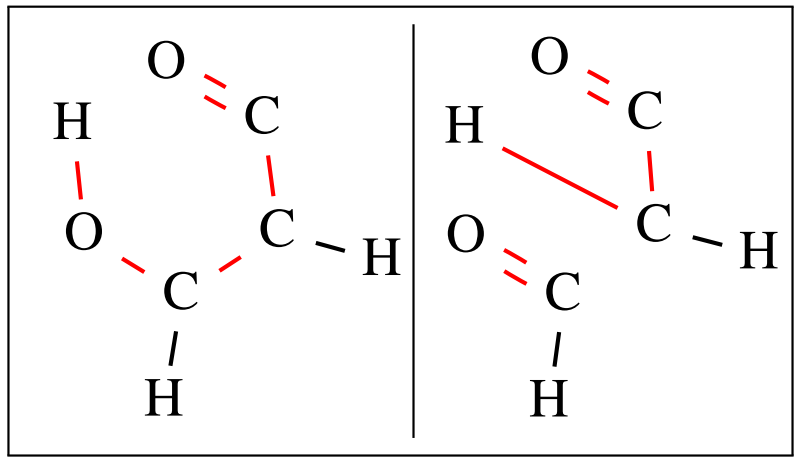
Composition of Rules



r_1: aldol-addition, backwards



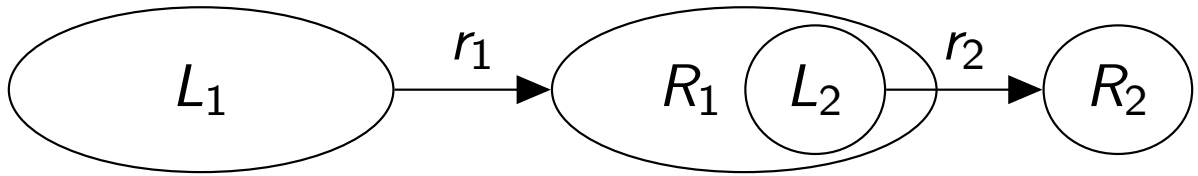
r_2: keto-enol-tautomerisation



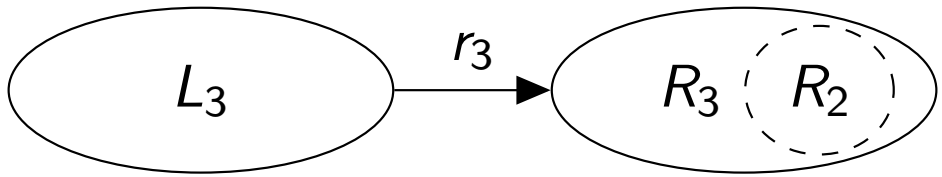
$r_2 \circ r_1$

Composition of Rules

Abstract example:

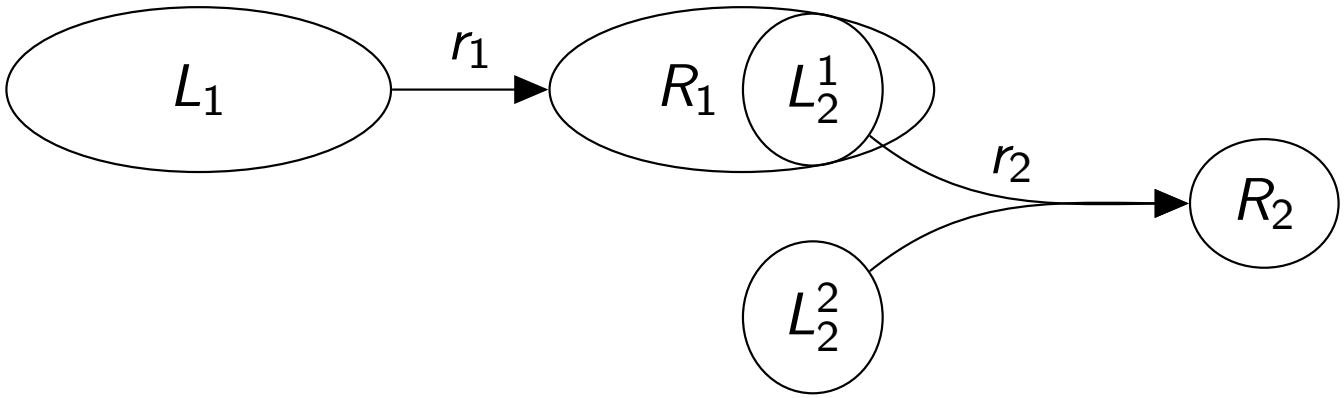


$$r_1 = (L_1, K_1, R_1),\ r_2 = (L_2, K_2, R_2)$$

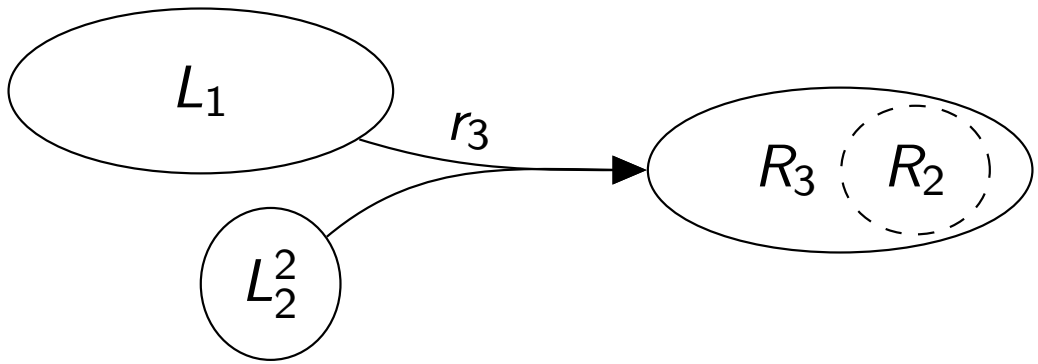


$$r_3 = r_2 \circ r_1 = (L_3, K_3, R_3)$$

Partial Composition

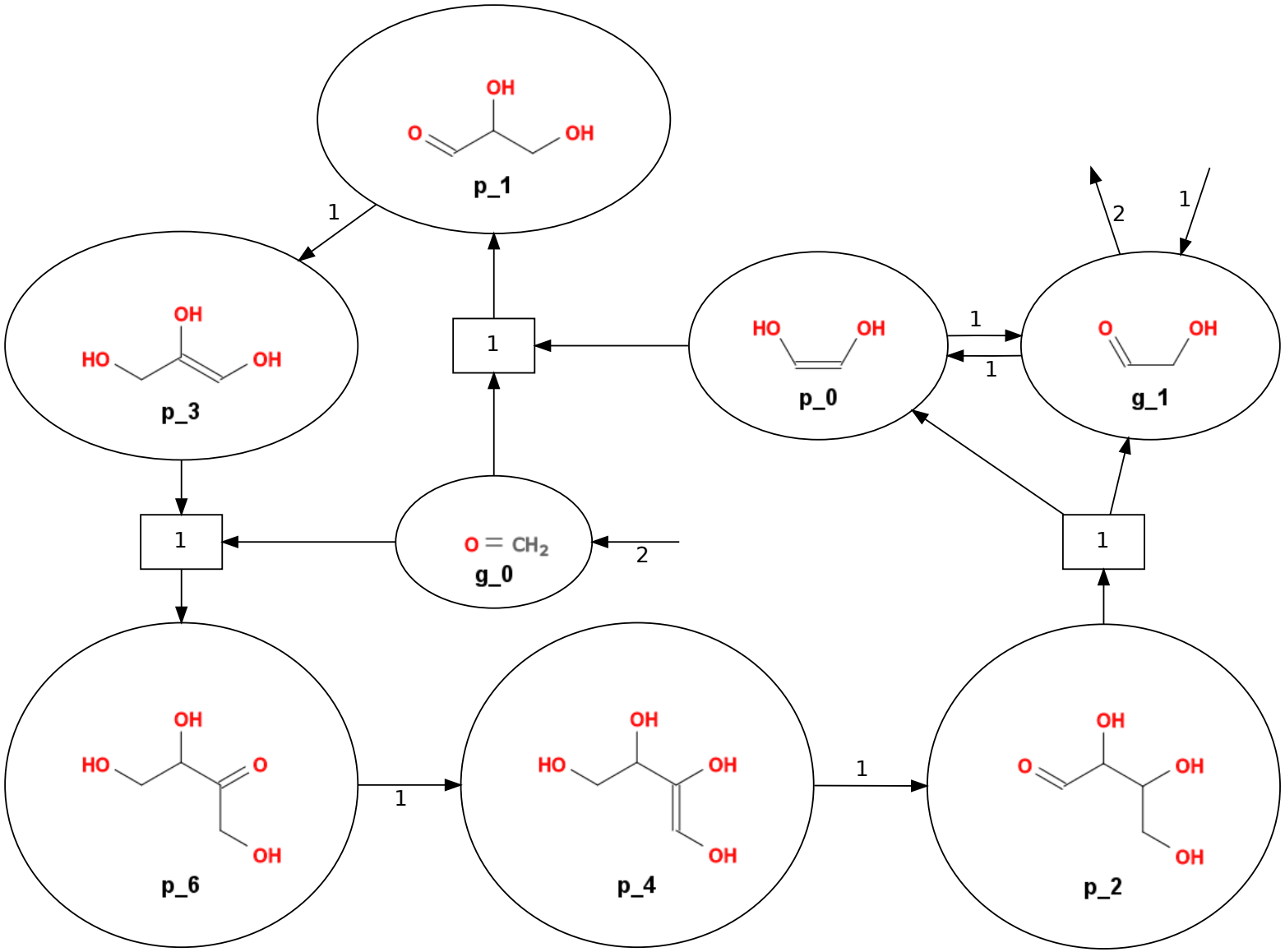


$$r_1 = (L_1, K_1, R_1),\ r_2 = (\{L_2^1, L_2^2\}, K_2, R_2)$$

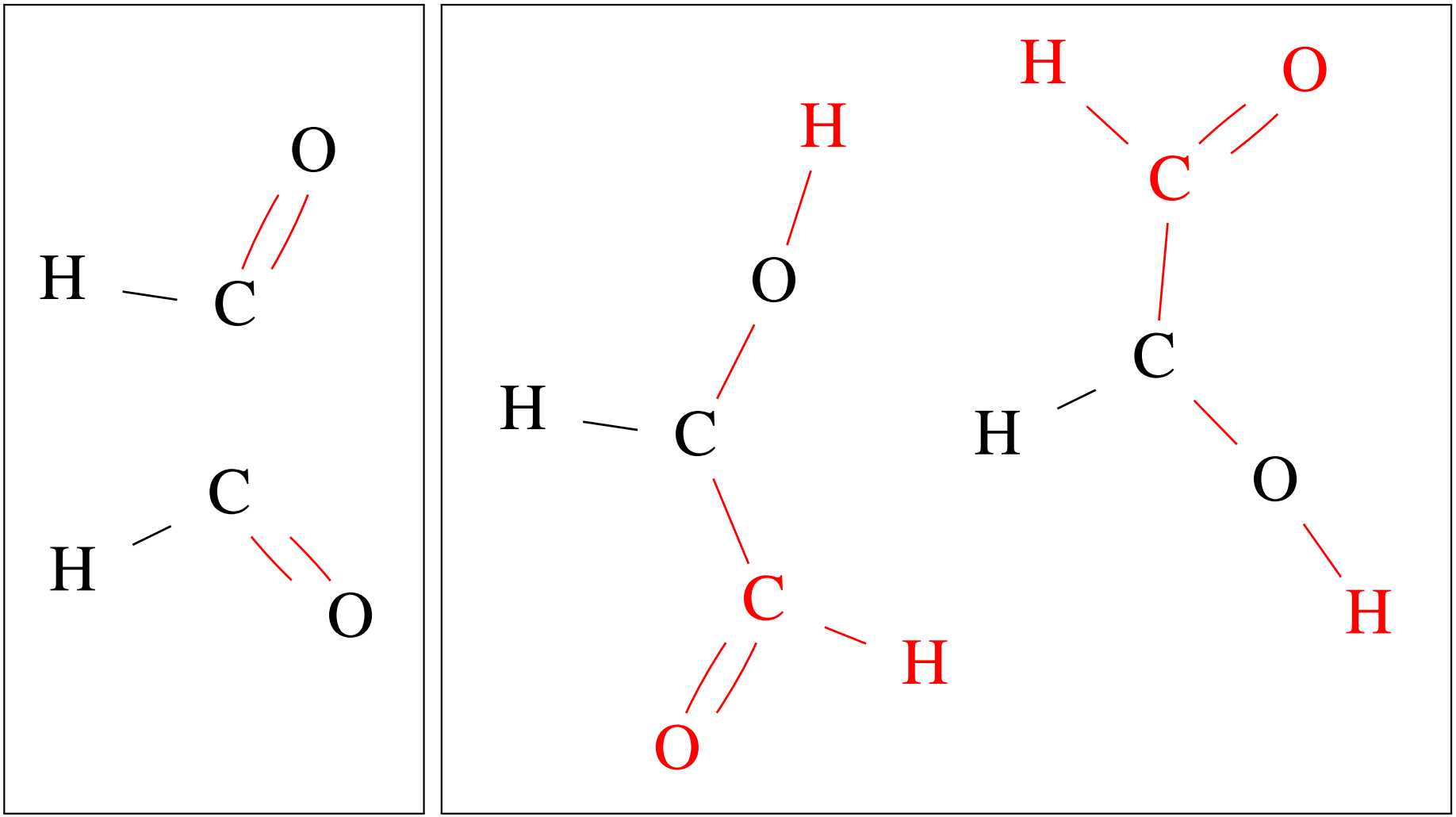


$$r_3 = r_2 \circ r_1 = (\{L_1, L_2^2\}, K_3, R_3)$$

Autocatalysis in Formose



Composition with the Formose Grammar



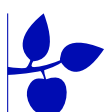
$r_1 \circ r_3 \circ r_1 \circ r_0 \circ r_2 \circ r_0 \circ r_2 \circ r_0 \circ g_1$

Strategies for Derivation Graph Generation

Current algorithm: breadth-first application of rules

`repeat({r1 r2 ... rN})`

Idea: general framework for generation



Strategies for Derivation Graph Generation

Current algorithm: breadth-first application of rules

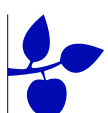
`repeat({r1 r2 ... rN})`

Idea: general framework for generation

Examples:

Limit reaction participation for molecule size:

`repeat( filter(size < 42) . {r1 r2 ... rN} )`



Strategies for Derivation Graph Generation

Current algorithm: breadth-first application of rules

`repeat({r1 r2 ... rN})`

Idea: general framework for generation

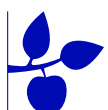
Examples:

Limit reaction participation for molecule size:

`repeat( filter(size < 42) . {r1 r2 ... rN} )`

Prioritize expansion by yield:

`repeat( limit[-10]{r1 r2 ... rN} . sort(yield) )`



Strategies for Derivation Graph Generation

Current algorithm: breadth-first application of rules

`repeat({r1 r2 ... rN})`

Idea: general framework for generation

Examples:

Limit reaction participation for molecule size:

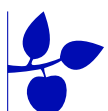
`repeat( filter(size < 42) . {r1 r2 ... rN} )`

Prioritize expansion by yield:

`repeat( limit[-10]{r1 r2 ... rN} . sort(yield) )`

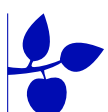
Catalan:

```
repeat( unmark . repeat[3](deleteNode) .
                                repeat(fixEdge) . mark )
```



Summary

- Generation of reaction networks from grammars
- Model for chemical pathways: flows in hypergraphs with ILP
- Strict model for autocatalysis and catalysis
- Detection of autocatalysis in metabolic networks
- Composition of transformation rules

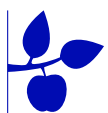


Summary

- Generation of reaction networks from grammars
- Model for chemical pathways: flows in hypergraphs with ILP
- Strict model for autocatalysis and catalysis
- Detection of autocatalysis in metabolic networks
- Composition of transformation rules

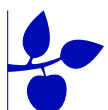
Future work:

- Pathways, catalysis, and autocatalysis in more chemistries (Citric acid cycle, Calvin cycle, HCN, ...)
- Detection of polymerization (Terpene, Polyketide, HCN, ...)
- Synthesis planning
- Strategy framework for network generation
- (Use graph grammars to solve more games)



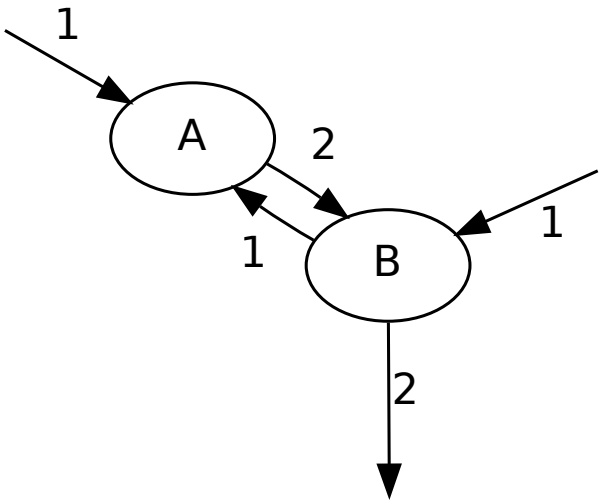
Thanks

- Daniel Merkle
- Christoph Flamm
- Peter F. Stadler
- Martin Mann

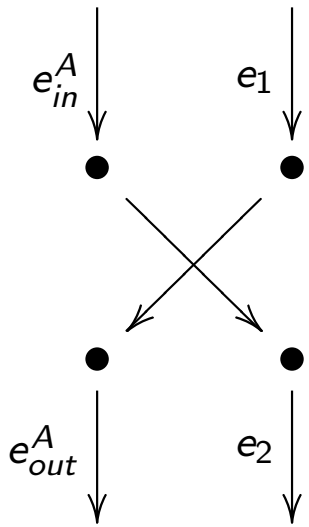


Bonus – Exampple of Problemaic Flow

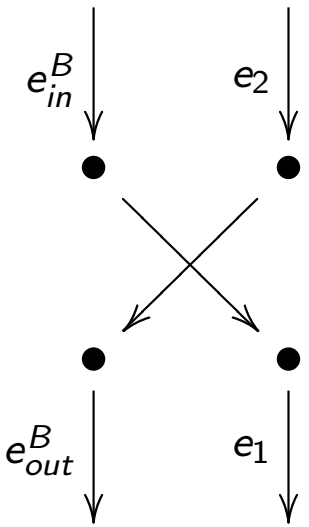
Problematic flow



A



B

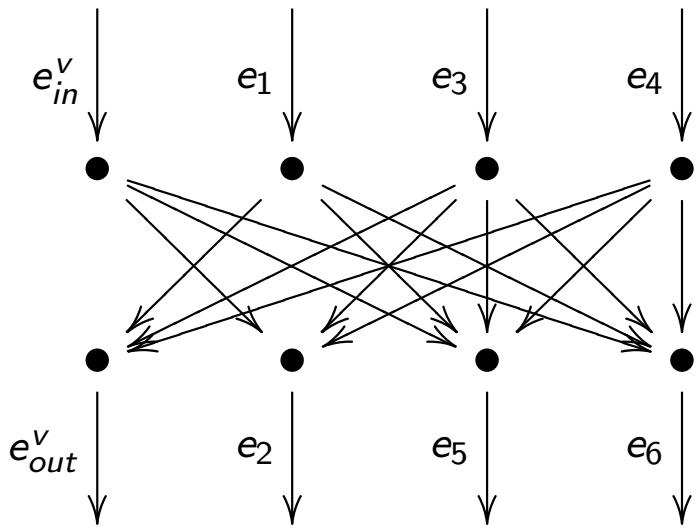


The problematic flow is no longer feasible

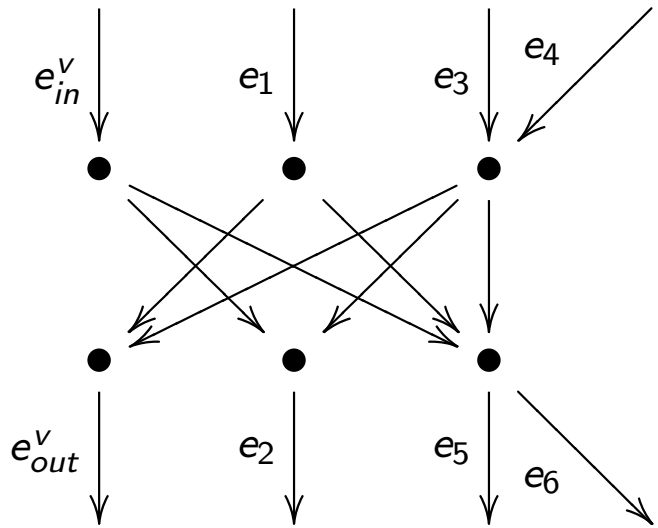
Bonus – Optimization

Example: e_1 is the inverse of e_2

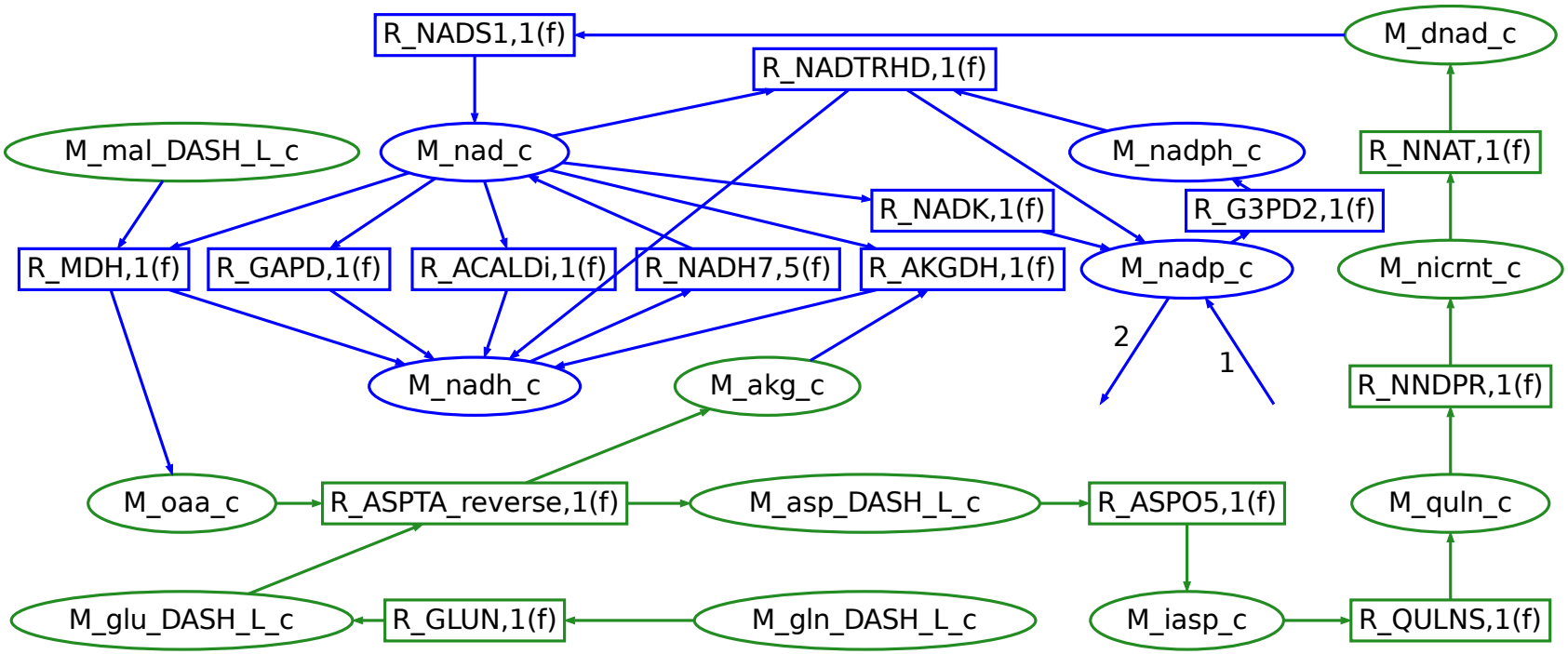
Without optimization



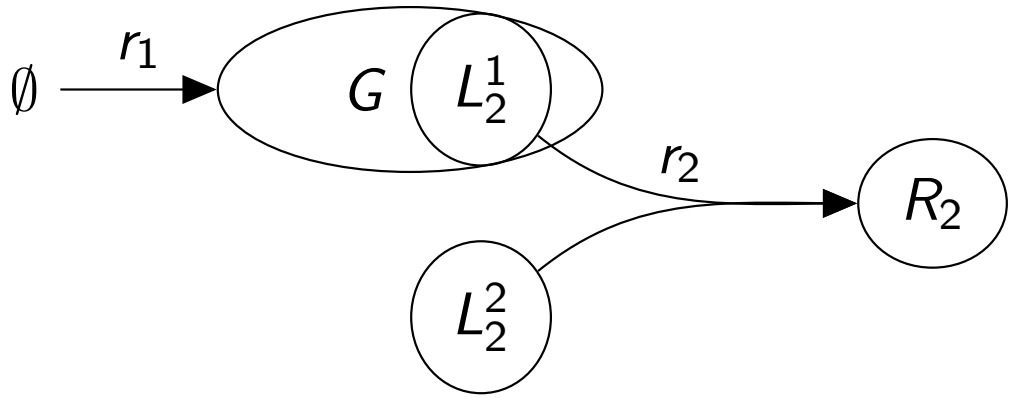
With optimization



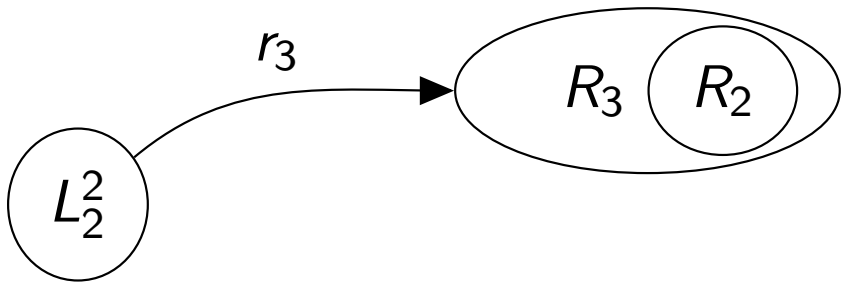
Bonus – Autocatalysis in *Escherichia coli*, NAD^+



Graph Binding



$$r_1 = (\emptyset, \emptyset, G),\ r_2 = (\{L_2^1, L_2^2\}, K_2, R_2)$$



$$r_3 = r_2 \circ r_1 = (L_2^2, K_3, R_3)$$