Computational Methods for Graph Grammar Analysis

Jakob Lykke Andersen, Daniel Merkle, Christoph Flamm

Master student at the
Department of Mathematics and Computer Science
University of Southern Denmark

Bled, February 2011

Introduction

- Graph grammars have a lot of computational power
- They can model chemical reactions
- Graph Grammar Library (Christoph Flamm and Martin Mann)
- Interesting properties: Chemical patterns

Outline

Graphs and Molecules

Undirected graphs, with labels on both nodes and edges.
Molecules: Node labels ≈ atom names, edge labels ≈ bond type

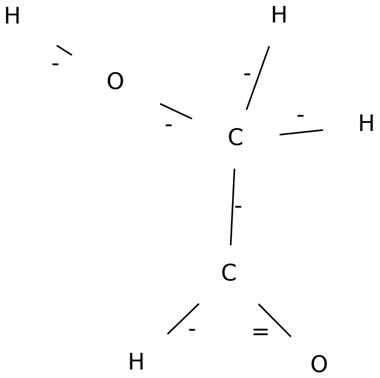


Figure: A labeled undirected graph

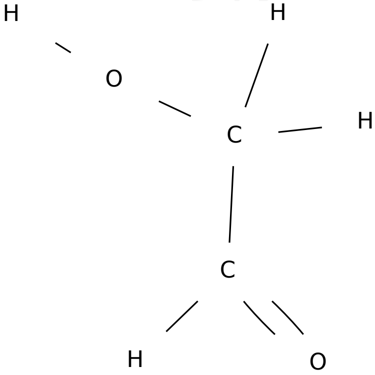


Figure: The same graph drawn like a molecule (glycolaldehyde)

Graph Grammars

- Graph grammar: A set of rules
- Rule: Left side, context, right side (all are subgraphs)
- Reactions as rules: Left ≈ broken bonds, right ≈ formed bonds, context ≈ atoms and unchanged bonds

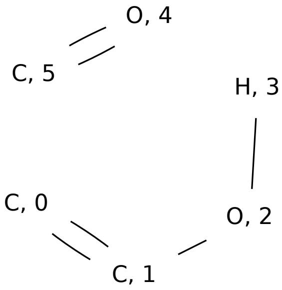


Figure: Aldol addition, left side and context

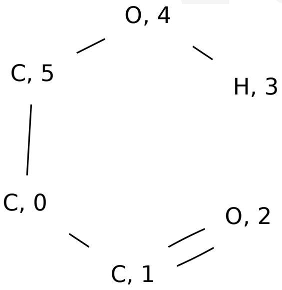


Figure: Aldol addition, right side and context

Graph Grammars

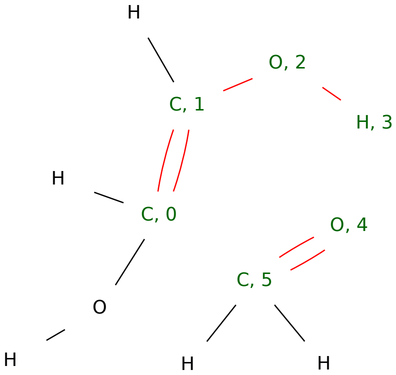


Figure: Addition of formaldehyde, before

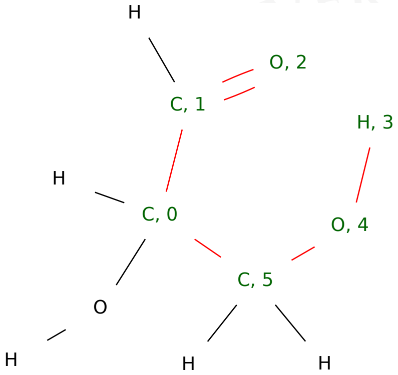


Figure: Addition of formaldehyde, after

Node Constraints

Constraint on the number of neighbours to a given node

Examples:

- At least 2 hydrogen atoms
- Exactly 0 double bounded oxygens
- At least 3 bonds (of any type)
- Exactly 1 neighbouring oxygen

Rules and Graphs for the Formose reaction

Graphs:

- Formaldehyde
- Glycolaldehyde

Rules:

- Keto-enol isomerization
- Reverse keto-enol isomerization
- Aldol-addition
- Reverse Aldol-addition

Derivation Graph

- Input: A set of graphs, a set of rules
- Output: A directed hypergraph, nodes are graphs (molecules), edges are rule applications (reations)
- Example: 2 generations of the formose reaction

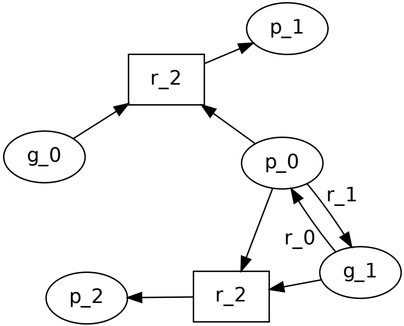


Figure: Hypergraph style

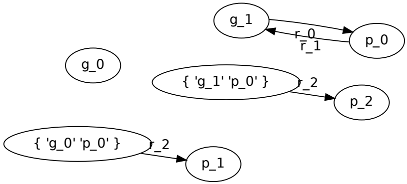


Figure: Normal graph style

Derivation Graph

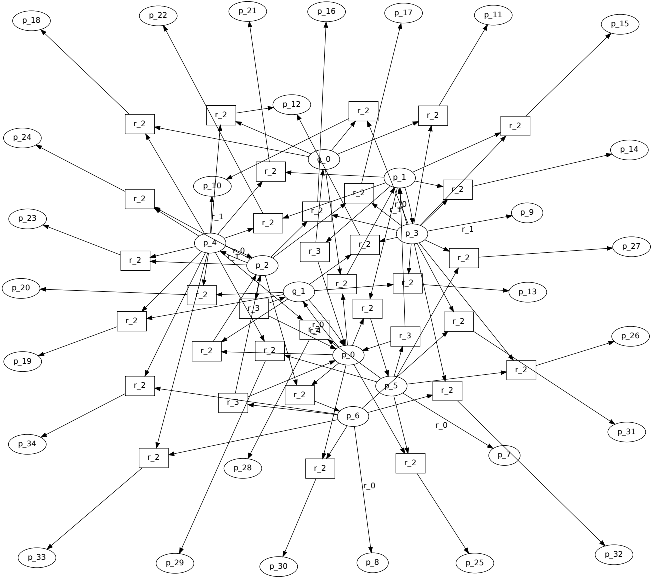


Figure: Formose with 4 generations, hypergraph style

Derivation Graph

Formose with all generations, but ≤ 43 nodes per reaction

External file due to size

Also available at http://imada.sdu.dk/~jla06/formose_large.pdf

Path Analysis

- Idea: Graphs (molecules) on a simple path in a derivation graph might have an interesting relationship
- E.g: Number of occurences of a specific subgraph

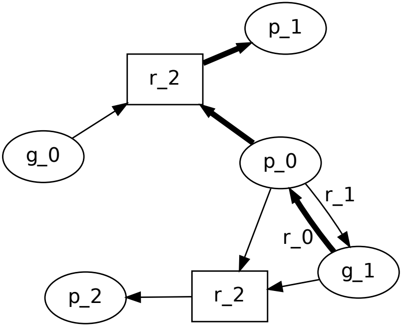


Figure: Formose with 2 generations

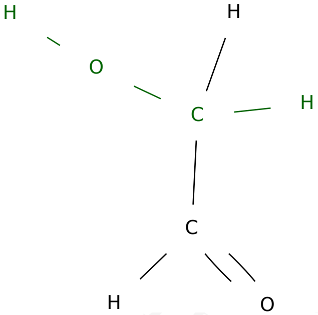


Figure: g_1, 1 match

Path Analysis

- Idea: Graphs (molecules) on a simple path in a derivation graph might have an interesting relationship
- E.g: Number of occurences of a specific subgraph

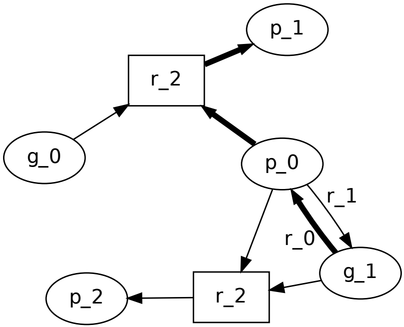


Figure: Formose with 2 generations

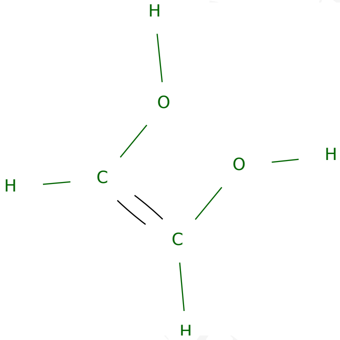


Figure: p_0, 2 matches

Path Analysis

- Idea: Graphs (molecules) on a simple path in a derivation graph might have an interesting relationship
- E.g: Number of occurences of a specific subgraph

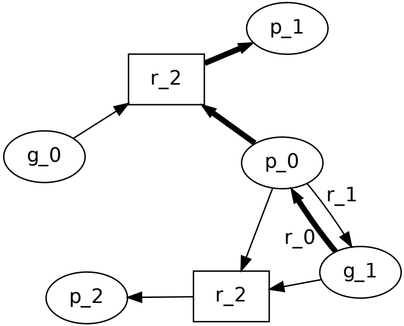


Figure: Formose with 2 generations

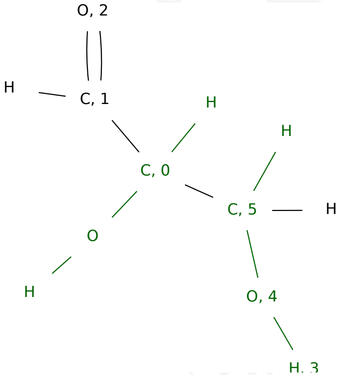


Figure: p_1, 2 matches

Flows on Derivation Graphs

- Idea: Use network flows to model interesting queries to the derivation graph
- Current implementation: Integer Linear Programming
- Example: Can 2 formaldehyde and 1 glycolaldehyde react and become only glycolaldehyde? and how?

IP Formulation

$$\text{minimize} \quad 0 \quad \text{s.t :}$$

$$
\begin{aligned}
x_3 + x_{17} + x_{18} - x_{21} - x_5 - x_{16} - x_{19} &= 0 \qquad g_0 \\
x_{23} + x_1 + x_6 - x_{22} - x_2 - x_8 &= 0 \qquad g_1 \\
x_2 + x_3 + x_6 - x_1 - x_5 - x_8 &= 0 \qquad p_0 \\
x_9 + x_{10} + x_{17} + x_{18} - x_4 - x_{13} - x_{16} - x_{19} &= 0 \qquad p_3 \\
x_{11} + x_{12} - x_7 - x_{14} &= 0 \qquad p_4 \\
x_{14} + x_{15} + x_{16} - x_{12} - x_{18} - x_{20} &= 0 \qquad p_6
\end{aligned}
$$

$$
\begin{aligned}
x_4 + x_5 - x_3 - x_9 &= 0 \quad p_1 & x_i &\geq 0, & \forall i &\in [1;20] \\
x_7 + x_8 - x_6 - x_{11} &= 0 \quad p_2 & x_{21} &= 2 & & g_0 \\
x_{13} - x_{10} &= 0 \quad p_5 & x_{22} &= 1 & & g_1 \\
x_{19} - x_{17} &= 0 \quad p_7 & x_{23} &\geq 0 & & g_1 \\
x_{20} - x_{15} &= 0 \quad p_8 & x_i &\in \mathbb{Z}, & \forall i &\in [1;23]
\end{aligned}
$$

Formose Cycle

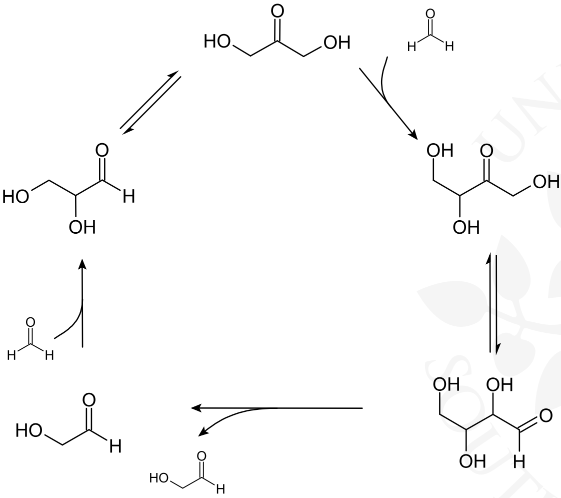
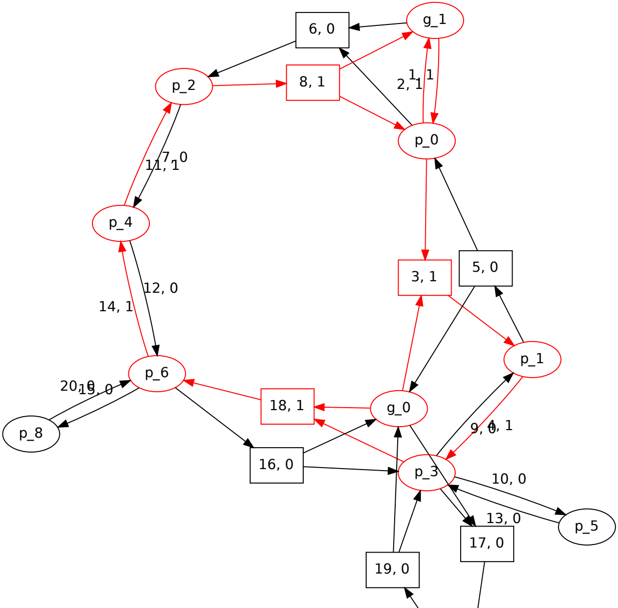


Figure: From http://de.wikipedia.org/wiki/Formose

Flows on Derivation Graphs



Summary

- GGL is used to explore graph grammars
- Derivation Graphs can represent chemical reaction networks
- Path analysis might find interesting properties
- Flows seem to capture the idea of chemical pathways
- A lot of possibilities to explore

Future Work

- Path analysis: Overlap, optimization, relationships
- Flows: More models, extra constraints, enumeration
- Grammars: Pentose-Phosphate Pathway
- ...