Learning Graph Canonicalization and Rule Inference

Kasper Halkjær Beider

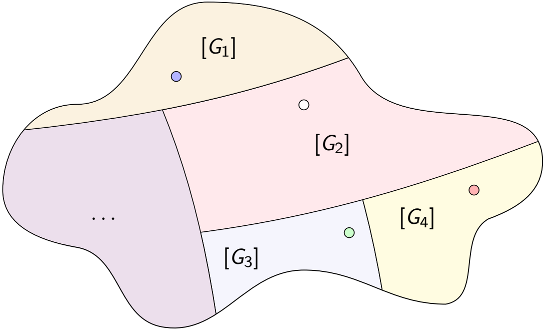
February 15, 2024

39th TBI winterseminar Bled

University of Southern Denmark (SDU)

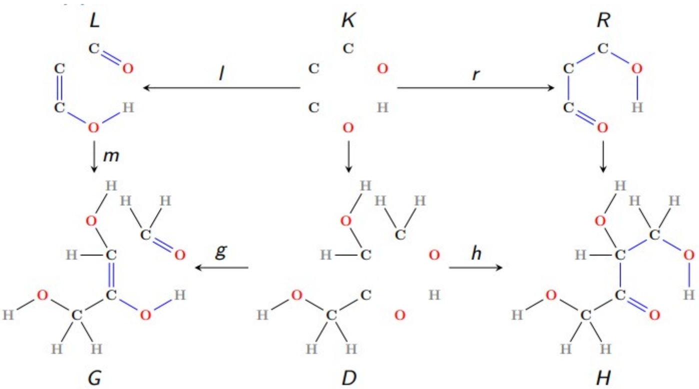
What I have worked on

- Python implementation of Graph Canonicalization



- Context inference for transformation rules
- Collaboration with Tobias Klink Lehn

Double Pushout Diagram (DPO)



Base idea

- Given n graphs $G_1, G_2, \ldots, G_n$
- Create line graphs: $L(G_1), L(G_2), \ldots, L(G_n)$
- Modular product of the line graphs: $PG = L(G_1) \times L(G_2) \times \cdots \times L(G_n)$
- Find maximal cliques in PG
- MCIS in line graphs → MCES in original graphs

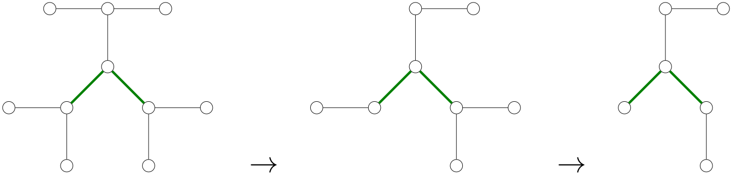
Progressive approach

- Utilizing the basic idea in an progressive manner.
- Look at two graphs at a time:

$$((((\underbrace{(G_1 \times G_2)}_{\text{result(s)}} \times G_3) \times \cdots \times G_{n-1}) \times G_n)$$

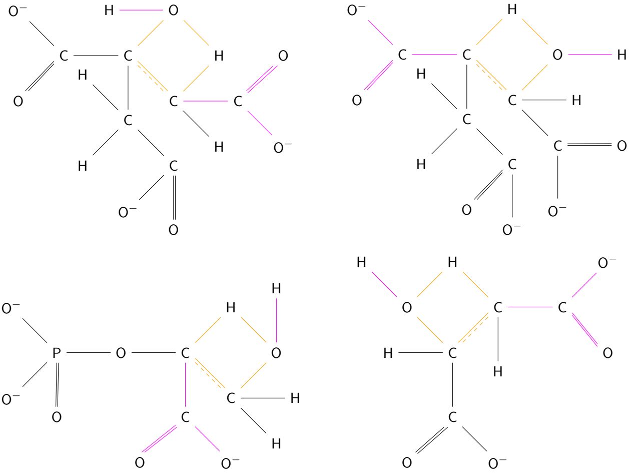
- Call recursively with each returned mapping and the next graph in line.
- Smaller product graphs, but more of them.

Progressive approach example



$$G_1 \times G_2 = E_1 \quad \rightarrow \quad E_1 \times G_3 = E_2 \quad \rightarrow \quad E_2 \times G_4$$

Example



Thank you!