Describing configurable systems

Multiple options:

- Logic gate circuits
- State machines
- Petri nets
- Process algebra
- High-level languages

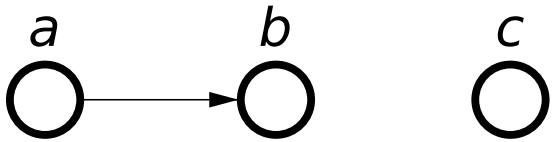
Conditional Partial Order Graphs

- Vertices represent actions
- Edges represent dependencies TODO: picture

Parametrised Graph Algebra

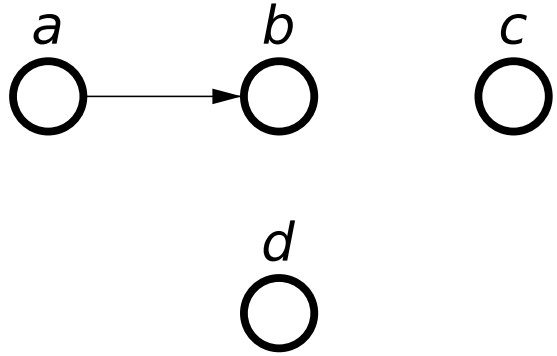
PG Algebra is a generalisation of CPOGs:

- Arbitrary set together with algebraic operations on it
- Equivalence relation satisfying certain laws.

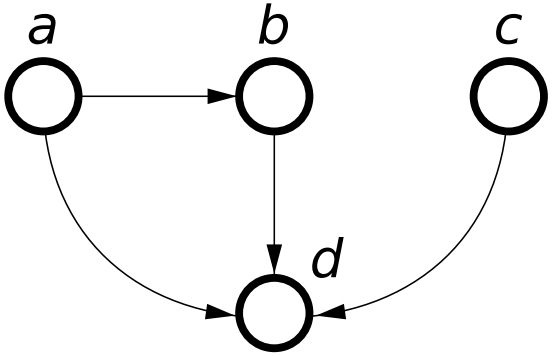


(a) Graph G_1

(b) Graph G_2



(c) Graph $G_1 + G_2$



(d) Graph $G_1 \rightarrow G_2$

Figure: Overlay and sequence example (no common vertices)

PG Software Tools

- Formula manipulations
- Conversions to/from different formalisms PG formula a CPOG
- Hardware synthesis

Formal methods

- Need a way to ensure correctness

Agda

Why Agda?

- A total functional programming language
- A proof environment based on Curry-Howard isomorphism
- Easy to learn when you know Haskell
- Newbie-friendly community