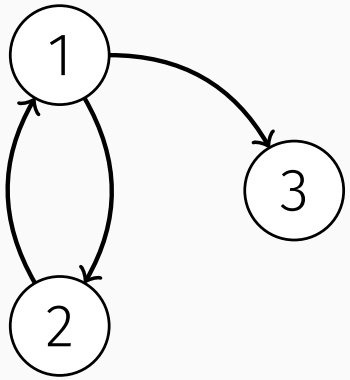
Algebraic Graphs with Class

Christoph Madlener

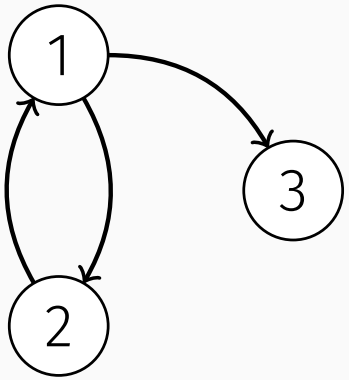
01.06.2021

Introduction



G_1

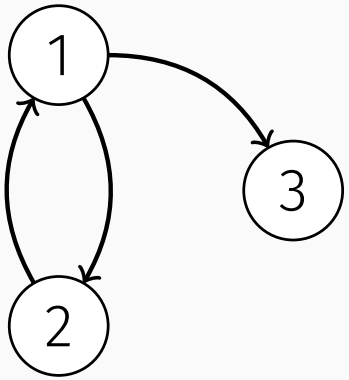
Introduction



G_1

$G = (V, E)$ s.t. $E \subseteq V \times V$

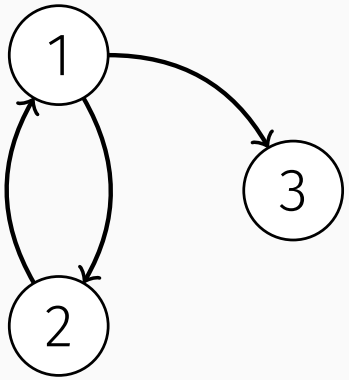
Introduction



$G = (V, E)$ s.t. $E \subseteq V \times V$

$$G_1 = \Big(\{1, 2, 3\}, \{(1, 2), (1, 3), (2, 1)\}\Big)$$

Introduction



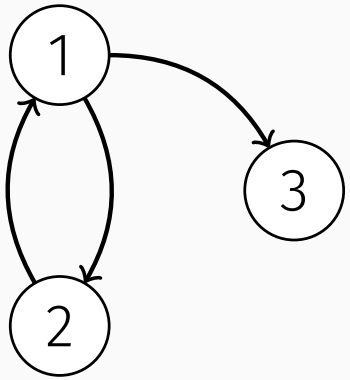
$G = (V, E)$ s.t. $E \subseteq V \times V$

$$G_1 = \Big(\{1,2,3\}, \{(1,2),(1,3),(2,1)\}\Big)$$

In Haskell?

```
data G a = G { vs :: Set a, es :: Set (a,a) }
```

Introduction



$G = (V, E)$ s.t. $E \subseteq V \times V$

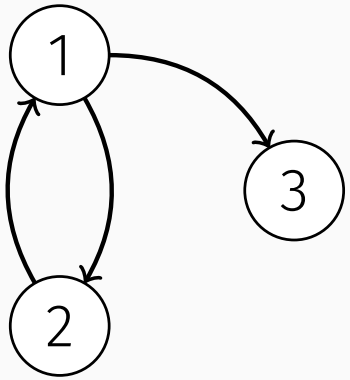
$G_1 = \Big(\{1,2,3\}, \{(1,2),(1,3),(2,1)\}\Big)$

In Haskell?

```
data G a = G { vs :: Set a, es :: Set (a,a) }

g1 = G [1,2,3] [(1,2), (1,3), (2,1)]
```

Introduction



$G = (V, E)$ s.t. $E \subseteq V \times V$

$G_1 = \Big(\{1,2,3\}, \{(1,2),(1,3),(2,1)\}\Big)$

In Haskell?

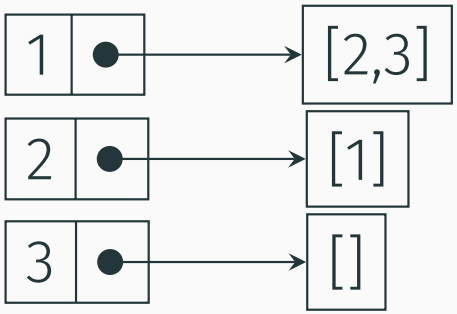
```
data G a = G { vs :: Set a, es :: Set (a,a) }

g1 = G [1,2,3] [(1,2), (1,3), (2,1)]
g2 = G [1,2] [(1,3)]
```

$E \not\subseteq V \times V$

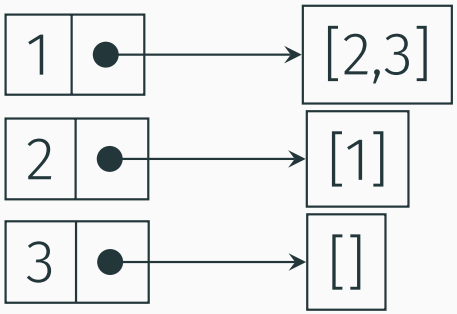
containers & fgl

containers
adjacency lists



containers & fgl

containers
adjacency lists

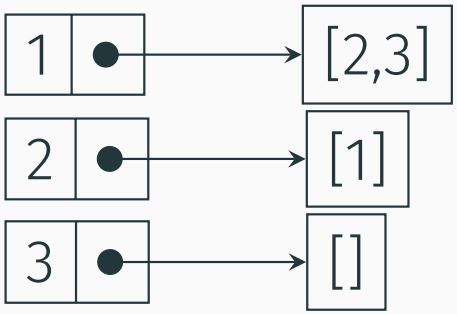


fgl
inductive graphs

- inductive datatype: Context of a vertex + Graph

containers & fgl

containers
adjacency lists



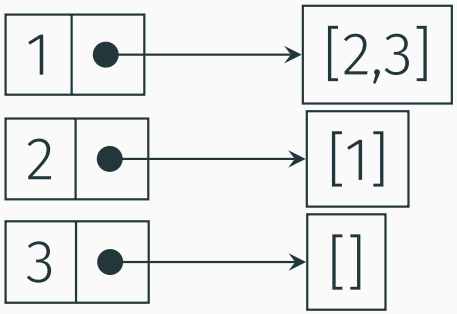
$E \subseteq V \times V$?

fgl
inductive graphs

- inductive datatype: Context of a vertex + Graph

containers & fgl

containers
adjacency lists



$E \subseteq V \times V$ **?**

- partial functions →runtime errors

fgl
inductive graphs

- inductive datatype: Context of a vertex + Graph

Algebraic Graphs

- complete and consistent graph representation
- simple construction primitives (*"the core"*)

Algebraic Graphs

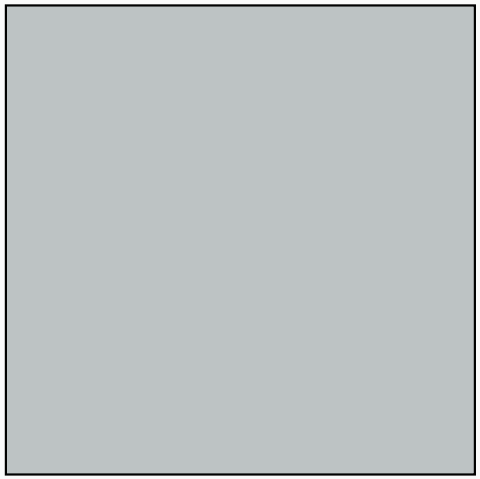
- complete and consistent graph representation
- simple construction primitives (*"the core"*)

Achieved by the datatype

```
data Graph a = Empty
             | Vertex a
             | Overlay (Graph a) (Graph a)
             | Connect (Graph a) (Graph a)
```

Empty (ε) & Vertex

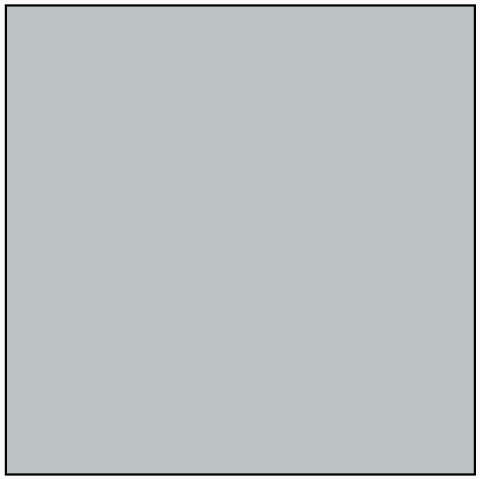
Empty - ε



$\texttt{Empty} = \varepsilon = (\emptyset, \emptyset)$

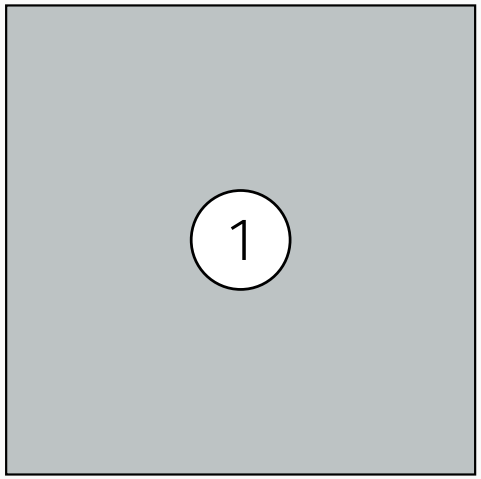
Empty (ε) & Vertex

Empty - ε



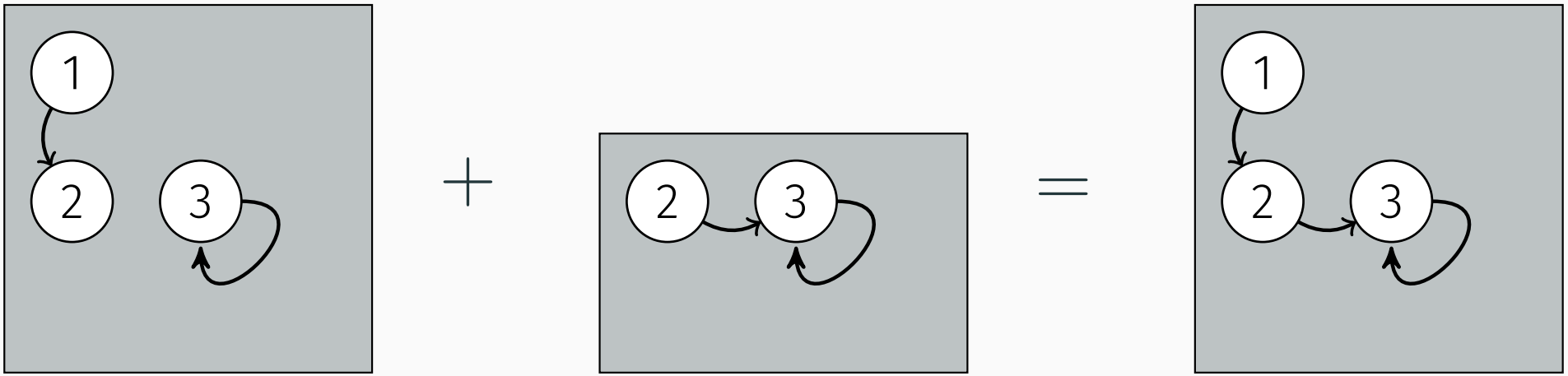
$$\texttt{Empty} = \varepsilon = (\emptyset, \emptyset)$$

Vertex



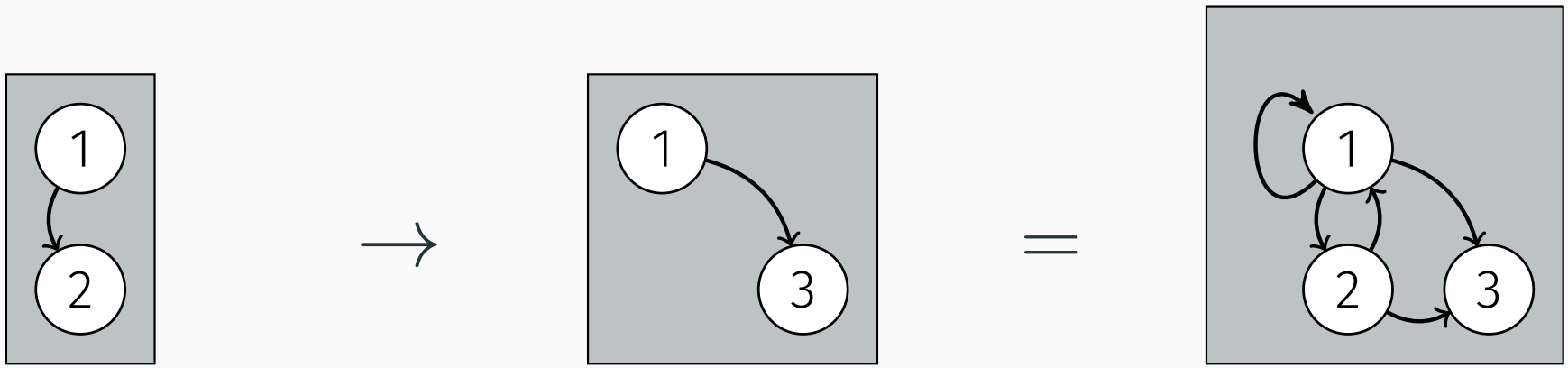
$$\texttt{Vertex}\ 1 = (\{1\}, \emptyset)$$

Overlay ($+$)



$$(V_1, E_1) + (V_2, E_2) := (V_1 \cup V_2, E_1 \cup E_2)$$

Connect ($\rightarrow$ / $*$)



$$(V_1, E_1) \rightarrow (V_2, E_2) \coloneqq (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$$

Graph Construction

Graph Transformation

Algebraic Graphs with Class

Undirected Graphs & more

Formal Verification

Deep Embedding

Formal Verification

Conclusion