Diagram Combinators in MMT

Approach to Building Large Libraries

Florian Rabe[1] and Yasmine Sharoda[2]

[1]FAU Erlangen-Nürnberg, Germany and LRI, Université Paris Sud, France

[2] McMaster University, Canada

Motivation

- Formal libraries are growing

Motivation

- Formal libraries are growing
- Theory graph structure
  - A network of little theories
- Theory Combinators

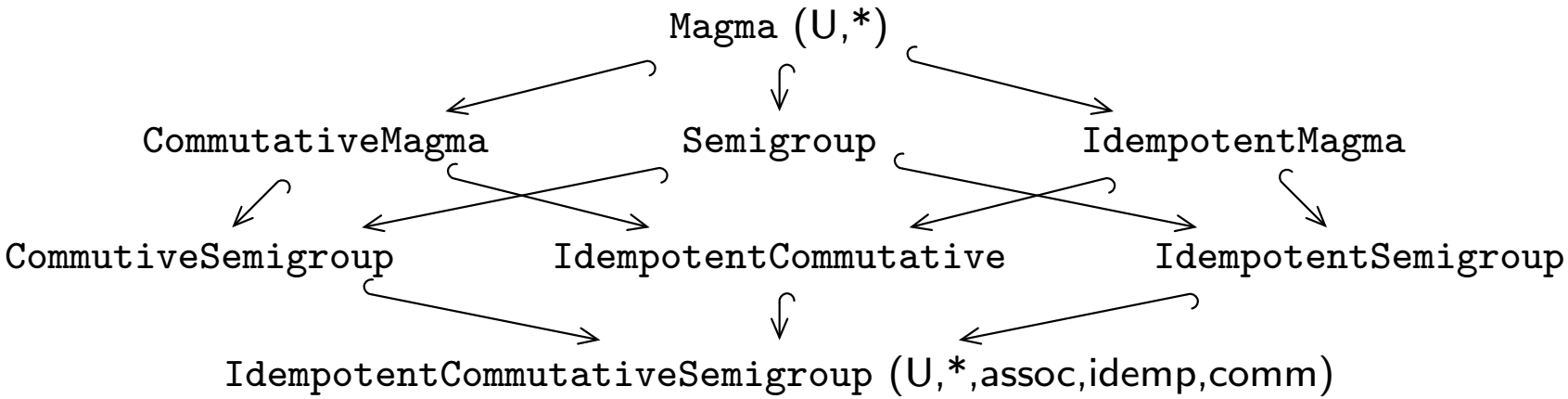
Motivation

- Theories are becoming hard to manage

Motivation

- Theories are becoming hard to manage

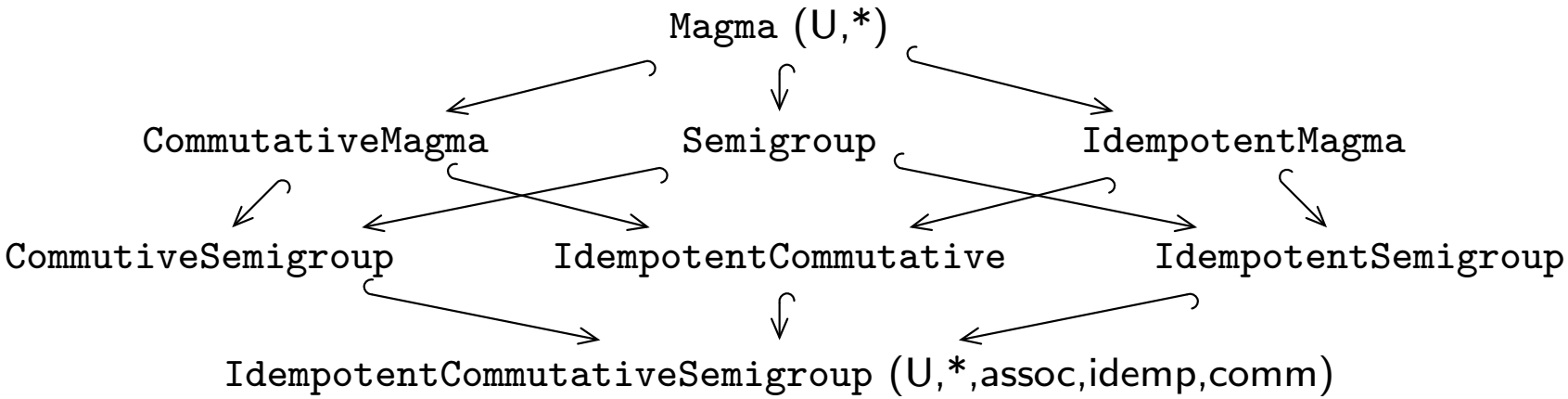
Example:



Motivation

- Theories are becoming hard to manage

Example:



The same hierarchy based on `AdditiveMagma (U,+)`?

Motivation

- Structure library into network of diagrams
- Diagram operators for computing entire diagrams

Diagram Definition

A **diagram** D consists of

- a list of **nodes**: $\mathrm{Node}(l, D_n(l))$

Diagram Definition

A **diagram** D consists of

- a list of **nodes**: $\mathrm{Node}(I, D_n(I))$
- a list of **edges**, $\mathrm{Edge}(I, d \to c, D_E(I))$

Diagram Definition

A **diagram** D consists of

- a list of **nodes**: $\mathrm{Node}(I, D_n(I))$
- a list of **edges**, $\mathrm{Edge}(I, d \to c, D_E(I))$
- an optional label $D^{\mathtt{dist}}$ of a node — the **distinguished** node
- an optional label $\mathrm{Edge}(I, d \xrightarrow{i} c, D(I))$ of an arrow – the **implicit** arrow

Diagram Operations

Based on

- Theory Formation Operations
- Set-Theoretic Operations
- Batch Formation of Theories

Diagram Operations

Based on

- Theory Formation Operations
- Set-Theoretic Operations
- Batch Formation of Theories

Running Example: Building the algebraic hierarchy as a theory graph

From: `Magma` (`U`, `_*_`)

To: `IdempotentCommutativeSemigroup`
(`U`, `_*_`, `assoc`, `comm`, `idemp`)

Diagram Operations

Based on

- Theory Formation Operations
- Set-Theoretic Operations
- Batch Formation of Theories

Theory Formation Operations

Interpreting combinators from MathScheme [CO12] as diagram operations

Theory Formation Operations

1. Extension

```
diagram d′ := d extended_by Σ
```

Theory Formation Operations

1. Extension

```
diagram d' := d extended_by Σ
```

Input Diagram

$\longrightarrow D^{\texttt{dist}}$

Output Diagram

$\longrightarrow D^{\texttt{dist}} \longrightarrow \texttt{pres}$

Theory Formation Operations

1. Extension

```
diagram d′ := d extended_by Σ
```

Input Diagram

$\longrightarrow D^{\texttt{dist}}$

Output Diagram

$\longrightarrow D^{\texttt{dist}} \longrightarrow \texttt{pres}$

```
diagram Magma = Carrier  extended_by {_*_:U → U → U}
diagram Semigroup = Magma extended_by {assoc : ···}
```

$Carrier^{\texttt{dist}} \hookrightarrow Magma^{\texttt{dist}} \hookrightarrow \texttt{pres}$

Theory Formation Operations

2. Rename

```
diagram d′ := d rename r
```

Input Diagram

$\longrightarrow D^{\mathtt{dist}}$

Output Diagram

$\longrightarrow D^{\mathtt{dist}} \longrightarrow \mathtt{pres}$

```
diagram AdditiveMagma = Magma rename { * ⇝ +}
```

$Carrier^{\mathtt{dist}} \hookrightarrow Magma^{\mathtt{dist}} \longmapsto \mathtt{pres}$

Theory Formation Operations

3. Combine

```
diagram d′ := combine d₁ r₁ d₂ r₂
```

Input Diagram

$$\begin{tikzcd} S \arrow[r] \arrow[d] & D_1^{\mathtt{dist}} \\ D_2^{\mathtt{dist}} & \end{tikzcd}$$

Output Diagram

$$\begin{tikzcd} S \arrow[r] \arrow[d] \arrow[drr] & D_1^{\mathtt{dist}} \arrow[r] & D_1^{\mathtt{dist}} \cdot r_1 \arrow[dd] \\ D_2^{\mathtt{dist}} \arrow[d] & & \\ D_2^{\mathtt{dist}} \cdot r_2 \arrow[rr] & & \mathtt{pres} \end{tikzcd}$$

Theory Formation Operations

3. Combine

```
diagram PointedMagma := Combine Pointed {} Magma {}
```

$$
\begin{tikzcd}
\mathit{Carrier}(U) \arrow[r] \arrow[d] \arrow[drr] & \mathtt{Magma}(U,*) \arrow[r] & \mathtt{Magma}(U,*) \arrow[d] \\
\mathtt{Pointed}(U,e) \arrow[d] & & \\
\mathtt{Pointed}(U,e) \arrow[rr] & & \mathtt{pres}(U,*,e)
\end{tikzcd}
$$

Theory Formation Operations

3. Combine

```
diagram AdditiveSemigroup :=
   Combine Semigroup { * ⇝ + } AddMagma {}
```

$$\begin{tikzcd}
\mathtt{Magma}(U,*) \arrow[r] \arrow[d] & \mathtt{AdditiveMagma}(U,+) \\
\mathtt{Semigroup}(U,*) &
\end{tikzcd}$$

Theory Formation Operations

3. Combine

```
diagram AdditiveSemigroup :=
   Combine Semigroup { * ⇝ + } AddMagma {}
```

$$
\begin{tikzcd}
\mathtt{Magma}(U,*) \arrow[r] \arrow[d] \arrow[drrr] & \mathtt{AdditiveMagma}(U,+) \arrow[r] & \mathtt{AdditiveMagma}(U,+) \arrow[dd] & \\
\mathtt{Semigroup}(U,*) \arrow[d] & & & \\
A\,\mathtt{Semigroup}(U,+) \arrow[rrr] & & & \mathtt{pres}
\end{tikzcd}
$$

Theory Formation Operations

3. Mixin

```
diagram d′ := Mixin d₁ r₁ d₂ r₂
```

Input Diagram

$$\begin{tikzcd} S \arrow[r] \arrow[d] & D_1^{\mathtt{dist}} \\ D_2^{\mathtt{dist}} & \end{tikzcd}$$

Output Diagram

$$\begin{tikzcd} S \arrow[r] \arrow[d] \arrow[drr] & D_1^{\mathtt{dist}} \arrow[r] & D_1^{\mathtt{dist}} \cdot r_1 \arrow[dd] \\ D_2^{\mathtt{dist}} \arrow[d] & & \\ D_2^{\mathtt{dist}} \cdot r_2 \arrow[rr] & & \mathtt{pres} \end{tikzcd}$$

Theory Formation Operations

Theory Expression	Input Diagram(s)	Output Diagram
`D extends Σ`	$\longrightarrow D^{\mathtt{dist}}$	$\longrightarrow D^{\mathtt{dist}} \longrightarrow \mathtt{pres}$
`D rename r`	$\longrightarrow D^{\mathtt{dist}}$	$\longrightarrow D^{\mathtt{dist}} \longrightarrow \mathtt{pres}$
`combine D₁ r₁ D₂ r₂` `mixin D₁ r₁ D₂ r₂`	$\begin{tikzcd} S \arrow[r] \arrow[d] & D_1^{\mathtt{dist}} \\ D_2^{\mathtt{dist}} & \end{tikzcd}$	$\begin{tikzcd} S \arrow[r] \arrow[d] \arrow[ddr] & D_1^{\mathtt{dist}} \arrow[r] & D_1^{\mathtt{dist}} \cdot r_1 \arrow[dd] \\ D_2^{\mathtt{dist}} \arrow[d] & & \\ D_2^{\mathtt{dist}} \cdot r_2 \arrow[rr] & & \mathtt{pres} \end{tikzcd}$

Diagram Operations

Based on

- Theory Formation Operations
- Set-Theoretic Operations
- Batch Formation of Theories

Set Theoretic Operations

- Diagram from named elements

  $\mathbf{diag}(\mathsf{n}_1, \cdots, \mathsf{n}_r)$

Set Theoretic Operations

- Diagram from named elements

  $$\mathbf{diag}(\mathsf{n}_1, \cdots, \mathsf{n}_r)$$

  - Listed theories and morphisms
  - For any listed diagram, its distinguished nodes and all implicit arrows
  - Domain and codomain of every morphism

Set Theoretic Operations

- Diagram from named elements

  $$\mathbf{diag}(\mathsf{n}_1, \cdots, \mathsf{n}_r)$$

  - Listed theories and morphisms
  - For any listed diagram, its distinguished nodes and all implicit arrows
  - Domain and codomain of every morphism

Example:

```
diagram MagmaExtensions :=
   diag(Magma, Semigroup, IdempotentMagma, CommutativeMagma)
```

$$\begin{tikzcd}
 & \mathtt{Magma} \arrow[dl, hook] \arrow[d, hook] \arrow[dr, hook] & \\
\mathtt{CommutativeMagma} & \mathtt{Semigroup} & \mathtt{IdempotentMagma}
\end{tikzcd}$$

Set Theoretic Operations

- Union
- Intersection
- Difference

Diagram Operations

Based on

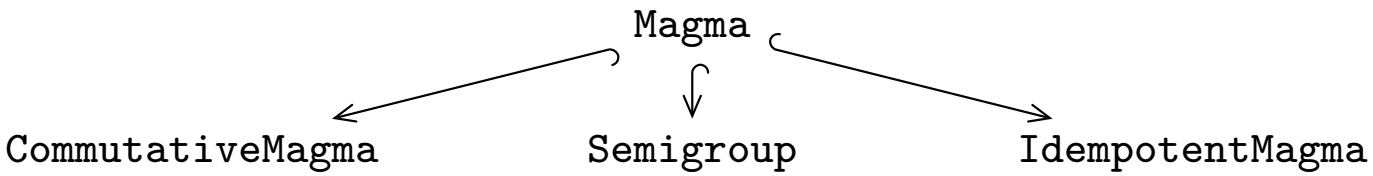
- Theory Formation Operations
- Set-Theoretic Operations
- Batch Formation of Theories

Batch Operations

Systematically apply an operation to a diagram

- Batch Combine

```
diagram IdempotentCommutativeSemigroup = BCOMBINE MagmaExtensions
```

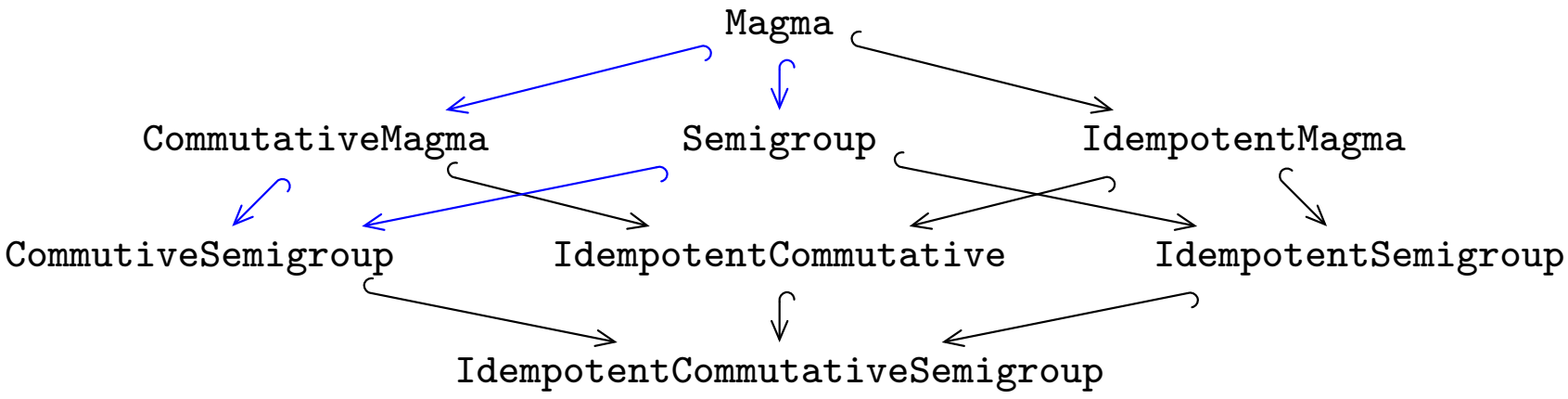


Batch Operations

Systematically apply an operation to a diagram

- Batch Combine

```
diagram IdempotentCommutativeSemigroup = BCOMBINE MagmaExtensions
```

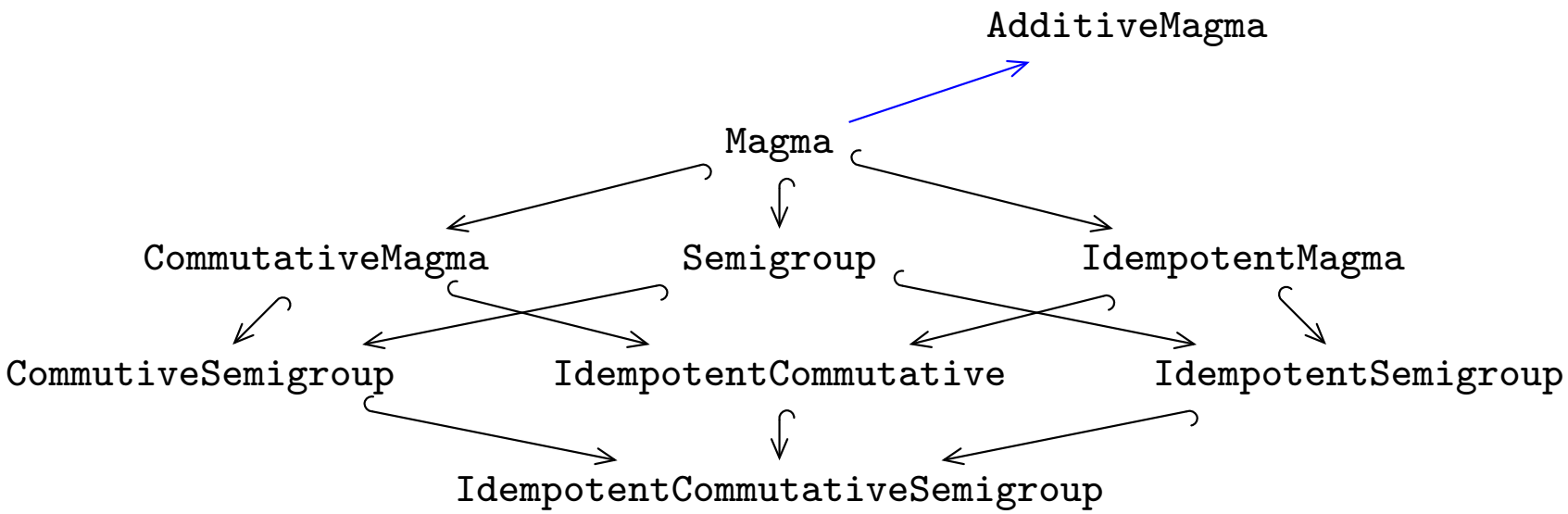


Batch Operations

Systematically apply an operation to a diagram

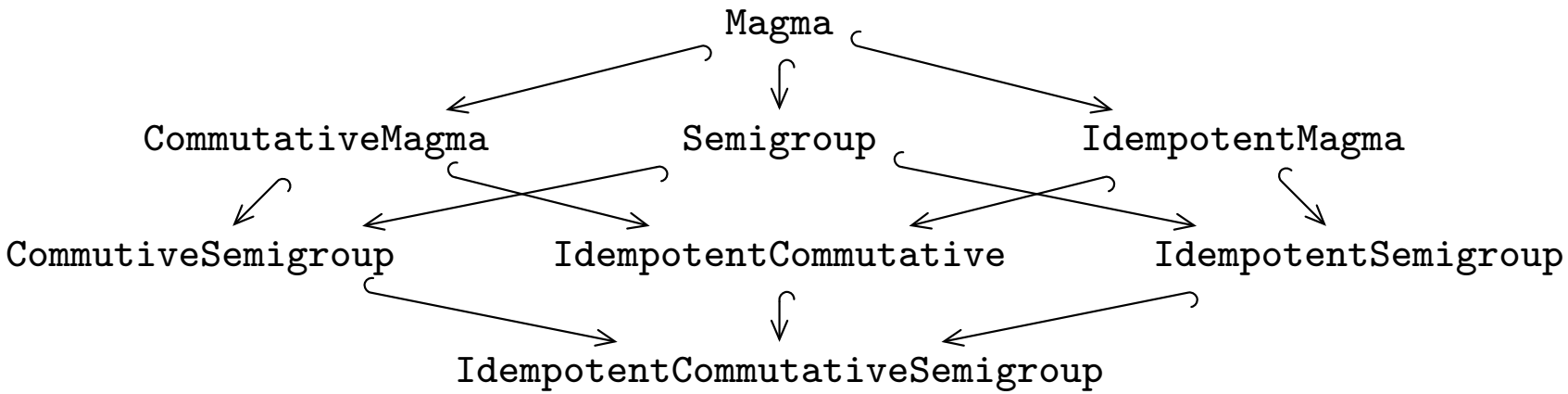
- Batch Combine
- Batch Mixin

```
diagram AdditivaMagma := Magma rename {* ⇝ + }
diagram AddIdemptCommSemigroup :=
   BMIXIN AdditiveMagma IdempotentCommutativeSemigroup
```



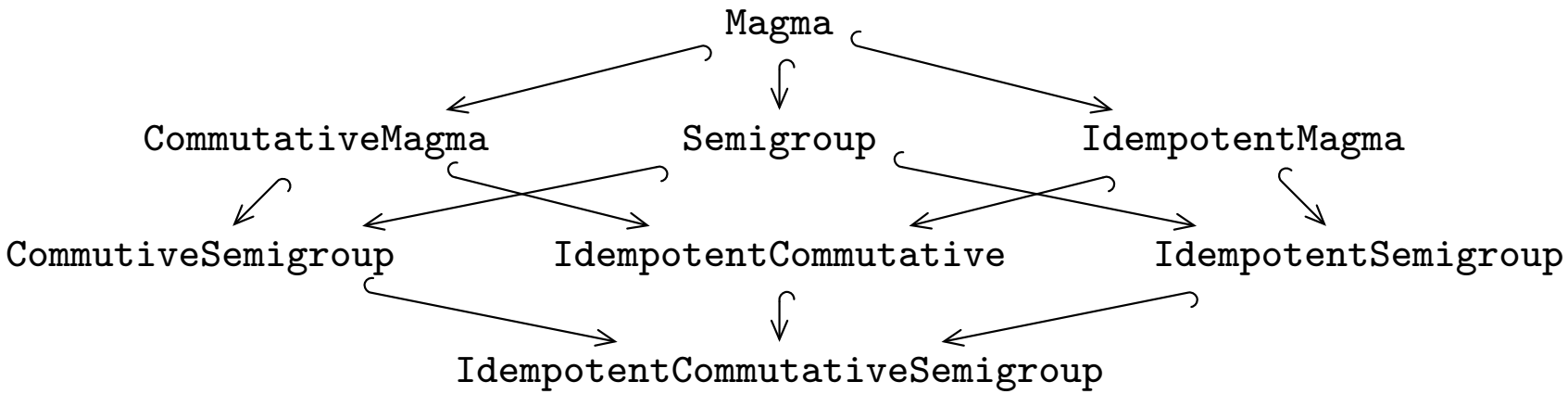
Choosing Names

```
diagram IdempotentCommutativeSemigroup =
  BCOMBINE MagmaExtensions
```



Choosing Names

```
diagram IdempotentCommutativeSemigroup =
  BCOMBINE MagmaExtensions
```



alias n := N

Extensible Framework

How to define a new combinator?

1. Define a new theory extending `diagrams`

```
theory Combinators =
  include ?Diagrams
  extends # 1 extended_by {%L1_L2,···}
  combine # COMBINE 1 {2,···} 3 {4,···}
```

Extensible Framework

How to define a new combinator?

1. Define a new theory extending `diagrams`

```
theory Combinators =
  include ?Diagrams
  extends # 1 extended_by {%L1_L2,···}
  combine # COMBINE 1 {2,···} 3 {4,···}
```

2. Define a scala rule for computing the output diagram

```
rule rules?ComputeExtends
rule rules?ComputeCombine
```

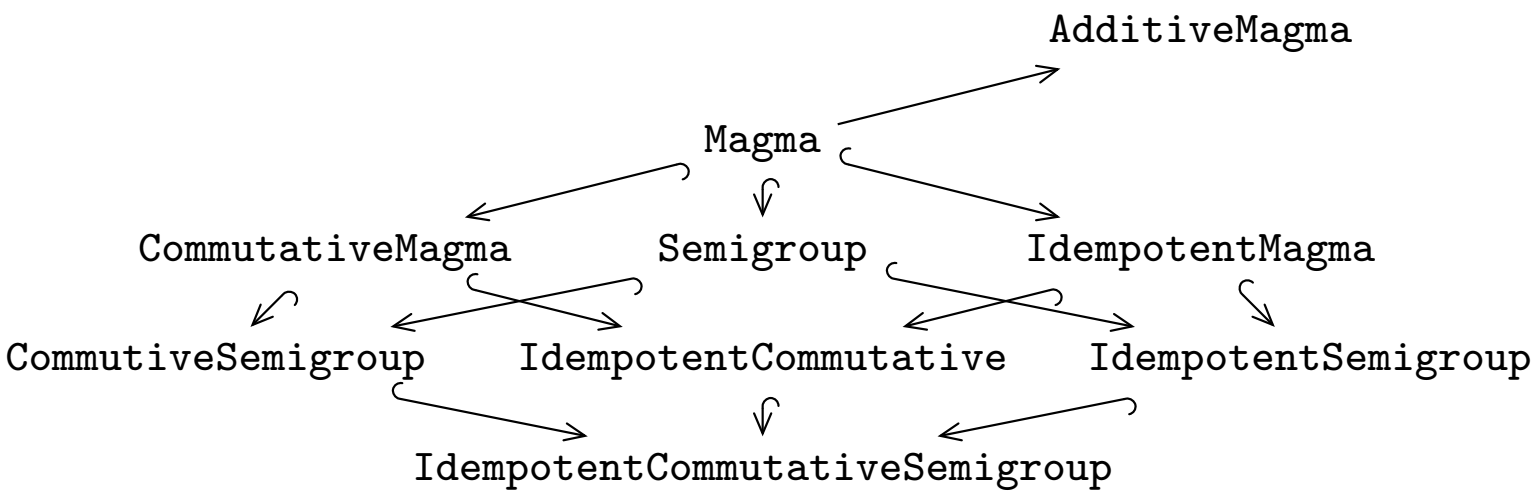
Future Work

- Implement universal-algebra inspired combinators
- Build the MathScheme library into MMT
- Giving user control over choosing the names of the generated theories / morphisms

Conclusion

- Formal language for creating and manipulating diagrams.
- Environment to extend the language to allow user defined operations

```
diagram IdempotentMagma := Magma extended_by {idemp : ···}
diagram CommutativeMagma := Magma extended_by {comm : ···}
diagram Semigroup := Magma extended_by {assoc : ···}
diagram AdditiveMagma := Magma rename { * ⇝ + }
diagram MagmaExtensions :=
  Union (IdempotentMagma, CommutativeMagma, Semigroup)
diagram IdempCommSemigroup := BCOMBINE (MagmaExtensions)
diagram AddIdempCommSemigroup := BMIXIN AdditiveMagma
    IdempotentCommutativeSemigroup
```



Related Work

- Jacques Carette and Russel O'Connor, *Theory Presentation Combinators*, Intelligent Computer Mathematics (J. Jeuring, J. Campbell, J. Carette, G. Dos Reis, P. Sojka, M. Wenzel, and V. Sorge, eds.), vol. 7362, Springer, 2012, pp. 202–215.
- Jacques Carette and Russell O'Connor, *Theory presentation combinators*, CoRR: http://arxiv.org/abs/1812.08079 (2018).
- *The distributed ontology, modeling, and specification language*, Tech. report, Object Management Group (OMG), 10 2018, Version 1.0.

Thank You!