An interface for diagrammatic proofs in Coq

Luc Chabassier

Université Paris-Saclay, INRIA project Deducteam,
LMF, ENS Paris-Saclay

TYPES 2023

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

- Partial operation in Coq are hard

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

- Partial operation in Coq are hard
- Partiality has a graph structure :

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

- Partial operation in Coq are hard
- Partiality has a graph structure :
  - Easy to encode using dependent types

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

- Partial operation in Coq are hard
- Partiality has a graph structure :
  - ~~Easy to encode using dependent types~~

Category theory difficulties in Coq

Category theory

An algebra based on a *partial* operation : composition.

- Partial operation in Coq are hard
- Partiality has a graph structure :
  - ~~Easy to encode using dependent types~~
  - Enables graphical reasonning : diagrams

Presentation plan

Basic usage

Partie 1

Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: morphism\ C\ b\ c \\
m' &: morphism\ C\ c\ d \\
m'' &: morphism\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$

Extracting the graph from the context

$$\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \mathit{morphism}\ C\ b\ c \\
m' &: \mathit{morphism}\ C\ c\ d \\
m'' &: \mathit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}$$

$$\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}$$

Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \textit{morphism}\ C\ b\ c \\
m' &: \textit{morphism}\ C\ c\ d \\
m'' &: \textit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

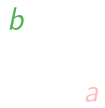
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$

a

Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \textit{morphism}\ C\ b\ c \\
m' &: \textit{morphism}\ C\ c\ d \\
m'' &: \textit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

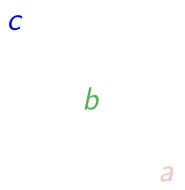
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \mathit{morphism}\ C\ b\ c \\
m' &: \mathit{morphism}\ C\ c\ d \\
m'' &: \mathit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

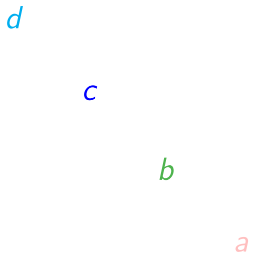
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \mathit{morphism}\ C\ b\ c \\
m' &: \mathit{morphism}\ C\ c\ d \\
m'' &: \mathit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

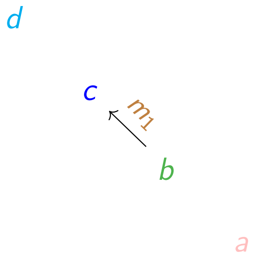
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$C : \text{Category}$$
$$a, b, c, d : C$$
$$m_1, m_2, m_3 : \textit{morphism } C\ b\ c$$
$$m' : \textit{morphism } C\ c\ d$$
$$m'' : \textit{morphism } C\ a\ b$$
$$H_{12} : 1 \circ m_1 = m_2$$
$$H_{13} : m_1 = m_3$$

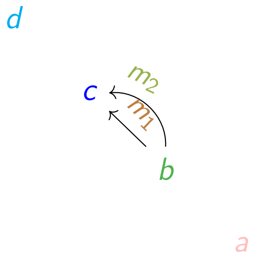
$$1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1$$
$$= 1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''$$



Extracting the graph from the context

$$C : \text{Category}$$
$$a, b, c, d : C$$
$$m_1, m_2, m_3 : \mathit{morphism}\ C\ b\ c$$
$$m' : \mathit{morphism}\ C\ c\ d$$
$$m'' : \mathit{morphism}\ C\ a\ b$$
$$H_{12} : 1 \circ m_1 = m_2$$
$$H_{13} : m_1 = m_3$$

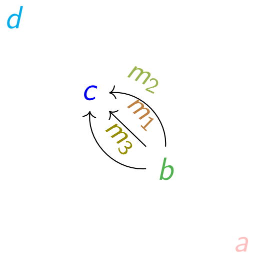
$$1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1$$
$$= 1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''$$



Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \mathit{morphism}\ C\ b\ c \\
m' &: \mathit{morphism}\ C\ c\ d \\
m'' &: \mathit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

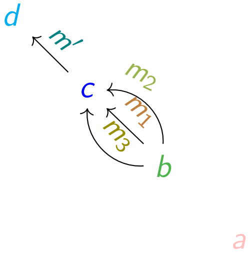
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \mathit{morphism}\ C\ b\ c \\
m' &: \mathit{morphism}\ C\ c\ d \\
m'' &: \mathit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

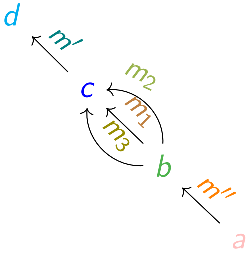
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$
\begin{aligned}
C &: \text{Category} \\
a, b, c, d &: C \\
m_1, m_2, m_3 &: \textit{morphism}\ C\ b\ c \\
m' &: \textit{morphism}\ C\ c\ d \\
m'' &: \textit{morphism}\ C\ a\ b \\
H_{12} &: 1 \circ m_1 = m_2 \\
H_{13} &: m_1 = m_3
\end{aligned}
$$

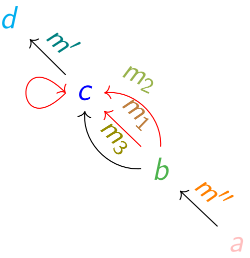
$$
\begin{aligned}
&1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1 \\
=&1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''
\end{aligned}
$$



Extracting the graph from the context

$$C : \text{Category}$$
$$a, b, c, d : C$$
$$m_1, m_2, m_3 : \textit{morphism}\ C\ b\ c$$
$$m' : \textit{morphism}\ C\ c\ d$$
$$m'' : \textit{morphism}\ C\ a\ b$$
$$H_{12} : 1 \circ m_1 = m_2$$
$$H_{13} : m_1 = m_3$$

$$1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1$$
$$= 1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''$$

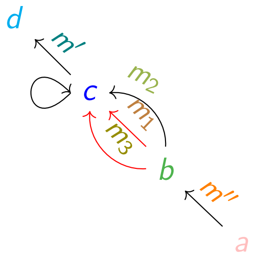


$$H_{12} \doteq H_{12}$$

Extracting the graph from the context

$$C : \text{Category}$$
$$a, b, c, d : C$$
$$m_1, m_2, m_3 : morphism\ C\ b\ c$$
$$m' : morphism\ C\ c\ d$$
$$m'' : morphism\ C\ a\ b$$
$$H_{12} : 1 \circ m_1 = m_2$$
$$H_{13} : m_1 = m_3$$

$$1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1$$
$$= 1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''$$

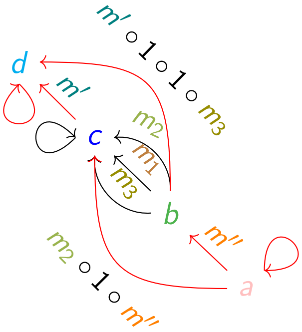


$$H_{12} \doteq H_{12} \qquad H_{13} \doteq H_{13}$$

Extracting the graph from the context

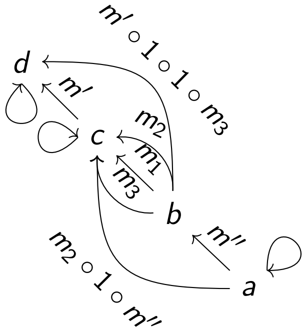
$$C : \text{Category}$$
$$a, b, c, d : C$$
$$m_1, m_2, m_3 : \mathit{morphism}\ C\ b\ c$$
$$m' : \mathit{morphism}\ C\ c\ d$$
$$m'' : \mathit{morphism}\ C\ a\ b$$
$$H_{12} : 1 \circ m_1 = m_2$$
$$H_{13} : m_1 = m_3$$

$$1 \circ m' \circ (m_2 \circ 1 \circ m'') \circ 1$$
$$= 1 \circ (m' \circ 1 \circ 1 \circ m_3) \circ m''$$

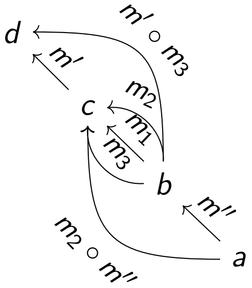


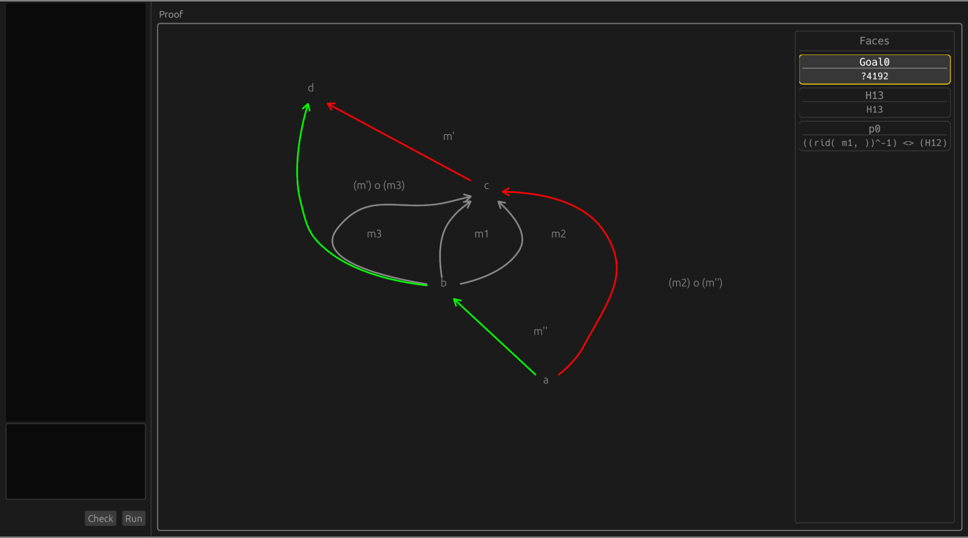
$$H_{12} \doteq H_{12} \qquad H_{13} \doteq H_{13}$$
$$\text{Goal} \doteq ?0$$

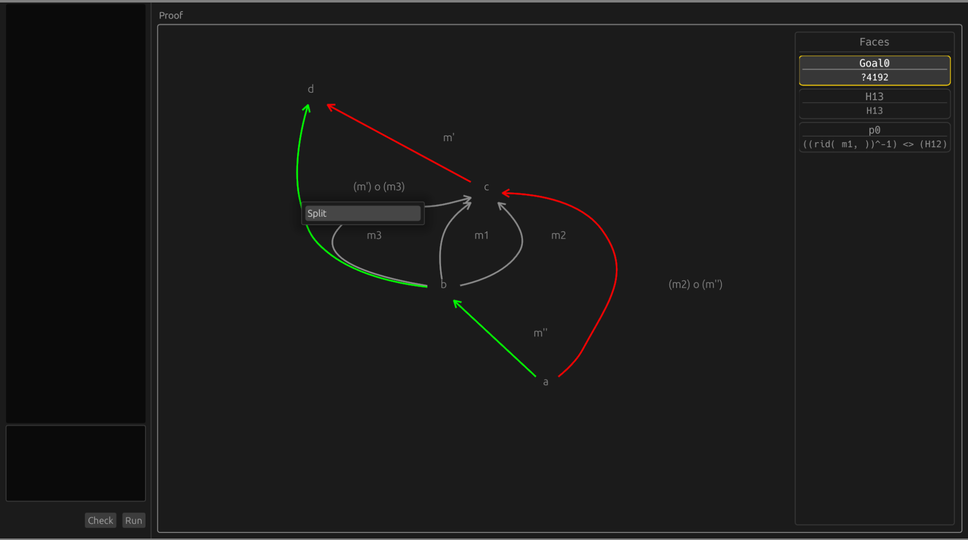
Removing identities

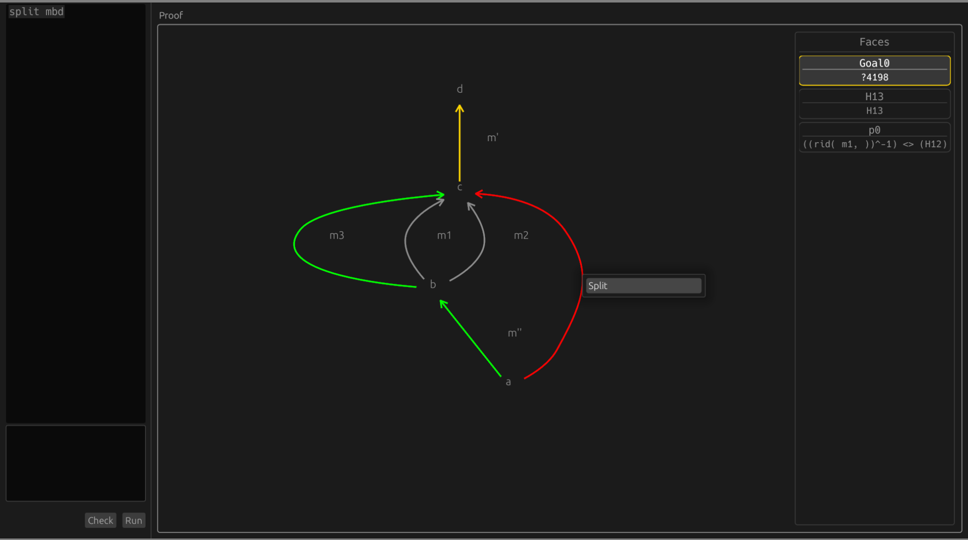


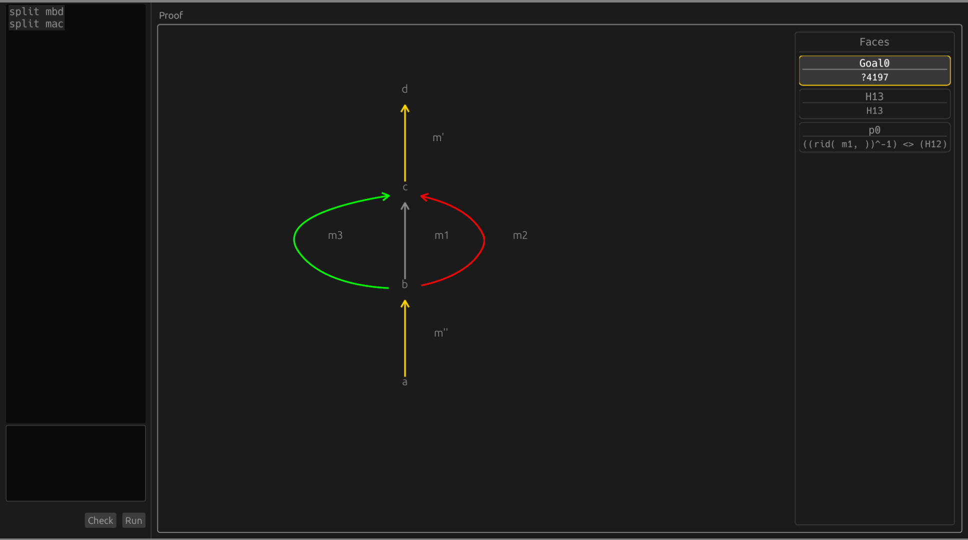
$\Rightarrow$




Proof
d
m'
(m') o (m3)
c
m3
m1
m2
b
(m2) o (m'')
m''
a
Faces
Goal0
?4192
H13
H13
p0
((rid(m1,))^-1) <> (H12)
Check
Run


Proof
Faces
Goal0
?4192
H13
H13
p0
((rid(m1,))^-1) <> (H12)
d
m'
(m') o (m3)
c
Split
m3
m1
m2
b
(m2) o (m'')
m''
a
Check
Run

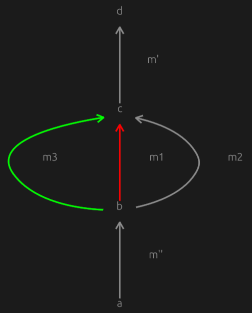

split mbd
Proof
Faces
Goal0
?4198
H13
H13
p0
((rid(m1,))^-1) <> (H12)
d
m'
c
m3
m1
m2
b
Split
m''
a
Check
Run


split mbd
split mac
Check
Run
Proof
d
m'
c
m3
m1
m2
b
m''
a
Faces
Goal0
?4197
H13
H13
p0
((rid(m1,))^-1) <> (H12)

```
split mbd
split mac
shrink Goal0
```

Check Run

Proof



Faces

Goal0

?4200

H13

H13

p0

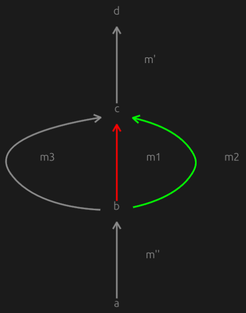
`((rid( m1, ))^-1) <> (H12)`

```
split mbd
split mac
shrink Goal0
```

Check Run

Proof

[redacted]

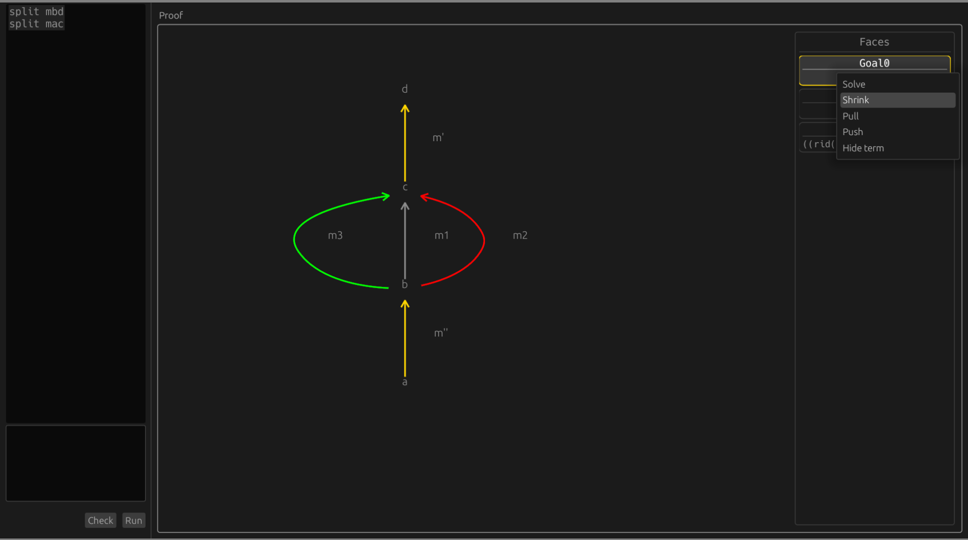


Faces

Goal0
?4200

H13
H13

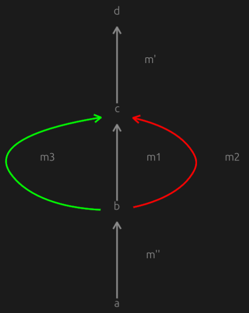
p0
((rid(m1,))^-1) <> (H12)


split mbd
split mac
Check
Run
Proof
Faces
Goal0
Solve
Shrink
Pull
Push
Hide term
((rid(
d
m'
c
m3
m1
m2
b
m''
a

```
split mbd
split mac
shrink Goal0
```

Check Run

Proof

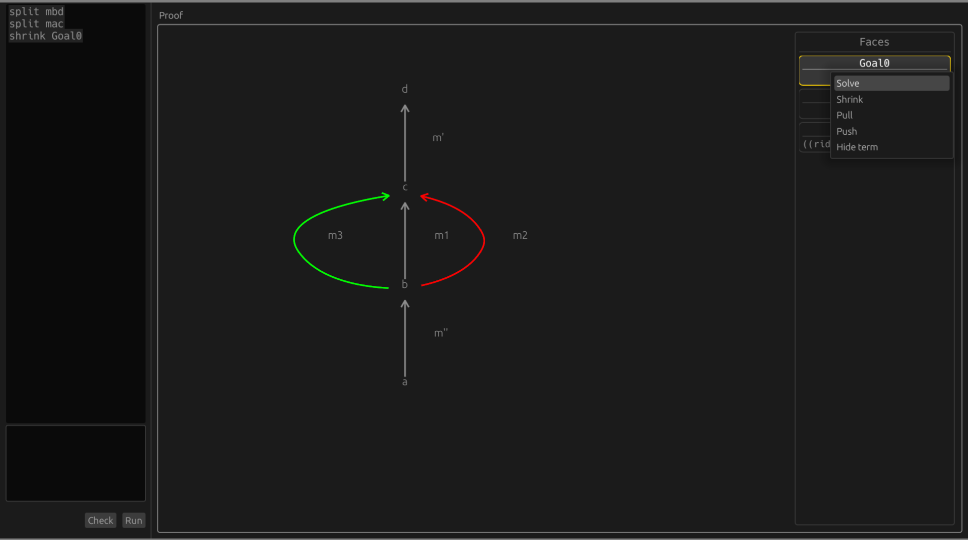


Faces

Goal0
?4200

H13
H13

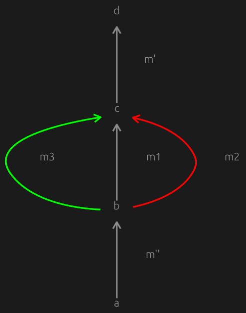
p0
((rid(m1,))^-1) <> (H12)


split mbd
split mac
shrink Goal0
Check
Run
Proof
d
m'
c
m3
m1
m2
b
m''
a
Faces
Goal0
Solve
Shrink
Pull
Push
Hide term
((rid

```
split mbd
split mac
shrink Goal0
solve Goal0
```

Check Run

Proof



Faces

Goal0

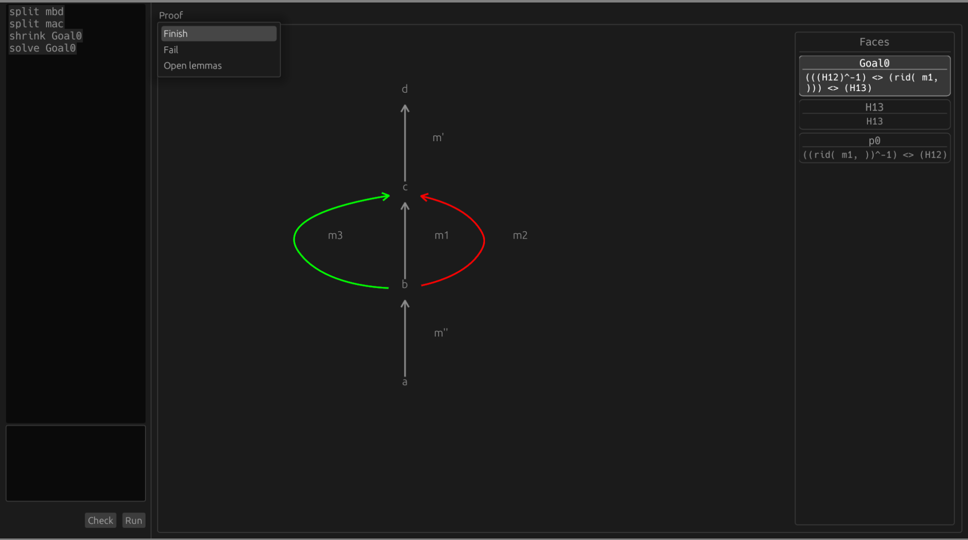
`(((H12)^-1) <> (rid( m1, ))) <> (H13)`

H13

`H13`

p0

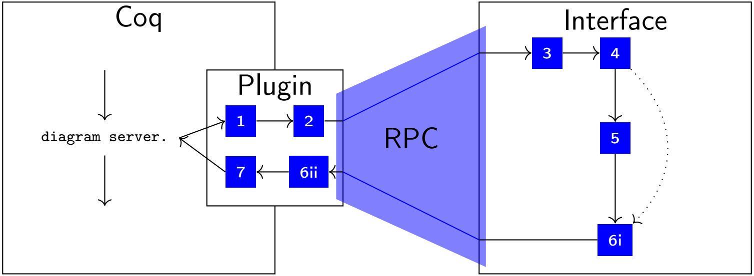
`((rid( m1, ))^-1) <> (H12)`


split mbd
split mac
shrink Goal0
solve Goal0
Check
Run
Proof
Finish
Fail
Open lemmas
d
m'
c
m3
m1
m2
b
m''
a
Faces
Goal0
(((H12)^-1) <> (rid(m1,))) <> (H13)
H13
H13
p0
((rid(m1,))^-1) <> (H12)

Architecture

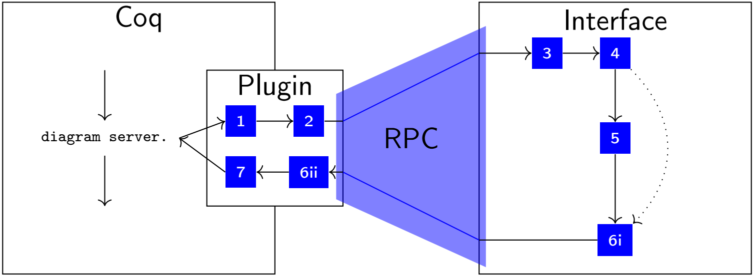
Partie 2

Plugin architecture



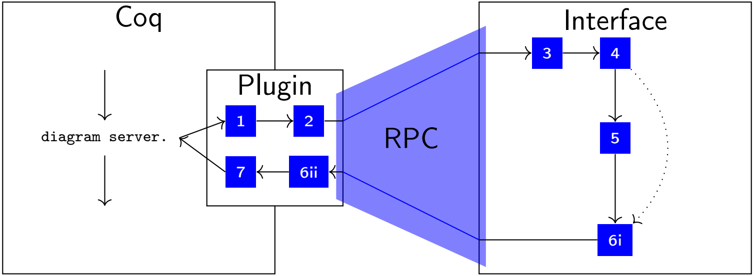
1. Construct graph from proof context

Plugin architecture



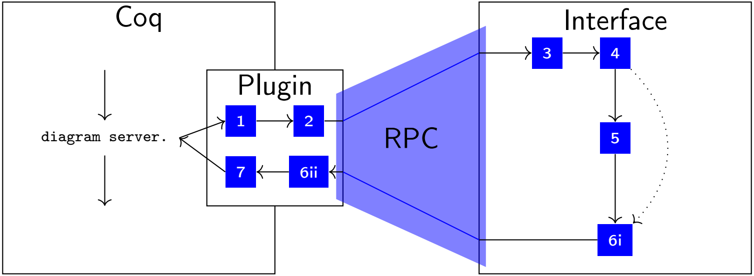
2 Extract lemmas

Plugin architecture



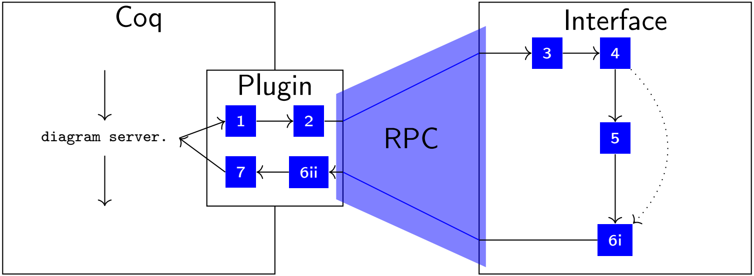
3 Preprocess and layout graphs

Plugin architecture



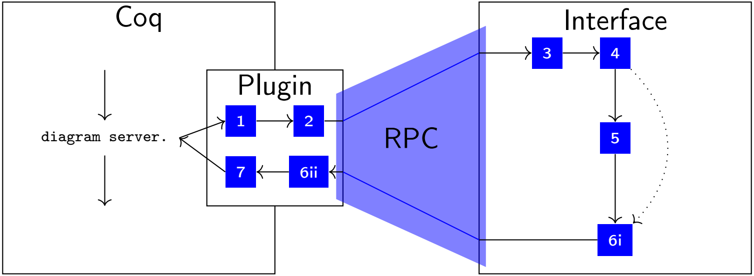
4 *(optional)* If available, replay proof script

Plugin architecture



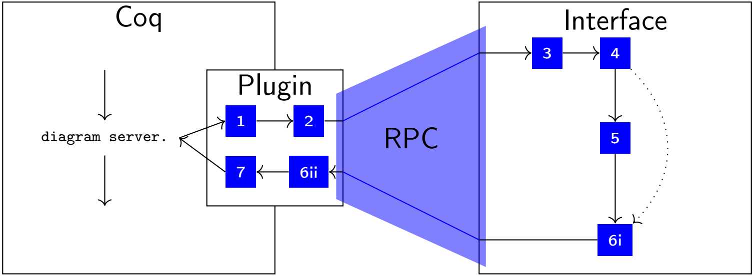
5 Let the user interact with the interface

Plugin architecture



6 Construct a proof term

Plugin architecture



7 Use the proof term to refine the goal

Lemmas

Partie 3

Lemma extraction

$$\forall (C\ \ D : \text{Category}),$$
$$\forall (x\ \ y : C),$$
$$\forall (m_1\ \ m_2 : \text{morphism}\ \ C\ \ x\ \ y),$$
$$\forall (F : \text{Functor}\ \ C\ \ D),$$
$$m_1 = m_2 \implies F\ \ m_1 = F\ \ m_2$$

Lemma extraction

$$
\begin{aligned}
&\forall (C\ D : \text{Category}),\\
&\forall (x\ y : C),\\
&\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),\\
&\forall (F : \text{Functor}\ C\ D),\\
&\quad m_1 = m_2 \implies F\ m_1 = F\ m_2
\end{aligned}
$$

Lemma extraction

$$\forall (C\ D : \text{Category}),$$
$$\forall (x\ y : C),$$
$$\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),$$
$$\forall (F : \text{Functor}\ C\ D),$$
$$m_1 = m_2 \implies F\ m_1 = F\ m_2$$

Lemma extraction

$?x$

$$\forall (C\ D : \text{Category}),$$
$$\forall (x\ y : C),$$
$$\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),$$
$$\forall (F : \text{Functor}\ C\ D),$$
$$m_1 = m_2 \implies F\ m_1 = F\ m_2$$

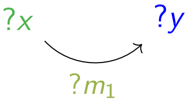
Lemma extraction

$$\forall (C\ D : \text{Category}),$$
$$\forall (x\ y : C),$$
$$\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),$$
$$\forall (F : \text{Functor}\ C\ D),$$
$$m_1 = m_2 \implies F\ m_1 = F\ m_2$$

$?x$ $?y$

Lemma extraction

$\forall(C\ D : \text{Category}),$
$\forall(x\ y : C),$
$\forall(m_1\ m_2 : \text{morphism}\ C\ x\ y),$
$\forall(F : \text{Functor}\ C\ D),$
$m_1 = m_2 \implies F\ m_1 = F\ m_2$



Lemma extraction

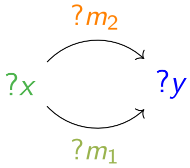
$\forall(C\ D : \text{Category}),$

$\forall(x\ y : C),$

$\forall(m_1\ m_2 : \text{morphism}\ C\ x\ y),$

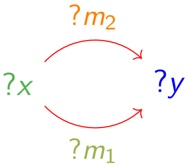
$\forall(F : \text{Functor}\ C\ D),$

$m_1 = m_2 \implies F\ m_1 = F\ m_2$



Lemma extraction

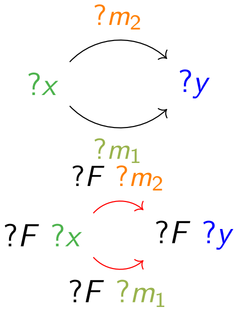
$$
\begin{aligned}
&\forall (C\ D : \text{Category}),\\
&\forall (x\ y : C),\\
&\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),\\
&\forall (F : \text{Functor}\ C\ D),\\
&\quad m_1 = m_2 \implies F\ m_1 = F\ m_2
\end{aligned}
$$



$$p0 \doteq ?p$$

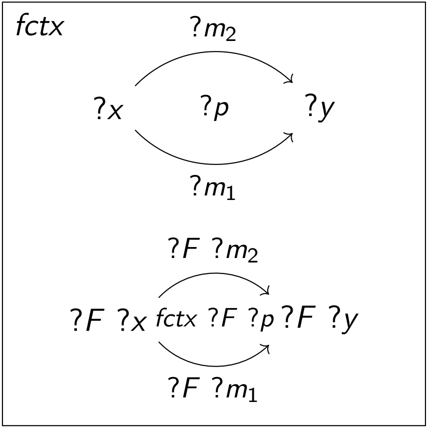
Lemma extraction

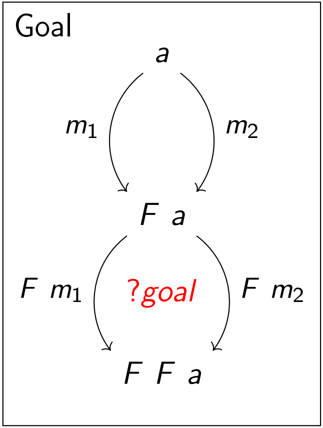
$\forall (C\ D : \text{Category}),$
$\forall (x\ y : C),$
$\forall (m_1\ m_2 : \text{morphism}\ C\ x\ y),$
$\forall (F : \text{Functor}\ C\ D),$
$m_1 = m_2 \implies F\ m_1 = F\ m_2$



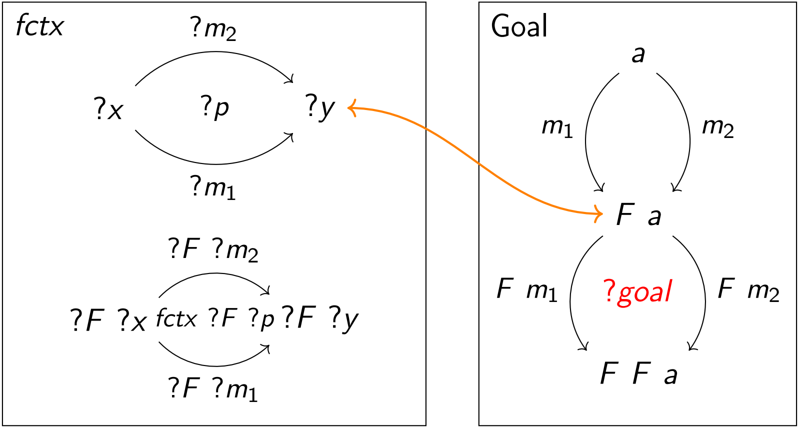
$p0 \doteq ?p$
$p1 \doteq \text{fctx}\ ?F\ ?p$

Lemma application

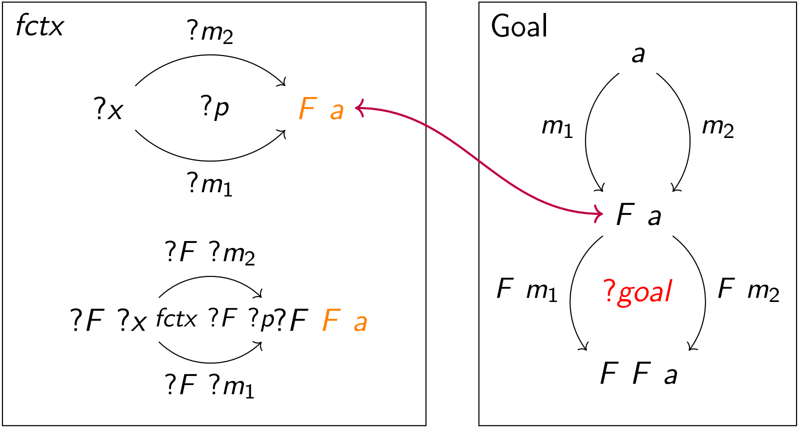




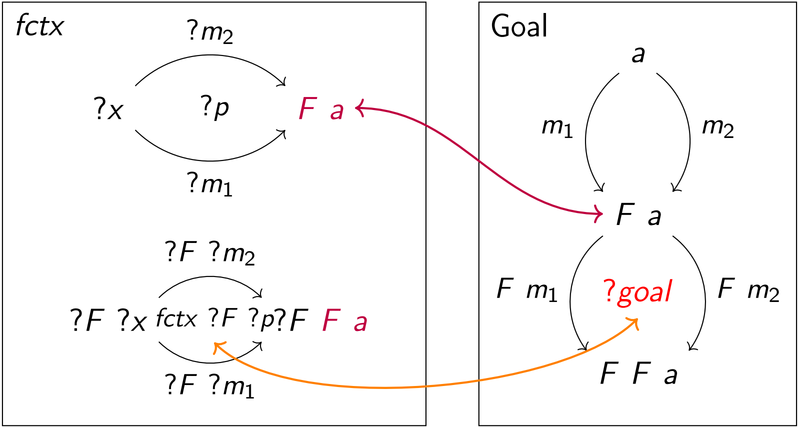
Lemma application



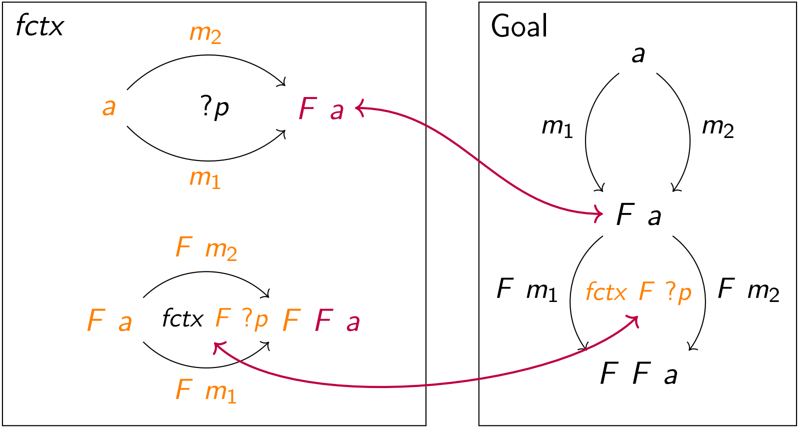
Lemma application



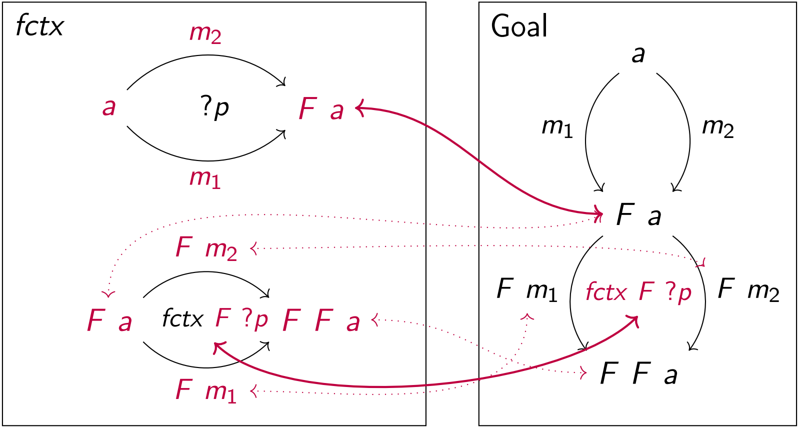
Lemma application



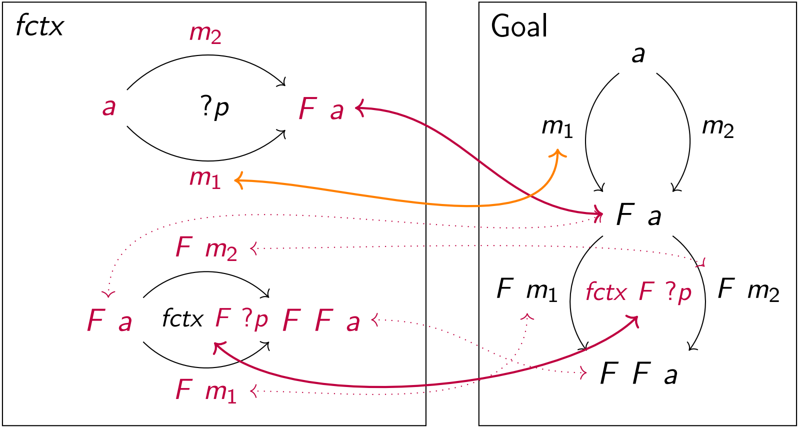
Lemma application



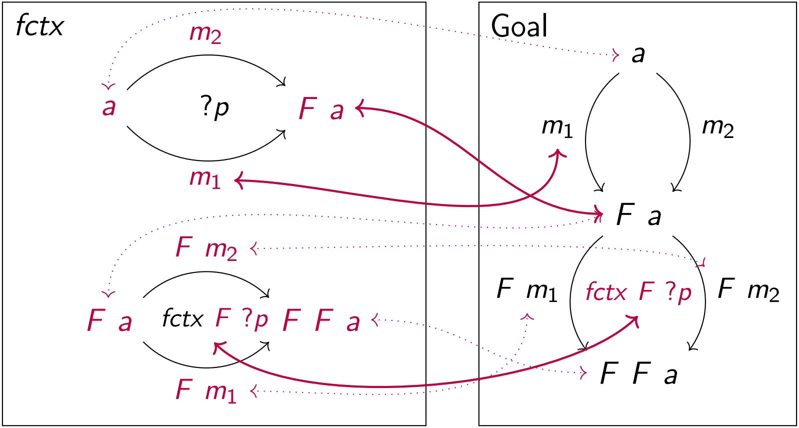
Lemma application



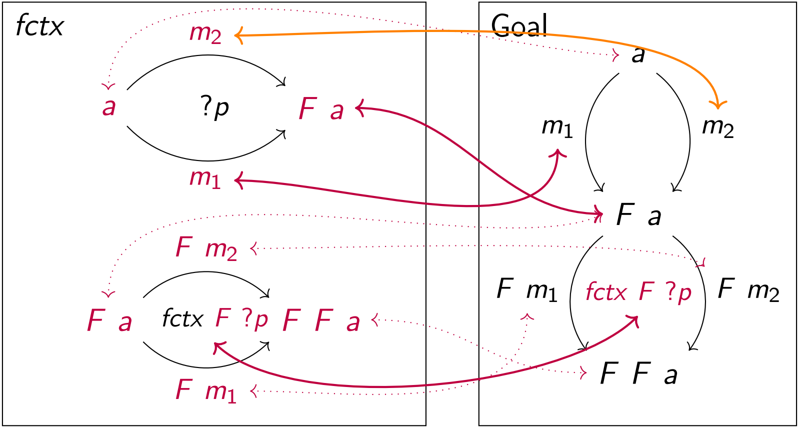
Lemma application



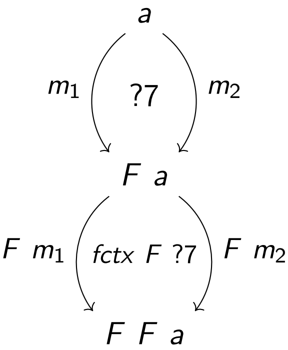
Lemma application



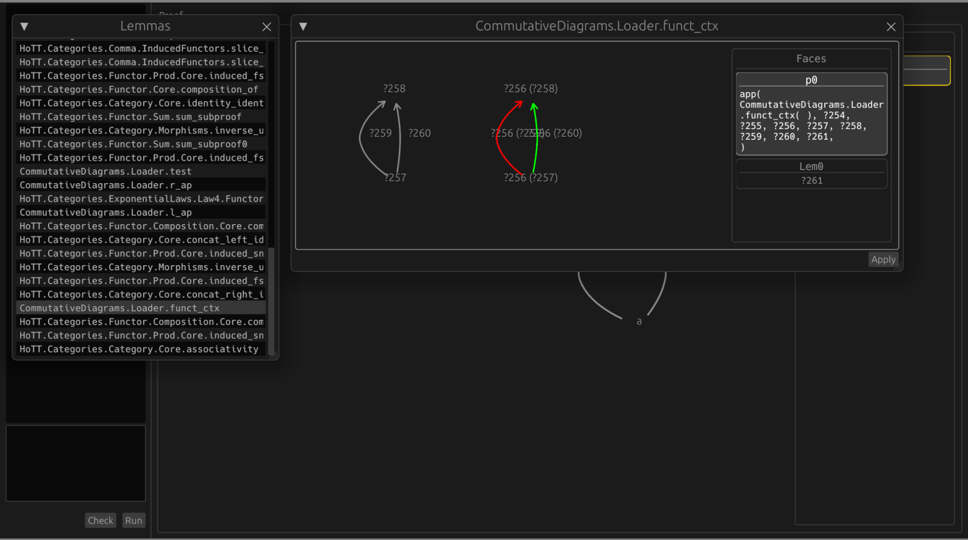
Lemma application

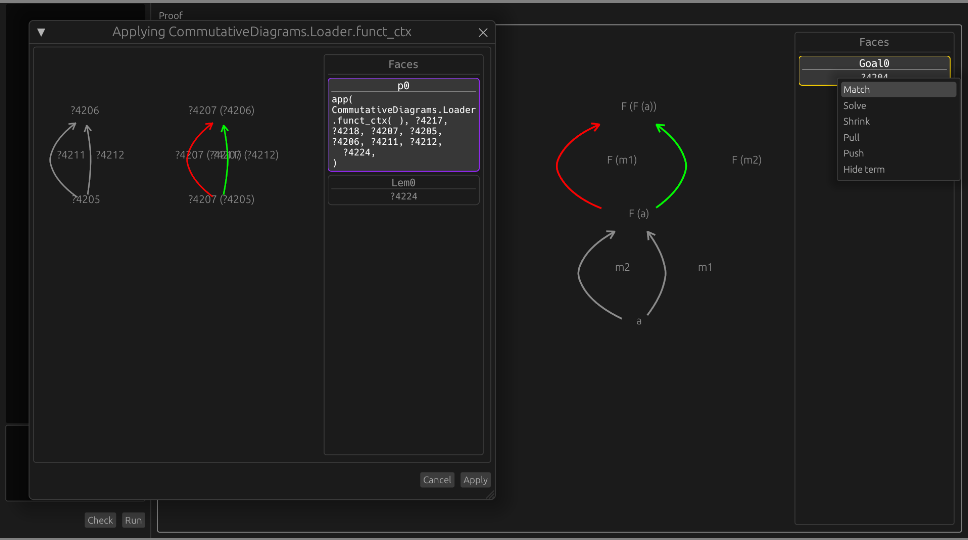


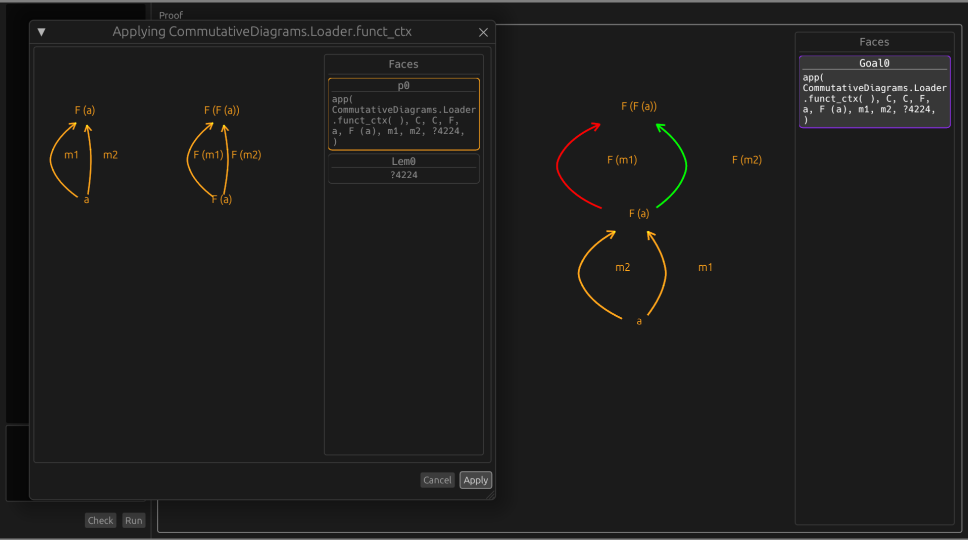
Lemma application : pushout




Proof
Finish
Fail
Open lemmas
Faces
Goal0
?4204
F (F (a))
F (m1)
F (m2)
F (a)
m2
m1
a
Check
Run


Lemmas
HoTT.Categories.Comma.InducedFunctors.slice_
HoTT.Categories.Comma.InducedFunctors.slice_
HoTT.Categories.Functor.Prod.Core.induced_fs
HoTT.Categories.Functor.Core.composition_of
HoTT.Categories.Category.Core.identity_ident
HoTT.Categories.Functor.Sum.sum_subproof
HoTT.Categories.Category.Morphisms.inverse_u
HoTT.Categories.Functor.Sum.sum_subproof0
HoTT.Categories.Functor.Prod.Core.induced_fs
CommutativeDiagrams.Loader.test
CommutativeDiagrams.Loader.r_ap
HoTT.Categories.ExponentialLaws.Law4.Functor
CommutativeDiagrams.Loader.l_ap
HoTT.Categories.Functor.Composition.Core.com
HoTT.Categories.Category.Core.concat_left_id
HoTT.Categories.Functor.Prod.Core.induced_sn
HoTT.Categories.Category.Morphisms.inverse_u
HoTT.Categories.Functor.Prod.Core.induced_fs
HoTT.Categories.Category.Core.concat_right_i
CommutativeDiagrams.Loader.funct_ctx
HoTT.Categories.Functor.Composition.Core.com
HoTT.Categories.Functor.Prod.Core.induced_sn
HoTT.Categories.Category.Core.associativity
CommutativeDiagrams.Loader.funct_ctx
?258
?259
?260
?257
?256 (?258)
?256 (?259)
?256 (?260)
?256 (?257)
Faces
p0
app(
CommutativeDiagrams.Loader
.funct_ctx( ), ?254,
?255, ?256, ?257, ?258,
?259, ?260, ?261,
)
Lem0
?261
Apply
a
Check
Run

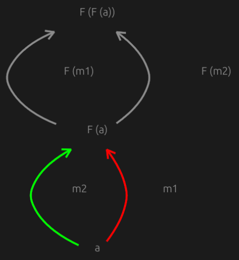

Proof
Applying CommutativeDiagrams.Loader.funct_ctx
?4206
?4211
?4212
?4205
?4207 (?4206)
?4207 (?4211)
(?4212)
?4207 (?4205)
Faces
p0
app(
CommutativeDiagrams.Loader
.funct_ctx( ), ?4217,
?4218, ?4207, ?4205,
?4206, ?4211, ?4212,
?4224,
)
Lem0
?4224
Cancel
Apply
Check
Run
F (F (a))
F (m1)
F (m2)
F (a)
m2
m1
a
Faces
Goal0
Match
Solve
Shrink
Pull
Push
Hide term


Proof
Applying CommutativeDiagrams.Loader.funct_ctx
F (a)
m1
m2
a
F (F (a))
F (m1)
F (m2)
F (a)
Faces
p0
app(
CommutativeDiagrams.Loader
.funct_ctx( ), C, C, F,
a, F (a), m1, m2, ?4224,
)
Lem0
?4224
Cancel
Apply
Check
Run
F (F (a))
F (m1)
F (m2)
F (a)
m2
m1
a
Faces
Goal0
app(
CommutativeDiagrams.Loader
.funct_ctx( ), C, C, F,
a, F (a), m1, m2, ?4224,
)

```
apply CommutativeDiagrams.Loader.funct_ctx x_0:c m:m1 p0:Goal0 x_2:c_0 m_2:m_0 m_0:m2 x:a x_1:c m_1:m
```

Check Run

Proof



Faces

Goal0

```
app(
CommutativeDiagrams.Loader
.funct_ctx( ), C, C, F,
a, F (a), m1, m2, ?4224,
)
```

Goal1

?4224

Conclusion

Partie 4

Future work

- Use ELPI to analyze coq terms

Future work

- Use ELPI to analyze coq terms
- Interactive graph construction

Future work

- Use ELPI to analyze coq terms
- Interactive graph construction
- Semi-interactive graph layouting

Future work

- Use ELPI to analyze coq terms
- Interactive graph construction
- Semi-interactive graph layouting
- Better replayability of proofs

Future work

- Use ELPI to analyze coq terms
- Interactive graph construction
- Semi-interactive graph layouting
- Better replayability of proofs
- **Support additional structure (pushouts, . . .)**

Future work

- Use ELPI to analyze coq terms
- Interactive graph construction
- Semi-interactive graph layouting
- Better replayability of proofs
- **Support additional structure (pushouts, . . .)**
- Support other proof assistants

Thank you for listening

https://github.com/dwarfmaster/commutative-diagrams