

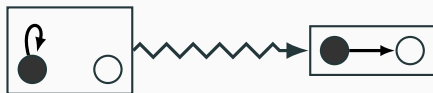
Formalizing adhesive category theory in Rocq

Samuel Arsac, Russ Harmer, Damien Pous

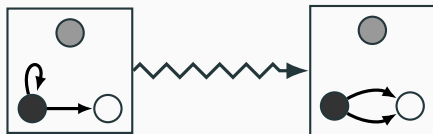
28 janvier 2025

Réécriture de graphes

Règle de réécriture :

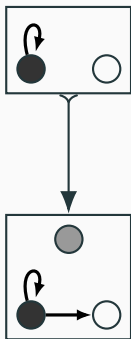


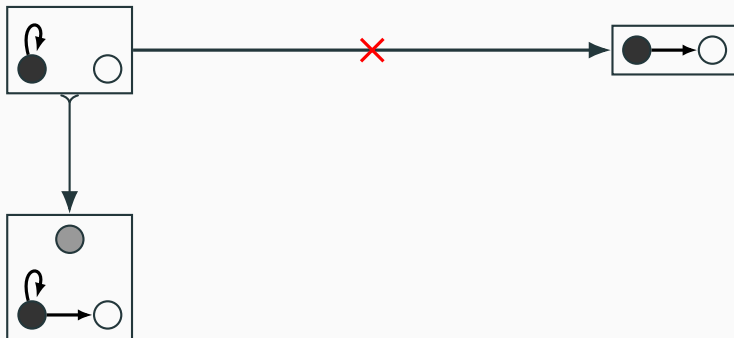
Application :

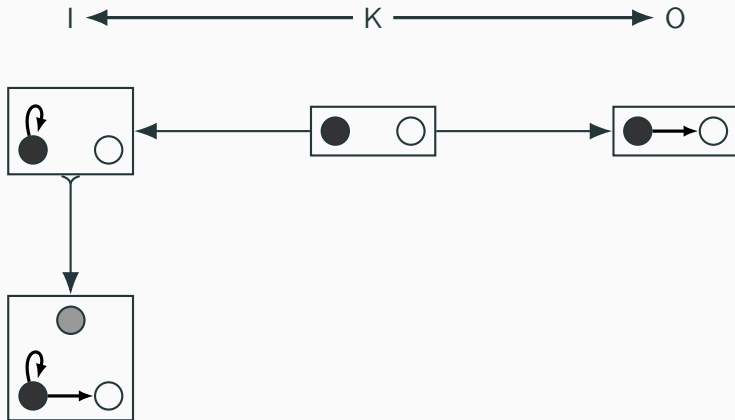


Catégories :

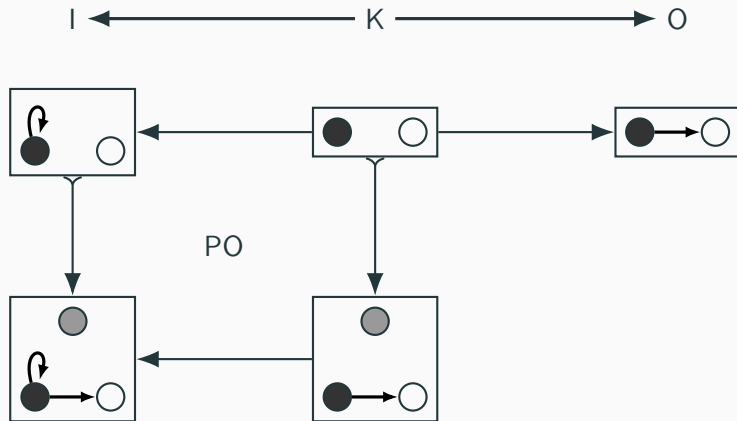
- Objets : graphes
- Morphismes : homomorphismes de graphes



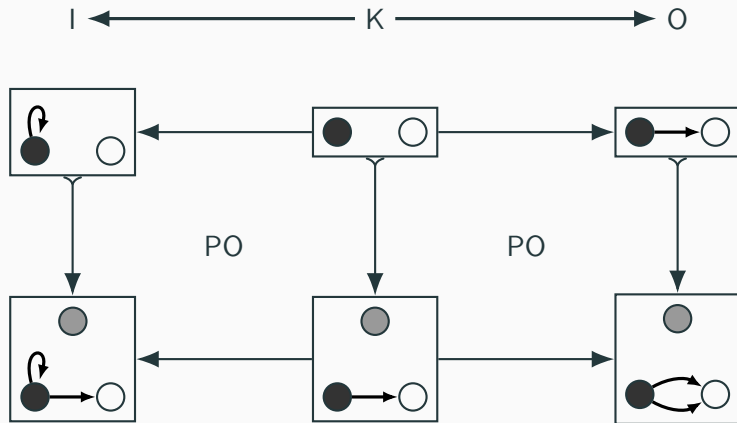




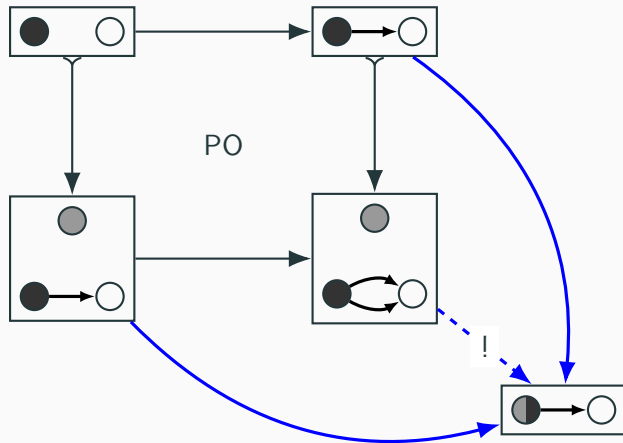
Réécriture DPO



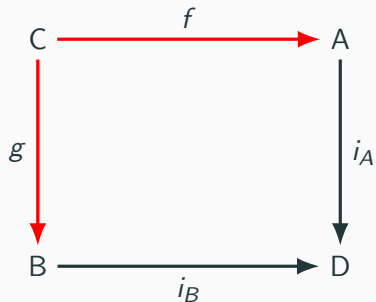
Réécriture DPO



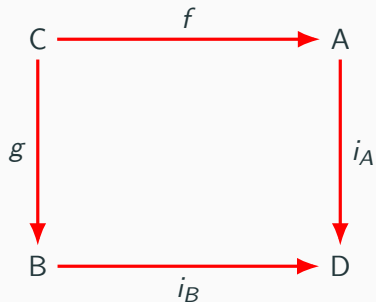
Pushout



Bundling

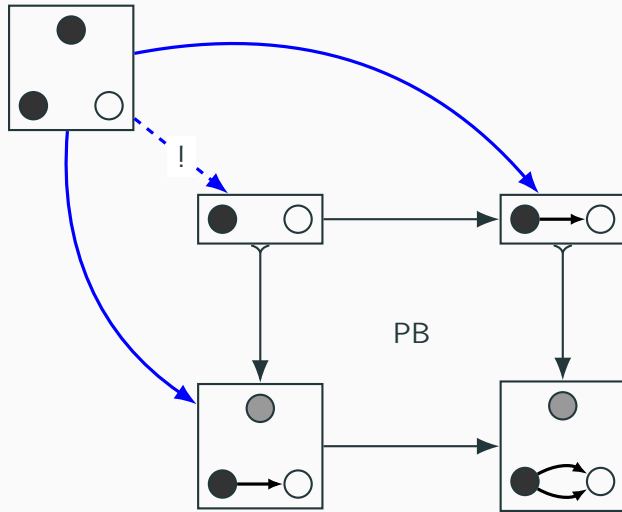


```
Record Pushout ... A B C f g := {  
  D: C;  
  iA: A  $\rightsquigarrow$  D;  
  iB: B  $\rightsquigarrow$  D;  
  pushout_commutates: ...;  
  ump_pushouts: ...  
}.
```



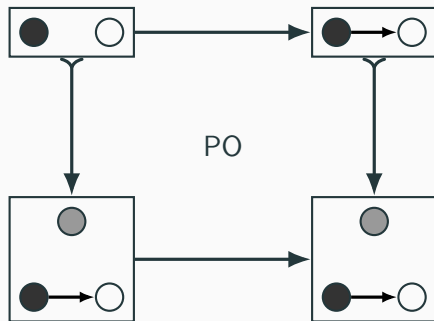
```
Record Pushout ... A B C D
  f g iA iB := {
    pushout_commutates: ...;
    ump_pushouts: ...
  }.
```

Graph vs SGraph

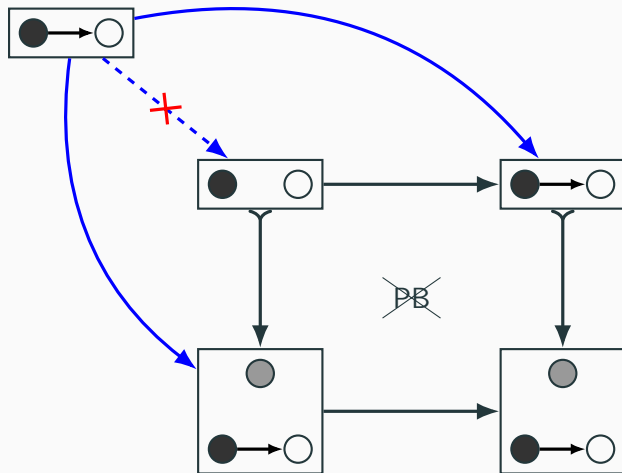


Graph vs SGraph

On obtient un pushout différent dans SGraph.



Graph vs SGraph



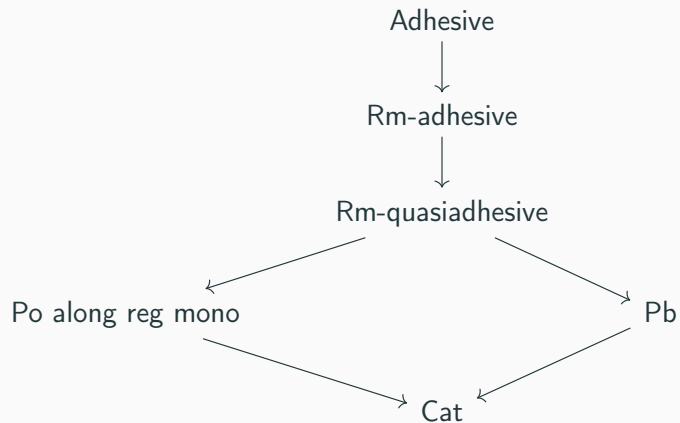
Hierarchie des catégories adhésives

Ces différences sont abstraites en trois types de catégories différentes, principalement en fonction des propriétés vérifiées par les pushouts :

- rm-quasiadhesive (ex : SGraph),
- rm-adhesive ,
- adhesive (ex : Graph , Sets , ...).

Chacune de ces définitions possède plusieurs caractérisations.

Hiérarchie des catégories adhésives



Utilisation de Hierarchy Builder (MathComp).

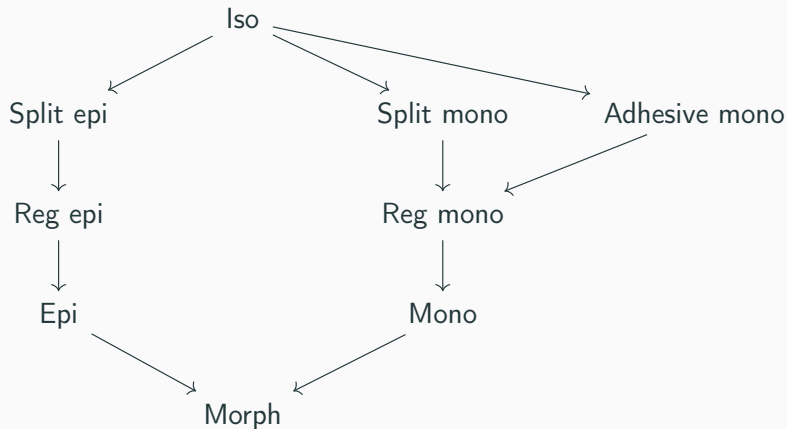
- Ajout de structures héritant des propriétés d'autres structures.
- Usage transparent des lemmes en cas d'héritage.
- Définition de *factories* pour définir des instances à partir des caractérisations.
- Possibilité d'étendre la hiérarchie.

Formalisation de la hiérarchie des catégories adhésives et de leurs caractérisations.

Bibliothèque de lemmes sur les pullbacks, pushouts, monos réguliers...

Environ 6300 lignes de code.

Définir une hiérarchie pour les morphismes.



Simplifier la manipulation des isomorphismes, pour se rapprocher des preuves sur papier.

Formaliser des résultats sur la réécriture de graphes.