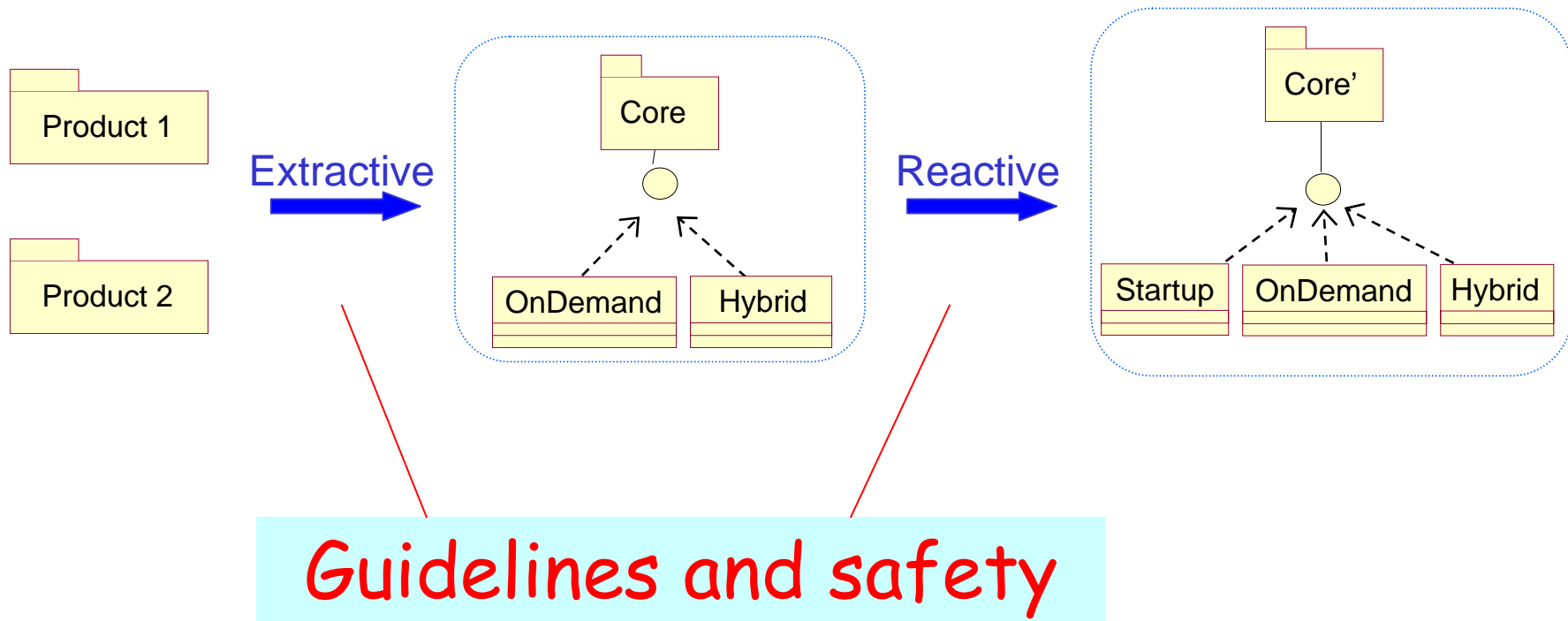


Refactoring Product Lines

V. Alves, R. Gheyi, T. Massoni, U.
Kulesza, Paulo Borba, C. Lucena

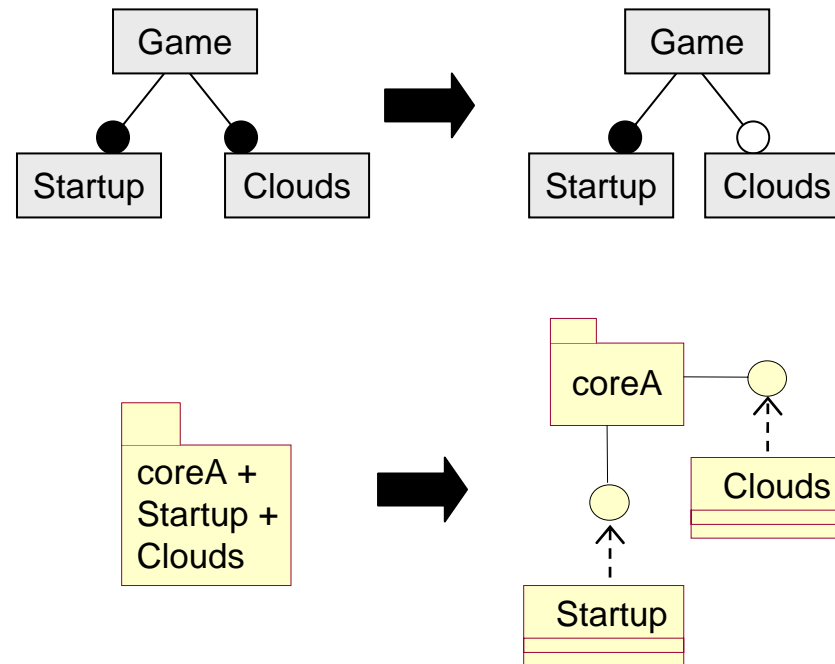
Federal University of Pernambuco
and PUC-Rio

Product line derivation and evolution benefit from refactoring

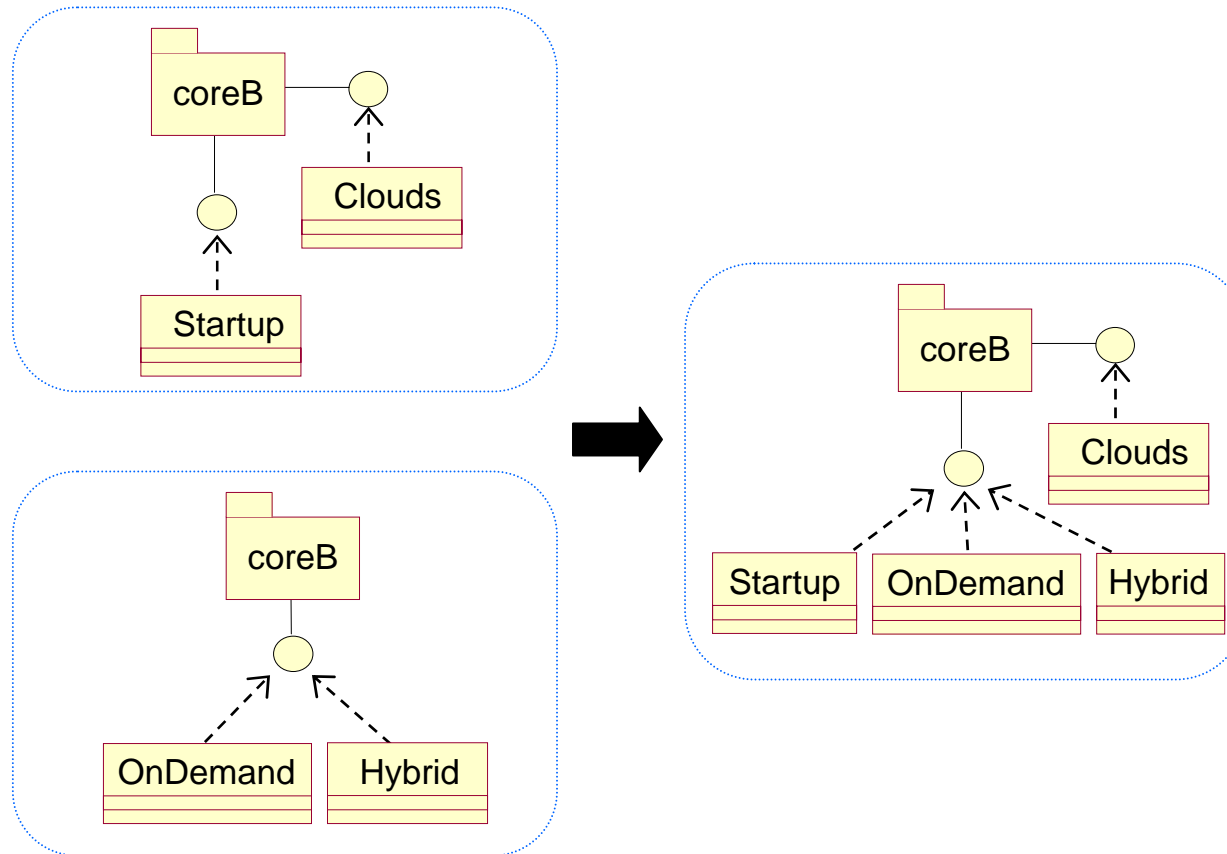


But PL refactoring should go beyond code...

Rain of fire



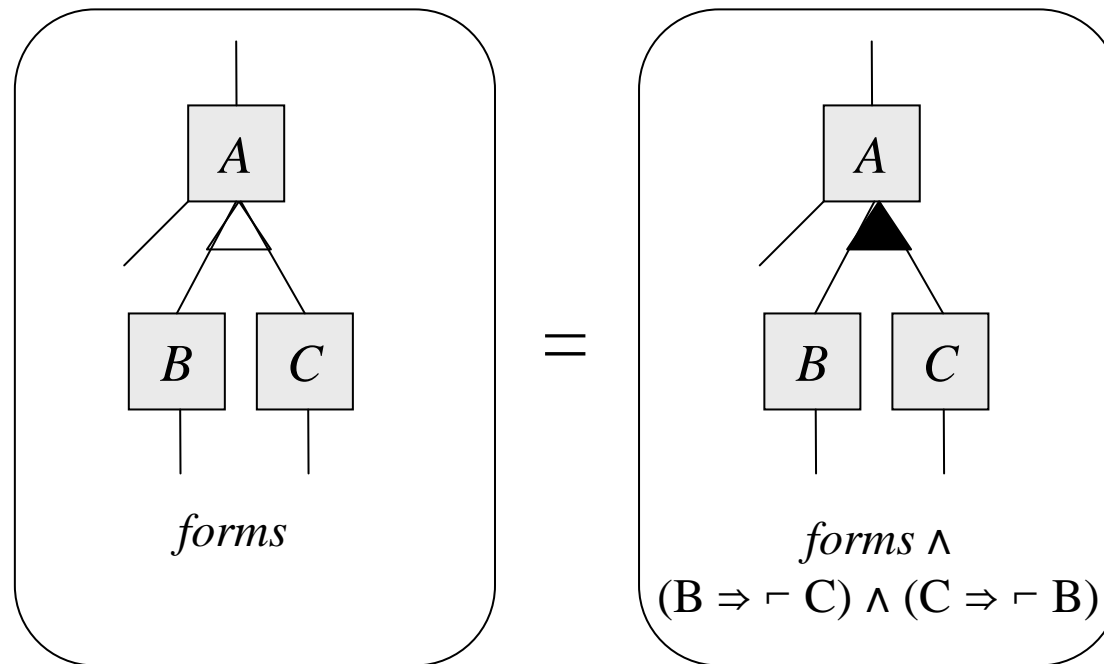
and deal with populations and families too!



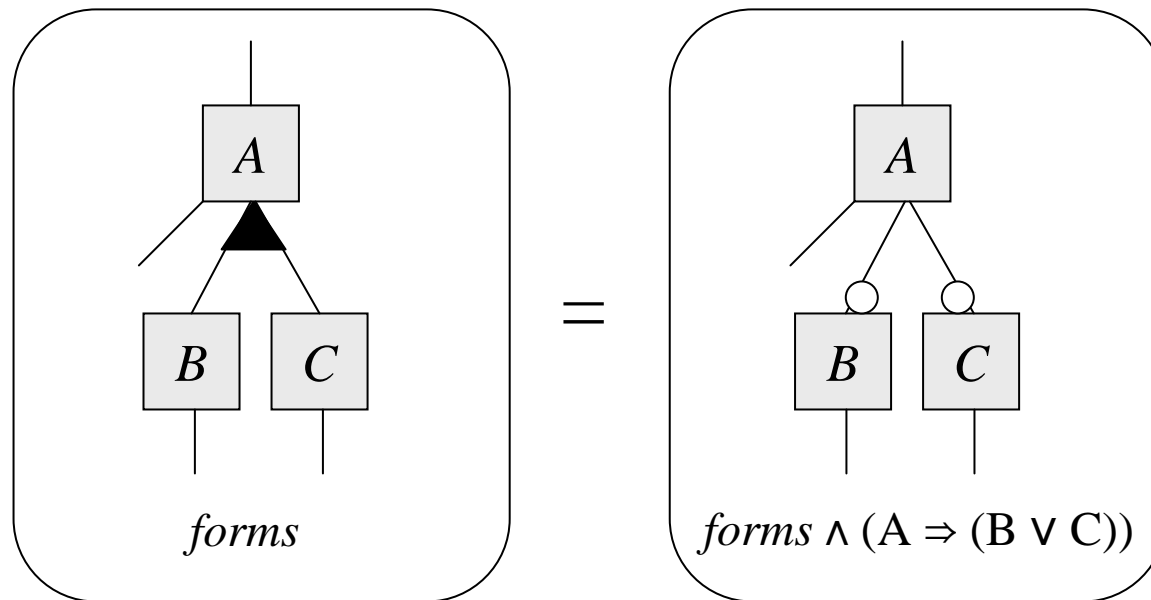
Outline

- Feature model semantics and refactoring
- Feature model refactoring catalog
- Population and family refactoring
- Case study

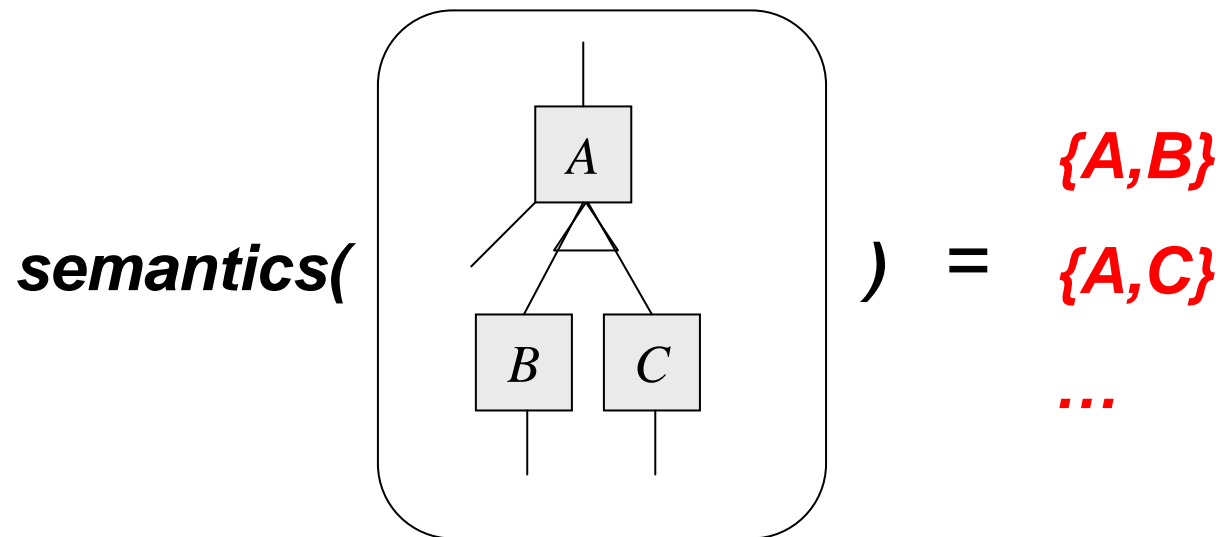
Feature model transformational semantics



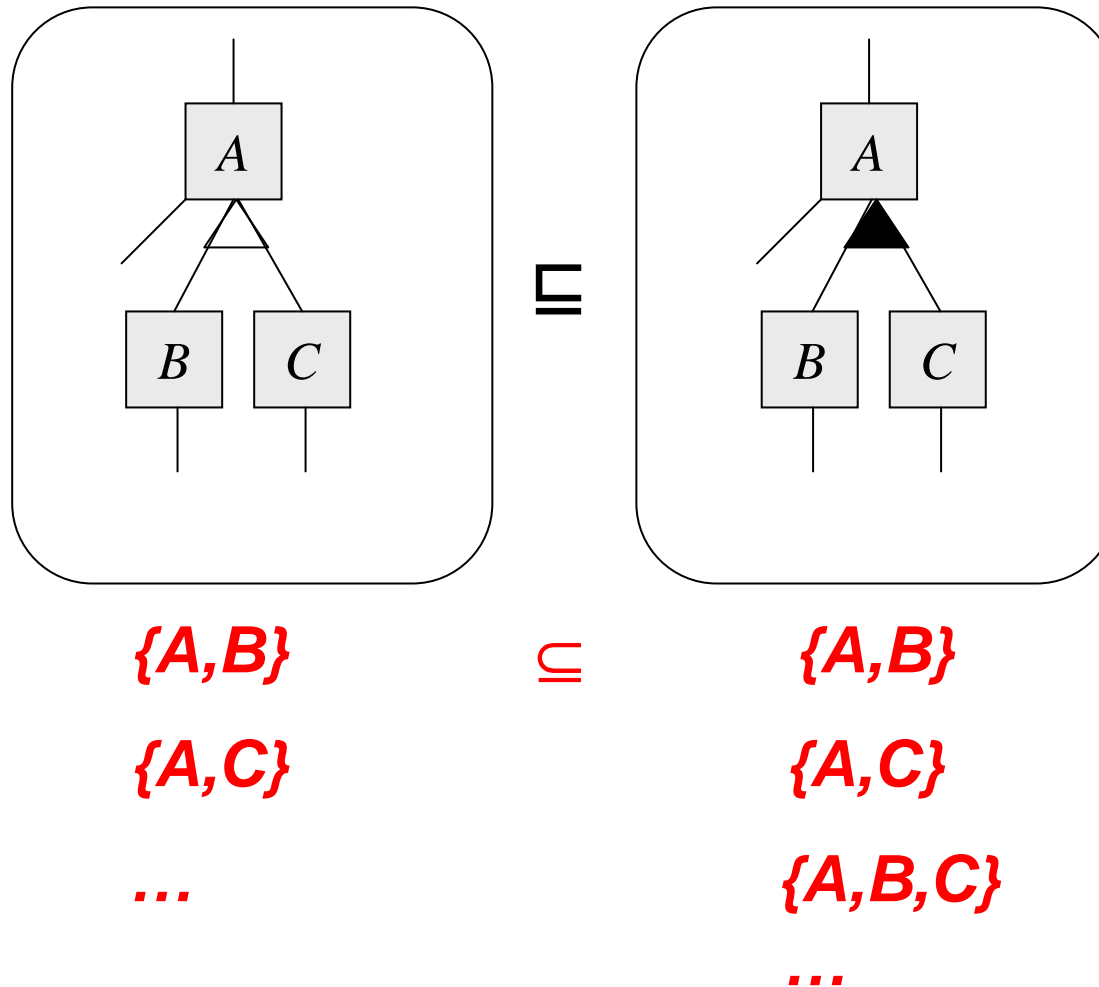
Reduction strategy to features and formulae language



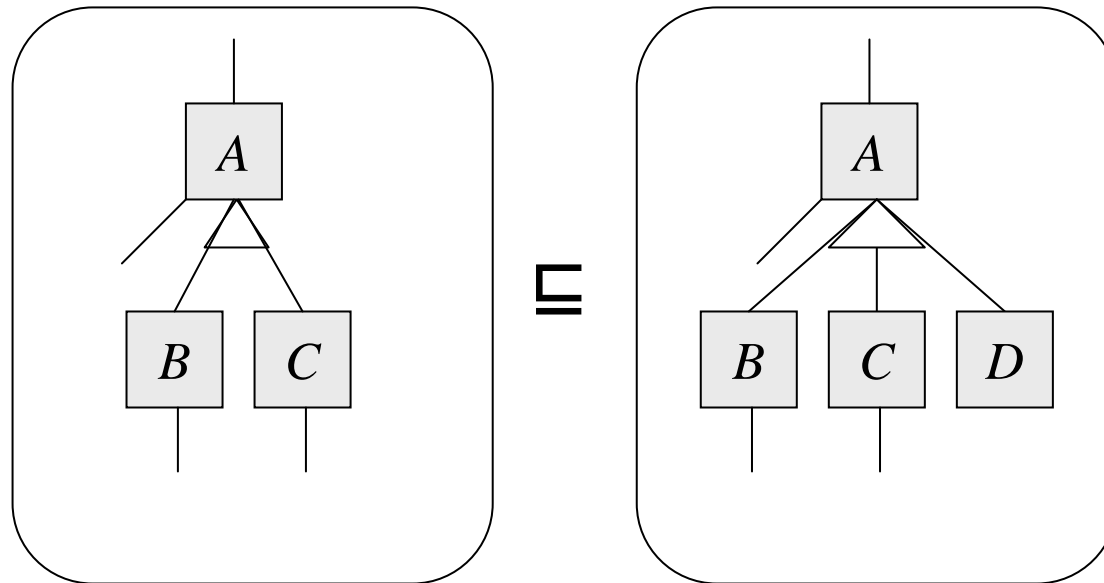
Valid configurations semantics



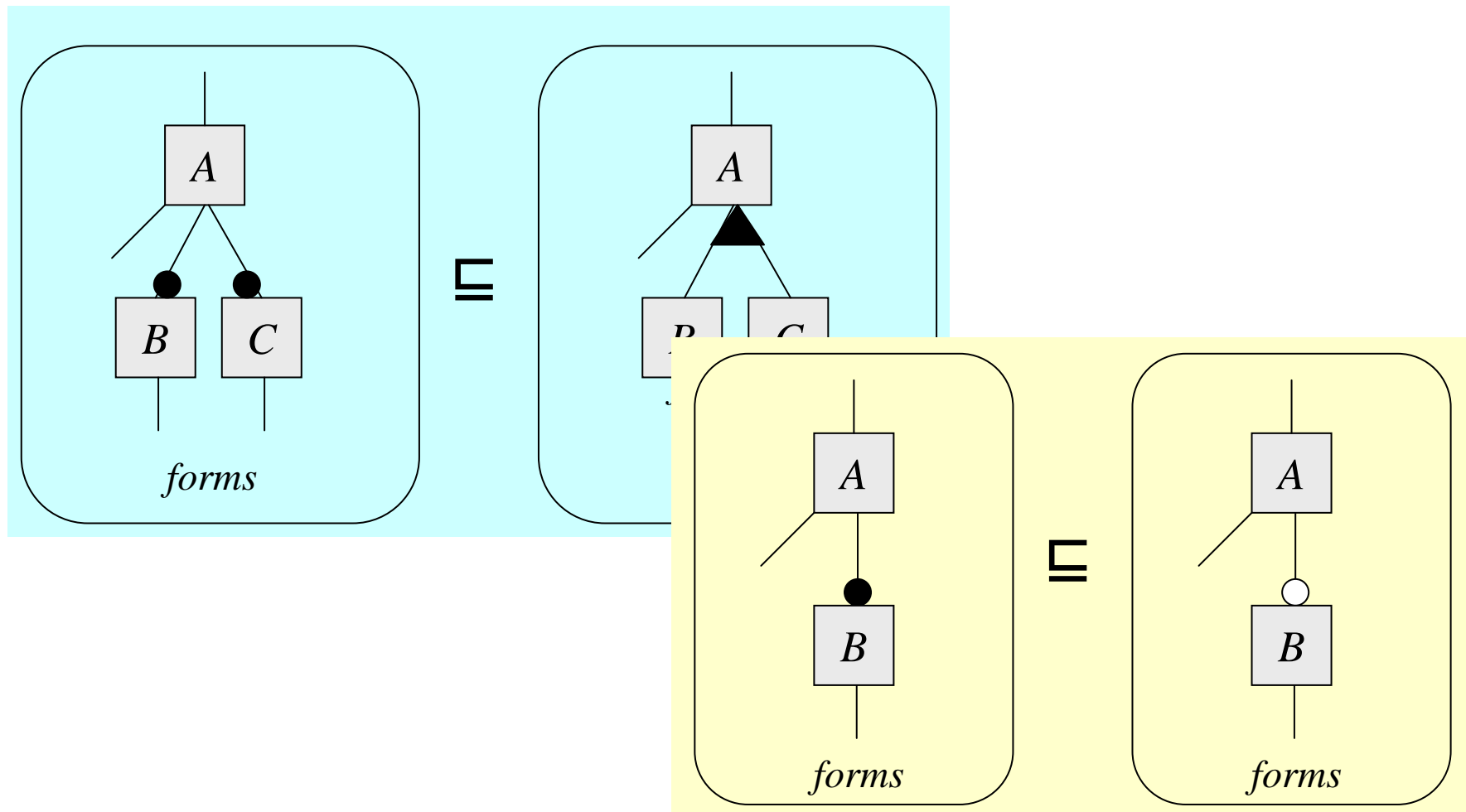
Feature model refactorings as improved configurability



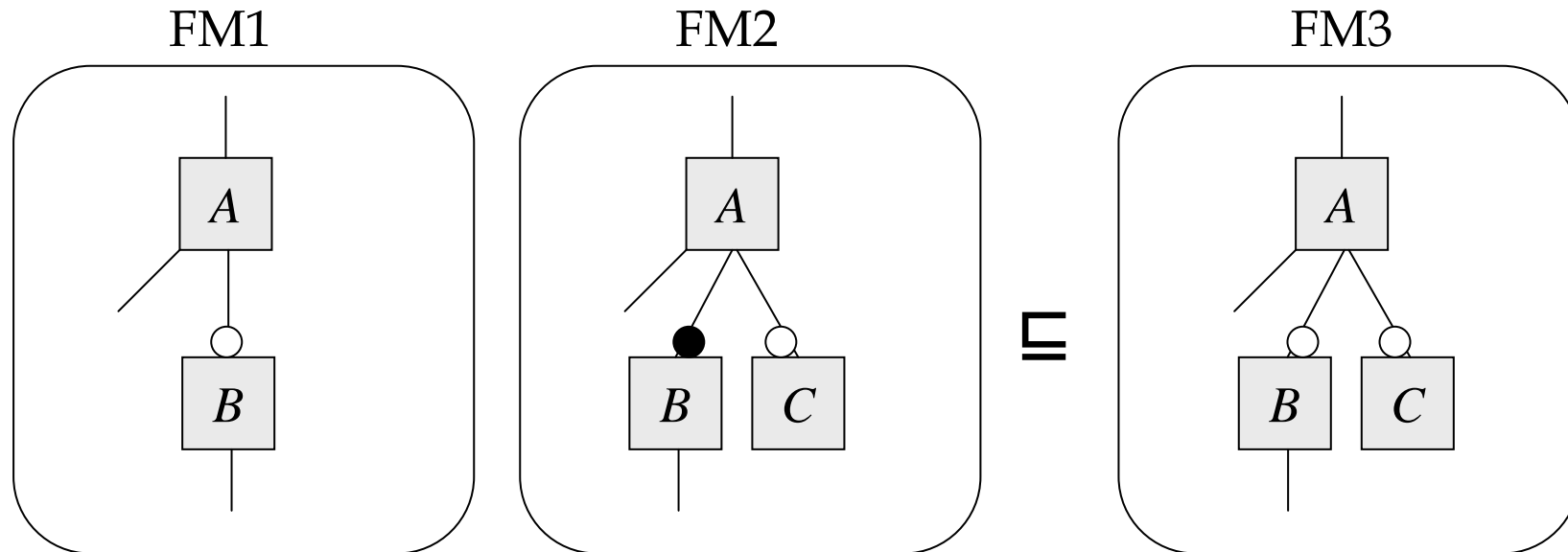
But no need to think about semantics directly: add alternative node



Refactoring catalog

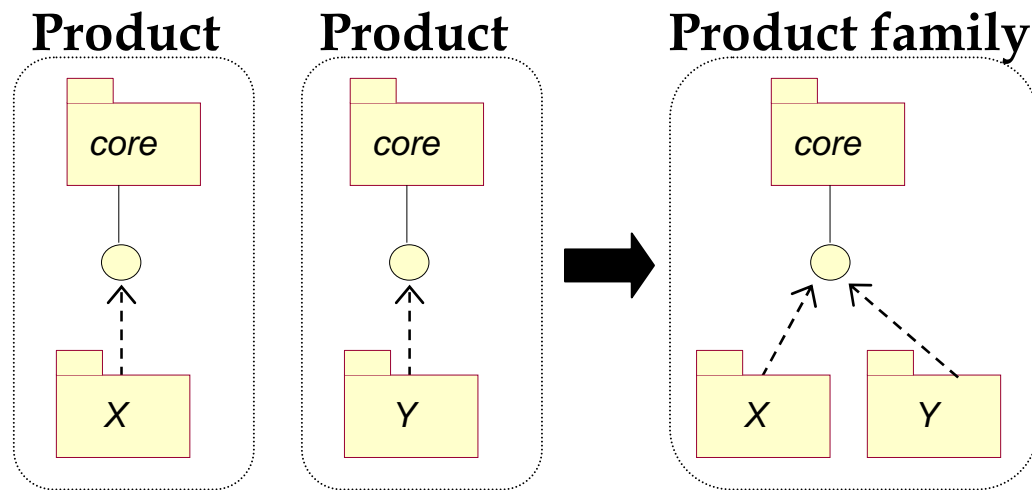


Refactoring populations and families: feature models



$$fm1 \sqcup fm2 \sqsubseteq fm =$$
$$fm1 \sqsubseteq fm \wedge fm2 \sqsubseteq fm$$

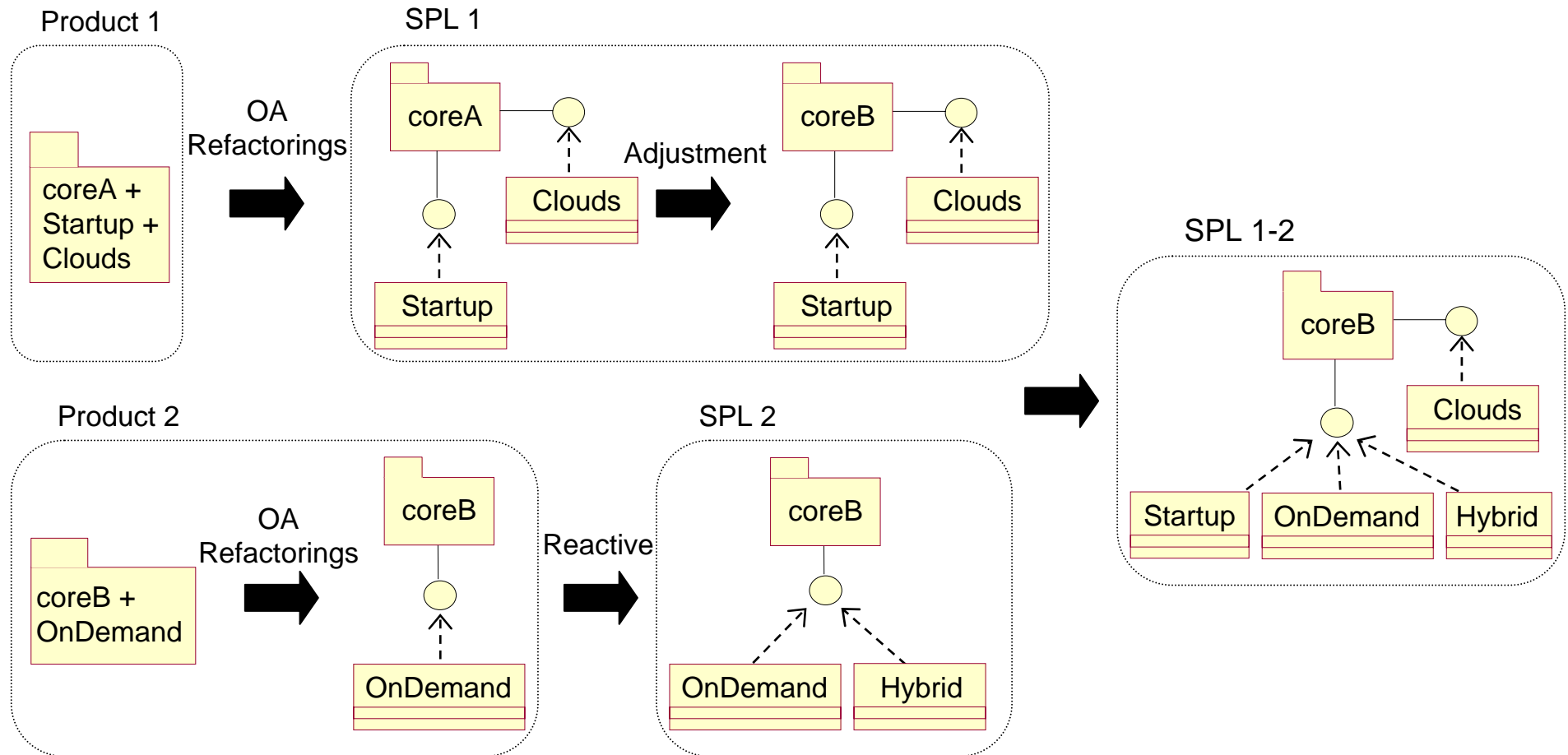
Refactoring populations and families: code



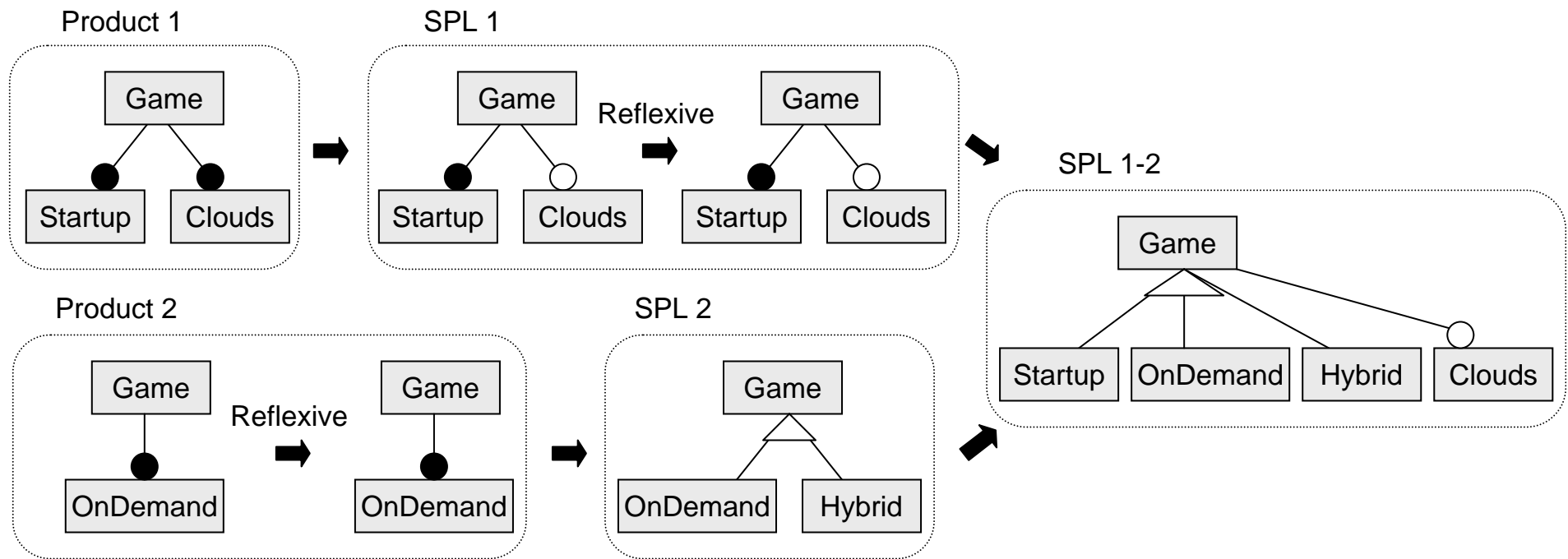
$$p1 \ p2 \sqsubseteq pL =$$

$$\exists p, q \in pL \bullet p1 \sqsubseteq p \wedge p2 \sqsubseteq q$$

Case study: code refactoring



Case study: feature model refactoring



Conclusions

- Extended refactoring notion
 - feature models
 - populations and families
- Formalization
- Refactoring catalog
- Need to deal with
 - configuration knowledge
 - refactorings relationships

Refactoring Product Lines

V. Alves, R. Gheyi, T. Massoni, U.
Kulesza, Paulo Borba, C. Lucena

Federal University of Pernambuco
and PUC-Rio