Software product lines

Paulo Borba
Informatics Center
Federal University of Pernambuco

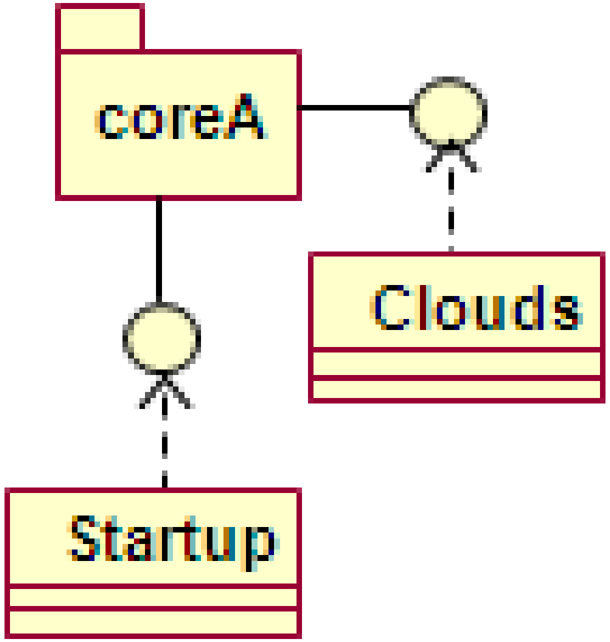
Models for software product lines

Paulo Borba

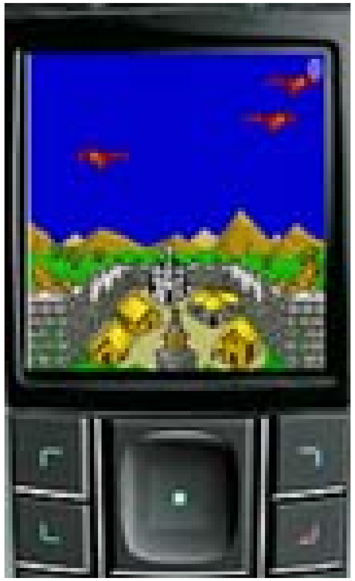
Informatics Center

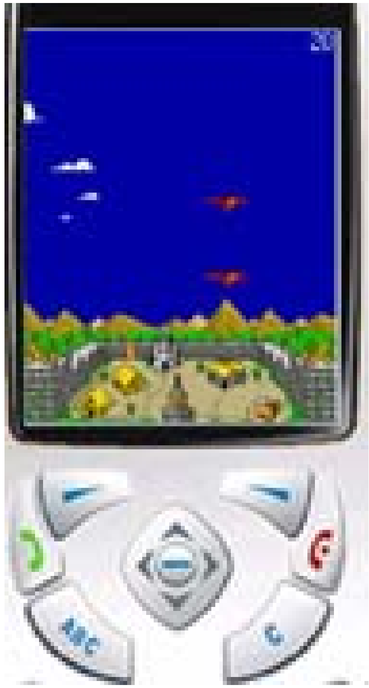
Federal University of Pernambuco

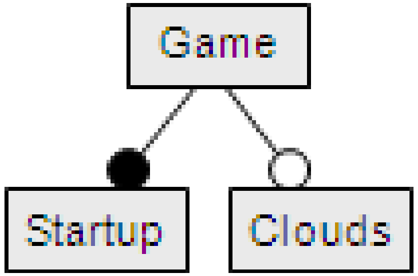
The code shows possible variation points and variations...



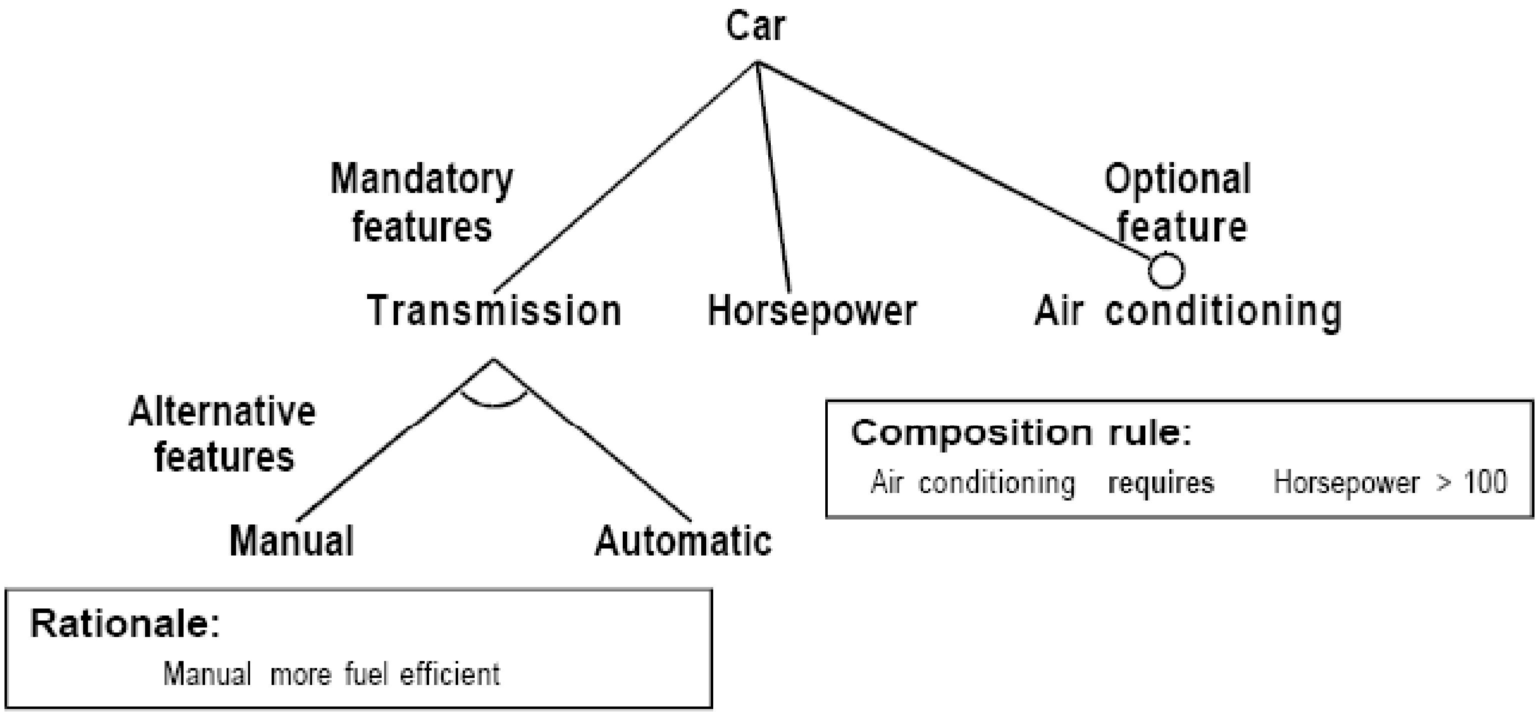
but feature models show more...







and help visualize and manage variations…

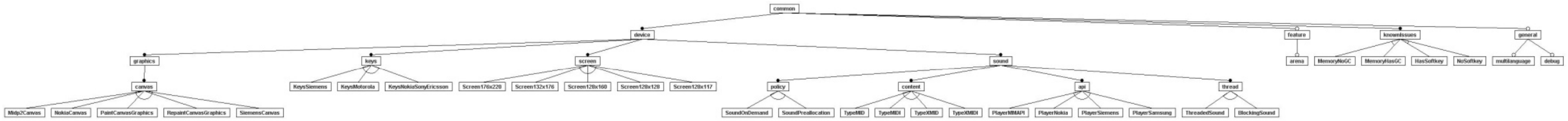


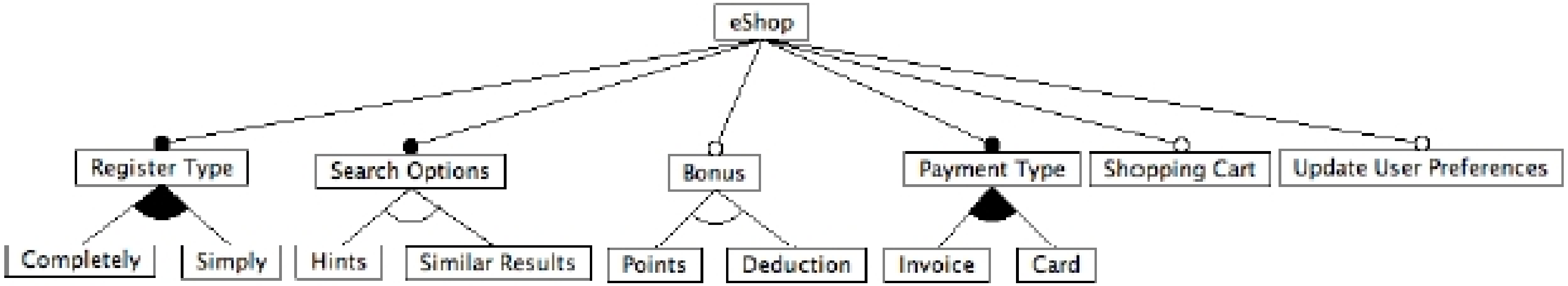
Composition rule:
Air conditioning requires Horsepower > 100

Rationale:
Manual more fuel efficient

From Feature-Oriented Domain Analysis, Kang et al 19

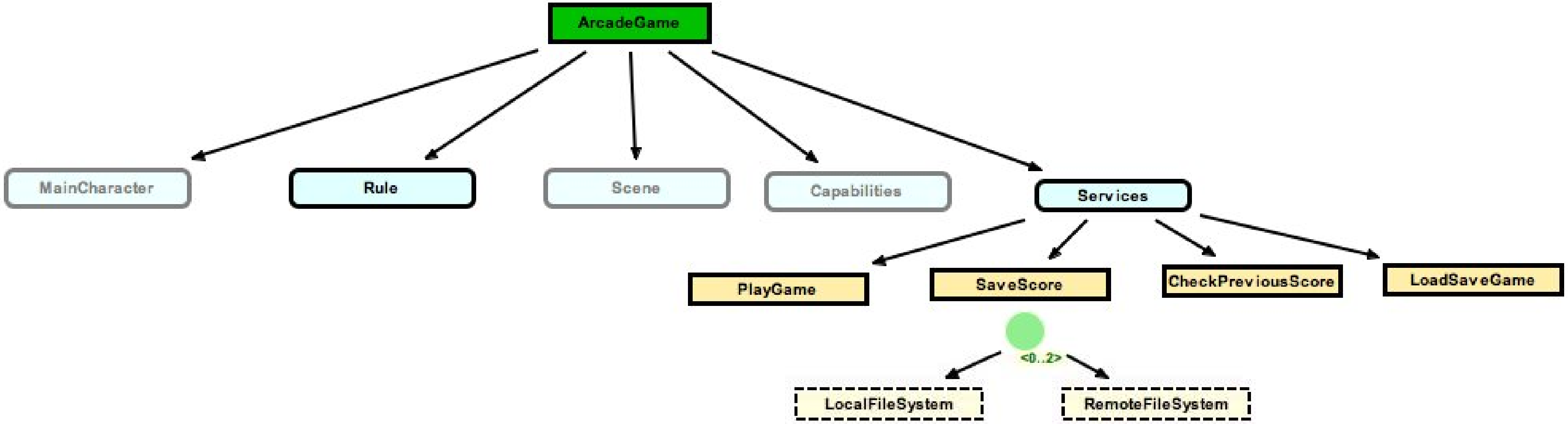
for non trivial product lines





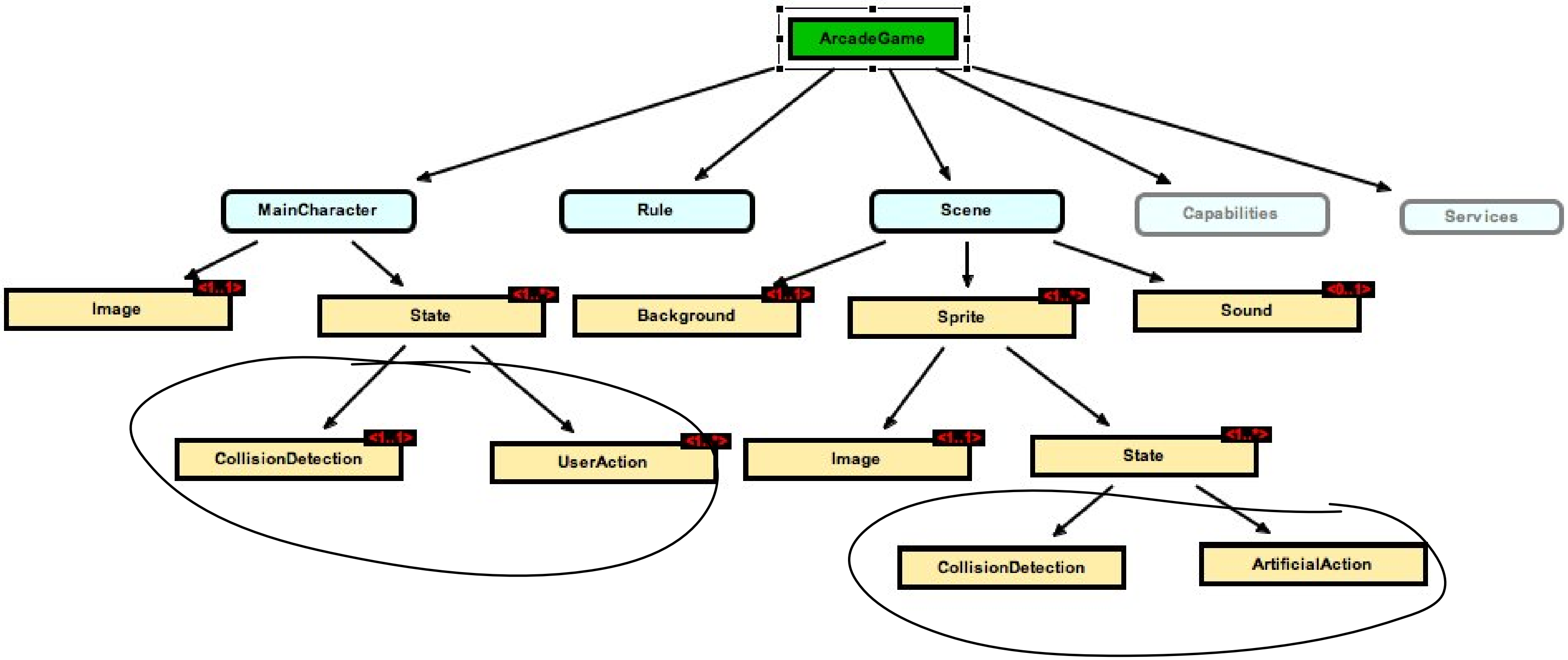
Based on http://gp.uwaterloo.ca/files/2006-lau-masc-thesis.pdf

Cardinalities and attributes yield the power of DSLs...



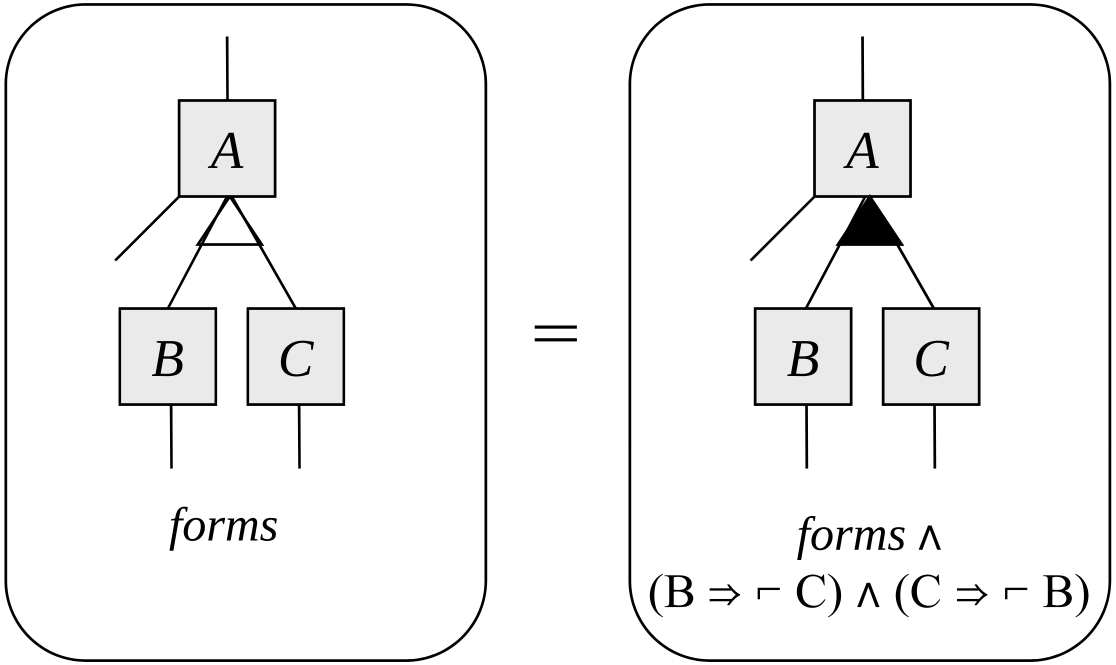
From http://www.sei.cmu.edu/productlines/ppl/

when model is targeted at configuration



From http://www.sei.cmu.edu/productlines/ppl/

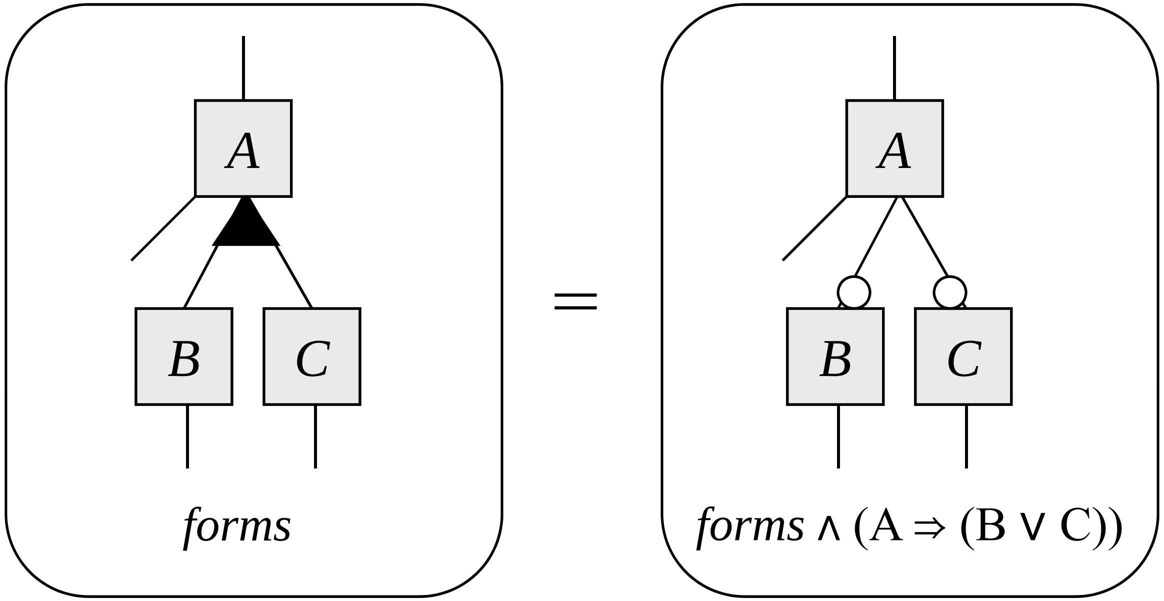
Feature model transformational semantics



forms

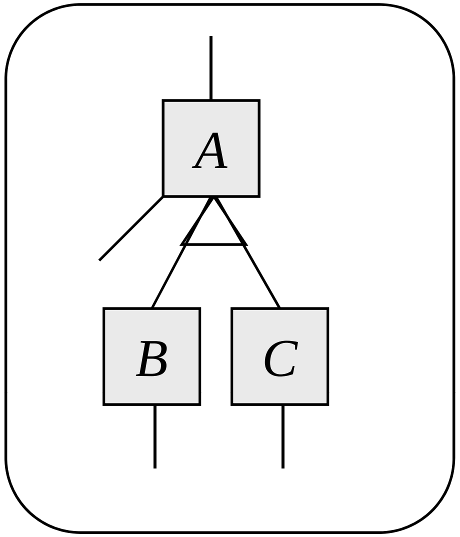
forms ∧
(B ⇒ ⌐ C) ∧ (C ⇒ ⌐ B)

Reduction strategy to features and formulae language



Valid configurations semantics

semantics(

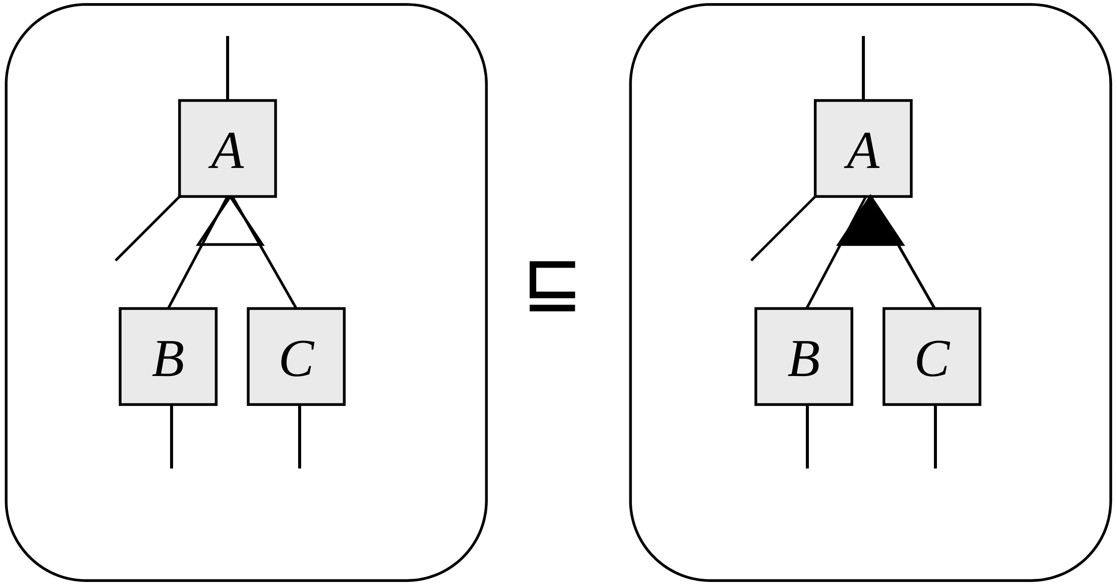


) = {A,B}

{A,C}

...

Feature model refactorings as improved configurability



{A,B}	⊆	{A,B}
{A,C}		{A,C}
…		{A,B,C}
		…

Feature models and configurability improvement

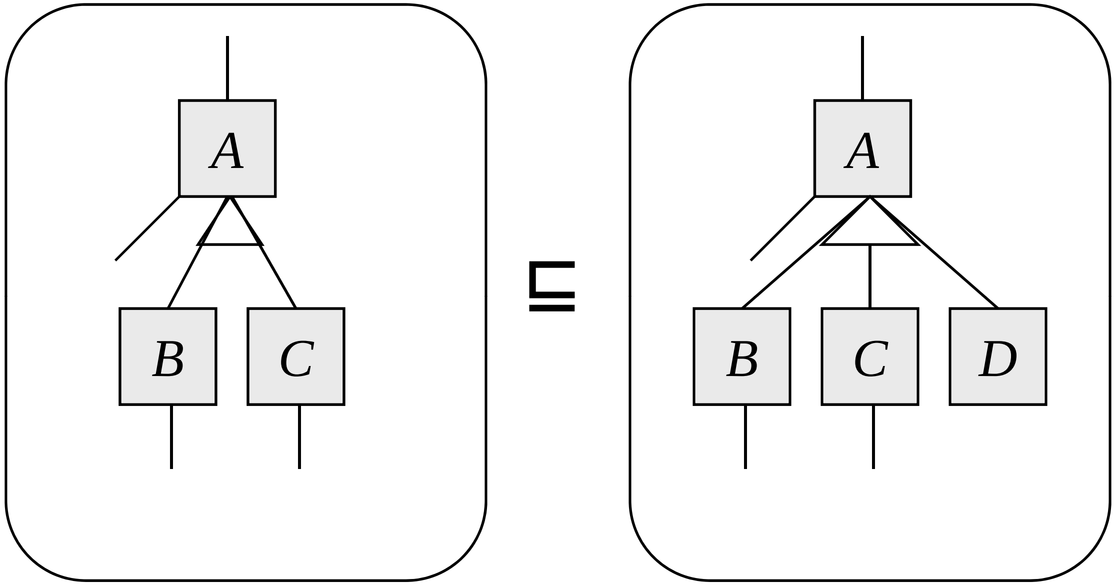
$$\boldsymbol{semantics}(fm) \;=$$

$$\{c{:}Config \mid satConstraints(fm,c)\}$$

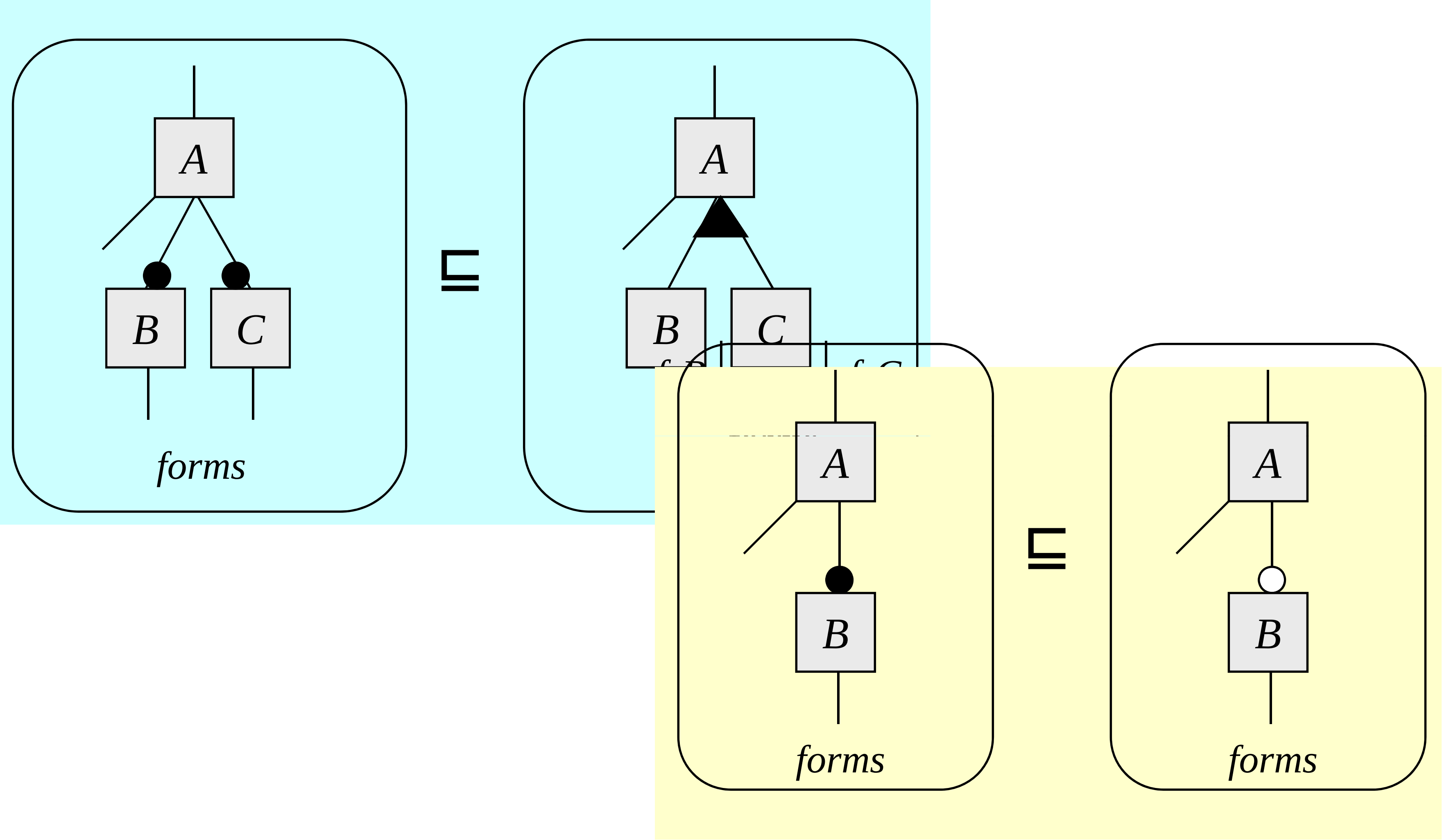
$$fm \sqsubseteq ifm =$$

$$semantics(fm) \subseteq semantics(ifm)$$

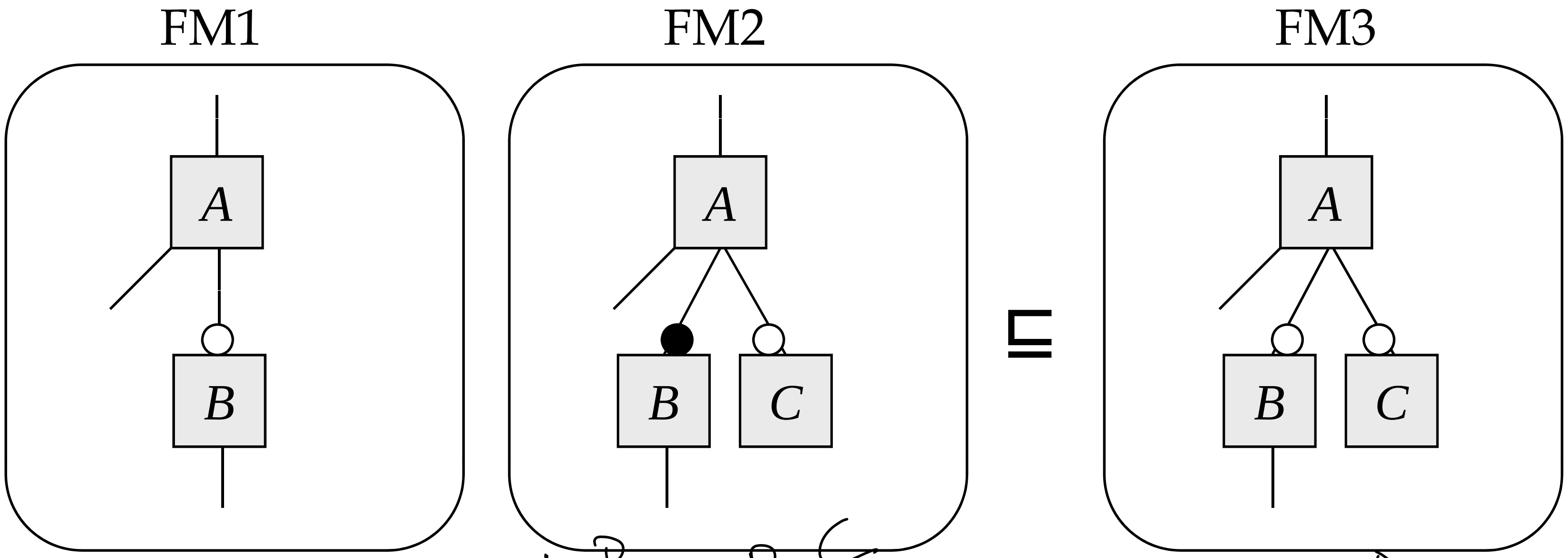
But no need to think about semantics directly: add alternative node



Refactoring catalog



Refactoring populations and families: feature models



A

A,B

A,B A,B,C

A

A,B C

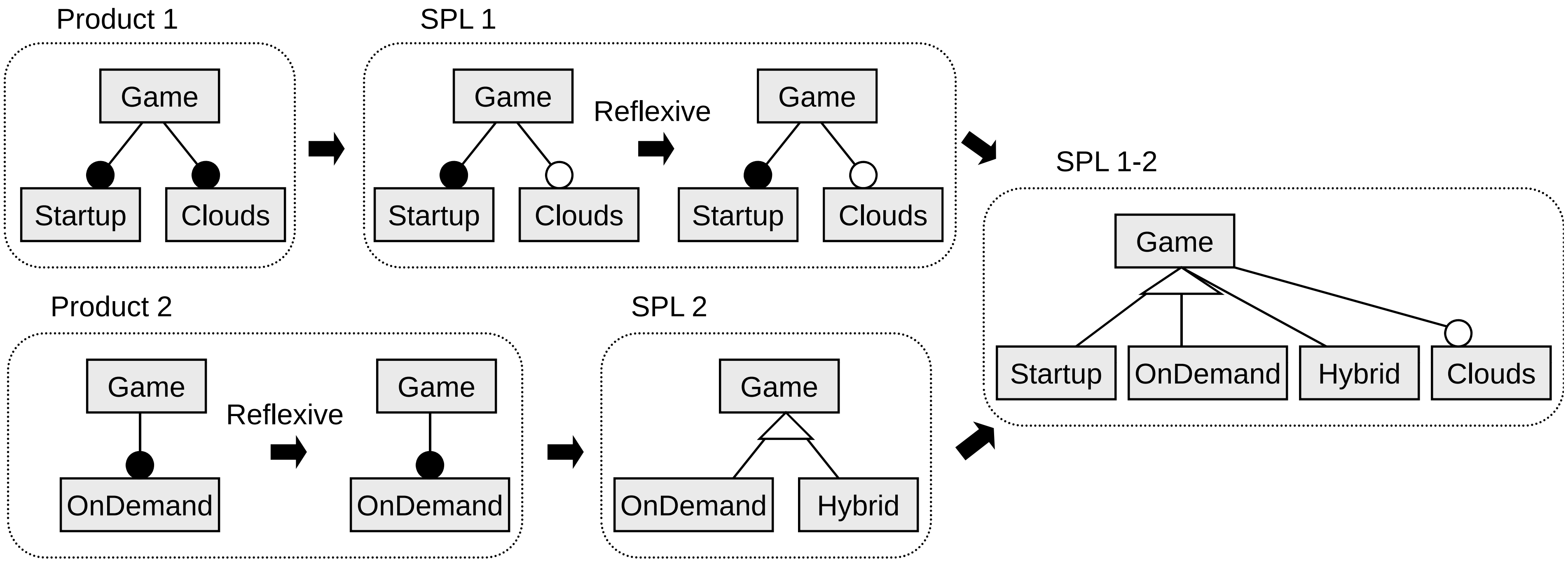
A,

A,B,C

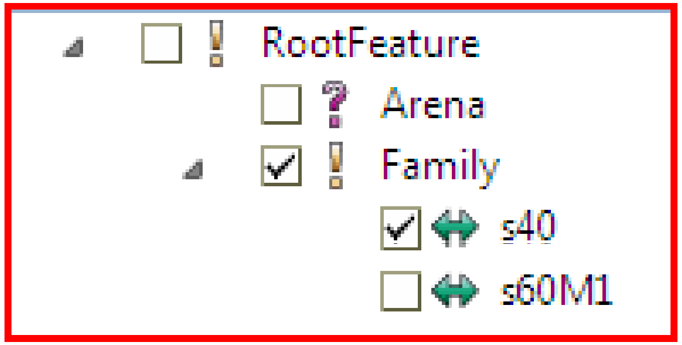
$$fm1\ fm2 \sqsubseteq fm =$$

$$fm1 \sqsubseteq fm \wedge fm2 \sqsubseteq fm$$

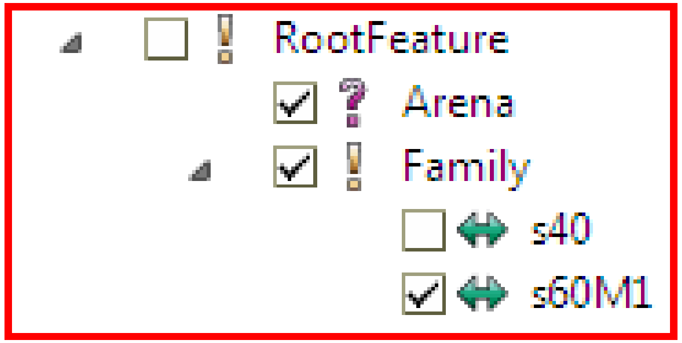
Case study: feature model refactoring



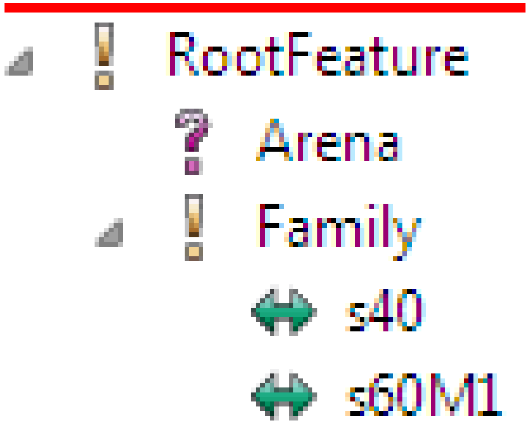
Instance models define products



- ☐ ! RootFeature
  - ☐ ? Arena
  - ☑ ! Family
    - ☑ ⇔ s40
    - ☐ ⇔ s60M1



- ☐ ! RootFeature
  - ☑ ? Arena
  - ☑ ! Family
    - ☐ ⇔ s40
    - ☑ ⇔ s60M1



- ! RootFeature
  - ? Arena
  - ! Family
    - ⇔ s40
    - ⇔ s60M1

Feature model instantiation might be…

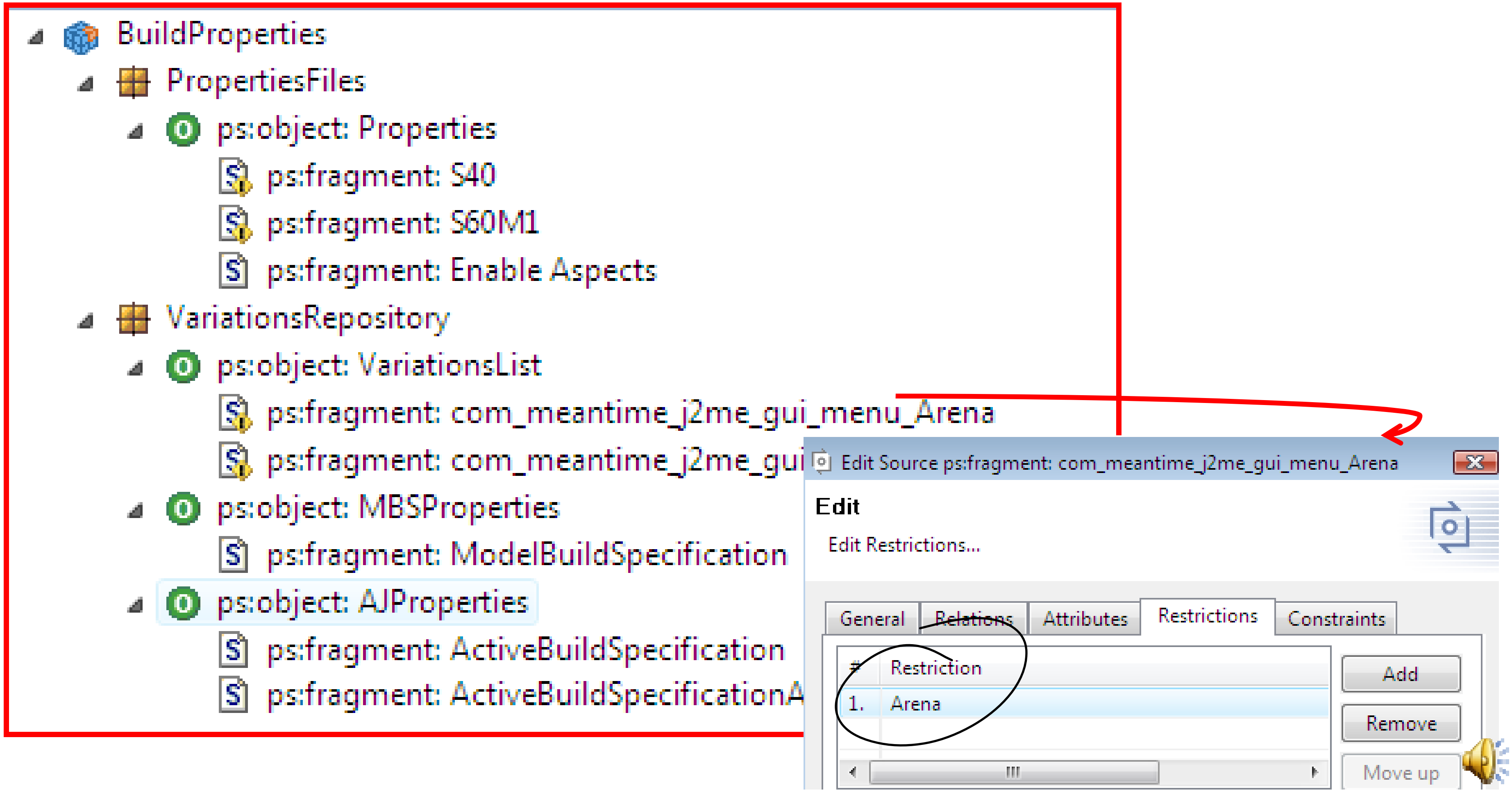
- Manual, defined by developers
- Automatic, inferred from context

Instance models as equations

```
p1 = base + s40 + Screen128x128

p2 =   base + Arena + s60M1
     + Screen128x128

p3 = base + s60M1 + Screen128x160
```

Doesn't work for feature interaction

Configuration knowledge builds products



```
BuildProperties
  PropertiesFiles
    ps:object: Properties
      ps:fragment: S40
      ps:fragment: S60M1
      ps:fragment: Enable Aspects
  VariationsRepository
    ps:object: VariationsList
      ps:fragment: com_meantime_j2me_gui_menu_Arena
      ps:fragment: com_meantime_j2me_gui
    ps:object: MBSProperties
      ps:fragment: ModelBuildSpecification
    ps:object: AJProperties
      ps:fragment: ActiveBuildSpecification
      ps:fragment: ActiveBuildSpecificationA
```

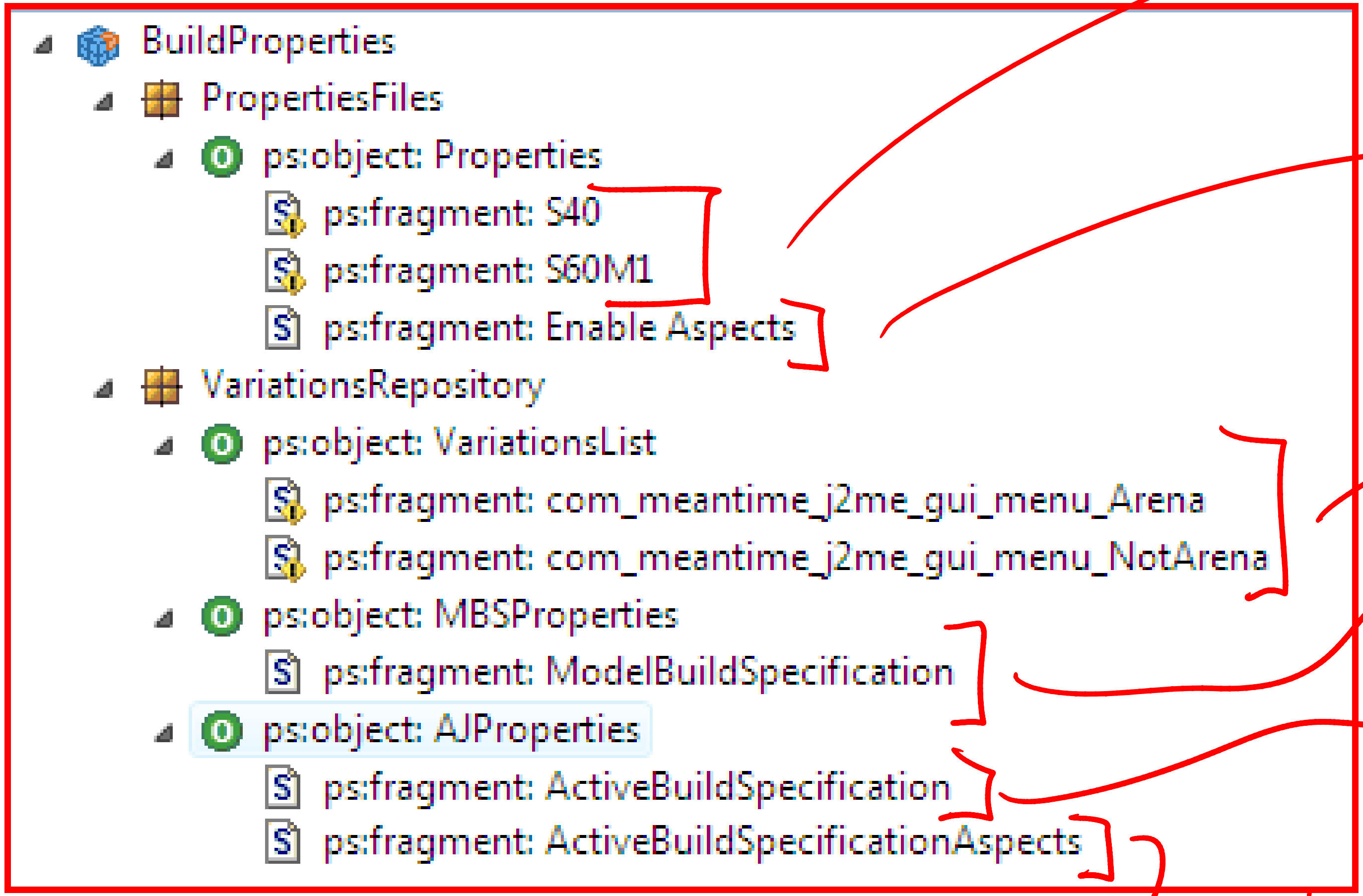
Edit Source ps:fragment: com_meantime_j2me_gui_menu_Arena

Edit

Edit Restrictions...

#	Restriction
1.	Arena

In this case, property files...



- BuildProperties
  - PropertiesFiles
    - ps:object: Properties
      - ps:fragment: S40
      - ps:fragment: S60M1
      - ps:fragment: Enable Aspects
  - VariationsRepository
    - ps:object: VariationsList
      - ps:fragment: com_meantime_j2me_gui_menu_Arena
      - ps:fragment: com_meantime_j2me_gui_menu_NotArena
    - ps:object: MBSProperties
      - ps:fragment: ModelBuildSpecification
    - ps:object: AJProperties
      - ps:fragment: ActiveBuildSpecification
      - ps:fragment: ActiveBuildSpecificationAspects

properties header

properties end

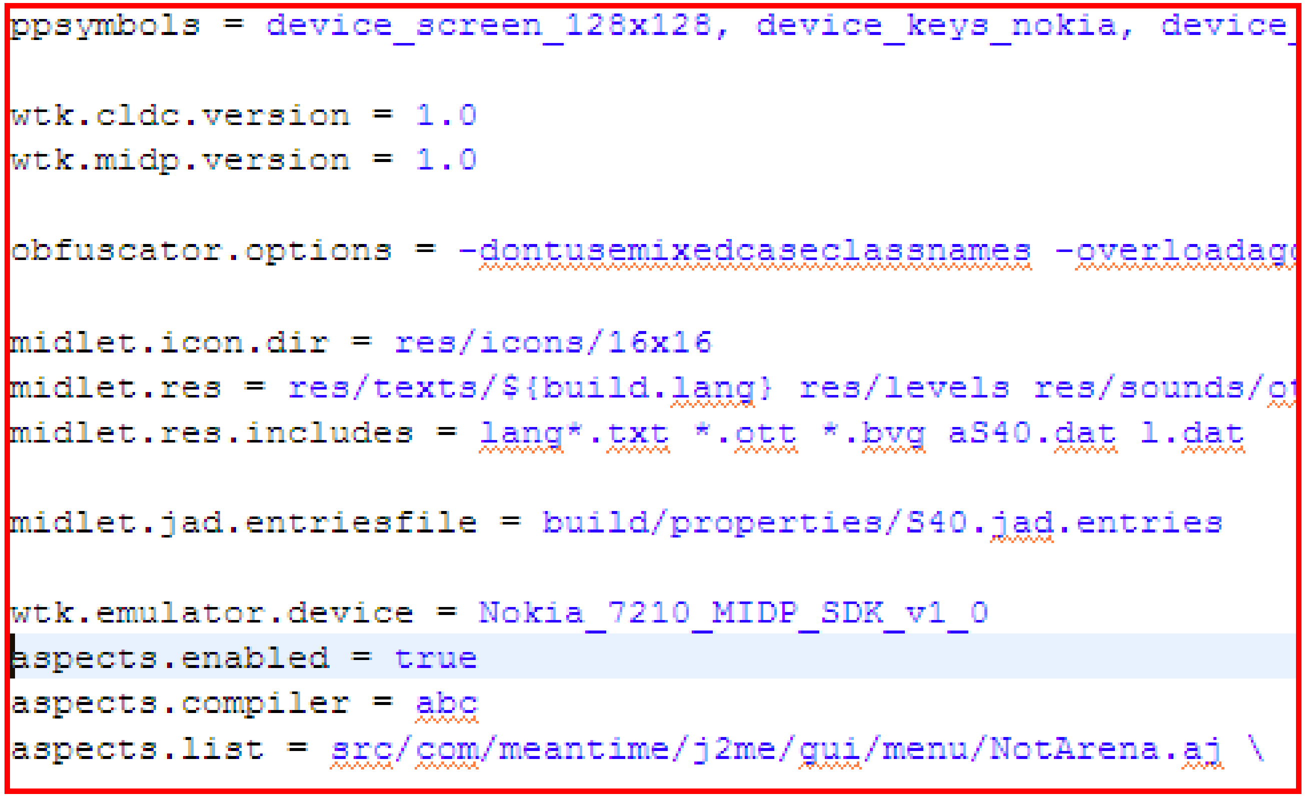
aj list

properties <<aj list

ajproperties header

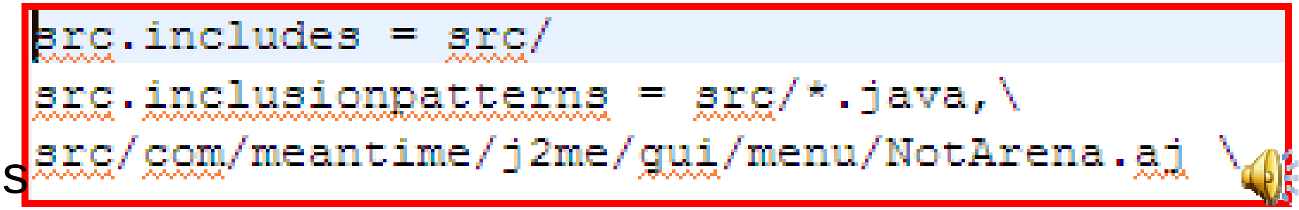
ajproperties <<ajlist

that are used to build products



```
ppsymbols = device_screen_128x128, device_keys_nokia, device_

wtk.cldc.version = 1.0
wtk.midp.version = 1.0

obfuscator.options = -dontusemixedcaseclassnames -overloadag

midlet.icon.dir = res/icons/16x16
midlet.res = res/texts/${build.lang} res/levels res/sounds/o
midlet.res.includes = lang*.txt *.ott *.bvg aS40.dat 1.dat

midlet.jad.entriesfile = build/properties/S40.jad.entries

wtk.emulator.device = Nokia_7210_MIDP_SDK_v1_0
aspects.enabled = true
aspects.compiler = abc
aspects.list = src/com/meantime/j2me/gui/menu/NotArena.aj \
```

S40.properties

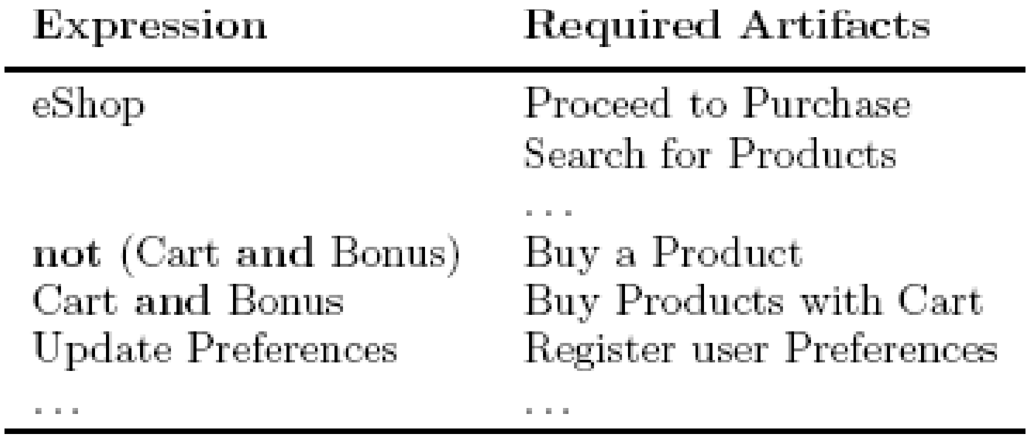


```
src.includes = src/
src.inclusionpatterns = src/*.java,\
src/com/meantime/j2me/gui/menu/NotArena.aj \
```

S40.ajproperties

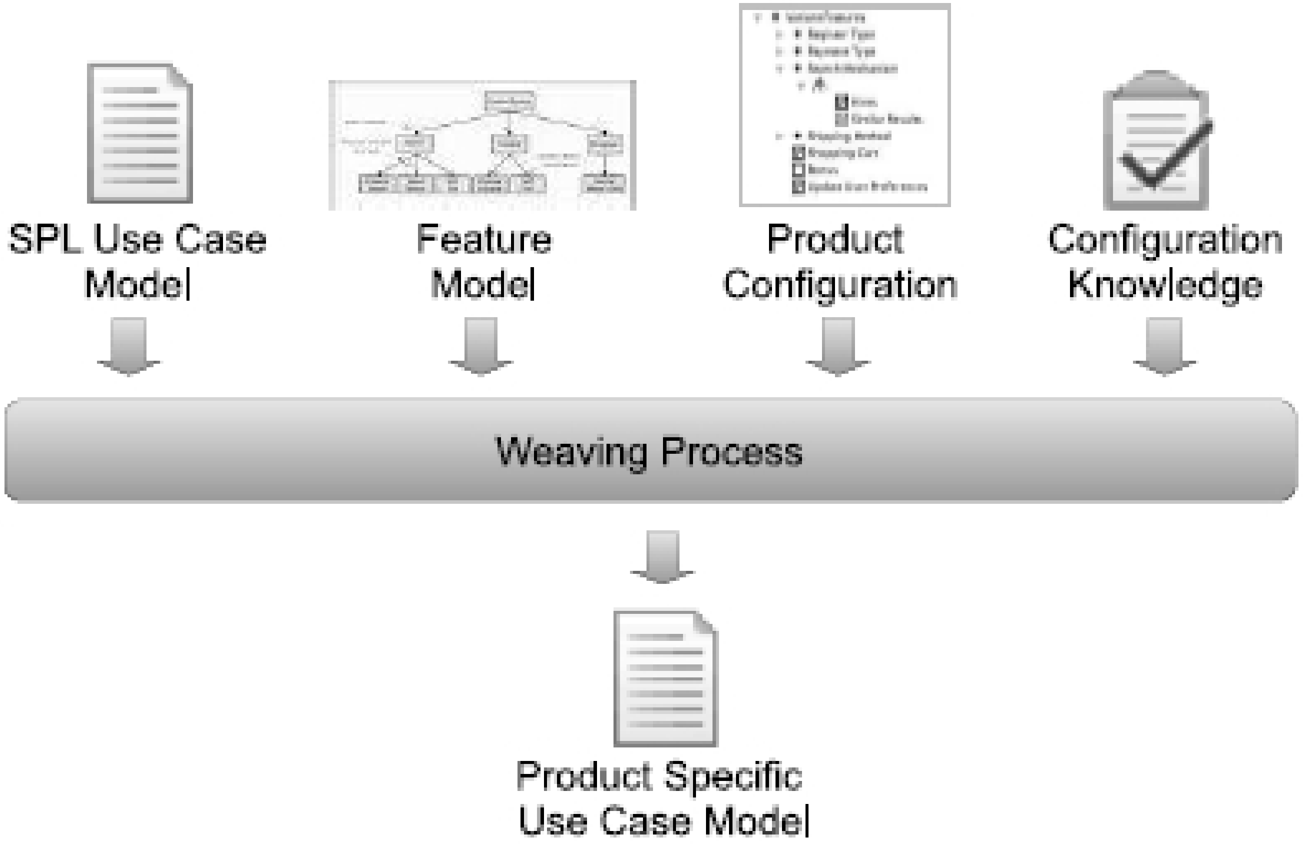
Configuration knowledge essentially is...

a mapping from feature expressions to core assets



Expression	**Required Artifacts**
eShop	Proceed to Purchase Search for Products ...
not (Cart **and** Bonus)	Buy a Product
Cart **and** Bonus	Buy Products with Cart
Update Preferences	Register user Preferences
...	...

Product generation, use cases view



SPL Use Case Model

Feature Model

Product Configuration

Configuration Knowledge

Weaving Process

Product Specific Use Case Model

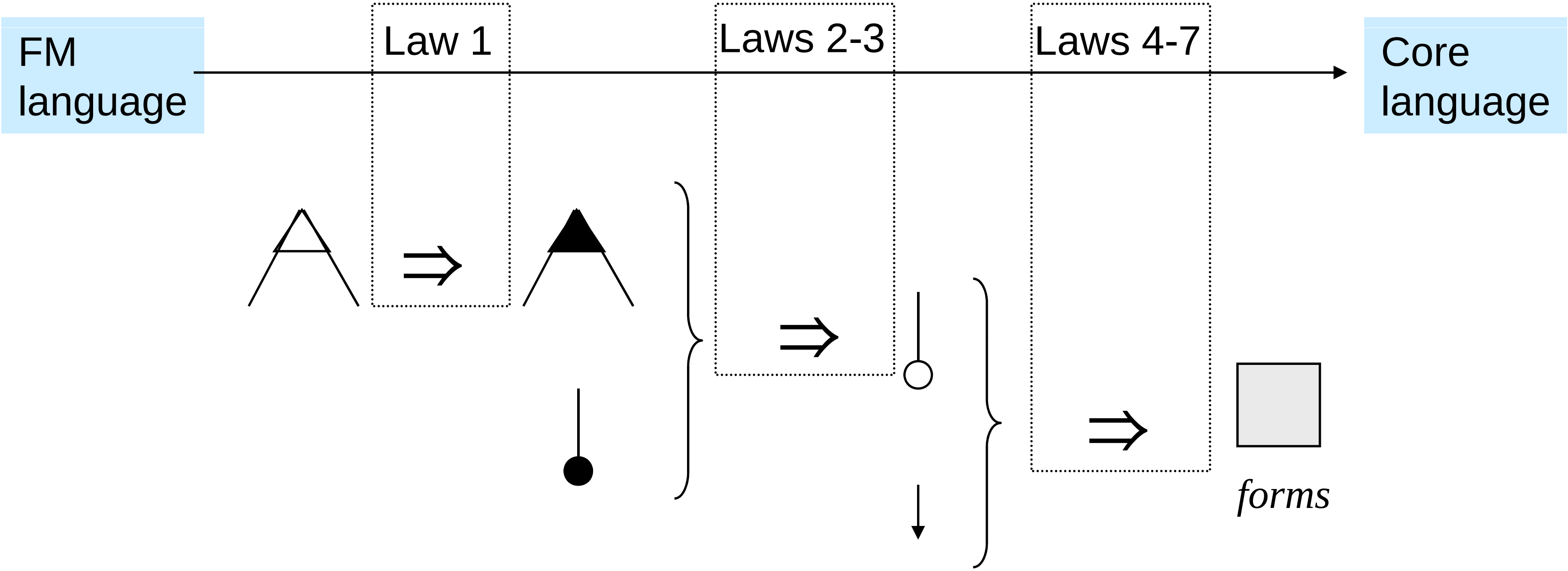
Models for software product lines

Paulo Borba

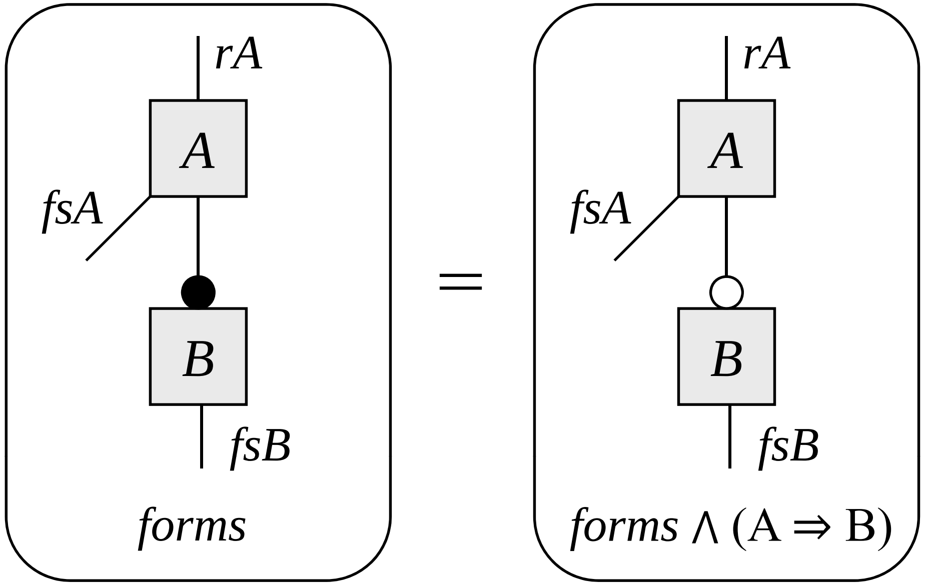
Informatics Center

Federal University of Pernambuco

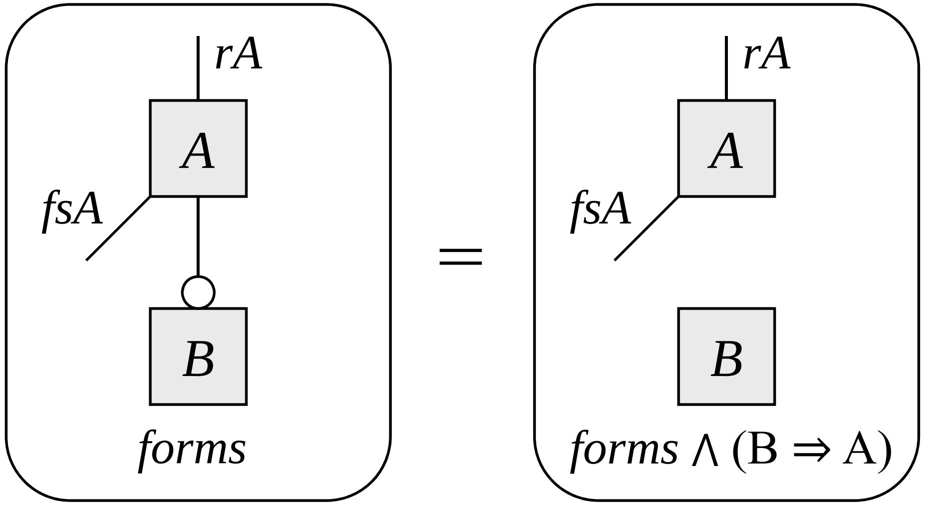
Reduction strategy to features and formulae language



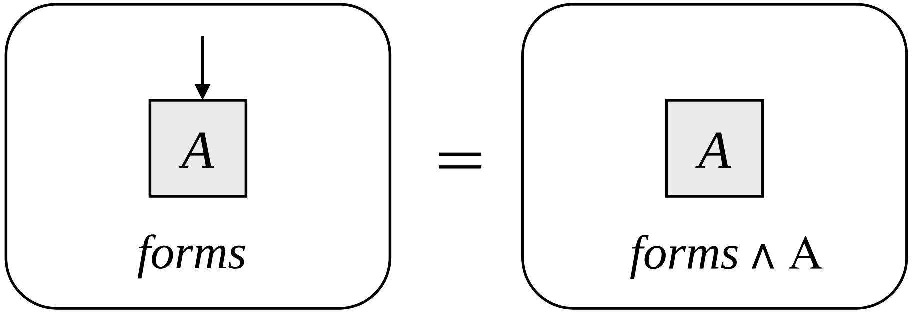
Replace Mandatory (Law 3)



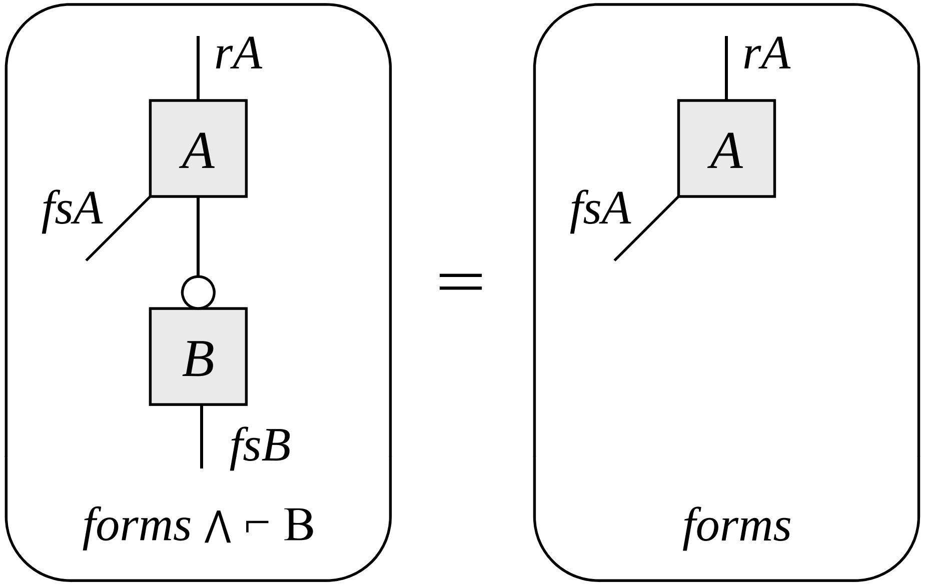
Replace Node (Law 4)



Remove Root (Law 5)



Remove Node (Law 6)



Add Formula (Law 7)

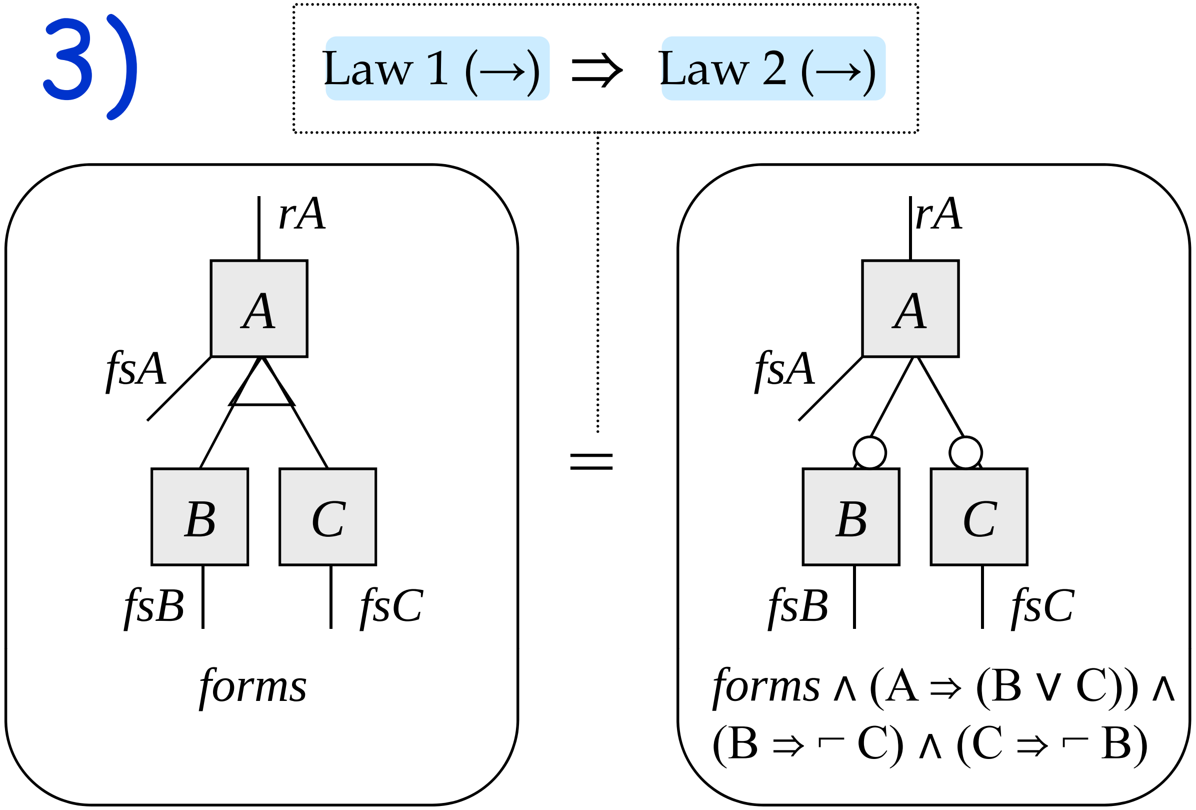
$$\boxed{\begin{array}{c} fs \\ forms \end{array}} = \boxed{\begin{array}{c} fs \\ forms \wedge f \end{array}}$$

(↔) *f* can be deduced from *forms* and *fs*.

Proved in PVS

Change Alternative to Optional (Ref 3)

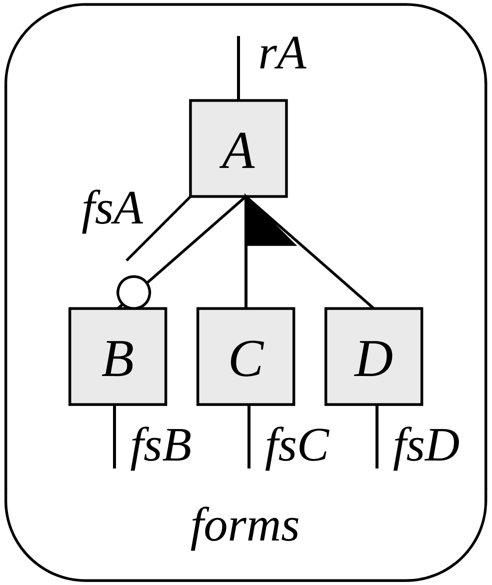
Law 1 (→) ⇒ Law 2 (→)



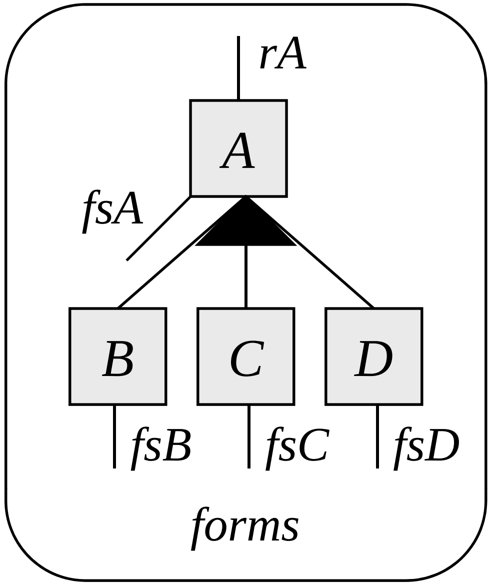
forms

forms ∧ (A ⇒ (B ∨ C)) ∧ (B ⇒ ¬ C) ∧ (C ⇒ ¬ B)

Collapse Optional and Or (Ref 2)

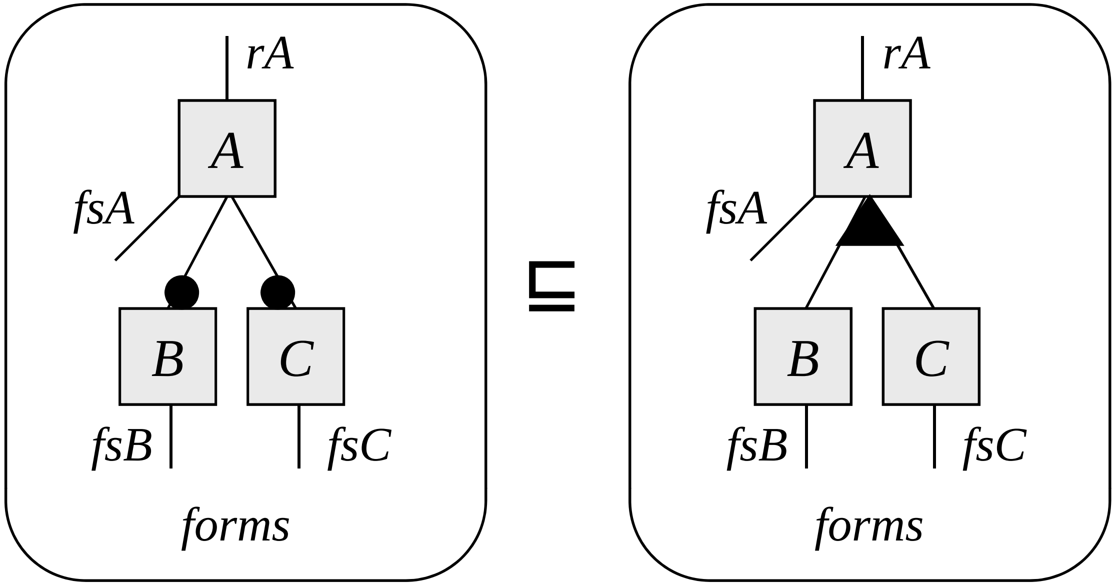


$\sqsubseteq$



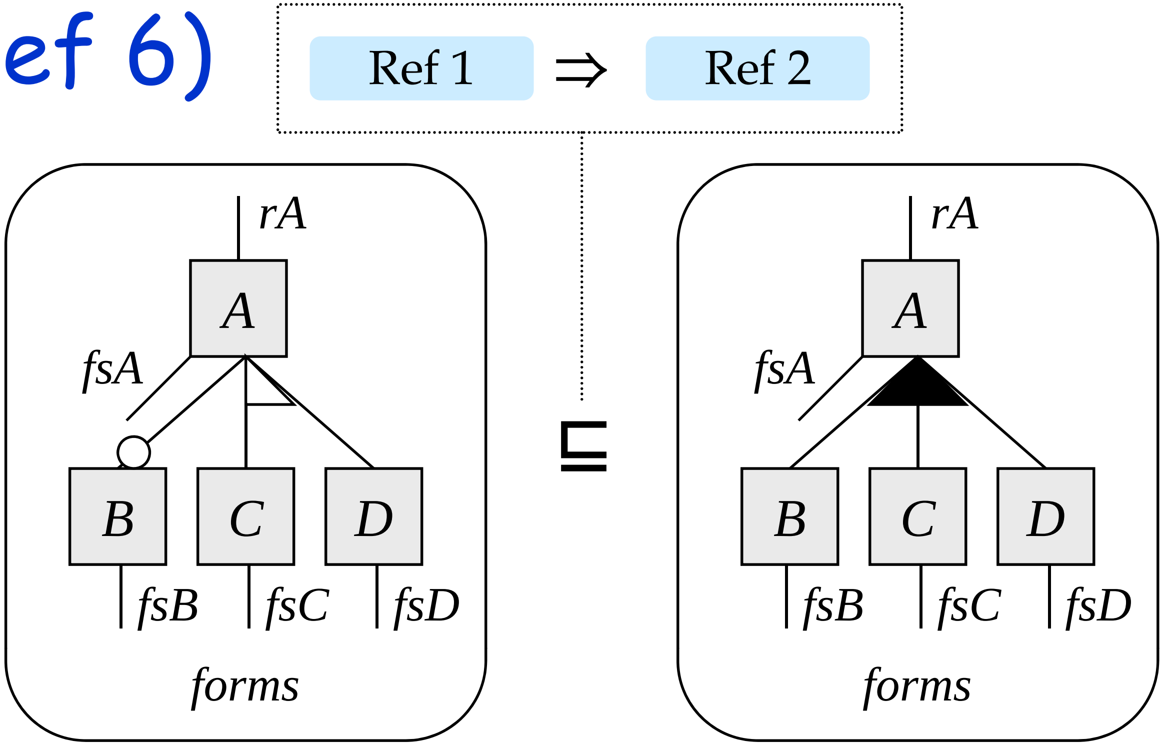
Proved in PVS

Add Or Between Mandatory (Ref 4)

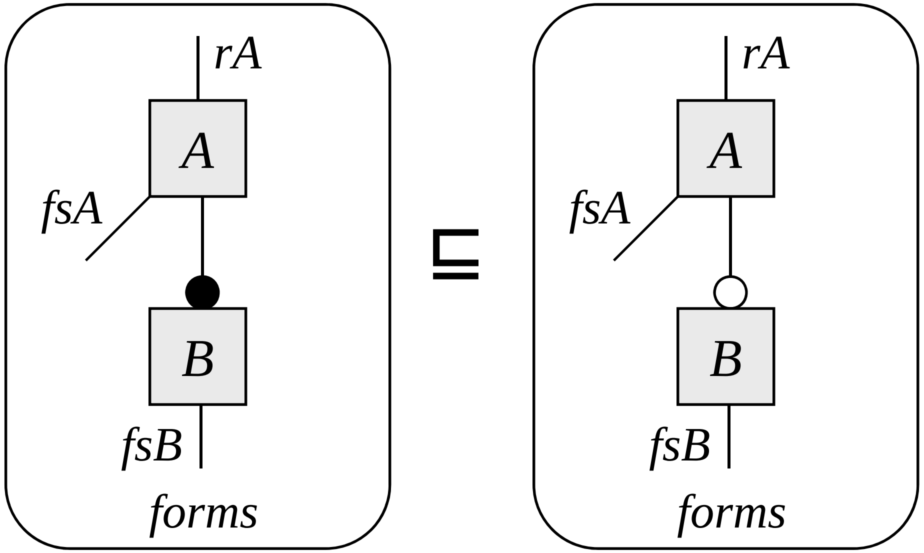


Proved in PVS

Collapse Optional and Alternative to Or (Ref 6)

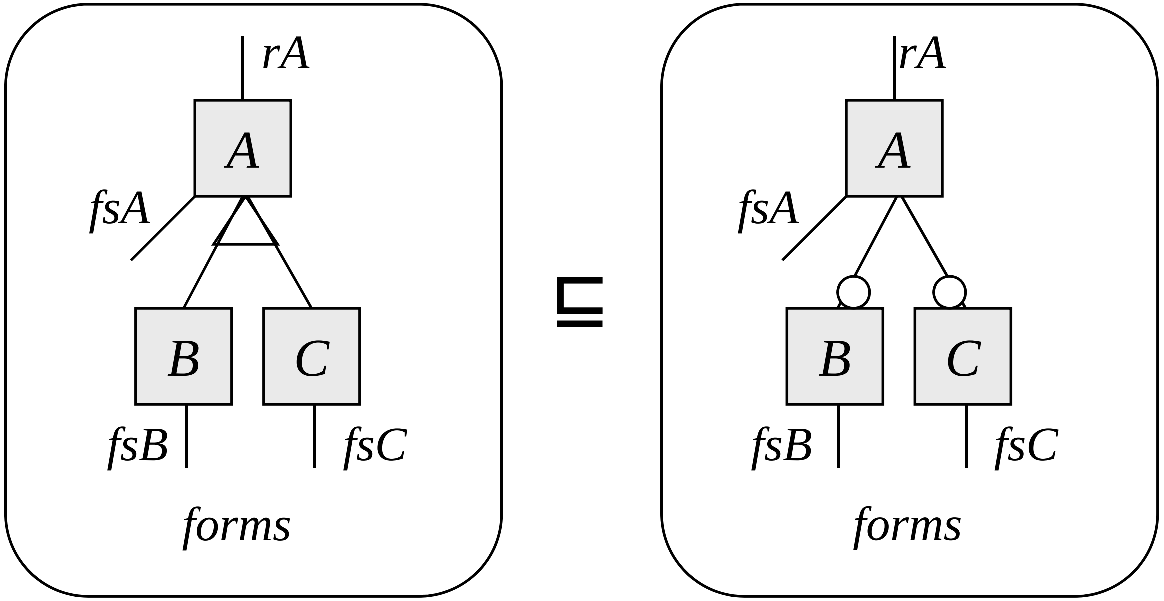


Remove Mandatory (Ref 7)



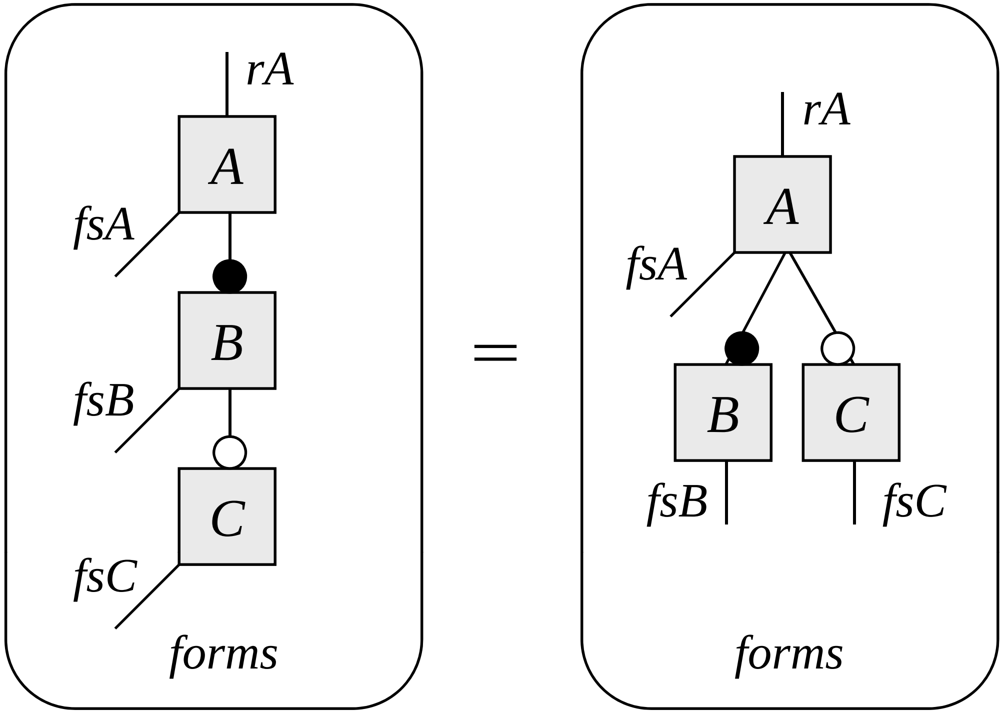
Proved in PVS

Remove Alternative (Ref 8)



Proved in PVS

Pull up Node (Ref 9-10)



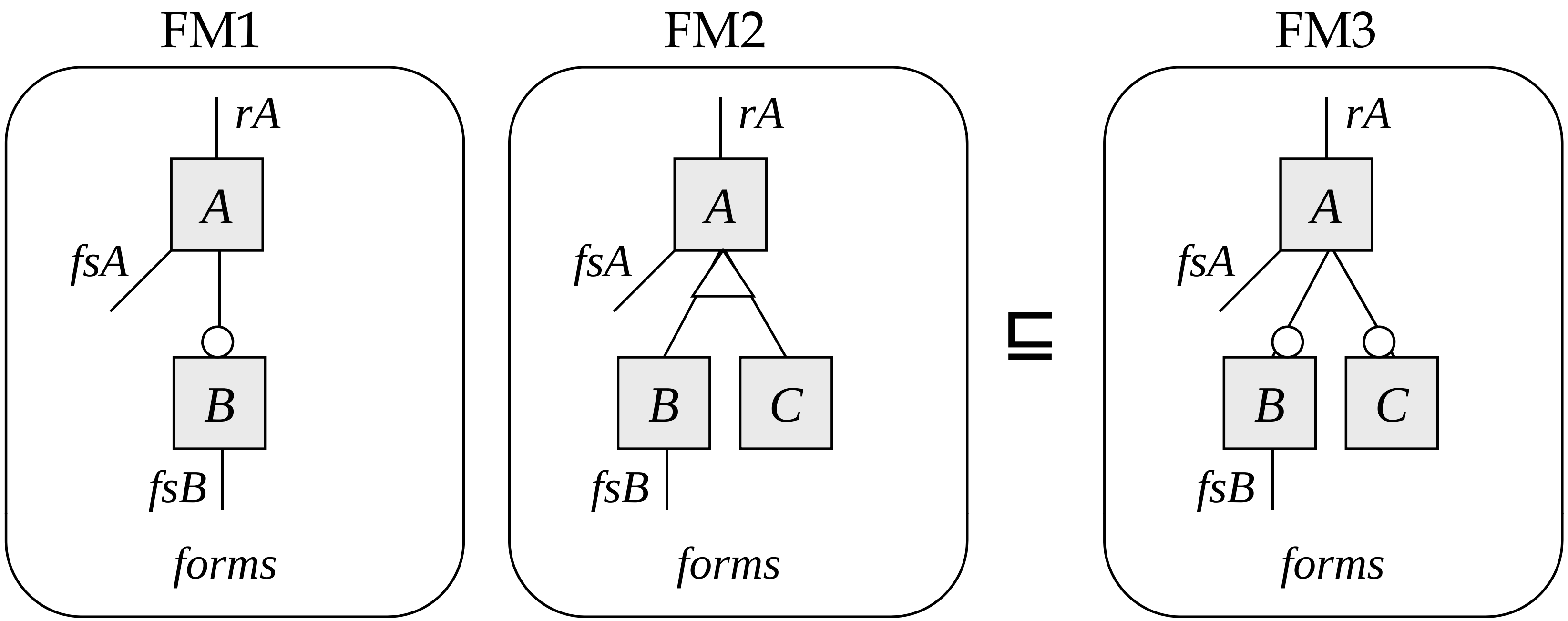
Proved in PVS

Remove Formula (Ref 11)

$$\boxed{\begin{array}{c} fs \\ forms \wedge f \end{array}} \sqsubseteq \boxed{\begin{array}{c} fs \\ forms \end{array}}$$

Proved in PVS

Extractive 1

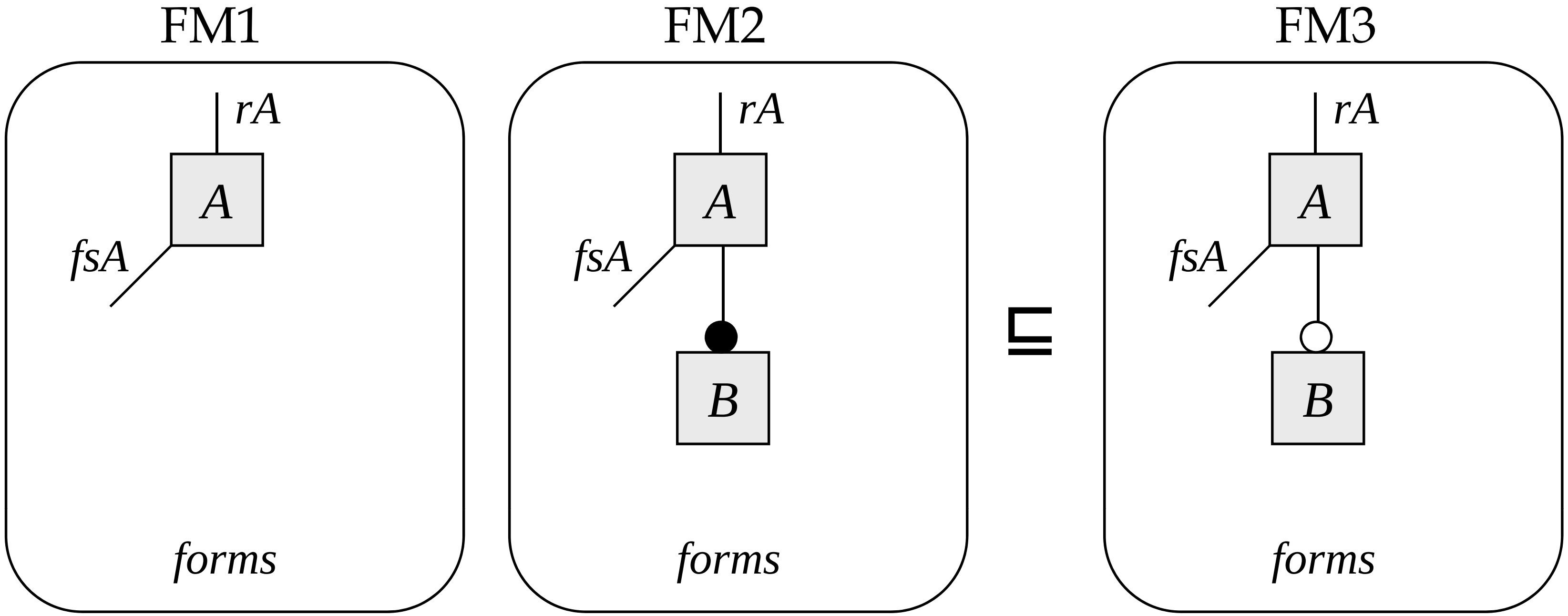


Proofs:

FM1 ⊑ FM3: Ref 12 (intro C)

FM2 ⊑ FM3: Ref 8

Extractive 2

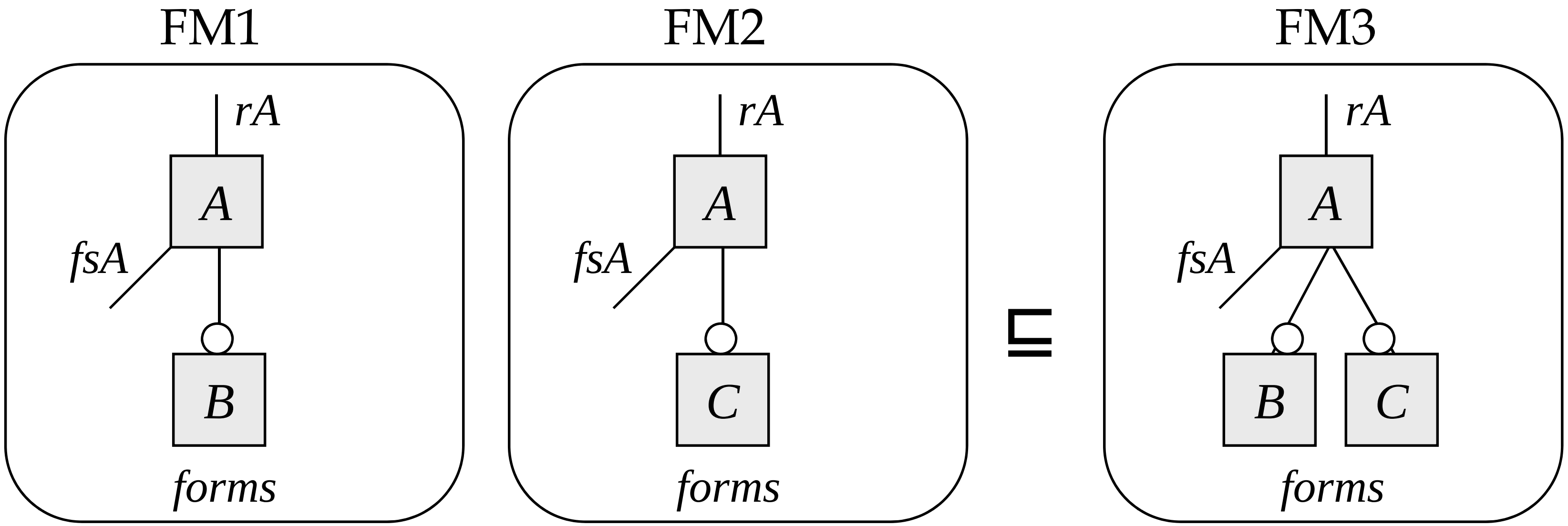


Proofs:

FM1 ⊑ FM3: Ref 12 (intro B)

FM2 ⊑ FM3: Ref 7 (B)

Extractive 4



Proofs:
FM1 ⊑ FM3: Ref 12 (intro C)
FM2 ⊑ FM3: Ref 12 (intro B)