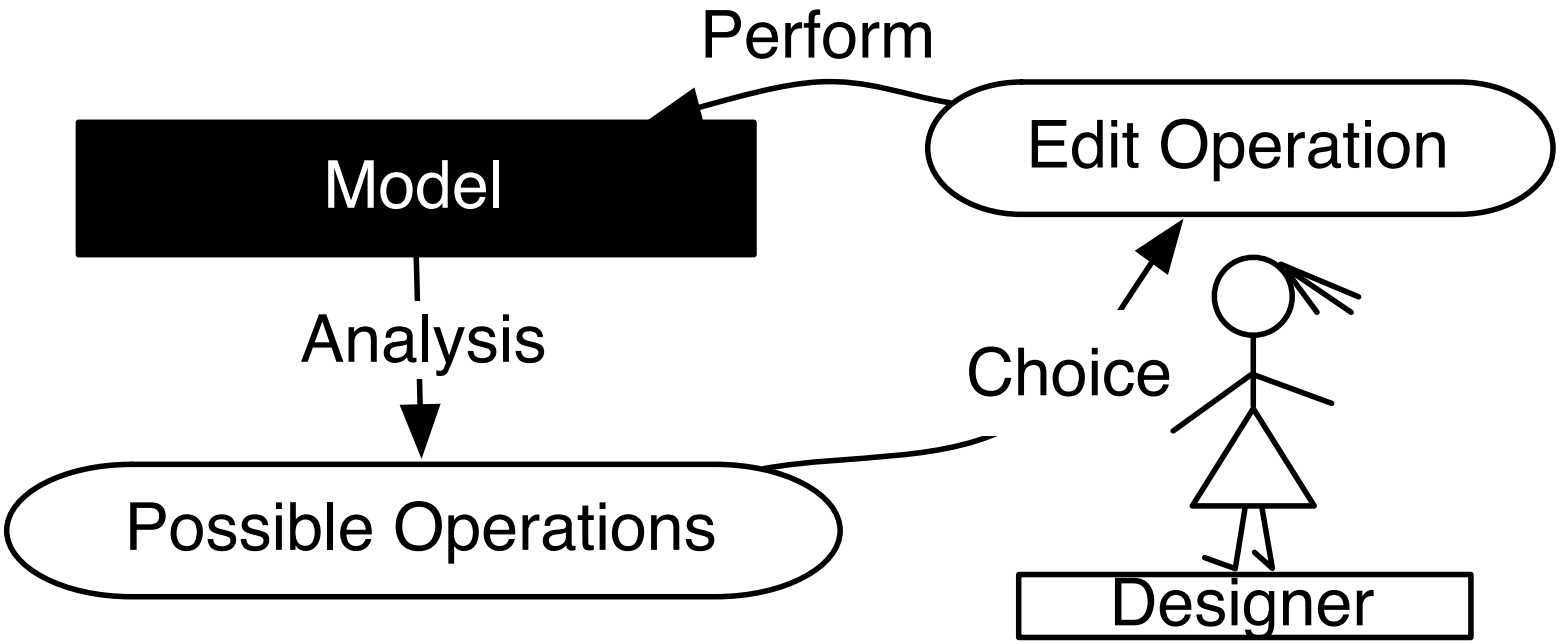
Interactive Model Derivation with External Constraints

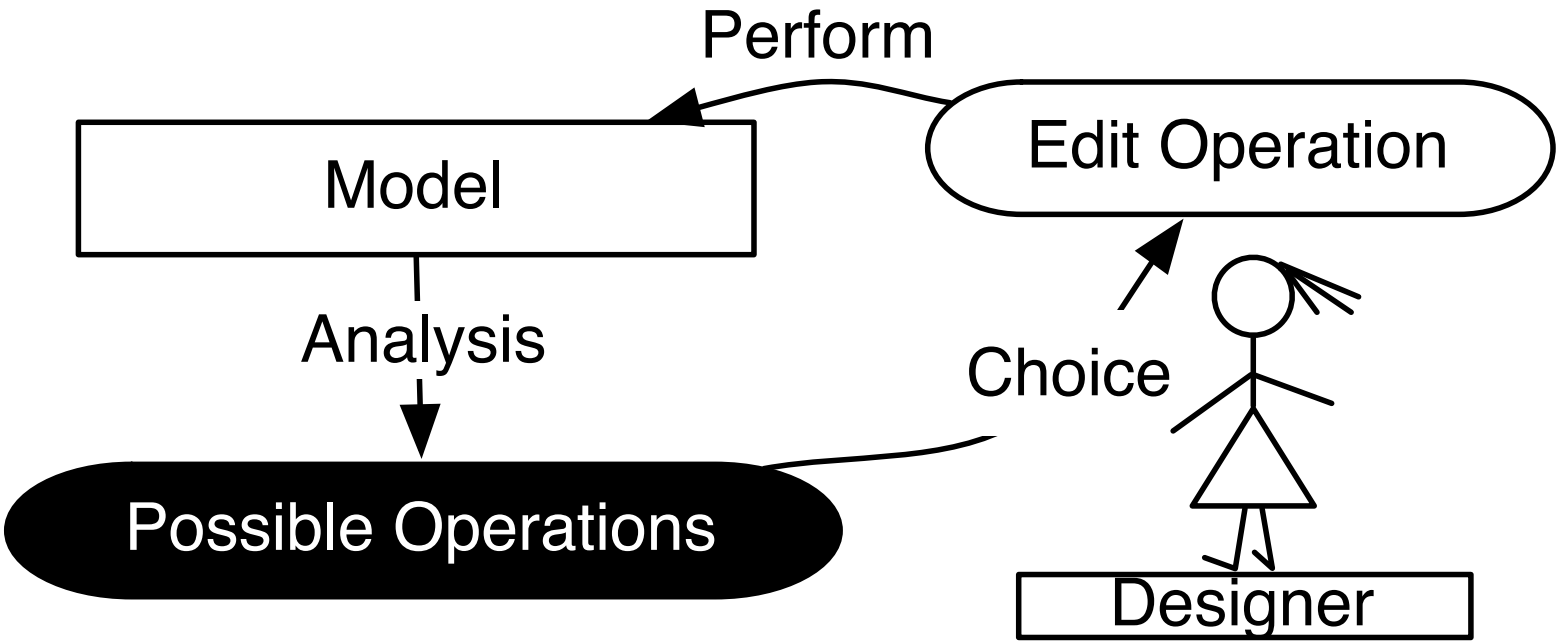
Mikoláš **J**anota
University College Dublin, Ireland

Victoria **K**uzina
Andrzej **W**ąsowski
IT University of Copenhagen, Denmark

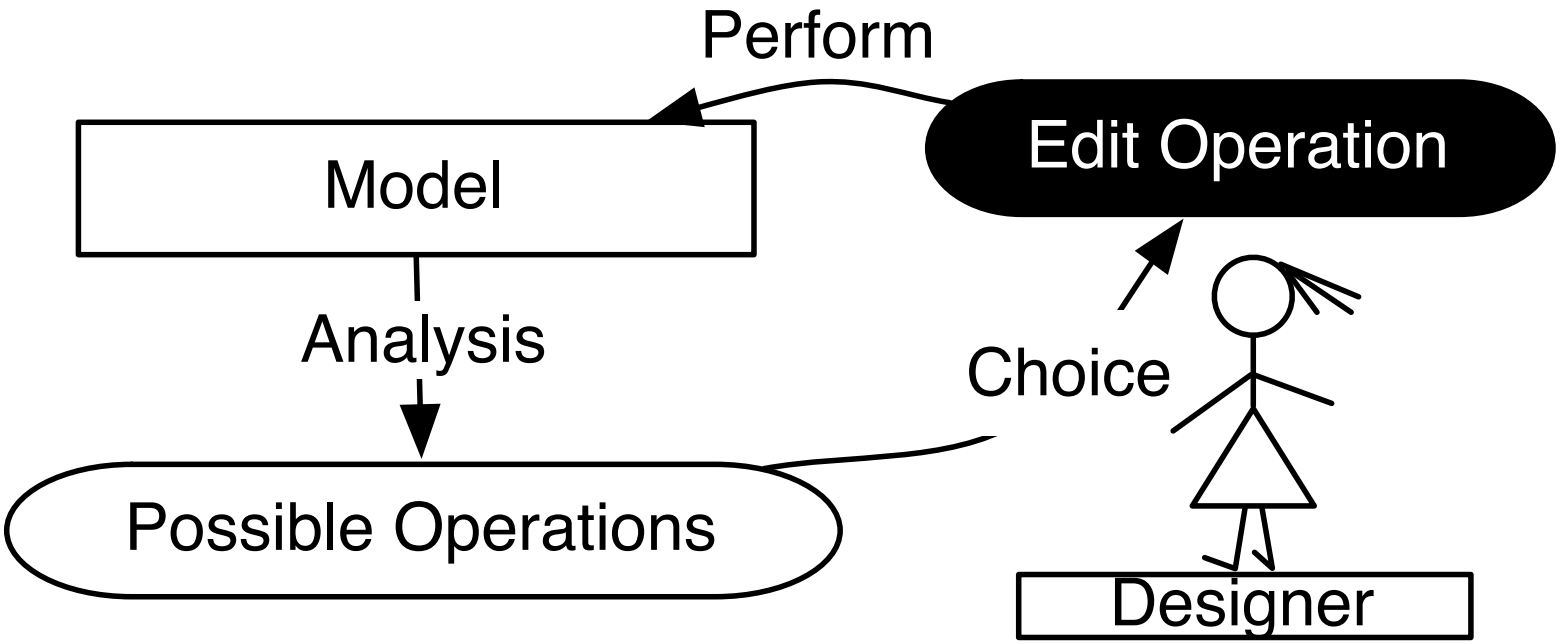
Inte


Perform
Edit Operation
Model
Analysis
Choice
Possible Operations
Designer

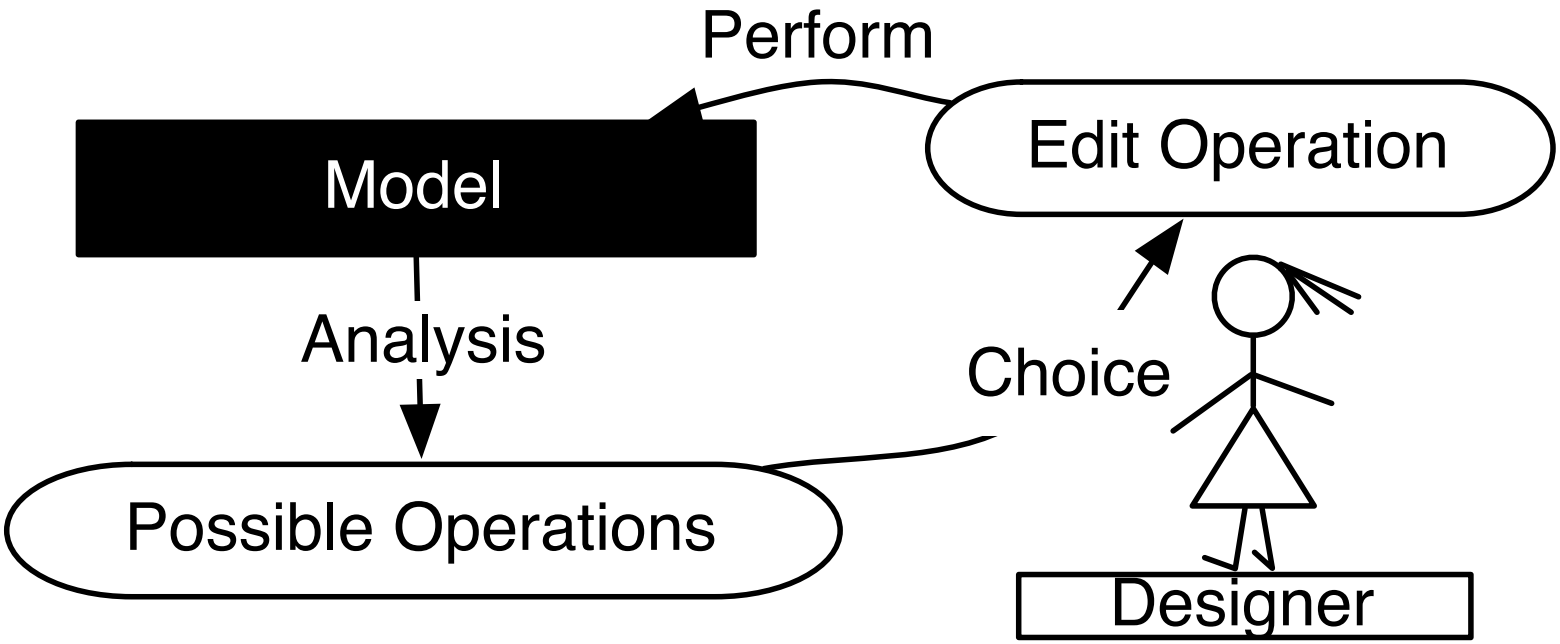
Inte


Perform
Model
Edit Operation
Analysis
Choice
Possible Operations
Designer

Inte


Perform
Model
Edit Operation
Analysis
Choice
Possible Operations
Designer

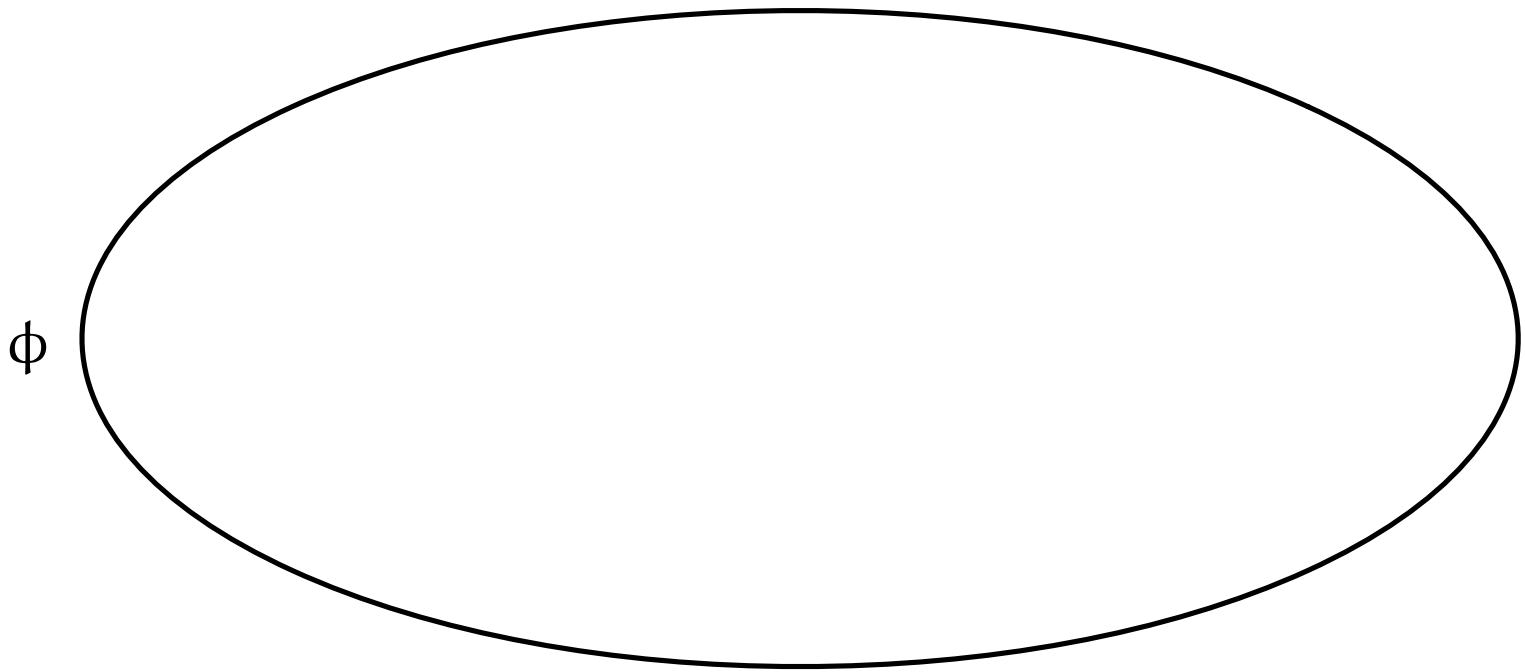
Inte


Perform
Edit Operation
Model
Analysis
Choice
Possible Operations
Designer

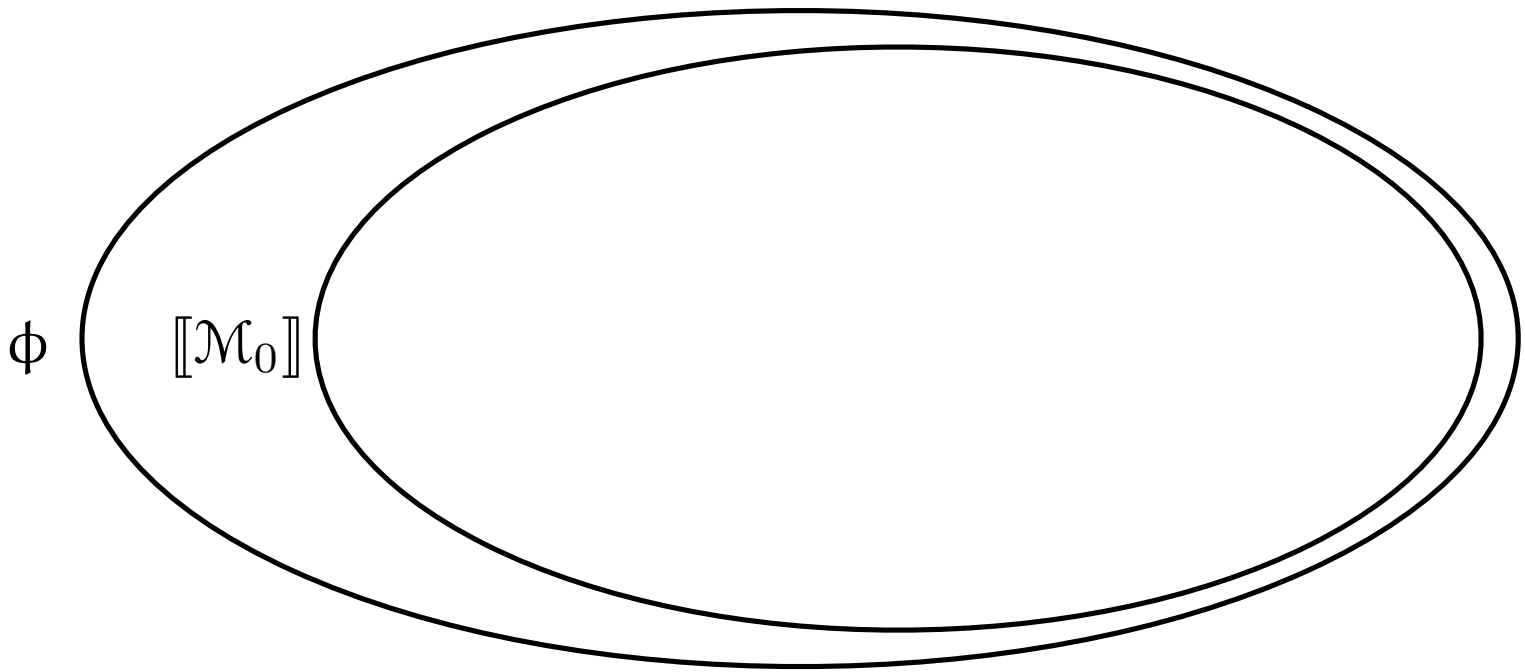
- Soundness-preserving derivation seen in instance derivation
- Completeness-preserving derivation will be illustrated by a prototype for feature diagrams
- Semantics-preserving derivation will be illustrated by a prototype for feature models

- Soundness-preserving derivation seen in instance derivation
- Completeness-preserving derivation will be illustrated by a prototype for feature diagrams
- Semantics-preserving derivation will be illustrated by a prototype for feature models

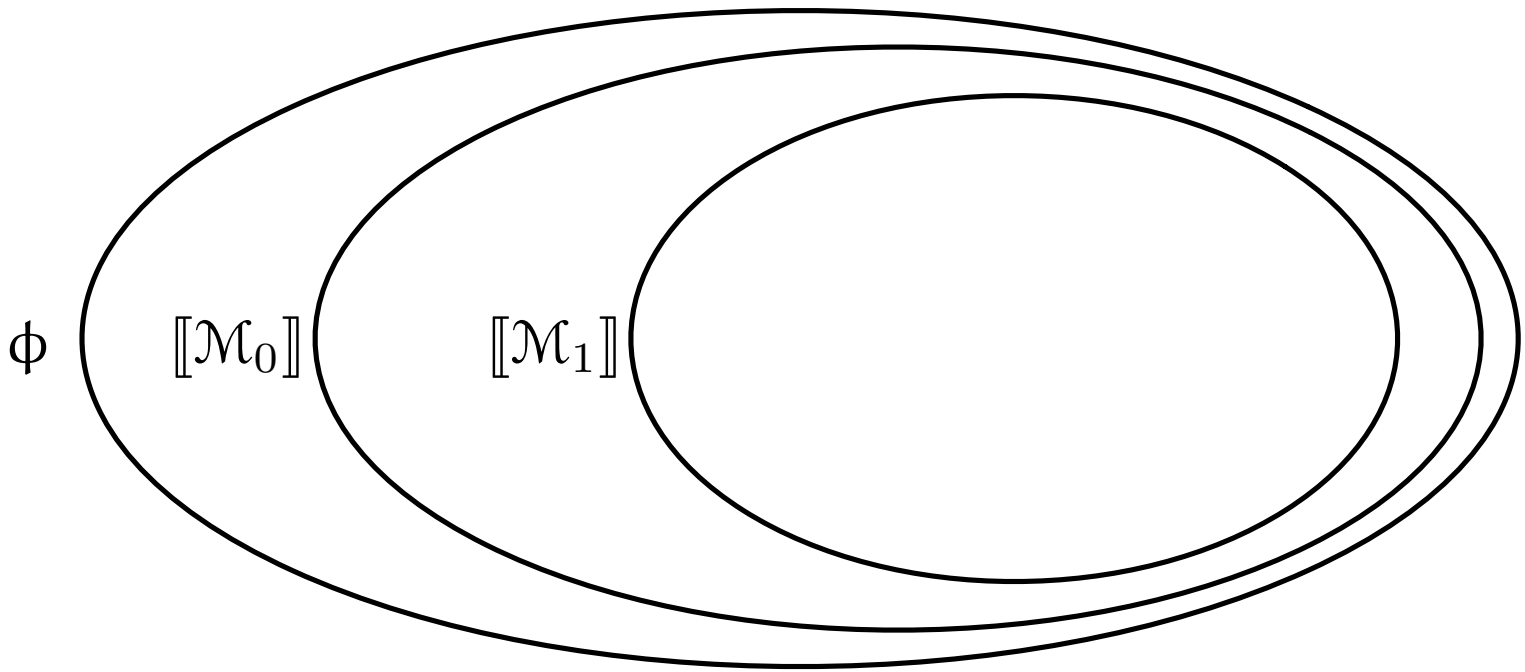
- Soundness-preserving derivation seen in instance derivation
- Completeness-preserving derivation will be illustrated by a prototype for feature diagrams
- Semantics-preserving derivation will be illustrated by a prototype for feature models



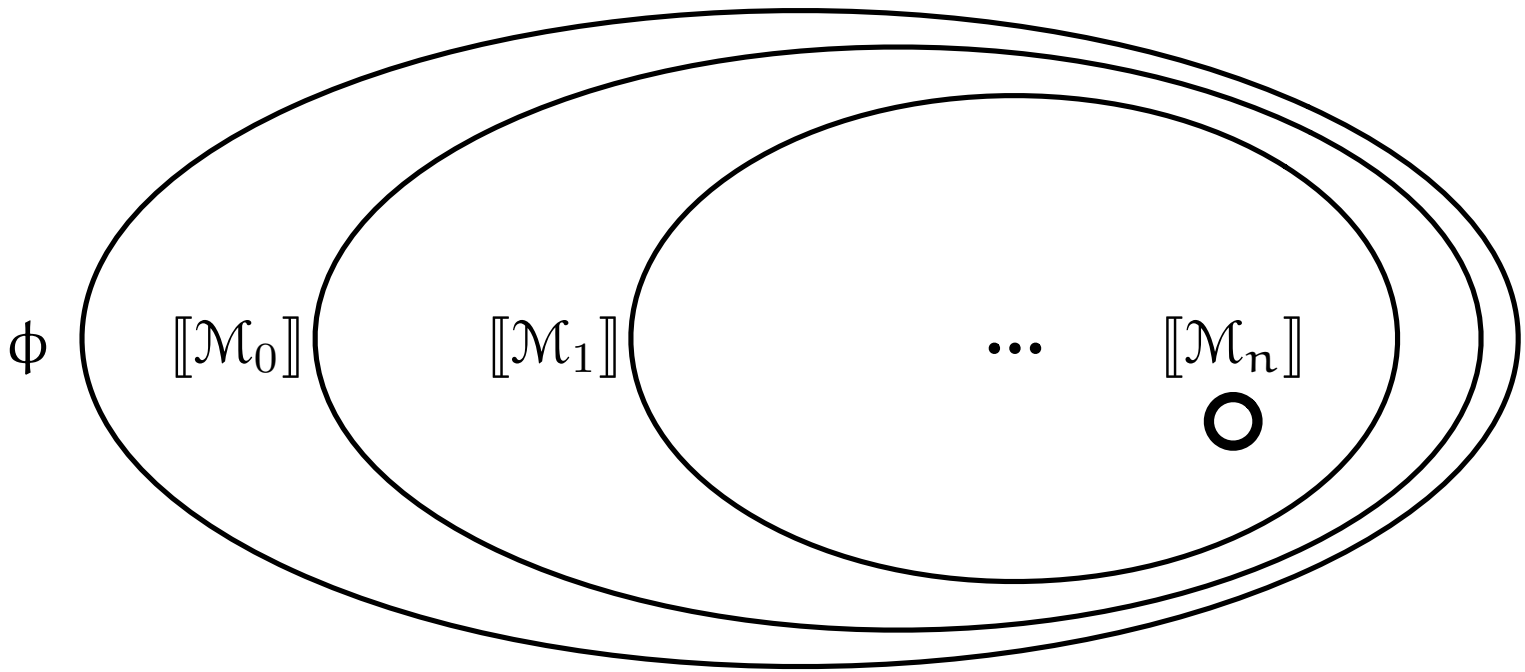
model $\rightarrow$ an instance



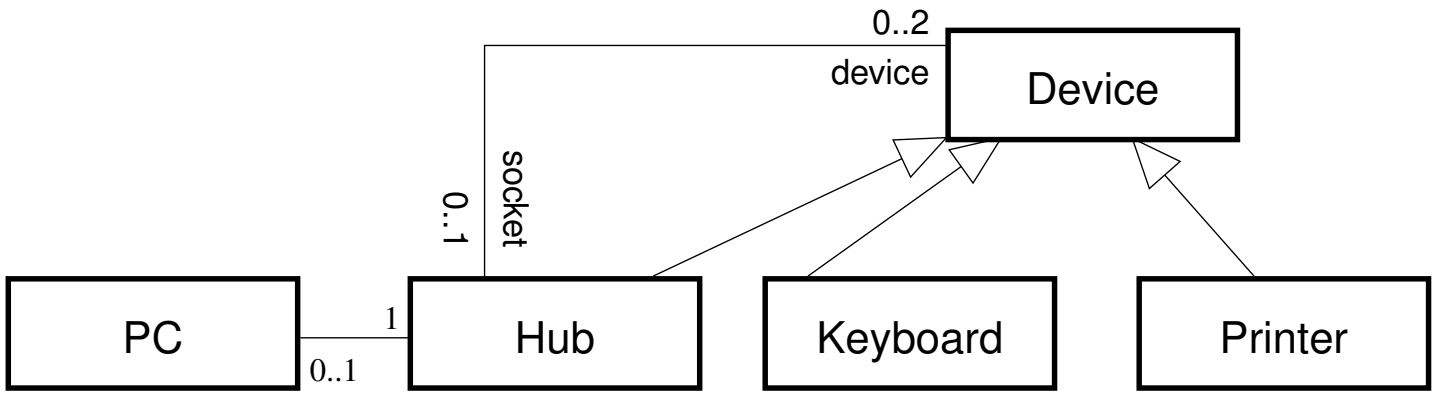
model $\rightarrow$ an instance



model $\rightarrow$ an instance



model $\rightarrow$ an instance

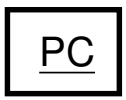


C1 Each USB must contain exactly one instance of PC.

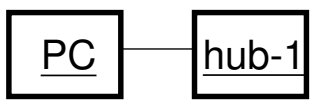
C2 Every device is connected to a port or to the PC instance.

C3 Every USB has a keyboard connected or a free port to connect one.

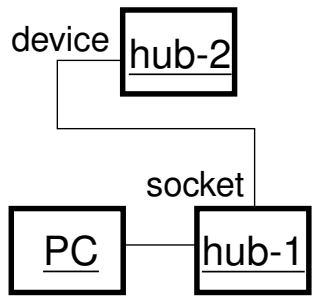
1.


PC

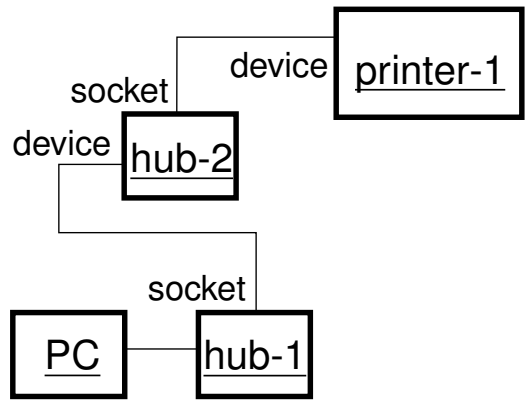
2.


PC
hub-1

3.

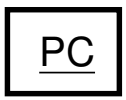

device
hub-2
socket
PC
hub-1

4.

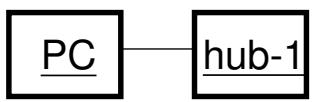

device
printer-1
socket
device
hub-2
socket
PC
hub-1

PC

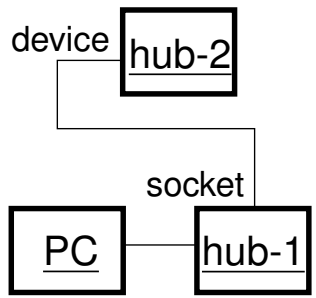
1.


PC

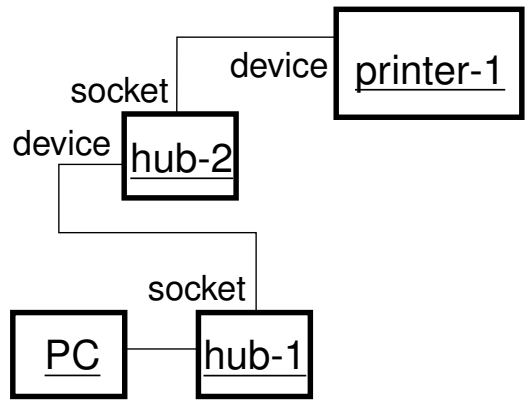
2.


PC
hub-1

3.

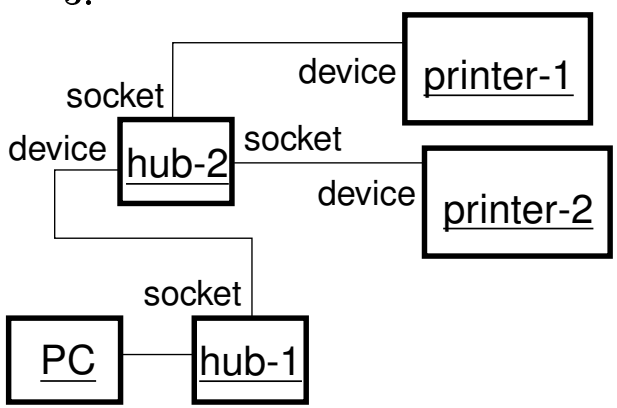

device
hub-2
socket
PC
hub-1

4.

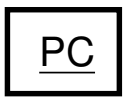

device
printer-1
socket
device
hub-2
socket
PC
hub-1

PC

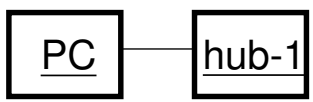
5.


device
printer-1
socket
device
hub-2
socket
device
printer-2
socket
PC
hub-1

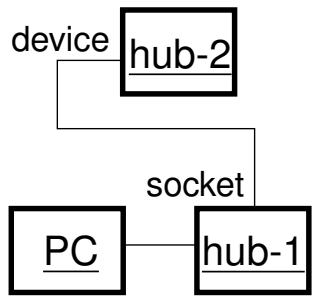
1.


PC

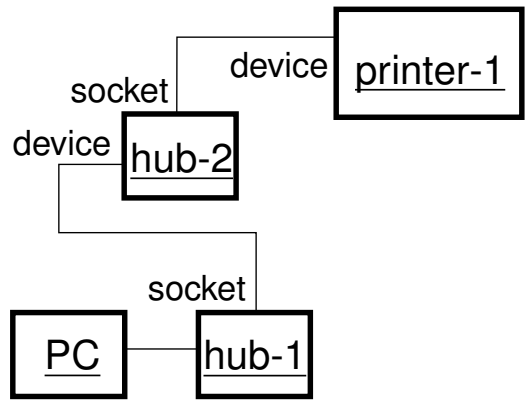
2.


PC
hub-1

3.

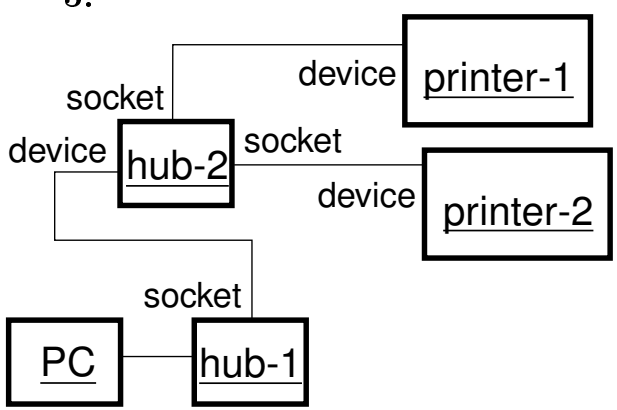

device
hub-2
socket
PC
hub-1

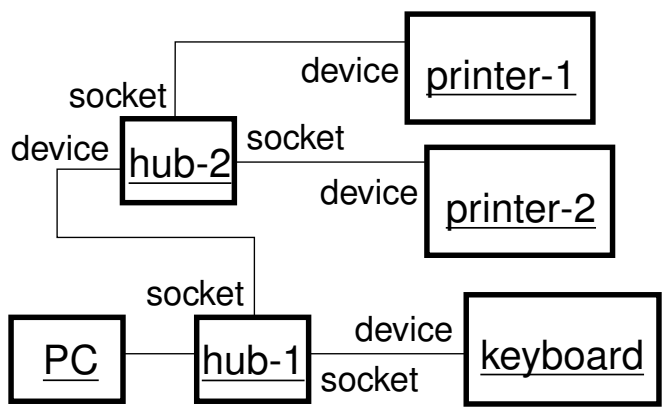
4.


device
printer-1
socket
device
hub-2
socket
PC
hub-1

PC

5.


device
printer-1
socket
device
hub-2
socket
device
printer-2
socket
PC
hub-1

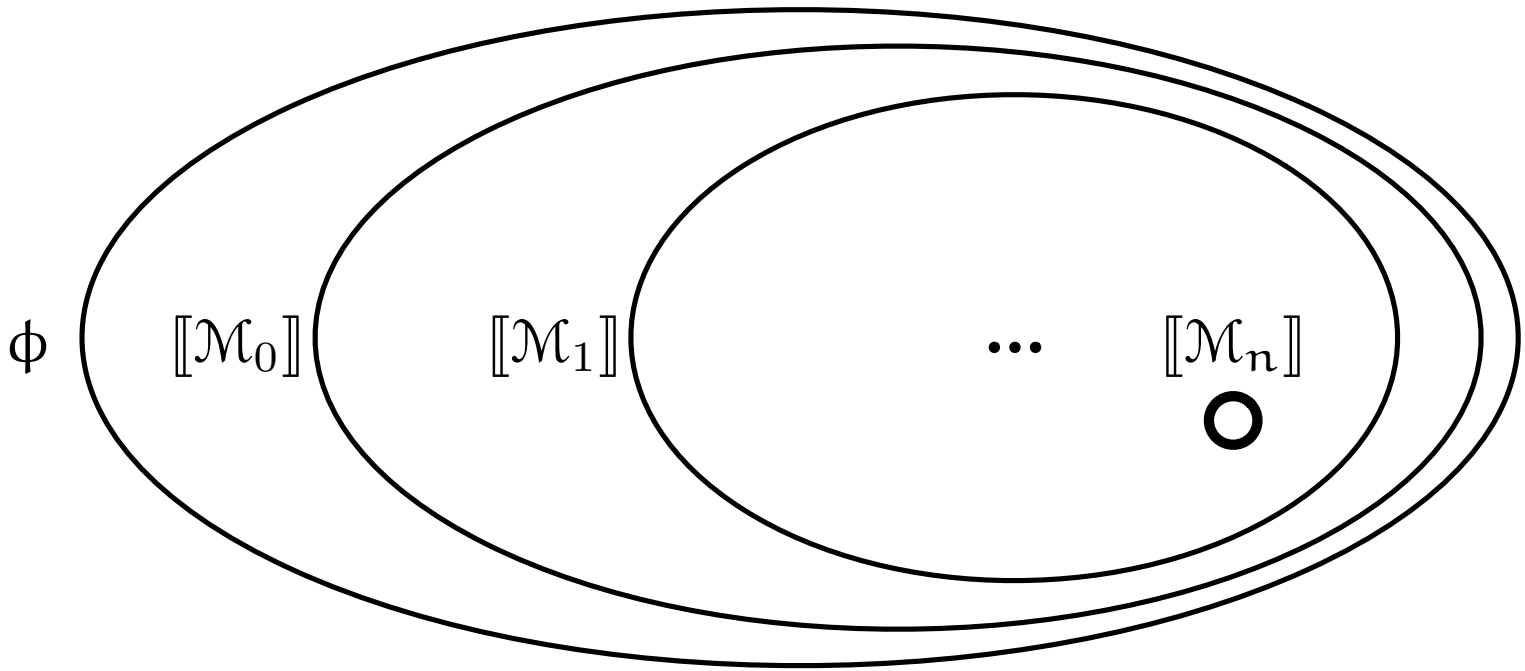

device
printer-1
socket
device
hub-2
socket
device
printer-2
socket
PC
hub-1
device
socket
keyboard

- **Validity of advice:** no sequence of operations leads to an invalid model
  - For the USB language: all derivable USBs satisfy the language constraints (the diagram and C1–C3)
- **Exhaustiveness of advice:** all conforming instances are derivable.
  - For the USB language: all legal USBs can be derived (*van der Meer* 2006)

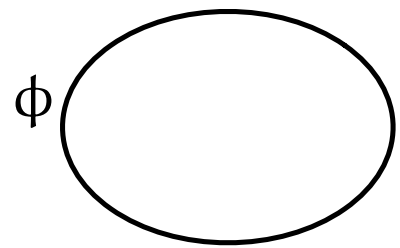
- **Validity of advice:** no sequence of operations leads to an invalid model
  - For the USB language: all derivable USBs satisfy the language constraints (the diagram and C1–C3)
- **Exhaustiveness of advice:** all conforming instances are derivable.
  - For the USB language: all legal USBs can be derived (*van der Meer* 2006)

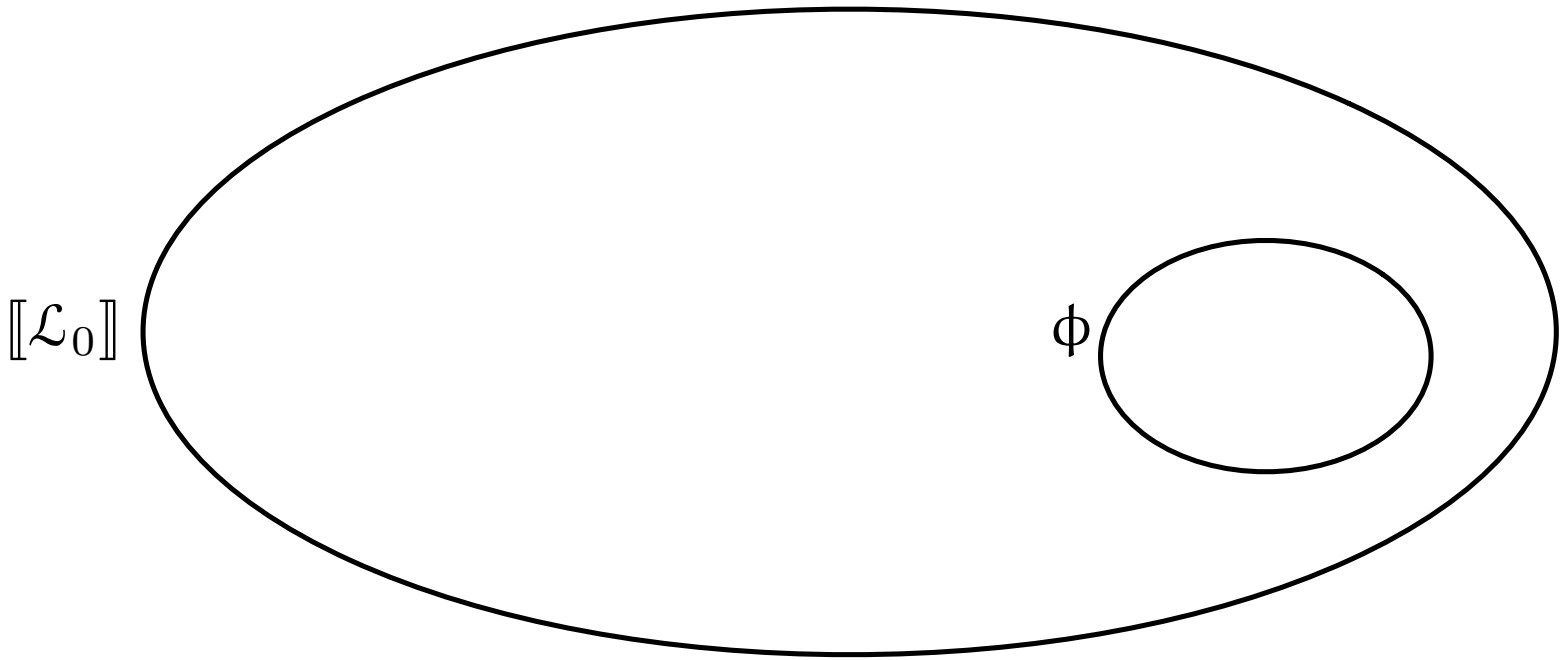
- **Validity of advice:** no sequence of operations leads to an invalid model
  - For the USB language: all derivable USBs satisfy the language constraints (the diagram and C1–C3)
- **Exhaustiveness of advice:** all conforming instances are derivable.
  - For the USB language: all legal USBs can be derived (*van der Meer* 2006)

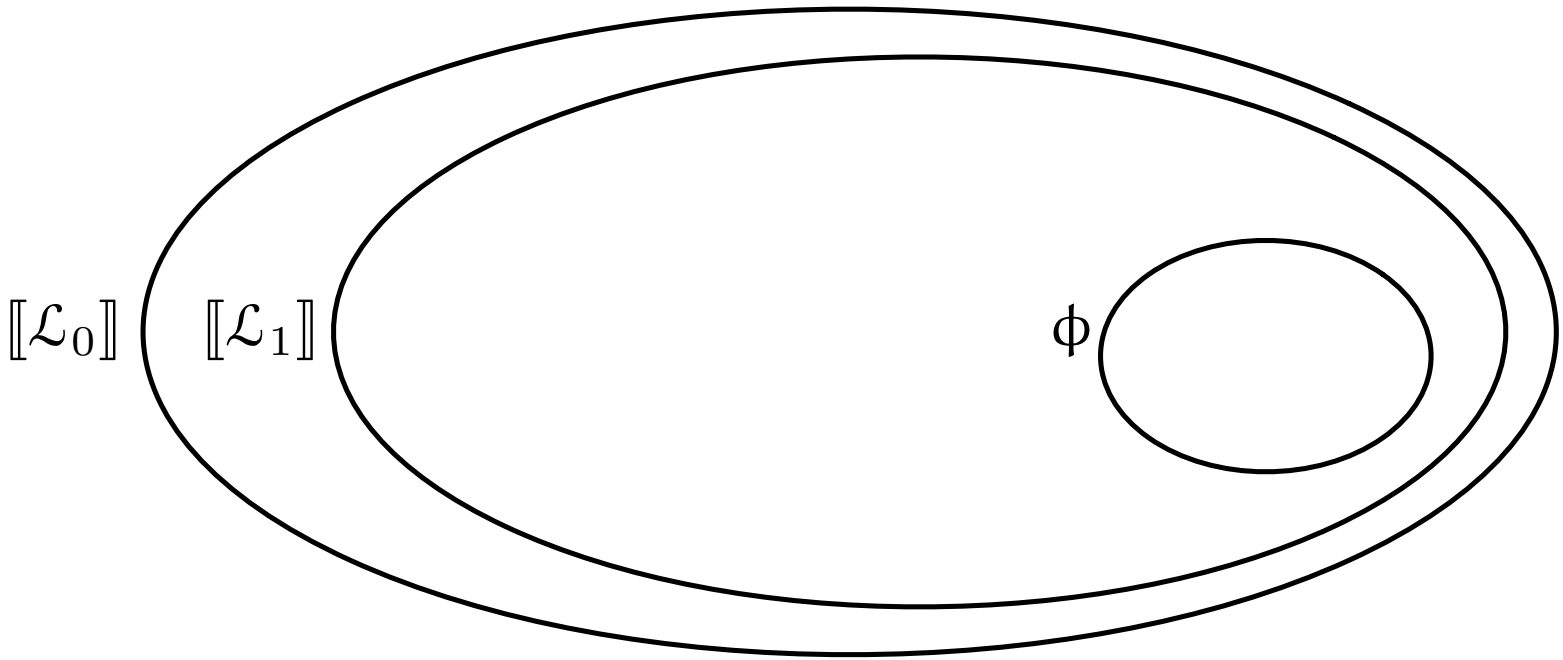
- **Validity of advice:** no sequence of operations leads to an invalid model
  - For the USB language: all derivable USBs satisfy the language constraints (the diagram and C1–C3)
- **Exhaustiveness of advice:** all conforming instances are derivable.
  - For the USB language: all legal USBs can be derived (*van der Meer* 2006)

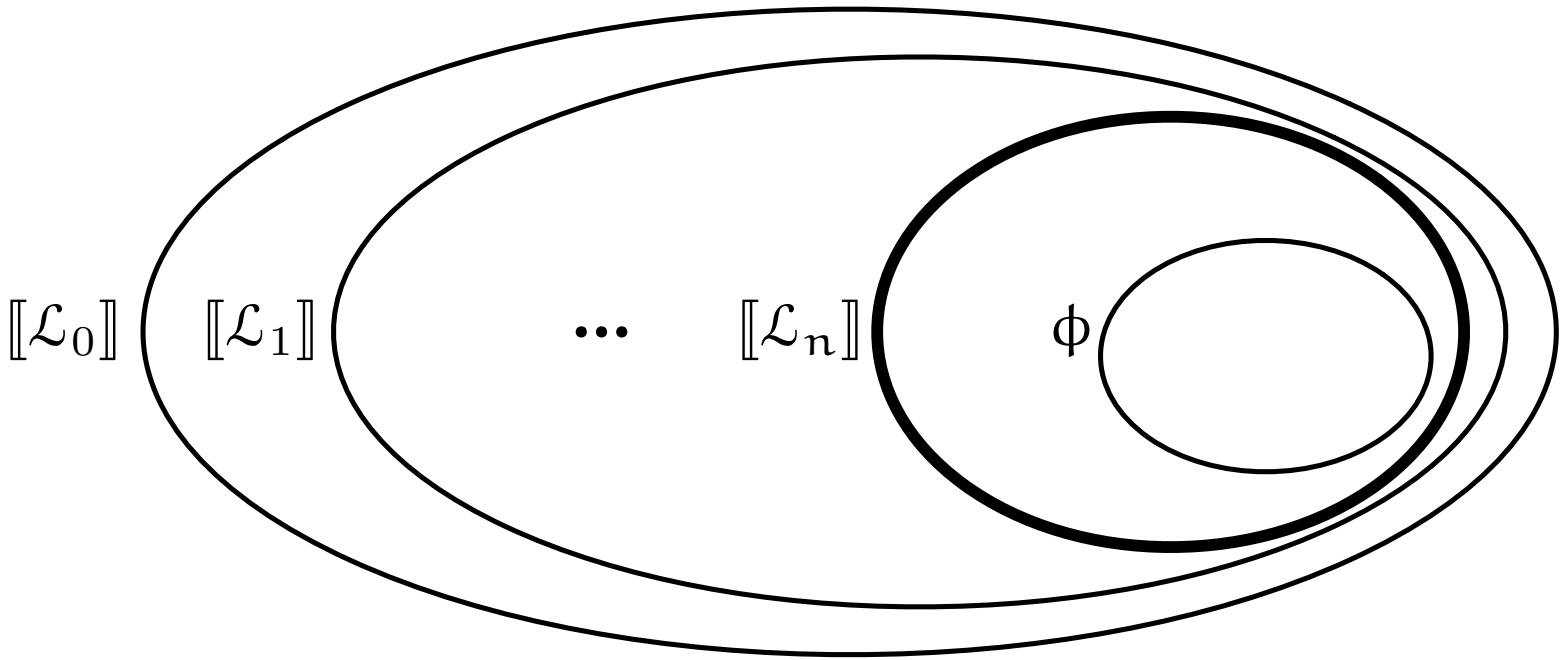


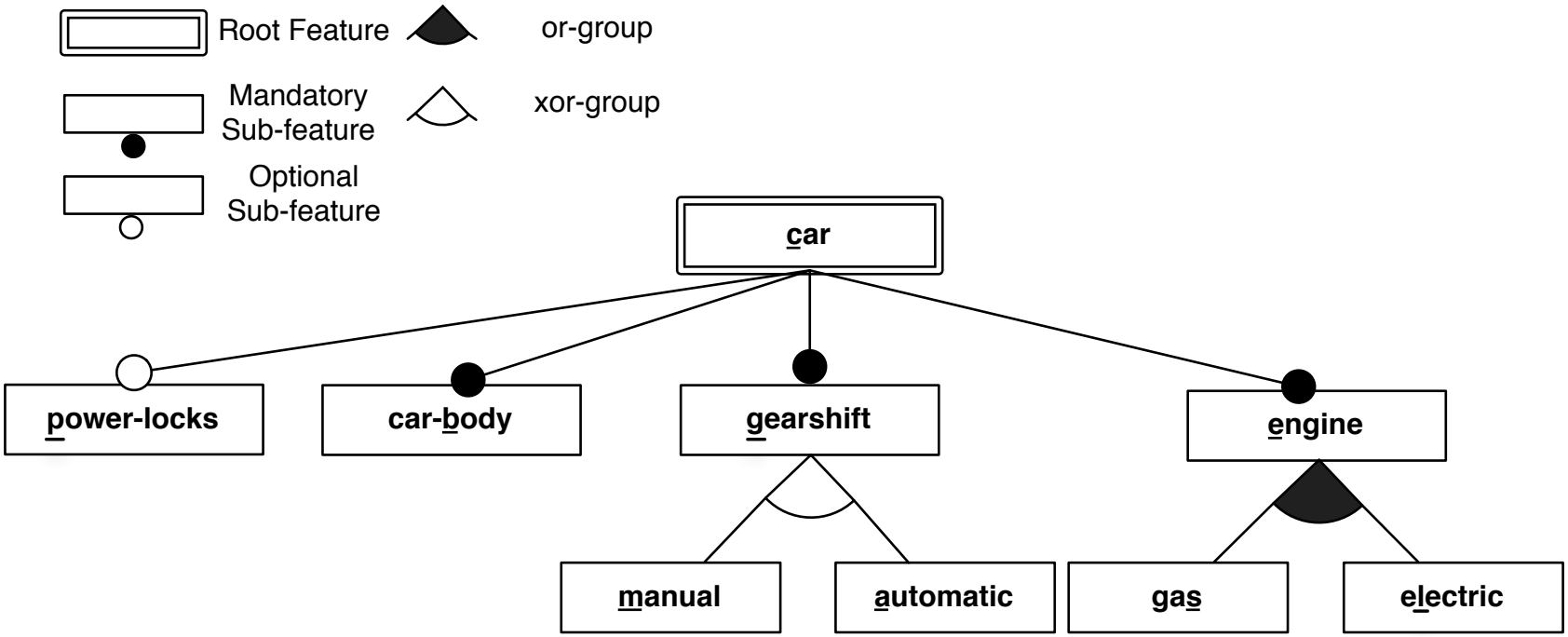
model $\rightarrow$ an instance


ϕ


$[\![\mathcal{L}_0]\!]$
ϕ

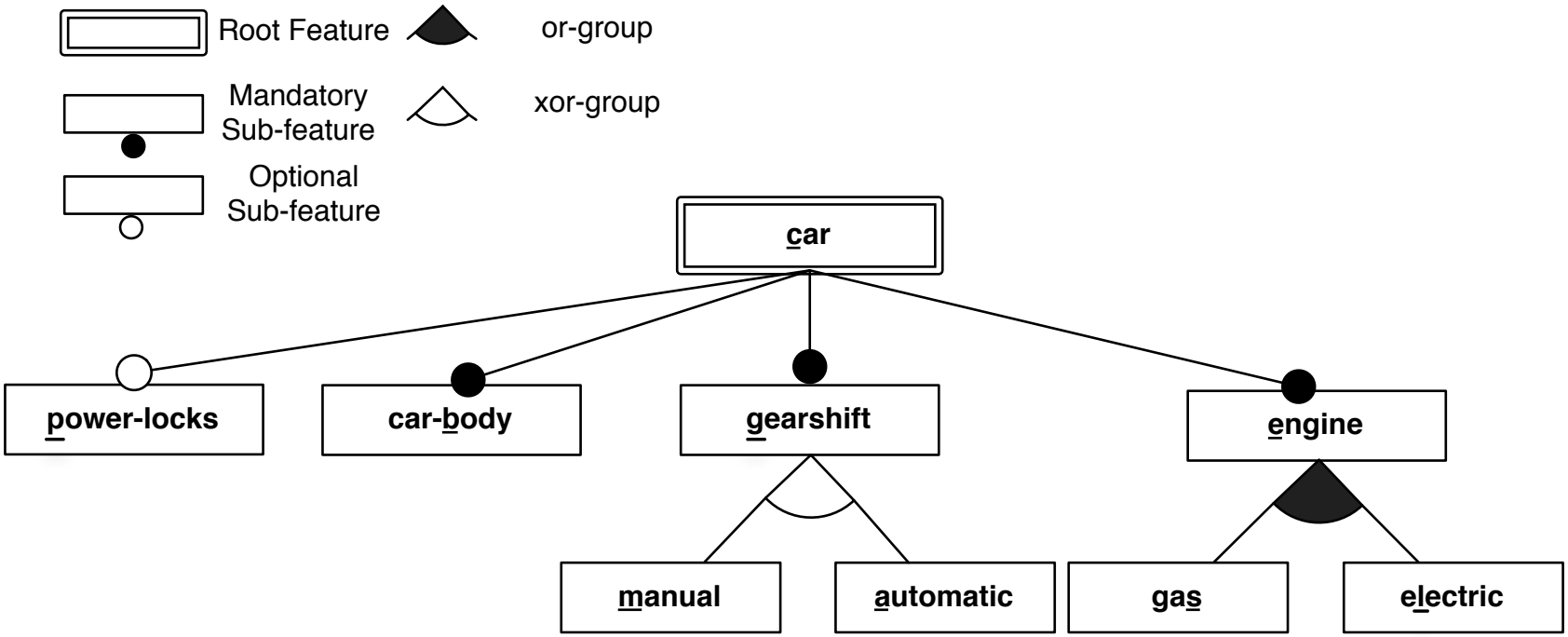

$[\![\mathcal{L}_0]\!]$
$[\![\mathcal{L}_1]\!]$
ϕ


$[\![\mathcal{L}_0]\!]$
$[\![\mathcal{L}_1]\!]$
$\cdots$
$[\![\mathcal{L}_n]\!]$
ϕ



additional constraint:

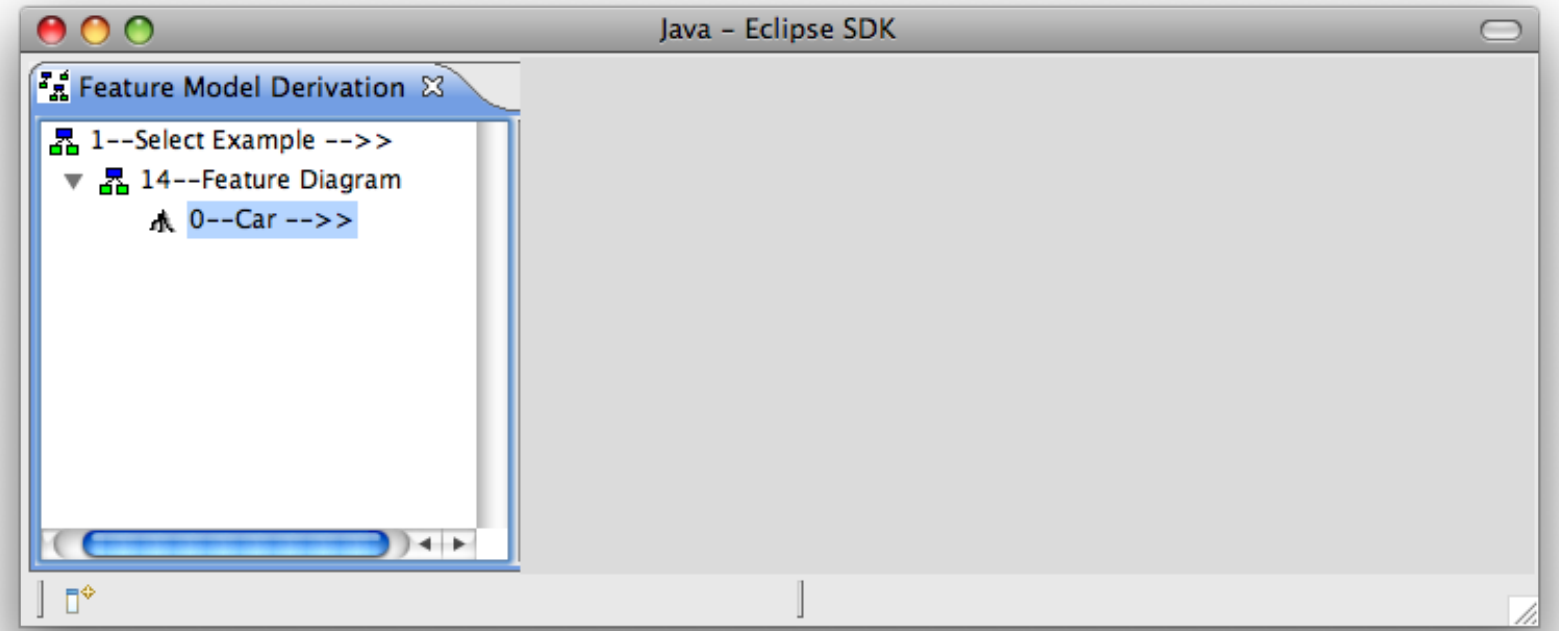
electric $\rightarrow$ automatic



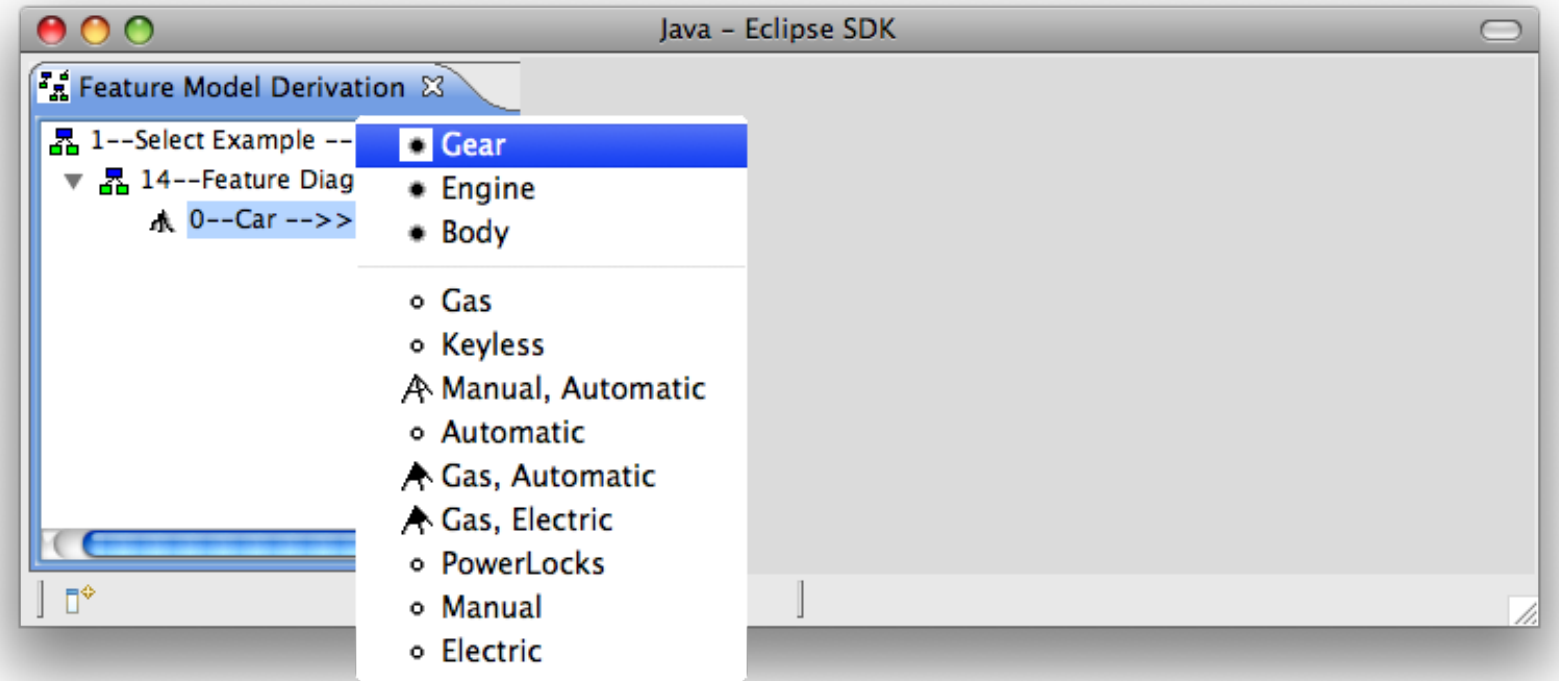
additional constraint:

$$\text{electric} \rightarrow \text{automatic}$$

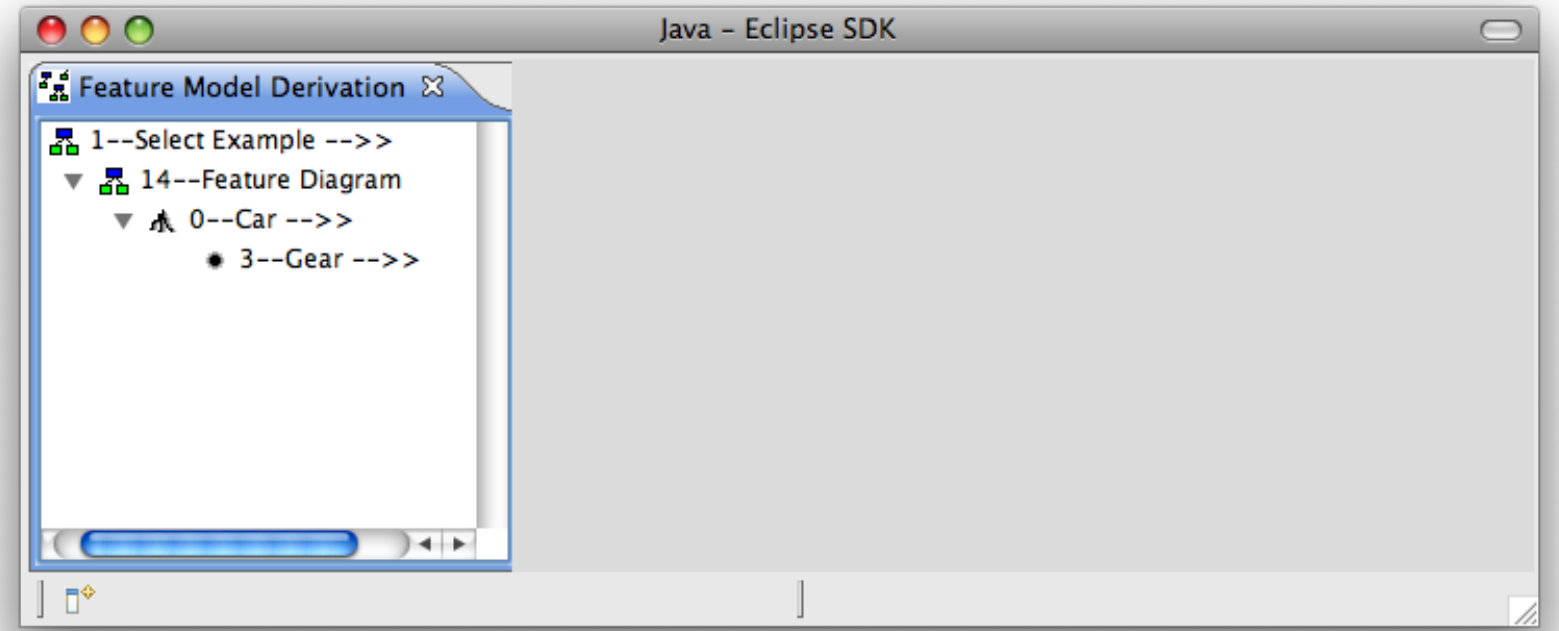
Feature Model approximating a constraint



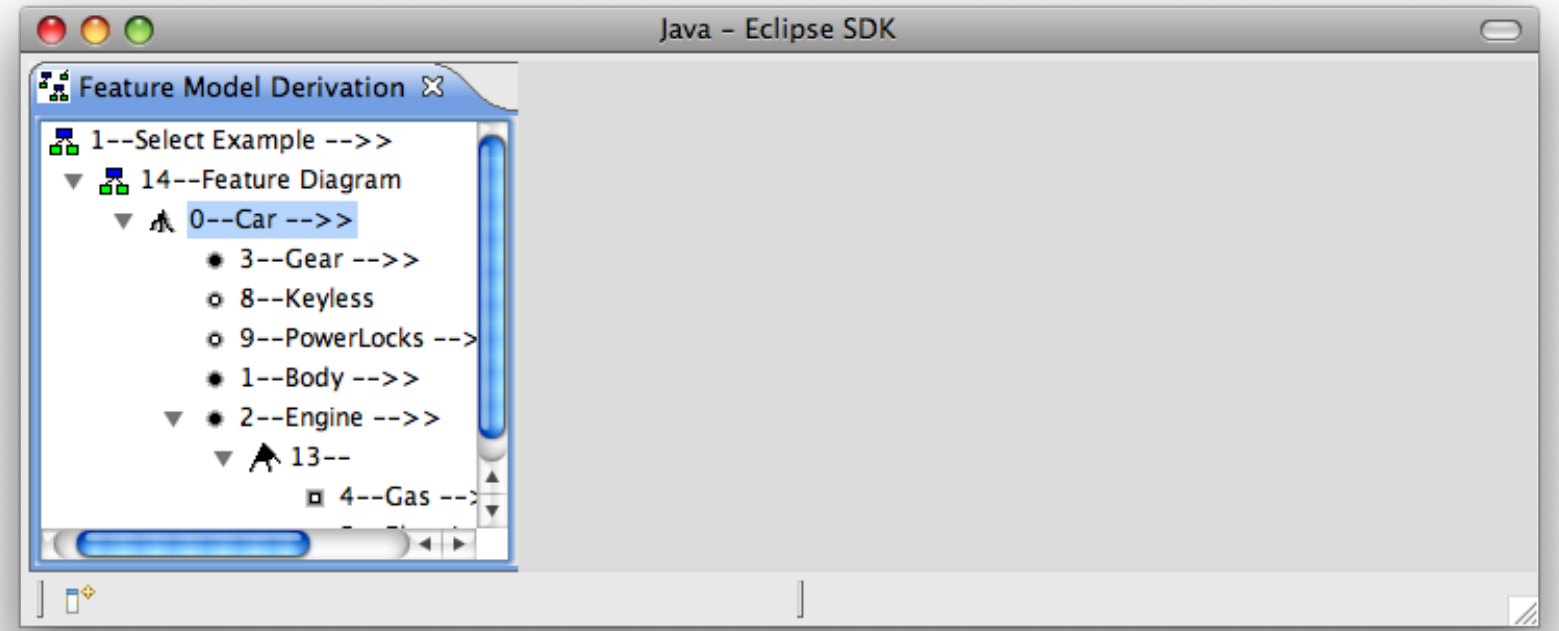
Feature Model approximating a constraint



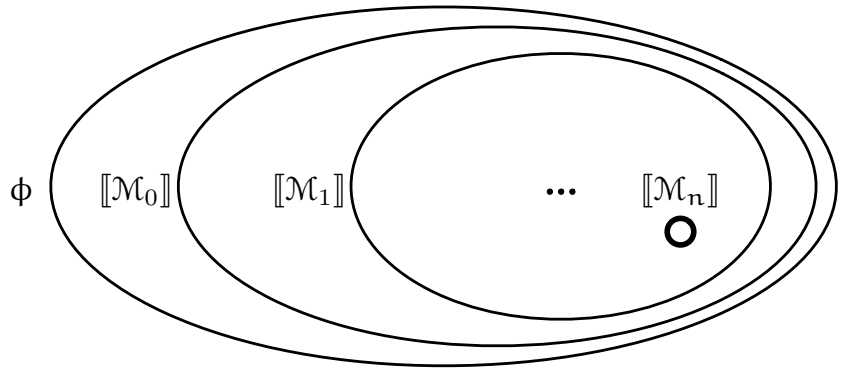
Feature Model approximating a constraint

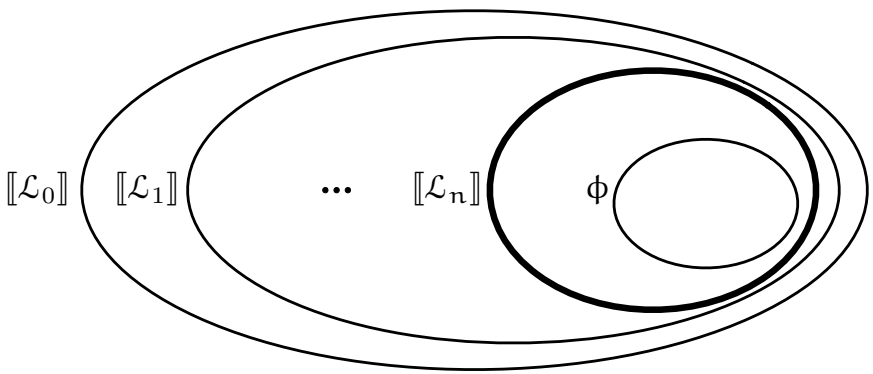


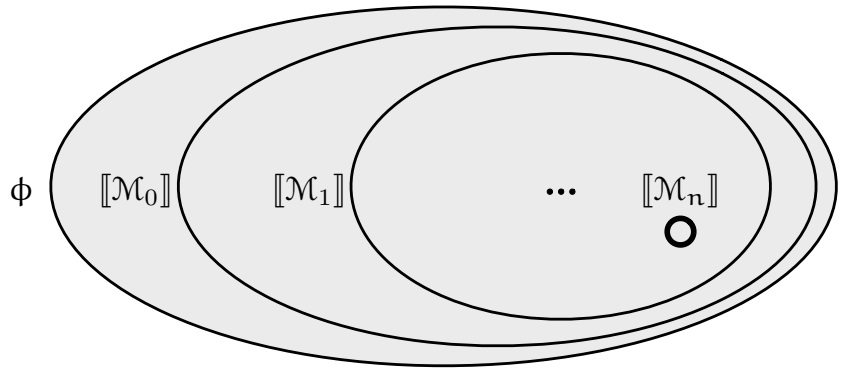
Feature Model approximating a constraint

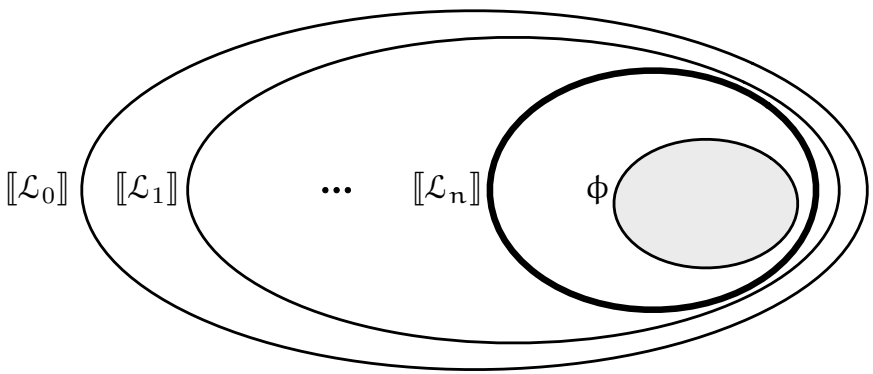


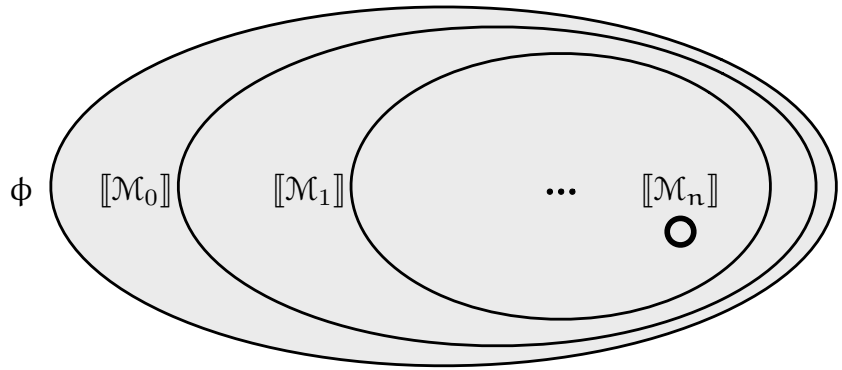
- The algorithm has the properties of **validity** and **exhaustiveness** of advice.
- The algorithm is efficient compared to approaches based on Constraint Satisfaction or Logic Programming.


ϕ
$[\![\mathcal{M}_0]\!]$
$[\![\mathcal{M}_1]\!]$
$\cdots$
$[\![\mathcal{M}_n]\!]$

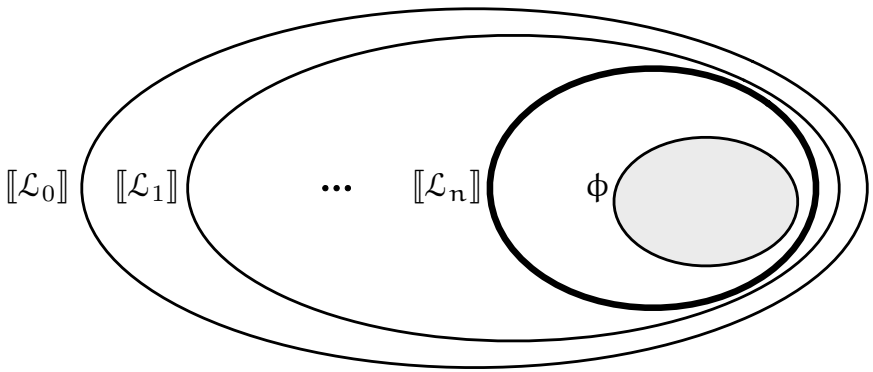

$[\![\mathcal{L}_0]\!]$
$[\![\mathcal{L}_1]\!]$
$\cdots$
$[\![\mathcal{L}_n]\!]$
ϕ


ϕ
$[\![\mathcal{M}_0]\!]$
$[\![\mathcal{M}_1]\!]$
$\cdots$
$[\![\mathcal{M}_n]\!]$


$[\![\mathcal{L}_0]\!]$
$[\![\mathcal{L}_1]\!]$
$\cdots$
$[\![\mathcal{L}_n]\!]$
ϕ



model $\rightarrow$ an instance



meta-model $\rightarrow$ model

semantics-preserving =

completeness-preserving

+ soundness-preserving

semantics-preserving =
completeness-preserving
+ soundness-preserving

Example: any refactoring

semantics-preserving =
completeness-preserving
+ soundness-preserving

Example: any refactoring
Example: feature **model** derivation

- the model comprises two components
  - feature diagram (M)
  - additional constraint (ψ)
- overall semantics must be preserved

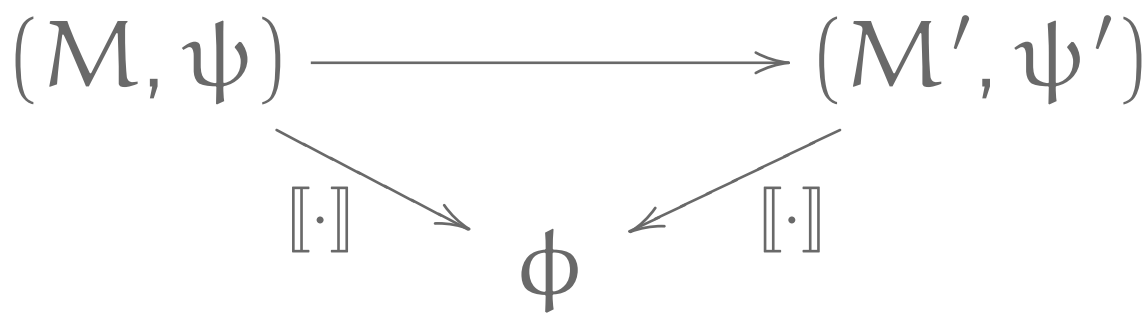
  "$M + \psi = M' + \psi'$"

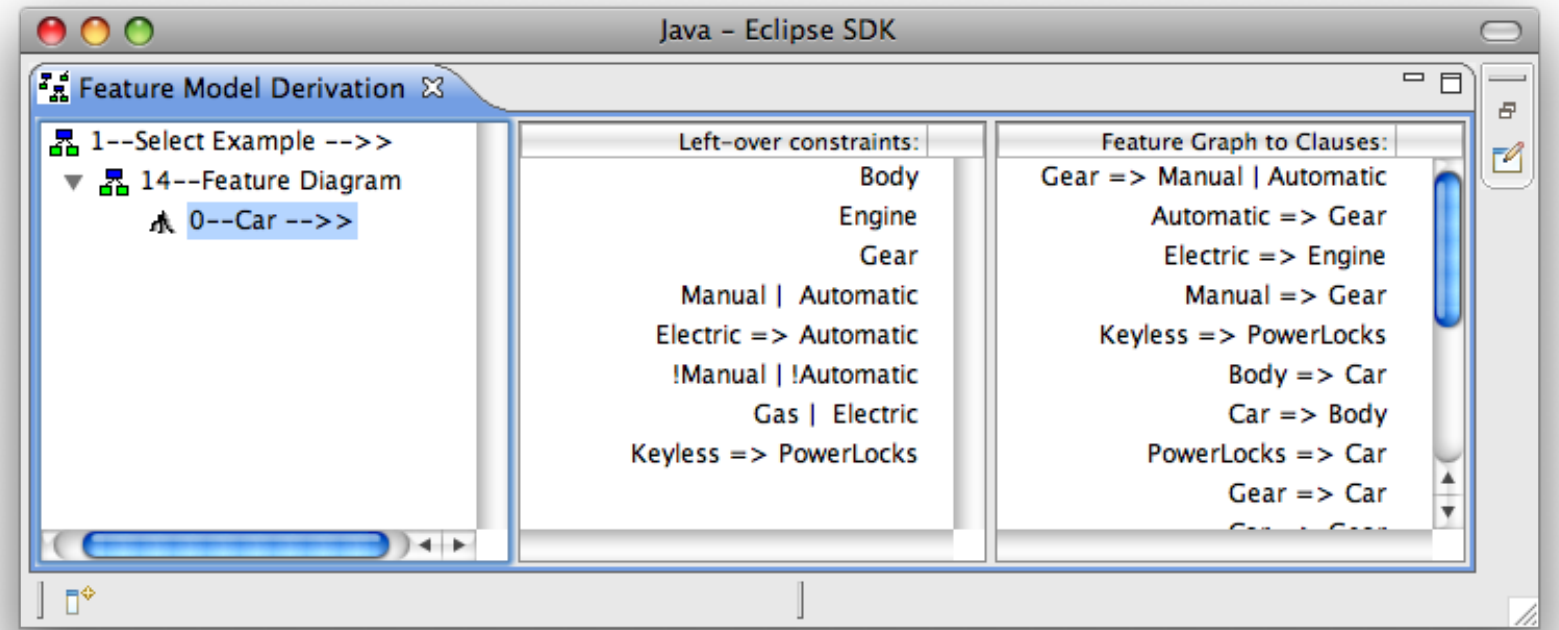
or

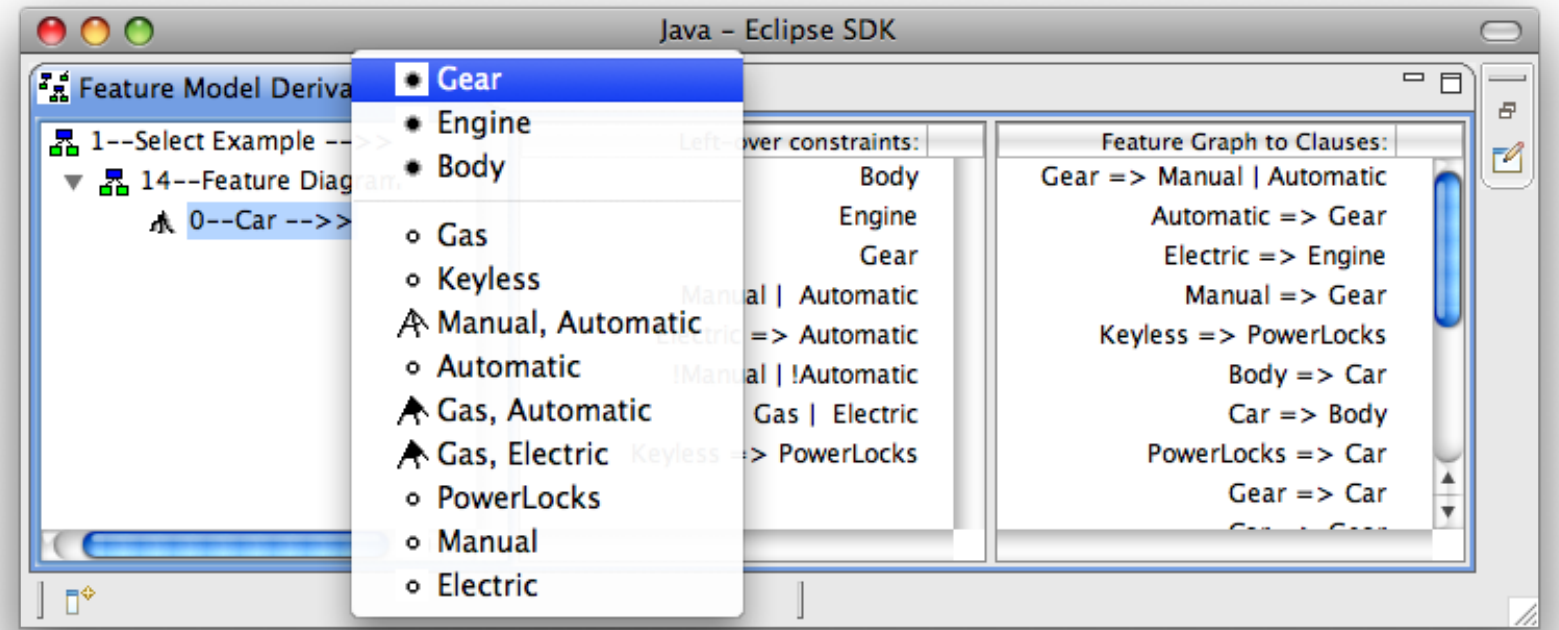
- the model comprises two components
  - feature diagram (M)
  - additional constraint (ψ)
- overall semantics must be preserved

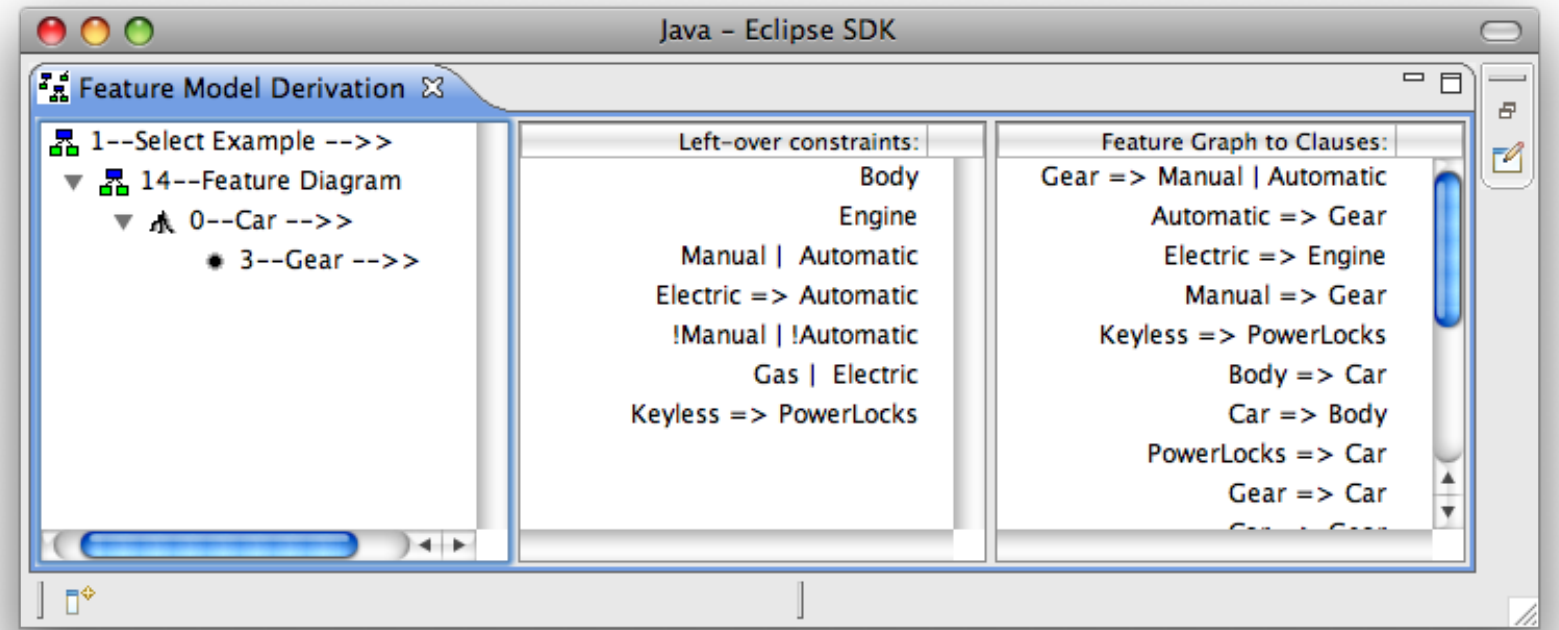
$$\text{“ } M + \psi = M' + \psi' \text{”}$$

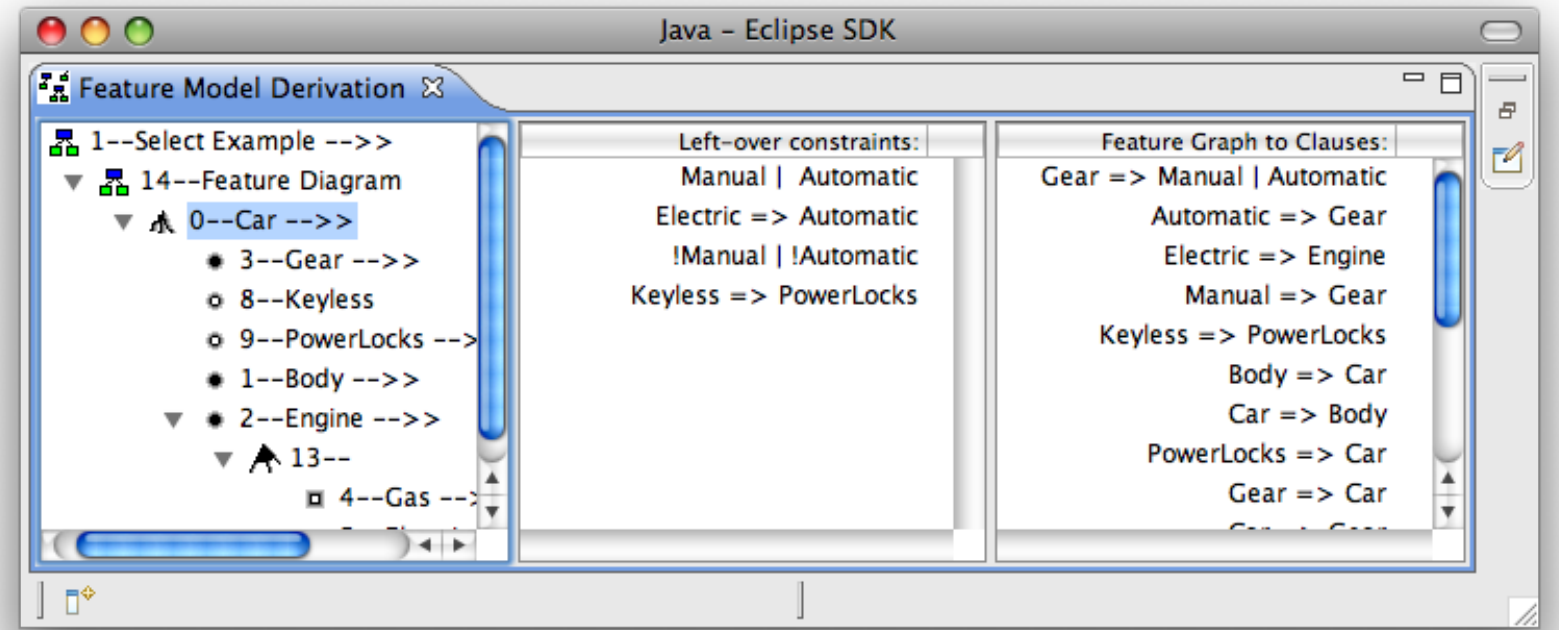
or

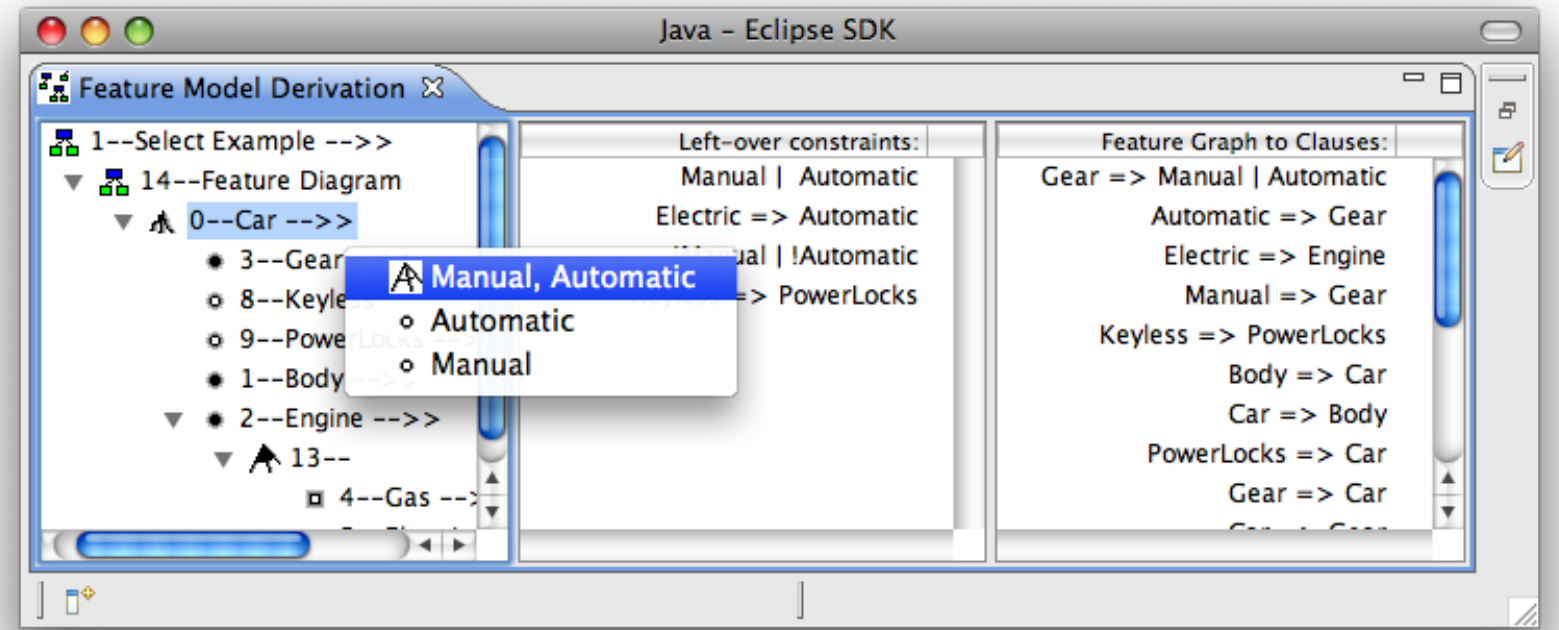


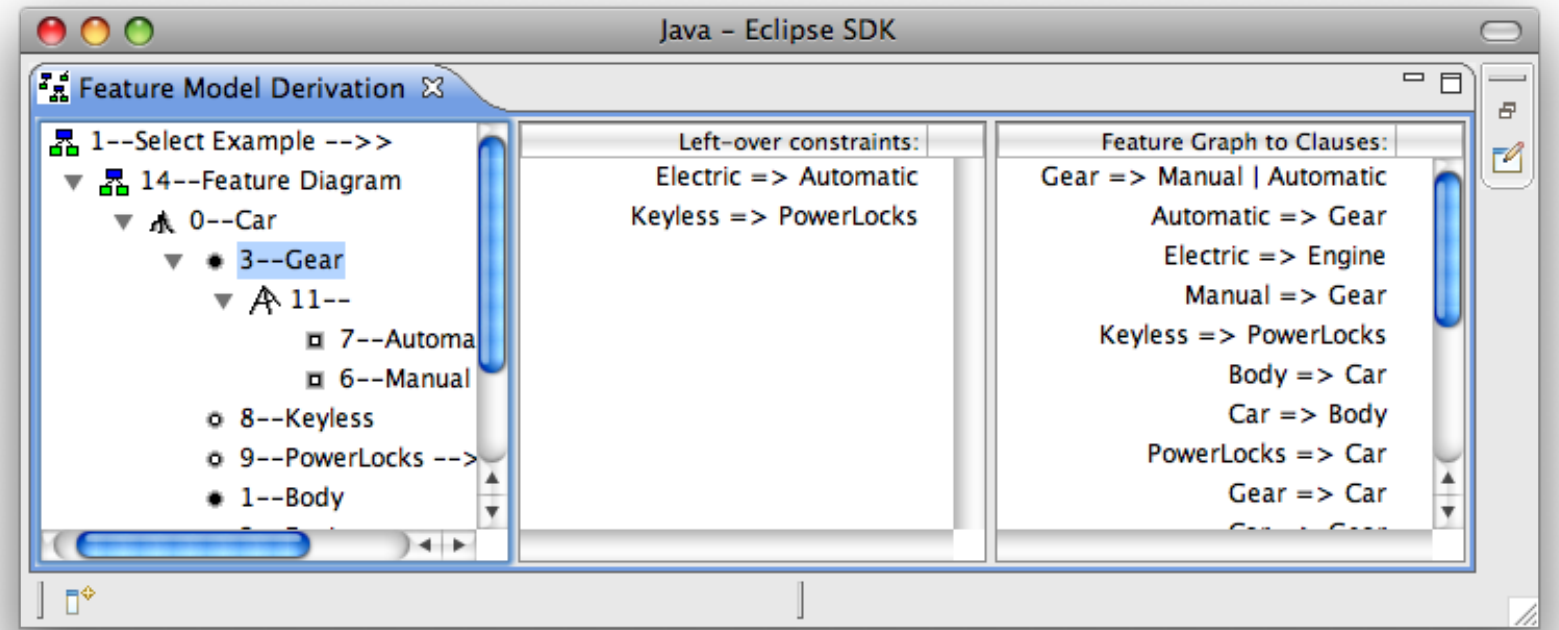

Java – Eclipse SDK
Feature Model Derivation
1--Select Example -->>
14--Feature Diagram
0--Car -->>
Left-over constraints:
Body
Engine
Gear
Manual | Automatic
Electric => Automatic
!Manual | !Automatic
Gas | Electric
Keyless => PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car


Java – Eclipse SDK
Feature Model Deriva
1--Select Example -->>
14--Feature Diag
0--Car -->>
Gear
Engine
Body
Gas
Keyless
Manual, Automatic
Automatic
Gas, Automatic
Gas, Electric
PowerLocks
Manual
Electric
ver constraints:
Body
Engine
Gear
al | Automatic
=> Automatic
al | !Automatic
Gas | Electric
> PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car


Java – Eclipse SDK
Feature Model Derivation
1--Select Example -->>
14--Feature Diagram
0--Car -->>
3--Gear -->>
Left-over constraints:
Body
Engine
Manual | Automatic
Electric => Automatic
!Manual | !Automatic
Gas | Electric
Keyless => PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car


Java – Eclipse SDK
Feature Model Derivation
1--Select Example -->>
14--Feature Diagram
0--Car -->>
3--Gear -->>
8--Keyless
9--PowerLocks -->
1--Body -->>
2--Engine -->>
13--
4--Gas -->
Left-over constraints:
Manual | Automatic
Electric => Automatic
!Manual | !Automatic
Keyless => PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car


Java – Eclipse SDK
Feature Model Derivation
1--Select Example -->>
14--Feature Diagram
0--Car -->>
3--Gear
8--Keyle
9--Powe
1--Body
2--Engine -->>
13--
4--Gas -->
Manual, Automatic
Automatic
Manual
Left-over constraints:
Manual | Automatic
Electric => Automatic
ual | !Automatic
=> PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car


Java – Eclipse SDK
Feature Model Derivation
1--Select Example -->>
14--Feature Diagram
0--Car
3--Gear
11--
7--Automa
6--Manual
8--Keyless
9--PowerLocks -->
1--Body
Left-over constraints:
Electric => Automatic
Keyless => PowerLocks
Feature Graph to Clauses:
Gear => Manual | Automatic
Automatic => Gear
Electric => Engine
Manual => Gear
Keyless => PowerLocks
Body => Car
Car => Body
PowerLocks => Car
Gear => Car

- **glossary**

  soundness-preserving derivation
  completeness-preserving derivation
  semantics-preserving derivation

  validity of advice
  exhaustiveness of advice

- Work this out for a rich subset of ECORE models, not only for Feature Models