

## TD/TP : Ligne de produits logiciels, feature model

Accéder à la page Web :

<http://familiar.viability.io/ide/familiar>

Vous pouvez commencer le travail et exécuter votre premier script :

```
1
2 // your FAMILIAR code here!
3 fm1 = FM (MDI : UML DesignPatterns (SI|IN) Instructors ;
4 Instructors : (Mathieu | Guillaume); Mathieu <-> IN ; Guillaume <-> SI; )
5 s1 = configs fm1
6 c1 = counting fm1
7
8
9 // fm2 = slice fm1 excluding { Mathieu }
10
11
12
```

Execute FAMILIAR code Reset

fm1 ▼  
s1 ▼  
c1 ▼

c1 = 2.0

Les « opérations » de FAMILIAR (configs, cores, counting, isValid, etc.) sont documentées ici:

<https://github.com/FAMILIAR-project/familiar-documentation/blob/master/manual/>

## Exercice 1. Syntaxe et sémantique des feature models (avec un outil)

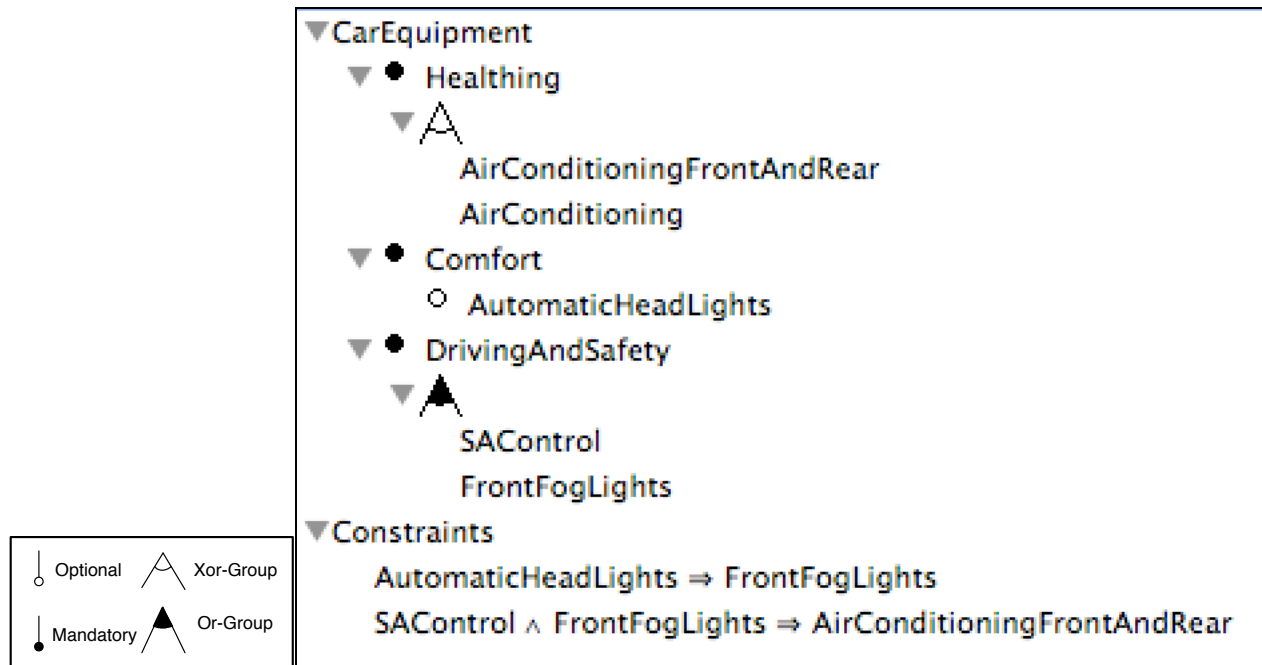
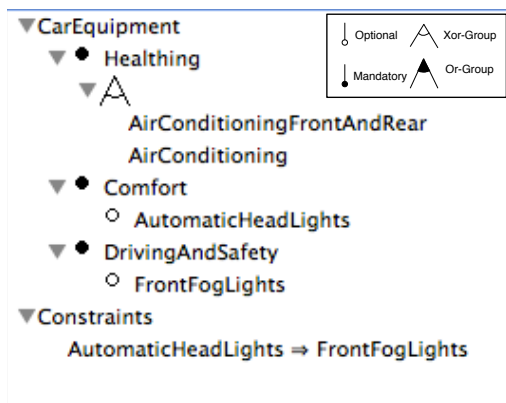


Figure 1.

**Question #0:** Spécifier le feature model de la Figure 1 avec FAMILIAR, en utilisant la notation interne textuelle : <https://github.com/FAMILIAR-project/familiar-documentation/blob/master/manual/featuremodel.md>



{CarEquipment, Comfort,  
DrivingAndSafety,  
Healthing}



{AirConditioning, FrontFogLights}  
{AutomaticHeadLights, AirConditioning,  
FrontFogLights}  
{AutomaticHeadLights, FrontFogLights,  
AirConditioningFrontAndRear}  
{AirConditioningFrontAndRear}  
{AirConditioning}  
{AirConditioningFrontAndRear, FrontFogLights}

```
fmCarEquipment = FM (CarEquipment : Healthing DrivingAndSafety Comfort ; // 3 mandatory features
    Healthing : (AirConditioning|AirConditioningFrontAndRear) ; // Xor
    DrivingAndSafety : [FrontFogLights] ; // optional
    Comfort : [AutomaticHeadLights] ; // optional
    // cross-tree constraints
    AutomaticHeadLights -> FrontFogLights ; )
```

```
fml> co = cores fmCarEquipment
co: (SET) {CarEquipment;Healthing;DrivingAndSafety;Comfort}
fml> fmCarEquipment.*
res6: (SET) {DrivingAndSafety;AirConditioningFrontAndRear;Comfort;Healthing;FrontFogLights;AirConditioning;AutomaticHeadLights;CarEquipment}
fml> setDiff fmCarEquipment.* co
res7: (SET) {AutomaticHeadLights;FrontFogLights;AirConditioning;AirConditioningFrontAndRear}
fml>
```

res1: (DOUBLE) 6.0

6

**Question #1:** Vérifier que votre spécification textuelle est conforme à la Figure 1 en produisant une énumération exhaustive de l'ensemble des configurations valides avec l'opération « configs »

**Question #2:** Quelles sont les features qui sont incluses dans n'importe quelle configuration? Utiliser l'opération « cores »

**Question #4:** Proposer un feature model avec une hiérarchie différente mais caractérisant le même ensemble de configurations. Vérifier le résultat en utilisant l'opération « compare »

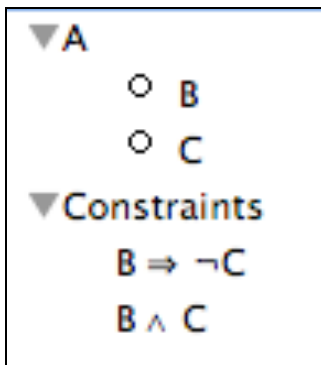


Figure 2

**Question #5:** Spécifier le feature model de la Figure 2 avec FAMILIAR. Vérifier que le nombre de configurations valides du feature model de la Figure 2 est 0 en utilisant l'opération « counting » et « isValid ».

**Question #6:** Que se passe-t-il si on relâche la première contrainte? Que se passe-t-il si on relâche la deuxième contrainte? Adresser la question avec les opérations de FAMILIAR.

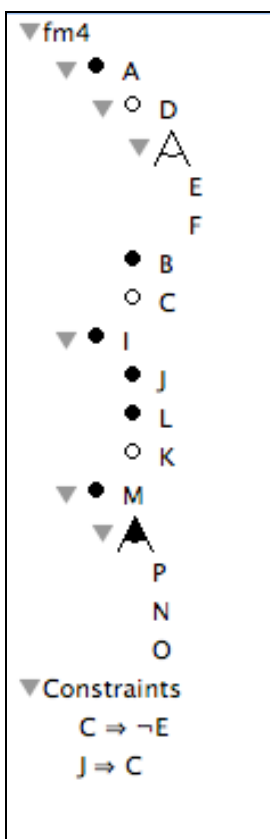


Figure 3

**Question #7:** Spécifier le feature model de la Figure 3 avec FAMILIAR.

Enumérez l'ensemble des configurations valides.

Que peut-on dire sur les features C et F ?

Que peut-on dire sur la feature E ?

Utiliser les opérations « deads » et « falseOptionals » pour vérifier vos dires.

**Question #8:** Corrigez les « anomalies », i.e., réécrire le feature model de manière à ce qu'il exprime le même ensemble de configuration mais cette fois-ci les informations de variabilité sont en adéquation avec les configurations valides du feature model.

Vérifier avec FAMILIAR que les anomalies sont bien corrigées, i.e., qu'elles ne sont plus présentes.

**Question #9:** Que se passe-t-il si on relâche la première contrainte? Que se passe-t-il si on relâche la deuxième contrainte? Réitérez la série de questions précédentes avec FAMILIAR pour chaque suppression de contrainte.



### Exercice 3. Des matrices aux feature models

Nous considérons les 2 matrices de comparaison ci-dessous (extraites de [https://en.wikipedia.org/wiki/Comparison\\_of\\_video\\_converters](https://en.wikipedia.org/wiki/Comparison_of_video_converters))

#### Overview [\[ edit \]](#)

Video converter	Developer	Licensing scheme	Supported platform			Website
			Windows	Mac OS X	Linux	
Any Video Converter	AVCLabs	Freeware	Yes	Yes	No	<a href="http://any-video-converter.com">any-video-converter.com</a>
Audials Tunebite 10 Platinum	Audials	Shareware	Yes	Yes	No	<a href="http://audials.com">audials.com</a>
Avidemux	Mean,Gruntster,Fahr	Free and open-source	Yes	Yes	Yes	<a href="http://www.avidemux.org">www.avidemux.org</a>
Dr. DivX	DivX, Inc.	Free and open-source	Yes	Yes	No	<a href="http://labs.divx.com/DrDivX">labs.divx.com/DrDivX</a>
DVDVideoSoft Free Studio	DVDVideoSoft	Freeware (ad supported)	Yes	No	No	<a href="http://dvdvideosoft.com">dvdvideosoft.com</a>
FFmpeg	FFmpeg project	Free and open-source	Yes	Yes	Yes	<a href="http://ffmpeg.org">ffmpeg.org</a>
FormatFactory	Chen Jun Hao	Freeware (ad supported)	Yes	No	No	<a href="http://formatoz.com">formatoz.com</a>
Freemake Video Converter	Freemake	Freeware (ad supported)	Yes	No	No	<a href="http://freemake.com">freemake.com</a>
HandBrake	Handbrake Project	Free and open-source	Yes	Yes	Yes	<a href="http://handbrake.fr">handbrake.fr</a>
MediaCoder	Stanley Huang	Freeware (ad supported)	Yes	No	No	<a href="http://www.mediacoderhq.com">www.mediacoderhq.com</a>
MEncoder	The MPlayer Project	Free and open-source	Yes	Yes	Yes	<a href="http://mplayerhq.hu">mplayerhq.hu</a>
OggConvert	Tristan Brindle	Free and open-source	Experimental	No	Yes	<a href="http://oggconvert.tristanb.net">oggconvert.tristanb.net</a>
Prism Video Converter	NCH Software	Freeware (and paid version)	Yes	Yes	No	<a href="http://nchsoftware.com/prism/">nchsoftware.com/prism/</a>
SUPER	eRightSoft	Freeware (ad supported)	Yes	No	No	<a href="http://www.erightssoft.com">www.erightssoft.com</a>
Transcode	Transcode Team	Free and open-source	No	No	Yes	<a href="http://transcoding.org">transcoding.org</a>
VirtualDub	Avery Lee	Free and open-source	Yes	No	No	<a href="http://virtualdubmod.sourceforge.net">virtualdubmod.sourceforge.net</a>
XMedia Recode	Sebastian Dörfler	Freeware	Yes	No	No	<a href="http://www.xmedia-recode.de">www.xmedia-recode.de</a>

#### Input [\[ edit \]](#)

Video converter	Supported input container formats														
	3GP	AVI	Blu-ray video	DVD video	FLV	Matroska	MP4	MPEG-PS	Ogg	QuickTime	SVCD	TS	TOD	VCD	WMV
Any Video Converter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Audials Tunebite 10 Platinum	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Avidemux	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FFmpeg	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Freemake Video Converter	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FormatFactory	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DVDVideoSoft Free Studio	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
HandBrake	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SUPER <sup>[1]</sup>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Prism Video Converter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
XMedia Recode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

**Question #13 :** L'objectif est de produire un feature model qui correspond à ces deux matrices de produit (donc on veut un seul feature model). A toute configuration valide de ce feature model correspondra au moins un produit. Les « features » seront donc issues des les 2 matrices. Avant d'élaborer le feature model, il faut s'interroger (1) sur la pertinence de certaines caractéristiques (e.g., Website) ; (2) la manière de structurer l'information sous forme de hiérarchie. Attention également car certains produits ne sont que dans une matrice (en première approximation, on s'intéressera uniquement aux produits qui sont dans les deux matrices)

## Exercice 4. JHipster

L'énoncé est en ligne : <http://tinyurl.com/varyhipster1617>