Software Variability and Artificial Intelligence

Mathieu Acher (Associate Professor)

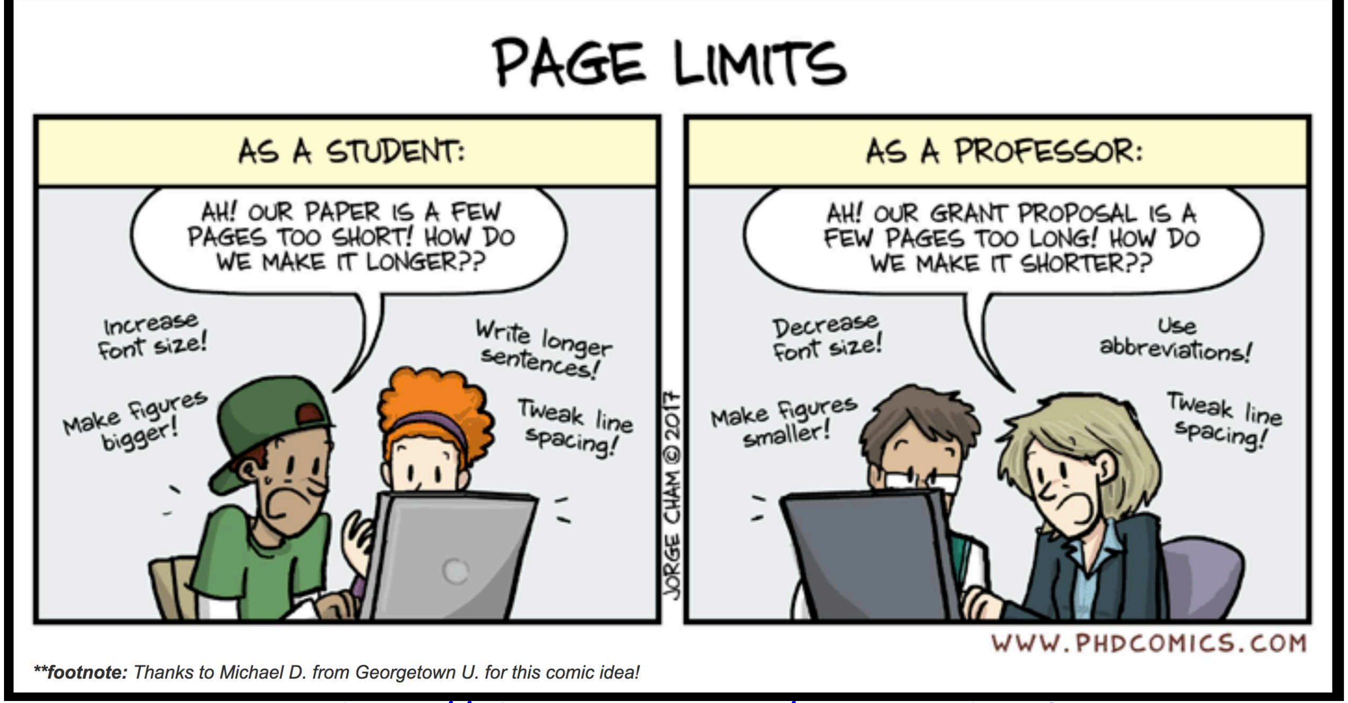
https://www.mathieuacher.com

https://teaching.variability.io

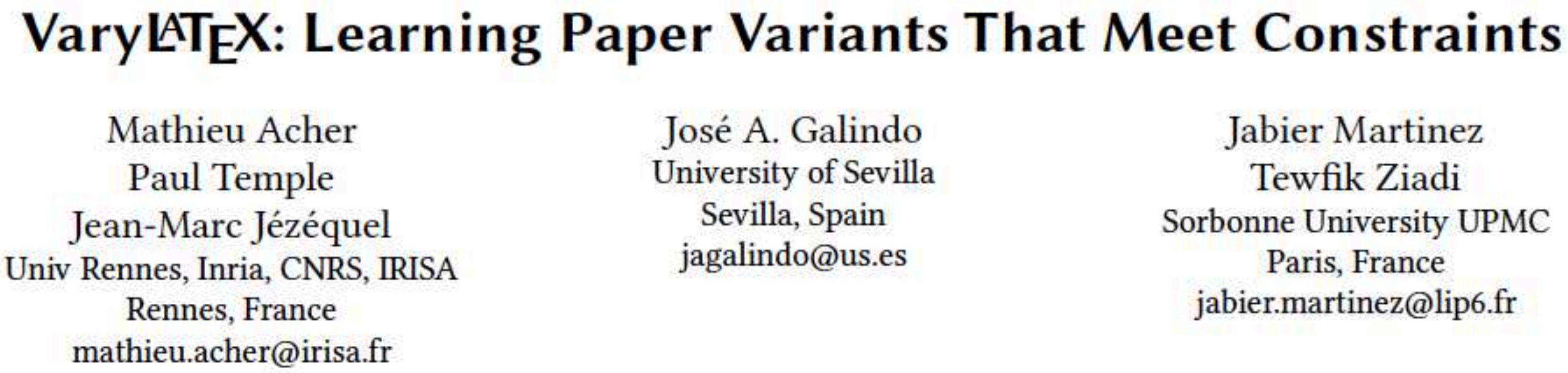
https://varyvary.github.io/

Disclaimer

- Slides for the EJCP 2018 course
  - ~French summer school for PhD candidates in programming, verification, software engineering, etc.
- Abstract: *Most modern software systems are subject to variation or come in many variants. Web browsers like Firefox or Chrome are available on different operating systems, in different languages, while users can configure 2000+ preferences or install numerous 3rd parties extensions (or plugins). Web servers like Apache, operating systems like the Linux kernel, or a video encoder like x264 are other examples of software systems that are highly configurable at compile-time or at run-time for delivering the expected functionality andmeeting the various desires of users. Variability ("the ability of a software system or artifact to be efficiently extended, changed,customized or configured for use in a particular context") is therefore a crucial property of software systems. Organizations capable of mastering variability can deliver high-quality variants (or products) in a short amount of time and thus attract numerous customers, new use-cases or usage contexts. A hard problem for end-users or software developers is to master the combinatorial explosion induced by variability: Hundreds of configuration options can be combined, each potentially with distinct functionality and effects on execution time, memory footprint, quality of the result, etc. The first part of this course will introduce variability-intensive systems, their applications and challenges, in various software contexts. We will use intuitive examples (like a generator of LaTeX paper variants) and real-world systems (like the Linux kernel). A second objective of this course is to show the relevance of Artificial Intelligence (AI) techniques for exploring and taming such enormous variability spaces. In particular, we will introduce how (1) satisfiability and constraint programming solvers can be used to properly model and reason about variability; (2) how machine learning can be used to discover constraints and predict the variability behavior of configurable systems or software product lines.*
- https://ejcp2018.sciencesconf.org/resource/page/id/5
- I had 45 minutes + 105 minutes (less than 3 hours)
- Some results have not been published yet



http://phdcomics.com/comics.php?f=1971



VaryLaTeX: Learning Paper Variants That Meet Constraints

Mathieu Acher
Paul Temple
Jean-Marc Jézéquel
Univ Rennes, Inria, CNRS, IRISA
Rennes, France
mathieu.acher@irisa.fr

José A. Galindo
University of Sevilla
Sevilla, Spain
jagalindo@us.es

Jabier Martinez
Tewfik Ziadi
Sorbonne University UPMC
Paris, France
jabier.martinez@lip6.fr

Successfully submitted for VaMoS'18

(on time and meeting formatting instructions)

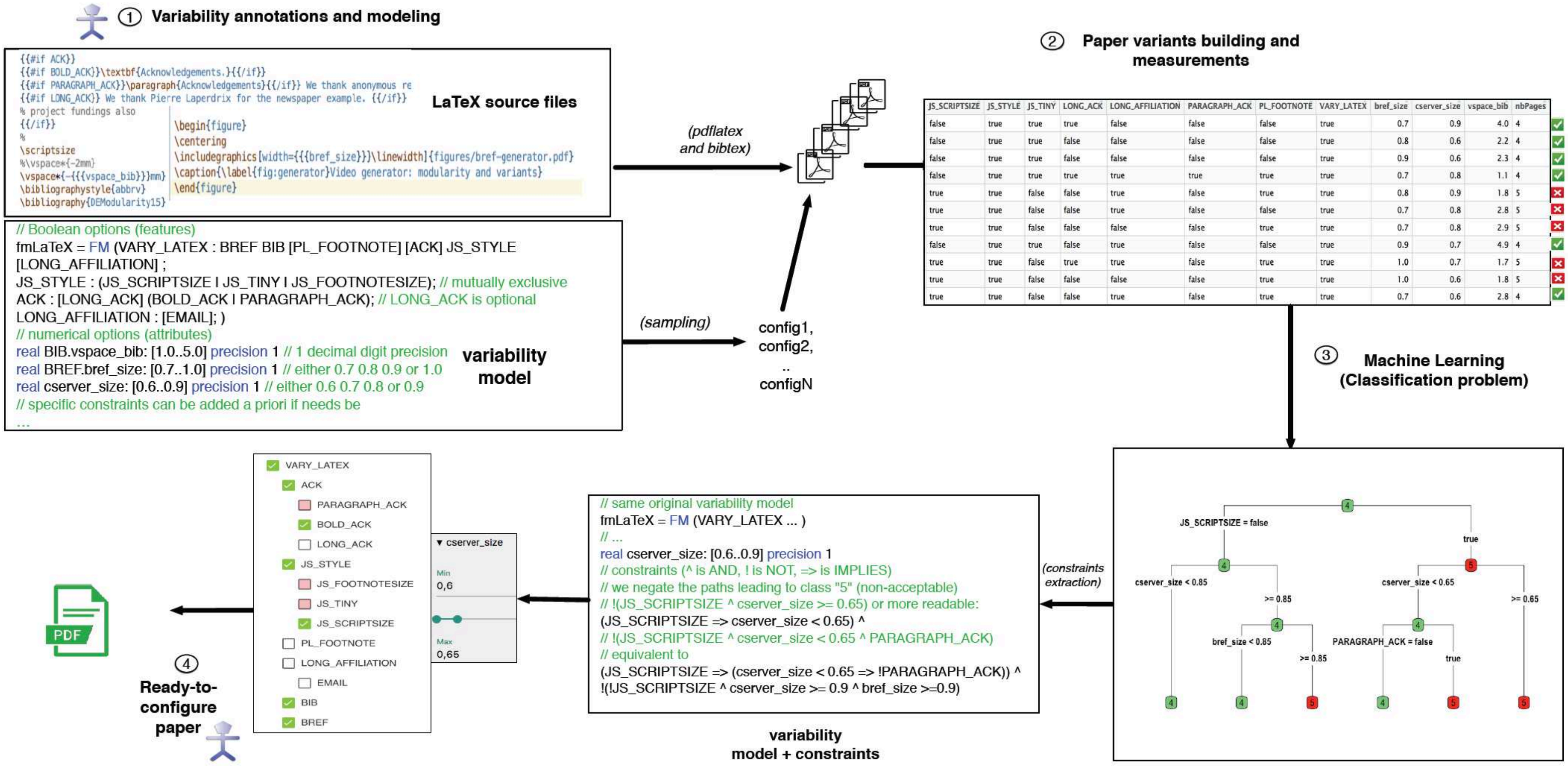
and then accepted

(live demonstration)

Two case studies

- FSE paper (see demonstration)
  - Page limit: 4
  - Accuracy: ~85% with 40 papers in the training set (there are 73,440 valid configurations)
- Curiculum vitae generation
  - 18 pages limit; 5 Boolean options; full generation, only 32 papers (not need to learn here)

Process



① **Variability annotations and modeling**

LaTeX source files

```
{{#if ACK}}
{{#if BOLD_ACK}}\textbf{Acknowledgements.}{{/if}}
{{#if PARAGRAPH_ACK}}\paragraph{Acknowledgements}{{/if}} We thank anonymous re
{{#if LONG_ACK}} We thank Pierre Laperdrix for the newspaper example. {{/if}}
% project fundings also
{{/if}}
%
\scriptsize
%\vspace*{-2mm}
\vspace*{-{{{vspace_bib}}}mm}
\bibliographystyle{abbrv}
\bibliography{DEModularity15}
```

```
\begin{figure}
\centering
\includegraphics[width={{{bref_size}}}\linewidth]{figures/bref-generator.pdf}
\caption{\label{fig:generator}Video generator: modularity and variants}
\end{figure}
```

variability model

```
// Boolean options (features)
fmLaTeX = FM (VARY_LATEX : BREF BIB [PL_FOOTNOTE] [ACK] JS_STYLE
[LONG_AFFILIATION] ;
JS_STYLE : (JS_SCRIPTSIZE I JS_TINY I JS_FOOTNOTESIZE); // mutually exclusive
ACK : [LONG_ACK] (BOLD_ACK I PARAGRAPH_ACK); // LONG_ACK is optional
LONG_AFFILIATION : [EMAIL]; )
// numerical options (attributes)
real BIB.vspace_bib: [1.0..5.0] precision 1 // 1 decimal digit precision
real BREF.bref_size: [0.7..1.0] precision 1 // either 0.7 0.8 0.9 or 1.0
real cserver_size: [0.6..0.9] precision 1 // either 0.6 0.7 0.8 or 0.9
// specific constraints can be added a priori if needs be
...
```

(pdflatex and bibtex)

(sampling)

config1,
config2,
..
configN

② **Paper variants building and measurements**

JS_SCRIPTSIZE	JS_STYLE	JS_TINY	LONG_ACK	LONG_AFFILIATION	PARAGRAPH_ACK	PL_FOOTNOTE	VARY_LATEX	bref_size	cserver_size	vspace_bib	nbPages	
false	true	true	true	false	false	false	true	0.7	0.9	4.0	4	✓
false	true	true	false	false	false	false	true	0.8	0.6	2.2	4	✓
false	true	true	false	false	false	false	true	0.9	0.6	2.3	4	✓
false	true	true	true	true	true	true	true	0.7	0.8	1.1	4	✓
true	true	false	false	true	false	true	true	0.8	0.9	1.8	5	✗
true	true	false	false	true	false	false	true	0.7	0.8	2.8	5	✗
true	true	false	false	false	false	true	true	0.7	0.8	2.9	5	✗
false	true	true	false	true	false	false	true	0.9	0.7	4.9	4	✓
true	true	false	true	true	false	true	true	1.0	0.7	1.7	5	✗
true	true	false	false	false	false	true	true	1.0	0.6	1.8	5	✗
true	true	false	false	true	false	true	true	0.7	0.6	2.8	4	✓

③ **Machine Learning (Classification problem)**

(constraints extraction)

```
// same original variability model
fmLaTeX = FM (VARY_LATEX ... )
// ...
real cserver_size: [0.6..0.9] precision 1
// constraints (^ is AND, ! is NOT, => is IMPLIES)
// we negate the paths leading to class "5" (non-acceptable)
// !(JS_SCRIPTSIZE ^ cserver_size >= 0.65) or more readable:
(JS_SCRIPTSIZE => cserver_size < 0.65) ^
// !(JS_SCRIPTSIZE ^ cserver_size < 0.65 ^ PARAGRAPH_ACK)
// equivalent to
(JS_SCRIPTSIZE => (cserver_size < 0.65 => !PARAGRAPH_ACK)) ^
!(!JS_SCRIPTSIZE ^ cserver_size >= 0.9 ^ bref_size >=0.9)
```

variability model + constraints

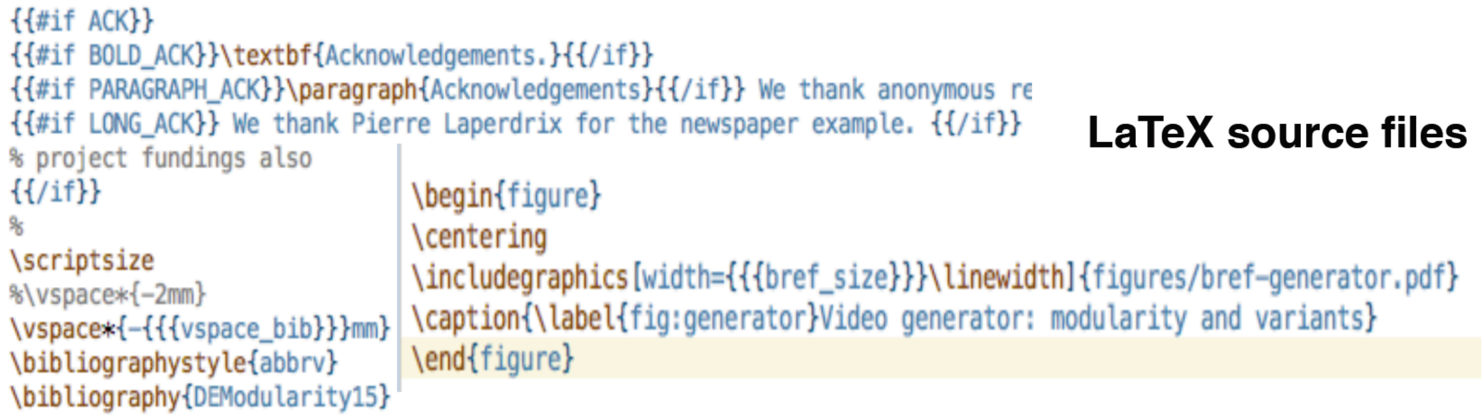
- ☑ VARY_LATEX
  - ☑ ACK
    - 🟥 PARAGRAPH_ACK
    - ☑ BOLD_ACK
    - □ LONG_ACK
  - ☑ JS_STYLE
    - 🟥 JS_FOOTNOTESIZE
    - 🟥 JS_TINY
    - ☑ JS_SCRIPTSIZE
  - □ PL_FOOTNOTE
  - □ LONG_AFFILIATION
    - □ EMAIL
  - ☑ BIB
  - ☑ BREF

▼ cserver_size
Min
0,6
Max
0,65

④ **Ready-to-configure paper**

AI#1 Logic, satisfiability, constraints, reasoning, solving

① **Variability annotations and modeling**



```
{{#if ACK}}
{{#if BOLD_ACK}}\textbf{Acknowledgements.}{{/if}}
{{#if PARAGRAPH_ACK}}\paragraph{Acknowledgements}{{/if}} We thank anonymous re
{{#if LONG_ACK}} We thank Pierre Laperdrix for the newspaper example. {{/if}}
% project fundings also
{{/if}}
%
\scriptsize
%\vspace*{-2mm}
\vspace*{-{{{vspace_bib}}}mm}
\bibliographystyle{abbrv}
\bibliography{DEModularity15}
```

```
\begin{figure}
\centering
\includegraphics[width={{{bref_size}}}\linewidth]{figures/bref-generator.pdf}
\caption{\label{fig:generator}Video generator: modularity and variants}
\end{figure}
```

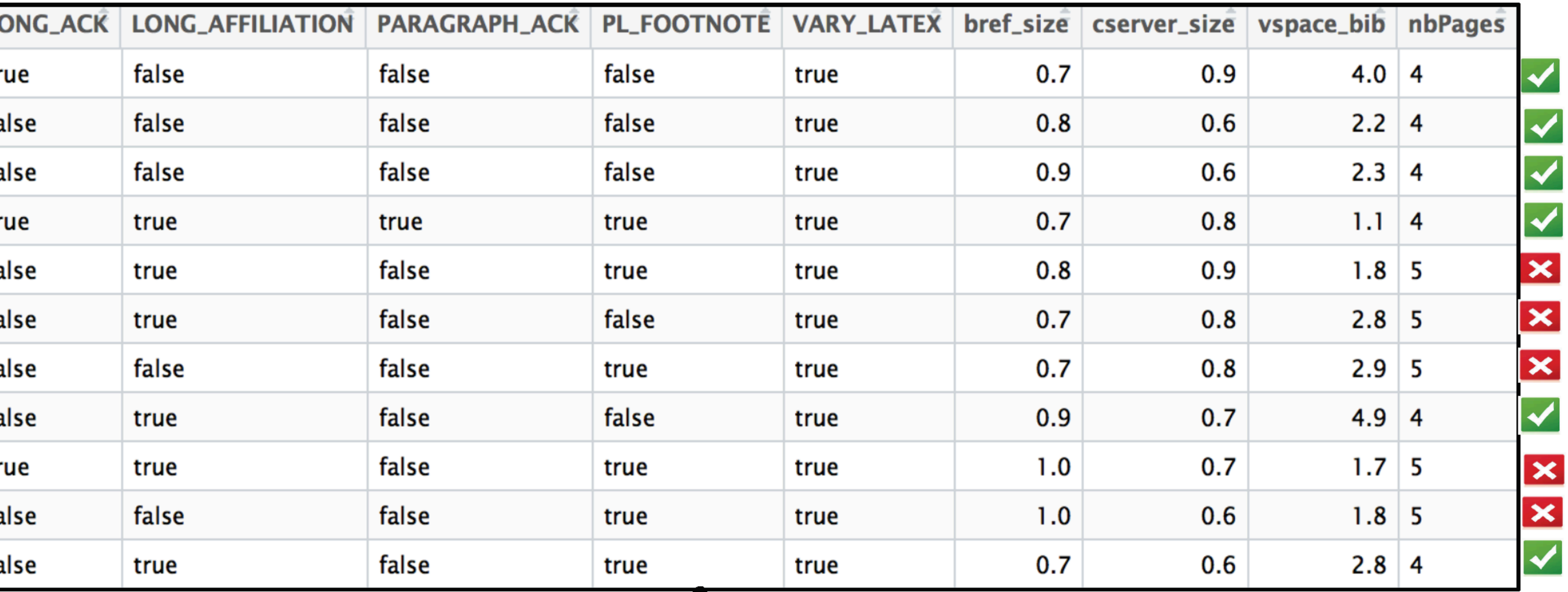
LaTeX source files

```
// Boolean options (features)
fmLaTeX = FM (VARY_LATEX : BREF BIB [PL_FOOTNOTE] [ACK] JS_STYLE
[LONG_AFFILIATION] ;
JS_STYLE : (JS_SCRIPTSIZE I JS_TINY I JS_FOOTNOTESIZE); // mutually exclusive
ACK : [LONG_ACK] (BOLD_ACK I PARAGRAPH_ACK); // LONG_ACK is optional
LONG_AFFILIATION : [EMAIL]; )
// numerical options (attributes)
real BIB.vspace_bib: [1.0..5.0] precision 1 // 1 decimal digit precision
real BREF.bref_size: [0.7..1.0] precision 1 // either 0.7 0.8 0.9 or 1.0
real cserver_size: [0.6..0.9] precision 1 // either 0.6 0.7 0.8 or 0.9
// specific constraints can be added a priori if needs be
…
```

variability model

AI#2 Statistical, supervised machine learning (classification problem)

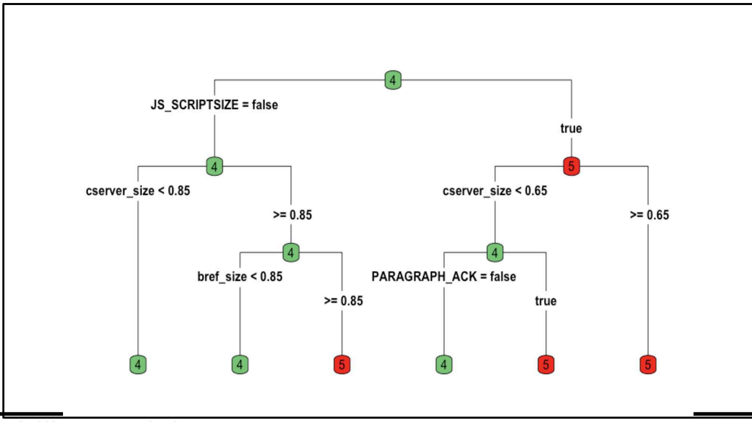
Paper variants building and measurements

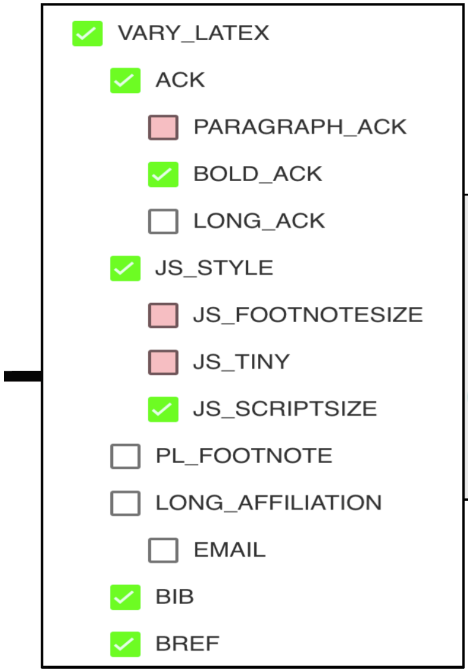


ONG_ACK	LONG_AFFILIATION	PARAGRAPH_ACK	PL_FOOTNOTE	VARY_LATEX	bref_size	cserver_size	vspace_bib	nbPages	
ue	false	false	false	true	0.7	0.9	4.0	4	✓
alse	false	false	false	true	0.8	0.6	2.2	4	✓
alse	false	false	false	true	0.9	0.6	2.3	4	✓
ue	true	true	true	true	0.7	0.8	1.1	4	✓
alse	true	false	true	true	0.8	0.9	1.8	5	✗
alse	true	false	false	true	0.7	0.8	2.8	5	✗
alse	false	false	true	true	0.7	0.8	2.9	5	✗
alse	true	false	false	true	0.9	0.7	4.9	4	✓
ue	true	false	true	true	1.0	0.7	1.7	5	✗
alse	false	false	true	true	1.0	0.6	1.8	5	✗
alse	true	false	true	true	0.7	0.6	2.8	4	✓

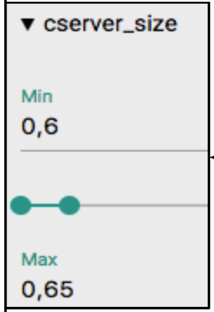
#AI1 + #AI2

Specialization of the variability model





☑ VARY_LATEX
- ☑ ACK
  - 🟥 PARAGRAPH_ACK
  - ☑ BOLD_ACK
  - □ LONG_ACK
- ☑ JS_STYLE
  - 🟥 JS_FOOTNOTESIZE
  - 🟥 JS_TINY
  - ☑ JS_SCRIPTSIZE
- □ PL_FOOTNOTE
- □ LONG_AFFILIATION
  - □ EMAIL
- ☑ BIB
- ☑ BREF



▼ cserver_size

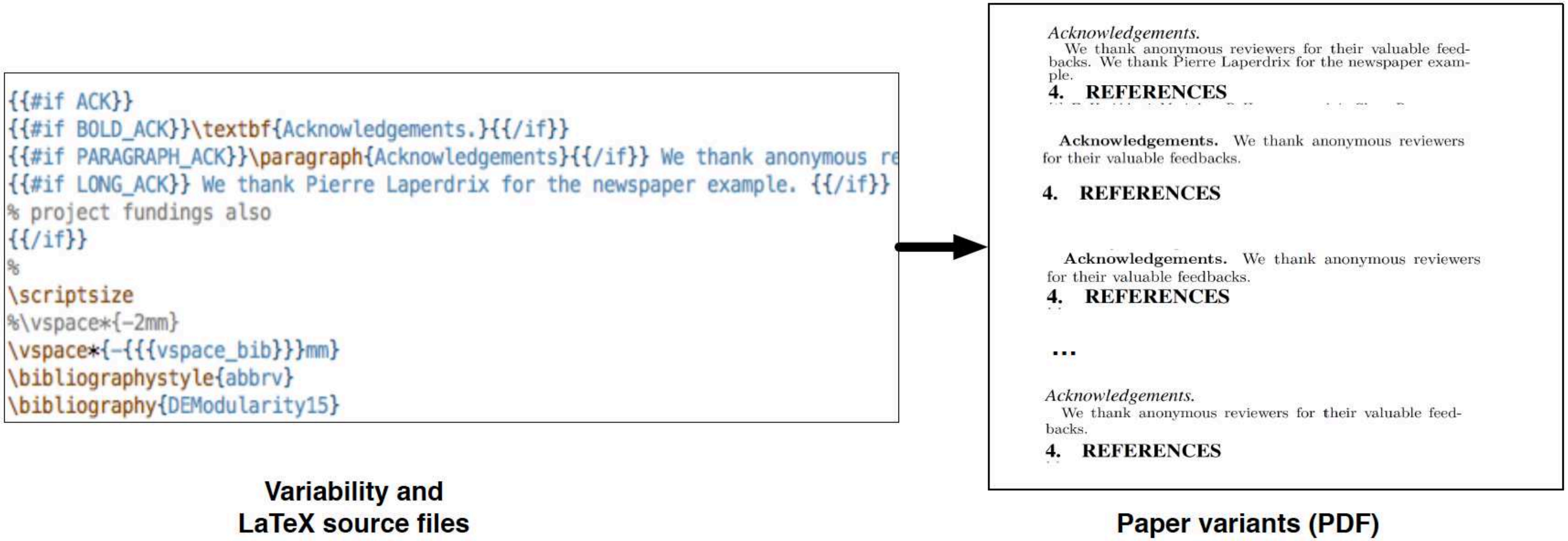
Min: 0,6

Max: 0,65

```
// same original variability model
fmLaTeX = FM (VARY_LATEX ... )
// ...
real cserver_size: [0.6..0.9] precision 1
// constraints (^ is AND, ! is NOT, => is IMPLIES)
// we negate the paths leading to class "5" (non-acceptable)
// !(JS_SCRIPTSIZE ^ cserver_size >= 0.65) or more readable:
(JS_SCRIPTSIZE => cserver_size < 0.65) ^
// !(JS_SCRIPTSIZE ^ cserver_size < 0.65 ^ PARAGRAPH_ACK)
// equivalent to
(JS_SCRIPTSIZE => (cserver_size < 0.65 => !PARAGRAPH_ACK)) ^
!(!JS_SCRIPTSIZE ^ cserver_size >= 0.9 ^ bref_size >=0.9)
```

variability

https://github.com/FAMILIAR-project/varylatex/



```
{{#if ACK}}
{{#if BOLD_ACK}}\textbf{Acknowledgements.}{{/if}}
{{#if PARAGRAPH_ACK}}\paragraph{Acknowledgements}{{/if}} We thank anonymous re
{{#if LONG_ACK}} We thank Pierre Laperdrix for the newspaper example. {{/if}}
% project fundings also
{{/if}}
%
\scriptsize
%\vspace*{-2mm}
\vspace*{-{{{vspace_bib}}}mm}
\bibliographystyle{abbrv}
\bibliography{DEModularity15}
```

Acknowledgements.
We thank anonymous reviewers for their valuable feedbacks. We thank Pierre Laperdrix for the newspaper example.

4. REFERENCES

Acknowledgements. We thank anonymous reviewers for their valuable feedbacks.

4. REFERENCES

Acknowledgements. We thank anonymous reviewers for their valuable feedbacks.

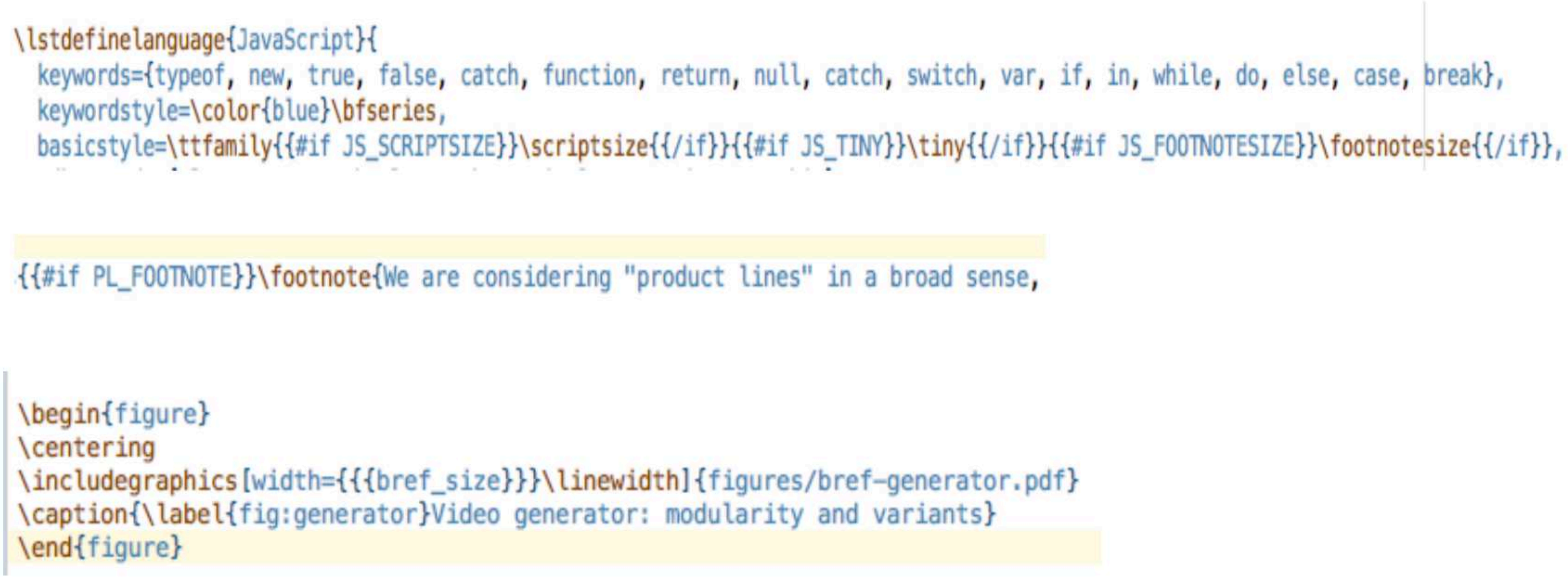
4. REFERENCES

...

Acknowledgements.
We thank anonymous reviewers for their valuable feedbacks.

4. REFERENCES

(a) Variability annotations and excerpt of some possible paper variants



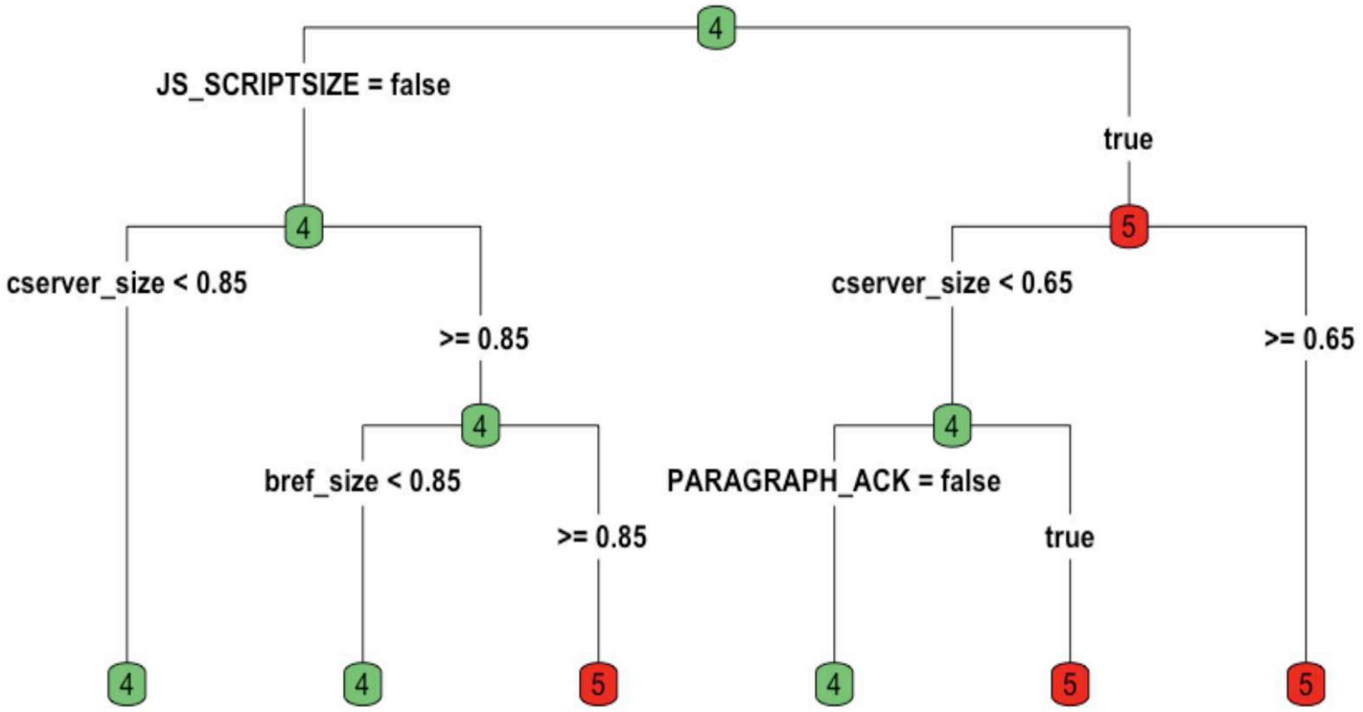
```
\lstdefinelanguage{JavaScript}{
  keywords={typeof, new, true, false, catch, function, return, null, catch, switch, var, if, in, while, do, else, case, break},
  keywordstyle=\color{blue}\bfseries,
  basicstyle=\ttfamily{{#if JS_SCRIPTSIZE}}\scriptsize{{/if}}{{#if JS_TINY}}\tiny{{/if}}{{#if JS_FOOTNOTESIZE}}\footnotesize{{/if}},
```

```
{{#if PL_FOOTNOTE}}\footnote{We are considering "product lines" in a broad sense,
```

```
\begin{figure}
\centering
\includegraphics[width={{{bref_size}}}\linewidth]{figures/bref-generator.pdf}
\caption{\label{fig:generator}Video generator: modularity and variants}
\end{figure}
```

(b) Users can vary the font size of a code snippet, activate a footnote, vary the font size of a figure, *etc.*

Classification tree



Agenda

- Software Variability: An Overview
  - VaryLaTeX
  - Linux, video generator, 3D printing, etc.
  - Testing 26K+ configurations of JHipster
- AI1: Modeling and Reasoning about Variability
  - Feature models: syntax, semantics, and logics
- AI2: Learning Variability
  - Statistical supervised machine learning
- AI for fitting Software Variability

VaryLaTeX

an instance of a more general problem

(and solution based on artificial intelligence and software engineering techniques)

Variability

- "the **ability** of a software system or artifact to be efficiently extended, changed, customized or configured for use in a particular context" (Svahnberg et al. 2005)
  - software/**customization** perspective

- Terminology
  - Software product lines, configurable systems, variability-intensive systems
  - Options, features, variation points

Software Variability

- Configurable system

VaryLaTeX

- Configuration options (aka software features)

template variables of a LaTeX file

- Variants

LaTeX source and PDF variants (papers)

- Large variability spaces

73,440 possible variants

Software Variability

- Configurable system

Linux operating system

- Configuration options (aka software features)

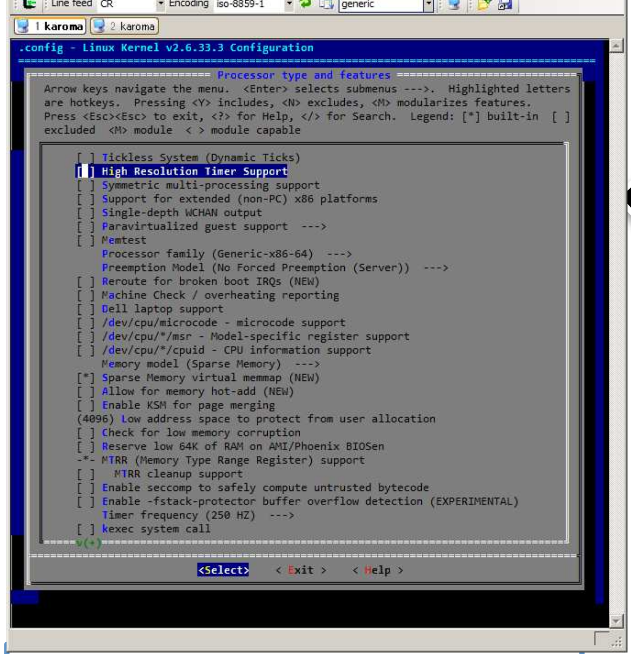
conditional compilation (#ifdef) in C files

- Variants

Linux kernel variants

- Large variability spaces

16,000 options (~“yes”, “no”, “module”)



Line feed CR Encoding iso-8859-1 generic

```
.config - Linux Kernel v2.6.33.3 Configuration
 ==============================================================================
  ======================== Processor type and features ========================
    Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters
    are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.
    Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
    excluded  <M> module  < > module capable

          [ ] Tickless System (Dynamic Ticks)
          [ ] High Resolution Timer Support
          [ ] Symmetric multi-processing support
          [ ] Support for extended (non-PC) x86 platforms
          [ ] Single-depth WCHAN output
          [ ] Paravirtualized guest support  --->
          [ ] Memtest
              Processor family (Generic-x86-64)  --->
              Preemption Model (No Forced Preemption (Server))  --->
          [ ] Reroute for broken boot IRQs (NEW)
          [ ] Machine Check / overheating reporting
          [ ] Dell laptop support
          [ ] /dev/cpu/microcode - microcode support
          [ ] /dev/cpu/*/msr - Model-specific register support
          [ ] /dev/cpu/*/cpuid - CPU information support
              Memory model (Sparse Memory)  --->
          [*] Sparse Memory virtual memmap (NEW)
          [ ] Allow for memory hot-add (NEW)
          [ ] Enable KSM for page merging
          (4096) Low address space to protect from user allocation
          [ ] Check for low memory corruption
          [ ] Reserve low 64K of RAM on AMI/Phoenix BIOSen
          -*- MTRR (Memory Type Range Register) support
          [ ]   MTRR cleanup support
          [ ] Enable seccomp to safely compute untrusted bytecode
          [ ] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
              Timer frequency (250 HZ)  --->
          [ ] kexec system call
  ------v(+)-----------------------------------------------------------------
  ---------------------------------------------------------------------------
                        <Select>    < Exit >    < Help >
```

Linux Kernel

Software Variability

- Configurable system

Firefox web browser

- Configuration options (aka software features)

feature flags (about:config)

- Variants

Firefox behavior (e.g., security)

- Large variability spaces

2000+ options (Boolean, categorical, numeric)

Software Variability

- Configurable system

Scikit

- Configuration options (aka software features)

Hyper-parameters

- Variants

Machine learning algorithm behavior

- Large variability spaces

Dozens of options (Boolean, categorical, numerical)

Software Variability

- Configurable system

x264 video encoder

- Configuration options (aka software features)

command line parameters

- Variants

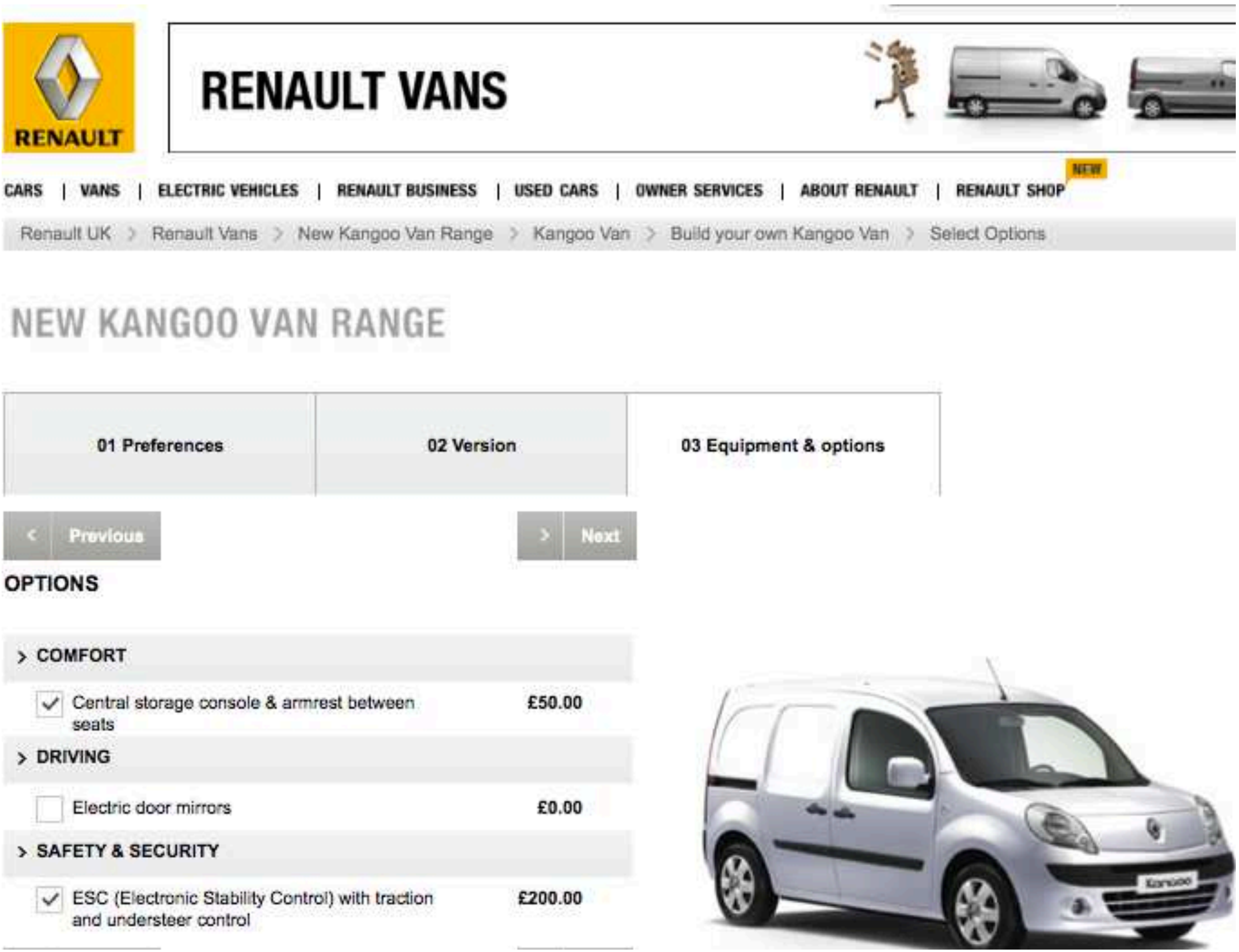
x264 behavior (different outputs, execution time, etc.)

- Large variability spaces

Dozens of options (Boolean, categorical, numeric)

```
x264 --no-progress
     --no-asm
     --rc-lookahead 60
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

40 seconds



RENAULT

RENAULT VANS

CARS | VANS | ELECTRIC VEHICLES | RENAULT BUSINESS | USED CARS | OWNER SERVICES | ABOUT RENAULT | RENAULT SHOP NEW

Renault UK > Renault Vans > New Kangoo Van Range > Kangoo Van > Build your own Kangoo Van > Select Options

NEW KANGOO VAN RANGE

01 Preferences | 02 Version | 03 Equipment & options

< Previous   > Next

OPTIONS

> COMFORT

☑ Central storage console & armrest between seats £50.00

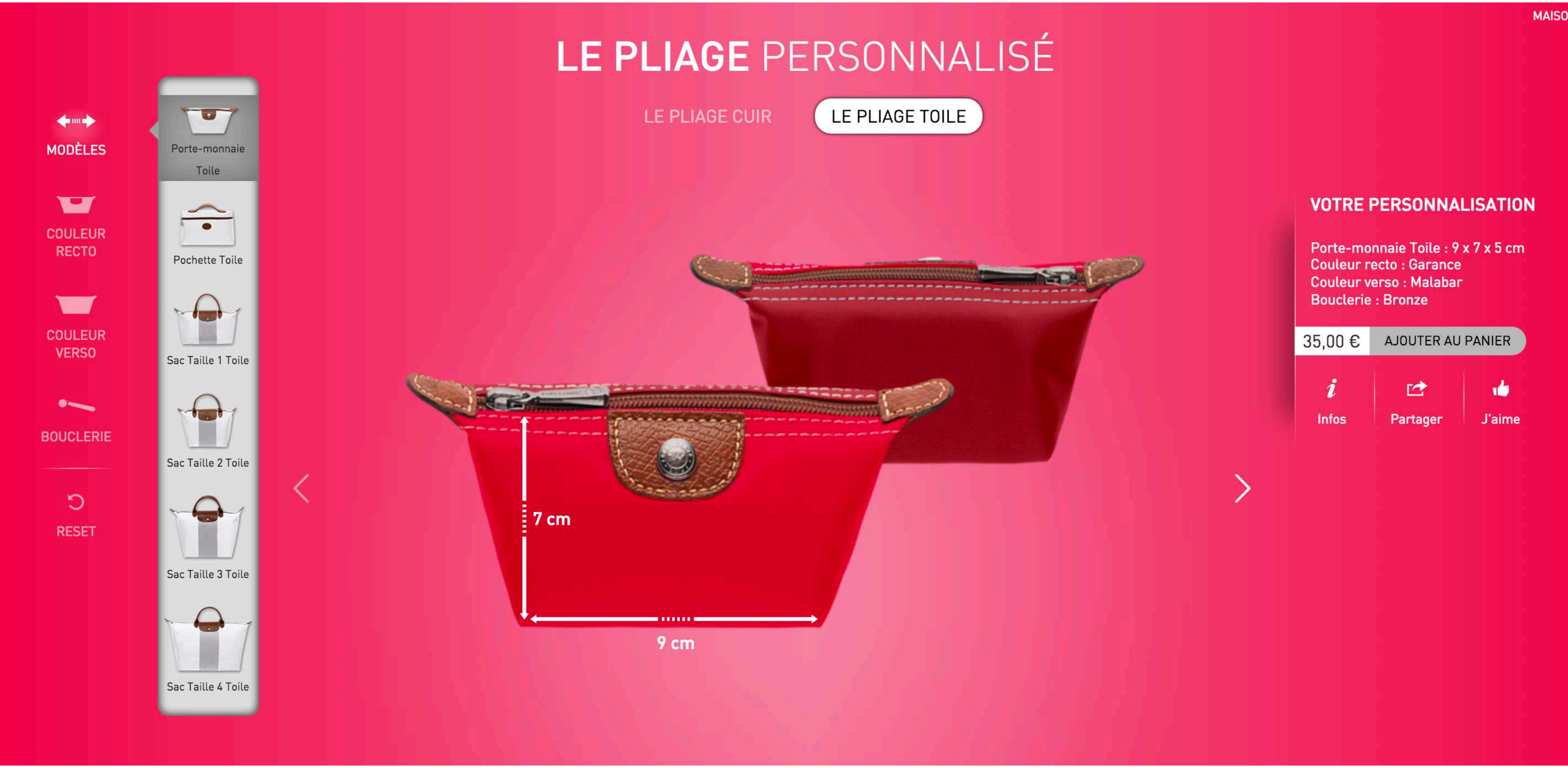
> DRIVING

□ Electric door mirrors £0.00

> SAFETY & SECURITY

☑ ESC (Electronic Stability Control) with traction and understeer control £200.00

“Reverse Engineering Web Configurators” Ebrahim Khalil Abbasi, Mathieu Acher, Patrick Heymans, and Anthony Cleve. In 17th European Conference on Software Maintenance and Reengineering (CSMR'14)

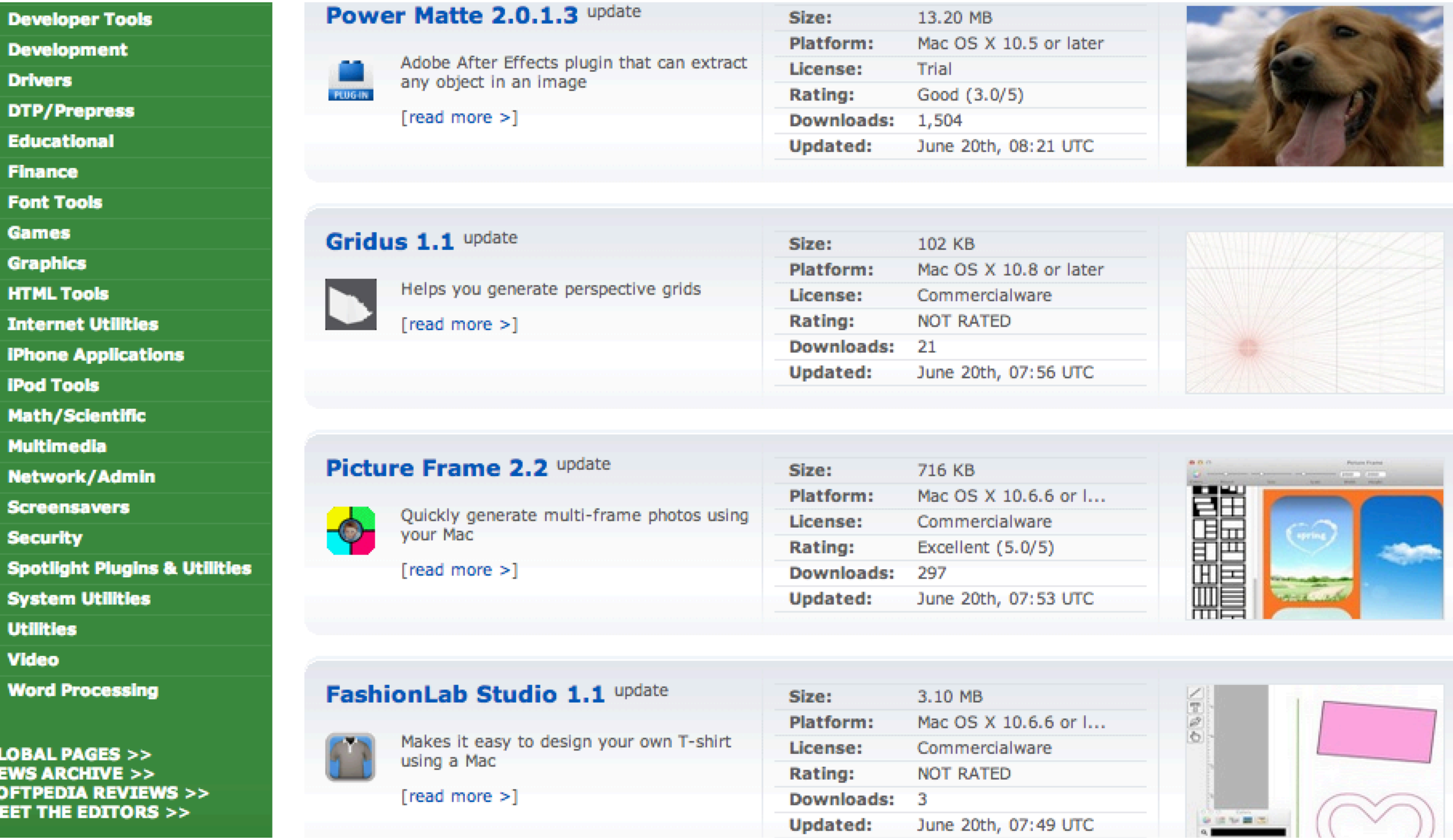


LE PLIAGE PERSONNALISÉ

VOTRE PERSONNALISATION

Porte-monnaie Toile : 9 x 7 x 5 cm
Couleur recto : Garance
Couleur verso : Malabar
Bouclerie : Bronze

35,00 €



- Developer Tools
- Development
- Drivers
- DTP/Prepress
- Educational
- Finance
- Font Tools
- Games
- Graphics
- HTML Tools
- Internet Utilities
- iPhone Applications
- iPod Tools
- Math/Scientific
- Multimedia
- Network/Admin
- Screensavers
- Security
- Spotlight Plugins & Utilities
- System Utilities
- Utilities
- Video
- Word Processing

LOBAL PAGES >>
EWS ARCHIVE >>
OFTPEDIA REVIEWS >>
EET THE EDITORS >>

Power Matte 2.0.1.3 update

Adobe After Effects plugin that can extract any object in an image

[read more >]

Size:	13.20 MB
Platform:	Mac OS X 10.5 or later
License:	Trial
Rating:	Good (3.0/5)
Downloads:	1,504
Updated:	June 20th, 08:21 UTC

Gridus 1.1 update

Helps you generate perspective grids

[read more >]

Size:	102 KB
Platform:	Mac OS X 10.8 or later
License:	Commercialware
Rating:	NOT RATED
Downloads:	21
Updated:	June 20th, 07:56 UTC

Picture Frame 2.2 update

Quickly generate multi-frame photos using your Mac

[read more >]

Size:	716 KB
Platform:	Mac OS X 10.6.6 or l...
License:	Commercialware
Rating:	Excellent (5.0/5)
Downloads:	297
Updated:	June 20th, 07:53 UTC

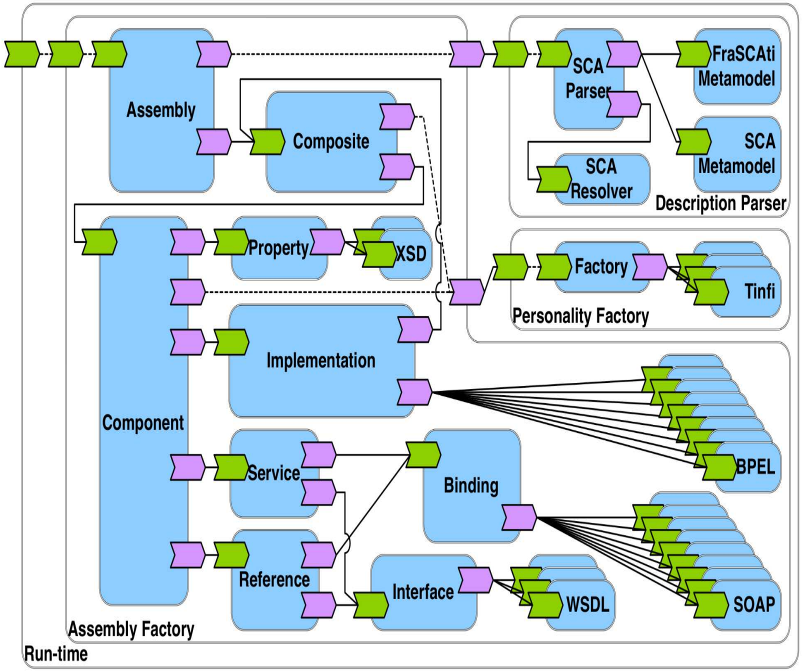
FashionLab Studio 1.1 update

Makes it easy to design your own T-shirt using a Mac

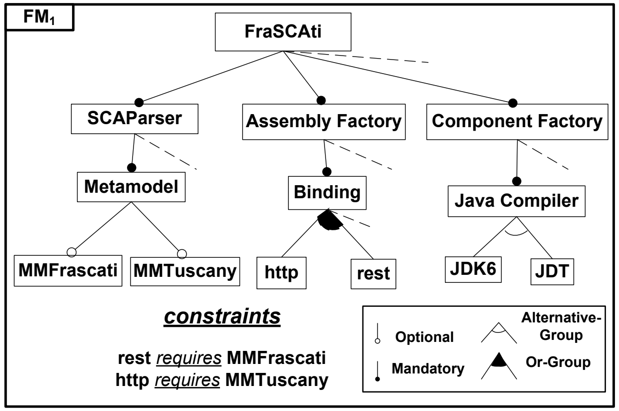
[read more >]

Size:	3.10 MB
Platform:	Mac OS X 10.6.6 or l...
License:	Commercialware
Rating:	NOT RATED
Downloads:	3
Updated:	June 20th, 07:49 UTC

« Feature Model Extraction from Large Collections of Informal Product Descriptions »
Jean-Marc Davril, Edouard Delfosse, Negar Hariri, Mathieu Acher, Jane Cleland-Huang, Patrick Heymans (ESEC/FSE'13)



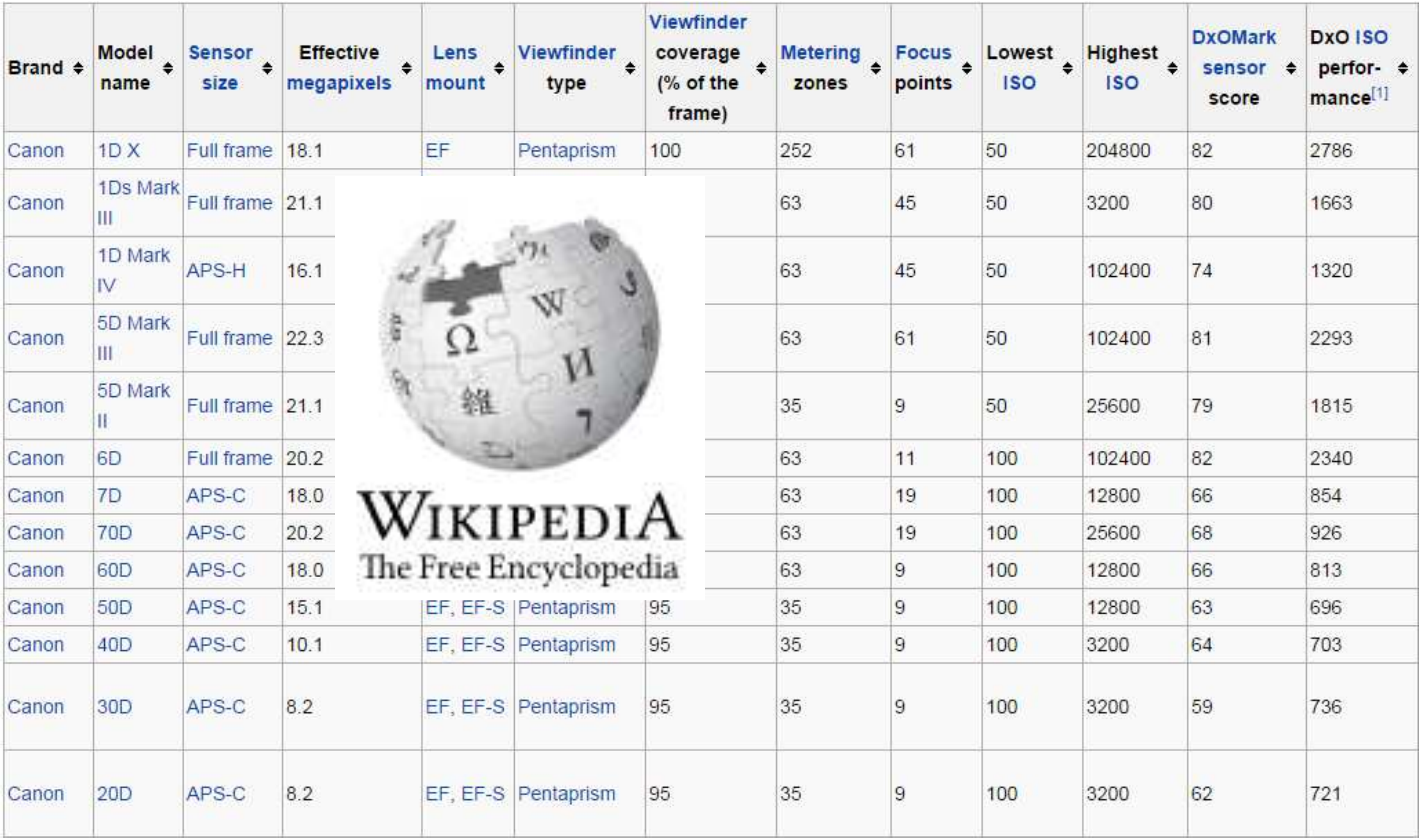
maven



Variability Model

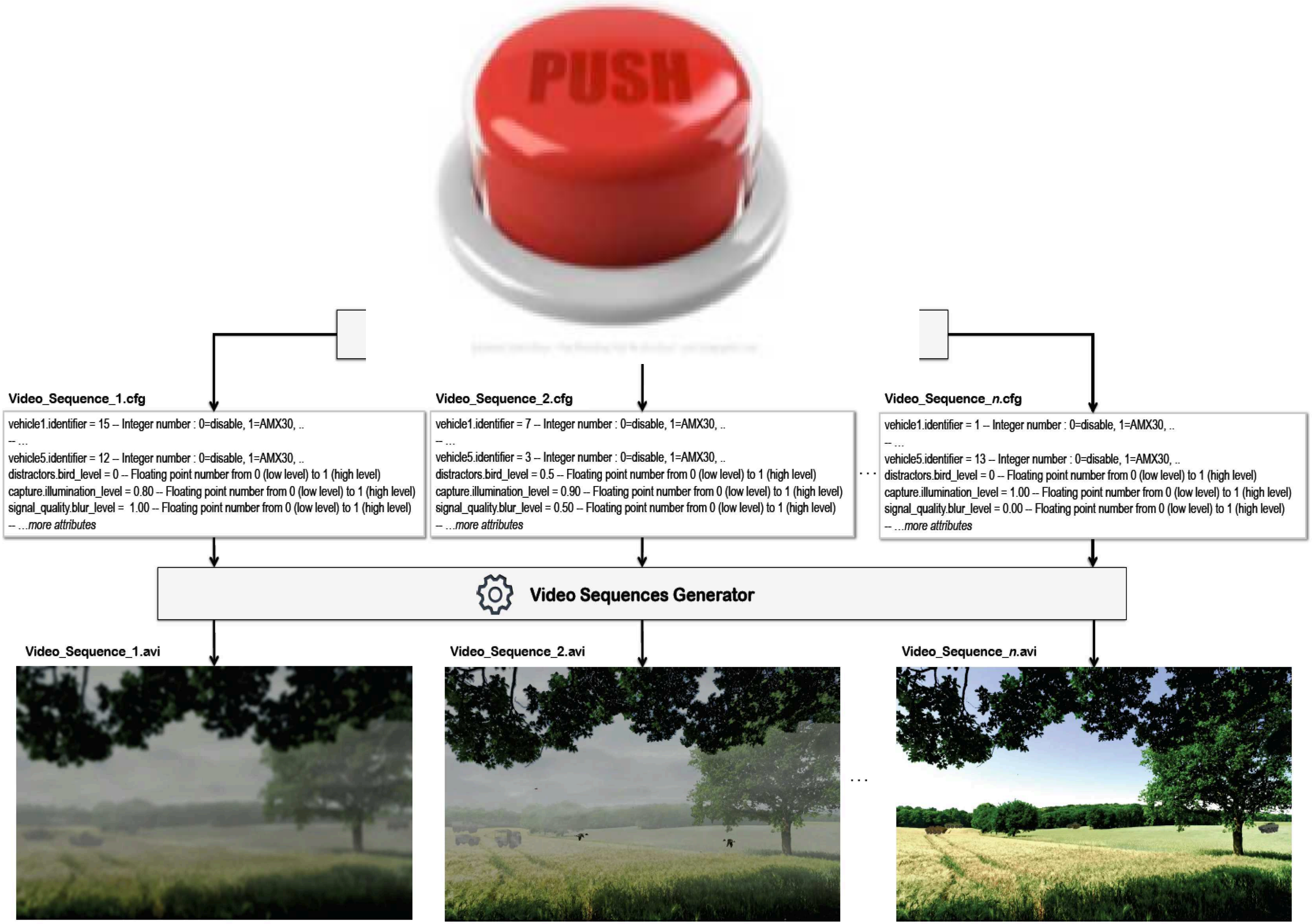
« Extraction and Evolution of Architectural Variability Models in Plugin-based Systems »
Mathieu Acher, Anthony Cleve, Philippe Collet, Philippe Merle, Laurence Duchien, Philippe Lahire ECSA/SoSyM'14

Printer
Firmware



Brand	Model name	Sensor size	Effective megapixels	Lens mount	Viewfinder type	Viewfinder coverage (% of the frame)	Metering zones	Focus points	Lowest ISO	Highest ISO	DxOMark sensor score	DxO ISO perfor-mance[1]
Canon	1D X	Full frame	18.1	EF	Pentaprism	100	252	61	50	204800	82	2786
Canon	1Ds Mark III	Full frame	21.1	[illegible]	[illegible]	[illegible]	63	45	50	3200	80	1663
Canon	1D Mark IV	APS-H	16.1	[illegible]	[illegible]	[illegible]	63	45	50	102400	74	1320
Canon	5D Mark III	Full frame	22.3	[illegible]	[illegible]	[illegible]	63	61	50	102400	81	2293
Canon	5D Mark II	Full frame	21.1	[illegible]	[illegible]	[illegible]	35	9	50	25600	79	1815
Canon	6D	Full frame	20.2	[illegible]	[illegible]	[illegible]	63	11	100	102400	82	2340
Canon	7D	APS-C	18.0	[illegible]	[illegible]	[illegible]	63	19	100	12800	66	854
Canon	70D	APS-C	20.2	[illegible]	[illegible]	[illegible]	63	19	100	25600	68	926
Canon	60D	APS-C	18.0	[illegible]	[illegible]	[illegible]	63	9	100	12800	66	813
Canon	50D	APS-C	15.1	EF, EF-S	Pentaprism	95	35	9	100	12800	63	696
Canon	40D	APS-C	10.1	EF, EF-S	Pentaprism	95	35	9	100	3200	64	703
Canon	30D	APS-C	8.2	EF, EF-S	Pentaprism	95	35	9	100	3200	59	736
Canon	20D	APS-C	8.2	EF, EF-S	Pentaprism	95	35	9	100	3200	62	721

Guillaume Bécan, Nicolas Sannier, Mathieu Acher, Olivier Barais, Arnaud Blouin, and Benoit Baudry. Automating the Formalization of Product Comparison Matrices (2014). In 29th IEEE/ACM International Conference on Automated Software Engineering (ASE'14)



Video_Sequence_1.cfg

```
vehicle1.identifier = 15 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 12 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.80 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level =  1.00 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

Video_Sequence_2.cfg

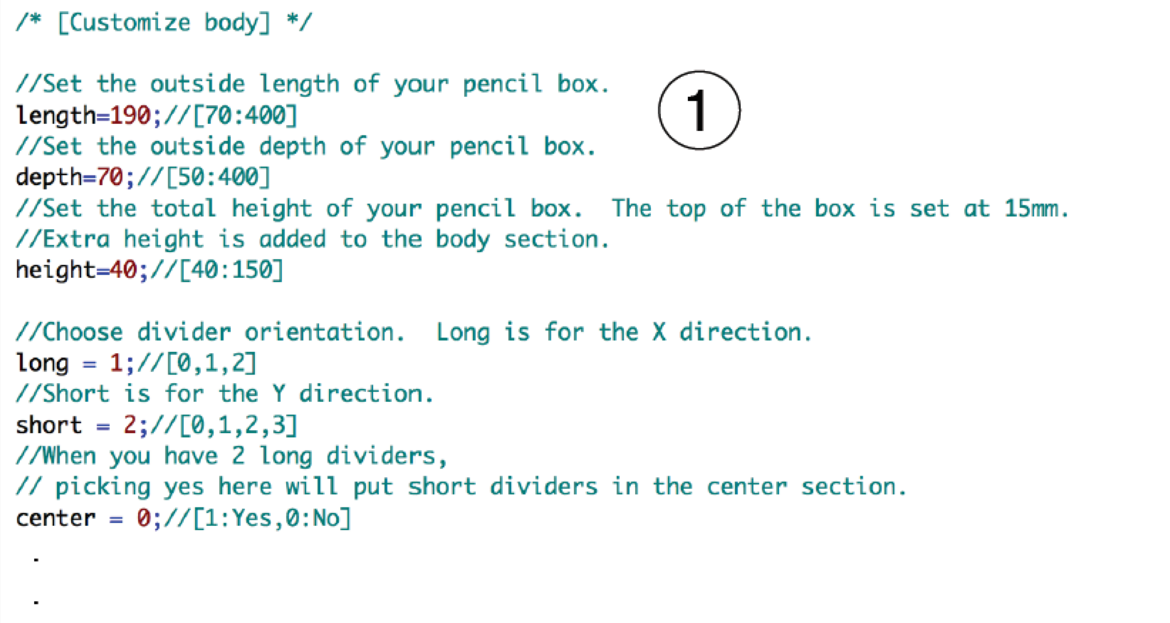
```
vehicle1.identifier = 7 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 3 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0.5 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.90 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.50 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

. . .

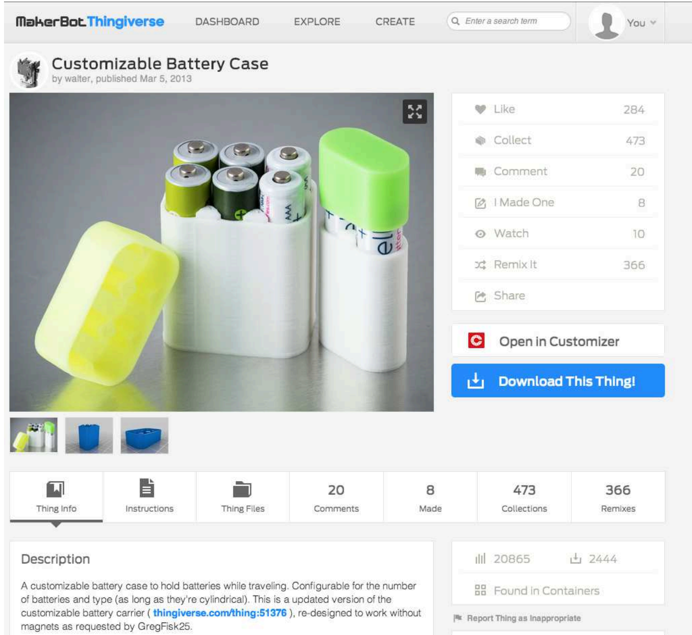
Video_Sequence_*n*.cfg

```
vehicle1.identifier = 1 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 13 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 1.00 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.00 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

José A. Galindo, Mauricio Alférez, Mathieu Acher, Benoit Baudry, David Benavides:
A variability-based testing approach for synthesizing video sequences. ISSTA 2014:



```
/* [Customize body] */

//Set the outside length of your pencil box.
length=190;//[70:400]
//Set the outside depth of your pencil box.
depth=70;//[50:400]
//Set the total height of your pencil box.  The top of the box is set at 15mm.
//Extra height is added to the body section.
height=40;//[40:150]

//Choose divider orientation.  Long is for the X direction.
long = 1;//[0,1,2]
//Short is for the Y direction.
short = 2;//[0,1,2,3]
//When you have 2 long dividers,
// picking yes here will put short dividers in the center section.
center = 0;//[1:Yes,0:No]
 .
 .
```



Customizable Battery Case

by walter, published Mar 5, 2013

Like	284
Collect	473
Comment	20
I Made One	8
Watch	10
Remix It	366
Share	

Thing Info	Instructions	Thing Files	Comments	Made	Collections	Remixes
			20	8	473	366

Description

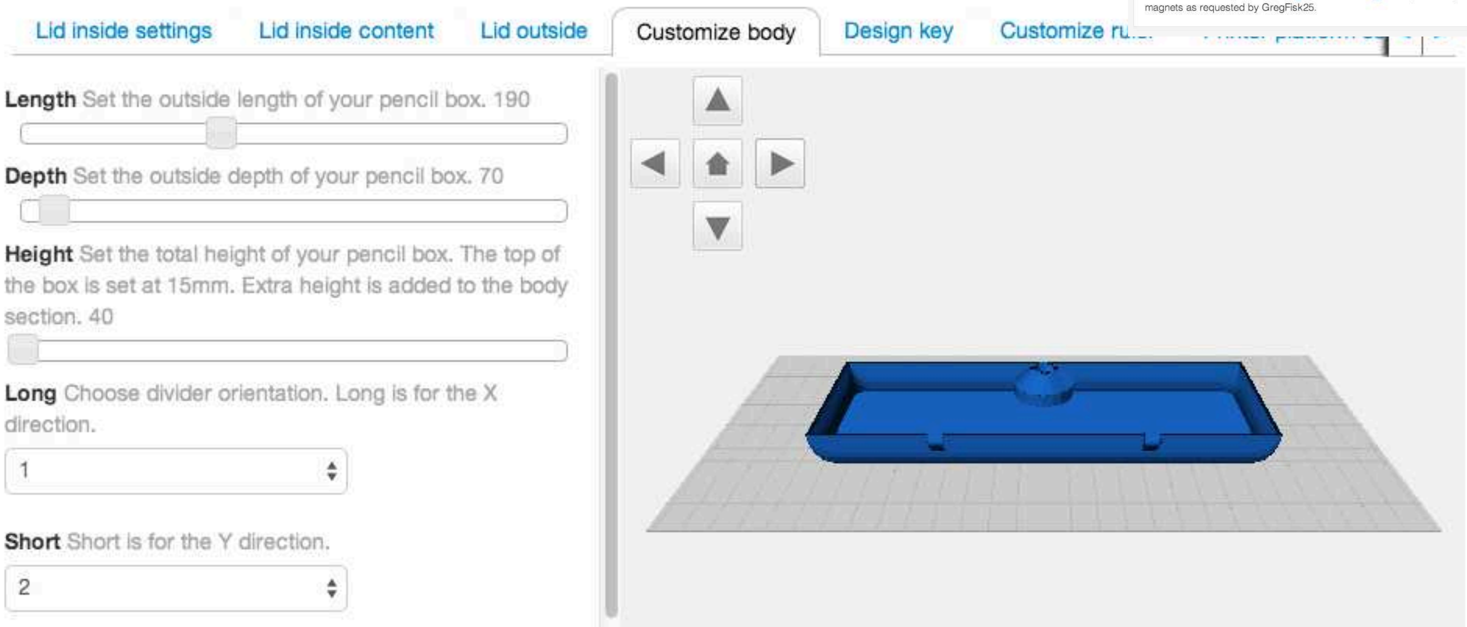
A customizable battery case to hold batteries while traveling. Configurable for the number of batteries and type (as long as they're cylindrical). This is a updated version of the customizable battery carrier (thingiverse.com/thing:51376), re-designed to work without magnets as requested by GregFisk25.

20865 2444

Found in Containers

Report Thing as Inappropriate

Makes view more



Length Set the outside length of your pencil box. 190

Depth Set the outside depth of your pencil box. 70

Height Set the total height of your pencil box. The top of the box is set at 15mm. Extra height is added to the body section. 40

Long Choose divider orientation. Long is for the X direction.

1

Short Short is for the Y direction.

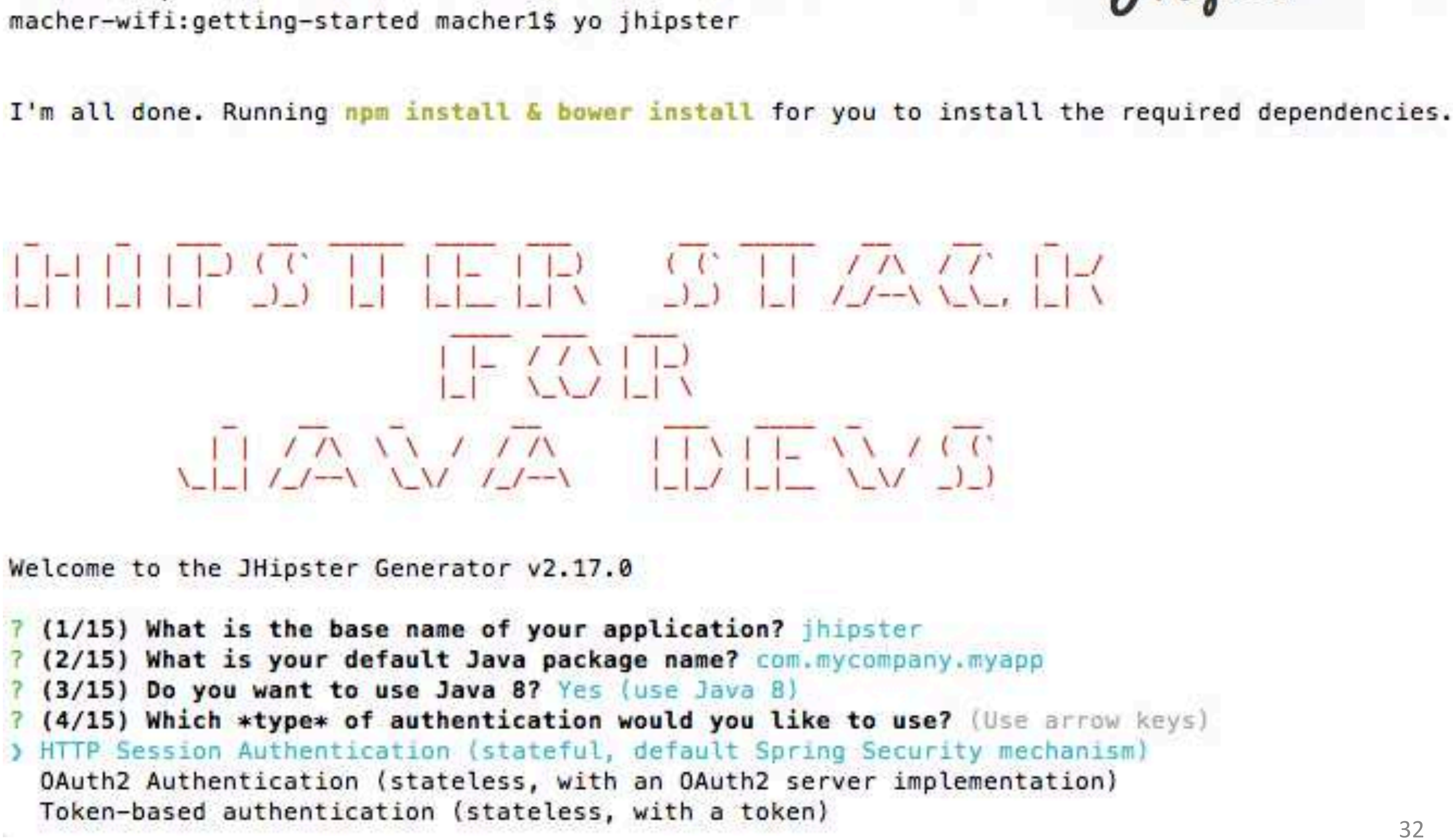
2

Mathieu Acher, Benoit Baudry, Olivier Barais, Jean-Marc Jézéquel:
Customization and 3D printing: a challenging playground for software product lines.
SPLC 2014: 142-146

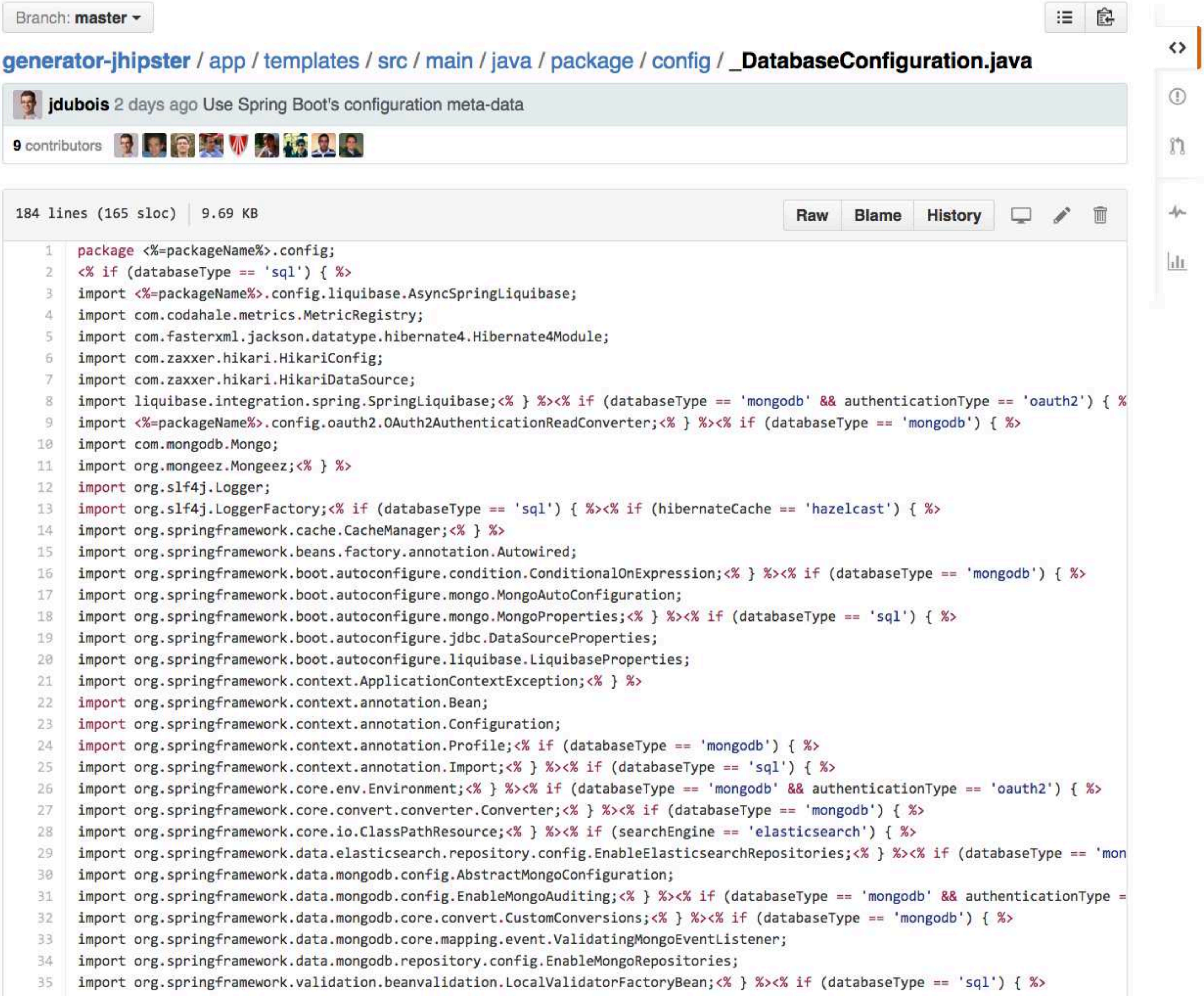
Case study: JHipster

- Web-apps generator
  - Spring-Boot
  - Bootstrap / AngularJS
  - 100 % Open Source
- Yeoman
  - Bower, npm
  - yo
- Used all over the world
  - Large companies (HBO, Google, Adobe)[1]
  - Independent developers
  - Our students
- GitHub
  - 6000+ stars
  - 118 releases (JHipster 3.6.1, 18 Aug 2016)
  - 300+ contributors

[1] https://jhipster.github.io/companies-using-jhipster/



```
macher-wifi:getting-started macher1$ yo jhipster


I'm all done. Running npm install & bower install for you to install the required dependencies.



 _     _   ___   ___   ___   _____   ___   ___      ___   _____   ___    ___   _   _
| |_| | | | | | |   | ( (   |_   _| | |_  |   |    ( (   |_   _| /   \  / __| | |_/
|_| |_| | |_| | |_|   _) )  |_|   |_|__ |_|  \     _) )  |_|   /__ \ \ \__,  |_| \
                          ___   ___   ___
                         | |_  / _ \ |   |
                         |_|   \___/ |_|  \
           _   ___   _   _   ___      ___   ___  _   _   ___
          | | /   \  \ \ / /  /   \    |   \ | |_  \ \ / / ( (
         \_|_| /__ \  \_/  /__ \   |_|_/ |_|__  \_/   _) )

Welcome to the JHipster Generator v2.17.0

? (1/15) What is the base name of your application? jhipster
? (2/15) What is your default Java package name? com.mycompany.myapp
? (3/15) Do you want to use Java 8? Yes (use Java 8)
? (4/15) Which *type* of authentication would you like to use? (Use arrow keys)
❯ HTTP Session Authentication (stateful, default Spring Security mechanism)
  OAuth2 Authentication (stateless, with an OAuth2 server implementation)
  Token-based authentication (stateless, with a token)
```



generator-jhipster / app / templates / src / main / java / package / config / _DatabaseConfiguration.java

jdubois 2 days ago Use Spring Boot's configuration meta-data

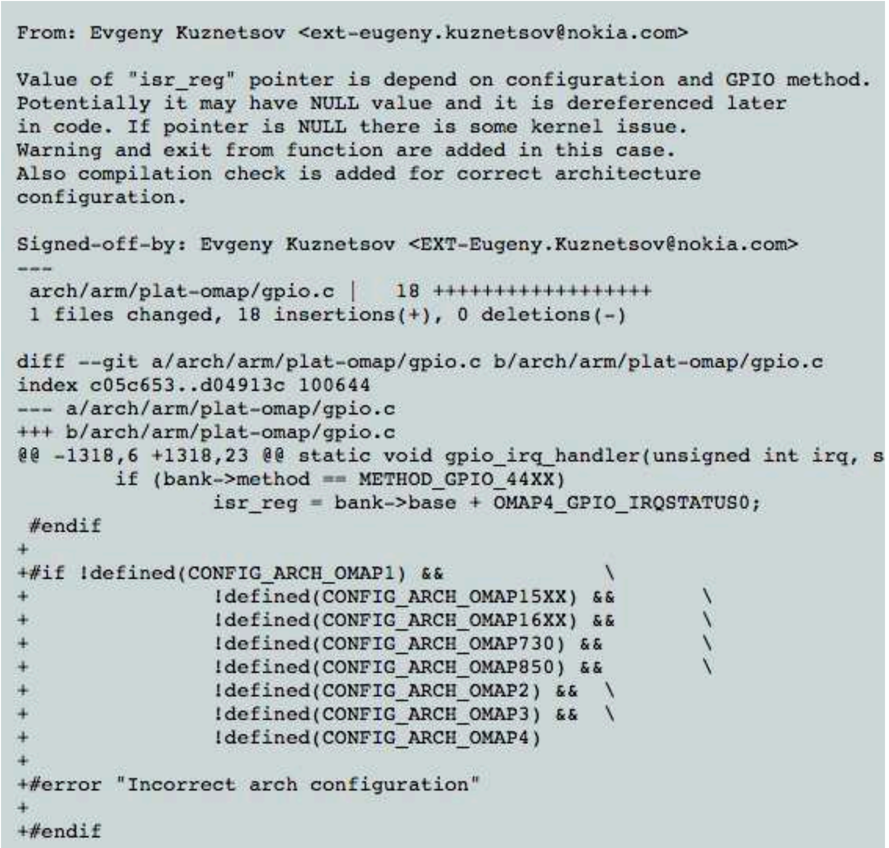
9 contributors

184 lines (165 sloc) | 9.69 KB

```
package <%=packageName%>.config;
<% if (databaseType == 'sql') { %>
import <%=packageName%>.config.liquibase.AsyncSpringLiquibase;
import com.codahale.metrics.MetricRegistry;
import com.fasterxml.jackson.datatype.hibernate4.Hibernate4Module;
import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;
import liquibase.integration.spring.SpringLiquibase;<% } %><% if (databaseType == 'mongodb' && authenticationType == 'oauth2') { %
import <%=packageName%>.config.oauth2.OAuth2AuthenticationReadConverter;<% } %><% if (databaseType == 'mongodb') { %>
import com.mongodb.Mongo;
import org.mongeez.Mongeez;<% } %>
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;<% if (databaseType == 'sql') { %><% if (hibernateCache == 'hazelcast') { %>
import org.springframework.cache.CacheManager;<% } %>
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.autoconfigure.condition.ConditionalOnExpression;<% } %><% if (databaseType == 'mongodb') { %>
import org.springframework.boot.autoconfigure.mongo.MongoAutoConfiguration;
import org.springframework.boot.autoconfigure.mongo.MongoProperties;<% } %><% if (databaseType == 'sql') { %>
import org.springframework.boot.autoconfigure.jdbc.DataSourceProperties;
import org.springframework.boot.autoconfigure.liquibase.LiquibaseProperties;
import org.springframework.context.ApplicationContextException;<% } %>
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Profile;<% if (databaseType == 'mongodb') { %>
import org.springframework.context.annotation.Import;<% } %><% if (databaseType == 'sql') { %>
import org.springframework.core.env.Environment;<% } %><% if (databaseType == 'mongodb' && authenticationType == 'oauth2') { %>
import org.springframework.core.convert.converter.Converter;<% } %><% if (databaseType == 'mongodb') { %>
import org.springframework.core.io.ClassPathResource;<% } %><% if (searchEngine == 'elasticsearch') { %>
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;<% } %><% if (databaseType == 'mon
import org.springframework.data.mongodb.config.AbstractMongoConfiguration;
import org.springframework.data.mongodb.config.EnableMongoAuditing;<% } %><% if (databaseType == 'mongodb' && authenticationType =
import org.springframework.data.mongodb.core.convert.CustomConversions;<% } %><% if (databaseType == 'mongodb') { %>
import org.springframework.data.mongodb.core.mapping.event.ValidatingMongoEventListener;
import org.springframework.data.mongodb.repository.config.EnableMongoRepositories;
import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;<% } %><% if (databaseType == 'sql') { %>
```

Software Variability: Problems

- **Very large variability spaces**
- Software developers: How to ensure that all software variants are “valid”?

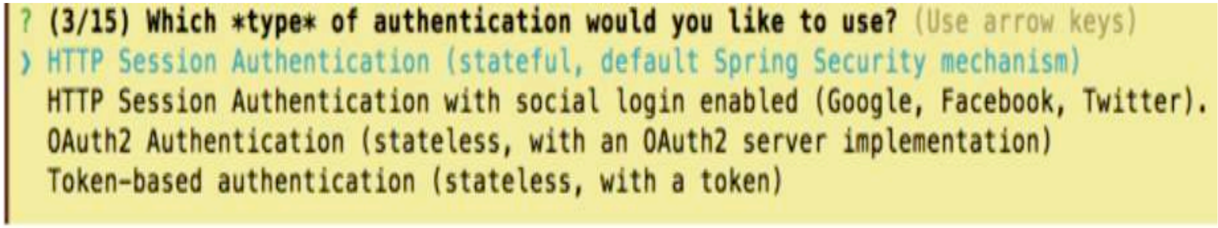


```
From: Evgeny Kuznetsov <ext-eugeny.kuznetsov@nokia.com>

Value of "isr_reg" pointer is depend on configuration and GPIO method.
Potentially it may have NULL value and it is dereferenced later
in code. If pointer is NULL there is some kernel issue.
Warning and exit from function are added in this case.
Also compilation check is added for correct architecture
configuration.

Signed-off-by: Evgeny Kuznetsov <EXT-Eugeny.Kuznetsov@nokia.com>
---
 arch/arm/plat-omap/gpio.c |   18 ++++++++++++++++++
 1 files changed, 18 insertions(+), 0 deletions(-)

diff --git a/arch/arm/plat-omap/gpio.c b/arch/arm/plat-omap/gpio.c
index c05c653..d04913c 100644
--- a/arch/arm/plat-omap/gpio.c
+++ b/arch/arm/plat-omap/gpio.c
@@ -1318,6 +1318,23 @@ static void gpio_irq_handler(unsigned int irq, s
        if (bank->method == METHOD_GPIO_44XX)
                isr_reg = bank->base + OMAP4_GPIO_IRQSTATUS0;
 #endif
+
+#if !defined(CONFIG_ARCH_OMAP1) &&                      \
+               !defined(CONFIG_ARCH_OMAP15XX) &&        \
+               !defined(CONFIG_ARCH_OMAP16XX) &&        \
+               !defined(CONFIG_ARCH_OMAP730) &&         \
+               !defined(CONFIG_ARCH_OMAP850) &&         \
+               !defined(CONFIG_ARCH_OMAP2) &&   \
+               !defined(CONFIG_ARCH_OMAP3) &&   \
+               !defined(CONFIG_ARCH_OMAP4)
+
+#error "Incorrect arch configuration"
+
+#endif
```

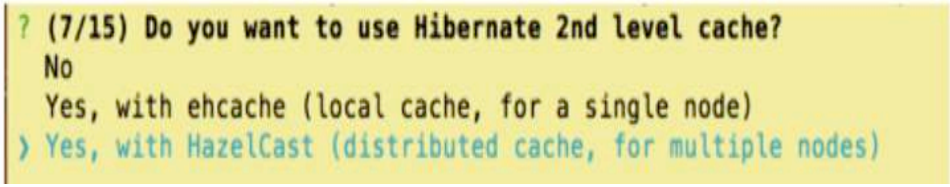
Software Variability

Software is working (sometimes)

- ○ yes but perhaps for one specific configuration (the default one)
- ○ is it working for **all configurations**?



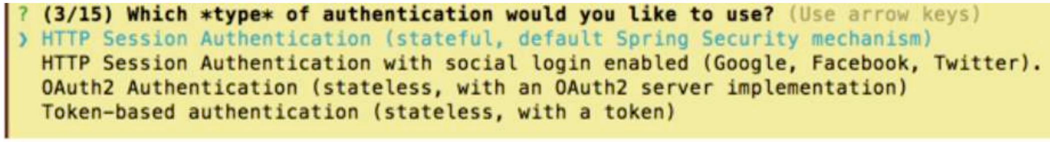
```
? (3/15) Which *type* of authentication would you like to use? (Use arrow keys)
❯ HTTP Session Authentication (stateful, default Spring Security mechanism)
  HTTP Session Authentication with social login enabled (Google, Facebook, Twitter).
  OAuth2 Authentication (stateless, with an OAuth2 server implementation)
  Token-based authentication (stateless, with a token)
```



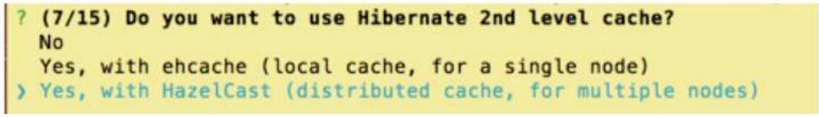
```
? (7/15) Do you want to use Hibernate 2nd level cache?
  No
  Yes, with ehcache (local cache, for a single node)
❯ Yes, with HazelCast (distributed cache, for multiple nodes)
```

At each modification/commit/push/release, do you test all configurations?

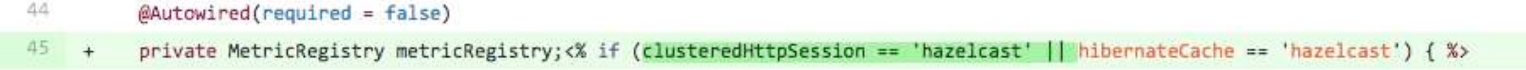
- No and you certainly have very good reasons
  - needs lots of resources (machines!); don’t want to burn the planet
  - needs an engineering effort to instrument testing of all configurations
  - the number of configurations is too important (eg 2^16000 for Linux)



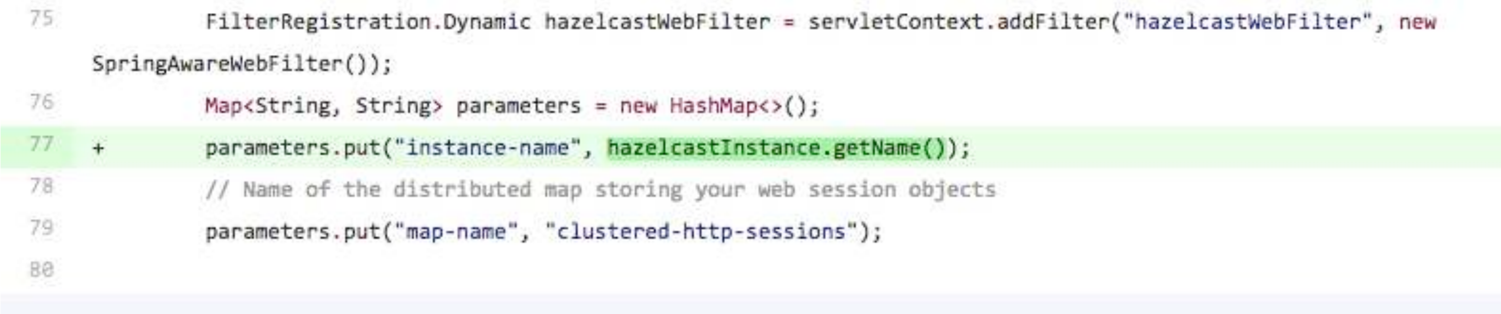
```
? (3/15) Which *type* of authentication would you like to use? (Use arrow keys)
❯ HTTP Session Authentication (stateful, default Spring Security mechanism)
  HTTP Session Authentication with social login enabled (Google, Facebook, Twitter).
  OAuth2 Authentication (stateless, with an OAuth2 server implementation)
  Token-based authentication (stateless, with a token)
```



```
? (7/15) Do you want to use Hibernate 2nd level cache?
  No
  Yes, with ehcache (local cache, for a single node)
❯ Yes, with HazelCast (distributed cache, for multiple nodes)
```



```
       @Autowired(required = false)
   +   private MetricRegistry metricRegistry;<% if (clusteredHttpSession == 'hazelcast' || hibernateCache == 'hazelcast') { %>
```



```
            FilterRegistration.Dynamic hazelcastWebFilter = servletContext.addFilter("hazelcastWebFilter", new SpringAwareWebFilter());
            Map<String, String> parameters = new HashMap<>();
   +        parameters.put("instance-name", hazelcastInstance.getName());
            // Name of the distributed map storing your web session objects
            parameters.put("map-name", "clustered-http-sessions");
```

At each modification/commit/push/release, do you test all configurations?

- No since too much resources and effort (impossible and unpractical)
- **"Sampling"** techniques (subset of configurations)
  - Apel et al. ICSE'16, Kaestner et al. ICSE'14 and ASE'16, Ana B. Sánchez et al. SoSyM 2017, Perrouin et al. ICST'10, Cohen et al. TSE'06, Henard et al. TSE'14, etc.
    - Many papers at SPLC, FSE, ASE, ICSE, ESE, TSE on this topic

What is the cost-effective sampling strategy to test configurations of a system?

Is it Worth testing All Configurations?

Testing with the community

ALL

Sampling

- We have tested all configurations of an industrial-strength, open-source generator (Jhipster)
- 26K+ configurations, 4376 hours/machine, 8 man/month
- “Ground truth” allows us to *precisely* assess sampling

36% failures explained by 6 feature interactions (faults)

- What is the most cost-effective sampling strategy?
  - T-wise or dissimilarity are very effective
  - **with “only” 126 configurations you can detect all 6 most important faults**

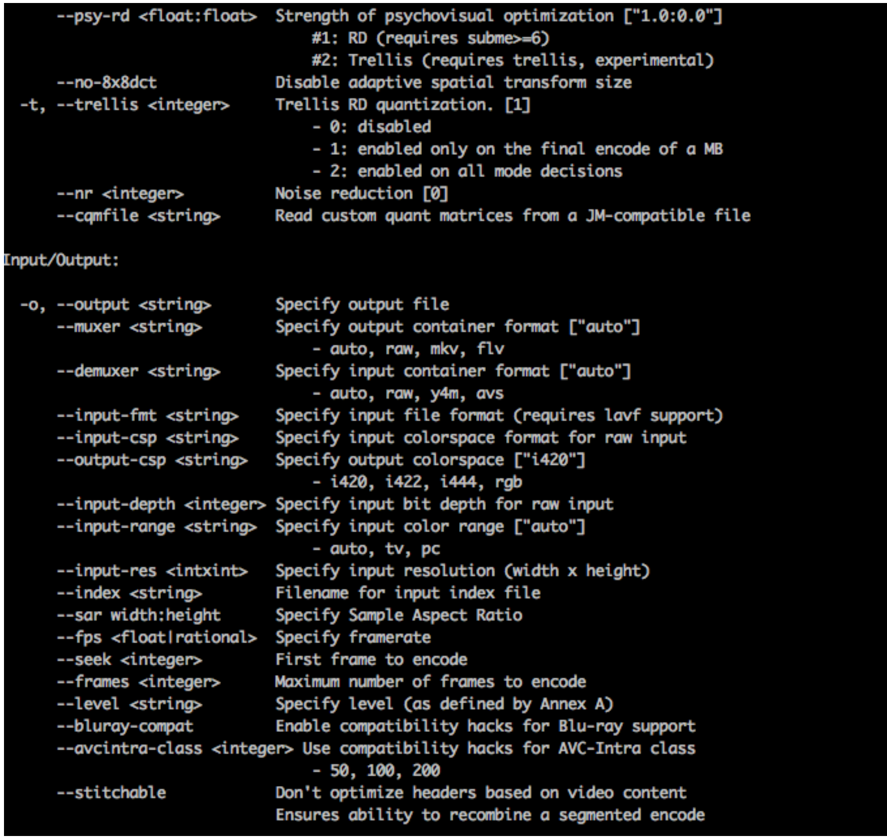
Axel Halin, Alexandre Nuttinck, Mathieu Acher, Xavier Devroey, Gilles Perrouin, Benoit Baudry.
Test them all, is it worth it? Assessing configuration sampling on the JHipster Web development stack
(2018). In Empirical Software Engineering journal

Software Variability: Problems

- **Very large variability spaces**
- Software users: How to choose the configuration that fits my requirements?

x264 --longhelp | wc -l

176



```
      --psy-rd <float:float>  Strength of psychovisual optimization ["1.0:0.0"]
                                  #1: RD (requires subme>=6)
                                  #2: Trellis (requires trellis, experimental)
      --no-8x8dct             Disable adaptive spatial transform size
  -t, --trellis <integer>     Trellis RD quantization. [1]
                                  - 0: disabled
                                  - 1: enabled only on the final encode of a MB
                                  - 2: enabled on all mode decisions
      --nr <integer>          Noise reduction [0]
      --cqmfile <string>      Read custom quant matrices from a JM-compatible file

Input/Output:

  -o, --output <string>       Specify output file
      --muxer <string>        Specify output container format ["auto"]
                                  - auto, raw, mkv, flv
      --demuxer <string>      Specify input container format ["auto"]
                                  - auto, raw, y4m, avs
      --input-fmt <string>    Specify input file format (requires lavf support)
      --input-csp <string>    Specify input colorspace format for raw input
      --output-csp <string>   Specify output colorspace ["i420"]
                                  - i420, i422, i444, rgb
      --input-depth <integer> Specify input bit depth for raw input
      --input-range <string>  Specify input color range ["auto"]
                                  - auto, tv, pc
      --input-res <intxint>   Specify input resolution (width x height)
      --index <string>        Filename for input index file
      --sar width:height      Specify Sample Aspect Ratio
      --fps <float|rational>  Specify framerate
      --seek <integer>        First frame to encode
      --frames <integer>      Maximum number of frames to encode
      --level <string>        Specify level (as defined by Annex A)
      --bluray-compat         Enable compatibility hacks for Blu-ray support
      --avcintra-class <integer> Use compatibility hacks for AVC-Intra class
                                  - 50, 100, 200
      --stitchable            Don't optimize headers based on video content
                              Ensures ability to recombine a segmented encode
```

Software Variability and Artificial Intelligence

- **Very large variability spaces**
- **AI#1** Abstraction/languages to formally and efficiently reason about configuration spaces
  - with SAT/CSP/SMT solvers
  - Eg constrained sampling
- **AI#2** Statistical machine learning to (out of a sample):
  - Understand the configuration space
  - Find the best configuration
  - Specialize the configuration space (e.g., by capturing constraints)
  - In a cost-effective way
- Humans (developers, end-users, integrator, scientists, etc.) and machines

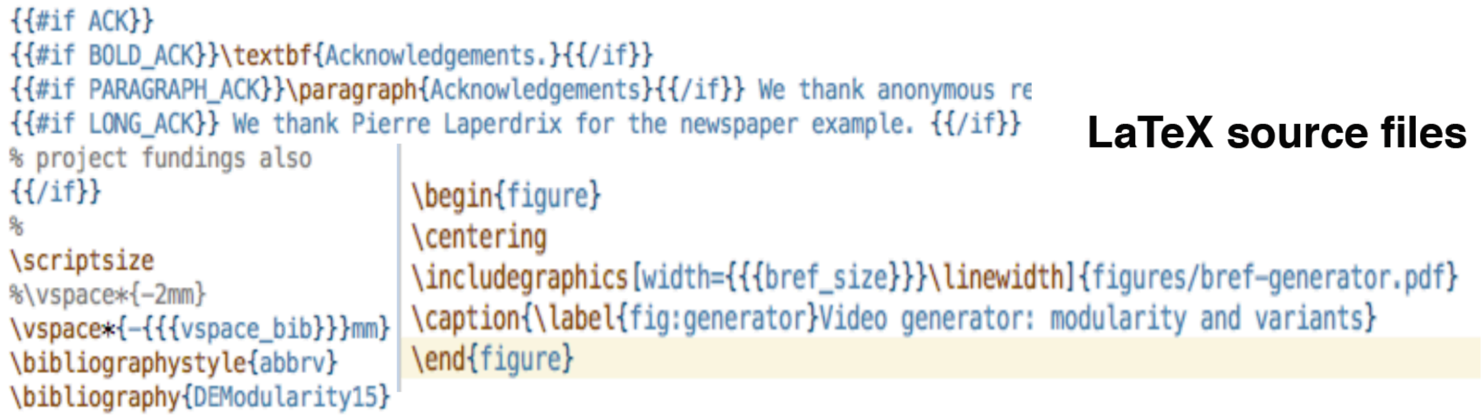
(end of the first part)

Modeling Variability

- Very large variability spaces
- AI#1 **Abstraction/languages to formally and efficiently reason about configuration spaces**
  - with SAT/CSP/SMT solvers
  - Eg constrained sampling
- Variability Models
  - Elaborated by humans
  - Reverse engineered from existing artefacts/systems
  - Promise: sound and complete representation of the configuration space

AI#1 Logic, satisfiability, constraints, reasoning, solving

① **Variability annotations and modeling**



```
{{#if ACK}}
{{#if BOLD_ACK}}\textbf{Acknowledgements.}{{/if}}
{{#if PARAGRAPH_ACK}}\paragraph{Acknowledgements}{{/if}} We thank anonymous re
{{#if LONG_ACK}} We thank Pierre Laperdrix for the newspaper example. {{/if}}
% project fundings also
{{/if}}
%
\scriptsize
%\vspace*{-2mm}
\vspace*{-{{{vspace_bib}}}mm}
\bibliographystyle{abbrv}
\bibliography{DEModularity15}
```

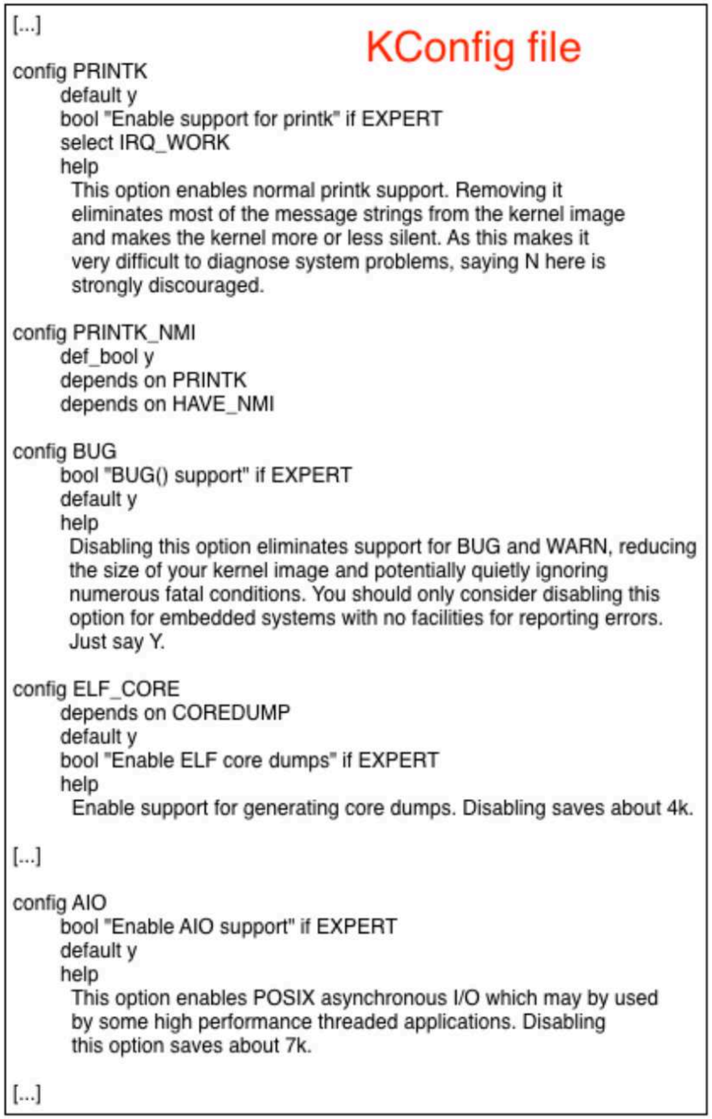
```
\begin{figure}
\centering
\includegraphics[width={{{bref_size}}}\linewidth]{figures/bref-generator.pdf}
\caption{\label{fig:generator}Video generator: modularity and variants}
\end{figure}
```

LaTeX source files

```
// Boolean options (features)
fmLaTeX = FM (VARY_LATEX : BREF BIB [PL_FOOTNOTE] [ACK] JS_STYLE
[LONG_AFFILIATION] ;
JS_STYLE : (JS_SCRIPTSIZE I JS_TINY I JS_FOOTNOTESIZE); // mutually exclusive
ACK : [LONG_ACK] (BOLD_ACK I PARAGRAPH_ACK); // LONG_ACK is optional
LONG_AFFILIATION : [EMAIL]; )
// numerical options (attributes)
real BIB.vspace_bib: [1.0..5.0] precision 1 // 1 decimal digit precision
real BREF.bref_size: [0.7..1.0] precision 1 // either 0.7 0.8 0.9 or 1.0
real cserver_size: [0.6..0.9] precision 1 // either 0.6 0.7 0.8 or 0.9
// specific constraints can be added a priori if needs be
…
```

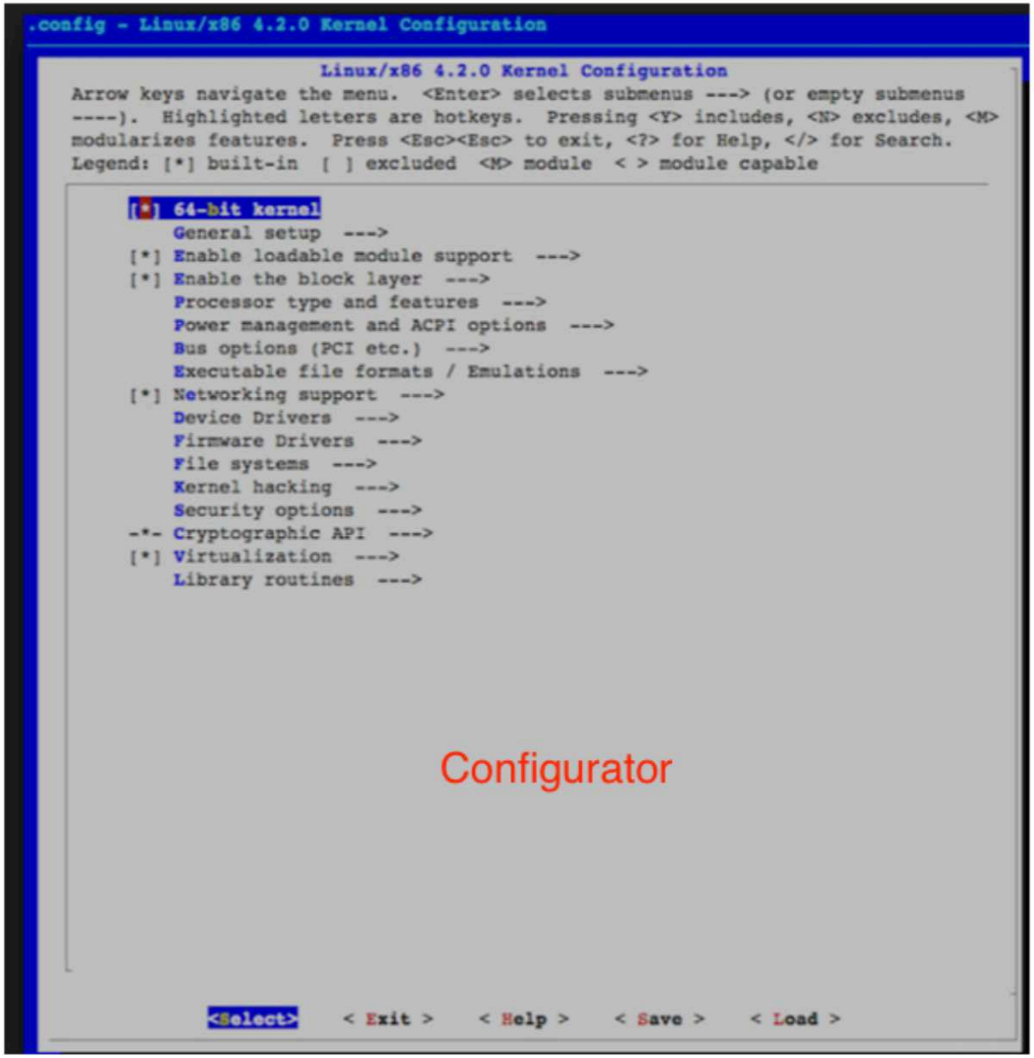
variability model

Linux



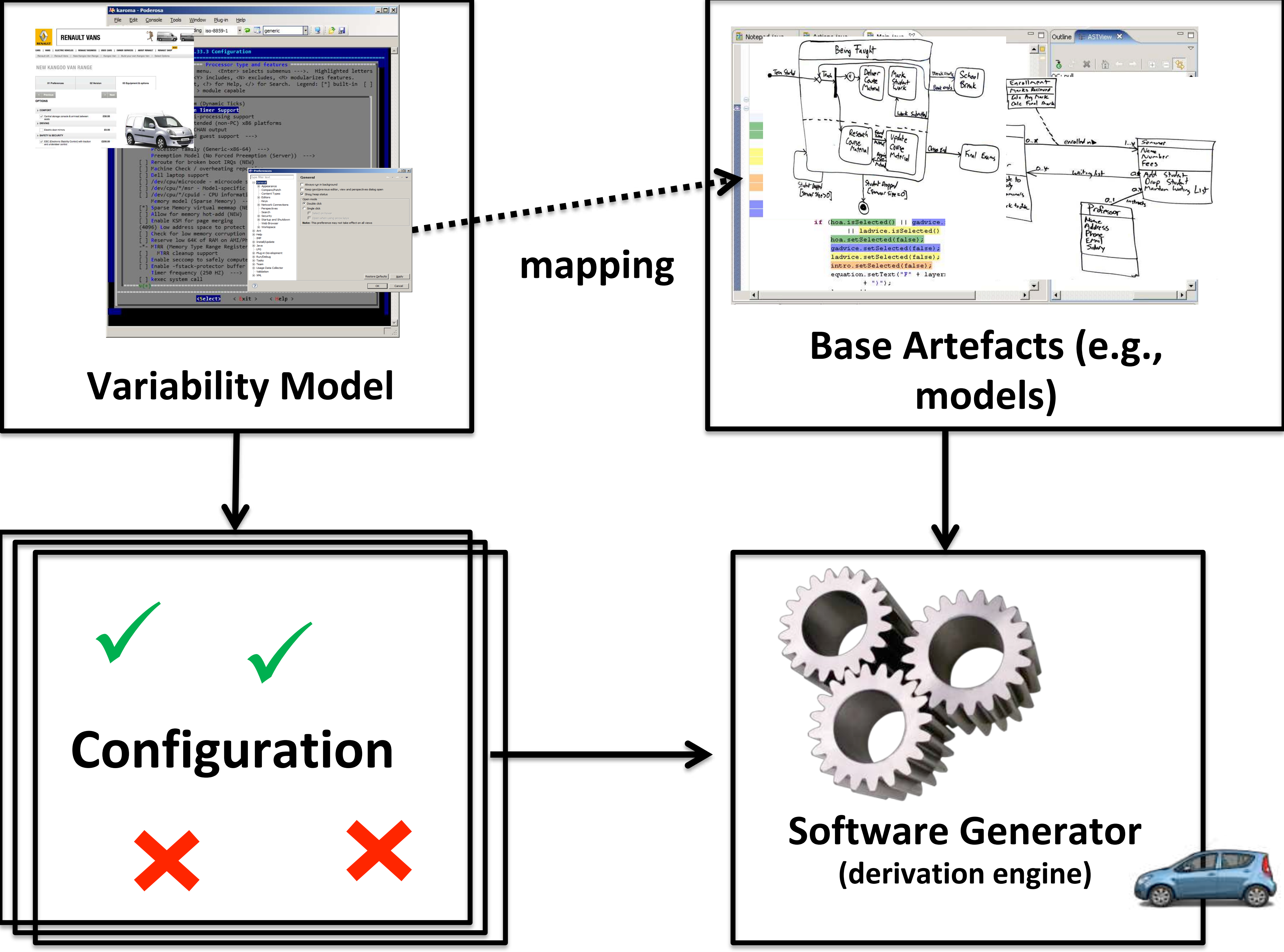
KConfig file

```
[...]

config PRINTK
        default y
        bool "Enable support for printk" if EXPERT
        select IRQ_WORK
        help
          This option enables normal printk support. Removing it
          eliminates most of the message strings from the kernel image
          and makes the kernel more or less silent. As this makes it
          very difficult to diagnose system problems, saying N here is
          strongly discouraged.

config PRINTK_NMI
        def_bool y
        depends on PRINTK
        depends on HAVE_NMI

config BUG
        bool "BUG() support" if EXPERT
        default y
        help
          Disabling this option eliminates support for BUG and WARN, reducing
          the size of your kernel image and potentially quietly ignoring
          numerous fatal conditions. You should only consider disabling this
          option for embedded systems with no facilities for reporting errors.
          Just say Y.

config ELF_CORE
        depends on COREDUMP
        default y
        bool "Enable ELF core dumps" if EXPERT
        help
          Enable support for generating core dumps. Disabling saves about 4k.

[...]

config AIO
        bool "Enable AIO support" if EXPERT
        default y
        help
          This option enables POSIX asynchronous I/O which may by used
          by some high performance threaded applications. Disabling
          this option saves about 7k.

[...]
```



```
.config - Linux/x86 4.2.0 Kernel Configuration
                      Linux/x86 4.2.0 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

          [*] 64-bit kernel
              General setup  --->
          [*] Enable loadable module support  --->
          [*] Enable the block layer  --->
              Processor type and features  --->
              Power management and ACPI options  --->
              Bus options (PCI etc.)  --->
              Executable file formats / Emulations  --->
          [*] Networking support  --->
              Device Drivers  --->
              Firmware Drivers  --->
              File systems  --->
              Kernel hacking  --->
              Security options  --->
          -*- Cryptographic API  --->
          [*] Virtualization  --->
              Library routines  --->

                  <Select>    < Exit >    < Help >    < Save >    < Load >
```

Configurator



Variability Model

mapping

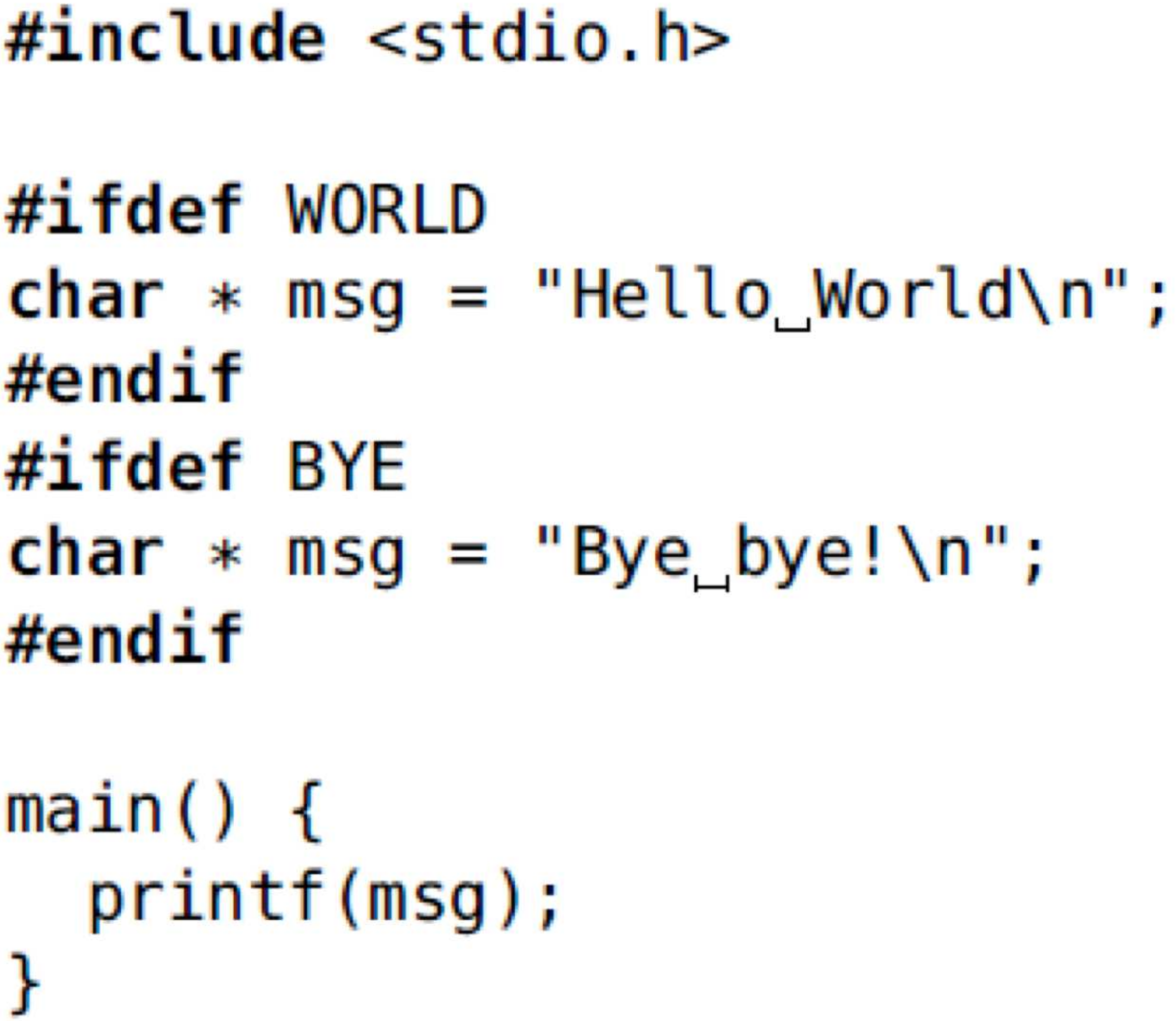
Base Artefacts (e.g., models)

Configuration

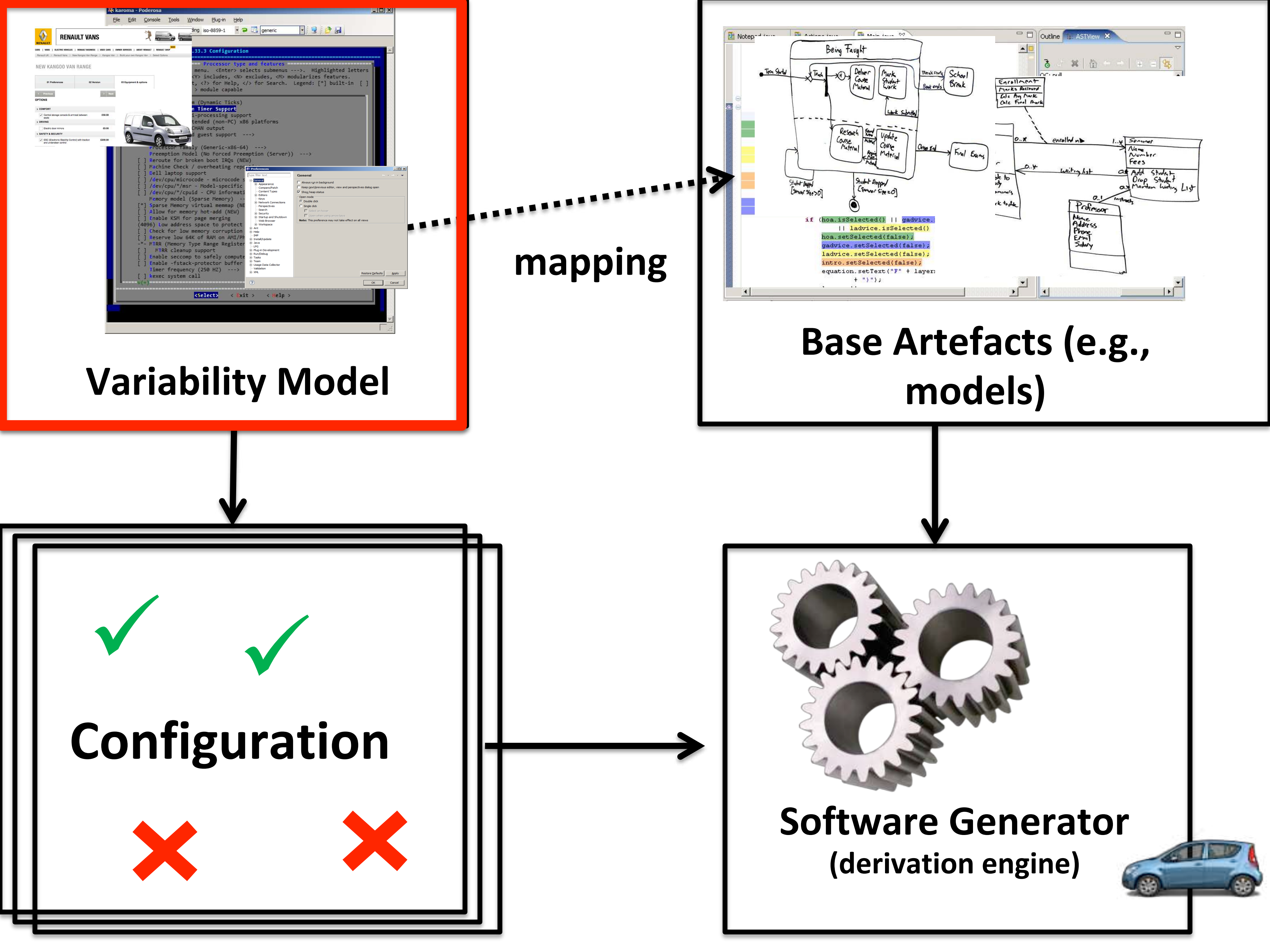
Software Generator

(derivation engine)

Simple question:
what are the constraints over
WORLD and BYE?



```
#include <stdio.h>

#ifdef WORLD
char * msg = "Hello␣World\n";
#endif
#ifdef BYE
char * msg = "Bye␣bye!\n";
#endif

main() {
  printf(msg);
}
```



Variability Model

mapping

Base Artefacts (e.g., models)

✓ ✓

Configuration

✗ ✗

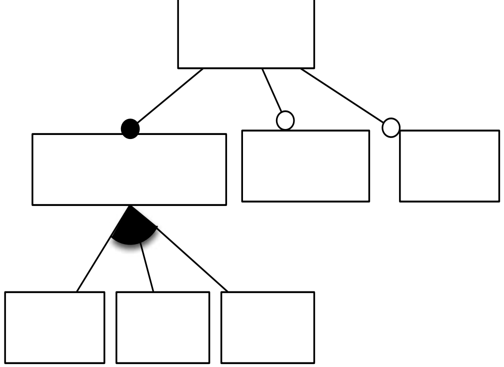
Software Generator

(derivation engine)

Unused flexibility

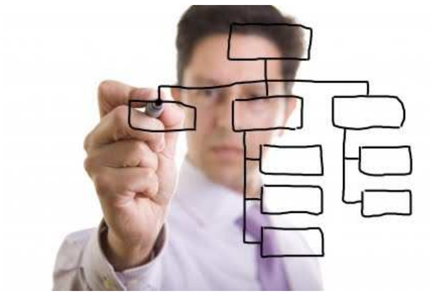
Illegal variant

Feature Model



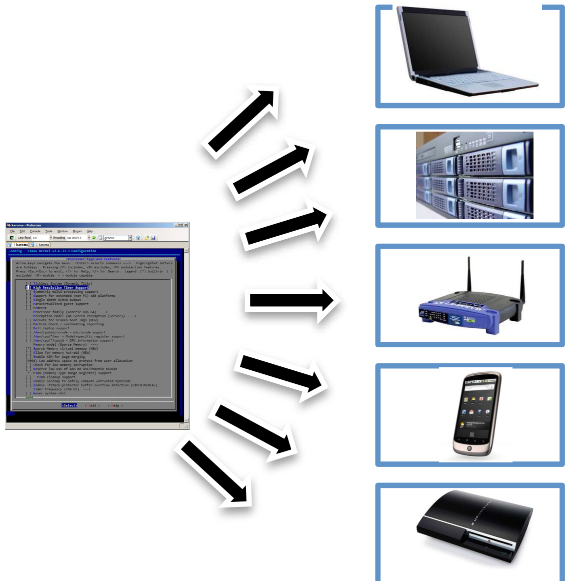
not, and, or, implies

Communicative



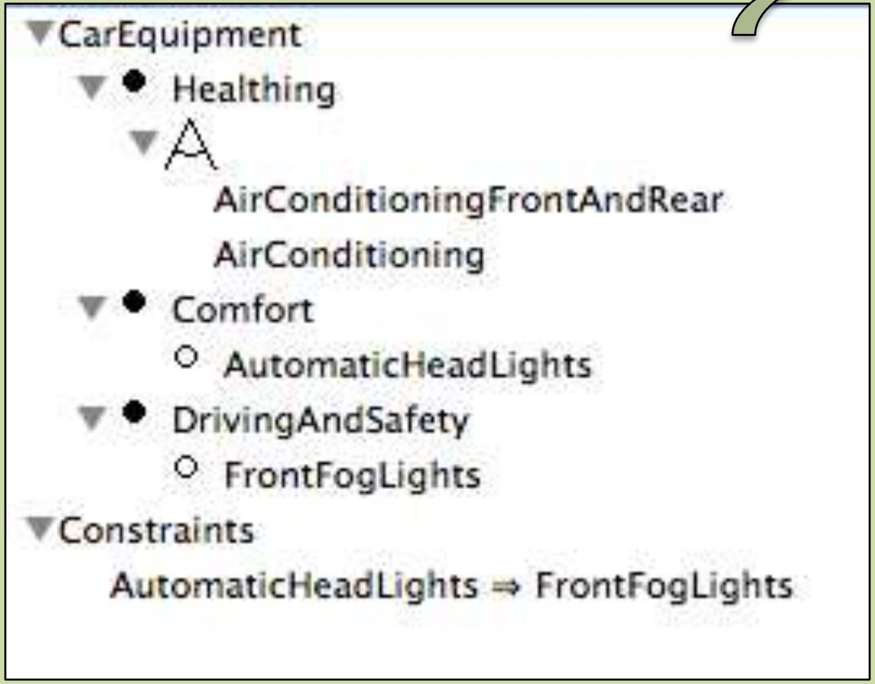
Analytic

Generative

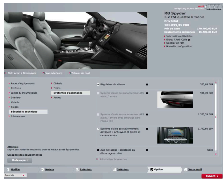


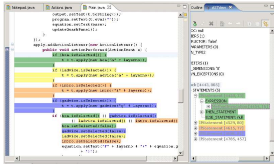
Feature Models

(defacto standard for modeling variability)



- ▼ CarEquipment
  - ▼ ● Healthing
    - ▼ A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ▼ ● Comfort
    - ○ AutomaticHeadLights
  - ▼ ● DrivingAndSafety
    - ○ FrontFogLights
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

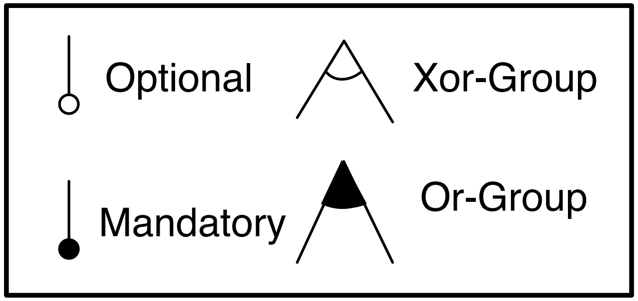


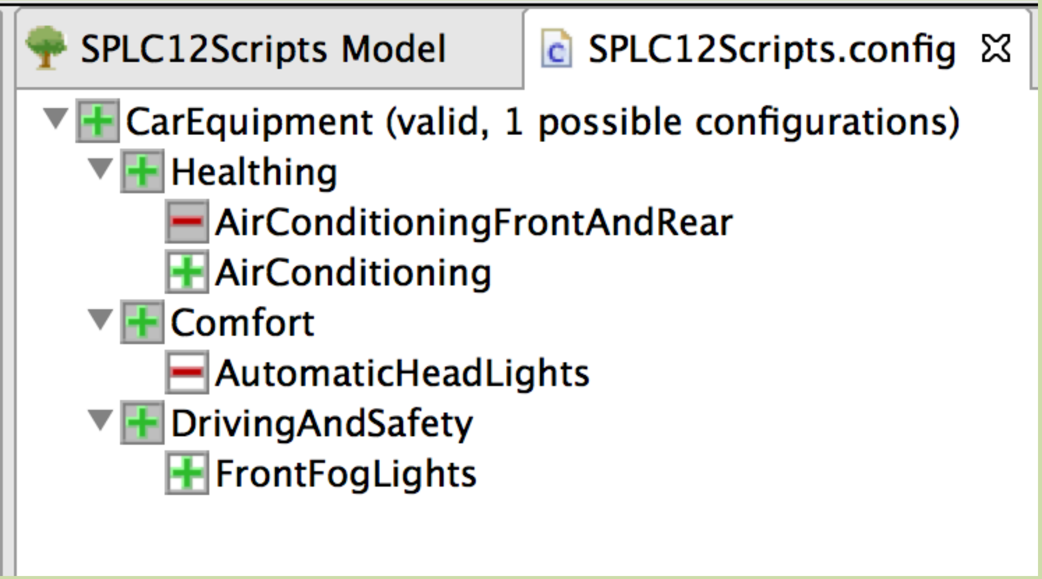


Hierarchy: rooted tree

Variability:

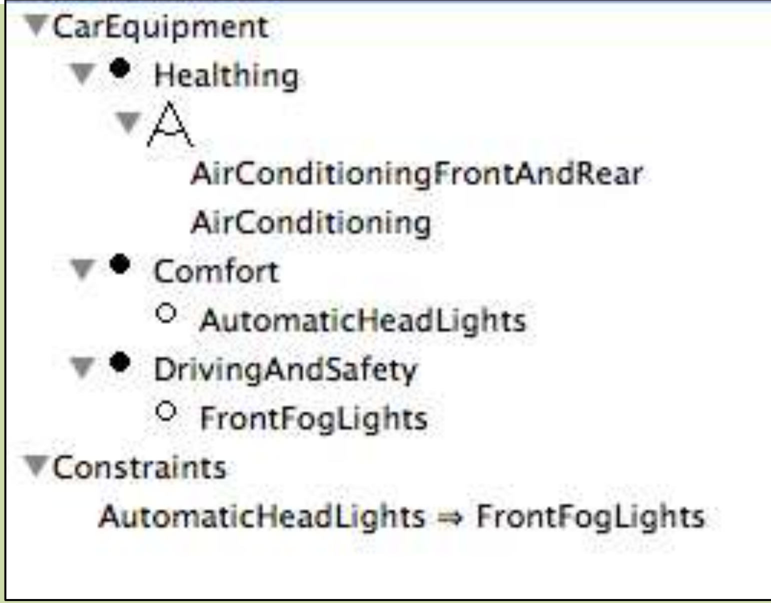
- mandatory,
- optional,
- Groups: exclusive or inclusive features
- Cross-tree constraints



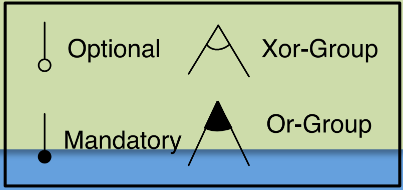


SPLC12Scripts Model | SPLC12Scripts.config

- ⊞ CarEquipment (valid, 1 possible configurations)
  - ⊞ Healthing
    - ⊟ AirConditioningFrontAndRear
    - ⊞ AirConditioning
  - ⊞ Comfort
    - ⊟ AutomaticHeadLights
  - ⊞ DrivingAndSafety
    - ⊞ FrontFogLights



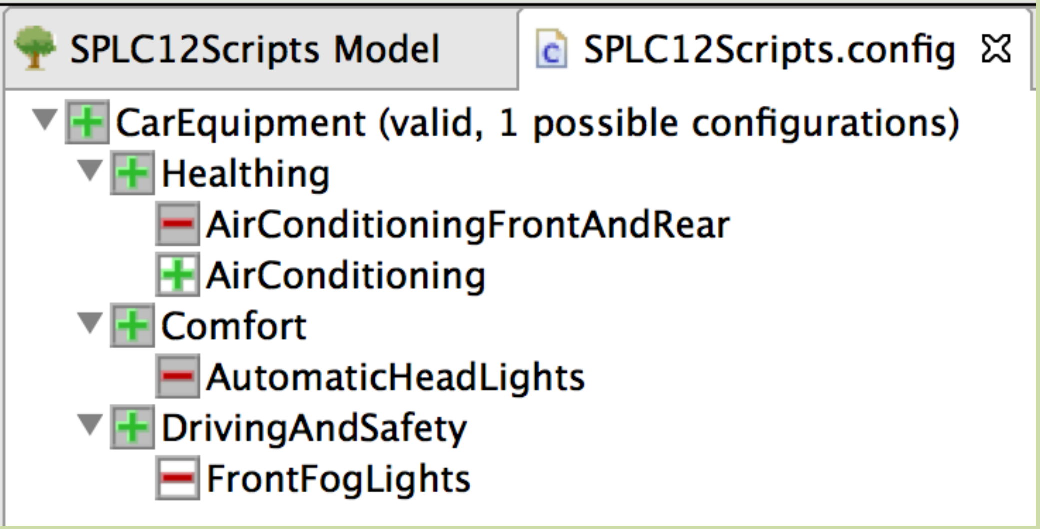
- CarEquipment
  - ● Healthing
    - A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - ○ FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights



Hierarchy + Variability
=
set of valid configurations

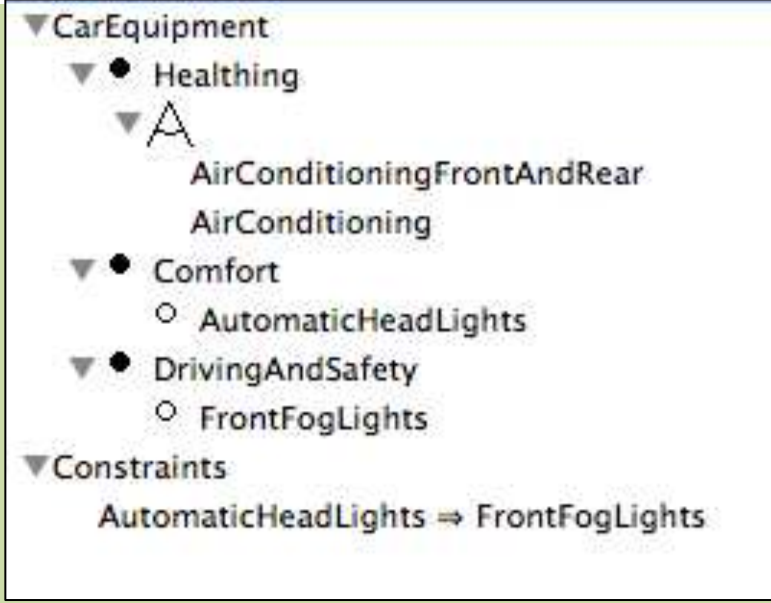
configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning, FrontFogLights}

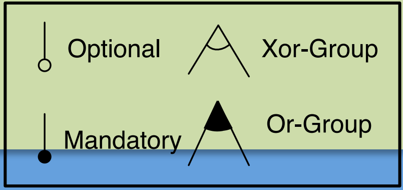


SPLC12Scripts Model | SPLC12Scripts.config

```
▼ ⊞ CarEquipment (valid, 1 possible configurations)
    ▼ ⊞ Healthing
          ⊟ AirConditioningFrontAndRear
          ⊞ AirConditioning
    ▼ ⊞ Comfort
          ⊟ AutomaticHeadLights
    ▼ ⊞ DrivingAndSafety
          ⊟ FrontFogLights
```



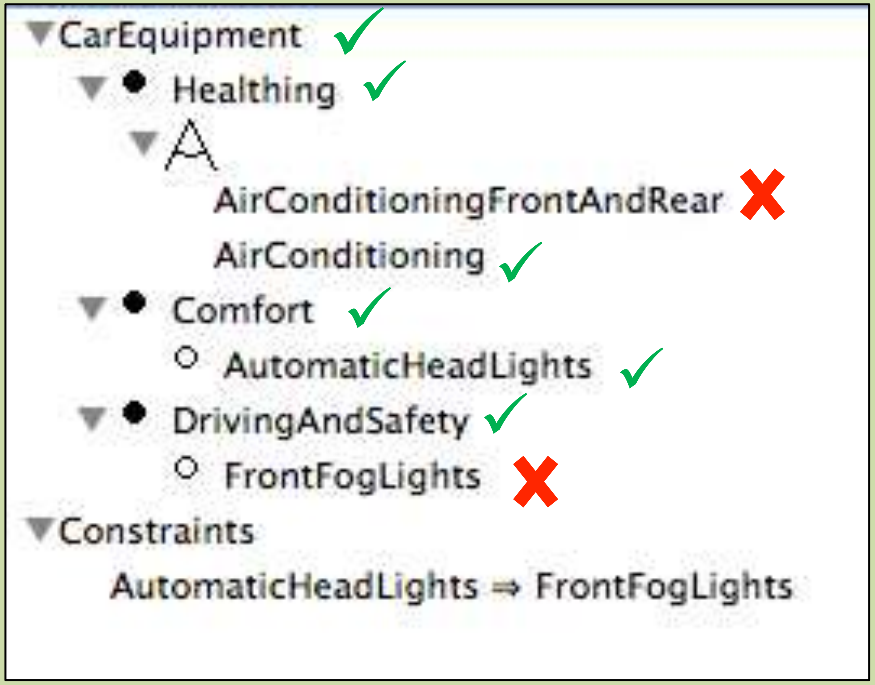
```
▼CarEquipment
    ▼ ● Healthing
        ▼ A
            AirConditioningFrontAndRear
            AirConditioning
    ▼ ● Comfort
        ○ AutomaticHeadLights
    ▼ ● DrivingAndSafety
        ○ FrontFogLights
▼Constraints
    AutomaticHeadLights ⇒ FrontFogLights
```



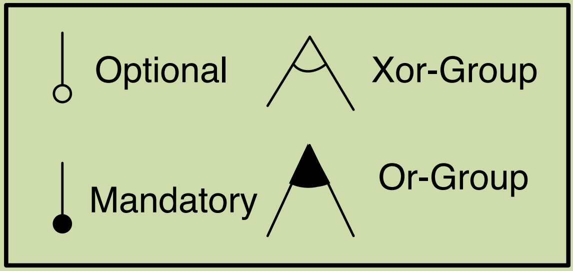
Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}



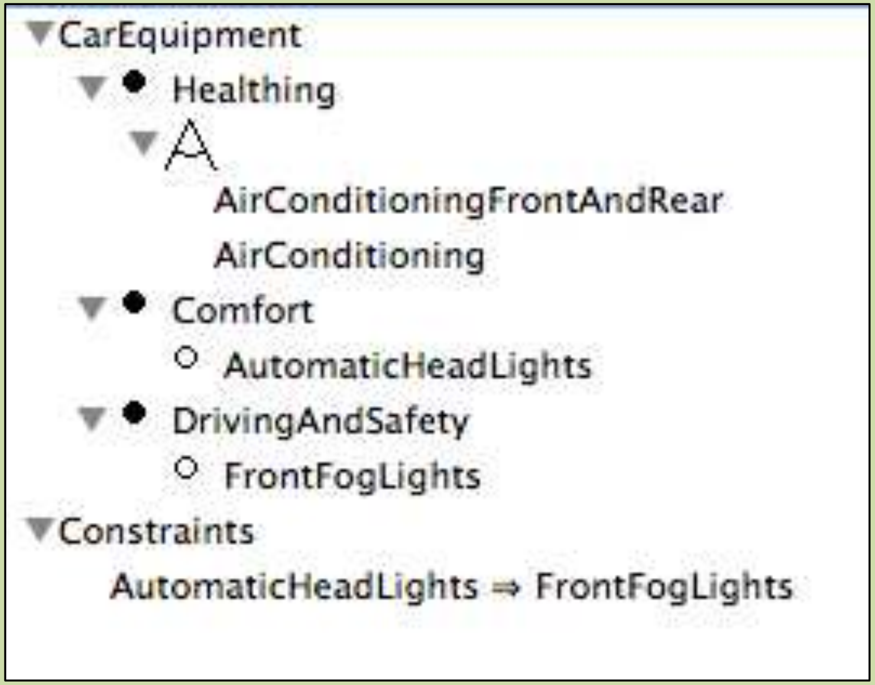
- CarEquipment ✓
  - ● Healthing ✓
    - A
      - AirConditioningFrontAndRear ✗
      - AirConditioning ✓
  - ● Comfort ✓
    - ○ AutomaticHeadLights ✓
  - ● DrivingAndSafety ✓
    - ○ FrontFogLights ✗
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights



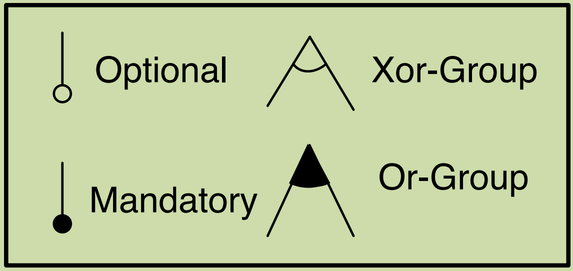
Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning, AutomaticHeadLights}



```
▼CarEquipment
  ▼ ● Healthing
      ▼ A
          AirConditioningFrontAndRear
          AirConditioning
  ▼ ● Comfort
        ○ AutomaticHeadLights
  ▼ ● DrivingAndSafety
        ○ FrontFogLights
▼Constraints
    AutomaticHeadLights ⇒ FrontFogLights
```

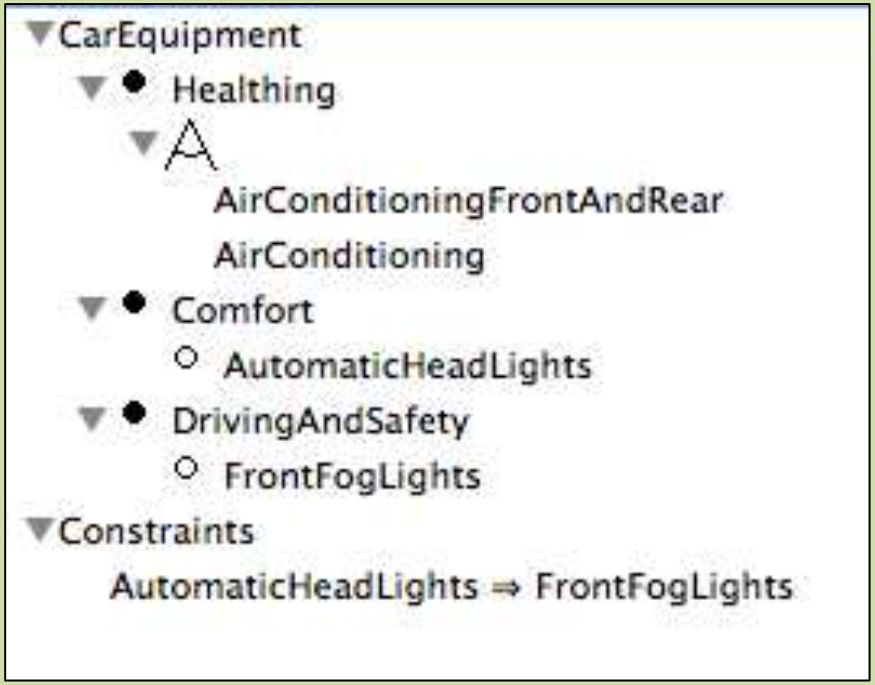


Hierarchy + Variability
=
set of valid configurations

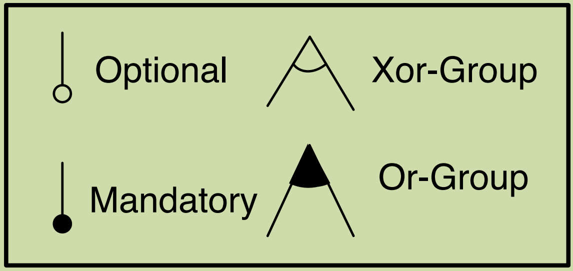
{CarEquipment, Comfort, DrivingAndSafety, Healthing}

X

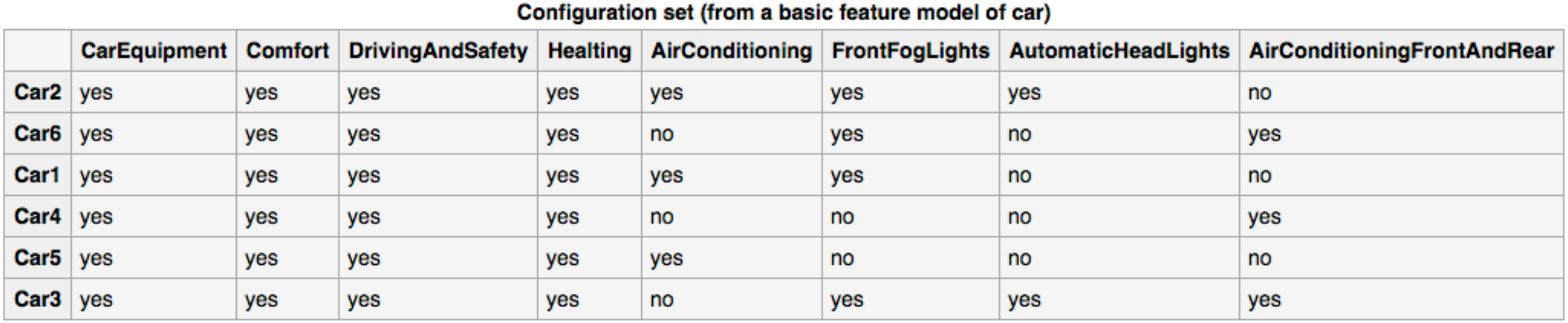
{AirConditioning, FrontFogLights}
{AutomaticHeadLights, AirConditioning, FrontFogLights}
{AutomaticHeadLights, FrontFogLights, AirConditioningFrontAndRear}
{AirConditioningFrontAndRear}
{AirConditioning}
{AirConditioningFrontAndRear, FrontFogLights}



- ▼ CarEquipment
  - ▼ ● Healthing
    - ▼ A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ▼ ● Comfort
    - ○ AutomaticHeadLights
  - ▼ ● DrivingAndSafety
    - ○ FrontFogLights
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

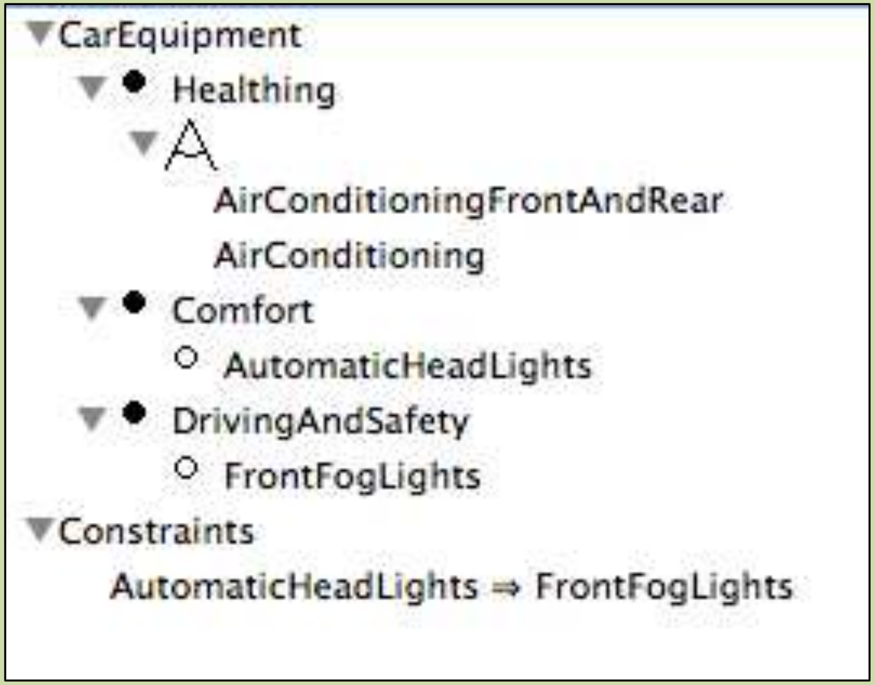


Hierarchy + Variability
=
set of valid configurations

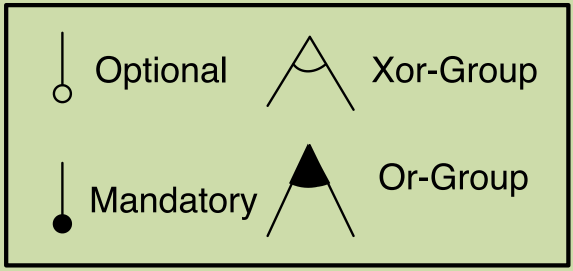


Configuration set (from a basic feature model of car)

	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car6	yes	yes	yes	yes	no	yes	no	yes
Car1	yes	yes	yes	yes	yes	yes	no	no
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car3	yes	yes	yes	yes	no	yes	yes	yes



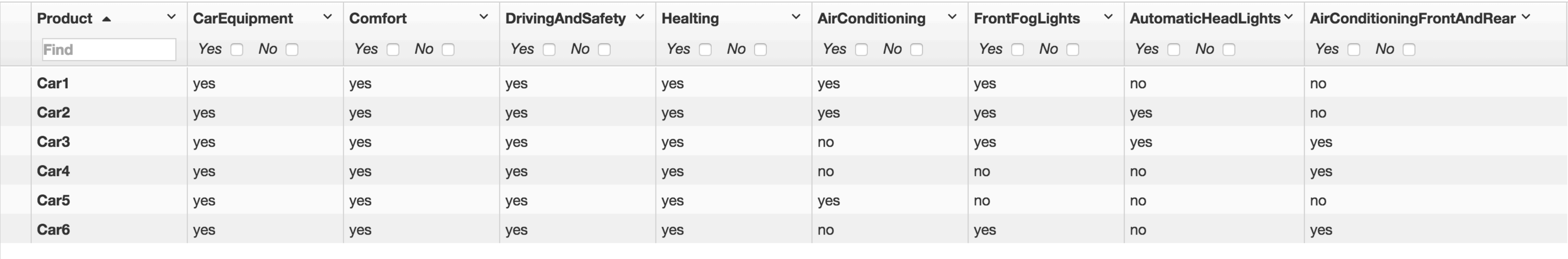
- ▼ CarEquipment
  - ▼ ● Healthing
    - ▼
      - AirConditioningFrontAndRear
      - AirConditioning
  - ▼ ● Comfort
    - ○ AutomaticHeadLights
  - ▼ ● DrivingAndSafety
    - ○ FrontFogLights
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights



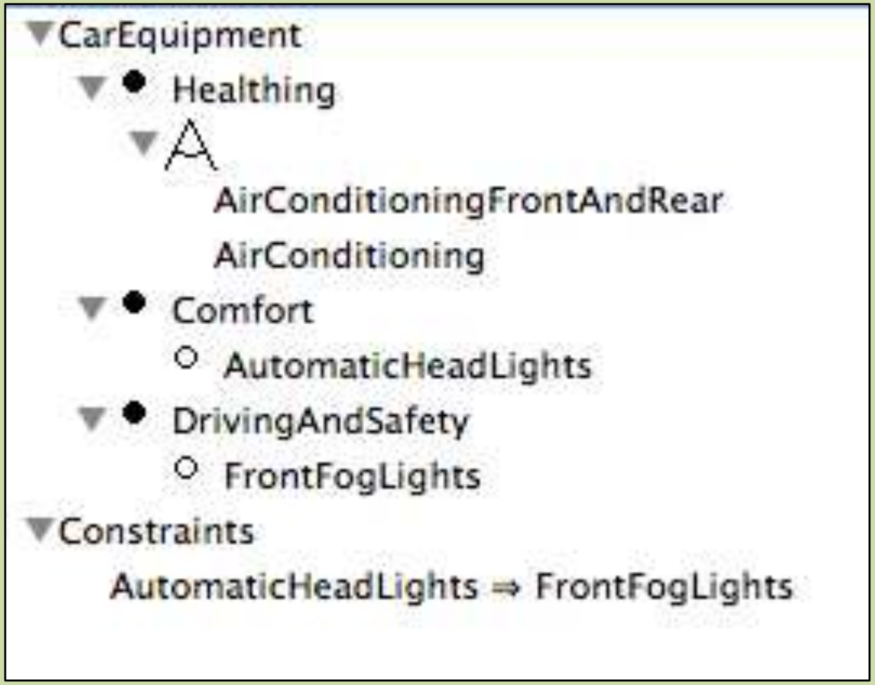
Hierarchy + Variability

=

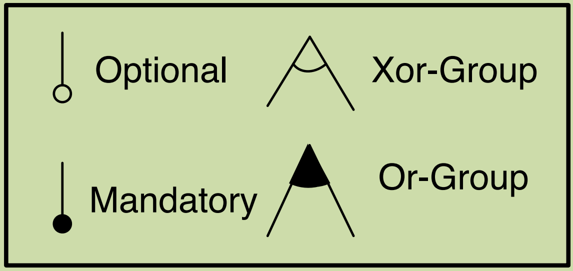
set of valid configurations



Product ▲	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □
Car1	yes	yes	yes	yes	yes	yes	no	no
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car3	yes	yes	yes	yes	no	yes	yes	yes
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car6	yes	yes	yes	yes	no	yes	no	yes



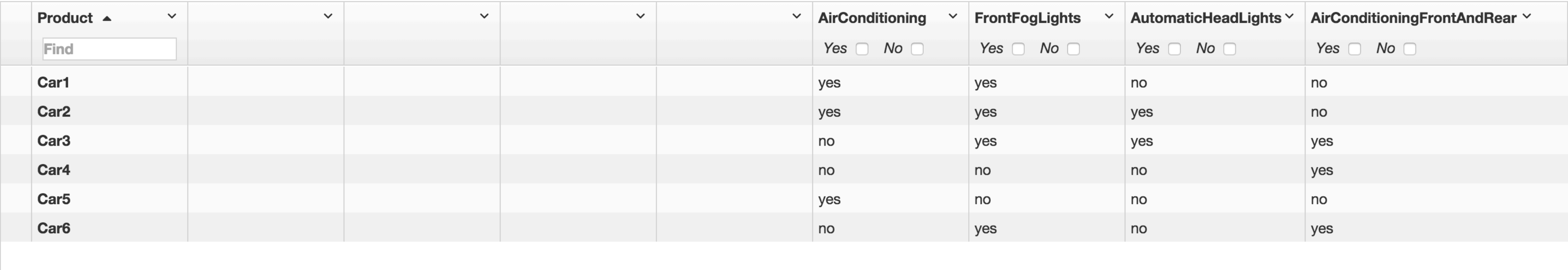
- CarEquipment
  - ● Healthing
    -
      - AirConditioningFrontAndRear
      - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - ○ FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights



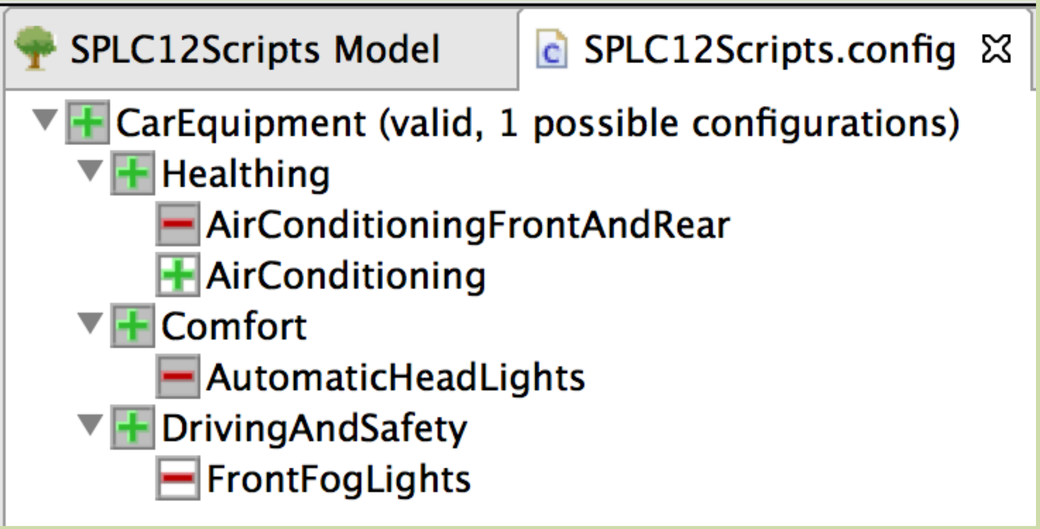
Hierarchy + Variability

=

set of valid configurations

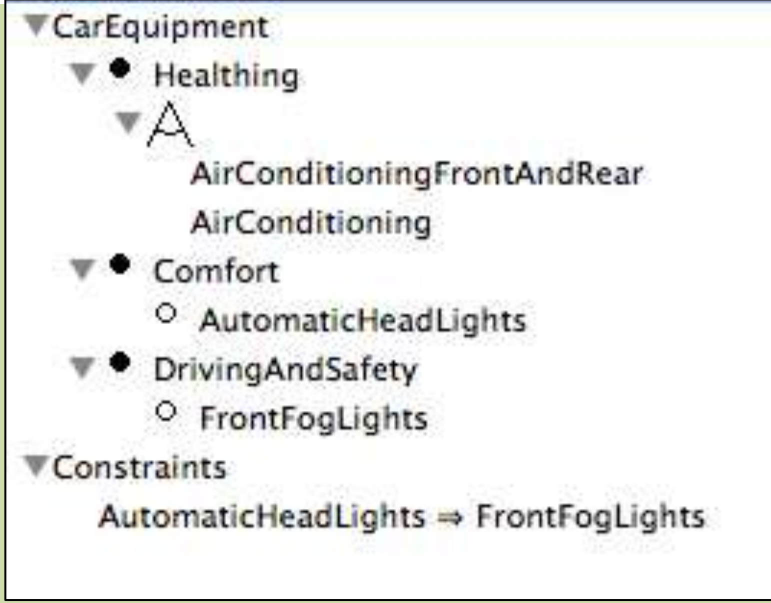


Product ▲					AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find					Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □
Car1					yes	yes	no	no
Car2					yes	yes	yes	no
Car3					no	yes	yes	yes
Car4					no	no	no	yes
Car5					yes	no	no	no
Car6					no	yes	no	yes

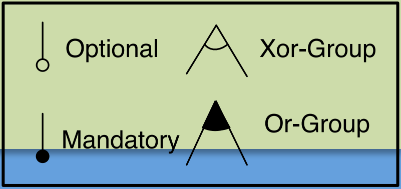


SPLC12Scripts Model | SPLC12Scripts.config

```
▼ + CarEquipment (valid, 1 possible configurations)
    ▼ + Healthing
          − AirConditioningFrontAndRear
          + AirConditioning
    ▼ + Comfort
          − AutomaticHeadLights
    ▼ + DrivingAndSafety
          − FrontFogLights
```



```
▼CarEquipment
    ▼ ● Healthing
        ▼ A
            AirConditioningFrontAndRear
            AirConditioning
    ▼ ● Comfort
        ○ AutomaticHeadLights
    ▼ ● DrivingAndSafety
        ○ FrontFogLights
▼Constraints
    AutomaticHeadLights ⇒ FrontFogLights
```



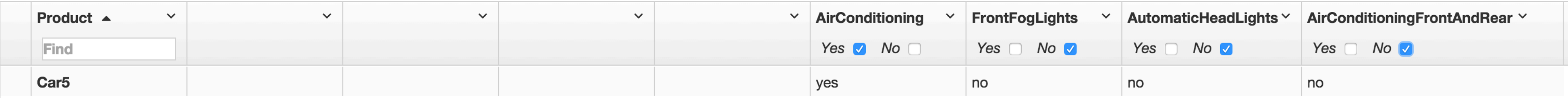
Hierarchy + Variability

=

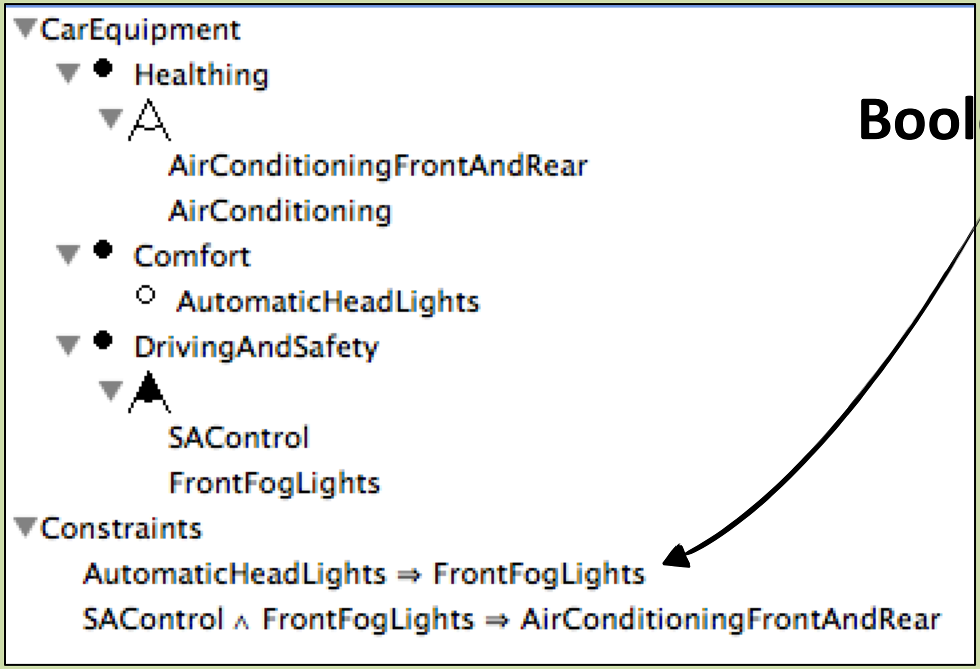
set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}

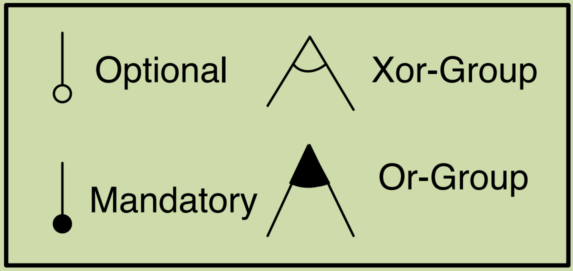


Product ▲					AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find					Yes ☑ No ☐	Yes ☐ No ☑	Yes ☐ No ☑	Yes ☐ No ☑
Car5					yes	no	no	no



- CarEquipment
  - ● Healthing
    - AirConditioningFrontAndRear
    - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - SAControl
    - FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights
  - SAControl ∧ FrontFogLights ⇒ AirConditioningFrontAndRear

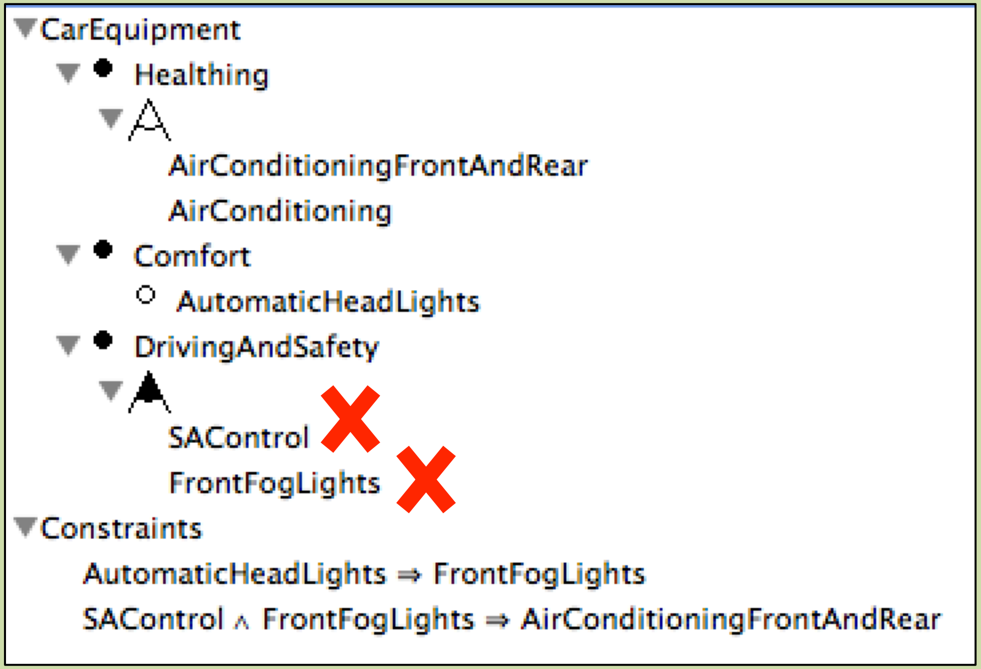
Boolean logic: ^, v, not, implie



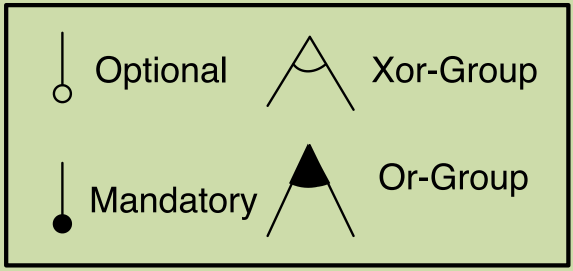
Hierarchy + Variability

=

set of valid configurations



- ▼ CarEquipment
  - ▼ ● Healthing
    - ▼ A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ▼ ● Comfort
    - ○ AutomaticHeadLights
  - ▼ ● DrivingAndSafety
    - ▼ ▲
      - SAControl ✗
      - FrontFogLights ✗
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights
  - SAControl ∧ FrontFogLights ⇒ AirConditioningFrontAndRear

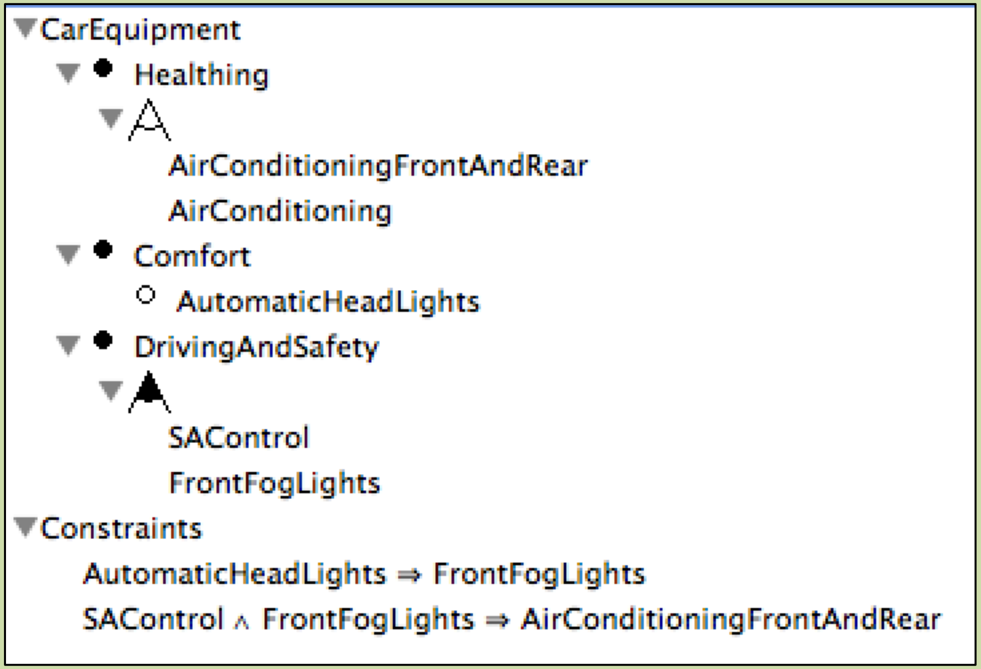


Hierarchy + Variability

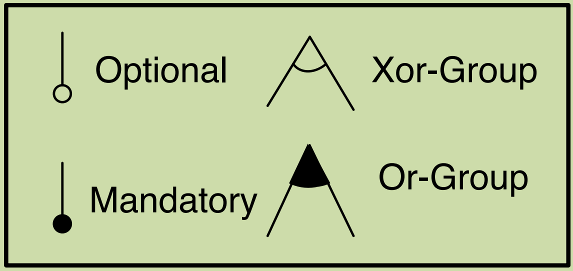
=

set of valid configurations

Or-group: at least one!



- CarEquipment
  - ● Healthing
    - AirConditioningFrontAndRear
    - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - SAControl
    - FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights
  - SAControl ∧ FrontFogLights ⇒ AirConditioningFrontAndRear

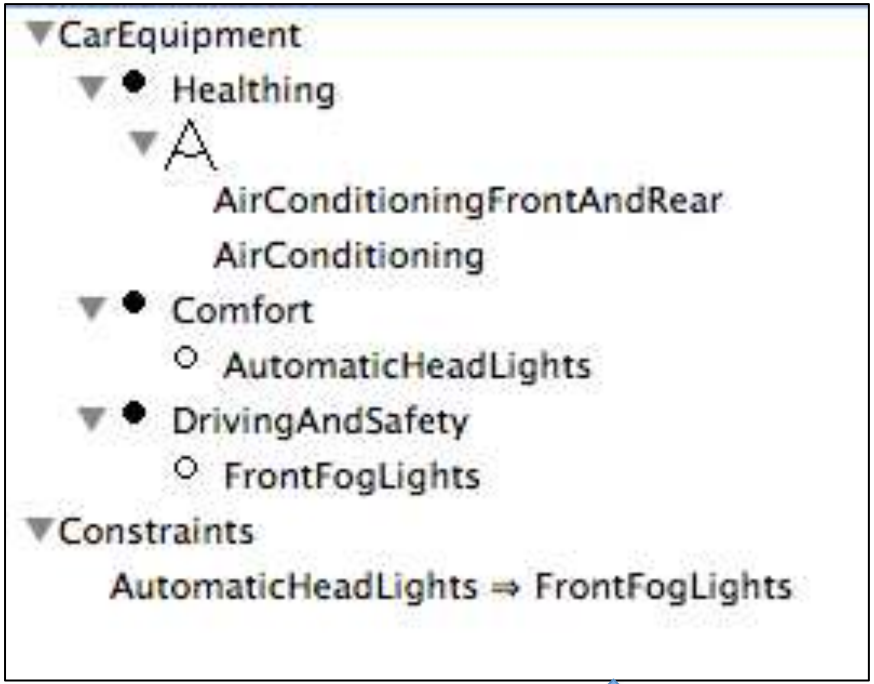


Hierarchy + Variability
=
set of valid configurations

{CarEquipment, Comfort, DrivingAndSafety, Healthing}

X

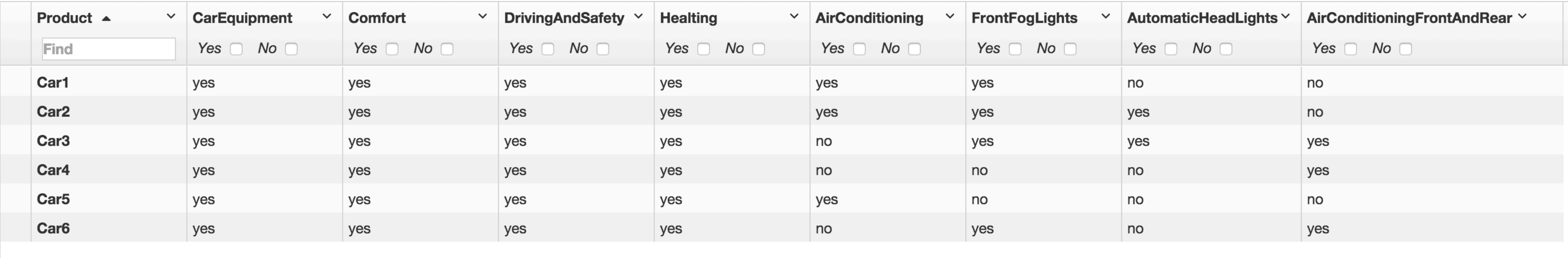
{AirConditioningFrontAndRear, FrontFogLights, SAControl}
{AirConditioningFrontAndRear, SAControl}
{AutomaticHeadLights, AirConditioning, FrontFogLights}
{AirConditioningFrontAndRear, SAControl, AutomaticHeadLights, FrontFogLights
{FrontFogLights, AirConditioning}
{AutomaticHeadLights, AirConditioningFrontAndRear,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



```
▼CarEquipment
  ▼ ● Healthing
     ▼A
        AirConditioningFrontAndRear
        AirConditioning
  ▼ ● Comfort
       ○ AutomaticHeadLights
  ▼ ● DrivingAndSafety
       ○ FrontFogLights
▼Constraints
    AutomaticHeadLights ⇒ FrontFogLights
```

(Boolean)
Feature Models

(Boolean)
Formula φ

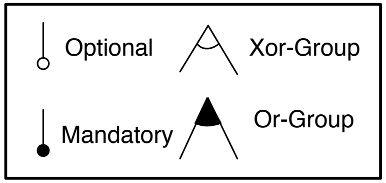


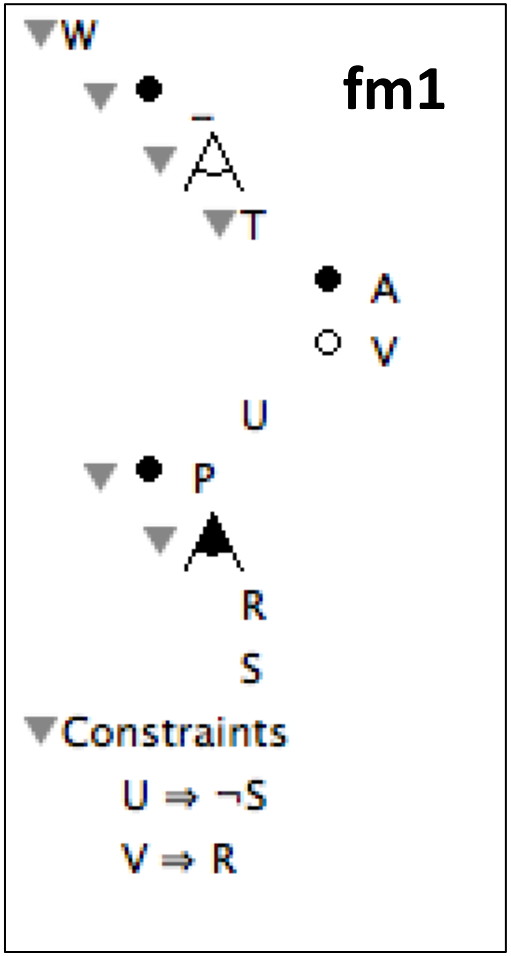
Product ▲	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □	Yes □ No □
Car1	yes	yes	yes	yes	yes	yes	no	no
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car3	yes	yes	yes	yes	no	yes	yes	yes
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car6	yes	yes	yes	yes	no	yes	no	yes

(Boolean)
Product Comparison Matrix

(Boolean) Feature Models

Hierarchy + Variability = set of valid configurations

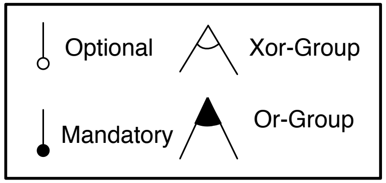


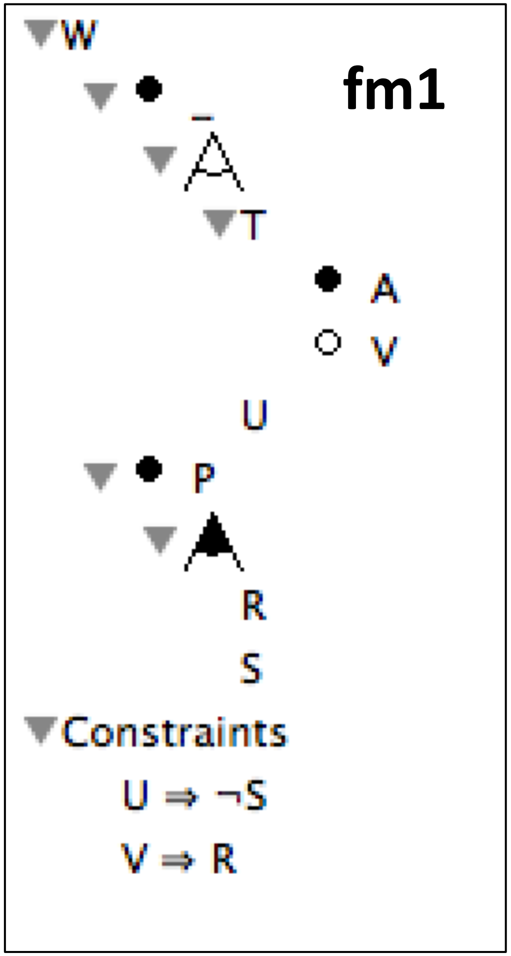


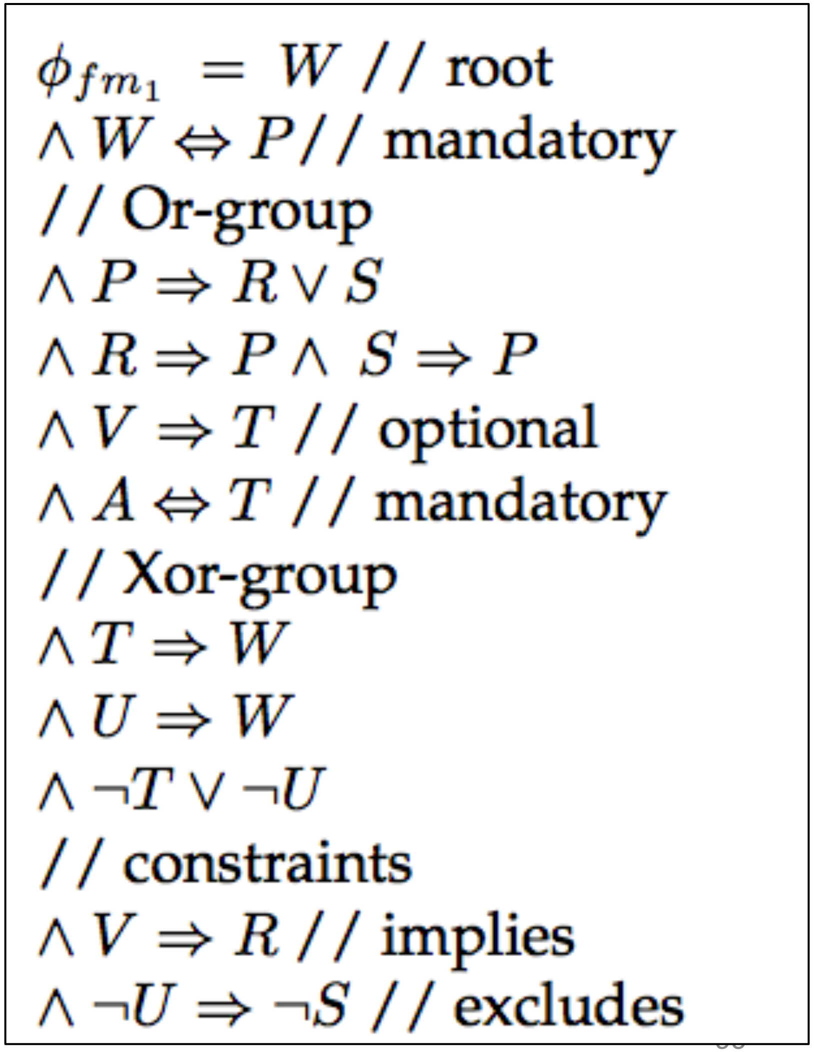
$$
\begin{aligned}
&[\![fm1]\!] = \{\\
&\{W, P, R, S, T, A, V\},\\
&\{W, P, S, T, A\},\\
&\{W, P, R, T, A\},\\
&\{W, P, R, U\},\\
&\{W, P, R, T, V, A\},\\
&\{W, P, R, S, T, A\},\\
&\}
\end{aligned}
$$

(Boolean) Feature Models

~ Boolean formula







$\phi_{fm_1} = W$ // root

$\wedge W \Leftrightarrow P$// mandatory

// Or-group

$\wedge P \Rightarrow R \vee S$

$\wedge R \Rightarrow P \wedge \; S \Rightarrow P$

$\wedge V \Rightarrow T$ // optional

$\wedge A \Leftrightarrow T$ // mandatory

// Xor-group

$\wedge T \Rightarrow W$

$\wedge U \Rightarrow W$

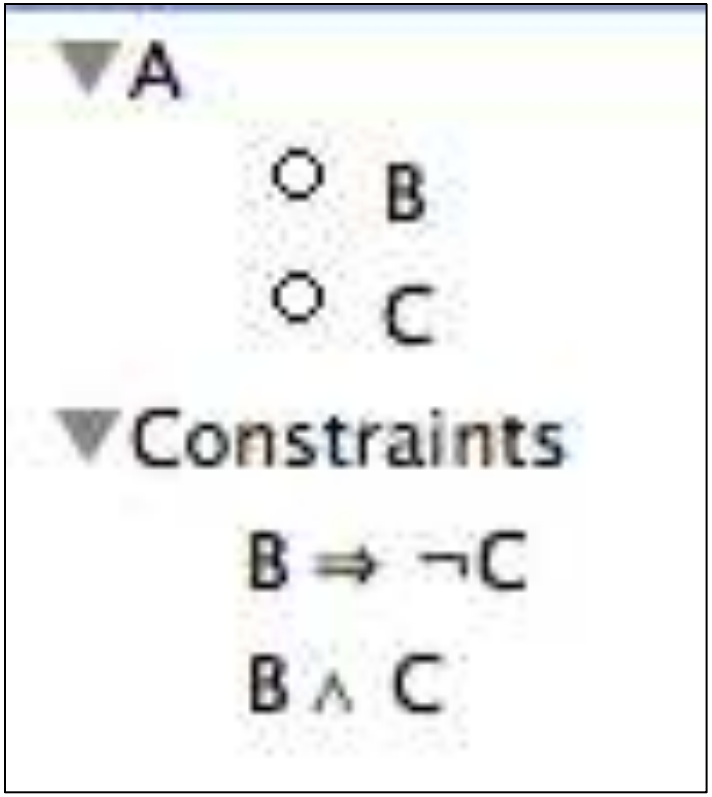
$\wedge \neg T \vee \neg U$

// constraints

$\wedge V \Rightarrow R$ // implies

$\wedge \neg U \Rightarrow \neg S$ // excludes

I want to analyze and play with my specification!



▼A

○ B

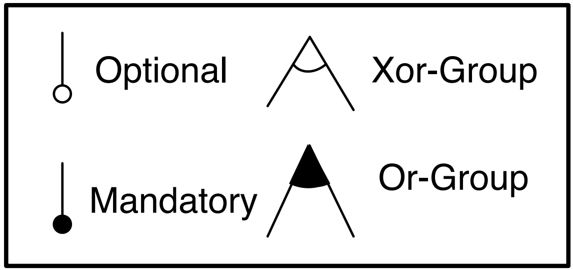
○ C

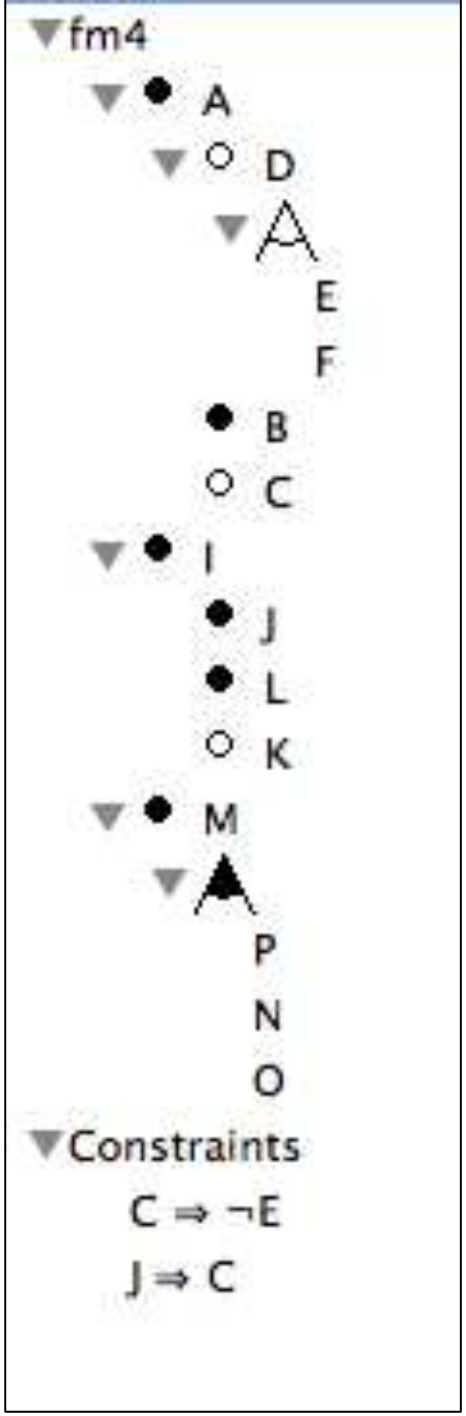
▼Constraints

$B \Rightarrow \neg C$

$B \wedge C$

Empty set of configurations

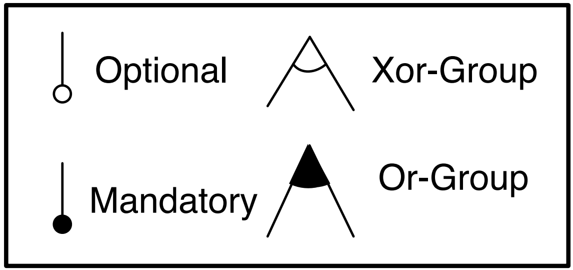


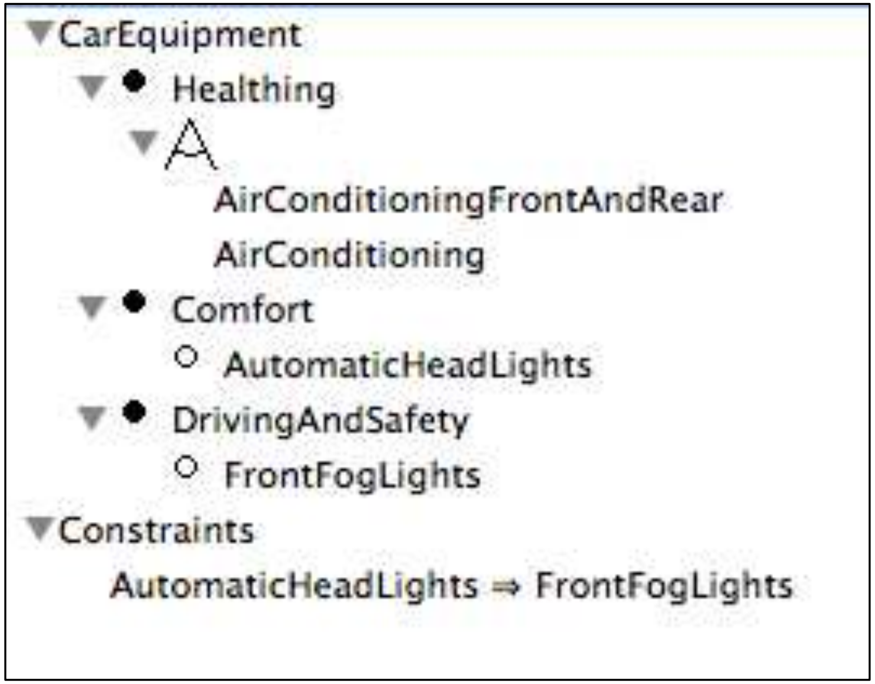


```
▼fm4
  ▼ ● A
      ▼ ○ D
          ▼
              E
              F
        ● B
        ○ C
  ▼ ● I
        ● J
        ● L
        ○ K
  ▼ ● M
      ▼
          P
          N
          O
▼Constraints
    C ⇒ ¬E
    J ⇒ C
```

Dead feature

False optional feature

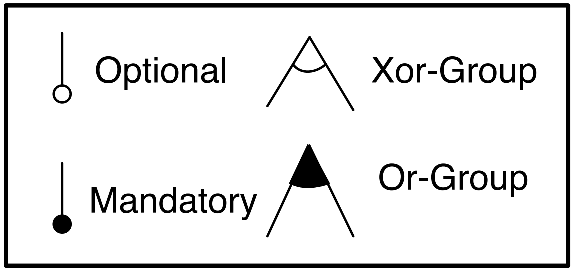


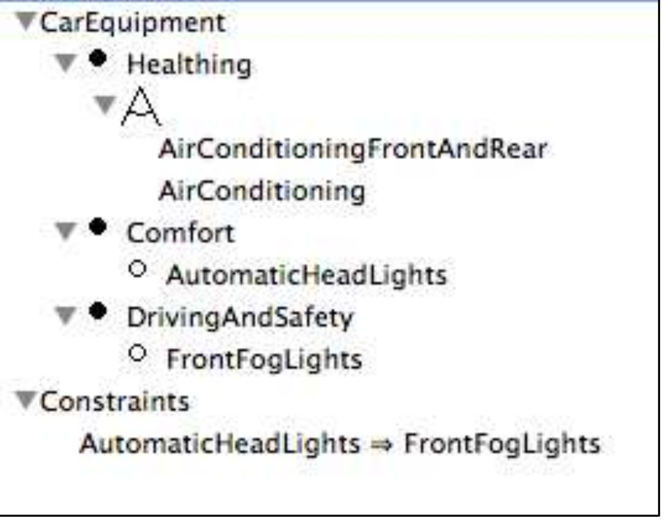


- ▼ CarEquipment
  - ▼ ● Healthing
    - ▼ ∧
      - AirConditioningFrontAndRear
      - AirConditioning
  - ▼ ● Comfort
    - ○ AutomaticHeadLights
  - ▼ ● DrivingAndSafety
    - ○ FrontFogLights
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

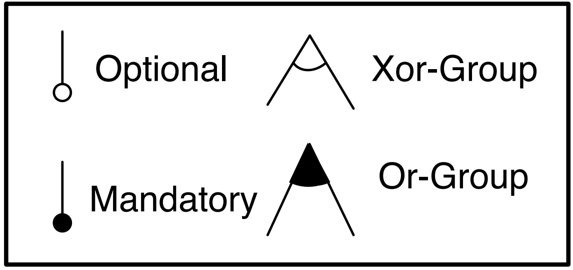
Core features

{CarEquipment, Comfort, DrivingAndSafety, Healthing}

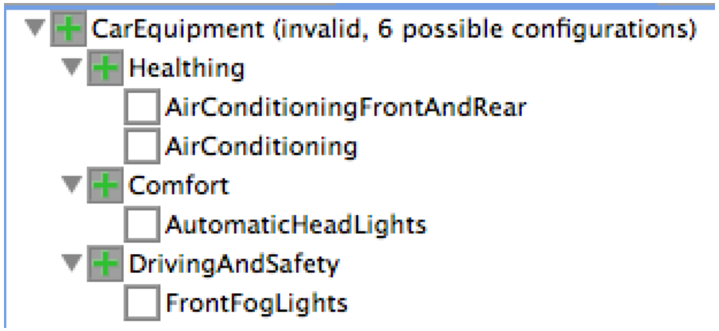




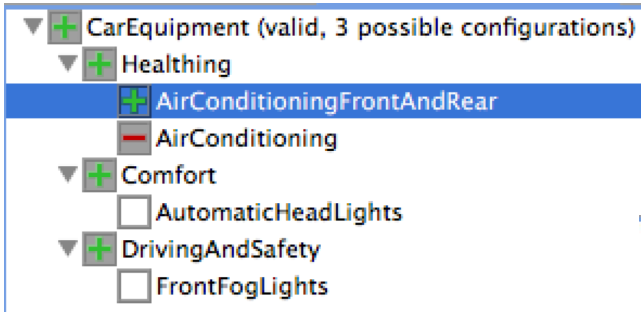
```
▼CarEquipment
    ▼ ● Healthing
        ▼ A
            AirConditioningFrontAndRear
            AirConditioning
    ▼ ● Comfort
        ○ AutomaticHeadLights
    ▼ ● DrivingAndSafety
        ○ FrontFogLights
▼Constraints
    AutomaticHeadLights ⇒ FrontFogLights
```



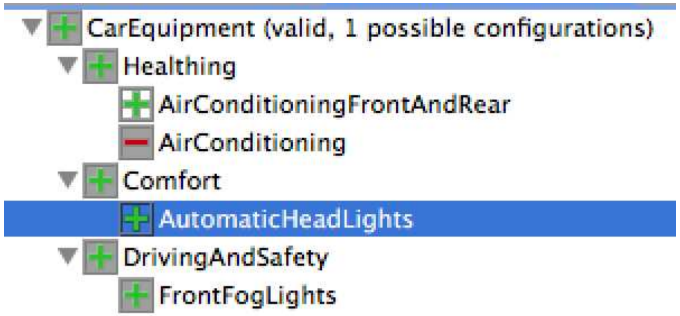
Interactive Configuration



```
▼ + CarEquipment (invalid, 6 possible configurations)
    ▼ + Healthing
        □ AirConditioningFrontAndRear
        □ AirConditioning
    ▼ + Comfort
        □ AutomaticHeadLights
    ▼ + DrivingAndSafety
        □ FrontFogLights
```



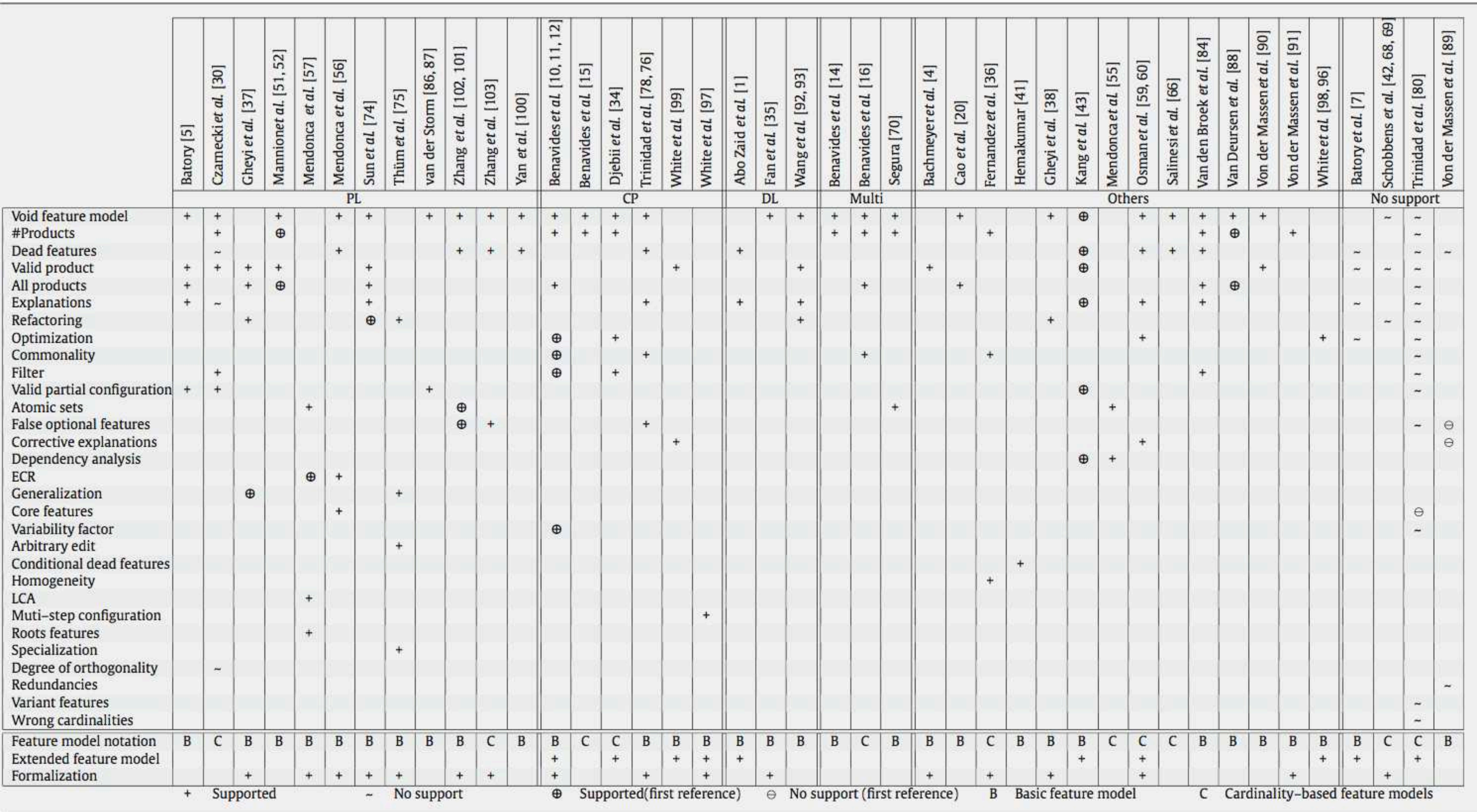
```
▼ + CarEquipment (valid, 3 possible configurations)
    ▼ + Healthing
        + AirConditioningFrontAndRear
        − AirConditioning
    ▼ + Comfort
        □ AutomaticHeadLights
    ▼ + DrivingAndSafety
        □ FrontFogLights
```



```
▼ + CarEquipment (valid, 1 possible configurations)
    ▼ + Healthing
        + AirConditioningFrontAndRear
        − AirConditioning
    ▼ + Comfort
        + AutomaticHeadLights
    ▼ + DrivingAndSafety
        + FrontFogLights
```

Feature Models and Automated Reasoning

Benavides et al. survey, 2010



	Batory [5]	Czarnecki *et al.* [30]	Gheyi *et al.* [37]	Mannion*et al.* [51, 52]	Mendonca *et al.* [57]	Mendonca *et al.* [56]	Sun *et al.* [74]	Thüm *et al.* [75]	van der Storm [86, 87]	Zhang *et al.* [102, 101]	Zhang *et al.* [103]	Yan *et al.* [100]	Benavides*et al.* [10, 11, 12]	Benavides *et al.* [15]	Djebii *et al.* [34]	Trinidad *et al.* [78, 76]	White *et al.* [99]	White *et al.* [97]	Abo Zaid *et al.* [1]	Fan *et al.* [35]	Wang *et al.* [92, 93]	Benavides *et al.* [14]	Benavides *et al.* [16]	Segura [70]	Bachmeyer *et al.* [4]	Cao *et al.* [20]	Fernandez *et al.* [36]	Hemakumar [41]	Gheyi *et al.* [38]	Kang *et al.* [43]	Mendonca *et al.* [55]	Osman *et al.* [59, 60]	Salinesi *et al.* [66]	Van den Broek *et al.* [84]	Van Deursen *et al.* [88]	Von der Massen *et al.* [90]	Von der Massen *et al.* [91]	White *et al.* [98, 96]	Batory *et al.* [7]	Schobbens *et al.* [42, 68, 69]	Trinidad *et al.* [80]	Von der Massen *et al.* [89]
	PL												CP						DL			Multi			Others														No support			
Void feature model	+	+		+		+	+		+	+	+	+	+	+	+	+				+	+	+	+	+		+			+	⊕		+	+	+	+	+				~	~	
#Products		+		⊕									+	+	+							+	+	+			+							+	⊕		+				~	
Dead features		~				+				+	+	+				+			+											⊕		+	+	+					~		~	~
Valid product	+	+	+	+			+										+				+				+					⊕						+			~	~	~	
All products	+		+	⊕			+						+										+			+								+	⊕						~	
Explanations	+	~					+									+			+		+									⊕		+		+					~		~	
Refactoring			+				⊕	+													+								+											~	~	
Optimization													⊕		+																	+						+	~		~	
Commonality													⊕			+							+				+														~	
Filter		+											⊕		+																			+							~	
Valid partial configuration	+	+							+																					⊕											~	
Atomic sets					+					⊕														+							+											
False optional features										⊕	+					+																									~	⊖
Corrective explanations																	+															+										⊖
Dependency analysis																														⊕	+											
ECR					⊕	+																																				
Generalization			⊕					+																																		
Core features						+																																			⊖	
Variability factor													⊕																												~	
Arbitrary edit								+																																		
Conditional dead features																												+														
Homogeneity																											+															
LCA					+																																					
Muti-step configuration																		+																								
Roots features					+																																					
Specialization								+																																		
Degree of orthogonality		~																																								
Redundancies																																										~
Variant features																																									~	
Wrong cardinalities																																									~	
Feature model notation	B	C	B	B	B	B	B	B	B	B	C	B	B	C	C	B	B	B	B	B	B	B	C	B	B	B	C	B	B	B	C	C	C	B	B	B	B	B	B	C	C	B
Extended feature model													+		+		+	+	+											+		+						+	+		+	
Formalization			+		+	+	+	+		+	+		+			+		+		+					+		+		+			+					+			+		

+ Supported ~ No support ⊕ Supported(first reference) ⊖ No support (first reference) B Basic feature model C Cardinality–based feature models

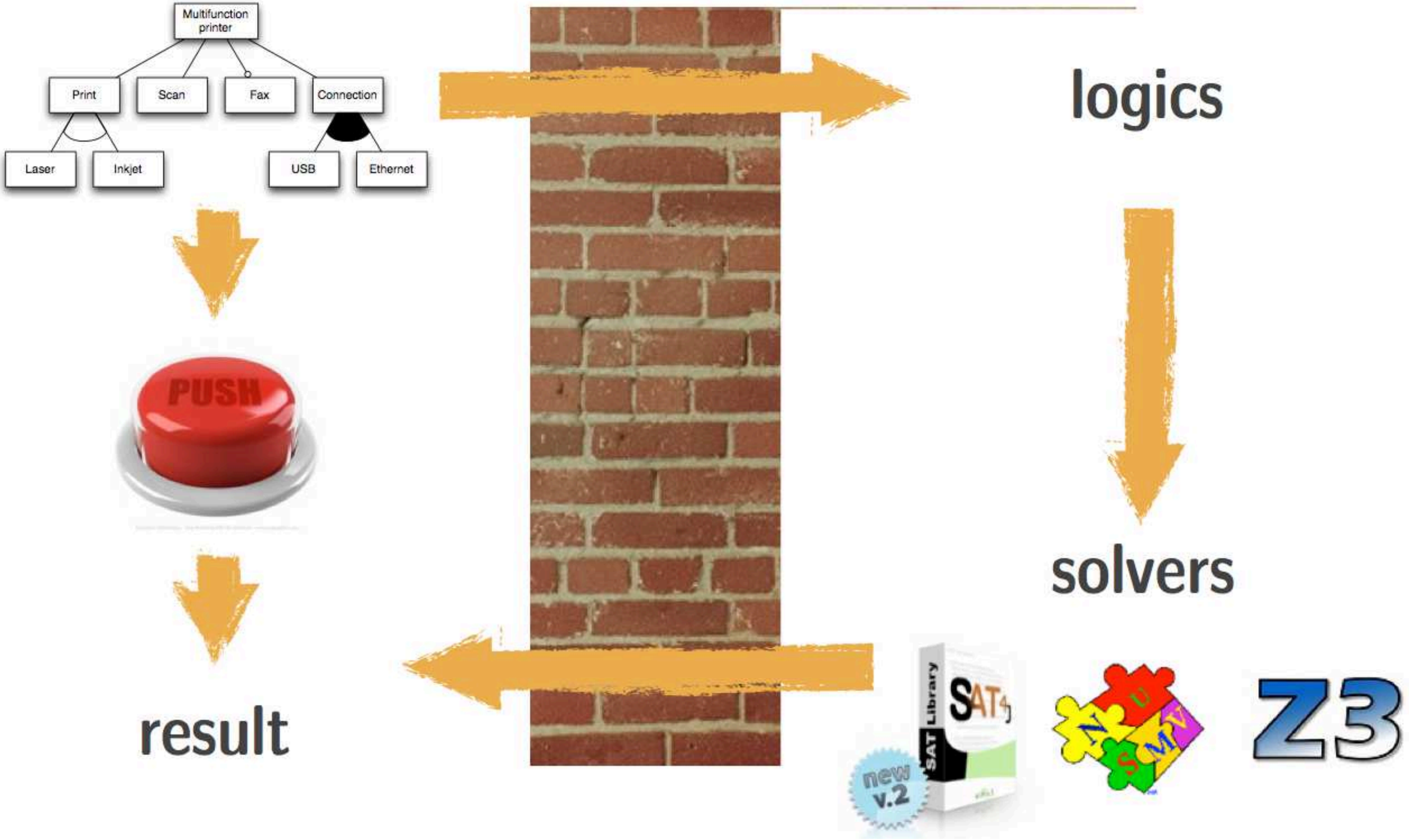
Decision problems and complexity

- Validity of a feature model
- Validity of a configuration
- Computation of dead and core features
- Counting of the number of valid configurations
- Equivalence between two feature models
- Satisfiability (SAT) problem
  - NP-complete

How to automate analysis of your feature models?

Binary Decision Diagram (BDD)
SAT solver

Typical implementations



logics

solvers

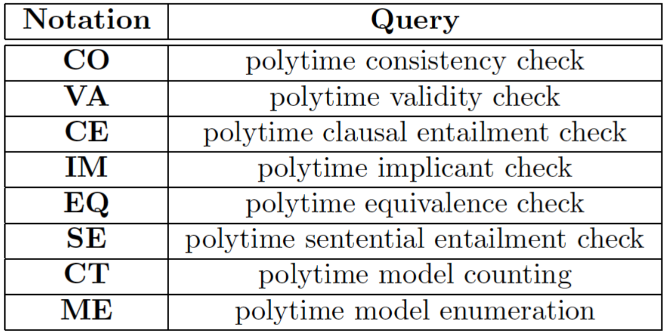
result

A knowledge compilation map

Adnan Darwiche and Pierre Marquis

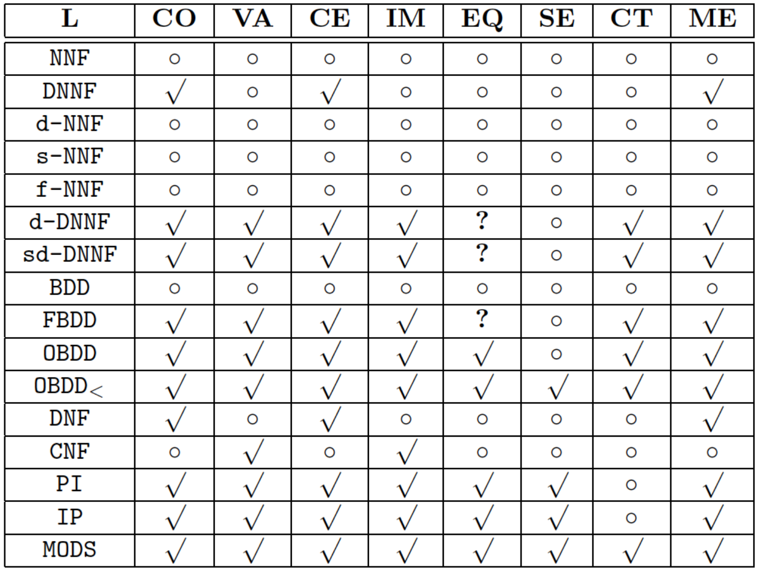
Journal of Artificial Intelligence Research Volume 17 Issue 1, July 2002, Pages 229-264

(note: one of the best paper I ever read)



Notation	Query
CO	polytime consistency check
VA	polytime validity check
CE	polytime clausal entailment check
IM	polytime implicant check
EQ	polytime equivalence check
SE	polytime sentential entailment check
CT	polytime model counting
ME	polytime model enumeration

Table 4: Notations for queries.



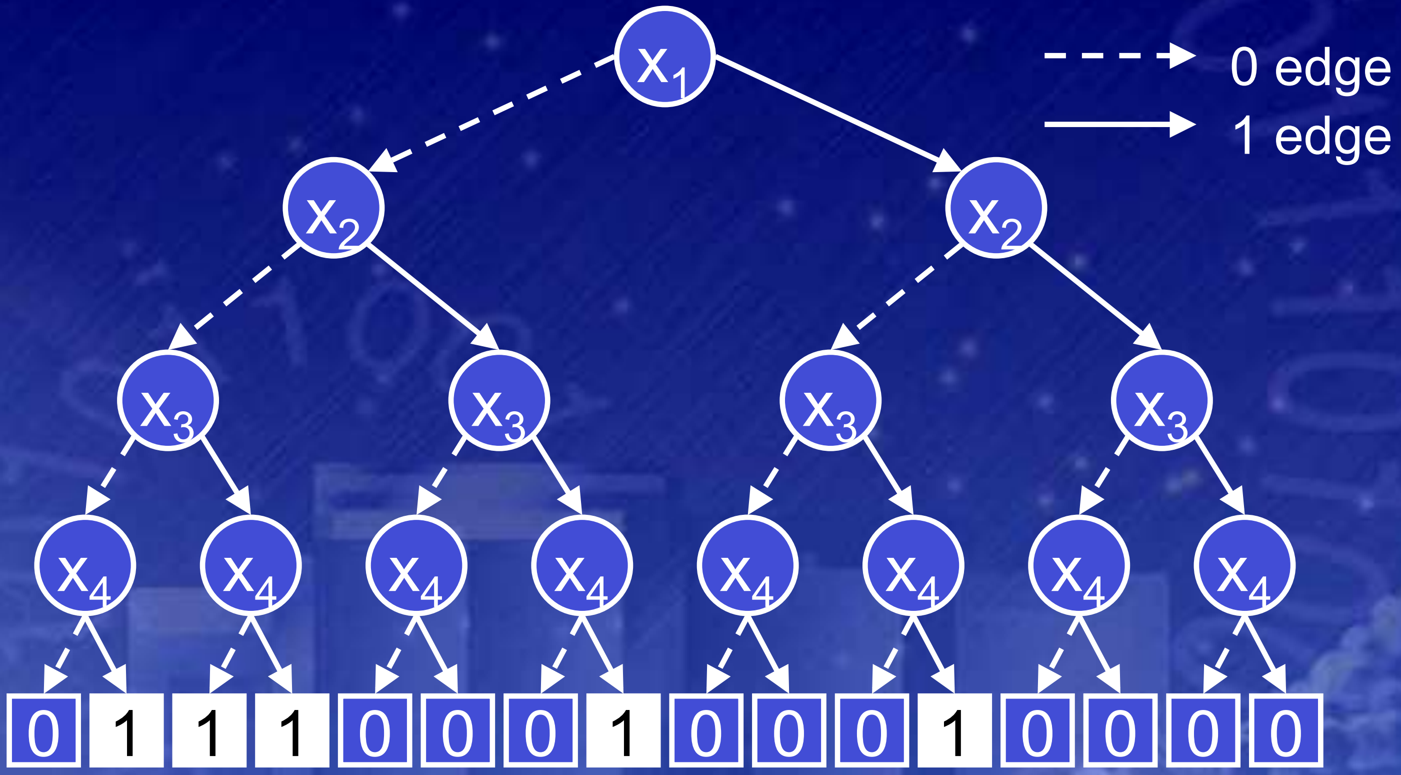
L	CO	VA	CE	IM	EQ	SE	CT	ME
NNF	○	○	○	○	○	○	○	○
DNNF	√	○	√	○	○	○	○	√
d-NNF	○	○	○	○	○	○	○	○
s-NNF	○	○	○	○	○	○	○	○
f-NNF	○	○	○	○	○	○	○	○
d-DNNF	√	√	√	√	?	○	√	√
sd-DNNF	√	√	√	√	?	○	√	√
BDD	○	○	○	○	○	○	○	○
FBDD	√	√	√	√	?	○	√	√
OBDD	√	√	√	√	√	○	√	√
$\text{OBDD}_<$	√	√	√	√	√	√	√	√
DNF	√	○	√	○	○	○	○	√
CNF	○	√	○	√	○	○	○	○
PI	√	√	√	√	√	√	○	√
IP	√	√	√	√	√	√	○	√
MODS	√	√	√	√	√	√	√	√

Table 5: Subsets of the NNF language and their corresponding polytime queries. √ means "satisfies" and ○ means "does not satisfy unless P = NP."

Binary Decision Diagrams (Bryant 1986)

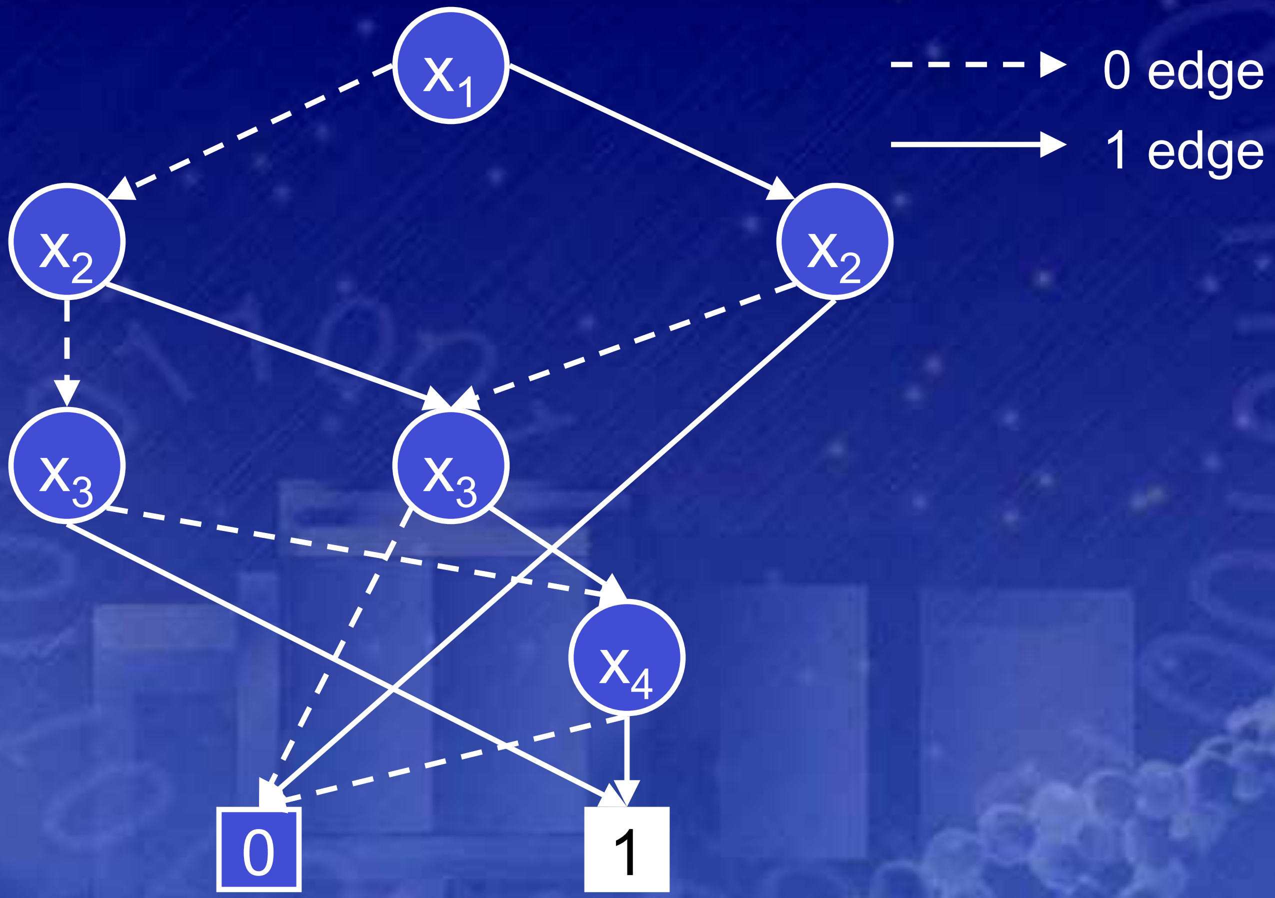
encoding of a truth table.

from		*to*		
x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Binary Decision Diagrams

(after reduction)

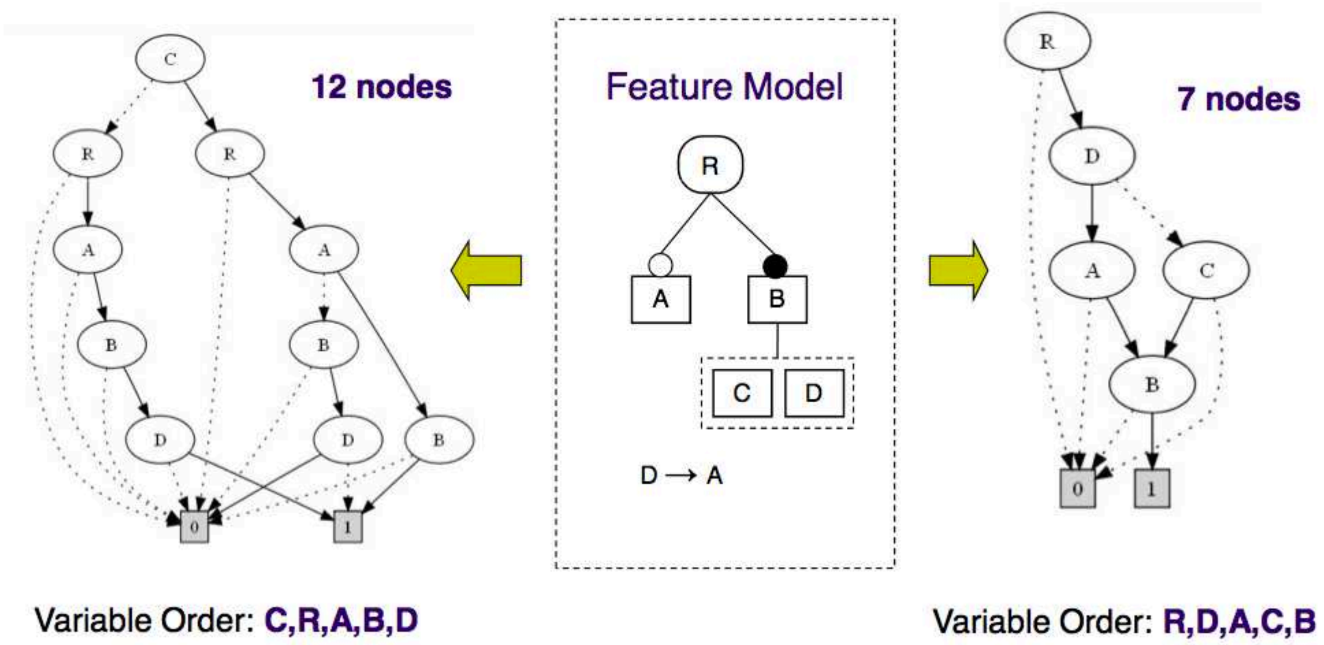


Binary Decision Diagrams (BDDs)

- Very efficient structure for most of the satisfiability operations
- Polynomial in time for checking satisfiability and determining equivalence between two BDDs
- Graph trasversal
- So great?

Binary Decision Diagrams (BDDs): Theoretical Problem

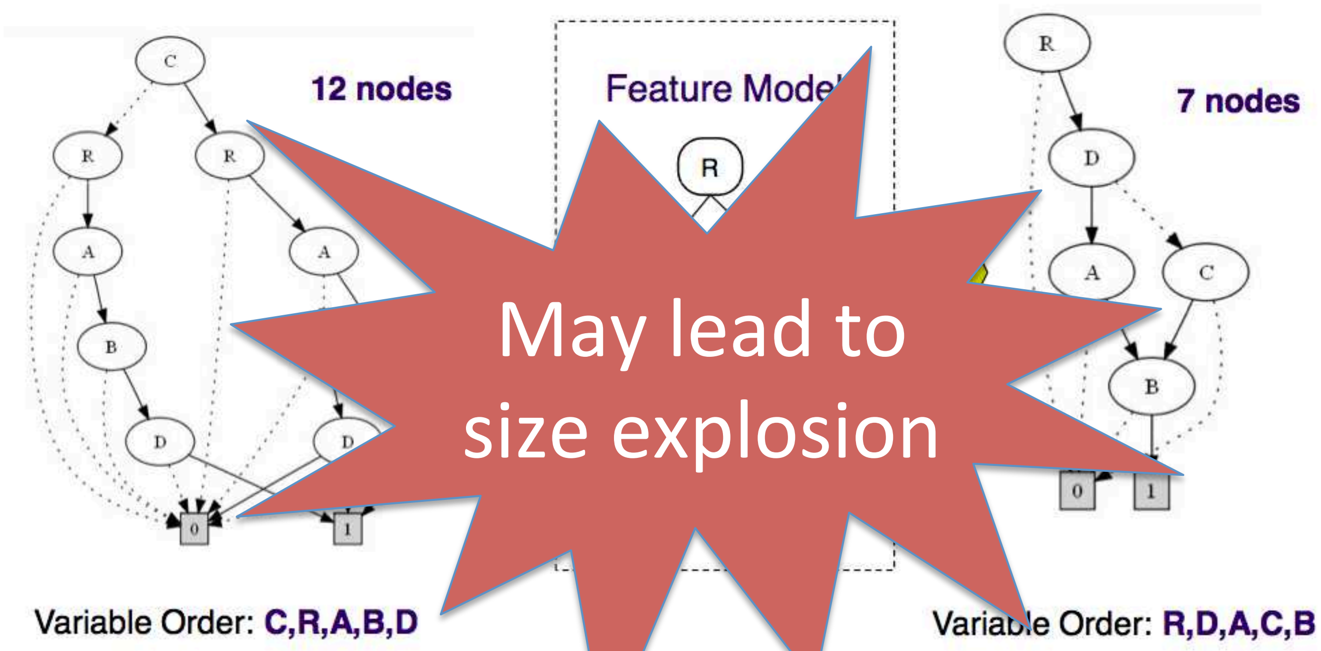
- The size of the BDD is very sensitive to the order of the BDD variables
  - **e.g. two equivalent BDDs for the same feature**



[Mendonca, slide]

Binary Decision Diagrams (BDDs): Theoretical Problem

- The size of the BDD is very sensitive to the order of the BDD variables
  - **e.g. two equivalent BDDs for the same feature**

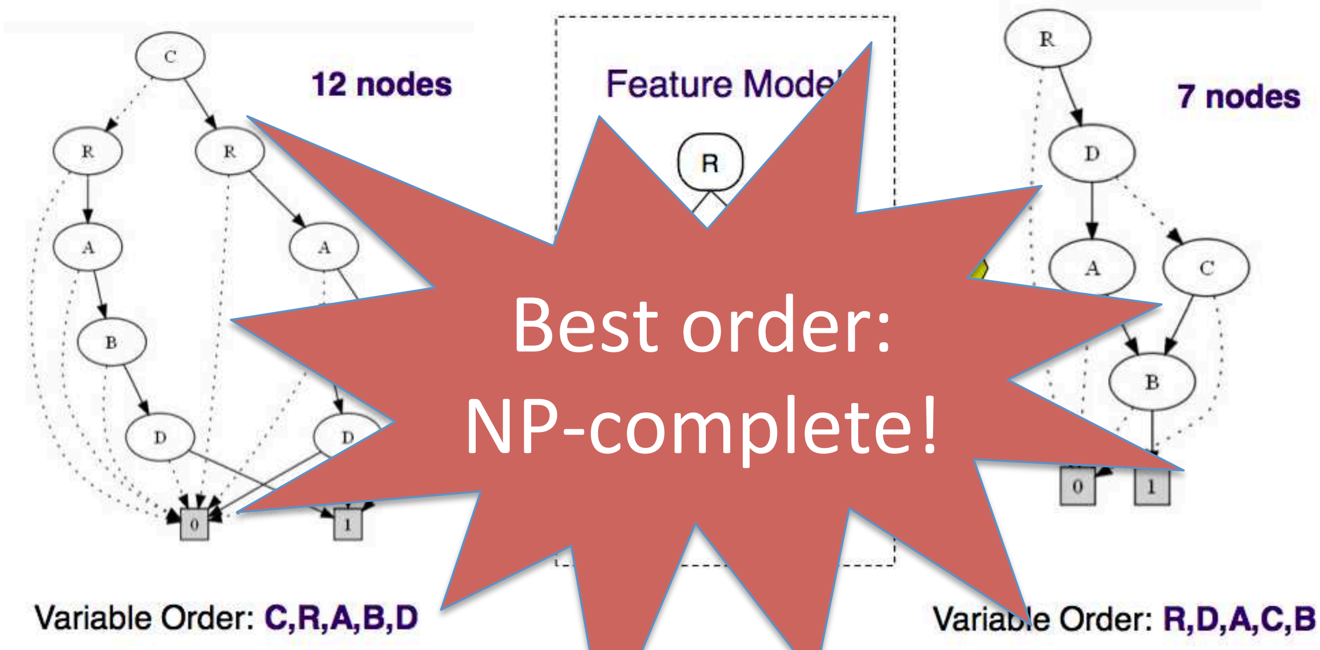


May lead to size explosion

[Mendonca, slide]

Binary Decision Diagrams (BDDs): Theoretical Problem

- The size of the BDD is very sensitive to the order of the BDD variables
  - **e.g. two equivalent BDDs for the same feature**

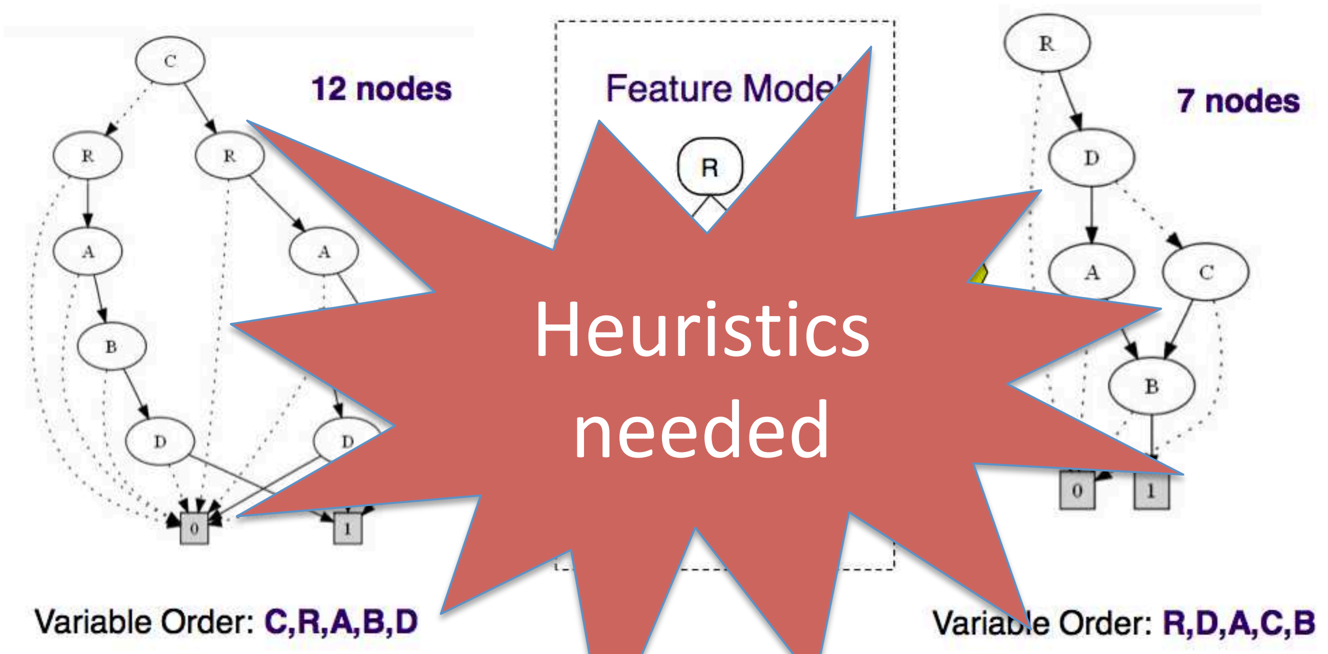


Best order: NP-complete!

[Mendonca, slide]

Binary Decision Diagrams (BDDs): Practical Problem

- The size of the BDD is very sensitive to the order of the BDD variables. In practice: **BDDs cannot be build for feature models with 2000+ features**



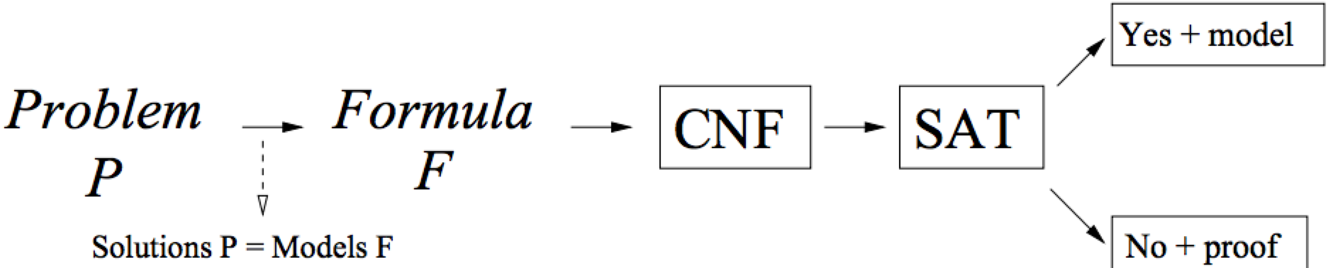
[Mendonca, slide]

How to automate analysis of your feature models?

Let us try with SAT solvers

Satisfiability (SAT) solver

- A "SAT solver" is a program that automatically decides whether a propositional logic formula is satisfiable.
  - If it is satisfiable, a SAT solver will produce an example of a truth assignment that satisfies the formula.



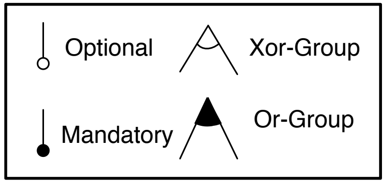
- Basic idea: since all NP-complete problems are mutually reducible:
  - Write one really good solver for NP-complete problems (in fact, get lots of people to do it. Hold competitions.)
  - Translate your NP-complete problems to that problem.

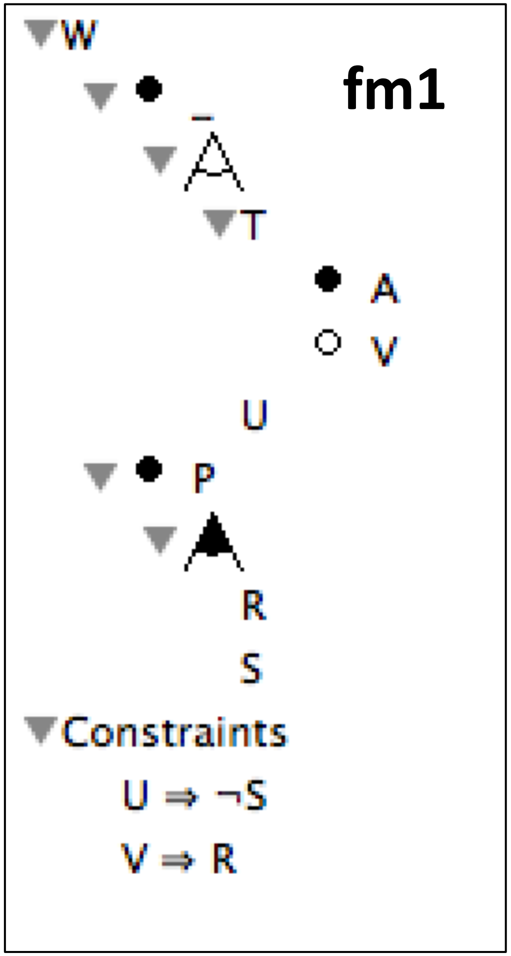
SAT solver and CNF

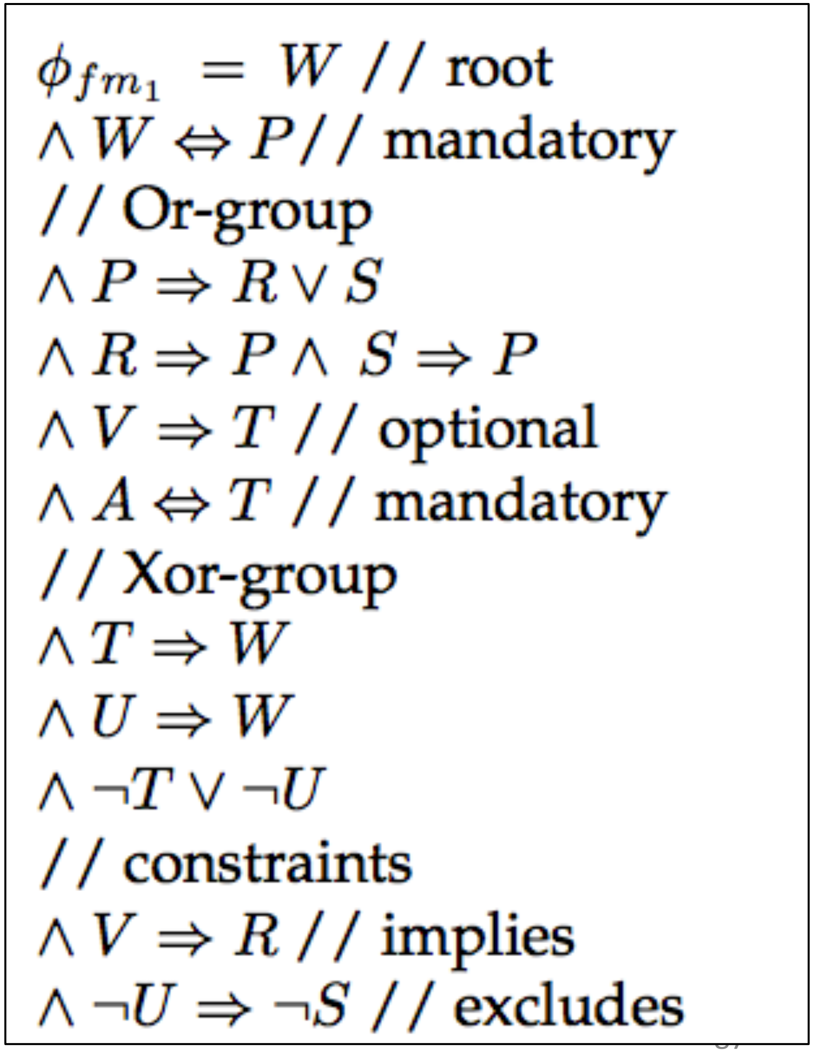
- All current fast SAT solvers work on CNF
- Terminology:
  - A literal is a propositional variable or its negation (e.g., p or ¬q).
  - A clause is a disjunction of literals (e.g., (p ⋁ ¬q ⋁ r)). Since ⋁ is associative, we can represent clauses as lists of literals.
- A formula is in conjunctive normal form (CNF) if it is a conjunction of clauses
  - e.g., (p ⋁ q ⋁ ¬r) ⋀ (¬p ⋁ s ⋁ t ⋁ ¬u)

(Boolean) Feature Models

~ Boolean formula







$\phi_{fm_1} = W$ // root

$\wedge\, W \Leftrightarrow P$// mandatory

// Or-group

$\wedge\, P \Rightarrow R \vee S$

$\wedge\, R \Rightarrow P \wedge\ S \Rightarrow P$

$\wedge\, V \Rightarrow T$ // optional

$\wedge\, A \Leftrightarrow T$ // mandatory

// Xor-group

$\wedge\, T \Rightarrow W$

$\wedge\, U \Rightarrow W$

$\wedge\, \neg T \vee \neg U$

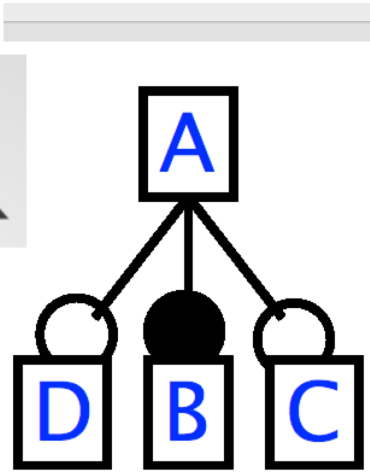
// constraints

$\wedge\, V \Rightarrow R$ // implies

$\wedge\, \neg U \Rightarrow \neg S$ // excludes

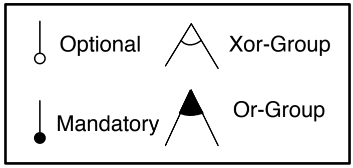
A ^
A ⇔ B ^
C => A ^
D => A

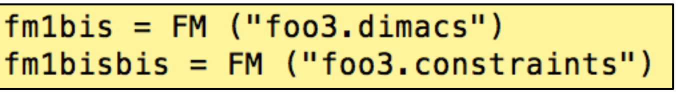
FAMiliAR



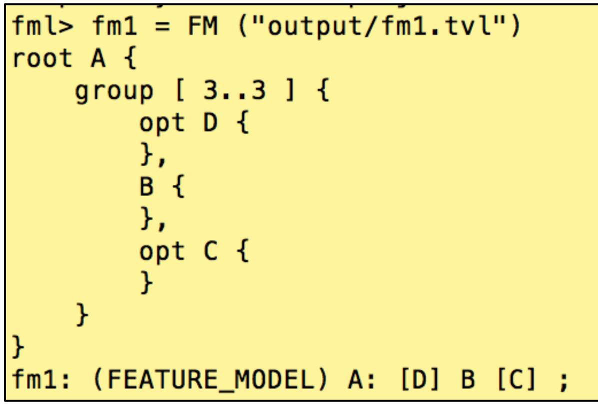
φ

FM

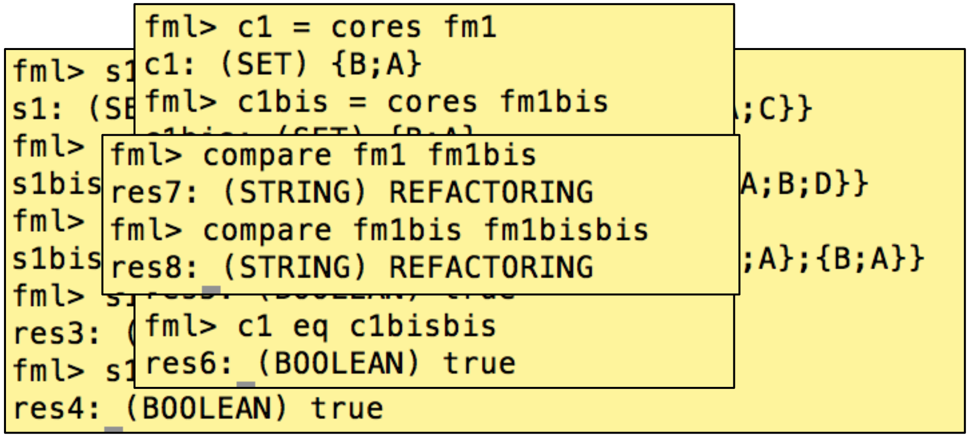




```
fm1bis = FM ("foo3.dimacs")
fm1bisbis = FM ("foo3.constraints")
```



```
fml> fm1 = FM ("output/fm1.tvl")
root A {
    group [ 3..3 ] {
        opt D {
        },
        B {
        },
        opt C {
        }
    }
}
fm1: (FEATURE_MODEL) A: [D] B [C] ;
```



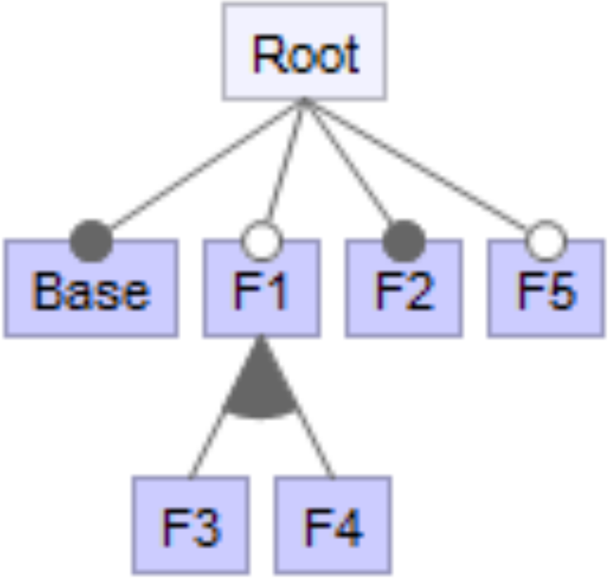
```
fml> s1
s1: (SE                                          ;C}}
fml>
s1bis                                            A;B;D}}
fml>
s1bis                                            ;A};{B;A}}
fml> s1
res3: (
fml> s1
res4: (BOOLEAN) true
```

```
fml> c1 = cores fm1
c1: (SET) {B;A}
fml> c1bis = cores fm1bis
fml> c1 eq c1bisbis
res6: (BOOLEAN) true
```

```
fml> compare fm1 fm1bis
res7: (STRING) REFACTORING
fml> compare fm1bis fm1bisbis
res8: (STRING) REFACTORING
```

Consistency

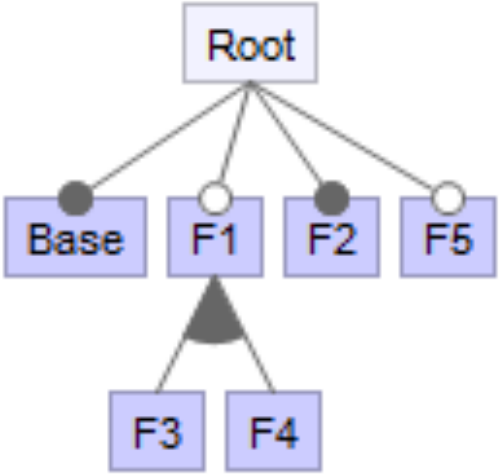
- SAT-Solver
  - SAT(FM)



F3 ∧ F4

Core and dead features

- Dead : SAT(FM ^ F)
- Core: SAT(FM ^ not(F))



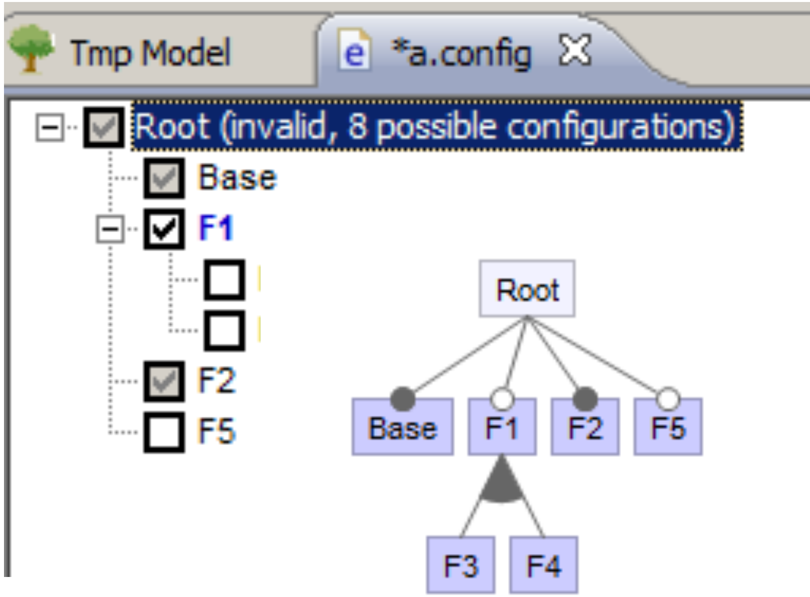
F5 ⇒ F4 ∨ Base

F3 ⇒ F2 ∧ F5

¬(F4 ∧ F2)

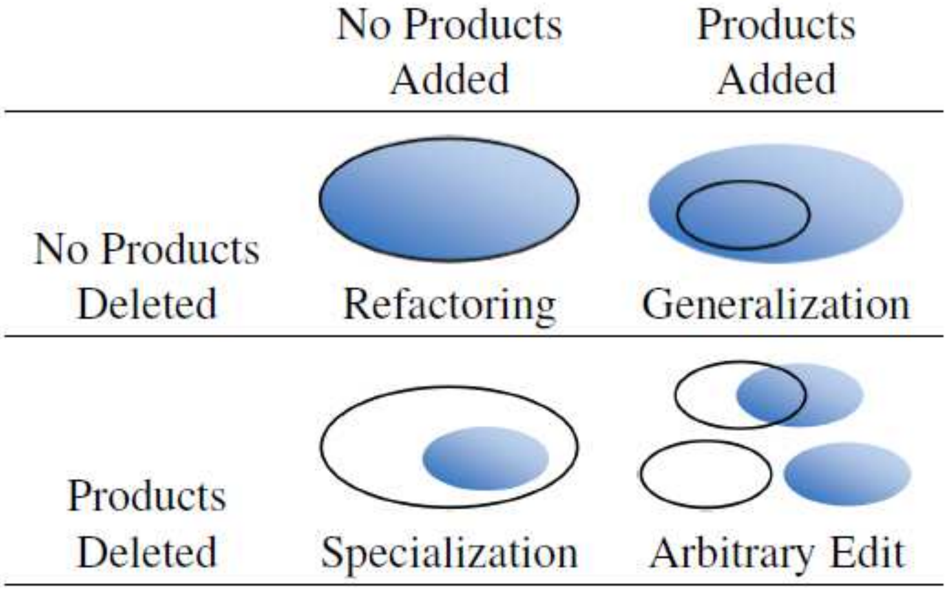
Partial configuration

- SAT(FM ^ PK ^ F)
- SAT(FM ^ PK ^ not(F))



☑ Root (invalid, 8 possible configurations)
- ☑ Base
- ☑ F1
  - □
  - □
- ☑ F2
- □ F5

Relationship between feature models



	No Products Added	Products Added
No Products Deleted	Refactoring	Generalization
Products Deleted	Specialization	Arbitrary Edit

- Refactoring
  - Tautology: (FM1 <=> FM2)
    = not SAT(not (FM1 <=> FM2))

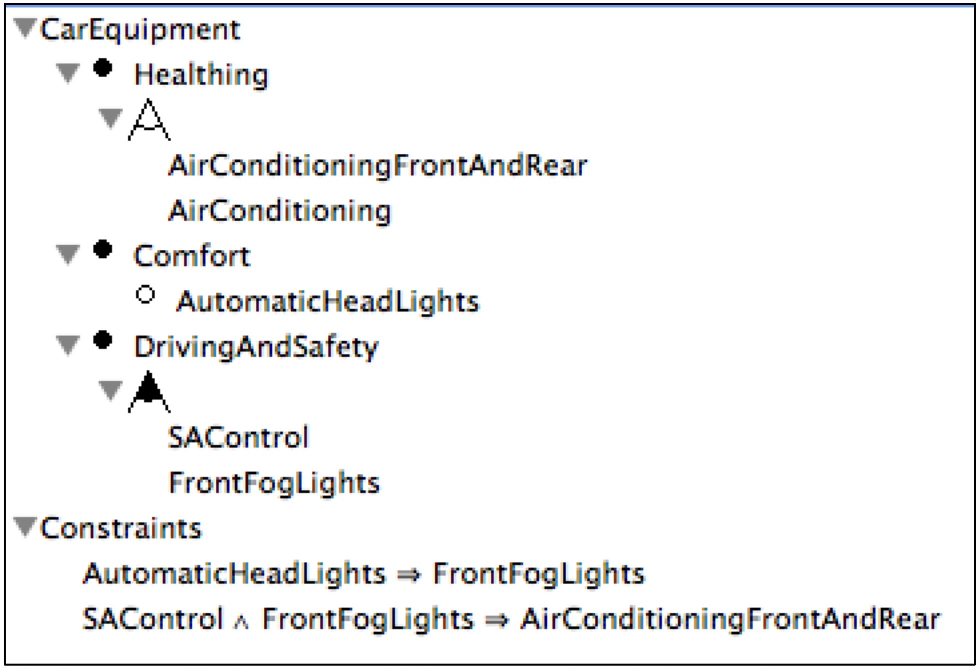
How to automate analysis of your feature models?

You can encode a feature model as a CSP problem or as an SMT problem

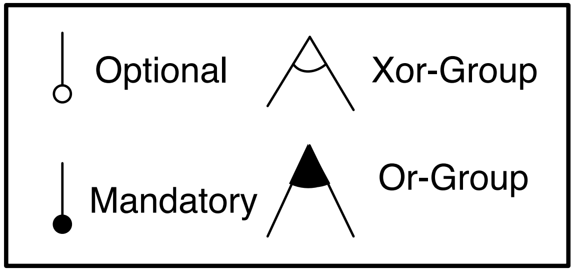
Formal semantics of a language

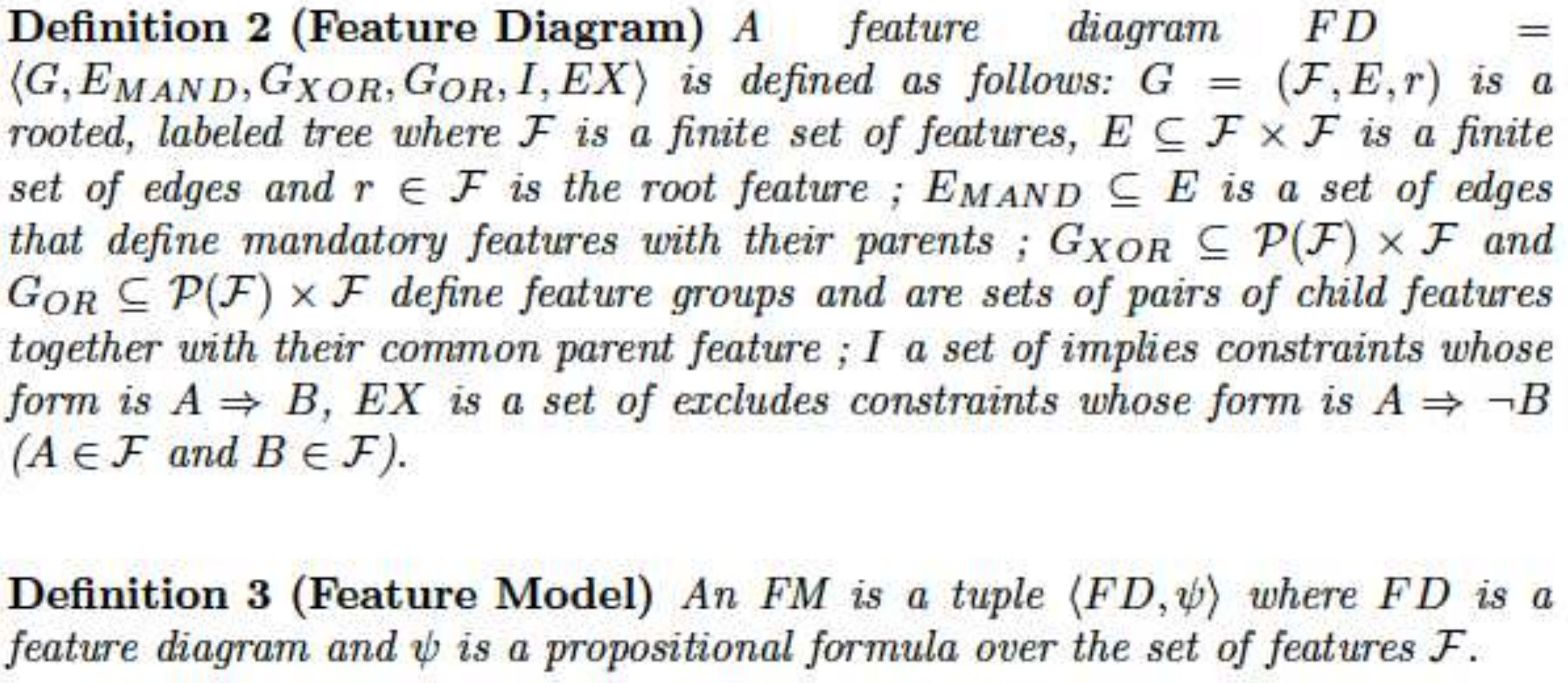
- – **formal syntax** (L) – clearcut syntactic rules defining all legal diagrams, a.k.a. syntactic domain
- – **semantic domain** (S) – a mathematical abstraction of the real-world concepts to be modelled
- – **semantic function** (M: L ⟶ S) – clearcut semantic rules defining the meaning of all legal diagrams

[Harel & Rumpe, IEEE Computer, 2004]



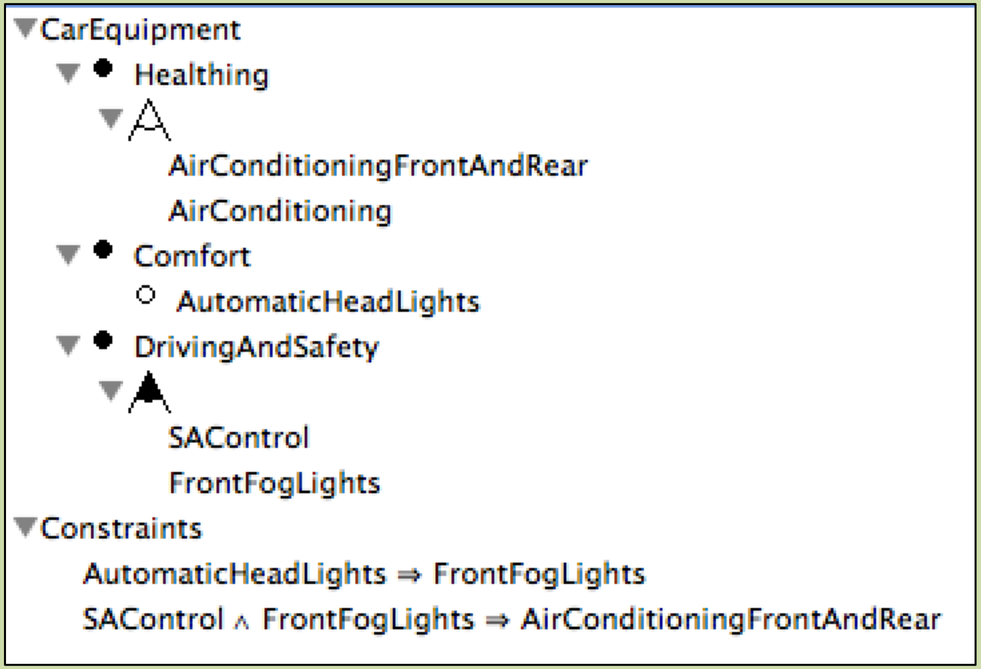
- CarEquipment
  - ● Healthing
    - ▼ A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - ▼ ▲
      - SAControl
      - FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights
  - SAControl ∧ FrontFogLights ⇒ AirConditioningFrontAndRear



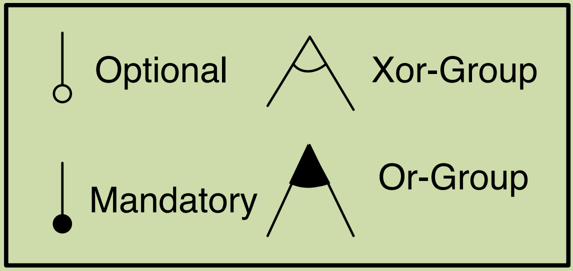


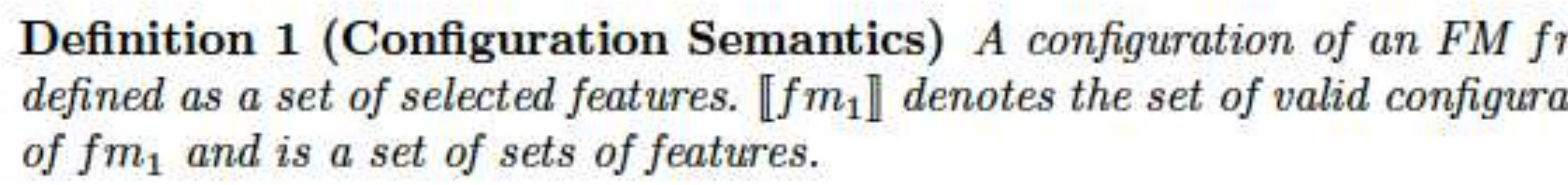
Definition 2 (Feature Diagram) *A feature diagram* $FD = \langle G, E_{MAND}, G_{XOR}, G_{OR}, I, EX \rangle$ *is defined as follows:* $G = (\mathcal{F}, E, r)$ *is a rooted, labeled tree where* $\mathcal{F}$ *is a finite set of features,* $E \subseteq \mathcal{F} \times \mathcal{F}$ *is a finite set of edges and* $r \in \mathcal{F}$ *is the root feature ;* $E_{MAND} \subseteq E$ *is a set of edges that define mandatory features with their parents ;* $G_{XOR} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{F}$ *and* $G_{OR} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{F}$ *define feature groups and are sets of pairs of child features together with their common parent feature ;* I *a set of implies constraints whose form is* $A \Rightarrow B$, EX *is a set of excludes constraints whose form is* $A \Rightarrow \neg B$ $(A \in \mathcal{F}$ *and* $B \in \mathcal{F})$.

Definition 3 (Feature Model) *An FM is a tuple* $\langle FD, \psi \rangle$ *where* FD *is a feature diagram and* ψ *is a propositional formula over the set of features* $\mathcal{F}$.



- CarEquipment
  - ● Healthing
    - A
      - AirConditioningFrontAndRear
      - AirConditioning
  - ● Comfort
    - ○ AutomaticHeadLights
  - ● DrivingAndSafety
    - ▲
      - SAControl
      - FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights
  - SAControl ∧ FrontFogLights ⇒ AirConditioningFrontAndRear





Definition 1 (Configuration Semantics) *A configuration of an FM fr defined as a set of selected features.* $[\![fm_1]\!]$ *denotes the set of valid configura of* fm_1 *and is a set of sets of features.*

{CarEquipment, Comfort, DrivingAndSafety, Healthing} X

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

Quizz

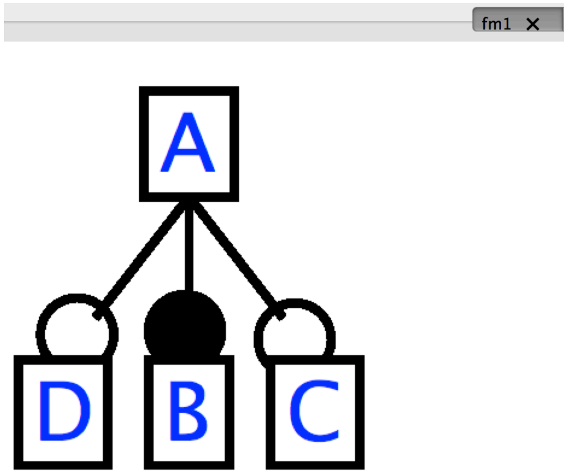
1) Give two feature models with the same configuration semantics but with different syntax

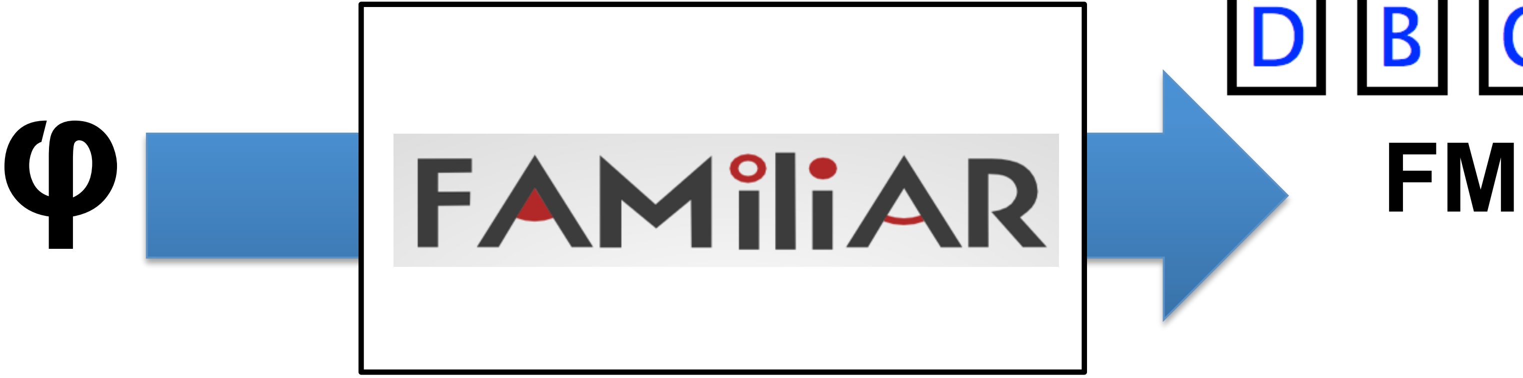
2) Does it matter ?

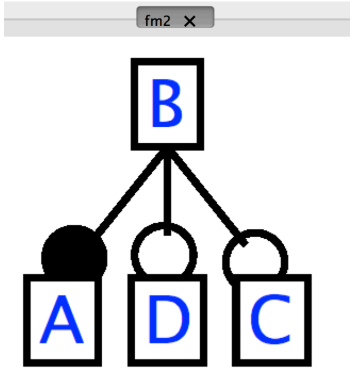
A ^
A ⇔ B ^
C => A ^
D => A

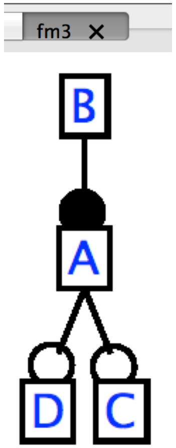
Feature Model Synthesis Problem

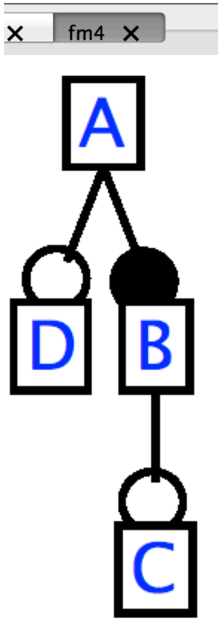
[Czarnecki et al., SPLC'07]
[She et al., ICSE'11]
[Andersen et al., SPLC'12]







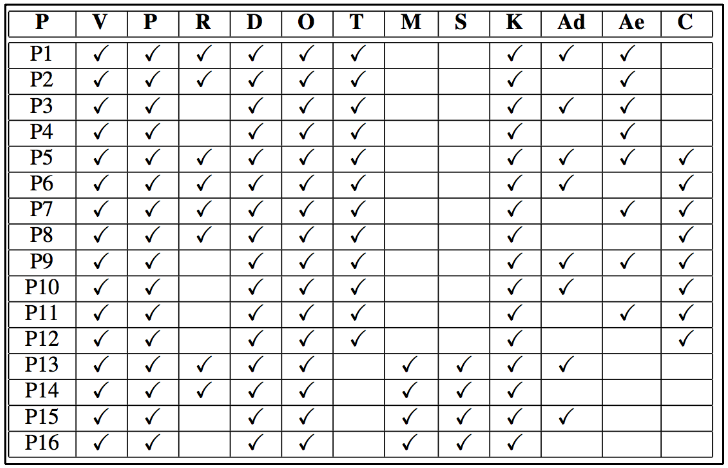




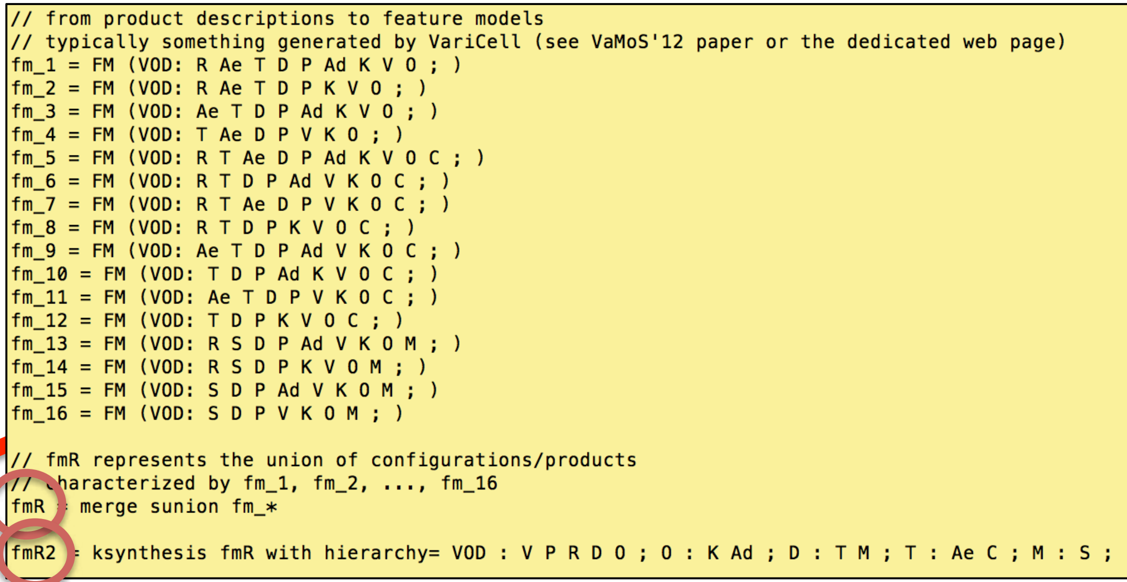
```
fm2 = FM (B : A [C] [D] ; )
fm3 = FM (B : A ; A : [C] [D] ; )
fm4 = FM (A : B [D] ; B : [C] ; )
fm5 = FM (A : B [C] ; B : [D] ; )

b12 = compare fm1 fm2
b13 = compare fm1 fm3
b14 = compare fm1 fm4
b15 = compare fm1 fm5
assert (b12 eq REFACTORING)
```

#1 Reverse Engineering Scenarios

- [Haslinger et al., WCRE’11], [Acher et al., VaMoS’12]

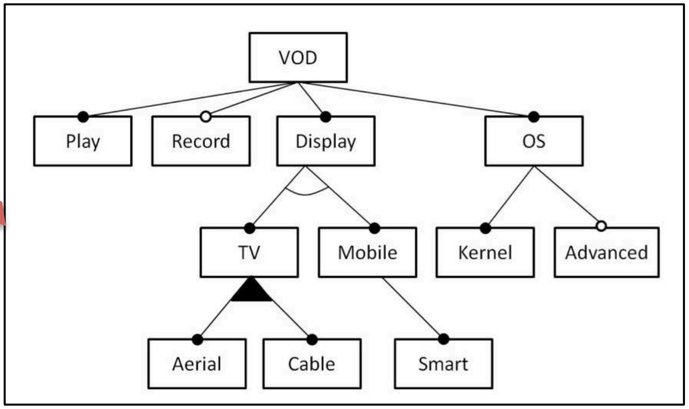


P	V	P	R	D	O	T	M	S	K	Ad	Ae	C
P1	✓	✓	✓	✓	✓	✓			✓	✓	✓	
P2	✓	✓	✓	✓	✓	✓			✓		✓	
P3	✓	✓		✓	✓	✓			✓	✓	✓	
P4	✓	✓		✓	✓	✓			✓		✓	
P5	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
P6	✓	✓	✓	✓	✓	✓			✓	✓		✓
P7	✓	✓	✓	✓	✓	✓			✓		✓	✓
P8	✓	✓	✓	✓	✓	✓			✓			✓
P9	✓	✓		✓	✓	✓			✓	✓	✓	✓
P10	✓	✓		✓	✓	✓			✓	✓		✓
P11	✓	✓		✓	✓	✓			✓		✓	✓
P12	✓	✓		✓	✓	✓			✓			✓
P13	✓	✓	✓	✓	✓		✓	✓	✓	✓		
P14	✓	✓	✓	✓	✓		✓	✓	✓			
P15	✓	✓		✓	✓		✓	✓	✓	✓		
P16	✓	✓		✓	✓		✓	✓	✓			



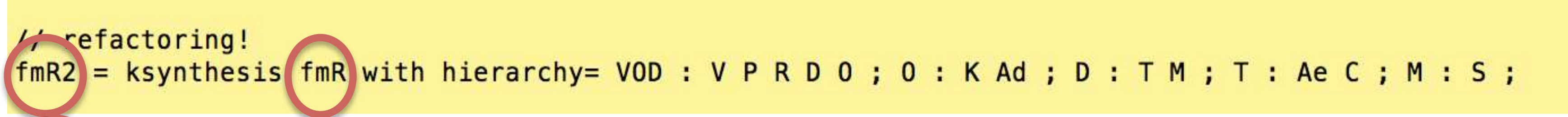
```
// from product descriptions to feature models
// typically something generated by VariCell (see VaMoS'12 paper or the dedicated web page)
fm_1 = FM (VOD: R Ae T D P Ad K V O ; )
fm_2 = FM (VOD: R Ae T D P K V O ; )
fm_3 = FM (VOD: Ae T D P Ad K V O ; )
fm_4 = FM (VOD: T Ae D P V K O ; )
fm_5 = FM (VOD: R T Ae D P Ad K V O C ; )
fm_6 = FM (VOD: R T D P Ad V K O C ; )
fm_7 = FM (VOD: R T Ae D P V K O C ; )
fm_8 = FM (VOD: R T D P K V O C ; )
fm_9 = FM (VOD: Ae T D P Ad V K O C ; )
fm_10 = FM (VOD: T D P Ad K V O C ; )
fm_11 = FM (VOD: Ae T D P V K O C ; )
fm_12 = FM (VOD: T D P K V O C ; )
fm_13 = FM (VOD: R S D P Ad V K O M ; )
fm_14 = FM (VOD: R S D P K V O M ; )
fm_15 = FM (VOD: S D P Ad V K O M ; )
fm_16 = FM (VOD: S D P V K O M ; )

// fmR represents the union of configurations/products
// characterized by fm_1, fm_2, ..., fm_16
fmR = merge sunion fm_*

fmR2 = ksynthesis fmR with hierarchy= VOD : V P R D O ; O : K Ad ; D : T M ; T : Ae C ; M : S ;
```

φ

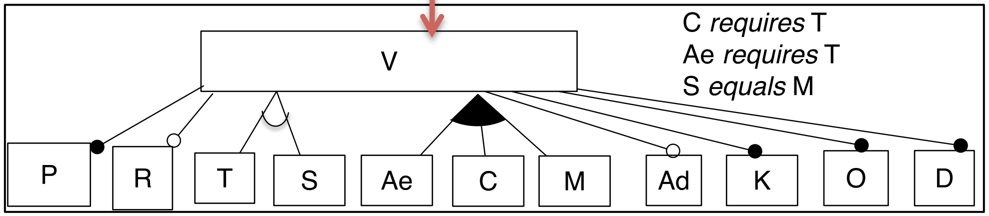


#2 Refactoring

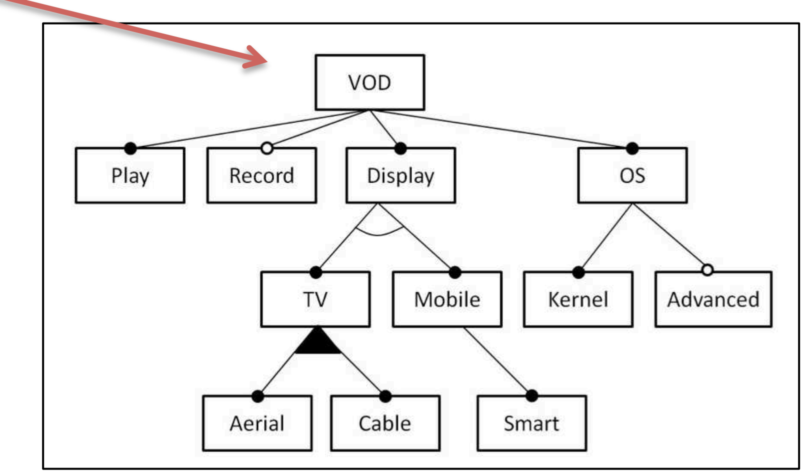
- [Alves et al., GPCE’06], [Thuem et al., ICSE’09]



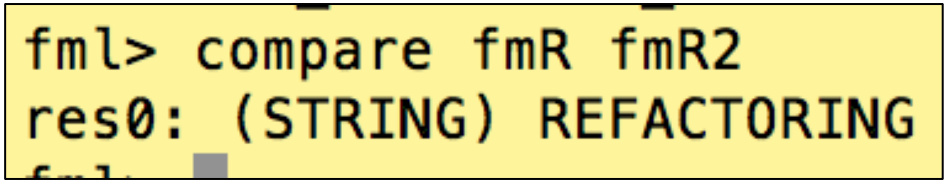
```
// refactoring!
fmR2 = ksynthesis fmR with hierarchy= VOD : V P R D O ; O : K Ad ; D : T M ; T : Ae C ; M : S ;
```



C *requires* T
Ae *requires* T
S *equals* M



φ



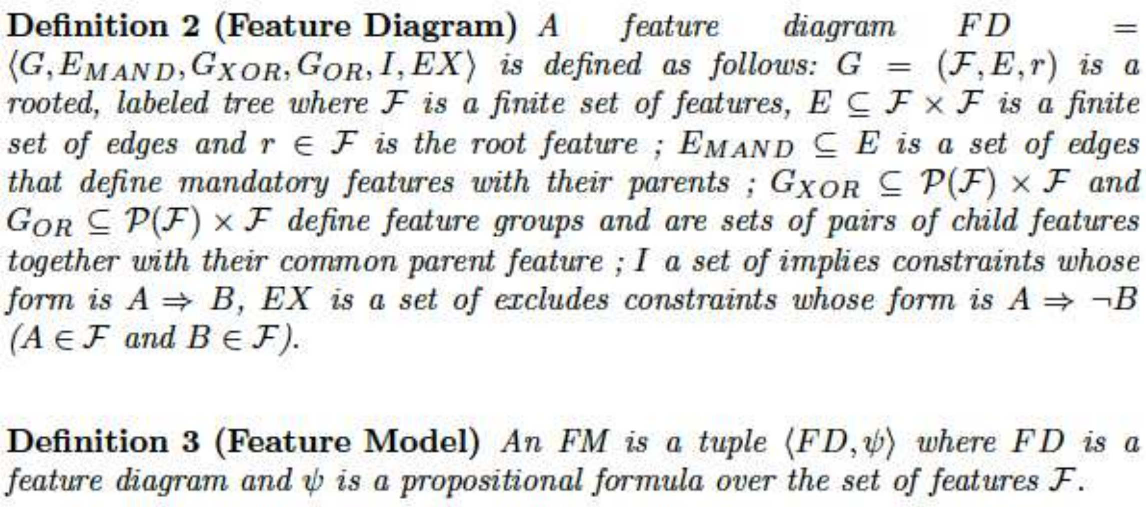
```
fml> compare fmR fmR2
res0: (STRING) REFACTORING
fml>
```

Feature Model SemanticS

- As configuration semantics is not sufficient…

- **Ontological** semantics
  - Hierarchy
  - And feature groups

Quizz (back to Feature Model)



Definition 2 (Feature Diagram) *A feature diagram $FD = \langle G, E_{MAND}, G_{XOR}, G_{OR}, I, EX \rangle$ is defined as follows: $G = (\mathcal{F}, E, r)$ is a rooted, labeled tree where $\mathcal{F}$ is a finite set of features, $E \subseteq \mathcal{F} \times \mathcal{F}$ is a finite set of edges and $r \in \mathcal{F}$ is the root feature ; $E_{MAND} \subseteq E$ is a set of edges that define mandatory features with their parents ; $G_{XOR} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{F}$ and $G_{OR} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{F}$ define feature groups and are sets of pairs of child features together with their common parent feature ; I a set of implies constraints whose form is $A \Rightarrow B$, EX is a set of excludes constraints whose form is $A \Rightarrow \neg B$ ($A \in \mathcal{F}$ and $B \in \mathcal{F}$).*

Definition 3 (Feature Model) *An FM is a tuple $\langle FD, \psi \rangle$ where FD is a feature diagram and ψ is a propositional formula over the set of features $\mathcal{F}$.*

Given a set of configurations *s*, can we always characterize *s* with a feature diagram *fd* ?

ie [[fd]] = s

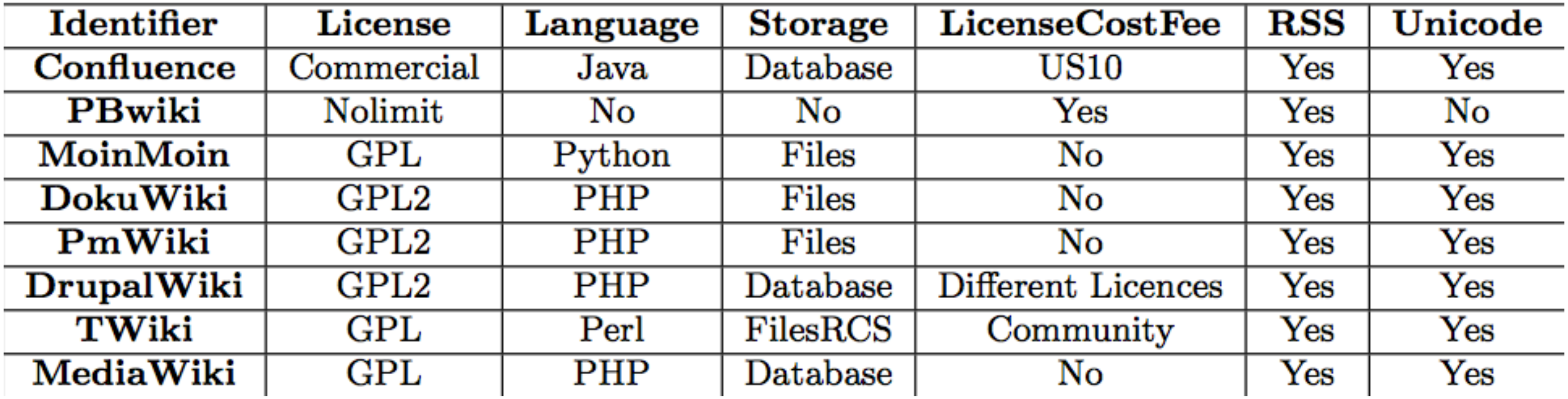
In other words: is the formalism of feature diagram expressive enough wrt Boolean logic?

Feature Diagram ?

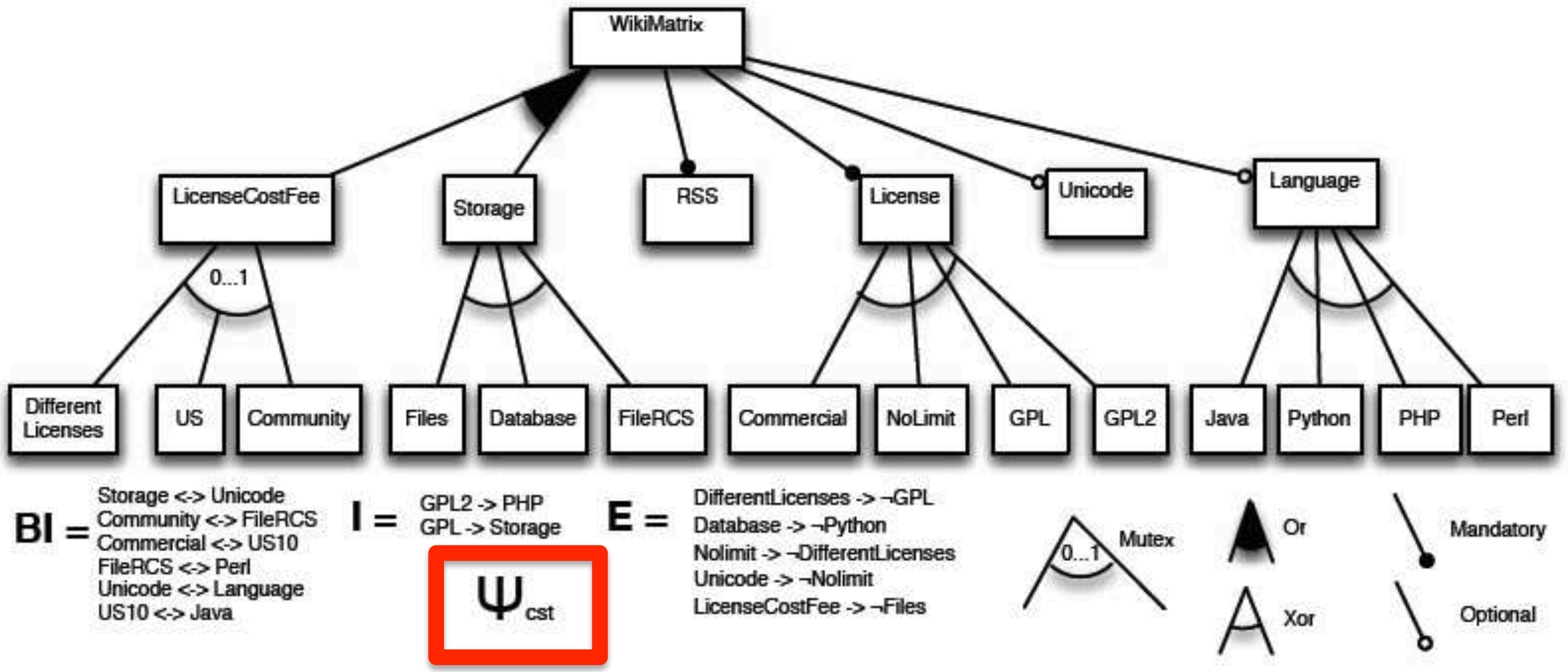
s = {{A},
{A,C,B},
{B,A},
{C,D,A},
{D,A},
{A,D,B},
{A,C}
}

```
fm1 = FM (A : [B] [C] [D] ^
// B, C and D are optional features of A

((B & C) -> !D)

)
```

Feature Diagram ?



Identifier	License	Language	Storage	LicenseCostFee	RSS	Unicode
Confluence	Commercial	Java	Database	US10	Yes	Yes
PBwiki	Nolimit	No	No	Yes	Yes	No
MoinMoin	GPL	Python	Files	No	Yes	Yes
DokuWiki	GPL2	PHP	Files	No	Yes	Yes
PmWiki	GPL2	PHP	Files	No	Yes	Yes
DrupalWiki	GPL2	PHP	Database	Different Licences	Yes	Yes
TWiki	GPL	Perl	FilesRCS	Community	Yes	Yes
MediaWiki	GPL	PHP	Database	No	Yes	Yes



BI = Storage <-> Unicode
Community <-> FileRCS
Commercial <-> US10
FileRCS <-> Perl
Unicode <-> Language
US10 <-> Java

I = GPL2 -> PHP
GPL -> Storage

Ψ_{cst}

E = DifferentLicenses -> ¬GPL
Database -> ¬Python
Nolimit -> ¬DifferentLicenses
Unicode -> ¬Nolimit
LicenseCostFee -> ¬Files

Feature Model (bis)

s = {{A}, {B}}

fd = ?

Feature Model: Key Insights

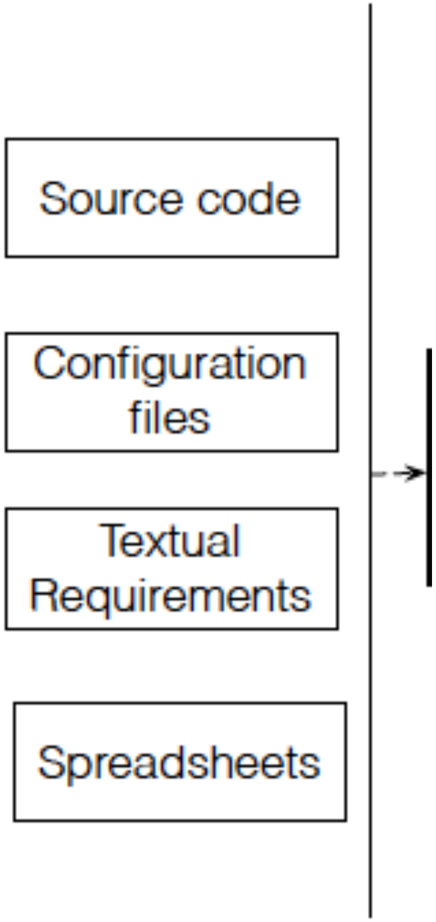
- Semantics
  - Configuration and ontological

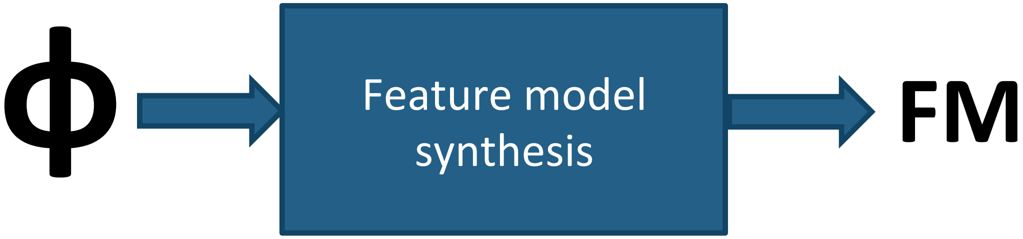
- Syntax
  - Feature diagram vs Feature Model
  - Feature diagram not expressively complete

- Feature models are a (syntactical) view of a propositional formula

From logics to variability model (there and back again) Czarnecki et al. SPLC'07

Bécan et al. Breathing Ontological Knowledge Into Feature Model Synthesis. In Empirical Software Engineering (ESE)





Feature model synthesis problem

Input: **ϕ**, a propositional formula representing the **dependencies** over a set of features *F*.

Output: a maximal feature model with a **sound configuration semantics**

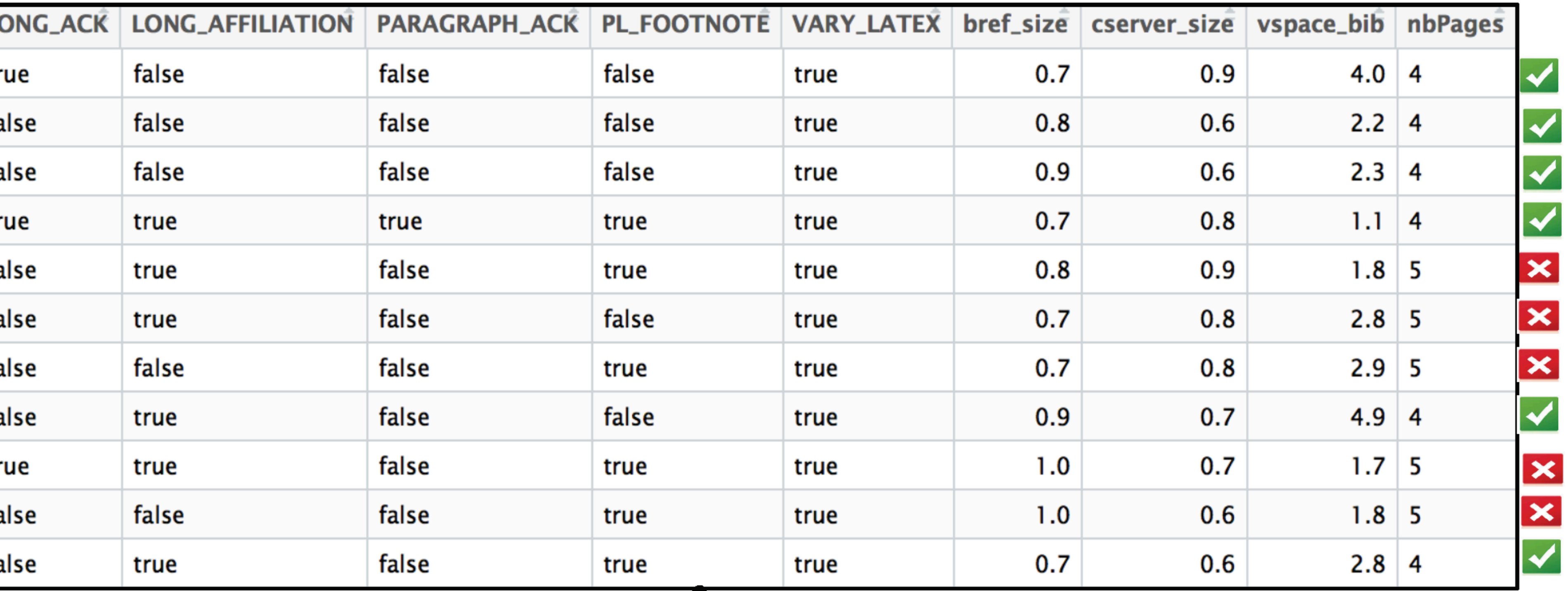
(end of second part)

Software Variability and Artificial Intelligence

- Very large variability spaces
- **AI#2 Statistical, supervised machine learning** to (out of a sample):
  - Understand the configuration space
  - Find the best configuration
  - Specialize the configuration space (e.g., by capturing constraints)
  - In a cost-effective way

AI#2 Statistical, supervised machine learning (classification problem)

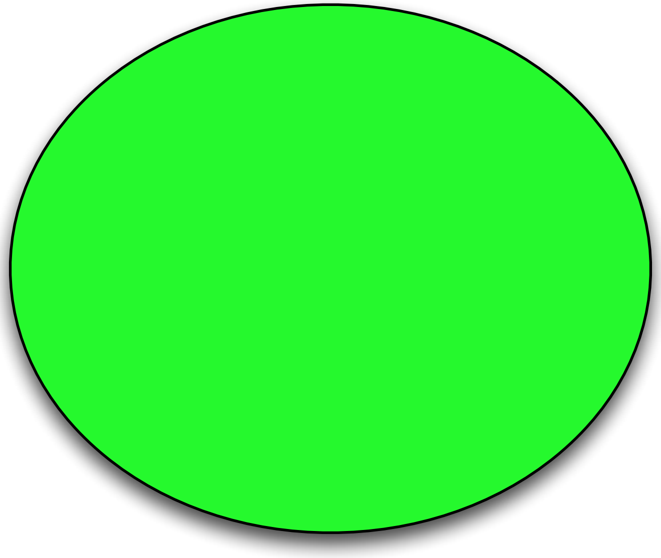
Paper variants building and measurements



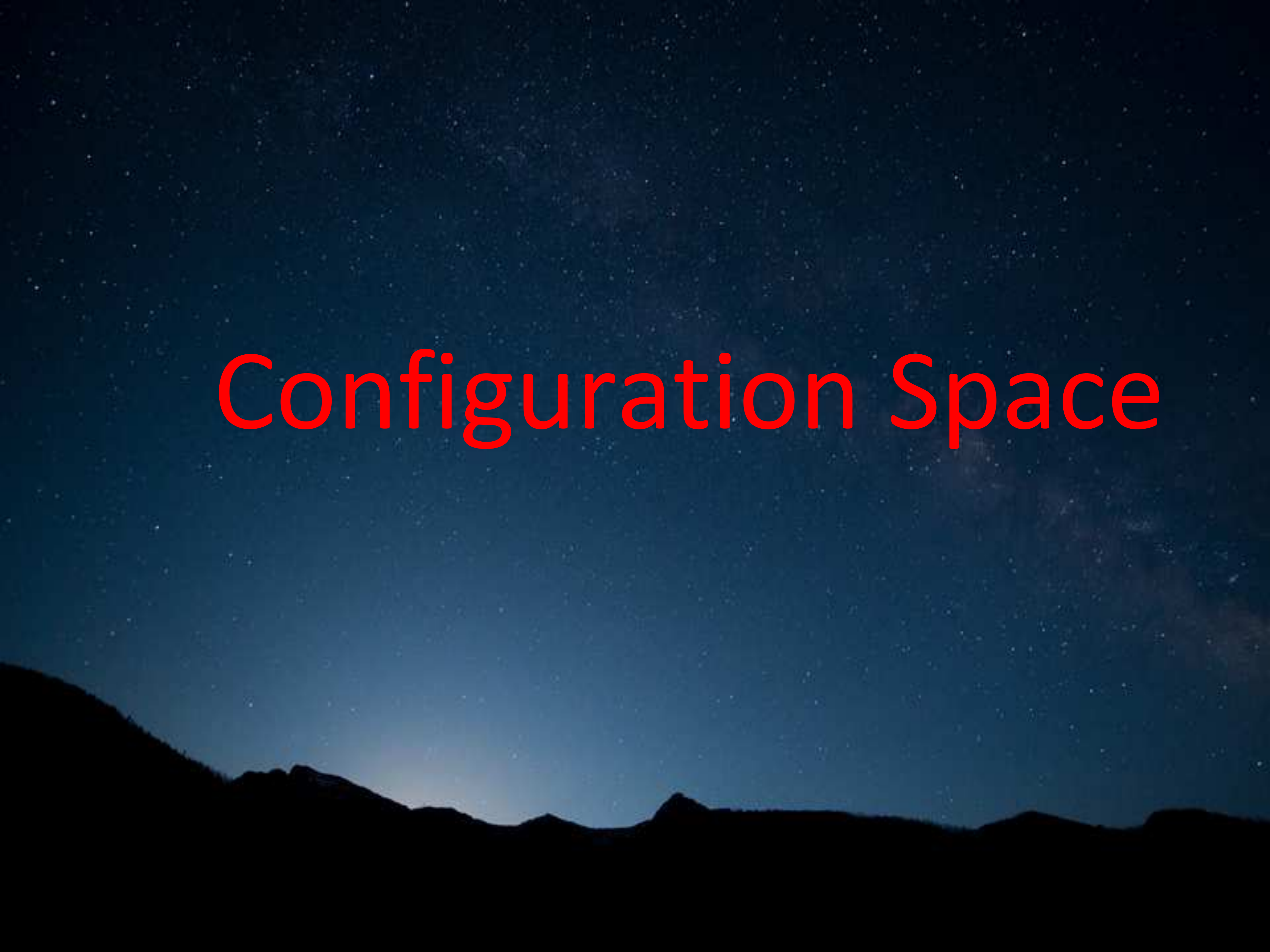
ONG_ACK	LONG_AFFILIATION	PARAGRAPH_ACK	PL_FOOTNOTE	VARY_LATEX	bref_size	cserver_size	vspace_bib	nbPages	
ue	false	false	false	true	0.7	0.9	4.0	4	✓
alse	false	false	false	true	0.8	0.6	2.2	4	✓
alse	false	false	false	true	0.9	0.6	2.3	4	✓
ue	true	true	true	true	0.7	0.8	1.1	4	✓
alse	true	false	true	true	0.8	0.9	1.8	5	✗
alse	true	false	false	true	0.7	0.8	2.8	5	✗
alse	false	false	true	true	0.7	0.8	2.9	5	✗
alse	true	false	false	true	0.9	0.7	4.9	4	✓
ue	true	false	true	true	1.0	0.7	1.7	5	✗
alse	false	false	true	true	1.0	0.6	1.8	5	✗
alse	true	false	true	true	0.7	0.6	2.8	4	✓

Configuration space

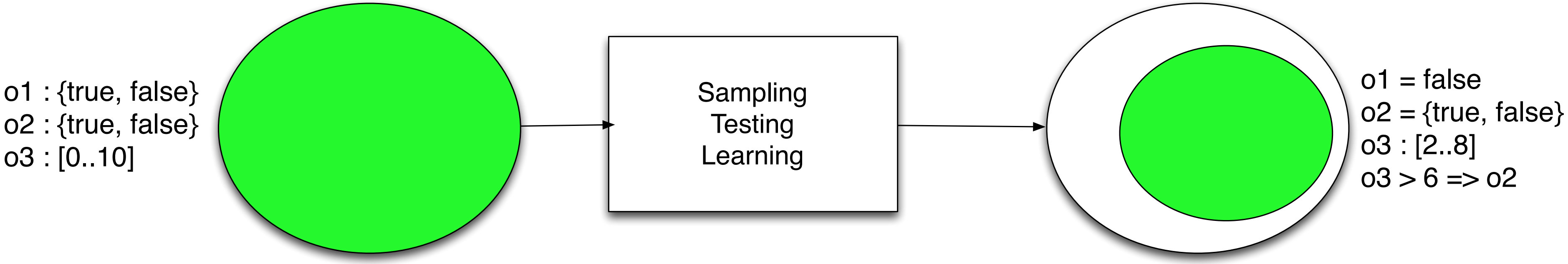
o1 : {true, false}
o2 : {true, false}
o3 : [0..10]



Configuration Space



How to ensure that all variants compile? boot? are secured?



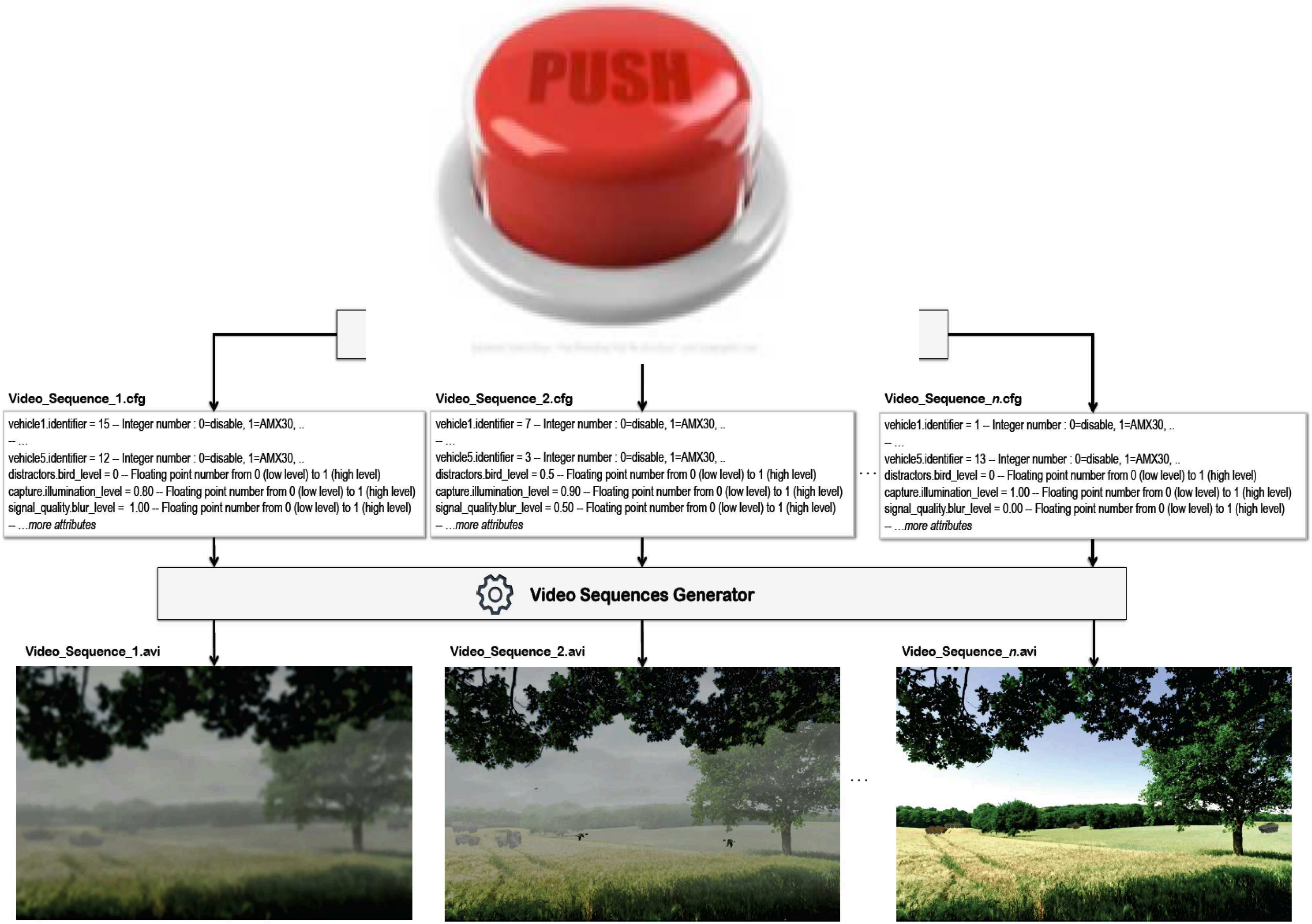
o1 : {true, false}
o2 : {true, false}
o3 : [0..10]

Sampling
Testing
Learning

o1 = false
o2 = {true, false}
o3 : [2..8]
o3 > 6 => o2

Enormous configurations space eg Linux has 15000+ options, tri-state values {y, n, m}; you cannot test all variants

Learning over a small sample



Video_Sequence_1.cfg

```
vehicle1.identifier = 15 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 12 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.80 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level =  1.00 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

Video_Sequence_2.cfg

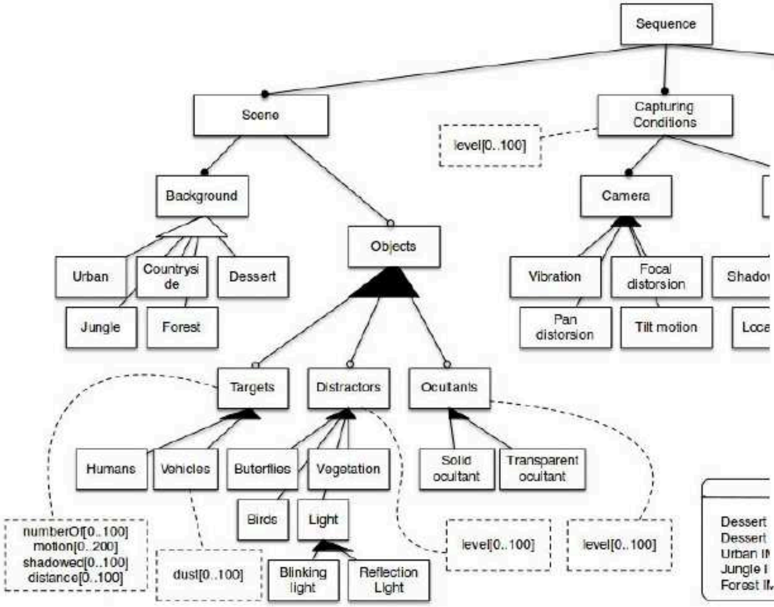
```
vehicle1.identifier = 7 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 3 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0.5 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.90 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.50 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

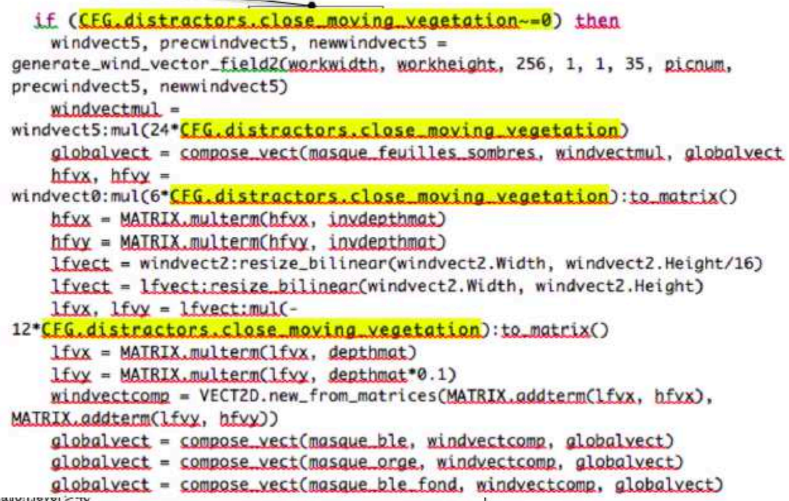
Video_Sequence_*n*.cfg

```
vehicle1.identifier = 1 – Integer number : 0=disable, 1=AMX30, ..
– ...
vehicle5.identifier = 13 – Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 – Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 1.00 – Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.00 – Floating point number from 0 (low level) to 1 (high level)
– ...more attributes
```

José A. Galindo, Mauricio Alférez, Mathieu Acher, Benoit Baudry, David Benavides:
A variability-based testing approach for synthesizing video sequences. ISSTA 2014:

Industrial video generator





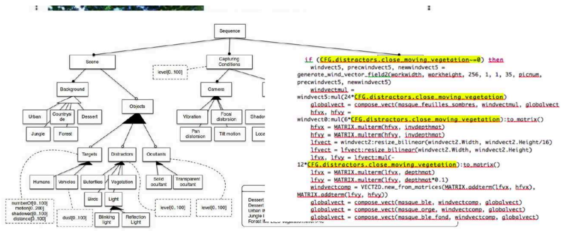
```
  if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
    hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(-
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfvy, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
```

(Lua code)

Large FM (80 features)

Features are mainly described as float-values

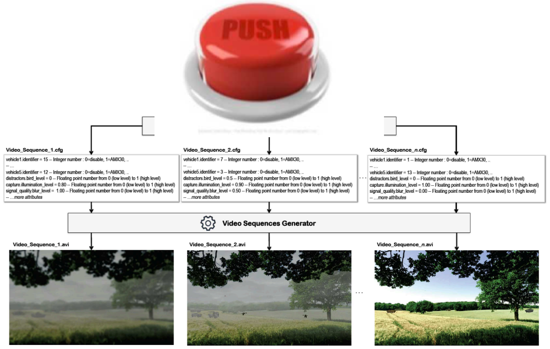
José A. Galindo, Mauricio Alférez, Mathieu Acher, Benoit Baudry, David Benavides:
A variability-based testing approach for synthesizing video sequences. ISSTA 2014:



```
  if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
    hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(-
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfvy, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
```

(Lua code)

Large FM (80 features)
Features are mainly described as float-values



Video_Sequence_1.cfg

```
vehicle1.identifier = 15 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 12 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.80 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level =  1.00 -- Floating point number from 0 (low level) to 1 (high level)
-- ...more attributes
```

Video_Sequence_2.cfg

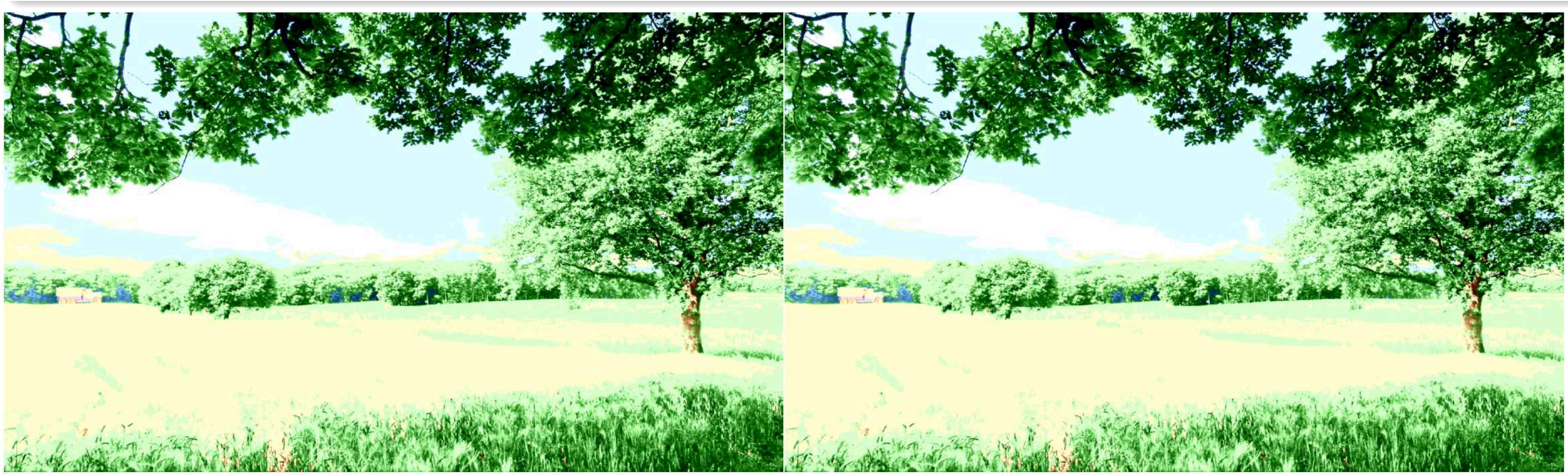
```
vehicle1.identifier = 7 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 3 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0.5 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.90 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.50 -- Floating point number from 0 (low level) to 1 (high level)
-- ...more attributes
```

Video_Sequence_n.cfg

```
vehicle1.identifier = 1 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 13 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 1.00 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.00 -- Floating point number from 0 (low level) to 1 (high level)
-- ...more attributes
```

Problem: some video variants are non-acceptable despite specification of numerous constraints

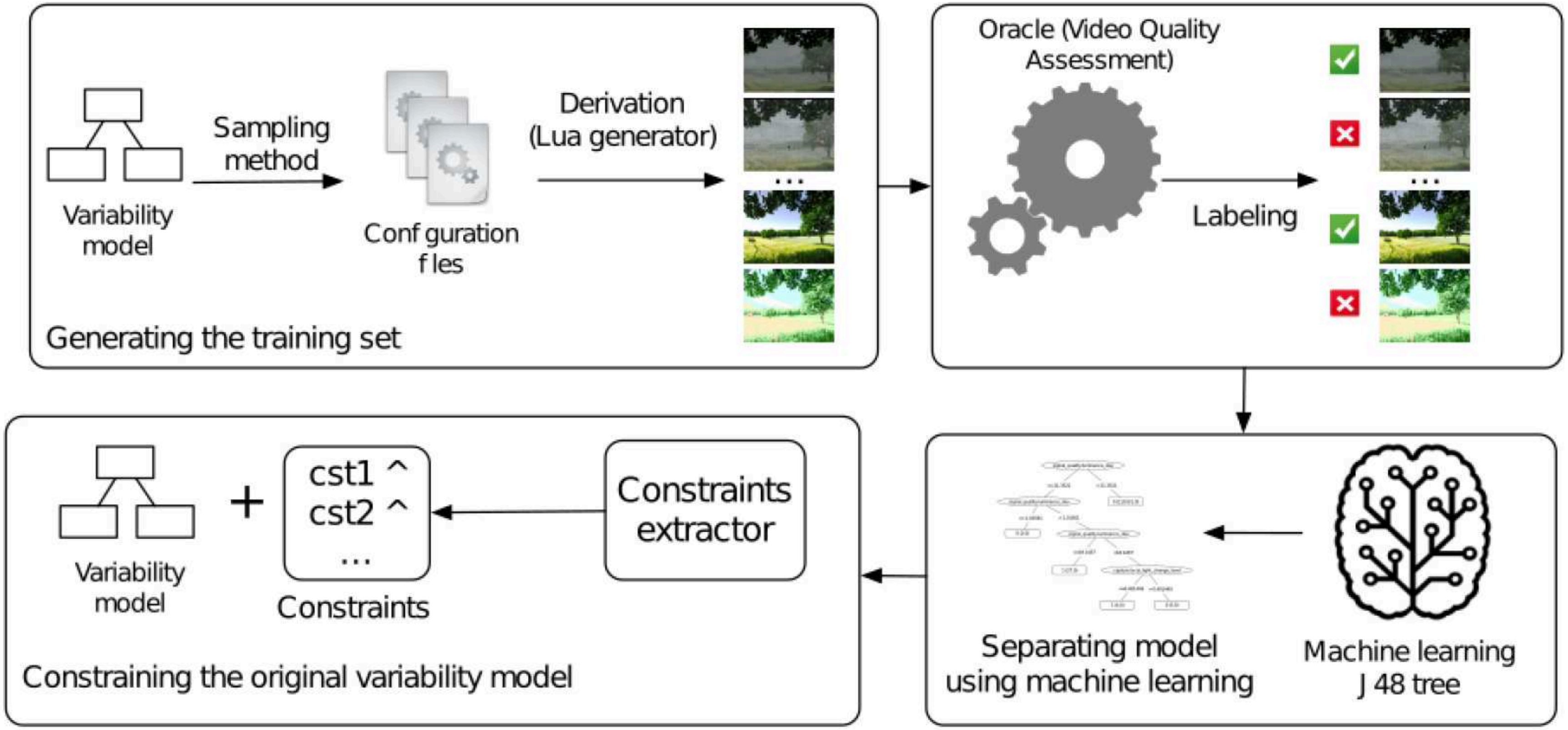
(note: synthesizing a variant takes 30 minutes)



Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines, SPLC’16

Problem: some video variants are non-acceptable despite specification of numerous constraints

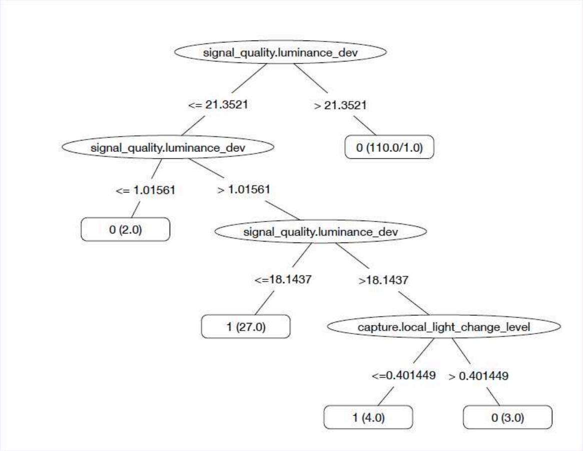
(note: synthesizing a variant takes 30 minutes)


Variability model
Sampling method
Configuration files
Derivation (Lua generator)
...
Generating the training set
Oracle (Video Quality Assessment)
Labeling
...
Separating model using machine learning
Machine learning J 48 tree
Variability model
+
cst1 ^
cst2 ^
...
Constraints
Constraints extractor
Constraining the original variability model

Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines, SPLC’16

Results (training set: 500 video variants; validation set: 4000 variants)

Decision Tree



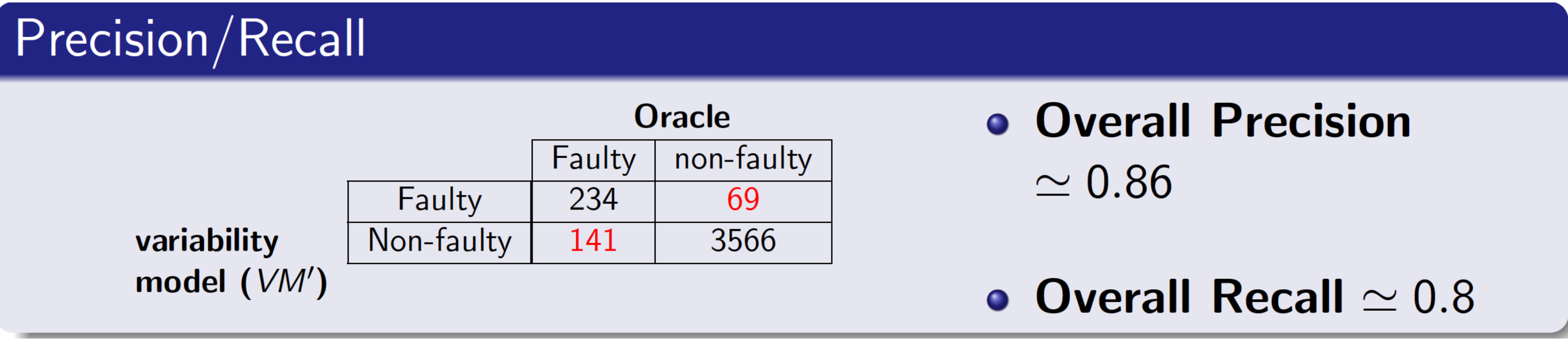
Constraints

!(signal_quality.luminance_dev > 1.01561 && signal_quality.luminance_dev <= 18.1437)

!(signal_quality.luminance_dev <= 21.3521 && signal_quality.luminance_dev > 18.1437 && capture.local_light_change_level <= 0.481449)

Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines, SPLC'16

Results (training set: 500 video variants; validation set: 4000 variants)

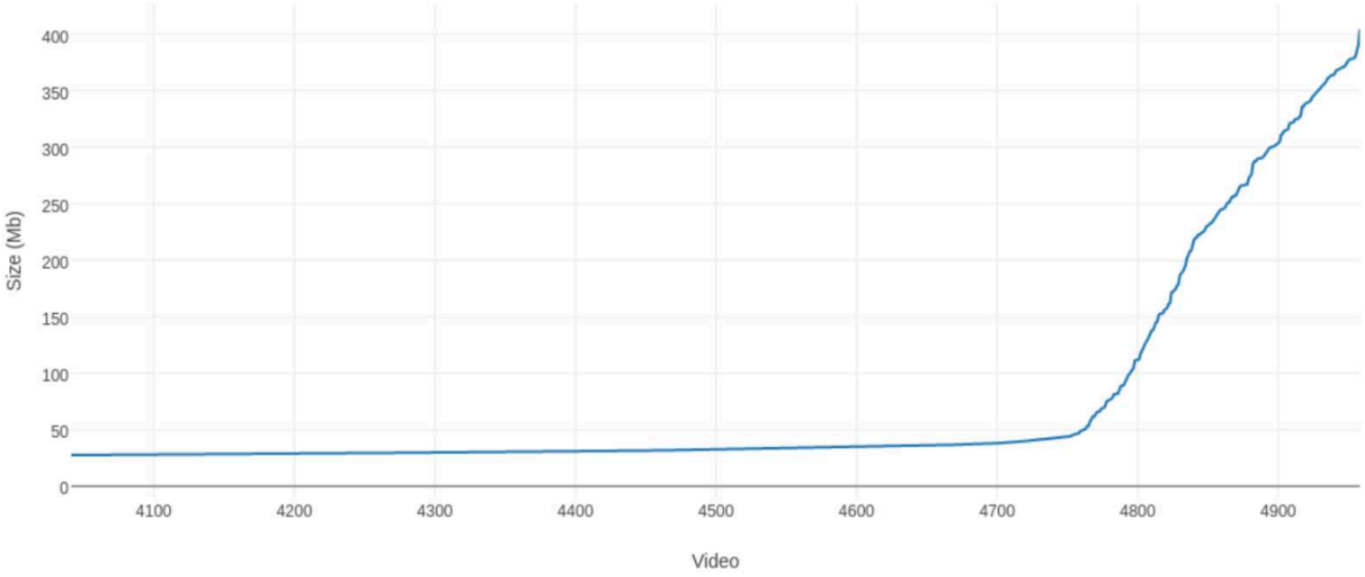


Precision/Recall

		Oracle	
		Faulty	non-faulty
variability model (VM')	Faulty	234	69
	Non-faulty	141	3566

- **Overall Precision** $\simeq 0.86$
- **Overall Recall** $\simeq 0.8$

Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines, SPLC'16



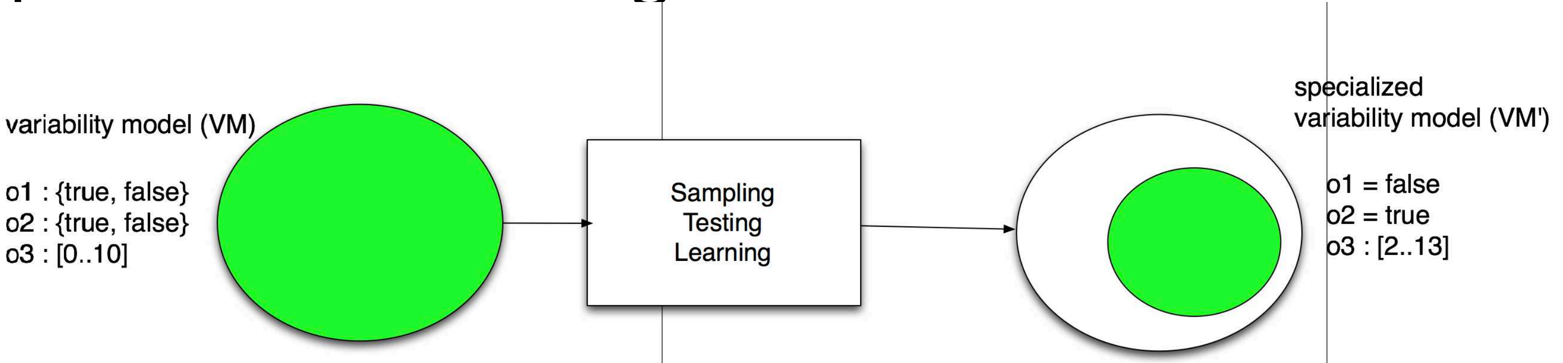
Constraints

!(signal_quality.dynamic_noise_level $>$ 0.171472 &&
signal_quality.compression_artefact_level $<=$ 0.180349)
!(signal_quality.dynamic_noise_level $>$ 0.171472)

Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines, SPLC'16

Generalization of learning-based specialization

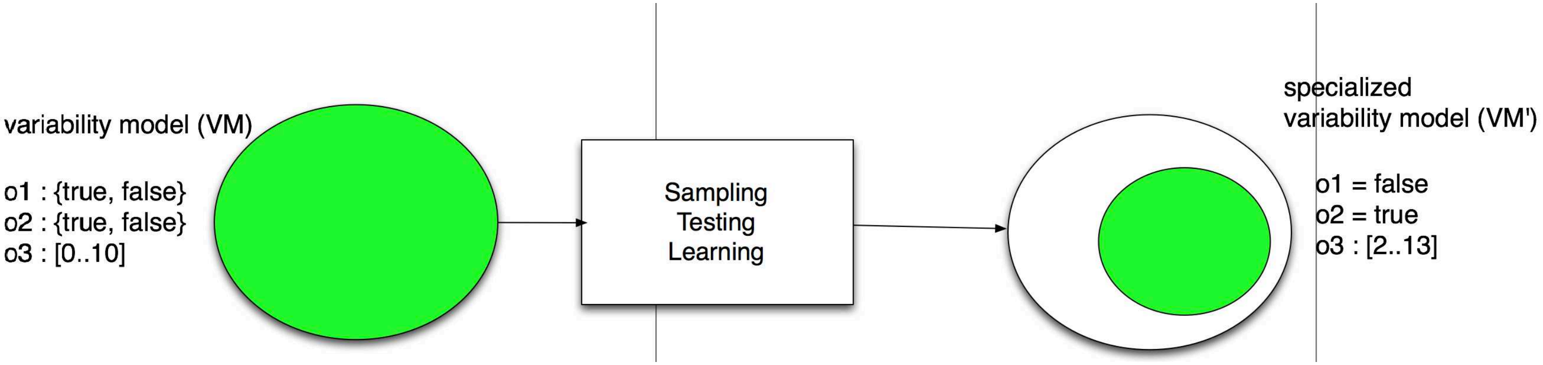
- Configurations have a label/class
  - true/false (video gen) or nbPages={4,5} (VaryLaTeX); without any discussion a classification problem
- However there are scenarios in which the acceptability is defined in terms of performance
- Specialization is a classification problem; we boil down to this problem through a threshold over a quantitative value eg execution time < 1s



Automated Specialization

- Problem: configuring a system is hard
  - combinatorial explosion
  - functional concerns and performance qualities
  - users want to have a maximum of flexibility and perform no configuration error
- Configuration « envelope »
  - Safety (beware of being too permissive)
  - Flexibility (beware of being too restrictive)
- Solution: all option values (and combinations thereof) presented to users should satisfy an “objective”

```
x264 --quiet
     --no-progress
     --no-asm
     --rc-lookahead 60
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```



variability model (VM)

o1 : {true, false}
o2 : {true, false}
o3 : [0..10]

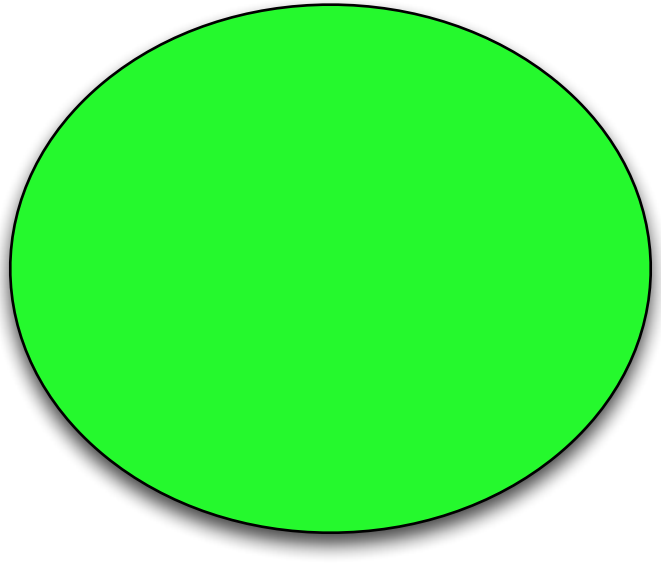
Sampling
Testing
Learning

specialized
variability model (VM')

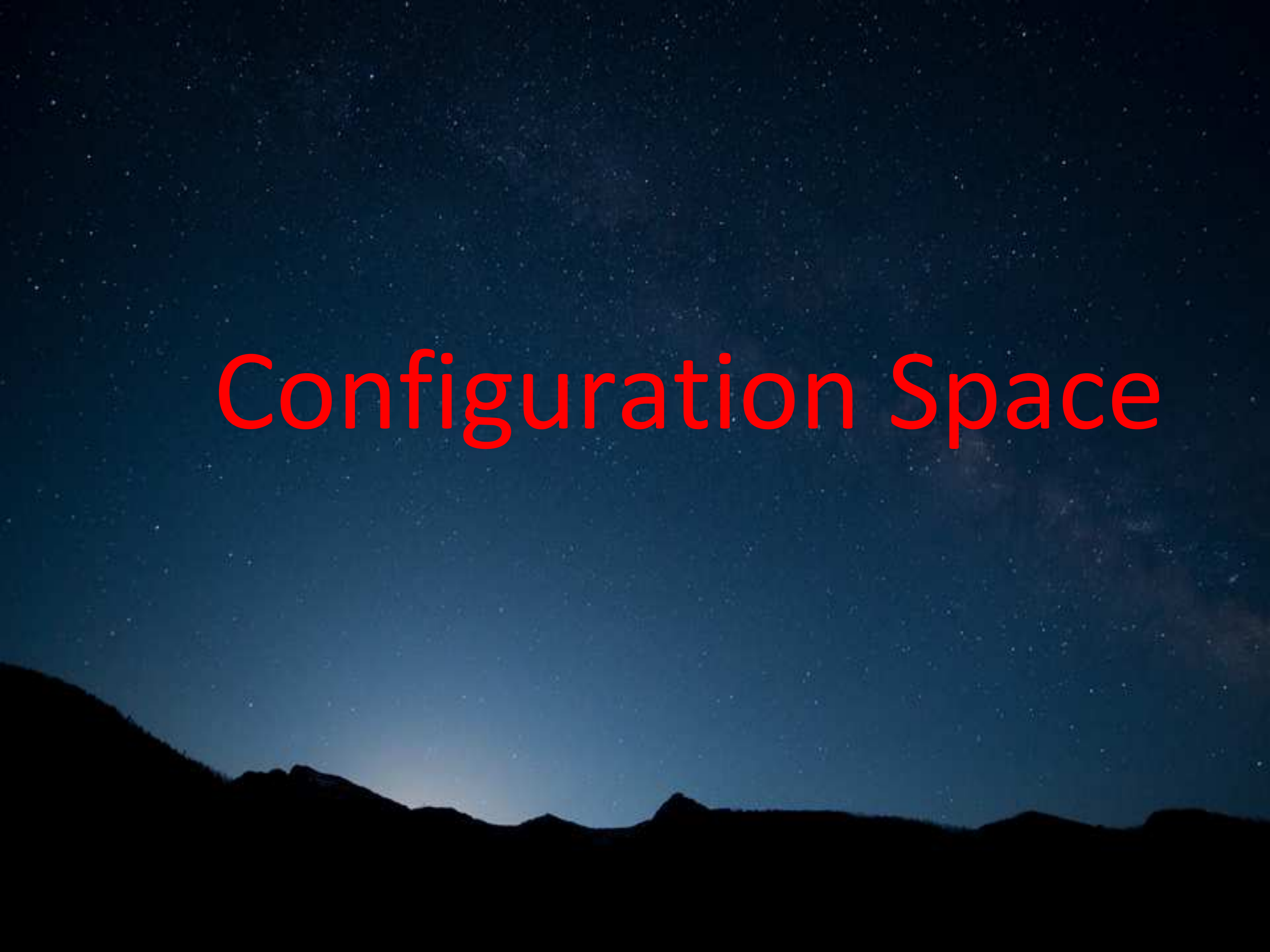
o1 = false
o2 = true
o3 : [2..13]

Configuration space

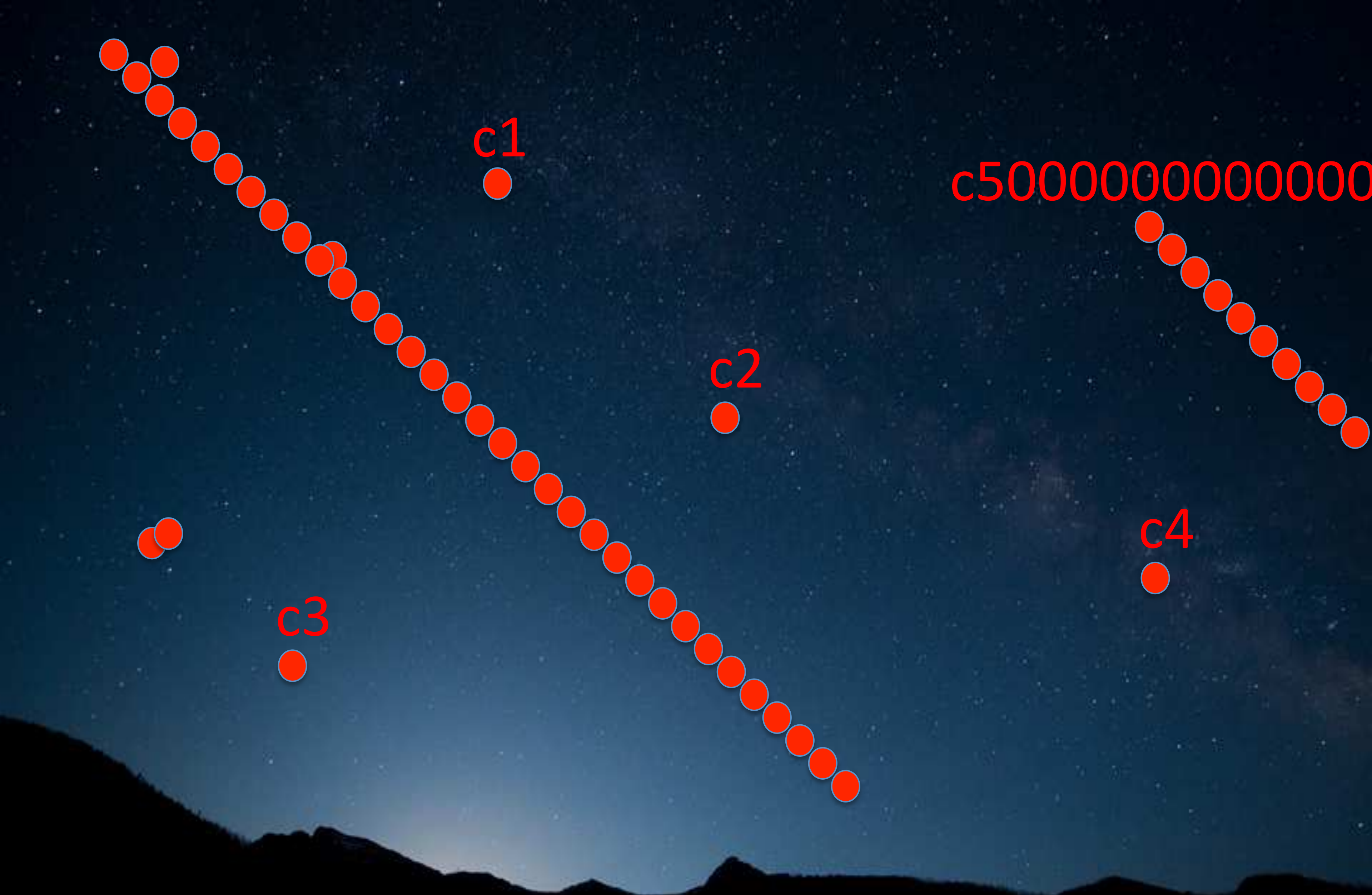
o1 : {true, false}
o2 : {true, false}
o3 : [0..10]



Configuration Space



Configuration Space



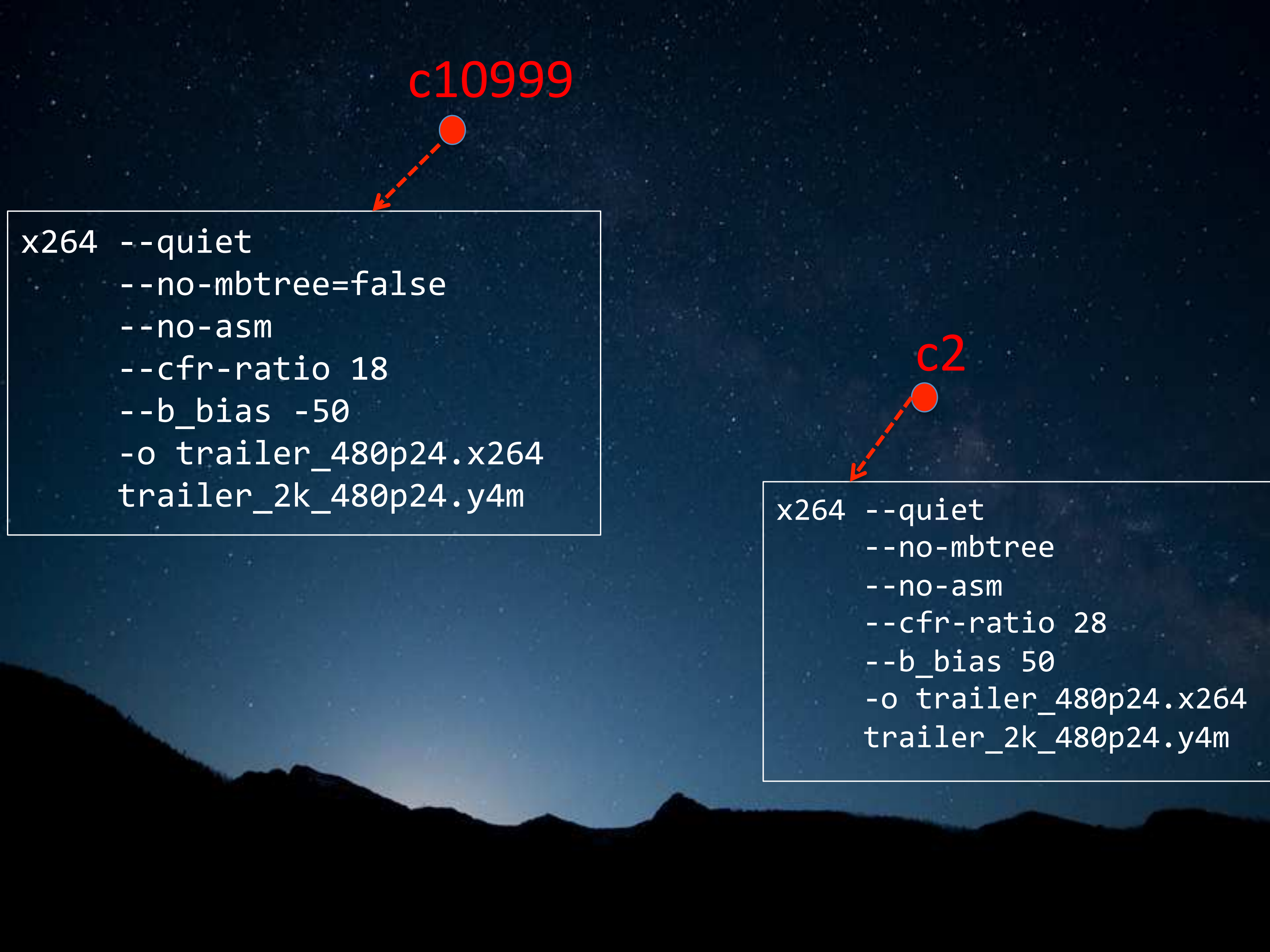
c1000229

c199882

c91989882

c399888

c182



c10999

```
x264 --quiet
     --no-mbtree=false
     --no-asm
     --cfr-ratio 18
     --b_bias -50
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

c2

```
x264 --quiet
     --no-mbtree
     --no-asm
     --cfr-ratio 28
     --b_bias 50
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

c10999

```
x264 --quiet
     --no-mbtree=false
     --no-asm
     --cfr-ratio 18
     --b_bias -50
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

1000s ✗

```
x264 --quiet
     --no-mbtree=false
      --cfr-ratio 28
     --b_bias 50
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

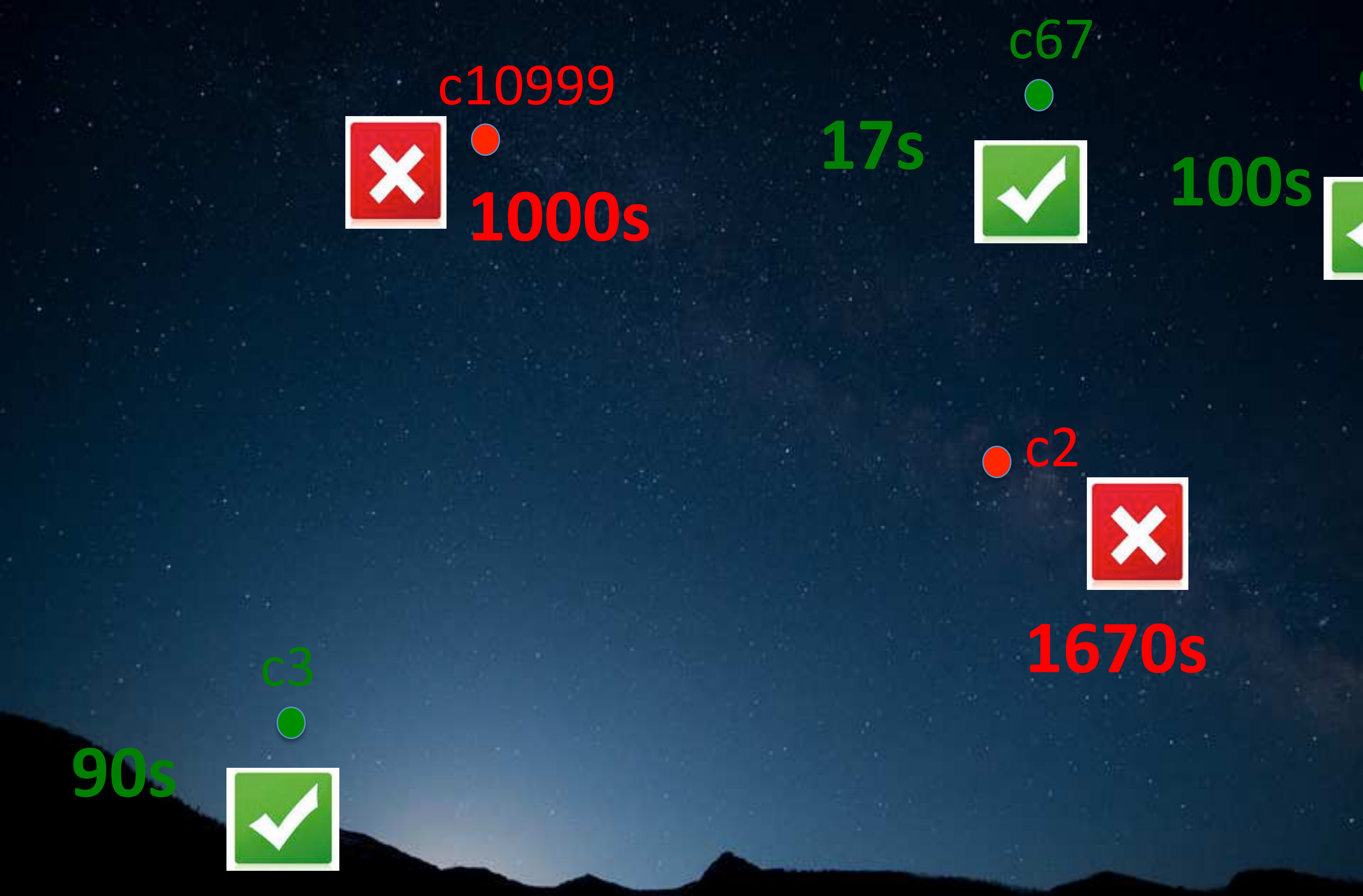
100s ✓

c2

```
x264 --quiet
     --no-mbtree
     --no-asm
     --cfr-ratio 28
     --b_bias 50
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

1670s ✗

I want an execution time < 145s



c10999 ☒ 1000s

17s c67 ☑

100s ☑

c2 ☒ 1670s

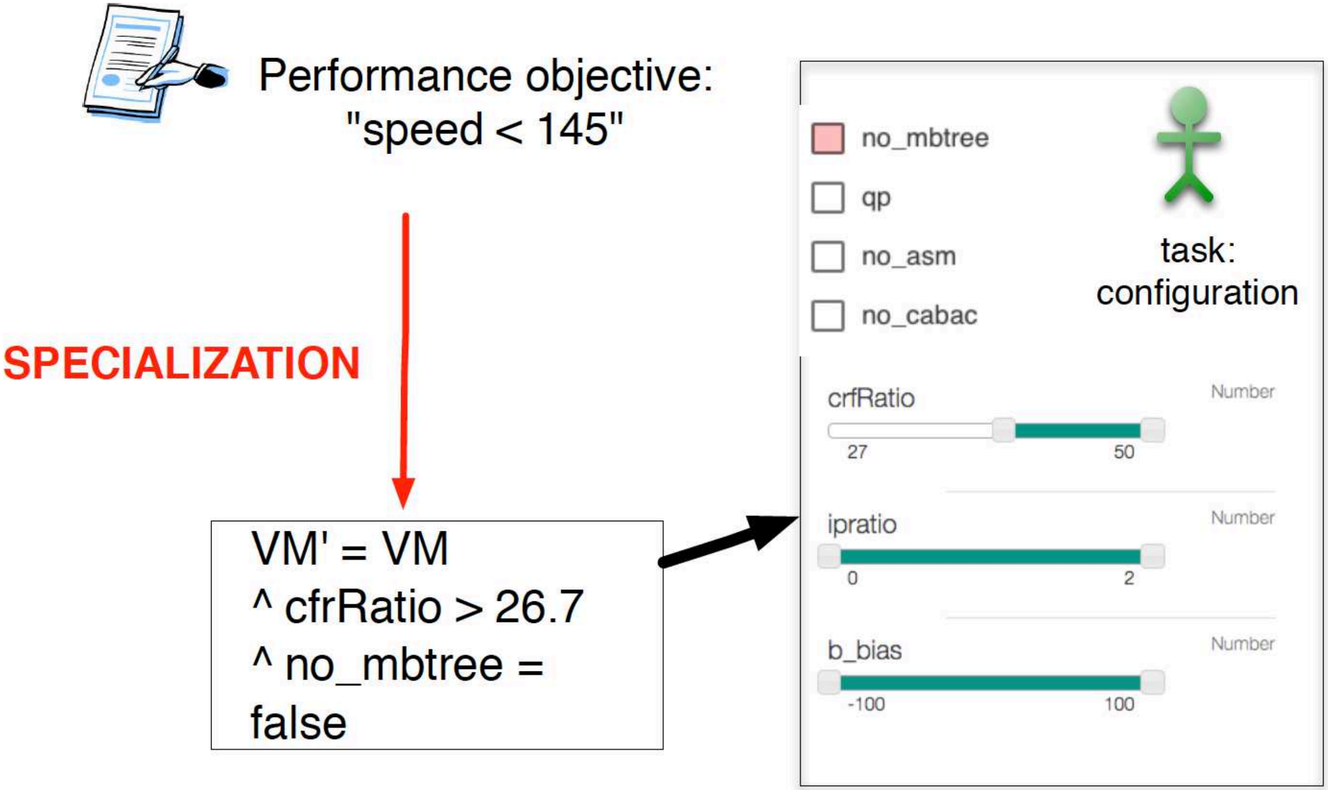
c3 90s ☑

I want an execution time < 145s

Automated specialization problem:

synthesizing constraints such that
each configuration meets an objective

(you have typically to execute the configuration to know that)



Performance objective:
"speed < 145"

SPECIALIZATION

VM' = VM
^ cfrRatio > 26.7
^ no_mbtree =
false

□ no_mbtree
□ qp
□ no_asm
□ no_cabac

crfRatio Number
27 50

ipratio Number
0 2

b_bias Number
-100 100

320 optional, independent, Boolean features

more variants than estimated

atoms in the universe

Impossible to execute and test all configurations

I want an execution time < 145s

I want an execution time < 145s

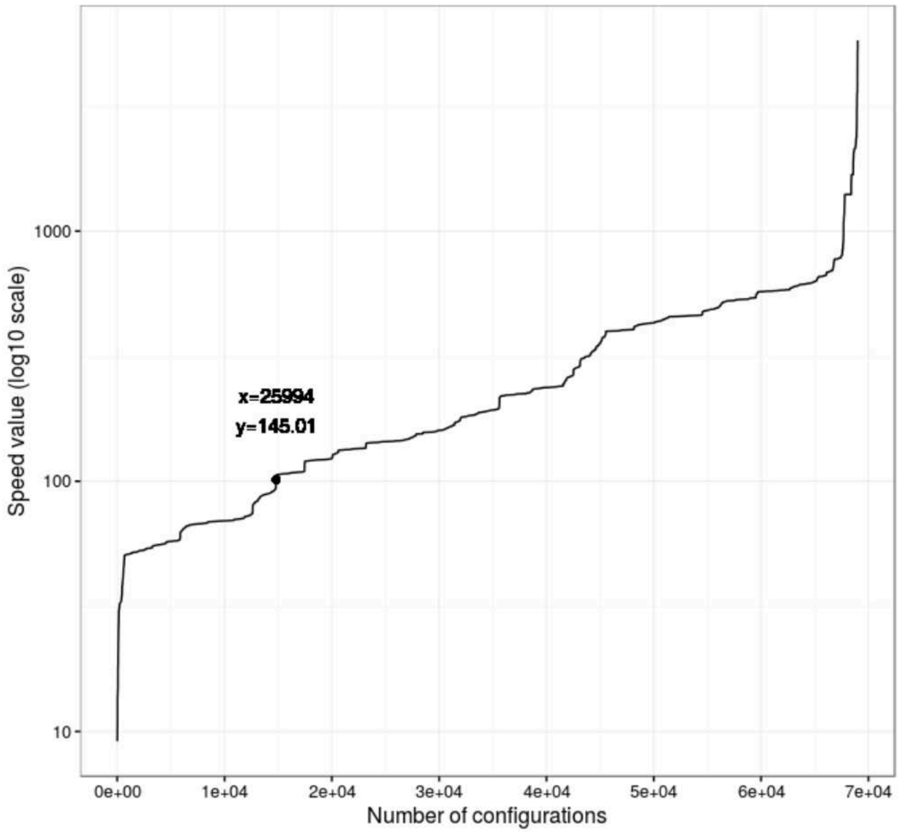
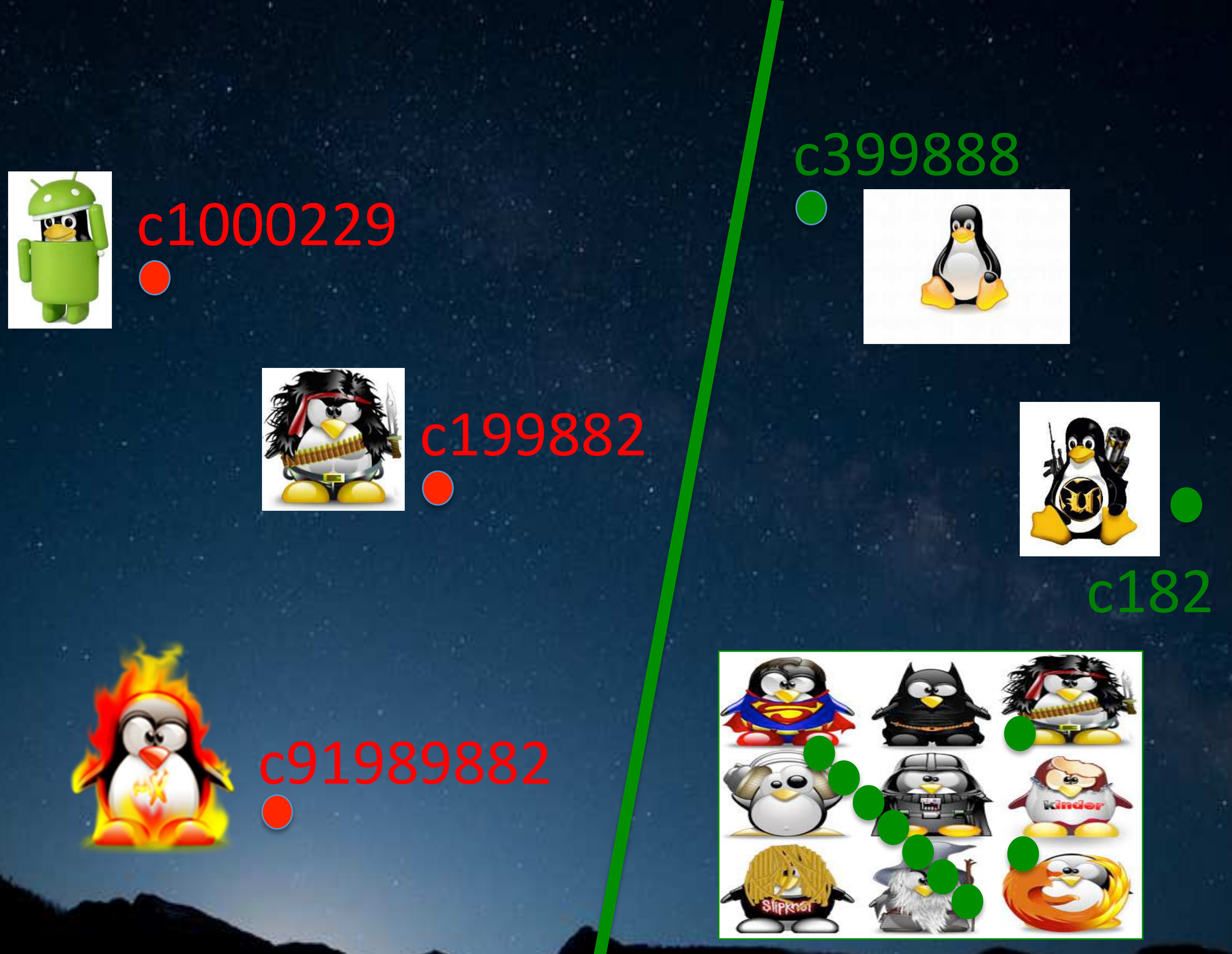


Fig. 2: Number of x264 configurations running under a certain time: X-axis represents a number of configurations; Y-axis represents the execution speed (in seconds) to encode a video benchmark; *e.g.,* about 25994 configurations can encode the video in less than 145.01 seconds.



c1000229

c199882

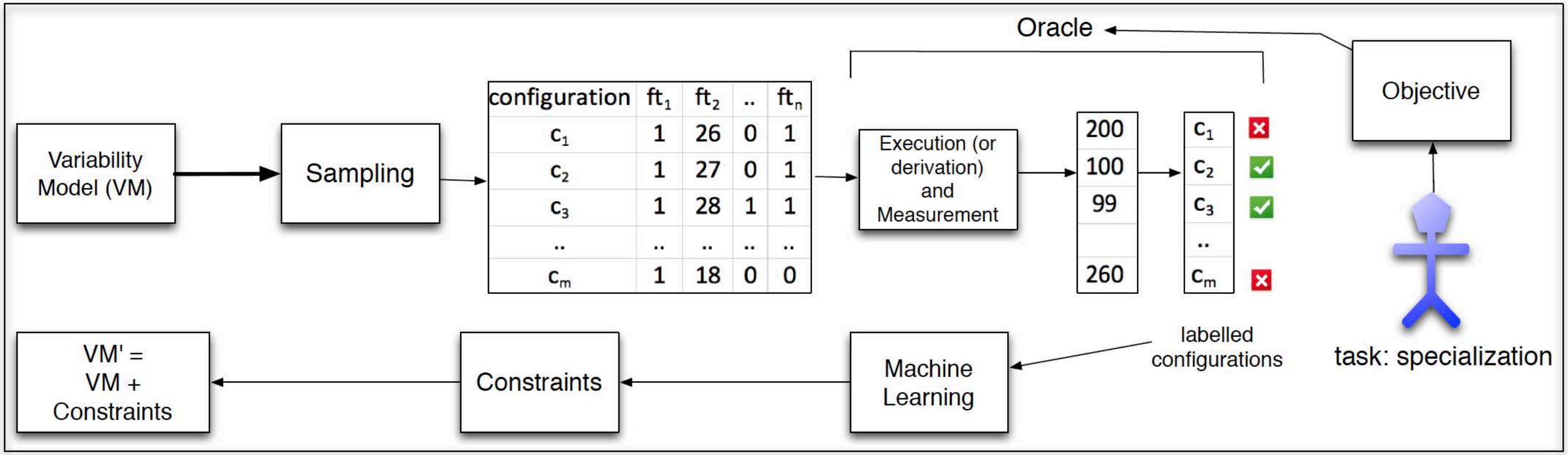
c91989882

c399888

c182

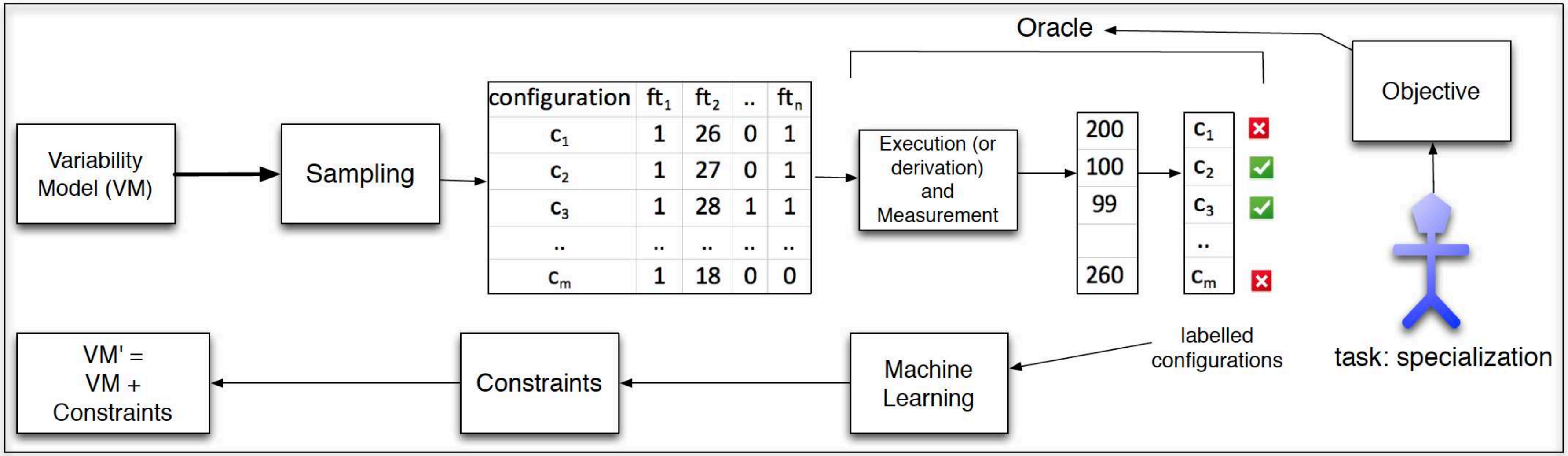
Sampling, Testing, and Learning

Learning-Based Specialization



configuration	ft_1	ft_2	..	ft_n
c_1	1	26	0	1
c_2	1	27	0	1
c_3	1	28	1	1
..	..	..	..	..
c_m	1	18	0	0

200	c_1	☒
100	c_2	✓
99	c_3	✓
	..	
260	c_m	☒



Oracle

Objective

task: specialization

Variability Model (VM)

Sampling

configuration	ft_1	ft_2	..	ft_n
c_1	1	26	0	1
c_2	1	27	0	1
c_3	1	28	1	1
..	..	..	..	..
c_m	1	18	0	0

Execution (or derivation) and Measurement

200	c_1	✗
100	c_2	✓
99	c_3	✓
	..	
260	c_m	✗

labelled configurations

Machine Learning

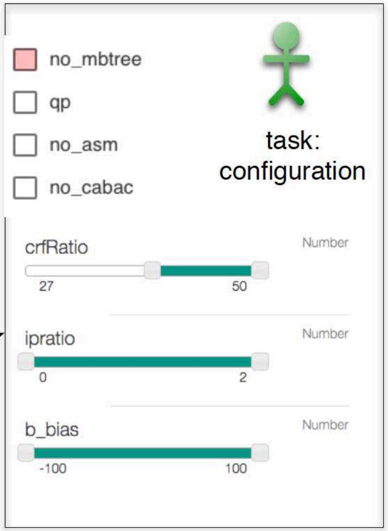
Constraints

VM' =
VM +
Constraints

Performance objective:
"speed < 145"

SPECIALIZATION

VM' = VM
^ cfrRatio > 26.7
^ no_mbtree =
false



■ no_mbtree

□ qp

□ no_asm

□ no_cabac

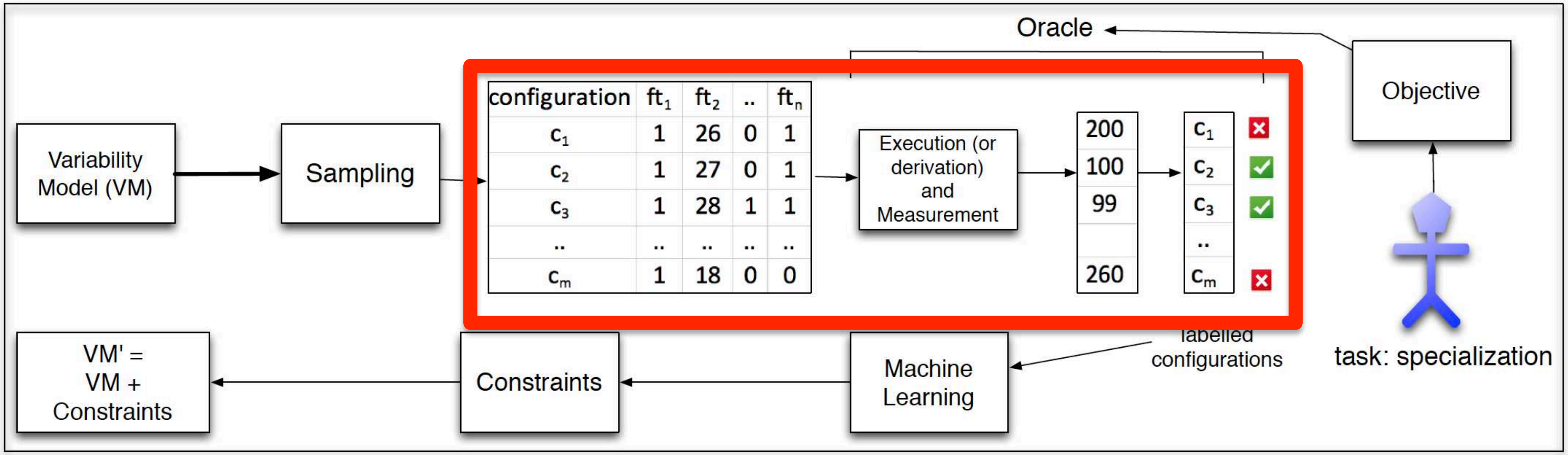
task:
configuration

crfRatio Number 27 50

ipratio Number 0 2

b_bias Number -100 100

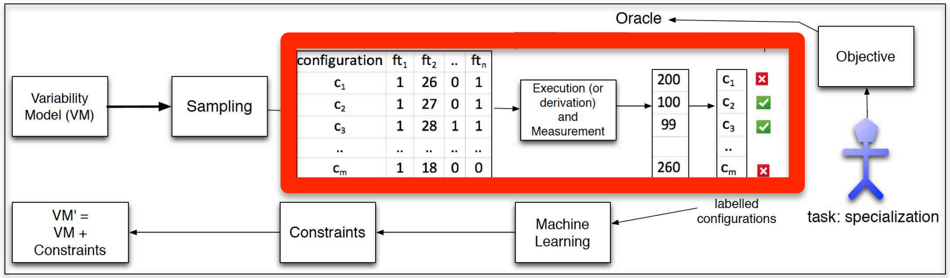
Problem reduction: a binary classification problem
Learning approach: decision trees (classification trees)



configuration	ft_1	ft_2	..	ft_n
c_1	1	26	0	1
c_2	1	27	0	1
c_3	1	28	1	1
..	..	..	..	..
c_m	1	18	0	0

200	c_1	✗
100	c_2	✓
99	c_3	✓
	..	
260	c_m	✗

Classification trees and constraints



configuration	ft_1	ft_2	..	ft_n
c_1	1	26	0	1
c_2	1	27	0	1
c_3	1	28	1	1
..	..	..	..	..
c_m	1	18	0	0

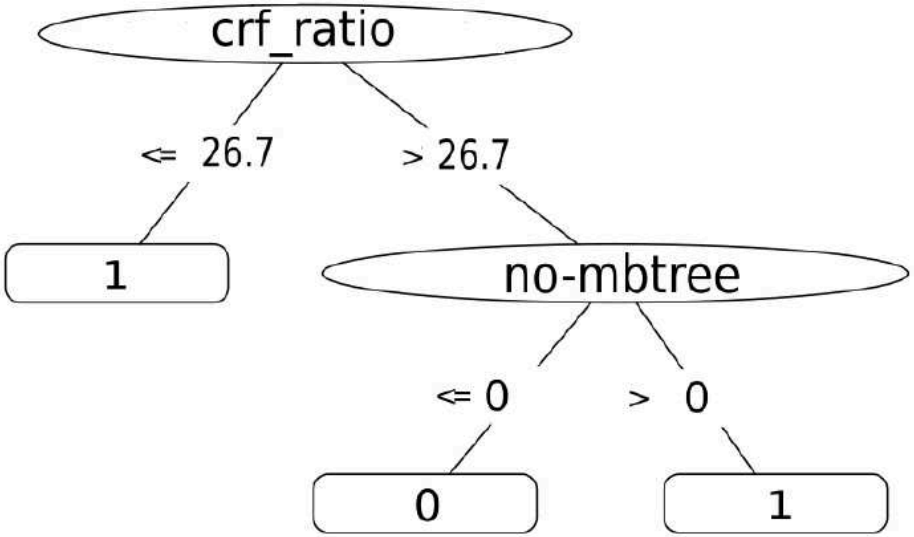
200	c_1	☒
100	c_2	✓
99	c_3	✓
	..	
260	c_m	☒

Why decision trees?

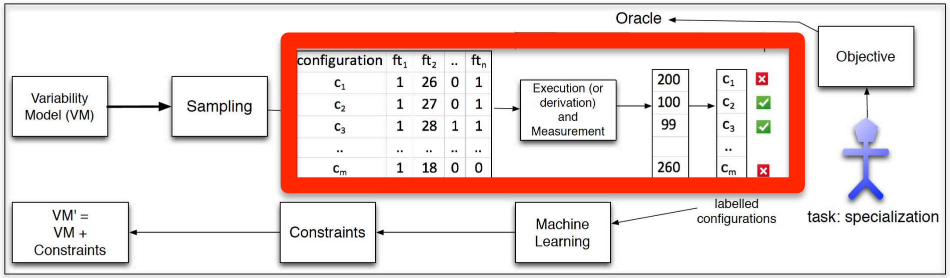
++ Can handle categorical and numerical values

++ Constraints can be extracted into logics

++ Human-readable constraints

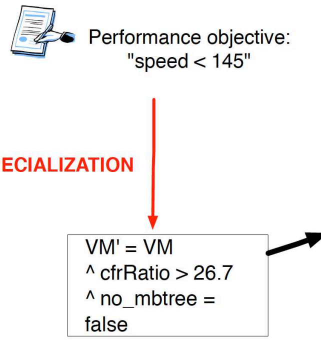


Classification trees and constraints

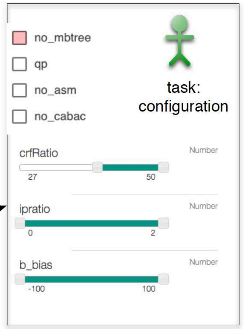


configuration	ft_1	ft_2	..	ft_n
c_1	1	26	0	1
c_2	1	27	0	1
c_3	1	28	1	1
..	..	..	..	..
c_m	1	18	0	0

200	c_1	☒
100	c_2	✓
99	c_3	✓
	..	
260	c_m	☒



VM' = VM
^ cfrRatio > 26.7
^ no_mbtree = false

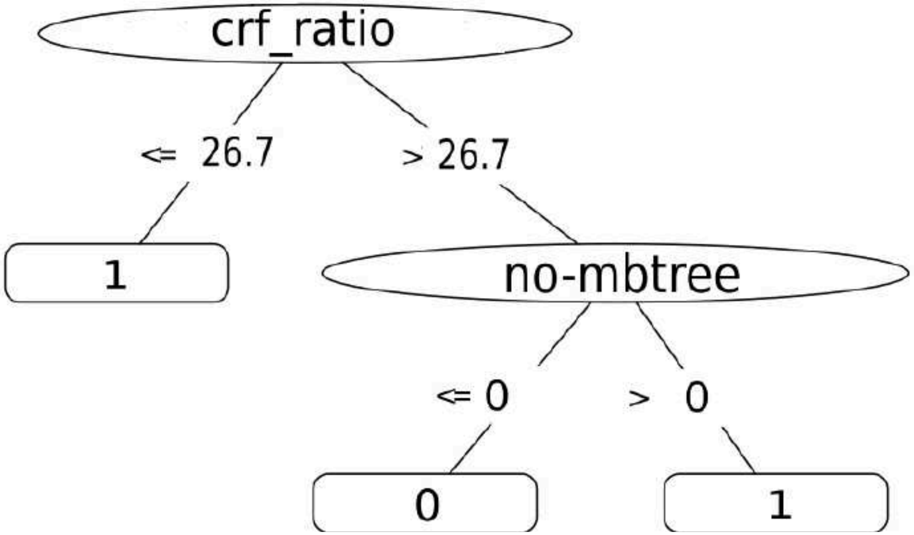


- 🟥 no_mbtree
- □ qp
- □ no_asm
- □ no_cabac

crfRatio — Number — 27 … 50

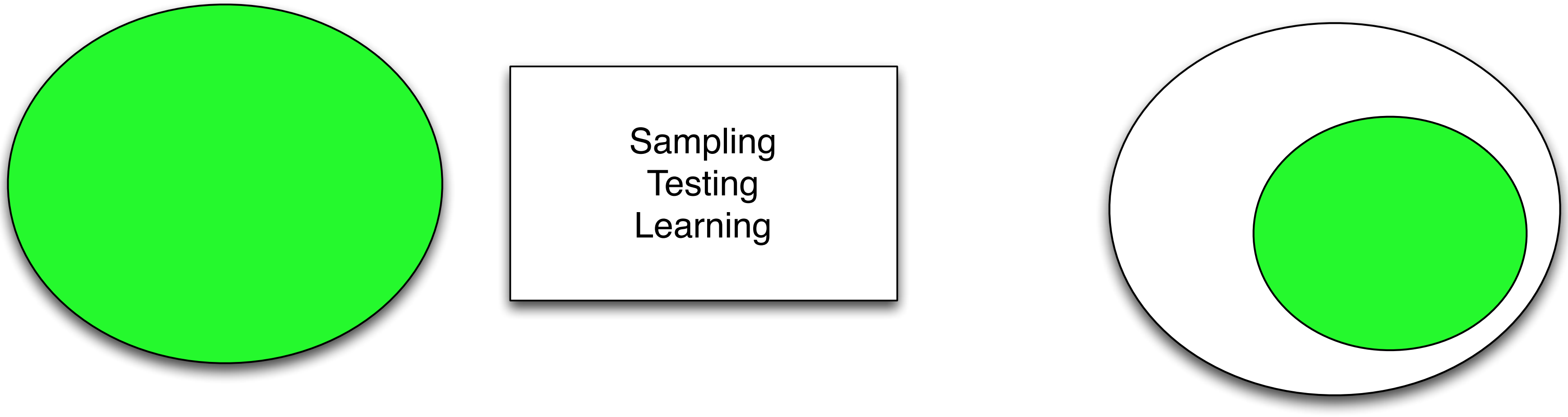
ipratio — Number — 0 … 2

b_bias — Number — -100 … 100



```
!(crf_ratio <= 26.7)
!( crfRatio > 26.7 & no_mbtree > 0 )
```

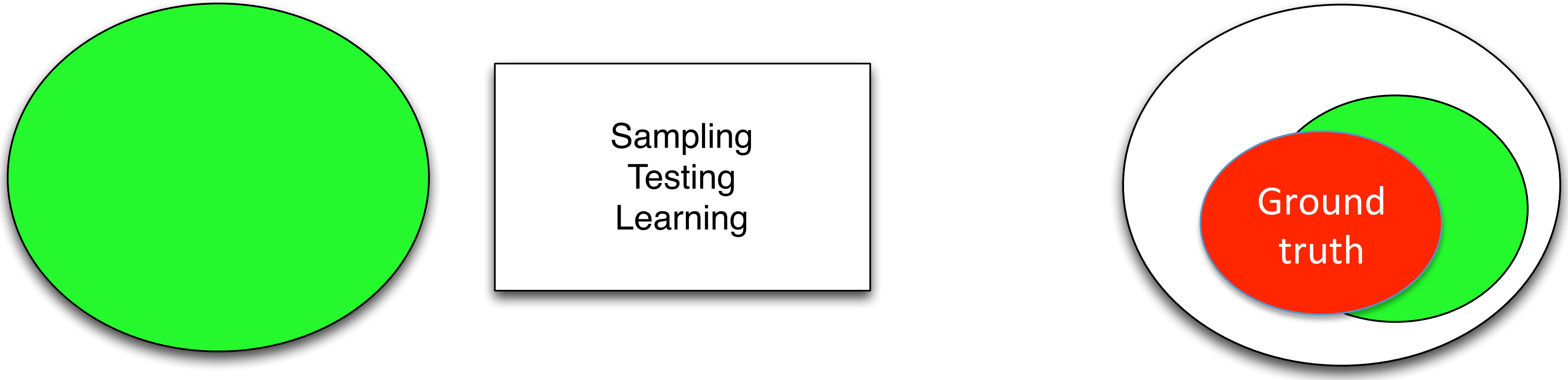
Specialization of the configuration set



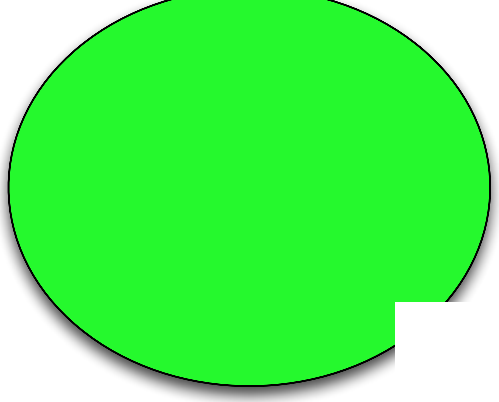
Sampling
Testing
Learning

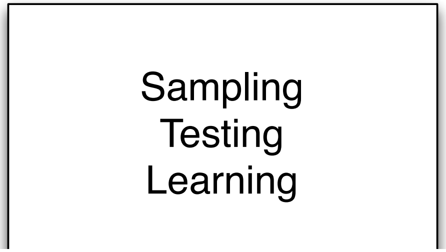
Can discard lots of non-acceptable configurations
(safer)
But can also be too restrictive
(losing flexibility)

Specialization of the configuration set

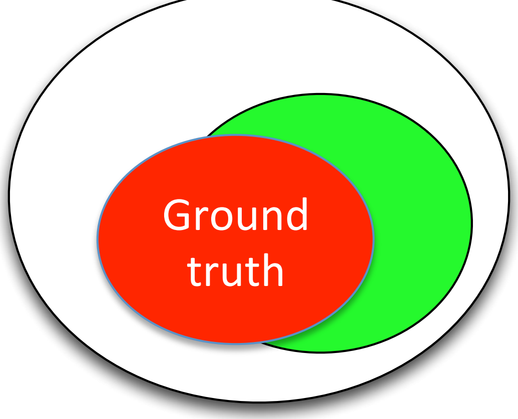


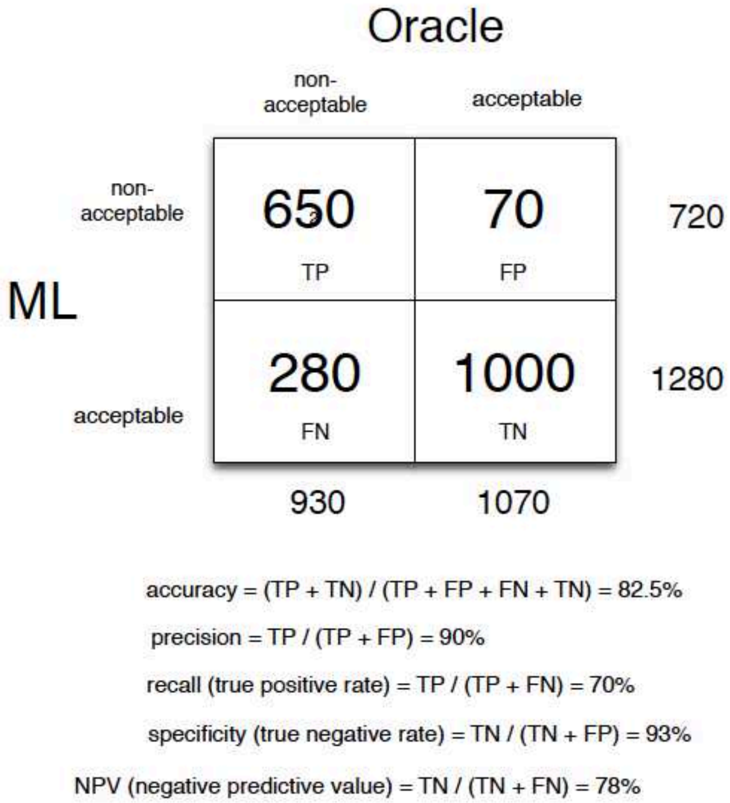
Can identify false positives or false negatives ("missing" flexibility or safety)





Sampling
Testing
Learning





Oracle

		non-acceptable	acceptable	
ML	non-acceptable	650 TP	70 FP	720
	acceptable	280 FN	1000 TN	1280
		930	1070	

accuracy = (TP + TN) / (TP + FP + FN + TN) = 82.5%

precision = TP / (TP + FP) = 90%

recall (true positive rate) = TP / (TP + FN) = 70%

specificity (true negative rate) = TN / (TN + FP) = 93%

NPV (negative predictive value) = TN / (TN + FN) = 78%

Can identify false positives or false negatives (“missing” flexibility or safety)

Evaluation

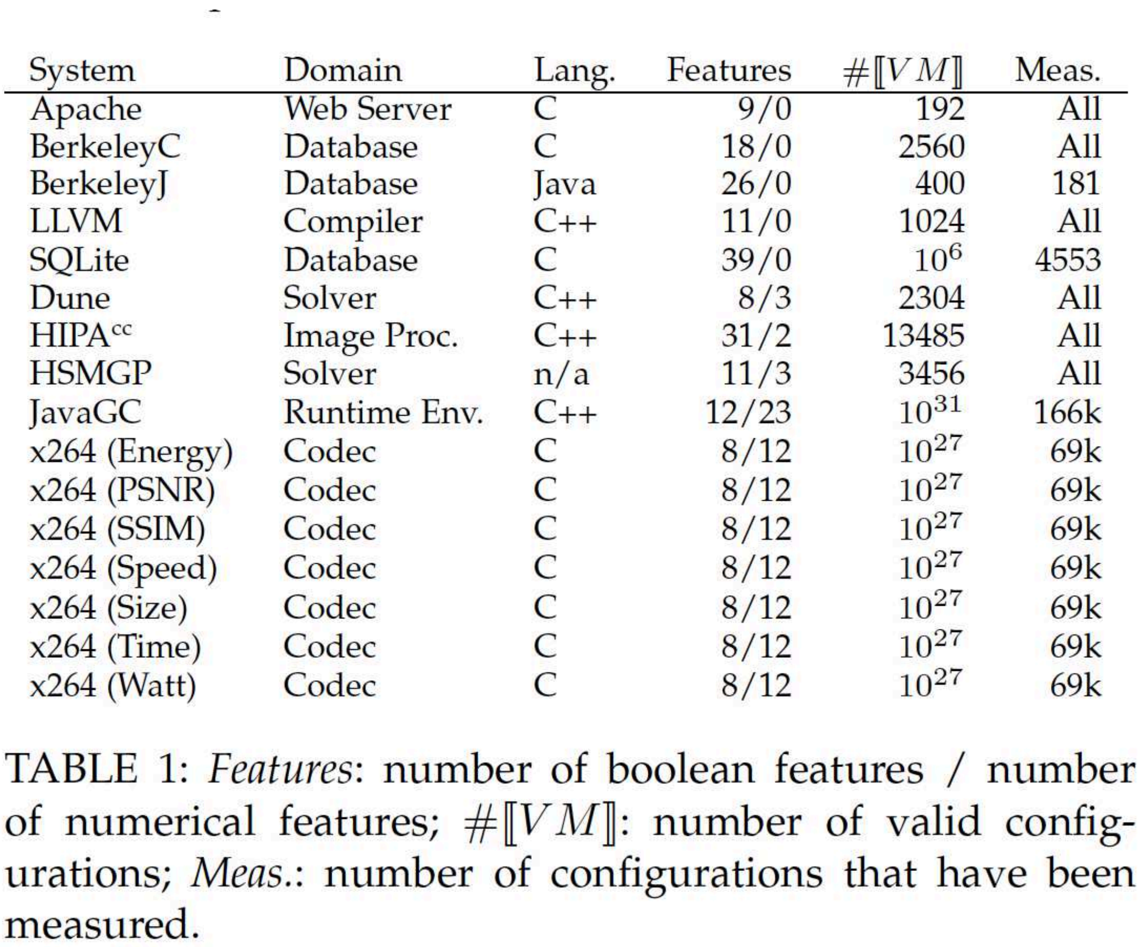
What is the accuracy of our specialization method for classifying configurations?

What is the precision and recall of our specialization method for classifying configurations?

How safe and flexible are specialized configurable systems when applying our method?

How effective is our learning technique compared to a non-learning technique?

Evaluation



System	Domain	Lang.	Features	$\#[\![VM]\!]$	Meas.
Apache	Web Server	C	9/0	192	All
BerkeleyC	Database	C	18/0	2560	All
BerkeleyJ	Database	Java	26/0	400	181
LLVM	Compiler	C++	11/0	1024	All
SQLite	Database	C	39/0	10^6	4553
Dune	Solver	C++	8/3	2304	All
HIPA^{cc}	Image Proc.	C++	31/2	13485	All
HSMGP	Solver	n/a	11/3	3456	All
JavaGC	Runtime Env.	C++	12/23	10^{31}	166k
x264 (Energy)	Codec	C	8/12	10^{27}	69k
x264 (PSNR)	Codec	C	8/12	10^{27}	69k
x264 (SSIM)	Codec	C	8/12	10^{27}	69k
x264 (Speed)	Codec	C	8/12	10^{27}	69k
x264 (Size)	Codec	C	8/12	10^{27}	69k
x264 (Time)	Codec	C	8/12	10^{27}	69k
x264 (Watt)	Codec	C	8/12	10^{27}	69k

TABLE 1: *Features*: number of boolean features / number of numerical features; $\#[\![VM]\!]$: number of valid configurations; *Meas.*: number of configurations that have been measured.

Independent variables

- Subject systems
- Sampling size
- Performance objective
  - % of non-acceptable configurations
- For each subject system, we compute numerous metrics and perform a sensitivity analysis wrt sampling size and performance objective

Learning-Based Performance Specialization of Configurable Systems

Paul Temple, Mathieu Acher, Jean-Marc Jézéquel, Léo Noel-Baron
Univ Rennes, IRISA
Rennes, France
Emails: firstname.lastname@irisa.fr
José A. Galindo
University of Sevilla
Sevilla, Spain
Email: jagalindo@us.es

paper: https://hal.archives-ouvertes.fr/hal-01467299

Note: we are currently further experimenting with new data and algorithms

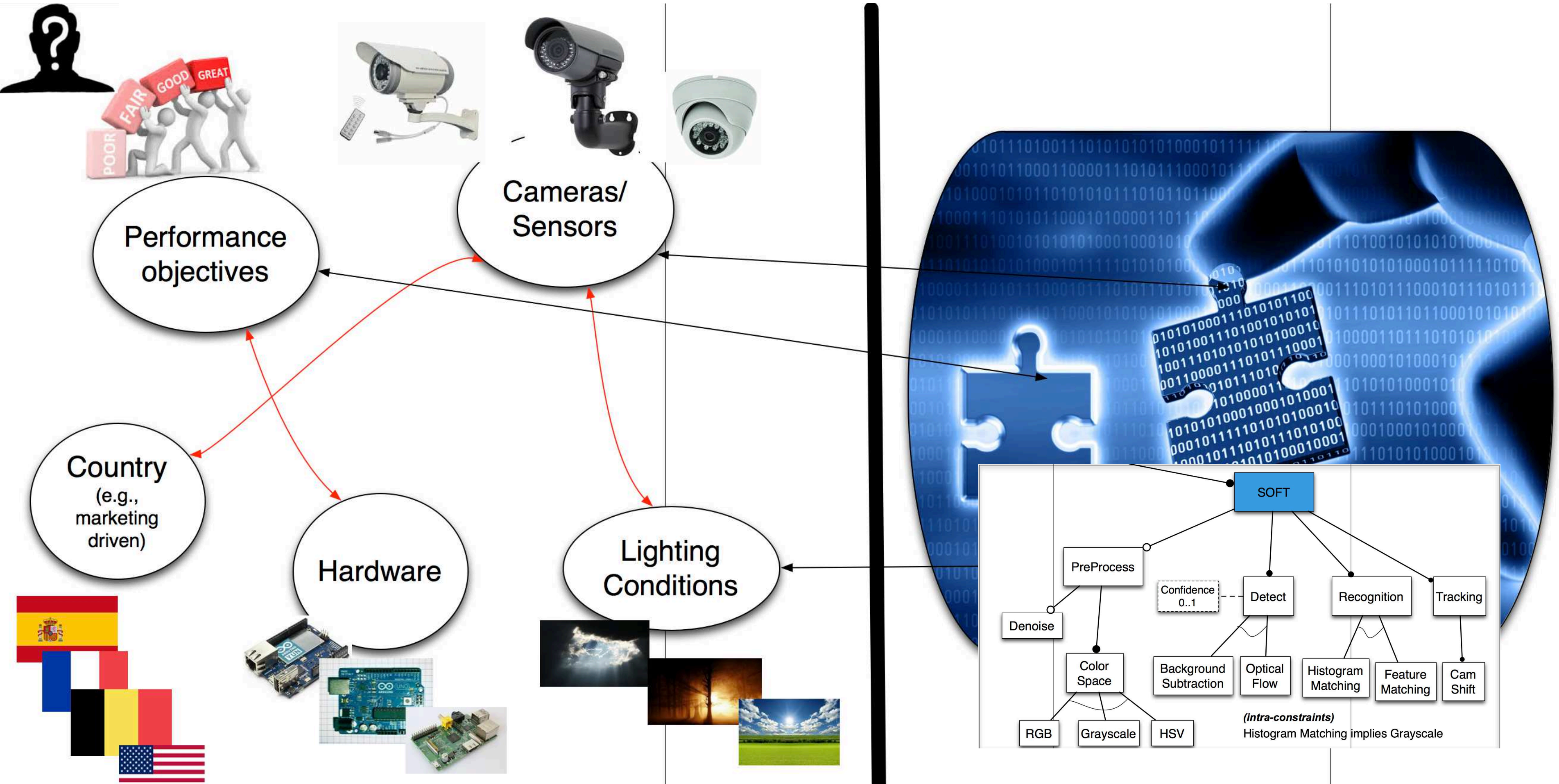
Main conclusions

- **High precision and recall** can be obtained with a relative **small number of configurations** with the exception of some "hard" objective values for which the configurable system can be seen as too permissive.
- Our approach can be effective to produce a **safe and flexible system** with a relative small number of configurations
- Even and especially for hard objectives, our specialization method **significantly outperforms a non-learning approach**

Conclusion

- Software variability everywhere for fitting users’ requirements
- Variability is complexity **(very large configuration spaces)**
- **AI#1** Abstraction/languages to formally and efficiently reason about configuration spaces
  - with SAT/CSP/SMT solvers
  - Eg constrained sampling
- **AI#2** Statistical machine learning to (out of a sample):
  - Understand the configuration space
  - Find the best configuration
  - Specialize the configuration space (e.g., by capturing constraints)
  - In a cost-effective way
- Artificial intelligence for fitting software variability
- Human/machines interplay

Context and Variability

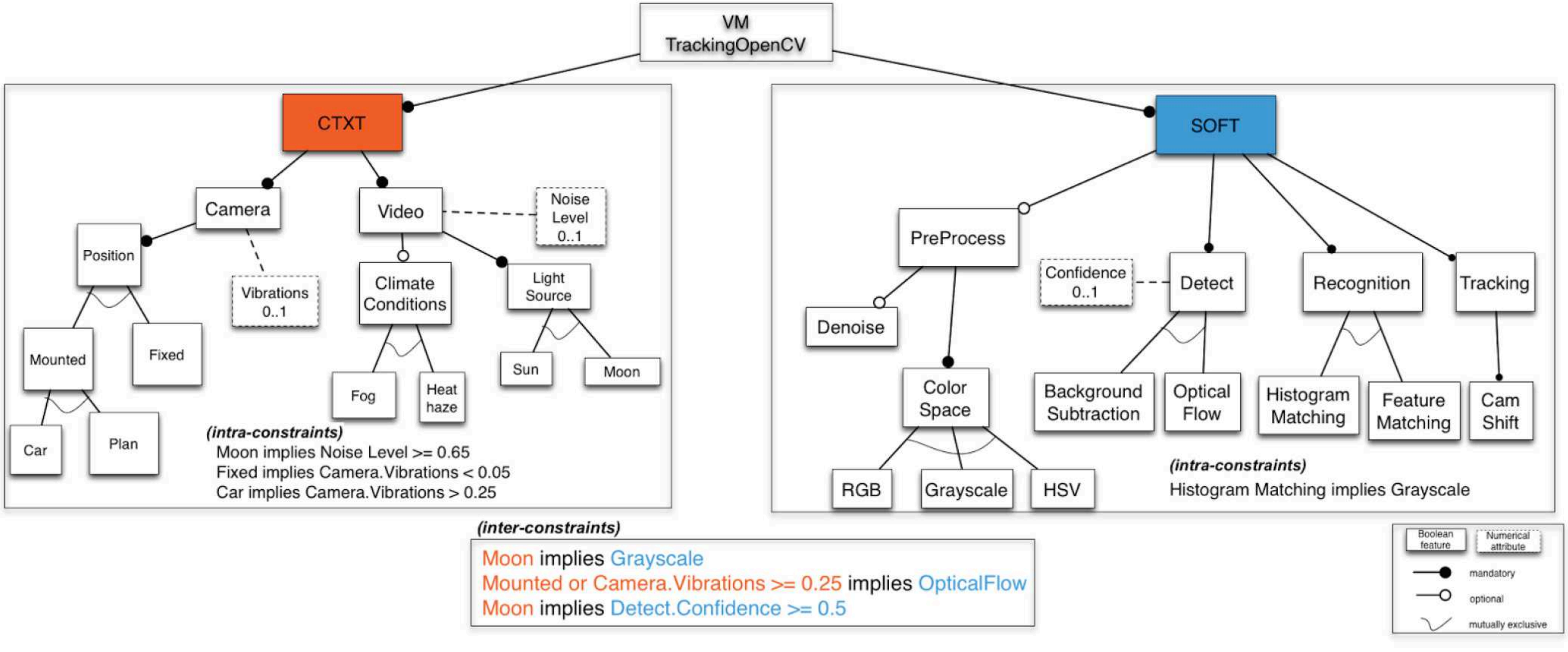


Contextual Variability

Software Variability

Paul Temple, Mathieu Acher, Jean-Marc Jézéquel, and Olivier Barais. Learning-Contextual Variability Models (2017). In IEEE Software

Learning
Contextual Variability Models

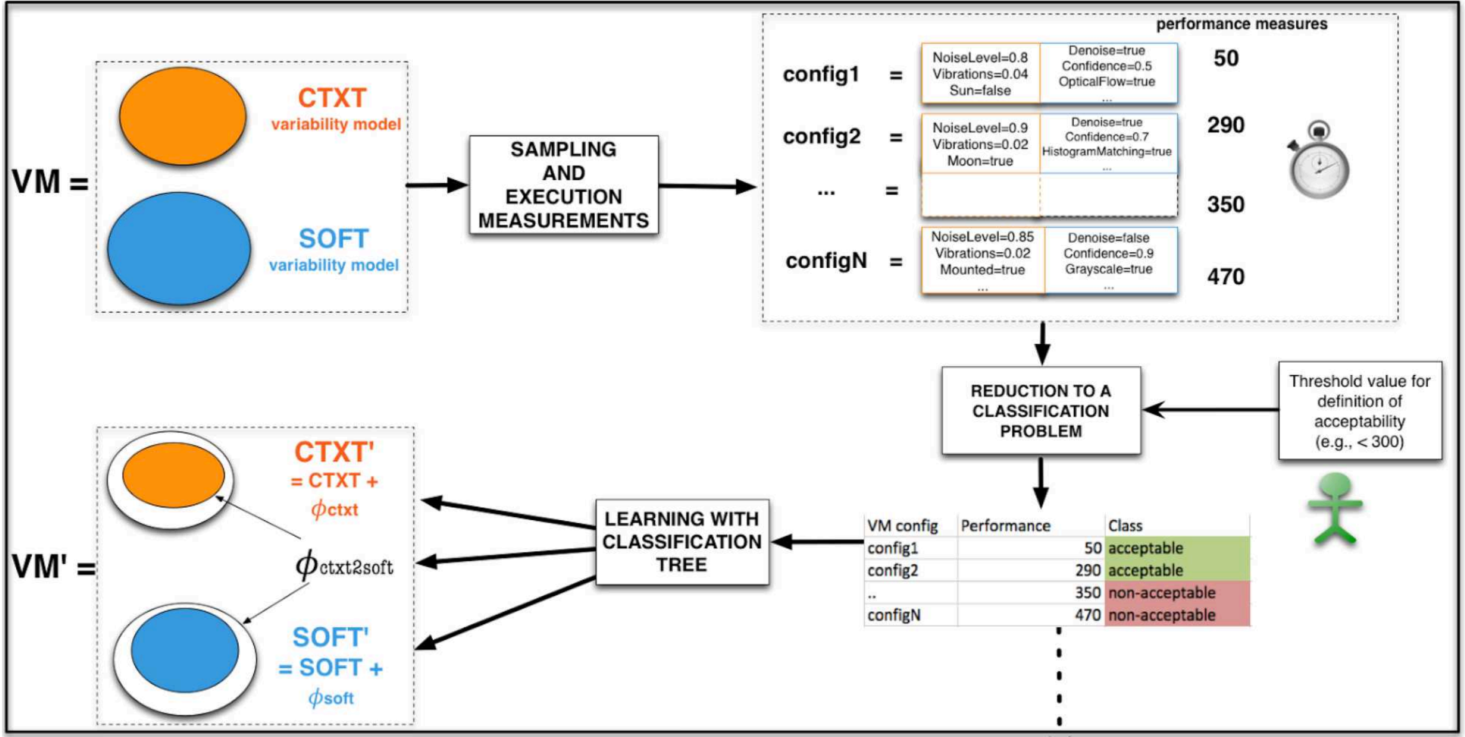


(intra-constraints)
Moon implies Noise Level >= 0.65
Fixed implies Camera.Vibrations < 0.05
Car implies Camera.Vibrations > 0.25

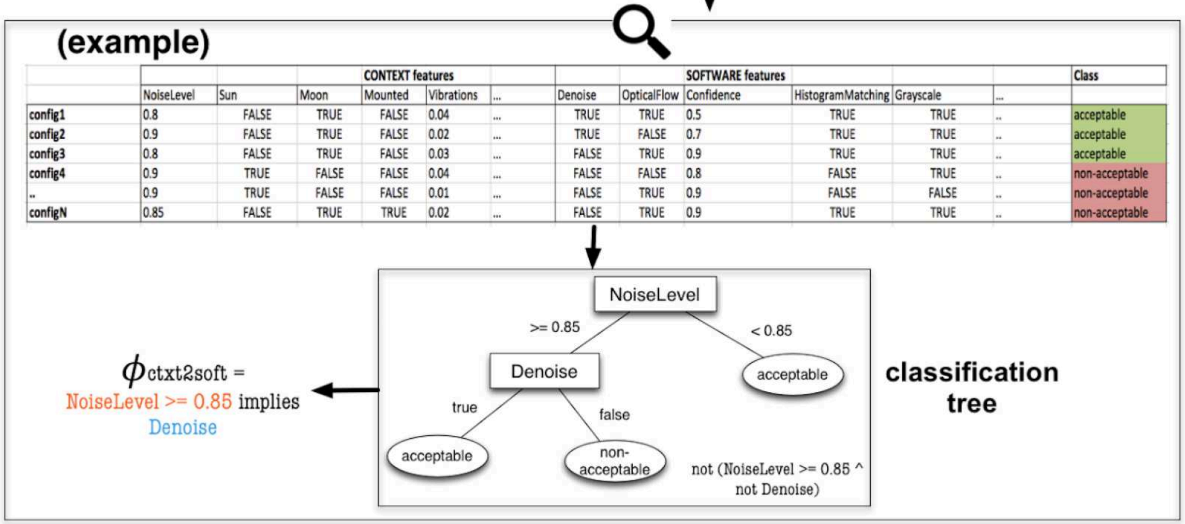
(intra-constraints)
Histogram Matching implies Grayscale

(inter-constraints)
Moon implies Grayscale
Mounted or Camera.Vibrations >= 0.25 implies OpticalFlow
Moon implies Detect.Confidence >= 0.5

Paul Temple, Mathieu Acher, Jean-Marc Jézéquel, and Olivier Barais. Learning-Contextual Variability Models (2017). In IEEE Software

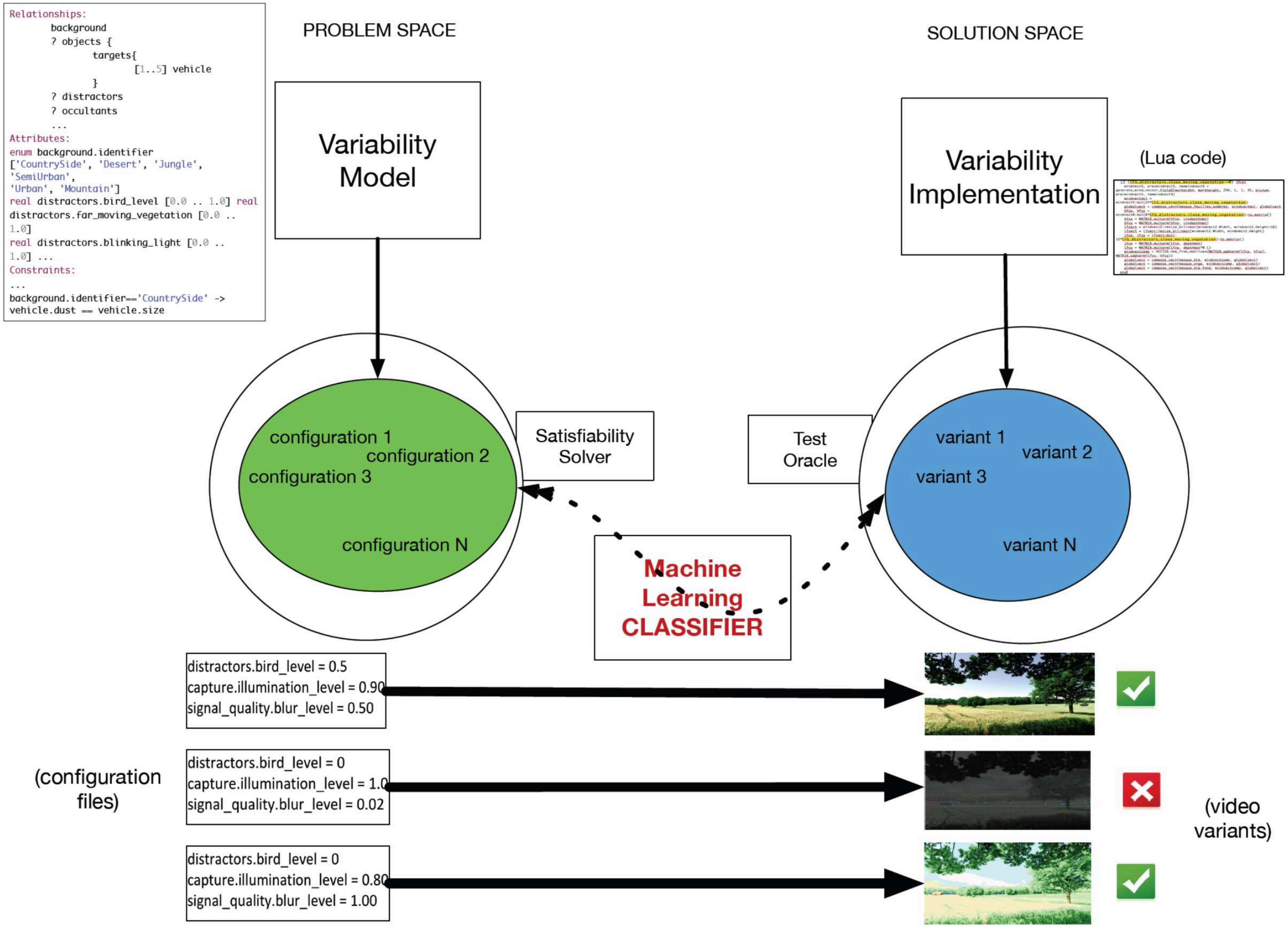


VM config	Performance	Class
config1	50	acceptable
config2	290	acceptable
..	350	non-acceptable
configN	470	non-acceptable



				CONTEXT features					SOFTWARE features				Class
	NoiseLevel	Sun	Moon	Mounted	Vibrations	...	Denoise	OpticalFlow	Confidence	HistogramMatching	Grayscale	...	
config1	0.8	FALSE	TRUE	FALSE	0.04	...	TRUE	TRUE	0.5	TRUE	TRUE	..	acceptable
config2	0.9	FALSE	TRUE	FALSE	0.02	...	TRUE	FALSE	0.7	TRUE	TRUE	..	acceptable
config3	0.8	FALSE	TRUE	FALSE	0.03	...	FALSE	TRUE	0.9	TRUE	TRUE	..	acceptable
config4	0.9	TRUE	FALSE	FALSE	0.04	...	FALSE	FALSE	0.8	FALSE	TRUE	..	non-acceptable
..	0.9	TRUE	FALSE	FALSE	0.01	...	FALSE	TRUE	0.9	FALSE	FALSE	..	non-acceptable
configN	0.85	FALSE	TRUE	TRUE	0.02	...	FALSE	TRUE	0.9	TRUE	TRUE	..	non-acceptable

Paul Temple, Mathieu Acher, Jean-Marc Jézéquel, and Olivier Barais. Learning-Contextual Variability Models (2017). In IEEE Software



```
Relationships:
        background
        ? objects {
                targets{
                        [1..5] vehicle
                }
        ? distractors
        ? occultants
        ...
Attributes:
enum background.identifier
['CountrySide', 'Desert', 'Jungle',
'SemiUrban',
'Urban', 'Mountain']
real distractors.bird_level [0.0 .. 1.0] real
distractors.far_moving_vegetation [0.0 ..
1.0]
real distractors.blinking_light [0.0 ..
1.0] ...
Constraints:
...
background.identifier=='CountrySide' ->
vehicle.dust == vehicle.size
```

```
  if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 250, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect1:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
    hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmap)
    hfvy = MATRIX.multerm(hfvy, invdepthmap)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(-
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmap)
    lfvy = MATRIX.multerm(lfvy, depthmap*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
  end
```

```
distractors.bird_level = 0.5
capture.illumination_level = 0.90
signal_quality.blur_level = 0.50
```

✓

```
distractors.bird_level = 0
capture.illumination_level = 1.0
signal_quality.blur_level = 0.02
```

✗

```
distractors.bird_level = 0
capture.illumination_level = 0.80
signal_quality.blur_level = 1.00
```

✓

Paul Temple, Mathieu Acher, Battista Biggio, Jean-Marc Jézéquel, Fabio Roli:
Towards Adversarial Configurations for Software Product Lines. CoRR abs/1805.12021 (2018)

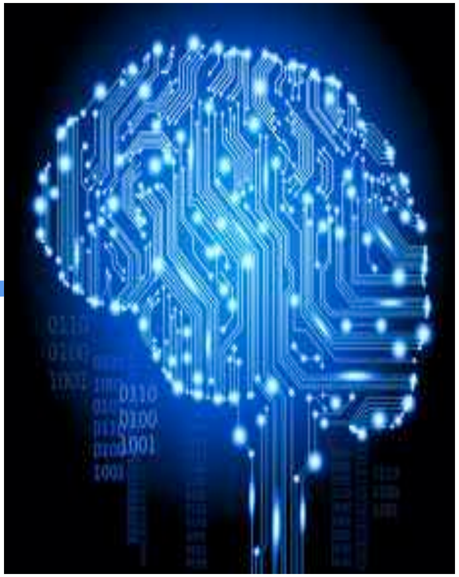
Software Variability and EJCP

- Empirical Software Engineering
  - We aim to understand real-world variability (data)
  - We aim to develop techniques that are effective on real-world systems
- Constraint Programming
  - SAT/SMP/CP solvers to reason about variability
- Coccinelle and the Linux kernel: a challenging case study for software variability
- Formal verification: many papers on verifying software product lines (Thuem et al. ACM Survey 2014)
- Privacy/security: some configurations can raise problems we don’t see with default configurations

Ongoing works

Linux

Machine
Learning



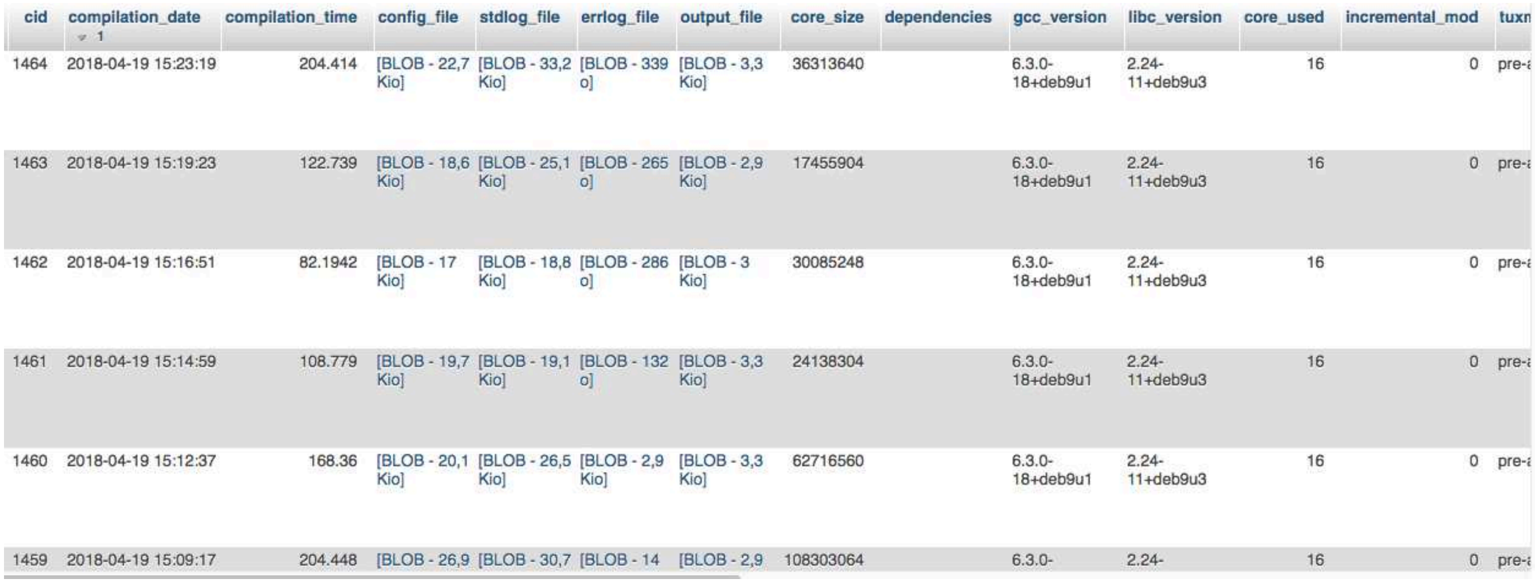
TUXML

Grid'5000

Software Engineering and Machine Learning

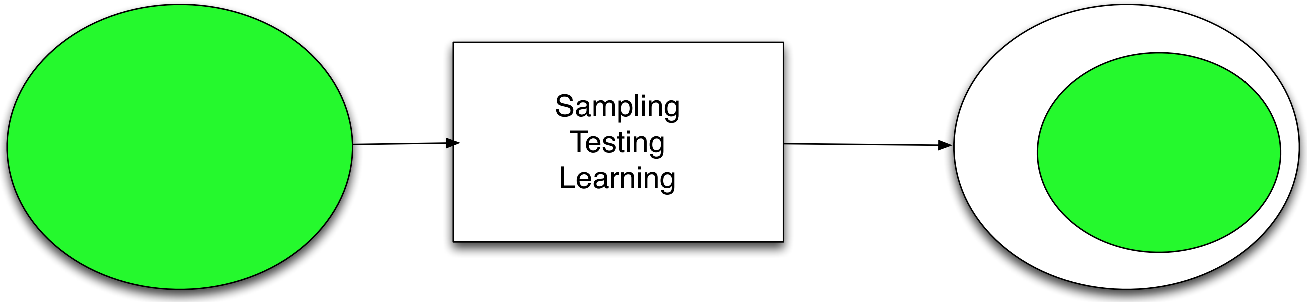
- Automated measurements of thousands of Linux variants
- Learning with a high precision, with a small sample

configuration options: 12K+
70K+ configurations (!!)



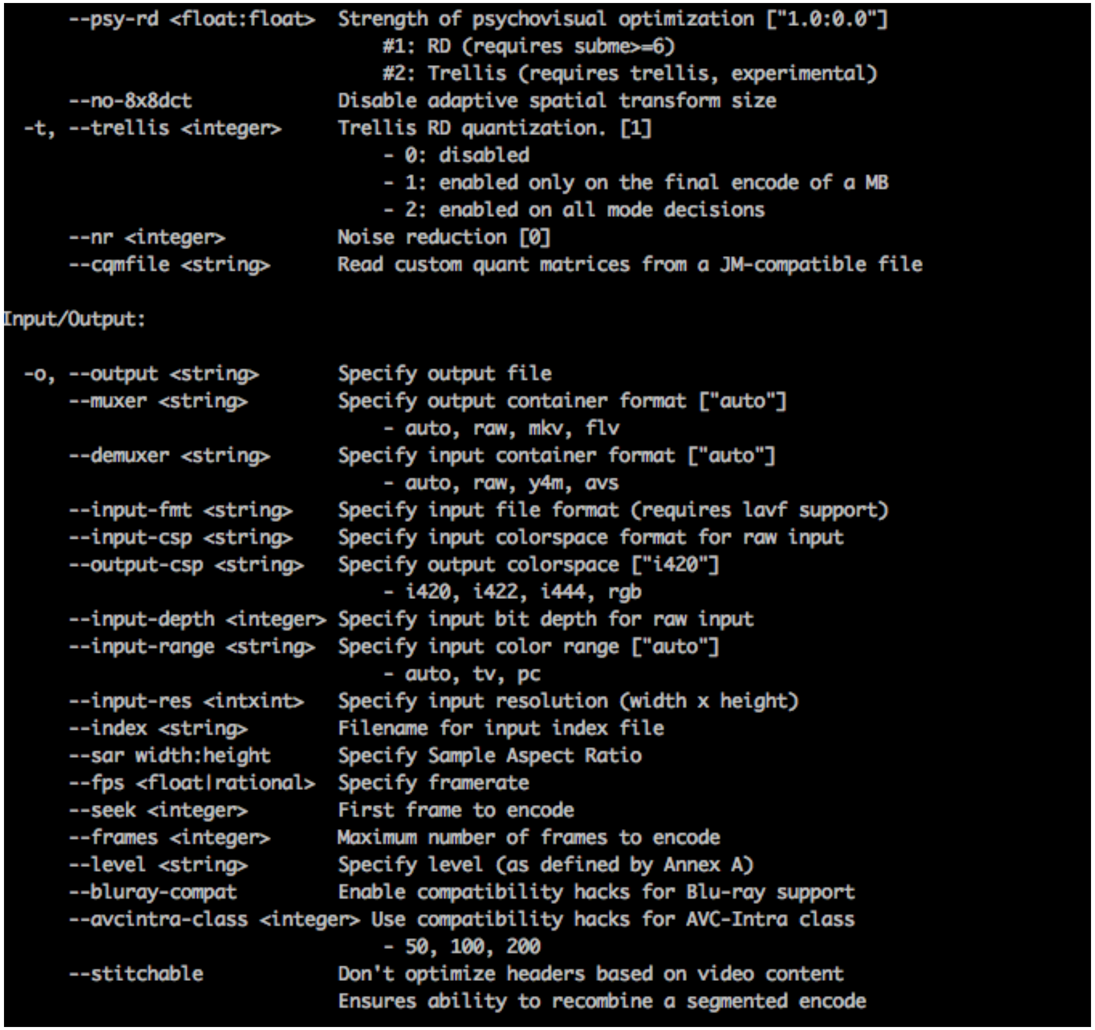
cid	compilation_date ▼ 1	compilation_time	config_file	stdlog_file	errlog_file	output_file	core_size	dependencies	gcc_version	libc_version	core_used	incremental_mod	tuxn
1464	2018-04-19 15:23:19	204.414	[BLOB - 22,7 Kio]	[BLOB - 33,2 Kio]	[BLOB - 339 o]	[BLOB - 3,3 Kio]	36313640		6.3.0-18+deb9u1	2.24-11+deb9u3	16	0	pre-a
1463	2018-04-19 15:19:23	122.739	[BLOB - 18,6 Kio]	[BLOB - 25,1 Kio]	[BLOB - 265 o]	[BLOB - 2,9 Kio]	17455904		6.3.0-18+deb9u1	2.24-11+deb9u3	16	0	pre-a
1462	2018-04-19 15:16:51	82.1942	[BLOB - 17 Kio]	[BLOB - 18,8 Kio]	[BLOB - 286 o]	[BLOB - 3 Kio]	30085248		6.3.0-18+deb9u1	2.24-11+deb9u3	16	0	pre-a
1461	2018-04-19 15:14:59	108.779	[BLOB - 19,7 Kio]	[BLOB - 19,1 Kio]	[BLOB - 132 o]	[BLOB - 3,3 Kio]	24138304		6.3.0-18+deb9u1	2.24-11+deb9u3	16	0	pre-a
1460	2018-04-19 15:12:37	168.36	[BLOB - 20,1 Kio]	[BLOB - 26,5 Kio]	[BLOB - 2,9 Kio]	[BLOB - 3,3 Kio]	62716560		6.3.0-18+deb9u1	2.24-11+deb9u3	16	0	pre-a
1459	2018-04-19 15:09:17	204.448	[BLOB - 26,9	[BLOB - 30,7	[BLOB - 14	[BLOB - 2,9	108303064		6.3.0-	2.24-	16	0	pre-a

o1 : {true, false}
o2 : {true, false}
o3 : [0..10]



o1 = false
o2 = {true, false}
o3 : [2..8]
o3 > 6 => o2

Learning-based specialization for only keeping Linux kernels that are less than 20Mb



```
    --psy-rd <float:float>  Strength of psychovisual optimization ["1.0:0.0"]
                                #1: RD (requires subme>=6)
                                #2: Trellis (requires trellis, experimental)
    --no-8x8dct             Disable adaptive spatial transform size
-t, --trellis <integer>     Trellis RD quantization. [1]
                                - 0: disabled
                                - 1: enabled only on the final encode of a MB
                                - 2: enabled on all mode decisions
    --nr <integer>          Noise reduction [0]
    --cqmfile <string>      Read custom quant matrices from a JM-compatible file

Input/Output:

-o, --output <string>       Specify output file
    --muxer <string>        Specify output container format ["auto"]
                                - auto, raw, mkv, flv
    --demuxer <string>      Specify input container format ["auto"]
                                - auto, raw, y4m, avs
    --input-fmt <string>    Specify input file format (requires lavf support)
    --input-csp <string>    Specify input colorspace format for raw input
    --output-csp <string>   Specify output colorspace ["i420"]
                                - i420, i422, i444, rgb
    --input-depth <integer> Specify input bit depth for raw input
    --input-range <string>  Specify input color range ["auto"]
                                - auto, tv, pc
    --input-res <intxint>   Specify input resolution (width x height)
    --index <string>        Filename for input index file
    --sar width:height      Specify Sample Aspect Ratio
    --fps <float|rational>  Specify framerate
    --seek <integer>        First frame to encode
    --frames <integer>      Maximum number of frames to encode
    --level <string>        Specify level (as defined by Annex A)
    --bluray-compat         Enable compatibility hacks for Blu-ray support
    --avcintra-class <integer> Use compatibility hacks for AVC-Intra class
                                - 50, 100, 200
    --stitchable            Don't optimize headers based on video content
                            Ensures ability to recombine a segmented encode
```

Performance Prediction

```
x264 --no-progress
     --no-asm
     --rc-lookahead 60
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

40 seconds

Performance Prediction

```
x264 --no-mbtree
     --rc-lookahead 40
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

10 seconds

Performance Prediction

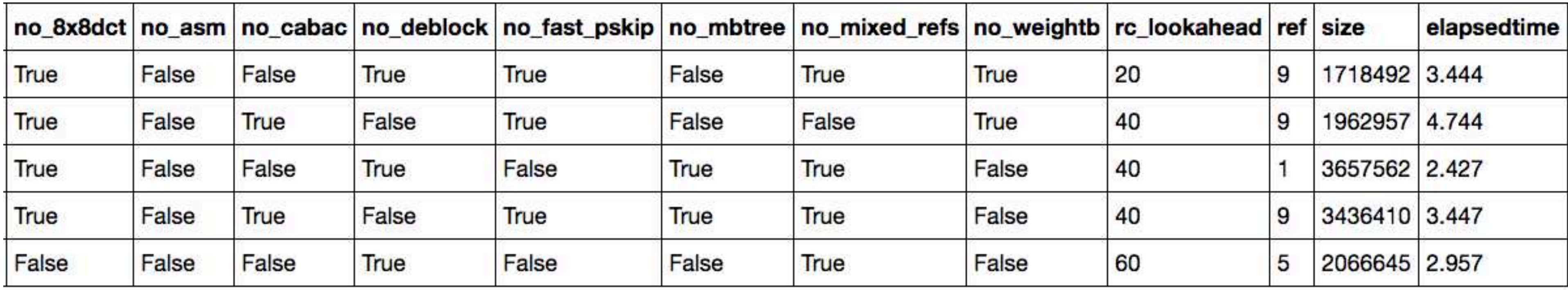
```
x264 ...
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

??? seconds

Performance Prediction

```
x264 --no-mbtree
     --rc-lookahead 40
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

??? seconds



no_8x8dct	no_asm	no_cabac	no_deblock	no_fast_pskip	no_mbtree	no_mixed_refs	no_weightb	rc_lookahead	ref	size	elapsedtime
True	False	False	True	True	False	True	True	20	9	1718492	3.444
True	False	True	False	True	False	False	True	40	9	1962957	4.744
True	False	False	True	False	True	True	False	40	1	3657562	2.427
True	False	True	False	True	True	True	False	40	9	3436410	3.447
False	False	False	True	False	False	True	False	60	5	2066645	2.957

Regression problem (linear regression, regression tree, random forest, gradient boosting, SVM, etc.)

Guo et al. ASE 2013, Apel et al. ASE’15, Czarnecki et al. SPLC’15, Siegmund et al. FSE’15, Kastner et al. ASE’17, Menzies et al. FSE’17, Batory et al. FSE’17

Input Sensitivity and Transferability of Performance Prediction Models

(ongoing work)

What if I change the input video?

Can I reuse my performance prediction model?

```
x264 --no-mbtree
     --rc-lookahead 40
     --ref 9
     -o trailer_480p24.x264
     trailer_2k_480p24.y4m
```

```
x264 --no-mbtree
     --rc-lookahead 40
     --ref 9
     -o football.x264
     football.y4m
```

55 seconds

?

?? seconds