



[Magdeburger Dom]

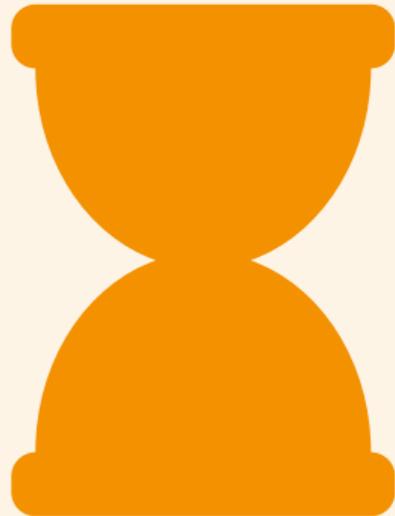
Towards Effective and Efficient Feature-Model Analyses for Evolving System Software

SPLC 2025 — September 1–5 — A Coruña, Spain

Elias Kuiter¹ (Advisors: Gunter Saake¹, Thomas Thüm²)

University of Magdeburg¹ (TU Braunschweig²)

Motivation

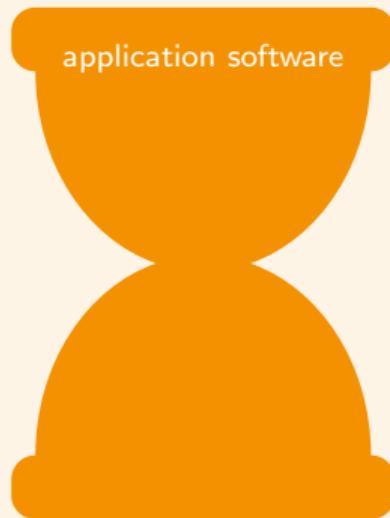


Motivation ► Pipeline ► Contributions ► Snapshots ► Evolution ► Timeline

Motivation

e.g., business software, databases

application software



Motivation

e.g., business software, databases

application software

embedded and
cyber-physical systems

e.g., cars, internet of things



Motivation

e.g., business software, databases

application software



system
software

e.g., operating
systems, firmware

embedded and
cyber-physical systems

e.g., cars, internet of things



Motivation

Variability in System Software

e.g., business software, databases

application software



system
software

e.g., operating
systems, firmware



embedded and
cyber-physical systems

e.g., cars, internet of things



Motivation

Variability in System Software

e.g., business software, databases

application software



system
software

e.g., operating
systems, firmware



embedded and
cyber-physical systems

e.g., cars, internet of things

System Software

Safety

Security

Variability



Motivation

System Software

Safety

Security

Variability

Motivation

Variability in Space

- **product lines** for mass customization
(in system software: Linux, BusyBox, ...)

System Software

Safety

Security

Variability

Variability in Space

Motivation

Variability in Space

- **product lines** for mass customization
(in system software: Linux, BusyBox, ...)
- **feature models** to describe variability
(in system software: KConfig language)

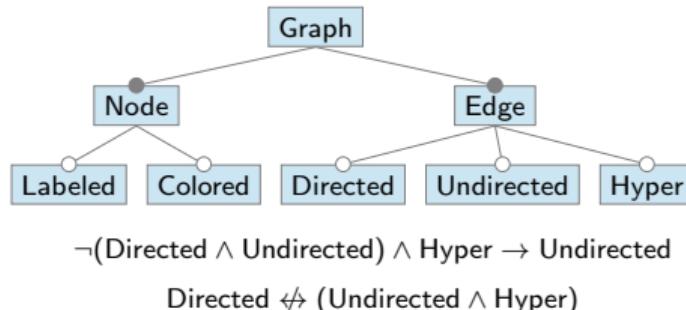
System Software

Safety

Security

Variability

Variability in Space



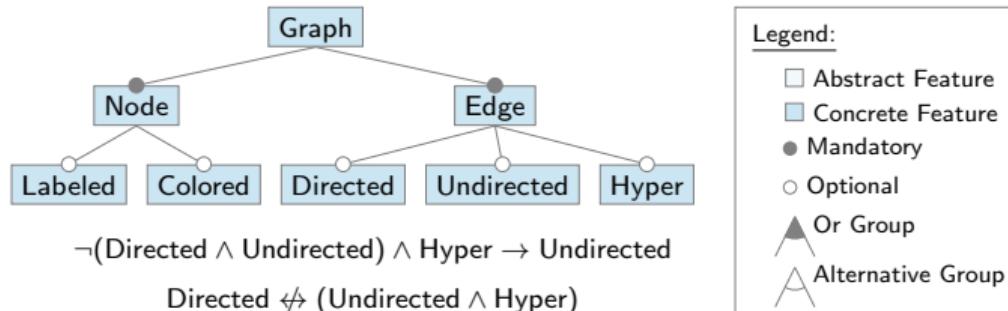
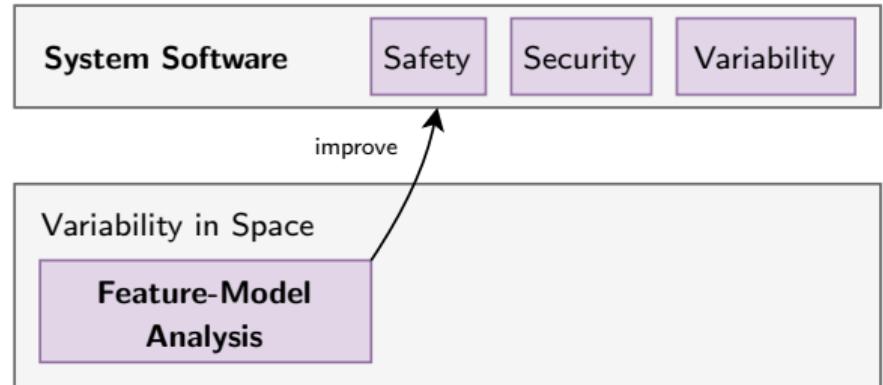
Legend:

- Abstract Feature
- Concrete Feature
- Mandatory
- Optional
- ▲ Or Group
- △ Alternative Group

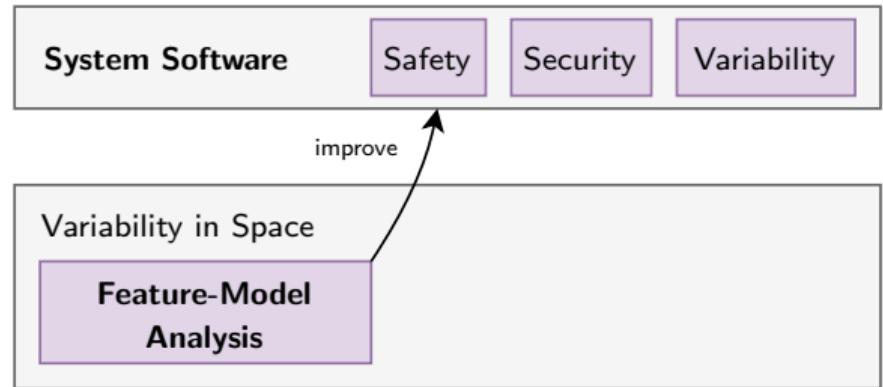
Motivation

Variability in Space

- **product lines** for mass customization
(in system software: Linux, BusyBox, ...)
- **feature models** to describe variability
(in system software: KConfig language)
- **automated analyses** with formulas



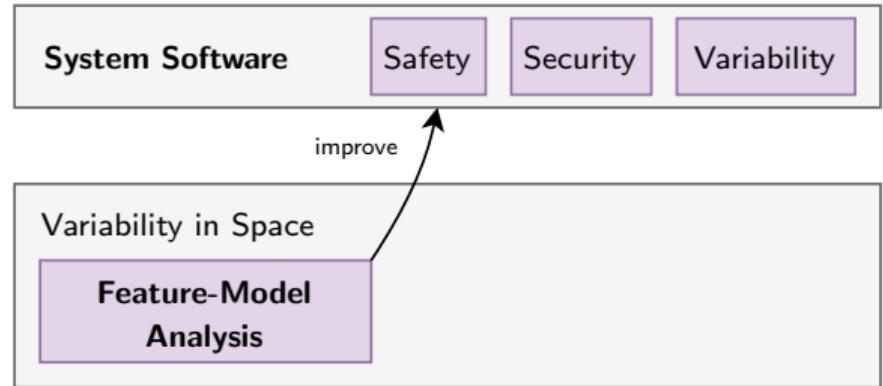
Motivation



Motivation

Variability in Time

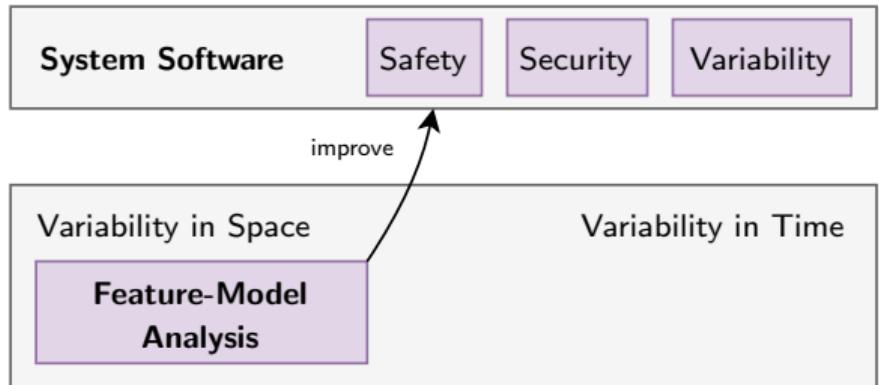
- most software **evolves** over time



Motivation

Variability in Time

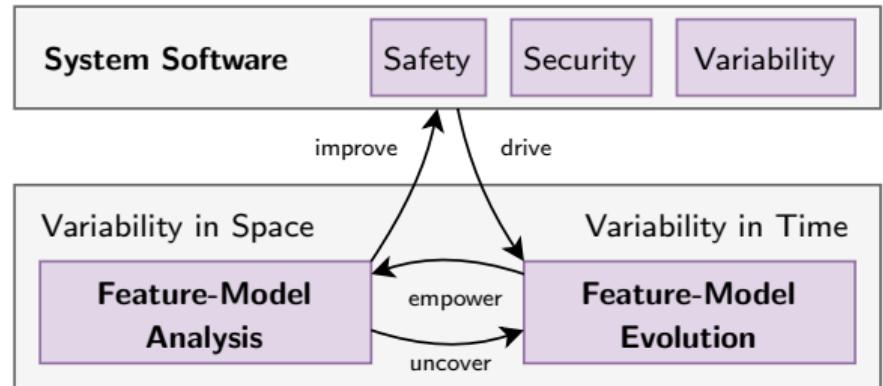
- most software **evolves** over time
- so do product lines
(long-term, open histories in system software)



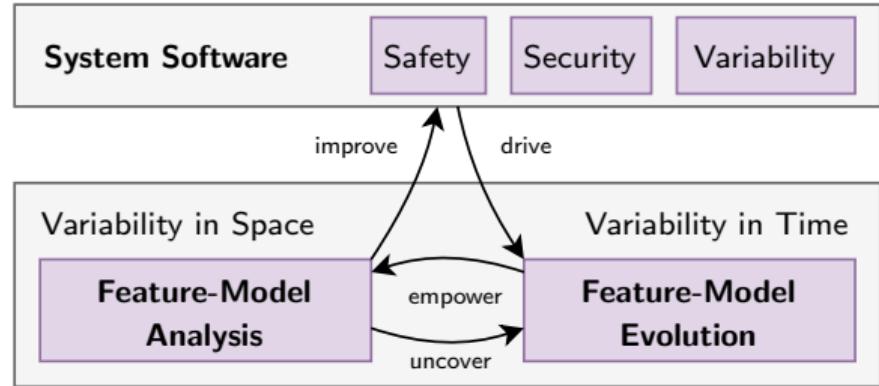
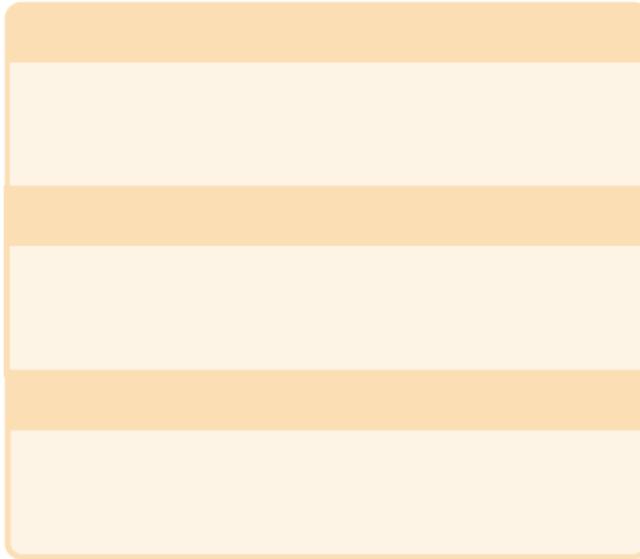
Motivation

Variability in Time

- most software **evolves** over time
- so do product lines
(long-term, open histories in system software)
- symbiosis: variability in space \Leftarrow time
(many field configurations in system software)



Effective and Efficient Feature-Model Analyses for Evolving System Software

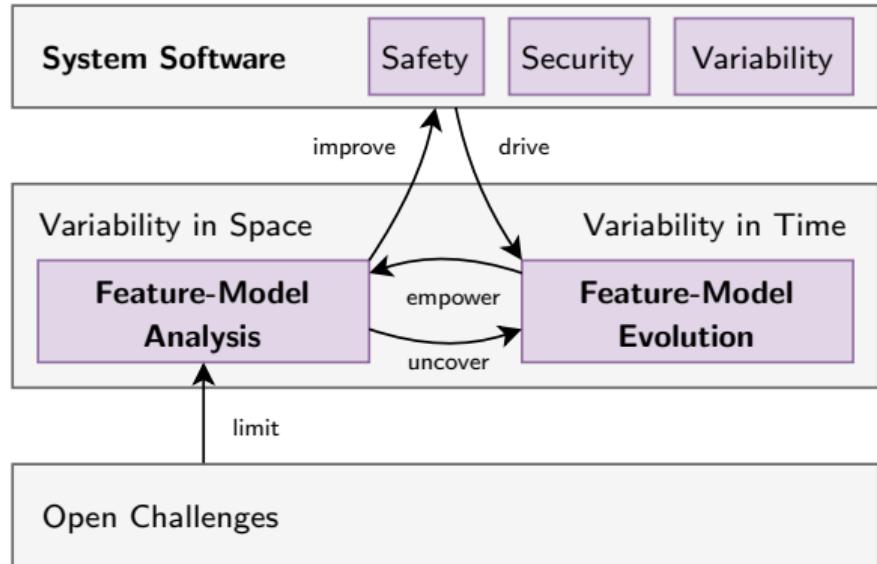


Effective and Efficient Feature-Model Analyses for Evolving System Software

Challenge:

Challenge:

Challenge:



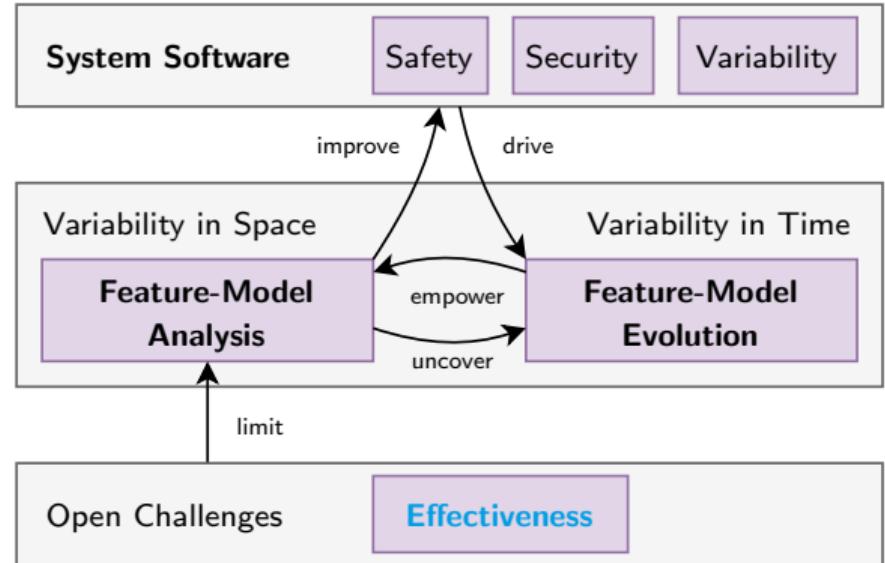
Effective and Efficient Feature-Model Analyses for Evolving System Software

Challenge: Effectiveness

- mandatory computation steps
⇒ may impact correctness + accuracy

Challenge:

Challenge:



Effective and Efficient Feature-Model Analyses for Evolving System Software

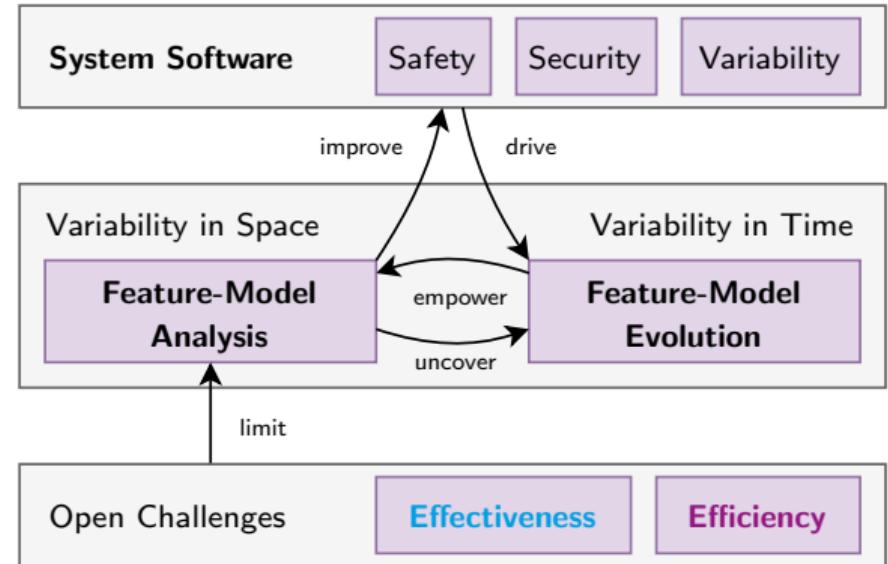
Challenge: Effectiveness

- mandatory computation steps
⇒ may impact correctness + accuracy

Challenge: Efficiency

- large and complex feature models
⇒ may impact scalability + performance

Challenge:



Effective and Efficient Feature-Model Analyses for Evolving System Software

Challenge: Effectiveness

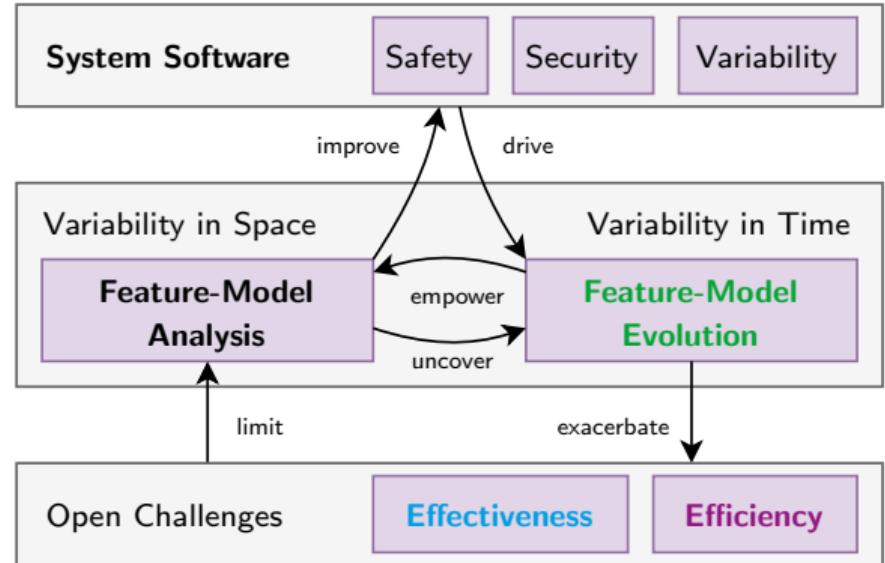
- mandatory computation steps
⇒ may impact correctness + accuracy

Challenge: Efficiency

- large and complex feature models
⇒ may impact scalability + performance

Challenge: Evolution

- variability grows, languages evolve
⇒ may impact feasibility



Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]

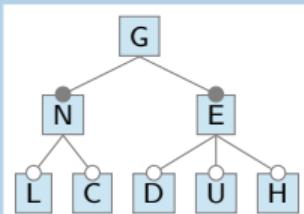
Motivation ► **Pipeline** ► Contributions ► Snapshots ► Evolution ► Timeline

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



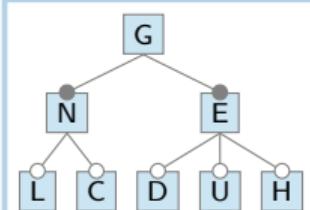
$$\begin{aligned}\neg(D \wedge U) \\ H \rightarrow U \\ D \not\leftrightarrow (U \wedge H)\end{aligned}$$

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



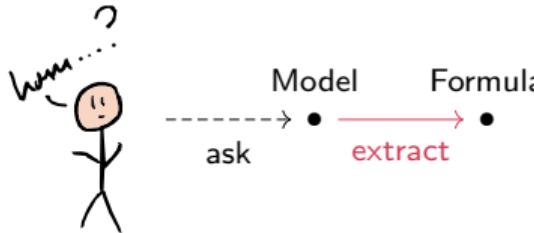
$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Core Feature F ?

#Configurations?

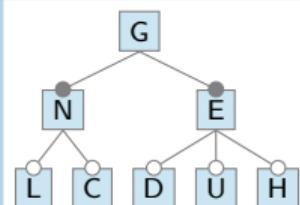
Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



→ interactive
→ automated

Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

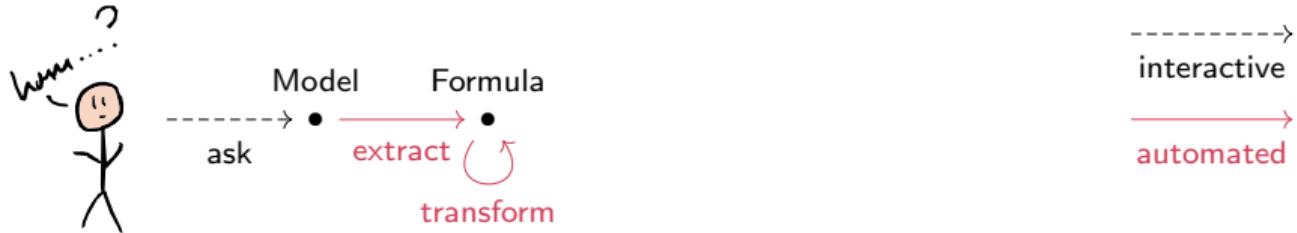
$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

Core Feature F ?

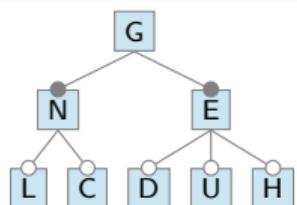
#Configurations?

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

CNF

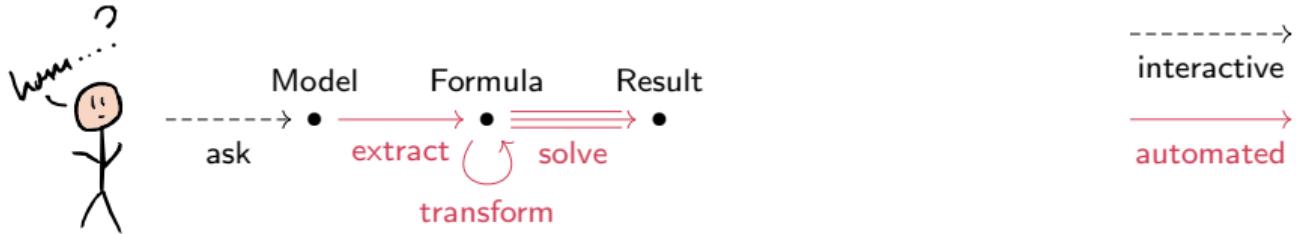
$$\begin{aligned} &\{\{G\}, \{\neg N, G\}, \\ &\{\neg N, \neg G\}, \{\neg E, G\}, \\ &\{E, \neg G\}, \{\neg L, N\}, \\ &\{\neg C, N\}, \{\neg D, E\}, \\ &\{\neg U, E\}, \{\neg H, E\}, \\ &\{\neg D, \neg U\}, \{\neg H, U\}, \\ &\{\{D, U\}, \{D, H\}, \\ &\{\neg D, \neg U, \neg H\}\} \} \end{aligned}$$

Core Feature F ?

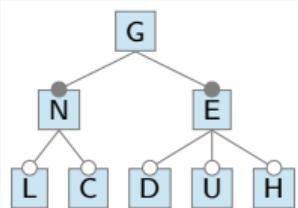
#Configurations?

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

CNF

$$\begin{aligned} &\{\{G\}, \{\neg N, G\}, \\ &\{\neg N, \neg G\}, \{\neg E, G\}, \\ &\{E, \neg G\}, \{\neg L, N\}, \\ &\{\neg C, N\}, \{\neg D, E\}, \\ &\{\neg U, E\}, \{\neg H, E\}, \\ &\{\neg D, \neg U\}, \{\neg H, U\}, \\ &\{\{D, U\}, \{D, H\}, \\ &\{\neg D, \neg U, \neg H\}\} \} \end{aligned}$$

Core Feature F ?

$$SAT(\Phi \wedge \neg F)$$

Core Features

$$\{G, N, E\}$$

#Configurations?

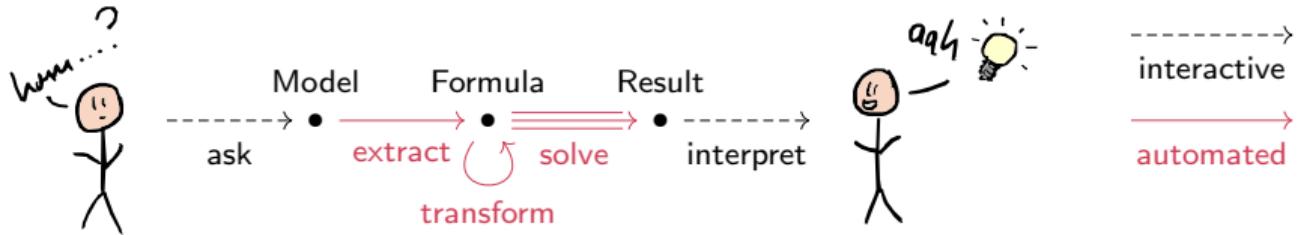
$$\#SAT(\Phi)$$

#Configurations

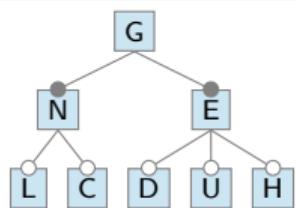
$$8$$

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

CNF

$$\begin{aligned} &\{\{G\}, \{\neg N, G\}, \\ &\{\neg N, \neg G\}, \{\neg E, G\}, \\ &\{E, \neg G\}, \{\neg L, N\}, \\ &\{\neg C, N\}, \{\neg D, E\}, \\ &\{\neg U, E\}, \{\neg H, E\}, \\ &\{\neg D, \neg U\}, \{\neg H, U\}, \\ &\{\{D, U\}, \{D, H\}, \\ &\{\neg D, \neg U, \neg H\}\} \} \end{aligned}$$

Core Feature F ?

$$SAT(\Phi \wedge \neg F)$$

Core Features

$$\{G, N, E\}$$

#Configurations?

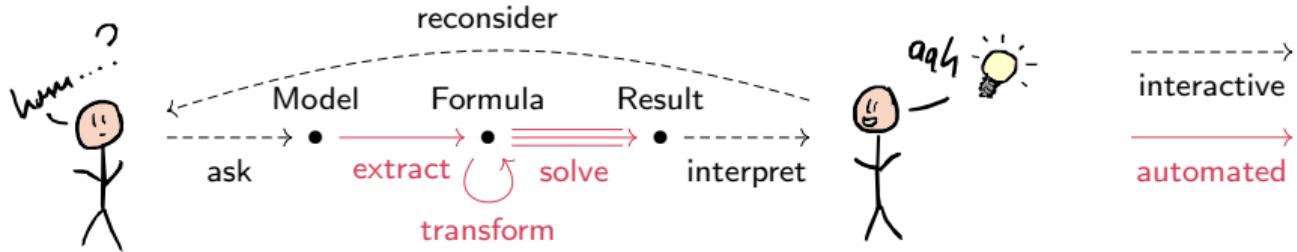
$$\#SAT(\Phi)$$

#Configurations

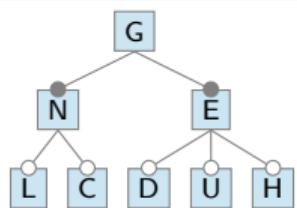
$$8$$

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

CNF

$$\begin{aligned} &\{\{G\}, \{\neg N, G\}, \\ &\{\neg N, \neg G\}, \{\neg E, G\}, \\ &\{E, \neg G\}, \{\neg L, N\}, \\ &\{\neg C, N\}, \{\neg D, E\}, \\ &\{\neg U, E\}, \{\neg H, E\}, \\ &\{\neg D, \neg U\}, \{\neg H, U\}, \\ &\{\{D, U\}, \{D, H\}, \\ &\{\neg D, \neg U, \neg H\}\} \} \end{aligned}$$

Core Feature F ?

$$SAT(\Phi \wedge \neg F)$$

Core Features

$$\{G, N, E\}$$

#Configurations?

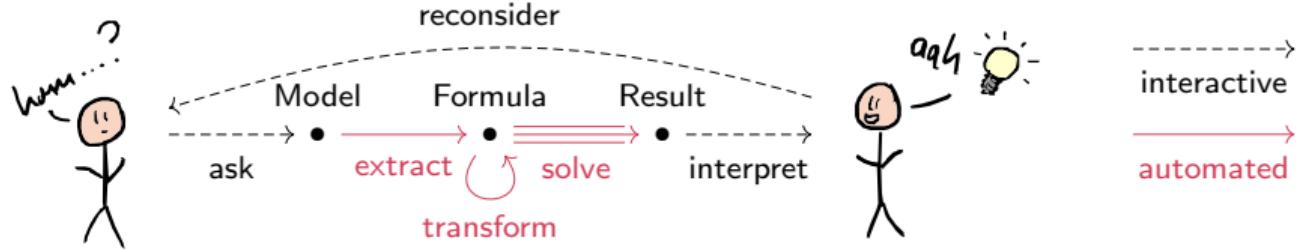
$$\#SAT(\Phi)$$

#Configurations

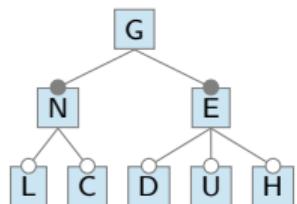
$$8$$

Analysis Pipeline

[Kuiter et al. '24, Sundermann et al. '24]



Feature Model



$$\begin{aligned} &\neg(D \wedge U) \\ &H \rightarrow U \\ &D \not\leftrightarrow (U \wedge H) \end{aligned}$$

Formula

$$\begin{aligned} &G \\ &\wedge (N \leftrightarrow G) \wedge (E \leftrightarrow G) \\ &\wedge ((L \vee C) \rightarrow N) \\ &\wedge ((D \vee U \vee H) \rightarrow E) \\ &\wedge \neg(D \wedge U) \wedge (H \rightarrow U) \\ &\wedge (D \not\leftrightarrow (U \wedge H)) \end{aligned}$$

CNF

$$\begin{aligned} &\{\{G\}, \{\neg N, G\}, \\ &\{\neg N, \neg G\}, \{\neg E, G\}, \\ &\{E, \neg G\}, \{\neg L, N\}, \\ &\{\neg C, N\}, \{\neg D, E\}, \\ &\{\neg U, E\}, \{\neg H, E\}, \\ &\{\neg D, \neg U\}, \{\neg H, U\}, \\ &\{\{D, U\}, \{D, H\}, \\ &\{\neg D, \neg U, \neg H\}\} \} \end{aligned}$$

Core Feature F ?

$$SAT(\Phi \wedge \neg F)$$

Core Features

$$\{G, N, E\}$$

#Configurations?

$$\#SAT(\Phi)$$

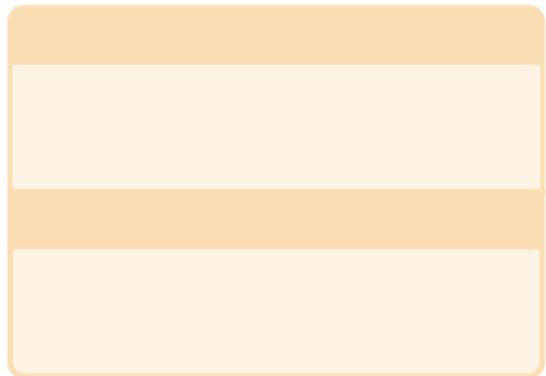
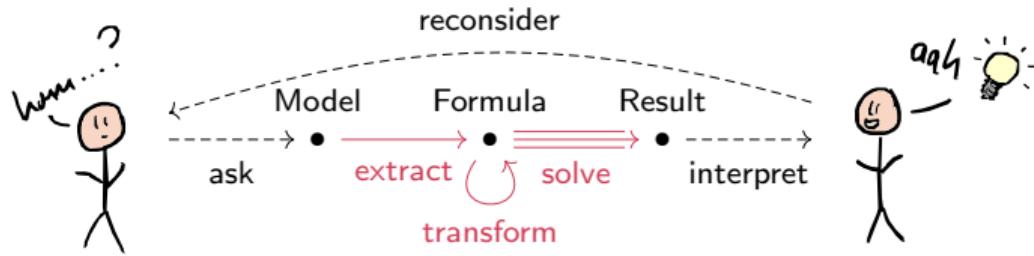
#Configurations

$$8$$

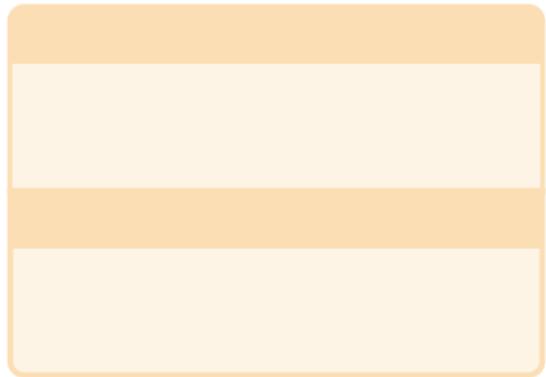
$\Rightarrow (\#)SAT$ enables advanced analyses

e.g., core/dead features, sampling, static analysis, verification, ...

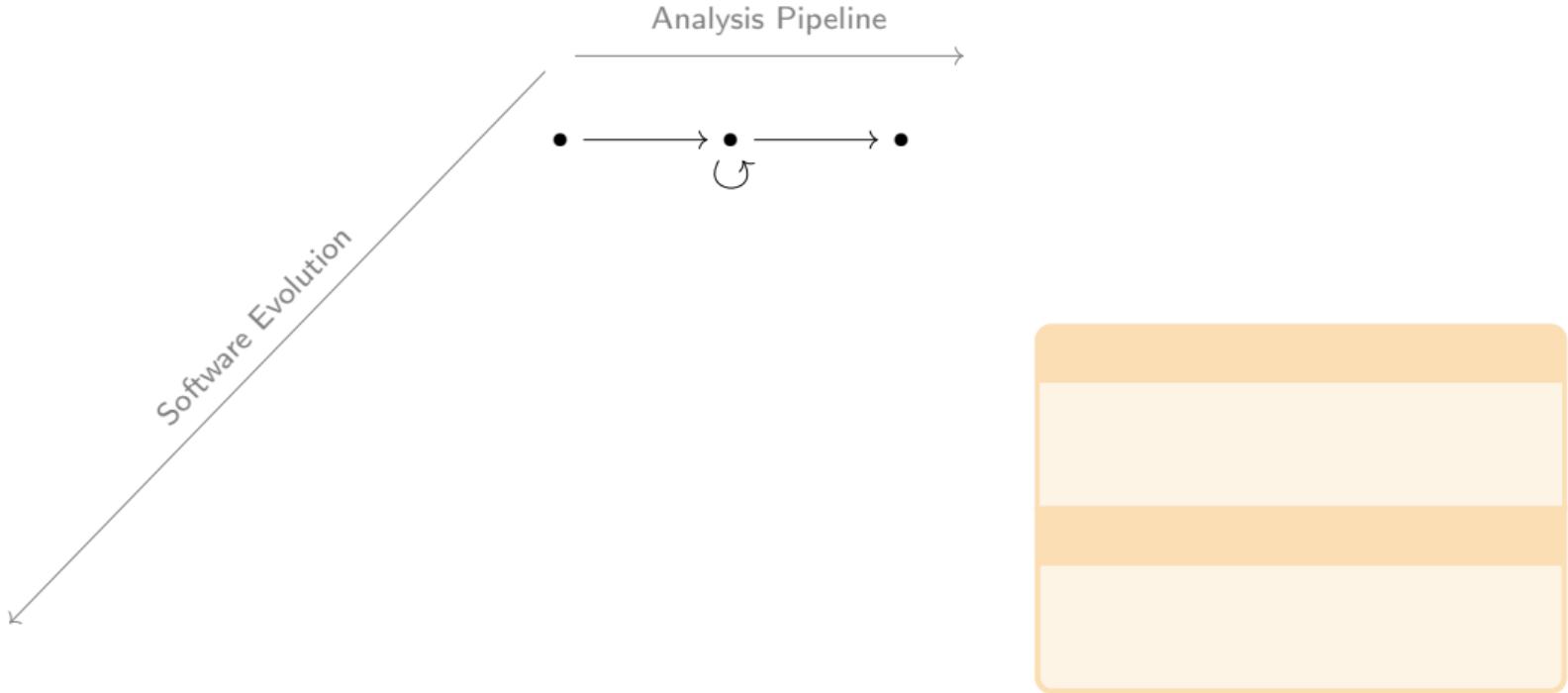
Contributions



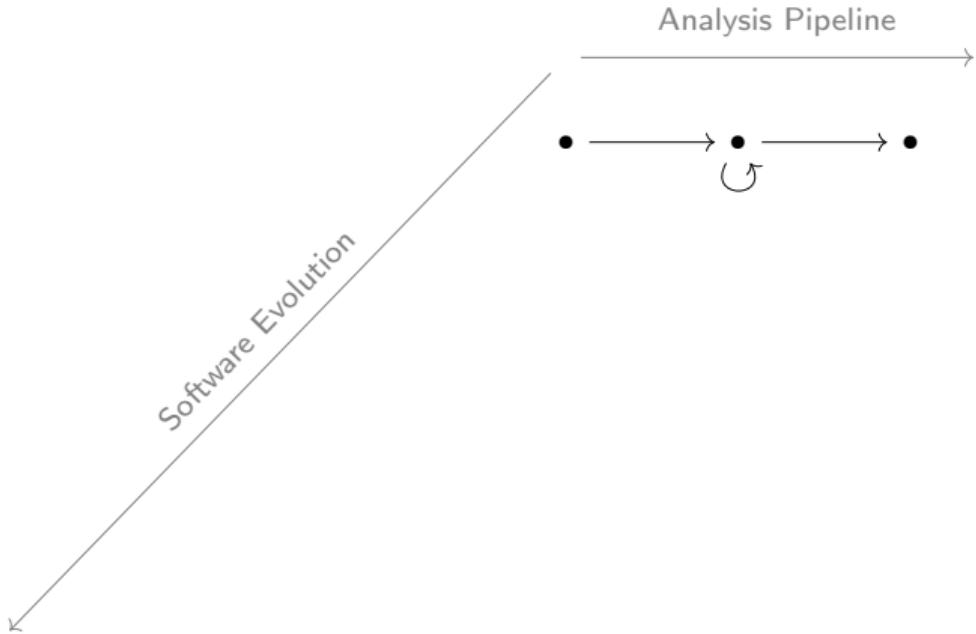
Contributions



Contributions

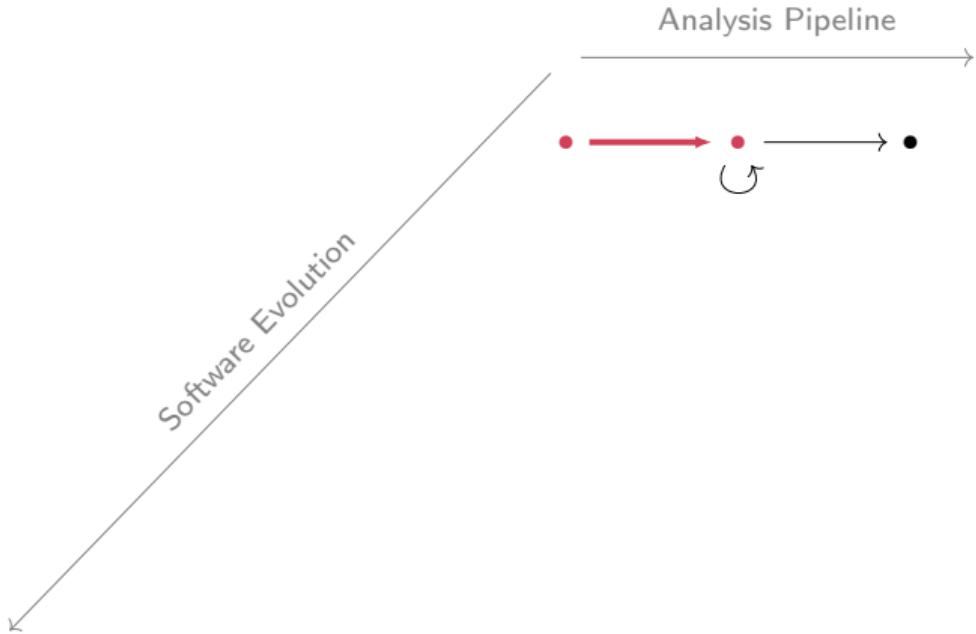


Contributions



Part I – Analyzing Snapshots

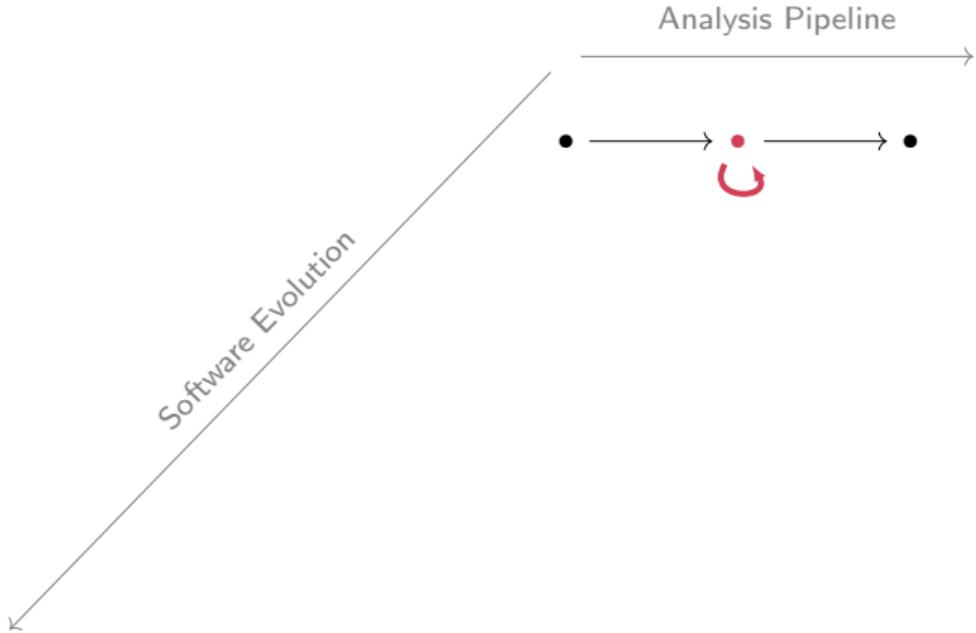
Contributions



Part I – Analyzing Snapshots

1. Extracting Feature Models

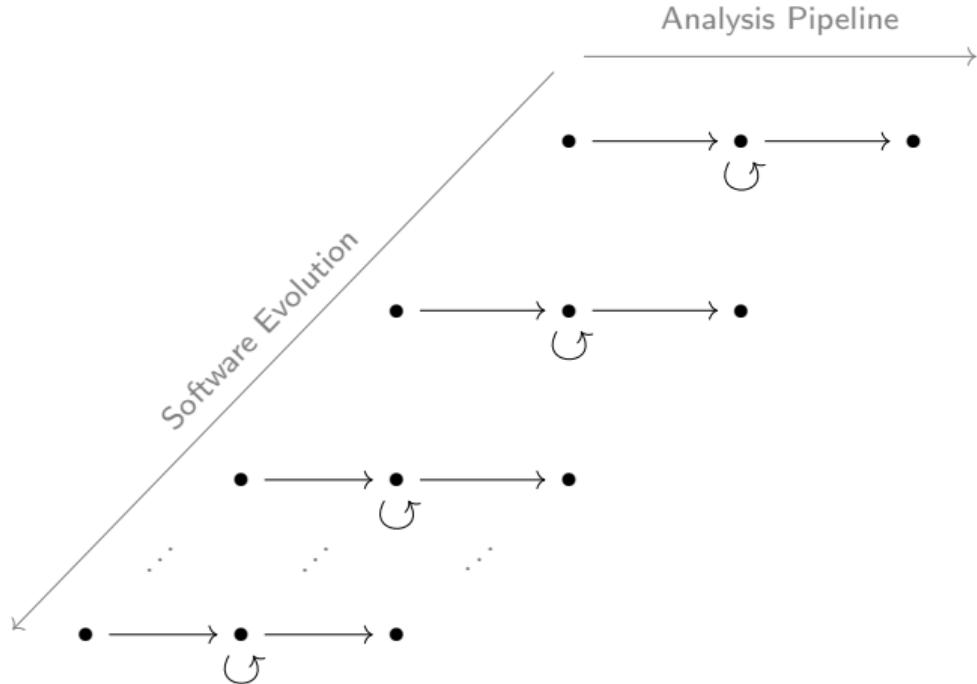
Contributions



Part I – Analyzing Snapshots

1. Extracting Feature Models
2. **Transforming Feature Models**

Contributions

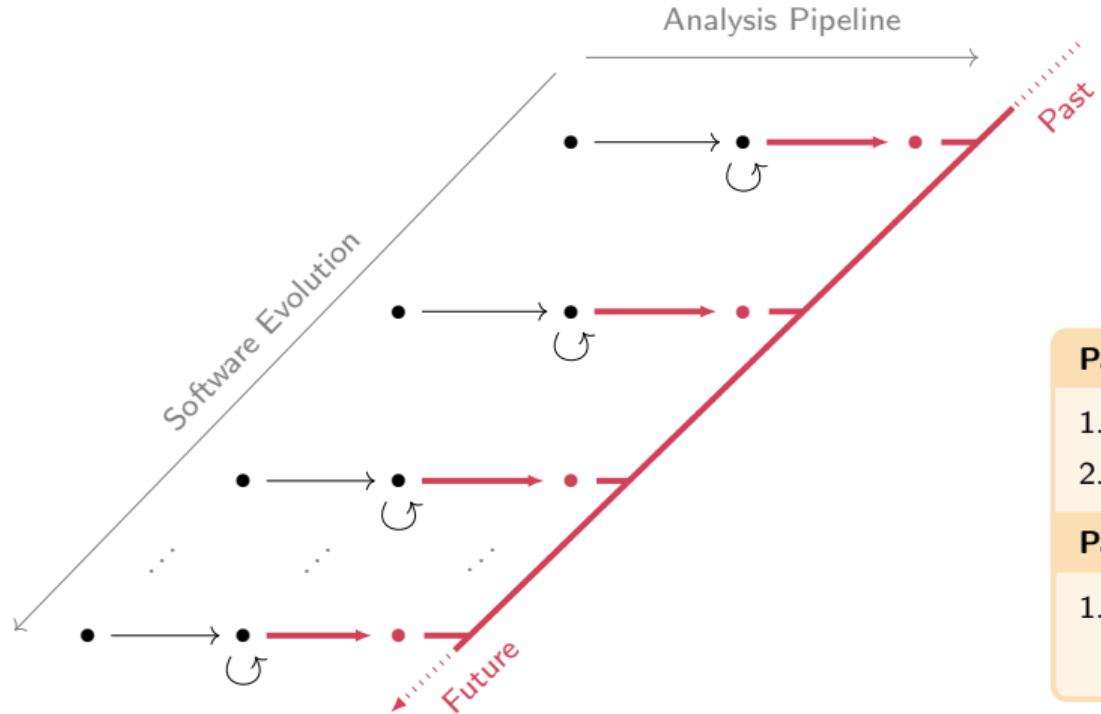


Part I – Analyzing Snapshots

1. Extracting Feature Models
2. Transforming Feature Models

Part II – Analyzing Evolution

Contributions



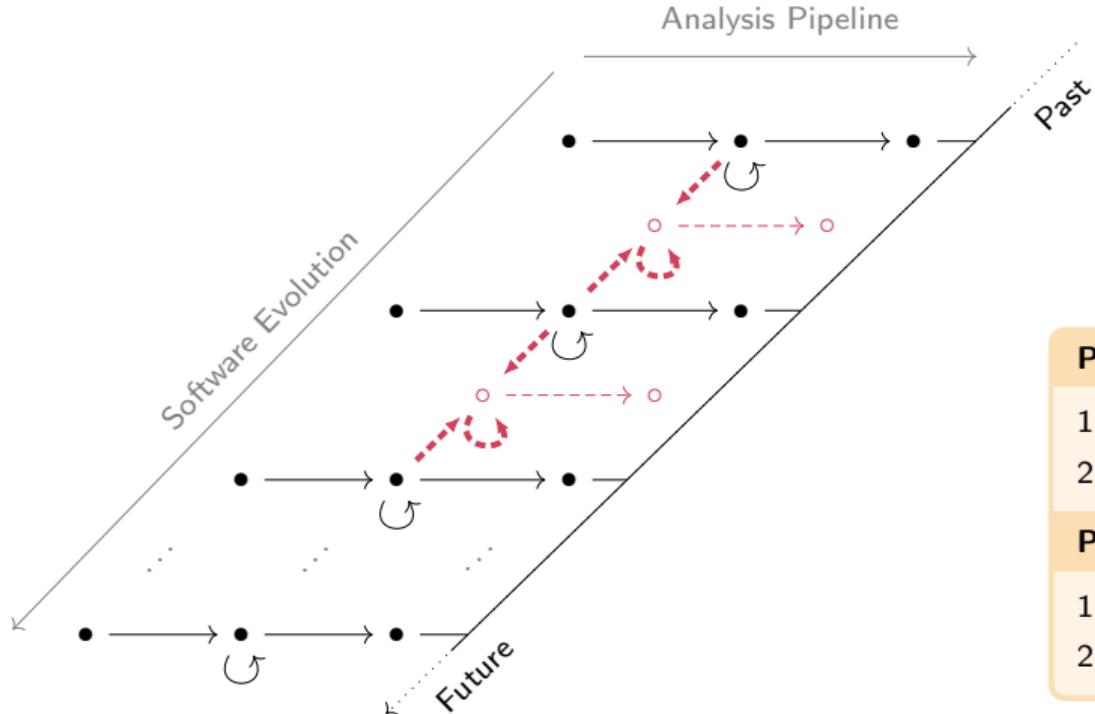
Part I – Analyzing Snapshots

1. Extracting Feature Models
2. Transforming Feature Models

Part II – Analyzing Evolution

1. Long-Term History

Contributions



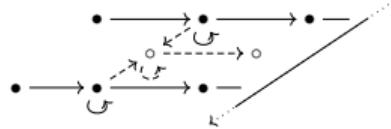
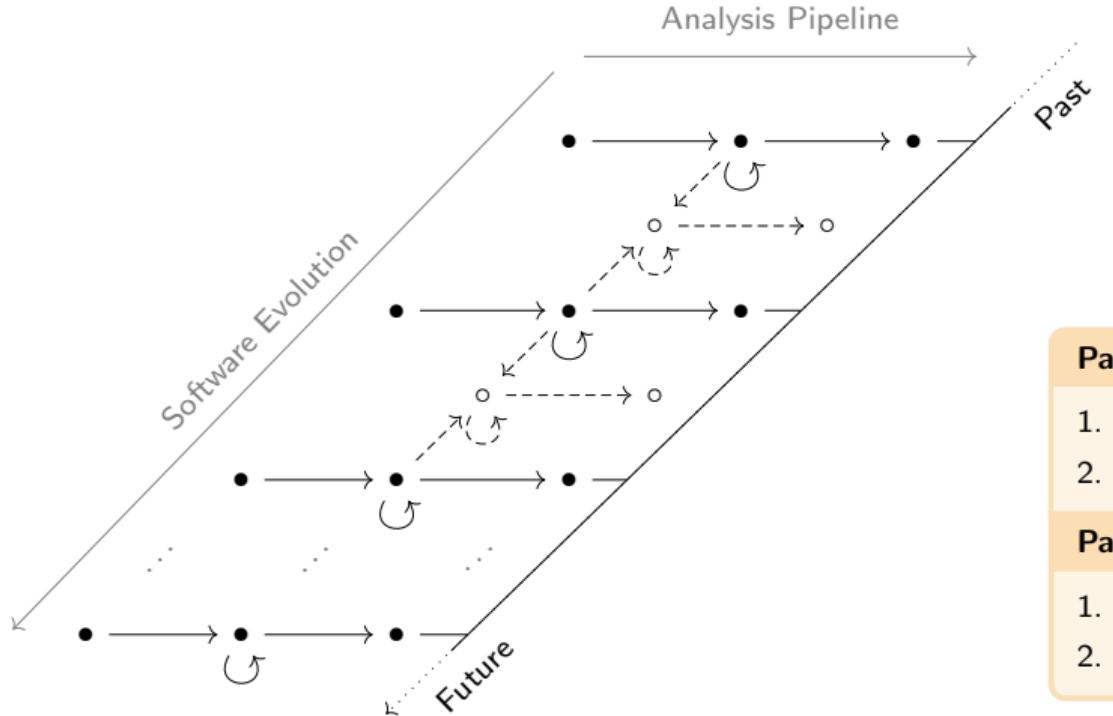
Part I – Analyzing Snapshots

1. Extracting Feature Models
2. Transforming Feature Models

Part II – Analyzing Evolution

1. Long-Term History
2. Individual Steps

Contributions



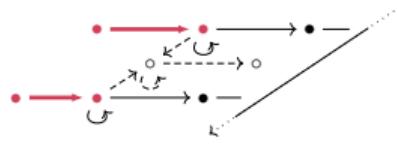
Part I – Analyzing Snapshots

1. Extracting Feature Models
2. Transforming Feature Models

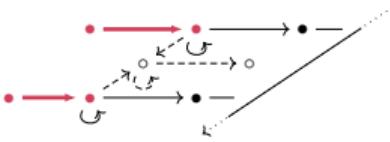
Part II – Analyzing Evolution

1. Long-Term History
2. Individual Steps

Analyzing Snapshots – Extracting Feature Models



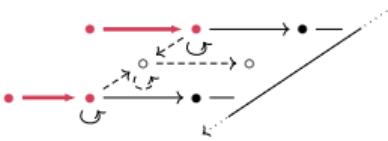
Analyzing Snapshots – Extracting Feature Models



What is **system software**?
Which **modeling languages** are used?
Which **feature models** to consider?



Analyzing Snapshots – Extracting Feature Models



What is **system software**?

Which **modeling languages** are used?
Which **feature models** to consider?

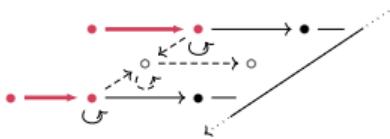


system software \simeq **KConfig**
KConfig evolves and **diverges**

[**todo**, Sundermann et al. '24, Karakaya '25, Dordevic '25]



Analyzing Snapshots – Extracting Feature Models



What is **system software**?

Which **modeling languages** are used?
Which **feature models** to consider?



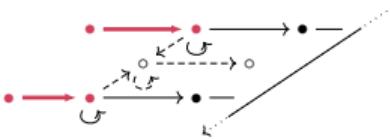
system software \simeq **KConfig**
KConfig **evolves** and **diverges**

[todo, Sundermann et al. '24, Karakaya '25, Dordevic '25]



LKC, KConfigLib, Standalone, ... \Rightarrow not unified

Analyzing Snapshots – Extracting Feature Models



What is **system software**?

Which **modeling languages** are used?
Which **feature models** to consider?



If KConfig is so messy, how
to **extract models** reliably?

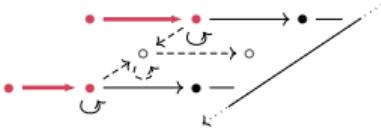
system software \simeq **KConfig**
KConfig **evolves** and **diverges**

[todo, Sundermann et al. '24, Karakaya '25, Dordevic '25]



LKC, KConfigLib, Standalone, ... \Rightarrow not unified

Analyzing Snapshots – Extracting Feature Models



What is **system software**?

Which **modeling languages** are used?
Which **feature models** to consider?



If KConfig is so messy, how
to **extract models** reliably?

system software \simeq **KConfig**
KConfig evolves and **diverges**

[todo, Sundermann et al. '24, Karakaya '25, Dordevic '25]



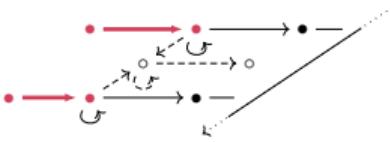
torte = KConfig extractor
that **tackles evolution**

[todo, Alfish '25, Ketzler '25]



LKC, KConfigLib, Standalone, ... \Rightarrow not unified

Analyzing Snapshots – Extracting Feature Models

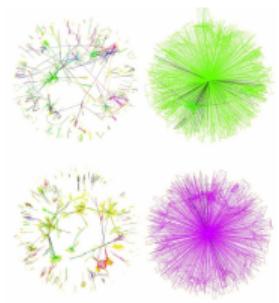


What is **system software**?

Which **modeling languages** are used?
Which **feature models** to consider?



If KConfig is so messy, how
to **extract models** reliably?



system software \simeq **KConfig**
KConfig evolves and **diverges**

[todo, Sundermann et al. '24, Karakaya '25, Dordevic '25]



torte = KConfig extractor
that **tackles evolution**

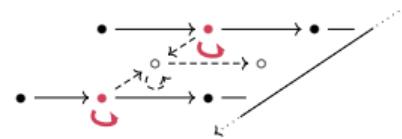


[todo, Alfish '25, Ketzler '25]

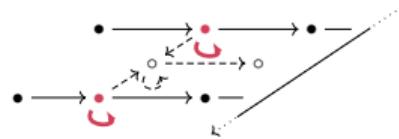
LKC, KConfigLib, Standalone, ... \Rightarrow not unified

3 extractors, **full histories**, hierarchies, UVL, ...

Analyzing Snapshots – Transforming Feature Models



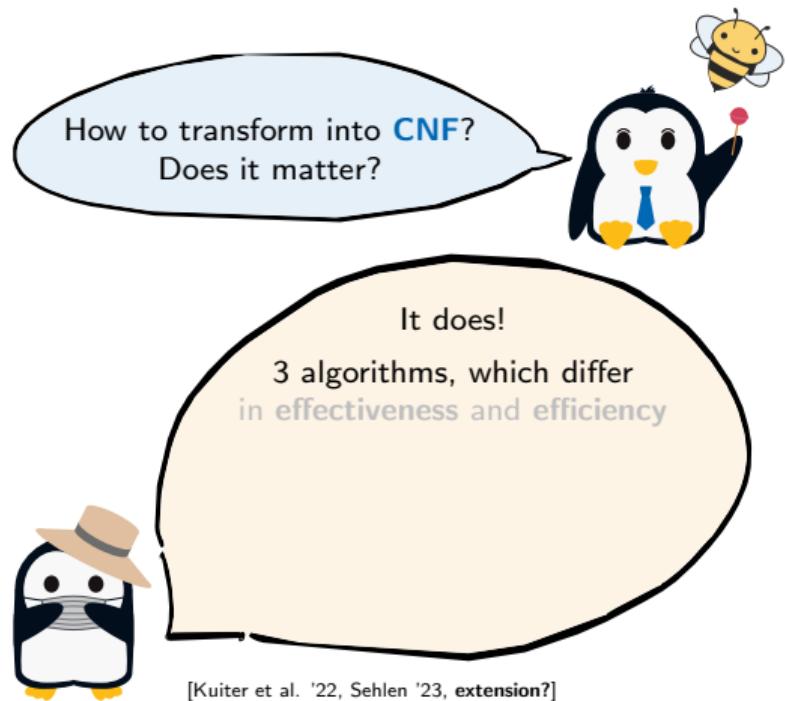
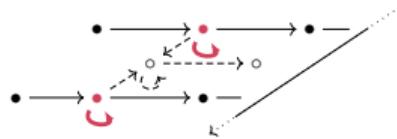
Analyzing Snapshots – Transforming Feature Models



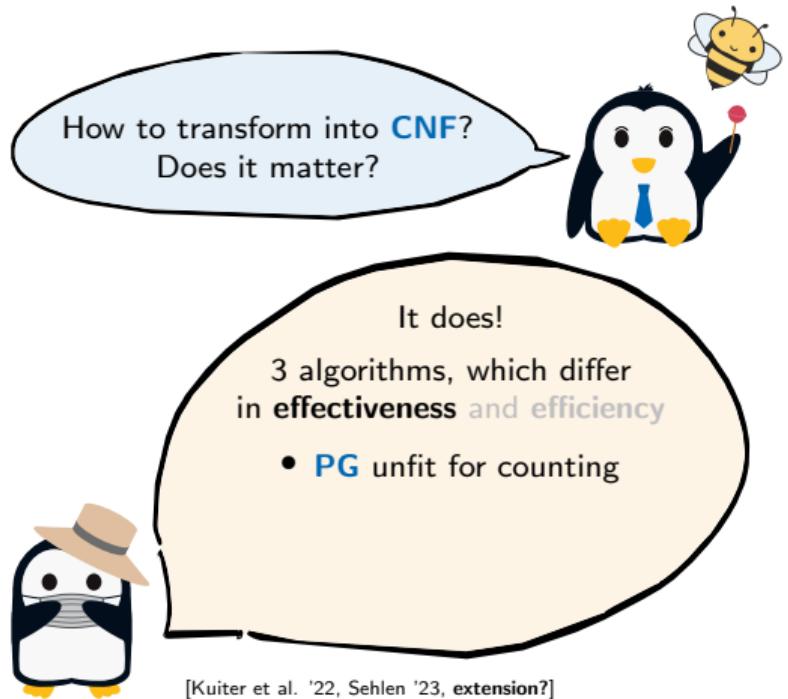
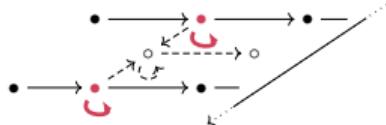
How to transform into **CNF**?
Does it matter?



Analyzing Snapshots – Transforming Feature Models



Analyzing Snapshots – Transforming Feature Models



How Many Configurations in BusyBox 1.35.0?

FeatureIDE (Distributive) says:

4784204604 4873008384 1351764949 6919484532 1798073727 ... 000000000

Tseitin (Z3) says:

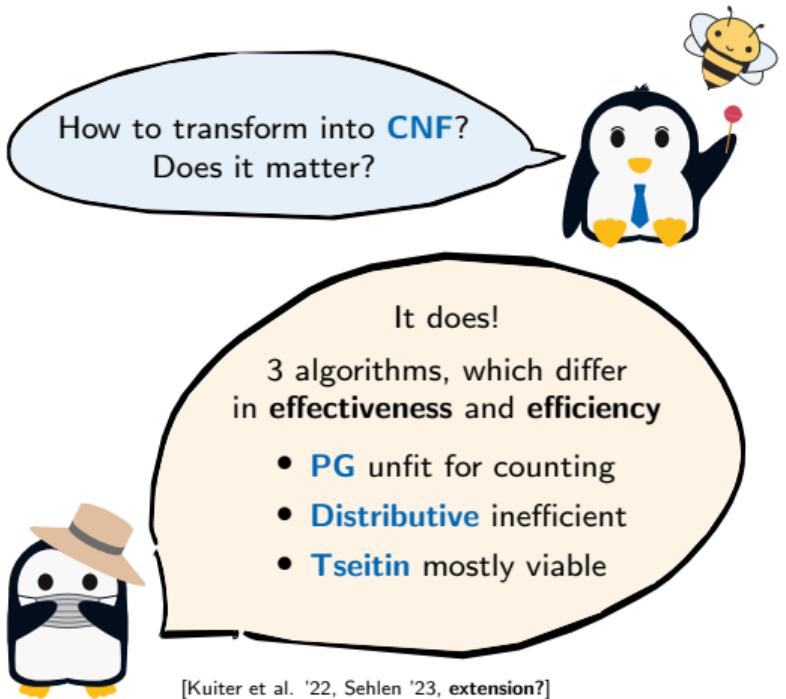
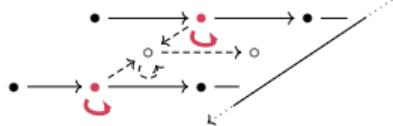
4784204604 4873008384 1351764949 6919484532 1798073727 ... 000000000

KConfigReader (Plaisted-Greenbaum) says:

1575135744 6718468213 9013565599 6554596226 7796564828 ... 000000000 0

⇒ off by factor 3.292

Analyzing Snapshots – Transforming Feature Models



How Many Configurations in BusyBox 1.35.0?

FeatureIDE (Distributive) says:

4784204604 4873008384 1351764949 6919484532 1798073727 ... 000000000

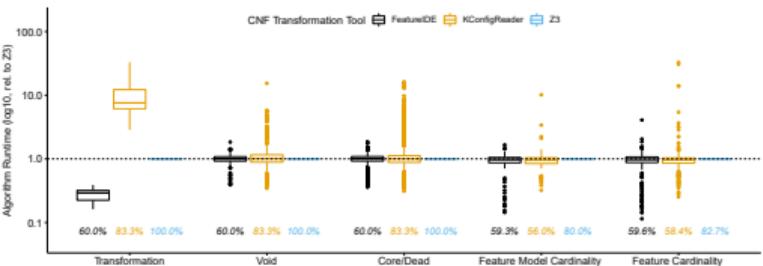
Tseitin (Z3) says:

4784204604 4873008384 1351764949 6919484532 1798073727 ... 000000000

KConfigReader (Plaisted-Greenbaum) says:

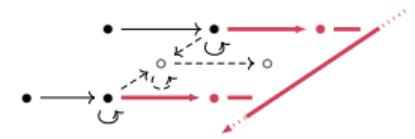
1575135744 6718468213 9013565599 6554596226 7796564828 ... 000000000 0

⇒ off by factor 3.292



Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



#features varies wildly
#configurations unknown
computationally challenging!



[Kuiter et al. '25, extension?]

Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



#**features** varies wildly
#**configurations** unknown
computationally challenging!

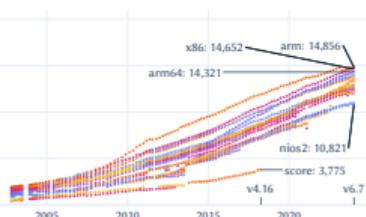


[Kuiter et al. '25, extension?]

today: 20k features, $10^{1.6k} - 10^{3.6k}$ configurations

Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



Can **SAT solvers** keep up with
the Linux kernel's feature model?

#**features** varies wildly
#**configurations** unknown
computationally challenging!



[Kuiter et al. '25, extension?]

today: 20k features, $10^{1.6k} - 10^{3.6k}$ configurations

Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



Can **SAT solvers** keep up with
the Linux kernel's feature model?

#**features** varies wildly
#**configurations** unknown
computationally challenging!



[Kuiter et al. '25, extension?]

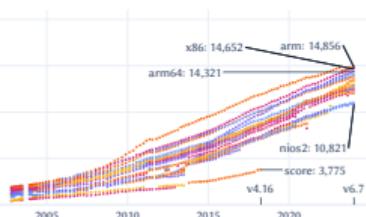
No, not really.

[under review, Braun '25]

today: 20k features, $10^{1.6k} - 10^{3.6k}$ configurations

Analyzing Evolution – Long-Term History

Case Study: Linux Kernel



How **configurable** is the Linux kernel?
How to **measure** configurability?



Can **SAT solvers** keep up with
the Linux kernel's feature model?

#**features** varies wildly
#**configurations** unknown
computationally challenging!



No, not really.

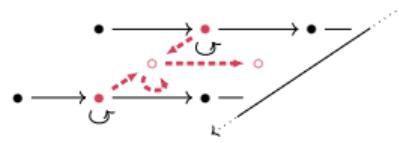
[Kuiter et al. '25, extension?]



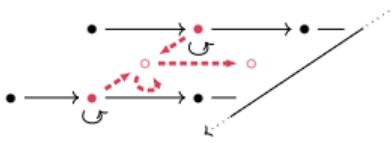
today: 20k features, $10^{1.6k} - 10^{3.6k}$ configurations

every year, SAT solvers **slow down** by 13%–29%

Analyzing Evolution – Individual Steps

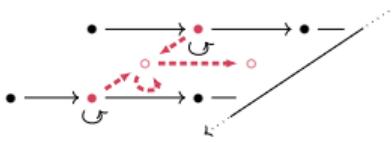


Analyzing Evolution – Individual Steps



What's the **difference**?
Is it **backward compatible**?

Analyzing Evolution – Individual Steps



What's the **difference**?
Is it **backward compatible**?

Tseitin + #SAT + slicing!

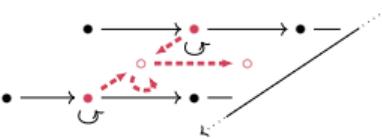
for measuring inadvertent

- variability growth
- variability reduction

[todo, see FOSD '25]



Analyzing Evolution – Individual Steps



What's the **difference**?
Is it **backward compatible**?

Tseitin + #SAT + slicing!

for measuring inadvertent

- variability growth
- variability reduction

[todo, see FOSD '25]



Introducing #SAT-Based Reasoning

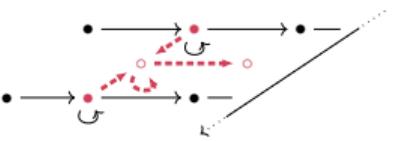
the degree of removed configurations is

$$(\text{none}) \quad 0 \leq \frac{\#SAT(\theta_T(\phi \wedge \neg\psi))}{\#SAT(\theta_T(\phi))} \leq 1 \quad (\text{all})$$



we can now **quantify** the degree of generalization!

Analyzing Evolution – Individual Steps



What's the **difference**?
Is it **backward compatible**?

Tseitin + #SAT + slicing!

for measuring inadvertent

- variability growth
- variability reduction

[todo, see FOSD '25]



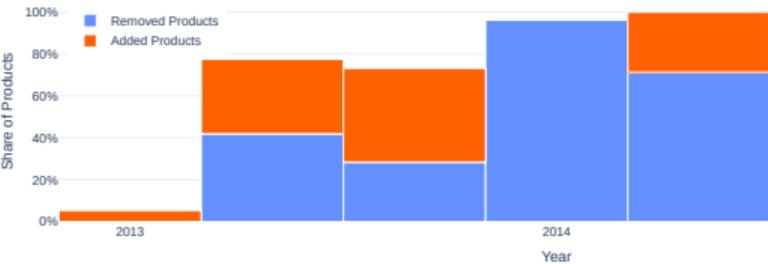
Introducing #SAT-Based Reasoning

the degree of removed configurations is

$$(\text{none}) \quad 0 \leq \frac{\#\text{SAT}(\theta_T(\phi \wedge \neg\psi))}{\#\text{SAT}(\theta_T(\phi))} \leq 1 \quad (\text{all})$$



we can now **quantify** the degree of generalization!



Timeline and Novelty

Chapter	Papers
Motivation + Analysis Pipeline	VaMoS '24, SPLC '25
Analyzing Snapshots – Extracting Feature Models	2 planned
Analyzing Snapshots – Transforming Feature Models	ASE '22 + 1?
Analyzing Evolution – Long-Term History	TOSEM '25 + 1 submitted
Analyzing Evolution – Individual Steps	1 planned

Batory '05

SPLC'17 test of time award

but: CNF is mandatory, yet often overlooked

She et al. '10

VaMoS'20 MIP award

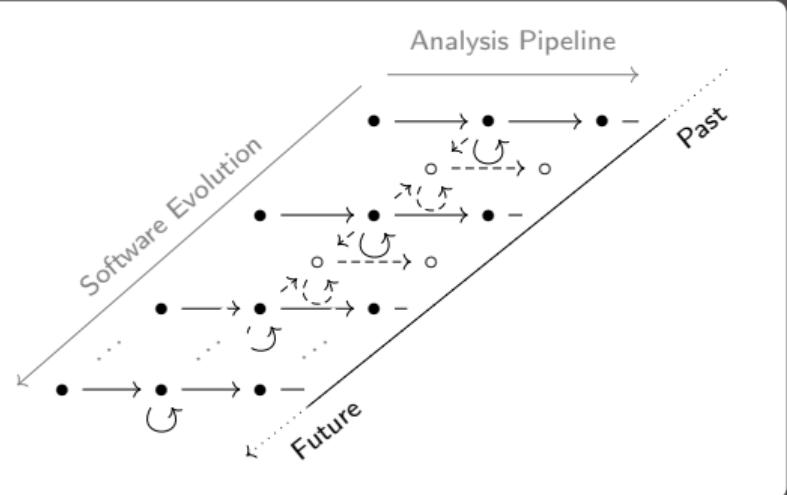
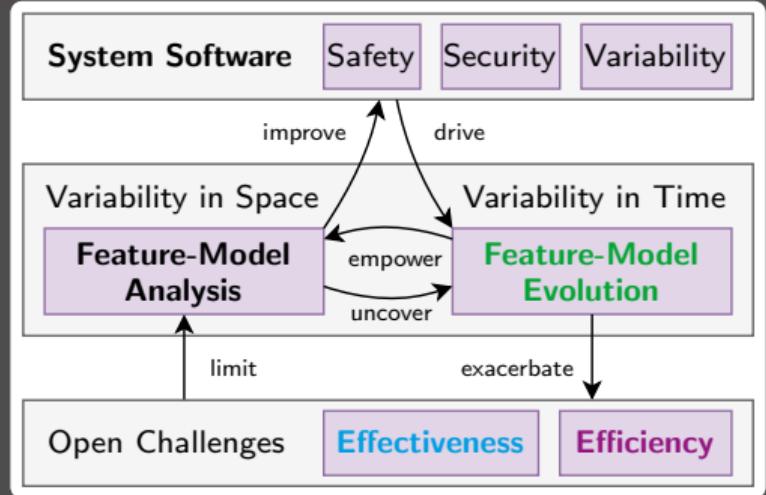
but: configurability not well-defined, inconsistently reported

Thüm et al. '09

SPLC'23 MIP award

but: current differencing impractical and coarse-grained

Conclusion



Thank you for listening!