



[Elbe, Magdeburg]

Quantified Reasoning About Edits to Feature Models

FOSD 2025 — March 25–28 — Köthen

Elias Kuiter¹, Thomas Thüm², Gunter Saake¹

University of Magdeburg¹, TU Braunschweig²

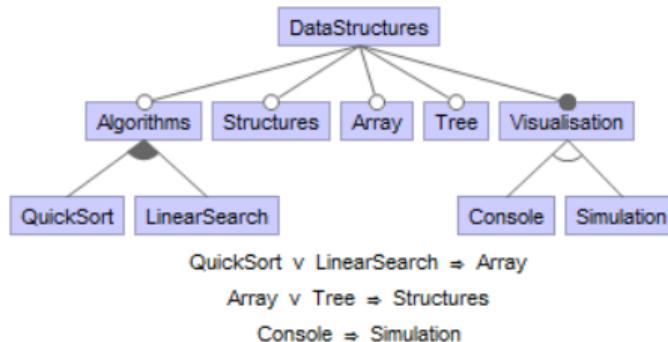
Quantified Reasoning About Edits to Feature Models

Implementierungstechniken für Software-Produktlinien

Übung 10: Analyse von Produktlinien

1. Feature-Modell-Analyse

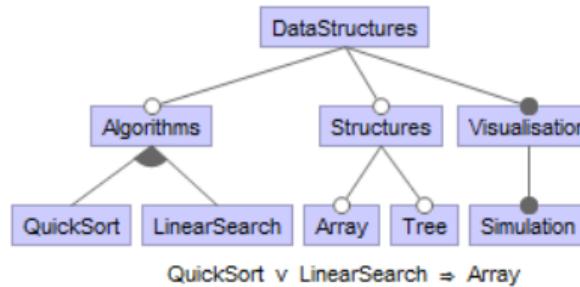
Gegeben sei das folgende Feature-Modell FM .



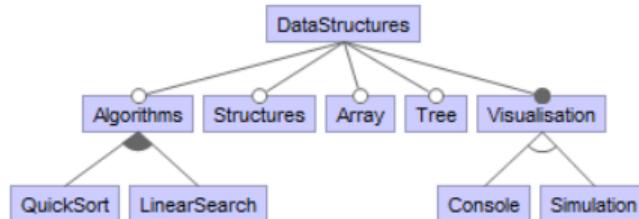
Quantified Reasoning About Edits to Feature Models

2. Evolution von Feature-Modellen

- (a) Welche semantischen Änderungen an Feature-Modellen können vorgenommen werden?
- (b) Gegeben sei das folgende Feature-Modell FM' . Welche Änderung gegenüber dem obigen Modell FM wurden vorgenommen? Was bringen diese Änderungen?



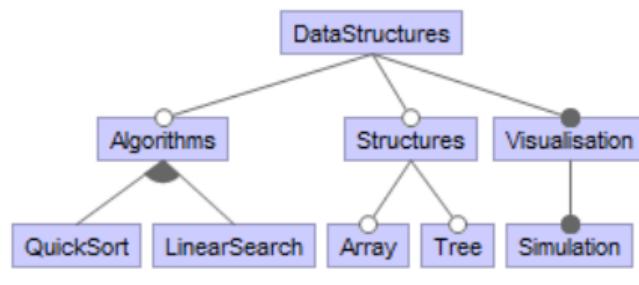
Quantified Reasoning About Edits to Feature Models



$\text{QuickSort} \vee \text{LinearSearch} \Rightarrow \text{Array}$

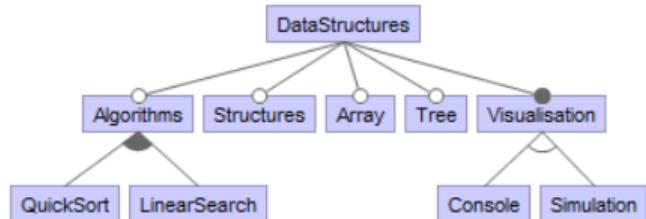
$\text{Array} \vee \text{Tree} \Rightarrow \text{Structures}$

$\text{Console} \Rightarrow \text{Simulation}$



$\text{QuickSort} \vee \text{LinearSearch} \Rightarrow \text{Array}$

Quantified Reasoning About Edits to Feature Models

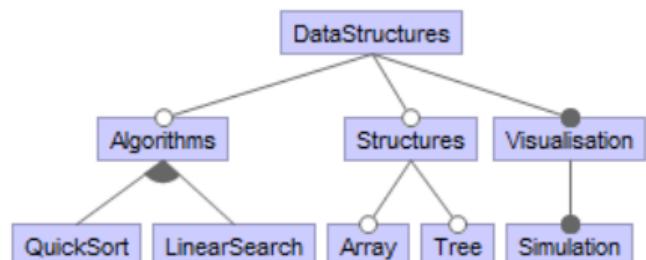


$\text{QuickSort} \vee \text{LinearSearch} \Rightarrow \text{Array}$

$\text{Array} \vee \text{Tree} \Rightarrow \text{Structures}$

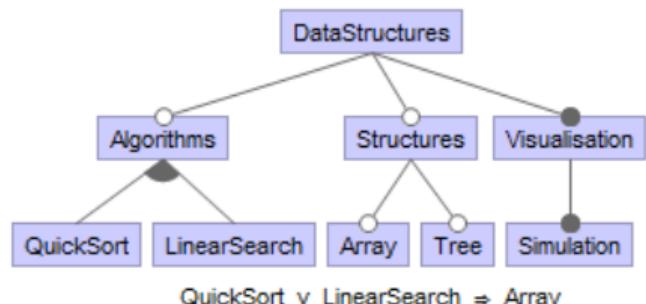
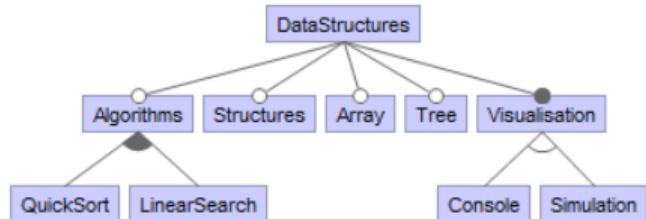
$\text{Console} \Rightarrow \text{Simulation}$

What Happens to the Configuration Space?



$\text{QuickSort} \vee \text{LinearSearch} \Rightarrow \text{Array}$

Quantified Reasoning About Edits to Feature Models



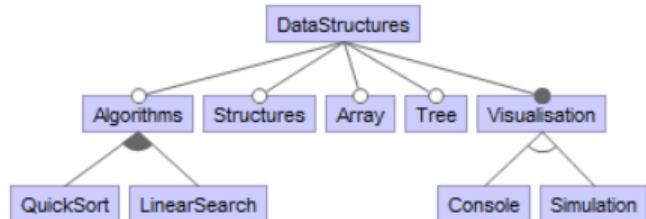
What Happens to the Configuration Space?

No Products Added	Products Added
-------------------	----------------

No Products Deleted

Products Deleted

Quantified Reasoning About Edits to Feature Models

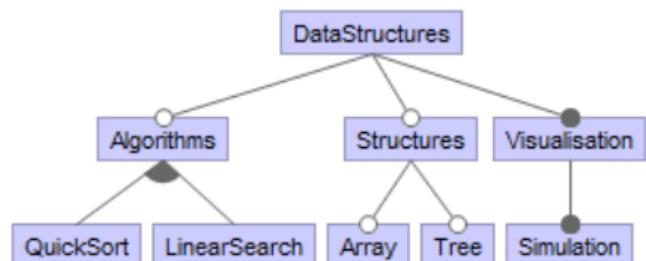


QuickSort v LinearSearch \Rightarrow Array

Array v Tree \Rightarrow Structures

Console \Rightarrow Simulation

?



QuickSort v LinearSearch \Rightarrow Array

What Happens to the Configuration Space?

No Products
Added

Products
Added

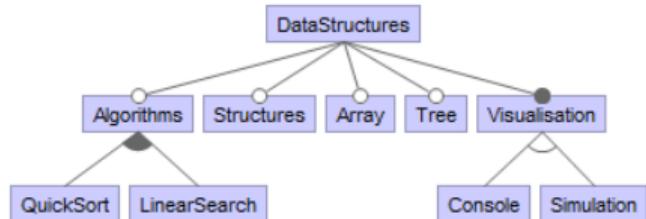


No Products
Deleted

Refactoring

Products
Deleted

Quantified Reasoning About Edits to Feature Models

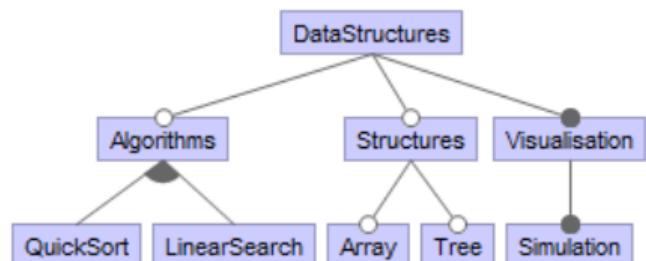


QuickSort v LinearSearch \Rightarrow Array

Array v Tree \Rightarrow Structures

Console \Rightarrow Simulation

? = ?
C



QuickSort v LinearSearch \Rightarrow Array

What Happens to the Configuration Space?

No Products
Added

Products
Added



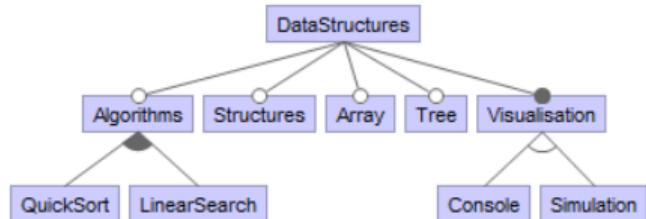
No Products
Deleted

Refactoring

Generalization

Products
Deleted

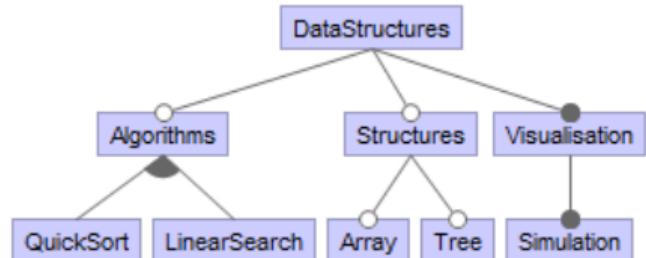
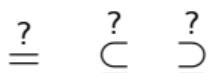
Quantified Reasoning About Edits to Feature Models



QuickSort v LinearSearch \Rightarrow Array

Array v Tree \Rightarrow Structures

Console \Rightarrow Simulation

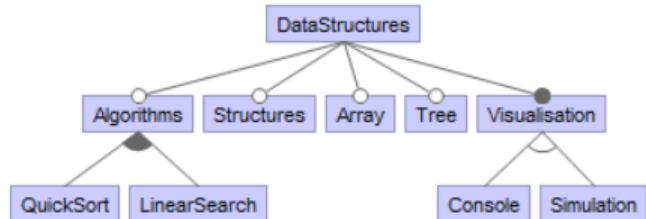


QuickSort v LinearSearch \Rightarrow Array

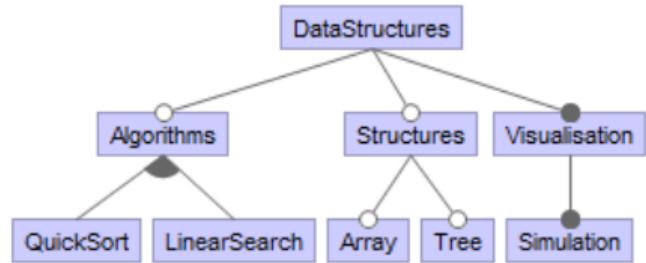
What Happens to the Configuration Space?

No Products Added	Products Added	
No Products Deleted	Refactoring	Generalization
Products Deleted	Specialization	

Quantified Reasoning About Edits to Feature Models



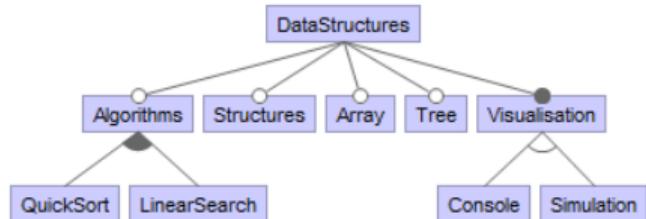
$$= \quad ? \subseteq \quad ? \supseteq \quad \cap = ? = \emptyset$$



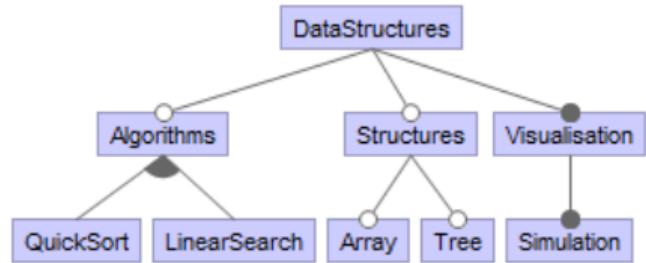
What Happens to the Configuration Space?

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

Quantified Reasoning About Edits to Feature Models



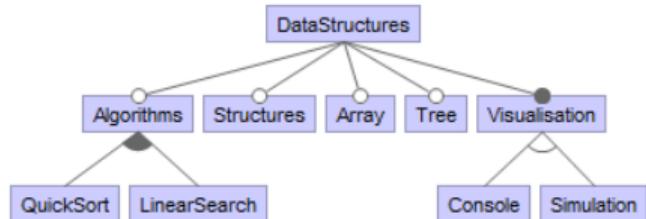
$$= \quad ? \subseteq \quad ? \supseteq \quad \cap = ? = \emptyset$$



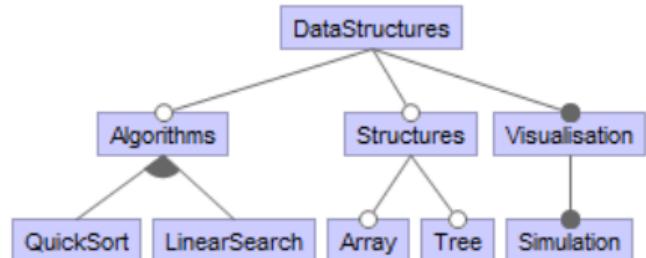
What Happens to the Configuration Space?

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

Quantified Reasoning About Edits to Feature Models



$$= \quad ? \subseteq \quad ? \supseteq \quad \cap = ? = \emptyset$$



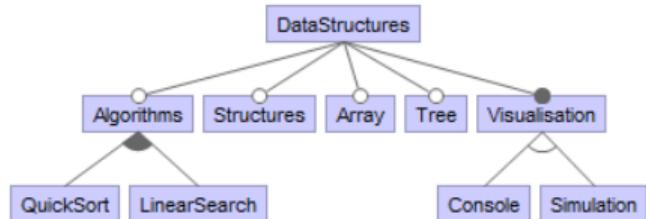
QuickSort v LinearSearch \Rightarrow Array

Feature-Model Edits

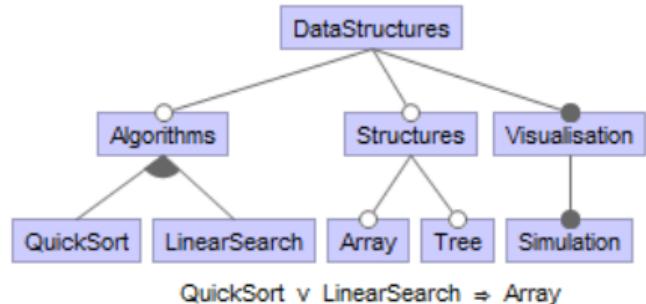
[Thüm et al. '09]

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

Quantified Reasoning About Edits to Feature Models

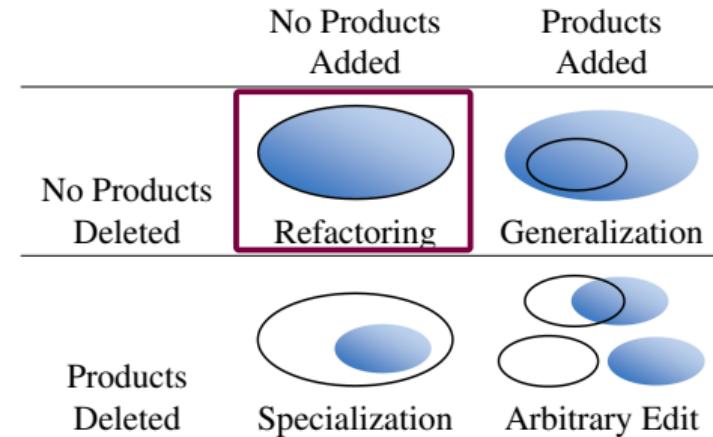


$$= \quad ? \subseteq ? \supseteq ? \cap ? = \emptyset$$



Feature-Model Edits

[Thüm et al. '09]



- goal**: compare versions of a feature model
- use cases**: e.g., to avoid unintentional changes, understand patterns in evolution, or support continuous integration \Rightarrow **quality assurance**

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes 

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes 
- **solution:** split into many smaller SAT calls

Quantified Reasoning About Edits to Feature Models

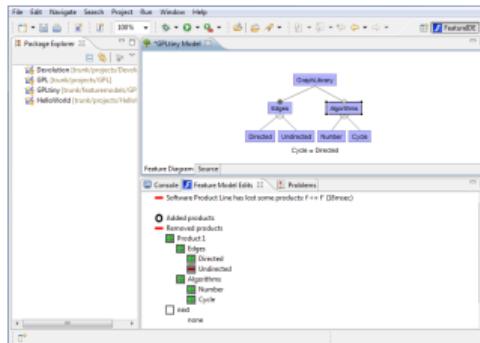
SAT-Based: Simplified Reasoning

[Thüm et al., '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
 - **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes 
 - **solution:** split into many smaller SAT calls



Implemented in FeatureIDE



Formal tool demonstration tomorrow at 4:30pm
Available open source at <http://www.fosd.de/featureide>



Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg\text{SAT}(\phi \wedge \neg\psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\text{SAT}(\psi \wedge \neg\phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg\psi$ is large and explodes \bullet^*
- **solution:** split into many smaller SAT calls

Weaknesses

let's see ...

Implemented in FeatureIDE

The screenshot shows the FeatureIDE interface. On the left, the 'Package Explorer' view displays several projects: 'Description Translations/DSL', 'GR4jy (trunk/features/DSL)', and 'HelloWorld (trunk/projects/Hello)'. On the right, the 'Feature Diagram' view shows a tree structure with 'Requirements' at the root, which branches into 'Scope' and 'Implementation'. 'Scope' further branches into 'Added', 'Unchanged', 'Removed', and 'Cycle'. Below the tree, there is a status bar message: 'Software Product Line has lost some products: F <= F (Dense)'. The bottom status bar also shows 'Added products', 'Removed products', and a list of items: 'Product1', 'Type', 'Unchanged', 'Algorithms', 'Number', 'Cycle', 'Add', and 'None'. A blue banner at the bottom of the slide contains the text: 'Formal tool demonstration tomorrow at 4:30pm Available open source at <http://www.fosd.de/featureide>'.

Formal tool demonstration tomorrow at 4:30pm
Available open source at <http://www.fosd.de/featureide>

f IDE
feature

Quantified Reasoning About Edits to Feature Models

Remarks on Significance (Only 1 Day Old)

Tobias Heß in Home Office

"Hey Thomas,

I have good news!

I checked every commit in the history of Busybox, Fiasco, Soletta, uclibc, Toybox, and FinancialServices.

We can **significantly** simplify your classification algorithm, as 100 % of the changes are arbitrary edits:"

The New Algorithm

```
String classify(FM f, FM g) {  
    return "Arbitrary Edit";  
}
```

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

implemented in  fIDE feature

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 😬

implemented in  fIDE feature

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes 
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$

implemented in  fIDE feature

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes 
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  fIDE
feature

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  f feature IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

Weaknesses

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  f feature IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model

Weaknesses

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤯
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  f feature IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)

Weaknesses

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤯
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  f feature IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.
- requires a BDD, which currently does not scale to very large feature models

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.
- requires a BDD, which currently does not scale to very large feature models
- no empirical evaluation

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  IDE

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.
- requires a BDD, which currently does not scale to very large feature models
- no empirical evaluation
- assumes added and removed features to be dead

implemented in  FAMILIAR

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  IDE
feature

SAT-Based (efficient, but coarse-grained)

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.
- requires a BDD, which currently does not scale to very large feature models
- no empirical evaluation
- assumes added and removed features to be dead

implemented in  FAMILIAR

BDD-Based (fine-grained, but inefficient)

Quantified Reasoning About Edits to Feature Models

SAT-Based: Simplified Reasoning

[Thüm et al. '09]

- **idea:** ideally, make two calls to a SAT solver
 - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg \text{SAT}(\phi \wedge \neg \psi)$
 - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg \text{SAT}(\psi \wedge \neg \phi)$
- **but:** SAT requires conjunctive normal form (CNF), and $\neg \psi$ is large and explodes ⚡
- **solution:** split into many smaller SAT calls

Weaknesses

- all nontrivial edits are arbitrary 🤦
- requires $\mathcal{O}(n)$ SAT calls given length n of $\phi \wedge \psi$
- assumes added and removed features to be dead

implemented in  fIDE
feature

BDD-Based: Semantic Differencing

[Acher et al. '12]

- **idea:** reify differences as another feature model
- compile $\phi \wedge \neg \psi$ into a **binary decision diagram** (BDD)
- perfectly captures differences between versions

Weaknesses

- same coarse-grained classification as Thüm et al.
- requires a BDD, which currently does not scale to very large feature models
- no empirical evaluation
- assumes added and removed features to be dead

implemented in  FAMILIAR

SAT-Based (efficient, but coarse-grained) < **#SAT-Based (?)** < **BDD-Based** (fine-grained, but inefficient)

Quantified Reasoning About Edits to Feature Models – Contribution

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ

$\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 



Generalization

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$



$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 🌐
 $\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 😊

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin [‘83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$



$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 🎯

$\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 😊

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[‘83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 

$\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 



$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[‘83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 

$\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$



Generalization

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[‘83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 

$\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$



Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF
we can now **quantify** the degree of generalization!

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[‘83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 
 $\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$



Generalization

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[83]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF
we can now **quantify** the degree of generalization!

How to Handle Added and Removed Features?

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 

$\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 



Generalization

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[^{'83}]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF

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

How to Handle Added and Removed Features?

depends on the use case:
$$\frac{\#\text{SAT}(\theta_T(\pi_1 \phi \wedge \pi_2 \psi))}{\#\text{SAT}(\theta_T(\pi_1 \phi))}$$

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 
 $\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 



Generalization

$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[^{'83}]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF
we can now **quantify** the degree of generalization!

How to Handle Added and Removed Features?

depends on the use case:
$$\frac{\#\text{SAT}(\theta_T(\pi_1 \phi \wedge \pi_2 \psi))}{\#\text{SAT}(\theta_T(\pi_1 \phi))}$$

e.g., $\pi_1 = \exists(V_\phi \setminus V_\psi) \theta_D$ removed f's

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ
 $\Leftrightarrow \models \phi \rightarrow \psi$

$\Leftrightarrow \neg \text{SAT}(\theta_D(\phi \wedge \neg \psi))$ exponential 
 $\Leftrightarrow \neg \text{SAT}(\theta_T(\phi \wedge \neg \psi))$ linear + quasi-equiv. 



$\mathcal{O}(n) * \text{SAT} \Rightarrow \mathcal{O}(n)$ transformation + $\mathcal{O}(1) * \text{SAT}$

CNF Transformation θ_D : Distributive

apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T : Tseitin

[⁸³]

abbreviate every subformula χ with an **auxiliary variable** defined as $\text{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

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



#SAT = DPLL-style exhaustive search or d-DNNF
we can now **quantify** the degree of generalization!

How to Handle Added and Removed Features?

depends on the use case:
$$\frac{\#\text{SAT}(\theta_T(\pi_1 \phi \wedge \pi_2 \psi))}{\#\text{SAT}(\theta_T(\pi_1 \phi))}$$

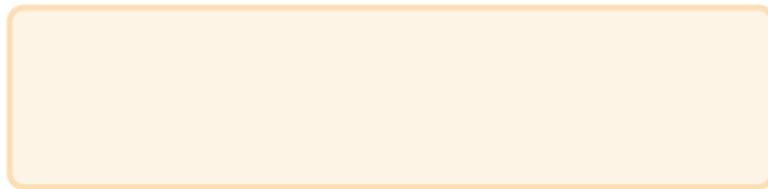
e.g., $\pi_1 = \exists(V_\phi \setminus V_\psi) \theta_D$ removed f's

e.g., $\pi_2 = \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg$ added f's

Future Work – Where Do We Go From Here?

Future Work – Where Do We Go From Here?

Applications



Future Work – Where Do We Go From Here?

Applications

- measure inadvertent **variability reduction**

Future Work – Where Do We Go From Here?

Applications

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**

Future Work – Where Do We Go From Here?

Applications

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

Future Work – Where Do We Go From Here?

Applications, Evaluations

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses



Future Work – Where Do We Go From Here?

Applications, Evaluations

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]

Future Work – Where Do We Go From Here?

Applications, Evaluations

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]

Future Work – Where Do We Go From Here?

Applications, Evaluations

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]

Future Work – Where Do We Go From Here?

Applications, Evaluations

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

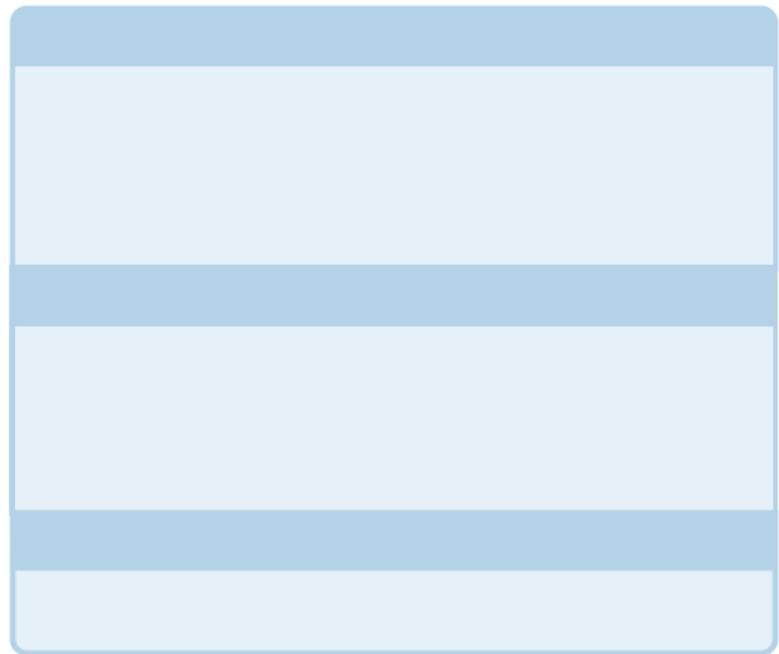
- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**

Future Work – Where Do We Go From Here?

Applications, Evaluations, and Algorithmic Improvements

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**



Future Work – Where Do We Go From Here?

Applications, Evaluations, and Algorithmic Improvements

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**

Total Vs. Partial Tseitin Transformation

- **total**: abbreviate every subformula
⇒ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
⇒ introduces fewer auxiliary variables

Future Work – Where Do We Go From Here?

Applications, Evaluations, and Algorithmic Improvements

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**

Total Vs. Partial Tseitin Transformation

- **total**: abbreviate every subformula
⇒ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
⇒ introduces fewer auxiliary variables

Projected Model Counting (# \exists SAT)

$$\frac{\# \exists \text{SAT}(\theta_T(\phi \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg \psi), V_\pi)}{\# \exists \text{SAT}(\theta_T(\phi), V_\pi)}$$

where $V_\pi = (V_\phi \setminus V_\psi) \cup \text{aux}$ [Sundermann et al. '24]

Future Work – Where Do We Go From Here?

Applications, Evaluations, and Algorithmic Improvements

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**

Total Vs. Partial Tseitin Transformation

- **total**: abbreviate every subformula
⇒ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
⇒ introduces fewer auxiliary variables

Projected Model Counting (# \exists SAT)

$$\frac{\# \exists \text{SAT}(\theta_T(\phi \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg \psi), V_\pi)}{\# \exists \text{SAT}(\theta_T(\phi), V_\pi)}$$

where $V_\pi = (V_\phi \setminus V_\psi) \cup \text{aux}$ [Sundermann et al. '24]

Non-Clausal Slicing

$$\pi_1 = \exists(V_\phi \setminus V_\psi) \theta_D \longrightarrow \pi_1 = \exists(V_\phi \setminus V_\psi)$$

Future Work – Where Do We Go From Here?

Applications, Evaluations, and Algorithmic Improvements

- measure inadvertent **variability reduction**
- measure inadvertent **variability growth**
- **reify differences** to explore them and lift analyses

- compare to **SAT-based reasoning** [Thüm et al. '09]
- and to **BDD-based differencing** [Acher et al. '12]
- and to **syntactic differencing** [Dintzner et al. '18]
- evaluate on models + investigate their **evolution**

Eliminate Tseitin and Negation

$$\frac{\#SAT(\theta_T(\phi \wedge \neg\psi))}{\#SAT(\theta_T(\phi))} = 1 - \frac{\#SAT(\theta_D(\phi \wedge \psi))}{\#SAT(\theta_D(\phi))}$$

Total Vs. Partial Tseitin Transformation

- **total**: abbreviate every subformula
⇒ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
⇒ introduces fewer auxiliary variables

Projected Model Counting (# \exists SAT)

$$\frac{\#\exists SAT(\theta_T(\phi \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg\psi), V_\pi)}{\#\exists SAT(\theta_T(\phi), V_\pi)}$$

where $V_\pi = (V_\phi \setminus V_\psi) \cup \text{aux}$ [Sundermann et al. '24]

Non-Clausal Slicing

$$\pi_1 = \exists(V_\phi \setminus V_\psi) \theta_D \longrightarrow \pi_1 = \exists(V_\phi \setminus V_\psi)$$

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)



MME TORTUE

renowned feature-model historian
studies evolution for a living

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)



MME TORTUE

renowned feature-model historian
studies evolution for a living

torte 🍰

[tɔrtə]

- extracts **feature-model histories**
- extends KConfigReader, KClause (+ ConfigFix)
- reproducible + fully automated

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)



MME TORTUE

renowned feature-model historian
studies evolution for a living



SIR KLAUS

practiced feature-model surgeon
always eager to slice and diff

torte

[tɔrtə /ekuitər/torte]

- extracts **feature-model histories**
- extends KConfigReader, KClause (+ ConfigFix)
- reproducible + fully automated

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)



MME TORTUE

renowned feature-model historian
studies evolution for a living

torte 🍰

[@/ekuiter/torte]

- extracts **feature-model histories**
- extends KConfigReader, KClause (+ ConfigFix)
- reproducible + fully automated



SIR KLAUS

practiced feature-model surgeon
always eager to slice and diff

clausy 😊

[@/ekuiter/clausy]

- transforms feature models **into CNF**
- competes with Z3 in performance
- supports diffing (slicing planned)

Conclusion

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

Improving SAT-Based Reasoning

ψ generalizes ϕ iff $\neg \text{SAT}(\theta_D(\phi \wedge \neg\psi))$

ψ generalizes ϕ iff $\neg \text{SAT}(\theta_T(\phi \wedge \neg\psi))$

Introducing #SAT-Based Reasoning

the degree of removed configurations is

$$(\text{none}) \quad 0 \leq \frac{\# \text{SAT}(\theta_T(\pi_1 \phi \wedge \pi_2 \psi))}{\# \text{SAT}(\theta_T(\pi_1 \phi))} \leq 1 \quad (\text{all})$$

e.g., $\pi_1 = \exists(V_\phi \setminus V_\psi) \theta_D$

removed f's

e.g., $\pi_2 = \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg$ added f's



@ekuiter/torte 🍰



@ekuiter/clausy 🎅

⚠ Disclaimer: No penguins were AI-generated in the making of this presentation. All were returned to TIKZPINGUS, their natural habitat.

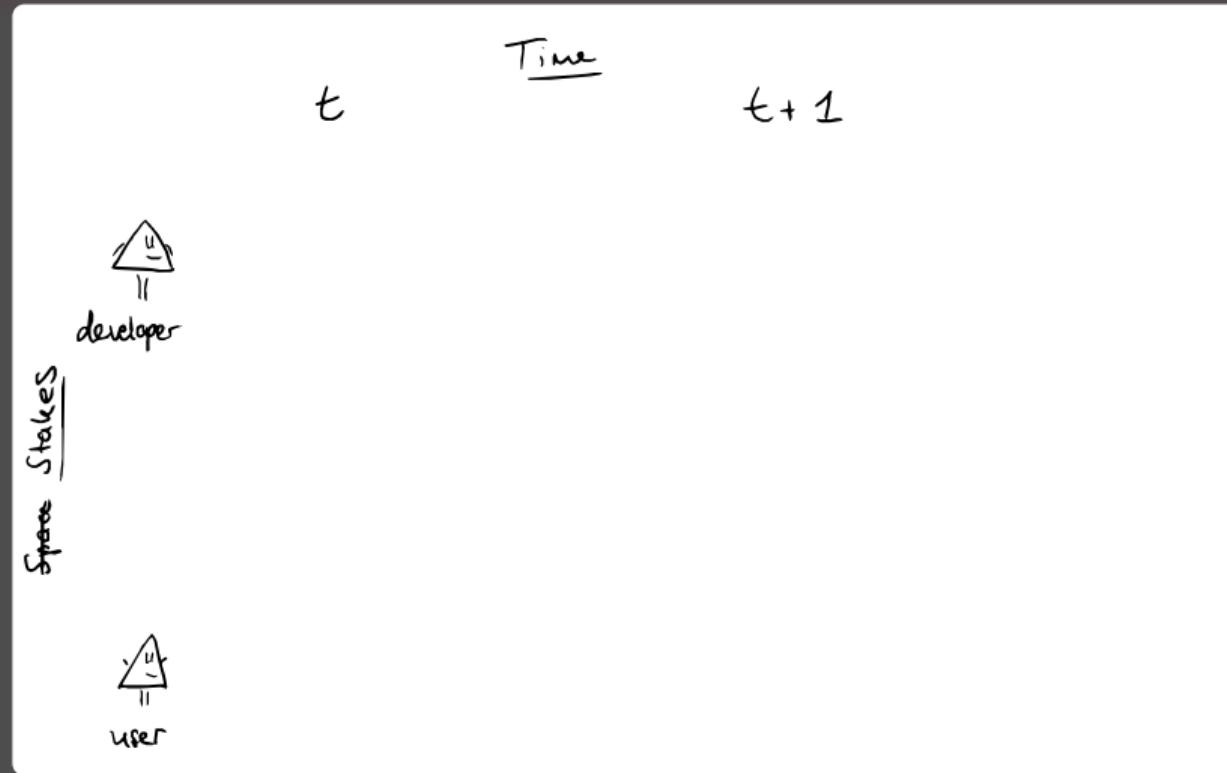
Inadvertent Variability Reduction – The Use Case

Inadvertent Variability Reduction – The Use Case

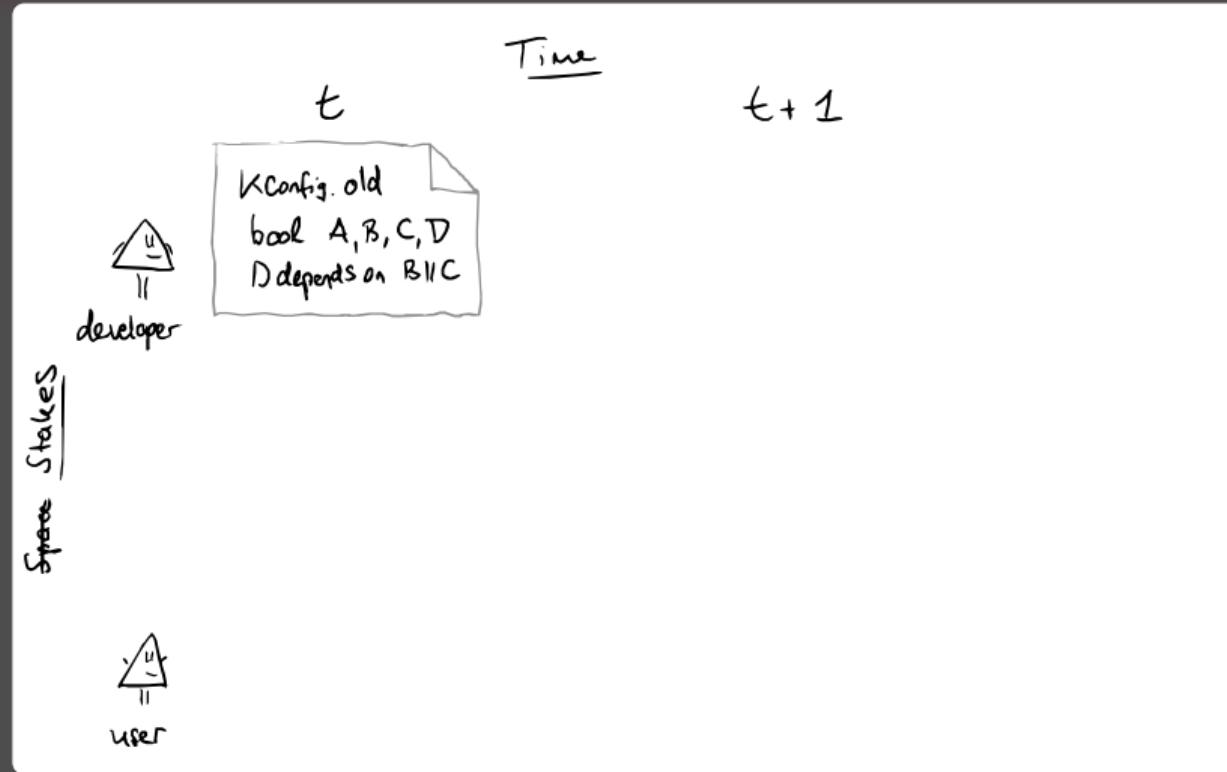
Space Stakes

Time

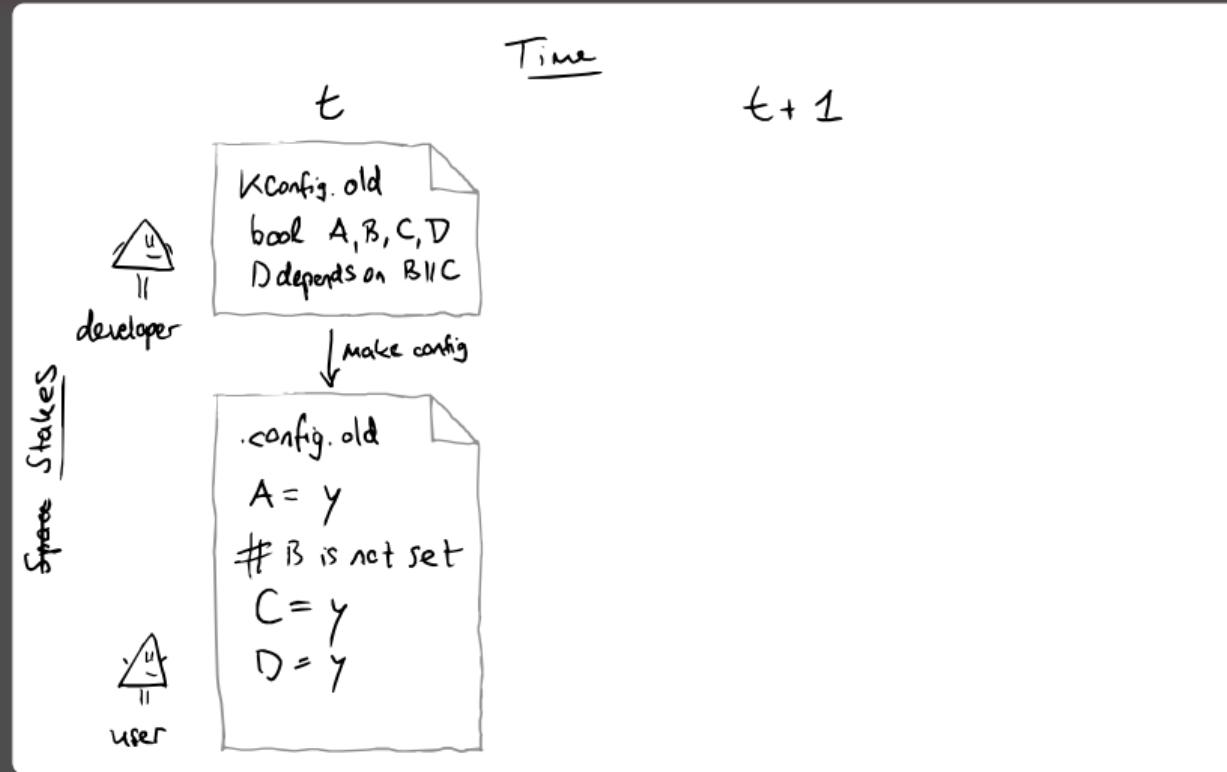
Inadvertent Variability Reduction – The Use Case



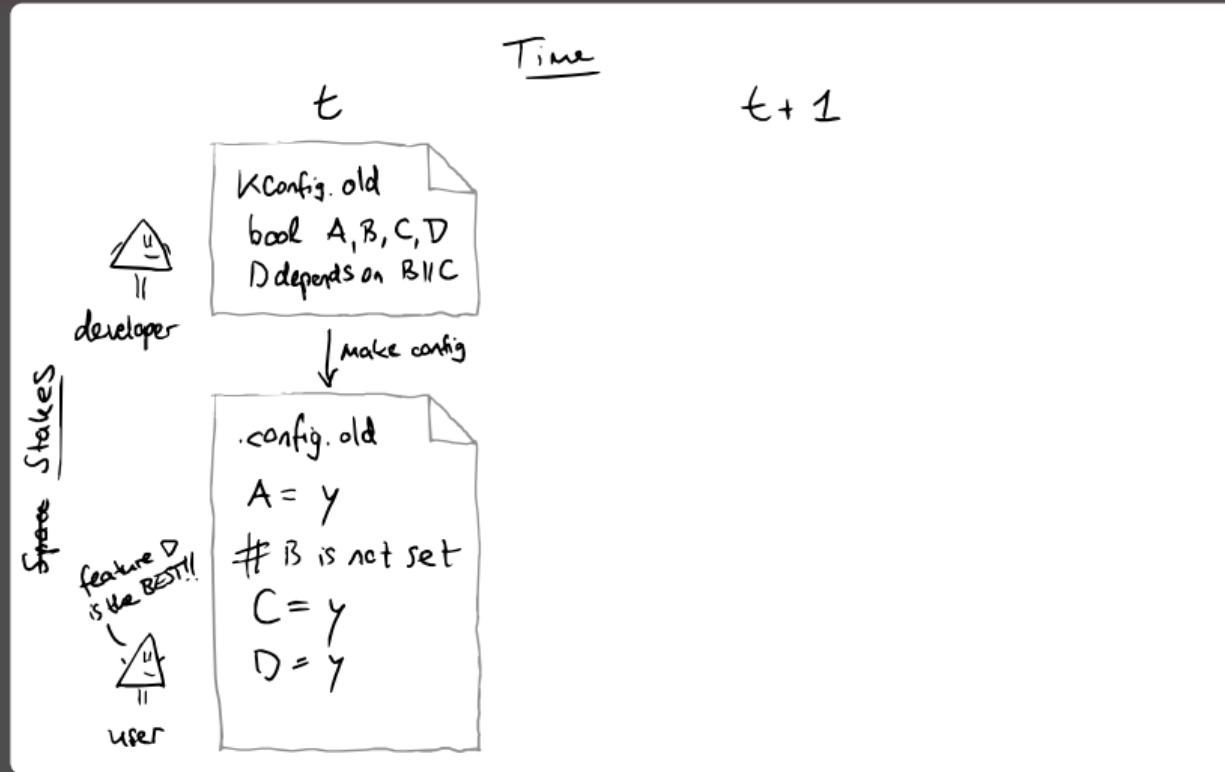
Inadvertent Variability Reduction – The Use Case



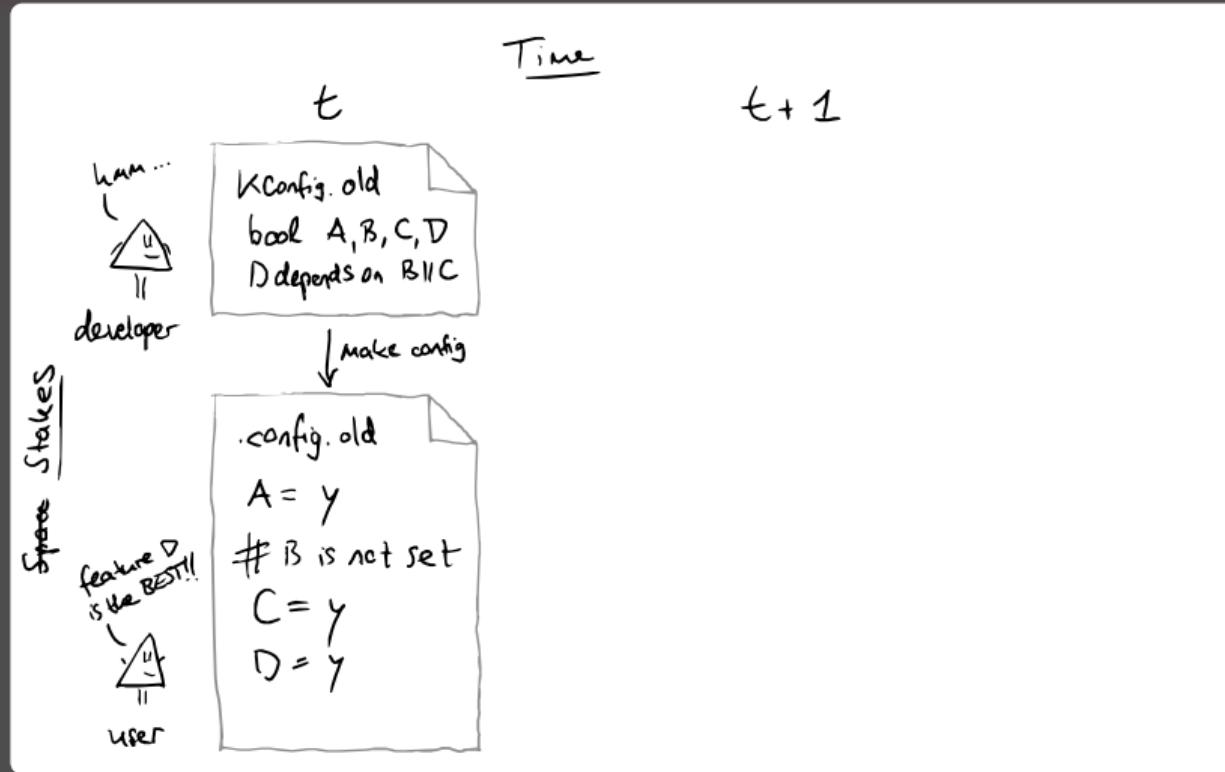
Inadvertent Variability Reduction – The Use Case



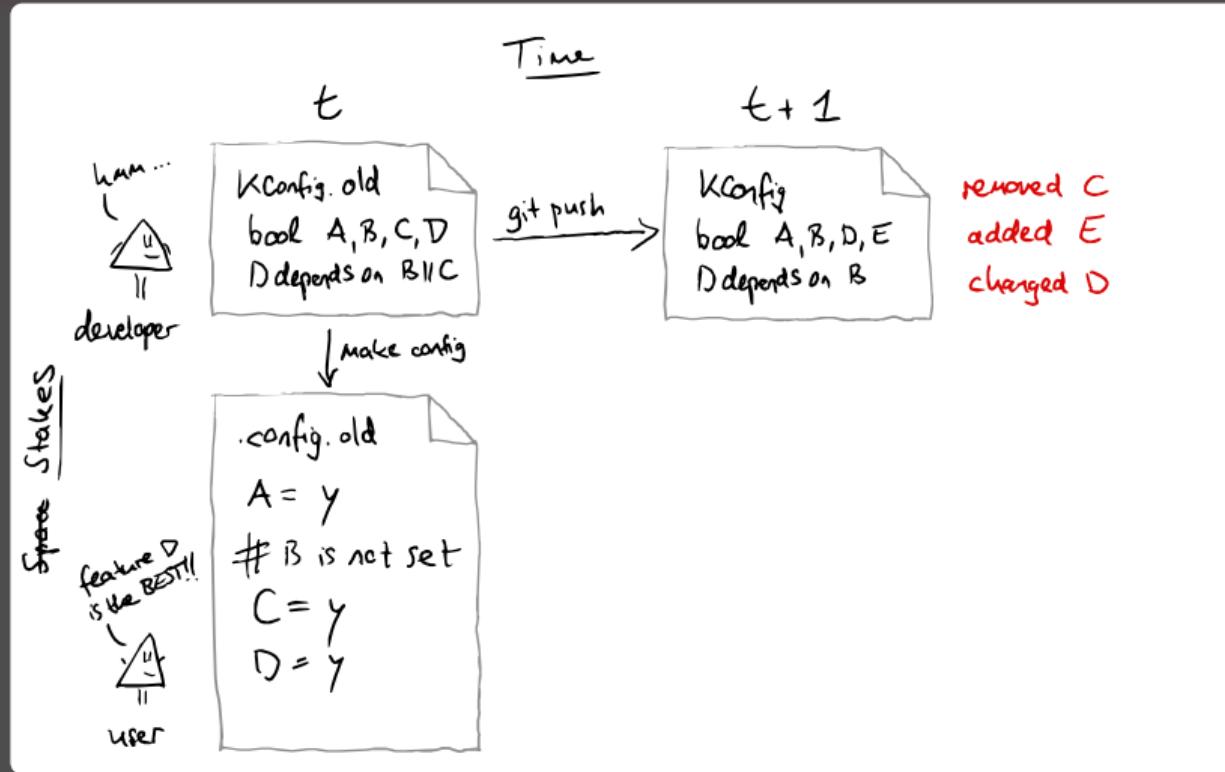
Inadvertent Variability Reduction – The Use Case



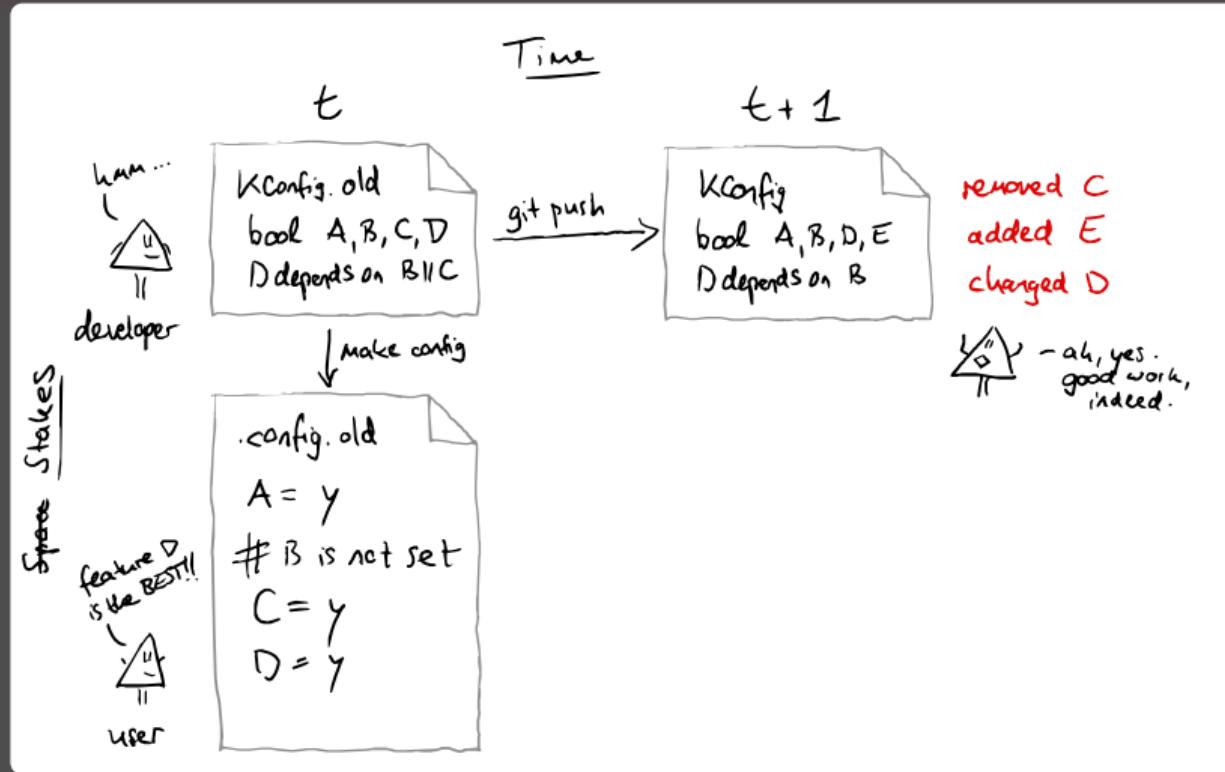
Inadvertent Variability Reduction – The Use Case



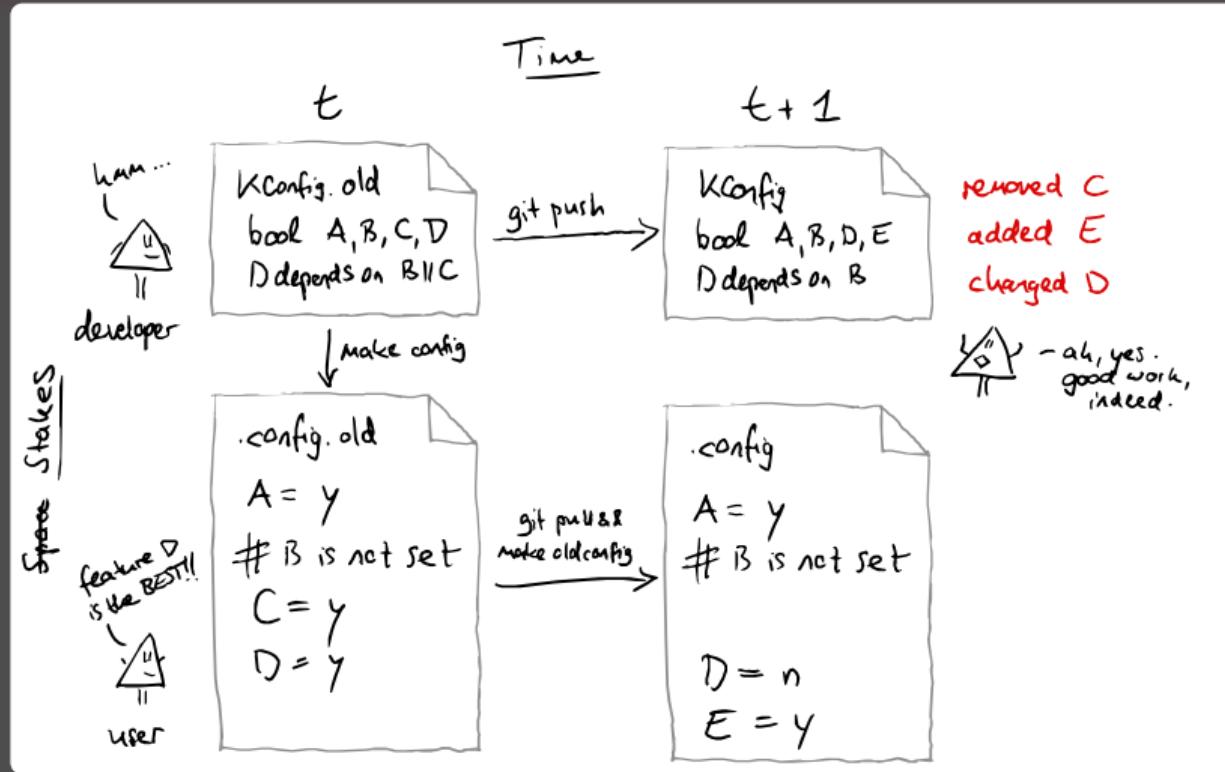
Inadvertent Variability Reduction – The Use Case



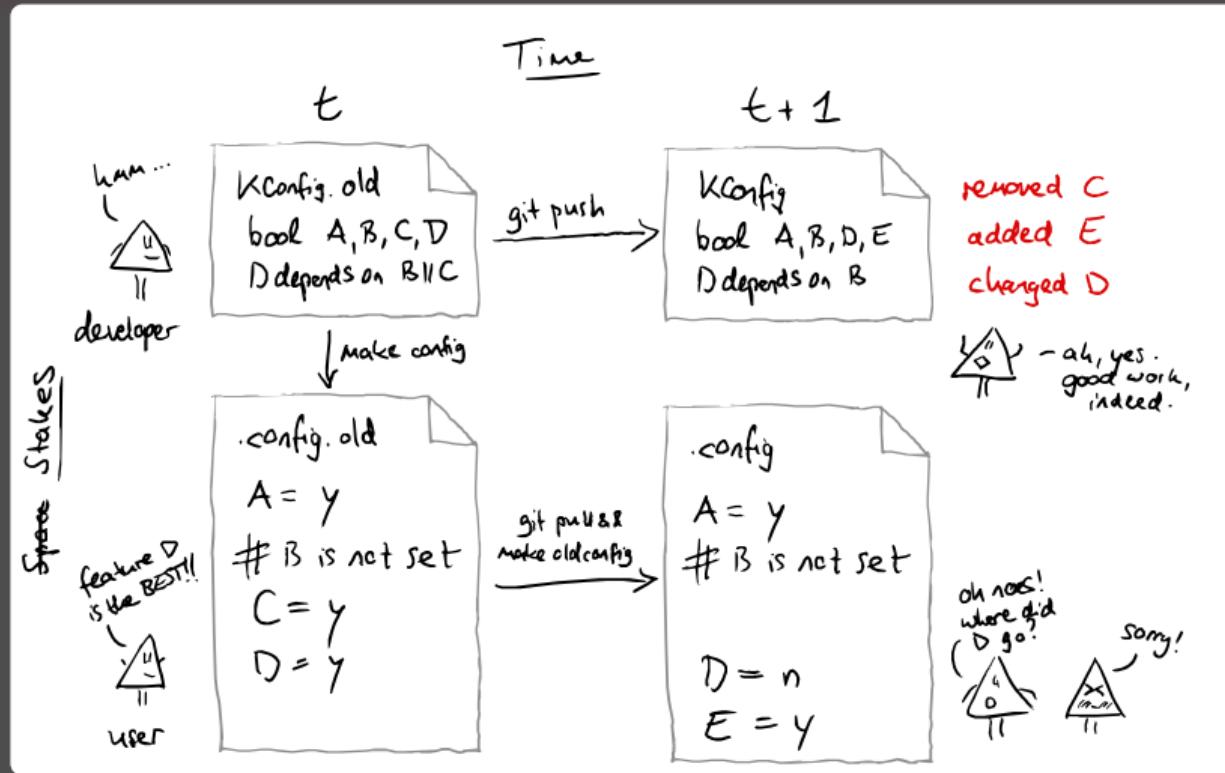
Inadvertent Variability Reduction – The Use Case



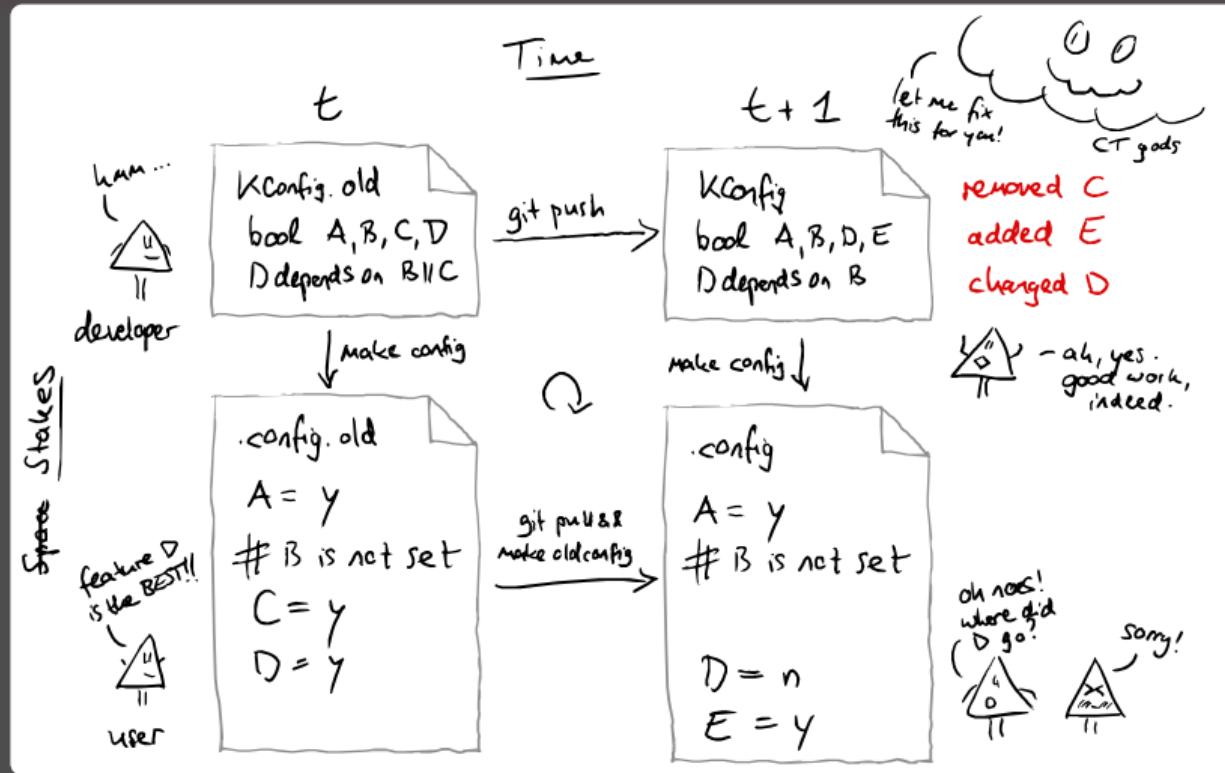
Inadvertent Variability Reduction – The Use Case



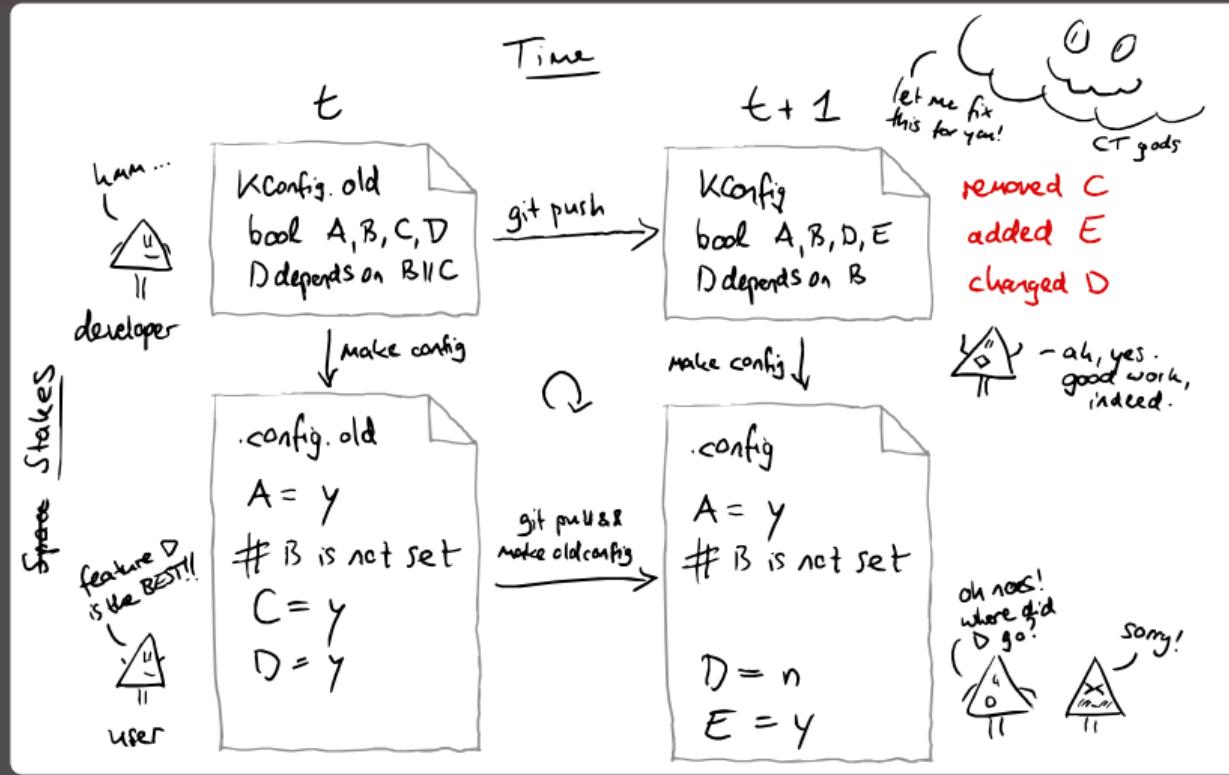
Inadvertent Variability Reduction – The Use Case



Inadvertent Variability Reduction – The Use Case



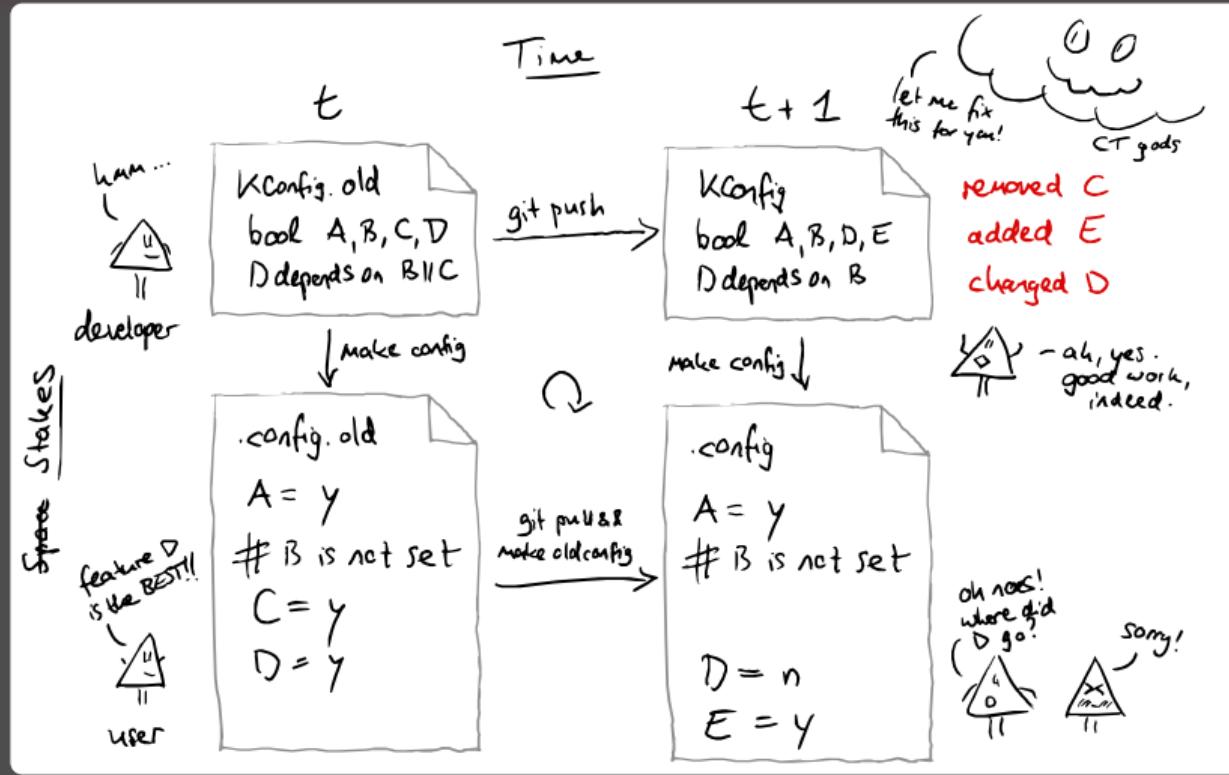
Inadvertent Variability Reduction – The Use Case



Assuming ...

- users are uniform over configurations
- updates are non-interactive
- users expect choices to be preserved

Inadvertent Variability Reduction – The Use Case



Assuming ...

- users are uniform over configurations
- updates are non-interactive
- users expect choices to be preserved

... how many of our users are negatively impacted by an update?

- ⇒ decision-making
- ⇒ understanding configuration spaces

Inadvertent Variability Reduction – The Math

$\text{cfg_loss} = \dots$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\phi}{\phi \wedge \neg \psi}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg_loss} = \frac{\#SAT(\phi \wedge \neg \psi)}{\#SAT(\phi)}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg_loss} = \frac{\#\text{SAT}(\theta_T(\phi \wedge \neg \psi))}{\#\text{SAT}(\theta_T(\phi))}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT(\Theta_T(\exists v_\phi | v_\psi \theta_b(\phi) \wedge \neg \psi))}{\#SAT(\Theta_T(\exists v_\phi | v_\psi \theta_b(\phi)))}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#\text{SAT}\left(\Theta_T\left(\exists v_\phi \vee_\psi \Theta_D(\phi)\right) \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg \psi\right)}{\#\text{SAT}(\Theta_T(\exists v_\phi \vee_\psi \Theta_D(\phi)))}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT\left(\Theta_T\left(\exists v_\phi | v_\psi \theta_b(\phi) \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg \psi\right)\right)}{\#SAT\left(\Theta_T\left(\exists v_\phi | v_\psi \theta_b(\phi)\right)\right)}$$

a clever combination of distributive and Tseitin transformation, #SAT, and slicing

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT\left(\Theta_T\left(\exists v_\phi \vee_\psi \Theta_\phi(\phi) \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \wedge \neg \psi\right)\right)}{\#SAT\left(\Theta_T\left(\exists v_\phi \vee_\psi \Theta_\phi(\phi)\right)\right)}$$

a clever combination of distributive and Tseitin transformation, #SAT, and slicing

$$\begin{aligned} & \#SAT\left(\Theta_T\left(\left(\exists v_\phi \vee_\psi \Theta_\phi(\phi)\right) \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \circ \psi\right)\right) \\ &= \# \exists SAT\left(\Theta_T\left(\left(\phi \wedge \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \text{def}(v)) \circ \psi\right), V_\phi \vee_\psi \text{aux}\right)\right) \end{aligned}$$

instead of #SAT and slicing (e.g., FeatureIDE), we can also use # \exists SAT (e.g., pd4)

negation can also be avoided when using $1 - \#SAT(x)$

