
[Elbe, Magdeburg]

Quantified Reasoning About Edits to Feature Models

FOSD 2025 — March 25–28 — Köthen

Elias Kuiter[1], Thomas Thüm[2], Gunter Saake[1]

University of Magdeburg[1], TU Braunschweig[2]

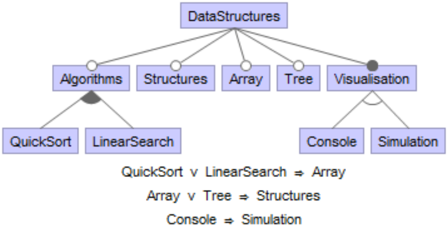
Quantified Reasoning About Edits to Feature Models

Implementierungstechniken für Software-Produktlinien

Übung 10: Analyse von Produktlinien

1. **Feature-Modell-Analyse**

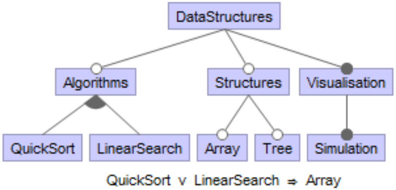
   Gegeben sei das folgende Feature-Modell FM.



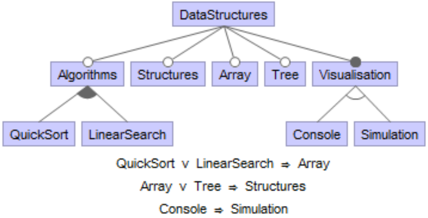
2. **Evolution von Feature-Modellen**

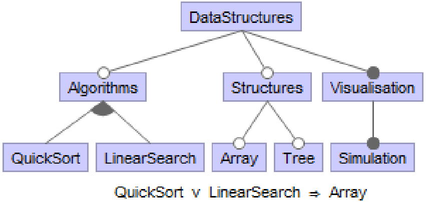
   (a) Welche semantischen Änderungen an Feature-Modellen können vorgenommen werden?

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

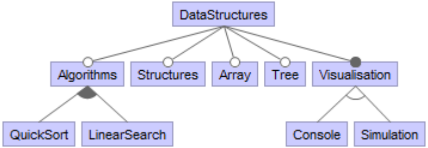


Quantified Reasoning About **Edits to** Feature Models





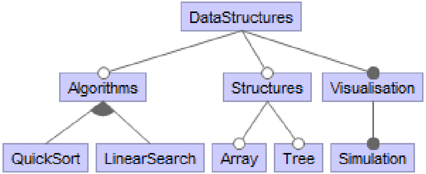
Quantified Reasoning About **Edits to** Feature Models



QuickSort ∨ LinearSearch ⇒ Array

Array ∨ Tree ⇒ Structures

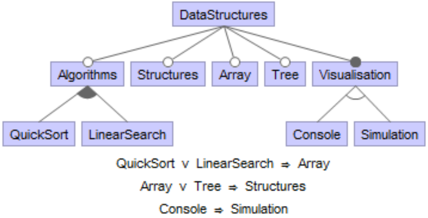
Console ⇒ Simulation

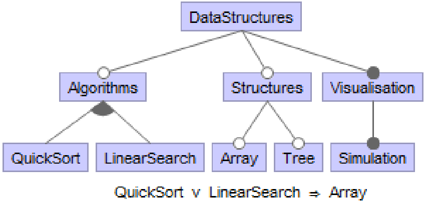


QuickSort ∨ LinearSearch ⇒ Array

What Happens to the Configuration Space?

Quantified Reasoning About Edits to Feature Models

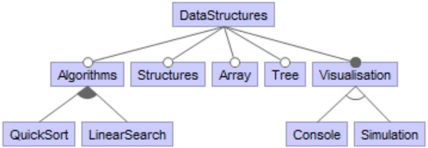




What Happens to the Configuration Space?

	No Products Added	Products Added
No Products Deleted		
Products Deleted		

Quantified Reasoning About Edits to Feature Models

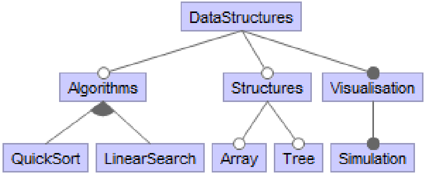


QuickSort ∨ LinearSearch ⇒ Array

Array ∨ Tree ⇒ Structures

Console ⇒ Simulation

$\stackrel{?}{=}$

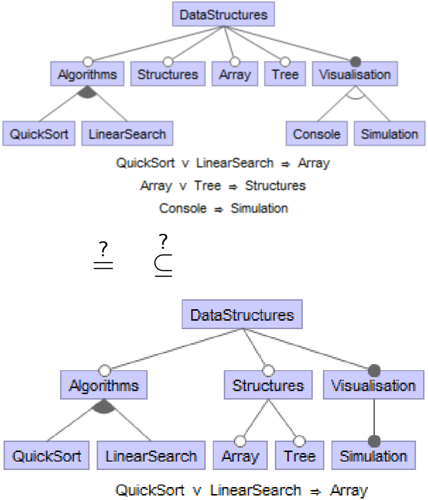


QuickSort ∨ LinearSearch ⇒ Array

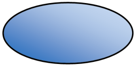
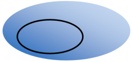
What Happens to the Configuration Space?

	No Products Added	Products Added
No Products Deleted	Refactoring	
Products Deleted		

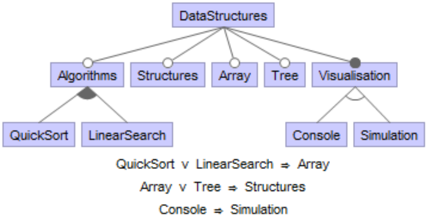
Quantified Reasoning About Edits to Feature Models



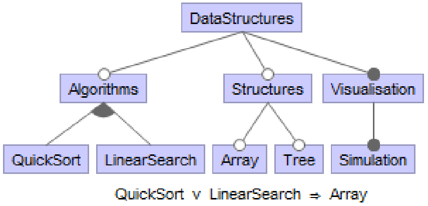
What Happens to the Configuration Space?

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

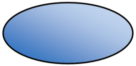
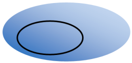
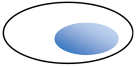
Quantified Reasoning About **Edits to** Feature Models



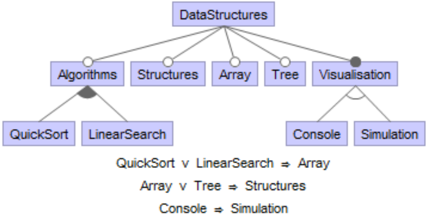
$\stackrel{?}{=}$ $\stackrel{?}{\subseteq}$ $\stackrel{?}{\supseteq}$



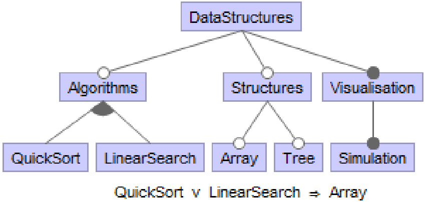
What Happens to the Configuration Space?

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

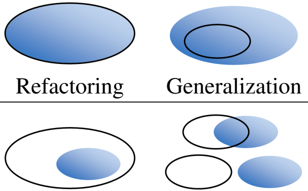
Quantified Reasoning About <u>Edits to</u> Feature Models



$$\overset{?}{=} \quad \overset{?}{\subseteq} \quad \overset{?}{\supseteq} \quad \cap \overset{?}{=} \emptyset$$

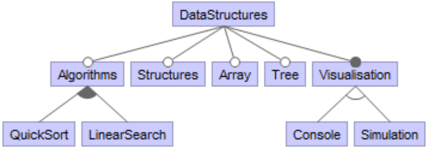


What Happens to the Configuration Space?



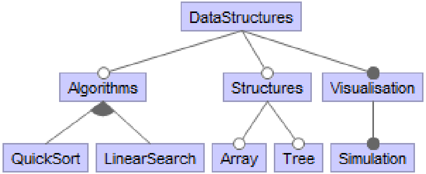
	No Products Added	Products Added
No Products Deleted	Refactoring	Generalization
Products Deleted	Specialization	Arbitrary Edit

Quantified Reasoning About **Edits to** Feature Models



QuickSort ∨ LinearSearch ⇒ Array
Array ∨ Tree ⇒ Structures
Console ⇒ Simulation

$\overset{?}{=} \quad \overset{?}{\subseteq} \quad \overset{?}{\supseteq} \quad \cap \overset{?}{=} \emptyset$

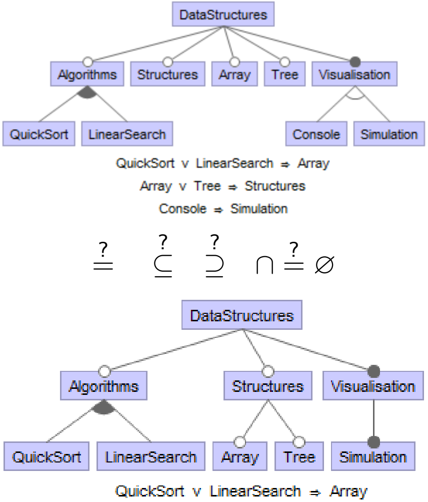


QuickSort ∨ LinearSearch ⇒ Array

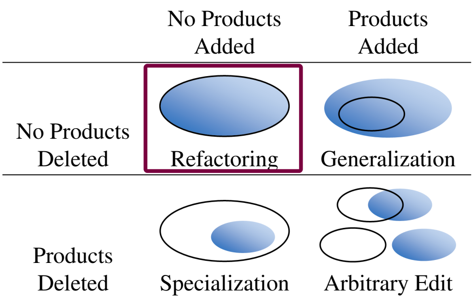
What Happens to the Configuration Space?

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

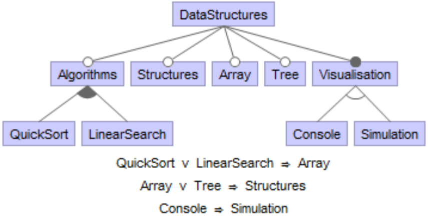
Quantified Reasoning About **Edits to** Feature Models



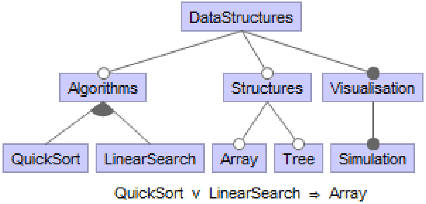
Feature-Model Edits [Thüm et al. '09]



Quantified Reasoning About Edits to Feature Models



$$\stackrel{?}{=} \qquad \stackrel{?}{\subseteq} \qquad \stackrel{?}{\supseteq} \qquad \cap \stackrel{?}{=} \emptyset$$



Feature-Model Edits [Thüm et al. '09]

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

- **goal**: compare versions of a feature model
- **use cases**: e.g., to avoid unintentional changes, understand patterns in evolution, or support continuous integration ⇒ **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 $\models \phi \to \psi$ iff $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$

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 $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$

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]

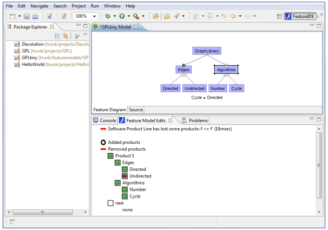
- **idea**: ideally, make two calls to a SAT solver
  - ψ **generalizes** ϕ iff $\models \phi \rightarrow \psi$ iff $\neg\mathsf{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathsf{SAT}(\psi \wedge \neg\phi)$
- **but**: SAT requires conjunctive normal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

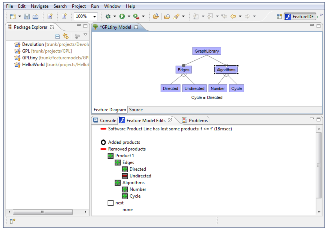
SAT-Based: Simplified Reasoning [Thüm et al. '09]

- **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 💣
- **solution**: split into many smaller SAT calls

Weaknesses

let's see . . .

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

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 $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **but**: SAT requires conjunctive normal form (CNF), and $\neg\psi$ is large and explodes
- **solution**: split into many smaller SAT calls

Weaknesses

implemented in 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 $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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

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 $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 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 $\models \phi \to \psi$ iff $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 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 $\models \phi \to \psi$ iff $\neg$SAT$(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg$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 FeatureIDE

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 $\models \phi \to \psi$ iff $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 FeatureIDE

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 FeatureIDE

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 $\models \phi \to \psi$ iff $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 FeatureIDE

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 $\models \phi \to \psi$ iff $\neg$SAT$(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg$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 FeatureIDE

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 $\models \phi \to \psi$ iff $\neg$SAT($\phi \land \neg\psi$)
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg$SAT($\psi \land \neg\phi$)
- **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 $\phi \land \psi$
- assumes added and removed features to be dead

implemented in FeatureIDE

BDD-Based: Semantic Differencing [Acher et al. '12]

- **idea**: reify differences as another feature model
- compile $\phi \land \neg\psi$ 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 $\models \phi \to \psi$ iff $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \to \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 FeatureIDE

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 $\neg$SAT$(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg$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 FeatureIDE

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 $\neg\mathrm{SAT}(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg\mathrm{SAT}(\psi \wedge \neg\phi)$
- **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 FeatureIDE

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) **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 $\neg$SAT$(\phi \wedge \neg\psi)$
  - ψ **specializes** ϕ iff $\models \psi \rightarrow \phi$ iff $\neg$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 FeatureIDE

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

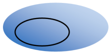
implemented in FAMILIAR

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

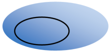


Generalization

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

ψ generalizes ϕ

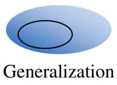

Generalization

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning

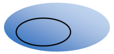
ψ generalizes ϕ

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



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

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

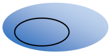
$\Leftrightarrow \neg\mathrm{SAT}(\theta_D(\phi \wedge \neg\psi))$ exponential

CNF Transformation θ_D: Distributive

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

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

$\Leftrightarrow \neg\mathsf{SAT}(\theta_D(\phi \land \neg\psi))$ exponential

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

CNF Transformation θ_D: Distributive

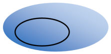
apply laws of logic: **De Morgan** + **distributivity**

CNF Transformation θ_T: Tseitin ['83]

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

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

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

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

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

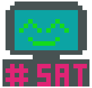
CNF Transformation θ_D: Distributive

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

CNF Transformation θ_T: Tseitin ['83]

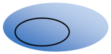
abbreviate every subformula χ with an **auxiliary variable** defined as $\mathrm{aux}_\chi \leftrightarrow \chi$

Introducing #SAT-Based Reasoning



Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

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

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

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

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

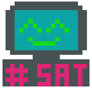
CNF Transformation θ_D: Distributive

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

CNF Transformation θ_T: Tseitin ['83]

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

Introducing #SAT-Based Reasoning

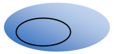


the degree of removed configurations is

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

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

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

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

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

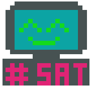
CNF Transformation θ_D: Distributive

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

CNF Transformation θ_T: Tseitin ['83]

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

Introducing #SAT-Based Reasoning



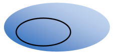
the degree of removed configurations is

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

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



ψ generalizes ϕ

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

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

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

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

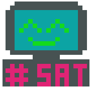
CNF Transformation θ_D: Distributive

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

CNF Transformation θ_T: Tseitin ['83]

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

Introducing #SAT-Based Reasoning



the degree of removed configurations is

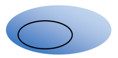
$$\text{(none) } 0 \leq \frac{\#\mathsf{SAT}(\theta_T(\phi \wedge \neg\psi))}{\#\mathsf{SAT}(\theta_T(\phi))} \leq 1 \text{ (all)}$$

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

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



ψ generalizes ϕ

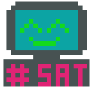
$\Leftrightarrow\ \models \phi \rightarrow \psi$

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

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

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

Introducing #SAT-Based Reasoning



the degree of removed configurations is

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

#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**

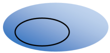
CNF Transformation θ_T: Tseitin ['83]

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

How to Handle Added and Removed Features?

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

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

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

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

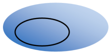
we can now **quantify** the degree of generalization!

How to Handle Added and Removed Features?

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

Quantified Reasoning About Edits to Feature Models – Contribution

Improving SAT-Based Reasoning



Generalization

ψ generalizes ϕ

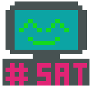
$\Leftrightarrow \models \phi \rightarrow \psi$

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

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

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

Introducing #SAT-Based Reasoning



the degree of removed configurations is

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

#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 $\mathrm{aux}_\chi \leftrightarrow \chi$

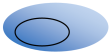
How to Handle Added and Removed Features?

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

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



Generalization

ψ generalizes ϕ

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

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

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

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

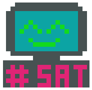
CNF Transformation θ_D: Distributive

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

CNF Transformation θ_T: Tseitin ['83]

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

Introducing #SAT-Based Reasoning



the degree of removed configurations is

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

#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: $\dfrac{\#\mathrm{SAT}(\theta_T(\pi_1 \ \phi \wedge \pi_2 \ \psi))}{\#\mathrm{SAT}(\theta_T(\pi_1 \ \phi))}$

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

e.g., $\pi_2 = \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \mathrm{def}(v)) \ \wedge \neg$ 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
  $\Rightarrow$ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
  $\Rightarrow$ 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
  $\Rightarrow$ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
  $\Rightarrow$ introduces fewer auxiliary variables

Projected Model Counting (#∃SAT)

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

where $V_\pi = (V_\phi \setminus V_\psi) \cup \mathrm{aux}$ [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
  $\Rightarrow$ negation in $\mathcal{O}(1)$, greenuces transform effort
- **partial**: abbreviate only selected subformulas
  $\Rightarrow$ introduces fewer auxiliary variables

Projected Model Counting (#∃SAT)

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

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

Non-Clausal Slicing

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

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

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

Total Vs. Partial Tseitin Transformation

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

Projected Model Counting (#∃SAT)

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

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

Non-Clausal Slicing

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

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)

<u>Mme Tortue</u>

renowned feature-model historian
studies evolution for a living

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)

<u>Mme Tortue</u>

renowned feature-model historian
studies evolution for a living

torte 🍰 [/ekuiter/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

Sir Klaus

practiced feature-model surgeon
always eager to slice and diff

torte 🍰 [/ekuiter/torte]

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

Implementation – Meet Two Enthusiastic Researchers (and Their Tools)

<u>MME TORTUE</u>

renowned feature-model historian
studies evolution for a living

<u>SIR KLAUS</u>

practiced feature-model surgeon
always eager to slice and diff

torte 🍰 [/ekuiter/torte]

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

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 $\neg\mathsf{SAT}(\theta_D(\phi \wedge \neg\psi))$

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

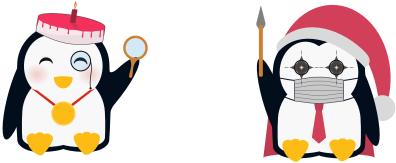
Introducing #SAT-Based Reasoning

the degree of removed configurations is

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

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

e.g., $\pi_2 = \bigwedge_{v \in V_\psi \setminus V_\phi} (v \leftrightarrow \mathsf{def}(v))\ \ \wedge \neg$ 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.

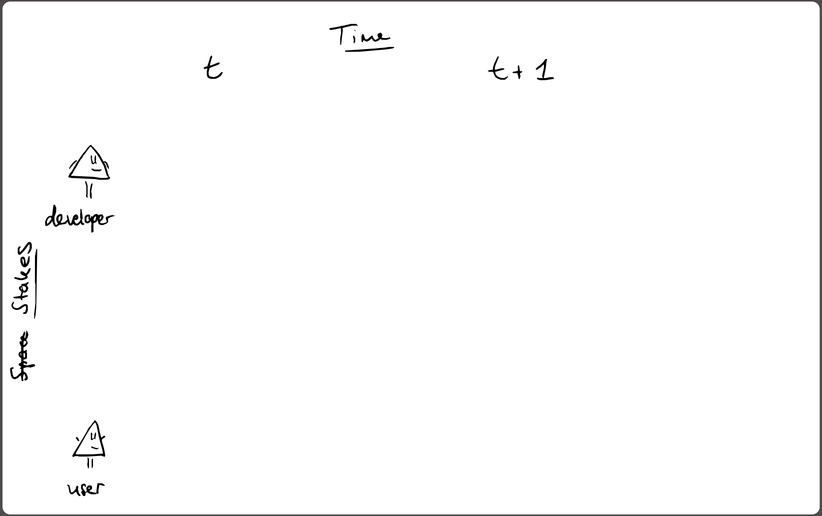
Inadvertent Variability Reduction – The Use Case

Inadvertent Variability Reduction – The Use Case

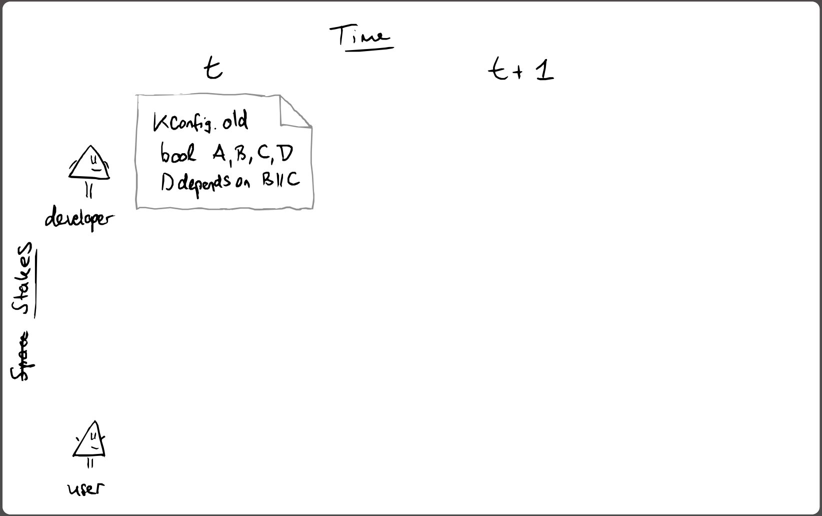
Time

~~Space~~ Stakes

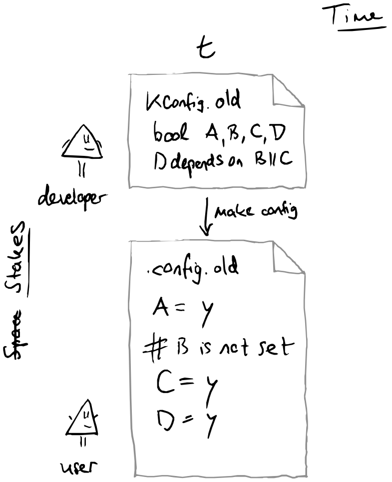
Inadvertent Variability Reduction – The Use Case



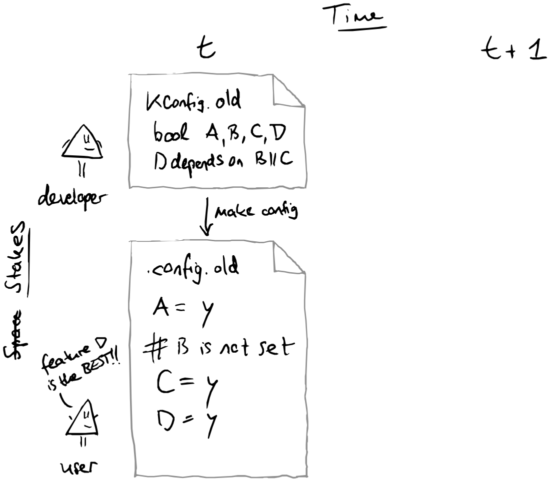
Inadvertent Variability Reduction – The Use Case



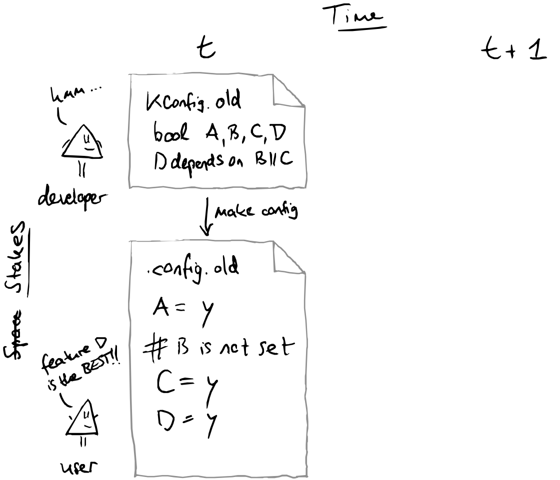
Inadvertent Variability Reduction – The Use Case



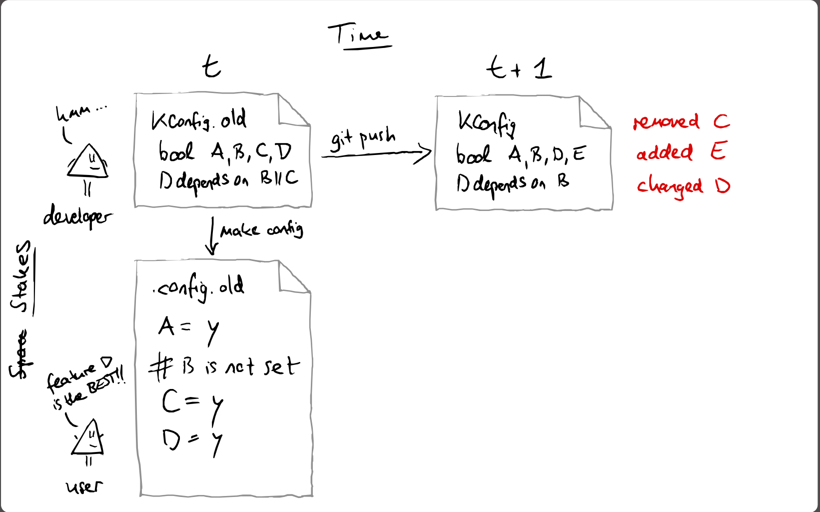
Inadvertent Variability Reduction – The Use Case



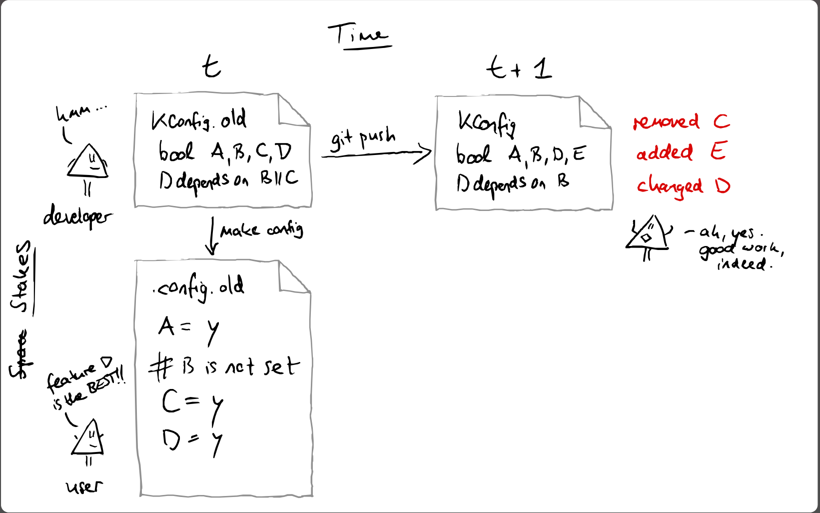
Inadvertent Variability Reduction – The Use Case



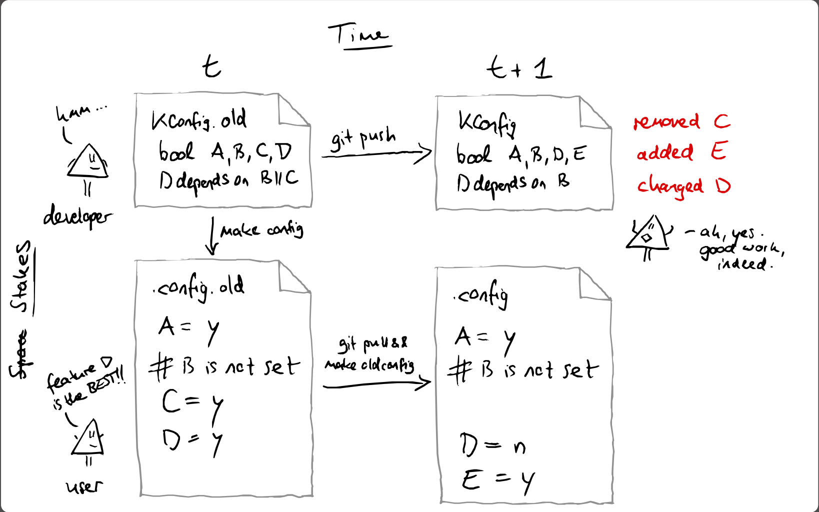
Inadvertent Variability Reduction – The Use Case



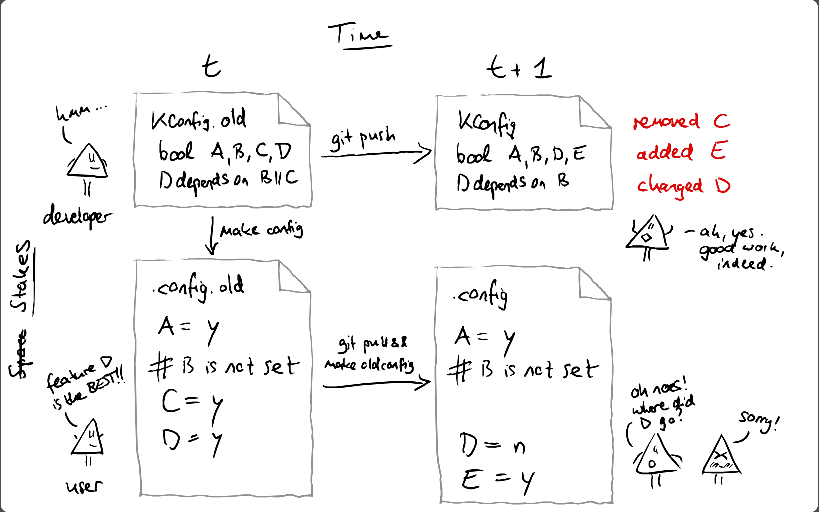
Inadvertent Variability Reduction – The Use Case



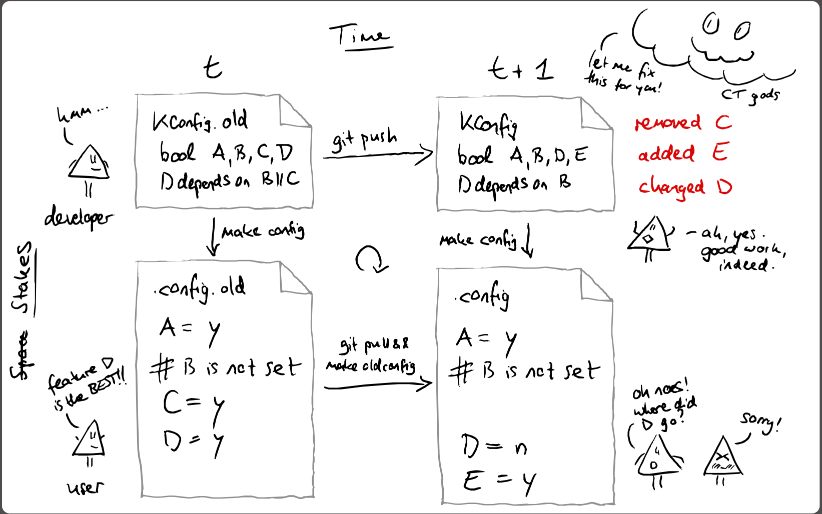
Inadvertent Variability Reduction – The Use Case



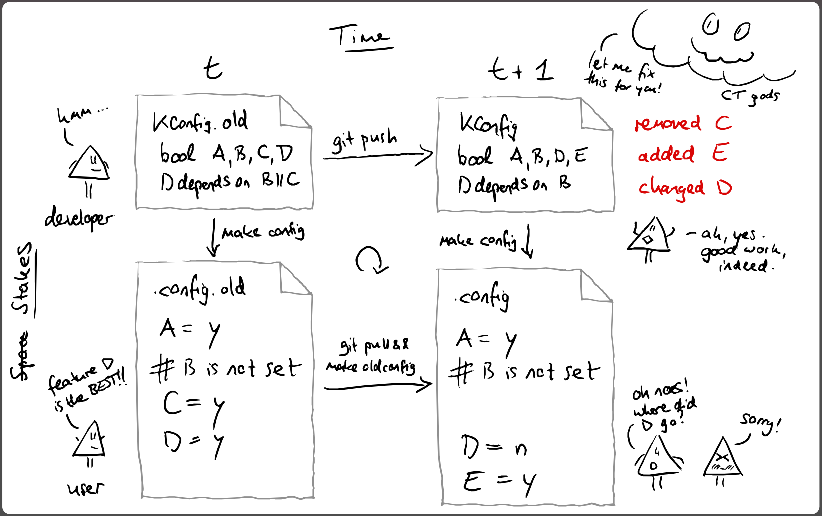
Inadvertent Variability Reduction – The Use Case



Inadvertent Variability Reduction – The Use Case



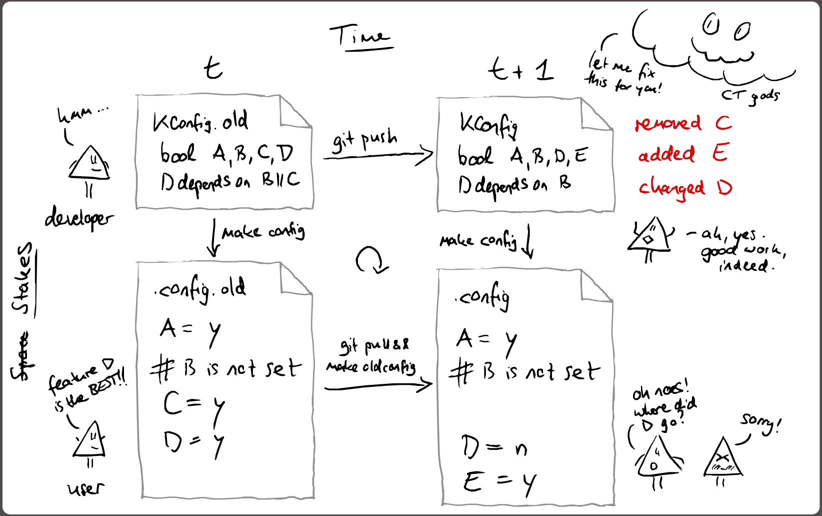
Inadvertent Variability Reduction – The Use Case



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

cfg loss = ______________________________

Inadvertent Variability Reduction – The Math

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

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT(\phi \land \neg\psi)}{\#SAT(\phi)}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT(\Theta_T(\quad \phi \quad \wedge \neg \psi))}{\#SAT(\Theta_T(\quad \phi))}$$

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT(\Theta_T(\exists v_\phi \setminus v_\psi \, \Theta_D(\phi) \qquad \wedge \neg \psi))}{\#SAT(\Theta_T(\exists v_\phi \setminus v_\psi \, \Theta_D(\phi)))}$$

Inadvertent Variability Reduction – The Math

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

Inadvertent Variability Reduction – The Math

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

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

Inadvertent Variability Reduction – The Math

$$\text{cfg loss} = \frac{\#SAT\left(\Theta_T\left(\exists V_\Phi \setminus V_\Psi\, \Theta_D(\Phi) \wedge \bigwedge_{v \in V_\Psi \setminus V_\Phi} (v \leftrightarrow def(v)) \wedge \neg \Psi\right)\right)}{\#SAT\left(\Theta_T\left(\exists V_\Phi \setminus V_\Psi\, \Theta_D(\Phi)\right)\right)}$$

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

---

$$\#SAT\left(\Theta_T\left(\left(\exists V_\Phi \setminus V_\Psi \left(\Theta_D(\Phi)\right) \wedge \bigwedge_{v \in V_\Psi \setminus V_\Phi} (v \leftrightarrow def(v))\right) \circ \Psi\right)\right)$$

$$= \#\exists SAT\left(\Theta_T\left(\left(\Phi \wedge \bigwedge_{v \in V_\Psi \setminus V_\Phi} (v \leftrightarrow def(v))\right) \circ \Psi\right), V_\Phi \setminus V_\Psi \cup aux\right)$$

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

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

Removed Features
Added Features
Share of Features
Year
Removed Constraints
Added Constraints
Share of Constraints
Year
left_diff_kind=weak
Removed Products
Added Products
Share of Products
Year
diff_kind=bottom-strong
diff_kind=top-strong
left_diff_kind=weak