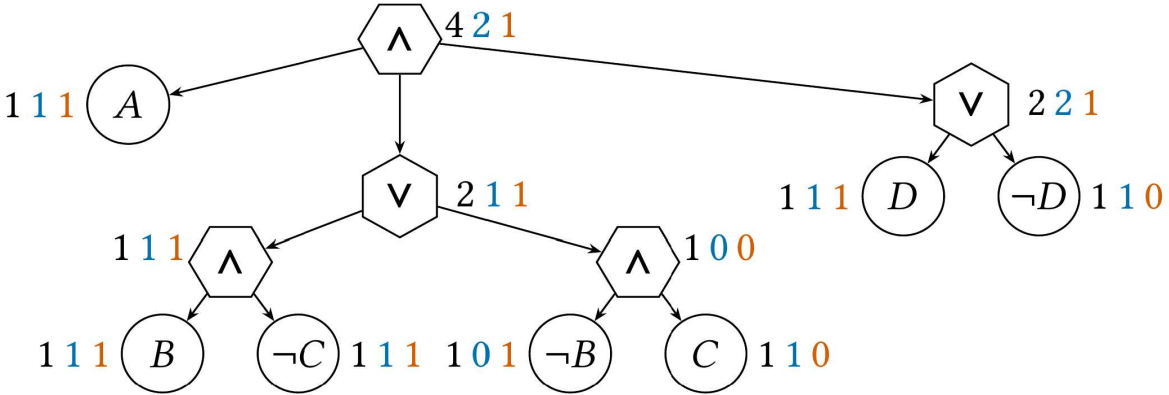
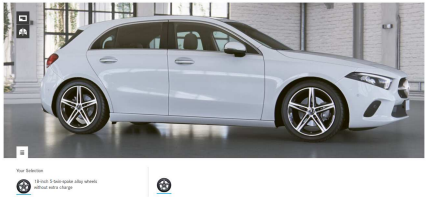
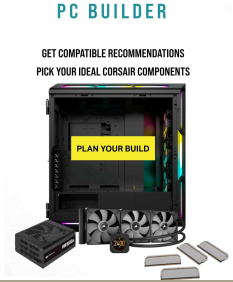


Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

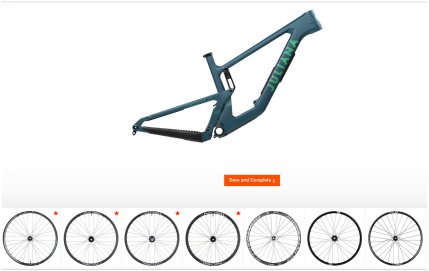
Motivation Product Lines

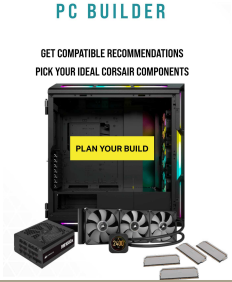


Motivation Product Lines

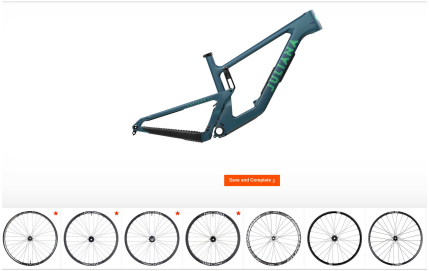


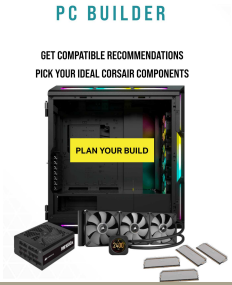
Motivation Product Lines

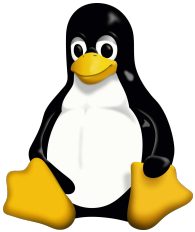




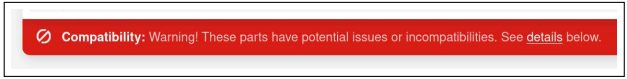
Motivation Product Lines





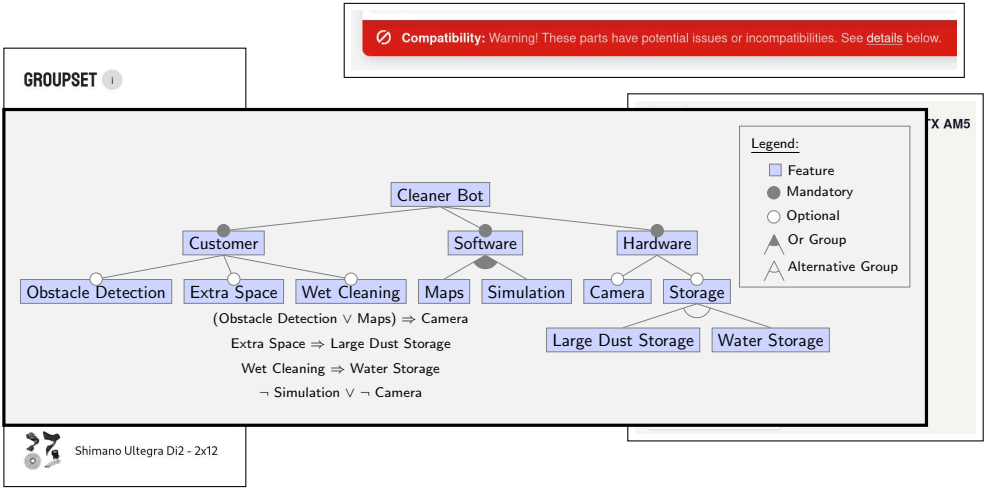


Motivation Feature Dependencies

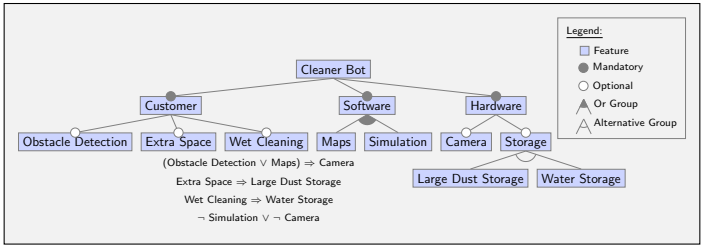



```
config SECURITY_INFINIBAND
            bool "Infiniband Security Hooks"
            depends on SECURITY && INFINIBAND
```

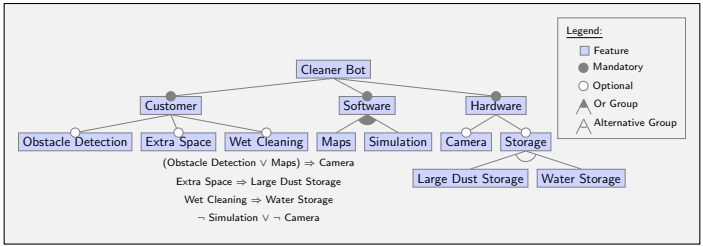
Motivation Feature Dependencies



Motivation Plenty Analyses

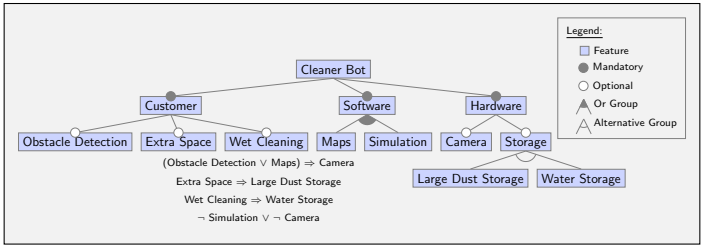


Motivation Plenty Analyses



Can we configure a robot?

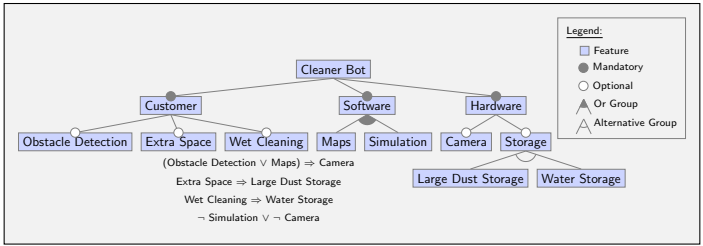
Motivation Plenty Analyses



Can we configure a robot?

Is there a robot with camera?

Motivation Plenty Analyses

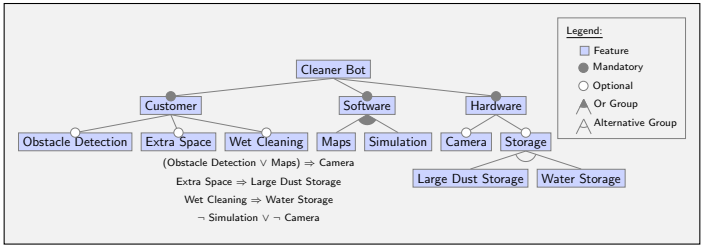


Can we configure a robot?

Is there a robot with camera?

How many robots?

Motivation Plenty Analyses



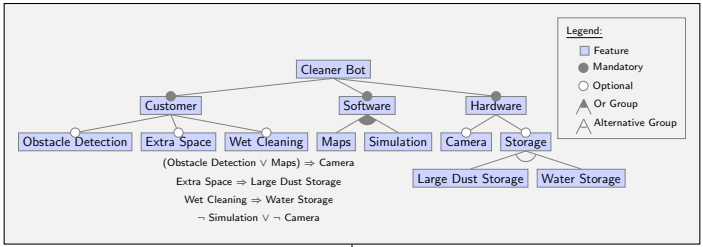
Can we configure a robot?

Is there a robot with camera?

How many robots?

Which robots to test?

Motivation Plenty Analyses



Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .

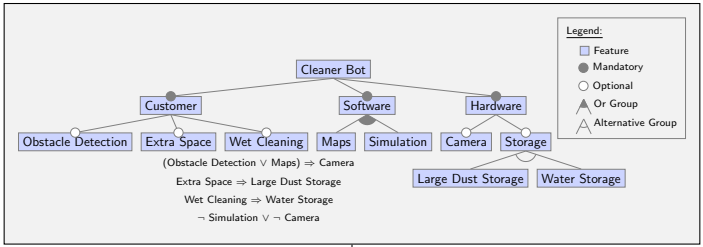
Can we configure a robot?

Is there a robot with camera?

How many robots?

Which robots to test?

Motivation Plenty Analyses



Can we configure a robot?

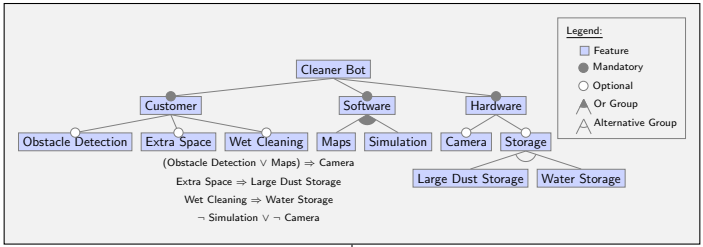
Is there a robot with camera?

How many robots?

Which robots to test?

Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .

Motivation Plenty Analyses



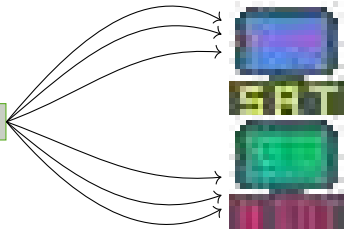
Can we configure a robot?

Is there a robot with camera?

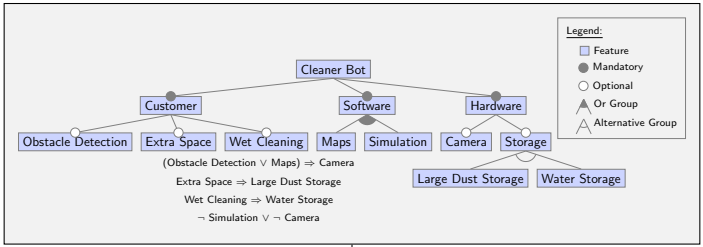
How many robots?

Which robots to test?

Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .

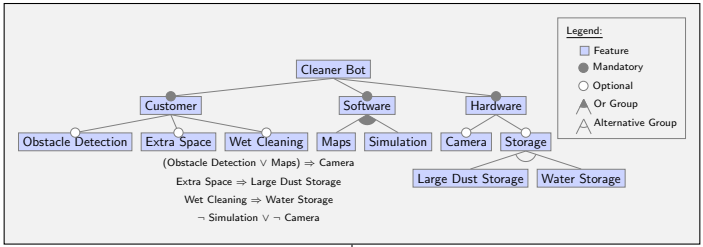


Motivation Knowledge Compilation



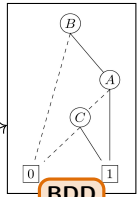
$$\text{Customer} \wedge \text{Software} \wedge (\text{Maps} \vee \text{Simulation}) \wedge \ldots$$

Motivation Knowledge Compilation

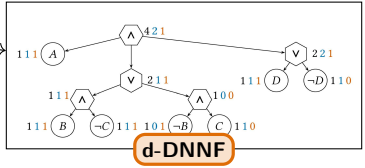


Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .

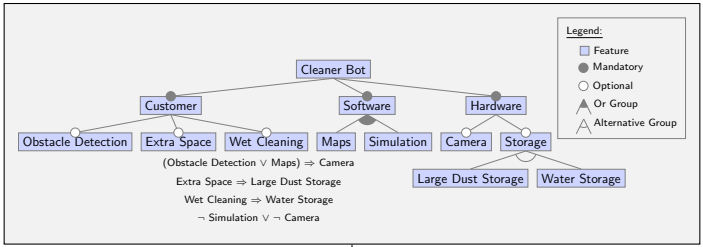
Compile



Compile



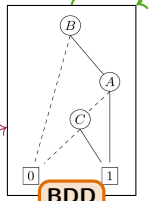
Motivation Knowledge Compilation



Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .

One-time expensive

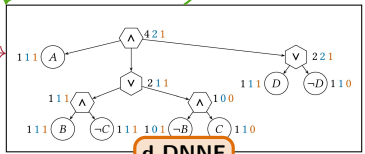
Many cheap queries



BDD

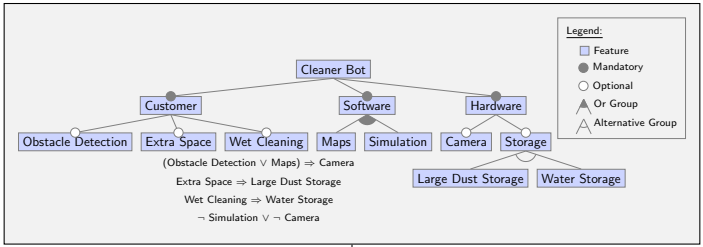
One-time expensive

Many cheap queries

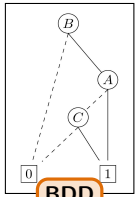


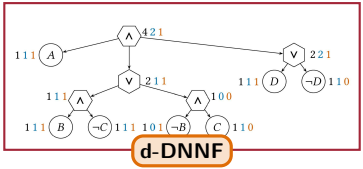
d-DNNF

Motivation Knowledge Compilation

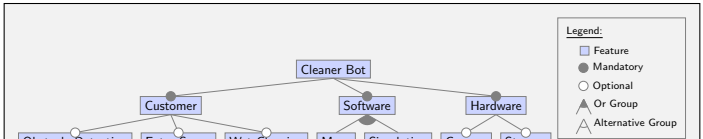


Customer ∧ Software ∧ (Maps ∨ Simulation) ∧ . . .





Motivation Knowledge Compilation

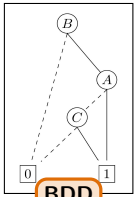


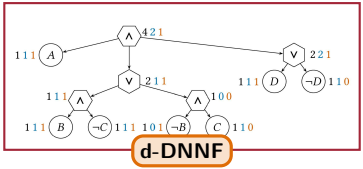
Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-023-09906-6

Check for updates

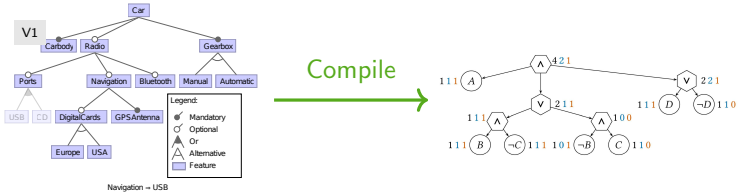
On the benefits of knowledge compilation for feature-model analyses

Chico Sundermann[1] · Elias Kuiter[2] · Tobias Heß[1] · Heiko Raab[1] · Sebastian Krieter[1] · Thomas Thüm[1]

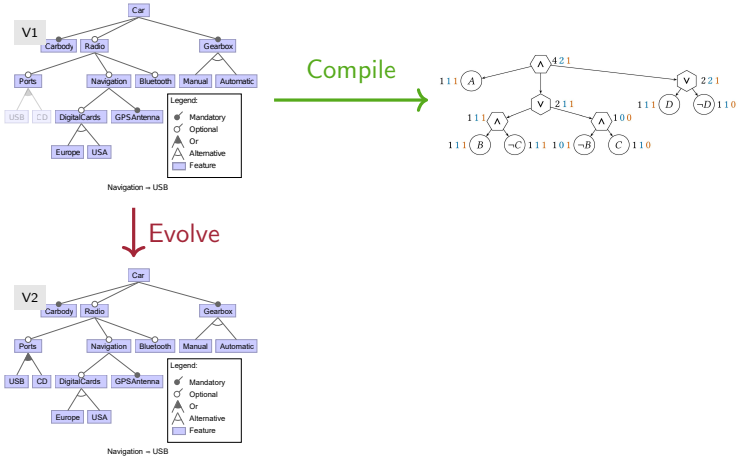




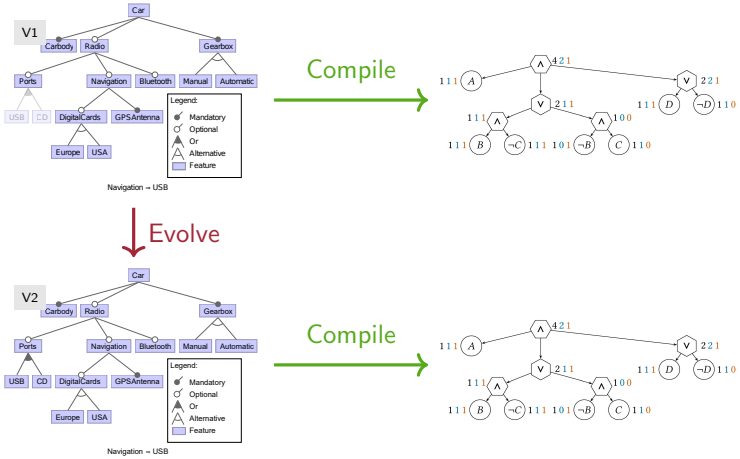
Incremental d-DNNF Compilation



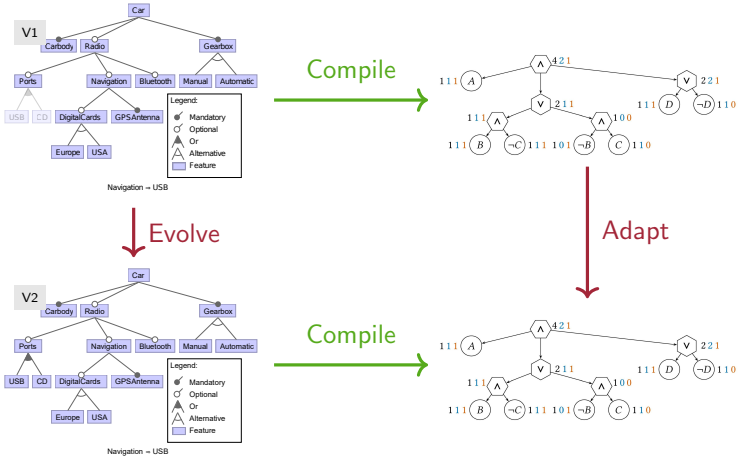
Incremental d-DNNF Compilation



Incremental d-DNNF Compilation

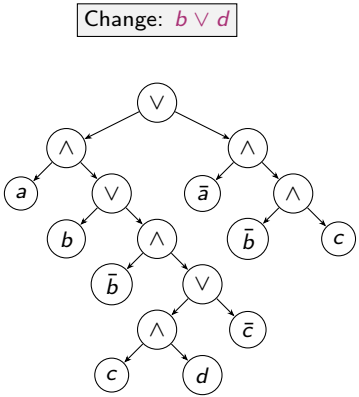


Incremental d-DNNF Compilation



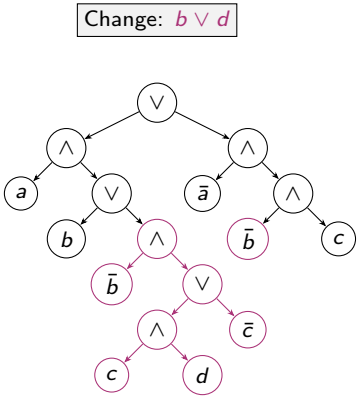
Approach Identify Part Requiring Update

- Recurse d-DNNF to find parts sharing variables with clause
- Apply assumptions while recursing to skip parts unchanged by clause

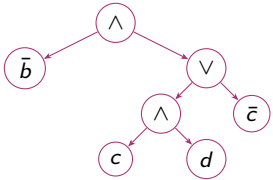


Approach Identify Part Requiring Update

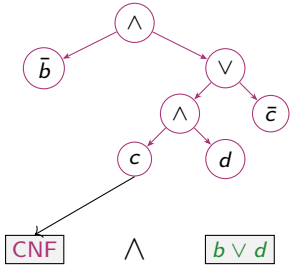
- Recurse d-DNNF to find parts sharing variables with clause
- Apply assumptions while recursing to skip parts unchanged by clause



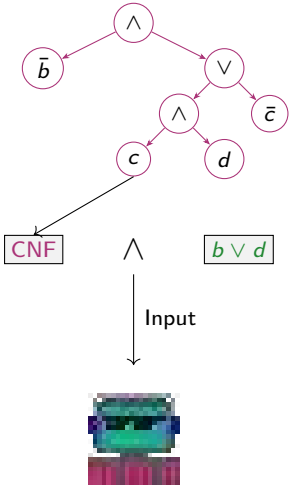
Approach Partially Recompile & Reattach



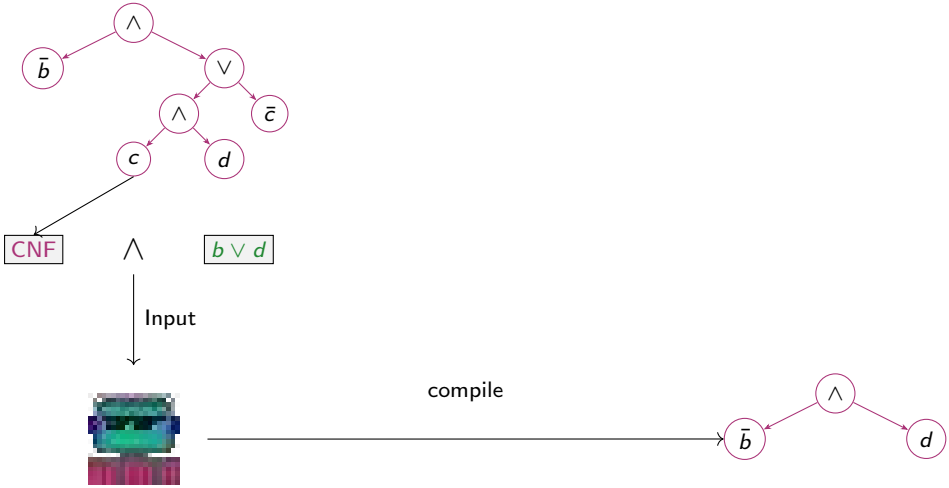
Approach Partially Recompile & Reattach



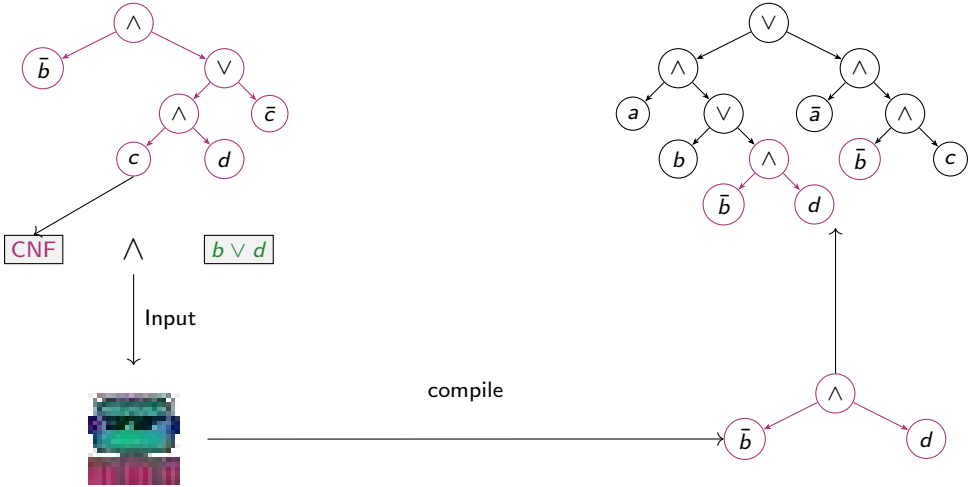
Approach Partially Recompile & Reattach



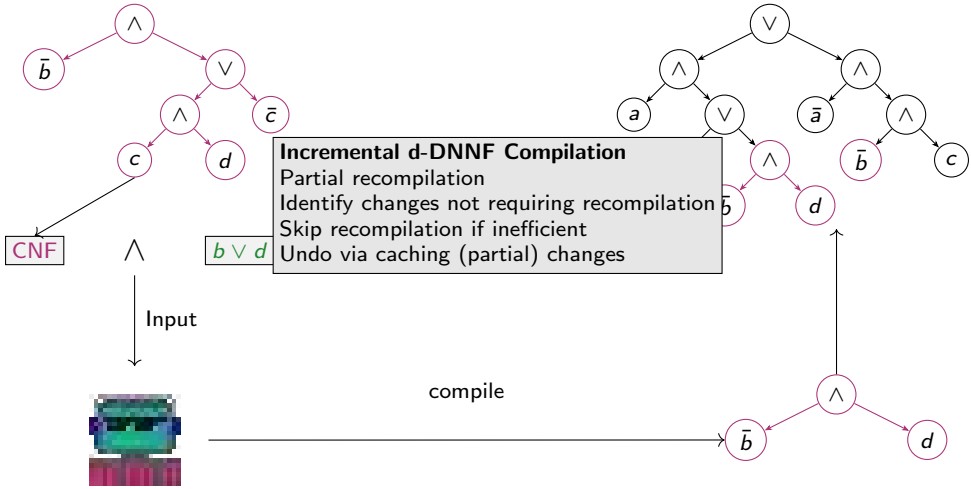
Approach Partially Recompile & Reattach



Approach Partially Recompile & Reattach



Approach Partially Recompile & Reattach



Evaluation Experiment Design

- Goal: Examine performance benefits

Domain	Models	Features	Clauses	Cardinality
automotive	2	$2,513 - 18,616$	$10,311 - 350,149$	$5.3^{210} - 1.7^{1534}$
business	1	$1,920$	$61,362$	2.0^{397}
cloud	1	106	156	5.4^{20}
database	1	117	417	3.2^{1}
deep-learning	2	$3,296 - 6,867$	$8,004 - 17,431$	$1.3^{185} - 2.8^{630}$
e-commerce	2	$287 - 2,238$	$421 - 2,615$	$2.3^{49} - 8.0^{558}$
finance	3	$176 - 771$	$280 - 7,238$	$9.7^{13} - 5.3^{31}$
games	1	144	769	4.2^{18}
hardware	2	$172 - 377$	$309 - 1,356$	$3.3^{21} - 2.3^{27}$
navigation	2	$103 - 145$	$185 - 378$	$3.2^{10} - 1.8^{12}$
security	1	158	457	1.2^{14}
systems software	14	$96 - 1,580$	$183 - 15,692$	$8.3^{11} - 4.1^{409}$
text	1	137	179	1.3^{23}

Evaluation Experiment Design

- Goal: Examine performance benefits
- Subject systems
  - Industrial feature Models
  - Various domains, sizes

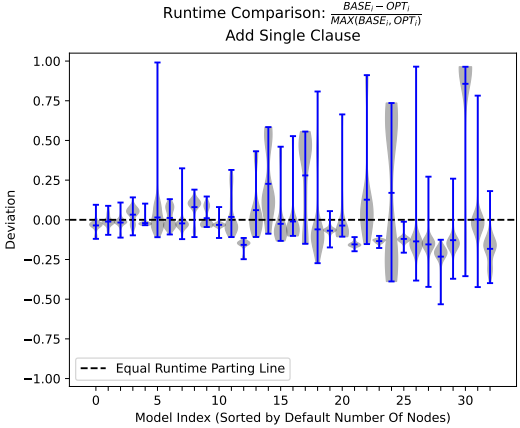
Domain	Models	Features	Clauses	Cardinality
automotive	2	$2,513 - 18,616$	$10,311 - 350,149$	$5.3^{210} - 1.7^{1534}$
business	1	$1,920$	$61,362$	2.0^{397}
cloud	1	106	156	5.4^{20}
database	1	117	417	3.2^{1}
deep-learning	2	$3,296 - 6,867$	$8,004 - 17,431$	$1.3^{185} - 2.8^{630}$
e-commerce	2	$287 - 2,238$	$421 - 2,615$	$2.3^{49} - 8.0^{558}$
finance	3	$176 - 771$	$280 - 7,238$	$9.7^{13} - 5.3^{31}$
games	1	144	769	4.2^{18}
hardware	2	$172 - 377$	$309 - 1,356$	$3.3^{21} - 2.3^{27}$
navigation	2	$103 - 145$	$185 - 378$	$3.2^{10} - 1.8^{12}$
security	1	158	457	1.2^{14}
systems software	14	$96 - 1,580$	$183 - 15,692$	$8.3^{11} - 4.1^{409}$
text	1	137	179	1.3^{23}

Evaluation Experiment Design

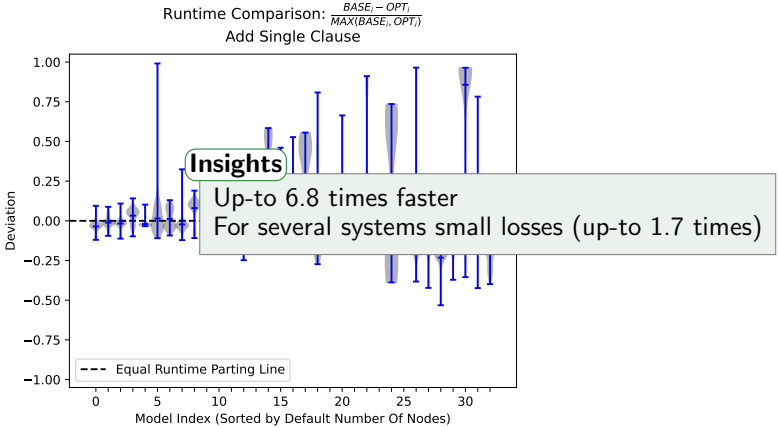
- Goal: Examine performance benefits
- Subject systems
  - Industrial feature Models
  - Various domains, sizes
- Emulate realistic changes
  - Remove one constraint → Compile without → Incrementally add removed
  - Apply typical edit patterns e.g.:
    - Add/remove feature
    - Change feature group
    - Move feature

Domain	Models	Features	Clauses	Cardinality
automotive	2	$2,513 - 18,616$	$10,311 - 350,149$	$5.3^{210} - 1.7^{1534}$
business	1	$1,920$	$61,362$	2.0^{397}
cloud	1	106	156	5.4^{20}
database	1	117	417	3.2^{1}
deep-learning	2	$3,296 - 6,867$	$8,004 - 17,431$	$1.3^{185} - 2.8^{630}$
e-commerce	2	$287 - 2,238$	$421 - 2,615$	$2.3^{49} - 8.0^{558}$
finance	3	$176 - 771$	$280 - 7,238$	$9.7^{13} - 5.3^{31}$
games	1	144	769	4.2^{18}
hardware	2	$172 - 377$	$309 - 1,356$	$3.3^{21} - 2.3^{27}$
navigation	2	$103 - 145$	$185 - 378$	$3.2^{10} - 1.8^{12}$
security	1	158	457	1.2^{14}
systems software	14	$96 - 1,580$	$183 - 15,692$	$8.3^{11} - 4.1^{409}$
text	1	137	179	1.3^{23}

Evaluation Single Clause



Evaluation Single Clause

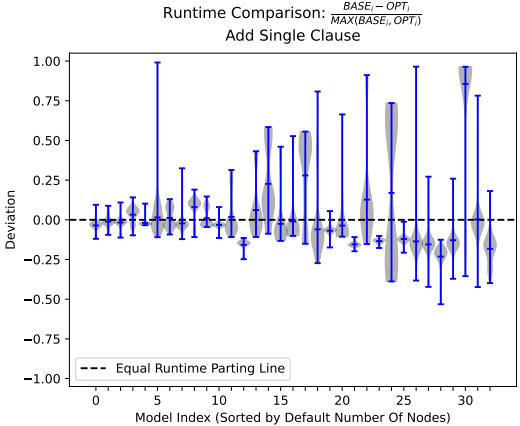


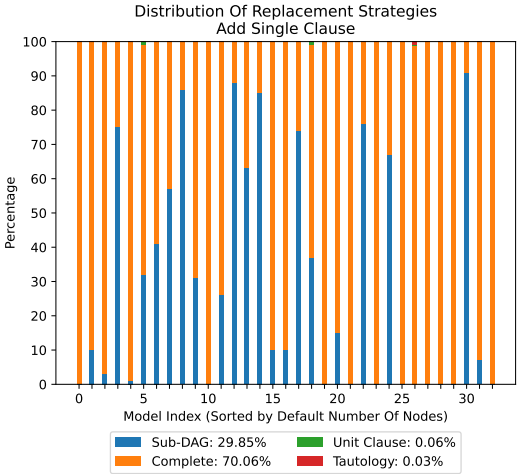
Insights

Up-to 6.8 times faster

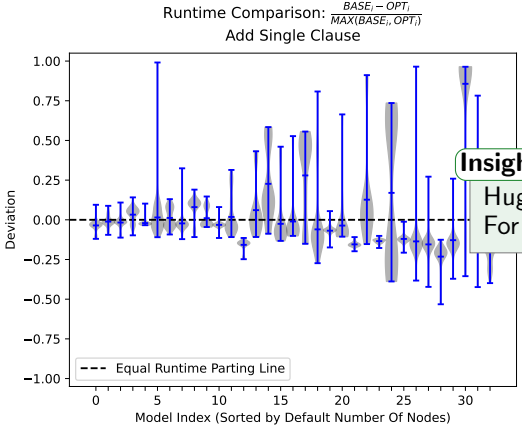
For several systems small losses (up-to 1.7 times)

Evaluation Single Clause



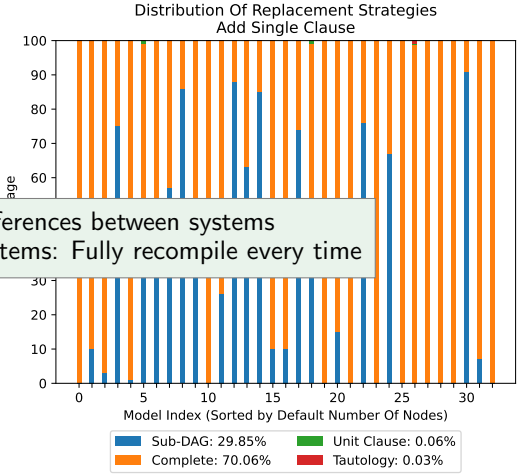


Evaluation Single Clause

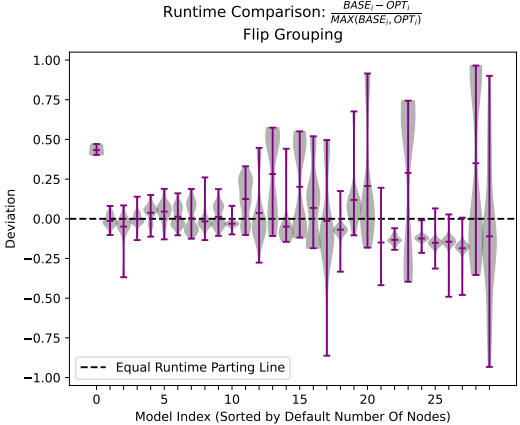


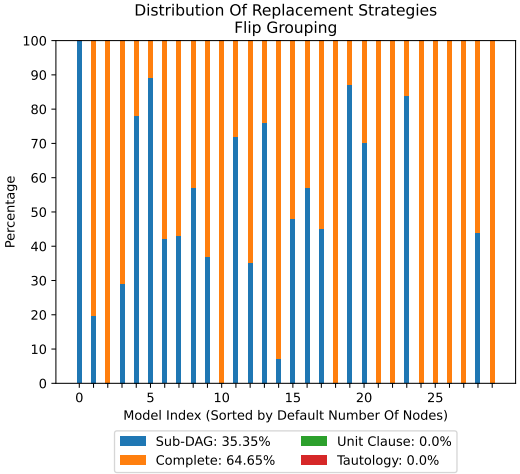
Insights

Huge differences between systems
For 6 systems: Fully recompile every time

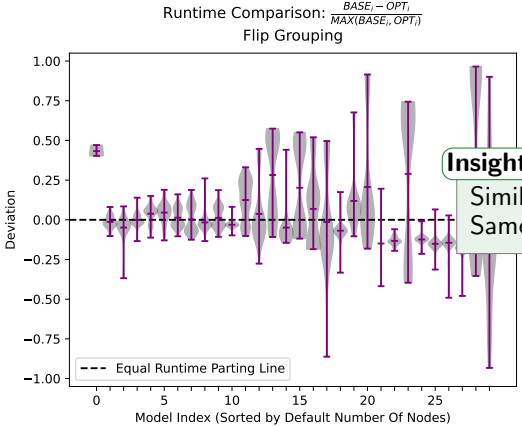


Evaluation Flip Group



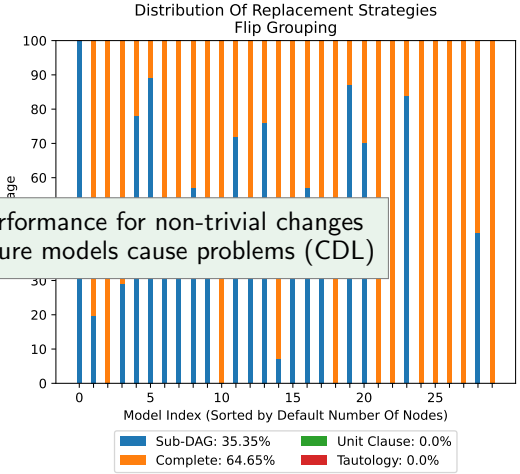


Evaluation Flip Group

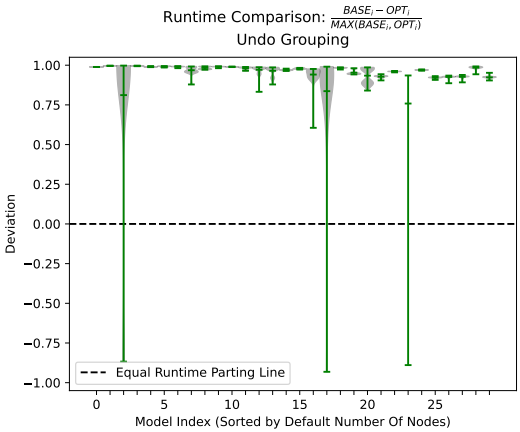


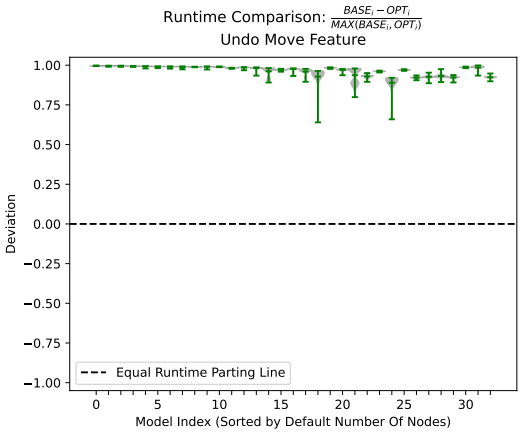
Insights

Similar performance for non-trivial changes
Same feature models cause problems (CDL)

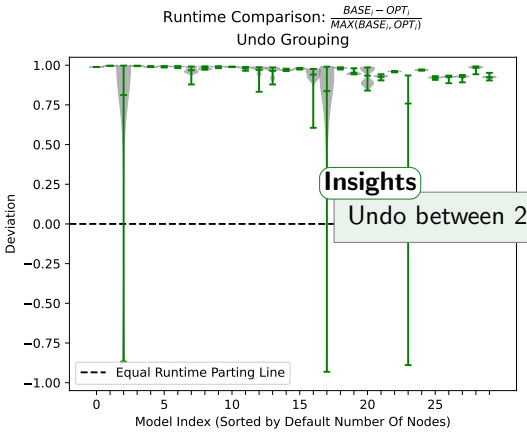


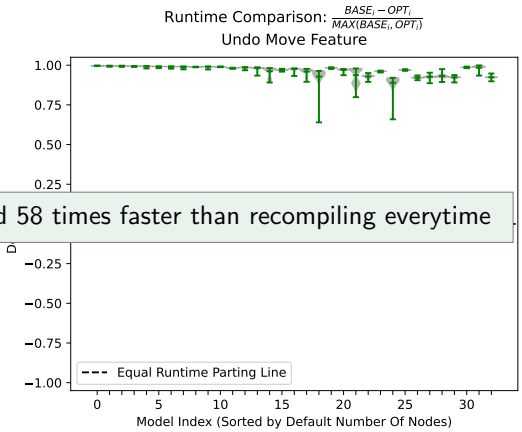
Evaluation Undo





Evaluation Undo





Insights

Undo between 23 and 58 times faster than recompiling everytime

Conclusion & Outlook

Insights

Substantial runtime improvements for some feature models
Undo by caching performs very well

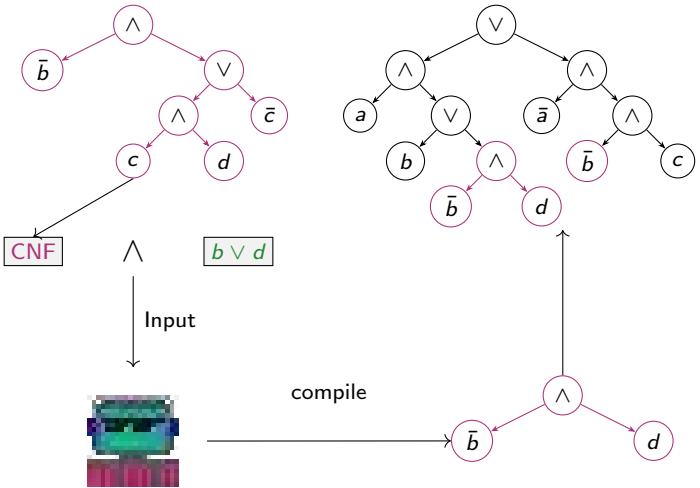
Limitations

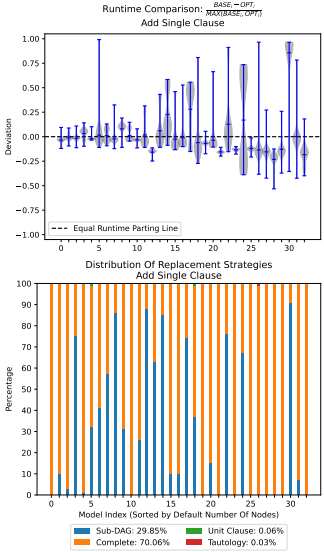
Performance highly depends on input model
Some models always need full recompilation

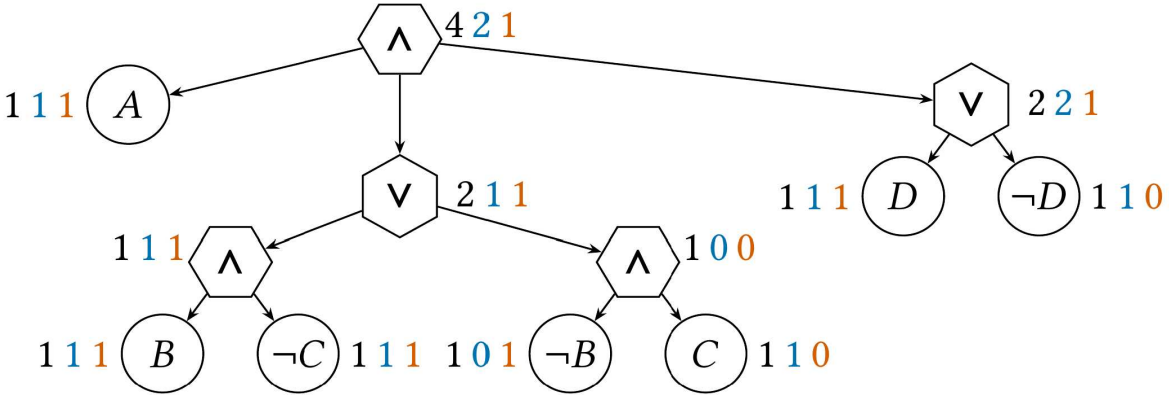
Future Work

How to improve for unfavourable d-DNNFs?
How to prepare the initial d-DNNF to be more favourable?
How to recognize hopeless d-DNNFs?

Incorporating Feature-Model Edits with Incremental d-DNNF Compilation







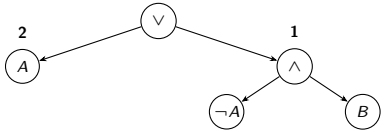
Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

How does this even work? d-DNNFs (Darwiche 2000)

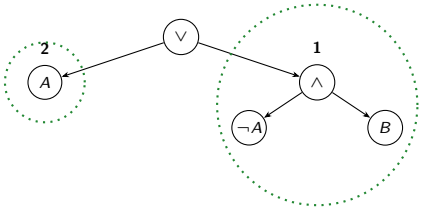
deterministic
decomposable
negation
normal
form

How does this even work? d-DNNFs (Darwiche 2000)



deterministic
decomposable
negation
normal
form

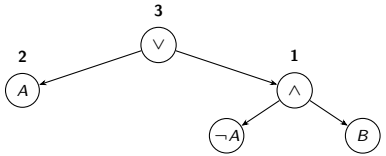
How does this even work? d-DNNFs (Darwiche 2000)



No shared solutions

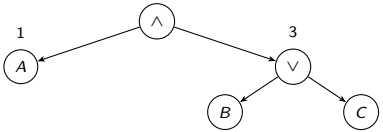
deterministic
decomposable
negation
normal
form

How does this even work? d-DNNFs (Darwiche 2000)



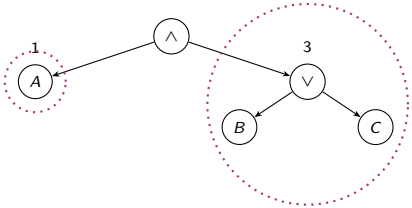
deterministic (Sum for $\vee$)
decomposable
negation
normal
form

How does this even work? d-DNNFs (Darwiche 2000)



deterministic (Sum for ∨)
decomposable
negation
normal
form

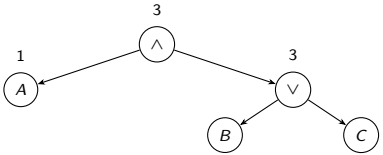
How does this even work? d-DNNFs (Darwiche 2000)



Independent
$\Rightarrow$ All pairs valid

deterministic (Sum for $\vee$)
decomposable
negation
normal
form

How does this even work? d-DNNFs (Darwiche 2000)



deterministic (Sum for $\vee$)
decomposable (Product for $\wedge$)
negation
normal
form