A Verified SAT Solver Framework including Optimization and Partial Valuations

Mathias Fleury[†] and Christoph Weidenbach[‡]

2021/01/12, LPAR-23

†

†‡

JOHANNES KEPLER UNIVERSITÄT LINZ †

max planck institut informatik

Introduction

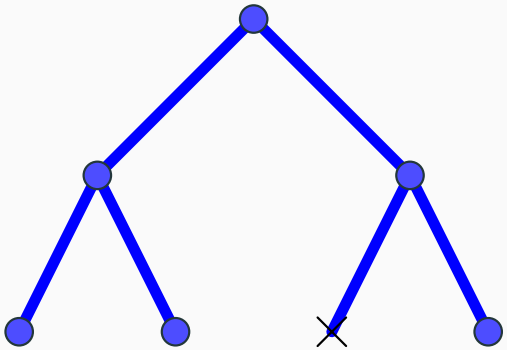
Motivating Example: a Car

Options	Constraints
blue, red, black	blue $\vee$ red $\vee$ black
bluetooth	armour $\rightarrow$ bluetooth
armour	armour $\rightarrow$ black

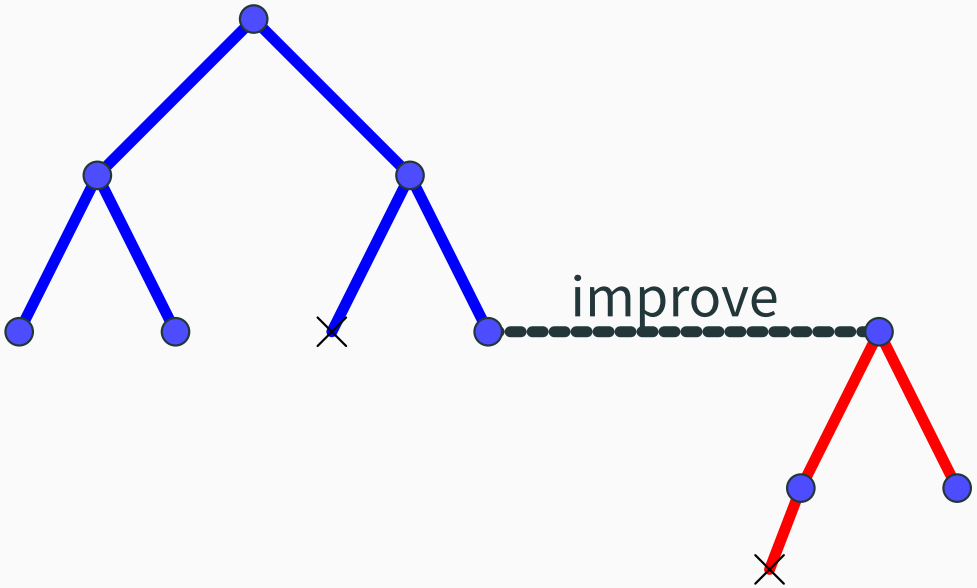
Option	Cost
color	1
bluetooth	with: 1, without: 0.5
armour	5
environment friendly	10

$$\text{weight}\ (M) = \sum_{\ell \in M} \text{weight}(L)$$

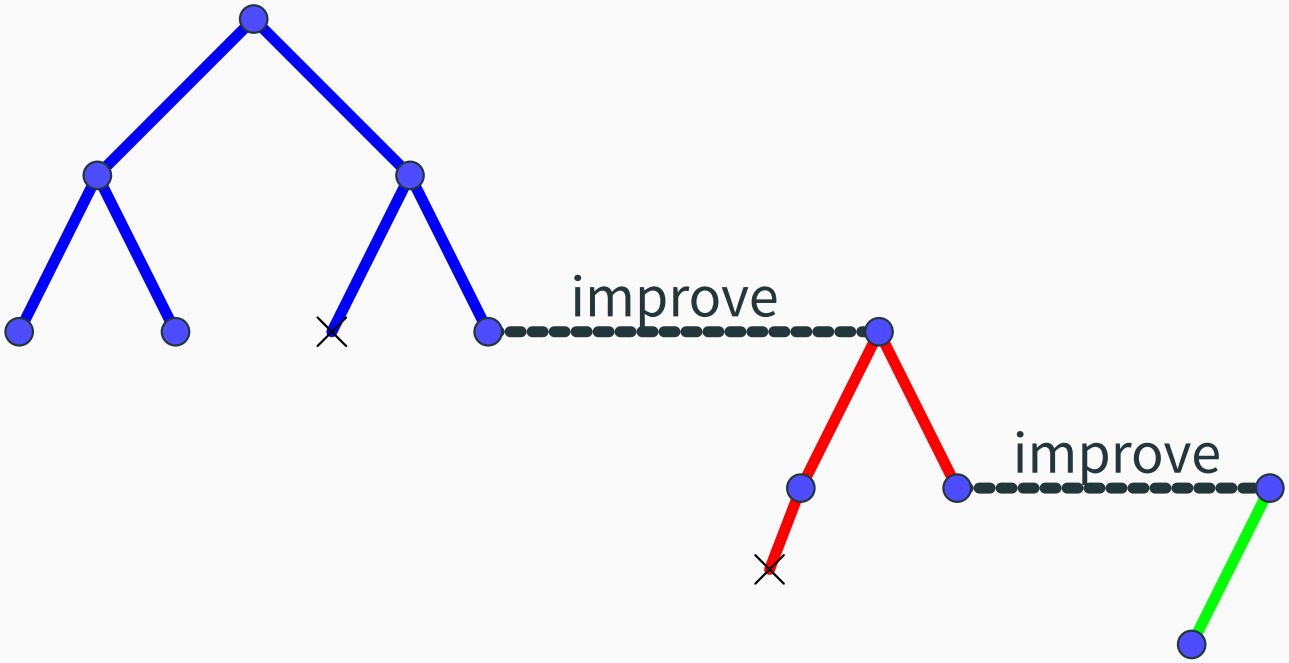
OCDCL = CDCL + optimizing



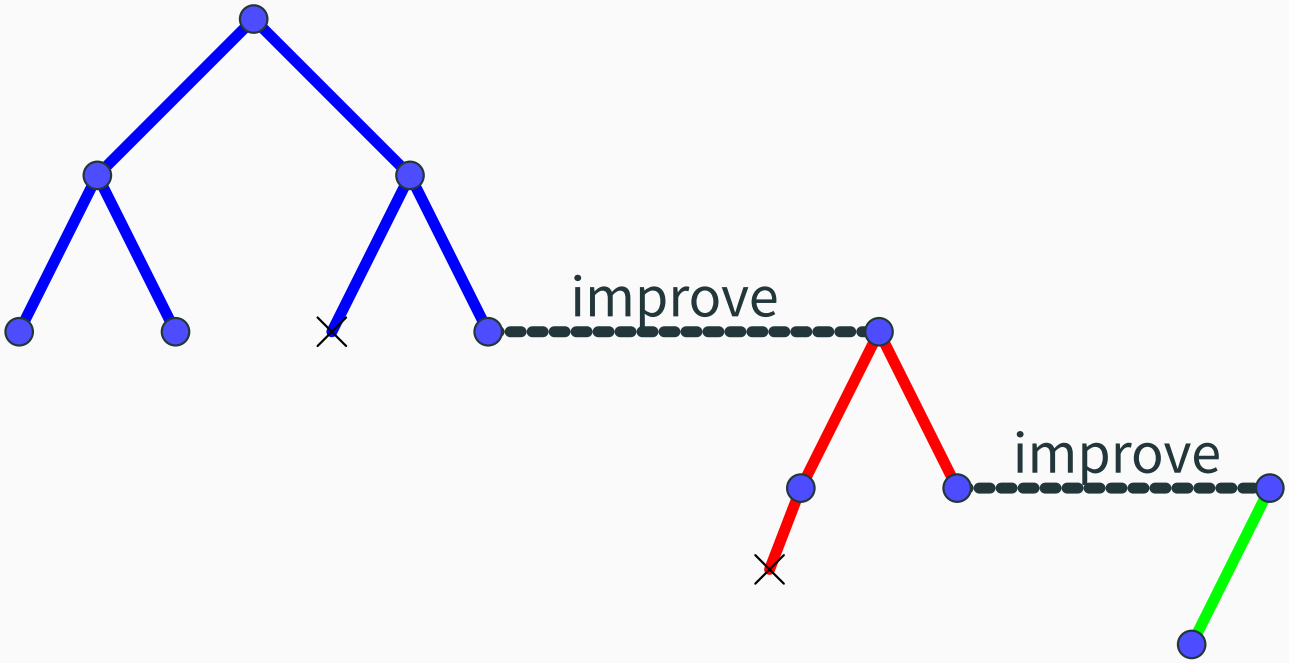
OCDCL = CDCL + optimizing



OCDCL = CDCL + optimizing



OCDCL = CDCL + optimizing



OCDCL = CDCL + conflictOpt + Improve

Optimizing CDCL and more

OCDCL

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save better models in O
ConflictOpt	add conflicts in S, if search cannot improve model

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S, ~~if search cannot improve model~~ based on O

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S, ~~if search cannot improve model~~ based on O

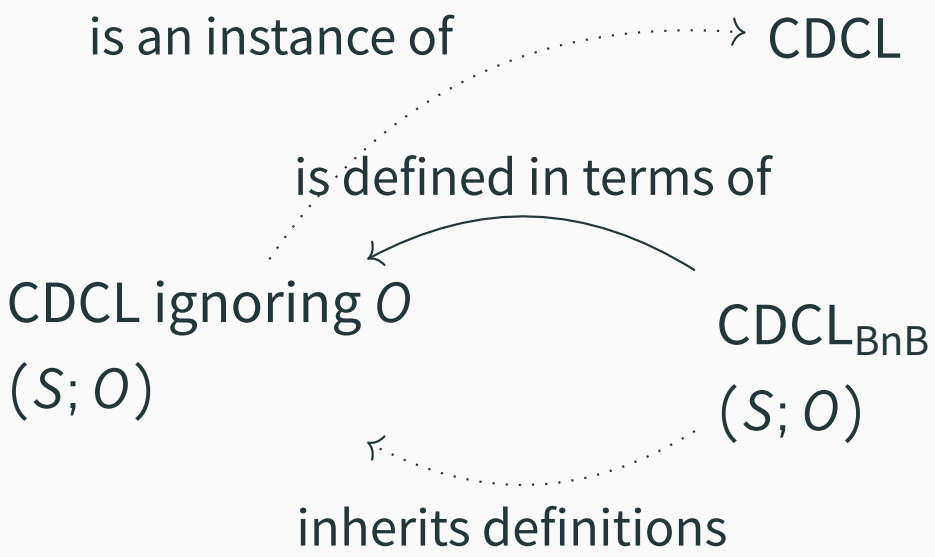
Does not work for partial models

The opposite of L is not $\neg L$, but $\neg L$ or L undefined.

Existing CDCL

```
locale CDCL_locale =
  fixes prepend_trail :: lit => 'st => 'st and
        trail  :: 'st => lits
  assumes !!L S. trail(prepend_trail L S) =
     L·trail S and
  ...
```

Isabelle Formalization



Reusing Definitions

```
locale OCDCL_locale =
  ...
begin
  interpretation CDCL_locale where
      prepend_trail = prepend_trail and
      trail = trail
  (*definitions are for free*)
```

Reusing Properties (I)

ConflictOpt add conflicts in S, ~~if search cannot improve model~~ based on O

So we consider CDCL over all clauses and all clauses that are based on O (CDCL_e)

Reusing Properties (I)

ConflictOpt add conflicts in S, ~~if search cannot improve model~~ based on O

So we consider CDCL over all clauses and all clauses that are based on O (CDCL_e)

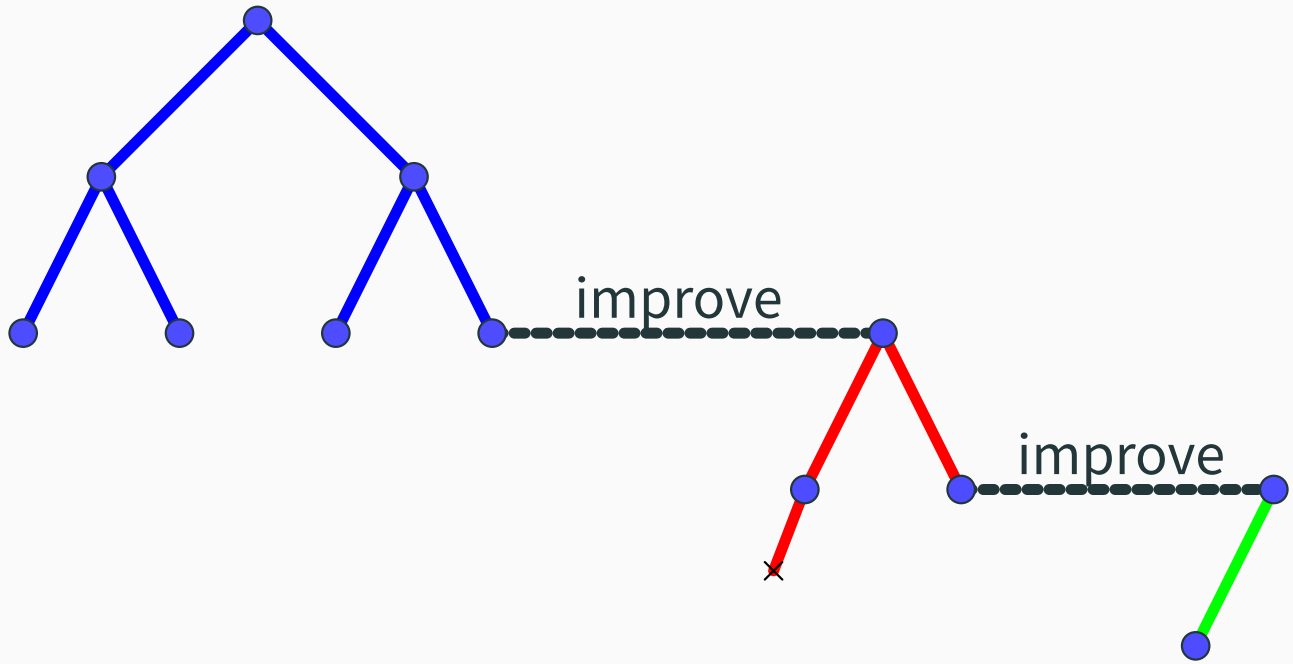
So, ConflictOpt becomes normal CDCL conflict application!

Reusing Properties (II)

$$\text{OCDCL} = \underbrace{\text{CDCL} + \text{conflictOpt}}_{\subseteq \text{CDCL}_e} + \text{improve}$$

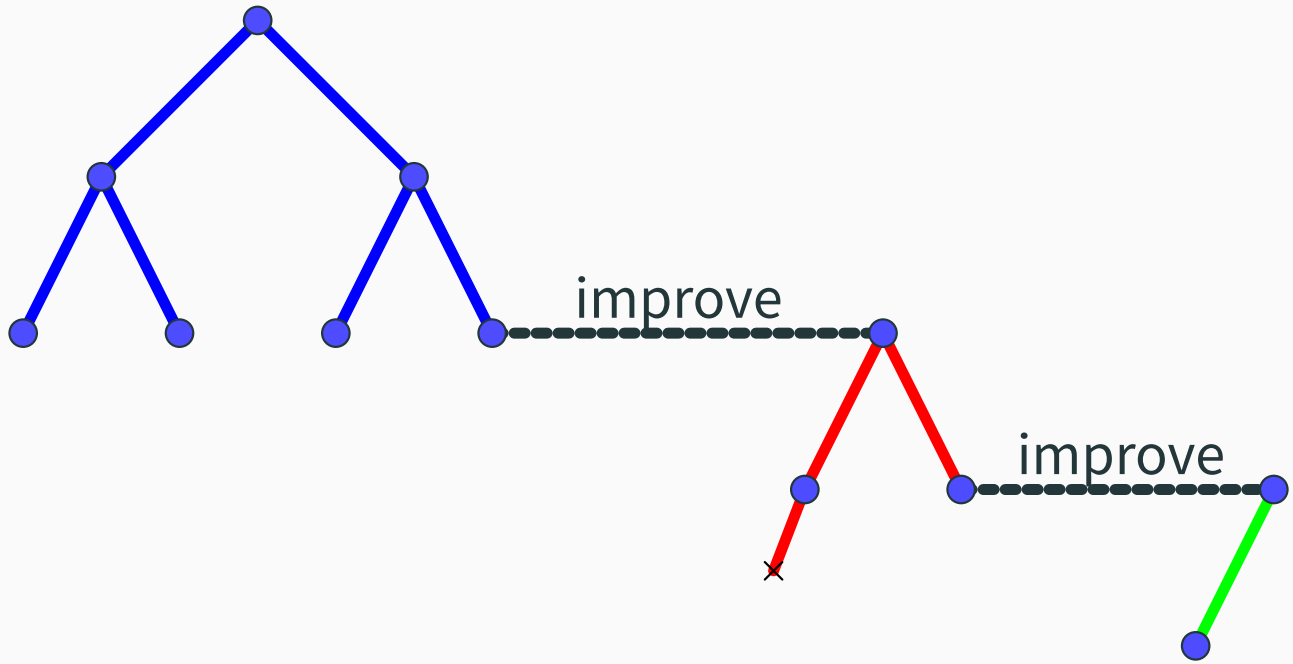
Reusing Properties (II)

$$\text{OCDCL} = \underbrace{\text{CDCL} + \text{conflictOpt}}_{\subseteq \text{CDCL}_e} + \text{improve}$$



Reusing Properties (II)

$$\text{OCDCL} = \underbrace{\text{CDCL} + \text{conflictOpt}}_{\subseteq \text{CDCL}_e} + \text{improve}$$

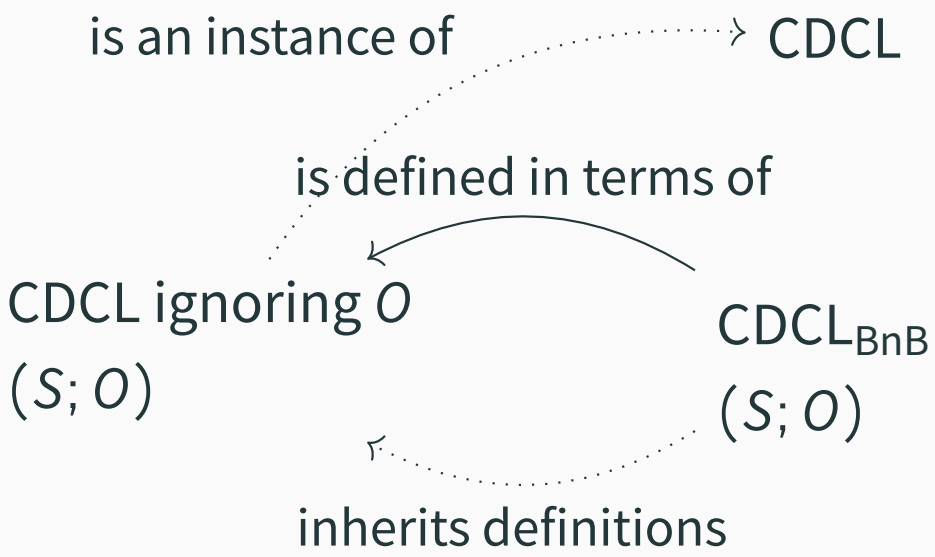


(S, O) are manipulated independently

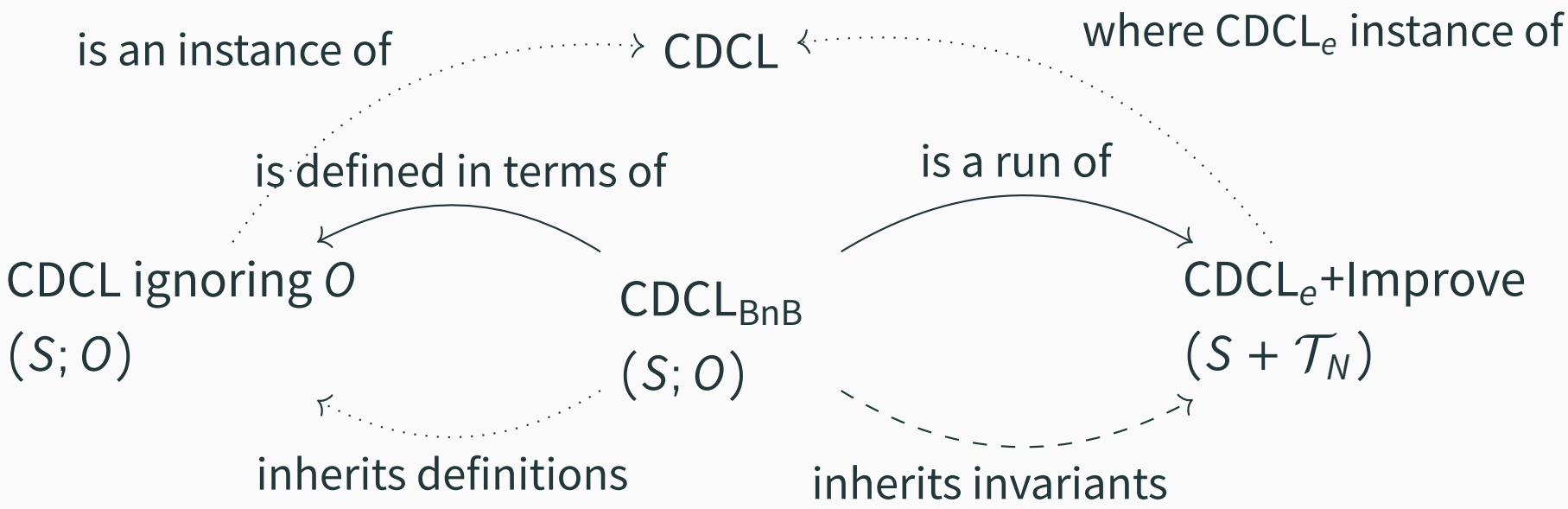
Reusing Properties (III)

```
locale OCDCL_locale =
  ...
begin
  interpretation CDCLe: CDCL_locale where
      prepend_trail = prepend_trail and
      trail = trail and
      clauses = clauses + clausesO
  (*some theorems are for free*)
```

Isabelle Formalization



Isabelle Formalization



Termination (Paper)

$$\text{OCDCL} = \text{CDCL} + \text{conflictOpt} + \text{improve}$$

Theorem

OCDCL terminates.

Proof.

Adapted proof (the measure) from the CDCL case. □

Termination (Isabelle)

$$\text{OCDCL} = \underbrace{\text{CDCL} + \text{conflictOpt}}_{\subseteq \text{CDCL}_e} + \text{improve}$$

Theorem

OCDCL terminates.

Proof.

CDCL_e terminates without changing O. Improve terminates without changing the CDCL state. □

From CDCL_{BnB} to OCDCL

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save better models in O
ConflictOpt	add conflicts in S, if search cannot improve model

From CDCL_{BnB} to OCDCL

We use the state (S, O)

CDCL rules from standard CDCL on S

Improve save ~~better models~~ something in O

ConflictOpt add conflicts in S, ~~if search cannot improve model~~ based on O

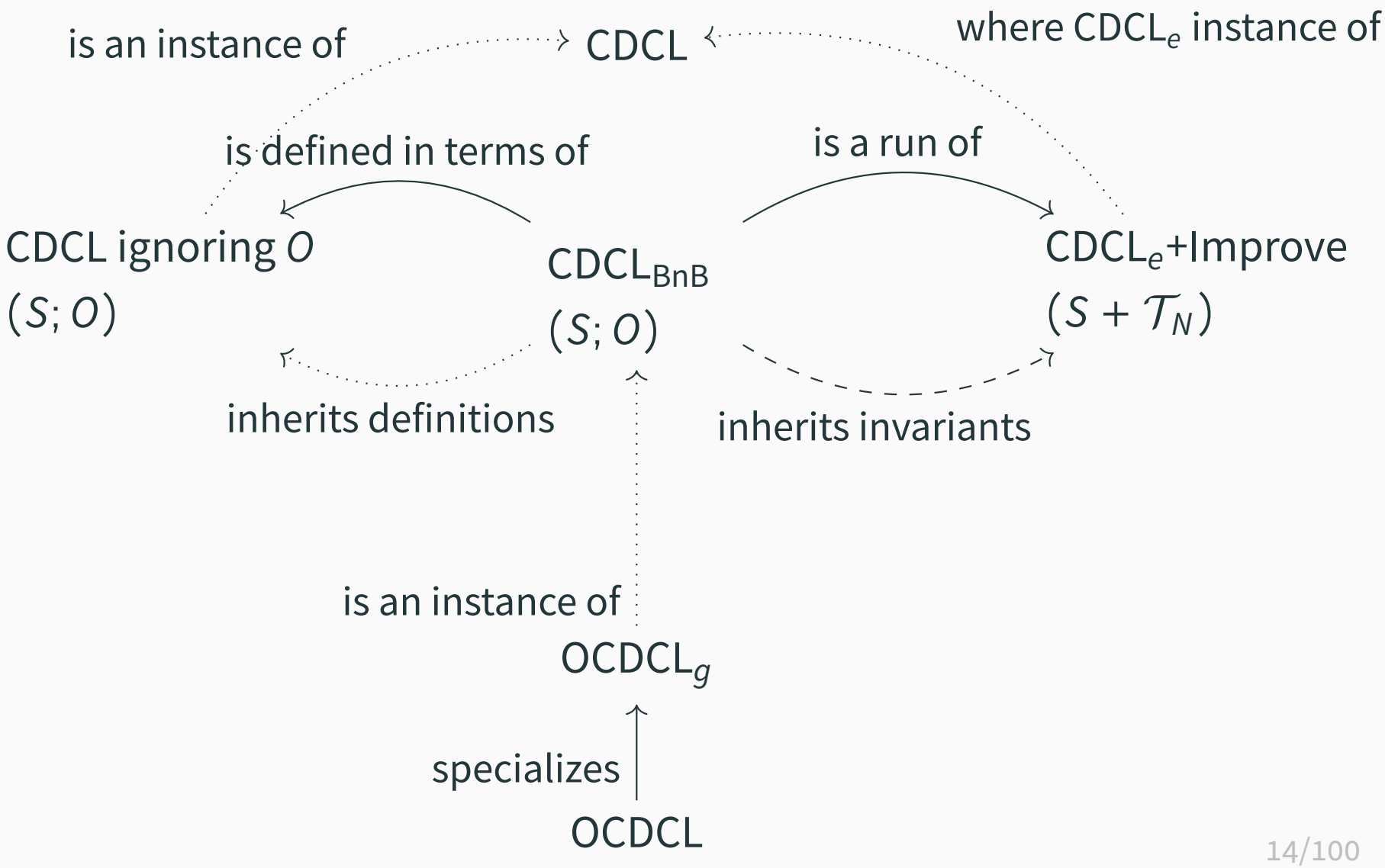
From CDCL_{BnB} to OCDCL

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S, ~~if search cannot improve model~~ based on O

Improve	saves best models found so far
ConflictOpt	add conflicts in S if their weight is larger than the best so far

Isabelle Formalization



Formalization length

Calculus	length (loc)
$\mathrm{CDCL}_{\mathrm{BnB}}$	1600
OCDCLg	1200
OCDCL	600

Model Covering

In a configuration system, can every option be chosen?

Isabelle Formalization

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S,~~if search cannot improve model~~ based on O

CDCL + Model Covering:

Isabelle Formalization

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S,~~if search cannot improve model~~ based on O

CDCL + Model Covering:

Isabelle Formalization

We use the state (S, O)

CDCL	rules from standard CDCL on S
Improve	save ~~better models~~ something in O
ConflictOpt	add conflicts in S,~~if search cannot improve model~~ based on O

CDCL + Model Covering:

Improve	save new model found so far
ConflictOpt	add conflicts in S if they have already been found

Isabelle Formalization

The model covering is not optimal (that is another NP-hard problem)

But (future work) minimization fits into CDCL_{BnB}.

Conclusion

Conclusion

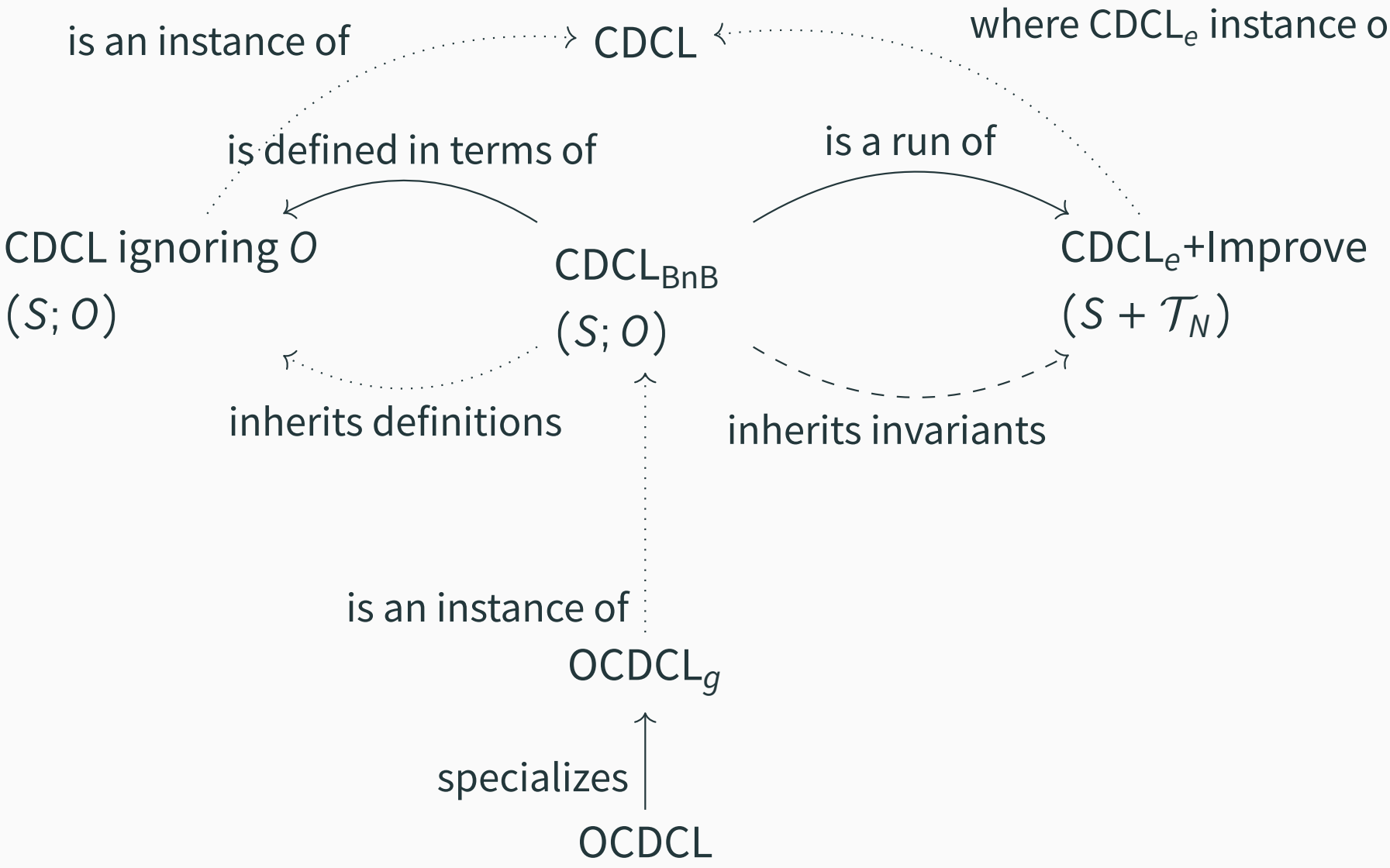
- Formalization of CDCL_{BnB}, instantiated with miminal weight and model covering
- Formalization was easy to do
- Dual rail way encoding to find partial optimal model
  - based on adding new literals s.t. $\neg L'$ can be undefined
  - a reviewer pointed out a better solution, half day to fix

Conclusion

- Formalization completes paper proof, by forcing things to be more precise
- ... and become easier with libraries
- part of IsaFoL[1]

Close to CDCL($\mathcal{T}$), except for propagation, but CDCL_e (unlike CDCL_{BnB}) already supports some.

[1] https://bitbucket.org/isafol/isafol/


is an instance of
CDCL
where CDCL$_e$ instance of
is defined in terms of
is a run of
CDCL ignoring O
$(S; O)$
CDCL$_{\text{BnB}}$
$(S; O)$
CDCL$_e$+Improve
$(S + \mathcal{T}_N)$
inherits definitions
inherits invariants
is an instance of
OCDCL$_g$
specializes
OCDCL