

# Practical SAT Solving

## Lecture 8

Markus Iser, Dominik Schreiber, Tomáš Balyo | June 10, 2024



# Recap

## Lecture 7: Preprocessing

- Subsumption, BVE, BCE, Gates

## Today

### Propagation-based Redundancy Notions

# Preprocessing: Unit Propagation, Probing

## Propagation-based Redundancy

Let a formula  $F$ , a clause  $C \in F$ , and a literal  $x \in C$  be given.

- **Failed Literal Probing**

If  $F \wedge x \vdash_{UP} \perp$ , then  $F \models \neg x$ .

$\implies$  add  $\{\neg x\}$  to  $F$

- **Asymmetric Literal Elimination (ALE)**

If  $F \setminus C \wedge \overline{C \setminus \{x\}} \vdash_{UP} \bar{x}$ , then  $F \models C \setminus \{x\}$ .

$\implies$  strengthen  $C$  to  $C \setminus \{x\}$

- **Asymmetric Tautology Elimination (ATE)**

If  $F \setminus C \wedge \overline{C} \vdash_{UP} \perp$ , then  $F \models C$ .

$\implies$  remove  $C$  from  $F$

# Preprocessing: Unit Propagation, Probing

## Optimizations of Probing Techniques

- **Restricted Form of ATE/ALE**

Hidden Tautology Elimination (HTE), Hidden Literal Elimination (HLE) only propagate over binary clauses.

- **Avoidance of Redundant Propagations**

Sort literals and clauses in a formula to simulate a trie, and reuse propagations that share the same prefix. the binary implication graph and application of the *parenthesis theorem*.

- **Variants: Distillation / Vivification**

Interleave assignments and propagations to detect ATs / ALs early.

# Probing: Relationship with Proof Checking

## Generalizations of Blocked Clauses

- **Reverse Unit Propagation (RUP)**

A clause has the property RUP if and only if it is an Asymmetric Tautology (AT). All learned clauses are RUP at the moment of their learning. RUP checking is basic proof checking.

- **Resolution Asymmetric Tautologies (RATs)**

A clause  $C$  is a RAT in a formula  $F$  if it contains a literal  $x$  such that **each resolvent in  $C \otimes_x F_{\bar{x}}$  is an asymmetric tautology**. Modern proof systems are based on RATs.

## Recap.

### Recap.

- Classic Preprocessing Techniques: Subsumption, Self-subsuming Resolution, Bounded Variable Elimination, Blocked Clause Elimination
- Relationship between Preprocessing Techniques and Gate Encodings
- Probing Techniques: Failed Literal Probing, Asymmetric Literal Elimination, Asymmetric Tautology Elimination
- Relationship between Probing Techniques and Proof Checking

### Next Up

Scheduling of Preprocessing Techniques, Inprocessing, Autarky Reasoning

# Preprocessing: Scheduling of Preprocessing Techniques

At a point where one technique is unable to make further progress, another technique might be applicable and even modify the problem in a way that the first technique can make further progress.

## Scheduling of Preprocessing Techniques

- **Heuristic Limits**

Bound the number of applications of a technique.

- **Scheduling of Techniques**

Non-trivial, benefit of techniques depends on the formula.

- **Interleaving of Techniques**

Apply techniques in a round-robin fashion.

- **Inprocessing**

Interleave search and preprocessing.

# Inprocessing

## Idea: Interleave search and preprocessing

- Preprocessing can be extremely beneficial: most solvers in SAT competitions use bounded variable elimination, subsumption and self-subsuming resolution
- Problem: Many preprocessing techniques, though polynomial, require considerable time
- Possible Solution:
  - Interrupt preprocessing techniques after some time
  - Resume on restart
  - Limit preprocessing time in relation to search
- **Discussion:** What are the problems that can arise in practice when SAT instances are solved incrementally?



# Inprocessing: Autarkies

Autarky reasoning is used by state-of-the-art SAT solvers (cf. Kissat) to remove clauses from a formula.

## Autarky-based Clause Removal

Let a formula  $F$  and a partial assignment  $A$  be given.

- A clause  $C \in F$  is **touched** by  $A$  if it contains the negation of a literal assigned in  $A$
- A clause  $C \in F$  is **satisfied** by  $A$  if it contains a literal assigned to *True* by  $A$

An autarky is a partial assignment  $A$  such that **all touched clauses are satisfied**.

All clauses touched by an autarky can be removed.

**Discussion:** How to obtain partial assignments to probe for autarky-based clause removal?

## Autarky-based Clause Removal

The partial assignment  $A = \{\neg a, \neg c\}$  is an autarky for  $F := \{\{\neg a, b\}, \{\neg a, c\}, \{a, \neg b, \neg c\}\}$

# Techniques that do not Reduce the Formula Size

## Next Time

- **Resolution Calculus** (a.k.a. Clause Learning)
- **Tseitin's Extension Rule**: Introduce definitions of new variables as a conjunction of existing literals (a.k.a. Bounded Variable Addition (BVA)). Some formulas have refutations of exponential size in the resolution calculus, but of polynomial size in extended resolution, e.g., pigeonhole formulas, mutilated chessboard, ...
  - SBVA-CaDiCaL: Winner of SAT Competition 2023
- **Symmetry Breaking Predicates**: Exclusion of Symmetric Solutions
  - BreakId-Kissat: Special Price at SAT Competition 2023
- **PReLearning**: Preprocessing adds specific Propagation Redundant (PR) clauses
  - KissatMAB-Prop: Winner of SAT Competition 2023 on UNSAT instances