Hash-based Preprocessing and Inprocessing Techniques in SAT Solvers

Henrik Cao

Deparment of Computer Science
Aalto University

June 27, 2021

Aalto University

Content

- **Hash-based methods (Section 3)**
- **Probabilistic analysis (Section 4)**
- **Experimental results (Section 5)**

Hash-based methods

Processing techniques

- Subsumption algorithms [Bayardo and Panda, 2011]
- Variable Elimination [Eén and Biere, 2005]
- Blocked Clause Elimination [Järvisalo et al., 2010]

Subsumption

$C \subseteq D$ for clauses C, D.

Tautological resolvency

$C \otimes_l D = \top$ for clauses C, D with $l \in C$ and $\bar{l} \in D$.

Hash functions

$$h(C) = \sum_{i \in [C]_m} 2^i$$
$$[C]_m = \{|l| \bmod m \mid l \in C\}$$

$|8| \mod 8 = 0$

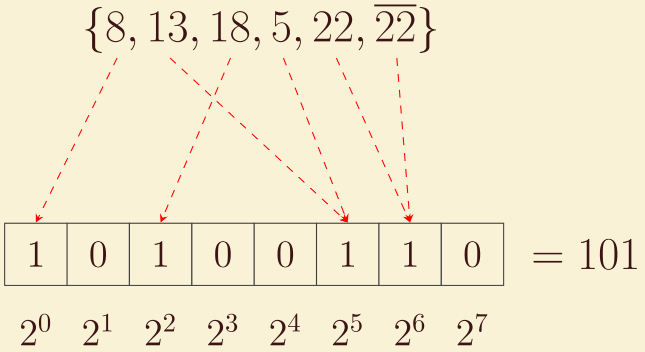
$|13| \mod 8 = 5$

$|18| \mod 8 = 2$

$|5| \mod 8 = 5$

$|22| \mod 8 = 6$

$|\text{-}22| \mod 8 = 6$

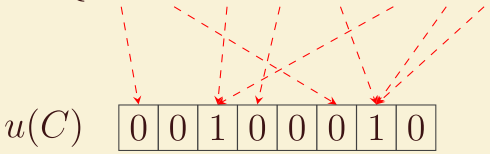


Collision signature

Collision signature

The collision signature $u(C)$ of a clause C and hash map h is the m-bit signature with the ith bit marked if h maps at least two literals in C to the corresponding index.

$$\{16, \overline{13}, 2, 11, 14, 10, 6, \overline{6}\}$$



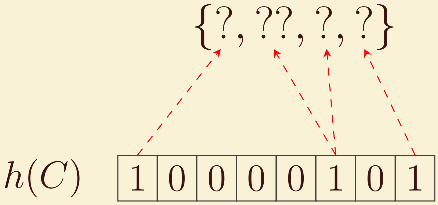
Clause relations

Subsumption

$C \subseteq D$ for clauses C, D.

Tautological resolvency

$C \otimes_l D = \top$ for clauses C, D with $l \in C$ and $\bar{l} \in D$.



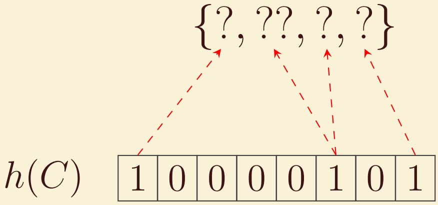
Clause relations

Subsumption

$C \subseteq D$ for clauses C, D.

Tautological resolvency

$C \otimes_l D = \top$ for clauses C, D with $l \in C$ and $\bar{l} \in D$.



$\Longrightarrow$ Inadmissible due to non-injectiveness of h.

Clause relations

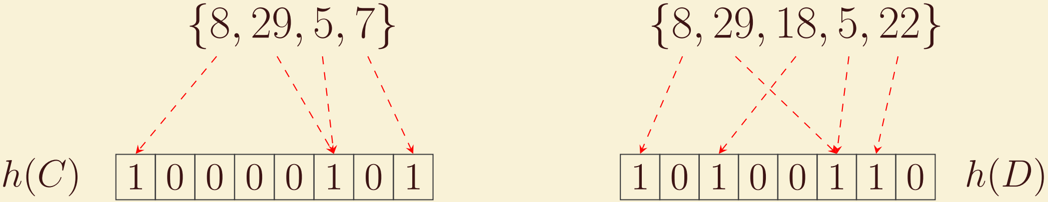
Non-subsumption

$C \not\subseteq D$ for clauses C, D.

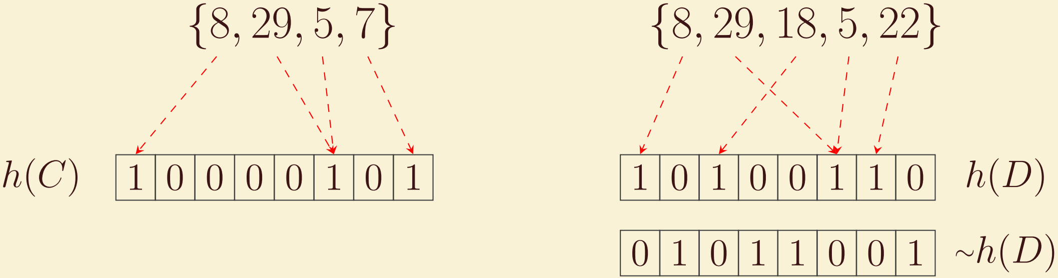
Non-tautological resolvency

$C \otimes_l D \neq \top$ for clauses C, D with $l \in C$ and $\bar{l} \in D$.

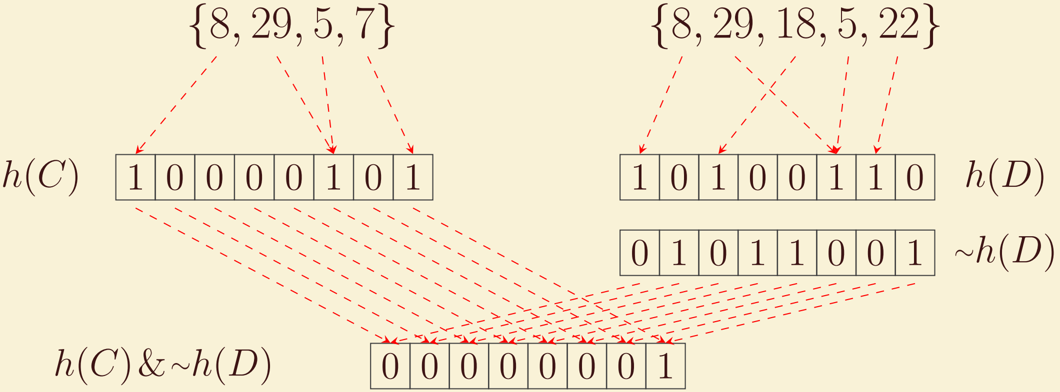
Non-Subsumption $C \not\subseteq D$



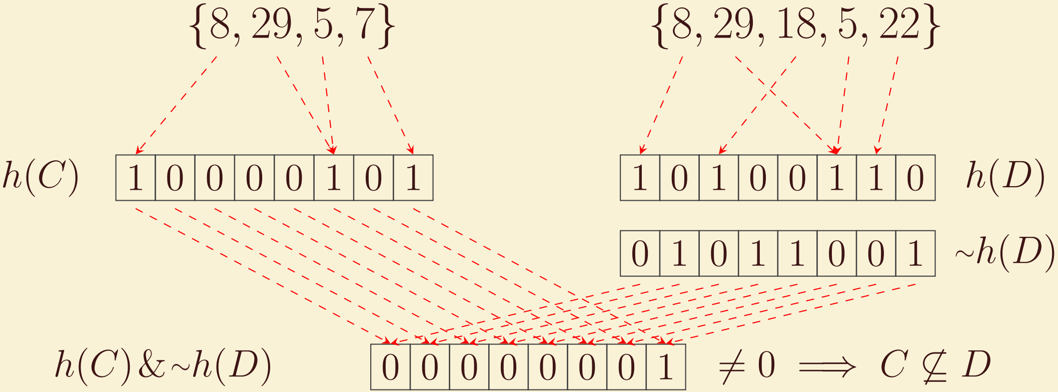
Non-Subsumption $C \not\subseteq D$



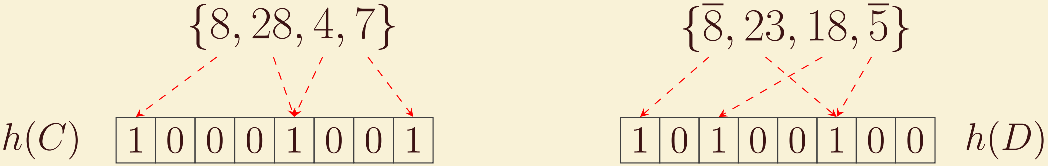
Non-Subsumption $C \not\subseteq D$



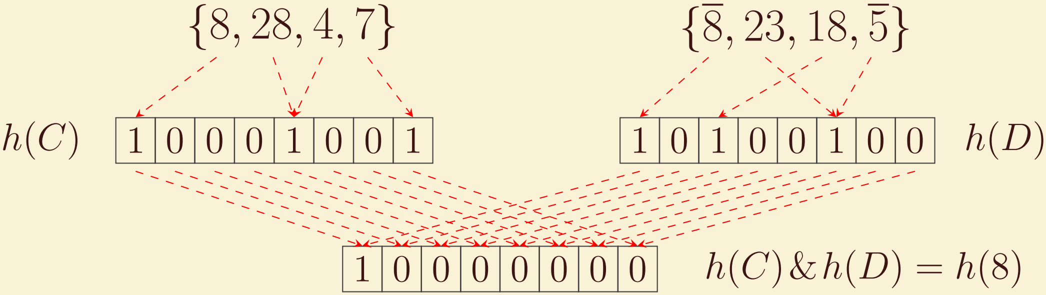
Non-Subsumption $C \not\subseteq D$



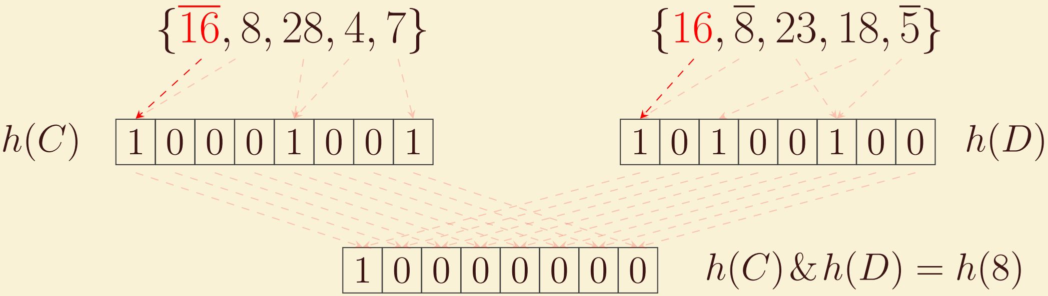
Non-tautological Resolvency $C \otimes_l D \neq \top$



Non-tautological Resolvency $C \otimes_l D \neq \top$

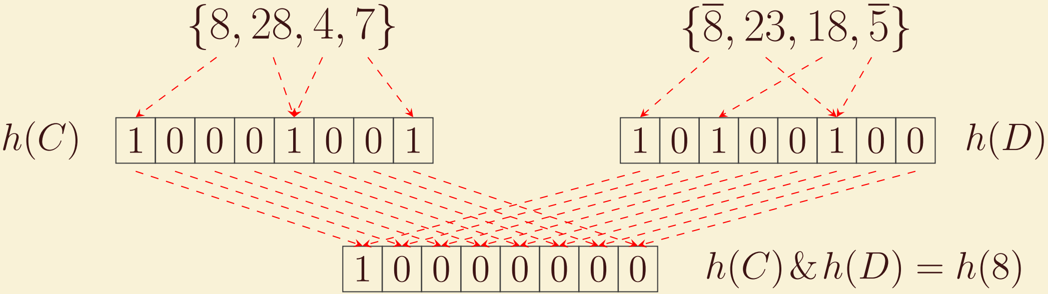


Non-tautological Resolvency $C \otimes_l D \neq \top$

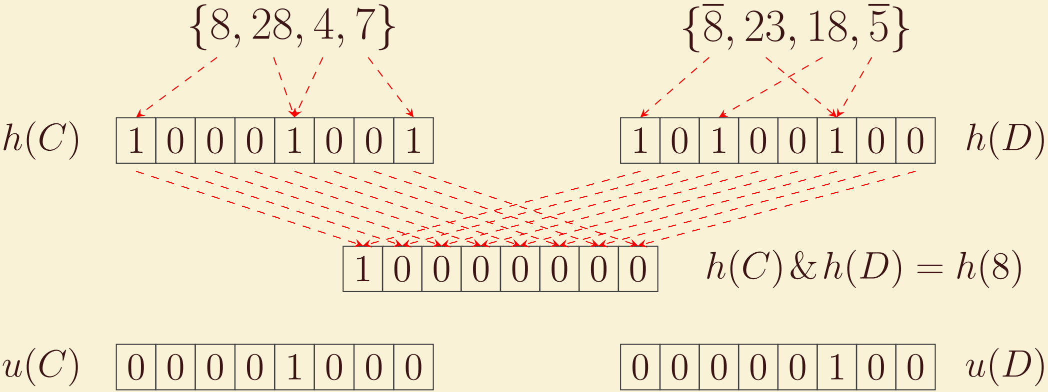


$$C \otimes_8 D = \top$$

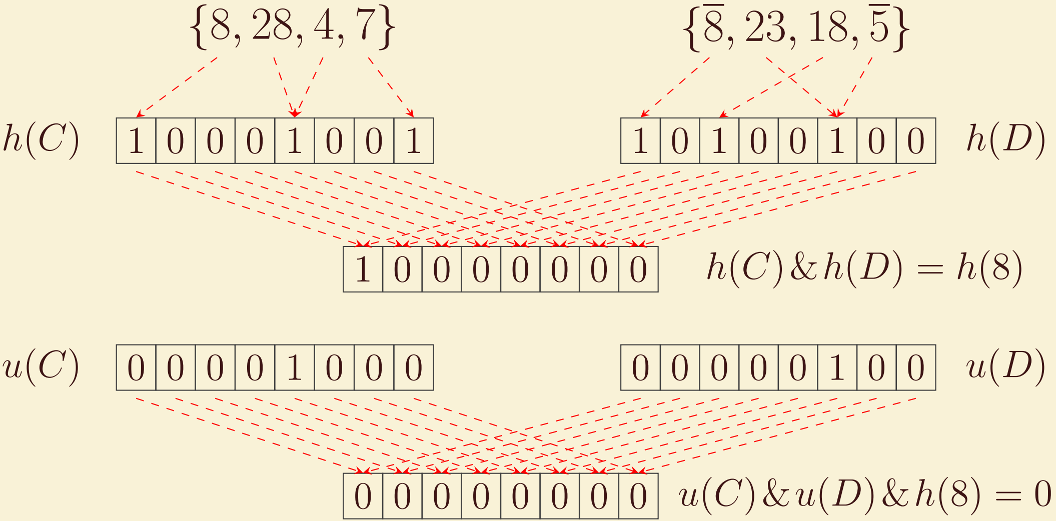
Non-tautological Resolvency $C \otimes_l D \neq \top$



Non-tautological Resolvency $C \otimes_l D \neq \top$



Non-tautological Resolvency $C \otimes_l D \neq \top$



Comparing Signatures

Proposition 1 (Non-subsumption)

Let $h \in \mathcal{H}$. If $h(C) \,\&\, {\sim}h(D) \neq 0$ *or* $u(C) \,\&\, {\sim}u(D) \neq 0$, then $C \not\subseteq D$.

Proposition 2 (Disjointness)

Let $h \in \mathcal{H}$. If $h(C) \,\&\, h(D) = 0$, then $C \cap D = \emptyset$.

Proposition 3 (Non-tautological resolvency)

Let $h \in \mathcal{H}$, $l \in C$ and $\bar{l} \in D$. If $h(C) \,\&\, h(D) = h(l)$ *and* $u(C) \,\&\, u(D) \,\&\, h(l) = 0$, then $C \otimes_l D$ is *non-tautological*.

Proposition 4 (Non-membership)

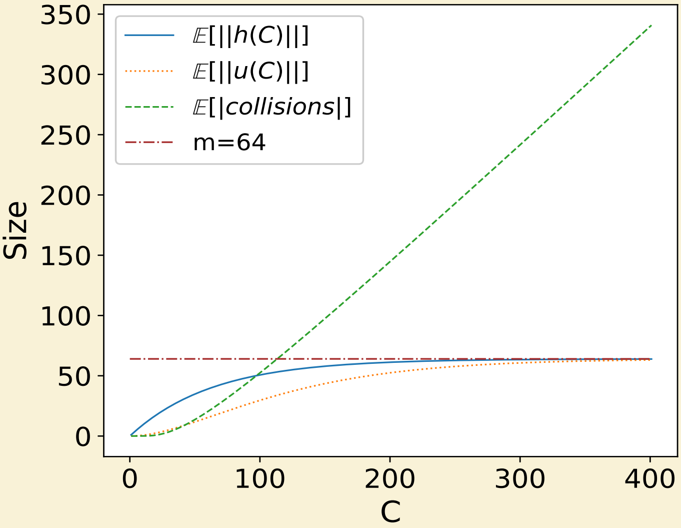
Let $h \in \mathcal{H}$. If $h(C) \,\&\, h(l) = 0$, then $l \notin D$.

Probabilistic Analysis

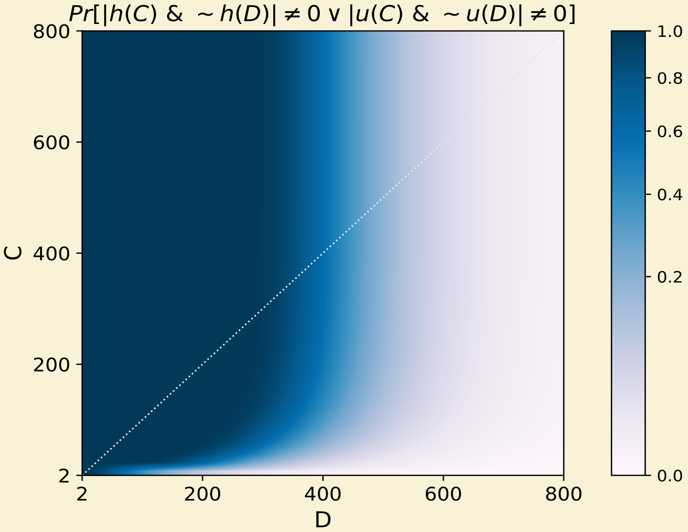
A family of hash functions

- $h \in \mathcal{H}$ maps variables indepdendently and uniformly at random.
- $h(l) = h(\bar{l})$, i.e., l and $\bar{l}$ map to the same index.
- $\|h(C)\|$ = number of bits set in $h(C)$.

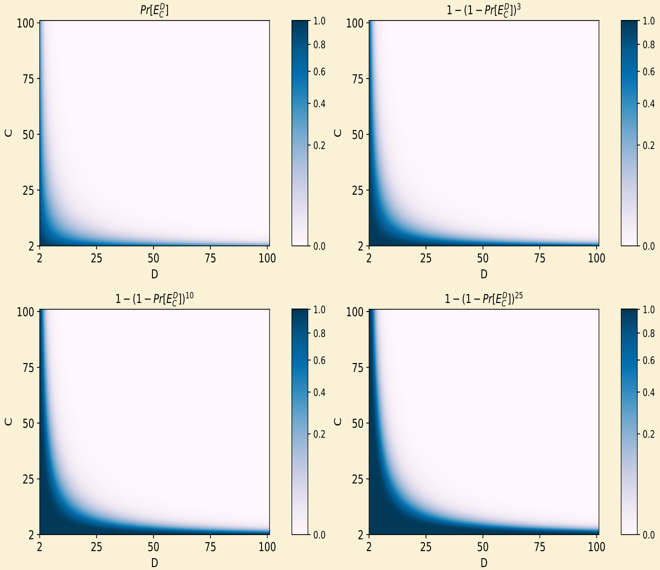
Clause signatures



Non-subsumption



Non-tautological resolvency



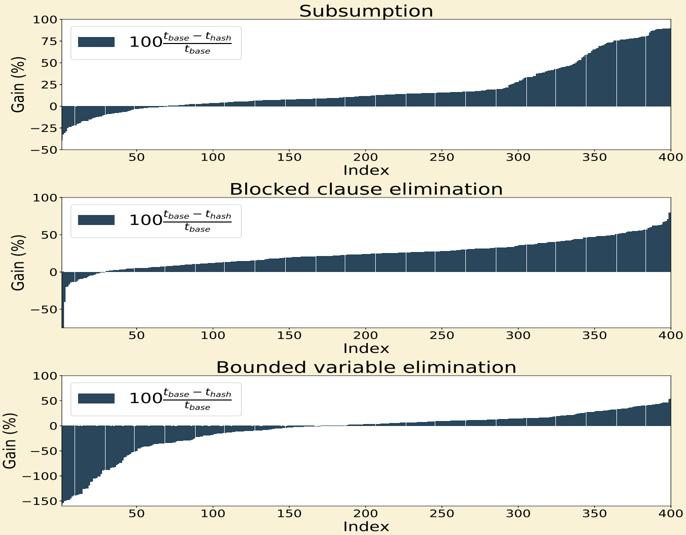
Experimental Results

Experimental Results

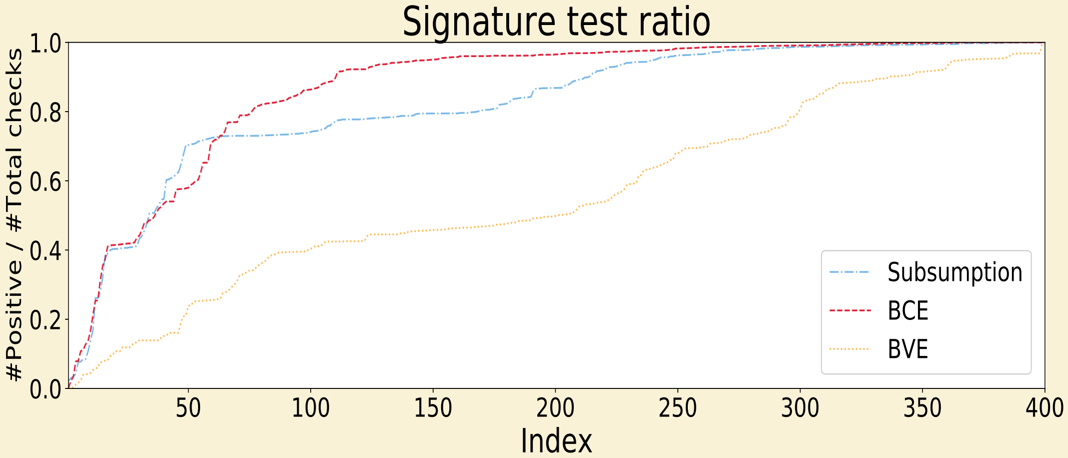
- Implementations of Subsumption, Blocked Clause Elimination (BCE) and Bounded Variable Elimination (BVE) as preprocessing techniques utilizing Propositions 1-4

- Report gain in processing time $(t_{base} - t_{hash})/t_{base}$, where t_{hash} and t_{base} are the processing times (per instance) with signature-checks enabled / disabled respectively.

Processing time



Fraction of Signature Checks



Conclusion

- Signature-based checking useful for subsumption / BCE
- Probably counter-productive for BVE
- Other areas of application in SAT

Thank you!

References

Bayardo, R. J. and Panda, B. (2011).
Fast algorithms for finding extremal sets.
In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 25–34. SIAM.

Eén, N. and Biere, A. (2005).
Effective preprocessing in sat through variable and clause elimination.
In *International conference on theory and applications of satisfiability testing*, pages 61–75. Springer.

Järvisalo, M., Biere, A., and Heule, M. (2010).
Blocked clause elimination.
In *International conference on tools and algorithms for the construction and analysis of systems*, pages 129–144. Springer.