Combining Conflict-Driven Clause Learning and Chronological Backtracking for Propositional Model Counting

Sibylle Möhle and Armin Biere

Institute for Formal Models and Verification
LIT Secure and Correct Systems Lab

5th Global Conference on Artificial Intelligence (GCAI 2019)

17-19 September 2019

Who Wants the Model Count Anyway?

Software Verification

Planning

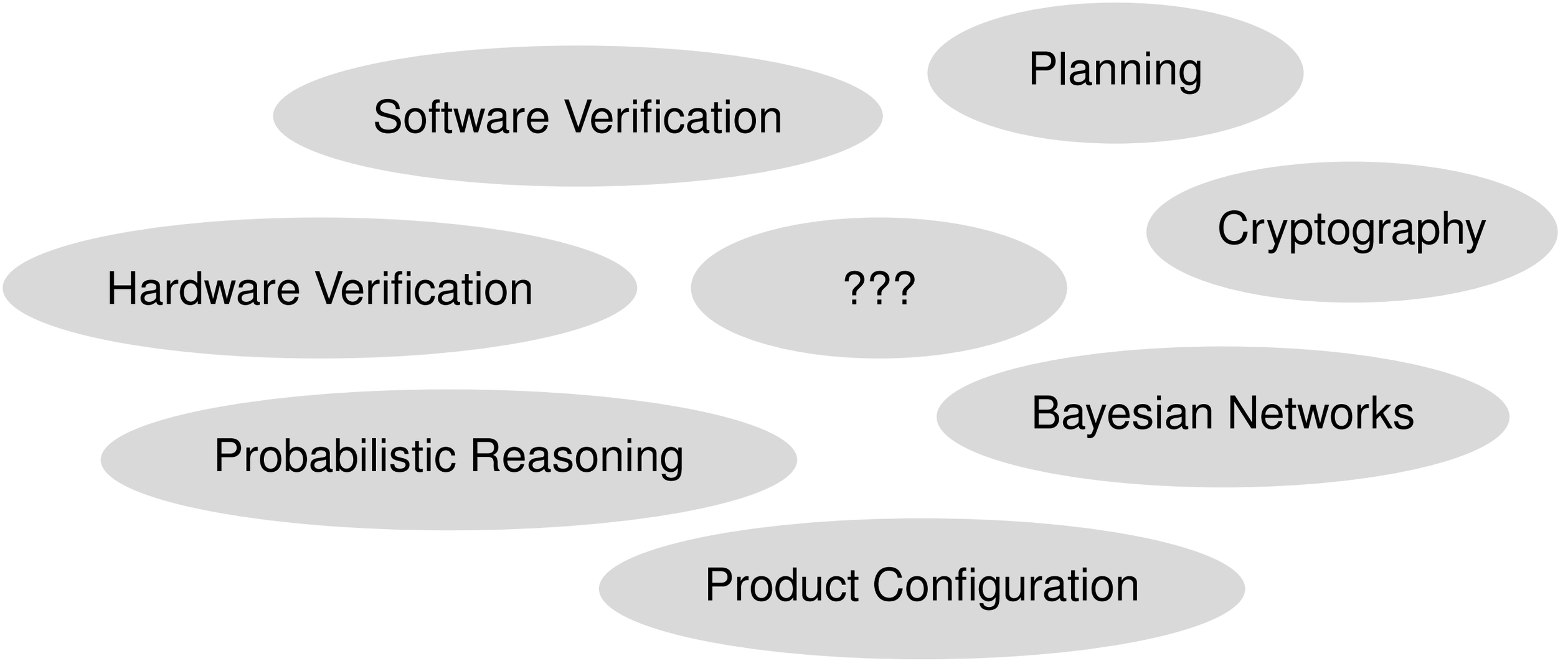
Cryptography

Hardware Verification

Bayesian Networks

Probabilistic Reasoning

Product Configuration


Planning
Software Verification
Cryptography
Hardware Verification
???
Bayesian Networks
Probabilistic Reasoning
Product Configuration

State of the Art in Exact Propositional Model Counting (#SAT)

Counting Based on the Davis-Putnam (DP) Algorithm [1]

- Explore the search space in an ordered manner

Component-Based Reasoning [2,3]

- Decompose formula into subformulae with distinct sets of variables, solve them independently and multiply their model counts
- Parallel and distributed version available [4,5]

[1] E. Birnbaum, E.L. Lozinskii, “The Good Old Davis-Putnam Procedure Helps Counting Models”, JAIR, 1999.
[2] R.J. Bayardo, J.D. Pehoushek, “Counting Models Using Connected Components”, AAAI’00.
[3] M. Thurley, “sharpSAT – Counting Models with Advanced Component Caching and Implicit BCP”, SAT’06.
[4] J. Burchard, T. Schubert, B. Becker, “Laissez-Faire Caching for Parallel #SAT Solving”, SAT’15.
[5] J. Burchard, T. Schubert, B. Becker, “Distributed Parallel #SAT Solving”, CLUSTER’16.

Related Work

Dual Reasoning [6,7]

- Run one SAT solver on the formula and its negation simultaneously
- If the negation of a formula evaluates to true under a variable assignment, the assignment is a model of the formula and vice versa

Chronological Conflict-Driven Clause Learning (CDCL) [8,9]

- Combine chronological backtracking with CDCL
- Fix of several invariants violated by chronological backtracking in combination with CDCL

[6] A. Biere, S. Hölldobler, S. Möhle, “An Abstract Dual Propositional Model Counter”, YSIP’17.
[7] S. Möhle, A. Biere, “Dualizing Projected Model Counting”, ICTAI’18.
[8] A. Nadel, V. Ryvchin, “Chronological Backtracking”, SAT’18.
[9] S. Möhle, A. Biere, “Backing Backtracking”, SAT’19.

Challenges in Exact Propositional Model Counting (#SAT)

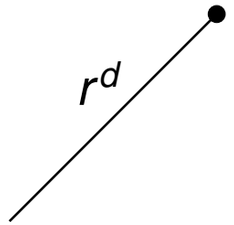
The Search Space Needs to be Explored Exhaustively

$F \quad = (\overline{p} \vee q) \wedge (p \vee q) \qquad M = 0$

- $V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

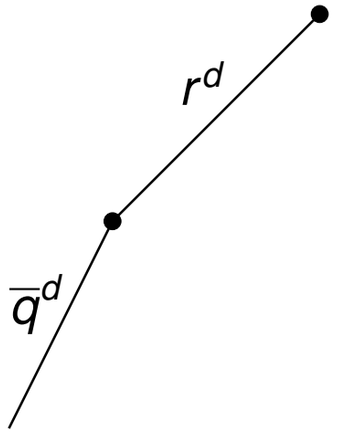
$$F = (\overline{p} \vee q) \wedge (p \vee q) \qquad M = 0$$
$$F|_r = (\overline{p} \vee q) \wedge (p \vee q) \qquad M = 0$$



$$V = \{p, q, r\}$$

The Search Space Needs to be Explored Exhaustively

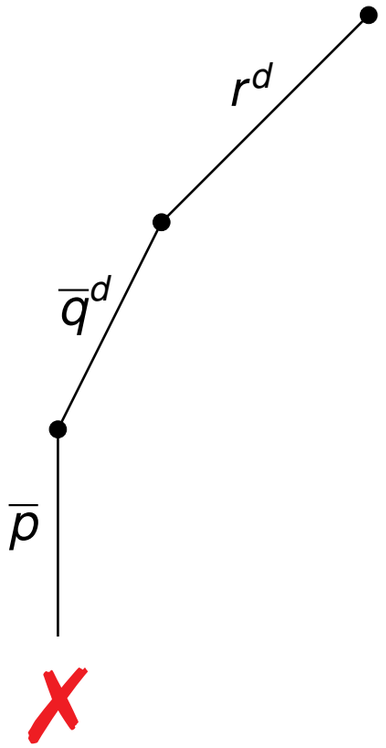
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) & M &= 0
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

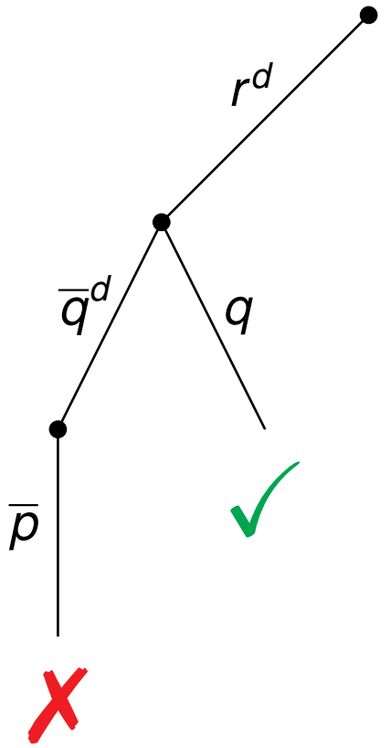
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) & M &= 0 \\
F|_{r\overline{q}\overline{p}} &= \bot & M &= 0
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

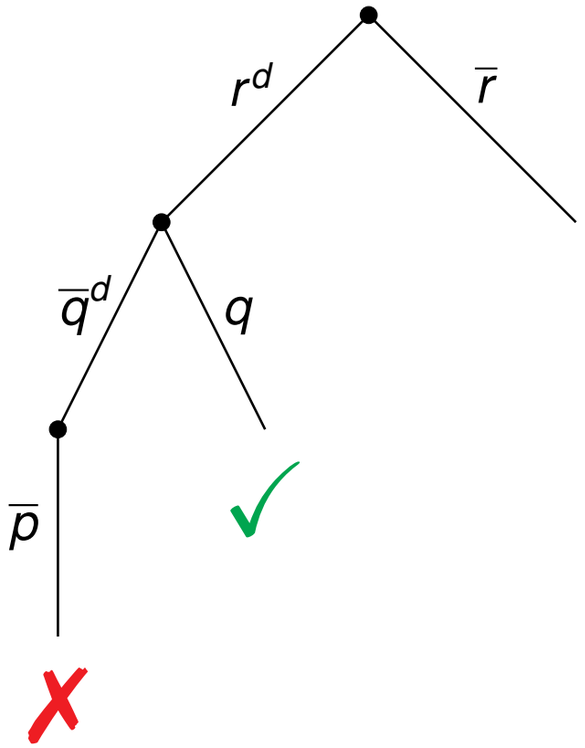
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) & M &= 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) & M &= 0 \\
F|_{r\overline{q}\overline{p}} &= \bot & M &= 0 \\
F|_{rq} &= \top & M &= 2
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

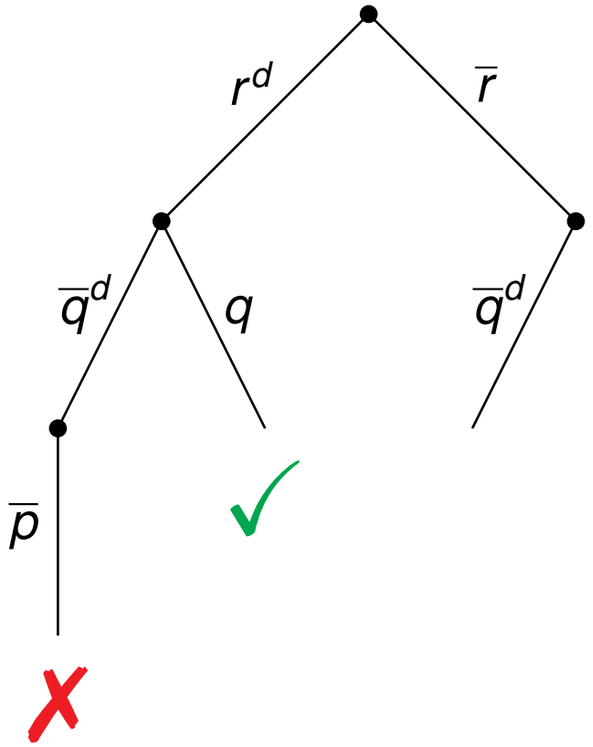
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) && M = 0 \\
F|_{r\overline{q}\overline{p}} &= \bot && M = 0 \\
F|_{rq} &= \top && M = 2 \\
F|_{\overline{r}} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 2
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

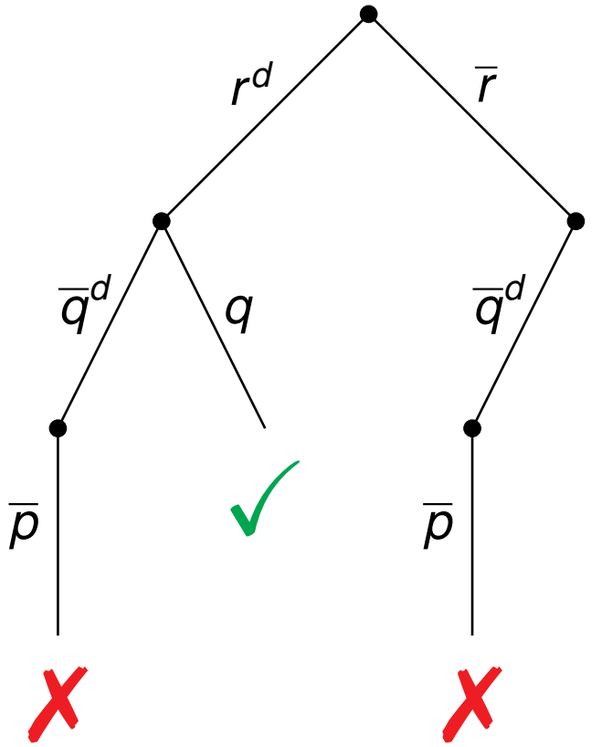
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) && M = 0 \\
F|_{r\overline{q}\overline{p}} &= \bot && M = 0 \\
F|_{rq} &= \top && M = 2 \\
F|_{\overline{r}} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 2 \\
F|_{\overline{r}\overline{q}} &= (\overline{p}) \wedge (p) && M = 2
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

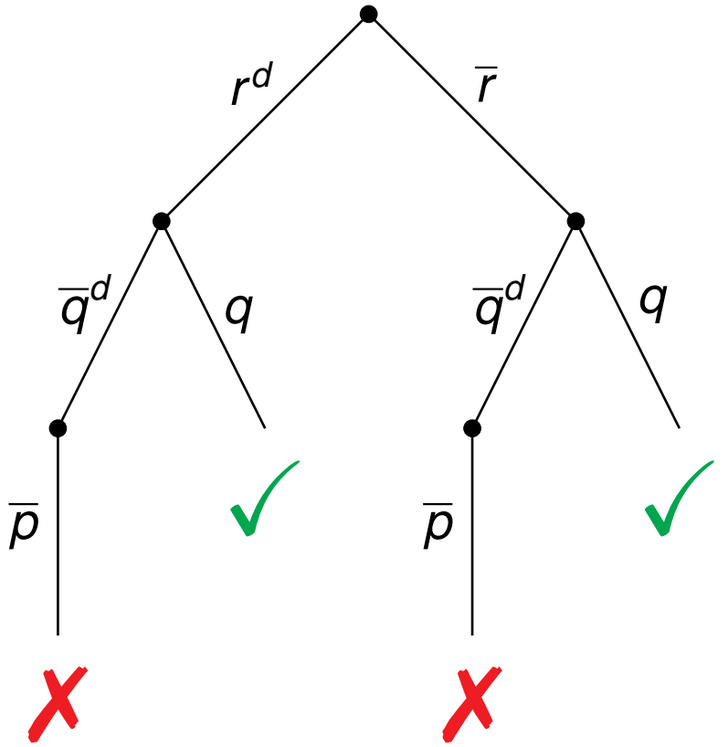
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) && M = 0 \\
F|_{r\overline{q}\overline{p}} &= \bot && M = 0 \\
F|_{rq} &= \top && M = 2 \\
F|_{\overline{r}} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 2 \\
F|_{\overline{r}q} &= (\overline{p}) \wedge (p) && M = 2 \\
F|_{\overline{r}q\overline{p}} &= \bot && M = 2
\end{aligned}
$$



$V = \{p, q, r\}$

The Search Space Needs to be Explored Exhaustively

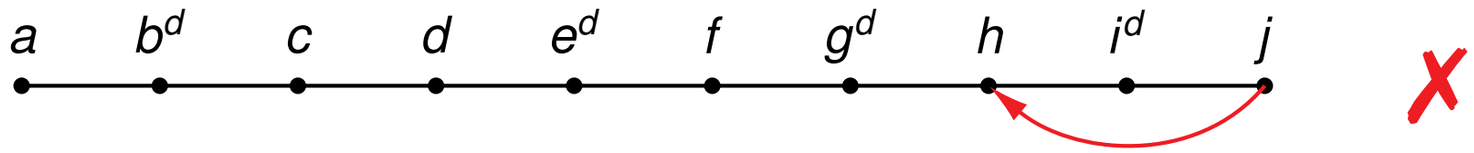
$$
\begin{aligned}
F &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 0 \\
F|_{r\overline{q}} &= (\overline{p}) \wedge (p) && M = 0 \\
F|_{r\overline{q}\overline{p}} &= \bot && M = 0 \\
F|_{rq} &= \top && M = 2 \\
F|_{\overline{r}} &= (\overline{p} \vee q) \wedge (p \vee q) && M = 2 \\
F|_{\overline{r}\overline{q}} &= (\overline{p}) \wedge (p) && M = 2 \\
F|_{\overline{r}\overline{q}p} &= \bot && M = 2 \\
F|_{\overline{r}q} &= \top && M = 4
\end{aligned}
$$



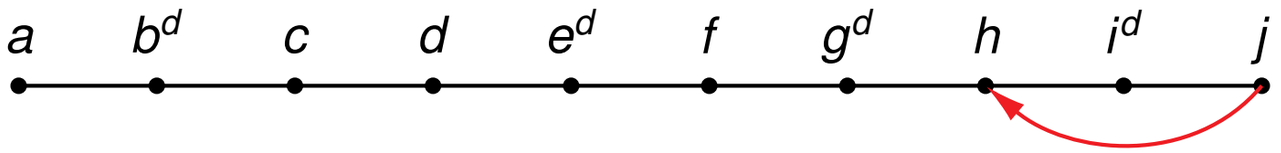
$V = \{p, q, r\}$

Chronological Backtracking Without Conflict-Driven Clause Learning (CDCL)

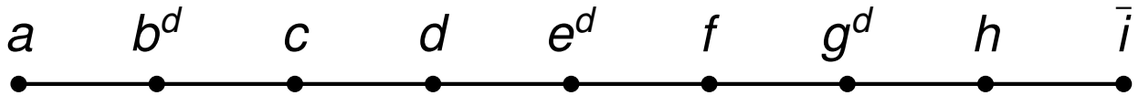
Chronological Backtracking Without Conflict-Driven Clause Learning (CDCL)



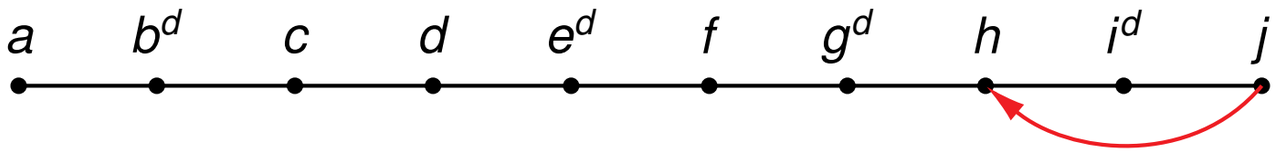
Chronological Backtracking Without Conflict-Driven Clause Learning (CDCL)



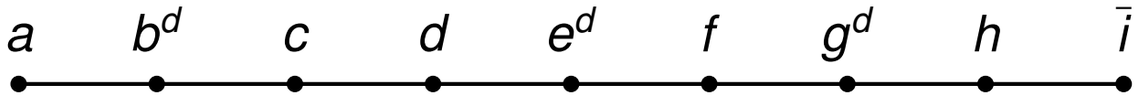
✗



Chronological Backtracking Without Conflict-Driven Clause Learning (CDCL)



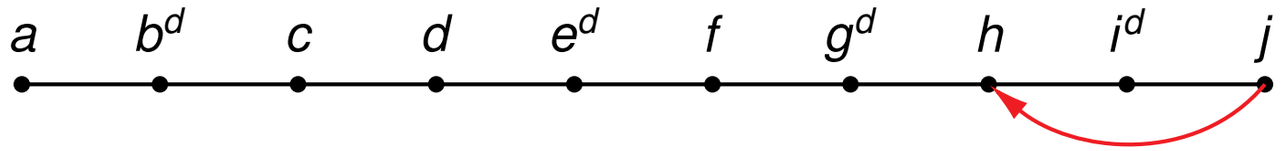
✗



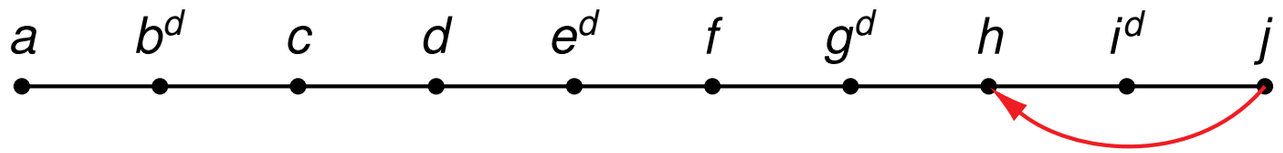
Suitability for #SAT

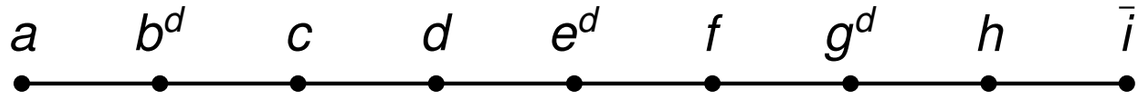
- \+ Search space is traversed in an ordered manner
- \+ The correct model count is returned
- – Regions of the search space with no solution can not be escaped easily
- – Inefficient in terms of execution time

Non-Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)

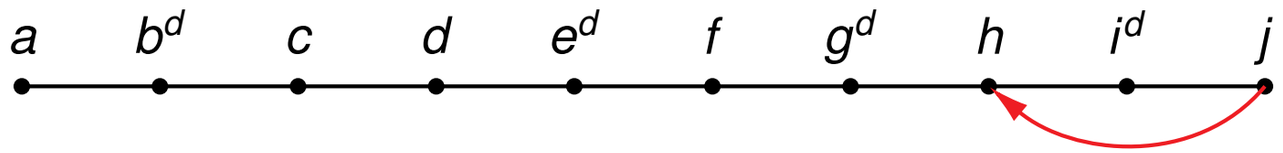


Non-Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)

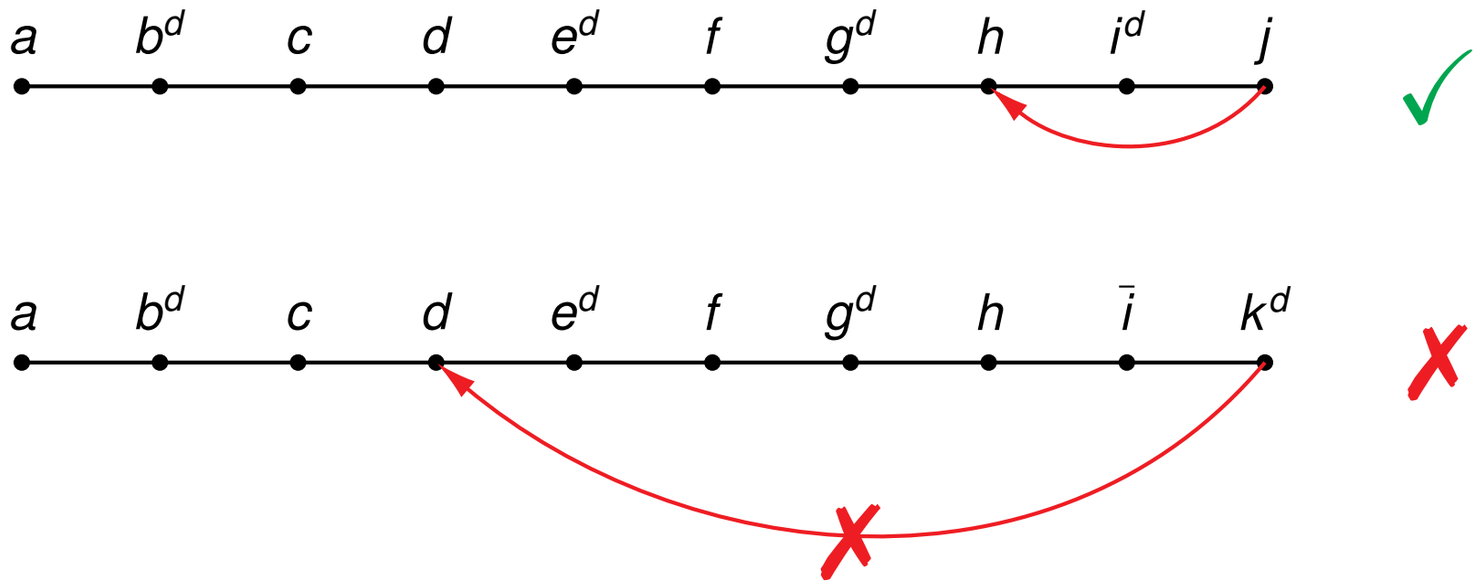




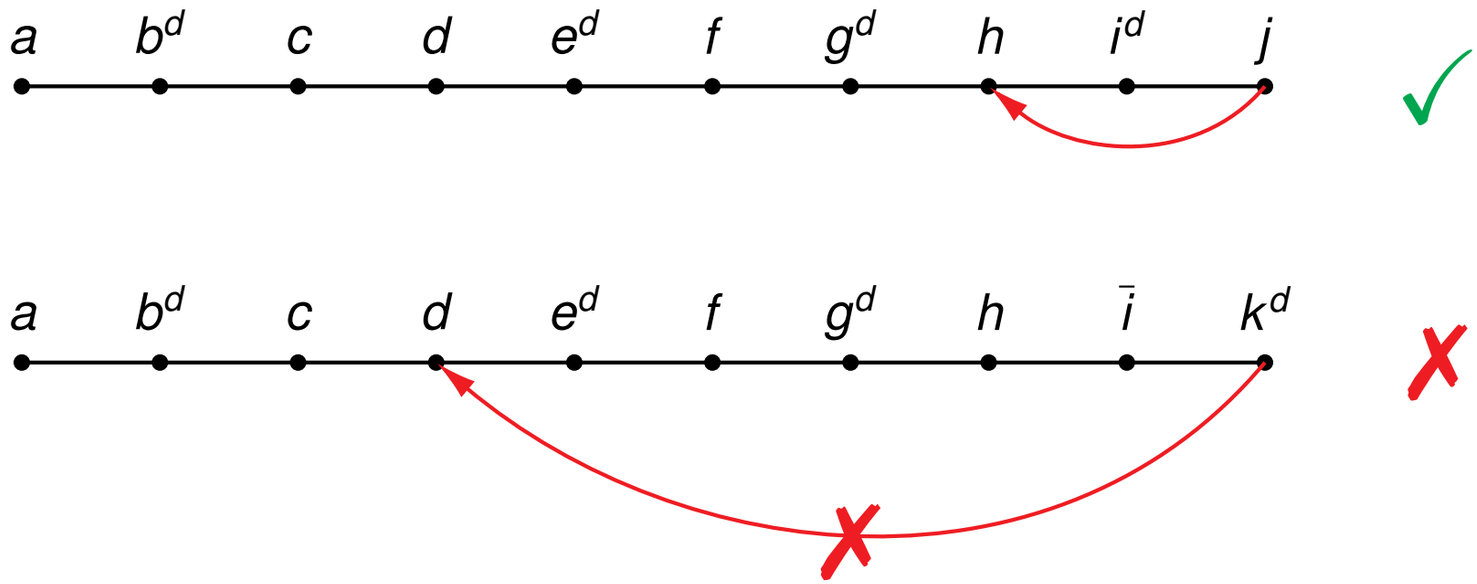
Non-Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)



Non-Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)



Non-Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)



Suitability for #SAT

- \+ Enables the solver to escape regions of the search space with no solution
- \+ Gain in performance (for SAT)
- – Might result in an wrong model count
- – Might lead to redundant work

Chronological Backtracking with Conflict-Driven Clause Learning (CDCL)

Suitability for #SAT

- + Enables the solver to escape regions of the search space with no solution
- + Returns the correct model count
- + Avoids (at least some) redundant work
- + Does not significantly degrade solver performance for SAT

Counting via Enumeration with Chronological CDCL

The Main Idea

$F = (\overline{p} \vee q) \wedge (p \vee q)$

$V = \{p, q, r\}$

$M = (r \wedge q) \vee (\overline{r} \wedge p \wedge q) \vee (\overline{r} \wedge \overline{p} \wedge q) = C_1 \vee C_2 \vee C_3$

$M \equiv F$ and $\#M = \sum_{i=1}^{3} 2^{|V - C_i|} = 4 = \#F$

Generalizing, $\#F = \sum_{C \in M} 2^{|V - C|}$ and

M is a Disjoint-Sum-of-Products (DSOP) representation of F

- M is a disjunction of conjunctions of literals (cubes)
- The cubes in M are pairwise contradicting
- M is logically equivalent to F
- M is not unique

The Main Idea

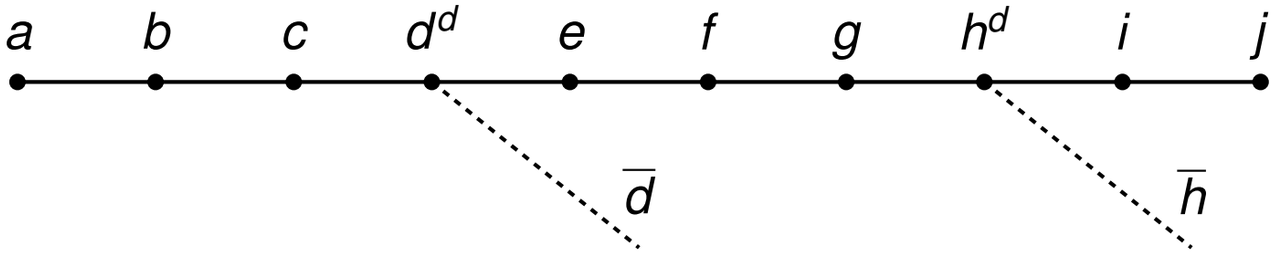
Assignment Trail I

$$I = abcd^d efgh^d ij$$

Pending Search Space $O(I)$

$$O(I) = abc\overline{d} \vee abcdefg\overline{h} \vee I$$

$O(I)$ is a DSOP



Pending Models of F $F \wedge O(I)$

Models of F found M

The Main Idea

During execution, we have that

$$O(I) \wedge F \vee M \equiv F \quad \text{and} \quad \#F = \#(F \wedge O(I)) + \sum_{C \in M} 2^{|V-C|}$$

Upon termination, we have $O(I) = \bot$, hence

$$M \equiv F \quad \text{and} \quad \#F = \sum_{C \in M} 2^{|V-C|}$$

Example

$F = (\overline{p} \vee q) \wedge (p \vee q)$ $\quad\quad$ $V = \{p, q, r\}$

Step	Rule	I	$F\|_I$	M
0		ε	$(\overline{p} \vee q) \wedge (p \vee q)$	$\bot$
1	Decide	r^d	$(\overline{p} \vee q) \wedge (p \vee q)$	$\bot$
2	Decide	$r^d q^d$	$\top$	$\bot$
3	BackTrue	$r^d \overline{q}$	$(\overline{p}) \wedge (p)$	rq
4	Unit	$r^d \overline{q} \overline{p}$	$\bot$	rq
5	BackFalse	$\overline{r}$	$(\overline{p} \vee q) \wedge (p \vee q)$	rq
6	Decide	$\overline{r} p^d$	(q)	rq
7	Unit	$\overline{r} p^d q$	$\top$	rq
8	BackTrue	$\overline{r} \overline{p}$	(q)	$rq \vee \overline{r} pq$
9	Unit	$\overline{r} \overline{p} q$	$\top$	$rq \vee \overline{r} pq$
10	EndTrue			$rq \vee \overline{r} pq \vee \overline{r} \overline{p} q$

Calculus

EndTrue: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{EndTrue}} M \vee I$ if $F|_I = \top$ and $\text{decs}(I) = \emptyset$

EndFalse: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{EndFalse}} M$ if exists $C \in F$ and $C|_I = \bot$ and $\delta(C) = 0$

Unit: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{Unit}} (F,\ I\ell,\ M,\ \delta[\ell \mapsto a])$ if $F|_I \neq \top$ and $\bot \notin F|_I$ and exists $C \in F$ with $\{\ell\} = C|_I$ and $a = \delta(C \setminus \{\ell\})$

BackTrue: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{BackTrue}} (F,\ PK\ell,\ M \vee I,\ \delta[L \mapsto \infty][\ell \mapsto e])$ if $F|_I = \top$ and $PQ = I$ and $D = \overline{\text{decs}(I)}$ and $e + 1 = \delta(D) = \delta(I)$ and $\ell \in D$ and $e = \delta(D \setminus \{\ell\}) = \delta(P)$ and $K = Q_{\leqslant e}$ and $L = Q_{>e}$

BackFalse: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{BackFalse}} (F,\ PK\ell,\ M,\ \delta[L \mapsto \infty][\ell \mapsto j])$ if exists $C \in F$ and exists D with $PQ = I$ and $C|_I = \bot$ and $c = \delta(C) = \delta(D) > 0$ such that $\ell \in D$ and $\bar{\ell} \in \text{decs}(I)$ and $\ell|_Q = \bot$ and $F \wedge \overline{M} \models D$ and $j = \delta(D \setminus \{\ell\})$ and $b = \delta(P) = c - 1$ and $K = Q_{\leqslant b}$ and $L = Q_{>b}$

Decide: $(F,\ I,\ M,\ \delta) \rightsquigarrow_{\text{Decide}} (F,\ I\ell^d,\ M,\ \delta[\ell \mapsto d])$ if $F|_I \neq \top$ and $\bot \notin F|_I$ and $\text{units}(F|_I) = \emptyset$ and $V(\ell) \in V$ and $\delta(\ell) = \infty$ and $d = \delta(I) + 1$

Conclusion

Our Contribution

- Combined chronological backtracking with CDCL for propositional model counting
- Formal calculus for propositional model counting based on these ideas
  - enumeration approach
  - no blocking clauses
  - escape search space regions with no solution
- Formal proof of correctness

Further Research

- Implement our rules to experimentally validate their effectiveness
- Investigate possible applications in SMT and QBF
- Extend our approach to projected model counting in combination with dual reasoning
- Target component-based reasoning