Exploiting Isomorphic Subgraphs in SAT

OFER STRICHMAN, TECHNION, ISRAEL

ALEXANDER IVRII, IBM-HAIFA, ISRAEL

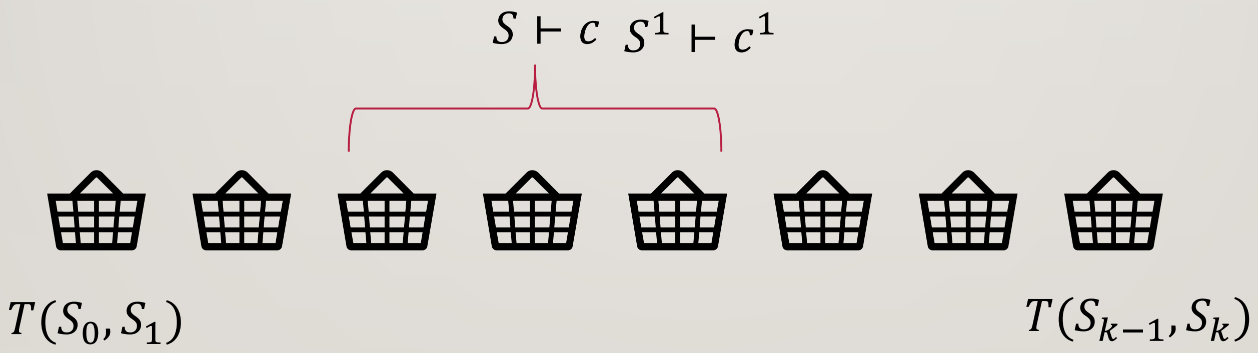
CLAUSE REPLICATION

- Recall clause replication [S'01,S'04] for BMC.
  - Enhances learning with extra clauses, not related to the current path.
  - Based on the regular structure of BMC formulas.

- Our contributions here:
  - Identify this property in many other problem domains
  - Theory: Show how it extends dynamic symmetry handling [DBB'17,TD'19]
  - Practice: Experiments with various learning / forgetting strategies.

CLAUSE REPLICATION IN BMC [S'01,S'04]

- BMC (of safety properties): $\mathrm{BMC_k} \equiv I(S_0) \wedge \bigwedge_{i=0..k-1} T(S_i, S_{i+1}) \wedge \neg P(S_k)$
- At the CNF level $T(S_i, S_{i+1})$ is the same for each i, up to renaming.
- This can be exploited for learning additional clauses:



Gliding symmetry: 

Clause replication in BMC [S'01,S'04]

Maintain additional clause header data:

1. Is this clause 'replicable' ?
2. (min,max) cycle used for deriving this clause:

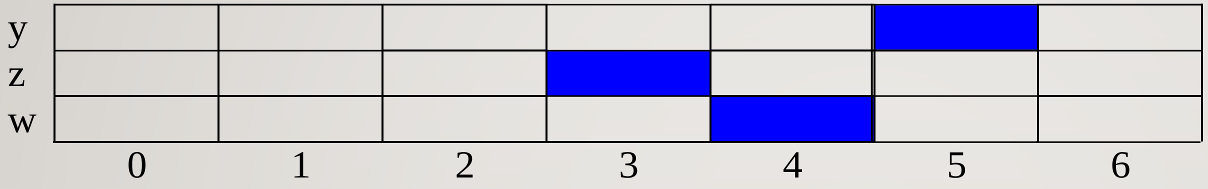
When learning a new clause c from antecedents S (i.e., $S \vdash c$):

If all of S clauses are marked:

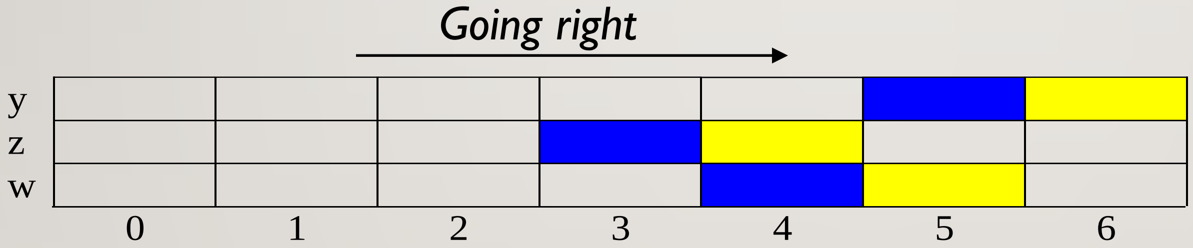
1. Mark c.
2. Record *(min,max)* cycle indices in S.
3. Learn c^i for $i \in -min..(k - max)$.

EXAMPLE

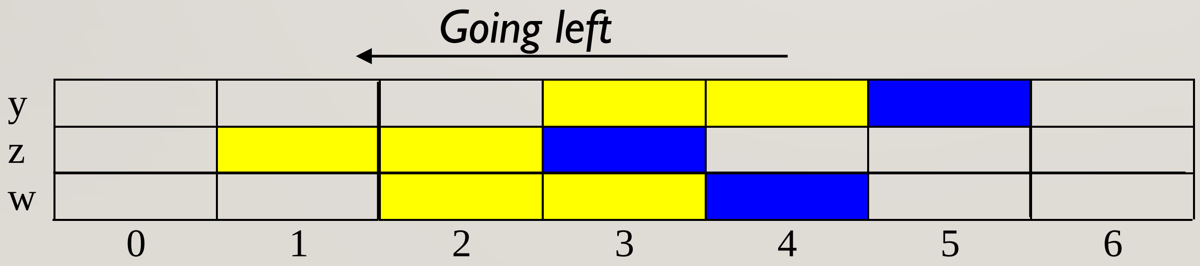
$s = (-x_2\ y_5), (x_2\ y_5\ z_3\ w_4)$ $(min, max) = (2,5),\ k = 6$



$c = (y_5\ z_3\ w_4)$



$c^1 = (y_6\ z_4\ w_5)$



$c^{-1} = (y_4\ z_2\ w_3)$

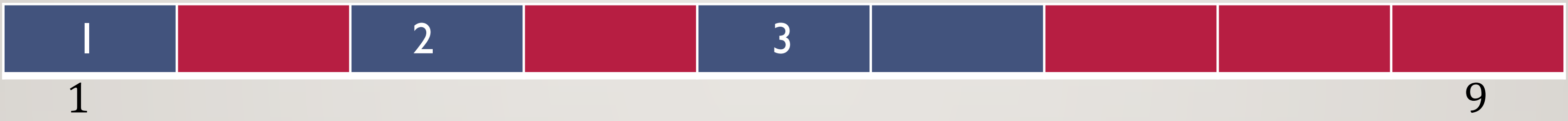
$c^{-2} = (y_3\ z_1\ w_2)$

THE SAME PRINCIPLE APPLIES TO MANY OTHER FORMULAS

- In previous works:
  - Bounded model checking [S'01,S'04]
  - Planning with neural networks [SDNS'20]
- Here we apply it to various problems that drew attention in recent years:
  - Van-Der Waerden numbers
  - Pythagorean triples
  - The 'Sweep' problem
  - The 'Anti-bandwidth' problem
  - Cardinality constraints

EXAMPLE 1: VAN-DER WAERDEN NUMBERS

- The van der Waerden number $W(k)$ is the smallest integer n such that every 2-coloring of $1..n$ has a monochromatic arithmetic progression of length k.
- E.g., a bad coloring for $n = 9, k = 3$



- It can be shown that $W(3) = 9$.
- We have a witness for $W(3) > 8$

EXAMPLE 1: VAN-DER WAERDEN NUMBERS

- For a sequence length n, define n variables
  - x_i- for $1 \leq i \leq n$, location i is with color '1'.
- Suppose $k = 3, n = 10$. Then:
  - No 3 consecutive literals with gap 1 are all '0': $(1\ 2\ 3)\ (2\ 3\ 4)\ (3\ 4\ 5)\ \ldots\ (8\ 9\ 10)$
  - No 3 consecutive literals with gap 2 are all '0': $(1\ 3\ 5)\ (2\ 4\ 6)\ (3\ 5\ 7)\ \ldots\ (6\ 8\ 10)$
  - No 3 consecutive literals with gap 3 are all '0': $(1\ 4\ 7)\ (2\ 5\ 8)\ (3\ 6\ 9)\ (4\ 7\ 10)$
  - No 3 consecutive literals with gap 4 are all '0': $(1\ 5\ 9)\ (2\ 6\ 10)$
- + same for all color '1': negate all literals, e.g.,

$$(-1, -2, -3)\ (-2, -3, -4)\ (-3, -4, -5)\ \ldots\ (-8, -9, -10)$$

EXAMPLE 2: PYTHAGOREAN TRIPLES

- Triples (a, b, c) such that $a^2 + b^2 = c^2$
- Q: for a given $n \in \mathrm{N}$, can $1..N$ be separated to two sets, such that no set contains a Pythagorean triple ?
- Example CNF for $n = 17$

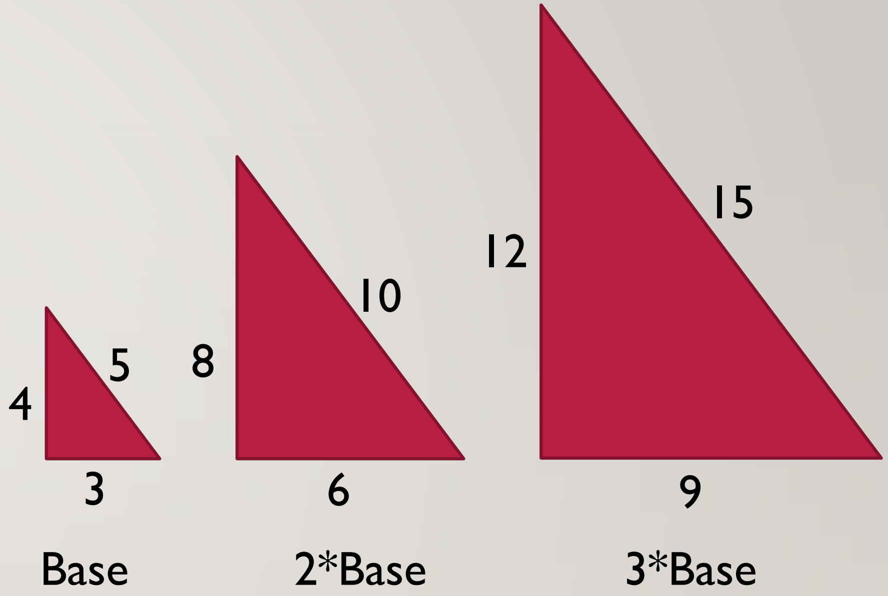
(3 4 5)	(-3 -4 -5)
(5 12 13)	(-5 -12 -13)
(6 8 10)	(-6 -8 -10)
(9 12 15)	(-9 -12 -15)
(8 15 17)	(-8 -15 -17)

EXAMPLE 2: PYTHAGOREAN TRIPLES

- Example CNF for $n = 17$

3 4 5	-3 -4 -5
5 12 13	-5 -12 -13
6 8 10	-6 -8 -10
9 12 15	-9 -12 -15
8 15 17	-8 -15 -17

- Symmetry emanates from factoring triples



- Going left: divide c by a common divisor of the antecedents
- Going right: multiply c by a factor f, as long as $f \cdot \max \leq n$ (max = maximal literal in c's antecedents).

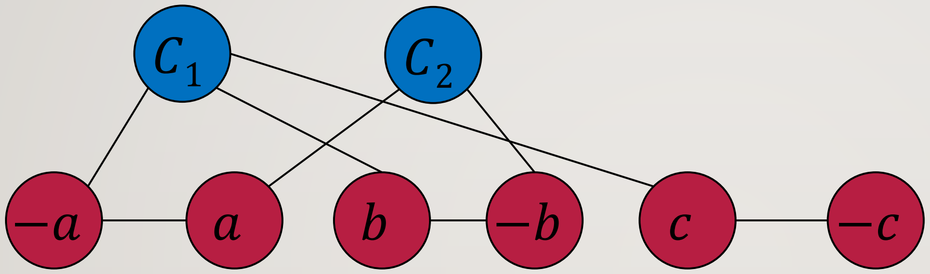
A THEORETICAL FRAMEWORK

- Static (full) symmetry –
  - Static symmetry breaking [Shatter, BreakID] – Statically adding Symmetry-Breaking constraints.
  - Dynamic symmetry handling [e.g., SEL (DBB'17)] – Given a learned clause c, adding extra clauses ("eclauses") based on symmetry data.

- This talk – we find eclauses regardless of static symmetry.
  - The theory is based on isomorphic subgraphs

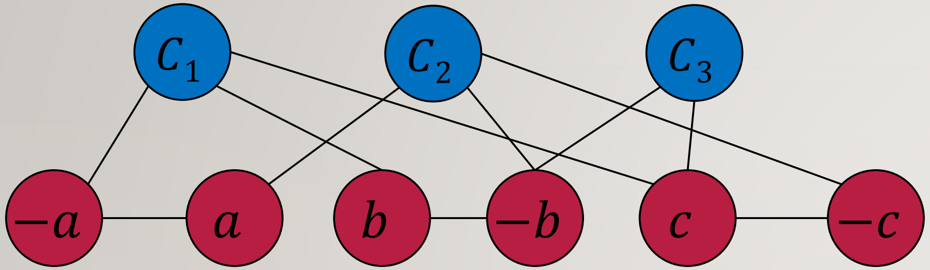
THE CNF INCIDENCE GRAPH

- The colored literals incidence graph of $\overset{C_1}{(-a, b, c)}\ \overset{C_2}{(a, -b)}$:



- Opposite literals are connected
- A clause node is connected to its literals
- Literals have one color, clauses another.

STATIC SYMMETRY IN CNF, BY EXAMPLE

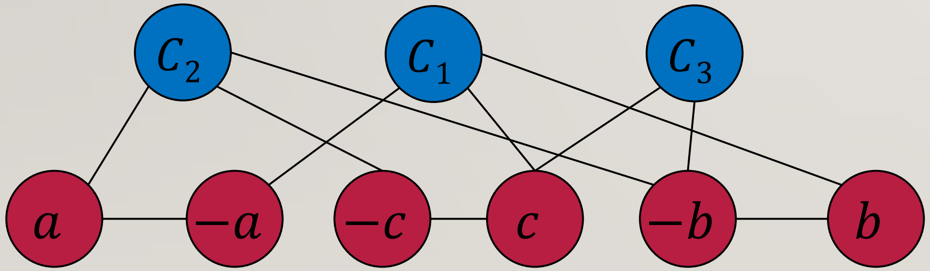


φ:
C_1: $(-a, b, c)$
C_2: $(a, -b, -c)$
C_3: $(-b, c)$

Syntactic equivalence up to clause/literals reordering

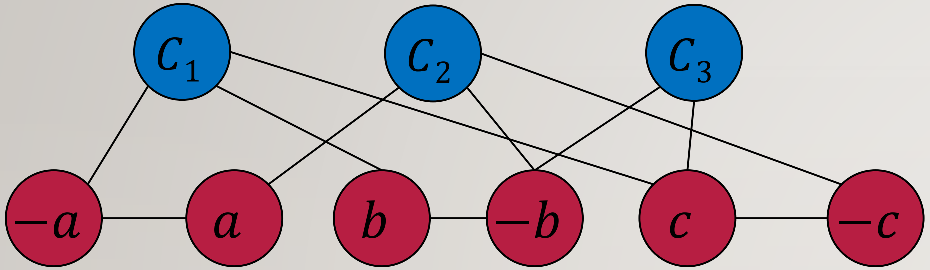
Find a Boolean-consistent map σ between the labels, such that $\sigma(\varphi) \equiv \varphi$.

Example: σ: $(a, -a)(b, -c)(C_1, C_2)$



$\sigma(\varphi)$:
$(a, -c, -b)$
$(-a, c, b)$
$(c, -b)$

STATIC SYMMETRY IN CNF, BY EXAMPLE

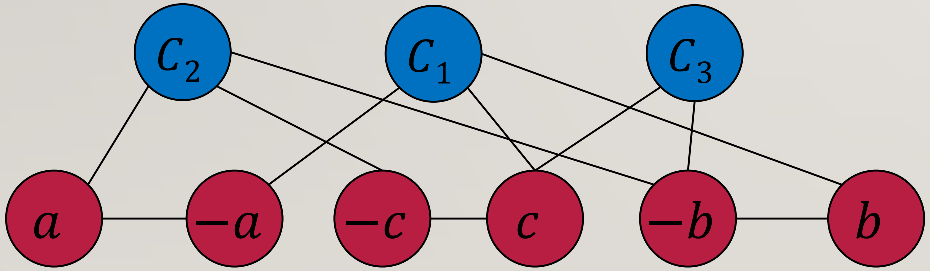


α: 1 0 0 1 0 1

φ:
C_1: $(-a, b, c)$
C_2: $(a, -b, -c)$
C_3: $(-b, c)$

Find a Boolean-consistent map σ between the labels, such that $\sigma(\varphi) \equiv \varphi$.

Example: σ: $(a, -a)(b, -c)(C_1, C_2)$



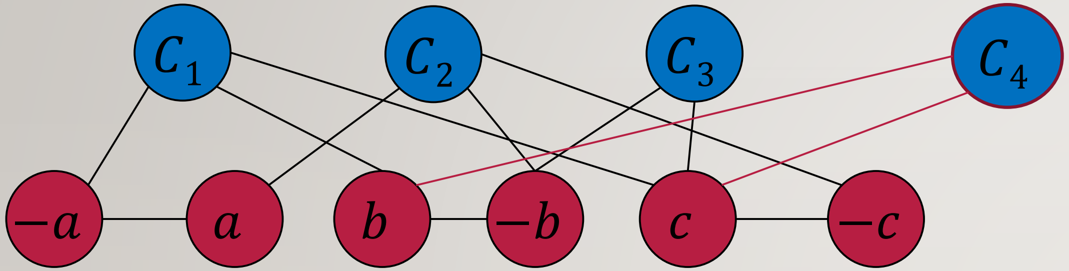
$\sigma(\alpha)$: 1 0 0 1 0 1

Hence, if $\alpha \models \varphi$ then $\sigma(\alpha) \models \varphi$

STATIC SYMMETRY BREAKING

- So σ in our example has the property that $\forall \alpha. \alpha \vDash \varphi \Rightarrow \sigma(\alpha) \vDash \varphi$.
- We only need one representative to maintain satisfiability.
- Shatter/BreakID find such mappings, and add symmetry-breaking constraints.
    - How ? See Crawford et al [CGLR96].

DYNAMIC SYMMETRY HANDLING*



Suppose we learned a new clause C_4.

Hence $\varphi \vdash_{res} C_4$

$\Rightarrow \ \sigma(\varphi) \vdash_{res} \sigma(C_4)$

$\Rightarrow \quad \varphi \vdash_{res} \sigma(C_4)$

Conclusion: we can learn also $\sigma(C_4)$

Note that:

1. This does not break the symmetry; all solutions remain.
2. The map σ was built statically, according to symmetries in φ.

* Used by SEL [BDB'17], SLS [BNOS'10], SP [BBDDM'12]

“ALMOST SYMMETRIES” [CBMS14,…]

- Suppose we have $\varphi \equiv \varphi_1 \cup \varphi_2$
  - φ_1 prevents symmetry.
  - But there is still a mapping σ such that $\sigma(\varphi_2) \equiv \varphi_2$

- If we learn a clause c from φ_2, we can also add the eclause $\sigma(c)$.

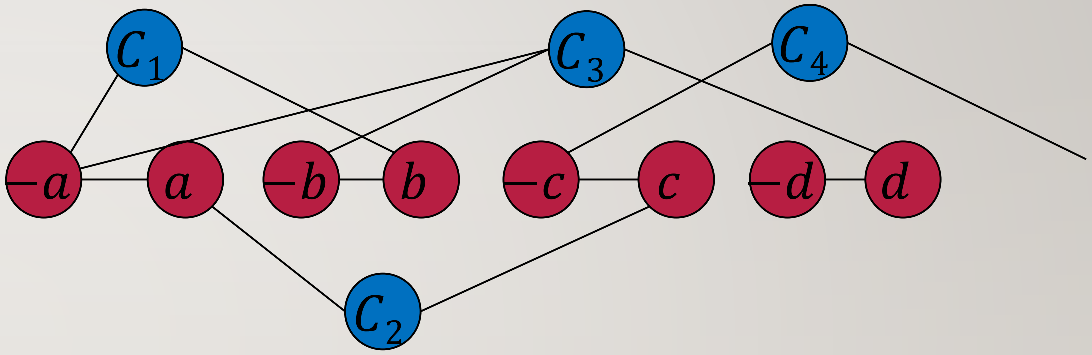
Our work: a weaker condition for eclauses.

	Symmetry	**Almost Symmetry**
Usage	Static, dynamic	Dynamic
Formula	φ	$\varphi_1 \cup \varphi_2$
Requires	$\sigma(\varphi) \equiv \varphi$	$\sigma(\varphi_2) \equiv \varphi_2$

THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Consider the resolution process (root clauses in green):

$$\frac{\dfrac{(-a, b) \qquad (a, c)}{(b, c)} \qquad (a, -b, d)}{(a, c, d)}$$



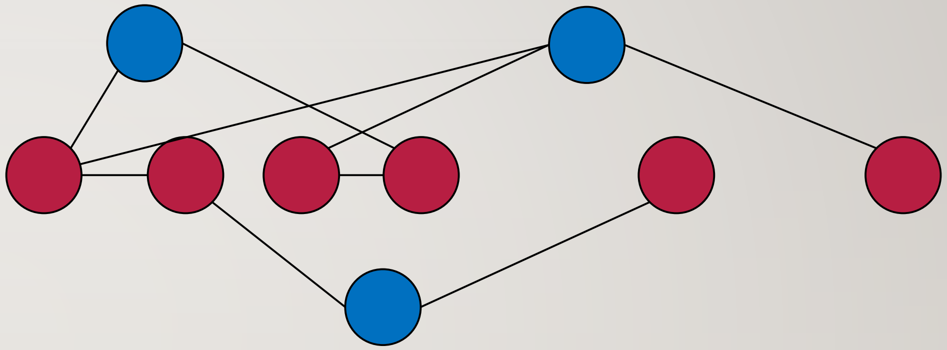
The subgraph induced by the resolution process is a union of

- The subgraphs corresponding to the root clauses
- The edges of the resolved variables

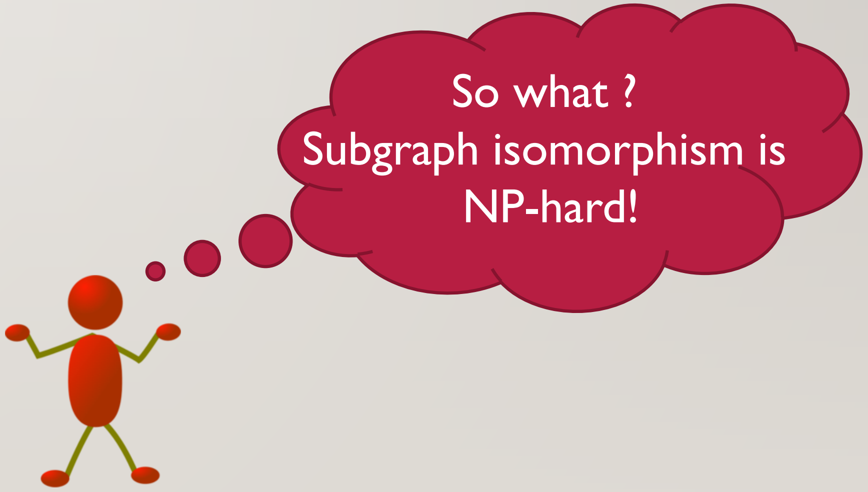
THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Consider the resolution process (root clauses in green):

$$\frac{\frac{(-a, b) \qquad (a, c)}{(b, c)} \qquad (a, -b, d)}{(a, c, d)}$$



Any subgraph isomorphic to this one, corresponds to a legal resolution.



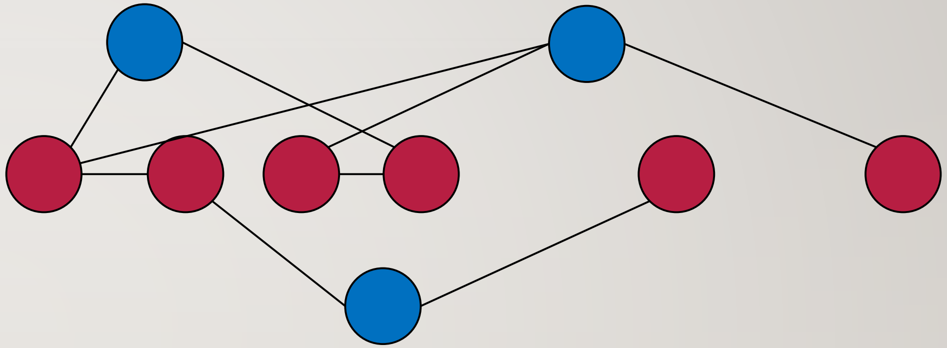
THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Consider the resolution process (root clauses in green):

$(-a, b) \qquad (a, c)$

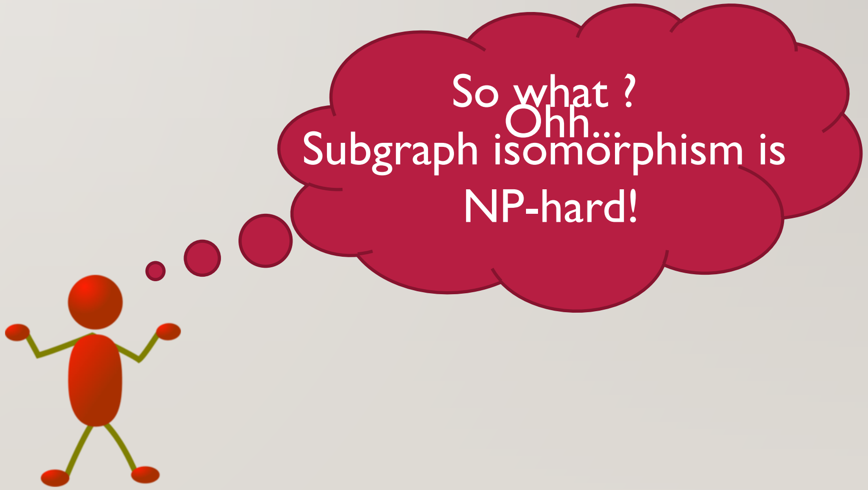
$(b, c) \qquad (a, -b, d)$

(a, c, d)



Isomorphic subgraphs ⇔ isomorphic subformulas.

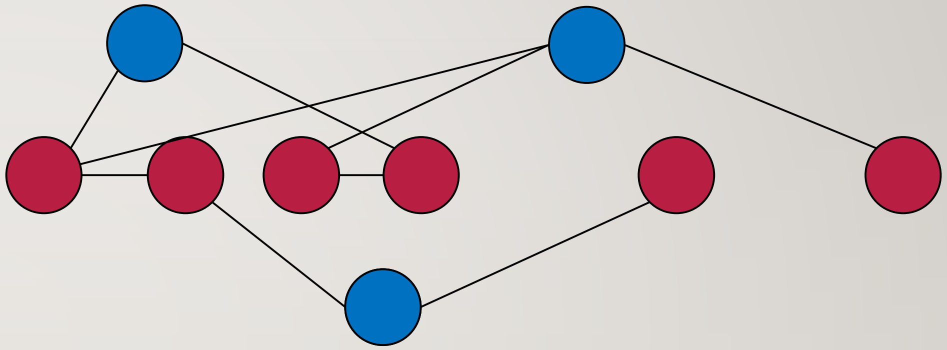
Q: How can this fact be used?
A: look for formulas in which this property can be detected statically



THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Consider the resolution process (root clauses in green):

$$\dfrac{\dfrac{(-a,b) \qquad (a,c)}{(b,c)} \qquad (a,-b,d)}{(a,c,d)}$$



Isomorphic subgraphs ⇔ isomorphic subformulas.

Q: How can this fact be used?
A: look for formulas in which this property can be detected statically

THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Q: How can this fact be used?
A: look for formulas in which this property can be detected statically

- Let $\varphi \equiv \varphi_1 \cup \varphi_2 \cup \varphi_3$ and let σ be a partial map such that $\sigma(\varphi_2) \equiv \varphi_3$
- Typically $\varphi_2 \cap \varphi_3 \neq \emptyset$.

	Symmetry	Almost Symmetry	This work
Usage	Static, dynamic	Dynamic	Dynamic
Formula	φ	$\varphi_1 \cup \varphi_2$	$\varphi_1 \cup \varphi_2 \cup \varphi_3$
Requires	$\sigma(\varphi) \equiv \varphi$	$\sigma(\varphi_2) \equiv \varphi_2$	$\sigma(\varphi_2) \equiv \varphi_3$

THE SUBGRAPH INDUCED BY A RESOLUTION PROCESS

Q: How can this fact be used?
A: look for formulas in which this property can be detected statically

- Let $\varphi \equiv \varphi_1 \cup \varphi_2 \cup \varphi_3$ and let σ be a partial map such that $\sigma(\varphi_2) \equiv \varphi_3$
- Typically $\varphi_2 \cap \varphi_3 \neq \emptyset$.
- Examples:

Problem	Map type	φ_1	φ_2	φ_3
BMC	$+j$	$I(0), P(k)$	$c \in T(i, i+1)$	$c^j \in T(i+j, i+j+1)$
Van-Der Waerden	$+j$		$c \in \varphi$	$c^j \in \varphi$
Pyth. triples	$*j$		$c \in \varphi$	$c^{*j} \in \varphi$

THE E-CLAUSES: WHAT KIND OF CLAUSES ARE THESE ?

- They are loosely related to the search
  - On the one hand, they refer to a different set of variables than the current focus
  - On the other, they build a clause structure (proof?) which is symmetric to the learned one.
    - In that sense, they are not 'arbitrary' implied clauses.

- Does adding them as additional learned clauses improve performance?

ADDITION / DELETION STRATEGY FOR E-CLAUSES

- Addition:
  - During search / restart: restart
  - Maximal size: 20
  - Maximal (partial*) LBD: 6
    - Measured with respect to the current trail
    - It does not necessarily include a full assignment of the e-clause.
  - Maximal # of non-false literals: 3
- Deletion:
  - Initial score: 0.8x
  - Category (core / Tier-2 / Local): Local
  - Deletion ratio (% of local clauses removed during 'reduceDB'): 66%

RESULTS

Symmetry	Replication	Time (par-2)	Conflicts	e-clauses
	✓	111.2	1,079,719	30568
static		149.8	2,110,472	0
		190.4	2,112,666	0
dynamic	✓	198.5	1,963,104	50618
dynamic		233.2	2,477,840	6,729

30 non-trivial instances, 16 unsat.
Includes eclause filtering.

CONCLUSIONS

- Future work:
  - Better adaptation of solvers to this extra information
  - “Symbolic clauses” – generate eclauses lazily.