Mining-Based Compression Approach of Propositional Formulae

Lakhdar Saïs[1]

20 février 2014

Joint work with Saïd Jabbour[1], Yakoub Salhi[1] and Takeaki Uno[2]

[1] CRIL, CNRS, Université Lille Nord de France, Artois, Lens, France

[2] National Institute of Informatics (NII), Tokyo, Japan

Plan

Propositional logic

- Classical propositional logic :

$$A ::= p \mid \neg A \mid A \wedge A \mid A \vee A \mid A \rightarrow A$$

- De Morgan laws :

$$A \vee B \equiv \neg(\neg A \wedge \neg B) \quad A \wedge B \equiv \neg(\neg A \vee \neg B)$$
$$A \rightarrow B \equiv \neg A \vee B \quad \neg\neg A \equiv A$$

- Boolean interpretation :
  - $[\![\cdot]\!] : \text{Prop} \rightarrow \{0, 1\}$
  - Extension to formulae : $[\![\neg A]\!] = 1 - [\![A]\!]$, $[\![A \wedge B]\!] = min([\![A]\!], [\![B]\!])$

- Satisfiability : $\exists [\![\cdot]\!], [\![A]\!] = 1$ (NP-complete [Cook 71])

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses :

$$\overbrace{(x_1 \vee \cdots \vee x_l)}^{clause} \wedge (y_1 \vee \cdots \vee y_m) \wedge (z_1 \vee \cdots \vee z_n) \cdots$$

- Clause : a disjunction of literals (p, $\neg p$)

- Example :

$$(p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee s) \wedge p \wedge (r \vee \neg s)$$

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses :

$$\overbrace{(x_1 \vee \cdots \vee x_l)}^{clause} \wedge (y_1 \vee \cdots \vee y_m) \wedge (z_1 \vee \cdots \vee z_n) \cdots$$

- Clause : a disjunction of literals (p, $\neg p$)

- Example :

$$\overbrace{(p \vee \neg q \vee \neg r)}^{1} \wedge \overbrace{(p \vee \neg q \vee s)}^{1} \wedge \overbrace{p}^{1} \wedge (r \vee \neg s)$$

$$[\![p]\!] = 1$$

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses :

$$\overbrace{(x_1 \vee \cdots \vee x_l)}^{clause} \wedge (y_1 \vee \cdots \vee y_m) \wedge (z_1 \vee \cdots \vee z_n) \cdots$$

- Clause : a disjunction of literals (p, $\neg p$)

- Example :

$$\overbrace{(p \vee \neg q \vee \neg r)}^{1} \wedge \overbrace{(p \vee \neg q \vee s)}^{1} \wedge \overbrace{p}^{1} \wedge \overbrace{(r \vee \neg s)}^{1}$$

$[\![p]\!] = 1$ et $[\![r]\!] = 1$ (Partial interpretation)

Transformation - Extension principle [G. Tseitin 1965]

- Introduce new variables to represent truth value of sub-formulae
- Example : DNF $\longrightarrow$ CNF

$$(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$$

- Naïve approach : 2^n clauses and $n \times 2^n$ literals
  $(x_1 \vee \cdots \vee x_{n-1} \vee x_n) \wedge (x_1 \vee \cdots \vee x_{n-1} \vee y_n) \wedge \cdots \wedge$
  $(y_1 \vee \cdots \vee y_{n-1} \vee y_n)$
- Tseitin approach : $2 \times n + 1$ clauses and $n + 2 \times 2 \times n$ literals
  $(z_1 \vee \cdots \vee z_n) \wedge (\neg z_1 \vee x_1) \wedge (\neg z_1 \vee y_1) \wedge \cdots \wedge (\neg z_n \vee x_n) \wedge (\neg z_n \vee y_n)$

Extended resolution proof system [G. Tseitin 1965]

- Extended resolution :

$$\frac{l \vee \alpha \in \Sigma \qquad \bar{l} \vee \beta \in \Sigma}{\alpha \vee \beta} \; [Res]$$

Extension : $x \leftrightarrow F$

- Shorten resolution proofs
- Open question : automatization of extended resolution proof systems ?

Modeling in SAT

- Knowledge representation using CNF formulae

		6	1		2	5		
	3	9				1	4	
				4				
9		2		3		4		1
	8						7	
1		3		6		8		9
				1				
	5	4				9	1	
		7	5		3	2		

8	4	6	1	7	2	5	9	3
1	3	9	6	5	8	1	4	2
5	2	1	3	4	9	7	6	8
9	6	2	8	3	7	4	5	1
4	8	5	9	2	1	3	7	6
1	7	3	4	6	5	8	2	9
2	9	8	7	1	4	6	3	5
3	5	4	2	8	6	9	1	7
6	1	7	5	9	3	2	8	4

- Example : $n \times n$ Sudoku
  - Associate to each cell, n propositional variables
  - Each cell contains at least one value :
    $\bigwedge_{l=1}^{n} \bigwedge_{c=1}^{n} (\bigvee_{v=1}^{n} p_{(l,c,v)}) \implies n^2$ clauses of size n
- Leads usually to formulae of huge size

Modeling in SAT : an example from formal verification

Name of the CNF instance : post-cbmc-zfcp-2.8-u2.cnf (BMC)
p cnf **11 483 525** (vars) **32 697 150** (clauses)

1 -3 0
2 -3 0 $\quad x_3 = x_1 \wedge x_2$
1 -2 3 0
. . . 1million pages later
-11482897 -11483041 -11483523 0
11482897 11483041 -11483523 0 $\quad x_3 \leftrightarrow x_4 \leftrightarrow x_5$
11482897 -11483041 11483523 0
-11482897 11483041 11483523 0
-11483518 -11483524 0
-11483519 -11483524 0
-11483520 -11483524 0
-11483521 -11483524 0 $\quad x_6 = (x_7 \wedge x_8 \wedge x_9 \wedge x_{10} \wedge x_{11} \wedge x_{12})$
-11483522 -11483524 0
-11483523 -11483524 0
11483518 11483519 11483520 11483521 11483522 11483523 11483524 0
-8590303 -11483524 -11483525 0
8590303 11483524 -11483525 0 $\quad x_{13} \leftrightarrow x_{14} \leftrightarrow x_{15}$
8590303 -11483524 11483525 0
-8590303 11483524 11483525 0
-11483525 0

Plan

Frequent *itemsets* mining

- Transactions database

tid	itemset
001	*Joyce*, *Beckett*, *Proust*
002	*Faulkner*, *Hemingway*, *Melville*
003	*Joyce*, *Proust*
004	*Hemingway*, *Melville*
005	*Flaubert*, *Zola*
006	*Hemingway*, *Golding*

- Support : $\mathcal{S}(\{\textit{Hemingway}, \textit{Melville}\}, \mathcal{D}) = |\{002, 004\}| = 2$
- Enumerating frequent *itemsets* :
  $\mathcal{FIM}(\mathcal{D}, \lambda) = \{A \subseteq \mathcal{I} | \mathcal{S}(A, \mathcal{D}) \geqslant \lambda\}$
- Example : $\mathcal{FIM}(\mathcal{D}, 2) =$
  $\{\{\textit{Hemingway}\}, \{\textit{Melville}\}, \{\textit{Hemingway}, \textit{Melville}\}, \{\textit{Joyce}\},$
  $\{\textit{Proust}\}, \{\textit{Joyce}, \textit{Proust}\}\}$

Frequent *itemsets* mining

Condensed representations of frequent *itemsets*

- Maximal frequent *itemsets* :

$$Max(\mathcal{D}, \lambda) = \{A \in \mathcal{FIM}(\mathcal{D}, \lambda) | \forall B \supset A, B \notin \mathcal{FIM}(\mathcal{D}, \lambda)\}$$

- Closed frequent *itemsets* :

$$Cl(\mathcal{D}, \lambda) = \{A \in \mathcal{FIM}(\mathcal{D}, \lambda) | \forall B \supset A, \mathcal{S}(B, \mathcal{D}) \neq \mathcal{S}(A, \mathcal{D})\}$$

- Example :

$$Max(\mathcal{D}, 2) = \{\{Joyce, Proust\}, \{Hemingway, Melville\}\}$$

$$Cl(\mathcal{D}, 2) = \{\{Hemingway\}, \{Joyce, Proust\}, \{Hemingway, Melville\}\}$$

Plan

CNF formula as transactions database

- Goal : reduce the number of literals using the frequent sets of literals : similar to Tseitin approach (introduce new Boolean variables)
- Items : literals
- Transactions : clauses > 2

Reduce the number of literals

- Introduce new Boolean variables :

$$(x_1 \vee \cdots \vee x_n \vee \alpha_1) \wedge \cdots \wedge (x_1 \vee \cdots \vee x_n \vee \alpha_k)$$

equivalent w.r.t. *SAT*

$$\Longrightarrow$$

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_k) \wedge (x_1 \vee \cdots \vee x_n \vee \neg y)$$

- $n \geqslant 2$ et $k > \frac{n+1}{n-1}$

- $n \times k$ literals substituted by $k + n + 1$ literals
- Quorum : $k \begin{cases} \geqslant 4 & \text{si } n = 2 \\ \geqslant 3 & \text{si } n = 3 \\ \geqslant 2 & \text{otherwise} \end{cases}$
- Not interesting to associate new variables to subsets of $\{x_1, \ldots, x_n\}$: use of condensed representation

Closed Vs. Maximal

- Maximal $\subseteq$ Closed : more informations with closed

$$(x_1 \vee \ldots \vee x_k \vee \ldots \vee x_n \vee \alpha_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \ldots \vee x_n \vee \alpha_m) \wedge$$
$$(x_1 \vee \ldots \vee x_k \vee \beta_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \beta_{m'})$$

  with $k \geqslant 2$ and $m, m' \geqslant 4$
  we suppose that the set of itemsets are frequent $\mathcal{P}(\{x_1, \ldots, x_n\})$
  $\Rightarrow$ Max $= \{\{x_1, \ldots, x_n\}\}$ and closed $= \{\{x_1, \ldots, x_k\}, \{x_1, \ldots, x_n\}\}$

- Use of $\{x_1, \ldots, x_n\}$:

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_m) \wedge$$
$$(x_1 \vee \ldots \vee x_k \vee \beta_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \beta_{m'}) \wedge$$
$$(x_1 \vee \ldots \vee x_n \vee \neg y)$$

- Use of $\{x_1, \ldots, x_k\}$:

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_m) \wedge$$
$$(z \vee \beta_1) \wedge \cdots \wedge (z \vee \beta_{m'}) \wedge$$
$$(z \vee x_{k+1} \vee \ldots \vee x_n \vee \neg y) \wedge (x_1 \vee \ldots \vee x_k \vee \neg z)$$

Subsets

- X and Y ($Y \subset X$) are both interesting if

$$\mathcal{S}(Y) - \mathcal{S}(X) > \frac{|Y| + 1}{|Y| - 1} - 1$$

- The best :
  - X if $|X| \times \mathcal{S}(X) - (\mathcal{S}(X) + |X| + 1) \geqslant |Y| \times \mathcal{S}(Y) - (\mathcal{S}(Y) + |Y| + 1)$
  - Y otherwise

- Associates a weight to frequent itemsets :

$$|X| \times \mathcal{S}(X) - (\mathcal{S}(X) + |X| + 1)$$

Overlaps

- X overlaps with Y ($X \sim Y$) : $X \cap Y \neq \emptyset$
- overlaps class (*Overlap class*) : an equivalence class (transitive closure of $\sim$)

$$Y \in [X] \;\; ssi \;\; Y = Y_1 \sim Y_2 \sim \ldots \sim Y_k = X$$

- Optimal solution $\longrightarrow$ optimal solution in an overlaps class

Overlap

- Problem :
  - $\{x_1, x_2, x_3\}$ et $\{x_2, x_3, x_4\}$ two frequents *itemsets* s.t. $\mathcal{S}(\{x_1, x_2, x_3\}) = 3$, $\mathcal{S}(\{x_2, x_3, x_4\}) = 3$ and $\mathcal{S}(\{x_1, x_2, x_3, x_4\}) = 2$
  - Use of $\{x_1, x_2, x_3\} \longrightarrow \mathcal{S}(\{x_2, x_3, x_4\}) = 2$

- X and Y ($Y \sim X$) are both interesting if
  - $\mathcal{S}(X) - \mathcal{S}(X \cup Y) > \frac{|X|+1}{|X|-1} - 1$,
  - $\mathcal{S}(Y) - \mathcal{S}(X \cup Y) > \frac{|Y|+1}{|Y|-1} - 1$, or
  - $|X \setminus Y| \geqslant k$ (resp. $|Y \setminus X| \geqslant k$) where
    - $k = 2$ if $\mathcal{S}(X) \geqslant 4$ (resp. $\mathcal{S}(Y) \geqslant 4$)
    - $k = 3$ if $\mathcal{S}(X) = 3$ (resp. $\mathcal{S}(Y) = 3$)
    - $k = 4$ otherwise

    $\Rightarrow$ Use of $X \setminus Y$ (resp. $Y \setminus X$)

Problems summary

- Choose of the quorum ?
  - 2 $\longrightarrow$ lot of useless *itemsets*
  - 3 and 4 $\longrightarrow$ loss of interesting *itemsets*
- Overlap (subsets) :
  - Simplification : overlaps classes
  - Need to compute $\mathcal{S}(X \cup Y)$
  - Optimal solution in one class ?
- A gready algorithm : loss of interesting *itemsets*

Gready Algorithm

Require: A formula ϕ, an overlap class of closed frequent itemsets C

1: **while** $C \neq \emptyset$ **do**
2: $I \leftarrow C.MostInterstingElement()$;
3: $\phi.replace(I, x)$;
4: $\phi.Add(I, x)$:
5: $C.remove(I)$;
6: $C.replaceSubset(I, x)$;
7: $C.removeUninterestingElements()$;
8: $C.updateSupports()$;
9: **end while**
10: **return** ϕ

Plan

Experiments : *Industrial* SAT instances

Instance	orig.	comp.	% red
1dlx_c_iq57_a	190 Mb	164 Mb	13.68 %
6pipe_6_ooo.*-as.sat03-413	11 Mb	7.7 Mb	30.00 %
9dlx_vliw_at_b_iq6.*-*04-347	76 Mb	65 Mb	14.47 %
abb313GPIA-9-c.*.sat04-317	21 Mb	6.9 Mb	67.14 %
E05F18	3.7 Mb	2.2 Mb	40.54 %
eq.atree.braun.11.unsat	120 Kb	72 Kb	40.00 %
eq.atree.braun.12.unsat	144 Kb	88 Kb	38.88 %
k2mul.miter.*-as.sat03-355	1.5 Mb	1.3 Mb	13.33 %
korf-15	1.2 Mb	752 Kb	37.33 %
rbcl_xits_08_UNSAT	1.1 Mb	856 Kb	22.18 %
SAT_dat.k45	3.5 Mb	2.6 Mb	25.71 %
traffic_b_unsat	18 Mb	12 Mb	33.33 %
x1mul.miter.*-as.sat03-359	1.1 Mb	928 Kb	15.63 %
9dlx_vliw_at_b_iq3	19 Mb	15 Mb	21.05 %
9dlx_vliw_at_b_iq4	31 Mb	26 Mb	16.12 %
AProVE07-09	2.8 Mb	2.7 Mb	3.57 %
eq.atree.braun.10.unsat	96 Kb	56 Kb	41.66 %
goldb-heqc-frg1mul	348 Kb	328 Kb	5.74 %
minand128	7.7 Mb	2.6 Mb	66.23 %
ndhf_xits_09_UNSAT	2.6 Mb	2.1 Mb	19.23 %
velev-pipe-o-uns-1.1-6	5.5 Mb	4.4 Mb	20.00 %

TABLE : Results of Mining4SAT : a general approach

Plan

Application : A compact representation of 2-CNF

instance	#cls	#bin	(%) bin
velev-pipe-o-uns-1.1-6	304026	268354	88,26 %
9dlx_vliw_at_b_iq2	542253	500227	92,24 %
1dlx_c_iq57_a	8562505	7567948	88,38 %
7pipe_k	751116	722278	96,16 %
SAT_dat.k100.debugged	670701	523153	78,00 %
BM_FV_2004_rule_batch	445444	339588	76,23 %
sokoban-sequential-p145-*.040-*	1413816	1364160	96,48 %
openstacks-*-p30_1.085-*	1621926	1601145	98,71 %
aaai10-planning-ipc5-*-12-step16	1029036	991140	96,31 %
k2fix_gr_rcs_w8.shuffled	271393	270136	99,53 %
homer17.shuffled	1742	1716	98,50 %
gripper13u.shuffled-as.sat03-395	38965	35984	92,34 %
grid-strips-grid-y-3.045-*	2750755	2695230	97,98 %

TABLE : Ratio of binary clauses in some SAT instances

Application : A compact representation of 2-CNF

Example

Let us consider the following 2-CNF Φ :

$$\begin{aligned}\Phi = \quad & (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \quad \wedge \\ & (x_1 \vee x_6) \wedge (x_1 \vee x_7) \wedge (x_2 \vee x_3) \wedge (x_2 \vee x_4) \quad \wedge \\ & (x_2 \vee x_5) \wedge (x_2 \vee x_6) \wedge (x_2 \vee x_7) \wedge (x_3 \vee x_4) \quad \wedge \\ & (x_3 \vee x_6) \wedge (x_3 \vee x_7) \wedge (x_3 \vee x_5) \wedge (x_4 \vee x_5) \quad \wedge \\ & (x_4 \vee x_6) \wedge (x_4 \vee x_7) \wedge (x_5 \vee x_6) \wedge (x_5 \vee x_7) \quad \wedge \\ & (x_6 \vee x_7)\end{aligned}$$

Definition (B-implication)

A *B-implication* is a Boolean formula of the following from : $x \vee \beta(x)$ where $\beta(x)$ is a conjunction of literals.

Application : A compact representation of 2-CNF

Using the complete order relation $x_1 \prec \ldots \prec x_7$ over $\mathcal{L}_\Phi$ rewrite Φ as set of B-implications $B^1_{[\vee(\wedge)]}(\Phi)$:

$\{[x_1 \vee (x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$
$[x_2 \vee (x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$
$[x_3 \vee (x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$
$[x_5 \vee (x_6 \wedge x_7)],$
$[x_6 \vee (x_7)]\}$

tid	itemset					
tid_{x_1}	x_2	x_3	x_4	x_5	x_6	x_7
tid_{x_2}		x_3	x_4	x_5	x_6	x_7
tid_{x_3}			x_4	x_5	x_6	x_7
tid_{x_4}				x_5	x_6	x_7
tid_{x_5}					x_6	x_7
tid_{x_6}						x_7

Application : A compact representation of sets of 2-CNF

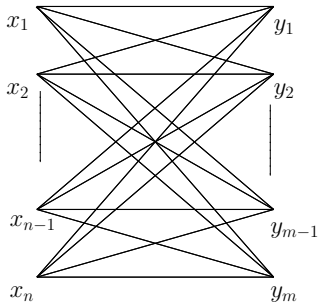
FIM process on the conjunctive part of $B^1_{\vee[\wedge]}(\Phi)$

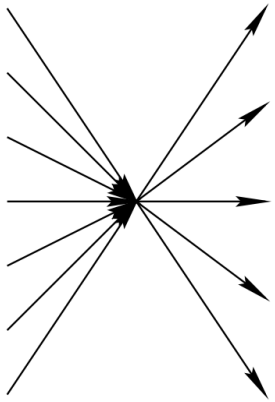
Using $\{x_5, x_6, x_7\}$ a 4-frequent itemset, we can rewrite $B^1_{[\vee(\wedge)]}(\Phi)$ as :

$$
\begin{aligned}
B^2_{\vee[\wedge]}(\Phi) = \quad & \{[x_1 \vee (x_2 \wedge x_3 \wedge y)] , \\
& [x_2 \vee (x_3 \wedge x_4 \wedge y)] , \\
& [x_3 \vee (x_4 \wedge y)] , \\
& [x_5 \vee (x_6 \wedge x_7)] , \\
& [x_6 \vee (x_7)] , \\
& [\neg y \vee (x_5 \wedge x_6 \wedge x_7)]\}
\end{aligned}
$$

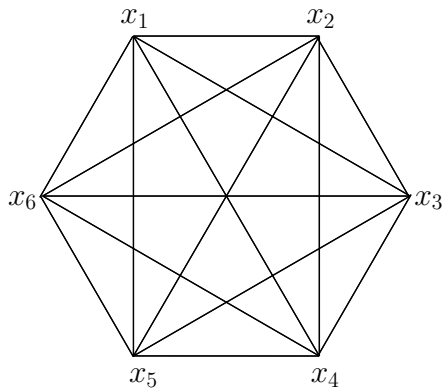
$$
\begin{aligned}
\mathrm{CNF}(B^2_{[\vee(\wedge)]}(\Phi)) = \\
& (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee y) \quad \wedge \\
& (x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee y) \quad \wedge \\
& (x_3 \vee x_4) \wedge (x_3 \vee y) \quad \wedge \\
& (x_5 \vee x_6) \wedge (x_5 \vee x_7) \quad \wedge \\
& (x_6 \vee x_7) \quad \wedge
\end{aligned}
$$

Two particular cases : bi-cliques and cliques





$n \times m$ binary clauses $\Rightarrow$ $n + m$ binary clauses and 1 new variable



$\mathcal{O}(n^2)$ binary clauses $\Rightarrow$ $\mathcal{O}(n)$ binary clauses and $\mathcal{O}(n)$ new variables

More details on bi-cliques

Let $\Phi =$
$[(x_1 \vee y_1) \wedge (x_1 \vee y_2) \wedge \cdots \wedge (x_1 \vee y_m)] \ldots [(x_n \vee y_1) \wedge (x_n \vee y_2) \wedge \cdots \wedge (x_n \vee y_m)]$

- Using a complete order relation defined by : $f(x_i) = i, f(y_j) = n + j$.
- $B_{[\vee(\wedge)]}(\Phi)$ corresponds exactly to
  $\{(x_i \vee [y_1 \wedge y_2 \wedge \cdots \wedge y_m]) | 1 \leqslant i \leqslant n\}$
- Using a single closed frequent itemset $\{y_1, y_2, \ldots, y_m\}$

$\Phi' = [\bigwedge_{1 \leqslant i \leqslant n}(x_i \vee z)] \wedge [\bigwedge_{1 \leqslant j \leqslant m}(\neg z \vee y_j)]$.

Experiments : *Industrial* SAT instances

Instance	orig.	comp.	% red
velev-pipe-o-uns-1.1-6	5.5 Mb	3.2 Mb	41.81 %
9dlx_vliw_at_b_iq2	11 Mb	6 Mb	44.45 %
1dlx_c_iq57_a	190 Mb	124 Mb	34.73 %
7pipe_k	14 Mb	5.4 Mb	61.42 %
SAT_dat.k100.debugged	16 Mb	13 Mb	18.75 %
IBM_FV_2004_rule_batch _2_31_1_SAT_dat.k80.debugged	9.7 Mb	7.5 Mb	22.68 %
sokoban-sequential-p145-*.040-*	24 Mb	14 Mb	41.66 %
openstacks-*-p30_1.085-*	30 Mb	26 Mb	13.33 %
aaai10-planning-ipc5-*-12-step16	17 Mb	12 Mb	29.41 %
k2fix_gr_rcs_w8.shuffled	3.4 Mb	1.7 Mb	50.00 %
homer17.shuffled	20 Kb	16 Kb	20.00 %
gripper13u.shuffled-as.sat03-395	524 Kb	364 Kb	30.35 %
grid-strips-grid-y-3.045-*	52 Mb	42 Mb	19.23 %

TABLE : Results of Mining4Binary : a 2-CNF approach

Application : A compact Graph Representation

For free, we can apply our approach for graphs.

- 2-CNF $\leftrightarrow$ graphs
- Adjacency lists $\leftrightarrow$ A set of B-implications
- $2 \rightarrow [4, 6, 8, 12] \leftrightarrow 2 \vee [4 \wedge 6 \wedge 8 \wedge 12]$

Plan

Compression as an Optimisation Problem

The compression problem can be formulated as an optimisation problem

Problem : $\mathcal{C}omp(\mathcal{F}, \mathcal{P})$

- **Input** : $\mathcal{F}$ a CNF formula, and $\mathcal{P}$ a set of patterns
- **Output** : a compressed formula $\mathcal{F}$ of minimal size using $\mathcal{P}$

Compression as an Optimisation Problem

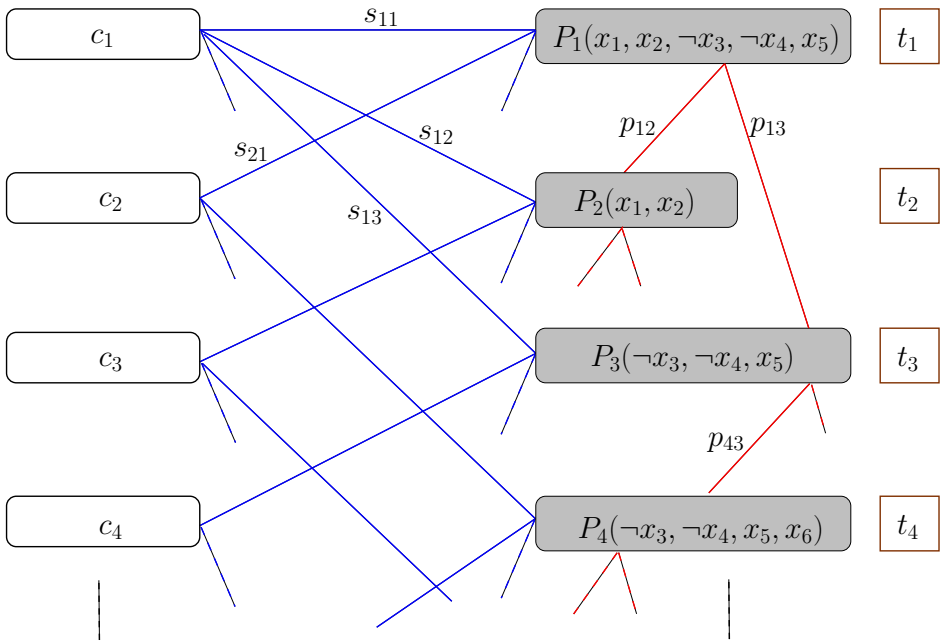
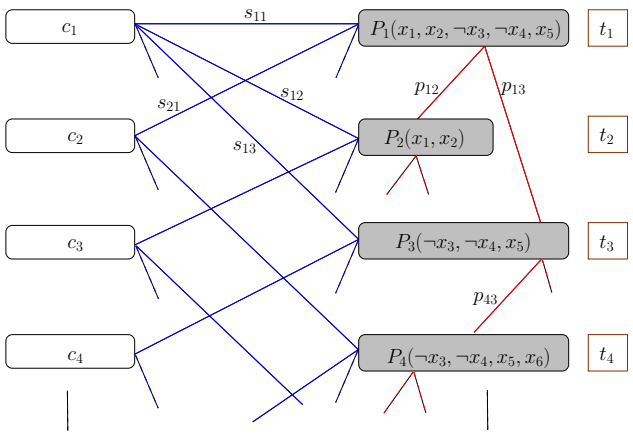


FIGURE : Compression using location problem

- If we use pattern P_j, we set t_j to 1, otherwise t_j is 0
- If we replace the literals in c_i by P_i, then we set s_{ij} to 1, and 0 otherwise.

Compression as an Optimisation Problem

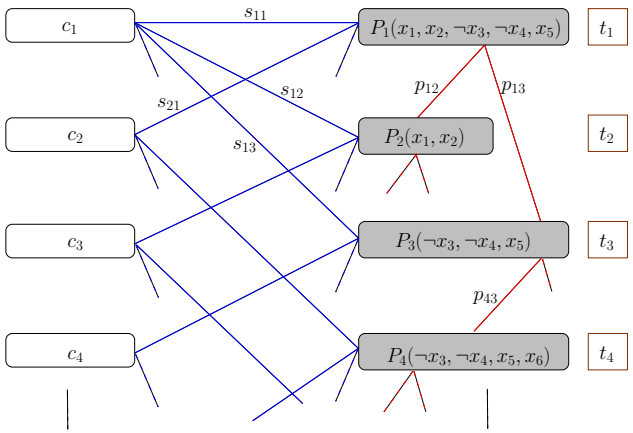


A **first formulation** as 0/1 linear program

$Max \sum(|P_j| - 1)s_{ij} - \sum(|P_j| + 1)t_j$ Maximize the reduction

1. $s_{ij} \leqslant t_j$ (C_i is replaced by P_j only when P_j is used)
2. $\sum_j s_{ij} \leqslant 1$ (C_i is replaced by only one pattern)
3. $s_{ij} \in \{0, 1\}$, $t_j \in \{0, 1\}$

Compression as an Optimisation Problem



A **second formulation** as 0/1 linear program

$Max \sum(|P_j| - 1)s_{ij} - \sum(|P_j| + 1)t_j$ Maximize the reduction

1. $s_{ij} \leqslant t_j$ (C_i is replaced by P_j only when P_j is used)
2. $s_{ij} + s_{ik} \leqslant 1$ if $P_j \cap P_k \neq \emptyset$ (C_i can be replaced by a set of disjoint patterns)
3. $s_{ij} \in \{0, 1\}$, $t_j \in \{0, 1\}$

Plan

Conclusion

- A nice application of data mining to SAT
- Size reduction of general CNF formula
- Applications to 2-CNF $\rightarrow$ automatic discovery of efficient encodings

Perspectives

- Formulate the problem as an optimization problem (reduction to location problem)
- Compress graphs and hypergraphs (under investigation)
- Find better encodings of other well known constraints (e.g. all different constraint)
- Discover other structures in Boolean formulae
- . . .

- Cross-fertilization between SAT/CP and Data mining

Thank you for your attention.