Explication de clauses et de formules dans Max-SAT

Matthieu Py, Mohamed Sami Cherif, Djamal Habet

Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

{matthieu.py, mohamedsami.cherif, djamal.habet}@univ-amu.fr

Résumé

Dans ce papier, on s'intéresse à la construction de transformations préservant l'équivalence Max-SAT afin d'inférer de l'information (clause ou formule) à partir d'une formule donnée. Dans ce but, on définit et caractérise la notion de clauses et formules explicables, on propose un système de preuve complet pour l'inférence dans Max-SAT et un algorithme associé pour expliquer ou réfuter l'explicabilité de n'importe quelle clause ou formule. On donne enfin des bornes théoriques sur la taille des transformations calculées. Cet article résume le travail publié à la conférence ICTAI 2021 [3].

Mots-clés

Max-SAT, Systèmes de preuve, Inférence

1 Introduction

Étant donnée une formule sous Forme Normale Conjonctive, le problème Max-SAT consiste à déterminer le nombre maximum de clauses qu'il est possible de satisfaire par une affectation des variables. Un système de preuves bien connu pour Max-SAT est basé sur la règle de la max-résolution [1] qui est l'adaptation de la règle de résolution définie dans le contexte du problème SAT [4]. La max-résolution, en tant que système de preuve, est complète pour la résolution du problème Max-SAT, c'est à dire qu'il est possible, à partir de n'importe quelle formule sous forme normale conjonctive ϕ, de déduire une formule équivalente composée d'une sous-formule satisfiable et d'un multi-ensemble de clauses vides, dont la taille est l'optimum de ϕ.

Cet article s'intéresse aussi aux systèmes de preuve pour Max-SAT et en particulier à la capacité qu'ont ces systèmes pour la déduction d'informations (sous forme de clause ou formule) à partir d'une formule donnée. Comme la max-résolution est inférentiellement incomplète, on propose alors un nouveau système de preuve (ExC) inférentiellement complet dont on étudie les relations avec d'autres système de preuve connus. On propose aussi un algorithme (l'algorithme d'explication) permettant d'expliquer ou de démontrer l'inexplicabilité de n'importe quelle clause ou formule dans Max-SAT. On démontre enfin les bornes théoriques sur la taille, en nombre d'étapes d'inférence, des explications calculées.

2 Explication de clauses

Pour commencer, on introduit la notion d'explicabilité, qui peut s'étendre naturellement aux formules (on détaille ici uniquement l'explication de clauses).

Définition 1 (Clause explicable et explication de clause)**.** *Soit ϕ une formule CNF et c une clause non tautologique, on dit que c est explicable dans ϕ s'il existe une formule CNF ϕ' telle que $\phi \equiv c \wedge \phi'$. La transformation de ϕ vers $c \wedge \phi'$ est appelée explication de c dans ϕ.*

Pour faire l'explication de n'importe quelle clause dans Max-SAT, on va simplement se ramener à deux cas extrêmes de clauses explicables ou inexplicables.

Proposition 1. *Soit ϕ une formule CNF et c une clause non tautologique. Si $\exists c' \in \phi$ telle que c' sous-somme c (les littéraux de c' sont dans c) alors c est explicable dans ϕ.*

Proposition 2. *Soit ϕ une formule CNF et c une clause non tautologique. Si $\forall c' \in \phi$, c' s'oppose à c (deux littéraux de c' et c s'opposent), alors c est inexplicable dans ϕ.*

Pour faire l'explication de formule, on va construire itérativement des explications de clauses. Remarquons maintenant que la max-résolution est inférentiellement incomplète, c'est à dire qu'il existe des clauses explicables qui ne peuvent pas être expliquées par max-résolution.

Définition 2 (Max-Résolution [1])**.** *Étant données deux clauses $c_1 = x \vee a_1 \vee \cdots \vee a_s$ et $c_2 = \overline{x} \vee b_1 \vee \cdots \vee b_t$ avec $A = a_1 \vee \cdots \vee a_s$ et $B = b_1 \vee \cdots \vee b_t$, la max-résolution remplace c_1 et c_2 par un ensemble de nouvelles clauses dont c_3 est la clause résolvante et $cc_1, \ldots, cc_{t+s}$ sont des clauses de compensation :*

$$\frac{c_1 = x \vee A \qquad\qquad c_2 = \overline{x} \vee B}{\begin{array}{c} c_3 = A \vee B \\ cc_1 = x \vee A \vee \overline{b_1} \\ cc_2 = x \vee A \vee b_1 \vee \overline{b_2} \\ \cdots \\ cc_t = x \vee A \vee b_1 \vee \cdots \vee b_{t-1} \vee \overline{b_t} \\ cc_{t+1} = \overline{x} \vee B \vee \overline{a_1} \\ cc_{t+2} = \overline{x} \vee B \vee a_1 \vee \overline{a_2} \\ \cdots \\ cc_{t+s} = \overline{x} \vee B \vee a_1 \vee \cdots \vee a_{s-1} \vee \overline{a_s} \end{array}}$$

Proposition 3. *[2] La max-résolution n'est pas inférentiellement complète.*

3 Le système de preuve ExC

Comme la max-résolution est inférentiellement incomplète, on propose un système inférentiellement complet.

Définition 3. *Le système de preuve ExC est composé de deux règles : la coupe symétrique et l'expansion.*

Définition 4 (Coupe symétrique)**.** *Soient deux clauses $c_1 = x \vee A$ et $c_2 = \overline{x} \vee A$ où A est une disjonction de littéraux, la coupe symétrique remplace c_1 et c_2 par $c_3 = (A)$:*

$$\frac{c_1 = (x \vee A) \quad c_2 = (\overline{x} \vee A)}{c_3 = (A)}$$

Définition 5 (Expansion)**.** *Soit une clause $c = (A)$ où A est une disjonction de littéraux et $B = b_1 \vee ... \vee b_k$, la règle d'expansion remplace c par les clauses suivantes :*

$$\frac{c_1 = (A)}{\begin{array}{c} cc_1 = (A \vee \overline{b_1}) \\ cc_2 = (A \vee b_1 \vee \overline{b_2}) \\ ... \\ cc_k = (A \vee b_1 \vee ... \vee b_{k-1} \vee \overline{b_k}) \\ c_2 = (A \vee B) \end{array}}$$

Proposition 4. *ExC est inférentiellement complet.*

4 L'algorithme d'explication

L'algorithme d'explication permet d'expliquer n'importe quelle clause ou de réfuter son explicabilité. Pour cela, on teste si la clause à expliquer tombe sur des deux cas extrêmes évoqués avant. Si c'est le cas, on conclut (explicable ou non explicable). Sinon, on remplace la clause c par deux clauses $c \vee x$ et $c \vee \overline{x}$ et il suffit d'expliquer ces deux clauses pour expliquer c.

Théorème 1. *Soit ϕ une formule CNF avec n variables et c une clause explicable dans ϕ. Il existe une explication de c dans ϕ utilisant les règles de ExC et contenant $O(2^n)$ étapes d'inférence.*

Exemple 1. *On considère la formule $\phi = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3) \wedge (\overline{x_1})$ et on veut expliquer la clause $c = (x_1)$. On note T et C respectivement la séquence de transformations et la séquence de clauses qui doivent être expliquées. On construit progressivement l'explication $c = (x_1)$, où l'ordre des variables est l'ordre lexicographique, comme ci-dessous :*

1. *$T = ()$ et $C = \{(x1)\}$*
   - *(x_1) n'est sous-sommée par aucune clause de ϕ.*
   - *(x_1) n'est pas opposé à toutes les clauses de ϕ.*
   - *On choisit la variable x_2 qui n'est pas dans (x_1). On doit maintenant expliquer les clauses $(x_1 \vee x_2)$ et $(x_1 \vee \overline{x_2})$. L'application de la coupe symétrique sur ces clauses est aussi ajoutée à T.*
2. *$T = (T_1)$ où $T_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \vdash (x_1)$ and $C = \{(x_1 \vee x_2), (x_1 \vee \overline{x_2})\}$*
   - *La clause $(x_1 \vee x_2) \in \phi$ et est donc trivialement expliquée.*
3. *$T = (T_1)$ et $C = \{(x_1 \vee \overline{x_2})\}$*
   - *$(x_1 \vee \overline{x_2})$ n'est sous-sommée par aucune clause de ϕ.*
   - *$(x_1 \vee \overline{x_2})$ n'est pas opposé à toutes les clauses de ϕ.*
   - *On choisit la variable x_3 qui n'est pas dans $(x_1 \vee \overline{x_2})$. On doit maintenant expliquer les clauses $(x_1 \vee \overline{x_2} \vee x_3)$ et $(x_1 \vee \overline{x_2} \vee \overline{x_3})$. L'application de la coupe symétrique sur ces clauses est aussi ajoutée à T.*
4. *$T = (T_2, T_1)$ où $T_2 = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \vdash (x_1 \vee \overline{x_2})$ and $C = \{(x_1 \vee \overline{x_2} \vee x_3), (x_1 \vee \overline{x_2} \vee \overline{x_3})\}$*
   - *La clause $(x_1 \vee \overline{x_2} \vee x_3)$ est expliquée par expansion de $(x_3) \in \phi$ qui la sous-somme.*
5. *$T = (T_3, T_2, T_1)$ où $T_3 = (x_3) \vdash (\overline{x_1} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$ and $C = \{(x_1 \vee \overline{x_2} \vee \overline{x_3})\}$*
   - *La clause $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \in \phi$ et est donc trivialement expliquée.*
6. *$T = (T_3, T_2, T_1)$ et $C = \emptyset$. L'explication de la clause (x_1) est complète.*

On conclut que $\phi \vdash (\overline{x_1} \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1) \wedge (\overline{x_1})$ en appliquant la transformation T, représentée dans la Figure 1. Remarquez comment la transformation T a été construite depuis la fin. Les clauses $(\overline{x_1} \vee x_3)$ et $(x_1 \vee x_2 \vee x_3)$ sont des clauses de compensation, essentielles pour préserver l'équivalence Max-SAT.

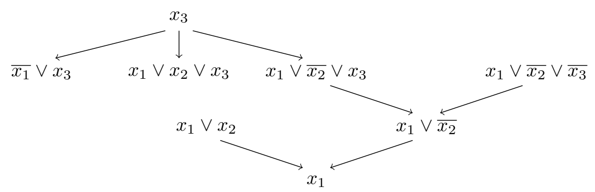

FIGURE 1 – Explication de la clause (x_1) dans ϕ

Ce résultat sur l'explication de clauses s'étend également aux formules ainsi qu'à la construction de preuves pour Max-SAT (voir les détails dans l'article [3]).

Références

[1] María Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for Max-SAT. In *Artificial Intelligence*, volume 171, pages 606–618, 2007.

[2] Javier Larrosa and Emma Rollon. Towards a Better Understanding of (Partial Weighted) MaxSAT Proof Systems. In *SAT*, pages 218–232, 2020.

[3] Matthieu Py, Mohamed Sami Cherif, and Djamal Habet. Inferring Clauses and Formulas in Max-SAT. In *33rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2021*. IEEE, 2021.

[4] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, 12 :23–41, 1965.