

De la résolution à la max-résolution

Mohamed Sami Cherif, Djamel Habet, Matthieu Py

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

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

Résumé

Adapter une preuve par résolution en une preuve par max-résolution sans augmenter considérablement sa taille est une question ouverte. En effet, la seule classe dont l'adaptation est connue et triviale est celle où les clauses sont utilisées au plus une fois dans les preuves. Dans ce papier, on propose une nouvelle classe de résolution, appelée résolution sans croisement, dans laquelle les clauses réutilisées plusieurs fois sont exploitées indépendamment pour générer de nouvelles informations. On démontre que les preuves de cette classe peuvent être adaptées en preuves par max-résolution sans augmentation considérable de leur taille en utilisant les clauses de compensation générées par la max-résolution.

Mots-clés

Preuves, Résolution sans croisement, Max-résolution.

Abstract

Adapting a SAT resolution proof into a Max-SAT resolution proof without considerably increasing the size of the proof is an open problem. Read-once resolution, where each clause is used at most once in the proof, represents the only fragment of resolution for which an adaptation using exclusively Max-SAT resolution is known and trivial. Proofs containing non read-once clauses are difficult to adapt because the Max-SAT resolution rule replaces the premises by the conclusions. This paper contributes to this open problem by defining, for the first time since the introduction of Max-SAT resolution, a new fragment of resolution whose proofs can be adapted to Max-SAT resolution proofs without substantially increasing their size. In this fragment, called crossing-free resolution, non read-once clauses are used independently to infer new information thus enabling to bring along each non read-once clause while unfolding the proof until a substitute is required.

Keywords

Proof, Max-SAT, Resolution.

1 Introduction

Le problème Max-SAT est une extension d'optimisation du problème de satisfiabilité (SAT) et consiste, étant donné une formule en Forme Normale Conjonctive (FNC), à déterminer le nombre maximum de clauses

qu'il est possible de satisfaire par une interprétation des variables. Ce formalisme bien connu est utilisé pour représenter et résoudre de nombreux problèmes industriels et académiques [3, 4]. SAT et Max-SAT sont fortement liés et partagent de nombreux aspects. En effet, les techniques de résolution pour SAT sont souvent utilisées dans le cadre de Max-SAT, en particulier dans les algorithmes faisant appel à des oracles SAT ou encore dans les algorithmes par séparation et évaluation pour Max-SAT [1, 2, 20]. Cependant, en théorie, combler le fossé entre l'inférence SAT et Max-SAT reste l'un des principaux défis de ces dernières années.

L'un des premiers systèmes de preuve pour Max-SAT est basé sur une règle d'inférence, appelée max-résolution [6, 7, 16], qui est une extension de la règle de résolution [27], introduite dans le contexte de SAT. La max-résolution est la règle d'inférence la plus étudiée dans le cadre Max-SAT, à la fois en théorie et en pratique [1, 5, 17, 18, 22, 23, 26]. Cependant, adapter une preuve par résolution pour obtenir une preuve par max-résolution valide de taille raisonnable reste un problème ouvert. Bonnet et al. affirment que "il semble difficile d'adapter une preuve par résolution classique pour obtenir une preuve par max-résolution, et démontrer que cela est possible sans augmenter considérablement la taille de la preuve est une question ouverte"¹. En effet, contrairement à la résolution, la max-résolution remplace les prémisses par les conclusions. Par conséquent, les preuves par résolution *read-once*, où chaque clause est utilisée une fois dans la preuve, représentent le seul fragment de résolution pour lequel une adaptation immédiate et triviale est possible [7, 12].

Des travaux récents [11, 23] tentent de contourner ce problème en permettant l'utilisation de la règle *split*, qui permet intuitivement de dupliquer une clause en ajoutant un littéral, pour adapter linéairement les réfutations par résolution *tree-like*. Plus précisément, l'adaptation tire parti de la structure de telles preuves et applique le *split* pour dupliquer les clauses utilisées plusieurs fois dans la preuve. Cependant, les preuves résultantes sont dans un nouveau système de preuve, appelé ResS [17], dans lequel la max-résolution est augmentée avec le *split*. Pour combler réellement le fossé entre la résolution SAT et Max-SAT, les clauses réutilisées doivent être inférées au travers des

¹traduction du passage original en anglais dans [7].

clauses de compensation générées par la max-résolution. Dans cet article, nous contribuons à ce problème ouvert en identifiant un nouveau fragment de résolution, que nous appelons résolution sans croisement, pour lequel une adaptation utilisant uniquement la max-résolution est possible sans augmenter considérablement la taille de la preuve. Les preuves sans croisement sont définies à l'aide des sous-dérivations des clauses utilisées plusieurs fois. Informellement, ces clauses sont utilisées indépendamment pour déduire de nouvelles informations dans de telles preuves. Nous montrons que l'adaptation de preuves par résolution sans croisement à des preuves par max-résolution sans augmentation de leur taille est possible modulo quelques subtilités syntaxiques mineures. De plus, nous montrons que les motifs diamants, qui sont exponentiels pour l'adaptation introduite dans [23], appartiennent au fragment de résolution sans croisement et peuvent être adaptés sans augmentation de leur taille.

Ce papier est organisé comme suit. Dans, la section 2, on présente quelques définitions et notations, ainsi que le contexte nécessaire et les travaux connexes sur la résolution SAT et Max-SAT. Le fragment de résolution sans croisement est introduit dans la section 3 et son adaptation est présentée dans la section 4. On étudie les motifs diamants et on montre qu'ils peuvent être adaptés sans augmenter leur taille dans la section 5. Enfin, on conclut et discute les perspectives dans la section 6.

2 Préliminaires

2.1 Définitions and Notations

Soit X l'ensemble des variables propositionnelles. Un littéral l est une variable $x \in X$ ou sa négation \bar{x} . Une clause C est une disjonction (ou un ensemble) de littéraux. Si $|C| = 1$, C est une clause unitaire. Une formule en Forme Normale Conjonctive (FNC) ϕ est une conjonction (ou un multi-ensemble) de clauses. Une interprétation $I : X \rightarrow \{\text{vrai}, \text{faux}\}$ affecte une valeur booléenne à chaque variable et peut être représentée comme un ensemble de littéraux. Un littéral l est satisfait (resp. falsifié) par une affectation I si $l \in I$ (resp. $\bar{l} \in I$). Une clause C est satisfaite par une affectation I si au moins un de ses littéraux est satisfait par I , sinon elle est falsifiée par I . La clause vide \square ne contient aucun littéral et est toujours falsifiée. Une clause C est une tautologie si elle contient à la fois un littéral et sa négation, i.e. $\exists l \in C$ tel que $\bar{l} \in C$, et dans ce cas, elle est toujours satisfaite. Une clause C s'oppose à une clause C' si C contient un littéral dont la négation est dans C' , i.e. $\exists l \in C$ tel que $\bar{l} \in C'$. On note $\text{var}(l)$ et $\text{vars}(C)$ les variables apparaissant respectivement dans le littéral l et dans la clause C . La taille d'une clause est le nombre de ces littéraux. Une formule FNC ϕ est satisfaite par une affectation I , que l'on appelle modèle de ϕ , si chaque clause $C \in \phi$ est satisfaite par I , sinon elle est falsifiée par I .

Le problème de Satisfiabilité (SAT) consiste à déterminer s'il existe une affectation I qui satisfait une formule FNC donnée ϕ . Dans le cas où une telle affectation existe, on

dit que ϕ est satisfiable, sinon on dit que ϕ est insatisfiable. Le coût d'une interprétation I , noté $\text{cost}_I(\phi)$, est le nombre de clauses falsifiées par I . Le problème Max-SAT est une extension d'optimisation de SAT qui, pour une formule CNF donnée ϕ , consiste à déterminer le nombre maximum de clauses pouvant être satisfaites par une affectation des variables. De manière équivalente, il consiste à déterminer le nombre minimum de clauses que chaque affectation doit falsifier, i.e. $\min_I \text{cost}_I(\phi)$.

2.2 Résolution SAT

Un système de preuve et de réfutation bien connu pour SAT est basé sur la règle de résolution [27]. Étant donné deux clauses opposées, cette règle, définie ci-dessous, déduit une clause résolvente qui peut être ajoutée à la formule. Une preuve (ou dérivation) par résolution d'une clause C à partir d'une formule ϕ est une suite finie de résolutions partant des clauses de ϕ et déduisant C , habituellement représentée comme une suite finie de clauses. Si C est la clause vide \square , la preuve est appelée réfutation de ϕ . Une preuve par résolution peut également être représentée sous la forme d'un graphe acyclique orienté dont les nœuds sont des clauses de la preuve ayant deux ou zéro arcs entrants (resp. s'il s'agit de résolventes ou de clauses de la formule initiale). La taille d'une dérivation par résolution π , notée $s(\pi)$, est le nombre de résolventes qu'elle contient tandis que sa largeur, notée $w(\pi)$, est la taille maximale de toutes ses clauses.

Définition 1 (Résolution [27]). *Étant donné deux clauses opposées C_1 et C_2 , la règle de résolution est définie comme suit :*

$$\frac{C_1 = x \vee A \quad C_2 = \bar{x} \vee B}{C_3 = A \vee B}$$

De nombreuses classes restreintes de résolution ont été étudiées dans la littérature, par exemple la résolution *read-once* [13], la résolution arborescente (*tree-like*) [15] et la résolution linéaire [21] entre autres. En particulier, une preuve par résolution est *read-once* si chaque clause est *read-once*, c'est à dire utilisée au plus une fois dans la preuve. De même, une dérivation par résolution est arborescente si chaque clause intermédiaire, c'est-à-dire résolvente, est utilisée au plus une fois dans la dérivation. La résolution linéaire, définie ci-dessous, se situe entre la résolution arborescente et la résolution générale en termes de complexité de preuves [8, 9]. Notons que, lorsque la première condition de (b) est vérifiée dans la définition, la clause D_i est appelée la clause *input parent* de C_{i+1} .

Définition 2 (Résolution linéaire [21]). *Soit ϕ une formule FNC et C une clause. Une dérivation par résolution linéaire de C à partir de ϕ est une suite de clauses C_1, \dots, C_m tel que :*

- (a) $C_1 \in \phi$ et $C_m = C$
- (b) Pour tout $i < m$, C_{i+1} est la résolvente de C_i soit avec une clause D_i de ϕ soit avec une clause C_k pour un certain $k < i$.

2.3 Résolution Max-SAT

Le calculus de max-résolution (MaxRes) est l'un des systèmes de preuve pour Max-SAT les plus étudiés dans la littérature qui repose sur une règle d'inférence étendant la résolution pour Max-SAT [6, 7, 16]. Outre la clause résolvente, cette règle, appelée max-résolution et définie ci-après, introduit de nouvelles clauses dites de compensation et qui sont indispensables pour préserver l'équivalence Max-SAT. En tant que règle valide et complète pour Max-SAT [7], la max-résolution joue un rôle important dans le cadre de Max-SAT [5, 17, 23, 26]. En particulier, pour une formule CNF donnée, il est possible de générer une preuve par max-résolution de son optimum en appliquant l'algorithme de saturation [7]. De plus, cette règle est largement utilisée et étudiée dans le cadre des algorithmes par séparation et évaluation pour Max-SAT [1, 10, 14, 18] et plus marginalement dans le cadre des algorithmes faisant appel à des oracles SAT [12, 22].

Définition 3 (Max-résolution [6, 7, 16]). *Étant donné deux clauses opposées C_1 et C_2 , la règle de max-résolution est définie comme suit :*

$$\frac{C_1 = x \vee A \quad C_2 = \bar{x} \vee B}{C_r = A \vee B}$$

$$CC_1 = x \vee A \vee \bar{B}$$

$$CC_2 = \bar{x} \vee \bar{A} \vee B$$

où C_r est la clause résolvente et CC_1, CC_2 sont les clauses de compensation.

Notons que la réécriture suivante est utilisée pour représenter les clauses de compensation sous forme compacte : $C \vee \overline{a_1 \vee a_2 \vee \dots \vee a_n} = (C \vee \bar{a}_1) \wedge (C \vee \bar{a}_2) \wedge \dots \wedge (C \vee \bar{a}_n)$. Cette réécriture a été introduite dans [16] comme une règle récursive pour transformer les clauses de compensation en forme CNF. Cela implique également que la max-résolution dépend de l'ordre des littéraux dans les clause prémisses, comme indiqué dans [7, 16]. Par souci de simplification et en abusant certaines notations, nous autoriserons l'utilisation de cette réécriture comme deux règles à part entière pour manipuler les clauses sous forme compacte. Nous appellerons la réécriture gauche-droite *expansion* et droite-gauche *compactage*. Nous discutons cette subtilité à la suite du théorème 1 dans la section 4.

Une preuve (ou dérivation) par max-résolution d'une formule ϕ' (ou simplement une clause $C \in \phi'$) à partir de ϕ est une séquence finie de max-résolutions à partir des clauses de ϕ et déduisant ϕ' , généralement représentée par une suite finie de formules. Nous autoriserons l'ajout de clauses tautologiques à n'importe quelle formule dans la preuve. Nous discutons cette subtilité syntaxique à la fin de la section 4. Une preuve par max-résolution peut également être représentée comme un graphe biparti acyclique orienté dont les nœuds sont soit des clauses, soit des étapes d'inférence (auquel cas ils seront omis pour plus de simplicité). Une séquence d'étapes de max-résolution déduisant une clause vide est appelée réfutation

par max-résolution. Notons que d'autres règles d'inférence et systèmes de preuve ont également été étudiés dans le cadre de Max-SAT [5, 17, 19, 26].

Contrairement à la résolution, la max-résolution remplace les prémisses par les conclusions. Larrosa et al. décrivent la max-résolution comme un "*mouvement de connaissances*" [16]. Du fait de cette spécificité, il n'est pas facile d'adapter une preuve par résolution en une preuve par max-résolution. En effet, dans les preuves par résolution, plusieurs étapes de résolution peuvent partager la même prémisse, car les prémisses ne sont pas consommées après l'application d'une étape de résolution. En revanche, les prémisses d'une étape de max-résolution sont consommées après son application. Par conséquent, l'adaptation immédiate d'une preuve par résolution pour Max-SAT n'est possible que si elle est *read-once* [7, 6, 12]. Adapter n'importe quelle preuve par résolution en une preuve pour Max-SAT sans augmenter considérablement sa taille reste un problème ouvert.

Des travaux récents [11, 23] augmentent la règle de max-résolution par la règle *split*, formant un nouveau système plus fort que MaxRes et appelé ResS [17], pour adapter linéairement les réfutations de résolution arborescente en réfutations ResS. Plus précisément, l'adaptation tire parti de la structure de telles preuves et applique le *split*, qui permet intuitivement de dupliquer une clause en ajoutant un littéral, pour dupliquer les clauses utilisées plusieurs fois dans la preuve. De plus, l'algorithme de substitution introduit dans [25] permet également de générer des remplaçants pour les clauses non *read-once* en faisant appel à des oracles SAT, mais aucune garantie n'est fournie sur la taille des réfutations ResS calculées. La résolution *read-once* reste donc le seul fragment de résolution pour lequel une adaptation utilisant exclusivement la max-résolution est possible sans augmenter considérablement la taille de la preuve. Dans la section suivante, nous définissons un nouveau fragment de résolution pour lequel une telle adaptation est possible.

3 Résolution sans croisement

La difficulté majeure dans l'adaptation des preuves par résolution à celles par résolution Max-SAT réside dans l'inférence de remplaçants aux clauses non *read-once*. En effet, de tels remplaçants doivent être naturellement déduits à l'aide de la max-résolution lors du déroulement de la preuve initiale, contrairement aux travaux précédents [11, 23] où les clauses non *read-once* sont artificiellement dupliquées à l'aide de la règle *split* avant de dérouler effectivement la preuve. Dans cette section, nous définissons un nouveau fragment de résolution, appelé résolution sans croisement. L'idée derrière ce raffinement est d'assurer une maniabilité suffisante des preuves en termes de structure afin d'inférer des remplaçants aux clauses non *read-once* lorsque cela est nécessaire. À cette fin, nous définissons ci-dessous la notion de sous-dérivation d'une clause non *read-once*.

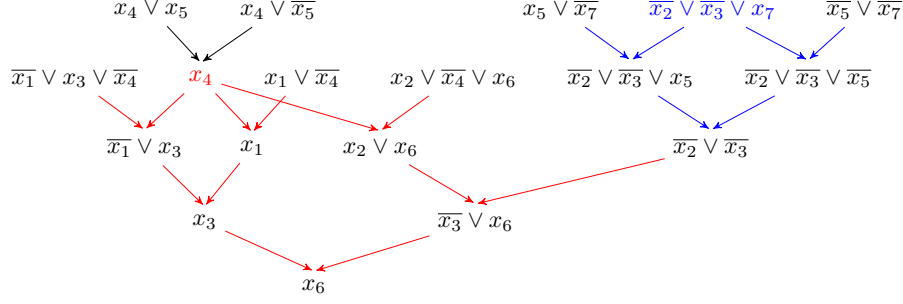


Figure 1: Sous-dérivations de clauses non *read-once* dans une preuve par résolution sans croisement.

Définition 4 (sous-dérivation de clause non *read-once*). Soit ϕ une formule FNC et π une dérivation par résolution de la clause C à partir de ϕ . La sous-dérivation d'une clause non *read-once* C' dans π , notée $ED(C')$, est la sous-dérivation de π formée par toutes les étapes de résolution dans les chemins commençant par C' dans π jusqu'à leur premier nœud de jonction. Nous appelons la clause dérivée dans le nœud de jonction, la clause issue de C' , notée $EC(C')$.

Exemple 1. On considère la dérivation par résolution π représentée dans la figure 1 de la clause $C = x_6$ à partir de la formule $\phi = \{\bar{x}_1 \vee x_3 \vee \bar{x}_4, x_4 \vee x_5, x_4 \vee \bar{x}_5, x_1 \vee \bar{x}_4, x_2 \vee \bar{x}_4 \vee x_6, x_5 \vee \bar{x}_7, \bar{x}_2 \vee \bar{x}_3 \vee x_7, \bar{x}_5 \vee \bar{x}_7\}$. Les clauses non *read-once* x_4 et $\bar{x}_2 \vee \bar{x}_3 \vee x_7$ et leurs sous-dérivations sont respectivement représentées en rouge et en bleu. De plus, on a $EC(x_4) = x_6$ et $EC(\bar{x}_2 \vee \bar{x}_3 \vee x_7) = \bar{x}_2 \vee \bar{x}_3$.

Rappelons que les clauses prémisses sont consommées après l'application de la max-résolution. Par conséquent, il semble difficile d'adapter les dérivations par résolution dans lesquelles sous-dérivations de clauses non *read-once* se croisent. En effet, dans de tels cas, la formule peut évoluer significativement car les clauses de compensation peuvent être utilisées tandis que d'autres peuvent être générées. C'est pour cela que la résolution sans croisement garantit que toutes les sous-dérivations de clauses non *read-once* soient disjointes, c'est-à-dire qu'elles ne se croisent pas, comme défini ci-dessous.

Définition 5 (Résolution sans croisement). Soit ϕ une formule FNC et π une preuve par résolution de la clause C à partir de ϕ . π est dite sans croisement ssi pour chaque paire de clauses non *read-once* (C_1, C_2) de π , $ED(C_1)$ et $ED(C_2)$ sont disjointes, c'est-à-dire qu'ils ne partagent aucun arc.

Exemple 2. On considère la même formule ϕ que l'exemple 1. La preuve par résolution π de la clause $C = x_6$ à partir de ϕ représentée dans la figure 1 est sans croisement puisque les sous-dérivations des clauses non *read-once* x_4 et $\bar{x}_2 \vee \bar{x}_3 \vee x_7$ sont disjointes.

Notons que le fragment de résolution sans croisement implique une propriété intéressante établie dans la proposition suivante. Intuitivement, cette propriété

garantit que les clauses non *read-once* soient utilisées indépendamment pour déduire de nouvelles informations dans les preuves par résolution sans croisement. Cela implique que chaque sous-dérivation de clause non *read-once* dans une preuve par résolution sans croisement peut être adaptée indépendamment comme décrit dans la section suivante.

Proposition 1. Soit ϕ une formule FNC, π une preuve par résolution sans croisement de la clause C à partir de ϕ et C' une clause non *read-once* dans π . toute clause Cl dans $ED(C')$ telle que $Cl \notin \{C', EC(C')\}$ est *read-once*.

Preuve. Soit Cl une clause dans $ED(C')$ telle que $Cl \notin \{C', EC(C')\}$. Clairement, si Cl n'est pas *read-once* alors $ED(Cl)$ partage au moins un arc avec $ED(C')$ ce qui est absurde puisque π est sans croisement.

4 De la résolution sans croisement à la max-résolution

Dans cette section, nous montrons que les preuves par résolution sans croisement peuvent être adaptées en dérivations par max-résolution modulo quelques subtilités syntaxiques mineures sans augmenter considérablement leur taille. Dans la proposition suivante, nous fournissons d'abord quelques motifs qui seront utilisés dans l'adaptation.

Proposition 2. Soient A, B, C et $\{l\}$ quatre ensembles de littéraux tels que $|C| > 0$. Les déductions suivantes peuvent être effectuées en $O(|C|)$ étapes d'inférence :

- $(A \vee C) \wedge (B \vee \bar{C}) \vdash_{MaxRes} A \vee B$
- $(l \vee A \vee \bar{C}) \wedge (\bar{l} \vee B) \vdash_{MaxRes} A \vee B \vee \bar{C}$
- $(l \vee A \vee \bar{C}) \wedge (\bar{l} \vee B \vee \bar{C}) \vdash_{MaxRes} A \vee B \vee \bar{C}$

Preuve. On donne la preuve du motif (a) par récurrence sur $|C| = n$:

- Si $n = 1$, alors $C = \{l'\}$. Clairement, $(A \vee l') \wedge (B \vee \bar{l}') \vdash_{MaxRes} A \vee B$ par application d'une seule étape de max-résolution sur la variable $var(l')$.
- Supposons que $n > 1$ et soit $l' \in C$. Par récurrence, on a $(A \vee C) \wedge (B \vee \bar{C} \setminus \{l'\}) \vdash_{MaxRes} A \vee B \vee l'$

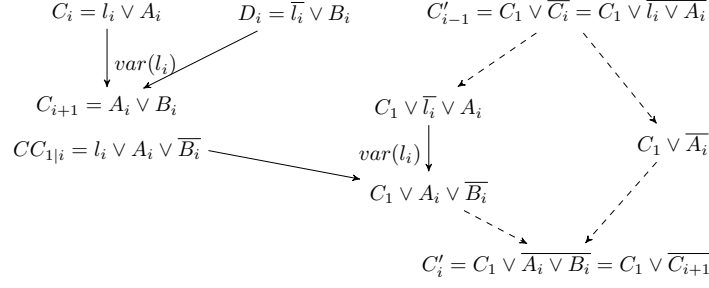


Figure 2: Étape d'induction pour déduire C'_i à la $i^{\text{ème}}$ étape. Les arcs continus représentent l'application de la max-résolution tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

en $n - 1$ étapes d'inférence. De plus, on a $B \vee \bar{C} = (B \vee \bar{C} \setminus \{l'\}) \wedge (B \vee l')$ par expansion et $(A \vee B \vee l') \wedge (B \vee \bar{l}') \vdash_{MaxRes} A \vee B$ par application d'une étape de max-résolution sur $var(l')$. Par conséquent, on conclut que l'inférence $(A \vee C) \wedge (B \vee \bar{C}) \vdash_{MaxRes} A \vee B$ est possible en $O(n)$ étapes.

Les preuves pour les motifs (b) et (c) sont similaires par récurrence sur $|C|$

Dans la suite, on commence à traiter l'adaptation des preuve par résolution sans croisement et en particulier les sous-dérivations des clauses non *read-once*. Pour générer un remplaçant à une clause non *read-once*, notons qu'on peut utiliser les littéraux dans les nœuds de jonction (c.f. lemme 1 dans [23]) de la sous-dérivation, c'est-à-dire les nœuds où les chemins partant de la clause non *read-once* s'intersectent. Pour générer de tels substituts en utilisant la max-résolution, nous commençons par traiter les parties linéaires *read-once* dans la preuve. De manière informelle, on veut emmener chaque clause non *read-once* quand on déplie la preuve jusqu'à ce qu'elles soient réutilisées. Ceci est formellement établi pour les parties linéaires *read-once* de la preuve dans le lemme suivant. Notons que les implications de l'égalité $\stackrel{*}{=}$ dans la preuve seront discutées plus en détail à la fin de la section.

Lemme 1. Soit ϕ une formule FNC, $\pi = C_1, \dots, C_{s(\pi)}$ une preuve par résolution linéaire *read-once* de la clause $C \neq \square$ à partir de ϕ . On peut déduire $\phi \vdash_{MaxRes} C \wedge (C_1 \vee \bar{C})$ en $O(s(\pi) \times w(\pi))$ étapes d'inférence.

Preuve. Soit $m = s(\pi)$. Étant donné que π est *read-once*, elle peut être trivialement adaptée en une dérivation par max-résolution de C à partir de ϕ qui est de même taille, en remplaçant chaque étape de résolution par une max-résolution [7, 12]. Dans la suite, on prouve par induction sur $i \in \{1, \dots, m - 1\}$ qu'on peut inférer $C'_i = C_1 \vee \bar{C}_{i+1}$ à la $i^{\text{ème}}$ étape de max-résolution :

- Pour $i = 1$, la première max-résolution sur les clauses $C_1 = l_1 \vee A_1$ et $D_1 = \bar{l}_1 \vee B_1$ et la variable $var(l_1)$ génère la clause de compensation suivante :

$$CC_{1|1} = l_1 \vee A_1 \vee \bar{B}_1 \stackrel{*}{=} l_1 \vee A_1 \vee \overline{A_1 \vee B_1} = C_1 \vee \bar{C}_2 = C'_1$$

Notons que pour établir l'égalité $\stackrel{*}{=}$, on peut ajouter les clauses tautologiques $l_1 \vee A_1 \vee B_1 \vee \bar{A}_1$ (ou alternativement $l_1 \vee A_1 \vee \bar{A}_1$) à la formule), auquel cas $l_1 \vee A_1 \vee \bar{A}_1 \vee \bar{B}_1$ peut être trivialement déduite par compactage. De plus, si D_1 est une clause unitaire, $CC_{1|1}$ n'est pas générée. Cependant, on peut simplement ajouter les clauses tautologiques $l_1 \vee A_1 \vee \bar{A}_1$ qui correspondent à $C_1 \vee \bar{C}_2$ puisque $D_1 = \bar{l}_1$ (B_1 est vide).

- Supposons qu'on peut générer $C'_{i-1} = C_1 \vee \bar{C}_i$ à la $i^{\text{ème}} - 1$ étape de max-résolution. La $i^{\text{ème}}$ étape sur $C_i = l_i \vee A_i$ et $D_i = \bar{l}_i \vee B_i$ et la variable $var(l_i)$ génère la résolvente $C_{i+1} = A_i \vee B_i$ et les clauses de compensation $CC_{1|i} = l_i \vee A_i \vee \bar{B}_i$ et $CC_{2|i} = \bar{l}_i \vee \bar{A}_i$ pour B . L'induction pour inférer C'_i est représentée dans la figure 2. Notons que, comme dans le cas de base, si $D_i = \bar{l}_i$ ($i > 1$) est une clause unitaire, c'est-à-dire que la $i^{\text{ème}}$ étape correspond à une suppression du littéral l_i de $C_i = l_i \vee A_i$ en déduisant la résolvente $C_{i+1} = A_i$, les clauses tautologiques $l_i \vee A_i \vee \bar{A}_i$ peuvent être ajoutées à la formule remplaçant ainsi $CC_{1|i}$ dans la figure 2. Cependant, comme le montre la même figure, l'ajout de telles clauses dans le cas où D_1 est unitaire peut être évité puisque l'étape d'expansion initiale sur C'_{i-1} suffit pour générer $C_1 \vee \bar{A}_i = C_1 \vee \bar{C}_{i+1} = C'_i$.

Enfin, d'après la proposition 2 (motif b), l'inférence de $C_1 \vee A_i \vee \bar{B}_i$ dans la figure 2 requiert $O(|B_i|)$ étapes de max-résolution et, par conséquent, chaque étape de π est clairement adaptée en $O(w(\pi))$ étapes d'inférence pour générer C et $C_1 \vee \bar{C}_m$. On conclut qu'on peut déduire $\phi \vdash_{MaxRes} C \wedge (C_1 \vee \bar{C}_m)$ en $O(s(\pi).w(\pi))$ étapes d'inférence.

Exemple 3. On considère la preuve linéaire *read-once* de la clause $\bar{x}_3 \vee x_6$ de $\phi = \{x_4, x_2 \vee x_4 \vee x_6, \bar{x}_2 \vee \bar{x}_3\}$ représentée à gauche de la figure 3. La preuve par max-résolution déduisant $\bar{x}_3 \vee x_6$ et $x_4 \vee \bar{x}_3 \vee x_6$ est représentée à droite de la figure 3.

Dans la suite, nous établissons notre résultat sur l'adaptation des preuve par résolution sans croisement. La preuve dans le théorème suivant traite en particulier les

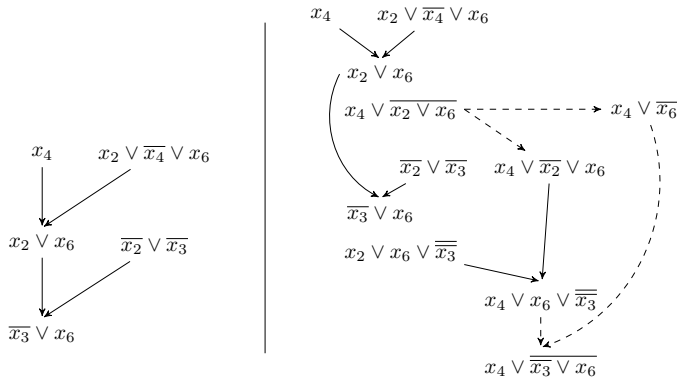


Figure 3: Maintien de la clause non *read-once* tout en dépliant une section linéaire *read-once* de la preuve. Les arcs continus représentent l'application de la max-résolution tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

nœuds de jonction dans les sous-dérivations des clauses non *read-once*, c'est-à-dire les nœuds où les chemins partant des clauses non *read-once* se croisent. Plus précisément, l'idée est de ramener la clause non *read-once* à travers ces nœuds particuliers. Une illustration d'une adaptation complète est donnée dans l'exemple 4.

Théorème 1. Soit ϕ une formule FNC et π une preuve par résolution sans croisement de la clause C à partir de ϕ . On peut déduire $\phi \vdash_{MaxRes} C$ en $O(s(\pi) \times (s(\pi) + w(\pi))^2)$ étapes d'inférence.

Preuve. La propriété 1 garantit que chaque sous-dérivation de clause non *read-once* puisse adaptée

indépendamment. Soit Cl une clause non *read-once* dans π et, sans perte de généralité, on considère la sous-dérivation $ED(Cl)$ d'une clause non *read-once* Cl . Dans la suite, on prouve qu'à chaque étape de $ED(Cl)$ C' , on peut inférer C' et $SC \vee \overline{C'}$ où SC est soit Cl soit son remplaçant dans le chemin menant à C' . La preuve se fait par induction sur la taille de la dérivation. Le cas de base où la dérivation est vide est trivial. Ensuite, en utilisant le lemme 1, on peut supposer sans perte de généralité que C' est dérivée dans un nœud de jonction (de chemins partant de Cl). Soient $l \vee A$ et $\bar{l} \vee B$ les prémisses de l'étape de résolution dérivant $C' = A \vee B$. L'hypothèse d'induction garantit qu'il existe une preuve par max-résolution de $l \vee A$ et $C \vee \bar{l} \vee \bar{A}$. Comme le montre la figure 4, $Cl \vee \bar{l}$ peut être utilisée pour remplacer les occurrences de Cl dans la dérivation de $\bar{l} \vee B$. Notons que pour éviter d'utiliser des remplaçants tautologiques, on peut supposer sans perte de généralité que $l \notin Cl$ en interchangeant les preuves de $l \vee A$ et $l \vee B$ lorsque cela est nécessaire imposant ainsi un ordre de déroulement de la preuve et entraînant la génération de la même clause comme substitut dans de tels nœuds. De manière similaire au côté gauche, l'hypothèse d'induction garantit aussi l'existence d'une preuve par max-résolution de $\bar{l} \vee B$ et $Cl \vee \bar{l} \vee \bar{l} \vee B$ et, par conséquent, $Cl \vee \bar{l} \vee \bar{B}$ par expansion. Clairement, $C' = A \vee B$ peut être dérivée par max-résolution et on montre dans la figure 4 comment $Cl \vee \overline{C'} = C \vee \overline{A \vee B}$ peut être déduite en utilisant les clauses de compensation ainsi que $Cl \vee \bar{l} \vee \bar{A}$ et $Cl \vee \bar{l} \vee \bar{B}$. Notons que les cas particuliers suivants peuvent se produire :

- A ou B est vide, auquel cas une clause unitaire est utilisée pour dériver $C' = A \vee B$. Nous représentons dans la figure 5 comment dériver $Cl \vee \overline{C'}$ dans le cas où A est vide. La dérivation dans le cas où B est vide

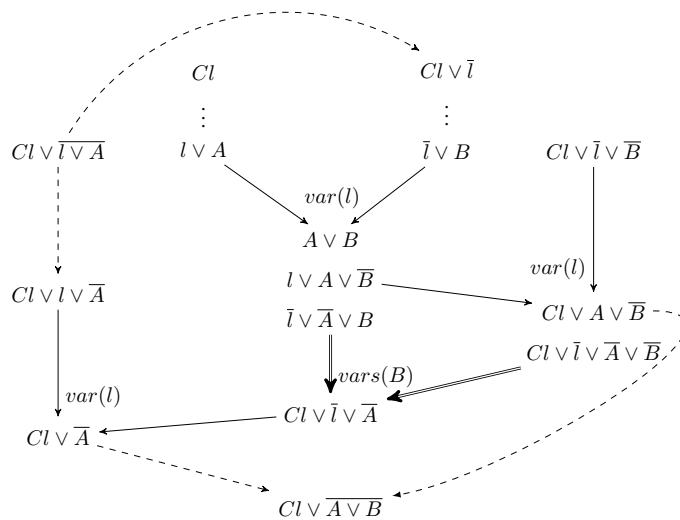


Figure 4: Inférence de $Cl \vee \overline{A \vee B}$ dans un nœud de jonction de $ED(Cl)$. Les arcs continus représentent l'application de la max-résolution, les doubles arcs continus en gras représentent l'application de la max-résolution pour supprimer des ensembles opposés de littéraux tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

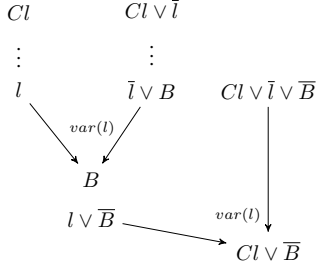


Figure 5: Inférence de $Cl \vee \overline{C'}$ dans le cas où A est vide dans un nœud de jonction de $ED(Cl)$. Les arcs continus représentent l'application de la max-résolution tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

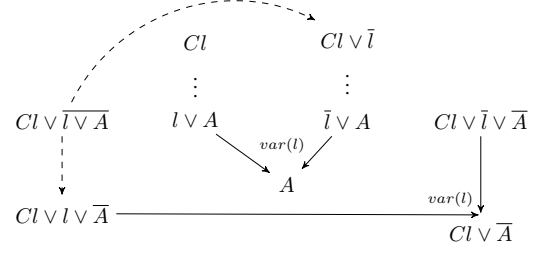


Figure 6: Inférence de $Cl \vee \overline{C'}$ dans le cas où $A = B$ dans un nœud de jonction de $ED(Cl)$. Les arcs continus représentent l'application de la max-résolution tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

est symétrique et donc omise. Notons que dans le cas où A et B sont vides, π est une réfutation et il n'est pas nécessaire de dériver $Cl \vee \overline{C'}$ dans la dernière étape de max-résolution. En effet, plus généralement, cela n'est pas non plus nécessaire pour le dernier nœud de jonction dans une sous-dérivation d'une clause non read-once de π .

- $A = B$, auquel cas les clauses de compensation générées sont tautologiques et ne sont pas nécessaires pour dériver $Cl \vee \overline{C'} = Cl \vee \overline{A}$ comme le montre la figure 6.

Enfin, $O(|B|)$ étapes d'inférence sont nécessaires pour déduire $Cl \vee A \vee \overline{B}$ dans chaque nœud de jonction en se référant au motif (c) dans la proposition 2. De même, en utilisant l'expansion sur A et le motif (b) dans la proposition 2, il suffit de $O(|A| \times |B|)$ étapes d'inférence pour déduire $Cl \vee \overline{l} \vee \overline{A}$. Il est important de noter

que la largeur de la preuve peut évoluer tout en générant des remplaçants pour les clauses non read-once car des littéraux peuvent être ajoutés dans les nœuds de jonction. Cependant, la largeur reste bornée par $w(\pi) + s(\pi)$ et donc chaque nœud de jonction peut être adapté en $O((w(\pi) + s(\pi))^2)$ étapes d'inférences. Par conséquent, on conclut qu'on peut déduire $\phi \vdash_{MaxRes} C$ en $O(s(\pi) \times (s(\pi) + w(\pi))^2)$ étapes d'inférence.

Exemple 4. On considère la formule $\phi = \{\overline{x}_1 \vee x_3 \vee \overline{x}_4, x_4 \vee x_5, x_4 \vee \overline{x}_5, x_1 \vee \overline{x}_4, x_2 \vee \overline{x}_4 \vee x_6, \overline{x}_2 \vee \overline{x}_3\}$ et la preuve π de la clause x_6 à partir de ϕ représentée dans la figure 1. Pour simplifier, on omet la section (en bleu) déduisant la clause $\overline{x}_2 \vee \overline{x}_3$. Notons que cette section de la preuve, c'est à dire la sous-dérivation de la clause non read-once $\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_7$, correspond à un motif diamant [23]. Ces motifs seront étudiés dans la section 5. L'adaptation de la preuve π est reportée dans la Figure 7. Nous réutilisons l'adaptation de la section

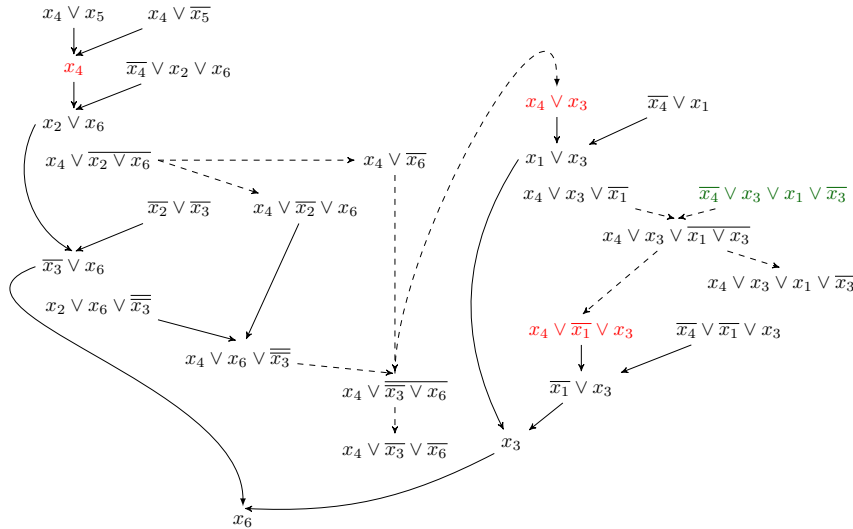


Figure 7: Adaptation d'une preuve par résolution sans croisement. Les arcs continus représentent l'application de la max-résolution tandis que les arcs discontinus représentent le compactage ou l'expansion. Les clauses de compensation non utilisées sont omises.

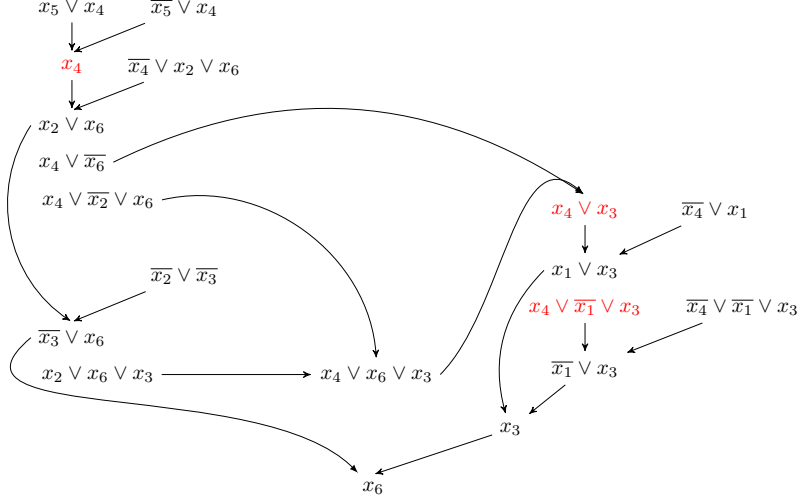


Figure 8: Adaptation d'une preuve par résolution en une preuve par max-résolution (exclusivement). Les clauses de compensation non utilisées sont omises.

linéaire read-once dans l'exemple 3. La clause non read-once et ses remplaçants sont colorés en rouge et les clauses tautologiques ajoutées sont représentées en vert. Notons qu'il s'agit d'une des adaptations possibles selon l'ordre choisi pour adapter les branches de $ED(x_4)$. Enfin, nous soulignons le fait que nous aurions pu générer la clause $C = x_4 \vee \bar{x}_6$ après la dernière étape de max-résolution sur les clauses $\bar{x}_3 \vee x_3$ (mais cette inférence n'est pas nécessaire puisque $x_6 = EC(x_4)$ comme expliqué dans la preuve du théorème 1). En effet, C peut être déduite par une étape supplémentaire de max-résolution sur les clauses de compensation obtenues à la dernière étape, c'est-à-dire les clauses $x_3 \vee \bar{x}_6$ et $x_4 \vee \bar{x}_3 \vee \bar{x}_6$. Dans la preuve du théorème 1, cela correspond au cas où B est vide dans un nœud de jonction d'une sous-dérivation.

Dans la suite, nous discutons de quelques subtilités syntaxiques mineures qui se produisent dans l'adaptation. Tout d'abord, il est important de noter que l'utilisation des réécritures d'expansion et de compactage en tant que règles à part entière est pertinente pour la simplification mais pas nécessaire. Rappelons que ces deux règles sont principalement utilisées pour alterner entre les différentes formes équivalentes de \bar{C} lorsqu'elle est écrit sous forme FNC. Chaque forme correspond à un ordre différent des littéraux de C . Lors de l'application de la max-résolution, un ordre pertinent peut être choisi si nécessaire. Cependant, une application d'un compactage suivi d'une expansion peut correspondre à un certain réarrangement des variables sous forme FNC. Cela peut se produire lors de l'adaptation de la partie linéaire read-once de la preuve. En effet, comme le montre la figure 2, un compactage peut être suivi d'une expansion pour isoler la clause $C_1 \vee \bar{l}_i \vee A_i$ de la forme compacte $C_1 \vee \bar{A}_i$. De même, comme le montre la figure 4, il peut être nécessaire d'isoler la clause $Cl \vee \bar{l}$ de la forme compacte $Cl \vee l \vee \bar{A}$ dans les nœuds de jonction. Plus précisément, il pourrait être nécessaire de réorganiser un certain littéral au début ou à la fin de l'ordre. Pour

cela, nous prouvons dans la proposition 3 qu'il est possible d'alterner le premier et le dernier littéral dans la forme FNC de \bar{C} en $O(|C|)$ étapes d'inférence. cela implique que, dans la preuve du théorème 1, les règles de compactage et d'expansion peuvent être omises et remplacées par $O(s(\pi) \times (s(\pi) + w(\pi)))$ max-résolutions. Cela n'a évidemment pas d'impact sur notre résultat en termes de taille de l'adaptation résultante. Dans l'exemple 5, nous fournissons une adaptation entièrement simplifiée de la preuve de l'exemple 4 sans l'utilisation de règles de réécriture.

Proposition 3. Soit n un entier naturel et l_1, \dots, l_n n littéraux. On peut déduire $(\bar{l}_1) \wedge (l_1 \vee \bar{l}_2) \wedge \dots \wedge (l_1 \vee \dots \vee l_{n-1} \vee \bar{l}_n) \vdash_{MaxRes} (\bar{l}_n) \wedge (l_n \vee \bar{l}_2) \wedge \dots \wedge (l_n \vee l_2 \vee \dots \vee l_{n-1} \vee \bar{l}_1)$ en $O(n)$ étapes d'inférence.

Preuve. By induction on n we have:

- Pour $n = 1$, le résultat est trivial.
- Pour $n > 1$, l'application de la max-résolution sur les clauses $l_1 \vee \dots \vee l_{n-2} \vee \bar{l}_{n-1}$ et $l_1 \vee \dots \vee l_{n-1} \vee \bar{l}_n$ w.r.t $var(l_{n-1})$ génère la clause résolvente $C = l_1 \vee \dots \vee l_{n-2} \vee \bar{l}_n$ et la clause de compensation $CC = l_1 \vee \dots \vee l_{n-2} \vee l_n \vee \bar{l}_{n-1}$. De plus, par récurrence, on peut déduire $(\bar{l}_1) \wedge (l_1 \vee \bar{l}_2) \wedge \dots \wedge (l_1 \vee \dots \vee l_{n-2} \vee \bar{l}_n) \vdash_{MaxRes} (\bar{l}_n) \wedge (l_n \vee \bar{l}_2) \wedge \dots \wedge (l_n \vee l_2 \vee \dots \vee l_{n-2} \vee \bar{l}_1)$ en $O(n-1)$ étapes d'inférence. Une seule étape de max-résolution supplémentaire sur les clauses CC et $l_n \vee l_2 \vee \dots \vee l_{n-2} \vee \bar{l}_1$ w.r.t $var(l_1)$ est suffisante pour générer la clause résolvente $l_n \vee l_2 \vee \dots \vee l_{n-2} \vee \bar{l}_{n-1}$ et la clause de compensation $l_n \vee l_2 \vee \dots \vee l_{n-1} \vee \bar{l}_1$. Par conséquent, nous déduisons le résultat voulu en $O(n)$ étapes d'inférence.

Exemple 5. On considère la même formule ϕ dans l'exemple 4. Nous représentons sur la figure 8 une preuve par max-résolution (sans réécriture) de la clause x_6 à partir de ϕ . Notons que le réarrangement suivant $\bar{x}_6 \wedge (x_6 \vee$

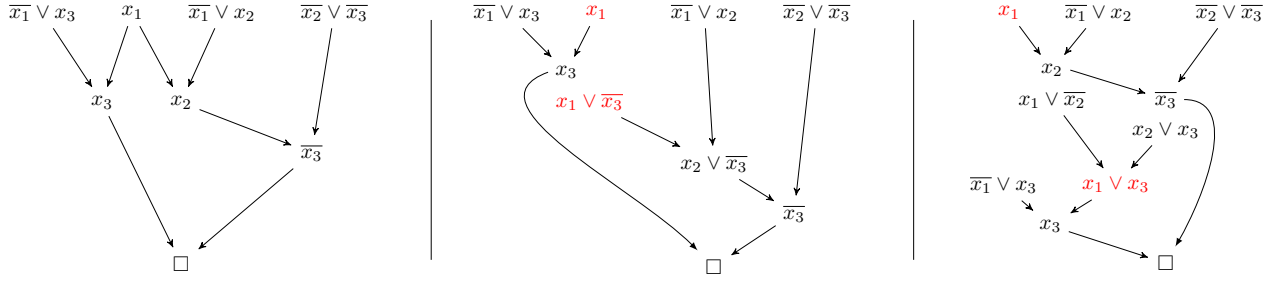


Figure 9: Deux adaptations possibles de la réfutation par résolution sans croisement représentée à gauche en fonction de l'ordre des étapes de résolution impliquant la clause non read-once x_1 . Les clauses de compensation non utilisées sont omises.

$x_3 \vdash_{MaxRes} x_3 \wedge (x_3 \vee x_6)$ a été utilisé pour générer le remplaçant $x_4 \vee x_3$. De plus, la clause tautologique $\overline{x_4} \vee x_3 \vee x_1 \vee \overline{x_3}$ colorée en vert dans la figure 7 et le réarrangement dans lequel elle est impliquée ne sont pas nécessaires puisque le dernier remplaçant requis pour x_1 , c'est-à-dire $x_4 \vee \overline{x_1} \vee x_3$, est naturellement généré par l'étape de max-résolution précédente. Par conséquent, ils peuvent être supprimés comme c'est le cas pour l'adaptation complète sans réécriture dans la figure 8.

Dans la suite, nous discutons des implications de l'égalité $\stackrel{*}{=}$ utilisée dans la preuve du lemme 1, c'est-à-dire $l \vee A \vee \overline{B} \stackrel{*}{=} l \vee A \vee \overline{A} \vee \overline{B}$. Cette transformation est valide pour Max-SAT (c.f. Remarque 13 dans [16]). Cependant, pour éviter de l'ajouter comme règle à part entière et comme expliqué dans la preuve du lemme 1, on peut envisager l'ajout de clauses tautologiques. Cela peut également être nécessaire en cas de clauses unitaires. Il est important de noter que le nombre de clauses tautologiques ajoutées à la formule dans une adaptation d'une preuve par résolution sans croisement π est en $O(s(\pi) \times (w(\pi) + s(\pi)))$. Un phénomène similaire a également été noté dans [8]. De plus, on remarque que l'adaptation peut également s'appuyer sur des clauses de compensation tautologiques qui sont générées par la max-résolution. De telles clauses sont généralement supprimées ou omises dans la littérature [6, 7, 16] mais elles peuvent contenir des informations importantes qui sont nécessaires pour déduire des substituts aux clauses non read-once. Enfin, on établit notre résultat sur les réfutations sans croisement dans le corollaire suivant. Nous illustrons également dans l'exemple 6 une adaptation d'une réfutation par résolution sans de croisement à une réfutation par max-résolution.

Corollaire 1. Soit ϕ une formule FNC insatisfiable et π une réfutation par résolution sans croisement de ϕ . On peut déduire $\phi \vdash_{MaxRes} \square$ de ϕ en $O(s(\pi)^3)$ étapes d'inférence.

Preuve. Résultat trivialement déduit du théorème 1 puisque $w(\pi) = O(s(\pi))$ pour les réfutations.

Exemple 6. On considère la formule FNC insatisfiable $\phi = \{x_1, \overline{x_1} \vee x_3, \overline{x_1} \vee x_2, \overline{x_2} \vee \overline{x_3}\}$ et la réfutation π de ϕ représentée à gauche dans la Figure 9. Clairement,

π est sans croisement car elle ne contient qu'une seule clause non read-once x_1 . π correspond également à la sous-dérivation de cette clause, c'est-à-dire $ED(x_1) = \pi$ avec $EC(x_1) = \square$. Deux adaptations possibles de π sont illustrées dans la figure 9. La clause non read-once et ses remplaçants sont colorés en rouge. Les adaptations possibles correspondent à différents ordre possibles de la preuve. Dans la première adaptation, on considère que l'étape de résolution sur les clauses $\overline{x_1} \vee x_1$ et x_1 précède celle sur les clauses x_1 et $\overline{x_1} \vee x_2$, et inversement pour l'adaptation à droite. Notons que la première adaptation correspond à l'exemple fourni par Bonet et al. dans [6, 7] (c.f. Exemple 1 dans [6] ou Exemple 3 dans [7]).

5 Motifs diamants

Dans cette section, nous étudions des réfutations de résolution particulières, appelées motifs diamants k-empilés, qui ont été introduites et montrées exponentielle pour l'adaptation (à des réfutations ResS) dans [23]. Un motif diamant (x, y, A) où $x, y \notin A$ est la séquence de résolutions représentée sur la figure 10. Notons que le motif diamant particulier (x, y, \square) est une réfutation par résolution. Maintenant, imaginons que la clause x de (x, y, \square) est dérivée d'un autre motif diamant. Nous itérons le même raisonnement pour définir les motifs diamants k-empilés comme dans la définition 6.

Définition 6 (Motif diamants k-empilés). Soit $k \geq 1$ un entier naturel et soit x_i et y_i où $1 \leq i \leq k$ des variables distinctes. Un motif diamants k-empilé est formé de k diamants (x_i, y_i, A_i) où $1 \leq i \leq k$ tels que $A_1 = \square$ et $A_i = (x_1 \vee \dots \vee x_{i-1})$ pour $1 < i \leq k$. Chaque diamant (x_i, y_i, A_i) est empilé sur $(x_{i-1}, y_{i-1}, A_{i-1})$ de sorte que la dernière conclusion du premier est la prémisse centrale de ce dernier.

Lorsque $k > 2$, la taille d'un motif diamants k-empilés P

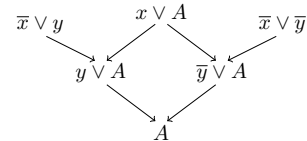


Figure 10: Motif Diamant (x, y, A)

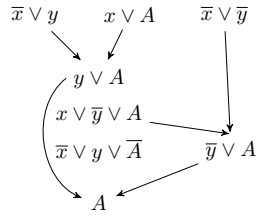


Figure 11: Adaptation d'un diamant (x, y, A)

est $s(P) = 3k$ tandis que la taille de la réfutation calculée dans ResS [17], c'est-à-dire la max-résolution augmentée avec la règle *split*, par l'adaptation dans [23] est au moins 2^{k-1} ce qui est exponentiel en la taille de P . Dans la figure 11, nous montrons comment adapter chaque motif diamant sans augmenter sa taille. Ceci est impliqué par le fait que chaque diamant est clairement une preuve par résolution sans croisement et plus spécifiquement une sous-dérivation d'une clause non *read-once*. Ainsi, un diamant et plus généralement un motif diamants k -empilés P peut être adapté en au plus $s(P)$ étapes de max-résolution.

6 Conclusion

Dans ce papier, nous avons introduit un nouveau fragment de résolution, appelé résolution sans croisement, dans lequel les sous-dérivations des clauses non *read-once* sont disjointes. Nous avons montré que les preuves par résolution sans croisement et en particulier les réfutations sans croisement peuvent être adaptées en preuves par max-résolution sans augmenter considérablement leur taille. À notre connaissance, il s'agit du premier fragment non trivial, c'est-à-dire différent de la résolution *read-once*, dont l'adaptation est montrée possible en utilisant uniquement la max-résolution avec une garantie raisonnable sur la taille des preuves adaptées. L'idée derrière l'adaptation est d'inférer naturellement des remplaçants aux clauses non *read-once* en les ramenant tout en déroulant la preuve par résolution initiale jusqu'à leur réutilisation et en s'appuyant sur les clauses de compensation produites par la max-résolution. De plus, nous montrons que les motifs diamants (k -empilés), qui ont été montrés exponentiels pour l'adaptation dans [23], appartiennent au ce nouveau fragment et peuvent être adaptés en preuves par max-résolution sans augmenter leur taille.

Nos résultats contribuent au problème ouvert d'adapter les preuves par résolution en preuves par max-résolution sans augmenter considérablement leur taille [6, 7] et, par conséquent, aident à combler l'écart entre la résolution pour SAT et Max-SAT. Enfin, contrairement aux solveurs SAT, les solveurs Max-SAT ne sont toujours pas en mesure de produire des certificats pour Max-SAT, principalement en raison de la variété des paradigmes de résolution et du manque de caractérisation d'une intersection plus large entre la résolution SAT et Max-SAT. Notre travail peut être utile dans ce contexte et en particulier pour améliorer l'efficacité des constructeurs de preuves indépendants pour le problème Max-SAT [24].

References

- [1] André Abramé and Djamel Habet. Ahmaxsat: Description and Evaluation of a Branch and Bound Max-SAT Solver. *JSAT*, 9(1):89–128, 2014.
- [2] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. SAT-based MaxSAT algorithms. *Artif. Intell.*, 196:77–105, 2013.
- [3] F. Bacchus, J. Berg, M. Jarvisalo, and R. Martins, editors. *MaxSAT Evaluation 2021: Solver and Benchmark Descriptions*. University of Helsinki, Department of Computer Science, Finland, 2021.
- [4] Fahiem Bacchus, Matti Jarvisalo, and Ruben Martins. *Maximum Satisfiability*, pages 929 – 991. FAIA. IOS PRESS, Netherlands, 2 edition, 2021.
- [5] Maria Luisa Bonet and Jordi Levy. Equivalence Between Systems Stronger Than Resolution. In *SAT*, 2020.
- [6] Maria Luisa Bonet, Jordi Levy, and Felip Manyà. A Complete Calculus for Max-SAT. In Armin Biere and Carla P. Gomes, editors, *SAT*, pages 240–251, 2006.
- [7] Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for Max-SAT. *Artif. Intell.*, 171(8-9):606–618, 2007.
- [8] Joshua Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. *J. Symb. Log.*, 72(4):1336–1352, 2007.
- [9] Sam Buss and Jan Johannsen. On Linear Resolution. *JSAT*, 10(1):23–35, 2016.
- [10] Mohamed Sami Cherif, Djamel Habet, and André Abramé. Understanding the power of Max-SAT resolution through UP-resilience. *Artif. Intell.*, 289:103397, 2020.
- [11] Yuval Filmus, Meena Mahajan, Gaurav Sood, and Marc Vinyals. MaxSAT Resolution and Subcube Sums. In *SAT*, volume 12178, pages 295–311. Springer, 2020.
- [12] Federico Heras and João Marques-Silva. Read-Once Resolution for Unsatisfiability-Based Max-SAT Algorithms. In *IJCAI*, pages 572–577, 2011.
- [13] Kazuo Iwama and Eiji Miyano. Intractability of Read-Once Resolution. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 29–36, 1995.
- [14] Adrian Kügel. Improved Exact Solver for the Weighted MAX-SAT Problem. In *POS-10. Pragmatics of SAT*, volume 8 of *EPiC Series in Computing*, pages 15–27, 2010.
- [15] Sukhamay Kundu. Tree resolution and generalized semantic tree. In *Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems*, pages 270–278, 1986.
- [16] Javier Larrosa, Federico Heras, and Simon de Givry. A logical approach to efficient Max-SAT solving. *Artif. Intell.*, 172(2-3):204–233, 2008.
- [17] Javier Larrosa and Emma Rollon. Towards a Better Understanding of (Partial Weighted) MaxSAT Proof Systems. In *SAT*, volume 12178, pages 218–232, 2020.
- [18] Chu Min Li, Felip Manyà, and Jordi Planes. New Inference Rules for Max-SAT. *JAIR*, 30:321–359, 2007.
- [19] Chu Min Li, Felip Manyà, and Joan Ramon Soler. A Clause Tableau Calculus for MaxSAT. In *IJCAI*, pages 766–772, 2016.
- [20] Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamel Habet, and Kun He. Combining Clause Learning and Branch and Bound for MaxSAT. In *CP*, pages 38:1–38:18, 2021.
- [21] Donald W Loveland. A linear format for resolution. *Symposium on Automatic Demonstration*, pages 147–162, 01 1970.
- [22] Nina Narodytska and Fahiem Bacchus. Maximum Satisfiability Using Core-Guided MaxSAT Resolution. In *AAAI*, pages 2717–2723, 2014.
- [23] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. Towards Bridging the Gap Between SAT and Max-SAT Refutations. In *ICTAI*, pages 137–144. IEEE, 2020.
- [24] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. A Proof Builder for Max-SAT. In Chu-Min Li and Felip Manyà, editors, *SAT*, volume 12831 of *Lecture Notes in Computer Science*, pages 488–498. Springer, 2021.
- [25] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. Computing Max-SAT Refutations using SAT Oracles. In *ICTAI*, pages 404–411. IEEE, 2021.
- [26] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. Inferring Clauses and Formulas in Max-SAT. In *ICTAI*, pages 632–639. IEEE, 2021.
- [27] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *J. ACM*, 12(1):23–41, 1965.