Preuves par résolution

Vous avez vu en cours le système de preuve de Hilbert-Frege. Robinson [3] a proposé un système consistant en une unique règle, la *résolution*.

Cet exercice se place en logique propositionnelle. Nous rappelons qu'un *littéral* est une variable propositionnelle ou sa négation. Nous appelons *clause* une disjonction $L_1 \vee \cdots \vee L_n$ de littéraux deux-à-deux distincts, et sans qu'on n'y trouve à la fois une variable et sa négation. Nous identifierons une clause à l'ensemble des littéraux qui y sont présents. La clause vide $\bot$, disjonction de zéro littéraux, est donc la formule « faux ». On rappelle que la notation $\mathcal{M} \models C$ où $\mathcal{M}$ est une valuation et C est une clause signifie que $\mathcal{M}$ *satisfait* C, et que $\mathcal{M} \models F$ où F est un ensemble de clauses si et seulement si $\mathcal{M} \models C$ pour toute clause $C \in F$.

La méthode de résolution fonctionne à partir d'une conjonction de clauses, autrement dit d'une formule en *forme normale conjonctive*. Remarquons bien la différence en un ensemble vide de clauses (qui, considéré comme une conjonction, est la formule « vrai ») et une clause vide $\bot$.

1 Preuve par résolution

La *règle de résolution* dit que si l'on a $C_1 \vee a$ et $C_2 \vee \neg a$, où a est une variable propositionnelle et C_1 et C_2 sont deux clauses, alors on peut en déduire $C_1 \vee C_2$. On note :

$$\frac{C_1 \vee a \quad C_2 \vee \neg a}{C_1 \vee C_2}\ a$$

Le symbole a à droite de la barre horizontale indique la variable vis-à-vis de laquelle la règle est appliquée. Une *preuve par résolution* d'une clause C, appelée *conclusion*, à partir d'un ensemble de clauses $\mathcal{H}$, appelées *hypothèses*, est un arbre formé d'applications (correctement formées, bien sûr) de la règle de résolution. Par exemple la preuve que l'ensemble de clauses $\mathcal{H} = \{a \vee b, \neg a, \neg b\}$ n'est pas satisfiable peut s'écrire

$$\dfrac{\dfrac{a \vee b \quad \neg a}{b}\ a \qquad \neg b}{\bot}\ b$$

On s'intéresse en particulier aux preuves de l'absurde comme ci-dessus, c'est-à-dire que la conclusion est la clause vide $\bot$.

Question 1.1. *Prouver par résolution que l'ensemble de clauses* $\{a \vee \neg c, a \vee b \vee c, \neg a, a \vee \neg b\}$ *n'est pas satisfiable.*

Question 1.2. *Montrer que la règle de résolution est correcte, autrement dit que si* $\mathcal{M} \models C_1 \vee a$ *et* $\mathcal{M} \models C_2 \vee \neg a$*, alors* $\mathcal{M} \models C_1 \vee C_2$*.*

2 Forme normale conjonctive

L'algorithme de preuve par résolution prend en entrée une formule en forme normale conjonctive. Il est toujours possible de transformer une formule quelconque en une formule en forme normale conjonctive, même si le coût est potentiellement exponentiel si fait naïvement, et la plupart des outils de SAT-solving (recherche de solution de formule propositionnelle), d'une grande importance industrielle, partent d'une forme normale conjonctive.

Question 2.1. *Mettre* $(a \wedge b \wedge c) \vee \neg(a \vee b)$ *en forme normale conjonctive.*

Il est possible de transformer une formule F en une formule equisatisfiable F' en forme normale conjonctive de taille linéaire en celle de F, à condition de rajouter des variables. La méthode la plus courante est l'encodage de Tseitin [4, 1], qui consiste à ajouter une variable pour chaque sous-formule de F, ainsi que des clauses qui capturent les relations entre les sous-formules. Ici, nous considérons cette transformation pour des formules propositionnelles contenant des conjonctions, des disjonctions, et des négations, mais l'approche s'étend facilement à n'importe quel système des connecteurs booléens.

Question 2.2. *Donner un ensemble des clauses $F_\wedge^{a,b,c}$ en trois variables a, b, c tel que $\mathcal{M} \models F_\wedge^{a,b,c}$ si et seulement si $\mathcal{M}(c) = \mathcal{M}(a) \wedge \mathcal{M}(b)$ (où ici on écrit $\wedge$ pour l'opération $\wedge : \{0,1\}^2 \to \{0,1\}$ de conjonction des valeurs booléennes). De la même manière, donner un ensemble des clauses $F_\vee^{a,b,c}$ tel que $\mathcal{M} \models F_\vee^{a,b,c}$ ssi $\mathcal{M}(c) = \mathcal{M}(a) \vee \mathcal{M}(b)$. Enfin, donner un ensemble des clauses $F_\neg^{a,b}$ en deux variables a et b tel que $\mathcal{M} \models F_\neg^{a,b}$ ssi $\mathcal{M}(b) = 1 - \mathcal{M}(a)$.*

Question 2.3. *Expliquer comment transformer une formule F avec les connecteurs $\wedge$, $\vee$, et $\neg$ en une formule F' en forme normale conjonctive de taille linéaire en celle de F, tel que les valuations satisfaisantes de F sont en correspondence biunivoque avec les valuations satisfaisantes de F'.*

3 Algorithme de preuve

Soit F un ensemble de clauses sur un ensemble de variables $a, a_1, \ldots, a_n$. Nous définissons l'ensemble *résol*(F, a) comme l'ensemble de clauses contenant exactement :

- les clauses de F ne contenant ni a ni $\neg a$,
- les clauses de la forme $C_1 \vee C_2$, obtenues par application de la règle de résolution, à partir de deux clauses $C_1 \vee a$ et $C_2 \vee \neg a$ appartenant à F, à condition que $C_1 \vee C_2$ ne contienne pas deux littéraux b et $\neg b$ opposés.

Question 3.1. *Calculer résol(F, a) sur l'exemple de la question 1.1 ainsi que sur l'ensemble $F = \{b \vee c, a \vee b \vee c, \neg a \vee c \vee d, \neg a \vee \neg b \vee e\}$.*

Étant donnés une valuation $\mathcal{M}$, une variable a et un booléen $v \in \{0,1\}$, on note $(\mathcal{M}, a \mapsto v)$ la valuation telle que $(\mathcal{M}, a \mapsto v)(a) = v$ et $(\mathcal{M}, a \mapsto v)(b) = \mathcal{M}(b)$ pour $b \neq a$.

Question 3.2. *On note $\mathcal{M}$ une valuation pour les variables $a_1, \ldots, a_n$. Montrer que*

$$\mathcal{M} \models \textit{résol}(F, a)$$

si et seulement si il existe une valeur $v \in \{0,1\}$ telle que $(\mathcal{M}, a \mapsto v) \models F$.

Question 3.3. *Montrer que la règle de résolution est complète pour la réfutation, autrement dit que pour tout ensemble contradictoire de clauses F, il existe une preuve de l'absurde (la clause vide $\perp$) à partir des clauses de F n'utilisant que la règle de résolution.*

Question 3.4. *Montrer que la règle de résolution* n'est pas *complète pour la déduction, c'est-à-dire qu'il existe un ensemble Γ d'hypothèses et une conclusion C telles que $\Gamma \vDash C$ (autrement dit, $\Gamma \Rightarrow C$ est une tautologie), mais C ne s'obtient pas en conclusion de résolution.*

Question 3.5. *Proposer un algorithme simple qui, étant donné un ensemble fini Γ d'hypothèses et une conclusion C, décide si $\Gamma \vDash C$, en utilisant la règle de résolution.*

4 Conclusion

En pratique, la méthode que nous avons vue est très inefficace, mais on peut l'améliorer au point d'obtenir les algorithmes à l'état de l'art, utilisés industriellement. On ne recherche pas la preuve par résolution par la méthode brutale consistant à tout dériver. L'algorithme DPLL (Davis - Putnam - Logemann - Loveland), et sa variante moderne CDCL (*conflict-driven clause learning*) peut être vu comme une construction astucieuse de la preuve par résolution. On se rapportera pour plus de renseignements par exemple à Biere et al. [1] ou Knuth [2].

Références

[1] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, 2009. ISBN 978-1-58603-929-5.

[2] Donald Ervin Knuth. *The Art of Computer Programming, Volume 4 Fascicle 6 : Satisfiability*. Addison-Wesley, 2015. ISBN 978-0-13-439760-3.

[3] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12 :23–41, January 1965. ISSN 0004-5411. doi: 10.1145/321250.321253.

[4] Grigorii Samuilovich Tseitin. On the complexity of derivation in propositional calculus. In Anatol Oles'evich Slisenko, editor, *Studies in constructive mathematics and mathematical logic, part II*, volume 8. Consultants Bureau, 1970. URL `http://www.decision-procedures.org/handouts/Tseitin70.pdf`.