École polytechnique — X2022 — INF412 Fondements de l'informatique

Preuves par résolution

(corrigé)

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.*

Solution :

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

□

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 $\mathcal{M} \models C_1 \vee C_2$.*

Solution : Notons c_1 et c_2 les valeurs de vérité de C_1 et C_2 dans le modèle $\mathcal{M}$. Distinguons les cas $\mathcal{M} \models a$ et $\mathcal{M} \models \neg a$. Dans le premier cas, $C_2 \vee \neg a$ vaut c_2 ; comme $\mathcal{M} \models C_2 \vee \neg a$ on en déduit que $c_2 = 1$. Mais alors, $\mathcal{M} \models C_1 \vee C_2$. Un raisonnement similaire conclut pour l'autre cas. □

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.*

Solution : Une manière de trouver une forme normale conjonctive est d'appliquer une suite des équivalences :

$$\begin{aligned}(a \wedge b \wedge c) \vee \neg(a \vee b) &\equiv (a \wedge b \wedge c) \vee (\neg a \wedge \neg b) && \text{(loi de Morgan)}\\ &\equiv (a \vee (\neg a \wedge \neg b)) \wedge (b \vee (\neg a \wedge \neg b)) \wedge (c \vee (\neg a \wedge \neg b)) && \text{(distributivité)}\\ &\equiv (a \vee \neg a) \wedge (a \vee \neg b) \wedge (b \vee \neg a) \wedge (b \vee \neg b) \wedge (c \vee \neg a) \wedge (c \vee \neg b) \\ & && \text{(distributivité et associativité)}\end{aligned}$$

Cette formule est déjà en forme normale conjonctive, mais on peut obtenir une formule plus compacte en éliminant toutes les clauses où une variable apparait avec sa négation :

$$(a \vee \neg b) \wedge (b \vee \neg a) \wedge (c \vee \neg a) \wedge (c \vee \neg b) \tag{1}$$

Une autre façon de procéder est de construire d'abord la table de vérité :

a	b	c	$(a \wedge b \wedge c) \vee \neg(a \vee b)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Pour satisfaire la formule, une valuation doit être différente de toutes les valuations qui la rendent fausse, dont il y en a cinq dans ce cas. La propriété « différente de toutes les valuations falsifiante » s'exprime par la formule

$$(\neg a \vee b \vee \neg c) \wedge (\neg a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c) \wedge (a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee c) \tag{2}$$

qui est donc une autre forme normale conjonctive pour la formule originale (qui peut encore être simplifiée). □

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* $\mathcal{M}(b) = 1 - \mathcal{M}(a)$*.*

Solution : Ces conditions respectives peuvent être décrites par les formules

$$\begin{aligned} c &\iff a \wedge b \\ c &\iff a \vee b \\ b &\iff \neg a \end{aligned}$$

qui ont les formes normales conjonctives suivantes :

$$\begin{aligned} F_{\wedge}^{a,b,c} &:= \{a \vee \neg c, b \vee \neg c, \neg a \vee \neg b \vee c\} \\ F_{\vee}^{a,b,c} &:= \{\neg a \vee c, \neg b \vee c, a \vee b \vee \neg c\} \\ F_{\neg}^{a,b} &:= \{a \vee b, \neg a \vee \neg b\} \end{aligned}$$

□

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'.*

Solution : Soit $G_1, \ldots, G_n$ une énumération des sous-formules de F, avec $F = G_n$. En ajoutant une variable pour chaque sous-formule non-atomique, on peut associer une variable unique x_i à chaque sous-formule G_i de F. (Ici, le fait de distinguer ou non différentes occurrences de la même sous-formule n'a pas d'importance.) Puis, on construit l'ensemble des clauses suivant (en veillant à ne pas oublier la dernière clause unaire !) :

$$\begin{aligned} F' := &\{F_{\wedge}^{x_i,x_j,x_k} \mid G_k = G_i \wedge G_j \text{ est une sous-formule de } F\} \\ &\cup \{F_{\vee}^{x_i,x_j,x_k} \mid G_k = G_i \wedge G_j \text{ est une sous-formule de } F\} \\ &\cup \{F_{\neg}^{x_i,x_j} \mid G_j = \neg G_i \text{ est une sous-formule de } F\} \\ &\cup \{x_n\} \end{aligned}$$

Par les propriétés des clauses construites dans la question précédente, tout modèle de F se prolonge en un modèle de F' en prenant les valeurs appropriés pour les variables supplémentaires, et tout modèle de F' donne un modèle de F en supprimant les valeurs de ces variables supplémentaires, qui sont déterminées de manière unique. On conclut que F est satisfiable si est seulement si F' est satisfiable, avec une correspondance biunivoque entre leurs valuations satisfaisantes. De plus, F' est de taille linéaire en celle de F, puisque F' contient un nombre constant de clauses de taille constante pour chaque sous-formule 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 $F = \{b \vee c, a \vee b \vee c, \neg a \vee c \vee d, \neg a \vee \neg b \vee e\}$.*

Solution : Toutes les clauses contiennent a ou $\neg a$, donc seule la seconde règle s'applique, et on obtient

$$\textit{résol}(F, a) = \{\neg c, b \vee c, \neg b\}.$$

Pour le second exemple, on a

$$\textit{résol}(F, a) = \{b \vee c, b \vee c \vee d\}.$$

□

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 $(\mathcal{M}, a \mapsto v) \models F$.

Solution : $\Leftarrow$. Si $(\mathcal{M}, a \mapsto v) \models F$ alors $\mathcal{M} \models \textit{résol}(F, a)$, c'est une conséquence de la correction de la règle de résolution.

$\Rightarrow$. Si $\mathcal{M} \models \textit{résol}(F, a)$, on veut montrer qu'il existe v tel que $(\mathcal{M}, a \mapsto v) \models F$. Si F ne contient aucune clause $C \vee a$ telle que $\mathcal{M} \not\models C$, alors le choix $v = 0$ convient. Sinon, on va montrer que le choix $v = 1$ convient. En effet, soit C_1 une telle clause. Pour toute clause $C_2 \vee \neg a$ de F telle que $C_1 \vee C_2 \in \textit{résol}(F, a)$, $\mathcal{M} \models C_1 \vee C_2$ et comme par hypothèse $\mathcal{M} \not\models C_1$, on en déduit que $\mathcal{M} \models C_2$ et donc $(\mathcal{M}, a \mapsto v) \models C_2 \vee \neg a$. Reste le cas où $C_2 \vee \neg a$ est dans F mais $C_1 \vee C_2 \notin \textit{résol}(F, a)$ parce que $C_1 \vee C_2$ contient un $b \vee \neg b$, mais alors à nouveau, puisque $\mathcal{M} \not\models C_1$, la valeur de b est fixée par cette condition et permet à C_2 d'être satisfaite. Ainsi, $(\mathcal{M}, a \mapsto v)$ satisfait toutes les clauses de 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.*

Solution : Notons $a_1, \ldots, a_n$ les variables propositionnelles intervenant dans F et considérons $F_0 = F$, $F_i = \textit{résol}(F_{i-1}, a_i)$ pour $1 \leq i \leq n$. D'après la question précédente, par récurrence, $\mathcal{M} \models F_i$ (pour $\mathcal{M}$ valuation de $a_{i+1}, \ldots, a_n$) si et seulement si il existe des valeurs $v_1, \ldots, v_i$ telles quel $(a_1 \mapsto v_1, \ldots, a_i \mapsto v_i, \mathcal{M}) \models F$. C'est notamment le cas pour $i = n$, avec la formule F_i qui n'a pas de variables, et donc $\models F_n$ si et seulement si F est satisfiable. F_n est une conjonction de clauses à 0 variables : soit cet ensemble est vide, et F_n est vraie, soit il contient une unique clause, la clause vide $\perp$, et F_n est fausse.

Ainsi, si F n'est pas satisfiable, F_n contient la clause vide, mais cette clause vide est obtenue par application de la règle de résolution à partir de F_{n-1}, qui est obtenu par application etc. jusqu'à $F_0 = F$. Donc, si F n'est pas satisfiable, il existe un arbre de dérivation de la clause vide à partir 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.*

Solution : Prendre $\Gamma = \{a\}$ et $C = (a \vee b)$. □

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.*

Solution : Vu la question précédente, il ne suffit pas de faire « tourner » la résolution à partir de Γ et regarder si on obtient C.

Il suffit de décider si $\bigwedge_{H\in\Gamma} H \wedge \neg C$ est contradictoire (auquel cas $\Gamma \vDash C$) ou non (auquel cas $\Gamma \nvDash C$). On met en forme normale conjonctive et on obtient un ensemble fini de clauses contradictoire si et seulement si cette formule l'est. Il suffit ensuite d'appliquer la méthode de résolution et de regarder si on dérive la clause vide. □

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`.