Complétude, décidabilité, compacité du calcul propositionnel (résolution)

HLIN602 Logique II 2019-2020 Christian Retoré

mercredi 29 janvier 2020

1. Résolution — principe

Le principe de la résolution est l'équivalence suivante :

$[1] \quad (A \vee p) \wedge (B \vee (\neg p))$

$\equiv (A \vee p) \wedge (B \vee (\neg p)) \wedge (A \vee B) \quad [2]$

$[1]$ entraîne $[2]$.

Il suffit de montrer que

$F = ((A \vee p) \wedge (B \vee (\neg p)) \Rightarrow (A \vee B))$ est vrai.

$F[\top/p] \equiv (((A \vee \top) \wedge (B \vee \bot)) \Rightarrow (A \vee B))$

$\equiv (B \Rightarrow (A \vee B)) \equiv \top$

$F[\bot/p] \equiv (((A \vee \bot) \wedge (B \vee \top)) \Rightarrow (A \vee B))$

$\equiv (A \Rightarrow (A \vee B)) \equiv \top$.

Donc $(F \Rightarrow (A \vee B)) \equiv \top$.

$[2]$ entraîne $[1]$ est une évidence.

2. Résolution — méthode

Pour voir si un ensemble de clauses est contradictoire, on ajoute à l'ensemble de clauses de nouvelles clauses en appliquant le principe ci-dessus : si une clause C_1 contient p et une autre C_2 contient $(\neg p)$, on ajoute la clause C appelée *résolvant* et dont les termes sont ceux de C_1 moins p et ceux de C_2 moins $(\neg p)$.
Par exemple le résolvant de
$C_1 = (p \vee q \vee s \vee (\neg t))$ et $C_2 = (u \vee (\neg q) \vee v \vee (\neg t))$
est $C = (p \vee s \vee (\neg t) \vee u \vee v)$
— comme $(X \vee X) \equiv X$ il est inutile de répéter les littéraux communs.

D’après ce qui précède l’ensemble de clauses avant l’ajout d’un résolvant est équivalent à l’ensemble de clauses après ajout d’un résolvant.
S’il on obtient un résolvant vide à partir des clauses p et $(\neg p)$ on ajoute le résolvant $\perp$, et l’ensemble de clauses est clairement contradictoire.

3. Résolution — ex. 1

Soit la formule F (déjà vue)

$$\Big((s \Rightarrow (b \vee t)) \wedge ((b \vee a) \Rightarrow (r \wedge m)) \wedge \neg r\Big) \Rightarrow (s \Rightarrow t)$$

Pour montrer que F est une tautologie, on va monter que $(\neg F)$ est contradictoire.

$(\neg F) \equiv C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6 \wedge C_7 \wedge C_8$

où les clause C_i $1 \leq i \leq 8$ sont :

$$\begin{array}{lcl}
C_1 & = & ((\neg s) \vee b \vee t) \\
C_2 & = & ((\neg b) \vee r) \\
C_3 & = & ((\neg b) \vee m) \\
C_4 & = & ((\neg a) \vee r) \\
C_5 & = & ((\neg a) \vee m) \\
C_6 & = & (\neg r) \\
C_7 & = & s \\
C_8 & = & (\neg t)
\end{array}$$

4. Résolution — ex. 1 (suite)

$$
\begin{aligned}
C_1 &= ((\neg s) \vee b \vee t) \\
C_2 &= ((\neg b) \vee r) \\
C_3 &= ((\neg b) \vee m) \\
C_4 &= ((\neg a) \vee r) \\
C_5 &= ((\neg a) \vee m) \\
C_6 &= (\neg r) \\
C_7 &= s \\
C_8 &= (\neg t)
\end{aligned}
$$

	Résolution :
$C_9 = ((\neg s) \vee t \vee r)$	résolvant de C_1 et C_2
$C_{10} = ((\neg s) \vee t)$	résolvant de C_{10} et C_6
$C_{11} = t$	résolvant de C_{11} et C_7
$C_{12} = \perp$	résolvant de C_{12} et C_8
	contradiction trouvée

5. Résolution — ex. 2

Soit la formule $F = C_a \wedge C_b \wedge C_c$

$$
\begin{aligned}
C_a &= ((\neg u) \vee v \vee w) \\
C_b &= (t \vee w) \\
C_c &= ((\neg w) \vee u)
\end{aligned}
$$

	Résolution
$C_d = v \equiv (u \vee (\neg u) \vee v)$	résolution de C_a et C_c
$C_e = (u \vee w)$	résolution de C_b et C_c
$C_f = (v \vee w)$	résolution de C_e et C_a
	plus de résolution
	clauses consistantes

6. Propriétés de l’algorithme de résolution

Cet algorithme termine toujours — il n’y a qu’un nombre fini de clauses possible sur un alphabet donné.
Lorsque la résolution n’ajoute plus de nouvelles clauses, soit la clause vide $\perp$ a été obtenue et l’ensemble de clauses de départ est contradictoire — comme on l’a vu, les clauses ajoutées sont conséquences logiques des clauses de départ, soit il n’y a plus de résolutions possibles.

7. Complétude de la résolution propositionnelle

Si $\mathcal{C} = \{\mathcal{C}_i, i \in I\}$ (clauses) pas satisfiable alors on peut montrer que $\mathcal{C} = \{\mathcal{C}_i, i \in I\}$ entraîne la clause vide par résolution.

Il sera pratique de noter les clauses $A_1, ..., A_n \vdash B_1, ..., B_p$ plutôt que $\neg A_1 \vee ... \vee \neg A_n \vee B_1 \vee ... \vee B_p \equiv (\neg A_1 \wedge ... \wedge \neg A_n) \Rightarrow (B_1 \vee ... \vee B_p)$. afin de séparer les atomes positifs des atomes négatifs.

Sous cette forme la résolution, c'est la règle de coupure :

$$\frac{\Gamma \vdash \Delta, E \qquad E, \Theta \vdash \Phi}{\Gamma, \Theta \vdash \Delta, \Phi}\, cut$$

8. Clauses simplifiées

On supprime les clauses $A \vee \neg A \vee \ldots$ (une telle clause est vraie)

On ne met qu'une fois un A ou un $\neg A$:

$A \vee A \vee \ldots \rightarrow A \vee \ldots$

$\neg A \vee \neg A \vee \ldots \rightarrow \neg A \vee \ldots$

Dans une clause simplifiée chaque lettre apparaît au plus une fois soit positivement soit négativement.
Il y a donc au plus 3^n clauses sur n lettres.

9. Variables positives inutiles

Si un ensemble de clauses $\mathcal{C}$ n'est pas satisfiable et si A (symbole propositionnel) n'apparait dans aucune prémisse de clause (c.-à-d. A n'apparaît que positivement dans les disjonctions) alors l'ensemble des clauses sans A de $\mathcal{C}$, disons $\mathcal{C}'$, n'est pas satisfiable non plus.

Si les clauses de $\mathcal{C}'$ étaient simultanément satisfiables, par une valuation v en prenant $v(A) = 1$ pour A on obtient une valuation qui rend vrai les clauses de $\mathcal{C}'$ (sans A) mais aussi les clauses avec A qui apparâit positivement dans ces clauses.

10. Variables négatives inutiles

Si un ensemble de clauses $\mathcal{C}$ n'est pas satisfiable et si A (symbole propositionnel) n'apparait dans aucune conclusion de clause (c.-à-d. A n'apparaît que négativement dans les disjonctions) alors l'ensemble des clauses sans A de $\mathcal{C}$, disons $\mathcal{C}'$, n'est pas satisfiable non plus.

Si les clauses de $\mathcal{C}'$ étaient simultanément satisfiables, par une valuation v en prenant $v(A) = 0$ pour A on obtient une valuation qui rend vrai les clauses de $\mathcal{C}'$ (sans A) mais aussi les clauses avec A qui apparait négativement dans ces clauses.

11. Variables inutiles

D’après deux diapos précédentes,
dans un ensemble de clauses non satisfiables,
on peut toujours supposer que chaque variable apparait
au moins une fois en positif et au moins une fois en négatif.

12. Non satisfiable => réfutable (base : 1 variable)

S'il n y a que 1 symbole de proposition,
il n'y a que deux clauses simplifiées possibles A et $\neg A$.

Pour avoir un ensemble de clauses non satisfiable
il faut prendre les deux.

L'ensemble des deux clauses p_1 et $\neg p_1$
($\vdash p_1$ et $p_1 \vdash$)
est réfutable par résolution.

13. Non satisfiable => réfutable $n \to n+1$

On suppose que tous les ensembles de clauses non satisfiables avec au plus n lettres sont réfutables.

On montre que c'est vrai pour un ensemble de clauses à $n+1$ variables.

On choisit une variable A parmi les $n+1$ et on partage les clauses en trois ensembles disjoints (il n'y a pas d'autre possibilité, pas plusieurs A, pas de A et en prémisse et en conclusion) :

$\mathcal{C}^0$: pas de A

$\mathcal{C}^+$: un A en conclusion de la clause $\Theta_i \vdash \Delta_i, A$

$\mathcal{C}^-$: un A en prémisse de la clause $A, \Psi_j \vdash \Phi_j$

14. Non satisfiable => réfutable $n \to n+1$ — suite

On peut résoudre n'importe quelle clause $\Theta_i \vdash \Delta_i, A$ de Γ^+
avec n'importe quelle clause $A, \Psi_j \vdash \Phi_j$ de Γ^-,
cela donne une clause sans A :

$$\Theta_i, \Psi_j \vdash \Delta_i, \Phi_j$$

appelons $\bar{\mathcal{C}}^0$ la réunion de $\mathcal{C}^0$ avec ces clauses obtenues par résolution dune clause de $\mathcal{C}^+$ avec un clause de $\mathcal{C}^-$ ce sont donc des clauses sans A.
On va montrer que $\bar{\mathcal{C}}^0$ nest pas satisfiable car sinon $\mathcal{C}$ le serait.

15. Preuve : $\bar{\mathcal{C}}^0$ satisfiable $\Rightarrow \mathcal{C}$ satisfiable (1/4)

Supposons que $\bar{\mathcal{C}}^0$ satisfiable pour une valuation v, valuation que l'on complète par $v(A) = 0$ — on notera v' la valuation v complétée par $v'(A) = 1$.
Si v rend toutes les clauses de $\mathcal{C}$ vrai, on a montré ce qu'il fallait.
Sinon une des clauses de $\mathcal{C}$ n'est pas satisfaite par v,
et comme $v(A) = 0$ c'est une clause de $\mathcal{C}^+$ qui n'est pas satisfaite.

16. Preuve : $\bar{\mathcal{C}}^0$ satisfiable $\Rightarrow \mathcal{C}$ satisfiable (2/4)

Donc pour au moins une clause $\Theta_{i_0} \vdash \Delta_{i_0}, A$ de Γ^+ (avec un A positif) , $v(\Theta_{i_0}) = 1$ et $v(\Delta_{i_0}) = 0$.
Pour chaque clause $A, \Psi_j \vdash \Phi_j$ de Γ^- la combinaison par résolution donne
$\Theta_{i_0}, \Psi_j \vdash \Delta_{i_0}, \Phi_j$.
Cette clause étant dans $\bar{\mathcal{C}}^0$, elle est validée par v : donc pour chaque clause $A, \Psi_j \vdash \Phi_j$ de $\mathcal{C}^-$ $v(\Psi_j) = 0$ ou $v(\Phi_j) = 1$.

17. Preuve : $\bar{\mathcal{C}}^0$ satisfiable $\Rightarrow \mathcal{C}$ satisfiable (3/4)

Considérons maintenant la valuation v' identique à v sur toutes les lettres sauf sur A : $v'(A) = 1$.
Comme il n'y a pas de A dans les clauses de $\bar{\mathcal{C}}^0$, v' valide toutes les clauses de $\bar{\mathcal{C}}^0$ et donc toutes les clauses de $\bar{\mathcal{C}}$.
Chaque clause $A, \Psi_j \vdash \Phi_j$ de $\mathcal{C}^-$ est validée par v' car $v'(\Psi_j) = v(\Psi_j) = 0$ ou $v'(\Phi_j) = v(\Phi_j) = 1$: il n'y a pas de A dans Ψ_j ni dans Φ_j.
Chaque clause $\Theta_i \vdash \Delta_i, A$ de Γ^+ est validée par v' car $v'(A) = 1$.

18. Preuve : $\bar{\mathcal{C}}^0$ satisfiable $\Rightarrow$ $\mathcal{C}$ satisfiable (4/4)

v' valide donc toutes les clauses de $\mathcal{C}^0$ de $\mathcal{C}^+$ et de $\mathcal{C}^-$, donc v' valide $\mathcal{C}$.
Donc $\bar{\mathcal{C}}^0$ satisfiable entraine $\mathcal{C}$ satisfiable ou :
$\mathcal{C}$ insatisfiable entraine $\bar{\mathcal{C}}^0$ insatisfiable.
par hypothèse d'induction la clause vide s'obtient à partir de $\bar{\mathcal{C}}^0$ qui a une variable de moins
$\bar{\mathcal{C}}^0$ s'obtient par résolution à partir de $\mathcal{C}$

19. Récapitulatif

Etant donné un ensemble de clauses, en appelant $\mathcal{C}^0$ les clauses sans A $\mathcal{C}^+$: les clauses avec un A en conclusion de la clause $\Theta_i \vdash \Delta_i, A$ et $\mathcal{C}^-$ les clauses un A en prémisse de la clause $A, \Psi_j \vdash \Phi_j$

Nous avons montré que si $\mathcal{C}$ est insatisfiable si et seulement si l'ensemble des clauses sans A constituées de la réunion de $\mathcal{C}^0$ avec les résolvant de $\mathcal{C}^+$ et $\mathcal{C}^-$ (eux aussi sans A) est insatisfiable.

Comme les clauses de $\bar{\mathcal{C}}^0$ ont une lettre de moins (A), par hypothèse de récurrence , $\bar{\mathcal{C}}^0$ engendre $\perp$ par résolution et comme $\mathcal{C}$ engendre $\bar{\mathcal{C}}^0$ par résolution, $\mathcal{C}$ engendre $\perp$ par résolution.

La réciproque est évidente : un ensemble de clause engendrant $\perp$ par résolution est instatisfiable, puisque, comme vu au début toute valuation rendant vrai un ensemble de clause rend vrai tout résolvant de ces clauses.

Un ensemble de clauses est insatisfiable si et seulement si il engendre $\perp$ par résolution.

20. Décidabilité du calcul propositionnel

On dispose d'un algorithme pour vérifier
qu'une formule ou un ensemble de clause est insatisfiable
ce qui donne une procédure pour décider
si une proposition est satisfiable ou non.

Cette méthode consiste à mettre la formule sous forme clausale,
à simplifier en supprimant les clauses qui contiennent
un même atome en prémisse et en conclusion (simplification)
en supprimant les répétitions à gauche comme à droite (simplification),
et à appliquer librement la résolution,
et à simplifier les clauses obtenues et à recommencer.

21. Correction de l’algo décidabilité

Comme l’ensemble des symboles de proposition est fini,
l’ensemble des clauses simplifiées est aussi fini.
Donc l’algorithme atteint un point fixe :
si la clause vide est atteinte,
l’ensemble des clause était insatisfiable,
et sinon, il est satisfiable.

22. Compétude et équivalence séquents / résolution

D'après ce qu'on a vu sur les calcul des séquents et sur la résolution, on a 4 propriétes équivalentes :

F est insatisfiable
sa négation $\neg F$ est une tautologie,
F est réfutable par résolution
$\neg F$ est démontrable dans le calcul des séquents (déjà vu).

C'est une manière de voir que calcul des séquents correspondent à la même notion de validité et de satisfiabilité.