Chapitre 4

Résolution Propositionnelle

Une *tautologie* est une formule propositionnelle vraie pour toutes les valeurs des propositions atomiques qui la composent. Comment détecter automatiquement qu'une formule est une tautologie ? On peut construire sa table de vérité, mais, s'il y a n propositions atomiques différentes dans la formule, sa table de vérité compte 2^n lignes, donc son calcul peut être trop long. On cherche des méthodes plus efficaces.

On étudie ici l'une de ces méthodes, appelée "résolution", dans le cadre de la logique des propositions.

La résolution propositionnelle est une procédure de décision de la validité des formules propositionnelles, qui procède par *réfutation*. Ceci est expliqué dans la partie 4.1. Ensuite, la méthode procède en deux étapes. La première étape, détaillée dans la partie 4.2, réduit toute formule en une conjonction de clauses. La seconde étape, détaillée dans la partie 4.4, sature cet ensemble de clauses en n'appliquant qu'une seule règle de déduction, appelée *résolution*, présentée dans la partie 4.3.

4.1 Réfutation

Pour prouver qu'une formule F est une tautologie (= une formule valide), la méthode cherche à montrer que $\neg F$ n'est pas satisfaisable, c'est-à-dire est fausse pour toute les valuations. Autrement dit, que $\neg F$ est « contradictoire ». Nous verrons que la méthode décompose $\neg F$ en un ensemble de formules plus simples, dans lequel on cherche une contradiction interne. Ce « raisonnement par contradiction » est appelé *réfutation*.

Remarque. Si la question initiale n'est pas celle de la *validité* (tautologie) de F, mais celle de sa *satisfaisabilité*, alors cette première étape de négation est inutile. On cherche directement une contradiction dans F, et on échange les conclusions. Autrement dit, on poursuit la méthode avec F au lieu de $\neg F$.

4.2 Mise en forme clausale

Afin d'écrire des algorithmes plus simples de raisonnement avec des formules propositionnelles, on transforme souvent ces formules en formules équivalentes, mais plus structurées, appelées *formes normales*. Cette partie définit la forme normale conjonctive (CNF, pour *Conjunctive Normal Form*), également appelée *forme clausale*.

4.2.1 Définitions

Un *littéral* est soit une proposition atomique, soit la négation d'une proposition atomique. Une *clause* est une disjonction $L_1 \vee \ldots \vee L_n$ ($n \geq 0$) de littéraux. La clause vide ($n = 0$) est la constante *faux*. Elle est notée $\perp$. Une formule est en CNF si c'est une conjonction $C_1 \wedge \ldots \wedge C_k$ ($k \geq 0$) de clauses. Si $k = 0$, c'est la constante *vrai*, notée $\top$.

L'algorithme de mise en forme normale consiste à appliquer répétitivement les règles de la figure 4.1, jusqu'à ce qu'aucune règle ne soit plus applicable. C'est une technique dite de *réécriture*. Ce système de réécriture a deux bonnes propriétés : il termine toujours, et la formule qui en résulte est en CNF. Le symbole $\rightsquigarrow$ signifie « se réécrit en ».

$$
\begin{array}{rcl}
F \Rightarrow G & \rightsquigarrow & \neg F \vee G \\
\neg\neg F & \rightsquigarrow & F \\
\neg(F \wedge G) & \rightsquigarrow & (\neg F \vee \neg G) \\
\neg(F \vee G) & \rightsquigarrow & (\neg F \wedge \neg G) \\
F \vee (G \wedge H) & \rightsquigarrow & (F \vee G) \wedge (F \vee H) \\
(F \wedge G) \vee H & \rightsquigarrow & (F \vee H) \wedge (G \vee H) \\
(F \vee F) & \rightsquigarrow & F
\end{array}
$$

FIG. 4.1 – Règles de mise en forme clausale.

Pour toute formule propositionnelle F, on note $\mathit{cnf}(F)$ le résultat de cet algorithme appliqué à cette formule. L'intérêt de la décomposition d'une formule en CNF est donné par le théorème suivant, où $\vdash$ est la règle de résolution, définie dans la partie suivante.

PROPRIÉTÉ 4.1 : F est non satisfaisable si et seulement si $\mathit{cnf}(F) \vdash \bot$.

4.3 Règle de résolution propositionnelle

La notation $\vdash$ désigne ici l'application répétitive de la règle suivante, dite de résolution propositionnelle :

$$\frac{\neg P \vee C, \qquad P \vee D}{C \vee D} \tag{4.1}$$

où P est un atome (une proposition atomique) et où C et D sont des clauses.

Cette règle déduit une nouvelle clause à partir de deux clauses connues. Elle déduit la clause vide si C et D sont vides. Si C et D ont des littéraux communs,

– soit on les élimine par la règle de factorisation

$$\frac{L \vee L \vee C}{L \vee C}$$

où L est un littéral et C est une clause,

– soit on considère les clauses comme des ensembles de littéraux, donc sans doublons.

Remarque. Pour être tout à fait précis, on précise qu'il faut au besoin réordonner les littéraux d'une clause par commutativité pour que la règle de résolution s'applique littéralement :

$$\frac{F \vee G}{G \vee F}$$

En pratique, on appliquera cette réorganisation implicitement, de sorte que directement

$$\frac{C \vee \neg P, \qquad P \vee D}{C \vee D}$$

et

$$\frac{\neg P \vee C, \qquad D \vee P}{C \vee D}$$

Soit E un ensemble de clauses et C une clause. On note $E \vdash C$ si des applications répétitives de la règle de résolution permettent de déduire la clause C à partir de l'ensemble de clauses E.

Par exemple, $\{p_1 \vee p_2, \neg p_2 \vee \neg p_3, p_3\} \vdash p_1$ en deux étapes.

$$\frac{p_1 \ \vee \ p_2, \qquad \neg\, p_2 \ \vee \ \neg\, p_3}{p1 \ \vee \ \neg\, p_3}$$

puis

$$\frac{p1 \ \vee \ \neg\, p_3, \qquad p_3}{p_1}$$

La partie suivante introduit du vocabulaire pour décrire plus précisément une étape d'application de la règle de résolution propositionnelle.

4.3.1 Résolvante d'une paire de clauses

Deux clauses C_1 et C_2 forment une *paire résoluble* s'il existe une **et une seule** proposition atomique P telle que P appartient à C_1 et $\neg P$ appartient à C_2. Dans ce cas, on appelle *résolvante* de C_1 et C_2 la clause, notée $res(C_1, C_2)$, obtenue en prenant la disjonction des littéraux de C_1 et C_2 moins cette paire opposée.

Par exemple, $C_1 \stackrel{\text{def}}{=} A \ \vee \ \neg\, B \ \vee \ \neg\, C$ et $C_2 \stackrel{\text{def}}{=} B \ \vee \ \neg\, C \ \vee \ D$ sont résolubles, et

$$res(C_1, C_2) = A \ \vee \ \neg\, C \ \vee \ D$$

En revanche, $C_3 \stackrel{\text{def}}{=} A \ \vee \ \neg\, B$ et $C_4 \stackrel{\text{def}}{=} \neg\, A \ \vee \ B$ ne forment pas une paire résoluble, car il y'a **deux** propositions atomiques (A et B) qui apparaissent affirmée dans une clause et niée dans l'autre.

4.4 La méthode de résolution propositionnelle

Cette méthode est connue sous le nom de « méthode de Robinson ». Elle est simple à mettre en œuvre, mais il existe des méthodes plus efficaces en pratique.

Pour toute formule propositionnelle F dont on cherche à établir la validité :

1. Calculer $cnf(\neg F)$, pour mettre la négation de F sous forme clausale.
2. Appliquer répétitivement la règle de résolution aux clauses résultant de $cnf(\neg F)$.
3. Si la clause vide $\perp$ est produite, $\neg F$ n'est pas satisfaisable, donc F est valide.
4. Sinon, si plus aucune déduction nouvelle n'est possible, $\neg F$ est satisfaisable, donc F n'est pas valide.

Lorsqu'on ne peut plus déduire de nouvelle clause, on dit qu'on a « saturé » l'ensemble de clauses initial.

Exercice 4.1. *Démontrer par cette méthode la validité de la formule* $G \stackrel{\text{def}}{=} ((A \ \wedge \ B) \ \Rightarrow \ (A \ \vee \ B))$.

Solution

Validité de $G \stackrel{\text{def}}{=} ((A \ \wedge \ B) \ \Rightarrow \ (A \ \vee \ B))$?

mise en CNF de $\neg G$.

$$\begin{aligned} \neg G = \neg((A \ \wedge \ B) \ \Rightarrow \ (A \ \vee \ B)) & \\ & \leadsto \neg(\neg(A \ \wedge \ B) \ \vee \ (A \ \vee \ B)) \\ & \leadsto (\neg\neg(A \ \wedge \ B) \ \wedge \ \neg(A \ \vee \ B)) \\ & \leadsto ((A \ \wedge \ B) \ \wedge \ \neg(A \ \vee \ B)) \\ & \leadsto ((A \ \wedge \ B) \ \wedge \ (\neg A \ \wedge \ \neg B)) \end{aligned}$$

soit $A \ \wedge \ B \ \wedge \ \neg\, A \ \wedge \ \neg\, B$ en simplifiant le parenthésage.

Résolution.

On obtient donc l'ensemble suivant $\{C_1, C_2, C_3, C_4\}$ de 4 clauses, avec :

$$C_1 = A, \quad C_2 = B, \quad C_3 = \neg A, \quad C_4 = \neg B$$

Comme $res(C_1, C_3) = \perp$, alors $\neg G$ est contradictoire, et donc G est valide.