NOTES DE COURS
LA MÉTHODE DES ARBRES POUR LE CALCUL DES PROPOSITIONS

JOSEPH VIDAL-ROSSET

TABLE DES MATIÈRES

1. L'ESSENTIEL SUR LA MÉTHODE DES ARBRES

1.1. **Méthode sémantique ou syntaxique ?** La méthode des arbres est une méthode de décision syntaxique, qui est issue de l'invention par Evert Willem Beth (7 juillet 1908 - 12 avril 1964) philosophe et logicien néerlandais de la méthode des tableaux sémantiques . En réalité c'est une question délicate de savoir si cette méthode est syntaxique, sémantique, ou « entre les deux », ou les deux à la fois. On peut lire sur cette question à l'article de Philippe de Rouilhan [4].

On dira simplement ici, qu'à la différence de la méthode des tables de vérité ou de l'algorithme de Quine, la méthode des arbres est une procédure qui ne fait pas un usage direct des valeurs de vérité pour décider de la validité d'une formule. C'est pourquoi Lepage [3, p. 83] la considère avec raison comme une méthode syntaxique (Ruyer [5, p. 93] la qualifie d'« analytique », parce que cette méthode procède par la décomposition de la formule testée en ses sous-formules, et qu'à la différence de la déduction naturelle, toutes les règles sont des règles d'élimination.)

1.2. **Le comment et le pourquoi de cette méthode.** Pour comprendre l'essentiel de cette méthode, il faut avoir à l'esprit les points suivants :

i) La transformation d'une formule φ en arbre n'est que l'écriture de φ sous une forme différente, où l'on a fait disparaître tous les $\rightarrow$ et tous les $\leftrightarrow$.
ii) La méthode des arbres est une méthode *indirecte* : pour prouver la validité de φ on nie φ et si la négation de φ est une contradiction, alors φ est valide.

Date: 24 février 2011.

iii) Le fait que l'arbre d'une formule $\neg\varphi$ soit une contradiction *se voit* par le fait que tous les chemins de l'arbres contiennent une contradiction, c'est-à-dire l'affirmation d'une formule et de sa négation (comme p et $\neg p$ par exemple). Un chemin qui contient une contradiction est dit « fermé », on symbolise cette fermeture par une $\times$ à la fin du chemin.

iv) Si l'arbre de $\neg\varphi$ contient au moins un chemin « ouvert », c'est-à-dire un chemin qui ne contient pas de contradiction, alors la négation de φ est satisfaite (ou vraie) pour toutes les valeurs de ce chemin ouvert et donc φ n'est pas valide. L'ultime intérêt de la méthode des arbres est de montrer, lorsqu'une formule φ n'est pas valide, toutes les valeurs qui satisfont $\neg\varphi$.

1.3. **Les règles de la méthode des arbres pour le calcul des propositions.** Voici les règles de transformation de l'écriture des connecteurs pour la méthode des arbres.

(1) Règle dite d'absurdité classique , qui est l'essence même de la logique classique : la double négation d'une formule A se simplifie par l'affirmation de A :

$$\begin{array}{c} \neg\neg A \\ A \end{array}$$

(2) Règle pour la conjonction :

$$\begin{array}{c} A \wedge B \\ A \\ B \end{array}$$

(3) Règle pour la négation de la conjonction :

$$\begin{array}{c} \neg(A \wedge B) \\ \neg A \qquad \neg B \end{array}$$

(4) Règle pour la disjonction :

$$\begin{array}{c} (A \vee B) \\ A \qquad B \end{array}$$

(5) Règle pour la négation de la disjonction :

$$\begin{array}{c} \neg(A \vee B) \\ \neg A \\ \neg B \end{array}$$

(6) Règle pour le conditionnel :

$$\begin{array}{cc} \multicolumn{2}{c}{(A \to B)} \\ \neg A & B \end{array}$$

(7) Règle pour la négation du conditionnel :

$$\begin{array}{c} \neg(A \to B) \\ A \\ \neg B \end{array}$$

(8) Règle pour le biconditionnel :

$$\begin{array}{cc} \multicolumn{2}{c}{(A \leftrightarrow B)} \\ A & \neg A \\ B & \neg B \end{array}$$

(9) Règle pour la négation du biconditionnel :

$$\begin{array}{cc} \multicolumn{2}{c}{\neg(A \leftrightarrow B)} \\ A & \neg A \\ \neg B & B \end{array}$$

1.4. **Trois exemples.**

Exemple 1. *Test de la validité de*

$$\neg B \to \neg A \vdash_c A \to B \tag{1}$$

Démonstration. a) Première étape : On affirme l'hypothèse et l'on nie la conclusion, ce qui donne les formules initiales suivantes :

$$\begin{array}{c} \neg B \to \neg A \\ \neg(A \to B) \end{array}$$

b) Seconde étape : on transforme ensuite les formules initiales en arbre en appliquant les règles et en fermant les chemins où se trouvent les contradictions. On coche les formules pour indiquer qu'elles sont traitées et que l'on n'y revient plus. L'arbre se construit ici en deux étapes. On commence par la seconde formule, puisque sa transformation donne une conjonction et que, pour éviter des ramifications redondantes, on s'efforce de traiter les conjonctions avant les disjonctions ; on obtient donc :

$$
\begin{array}{c}
\neg B \to \neg A \\
\neg(A \to B) \checkmark \\
A \\
\neg B
\end{array}
$$

puis l'on traite la première formule et l'on ferme les chemins où se trouvent une variable propositionnelle et sa négation :

$$
\begin{array}{c}
\neg B \to \neg A \checkmark \\
\neg(A \to B) \checkmark \\
A \\
\neg B \\
\begin{array}{cc}
\neg\neg B \checkmark & \neg A \\
B & \times \\
\times &
\end{array}
\end{array}
$$

Conclusion : tous les chemins de l'arbre sont fermés, donc (1) est bien un théorème de logique classique. □

(Remarque : pour que le chemin à gauche de l'arbre ferme, il est nécessaire que l'on applique la règle d'absurdité classique, afin de simplifier $\neg\neg B$ *par* B*. Cela suffit à montrer qu'en revanche :*

$$\neg B \to \neg A \nvdash_i A \to B$$

en clair $(A \to B)$ *n'est pas déductible de* $(\neg B \to \neg A)$ *en logique intuitionniste.)*

Exemple 2. *Preuve de la validité de*

$$((A \vee B) \to C) \to ((A \to C) \vee (B \to C)) \tag{2}$$

Démonstration.

$$
\begin{array}{c}
\neg(((A \vee B) \to C) \to ((A \to C) \vee (B \to C))) \checkmark \\
((A \vee B) \to C) \quad \checkmark \\
\neg((A \to C) \vee (B \to C)) \checkmark \\
\neg(A \to C) \checkmark \\
\neg(B \to C) \quad \checkmark \\
B \\
\neg C \\
A \\
\neg C \\
\begin{array}{cc}
\neg(A \vee B) \checkmark & C \\
\neg A & \times \\
\neg B & \\
\times &
\end{array}
\end{array}
$$

Tous les chemins se ferment ce qui prouve que la formule (2) est valide en logique classique, ce qui s'écrit :

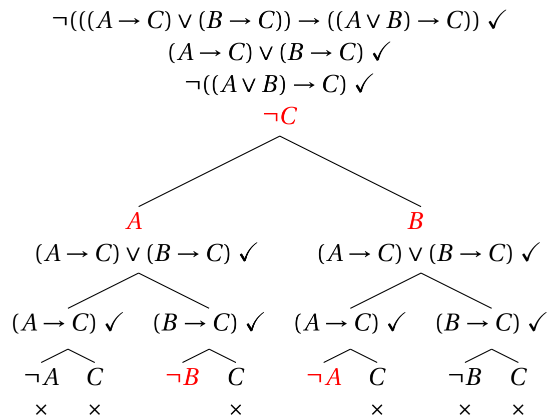
$$\vdash_c ((A \vee B) \to C) \to ((A \to C) \vee (B \to C))$$

□

Exemple 3. *Réfutation de la validité de*

$$(((A \to C) \vee (B \to C)) \to ((A \vee B) \to C)) \tag{3}$$

Démonstration. L'arbre de $\neg(((A \to C) \vee (B \to C)) \to ((A \vee B) \to C))$ permet de donner un contre-modèle de (3) en montrant les valeurs pour lesquelles la négation de (3) est satisfaite :



Pour savoir les valeurs pour lequelles la formule

$$\neg(((A \to C) \vee (B \to C)) \to ((A \vee B) \to C))$$

est satisfaite, il suffit de suivre tous les chemins qui ne ferment pas et d'interpréter toute formule positive P par $P = \top$ (autrement dit « P est vrai ») et toute formule négative $\neg P$ par $P = \bot$ (autrement dit « P est faux »). La lecture de l'arbre permet de dire que pour les valeurs suivantes

$$C = \bot, A = \top, B = \bot$$

ou

$$C = \bot, A = \bot, B = \top$$

la négation de (3) est satisfaite, ce qui réfute la validité de (3). □

2. Lectures conseillées

On peut compléter utilement ce résumé par la lecture de Lepage [3], Ruyer [5], Vernant [6] , Gochet *et al.* [2] , Bell *et al.* [1].

3. Outils

Pour vérifier les exercices, il existe un logiciel en ligne, le Tree Proof Generator. S'en servir intelligemment, en s'efforçant de comprendre comment procède le test, est une bonne manière de progresser dans l'apprentissage de cette méthode.

RÉFÉRENCES

[1] BELL, J.L. & DEVIDI, D. & SOLOMON, G. – *Logical Options : An Introduction to Classical and Alternative Logics*, Broadview Press, Peterborough, Ontario, Canada, 2001.

[2] P. GOCHET P., & GRIBOMONT – *Logique - méthodes pour l'informatique fondamentale*, vol. 1, Hermès, Paris, 1991.

[3] F. LEPAGE – *Eléments de logique contemporaine*, Presses de l'Université de Montréal, 2001.

[4] P. DE ROUILHAN – « Les tableaux de Beth : syntaxe ou sémantique ? », *Philosophia Scientiæ* **3** (1999), no. 4, p. 303–322.

[5] B. RUYER – *Logique*, P.U.F., Paris, 1990.

[6] D. VERNANT – *Introduction à la logique standard*, Champs Université, Flammarion, Paris, 2001.

UNIVERSITÉ DE NANCY 2 - DÉPARTEMENT DE PHILOSOPHIE -ARCHIVES POINCARÉ, UMR-7117 - CNRS -, 91, BD LIBÉRATION, F-54000 NANCY.

E-mail address: `joseph.vidal-rosset@univ-nancy2.fr`