

Diagrammes de décision binaires

Cours de Logique

ENS Cachan, L3, 2015–2016

On se restreint à la logique propositionnelle. Nous allons voir comment calculer “efficacement” une forme canonique des formules. Ceci nous permettra de décider la satisfiabilité, validité, et (surtout) d’autres choses.

Une façon de voir les choses est de partir des tables de vérité. Calculer la table de vérité d’une formule permet de déterminer si elle est SAT, valide, équivalente à une autre, etc. Cette méthode a la complexité optimale en théorie, mais en pratique c’est bien trop coûteux : le problème est qu’on est sûrs de payer le pire cas. On cherche donc une représentation potentiellement plus compacte qu’une table de vérité. Il s’agira d’un intermédiaire entre la syntaxe (pas assez canonique) et les tables de vérité (pas assez symboliques, ne permettant pas la paresse).

1 Graphes de Shannon

Ce sont, en langage plus courant, des arbres de décision. La raison pour laquelle on travaille sur des graphes plutôt que des arbres apparaîtra plus loin.

Definition 1.1. L’ensemble des *graphes de Shannon* est le plus petit ensemble de graphes contenant \top , \perp et $(A \rightarrow \Phi_+; \Phi_-)$.

Plus précisément, ces graphes sont des DAGs dont les feuilles sont étiquetées par un booléen, et les noeuds internes par des atomes.

On définit inductivement la formule associée à un graphe, en posant

$$(A \rightarrow \Phi_+; \Phi_-) = (A \Rightarrow \phi_+) \wedge (\neg A \Rightarrow \phi_-)$$

où ϕ_+ est la formule associée à Φ_+ et de même pour ϕ_- .

Proposition 1.2. Toute formule propositionnelle Φ est équivalente à $\Phi_A = (A \Rightarrow \Phi[\top/A]) \wedge (\neg A \Rightarrow \Phi[\perp/A])$.

Démonstration. Considérons une interprétation I . Si $I \models A$ alors $I \models \Phi_A$ ssi $I \models \Phi[\top/A]$ ssi $I \models \Phi$. \square

Évidemment, on peut calculer un graphe de Shannon équivalent à une formule en itérant le résultat précédent. On pourra remarquer que c’est toujours un arbre.

Example 1.3. On part de :

$$(((A \Rightarrow B) \wedge C) \Rightarrow A) \vee \neg C$$

On décompose selon A :

$$(A \Rightarrow (((\top \Rightarrow B) \wedge C) \Rightarrow \top) \vee \neg C) \wedge (\neg A \Rightarrow (((\perp \Rightarrow B) \wedge C) \Rightarrow \perp) \vee \neg C)$$

On simplifie ($\perp \Rightarrow \dots, \dots \Rightarrow \top$, etc.) :

$$(A \Rightarrow \top) \wedge (\neg A \Rightarrow \neg C)$$

Cela correspond au graphe $(A \rightarrow \top; (C \rightarrow \perp; \top))$. Ou encore, $\neg A \Rightarrow \neg C$.

À partir de cette formule, si on décompose selon C on obtient :

$$(C \Rightarrow (\neg A \Rightarrow \perp)) \wedge (\neg C \Rightarrow (\neg A \Rightarrow \top))$$

C'est à dire $(C \rightarrow A; \top)$.

Les graphes de Shannon sont symboliques, parfois très compacts. Mais ils ne sont pas canoniques : deux graphes distincts (non isomorphes) peuvent correspondre à la même formule. On cherche donc à faire mieux. Deux pistes : les graphes permettent le partage, il faut décider si on passe aux arbres ou si on impose un partage maximal ; même avec une notion “graphique” de partage, il reste de la non-canonlicité, notamment (mais pas seulement) dans l'ordre d'énumération des variables.

2 Diagrammes de décision binaires

On fixe par la suite un ordre arbitraire sur les atomes, noté $<$. Quand on parle d'égalité de graphes, il s'agit d'isomorphisme. On pourrait aussi supposer qu'on ne travaille que sur des graphes où les deux notions coïncident, ce qu'on obtiendra en pratique.

Definition 2.1. Un graphe de Shannon est un BDD s'il est ordonné et réduit :

- Ordonné : pour tout sous-graphe $(A \rightarrow \Phi_+; \Phi_-)$, tous les noeuds internes de Φ_+ et Φ_- sont étiquetés par des B tel que $A < B$.
- Réduit : il n'y a aucun sous-graphe de la forme $(A \rightarrow \Phi'; \Phi')$.

Ces deux conditions rendent les BDDs canoniques.

Proposition 2.2. Deux BDDs sont égaux (isomorphisme de graphe) ssi ils sont logiquement équivalents.

Démonstration. Un sens est immédiat : s'ils sont égaux ils sont logiquement équivalents. Sinon, considérons deux graphes logiquement équivalents, montrons qu'ils sont égaux. On procède par induction sur la somme des tailles des graphes.

Les cas des feuilles sont évidents.

Si les deux noeuds racines ont même étiquette, *i.e.*, $(A \rightarrow \Phi_+; \Phi_-) \sim (A \rightarrow \Phi'_+; \Phi'_-)$, on observe que $\Phi_+ \sim \Phi'_+$ (et de même côté $-$) : en effet on a $\Phi[\top/A] \sim \Phi'[\top/A]$

$\Phi_+[\top/A] \sim \Phi_+$, puis $\Phi'[\top/A] \sim \Phi'_+$ de même, et enfin $\Phi'_+ \sim \Phi_+$ car $\Phi \sim \Phi'$ et l'équivalence logique est préservée par substitution. On conclut alors par hypothèses d'induction.

Sinon on a $(A \rightarrow \Phi_+; \Phi_-) \sim (B \rightarrow \Phi'_+; \Phi'_-)$ avec $A < B$. Par le caractère ordonné de Φ' , A n'apparaît pas du tout dans le BDD Φ' . Par le même raisonnement qu'au dessus, on en déduit que $\Phi_+ \sim \Phi'$, mais aussi $\Phi_- \sim \Phi'$. Par hypothèse d'induction, Φ_+ et Φ_- seraient alors égaux, ce qui contredit le fait que le BDD soit réduit. \square

Pour construire le BDD associé à une formule, on peut construire comme précédemment un graphe de Shannon associé à la formule, en respectant l'ordre des variables. Il restera alors seulement à le réduire, itérativement, pour obtenir un BDD. Cette méthode est, à tout coup, exponentielle en espace et en temps; on peut faire mieux.

2.1 Calcul sur les BDDs

Attachons nous à définir les opérations logiques directement sur les BDDs.

Définition 2.3. Étant donné un BDD Φ , on définit $f_-(\Phi)$ comme suit :

- $f_-(\top) = \perp$ et vice versa;
- $f_-(A \rightarrow \Phi_+; \Phi_-) = (A \rightarrow f_-(\Phi_+); f_-(\Phi_-))$.

On vérifie tout d'abord que $f_-(\Phi)$ est un BDD : c'est évident pour la condition d'ordre; pour le caractère réduit, cela repose sur le fait que $f_-(\Phi) = f_-(\Psi)$ entraîne $\Phi = \Psi$.

On vérifie ensuite, par induction sur la taille du graphe, que $f_-(\Phi)$ est logiquement équivalent à $\neg\Phi$, *i.e.*, les formules associées sont logiquement opposées. Cela repose sur le fait que $(A \Rightarrow \neg\phi) \wedge (\neg A \Rightarrow \neg\psi)$ est logiquement équivalent à $\neg((A \Rightarrow \phi) \wedge (\neg A \Rightarrow \psi))$.

Définition 2.4. Étant donnés deux BDDs Φ et Ψ tels que A soit inférieur à toutes leurs variables libres, on définit le BDD $(A \rightsquigarrow \Phi; \Psi)$ comme $(A \rightarrow \Phi; \Psi)$ si $\Phi \neq \Psi$, et Φ sinon.

Définition 2.5. Étant donnés deux BDDs Φ et Ψ , et une opération $*$ $\in \{\wedge, \vee, \Rightarrow\}$, on définit $f_*(\Phi, \Psi)$ comme suit :

- $f_*(b, \Psi)$ et $f_*(\Phi, b)$ (pour $b \in \{\top, \perp\}$) sont donnés de façon évidente selon l'opération considérée, par exemple $f_\vee(\top, \Psi) = \top$ et $f_\vee(\perp, \Psi) = \Psi$.
- $f_*((A \rightarrow \Phi_+; \Phi_-), (A \rightarrow \Psi_+; \Psi_-)) = (A \rightsquigarrow f_*(\Phi_+, \Psi_+); f_*(\Phi_-, \Psi_-))$.
- $f_*((A \rightarrow \Phi_+; \Phi_-), \Psi) = (A \rightsquigarrow f_*(\Phi_+, \Psi); f_*(\Phi_-, \Psi))$ quand $A < B$, où B est la racine de Ψ .

Cela nous donne bien un graphe de Shannon équivalent à $\Phi * \Psi$. Ce graphe est ordonné. Il est réduit uniquement grâce au passage de \rightarrow à \rightsquigarrow .

Exemple 2.6. Calculer, en appliquant les fonctions associées aux connecteurs logiques, le BDD associé à $((A \Rightarrow B) \wedge C) \Rightarrow A) \vee \neg C$, selon l'ordre $B < A < C$.