Diagramme de décision binaire - ROBDD
Rapport pour le projet Mathématique et Informatique de L3 de 2018-2019

Sébastien Palmer et Xavier Durand

Encadré par Sedki Boughattas

1$^{\text{er}}$ mai 2019

Table des matières

Introduction

Présentation du sujet

Ce document est un projet de Mathématiques et d'Informatique suivi et encadré par un enseignant chercheur à l'Université Paris Diderot.
Nous allons ici traiter des arbres de décisions binaires. Le projet se base sur l'article de Henrik Reif Andersen « *An Introduction to Binary Decision Diagrams* ».
Ce document traite de ce qu'est un arbre de décision binaire, et de la représentation de toutes expressions propositionnelles en un arbre de décision binaire. Enfin, il expose un algorithme permettant de construire l'arbre de décision binaire correspondant à une expression propositionnelle quelconque.

Qu'est ce qu'une expression propositionnelle

Dans un premier temps, on va présenter ce qu'est une expression propositionnelle.

Variable propositionnelle

Une variable propositionnelle correspond à une variable comme en Mathématiques. Cependant, son ensemble de définition correspond à l'ensemble $\{0, 1\}$, où ici on peut apparenter 0 à *faux* et 1 à *vrai*.
Pour lier différentes variables propositionnelles entre elles, on va introduire des connecteurs logiques.

Connecteurs logiques

On va présenter ici les cinq symboles logiques les plus utilisés en logique propositionnelle :

- la négation : on la notera $\neg$. Elle correspond à une fonction ne prenant qu'une expression propositionnelle en argument et renvoie vrai si et seulement si l'argument est faux.
- la conjonction : on la notera $\wedge$. Elle correspond à une fonction prenant deux expressions propositionnelles en argument et renvoie vrai si et seulement si les deux arguments sont vrais.

- la disjonction : on la notera $\vee$. Elle correspond à une fonction prenant deux expressions propositionnelles en argument et renvoie faux si et seulement si les deux arguments sont faux (donc renvoie vrai si au moins l'un des deux arguments est vrai).
- l'implication : on la notera $\Rightarrow$. C'est une fonction prenant deux expressions propositionnelles en arguments et renvoyant faux si et seulement si le premier argument est vrai et le deuxième est faux.
- l'équivalence : on la notera $\Leftrightarrow$. C'est une fonction prenant deux expressions propositionnelles en argument et renvoyant vrai si et seulement si les deux arguments ont la même valeur de vérité.

Voici les tables de vérités de chacun des connecteurs logiques :

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\Rightarrow$	0	1
0	1	1
1	0	1

$\Leftrightarrow$	0	1
0	1	0
1	0	1

FIGURE 1 – Tables de vérités des cinq connecteurs logiques

Ces connecteurs logiques ne sont bien évidemment pas les seuls à exister, mais ce sont ceux qu'on accepte dans notre algorithme qu'on présentera plus tard.

Expression propositionnelle

Une expression propositionnelle correspond à une suite de variables propositionnelles liées par des connecteurs logiques.
Voici un exemple d'expression propositionnelle :

$$(x \wedge y) \vee ((\neg z) \Rightarrow (x \Leftrightarrow t))$$

avec x, y, z et t des variables propositionnelles.

Par soucis de lisibilité, lorsqu'on écrira une expression propositionnelle, on ajoutera des parenthèses.
Prenons l'expression $x \wedge y \vee 1$. On ne va pas avoir la même table de vérité si on écrit $(x \wedge y) \vee 1$ (qui sera toujours vrai) et si on écrit $x \wedge (y \vee 1)$ (qui sera faux si x est faux).
Cependant, il y a des expressions qui sont commutatives, et donc on peut omettre les parenthèses dans ces cas précis. Ces expressions commutatives sont la succession de conjonction et la succession de disjonction. La table de vérité des deux expressions propositionnelles suivantes ne change pas quelque soit la position des parenthèses :

$$x \wedge y \wedge z \qquad et \qquad x \vee y \vee z$$

Qu'est ce qu'une ROBDD (présentation rapide)

Problématique

plan

Chapitre 1

Représentation sous ROBDD

1.1 Système complet de φ

1.2 démo de l'existence

1.3 démo de l'unicité

1.4 A quoi correspond le graphe, et rapport avec φ

1.5 Représentation correcte -> pas de perte d'informations

Chapitre 2

Construction d'une ROBDD

2.1 présentation de l'algorithme

2.2 Présentation de l'algo

2.3 exemple d'exécution de l'algo

2.4 Etude des complexités

Chapitre 3

Intérêt et optimisation

3.1 Raison de la ROBDD

3.2 Utilisation possible comme Sat-solveur mais mauvais

3.3 Théorie sur les ordres

Conclusion

Nouvelle approche d'une expression propositionnelle

Optimisation que cela apporte en fonction de l'ordre

Représentation simple d'une expression prop

Théorie développée et approche de recherche pour les ordres

Ce que le projet nous a apporté

Bibliographie

Ajouter les différents articles sur lesquels on s'est basé.