Université Paris 7 DESS Logiciels Fondamentaux
U.F.R d'informatique Année 2002-2003

TD 11 – Les BDDs

Les BDDs sont des representations symboliques pour des formules booleennes. Ils sont construit en utilisant les redondances dans les arbres de choix.

Etant donne une formule booleennes de variables libres $x_1..x_n$ et un ordre sur ces variables (ici on garde l'ordre des indices) Un **arbre de choix** est construit avec des noeuds etiquetes par des variables $\cup\{V, F\}$. La racine de cet arbre est x_1, les fils gauche et droit de x_i sont etiquetes par x_{i+1}, et ces deux arcs sont etiquete l'un par V, et l'autre par F. Les fils possibles de x_n sont V et F. Etant donne une feuille de cet arbre, le chemin qui l'amene à la racine permet de determiner un choix unique pour chacune des variables. On a construit un arbre de choix si l'etiquete de chaque feuille correspond à la valeur de verite de la formule pour la valuation associee au chemin qui la mene à la racine.

Le Bdd unique associe à un arbre de choix donne est obtenu en iterant les simplifications suivantes :

- si deux sous arbres quelconques sont identiques, alors un seul est conserve, et pour l'autre le pointeur vers ce sous arbre est oriente vers celui que l'on a conservé.
- Si Pour un noeud donné, le fils gauche et le fils droit sont identiques, alors on supprime ce noeud : son pere, conserve l'etiquete de l'arc qui le reliait a son fils, et devient directement relié à ses petits-fils.

Exercice 1 : [Echauffement] Pour l'arbre de choix de la figure 4.1 page 52 de Model Checking schnoebelen, quelle est la tailel du BDD associé ?

Pourquoi le nombre d'etat est une bonne mesure de la taille du BDD ?

Donner un BDD pour l'ordre $b_1, b_3, b_2, b4$

Exercice 2 : [Choix de l'ordre] On pose :

$$f^1(a_1, b_1) := a_1 \Leftrightarrow b_1$$
$$f^2(a_1, a_2, b_1, b_2) := (a_1 \Leftrightarrow b_1) \wedge (a_2 \Leftrightarrow b_2)$$
$$\dots\ f^n(a_1, a_2..a_n, b_1, b_2, ..b_n) = \dots$$

Construire un BDD pour f^2 avec l'ordre a_1, a_2, b_2, b_1, puis avec l'ordre a_1, b_1, a_2, b_2.

On veut generaliser, essayer de regarder la taille de f^3. Quel est l'ordre de grandeur du BDD pour f^n avec les ordres $a_1, b_1, a_2, b_2, ...a_n, b_n$ puis $a_1, a_2, a_3, ..a_n, b_n, ..., b_1$.

Exercice 3 : [BDD vers formule] Donner une formule logique pour le BDD de la page 54, figure 5.2 de Clarke. Plus generalement donner une regle recursive pour obtenir une formule à partir d'un BDD.

Exercice 4 : [Operations sur les BDD] Pour le Bdd de la figure 5.3 page 55 de clarke, effectuer les operations : $b_1 \leftarrow F$, $\exists b_1$.

Correction 1 : Le Bdd est sur la page suivante. Le Bdd est binaire : sauf pour les 2 feuilles, il y a deux fils.

Correction 2 : Pour l'ordre $a_1, b_1, \ldots$ la construction recursive est :

a_1 a deux fils b_1, chaque b_1 a un fils vers F de sorte que le chemin a_i, b_1, F ait des V, F ou F, V. les autres arcs sortant des b_1 vont vers le BDD de $f^2(a_2, a_3 \ldots a_n, b_1, \ldots b_n)$, qui a renomage pres est celui de la recursion inferieure. Bref la taille du BDD pour f^{n+1} est celle de f^n + 3 etats ajoutes, ca fait du lineaire.

Pour l'autre ordre : on construit un BDD partant de a_1, avec 2 fils. Chacun etant une copie du BDD pour f^{n-1}, sauf que pour chacune des copies, on a remplace T par un noeud b_1, dont l'un des fils est T et l'autre F, conformement au choix fait pour a_1.

Bref pour le rang n, on a besoin de 2 copies ce qui fait du 2^n. Reste a montrer qu'on ne peut pas simplifier. (D'ailleurs je ne l'ai pas fait dans le premier ordre, mais c'est facile, et ce n'etait pas vraiemnt necessaire, vu que je cherchais une borne sup lineaire).

Alors : chaque paire de sous arbre du BDD global est, soit a qq chose pres un sous arbre de la meme copie de f^{n-1}, et dans ce cas ne sont pas comparables puisque c'etait un BDD, soit ils sont chacun dans des copie differentes. Si c'est le cas il faut montrer qu'ils sont differents.

Bien sur ce n'est pas vrai si on regarde V ou F qui doivent fusionnner. Par contre, tout sous arbre non trivial contient un noeud b_1. Et puisque les sous arbres consideres sont dans des copies differentes de f^{n-1}, ils ont un comportement differents pour b_1 : ils ne sont pas identiques. On a bien construit un BDD non simplifiable. Il etait exponentiel.

J'ai rapelle que le choix du meilleur ordre est NP complet, et qu'il existe des familles de formules ou quelque soit l'ordre choisit , la taille est exponentielle. Donc ce n'est pas la panacée.

Correction 3 : Si j'ai un BDD : $x \rightarrow^T y$ et $x \rightarrow^F z$, alors je peut ecrire : $f(x) = x \wedge f(y) \vee \neg x \wedge f(z)$.

Correction 4 : L'affectation : il suffit d'orienter le pere p de b_1 vers le fils faux de b_1 avec l'etiquete de p a b_1. Le resultat est a renormaliser.

Pour $\exists b_1$: je reecrit $\exists b_1.\phi$ en $b_1 \wedge \phi(b_1 \leftarrow V) \vee \neg b_1 \wedge \phi(b_1 \leftarrow F)$. Puis il reste a expliquer l'union et l'intersection de BDD.

Si on considere $x \rightarrow T_1$, $x \rightarrow T_2$ et $y \rightarrow T_1'$, $y \rightarrow T_2'$, et l'intersection des 2 ensembles decrit par le BDD (c'est le et des formules).

Si $x \neq y$ on peut suposer $x < y$ dans l'ordre des variables, alors il faut propager les contraintes sur les variables + petites que y dans l'ensemble decrit par le BDD de racine x. On construit donc un Bdd de racine x, et dont les fils sont respectivement $T_1 \wedge T'$ et $T_2 \wedge T'$.

Si $x = y$ alors il faut que la propagqtion soit coherente avec le choix que l'on fait pour x. Le BDD resultat est donc de racine x et de fils $T_1 \wedge T'1$ et $T_2 \wedge T'2$.

Voila, ce TD a pris 1h40, avec les rappels.