

Master SESI [M1]

UE CAO – Examen

Alain GREINER

Mai 2011

Durée : 2 heures – Tous documents autorisés

Le barème est donné à titre indicatif et peut être modifié par le correcteur

Écrivez lisiblement, un texte difficilement déchiffrable sera toujours considéré comme faux

Utilisation des ROBDD pour la représentation des fonctions booléennes

Soit une fonction booléenne quelconque \mathcal{F} , représentée par un ROBDD. On cherche à exprimer la fonction \mathcal{F} comme une somme de produits.

Il y a en général plusieurs chemins entre le noeud BDD représentant la fonction \mathcal{F} et la constante **1**. Tous ces chemins n'ont pas la même longueur, mais chaque chemin correspond à un produit de variable (opérateur `and`). L'expression Booléenne construite en effectuant la somme (opérateur `or`) entre tous les produits associés aux différents chemins répond à la question.

Question 1

2pt

Dessiner graphiquement le multi-ROBDD permettant de représenter les deux fonctions Booléennes \mathcal{F}_1 et \mathcal{F}_2 associées aux expressions booléennes infixées suivantes :

$$\begin{aligned}\mathcal{F}_1 &= A \oplus B \oplus C \\ \mathcal{F}_2 &= (A + B + \overline{C}).(\overline{B} + C)\end{aligned}$$

On construira le graphe pour l'ordre de décomposition des variables $A > B > C$. On donnera, pour chaque noeud BDD du graphe ROBDD une expression booléenne décrivant la fonction Booléenne représentée par ce noeud. On indiquera, pour chaque arc, une étiquette (**0** ou **1**) qui précise si le noeud destination est cofacteur F_H ou F_L .

Chemins et sommes de produits

Un chemin entre le noeud BDD représentant la fonction \mathcal{F} et le noeud BDD représentant le noeud **1** est une suite d'arcs. Dans un ROBDD, tout arc est caractérisé par deux informations :

- La variable associée au noeud source.
- L'étiquette qui précise si le noeud destination est le cofacteur F_H ou F_L .

Pour réaliser l'association d'un arc à une variable on procède de la façon suivante : soit un noeud BDD d'index x correspondant à la variable X . L'arc sortant étiqueté **1** sera associé à X , l'arc sortant étiqueté **0** sera associé à \bar{X} .

Question 2

2pt

En appliquant la méthode décrite précédemment, exprimer les fonctions \mathcal{F}_1 et \mathcal{F}_2 sous forme d'une somme de produits.

Construction des chemins

On définit $\text{chemin}(\mathcal{F})$ comme l'ensemble des chemins entre le noeud BDD représentant la fonction \mathcal{F} et le noeud BDD représentant la fonction **1**. Chaque chemin représente un produit de variables, directes ou complémentaires.

Question 3

4pt

Écrire, en français, la relation de récurrence existant entre les ensembles $\text{chemin}(\mathcal{F})$, $\text{chemin}(F_H)$ et $\text{chemin}(F_L)$, où F_H et F_L sont les cofacteurs dans la décomposition de Shannon de la fonction \mathcal{F} par rapport à la variable X d'index le plus élevé dans le support de \mathcal{F} .

On étudiera les six cas terminaux, avant de traiter le cas général :

1. $F_H = 1$ et $F_L = 0$
2. $F_H = 0$ et $F_L = 1$
3. $F_H = 1$ et $F_L \neq 0$
4. $F_H = 0$ et $F_L \neq 1$
5. $F_L = 1$ et $F_H \neq 0$
6. $F_L = 0$ et $F_H \neq 1$

On utilisera les notations suivantes :

- $\{X\}$ est l'ensemble des chemins contenant un seul chemin. Ce chemin étant composé d'un unique arc entre le noeud BDD représentant la fonction $\mathcal{F} = X$ et le noeud BDD représentant la constante **1**.
- $\{\bar{X}\}$ est l'ensemble des chemins contenant un seul chemin. Ce chemin étant composé d'un unique arc entre le noeud BDD représentant la fonction $\mathcal{F} = \bar{X}$ et le noeud BDD représentant la constante **1**.
- $\{X.\text{chemin}(\mathcal{F})\}$ est l'ensemble de chemins obtenus en concaténant l'arc X en tête de tous les chemins appartenant à l'ensemble $\text{chemin}(\mathcal{F})$. L'ensemble $\{X.\text{chemin}(\mathcal{F})\}$ a donc le même nombre d'éléments que l'ensemble $\text{chemin}(\mathcal{F})$ mais la longueur de chaque chemin est augmentée d'une unité.
- Si \mathcal{E}_1 et \mathcal{E}_2 sont deux ensembles de chemins alors $\mathcal{E}_1 \cup \mathcal{E}_2$ est l'ensemble constitué de tous les chemins de \mathcal{E}_1 et de tous les chemins de \mathcal{E}_2

Implantation en C++

Question 4

2pt

Proposer une implantation de la classe `Edge` représentant un arc. Les attributs ne devront être accessibles par l'extérieur qu'au travers d'accesseurs et de modificateurs.

Question 5

4pt

On représentera un ensemble de chemins sous forme d'un `set` de pointeurs sur `list` d'`Edge`. La liste d'`Edge` (arcs) représentant un chemin. Faut-il un destructeur spécifique? (justifiez).

```
class Chemins {
private:
    // Variables Membres -- Attributs
    std::set< std::list<Edge*>* > _chemins; // Ensemble de chemins.
public:
    // Modifieurs
    Chemins& concat (Edge*);
    Chemins makeUnion(Chemins& other);
public:
    // Constructeur
    Chemins (Edge*);
}; // Fin de "Chemin".
```

Question 6

6pt

On rapelle l'API de la classe Bdd. Écrire l'implantation de la méthode `buildChemins()`.

```

class Bdd {
public:
    class Key {
    private:
        unsigned int _index;
        unsigned int _highIdent;
        unsigned int _lowIdent;
    public:
        inline Key ( unsigned int index
                    , unsigned int high
                    , unsigned int low );
        friend bool operator< ( const Key& lhs, const Key& rhs );
    };
private:
    static unsigned int _maxIdent;
    static unsigned int _maxStamp;
    static std::map<Key, Bdd*> _bdds;
private:
    unsigned int _ident; // Unicity identifier.
    unsigned int _index; // Decomposition variable index.
    Bdd* _high; // Low cofactor pointer.
    Bdd* _low; // High cofactor pointer.
public:
    static Bdd* ConstHigh;
    static Bdd* ConstLow;
private:
    Bdd ( unsigned index, Bdd* high, Bdd* low );
    ~Bdd ();
public :
    static Bdd* create ( unsigned index, Bdd* high, Bdd* low );
    static Bdd* apply ( OperatorType, Bdd*, Bdd* );
public:
    inline unsigned getIdent ();
    inline unsigned getIndex ();
    inline Bdd* getHigh ();
    inline Bdd* getLow ();
    inline Chemins buildChemins (); // A creer.
};

```