

Programmation Fonctionnelle - Projet de TP

Algorithmes de décision pour les formules propositionnelles

La question de savoir, étant donnée une formule quelconque p du calcul propositionnel, si p est une "tautologie" ((c'est-à-dire: p est vraie pour toutes valeurs de ses variables)) est un problème NP-complet.

Divers algorithmes permettent de résoudre ce problème, avec des méthodes d'optimisation qui peuvent augmenter l'efficacité de manière remarquable dans certains cas.

On peut classer sommairement les algorithmes de base en deux catégories:

- les méthodes purement syntaxiques: on cherche à déduire le résultat par application de règles de transformation formelles qui permettent de "réécrire" la formule de départ.
- les méthodes sémantiques: on procède en remplaçant les variables par les valeurs Vrai, Faux.

Bien entendu, on peut combiner les deux approches.

Nous proposons ici d'étudier l'implémentation en CaML d'un algorithme mixte, à la fois syntaxique et sémantique: la méthode de normalisation d'expressions conditionnelles.

On suppose que les formules propositionnelles sont construites en utilisant

- un ensemble de symboles de variables en nombre illimité, représentant des *propositions atomiques* ("variables booléennes"),
- les constantes booléennes Vrai, Faux,
- les opérateurs booléens classiques suivantes: not, and, or, \Rightarrow

Les propositions sont supposées écrites dans une syntaxe abstraite manipulable directement par CaML.

Algorithme de décision par normalisation d'expressions conditionnelles.

Etant donnée une proposition quelconque, on la transforme d'abord en une expression conditionnelle, construite avec un seul opérateur noté IF, et on « normalise » la IF-expression obtenue.

Puis, on essaye de trouver s'il existe une affectation des variables qui réfute cette proposition, c'est-à-dire qui donne la valeur Faux à la proposition. Si c'est le cas, la proposition n'est pas une tautologie, sinon c'en est une. Le processus est, la plupart du temps, plus rapide que sur la proposition initiale.

- (1) Les *IF-expressions* comportent l'ensemble de variables propositionnelles, les constantes Vrai, Faux, et sont construites en utilisant uniquement l'opérateur IF :

si p, q, r sont des IF-expressions, alors $\text{IF } p \text{ } q \text{ } r$ est une IF-expression (avec la signification usuelle "si p alors q sinon r ")

- (2) Transformation d'une proposition en *IF-expression*.

Toute proposition peut être transformée en une IF-expression, grâce aux propriétés suivantes (ici, le signe « = » représente l'équivalence logique entre expressions) :

$$\begin{aligned} \text{not } p &= \text{IF } p \text{ Faux Vrai} \\ p \Rightarrow q &= \text{IF } p \text{ } q \text{ Vrai} \\ p \text{ and } q &= \text{IF } p \text{ } q \text{ Faux} \\ p \text{ or } q &= \text{IF } p \text{ Vrai } q \end{aligned}$$

- (3) **Forme normale** d'une IF-expression.

Une IF-expression est dite sous *forme normale* si : soit elle est *atomique*, soit c'est une *constante booléenne*, soit elle est écrite sous la forme

$\text{IF } \text{var } p \text{ } q$, où *var* est une simple variable propositionnelle et p, q sont des IF-expressions normales.

On peut transformer toute IF-expression en forme normale, en utilisant la propriété suivante:

$$\text{IF } (\text{IF } x \text{ } y \text{ } z) \text{ } p \text{ } q = \text{IF } x \text{ } (\text{IF } y \text{ } p \text{ } q) \text{ } (\text{IF } z \text{ } p \text{ } q)$$

- (4) **Affectation partielle** des variables propositionnelles et **décision partielle**.

Une *affectation partielle* σ , consiste simplement à choisir des valeurs Vrai /Faux pour un ensemble donné de variables booléennes.

Etant données une proposition p et une affectation partielle σ , on dit qu'on effectue une *décision partielle* quand on cherche à savoir si p est vraie pour toute affectation des variables de p qui « complète » σ :

si oui, p est dite une *tautologie partielle*. Dans le cas contraire, il existe une affectation qui complète σ et pour laquelle p est fausse, on dit alors que cette affectation est une *réfutation* de p .

Proposition. Une proposition est une tautologie si et seulement si, en appliquant une décision partielle pour une affectation partielle vide, on vérifie qu'elle est une tautologie partielle.

Problème à résoudre. Dédire de la proposition précédente (triviale), un algorithme de décision pour le calcul propositionnel.

Les étapes pour résoudre ce problème sont les suivantes:

-(1). Ecrire une fonction *Transforme* qui transforme toute proposition en IF-expression.
Essayer de **justifier clairement** pourquoi l'algorithme termine bien.

-(2). Ecrire une fonction *Normalise* qui donne la forme normale d'une IF-expression quelconque.
Essayer de **justifier clairement** pourquoi l'algorithme termine et donne bien une forme unique.

-(3). Définir une fonction *Decpart* qui, étant donnée une IF-expression p et une affectation partielle σ , effectue une décision partielle. Dans le cas où p n'est pas une tautologie partielle, cette fonction de décision partielle fournit alors une *réfutation* de p .

En déduire une fonction *Decision*, qui vérifie qu'une proposition est une tautologie ou sinon fournit une réfutation de cette proposition.

Exemples tests.

On testera l'algorithme de décision propositionnelle sur les exemples suivants.

1. Le poisson d'avril

Le poisson d'avril a les caractéristiques suivantes:

- tout poisson d'avril qui ne nage pas en mer chaude a des rayures rouges
- tout poisson d'avril a des nageoires bleues ou n'a pas de rayures rouges
- les poissons d'avril qui vivent dans le corail ne mangent pas de crevettes
- un poisson d'avril mange des crevettes si et seulement s'il nage en mer chaude
- tout poisson d'avril qui a des nageoires bleues, nage en mer chaude et vit dans le corail
- tout poisson qui nage en mer chaude a des nageoires bleues.

Montrer que les poissons d'avril n'existent pas

2. Le club écossais

Le club écossais en question obéit à des règles très strictes:

- tout membre non-écossais porte des chaussettes rouges
- tout membre porte un kilt ou ne porte pas de chaussettes rouges
- les membres mariés ne sortent pas le dimanche
- un membre sort le dimanche si et seulement s'il est écossais
- tout membre qui porte un kilt est écossais et marié
- tout membre écossais porte un kilt

Montrer que ce club est vide.

Note importante.

Il sera tenu le plus grand compte de la présentation et de la **clarté** des explications fournies.

Par ailleurs, essayer de construire des fonctions dont le codage soit le plus compréhensible possible.

Formuler explicitement et clairement la codage des exemples tests et la manière de les utiliser.

Fourniture du travail

Vous m'enverrez 2 fichiers (un pour le logiciel de décision, l'autre pour un ensemble de tests comprenant nécessairement les 2 exemples donnés dans l'énoncé, dans mon directory `/home/paillet/public` en donnant à vos fichiers toutes les permissions (`chmod 777 <fichier>`).

M'avertir de l'envoi en me mailant à Jean-Luc.Paillet@cmi.univ-mrs.fr

Vous me fournirez aussi un petit mémoire explicatif.