

# Informatique théorique et automates

## L3 CDA

### TP N<sup>o</sup> 1 : BDDC

Ce TP est à faire sous Linux.

## 1 bddc

Nous allons utiliser un logiciel appelé **bddc** conçu par Pascal Raymond du laboratoire VERIMAG. Comme son nom l'indique **bddc** est basé sur les diagrammes de décision binaires (bdd en anglais).

Lancer **bddc** en tapant en ligne de commande :

```
/home/commun_depinfo/enseignants/guillou/L3CDA/LogiqueEtAutomates/bddc-linux64/bddc
```

Si tout se passe normalement vous aurez l'invite suivante :

```
version 2.01
Type "help" for informations
>>
```

On peut alors saisir une expression du calcul propositionnel **terminée par un point-virgule** ou quitter la session en tapant **quit**.

Par exemple :

```
>>(a+b).c;
```

donnera :

```
a.c + b.c
```

Les identificateurs **a**, **b** et **c** n'ont pas à être déclarés. Les connecteurs se notent des façons suivantes :

Connecteur	se note	abréviation
conjonction $\wedge$	and	.
disjonction $\vee$	or	+
ou exclusif $\oplus$	xor	<>
négation $\neg$	not	-
implication $\implies$	=>	=>
équivalence $\iff$	=	=
vrai	true	1
faux	false	0

Pour déclarer une variable il suffit de faire :

```
f := a+b;
```

Par contre on ne peut affecter un identificateur déjà utiliser (comme **a**). Pour avoir la liste des identificateurs présents dans l'environnement taper **list**.

Sinon il est également possible de déclarer des fonctions, par exemple :

```
>>ou(a,b) := a+b;
--new function: ou, arity: 2
>>ou(0,1);
1
```

## 2 Un peu de syntaxe

### 2.1 Premiers pas

Évaluez les expressions suivantes :

1.  $\text{false or } p$  ;
2.  $1 \text{ or } p$  ;
3.  $p \Rightarrow p$  ;
4.  $(p=q)=r$  ;
5.  $f := (p=q)=r$  ;
6.  $f$  ;

### 2.2 Priorités

1.  $a+b.c$  et  $(a+b).c$  sont elles équivalentes ? On peut utiliser **compare**( $p_1, p_2$ ) pour tester si les propositions  $p_1$  et  $p_2$  sont égales. Donner une valuation pour laquelle ces formules ont des valeurs de vérités différentes.
2. Comment se lit  $a \Rightarrow -b.c + a = b$  (parenthéser la formule et vérifier avec **bddc** que c'est bien équivalent) ?
3. De quel côté associe-t-on  $a \Rightarrow b \Rightarrow c$  (à vérifier avec **bddc**) ?

## 3 Sémantique

### 3.1 Interprétation

Évaluer grâce à **bddc** les formules suivantes par rapport à la valuation  $\{a = 1, b = 0, c = 0, d = 1\}$  puis pour celles qui sont satisfaites trouver une valuation qui ne les satisfait pas :

1.  $(a \oplus c) \wedge (c \implies (\neg d \wedge b))$
2.  $\neg(a \vee (a \implies b) \wedge \neg(b \vee c \vee d))$
3.  $(a \vee (\neg a \wedge d)) \oplus (\neg d \implies a)$

### 3.2 Tables de vérité et arbres de Shannon

Construire à l'aide de `bddc` les tables de vérité et les arbres de Shannon de :

1.  $\neg, \vee$  et  $\implies$ .
2.  $\neg(\neg p \wedge q) \vee \neg(q \vee \neg r)$ .
3.  $((\neg p \wedge q \wedge r) \vee p) \wedge \neg r$ .

### 3.3 Diagrammes de décisions binaires

Dans `bddc` une expression est assimilée à un diagramme de décision binaire, c'est-à-dire un graphe enraciné (on dit *pointé* aussi), orienté, acyclique et pour lequel chaque sommet a exactement 2 ou aucun arc sortant. La fonction `root` donne la racine du graphe, `low` donne le sous-graphe "gauche" de l'expression, c'est-à-dire le sous-graphe atteint lorsqu'on affecte la valeur 0 à la racine. `high` donne le sous-graphe "droit". Construire les diagrammes de décisions binaires associés aux expressions suivantes en s'aidant de `bddc` :

1.  $\neg, \vee$  et  $\implies$ .
2.  $\neg(\neg p \wedge q) \vee \neg(q \vee \neg r)$ .
3.  $((\neg p \wedge q \wedge r) \vee p) \wedge \neg r$ .

### 3.4 Tautologies et antilogies

Parmi les expressions suivantes, lesquelles sont des tautologies et lesquelles sont des antilogies? Pour chaque expression qui n'appartient à aucune de ces deux catégories, trouvez une assignation des variables propositionnelles qui rende l'expression vraie (c'est ce qu'on appelle un *modèle* de la formule) ainsi qu'une deuxième qui la rende fausse.

1.  $p \wedge (p \implies q) \implies q$ .
2.  $(p \vee q) \wedge (p \implies r) \wedge (q \implies r) \implies r$ .
3.  $(a \vee b \vee c \vee d) \wedge (a \implies e) \wedge (b \implies e) \wedge (c \implies e) \wedge (d \implies e) \implies e$ .
4.  $a \wedge \neg a \implies b$ .
5.  $(p \iff (q \iff r)) \iff ((p \iff q) \iff r)$ .
6.  $(p \implies (q \implies r)) \iff ((p \implies q) \implies r)$ .
7.  $(p \implies q) \wedge (q \implies r) \wedge (r \implies \neg p)$ .
8.  $(p \implies (q \implies r)) \implies (p \wedge q \implies r)$ .
9.  $p \wedge ((p \implies q) \wedge \neg q)$ .
10. Vérifier les théorèmes de De Morgan et d'absorption.