

Definitions et props

Définition 1: Commutatif les variables peuvent etre inverses

Définition 2: L'arbre de Derivation C'est un format de pour represente une proposition

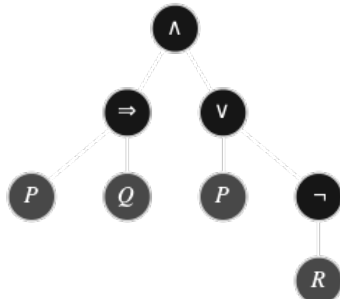


Figure 1: $(P \Rightarrow Q) \wedge (P \vee \neg R)$

Définition 3: Loi de De Morgan Soit P et Q deux assertions, alors
 $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
 $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

Tables de verite

il est assume qu'un connecteur est commutatif sauf mentionne autrement

table de \wedge : q binaire

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

table de \vee : q binaire

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

table de \oplus : q binaire

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

table de \Rightarrow : q binaire dit non commutatif

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

autrement dit, vrai sauf si p est vrai et q est faux

table de \Leftrightarrow : q binaire

⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥
⊥	⊥	⊥

vrai si les deux variables ont la meme valeur

Proprietes

- comutativite de \wedge et \vee

$$(p \wedge q) \equiv (q \wedge p)$$

$$(p \vee q) \equiv (q \vee p)$$

- associativite de \wedge et \vee

$$((P \wedge Q) \wedge R) \equiv ((Q \wedge R) \wedge P)$$

$$((P \vee Q) \vee R) \equiv ((Q \vee R) \vee P)$$

- idempotence de \wedge et \vee

$$(p \wedge p) \equiv p$$

$$(p \vee p) \equiv p$$

TPs

Question 1: Ecrire une fonction `interpretations(nbVar)` qui renvoie le tuple constitue de toutes les interpretations possible de nbvar variables propositionnelles

ici la strategie est d'imiter ce tableau en python

	v	f	v
f	f	f	v
v	v	v	v
f	v	f	v

qui, rempli, donne toutes les possibilites des variables

```
def interpretations(nbvar):
    vrai = [vrai for i in range(nbvar)]
    faux = [faux for i in range(nbvar)]
```

Question 2.

Une formule propositionnelle FP de n variables esst codee par une chiande de caracteres respectant la syntaxe python. les variables étant toujours codées $V[0]$, $V[1]$,... , $V[n-1]$. Écrivez une fonction `TV(FP,n)` qui renvoie la table de vérité de la formule FP sous forme de tuple de tuples à l'aide de la fonction `Inter` et la fonction d'évaluation

`eval(chaine)` du Python qui évalue une chaîne de caractères si elle respecte la syntaxe du langage Python.

Exemple. Avec la chaîne de caractère `FP = "V[0] and V[1]"`, l'appel de la fonction `TV(FP,2)` doit renvoyer le tuple
`((False,False,False),(False,True,False),`
`(True,False,False),(True,True,True))`

Predicats

Définition 4: Predicat énonce contenant des variables tel qu'en substituant chaque variables par une valeur choisi, on obtient une proposition

exemple: $x|P(x)$ (se lit x tel que $P(x)$) est un predicat dans lesquelles la proposition $P(x)$ est vraie pour x

Quantificateurs

Axiomes

]

bases et codage

rappels

decimale	binaire	octal	hexa
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Arithmetique tronque a gauche:

logique combinatoire

Définition 5: tableau de Karnaugh

Définition 6: forme nominal disjonctive (FND)

logique sequentielle

Conception d'algorithme

Définition 7: Invalidier un algo Pour montrer qu'un algo n'est pas valide, il suffit de montrer un contre exemple, soit un cas ou l'algorithme ne marcherait pas

Analyse asymptotique

Définition 8: Analyser un algorithme c'est analyser les couts par rapport au temps d'exécution, l'espace memoire, et la consommation électrique

Définition 9: le modele random access machine machine hypothetique ou:

- les operands consomment une unite de temps
- les boucles depend du nombre d'iterations et des operation inside
- un read consomme une unite de temps
- la memoire est illimite

l'efficacite d'un algo est defini par une fonction notee $C(n)$ ou $T(n)$, meme si dans un cas reel ca serait plutot note $O(n)$

exemple:

- recherche d'un element:
 - n cases a tester
 - 5 cases: > 5 tests
 - 10 cases: > 10 tests
- ramassage de plots:
 - $n!$ chemins a tester
 - 5 plots: 120 chemins possible

la notation est qui suit:

- $\Omega(n)$: meilleur cas
- $O(n)$: pire cas
- $\Theta(n)$: cas moyen

Bases d'algo

Algos de tri

Algos de recherche

pires et files

Fiches de revisions 2nd semestre

Brouillon

Fiches de revisions 2nd semestre

Brouillon