Text

Description automatically generated with medium confidence

LOG3430 -Méthodes de test et de validation du logiciel

TP2 – Fonctions logiques

Groupe 1

Équipe 7

Hugo Lachieze-Rey (1934177)

Dimitry Kamga (1898357)

Remis à :

Hanane Ikhelef

Hiver 2022

Table des matières

[1 Fonctions logiques 3](#_Toc95819453)

[2 Jeux de tests 4](#_Toc95819454)

[2.1 Tests ACC 4](#_Toc95819455)

[2.2 Tests ICC 6](#_Toc95819456)

[2.3 Tests DNF 8](#_Toc95819457)

# Fonctions logiques

Avant de commencer, il est nécessaire de dresser la table de vérité de l’équation logique utiliser pour classifier les messages :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ligne** | **P** | **H** | **U** | **G** | **H \* U** |  |  |  | **S** |
| 1 | T | T | T | T | T | F | F | T | T |
| 2 | T | T | T | F | T | T | T | T | T |
| 3 | T | T | F | T | F | F | F | F | F |
| 4 | T | T | F | F | F | T | F | F | F |
| 5 | T | F | T | T | F | F | F | F | F |
| 6 | T | F | T | F | F | T | T | T | T |
| 7 | T | F | F | T | F | F | F | F | F |
| 8 | T | F | F | F | F | T | F | F | F |
| 9 | F | T | T | T | T | F | F | T | F |
| 10 | F | T | T | F | T | T | T | T | F |
| 11 | F | T | F | T | F | F | F | F | F |
| 12 | F | T | F | F | F | T | F | F | F |
| 13 | F | F | T | T | F | F | F | F | F |
| 14 | F | F | T | F | F | T | T | T | F |
| 15 | F | F | F | T | F | F | F | F | F |
| 16 | F | F | F | F | F | T | F | F | F |

# Jeux de tests

## Tests ACC

GACC :

P : (1,9), (1,10), (1,14), (2,9), (2,10), (2,14), (6,9), (6,10), (6,14)

H : (1, 5)

U : (1,3), (1,4), (1,8), (2,3), (2,4), (2,8), (6,3), (6,4), (6,8)

G : (5, 6)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P** | **H** | **U** | **G** | **S** |
| **1** | T | T | T | T | T |
| **3** | T | T | F | T | F |
| **5** | T | F | T | T | F |
| **6** | T | F | T | F | T |
| **9** | F | T | T | T | F |

CACC :

P : (1,9), (1,10), (1,14), (2,9), (2,10), (2,14), (6,9), (6,10), (6,14)

H : (1, 5)

U : (1,3), (1,4), (1,8), (2,3), (2,4), (2,8), (6,3), (6,4), (6,8)

G : (5, 6)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P** | **H** | **U** | **G** | **S** |
| **1** | T | T | T | T | T |
| **3** | T | T | T | F | F |
| **5** | T | F | T | T | F |
| **6** | T | F | T | F | T |
| **9** | F | T | T | T | F |

→ image de notre jeu de test

RACC :

P : (1,9), (2,10), (6,14)

H : (1, 5)

U : (1,3), (2,4), (6,8)

G : (5, 6)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P** | **H** | **U** | **G** | **S** |
| **1** | T | T | T | T | T |
| **3** | T | T | F | T | F |
| **5** | T | F | T | T | F |
| **6** | T | F | T | F | T |
| **9** | F | T | T | T | F |

## Tests ICC

On distingue GICC et RICC :

* GICC : Aucune restriction sur les valeurs des cj ;
* RICC : Les cj doivent rester les mêmes quand ci est évalué àvrai et quand ci est évalué à faux.

GICC :

Nous utilisons la même méthode que vue en classe (chaque clause devient majeure) mais nous la regroupons sous forme de tableau pour une meilleure présentation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Ci true / S = F** | **Ci true / S = T** | **Ci false / S = F** | **Ci false / S = T** |
| **P** | 7 | - | 12 | - |
| **H** | 12 | 2 | 7 | 6 |
| **U** | 5 | - | 7 | - |
| **G** | 3 | 1 | 12 | 2 |

Lorsque la clause P est majeure, il n’existe pas de combinaisons de H, U et G permettant à P d’être inactive et à S d’être vraie.

Lorsque la clause U est majeure, il n’existe pas de combinaisons de P, H et G permettant à P d’être inactive et à S d’être vraie.

Nous avons donc les paires :

P : -, (7, 12)

H : (12, 7), (2, 6)

U : -, (5, 7)

G : (3, 12), (1, 2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P** | **H** | **U** | **G** | **S** |
| **1** | T | T | T | T | T |
| **2** | T | T | T | F | T |
| **3** | T | T | F | T | F |
| **6** | T | F | T | F | T |
| **7** | T | F | F | T | F |
| **12** | F | T | F | F | F |

CICC :

Nous utilisons la même méthode que vue en classe (chaque clause devient majeure) mais nous la regroupons sous forme de tableau pour une meilleure présentation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Ci true / S = F** | **Ci true / S = T** | **Ci false / S = F** | **Ci false / S = T** |
| **P** | 4 | - | 12 | - |
| **H** | 3 | 2 | 7 | 6 |
| **U** | 5 | - | 7 | - |
| **G** | 3 | 1 | 4 | 2 |

Lorsque la clause P est majeure, il n’existe pas de combinaisons de H, U et G permettant à P d’être inactive et à S d’être vraie.

Lorsque la clause U est majeure, il n’existe pas de combinaisons de P, H et G permettant à P d’être inactive et à S d’être vraie.

Nous avons donc les paires :

P : -, (4, 12)

H : (3, 7), (2, 6)

U : -, (5, 7)

G : (3, 4), (1, 2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P** | **H** | **U** | **G** | **S** |
| **1** | T | T | T | T | T |
| **2** | T | T | T | F | T |
| **3** | T | T | F | T | F |
| **4** | T | T | F | F | F |
| **6** | T | F | T | F | T |
| **7** | T | F | F | T | F |

Nous aurions pu choisir des jeux de tests couvrant à la fois RICC et GICC car par définition, RICC couvre également GICC. Comme nous l’avons mentionné plus haut, la seule différence se fait au niveau des restrictions sur les clauses cj lorsque c­i varie (vrai ou faux).

## Tests DNF

Voici la forme DNF de notre fonction logique :

Négation de S :

Nous avons donc 5 impliquants :

On peut dresser la table de vérité de la négation de S car elle nous sera utile pour les jeux de tests :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ligne** | **P** | **H** | **U** | **G** |  | **G** |  |  |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | F |
| 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | F |
| 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | T |
| 4 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | T |
| 5 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | T |
| 6 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | F |
| 7 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | T |
| 8 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | T |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | T |
| 10 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | T |
| 11 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | T |
| 12 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | T |
| 13 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | T |
| 14 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | T |
| 15 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | T |
| 16 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | T |

Pour le critère IC, notre code montre le jeu de test suivant :

→ image de notre jeu de test

Pour le critère IC, il est nécessaire de trouver les cas de tests ou chaque impliquant est évalué à vrai. On peut voir sur notre figure ci-dessous que c’est bel est bien le cas.

Pour le critère PIC, il faut que chaque impliquant soit évalué à vrai pendant que tous les autres sont évalués à faux.

Le jeu de tests suivant satisfait le critère PIC :

* d1 = <{P=true, H=true, U=true, G=true}, {true}>
  + PHU=true, PU = false, =false, = false, = false
* d2 = <{P=true, H=false, U=true, G=false}, {true}>
  + PHU=false, PU = true, =false, = false, = false
* d3 = <{P=false, H=false, U=true, G=false}, {false}>
  + PHU=false, PU = false, =true, = false, = false
* d4 = <{P=true, H=false, U=true, G=true}, {false}>
  + PHU=false, PU = false, =false, = true, = false
* d5 = <{P=true, H=true, U=false, G=true}, {true}>
  + PHU=false, PU = false, =false, = false, = true

Pour la stratégie de négation de variables (VNS), il est nécessaire de trouver les points uniques vrais (PUV) et les points presque faux (PPF) de la fonction logique Z :