

INFO0947: Construction de programme

Groupe 27: Alexandru DOBRE, Sami OUAZOUZ

Table des matières

1	Introduction	3
2	Formalisation du Problème	3
2.1	Utilisez les bons opérateurs	3
2.2	Trouver un symbole précis	3
3	Définition et Analyse du Problème	3
4	Specifications	3
5	Invariants	3
6	Approche Constructive	3
7	Code Complet	4
8	Complexité	4
9	Conclusion	4

1 Introduction

2 Formalisation du Problème

2.1 Utilisez les bons opérateurs

Voir la table 1.

Nom	Op
ET	\wedge
OU	\vee
Quantification universelle	\forall
Quantification existentielle	\exists

TABLE 1 – Opérateurs les plus usuels en logique

2.2 Trouver un symbole précis

Voir ce site : <http://detexify.kirelabs.org/classify.html>. Il suffit de dessiner le symbole dont vous avez besoin et le site trouvera (normalement) la bonne commande à taper (ainsi que le package à éventuellement inclure si besoin est).

3 Définition et Analyse du Problème

4 Specifications

5 Invariants

Pour inclure vos Invariants Graphique dans le rapport, nous vous rappelons que l’outil GLIDE (<https://cafe.uliege.be>) permet d’exporter au format PDF vos dessins d’Invariants.

6 Approche Constructive

```
1 int main(void)
2 {
3     // Les commandes Latex sont permises dans les commentaires sur une ligne. Exemple :  $x_i \leq a^b$ 
4     printf("Bonjour tout le monde !");
5     /*
6     Dans les commentaires sur plusieurs lignes, elles doivent être entourées
7     de symboles définis par l'option « escapeinside » de \lstset
8      $\sum_{i=1}^N 1 = N$ 
9     La commande « \coms » permet de colorer correctement le code latex ajouté.
10    Les accents et tous les autres diacritiques sont permis : àÀçÇéÊëËêËœœ...
11    */
12    return 1;
13 }
```

Extrait de Code 1 – Un programme tout simple

Il est possible de faire référence à la ligne 12 de l’extrait de code.

7 Code Complet

8 Complexité

9 Conclusion