

# Projet LO21: Rapport final

Adrien Burgun

Automne 2020

## Résumé

Le projet de ce semestre pour le cours de **LO21** (Algorithmique et Programmation II) porte sur un « *système expert* ». Un système expert est constitué de 3 éléments :

**Une base de connaissance**, qui prend la forme suivante :

$$A \wedge B \wedge \dots \wedge Z \Rightarrow \Omega$$

Où  $A, B, \dots$  sont les symboles (d'arité zéro, aussi appelés « propositions ») constituant la *prémisse* et  $\Omega$  est la *conclusion*.

**Une base de faits**, qui est la liste des symboles ayant la valeur « *Vrai* » (qui correspond à l'état « *Certain* »).

Un symbole ne faisant pas partie de cette liste a par défaut la valeur « *Faux* » (qui correspond à l'état « *Incertain* »).

**Un moteur d'inférence**, qui, à partir de la base de connaissance et la base de faits, déduit quels autres symboles sont aussi vrais et les ajoute à la base de faits.

Nous définirons d'abords le type « *Règle* », constituant la base de connaissance. Nous définirons ensuite le type « *BC* » (Base de Connaissance).

Nous décrirons enfin le moteur d'inférence comme implémenté dans ce projet, avec différents exemples.

## 1 Règles

Soit *regle\_t* le type représentant une règle sous la forme d'une liste chaînée de symboles :

L'implémentation faite dans ce projet utilise des noms anglais pour les variables, fonctions et types ; le type *Règle* a pour équivalent C le type *rule\_t*.

Le dernier élément d'une telle liste chaînée correspond à la conclusion de la règle, tandis que tous les autres éléments appartiennent à la prémisse.

## 1.1 Créer une règle vide

Nous représenterons une règle vide par un pointeur nul. Voici l'algorithme permettant de créer une règle vide :

Algorithme 1 : NouvelleRègle	
<b>Variables :</b> $R$ : La règle vide à retourner	
<b>Résultat :</b> $R$ : Règle	
1	<b>Début</b> NouvelleRègle
2	$R \leftarrow \text{NULL}$
3	<b>Fin</b>

## 1.2 Créer un noeud d'une règle

Voici la fonction permettant de créer un noeud (ou symbole) dans une règle :

Algorithme 2 : CréerNoeud	
<b>Variables :</b> $R$ : Le noeud de la règle à retourner	
$symbole$ : La valeur de $symbole$ à mettre dans le noeud	
$suivant$ : Pointeur sur le prochain noeud de la liste	
<b>Résultat :</b> $R$ : Règle	
1	<b>Début</b> CréerNoeud
	// Nous supposons que l'espace de mémoire requis pour $R$ est déjà alloué
2	$(R \rightarrow symbole) \leftarrow symbole$
3	$(R \rightarrow suivant) \leftarrow suivant$
4	<b>Fin</b>

## 1.3 Ajouter une proposition à la prémisse d'une règle

L'ajout des propositions (symboles) à la prémisse d'une règle se fait par l'algorithme *AjoutPrémisse* défini ci-dessous. Cet ajout se fait en queue de la liste chaînée (contrainte donnée pour ce projet). La liste chaînée donnée en entrée est modifiée par l'algorithme

et est retournée.

#### Algorithme 3 : AjoutPrémisse

**Variables :**  $R$ : La règle à modifier  
 $R'$ : Une variable temporaire pour traverser la liste chaînée  
 $symbole$ : Le nom de la proposition (symbole) à insérer  
**Données :**  $R$ : Règle,  $symbole$ : Chaîne de caractères  
**Résultat :**  $R$ : Règle

```

1 Début AjoutPrémisse
2   Si  $R = NULL$  alors
3      $R \leftarrow \text{CréerNoeud}(symbole, NULL)$ 
4   Sinon
5      $R' \leftarrow R$ 
6     // Répéter jusqu'à ce qu'on atteigne le dernier élément
7     Tant que  $R' \rightarrow next = NULL$  faire
8        $R' \leftarrow R' \rightarrow next$ 
9     Fin
10    //  $R'$  contient désormais le dernier élément de la liste
11     $(R' \rightarrow \text{suivant}) \leftarrow \text{CréerNoeud}(symbole, NULL)$ 
12  FinSi
13 Fin

```

## 1.4 Créer la conclusion d'une règle

Créer la conclusion d'une règle revient à ajouter une proposition (symbole) à la fin de la règle. Pour ce faire, nous ré-utilisons l'algorithme [AjoutPrémisse](#) défini plus tôt.

#### Algorithme 4 : AjoutConclusion

**Variables :**  $R$ : La règle à modifier  
 $symbole$ : Le nom de la proposition (symbole) à insérer comme conclusion  
**Données :**  $R$ : Règle,  $symbole$ : Chaîne de caractères  
**Résultat :**  $R$ : Règle

```

1 Début AjoutConclusion
2    $R \leftarrow \text{AjoutPrémisse}(R, symbole)$ 
3 Fin

```