LO21 A2020 Projet – Dewasmes Oscar

Titre

I) Introduction

Ce projet a été réalisé dans le cadre de l’UV LO21 enseignée à l’Université de Technologie de Belfort Montbéliard, au semestre d’automne 2020.

L’objectif du projet est d’implémenter un système expert en langage C. Le système expert est constitué d’une base de connaissance, une base de fait et d’un moteur d’inférence. La base de connaissance est définie par ses propositions et ses règles qui mettent en relation les propositions.

II) Implémentation du projet

Pour réaliser ce projet, j’ai décidé d’implémenter les structures C suivantes :

|  |  |  |  |
| --- | --- | --- | --- |
| Une liste chainée d’entiers sans répétitions | | | |
| **Nom Structure :** ListEntier | | **Nom dans le code :** i\_list | |
| **Nom valeurs** | **Type** | | **Description** |
| Valeur | Entier | | La valeur de l’élément de la liste |
| Suivant | Pointeur sur ListEntier | | Désigne l’élément suivant de la liste |

|  |  |  |  |
| --- | --- | --- | --- |
| Une liste chainée de propositions | | | |
| **Nom Structure :** Proposition | | **Nom dans le code :** s\_list | |
| **Nom valeurs** | **Type** | | **Description** |
| Id | Entier | | Identifiant de la proposition |
| Description | String | | Description de la proposition |
| Suivant | Pointeur sur Proposition | | Désigne l’élément suivant de la liste |

|  |  |  |  |
| --- | --- | --- | --- |
| Une liste chainée de règles | | | |
| **Nom Structure :** Règle | | **Nom dans le code :** r\_list | |
| **Nom valeurs** | **Type** | | **Description** |
| Id | Entier | | Identifiant de la règle |
| Prémisse | ListEntier | | Liste des propositions de la prémisse |
| Conclusion | Entier | | Identifiant de la proposition conclusion |
| Suivant | Pointeur sur Règle | | Désigne l’élément suivant de la liste |

|  |  |  |  |
| --- | --- | --- | --- |
| Structure de la Base de connaissance | | | |
| **Nom Structure :** BDC | | **Nom dans le code :** Database | |
| **Nom valeurs** | **Type** | | **Description** |
| Nom | String | | Nom de la BDC |
| Propositions | ListProposition | | Toutes les propositions de la BDC |
| Règles | ListRègle | | Toutes les règles de la BDC |

III) Algorithmes

|  |
| --- |
|  |
| Algorithme : |
| **Nom Fonction : Données**  **Résultat**  **Lexique :**  **Début**  **Fin** |

|  |
| --- |
| -R.Premisse est de type ListEntier, ce type de liste est considéré vide lorsqu’il est indéfinit |
| Algorithme : Créer une règle vide |
| **Nom Fonction :** RègleVide  **Données**  **Résultat** R : Règle  **Lexique :**  **Début**  R ← Allouer Mémoire pour Règle  R.id ← Indéfinit  R.Prémisse ← Indéfinit  R.Conclusion ← Indéfinit  R.Suivant ← Indéfinit  **Fin** |

|  |
| --- |
| -P est la liste de toutes les propositions de la BDC  -EstDansListProp(Proposition, Entier) est une fonction qui retourne vrai si l’identifiant donné est dans la liste de propositions donnée  -IsérerQueue(ListEntier, Entier) est une fonction qui insère à la liste d’entiers donnée l’entier donné en queue seulement si il n’est pas déjà dans la liste |
| Algorithme : Ajouter une proposition à la prémisse d’une règle, l’ajout se fait en queue |
| **Nom Fonction :** InsérerPremisse  **Données** R : Règle, P : Proposition, IdProp : Entier **Résultat** R : Règle  **Lexique :**  **Début**  Si EstDansListProp(P, IdProp) Alors  R.Prémisse ← InsérerQueue(R.Prémisse, IdProp)  Sinon  Annoncer Erreure  Fin si  **Fin** |

|  |
| --- |
| -P est la liste de toutes les propositions de la BDC  -EstDansListProp(Proposition, Entier) est une fonction qui retourne vrai si l’identifiant donné est dans la liste de propositions donnée |
| Algorithme : Définir la conclusion d’une règle |
| **Nom Fonction :** DéfinirConclusion **Données** R : Règle, P : ListProposition, IdProp : Entier **Résultat** R : Règle  **Lexique :**  **Début**  Si EstDansListProp(P, IdProp) Alors  R.Conclusion ← IdProp  Sinon  Annoncer Erreur  Fin si  **Fin** |

|  |
| --- |
| -P est la prémisse de la règle à tester  -ValeurTête(ListEntier) est une fonction qui retourne la valeur du premier élément de la liste d’entiers donnée  -Reste(ListEntier) est une fonction qui retourne la liste d’entiers donnée privée de son élément de tête |
| Algorithme : Teste si une proposition appartient à la prémisse d’une règle, de manière récursive |
| **Nom Fonction :** EstDansListEntier **Données** P : ListEntier, IdProp : Entier **Résultat** B : Booléen  **Lexique :**  **Début**  Si P = Indéfinit Alors  B ← Faux  Sinon Si ValeurTête(P) = IdProp Alors  B ← Vrai  Sinon  B ← EstDansListEntier(Reste(P), IdProp)  Fin Si  **Fin** |

|  |
| --- |
| - RemoveFromListEntier(ListEntier, Entier) est un fonction qui supprime un entier d’une liste d’entiers |
| Algorithme : Supprimer une proposition de la prémisse d’une règle |
| **Nom Fonction :** SuppDePremisse **Données** R : Règle, IdProp : Entier **Résultat** R : Règle  **Lexique :**  **Début**  R.Prémisse = RemoveFromListEntier(R.Prémisse, IdProp)  **Fin** |

|  |
| --- |
| -R.Premisse est de type ListEntier, ce type de liste est considéré vide lorsqu’il est indéfinit |
| Algorithme : Tester si la prémisse d’une règle est vide |
| **Nom Fonction :** PremisseEstVide **Données** R : Règle **Résultat** B : Booléen  **Lexique :**  **Début**  B ← R.Premisse = Indéfinit  **Fin** |

|  |
| --- |
| -P est la liste de toutes les propositions de la BDC |
| Algorithme : Accéder à la proposition se trouvant en tête d’une prémisse |
| **Nom Fonction : Données** R : Règle, P : ListProposition  **Résultat** out : Proposition  **Lexique :**  Id : Entier  **Début**  Id ← HeadValue(R.Premisse)  Tant Que P ≠ Indéfinit Et Identifiant(P) ≠ Id Faire  P ← Rest(P)  Fin Tant Que  Si P = Indéfinit Alors  Annoncer Erreur (id d’une proposition qui n’existe pas dans la prémisse)  Sinon  Out ← P  Fin Si  **Fin** |

IV) Jeux d’essais

V) Conclusion