

IA1 : Système Expert

Théo Dézé Charles Mallet

9 Novembre 2018

Table des matières

I	Introduction	2
1	Faits	2
2	Règles	2
3	Choix du langage	3
II	Système Expert	3
4	Base de connaissance	3
4.1	Fichier	4
4.2	Interface	4
5	Moteur d'inférence	4
6	Interface utilisateur	4
6.1	Ligne de commande	5
6.2	Graphique	5
III	Conclusion	6
7	Améliorations possibles	6
8	Le code source du projet	6

Première partie

Introduction

Pour le projet nous avons décidé de travailler sur un problème de diagnostique médical. Nous avons défini différentes maladies qui sont a vrais si le patient les a.

1 Faits

Nos faits sont de trois nature soit ce sont des informations sur les symptômes du patient, soit sur ces attributs ou sur les maladies qu'il a.

Comme nous sommes sur un moteur d'inférence 0+ nos faits peuvent valoir une valeur booléenne, une chaîne de caractère ou un chiffre (pas dans ce cas mais on peut le tester dans cas météorologie¹).

Exemple

Symptômes La zone des douleurs (Poitrine, Gorge, Abdomen, Aucun), si le patient a de la fièvre, toux et vomissement.

Attributs Le sexe de la personne.

Maladies Maladie que le patient peut avoir (Rhume, Infarctus, Appendicite, etc).

2 Règles

Nous avons organisé notre choix des règles selon un arbre décisionnelle voir figure 1. Cela correspond a des questions que pourrais poser un médecin a un patient pour déterminer la maladie du patient.

il est possibles que le patient est plusieurs maladie possible, cela est dus que notre arbre est très simplifier et que comme nous n'avons pas implémenter le chaînage mixte.

Ce qui oblige l'utilisateur à fournir tous les informations alors que si on avais un chaînage mixte on demanderais au patient que ce qui est utile.

1. Disponible dans le répertoire docs du projets.

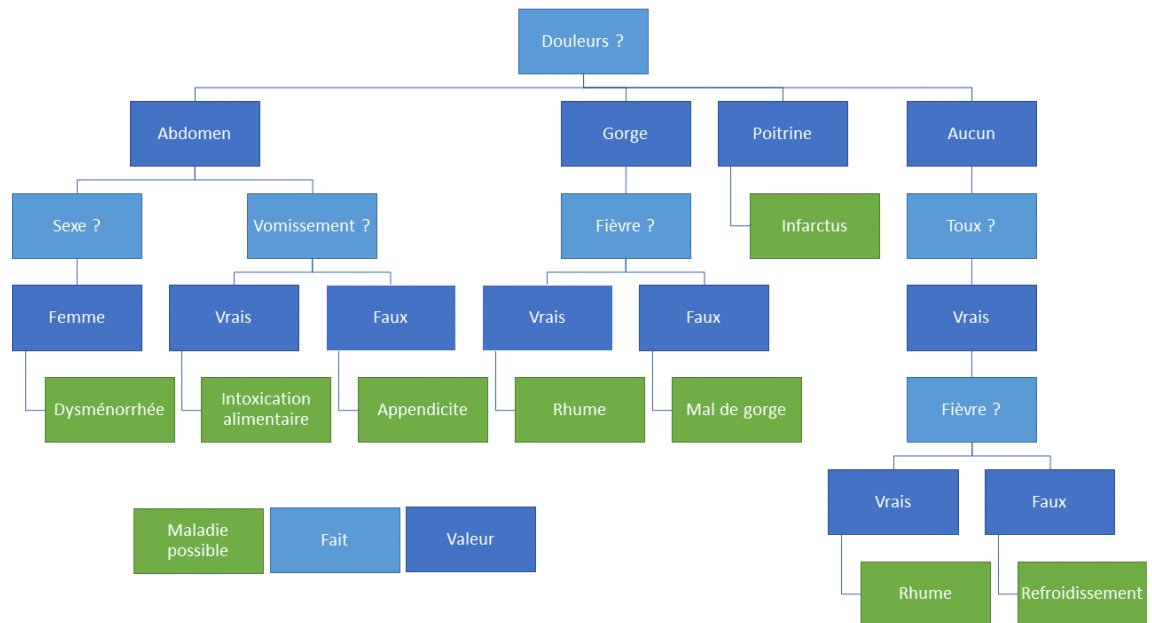


FIGURE 1 – Arbre décisionnelle

3 Choix du langage

Nous avons choisi de partir sur du python car il présente l'avantage d'être fortement typé mais être dynamique ce qui nous permet facilement de gérer les valeurs qui sont de nature différentes.

Pour l'interface on a choisi d'utiliser le package pyside2 qui permet d'utiliser du qt que nous avons déjà vus en C++ avec python et qui est le package officiel.

Deuxième partie Système Expert

4 Base de connaissance

Notre base de connaissance est constituer de la base de faits et la base de règles qui sont simplement des tableaux de faits et de règles.

Elle peut être enrichie a l'aide de fichier ou simplement de l'interface.

4.1 Fichier

Le fichier permet de déclarer des faits et des règles voir en-dessous. On ne peut avoir que un fait ou règle par ligne mais on faire une ligne de commentaire avec le symbole '#'.
Proposition on peut aussi ajouter des ou '|' et des parenthèses, les parenthèses nous on poser un problème ce qui nous a obliger a convertir nos expression in-fixe en post-fixe pour pourvoir les calculer.

```
Douleur = "Gorge"  
Fièvre = Vrai  
Sexe = "Homme"
```

```
Appendicite = Vrai := Douleur == "Abdomen" & Vomissement == Faux  
Intoxication alimentaire := Douleur == "Abdomen" & Vomissement == Vrai  
Dysménorrhée = Vrai := Douleur == "Abdomen" & Sexe = "Femme"  
Rhume = Vrai := Douleur == "Gorge" & Fièvre == Vrai  
Infarctus = Vrai := Douleur == "Poitrine"  
Mal de gorge = Vrai := Douleur == "Gorge" & Fièvre == Faux  
Rhume = Vrai := Douleur == "Aucun" & Toux == Vrai & Fièvre == Vrai  
Refroidissement = Vrai := Douleur == "Aucun" & Toux == Vrai & Fièvre == Faux
```

4.2 Interface

Selon l'interface plusieurs méthode sont disponible mais le plus simple qui marche sur les deux est de taper un ligne comme vous le feriez dans le fichier et de validé.

5 Moteur d'inférence

Nous avons implémenter le chaînage avant avec but ou juste pour sature la base de connaissance et le chaînage arrière.

A la fin de l'exécution, il affiche un résultat et le cheminement selon la configuration (changeable). Il est possible de voir le log en tapent la commande "log" pour voir tous le commandes disponible, il suffit de taper "aide".

6 Interface utilisateur

Nous avons développez deux interfaces, une en ligne de commande et une graphique.

6.1 Ligne de commande

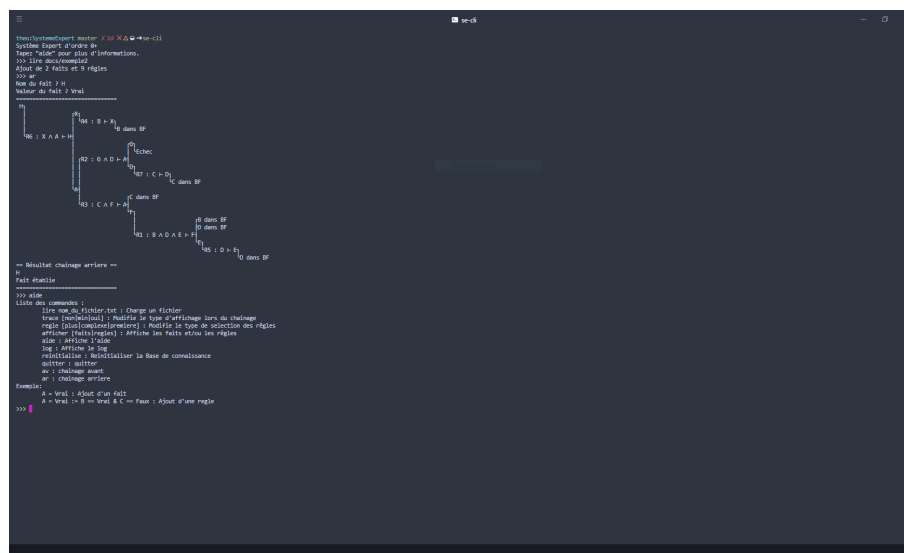


FIGURE 2 – SE-CLI

L'interface un ligne de commande s'inspire de l'interpréteur de python.

6.2 Graphique

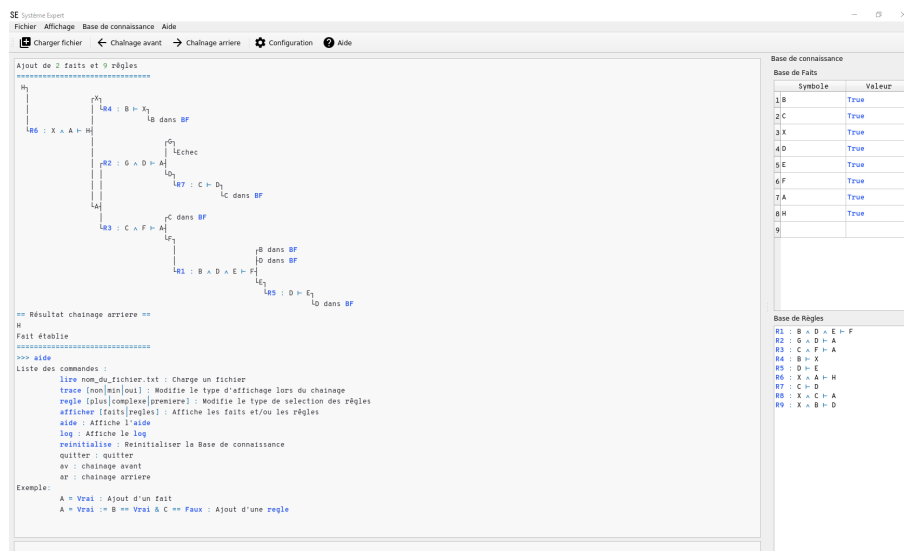


FIGURE 3 – SE-GUI

L'interface permet de simplifier l'utilisation pour un utilisateur qui n'est pas habitué par l'utilisation du terminal.

La plupart des commandes peuvent être exécutées à l'aide de boutons. Mais il est toujours possible de taper les commandes comme sur la version en ligne de commande.

Composition de l'interface

Terminal Il est constitué d'un affichage et d'une ligne de saisie. Il s'utilise exactement comme la version en ligne de commande.

Dock Il permet d'afficher la base de connaissance et permet de modifier/ajouter un fait très simplement.

Barre d'outils Il donne accès aux commandes importantes.

Troisième partie

Conclusion

7 Améliorations possibles

Voici une liste des améliorations possibles :

- Ajouter des coefficients de certitude. Pour représenter des événements incertains.
- Représenter les faits sous la forme de n-uplet pour faciliter les regroupements des faits exemple $\text{AgeBob} = 25$ devient $(\text{Bob}, \text{age}, 25)$ ou $\text{ParentsBob} = \text{"Léo Léa"}$ devient $(\text{Bob}, \text{parents}, \text{Léo}, \text{Léa})$.
- Transformer le moteur 0+ en 1.

8 Le code source du projet

Tous les codes sources du projet et ce document sont disponibles sur <https://github.com/theodeze/SystemeExpert>