MASTER M1 INFORMATIQUE

REPRÉSENTATION DES CONNAISSANCES *(IA SYMBOLIQUE)*

TP 1 – PARTIE 1 – durée estimée 1H

EXERCICE 1 - RAPPEL SUR PROLOG : PRISE EN MAIN DE L'ENVIRONNEMENT

a - Ecrivez le programme suivant dans un éditeur de texte et sauvegardez-le en menu.pl :

```
/* les entrées */
entree(crudites).
entree(terrine).
entree(melon).

/* les viandes (avec légumes associés) */
viande(steack).
viande(poulet).
viande(gigot).

/* les poissons (avec légumes associés) */
poisson(bar).
poisson(saumon).

/* les desserts */
dessert(sorbet).
dessert(creme).
dessert(tarte).

/* composition d'un menu simple : une entrée ET un plat ET un dessert */
menu_simple(E, P, D) :- entree(E), plat(P), dessert(D).

/* le plat de résistance : viande OU poisson */
plat(P) :- viande(P).
plat(P) :- poisson(P).
```

b – Lancez un interpréteur Prolog : SWI-Prolog, gProlog ou GNU Prolog (selon la salle) :

Commandes de base :

- ✗ Lancer l'interpréteur ;
- ✗ `?- consult(menu).` → choisir le menu à interroger ;
- ✗ `?- reconsult(menu).` → recharge le programme (effacer le programme précédent & charger les nouvelles règles) ;
- ✗ `?- halt.` → quitter swipl ;
- ✗ `?-help.` → aide en ligne ;

Remarque : `consult/1` et **`reconsult/1`** ne sont pas des prédicats logiques. Ils ont pour fonction d'exécuter des commandes d'enregistrement de programmes et sont appelés des *prédicats extra-logiques*.

- ✗ Pour connaître tous les desserts contenus dans la base de faits, vous devez taper:
  `?-dessert(X).`
- ✗ Pour avoir les réponses suivantes, tapez "**;**" sinon tapez "**Entrée**" pour que la résolution s'arrête.
- ✗ Attention, n'oubliez pas le point à la fin de chacune de vos requêtes.
- ✗ Faites afficher tous les menus simples possibles.

c - Formalisez en Prolog les questions suivantes et testez vos requêtes:

- ✗ Quels sont les menus simples avec des crudités en entrée ?
- ✗ Peut-on avoir un menu avec des crudités et une mousse au chocolat ?
- ✗ Quels sont les menus avec du poisson comme plat ?
- ✗ Quels sont les menus avec du melon en entrée et du poisson comme plat ?

d - Que se passe-t-il si vous inversez l'ordre des buts dans votre dernière requête ?

- ✗ Construisez (sur feuille) l'arbre de résolution dans les deux cas.
- ✗ Demandez à l'interpréteur d'afficher son raisonnement pas à pas (les traces) :
  Lancer la requête : **`?-trace, <vos requêtes> .`**

Instructions:

- ✗ creep (ou 'c') : pour passer au pas de résolution suivant;
- ✗ exit : représente une démonstration réussie d'un prédicat;
- ✗ redo : le système essaie de démontrer le prédicat d'une autre manière en faisant un retour sur un point de choix;
- ✗ echec : le système a échoué à démontrer un prédicat.

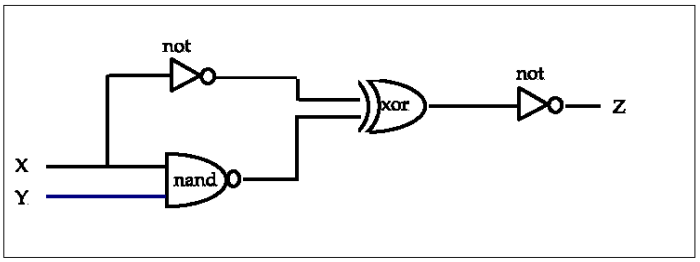
Faites afficher progressivement les traces des 2 requêtes en suivant en parallèle les arbres de résolution que vous venez de construire.

e- Que signifie les requêtes suivantes ? Testez-les :

```
?- menu_simple(E, P, D), entree(crudites).
?- menu_simple(E, P, D), !.
?- menu_simple(E, P, D), poisson(P),!.
?- menu_simple(E, P, D), !, poisson(P).
```

EXERCICE 2 - RAPPEL SUR PROLOG : REPRESENTATION DES CONNAISSANCES

Soit le circuit logique suivant :



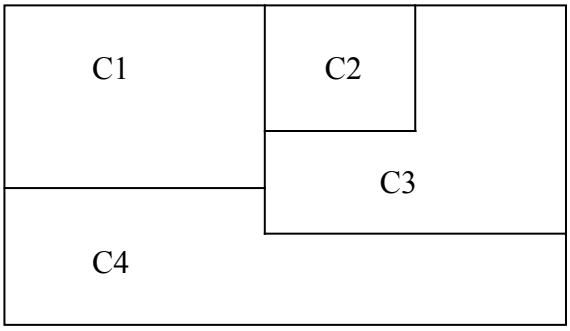
- ✗ Définissez les **prédicats** correspondant aux **composants** (ou connecteurs) logiques. Prenez pour valeurs possibles des entrées ou sorties 0 ou 1 (respectivement pour faux ou vrai).

- ✗ Définissez le **prédicat `circuit/3`**, tel que `circuit(X,Y,Z)` est vrai si et seulement si X et Y correspondent à des valeurs d'entrée pour lesquelles Z est la valeur de sortie du circuit.

- ✗ Testez les différentes **requêtes possibles** en faisant varier la valeur des arguments (entrée ou sortie). Déduisez-en la table de vérité du circuit.

EXERCICE 3 - RAPPEL SUR PROLOG : GESTION DES ENTRÉES / SORTIES – CUT ET FAIL

On se propose de définir un prédicat permettant de colorier la carte suivante :



Les règles sont les suivantes :

- On dispose de trois couleurs qui sont : vert, jaune et rouge;
- Deux zones contiguës doivent avoir des couleurs différentes.

✗ Ecrivez un prédicat **`coloriage(C1,C2,C3,C4)`** dont les littéraux sont ordonnés en deux temps. Tout d'abord on génère toutes les valeurs possibles de C1, C2, C3 et C4. Ensuite on vérifie si les colorations obtenues sont conformes à la carte par l'utilisation du prédicat **`X \== Y`** sur les couleurs des zones contiguës.

✗ Reprenez ce prédicat et modifiez le programme en plaçant les tests de différence de couleurs le plus tôt possible dans l'écriture du prédicat, c'est-à-dire en vérifiant les différences de couleurs dès que celles-ci sont instanciées. Quelle en est la conséquence ?

✗ **On veut permettre à l'utilisateur de choisir lui-même une couleur pour une des positions, s'il en a envie. Pour cela vous devez faire un menu lui laissant le choix entre laisser le système donner une réponse immédiatement ou faire d'abord le choix d'une couleur. Le résultat doit alors tenir compte du choix de l'utilisateur.**