

# RAPPORT TP5

## SYSTÈME EXPERT MÉDICAL EN PROLOG

**RÉALISÉE PAR :**

BENHAMIDA Khadija

**ENCADRÉ PAR :**

Pr. Lamghari Nidal

# PARTIE 1 : BASE DE CONNAISSANCES STATIQUE

## Objectif:

- Définir les symptômes et les règles pour chaque maladie.
- Tester le diagnostic sur des patients fictifs (p1, p2, p3).

## Code utilisé:

```
symptome(p1, fièvre).  
symptome(p1, toux).  
symptome(p1, fatigue).  
  
symptome(p2, mal_gorge).  
symptome(p2, fièvre).  
  
symptome(p3, eternuements).  
symptome(p3, nez_qui_coule).
```

```
diagnostic(Patient, covid) :-  
    symptome(Patient, fièvre),  
    symptome(Patient, toux),  
    symptome(Patient, fatigue).  
  
diagnostic(Patient, allergie) :-  
    symptome(Patient, eternuements),  
    symptome(Patient, nez_qui_coule),  
    \+ symptome(Patient, fièvre).
```

```
diagnostic(Patient, grippe) :-  
    symptome(Patient, fièvre),  
    symptome(Patient, courbatures),  
    symptome(Patient, fatigue).  
  
diagnostic(Patient, angine) :-  
    symptome(Patient, mal_gorge),  
    symptome(Patient, fièvre).
```

- Chaque patient a des symptômes définis par symptome/2.
- diagnostic/2 détermine la maladie si les symptômes correspondent.

## On peut tester dans SWI-Prolog :

```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help

?- diagnostic(p1, M).
M = covid ,

?- diagnostic(p2, M).
M = angine ,

?- diagnostic(p3, M).
M = allergie.

?- dynamic(answer/2).
true.
```

- SWI-Prolog avec l'exécution des commandes `diagnostic(p1, M)` et `diagnostic(p2, M)`.
- Résultats affichés pour montrer que le TP fonctionne.

## PARTIE 2 : INTERACTION AVEC L'UTILISATEUR

### Objectif:

- Poser des questions à l'utilisateur (oui/non) pour savoir quels symptômes sont présents.
- Ne pas poser deux fois la même question (mémoriser les réponses).
- Déduire les maladies possibles.

### Code utilisé:

```
:- dynamic(answer/2).

% Questions pour chaque symptôme
question(fievre, 'Avez-vous de la fievre (o/n) ?').
question(toux, 'Avez-vous de la toux (o/n) ?').
question(mal_gorge, 'Avez-vous mal de gorge (o/n) ?').
question(fatigue, 'Resentez-vous de la fatigue (o/n) ?').
question(courbatures, 'Avez-vous des courbatures (o/n) ?').
question(mal_tete, 'Avez-vous mal de tete (o/n) ?').
question(eternuements, 'Avez-vous des eternuements (o/n) ?').
question(nez_qui_coule, 'Avez-vous le nez qui coule (o/n) ?').
```

```

read_yes_no(yes) :-
    read_line_to_string(user_input, S),
    string_lower(S, S2),
    sub_string(S2, 0, 1, _, "o"), !.
read_yes_no(no).

ask_symptom(Symptom) :-
    question(Symptom, Prompt),
    format('~w ', [Prompt]),
    read_yes_no(Resp),
    assertz(answer(Symptom, Resp)).

```

```

has_symptom(Symptom) :-
    answer(Symptom, yes), !.
has_symptom(Symptom) :-
    answer(Symptom, no), !, fail.
has_symptom(Symptom) :-
    ask_symptom(Symptom),
    answer(Symptom, yes).

```

```

maladie(grippe) :- has_symptom(fievre), has_symptom(courbatures), has_symptom(fatigue).
maladie(angine) :- has_symptom(mal_gorge), has_symptom(fievre).
maladie(covid) :- has_symptom(fievre), has_symptom(toux), has_symptom(fatigue).
maladie(allergie) :- has_symptom(ternuements), has_symptom(nez_qui_coule),
    | | | | | (answer(fievre, no) -> true; \+ answer(fievre, yes)).

expert :-
    retractall(answer(_, _)),
    writeln('--- Debut du diagnostic ---'),
    findall(M, maladie(M), Liste),
    afficher_resultats(Liste),
    writeln('--- Fin du diagnostic ---').

```

- ask\_symptom/1 pose la question et mémorise la réponse.
- has\_symptom/1 permet de vérifier si le symptôme est présent.
- maladie/1 utilise ces informations pour déduire les maladies.
- expert/0 lance le diagnostic interactif.

## PARTIE 3 : EXPLICATION / JUSTIFICATION

### Objectif:

- Expliquer à l'utilisateur pourquoi une maladie est proposée.
- Afficher uniquement les symptômes confirmés.

### Code utilisé:

```

disease_symptoms(grippe, [fievre, courbatures, fatigue, toux]).
disease_symptoms(angine, [mal_gorge, fievre]).
disease_symptoms(covid, [fievre, toux, fatigue]).
disease_symptoms(allergie, [ternuements, nez_qui_coule]).

```

```

expliquer(Maladie, Confirmes) :-
    disease_symptoms(Maladie, Symptomes),
    include(symptom_confirmed, Symptomes, Confirmes).

symptom_confirmed(S) :- answer(S, yes).

afficher_resultats([]) :-
    writeln('Aucune maladie probable trouvee.'),
    writeln('Consultez un professionnel de sante si vous etes inquiet.').

afficher_resultats(L) :-
    writeln('Diagnostic(s) possible(s) :'),
    forall(member(M, L),
        (expliquer(M, Confirmes),
         format('~w : ~w~n', [M, Confirmes])
        )).

```

- disease\_symptoms/2 : stocke tous les symptômes caractéristiques.
- expliquer/2 : filtre les symptômes confirmés.
- afficher\_resultats/1 : affiche le diagnostic et les symptômes

## Screenshots:

Après avoir lancé ?- expert. et répondu aux questions, prendre un screenshot montrant :

- Les maladies détectées
- Les symptômes confirmés pour chaque maladie.

## Résultats:

```

?- expert.
--- Debut du diagnostic ---
Avez-vous de la fièvre (o/n) ? o
Avez-vous des courbatures (o/n) ? n
Avez-vous mal de gorge (o/n) ? o
Avez-vous de la toux (o/n) ? o
Ressez-vous de la fatigue (o/n) ? o
Avez-vous des eternuements (o/n) ? n
Diagnostic(s) possible(s) :
- angine : [mal_gorge,fièvre]
- covid : [fièvre,toux,fatigue]
--- Fin du diagnostic ---
true.

```