| Experiment No.7 |
| --- |
| Implement forward chaining reasoning for Medical Diagnosis System Using Prolog. |
| Date of Performance: 16/03/25 |
| Date of Submission: 25/03/25 |

**Aim:** Implement forward chaining reasoning for Medical Diagnosis System Using Prolog.

**Objective:** To implement a knowledge-based expert system for medical diagnosis using forward chaining in Prolog.

# Software Required:

* SWI-Prolog or any Prolog interpreter

**Theory:** Prolog is a logic programming language widely used for artificial intelligence and expert systems. Forward chaining is a data-driven reasoning approach where inference begins with known facts and applies rules to derive conclusions. In this experiment, we will implement a medical diagnosis system using forward chaining.

# Procedure:

1. **Install SWI-Prolog:** Ensure that SWI-Prolog is installed on your system.
2. **Create a Prolog file:** Open a text editor and save the file with a .pl extension, e.g., medical\_diagnosis\_fc.pl.
3. **Define the Knowledge Base:** List symptoms and corresponding diseases using facts. (use the knowledge base implemented in experiment no 6)
4. **Implement Forward Chaining Rules:** Use Prolog rules to iteratively apply facts and derive conclusions.
5. **Query the System:** Use Prolog queries to test the diagnosis system.

# Code Implementation:

% Facts defining initial symptoms symptom\_present(fever). symptom\_present(cough).

% Forward chaining rules for medical diagnosis diagnose :-

symptom\_present(fever), symptom\_present(cough), assert(disease(flu)),

write('The patient may have flu.'), nl.

diagnose :- symptom\_present(sneezing), symptom\_present(runny\_nose), assert(disease(common\_cold)),

write('The patient may have common cold.'), nl.

diagnose :- symptom\_present(fever), symptom\_present(cough), symptom\_present(loss\_of\_taste), assert(disease(covid\_19)),

write('The patient may have COVID-19.'), nl.

% Sample Query

% ?- diagnose.

# Expected Output:

?- diagnose.

The patient may have flu.

# Observations:

* The system successfully applies forward chaining to derive a diagnosis.
* The use of assert/1 allows the system to dynamically add inferred diseases.
* The system can be extended with more rules and symptoms.

**Your Program Code:**

% Facts defining symptoms

symptom\_present(fever).

symptom\_present(cough).

symptom\_present(sneezing).

symptom\_present(runny\_nose).

symptom\_present(loss\_of\_taste).

symptom\_present(sore\_throat).

symptom\_present(body\_aches).

symptom\_present(fatigue).

symptom\_present(chills).

symptom\_present(headache).

symptom\_present(weight\_loss).

symptom\_present(night\_sweats).

symptom\_present(difficulty\_breathing).

% Rule 1: If the patient has fever and cough, diagnose flu

diagnose :-

symptom\_present(fever),

symptom\_present(cough),

write('The patient may have Flu.\n'),

write('Common symptoms of Flu: fever, cough, body aches, fatigue.\n'),

write('Recommended action: Get rest, stay hydrated, and consult a doctor if symptoms persist.\n\n').

% Rule 2: If the patient has sneezing and runny nose, diagnose common cold

diagnose :-

symptom\_present(sneezing),

symptom\_present(runny\_nose),

write('The patient may have Common Cold.\n'),

write('Common symptoms of Common Cold: sneezing, runny nose, sore throat.\n'),

write('Recommended action: Rest, drink fluids, and take over-the-counter medicine for relief.\n\n').

% Rule 3: If the patient has fever, cough, and loss of taste, diagnose COVID-19

diagnose :-

symptom\_present(fever),

symptom\_present(cough),

symptom\_present(loss\_of\_taste),

write('The patient may have COVID-19.\n'),

write('Common symptoms of COVID-19: fever, cough, loss of taste or smell, shortness of breath.\n'),

write('Recommended action: Isolate yourself, get tested for COVID-19, and follow government guidelines.\n\n').

% Rule 4: If the patient has sore throat, fever, and body aches, diagnose Strep Throat

diagnose :-

symptom\_present(sore\_throat),

symptom\_present(fever),

symptom\_present(body\_aches),

write('The patient may have Strep Throat.\n'),

write('Common symptoms of Strep Throat: sore throat, fever, body aches.\n'),

write('Recommended action: Visit a doctor for a throat culture or rapid strep test, and take antibiotics if prescribed.\n\n').

% Rule 5: If the patient has fever, chills, and fatigue, diagnose Malaria

diagnose :-

symptom\_present(fever),

symptom\_present(chills),

symptom\_present(fatigue),

write('The patient may have Malaria.\n'),

write('Common symptoms of Malaria: fever, chills, fatigue, sweating.\n'),

write('Recommended action: Seek medical attention immediately for blood tests and antimalarial treatment.\n\n').

% Rule 6: If the patient has chronic cough, weight loss, and night sweats, diagnose Tuberculosis (TB)

diagnose :-

symptom\_present(cough),

symptom\_present(weight\_loss),

symptom\_present(night\_sweats),

write('The patient may have Tuberculosis (TB).\n'),

write('Common symptoms of TB: chronic cough, weight loss, night sweats, fatigue.\n'),

write('Recommended action: Consult a healthcare provider for a TB test and start treatment if necessary.\n\n').

% Rule 7: If the patient has fever, cough, and difficulty breathing, diagnose Pneumonia

diagnose :-

symptom\_present(fever),

symptom\_present(cough),

symptom\_present(difficulty\_breathing),

write('The patient may have Pneumonia.\n'),

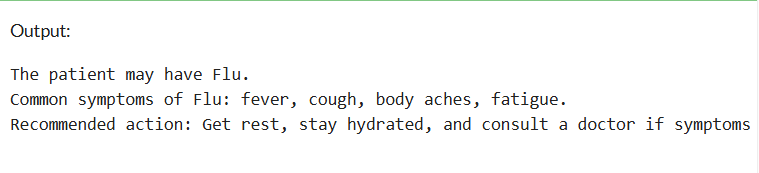
write('Common symptoms of Pneumonia: fever, cough, difficulty breathing, chest pain.\n'),

write('Recommended action: Seek urgent medical care for diagnosis, imaging (X-ray), and antibiotic treatment.\n\n').

% Initialization goal to run the diagnosis when the program starts

:- initialization(diagnose).

**Output:**

****

**Conclusion:**

In this experiment, we successfully implemented a medical diagnosis system using Prolog, based on forward chaining reasoning. By defining facts (symptoms) and rules (diagnoses), we created a knowledge-based expert system that can suggest potential diseases based on the symptoms provided. The system can be extended easily by adding more symptoms and diseases, and we enhanced the output to provide detailed information, including common symptoms and recommended actions. This approach demonstrates the power of Prolog in developing intelligent systems that can infer conclusions from a set of predefined rules, making it a valuable tool for building medical diagnosis application