Introduction to Artificial Intelligence

Lab 12 (Week 14) - Logic Based Agents -

2023 - 2024

May 19^{th}, 2024

Objectives

- To represent a `knowledge base` with a python data structure and implement `forward chaining` for simple rules.
- To represent and implement a `knowledge base` with a python data structure and for the `Wumpus` world problem.
- To understand `DPLL` and its heuristics and comment the corresponding code.

Overview:

In this Lab, we will delve into Logic-based Agents, particularly agents that use Propositional Logic to reason and interact with the world.
The primary objectives of this lab are to understand how to formulate facts and rules for a given problem and how to construct a corresponding knowledge base that enables the agent to make inferences and learn new facts based on observed knowledge.

In the first part of this lab, you will represent a set of given facts and rules using an appropriate data structure. Subsequently, you will implement a forward chaining algorithm to infer new facts based on the provided rules and initial facts.

For the second task, you will be presented with a simplified version of the Wumpus World problem (detailed in CHAPTER 7). You will formulate facts and rules specific to this problem. Additionally, you will explore a satisfiability solver algorithm with three different heuristics.

Your mission

Your mission is to perform the following tasks:

Task 1:

Given the following facts and rules, along with the pseudo algorithm provided in Algorithm 1, you are tasked with designing a logic-based agent equipped with a knowledge base containing these facts and rules. The agent should be capable of inferring new facts based on its knowledge base using a forward chaining algorithm.

The facts represent various environmental conditions and situations. These facts include "**morning**" being **true**, "**weekend**" being **false**, the "**temperature**" set to **20** degrees Celsius, "**sunny**" weather, no "**traffic_jam**," and the "**car_started**" being **true**.

Alongside these facts, we have defined a series of rules that leverage these conditions to draw logical conclusions. For instance, if it is morning and not the weekend, it implies a "**workday**." Similarly, if the temperature is equal to 25 degrees Celsius and it is sunny, we infer a "**hot_day**." Conversely, if the temperature is equal 15, it signifies a "**mild_day**," while temperatures of 10 degrees Celsius suggest a "**cold_day**." Other rules account for situations like : being **late for work** due to a traffic jam without the car starting, or **enjoying outdoor** activities on a sunny weekend. Additionally, the **morning being pleasant** is inferred when it's morning and the temperature is 20 degrees Celsius.

Algorithm 1: Forward Chaining Algorithm

Input: KB ;
Output: New_facts or $None$;
rules = KB.rules;
facts = KB.facts;
inferred_facts = empty dictionary;
agenda = empty list;
Add eligible rules to agenda (conditions values in rule == facts values);
while *agenda* ***not empty*** **do**
    rule = get the first rule of agenda;
    conclusion = rule.then;
    if *conclusion not in* $inferred\_facts$ **then**
        Add conclusion to $inferred\_facts$;
        Add conclusion to the main facts;
        Add satisfactory rules (eligible) to agenda;
    end
end
return $inferred\_facts$;

- Adjust the temperature rules to evaluate whether the temperature is greater than or less than a specified threshold rather than checking for equality. (Maybe using a function instead of a value 🤔).

Task 2:

The Wumpus World is a classic example used to demonstrate the value of a knowledge-based agent and the principles of knowledge representation.
The figure below depicts a Wumpus World containing **one pit** and **one Wumpus**. An agent starts in room **[1,1]** with the objective of exiting the Wumpus World alive by reaching room **[4,4]**. The Wumpus World has **exactly one** Wumpus and one pit. A **breeze** is present in the rooms **adjacent** to the pit, and a **stench** is present in the rooms adjacent to the Wumpus.
In this task, you have been provided with the `WumpusWorldProblem` class as an attached Python file. Your objectives are as follows:

- `Initialize Method`: In the initialize method, ensure that you add the appropriate clauses in Conjunctive Normal Form (CNF), considering the constraints of the provided Wumpus world problem. The initial configuration of the problem is depicted in Figure 1.
- `Code Understanding`: Carefully read and understand the provided code. The implemented method is called DPLL (Davis-Putnam-Logemann-Loveland), which falls into the category of algorithms used for solving satisfiability problems (SAT). Its primary purpose is to determine whether a given propositional logic formula is satisfiable.
- `Heuristics Explanation`: Learn about the heuristics used within the DPLL algorithm, namely Pure symbols, Unit clauses, and early termination. Understand the advantages of using such heuristics.
- `Heuristic Removal and Observation`: Remove the Pure symbols, Unit clauses, and Early termination heuristics from the code, and then execute the modified code. Take note of any differences or observations in terms of performance, efficiency, or correctness. This exercise will demonstrate the impact and importance of these heuristic strategies in the DPLL algorithm.

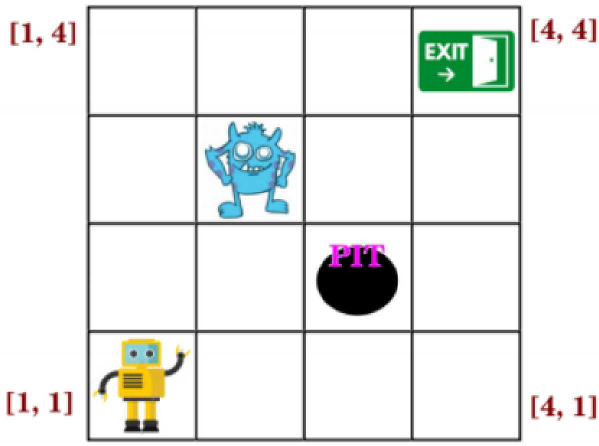

Figure 1: Wumpus world