CS-208:Artificial Intelligence
L-02
Solving AI Problems

Distinguishing AI Problems

The problems for which can not be solved through precise models, methods and algorithms as well as involve uncertainty are classified as AI problems.

Two major approaches to solve AI problems:

I. One is using Hard-Computing Techniques which are classical AI problem solving methods based on symbolism and search with an aid of symbolic logic.

II. The other uses soft-computing techniques- which consist of five major components namely Fuzzy Logic, Neural Networks, Probabilistic Reasoning, Genetic Algorithm and Chaos Theory.

In this course we focus on the first approach for solving AI problems

How to build a system that solve AI Problems

This involves the following four important tasks:

1. **Define the Problem Precisely**: This definition must include the precise specification of what the initial situation will be as well as what final situations constitute the acceptable solution to the problem.
2. **Analyze the Problem** : A few very important features can have immense impact on the appropriateness of the various possible techniques for solving the problem.
3. I**solate and Represent the task knowledge** that is necessary to solve the problem
4. **Choose the best problem solving technique(s)** and apply it(them) to the particular problem

An Example of Build a Program That Could Play Chess

Before start writing a program that can play chess, we need to specify the following:

1. ***The starting position of the chess board***
2. ***The rules that defines the legal moves***
3. ***The board positions that represent a win for one side or other***

Goal: The program not only play a legal game of chess, but also winning the game if possible

Formal Definition

(Representing Chess Playing Problem from programming point of view)

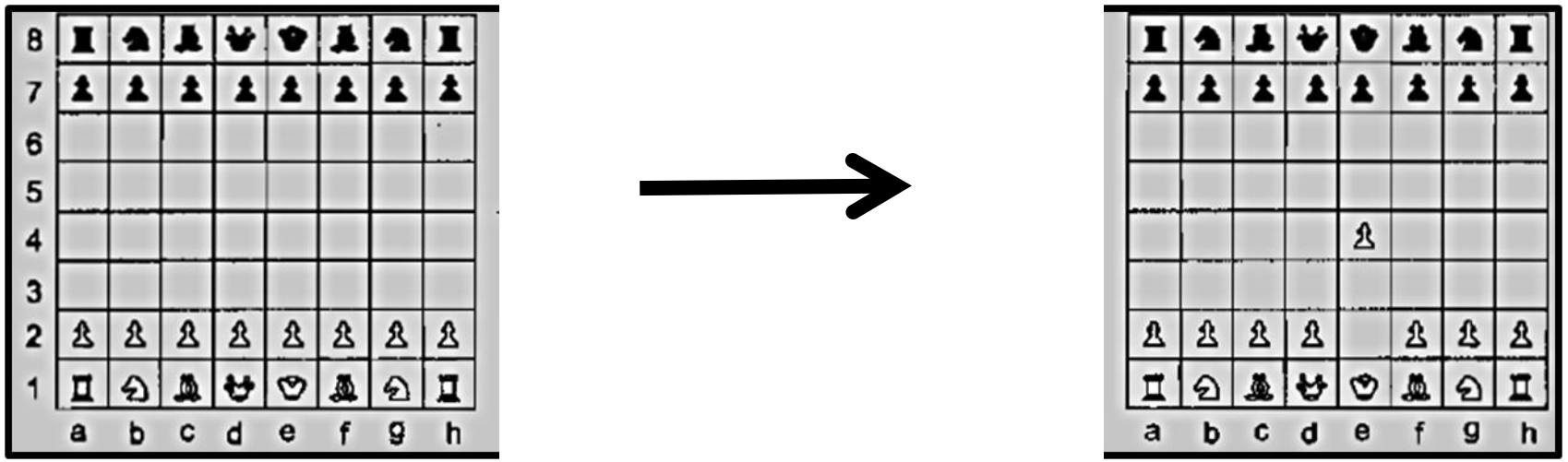
1. **The Starting Position:** it can be described by 8x8 array where each position contains a symbol at appropriate place same as in the official chess opening board position.
2. **The Goal Position**: Any board position that the opponent can not make any legal move or his/her king is under attack.
3. **Legal moves:** These provide the ways of getting from the starting state to goal state

Describes the starting position and the goal position can be done very easily, but the real difficulty lies in how to describe the rules for legal moves.

One Way of Describing the Legal Moves

This left side serves as a pattern to be matched against the current board position

This right sides reflects the change in the board after the application of the legal moves.



Difficulties in This Legal Move Representation

If we write rules in this manner, there will be 10^{120} rules equal to the number of possible board positions. This creates the following implementation difficulties

I. *Nobody can supply the complete set of rules*

- It takes too much time (more than life span of our earth)

- It can not done without mistake or repetition

II. *No program can handle this huge number of rules*

- It is not possible for searching applicable rules for the current board position

- It is not possible to Store 10^{120} rules in computer memory

What is State Space?

Background Knowledge

Definition of a State: *A situation in the world described by some facts is a state.*

A state can be *instantaneous* or it can *last for some time (non-instantaneous)*.

The state changes(transitions) can be either *discrete* or *continuous*.

In physics: *A physical state is a set of measurable physical parameters that is fully describing some part of the universe*.

In computer science: *A process state is a set of values that are currently stored in registers, buffers and stack. So process state is a snap shot of a process in time.*

In Artificial Intelligence: *A problem state* includes everything important about a situation in one bundle (that is everything necessary to reason about it). It usually means a list of facts.. There may be lot of facts and we often include only those facts that are necessary for reason about in solving the AI problems.

Two tricks to simplify the state description:

1. *Record only facts relevant to the very specific problem.*
2. *Compressing the facts.*

State Space Search as Solving the AI Problem

- We can view the solving of an AI problem as *moving around the state space that begins with a starting state and ends with one of the goal state*.
- The set of all possible states of a problem constitute *problem universe*.
- Now the problem of playing chess can be defined as a problem of moving around in a state space, where each state corresponds to a legal position of the chess board.
- chess can be played by starting at an starting/initial state, using a set of legal rules to move from one state to another and attempt to end up in one of the goal/final state.

Structure State Space Representation corresponds to the Structure of Problem Solving

- State space representation allows the **formal** definition of a problem as the need to convert some given situation into some desired situation using a set of permissible operation

- State space representation also permits us to define the process of solving a particular problem as a combination of known techniques and the search.

An Illustrative Example

Water Jug Problem: There are two water jugs, one is 4-liter capacity and the other is 3-liter capacity. Neither has any measuring marks on it. There is a water pump that can be used to fill the jugs with water. How can we get 2 liters of water in the 4-liter jug?

- How to make a good representation of a state for this problem?
- How to describe a set of state transition rules?

Representation of the (Water Jug) Problem State

The problem state can be represented as an ordered pairs of integers **(*a*, *b*)** that represent the quantities of water in 4-liter and 3-liter jugs respectively.

Where,

a represents the quantity of water in 4-liter jug and takes one of the values 0, 1, 2, 3, 4.

b represents the quantity of water in 3-liter jug and takes one of the values 0, 1, 2, 3.

Starting(or Initial) State can be represented by (0,0).

Goal(or End) State is represented by (2,*x*).

State Transition Rules or Production Rules

Before describing the rules/operators, we need to make few assumptions which are not explicitly mention in the informal problem statement. The unstated assumptions for the water jug problem are:

a. *We can fill water in a jug using the pump.*

b. *We can pour water out of a jug onto the ground.*

c. *We can transfer water from one jug to another.*

d. *There no other measuring device available.*

- ✓ Left side of the rule represent the current state.
- ✓ Right side of the rule represent the state that reflect the changes in the current state by application of the rule.
- ✓ Group of similar branches into categories called operators or production rules. It is important to note that the pre-condition of the rule has to be satisfied by the current state.

Production Rules for Water Jug Problem

Filling the water jug up to its full capacity

1. (a, b) → (4, b) Fill 4-liter jug

   If a<4 (Pre-Condition)

2. (a, b) → (a, 3) Fill 3-liter jug

   If b<3

Emptying the water jug completely

3. (a, b) → (0, b) Empty the 4-liter jug on the ground

   If a>0

4. (a, b) → (a, 0) Empty the 3-liter jug on the ground

   If b>0

Transferring the water from one jug to another

5. (a, b) $\rightarrow$ (4, b-(4-a)) Pour water from 3 liter jug into 4 liter jug until it becomes full.
   If (a+b) ≥ 4 and b>0

6. (a, b) $\rightarrow$ (a-(3-b), 3) Pour water from 4 liter jug into 3 liter jug until it becomes full.
   If (a+b) ≥ 3 and a>0

7. (a, b) $\rightarrow$ (a+b, 0) Pour all the water from 3 liter jug into 4 liter jug.
   If (a+b) ≤ 4 and b>0

8. (a, b) $\rightarrow$ (0, a+b) Pour all the water from 4 liter jug into 3 liter jug.
   If (a+b) ≤3 and a>0

Pour some quantity of water(d) out of the water jug

9. (a, b) → (a-d, b) Pour some quantity of water(d) out of 4-liter jug
   If a>0

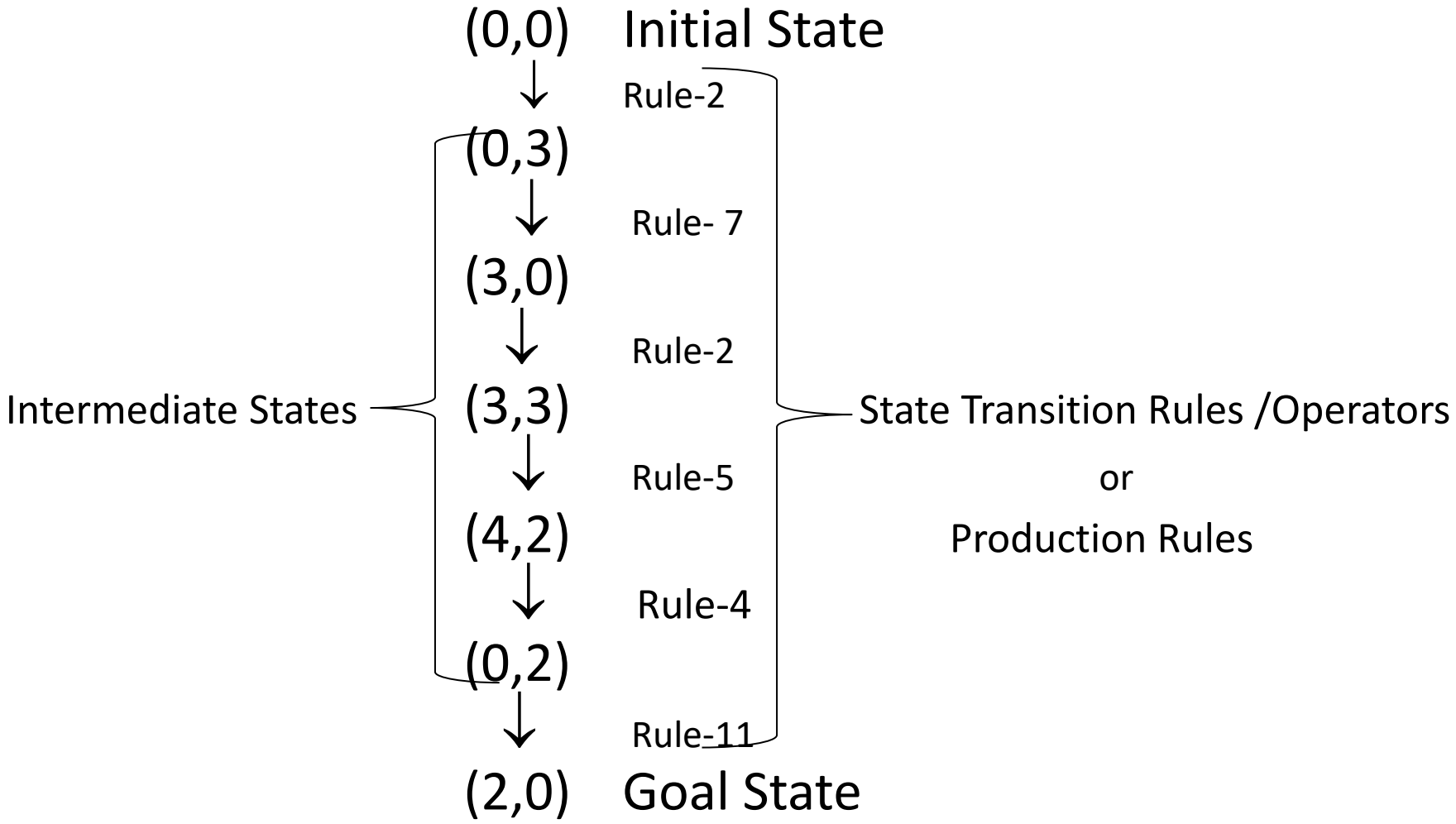
10. (a, b) → (a, b-d) *Pour some quantity of water(d) out of 3-liter jug*
    If b>0

Special Case Rules That Represent the Problem Solving Knowledge

11. (0, 2) → (2, 0) Transfer 2 liter of water from the 3-liter jug into the 4-liter jug

12. (a, 2) → (0, 2) Empty the 4-liter jug

One of the solutions to Water Jug Problem



Translating Informal Description of an AI Problem into Formal Description

1. Define the Problem State Space-the problem state contains al possible configuration of relevant objects
2. Specify one or more states within state space that describe possible situation from which the problem solving process may start. These states are called initial states
3. Specify one or more stats that would be acceptable as a solution to the problem. These states are called goal state.
4. Specify a set of rules that describe the action available
   - What are all the unstated assumptions that are hidden in the informal problem description.
   - How general should the rules be made?
   - How much work required to solve the problem should be pre-computed and represented in the rules.

Production System

A production system consists of

1. **A Set of Rules** *each consisting of left side that determines the applicability of the rule and the right side that describe the operation to be performed, if the rule is applied.*
2. **One or more Knowledge/Databases** *that contain whatever information is appropriate for the particular task.*
3. **Control Strategy** *that specifies the order in which the rules will compared to the databases and way of resolving conflicts that arise several rules match at once*.
4. **A Rule Applier**.

****

- ***Heuristic*** **is a rule of thumb, a piece of information that can provide useful guidance to a problem solver, but which is not guaranteed to be applicable or beneficial in all situations.**
- ***Heuristic Search*** **is a search using heuristic knowledge to focus the exploration of the state space on more promising path.**

****