SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

19AMB303-FULL STACK AI

M.POORNIMA DEVI,AP/AIML

olving Problems by Searching

- Reflex agent is simple
  - base their actions on
  - a direct mapping from states to actions
  - but cannot work well in environments
    - which this mapping would be too large to store
    - and would take too long to learn
- Hence, goal-based agent is used

Problem-solving agent

- Problem-solving agent
  - A kind of goal-based agent
  - It solves problem by
    - finding sequences of actions that lead to desirable states (goals)
  - To solve a problem,
    - the first step is the ***goal formulation***, based on the current situation

Goal formulation

- The goal is formulated
  - as a set of states, in which the goal is satisfied
- Reaching from initial state → goal state
  - Actions are required
- *Goal formulation, based on the current situation and the agent's performance measure, is the first step in problem solving.*
- Actions are the operators
  - causing transitions between states
  - **Actions** should be abstract enough at a certain degree, instead of very detailed
  - E.g., turn left VS turn left 30 degree, etc.

Problem formulation

- The process of deciding
  - what actions and states to consider, given a goal
- E.g., driving Amman → Zarqa
  - in-between states and actions defined
  - States: Some places in Amman & Zarqa
  - Actions: Turn left, Turn right, go straight, accelerate & brake, etc.

Search

Because there are many ways to achieve the same goal

- Those ways are together expressed as a tree
- Multiple options of unknown value at a point,
  - the agent can examine different possible sequences of actions, and choose the best
- This process of looking for such a sequence is called ***search***
- A search algorithm takes a problem as input and returns a ***solution*** *in the form of an action sequence.*

Search algorithm

- Defined as
  - taking a *problem*
  - and returns a *solution*
- Once a solution is found
  - the agent follows the solution
  - and carries out the list of actions – execution phase
- Design of an agent
  - “Formulate, search, execute”

A simple problem-solving agent

```
function SIMPLE-PROBLEM-SOLVING-AGENT(p) returns an action
    inputs: p, a percept
    static: s, an action sequence, initially empty
            state, some description of the current world state
            g, a goal, initially null
            problem, a problem formulation
    state ← UPDATE-STATE(state, p)
    if s is empty then
        g ← FORMULATE-GOAL(state)
        problem ← FORMULATE-PROBLEM(state, g)
        s ← SEARCH(problem)
    action ← RECOMMENDATION(s, state)
    s ← REMAINDER(s, state)
    return action
```

ell-defined problems and solutions

- A problem is defined by 4 components:
  - The ***initial state***
    - that the agent starts in
  - The set of possible **actions**

Transition model: *description of what each action does.*

(***successor functions***): refer to any state reachable from given state by a single action

Initial state, actions and Transition model define the ***state space***

- the set of all states reachable from the initial state by any sequence of actions.

A ***path*** in the state space:

- any sequence of states connected by a sequence of actions.

Well-defined problems and solutions

- The ***goal test***
  - Applied to the current state to test
    - if the agent is in its goal

-Sometimes there is an explicit set of possible goal states. (example: in Amman).

-Sometimes the goal is described by the properties

- instead of stating explicitly the set of states

Example: Chess

- the agent wins if it can capture the KING of the opponent on next move (checkmate).
- no matter what the opponent does

Well-defined problems and solutions

- A ***path cost*** function,
  - assigns a numeric cost to each path
  - = performance measure
  - denoted by ***g***
  - to distinguish the best path from others
- Usually the path cost is
  - the sum of the **step costs** of the individual actions (in the action list)

Well-defined problems and solutions

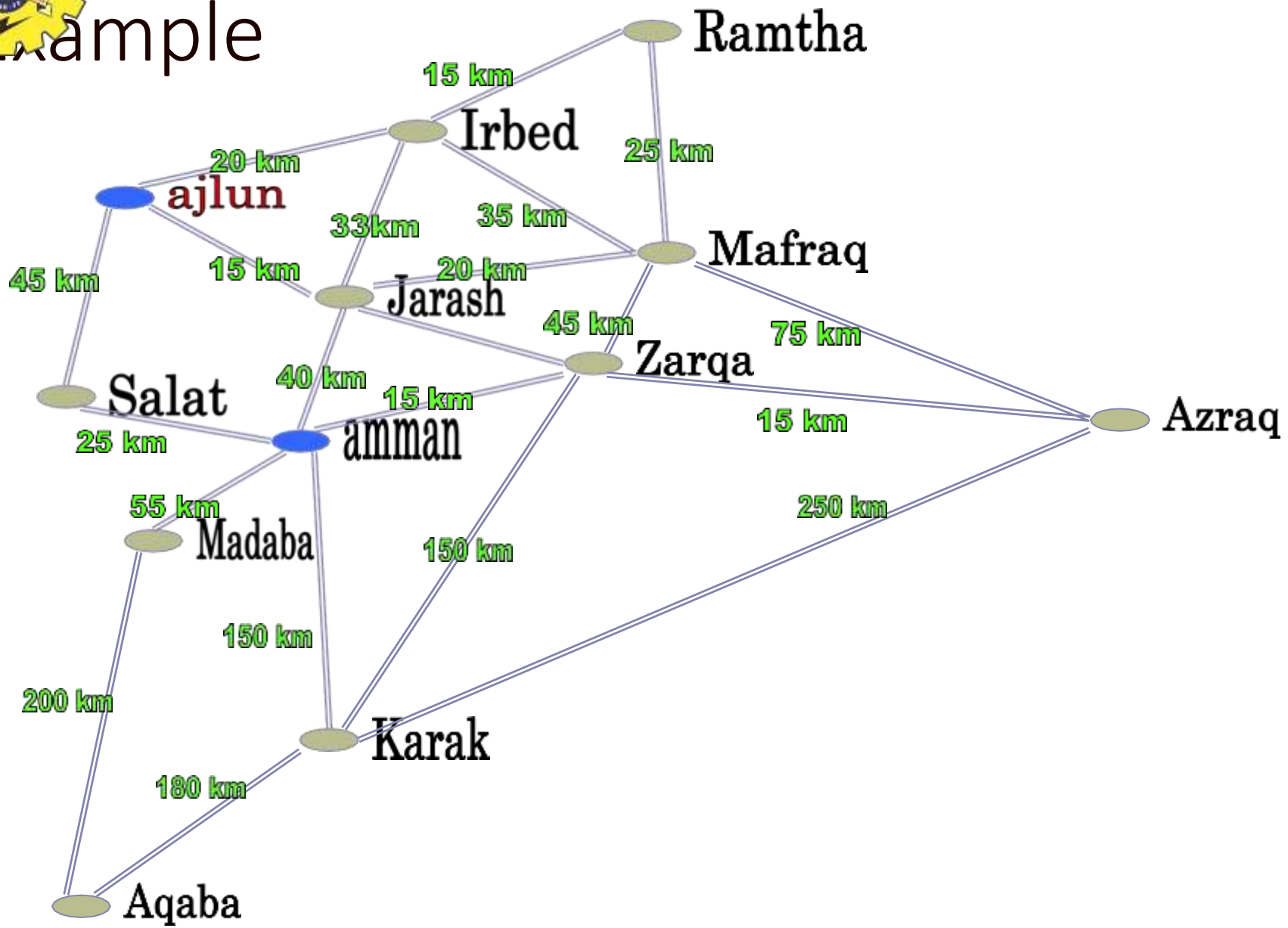
- Together a problem is defined by
  - Initial state
  - Actions
  - Successor function
  - Goal test
  - Path cost function
- The ***solution*** of a problem is then
  - *a path from the initial state to a state satisfying the goal test*
- ***Optimal*** solution
  - the solution with lowest path cost among all solutions

Formulating problems

- Besides the four components for problem formulation
  - anything else?
- ***Abstraction***
  - the process to take out the irrelevant information
  - leave the most essential parts to the description of the states

  (Remove detail from representation)
  - **Conclusion**: Only the most important parts *that are contributing to searching* are used

Example


Ramtha
15 km
Irbed
25 km
20 km
ajlun
35 km
33km
Mafraq
45 km
15 km
20 km
Jarash
45 km
75 km
Zarqa
Salat
40 km
15 km
15 km
Azraq
25 km
amman
55 km
250 km
Madaba
150 km
150 km
200 km
Karak
180 km
Aqaba

From our Example

1. Formulate Goal

- **Be In Amman**

2. Formulate Problem

- **States : Cities**
- **actions : Drive Between Cities**

3. Find Solution

- **Sequence of Cities : ajlun – Jarash - Amman**

Our Example

1. **Problem : To Go from Ajlun to Amman**

2. **Initial State : Ajlun**

3. **Operator : Go from One City To another .**

4. **State Space : {Jarash , Salat , irbed,……..}**

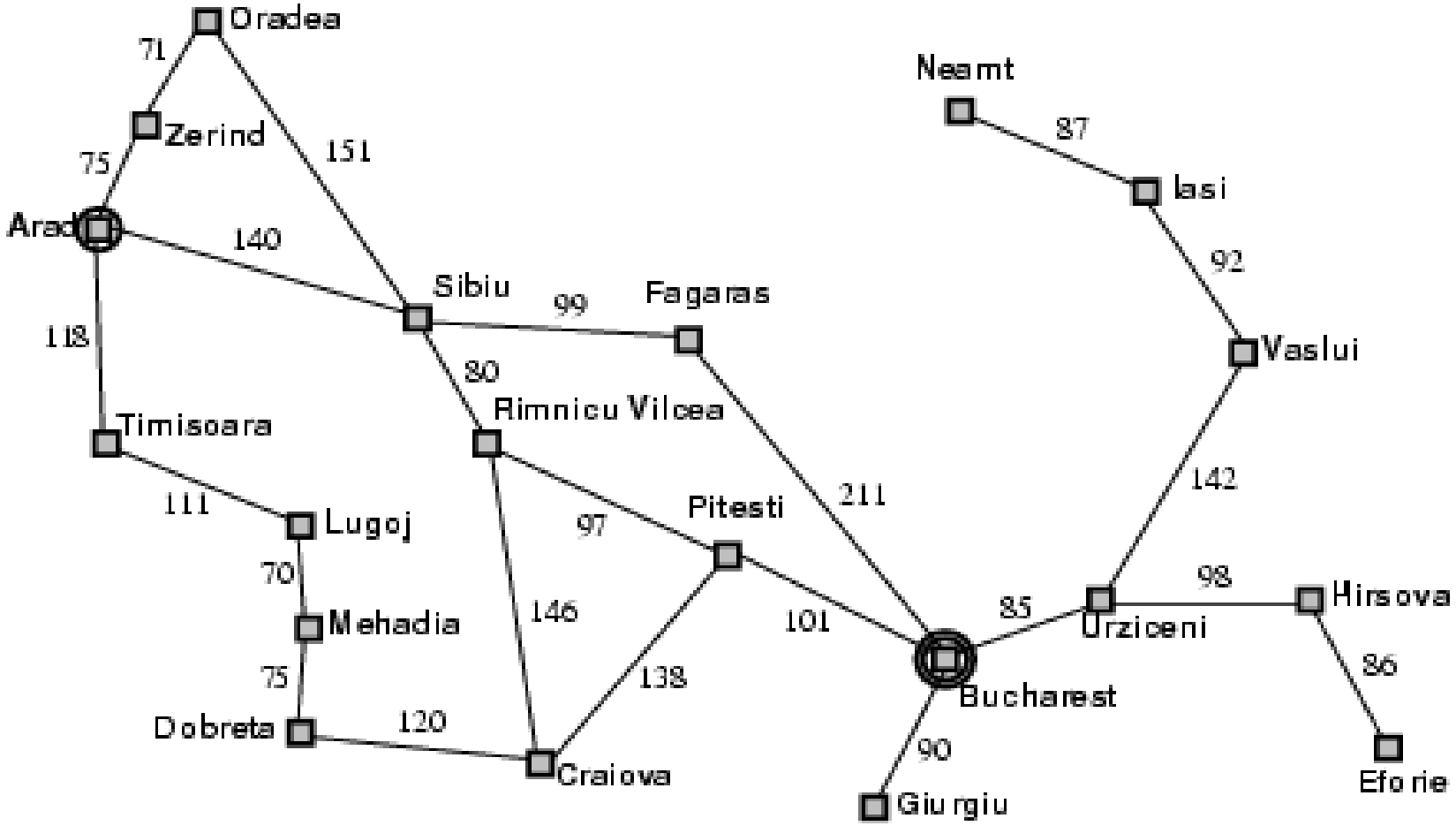
5. **Goal Test : are the agent in Amman.**

6. **Path Cost Function : Get The Cost From The Map.**

7. **Solution :** { {Aj → Ja → Ir → Ma → Za → Am} , {Aj →Ir → Ma → Za → Am} … . {Aj → Ja → Am} }

8. **State Set Space :** {Ajlun → Jarash → Amman}

Example: Romania

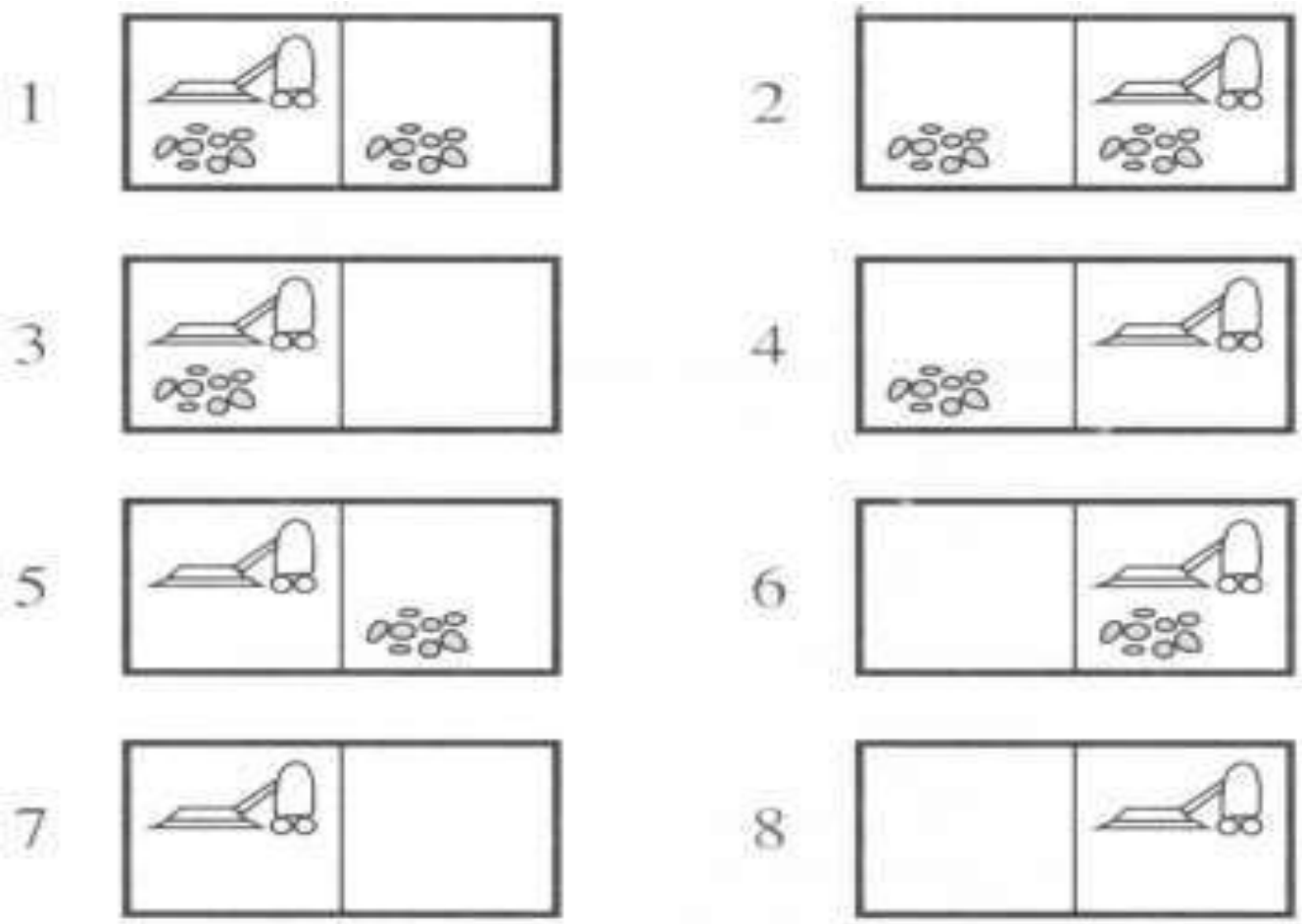


Example problems

- *Toy problems*
  - those intended to illustrate or exercise various problem-solving methods
  - E.g., puzzle, chess, etc.
- *Real-world problems*
  - tend to be more difficult and whose solutions people actually care about
  - E.g., Design, planning, etc.

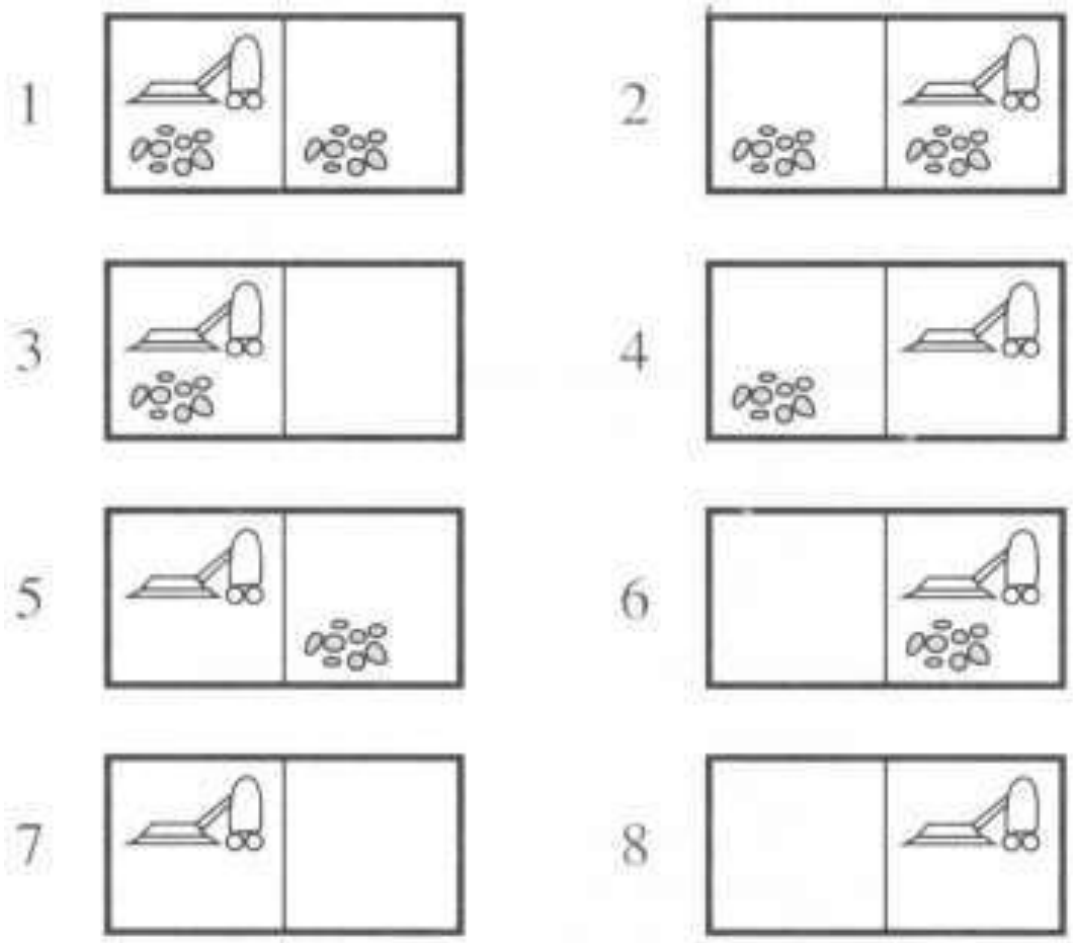
Toy problems

- Example: vacuum world

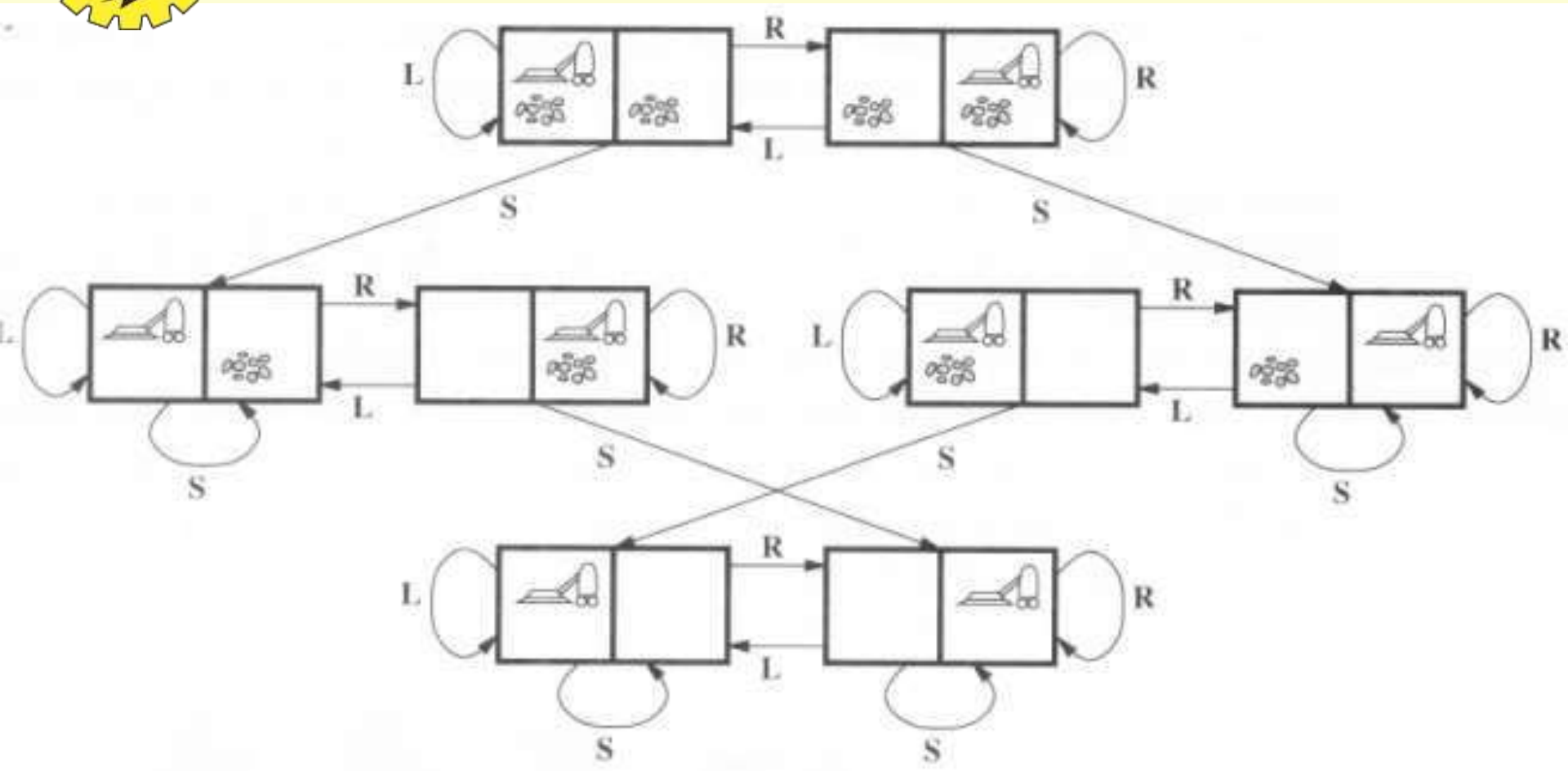


Toy problems

- Example: vacuum world



- Number of states: 8
- Initial state: Any
- Number of actions: 4
  - *left, right, suck, noOp*
- Goal: clean up all dirt
  - Goal states: {7, 8}
- Path Cost:
  - Each step costs 1


L
R
L
R
S
S
L
R
R
L
S
S
L
R
R
L
S
S
L
R
R
L
S
S

THANKYOU