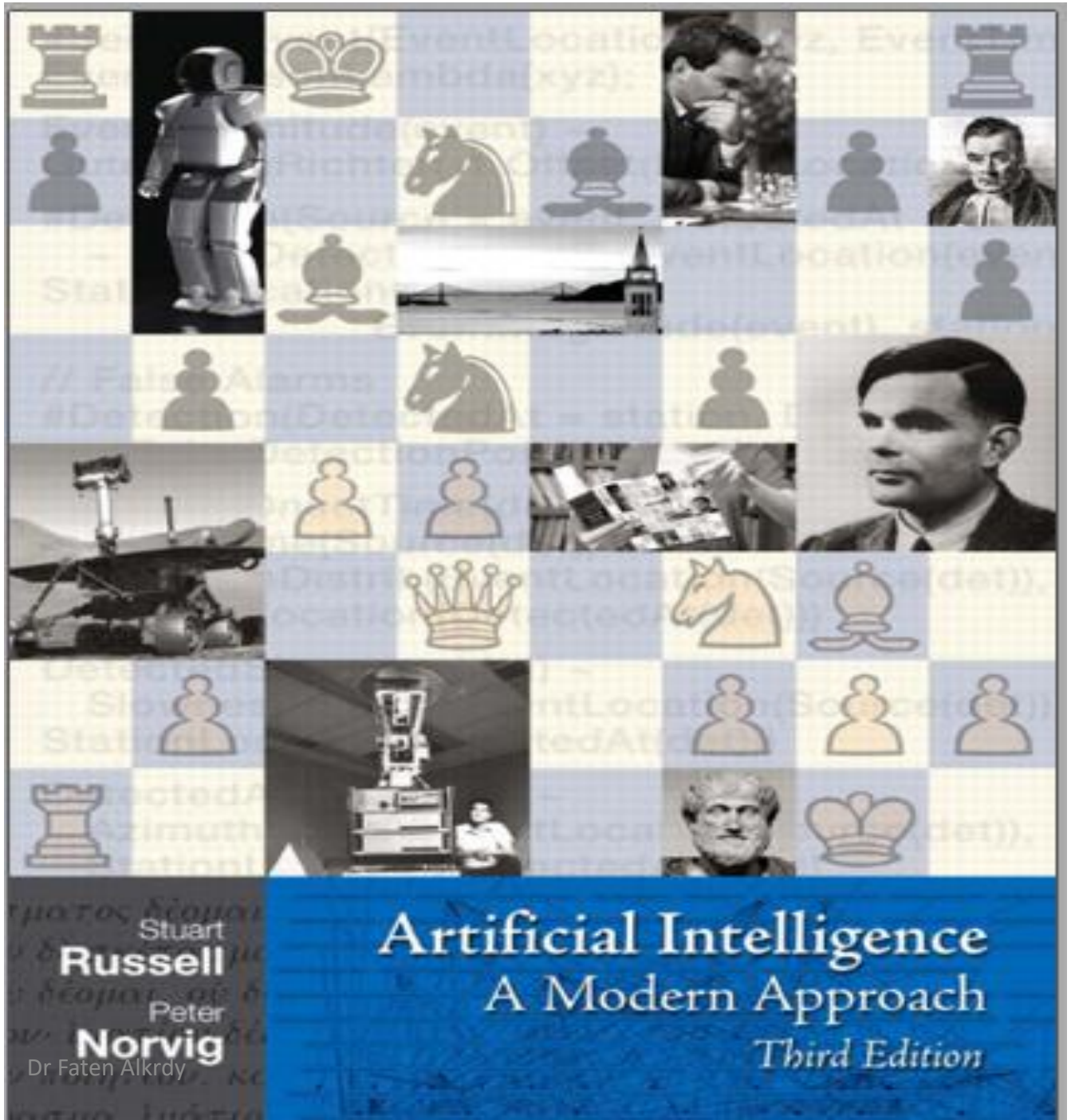
Search Algorithms

Artificial Intelligence
A Modern Approach

Stuart Russell and Peter Norvig



INTRODUCTION
What is AI(Artificial Intelligence)?

- The field of classical Artificial Intelligence (AI) coalesced in the 1950s drawing on an understanding of the brain from neuroscience, the new mathematics of information theory, control theory referred to as cybernetics, and the dawn of the digital computer.
- AI is a **cross-disciplinary** field of research that is generally concerned with developing and investigating systems that operate or act intelligently.
- It is considered a discipline in the field of computer science given the strong focus on computation.

INTRODUCTION

What is AI(Artificial Intelligence)?

- AI defines in 4 categories:

✓systems that think like humans:

✓systems that act like humans

✓systems that think rationally.

✓systems that act rationally.

INTRODUCTION
What is AI(Artificial Intelligence)?

✓ **systems that think like humans**:

✓ أن يكون للآلة عمليات تفكير وتحليل مشابهة للعقل البشري. هذا يركز على النموذج الداخلي وليس على النتيجة النهائية ويرتبط بمحاولة فهم الذكاء البشري نفسه.

✓ مثال شبكة عصبية صناعية تحلل صورة للتعرف على قطة تحاول محاكاة الدماغ البشري في معالجة المعلومات.

- **suggests systems** that model the cognitive information processing properties of humans, for example a general **problem solver** and systems that **build internal models** of their world .

✓ **systems that act like humans**

✓ **تتيح سلوكا خارجيا مشابه للسلوك البشري بغض النظر عن الطريقة التي تستخدمها. التركيز هنا على محاكاة السلوك**

✓ **مثال روبوت محادثة عندما تتحدث معه قد تجده يستخدم لغة طبيعية، يطرح الأسئلة ويبدو كأنه إنسان يعمل بمطابقة الأنماط دون فهم حقيقي**

- suggests that a system can do some specific things humans can do,
  this includes fields such as the Turing test, natural language processing, automated reasoning, knowledge representation, machine learning, computer vision, and robotics.

INTRODUCTION
What is AI (Artificial Intelligence)?

✓systems that think rationally

- suggests laws of rationalism and structured thought, such as syllogisms and formal logic.

- النظام يفكر بعقلانية أي تتبع الآلة قواعد الاستدلال والمنطق الصارمة للوصول إلى استنتاجات صحيحة بناء على مقدمات معينة. الهدف هو التفكير السليم وفقا لمبادئ المنطق والذي يختلف عن الطريقة الغير المنطقية التي يفكر بها بعض البشر أحيانا.
- مثال نظام خبير طبي اذا أدخلت الأعراض يستخدم النظام قاعدة معرفية ومنطقية ليستنتج التشخيص الأكثر منطقية.

✓systems that act rationally.

- suggests systems that do rational things such as expected utility maximization and rational agents.

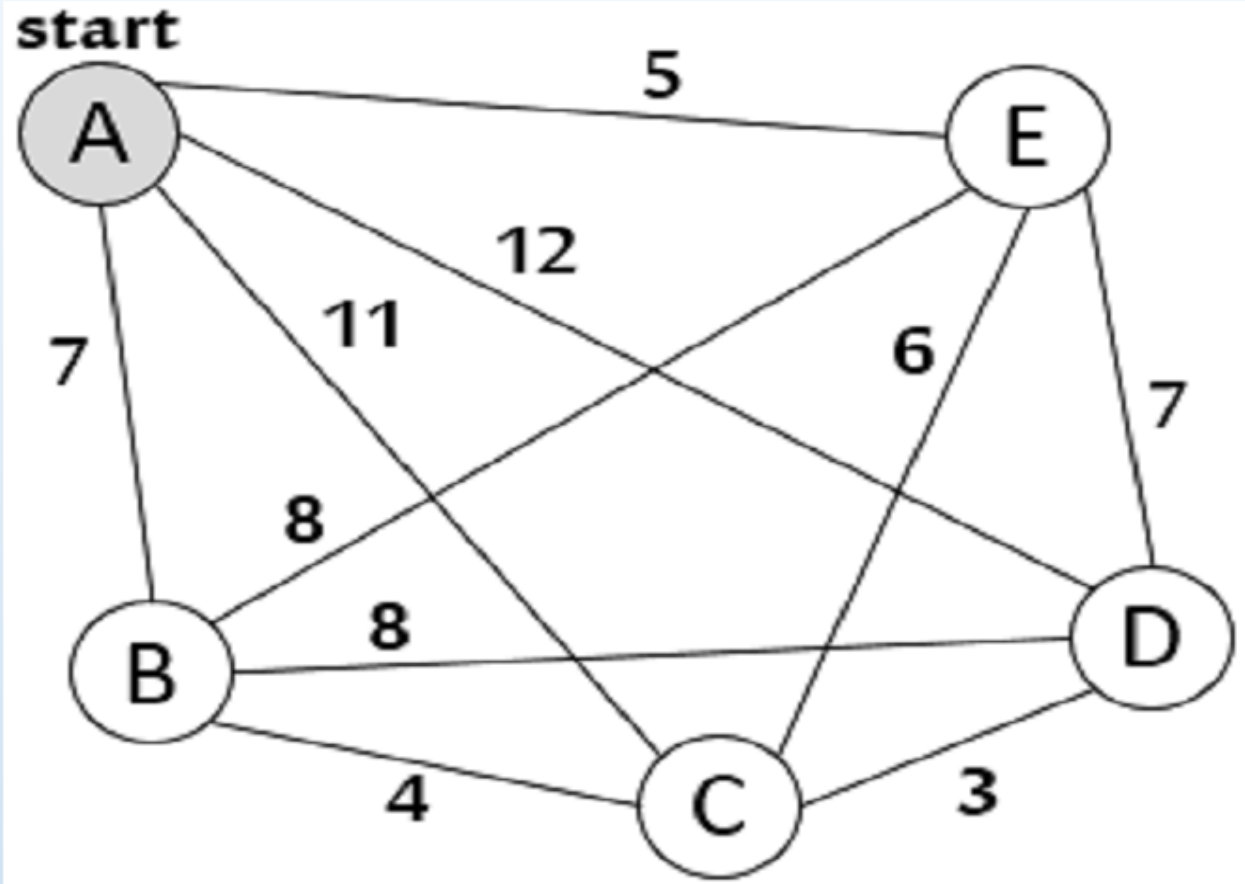
- النظام يتصرف بعقلانية يعني أن تتخذ الآلة القرار الأمثل لتحقيق هدف معين في بيئتها. لا يهم طريقة تفكيرها سواء كانت تشبه البشر أو لا. المهم هو السلوك العقلاني الذي يزيد من فرص النجاح.
- مثال سيارة ذاتية القيادة هدفها الوصول بأمان وسلاسة لتتصرف بطريقة عقلانية يجب أن تلتزم بقواعد المرور وتجنب العوائق (مثل كبح الفرامل بناء على بيانات المستشعرات حتى لو لم يكن هذا القرار هو الذي سيفعله سائق بشري عادي.

Examples of Problems in Artificial Intelligence

- In today's fast-paced digitized world, artificial intelligence techniques are used widely to automate systems that can use the resource and time efficiently.
- Some of the well-known problems experienced in everyday life are games and puzzles.
- Using AI techniques, we can solve problems efficiently.
- In this sense, **some of the most common problems resolved by AI** are:
  - Travelling Salesman Problem
  - Tower of Hanoi Problem
  - Water-Jug Problem
  - N-Queen Problem
  - Chess Sudoku
  - Crypt-arithmetic Problems Magic Squares
  - Logical Puzzles and so on.

Examples of Problems in Artificial Intelligence

TRAVELLING SALESMAN PROBLEM (TSP)



(**TSP**) هي مشكلة إيجاد مسار بائع متجول، و تهدف لإيجاد أقصر مسار يسلكه البائع،
ضمن الشروط:

1. الإنطلاق من مدينة ما و العودة إليها.
2. زيارة كل المُدن
3. زيارة كل مدينة لمرة واحدة فقط؛

Examples of Problems in Artificial Intelligence

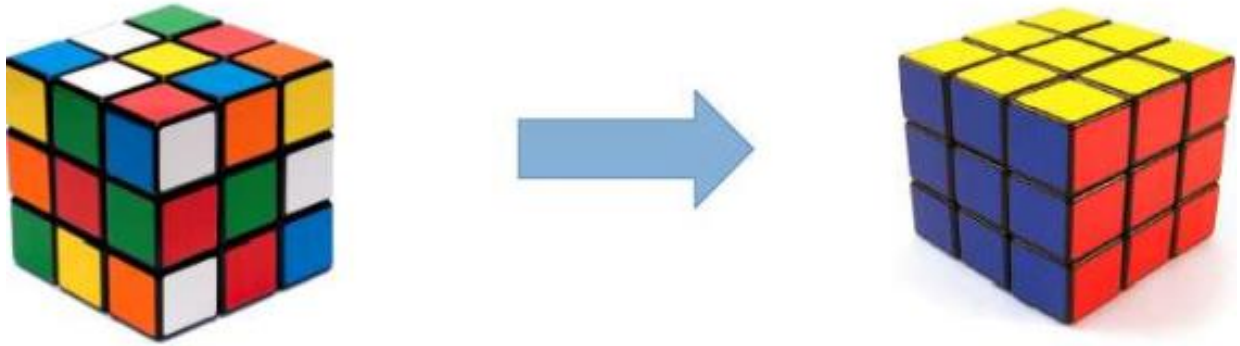
مشكلة إبريق الماء (Water Jug Problem)

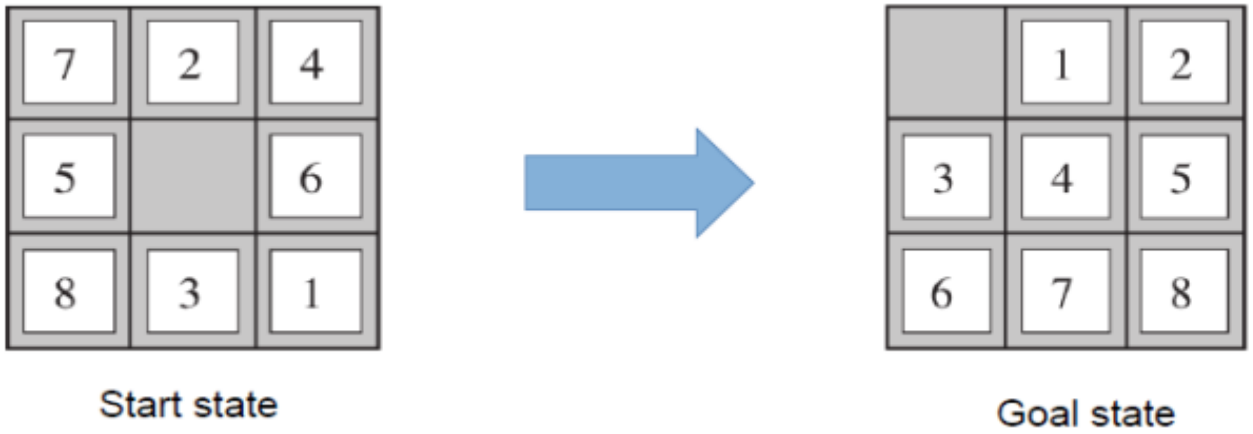
- سعة أحد الإباريق (3 لتر) و الآخر (4 لتر)
- ليس لديهما أي علامات لقياس كمية الماء فيهما.
- الأباريق فارغة.
- هناك مضخة و التي يمكن استخدامها لملء الأباريق بالماء.

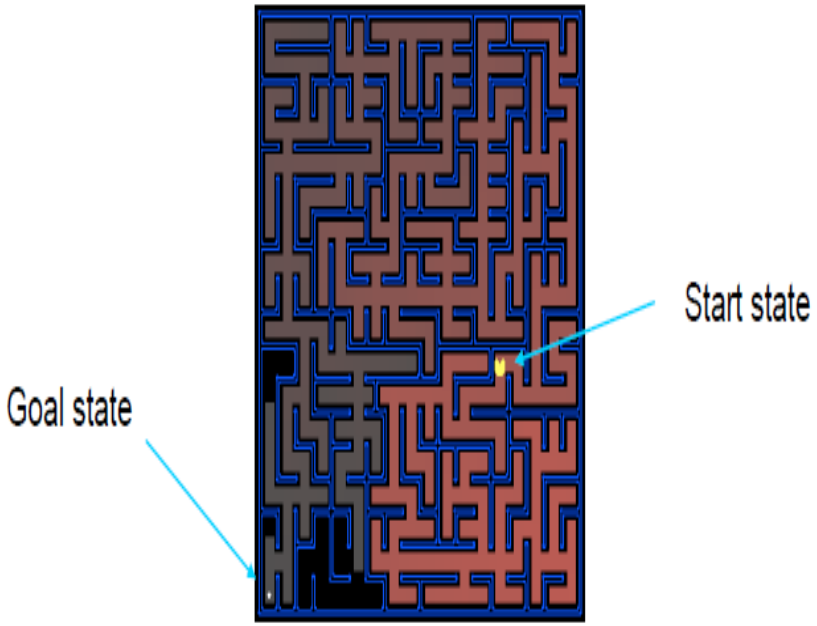
- كيف يمكنك الحصول على:
  (2 لتر) بالضبط من الماء في الإبريق (4 لتر).

- من أهم منهجيات حل المشاكل في الذكاء الصنعي تحويل المسألة الى مسألة البحث في فضاء الحالة والتي سنسميها لاحقا (البحث في بيان graph)

Problem Examples

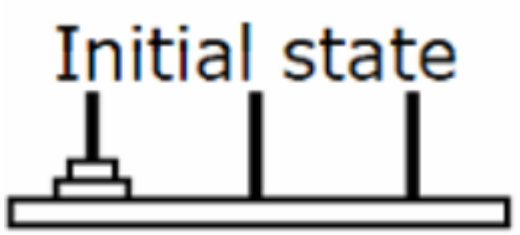


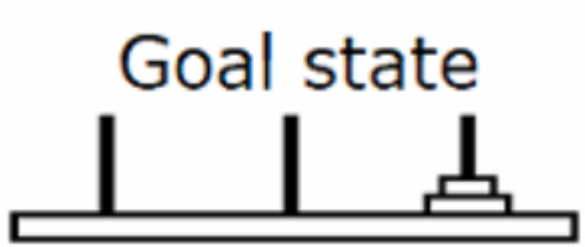




Problem Example Tower of Hanoi

- **مثال توضيحي**
  لعبة أبراج هانوي بقرصين اثنين
- حل المسألة التالية معطى الحالة الابتدائية والهدف

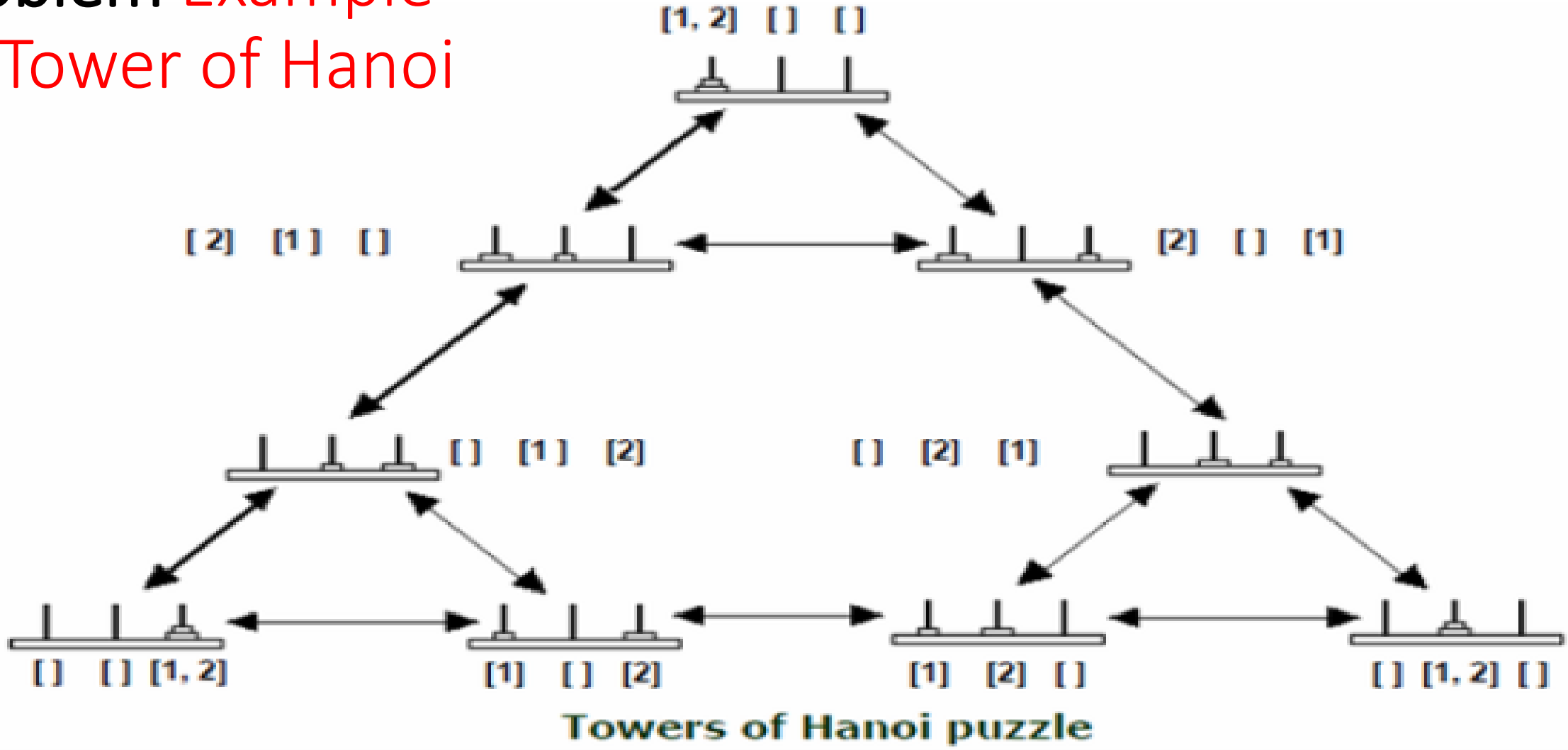




- الشروط:
- لا تضع القرص الأكبر فوق القرص الأصغر
- حرك قرص واحد في المرة من وتد إلى وتد آخر.
- الوتد الأوسط يمكن أن يستخدم
- المطلوب: نفذ اللعبة لتحقيق الهدف بأقل عدد ممكن من الخطوات

Problem Example
Tower of Hanoi

- فيما يلي جميع حالات الانتقال المحتملة في لعبة أبراج هانوي



Towers of Hanoi puzzle

- أقصر الحلول shortest solution هو متسلسلة من الحالات الانتقالية ابتداء من الحالة العليا في الشجرة (الحالة الابتدائية) لأسفل الشجرة وصولاً للجزء الأدنى يساراً

problem-solving using searching

- *In which we see how an agent can find a sequence of actions that achieves its goals when no single action will do.*
- The basic crux of artificial intelligence is to solve problems just like humans.

problem-solving using searching

- In artificial intelligence, problems can be solved by using searching algorithms, evolutionary computations, knowledge representations, etc.
- In general, searching is referred to as finding information one needs.
- The process of problem-solving using searching consists of the following steps.
  - **Define the problem**
  - **Analyze the problem**
  - **Identification of possible solutions**
  - **Choosing the optimal solution**
  - **Implementation**

problem-solving using searching

- Search can be defined as a problem-solving technique that enumerates a problem space from an initial position in search of a goal position (or solution).

- The manner in which the problem space is searched is defined by the **search algorithm** or **search strategy**.

- As search strategies offer different ways to enumerate the search space, how well a strategy works is based on the problem at hand.

❑ Outlines

- Problem-Solving Agent
- Problem Examples
- planning for solving problems
- basic search algorithms

Problem-Solving Agent

- A **problem-solving agent** is a *goal-based agent* and use *atomic representations.*
  - In atomic representations, states of the world are considered as wholes, with no internal structure visible to the problem solving algorithms.
  - *Intelligent agents* are supposed to maximize their *performance measure*. Achieving this is sometimes simplified if the agent can adopt a **goal** and aim at satisfying it.
  - **Problem formulation** is the process of deciding what actions and states to consider, given a *goal*.
  - The process of looking for a sequence of actions that reaches the goal is called **search**.
  - A ***search algorithm*** takes a problem as input and returns a **solution** in the form of an action
    sequence.
  - Once a *solution* is found, the carrying actions it recommends is called the **execution phase**.
  - A problem-solving agent has three phases:
  - **problem formulation, searching solution and executing actions in the solution.**

ما هو عامل حل المشكلة Problem-Solving Agent

- هو نوع من الوكلاء الأذكياء (Intelligent Agents) في الذكاء الاصطناعي، مصمم خصيصًا للوصول إلى هدف محدد من خلال البحث عن تسلسل الإجراءات الصحيحة التي تقوده من الحالة الأولية إلى حالة الهدف.
- كيف يعمل؟
- ١. صياغة المشكلة (Problem Formulation):
- · هذه هي أهم خطوة. يقوم العامل بتحويل هدفه إلى صياغة قابلة للحل. يتم ذلك بتحديد أربعة عناصر:
- · الحالة الأولية (Initial State): نقطة البداية.
- · الحالة الهدف (Goal State): ما يريد تحقيقه.
- · الإجراءات (Actions): الخيارات أو العمليات المتاحة في كل حالة.
- · دالة الانتقال (Transition Model): تصف نتيجة تطبيق إجراء على حالة معينة.
- ٢. البحث عن الحل (Search):
- · لا يعرف العامل الحل مسبقًا، لذلك يستخدم خوارزميات بحث لاستكشاف فضاء الحالات (State Space) – وهو كل الحالات الممكنة.
- · تبحث الخوارزمية عن مسار (تسلسل من الإجراءات) يصل من الحالة الأولية إلى إحدى حالات الهدف.
- ٣. تنفيذ الحل (Solution Execution):
- · بعد العثور على المسار، يقوم العامل بتنفيذ تسلسل الإجراءات خطوة بخطوة في العالم الحقيقي (أو البيئة المحاكاة) لتحقيق الهدف.

- مثال توضيحي:
-
- . المشكلة: لعبة الـ ٨-Puzzle (اللغز المنزلق).
- . الحالة الأولية: ترتيب عشوائي للقطع.
- . الحالة الهدف: ترتيب القطع بشكل صحيح.
- . الإجراءات: تحريك الفراغ لليسار، لليمين، لأعلى، أو لأسفل.
- . البحث: تجربة جميع التحركات الممكنة حتى يتم الوصول إلى الترتيب الصحيح.
-
- الخلاصة:
-
- عامل حل المشكلة هو كيان يتبع منهجية "التفكير أولاً، ثم التصرف" (Think then act). فهو يصمم خطة قبل التنفيذ، مما يجعله فعالاً في التعامل مع المشاكل المعقدة التي تتطلب تخطيطًا.

- ملاحظة: الفرق بين **problem-solving agent** is a *goal-based agent*
- *goal-based agent هو عامل ذكي يكون قراره وتقييمه للإجراءات مبنيا على مدى قربها او بعدها عن الهدف.*
- *يستخدم طرقا مختلفة لتحقيق الهدف وليس بالضرورة عبر صياغة المشكلة والبحث قد يسخدم قواعد كلاسيكية if then او تعلم الة او أي إجراء اخر.*
- **problem-solving agent لديه منهجية محددة وصارمة لتحقيق الهدف تعتمد على صياغة المشكلة تحديد الحالة الابتدائية والنهائية والإجراءات**
- ***يسخدم خوارزميات بحث للعثور على تسلسل الإجراءات او الخطة***
- ***تنفيذ الخطة***

Problem-Solving Agent
Well-defined problems

A **problem** can be defined formally by four components:

1. **initial state**

   that the agent starts in. For example, the initial state for our agent in Romania might be described as In(Arad).

   Actions

   A description of the possible actions available to the agent. Given a particular state s,

   For example, from the state In(Arad), the applicable actions are {Go(Sibiu), Go(Timisoara), Go(Zerind)}

Problem-Solving Agent
Well-defined problems

2. transition model

- specified by a function that returns the state that results from SUCCESSOR doing action in old state.

$$state \rightarrow (Action, state)$$

- We also use the term **successor function** to refer to any state reachable from a given state by a single action. For example, we have

$$in(Arad) \rightarrow ((Go(Timisora), in(Tamisora)$$
$$in(Arad) \rightarrow (Go(Zerind), in(Zerind)$$

  - path in the state space is a sequence of states connected by a sequence of actions.

Problem-Solving Agent
Well-defined problems
3. Goal test

- which determines whether a given state is a goal state.
- Sometimes there is an explicit set of possible goal states, and the test simply checks whether the given state is one of them.
  - The agent's goal in Romania is the single set {In(Bucharest)}.

- Sometimes the goal is specified by an **abstract** property rather than an explicitly enumerated set of states.

  - For example, in chess, the goal is to reach a state called "checkmate," where the opponent's king is under attack and can't escape.
  - In Romania Abstract goal to be in Bucharest (when we take only the distance).

Problem-Solving Agent
Well-defined problems

4. path cost function

- that assigns a numeric cost to each path.
- The problem-solving agent chooses a cost function that reflects its own performance measure.
  - For the agent trying to get to Bucharest, time is of the essence, so the cost of a path might be its length in kilometers.
  - the cost of a path can be described as the sum of the costs of the individual actions along the path.

- The step cost of taking action a in state (s) to reach state (s') is denoted by c(s, a, s'). step costs are nonnegative.

- **solution:** solution to a problem is an action sequence that leads from the initial state to a goal state.
  - Solution quality is measured by the path cost function.
  - optimal solution has the lowest path cost among all solutions.

Problem Example
Travelling in Romania

- On holiday in Romania; currently in Arad.
  - Flight leaves tomorrow from Bucharest

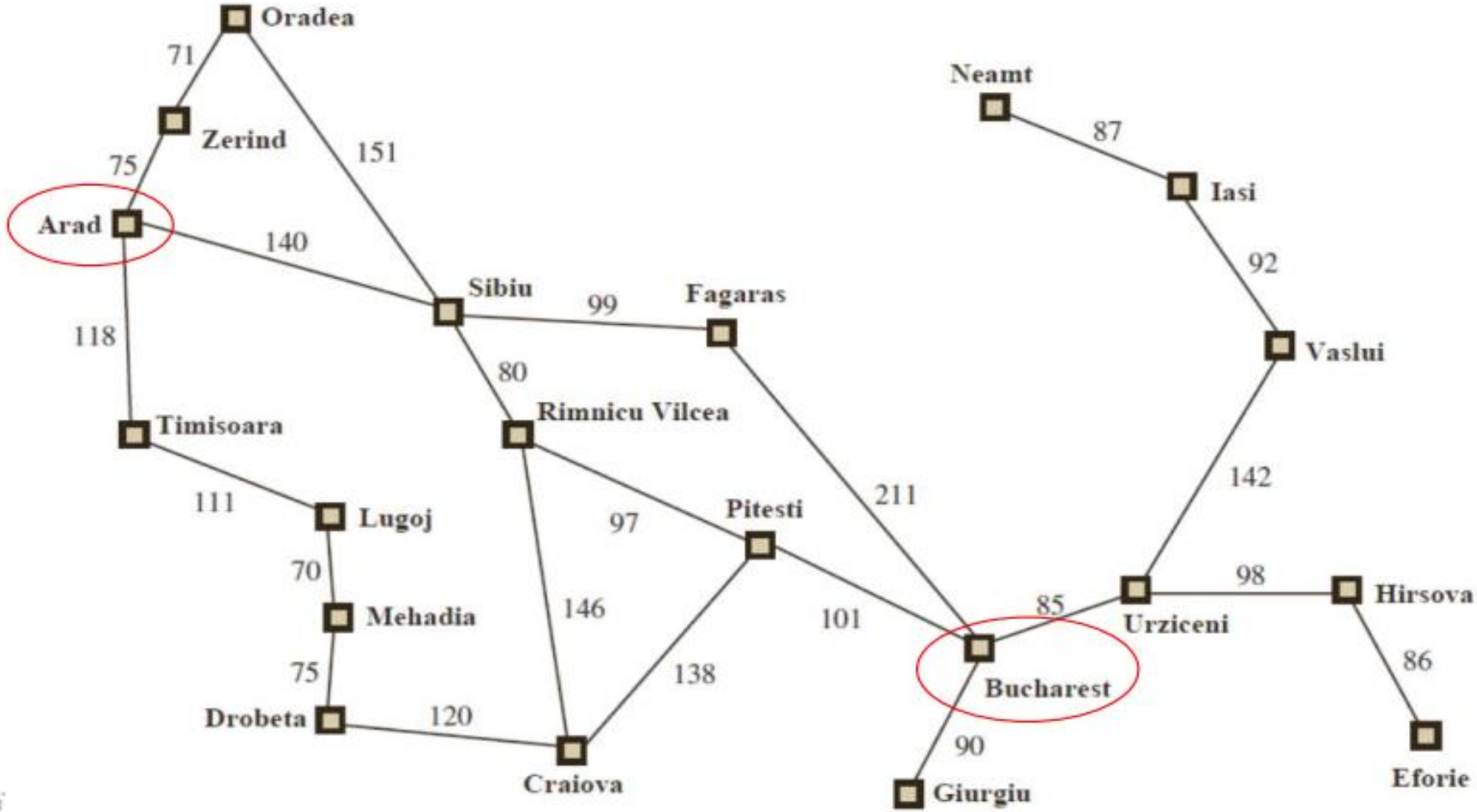
  Formulate goal:
  - be in Bucharest

  Formulate problem:
  - states: various cities
  - actions: drive between cities

  Find solution:
  - sequence of cities, e.g., Arad, Sibiu, Fagaras, Bucharest

Problem Example

Travelling in Romania: A simplified road map



Problem Example
Travelling in Romania

- Travelling in Romania problem can be defined by:
  initial state: in Arad
  actions: transition model: is the graph
  successor function S(x) = set of action{state pairs
  e.g., S(Arad) = { <Arad → Zerind,Zerind>, <Arad → Sibiu,Sibiu>,
  < Arad → Timisoara,Timisoara> }
  goal test: in Bucharest
  path cost: sum of distances, number of actions executed, etc.
  c(x; a; y) is the **step cost**, assumed to be ≥ 0
- A **solution** is a sequence of actions leading from the *initial state* to a *goal state*

Problem Example

vacuum world

- **States**:
  - The state is determined by both the agent location and the dirt locations.
  - The agent is in one of two locations, each of which might or might not contain dirt.
  - Thus, there are $2 \times 2^2 = 8$ possible world states.

Initial state: Any state can be designated as the initial state.
Actions: Each state has just three actions: *Left*, *Right*, and *Suck*.
Transition model:
- The actions have their expected effects, except that moving *Left* in the leftmost square, moving *Right*
in the rightmost square, and *Suck*ing in a clean square have no effect.
- The transition model defines a state space.

Goal test: This checks whether all the squares are clean.
Path cost: Each step costs 1, so the path cost is the number of steps in the path.

Problem formulate
Vacuum cleaner Example

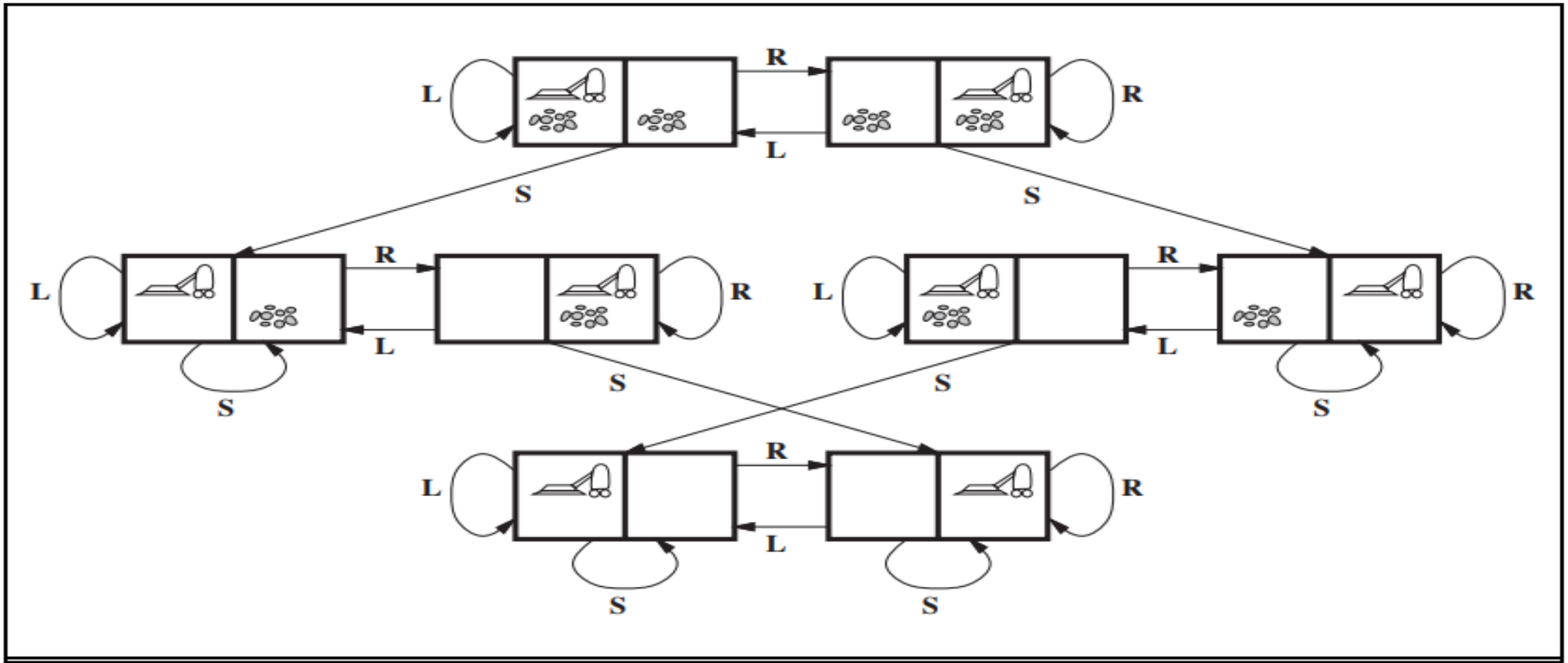


Figure 3.3 The state space for the vacuum world. Links denote actions: L = *Left*, R = *Right*, S = *Suck*.

Problem formulate
Vacuum cleaner Example

- **States**: The state is determined by both the agent **location and the dirt** locations.

The agent is in one of two locations, each of them might or might not contain dirt.

there are $2 \times 2^2 = 8$ possible world states. A larger environment with n locations has $n \cdot 2^n$ states.

- **Initial state**: Any state can be designated as the initial state.
- **Actions: Left, Right, and Suck.** Larger environments might also include *Up* and *Down*.
-

Problem formulate
Vacuum cleaner Example

- **Transition model**: The actions have their expected effects, except that moving *Left* in the leftmost square, moving *Right* in the rightmost square, and *Suck*ing in a clean square have no effect. The complete state space is shown in Figure 3.3.

$$(A, dirt, clean) \rightarrow ('left', (A, dirty, clean)$$

$$(A, dirt, clean) \rightarrow ('Right', (B, dirty, clean)$$

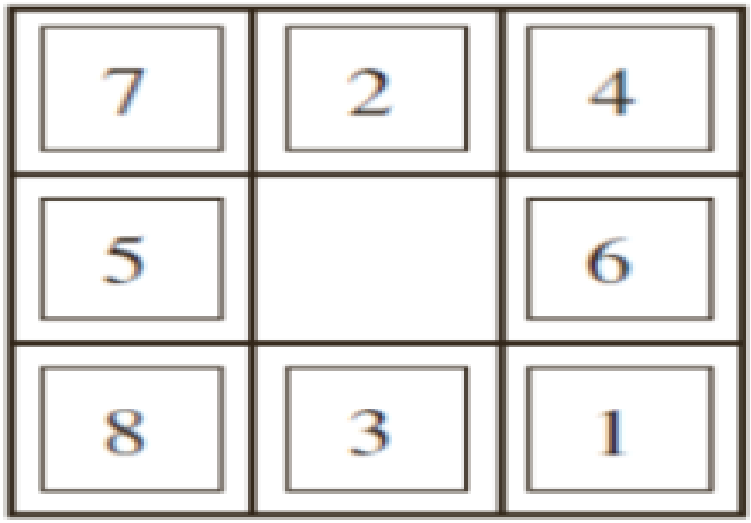
$$(A, dirt, clean) \rightarrow ('suck', (A, dirty\ clean)$$

- **Goal test**: This checks whether all the squares are clean.
- **Path cost**: Each step costs 1, so the path cost is the number of steps in the path.

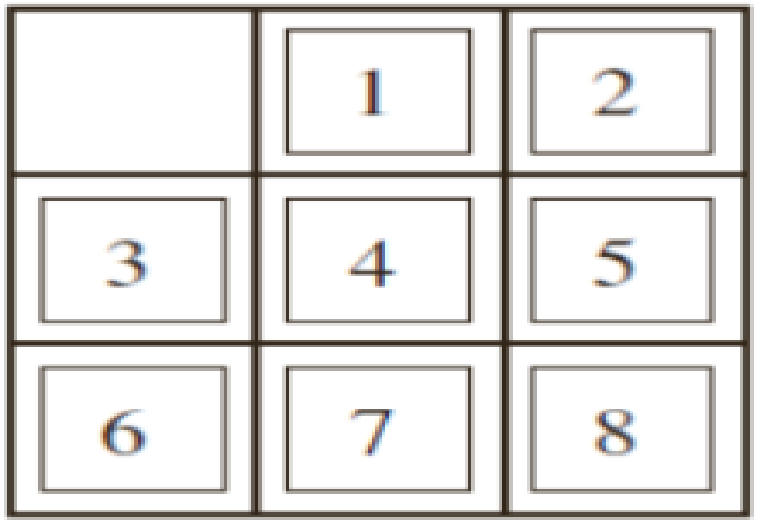
Problem Example

8-puzzle

- The **8-puzzle** consists of a 3×3 board with eight numbered tiles and a blank space.
  - A tile adjacent to the blank space can slide into the space.
  - The object is to reach a specified goal state.



Start State



Goal State

Problem Example

8-puzzle

- **States:** A state specifies the location of each of the eight tiles and the blank in one of the nine squares.
  Initial state: Any state can be designated as the initial state.
  • Note that goal can be reached from exactly half of the possible initial states.
  Actions: Movements of the blank space ***Left***, ***Right***, ***Up***, or ***Down***.
  • Different subsets of these are possible depending on where the blank is.
  Transition model: Given a state and action, this returns the resulting state;
  Goal test: This checks whether the state matches the goal configuration
  Path Cost: Each step costs 1, so the path cost is the number of steps in the path.

Problem Example

8-puzzle

- The 8-puzzle belongs to the family of **sliding-block puzzles**,
  - This family is known to be **NP-complete**.
  - Optimal solution of n-Puzzle family is NP-hard. ie NO polynomial solution for the problem.
  - The 8-puzzle has 9!/2=181, 440 reachable states.
  - The 15-puzzle (on a 4×4 board) has around 1.3 trillion states,
  - The 24-puzzle (on a 5 × 5 board) has around 10^{25} states