Computer Science Guidance

Jianhui Zhang,

Ph.D., Associate Prof.

College of Computer Science and Technology, Hangzhou Dianzi Univ.

Email: jh_zhang@hdu.edu.cn

Addison-Wesley
is an imprint of
PEARSON

Chapter 11: Artificial Intelligence

Computer Science: An Overview
Twelfth Edition

by
J. Glenn Brookshear
Dennis Brylow

Addison-Wesley is an imprint of PEARSON

Chapter 11: Artificial Intelligence

- 11.1 Intelligence and Machines
- 11.2 Perception
- 11.3 Reasoning
- 11.4 Additional Areas of Research
- 11.5 Artificial Neural Networks
- 11.6 Robotics
- 11.7 Considering the Consequences

What is artificial intelligence?

- Although there is no universally-agreed definition of artificial intelligence, we accept the following definition that matches the topics covered here
- Artificial intelligence is the study of programmed systems that can simulate, to some extent, human activities such as perceiving, thinking, learning and acting.

A brief history of artificial intelligence

- Although artificial intelligence as an independent field of study is relatively new, it has some roots in the past.
- We can say that it started 2,400 years ago when the Greek philosopher Aristotle invented the concept of logical reasoning.
- The effort to finalize the language of logic continued with Leibniz and Newton.
- George Boole developed Boolean algebra in the nineteenth century that laid the foundation of computer circuits.
- However, the main idea of a thinking machine came from Alan Turing, who proposed the Turing test.
- The term “artificial intelligence” was first coined by John McCarthy in 1956.

Intelligent Agents

- **Agent:** A “device” that responds to stimuli from its environment
  - Sensors
  - Actuators
- Much of the research in artificial intelligence can be viewed in the context of building agents that behave intelligently

Levels of Intelligent Behavior

- Reflex: actions are predetermined responses to the input data
- More intelligent behavior requires knowledge of the environment and involves such activities as:
  - Goal seeking
  - Learning: procedural vs declarative knowledge

Figure 11.1 The eight-puzzle in its solved configuration

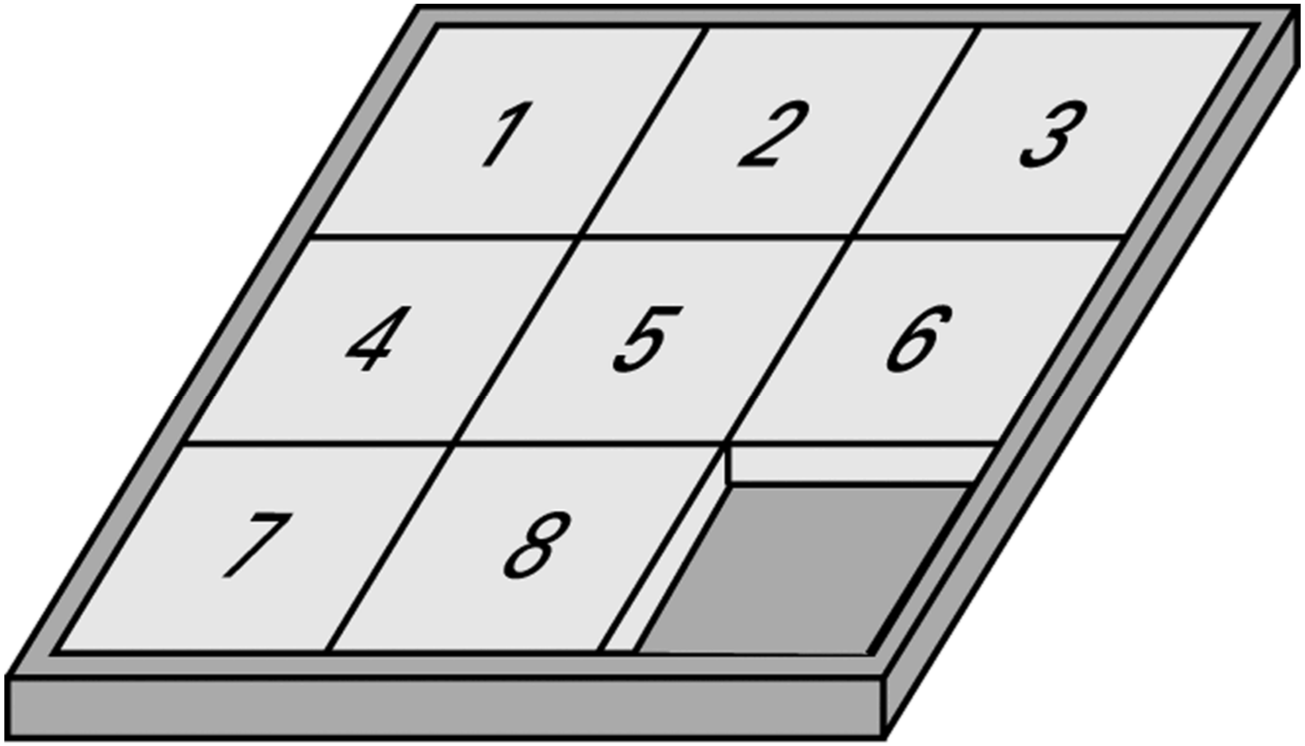
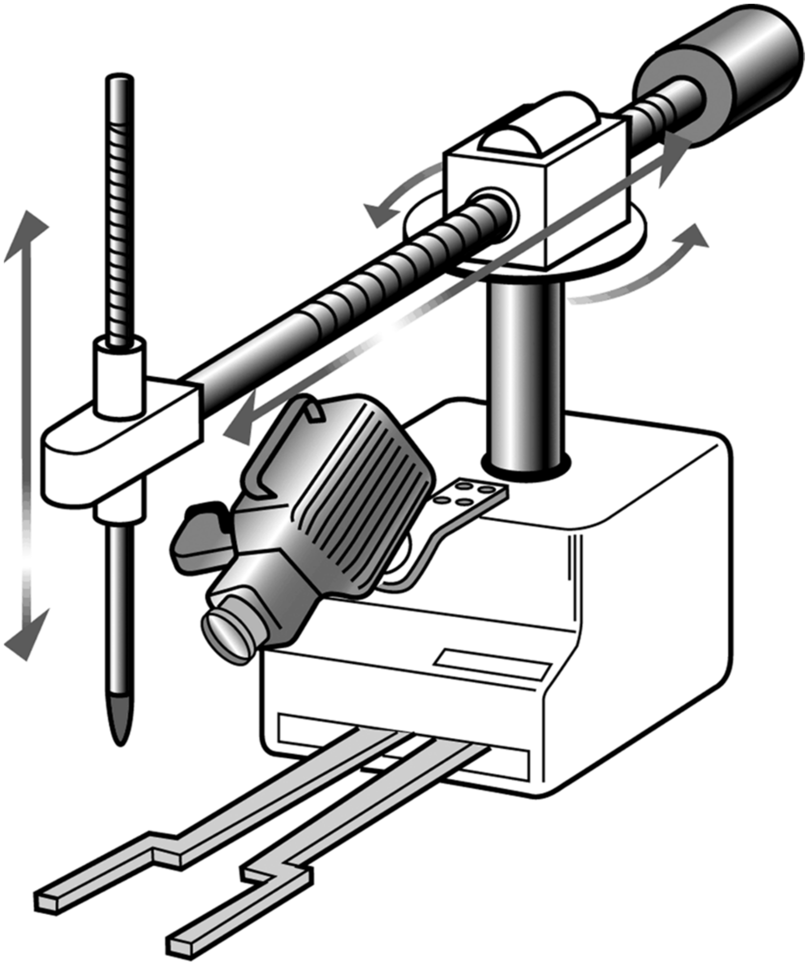


Figure 11.2 Our puzzle-solving machine



C:\teaching\
斗学基础\ppt\身手

Approaches to Research in Artificial Intelligence

- Engineering track
  - Performance oriented
  - Try to develop systems that exhibit intelligent behaviour
- Theoretical track
  - Simulation oriented
  - Try to develop a computational understanding of animal especially human intelligence
- Natural language Processing vs Linguistics

Turing Test

- In 1950, Alan Turing proposed the Turing test, which provides a definition of intelligence in a machine.
- The test simply compares the intelligent behavior of a human being with that of a computer.
- An interrogator asks a set of questions that are forwarded to both a computer and a human being.
- The interrogator receives two sets of responses, but does not know which set comes from the human and which set from the computer.
- After careful examination of the two sets, if the interrogator cannot definitely tell which set has come from the computer and which from the human, the computer has passed the Turing test for intelligent behavior.

Turing Test

- Test setup: Human interrogator communicates with test subject by typewriter.
- Test: Can the human interrogator distinguish whether the test subject is human or machine?

Perception

- Understanding what is received through the senses—sight, hearing, touch, smell, taste.
- A human being sees a scene through the eyes, and the brain interprets it to extract the type of objects in the scene.
- A human being hears a set of voice signals through the ears, and the brain interprets it a meaningful sentence, and so on.
- Two areas of research in perception:
  - Understanding image
  - Understanding language

Techniques for Understanding Images

- Template matching
- Image processing
  - Identifying characteristics
  - edge enhancement
  - region finding
  - smoothing
- Image analysis
  - Identifying the meaning of the features

Language Processing

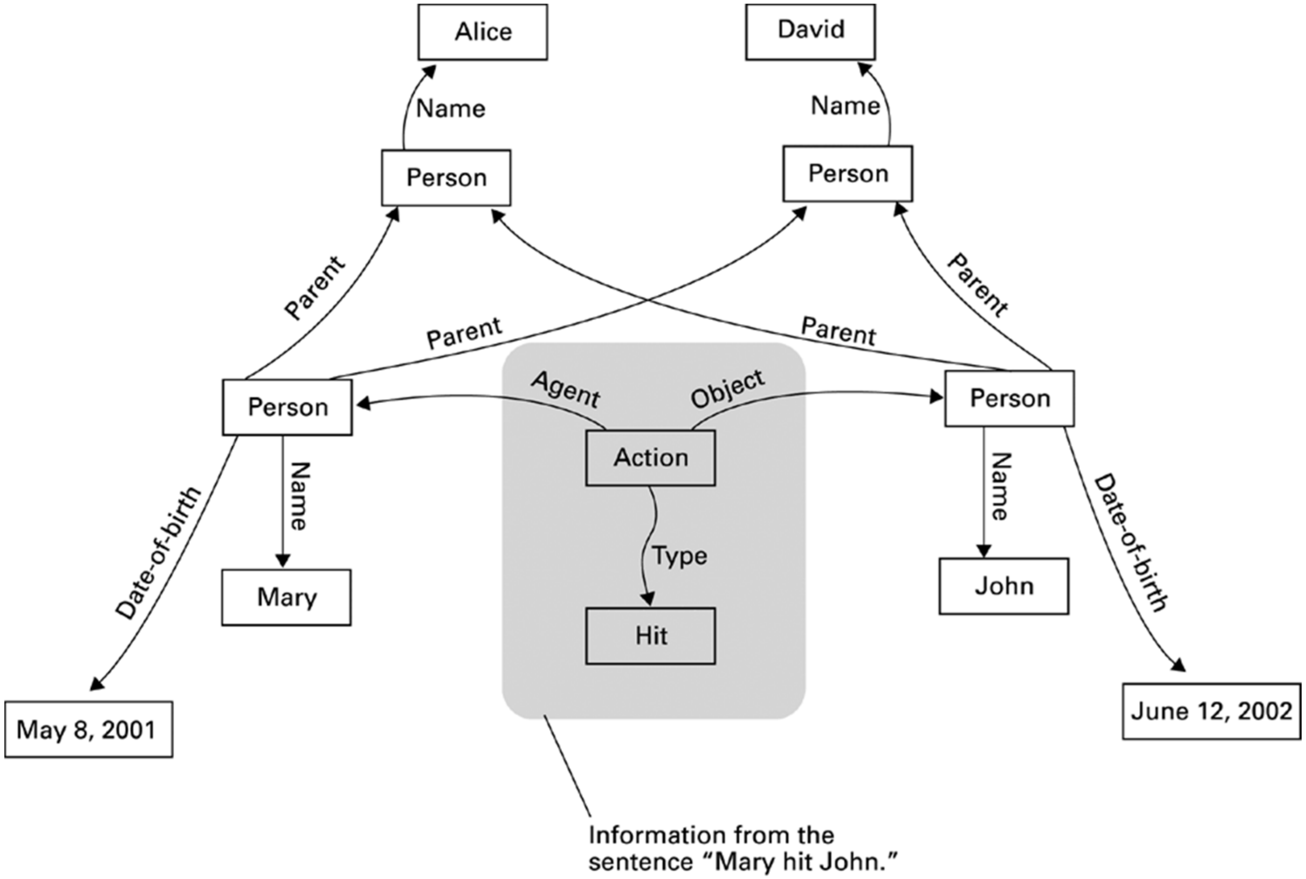
- Syntactic Analysis
  - Parsing
  - Identify the grammatical role of each word
- Semantic Analysis
  - Identify the semantic role each word
- Contextual Analysis
  - remove ambiguities
  - “The bat fell to the ground”

Knowledge Representation

- If an artificial agent is supposed to solve some problems related to the real world, it somehow related to the real world, needs to be able to represent knowledge.
- Facts are represented as data structures that can be manipulated by programs stored inside the computer.
- Four common methods for representing knowledge: semantic networks, frames, predicate logic and rule based systems.

Figure 11.3 A semantic net

A semantic network uses directed graphs to represent knowledge.

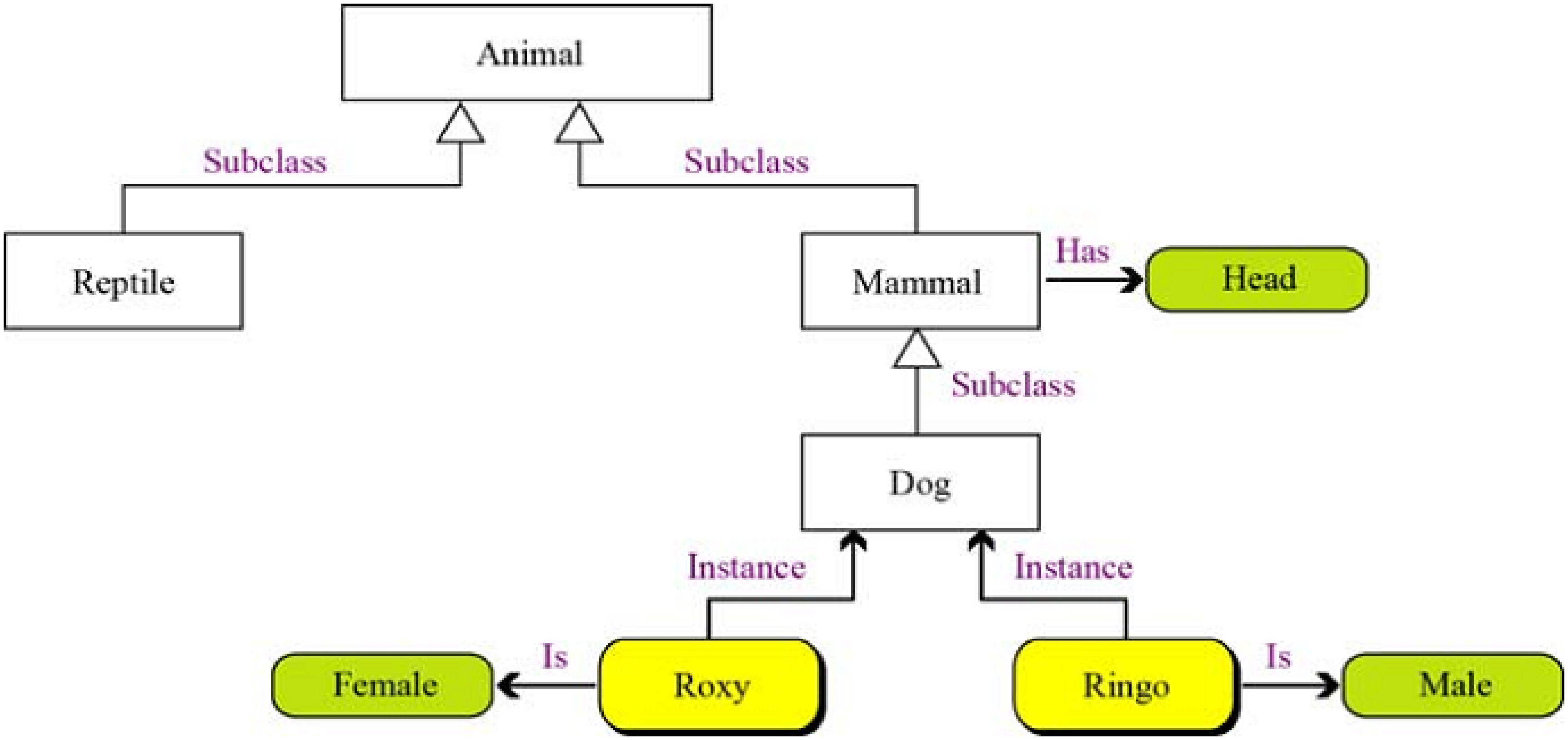


Information from the sentence "Mary hit John."

Semantic net

- Concepts
  - A concept can be thought of as a set or a subset.
  - For example, animal defines the set of all animals, horse defines the set of all horses and is a subset of the set animal.
- Relations
  - In a semantic network, relations are shown by edges.
  - An edge can define a subclass relation, an instance relation attribute, an object (color, size, …), or a property of an object.

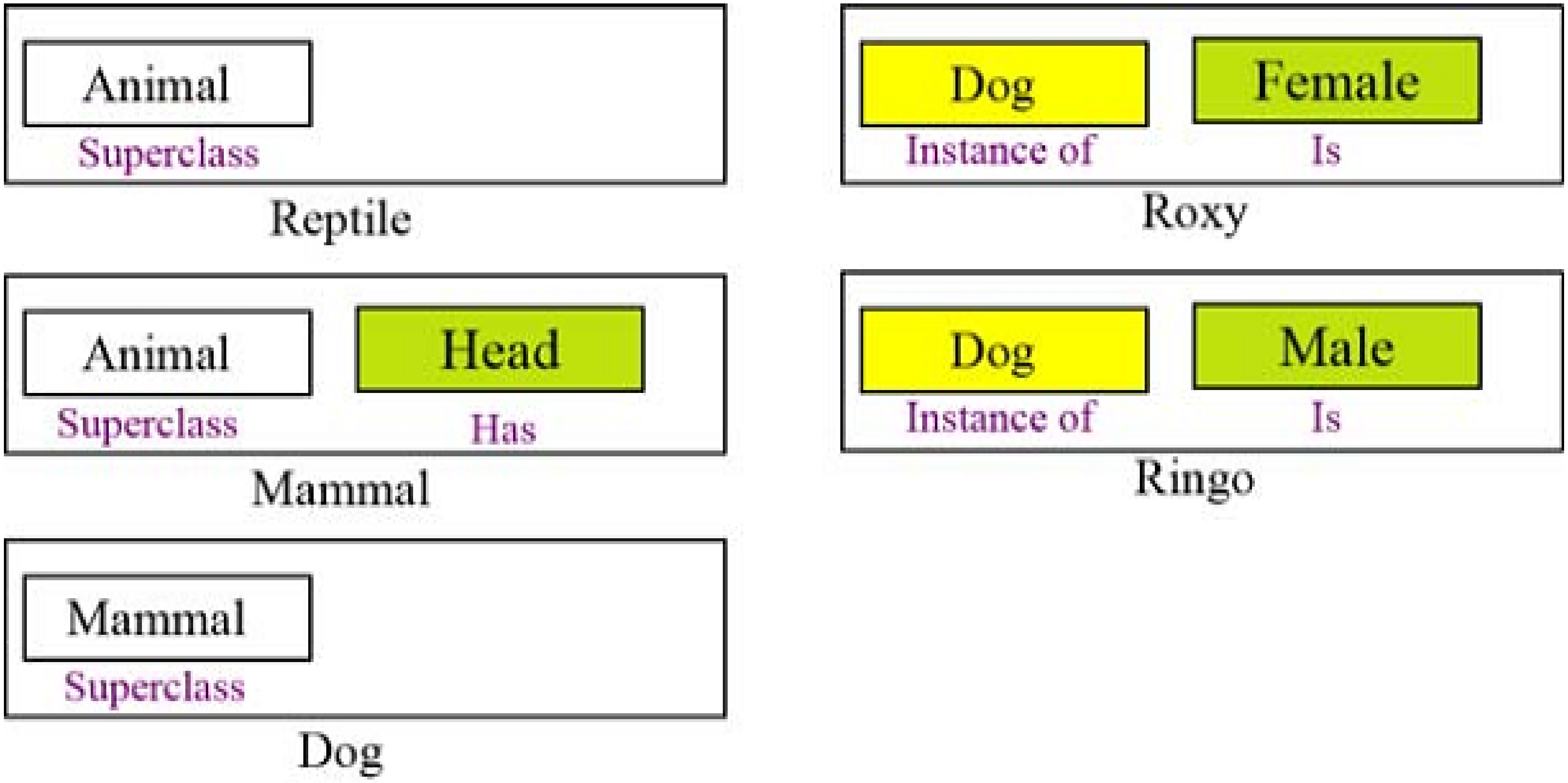
Another simple semantic network



Frames

- Frames are closely related to semantic networks.
- In frames, data structures (records) are used to represent the same knowledge.
- One advantage of frames over semantic networks is that programs can handle frames more easily than semantic networks.

A set of frames representing the previous semantic network



Frames

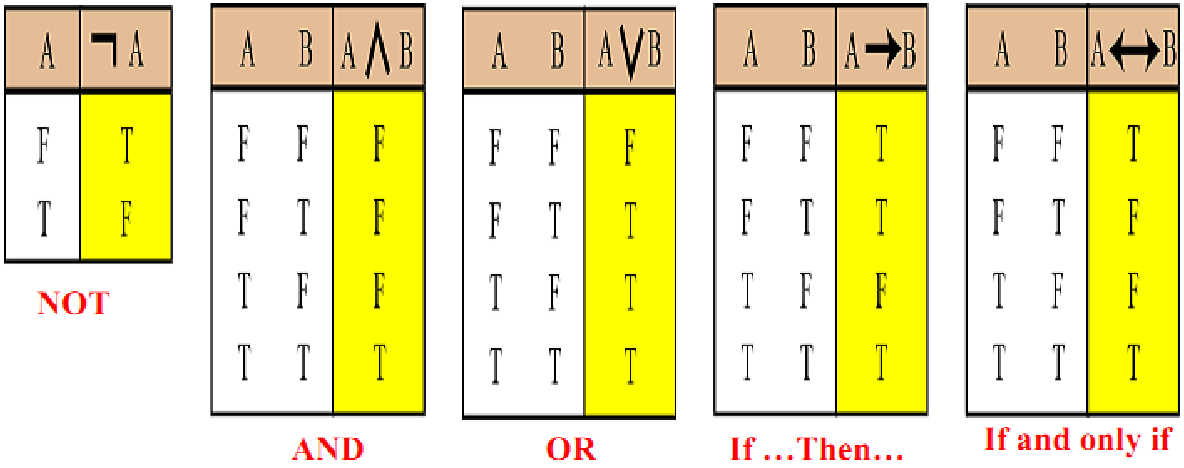
- Objects
- A node in a semantic network becomes an object in a set of frames, so an object can define a class, a subclass or an instance of a class. For example, reptile, mammal, dog, Roxy and Ringo are objects.
- Slots
- Edges in semantic networks are translated into slots—fields in the data structure.
- The name of the slot defines the type of the relationship and the value of the slot completes the relationship. For example, animal is a slot in the reptile object.

Predicate logic

- The most common knowledge representation
- Predicate logic can be used to represent complex facts.
- It is a well-defined language developed via a long history of theoretical logic.
- Propositional logic: simpler language

Propositional logic

- A language made up from a set of sentences that can be used to carry out logical reasoning about the world.
- Propositional logic uses five operators.



A	¬A
F	T
T	F

NOT

A	B	A∧B
F	F	F
F	T	F
T	F	F
T	T	T

AND

A	B	A∨B
F	F	F
F	T	T
T	F	T
T	T	T

OR

A	B	A→B
F	F	T
F	T	T
T	F	F
T	T	T

If …Then…

A	B	A↔B
F	F	T
F	T	F
T	F	F
T	T	T

If and only if

sentence

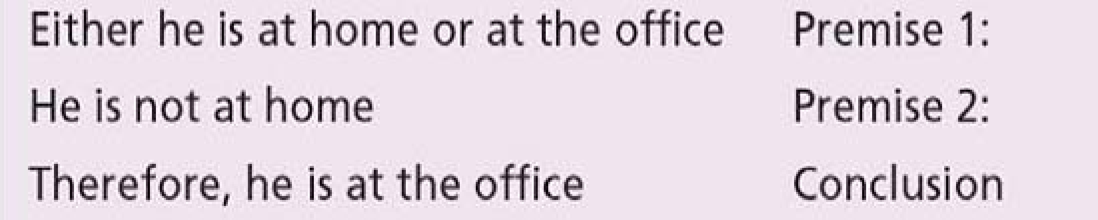
- A sentence is defined recursively as shown below:
  - An uppercase letter, such as A, B, S or T, that represents a statement in a natural languages, is a sentence.
  - Any of the two constant values (true and false) is a sentence.
  - If P is a sentence, then ¬P is a sentence.
  - If P and Q are sentences, then P∨Q, P∧Q, P→Q and P↔Q are sentences.

Examples

- The following are sentences in propositional language:
  - Today is Sunday (S).
  - It is raining (R).
  - Today is Sunday or Monday ($S \vee M$).
  - It is not raining ($\neg R$)
  - If a dog is a mammal then a cat is a mammal ($D \rightarrow C$).

Deduction

- In AI we need to create new facts from the existing facts. In propositional logic, the process is called deduction.
- Given two presumably true sentences, we can deduce a new true sentence. For example:

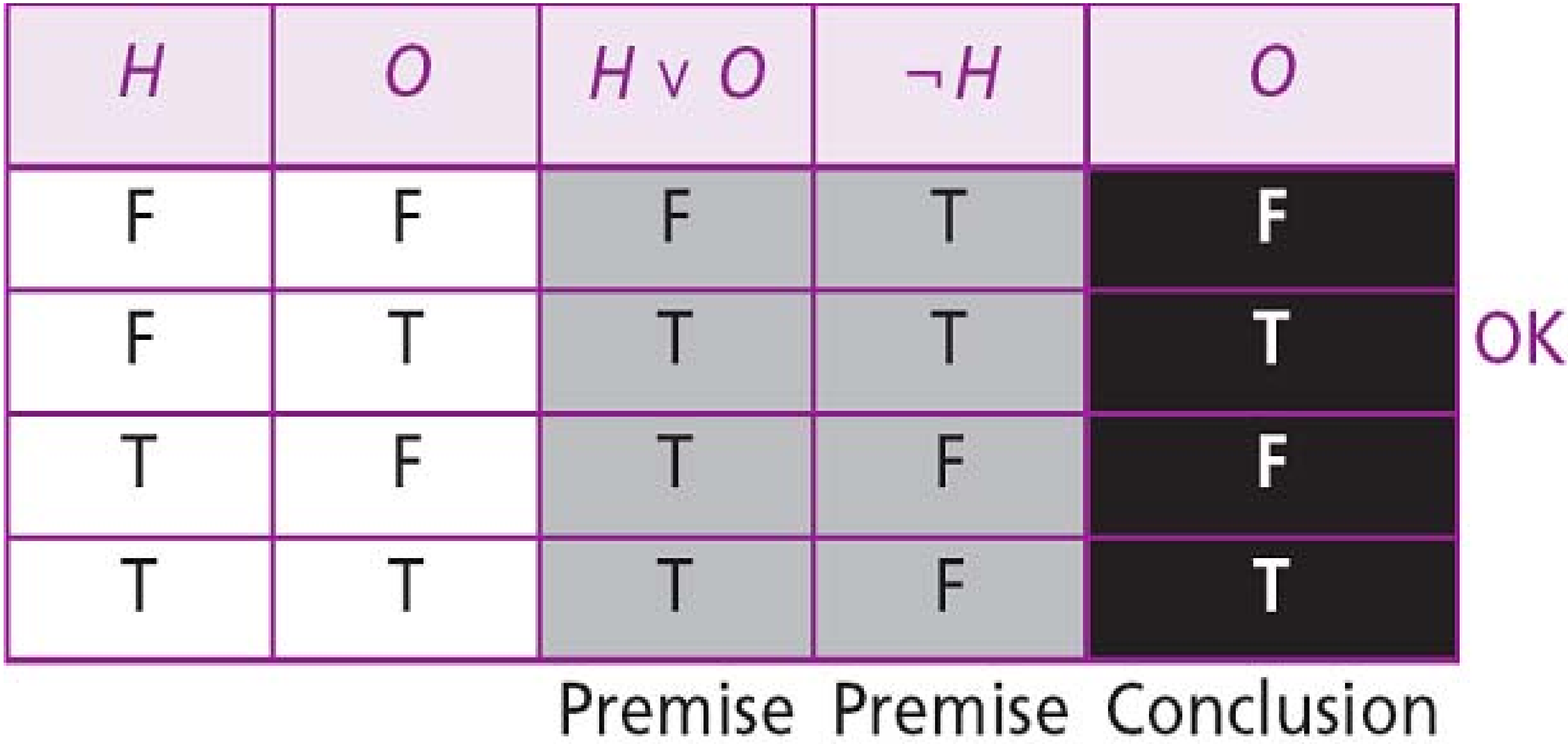


Either he is at home or at the office	Premise 1:
He is not at home	Premise 2:
Therefore, he is at the office	Conclusion

$$\{H \vee O,\ \neg H\} \vdash O$$

- One way to find if an argument is valid is to create a truth table for the premisses and the conclusion.
- A conclusion is invalid if we can find a counterexample case: a case in which both premisses are true, but the conclusion is false.

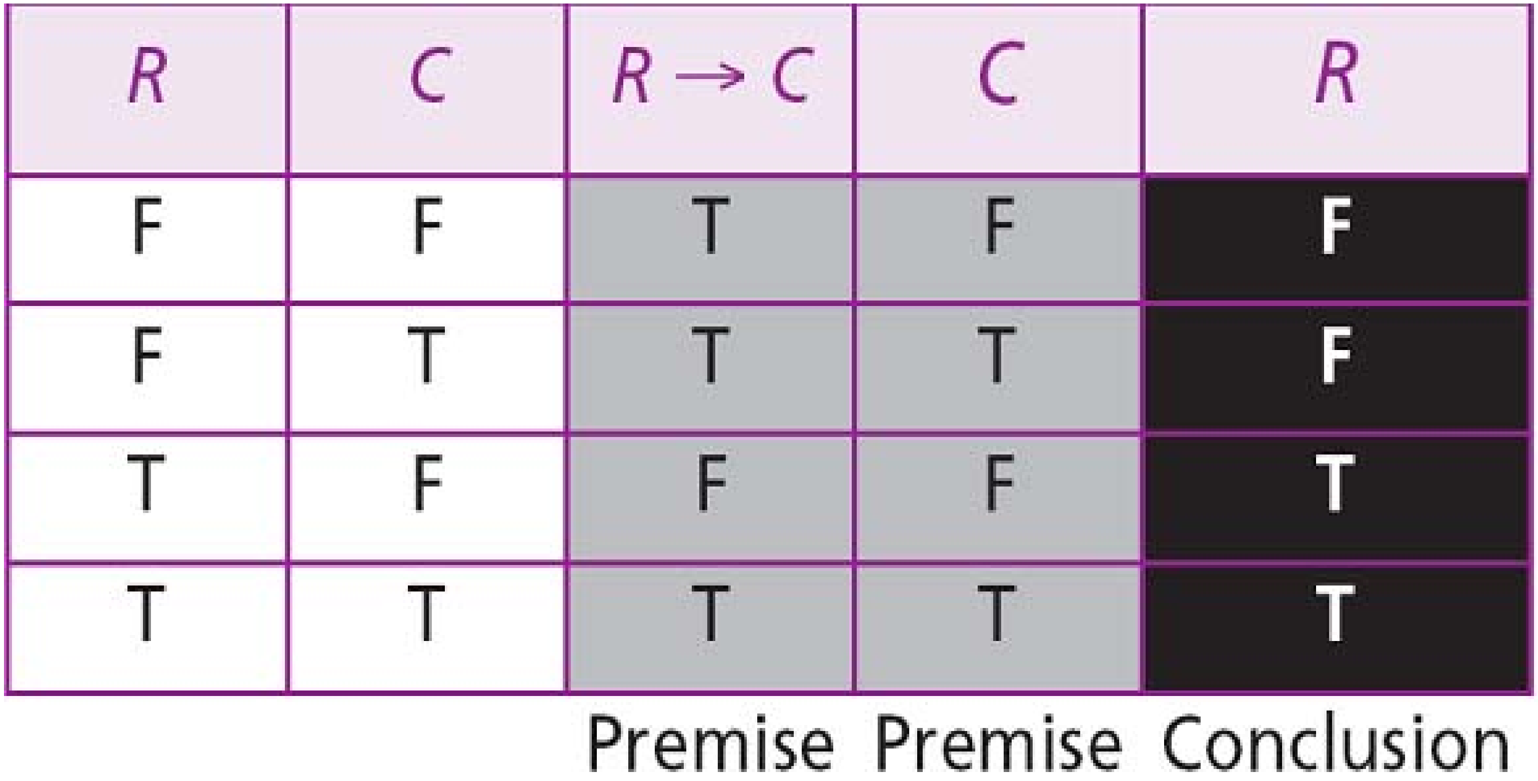
example



H	O	$H \vee O$	$\neg H$	O	
F	F	F	T	F	
F	T	T	T	T	OK
T	F	T	F	F	
T	T	T	F	T	
		Premise	Premise	Conclusion	

Another example

The argument $\{R \to C, C\} \mid\!- R$



R	C	$R \to C$	C	R
F	F	T	F	F
F	T	T	T	F
T	F	F	F	T
T	T	T	T	T
		Premise	Premise	Conclusion

Predicate logic

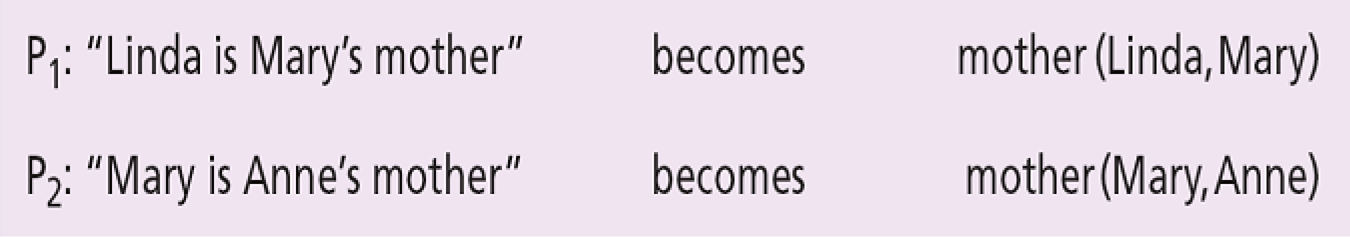
- In propositional logic, a symbol that represents a sentence is atomic: it cannot be broken up to find information about its components. For example, consider the sentences:

P_1: "Linda is Mary's mother" P_2: "Mary is Anne's mother"

- We can combine these two sentences in many ways to create other sentences, but we cannot extract any relation between Linda and Anne
  - For example, we cannot infer from the above two sentences that Linda is the grandmother of Anne.
- To do so, we need predicate logic: the logic that defines the relation between the parts in a proposition.

predicate logic

- In predicate logic, a sentence is divided into a predicate and arguments. For example, each of the following propositions can be written as predicates with two arguments:



P_1: "Linda is Mary's mother"	becomes	mother (Linda, Mary)
P_2: "Mary is Anne's mother"	becomes	mother (Mary, Anne)

- The relationship of motherhood in each of the above sentences is defined by the predicate mother. If the object Mary in both sentences refers to the same person, we can infer a new relation between Linda and Anne:

grandmother (Linda, Anne))

A sentence in predicate language

- A sentence in predicate language is defined as follows:
  - A predicate with n arguments is a sentence.
  - Any of the two constant values (true and false) is a sentence.
  - If P is a sentence, then ¬P is a sentence.
  - If P and Q are sentences, then $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$, andP $\leftrightarrow$ Q are sentences.

Quantifiers

- Predicate logic allows us to use quantifiers. Two quantifiers are common in predicate logic:

∀ and ∃

- The first, which is read as “for all”, is called the universal quantifier: it states that something is true for every object that its variable represents.
- The second, which is read as “there exists”, is called the existential quantifier: it states that something is true for one or more objects that its variable represents.

examples

1. The sentence “All men are mortals” can be written as:

$$\boxed{\forall x[\text{man }(x) \rightarrow \text{mortal }(x)]}$$

2. The sentence “Frogs are green” can be written as:

$$\boxed{\forall x[\text{frog }(x) \rightarrow \text{green }(x)]}$$

3. The sentence “Some flowers are red” can be written as:

$$\boxed{\exists\, x[\text{flower }(x) \wedge \text{red}(x)]}$$

More examples

4. The sentence “John has a book” can be written as:

$$\boxed{\exists\, x[\mathrm{book}\ (x) \wedge \mathrm{has}\ (\mathrm{John}, x)]}$$

5. The sentence “No frog is yellow” can be written as:

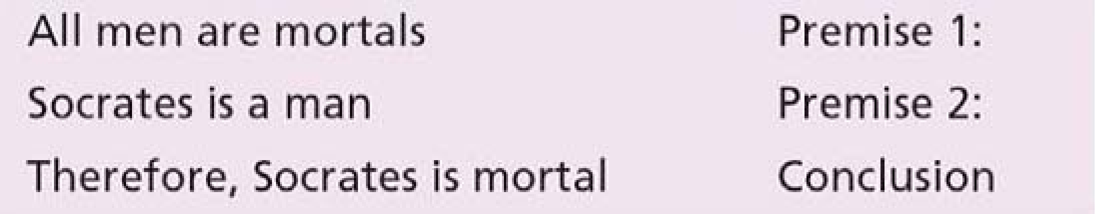
$$\boxed{\forall x[\mathrm{frog}\ (x) \rightarrow \neg \mathrm{yellow}\ (x)]}$$

or

$$\boxed{\neg\exists\, x[\mathrm{frog}(x) \wedge \mathrm{yellow}(x)]}$$

Deduction

- In predicate logic, if there is no quantifier, the verification of an argument is the same as that which we discussed in propositional logic.
- However, the verification becomes more complicated if there are quantifiers. For example, the following argument is completely valid.



All men are mortals	Premise 1:
Socrates is a man	Premise 2:
Therefore, Socrates is mortal	Conclusion

$$\forall x\,[\text{man}(x) \rightarrow \text{mortal}(x)]\,,\ \text{man}(\text{Socrates}) \vdash \text{mortal}(\text{Socrates})$$

Beyond predicate logic

- There have been further developments in logic to include the need for logical reasoning.
- Some examples of these include：
  - high-order logic
  - default logic
  - modal logic
  - temporal logic

Components of a Production Systems

1. Collection of states
   - Start (or initial) state
   - Goal state (or states)
2. Collection of productions: rules or moves
   - Each production may have preconditions
3. Control system: decides which production to apply next

Reasoning by Searching

- **State Graph:** All states and productions
- **Search Tree:** A record of state transitions explored while searching for a goal state
  - Breadth-first search
  - Depth-first search

Figure 11.4 A small portion of the eight-puzzle's state graph

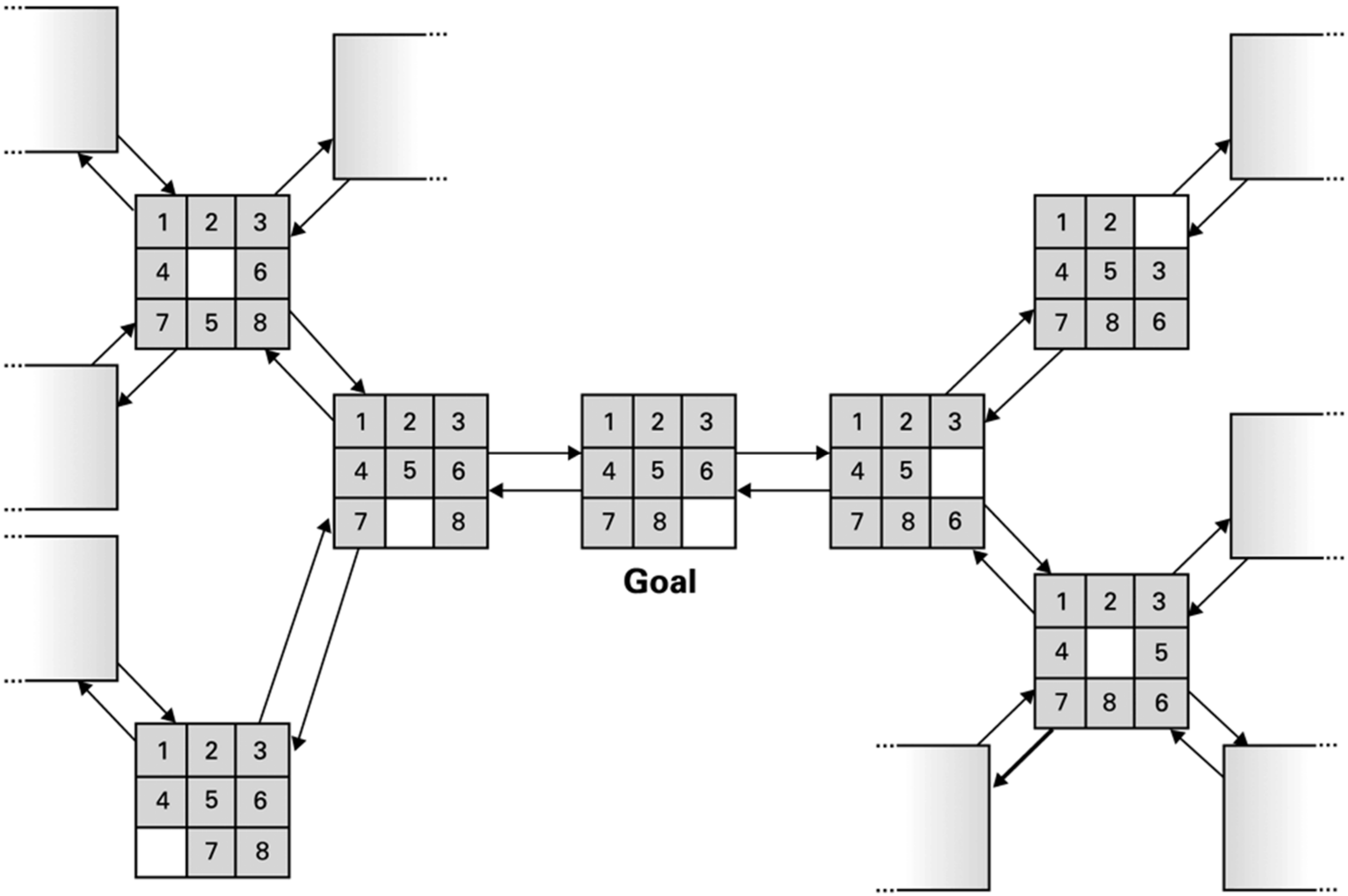


Figure 11.5 Deductive reasoning in the context of a production system

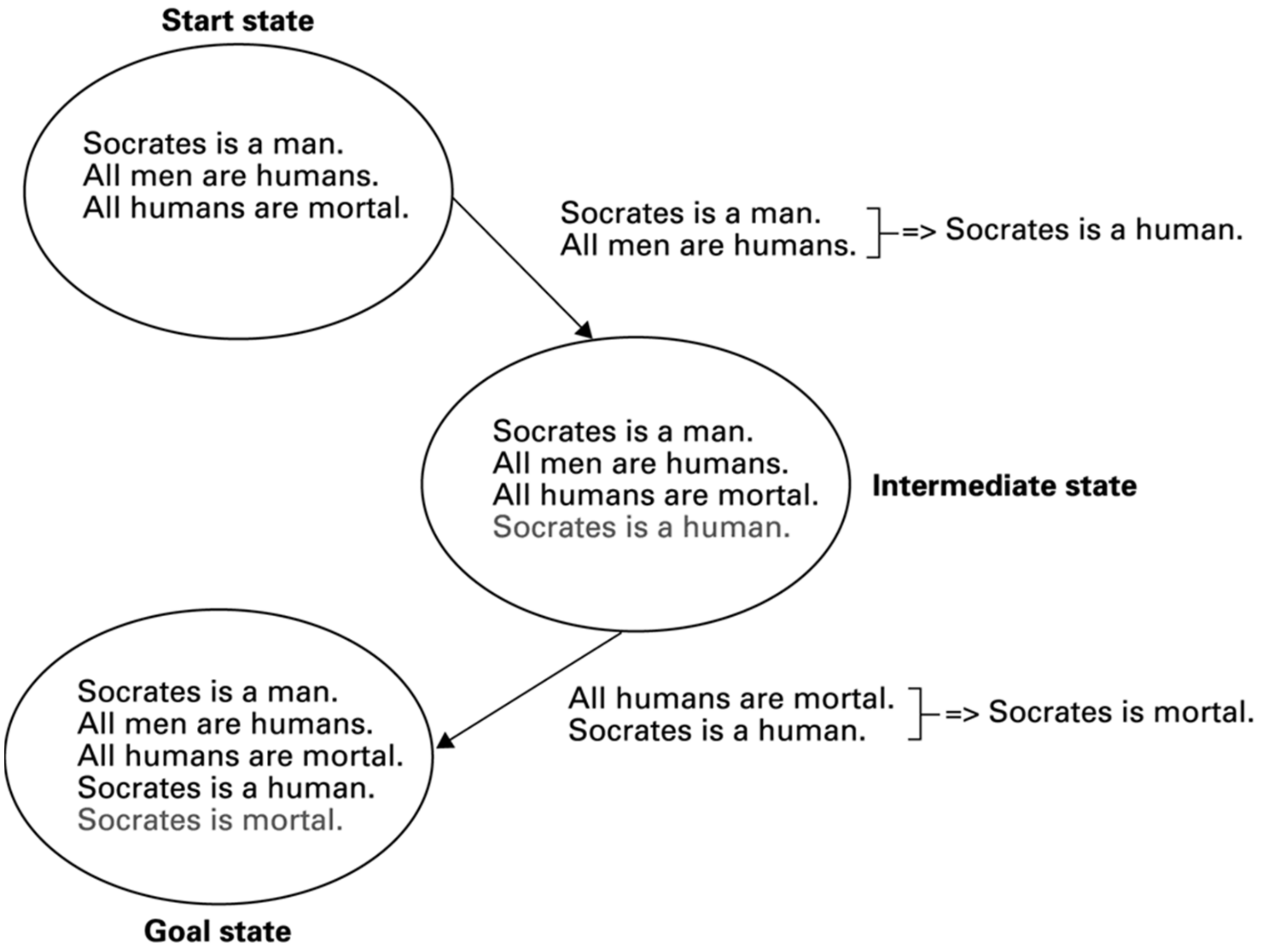
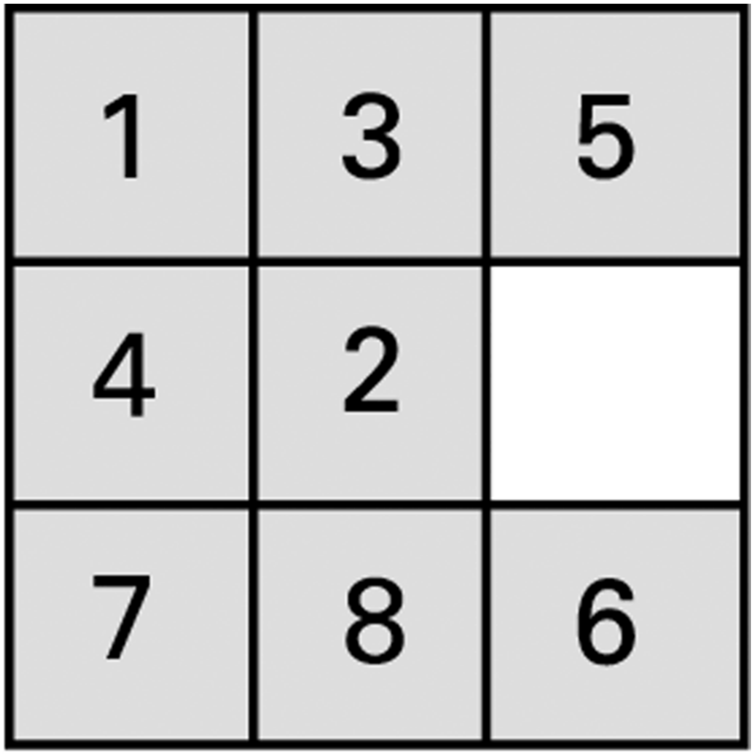


Figure 11.6 An unsolved eight-puzzle



1	3	5
4	2	
7	8	6

Figure 11.7 A sample search tree

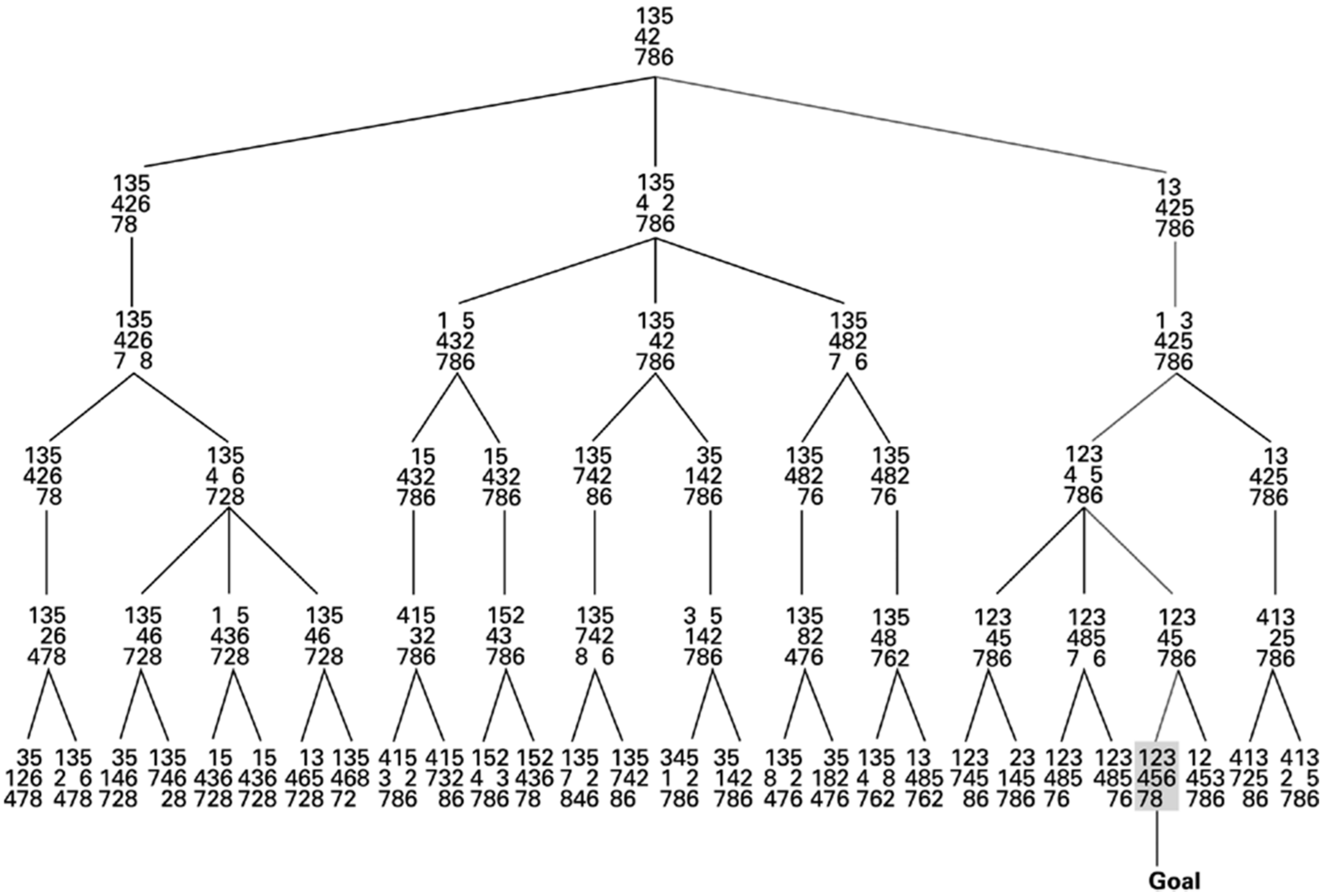
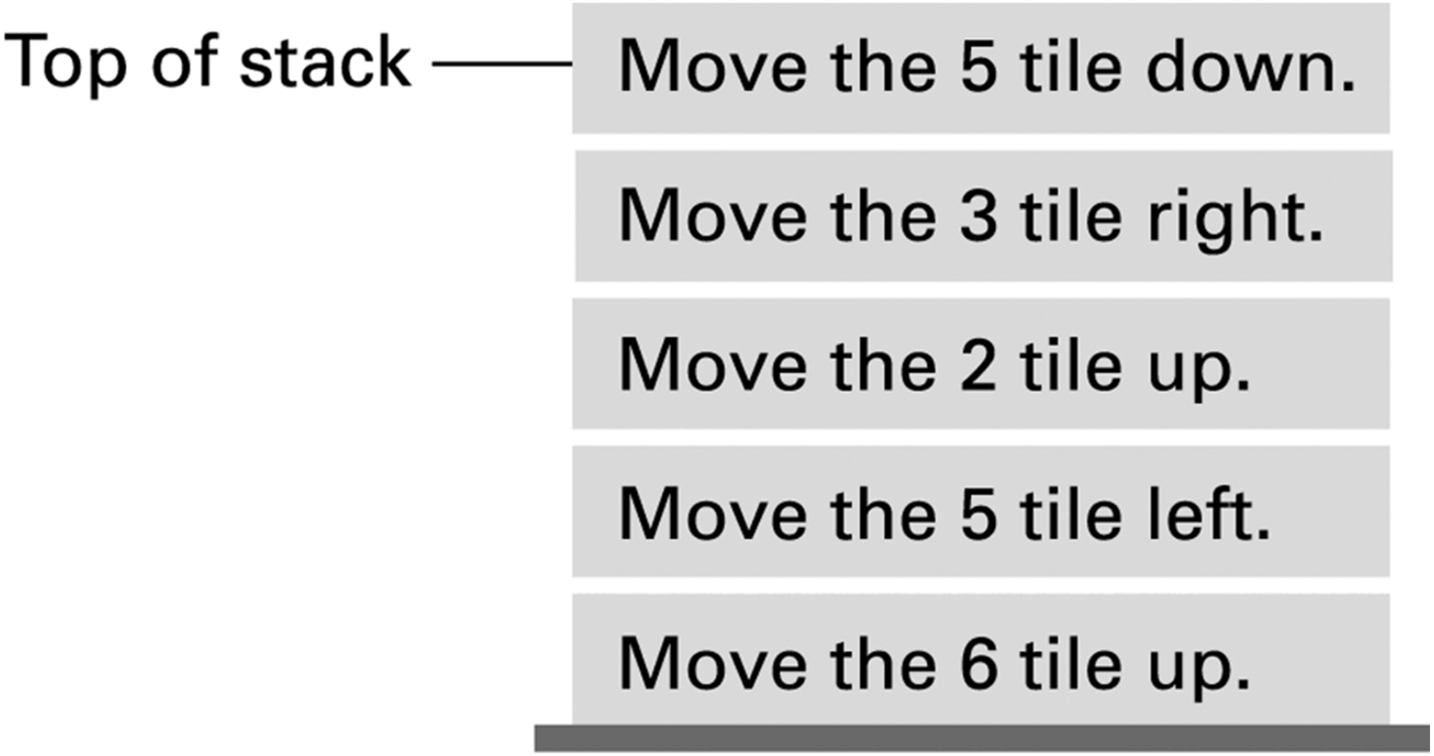


Figure 11.8 Productions stacked for later execution



Top of stack

Move the 5 tile down.

Move the 3 tile right.

Move the 2 tile up.

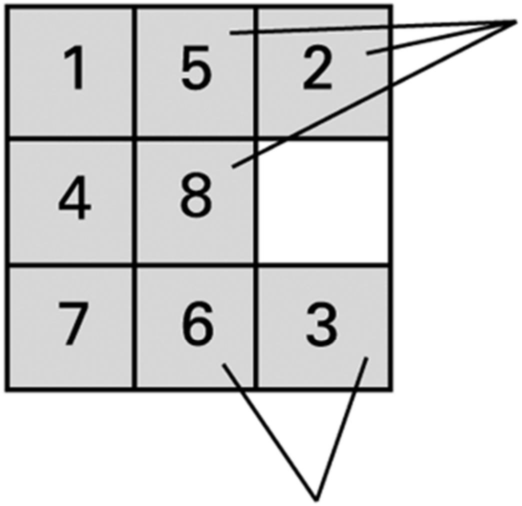
Move the 5 tile left.

Move the 6 tile up.

Heuristic Strategies

- **Heuristic:** A “rule of thumb” for making decisions
- Requirements for good heuristics
  - Must be easier to compute than a complete solution
  - Must provide a reasonable estimate of proximity to a goal

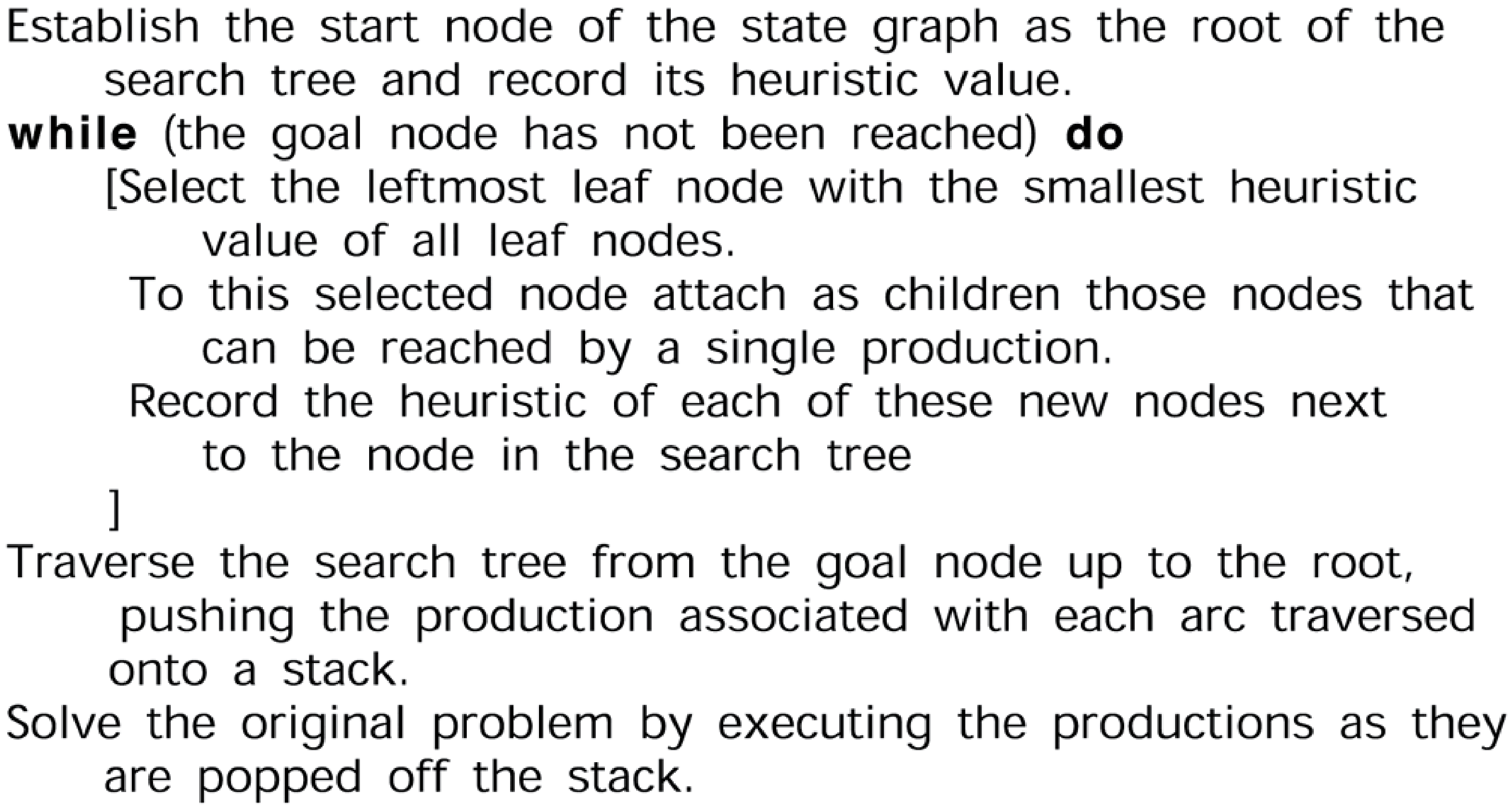
Figure 11.9 An unsolved eight-puzzle



These tiles are at least one move from their original positions.

These tiles are at least two moves from their original positions.

Figure 11.10 An algorithm for a control system using heuristics



```
Establish the start node of the state graph as the root of the
    search tree and record its heuristic value.
while (the goal node has not been reached) do
    [Select the leftmost leaf node with the smallest heuristic
        value of all leaf nodes.
     To this selected node attach as children those nodes that
        can be reached by a single production.
     Record the heuristic of each of these new nodes next
        to the node in the search tree
    ]
Traverse the search tree from the goal node up to the root,
     pushing the production associated with each arc traversed
    onto a stack.
Solve the original problem by executing the productions as they
    are popped off the stack.
```

Figure 11.11 The beginnings of our heuristic search

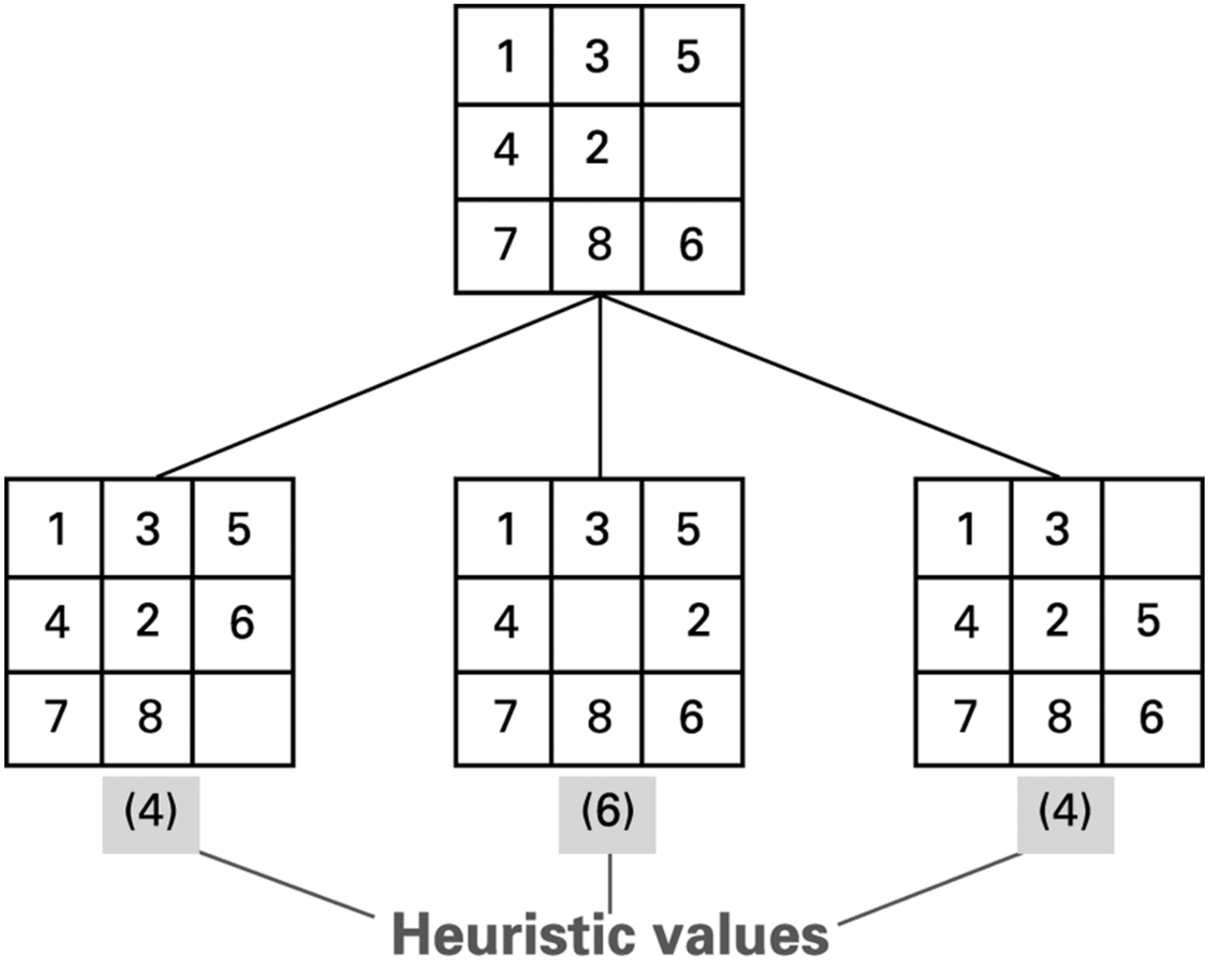


Figure 11.12 The search tree after two passes

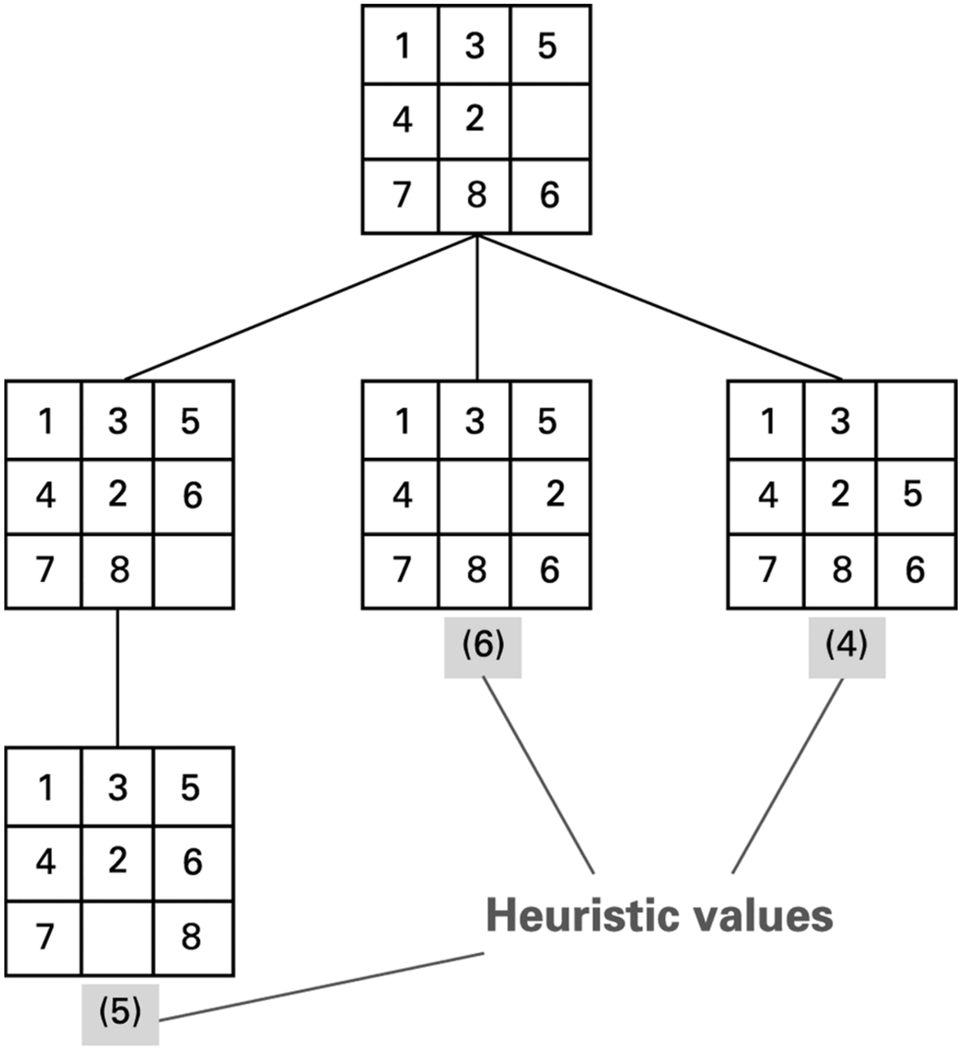


Figure 11.13 The search tree after three passes

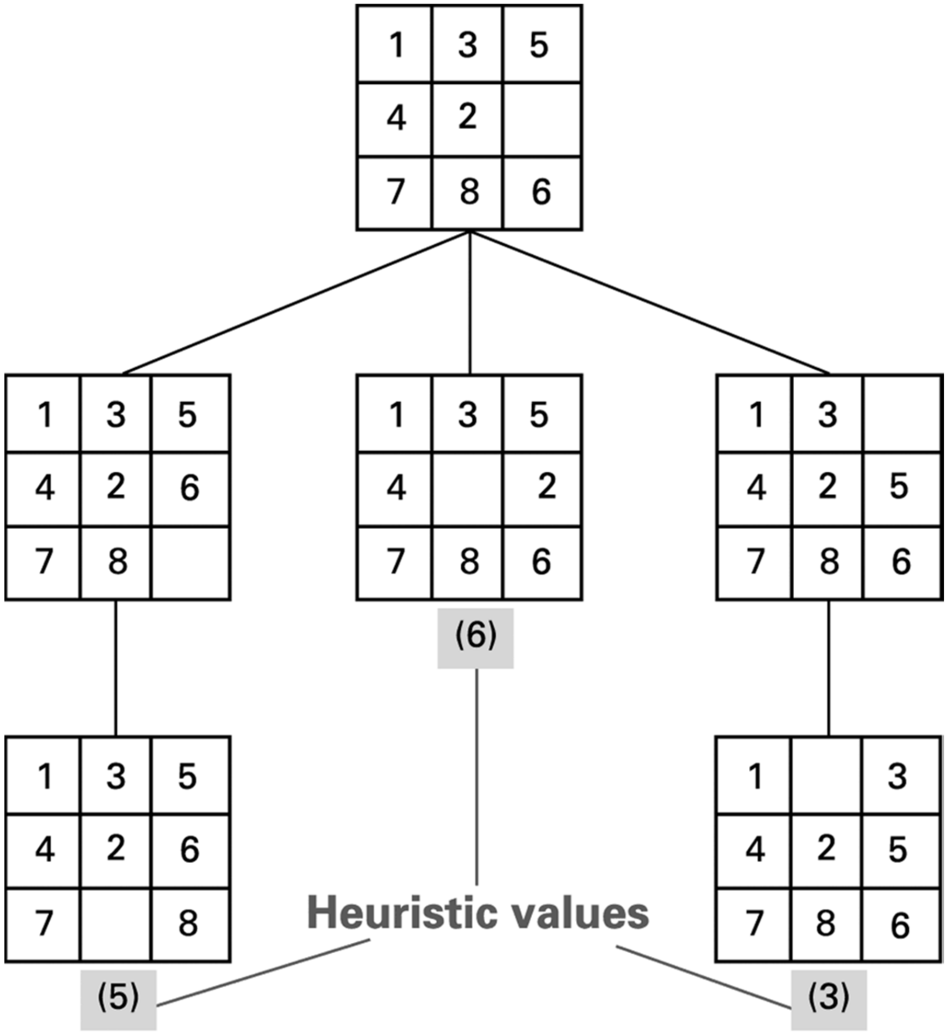
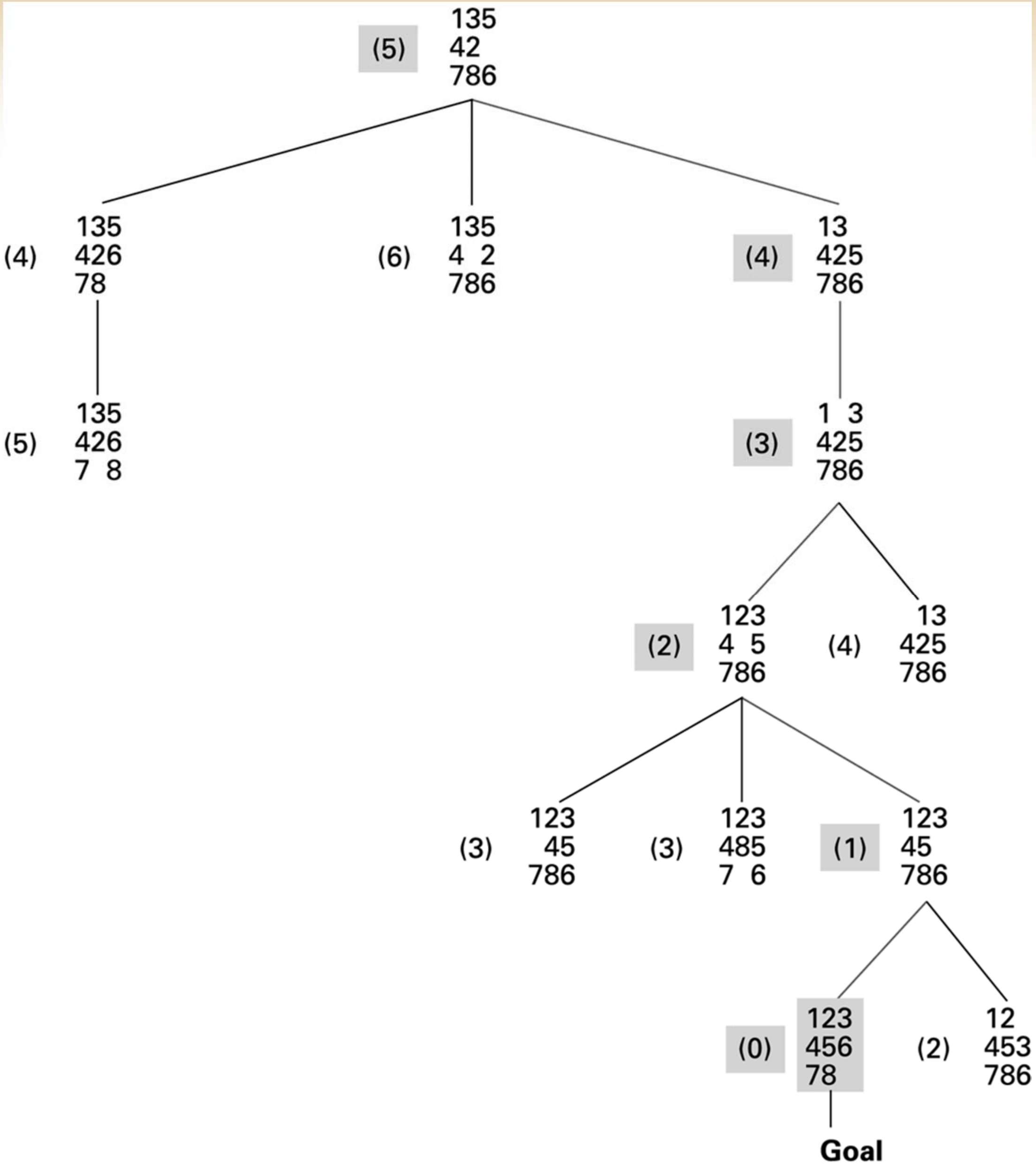


Figure 11.14
The complete search tree formed by our heuristic system



Handling Real-World Knowledge

- Representation and storage
- Accessing relevant information
  - Meta-Reasoning
  - Closed-World Assumption
- Frame problem

Learning

- Imitation
- Supervised Training
- Reinforcement
- Evolutionary Techniques

Artificial Neural Networks

- Artificial Neuron
  - Each input is multiplied by a weighting factor.
  - Output is 1 if sum of weighted inputs exceeds the threshold value; 0 otherwise.
- Network is programmed by adjusting weights using feedback from examples.

Figure 11.15 A neuron in a living biological system

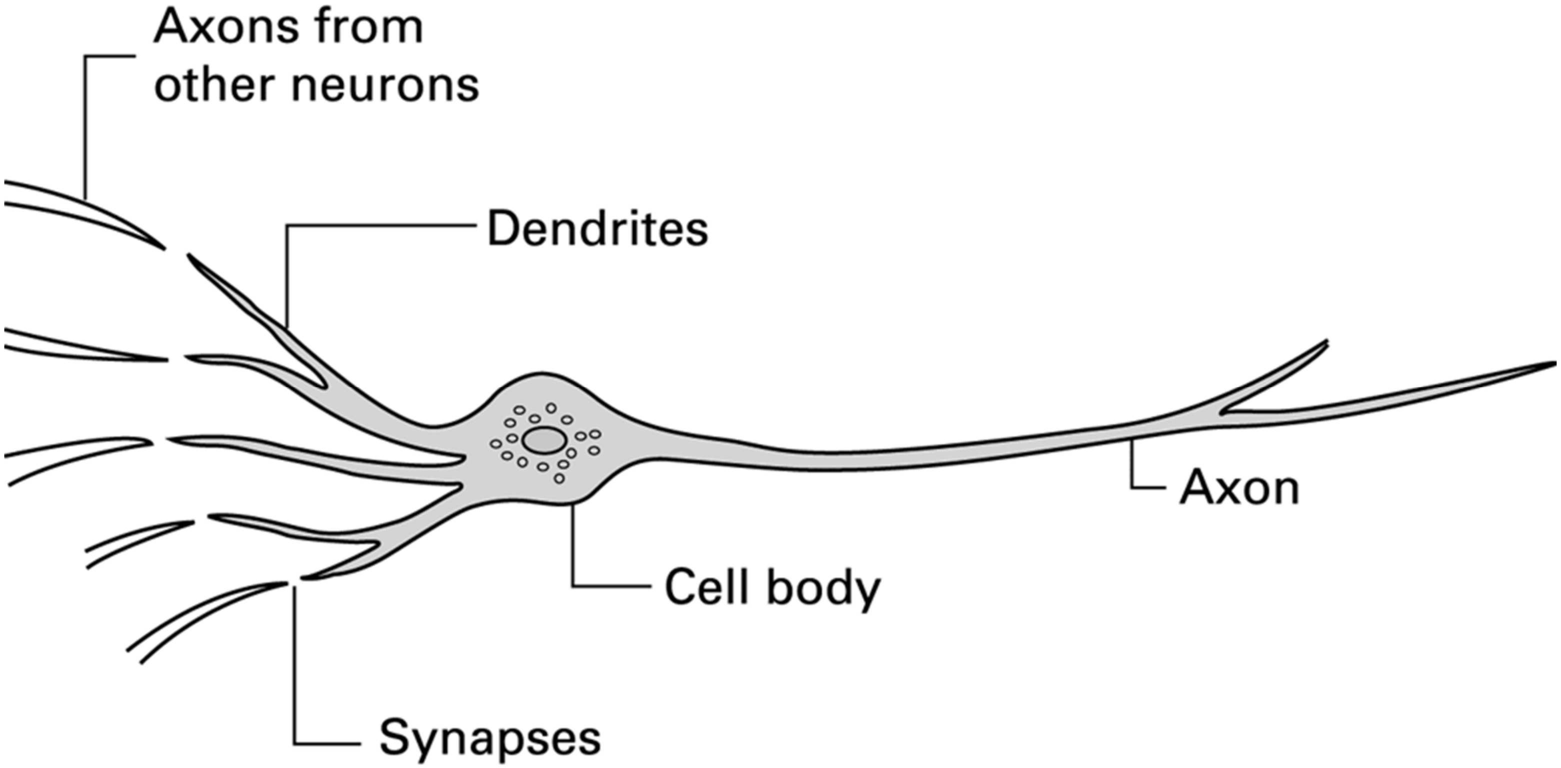


Figure 11.16 The activities within a processing unit

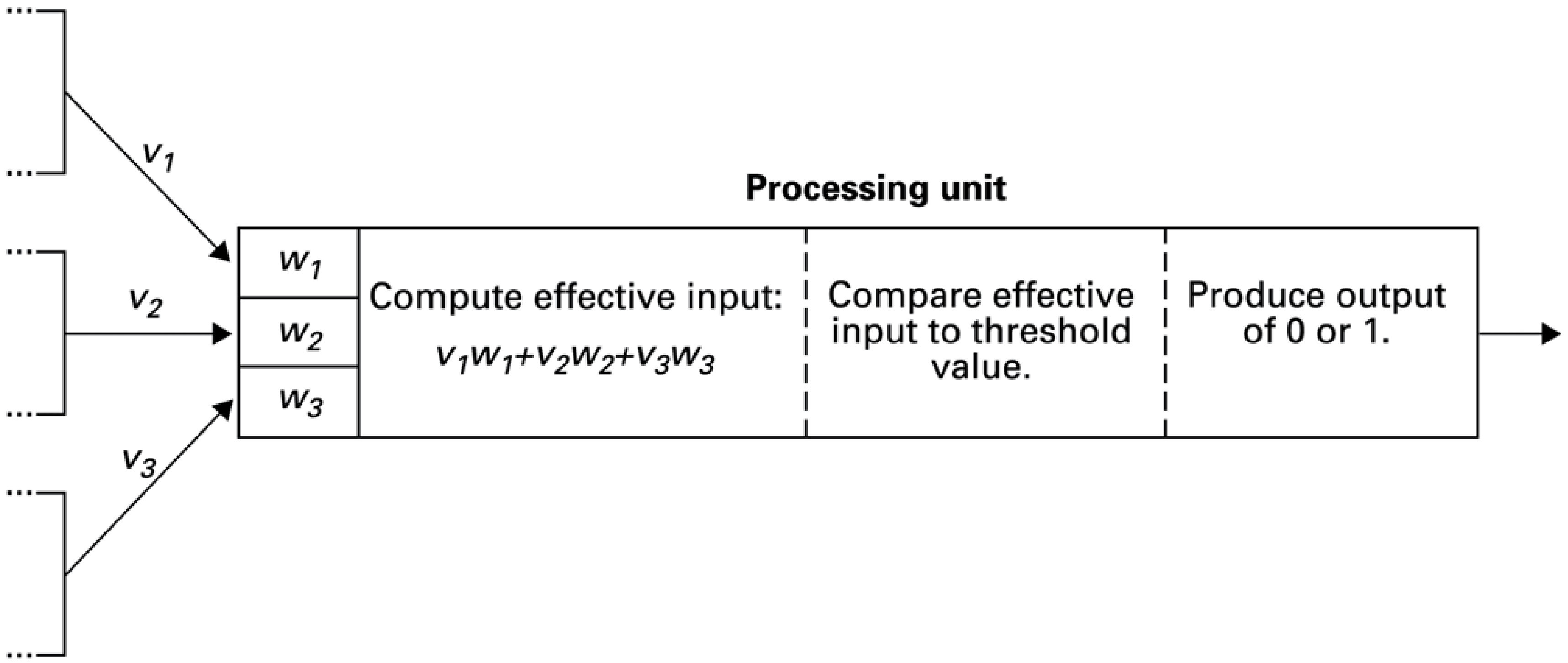


Figure 11.17 Representation of a processing unit

Figure 11.18 A neural network with two different programs

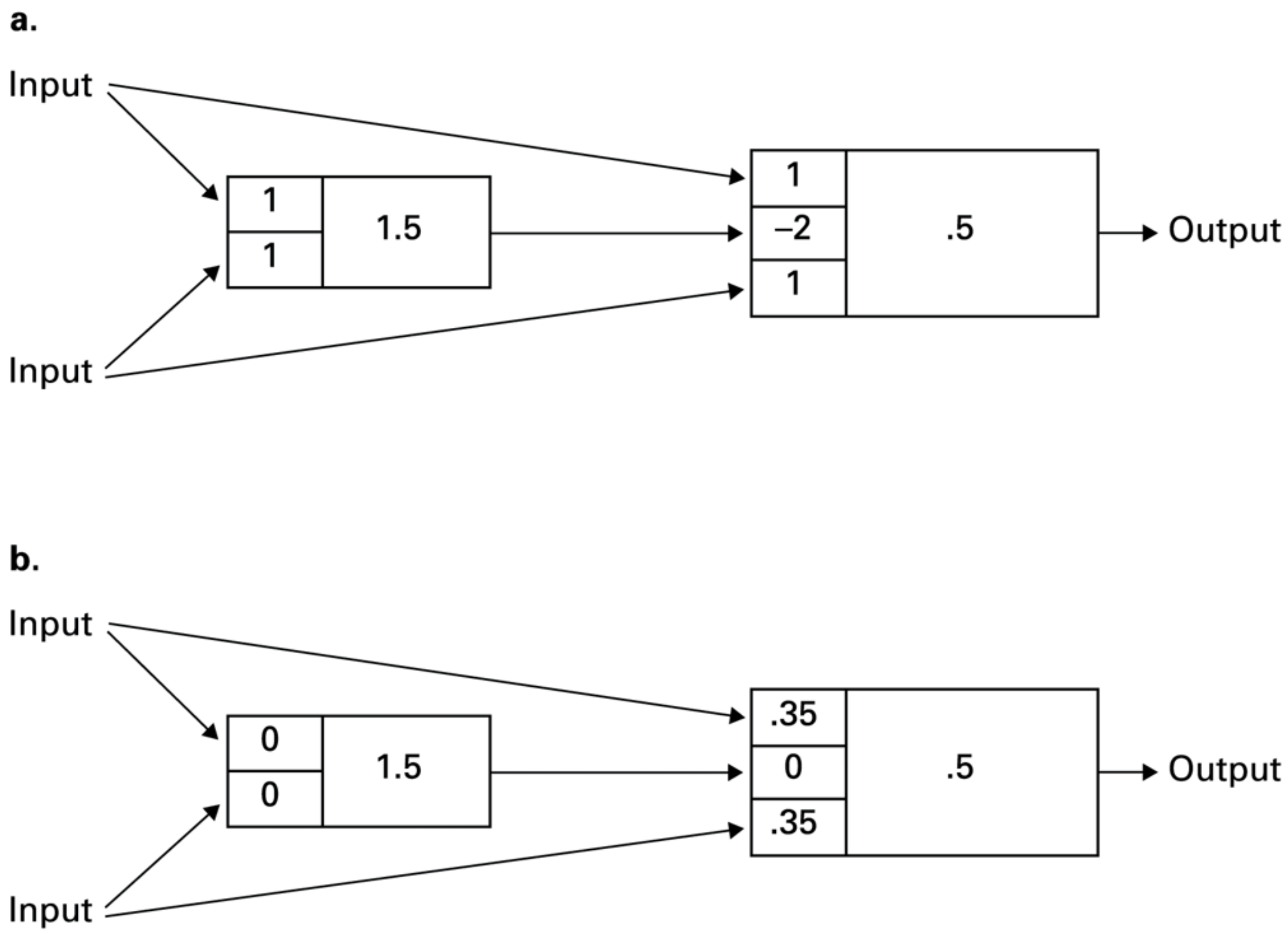


Figure 11.19 An artificial neural network

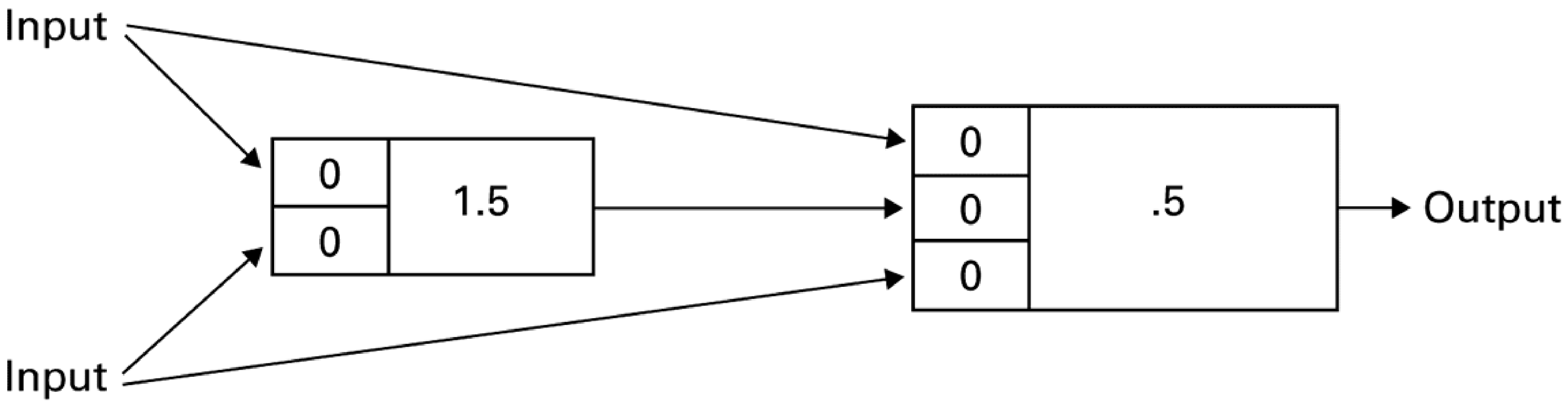
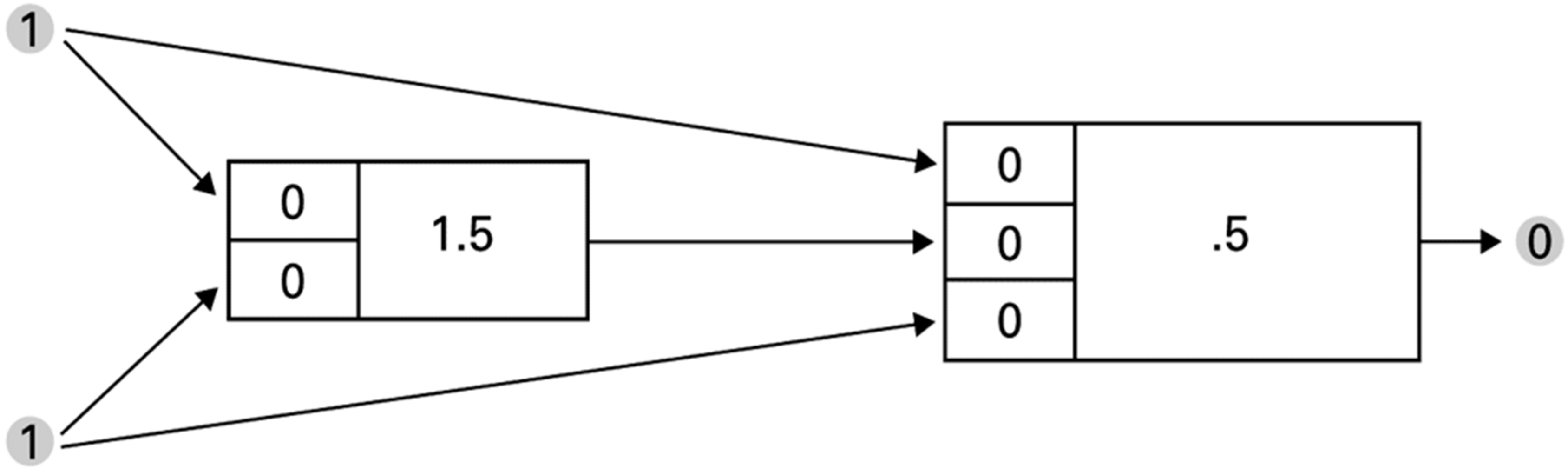
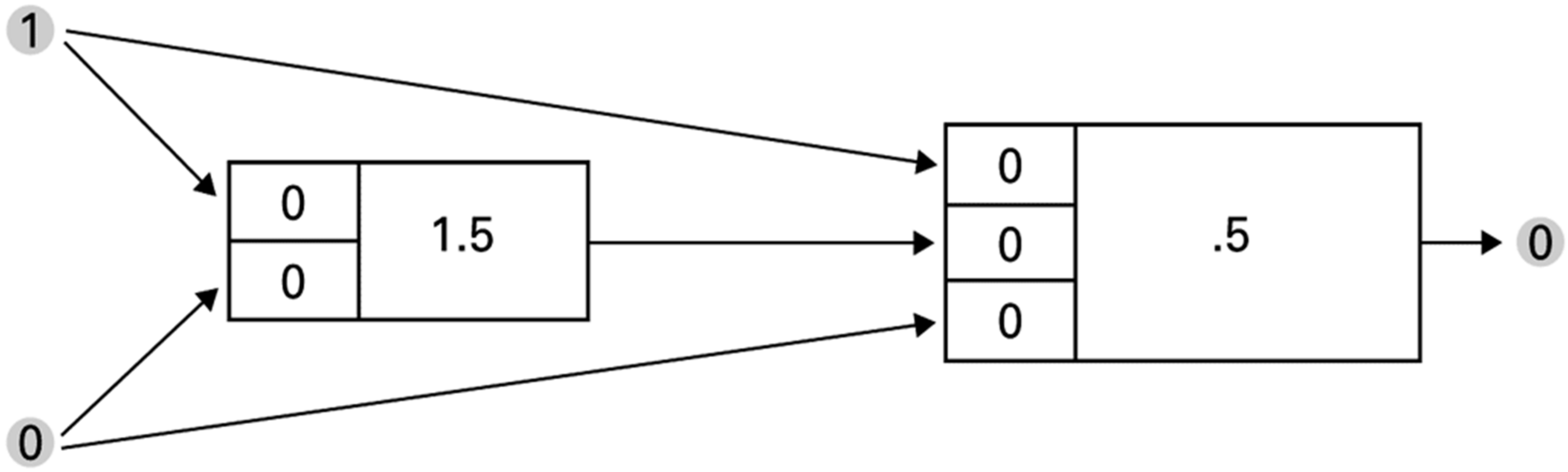


Figure 11.20 Training an artificial neural network



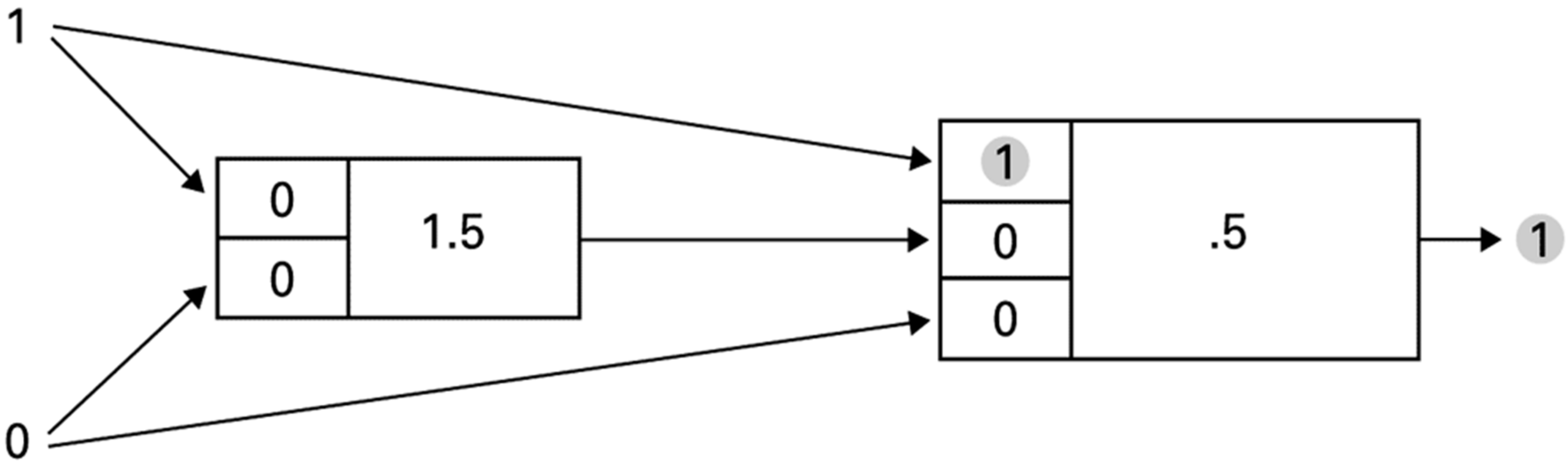
a. The network performs correctly for the input pattern 1, 1.

Figure 11.20 Training an artificial neural network (continued)



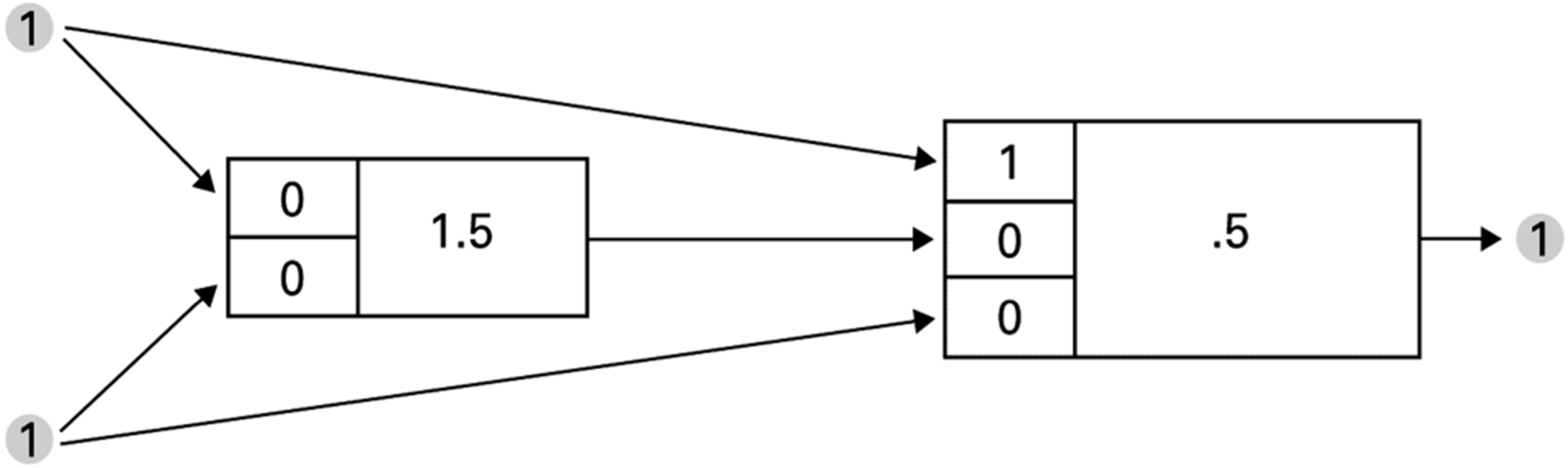
b. The network performs incorrectly for the input pattern 1, 0.

Figure 11.20 Training an artificial neural network (continued)



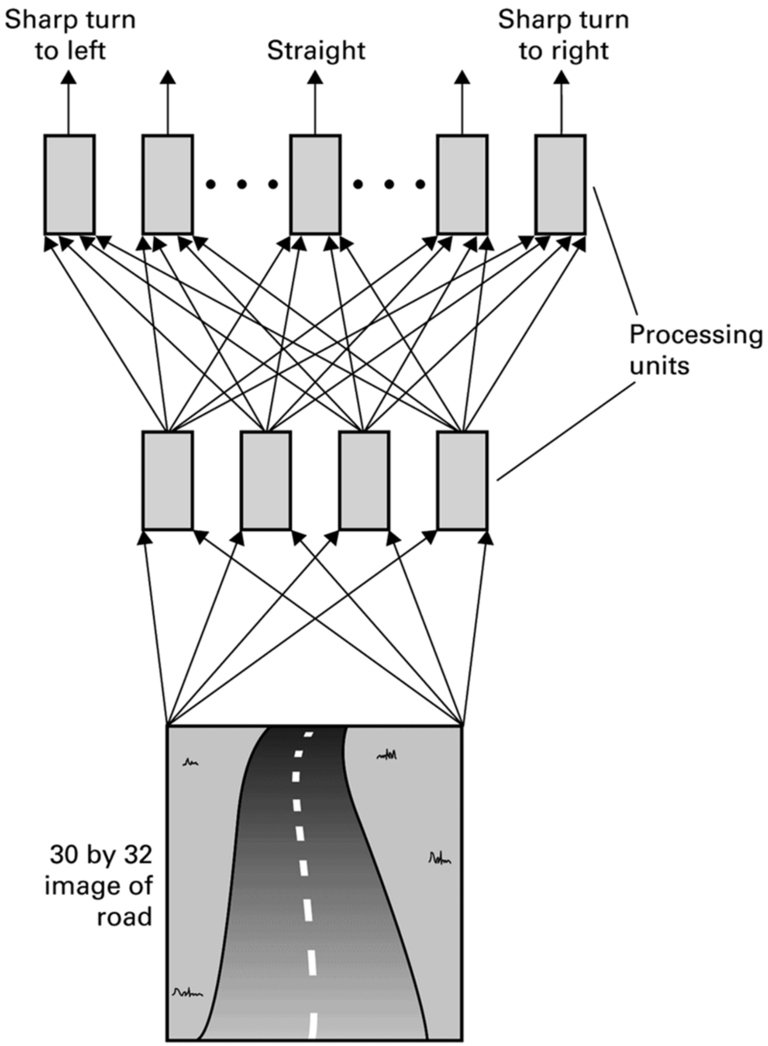
c. The upper weight in the second processing unit is adjusted.

Figure 11.20 Training an artificial neural network (continued)



d. However, the network no longer performs correctly for the input pattern 1, 1.

Figure 11.21 The structure of ALVINN



Associative Memory

- **Associative memory:** The retrieval of information relevant to the information at hand
- One direction of research seeks to build associative memory using neural networks that when given a partial pattern, transition themselves to a completed pattern.

Figure 11.22 An artificial neural network implementing an associative memory

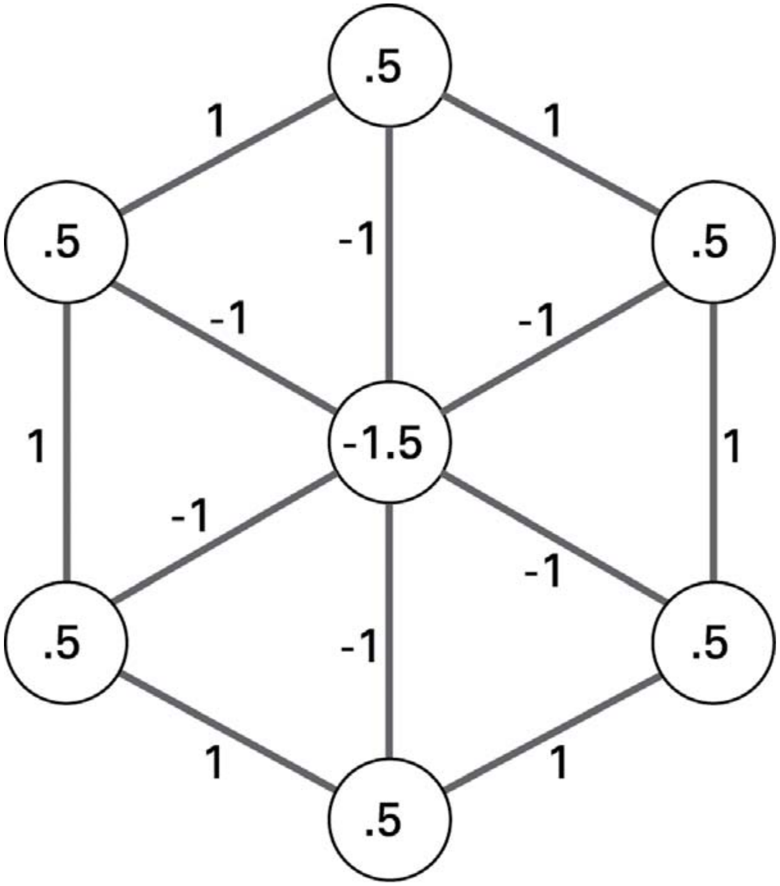
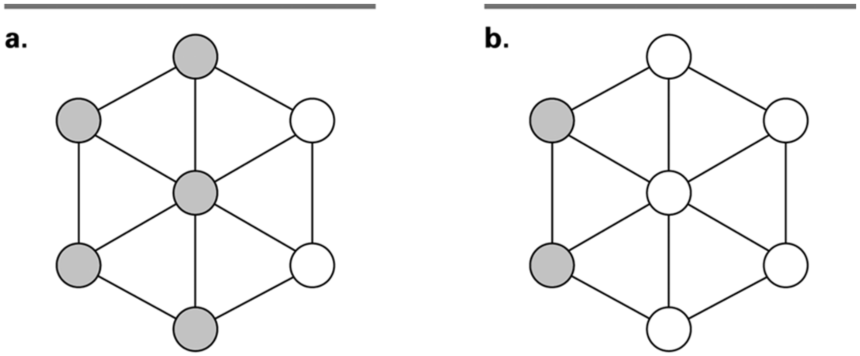
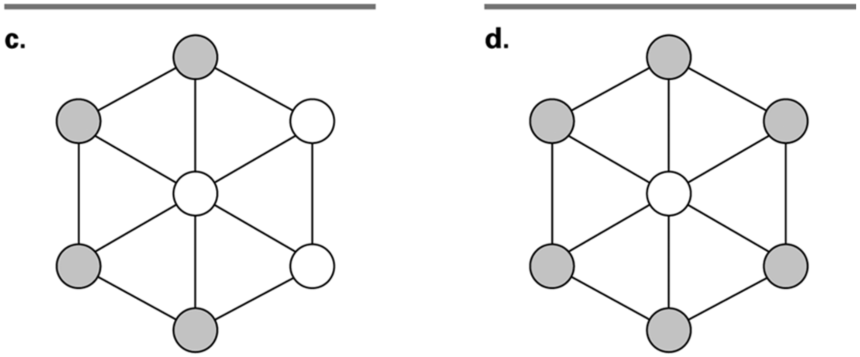


Figure 11.23 The steps leading to a stable configuration



Start: All but the rightmost units are excited

Step1: Only the leftmost units remain excited



Step 2: The top and bottom units become excited

Final: All the units on the perimeter are excited

Robotics

- Truly autonomous robots require progress in perception and reasoning.
- Major advances being made in mobility
- Plan development versus reactive responses
- Evolutionary robotics

Issues Raised by Artificial Intelligence

- When should a computer’s decision be trusted over a human’s?
- If a computer can do a job better than a human, when should a human do the job anyway?
- What would be the social impact if computer “intelligence” surpasses that of many humans?

Q&A