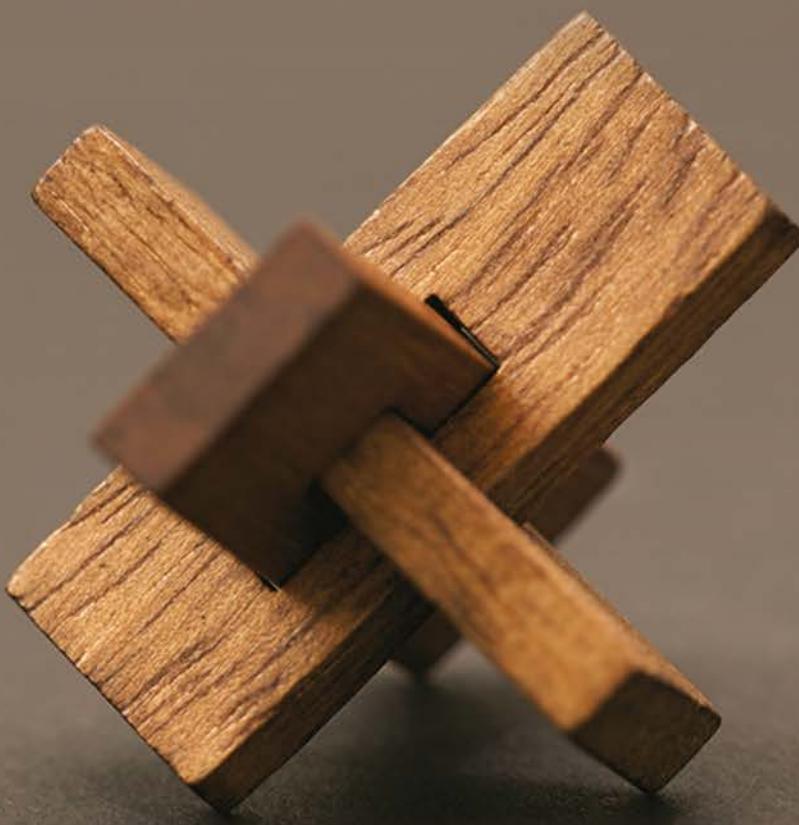


SECOND EDITION

PROBLEM SOLVING

Perspectives from Cognition
and Neuroscience



S. IAN ROBERTSON



PROBLEM SOLVING

The way that we assess and overcome problems is an essential part of everyday life. *Problem Solving* provides a clear introduction to the underlying mental processes involved in solving problems. Drawing on research from cognitive psychology and neuroscience, it examines the methods and techniques used by both novices and experts in familiar and unfamiliar situations.

This edition has been comprehensively updated throughout, and now features cutting-edge content on creative problem solving, insight and neuroscience. Each chapter is written in an accessible way, and contains a range of student-friendly features such as activities, chapter summaries and further reading. *Problem Solving* also provides clear examples of studies and approaches that help the reader fully understand important and complex concepts in greater detail.

Problem Solving fully engages the reader with the difficulties and methodologies associated with problem solving. This book will be of great use to undergraduate students of cognitive psychology, education and neuroscience as well as readers and professionals with an interest in problem solving.

S. Ian Robertson is former head of psychology at the University of Bedfordshire, United Kingdom.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

PROBLEM SOLVING

Perspectives from Cognition and
Neuroscience

Second Edition

S. Ian Robertson

Second edition published 2017
by Routledge
2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN

and by Routledge
711 Third Avenue, New York, NY 10017

Routledge is an imprint of the Taylor & Francis Group, an informa business

© 2017 S. Ian Robertson

The right of S. Ian Robertson to be identified as author of this work has been asserted by him in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

First edition published by Psychology Press 2001

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
Names: Robertson, S. Ian, 1951– author.

Title: Problem solving : perspectives from cognition and neuroscience /
S. Ian Robertson.

Description: 2nd edition. | Abingdon, Oxon ; New York, NY : Routledge, 2017.

Identifiers: LCCN 2016025185 | ISBN 9781138889569 (hardback : alk. paper) |

ISBN 9781138889576 (pbk. : alk. paper)

Subjects: LCSH: Problem solving.

Classification: LCC BF449 .R63 2017 | DDC 153.4/3—dc23

LC record available at <https://lccn.loc.gov/2016025185>

ISBN: 978-1-138-88956-9 (hbk)

ISBN: 978-1-138-88957-6 (pbk)

ISBN: 978-1-315-71279-6 (ebk)

Typeset in Bembo
by Apex CoVantage, LLC

For Kathryn, Poppy and Aidan
All life is problem solving – Popper, 1999



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

CONTENTS

<i>Preface</i>	x
1 What is involved in problem solving	1
<i>What exactly is a problem?</i> 1	
<i>Where do problems come from?</i> 5	
“Natural” and “unnatural” problems 6	
<i>What’s involved in solving problems?</i> 7	
<i>Approaches to the study of problem solving</i> 8	
<i>Categorising problems</i> 17	
<i>Summary</i> 24	
<i>References</i> 25	
2 Problem representation	27
<i>Representations and processes</i> 29	
<i>Analysing well-defined problems</i> 33	
<i>The interaction of the problem solver and the task environment</i> 40	
<i>Heuristic search strategies</i> 42	
<i>Summary</i> 52	
<i>References</i> 53	
3 Transfer	55
<i>Negative transfer – mental set</i> 58	
<i>Mechanisms of knowledge transfer</i> 60	
<i>Transfer in well-defined problems</i> 61	
<i>Studies of analogical problem solving</i> 66	

viii Contents

<i>Cognitive processes in analogical problem solving</i>	68
<i>Types of similarity</i>	70
<i>When relational structures are ignored</i>	78
<i>Summary</i>	87
<i>References</i>	88
4 Worked examples and instructional design	92
<i>Difficulties facing textbook writers</i>	93
<i>The role of examples in textbooks</i>	96
<i>The processes involved in textbook problem solving</i>	103
<i>Understanding problems revisited</i>	108
<i>Approaches to the design of instruction</i>	109
<i>Conclusion</i>	118
<i>Summary</i>	119
<i>References</i>	120
5 Developing skill	126
<i>Induction</i>	127
<i>Schema development and the effects of automatisation</i>	135
<i>Cognitive architectures</i>	136
<i>Potential criticisms of cognitive models</i>	146
<i>Summary</i>	148
<i>References</i>	149
6 Developing expertise	152
<i>Stage models of expertise</i>	152
<i>The intermediate effect</i>	156
<i>What distinguishes experts and novices</i>	158
<i>Are experts smarter? Are there differences in abilities?</i>	159
<i>Is expertise due to talent or deliberate practice?</i>	161
<i>Does expertise cross domains?</i>	162
<i>Cognitive processes in expertise</i>	163
<i>Flexibility in thinking</i>	166
<i>Some potential side effects of expertise . . .</i>	169
<i>Summary</i>	169
<i>References</i>	171
7 Insight	176
<i>Insight problems</i>	176
<i>Gestalt accounts of problem solving</i>	177
<i>Information processing approaches to insight</i>	184

<i>Classifying insight problems</i>	186
<i>Insight as something special</i>	190
<i>Insight as “business as usual”</i>	191
<i>Representational change theory (redistribution theory)</i>	195
<i>Summary</i>	201
<i>References</i>	205
8 Creative problem solving	208
<i>Breaking free of self-imposed constraints</i>	209
<i>Creative individuals</i>	211
<i>Theories of creativity: generation, evaluation and selection</i>	215
<i>CoRT</i>	217
<i>Synectics</i>	218
<i>Creative Problem Solving (CPS) – Osborn–Parnes</i>	219
<i>Summary</i>	220
<i>References</i>	221
9 The neuroscience of problem solving	225
<i>Methods used in studying brain functions</i>	225
<i>Arithmetic in the brain</i>	228
<i>Stages in problem solving</i>	232
<i>Neurological processes in analogical reasoning</i>	234
<i>Neurocomputational models</i>	239
<i>Designing instruction – what can studies of the brain tell us?</i>	243
<i>Neurological aspects of insight and creativity</i>	245
<i>Summary</i>	247
<i>References</i>	248
10 Conclusion	253
<i>Problems, problems</i>	253
<i>References</i>	260
<i>Index</i>	263

2

PROBLEM REPRESENTATION

Imagine for a moment that you are a parent and that you normally drop your daughter off at school on your way to work. One day you both pile into the car and discover that the car won't start. What do you do?

Let's suppose, for the sake of argument, that it is important that you get to work and that your daughter gets to school as soon as possible. However, you have to get your car fixed. You decide to call a taxi and call a garage from work and ask them to go out and look at the car. You then realise the garage will need a key for the car. While you are waiting for the taxi you call the garage and explain the situation. You say you will drop the keys off at the garage on your way to work. Okay so far. The next problem is how to get home and pick your daughter up from school in the evening. You decide that the simplest thing would be to take a bus from work that stops close by the garage and pick up the car (assuming there's nothing seriously wrong with it). You can then go and pick up your daughter in the car. She may have to wait in school for a while but, with a bit of luck, she shouldn't have to wait longer than about quarter of an hour.

Ten minutes later, the taxi arrives.

The point of this little story is that all you have done to solve the problem is to stand beside the telephone and *think*. It is exceedingly useful to be able to imagine – to think about – the results of an action or series of actions before you actually perform them. Thinking, in this sense, involves reasoning about a situation, and to do that we must have some kind of dynamic “model” of the situation in our heads. Any changes we make to this mental model of the world should ideally mirror potential changes in the real world (see Figure 5.3 in Chapter 5).

What kind of representation do we form of a problem? When psychologists talk of a mental representation they are usually referring to the way that information is encoded. The word “rabbit” can be represented visually as a visual code, and by what it sounds like, as a phonological code. We also know what “rabbit” means so there must be a semantic code. There are different representations for different processes that perform different functions and most of them are unconscious. While these and other forms of representation are necessary for our ability to acquire useful information about the world, they are not what we normally think of when we talk of a problem representation. In the preceding scenario the person doing the

thinking is running a mental model of the world to see what actions could be taken and their probable outcomes. This is, if you like, a higher-level representation than those involved in basic encoding processes, and one that we are consciously aware of.

The preceding scenario is the kind of problem that the environment throws at us unexpectedly. Problems can also arise from what we are told or what we read. When we read a piece of text, for example, we not only encode the information that is explicitly stated but we also have to make inferences as we read to make sense of the text. Most of these inferences are so automatic that we are often unaware that we made any inferences at all. Bransford, Barclay and Franks (1972) presented people with the following sentence:

Three turtles rested on a floating log and a fish swam beneath them.

They later gave a recognition test to some of the subjects that included the sentence:

Three turtles rested on a floating log and a fish swam beneath it.

Bransford et al. had hypothesised that participants would draw the inference that the fish swam beneath the log (notice that this is not stated in the original sentence). Indeed, the participants who were presented with the second sentence on a recognition task were as confident that the second sentence was the one that had been presented originally as those subjects who had been given the original sentence on the recognition task. The point here is that one's memory of a situation, based on a reading of a text, may include the inferences that were drawn at the time the representation of the text was constructed or retrieved. Furthermore, "we can think of the information that is captured by a particular representation as its meaning" (Markman, 1997, p. 38). So if three turtles were resting on a floating log and a fish swam beneath them, then this means that the fish swam beneath the log as well.

Problem solving, then, involves building a mental representation that performs some useful function (Markman, 2006). Now it follows that if you don't know much about the domain or you have never attempted this kind of problem before, then your understanding of the problem is unlikely to be all that good. Glaser (1984) explains why:

At the initial stage of problem analysis, the problem solver attempts to "understand" the problem by construing an initial problem representation. The quality, completeness, and coherence of this internal representation determine the efficiency and accuracy of further thinking. And these characteristics of the problem representation are determined by the knowledge available to the problem solver and the way the knowledge is organised.

(p. 93)

This does not mean that the representation has to be "complete" before any problem solving can take place. If you had a complete representation of a problem then you wouldn't have much of a problem since you would know exactly how to get from where you are now to where you want to be. Typical complex problems, such as the one involved in the preceding scenario or producing the plotline for *Game of Thrones*, tend to involve complex plans and as problem solving progresses the nature of the task is likely to change and the plans may need to be modified. As Simon (1973) pointed out, the same is true of a chess game. This is because

the task environment changes in the course of moving through the problem, and the environment constrains the behaviour of the solver and often dictates the course that problem solving has to take (the car needs a new part that won't be delivered till the next day; your opponent moves his knight to a position you hadn't expected). An adequate representation, on the other hand, should at least allow you to see what moves you can possibly make and allow you to start off heading towards your goal.

However, there is also the case of those problems where our initial representation gets us nowhere. The difficulty here lies in finding a new way of representing the problem (this is sometimes referred to as "lateral thinking" although the concept of re-representing problems in order to find a solution goes back long before the term was invented). Unfortunately, knowing that you should find a new way of representing the problem does not always help you very much – you still have the problem of finding this "new way of representing" it. Nevertheless, when a new representation comes to mind a solution is often immediately obvious; or, at least, you often know what to do so that a solution can be reached very quickly. This is generally known as insight. However, insight is not confined to so-called insight problems. The same phenomenon may occur when solving typical textbook algebra problems, for example, or in solving simple everyday problems. How do you get your floor tiles to look good when the walls don't seem to meet at right angles? Here again the initial representation may be extremely unhelpful, and only when a new representation is found can the poor student or do-it-yourselfer apply the solution procedure that is made obvious by the new representation. The study of insight problems can therefore tell us something about everyday problems and textbook problems and is the subject of Chapter 7.

Representations and processes

The information processing approach to thinking and problem solving owes a very great deal to the work of Alan Newell and Herb Simon and is described in detail in their book *Human Problem Solving* (Newell & Simon, 1972). Indeed, their model of human and computer problem solving could be termed the *modal model* of problem solving given that it is used to explain a wide variety of studies of thinking (see e.g., Ericsson & Hastie, 1998). This would be the problem solving equivalent of the modal model of memory (Atkinson & Shiffrin, 1968). They proposed two major problem solving processes: *understanding* and *search*. Understanding involves generating a representation of what the task involves, and search refers to finding a strategy that will lead to the goal. To understand Newell and Simon's model we shall take a simple example of thinking. Take a few moments to do Activity 2.1.

ACTIVITY 2.1

Look at the fairly trivial Tower of Hanoi problem in Figure 2.1. Using only your imagination, how would you get the two rings from peg A to peg C in three moves bearing in mind that

- You can move only one ring at a time from one peg to another.
- You cannot put the large one on the small one.

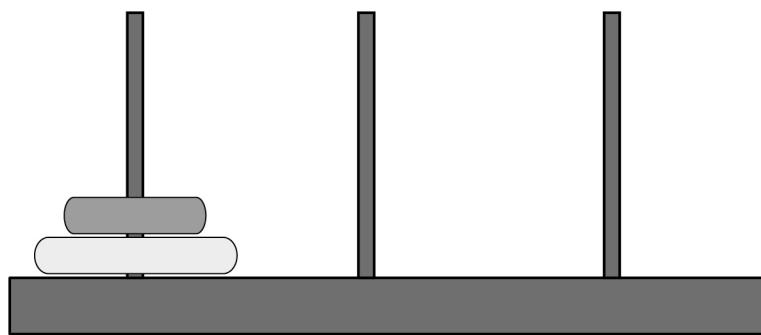


FIGURE 2.1 A trivial version of the Tower of Hanoi problem

As you solve the problem, try to be aware of how you go about doing it; that is, try to imagine the individual steps you would go through.

Your mental solution to the problem may have been something like that represented in Figure 2.2. Using the diagram in Activity 2.1 you probably formed a mental image similar to that in “thought 1” in Figure 2.2. Activity 2.1 describes the state of the problem at the start (the initial state) and “thought 1” is simply a mental representation that is analogous to this initial state of the problem. The state of the problem at the goal state can be represented as in “thought 4”. Notice here that there is no diagram given in the Activity to correspond to “thought 4” to show you what it should look like. Instead you had to construct the representation from the verbal description of the goal. In between there are intermediate states, “thought 2” and “thought 3”, which you reach after moving one ring each time. You created a *model* of the problem – a specific concrete situation in the external world – in your head and solved the problem within that model because the statement of the problem provided enough information to allow you to *reason* about the external situation. The process of *understanding*, then, refers to constructing an initial model (a mental representation) of what the problem is, based on the information in the problem statement about the goal, the initial state, what you are not allowed to do and what operator to apply, as well as your own personal past experience. Past experience, for example, would tell you that moving the small ring from peg A to peg C, then moving it to peg B, is a complete waste of time. Since your knowledge of the problem at each state was inside your head, each “thought” corresponds to a *knowledge state*.

A second aspect of your thought processes you may have been aware of is that they were *sequential*, which simply means that you had one thought after another as in Figure 2.2. One consequence of the fact that much of our thinking appears conscious and sequential in nature is that we can often easily verbalise what we are thinking about. You may have heard a voice in your head saying something like “okay, the small ring goes there . . . no, there. And then the large ring goes there” and so on. In fact, for many people saying the problem out loud helps them solve it possibly because the memory of hearing what they have just said helps reduce the load on working memory. The fact that a lot of problem solving is verbalisable in this way provides psychologists with a means of finding out how people solve such problems (Ericsson & Simon, 1993; see Information Box 1.2 in Chapter 1).

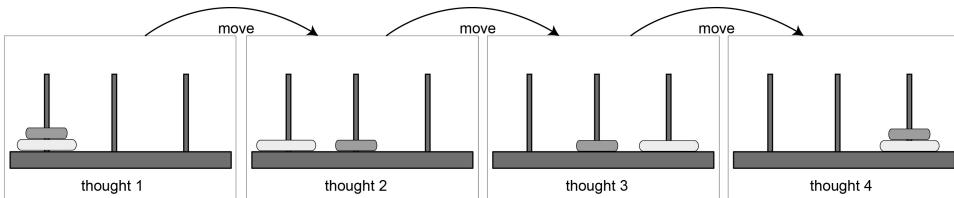


FIGURE 2.2 The sequence of imagined moves in solving the two-ring Tower of Hanoi problem

Next, notice that you moved from one state to another as if you were following a path through the problem. If you glance ahead at Figures 2.4 and 2.7 you will see that harder versions of the problem involve a number of choices. Indeed, in the simple version you had a choice to start with of moving the small ring to peg B or peg C. As such problems get harder it becomes less obvious which choices you should make to solve the problem in the smallest number of moves. In this case the ideal path through the problem is unclear and you have to *search* for the right path.

One further aspect you may have been aware of was the difficulty you may have had keeping all the necessary information in your mind at once. Try Activity 2.2 and keep a mental watch on your thought processes as you do so.

ACTIVITY 2.2

Try to multiply 243 by 47 in your head.

Tasks such as the one in Activity 2.2 are tricky because the capacity of our working memory is limited – we can only keep so much information in our heads at any one time; overload it and we forget things or lose track of where we are in the problem (see Chapter 4). Other examples of the limits to our capacity to process information are given in Information Box 2.1.

INFORMATION BOX 2.1 PROCESSING LIMITS AND SYMBOL SYSTEMS

The information processing account of problem solving views problem solving as an interaction between the information processing system (the problem solver; either human or computer) and the *task environment* (the problem). By characterising the human problem solver as an information processing system (IPS), Newell and Simon saw no qualitative distinction between human information processors and any other kind, the digital computer being the most obvious example. An IPS processes information in the form of symbols and groups of symbols, and a human IPS has built-in limitations as to how well it processes information. This Information Box gives a very brief sketch of some of the processing limits of human problem solving and what is meant by *symbols*, *symbol structures* and *tokens*.

Processing limitations

The human IPS has certain limitations. It is limited in:

- How much it can keep active in working memory at any one time;
- Its ability to encode information – we may not be able to recognise what aspects of a task are relevant; we don't have the capacity to encode all the information coming through our senses at any one time;
- Its ability to store information – memories laid down at one time can suffer interference from memories laid down later, or may be distorted in line with prior expectations (e.g., Brewer & Treyens, 1981; Loftus, 1996);
- Its ability to retrieve information – human memory, as you may have noticed, is fallible;
- Its ability to maintain optimum levels of motivation and arousal – we get bored, we get tired.

Symbols, symbol structures and tokens

An IPS encodes individual bits of information as symbols which are representations of something. Symbols include things like words in sentences, objects in pictures, numbers and arithmetic operators in equations and so on. These symbols are grouped into patterns known as symbol structures. Knowledge is stored symbols and symbol structures. Figure 2.3 shows an example of a symbol structure for "cat".

A specific occurrence of the word "cat" in the phrase "the cat sat on the mat" is known as a *symbol token*. A symbol token refers the information processor to the symbol itself. As you read the phrase "the cat sat on the mat," you process the individual words. Processing the word "cat" means accessing your stored knowledge associated with the word "cat" and retrieving something that permits the processing (also referred to as "computation") to continue.

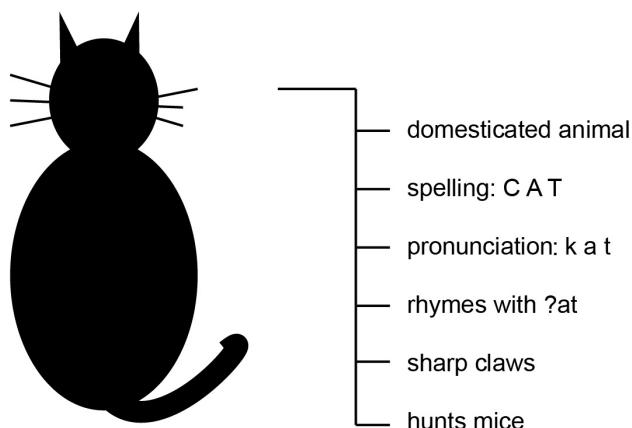


FIGURE 2.3 Symbol structure representing "cat"

When processing “the cat sat on the mat” (which is itself a physical structure of some sort) the local computation at some point encounters “cat”; it must go from “cat” to a body of (encoded) knowledge associated with “cat” and bring back something that represents that a cat is being referred to, that the word “cat” is a noun (and perhaps other possibilities), and so on. Exactly what knowledge is retrieved and how it is organized depend on the processing scheme. In all events, the structure of the token “cat” does not contain all the needed knowledge. It is elsewhere and must be accessed and retrieved.

(Newell, 1990, p. 74)

In order to investigate the processes used in problem solving we first need to find a way to characterise or analyse the task. The next few sections therefore deal with task analysis and how the task is understood by the solver.

Analysing well-defined problems

To see how we can analyse well-defined problems we shall use the Tower of Hanoi problem as our example. Well-defined problems conform to the IGOR format. The initial state, goal state, operators and restrictions for the Tower of Hanoi problem are given in Figure 2.4.

operators: Move rings
 restrictions: Move only one ring at a time

Do not put a ring on a smaller ring
 Rings can be placed only on pegs (not on the table, etc.).

Now try Activity 2.3.

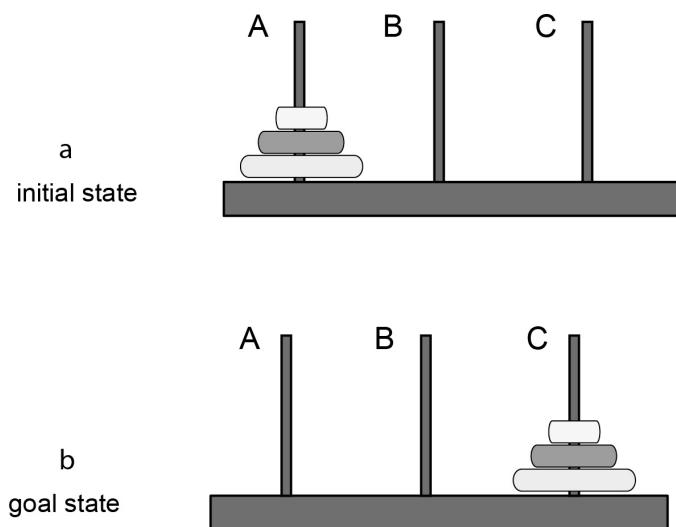


FIGURE 2.4 The initial state, goal state, operators and restrictions in the Tower of Hanoi problem

ACTIVITY 2.3

Identify the initial state, goal state, operators and restrictions for the following coin problem:

Starting with an arrangement of coins in Figure 2.5:



FIGURE 2.5A The eight coins problem

rearrange them so that they end up in the following arrangement:



FIGURE 2.5B The eight coins problem

A coin can only move to an empty space adjacent to it. A coin can jump over only ONE coin of either colour. Silver (white) coins can only move to the right and copper (black) coins to the left.

In the Tower of Hanoi problem and the Eight Coins problem in Activity 2.3 the operator is simply *move*. For example, you can apply the *move* operator to the smallest ring in the Tower of Hanoi problem and put it either on the middle or rightmost peg. In the Eight Coins problem there are four possible initial moves. In either case you will have changed the state of the problem. Figure 2.6 shows the different states that can be reached when the move operator is applied twice to the Tower of Hanoi problem.

In state 1 only the smallest ring can move and there are two free pegs it can move to. If the solver places it on peg C then the problem is now in state 2. In state 2 there are three moves that can be made. The smallest ring can move from peg C back to peg A, which takes the solver back to the initial state, state 1. Alternatively the smallest ring can move from peg C to peg B leading to state 3, or the middle-sized ring can move to peg B leading to state 4. If you carry on this type of analysis then you end up with a diagram containing all possible states and all possible moves leading from one state to another. Although the only action you need to perform in the Tower of Hanoi problem is “move”, other problems may involve a variety of mental operators. For this reason the diagram you end up with is known as a *state-action diagram*.

State-action spaces

Thinking through a problem can be a bit like trying to find a room in an unfamiliar complex of buildings such as a university campus or hospital. Suppose you have to get to room A313 in a complex of buildings. Initial attempts to find your way may involve some brief exploration of the

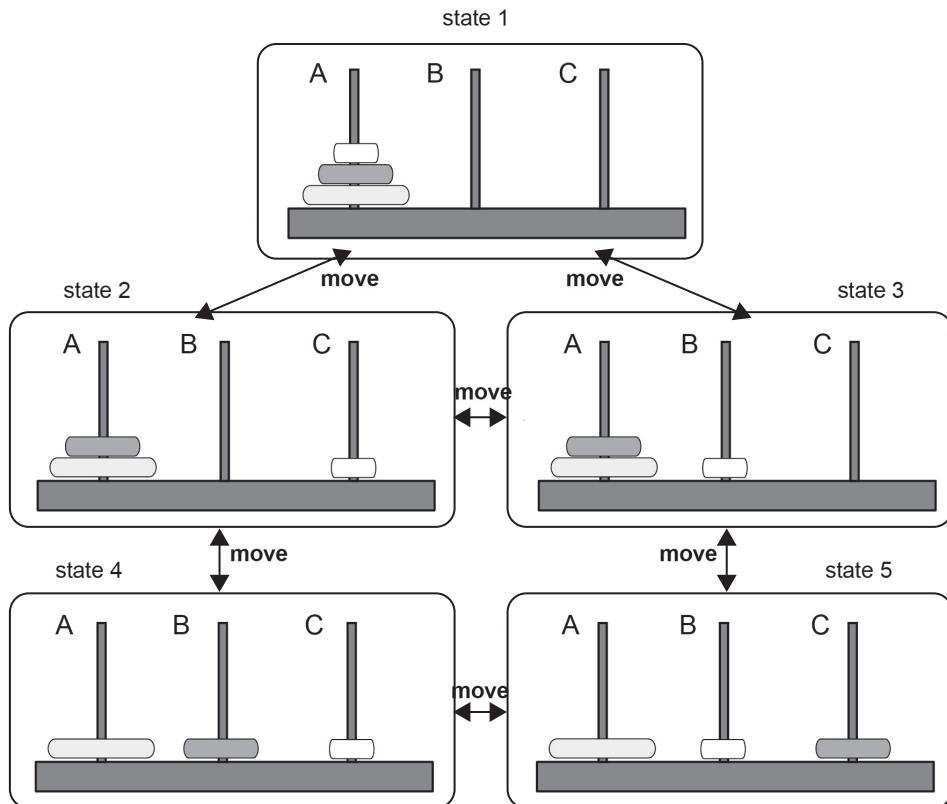


FIGURE 2.6 The possible states that can be reached after two moves in the Tower of Hanoi problem

buildings themselves. This initial exploration, where you are trying to gather useful information about your problem environment, can be characterised as an attempt to understand the problem. You discover that the buildings have names but not letters. However, one building is the Amundsen Building. Since it begins with an “A” you assume (hypothesise) that this is the building you are looking for so you decide to find out (test this hypothesis). You enter and look around for some means of getting to the third floor (accessing relevant operators). You see a stairwell and a lift next to it. You take the lift. When you get out on the third floor you see swing doors leading to corridors to the right and left. Outside the swing doors on the left is a notice saying “301–304, 321–324” and outside the one on the right is the sign “305–308, 317–320”. You want 313, so now what do you do? (This, by the way, is a real example from a real university).

This analogy likens problem solving to a search through a three-dimensional space. Some features of the environment in which the problem is embedded are helpful, and some less so, leaving you to make inferences. The room numbers are sequential to some extent, although it's not obvious why there are gaps. Furthermore, in trying to find your way through this space you call upon past knowledge to guide your search. The problem of finding the room is actually fairly well-defined – you know where you are, you know where you want to be, you know how to get there (walk, take the lift, take the stairs) even if you don't actually know the way.

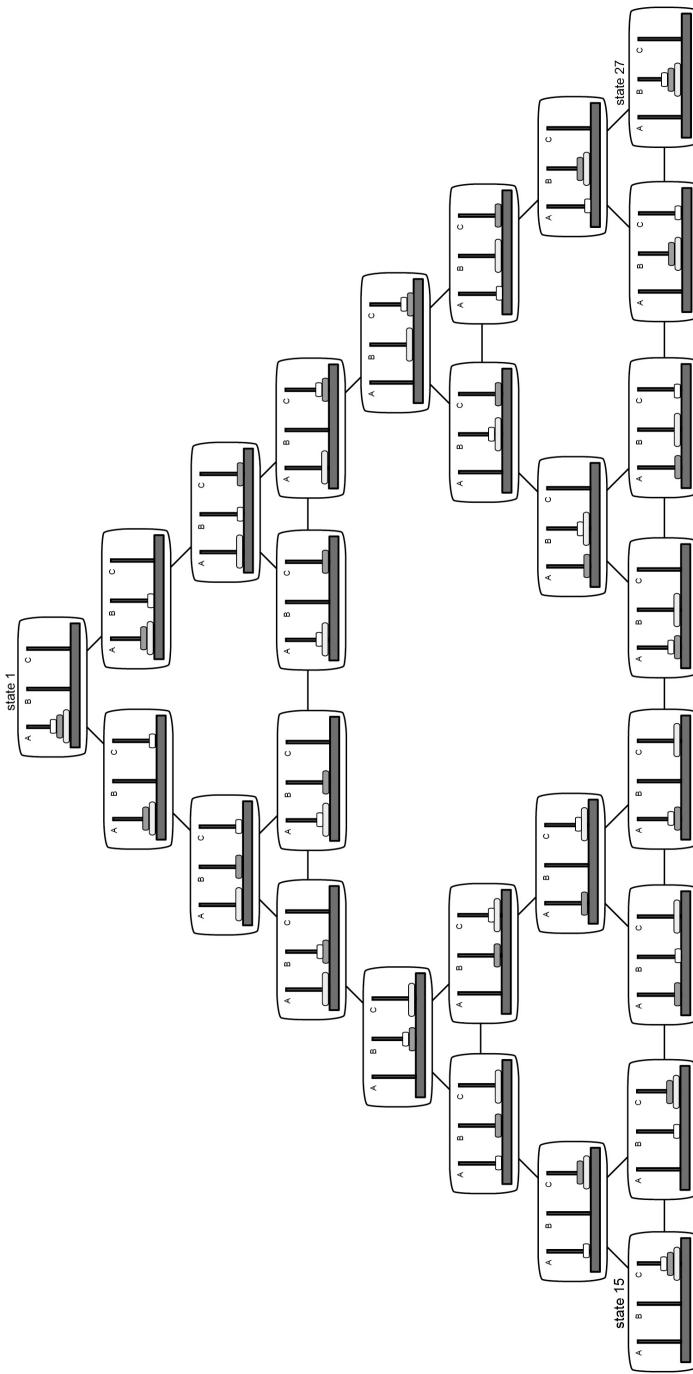


FIGURE 2.7 State space of all legal moves for the three-ring Tower of Hanoi problem

Activity 2.1 showed that problem solving can be regarded as moving from one “place” in the problem to another. As you move through the problem your knowledge about where you are in the problem has to be updated; that is, you go through a sequence of knowledge states. If you carry on the analysis of what choices are available at each state of the Tower of Hanoi problem as in Figure 2.6 you end up with a complete *search tree* for the three-ring version of the problem (Figure 2.7). In Figure 2.7 you can see that each state is linked to three others except at the extremities of the triangle where there is a choice of only two moves: at states 1, 15 and 27. Figure 2.7 constitutes a *state-action space*, or more simply *state space*, of all legal moves (actions) for the three-ring Tower of Hanoi problem, and all possible states that can be reached. In tree diagrams of this sort the points at which the tree branches are called *nodes*. Each of the numbered states is therefore a node of the state space diagram. The Eight Coins problem has some dead ends, as well as a goal state from which no other legal moves are possible. These would therefore constitute *terminal nodes* or leaf nodes.

The space of all possible states in a problem as exemplified in Figure 2.7, “represents an omniscient observer’s view of the structure of a problem” (Kahney, 1993, p. 42). Limits to how much we can store in short-term memory mean that we cannot mentally represent the entire search tree for the Tower of Hanoi problem. Indeed, other problems have search trees vastly more complicated than the Tower of Hanoi problem; there are hundreds of possible states you can reach in the Eight Coins problem in Activity 2.3. This means that our mental representation of the problem is likely to be impoverished in some way, which, in turn, means that the path through the problem may not be all that clear. No system, human or computer, can encompass the entire state space of a game of chess, for example. Indeed the size of the space for a typical chess game is estimated to be 10^{120} . Newell and Simon (1972) referred to the representation we build of a problem as the *problem space* which is “the fundamental organizational unit of all human goal-oriented activity” (Newell, 1980, p. 696).

To sum up: “The task is defined objectively (or from the viewpoint of an experimenter, if you prefer) in terms of a task environment. It is defined by the problem solver, for purposes of attacking it, in terms of a problem space” (Simon & Newell, 1971, p. 148). A person’s mental representation of a problem, being a personal representation, cannot be “pointed to and described as an objective fact” (Newell & Simon, 1972, p. 59). As mentioned earlier, various sources of information combine together to produce a problem representation. These are summarised in Information Box 2.2.

INFORMATION BOX 2.2 TYPES OF INFORMATION THAT CAN BE USED TO GENERATE A PROBLEM REPRESENTATION

The main sources of information are:

- *The task environment.* A well-defined problem itself is the main source of information about how to construct a relevant problem space. It defines the initial state and goal state and may provide information about possible operators and restrictions. People are also influenced by parts of the problem statement that appear particularly salient.

- *Inferences about states, operators and restrictions.* Any information missing from the problem statement may have to be inferred from the person's long-term memory. For a problem such as "Solve: $(3x + 4) + x = 20$," not all operators are provided and the solver has to access the necessary arithmetic operators from memory. It is also left to the solver to infer what the final goal state is likely to look like so it can be recognised when it is reached.
- *Text-based inferences.* Other inferences may have to be generated from the text of a problem. For example, if the problem involves one car leaving half an hour after another and overtaking it, the solver will (probably) infer that both cars have travelled the same distance when they meet (Nathan, Kintsch, & Young, 1992).
- *Previous experience with the problem.* The solver may have had experience with either the current or a similar problem before, and can call upon this experience to help solve it.
- *Previous experience with an analogous problem.* The solver may recognise that the structure of an earlier problem which, superficially at least, seems unrelated to the current one is actually relevant to the solution to the current one. For example, the equation in a problem involving the distance travelled by a car travelling at a certain speed may be identical to one involving currency exchange rates even though both problems are from different domains. The likelihood of this happening is usually fairly low. When it does happen it may constitute an "insight".
- *Misinformation.* The solver may construct a problem space based on a misapprehension of some aspect of the problem.

States represented by nodes of the search space need not correspond with realizable states of the outside world but can be imaginable states – literally so since they are internal to the problem solver. These states, in turn, may be generated, in turn, by operators that do not satisfy all the conditions for admissibility.

(Newell & Simon, 1972, p. 76)

For example, someone might decide to move all three rings of the three-ring Tower of Hanoi problem at once, not realising or remembering that there is a restriction on the number of rings that can be moved at once.

- *Procedures for dealing with problems.* From general problem solving experience the solver has a number of procedures for combining information in the external environment (e.g., the written statement of the problem, the state of the rings and pegs of the Tower of Hanoi problem) with information in long-term memory. Thus the solver might generate context-specific *heuristics* (rules of thumb) for dealing with the problem. For example, in a crossword puzzle the solver may pick out the letters believed to form an anagram and write them down in a circle to make it easier to solve.
- *External memory.* The external environment may contain clues to the particular state a problem is in. The Tower of Hanoi problem, for example, provides constant information about the state of the problem since it changes to a visibly new state each time you move a ring. Similarly, the little scribbles or 1s one might write on a subtraction or addition problem serve as an external memory to save us having to try

to maintain the information in working memory. These days, diaries, organisers, telephone numbers and to-do lists stored in smartphones, tablets and so forth all serve as forms of external memory.

- *Instructions.* Newell (1990, p. 423) further argued that problem spaces come from instructions. In a psychological experiment involving reaction times, for example, there are liable to be trial-specific instructions a participant would be given immediately before a particular trial of the experiment. There may also be prior general instructions before the experiment begins, and introductory instructions when the participant walks through the door that provide the general context for the experiment and what the participant is expected to do.

All these sources of information together constitute the “space” in which a person’s problem solving takes place. Together they allow us to define the nodes (the states) of a problem and links between them along with a possible strategy for moving from node to node. Hayes (1989) provides the following analogy for a problem space:

As a metaphor for the problem solver’s search for a solution, we imagine a person going through a maze. The entrance to the maze is the initial state of the problem and the exit is the goal. The paths in the maze, including all its byways and blind alleys, correspond to the problem space – that is, to all the possible sequences of moves available to the solver.
(p. 35)

According to Newell and Simon, problem solving involves finding your way through this problem space (searching for a way through the maze or looking for room A313). Because of the limits of working memory we can only see a very few moves ahead and may not remember all the states that have been visited before. Figure 2.8 tries to capture some of the information an individual might access when trying to understand a novel problem. The shading in the figure represents the fact that we can only see a few moves ahead and are likely to have only a couple of previous states in working memory at one time. The state of the problem that is in focal attention is the currently active knowledge state.

Although I have made a distinction between a state space as the space of all possible moves in a problem, and a problem space as the individual representation a person has of the problem, you will often see the term “problem space” used to describe something like a search tree such as Figure 2.7. Reed, Ernst and Banerji (1974), for example, refer to their version of Figure 2.7 as the problem space of legal moves. Similarly, Hunt (1998) describes the game of chess thus:

In chess the starting node is the board configuration at the beginning of the game. The goal node is any node in which the opponent’s king is checkmated. The (large) set of all legal positions constitutes what Newell and Simon call the *problem space*.
(p. 221)

The kind of problem space referred to in both cases is the *objective problem space* – the set of all possible paths and states a solver *could* theoretically reach given the initial state, operators and

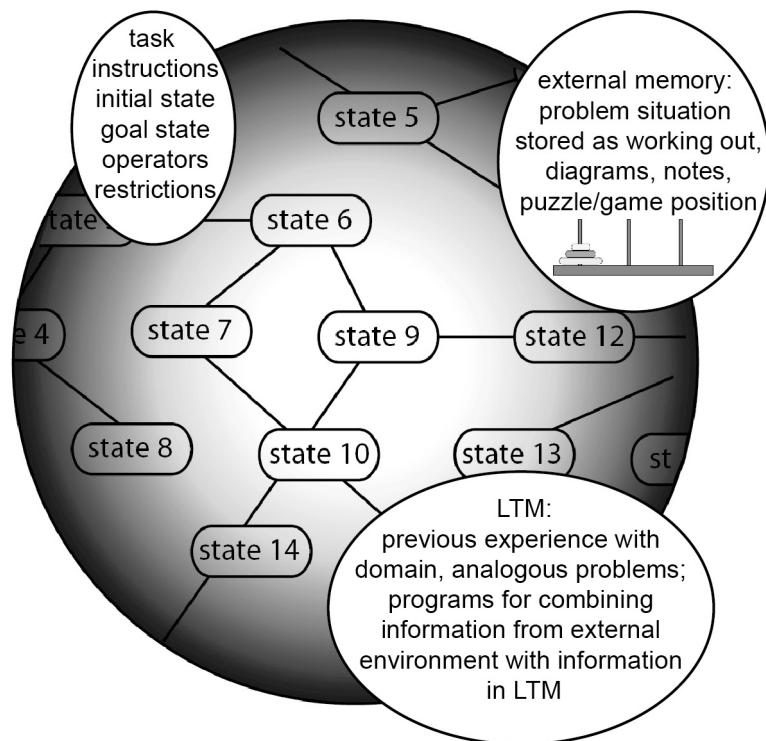


FIGURE 2.8 Sources of information used to determine a problem space

some way of evaluating when the goal has been reached. The solver (Newell and Simon refer to the IPS) “incorporates the problem space, not in the sense of spanning its whole extent, but in possessing symbol structures and programs that provide access to that space via the system’s processes for generating, testing and so on” (Newell & Simon, 1972, p. 78).

Apart from being unable to encompass the sheer size of some state spaces, people are usually just not aware of them.

An incontestable principle of cognition is that people are not necessarily aware of the deductive consequences of their beliefs, and this principle applies to problem spaces as well. Although the state space is a deductive consequence of the initial state and operators, people are not aware of all of it.

(VanLehn, 1989, pp. 531–532)

The actual path solvers take through a state space does not depend on them being aware of all of it. People usually manage to find ways of limiting the search space in some way, and that is what problem solving research is interested in.

The interaction of the problem solver and the task environment

Concentrating on the task environment sounds as though an analysis of the task itself is enough to show how a person can perform the task (i.e., solve the problem). This, of course, is

not the case. People are not perfectly rational, and analysing the task in detail does not necessarily tell us how or if the solver can actually solve the problem. Nor does a task analysis tell us that the solver will use that particular representation (problem space) to solve the problem. So what is the point of such analyses?

Laird and Rosenbloom (1996) refer to the “principle of rationality” ascribed to Newell that governs the behaviour of an intelligent “agent” whereby “the actions it intends are those that its knowledge indicates will achieve its goals” (p. 2). Newell and Simon argue that behaviour is usually *rational* in the sense that it is adaptive. This means that people’s problem solving behaviour is an appropriate response to the task, assuming that they are motivated to achieve the goal demanded by the task. Todd and Gigerenzer (Gigerenzer, 2015; Gigerenzer & Todd, 1999; Todd & Gigerenzer, 2000) would also add that we have developed problem solving shortcuts (“fast and frugal heuristics”) that achieve our goals with the least computational effort. They make the point that “if there is such a thing as behavior demanded by a situation, and if a subject exhibits it, then his [sic] behavior tells us more about the task environment than about him” (Newell & Simon, 1972, p. 53). If we want to know about the psychology of problem solving behaviour, then examining behaviour that is entirely governed by the task environment tells us nothing. If, on the other hand, our problem analysis reveals (as far as such a thing is possible) how a perfectly rational person would solve the problem and we then compare this to what someone actually does when confronted with the problem, then the difference between the two tells us something about the psychology of the solver. We can therefore get some idea of how difficult a problem is by examining the interaction between the task environment and what the problem solver does. Figure 2.9 represents the interaction of the solver in a task environment.

When the IPS examines the environment, it generates an internal representation of that environment based on the problem statement in its context. This representation involves the selection of a problem space. The main source of a problem space is therefore the instructions given to the solver which can be specific to the problem or general – “I’d like you to take part in a psychology experiment in sentence verification” (Newell, 1990). It would follow that the choice of a problem space can be influenced by manipulating the salience of objects in the task environment or the givens in a problem statement. Input processes are under the control of the system’s general knowledge and overall goals. The selection of a problem space results in the system choosing appropriate problem solving methods. A method is “a process that bears some rational relation to attaining a problem solution” (Newell & Simon, 1972, p. 88). Problem solving methods come in two general types: *strong* and *weak*. Strong methods are domain-specific, learned methods that are pretty much guaranteed to get a solution and used when you already know how to go about solving the problem. They can incorporate an algorithmic method: a sequence of specific instructions that will lead to a goal. Of course, if as a result of reading a problem you already know what to do (you have an available strong method), then the problem should be straightforward. Weak methods are general-purpose problem solving strategies that solvers fall back on when they don’t know what to do directly to solve the problem. These methods are discussed in the next section. The particular method chosen thereafter controls further problem solving. The outcome of applying problem solving methods is monitored; that is, there is feedback about the results of applying any particular step in the application of the method. This feedback may result in a change in the representation of the problem (see Figure 1.1 in Chapter 1).

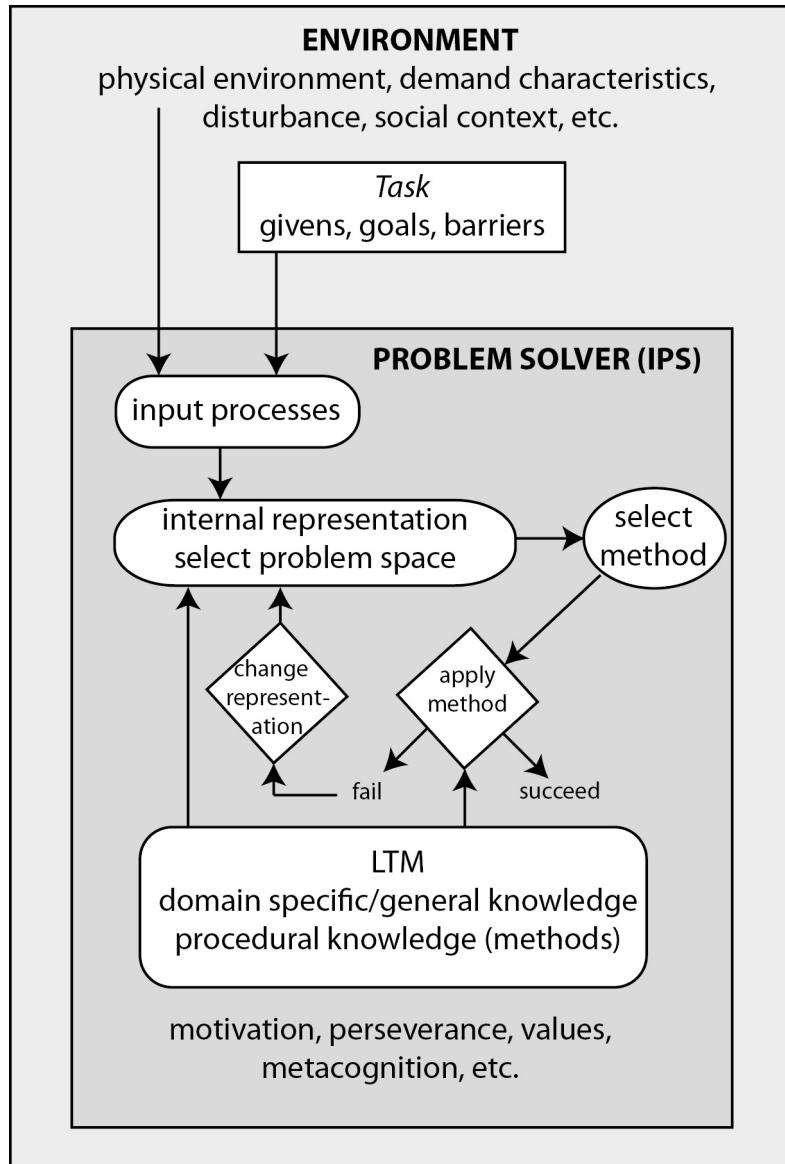


FIGURE 2.9 The problem solver and the task environment

Heuristic search strategies

Searching for a solution path is not usually governed by trial and error, except in a last resort or where the search space is very small. Instead people try to use heuristics to help them in their search. Heuristics are rules of thumb that help constrain the problem in certain ways (in other words they help you to avoid falling back on blind trial and error), but they don't guarantee that you will find a solution. Heuristics can be contrasted with algorithms that will guarantee that you find a solution – it may take forever, but if the problem is algorithmic you will get there. Examples of the two are provided in Information Box 2.3.

INFORMATION BOX 2.3 ALGORITHMS AND HEURISTICS

To illustrate the difference between algorithms and domain-specific heuristics, imagine how you might go about solving a jigsaw puzzle.

Algorithmic approach

Starting off with a pile of jigsaw pieces, an algorithm that is guaranteed to solve the jigsaw might proceed as follows:

- 1 Select piece from pile and place on table.
- 2 Check > 0 pieces left in pile.
IF YES go to 4
ELSE go to 3
- 3 Check > 0 pieces in discard pile.
IF YES discard pile becomes pile; go to 2
ELSE go to 7
- 4 Select new piece from pile.
- 5 Check whether new piece fits piece (or pieces) on table.
IF YES go to 6
ELSE put new piece on discard pile; go to 2
- 6 Check colour match
IF YES leave new piece in place; go to 2
ELSE put new piece on discard pile; go to 2
- 7 FINISH

Domain-specific heuristics serve to narrow your options and thus provide useful constraints on problem solving. However, there are other, more general heuristics that a solver might apply. When you don't know the best thing to do in a problem the next best thing is to choose to do something that will reduce the difference between where you are now and where you want to be. Suppose you have a 2,000-word essay to write and you don't know how to go about writing a very good introductory paragraph. The next best thing is to write down something that seems vaguely relevant. It might not be particularly good but at least you've only got 1,800 words left to write. You are a bit nearer your goal. In the Tower of Hanoi problem this means that the solver will look at the state she is in now, compare it with where she wants to be (usually the goal state) and choose a path that takes her away from the initial state and nearer to the goal state. This general type of heuristic is called *difference reduction*, the most important examples of which are known as *hill climbing* and *means–ends analysis*.

Hill climbing

The term hill climbing is a metaphor for problem solving in the dark, as it were. Imagine that you are lost in a thick fog and you want to climb out of it to see where you are. You have a choice of four directions: north, south, east and west. You take a step north – it seems to lead down, so you withdraw your foot and turn 90° to the east and take another step. This time it seems to lead upward, so you complete the step and try another step in the same direction. It also seems to lead upward, so you complete the step and try again. This time the path leads downwards, so you withdraw your foot, turn 90° and try again. You carry on doing this until there comes a time when no matter which direction you turn, all steps seem to lead down. In this case you are at the top of the hill. This kind of local search heuristic is useful if, say, you are trying to find your way out of a maze when you know which direction the exit is and you try to get closer and closer to it. Inevitably there will be dead ends requiring backtracking but, nevertheless, choosing a path that seems to lead closer to the exit is a reasonable strategy. Suppose you are trying to find a video channel on a television with which you are unfamiliar. There are two buttons marked $-P$ and $+P$ and you can't work out what they stand for. Lacking any strong method for finding the correct channel, you fall back on weak methods. There are three choices for you: you can either press $-P$, press $+P$ or press them both together. Past experience might tell you that pressing both together might not be a good idea – at best they might just cancel each other out. That reduces the choice to two. What you are left with is a kind of trial and error method which is about as weak a method as you can get. You decide to press $+P$ and a recognisable channel appears along with the channel number on the top left of the screen. As a result, you may infer that pressing $+P$ steps through the channels in an ascending order and $-P$ in a descending order. Applying a trial and error method once may therefore be enough to allow you to re-represent the problem based on monitoring the result of your actions. On the other hand if the button pressing strategy fails and nothing appears on the screen, you may have to change the problem space to one that involves searching for a remote control or finding someone who knows how this television works, or – heaven forfend – trying to find the instructions.

So although hill climbing will take you eventually to the top of a hill, there is no guarantee that it is the top of the highest hill. You may emerge from the fog only to find you are in the foothills and the mountain peaks are still some way off. (Dennett, 1996, has argued that this “local” selecting of the next step is how Darwinian selection operates.) Another problem with the method is that it only applies if there is some way of measuring whether you are getting closer to the goal. If no matter which way you step the ground remains flat, then any direction is as good as any other and your search for a way out of the fog will be random. Anyone who has ever got lost in the middle of an algebra problem will know what this feels like. You might end up multiplying things and subtracting things and nothing you do seems to be getting anywhere near the answer.

A puzzle that has been often used to examine hill climbing is the so-called Missionaries and Cannibals problem. Subjects therefore tend to use hill climbing as their main strategy to reduce the difference between where they are in the problem and where they want to be. The basic version of the problem is in Activity 2.4. Have a go at it before viewing Figure 2.10 to see the complete problem space of legal moves.

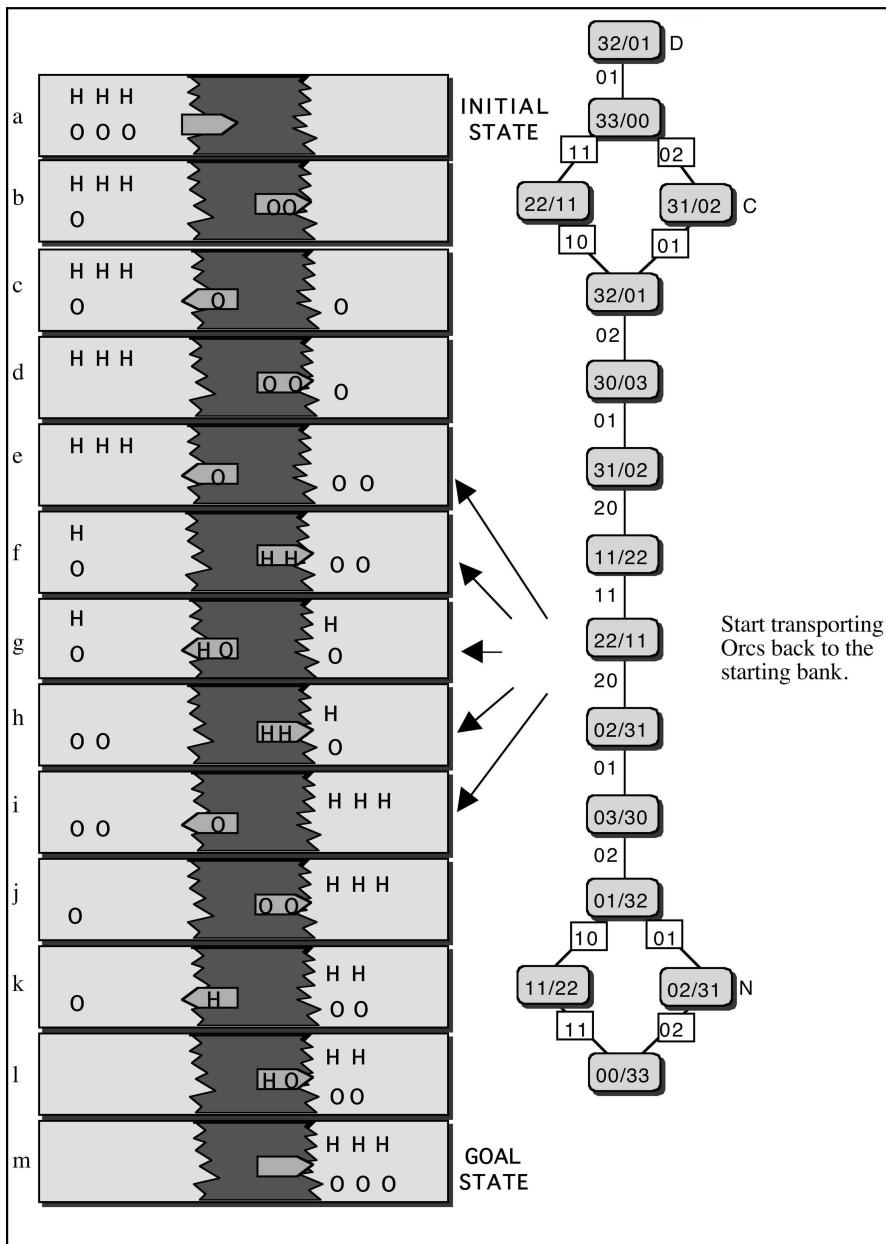


FIGURE 2.10 Solution and state-action graph of the Hobbits and Orcs problem

Adapted from Reed et al. (1974).

ACTIVITY 2.4

Three missionaries and three cannibals having to cross a river at a ferry, find a boat but the boat is so small that it can contain no more than two persons. If the missionaries on either bank of the river, or in the boat, are outnumbered at any time by cannibals, the cannibals will eat the missionaries. Find the simplest schedule of crossings that will permit all the missionaries and cannibals to cross the river safely. It is assumed that all passengers on the boat disembark before the next trip and at least one person has to be in the boat for each crossing.

(Reed et al., 1974, p. 437)

There are a number of variations of the basic problem including the Hobbits and Orcs problem (Greeno, 1974; Thomas, 1974), where Orcs will gobble up Hobbits if there are fewer Hobbits than Orcs; book-burners and book-lovers (Sternberg, 1996), where book-burners will burn the books of the book-lovers if they outnumber them; and scenarios where, if there are fewer cannibals than missionaries, the missionaries will convert the cannibals (Eisenstadt, 1988; Solso, 1995). The structure of the problem is interesting since there are always two legal moves. One will take you back to the previous state and one will take you nearer the solution (Figure 2.10).

Figure 2.10 contains a pictorial representation and a state-action graph of the Hobbits and Orcs problem to make it a little easier to see what the graph represents. The figures in the ovals represent the numbers of Hobbits and Orcs, in that order, on both banks at any one time (the figure on the left always refers to the number of Hobbits and the figure on the right always refers to the number of Orcs). Thus 33/00 means that there are 3 Hobbits and 3 Orcs on the left bank (left of the slash) and 0 Hobbits and 0 Orcs on the right bank (right of the slash). On or beside the lines linking the ovals there are figures representing who is in the boat. Thus, on the line linking state B to state E, the 10 means that there is one Hobbit and no Orcs on the boat. From the initial state (state A) there are three legal moves that will lead you to states B, C or D. If one Orc were to take the boat across from the initial state, then there would be 3 Hobbits and 2 Orcs on the left bank and no Hobbits and 1 Orc on the right bank, as in state D. This is a pointless move, since the only possible next move is for the Orc to take the boat back to the left bank (i.e., back to state A). It makes more sense for two individuals to take the boat: either two Orcs (leading to state C) or 1 Hobbit and one Orc (leading to state B).

It looks from Figure 2.10 that, if you avoid illegal moves and avoid going back to an earlier state, you ought to get straight to the solution. So what makes this problem hard? If hill climbing is the method used to solve this task, then one can make some predictions about where people will have difficulty. If you are standing in a metaphorical fog at night and you are trying to take steps that will lead you uphill, and if you get to a “local maximum” (where every step you take seems to take you downhill), then you are stuck. If the same thing happens in the middle of a problem, then problem solving will be slowed down or you will make mistakes.

The Hobbits and Orcs problem was also studied by Thomas (1974) and by Greeno (1974). Thomas’s study is described in Information Box 2.4. If you look back at state H in Figure 2.10 you will see that there are two Hobbits and two Orcs on the right bank. If the solvers are using some form of hill climbing strategy and trying to reduce the difference between where

they are at state H and where they want to be (*increase* the number of people on the right bank), then state H poses a problem. To move on in the problem, one Hobbit (or missionary) and one Orc (or cannibal) have to move *back* to the left bank, so the subject seems to be moving away from the goal at this point. This is equivalent to taking a step that seems to lead downhill when trying to climb uphill using a hill climbing strategy.

INFORMATION BOX 2.4 HOBBITS AND ORCS (THOMAS, 1974)

Thomas used two groups of subjects. The control group were asked to solve the problem on a computer, and the times taken to solve the first half and second half of the problem were noted. A second group (known as the “part-whole” group) were given prior experience of the second part of the problem starting from state H and were then asked to solve the whole problem (see Table 2.1).

TABLE 2.1 Average number of moves in each part of the Hobbits and Orcs task

<i>Group</i>	<i>First part of task</i>	<i>Second part of task</i>	<i>First attempt (part-whole group)</i>
Control	13.0	15.5	
Part-whole	10.8	14.3	12.0

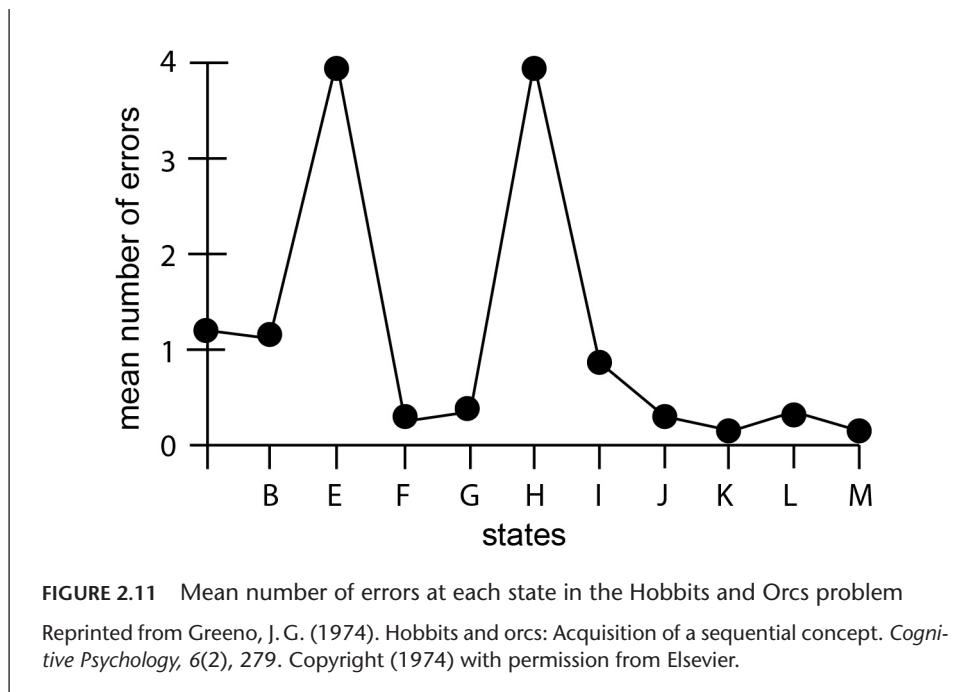
Results

Discussion

The part-whole group performed significantly better than the control group on the first half of the task but there was no significant difference between them on the second part. This suggests that the prior experience the part-whole group had on the second half of the problem benefited them when they came to do the first part of the whole problem afterward. However, the fact that there was no difference between the groups on the second half – in fact, the part-whole group did worse on their second attempt than they did on their first – suggests that there was a strong context effect when they came to state H and were reluctant to make a “detour”. The results show that a hill climbing / difference reduction strategy exerts a strong influence on problem solving, making subjects reluctant to choose moves that seem to be leading them away from the goal.

There was also another point at which subjects in Thomas’s study seemed to get stuck. At state E in Figure 2.10, there are three legal moves leading to states F, B and C. It is the only place in the problem where subjects can move backward without returning to an earlier problem state. In fact, solvers may find themselves caught in a loop at this point, moving, for example, from B to E to C to A to B and then back to E.

Another difficulty subjects seemed to come up against was not realising that a particular move would lead to an error. In other words their Hobbits kept getting eaten by the Orcs. Again the problems centred around states E and H. Figure 2.11 shows the average number of errors made by subjects in Greeno’s (1974) experiments.



Figures showing delays before making a move (the latencies) at each state follow a very similar pattern with states E and H once again causing difficulties for the subjects. The longest response time occurs at state A where subjects are thinking about the problem and planning how to move. On successive attempts at the problem this delay is greatly reduced.

What these studies show us is that you don't need to have a complicated state space for problem solving to be hindered. If the main method used is hill climbing, then you are likely to experience difficulty when you have to move away from the goal state to solve the problem. There is, however, a more powerful difference reduction strategy for searching through a problem space.

Means–ends analysis

The most important general problem solving heuristic identified by Newell and Simon was means–ends analysis, where solvers also try to reduce the difference between where they are in a problem and where they want to be. They do so by choosing a mental operator or choosing one path rather than another that will reduce that difference, but the difference between this heuristic and hill climbing is that the problem solver has a better idea of how to break the problem down into sub-problems. Suppose you want to go on holiday to Georgioúpoli in Crete. Let's assume for the sake of the example that you like to organise holidays by yourself and avoid package holidays as much as possible. You can begin to characterise the problem in the following way:

INITIAL STATE: you at home in Milton Keynes (well, somebody has to live there)

GOAL STATE: you at Georgioúpoli

There are several *means* (operators) by which you normally travel from one place to another. You can go: on foot, by boat, by train, by plane, by taxi, and so on. However, your general knowledge of the world tells you that Crete is a long way away and that it is an island. This knowledge allows you to restrict your choice of operators.

RESTRICTIONS: Crete is a long way away
you want to get there as quickly as possible

Your general knowledge tells you that the fastest form of transport over land and sea is the plane. So,

OPERATOR: go by plane

Unfortunately your problem is not yet solved. There are certain preconditions to travelling by plane, not the least of which is that there has to be an airport to travel from and there is no airport in Milton Keynes. You are therefore forced to set up a *sub-goal* to reduce the difference between you at home and you at the airport. Once again you have to search for the relevant *means* to achieve your *end* of getting to the airport. Some operators are not feasible due to the distance (walking); others you might reject due to your knowledge of, say, the cost of parking at airports, the cost of a taxi from home to the airport and so on. So you decide to go by train.

INITIAL STATE: you at home

SUB-GOAL: you at the airport

RESTRICTIONS: you don't want to spend a lot of money
you don't want to walk

OPERATOR: go by train

Once again the preconditions for travelling by train are not met, since trains don't stop outside your house, so you set up a new sub-goal of getting from your home to the station and so on.

Means–ends analysis therefore involves breaking a problem down into its goal–sub-goal structure and should provide a chain of operators that should eventually lead you to the goal. This method of problem solving is also known as *sub-goaling*. It can also be applied to the Tower of Hanoi problem. Simon (1975) outlined three different strategies that involved decomposing the goal into sub-goals. One of them is the *goal recursion strategy*. If the goal is to move all three rings from peg A (see Figure 2.12a) to peg C, then first move the two top rings from peg A to peg B so that the largest ring can move to peg C (Figure 2.12b). Then set up the sub-goal of moving the two ring pyramid on peg B to peg C. Another more “trivial” sub-goal involves moving the two ring pyramid from one peg to another. If the goal is to move two rings from peg A to peg B, then first move the smallest ring from peg A to peg C (Figure 2.12c).

This form of sub-goaling is also known as *working backwards* since you are analysing the problem by starting off from the goal state and working backwards from it to see what needs to be done (i.e., what sub-goals need to be achieved). The reason why it's called a *recursion* strategy is because the procedure for moving entire pyramids of rings (e.g., three rings, five

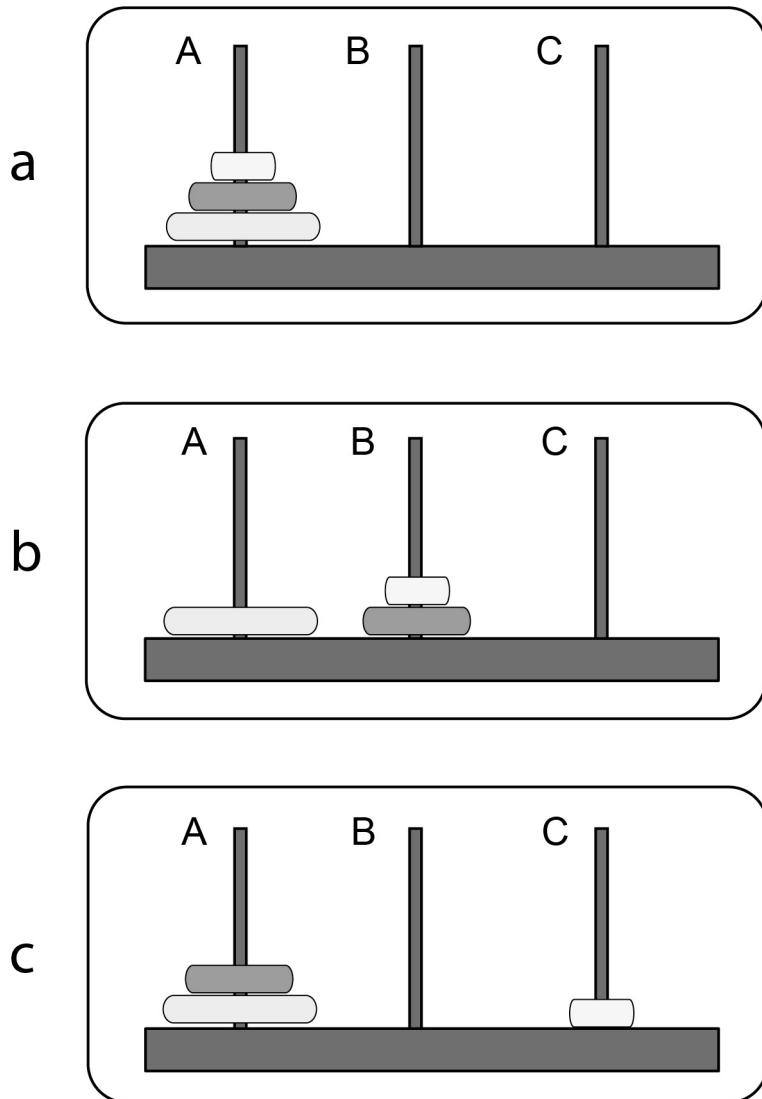
50 Problem representation

FIGURE 2.12 The goal recursion strategy of solving the Tower of Hanoi problem

rings, 64 rings) involves moving the entire pyramid *minus one* ring. And the procedure for moving the pyramid-minus-one-ring involves moving the pyramid-minus-one-ring *minus one* ring, and so on. The recursive procedure can be written as follows:

To move Pyramid(k) from A to C
Move Pyramid($k - 1$) from A to B
Move Disk(k) from A to C
Move Pyramid($k - 1$) from B to C

(Simon, 1975, p. 270)

where k is an (odd) number of rings (the substructure of the Tower of Hanoi problem is illustrated in Figure 2.7). The difficulty with this strategy for solving the problem is the potential load on short-term memory.

For a subject to use this recursive strategy, he must have some way of representing goals internally, and holding them in short-term memory while he carries out the sub-goals. How large a short-term memory this strategy calls for depends on how many goals have to be retained in short-term memory simultaneously.

(Simon, 1975, p. 270)

Having a set of goals and sub-goals, presumably in some kind of priority ordering, is the idea behind a *goal stack*.

Goal stacks

The idea that we have a “stack” of goals that we keep in mind as we solve some types of problem comes from computer science. Some computer programs are hierarchically ordered so that in order to achieve goal A you first have to achieve goal B, to achieve goal B you have to achieve goal C and so on. Newell and Simon (1972) give the example of driving a child to school (the main goal), but the car doesn’t work so you have to fix the car first (a new goal), and to fix the car you have to call the garage (a new goal) and so on. Notice that these goals and sub-goals are interdependent, each one depending on the next one “down”. When you embark on a new goal – when a new goal comes to the top of the *goal stack* – this is called *pushing* the stack. When the goal is satisfied (for example, when you have made the phone call to the garage), the goal of phoning is dropped from the stack – known as *popping* the stack – and you move to the next goal down: waiting for the mechanic to repair the car.

The psychological reality of human beings using such a goal stack and the effects of interruptions or interference on tasks as they solve problems is discussed in Anderson (1993), Anderson and Lebiere (1998) and Hodgetts and Jones (2006). Aspects are discussed in Information Box 2.5.

INFORMATION BOX 2.5 THE PSYCHOLOGICAL VALIDITY OF GOAL STACKS

Anderson (1993) argues that much of our behaviour can be described as arising from a hierarchy of goals and sub-goals. For example, a subject in a laboratory experiment

may be trying to solve a Tower of Hanoi problem in order to satisfy a subject participation requirement in order to pass a psychology course in order to get a college degree in order to get a job in order to earn social respect.

(Anderson, 1993, p. 48)

This hierarchy of goals is modelled in the ACT-R model of cognition (see Chapter 5) as a goal stack. The goals, as in the example of the laboratory subject, are interdependent but

one can only make progress by dealing with the currently active goal – the one at the top of the goal stack. Because of this interdependency of goals, and the logic of having to deal with the currently active one, “goal stacks are a rational adaptation to the structure of the environment” (p. 49). Since goal stacks are an “adaptation to the environment” Anderson argues that (1) we have evolved cerebral structures that coordinate planning (the prefrontal cortex) and (2) we can find evidence of hierarchical planning in other species.

The decision to design a tool is a significant subgoal (a means to an end), and the construction of a tool can involve complex co-ordination of subgoals under that. Neither hierarchical planning nor novel tool use are uniquely human accomplishments and they are found to various degrees in other primates.

(Anderson, 1993, p. 49)

There is an argument that certain types of forgetting are not dealt with adequately in the notion of a goal stack. For example, people sometimes forget to remove the original from a photocopier after making photocopies. This can be explained by arguing that making and collecting the photocopies is the goal and this does not include removing the original (Anderson, 1993, p. 48). Byrne and Bovair (1997) argue that, when the super-goal of making and getting copies is satisfied, it is popped from the stack along with its associated sub-goals, and so forgetting the original should happen every time: “the goal structure for tasks like the photocopier never changes, so the error should persist indefinitely” (Byrne & Bovair, 1997, p. 36).

On the other hand, if the post-completion step (removing the original) is always placed on the stack as part of the overall goal, then the error should *never* happen. However, the important words in that last sentence are “if” and “always”.

Another spanner in the works is that of interruptions. If a new interrupting goal is presented, then this is added to the top of the goal stack and the rest are “pushed down”. When that goal is satisfied, ideally the previous goal should “reappear” and be dealt with, but there are often failures of prospective memory and action slips are a result (Hodgetts & Jones, 2006).

Regarding human problem solving behaviour as the outcome of an IPS provides a way of testing theories by modelling them on another IPS (a computer, a robot) and also provides a way to test our theories against the processing that goes on in the brain. The information processing approach can describe the general processes that govern problem solving and provides a nomothetic account of such behaviour (i.e., it provides something close to general laws), but what is less clear is how it might account for individual differences in aspects of problem solving. These are presumably due to differences in knowledge, experience, motivation, personality traits and indeed intelligence.

Summary

- 1 Problem solving involves building a mental representation of a situation using whatever information is available in the environment or from long-term memory.

- 2 Information processing accounts of problem solving emphasise the interaction of the problem solver and the environment (in this case the problem). Newell and Simon (1972) suggested that problem solving involved two co-operating processes called *understanding* and *search*.
- 3 Understanding refers to the process of building a representation of a problem. This representation constitutes a problem space derived from a combination of:
- What it says in the problem statement;
 - The problem solving context;
 - What inferences you can draw from it based on general knowledge;
 - Your past problem solving experience.
- 4 Armed with this representation, the IPS engages in a search to find a path through the problem that will lead to a solution. Search is the process whereby the solver attempts to find a solution within the problem space.
- 5 *Search* and *understanding* interact. The search process might lead to the solver revising the mental representation or the solver may re-read the problem or parts of it (an aspect of problem understanding), which in turn may suggest ways in which the search for the solution can continue.
- 6 To guide search through a problem space people tend to use strategies called heuristics. The main types involve trying to find a way of reducing the difference between where you are now and where you want to be. One such fairly blind method is called hill climbing, where the solver heads blindly in a direction that seems to lead to the solution. A more powerful method is means–ends analysis, which can take into account the goal–sub-goal structure of problems.
- 7 Working memory can be seen as a “goal stack”. A goal stack means that behaviour involves making plans, which in turn involves breaking plans down into goals and sub-goals. Goal stacks are a rational adaptation to a world that is structured and in which we can identify causes and effects.
- 8 This sense of rationality implies:
- Our ways of thinking are the product of evolutionary processes.
 - As far as we can, we use our knowledge in pursuit of our goals.
 - Our thinking is likely to be biased by the knowledge we have available.
 - Our ability to pursue our goals is constrained by the limited capacity of our information processing system.

References

- Anderson, J.R. (1993). *Rules of the Mind*. Hillsdale, NJ: Erlbaum.
- Anderson, J.R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum.
- Atkinson, R.C., & Shiffrin, R.M. (1968). Human memory: A proposed system and its control processes. In K.W. Spence & I.T. Spence (Eds.), *The Psychology of Learning and Motivation* (Vol. 2, pp. 89–195). London: Academic Press.
- Bransford, J.D., Barclay, J.R., & Franks, J.J. (1972). Sentence memory: A constructive versus interpretative approach. *Cognitive Psychology*, 3, 193–209.
- Brewer, W.F., & Treyens, J.C. (1981). Role of schemata in memory for places. *Cognitive Psychology*, 13, 207–230.

54 Problem representation

- Byrne, M.D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21(1), 31–61.
- Dennett, D.C. (1996). *Darwin's Dangerous Idea*. London: Penguin.
- Eisenstadt, M. (1988). *PD622 Intensive Prolog*. Milton Keynes: Open University Press.
- Ericsson, K.A., & Hastie, R. (1998). Contemporary approaches to the study of thinking and problem solving. In R.J. Sternberg (Ed.), *Thinking and Problem Solving* (2nd revised ed., pp. 37–82). New York: Academic Press.
- Ericsson, K.A., & Simon, H.A. (1993). *Protocol Analysis: Verbal Reports as Data. Revised Edition*. Cambridge, MA: MIT Press.
- Gigerenzer, G. (2015). *Simply Rational: Decision Making in the Real World*. Oxford: Oxford University Press.
- Gigerenzer, G., & Todd, P.M. (Eds.). (1999). *Simple Heuristics That Make Us Smart*. Oxford: Oxford University Press.
- Glaser, R. (1984). The role of knowledge. *American Psychologist*, 39(2), 93–104.
- Greeno, J.G. (1974). Hobbits and Orcs: Acquisition of a sequential concept. *Cognitive Psychology*, 6, 270–292.
- Hayes, J.R. (1989). *The Complete Problem Solver* (2nd ed.). Hillsdale, NJ: Erlbaum.
- Hodgetts, H.M., & Jones, D.M. (2006). Interruption of the Tower of London task: Support for a goal-activation approach. *Journal of Experimental Psychology: General*, 135(1), 103–115. doi:10.1037/0096-3445.135.1.103
- Hunt, E. (1998). Problem solving. In R.J. Sternberg (Ed.), *Thinking and Problem Solving* (2nd revised ed., pp. 215–234). New York: Academic Press.
- Kahney, H. (1993). *Problem Solving: Current Issues* (2nd ed.). Milton Keynes: Open University Press.
- Laird, J.E., & Rosenbloom, P.S. (1996). The evolution of the SOAR cognitive architecture. In D. Steier & T.M. Mitchell (Eds.), *Mind Matters: A Tribute to Allen Newell* (pp. 1–50). Mahwah, NJ: Erlbaum.
- Loftus, E.F. (1996). *Eye witness Testimony*. Cambridge, MA: Harvard University Press.
- Markman, A.B. (1997). Constraints on analogical inference. *Cognitive Science*, 21(4), 373–418.
- Markman, A.B. (2006). *Knowledge Representation*, *Psychology of Encyclopedia of Cognitive Science*: John Wiley & Sons.
- Nathan, M.J., Kintsch, W., & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition and Instruction*, 9(4), 329–389.
- Newell, A. (1980). Physical symbol systems. *Cognitive Science*, 4, 135–183.
- Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Newell, A., & Simon, H.A. (1972). *Human Problem Solving*. Upper Saddle River, NJ: Prentice Hall.
- Reed, S.K., Ernst, G.W., & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology*, 6, 436–450.
- Simon, H.A. (1973). The structure of ill-structured problems. *Artificial Intelligence*, 4, 181–201.
- Simon, H.A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology*, 7, 269–288.
- Simon, H.A., & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2), 145–159. doi:10.1037/h0030806
- Solso, R.L. (1995). *Cognitive Psychology* (4th ed.). London: Allyn and Bacon.
- Sternberg, R.J. (1996). *Cognitive Psychology*. Orlando, FL: Harcourt Brace.
- Thomas, J.C. (1974). An analysis of behavior in the Hobbits-Orcs program. *Cognitive Psychology*(6), 257–269.
- Todd, P.M., & Gigerenzer, G. (2000). Précis of simple heuristics that make us smart. *Behavioral and Brain Sciences*, 23, 727–780.
- VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M.I. Posner (Ed.), *Foundations of Cognitive Science*, pp. 527–79. Cambridge, MA: MIT Press.