Chapter 5

Cognition

5.1 Introduction

In cognitive science, rationality in scientific discovery itself is being studied as an interesting cognitive phenomenon. One popular view is taking scientific discovery as just a form of human problem solving (Langley et al. 1987). One of the most successful theories about human problem solving is developed by John R. Anderson (Anderson 1993, Anderson & Lebiere 1998). It is called ACT-R, meaning Adaptive Control of Thought – Rational. The ACT-R theory deals with the cognitive mechanisms of learning and rational behavior. It aims to explain how people make an assumption or take an action to observe or change something in the world, in such a way that the probability to achieve a specific goal is high and the cost of time to achieve it is low. ACT-R is implemented in a computer program to test the performance of specific models of problem solving strategies.

The general question of this chapter is: what is rationality in scientific discovery, according to the psychological study of cognition? As a general model of human cognitive abilities, ACT-R should also be able to model specific cognitive processes involved in scientific problem solving. In this chapter I investigate how it could do that. The particular question that is answered in this chapter is: how can one understand and model scientific discovery with ACT-R?

I will first, in section 5.2, introduce a distinction between primary and secondary epistemology. Analogously to these types I make a distinction between primary (or native) and secondary (or acquired) processes of cognition. I will use this distinction to discuss how beliefs, goals and search methods are created, selected and evaluated according to the ACT-R theory in section 5.3 to 5.5. In section 5.6, I discuss how scientific discovery, as modeled in Simon and Langley's BACON.1 (Langley et al 1987) and Thagard's PI (Thagard 1988) can both be modeled in ACT-R as similar forms of abductive inference. I demonstrate and discuss how ACT-R's primary mechanisms nicely subsume PI's hypothesis evaluation process. Then, I discuss BACON.1's search methods and how they can be learned by analogy from examples. In section 5.7 I discuss the nature of theory and method in the different models. 5.8 discusses the difference between the logical and psychological views on explanation and prediction. I end this chapter in section 5.9 with a discussion and general conclusion, answering the specific questions from section 1.3.

5.2 Primary and secondary

The claim that philosophy of science can learn something from cognitive psychology is endorsed by the philosopher Alvin Goldman. He argues that epistemology, the study of justified belief, should take explicit account of empirical studies of cognitive processes (Goldman 1986). Among the many factors that influence the forming of belief he distinguishes basic cognitive processes from acquired belief forming methods.

The first category, *basic processes*, include processes of perception, memory, attention, concept formation, problem solving, learning and reasoning. Goldman argues that these natural or native processes are suitable objects for normative epistemic evaluation, and comprise the domain of *primary* epistemology. *Secondary* epistemology comprises the normative evaluation of acquired belief forming *methods* like algorithms, techniques or procedures. A method can either be a general, topic neutral, or a task specific procedure for arriving at beliefs.

In forming a belief, basic processes and methods are intrinsically intertwined. When someone needs to solve a problem and several methods are available, the basic processes determine which method is applied, and also which new methods are created or added. So evaluating a resulting new belief depends on the reliability of both the basic processes and the specific applied method.

So in short, primary epistemology is concerned with the evaluation of basic, *i.e.* native or natural, cognitive processes, and secondary epistemology is concerned with the correctness of acquired belief forming methods. To explain how such processes and methods are explicated in the ACT-R theory, I will first use Goldman's distinction to differentiate between two general types of cognition, *i.e.* primary and secondary cognition.

By *primary cognition* I mean native or basic cognitive processes and structures, whereas by *secondary cognition* I mean acquired cognitive processes and structures. In this way we can also distinguish acquired *structures*, like beliefs and goals, from basic structures, like the memory activation values used by basic or primary cognitive processes in ACT-R.

5.3 Declarations and procedures

Anderson's ACT-R explains human (problem solving) behavior as the result of acting according to two types of knowledge: declarative and procedural knowledge (Anderson 1993). Declarative knowledge consists of declarations of beliefs and goals, and resides in a person's declarative memory. Procedural knowledge consists of procedures that can create and modify a persons beliefs and goals. It contains our cognitive skills, or our *know how*. In ACT-R declarative knowledge is represented as a collection of memory structures called *chunks*. A chunk is an abstract representation of a belief or goal structure. Its basic elements consists of a list with slots and slot values. For example:

```
(Johannes_Kepler
      ISA                  person
      BORN                 "27 December 1571"
      PROFESSION           scholar
      ACHIEVED             "discovery laws of planetary motion"
      FEARED-MOST          "invasion by the Turks"
      ETC                  ...)
```

The ISA ('is a') slot value represents the type of the chunk, and can be seen as a concept type name. Every concept type has the same slot-names, or concept attributes. So in our example, a person is something with a date of birth, a profession, etc. A slot value can in its turn also be a chunk. In this way declarative knowledge is structured in a network of memory chunks. In our example:

```
   (scholar
      ISA                  profession
      ACTIVITY             research
      ETC                  ...)
```

Procedural knowledge, or know how, is represented by production rules, or productions for short. Such a rule consists of a set of conditions and actions. The conditions, or left hand side (LHS), of a production can match with memory chunks which satisfy given constraints. When a matching succeeds, certain actions can be performed which are specified in the action, or right hand side (RHS), of a production. For example:

```
   (SUBTRACT
      =goal>
         ISA subtract
         VAR1              =x
         VAR2              =y
         ANSWER            nil
      =addition-fact>
         ISA addition-fact
         ADDEND1           =y
         ADDEND2           =z
         SUM =x
   ==>
      =goal>
         ANSWER            =z)
```

This production uses declarative knowledge of an addition fact to find the answer for a subtraction problem. A string with an '=' sign is a variable that is bound to a value by matching a chunk. The LHS, before the arrow, matches against any goal of which no answer is known and a fact (an addition fact in the example) that satisfies the values =x and =y of the goal slots. In the RHS, after the arrow, the found value =z of the addition fact is added to the ANSWER slot of the subtract goal.

In summary, this is what ACT-R poses that human problem solving is all about: matching productions (skills) to memory chunks (beliefs and desires). We can say that the chunks and productions themselves, constitute secondary cognition. A memory chunk is an acquired structure, a production is an acquired process. However, the processes that ACT-R really is about are the (native) mechanisms about *how* and *what* memory chunks and productions are used in problem solving.

In human problem solving often several (possibly mutually inconsistent) belief chunks can match a production's LHS. And for a given problem goal more than one production may apply. The ways the cognitive mechanism efficiently evaluate alternative chunks and productions constitute the main aspects of primary cognition.

5.4 Structures and processes

In Table 5.1, I summarize the main cognitive mechanisms according to the ACT-R theory, explicating their primary processes and structures. In the process of problem solving, (secondary) knowledge, containing of chunks and productions, is created, selected and evaluated by (primary) learning mechanisms. (This section discusses the ACT-R architecture up to version 3, primarily based on Anderson (1993).)

Cognitive mechanisms	Primary processes	Primary structures
Creation of chunks by:		
Concept-formation	(Specifying chunk types)	(Basic types?)
Perception	Specifying (new) chunks	(Constraints?)
Productions (RHS)	Specifying RHS chunks	-
Selection of chunks by:		
Productions (LHS)	Matching LHS chunks	-
Goal focus	Goal stack control	-
Activation	Preferring high $A_i=B_i+S_jW_jS_{ji}$	Value A_i
Base-level activation	Computing & learning B_i	Value B_i
Salience strength of j to I	Computing & learning S_{ji}	Value S_{ji}
Association of i with j	Computing W_j	Value W_j
Evaluation of chunks by:		
Activation	Preferring highest A_i	Value A_i
Creation of productions by:		
Analogy	Generalizing example chunks	Special slots
Selection of productions by:		
Goal focus	Matching LHS to goal focus	-
Chunks	Matching LHS to chunks	-
Matching time (latency)	Preferring low $T_p = S_ie^{-}(A_i+ S_p)$	Value T_p
LHS chunks activation	Computing & learning A_i	Value A_i
Strength of production	Computing & learning S_p	Value S_p
Eval. of productions by:		
Expected gain	Preferring high value $E = PG – C$	Value E
Probability of success	Computing $P = qr$	Value P
Prob. of intended effect	Computing & learning *q*	Value *q*
Prob. of suc. after firing	Computing & learning *r*	Value *r*
Value of the goal	Specifying value G	Value G
Cost of production	Computing $C = a + b$	Value C
Cost of firing production	Computing & learning *a*	Value *a*
Cost of actions after firing	Computing & learning *b*	Value *b*

Table 5.1: Primary aspects of ACT-R's cognitive mechanisms

Table 5.1 summarizes the primary aspects of ACT-R's cognitive mechanisms (version 2.0). In the first column I list different kinds of primary cognitive mechanisms. These essentially control the creation, modification, selection and evaluation of secondary cognitive structures (memory chunks) and processes (productions). The primary cognitive mechanisms consist of primary processes (column 2), guided by, and modifying primary structures (column 3). I will discuss them briefly in the following subsections.

Creation

In the ACT-R theory, memory is ordered by *types* of memory chunks. A concept like 'person' in the example above, is supposed to have a given template of attributes. Every instantiation of a concept shares the same attribute slot names, but differs in their values. If you want to add something to memory, a concept type is necessary. But how do concepts come about in ACT-R?

In any cognitive creation or modification process we can make a distinction between the process that actually makes the creation or modification and that *what* is created or modified. In connectionist theories of cognition we often see that both are the same, that the concept creation process 'decides' on the concept types 'on the run'. In ACT-R there is no primary process specified that creates types, and the theory is silent about what types there should be. The modeler has to define them up front. Chunk types can also not be created or changed by learned productions, while chunk type instantiations can. So it is not clear whether we can consider concept types as primary or secondary structures, and if there are any basic constraints, or even basic or native types (like Jerry Fodor suggests).

The process of *perception* can add new chunks to memory. Again we can say that in ACT-R the process of adding them is a primary process. Yet how perception is constrained by concept types, or guided by problem solving is not defined in ACT-R, but in the perceptual/motor extension of the theory ACT-R/PM. I will not go into this extension here (see Anderson & Lebiere 1998).

Finally *productions* can add and modify memory chunks. That is what ACT-R is (mostly) all about, how and which productions modify and add chunks to memory. Once a chunk is added it will never be deleted. Its worst fate is never to be recalled. How, and what chunks are recalled is governed by processes of chunk selection. Productions themselves, as representations of learned skills, can only be created and added by a primary process of analogy. To connect actions to conditions, ACT-R starts out with a declaration of a problem example and its solution. When another problem of the same type is encountered, analogy will generalize a solution strategy from the known example. How that process works is discussed in the next section.

Selection

In a process of problem solving the selection of relevant chunks and productions is constrained in several ways. The main guiding mechanism of problem solving in ACT-R is *goal focus*. Goal focus is a kind of pointer to a chunk saying, "this chunk represents the goal I want to achieve", which in ACT-R means "that is the chunk a production should match with". ACT-R does not say how goal focus is initially specified. How a person is motivated to desire the accomplishment of a goal, however, is determined rationally in ACT-R. After setting the first goal, several primary

and secondary processes influence how to achieve that particular goal by specifying and focusing on subgoals. The action, or right hand side (RHS) of a production can shift focus to another goal, which is implemented by a *push* of a new goal on *stack*. When a production has achieved the new goal, it can *pop* it from the stack, thereby changing focus to the next goal below it on the stack.
When an initial goal is set, ACT-R first selects a set of potential productions that can match with it. For a production to match, the given goal must be the first part of the production's condition or left hand side (LHS). An LHS usually contains other chunks which should match as well, given specified constraints, and need to be retrieved from memory.

ACT-R also models *latency*, which is the time it takes to match a production to memory and perform the action. How long that takes depends on the activation of the chunks needed. The latencies in the model should reflect the latencies in reaction time of subjects, measured in psychological experiments.

Activation is a basic property of every chunk. A chunk's activation value is the result of its prior base level activation plus the contribution of chunks that are part of the current goal context. This value increases with use. A primary learning process increases the association between two chunks every time they are both needed to solve a problem. According to Anderson, a chunk's activation denotes its posterior (logarithmic) odds that it will be needed in a given context, and the learning process is supposed to give the best estimate of that chance. When a chunk is not used its activation decays logarithmically. When it drops below a certain threshold, it can no longer be retrieved in the current context. Another context might however contain the right cues to boost the activation above the threshold again, re-enabling retrieval. Next to chunk activation, a production's strength also controls production selection. A production's strength increases after use, and is learned accordingly. Again its strength denotes its (logarithmic) odds of being needed.

So in sum, when focus is set to a goal, primary processes in ACT-R start to select productions that can match with it. A set of alternatives is gradually selected, depending on the activation of chunks in the productions' LHS, and the strength of the productions.

Evaluation

When several chunks can match a production's LHS, the chunks with the highest activation will be used. However, that is not the case for productions. Next to the time it takes to retrieve relevant productions, other primary evaluation processes contribute to determine what production will determine the next action.

During selection, potential productions are evaluated simultaneously by a primary process of rational analysis. This process diagnoses whether a given production is worth it to be fired. In order to do so it takes three estimations into account: the probability the production will be successful ($P = qr$); the value of the goal that is desired (G); and the cost of firing that production ($C = a + b$). A production's probability of success is a product of the probability of its intended effect (q) and the probability of achieving the goal after having achieved intended effort (r). The cost of a production is the result of adding the cost of the cognitive effort to fire the production (a) with the cost of actions needed to reach the goal after firing the production (b).

For example, if your goal is to lessen your thirst, and you are in front of a coffee machine, a production may be evaluated that urges to throw a coin in the machine to get a cup of coffee. Now *q* is the estimation that the machine will indeed return a cup, and *r* is the chance that only one cup will quench your thirst. The quantity *a* denotes the effort of putting in a coin, while *b* stands for the effort of emptying the cup. The quantities *q* and *a* can be estimated by repeated applications of the production. For example, if the machine is old and failed a number of times in the past, *q* will be low.

The quantities *b* and *r* are more difficult to estimate because they may refer to yet unknown actions. Anderson's solution is to base their estimates on how much the state achieved by the production differs from the desired goal. If the action of putting in a coin fails to provide you with a coffee, it is less likely that you will quench your thirst (*r'*) and more effort will be needed to get a drink (*b'*). And in general the more effort already spent, the less likely you will achieve your goal at all, so the lower the probability (*r'*).

The production with the highest estimated gain E (= PG-C) of the selected productions is generally preferred. In this way when the value of a goal or the probability of its success is high, the cost of a production plays a less important role. When you know the coffee machine often fails and is situated on another floor of the building, the cost of walking to it may not be worth one's while. But when you are really thirsty the cost loses out to the value of the goal. The best production rule given its PG-C is not always selected, but it has the highest chance of being fired.

When a production finally fires, its RHS or action side will be executed, changing beliefs or goals, or initiating hand an eye movement, like looking for the slit on the coffee machine and putting a coin in it. After firing, a new (sub)goal may be set by the production or from the goal stack, and the process of selecting, evaluating and firing a production starts all over. ACT-R stops when the initial goal is achieved and popped from the goal stack.

5.5 BACON and PI

In this section I discuss two computational models of scientific discovery, and how the structures and processes of these models can be modeled in ACT-R. Typical scientific problems are searching and evaluating descriptions and explanations for interesting observations. Herbert Simon and Paul Thagard proposed different explanations about how scientists (could) solve those tasks. They both modeled their theory in computer programs, respectively called BACON and PI.

The first of the BACON programs models the search for simple quantitative laws that describe the numerical data of observations, like Kepler's third law of planetary motion and Boyle's gas law. PI searches and evaluates qualitative explanations, like the explanation of the propagation of sound from its being a wave.

In PI, new hypotheses are searched and evaluated through a primary process of abduction and inference to the best explanation (IBE). In this section I will argue that abduction is better thought of as a secondary *acquired* process in ACT-R, generalized from examples by analogy, while IBE is subsumed by ACT-R's primary processes. I will further demonstrate that the heuristic search method for laws as implemented in BACON.1 can also be learned from examples.

Simple abduction in PI

Paul Thagard's theory of cognitive inductive processes, modeled in PI (processes of induction), includes several forms of abduction. I will consider its simplest form. Abduction, as discussed in Chapter 4, is a form of inductive inference. It is inductive in the sense that the truth of the conclusion of the abductive inference does not follow from the truth of the premises. As stated in Chapter 4, Peirce defined abduction as follows:

(P_1) "The surprising fact, C, is observed;
(P_2) But if A were true, C would be a matter of course.

(C) Hence, there is a reason to suspect that A is true."

In Peirce's original definition the selection and evaluation of explanation A is all part and parcel of the same inference. But usually not only the truth of A would make C a matter of course. Say B could also lead to the truth of C. So clearly Peirce's definition is not enough for an inference to the *best* explanation. Thagard made a clear distinction between the inference of possible explanations for surprising facts, and their evaluation. Peirce's original definition of abduction is a clear form of inferring from P_2 a possible explanation for P_1. But before jumping to conclusion C, other known premises like P_2 should be considered first.

Thagard defined a separate process to evaluate the resulting set of possible explanations, and called that process inference to the best explanation (IBE). Thagard defined IBE as an inference to a known explanation which explains the highest number of other known facts, needing the lowest number of auxiliary hypotheses as background assumptions. An explanation's value can be calculated by subtracting the number of auxiliary hypotheses from the number of explained facts. In that way, adding an explained fact *ad hoc* by an auxiliary hypothesis makes no difference for an explanation's value.

In PI, abduction and IBE are modeled as a process of problem solving. An explanation problem is represented by a basic memory structure, including the slot `START` containing context facts, and the slot `GOAL`, containing the explananda, the facts to be explained. Theories are represented as (secondary) processes called *rules*, with slots `CONDITION`, which contain premises and `ACTION`, containing conclusions. When a problem is set, a primary process of spreading activation activates rules linked to the problem slots. Only active rules are used to infer possible explanations for the slot value of `GOAL`. IBE decides which explanation is the most favorable. For example, we have three possible explanations of an observation E, represented in three rules. Activation from E activates the rules, which generate possible explanations by abduction. IBE selects the best as a conclusion of solving the explanation problem, see Table 5.2.

In PI, rules, problems and concepts all have basic structure types. Among the basic slots are `ACTIVATION`, `STRENGTH`, and `OLD-MATCHES`. The processes of activation, abduction and IBE are all primary. Only instances of concepts, rules and problems are secondary. IBE in PI is a process specially used for making evaluations of abductions, which only occur during explanation problems.

Explanation	Structure	Process	Example
	EXPLANATION		
Premise	START		F (is known to be true)
	GOAL		E (is to be explained)
Background		RULE-1	CONDITION H1 ACTION E
		RULE-2	CONDITION H2 H3 ACTION E
		RULE-3	CONDITION H4 ACTION E F
Inference		Activation	(E activates rules 1 to 3)
		Abduction	H1, H2&H3, H4 (possible explanations)
		IBE	H4 (explains the most facts with the least auxiliary hypotheses)
Conclusion			H4 (is the best explanation)

Table 5.2: Explanation as modeled in the PI program

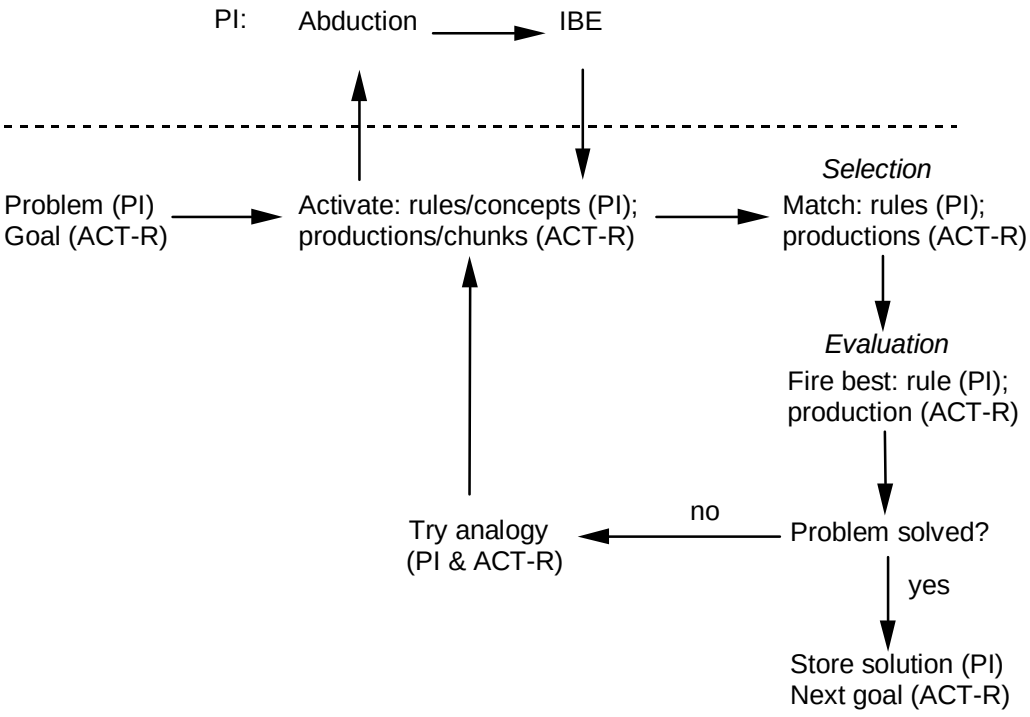


Figure 5.1: Problem solving in PI and ACT-R

Problem solving in ACT-R is similar to that of PI (see Figure 5.1), but with a few important differences. In both PI and ACT-R memory structures match with rules, which can add to memory and influence problem solving control. Yet productions in

ACT-R are of a different type than PI's rules. They represent a skill, and not an explanatory relation. And, more important for modeling explanation, ACT-R lacks a primary abduction mechanism. Because of the nature of the ACT-R theory such a primary mechanism is not appropriate. Productions in ACT-R are steps of practiced problem solving, generalized from example problem solutions by analogy (PI can also employ analogy to suggest rules, but I will not go into that here). So if a cognitive model in ACT-R needs to employ abduction in problem solving, then the abduction inference rule has to be learned first. And that turns out to be no problem at all.

Learning abduction by example - part 1

The ACT-R theory assumes that part of the process of solving a particular problem, is trying to recall an example of a problem that was solved earlier and had a goal similar to the current problem. When such an example problem is retrieved from memory, the structure of that example problem, and the solution of that example problem, is mapped to the current problem. When the solution of the example problem can be used to solve the current problem, a production rule is proposed, as a generalization of a strategy for solving problems that share the particular goal. It is currently assumed that all procedural skills, represented by production rules, are learned by this process of generalization from declarative examples.

The discussion and models in this section are based on the analogy mechanism of ACT-R, release 3.0. The details of implementing the mechanism of analogy have been changed in the 4.0 version that was introduced after I wrote this chapter. Learning by examples in ACT-R is studied extensively by Niels Taatgen (1999).

In this subsection I model an example of Paul Thagard's from his (Thagard 1988). He tells about his encounter with a group of outrageously dressed persons at the airport. He wonders why theses people are dressed up that way. Maybe they are rock musicians, he thinks, because rock musicians usually dress outrageously. ACT-R has to know only this example to generate, by analogy, a production that can make similar abductive inferences in the future.

As a similar explanation problem I use another example from (Thagard 1988). In this simple historical example the goal is to explain why sound propagates. It is known that waves propagate, so maybe sound is a wave. I started out with the following memory chunks:

```
(Example-Problem
    ISA               explanation-problem
    GOAL              Dressed-Outrageously)
(Example-Rule
    ISA               pi-rule
    CONDITION         Rock-Musician
    ACTION            Dressed-Outrageously)
(Example-Solution
    ISA               explanation-solution
    EXPLANATION       Rock-Musician)

(Example-Dependency
    ISA               dependency
    GOAL              Example-Problem
    SUBGOALS          Example-Solution
    CONSTRAINTS       (Example-rule))
```

```
(Problem-1
   ISA               explanation-problem
   START             sound
   GOAL              Propagates)
(Rule-1
   ISA               pi-rule
   CONDITION         wave
   ACTION            Propagates)
```

The example-dependency chunk is used (In AC-R 3.0) to represent the link between a problem chunk and the chunk that represents the solution to that problem. The constraint slot is used to represent that additional chunks that where involved in solving the problem.

The slot values `Rock-Musician`, `Dressed-outrageously`, `Propagates`, and `Wave` are also added as memory chunks of type `concept`. This chunk type also has a slot `INSTANCES`, which is filled with `Sound` for concept `Propagates`. The goal focus is set on `Problem-1`, which represents the problem to explain why sound propagates.

When ACT-R is started it first tries to match the goal `Problem-1` with available productions. After failing to do so (there are none defined) ACT-R searches for an analogous problem and finds `Example-Problem`. The special dependency chunk is used to find its solution. ACT-R uses the `Example-Rule` to map the solution to the problem, and uses it to make a new production. It then tests whether the new production will match the focused goal. Only if that succeeds will the new production be added to production memory. In my example ACT-R produces the following production:

```
(EXPLANATION-PROBLEM-PRODUCTION0
  =Example-Problem-Variable>
    ISA                    explanation-problem
    GOAL                   =dressed-outrageously-variable
  =Example-Rule-Variable>
    ISA                    rule
    CONDITION              =rock-musician-variable
    ACTION                 =dressed-outrageously-variable
==>
  =Example-Solution-Variable>
    ISA                    explanation-solution
    EXPLANATION            =rock-musician-variable
  !focus-on! =Example-Solution-Variable)
```

The first condition chunk matches with `Problem-1` and the second with `Rule-1`. As a result the production creates a solution and changes focus of attention to it. This rule now serves as a secondary simple abduction process, generating hypothetical explanations, given explanation problems and rules that may explain it. The resulting explanation for the example is:

```
(**Example-Solution-Variable$1>
    ISA                    explanation-solution
    EXPLANATION            Wave)
```

This example has only one rule to abduce from. Usually several rules can be used to generate an explanation. Thagard employed IBE in PI to evaluate possible explanations before jumping to a best conclusion.

It can be argued that the general idea of Thagard's IBE is subsumed by ACT-R's primary, processes that subsymbolically select and decide which chunks and productions to match. Thagard's IBE favors the hypothesis that explains most known facts with the least number of auxiliary hypotheses. So there is a constraint on explanatory success and hypothesis simplicity. The simplicity constraint is met by ACT-R's primary process of latency, which is related to the probabilistic evaluation whether a chunk is relevant in a particular context. A more complex rule will contain more chunks in the condition, which will take longer to match. So more simple hypotheses will be considered first. Yet a very successful rule will have a higher activation because it is associated with more active facts in memory. So the constraint on explanatory success, is met by the process of preferring high activation.

One could compare the effect of the activation of chunks as a result of their probabilistic association with other chunks in ACT-R, with the effect of the activation of propositions as a result of their explanatory relation with other propositions in ECHO, Thagard's refined explanation evaluation model (Thagard, 1992).

Yet, several other factors, such as the production's expected gain (PG-C) value, play a role in the final decision to fire a rule. Hence ACT-R might not always come to similar conclusions as PI. Whether ACT-R's conclusions are more plausible is another question altogether, belonging to primary epistemology. However, because of the fact that ACT-R is a more sophisticated model of primary cognition than PI is, ACT-R is likelier to make abductive inferences that are closer to actual human problem solving. Whether that is relevant for epistemology is discussed in section 5.8.

Heuristic search as abduction in BACON

The BACON models (Langley et al, 1987) constitute a set of productions that try to find algebraic laws that describe given numerical observations. Several versions of BACON were originally implemented as a set of productions in the problem solving architecture PRISM, an old cousin of ACT-R (they both have ACTE as an ancestor). Hence it was relatively easy to model BACON.1 in ACT-R. Yet, a distinguishing claim of the ACT-R theory is that productions are not learned passively by *e.g.* reading, but by analogy during problem solving, by doing. Therefore I tried to model learning BACON's main productions by analogy. Doing so made apparent that in fact BACON's heuristic search method makes use of abductive inference in a way similar to PI's method.

The first of the BACON series searches for simple algebraic laws, which are all of the form $X^kY^l=aX^mY^n+b$. It tries to find appropriate values for *k, l, m, n, a* and *b* given a set of different observed values for X and Y. Laws that fit this template are, for example, Kepler's third law of planetary motion $D^3P^2=k$, Boyle's gas law $PV=c$, Galilei's law of acceleration $D/T^2=g$, and Ohm's law $IL=-rI+v$.

BACON.1's search starts out with two observational terms X and Y, together with a set of values. For example, X is (1 2 4) and Y is (1 0,5 0,25), meaning that when X is 1 Y is 1, etc. The next step is to combine two terms as a product or a ratio and evaluate the resulting set of values, *e.g.* X*Y is (1 1 1). When the values of a term are found to be constant, a law is inferred. In the example $X*Y=c$. The same happens when two terms are related linearly. If the new term does not turn out to have constant values, or to be linearly related with other terms, then it can be used to make a next new term by combining it with the other available terms, *e.g.* (X*Y)*Y.

The BACON productions do not produce new terms at random, but *heuristically*. A heuristic method does not guarantee that a solution will be found, but often a solution can be found without evaluating every possible solution by brute force search. BACON.1's heuristic term generation is implemented in productions called Increasing and Decreasing. These productions determine what new term to consider as a possible law. Given that the absolute values of two terms both increase Increasing suggests to consider their ratio as a new term. Decreasing suggests to consider the product of two term when the absolute values of one terms decrease while the absolute values of the other increase.

These productions, together with the main productions that implement the search process are listed in Table 5.3. The search process itself is depicted in Figure 5.2, and summarized in Table 5.4. As an example, I listed the terms used and defined in the process of finding Kepler's third law of planetary motion in Table 5.5, based on Borelli's observations of the moons of Jupiter that were discovered by Galileo.

Production	Conditions (LHS)	Actions (RHS)
Find-Laws	Goal = describe data Law not already defined?	New goal = find-laws
Increasing	Goal = find-laws Term-1 increasing values? Term-2 increasing values?	New goal = consider-ratio
Decreasing	Goal = find-laws Term-1 increasing values? Term-2 decreasing values?	New goal = consider-product
Constant	Goal = find-laws Term constant values?	New goal = define-new-law
Linear	Goal = find-laws Term values linear related?	New goal = define-new-law
Define-Ratio-or-Product	Goal = consider-ratio/product Term not already defined?	New goal = define-new-term

Table 5.3: Overview of the main productions of BACON.1

ACT-R can learn the productions Increasing and Decreasing from given examples. The examples I used constituted algebraic rules that can be used abductively by ACT-R's process of analogy. For example it is true for the function X/Y=*c*, that if the absolute values of X increase, the absolute values of Y increase as well. On the other hand it is true for the function X*Y=*c*, that if the absolute values of X increase, the absolute values of Y decrease (see Figure 5.3).

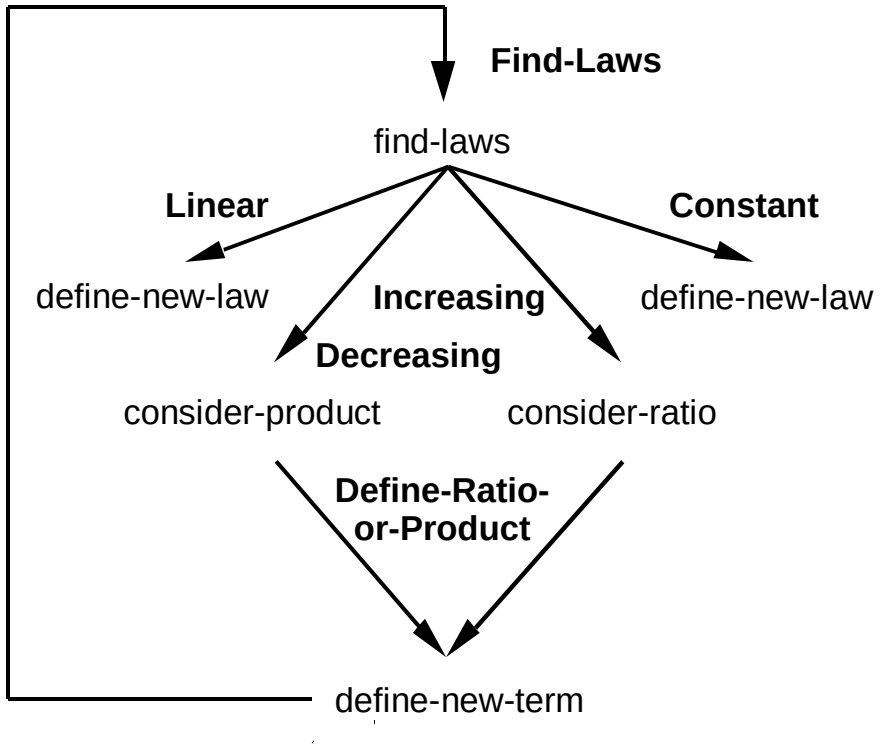


Figure 5.2: BACON.1's search for a law with constant or linearly related values

Description	Structure	Process	Example
Premise	X		1 4 9
	Y		1 8 27
	Goal		Describe X and Y
Background		Production-1	Find-Laws
		Production-2	Increasing
		Production-3	Decreasing
		Production-4	Constant
		Production-5	Linear
		Production-6	Define-Ratio-or-Product
Inference		Repeated matching of production rules	
Conclusion	Law X Y		$X^3/Y^2=1$

Table 5.4: Inferring a description in BACON.1

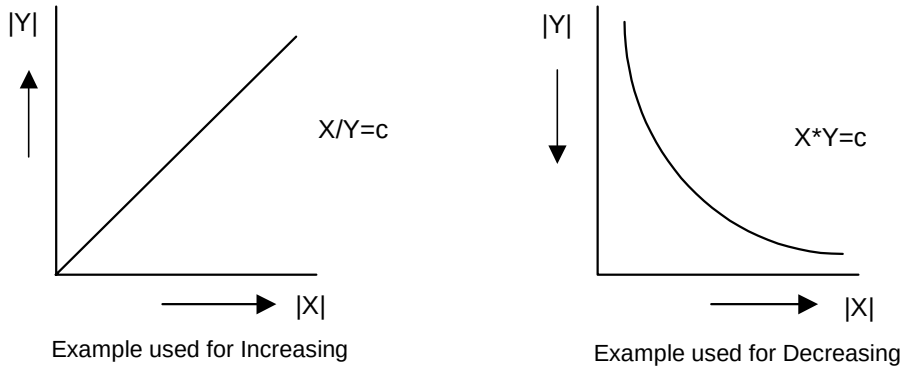


Figure 5.3: Example functions used for creating the main BACON productions

The other way does not always hold. For example, when both values of an X and Y increase, the relation may just as well be an exponential function. Hence BACON.1's productions actually infer by abduction that X and Y are related as a product or ratio. Increasing employs actually the following abductive inference:

> The absolute values of X and Y both increase (C)
> But if X/Y = c (A) then the absolute values of X and Y would increase (C)
> ---
> Hence there is a reason to suspect that X/Y = *c* (A)

If the inferred new term is not evaluated to be a law, like *e.g.* D/P in the Kepler example, then values of the term can be treated as part of the background in a new abductive inference. The same compositional process is used in PI, see for example Table 5.6.

Explanation	Structure	Process	Example
Premise	PROBLEM		
	START		A (is known to be true)
	GOAL		C (is to be explained)
Background		RULE-1	CONDITION B ACTION C
		RULE-2	CONDITION A ACTION B
Inference		Activation	(C activates RULE-1)
		Abduction	B (possible explanation)
		Activation	(B activates RULE-2)
		Abduction	A (possible explanation)
		IBE	A
Conclusion	EXPLANATION		A (is the best explanation)

Table 5.6: Example of compositional abduction in PI

Learning abduction by example - part 2

To learn ACT-R BACON's heuristics I provide the functions of Figure 5.2. as solutions to a BACON search problem. The example for Decreasing was given as follows:

```
(X1>
  ISA                          term
  PATTERN                      Increasing
  EXP                          Example-Experiment)

(Y1>
  ISA                          term
  PATTERN                      Decreasing
  EXP                          Example-Experiment)

(Example-Problem1>
  ISA                          find-laws
  EXP                          Example-Exp
  ACHIEVED-BY                  Consider1)

(Example-Solution1>
  ISA                          consider
  OP                           Product
  TERM-1                       X1
  TERM-2                       Y1
  CONSTRAINTS                  (Decreasing Increasing))

(X1*Y1>
  ISA                          term
  VALUES                       nil
  OP                           Product
  TERM-2                       Y1
  TERM-1                       X1
  PATTERN                      Constant)
```

When the product of X and Y is constant, the values of X increase while the values of Y decrease. So if you want to find a law for the terms of an experiment Example-Exp, which are X1 and Y1, then by abduction BACON should consider their product. The production Consider-Ratio-or-Product would then define the term X1*Y1. To trigger ACT-R's analogy mechanism I set the following problem:

```
(Pressure>
  ISA                          term
  PATTERN                      Increasing
  EXP                          Boyle-Exp)

(Volume>
  ISA                          term
  PATTERN                      Decreasing
  EXP                          Boyle-Exp)

(**Boyle>
  ISA                          find-laws
  ACHIEVED-BY                  nil
  EXP                          Boyle-Exp)
```

By analogy with `Example-Problem-1`, using chunk `example-experiment` to map the solution to the problem, ACT-R composes a new production to solve the Boyle problem:

```
(FIND-LAWS-PRODUCTION1
  =Example-Problem-Variable>
    ISA                             find-laws
    EXP                             =example-exp-variable
  =Y1-Variable>
    ISA                             term
    EXP                             =example-exp-variable
    PATTERN                         decreasing
  =X1-Variable>
    ISA                             term
    EXP                             =example-exp-variable
    PATTERN                         increasing
==>
  =Example-Solution-Variable>
    ISA                             consider
    OP                              product
    TERM-1                          =X1-Variable
    TERM-2                          =Y1-Variable
  !focus-on! =Example-Solution-Variable)

(**Example-Solution$1>
    ISA                             consider
    OP                              product
    TERM-2                          Volume
    TERM-1                          Pressure)
```

The analogy mechanism of ACT-R (3.0) would overgeneralize the example problem without further constraints. The resulting production would match any two terms and consider their product. Yet with constraints, the inferred rule is functionally equivalent with BACON's original Decreasing production, and can hence be employed to find more complex laws.

In sum, the computer programs BACON and PI model cognitive mechanisms of scientists that work on particular scientific problems. In this section I argued and showed how those same mechanisms can be learned and explained, by modeling that learning process in the unified cognitive theory ACT-R. Yet between the different cognitive architectures there remains a difference in approach to understanding the nature of scientific theory and reasoning. This is treated in the next section.

5.6 Theory and method

In this section I go further into the specific questions about the structure of theory and process of reasoning as implied by the different cognitive models BACON, PI, and ACT-R. Thagard's model PI maintains a procedural explanation of the nature of a theory. In the logical approach a theory is a set of atomic and conditional propositions, accompanied by a set of relatively independent inference rules that are used to infer valid consequences from them.

In Thagard's model a theory consists of rules and concepts that more or less represent conditional propositions and predicates, respectively. Which inference rule to apply to determine a consequence of a theory is arbitrary in logic, but controlled by a mechanism of spreading activation in PI. Superficially, this difference only has consequences in the performance of the process of generating an explanation or predic-

tion. In principle the same consequences could be inferred from the different representations of a theory in both the cognitive and the logical model.

Even in the process of inferring an explanation the main difference between the cognitive model PI and the logical model lies in their performance and the specific extra conditions. Thagard selects the best explanation on the equally decisive criteria of explanatory breath and simplicity, while the logic approach puts the priority on explanatory breath and consistency.

One important difference is that PI maintains different theories simultaneously, basing the use of any of the rules in prediction or explanation on its success in solving problems earlier. This allows PI's predictions to be inconsistent due to the firing of competing rules.

The nature of the heuristic rules of BACON differ in type from those of PI. The BACON heuristics represent a very specific kind of abduction. PI's primary abductive mechanism reasons from all kinds of conditional assumptions represented in the PI rules. The BACON heuristics incorporate an abductive suggestion based on a conditional proposition, *e.g.* the proposition that if the quotient of two variables is constant, then the values increase together. Any term proposed by those heuristics (INCREASING/DECREASING) is tested on the available data terms by other heuristic rules (LINEAR/CONSTANT) that propose it as a law or ignore it if it does not fit the data.

The BACON production rules implement a particular heuristic method, and not a part of a theory as in PI. The rule representation of either a theory or heuristic is subtle. For the predictive nature of a theory it is not important whether you represent a theory as a set of conditional statements or as a set of production rules, as long as the specific production rules, or the conditional statement together with general inference rules produce or define the same consequences.

It is possible to understand a theory both declaratively and procedurally in ACT-R. The structure of a theory can start out as a declaration in memory chunks. What consequences will follow from it depend on the production rules that can make an inference about it. It is possible to represent both the axioms of a theory and general inference rules declaratively, and a method to infer deductive consequences from them can be represented by a set of productions.

I summarize the different uses of the rule concept in explanation models in Table 5.6. Rules can be considered to be secondary processes in all the cognitive models. But in PI they are part of theory, while in BACON.1 they are part of method. In ACT-R a production can be understood as both part of theory and/or heuristic method.

In ACT-R a production, seen as an inference procedure, takes as premises a goal and an assumption, and produces a new goal as conclusion. This goal can be either to make a new assumption, to observe or to intervene within something in the world. If the premise of a specific production includes declarative assumptions of concept types A and $A \rightarrow C$ and the goal is to produce a valid consequence, then the production represents the application of *modus ponens* if its new goal is to assume C.

Explanation	Logic	PI	BACON.1	ACT-R
Background Structures	B: {A→C}		(if X/Y=c then inc)	Chunk: (rule A C)
Processes		(If A then C) Rule = theory	(If inc then ratio) Rule = method	(If (rule A C) goal C then A) Rule = method & theory
Premise	P: {C}	(START C)	(goal) (X …) (Y…)	(goal C)
Inference	Abduction Conditions	Activation Abduction IBE	Rule matching	Creation (Analogy) Selection (Activation) Evaluation (PG-C)
Conclusion	H*: {A}	(EXPL. A)	(X^nY^m=c)	(A)

Table 5.6: Different uses of the rule concept in explanation

But how to understand the generation of specific explanations? As we saw in the earlier sections, in ACT-R this question is not so much about what productions can find an explanation, but how productions that can find explanations are created and evaluated themselves. This is a process in ACT-R that starts with an example of a specific explanation and a similar example solution to another problem. The example is mapped to the new problem, resulting in new productions. These productions can become either applicable to very specific cases or very general cases, for which the inferred explanation has a very high or low probability of being correct. By solving many explanation and prediction problems, use and experience will determine their success. The resulting productions can be associated with the typology of strong and weak heuristics, see Table 5.7.

The term heuristic comes from the Greek *heuriskein* meaning “to discover”. (*Heuriskein* is also at the origin of *eureka*, derived from Archimedes’ reputed exclamation, *heurika* (for “I have found”), uttered when he had discovered a method for determining the purity of gold by taking a bath) In artificial intelligence it is generally used to describe a process of learning by trying. It is often contrasted by the term algorithm, which is a derivation of the name of the Arab mathematician, Al-Khowarizmi (±825 AD). Both an algorithm and a heuristic are procedures for solving a problem.

The main difference between them is that an algorithm is meant to effectively solve a particular type of problem, often at high cost in time depending on the complexity of the problem. A heuristic is a tradeoff between time and optimality, it may solve a problem, usually at lower cost in time, but then it may not provide the best solution. Another difference is that the effectivity of an algorithmic procedure can usually be established analytically by mathematical proof, while the effect of a heuristic procedure is often established empirically, by experience of use.

Productions	High cost	Low cost (efficient)
High probability (effective)	Strong heuristic Specific method/theory	…
Low probability	…	Weak heuristic General method/theory

Table 5.7: Typology of productions in the light of expected gain (E = PG – C)

The heuristic procedure in ACT-R differs from the static heuristic procedure in BACON in such a way that the estimation of the cost and chance of success of a certain production (the estimated gain PG – C) is constantly evaluated and adjusted. So if we want to explain the process of discovering a theory or law it is not enough to point to a set of heuristics as the cause of that discovery. The heuristics are usually part of the product of that discovery. For Kepler to discover his law he first had to discover that he could compare Borelli's data with a particular kind of example functions.

Now if we understand the nature of a theory as being partly procedural we can also better understand how to see Kuhn's picture of science as a practice that reasons on the basis of paradigms as shared examples. In normal science a set of successful examples of explanations leads to a strong heuristic that can successfully solve highly specific problems within a domain. At a given time these heuristics, that incorporate part of the theory of that domain, may not be able to handle novel problems. A revolution is needed to start off a different approach, where only weak heuristics may be of some help. More specific and stronger heuristics will be learned once some success is booked.

So, to understand a theory and use it rationally is to learn a skill of a specific practice. You can tell a lay person that $E=mc^2$, but without a general skill in mathematics and a specific skill of how to apply those variables to a domain of phenomena, that person will not be able to predict or explain specific facts with that statement.

In Kepler's and Galileo's time science had become successful by applying simple and general mathematical functions to empirical phenomena en testing the predictions of those functions in experiments. But the practice of empirical science consist of the use of many, highly specific, constantly adjusted rules to explain and predict phenomena. A reflection of those rules can be declaratively represented and communicated, that is what this thesis is all about. Their use cannot be learned otherwise then by taking part in that practice. But is the way scientists actually use method and theory a criterion for what is rational, from an epistemological point of view?

5.7 Descriptive and normative

I argued how the ACT-R theory can provide an explanation of the rational behavior of scientists. But what does that tell us about what is rational? It is argued in epistemology that explaining the beliefs and methods of scientists by pointing to the cognitive process that creates and evaluates them is not sufficient for epistemically justifying those beliefs and methods.

People, scientists not excluded, make mistakes in their reasoning, as psychological experiments prove. It should be the role of epistemology to point out those errors, so that human reasoning can improve. So let us look at the rationality of human prediction and explanation.

Prediction

Human performance in logical reasoning has been a much studied subject in cognitive psychology (Anderson, 1995). In experiments by *e.g.* (Marcus & Rips, 1977) subjects were asked to evaluate the correctness of hypothetical syllogisms, represented as relatively neutral arguments such as:

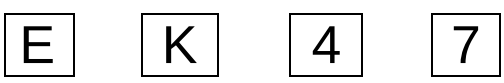
> If the ball rolls left, the lamp will switch on.
> The ball rolls left
> Therefore, the lamp will switch on.

It was asked if a conclusion is always, sometimes, or never correct. It was shown that in 100 percent of the cases subjects have no problems with judging the conclusion of *modus ponens* (affirming the antecedent) to be always correct, but that in only some 80 percent of the cases subjects judged the conclusion of denial of the antecedent and affirming the consequent to be merely sometimes correct. Still worse, in only 60 percent of the cases subjects thought that *modus tollens* (denying the consequent) is always correct. This performance was initially explained by the assumption that subjects interpret "if A then C as a biconditional instead of a conditional statement. Subjects were thought to understand the antecedent to be a necessary condition for the consequent, explaining why in some cases it was thought that the conclusion of denial of the antecedent or affirming the consequent is always correct. However, this does not explain the poor performance on judging the validity of *modus tollens*.

It is remarkable to see that the inference that is the hallmark of valid reasoning in science according to Popper is so often misjudged in common sense reasoning. It also testifies to the unpopularity of Popper's method of falsification as noted by Kuhn and many others. But it would be too swift a conclusion to mark the disregard of *modus tollens* in the practice of both common sense and scientific reasoning as irrational. It becomes more clear if this practice is seen as based on a probability assessment. I will demonstrate this by discussing another much studied task, called the Wason selection task.

Understanding the performance of subjects on the selection task is relevant for understanding how scientists evaluate potential hypothesis in the process of scientific discovery. This task is argued to demonstrate the failure of applying *modus tollens*. However, I will argue how this task shows how subjects make a perfectly rational probabilistic assessment.

In the selection task subjects are shown four cards with the following symbols:



They are told that every card contains a number on one side and a letter on the other.

The task is to test the validity of the following rule for these four cards:

If there is a vowel on one side, then there is an even number on the other side.

Subjects were asked to turn over only those cards that need to be turned over to test the rule. On average (Anderson, 1995) 89 percent chose to turn the E, affirming the antecedent of the rule. Logically this is an informative choice because the outcome of the experiment either falsifies or confirms the rule. However, 62 percent chose to also turn over the 4, affirming the consequent. Logically this provides no information because the outcome confirms the rule either way. The same goes for turning over the K denying the antecedent, which was done by 16 percent. Only 25 percent chose to turn the 7, denying the consequent, which logically also can confirm or refute the rule.

Oaksford and Chater (1996) argued that what subjects do is make a choice of the most informative cards in a statistical sense. They presupposed a probabilistic model of the rule $A \rightarrow C$, see Table 5.8. It provides the probabilities for the four possible states of the world where A and C are either true or false. Given this probabilistic model for the rule $A \rightarrow C$ and a null rule, *i.e.* a rule which does not have any probabilistic contingency between A and C, the interpretation of the conditional probabilities of A and C can be calculated, see Table 5.9a, see also Table 4.4 prediction.

Given the probabilistic interpretation both the AA and DC predictions are probable, while AC and DA are less probable. Yet subjects prefer AC much more than DC. To explain this, Oaksford an Chater argued that a card would be informative if the expectation of its outcome would differ from the expectation based on a null rule that assumes no relation between the antecedent and the consequent. However, in their model they need to set the conditional probability of the consequent C, given the antecedent A and vice versa to be 40% instead of a neutral 50% to explain the preference order of subjects.

Antecedent A	Consequent C	$A \rightarrow C$	$A \rightarrow C$	null	$A \rightarrow C$ *	null *
True	True	True	.40	.16	.50	.25
False	True	True	.20	.24	.23	.25
False	False	True	.30	.24	.18	.25
True	False	False	.10	.36	.09	.25

Table 5.8: The logical, and two possible probabilistic models of $A \rightarrow C$ and null

I think that there are three problems with the explanation of Oaksford and Chater. First, the particular probability distribution of the conditional statement is not properly defended. Secondly, a proper 50/50 null rule defeats their ordering. Thirdly and most importantly, the probability of a rule's prediction does not reflect the rule's probability given the outcome of the experiment.

It may well be possible that for subjects the probability of a rule $A \rightarrow C$ depends on the assumed model of the rule, not on the probability of a rule's prediction. In this interpretation the value of an experiment is the difference between the probabilities of a rule given the possible outcomes of that experiment. Given this interpretation the second problem becomes obsolete by addressing the first problem.

What is a proper model for a general conditional statement? One could argue that the preference of subjects in the card selection task actually reflects an average model for a conditional rule. If we redistribute the preferences of subject over 100% and take that as a value estimate, then we come to an average model that is approximated in Table 5.8 for rule A → C *. In this estimate subjects tend to regard the average probability of a rule slightly higher when only C is observed (.23), compared to when A nor C is observed (.18), see Table 5.9 b. It can be assumed that these numbers at best reflect a base rate probability that is different and adjusted for every particular conditional assumption that is maintained in memory.

	B	H	P	p(P|B&H)	p(P|B & null)	Difference	Subj. pref.
a.							
AAH	A	A → C	C	.80	.40	.40	89% E
ACH	C	A → C	A	.67	.40	.27	62% 4
DCH	Not C	A → C	Not A	.75	.60	.15	25% 7
DAH	Not A	A → C	Not C	.60	.60	.00	16% K
b.					p(P|B & null*)		
AAH	A	A → C *	C	.84	.50	.34	89% (47%)
ACH	C	A → C *	A	.68	.50	.18	62% (32%)
DCH	Not C	A → C *	Not A	.67	.50	.17	25% (13%)
DAH	Not A	A → C *	Not C	.56	.50	.06	16% (8%)
							(100%)
c.				p(H|B&P)	p(H|B & ¬P)		
AAH	A	A → C *	C	.50 (C)	.09 (R)	.41	47%
ACH	C	A → C *	A	.50 (C)	.23 (C)	.27 (-.14)	32% (-15)
DCH	Not C	A → C *	Not A	.18 (C)	.09 (R)	.09 (-.18)	13% (-19)
DAH	Not A	A → C *	Not C	.23 (C)	.18 (C)	.05 (-.04)	8% (-5)
d.							
HAA	A → C *	A	C	.68 (?)	.33 (R)	.35	
HAC	A → C *	C	A	.84 (C)	.44 (?)	.40	
HDC	A → C *	Not C	Not A	.56 (?)	.16 (R)	.40	
HDA	A → C *	Not A	Not C	.67 (C)	.32 (?)	.35	

Table 5.9: Different kinds and models of probabilistic prediction

So given the above model the value of AA is the highest because that model assumes the rule is the most probable if A and C are true (.50) and the least probable if A is true and C is false (.09). The value for DA is the lowest because either outcome says about the same (.18/.23) about the probability of the rule, given the model. The reason that AC is more preferable than DC is that the difference between the outcomes for the former experiment is much higher (.50 – .23) than that of the second (.18 – .09). The outcome of DC may logically be able to either defeat or confirm the rule given the logical model of the rule, but with a probabilistic model either outcome of a DC experiment will result in a low probability.

To make the comparison with the logical model complete I also listed the kind of predictions where the rule is assumed and the antecedent or consequent is hypothetically affirmed or denied. A probabilistic interpretation now provides an assessment where the logical approach could not give an answer about the probability of the hypothesis, see Table 5.9d.

From this viewpoint subjects predictions and experiments do not seem to be all that irrational, as long as hypotheses are interpreted to be more or less probable instead of just true or false. In a game like situation, where the rules are strict and given, it is rational to follow the logical model of a rule. But in an empirical situation where rules are not known to be true and almost all rules have exceptions acting on a probabilistic assessment is more rational. Yet Popper would probably argue that the question remains how probability assignments to hypotheses can be rational. This question will be addressed in the Chapter 6.

5.8 Explanation and evaluation

According to Langley et al (1987, p.47) in discovering a hypothesis "rationality for a scientist consists in using the best heuristics available for narrowing the search down to manageable proportions. A normative theory of creativity and scientific discovery is concerned with this kind of rationality." So instead of focussing on the validity or probability of hypotheses found by heuristics, they emphasize the efficiency part of rationality. They assume you know what you are looking for. For Bechtel (1988) to normatively evaluate a heuristic is to identify its failure. He assumes you know when a heuristic fails. But how to know what you are looking for and how to know you failed to find it?

In epistemology it is a much debated question whether the identification of the failure or success of assumptions is an analytical or empirical matter. This holds for both theoretical and methodological assumptions. In a psychological explanation of scientific practice the identification of epistemic success or failure seems foremost an empirical matter. Productions and chunks are created and evaluated by their success in use, whether they are part of theory or of method. But the success of productions can only be measured by given conditions for success. And testing if a proposed solution satisfies those conditions is an analytical matter.

According to logic the best theory should be: consistent, internally and with respect to background knowledge; complete and correct with respect to the phenomena it explains; non-trivial; informative, and it should be simple. So different methods are suggested that prefer theories with regard to their competitors by their consistency, correctness and completeness, non-triviality, empirical content, and simplicity. Different philosophers prefer one condition above another on the basis of different arguments.

Scientists usually also entertain other preferences such as analogy, beauty or symmetry in a theory. Finding a theory that satisfies those conditions means success. But the most important condition of any theory is that it should remain successful in the future. So the questions with respect to the probability part of the rationality of reasoning are: 1. which conditions are conductive to empirical success, 2. why are they conductive to success, and 3. how to pursue them?

First there should be made a distinction between conditions that are part of the main goal of science, and those that may be conductive to it. I gather that conditions such as:

C_1. Correctness C_2. Consistency C_3. Completeness

are part of the main goal of science. This is what we want to achieve: a theory that has no anomalies, covers the domain, and is not trivial by allowing everything. These conditions are not conductive to empirical success, they define it. But how to pursue them? The satisfaction of the first two conditions can never be validly established in an empirical domain. The future can always bring a situation that is not allowed or included in the theory. The best we can do is to analytically check for internal consistency and to check for correctness and completeness with respect to all available observations. However: in principle, infinitely many theories can be entertained that satisfy all three conditions; in practice, however, it is hard to find even one theory that comes close to that goal.

Scientists build theories incrementally, constantly proposing and revising hypotheses, often within the conceptual boundaries of a research program. The question is whether it is rational to pursue correctness and completeness by preferring to pursue only a proposal that is closest to the goal. At any given time that goal seems clear. There is a set of current observations and the problem is to find that theory that covers most of them.

So, is it rational to entertain and pursue a consistent theory that explains most data and has the fewest number of counterexamples? By definition that theory is closest to the goal of science, assuming that all other possible theories are known to be worse. Yet in practice we do not know the merits of all possible other theories since we do not know them all. It may turn out that amending a theory that was further from the goal proves more successful than working on the best one available. Given a conceptual space of all possible theories and a set of all observations, the theory that best satisfies the goal at any moment of development may be stuck in a so-called local maximum. Pursuing predictions and revisions of a theory that is further from the goal may reveal a better approximation. In cognitive psychology and AI the first approach is know as hill-climbing. Going straight for the top may bring you to the top of the hill, but may miss the mountain. A scientist that chooses to stay with a successful theory that lacks progression is as rational as a chicken that gets stuck in a fence when running toward the corn in view, not able to back up to go around the open gate door.

In practice it does not always work that way. Scientists do not only pursue correctness, completeness and consistency. They also entertain conditions such as *e.g.*:

C_4. Simplicity C_5. Analogy C_6. Symmetry

In logic these conditions are meta-epistemical, they do not inform us about the truth of a theory. However, in scientific practice these conditions often prevail above correctness, completeness and consistency. (We will see an example of this in the case study in the next part of the thesis.)

Thagard incorporated these conditions in his theory of explanatory coherence, which meant to explain scientists' preferences. The program ECHO implements a model of a neural network that can evaluate how close a theory is to all conditions, as compared with a competitor (Thagard, 1992). Yet this theory fails to explain why it is

sometimes rational to prefer conditions C_4-C_6 above C_1-C_3. How can these conditions, or methods based on them be conductive to the empirical success of a theory?

A naturalistic way out to this question is to explain why scientists have certain preferences by bringing in evolution, both biologically and socially. Primary mechanisms in our brain have preferences for certain assumptions and methods given experience. Survival depends on being able to make methodological decisions and retrieve memories of experiences that are relevant to the current situation or problem. An organism that is not able to make decisions or assumptions successfully is less likely to survive. In the development of our species nature favors particular primary cognitive mechanisms in the face of lions and gathering food; in the development of science nature favors particular theories, methods and scientists, in the face of peers and trying to get tenured positions.

To return to Goldman's distinction (Section 5.2): we have gone through an exposition of some (secondary) methods and theories and how they are generated and evaluated by some (primary) mechanisms of the brain during scientific discovery. I argued how these mechanisms tell us something about rationality. They inform us what rationality is, for a scientist.

However, these primary mechanisms still do not inform us *why* it is epistemically rational to maintain certain theories and methodologies. These mechanisms prefer a theory or method if it proves successful in solving problems, in reaching certain goals, satisfying certain conditions. But why are some conditions more rational to pursue than others, why are they more successful? A naturalistic stance would be happy with just the observation that certain conditions, methods, hypotheses and theories are more successful than others, as an inductively assumed fact of the world. Yet, in the next chapter I will pursue an explanation of one of those facts, why one of those conditions, simplicity, is conductive to attaining the goal of science.

Epistemologists reason to study reason is to be able to improve it. In this chapter we have come to understand reasoning as a process of inferring conclusions that satisfy certain conditions, given a certain problem. So to understand and evaluate the reasoning in a specific discovery process normatively it is first of all important to understand the details of a specific problem, *i.e.*:

- starting situation
- background assumptions
- process to reach the goal
- goal properties
- end results

In practice none of the above stay constant in the process. The starting situation changes, new background assumptions and concepts are added or withdrawn, end results are different from the goal, the goal conditions shift, and new methods to reach the goal are introduced. All under influence of primary cognitive mechanisms and social interaction. How this process goes about in the practice of neuropharmacology will be discussed in the next part of this thesis.

5.9 Conclusion

The particular question of this chapter was: how to understand and model scientific discovery, in ACT-R? I will answer this question by going through the answers for the specific questions of this thesis from Section 1.3:

Question 1 What is the structure of a scientific theory? In ACT-R theories can be understood as a collection of statements containing laws, examples and solutions to earlier explanation and prediction problems, represented declaratively in memory chunks, and specific and general procedures, represented in production rules. Chunks, represented as sets of slots and values of a certain type, are assumed to be the results of perception and solutions to solved problems. Production rules are represented as condition-action pairs: given a goal and an assumption chunk a new goal is set which can lead to either a new assumption or doing a particular observation or intervention in the world. Productions can be part of both theory and method.

Question 2 What is the process of scientific reasoning? The process of scientific reasoning in ACT-R contains of learning heuristic problem solving skills in searching and evaluating explanations and predictions of phenomena, see Table 5.10.

Problem	Start	Background	Process	Goal	Goal properties
Explanation	Goal = explain observation P	H' explains P' Productions	Creation Selection Evaluation	H*	H* explains P Analogy Probability
Prediction	Goal = predict hypothesis H	H' predicts P' Productions	Creation Selection Evaluation	P*	H explains P* Analogy Probability

Table 5.10: Short overview of reasoning problems discussed in this chapter

The process of both explanation and prediction starts with a goal chunk together with examples and productions in memory in the background. A solution to the problem is either selected from memory by productions or created based upon examples by analogy, and evaluated probabilistically.

Question 3 What is the route between theory and experiment? The assumed route between theory and experiment walked by a scientist starts with a goal and assumptions in memory that determine new assumptions and actions, based on learned productions. Failure to achieve a goal decreases the potential to recall an assumption and the chance that the used productions will be employed in the future.

This can explain how scientists go through the ideal six steps introduced in the last chapter. In a scientific study of scientists doing their work you would get the following scheme, where lowercase p denotes a phenomenon and uppercase P a proposition about that phenomenon:

1. Observe phenomenon p: see $p_m,\ldots,p_n$ (activities of scientist x at work)
2. Describe p: $P_m \rightarrow P_n$ (problem solving behavior)
   P_1: {x observes phenomenon *p*: x sees p_m}
   $P_1 \rightarrow P_2$: {x describes *p*: $P_m \rightarrow P_n$}
   $P_2 \rightarrow P_3$: {x explains *p*: x finds $B \cup H^* \models P_m \rightarrow P_n$}
   $P_3 \rightarrow P_4$: {x predicts *p*: x finds $B \cup H \models P_i^* \rightarrow P_j^*$}
   $P_4 \rightarrow P_5$: {x intervenes *p*: x creates P_i^*}
   $P_5 \rightarrow P_6$: {x observes *p*: x sees P_j^*}
3. Explain p: $B \cup H^* \models P_m \rightarrow P_n$
   H^*: {ACT-R cognitive mechanisms} $\models P_m \rightarrow P_n$
4. Predict p: $B \cup H \models P_i^* \rightarrow P_j^*$
   B: {specific chunks and productions of BACON}
   P_2^*: {x describes p: P_1 : {D = ⟨1, 4, 9⟩} $\rightarrow P_2$: {P = ⟨1, 8, 27⟩}}
   $P_2^* \rightarrow P_3^*$: {x explains p: x finds *H*: $\{D^3/P^2 = c\} \models P_1 \rightarrow P_2$}
5. Intervene in p: create p_i^*
6. Observe p: observe p_j^* ?

In this way a step in the process of scientific problem solving is described as a conditional statement. In step 1. the activities of a scientist are observed as a phenomenon. In step 2. these activities are described. One can observe a scientist making observations (P_1) and describing them (P_2). Logically one can describe the link between those activities by a conditional statement ($P_1 \rightarrow P_2$). The antecedent of the conditional statement represents the start situation, the consequent represents a goal situation. In step 2. of describing the activities of a scientist, one can further describe how a scientist explains (P_3) predicts (P_4) intervenes in (P_5) and again observes (P_6) a phenomenon. In step 3. of our cognitive research of scientific activities an hypothesis is searched to explain the process of those scientific activities, in this example cognitive models in the ACT-R architecture are proposed. In step 4. we make a prediction about how our scientist under study can find a law (P_3^*) that can imply data that describes a phenomenon (P_2^*). This prediction can be tested in step 5. and 6.

It can be a task for cognitive psychology to explain and predict how scientists search for a solution of scientific problems. For naturalistic epistemology it is the task to find an intervention in step 5. such that scientists can observe, describe, explain, predict and intervene the phenomena they are interested in more effectively and efficiently, and to explain why they do so. Why some explanations might be more effective than others will be the topic of the next chapter.

* * * * *