3

Two notions of representation in the classical computational framework

In the last chapter, we saw how representation in the CCTC is commonly regarded. In this framework, representation is generally treated as closely linked to our commonsense conception of the mind and, in particular, to our understanding of propositional attitudes. We also saw how this perspective fails to provide an adequate account of why the CCTC needs to appeal to representations at all. If the Standard Interpretation was the only interpretation, we would have little reason to suppose that there is any real explanatory pay-off in treating the posits of classical AI as standing for something else.

But the Standard Interpretation is not the only way to look at things. In this chapter I want to present another perspective on the CCTC, one that I think reveals why the classical framework provides a legitimate home for a robust notion of internal representation. Actually, my claim will be that there are two related notions playing somewhat different but nonetheless valuable explanatory roles. One notion pertains to the inputs and outputs of computational processes which help to define the cognitive task being performed. As we'll see, given the sort of explanatory strategy usually adopted by the CCTC, this also provides a notion of *inner* representation as well. The second notion pertains to data structures that in classical explanations serve as elements of a model or simulation. That is, according to many theories associated with the CCTC, the brain solves various cognitive problems by constructing a model of some target domain and, in so doing, employs symbols that serve to represent aspects of that domain. Since other authors have already provided detailed explications of these representational notions, my goal will be to provide an overview and, where necessary, perhaps modify or extend these earlier analyses.

Both of the notions of representation I am going to defend in this chapter have been criticized as suffering from serious flaws. One alleged problem, related to concerns discussed in the last chapter, challenges the idea that these representational notions are sufficiently robust to qualify as

real representations, as opposed to merely instrumental or heuristic posits. A second worry is that the account of content connected to these notions is unacceptably indeterminate between different possible interpretations. I plan to demonstrate that once we appreciate the sort of explanatory work these notions are doing, we can see that their alleged shortcomings are actually much less serious than is generally assumed. Both notions are quite robust, and while there is indeed an issue of indeterminacy associated with them, it doesn't have any bearing on the explanatory work they do in the CCTC. I should say up front, however, that I have fairly modest goals in this chapter. I do not intend to address all of the various problems and challenges that have been raised (or could be raised) in connection with these notions of representation (in fact, I doubt if such an exhaustive defense is possible for *any* representational posit). My aim is simply to show that there are notions of representation in the CCTC that are not based on folk psychology, that are essential to the explanatory strategies offered by the CCTC, and that can handle some of the more basic worries associated with naturalistic accounts of representation. If I can demonstrate that the CCTC posits internal representations for good explanatory reasons, then I will have accomplished my primary objective.

To show all this, the chapter will have the following organization. First, I'll provide a sketch of each notion of representation and show how it does valuable explanatory work in the CCTC. Then, I'll consider two popular criticisms against these notions – that they are merely useful fictions and that the associated theory of content is plagued with rampant indeterminacy. I'll argue that both criticisms can be handled by paying close attention to the way these notions are actually invoked in accounts of cognition. Finally, there are a number of side issues that it will help to address for a more complete picture. In the final section, I offer a brief discussion of each of these important side issues.

3.1 IO-REPRESENTATION

In the last chapter, we saw how Marr's model of cognitive science involved three levels of description and how the "top" level involved the specification of a function that more or less defines the sort of cognitive capacity we want explained. Consider again a simple operation like multiplication. Although we say various mechanical devices do multiplication, the transformation of numbers into products is something that, strictly speaking, no physical system could ever do. Numbers and products are abstract entities, and physical systems can't perform operations on abstract entities. So, at the

algorithmic level we posit symbolic representations of numbers as inputs to the system and symbolic representations of products as outputs. We re-define the task of multiplication as the task of transforming numerals of one sort (those standing for multiplicands) into numerals of another sort (those standing for products). The job of a cognitive theory is to explain (at this level of analysis) how this sort of transformation is done in the brain.

In fact, this general arrangement, whereby the explanandum is characterized as the conversion of representational inputs into representational outputs, will apply to most approaches to cognitive explanation. This is simply because cognitive processes themselves are typically characterized as an input–output conversion couched in representational terms. Pick any cognitive capacity that you think a scientific psychology should attempt to explain, and then consider how it should be characterized. For example, take the ability to recognize faces. The input to any cognitive system that recognizes faces will not be actual faces, of course, but some sort of visual or perhaps tactile representation presented by the sensory system. The output will also be a representation – perhaps something like the recognition, "That's so-and-so," or perhaps a representation of the person's name. Or consider linguistic processing. The challenge for most cognitive theories is *not* to explain how an event characterized in physiological terms (say, eardrum motion) brings about some other event characterized in physiological terms, but rather, how an acoustic input that represents a certain public-language sentence winds up generating a representation of, say, a parse-tree for that sentence. A theory about how the visual system extracts shape from shading is actually a theory about how we convert representations of shading into representations of shape. The same general point holds for most of the explananda of cognitive science. Indeed, this is one of the legitimate senses in which cognitive systems can be viewed as doing something called "information processing." While automobile engines transform fuel and oxygen into a spinning drive-shaft, and coffee-makers convert ground coffee to liquid coffee, cognitive systems transform representational states into different representational states.

Given the sort of analysis I am offering, an immediate question that arises about these types of input–output representations concerns the way they meet the job description challenge. In what sense do they function *as* representations, not just for our explanatory purposes, but for the actual cognitive system in question? There are two possible answers that could be offered. The first is to avoid the question altogether and say that the question is outside of the domain of cognitive theorizing. Cognitive theories are in the business of explaining the processes and operations that convert input

representations into output representations; the concern of these theories (and therefore my analysis) is with the nature of *internal* representations. The nature of the input and output representations that define cognitive operations (and thereby define psychological explananda), while perhaps an important topic, is not an important topic that is the primary concern of cognitive modelers. Theoretical work has to start somewhere, and in cognitive science it starts with an explanandum defined in this way.

However, while there is some truth to this answer, it is as unsatisfying as it is evasive. A second and better (though admittedly controversial) answer is to say that there is considerable evidence that minds do certain things, and one of the main things they do is perform cognitive tasks properly described as the transformation of types of representations. It appears to be a fact of nature that certain minds can do multiplication, recognize faces, categorize objects, and so on. Well, what does that mean, exactly? It means that the cognitive system in question can convert, say, representations of numbers into representations of their product, or perceptual representations of an object into a verbal classification. The states that are the end-points of these processes are thereby serving as input–output representations for the cognitive system in question. The end-points serve as representations not because cognitive researchers choose to define them that way, but because we've discovered that cognitive systems employ them that way, given the sorts of tasks they actually perform. Below, I'll return to this question as it pertains to an *internal* sort of input–output representation. For now, the key point is that we are justified in treating a cognitive system's inputs and outputs as representations because, given what we know about cognitive systems, we are justified in characterizing many of their operations as having certain types of starts and finishes; namely, starts and finishes that stand for other things.

Cummins offers this explanation of the input–output notion:

> For a system to be an adder is for its input–output behavior to be described by the plus function, $+(<m, n>) = s$. But + is a function whose arguments and values are numbers, and whatever numbers are, they are not states or processes or events in any physical system. How, then, can a physical system be described by +? How can a physical system traffic in numbers, and hence add? The answer, of course, is that numerals – that is, representations of numbers – can be states of a physical system, even if the numbers themselves cannot ... The input to a typical adding machine is a sequence of button pressings: $<C, A1 + A2, =>$, that is, < clear, first addend, plus, second addend, equals >. The output is a display state, *D*, which is a numeral representing the sum of the two addends. (Cummins 1991, p. 92)

Cummins calls this the "Tower-Bridge" picture of representation, because it involves two levels of transformations – physical and, in the case of

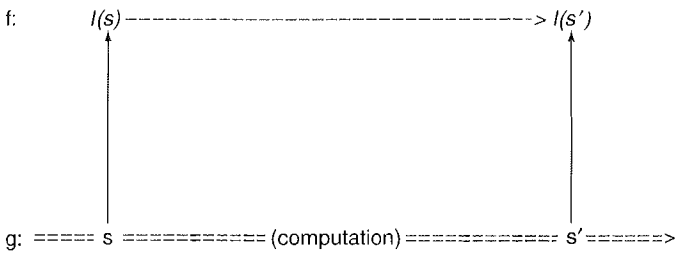

Figure 3a Cummins's proposed Tower-Bridge picture of computation (1991). The top level is the cognitive task being explained (f), the bottom level (g) is the algorithmic level of computational processes. The vertical arrows correspond with the interpretation of the bottom level input and output symbols, s and s'. Reprinted by permission from MIT Press.

addition, mathematical, which are conjoined on either end by semantic links between the physical representations and the things they stand for. Schematically, the picture is illustrated in figure 3a.

In much of his writing, Cummins characterizes this notion of representation as *the* notion employed in the CCTC. Because connectionist accounts also appeal to representations as the inputs and outputs of their networks, this leads him to the surprising conclusion that the CCTC and connectionists use the same notion of representation. This outlook is correct if we only consider the way both theoretical frameworks adopt similar specifications of psychological explananda. However, it is important not to confuse theory-neutral specifications of the explananda with the internal explanatory posits of particular cognitive theories. Since cognitive processes are defined with representational states as their end-points, it is a mistake to treat this notion of representation as *belonging to* the CCTC, or invoked *by* the CCTC. Since most theories treat types of input–output transformations as their starting point, the input and output themselves are not part of any particular theory's explanatory apparatus.

Nevertheless, a very similar sort of representational notion *does* play a critical role in the CCTC. This becomes clear once we look *inside* of cognitive systems as they are understood by the CCTC accounts. As we saw in the last chapter, sophisticated cognitive capacities are typically explained by the CCTC by supposing that the system is composed of an organized system of less sophisticated sub-systems. By decomposing complex systems into smaller and smaller sub-systems, we can adopt a divide-and-conquer style of explanation whereby the performance of complex tasks is explained by the performance of increasingly simpler tasks (Fodor 1968; Cummins 1975, 1983; Dennett 1978). As Cummins puts it,

"psychological phenomena are typically not explained by subsuming them under causal law, but by treating them as manifestations that are explained by analysis" (1983, p. 1). Task-decompositional explanations are the norm in the CCTC, and they give rise to the popular "flow-chart" style of explanatory theory. It is this conception of cognitive systems that requires us to posit representations that serve as the inputs and outputs for the inner sub-systems that comprise the CCTC account. Internal mini-computations demand their *own* inputs and outputs, and these representations that are external to the mini-computation are, of course, *internal* to the overall system.

Task-decompositional analysis is a popular explanatory strategy in several different domains (like biology), yet theories in these domains don't all appeal to internal representations. So why are internal representations necessary for functional analysis when we are dealing with cognitive systems? The answer stems from the way the sub-systems and sub-routines in computational processes are typically understood. A general assumption of the CCTC is that many of the tasks performed by the inner sub-systems should be seen as natural "parts" of the main computations that form the overall explanandum. That is, they should be defined as procedures or sub-routines that are natural steps in a process that instantiates the more sophisticated capacity that is ultimately being explained. Our ability to do multiplication, for example, might be explained by appealing to a sub-process that repeatedly adds a number to itself (Block 1990). But to view the sub-process in this way – as a sort of internal mini-computation – then we need to regard *its* inputs and outputs as representations as well. If there is an inner sub-system that is an adder, then its inputs must be representations of numbers and its outputs representations of sums. If these internal structures are not serving as representations in this way, then the sort of task-decompositional analysis provided by the CCTC doesn't work. We won't be able to view the sub-system as an adder, and hence we won't be able to see how and why its implementation is essential to the overall capacity being explained. Consequently, certain structures that are internal to the system – structures that serve as inputs and outputs of certain intermediary sub-systems – must be seen as functioning as representations of matters that are germane to the overarching explanandum.

This point has been made with different terminology in Haugeland's classic treatise on cognitivism (1978). Haugeland introduces the notion of an intentional black box (IBB), which is (roughly) a system that regularly produces reasonable outputs when given certain inputs under a systematic interpretation of the inputs and outputs. Haugeland suggests that an information processing system (IPS) should be viewed as a type of

intentional black box that lends itself to a further analysis. Such an analysis usually involves an appeal to IBBs that are internal to the IPS – i.e., a task-decomposition of the larger system into smaller sub-systems. A crucial feature of this type of explanation, then, is that certain internal states are interpreted as representing facets of the task in question:

> Moreover, all the interpretations of the component IBBs must be, in a sense, the same as that of the overall IBB (=IPS). The sense is that they all must pertain to the same subject matter or problem . . . Assuming that the chess playing IBB is an IPS, we would expect its component IBBs to generate possible moves, evaluate board positions, decide which lines of play to investigate further, or some such . . . Thus, chess player inputs and outputs include little more than announcements of actual moves, but the components might be engaged in setting goals, weighing options, deciding which pieces are especially valuable, and so on. (1978, p. 219)

To avoid confusion, I'll refer to input–output representations that make up the explanandum of the cognitive theory as "exterior" inputs and outputs, and input–output representations that help comprise the explanans of the CCTC as "interior" inputs and outputs. Interior input–output representations are a sub-system's own inputs and outputs that are internal to the larger super-system's explanatory framework. Since it is not uncommon to have nested computational processes, the sub-system itself may have *its* own internal representations, which are themselves the inputs and outputs of a sub-sub-system operating inside the subsystem in question. Hence, being "exterior" and "interior" is always relative to the system under consideratation.

We can now see that Cummins's Tower-Bridge picture needs augmentation. In between the two main end-point spans, there should be several internal bridges with end-points defined by their own mini-towers, linking internal physical states (the interior input–output representations) to aspects of the target domain that they represent. A more accurate portrayal of the CCTC would be something like what is presented in figure 3b,

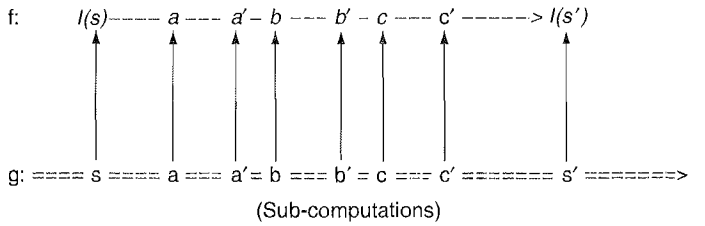

Figure 3b Cummins's Tower-Bridge diagram modified to accommodate inner computational sub-routines and representational states.

where the letters a, b, c correspond to the symbolic inputs of internal processes, while a′, b′, and c′ correspond to the representational outputs.

For our purposes, the most important aspect of this notion of representation is how it succeeds in meeting the job description challenge in a way that was not met on the Standard Interpretation. At least initially, we can see how interior inpur–output notion (or, the "IO notion") reveals how symbols *serve as* representations, given the hypothesized organizational architecture of the system. Data structures serve as representations because that is how the internal sub-systems treat them, given *their* job descriptions (e.g., performing addition, assessing chess moves, etc.). Serving as a representation of some feature of a target domain here amounts to serving as the sort of input or output required by a sub-processor solving a problem related to that domain. The content of the representation is critical for this role because unless the symbol stands for the relevant computational argument or value, it is impossible to make sense of the sub-system as a computational sub-system doing *its* job. Because it is an important element of this style of explanation, the interior IO notion of representation is not directly based on our folk notions of mental representation. We may come to view these inputs and outputs as thoughts, but the motivation to treat them as internal representations is not dependent upon our doing so. Even if folk psychology had never posited mental representations, the CCTC would still need to invoke interior IO-representations, given its explanatory framework. Yet while IO-representations don't accord with our commonsense understanding of *mental* representations, they nevertheless play a functional role that is intuitively representational in nature. It is an intuitively representational role because we recognize that systems doing things like addition, or comparing chess moves, treat their inputs and outputs as symbols standing for things like numbers or chess game scenarios.

Thus, the CCTC invokes a notion of internal representation that, contrary to what is implied by the Standard Interpretation, is actually built into the fabric of its explanatory framework and thereby does essential explanatory work. We can see this better if we briefly reconsider the criticisms of representationalism offered by Searle and Stich. The IO notion doesn't answer all of Searle's concerns about content and computational symbols. But consider the claim that there is *no* sense in which the symbols in the Chinese Room serve as representations for the system. On Searle's own account of the Chinese Room, the room does manage to provide appropriate answers to sophisticated questions about various topics. Suppose, in keeping with our algebraic theme, the questions asked are about the product of various numbers. So the input to the

Chinese room would be questions like "What is 3 x 7?," only written in Chinese. How does the room always manage to produce the right answer? According to the CCTC and Searle, the system does this by symbol manipulations that instantiate some sort of program. Let's assume the program is one that involves a sub-routine whereby one of the multiplicands is added to itself repeatedly.[1] We cannot understand this explanation unless we recognize that the man in the room's manipulations are, unbeknown to him, an adding process. And we cannot understand these manipulations as an adding process unless we recognize that Chinese characters generated by this process are serving as representations of sums. Putting it another way, we can't even make sense of how the symbol manipulations in the Chinese room succeed in generating the appropriate responses without invoking interior IO-representations. It doesn't matter that the person or thing manipulating the symbols doesn't understand what it is doing, or that the symbols lack the sort of intentionality associated with our thoughts. What matters is that we have an explanatory strategy that breaks a complex task (in this case, multiplication) into smaller tasks (i.e., addition) whereby the smaller tasks, by their very nature, require their inputs and outputs to be representations.

A similar point applies to Stich's anti-representationalism. Since on the Standard Interpretation, representational content appears to be superfluous to the CCTC type of explanations, Stich argues that the CCTC could get along just fine without it. But Stich's analysis is built on the assumption that the notions of representation at work in computational explanations are those derived from folk psychology. It neglects the possibility that there are notions of representation built into the sort of explanatory scheme adopted by the CCTC that need to be invoked for such a scheme to work. If we were to adopt the Syntactic Theory, avoiding all talk of representation and content, we would also be forced to abandon the type of task-decompositional explanation that is central to classical cognitive science. Since we couldn't treat the symbols as interior IO-representations, we couldn't

[1] The details might work as follows. After checking to see if one of the input characters represents either "0" or a "1," in which case special instructions would be followed, the man in the room is instructed to pick one of the input symbols and find its match on both the vertical column and horizontal row of what is actually an addition table. The syntactic equivalent of the other symbol is placed in a box. Once the symbol at the cross-section of the table is found (which would be the sum of one of the multiplicands added to itself), yet another symbol, designated by the instructions, is placed in another box. This is the system's counter. The symbol at the cross section of the addition table is then used to match a further symbol on the horizontal column, and the process repeats itself until the symbols in the two boxes match. At that point, a symbol matching the intersection symbol is handed through the output slot.

understand how the system succeeds by breaking a large computational operation down into related sub-operations. We could, of course, employ a syntactic type of task-decompositional explanation. We could track the causal roles of the syntactically individuated symbols, and thereby divide the internal processes into syntactic sub-processes. But we wouldn't be able to make sense of these operations as computationally pertinent stages of the larger task being explained. It is both explanatorily useful and informative to see a sub-system of a multiplier as an adder. It is not so useful or informative to see it as a mere syntactic shape transformer.

In accounting for the IO notion of representation, I've leaned very heavily upon the sort of explanatory strategy employed in the CCTC. I've suggested that because the CCTC uses a task-decompositional strategy that treats inner sub-systems as performing computations, then we need to regard the inputs and outputs of those sub-systems as representations. But this raises an important question – does the task-decompositional strategy provide a reason to think the inputs and outputs *actually are* representations, or does it instead merely provide us with a reason to *want* or *need* the inputs and outputs of these internal processes to be representations. Does it, from a metaphysical perspective, show us what serving as a representation amounts to? Or does it rather, from an epistemological perspective, create a need to have things serving as representations be the inputs and outputs for the inner computations?[2]

This is a difficult question and, quite frankly, I have changed my mind about its answer more than once. My current view is that the CCTC is committed to a sort of realism about inner computational processes, and this in turn reveals how the IO-representations actually function as representations, independent of our explanatory concerns. To adopt the language of Millikan (1984, 1993), the sub-systems act as representation "consumers" and "producers." But it is actually more complicated than this. They are consumers and producers of representations in a way that helps make the symbolic structures consumed and produced into representations (just as our consumption of a substance is what makes it have the status of food). The admittedly rough idea, briefly discussed above, is that computational processes treat input and output symbolic structures a certain way, and that treatment amounts to a kind of job assignment –

[2] As Dan Weiskopf has put it, "we seem forced to suppose that IO-representations are indeed representations because their being so is constitutive of the thing being explained (a kind of cognitive processor, i.e., a representation transformer). This doesn't directly answer the job description question, since we still don't know what properties metaphysically constitute IO-representations being representations" (personal communication).

the job of standing for something else. While an adder is something that transforms representations of addends into representations of sums, there is also a sense, given this arrangement, in which representations of addends are those symbolic structures that an adder takes as inputs, and representations of sums are structures an adder produces as outputs. There exists, then, a sort of mutual support between computational processes and representational states of this sort, with neither being explanatorily prior. Serving as a representation in this sense is thus to function as a state or structure that is used by an inner module as a content-bearing symbol. The inner modules are themselves like the inner homunculi discussed in chapter 1, whose treatment of their input and output can be seen as a type of interpretation. If the brain really does employ inner mini-computers, then their operations and transformations are, to some degree, what makes their input and output symbols into something functioning in a recognizably representational fashion. Below, in section 3.3.2, I'll address further the question of whether or not we can say the brain actually is performing inner computations in an objective, observer-independent sense.

These are just some of the issues that a sophisticated account of IO-representation would need to cover, and a complete account would need to explain considerably more. Yet remember that my primary objective here is fairly modest. Rather than provide a detailed and robust defense of this notion of representation in the CCTC, I merely want to reveal how the kind of explanations offered by the CCTC makes the positing of internal representations an essential aspect of their theoretical machinery. My aim is to demonstrate that there is a technical notion of representation at work within the CCTC and to show how that notion has a considerable degree of intuitive and explanatory legitimacy. Although the notion has little to do (at least directly) with our commonsense understanding of mental representations, it has a lot to do with the kind of explanations provided by classical computation theories. Yet it is not the only notion of representation in the CCTC that answers the job description challenge, has intuitive plausibility and does important explanatory work. The other notion is related, but nonetheless involves a different sort of representational posit that does different explanatory work. We turn to that notion now.

3.2 S-REPRESENTATION

In the first chapter we discussed Peirce's three different types of signs, noting that one of these, his notion of icons, is based on some sort of similarity or

isomorphism between the representation and what it represents. The idea that the representation relation can be based on some sort of resemblance is, of course, much older than Peirce and is probably one of the oldest representational notions discussed by philosophers. But there is also the related though different idea that there can be a type of representation based not on the structural similarity between a representation and its object, but between the system in which the representation is embedded and the conditions or state of affairs surrounding what is represented. A map illustrates this type of representation. The individual features on a map stand for parts of the landscape not by resembling the things they stand for, but rather by participating in a model that has a broader structural symmetry with the environment the map describes. A map serves as a useful and informative guide because its lines and shapes are organized in a manner that mirrors the relevant paths and entities in the actual environment. Given this structural isomorphism between the map and the environment, the map can answer a large number of questions about the environment without the latter being directly investigated. Of course, this is possible only if the specific elements of the map are treated as standing for actual things in the environment. The map is useful as a map only when its diagrams and shapes are employed to represent the actual things, properties and relations of some specified location. The same basic notion of representation is at work when we use models, such as a model airplane in a wind tunnel, or computer simulations of various phenomena. It is also at work when numerical systems are used to model real-world parameters or when geometrical figures are used to understand aspects of physical systems. These and other predictive/explanatory arrangements share with maps the core idea that some sort of structural or organizational isomorphism between two systems can give rise to a type of representational relation, whereby one system can be exploited to draw conclusions about the other system.

Along with Pierce, many philosophers have offered accounts of representation based upon these themes. For example, it forms an important part of Leibniz's theory of representation, where he tells us that representations involve "some similarity, such as that between a large and a small circle or between a geographic region and a map of the region, or require some connection such as that between a circle and the ellipse which represents it optically, since any point whatever on the ellipse corresponds to some point on the circle according to a definite law" (Leibniz 1956, pp. 207–208). More recently, Chris Swoyer has developed a more detailed general account of this type of representation, which he refers to as

"structural representation" (1991). Swoyer makes an impressive stab at constructing a detailed formal analysis of this notion, but even more beneficial is his analysis of the kind of explanatory framework it yields, which he calls "surrogative reasoning." As Swoyer notes, when maps, models and simulations are used, we typically find out something directly about the nature of the representational system, and then, exploiting the known structural symmetry, make the appropriate inferences about the target domain. As he puts it,

> [T]he *pattern* of relations among the constituents of the represented phenomenon is mirrored by the pattern of relations among the constituents of the representation itself. And because the arrangement of things in the representation are like shadows cast by the things they portray, we can encode information about the original situation as information about the representation. Much of this information is preserved in inferences about the constituents of the representation, so it can be transformed back into information about the original situation. And this justifies surrogative reasoning . . . (1991)[3]

What does this have to do with cognitive science and the CCTC? While this notion of representation may not capture all of the ways in which computational processes are regarded as representations, it serves as an important, distinct, and explanatorily valuable posit of classical computational accounts of cognition. Just a quick survey of many well-known computational theories of cognition finds this representational notion repeatedly invoked in one form or another. This includes such diverse cognitive theories as Newell's production-based SOAR architecture (1990), Winograd's SHRDLU model (1972), Anderson's various ACT theories (1983), Collins and Quillian's semantic networks (1972), Gallistel's computational accounts of insect cognition (1998),[4] and many other types of CCTC accounts. Stephen Palmer (1978) presents an excellent overview of the many ways in which this type of isomorphism-based representation appears in classical cognitive theories. While Palmer notes that the form these representations take in different theories can vary widely, they all share a basic nature whereby "there exists a correspondence (mapping) from objects in the represented world to objects in the representing world such that as least some of relations in the

[3] Here Swoyer refers to the entire pattern as the structural representation, but in other spots he seems to treat the *constituents* of the patterns as representations. I'm inclined to adopt the latter perspective, though as far as I can tell, very little rides on this besides terminology.

[4] Gallistel tells us, "[a] *mental representation* is a functioning isomorphism between a set of processes in the brain and a behaviorally important aspect of the world" (1998, p. 13).

represented world are structurally preserved in the representing world" (Palmer 1978, pp. 266–267).[5]

Perhaps the main proponent of the view that cognition is computational modeling is Philip Johnson-Laird (1983). Echoing one of our general concerns, Johnson-Laird laments the fact that most symbol-based approaches to explaining cognition "ignore a crucial issue: what it is that makes a mental entity a representation *of* something" (1983, p. x). To correct this oversight, he suggests we need to understand the way cognitive systems employ mental models, and how elements of such models thereby function as representations. For Johnson-Laird, the idea that problem-solving is modeling applies even for what seem to be purely formal, rule-driven cognitive tasks such as deductive inference. He offers a compelling and detailed theory of different mental capacities that is built upon the core idea that computational states serve as representations by serving as elements of different models.

Besides Swoyer, the philosopher who has done the most to explain this notion of representation – especially as it applies to the CCTC – is Cummins (1989, 1991). Cummins calls this notion of representation "simulation representation." Since Cummins's simulation representation is sufficiently similar to Swoyer's structural representation, I'll stick with the conveniently ambiguous term "S-representation" to designate the relevant category. Cummins first explicates S-representation by noting how, following Galileo, we can use geometric diagrams to represent not just spatial configurations, but other magnitudes such as velocity and acceleration. While there need be no *superficial* visual resemblance between representation and what is represented (velocity doesn't look like anything), there is a significant type of isomorphism that exists between the spatial properties of certain geometric diagrams and the physical properties of moving bodies that allows us to use diagrams to make inferences about the nature of motion. It is this same notion that Cummins argues is at the heart of the CCTC. In other words, when classical computational processes are introduced to explain psychological capacities, this often includes an invoking of symbols to serve as S-representations. The mind/brain is claimed to be using a computational model or simulation, and the model/

[5] I am claiming that the classical computational framework has been the main home for a model-based conception of representation. In the next two chapters I'll argue that non-classical frameworks, like connectionism, employ different notions of representation, notions that fail to meet the job description challenge. But there are also a few connectionist-style theories that invoke model-based representations in their explanations. See Grush (1997, 2004) and Ryder (2004) for nice illustrations of such theories.

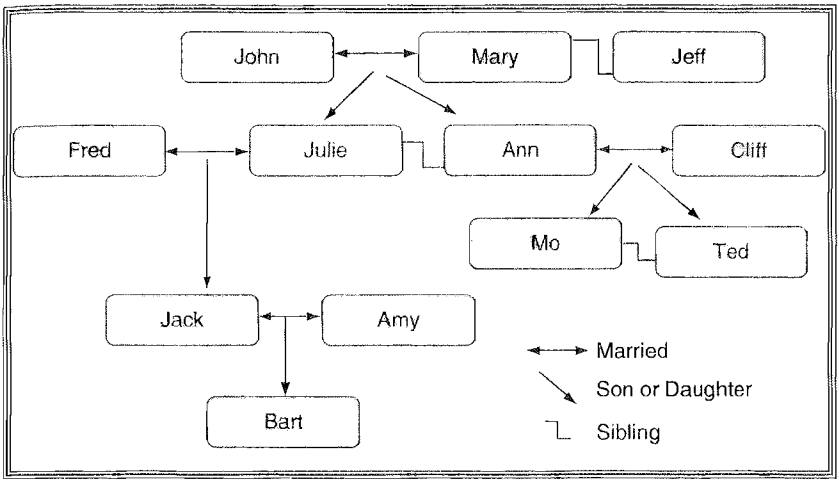

Figure 3c The family tree model used to determine familial links.

simulation is constructed out of symbols that are thereby serving as S-representations.

To get a better handle on how S-representation works in a CCTC system, it will help to step back and consider just how *we* might invoke similar sorts of representation to solve a problem. Suppose a person, Bob, is trying to determine whether two people are related, and if so, how. Bob knows many of the familial relations, but since the family is large he sometimes has trouble remembering how two people are related. So he gets a pen and a pad of paper and begins writing down the familial network. He does this by writing the name of each person, and then adding links to the names of other people, with each connecting link designating a specific type of relation (e.g., sibling, daughter/son, etc.). The result looks like the diagram in figure 3c. At times, Bob fills in blanks in his knowledge by making inferences about relations based on what he has already written (for example, he might come to realize that two people must be related in a way that had never before occurred to him). If two people are related in a certain way, then so and so follows, but if they are related in a different way, then something else follows. Eventually, Bob completes the diagram and then uses it to retrace the pertinent links and thereby establish how different people are related.

The manner by which Bob solves his problem is easy to see. He succeeds by constructing a *model* of real-world people and familiar relations which is then used to discover new facts. The relevant familial link between two

people is discovered by exploiting analogous links in the model. The representational elements (the written names and lines standing for people and their relations) of his diagram re-create the specific real-world conditions he is seeking to learn more about. Moreover, we can easily imagine Bob doing something similar when working through other sorts of problems, including those where the pertinent relations are not familial, but causal, spatial, mathematical, modal or any of a variety of other possibilities. For example, if Bob is trying to work out what repercussions he should expect in light of certain events, he can once again use a pen and paper and draw a diagram linking the relevant events, states of affairs, and possible consequences. This time, instead of representing familial relations, the lines and arrows may represent causal or entailment connections between different propositions. Or perhaps instead of linking the pertinent elements with lines drawn with labels, he simply uses "if-then" statements to represent the relevant entailment relations. He might use a sketch that winds up looking more like a lengthy logical argument than a schematic, pictorial diagram. But it will arguably still be a representational model that invokes elements that serve to mirror the conditions and states of affairs and entailment relations that Bob is trying to understand. There will still be a type of isomorphism between the sketch and the target that can be exploited to learn certain facts about the target.[6] And in such an arrangement, elements of the model perform a certain job – they serve as representations of particular elements of the target domain that is being modeled.

Returning to cognitive science, the basic point that is generally ignored by the Standard Interpretation is that the CCTC is, by and large, a framework committed to the claim that when the brain performs cognitive operations, it does the same sort of thing as Bob. Of course, the CCTC doesn't claim the brain uses pen and paper. Instead, it uses the neural equivalent of a buffer or short-term memory device and some sort of process for encoding neural symbols. But just like Bob's diagram, the symbol manipulations alleged to occur in the brain allow for problem solving because they generate a symbolic model of a target domain. That is, the symbol manipulations should be seen as the implementing of a model

[6] Of course, questions about more abstract mappings and increasingly obscure forms of isomorphism loom large. We can imagine gradually transforming a map so that it no longer resembles any sort of map at all, and yet it still somehow encodes all of the same information about the relevant terrain. I'm willing to be fairly unrestrictive about what qualifies as a map, model or simulation, as long as there is a clear explanatory benefit in claiming the system *uses* the structure in question as such. See also Palmer (1978) and Cummins (1989).

or simulation[7] which is then used to perform some cognitive task. The symbols themselves serve as S-representations by serving as parts of the model. As Cummins puts it, "Representation, in this context, is simply a convenient way of talking about an aspect of more or less successful simulation" (1989, p. 95).

For example, many production-based systems, like Newell's all-purpose SOAR architecture (1990), function by invoking a "problem-space" of a given domain and then executing various symbolic operations or "productions" that simulate actual real-world procedures, thereby moving the system from a representation of a starting point to a representation of some goal state. If the system is trying to re-arrange a set of blocks (imagine it controls a robot arm), then it executes a number of operations on computational symbols that represent the blocks and their positions. By manipulating these representations in a systematic way, determined by the SOAR's own procedural rules, the system succeeds in constructing a model of the world that it can then transform in various ways that mimic real-world block transformations. To make sense of all this, we cannot avoid treating the various data structures of the computational architecture as representations of elements of the relevant problem-space. As Newell puts it, "This problem space is useful for solving problems in the blocks world precisely because there is a representation law that relates what the operators do to the data structure that is the current state and what real moves do to real blocks in the external world" (1990, p. 162). This sort of "problem-solving-by-model/simulation" is at the heart of the CCTC style of explanation. These processes are a mechanized version of what Swoyer referred to as "surrogative reasoning."

An obvious complaint about the analogy between Bob's use of the diagram and what goes on in classical computational systems is that Bob mindfully interprets the marks of his diagram (thereby bestowing them with meaning) while, as Searle would argue, the computational system has no idea what its symbols means. Isn't it right to say that Bob is using a representational system in a way that Newell's computational device *isn't*, given that Bob – but not the computer – is assigning meaning to the symbols?

[7] There may be significant differences between a model and a simulation, but here I will use these two terms interchangeably. In other words, I won't assume that there is a significant difference between a computer model of some phenomenon like a hurricane, and a computer simulation of the phenomenon. Some might say that models are static representations whereas simulations involve a *process*, but it seems there are plenty of uses of "model" whereby it designates a process as well; indeed, a computer model is just such a case.

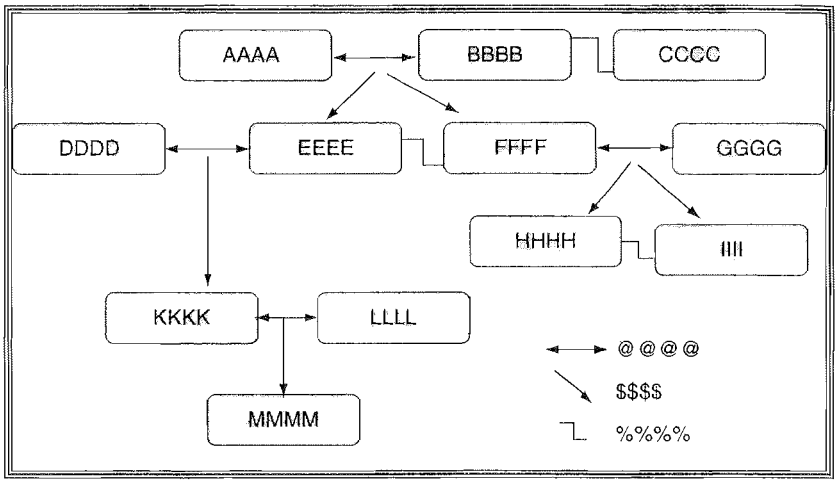

Figure 3d The opaque family tree model with meaningless symbols.

To answer this worry and get a better sense of the explanatory role of S-representation in the CCTC, we can consider what would happen to our explanation of Bob's problem-solving strategy if we were to substantially dumb him down and remove his own interpretive capacities. Suppose Bob doesn't understand what he is doing; the symbols are written in some language he doesn't comprehend, and he is simply following procedural rules that tell him such things as when to copy the symbols (by virtue of their shape) and when to look for matches. The diagram he is working with might now look like the diagram in figure 3d. A procedural rule might say, "If symbols AAAA and BBBB are connected by a line with arrows on each end, then put an X here." Bob has no idea that the letters stand for people, or that lines indicate different types of relationships. In this way, Bob becomes just like the man in Searle's Chinese Room – something that mindlessly manipulates syntactic structures in a manner that nonetheless generates solutions to some problem domain. Bob's use of the diagram becomes more like what might take place in a real computational device. The critical question now is this: Does making Bob more like an automated system substantially alter the representational role of the figures in his diagram? Or, alternatively, can we now explain and fully understand Bob's performance without ever characterizing the marks on the paper in representational terms?

On one reading of the question, what is being asked is whether or not Bob's sequence of operations is directly guided by the meaning or semantic character of the figures on his paper. With regard to this question,

the answer is famously "no." As we saw with the Chinese Room, the features of the symbols that allow mindless-Bob (or computational central processing units) to recognize, organize, arrange, etc. the symbolic structures are the non-semantic "syntactic" features – the symbols' shapes. It is by virtue of the shapes of the symbols (and the traced lines between those symbols) that mindless-Bob is guided though his various operations, and to understand the purely mechanical nature of those operations we needn't treat the symbols as representing anything.

At the same time, however, if told only that familial relations were discovered through focused attention to shapes and marks on paper, we would find this explanation of Bob's performance grossly inadequate. As we saw with the IO notion, we would still want to know how he was able to achieve success. We would want to be told what this arrangement of marks could possibly have to do with a familial connection between two people, and how it is that making marks on a piece of paper, and focusing on their shape, could lead to a discovery of that relationship. If told nothing more than mindless-Bob drew a diagram according to instructions, we would be replacing one mystery for another. The mystery of how mindless-Bob discovered a familial connection would be replaced by the mystery of how he discovered a familial connection by playing around with a diagram with distinctive shapes. Hence, there is more to understanding this process than simply describing the syntactic features of the diagram and how Bob responds to them.

This suggests a very different reading of the question we posed earlier. Instead of asking what features of the marks mindless-Bob uses to solve the problem, we can instead treat the question as asking what it is *about those* marks that, when used in that way, lead him to successfully perform the task in question. We are now asking *why* those markings eventually provide mindless-Bob with a solution when he uses them in accordance with the instructions. And the answer is that the marks on the paper do, in fact, accurately model the real-world family trees. Even when mindless-Bob fails to consciously interpret the marks on the paper, they are still serving as representations in a model that he has (unknowingly) built and is now exploiting. His scribblings on the paper help generate answers because those scribblings share a structural similarity to the relevant state of affairs he is investigating. We can't fully understand how mindless-Bob performs the operation of figuring out how two people are related unless we understand his operations as involving the implementation of a model. And to understand his operations as an implementation of a model, we need to look at the elements of these operations – in particular, the marks on the page – as representations of people and kinship relations.

In much the same manner, theories of the CCTC claim that to understand how the brain solves various cognitive tasks, we need to see it as implementing a model or simulation of the relevant target domain via a process of symbol manipulation. And to understand it as implementing a model via symbol manipulation, we need to treat the symbols themselves as representations of aspects of whatever is being modeled. Understanding how computers work involves understanding more than the nature of their physical operations. We also want to understand what it is about those specific operations that enable the system to perform some sort of task. We need to ask not only, "What is the causal/physical nature of this system?," but also, "What is it about the causal/physical nature of this system that enables it to solve this particular problem?" And the CCTC answer is this: These syntactic/physical operations are successful in solving this problem because they implement a model of a problem domain, and, as such, employ elements that *stand for* various aspects of that domain.[8] It is irrelevant that there are no components of the system that consciously interpret the symbols; that doesn't prevent the system from using some of its components as symbolic surrogates while running its simulations. The CCTC says that we should understand human cognition in exactly the same way. It claims that cognition should be understood as a process in which brains run simulations, and consequently employ representations of aspects of whatever domain is simulated.

Because the usual sense in which computational systems are said to do modeling is different than the sense I intend here, it is important to make the following distinction to avoid confusion. Computational systems and theories are themselves often regarded as providing models of brain processes. In this sense of cognitive modeling, the brain is the target of the modeling. But in the sense of modeling I am now speaking of, the brain is claimed by CCTC to be the "modeler," not the "modelee." That is, classical computational theories say that when the brain performs a given

[8] A number of people have suggested that I link the explanatory importance of S-representation too closely to the *success* of the cognitive system. But it is important to see that some degree of success is always presupposed in any specification of the cognitive task or capacity we are trying to explain. Without some success, we can't say that the cognitive operation we are trying to understand is actually instantiated by the system. After all, we don't say a rock does face recognition very, very poorly - we say it doesn't do face recognition at all. So the claim here is that one of the main explanatory goals of a cognitive theory is to explain how a given system (like the human brain) performs a cognitive task, and that requires assuming that it actually does perform that task, which in turn requires assuming that it performs it at least somewhat successfully. S-representation is needed for achieving this explanatory goal because it enables us to see how an internal structure is functioning as a model or simulation that enables certain systems to perform the operation in question.

cognitive task, the brain itself constructs a model of the relevant domain, and consequently uses representations of aspects of that domain as elements of the model. Cognitive models (in the usual sense) are models of what, according to the CCTC, is *itself* a modeling process. In effect, the computer model of the mind claims that the brain is actually doing what ordinary computers often do when they run simulations of various real-world processes.

Hence, it should be clear how, on this conception, brain states that are posited as part of a computational process (brain states that function as data structures) actually *serve as* representations in such a process. They do so by serving as constituent elements of a model or simulation that is exploited by the system when doing some cognitive task. In this context, "standing for" amounts to "standing in for," since problem-solving as model-building (or simulation-running) makes little sense without component elements of the model (or simulation) that function as representations. Haugeland captures the basic idea when he says, "[t]hat which stands in for something else in this way is a *representation*; that which it stands in for is its *content*; and its standing in for that content is *representing* it" (1991, p. 62). Or, to adopt Swoyer's language, computational systems (and, ex hypothesis, the brain) perform a type of mechanized surrogative reasoning. Surrogative reasoning requires surrogates, i.e., representations, and in computational accounts that job description goes to the symbolic data-structures. The content of the symbols is explanatorily relevant for their job because if the symbols don't stand for anything, the system in which they function can't itself serve as a model or simulation of the target domain, and we would have to abandon the central explanatory strategy offered by the CCTC. The job description challenge is successfully met with S-representation because we are provided with a unique role that is recognizably representational in nature and that fully warrants our saying the state serves a representational function. Moreover, this role of serving as a stand-in for some aspect of a target domain in a computational model or simulation is sufficiently distinctive and special to allow us to distinguish the representational elements of a system from the non-representational.

Besides answering the job description challenge, the functional role associated with S-representation allows us to account for other intuitive aspects of representation. For example, while a variety of factors may be necessary for a fully satisfactory account of S-representation content, it is clear that one significant factor will be a symbol's functional "place" in a model or simulation. If we ask how the tail section of a model plane in a wind tunnel comes to represent the tail section of a real plane – how the tail

section of the real plane (and not the front nose section) comes to be the intentional object of the model's tail – the answer will appeal to the way the model is structured, how that structure leads to a certain kind of isomorphism with the real plane, and how that isomorphism maps one element of the model to an element of the target. Thus, S-representational content is linked to the sort of role the representation plays in the system's overall problem-solving strategy. In the CCTC-style explanations, the organization of the model and the nature of the resulting isomorphism with the target determines, in part, what it is that a given component of the model or simulation represents. Moreover, we get a sort of misrepresentation when the isomorphism breaks down. If the wingspan of the model tail wing is disproportionately longer than the wingspan of the actual tail, then that aspect of the model is in error and misrepresents the tail section of the real plane. Misrepresentation is a case of inaccurate replication of the target domain. Inaccurate replication occurs when and where the model or simulation fails to maintain its organizational (or structural) isomorphism with that which is being modeled.

In our discussion of the IO notion, we noted that there is a sort of mutual dependence between something having the function of serving as an IO representation, and a sub-system having the function of performing some internal computation. A similar sort of mutual dependence exists with S-representation. S-representations are necessary for a system's construction of a model *and* it is a state's participation in a model that makes it an S-representation. The constituent representations help make the embedding structure a model or simulation, and it is the embedding structure's status as a model or simulation that makes the constituent elements representations. This may initially appear to be a vicious circle. But remember that S-representation is a functional notion; hence, to do their job S-representations need to be embedded in the right sort of system. But such a system (i.e., a modeling or simulating system) isn't possible unless there are structures serving as S-representations. Thus, we have the same sort of mutual dependence that one often finds with functional kinds. Something must play a certain role in a system; but the relevant system itself isn't possible without components playing such a role. A person is a soldier only by virtue of being a member of an army. But there can be no army without members serving as soldiers. So soldiers and armies come together as a package. Similarly, something is an S-representation by virtue of the role it plays as part of a model of the target domain. But something needs to play that representational role (the role of standing for specific elements of the target) for there to actually be such a model. Models

and simulations require S-representations to exist and nothing is an S-representation unless it functions as part of a model.

The explanatory value of S-representation becomes clearer if we consider how this notion offers an avenue of rebuttal to the anti-representational challenges posed by Searle and Stich. Since the case of mindless-Bob is simply a variant on the Chinese Room, we have already seen how the notion of S-representation comes into play under the sort of conditions Searle's argument exploits. Ex hypothesis, the room produces appropriate outputs in response to the inputs it receives. Thus, a computational account of the room would need to include an explanation of how it consistently does this. Syntactic symbol manipulations are only part of the story – we also want an explanation that tells us what it is about those manipulations that produces continued success (despite the ignorance and lack of understanding on the part of the manipulator). Depending on the details, one answer proposed by CCTC theories is that those manipulations model some target domain, and thus involve S-representations. In fact, if the program used by the Chinese Room is like the sort of architecture that inspired Searle's argument, such as Shank and Abelson's (1977) SAM (for "Script Applier Mechanism"), then its success would clearly involve models and hence S-representations. Shank and Abelson's theory uses "scripts," which are stored data structures that serve to symbolically replicate some state of affairs, like the central features of riding a bus or eating at a restaurant. As models of those activities, they allow the system to answer various questions which can be generated from the stored "background knowledge" in the script. If the symbols in the Chinese Room constitute scripts of this sort, then they serve as representations, not because there is some conscious interpreter who understands them as such, or because the people who designed the system intended them that way, but because the overall system succeeds by exploiting the organizational symmetry that exists between its internal states and some chunk of the real world.

Of course, the details of any specific account are less important than the core idea that classical theories invoke models and simulations and thereby invoke S-representations. Is S-representation comparable to full-blown conscious thoughts? No, it is a technical notion of representation based on our commonsense understanding of things like maps, invoked by a theory to explain cognition in a certain way. Searle is correct that the account of cognition offered by the CCTC fails to present a notion of representation that captures all aspects of our ordinary, commonsense understanding of thinking and thought. Nothing in the CCTC should

lead us to conclude that Searle is wrong in asserting that the Chinese Room, as such, does not instantiate a full-blown mind. But the issue of whether or not full-blown minds could be instantiated by any system running the right program can be separated from the question of whether or not the CCTC provides a representational theory of how the brain works. What should *not* be conceded to Searle is the proposition that the CCTC fails to invoke *any* explanatorily valuable notion of representation. It should not be conceded that the only sense in which classical symbols serve as representations in computational processes is the artificial "as if" sense that is only metaphorical and has nothing to do with real representation. What the CCTC shows us is that a notion of representation can do explanatory work, *qua* representation, even in a purely mechanical problem-solving system.

Similarly, our earlier discussion revealed what Stich's purely syntactic account of computational processes would leave out. The question of why a system works is every bit as important as *how* it works. But a syntactic approach would largely ignore the former question. A syntactic story would reveal the process whereby the symbols come to be shuffled about in various ways. But it would not tell us what it is about those symbol shufflings that leads the system to consistently produce the appropriate responses. It would ignore the central aspect of the classical account that answers the question, "why do *these* syntactic operations enable the system to perform as well as it does?" A purely syntactic account would leave us blind to the fact that computational systems are doing surrogative reasoning because it would prevent us from seeing that computational structures serve as representational surrogates.

In fact, S-representation reveals a weakness in one of Stich's main arguments for the syntactic theory. Stich suggests scientific psychology should be guided by the "autonomy principle," which holds that "any differences between organisms which do not manifest themselves as differences in their current, internal, physical states ought to be ignored by a psychological theory" (1983, p. 164). In defense of this principle, he offers what he calls the "replacement argument." Since cognitive psychology is in the business of explaining the inner workings of the mind/brain, any historical or environmental dissimilarities between an original and physical duplicate that fail to generate actual causal/physical dissimilarities should be ignored by cognitive (and, by extension, computational) psychology. If a robot on an assembly line is replaced with an identical duplicate, then, Stich argues, cognitive psychology should treat the original and the double as the same. The same goes for human agents. But since the content of the agent's

belief-type states *is* based upon historical or environmental factors, then a psychology that pays attention to content will treat the human duplicate as different from the original, thereby violating the autonomy principle. Given the intuitive plausibility of the autonomy principle, Stich takes this to show that computational psychology ought to *ignore* content and drop the notion of representation altogether.

The problem with this analysis is that it is based upon a conception of representation in the CCTC that is too narrow. Stich's argument at least tacitly adopts the Standard Interpretation and thereby treats computational representations as analogues for propositional attitudes. For propositional attitudes, content is arguably due entirely to external, causal-historical facts, and thus physically identical systems may differ in terms of folk mental representations. By contrast, S-representation carries the possibility of understanding content in a way that is far less dependent upon causal-historical details. Above I've suggested that S-representation stems from the use of an inner model in some cognitive task that is properly isomorphic with its target. Because any replica placed in the same situation will (by virtue of being a replica) employ internal structures functioning in the same manner as the original, it is also presumably using the same sort of model and thus the same sort of S-representations. If, say, a robot that is using an inner map to successfully maneuver in some environment is replaced by a physically identical system, then the explanation of how the duplicate performs the same task would also need to appeal to the same sort of inner map. The use of a map is not so directly dependent upon the history or source of the map, and it would be bizarre to claim that the performance of the original robot involves inner elements that cannot be invoked when we explain the success of the physically indistinguishable duplicate in the same situation. So, intuitively, the same S-representational account would apply to the replacement that applied to the original. Thus, S-representation passes the replacement test and thereby satisfies the autonomy principle. If the replacement argument is intended to show how the syntactic approach to the CCTC is superior to a representational approach, then it fails to do so once we appreciate the importance of S-representation to CCTC explanations of cognitive processes.

By suggesting that there is a theory-based notion of representation that is built into the explanatory framework of the CCTC and that accords with the autonomy principle, I do not mean to suggest that all external, environmental factors are completely irrelevant. In the case of the duplicate robot, I'm suggesting its *use* of a model (and thus S-representations) depends upon its performance of a specific task, and its performance of a specific

task depends upon the circumstances in which the system is embedded – on the problem-solving environment. I'm suggesting that while the causal history of a map or model is (perhaps) irrelevant to its current usage, the specific environment in which it is employed may be highly relevant to questions about content. In the next section, I'll argue that the task environment helps determine how a posited model is used, which in turn determines, in part, what S-representations represent. In short, the task-environment a model is plugged into helps determine the model's target, and the model's target helps determine the content of S-representations. This arrangement suggests a possible solution to a traditional problem associated with S-representation.

This barely scratches the surface of all of the different dimensions and worries connected to S-representational content,[9] some of which will be further addressed below in sections 3.3 and 3.4. As with the interior IO notion of representation, the S-representation notion requires a more detailed and sophisticated elaboration than I can provide here. But remember my aim is to only show that there is a notion of representation at work in CCTC theories that answers the job description challenge by describing structures playing a functional role that is both naturalistic and recognizably representational in nature. Both the IO notion and the S-representation notion do this. And yet both notions are a bit like the Rodney Dangerfields of representational posits – they get no respect, or at least not as much respect as they deserve. Much of this is because of two traditional problems that are often assumed to undermine their theoretical value. Since I think these problems are overblown, it will help to look at them more closely.

3.3 TWO OBJECTIONS AND THEIR REPLIES

All notions of representation have their difficulties, and the two notions which I have argued are central to the CCTC explanations are no exception. In this section I would like to address what I take to be the two most common criticisms of these notions. One charges that mere isomorphism does not provide a sufficiently determinate sort of representational content, and thus the notion of S-representation is critically flawed. The other criticism suggests that these representational notions are really too weak and make representation nothing more than a heuristic device. My aim will be to demonstrate that a better understanding of the explanatory work that

[9] For a much fuller discussion of these issues, see Cummins (1989, 1996) and Horgan (1994).

these notions are doing in the CCTC reveals that the criticisms are much less damaging than they initially seem.

3.3.1 Challenge 1: indeterminacy in S-representation (and IO-representation) content

A problem for S-representation in particular (but also could be developed into a complaint about IO-representation) is one that we've already briefly touched on. As we've seen, the notion of S-representation is based upon some sort of isomorphism between the model or simulation and the target being modeled or simulated. But, notoriously, isomorphisms are cheap – any given system is isomorphic with many other systems. For instance, in the non-mental realm, maps provide an example of S-representation whereby the individual figures and lines on the map stand for specific aspects of an environment by virtue of the overall isomorphism between the map and that environment. But in truth, the collection of lines and figures on a simple map can be equally isomorphic with a range of different environments and geographic locations. Hence, which parts of the world they *really* designate cannot be determined by appealing to isomorphism alone. Or returning to the computational realm, Fodor and others have emphasized that two systems simulating different events – one, say, the Six Day War, the other a chess game – could be functionally identical, so that "the internal career of a machine running one program would be identical, step by step, to that of a machine running the other" (1981, p. 207). Consequently, if presented with such a program, there would be no fact about whether the inner symbols S-represent the Sinai Peninsula and Egyptian tanks, or chess board locations and pawns and rooks. So it looks like the S-representation notion has a serious problem of content indeterminacy. Something is a representation by participating as part of a simulation or model, which in turn is a simulation or model *of* that aspect of the world with which it is isomorphic. But models and maps are isomorphic with many different aspects of the world, so the representation is potentially about a wide array of things. Simply mirroring the organizational configuration of some state of affairs is not sufficient to make something a model or simulation *of* that state of affairs, even if we limit the amount of complexity we build into our interpretation scheme. The target of any model is inherently indeterminate, and thus the content of any element of that model is indeterminate. But actual representations have determinate content, so being an element of such a model or simulation is not enough to make something a representation.

Yet I believe this verdict is too quick. For those invoking representations as part of the CCTC, the indeterminism worry is, I believe, a red herring that stems from a failure to appreciate the nature of the explanatory challenge facing the cognitive theorist. The explanatory challenge is typically not to explain *what* a cognitive system is doing, but *how* it is doing it. Competing theories of cognition, including the classical paradigm, offer hypotheses about inner processes that are designed to account for our various mental abilities. These abilities are thus taken as a given, as a starting point for our theorizing. The explanatory challenge can be characterized in this way: "Given that we successfully do such and such (recognize faces, maneuver through a complex environment, determine the grammaticality of sentences, etc.), provide an explanation of how our brains do it." So the CCTC starts with a specified cognitive ability, and then offers an explication of how that ability is realized in the brain. In the process, the CCTC posits S-representations that play a certain sort of functional role. In the case of mindless Bob, that role was instrumental in helping him to answer various questions about a given family, despite Bob's ignorance of what he was actually doing. Even though Bob isn't interpreting his model, ir still makes perfectly good sense to explain his success by saying he is using a model, and moreover, a model of one particular family. It doesn't really matter that the drawing is isomorphic with dozens of other family trees or, for that matter, dozens of other states of affairs. It is used as a model of *this* family, and not some other, because this is the family that is the subject of Bob's problem-solving efforts. In other words, the problem domain itself – the situation that Bob finds himself in, and for which he (mindlessly) employs his diagram – is what makes determinate the target of his model, and thus the content of the representations that make up the model. The indeterminacy problem arises when there is no way to fix the target of a model. In cognitive explanations, however, the explanandum itself typically *does* fix the pertinent target and thereby determines what it is that is being modeled. Looked at another way, if the brain is indeed performing a specific cognitive task, then a classical computationalist gets to posit representations with determinate content because she gets to posit models and simulations used in the performance of *that* task (and not some other). If the true explanation of how my brain succeeds in some mental task is that it relies on a model, it simply doesn't matter that, taken as an abstract structure, the model is isomorphic with other things. It may be, but that doesn't undermine the way my brain is using it here.

Of course, we've now traded one sort of problem for another. The original problem was determining what a simulation or model is a simulation

or model of. I'm claiming this can be settled by looking at how the model is actually being used in specific problem-solving situations – by looking at the actual task for which the structure is used. But that just shifts the problem of determining the target of the model to one of determining the exact nature of the task the system is performing. What warrants my saying that Bob is constructing answers to questions about a family, and not doing something else?

Well, lots of things. First, it is important to see that this is an issue that we are going to need to address no matter what our theory is. Specifying the nature of the cognitive task a system is performing, while a deep and thorny topic, is a deep and thorny topic that everyone needs to confront – it is not a problem that is unique to accounts that appeal to S-representation. Second, there are various promising strategies for addressing this issue. The mosr obvious builds on the fact that representational systems do not operate in a vacuum; they are embedded in the world in various ways, and this embedding helps determine the actual tasks they are attempting to perform. Consider ordinary psychological explananda. There is no deep mystery in trying to decipher the task a rat is performing as it attempts to make its way through a maze for some food. It is trying to navigate its way through a maze. Which maze? The one it actually finds itself in. If a theory says the rat is using an internally stored map for navigation, then the map is, by virtue of that very use, a structure that is used to model this particular maze. It simply doesn't matter for our explanatory purposes that there are lots of other mazes or terrains in the world that the map is isomorphic to. So, by looking at the way representational systems are embedded in the world, we can specify the particular tasks they are confronting. And by specifying the particular tasks they are confronting, we can specify the particular target of whatever models or maps the system is using for that task. And by specifying the targets of whatever models or maps that are being used, we can specify what it is that the elements of those models and maps actually stand for. A model's constituent parts stand for those things they are used to stand *in* for during actual episodes of surrogative reasoning.

In his more recent writings, Cummins (1996) has suggested that physical structures represent all things with which they are isomorphic and, thus, representations never have a fixed, determinate content. The way I suggest we understand S-representation is quite different. Parts of a model don't automatically represent aspects of some target *simply* because the model is isomorphic to the target. Rather, components of the model *become* representations when the isomorphism is exploited in the execution of

surrogative problem-solving. The CCTC claims that the brain employs representations because the brain uses some sort of a model of the target (or some aspect of the target) and neural states serve as representational parts of that model. Yet it's possible the same basic computational model or simulation could be isomorphic with some other target, and therefore could be used in the execution of some other cognitive task, in some other problem-solving situation. But that possibility doesn't matter because that's not what *this* brain is using it for *now*. The neural symbols really do stand for, let's say, board positions of a chess game, and not the positions of armies in the Sinai Peninsula, because what the theory is about is a cognitive agent playing chess and not fighting a war. Given that the agent is playing chess, classical computationalists can say he is doing so by running simulations of possible moves – simulations that are comprised of representations. A cognitive agent is figuring out chess moves and not battle strategy for the Six Day War because the agent is causally linked to a chess game and not a battlefield.[10] Thus, the content of S-representation can be fixed by the target of the model, and the target of the model is fixed by the cognitive activity we want explained. The cognitive activity we want explained, moreover, is typically dependent upon the way the system is currently and causally engaged in the world. The upshot is that the content indeterminacy problem is simply not as big a challenge for S-representation as is generally assumed.[11]

3.3.2 Challenge 2: IO-representation and S-representation aren't sufficiently real

Throughout this discussion I've made repeated appeals to the *explanatory benefit* of positing representations, or to the *explanatory pay-off* of invoking IO and S notions of representation. Moreover, I've defended both notions by insisting that there actually is such a pay-off. In the case of the

[10] Of course, the same sort of causal considerations could *not* help us define mathematical cognitive processes or different forms of abstract or hypothetical reasoning in which we have no clear causal connection to the target domain.

[11] In my analysis, I've deliberately avoided appealing to historical factors or to the way the cognitive map or model is constructed when specifying the map or model's target. This is because I believe that what a map or model targets is intuitively more dependent upon how it is used than on where it came from. I believe, say, that a map that is used to navigate a particular terrain is serving as a map *of* that terrain, even if it was originally written as a map of some other environment. But others may find this implausible and believe that diachronic considerations are indeed key to determining a map or model's target. This would provide another strategy for handling the indeterminacy problem, and would support my main point that the problem can indeed be handled.

IO-representation, the notion allows us to employ an explanatory strategy of functional analysis whereby the inner sub-systems can be seen to perform tasks germane to the larger explanandum. In the case of S-representation, the notion allows us to treat the system as employing a model or simulation, which in turn helps us to explain how it succeeds in performing a given cognitive task. However, this emphasis upon the explanatory role of representations has a down side. It suggests that a structure's status as a representation is entirely dependent upon our explanatory interests and goals – that things are representations only insofar as we gain some explanatory benefit from treating them that way. This implies that the notions of representation under consideration serve as something like heuristics or useful fictions, similar to a frame of reference, or the proverbial family with 1.5 children. As such, they don't correspond to anything objectively real in nature. IO- and S-representations would exist only to the extent that we take a certain explanatory stance toward a system, and without the projection of our intentional gloss, there would be no such sorts of things.[12]

The philosopher who has been the strongest advocate of the view that the having of representations depends on our trying to explain or predict the behavior of the system is Daniel Dennett (1978, 1987). In our efforts to understand any given complex (or even simple) system, there are, according to Dennett, different explanatory stances or strategies that we can adopt. First, we can adopt the "physical stance" and use an understanding of the physical inner workings of the system to explain and predict how it responds to different inputs. Or, alternatively, we can adopt the "design stance" and predict behavior by using what we know about the sort of tasks the system was designed to perform. Finally, we can sometimes adopt what he calls the "intentional stance." The intentional stance involves treating a system as a rational agent with beliefs, desires and other folk representational states. Dennett has argued extensively that being a "true believer" is little more than being a system whose behavior can be successfully explained and predicted through the ascription of beliefs and other propositional attitudes. If we can gain something by treating a Coke machine as having, say, the thought that it has not yet received enough money, then the Coke machine really does have such a thought.

[12] Michael Devitt has put it to me this way: "It is as if you don't think representation is a REAL PROPERTY of anything; it's just a matter of when it's appropriate to take the 'intentional stance' toward it" (personal correspondence).

Despite his extensive and often inventive arguments for this perspective on intentional states, few philosophers or cognitive scientists have adopted Dennett's interpretationalist account. For any given system, Dennett closely ties the possession of mental representations to the explanatory activities of other cognitive agents – on the sort of explanatory perspective they adopt. Yet most people think these are the wrong *type* of considerations for identifying representational systems because mental representations are regarded as objectively real, observer-independent aspects of the world. The criteria for being a representation should be of the same nature as the criteria for being a kidney or a virus; namely, actual intrinsic or relational properties that are either present or not, regardless of how the system could be explained by others. Returning, then, to my analysis of computational representations, the complaint is that the notions we've explored in this chapter are overly "Dennettian." It appears that, like Dennett, I've offered an analysis whereby the positing of representations stems from the explanatory strategies and goals of cognitive scientists who need them to adopt certain perspectives on a proposed system's operations. It seems these notions of representation serve as representations not for the system, but for the psychologists attempting to understand the system in a certain way. Yet there is surely more to representation than that. If the CCTC is a framework that invokes *real* representations, then it needs to do so in a way that is far less observer-dependent and far more objectively real than I've suggested here.

My response to this challenge is that the notions of representation we've looked at here *are* fully objective and observer-independent, and any appearance to the contrary is simply an artifact of my emphasis upon the explanatory role of representations in science, and not a deep fact about their metaphysical status. Indeed, when we step back and look at how theories in the CCTC invoke representational states, we can see they have the same status as other scientific posits whose existence is assumed to be objectively real.

To begin with, it should be acknowledged by all that there is a very weak sense in which most of the things we deal with in our daily lives are observer-dependent. The sense I have in mind is just this: It is at least theoretically possible to view any system as nothing more that a cloud of interacting molecules or even atoms; hence, our *not* viewing the system in that way requires our adopting a certain stance. In this weak and uninteresting sense of "stance dependence," any notion that is not part of the vocabulary of basic physics can be treated as unreal or "merely heuristic." Yet it shouldn't bother the advocate of the notions of representations

presented here if it should turn out that the representations are observer-dependent in *this* sense. In other words, if the argument that representations are observer-dependent is simply that it is *possible* for us to view a computational system as nothing more than a bunch of interacting molecules (that is, that it is possible to adopt the physical stance), then this way of being observer-dependent shouldn't cause us concern. If IO and S-representations are unreal only in the sense in which trees, minerals, hearts, mountains and species are unreal, then a realist about representation should be able to live with *that* sort of "anti-realism."

We can therefore assume that the anti-realism challenge alleges that the IO and S notions of representation are observer-dependent in a stronger sense than this. But it is much harder to see how a stronger sense of observer-dependence applies to these notions. Consider again the way in which the IO and S notions of representation are invoked. The CCTC is a general account of how the brain works; more specifically, it is a theory about how the brain performs various cognitive tasks. It claims that the brain does this by performing symbol manipulations of a certain sort. In many (or even most) versions of the theory, the symbol manipulations attributed to the brain are those that involve a) inner sub-systems that perform various sub-tasks that are stages of the task being explained and/or, b) models or simulations of the relevant target domain. Both of these types of operations require representational states. The sub-systems employ representations because the sub-tasks convert representations relevant to one aspect of the cognitive tasks into representations of another aspect. And the models employ representations because the components of all models stand in for different elements of whatever is being modeled. So, if the CCTC is the correct theory of how the brain works, then the brain really uses inner representations.

Now it is not at all clear where in this account an anti-realist or observer-dependent interpretation of representation is supposed to arise. While it is true that the account links the having of representations to other things the system is doing (namely, using inner sub-systems and models), it is unclear how this alone is supposed to make representations useful fictions. There are, it seems, only two possible strategies for arguing that the CCTC leads to a sort of anti-realism. The first would be to drive a wedge between the status of the representation and the sorts of processes that the CCTC invokes to explain cognition. On this scheme, one might claim that inner sub-systems and models don't actually require representational states; hence, if we treat structures as representations, we are simply adopting the intentional stance for heuristic or instrumental reasons. The second way would be to concede that sub-systems and models require

representations, but to then argue that the sub-systems and models themselves are also observer-dependent and therefore not sufficiently real. Yet neither of these strategies is terribly compelling.

With the first strategy, it is difficult to see how the argument could even get started. There is no discernible way that something could serve as, say, an adder, without it also being the case that it converts representations of numbers into representations of sums. Without such a conversion of representations, it simply wouldn't be doing addition. So too for countless other tasks that inner sub-systems are routinely characterized as performing. Along similar lines, it is hard to see how anything could employ a model or simulation of some target domain, and yet, at the same time, not have it be the case that the individual elements of the model or simulation stand for aspects or features of the target. If we are committed to the reality of the kind of processes classical computationalists claim are going on in the brain, then we are committed to neural structures really serving as inner representations. Of course, one could argue that brains don't actually implement the sorts of processes described by the CCTC. But that would be to argue that the CCTC is false – not that it employs an observer-dependent notion of representation.

The second strategy is to allow that representations are as real as the processes in which they are embedded, but to then argue that those processes themselves are useful fictions, interpretation-dependent, or subjective in some similar sense. With this view, the brain is not really employing inner sub-systems that perform computations which are stages of larger cognitive tasks; or, alternatively, the brain is not using models or simulations in any sort of objective sense. These are just subjective interpretations of physical processes. Indeed, this challenge could be seen as a more general worry about the very nature of computational processes or simulations themselves. The writer who is best known for this sort of criticism of computation is, once again, Searle (1990, 1991). Searle originally allowed that the Chinese Room, though lacking symbols with real meaning, was at least performing syntactic operations on formal tokens. He has since reconsidered this matter and now holds that computational systems and syntactic processes also fail to exist in any robust, objective sense. He tells us,

> Computational states are not *discovered within* physics, they are *assigned* to the physics ... There is no way you could discover that something is intrinsically a digital computer because the characterization of it as a digital computer is always relative to an observer who assigns a syntactical interpretation to the purely physical features of the system ... to say that something is *functioning as* a computational process is to say something more than that a pattern of physical events is

occurring. It requires the assignment of a computational interpretation by some agent. (1990, pp. 27–28)

Unlike the Chinese Room argument, Searle's argument for the position that computational processes are observer-dependent is somewhat hard to discern. In fact, Searle's discussion appears to provide us with less in the way of an argument and more in the way of alternative characterizations of his conclusions. In spots, when Searle tells us that "syntax is not intrinsic to physics" (1990, p. 26) and "syntax is not the name of a physical feature like mass or gravity" (1990, p. 27) it sounds as though he is defending the uninteresting view discussed above, that anything that is not described using the terminology of basic physics is observer-dependent. In other spots, Searle muddies the water by lumping together user-dependence and observer-dependence. Yet if the brain uses computational programs in the same sense in which, say, our body uses an immune system, this notion of use would be fully objective (after all, chairs may be sitter-dependent, but this doesn't make chairs observer-dependent). At one point, Searle tells us that "on the standard definitions of computation, computational features are observer relative" (1991, p. 212). But he doesn't tell us exactly how the "standard definitions of computation" lead to the view that differences in computational systems are in the eye of the beholder. There really is no generally accepted principle of computation that would rule out the possibility of distinguishing computational systems or programs by appealing to their causal/physical architecture, or that would entail that all computational processes are objectively indistinguishable, or that would suggest that there is no observer-independent sense in which my laptop is using Wordstar but the wall behind me is not. So on the one hand, there is a sense of "observer-dependent" that applies to computational systems and processes. But it is a completely uninteresting sense in which virtually everything is observer-dependent. On the other hand, there is a more interesting sense in which things might be observer-dependent – like being a funny joke. But as far as I can tell, Searle hasn't given us an argument that programs are observer-dependent in *that* sense.[13]

[13] Searle's view is also puzzling given his own claims about the value of weak AI. Since the sort of program a system would be running would be a matter of interpretation, there wouldn't be any objective fact about the quality or even the type of program implemented on a machine. Any problems that arose could be attributed not to the program, but to our perspective, so the difference between a good computer simulation of a hurricane and a bad one, or, for that matter, between a simulation of a hurricane and a simulation of language processing would all be in the eye of the beholder. It is hard to see how such an observer-dependent set-up could serve to *inform* us about the actual nature of various phenomena.

Since my goal is not to defend the CCTC but to instead defend the idea that the CCTC makes use of valuable and real notions of representation, perhaps the appropriate way to handle this worry is as follows. If you want to claim that the notions of representation discussed in this chapter are observer-dependent fictions, then you must do so at a very high cost. You must also adopt the view that computational processes themselves are observer-dependent, and this has a number of counter-intuitive consequences. For example, such an outlook would imply that a pocker calculator is *not really* an adder, computers *don't actually* run models of climate change, and no one has ever *truly* played chess with a computer. It would imply that whether or not your lap-top is infected with a virus or running a certain version of Windows is simply a matter of your perspective – that the distinction between different programs is no more objective than the distinction between images seen in ink-blots. Moreover – and this is the key point – you would be denying that we could discover that, at a certain level of analysis, the brain really does implement symbolic processes of the sort described in the CCTC, particularly those that appeal to inner sub-systems and models, and does so in a way that it might not have. Since this strikes me as a radical and counter-intuitive perspective on the nature of computation, the burden of proof is on someone to make it believable. As far as I can see, this hasn't been done.

3.4 CCTC REPRESENTATION: FURTHER ISSUES

No doubt many will find this analysis of CCTC notions of representation incomplete, which, in many respects, it is. But recall that our goal has been fairly modest. It has not been to provide a complete theory of representation that solves all problems associated with a naturalistic account of representation. Rather, it has been to defend the idea that the CCTC posits representations with considerable justification. It has been to show that IO-representation and S-representation are sufficiently plausible, robust, and explanatorily valuable notions to warrant the claim that, contrary to what the Standard Interpretation might lead one to think, the CCTC is indeed committed to internal representations. Nonetheless, despite these limited goals, there are further issues associated with these concepts of representation that warrant further attention.

3.4.1 Is IO-representation distinct from S-representation?

While Cummins's 1989 account of computational representation has served as the basis for much of the analysis provided here, my account

departs from his in two important respects. First, Cummins treats the *exterior* IO notion of representation as the central notion at work in the CCTC, and fails to say much about the interior IO notion. Moreover, Cummins argues that the same notion is at work in connectionist accounts of cognition, as networks also convert representational inputs into representational outputs (1989, 1991). According to this view the CCTC and connectionist theories actually employ the same notion of representation. I believe this is a mistake. The error stems from treating exterior IO-representations as *part of* either theory's explanatory machinery. As I argued above, the exterior notion generally serves to frame the *explanandum* of cognitive science. That is, the typical cognitive task we ask a theory to explain is characterized as a function whereby input representations of one sort are converted into output representations of another sort. Thus, these exterior representations are not so much a part of the explanatory theory as they are a part of the phenomenon we want explained. This is not the case for the notion of *interior* IO-representations. These actually do form part of the distinctive explanatory machinery of the CCTC because classical theories often explain cognition by appealing to a hierarchically organized flow-chart. Since the internal sub-routines require their own inputs and outputs to be treated as representations, the interior IO notion becomes a notion of internal representation that is, by and large, unique to the CCTC.[14]

The second area where my analysis has departed from Cummins's original treatment concerns my distinction between two sorts of notions of representation at work in the CCTC. Whereas Cummins appears to treat the inputs and outputs of computational processes as S-representations, I've chosen to separate these as two distinct notions doing different explanatory jobs. It is fair to ask if this is the right way to look at things: if there really are two separate notions at work, as opposed to just one.

The reason I distinguish IO-representations from S-representation is because I am, recall, demarcating notions of representation in terms of the sort of explanatory work they do. Putting this another way, I am distinguishing notions of representation in terms of the way they actually *serve as* representations according to the theory in which they are posited. My position is that the way in which a structure serves as an interior IO-representation is different from the way it serves as an S-representation. In the case of the former, the job is linked to an internal sub-module or

[14] Of course, there are a number of elaborate connectionist networks that also invoke inner subsystems, and thus also employ the interior IO notion. Yet as I've noted in other works (Ramsey 1997), this is not the standard notion of representation that appears in connectionist modeling.

processor performing computations relevant to the overall capacity being explained. Such an inner sub-system typically receives representations as inputs and generates representations as outputs, so that is how representation comes into the explanatory picture. In the case of S-representation, the story is quite different. There the job of representing is linked to the implementation of a model or simulation, which requires components that stand for the relevant aspects of the target domain. Thus, the explanatory appeal to representation is quite different than it is with the IO notion; in fact, it is quite possible to have one without the other. For example, we could have a cognitive system that is explained with a task-decompositional analysis invoking inner sub-systems transforming IO-representations, but that makes no use of a model or simulation (an organized cluster of connectionist networks might be such a system). Or, alternatively, there could be theories that use inner models or simulations of some target domain but that don't appeal to inner sub-systems that require representational inputs and outputs (some simple production systems might have this feature). Consequently, the two notions of representation are distinct and should be treated as such.

Of course, this is not to say that computational structures never play *both* representational roles. In CCTC accounts, data structures can serve as both IO-representations and S-representations. This might happen whenever the simulation involves sub-computational systems that serve as stages or segments of the model or simulation. For instance, in our original multiplication example, the internal addition sub-system would be part of a simulation of a mathematical algorithm in which numbers are multiplied via addition. The data structures generated by the adder represents sums *both* because they are produced by an adder and because, as such, they are part of the simulation of a mathematical process (a type of multiplication) that involves sums. In fact, many would claim that *all* numerical computations are simulations of various mathematical functions. If this is true, then the IO notion could be reduced to a special type of S-representation for these types of computational operations. It wouldn't follow that all IO-representations are special cases of S-representation, or that there aren't really two different explanatory roles associated with these two notions. However, I would not be surprised if a more detailed analysis revealed that a large number of CCTC models posited representational structures that do double duty in this way.

3.4.2 Cummins's abandonment of S-representation

As we noted, the idea that the CCTC framework typically invokes S-representations is at least partly due to the analysis provided by Cummins,

as presented in his 1989 book *Meaning and Mental Representation*. In more recent writings, however, Cummins appears to claim that this notion of representation, at least as originally presented, is severely flawed (Cummins, 1996). While there is much that could be said on this topic, I briefly want to consider Cummins's reasons for rejecting S-representation as an account of computational representation to see if his new position undermines our analysis.

In his 1996 book, *Representations, Targets and Attitudes*, Cummins develops an account of representation that puts the problem of error at center stage. His account of error dwells on the possible mismatch between the actual content of a representation and what he refers to as its "application" to an intended target. To illustrate this, Cummins exploits a common type of classical computer architecture in which symbolic variables take specific values. Suppose the system is a chess-playing program with sub-systems that generate board states corresponding to actual elements of the game (these Cummins refers to as "intenders"). Suppose further that one such sub-system generates a slot that is supposed to be filled with a representation of the next board configuration, which happens to be P2. P2 is thus the target for representation. If all goes well, the slot will be filled with a representation of P2, i.e., RP2. This slot-filling (variable-binding) is what Cummins calls the application of the representation. Now, suppose the slot is instead filled with a representation of a different board position, namely, P3. An error would thereby occur because the intended target (P2) would not be represented by the representation that is applied (RP3). This sort of error is possible only when there is a mismatch between representation and target. Error is thus a form of *mis-application* of a representation with a fixed content to the wrong target. Because the content of the representation itself has no truth-value (it represents only the board position, not its status) the representation itself can't be false. The application, however, *does* have propositional content – in this case, it represents something like "The next board configuration will be P3." Since the next board position is P2, the content of the *application* is what is false.

Initially, Cummins's discussion appears to be only an extension (or perhaps special application) of his earlier account of S-representation. After all, the sort of computational account he invokes while describing representational error looks just like a computational account that uses S-representations as part of a simulation of a chess game. But Cummins explicitly rejects S-representation because, he claims, S-representational content cannot account for the sort of error just described. The content of S-representation depends on how the representation is used by the

system. But all use-based accounts of content, he argues, identify representational content with the intended target, thereby making it impossible for misapplication to occur. Cummins's reasoning runs as follows: Use-based accounts of representational content make the content of the representation a function of how the representation is used by the system. But use amounts to the same thing as intended application. Any theory that makes content a function of how the representation is used claims that content is determined by its target application, so what the representation actually means must correspond with what it is intended (or applied) to mean. Hence, the content of the representation will always correspond with the intended target; hence, the two can never pull apart; hence, there can be no error. But error is something any serious theory of representation must explain, so use-based theories of representation don't work. The S-representation notion is also use-based, so it too doesn't work. What is needed is an account that makes content an intrinsic feature of the representation, something that is independent of how it is employed. For Cummins, a picture-based account of representation provides this, since the structural properties that make a picture a representation of something are intrinsic features.

For those of us impressed with Cummins's original account of how the notion of representation is employed in the CCTC, this newer position is a bit confusing. On the one hand, he appeals to a familiar sort of computational process (role-filling) to attack what appears to be a natural way to think about computational representation that he once endorsed. On the other hand, he rejects an account of representation based on isomorphic relations to targets, but he endorses an account of representation that also appeals to a form of isomorphism. Unfortunately, it would take us too far afield to provide a completely detailed analysis of this apparent change of heart. Instead, I'll offer a not-so-detailed analysis, suggesting that Cummins's newer account is mistaken about one key issue.

The crux of Cummins's argument is the idea that use-based accounts of content cannot drive a wedge between a representation's target and its content. But why should we think this? All that is needed is a way to distinguish between what the system needs or intends to represent on the one hand, and what the system actually represents on the other hand. Contrary to what Cummins suggests, accounts in which the content is based upon the representation's use have little trouble doing this. One way is to tell a story whereby the system intends to token a structure that, given how it is used, represents X, but accidently tokens a structure that, given how *it* is used by the system, represents Y. It needn't be the case that the

intended causal role is the same as the actual (content bestowing) causal role for any given representational structure. That is, it needn't be the case that with a use-based account, a symbol slotted into the "Next Move" variable would automatically stand for the next move which, in the scenario described, would be P2. Instead, for sophisticated use-based accounts, a symbol would retain its content in such an application because it would retain a distinctive role in that application. When plugged into such a slot, a symbol would have a distinctive effect on the system, and this distinctive effect would contribute to its content *and* in certain situations, give rise to error. For example, the symbol RP3 would cause the system to respond differently in the "Next Move" application than the symbol RP2. The different effects of these symbols when applied to the same application contribute to their having different representational content. RP3, when put into the "Next Move" slot, has the sort of effects that are appropriate if in fact the next move is going to be P3. But since the next move isn't P3, this is a case of error. Since the next move is actually P2, the system needed to use RP2 to fill the "Next Move" variable because RP2 generates the correct simulation.

Part of what makes the tail-section of a model plane in a wind tunnel stand for the tail-section of a real plane is the role this segment plays in an explanatory and predictive model. But that doesn't prevent the possibility that the dimensions of the model's tail-section are in error, given the actual dimensions of the real plane's tail. So too, computational models can have faulty settings. The key point, then, is that you don't give up on error just because you think differences in computational roles also contribute to differences in the content of computational symbols. As noted above, it is far from clear that S-representational content is *only* determined by internal use – embeddedness can also intuitively contribute to content. But even if S-representation content should prove to be *entirely* a matter of use, Cummins's more recent analysis fails to give us good grounds for rejecting S-representation. Given the ways in which content and target can come apart even with a use-based account of content, it is still possible for S-representations to be mis-applied and thus it is still possible for S-representations to lead to the sort of error Cummins cares about.

3.4.3 What about rules?

There is yet another notion of representation traditionally associated with the CCTC that we have not yet explored in detail but that needs to be discussed. Classical systems are often characterized as employing a "rules

and representations" architecture. This characterization is misleading insofar as it suggests that computational rules are somehow different from representations. Since computational rules are generally viewed as standing for various things, like discrete commands or instructions that serve to guide the system's operations, rules clearly are meant to serve as a *type* of representation. Indeed, the rules are often said to encode a computational system's explicit "know-how." The core idea is that the system performs its various internal processes by "consulting" symbols that encode instructions pertaining to specific operations. In other words, the architecture of the system is designed so that various causal transitions are mediated by, or indeed, "guided by," these representations. While the simulation notion of representation can be seen as the computational equivalent of a road map representing the relevant terrain, the explicit rule notion is the computational equivalent of traffic signs directing traffic.

With regard to rules, the central questions we need to ask are these: Are representations of rules a distinctive type of representation? If so, what type of explanatory work do they do? If not, can they be subsumed under the heading of interior IO-representation or S-representation, or should we instead just stop treating them as representations altogether? My position is mixed. Some types of rule representations are just a special case of S-representation, and thereby have real explanatory value. There are, however, some structures characterized as rule representations that cannot be treated as a type of S-representation or interior IO-representation. In these cases, I will argue, the structures are not actually serving as representations of rules at all.

We have been demarcating notions of representation by appealing to their alleged explanatory role. In the case of rules, that explanatory role is intimately connected to the sort of command the rule is thought to encode – what it is "telling" the system to do. However, computational structures are sometimes characterized as "rules" even though their content doesn't actually tell the system to *do* anything. For example, computational rules can encode conditionals, where both the antecedent and consequence of the conditional designate aspects of the target domain. Suppose the symbolic structure encodes an entailment relation like, "If condition X obtains, then state Y will come about." Generally, such a representation will be a component of some larger model of a target set of conditions that includes conditions X and Y. When this occurs, it is clear that such a representation is just a special form of S-representation. These counterfactual representations designate actual entailment relations and so make up an important element of a model, even if the antecedent does not obtain in the actual

world. So in classical computational systems, conditional statements of this sort that are referred to as "rules" are just a special case of S-representation.

More often, though, a significantly different sort of conditional statement is assumed to be encoded by a rule. Here the antecedent still designates a real-world condition, but the consequence is thought to designate a real-world course of action. In other words, computational rules are often thought to have the content, "If condition X obtains, then *do* Y." When this happens – when the consequence is something like, "pick up the square block" or "move the Bishop to position Y" – we have what looks like a completely different sort of representation, one that is *prescriptive* as opposed to merely *descriptive*. Thus, it is less clear that this sort of rule qualifies as a special case of S-representation.

Still, I think a strong case can be made for treating prescriptive rules of this sort as a case of either interior IO-representation or S-representation. On the one hand, a computational sub-process may have as its output a representation of some conditional action rule. That is, there may be a sub-routine in a system that is designed to generate different strategies for responding to certain conditions. If so, then to regard this sub-system as having this function, we need to view its outputs as representations of the sort, "If X, then do Y." This would clearly be an instance of IO-representation. On the other hand, sometimes it may be more appropriate to regard the command as representing a stage of some real-world process being simulated, *even though* the computational system (or its real-world extension) is itself causally responsible for that particular aspect of the process being modeled. There is no obvious reason to claim that a computational system cannot, itself, participate in some of the transactions that comprise the target of its own simulations or models. There is no reason why a model user can't bring about some of the events that are part of what is being modeled. Given this, these sorts of prescriptive commands would also qualify as types of S-representation.

Yet there is a third sort of rule that doesn't appear to be a special case of either the IO-representation notion or the S-representation notion. This type of rule is thought to encode conditional commands that are couched in purely computational terms, where the rule is not about the simulated target domain but instead about some internal operation that the system itself must perform. Instead of "pick up block" or "move Bishop to such-and-such position," the command is thought to mean something like "perform computational sub-routine Z" or "re-write symbol W in position S." In other words, the command refers to various aspects of the model or simulation process itself rather than to aspects of processes that are being modeled

or simulated. Consequently, the content of such a command never goes outside of the realm of the computational system. It is this third notion of rule representation that I want suggest is *not* doing any valuable explanatory work.

While these issues are notoriously tricky, it should first be pointed out that it is actually far from clear that these sorts of computational rules actually *are* part of the explanatory framework offered by the CCTC. Remember that the CCTC is a theory of how cognitive systems work. It is not a theory of how to *implement* the states and processes described by that theory in an actual machine. It could be argued that the structures in computational systems that serve as rules in this sense are really just part of the implementing architecture, and not an essential part of the CCTC's explanatory apparatus.[15] They are perhaps essential for programming actual physical machines, but they aren't essential for understanding the sense in which cognitive processes are said to be computational.

Yet some might say that in certain theories, these types of rules are indeed intended as part of the theory's explanatory apparatus. Let's assume for the sake of argument that this is so. We can see that this "internal" notion of rules cannot play the same sort of explanatory role played by either the IO notion or the S notion. Suppose there is a sub-component of the system with the mechanical job of erasing and writing symbols. Moreover, suppose this sub-component is triggered to erase the symbol "X" and re-write the symbol "Y" by receiving as input yet a third formal token. Do we need to treat this third data structure as representing the command "erase symbol 'X' and re-write symbol 'Y'" in order to treat this sub-component as a symbol eraser/writer? Surely the answer is "no." Because the sub-component is doing purely mechanical operations, we can treat the sub-component's inputs as merely formal tokens and its outputs as actual symbol erasings and writings without treating either the input or output as representations. To view a sub-system as an adder, we need to view its inputs and outputs as representations of numbers. But to view a yet more basic sub-system as an eraser and re-writer of formal symbols, we don't need to treat *its* inputs or outputs as representations of anything. We need to treat its outputs as the erasing of formal symbols, but we don't need to pay any attention to what these symbols represent.

The S-representation notion wouldn't apply to these "rules" either since they don't serve as elements of a model or simulation that the system is using. If anything, the rules are thought to correspond to the mechanical details of the simulation itself – about the simulat*or*, not the simulat*ed*.

[15] See Fodor and Pylyshyn (1988).

Rules of this sort refer (allegedly) to the "behind-the-scenes" mechanical steps or processes that are necessary for the simulation's execution. They aren't themselves *part of* the simulation or model. Hence, this notion of representation can't serve as a form of S-representation either.

If internal rules of this sort (that is, rules that designate specific mechanical operations) can't serve as either IO-representations or as S-representations, then in what sense are they supposed to serve as representations? One proposed answer is that we should treat these internal states as representations of rules simply because they generate various state-transitions in the computational process. Because these structures are causally pertinent, the system is thought to "follow" commands or instructions that they encode. For example, Newell and Simon (1976) offer the following account of what it is for a computational system to "interpret" a symbolic expression: "The system can interpret an expression if the expression designates a process and if, given the expression, the system can carry out the process . . . which is to say, it can evoke and execute its own processes from expressions that designate them" (1976, p. 116). So on this view, executing a process amounts to interpreting a rule or command "expressing" the procedure that needs to be implemented. We view structures as representations of commands because these structures cause the system to carry out the expressed procedure.

The problem with this perspective is that it suggests a notion of representation that is too weak to have any real explanatory value. There is no beneficial level of analysis or explanatory perspective that motivates us to regard things as representations simply because they influence the processing. There is nothing gained by treating them as anything other than causally significant (but non-representational) components of the computational system. Of course, we can always cook up a command corresponding to the relevant causal role, and then allege that the structure represents that command. For example, we can say that a spark plug's firing expresses the rule, "piston, go down now," or that a door-stop represents to the door the command, "stop moving here." But there is no explanatory pay-off in treating these things in this way. Or, putting things another way, spark plugs and door stops don't actually *serve as* representations. Similarly, calling computational elements representations of rules simply because they initiate certain computational operations adds nothing to our understanding of how computational processes are carried out. There is no sense in which states that cause different stages of computational processes actually play a representational role, and nothing is added to our understanding of computational systems by treating *these* sorts of structures as things that encode rules.

The fact that the so-called rules can be modified, so that they have different influences on the processing at different times, is itself sometimes suggested as a justification for treating them as encoding instructions. That is, because the causal influence of a computational element can be altered, it is suggested that this alterability gives rise to their status as representations. But it is hard to see why this should matter. There are plenty of causal systems in which the inner elements can be adjusted but clearly aren't serving a representational role. For example, a commonplace timer mechanism turns lights and other appliances on and off by closing an electrical circuit at specific times. The activation times can be changed by moving pegs to different locations on a 24-hour dial, so the pegs control the timing of the flow of electricity by their position on the dial. Given this modifiable causal role, someone might propose that the pegs in specific slots encode "rules" like, "If it is 6:45 p.m., then turn on the lamp" or "If it is 11:30 p.m., then turn off the lamp." We can, in other words, adopt the intentional stance with regard to the timer pegs, and claim that the timer "interprets" and obeys these commands. However, there is no reason to adopt this perspective. We can understand everything we need to know about how the pegs operate in the timer without introducing representational language. The same goes for causally relevant components of computational systems that are necessary for the implementation of computational processes. Unlike the situation with IO-representations or S-representation, the intentional stance buys us nothing with these structures, and the fact that their influence can be modified doesn't change this. We can understand everything about the way they function in the system – about their purpose and computational significance – without regarding them as rules that are in some sense interpreted by the system.[16]

In chapter 5, we will return to the idea that things serve as representations because of what they cause. For now, the point of this digression is that when

[16] There is yet another feature of computer elements, besides their causal relevance, that invites researchers to regard them as representations of commands. The feature concerns the way these elements are typically created and modified in actual programs, which is by a programmer typing out instructions in a programming language that *we* would translate as something like "when X, do subroutine Y." It is not so surprising, then, that something is thought to be a representation of such a rule *for the system*, especially since the system appears to be following just such a command. Yet, this is just an artefact of the way computer programs are written, one that doesn't change our earlier verdict that we lack a good reason for positing encoded rules that the system interprets. Suppose we altered the way in which the timer pegs get placed, so that it now happens via typed commands. To get the "on" peg to move to the position that will turn on the lamp at 6:45, I type on some keyboard, "If 6:45 p.m., then turn on lamp." It seems intuitively clear that this modification in the way the pegs get positioned does nothing to provide a justification for viewing them as representations of rules. It might explain why we would be tempted to call them rules, but it doesn't alter the basic fact that their functionality is no different from other basic causal elements.

proponents of the CCTC posit rules that are employed by computational systems, they are often referring to structures that are indeed representations of rules, because they are special cases of either interior IO-representation or S-representation. But sometimes commentators refer to something that is not serving as a representation of a rule at all. In the case of the former, the nature of the CCTC explanations demands we treat these structures as representations of rules; in the case of the latter, it does not.

3.4.4 The vindication of folk psychology revisited

In the last chapter, we saw how the Standard Interpretation links the positing of representations in the CCTC to folk psychology. On the Standard Interpretation computational structures receive their representational gloss by serving as realizers of beliefs and desires. Rather than first demonstrating how the CCTC itself invokes inner representations and then exploring if and how the folk notions map onto this account, this perspective suggests that the CCTC applies the folk notion of representation to computational structures as a way of showing that the notion of representation is needed. Representational states are thereby seen as theoretical add-ons that are not directly motivated by the CCTC explanatory framework. The end result is a picture in which the explanatory value of representation becomes questionable, and the representational nature of the CCTC is called into doubt.[17]

Yet we can now see that on the proper interpretation, representational notions are actually built right into the explanatory pattern offered by the CCTC. IO-representations and S-representations are an indispensable feature of the theoretical framework, and the explanatory value of these notions is independent of anything associated with folk psychology. With this corrected picture of CCTC representation in hand, we can now return to the question of whether or not, if true, the CCTC would provide a vindication of folk psychology.

If we are going to show that a folk concept of some sort is vindicated by a scientific theory, then the first obvious step is to establish that the scientific

[17] Like the S-representation notion, the Standard Interpretation appeals to a sort of isomorphism to establish the need for representations. But it is the wrong sort of isomorphism. The isomorphism it exploits is between the causal structure of symbol manipulations and the sort of psychological processes stipulated by folk psychology. This merely tells us that computational symbols can behave like the posits of folk psychology; it doesn't provide us a reason for thinking those symbols should be treated as representations. S-representation, on the other hand, says the isomorphism that matters is between the symbol manipulations on the one hand, and whatever it is that is being modeled or simulated on the other hand. This sort of isomorphism establishes how computational structures serve as representations because it requires computational structures to serve as components of models and simulations.

theory is actually committed to something with the central features associated with the folk notion in question. Unfortunately, there are no clear criteria for what in general counts as "central features." Nor is there a clear consensus on how many central features need to be possessed by the scientific posit to distinguish cases of retention from cases of elimination (Ramsey, Stich, and Garon 1990; Stich 1996). Consequently, the analysis must be done on a case-by-case basis and unavoidably involves a judgment call. In some cases, reduction only requires that the scientific posit play the same causal roles as the folk posit. But this is often not enough – epileptic seizures don't vindicate demonology even though epileptic seizures cause many of the behaviors associated with demonic possession.

Folk psychology is committed to the existence of mental representations. Therefore, for folk psychology to be vindicated, the correct scientific theory needs to invoke, at the very least, inner cognitive representations as well. What we can now see (but couldn't from the perspective of the Standard Interpretation) is that the CCTC meets this minimal requirement. The CCTC is indeed a representational theory of the mind – one that is committed to the idea that the brain employs structures that are inner representations. If the CCTC is correct, then at least *this* aspect of folk psychology will be vindicated.

Of course, this is only part of the story. The scientific account must also posit representations with the right sort of properties. Since the central properties of propositional attitudes are their intentional and causal properties, the scientific theory must posit representational states with similar intentional and causal properties. If the posits are too dissimilar from our ordinary notions of mental representation, then, despite serving as representations, the psychological theory may be too unlike our commonsense psychology to provide a home for the posits of the latter. For example, Stephen Stich, Joseph Garon, and myself have argued that connectionist distributed representations don't qualify as the right sort of representations because they lack the requisite functional discreteness to act in the manner commonsense psychology assumes of beliefs and propositional memories (Ramsey, Stich, and Garon 1990). Distributed connectionist representations can't vindicate folk mental representations because the former lack the sort of causal properties the latter needs.[18]

[18] I now believe that our eliminativist analysis of connectionist networks didn't go far enough, since my current view is that it was a mistake to allow that distributed networks employ *any* legitimate notions of inner representation. My reasons for this view will be spelled out in the next two chapters. See also Ramsey (1997).

Our current concern is not with connectionism, however, but with the CCTC. Are the notions of IO-representation and S-representation the sort of posits with which beliefs and other folk notions could be identified? While the two computational notions are not the same as the folk notions, they clearly share many of the same features. They both have the sort of intentionality that we associate with our thoughts and they are also capable of the kind of functional discreteness that folk psychology assigns to beliefs and desires. Moreover, in many respects, the sense in which they serve as representations overlaps with the sense in which we arguably think thoughts serve as representations. To see this last point better, consider a piece of folk psychological reasoning that Fodor treats as instructive, offered by Sherlock Holmes in the "The Speckled Band":

> "... it became clear to me that whatever danger threatened an occupant of the room couldn't come either from the window or the door. My attention was speedily drawn, as I have already remarked to you, to this ventilator, and to the bell-rope which hung down to the bed. The discovery that this was a dummy, and that the bed was clamped to the floor, instantly gave rise to the suspicion that the rope was there as a bridge for something passing through the hole, and coming to the bed. The idea of a snake instantly occurred to me ..." (In Fodor 1987, pp. 13–14)

Here Holmes is offering, as Fodor notes, a bit of reconstructive psychology. He is applying commonsense psychology to himself to explain how his realizations, thoughts, observations and ideas led to his conclusion that the victim died of a snakebite. What Fodor asks us to note is how much Holmes's account resembles an argument, with clear premises, conclusions and chains of rational inference. Because classical computational systems are good at this type of formal and explicit reasoning, they provide, according to Fodor, the avenue for vindicating commonsense psychology.

But now consider the same passage from the standpoint of S-representation. Instead of describing a reasoning process that looks like a formal argument, Holmes's account of his own reasoning can be seen as involving something like a model of the events that led to the victim's demise. In this passage, Holmes at least implies that he discovered the solution by mentally reconstructing the critical series of events that were involved in the murder – a reconstruction that included representations of the relevant elements (vent, dummy rope, snake) and the pertinent events (the snake slithering down the rope) to complete the picture and solve the crime. Holmes's version of folk psychology makes it sound a lot like running a simulation of events and processes, or building a model and then, as we say, "connecting the dots."

My point is not to challenge Fodor's Conan Doyle scholarship. Rather, the point is that folk psychological explanations of mental processes can often be seen to characterize those processes as involving models or simulations, or what we earlier referred to as "surrogative reasoning." If this is correct, then folk notions of mental representations may well be very close to the notion of S-representation proposed by the CCTC. The S-representation notion, although not identical to our ordinary notion of propositional attitudes, may well be in the ballpark of the kind of representational state that could vindicate a modified version of folk psychology.[19] While it is hard to see how beliefs could turn out to be mere syntactic states with an unspecified representational role (as suggested by the Standard Interpretation), it *does* seem they could turn out to be representational components of models that our brains use to find our way in the world. Consequently, if the CCTC should prove correct, then that may provide us with good reason to think that belief-like states will find a home in a serious scientific psychology after all. The CCTC may indeed vindicate commonsense psychology, but not without first being understood as a theory that invokes inner representations for its *own* explanatory reasons.

3.5 SUMMARY

It is important to be clear on the objective of this chapter. The aim has not been to defend the CCTC as a true theory of our cognitive processes. Rather, it has been to defend the idea that the CCTC is indeed a *representational* theory of our cognitive processes. My goal has been to show how the CCTC framework makes use of notions of representation that, contrary to the Standard Interpretation, are needed for reasons that are independent of any desire to vindicate folk psychology. As we've seen, one notion is connected to the hierarchically organized, sub-modular nature of cognitive processes generally assumed by the CCTC. The other notion is connected to the sorts of models and simulations many versions of the CTCC paradigm invoke. Both notions of representation appear to meet the job description challenge and reveal how CCTC theories of the mind are

[19] A question some have posed is this: How do folk notions of mental representation meet the job description challenge? Initially, it seems that they clearly don't. Folk psychology doesn't tell us *how* mental states like beliefs come to serve as representations: it simply presupposes their representational status without trying to explain it. This is one of the key differences between folk psychology and many sorts of scientific psychology. Yet on second thoughts, it might turn out that, deep down, our concept of belief includes the role of serving as part of a person's inner model of the world. If this is so, then beliefs would simply be a type of S-representation.

representational theories of the mind. While there are difficulties associated with each of these posits, these are perhaps no worse than the sort of philosophical problems associated with many posits of scientific theories.

This analysis of representational notions that succeed in meeting the job description challenge will serve as a contrast to what comes next. In the next two chapters, we'll look at two different notions of cognitive representation that have become popular among those working in the cognitive neuroscience and connectionist modeling. Unlike my treatment of the notions of representation discussed here, I'll argue that these notions fail to meet the job description challenge and do no real explanatory work. My claim won't be that the non-CCTC theories are false. Rather, my claim will be that, contrary to the way they are advertized, many of these accounts fail to invoke internal states and structures that are playing a representational role. When we look closely at these other representational notions, we can see that the states they describe are not really serving as representations at all.