COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 37 (2013) 1171–1191
Copyright © 2013 Cognitive Science Society, Inc. All rights reserved.
ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/cogs.12062

# Programs as Causal Models: Speculations on Mental Programs and Mental Representation

Nick Chater,[a] Mike Oaksford[b]

[a]*Behavioural Sciences Group, Warwick Business School, University of Warwick*
[b]*Department of Psychological Sciences, Birkbeck College, University of London*

Received 6 June 2012; received in revised form 22 October 2012; accepted 6 December 2012

**Abstract**

Judea Pearl has argued that counterfactuals and causality are central to intelligence, whether natural or artificial, and has helped create a rich mathematical and computational framework for formally analyzing causality. Here, we draw out connections between these notions and various current issues in cognitive science, including the nature of mental "programs" and mental representation. We argue that programs (consisting of algorithms and data structures) have a causal (counterfactual-supporting) structure; these counterfactuals can reveal the nature of mental representations. Programs can also provide a *causal model* of the external world. Such models are, we suggest, ubiquitous in perception, cognition, and language processing.

Judea Pearl has helped lead two revolutions in the understanding of intelligent systems, both natural and artificial. The first revolution (e.g., Pearl, 1988) involved the creation of a new type of probabilistic model, graphical models, which provide a compact and transparent formalism for representing probabilistic knowledge; over which elegant methods for learning and inference can be defined; and which can, in principle, be implemented in the highly parallel, distributed, computational architecture that appears characteristic of the brain. The impact of this and related work (Lauritzen & Spiegelhalter, 1988; Pearl, 1985) on artificial intelligence and machine learning has been enormous (e.g., Jordan, 1999). And graphical modeling tools have been central to the recent surge of research in Bayesian cognitive science (e.g., Chater & Oaksford, 2008; Glymour, 2001; Gopnik & Tenenbaum, 2007; Gopnik et al., 2004; Griffiths, Kemp, & Tenenbaum, 2008; Oaksford & Chater, 2007; Sloman, 2005; Tenenbaum, 1999; Tenenbaum, Kemp, Griffiths, &

Correspondence should be sent to Nick Chater, Behavioural Sciences Group, Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK. E-mail: nick.chater@wbs.ac.uk

Goodman, 2011), an approach that has recently been the focus of intense debate (e.g., Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010 vs. McClelland et al., 2010; Jones & Love, 2011 vs. Chater et al., 2011). The second revolution (Pearl, 2000) that Pearl has helped to lead concerns the modeling of causality. This, indeed, involves a shift away from a focus on purely *probabilistic* models to *causal* models (see also Spirtes, Glymour, & Scheines, 1993). This second revolution has had a huge impact in statistics, epidemiology, and philosophy; and it has led to a flurry of important work on the cognitive science of causal and counterfactual reasoning in both adults and children (e.g., Gopnik, Sobel, Schulz, & Glymour, 2001). In this article, we suggest that the impact of this second revolution is only just beginning: that causality and counterfactuals may turn out to be of central importance across almost every area of cognition.

Let us begin a simple example of Pearl (2000, p. 346; see Woodward, 1999). Pearl considers the difference between two representations of simple algebraic relations, which he terms "equations versus diagrams," contrasting:

$$\begin{aligned} Y &= 2X \\ Z &= Y + 1 \end{aligned} \tag{1}$$

with

$$X \rightarrow \times 2[Y] \rightarrow +1[Z] \tag{2}$$

The former describes relations between variables; the latter specifies a simple computer program, in the form of a flowchart, indicating the order in which operations are carried out. What is the difference between the equations and the flowchart? The critical distinction that Pearl points out concerns *interventions*. We can consider the flowchart to specify a *process*: a sequence of operations over registers which store numerical values. We start with a register which stores a value of $X$, say 3. The first arrow in the flowchart indicates that this value should be multiplied by 2, to give 6; and this is stored in a register holding the value of $Y$. The second arrow in the flowchart specifies that we should next retrieve this value and add 1, so yielding 7, which is then stored as the value of a third variable, $Z$. Crucially, we can *intervene* midway through this process and, say, set $Y$ to have, instead, the value of 9 (rather than 6). The initial assignment $X = 3$ is unchanged; but the new value of $Y$ is now fed into the next step (to add 1), yielding $Z = 10$. Thus, and crucially for our discussion below, the flowchart thus defines counterfactuals such as *if Y had been 9, then Z would have been 10*. By virtue of being counterfactual supporting, we shall call a program a *causal* system.[1]

Flow diagrams are a rather restricted type of program. But the same distinction arises more generally. Consider the difference between equality, as above, and the operation of assignment, often written ":=" or similar in computer programming languages, in clauses such as:

$$\begin{aligned} X &:= 1; Y := 1 \\ Y &:= 2X \\ X &:= 2Y \end{aligned} \tag{3}$$

These clauses have similar import to our flow diagram. Two variables $X$ and $Y$ initially have value 1. Then $Y$ is assigned to be twice the current value of $X$ (i.e., $Y$ has the value 2); and $X$ is finally assigned twice this new value of $Y$ ($X$ now has the value 4). Viewed as *equations*, these clauses make no sense (e.g., $X = 1$, $Y = 1$, $Y = 2X$; $X = 2Y$ yields a mathematical contradiction). Instead, they are (counterfactual supporting) *programs*: Each statement indicates how one variable is dependent on the value of the other (but not vice versa).

The close relation between descriptions and programs can be illustrated using a familiar example from computational linguistics. A fragment of a language might be *described* with a simple phrase structure grammar. Such a description specifies relationships between linguistic representations, with no commitment to causality or counterfactuals:

$$\begin{aligned} \text{S} &\rightarrow \text{NP VP} \\ \text{VP} &\rightarrow \text{V NP} \\ \text{NP} &\rightarrow \text{Det N} \\ \text{Det} &\rightarrow \text{the, a} \\ \text{N} &\rightarrow \text{bird, fish} \\ \text{V} &\rightarrow \text{likes} \end{aligned} \tag{4}$$

But this grammar can also be viewed as a *program* (Pereira & Warren, 1983)—for example, as a definite clause grammar, as used in the logic programming language PROLOG[2] :

$$\begin{aligned} \text{S} &\rightarrow \text{NP VP} \\ \text{VP} &\rightarrow \text{V NP} \\ \text{NP} &\rightarrow \text{Det N} \\ \text{Det} &\rightarrow [\text{the}] \\ \text{Det} &\rightarrow [\text{a}] \\ \text{N} &\rightarrow [\text{bird}] \\ \text{N} &\rightarrow [\text{fish}] \\ \text{V} &\rightarrow [\text{likes}] \end{aligned} \tag{5}$$

The program specifies operations over particular data structures: lists.[3] Generating a sentence involves assigning the value of the left hand item to that of the right hand item or items, until this process terminates: that is, creating a causal sequence, starting with S, going through a sequence of assignments (by applying the rules above), and finishing with a list of words corresponding to a sentence. Using the symbol "↓" to indicate the operation of a computational step, a causal sequence operating over lists of items might informally be written as:

$$\begin{aligned}&\text{S} \downarrow \text{NP, VP} \downarrow \text{Det, N, VP} \downarrow \text{Det N V NP} \downarrow \text{Det N V Det N} \downarrow\\&\text{the N V Det N} \downarrow \text{the bird V Det N} \downarrow \text{the bird likes Det N} \downarrow\\&\text{the bird likes a N} \downarrow \text{the bird likes a fish}\end{aligned} \tag{6}$$

where aspects of the ordering are arbitrary (and could be carried out in parallel). The sequence of operations is, as above, counterfactual supporting. For example, *if* S was evaluated to NP NP rather than NP VP, *then* the rest of the sequence of possible sentences is still well defined, so that, for example, *the bird the fish,* would be a possible sentence. Or, given a sentence such as *the bird likes the fish*, we might remark that *if* the first NP had been evaluated as *the bird with the long beak* or, more bizarrely, *the the the*, then the whole sentence would have been as *the bird with the long beak likes the fish* or *the the the likes the fish.*[4]

The key point is that a successful implementation of the DCG requires not just that it correctly deals with the generation of *actual* sentences, but that it correctly captures the causal impact of possible *interventions* on the data structures during the operation of the program. That is, the DCG, like any other program, specifies what *would have happened* if interventions had been made on relevant data structures. For the computation to be meaningful, it is crucial that the counterfactuals generated by the causal structure of the algorithm fit with the counterfactuals appropriate to the interpretation of the data structures.

The mainstream generative grammar perspective moved away from the idea that grammar can be mapped into a "mental program" which generates and understands sentences. But the idea remains popular with proponents of various grammatical formalisms inspired by computational linguistics, such a Combinatory Categorial Grammar (Steedman, 2001). In such formalisms, the grammar maps directly into rules defining a program for parsing and production—and, indeed, the grammar may be viewed as *defined* by these (counterfactual supporting) rules. We shall return to the relation between grammar and processor below. First, though, with these motivating examples in mind, we move to a slightly more abstract discussion, before returning to specific issues in cognitive science.

## 1. What is a program?

We have drawn a distinction between descriptions, which say how the world is (or which sentences a language contains) and programs, which provide a causal mechanism for *generating* material (e.g., numbers, sentences, or data of any other kind). We have so far specified programs in symbolic terms—but these provide *specifications* or *descriptions* of a program, rather than *being* the program (i.e., an implementation of a flowchart is not itself a flowchart).

So what, in more abstract terms, is a program? Following standard usage in computer science, a program consists of two things: the *data structures* in which information is stored (e.g., registers for storing integers, lists, trees, etc.) and an *algorithm* which operates over these data structures. This yields the familiar formula: Algorithms + Data Structures = Programs (Wirth, 1975).[5]

The program specified by the flowchart ((2), above) is, then, defined by an algorithm operating over data structures representing numerical values; the program captured by a set of DCG rules ((5), above) is defined by an algorithm over lists representing linguistic categories and words.

As we have stressed, an algorithm determines not just what happens given a specific input but what *would* happen, if the content of the data structure were different. Thus, interventions can be specified over data structures *during the computation*; and the operation of the algorithm will lead to well-defined *counterfactuals* as a consequence of these interventions (it will, of course, sometimes crash or loop). Let us briefly consider some familiar examples.

1. *The Turing Machine*. This consists of two key elements: a tape (the data structure), on which binary symbols (by convention, “0” and “1”) may be written and read; and a finite state automaton (embodying the algorithm) which hovers over a single square on the tape. The automaton’s next move is determined by its current state (of a finite number of states) and what it reads on the tape at its present square. Its “actions” consist of (optionally) modifying what is written on the current square of the tape; and moving to the right or to the left. The machine starts with a specific binary input string; when it halts, the state of the binary string encodes the output of the computation. Any specific Turing Machine can be viewed as implementing a *function* (or rather a partial function, because some inputs may, for example, lead the machine into an infinite loop and return no value) from input to output states of the tape. But when we make the split between algorithm and data structure, the Turing Machine can also be viewed as encoding a rich set of counterfactuals about what *would* happen in the light of different possible interventions over its data structure, the tape. Specifically, suppose that we were, at any point in the computation, to modify the contents of one square of the tape. The Turing Machine will continue to run and produce an output (or loop indefinitely). Indeed, at *any* point in the computation, we can modify any number of states of the tape; the behavior of the Turing Machine remains well defined. So the Turing Machine does not merely implement a function; it implements a mapping from inputs and interventions to outputs, where interventions here consist of one or many modifications of the contents of the tape while the computation is in progress.
2. *Computer programs*. Conventional computation is defined not by specifying automata to operate over Turing machine tapes, but rather by programs which operate over a variety of data structures (e.g., lists, stacks, arrays, and so on). The program specifies an algorithm (like the automaton in a Turing Machine); this algorithm is viewed as fixed, again like the automaton, and operates on the data structures in a

well-defined way. In the absence of interventions, the program will specify a partial function from initial to final configurations of the data structures. But at any point in this computation, the contents of the data structures can be modified (e.g., the top item of a stack might be deleted; one or more elements in an array might be switched). The output the processor produces is well defined and can, as before, be viewed as specifying counterfactuals—that is, specifying what *would* have happened if certain interventions had occurred.

3. *Interpreters*. An especially important class of programs takes a description of a program as data (in a programming language) and maps this into a set of executable instructions on some machine—that is, the interpreter actually builds the data structures and runs the algorithm.[6] Following the general principle that counterfactuals are defined over modifications of data structures, the interpreter therefore specifies counterfactuals over *descriptions* of programs. Thus, we can consider what would have happened had the program been different. So, for example, we can ask what would happen if a *description* of the program for, say, the factorial function, were modified in various ways. Or, given an interpreter for flowcharts, we can meaningfully ask what program would result if our flowchart were modified; given an interpreter for DCGs, we can meaningfully ask what the language would contain, if one or more grammatical rules were added, deleted, or modified.
4. *Neural networks*. We have so far focused on symbolic computation. But the framework is not tied to any specific model of computation. For example, the connectivity, weights, and biases of a standard feedforward neural network specify a function from inputs to outputs; but also a set of counterfactuals concerning what the output would be, if the values of hidden units had been different. The network structure defines the algorithm; the units, each associated with numerical values (typically, in the interval [0, 1]) correspond to the data structure over which the algorithm operates. By contrast, a *learning algorithm* for such a network, such as back propagation (Rumelhart, Hinton, & Williams, 1985), typically operates over data structures, including the weights and biases of the network. It will specify counterfactuals concerning future learning and network behavior that would obtain if, say, one or more weights had different values.

So, in summary, cognitive science typically views thought processes as *programs* and programs consist of algorithms and data structures. This division allows us to define not only the input–output function to which it computation corresponds; but how the computation *would* have run, had one or more pieces of data be modified *while* the computation was in progress. Thus, for any program, we can separate algorithm and data structure; and then we are able to define a class of *counterfactuals*, by manipulating the contents of the data structure and holding the algorithm constant.[7]

How do these ideas relate to the question of representation—which, as we see below, is a vexed one in cognitive science? The answer appears simple and direct. Note that data structures can have representational properties (numbers might represent bank balances; trees might represent structure of sentences; and so on); but algorithms cannot: They are simply processes operating over the data structures.[8] So there is a correspondence between the information that the program represents, and the counterfactual is that program supports that which is represented is treated as data; and that which is treated as data can be manipulated during the computation, with well-defined causal consequences for the progress of computation. We express this in the slogans: *no representation without causation*; and *no causation without intervention*.[9]

This simple formulation is, we argue, deceptively powerful as a route to gain insight into mental representation.

## 2. Breaking open the mental program

If cognition is computation, then the task of understanding the mind requires specifying the algorithms and data structures (i.e., the *programs*) underlying thought. Theorists differ fundamentally on what might be plausible assumptions concerning such algorithms and data structures. Such questions may be general, for example, do the mind's data structures include symbols, pictures, or patterns of activity across the distributed neural network? But they are often quite specific: Is English past test morphology mentally represented in terms of rules, exemplars, or connection strengths in a neural network? Does the language processing system compute syntactic trees? In what sense, if any, is grammar mentally represented? Do people and non-human animals learn by forming associations or by representing the causal structure of the external world? Are mental images represented pictorially or propositionally?

Such questions are difficult to resolve in practice; and some theorists have raised the possibility that they are unsolvable in principle. One general, and worrying, observation is that, if we fix a general purpose programming language (say, Java or C++), the following result holds. If an input–output function can be computed by algorithm/data structure pair at all (i.e., if the function is *computable*), then there is an infinite number of algorithm/data structure pairs that can be specified in this language that will compute precisely the same function. A further observation is each general purpose computer language (given very minimal conditions) can capture precisely the same input–output functions. So "reverse engineering" the representation/process structure of a computational device from its inputs and outputs seems to be a hopeless endeavor. Such concerns can be deepened further, in the light of a wide range of "mimicry theorems," which aim to show that, in specific domains, models with fundamentally different representational/process assumptions can mimic each other (e.g., between pictorial and propositional representations in visual imagery, Anderson, 1978; or rule and similarity-based models of categorization, Hahn & Chater, 1998).

In the light of the discussion above, it seems clear what is needed to break open the cognitive "program" into its algorithm and data structures. In line with our slogans, what is required is the ability to *intervene* on the internal data structures and observe the causal consequences. This would allow the direct exploration of the counterfactuals associated with the particular data structure/algorithm pair.

How might this be done? There are various possibilities. But the simplest is to attempt to manipulate our internal data structures by directed *top-down control*: In essence, this means considering what we can actively *imagine*. The rationale is that whatever can be actively manipulated must be modifiable, that is, represented in the data structure. Moreover, we might reasonable conjecture that "easy" manipulations (in our imagination) will correspond to simple modifications of the data. The claim is, then, that *if* top-down manipulation is possible, *then* the relevant variables must be represented in the relevant data structures. The reverse inference does not, of course, follow—there may be data structures which top-down control cannot modify (but which might, in principle, be modified by other mental processes [perhaps non-conscious], by direct electrical stimulation with a microelectrode, or by some other means). But the argument is, nonetheless, widely applicable.

Consider viewing an everyday scene, such as an urban street. Is the street scene represented in 2D or 3D? Counterfactuals give an immediate answer: We can easily imagine various transformations of the 3D scene (e.g., a car drives away; a person walks nearer; a shop "magically" disappears; a small sapling is, in our imagination, transformed into a giant tree). But it is more or less impossible to imagine any transformations of the 2D *image*: We cannot, for example, envision two equal sized patches of the image and switch their locations. We cannot delineate a circular patch in the image (containing a clutter of object are different distances) and rotate it, while leaving the rest of the image fixed. We *can*, by contrast, readily mentally rotate *objects* and do so in 3D. For example, it is not hard to imagine a parked car flipped upside down and spinning gently upon its roof. Similarly, we can imagine parts of objects being modified: street signs appearing and disappearing; windows and doors being opened. Objects and their parts can be imagined to have their properties modified: A shop awning might be imagined to change color or material or to start billowing in the wind; a car might, in our imagination, be elongated into a stretch limousine; and so on. Note, crucially, that we can trace the consequences of these counterfactuals: The great tree would block out the light and break up the road surface; the imaginary stretch limo would be too long to turn down a narrow side street; the imagined new color of a shop awning would clash horribly with that of the shop next door; and so on.

If we trust introspection, at least, then noting these counterfactuals appears immediately to imply: (a) that we perceive the world in three dimensions (it does not, of course, rule out the possibility that the brain *also* uses 2D representations, e.g., in early stages of vision); (b) that we represent the environment in terms of objects; (c) and parts of objects; (d) and properties of objects and their parts (e.g., shapes, colors, etc.); (e) although the computational counterfactual that we can envisage can violate causal structure, their consequences typically do not (although we can imagine violations of normal causal laws to a limited degree; for example, we can imagine the appearance of a ghostly car that can drive through solid objects).

These simple observations appear to provide a powerful argument for a representation of the visual world in hierarchical terms, where objects, parts, and properties are represented; and that the algorithms operating over those representations embody rich causal information (because the causal consequences of our imagined interventions can readily be envisaged). Importantly, though, it is not clear how any of these properties can be supported by a representation in the form of a picture or a map. The interventions that can be made over a real picture or map, after all, are rooted in 2D: We can cut up, remove, or perhaps photocopy pieces of picture or map, but little else. Contrast these 2D representations with the symbolic hierarchical data structures underlying an electronic representation of the world underpinning computer graphics software, according to which objects and their properties can be readily manipulated (e.g., it is possible to change the location of "virtual" buildings, change their dimensions and locations, and so on). If pictures and maps provide useful models for internal representations of the perceptual environment, it is the symbolic, electronic images, maps, and computer graphics or animations generated by highly sophisticated databases that provide the appropriate analog, not their instantiations as ink on article (see Pylyshyn, 1984; for in-depth analysis along these lines). There are, of course, many independent lines of evidence and argument for this perspective on vision (e.g., Marr, 1982; Rock, 1985). But we suggest that considering the nature of what we can imagine provides a powerful and direct route.

Notice that this viewpoint also suggests some deep links between perception and imagery because, as in the examples above, imagery can operate on perceptual representations. That is, we can visualize, to some degree at least, the result of some transformation on the real perceptual world; or, in the laboratory, we can be tested in our ability to engage in specific transformations of perceptually presented objects, such as the mental rotation of shapes or objects (e.g., Shepard & Metzler, 1971). But if this is so, perception and imagery must use the same representations—and must be deeply intertwined (a view for which there is, of course, substantial independent evidence, e.g., Kosslyn, Thompson, & Ganis, 2006; Shepard, 1984). Moreover, we suggest that both involve hierarchical, symbolic representations of the structure of the world.

The style of argument outlined here may appear to have, in Russell's (1919) phrase, all the virtues of theft over honest toil. Can casual reflections on what we can imagine really yield insights into the nature of mental representation that resist careful experimental analysis? Matters are not so straightforward. Crucially, we cannot and need not simply rely on introspection concerning how we can manipulate language or visual material; these questions can be tested experimentally (and indeed some early work in psycholinguistics took this line, e.g., Fodor & Garrett, 1967; as does the large literature of the manipulation of mental images, e.g.,. Kosslyn, 1980; Shepard & Metzler, 1971). Nonetheless, we suggest that to the extent that we can empirically understand how people can manipulate language and imagery, we shall thereby gain immediate insight into the nature of the underlying mental representations.

## 3. Programs as causal generative models

We have argued that programs generate counterfactuals—and to this extent have a causal structure. But our reflections on perception and imagery suggest that perception and imagery involve constructing programs which, crucially, capture the *causal structure of the world*. Put simply, algorithms capture putative "laws of nature"; data structures describe the state of the world (and modifications of those data structures, and tracing their consequences, generate counterfactuals).

Let us elaborate this viewpoint, by drawing a parallel with technologies that use hierarchical symbolic methods to generate images. Consider how a computer graphics program generates a simple scene: a cube resting on an infinite flat surface, illuminated from above by a point source. A reasonable flexible program will allow the user to move the cube, allowing it to have various orientations and locations in the plane. A suitably rich model that embodies principles of gravitation and Newtonian mechanics will automatically set the cube immediately above, and aligned with, the surface on which rests. A program generating animated sequences, rather than graphics stills, will be able to go further: to simulate the movement of a cube, starting with an arbitrary initial position, and modeling how it falls, bounces, and comes to rest flush with the surface below. A technique such as "ray-tracing" will, moreover, embody the principles of optics. Thus, given a light source, and information about the reflectance functions of the surfaces of the objects in the scene, the algorithm will recreate shadows and reflections in line with physical principles. Notice, crucially, that for the processes that reconstruct the physical properties of the environment to create a convincing "virtual world" they need to recreate (to some level of precision) the causal sequence observed in the "real world." Thus, these computational processes are subject to the same types of counterfactuals as the real world. So, for example, if location of the cube is changed, or the strength or direction of the lighting is modified, then the shadows and reflections will be adjusted automatically. If, in an animation, the user specifies a different starting point or orientation for the cube, then its subsequent trajectory and resting place will be adjusted accordingly, and so on.

So the creation of realistic computer graphic images requires the recapitulation, to some degree of fidelity, of causal processes that would give rise to the corresponding naturally occurring image. Possible *interventions* into the computational process of generating the artificial image will correspond (to a degree depending on the fidelity of the recapitulation process) to the causal consequences of interventions in the real world. This is, perhaps, most evident in the context of interactive virtual worlds: A game player will have the sense of the "reality" of the virtual world, to the extent that the interventions made by the player, perhaps through an avatar, lead to a chain of causal consequences that would be observed in the real world. Achieving this requires the virtual world to recapitulate, *in real-time*, aspects of the causal relationships present in the natural world.

Thus, it is natural to suggest that, to the extent that our mental imagery creates a convincing simulacrum of the real world, such imagery this must, to some degree at least, recapitulate aspects of the causal structure of that world. And such modifications to the causal structure, and their consequences, can be imagined with remarkable flexibility. So, in wondering whether we can carry a sofa up a winding staircase, we can imagine what would happen if the sofa were larger or smaller; whether it would help to remove the legs; how things would be different if the banister were rubber not wood; and so on.

We have so far focused on perception (and imagery). But how broadly can "perception" be interpreted? We suggest that the approach may apply quite broadly. After all, people readily attribute physical causality even to simple two dimensional shapes in "collision" or other simple interactions (e.g., Michotte, 1963/1946); but they are just as ready to attribute mental properties, such as aggression or timidity, to animated geometric shapes (e.g., Heider & Simmel, 1944). Indeed, people will even attribute propositional attitudes such as knowledge and desire to such shapes, on the basis of their movements (along the general lines of "the big triangle does not know where the circle and the small triangle are," "the circle wants to help but is too scared of the big triangle").[10] Moreover, such rich attributions are, of course, routine in our perception of everyday scenes, including people, faces, gestures, and speech—and more broadly there seems no natural boundary between perception and cognition (although see, e.g., Fodor, 1983). Consonant with this broad interpretation of the scope of perception is the fact that the notion that perception works by inverting causal generative models can fruitfully been applied vary widely, from fairly low-level perception (e.g., Weiss, Simoncelli, & Adelson, 2002), to understanding naïve physics of colliding particles (Sanborn, Mansinghka, & Griffiths, 2009), and inferring the "hidden intentions" of simple moving animated agents (Baker, Saxe, & Tenenbaum, 2009) and intentions revealed by ethically contentious decisions (Sloman, Fernbach, & Ewing, 2012). We may conjecture that, if perception arises from the application of causal generative models, such models must capture regularities of all kinds, whether physical or social.

## 4. Causal models in communication and language processing

The general approach we have outlined so far can be explored in a range of other domains. Here, we briefly consider how the same approach may apply to language. Note that the relation between perception and imagery maps on to the relation between language understanding and language generation; and the ability to imagine how the world might *look* different maps on to our ability to imagine what *might* have been said. That is, we speculate that language understanding requires drawing on a causal model of the process by which language is generated.

Now this model will, presumably, contain two types of causal process. One type of process (if we restrict ourselves to spoken language, for concreteness) concerns the acoustic analogs of the processes that give rise to visual inputs. The world consists of a wide variety of processes that, either directly or indirectly, create local changes in air pressure; the resulting sound waves propagate, are selectively absorbed, and reflected, in the environment, and finally pass through the complex machinery of the outer, middle, and inner ear. The human auditory system must invert this causal process to distinguish, identify, and localize the objects and events giving rise to the auditory scene (Bregman, 1990). In the case of speech, however, the listener must also invert a second type of causal process: the sequence of computational operations within the mind of the speaker that lead from the intention to communicate to the movements of the speech articulatory apparatus (vocal folds; tongue, lips, jaw, and so on).[11] Indeed, the idea that speech input is coded in terms of articulatory gestures, the motor theory of speech perception (e.g., Liberman & Mattingly, 1985), has a long history.

Now because the listener is also typically a speaker of the language, the causal generative model for language does not have to be created from scratch. Instead, it can borrow from causal processes underlying the speaker's own utterances. This is one reason to expect close links, at all levels of language processing, between comprehension and production. A particularly simple but compelling illustration is that people can easily and fluently complete each other's sentences (Clark, 1996; Pickering & Garrod, in press). Neural (e.g., Wilson, Saygin, Sereno, & Iacoboni, 2004) and behavioral (Pickering & Garrod, in press) evidence appears to back up this connection. Moreover, this picture is consonant with apparent cognitive and neural integration between production and perception for non-linguistic actions (Prinz, 1997; Schütz-Bosbach & Prinz, 2007).

Rather than attempt an exhaustive review and discussion, we focus on two issues which illustrate the potential breadth of the relevance of causality for understanding language and communication: the mental representation of different aspects of syntax, and theory of mind and pragmatics.

### *4.1. Causality and syntactic representation*

The project of generative grammar appears to fit nicely within the present framework (Chomsky, 1957). We noted above that the definite clause grammar (DCG) in Prolog provides an illustration of how grammatical rules can map directly into a program for both producing and parsing language (although the DCG is, of course, far too simple to capture the enormous subtlety of natural language syntax). The same general picture is reflected in a wide range of computational models of language processing, often using Bayesian methods to choose the most probable interpretations in the light of the massive local ambiguity of natural language (e.g., Klein & Manning, 2004).

How can we determine what linguistic information is mentally represented, using the present account? As we discussed in the context of vision, we can potentially gain insights into the data structures underlying language by directed *top-down control*: In essence, given a sentence, we can consider how that sentence might have been different.

For example, given the rather unpromising sentence *John really likes the green bird*, we can immediately envisage sentences expressing any number of alternative "likers," any number of other things liked, different relations (loving, hating,…), and so on; we see that *really* might have been eliminated or intensified (*really and truly*); that the agent–patient relation might have been reversed; that the sentence might have been in a past or future tense; and so on. Notice that the very possibility of such manipulations immediately suggests that our top-down influence operates on highly abstract data structures (roughly, at the level of abstraction agent–patient-relation, or perhaps subject–object–verb; and involving rich information about tense, aspect, etc.). Notice, by contrast, that we cannot imagine transpositions, additions, or deletions of arbitrary strings of words; and still less of manipulations of streams of raw acoustic material.[12]

These rather casual observations are, of course, no substitute for precise linguistic analysis and psycholinguistic experimentation. But this line of reasoning is potentially heuristically useful; and, indeed, it illustrates a style of argument often used in linguistics (without mention of causality or counterfactuals, of course).

Consider, for example, the variety of structural manipulations we can imagine operating on our sentence, that is, ways in which we can imagine it might have been different. These include topicalization, for example, *The green bird, John really likes*; passivization: *the green bird is really liked by John*; clefting: *It is the green bird that John really likes*; use with proforms (e.g., *John really likes it*; coordination: *John really likes the green bird and the blue fish*). Various interesting generalizations may be drawn from patterns of this kind. Here, we note only that these observations are typically used to establish *constituency*: that is, which word strings correspond to linguistic *units*, that is, to assign hierarchical phrase structure to the sentence. So, in such examples, the string *the green bird* maintains its integrity; the string *bird that John* does not. This suggests that *the green bird* is represented as a unit in some data structure over which various manipulations operate—and hence that this unit moves "as a whole," just as the fact that our ability to imagine visual objects (but not arbitrary chunks of 2D visual input) moving, disappearing, or being modified in some way suggests that these are represented as units in the visual system. From this perspective, controversy over the linguistic notion of constituency (regarding, NP, VP, etc.) may usefully be reinterpreted as concerning causal processes involved in sentence generation.

We have considered whether constituents are mentally represented (i.e., modifiable in the causal processes which generate sentences). This appears to suggest something like a hierarchical grouping of linguistic material (although some accounts, and indeed some constituency tests, may suggest somewhat more flexible notions of constituency, so that both *John loves* and *loves Mary* might both be constituents; Steedman, 2001). This raises the following intriguingly simple possibility that constituents are that which is represented (and modifiable) in the data structures over which syntactic rules operate; syntactic rules are the algorithms operating over those constituents (and are not modifiable, and indeed, not represented).[13]

But what about labeled syntactic *trees*? As ever, we can postulate that trees are mentally represented only if they are causally efficacious; and this can be demonstrated only if such trees can be *modified* and the impact of these results observed. If, in line with early transformational grammar, the relation between *John really likes the green bird* and questions such as *What is it that John really likes?* is a transformation (i.e., a particular type of modification) defined over syntactic trees, then, prima facie this suggests the existence of such trees. There are, of course, a range of other linguistic perspectives which do not postulate any such transformational relationship—and hence do not necessarily require computation of a syntactic tree. Indeed, in the DCG, the syntactic tree is no more than a trace of processing operations and need not be represented explicitly at all. In this spirit, Steedman (2001) developed an account, using Combinatory Categorial Grammar, in which syntactic trees are merely traces of processing operations that combine semantic representations: They are not represented in any data structure.

Note, too, a further source of information about what is represented arises not from active top-down manipulation but "spontaneous" and inadvertent modifications to data structures (perhaps memory buffers at various levels of linguistic analysis, e.g., Levelt, 1998) involved in planning speech. So, for example, suppose transposition errors typically switch phonetic features, phonemes, or words (e.g., Fromkin, 1973). Prima facie, such errors correspond to "interventions" on the data structures encoding phonetic features, phonemes, or words (although it is not necessarily obvious precisely what is being transposed). More abstract concepts such as past tense markers appear also to be shifted, as in the celebrated **Rosa always date shranks*, instead of the intended *Rosa always dated shrinks* (Fromkin, 1973). Here, an abstract past tense marker seems to have shifted to attach to the "wrong" target: Its abstract nature is indicated by the fact that the phonological instantiation of the past tense marker is different in the two cases (*date*→*dated*; *shrink*→*shrank*).

### 4.2. *Counterfactuals, theory of mind, pragmatics*

Understanding the minds of other agents seems fundamental to human cognition. It is typically assumed that we attempt to assign beliefs and preferences to other agents, so that the agent's utterances and actions make sense as far as possible (e.g., Davidson, 1984; Quine, 1960).[14] Assessing what actions or utterances make best sense requires considering what other options were available but rejected; and this involves, of course, considering counterfactuals. Consider the celebrated experiment by Gergely, Bekkering, and Király (2002) in which 14-month infants watch an experimenter press a button in front of them with her forehead. When the experimenter's hands are occupied, the infant "copies" by pressing the button with his own hands; when the experimenter's hands are free, the infant attempts to press the button with his own forehead. Thus, it appears that observing that the hands are occupied rules out a counterfactual possibility (i.e., that the experimenter could have touched the button with her hands) and suggests to the infant that the experimenter may be aiming to press the button by any means available; pressing with the forehead just happened to be the easiest such means. When the hands are not occupied, by contrast, this interpretation cannot be right: If the intention were "merely" to press the button, the hands would surely have been used rather than the head. But to realize this is, of course, precisely to reason about a counterfactual. The infant infers the experimenter's intentions must have been to press the button with the head; and the infant therefore copies both means and end. This type of counterfactual reasoning arises in any model of attempting to make the best sense of another's mind, given her actions or utterances: because the postulated beliefs, intentions, and other mental properties only make sense in the light of counterfactual comparisons concerning what the person *might* have thought, said, and done.

The same point applies in understanding others' intentions by observing their utterances, as well as actions. So, in natural language pragmatics, counterfactual reasoning concerning mental states seems ubiquitous. For example, anything which is *on* the table is, geometrically, *above* the table. But on hearing *the light is above the table*, we assume the light is suspended over the table, because of the following counterfactual: If the light were *on* the table, the speaker would have used the more specific word *on*. Or consider *the marathon runner got to within feet of the finishing line*. This will be true of all runners who cross the line; but we tentatively infer that this particular runner did not cross the line, because if she had, the speaker would have said this directly.[15] The general moral is that the pragmatic interpretation of language attempts to make best sense of why the speaker chose a particular communicative action; and assessing this requires considering other things that might have been said. Moreover, as Gergely et al.'s (2002) experiment described above illustrates, the ability to carry out this type of counterfactual reasoning seems both to arise very early in human development and to be basic to the simplest interpretations of human action.

## 5. General discussion

There are long traditions in philosophy, psychology, and statistics that treat causality and counterfactuals with suspicion. It is tempting to focus on observables. As Hume (1739/1978) noted, we can observe one billiard ball striking a second; and the second moving away; and we can measure the spatial and temporary characteristics of the interaction. But we cannot observe any relation of *causation* between them (and our perceptual systems are, indeed, easily misled, e.g., Michotte, 1963/1946). In psychology, behaviorism emphasizes observed contingencies in the world—carefully focusing on observable associations between events, not causal relations. And, while practical people wish to understand the *causal* links between economic variables, or between diet and health, and so on, in order to help guide their actions, there remains skepticism concerning even the meaning of causal claims within the statistical community.[16]

Following Pearl (2000), we have argued for the opposite point of view: that counterfactuals and causation are fundamental to computation and cognition. A program is a causal system: the algorithm determines counterfactuals over potential interventions on the data structures over which it operates. Moreover, we have suggested that, to the extent that the perceptual system builds models of the environment (rather than, for example, there being direct perception-action mappings), these models are causal; and that language understanding, across a wide range of levels of analysis, involves reconstructing aspects of the causal process generating language. We suggest that the causal revolution that is gradually spreading across statistics will have equal significance for the future of cognitive science.

## Notes

1. According to many theorists, there is more to causality than counterfactuals (e.g., Menzies, 1999; Salmon, 1984; Sloman, 2005). We are neutral on this issue but will here use "causal" to mean just "counterfactual supporting," following in the

tradition of Lewis (1973) and Pearl (2000). We do not mean to presuppose the correctness of a counterfactual theory of causality more broadly.

2. Syntactically, these clauses can be viewed as production rules, as used widely in cognitive science (e.g., Anderson, 1983; Newell, 1994). In more complex DCG rules (e.g., which pass variables, e.g., to ensure subject–verb agreement), their computational properties are somewhat more elaborate.
3. Implemented, for technical convenience, as difference lists (L1, L2), that is, a pair of lists which denotes the contents of the list L1, once the latter part of the list, L2, is subtracted from it. So ([the, cat, sat, on, the], [on, the]) is one representation of the list [the, cat, sat].
4. Such bizarre counterfactuals might be ruled out by introducing *typing* into our data structure; that is, only strings of the right type could be substituted, as is common in computer languages. Types might correspond to syntactic categories, such as NP, Det, N, and so on. From the present perspective, typing serves to restrict the counterfactuals that can be considered (e.g., helping to ensure, for example, that the program does not crash or loop). But there may be empirical evidence that such typing restrictions apply to possible interventions in, for example, the language processing system. After all, in speech errors, people generally substitute within category, switching, for example, nouns for other nouns, providing potential evidence for the "psychological reality" of such categories (e.g., Garrett, 1975).
5. Note that this formulation identifies a program by what it *does*, not as a string of symbols describing what it does in a programming language (e.g., Java, C++, etc.). In this usage, the same program might be "written" in many ways, and in many different programming languages.
6. Many languages are compiled, not interpreted; that is, they are translated into another language and the resulting code is then implemented. We ignore this more indirect process here for simplicity. Note, too, that there is a variety of types of programming language, from imperative languages to functional and declarative languages. Theories of their counterfactual properties will, in consequence, be very different.
7. We do not mean to downplay the importance of non-causal descriptions in cognitive science or elsewhere. Many regularities can usefully be captured simply by outlining relationships between different entities, without making causal claims. Indeed, we have argued elsewhere that this is one natural perspective about rational explanation in cognitive science: that it spells out coherence relations between different aspects of thought and behavior but does not necessarily pin down a causal direction (Chater & Oaksford, 2012). For an earlier discussion of the role causality plays in perception and reasoning, see Chater and Oaksford (2006).
8. Of course, a *description* of an algorithm, as part of a computer program, can be viewed as having representational properties—indeed, the operational and denotational semantics of programs specify such properties (Winskel, 1993). But the description is itself represented a data structure, from the perspective of the interpreter or compiler.

9. A natural question is how the present approach relates to Marr's levels of analysis. We suggest that the present discussion is focused exclusively at the algorithmic level, that is, the specification of algorithms and the representations over which they operate. We note, though, that, as in conventional computers, it is quite likely that neural and cognitive processes can be viewed as implementing algorithms at a number of levels of explanation (e.g., from elementary computations in neural circuits to abstract symbolic computations).
10. We leave aside the question of whether intentional explanation, that is, explanation which relies on the *meaning* of mental states rather than their formal or physical properties, is properly viewed as causal. What is important, in this article, is that intentional explanation generates counterfactuals—for example, "if I had known you were in the next room, I wouldn't have shouted." There may be more to causal explanation than generating counterfactuals—but if so, this is not our focus here.
11. Of course, some theories of perception and speech processing do not suppose that any such model is built; and practical speech technology typically does not. We take no strong stance on this issue here for "low-level" aspects of speech. Note, though, that there is evidence of overlap between representations of the articulators in motor cortex and speech perception. For example, using Transcranial Magnetic Stimulation to disrupt the motor representation of the lip (rather than the hand) differentially modulates the perception of speech, rather than non-speech, signals (Mottonen, Dutton, & Watkins, 2013).
12. Modifications at a lower level are also possible. For example, we can readily follow the children's game of beginning each word with a specific phoneme (i.e., inserting a specific phoneme in the appropriate data structure), creating some strange non-sense along the lines of: **Pon peerly pikes pah preen pird.* But the very ease of such bizarre transformations suggests a code for word initial speech sounds. Interestingly, it appears much more difficult to generate the sentence with *all* consonants (or consonant clusters) replaced with a single phoneme, yielding something like *pop peepy pipes pah peep pirp. This difference perhaps suggests that the representation of the consonant/p/is not fully abstract with respect to location.
13. We seem unable easily to modify syntactic rules. But how then does language acquisition proceed? With concrete rules, such as in our DCG, acquiring a language would seem inevitably to require learning such rules. That is, following the present logic, it would seem that learning requires exploring the consequences of different possible interventions on the current set of rules; and this in turn seems to imply that learning requires actively exploring different sets of rules and their consequences. In short: *if rules can be learned, then they must be represented.* But there is an alternative possibility. In lexicalist approaches to generative grammar (e.g., Bresnan, 2001; Chomsky, 1995; Steedman, 2001), rules are extremely general and unconstrained (unlike the rules in our DCG above); so perhaps all language-specific information is represented in lexical entries (which, presumably,

*are* represented); and the rules are simple and invariant. According to this viewpoint, language acquisition involves learning new lexical entries (or, in some frameworks, larger units such as constructions, e.g., Goldberg, 2006); rules define the "laws of nature" which determine how lexical entries combine; but cannot themselves be modified.

14. On some readings, for example, in rational choice theory (e.g., Allingham, 2002), making sense can be view as something like "being consistent according to a normative standard, such as Bayesian decision theory."
15. See, for example, Grice (1989) and Levinson (2000), for rich theoretical explorations of pragmatic reasoning along these lines.
16. At least, outside domains where controlled experiments are possible—and the results of controlled experiments are surely not available to a cognitive agent in acquiring knowledge of the perceptual world, language, other minds, or just about any other interesting cognitive domains.

## Acknowledgments

We would like to thank Howard Bowman, William Jimenez Leal, Steven Sloman, and an anonymous reviewer for insightful comments on the ideas in this paper. Nick Chater was partially supported by ERC Advanced Grant 295917-RATIONALITY and also by a grant Decision Time for Free Will from the John Templeton Foundation.

## References

Allingham, M. (2002). *Choice theory: A very short introduction*. Oxford, England: Oxford University Press.

Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review*, *85*, 249–277.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Baker, C. L., Saxe, R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, *113*, 329–349.

Bregman, A. S. (1990). *Auditory scene analysis: The perceptual organization of sound*. Cambridge, MA: Bradford Books, MIT Press.

Bresnan, J. (2001). *Lexical functional syntax*. Oxford, England: Blackwell.

Chater, N., Goodman, N., Griffiths, T., Kemp, C., Oaksford, M., & Tenenbaum, J. (2011). The imaginary fundamentalists: The unshocking truth about Bayesian cognitive science. *Behavioral & Brain Sciences*, *34*, 194–196.

Chater, N., & Oaksford, M. (2006). Mental mechanisms: Speculations on human causal learning and reasoning. In K. Fiedler & P. Juslin (Eds.), *In the beginning there is a sample: Information sampling as a key to understanding adaptive cognition* (pp. 210–238). Cambridge, England: Cambridge University Press.

Chater, N., & Oaksford, M. (Eds.) (2008). *The probabilistic mind*. Oxford, England: Oxford University Press.

Chater, N., & Oaksford, M. (2012). Normative systems: Logic, probability, and rational choice. In K. Holyoak & R. Morrison (Eds.), *The Oxford handbook of thinking and reasoning* (pp. 11–21). New York: Oxford University Press.

Chomsky, N. (1957). *Syntactic structures*. The Hague, The Netherlands: Mouton.

Chomsky, N. (1995). *The minimalist program*. Cambridge, MA: MIT Press.

Clark, H. H. (1996). *Using language*. Cambridge, England: Cambridge University Press.

Davidson, D. (1984). *Inquiries into truth and interpretation*. Oxford, England: Oxford University Press.

Fodor, J. A. (1983). *Modularity of mind*. Cambridge, MA: MIT Press.

Fodor, J. A., & Garrett, M. F. (1967). Some determinants of sentential complexity. *Perception and Psychophysics*, *2*, 289–296.

Fromkin, V. A. (1973). *Speech errors as linguistic evidence*. The Hague, The Netherlands: Mouton.

Garrett, M. F. (1975). The analysis of sentence production. In G. H. Bower (Ed.), *The psychology of learning and motivation, Volume 9: Advances in research and theory* (pp. 133–177). New York: Academic Press.

Gergely, G., Bekkering, H., & Király, I. (2002). Developmental psychology: Rational imitation in preverbal infants. *Nature*, *415*, 755.

Glymour, C. (2001). *The mind's arrows: Bayes nets and graphical causal models in psychology*. Cambridge, MA: MIT Press.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language*. Oxford, England: Oxford University Press.

Gopnik, A., Glymour, C., Sobel, D. M., Schulz, L. E., Kushnir, T., & Danks, D. (2004). A theory of causal learning in children: Causal maps and Bayes nets. *Psychological Review*, *111*, 3–32.

Gopnik, A., Sobel, D., Schulz, L., & Glymour, C. (2001). Causal learning mechanisms in very young children: Two, three, and four-year-olds infer causal relations from patterns of variation and covariation. *Developmental Psychology*, *37*, 620–629.

Gopnik, A., & Tenenbaum, J. B. (2007). Bayesian networks, Bayesian learning, and cognitive development. *Developmental Science*, *10*, 281–287.

Grice, H. P. (1989). *Studies in the way of words*. Cambridge, MA: Harvard University Press.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences*, *14*, 357–364.

Griffiths, T. L., Kemp, C., & Tenenbaum, J. B. (2008). Bayesian models of cognition. In R. Sun (Ed.), *Cambridge handbook of computational psychology* (pp. 59–100). New York: Cambridge University Press.

Hahn, U., & Chater, N. (1998). Similarity and rules: Distinct? Exhaustive? Empirically distinguishable? *Cognition*, *65*, 197–230.

Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *American Journal of Psychology*, *57*, 243–259.

Hume, D. (1978). *A treatise of human nature*. Oxford: Oxford University Press. (Original work published 1739).

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of bayesian models of cognition. *Behavioral and Brain Sciences*, *34*, 169–231.

Jordan, M. I. (Ed.) (1999). *Learning in graphical models*. Cambridge, MA: MIT Press.

Klein, D., & Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In D. Scott (Ed.), *Proceedings of the Association for Computational Linguistics (ACL)* Stroudsburg, PA: ACL.

Kosslyn, S. M. (1980). *Image and mind*. Cambridge, MA: Harvard University Press.

Kosslyn, S. M., Thompson, W. L., & Ganis, G. (2006). *The case for mental imagery*. Oxford, England: Oxford University Press.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, *50*, 157–224.

Levelt, W. J. M. (1998). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.

Levinson, S. (2000). *Presumptive meanings*. Cambridge, MA: MIT Press/Bradford Books.

Lewis, D. (1973). Causation. *Journal of Philosophy*, *70*, 556–67.

Liberman, A. M., & Mattingly, I. G. (1985). The motor theory of speech perception revised. *Cognition*, *21*, 1–36.

Marr, D. (1982). *Vision*. San Francisco: W. H. Freeman.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: Connectionist and dynamical systems approaches to understanding cognition. *Trends in Cognitive Sciences*, *14*, 348–356.

Menzies, P. (1999). Intrinsic versus extrinsic conceptions of causation. In H. Sankey (Ed.), *Causation and laws of nature* (pp. 313–29). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Michotte, A. (1963/1946) *The perception of causality*. J. R. Miles & E. Miles (trans.). London: Methuen (originally published, 1946).

Mottonen, R., Dutton, R., & Watkins, K. E. (2013). Auditory-motor processing of speech sounds. *Cerebral Cortex*, *23*, 1190–1197.

Newell, A. (1994). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Oaksford, M., & Chater, N. (2007). *Bayesian rationality*. Oxford, England: Oxford University Press.

Pearl, J. (1985). *Bayesian networks: A model of self-activated memory for evidential reasoning*. Computer Science Department, University of California.

Pearl, J. (1988). *Probabilistic reasoning in intelligent system: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge, England: Cambridge University Press.

Pereira, F. C. N., & Warren, D. H. D. (1983). *Parsing as deduction*. In M. Marcus (Ed.), *Proceedings of the 21st annual meeting on association for computational linguistics* (pp. 137–144). Morristown, NJ: Association for Computational Linguistics.

Pickering, M. J., & Garrod, S. (in press). An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*.

Prinz, W. (1997). Perception and action planning. *European Journal of Cognitive Psychology*, *9*, 129–154.

Pylyshyn, Z. W. (1984). *Computation and cognition: Towards a foundation for cognitive science*. Cambridge, MA: MIT Press.

Quine, W. V. O. (1960). *Word and object*. Cambridge, MA: Harvard University Press.

Rock, I. (1985). *The logic of perception*. Cambridge, MA: Bradford Books.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Russell, B. (1919). *Introduction to mathematical philosophy*. London: George Allen & Unwin.

Salmon, W. (1984). *Scientific explanation and the causal structure of the world*. Princeton, NJ: Princeton University Press.

Sanborn, A. N., Mansinghka, V. K., & Griffiths, T. L. (2009). A Bayesian framework for modeling intuitive dynamics. In N. Taatgen & H. van Tijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society*.

Schütz-Bosbach, S., & Prinz, W. (2007). Perceptual resonance: Action-induced modulation of perception. *Trends in Cognitive Sciences*, *11*, 349–355.

Shepard, R. N. (1984). Ecological constraints on internal representation: Resonant kinematics of perceiving, imagining, thinking, and dreaming. *Psychological Review*, *91*, 417–447.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, *171*, 701–703.

Sloman, S. A. (2005). *Causal models: How people think about the world and its alternatives*. New York: Oxford University Press.

Sloman, S. A., Fernbach, P. M., & Ewing, S. (2012). A causal model of intentionality judgment. *Mind and Language*, *27*, 154–180.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction and search*. New York: Springer-Verlag.

Steedman, M. (2001). *The syntactic process*. Cambridge, MA: MIT Press.

Tenenbaum, J. B. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis. Cambridge, MA: MIT.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure and abstraction. *Science*, *331*, 1279–1285.

Weiss, Y., Simoncelli, E. P., & Adelson, E. H. (2002). Motion illusions as optimal percepts. *Nature Neuroscience*, *5*, 598–604.

Wilson, S. M., Saygin, A. P., Sereno, M. I., & Iacoboni, M. (2004). Listening to speech activates motor areas involved in speech production. *Nature Neuroscience*, *7*, 701–702.

Winskel, G. (1993). *The formal semantics of programming languages: An introduction*. Cambridge, MA: MIT Press.

Wirth, N. (1975). *Algorithms + Data Structures = Programs*. Englewood Cliffs, NJ: Prentice Hall.

Woodward, J. (1999). Causal interpretation in systems of equations. *Synthese*, *121*(2), 199–247.