How to Integrate Representation into Computational Modeling, and Why We Should

Abstract: I argue that Chalmers's proposed computational foundation conflicts with contemporary cognitive science. I sketch an alternative approach to modeling the mind computationally. On my alternative approach, computational models can individuate mental states in representational terms, without any appeal to organizationally invariant properties. I develop my approach through case studies drawn from cognitive psychology, CS, and AI.

§1. Computation and representation

The computational theory of mind (CTM) holds that mental processes are computations. Chalmers (2012) offers an especially well-developed version of CTM. His account hinges upon two key definitions:

- The *causal topology* of a system is "the pattern of interaction among parts of the system, abstracted away from the make-up of individual parts and from the way the causal connections are implemented" (2012, §3.1).
- A property *P* is *organizationally invariant* just in case "any change to the system that preserves the causal topology preserves *P*" (2012, §3.1).

According to Chalmers, a computational model individuates computational states through organizationally invariant properties. Computation in this sense "provides a general framework for the explanation of cognitive processes" (2012, §3.3). A mental state may have many properties that are not organizationally invariant, such as representational, phenomenal, or neural properties. Chalmers concedes that we can *supplement* computational explanation by citing such properties. But “there is a clear sense in which they are not vital to the explanation” (2012, §3.3). Cognitive science, at its core, studies organizationally invariant properties.

I disagree. Numerous cognitive science explanations individuate mental states through their representational properties. Typically, those properties are not organizationally invariant. By elevating organizational invariance over intentionality, Chalmers flouts contemporary scientific practice. I will present an alternative version of CTM that places representation at center stage. On my approach, computational theories can individuate mental states in representational terms. A computational model can specify a transition function over mental states *type-identified by their representational import*. Computational explanations need not mention organizationally invariant properties.

§2. Bayesian psychological modeling

Bayesian models of mental activity have proved explanatorily successful, especially within *perceptual psychology*. I will present basic elements of the Bayesian framework, taking perception as my primary case study.

The perceptual system estimates states of the environment, including shapes, sizes, and motions of distal objects. As Helmholtz emphasized, perceptual estimation faces an underdetermination problem. Proximal sensory stimulation underdetermines distal states. Helmholtz postulated that the perceptual system solves this underdetermination problem through “unconscious inference.” Bayesian models elaborate Helmholtz’s approach (Knill and Richards, 1996). They treat the perceptual system as executing an unconscious *statistical* inference, governed by Bayesian norms. The perceptual system assigns *prior probabilities* $p(h)$ to hypotheses h about the environment. It also assigns *prior likelihoods* $p(e \mid h)$, reflecting the probability of sensory input *e* given *h*. Upon receiving sensory input *e*, the perceptual system reallocates probabilities across the hypothesis space through *conditionalization*, yielding a *posterior probability* $p(h \mid e)$:

$$p(h \mid e) = \eta\, p(h)\, p(e \mid h),$$

where η is a normalizing constant to ensure that probabilities sum to 1. Based on the posterior, the perceptual system selects a unique hypothesis *h*. It may select the hypothesis that maximizes the posterior, or it may deploy a more complex selection rule.

Example: Shape from shading

Retinal stimulations underdetermine shapes of perceived objects. For instance, a perceived object might be convex and lit from overhead, or it might be concave and lit from below. The same retinal input would result either way. How do we reliably perceive distal shape based upon inherently ambiguous retinal input? Let *s* reflect possible shapes, θ possible lighting directions, and *e* possible patterns of retinal illumination. According to current Bayesian models (Stone, 2011), the visual system encodes:

> A prior probability $p(s)$, which assigns higher probabilities to certain distal shapes than others (e.g. it may assign higher probability to convex shapes).
>
> A prior probability $p(\theta)$, which assigns much higher probability to an overhead lighting direction than to alternative lighting directions.
>
> A prior likelihood $p(e \mid s, \theta)$, which assigns higher probability to an (e, s, θ) triplet if distal shape *s* and lighting direction θ are likely to cause retinal illumination *e*.

Upon receiving retinal input *e*, the perceptual system redistributes probabilities over shape-estimates, yielding a posterior $p(s \mid e)$. Based on this posterior, the visual system selects a final shape-estimate *s*. Since light usually emanates from overhead, the final shape-estimate is usually accurate, or at least approximately accurate. □

Example: Surface color perception

Light reflected from a surface generates retinal stimulations consistent with various surface colors. For instance, a surface might be red and bathed in daylight, or it might be white and bathed in red light. How do we reliably perceive surface colors? To a first approximation, current Bayesian models operate as follows (Brainard, 2009). A surface has reflectance $R(\nu)$, specifying the fraction of incident light reflected at each wavelength ν.[1] The illuminant has spectral power distribution $I(\nu)$: the light power at each wavelength. The retina receives light spectrum $C(\nu) = I(\nu)\, R(\nu)$ from the surface. The visual system seeks to estimate surface reflectance $R(\nu)$. This estimation problem is underdetermined, since $C(\nu)$ is consistent with numerous $I(\nu)$-$R(\nu)$ pairs. Bayesian models posit that two surfaces have the same color appearance for a perceiver when her perceptual system estimates the same reflectance for each surface.[2] To estimate $R(\nu)$, the visual system estimates $I(\nu)$. It does so through a Bayesian inference that deploys a prior over illuminants and reflectances. Roughly speaking, the prior assigns higher probability to illuminants that resemble natural daylight and to reflectances that occur more commonly in the natural environment. This framework can explain successes and failures of human color perception under various experimental conditions. □

[1] The models described in this paragraph assume *diffusely illuminated flat matte surfaces*. To handle other viewing conditions, we must replace $R(\nu)$ with a more complicated property, such as a *bidirectional reflectance distribution*. My talk about "surface reflectance" should be construed as allowing such generalizations.

[2] Such models need not *identify* colors with surface reflectances. For instance, one might combine such models with the familiar view that colors are dispositions to cause sensations in normal human perceivers.

Cognitive scientists have successfully extended the Bayesian paradigm beyond vision to diverse phenomena, including *sensorimotor control* (Bays and Wolpert, 2007), *language parsing* and *acquisition* (Chater and Manning, 2006), "central" cognitive processes such as *concept acquisition* and *causal reasoning* (Chater and Oaksford, 2008), and non-human *navigation* (Cheng, Shettleworth, Huttenlocher, and Rieser, 2007). A core postulate underlying all these models is that the mind reallocates probabilities over a "hypothesis space." The "hypotheses" may concern: bodily configurations; or parsing trees for utterances; or causal relations among events; or locations in the spatial environment; and so on.

What are these "hypotheses"? For present purposes, the key point is that current Bayesian psychological models individuate hypotheses in representational terms. For instance, Bayesian perceptual psychology describes how the perceptual system reallocates probabilities over hypotheses *about specific environmental properties*. Bayesian models of shape perception assume that the perceptual system begins with a prior probability *over estimates of specific shapes*. The models describe how retinal input prompts reallocation of probabilities *over estimates of specific shapes*. Similarly, Bayesian models of surface color perception describe reallocation of probabilities over *estimates of specific reflectances*. Bayesian models describe how the perceptual system, exercising standing capacities to represent *specific environmental properties*, transits from retinal input to perceptual states that estimate *specific environmental properties*. Perceptual psychology classifies perceptual states through representational relations to specific environmental properties.[3] A similar diagnosis applies to Bayesian psychological modeling more generally. Current science postulates probabilistic updating over hypotheses that represent environmental states or properties: bodily configurations; parsing trees of utterances;

[3] For extended defense of my analysis, see (Rescorla, forthcoming b). My analysis is heavily influenced by Burge (2010a, pp. 82-101, pp. 342-366).

causal relations; and so on. The science individuates hypotheses through representational relations to specific environmental states or properties.

§3. Organizational invariance?

Many representational properties do not supervene upon the thinker's internal neurophysiology. The classic illustration is Putnam's (1975) Twin Earth thought experiment, which Burge (2007) applies to mental content. Quite plausibly, one can extend the Twin Earth methodology to many mental states, including numerous representational mental states cited within Bayesian psychology. For example, Block (2003) mounts a convincing case that the perceptual states of neural duplicates can represent different surface reflectances, if the duplicates are suitably embedded in different environments. Chalmers (2006) himself argues that a brain suitably linked to a Matrix-style computer simulation would not represent distal shapes, reflectances, and so on. I conclude that cognitive science employs an *externalist* explanatory template: the science taxonomizes mental states partly through representational properties that do not supervene upon internal neurophysiology (Burge, 2007, 2010).

Neurophysiological duplicates share the same causal topology. If a property does not supervene upon internal neurophysiology, then it does not supervene upon causal topology. So numerous representational properties cited within cognitive science are not organizationally invariant. For that reason, Chalmers's organizationally invariant paradigm diverges fundamentally from the explanatory paradigm employed within actual cognitive science. Cognitive science may describe certain phenomena in organizationally invariant terms. But it studies numerous phenomena through a representational paradigm *as opposed to* an organizationally invariant paradigm. For example, Bayesian models of surface color perception cite representational relations to specific reflectances, without even mentioning organizationally invariant properties.

Representational individuation is not essential to Bayesian modeling *per se*. In principle, one can imagine a Bayesian model that individuates hypotheses in organizationally invariant terms. However, any such model would differ significantly from *the Bayesian models employed within actual cognitive science*.

Chalmers might suggest that we can reinterpret current psychology in organizationally invariant terms. A Bayesian theory *T* delineates a pattern of causal interaction among mental states. It thereby determines a causal topology. One can describe this causal topology through an organizationally invariant theory *T**. *T** might even cite some kind of "narrow content" that supervenes upon causal topology. Why not replace *T* with *T**? After all, both theories predict the same relations between sensory inputs and motor outputs. As Chalmers puts it, "[a] system's behavior is determined by its underlying causal organization," so organizationally invariant descriptions "provide a general framework for the explanation of behavior" (2012, §3.3).

I respond that cognitive science does not simply map sensory inputs to behavioral outputs. It explains mental states *under intentional descriptions*. For example, Bayesian perceptual psychology studies perceptual estimation of environmental states. It seeks to explain veridical perception (*How does the perceptual system accurately estimate environmental states, even though sensory stimulations underdetermine those states?*), illusions (*Why does the perceptual system form an incorrect estimate in certain circumstances?*), and constancies (*How does the perceptual system estimate that a distal property --- such as shape --- remains constant despite large changes in retinal input?*). Thus, the science seeks to explain intentional properties of perceptual states. To illustrate, consider shape perception. Bayesian models postulate probabilistic updating over hypotheses individuated through representational relations to specific shapes. Current science delineates explanatory generalizations that describe how the perceptual system transits from retinal input to estimates *of specific shapes*. Using these generalizations, we explain why a perceiver enters into a perceptual state *that represents a specific distal shape*.

Organizationally invariant theories ignore representational relations to specific distal shapes. More generally, organizationally invariant theories ignore numerous representational properties that figure as *explananda* within contemporary cognitive science. Thus, Chalmers's favored paradigm sacrifices key benefits offered by current science. Organizationally invariant theories cannot explain mental states under desired intentional descriptions. If current science is on the right track, then Chalmers's version of CTM does not provide "a general framework for the explanation of cognitive processes."

Scientific practice is not sacrosanct. One might argue that scientists are confused or otherwise misguided. In particular, one might attack *intentionality* (or *representationality*) as problematic. Beginning with Quine (1960), various philosophers have argued that intentionality deserves no place in serious scientific discourse. Most relevantly for us, Stich (1983) argues that we should replace intentional psychology with a purely syntactic version of computational psychology. Chalmers sometimes hints that he harbors broadly Quinean worries about the scientific credentials of intentionality (2012, §2.2).

I will not review the well-known Quinean arguments that intentionality is illegitimate or unscientific. I agree with Burge (2010a, pp. 296-298) that those arguments are unconvincing. In any event, I favor an opposing methodology. I do not dictate from the armchair how scientific psychology should proceed. Instead, I take current scientific practice as my guide to clarity, rigor, and explanatory success. I examine how science individuates mental states, and I take that individuative scheme as my starting point. This methodology provides strong reason to embrace intentional explanation and scant reason to embrace organizationally invariant explanation, at least for certain core mental phenomena.

In note 6 [added in 2011], Chalmers modifies his position. He concedes that we cannot wholly explain intentional aspects of mental activity within his organizationally invariant framework. Nevertheless, he insists that his framework "can undergird intentional explanation when appropriately supplemented, perhaps by phenomenal and environmental elements." This modified position seems to allow a valuable explanatory role for organizationally invariant description *and* representational description.

In my view, the key question here is whether we have any reason to seek organizationally invariant explanations. Taking current science as our guide, there is excellent reason to believe that a complete psychology will cite representational relations to the environment. A complete theory will also feature non-representational neural descriptions, so as to illuminate the neural mechanisms that implement Bayesian updating (Knill and Pouget, 2004). But neural description is not organizationally invariant. Should we supplement representational and neural description with a third organizationally invariant level of description? We *can* describe the mind in organizationally invariant terms. My question is whether we *should*. Why insist that science describe the mind's causal topology in abstraction from representational and neural properties? Chalmers must show that psychological processes fall under explanatorily fruitful organizationally invariant descriptions.

Science does not usually study organizationally invariant properties. One can specify any physical system's causal topology, but the result usually lacks scientific interest. To borrow Chalmers's example, the science of digestion cites phenomena (such as energy extraction) that outstrip any relevant causal topology. In studying digestion, we are not merely studying a causal topology that mediates between food inputs and waste outputs. We are studying *digestive processes* that *instantiate* a causal topology. Anyone who suggests that we supplement digestive science with organizationally invariant descriptions owes us an excellent argument. Similarly, Bayesian psychology does not merely study a causal topology that mediates between sensory inputs and behavioral outputs. It studies *representational mental processes* that *instantiate* a causal topology. Anyone who suggests that we supplement scientific psychology with organizationally invariant descriptions owes us an excellent argument.

One popular argument emphasizes *generality*. Suppose we compare intentional psychological theory *T* and organizationally invariant theory *T**. *T** is more general than *T*, since *T** applies equally well to causal-topological duplicates who have different representational properties. Doesn't increased generality yield explanatory dividends? (Cf. Egan, 1999.)

This argument faces a serious problem: it overgeneralizes. Analogous arguments would show that we should supplement any other scientific theory, such as our theory of digestion, with organizationally invariant descriptions. Clearly, most sciences do not include such descriptions. Why? Because increased generality is not always an explanatory desideratum. One must isolate *the right kind* of increased generality. Increased generality has "the right kind" when it contributes explanatory power. To show that organizational invariance contributes explanatory power to cognitive science, one must advance an argument geared specifically to mental activity. One must argue that mental processes, unlike most other processes, fall under explanatorily fruitful organizationally invariant descriptions.

Chalmers provides an argument along these lines (2012, §3.2). He cites functionalism in the style of Lewis (1972): a psychological state is individuated by how it mediates between inputs, outputs, and other psychological states. From his functionalist premise, Chalmers concludes that scientific psychology requires organizationally invariant descriptions.

Lewis offers functionalism as a conceptual analysis of folk psychology. Yet folk psychology routinely cites representational properties that do not supervene upon causal topology. So it is unclear how widely, if at all, Lewis-style functionalist reduction applies to folk psychology. More importantly, folk psychology is not directly relevant to our concerns. Science can consult folk psychology for inspiration --- as illustrated by the Bayesian paradigm. But science does not answer to folk psychology. Our question is how scientific explanations should individuate mental states. Our best strategy for answering that question is to examine science, not folk psychology. As I have argued, there are numerous mental phenomena that current science studies by citing representational properties *as opposed to* Lewis-style functional properties.

Chalmers suggests a further argument for embracing organizationally invariant descriptions: they are needed for modeling the mind *mechanistically*. He claims that his approach "provides a general framework for the mechanistic explanation of cognitive processes and behavior" (note 6). Is he right? Must mechanical explanation operate in organizationally invariant fashion? I will now argue otherwise. Genuinely computational models can individuate mental states through their representational properties, including representational properties that do not supervene upon internal neurophysiology. Such models incorporate representationally-specified mechanical rules.[4]

§4. Computation, syntax, and semantics

A computational model specifies possible states of a system, and it delineates a transition function dictating how the system transits between states. The transition function may be either

[4] Burge (2010a, pp. 95-101), (2010b) and Peacocke (1994) propose similar treatments of computation.

deterministic or stochastic. Either way, as Chalmers emphasizes, it supports counterfactuals. In the deterministic case, it supports counterfactuals of the form:

If the system were in state *S*, then it would transit to state *S**.

In the stochastic case, it supports counterfactuals concerning the probability of transiting from state *S* from state *S**. Let us consider more carefully the nature of *S* and *S**. How does the transition function individuate computational states?

According to Chalmers, "computations are specified syntactically, not semantically" (2012, §2.2). To illustrate, consider a Turing machine table that describes how a scanner manipulates strokes on a machine tape. The machine table does not mention semantics. It describes formal manipulation of syntactic items. Chalmers holds that *all* computational models operate similarly: the transition function individuates computational states without regard to their representational import. A physical computing system may *have* representational properties, but we do not mention those properties when modeling how the system transits between states. This view of computation is quite popular (Egan, 1999), (Fodor, 1981), (Piccinini, 2008), (Stich, 1983). Chalmers incorporates it into a theory of *computational implementation*: a physical system executes the formal syntactic manipulations posited by some computational model just in case the system instantiates the causal topology dictated by the model (Chalmers, 2012, §2).

I agree that *some* computational models individuate computational states in syntactic, non-semantic fashion. But I contend that *other* computational models individuate computational states representationally. The transition function can individuate states *S* and *S** through their representational properties.

Example: A numerical register machine

A register machine contains a set of memory locations, called *registers*. A program governs the evolution of register states. The program may individuate register states syntactically. For instance, it may describe the machine as storing *numerals* in registers, and it may dictate how to manipulate those syntactic items. Alternatively, the program may individuate register states representationally. Indeed, the first register machine in the published literature models computation *over natural numbers* (Shepherdson and Sturgis, 1961, pp. 219). A program for this numerical register machine contains instructions to execute elementary arithmetical operations, such as *add 1* or *subtract 1*. A physical system implements the program only if can execute the relevant arithmetical operations. A physical system executes arithmetical operations only if it bears appropriate representational relations to numbers. Thus, a physical system implements a numerical register machine program only if it bears appropriate representational relations to numbers.[5] Notably, a numerical register machine program ignores *how* the physical system represent numbers. It applies whether the system's numerical notation is unary, binary, decimal, etc. The program characterizes internal states representationally (e.g. *a numeral that represents the number 20 is stored in a certain memory location*) rather than syntactically (e.g. *decimal numeral "20" is stored in a certain memory location*). It individuates computational states through denotational relations to natural numbers. It contains mechanical rules (e.g. *add 1*) that characterize computational states through their numerical denotations. □

A physical system often represents the same denotation in different ways. For example, "4" and "2+2" both denote the number 4. These two expressions occupy different roles within arithmetical computation. So denotation by itself does not always determine computational role.

[5] For further discussion of numerical register machines, see (Rescorla, forthcoming a).

In general, adequate computational models must address the *way* that a computational state represents its denotation. To borrow Frege's terminology, adequate models should individuate computational states by citing *modes of presentation* (MOPs). But what are MOPs?

Fodor (1981, pp. 234-241) offers a classic discussion of this question. He considers two ways of taxonomizing mental states:

- A *transparent* taxonomic scheme classifies mental states through their denotational properties.
- A *formal* taxonomic scheme ignores any representational relations that mental states bear to the environment.

For example, a transparent scheme type-identifies the belief that Hesperus has craters and the belief that Phosphorus has craters. This approach has trouble explaining why the two beliefs have different functional roles. In contrast, a formal scheme can associate the belief that Hesperus has craters and the belief that Phosphorus has craters with "formally distinct internal representations" (p. 240), thereby explaining why the beliefs have different functional roles. Fodor concludes: "a taxonomy of mental states which honors the formality condition seems to be required by theories of the mental causation of behavior" (p. 241). He posits a *language of thought* containing formal syntactic types. Formal syntactic type does not determine representational content: "mental representations can differ in content without differing in their intrinsic, formal, nonrelational, nonsemantic properties" (Fodor, 1991, p. 298). Fodor acknowledges that a mental representation *has* representational content. He insists that it *also* has a syntactic type compatible with diverse alternative contents.[6]

[6] In his early work, Fodor (1981, p. 227, p. 240) holds that formal syntactic type determines a unique *narrow* content but not a unique *wide* content. His later work, beginning in mid-1990s, abandons narrow content while retaining the emphasis on formal syntactic types that underdetermine wide content.

Fodor’s dichotomy between transparent and formal taxonomization ignores a third option. We can postulate MOPs individuated *partly by their representational import*. Frege proceeds in this way, as do such “neo-Fregeans” as Burge (2007, pp. 291-306), Evans (1982, pp. 100-105), and Peacocke (1992, pp. 16-27). I will pursue a neo-Fregean approach. I suggest that we postulate mental computations operating over *inherently meaningful* mental representations. The rest of my discussion unpacks this suggestion.

§5. Semantic permeation versus semantic indeterminacy

To develop my approach, I introduce some terminology. An entity is *semantically indeterminate* when it does not have its meaning essentially. The entity could have had a different meaning without any change in its fundamental nature, identity, or essence. Fodorian formal syntactic types are semantically indeterminate. A formal syntactic type might express different representational contents, depending upon how it figures in the thinker’s mental activity or her causal relations to the environment. In contrast, an entity is *semantically permeated* when it has its meaning essentially. We cannot change its meaning while holding fixed its fundamental identity, nature, or essence. For example, we can postulate a mental representation water that necessarily denotes water, a mental representation dog that necessarily denotes dogs, a mental representation 0 that necessarily denotes the number 0, and so on.[7]

Mental representations are *types*. We cite them to taxonomize token mental states. The types are abstract entities corresponding to our classificatory procedures. A semantically indeterminate type corresponds to a taxonomic scheme that underdetermines representational content. Different tokens of a semantically indeterminate type can express different contents. A

[7] Throughout my discussion, I use outline formatting to signal metalinguistic ascent. For example, “water” denotes the mental representation water, which in turn denotes the potable substance water.

semantically permeated type corresponds to a taxonomic scheme that takes representational content into account. Each token of a semantically permeated type expresses a uniform content.

Semantically permeated mental representations are either structured or primitive. Structured representations arise from applying compounding operations to primitive representations. For example, we can postulate a mental representation 𝕊 that necessarily denotes the successor function and a mental representation + that necessarily denotes the addition function. We can then postulate an infinite array of structured mental numerals that arise from appropriately combining 𝟘, 𝕊, and +:

𝟘 is a numeral.

If *w* is a numeral, then 𝕊*w* is a numeral.

If *v* and *w* are numerals, then ⦅*v* + *w*⦆ is a numeral.

The denotation of a complex numeral follows compositionally from the denotations of its parts:

𝕊*w* denotes the successor of the denotation of *w*.

⦅*v* + *w*⦆ denotes the sum of the denotations of *v* and *w*.

Each complex numeral necessarily satisfies an appropriate clause from the compositional semantics. Thus, each complex numeral is semantically permeated.

Semantically permeated taxonomization need not be transparent. ⦅𝕊𝕊𝟘 + 𝕊𝕊𝟘⦆ and 𝕊𝕊𝕊𝕊𝟘 both denote the number 4, but they are distinct types. Likewise, we can postulate distinct but co-referring mental types ℍ𝕖𝕤𝕡𝕖𝕣𝕦𝕤 and ℙ𝕙𝕠𝕤𝕡𝕙𝕠𝕣𝕦𝕤. Distinct but co-referring types reflect different *ways of representing* the same denotation.

A satisfactory development of the semantically permeated approach must elucidate how semantically permeated types are individuated. When do two token mental states share the same semantically permeated type? I want to leave room for conflicting answers to this question. But I follow Burge (2009) and Evans (1982, pp. 100-105) in assigning a central role to *representational capacities*. On my favored approach, we type-identify mental states by citing representational capacities deployed by those states. We reify the relevant mental state types by positing semantically permeated mental representations. For example, I have a capacity to represent water. That capacity is deployed by my belief *that water is thirst-quenching*, my desire *that I drink water*, and other mental states. We posit the mental symbol water so as to capture what those states have distinctively in common: exercise of the relevant representational capacity. A token mental state has the type water only if it deploys this capacity, to do which it must represent water. So water has fixed representational import *by its essential nature*.

By citing representational capacities, we illuminate what it is for semantically permeated mental representations to have "structure." Their structure consists in the appropriate joint exercise of distinct capacities. To illustrate, imagine an idealized mathematical reasoning agent with capacities to represent 0, successor, and addition. We posit mental symbols 0, S, and + so as to mark the exercise of those three capacities. The agent also has a capacity to apply functions to arguments. These four capacities yield complex capacities to represent natural numbers. Complex mental numerals mark the exercise of the resulting complex capacities. For example, mental numeral SSSS0 marks the exercise of a complex capacity that deploys three capacities:

A capacity to represent 0

A capacity to represent the successor function (deployed four times)

A capacity to apply a function to an argument (deployed four times)

A mental state is a token of SSSS0 only if it exercises this complex capacity, to do which it must satisfy the appropriate clause from the compositional semantics:

> 𝕊𝕊𝕊𝕊𝟘 denotes the successor of the successor of the successor of the successor of the denotation of 𝟘.

Likewise, mental numeral (𝕊𝕊𝟘 + 𝕊𝕊𝟘) marks the exercise of a complex capacity that combines the foregoing capacities in a different way, along with a capacity to represent addition.

As my examples illustrate, a single agent may have different capacities for representing the number 4. Similarly an agent may have different capacities for representing water (representing it *as* water versus representing it *as* $H_2$0), or the planet Venus (representing it *as* Hesperus versus representing it *as* Phosphorus), and so on. Distinct but co-referring semantically permeated types correspond to different capacities for representing the same denotation. In that sense, the types reflect different “ways of representing” the denotation.

A complete theory must elucidate the representational capacities that individuate semantically permeated types. But I am not trying to offer a complete theory. For present purposes, I simply assume that normal humans have various representational capacities. Current science amply vindicates that assumption. Cognitive science routinely type-identifies mental states through representational capacities deployed by those states. Each semantically permeated mental representation marks the exercise of a particular representational capacity.

§6. Mental computation over a semantically permeated language of thought

A computational model delineates counterfactual-supporting mechanical rules governing how a computational system transits between states. I claim that, *in some cases*, the rules describe manipulation of semantically permeated mental representations. We can type-identify mental computation through representational capacities deployed during each stage of computation. We thereby delineate a psychological model that is both intentional *and* computational. In offering such a model, we need not mention formal mental syntax. Thus, CTM does not require us to associate each mental representation with formal syntactic properties.[8]

To illustrate, consider the mathematical reasoning agent introduced in §5. She can entertain infinitely many mental numerals, generated by combining mental symbols $\mathbb{0}$, $\mathbb{S}$, and $+$. Let us stipulate that her computations conform to the following symbol transformation rules: for any numerals *v* and *w*,

Rule A: $(\mathbb{0} + w) \rightarrow w$

Rule B: $(\mathbb{S}v + w) \rightarrow \mathbb{S}(v + w)$

where the arrow signifies that one can substitute the right-hand side for the left-hand side. One can compute the sum of any two numbers by applying rules A and B. For example:

$(\mathbb{SS0} + \mathbb{SS0})$

$\mathbb{S}(\mathbb{S0} + \mathbb{SS0})$ *by rule B*

$\mathbb{SS}(\mathbb{0} + \mathbb{SS0})$ *by rule B*

$\mathbb{SSS0}$ *by rule A*

A and B are mechanical rules that dictate how to manipulate inherently meaningful mental symbols. They describe transitions among mental states type-identified through the representational capacities that those states exercise.

I stated rules A and B by inscribing geometric shapes on the page. Geometric shapes are subject to arbitrary reinterpretation, so they are semantically indeterminate. But our question concerns the mental symbols that geometric shapes represent. I am *using* geometric shapes to

[8] Fodor (1981, pp. 226-227) holds that a physical system is computational only if it has representational properties ("no computation without representation"). Chalmers demurs, and rightly so. I *do not* claim that computation requires representation. I claim that *some* computational models specify computations representationally, without any mention of formal syntactic types. In this respect, I disagree with both Chalmers and Fodor. Despite their differences, both philosophers agree that every computational model features a level of purely syntactic, non-semantic description.

mention mental symbols. Do those symbols have formal syntactic types? Nothing about rules A and B suggests an affirmative answer. Admittedly, one can articulate *analogous* rules that describe formal manipulation of geometric shapes. Those analogous rules mention geometric shapes. Nevertheless, rules A and B do not mention geometric shapes or any other semantically indeterminate types.

I propose that we take rules A and B as paradigmatic. We should describe various mental processes through mechanical rules that cite semantically permeated mental representations, without any mention of semantically indeterminate syntactic types.

Many readers will regard my proposal warily. How can semantics inform mental computation, except as mediated by formal syntax? My proposal may seem especially suspect when combined with an externalist conception of mental content. As Fodor puts it, "the effects of semantic identities and differences on mental processes must always be mediated by 'local' properties of mental representations, hence by their nonsemantic properties assuming that semantics is externalist" (1994, p. 107). Mustn't computation manipulate mental representations based solely upon their "local" properties, ignoring any relations to the external environment?

I agree that mental representations have local, non-semantic properties: namely, *neural* properties. I agree that mental computation distinguishes mental representations through their neural properties. On that basis, it manipulates the representations appropriately. Thus, I agree that semantic identities and differences inform mental processes only as mediated by local, non-semantic, neural properties. It does not follow that mental representations have *formal syntactic* properties. Syntax is multiply realizable in Putnam's (1975) sense: systems with wildly different physical properties can satisfy the same syntactic description. In particular, systems that are heterogeneous under neural description may share the same syntactic properties. Multiple realizability plays a central role in standard versions of CTM (Fodor, 1981, p. 13), (Haugeland, 1985, p. 5), (Stich, 1983, p. 151). It crucially informs Chalmers's treatment, since organizational invariant properties are multiply realizable. In contrast, neural properties are not multiply realizable. So neural properties are not syntactic properties. The undisputed fact that mental computation responds to local properties does not establish a valuable explanatory role for formal mental syntax. The relevant local properties may be neural rather than syntactic.[9]

Current science provides strong evidence that a complete theory of the mind will include at least two levels of description: *representational* description and *neural* description. Fodor insists that a complete theory will include an *additional* level that taxonomizes mental states in formal syntactic terms, without regard to neural or representational properties. Yet there are numerous mental processes that current science studies without positing formal syntactic types. For example, formal mental syntax plays no role within our best science of perception. From the perspective of current perceptual psychology, formal mental syntax is a gratuitous theoretical posit. Thus, current science provides no evidence that formal syntactic descriptions should figure in computational modeling of all mental phenomena.

I now want to elaborate the semantically permeated approach by examining case studies drawn from CS and AI. In §7, I analyze a powerful computational model of *mathematical* reasoning. In §8, I extend my analysis to *empirical* cognition. The case studies demonstrate how much one can achieve through computational models that cite representational capacities *as opposed to* formal mental syntax. In particular, the case studies show that semantically permeated computation can provide a foundation for Bayesian psychological modeling.

[9] The philosophical literature offers additional well-known arguments for postulating formal mental syntax. A complete defense of my approach would scrutinize all such arguments, a task that I defer for another occasion.

§7. The lambda calculus as a programming language

The *lambda calculus*, introduced by Church in the 1930s, embodies an elegant model of symbolic computation that informs many modern functional programming languages. I will discuss one notable example: the language PCF (for *programming computable functions*), introduced by Scott in a privately circulated 1969 manuscript eventually published as (Scott, 1993). PCF can represent all Turing-computable numerical functions, not to mention diverse higher-type functions (e.g. functions over numerical functions). The language is useful primarily for theoretical analysis, rather than practical programming purposes. My presentation of PCF is informal and incomplete. See (Mitchell, 1996) for full details.[10]

PCF contains primitive symbols, including $+$, 0, 1, 2, 3, … and devices for generating complex expressions. One notable device is *lambda abstraction*. For example, $\lambda x\colon nat.\ x + x$ denotes the function that maps each natural number *n* to 2*n*. Another notable device is *functional application*. For example, $(\lambda x\colon nat.\ x + x)\ 2$ denotes the result of applying the doubling function to 2. Denotations of PCF expressions are determined compositionally:

> If expression *M* contains no free variables except x, then the expression $\lambda x\colon \sigma.$ *M* denotes the function over domain σ that carries each $d \in \sigma$ to whatever *M* denotes once we assign value *d* to all free occurrences of x.
>
> If expression *M* denotes a function, then expression *M N* denotes the result of applying that function to the input denoted by expression *N*.

[10] Technically, PCF extends the pure simply-typed lambda calculus with primitive numerals, primitive Boolean terms, and fixed-point operators at each type.

One can convert these informal clauses into a rigorous *denotational semantics*. One can use the semantics to prove various intuitive facts, such as that $(\lambda x{:}\ nat.\ x + x)\ 2$ denotes 4.[11]

PCF comes equipped with an *operational semantics*, due to (Plotkin, 1977). As Fodor (1981, pp. 204-224) notes, "operational semantics" is not a semantics in the most familiar philosophical sense, because it does not address reference, truth-conditions, or the like. It simply offers mechanical rules governing symbol transformation. For example, the β rule is

$$(\lambda x{:}\ \sigma.\ M)\ N \ \rightarrow\ [N\,/\,x]\ M$$

where the expression on the right is the result of substituting N for all free occurrences of x within M. There are additional transformation rules governing other PCF locutions. The transformation rules generate mathematical computations, such as the following:

$(\lambda x{:}\ nat.\ x + x)\ 2$	
$2 + 2$	*by the β rule*
4	*by the transformation rules for $+$*

I write $M \rightarrow^* N$ when we can transform M into N through iterated application of transformation rules. Thus, $(\lambda x{:}\ nat.\ x + x)\ 2 \rightarrow^* 4$. By specifying a canonical order for applying transformation rules, we delineate a deterministic symbol manipulation model (Mitchell, 1996, pp. 84-96).[12]

PCF's denotational semantics is no mere rhetorical appendage to its operational semantics. As Scott emphasizes (1993, p. 413), denotational semantics is what elevates PCF from a formal calculus to a model of mathematical reasoning. Denotational semantics provides a

[11] Several complexities arise in providing a rigorous denotational semantics. First, one must employ familiar Tarskian techniques so as to handle free variables. Second, the language has a type structure, which one must treat more gingerly than I do in my informal exposition. Third, and most seriously, the language expresses recursive definitions through primitive fixed point operators at each type. The semantics for fixed point operators requires serious machinery too complicated to discuss here (Mitchell, 1996, pp. 305-333). Taking such complexities into account would muddy the exposition without affecting my overall argument.

[12] *Almost* deterministic. The technical definition of "substitution" engenders a subtle element of indeterminacy (Mitchell, pp. 53-54).

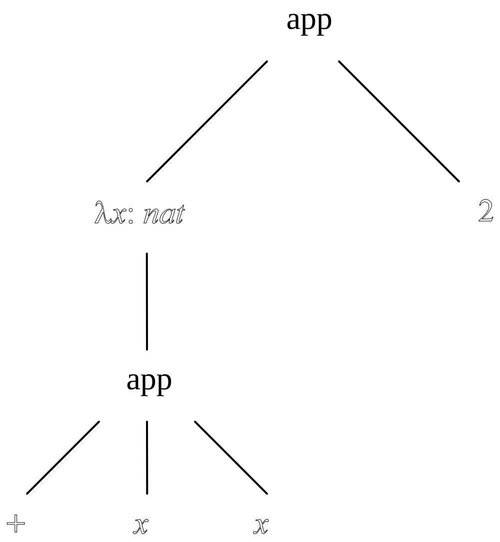
standard for evaluating whether the operational semantics is satisfactory. Transformation rules must honor the intuitive meanings of PCF expressions, as codified by the denotational semantics. Two theorems reflect this desideratum (Gunter, 1992, pp. 133-137):

Soundness: If $M \rightarrow^* n$, then M denotes n.

Computational adequacy: If M denotes n, then $M \rightarrow^* n$.

Failure of soundness would entail that our operational semantics yields incorrect results. Failure of computational adequacy would entail that our transformation rules do not generate all the computations we want them to generate.

I have represented PCF expressions through strings of shapes. But PCF expressions are *not* strings of shapes. Strings of shapes enshrine extraneous notational detail (Gunter, 1992, pp. 32-34), (Mitchell, 2006, pp. 21-26). For example, we can represent the same expression using prefix, infix, or postfix notation. To minimize extraneous notational detail, computer scientists often employ *parse tree diagrams*. Adapting Gunter's (1992, p. 33) notation, we replace the string "$(\lambda x\colon nat.\ x + x)\ 2$" with the tree diagram



Yet even this tree diagram contains extraneous detail. It uses arbitrary shapes, which we could vary while representing the same underlying PCF expression. The diagram reads from left to right, but we could instead use a diagram that reads from right to left. Analogous problems arise if we replace tree diagrams with *set-theoretic trees*, i.e. partially ordered sets. There are various set-theoretic techniques for encoding the ordered pairs that compose a partial ordering. We can change our set-theoretic representation without changing the underlying expression.

So what *is* the underlying expression? In my opinion, CS does not answer this question. CS offers a *formalism* subject to conflicting analyses. Chalmers's theory suggests one analysis:

> A PCF expression is individuated by its role in symbol transformation. The operational semantics for PCF dictates a unique causal topology. A PCF expression is an item that plays the right role in this causal topology. The causal topology does not fix a unique denotational semantics.

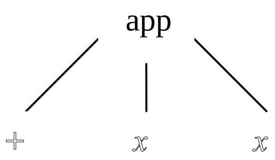
On this analysis, PCF expressions are semantically indeterminate. They are formal syntactic items subject to reinterpretation. The operational semantics describes formal syntactic manipulation, without regard to any meanings syntactic items may have.

Alternatively, we can construe PCF in semantically permeated fashion. We can construe it as modeling how an idealized cognitive agent transits among representational mental states:

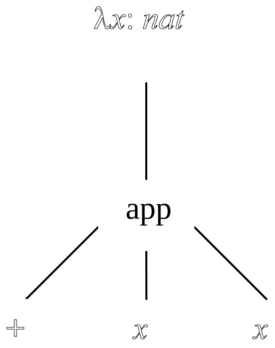
> The agent has capacities to represent certain mathematical entities. Each PCF expression marks the exercise of a representational capacity. Primitive expressions correspond to capacities that we take as primitive (for present purposes). Complex expressions correspond to complex capacities that decompose into the exercise of simpler capacities. The operational semantics delineates mechanical rules governing the manipulation of these semantically permeated types.

To illustrate, consider the tree diagram inscribed above:

- The agent has a capacity to apply functions to arguments. Each “app” node registers an exercise of this capacity.
- x corresponds to a capacity to represent objects drawn from the overall domain, including natural numbers, numerical functions, and so on.
- $+$ corresponds to a capacity to represent the addition function.
- The foregoing three capacities jointly yield a complex capacity to add any natural number to itself. More precisely, they yield a capacity to convert any numerical input *n* into a numerical output 2*n*. The subexpression



  corresponds to this complex capacity.
- λx: *nat* corresponds to a capacity *C* that satisfies the following constraint: given a capacity *C′* to convert numerical inputs into numerical outputs, *C* and *C′* jointly yield a capacity to represent the corresponding numerical function.
- The foregoing five capacities jointly yield a complex capacity to represent the doubling function. The subexpression



corresponds to this complex capacity.

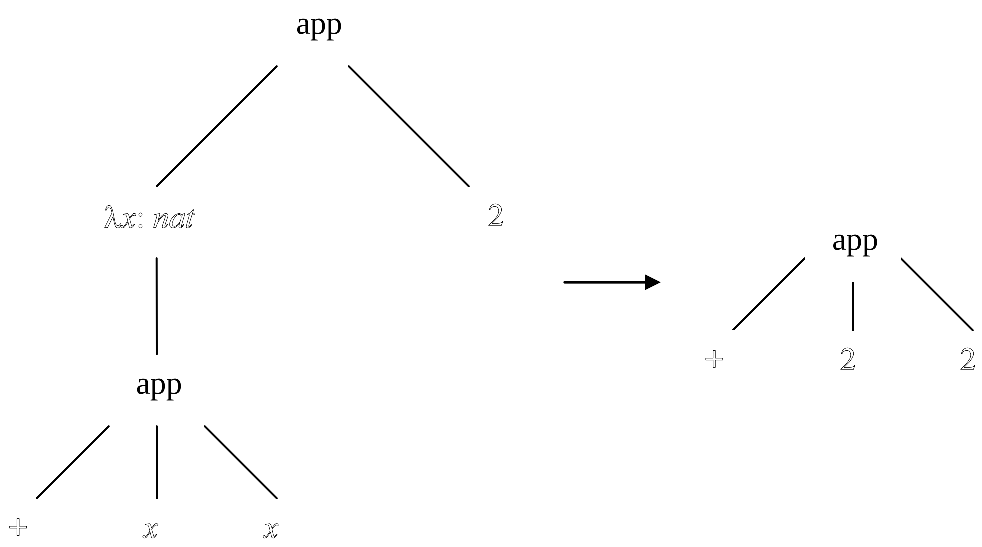
- $\mathbb{2}$ corresponds to a capacity to represent the number 2. We might decompose this capacity into more primitive capacities (e.g. capacities to represent 0 and the successor operation). For present purposes, we take it as primitive.
- The foregoing seven capacities jointly yield a complex capacity to apply the doubling function to 2. The main PCF expression corresponds to this complex capacity.

PCF expressions mark the exercise of representational capacities, so they determine specific representational contents. For example, $\mathbb{2}$ necessarily denotes 2, while $\lambda x\!: nat.\ x + x$ necessarily denotes the doubling function.

Say that a function is *PCF-definable* just in case some PCF expression denotes it under the standard denotational semantics. (One can prove that a numerical function is PCF-definable just in case it is Turing-computable.) We can characterize PCF-definability in the same way whether we employ semantically indeterminate or permeated taxonomization. On either approach, the same functions are PCF-definable. Thus, the notion of PCF-definability does not favor semantic indeterminacy over semantic permeation.[13]

PCF's operational semantics is similarly indifferent between semantically indeterminate and permeated taxonomization. We can construe the β rule as describing how to manipulate formal syntactic items. Alternatively, we can construe it as describing how to manipulate inherently meaningful mental representations. Under the latter construal, the rule dictates how to transit between mental states individuated by the representational capacities those states deploy. To illustrate, consider the following instance of β:

[13] If we countenance PCF-definability relative to sufficiently *deviant* semantic interpretations, then intuitively uncomputable functions become PCF-definable (Rescorla, 2007). However, the possibility of deviant semantic interpretations does not militate against semantic permeation. It merely demonstrates that one should not individuate semantically permeated types through a deviant denotational semantics, any more than one should interpret semantically indeterminate types through a deviant denotational semantics.



A description along these lines is too schematic to favor semantic indeterminacy or permeation. To choose between the two construals, one must say how one interprets tree diagrams. Of course, tree diagrams themselves are semantically indeterminate. But our question is how one should individuate the linguistic items that tree diagrams represent.

Computer scientists usually employ a semantically indeterminate individuative scheme for PCF expressions. They study diverse mathematical structures through which one can interpret PCF (Longley, 2005), (Mitchell, 1996, pp. 355-385, pp. 445-505). The denotational semantics sketched above furnishes one such mathematical structure: the *standard* model. This model contains numbers, numerical functions, functions of higher type, and so on. Alternative models contain quite different entities, such as game strategies or codes for Turing machines. There are notable mathematical results concerning these alternative models. Thus, the semantically indeterminate taxonomic scheme has proved mathematically fruitful.

Nevertheless, semantic indeterminacy is not obligatory. PCF's operational semantics does not mandate that we individuate PCF expressions entirely through the operational semantics. The operational semantics is consistent with a taxonomic scheme that takes a specific denotational semantics into account. We can employ this taxonomic scheme when modeling the mathematical cognition of an idealized agent. The agent's computations deploy *standing capacities to represent specific mathematical entities*. Alternative interpretations are irrelevant, because we are studying mental activity that deploys fixed, standing representational capacities. Those standing capacities determine a unique denotational semantics, which we cite when individuating the agent's mental computations. Using PCF's operational semantics, we specify mechanical transformation rules over semantically permeated types. We thereby delineate transitions among mental states *type-identified through their representational import*. In describing those transitions, we will undoubtedly *use* semantically indeterminate inscriptions. But we need not attribute semantically indeterminate syntactic types *to the agent's mental states*.

My position here goes far beyond the weak claim that we can describe PCF computation in representational terms. Chalmers and Fodor would presumably agree with that weak claim. My position is that nothing about PCF's operational semantics mandates an explanatory significant role for formal mental syntax. Mechanical transition rules governing mental computation can appeal instead to representational capacities deployed by mental states.

§8. Computational foundations for Bayesian inference

The previous section focused upon computations that represent the *mathematical* realm. How does the semantically permeated approach apply to computations that represent the *empirical* realm? In particular, how does it apply to Bayesian mental inference? To address these questions, I will examine some computational models drawn from AI. Over the past two decades, AI researchers have intensively studied how to implement Bayesian inference in a high-level programming language. This research illuminates how a possible mind (if not *actual* human minds) can implement Bayesian inferences resembling those postulated within cognitive science.

As many researchers have emphasized, Bayesian inference is often hopelessly inefficient. If the probability domain is continuous, then computing the constant η in

$$p(h \mid e) = \eta\, p(h)\, p(e \mid h)$$

requires integrating over the domain, a task which may be computationally intractable. Even when the domain is finite but extremely large, computing η may be computationally intractable. To construct computationally tractable models, we must resign ourselves to *approximating* precise Bayesian inference. AI offers several approximation strategies.

Example: Importance sampling

A *probability function* is a function from a *probability domain* to the real numbers.[14] The *Monte Carlo method* approximates a probability function through samples drawn from the probability domain. The samples serve as a proxy for the function. We compute over the samples, thereby approximating computation over the probability function. The more samples we generate, the more accurate our approximation.

Using the Monte Carlo method, we can approximate conditionalization through an *importance sampling* algorithm (Park, Pfenning, and Thrun, 2008):

- *Sample*: Draw n samples $h_1, \ldots, h_n$ from the probability domain. The probability of drawing h_i is proportional to $p(h_i)$. One may draw a sample multiple times.
- *Weight*: Assign weight w_i to h_i, in proportion to the prior likelihood:

$$w_i \propto p(e \mid h_i).$$

[14] By using the deliberately vague terms "probability function" and "probability domain," I conflate *probability measures*, *probability distributions*, and *probability density functions*. These distinctions are crucial for many purposes (Rescorla, forthcoming b), but we can safely ignore them here.

We now have a list of n weighted samples: $(h_1, w_1), \ldots, (h_n, w_n)$.

- *Resample*: Draw n samples $h_{m_1}, h_{m_2}, \ldots, h_{m_n}$ from the foregoing list. The probability of drawing a given hypothesis h_i is determined by the weights. Once again, we may draw a sample multiple times.

One can show that the probability of h_i appearing on the final list is approximately proportional to the posterior $p(h_i \mid e)$. Thus, the list is a good proxy for the posterior. □

The importance sampling algorithm presupposes a capacity to sample from a domain in accord with a prior probability over that domain. The algorithm also presupposes a capacity to assign weights in accord with a prior likelihood, along with a capacity to resample according to the assigned weights. There is no obvious respect in which these capacities require formal syntactic manipulation. There is no obvious respect in which the algorithm mandates that one manipulate formal syntactic types. The algorithm *can* operate over formal syntactic types, but it can operate just as well over semantically permeated mental representations.

A similar moral prevails when we implement importance sampling through a high-level programming language. I illustrate by discussing a particularly instructive case study.

Example: A probabilistic extension of the lambda calculus

The language λ_o, introduced by Park, Pfenning, and Thrun (2008), augments the lambda calculus with resources for representing and manipulating probability functions. To represent a probability function, λ_o specifies a procedure for drawing samples from the probability domain. λ_o specifies sampling procedures through *sampling expressions*. A sampling expression E consumes a string of random real numbers and outputs an element of the probability domain. The output depends on the random real numbers consumed. Thus, λ_o computation is stochastic. The expression prob *E* denotes the probability function induced by sampling expression *E*:

> If *E* is a sampling expression and *Prob* is a probability function, then "[w]e write *E* ~ *Prob* if outcomes of computing *E* are distributed according to *Prob*… If *E* ~ *Prob*, then *E* denotes a probabilistic computation for generating samples from *Prob*, and we regard *Prob* as the denotation of prob *E*" (Park, Pfenning, and Thrun, 2008, p. 19).[15]

For example, if outcomes of evaluating *E* are distributed according to a Gaussian distribution, then prob *E* denotes a Gaussian distribution. In this manner, λ_o can represent virtually any probability function one might reasonably require.

The operational semantics for λ_o resembles that for PCF. The main difference is that λ_o features new transformation rules governing new vocabulary. In particular, λ_o features an importance sampling rule that approximates Bayesian updating. To illustrate, consider a prior probability $p(h)$ and a prior likelihood $p(e \mid h)$. We represent $p(h)$ through a λ_o expression M_{prior}. We capture $p(e \mid h)$ through a λ_o expression $M_{likelihood}$ that represents a function from inputs e to probability functions over h. The operational semantics engenders a stochastic computation that transforms M_{prior}, $M_{likelihood}$, and input e into an expression:

(*) prob importance $\{(h_i, w_i) \mid 1 \leq i \leq n\}$

where h_i and w_i are computed as in the importance sampling algorithm. importance $\{(h_i, w_i) \mid 1 \leq i \leq n\}$ is a sampling expression. It instructs us to draw samples h_i in accord with weights w_i. So (*) denotes a probability function that approximates the desired posterior $p(h \mid e)$. □

[15] I have reformatted the quoted passage to ensure conformity with my formatting conventions. One can easily convert this informal clause into a rigorous denotational semantics for the substantial fragment of λ_o that excludes fixed point operators and higher-order type expressions (Park, Pfenning, and Thrun, 2008, pp. 21-23). A formal semantics for the full language should be possible in principle, but many technical details require investigation. For example, there may exist pathological expressions that do not determine probability distributions.

λ_o is a purely mathematical language. Thus, it cannot represent a probability function over an empirical domain. It can only represent a probability function over a mathematical domain, such as the real numbers. In itself, then, λ_o cannot provide a foundation for the Bayesian models postulated by cognitive science. Those models postulate probabilistic updating over hypotheses that represent environmental properties: shapes, reflectances, and so on. To represent the desired probability functions, we must supplement λ_o with resources for representing desired environmental properties. For example, we can supplement λ_o with symbols $\mathbb{R}_1$, $\mathbb{R}_2$, …, $\mathbb{R}_i$, …, where each $\mathbb{R}_i$ represents a specific surface reflectance R_i.

Once we supplement λ_o with suitable empirical vocabulary, we can define probability functions over the desired empirical domain. The most straightforward strategy is to assume a fixed mapping from a suitable mathematical domain (e.g. *n*-tuples of real numbers) to the desired empirical domain (e.g. reflectances). Using this mapping, we can convert any sampling procedure defined over the mathematical domain into a sampling procedure defined over the empirical domain. More specifically, suppose that expression E represents a procedure for sampling elements from the mathematical domain, and suppose that expression F represents a fixed mapping from the mathematical domain to the empirical domain. By applying function composition, we can construct an expression E_F that represents a procedure for sampling elements from the empirical domain. In this manner, we can define diverse expressions E_F that denote sampling procedures over the empirical domain. So we can define diverse expressions prob E_F that denote probability functions over that domain. We can then approximate our desired Bayesian model through symbolic transformations governed by λ_o's operational semantics.

The operational semantics is compatible with either semantically indeterminate or semantically permeated taxonomization. λ_o merely supplements PCF-style transformation rules with a few additional rules governing new vocabulary, including a rule that formalizes importance sampling. As with PCF, we can construe the transformation rules as describing how to manipulate formal syntactic items, or we can construe them as describing how to manipulate inherently meaningful mental representations. Under the latter construal, the rules govern transitions among mental states type-identified by the representational capacities those states deploy. Admittedly, λ_o's creators seem to have in mind a semantically indeterminate rather than semantically permeated taxonomic scheme. Nevertheless, both taxonomic schemes are equally consistent with the operational semantics.

To illustrate, imagine a hypothetical Bayesian agent who executes λ_o computation. She has capacities to represent various environmental properties: shapes, reflectances, etc. We type-identify her mental states by citing these and other representational capacities. Semantically permeated mental representations reify the resulting types:

- We postulate semantically permeated representations $\mathbb{R}_1$, $\mathbb{R}_2$, …, $\mathbb{R}_i$, …, where each $\mathbb{R}_i$ corresponds to a capacity to represent a specific reflectance R_i. A token mental state has type $\mathbb{R}_i$ only if it deploys that capacity, to do which it must represent R_i. All possible tokens of $\mathbb{R}_i$ denote R_i. Likewise, we postulate semantically permeated representations that necessarily represent specific shapes, sizes, and so on.
- Given a capacity to represent elements of domain D, the agent can deploy λ_o's mathematical resources to represent procedures for sampling from D. Each sampling expression E corresponds to a capacity to sample in a certain way from a domain D.
- $\mathbb{prob}$ corresponds to a capacity C that satisfies the following constraint: if C' is a capacity for sampling from domain D, and if exercises of C' yield outcomes

distributed according to probability function *Prob* over *D*, then *C* and *C′* jointly yield a capacity to represent *Prob*.

- The foregoing capacities jointly yield a complex capacity to represent a probability function *Prob*, where outcomes of evaluating *E* are distributed according to *Prob*. The expression $\mathbb{prob}$ *E* corresponds to this complex capacity.

Thus, $\mathbb{prob}$ necessarily satisfies the following compositional clause:

> If there exists a probability function *Prob* such that outcomes of evaluating *E* are distributed according to *Prob*, then $\mathbb{prob}$ *E* denotes *Prob*.

The agent computes over semantically permeated mental representations. She thereby approximately implements Bayesian inference.

More specifically, consider a Bayesian model of surface reflectance estimation. The model postulates prior probabilities over reflectances and illuminants. It postulates a prior likelihood relating reflectances, illuminants, and retinal inputs. It describes the perceptual system as converting these priors and retinal input into a posterior over reflectances. We do not yet know how exactly the human perceptual system executes (or approximately executes) this Bayesian inference. But we can delineate a computational model describing *one possible way that a hypothetical perceptual system* might approximately execute the desired inference:

(1) Postulate semantically permeated representations $\mathbb{R}_1$, $\mathbb{R}_2$, …, $\mathbb{R}_i$, … and $\mathbb{I}_1$, $\mathbb{I}_2$, …, $\mathbb{I}_i$, … Each $\mathbb{R}_i$ corresponds to a capacity to represent a reflectance R_i. Each $\mathbb{I}_i$ corresponds to a capacity to represent an illuminant I_i.

(2) Using λ_o's representational resources, supplemented as in (1), construct diverse semantically permeated sampling expressions *E*. Each expression corresponds to a capacity to sample from the set of reflectances or the set of illuminants.

(3) Construct semantically permeated expressions prob E. Each expression corresponds to a capacity to represent a probability function over reflectances or over illuminants.

(4) Choose specific semantically permeated expressions $M_{reflectance\ prior}$, $M_{illuminant\ prior}$, and $M_{likelihood}$ denoting the reflectance prior, the illuminant prior, and the prior likelihood.

(5) Using λ_o's operational semantics, delineate symbol transformation rules governing perceptual inference. The resulting computations respond to retinal input by transforming $M_{reflectance\ prior}$, $M_{illuminant\ prior}$, and $M_{likelihood}$ into a new expression prob importance $\{(R_i, w_i) \mid 1 \leq i \leq n\}$. The new expression denotes a probability function *Prob* over reflectances, where *Prob* approximates the desired posterior probability.

A model along these lines need not assign any role to formal mental syntax. The transformation rules do not associate mental states with formal syntactic types. Instead, they individuate mental states through the representational capacities that those states exercise.

I have sketched how a hypothetical computational system might approximately implement Bayesian inference. To what extent do the resulting computations resemble *actual human mental activity*? This question is difficult to answer, because we know very little about the computations through which humans approximately implement Bayesian updating. However, several psychologists argue that Monte Carlo algorithms play a central role in human Bayesian inference, including perceptual inference (Gershman, Vul, and Tenenbaum, 2012). A few psychologists suggest that the lambda calculus may provide a scaffolding for the language of thought (Piantadosi, Tenenbaum, and Goodman, forthcoming). So the proposed computational model finds some grounding in current science. On the other hand, normal human minds probably cannot harness the full mathematical power of λ_o, which represents a vast range of higher-type functions. In that respect, the computational model seems unrealistic.

I do not present the model as an empirical hypothesis. I offer it as an existence proof: there exist semantically permeated computational models that tractably approximate Bayesian inference. In principle, then, semantically permeated computation can provide a foundation for Bayesian psychological modeling. Future scientific progress will reveal whether semantically permeated computation provides an *empirically well-confirmed* foundation.

§9. Representational capacities versus causal topologies

Philosophers usually take computational modeling to embody an *internalist* template that ignores matters outside the subject's skin (Fodor, 1981). I have presented an alternative *externalist* version of CTM. On my approach, computational models can individuate mental states through representational properties that do not supervene upon internal neurophysiology. Naturally, there is an important level of description that type-identifies neural duplicates: namely, *neural* description. A complete cognitive science will unveil the neural mechanisms that implement semantically permeated mental computation. The question is whether we require computational descriptions that prescind from neural *and* representational details. Should we posit formal syntactic types or any other semantically indeterminate, multiply realizable items? My externalist version of CTM requires no such items.

You may ask: how can a system "know" whether the symbols it manipulates have appropriate representational properties? Am I presupposing an inner homunculus who inspects a mental symbol's meaning before deciding how to manipulate the symbol?

I reply that a system can conform to representationally-specified rules even if the system does not interpret the symbols it manipulates. My approach does *not* require that the mind examine its own representational relations to the environment. I do *not* posit an inner homunculus who inspects a mental symbol's meaning. For example, §8's model of color perception does not require the perceptual system to evaluate whether $\mathbb{R}_i$ represents reflectance R_i or R_j. The model simply postulates certain representational capacities (e.g. standing capacities to represent specific reflectances), reliably deployed in response to retinal input. The model articulates mechanical rules governing deployment of those capacities. The rules operate over semantically permeated types, which mark the exercise of representational capacities. Conformity to the rules does not require the system to interpret its own mental representations.

Given current scientific knowledge, a semantically permeated version of CTM is speculative. But it is no more speculative than Chalmers's semantically indeterminate account. Furthermore, it offers a crucial advantage: it preserves the representational explanatory paradigm widely employed within current science. Chalmers replaces that paradigm with an organizationally invariant alternative that finds no echo within Bayesian psychology. I have shown that we can model the mind computationally while avoiding Chalmers's radical revisionism. We can integrate intentionality directly into computational models of mental activity, without articulating an organizationally invariant level of description.

Chalmers might retort that semantically permeated models are not genuinely *mechanical*. But I see no reason to agree. The semantically permeated symbol transformation rules isolated above seem as precise, routine, and mechanical as one could desire. Any argument to the contrary requires an *independent, non-question-begging* criterion of the "mechanical." I find no such criterion in Chalmers's discussion. Certainly, no such criterion emerges from the mathematical study of computation (including CS and AI). On the contrary, numerous mathematical models of computation are perfectly congenial to semantically permeated computation. I submit that mechanical rules can individuate computational states in representational fashion. Genuinely mechanistic explanations can proceed at the representational rather than the organizationally invariant level.

Chalmers and I agree that minds have causal topologies. We agree that minds have representational capacities. Our disagreement concerns whether causal topologies or representational capacities will prove central to a developed science of the mind. I contend that many mental phenomena are best studied by citing representational capacities rather than causal topologies. In contrast, Chalmers elevates causal topologies at the expense of representational capacities. Future scientific developments will settle which paradigm is more fruitful.

Works Cited

Bays, P., and Wolpert, D. 2007. "Computational Principles of Sensorimotor Control that Minimize Uncertainty and Variability." *Journal of Physiology* 578: 387-396.

Block, N. 2003. "Mental Paint." In *Reflections and Replies: Essays on the Philosophy of Tyler Burge*, eds. M. Hahn. and B. Ramberg. Cambridge: MIT Press.

Brainard, D. 2009. "Bayesian Approaches to Color Vision." In *The Visual Neurosciences*, 4th ed., ed. M. Gazzaniga. Cambridge: MIT Press.

Burge, T. 2007. *Foundations of Mind*. Oxford: Oxford University Press.

---. 2009. "Five Theses on *De Re* States and Attitudes." In *The Philosophy of David Kaplan*, ed. J. Almog and P. Leonardi. Oxford: Oxford University Press.

---. 2010a. *Origins of Objectivity*. Oxford: Oxford University Press.

---. 2010b. "Origins of Perception." *Disputatio* 4: 4-38.

Chalmers, D. 2006. "Perception and the Fall from Eden." In *Perceptual Experience*, eds. T. Gendler and J. Hawthorne. Oxford: Oxford University Press.

---. 2012. "A Computational Foundation for the Study of Cognition." This volume.

Chater, N., and Manning, C. 2006. "Probabilistic Models of Language Processing and Acquisition." *Trends in Cognitive Science* 10: 335-344.

Chater, N., and Oaksford, C. 2008. *The Probabilistic Mind*. Oxford: Oxford University Press.

Cheng, K., Shettleworth, S., Huttenlocher, J., and Rieser, J. 2007. "Bayesian Integration of Spatial Information." *Psychological Bulletin* 133: 625-637.

Egan, F. 2003. 1999. "In Defense of Narrow Mindedness." *Mind and Language* 14: 177-194.

Evans, G. 1983. *The Varieties of Reference*, ed. J. McDowell. Oxford: Oxford University Press.

Fodor, J. 1981. *Representations*. Cambridge: MIT Press.

---. 1991. "Replies." In *Meaning in Mind*, eds. B. Loewer and G. Rey. Cambridge: Blackwell.

---. 1994. *The Elm and the Expert*. Cambridge: MIT Press.

Gershman, S., Vul, E., and Tenenbaum, J. 2012. "Multistability and Perceptual Inference." *Neural Computation* 24: 1-24.

Gunter, C. 1992. *Semantics of Programming Languages*. Cambridge: MIT Press.

Haugeland, J. 1985. *Artificial Intelligence: The Very Idea*. Cambridge: MIT Press.

Knill, D. and Pouget, A. 2004. "The Bayesian Brain: The Role of Uncertainty in Neural Coding and Computation." *Trends in Neuroscience* 27: 712-719.

Knill, D., and Richards, W. 1996. *Perception as Bayesian Inference*. Cambridge: Cambridge University Press.

Lewis, D. 1972. "Psychophysical and Theoretical Identifications." *Australasian Journal of Philosophy* 50: 249-58.

Longley, J. 2005. "Notions of Computability at Higher Types I." in *Logic Colloquium 2000*, eds. R. Cori, et al. Wellesley: A. K. Peters.

Mitchell, J. 1996. *Foundations for Programming Languages*. Cambridge: MIT Press.

Park, S., Pfenning, F., and Thrun, S. 2008. "A Probabilistic Language Based upon Sampling Functions." *ACM Transactions on Programming Languages and Systems* 5: 1-45.

Peacocke, C. 1992. *A Study of Concepts*. Cambridge: MIT Press.

---. 1994. "Content, Computation, and Externalism." *Mind and Language* 9: 303-335.

Piantadosi, S., Tenenbaum, J., and Goodman, N. Forthcoming. "Bootstrapping in a Language of Thought: A Formal Model of Numerical Concept Learning." *Cognition*.

Piccinini, G. 2008. "Computation without Representation." *Philosophical Studies* 137: 205-241.

Plotkin, G. 1977. "LCF Considered as a Programming Language." *Theoretical Computer Science* 5: 223-255.

Putnam, H. 1975. *Mind, Language, and Reality: Philosophical Papers, vol. 2*. Cambridge University Press.

Quine, W. V. 1960. *Word and Object*. Cambridge, MA: MIT Press.

Rescorla, M. 2007. "Church's Thesis and the Conceptual Analysis of Computability." *Notre Dame Journal of Formal Logic* 48: 253-280.

---. Forthcoming a. "Against Structuralist Theories of Computational Implementation." *British Journal for the Philosophy of Science*.

---. Forthcoming b. "Bayesian Perceptual Psychology." In *The Oxford Handbook of the Philosophy of Perception*, ed. M. Matthen. Oxford: Oxford University Press.

Scott, D. 1993. "A Type-Theoretical Alternative to ISWIM, CUCH, and OWHY." *Theoretical Computer Science* 121: 411-440.

Shepherdson, J. and Sturgis, H. E. 1961. "Computability of Recursive Functions." *Journal of the Association of Computing Machinery* 10: 217-255.

Stich, S. 1983. *From Folk Psychology to Cognitive Science*. Cambridge, MA: MIT Press.

Stone, J. 2011. "Footprints Sticking Out of the Sand, Part 2: Children's Bayesian Priors for Shape and Lighting Direction." *Perception* 40: 175-190.