From "Naturalizing Phenomenology" to Formalizing Cognitive Linguistics (III): Externalism and the Interface Theory of Meaning

Nathaniel Christen

2021/05/23

Abstract

This essay continues the themes of Part I and II, but continues and expands on Part II's emphasis on methodology and "Philosophy of Linguistics". I also address some general philosophical themes and debates which are relevant to issues in language and communicative processes, such as externalism/internalism and John Searle's "Chinese Room" argument.

> On connaît la célèbre affirmation de Claude Lévi-Strauss: "les sciences humaines seront structurales ou ne seront pas". Nous aimerions lui en adjoindre une autre: "les sciences humaines seront des sciences naturelles ou ne seront pas". Evidemment, sauf à en revenir à un réductionnisme dogmatique, une telle affirmation n'est soutenable que si l'on peut suffisamment généraliser le concept classique de "naturalité", le généraliser jusqu'à pouvoir y faire droit, comme à des phénomènes naturels, aux phénomènes d'organisation structurale.
> — Jean Petitot, [61, p. 1]

> The nature of any entity, I propose, divides into three aspects or facets, which we may call its form, appearance, and substrate. In an act of consciousness, accordingly, we must distinguish three fundamentally different aspects: its form or intentional structure, its appearance or subjective "feel", and its substrate or origin. In terms of this three-facet distinction, we can define the place of consciousness in the world.
> — David Woodruff Smith, [77, p. 11]

A sign demands an individuation — a criteriology, for anyone addressed and solicited by each sign, to recognize and isolate it as such. Signs, and their referents, need an isolating, from the world around (and one another), or they are not signs. In extreme cases a sign may stand alone, like the smoke fire telling a hiker's location; but by norm, embedded in discourse and performance, signs require (and carry with them, internally) understood inter-boundaries. Both the recognition and the interpretation of signs therefore implicates cognition-logics of part/whole (mereology), and (dis/)continuity, each sign/referent disconnected in some ways, and continuous in others, with larger wholes inside which they are semi-autonomous parts.

Ad-hoc signs can blur the boundaries between conventionalized languages (verbal or not) and more impromptu social interactions — the smoke fire is in part a conventional distress signal and in part a tool, an engineered natural phenomenon designed with intended effect, like causing rescuers to see it. The distress fire is also in a sense its own referent: its function as a sign is to call attention to itself, and so its location. Or, choosing to write on one's own body — like Hamas commander Mahmoud Ishtiwi, betrayed and killed by his own movement, carving into his leg the word "zulum" ("wronged") — is expression spreading beyond the conventions of words; the signs are made to signify the conditions of their execution. These rather dramatic examples are more the provenance of semiotics than linguistics. But people in ordinary verbal communication equally rely on a mixture of linguistic and other signs — we point, make gestures, use "body language" and tone of voice. When conversation turns to some topic, like "that building

over there", such cues help speakers synchronize their attentions. Tone and gestures clarify sentiment (honest, joking, sarcastic, anger) that may be ambiguous in spoken words alone, taken "out of context". The weight of linguistic meaning is borne by semantics and pragmatics in fusion.

Semantics has both formal and informal dimensions — linking first to cognitive schema, or (as I will argue) prototypes of how schema are triggered; and second to pragmatics and contexts. Conventionalized in semantic norms, schema, part abstract and part cognitive, help prime language users to manipulate formal structures in language, relative to the situational aether. (Dis/)continuity in the plane of reference brings consciousness to a mereo-logic [27], [75] that language-cognition can then reshape into syntax and semantics [12]. Semantic layers are abstract tools, but they offer a tableau of forms and combinations which users adapt, concretely, to each context. The deep potential of language, I believe, comes from the perpetual combination of the formal/abstract and the concrete/phenomenological.

For signs, the largest whole is a "plane" of articulation; for referents, it is an overall phenomenological surround; or, for more abstract signifieds, a space of concepts. Like a footprint, whose very existence depends on both material continuity and visual break, for each sign there must be a blend of continuity and discontinuity, both around the sign and its referent. Attending to a mereologically ordered world, we need innate theories warranting criteria for seeing things as both individuals and as causally/behaviorally constrained by and from a whole. These criteria include structural consideration of the whole, and it is often in structural terms that the blend of autonomy and linkage for each part is realized. Attunement to structured organization therefore warrants the perceptual and mental isolation of particular foci of attention [95], [96].

As this plays out on planes of articulation alongside general situational awareness, the structures of discourse — its division into distinct signs and their structural interrelationships — and that of patterns we identify in our surroundings, that provide a context of discourse, play off one another. Grammar does not iconify interrelationships among referents — unlike diagrams, maps, scientific simulations, or scale models — but it ensures communicators may create structures among words that suggest, in each others' minds, concordant patterns in the environing world/situation. Even where there is direct sensory and perceptual evidence for objects' individuation and intercontinuity, visually and experientially present (which of course is only one kind of talk and reference), our preparedness to focus attention here or there depends on mental models of situations which are more abstract and schematic, and receptive to functional and interpersonal details. The objects around us are not just blobs of matter, but usually have a constructed purpose, socially sanctioned meaning, nostalgic weight, and other significance that cannot be grasped by perception alone.

A central theme in Cognitive Linguistics is that language meaning depends on situational understanding, and by extension on mental schema of spatial, temporal, and functional organization — not only how environments are arranged, but how they are causally and physically determined [63], [84]. The difference between *pour water* and *spill water*, for example, is the person's deliberate intentions in relation to natural forces and tendencies (such as that of water to fall downward). To "apply paint" to something, compared with to "cover" with paint, suggests different spatial configurations; to *fill a glass with water*, versus *pour water into* the glass, suggests both different spatial details and maybe rationale as well. These are differences in emphasis, not necessarily in actuality: those pairs of alternatives could describe identical state of affairs. But they direct conversants' attention in different ways, they choose one or another part of a scene as a reference frame, and suggest different "takes". These are driven by *semantic* variations — the choice of verbs like *pour* versus *fill*, *pour* versus *spill*, or *apply* versus *cover*. But semantic and syntactic rules work in federation, relative to context: for example, different verbs take different prepositions in different situations. Pour *into* vs. fill *with*. To join "pour" with *with* places emphasis elsewhere — onto the device which enables the pourer to do the pouring. So the grammatic and semantic norms of a language jointly offer a terrain of options from which speakers assemble combinations invoking those aspects of situations that they wish to emphasize.

In short, grammar is language's substitute for *visual* or *physical* resemblance-to-structure. Take Ronald Langacker's "landmark"/"trajector" model as an example:

one (very general) manner of spatial gestalt, subject to either intuitive, reflective analysis or to formalization ([10], [92]). "That boat crossing the lake": *boat* (trajector) perceived against *lake* (landmark), which provides context; together they produce a mental model; a figured spatial relationship. This is communicated, not by visual or kinaesthetic effect,[1] but by the more abstract effects of intentions signaled, via both exact words ("crossing" paints a different picture than would *across*, *on*, *by*; still more so, *at the bottom of*), and morphosyntactic tropes (like the form x `r` y; where "`r`" here means one from many spatial relations, taking *trajector* to the left and *landmark* to the right). *Landmark* and *trajector* are anchors around which both syntactic and semantic selections are organized.

Language needs both abstract laws and cognitively-mediated construals of ambient situations. The abstract laws are shaped by the situations, not directly — it is that extra indirection which cleaves language from other sign systems — but derivatively: language rules are optimized for conversants to mold linguistic possibilities into selection-spaces, which then become raw materials for representations of situational context. Each choice of word and form adds a piece to a representational complex, and the sum of those pieces — be this a sentence, a conversation turn, or an entire discourse — is a language act that hews to the structure of a situation, as the speaker wants to emphasize it. Here I take this perspective as pregiven: not as a perfected or homogenous theory, but as a working hypothesis on the origins of linguistic structuration as such. The scope of this paper is then to analyze its ramifications for our understanding of grammar and formal semantics.

This essay addresses topics in linguistics and the philosophy of language, though (by conventional measures of expertise) I am more of a Phenomenologist and a Computer Programmer than a linguist. I confess this not as biography, but to introduce my metatheoretical anchor points, from which derive intuitions that others might find unconventional. I am, in particular, sensitive to the experiential nuances of human cognition and skeptical that mechanical systems can emulate human minds except for narrowly defined tasks. At the same time, I think computational systems have interesting aspects that can enrich our understanding of cognition, even if we do not philosophically buy a "cognition is computation" metaphor.

To be precise, I am skeptical about "AI"; and I am also skeptical about a kind of logical reductionism that I believe exerts a definitive influence on several interrelated filds, including philosophy, linguistics, and computer science. As the paradigm seemingly goes, if we accept some form of "mind as computer" analogy, then we intrinsically accept *first* the idea that "mind" encompasses as some important part a logically articulated subsystem, which can be scientifically studied via formal logic; and *second* that as a consequence of this scientifically tractable logicality, AI is a good model or proxy for the study of mind. The unconscious deduction here seems to be that *mind as computer* has as a consequence that mind is (to some salient degree) a logical system, following a premise that computers are logical systems. But this premise is more false than it is true; so for me the whole paradigm is on shaky grounds. I will explain later why computers are not as logical as non-programmers seemingly believe. For now I'll just say this: there are rigorous accounts of computation that, I contend, are not grounded on formal logic in any technical or reductive sense. As a result, someone's non-logical-reductive views on language and consciousness do not *a priori* preclude computational models having some intuitive, explanatory, or structural-analogy place in their analyses of cognition.

My larger concerns are with the limits and ramifications of formal models that suggest cognitive patterns. I want to avoid both an eliminative metaphysics that reduces cognition to a computational metaphor — mind as computer, or perhaps a decentralized network of computer-like facilities — and also a disciplinary prejudice to reject formal models *a priori*. From a humanistic perspective, it can seem reductionistic — even dehumanizing — to ignore emotion, ethics, society, subjective experience, and personal identity, in any worldview about reason and consciousness. On the other hand, many areas of academia and (especially) industry evince a sincere faith in Artificial Intelligence simulating progressively more human-like behavior. This has led to useful technologies — better search engines, Optical Character Recognition (to automatically deposit checks, for example) — but it also offers up the question of whether real human behavior is just a totality of narrow mental

[1] At least in prose — poetry, which can bring back a semiotics of raw visual layout and auditory effect, is an exception that proves the rule.

abilities, or something more holistic. Meanwhile, technology need not *reproduce* cognition to be a window on cognitive-perceptual processes: 3D and Virtual Reality, for example, employ geometric calculations to represent shape, contour, color, texture, opacity, transparency, and lighting; the experiential realism of the resulting graphics implies retroactively that the mathematics built in to Computer Generated Imagery resonates with conscious experience — as a formal model of the world as encountered by the senses if not as a reproduction of how sense-data is intellected into consciousness.[2] At the same time, we are still aware that computer graphics are not fully real; the further experiential gap to bridge, the failure of this mathematical system to create *entirely* virtual reality, may also hold lessons about mind and consciousness [66].

Whether in vision and graphics, or in language and search engines, technology gives us solutions which are useful but imperfect. I believe therefore that we should neither reduce consciousness and cognition as "nothing but" computational activity ("implemented" in the brain somehow), but nor accept a disciplinary program where formal/computational methods are pro-forma jettisoned. Even granting that a "human" layer of meaning saturates language and thought and is beyond the realm of mechanical processing — not as a practical limitation on software (not enough memory, not enough parallelism) but something more entrenched and Ontological — this dualistic partition of the realm of (for example) language and meaning, into the functional and tractable and the emotive and expressive, can have the ironic effect of reaffirming a fairly reductionistic status quo (in the marketplace of ideas as well as services).

Cultural topics like art and literature are the most obvious examples where faith in the explanatory completeness of AI strains credulity, but they are also (with some justification) deemed tangential to the important research interests of those who do science or create products under the AI umbrella. It is more productive to apply both humanistic and formal/computational perspectives to the same phenomena; and there is a rich humanities tradition — including branches of linguistics, or Philosophy in the "Naturalizing Phenomenology" tradition (taking this term from the collection edited by Jean Petitot, Barry Smith, and David Woodruff Smith, who are also leading examples of the kind of hybrid scholarship I'm denoting) — which is sensitive to interpretive and experiential nuance but also accepts the occasional role for formal, mathematical models [60]. It is fascinating to explore "gendered" spatial experience in the context of Impressionistic painting, for example [50]; but approaching an arguably similar topic from the perspective of sentence-construction [18] allows greater comparison with mathematical grammars: at what level of formal description can issues of gender and of first-person spatial experience become causally relevant, or expressible in the terms of the theory? How should we think about computational and/or neuroscientific understandings of human reason, relative to the social, experienced world? Is consciousness (with all its personal and collective layers, its forming "a" world) an emergent property of microscale cognitive activity, by analogy to how electricity is an emergent property of electrons? Or is first-person reality fundamentally different from, if perhaps informed by, the world disclosed to us through rational cognitive-perceptual syntheses of sense-data and of situations?

The human mind is such an everpresent topic in almost any intellectual pursuit — since after all mind is the very medium of science and philosophy — that no one single cognitive, humanities, or neuroscience discipline can be realistically expected to exhaust its subject matter. So there is nothing intrinsically wrong with humanists seeing mind through its manifestation in culture and politics, or personal and collective identity, while cognitive scientists see mind through its manifestation in intelligent behaviors, functionalities, neural network architecture, and other formally analyzable organizations. The problem has emerged as the mind-as-computer motif migrates from a useful fiction in a specific academic community to a powerful cultural and industrial meme, and the scholarship best positioned to know its limitations — general humanistic and cultural studies, for example — have limited counterbalancing theoretical resources, acquiescing to a disciplinary core which in "science of

[2] Relatedly, reconstructing three-dimensional geometry, for instance from photographs, can be a valuable technology and shed light on mechanisms of human vision, without any presumption that computers have phenomenological "sight-experiences". I think this suggests how a mathematical and/or algorithmic process which takes givens in one form (like a photographic bitmap) and creates a (by some theory) equivalent representation in another form (like a 3D mesh or richer 3D scene notation), shows intelligent behavior, with useful affordances for human ends that are worth pursuing even apart from any clues to cognitive process — but that systems exhibiting the appropriate intelligent *behavior* also need not be deemed *intelligent* per se, or sentient or conscious.

mind" marginalizes, rather than engages with, those same humanities. Responding to "mind as computer" calls for extending cultural perspectives from macroscale to microscale, from art and politics to "everyday speech" and the single sentences, single speech-act, single moment of cognition — which is not necessarily discouraged in humanities scholarship but is not prioritized either, outside perhaps of relatively low-profile fields like Sociolinguistics.

Meanwhile, on the Phenomenological menu I am a committed "realist". What I mean is that, in a nutshell, we should renew our commitment not to read Husserl too psychologically; for instance, not to read *intentionality* as a psychological phenomenon. If I see a red sofa, we should go ahead and accept that what I see is a red sofa — that very object. I do not see a mental image of a red sofa or a phenomenal appearance of a red sofa or a token of red-sofa-appearance-ness. We should not be led astray by the sofa being a few feet away from me, so it is not "in my brain". Yes, my brain is over here, not over there — If I am suddenly distracted by something, look away, and forget about the sofa, my sofa-impression (but not the sofa) goes away, which seems to suggest that there my sofa-impression is not the same kind of thing as a sofa — which in turn invites us to question whether what I am really seeing is that sofa-impression, not that sofa.

But, without disputing that in *some* sense the impression is not ontologically identical to the thing itself, I still maintain that the best gloss on the situation still starts from the givenness that I do see the sofa (and not the sofa-impression or any other psychologistic posit). I will have more about to say about this realism, also. For now I'll say this: the case for "impressions" over "things themselves" seems stronger when talking about vision (which works at a distance) rather than touch — if I actually sit down on the sofa and physically contact it, we may feel more comfortable saying that my experience is directly encountering that physical object (though someone could still say that tactile sense-impressions are still not identical to objects; for one thing, the contact point between my hands/torso and the sofa — the locus of those haptic nerve cells — is not in my brain either, ergo a spatial gap still exists between brain and the sensed object). If we accept that our nervous system is in some sense a functionally organized complex, then an encounter between some external body and *part* of that system, with suitably holistic functional response, can plausibly be treated as "my brain" (or nervous system or mind) contacting the sofa — we don't need to rule out this gloss because my *central* nervous system remains physically isolated, any more than we would dispute that a knife has punctured a sealed carton when in fact only the knife-tip did so. In short, sense-causing physical contact as part of my embodied propensity to register tactile contact experientially, through the medium of functionally-organized processing that eventually includes the brain, is — I would say — a sensate manifestation of my contact with the sofa (not with a tactile-sense-impression or haptic-phenomenon of the sofa).

And if we accept this line of reasoning for touch, we should do so for vision also — partly because an intrinsic feature of *seeing* something is that we *could* with proper movement touch it, and apprehension of visual form includes anticipation of how surfaces will respond when we kinaesthetically interact with them (we might presume, for instance, that we can run our hand over the wall to the right but not to the left, if there's another wall there: this visual disclosure is also in a sense proto-kinetic).

So, before making claims about language, I have hereby asserted two main intuitive feints guiding my subsequent discussion: computer programs as useful but not logically reductive analogs for cognitive processes; and the virtues of a "realist" Phenomenology which accepts arguments to the effect that we experience "things themselves": that touching and seeing (etc.) are experiential encounters with real, external, non-psychological entities. This does not have to be a blunt realism — I don't dispute that we experience appearances in some sense — but we need to articulate the thing/appearance distinction in a way that does not disallow common-sense intuitions like "seeing the sofa" meaning that I do see some real, external sofa-thing. That would make for an analysis in pure Phenomenology if I just framed my arguments with reference to, say, Husserl's own treatment of the noemata/phenomena distinction. Here, however, I am going in a different direction and packaging a loose theory of "realism" about intended "external" objects within a treatment of "externalism" (and *internalism*) in the philosophy of language.

I am not ignoring that "external" in the sense of "wide scope" mind-world relations as a Semantics hypothesis is

only tangentially related to "external" in a phenomenological sense of experienced external objects (as opposed to experienced internal, e.g. somatic, states). But I *will* present a theory that connects these two senses of "external" (and likewise two senses of "internal").

All told, my goal here is to sketch a theory of cognitive linguistics which can resonate soundly with Phenomenology (while not being especially phenomenological on its own). This theory will be incomplete — deliberately, strategically incomplete. Indeed, every theory should be incomplete: an essential quality of modern science is our recognition that scientific explanation covers a vast breadth of scales and kinds of phenomena, and "science" as a singular human institution only exists insofar as there are many sciences, each with some measure of theoretical autonomy but also areas of overlap, so scientific explanations can bridge across scales. Biologists take it for granted that the basic intellectual structures of their disciplines can be justified by appeal to chemistry (as a causative or emergent base of biological phenomena); and the presence of *parts* of biology where this connection is explicit (like organic chemistry) is important for our overall sense of biology as something grounded in a general scientific method. But these "reductive" links are not typically operationalized in biology as a whole — a biologist is not "doing" chemistry, biological properties are not necessarily chemical properties, biological laws are not necessarily chemical laws, and biological terms are not semantically (or even arguably referentially) reducible to chemical terms.

We can consider whether biological concepts are "in some sense" reducible to (or extensionally equivalent to or "the same stuff as") chemical concepts, but framing this discussion as a nuanced debate implies that biology is not *trivially* reducible to chemistry, and we may accept such a reduction as a plausible option only insofar as some of us may hold philosophical commitments, which "we" collectively do not want to dismiss out of hand, that higher-scale sciences are necessarily reducible to lower-scale ones that are their causal or physical-constitutive base. But even if there is a sense of "reduction" and of "biology" and "chemistry" that makes biology reducible to chemistry, this dos not make biological *science* reducible to chemical *science* — that is, a well-constructed and discursively evaluable biological theory should not be expected to consider in any detail its own reductive interpretations, or express its concepts in chemical (rather than biological) terms, or attempt to *explain* rather than just *presuppose* chemical laws (preservation of quantities in chemical reactions, acid/base qualities, solvents and solubility, molecular interactions, etc.). Ditto for chemistry in relation to molecular physics, molecular physics in relation to quantum physics, neurology in relation to biology, and so forth. In short, whatever our philosophical intuitions about emergent phenomena and the ontological duality (or monism) between emergent and base scales, these philosophical points are only tangentially related to the equally important philosophical question about what makes a good theory in a science.

This bears reiterating: when considering a science (I'll include social sciences and humanities here) philosophically, there are two different sorts of questions that can arise. On the one hand, what is the ontological status of the entities, laws, and quantitative models postulated by the science and its currently influential theories? Should we understand terms to be proposed natural kinds (like "protons"), structural features that don't necessarily align with straightforward patterns of reference (like "dark matter"), referring expressions into complex systems whose parts have somehow fuzzy or underdetermined boundaries or criteria of individuation (like "storms" in the context of climate science), or quasi-references which have the form of concrete designations but are really just shorthand for elaborate paradigms (like "natural selection")? These are various options in the semantics of scientific jargon, which are clues to the proper ontological status of sciences' theoretical posits (this much applies to linguistics also, with its theoretical vocabulary of concepts, lexemes, syntax rules, generative semantic rules, and so forth — are these mental subsystems? Innate cognitive faculties? Clusters of nerve cells? Neural pathways reinforced during language acquisition?). But, on the other hand, there is a different order of question philosophers can ask with regard to a particular science: what qualifies as a well-constructed theory for that science? What sorts of formal models hold explanatory merit as, seemingly, capturing the causative factors determining the behavior of the systems that science investigates: continuum-based numerical models? Models in discrete mathematics? Systems of logic? State machines? And interconnected with that question is the proper scope of the science: a well-constructed theory needs to honor boundaries between and autonomy of different sciences. Having a clear picture of what beliefs in *other* sciences

to take as explanatory primitives in *this* science is an essential criteria of theoretical soundness — no less than the urge to pursue explanatory closure within the proper bounds of each science.

One of my objections to "logical reductive" paradigms in linguistics (and computer science) is their failure to distinguish these two aspects of a philosophy of science, by my lights. When discussing chemistry or biology, we can make a clear distinction between metaphysical commitments according to which higher-scale systems reduce (via physical composition and the propagation of causality across levels of organization) to lower ones — biology to chemistry to physics — as a genre of reduction obviously different from reducing sciences as collective intellectual exercises. We do not reduce the community of biologists to the community of chemists, or the kinds of expertise and fluency in certain mental gymnastics, or the criteria of what makes good biological theories, to the concordant community, conceptualizations, gymnastics, and theory-criteria of chemists. This is for me part of what makes biology a successful science — *it is incomplete in an ontologically necessary, intellectual fertile way.* But if this is a reasonable criterion, what can we say about linguistics as a science? Is it incomplete in an ontologically necessary and intellectually fertile way? In fact, I intend to argue here that some popular linguistic theories are *not incomplete enough.* They are (or would be, if successful) too complete — while also, I will claim, incomplete in the wrong ways, leaving too many *relevant* phenomena, issues that *are* in its scientific wheelhouse, incompletely explained.

I will make these arguments as a prelude to describing the (incomplete) linguistic theory I *am* prepared to defend. Specifically, the first two sections here will weigh in on Conceptual Role Semantics and Truth-Theoretic Semantics and explain why I believe some popular paradigms in the philosophy of language are problematic. While the details will vary, the main thrust of my points will be that philosophers of language fail to appreciate the importance of sciences' internalizing a map of the division of labor between sciences — a science is constituted in part by how it touches but remains autonomous from other (both higher- and lower-scale) sciences. So biology is constituted in part by its status as a potential reductive base for (e.g.) neuroscience, medicine, genetics, and paleontology, while having its own reductive base in chemistry and physics. Part of what it means to be biology is to be the explanatory bridge between, say, medicine and physics.

Analogously, I believe, part of what it means to be linguistics is to be the explanatory bridge between, say, sociology, anthropology, and ethnolinguistics, with cognitive science (or Cognitive Phenomenology). Language can be intrinsically characterized as the cognitive bridge between our everyday world — of social situations and kinaesthetic/pragmatic enaction and anticipation, planning and memory — with the neurophysical substratum (whatever it is) of our mental faculties. Language, that is, is an important tool for our negotiating the duality of our higher-scale social/situational world with our lower-scale neurophysical existence. Analogously, a phenomenon in language — say, a sentence — should be analyzed as a kind of transition-system between a social/situational layer of reality and a cognitive/neurological layer. Linguistics is accordingly suspended between these layers — or, better, I claim that linguistics should be the *theory of being suspended* between social/situational and cognitive/neurological strata. A linguistic analysis starts with entities shooting in from the first stratum (sentences we hear uttered, canonically), and it ends with some restructured representation or consummation of that sentence (parsed, lexified, etc.) understood as inputs to some neurophysical process belonging (ontologically, and as a matter of scientific jurisdiction) to the second stratum. Such an analysis is *correctly* incomplete because it recognizes that a basic criterion of well-formedness for linguistic theories is that they *refrain from* direct analysis of either societal/interpersonal or cognitive/neurophysical processes. Linguistic analysis is incomplete because a theoretical machinery fine-tuned for analyzing processes of linguistic understanding at the intermediate level between social/situational and neurological strata cannot be the same as a theoretical machinery for analyzing either (in one explanatory direction) sociological or (in the other) neurophysical laws in turn — by analogy, the experimental (and theoretical) machinery for detecting Earthlike exoplanets cannot be the machinery for detecting Higgs bosons (and vice-versa).

Here I find an analogy to computer software useful: programs don't run themselves, so application developers realize that they do not control, or have access to much information about, when applications are launched (or when users will perform actions that require response from the software, like clicking a mouse button or press-

ing a key). Nor do programmers control input/output commands like emitting colors to the screen: they only influence electronic devices (like displays and networking capabilities) indirectly, via preimplemented system calls. In other words, the essential structure of a computer application is to be poised to react to various events (a mouse click, a key click, plus of course program startup initially) by eventually requesting certain operations (like changing the state of the screen) whose exact functioning remains outside the programmer's theoretical arsenal. Application developers have only a vague idea of how values and types in code are marshaled to and from electrical signals physically affecting (or reporting state from) devices like monitors, mouses, and keyboards. This is by design: if you are too closely attuned to low-level cyberphysical details, like how source code function calls map to digital signals, you're no longer doing computer programming (maybe you're doing chip design). To the degree that programming has a theory, it's a theory of how to *bridge* users' desired interactions with the software you are building *to* the digital structures encoded at the level of microprocessors and machine language. It is not a theory of microprocessors themselves. Theory well-formedness in the realm of programming — the field sometimes called Software Language Enginering — reflects the transforms bridging "Human Computer Interaction" with machine language; it is not a theory of HCI or of machine languages themselves. Indeed, HCI methodology is subjective and statistical; and the methods of physically realizing machine language in microprocessors depend on physical and nanochemical properties. Well-formed Software Language Enginering theories *have* to leave both HCI and microprocessors out of the frame, since software programming languages are not statistical or subjective, nor physical or nanochemical.

The hierarchical nature of computer architecture complicates any "mind as computer" metaphor: computers have many subsystems, with significantly different structures and properties. Using computers as case-studies of artifacts that are in some sense "intelligent" can take us in different directions for different answers as to what scale of computers' organization we propose to inform, say, cognitive-linguistic research: microprocessors? Machine Language? Programming languages? Software systems? The internet? Many instances of "mind as computer" analogic reasoning are not explicit research paradigms being proposed forth but are more like reports of intuitions: a community of linguists feeling that there's something going on in how computers work that usefully models or resembles how human rasoning or language-processing works. But cashing these intuitions into systematic models can prove challenging: even insofar as a computer may exhibit intelligent behavior, it does so only in an emergent manner, the whole "intelligence" being possible only through specific kinds of interaction between subsystems, in particular a tightly determined transition between high-level systems (like application source code) and low-level systems (like machine code).

Of course, many researchers probably believe that this emergent dimension is precisely why computers are a plausible cognitive analogy: they suggest that intelligence can be realized in structural systems whose lowest-level operations are not particularly complex. No-one would argue that in and of itself a simple Van Neumann machine is particularly "intelligent"; but software evincing intelligent behavior can be implemented as emergent phenomena for which Van Neumann machines are their reductive base. This may seem like a useful analogy to consciousness, realized in neurons and synapses even though neurons and synapses are not themselves conscious. That's an acceptable intuition, but it also leads to a kind of philosophical bait-and-switch: what starts as a "mind as computer" intuition ends up as a different kind of analogy, something more like comparing minds to functional systems *implemented* on a computer. There is a difference between being a system realized on a computer and actually being a computer.

In the case of language comprehension, someone may find a useful analogy in database-like constraint-solving applications, like Prolog: language users maintain an internal store of beliefs — about both language and the world — and a record of prior steps in the current conversation. This "database" gets updated as we hear new sentences, and we are equipped to make or reject inferences based on inference rules and constraints, respectively: from "John is my younger step-brother" we can conclude both that the speaker's parents are divorced and that John is not female. Of course, real-world complications sometimes intrude on the kinds of tidy frames linguists build around words: a brother can actually be a transgender woman, and divorcés can remarry each other. We can debate whether these are semantic or pragmatic issues (I think they're the former, but let's say they are the latter for sake of argument). So let's say

language has enough logical order that conversations can be modeled rather like Prolog programs. This leads to a maybe-interesting mind-as-Prolog analogy, but — here's the crux — mind-as-Prolog analogies are *not* mind-as-computer analogies. Computers *run* Prolog programs; it's not that they *are* Prolog sessions.

Indeed, I think many "mind-as-computer" analogies are actually more like "mind-as-Prolog" analogies, or substitute some other technology for Prolog. For instance, mind-as-artifical-neural-network analogies are not mind-as-computer analogies, because computers are not ANNs (though they may implement them). Indeed, ANNs are designed to make computers more humanlike: to transcend the mechanistic limitations of Van Neumann architecture by realizing, at some virtual level, a more connectionist manifestation of computation. A mind-as-ANN analogy is therefore really mind figured as a computer programmed to operate like a mind (so, the analogy is basically circular). Mind-as-symbol-processing analogies have similar issues: computers are not symbol-processors, though they can implement symbol processing systems. As I'll defend below, I think computers are basically stack machines, and stack machines do not "process" symbols — what they do is process stacks, and jump around to different subroutines. When people think about "computers" in mind-as-computer analogies — or write in ways suggestive of such an analogy — I often get the impression that what people are really thinking about is not "computers" but some sort of mathematically formalizable, functionally specified system that can be *realized* on a computer.

These other analogies are not *a priori* bad — it's reassuring if we have accounts of intelligence that traffic through functional organizations that can be realistically embodied in mundane physical artifacts, rather than needing some magical mind-gunk. But if our "mind-as-computer" analogies are nothing more than a desire to find logico-functional systems that can credibly undergird cognitive behavior, computers are basically irrelevant: the computational realizability of such logico-functional systems is a nice reminder that we're not asking non-philosophers to believe in magic, but the structure of these systems are sufficiently remote from how computers internally operate that computer realizability should have no *theoretical* role. In other words, mind-as-computer analogies are usually basically mind-as-logico-functional-system analogies.

We're entitled to find these latter analogies intuitive. My problem is only with mind-as-logico-functional-system analogies that get defended *by appeal to* mind-as-computer analogies. There seems to be a kind of metatheoretical pattern that goes something like this: mind is metaphorically a logico-functional system *because* mind is metaphorically a computer, and logico-functional systems (when not just abstract mathematical territories) are realized via implementation in computer architecture. But mind-as-computer is not logically related to mind-as-logico-functional-system — any apparent link between these analogies is a byproduct of intellectually backgrounding the distinction between *being* and *implementing*. We may or may not like a mind-as-computer analogy, but even if we *do* accept such a perspective, even if provisionally, this does not then legitimize or entail that we are accepting a mind-as-logico-functional-system analogy. Of course, we can judge the latter analogy on its own merits, but the former analogy in no way retroactively justifies the latter.

So, my strategy for the remaining sections of this paper is as follows: I will review some language-philosophy controversies and argue that disentangling mind-as-logico-functional-system from mind-as-computer analogies should change our estimation of theoretical claims apparently motivated by mind-as-logico-functional-system paradigms. I will then present a different basic account of language processing which, I believe, does not work in any logico-functional-system orientation, though I will recognize some fashion of functional orientation. I will in places appeal to computer architecture, though the framework I propose will only absorb "mind as computer" analogies to a limited, targeted extent. A central theme will be that *logic* is not terribly relevant for either language or computers: functionally-organized systems do not have to be *logico-functional* systems. The ambient philosophy guiding these arguments is that "logic" in any formal, symbolic sense is not the proper vehicle for understanding the structured transitions and causative propagation endemic to multiscale, emergent systems.

Biology, for example, is not a *logical* intermediary between medicine and physics. We can consider how best to describe its "intermediariness": as a theory-construction maxim (in the sense that intermediariness is expressed in which laws/observables are thematized and which are deferred to other sciences), as a causal network grounded in cross-scale physical constitutions and mereologies (e.g.,

tissues are both physically composed of cells and are wholes where cells are parts), as an emergent system which both *is* (vis-à-vis other sciences) and *has* a reductive base. Sciences are like computer programs in that they have *inputs* (observables from other sciences) and "outputs" (laws from other sciences). I use these terms because they track analytic trajectories: medical *observables* (e.g. that many people who are exposed to a toxin develop neurological damage) are linked *via biological analysis* to causal/material explanations (how the toxin chemically damages nerve cells). Biology neither statistically models the medical observations nor physically explains the causative mechanism, but it provides a theoretical machinery for rigorously modeling and analyzing the transition between them. It writes the second act of the explanatory play, so to speak. Pictured computationally, the analysis is like a computer program whose inputs are higher-scale observables (say, medical data) and whose "outputs" are numerical models whose formulae or justifications are solved by other sciences — by analogy to programmers calling system-kernel functions. In this analogy, medicine is like the end-user, biology is the application developer, and physics is the system kernel.

These are analogies informed by computers, of course, but I am trying to focus in on the "intermediariness" they evoke. How computer software bridges users and bare-metal is a useful metaphor for how scientific theories bridge observations and causal/mathematical microphysical models. And, correlatively, how sciences bridge between other sciences — higher sciences yielding observations that are "inputs" to intermediary explanations, lower sciences defining formats for "outputs". Biology, for example, can defer to chemical or physical explanation if it can provide data in structures adequate to chemical or physical formulae: the reference frames, quantitative measures, dimensional systems. Biology does not need to solve such equations, just marshal data into their form. So a good biological analysis will take observation data (e.g. from medicine) and transform it to equational data in some sense, wherein chemistry and physics take over. By analogy, correct computer software takes observations (data in computer files and user actions) and translates these observations to the proper system-kernel calls, wherein the Operating System takes over.

Sometimes logical constraints come to bear on these transitions, but the importance of logic per se is overshadowed by the overarching phenomenon of "intermediariness": how the technical and ontological status of computer applications is defined by their intermediary position between input data/user actions and low-level system calls. Analogously, sciences are characterized by their posits' ontological status as — and their theories' structural criteria regulated by — intermediariness between observational data from one peer science and causal/mathematical formula from another. As a philosophical gestalt, such "intermediariness", I believe, should take the place of "logic" in our intuitions.

In the specific contexts of Conceptual-Role and Truth-Thoretic Semantics, I will now show what for me this means in practice.

In the first part this paper, I will consider several semantic paradigms: Conceptual Role Semantics, Conceptual Space Theory, Truth-Theoretic Semantics, and ultimately Interface Theory of Meaning (a term due to Orlin Vakarelov). I approach this material as, hopefully, a mediator: I'd like to negotiate the differences and aggregate the best parts of these analyses, while avoiding reductionism and remaining true to phenomenology. I will eventually tie the pieces together into a semantic account, but in this case semantic analysis overlaps with theories of syntax, so I will develop this aggregative perspective toward the end of the paper, addressing syntax more head-on.

1 Conceptual Role Semantics and Externalism

Conceptual Role Semantics is often discussed together with a particular internalism/externalism debate which it tends to engender. Here I want to defend a kind of Conceptual Role Semantics (hereafter CRS) but I will first outline an account of compromise between externalism and internalism. I will suggest a compromise different, I believe, than Ned Block's "two factor" model that seems considered the leading example of an externalist/internalist hybrid.

The basic CRS picture is that linguistic meanings should be associated with conceptual roles in our understanding situations more than in terms of their reference to external objects. Given sentences like:

- (1) He opened the wine bottle with an ornate corkscrew.
- (2) He opened the beer bottle with a butterfly corkscrew.
- (3) He collects antique corkscrews and just bid on one online.
- (4) I thought this was a screw-top but it turns out I need a corkscrew.
- (5) This X3D file shows a very realistic corkscrew created with NURBS surfaces.
- (6) Could you send me the corkscrew (the X3D file you just mentioned)?

we should interpret "corkscrew", first, as a concept in a kind of functional organization. In some of these sentences there is also a specific corkscrew (qua phsyical object) on hand as a referent, but its actual physical properties — or even identity — is not decisive for the meaning of the sentence. After all, in (4) the speaker is not thinking of any corkscrew in particular (probably — more on that later) and in (5) and (6) the corkscrew is not real (at least not real qua corkscrew). But the conceptualization associated with "corkscrew" does not seem markedly different in (1) or (2) versus (4), at least (more on the other three later).

Not only physical details but even lexical identity seems tangential to the important conceptual meanings. Suppose I am hosting two guests, one has a magnum of ale and one a bottle of Malbec. They ask, respectively:

- (7) Do you have a bottle opener?
- (8) Could you get me a corkscrew?

and I give the first guest a butterfly corkscrew and the second a folding multi-knife. What I gave them is different from their request, but they should think nothing of it insofar as the winged corkscrew has a gap on its handle suitable for beer bottles and the multi-knife has a fold-out corkscrew helix. I have not violated any conversational maxims, because I reasonably assume that the instruments I gave them are suitable for the desired goals, of opening their bottles. Semantically "corkscrew" really means "something that can be used to open a wine bottle", and in that sense the lexeme gets its principle content from this operational role, not some list of attributes (like spirally and graspable) or prototypes.

Granted, a suitably designed winged corkscrew can be construed as a kind of bottle opener, and a multi-knife a kind of corkscrew respectively. We are prepared to accept these tools as examples of the respective concepts if they are functionally designed to support those tasks, even if such are not the primary function. But our inclination allowing concepts to dilate modulo functional criteria suggest that our grasp of concepts is first and foremost functional-pragmatic: we tend to internalize concepts in reference to (extralinguistic) functional roles and expand concepts to accommodate variegated implementers of those roles.

We can indeed accept sentences like:

- (9) He opened the bottle of beer with a hammer.
- (10) He pounded the nail with a lever corkscrew.

Of course here we are inserting objects into a conceptual nexus where they are not usually found. Winged corkscrews are often *designed* to double as bottle-openers, but lever corkscrews are not designed to double as hammers. Nevertheless we have no trouble imagining the scenarios being described, where someone uses the thick part of a corkscrew to pound a nail, or a hammer's handle/claw gap to pry off a bottle cap. We have schemata for "a tool to open a capped bottle" and "a tool to pound a nail", and the concepts of bottle-opener and hammer occupy that conceptual niche insofar as they are artifacts designed for those purposes. But the conceptual "slot" for, say, "a tool to open a capped bottle" is more general than the specific tools designed for those purposes.

We nonetheless *would* be presumably violating conversational maxims if we handed our friend who wanted to open a beer bottle a hammer. Even if there's a way to make the hammer work for that purpose, it's further outside the norm than, referring back to (2), proposing to use a winged corkscrew. So the implicature in (2) is satisfied, let's say, by bringing my guest a winged corkscrew, but not a hammer. But we can entertain the *thought* of using a hammer as a bottle-opener, and even this possibility presents problems for simplistic theories of language acquisition as essentially learning a static set of word correspondences, like "a hammer is used to pound nails" or "a corkscrew is used to open wine" — after all, you cannot conclude from:

- (11) A hammer is something used to pound nails, *and*
- (12) A lever corkscrew is something used to open wine, *and*
- (13) A lever corkscrew can be used to pound nails

that a hammer is a kind of lever corkscrew and can therefore open wine. What we *do* have are conceptual

slots available encapsulating ideas like "that which can open bottles" or "that which can pound nails", and we "fill" these conceptual slots with different lexical content in different situations. The "that which can open capped bottles" slot can be filled descriptively — i.e., in declarative speech, like in (9) — by a hammer, but not in other kinds of speech acts (we cannot read the concept "bottle opener" as satisfied by "hammer" in the context of a request for a bottle opner). Note that the scope of conceptual roles can change merely by switching between locutionary modalities.

The takeaway from this discussion in the internalism/externalism setting is that conceptual roles have a linguistic priority over and against both lexical and physical realizers, and the scope for things inside and outside of language to play (or not play) such roles varies with context. I have introduced these issues via tool artifacts (like corkscrews) but would be closer to the spirit of the CRS internalism/externalism debate by discussing natural-kind concepts. Suppose I am building a sand castle on a beach and ask someone one of:

- (14) Can you bring me a bucket of water?
- (15) Can you bring me a glass of water?

For (14), a reasonable reaction would be a bucket filled with ocean water; but for (15) my addressee would probably infer that I was thirsty, and — since salt water is non-potable — was requesting water I could drink. But "*glass of* water" probably figures here just to establish my intention to drink it: you are entitled to bring me bottle of water instead. In other words, my request has implied content which in some aspects loosens and in some aspects restricts the conceptual scope of semantic entries in my utterance. Thus oceans are composed of water, and near a beach I can say:

- (16) The ocean is over there.
- (17) The water is over there.
- (18) You can see the ocean from here.
- (19) You can see the water from here.

Each pair is almost identical. But ocean-water ceases to fall under the conceptual role of "water" when we are in the context of drinking things instead of the context of geography. This suggests that water does not "mean" H_2O or other saline or non-saline water: the meaning is not fixed to any particular chemical composition but adapts to the situational context, including what the water is used for — e.g. as a drink or as a binder for a sand castle.

The most-discussed "water" analysis in the literature is less earthly than this: Putnam's "twin earth" argument about a planet whose substance (with chemical makeup) XYZ functionally indistinguishable from our (H_2O) water. Externalists and internalists use this thought-experiment to express their differences as disagreements over whether twin-earthers' XYZ concept is the same as our H_2O concept. For the latter, as the basic account goes, XYZ plays the same conceptual role in their lifeworld as H_2O plays in ours, so it is the same concept; for the former, the concepts designate different material substances (even if twin-earthers don't know this) so they can't mean the same thing, even if there is some sort of analogy or resemblance between them (concepts can be analogous or similar while still being different concepts).

Before making a case for one alternative here over the other, let me note the following: it is unfortunate that the case-study is formulated in terms of XYZ vs. H_2O, because at the level of molecular composition it is hard for us to conceive that XYZ is *really* indistinguishable from water. After all, our conceptual understanding of water includes things like electrolysis — if XYZ does not emit hydrogen and oxygen when electrically charged under certain controlled conditions, it is not behaving like water and can not be (even internalistically) construed as conforming to our concept of water. Of course, we are free to expand our water-concept, just as we contract it when switching from geology/geography to drinking. But here we expand it with full recognition that finer-grained conceptual distinctions are possible, just that there are many contexts where they are unnecessary.

We do not need to contemplate far-fetched twin-earth scenarios to see this in practice: here on earth we have deuterium water which is chemically different from normal water (but both have the H_2O signature, although heavy water is also described as D_2O). We are free to let "X" mean normal hydrogen, "Y" mean deuterium ions, and "Z" mean oxygn, so XYZ becomes what chemists call HDO — semi-heavy water. Most people would probably say that HDO is just a kind of water, and so can be subsumed under the concept "water", but this is not conclusive. In reality, I don't think the English community

has needed to establish whether "water" should mean ordinary H_2O or should include variations containing different hydrogen isotopes — whether heavy and semi-heavy and other variants of water should be considered "water" or some other concepts.

In practice, a fraction of ocean water has deuterium, which might argue for "water" subsuming heavy water — we don't point to the ocean and say:

- (20) The water and the Deuterium Dioxide is over there.

But this can alternatively be explained by the principle that referring to an impure sample of a substance is still a valid use of the concept:

- (21) Here's a glass of water (even though tap water is mixed with fluoride).
- (22) Bing cherries are dark red (even though the stem is brown).

In the second case, we can validly call something red even if something less than its whole surface shows a red color. Applying a similar rule, we can call a solution "water" if there are only "sufficiently small" amounts of solutes. Clearly we use "water" to designate many substances other than pure H_2O. I can think of two options for explaining that semantically: (1) Salt water, tap water, distilled water, (semi) heavy water, etc., are all different kinds of water, but our coarser "water" concept subsumes them all (in most contexts). (2) There is only one water concept, pure H_2O, but impure samples of liquid that are mostly water can be called "water" by the same principle that a mostly red-colored object can be called just "red".

The second option has a common-sensical appeal because it fits a succinct "concepts as natural kinds" paradigm but does not venture too far from normal language use — that "red" actually means "mostly red" is a pattern common with many nouns and adjectives (someone can be *bald* with a bit of hair; I can point to a turkey burger made with bread crumbs and spices and say "that's turkey"; I can tell someone listening to Keny Arkana's song "Indignados" "that's French", although some of the lyrics are Spanish). However, the "mostly water" reading has a couple of problems: first, what about cases like a "glass of water" where "mostly water" is not "mostly" enough to drink? And, second, why can't we refer to plasma, say — which is 92% water — as water? This is not just a matter of numbers: the dead sea water is much less pure than plasma in the hospital (in terms of percentage H_2O in solution) yet we are authorized to call the former "water" but not the latter. This certainly seems to be a matter of conceptual roles — plasma occupies a certain place in our conceptual systems about blood and medicine (largely because it plays a specific role in biology and medicine) which does not fit the profile of "water", while the stuff in lakes *does* fit that profile, even if the lakes are hypersaline. Blood fits a conceptual ecosystem where we are not tempted to subsume it under the concept *water*, whereas our conceptualization of lakes pulls in the opposite direction — even though by purity the water in Gaet'ale Pond in Ethiopia is apparently not much more watery than blood. Our disposition to either contract or dilate the sense of "water" seems to be determined by context — by the conceptual role water plays in different context — rather than by actual hydrological properties.

What about the hypothetical twin-earth XYZ that Putnam imagines is indistinguishable from our H_2O? Well, for this hypothesis to even make sense we have to assume that XYZ is scientifically indistinguishable from water, which is a matter not just of pure H_2O but of all solutions and deuterium- or triterium-related variants of water, and so forth. As a thought experiment, where we are free to conceive almost anything, this is not impossible. Let's imagine that there is an undiscovered subatomic particle that on some planets clings to atomic nuclei without affecting them in almost any way. We can call nuclei harboring these particles "twin nuclei", so hydrogen becomes "twin hydrogen", oxygen becomes "twin oxygn", and presumably water becomes "twin water". This twin water would essentially retrace the the compositional structure of water — since it would have to form (and unform, under electrolysis) just like "our" water. If we plug this "twin water" into Putnam's scnario, I can't see why we don't just call this a variant kind of water, water with some extra (but observationally negligible) particles, just like heavy water is water with extra neutrons.

This does not do perfect justice to "twin earth" discussions, because I am describing "twin" water as something whose composition is almost identical to "our" water. In the original story, "twater" is XYZ, which as written suggests something whose physical constituents are much different than water, even if all propensities that

influence our "water" conceptualizations are exactly the same as with our water. But something compositionally different than water *can't* be functionally identical to water, at least if any of the actions we can take that reveal water's composition come out different. In short, whatever XYZ are, they must have a capability to *become* hydrogen and oxygen, because XYZ's emulating water means it emits hydrogen and oxygen under electrolysis. Meanwhile there is no action that could "release" the "X" (or whatever) because that would also behaviorally differ from water. So XYZ would differ from water only insofar as in its "unobserved" states it can float around as something without hydrogen or oxygen but, whenever subject to actions that cause water proper to emit these gasses, it would somehow conjure them up in exactly the same patterns as water (which actually *is* composed of hydrogen and oxygen) does.

By dictum, then, XYZ is not actually composed of hydrogen and oxygen, but whatever it *is* composed of can act as *as if* it *does* contain these gasses so as to emit them. In that case I'd question the argumentative force of claiming that XYZ does not contain hydrogen and oxygen to begin with. We are asked to believe that XYZ is made up of some ethereal non-hydrogen and non-oxygen that can nevertheless become hydrogen and oxygen whenever it is in the physical states wherein water that *is* made of hydrogen and oxygen will release them. I am inclined to say that this is just another way of being made of hydrogen and oxygen. After all, atoms are not little ping-pong balls: what we picture as a water molecule is actually apparently much more ethereal, suspended in quantum indeterminacy. I take it there is some Shrödinger equation for a water molecule, and only when the "wave function" collapses — say, by our observing the water subject to electrolysis — do we actually get hydrogen or oxygen atoms. So "our" water isn't really "composed" of hydrogen or oxygen in its pure quantum state. Maybe XYZ "collapses" to hydrogen or oxygen in different ways than earthly water (but with no way to measure the difference), but this is still not divergent enough that for me to feel compelled to call XYZ anything other than some variant form of water.

Of course, I am assuming that twin earthers have *the same* water-concept that we do, *in all respects*. Maybe a more faithful review would consider that twin earthers might have a related but more primitive water-concept than ours — maybe some subset of our concept in terms of the scientific knowledge embedded in our concept. Before we earthers knew about hydrogen, oxygen, or electrolysis, the behavior of water under electrolysis was not a factor in our concept of water. So imagine if twin earthers' level of scientific knowledge was akin to that on earth centuries ago — their XYZ is measurably different from our water, but they have no experimental or scientific apparatus to notice the difference. But this is *contingent*: the twin earthers *could* some day discover hydrogen and oxygen. Then, if XYZ really is not composed of hydrogen and oxygen (or acts as if composed of them when not in a nonobservable ethereal state) their scientific theory of water, and accordingly their conceptualization, would diverge from ours.

We can imagine a non-water XYZ that is water-like enough to play an identical role to (our) water, but this story can go in two directions: either XYZ is *absolutely* identical to water, its differences from water so obscure as to be observationally and causally meaningless; or it has legitimate differences from water that *could* be conceptually significant but in some context are not (at least not yet). These are two different thought experiments. If some substance is in all respects and under any conceivable science identical to water, yet somehow compositionally different from it, I think the plausible response among normal language communities would be to extend the concept of water — subsuming XYZ under the concept, analogous to heavy water when it was discovered. We are generally prepared to expand the reach of concepts when there is no compelling reason not to do so. Whether a potential expansion takes hold probably varies by context. We are — a point that generally fits on the externalist side of the ledger — more willing to accept expansion when the revised conceptualization would not deviate too far from a basic alignment of natural kind concepts to scientifically reasonable classifications. We can readily extend "water" to D_2O because the two substances are compositionally very similar. We are less likely to accept conceptual mergers when they seem to violate our natural-kind pictures, even if they are functionally plausible: we do not accept "agave" as a subconcept of "honey", even though the two are physically rather similar and functionally very similar. Nor does physical form alone drive conceptual boundaries: we know full well that water vapor and ice are the same stuff as liquid water, but we recognize a conceptual distinction between them.

But these are not hard and fast rules: we may be inclined in many contexts to treat frozen-concentrate juice as conceptually subsumed under "juice" (as in "juice on sale"), and we will often accept almond milk or cashew milk as "milk", despite physical differences which we certainly acknowledge. In short, conceptual boundaries tend to be drawn to honor, albeit without excess granularity, both physical and functional factors — neither physical/compositional similitude alone, in the absent of functional resemblance (see water/ice) tends to earn concept dilation, nor vice-versa, but a mixture of functional and physical similarity even with *some* differences in both aspects tend to be likelier drivers of concept-expansion (see water vs. chlorinated water, or red wine vs. white wine). By these rules, expanding "water" to include XYZ — if XYZ is functionally identical to *but* compositionally different from water — would be abnormal, like expanding "milk" to — without any qualification — include almond milk. But these rules are approximate, and on the idiosyncratic case where XYZ is *completely* functionally like water but (stipulated to be) physically different (though by functional identity we could not detect as much), I think the normal "conceptual dilation" rules would side with the functional identity and ignore the physical differences.

On the other hand, if XYZ has real discoverable differences from water, then the potential exists for twin earthers' concept of water to diverge from our own, even if at any point in time the concepts are identical. The time "points" don't need to be simultaneous: we can compare one country's concept of water in the year 1800 with a different county's in the 16th century. It is plausible that different people at different times have effectively the same conceptual attitudes toward concepts that, with the benefit of hindsight and more science, we know have potential for differentiation. I think the mere potential for differentiation warrants our identifying conceptual differences even if the parties involved are not aware of this potential. I am prepared, for example, to accept that a child's water-concept in our time can be different from a medieval child's water-concept merely by virtue of the modern child potentially learning about deuterium, hypersalinity, and other scientific nuances that complicate the modern conception of water relative to our forebearers.

1.1 *Divorce or Dilate? On Widening or Narrowing Concepts*

We certainly accept that people may have different understandings of a concept and, on that basis, may judge that what two people are entertaining are two different concepts — though we may also feel that they entertain two variations of *the same* concepts. There's room for most concepts to "diversify", subsuming subconcepts and variations; hence there's room for a concept to expand (see water to heavy water) without fragmenting. But sometimes we *do* insist on splitting concepts — or, equivalently, refuse to accept a concept-enlargement — and *the reasons for this refusal may be external to some peoples' use of the concepts.* Current political discourse in the United States, for example, is driven by turns of phrase that are rather haphazardly defined: *Border Wall*, *Green New Deal*, *Free Tuition*, etc. Suppose a health policy expert observes that Bernie Sanders's use of the term "Medicare for All" is different from Kamala Harris's. She may conclude that Sanders's concept "Medicare for All" is different from Harris's concept — and the rationale for this conclusion need not take into account whether the two candidates are aware of the differences. Suppose, as an expert, she has to mentally track the differences — she has a well-informed judgment that each of the "Medicare for All" plans have different ramifications due to policy differences; as a result when discussing "Medicare for All" she needs to note in her own mind which version of that idea is under discussion at any moment in a discourse. That is to say, she needs to subsume them under different concepts. Moreover, we endorse that she *should* do so, even if she thereby makes a distinction that the politicians or their supporters themselves do not realize. In this kind of case we may defer to expert opinion when adjudicating a potential conceptual divorce, even if there is only minimal differences in the role of the concepts vis-à-vis the conceptual systems of many relevant parties.

The possibility that "Medicare for All" may play the same *role* in a Sanders supporter's and a Harris supporter's conceptualizations does not preclude our judging that they are nonetheless different concepts — if by virtue of more information and more access to expert counsel we can understand that there are potential differences in their conceptualizations that *could* drive the

conceptual roles to diverge. I think this is analogous to a "twin earth XYZ" scenario in that the thought experiment is set up as if we have access to expert confirmation that twin earth's XYZ is not physically the same substance as water. Projecting from earthly practice, we accordingly accept that "externalist" considerations may need to come to bear, and "XYZ" may need to be classified as a different concept that water *notwithstanding* the lack of any conceptual role difference between XYZ for twin earthers as compared to water for us. This is consistent with our tolerance for including factors beyond just conceptual roles in more mundane circumstances: we accept that sufficiently divergent notions of "Medicare for All" *could* be most appropriately classified as two different concepts. Such is not mandated — we could certainly describe the Sanders and Harris platform as "two different Medicare for All plans", subsuming them under one concept but acknowledging their differences — as token differences, like the conceptual difference between this apple and that apple, rather than concept-differences like apple vs. cherry. Analogously, we *could* subsume XYZ under the concept *water* — XYZ being a kind of water insofar as samples of XYZ (tokens of the XYZ-concept) bear some physical differences to tokens of ordinary water (like heavy-water samples do), but we can handle this variation on a token-token level (analogous to comparing two apples). But we can *also* split rather than expand the conceps — *divorce* rather than *dilate* — making XYZ a different concept than water, just as we can make Sanders supporters' Medicare for All a different concept that Harris supporters'. The key point is that our choice of "divorce or dilate" may be driven by factors wholly external to some concept-bearers' internal concept-uses. Two different concepts — recognized by us as different — may play identical conceptual roles for some people.

This stance is at least minimally Externalist in that I don't insist on internal conceptual-role similitude being an immovable criteria selecting "dilate" over "divorce". We as a language community can and sometimes should override the tendency for concepts to expand under role considerations. As I pointed out earlier, a corkscrew and even a hammer can sometimes satisfy the role "bottle opener" in specific contexts. Usually we distinguish context-specific conceptual role-playing from general concept dilation — I think this is the gist of Zhaohui Luo's analysis of "situations" and "Manifest entries". We can adopt a temporary frame of reference wherein, say, hammers are bottle openers — or in Luo's example (in a single zoo exhibit) all animals are snakes — without mutating the concepts so wildly that "hammers" become expanded to including anything that may open a capped bottle, or "snakes" become all animals. Yet such situational dilations can recur and eventually spill beyond their situational guard rails. In a vegan cafe I can imagine the staff converging on a usage that soy, almond, and cashew milks are collectively called just "milk". If veganism becomes entrenched in some English-speaking community I can similarly imagine that in their dialect "milk" will mean anything that can be used like milk in a culinary context. The warrants for such expansions seem to be driven by conceptual roles — situations present "slots", like *that which opens this bottle* or *that which I pour on cereal*, and existing concepts tend to expand to fit these slots.

These considerations follow the *internalist* line: we take attitudes based on conceptual role more than external natural-kinds when adjudicating conceptual boundaries. Thus situationally we may present almond milk and agave to satisfy a request for milk and honey. But superimposed on such "centrifugal" tendency for concepts to expand into "under-lexified" conceptual niches we have a counter tendency to question conceptual uses where functional resemblance strays *too far* from common sense. Someone may accept agave in lieu of honey, or a hammer as a bottle opener, in the context of how one situation plays out; but they are less likely to accept these uses becoming entrenched, compared to, say, refiguring "milk" to include almond and cashew milk. And our hesitation to accept concept-expansion in these latter kinds of cases seems to implicitly look beyond conceptual roles — we may insist on limiting concept dilations even if there are many people for whom there will never be situations where the differences between concept referents, over and above functional resemblance, would be important. In short, even if a community could do just fine with some dialect idiosyncrasy that ignores a conceptual distinction we would ordinarily make, we don't tend to take this as evidence that our multiple concepts can be merged into one more diverse concept.

Of course we *can* merge concepts, and the fact that many people can live their lives without a conceptual coarsening may render such merger likelier, but it seems we evaluate potential mergers more by reference to entire speech-communities, not isolated parts. Note that I

am specifically talking here about merging or splitting concepts, not word-senses or lexemes or any purely linguistic artifacts. Certainly we have variegated "water" concepts — salt, tap, distilled, heavy — but we have an overarching water concept that includes these as subconcepts. We can make a conscious decision to modify concept/subconcept relations — which is different from changing how concepts are mapped to lexemes. So I take it that Conceptual Role Semantics prioritizes role factors in drawing concept/subconcept relations and boundaries, and the consequence is a mostly Internalist intuitive model: we should accept concept maps where concepts are mostly drawn together when there is a functional resemblance between their roles: our concept/subconcept renderings should witness and help us exploit functional analogies.

At the same time, however, I think we instinctively project notions of conceptual role outward from individual people or subcommunities to the social totality. Even if technically distinct Medicare for All plans play similar conceptual roles in different voters' conceptions, we understand that such similarity may break down as we expand the community outward. Sanders and Haris supporters don't live on their own islands. There are factors outside their own minds that weigh on whether their functionally similar Medicare for All concepts are indeed *the same concept* from the larger community's point of view. But these external factors are not necessarily *extramental*: we can zoom outside the conceptual patterns of one subcommunity and argue that conceptual differences appear in the overall speech community that supersede functional resemblance in some subcommunity. Conceptual roles are not solipsistic: the role of the concept Medicare for All for a Sanders supporter is not just a role in *her* mind, but it becomes a role in *our* minds if we dialogically interact with her.

Insofar as people can make inferences about other peoples' conceptual role "system" — we can figure out the role which a concept plays in someone else's mind, to some approximation, even if analogous concepts play a different role in our own minds — conceptual roles are not private affairs; they have some public manifestation and there is a need for collective reconciliation of role differences, just as we need to identify when different people are using the same words in different ways and use lexical conventions to diminish the chance of confusion. To the extent that they have this public dimension, conceptual roles are not *internal*. But "externalism" in this sense is warranted because we want to look philosophically at entire speech or cognitive communities — it is not automatically a philosophy of conceptual content being external to "mind in general". Conceptual differences that could *potentially* become publicly observable from the vantage point of the *entire* cognitive community warrant consideration for conceptual divorce over dilation — overriding similar roles in some *part* of the community.

In the case of XYZ, insofar as the twin earth cognitive community and our own could *potentially* become part of a single overarching cognitive community, we have potential grounds for drawing comparisons between water and XYZ. Merely by contemplating their planet here on earth we are performatively drawing twin earthers into our cognitive community. By postulating that twin earthers think about XYZ the same way that we think about water — and that we know this — we implicitly assume that their conceptual role patterns are public observables in the context of our own community. If conceptual roles are observable, then there is a concept of a conceptual role: pundits can conceptually analyze how "Medicare for All" plays identical conceptual roles for Sanders and Harris supporters even if the candidates' plans are consequentially different. But this merely says that there are latent differences in two people's conceptual roles that they themselves may not actually experience. The public facet of conceptual roles complicates the notion of conceptual role similarity — two people's patterns of conceptual roles may be observably different as public phenomena even if they lack resources to realize the difference. Conceptual roles are therefore external to individual minds — but this is by scoping outside individual minds to holistic cognitive communities who can publicly observe our cognitive tendencies. We are still reasoning "internalistically" in the sense of considering cognitive patterns at the scale of an overall cognitive community.

In short, I will take the mantra of an "Externalist" when passing from individual minds and subcommunities to the public nature of conceptual roles and overarching cognitive communities. Once we get to the maximal possible community, however, I am inclined to revert to Internalism: if there is no broadening of communal scope that could make putative external differences meaningful to *anyone's* conceptual roles, I see no reason to account for *these* erstwhile externalities in a theory of concepts. If

XYZ has *some* not-water-like qualities that a sufficiently large cognitive community could confront — even if XYZ-conceptual-role and earthly-water-conceptual-role is identical for the two isolated communities — I am happy to accept that twin earthers' XYZ-concept is a different concept than earthers' water-concept. Similarly, I accept that Sanders supporters' Medicare for All concept may be a different concept than Harris supporters'. But in both cases I accept concept splitting to override role-similarity because I believe in an overarching cognitive community which has an interest in detecting differences or potential differences in conceptual roles qua public observables, which transcends our own internal awareness of what our conceptual roles entail. The fact that earthers and twin-earthers might never "discover" a water/XYZ difference is a contingent fact, not an essential structure in policing conceptual maps. When establishing how we should consider redrawing these maps, we should work from the picture of an overarching community — that can subsume isolated communities — as an abstract posit; the parts of the twin earth story that imply earthers and twin earthers could never actually discover their differences are not, I think, compelling as intrinsic features of the analysis. In short, if water and XYZ have some potentially observable differences, then we need to proceed as the community which is aware that these differences exist and that therefore, for us, water and XYZ need different conceptual slots. The only analysis then is how to reconcile the fact that we have multiple conceptual slots whereas twin earthers (and earthers who have not read Hillary Putnam) have just one.

But if we take a *maximal* cognitive community — the sum total of earthers and twin earthers and philosophers — this community *does* distinguish XYZ from water (surely XYZ plays a different role in Putnam's mind than water). And we should scope to the maximal community when determining whether smaller communities' conceptual roles are truly identical, because conceptual roles are, in part, potential public observables for any possible supercommunity.

On the other hand, if XYZ is so much like water that *no* community would *ever* have reason to contrast twin-earthers's XYZ-conceptual-role with our water-conceptual-role, then I think these roles are not just *internally* identical for each (twin-) earther, but *publicly* identical for any conceivable cognitive community for whom public observations of (twin-) earthers' conceptualizations are consequential givens. And in *that* case I think XYZ is the same concept as water notwithstanding putative compositional differences.

The whole idea that conceptual roles can be *public* complicates the Internalist/Externalist distinction, because each person's conceptual patterns can be evaluated from a vantage point external to *their* mind but still within the proclivities of a "maximal" cognitive community. Conceptual roles are not private to each person, but are private inclinations that get reshaped, corrected, influenced, or reinterpreted by a larger community. If we understand conceptual roles to include the totality not just of each person's conceptual role attitudes but the totality of how these attitudes are observed by others, then we should consider that concepts are not "external" to the *maximal* cognitive comunity. Externalism about *individual* minds can be wrapped inside Internalism at the *maximal* inter-cognitive level.

But, complicating matters further, the maximal community's observations of conceptual-role attitudes is often driven by at least our *beliefs* about external (i.e., extramental, natural-kind) criteria. For example, some companies want to rechristen "corn syrup" as "corn sugar", to make it seem more like a sugar-subconcept. Meanwhile, some dairy companies want laws restricting the use of "milk" for vegan products. In both cases our larger community has a chance to weigh the proper conventions for how our conceptual maps should be drawn. As I argued earlier, both functional and naturalistic criteria play a role in such deliberations. We are poised to distinguish transient situation-specific roles — that one time someone used a hammer as a bottle opener — from functional parallels that stretch across many contexts. Within the parameters of that contrast, we are receptive to redrawing maps on role criteria — allowing milk to subsume vegan milk-substitutes, for instance. But this tendency is balanced by a respect for some notion of coherent natural kinds — the distinct biological properties of vegan milks work against a *maximal* community subsuming them under "milk" outside of special contexts.

Both the Externalist and Internalist points of view have some traffic in the considerations that cognitive communities bring to bear on which conceptual maps should be endorsed by convention. Because ad-hoc con-

ceptual roles can be established for particular situations, we can be conservative about *conventionalizing* concept maps driven by functional correspondences too far removed from (what we think to be) scientifically endorsed, natural-kind boundaries. In other words, I think we *do* and *should* allow "naturalistic" considerations to be a factor in what concept maps we endorse. But this is not a claim about Externalism as a philosophical paradigm shaping how we should construe the triangulation between mind, world, and language, as a matter of metaphysical ideology. Rather I believe that "externalist" factors should and do come to bear on the deliberations *internal* to cognitive communities' (sometimes but not always explicit) evaluations of how to draw concept and subconcept boundaries and relations — when to split concepts and when to dilate them. Dilate-or-divorce options are pulled by both externalist and internalist considerations, sometimes in competing ways.

As as case-study, the wording "corn sugar" — which implies a "redistricting" wherein the concept "corn syrup" becomes part of the territory "sugar" — may be credible on purely biochemical grounds. But our community may feel that there is enough functional difference between sugar and corn syrup from a commercial and nutritional sense to reject a proposed merger — here functional considerations trump natural-kind ones. Conversely, the community may be sympathetic to claims that milk substitutes should be labeled to clearly indicate how they are are not *literally* milk — here natural-kind considerations trump functional ones.

If we consider language — and communally-endorsed conceptualizations — evolving in practice, then, by light of my claims until here, there is material for both Externalist and Internalist readings. This perhaps leaves room for a theory which accepts that both are partially true — each being logically founded under consideration of two different aspects of how concepts evolve. I will explore this possibility further, but first I want to shore up my account of conceptual roles themselves.

One complication I have glossed so far is that *functional* roles in an enactive and "pragmatic" (in the everyday-world sense) spheres are not *ipso facto* the same as either conceptualizations (conceptual-role-attitudes) or lexicosemantic conventions. These three are interrelated, but we need social and cognitive practices to get situational understandings entrenched in language and in communal concept-maps. Without a theory of this process, to speak of functional roles like *hammer* for *bottle opener* is not a substitute for speaking of conceptual roles *per se*. How to properly link "functionality" in an enactive quotidian sense — the data that various natural and man-made artifacts are used by people for concrete tasks, and we often talk about this — to the cognitive realm of concepts (and their boundaries and subconcept relations)? This is the main theme of Section 2. I will however conclude the present section by reviewing a useful critique of the conventional Externalism/Internalism dialectic. I will focus in on Orlin Vakarelov's "Interface Theory of Meaning", developed over several recent papers, which I will also (somewhat indirectly) use as a kind of metatheoretic guide when presenting my own theoretical attachments later on.

1.2 Orlin Vakarelov's Interface Theory of Maning

Vakarelov's theory (which I'll abbreviate to ITM) both critiques and suggests ways around the Externalism/Internalism impasse:

> An externalist theory focuses on constraints outside of the user of the informational state. Particularly, it focuses on the relation between the informational state and the sources or object of the information. The meat of the semantic connection derives from some nomic (or teleonomic) connection between the source system and the information medium (receiver) system. The focus of semantics for an externalist theory is the determination of the way the world is. ... An internalist theory, on the other hand, considers as the primary constraint of meaning what the information state does for the user. The model of the internalist account is not reference fixation and fact determination, but message interpretation. The question that an internalist asks is not *what m means*, but *what m means to a given user*. Of course, for *m* to be informative about the world, it better be sufficiently correlated with a source, but this is not a constitutive condition of the meaning of *m*. It is a condition of a good interpretation sys-

tem. ... One strategy for reconciling externalism and internalism is to take a hybrid account of meaning/content. Such hybrid theories are motivated by an observation that external or internal considerations are not sufficiently fine grained. ... Such hybrid theories of meaning have targeted cognitive information media — languages, mental states (beliefs), etc. This analysis of meaning cannot easily transfer to the domain of dynamical semantic information. In the case of dynamical semantic information, the externalist and internalist conceptions of meaning collapse into a single notion. The reason for this is the codetermination of macrostate structure of informational systems. [87, pages 13-14]

He then presents his ITM alternative (for terminological clarification, his symbol M roughly matches what I have called "cognitive frames", and S roughly matches our environing situations):

> It follows that neither an external relation between M and S, nor an internal function of "selecting conditional readiness states" is sufficient to provide a general notion of meaning, for they don't even fix the syntax of the information system independently. To specify the meaning of a state m we must do something different. What does M really do in the information system? It acts as an *interface* between the (external) world and the control system. It structures influences to allow focused purposeful control. If any sense of significance can be given to a particular state m of M, it must be related to this interface function. The significance of m is neither that it tracks something external nor that it can affect the control mechanisms of the system, but that it can connect one to the other. ... Let us go back to the observation that the definition collapses the external and internal conception of meaning. Specifying the differential interface function of a state requires looking at the entire system/environment complex. We can think of the datum state m as participating in a process of interaction where causal effects from the environment are channeled through the internal M-P control pathway to produce actions, which actions modify the system's behavior, and which in turn changes the state of the environment (including the relations between the system and other external systems). [87, pages 15-16]

Finally he extends this definition of *meaning* toward language itself:

> The story gets more interesting when ... the system utilizes different sub-systems that act as information media. The system may have [different] media, each with different roles and interface connections. Some media may be connected to different external systems or different aspects of the same systems, others may interface with other media, yet others may be connected with effectors or control the states of other media, etc. When the system is organized as a complex network of information media, complex interface (sub-)functions can emerge. Some can depend almost exclusively on external connections to outside sources, others can be analyzed entirely in terms of their control role or effects on other media. I conjecture that the canonical examples of information media that shape many of our intuitions about semantics are media that exist (within an information system) as only one of a large network of other information media that jointly control the system's behavior. Thus, to take correspondence theories of meaning as an example, it is tempting to say that the word 'chair' means a property of external objects. Thus, in the expression, "This is a chair", the meaning is given by some fact in the world that the object depicted by the indexical has the property of chairhood. In an information system using language we can analyze this idea in a different way. The language medium, whose datum may be some structural equivalent to the expression "This is a chair", interacts with other nonlinguistic media connected to perception, allowing the system to identify and interact with patterns in the world that can be clustered through some data state of some internal media. To make Fodor happy, we can assume that there is a single medium that gets in an information state uniquely correlated with chairhood — a kind of a concept of "chair". The language system,

> in this picture, is not interfaced with the world (or some abstract realm of propositions). It is interfaced with other information media. The properties of the interface relations look a lot like the properties that a correspondence semantics may have, but these interface relations do not capture the true interface roles of the language datums for the information system. To determine the true interface role, we need to link all local interfaces and see how the entire complex participates in the purposeful behavior. [87, page 17]

Interestingly, Vakarelov speaks not of "prelinguistic" cognition but of "precognitive" systems. This is partly, I believe, because Vakarelov wants to understand cognition as adaptation: "Nature, in its nomic patterns, offers many opportunities for data systems that can be given semantic significance, it offers ubiquitous potential datums, but it does not offer any well-defined and complete data sets" [87, page 4]. As I read it, Vakarelov conceives cognitive systems as dynamic systems that try to adapt to other dynamic systems — these latter being the environments where we (taking humans as example cognitive systems) need to act purposefully and intelligently. The "nomic patterns" are latent in our surroundings, and not created by intellect. So *this* kind of worldly order lies "outside" cognition in an ontological sense; it is not an order which exists (in itself) in our minds (though it may be mirrored there). Consciousness comports to an "extramentally" ordered world. However, "precognitive" does not necessarily mean "extramental": there is a difference between being *aware* of structural regularities in our environment, which we can perhaps deem a form of pre-cognitive mentality, and trying to *interpret* these regularities for practical benefit (and maybe a subjective desire for knowledge).

When distinguishing "cognitive" from "precognitive", however, we should also recognize the different connotations that the term "cognitive" itself has in different academic communities. In the context of Cognitive Linguistics, the term takes on an interpretive and phenomenological dimension which carries noticeably different implications in the "semantics of the theory" than in, say, conventional AI research. Vakarelov's strategy is to approach *human* cognition as one manifestation of structured systems which we can visualize as concentric circle, each ring implying greater sophistication and more rigorous criteriology than its outer neighbor:

> What is the function of cognition? By answering this question it becomes possible to investigate what are the simplest cognitive systems. It addresses the question by treating cognition as a solution to a design problem. It defines a nested sequence of design problems: (1) How can a system persist? (2) How can a system affect its environment to improve its persistence? (3) How can a system utilize better information from the environment to select better actions? And, (4) How can a system reduce its inherent informational limitations to achieve more successful behavior? This provides a corresponding nested sequence of system classes: (1) autonomous systems, (2) (re)active autonomous systems, (3) informationally controlled autonomous systems (autonomous agents), and (4) cognitive systems. [88, page 83]

> The most rudimentary design problem begins here: if there is cognition, there must be a system. Without a condition allowing a system to exist as an entity discernible from its environment and persisting sufficiently long as that same entity to allow qualification of its dynamical behavior, the question of cognition does not arise. The first design question that must be examined is: What allows systems to persist as individual entities? More specifically: For which of those systems that persist is a capacity of cognition relevant? [88, page 85]

But this intuition that human cognition can thematically extend out to other "cognitive systems" and then other structured systems — out of which *cognition* emerges by adding on criteria: is the system autonomous; reactive; information-controlled — suggests we are dealing with a different concept than in Cognitive Linguistics or Cognitive Phenomenology. For Vakarelov, "cognition, like most other biological categories, defines a gradation, not a precise boundary — thus, we can at best hope to define a direction of gradation of a capacity and a class of systems for which the capacity is relevant; [and] cognition is an operational capacity, that is, it is a condition on mechanisms of the system, not merely on the behavior of

the system — to say that a system is cognitive is to say something general about how the system does something, not only what it does" (p. 85). Conversely, the qualities that make "grammar", say, "Cognitive" seem uniquely human: our sociality in the complexity of social arrangements and cultural transmission; our "theory of other minds". Certainly animals can have society, culture, and empathy, but the human mind evidently takes these to a qualitatively higher level, making language *qua* cognitive system possible.

This argument does not challenge Vakarelov's program directly, but perhaps it shifts the emphasis. Our cognition may be only one example of cognitive systems — which in turn are examples of more general autonomous/reactive/information-controlled systems — but there may still be distinct phenomenological and existential qualities to how *we* achieve cognition, certainly including human language. I think there are several distinct features we can identify with respect to *human* "cognitive frames", which call for a distinct pattern of analysis compared to generic "M" systems, in Vakarelov's terms.

I'll mention the following:

Multi-Scale Situationality We understand situations as immediate contexts for our thoughts and actions, but we also recognize situations as parts of larger contexts, and connected to each other in chains stretching into past and future. For example, as a train pulls into a subway station, our immediate situation may be needing to determine if this is the train we need to board. But this is linked to the larger situation of traveling to our destination; and situations are strung together as enactive episodes: once I determine which is the correct train, I need to enact the process of boarding and getting comfortable on the train, then get ready to reverse the process and disembark at my station. All of this inter-situational orchestration can be planned and facilitated, to the degree that multiple people are involved, through language.

Conversational Frames Our *cognitive* frames modeling situations and our immediate environments include models of ongoing *conversations*. I think this is an example of what Vakarelov calls "subsystems": within our intellectual "systems" that track outside reality, there is a part that specifically tracks what people are saying — so that we can take note of what they believe, how they are using different words, what they consider or would deem relevant to the current topic (or situation) — all of which helps us use language to reason through situations intersubjectively. I will discuss the architecture of conversation frames more in Section 3.

Conceptual Roles We have, I believe, a unique ability to fuse perceptual and conceptual detail in understanding situations. That is, we identify objects perceptually while also placing them in a contextual matrix, where functional properties may be foregrounded above directly perceptual ones. If, say, we hear someone ask for a glass of water and see someone else hand her one, we understand the glass not only through its sensate qualities — or even through our pragmatic/operational interpretations, like believing that the solidity of the glass prevents the water from leaking out — but we also interpret people's practical intentions and mental attitudes. We infer that the first person was thirsty and the second cooperated by providing her with water to quench her thirst. Interpreting the situation at that interpersonal level, not just at a sensory/perceptual or a force-dynamic level, enables us to understand situational variations, like responding to requests for a *glass* of water by bringing a *bottle*.

In short, to understand *how* our cognitive frames align with environing patterns we have to understand the role language plays in this process: a role which can be intersubjective, empathic, context-sensitive, defined by conceptual substitutions and interpersonal cues as much as by rigid rules.

And yet, I think Vakarelov's larger point remains in force: we need to get beyond both Externalism and Internalism in the sense that we need to get beyond a debate as to whether *words* have "intramental" or "extramental" *meanings*. For instance, we need to think past an apparent choice between deciding that the word "water" has a *meaning* which is either intramental (determined by the sum of each person's beliefs and dispositions about water) or extramental (determined by how our water-experiences are structured, even beyond our knowledge, by the physical nature of water). In place of either option, we should say that the meaning of the word *water* — or *chair*, in Vakarelov's example — depends on all

the cognitive systems interacting with linguistic understanding. The word or concept does not exist in our "language-processing system" in isolation; so its meaning is not just *linguistic* meaning but how word-tokens and concept-instances become passed from system to system.

Insofar as we have a token of the word *water* — presumably tied to a concept-instance — the specific fact of our hearing the word is joined in with a plethora of other perceptual and rational events. Say, we hear someone ask for water, and soon after see someone bring her a glass. We instinctively connect our perceptual apprehension of the glass of water with the word heard spoken before, and we presumably remain vaguely aware of the situation as things unfold — if we see her drink from the glass, we connect this to our memory of her asking for water, indicating thirst, and then getting a glass in response. We do not need to track these affairs very attentively — it's not like we should or need to stare at her intently while she drinks — but it fades into the background rationality that we tend to attribute to day-to-day affairs. Her glass of water — how it continues to serve a useful purpose, how she and maybe others interact with it — becomes a stable if rather mundane part of our current situation.

In Vakarelov's words,

> To determine whether a particular macro-state of S is informationally relevant, i.e. whether it is differentially significant for the purposeful behavior of the system, we must trace the dynamical trajectories of the system and determine ... whether the microstate variation within the macro-states is insignificant for the purposeful behavior .... Let us call such macro-states *informationally stable*. [87, page 15]

An intrinsic dimension of situational models, surely, is that they recognize the relatively stable patterns of situations: a glass placed on a table will typically remain there until someone moves it. Situations are, in this sense, large compilations of distinct quanta of relative stability: in a dining context, every glass or plate or knife, every chair or table, every seated person, is an island of relative stability, whose state will change gradually if at all. So a large part of our cognitive processing can be seen as recognizing and tracking these stabilities. Stability is the underlying medium through which situational models are formed.

Ultimately, many cognitive systems contribute to such models: quanta of stability lie in the cross-hairs of multiple cognitive modalities. So we connect the water spoken about to water in a glass. If we have our own glass we connect both the linguistic and visual content to the tactile feel of the glass and the kinaesthetic intentionality exercised as we pick it up. We can imagine concepts like *this water* pinging between these various cognitive registers.

I take Vakerlov's ITM model (or metatheory, maybe) as saying that we should look at *meaning* through the interstices between systems, not as some semiotic accounting summed up either "inside" or "outside" the mind. The meaning of a broad concept like *water* is subsidiary to the meaning of more context-bound concepts like *glass of water*, *body of water*, *running water*: and to excavate conceptual meanings in these situationally anchored cognitions we need to think through the *conceptual roles* we instinctively pin onto the concept-exemplified: whether manifest as an element of language or perception/enaction, or both.

2 Conceptual Space Theory and the Interface Theory of Meaning

The phraseology that language is an "interface" — to some (at least partly) prelinguistic cognitive faculties — is inspired by Vakarelov's "interface theory of meaning", which I described at the end of Section 1. Here I want to explore something like an ITM as an extension to (or perhaps a foundation for) older language-philosophy paradigms, like Cognitive Grammar and Conceptual Role Semantics. I'll spend this section forecasting how that might work.

Conceptual Space Theory, on the other hand, originates with Peter Gärdenfors's books and articles — especially 2000's *Conceptual Spaces: The Geometry of Thought* — but has branched from linguistics to disciplines like computer science and the Philosophy of Science. Conceptual Space Theory emerges from Cognitive Linguistics, and therefore resists simple AI paradigms of language following essentially mechanical rules, or logically decoding and processing symbols. But at the same time, Gärdenfors argues that we can find some

quantitative structure in conceptual structures — including identifying axes of variation where notions of prototype and exemplars can be formally modeled; and representing perceptual and spatial features through numeric measures, like the color double-pyramid (which grounds the Hue, Saturation, Value color space widely used for computer graphics).

To the extent that perceptual and spatial features can be quantified, it is easy to develop intuitions for prototype theories and conceptual similarity: for example, an exemplary *red tablecloth* would have a certain almost-red hue and rectangular dimensions. Varying the color or making the shape too large or small, or oblong, corresponds to moving from the "prototypical" space of the concept to borderline cases. These examples are appealing because they suggest that conceptual dispositions can be systematically modeled; the behind-the-scenes mental gears that classify something as *tablecloth* or *knife* seem to have some scientifically tractable lawfulness, not just a cognitive black box that linguistics takes for granted.

As I see it, the challenge for Conceptual Space Theory is how to generalize outside the intuitively trenchant but rather narrow examples of conceptual "quantification", like color and shape, to model the full range of details — including functional and conceptual roles as well as perceptual form — which influence conceptualization. After all, while there *are* spatial differences between a *tablecloth*, *placemat*, and *ribbon* — or between a *knife*, *sword*, and *cleaver* — these represent different *concepts* because the objects serve different enactive ends. Their various spatial morphologies are byproducts of practical design, and do not *cause* differences in conceptualization, although it is often via spatial form that we *recognize* an object as a knife, etc. Integrating Conceptual Spaces into a multifaceted *cognitive* linguistics would seem to call for examining Conceptual Spaces not just as vehicles for *object recognition* but within the spectrum of interpersonal, enactive, and situational understanding that lies behind linguistic signification and performance.

Research in the overall context of Conceptual Space Theory has, however, examined these more situational and functional dimensions. There are several tactics for cashing the basic "Geometry of Thought" metaphor outside the obvious geometric model of, say, prototypical tablecloths having prototypical rectangular dimensions.

One option is to consider conceptual "space" as encompassing enactive dimensions as well as spatial/perceptual ones. We can do certain familiar things with tablecloths: place them over a table, fold them, launder them; and with knives: place them on a table, sharpen them, wash them under running water, use them to cut food. The more that two objects share a similar set of affordances, the more that they are likely to be conceptually similar. As such, conceptual *roles* can substitute for "metrizable" dimensions (like color and shape, which can be directly quantified in a distance space) as a ground for modeling similarity and prototypicality.

Another idea is to consider the kind of (canonically perceptual) attributes which via Conceptual Space we can analyze quantitatively as *triggers* to more multi-faceted cognitive activity. When we see an object which *looks* like a prototypical knife or tablecloth, this spurs further conceptualizations — and/or enactive engagement with the object — that thematize the object more functionally and situationally. The perceptual triggers then need to be analyzed as part of the overall cognitive process. In that case the "geometric" space where concepts can be situated represents not so much the definitive cognitive stature of the object, but a *provisional* conceptualization which unfolds into more complex (and less directly perceptual) machinations.

This latter model actually integrates well with Vakarelov's ITM: as I already argued in Section 1, language is best viewed as an integration of multiple cognitive subsystems. One such subsystem, which acts like a perceptual-cognitive interface, may be well-served by a Conceptual Space model that stays close to Gärdenfors's original geometric metaphor. Other subsystems, engaged more with the list of affordances embodied in a concept (and, concretely, in its tokens) needs to be modeled with a more abstract/functional kind of "space".

Given this possibility, I believe that the systematizing gambits of ITM and Conceptual Space Theory can be integrated. In this section I will argue that formal attempts to implement Conceptual Spaces in computational settings are consistent with the ITM architecture. This does not mean, however, that a Conceptual Space/ITM hybrid can be unproblematically lifted to a computational model for human cognition. The idea that our overall cognition *integrates* many subsystems means that a computational theory of any one subsystem is not necessarily a step toward genuine AI. It may be that there

is something saliently human about how we *integrate* the totality of our cognitive dispositions into socialized, context-sensitive, empathic behaviors.

So in this section I will generally approach efforts to *operationalize* Conceptual Spaces — and look back to Vakarelov's paradigm as well — with a mixture of critique and endoresement. In general I think that these theories are useful models — or at least useful starting-points — for understanding components within cognitive systems which are *locally* structured and formalizable. This does not mean I endorse a *holisitic* picture of human cognition as simplistically computational or simulatable. In any kind of "interface theory", there is an implicit distinction between *local* analysis and *global* systematic qualities. An interface is, canonically, poised between two other structures: it can be analyzed internally through the lens of its own structures — how it effects translations and routing between the structures of other systems which the interface interconnects — but this "local" analysis does not address how local structures fit into the "semantics" of the whole.

The "semantics of the whole" is often where science gives way to Philosophy of Science — or even to Phenomenology. The "local" language of chemistry, for example, principally describes phenomena at the molecular level, like chemical bonds and intermolecular forces. Analogously, the local language of biology describes phenomena at the cellular level, like compounds diffusing in the blood stream. The chemistry-scale phenomena may provide causal-explanatory grounds for the biological — chemical properties of blood and alcohol, for example, dictate how alcohol enters the blood stream. But we need a holistic integration to perceive this at a higher level, as an empirical phenomenon affecting the organism as a whole: alcohol enters the bloodstream and can impair our normal cognitive functioning, even causing harm if consumed at toxic levels. We need the everyday concepts and language — e.g., describing someone as *drunk* or *poisoned* — to orient the biological and chemical languages to empirical reality. Biology and chemistry are not abstract systems; they are intended to explain phenomena in the world, but *which* phenomena they are explaining is not something captured "locally" in either biological or chemical language.

To the degree, then, that we can ("metascientifically") analyze the interaction between biological and chemical laws/properties as an *interface*, an "interface theory" of this relationship — of biology supervening on chemistry, for instance, goes hand-in-hand with a *holistic* theory of the worldly phenomena which biology and chemistry (and their interaction/reuction) explain. The interface theory is not a self-contained explanation, but a *local* analysis which needs an overarching holistic picture to cement its explanatory value.

I believe this biology/chemistry case is a good metaphor for cognitive science in relation to Cognitive Linguistics and to phenomenology. We can theorize various cognitive subsystems, analogous to chemistry and biology — suppose we take both perceptual and affordance-based Conceptual Space models, Vakarelov's M and S subsystems, and the proto-computational frameworks like *feature vectors* or *expectations* developed in a Conceptual Space framework as I will mention below — so these theories become analogs in the explanation of human *mind* to chemistry and biology in explanations of the human *body*. But these are still local analyses, and we need an overarching account of how different cognitive subsystems "interface together" to yield, as an emergent totality, what we experience as human mind and consciousness.

Ultimately, I believe this holistic picture needs to be developed at a philosophical level, rooted in fields like phenomenology and Cognitive Grammar. This means that these fields should recognize the formal merit of scientific — even computational — analyses of *aspects* of cognition while arguing against reductive theories of cognition and consciousness as a holistic reality. In the absence of a phenomenological paradigm which is willing to both engage and transcend subsystem analyses, our *holistic* picture of mind tends to be dominated by AI and logical or computational reductionism. That's a subject for the next section; my goal here is to look at Conceptual Space theory as a useful but partial "subsystem" theory.

2.1 Conceptual Space Theory and Phenomenology

Towards the end of Section 1 I noted the contrast between how the word "cognitive" itself seems to do different "theoretical work" in Cognitive Linguistics compared

to, say, AI research. I also argued that the differences are not necessarily irreconcilable: while humans are not the only cognitive system, there is a distinctly human way of *being* a cognitive system. Analogously, while humans are not the only communicative species — actually, we are not the only *linguistic* species; it seems counter-productive not call bird sounds or cetacean and primate vocalizations as a kind of language — there is however a distinctly human way of *being* linguistic. Not all language is *human* language; but language which *is* human absorbs the specificity of human sociality and consciousness into its signifying processes. It is on *that* level, I would argue, that we should read the "cognition" in Cognitive Grammar, Cognitive Linguistics, or Cognitive Phenomenology.

Taking Langacker's Cognitive Grammar as canonical, I think scholars in that tradition would agree that we instinctively reach for cognitive frames to interpret linguistically-encoded situations. Research can uncover structural features of linguistic understanding by identifying frequent structural primitives of these frames: consider the landmark-trajector structure in (23), the force-dynamic contrast in (24) vs. (25) and (26) vs. (27), and the spatial/geometric variations in (28)-(30):

- (23) Our house is across the lake.
- (24) I poured wine from a decanter.
- (25) Some wine spilled from the decanter.
- (26) I put spackle on the wall with a knife.
- (27) Paint spattered all over the wall after a can droppped.
- (28) There's a purple-and-blue color pattern all over the wall.
- (29) There are drawings all over the wall.
- (30) There's a plastic sheet all over the wall.

There are underlying perceptual gestalts which seem apparent in these examples, and their linguistic expression seems to take these as cognitive-perceptual primitives rather than grist for analysis. This is consistent with the phenomenological intuition that consciousness includes a primordial structural awareness, and the role of intellect and attention is to focus on local regions of the whole structural cloth of experience, for enactive deliberateness and/or information extraction at a level of precision that "raw" experience cannot provide. The important phenomenological contrast is not between "sense data", on the one hand, and intellectually filtered or reified world-apprehension, on the other; but rather between a structured cognitive-perceptual complex which we feel as *ambient* experience and, within that, an actively thematized attentional focal-region that we experience ourselves to be forcefully studying and interacting with.

For phenomenology, then, ambient "background experience" is already richly structured and is not really "pre-cognitive", because its structure evinces the "grammar" of cognitive frames. On the other hand, there are other intellectual traditions where "cognitive" carries more of a rational-analytic overtone. I suspect those who identify as Cognitive Linguists understand the word in a more phenomenological mien, whereas the AI and formal logic community places greater emphasis on how cognitive *systems* may be formally tractable. This can yield confusion in linguistics proper, where AI (at least in the sense of Natural Language Processing) and Cognitive Linguistics co-exist. One solution is to qualify "cognitive" in contexts where confusion could arise, e.g. "cognitive-perceptual" as a more phenomenological sense and "cognitive-analytic" as a more computational sense.

Note, however, that the re-occurrence of "cognitive" in both terms is accidental: as suggested by the terminological pattern, I think we should see "cognitive-perceptual" and "cognitive-analytic" as part of a spectrum whose "axis" represents attention and dispositional structurality. That is to say, on the more cognitive-percepetual side we may be aware of structural details (cf. Vakarelov's "nomic patterns") but do not consciously attend to them, such that they remain latent as the manner of disclosure of sensate content: for example, the way in which a certain car appears as red is to appear as a metallic red hue with a glinting lighter patch following the length of the car. This perceptual complex has geometric structure — it is not an undifferentiated red-sensation — but I comport to such content specificity in a passive manner. Towards the other (cognitive-analytic) end of the spectrum, I deliberately seek out awareness of structural forms, analyzing them in relation to schemas and prototypes (consider a rock-climber planning how to scale a wall). Within this spectrum I think there are continuous gradations; and such a picture seems more phenomenologically well-motivated than a cognitive/pre-cognitive duality.

Concepts qua cognitive tools are influential across this spectrum. An architect analyzing the facade of a historic building will experience its structure in greater detail and attention than a bystander who's meeting a friend in front of the building. The concept "facade" will nonetheless shape how both people make sense of their surroundings. The bystander may have a more passive acknowledgment that she is before the facade, compared to the architect (but it will nonetheless be part of her relatively deliberate attempt to coordinate with her friend's expectation that they meet in front of the building). Moreover, a child who had not yet learned the word "facade" would see the characteristics of buildings' exterior that fall under the concept, but more passively still. Merely learning the word presumably alters our perception of exteriors qua facades vs. "ordinary" exteriors, even if we are not currently using the word in any conversation — just as the word "hail" sharpens our perception to how hail differs from snow, since we have a compilation of beliefs specific to *hail* (apart from *snow*), and thinking (even if passively) that some precipitation is the former, not the latter, triggers us to activate those hail-specific beliefs. Another analogous case would be identifying milk as actually almond milk, or water as actually salt-water: the more granular our inventory of lexicalized concepts, the more precise becomes the package of prior knowledge we instinctively make on hand in the current situation.

Insofar as knowledge of the word reinforces the concept, we can assume the concept and our disposition to name it lexically is latent in situations where the concept *may* be relevant. Thus the friend might comment on the facade once they have met in front of the building: making explicit something that hitherto the parties, we assume, had just passively noticed. This is an example of the kind of unstated assumptions about others' beliefs that lie beneath explicit linguistic content: "I love this building's facade" presupposes both that the hearer sees the facade and understands the concept.

I use the "facade" example strategically, to reference Martin Raubal's analysis of this word via Conceptual Space Theory [65]. Raubal proposes a "conceptual vector space" to distinguish *facades* from other spatial arrangements that (for instance) we might encounter outdoors in an urban setting. His apparent goal is to quantitatively model the terrain of "facade" in contrast to other, lexically related words, which would yield a basically mechanical, computationally tractable account of how to recognize a facade — perhaps for programming a robot, or a navigation tool for people, as he proposes.

Such potential applications trade on the possibility that we can reach beneath the nuance of language and uncover logically straightforward encodings of, or criteriology for, concepts. Obviously, finding a logical matrix beneath the surface fluidity of language is an essential first step toward legitimate Natural Language Processing.

But trying to map an everyday (e.g., non-technical) concept to a readily-enumerated "feature vector" is not without problems, I think. Conceptual Space Theory is not the same as a prototype-based semantics, but it could share some of its problems when dealing with shape-shifting everyday concepts; the likes of *house* or *restaurant* or *water*. A prototype (or feature-vector) theory of *house* would need to unify mansions with hovels but exclude hotels, tents, apartments, apartment-buildings, and historical estates that have become museums. The criteria for "house" and "restaurant" seem mostly functional, although we are still aware in English of a conceptual incongruity in extending the concept on purely functional terms. We can acceptably use "house", really, for any place of residence — and restaurant for anywhere to buy prepared meals:

- (31) I'm going to a party at my brother's house (suppose he actually lives in an apartment).
- (32) This restaurant has the best Hokkien noodles (said of a stall in a Chinatown food court).

These feel (at least to my ears) like idiomatic expressions, however, as if we know not to casually overstretch the concepts. As I proposed earlier, our criteria for concept-mappings seems to be *mostly* functional but to incorporate spatial, configurational, visual, and natural-kind features also as *secondary* criteria. I would argue that a Conceptual Space model intuitively grounded on these latter features would supplement, rather than displace, a Conceptual Role theory (Conceptual Space Theory does account for functional roles, but arguably a little awkwardly).[3]

[3] For instance, Raubal says that "Meanings of concepts change over time and depending on the context in which they are used. In a conceptual vector space it is possible to account for these changes by adding or deleting quality dimensions and by assigning different saliences (as weights) to the existing dimensions" [65, page 5]. For sure, our readiness to (continuing my example) accept "house" for any place of residence

But setting this objection aside, we can defer to Raubal's analysis to the effect that a "conceptual vector space" can model our disposition to actively or passively identify concept-instances as such. Standing before a building, the proper synergy between properties of a a facade and my own mental "vector" of the colors, spatial arrangements, patterns, and so forth iconifying the idea "facade" — if the synergy resonates enough — primes me to know instinctively that the exterior is a facade, a pasive belief which could potentially be "activated" should that become relevant. One way this could happen is if a conversation partner says something about "this facade" — entering that referent in the "ledger" of dialogically salient things and topics.

So the efficacy of the concept lies not just in the reality available for us to perceive, nor in our minds, but in a synergy between reality-structures and activatable conceptual models. This kind of partial-but-not-total externalism is perhaps roughly what Vakarelov considers to be "precognitive": the phenomenon of our perceiving an exterior as a "facade" depends on both mental and extramental factors. Gärdenfors's Conceptual Space Theory can be seen in this context as an attempt to imagine an "abstract geometry" to quantify (or to suggestively intimate the possibility of quantifying) the world-to-word fit that predetermines (and is witnessed by) well-founded conceptualizations. Gärdenfors's "geometry of thought" can accordingly be seen as an attempt to capture via quantitative intuition an insight Vakarelov's ITM broaches qualitatively: the idea that cognition is a structural *correlation* between the reality out there and what we're equipped to conceptualize.

Perhaps, then, Conceptual Space Theory is (or can be applied for) one example of a Vakarelov-style ITM. Raubal proposes conceptual vector spaces not just as theoretical explanantia but as technological artifacts; he envisions software employing these vectors as assistive technologies capable of some natural-language understanding. A computational system which properly activated the "facade" concept, let's say, given sufficiently proximate feature-vectors, would perhaps exemplify Vakarelov's idea of an "information system" that resembles human cognition, to some functional degree. The fact that such-and-such an environmental given resembles (in the conceptual-space-vector metric) a prototypical facade, or falls in the facade "region" (in a high-dimesnional concept-vector space), acts as a kind of input or signal. For Vakarelov, such quasi-cognitive (or actually cognitive, like the human mind) systems are organized in layers; it is consistent with his subsystem model to say that concept-vector metrics would be recognized by one subsystem, as "effectors", generating signals to be received by other subsystems. One such signal would be, say, a passive awareness that — based on distances in some feature vector — we are now standing before the facade of a building.

Another computational strategy for Conceptual Space Theory is suggested by Kenneth Holmqvist's chapter on "conceptual engineering" (mentioned by Raubal's paper I've cited, but also noteworthy as an unusual attempt to apply computational methods to Cognitive Linguistics). Whereas Raubal skirts around functional-role issues, Holmqvist acknowledges that functionality can be the decisive factor in conceptual frames. He cites the example of a knife, which can on the one hand be prototyped spatially and mereologically (e.g., the relative sizes of blade and handle and the knife's status as the sum of those parts), but also functionally — "Take the lexical unit *knife* as an example ... *blade* and *handle* are clearly parts of *knife* [which also] has *silverware* as a *whole*: *knife* is one of the parts in collections making up silverware. But *knife* can also have *cut* as a whole, because *knife* can be the agent ... of the cutting process" [34, page 155]. As is clear, Holmqvist adopts mereology as a very broad domain of relations, representing different functional and aggregative connections as special cases of part/wholeness. But more significant is that Holmqvist (given this generality) is prepared to model a broad range of relationships — even if these can in principle be expressed mereologically (like a knife as part of a silverware set), we are not restricted to only visual or physical partonomies.

The parts of Holmqvist's analyses that are more perceptually grounded are also the more prototype-like. He comments, for instance, that "saying ... *blade* is part of *knife* is not sufficient. We must characterize this part-whole relation closer. For instance, the relative sizes of

varies with context: the idiomatic usage in (31) is less proper in the context of real estate transactions, or assessing property tax (an available apartment should not be called a "house for sale"). But while "assigning different saliences" may capture the relative weight of functional vs. more prototype-based classifications, attempts to quantify functional dimensions themselves as if they were, say, colors and spatial geometries — which do have convincing quantitative models (e.g. "red" on an HSV color pyramid) — strike me as forced and unpersuasive.

the blade and knife must not deviate outside certain limits. The relative spatial position of the blade and knife must also be correct, i.e., the blade must be correctly attached". This implies that the criteria for classifying something as a knife can be quantified, and regions on certain perceptual axes — say, the shape, length, and position of the handle and the blade — carve out (no pun intended) the conceptual space of *knife* from peer concepts like *sword*, *cleaver*, and *spatula*. Certainly such clusters of related lexemes suggest conceptual "terrains" that can be "mapped" — as in my earlier discussion of concept-mapping for water and milk — and Conceptual Space Theory draws on our intuition that such mappings are particularly elegant when there is a readily quantifiable system of dimensions that can be identified, like blade-length distinguishing knives from swords. Again, however, functional pragmatics, more than spatial form in itself, seems to dictate when and how we identify concept-instances with their concepts.

Holmqvist however recognizes this possibility by talking not only of perceptual part/wholes (like blade/knife) but of mereologies with more functional inflection, like a knife in a silverware set or as part of "cut" insofar as "cutting" something can be a perceptual-operational gestalt, whose "parts" are both the agent and patient of cutting. These more abstract mereologies find linguistic expression in cases like:

- (33) He had to cut the crusty bread with a serrated knife.
- (34) The museum had antique butter knives with intricate carvings.

The implied situational picture in each case is structured, in part, mereologically: a museum-piece knife potentially part of a valuable cutlery set; and when slicing bread with a serrated knife the knife is part of an enactive process. However, I'd say the functional position of the knives in these various situations is the key detail, over and above the partonomic significance of situational wholes. A butter knife rests in a different niche in culinary situations than a bread knife. Their roles are however similar enough that we can subsume them under a common knife-concept, although we can likewise distinguish them, *bread*-knife and *butter*-knife forming two sub-concepts.

We should highlight the functional roles because these dispose us to recognize the concept and the subconcepts. We reach for a butter knife if we want to butter bread; it is that practical goal which primes us to see the butter knife as a knife, in general, and a butter knife, in particular. Insofar as there is a synergy between our mind and our environment, manifest in the adequacy of concepts like "butter knife", this is primarily a matter of — in this case — the object conceptualized as a butter knife being suited for that task. Of course, part of the reason *why* it is so suited is how it is shaped and manufactured. Geometric and physical details are therefore relevant for our inclination to identify (butter) knives. Mostly, however, these details are derivative on functional roles, rather than the preeminent criteria of conceptualizations.

Having said that, an unused butter knife is still a butter knife. Table settings include butter knives so we can conveniently reach for one as needed. Our appreciation that we *might* need a butter knife, or how *some* diners might need one, and how they are used, informs our conceptualizing dispositions. It is true that perceptual details like color and shape provide visual cues to the nature of objects — partly because they need the design and material composition they have to perform their intended purpose. But we don't just troll sense-data looking for cues; our perceptual awareness is not a matter of decontextualized equations like "shiny and sharp means *knife*", "liquid and clear means *water*", etc. Our receptivity to concept-instances depends on our awareness of current situations. It's not like we are prepared to see examples of every kind of object that we are familiar with in every situation. We anticipate finding butter knives on a dining table, or in a kitchen. Situational awareness brings with it a selective anticipation — knowing what kinds of objects are likely to be associated with each situation prevents our having to devote excess thought to identifying objects, or misidentifying similar-looking ones.

So even if we accept features like color and shape as "triggers" for concept-recognition, our receptivity to these triggers is conditioned by situational understanding — which is an example of cognitive frames. These frames, moreover, are defined in terms of functional roles: the salient characteristic of a bread knife is the fact that it can cut bread, and the salient characteristic of a butter knife is the fact that it can spread butter. The situation provides the conceptual slots that objects can fit into.

There is, notwithstanding my suggestions to this point, a version of prototype theory which *is* broadly applicable. Situational understanding, we can say, *does* proceed from *situational* prototypes, so here is a domain where prototype theories are appropriate. Instead of a prototypical *knife* (or house, restaurant, corkscrew, etc.), I think we have *prototypical situations* where knives (etc.) play a role. Any particular knife is conceptualized against such a background: one kind of scenario is someone at the head of the table ceremonially carving a roast, wherein the knife is a "carving knife"; another scenario is someone spreading butter, wherein it is a "butter knife"; etc. Each situation-prototype is an architecture of roles, where for instance there is a person enacting the carving ritual, the instrument she uses, the food being carved, and so on. The building-blocks of these architectures then become solicited within language, for instance via case-markers like benefactive, locative, patientive: "carving *the turkey* for *grandma* with *the knife* at *the counter*".

In practice, our sensitivity to functional roles allows for ad-hoc practical configurations, like using a hammer as a bottle-opener. To the degree that situation-prototypes are *abstract* models, we nonetheless have narrower appraisals of functional roles: the lexeme "bottle opener" covers objects playing that role in *prototypical* situations, which is why it does not cover hammers. This is one reason why we should accept conceptual-role talk as more parsimonious than functional-role talk: conceptual roles *are* functional roles, but tapered down by the prototypicality of situations abstractly conceived.

In practical affairs, of course, we comport to real situations — that may embody situational prototypes, but no real context, with its idiosyncratic details, is entirely prototypical. This concreteness has a pair of distinct implications for my current analysis. First, we accept localized expansions of conceptual roles, like bottle-opener-to-corkscrew or even -to-hammer. Second, conceptual roles offer templates that allow cognitive-perceptual judgments to be passive or instinctive — we reach for a butter knife without being aware of concluding that said instrument is a butter knife, or even really being aware of knowing that a butter knife is there.

So the practical purpose of curating a "library" of conceptual-role accounts is to prime us, given each situation, to identify objects fulfilling conceptual roles *passively*, as part of unattended, background consciousness. Once we become aware of specific enactive needs — the thought that we need a knife or a corkscrew, part of some practical task being now phenomenologically active, a focus of attention — the more passive perceptual details (like a knife's shape and color) are poised to trigger more active conceptual recognition. Now we become consciously aware of the butter knife nearby, and of picking it up and using it.

Perceptual details can certainly be triggers of conceptual recognition, but a complex interleave of situational awareness, situational prototypes, pre-learned conceptual roles, and moment-to-moment enactive needs and processes, all establish an infrastructure within which perceptual content can actually "trigger" determinate conceptualizations. Most of this activity is prelinguistic — it establishes a cognitive baseline that language builds off of. But there is enough commonality between different persons' situational models that we can understand how these cognitive processes are working for other people, and therefore can draft them into the circle of language: if we, holding a slice of bread, ask someone for a butter-knife, we trust they will instinctively grasp both my enactive requirements and have the cognitive resources to help achieve them.

In sum, our ability to convert passive situational awareness and "background consciousness" perception, mediated hy situation-prototypes, into active cognitive-perceptual conceptualizations and pragmatic representations (of the "here's a butter knife I can use" variety) is itself, in total, a cognitive system which can be *targeted* by language, however much it is itself prelinguistic. That analysis, if it holds water, would make language an *interface* to the aforementioned cognitive system. Under that interpretation, my reading, originating in Conceptual Space Theory and then pivoting to Conceptual Roles, also presents as a flavor of ITM. I envision this hybrid theory as a kind of synthesis of Conceptual Space Theory, Conceptual Role Semantics, and (at least some variation on) Vakarelov's Interface Theory of Meaning.

Situational awareness, and situationally-mediated object recognition and associated conceptualizations, are highly subtle and multifaceted cognitive faculties — especially in our purposeful, socially normative, often emotionally charged human world. Some aspects of this

overall architecture may be interestingly modeled or emulated with computers. Examples include Raubal's and Holmqvist's implementations based on Conceptual Spaces, or Holmqvist's approach also intended to present computational models of Langacker-style Cognitive Linguistics, a goal shared by some other work, like Matt Selway's [72]. To this we could add certain models embraced by phenomenologists like Barry Smith and his collaborators (notably [12]) and Jean Petitot. I am skeptical that computer implementations will ever achieve more than a rough approximation of human enaction or language understanding — valuable perhaps as a research case-study and for specific useful tools, but nothing like robotic substitutes for human bodies and minds. But computer tools can still play an important role in research, by giving formal outlines to cognitive architectures which appear to have some formal dynamics, even if the raw materials of cognition — like sensation, situational awareness, and empathy — may not be formally tractable.

2.2 *The Chinese Room Revisited*

John Searle's "Chinese Room" argument — about someone who behaves like he understands Chinse by matching characters to responses from a vast table — is often understood as claiming that "symbol processing" by itself can never produce real understanding, which is *semantic* and *conceptual*. Modern technology makes this thought-experiment less hypothetical: automated telephone systems often use a template mechanism that is practically like Searle's Chinese Room, understanding a limited range of sentences and producing a limited range of responses. But there are two different kinds of questions we can ask in relation to Searle's argument: some more philosophical and some more practical.

On the philosophical side, we should properly assess the important questions as being qualitative and not quantitative: it's not as if a synthesized phone system is just not a very *good* conversationalist; it's that a software machine simply isn't the *kind of thing* that we can say actually understands language. This is plausible if we say that emotions and empathy are intrinsic to language; that we can't properly understand language if we do not grasp the emotions residing behind expressions. Indeed, as the case of Grandma's window shows (which I analyzed in Part I, where we should try to satisfy her request to close a window which in fact is already closed, perhaps finding a window which *is* open and causing a draft), our status as competent interlopers depends on reading intentions behind expressions, and it seems hard to do this if we can't experientially empathize with our linguistic partners.

Maybe we are now just pushing the important questions back to reappear: Ok, can computers be programmed to feel emotions? Is there a meaningful distinction between meaningfully, experientially having emotions and just behaving as if you have them? Are emotions themselves somehow functionalizable apart from their chemical/hormonal substrate so that systems with very different physical realization than ours can be said to have emotions? I can see how such a debate can go different ways. But I'd also argue that any well-organized dialog about these questions will be only tangentially about language — in which case, neither linguistics nor philosophy of language themselves can answer questions about what kind of systems (on metaphysical criteria) actually "do" language. That would imply that affirming a computer's linguistic capabilities as *real* linguistic understanding is a disciplinary non-sequitur for linguistics proper. Nothing in the linguist's arsenal either demonstrates or depends on AI agents actually *being* part of our linguistic community or just mimicking language-use to some (sometimes helpful) degree.

The more practical questions raised by Searle's Chinese Room come into play to the degree that the philosophical trail I just sketched turns many analyses into a non-starter. Consider these two questions to a hypothetical automated telephone service:

- (35) What time does the office open?
- (36) What time does train 100 depart from Newark?

While we can see a template holding canned responses for both cases, (36) needs to do more than just fit the input to the nearest pattern; it has to pull out the dynamically variant details (train *100* from *Newark*) and use those to fill in details in the response. This is something like *parsing* the original question. So we can add bits and pieces of genuine linguistic processing to a minimal response-template system — a real version of what Searle appeared to imagine in the Room. With enough added features the primitive template-driven

kernel can evolve into a complex AI-powered Natural Language Processor.

In that case we may imagine that "language understanding" exists on a spectrum. The primitive telephone service and an erudite bard may lie on opposite ends of a spectrum, but they share a spectrum between them. In this case, their differences are quantitative more than qualitative. The bard just has more *features* we associate with total linguistic behavior.

However, this quantitative view still leaves open the question of where among the "features" do we have something that actually drives language competence? Searle's Chinese Room helps point out these questions: it is reasonable to say that the simplest template-response system does not really understand language at all, since it is a pattern-matching system that does not have any structural relation to language itself. Analogous capabilities can be developed for a system which matches any kind of input to a pattern directing an output, based on any metric of similarity. The patterning reflects an actual *linguistic* parse only insofar as it selects elements via syntactic criteria, like grasping the non-template variables as *100* and *Newark*. So, even if the holistic behavior different systems lies on a linguistic-competence scale, not all *parts* of the system seem to bear the weight of actually *realizing* linguistic competence equally.

One reading of the Chinese Room is that *no* part of a system is truly linguistic. This includes the argument that holistically the Chinese Room *does* speak Chinese: Searle's discussion suggests that no *part* understands Chinese, but if we can imagine the entire room as a single system this "entity" can be treated as a fluent Chinese speaker. Even if we reject that analysis, we could agree that, even among humans, *parts* of our language system arguably do not understand language: not nerve cells, not neural clusters for auditory processing, or syntax, or conceptualization, etc. It is us, the whole system, that uses language. The reason why "holistic" claims that "the entire room" speaks Chinese sound dubious may not be because something is *structurally* lacking in that whole system, but because it's not the kind of whole system — with one body, one consciousness, one personhood — that we think of as a conversant.

Those who find Searle's analysis compelling probably believe that there *is* some meaningful difference between us (or at least people fluent in Chinese) and the Chinese Room. A further alternative, however, is that *we* are not language-users, at least not in the way we think we are. This claim can be expounded as follows: the philosophy of language, interactively with linguistics, seems to be looking for some essential kernel of linguistic capability that distinguishes us from AI engines or template-response system. That is, AI-skeptics want to sift through all the models of processes within languages, the central domains in linguistics, and find the few genres of linguistic processing that are unique to human language — and computationally intractable. These would be the smoking gun evidence that no artificial system can equate to human language-use because there is some essential stage in the linguistic pipeline that computers computationally can't realize.

However, even if we accept as premises first that the Chinese Room case suggests this analysis — and second that it agrees with our underlying intuitions — there remains the possibility that computers are indeed lacking some stage associated with language — but it is not a *linguistic* stage. If something like an Interface Theory of Meaning is correct, all linguistic processing is intermediary to some other cognitive layer: and perhaps the human quintessence lies on the far side, so it both limits what computers can linguistically achieve and lies outside of linguistics proper.

2.3 Distinguishing Computational Models From AI

I contend we need to tease apart the pursuit of valuable computational tools and models from an (often reductionistic) paradigm of seeking artificial, computationally engineered replicas of human cognition. *Computational* does not have to equal *AI* [93].

Holmqvist's and Selway's research that I have cited are good examples of paradigm-overlap between cognitive and computational linguistics. I will cite other scholarship which also finds philosophical inspiration in cognitive linguists like Langacker, Gärdenfors, George Lakoff, and Eleanor Rosch, but which also target cognitive-science formalizations and "cognitive architecture": [46], [94], [39], etc. A recurring pattern in this scholarship is to *first* propose a structural intermediate representation — a model of intellectual structures which plausibly em-

body the processing of language and cognitive-perceptual content, partly abstracted from surface-level sensory or signifying details — and *second* propose algorithmic or software models of how our minds translate linguistic and perceptual givens to abstract, or partly-abstract, schema.

I have argued that we bring abstract situational prototypes to bear on understanding all of the world and social situations around us, and that language taps into these models so that people can coordinate situation-appropriate activity. Given that there is an abstract and schematic dimension to how we understand situations, we should expect a partially abstract sheen to how we intellectually engage objects and concepts once they are situationally "located". Having identified objects as butter or carving knives, pitchers or glasses of water, wine or beer bottles, corkscrews and bottle openers — identifications themselves mediated by situational awareness, viz. if we are hosting or attending a dinner party — we no longer often attend actively to sense-perceptual minutiae. Our mental map of our surroundings — there's the corkscrew, there's the carving knife — pulls these referents outside the register of sensate consciousness and into the pragmatic hum of worldly activity. Insofar as they nestle in our intellectual faculties in that semi-abstract state, it seems fair to capture the schematic, structural appearance they have in this intellectual register — phenomena without the full-cloth phenomenology.

This in turn seems to invite us to imagine how the structural essentials of such "pragmatic appearance" may be captured by computers. We do not need to endow computers with human consciousness or emotions, because our mental traffic with the corkscrew or carving knife at some point evolves outside the sensate and passionate fabric of momentary consciousness. There is a schematic and mechanical dimension of human action, and we can imagine computers simulating human intelligence at least on *that* theatrical level.

Or at least, such seems to be the intuition behind attempts to model our human representations of objects and concepts in terms of abstract structures. But even a feasible theory of these semi-abstract layers of cognitive processing is only half the story. Suppose we agree that there are legitimate cognitive insights in Holmqvist's model of cognitive frames, incorporating (but also extending, including in a more pragmatist direction) Conceptual Space Theory — employing a generalized mereology that renders objects and concepts as *parts* of situations (I have suggested a more conceptual-role account for analogous phenomena). Suppose also we find plausible cognitive-frame models in Selway's intermediate representation for natural language, via which his proposed implementation can potentially map natural language to formal specifications. In these cases we have potentially valuable Intermediate Representations which capture cognition, in effect, mid-stream, or in-the-act: neither conscious phenomenology nor neurophysical hardware.

However, Holmqvist's and Selways' work appears to operate in an environment where these Intermediate Representations are valued primarily because and insofar as they allow human cognition to be mechanically recapitulated. This of course demands not only that computers *represent* IR models, but also *create* them — that is, when presented with an artifact of natural language, or the visual data of a scene, that computers should *automatically* map these givens to the theorized IR models, as if retracing the steps of human intelligence.

But just because IR models can be given computational form and representations, it does not automatically follow that automated generation of IRs is possible or effective. We can and should thereby distinguish the computational *study* of cognitive Intermediate Representation from the AI vision of programming computers not just to *host* but to *derive* Intermediate Representations. For instance, given a theory of the correct model for parsed Natural Language sentences, we can use computers to *study* and *present* parses — but this is separate from attempting to program computers to parse NL samples to such models on their own, without human intervention. I am sympathetic to the former methodology but skeptical of the latter.

I also believe that most research in, e.g., computational linguistics, ends up conflating those two goals. In that case, IR models are judged based on whether they facilitate automated, AI-driven generation of IR, not on whether the IRs are insightful suggestions of how human cognition itself builds an intermediary cognitive register — particularly if we accept Vakarelov's overall picture of language as an interface between speech-givens and prelinguistic cognitive faculties. Interface theories and Intermediate Representations tend to go together — the IR is the representation of some input during intermedi-

ate processing yielding an output; a structure between two other structures, where the role of the interface is to bridge the structures as well as to activate the correct capabilities via the output. This is the architecture of an "interface theory", in science or computer programming; it carries over to linguistics if we take Vakarelov's ITM seriously.

An equally intrinsic aspect of interface theories, however, is that the processes operative at the intermediate level are theoretically distinct from the realms which the interface bridges. For example, the theory of programming-language compilers and runtimes is distinct both from the theory of programming-language parsers and specifications, and from the theory of CPU architecture and system-kernel development. Runtime engineers can work through the medium of IR models, and compiler design itself is split between parsing surface-level source code *to* IR and mapping IR structures to their proper runtime paths of execution. It would be a breach of design architecture to attempt to solve source-to-IR problems within modules devoted to IR-to-runtime engineering.

Unfortunately, I get the sense that AI research does not respect a comparably disciplined Separation Of Concerns. There are multiple parts to a typical AI platform — modules for representing information (or knowledge/facts/beliefs, or the state of the system's physical or digital environs, etc.); for populating these representations with data deliberately introduced by human users or absorbed via some real-time engineering from the outside world; for analyzing representations to glean insights or calculate a course of action. Individual parts of the overall architecture can evince noteworthy engineering achievements, separate from the goals of the overall system. In this sense the pursuit of AI can yield positive contributions in other branches of computer science and other disciplines, without the stated rationale of AI realizing systems that exhibit humanlike intelligence. In this sense, perhaps "AI" is best understood as shorthand for a suite of research agendas across several aspects of computer science, not restricted to the fields — like Machine Learning, Robotics, and Artificial Neural Networks — that are publicly associated with the term. A diversity of research can be loosely aggregated under the AI umbrella, but only the more science-fictional facets of this science appear to excite public support: visions of humanoid conversationalists and robots provide a compact story that is more meaningful to non-experts than technical outlines of the intermediate machinery beneath the hopefully-intelligent surface. Nevertheless, residual disciplines which contribute to the engineering infrastructure that AI requires — but are agnostic as to the AI vision itself — have merits on their own and potentially avoid the problematic reductionism often found, I would argue, in the AI paradigm.

Whatever the paradigmatic forces and public perceptions in play, the technological infrastructure affording support for cognitive science/linguistics, or cognitive humanities/phenomenology, and so forth, is led by frameworks that seem to fall into one of two categories. On the one hand, there are "Cognitive AI" projects that realize different theoretical/proposed models of mental architecture as software systems, emulating our rational behaviors — Natural Language Processing, classification, judging similarity, optimizing problem-solving, and so on — to various degrees; simulative success or failure then becomes a litmus test for warranting or revising the implemented theory of intelligence. In the domain of linguistics, these projects serve as a paradigmatic complement (and sometimes testing-ground) for theories of how linguistic processes are computationally tractable.

On the other hand, there are technologies that fall under the rubric of "proof assistants", like Coq and Agda, which allow programmers to specify data structures with mathematical precision and potentially prove structural properties. Such technologies have computational models and type theories which overlap with general-purpose programming languages, like Haskell and Idris, and in that guise serve as linguistic research tools. That is, linguists have used tools like Coq directly or have achieved comparably rigorous analyses by developing linguistic models in rigorous languages like Haskell (representative are [5] and [42], respectively). These applications trade on the parable of language as *formal system*, and allow theories of *what* formal structures actually apply to language to be tested in environments well-established in other formal sciences, like mathematics and systems engineering.

In the specific context of linguistics, these two groups of technologies — AI platforms, and proof-assistants (or modeling frameworks in mathematically inspired programming languages) — reflect *computational* and *formal* analogies respectively: language *qua* computational

or *qua* formal system. This generalizes to other applications buttressing cognitive science: most concrete technology deployed in a research-oriented context projects either or both of these analogy/paradigms.

I believe that both analogies are flawed, although they may have value as indirect, theoretically productive models for *parts* of language and human intelligence. I think there are several theoretical frameworks, with at least partial realization in computational settings, that *do* avoid the reductive ideologies of both formalism and computationalism which yielding rigorous technological models: Barry Smith and colleagues' Granular Partition models [12], David Spivak's "Ologs" ("Ontology Logs") [82], Conceptual Space Markup Language [3], or the "Image-Schema Language" described in [4]. But in general there is a lack of adequate tooling which embraces *computational* representations of cognitive processes — for example, models of Intermediate Representations and Interface Theories — while simultaneously, mostly, rejecting both the formal and the computational metaphors. How this absence of "non-AI computational" projects may be explained, and what such projects might look like, is a question that will inform my discussions in the next section.

3 Channel-Algebraic Grammar

I earlier used computing procedures as loose metaphors for cognitive processes; my point was that both cognitive and computational processes can be hard to gloss logically because they are often dealing with incomplete logical information. More seriously — because *incomplete* logical information is not necessarily outside of logic; there are logical systems which can model information partiality in systematic ways — processes which occur in the midst of logically incomplete spaces (in the context of message-passing, function-routing, etc.) do not necessarily *operate* logically. That is, local rules for message-passing or function-routing systems do not seem to be (in their "native" semantics) logical rules. The truth-theoretic mapping of words to predicates (and so phrasal and sentence units to propositional structures) provides an obvious way to formalize linguistic structure by borrowing the analogous structuration from predicate complexes — except that such an attempt to find *direct* logic-to-language encoding usually fails, I have argued. Substituting a "procedural" semantic model allows a comparable formalization of linguistic structure through theories exploring procedural integration, for instance the interactions between computational procedures. Analysis of *computational* procedures can yield interesting ideas for linguistic theories of *cognitive* procedures — without endorsing a reductive metaphysics of cognitive procedures as "nothing but" computational procedures implemented in some sort of mental software.

In the current context, I want to consider in greater detail the interoperation *between* computational procedures, to consider how procedures form "networks" and whether this is a useful analogy for cognitive/linguistic processes.

3.1 *Link Grammar as the Syntax of Procedural-Network Semantics*

In Part I, I made an admittedly *philosophical* and speculative case for "interpretive mutual dependence" as a constituent building block of linguistic understanding. This theory will remain troublingly incomplete if the more philosophical presentation cannot be wed to a more rigorous formal methodology. True, an essential core of this theory is that interpretive "scripts" are largely prelinguistic and so not covered by linguistic formalisms in themselves. However, I have also argued that formal linguistic structures *do* govern how we identify which links apply to which word-pairs, and the general outlines of how word-pairing coordinates cognitive processes associated with single words — the fully contextualized synthesis of lexically triggered cognitive procedures may involve extra-linguistic grounding, but abstract prototypes of bidependnet relations are also prototypes of a synthesis between cognitive/interpretive functions. It would accordingly be reassuring if notions like "bidependency" and "mutual completion" could be employed as foundations for a formal theory of grammar and/or semantics with a degree of rigor comparable to, say, Link Grammar in its computational form, or type-Thoretic Semantics in the sense of Zhaohui Luo or James Pustejovsky. Such a theory — and potentially concrete technologies associated with it — would also then have a reasonable ground of comparison to the Semantic Web and, in the context of phenomenology, to the formalizing

influence which Semantic Web paradigms have exerted on projects to unite phenomenology with science and with Analytic Philosophy.

Given these considerations, I propose that formal grammars with the same underlying structure as Natural-Language Link Grammars can indeed be used as a foundation for type-theoretic and programming-language-design methodology. The key step here is to generalize Link Grammar's notion of a "connector" — the aspect of a word or lexeme that allows (or requires) completion via another word — to a generic data structure where connections can be made between different parts of a system on the basis of double potentials that must be in some sense "compatible" for the connection to be valid. One way to visualize such a system is via graph theory: imagine a form of graphs where nodes are annotated with "potentials" or "half-edges"; a complete edge is then a union of two half-edges. Half edges are also classified into different families, and there are rules governing when a half-edge of one family may link with a half-edge of another. In the case of Link Grammar, these classifications are based on surface language structure — head/dependent and left-to-right relations — from which a suite of links and connection rules are defined (for instance, abstractly a head/right word must link to a dependent/left word, a rule that then becomes manifest in specific syntactic rules, like how a verb links to its subject). For a more generic model, however, we can stipulate only that there is *some* classification of connectors governed by *some* linkage rules, to be specified in different details for different modeling domains.

Such a graph model expands upon the notion of *labeled* graphs, where edges are annotated with labels that characterize the kind of relation modeled via the edge itself. A canonical example is Semantic Web graphs: the edges in any Semantic Web structure are labeled with "predicates", defined in different Ontologies, specifying what sort of relation exists between its adjacent nodes. That is, in the Semantic Web, nodes are not "abstractly" linked but rather exhibit concrete relations: a person is a citizen of a country, two persons are married, and so forth. These structures are then concrete instances of Labeled Graphs as abstract mathematical structures. Based on Link Grammar, we can then refine this model be splitting labels into two parts, and allowing edges to be incomplete: a fully formed edge is possible when the label-parts on one side are compatible with the label-parts on another. One valid class of graph transforms is then a mapping where a graph is altered by unifying two half-edges into a complete edge, subject to the relevant linkage rules.

Another way of modeling this kind of structure is via edge-annotations and a rule for unifying two edges into an edge-annotation pairing. For sake of discussion, I will express this in terms of Directed Hypergraphs: assume that edges are "hyperedges", connecting *sets* of nodes. In Hypergraph theory, the nodes incident to a hyperedge are divided into a "head set" and "tail set"; these sts can then aggregate as "hypernodes". We can then define a kind of unification where the "tail hypernode" of one hyperedges joins with the "head hypernode" of another, producing a new hypredge whose head comes from the first former hyperdge and whose tail comes from the second. The merged hypernodes, in turn, form a new hypernode which "annotates" the new hyperedge (this new hypernode is not connected to the graph via other nodes, but is indirectly "part" of the graph through the hyperedge it annotates). *Annotated* hyperedges therefore differ from "non-annotated" hyperdges in that the former are the result of a merger between two of the latter. The rules governing when such merger is possible — and how to map a pair of hypernodes into a single "annotative" hypernode (which belongs to the graph through the aegis of its annotated hyperdge) are not internal to the graph theory, but presumed to be specified by the modeling environment where implementations of such graphs are technologically applied. Annotated Directed Hypergraphs are then "complete" in a sense if every "un-annotated" hyperedge has been subsumed into an annotated hyperedge, via a fusional process we can call a "annotative-fusional transform".

Extending this model further, we can say that the tail of an *un-annotated* hyperedge is a "tail pre-annotation", since it is poised to be merged into an annotation. Analogously, the head of an un-annotated hyperedge is a "head pre-annotation", and "annotative fusion" is the synthesis of a head and a tail pre-annotation (triggering a synthesis of their incident hyperedges). Correlate to annotative *fusion* we can define a notion of annotative *partiality*, referring to the "incompleteness" of pre-annotations which leaves room for their fusion.

It turns out that annotative fusion and partiality in this sense is a non-trivial model for computation in gen-

eral, and can be extended to a form of type theory and process calculus. The idea is that computational procedures can be modeled as hypergaphs (computer source code can certainly be modeled as hypergraphs which are productions of a certain class of parsing engines). Each "value" affected by a procedure — or more technically the source code symbols and memory addresses that "carry" a value — is then modeled as a hypernode that can link with other hypernodes in the scenario where one procedure calls a different one. Annotative fusion is then a phenomenon of values being transferred from one execution environment (associated with the caller procedure) to a second one (associated with the callee). The "annotations" themselves are then in this context the full set of type coercions, type checks, synchronization (e.g. resource locks or thread blocks depending on whether or not the caller waits for the callee to finish), and any other validations to ensure that the procedure call is appropriate. Annotative fusion also provides a formal basis for developing the intuition that "procedural networks" are rigorous representations of information spaces — annotative fusions capture the precise details of procedures linking (via caller/callee relations) with other procedures.

The constituent units of procedural networks are inter-procedure calls — but procedural networks also reveal dimensions of connectivity and and clustering characteristic of large, complex networks (and the graphs that represent them) in general. What appears as one function-call in source code can actually represent many different inter-procedure connections, a phenomenon reflecting "overloading" and "genericity" in programming language theory. Functions are generic in the sense that any one of their arguments can take multiple types — either because the function is explicitly declared to take a "typeclass" or a single type for that argument, or because an instance of a given type may actually be at runtime an instance of some subtype. The engine which actually implements inter-procedure calls — i.e., the programming-language implementation — needs to factor this genericity into runtime decisions, so a single expression in one function body can branch to many different called procedures. This is the essential core of the "semantics" of programming languages: data structures manipulated by computer code do not *intrinsically* represent real-world, non-digital phenomena, though they are engineered to model external data when used properly. However a code base does *internally* possess a space of implemented functions, and a symbol at one place in source code can match to some set of other functions so as to effect a procedure call. This "matching", and the rules governing how "overload resolution" occurs — "overload" meaning that a given notated procedure call can actually branch to multiple functions, so runtime information is needed to select the right one — are the essential formal principles governing the semantics of computer code.

From this basis, all the same, computer code can model a wide range of empirical phenomena. Generic code represents generic patterns of functional organization, allowing models to be built from varying layers of abstraction. From this perspective, to describe an empirical system it is necessary to identify important behaviors and functional patterns via which the system's observed behavior can be notated and/or simulated. To the degree that systems take on functional organizations that can be abstractly described, similar to the functional dynamics of other systems, their behavior can be represented and/or simulated via generic code. To the degree that there are particular details of a system's behavior that are more idiosyncratic to that system, and need to be modeled precisely, procedures can be crafted specifically for observing and emulating that exact behavior. More generic and more exact procedural implementations can coexist in a single code base, with generic functions calling granular functions narrowed to precise types, and vice-versa. The coexistence of generalization and specificity is an essential feature of code bases and, by extension, of procedural networks, ensuring the flexibility of these networks as tools to model information spaces.

Unfortunately, this kind of "procedural" modeling is hard to integrate with the more static techniques represented by the Semantic Web and the current "Big Data" fad. The latter paradigms tend to treat data as a static repository to be mined for patterns and insights, rather than a digital simulation or encoding of dynamic real-world systems. The Semantic Web, as a large, collaborative modeling project, evolved largely apart from the technological community concerned with computer simulations and the programming techniques which emerged from there, like Object-Orientation. This divergence is relevant for linguistics and cognitive science, because I would argue that the more "dynamic"

paradigm is actually more "Semantic" in a Natural Language sense. In other words, our cognitive dispositions when interpreting empirical phenomena — and matching these interpretations to linguistic cues — are more like procedural networks capturing functional patterns and layers of genericity in observed phenomena, rather than an accretion of static data. The techniques of procedural data modeling may therefore be relevant for Cognitive Linguistics and Cognitive Phenomenology because they aspire to something which, arguably, the mind does instinctively: build cognitive or computational models of dynamic, functionally organized phenomena.

As a corollary, the theoretical building-blocks of Procedural Data Modeling — how it leverages type theory, programming language semantics, and so forth — can provide at least analogs or case-studies for corresponding cognitive phenomena. Here I would argue that generalizing Link Grammar from Natural Language to formal languages, type systems, and lambda calculi yields added structures to type theory that are useful toward a more rigorous "theory" of Procedural Data Modeling — a theory of natural linguistics generalized to a theory of general data representation which, in turn, may offer insights onto the cognitive dynamics underlying (prelinguistic) situational/perceptual comportments and interpretation.

Type-theoretically, annotative partiality — which recall is my terminology for the abstract generalization of the mutual "incompleteness" in Link Grammar connectors, driving their link-fusion — extends conventional applied type theory (as in the Typed Lambda Calculus) in parallel to partial-labels extending labeled graph theory. It is paradigmatic in the theory of typed procedures and of "effect systems" that the type of a procedure is determined (up to certain equivalences that may discard overly granular type distinctions) by the types of all values affected by the procedure (including but not necessarily limited to the types of input and output parameters). We can then superimpose on this model an account of annotative partiality. Specifically, on the paradigm that procedure-calls are structurally represented as annotative fusions over Directed Hypergraphs, the values manipulated by a procedure are pre-annotations: they are not (in the *implementation* of a procedure, as a formal object) single values but rather typed spaces that can take on a spectrum of possible values depending on the inhabitants of their types. When a procedure is *called* these values become concretized, but as a formal system procedural networks model software in terms of its capabilities and expected behavior, rather than the state at any moment when the software is actually running. Partiality therefore models how procedures (as formal objects) deal with potentialities — we do not know what values will *actually* be present at runtime (e.g. what specific values passed to a procedure as arguments), so procedural analysis is essentially characterized by a partiality of information.

When one procedure calls another, the caller must build an *expression* — a gathering of values that provide all the information the callee requires — thereby creating a case of mutual-completion: the caller has values but not an algorithm to operate on them; the callee has an algorithm (that's what it implements) but needs concrete values so to produce concrete values. This dual partiality allows the caller to call the callee, via an *expression* which is part of the callee's implementation (represented as a hypergraph) which must in turn match the callee's signature — epigrammatically, we can say that "expressions are annotative-fusional duals of signatures". The point here is that whereas signatures are conventionally understood to be type-declarations assigning types to procedures, with annotative partially we can more precisely recognize signatures as stipulating *pre-annotative* types. The values carried within expressions also have pre-annotative types, but there is a distinction between types in the context of expressions and types in the context of signatures — and moreover this distinction is precisely the manifestation of the abstract head-pre-annotation and tail-pre-annotation contrast in the specific context of procedural networks. Just as signatures unify multiple types into one profile, we can analogously define "expression types" as the aggregate of all types from values affecting the expression — note that this is different from the type of the expression's calculated *result*, just as the type of a function is different from the type of its return value. Expression-types and signature-types are almost exact duals (the complication being default values for optional parameters, which are not directly represented in expressions — obviously, since then they would not be missing). The "duality" involved here derives from partitioning a type system into "expression annotative partials" and "signature annotative partials", a projection of head/tail duality in an abstract theory of Annotated Directed Hypergraphs (and analo-

gous to head/dependent and left/right partiality in Link Grammar).

3.2 Grouping Graph Edges

This section will briefly introduce what I call "Channel Algebra" and how it can lead to a theory (and practice, in a sense) of formal and natural-language grammar. Channel Algebra is discussed in greater detail in [20]. It is fairly divergent from other formalizations in computer science, though loosely descended from Process Algebras and from the "sigma calculus", which is a formal model of Object-Oriented programming [1]; [80]; [69]. Channel Algebra may also be seen as distantly related to Santanu Paul's "Source Code Algebra" [58] and to a network of discussions — not necessarily coalesced into technical publications — about how to unify Object Oriented and Functional Programming. There are many interesting analyses presented by scientists like Bartosz Milewski, on web forums such as Milewski's blog (the address is his full name as a dot-com domain). In general, though, I am developing Channel Algebra in an "experimental" manner, using a concrete software implementation in lieu of a technical or mathematical axiomatic description.

In the present context I want to focus on Channel Algebra as a potential theory in linguistics — particularly Cognitive Grammar — but initially I'll describe the underlying theory in a more computational manner. A lot of the Channel Algebra formalization carries between formal and natural languages.

A key notion in Channel Algebra is *procedures*. As in Part 1, we can think of procedures as either cognitive processes or as functions implemented in a software system, although for exposition the latter interpretation is simpler. So, assume we have a computing environment where many functions are available to be called — in effect, a bundle of software libraries each exposing some collection of function-implementations. For reasons I'll cover momentarily, I'll call these *ambient procedures*.

At one level, Channel Algebra is conceived as an alternative to data-sharing paradigms like the Semantic Web; so, one kind of analysis is concerned with cases where some body of information (which can be called a *data set*) needs to be transferred between two different computing environments. Channel Algebra takes the view that data does not have intrinsic semantics outside the computational environments where it is used. As I argued in the context of Searle's "Chinese Room", our identification that a software system represents facts — like someone's full name (the example I used over several paragraphs in the earlier discussion) — depends on the software possessing capabilities to display the information (usually visually). In other words, among the totality of all procedures that can be performed by the system, only a small set of procedures are involved in user-interactions where semantic intentions like "this piece of data represents someone's full name" are relevant.

As a consequence, when sharing data that includes information like *full-name*, we should not assume that the raw data, in its semantic interpretation, actually "means" *full-name*, or some kind of propositional assertion about full names. For example, a graph-edge in a Semantic Web resource intended to model the proposition "This person has full name Jane Doe" should not be seen as "meaning" anything about full names. Instead, it represents some computational artifact which *becomes* an assertion of that fact when a procedure is eventually called which converts the full name to a (usually visual) representation which a human user would recognize as a view on a full name (and hence on the proposition).

In sum, the Semantic Web (or any data sharing network) only *has* a semantics because software connected by the network has requisite procedures to make data-views that people can understand. Data does not have semantics (or at least not human-conceptual semantics); *views* do. This is consistent with an Interface Theory: most procedures manipulating Semantic Web data are part of an interface connecting networked data sources to the handful of procedures which create views for human users. Within the local structure of this interface, data does not have a "human" semantics; instead, it must be passed around between procedures before eventually reaching human-interaction procedures where (what we would call) the "real" procedures come into play.

When data is shared between localities, then, the procedures that will receive and manipulate this data are logically prior to the data itself and constrain when data-sharing is possible. Without the proper network of procedures, the data can never be transformed into the views where non-local semantics are relevant. This motivates my choosing the term "*ambient* procedures",

because a certain collection of procedures must be in place on the receiver end of a data-sharing event. This also implies that one important role of data modeling is to indicate which procures a potential receiver needs to have available — i.e., needs to implement — to qualify as a capable recipient of data conforming to the model. Data models should describe what procedural capabilities must be afforded by software libraries in order for the human-level, conceptual semantics of the data can actually emerge from humans' interactions with the system.

Analogous to Ontologies as data model specifications for the Semantic Web, I'll use the term "Ambient-Procedural Ontologies" to express the paradigm that implementing data-sharing protocols involves crafting software libraries around procedural requirements. This has two implications for how we theorize software systems. First, we need to characterize procedures in a manner that expresses the procedural capabilities that a system offers, or must have to satisfy a data-sharing protocol. Second, we can assume that whenever data is sent, received, manipulated, or visualized, there is a collection of procedures available in the system which enact these computational processes.

On this basis, then, I will develop a Type Theory that operationalizes this intuition about "Ambient Procedural Ontologies". The main outlines of this type theory are first that procedures have types; and second that procedures are "ambient" or logically prior to (or at least equiprimordial with) the type system itself. This is not a mathematical type system where every underlying type (like Natural Numbers) and every operation (like arithmetic operators) have to be mathematically described in the theory. Instead, we can always take certain types and procedures as "primitive" or (at least in their inner workings) external to the type theory. For any type system $\mathbb{T}$, whose structure is regulated by a type theory we intend to present or assess, we can say that $\mathbb{T}$ is built around a *kernel* $\mathcal{K}$ of "primitive" types and functions.

In general, an assumption of Channel Algebra is that a significant portion of information, present in some structured system, can be extracted by identifying elements in the system with types in a suitably developed type system $\mathbb{T}$. In the case of Natural Language — specifically Cognitive Grammar — this means that many syntactic and semantic details for each word in a sentence can be provided by mapping words to types. As I have mentioned, linguists like Luo and Pustejovsky have given persuasive analyses of certain type systems for *semantics*, but I intend to apply type theory also to *syntax*. Later in this section I will demonstrate this in practice, but for now I will just note that such analysis requires a sufficiently complex type system. For example, semantic notions like dot-types and dependent-product types, which have proven to be effective in shedding light on common lexico-semantic phenomena, may need to be expressible in a $\mathbb{T}$, *along with* link- or cognitive-grammatic notions like *connectors* and *expectations*.

Similar comments apply to modeling and sharing scientific data. Here, I have in mind projects like Conceptual Space Markup Language, and applications of Conceptual Spaces to study the nature and evolution of scientific theories, as reflected in research by (notably) Frank Zenker and Gregor Strle [29], [83]. Implicit in this research is the philosophy that scientific data models are not just electronic specifications for transmitting raw data, but embody scientific theories in terms of how data is structured and constrained. CSML, for example, defines criteria on data parameters such as ranges, dimensions (called "units" in CSML), and structural protocols (including CSML "scales") [3, page 6]. As a concrete example, the biomedical concept "blood pressure" is usually understood as a pair of numbers whose dimnsions are each in kilopascals (kPa) and whose first number (systolic pressure) is necessarily greater than the second (diastolic) — technically, the pair is *monotone decreasing*. In conventional Biomedical Ontology, cf. SNOMED-CT or the Vital Signs Ontology, these conditions might be stipulated by defining a "blood pressure measurement" concept subject to the relevant dimensional and range criteria (e.g. diastolic pressure must be greater than zero but less than systolic pressure), and/or by classifying systolic and diastolic pressure as subconcepts of blood pressure (see [76] and [33]). In a kind of Ontology paradigm incoporating type theory, the same conceptual structure can be represented concisely by defining a type whose structure conforms to the semantic requirements on the *blood pressure* concept as it is scientifically understood: i.e., a monotone-decreasing integer pair whose dimensional units are labeled "kPa" or "kilopascals". Note also that the "monotone decreasing" criterion is an example of the structure of dependent-product types (in

this case because the valid range for the second number depends on the value of the first), a construction which elsewhere is used for Natural Language semanticcs, e.g. [47, page 40] and [48].

The point of this example is that many scientific concepts — the semantic norms embedded in how scientific terms are defined and understood and how the concepts are used in scientific theories and research — can be rigorously specified with a suitably expressive type system. Therefore, a sophisticated system $\mathbb{T}$ is both a practical tool for scientific data sharing and scientific computing, but also an expository vehicle capturing theories' conceptual underpinnings. Developing scientifically useful type theories can then serve as a kind of formalized philosophy of science — type theory as metascientific analysis.

So in the past several paragraphs I have discussed the idea of using type attributions to represent semantic and syntactic details in natural language, and also metascientific concepts in scientific theories and data sharing. A suitably developed type-system can, in light of these possibilities, act as a kind of multi-purpose tool capturing semantic principles in a broad array of formal and informal contexts. This is possible insofar as type theory is developed on a flexible basis so that type systems can expand in different directions for different intellectual environments — dependent sum and product types for Natural Language semantics; "connectors" for Link Grammar; CSML-style units, ranges, and scales for scientific data; etc. The overarching goal of Channel Algebra is to enable these flexible, multi-purpose type systems.

Another way of expressing this idea is that type systems should be *multi-paradigm*. Suppose we are using a collection of types to model the semantics of some linguistic or scientific model. We may realize that there are some crucial semantic formulations that need to be recognized within the type system itself, if this type-oriented modeling is to be comprehensive. For example, in a link-grammar context, we may recognize that, in addition to assigning semantic and/or Part-of-Speech types to individual words, we need to represent the "potentialities" latent in words allowing them to link up (each word has a *connectors* such that a pair of "compatible" connectors can produce a "link"). Or, in a metascientific context, we may rcognize that we need to annotate type-attributions with CSML notions like range, scale, and units. The multi-paradigm criteria means that there would be a mechanism in place to enrich a type system $\mathbb{T}$ so that such semantic details can be seen as structural parts of types modeled via $\mathbb{T}$.[4]

One virtue of organizing a type system $\mathbb{T}$ around *procedures* is that this "multi-paradigm" flexibility becomes easier to achieve when we can directly model requirements on procedures, and how procedures interact with types (for instance, each type needs one or more procedures to construct elements of that type). As I said earlier in this section, we can start from any "kernel" of types and procedures and build new types by describing procedures which create, and/or operate on, values of $\mathcal{K}$-types. However, there are many different ways that procedures can act on values and call other procedures. This means there is a lot of room for extending type systems to represent different kinds of inter-procedural relations. For example, in Link Grammar a Part of Speech type is not just a "function" acting on other types — e.g., adjectives act on nouns; "red apple" modifies the image we have when we conceptualize "apple". The adjective/noun pair also needs the right potentials to create a connector. A type-thoretic model of this idea can involve stipulating that, in some cases, a function-type $\mathcal{F}_t$ is not only defined by the type of its argument(s), but by some added "connector" structure which must math between $\mathcal{F}_t$-instances and the instances of its argument types.

Channel Algebra, as I will now argue, gives us a way to encode "extra information" along these lines in type descriptions.

3.3 Channels as Type-System Extensions

In conventional type theory, every type system has *some* notion of functional types, but these are often treated in a simple form based on Typed Lambda Calculus. Canonically, a function *inputs* one or more values, and *outputs* on or more values (in many concrete type systems, only on output value). We can (still rather

[4] I refer to "semantic details" even in theories of Natural Language *syntax*, like Link Grammr, because many ideas about language *grammar* are relevant to the *semantics* of the linguistic *theory*, which is different than the Semantics of Natural Language which linguistic theories have as their subject matter.

informally) talk of these input and output collections as "channels". So, a procedure has an *input* channel — that's where it vgets its arguments from — and an *output* channel, where it provides a result. The term "channel" is usually used in the context of "process algebras" where procedures can run concurrently, and channels may be two-way means of communication between concurrent procedures. For this paper, I will only consider *sequential* procedures — where two procedures follow each other in time; no parallelism — and *one way* channels, which carry information from one procedure to another but have a fixed direction. Intuitively, if $\mathcal{P}_1$ sends a value to $\mathcal{P}_2$ via one channel, $\mathcal{P}_2$ cannot send a value back via the same channel (although it can *modify* the value in the channel).

While type theory may conceive functions as (in effect) a combination of input and output channels, actual programming practice reveals multiple distinct *kinds* of input and output channels. In Object-Oriented programming, Objects are passed to functions (i.e., "methods") using a different protocol than ordinary parameters. Also, many Object-Oriented languages support Exceptions, which are like non-standard return values that disrupt the normal program flow. This suggests two different *input* channels — I'll call them *lambda* and *sigma* for the calculi which can model their semantics — and two different *output* channeos, which I'll call *result* and *error*, for normal and exceptional return values, respectively.

Furthermore, many languages support "lambda" or "inline" functions which can be "closures", enabling a procedure to modify values in its "surrounding lexical scope". In other words, a *closure* is a procedure implemented inside the implementation of an enclosing procedure, and it has the ability to read and maybe modify data used by the enclosing procedure. We can consider this "handing down" of values from outer to inner procedures to be a kind of input channel, which I will call a "capture" channel. Some programming languages make this value-sharing explicit: in the case of modern-day C++ (since the 2011 standard) lambda functions have, as part of their definition, an explicit documentation of which symbols are "captured" from an enclosing lexical scope — these symbols are listed in square brackets, just as regular arguments are listed in parentheses, which helps reinforce the idea that inputs via symbol-capture are similar to inputs via argument-passing (which I say in Channel Alegebra as *lambda* and *capture* being both input channels).

Nested procedures can also be used to represent branching and control-flow, like *loop* and *if-then-else* formations. For instance, *if-then-else* can be represented as a structure which involves two nested procedures: if some condition (in the outer, enclosing procedure) is true, then the first nested procedure is called; if not, the second is called. Similarly, a *loop* takes some nested procedure and calls it repeatedly. In the simplest case, the nested procedure is called again and again until the nested procedure returns in a manner that signals the loop should be broken off (a common keyword for this condition, e.g. in C++, is "break").

To demonstrate with a trivial concrete example, we might have pseudo-code like this:

- int $x = 0$;
- loop {
- ⠀⠀⠀⠀if ($x > 10$) break;
- ⠀⠀⠀⠀$++x$;
- } // end the loop

If we want to analyze the inner code-block as a distinct (but nested) procedure, we would identify how the nested implementation "captures" the x value. We also have to identify how *break* causes the nested procedure to terminate — like *return* in an ordinary channel — but does so as a signal for *loop* (in the outer procedure) to break off. In effect, a nested procedure used as a loop block has a special kind of output channel which can represent two possible states: *continue* the loop or *break* the loop. I will call this hypothetical channel a *control* channel after Chung-Chieh Shan [73] and Oleg Kiselyov [40] (although they work in a different underlying context). Most programming language runtimes don't actually support the more "exotic" channels I discuss here, and without runtime implementations they can remain as just theoretical descriptions of programming phenomena implemented or analyzed via some theoretical framework quite different from what I am calling "Channel Algebra". However, the data set accompanying this text demonstrates a runtime engine where the channel structures I describe here can be directly implemented.

I'll also mention the Link Grammar example, say an adjective as a function modifying a noun. In a formal model, an adjective is therefore a kind of "procedure" which inputs a noun and outputs a noun. In my ter-

minology, a bread-and-butter input channel is called *lambda*, and a bread-and-butter output channel is called *return*. However, Link Grammar also recognizes various link kinds, each driven by connector-pairs. For example, consider a simple link-grammatic analysis of adjectives in the spirit of [74, page 16]. Note that most nouns take adjectives, but some words we might want to classify as nouns don't seem to:

- (37) Today is Tuesday.
- (38) The big departmental meeting is Tuesday.
- (39) It is very windy.
- (40) The lousy weather is very windy.
- (41) They are forecasting snow.
- (42) The latest reports are forecasting snow.

We might want to consider *today*, *it*, and *they* as de-facto nouns, but notice the adjectival constructions do not carry over: we can't say "big today", "lousy it", or "latest they".

In other words, some restriction on the "adjective" and "noun" types must be identified which blocks constructions like "big today" being parsed as valid examples of *big* as an adjective type-instance. One option might be to define the adjective and noun *types* more narrowly, which is more in the spirit of type-theoretic semantics. In that case, we don't take adjectives, say, as "functions" which input and output *any* noun, but rather model a suite of adjectival types operating on different noun types. For example, the adjectives *salaried* and *elected* in:

- (43) She is a salaried employee.
- (44) He is an elected official.

can only be attributed to persons, so qua functions these adjectives only "input" persons, as a subtype of nouns in general. Via Link Grammmar, on the other hand, the basic theory involves adding extra conditions on the adjectives and nouns involved, which are required (along with the underlying type-compatibility) to permit the function (the adjective) to input the argument (the noun). Then *today*, *it*, and *they* cannot take adjectives because they do not have the proper "connectors". Of course, both ideas can be combined, so we can define nouns and adjectives restricted to narrower subtypes (like person, living thing, physical object, etc.) and also marked with connectors, so adjective-noun pairings depend on compatibility both at the subtype level and the connector level.

The important point for the current context is that connectors essentially extend any type system compatible with Link Grammar, so we need to imagine an extra kind of input and output channel representing *connectors* as orthogonal to underlying Part of Speech or lexical types. In *big departmental meeting* we have to treat *departmental* as a kind of cognitive procedure modifying the concept *meeting*, and then *big* as a procedure modifying the "output" of the first procedure. Then we *also* have to represent a *connector* on *meeting* establishing that this concept/lexeme can be modified by an adjective, and similarly a related connector inheres to the *output* of the "departmental" procedure. So there is a kind of output-channel establishing which connectors are available on procedural outputs (connectors of the same varieties as would be available on individual words), and special input-channels which similarly model connectors which must be available on procedural inputs. I'll call these input-connector and output-connector channels. Type-thoretic semanticists like Luo and Pustejovsky might prefer to model connectors as aspects of types themselves, but we come closer to capturing the Link Grammar model if we represent the connector channels as distinct from the regular input and output channels, whose parameter-types are orthogonal to the language's connector-system.

I have, in any case, hereby presented eight different kinds of channels applicable to different programming- or natural-language constructions: *lambda*, *sigma*, *capture*, and *input-connector* on the input side; *return*, *error*, *control*, and *output-connector* on the output side (later I will introduce a few other channels). Notice that the range of different channels, each with distinct semantics and theoretical roles, extends far beyond the basic intuition that every procedure has some inputs and some outputs. Instead, type theories can evolve to capture theoretical structures in diverse domains, because many theoretical concepts can be systematically represented by describing special kinds of channels applicable to certain procedures. One way to capture data models and theoretical commitments is to envision scientific models as organized — implicitly or explicitly — around some sort of "procedures" and then analyze the various protocols by which information is mapped into and out of procedures, with

the *semantics* of these protocols described by stating requirements on specialized "channels".

The notion of *channel* is therefore closely tied to the notion of *procedure*, and *types* are similarly specified via both procedures and channels. That is, most of the complex types in a type system are functional types, which in turn are characterized in part by the kind of channels used by instantiating procedures. In principle, function-types are differentiated by the *kinds* of channels they use as well as the *types* of their arguments. For example, in Object-Oriented programming, a *method* in a string class might take a string *object* (representing a string of textual characters) as the nethod "receiver", aside from *arguments*: e.g., a "substring" method would take a pair of integers. This procedure would be considered to have a different type than an equivalent non-method function taking *three* arguments (one string and two integers) — even if both versions had the same number and types of input parameters. In C++, "pointer-to-member-function" types are never equated to function-pointer types.[5] The explanation for such distinctions in Channel Algebra is that a procedure which takes all inputs from a lambda channel has a different type than an (even if otherwise identical) procedure which takes one input from a *sigma* channel (equivalent to the C++ *this* keyword).

In Object-Oriented environments, non-static member-functions (aka "methods") are distinguished from non-methods in part because there are different rules for resolving method-calls when a given function name refers to several different procedural implementations (such as a base-class function and a derived-class function which overrides it). This is one example of how "sigma" and "lambda" channels have different semantics: the types carried within sigma channels are more consequential for overload-resolution than those in lambda channels. Conversely, sometimes channels are not semantically significant enough to decisively differentiate types. In C++, a function which *does* throw an exception cannot be overloaded with a function which *does not* throw an exception (assuming the rest of the functions' signatures are identical). This means that — if we model type systems like that of C++ via Channel Algebra — procedures with and without error channels can be the same type. But it is always possible *in principle* to differentiate function-types based on the kinds of channels their associated procedures use; just that sometimes this results in fine-grained distinctions not recognized by a particular type system.

The larger point is that function-types are specified via channels ("modulo" potential coarsening that equates types that could potentially be distinguished). Therefore, the fundamental fabric of the type system is dependent on modeling channels, because this determines how function-types themselves are modeled; and, as I will now explicate, function-types are the logical core of almost any practical type system.

3.4 Channels and Carriers

Most type systems take it as self-evident that any type is associated with *values* of that type, and usually *sets* of values. That is, intuitively, for any type (say, 16-bit signed integers) there is a set of values which are the "inhabitants" of that type. One weakness of this picture is that many types vary in terms of *what* set of values is actually representable in a given computational context. The case of 16-bit signed integers carries no such ambiguity; this type always has an extension equivalent to the interval -32768 to 32767 in $\mathbb{Z}$. However, consider the type of *lists* of 16-bit signed integers; depending on a program's available memory, some relatively long lists which can be represented on one computer will exceed the capacity of a different computer. Since computer code in general is not tied to a specific environment, we have to accept that for many types we do not know *a priori* what set of values conformant to that type can actually be used.

In addition to this practical problem, it is also difficult to describe exactly what a value *is*. Any instance of any type (at least in the software ecosystem) does have a numeric manifestation in computer memory — essentially an encoding as a sequence of bytes, that is, 8-bit unsigned integers — but it is not obvious that such encodings (sometimes called a "bit pattern") actually are "values" of types. In reality, computer code almost never deals with "values" per se, but rather deals with types represented symbolically in function signatures: passing "values" around from procedure to procedure.

[5] Note that C++ terminology is confusing in that pointer-to-member-functions can never point at *static* member functions (i.e., the fact that such functions are members has no bearing on their pointer-types), even though member functions themselves *can* be static.

One exception to this rule is that some values are written directly into computer code — the numeric literal "10" represents a specific value (usually in some integer type). However, even here some procedure is needed to interpret the character strings in source code to the actual typed value.

Given these considerations, while I will informally talk about "values" I try to minimize the use of "value" as a technical construct in Channel Algebra. With this foundation we can consider a kind of "value free" type theory by rough analogy to "point-free" topology (and therefore distantly to mereotopology). More central than the notion of *value* is the idea of a *carrier*. A carrier is a computational construct — symbolically represented by a source-code token — which represents an instance of a type. Each carrier is associated with one canonical type (though carriers may "hold values" associated with supertypes of its canonical type). When we talk about procedures calling other procedures, we really mean that the value of symbols in one body of code (where the calling procedure is implemented) is synchronized with the value of symbols in another stretch of code (where the callee is implemented). In short, at any stage of program execution, we can form groups of source-code symbols across the code base unified by the guarantee that these symbols carry "the same" value. But rather than talking of values directly we can take this synchronization between symbols as the deeper notion: the notion *value* is itself defined as the correlation exhibited by synchronized "symbols". The notion of "carrier" is then a more rigorous extension of the notion of source-code symbols or tokens.

The difference between a *carrier* and a *type* is that carriers have additional states. Some carriers are defind in function signatures, but others are introduced as lexical symbols in a procedure implementation. In many programming languages, lexical symbols can be *declared* before being *defined*. We can represent this scenario via a *carrier* which is in a particular *state* (I'll call it *preinitialized*). When a carrier is initialized, it takes on a state of holding a specific typed value — *values* are defined indirectly as characteristics of carriers in initialized states. At some point (consider a pointer to deleted memory) carriers no long hold meaningful values, and they enter a state I'll call *retired*. Introducing *preinitialized* and *retired* as carrier-states allows these to be separated from the type-system: we do not have to assume a "preinitialized" *value* which can be an *instance* of some types. In some cases, type systems *will* recognize values which play similar semantic roles to these carrier-states: for example, Haskell's "bottom" value is an instance of every type and represents a "null" or "missing" value. However, using semantics of carrier-states rather than type-instances means that we do not have to introduce extra structure to type systems which we may want to model via Channel Algebra.

Carriers which can be in preinitialized, retired, or initialized states I call *tropes* (there may be only one, or multiple, initialized states for tropes). A different class of carriers are called *emblems*, and represent abstract ("emblematic") specifications on carriers rather than carriers which hold concrete values: in effect, emblems are carriers present in function signatures. One or more carriers in an ordered list then form *channels* (though a channel can also be *empty*, with no carriers). I distinguish an *empty* channel, which exists but has no carriers, from a *vacant* channel which does not exist at all. For instance, a function that *can* throw exceptions but, at some point, has returned a normal value instead, has an *empty* error channel; a function which can *never* throw an exception (e.g. the C++ *nothrow* keyword) has a *vacant* error channel. Channels cannot include both tropes and emblems; those taking tropes are called *staged* channels, and those taking emblems are called *abstract* channels.

Carriers and channels interoperate according to several operators, which give Channel Algebra its "algebraic" character. To present these operators I'll also introduce the notion of *stages*. Briefly, any procedure is broken down into a sequence of stages, each of which involves constructing some aggregate of channels. The basic outline is as follows:

Carrier Append Any staged channel can append a trope, and any abstract channel can append an emblem (represented as $\mathcal{C} \oplus \mathfrak{c}$).

Channel Product Any collection of abstract channels can combine to a "product", called a *channel complex*. Similarly, any collection of staged channels can become a channel *package*. A channel complex or package is generically called a channel *product*. A channel product is considered "complex" if any of its channels are abstract. The basic channel-to-channel

operator $\mathcal{C}_1 \otimes \mathcal{C}_2$ represents a channel product formed by combining $\mathcal{C}_1$ and $\mathcal{C}_2$.

Carrier Handoff Given carriers $\mathfrak{c}_1$ and $\mathfrak{c}_2$, a *carrier handoff* $\mathfrak{c}_1 \looparrowright \mathfrak{c}_2$ means that the value carried by $\mathfrak{c}_1$ is (at last temporarily) carried to $\mathfrak{c}_2$. This means that there is some phase of program execution when $\mathfrak{c}_1$ and $\mathfrak{c}_2$ are synchronized, or "aligned"; exhibiting the same state (or sufficiently related) states. Alignment allows for imperfect handoffs, like coercing a floating-point number to an integer.

Digamma Reduction A channel *package* can be *allied with* a channel *complex* (written $\mathcal{C}_p \circledcirc \mathcal{C}_x$) if carriers in $\mathcal{C}_p$ and $\mathcal{C}_x$ are "alignable". The actual description of such alignment is non-trivial; in [20] I address this in terms of hypergraph models of computer code. But in general alignment means that handoffs are possible between tropes and emblems, and if there is enough alignment the channel *package* $\mathcal{C}_p$ can be interpreted as a call to a *procedure*, whose *signature* is modeled by the channel *complex* $\mathcal{C}_x$. In this case we have a "Digamma Reduction" operator $\varsigma \Psi \mathcal{L}$, meaning that the package $\mathcal{C}_p$ is "evaluated" and then control passes to the stage labeled $\mathcal{L}$.[6]

Each of these operators represent a step in a computational process whereby channel packages are constructed and then evaluated, which over the course of several stages (in general) provides implementation of procedures. A single operator is associated with a *microstage*, for example, appending one carrier to one channel. A sequence of microstages is than an *intermediate representation* used to translate high-level source code to data structures that can be executed directly (or more or less directly, by interfacing to "ambient procedures", e.g. via C++ reflection). This strategy is concretely put into practice in the sample code distributed with this paper. In effect, high-level source code is "compiled" into an intermediate representation, and this IR is a more or less direct translation of Channel Algebra operations.

One of the technical challenges of using this strategy for practical software developments involves mapping sophisticated Channel semantics to ordinary type systems. Since a language like C++ does not support the full range of channels I have presented here, the more detailed channel-complexes have to be mapped to function-call semantics which C++ will actually recognize. In practice, a lot of this work involves manipulating function pointers and casting carriers to generic binary representations, like arrays of void*-pointers. These techniques are not especially relevant to the philosophical issues I am focused on in this paper, so I'll leave them to the published data set (interested readers will find a relatively complete C++ development environment, intended to be used with the Qt platform, that operationalizes the theory I am developing in this section).

At this point however I *will* comment on the overall architecture of building and then evaluating channel packages. Supporting "Digamma Reduction" in a runtime environment requires reasonably complex C++ libraries (assuming C++ is the language of the runtime itself); but most of the Intermediate Representation is concerned with asserting carrier properties and then adding carriers to channels. Channels themselves act as stack machines, in that they can be built up like machine stacks and then cleared in the passage from one stage to the next. This is an example of a point I made much earlier, in the introduction, that computing environments ultimately involve stack machines at their most primitive level.

We are, in any case, operating here on several different semantic levels. The Channel Algebra operators define one, intermediate level whose main theme is building channel packages. At a "lower" level each package must be evaluated at runtime, for instance by converting it to a C++ function-call, so the semantic issues there are identifying the proper C++ function (or function-pointer) and marshaling the channels to mimic the C++ ABI (Application Binary Interface). Conversely, at a higher level, source-code formations are interpreted in terms of the channel products they imply: so in code like $x = y \to f(z)$ the implicit channels are likely populated as follows: z goes into f's *lambda* channel, y into its *sigma*, and x becomes bound to its *return* (of course whether this is the actual meaning of the code depends on the high-level language's formal grammar). This implies that grammars can be organized around the channel structures indicated by conformant code. To the degree that grammar engines adopt this paradigm they can be called *Channel Algebraic Grammars*.

[6] The motivation for the term "digamma" is first that the "sigma" in ς Calculus looks like a smaller version of the Greek letter Digamma, and second that "gamma" is often used to represent graphs, and "digamma reduction" can be sen as a relationship between two different source-code graphs.

This paper's data set includes one example of an implemented Channel Algebraic Grammar insofar as code written in a special high-level language is translated to a Channel Algebraic Intermediate Representation (and then evaluated). The syntax and semantics of this special-purpose language is, of course, only distantly related to *natural* language; I won't address whether a comparable architecture could yield a workable Natural Language Processing engine. However, the design and role of this Channel Algebraic Grammar in the *Software Language Engineering* problem-space can perhaps serve as at least an analogy for how *natural* language orchestrates the "flow" of information, and the ordering of operations, across the cognitive procedures which underly linguistic understanding. I will return to this possibility at the end of the section.

3.5 *Channels and Constructors*

As I indicated, Channel Algebra downplays the notion of typed *values* except in a derivative sense (e.g. a "handoff" means an alignment between carriers which we can picture as a value being copied). However, carriers do become initialized, and even if this initialization results from a handoff originating with another carrier, that carrier in turn had to be initialized. At some point these chains of initialization have to be grounded on some underlying data. This data may come from outside the software itself: for example, a program emulating a calculator relies on human users to type numbers on a keyboard or by pushing buttons on a User Interface. Other times values are literally written in computer code, or obtained from files, databases, or over a network.

For sake of discussion, we can limit attention to values read literally from source code. Even external data tends to depend on some numeric or string literal: reading data from a file requires specifying a file name, and obtaining data from a network location requires a URL. So computer code itself provides the primordial values from which other values circulating through the software are derived.

In a typical type system, then, at least some types should have procedures which construct values directly from source code literals. I'll call these *literal-constructors*.[7] In addition to literal-constructors, there are some procedures to create type values from other values (or from no values at all); I will call these *co-constructors*.[8]

There is no strict rule separating constructors for a type $\mathfrak{t}$ from other procedures that may return values of $\mathfrak{t}$. Intuitively, they play different roles: constructors are about creating new values, while other functions analyze or modify existing values. The canonical examples are so-called *trivial* constructors, which take no arguments — for instance, the constructor for a list of numbers returns an empty list. Such a function surely qualifies as a constructor. On the other hand, consider a function which takes a list of numbers and returns a new list with duplicate numbers removed — such a procedure acts more as a utility function and probably would not be classified as a constructor.

The constructor/non-constructor distinction is anyhow typically left open by the programming language environment, so coders have to signal their intention to treat a given function as a constructor by some extra syntax.[9] In Channel Algebra, one option is to define a special " *construct*" channel for values output from constructors.[10] Consider the case of appending values to a list of numbers: should a function which maps $\mathcal{L} \ll \mathcal{X}$ to $\mathcal{L}'$ — where $\mathcal{L}'$ is the same as $\mathcal{L}$ except it has x at the end — be considered a constructor? In most functional programming languages this is actually a classical case (along with trivial ones) of constructors, because they provide the basic mechanism wherein instances of list

[7] In C++, similar constructor-functions are called *literal operators*. This terminology only applies to "user-defined" constructors; the compiler itself handles initialization for built-in types like integer and floats. However the overarching term "literal-constructors" is helpful for theory-focused language-agnostic discussion about types and constructors.

[8] This terminology has the benefit of distinguishing the property of functions I call *co-constructors* from what existing programming languages call "constructors", which itself has different meanings for different languages. For example, in C++ one cannot take the address of a constructor function; however, co-constructors are implemented such that you *can* have co-constructor pointers, which is essential to the idea of "preconstructors" that I will address below.

[9] E.g. in C++ a constructor is given the same name as the type it constructs for. One consequence is that two different constructors cannot be declared without overload-resolution: each constructor has to have a different signature than any other constructor of the same type. However, the rationale for this restriction seems to be syntactic more than semantic.

[10] It is useful to pair this channel with a semantically similar channel I'll call *placement* (after "placment new" in C++) for scenarios where the constructed result should be written to a buffer provided by the calling procedure rather than returned.

types are defined. In other words, this *append* function seems logically anterior to other functions which may create a list — for instance, by *removing* the last element from a (non-empty) list.

Such a notion of "anteriority" can sometimes be made rigorous. If we build up a list of numbers by appending values to smaller lists, we can eventually construct any list whatsoever. We can also run this process in reverse: for non-empty $\mathcal{L}'$ there is only one way to construct $\mathcal{L}'$ from a pair $\mathcal{L} \ll \mathcal{X}$. In other words, an *append* constructor is "reversible". Moreover, we can repeatedly reverse constructors like these to get shorter and shorter (and eventually empty) lists. This means that algorithms can traverse a chain of constructors "backward" and will be guaranteed to terminate. For example, to determine if a list of numbers contains the number n, it is easy to check if it *ends* with n. If not, "reverse" the construction, and see if each smaller list ends with n. Eventually we would reach the empty list, meaning that the original list did *not* have n.

The possibility of "reversing" constructors is a familiar pattern in functional programming, where it is often called "pattern matching". It allows algorithms to be implemented in a functional style, with heavy use of recursive functions and sparing use of side-effects and mutable state. On the other hand, pattern matching relies on some simplifying assumptions that may not be consistent with all type systems. For example, C++ is laxer about how values are constructed; I can obtain a list of numbers by dereferencing a pointer to a list, with no information about the provenance of the referenced data. Data types in C++ do not usually carry around details about their "history", and it is not always easy to reconstruct that history the way we can "obviously" reverse the construction of a list.

In any case, these variations present some potentially informative characteristics about individual types. Given type $\mathfrak{t}$, we can ask questions like:

1. Which instances of $\mathfrak{t}$, if any, can be the result of a trivial constructor?
2. Does $\mathfrak{t}$ have a *default value* whish is the result of a trivial constructor?
3. Which instances of $\mathfrak{t}$ can be the result of a literal constructor?
4. Which instances of $\mathfrak{t}$, if any, can be the result of a reversible constructor?
5. If we have values which *are* the result of a reversible constructor, is there an efficient way to "un-construct" the value to support pattern matching?
6. Does $\mathfrak{t}$ have co-constructors (i.e., they are not literal) which also are neither trivial nor reversible?

Notice that a trivial constructor does not necessarily produce a default value. For example, a type meant to represent days of the week could default to whichever day is current when a constructor is called: if an application is run on Tuesday, the day-of-week trivial constructor would return the value for Tuesday. So a type may have *more than one* trivial-constructed values. But if a type has *exactly one* trivial-constructed value, this is *usually* a "default" value.

However, a type can have a default value without a trivial constructor. A default value plays the conceptual role of a "fallback": 0 is a reasonable default for most numeric types. However, some numeric types don't have trivial constructors at all. Meanwhile, types representing calendar dates sometimes default to a standardized "day zero", like January 1, 1970. But a trivial constructor for such a type may instead return today's date (when the function is called). When a type has a default value, a (co-)constructor which returns that value can be called a *default* constructor.[11] As the Calendar Date example shows, types can have default and trivial constructors that return different values.

Sometimes (co-)constructors are significant even if they are not actually used. To demonstrate how this may occur, consider the problem of enforcing dependent-type constraints without using dependent types explicitly. A canonical example is a function which must take a (maybe monotone) increasing or decreasing pair of numbers (I used the monotone-decreasing example for systolic and diastolic blood pressure). Suppose we implement a procedure to highlight a selection of characters in a screen display, whose inputs are the start and end character indices in some displayed text. We want to indicate that the second number (call it y) must be greater

[11] For an example of a non-trivial default constructor, consider a literal constructor for a ratio type that returns 0/1 given a malformed character string.

than the first (x). Via unrestricted dependent types, we could just define the type of y as "numbers greater than x"). The problem with this type-declaration is that we therefore do not know what type y has until the function is called (and x has a value). This could violate the principle that every argument to a function must have a type which is known in advance. Alternatively, we can say that y's type "provisionally" is, say, a 32-bit integer the same as x, but it's "real" type once x is known is a dependent type that depends on x's value. Not every type system however allows for this kind of distinction between "real" and "provisional" types.

One way to assert the dependent-type restriction while avoiding these problems is to note that x and y must form a monotone-increasing pair. That is, the procedure mandates the *possibility* to create the monotone-increasing pair xy from x and y, even though the procedure does not use this pair-value directly. As a construct in Channel Algebra, we can introduce the idea of a *rider* channel which asserts the *possibility* of creating certain values without actually creating them The simplest case is suggested by this number-pair example. Let's assume we have an implemented type modeling monotone-increasing pairs, with a co-constructor that creates such a pair from two numbers (verifying that the two numbers are indeed monotone-increasing). We can then form a pointer to this co-constructor function, or obtain some other unique identifier for it. I call a function-pointer or similar value a *preconstructor* if it references a co-constructor. In addition to being an indirect way of calling the co-constructor, a preconstructor serves as a *certification* that the co-constructor *could* be called. For example, one way to indicate that y has been checked and is indeed greater that x is to use the monotone-increasing pair preconstructor as a kind of signal value: test for $y > x$, but if so, instead of using boolean *true* for an affirmative, use the preconstructor.

The key point here is that many conceptual details or programming requirements that concern interrelationships between values can be modeled within a type system — even without full-fledged dependent types — via preconstructors used as signal values. For a given requirement, such as "is y greater than x?", we can identify a type that *could* be constructed if (and only if) the requirement is satisfied. For instance, *y is greater than x* is true iff we can form an instance of a monotone-increasing pair type out of x and y — and so, the preconstructor for this type becomes a convenient signaling value for the affirmation of $y > x$. Conceptual requirements can be defined by modeling types whose values necessarily exhibit some significant property, and then using preconstructors to those types as certificates that the property is true (in some context). A " *rider*" channel can then be populated with one (or possibly several) preconstructors affirming facts about the values carried in other channels. For instance, a *lambda* channel holding x and y can be paired with a *rider* channel holding a monotone-increasing-pair preconstructor, certifying that y has been checked to be greater than x.

The larger point of this whole subsection is that details about the nature and existence of types' constructors can provide useful conceptual information about types themselves. Often this information dovetails with the metadata pertinent to Conceptual Space Theory and CSML. Consider a simple but representative example of an object exhibiting core Conceptual Space "quality dimensions": a colored rectangle in a computer graphics environment. This object could have some 13 dimensions: colors for the shape's interior and border (including 3-dimensional color vectors plus transparency factors); the top-left position, width, and height of the rectangle; and the width of the border. A software engineer has numerous constructor options for this type of object: should there be one "pod-tuple" constructor that takes all 13 values in a flat list?[12] Should there be distinct data types for 3-vector colors or 4-vector colors (including transparency)? Should the top-left point be merged into one 2-vector point type? The rectangle-constructor could potentially have its 13 fields folded into 5: two color 4-vectors; one top-left point; width; height; and border-width. Or border width and color can be merged into one "border" type. On top of that there is the question of default values: should pure black, or maybe pure white, be a default color for the interior? Should the border default to width-one black, or width-zero (i.e., no border), or something else? However the types are designed, there is a mutual dependency between simpler types like colors and points and the constructors for complex objects like rectangles: since complex objects are built from simpler ones, a type interface should be designed with consideration of how to assemble the whole

[12] "POD" is a common coder's parlance for "plain old data", and it generally describes data structures conformant to *struct*s in the C language, rather than C++ classes. A typical feature of *struct*s is that they are initialized by listing all of their fields in one tuple.

from the parts, or retrieve the parts from the whole.

Some compound objects are conceptually analogous to basic pairs or tuples of values: 2D graphics points are 2-vectors, say, and solid colors are RGB 3-vectors. One way to signal that a compound type is like a tuple "product" of its component fields is to allow instancs of that type to be constructed just by listing each field separately. In C++, this is often implemented by defining a constructor which takes an "initializer list". Analogously, in Channel Algebra we can define a *pod-tuple* channel that serves as the *lambda* channel for pod-tuple co-constructors. A *pod-tuple* channel indicates that a given type behaves essentially like a tuple, with a sequence of values that, in most cases, vary independently of one another. These kinds of types hew closely to the Conceptual Space idea of multi-dimensional quality spaces. However, other kinds of types have more complex, interdependent internal structuration. I think Channel Algebra is a way to make Conceptual Space theory relevant for these more complex types also, using different Channel Semantics and the modeling roles of different forms of types' constructors; how these roles reveal types' conceptual foundations.

Having argued for Channel Algebra as a formalization potentially relevant to Conceptual Space models, I want to conclude this section by considering how Channel Algebra may be intuitively applicable to Cognitive Grammar.

3.6 Rider Channels and Deferred Coercions

Earlier in this section I described how Channel Algebra can represent dependent types and coercions, both of which are essential ingredients in type-theoretic semantics. I am not proposing Channel Algebra as a *literal* theory for natural language, particularly in its implemented form as a runtime and parsing environment for programming languages. However, some of its formal details for processing programming languages could be at last suggestive of procedural patterns factoring in to natural-language understanding.

Before going into detail, I think it's worth distinguishing different aspects of natural language where type theory may play a role. On one level, very general Part of Speech classifications can be approached type-theoretically. This helps reinforce the intuitive idea that syntactic categories like verbs, adjectives, and adverbs corresponds to conceptual, cognitive processes which effectuate changes in beliefs, conceptualizations, or situational records. A noun by itself is an abstract concept; during the course of a sentence, it is subject to concretization and predicatization, ending up in an idea that has some degree of concrete propositional content.

For example, we might go from *dog* to *dogs* to *those dogs* to the sentence *Those dogs are barking*. Logically we can see this as building up a propositional structure from sub-propositional parts. But cognitively the process is more like migrating from a cognitive register dealing in conceptual abstractions (like "dog") to a register for propositional attitudes and situational understanding. Logical constituents are more like stages in a concept-to-belief evolution than mereological units. That is, the relation between the signifying whole and its semantic or morphosyntactic parts is more the relationship between a process's ending to its intermediaries than a mereological hierarchy. If this picture seems compelling it can be conveyed by figuring the majority of Part of Speech types as function-like: the nature of verbs, for example, is to functionally transform (in our cognitive frames) nouns to propositions. Analogously, adjectives transform nouns to other nouns — in the sense that "red apples", say, can substitute as a noun-concept in most places that "apples" alone can; *red* being like a procedure that produces a modified *apples* concept which substitute for just *apples* in subsequent processes. Likewise, adverbs produce new verbs from other verbs; subordinators like *that* transform propositions back to nouns (as I argued in the last section), and so forth. So Parts of Speech can be usefully analyzed as "function-like" types (I will discuss specific language examples arguing for this perspective in the next section). These are, however, very general types, which would be the "uppermost" types in a natural language semantics; I will call these *macrotypes*.

On the other hand, at a much finer level, individual lexical "cliques"[13] — the concept or extension or intensional property-set associated with particular word-meanings — often seem to behave as *types* as well. In "Formal Concept Analysis", *concepts* are (statistically) defined as a combination of extensional and intensional

[13] Borrowing this term from [28] though I'm not using it exactly the same way.

criteria: neither extension nor attributes alone fully characterizes a concept, either cognitively or extra-mentally. In particular, practical fluency in a concept involves knowing canonical examples (like having seen many of the medium-size, traditionally architected dwellings that would be prototypical *houses*) and also the prototypical features that characterize the concept (houses are three-dimensional, enclosed, divided into rooms, function as a place of residence, etc.). A combination of exemplars and attribute-prototypes define the "core" or "center" of a concept: insofar as some example is "peripheral", an outlier which is somehow not representative of the concept (but still can be classified to it), it must have some *featural* difference from exemplars; but we can also compare it as an individual to more exemplary individuals. That is, we understand outliers both intensionally — we can articulate the features which make both a log cabin and a gated mansion atypical as houses — and extensionally: we can mentally juxtapose cabins and mansions with ordinary houses.

If this gloss on the nature of concepts is accurate, it helps explain why the lexical entrenchment of concepts often has a type-theoretic feel. In the more logical aspects of semantics, lexicalized concepts do present operationally as types:

- (45) A penguin is a flightless bird.
- (46) Rhinos in that park are threatened by poachers.
- (47) Baby elephants don't have tusks.

The point of these examples is that their signifying structures involve operations that can be explained type-theoretically. In (45), a concept is characterized via a subtype/supertype relation. In (46), a concept is being "extensionally filtered": we start with an expression that seems to designate the extension of some concept (*rhinos*) and then narrow it based on extensional criteria (*in that park*). This suggests a type/set interface similar to my analysis in Part I of "students polled". And (47) suggests an *intensional* filter, narrowing a type for contextual purposes into a subtype that is conceptually meaningful but not entrenched: there is no common word (peer to "kitten" or "puppy") for *baby elephant*.

All of these operations — fine-tuning concepts to the specific ideas salient in a specific signifying act — have plausible representations based on a concepts-as-types analogy, and suggest another level where type theory could be an effective formalizing tool. But in this case, we are discussing the almost minimal groupings of a linguistic hierarchy, individual word-senses — I'll call these *microtypes*.

Meanwhile, in between coarse Part of Speech types and fine lexical "cliques" are the kinds of Ontologically-characterized categories or "lexical sorts" [53] associated with linguists working in a formal type-thoretic vein, like Zhaohui Luo and Bruno Mery. I will call types in this vein *mesotypes*, being intermediate in generality between individual lexemes and syntactic categories (Parts of Speech and their refinements, like plural nouns). The broad philosophical implications of mesotypes is that they give some formal traction to what happens cognitive-procedurally when we negotiate between different word senses and/or use language in seemingly metaphoric (but not unrestricted) ways.

Consider these exmples:

- (48) My favorite Korean restaurant is adjacent to the bookstore.
- (49) My favorite Korean restaurant is closed today.
- (50) My favorite Korean restaurant is decorated with posters from the 2002 World Cup.
- (51) My favorite Korean restaurant started out in a food court.
- (52) My favorite Korean restaurant said I'm their most loyal customer.
- (53) She's constantly barking at her employees.
- (54) He's not playing well because his back is barking at him.
- (55) He's not playing well because his sore back is acting up.
- (56) The kitten barked at the dog.

The first three profile (using Langacker's term) a restaurant as a location, a building (with commercial and architectural properties), or an institution/organization. Note that we can distinguish the commercial/architectural sense from the institutional sense: to *close* can be temporary (as in 49), which we hear in conjunction with the former sense, or more permanent as in "cease operations", i.e., close "qua institution". The former sense seems to mix the institutional and location sense: while location is not explicit as in (48), an assertion that some *chain* restaurant is closed today would usually be heard as referring to one location, not the entire chain. In (53) through (56), a common idiom applies "bark" to

things that are not dogs; but this is not just a matter of metaphor, blending different "spaces" with no conventionalizing pressues. The (53) and (54) senses are rather entrenched, using *bark* to suggest something a little obnoxious or mean. But (56) does not sound right, even though it is no more of a stretch to imagine a kitten's hissing as angry bark-like than a boss's orders. Perhaps the *literal* similarity between cats and dogs makes the figurative usage harder to accept as a purported abstraction from literal situations.

The conventional type-semantic picture is that language has a collection of (what I am calling) "mesotypes" and we modulate word-uses by switching between this medium-grain types, sometimes in unexpected ways. An effective way to develop such an analysis is to identify one or two senses as the most prototypical for some concept/lexeme, and read other senses as transforms involving other types. So the commercial/architectural mesotype may be canonical for *restaurant*, and then we branch out to a more geo-spatial sense (*where* the building is located), or institutional (the food-court incarnation may have been in a different place with different staff and business classification). Hence a *commercial building* type is "cast" to a geospatial point or to a social institution. These "casts" are not really metaphors; they have conventions, limitations, and rules.

The mainstream theory is simplest when analyzing casts between mesotypes of similar generality, like plant-to-sentient in "flowers like water" or cerebral-to-physical in "heavy thoughts". But not all casts follow this recipe: (53) and (54) seems to cast from a *microtype* to a *mesotype*. The construction in (55) is a common "personification" of medical/anatomical concepts — a similarly scalar idiom is cases like *my arthritis is acting up*, or *flaring up* to render figure ailment as physical rather than personalistic — but here we care operating between types of comparable Ontological granularity (anatomical part/medical ailment to person/physical thing) But semantically and syntactically (55) seems a lot like (53), even though the ploy in that case is to compare things to a lexical "clique" (dogs). And in (52) we infer that some *person* complimented the speaker, but this is more of a synecdoche than a cross-type cast like (51)-(51): the point is not that a restaurant has a personhood "aspect" that can be highlighted (the way it *does* have an architectural or geo-spatial aspect) but that reference to the restaurant can be proxy for people closely associated with it.

Another complication for type-cast theories is potential ambiguity between word-senses and coercing "usages", as in:

- (57) Tea-smoked duck is a Sichuan delicacy.
- (58) The baby's favorite toy is a rubber duck.
- (59) Wedding ducks are traditional Korean gifts.

If the prototypical duck sense is an animal, these cases disrupt the basic type hierarchy: wedding ducks are made of wood; tea-smoked duck is a kind of food; rubber ducks are toys (is the phrase descriptive — ducks which happen to be rubber — or a distinct noun-concept, like *decoy duck*?). This translation from one branch of sub/supertype relations to another is what we would expect of type-cases, but we can also see these cases as distinct lexemes, or word-senses, or entrenched phrases with *de facto* lexical status. The overall picture which seems to emerge is that type-coercions remain relatively metaphorical or metonymic in cases like (53) or (52); become idiomatically entrenched in cases like (48) and (51) — especially where there is a similar scale of granularity between mesotypes in a coercion — and can become *lexically* entrenched in cases like (58)-(59).

Let's assume in any case that we can give a thorough analysis of type-coercions as semantic devices. This is still working on the level of generalized lexical use-patterns; we are not studying the *cognitive* steps involved in actually making whatever imagistic, situational, or conceptual modifications are directed by a type-cast. In other words, there is a cognitive process of thinking of a restaurant (say) and then revising this framing to accommodate senses like spatial points or social institutions. We can weave different senses and coercions into one sentence:

- (60) This book, which costs $40, has some crazy ideas inside.
- (61) This book, which the library classifies as young adult nonfiction, has some dude's phone number scrawled inside.
- (62) My boss barks louder than his dog does.

In short, we can't assume that type-coercions simply replace an image forged according to one micro- or meso-type with one shaped by an alternative type. Instead, signifying elements seem to carry a package of type attributions around, and different types are "activated"

to different extents, and in different ways, at different stages of linguistic processing.

Another complication is that types of different scales seem to coexist and yet in other respects macrotypes, mesotypes, and microtypes seem to be distinct hierarchies. A microtype like *dog* acts sometimes as a subtype for a mesotype like *animal*, but also mesotype concepts like *animal* and *person* are sometimes microtypes themselves: after all, these are lexical entries as well as Ontological categories. So on the one hand we may say that words have three different types — even excluding cross-mesotype or cross-microtype ("bosses barking" cases) casts — and that macro-, meso-, and microtypes are distinct thoretical posits with different analytic mthodology; but sometimes the boundaries between micro- and meso, and their analyses, seem to blur. Analogous comments can apply to the meso/macro boundary as evinced by issues like classifying forms of plurality, as in [52]'s analysis of noun plurals.

This metathoretic hemming and hawing may be acceptable — even desirable — at the philosophical level. But it causes problems if we want to see type theory as a formalizing (albeit simplifying) window onto cognitive-linguistic processes. In a computational context values have one canonical type; they can be cast to other types but do not maintain a kind of superposition "history" over multiple types. These kinds of phenomena can be technically modeled in various ways (e.g. Luo's "dot-product" types), but in that case we want a formalization that seems appropriate for how the corresponding cognitive procedures might unfold.

In the case of Channel Algebra, I mentioned the idea of "rider" channels which supplement the type information contained in other channels, adding more information without actually introducing new parameters into a procedure. So a rider asserting that y can be made into an xy pair is semantically analogous to giving y the type *integer greater than* x, but y itself is not presented as having a dependent type. In short, *greater than* x is not presented as an *alternative* type for y, nor is $y > x$ presented as an alternative *value* as if we are "casting" y from a wider to narrower type. However, we are asserting properties of y by certifying that such casts *would* be possible.

I think this may be a useful analogy to how type-casts work in practice in Natural Languages, especially in complex cases where multiple word-senses are involved. Not every type associated with a word makes sense in every conceptualization. When, for instance, we attend to the conceptual properties of a book as an intellectual object, there may be physical details which are conceptually incompatible with this attitude. It would be simpler if there were a neat partition among senses, and some overarching cognitive order — Ok, now we're figuring the book as physical; now it's an intellectual artifact; now it's a commodity — supervises the coercions back and forth. Neither semantic nor syntactic evidence warrants this simpler picture: it seems more as if conceptualization across types is a matter of networked cognitive procedures taking turns operating on one conceptual package, where latent type attributions are available to each procedure whether or not they are "usable" in the sense of involving type structures that logically fit each procedure's purpose.

I will give concrete cases of word-senses that I think substantiate this picture in the next section. Here, though, I'll conclude as follows: Channel Algebra can perhaps model what is going on insofar as procedures can take "rider" channels that can add detail to procedural inputs. In a cognitive-linguistic setting, we can imagine these riders as multifaceted and complex. Because riders are not part of procedures' actual inputs, they do not necessarily need to use types that the procedure recognizes or knows about; the riders may only come into play as values are passed among procedures. The analogous computational case would be some extra data associated with a value that is only relevant for certain security-oriented validations; i.e., permissions to modify a file. Functions can pass around such values without considering the security-related data, except for a few procedures where security-sensitive operations are attempted. So the security details are "part" of the data carried by a value, but only become semantically salient parts in certain procedural contexts. Rider channels are a way to represent this kind of extra information within a type system directly. I think they are a plausible analogy for cognitive re-inscriptions that occur in the evolving significations in a sentence.

4 Conclusion

Without reducing linguistic *performance* to language qua field of propositional expression, and without collapsing linguistic meaning to a computable/propositional fragment, we can still allow interpretive-phenomenological and formal/mathematical perspectives to co-exist. In the theory I have sketched, Cognitive Schema summarize lived, situated judgments and intentions that (in concrete form) are not "computable" (again with the caveat that our mostly science-driven worldview may imply that all reality is "computable" in some infinitely-powerful computation; I understand "computability" to terminologically exclude such a purely speculative level of capacity). However, our propensity to call up certain construals rather than others is triggered by linguistic formations, and in broad outline the catalog of these triggers, and their compositional structure, can be formalized (and even used to improve formal systems, like programming languages). The challenge is to advocate for this co-existence without implying that formal systems, and mathematically provable system-properties, are the only kind of research tools which have scientific merit.

Subjective assessments are intrinsic to most linguists' argumentation — warranting claims not with empirical data or logico-mathematical proof but by appealing to speakers' intuitions, so that reading linguistic texts is also collaborating on an ongoing research project (partly because language evolves, so word-meanings change, and formations which are ungrammatical for one generation may be experienced differently by others). Nevertheless, linguistics, like economics, seems broadly accepted as a human *science*, not just an interpretive discipline. The claim that an economist's equation or a linguist's meta-grammar are accurate explanations, useful explanatory frameworks, seems generally evaluated in terms of whether their framework captures emergent higher-order structure, and offers an explanatory potential that does not merely reiterate lower-scale paradigms. A theory expressed in the language of linguistics (not, say, neural networks), if it meets general criteria of testability and refutability (not necessarily empiricist/quantitative), arguably carries even more weight than lower-level neurophysical explanation — precisely because the higher-scale "theory language" carries the burden of explaining emergent properties, which as *emergent* bear some descriptive/behavioral (if not causal) autonomy. Likewise, a subjectively plausible and theoretically motivated equation which fits economic data probably carries more weight than a mere statistical analysis. An explanatory focus on the higher-scale in terms of its own distinct (emergent) structures and theorized entities (like words and morphemes, in the case of linguistics, or markets and commodities, in the case of economics), reflects the linguist's or economist's charge to connect human phenomena with mental (and therefore, ultimately physical) law. Nonetheless, even with liberal use of subjective judgments, economics and linguistics (and some other human sciences as well, potentially) are attached to the overall sphere of natural science, by virtue of causal links in principle even if not in practice. Scientific rigor in this humanistic setting is neither reducible to the techniques of natural science, nor dualistically separate from them. Natural science and humanities are certainly not mutually irrelevant, but nor is the proper vehicle for scientific literacy to find a forum in the humanities merely to emulate numeric methods, as with statistics in sociology, or a retreat to narrow and behavioristic reductionism, in place of localized interpretation and situational particularism.

Subjective impressions (conscious experiences, emotions, intuitions, qualia, qualitative universals and particulars — the qualitative characteristic in itself, and the hyletic-spatial trace, the site in experiential space as the quale becomes a moment of consciousness) — these are not scientifically tractable and do not have obvious physical location or measurability, which makes them controversial as objects of scientific method. Yet, even so, we do have conscious experiences, we do subconsciously (and when needed consciously, or with deliberate conscious attention) make judgments about classifications, or how parts aggregate into wholes, or are individuated apart from a larger whole in context; we can reflect on patterns in these judgments, not *introspectively* examining thoughts as they occur, but marshalling an overall familiarity with mental processes. Consciousness is not only a kind of mentality, shared by humans and some animals; it is also a metacognitive tool, something we deploy to focus attention on a certain object or topic. We "practice" how to *be* conscious, how best to distribute attention, in each setting (like an athlete maintaining a meditative state of ambient awareness, poised to latch conscious attention onto playing technique which is op-

timally instinctive, but "feels" different when degraded by fatigue or distraction). Our faculty for these modulations, switching among sub- and passive consciousness, attentive consciousness, "ambient" awareness, and back again, reveals that consciousness is not only an aspect of mind but a tool; it has a meta-cognitive and epistemic dimension, an awareness of what is known or not-yet-known and a technique of directing attention to the latter.

A case-study: in a motel I unexpectedly find a newspaper outside the door. Next morning I look outside curious whether a paper is there; after several days I come to expect the paper. So I open the door not preoccupied with confirming this, but with (maybe rather distractedly) fetching it. Initially I do not expect the paper, but, generally poised to notice both expected and unexpected circumstances, I make a mental adjustment and interpret the situation quickly; by the third day the paper has become expected, like other things I anticipate finding in a motel hallway, and the thrust of my attention, during the brief episode of my picking it up, is kinaesthetic and motor-intentional more than visual and inquisitive. Only on the second morning is the question of a paper's presence intended in an epistemic mode; but, while it is so thematized, I direct attention to optimize my ability to resolve the question. How we engage attention is a deliberate choice, reflecting and responding to our metacognitive attitudes, what we think we know and do not know.

Because consciousness is in some ways a mental tool, we have an intimate familiarity with it, a familiarity which extends beyond our own minds: we can make reasonable guesses about what others do or do not know and perceive. Our ability to anticipate others' epistemic states is an intrinsic feature of social interaction, of intersubjectivity; we therefore understand consciousness not only via our own use and possession/experience of it, but as a general feature of the human mind. We can accordingly make structured claims about conscious processes, not in the sense of introspective reports but of retrospective suggestions — by analogy, a pianist on reflection may have alot to say about playing technique, but she does not acquire this wisdom from introspective study of her own playing while it happens; rather with accrued wisdom and reflection. In terms of phenomenological method, our study of thought and consciousness is analogous: it is reflective examination of what it means to be consciously intelligent beings, not introspective psychology, or meditative meta-experience.

The methodological implications of this retrospection (as opposed to *intro*spection), how phenomenological writing seeks reflective consensus on claims about consciousness — this fashion of constructing a research community, a discursive-methodological field, does not conform to empirical scientific method, but is arguably a quite valid and defensible means of meeting the criteriological goals — the discourse ethics, the democratization of scientific participation — which physical science achieves via empiricist Ontology. For all its limitations, Positivism has the one virtue of disputational inclusiveness, demanding potential observability (not some special revelation or insight) for theoretic ur-entities. The civic norms of phenomenology are more complex, because both "transcendental" analysis of consciousness — as a kind of philosophical ground zero, a neo-Cartesian fortress against skepticism and empiricism — and also a more pluralistic, enculturated, embodied, social phenomenology, are well-represented (and interpenetrate in complex ways) in the continuing post-Husserl tradition. That being said, even in its most neo-Idealist, reifying consciousness as a primordial frame on any cognitive-scientific reasoning, as human sciences' condition of possibility, phenomenology cannot help but textually acknowledge pluralism, and philosophical collaboration — precisely because its claims are not descriptive of empirically locatable/observable objects.

Interestingly, the phenomenological tradition reveals substantial interest in both the socio-political and the formal-mathematical: this is not so noteworthy in itself, because Analytic philosophy also connects (say) language with (say) logic, but phenomenology is distinct in that it joins the humanistic and the formal/mathematical without the same tendency to hone in on a overlapping, logico-semantic core. In writings where Analytic philosophers appear to address both social and mathematical concerns, usually their underlying motivation, or so it seems to me, is to find some logical underpinnings to linguistic or cognitive structure (say, *implicatures*) — logic, subject to formal treatment, also manifesting itself in the organization of thoughts and expressions. Amongst phenomenologists, however, for example Husserl, Merleau-Ponty (in his science-oriented writings; [54]), and Anglo-American writers in the "Naturalizing Phenomenology" tradition, there is evident interest in mathematics *apart*

from logic: topology, differential geometry, mereotopology, multi-granularity.[14] Phenomenology therefore uncovers an arguably deeper and truer bridge between human and "eidetic" sciences, in Petitot's phrase, one which is not pre-loaded with logico-reductive presuppositions. If this is accurate, phenomenology can provide a deeper methodology for the humanities in their interactions with natural science. Even insofar as we stay committed to the idea that social/cultural/mental phenomena emerge from (neuro-)physical ones, we need to curate methods for these "emergent" sciences which have the requisite theoretical autonomy to actually extend the explanatory reach of the natural sciences on which they causally rest. Cognitive Linguistics, I would argue, is a good example of this notion of autonomy, and its methodology, I would also argue, bears an important resemblance to phenomenological research.

Another brief case-study (revisiting footnote 2): our environing world mostly discloses itself through objects' visible exterior: as much as we have on occasion a palpable sense of volume as well (as when looking through a fog) — and as much as what we see is inextricable from our embodied interactions with objects, adding tactile and kinaesthetic dimensions, a canonical sense of perception is still the vision of distant objects, usually through their surface geometry. A canonical example of perceptual cognition is therefore reconstructing geometry from visual appearances, especially color gradations — mathematically, converting "color" vector fields to curvature vector fields (it's worth noting that color is an almost primordial example of a Conceptual Space Theory as developed by Gärdenfors and others [83]). This kind of transformation, described (say) via differential geometry, is *qua* theoretical device an example of semiotic morphism, a mapping between representation disciplines [32], [31]. The point is not, however, that there are precise correlates in the brain which "implement" this procedure; that the semiotic morphism takes a domain and codomain that quantify over empirically locatable, neurophysical entities. We can study how software reconstructs geometry from color data as an approximation to a *process*, a model-building whose semiotics of approximation is coarse-grained and holisitc.[15] Formal devices like vectors or vector fields need not mold symbolic systems by mapping individual symbols to spacetime objects, or processes, but rather afford representation-mappings that capture cognition indirectly and patternwise.

I make this point using visual consciousness as an example, but it applies also to cognitive grammar, where the color -to- curvature-vector morphism has an analogue in the mapping of word-sequences to tree- or graph-algebras. I do not intend to claim that there are specific, individuated neurophysical analogues to theoretical posits in the symbolic regime I sketched earlier, in terms of POS and lexical annotations, inter-word and inter-phrase connections, applicative structures, and the rest. There are not, necessarily, for example, little brain regions whose role is to represent different types of phrase structures (e.g., different flavors of pluralization). Our explanatory ambitions, instead, should be cognitive-linguistic models of a global process-structure, agnostic about one-to-one correspondence between the posits of the theory and the empirical stuff whose behaviors it wants to explain. Cognitive triggers bridge formal/empirical sciences with the phenomenolog-

[14] Not that logic is wholly unrelated to these subjects: consider topological and type/Category-theoretic embeddings of logical systems within certain categories, or technical domains, like toposes, sheaves, granules; but logic in this sense, mathematically founded within spaces otherwise discussed at least as metaphoric guides within phenomenology, does not appear to be the dominant understanding of logic in the Analytic philosophical tradition. To be fair, style may dictate that argumentation should be trimmed to its essential elements, and mathematical deductions are rarely if ever essential for defending phenomenological claims. In Jean Petitot, for example, mathematics is sometimes intrinsic to empirical backing for phenomenological ideas, but other times (say, sheaf mereology), the formal theories, while useful analogies, do not clearly pair up with logico-deductive justifications. But, I would reply, there is so much unexplained about consciousness, and cognition as it occurs in conscious minds — the controversial "Explanatory Gap" between mind and matter — that much of the important argumentation does not yet have deductive signposts; we need an effective methodology which is not so linear. As we approach beyond a simplifying, logico-functionalist vantage, which we eventually must transcend, both functionalization and empiricism fall by the wayside as reasonable methods for "Naturalizing" consciousness. We have to accept when the formal/mathematical stands as more intuitive than rhetorical, on pain of "Naturalization" being quarantined from a humanistic core entirely.

[15] The experiential verisimilitude of computer graphics is a phenomenological data point, but so is their obvious unreality — the mathematics reveals something about, but is not an all-encompassing model for, shape and color *qua* material phenomenon, still less the neuroscience of color experience. Morphism between structures may model *processes* more correctly than the structures themselves approximate their substrata — but this is no longer a semiotics of causal/physical reductionism, a use of mathematics (like differential geometry) to iconify empirical givens, the way that (say) the Navier-Stokes equations are understood to refer explicitly to (even while idealizing and abstracting from) fluid-mechanical dynamics. Our theory-semiotics has to locate the site of designation at a more oblique scale, a different Ontological register, of processes and transformations — seeing in phenomena the image of a theoretical model because of its global structure, as a sign in its own right, rather than a collage of symbols and numbers to which are reduced spatializations and trajectories of causation and physical influence.

ical/humanistic: their causal engenderings are physical and structural phenomena, but their manifestation in the world is not fully tractable without an interpersonal deliberation accounting for both the privateness of consciousness and the sociality of mind, and, so, something akin to phenomenology.

It may appear that I am describing a weak-functional theory (or metatheory) which uses functional description in lieu of precise micro-physical explanation — in other words, that in lieu of explaining precisely how the brain achieves vision or language, we describe functional capabilities that are prerequisite for these competences, and refactor the goal of scientific explanation as to describe the system of intermediate functionality as correctly as possible, rather than describe how this functionality is physically realized. In a strong form, this re-orientation yields functionalism in theories/philosophies of Mind, that try to refrain from Ontological commitments to mental states or properties *apart from* descriptions of their functional roles. In other words, according to the parameters of the field of study and its institutions, even if not deep metaphysical beliefs, mental states are reducible to functional states, and cognitive systems are scientifically equivalent if they reveal similar functional organization, whether they belong to human or animal minds or computers or extra-terrestrials. A more modest functionalism would reject the implied reductionistic (maybe eliminative) Ontological stance, and maintain that mental things are not wholly, metaphysically subsumed by their functional organization, while still practicing a kind of theory whereby this functional organization is the proper object of study; the specific aspect of the mental realm which is scientifically tractable.

I do not believe I am making even such weak-functionalist claims: either branch of functionalism can misattribute the methodological association between theoretical structures and explanatory goals. We may be led toward the stronger or weaker functionalist viewpoints if we understand that a cognitive theory should task itself with making symbolic icons for scientifically grounded referents, grounded in an abstract space of functional organization if not in empirical space-time. Of course, most scientific explanation does construct a specialized, technical semiotics whose signs refer into either formal spaces or accounts of empirical space-bound things, however abstracted or idealized. But, conversely, insofar as I propose to focus on functional structures, and particularly cross-representation-framework transformations, my intent is to "functionalize" the discursive norms of the theory, not the phenomena it investigates. In order to negotiate between the competing demands of scientific rigor and formalization — on the one hand — with the immediacy and etheriality and subjectivity of consciousness, on the other, we need to "attach" theoretical structures to mental phenomena without getting bogged down in questions of the scientific or Ontological status of mental things, how they are "scientific" individually and collectively (collectively as in the Ontology of "Mind" overall).

This suggests adopting functional attitudes not in the theory but the metatheory: to use functionalism as an organizing principle on the theoretical *discourse*, on the attitudes of the scientists and scholars who want to straddle the divide between natural and mathematical sciences and humanism and consciousness. The "semiotic morphism" of color-to-curvature vector fields, or word-sequences to typed semantic graphs, are recommendations for guidelines on how researchers should write and communicate about cognitive processes in their global structure. I have tried to outline a metadiscourse more than a metalanguage — not a template for building theory-languages whose signs refer into a realm of posited empirical or abstract entities, but a template for using certain formal-mathematical constructions (in domains like typed lambda calculus, type theory, or differential geometry) as a textual prelude, a way to position the norms of writing to be receptive to both scientific-mathematical and phenomenological concerns. If semiotic morphisms like color-to-curvature or word-sequence-to-semantic-graph have explanatory merit as ways to picture cognitive processes, this merit is intended to be judged according to how it affects discursive norms on this scientific borderlands between mathematics and humanities, rather than how it reduces empirical phenomena to mathematizable abstractions. If there is *something* in cognition analogous to these morphisms, even if "analogous" means merely that holding the morphisms as formally defined in our minds while thinking about cognition can show us philosophical ways forward, then we should be interested in refining these formalizations as part of the overall Cognitive-Phenomenological project.

The Cognitive-phenomenological project is very different, I believe, than the AI or Artificial General Intelligence projects. Nevertheless, as I noted to conclude Section 5, AI — for all its reductive ideology — does show the benefits of an intellectual framework where researchers can experiment, try things out, and write code. We should not underestimate the power of technology and experimentation to ground and engage the scholarly process: it allows the scholar to program her own research environment, autonomous as necessary from academic and institutional paradigms — which, notwithstanding a general academic commitment to innovation, can get mired in inertia: particularly when it comes to interdisciplinary methodology and particularly when it comes to reengineering the publishing process and the dissemination of scholarship. There is a lot of technical and technological potential which in the academic and publishing communities is not being realized.

This is not just a procedural claim tangential to actual scholarly argumentation: we need new generations of publishing tools to properly synthesize computational technology with nonreductive, humanities-based philosophies of mind and consciousness. We need to properly implement the technological tools that empower individual scholars, without buying uncritically into academic and corporate appropriations of technology for regressive ends.

In this sense we should look at formalizing but not *reductionistic* projects as case-studies in software or investigative tools whose practical dimension spurs hands-on experimentation and decentralized, extra-institutional open-source collaboration, but whose theoretical commitments gravitate to cognitive linguistics and phenomenology — while bypassing an AI paradigm that underestimates the cognitive importance but complexity of social-situational awareness and of sensate consciousness. AI is not a canonical arbiter of software practicality (our contemporary instinct toward measuring all software around AI-driven analytics and "Big Data" reflects a clever marketing campaign by companies with financial incentives to prioritize AI research over other disciplines). Nor is AI a value-neutral or politically progressive vision of what human mind and society are like.

Perhaps this is part of what it means to be a phenomenologist in the 21st century: not to reject technology or computational models or to believe in a mode of phenomenological research carried by pure thought, but to embrace — as part of the research infrastructure, of our own respective academic identities — practical software that suggests interesting cognitive-humanistic paradigms without endorsing reductive AI hypotheses. Insofar as scholarship is a social phenomenon, the metaphilosophy of "pure thought" is an illusion anyhow: theory is inevitably mediated by the disciplinary expectations of its audience. Given this reality, software offers a renewed agency and autonomy to the researcher: computer code does not intrinsically know from disciplinary norms, and the code-writer is programming a medium where disciplinary boundaries can fade out — if the program compiles. The programmer does not *argue* for interdisciplinarity; she *implements* it.

Technology, in conclusion, can liberate scholarship from disciplinary inertia in the same gesture as open-source software liberates technology itself from commercial oligarchizing. Good open-source software programs monetary inequalities out of existence; as humanities scholars we have an analogous duty to program disparaties in intellectual capital and influence out of existence. Open-source software is the fiat currency of the digital commons; by analogy, phenomenology is the liberation theology of the intellectual commons. We don't argue for a just and existential foundation of the humanities and the natural/social science interface: we implement it.

Sophisticated but philosophically and morally responsible cognitive-computational paradigms are probably more likely to arise from adding formal methodology and open-source experimentalism to a fundamentally humanities foundation, rather than bringing sensitivity to human nuance to a natural-science academic tradition. The reasons for this are institutional as well as intellectual: insofar as formal-computational models are still rather unfamiliar in humanities contexts, practitioners in a hybrid cognitive-computational-humanities orientation can have a level of autonomy that helps us distinguish sophisticated computational models from simplistic philosophical (and commercial) paradigms. And the affordances of open-source code and digital publishing supports a robust but low-cost technological environment, tangential to academic laboratories and hierarchies.

Perhaps this open-source ecosystem is a worthy 21st-century field wherein to continue 20th-century phe-

nomenology. Let's not forget that phenomenology began as a philosophy of mathematics but evolved into a moral, political, and Existential system. Honoring the subtlety of human consciousness is a way to respect the technical priorities of phenomenological philosophy but also the political activism that — certainly often rendered into praxis by the intersectionality of lived experience with race, class, and gender — follows from phenomenological ethics.

Kant's critical philosophy did not only inspire generations of abstract Idealism; it spurred the cosmopolitan ideal of a Community of Nations and the municipalist axiology of, in particular, Kant's 19th century French translator, Jules Barni. Our communal existence is not intrinsically, cognitively, tribal or chauvinistic: the basic adaptation of human minds to ecologies transcends race, class, and culture. Cultural differences are real, but external: enculturation is minds being shaped by the natural and civil infrastructure around us. While preserving nation and community as a practical medium, Kantianism unmasks nationalism as a metaphysical gambit. This is perhaps how, for Barni, Critical Philosophy flows organically into municipalist activism: to understand the mind we have to embed ourselves into the cognitive patterning of mind, which means the environs where cognition is honed, which means our urban and neighborhood ecologies. The architecture of cognition lies not only in the cultures we receive via transmission — religion, education, inter-generational ideologies — but in the grid of the streets outside our front door.

The axis from Kant to Barni has 21st century analogs: Husserl as our Kant, and progressive ethical frameworks like Murray Bookshin's libertarian municipalism taking the mantle from Barni's post-1848 variety. But one key difference is that communities are now partly (though of course not entirely) digital, virtual, and technological. So, I believe, part of being a phenomenologist in the 21st century is to — as much as we can, and virtually if need be — implement a municipalism in our time that metastatizes Husserl in a recapitulation of how, for example, Barni evolved upon Kant. This is not just an abstract exercise, because intellectuals are performing their commitment to humanities scholarship, to the progressive and cosmopolitan spirit of humanities discourse, in environments far more challenging than we associated with Western academia — Budapest; Rojava. This is part of what I had in mind referring to "phenomenological ethics".

Wes Enzinna, a New York Times reporter who taught a journalism class at the Mesopotamian Social Sciences Academy in Qamishli, told a story (in the Times Sunday Magazine, November 29 2015) about his reconciliation with students after a brief culture-shock-like falling out:

> "We reject the master-and-slave relationship as a model for the teacher-and-student relationship," Ali said. "But we've decided that you're welcome to continue teaching us." Ramah, the atheist, stood up and said, "I'm so happy you're here." They all approached my desk and turned in their assignments.

Sherhad Naaima, another Kurdish activist a few months earlier, put it this way (in an interview with Eleanor Finley from the Institute for Social Ecology in Vermont):

> History is a river, it cannot be cut. We have no West or East, but rather one history which is moving and retaining all human culture ... [T]he Left needs to dive deeper into hidden history and revive their own traditions of freedom and the idea of a utopia of freedom. They then must build a holistic theory provided by the unity of natural sciences and social sciences. That new theory can be called "the epistemology of freedom".

These testimonies are what "to the things themselves" means today.

References

1 Martín Abadi and Luca Cardelli, "A Theory of Primitive Objects: Untyped and First-Order Systems" http://lucacardelli.name/Papers/PrimObj1stOrder.pdf

2 Benjamin Adams and Martin Raubal, "A Metric Conceptual Space Algebra". https://pdfs.semanticscholar.org/521a/cbab9658df27acd9f40bba2b9445f75d681c.pdf

3 Benjamin Adams and Martin Raubal, "Conceptual Space Markup Language (CSML): Towards the Cognitive Semantic Web". http://idwebhost-202-147.ethz.ch/Publications/RefConferences/ICSC_2009_AdamsRaubal_Camera-FINAL.pdf

4 Robert St. Amant, *et. al.*, "An Image Schema Language" https://apps.dtic.mil/dtic/tr/fulltext/u2/a458943.pdf

5 Houda Anoun, Pierre Castéran, and Richard Moot, "Proof Automation for Type-Logical Grammars" http://www.labri.fr/perso/moot/esslli_reader.pdf

6 Nicholas Asher and James Pustejovsky, "A Type Composition Logic for Generative Lexicon" https://www.cs.brandeis.edu/~jamesp/classes/cs216-2009/readings2009/TCLforGL.pdf

7 Jean-Françqis Baget *et. al*, "Translations between RDF(S) and Conceptual Graphs". http://www.lirmm.fr/~croitoru/rdfs.pdf

8 Chris Barker and Chung-Chieh Shan, "Continuations and Natural Language". http://semanticsarchive.net/barker/barker-shan-continuations-book.pdf

9 Radim Bělohlávek and Vladimır Sklenář, "Formal Concept Analysis Constrained by Attribute-Dependency Formulas" B. Ganter and R. Godin, eds.,: ICFCA 2005, LNCS 3403, pp. 176-191, Berlin, Springer-Verlag, 2005. http://belohlavek.inf.upol.cz/publications/BeSk_Fcacadf.pdf

10 Benjamin K. Bergen and Nancy Chang, "Embodied Construction Grammar in Simulation-Based Language Understanding". http://www1.icsi.berkeley.edu/~nchang/pubs/ecg.pdf

11 Jean-Philippe Bernardy, et. al., "Parametricity and Dependent Types". http://www.staff.city.ac.uk/~ross/papers/pts.pdf

12 Thomas Bittner, Barry Smith, and Maureen Donnelly, "The logic of systems of granular partitions." http://ontology.buffalo.edu/smith/articles/BittnerSmithDonnelly.pdf

13 Hans C. Boas and Ivan A. Sag, eds. "Sign-Based Construction Grammar". http://lingo.stanford.edu/sag/papers/theo-syno.pdf

14 Mathieu Bouchard, "A Type Theory for the Documentation of PureData" http://artengine.ca/~catalogue-pd/43-Bouchard.pdf

15 Line Brandt, "The Communicative Mind." Newcastle-upon-Tyne: Cambridge Scholars Publishing, 2013

16 Per Aage Brandt, "Spaces, Domains, and Meaning: Essays in Cognitive Semiotics". Newcastle-upon-Tyne: Cambridge Scholars Publishing, 2013 http://semiotics.au.dk/fileadmin/Semiotics/pdf/per-aage-brandt/Per_Aage_Brandt__Spaces__Domains__and_Meaning._Essays_in_Co.pdf

17 Roberto Casati and Achille C. Varzi, "Parts and Places: The Structures of Spatial Representation" MIT Press, 1999

18 *Speech Lost from Speech: The Cognitive Linguistics of Alienation, Objectification, and Reclaiming*. Doctoral Dissertation, Berkeley. http://escholarship.org/uc/item/7rv3j4nf

19 Stergios Chatzikyriakidis and Zhaohui Luo, " Individuation Criteria, Dot-types and Copredication: A View from Modern Type Theories." *Association for Computational Linguistics, Proceedings of the 14th Meeting on the Mathematics of Language (MoL 14)*, pp. 39{50, 2015. http://www.aclweb.org/anthology/W15-2304

20 Nathaniel Christen, "Hypergraph Type Theory for Specifications-Conformant Code and Generalized Lambda Calculus". *Advances in Ubiquitous Computing: Cyber–Physical Systems, Smart Cities, and Ecological Monitoring*, Amy Neustein, ed., Elsevier, 2019.

21 Mary Dalrymple, "Lexical Functional Grammar". http://web.stanford.edu/group/nasslli/courses/as-cr-da/apbook-nasslli.pdf

22 Antonin Delpeuch and Anne Preller, "From Natural Language to RDF Graphs with Pregroups". *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pp. 55{62, Gothenburg, Sweden, April 26-30 2014. https://www.aclweb.org/anthology/W/W14/W14-1407.pdf

23 Jean-Pierre Desclés, "Reasoning in Natural Language in Using Combinatory Logic and Topology: An Example with Aspect and Temporal Relations." https://www.aaai.org/ocs/index.php/FLAIRS/2010/paper/viewFile/1350/1736

24 Martín Escardó, "Synthetic topology of data types and classical spaces". *Electronic Notes in Theoretical Computer Science* 87, 2004. http://www.cs.bham.ac.uk/~mhe/papers/barbados.pdf

25 Henrietta Eyre and Jonathan Lawry, "Language Games with Vague Categories and Negations" http://adb.sagepub.com/content/early/2014/09/04/1059712314547318

26 Gilled Fauconnier, "Mental Spaces: Aspects of Meaning Construction in Natural Language". Cambridge, 1994

27 Kit Fine, "Part-whole", in Barry Smith and David Woodruff Smith, *The Cambridge Companion to Husserl*, Cambridge University Press, New York, 1995, pp. 463.

28 Bruno Gaume, Fabienne Venant, and Bernard Victorri, "Hierarchy in Lexical Organisation of Natural Languages". *Hiearchy in Natural and social Sciences*, Methodos series, vol 3, Springer, 2006, pp. 1039-1058, 2013. https://pdfs.semanticscholar.org/ac47/62895ba5019bd556a8c9b4cde0a36b20ea0c.pdf

29 Peter Gärdenfors and Frank Zenker, "Theory Change as Dimensional Change: Conceptual Spaces Applied to the Dynamics of Empirical Theories". *Synthese 190(6)*, pp. 1039-1058, 2013. http://lup.lub.lu.se/record/1775234

30 Gianluca Giorgolo and Ash Asudeh, "Monads as a Solution for Generalized Opacity". Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS), pages 19{27, Gothenburg, Sweden, April 26-30 2014. http://www.aclweb.org/anthology/W14-1403

31 Joseph Goguen, "What is a Concept?" https://cseweb.ucsd.edu/~goguen/pps/iccs05.pdf

32 ——— "Semiotic Morphisms, Representations, and Blending for Interface Design" https://cseweb.ucsd.edu/~goguen/pps/uid.pdf

33 Albert Goldfain, "Vital Sign Ontology". https://philarchive.org/archive/GOLVSOv1

34 Kenneth Holmqvist, "Conceptual Engineering: Implementing cognitive semantics", in Jens Allwood and Peter Gärdenfors, eds., *Cognitive Semantics*, pp 153 - 171, Amsterdam, Philadelphia: John Benjamins, 1999.

35 Kenneth Holmqvist, "Implementing Cognitive Semantics: Image schemata, valence accommodation, and valence suggestion for AI and computational linguistics". PhD thesis, Dept. of Cognitive Science, Lund University, Lund, Sweden, 1993.

36 The Univalent Foundations Program, "Homotopy Type Theory: Univalent Foundations of Mathematics" Princeton 2013 https://hott.github.io/book/nightly/hott-online-1075-g3c53219.pdf

37 Mark Johnson, "The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason". University of Chicago Press, 1990

38 Sean Dorrance Kelley, "Seeing Things in Merleau-Ponty". http://www.nyu.edu/gsas/dept/philo/courses/representation/papers/Kelly.pdf

39 Ozkan Kilic, "Intelligent Reasoning on Natural Language Data: A non-axiomatic reasoning system approach" https://pdfs.semanticscholar.org/2324/87388d5a11ec729e352501bd0012e89cf7de.pdf

40 Oleg Kiselyov, "How to Remove a Dynamic Prompt: Static and dynamic delimited continuation operators are equally expressible" https://www.researchgate.net/publication/238123187_How_to_remove_a_dynamic_prompt_static_and_dynamic_delimited_continuation_operators_are_equally_expressible

41 Oleg Kiselyov, "Applicative Abstract Categorial Grammars". http://okmij.org/ftp/gengo/applicative-symantics/AACG.pdf

42 Oleg Kiselyov and Chung-chieh Shan, "Lambda: the ultimate syntax-semantics interface" https://pdfs.semanticscholar.org/fb41/793ae7098c40fcdb6706b905c48a270b0b48.pdf

43 George Lakoff and Mark Johnson, "Philosophy in the Flesh: the Embodied Mind and its Challenge to Western Thought". New York: Basic Books, 1999.

44 Ronald Langacker, "Nouns and Verbs" *Language*, Vol. 63, No. 1 (March 1987) pp. 53-94.

45 Ronald Langacker, "Foundations of Cognitive Grammar, vol. 1". Stanford University Press, 1991

46 "Dual PECCS: A Cognitive System for Conceptual Representation and Categorization" https://www.researchgate.net/publication/304208795_Dual_PECCS_A_Cognitive_System_for_Conceptual_Representation_and_Categorization

47 Zhaohui Luo, "Type-Theoretical Semantics with Coercive Subtyping". https://www.cs.rhul.ac.uk/home/zhaohui/ESSLLI11notes.pdf

48 Zhaohui Luo and Sergei Soloviev, "Dependent Coercions". https://www.sciencedirect.com/science/article/pii/S1571066105803147

49 Jean Maillard, Stephen Clark, and Edward Grefenstette, "A Type-Driven Tensor-Based Semantics for CCG". http://www.cl.cam.ac.uk/~sc609/pubs/eacl14types.pdf

50 Griselda Pollock, *Modernity and the Spaces of Femininity*. https://msu.edu/course/ha/446/griseldapollock.pdf

51 Scott Martin and Carl Pollard, "A Dynamic Categorial Grammar". http://www.ling.ohio-state.edu/~murat/dycg.pdf

52 Bruno Mery, Richard Moot, and Christian Retoré, " Plurals: Individuals and sets in a richly typed semantics". http://arxiv.org/pdf/1401.0660.pdf 3 Jan 2014

53 Bruno Mery and Christian Retoré, " Semantic Types, Lexical Sorts and Slassifiers". *10th International Workshop on Natural Language Processing and Cognitive Science (NLPCS '10), 15 October 2013 - 17 October 2013 (Marseilles, France)* https://core.ac.uk/download/pdf/78384461.pdf

54 David Morris, "The Sense of Space". State University of New York, 2004

55 Stefan Müller, "Unifying everything: Some remarks on simpler syntax, construction grammar, minimalism, and HPSG" http://www.linguisticsociety.org/sites/default/files/Lg_89_4_Muller.pdf

56 Stefan Müller, "Grammatical theory: From transformational grammar to constraint-based approaches". Berlin: Language Science Press, 2016

57 Esther Pascual, "Fictive Interaction: The conversation frame in thought, language, and discourse". Philadelphia, John Benjamins, 2014.

58 Santanu Paul and Atul Prakash, "Supporting Queries on Source Code: A Formal Framework" https://www.cse.umich.edu/techreports/cse/94/CSE-TR-209-94.pdf

59 Efe Peker, "Following 9/11: George W. Bush's Discursive Re-Articulation of American Social Identity" Master's Dissertation, Linköping University, Sweden http://www.diva-portal.org/smash/get/diva2:21407/FULLTEXT01.pdf

60 Jean Petitot, et. al., eds., "Naturalizing Phenomenology: Issues in Contemporary Phenomenology and Cognitive Science". Stanford University Press, 1999.

61 Jean Petitot, "Syntax Topologique et Grammaire Cognitive". *Langages 25.103*, pp. 97-128, 1991.

62 Jean Petitot, "The morphodynamical turn of cognitive linguistics". https://journals.openedition.org/signata/549

63 Steven Pinker, "The Stuff of Thought: Language As a Window Into Human Nature". Penguin, 2007.

64 Carl Pollard, "Covert Movement in Logical Grammar". http://www.ling.ohio-state.edu/~pollard/cvg/covert.pdf

65 Martin Raubal, "Formalizing Conceptual Spaces". http://www.raubal.ethz.ch/Courses/288MR_Spring08_Papers/Raubal_FormalizingConceptualSpaces_FOIS04.pdf

66 Giuseppe Riva and Fabrizia Mantovani, "Being There: Understanding the Feeling of Presence in a Synthetic Environment and Its Potential for Clinical Change". http://cdn.intechopen.com/pdfs-wm/39042.pdf

67 Aurélie Rossi, "Applicative and Combinatory Categorial Grammar: Analysis of the French Interrogative Sentences". *FLAIRS Conference, Association for the Advancement of Artificial Intelligence*, 2008. https://www.aaai.org/Papers/FLAIRS/2008/FLAIRS08-118.pdf

68 Anthony J. Roy and John Stell, "Convexivity in Discrete Space". http://www.comp.leeds.ac.uk/jgs/RoyStellCOSIT03.pdf

69 Salman Saghafi, et. al., "Features and Object Capabilities: Reconciling Two Visions of Modularity". https://cs.brown.edu/~sk/Publications/Papers/Published/sfk-feat-ocap-reconcil/paper.pdf

70 Chung-Chieh Shan, "Monads for natural language semantics". http://arxiv.org/pdf/cs/0205026.pdf 17 May 2002

71 ——"Linguistic side effects." Harvard University, dissertation 2005.

72 Matt Selway, *et. al*, "Configuring Domain Knowledge for Natural Language Understanding" http://ceur-ws.org/Vol-1128/paper9.pdf

73 Chung-Chieh Shan, "Shift to control". Proceedings of the *Workshop on Scheme and Functional Programming'*, pp. 99{107, 2004. http://homes.sice.indiana.edu/ccshan/recur/recur.pdf

74 Daniel D. Sleator and Davy Tamperley, "Parsing English with a Link Grammar". https://www.link.cs.cmu.edu/link/ftp-site/link-grammar/LG-IWPT93.pdf

75 Barry Smith, "Mereotopology: A Theory of Parts and Boundaries". http://nemo.nic.uoregon.edu/wiki/images/3/38/Smith1996_Mereotopology.pdf .

76 Barry Smith, "The Ontology of Blood Pressure: A Case Study in Creating Ontological Partitions in Biomedicine". https://www.researchgate.net/publication/228961604_The_Ontology_of_Blood_Pressure_A_Case_Study_in_Creating_Ontological_Partitions_in_Biomedicine

77 David Woodruff Smith, "Mind World". Cambridge University Press, 2004.

78 ——"Mind and body", "The Cambridge Companion to Husserl", New York 1995.

79 David Woodruff Smith and Ronald McIntyre, "Husserl and Intentionality: A Study of Mind, Meaning, and Language". Dordrecht and Boston: D. Reidel, 1982.

80 Michael Anthony Smith and Jeremy Gibbons, "Unifying Theories of Objects" http://www.cs.ox.ac.uk/jeremy.gibbons/publications/uto.pdf

81 John F. Sowa, "From Existential Graphs to Conceptual Graphs". http://www.jfsowa.com/pubs/eg2cg.pdf

82 David Spivak and Robert Kent, "Ologs: A Categorical Framework for Knowledge Representation" https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0024274&type=printable

83 Gregor Strle, "Semantics Within: The Representation of Meaning Through Conceptual Spaces". Univ. of Novi Gorici, dissertation, 2012.

84 Leonard Talmy, "Force Dynamics in Language and Cognition". From "Toward a Cognitive Semantics", Vol. 1, MIT Press 2000. http://linguistics.buffalo.edu/people/faculty/talmy/talmyweb/Volume1/chap7.pdf

85 Ribeka Tanaka, et. al., "Factivity and Presupposition in Dependent Type Semantics" http://www.lirmm.fr/tytles/Articles/Tanaka.pdf

86 Elke Teich, "Systemic Functional Grammar & Natural Language Generation" New York: Cassell, 1999.

87 Orlin Vakarelov, "Pre-cognitive Semantic Information". https://link.springer.com/article/10.1007/s12130-010-9109-5

88 Orlin Vakarelov, "The cognitive agent: Overcoming informational limits" https://philarchive.org/archive/VAKTCAv1

89 "Representing Content: Semantics, Ontology, and their Interplay" Dissertation, Université Paul Sabatier, Toulouse https://www.irit.fr/publis/LILAC/LV-HDR09.pdf

90 Anne Vilnat, "Dialogue et analyse de phrases" Habilitation Mémoire, Université Paris-Sud https://perso.limsi.fr/anne/HDR/MemoireHDR.pdf

91 Jørgen Villadsen, "Multi-dimensional Type Theory: Rules, Categories, and Combinators for Syntax and Semantics." http://arxiv.org/pdf/cs/0408037.pdf 15 August 2004

92 Yves-Marie Visetti and Pierre Cadiot, "Instability and theory of semantic forms". S. Feigenbaum and D. Kurzon, eds., *Prepositions in their Syntactic, Semantic and Pragmatic Context*, Amsterdam, John Benjamins, 2002, pp. 9-39. http://formes-symboliques.org/IMG/pdf/doc-14.pdf

93 Pei Wang, "What Do You Mean by ''AI"?" https://cis.temple.edu/~pwang/Publication/AI_Definitions.pdf

94 "Understanding ACT-R --- an Outsider's Perspective" https://arxiv.org/pdf/1306.0125.pdf

95 Olav K. Wiegand, "A Formalism Supplementing Cognitive Semantics Based on a New Approach to Mereology". Ingvar Johansson, Bertin Klein and Thomas Roth-Berghofer, eds., *Contributions to the Third International Workshop on Philosophy and Informatics*, 2006.

96 ——"On referring to Gestalts". Mirja Hartimo, ed., *Phenomenology and Mathematics*, pp. 183-211. Springer, 2010.

97 Rudolf Wille, "Conceptual Graphs and Formal Concept Analysis". *Conceptual Structures: Fulfilling Peirce's Dream Fifth International Conference on Conceptual Structures*, 1997.

98 Yiyu Yao, "A Comparative Study of Formal Concept Analysis and Rough Set Theory in Data Analysis". http://www2.cs.uregina.ca/~yyao/PAPERS/Rough_concept.pdf

99 Jordan Zlatev, "The Dependence of Language on Consciousness". http://www.mrtc.mdh.se/~gdc01/work/ARTICLES/2014/4-IACAP%202014/IACAP14-GDC/pdf/Language-Consciousness-Zlatev.pdf