4.1.1 Semantic Symmetries

One way to look at our cognitive comfort zones is as regions of the information processing possibility space in which these asymmetries collapse, either because we're using more or less deterministic recipes that are as easy to follow as they are to check, as easy to learn as they are to teach, or because we've all got enough computational resources not to worry about them, when it's so easy to create/discover a workable rule that it makes no functional difference. Another way of understanding this is to say these are regions in which *communication* becomes possible. Where sender and receiver achieve something resembling computational symmetry. It's important to explain that what I mean by communication here is more than mere information transmission. When one looks at language from an information theoretic perspective, just how complex a signal it is once more depends on the size of the possibility space in terms which you calculate the relevant probabilities.

Let's take English. Are we looking at strings of *letters* (and punctuation), some thirty-odd possible symbols bunched into pseudo-sentential chunks? Are we looking at series of *phonemes*, forty-four in all combined in various limited ways? Or are we looking at sequences of *sentences*, each some combination of a near two-hundred thousand words, according to some more or less implicit set of grammatical rules? Even at this level, we're still not talking about semantics, only syntax, and the amount of Shannon information will vary with what we pick. Chomsky's famous example - "Colourless green ideas sleep furiously" - is *still* grammatically well formed and *still* completely meaningless. We have to see that in one sense, language transmits an incredibly small amount of information, and in another, it transmits a potentially vast amount. For example, a sign that reads "Beware the Dog" is only fourteen characters long, but it means an awful lot in certain situations. The presence of a dangerous dog in your immediate environment is a *very* significant and informationally rich signal to be looped through your various predictive layers. The semantic content begins to look like its hidden in the symmetric capacities of the sender and receiver to *compress* and *decompress* the syntactic signal with so little effort they don't even notice.

It's about now that I get to sneak in a little computational Kantianism. Kant would characterise these cognitive comfort zones in terms of what he calls analytic judgments, or those judgments which express the *content* of the concepts through which we *understand* the world (*Verstehen*). For instance, 'cats are mammals' and 'mammals are animals' are both analytic judgments. Taken together they allow you to infer the seemingly banal fact that any given cat is an animal. Kant thinks that there are other, synthetic judgments such as 'water boils at 100 degrees centigrade at one atmosphere of pressure', which enable us to draw much more interesting inferences (e.g., about how we should go about making a cup of tea), and whose truth is not dependent on the content of the concepts that compose them (i.e., water, temperature, and pressure). Indeed, once we know that 'mercury expands when heated' and precisely how much, we can make a nice little thermometer to check our water temperature, and thereby just how much its boiling point changes as we vary atmospheric pressure. I've cheated a bit here, because it's likely that the unit of measure (degrees centrigrade) was defined through using facts about the expansion of mercury, but I think you get the drift.

It would seem that synthetic judgments are in some important sense *asymmetric*. If nothing else, when you send one to someone else you can communicate information they don't already have. Whereas an analytic judgment is less like useful information and more like a *test* of whether someone understands the concepts needed to receive informative judgments. Seeing if someone else's understanding of where your cat sits in the hierarchy of genus and species matches yours is one way of checking whether you're talking about the same thing (e.g., "You know sphynxes are a type of cat, right?"), and thus whether you can tell them that cats have a toxic reaction to theobromine, and so they shouldn't give Chairman Meow chocolate (e.g., "You know, the sweet stuff made from cocoa?"). There's another asymmetry here though: it seems like synthetic judgments are like rules that *must* be discovered, whereas analytic judgments are like rules that can *only* be followed. How does this work?

4.1.2 Synthetic Asymmetries

One of the great philosophical ironies is that 'Analytic' philosophy has for the most part completely misunderstood the distinction between analytic and synthetic judgments. The relevant debates are dominated by philosophers who think that tame definitions of purely conventional terms such as 'bachelors are unmarried men' are sufficient examples of analyticity, while *slightly* more subtle thinkers like Quine and Brandom use the inadequacy of these tame examples to reject analyticity altogether, advocating some sort of unrestricted semantic holism. I think Mark Wilson has provided the correct response to Quine, which is that since unrestricted semantic holism is computationally intractable in practice, there can be nothing whose meaning is globally dependent on everything else, but only more local, if revisable, regions of significance. However, the best thinkers in this regard are two of the great modernist logicians: Per Martin-Löf and Jean-Yves Girard (I'd claim the third is William Lawvere, but he's not relevant here).

Martin-Löf has shown in an simple and elegant way what Kant's claim that mathematics is synthetic *a priori* really means. Those who, since Frege, have attempted to treat maths as purely analytic have simply failed to appreciate Gödel's incompleteness results, and, in their blind fidelity to classical logic, have refused to stop viewing mathematics through the lens of completeness, despite their claims to the contrary. They are so blinded by the *sheer obviousness* of judgements such as '5 + 7 = 12' that they cannot but see them as analytic, and on this basis they pretend that famous open problems in number theory such as Goldbach's conjecture must be analytic in the same sense, even though we know of no *rule* that could decide the question.

Here's where we return to my claim that novel mathematical proofs are immeasurably harder to discover than they are to check. This all depends on what we mean by 'novel'. We can work out *precisely* how much harder it is to solve a certain mathematical problem than another if we have algorithmic procedures for solving them. However, there are further ways of breaking down problems into more complicated computational complexity classes that enable us to reason about *possible* solutions. The most significant of these are P and NP: deterministic polynomial time and nondeterministic polynomial time. These articulate a precise difference between problems for which there are algorithmic solutions better than randomly exploring some possibility space, and problems for which there are not. As such, the important thing is that if one is confronting a genuinely *open* mathematical problem for which there is no way of *enclosing* the space of possible solutions in advance, then the computational difference is essentially unmeasurable.

What Martin-Löf makes clear is that the content of the concepts we use to *pose* these problems, the inductive definitions of the types that specify ranges of values for variables (e.g., natural numbers, rational numbers, real numbers), and the functions that specify operations on variables of the relevant types (e.g., addition/subtraction and multiplication/division), are precisely what we need to recursively *check* solutions to judgements formulated using them (e.g., '5 + 7 = 12'). Contra Kant, one might say that if the definition of the operation of addition is all you need to *compute* the results of these judgments (e.g., written in lambda calculus notation: (λ sz.s(s(s(s(s(z)))))) (λa bc.b(abc)) (λxy.x(x(x(x(x(x(x(y))))))) reduces to λ xy.x(x(x(x(x(x(x(x(x(x(x(x(y))))))))))))), then this makes them analytic. The difference between solving the problem and checking the solution has collapsed. However, something like Fermat's last theorem is still a synthetic judgment, because of the fact that the concepts sufficient to pose the problem are *insufficient* to solve it, and need to be supplemented by whole new clusters of types, culminating in extravagant abstractions such as elliptic curves and modular forms. This provides a delightful way of understanding the *paracompleteness* of the (intuitionistic) logic of mathematical justification. As we keep adding new types in order to solve problems we were able to pose with the previous ones, we open up whole new ranges of problems we can pose but not yet solve. The answers to our questions keep inviting us to ask new ones.

By contrast, Jean-Yves Girard has been trying to explore the analytic/synthetic distinction in more general terms. I alluded to his earlier work in ludics above, when talking about dialectical interaction. In order to relate Girard's work to Martin-Löf's it's important to briefly mention the

Curry-Howard correspondence, which is a correspondence between type theory and intuitionistic logic. This means that one can view propositions as types whose instances are proofs, and conditionals (if P then Q) as functions that take a proof from one type (P) and return a proof of another type (Q). Martin-Löf built his version of type theory (MLTT) using this correspondence, extending it to include a quantifiers as dependent types. This has gained new significance in recent years, as it is the basis for homotopy type theory (HoTT), a new approach to the foundations of mathematics that provides an alternative to set-theory. It's important to understand that, though all propositions are types, not all types need be propositions. Hence Martin-Löf's inductive definitions of types whose instances are things like natural numbers, rather than proofs.

Girard's work can be seen as an attempt to derive types from more fundamental interactive dynamics, rather than simply defining them. However, this is first and foremost an attempt to see how propositions emerge out of the abstract structure of dialogical interaction, along with the logical operators of linear logic. Girard is interested in more than just propositional types, and it's possible to reconstruct MLTT within the ludics framework. What's interesting about his more recent, explicitly Kantian work on transcendental syntax, in which he aims to study 'the conditions of the possibility of language' is that he has developed a new conception of the computational dynamics out of which types emerge, one that can in some ways more easily be extended beyond the mathematical cases Martin-Löf is describing, to the empirical and practical cases necessary to think about communication and cognition. Ludics was about finding the dynamics hidden in the syntax of proof trees, and transcendental syntax is about finding the dynamics hidden in the syntax of proof nets. This means Girard has made something like the move I recommended in thinking about problem spaces above, from thinking in terms of trees to thinking in terms of graphs.

How does this help us? Because Girard begins to conceive of types as bundles of *tests*, which are no longer restricted to processes like checking proofs. These tests can now involve interaction *with the world and one another*, in addition to interaction with other language users. Types are no longer restricted to *mathematical structures*, but can incorporate those *computational behaviours* involved in interacting with the world, and do so in ways that need neither be linear nor entirely fixed. Precisely the sort of information processing behaviours we discussed when talking about the predictive processing model, with its interacting loops of upstream and downstream signals; loops that extend into the environment as sensation feeds into simulation and simulation feeds into action. The way in which these processes *communicate* thus begins to look like a microcosmic model of linguistic communication, and the tests required to establish and calibrate communication begin look like analytic queries.

This is not something I wish to explore in too much more detail, but I do want to tie up the point: we can see the symmetric compression and decompression involved in communication as something like functionally bisimilar behaviours. The bundles of such behaviours that constitute sortal concepts (*qua* types) enable us to individuate empirical objects (*qua* instances) in ways that are sufficiently similar to talk about them, and condense aspects of behaviour (that we can interactively *test* for) in such a way that we can ascribe properties (*qua* predicates) to them. The high-bandwidth semantics riding the low-bandwidth syntax comes from our ability to *sync* our simulations of the world sufficiently. This almost seems like a trite point: I say 'my cat is a sphynx' and you develop certain expectations that shape your simulation of the world and my cat's role within it. But this is way more interesting because it neither involves idealised representational contents that are merely exchanged (I'm looking at you model theorists), nor a requirement that the neurological correlates of our simulations be innate and isomorphic (I'm looking at you Fodor), nor a structureless play of signifiers where meanings not only bleed into one another, but semiotically haemorrhage in such a way that we can no longer say anything about meaning at all (I'm looking at you Derrida). There are definite data structures here: adapted information schemas that are getting synced up through training and calibration.

We've got a marriage of structure and behaviour organised through symmetries (not unlike that proposed by James Trafford). Moreover, we can see how something like analyticity emerges out of the underlying behavioural bisimilarities. For instance, the basic interactions through which we bundle other tests, the ones that correspond to the *conditions of individuation* through which objects (*qua* behavioural invariants) get picked out and re-identified, are *inherited* in a manner that is expressed by analytic judgments such as 'cats are animals', 'oil is a liquid', and 'I am a person'. This forms provides a logical basis for the perennial distinction between properties that *inhere* in substances and those that are merely *accidental*; a distinction discovered by Aristotle, reworked by Kant, forgotten by Fregeans, and given back to us by Martin-LöF in the form of type definitions (essence), typing judgements (individuation), and the subtle complexities of subtyping (genus/species). There's a deeper story I could tell here about logical and computational dualities, but I've got to stop somewhere. I'll leave you with a last thought. If this hierarchy of relations through which information schemas (data structures) and interactive behaviours (methods) are inherited looks a bit like object-oriented programming to you, there's a reason for that. The irony involved in me describing experience in object-oriented terms is not lost on me.