Hybrid Type Theory:
A Quartet in Four Movements

Carlos Areces Patrick Blackburn Antonia Huertas Mara Manzano

Abstract

This paper sings a song — a song created by bringing together the work of four great names in the history of logic: Hans Reichenbach, Arthur Prior, Richard Montague, and Leon Henkin. Although the work of the first three of these authors have previously been combined, adding the ideas of Leon Henkin is the addition required to make the combination work at the logical level. But the present paper does not focus on the underlying technicalities (these can be found in Areces, Blackburn, Huertas, and Manzano [3]) rather it focusses on the underlying instruments, and the way they work together. We hope the reader will be tempted to sing along.

This paper is about a logic ... and about beauty. It tells the story of an idea born by bringing together the work of four great names in the history of logic.

The ideas of Hans Reichenbach on the importance of temporal reference and those of Arthur Prior on the importance of the internal perspective provided by modal and tense logic were first combined in [5]. In that paper, a simple hybrid logic (namely, propositional nominal tense logic) allowed the insights of Reichenbach and Prior to be integrated. A decade later, in [1], nominals were added to Richard Montague's Intensional Logic to build a richer hybrid logic (nominal intensional logic), a logic capable of capturing the insights of Reichenbach, Prior and Montague in a single system. In this paper we introduce a fourth dimension: Leon Henkin's ideas on completeness for higher order systems. We apply these ideas to nominal intensional logic, which is presented here as a hybrid type theory.

Thus, formally speaking, this paper is about a very expressive logical language, namely simple type theory enhanced with hybrid modal resources. But this paper is not about the formalities, it is about the underlying instruments. First we shall show why we need Prior to play with Reichenbach, and why we need them both to work with Montague. The addition of Henkin then turns the trio into a quartet, and as we shall seem Henkin has a crucial role to play: Whereas the first three members play the themes that weave together the many and varied requirements imposed by natural language semantics, Henkin provides the deeper line that satisfies the demands of logic. But the crucial tools provided by hybrid logic — namely nominals and the $@_i$ operators, the result of Prior's duet with Reichenbach — turn out to play a crucial role in the Henkin model construction. Thus we have not simply added Leon Henkin to the list of maestros that inspired this work —- we have shown how harmoniously they play together. We are sure Newton Da Costa will enjoy our quartet.

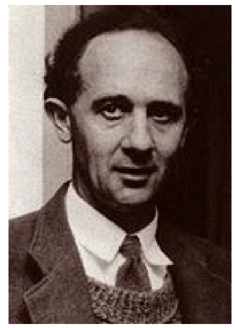
1 Assembling the Players

We'll begin with Jeffrey Tate, chief conductor of the Hamburg Symphony Orchestra, as quoted in *The New Yorker* (April 30, 1990):

The most perfect expression of human behavior is a string quartet.

Bold words! How can we possibly live up to them? Fortunately, we don't have to go it alone. Indeed, after seeing the lineup we have assembled, we feel sure that our audience will agree that our quartet is off to a promising start:

Hans Reichenbach



Arthur Prior

Richard Montague

Leon Henkin

The remainder of this section is a brief introduction to the logical themes this article will improvise over, and to the four maestros who first brought them to life. They are (in order of appearance) Hans Reichenbach (second violin), Arthur Prior (first violin), Richard Montague (viola) and Leon Henkin (cello). In subsequent sections we bring them together: first to play are Reichenbach and Prior (in Section 2), then Montague arrives (in Section 3), and finally (in Section 4) we add the deeper logical tones to the paper.

Reichenbach: Tense and Temporal Reference

In his introductory textbook *Elements of Symbolic Logic* [26], Hans Reichenbach included two chapters about linguistic applications of logic. While most of this work (now over 60 years old) has little relevance to contemporary logical semantics, there is an outstanding exception: his treatment of tense in natural language.

Reichenbach took as his starting point the observation that *"Tenses determine time with reference to the time point of the act of speech"*. But he gave it an interesting twist: instead of representing the semantics of tense with respect to *two* points of time, he represented it with respect to *three*. The first two points are straightforward enough: the *point of speech* (S) is the time at which the sentence is uttered, and the *point of event* (E) is the time at which the event spoken of takes place. But what is his third point, the *point of reference* (R)? To see its role, observe that if we only use the points S and E, there are only three possibilities for the point of event: it must occur either *"before the point of speech"*, *"simultaneously with the point of speech"*, or *"after the point of speech"*. Hence with these two points only three general tenses can be semantically distinguished, namely past, present and future. But tense systems are typically richer than this, and Reichenbach argued that his *point of reference* provided the key needed to explicate their semantics. Let us illustrate his approach with two English examples.

Consider the past perfect sentence *"Alba had sung"*. There is a clear intuition that uttering this evokes some past time *between* the point of speech and the point of event (when the singing occurred). For example, we might use this sentence if we were telling our friends that we traveled to Barcelona to attend Alba's concert, but arrived late, after Alba had already finished; in this case, the contextually determined reference time would be our (sadly belated) arrival at the concert hall. Accordingly, Reichenbach represents the semantics of the English past perfect using the temporal diagram E–R–S. This indicates that the point of event lies to the past of the point of reference, which in turn lies to the past of the point of speech. By way of contrast, consider an utterance of the past tense sentence *"Alba sang"*. Such a sentence would typically be used to locate a singing event that takes place at some contextually determined past time; for example we might say: *"First they cleared the stage, and then Alba sang"*, which locates the singing at the time when the stage was cleared. Accordingly, Reichenbach's temporal diagram here is E,R–S. This indicates that the point of event and the point of reference coincide, and that both lie to the past of the point of speech. Note that in both examples, the point of reference is some past time determined by the context (in the first example, the arrival at the hall, in the second, the stage clearance). Bearing these motivating examples in mind, consider Figure 1, which gives Reichenbach's systematic representation of the semantics of the English tense system:

Diagram	**Tense**	**Example**
E-R-S	Past perfect	*Alba had sung*
E,R-S	Past	*Alba sang*
R-E-S or R-S,E or R-S-E	Future in the past	*Alba would sing*
E-S,R	Present perfect	*Alba has sung*
S,R,E	Present	*Alba sings*
S,R-E	Prospective	*Alba is going to sing*
S-E-R or S,E-R or E-S-R	Future perfect	*Alba will have sung*
S-R,E	Future	*Alba will sing*
S-R-E	Future in the future	*Alba will be going to sing*

Figure 1: Reichenbach's analyses of the tense forms of English.

Reichenbach's analysis has been highly influential. It has shortcomings (for example, his analysis of the present perfect tense simply says that the point of reference corresponds to point of speech, which does scant justice to the subtle semantics of this tense) but many standard accounts of tense (for example, Comrie [10]) still find Reichenbach's ideas a fruitful starting point for further work.

One oddity of Reichenbach's analysis is worth noting: his use of three different diagrams to represent the future in the past and the future perfect tenses. In both cases his diagrams overspecify the temporal location of the point of event E relative to S and R. For the other tenses his diagrams are highly suggestive, but their overspecificity in these two cases suggests that the logic underlying his system has not been properly isolated. We'll return to this point in Section 2 when we motivate hybrid logic, but for now, let's introduce the second member of our quartet.

Prior: Tense Logic

Arthur Prior was a philosopher, best known for his invention of tense logic (see, in particular, his book *Past, Present and Future* [25]). Tense logic is a simple kind of modal logic for reasoning about time. Prior found the modal perspective, in which formulas are evaluated at some particular time, as a natural way of capturing the *temporally-relative* (or *deictic*) manner in which natural language tenses specify temporal information. We live *in* time, and natural language tenses locate events of interest relative to our temporal location (that is, relative to when we speak).

As the central elements of his tense logic, Prior defined the F and P modalities (meaning *"at some future time"*, and *"at some past time"*, respectively). Let us see how they work. Consider the untensed English sentence *"Alba sing"* and suppose that the propositional symbol *alba.sing* is its representation in tense logic. Now consider the expression obtained by prefixing this with the P operator, namely $P\mathit{alba.sing}$. This is true at a time t if and only if Alba does indeed sing at some time t' before t.

Now, this is nice as far as it goes, but (given what we have already learned from Reichenbach) it is clear that it doesn't go far enough. In particular, Prior's representation means that at some completely unspecified past time we have Alba singing, whereas the uttered sentence *"Alba sang"* actually means that at some particular, contextually determined, past time (for example, when they cleared the stage) Alba did in fact sing. Prior's modal representation captures less of the meaning of the past tense than Reichenbach's diagrams; in particular, the reference to a specific past time is simply not there.

Nonetheless, Prior's modal perspective has a lot to offer. For a start, the point of speech concept is *intrinsic* to Prior's tense logic: it is simply the particular time at which we evaluate a formula. The same can be said of the point of event: in Prior's representation, prefixing φ to form $P\varphi$ or $F\varphi$ locates the point of event to the past or future of the point of speech. In short, Reichenbach's point of speech and point of event are *automatically* built into Prior's semantics of tense; this is because of Prior's fundamental decision to work in the setting of modal logic. This decision opens the door to detailed logical analysis, and the subsequent history of tense logic supplied such analysis in abundance.

Summing up, Reichenbach gives us a well-motivated referential theory of tense that lacks a clear logical foundation, whereas Prior we have an elegant logical theory that does justice to point of speech and point of event but is silent of the matter of point of reference. As we shall soon see, moving from orthodox modal logic to the richer setting of hybrid logic will reveal that Reichenbach and Prior are actually singing the same song. But we are jumping too far ahead; it is time to introduce the third member of our quartet.

Montague: Higher-Order Intensional Logic

Richard Montague is one of the most significant figures in the development of logical semantics for natural language. He believed that there was no important theoretical difference between natural languages and artificial logical languages and set out to show this. In his paper *Universal Grammar* [21], he developed a general framework for exploring the syntax and semantics of both types of language. In his best known paper, *The Proper Treatment of*

Quantification in Ordinary English [22], he presented the syntax and semantics of a fragment of English completely rigorously, indeed *algorithmically*.

The algorithmic character of his approach sets Montague's work apart from those of his predecessors (for example, the work of Carnap [9]). Ever since Frege introduced modern logic, the links between logic and natural language have been explored by logicians and philosophers, and the idea that artificial logical languages can throw light on natural languages has been a motivation for many logical developments. Nonetheless, before the work of Montague, the link between logic and natural language was essentially informal: logics displayed mechanisms which might (or might not) throw light on language, but the link was never fully specified. Montague changed that. He showed how (given an explicitly defined grammar for a fragment of English) to define compositional model theoretic interpretations for the sentences the grammar generates. Sometimes this interpretation was direct (the approach taken in his paper *English as a Formal Language* [20]) and sometimes it was indirect. In particular, in *The Proper Treatment of Quantification in Ordinary English* he proceeds by algorithmically translating the sentences of English into his Intensional Logic (IL), and then using the model theoretic semantics of IL to induce a semantics on the original English sentence. The logic IL is a system of higher-order logic which handled intensional notions using built-in systems of modal and tense logic (Montague was familiar with Prior's work, and incorporated it into his system).

Montague's work radically transformed the practice of logical semantics, turning it from a philosophical enterprise to one firmly linked with linguistics and computation (for example, Heim and Kratzer [13], is a linguistics textbook linking Montagovian ideas with orthodox Chomskyan syntax, while Chapter 15 of Jurafsky and Martin [17], the standard introduction of computational linguistics, is a computational approach to semantics based on techniques originating with Montague). While many of his semantical analyses have been superseded, his general approach, and his insistence that natural languages can be modelled *explicitly* using formal tools, remains fundamental. Furthermore, his insistence on the importance of higher-order logic for natural language semantics continues to be taken seriously by semanticists. When a new semantic analyses comes along, a natural place to explore them is in a Montague-inspired version of type theory.

Now, as we have already said, one of the key moves we want to make in this paper is to bring together the ideas of Prior and Reichenbach; hence the natural place to do so is in the presence of Montague.

Henkin: Completeness of Type Theory

We come to the player who completes our quartet. Prior and Reichenbach have played together previously (see Blackburn [5]) as have Prior, Reichenbach and Montague (see Areces and Blackburn [1]). But while these duo and trio were interesting, to gain a truly satisfying synthesis we need the contribution of Leon Henkin. Henkin brings to the quartet a framework for handling higher-order inference. We have motivated the need for combining Prior and Reichenbach in the setting of Montague. But to understand the resulting system logically we need Henkin's ideas. Henkin was a logician best known for the method he used in proving the completeness theorem for several logical calculuses, nowadays called Henkin's method and still in use.

Devising complete inference systems for higher-order logic is difficult. Indeed, at first sight it appears impossible. At the start of his 1950 paper *Completeness in the Theory of Types* [15], Henkin recalls the fundamental problem (noted by Gödel) regarding incompleteness in higher-order logic:

> *...no matter what (recursive) set of axioms are chosen, the system will contain a formula which is valid but not a formal theorem.([15],p.81)*

Why was this? Because Gödel had shown how to construct logically valid sentences which were not formal theorems of the second-order calculus. These counterexamples were all constructed in *standard structures*, that is, models for higher-order logic where:

> *...the individual variables are interpreted as ranging over an (arbitrary) domain of elements while the functional variables of degree n range over all sets of ordered n-tuples of individuals.([15],p.81)*

Gödel proved that first-order logic is complete (there is adequacy between a calculus and semantics) and he also proved that any other logic containing arithmetic, in particular second order logic and type theory, are incomplete. With Gödel's results, the way to higher-order completeness seemed blocked. But Henkin found a path; the key is to abandon the use of standard structures. In [15], he showed that there was a wider class of models in which it was possible to redefine the notion of a valid formula in a way that was both natural and permitted a general completeness result to be proved. Henkin proved that accepting only definable sets and functions when interpreting formulas (no all sets and all functions), the valid formulas coincide with the theorems in a calculus.

Let's sketch what is involved in the second-order case; for a more detailed introduction, see Manzano [19].

In the first place we define the class of standard structures, and thus we obtain the set of second-order validities on that class. We know (thanks to Gödel) that there is no complete calculus for standard validities, since this set is not excursively enumerable. But even though we know that there is no complete calculus in the standard sense, we have certain axioms and rules which we know to be sound and can use these to define a calculus.

Since the set of logical theorems provable in that calculus is a proper subset of the set of standard validities, it is natural to try semantically characterize it. Henkin showed how do this: we need to widen the class of structures, thereby reducing the set of validities (the wider the class, the smaller this set, because to be valid in a wider class of structures a sentence needs to pass more, shall we say, *"quality controls"*). Now, standard second-order structures are those where n-ary relation variables range over $\wp A^n$ (that is, the powerset of the n-ary product of the domain of quantification). The key change is to allow non-standard structures — that is, structures where the relation variables are allowed to range across some *subset* of $\wp A^n$. Actually, we need a little more: if we do not impose any other condition on the universes of a structure, it may well fail to contain certain relations that are definable in the structure using second-order formulas: this means the comprehension schema does not necessarily hold. So we also need to insist that our structures obey certain technical closure conditions; such structures are called *Henkin's general models*. If we use such structures, logical elegance is restored. In particular it turns out that a neat semantical characterization of the set of logical theorems emerges: this set is *precisely* the set of second-order formulas valid on general models.

The idea of general models applies not only to second-order logic but to full type theory. Indeed, using the semantics of general structures, type-theoretical completeness (in both the weak and strong sense), compactness, and Löwenheim-Skolem can be proved in much the same fashion as for first-order logic. In fact, Henkin actually first proved the completeness of type theory with respect to general models and only later found a way to apply a similar method for first-order logic; for a detailed account, see *The Discovery of my Completeness Proofs* [16]. As Henkin explains, his working hypothesis was that Gödel's valid (in standard models) but not provable sentences were really very special, and that by redefining what a valid formula was, these troubled formulas could be banished from consideration. His results triumphantly vindicated his hypothesis. In fact, Henkin in 1950 invented a method for testing completeness generalizable almost to all formal systems.

For our purposes what is important is that Henkin's general models provide a flexible framework in which to approach higher-order completeness. In essence, we want to use his ideas to characterise the logical space created by bringing together the ideas of Reichenbach, Prior and Montague. In Section 4 we will show how this can be done.

2 The Modal Scale

The previous section introduced the four key themes: tense and temporal reference, tense logic, higher-order intensional logic, and completeness in the theory of types. In this section we start to bring them together and explore their interactions. The interactions are, of course, what is crucial. The main point of this paper is not that we have devised some new formalism for logical semantics; such formalisms are ten-a-penny and rarely of much intrinsic interest.[1] Rather, the claim is that the four themes this paper marries fit together in a natural, mutually reinforcing, fashion. As saxophonist Stan 'The Sound' Getz once put it (in Paris, on November 13, 1966):

> *A good quartet is like a good conversation among friends interacting to each others ideas.*

It's time to get this conversation started. We'll start at the high end: what do Reichenbach and Prior have to say to each other?

Let's start by thinking about modal logic. The traditional perspective (largely drawn from philosophy) views modal logic as arising from standard classical logical systems — propositional logic, first-order logic, second-order logic, and so on — via the addition of various non-truth-functional operators, called modalities, and most commonly notated as $\Diamond$ and $\Box$. Such operators are added to enable talk of *intensional concepts*, or to put it another way, to *qualify truth*. Qualifiers explored in traditional modal logic include: necessity, possibility, contingency, obligation, belief, knowledge, and the temporal relations (past, present and future) added by Prior. The traditional perspective makes it abundantly clear that there is not just one modal logic but a whole family of them, as different truth qualifiers typically give rise to different logics.

Contemporary perspectives reinforce this multiplicity. Indeed, they widen it: nowadays modal logics are viewed as general tools for modeling a range of concepts drawn from such disciplines as linguistics, knowledge representation, and computer science (philosophy is no longer the motor driving modal logic; it has been demoted to being merely one of many client disciplines). But the contemporary perspective also provides an answer to the question "What

[1] A notable exception is the language of DRSs, the "box language" characteristic of Hans Kamp's DRT (Discourse Representation Theory); see Kamp [18].

are modal logics?" that is interestingly different from the traditional response. Rather than viewing modal logics as ways of qualifying truth, modal logicians nowadays tend view them as *notations that identify interesting fragments of classical logic* (see Blackburn and van Benthem [8] for a technical introduction to this perspective). In particular, they tend to view modal logics as fragments that inherit the standard logical semantics (in terms of relational models) but with restricted expressive power. According to contemporary views, modal logics are important because they provide interesting compromises between expressive power and inferential complexity.

This perspective has had a valuable side effect: it has opened the door to widespread experimentation with modal syntax and semantics. The traditional perspective made clear that there were a multiplicity of modal logics; the contemporary perspective goes several steps further and actively encourages the modal logician to invent new formalisms to capture new ideas: *"Get out there and experiment with those logics!"* is the new call to arms. One of the simplest (and oldest) examples of this kind of experimentation is *hybrid logic*. A pleasing aspect of hybrid logic is that it can be seen of as a way of bringing together the ideas of Prior and Reichenbach. Hybrid logic will play a key role in what follows, and its central idea (the use of "nominals") will play a key role in building the Henkin-style general models used later in the paper.

Recall that Priorean tense logic (being a modal logic) incorporates the notions of point of speech and point of event automatically. What was missing, as we saw, was a way of capturing *point of reference*. Intriguingly, in the chapter *"Precursors of tense-logic"* in his *Past, Present and Future* [25], Prior dedicates a section to Reichenbach's work. But Prior's discussion is rather superficial. He rejects Reichenbach's scheme, objecting that it does not cover more complicated tenses (such as *"I shall have been going to see John"*) where there are two points of references (S-R2-E-R1), Reading Prior's rather perfunctory criticisms, one senses that Prior simply disliked the non logical nature of Reichenbach's work. This is ironic, since in that same book, Prior introduced the earliest version of hybrid logic, a logic that allows his approach to tense to be beautifully integrated with Reichenbach's, and indeed to extend and clarify Reichenbach's ideas!

How is this done? This key idea is to add a second sort of propositional symbol (usually written i, j, k, and so on), to the propositional symbols of orthodox tense logic (that is, p, q, r, and so on). These new symbols are nowadays called nominals,[2] and they are important because they make it possible to identify instants of time. This is because *each nominal is true at exactly one time in any model*; nominals thus *"name"* the unique time they are true at. Nominals are the central element of hybrid logic, and adding them to tense logic immediately produces a more expressive system called *nominal tense logic* [5]. Let's have a look at this logic in action.

Consider the formula of orthodox (Priorean) tense logic:

$$F(r \wedge p) \wedge \ F(r \wedge q) \rightarrow F(p \wedge q)$$

which says that if there is a future time where both r and p are true together, and a future time where r and q are true together, then there is a future time where p and q are true together. This formulas is clearly *not* true in every model: the future times that witness p (and r) and q (and r) could be different. But now consider the formula of nominal tense logic

[2] Prior usually called them *world propositions*, *world-state propositions*, *world variables*, or *instant propositions*; see Blackburn [6] for a detailed account of Prior's work in this area.

obtained by replacing the propositional symbol r by the nominal i:

$$F(i \wedge p) \wedge F(i \wedge q) \rightarrow F(p \wedge q).$$

This formula is true in every model because the future times witnessing p and witnessing q both make the nominal i true — *and there is only one time that does this.* Thus there *is* a future time where p and q are both true together, namely the time labelled i.

Moreover — although Prior was blissfully unaware of it — his nominals enable us to model Reichenbach's points of reference, and thus to incorporate Reichenbach's insights into modal logic. As an example, consider the formula

$$P(i \wedge P\varphi)$$

stating that there is some time (named i) in the past, and that event φ happened *before that.* But this, of course, is a logical representation of Reichenbach's semantics of the English past perfect, a representation which combines Reichenbach's insight about the role played by temporal reference with Prior's insistence on the privileged role of modal representations. Figure 2 shows how all of Reichenbach's representations can be captured using nominals:

Diagram	**Formula**	**Example**
E-R-S	$P(i \wedge P\varphi)$	*Alba had sung*
E,R-S	$P(i \wedge \varphi)$	*Alba sang*
R-E-S or R-S,E or R-S-E	$P(i \wedge F\varphi)$	*Alba would sing*
E-S,R	$i \wedge P\varphi$	*Alba has sung*
S,R,E	$i \wedge \varphi$	*Alba sings*
S,R-E	$i \wedge F\varphi$	*Alba is going to sing*
S-E-R or S,E-R or E-S-R	$F(i \wedge P\varphi)$	*Alba will have sung*
S-R,E	$F(i \wedge \varphi)$	*Alba will sing*
S-R-E	$F(i \wedge F\varphi)$	*Alba will be going to sing*

Figure 2: Reichenbach's referential analysis of tense using nominals

Note that these representations actually improve on Reichenbach's. For instance, the future-in-the-past now has a single representation: the formula $P(i \wedge F\varphi)$ expresses that there is a reference time i in the past, and that the point of event (where φ is true) is to the future of i. But this is precisely what the future-in-the-past means. This representation avoids the overspecification of the point of event E (relative to S and R) that is forced on Reichenbach by his use of diagrams.

But nominals are only the first step in hybrid logic.[3] With nominals at our disposal, it becomes natural to add new modalities of the form $@_i$, where i is a nominal. The meaning of these new operators is the obvious one: a formula of the form $@_i\varphi$ is true at any point if and only if φ is true *at the point named by the nominal i.* To put is another way: formulas of this form "jump" us to the point named by the nominal, and we evaluate φ there.

The result of enriching a modal logic (like Prior's tense logic) with both nominals and the $@_i$ operators is usually called *basic hybrid logic.* Let's play with this logic a little to see the

[3]And many more steps have been explored in the literature, most of which will not be discussed here; for a detailed account of contemporary hybrid logic, see Areces and ten Cate [4].

sort of expressivity and inferential capabilities it offers us. Zen philosophy provides a perfect starting spot. In *The Book of Perfect Emptiness* Tang de Ying asked Xia Ge: *"Did things exist at the dawn of time?"* And Xia Ge answered: *"If things had not existed at the dawn of time, how could they possibly exist today? By the same token, men in the future could believe that things did not exist today."* As a first step to expressing this argument in basic hybrid logic, let's reformulate it as three premises (α, β, γ) and a conclusion (δ):

α	:=	*If things exist at a given point in time, then at any given previous moment, things must have existed.*
β	:=	*Things exist today.*
γ	:=	*Nothing occurs before the dawn of time.*
δ	:=	*Things existed at the dawn of time.*

Next, let's express α, β, γ and δ in basic hybrid logic. We'll use the propositional symbol q to stand for *"things exist"*, and the nominals d and t to name *"the dawn of time"* and *"today"* respectively. We'll also make use of Prior's H operator: $H\varphi$ means "φ holds at all points in the past", and can be regarded as an abbreviation of $\neg P \neg \varphi$. The symbol $\perp$ (falsum) stands for some always-false proposition (for example, $p \wedge \neg p$). This yields the following premises:

$$\alpha \ := \ q \rightarrow Hq \qquad \beta \ := \ @_t q \qquad \gamma \ := \ @_d H \perp.$$

Note that premise γ reads: *at the dawn of time it holds that at all previous time points $\perp$ is true*; the only way this can happen is if *there are no points of time prior to d*, which is what we needed to express. As for the conclusion, this is simply:

$$\delta \ := \ @_d q.$$

Does the conclusion follow from these premises? Actually, no: a little thought reveals that we need the following general fact: given two instants of time, either they are the same, or one precedes the other. In particular, to drive this argument through, we need to know this property (known as the "trichotomy" property) for the two points of time used in the argument, namely t and d. But this hidden premise is easy to state using the resources of basic hybrid logic:

$$@_t d \ \vee \ @_t Pd \ \vee \ @_d Pt.$$

This tells us that either t and d name the same point, or that at t we have that d lies in the past, or that at d we have that t lies in the past. The last option is swiftly seen to be impossible, and then simple case-by-case reasoning using the first two disjuncts establishes the result. Thus Xia Ge's argument is an enthymeme, and once the hidden premise is located, its correctness can be established.

It is worth reflecting a little more on this proof, for it neatly illustrates an important point about basic hybrid expressivity.

The hidden premise played a key role in the previous argument: note the *form* of the subformulas used to express it. In particular, note that all three disjuncts are of the form $@_i\varphi$ where φ is either a nominal or of the form Pi. More generally, formulas of form $@_i j$, together with $@_i Pj$, $@_i Fj$ or (to use a generic modal notation) $@_i \Diamond j$ are extremely important in hybrid logic.

Why is this? Well, in the case where φ is a nominal, we have a handy way of expressing equality. After all, the formula

$$@_i j$$

asserts that i and j name the same point. To put it another way, $@_i j$ is a modal way of expressing what $i = j$ would express in classical logic. Indeed, it is easy to see that the following basic hybrid formulas, which express the *reflexivity*, *symmetry* and *transitivity* of equality, respectively, are all validities:

$$@_i i \qquad @_i j \to @_j i \qquad @_i j \wedge @_j k \to @_i k$$

We have already remarked that contemporary perspectives on modal logic emphasizes that it is a *fragment* of first-order order logic. We can now be a little more precise: the basic hybrid logic is special because it adds names and equality (normally the province of first-order logic) to the setting of modal logic.[4]

And when the φ (in $@_i\varphi$) is of the form Pi, Fi, or $\Diamond i$? As we saw in Xia Ge's argument, $@_i Pj$ says the point j lies to the past of the point named i. Analogously, $@_i Fj$ says the point j lies to the future of the point named i, while the generic form $@_i \Diamond j$ says the point named i is related (by whatever the relation of interest is: perhaps temporal, perhaps epistemic, perhaps something else) to the point j.

There is a great deal more that could be said (both technical and non-technical) about formulas of the form $@_i j$ and $@_i \Diamond j$. For the present we'll merely remark that such formulas will lie at the heart of the general model construction process outlined in Section 4; Reichenbach and Prior will resonate with Henkin.

3 Of Chocolate and Cembalos

In this section we combine the work of Richard Montague (our viola player) with that of Prior and Reichenbach. As we shall see, this can be straightforwardly done, but for those unfamiliar with contemporary logical semantics for natural language, something of a mystery may still remain. Why adopt Montague's approach to natural language semantics? What is so important about the tools he chose? In particular, why is higher order logic so useful in natural language semantics?

Much the same mystery surrounds the role of the viola in a string quartet: it's the odd one out, the precise nature of its role unclear to outsiders. Indeed, before the eighteenth century, trios consisting of two violins (covering the treble) and a cello (to cope with the bass line), optionally accompanied by cembalo, were common. The "logic" behind this setup is evident — nonetheless it was gradually replaced by the now-standard purely-string-based quartet configuration following the release of Alessandro Scarlatti's *Sonata à Quattro per due Violini, Violetta, e Violoncello senza Cembalo* (Sonata for two violins, viola, and cello, without cembalo).

Viewed from the outside, it can be difficult to explain exactly why Scarlatti's novel quartet lineup gained acceptance. The viola is clearly intended to play a "bridge role" between the treble of the violins and the bass of the cello — but why is it so good? For a start, its range is actually not that different from that of a violin (violas are standardly tuned only a fifth lower than violins, hence their pitch range is much closer to that of a violin than that of a cello).

But we suspect that viola players would have no doubts as to why their beloved instrument was chosen, or what their role in a string quartet is. The viola may be closer in range to

[4] This remark can be made mathematically precise in an elegant way using the notion of *bisimulation*; see Blackburn and van Benthem [8] and Areces and ten Cate [4] for further discussion.

the violin than the cello, but this is offset by its rich-toned sonority. Significantly, the viola's role in music (and indeed, the nature of the viola itself) has shifted more radically than any other instrument as the players explore its subtleties of form and function; the viola slowly but surely grew into its new role, and transformed itself in the process.[5]

It's much like that with Montague's use of higher-order logic in natural language. It *is* hard to explain why type-theoretic logics are so natural for this application. On the face of it, it seems strange to use ω-order logic when the ultimate representation is so often just a formula of first-order logic (perhaps decorated with various kinds of modalities).

But just as the viola play knows why his or her instruments is "just right", so the semanticist knows why Montague's use of higher-order logic is right for natural language. It enables a Fregean explanation of sentence semantics to be given — meaning is compositionally explained in terms of functions and argument, and (with the addition of intensional ideas) a distinction between sense and reference can be drawn. Moreover, the type theoretical mechanism of lambda calculus enables sentences to be glued together in uniform fashion: the motor is functional application accompanied by β-reduction. Moreover (and perhaps most importantly) it is flexible and adaptable. Alternative type theories have been developed and exploited to cope with questions (see Groenendijk and Stokhof [12]), partial information (see Muskens [24]) and the dynamic insights of formalism such as Discourse Representation Theory (see Muskens [11] and de Groote [23, 11]).

So without further ado lets build the ideas of basic hybrid logic into a Montague-style type theory which we will call $\mathcal{HTT}$ (Hybrid Type Theory). The following syntactic definitions follow Montague's treatment closely; the main deviations from his work are the clauses we have added for handling nominals and the introduction of the @ operator.

As usual the set TYPES is defined recursively

$$\mathsf{TYPES} ::= t \mid e \mid \langle a, b\rangle$$

The set ME_a of meaningful expressions of type a is then defined as follows. As our concerns in this paper are logical rather than semantic, we shall add a generic modality $\Diamond$ rather than Prior's F and P operator. This is to emphasize the fact that the logical results obtainable with the help of Henkin's insights are general, applicable to all types of modality, not just temporal ones.

1. Every nominal i is in ME_t
2. Every constant of type a is in ME_a, for any type a
3. Every variable of type a is in ME_a, for any type a
4. If $\alpha \in \mathsf{ME}_a$ and u_b is a variable of type b, then $\lambda u_b \alpha \in \mathsf{ME}_{\langle b,a\rangle}$
5. If $\gamma \in \mathsf{ME}_{\langle b,a\rangle}$ and $\beta \in \mathsf{ME}_b$ then $\gamma\beta \in \mathsf{ME}_a$
6. If α and β are both in ME_a, then $\alpha = \beta \in \mathsf{ME}_t$

[5] The viola's early role in the classical repertoire was often confined to note doubling, either in unison or an octave below. But from these humble beginnings, a varied and substantial viola repertoire emerged by the late twentieth century. Indeed, the instrument played a minor but intriguing role in the progressive rock (John Cale of the Velvet Underground used it in such classic tracks as "Venus in Furs" and "Heroin"). Viola makers continue to experiment (often radically) with the form of the instrument, attempting to improve it ergonomically while retaining its characteristic chocolatey tones.

7. If φ is in ME_t, then $\neg\varphi$ is also in ME_t
8. If φ and ψ are in ME_t , then $\varphi \wedge \psi$ is in ME_t
9. If φ is in ME_t and u_a is a variable of any type a, then $\exists u_a \varphi$ is in ME_t
10. If φ is in ME_t, then $\Diamond\varphi$ is in ME_t
11. If α is in ME_a, then $@_i\alpha \in \mathsf{ME}_a$

The structures used to interpret this language contain a hierarchy of types plus the usual ingredients of standard modal semantics: a universe of worlds and an accessibility relation.

Definition. A (standard) structure for $\mathcal{HTT}$ is a pair $\mathcal{M} = \langle \mathcal{S}, \mathsf{F} \rangle$ such that

$$\mathcal{S} = \langle \langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}, W, R \rangle$$

is a skeleton, where:

1. $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$*, the hierarchy of types, is defined recursively as follows (where a and b are types):*
   $\mathsf{D}_e = A$ *(where $A \neq \varnothing$ is the set of individuals)*
   D_t *is a two element set (the truth values)*
   $\mathsf{D}_{\langle a,b \rangle} = {\mathsf{D}_b}^{\mathsf{D}_a}$ *is the set of all functions from D_a into D_b*
2. *W is the set of worlds, $W \neq \varnothing$*
3. $\mathsf{R} \subseteq \mathsf{W} \times \mathsf{W}$ *is the accessibility relation*

We also have a function giving to each constant in the language its appropriate interpretation in the hierarchy. In particular, F is a function assigning to each nominal a function from W to the set of truth values and to each non-logical constant a function from W to an element in the hierarchy of types of the appropriate type.

That completes the definition. The changes we have made are subtle (as befits the contribution of the viola). For a start, we now have nominals in the presence of modalities. Moreover, we now have the @ operator. And note well: the @ operator is playing a richer role: it can be applied to *any* type. To put it another way, it can now be viewed as a rigidifying device across all types (thereby generalizing an idea first introduced in Blackburn and Marx [7]).

As we shall see, the combination of nominals and @ will lie at the heart of our adaptation of Henkin's completeness-via-general-models method. But before we turn to this, let's play the viola for Alba. Recall the sentence "Alba had sung", sadly uttered after arriving at Barcelona too late. Assume that this sentence can be syntactically analyzed as:

(Alba(had(sung))).

Then we can build the semantic representation for this sentence by first applying HAD to SING (where HAD is the semantic representation of *had* and SING is the semantic representation of the verb *sing*) and then applying ALBA (the semantic representation of *Alba*) to the result. That is, a Fregean analysis requires us to carry out the following sequence of functional applications:

$$(\mathsf{Alba}(\mathsf{had}(\mathsf{sung}))).$$

But what are the required semantic representations? They are easy to express in $\mathcal{HTT}$; we shall write $\Diamond$ as P here to emphasize that we are interested in the past-tense interpretation of the modality:

$$\begin{array}{rcl} \mathsf{ALBA} & \hookrightarrow & \lambda z.z\mathsf{Alba} \\ \mathsf{HAD} & \hookrightarrow & \lambda v \lambda x.P(i \wedge Pv(x)) \\ \mathsf{SING} & \hookrightarrow & \mathsf{sing} \end{array}$$

Hence we build the required representation for the sentence as follows, using β-reduction to glue the pieces together:

$$\begin{array}{rcl} \mathsf{HAD(SING)} & \hookrightarrow & \lambda v \lambda x.P(i \wedge Pv(x))\ \mathsf{sing} \\ & \hookrightarrow & \lambda x.P(i \wedge P\mathsf{sing}(x)) \\ \mathsf{ALBA(HAD(SING))} & \hookrightarrow & \lambda z.z\mathsf{Alba}\ \lambda x.P(i \wedge P\mathsf{sing}(x)) \\ & \hookrightarrow & \lambda x.P(i \wedge P\mathsf{sing}(x))\ \mathsf{Alba} \\ & \hookrightarrow & P(i \wedge P\mathsf{sing}(\mathsf{Alba})) \end{array}$$

See how the music unfolds. We start with Montague-style representations, using the lambdas provided by $\mathcal{HTT}$. We finish at the level of Prior and Reichenbach, modalities and nominals, with Alba singing (alas!) at some time before our arrival in Barcelona. We have added another instrument, but the song remains the same.

4 The richness below

We have an elegant trio, and as far as building representations for sentences of English are concerned we are playing interesting music. But logic is missing, and logic is important. We *don't* just want to build semantic representations, we want to be able to reason with them afterwards. This means that we are going to require a proof system, preferably a complete one.

How can this be done? We've already said that Henkin is the key: his work will carry the logical side, just as the cello carries the responsibility for the bass. But it's not obvious that his ideas fit naturally what has gone before. As Yo-Yo Ma, the celebrated cellist, puts it:

> *One is that you have to take time, lots of time, to let an idea grow from within. The second is that when you sign on to something, there will be issues of trust, deep trust, the way the members of a string quartet have to trust one another.*

In this section we will show in outline why this trust is justified; for a detailed technical account, see [3]. We won't attempt to prove the desired completeness theorem here (indeed we won't even spell out the proof system we are working with, as that would take too much space) but we *will* endeavor to indicate how the various instruments we have introduced lead towards a natural Henkin-style model construction.

For a logic $\mathfrak{L}$ and a suitable class of structures $\mathfrak{K}$, completeness (weak) and soundness states that:

$$\vdash_{\mathfrak{L}} \varphi \quad \text{iff} \quad \models_{\mathfrak{K}} \varphi$$

That is, a formula φ is a logical theorem of logic $\mathfrak{L}$ if and only if φ is valid with respect to the class of structures $\mathfrak{K}$.

We might say that we don't know a logic till we haven't identified its set of valid formulas. Intuitively, we can say that the *logicality* of a given formal language resides in the set of valid sentences. Each structure $\mathcal{A} \in \mathfrak{K}$ selects from the set of all sentences those which are true under this particular interpretation. This set of formulas is usually called the theory of a structure $Th(\mathcal{A})$, and it is characteristic of each structure. But all such theories share a common nucleus which is the set of validities. Because they are true in all structures, these sentences do not refer to particular properties of a particular structure. Hence, the set of validities clearly does not characterise any particular structure.

Does this set characterise anything? The answer is yes, it characterises the logic itself. It represents what the logic has to say about any arbitrary structure. If we are able to 'generate' this set easily, we will have finally capture the *essence* of a logic.

The semantic notion of consequence $\Gamma \models_{\mathfrak{K}} \varphi$ perfectly defines when a formula φ follows from a given set of formulas Γ (it is 'enough' to verify that φ is true in all models where Γ is true); but it does not provide for an 'algorithm' that helps us verify this relation (how do we access the models where Γ is true to see if φ is also true there?) In principle, we need to check every single model. This is when the set of logical theorems, and the notion of completeness, come to our help.

On the other hand, the syntactic notion of consequence $\Gamma \vdash_{\mathfrak{L}} \varphi$ is specified through a calculus that explicitly defines when a formula follows on from others: we can infer or deduce that φ follows in the deductive calculus using the set Γ as hypothesis. In other words, we can establish a chain of inference from the premises in Γ to the conclusion φ. Actually, this operational definition of consequence seems even more adequate and closer to the intuitive notion of inference, given that it reflects the discursive character of the process. The completeness theorem just tells us that both definitions of consequence are equivalent as strong completeness establishes

$$\Gamma \vdash_{\mathfrak{L}} \varphi \ \text{ iff } \ \Gamma \models_{\mathfrak{K}} \varphi$$

In modal logic the classical technique for proving completeness is the canonical model technique. But what about proving completeness for type theory or other higher order systems?

This is where Henkin comes in. As we discussed earlier, Henkin defined what he called *general models* and proved the completeness theorem for type theory [15]. He first proved the following theorem, linking consistency and satisfiability in a general model; the completeness theorem is an immediate corollary:

> *Theorem. If Λ is any consistent set of cwffs there is a general model (in which its domain D_α is denumerable) with respect to which Λ is satisfiable.([15], p.85)*

The proof follows three key steps. In the first one, we build a maximal consistence set extending the original Λ. Maximal consistent sets describe with enormous precision a possible model for themselves and the task in the two following steps is to define such a model. In the second place, we define a congruence relation on formulas (modulo the maximal consistence set), using the deductive calculus. Finally, we define by induction a hierarchy of domains D_α, where the congruence relation just defined, plays a crucial role. It is not difficult to prove that the hierarchy built in this way is a model of the original set Λ. Not surprisingly, as we built the model using the maximal consistence set as our oracle.

But Leon Henkin also proved the completeness of first-order logic [14] by using his new Henkin-proof style (a mix of semantics and syntax put together). That is, he proved the following result:

> *Theorem. If Λ is a set of formulas of S_0 in which no member has any occurrence of a free individual variables, and if Λ is consistent, then Λ is simultaneously satisfiable in a domain of individuals having the same cardinal number as the set of primitive symbols of S_0.([14], p. 162)*

Although completeness theorem for first-order logic was proved in the 1930's, Henkin's proof was considerably simpler.

How did he accomplish it? As in the proof for type theory, a maximal consistent set Γ is built. This is done by the famous "witnessing" construction, ensuring the fulfilment of the following condition: If an existential formula $(\exists x)A$ is in Γ then Γ also contains a witness for A obtained by substituting some constant for each free occurrence of the variable x.

An interpretation is constructed on top of this set. In fact, we take as our domain of interpretation simply the set of individual constants used as witnesses in the previous construction. Next, we assign to each such constant itself as its denotation (alternatively, we build the domain from equivalence classes of constants, in which case each constant denotes the equivalence class of which it is a member).

And now for the pay off: the model construction that proves the completeness of $\mathcal{HTT}$ following Henkin's general conception for type theoretical completeness. But there is a delicious twist — and this is where Prior and Reichenbach can be heard — building in the intensional apparatus is done in a way that mirrors Henkin's model construction for *first-order* logic, with nominals playing the role of witnessing constants, and @ providing us "rigid" structure across all types. Let's go a *little* deeper here... it's too pretty to pass over in silence!

- As we mentioned early, $\mathcal{HTT}$ is incomplete (with respect to standard models) due to its expressive power, but we know from Henkin's approach to completeness in the theory of types that the use of general models provide an escape route. The main notions will be:
  - A *pre-structure* is a structure verifying all the conditions for a standard structure, except for the fullness condition on the domains of the hierarchy of types; it is only required that $\mathsf{D}_{\langle a,b\rangle} \subseteq {\mathsf{D}_b}^{\mathsf{D}_a}$
  - A *general structure for $\mathcal{HTT}$* is a pre-structure closed under interpretation, that is, for any meaningful expression in ME_a, its interpretation is a member of D_a.
- $\mathcal{HTT}$ is not just type theory — it has a modal aspect. But here is where the hybrid apparatus comes to the rescue: the nominals and *rigid terms* play the role of the *constants* that Henkin used for completeness of the first-order logic. The denotation of a rigid term is maintained across the modal worlds. In $\mathcal{HTT}$ rigid terms are obtained using the @ operator on meaningful expressions of various types.

We are also loyal to Henkin's construction of the hierarchy of types [15] and this involves three main steps. In the first place, we define an extension to a maximal consistent set adding more conditions dealing with the new hybrid and modal operators. Afterwards, we define a congruence relation using derivability of equality from the extended maximal consistent set. Finally, the definition of the general structure for $\mathcal{HTT}$ using the maximal consistent set and the congruence relation.

Details of the proof can be found in Areces, Blackburn, Huertas and Manzano [3]).

5 Coda

Meredith Wilson (who wrote the music and lyrics for the 1958 Tony-winning Broadway musical hit *The Music Man*) once remarked that

> *Barbershop quartet singing is four guys tasting the holy essence of four individual mechanisms coming into complete agreement.*

In this paper we have shown how four *logical* mechanisms can be brought together in precisely this fashion. Starting with a simple propositional system, basic hybrid logic, which embodies the insights of Prior and Reichenbach, we moved to a powerful Montague-style framework which provided the combinatoric tools (namely the lambda calculus) needed to build natural language representations. This move from a baby propositional system, to the richness of type theory, was interesting in its own right (in particular, we saw that @ naturally generalized to play a rigidity operator applicable to all types). But it also drastically increased the deductive complexity.

Pleasingly, the ideas of Henkin came to our rescue. For a start, his notion of a general model provided the key conceptual tool needed to solve the problem of completeness. But in a pleasing twist, the nominals and @ operator played a central role in the model construction: the violins and cellos sounded of each other, harmonizing across the octaves. This is not the first time that Henkin has helped in hybrid logic; for example the use of Henkin models was essential to prove the interpolation results in Areces, Blackburn and Marx [2]. But this was work merely in the setting of first-order hybrid logic. This paper provides evidence that Henkin's ideas work with hybrid logic not merely at the first-order level, but all the way up the ω-hierarchy.

6 Acknowledgements

This work was supported by the Spanish government grant FFI2009-09345/FISO.

References

[1] Carlos Areces and Patrick Blackburn. Reichenbach, Prior and Montague: A semantic get-together. In S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb, and J. Woods, editors, *Essays in Honour of Dov Gabbay.* College Publications, 2005.

[2] Carlos Areces, Patrick Blackburn, and Maarten Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic*, 124:287–299, 2003.

[3] Carlos Areces, Patrick Blackurn, Antonia Huertas, and Mara Manzano. Completeness of hybrid type theory. To appear.

[4] Carlos Areces and Balder ten Cate. Hybrid logic. In *Handbook of Modal Logic*, pages 821–868. Elsevier, 2007.

[5] Patrick Blackburn. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101, 1994.

[6] Patrick Blackburn. Arthur Prior and hybrid logic. *Synthese*, 150:329–372, 2006.

[7] Patrick Blackburn and Maarten Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fernmüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002*, pages 38–52. Copenhagen, Denmark, 2002.

[8] Patrick Blackburn, Johan van Benthem, and Frank Wolter. *Handbook of Modal Logic*. Elsevier, 2007.

[9] Rudolph Carnap. *Meaning and Necessity*. The University of Chicago Press, 1947.

[10] Bernard Comrie. *Tense*. Cambridge Univerity Press, 1985.

[11] Philippe de Groote. Towards a montagovian account of dynamics. In M. Gibson and J. Howell, editors, *Semantics and Linguistic Theory (SALT) XVI*, pages 1–16. Cornell University, 2006.

[12] Jeroen Groenendijk and Martin Stokhof. Questions. In *Handbook of Logic in Language*, pages 1055–1124. MIT Press and North Holland, 1997.

[13] Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackkwell, 1998.

[14] Leon Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.

[15] Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15(2):81–91, 1950.

[16] Leon Henkin. The discovery of my completeness proofs. *The Bulletin of Symbolic Logic*, 2(2):127–158, 1996.

[17] Daniel Jurafsky and James Martin. *Speech and Language Processing*. Prentice Hall, 2000.

[18] Hans Kamp and Uwe Reyle. *From Discourse to Logic*. Kluwer, 1993.

[19] Maria Manzano. *Extensions of First Order Logic*. Cambridge University Press, 1996.

[20] Richard Montague. English as a formal language. In Bruno Visentini et al, editor, *Linguaggi nella Società e nella Tecnica*, pages 189–244. Milan, Edizioni di Communità, 1970. Reprinted in [27].

[21] Richard Montague. Universal grammar. *Theoria*, 36:373–398, 1970. Reprinted in [27].

[22] Richard Montague. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, 1973. Reprinted in [27].

[23] Reinhard Muskens. Combining Montague Semantics and discourse representation. *Linguistics and Philosophy*, 19:143–186, 1996.

[24] Reinhard Muskens. *Meaning and Partiality.* CSLI Publications, 1996.

[25] Arthur Prior. *Past, Present and Future.* Clarendon Press, 1967.

[26] Hans Reichenbach. *Elements of Symbolic Logic.* Random House, 1947.

[27] Richmond Thomason, editor. *Formal Philosophy. Selected Papers by Richard Montague.* Yale University Press, 1974.