ANNUAL REVIEWS

Annual Review of Linguistics

Types and Type Theories in Natural Language Analysis

Peter R. Sutton

Department of Translation and Language Sciences, Universitat Pompeu Fabra, Barcelona, Spain; email: peterroger.sutton@upf.edu

Annu. Rev. Linguist. 2024. 10:107–26

First published as a Review in Advance on October 10, 2023

The *Annual Review of Linguistics* is online at linguistics.annualreviews.org

https://doi.org/10.1146/annurev-linguistics-031422-113929

Keywords

type theory, semantics, hyperintensionality, theories of propositions

Abstract

This article reviews the set of possible paths from a semantics based on Simple Type Theories (STTs) toward one based on Rich Type Theories (RTTs) and the motivations behind the move from one to the other. The main elements of this review are threefold. First, it provides a systematic overview of different STTs, including options for what to include as members of the set of basic types, and whether to assume type constructors additional to the one for constructing functional types. Second, this review discusses the main differences between STTs and RTTs, namely, that in RTTs but not in STTs, types are part of the object language. That is, one can refer to and reason with and about types. In turn, this makes available an alternative account of propositions to the one assumed in semantics in the Frege–Church–Montague tradition: Instead of being characterized as sets of possible worlds, propositions can be treated themselves as types, that is, as structured semantic objects. Third and finally, this review provides an outline of some of the main applications of RTTs, including hyperintensionality, quantification, anaphora, polysemy, and modification.

1. INTRODUCTION

Type e: the type for physical entities

Type t: the type for truth values

Type s: the type for possible worlds

TY_2: an extensional theory of types with two sorts of "individuals," e.g., e and s, in addition to the truth value type t

Since the foundation of Montague's semantic program (see, e.g., Montague 1970, 1973), much work on formal semantics has been conducted within a theory grounded upon modal predicate logic enriched with a system of simple types and the simply typed λ-calculus (Church 1940), yielding a higher-order extension of predicate logic in which one can abstract and quantify over variables not just for individuals but also for other types such as predicates and propositions. Such a Simple Type Theory (STT) assumes a set of basic types [ι and o for Church; e, t for Montague; and e, t, and s for Gallin's (1975) TY_2] and a highly constrained set of type constructors (functional types for Church and Gallin, and an additional intensional type constructor for Montague). The role of types in semantic theory based on a simple theory of types is twofold (Kohlhase 1992). First, types prevent certain paradoxes that can arise, for instance, when a function may apply to itself (e.g., Curry's paradox; see Section 3.2). Second, types provide a means of sorting entities and functions into physical objects, first-order predicates, two-place relations, first-order properties, propositions, and so forth—the distinctions between which are fundamental to compositional semantic analyses of natural languages.

Combined with a theory of syntax, such as categorial grammar in the case of Montague, this formal foundation has yielded many advancements in the field. However, arguments have also been made for an extension of this type system to capture new data. Basic types beyond type e, type t, and type s have been proposed, such as type d for degrees and type v for eventualities. However, it has also been proposed that only a single basic type could be assumed (Partee 2007, Liefke 2014, Liefke & Werning 2018) and that the aforementioned two roles for types can be divided between types (to exclude paradoxes) and sorts (to sort entities). Type constructors additional to the one for functional types have also been argued for—for instance, product types (Rothstein 2010; Sutton & Filip 2018, 2020; Windhearn 2021) and dot types (e.g., Asher 2011, Asher & Pustejovsky 2013).

However, more substantial departures from Church–Montague simply typed semantics have also been thoroughly explored. These Rich Type Theories (RTTs) were born out of the intuitionistic and proof-theoretic traditions in pioneering work by Ranta (1994) based on Martin-Löf's (1984) type theory, developments of which include works by, for instance, Bekki (2014) and Grudzińska & Zawadowski (2014, 2019, 2020). Additional model-theoretic semantics based on Martin-Löf's type theory have been developed, primarily in the form of Type Theory with Records (TTR; Cooper 2012, 2023). Modern Type Theories (MTTs; see, e.g., Chatzikyriakidis & Luo 2013, 2017, 2020) have been argued to be both model- and proof-theoretic. What primarily sets RTTs aside from simple extensions of STTs is that types feature not only as metalanguage descriptions of object language expressions but as semantic objects themselves. Thus, just as one can have arbitrarily complex types generated inductively in STTs, one can have arbitrarily complex and structured semantic entities constructed in RTTs. One main consequence of this is that it facilitates an alternative theory of propositions, based on the Curry–Howard correspondence, known as the "propositions-as-types hypothesis" (Curry & Feys 1958, Howard 1980). Instead of expressions of type $\langle s, t\rangle$ denoting sets of possible worlds familiar from Montague semantics, propositions can be modeled as (inherently intensional) types (i.e., structured entities).[1]

This article, instead of being structured chronologically, is structured in terms of the set of possible paths from STTs toward RTTs, in relation to both the assumptions made (or dropped) and the motivations behind them (schematically represented in **Figure 1**). Section 2 outlines the

[1]For a related discussion of Frege–Church propositions as assumed in Montague semantics and Russellian propositions, readers are referred to the article in this volume by Liefke (2024). Arguably, the treatment of propositions as types follows more in the tradition of a form of the latter, qua propositions being structured.

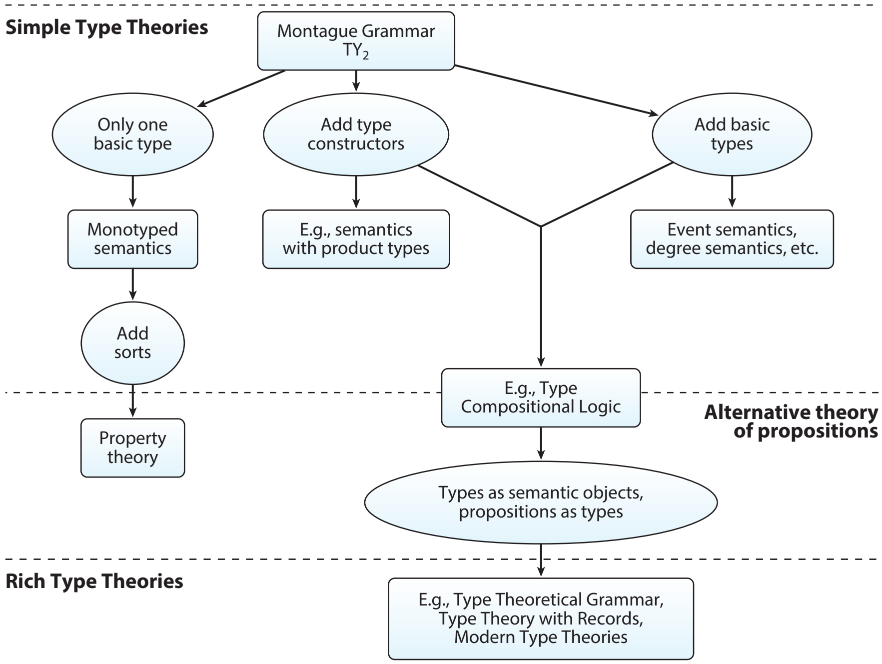

Figure 1

The path from Simple Type Theories to Rich Type Theories with additional annotations regarding a switch to a more hyperintensionalized theory of propositions. Round nodes detail theoretical decisions; rectangular nodes provide examples of theories.

basis of Church–Montague type theory. Section 3 gives a brief overview of the principal ways in which the set of basic types has been either extended, reduced, or supplemented while still maintaining an STT. Similarly, Section 4 reviews conservative STT extensions via the assumption of at least one extra type constructor. Section 5 discusses what sets RTTs apart from STTs. Finally, Section 6 provides an overview of the different applications of RTTs.

Basic type: a type taken as basic from which other types can be constructed or inductively defined

Type constructor: a means, typically inductively defined, of defining complex types from other types and sometimes also from entities of some type

Functional type: a type of function that maps entities of one type to entities of another or the same type

2. SYSTEMS OF SIMPLE TYPES

STT, formalized with the simply typed λ-calculus (Church 1940), underpins much work done in semantics to this day. In definition 1, clause a, **BasTyp** sets the set of basic types, and clause b defines a type constructor that constructs functional types from basic types:

(1) **Types**

From a nonempty set **BasTyp** of basic types, the set **Typ** of types is the smallest set such that:

a. $\mathbf{BasTyp} \subseteq \mathbf{Typ}$

b. $\langle \sigma, \tau \rangle \in \mathbf{Typ}$ if $\sigma, \tau \in \mathbf{Typ}$ (Functional types)[2]

(Carpenter 1997, p. 40)

For instance, Montague (1970, 1973) assumes $\mathbf{BasTyp} = \{e, t\}$ and adds an intensional type constructor clause that $\langle s, \tau \rangle \in \mathbf{Typ}$ if $\tau \in \mathbf{Typ}$. In this case, s is not a (basic) type but rather a

[2]Alternative notations for functional types, the type of functions that take entities of type τ into entities of type σ, include $(\sigma \rightarrow \tau)$ and $(\sigma)\tau$.

pseudotype that appears only as part of a complex (functional) type. The motivation for this was that *s* should only feature as part of expressing the intension of a formula. Gallin's (1975) "extensionalisation" of Montague's Intensional Logic (IL), TY_2, adds *s* as a basic type for world–time pairs. (So intensions are expressed within TY_2 by λ-abstracting over variables of type *s*.) In TY_2, Montague's extra clause is not needed since types featuring *s* are formed via clause b in definition 1. Note that there are more types in TY_2 than in Montague's semantics. For instance, via definition 1, $\langle e, s\rangle$ is a type in TY_2 but not in Montague semantics, in which all intensional types must be of the form $\langle s, \tau\rangle$.

One effect of adopting *s* as a basic type, as in TY_2, is that it makes the model theory simpler. To take an example, in Montagovian semantics, we need to introduce and interpret a necessity operator $\Box$ that universally quantifies over the set of possible worlds at which a formula is evaluated. In Gallin's system, we can simply quantify over a variable over worlds in the object language formula: $\forall w.\phi$. TY_2 also obviates the need for Montague's $^\wedge$ (cap) and $^\vee$ (cup) operators that map extensional type formulas to intensional type formulas and vice versa, since intensions are encoded directly via λ-abstractions over worlds.

Semantic analyses based on an STT have been provided for a broad range of phenomena, including but by no means limited to quantification, modification, attitudes, modal operators, and tense. However, reasons have been put forward for extending and/or modifying the system of types in definition 1. The strategy, taken in this article, for examining conservative and nonconservative divergences from definition 1 is given in item 2 with some of the key theoretical choices outlined in **Figure 1**:

(2) Ways of diverging from Simple Type Theory
  a. Add more members to **BasTyp** in definition 1, clause a (see Section 3).
  b. Add at least one type constructor clause to definition 1 (see Section 4).
  c. Include types as semantic objects in the object language, and interpret propositions as types (see Sections 5 and 6).

3. ADDITIONAL AND ALTERNATIVE BASIC TYPES

One way of departing from the system of types in definition 1 while still retaining an STT is to assume more basic types than *e* and *t* (and *s*). Some such proposals are reviewed in Section 3.1, including a basic type for eventualities and a basic type for degrees. Alternatively, one can reduce the number of basic types. Section 3.2 outlines two such approaches: property theory, which has one basic type *e* (and a set of sorts), and monotyped approaches, which propose one basic type that is distinct from *e*, *t*, and *s*.

3.1. Additional Basic Types

The expressivity of a semantics based on definition 1 has been argued to be insufficient for capturing a range of phenomena. For instance, the distribution of gradable adjectives and modifiers of degree (e.g., *many*, *much*) has motivated the addition of type *d* for degrees (see Seuren 1973, Cresswell 1977, Klein 1980, Kennedy & McNally 2005, among many others).[3]

Another extension to the set of basic types is within neo-Davidsonian event semantics (e.g., Landman 2000, Parsons 1990), which add the basic types *v* for eventualities (including states,

[3]As a historical note, the explicit addition of a basic type *d* was not made in initial work. For instance, Cresswell (1977) treats gradable adjectives (in predicate position) of type $\langle e, \langle e, t\rangle\rangle$, but where the second argument is restricted to be a special sort of complex entity. For *tall*, for instance, this complex entity is a pair of a measurement (the distance between an object's extremities) and a partial ordering relation.

processes, and events) and i for time intervals. Some motivations for this include semantically distinguishing the interpretations of VPs from those of common nouns, and interpreting adverbials as modifiers of eventualities. Aside from those of degrees and eventualities, the adoption of extra basic types is ubiquitous, including, for instance, kinds/concepts (Krifka 1995), numbers (e.g., Krifka 1989), and roles (Zobel 2017).

Higher-order logic: a logic that allows for quantification and binding of variables not only of type e

The much greater expansion of the set of basic types—for instance, allowing a basic type corresponding to every common noun—is an assumption found in some RTTs. This will be discussed in Section 5.

3.2. Monotyped Theories and Sorts

An alternative to increasing the number of basic types is to decrease them. Here, I discuss three such proposals. First, however, it will be helpful to distinguish types from sorts (see, e.g., Kohlhase 1992, 1994). Traditionally, types perform two roles: avoiding paradoxes (e.g., Curry's paradox)[4] and marking conceptual distinctions between entities. However, these roles can be separated such that the former is performed by types and the latter by sorts (Kohlhase 1994, chapter 3). In other words, one can have a set of basic types, minimally a single type, τ, and functional types inductively defined on this type, but also distinguish between sorts within that type to whatever degree of granularity is required.

One example of such a strategy is exemplified in property theory as presented by Chierchia & Turner (1988), in which one basic type e is assumed. Functional types can then be inductively defined on e; however, Chierchia & Turner (1988) allow for only one complex type $\langle e, e\rangle$ (and furthermore prohibit variables of this type). Within type e are distinguished three sorts: basic individuals u, nominalized functions nf, and information units i. In addition, a $^{\cup}$ operator type-shifts expressions of sort nf to functions of type $\langle e, e\rangle$, and a $^{\cap}$ operator "nominalizes" functions of type $\langle e, e\rangle$ into entities of sort nf. One difference between types and sorts is that types and type constructors are typically employed, albeit sometimes in a limited way, to construct complex types. This property-theoretic approach allows for a first-order logic (as opposed to a higher-order logic in the case of Montague semantics) while maintaining some of the advantages of types, performed instead by sorts, such as encoding selectional restrictions. To take an example, in a logic with a single basic type, σ, two functions may both apply to entities of type σ but be restricted to apply to entities of different sorts (e.g., one to physical entities and the other to eventualities). Notably, this approach endorses, and is indeed motivated by, providing an alternative to the Frege–Montague theory of propositions as sets of worlds due to problems with hyperintensionality. In brief, a problem with a possible-worlds-based definition of properties is that two properties may denote the same set at all possible worlds but not, intuitively, be the same property (see Section 6.1 for more discussion). Instead, properties in property theory (whether nominalized functions or functions of type $\langle e, e\rangle$) are inherently intensional, and propositions (of type i) inherit this intensionality.

Relatedly, frame semantics inspired by Barsalou's (1992) conception of a frame (see, e.g., Petersen 2015, Löbner 2015) can also be understood as assuming a simple monotyped type theory with a hierarchy of sorts dividing up the basic entity type. Petersen (2015) refers to this as a "type" hierarchy, but in the context of this review, it is better described as a sort hierarchy since, for

[4] The origins of the paradox are attributed to Curry (1942) and Löb (1945). Suppose there is a function f such that $f(f)$ is the proposition $f(f) \rightarrow p$ [that p is true if f(f) is true]. $f(f)$ cannot be false, since that would mean $f(f) \rightarrow p$ is false and so $f(f)$ is true (a contradiction). Therefore, $f(f)$ is true and so p is true, because $f(f) \rightarrow p$ is true. But that means we can prove the truth of any formula that we substitute for p, even one that is false.

example, no type constructors are assumed for the types in these hierarchies. Unlike in property theory, the number of sorts in frame theory is not highly constrained. Frames can be represented as graphs with typed nodes, each of which may have a specific value of type e, and attributes (the edges in the graph), each of which is of type $\langle e, e \rangle$. So, for instance, an entity of sort **cat** can be mapped via an attribute FUR to an entity of sort **fur** (i.e., a part of the cat), and the entity of sort **fur** can be mapped via an attribute COLOR to an entity of sort **black**. These types and sorts are in a hierarchy in the sense that, for instance, **cat** would be below **mammal**.

Product types: types for expressions that are ordered pairs of entities and expressions

Finally, arguments for a monotyped theory have also been put forward on the basis of different data—for instance, the observations that determiner phrases (DPs) and complementizer phrases (CPs) share distributional properties and can coordinate (Partee 2007, Liefke 2014, Liefke & Werning 2018):

(3a) $[_{\text{DP}}$Bill] destroyed his friendship with John.
(3b) $[_{\text{CP}}$That Bill suspected John of courting Pat] destroyed his friendship with John.
(Liefke & Werning 2018, p. 646)

(4) Pat remembered $[[_{\text{DP}}$ Bill] and $[_{\text{CP}}$that he was waiting for her]].
(Liefke & Werning 2018, p. 647)

In Liefke's (2014) work, for instance, a single-type semantics is given that assumes a single basic type o, which does not differentiate between Montague's e and $\langle s, t \rangle$. Thus, in the above examples, the interpretations of *Bill*, *(that) he was waiting for her*, and *Bill destroyed his friendship with John* are all of type o. Liefke & Werning (2018) provide an overview of arguments for and rebuttals of objections against single-type semantics (for further discussion, see Liefke 2024).

4. ADDITIONAL TYPE CONSTRUCTORS

In this section, we turn to an alternative means of expanding an STT via the addition of type constructors rather than the addition of basic types. To provide examples, work using product- and dot-type constructors is summarized.

4.1. Product Types

A type constructor that is commonly assumed in programming languages based on the simply typed λ-calculus, but that is more rarely assumed (explicitly) in natural language semantics, is the product type constructor, the type of ordered tuples of expressions:

(5) Definition 1 and:
c. $\sigma \times \tau \in \mathbf{Typ}$ if $\sigma, \tau \in \mathbf{Typ}$ (Product types)
(Carpenter 1997, p. 65)

Product types enable a simple method of adding an extra layer of structure in one's semantic theory and have been used for the semantics of nouns and NPs to record information about individuation criteria and countability (e.g., Rothstein 2010, Gotham 2014, Sutton & Filip 2016b) and to characterize structured propositions from which sets of alternatives can be derived (Windhearn 2021).[5] This structure can be unpacked via projection functions, which access the first and second

[5] It should be noted that product types are implicit even in Gallin's TY_2 semantics insofar as entities of type s are world–time pairs, which could be made explicit as a product type of worlds of type w and times of type t yielding entities of type $w \times t$.

projections of a tuple:

(6) If $\alpha = \langle \beta, \gamma \rangle : \sigma \times \tau$, then:
  a. $\pi_1(\alpha) = \beta : \sigma$
  b. $\pi_2(\alpha) = \gamma : \tau$

Dot type: a type for an expression that denotes something with different aspects (e.g., the physical and informational aspects for *book*)

The semantics of countability is one area in which product types have been employed relatively widely. In work by Rothstein (2010, 2017), grammatical counting is the counting of context-indexed entities, and so count noun extensions are of type $\langle e \times k, t \rangle$, with k a type for contexts. In fact, k can be seen, rather than as a basic type, as a special case of $\langle e, t \rangle$. That is, expressions of type k are (in default cases, disjoint) subsets of the domain of type e. Count nouns such as *fence* are interpreted as functions from entity–context pairs. The type $\langle e, t \rangle$ extension of *fence* is the intersection of the set *FENCE* (the set of all minimal fence entities and mereological sums thereof) with k, resulting in a set of entities countable as fences relative to k. This mechanism is used to explain how, for instance, some fencing around a square field can count as one fence in some contexts and as four fences in others. In a similar vein, Sutton & Filip (2016a, 2020) and Filip & Sutton (2017), among others, make explicit the use of product types that are implicit in work by Landman (2011, 2016) to account for the count/mass distinction within a bipartite approach to the lexicon. Relative to a context, common nouns denote functions from entities to a pair of a proposition and a set of entities, where the set of entities defines what is countable in the extension of the noun. This is used to articulate the difference between cointensional nouns that differ in countability such as *furniture* (mass) and *huuonekalu-t* ('item-s of furniture,' Finnish, count).

Relatedly, Gotham (2014, 2017, 2021) treats common nouns as denoting pairs of extensions and individuation criteria to account for "double distinctness" interpretations of copredication sentences such as the following, which are argued to require not only that there are (at least) three physically distinct books but furthermore that there are three distinct informational contents:

(7) Three informative books are heavy.
(Gotham 2014, p. 334)

A comparable bipartite analysis of copredication is articulated within MTTs in work by Chatzikyriakidis & Luo (2020). MTT, a variety of RTT, is discussed further in Section 5.

4.2. Dot Types

Building on earlier work on the generative lexicon (Pustejovsky 1994, 1995) that posits dot objects, polysemy and copredication have motivated the positing of a dot type constructor (e.g., Asher 2011, Asher & Pustejovsky 2013, Pustejovsky 2013, Chatzikyriakidis & Luo 2015), which in the simplest terms can be seen as a conservative extension to an STT:

(8) Definition 1 and:
  c. $\sigma \bullet \tau \in \textbf{Typ}$ if $\sigma, \tau \in \textbf{Typ}$ (Dot types)

The intuitive motivation behind dot types is to capture the idea that the denotations of polysemous nouns such as *book* have different aspects (Pustejovsky 1994, 1995, 2013; Asher 2011; Asher & Pustejovsky 2013; Chatzikyriakidis & Luo 2015). That is, we can conceive of a book as a book-qua-physical-object or as a book-qua-informational-contents. For instance, for two types i (for informational entity) and p (for physical entity), *the book* would denote an entity of type $p \bullet i$, namely, an entity that affords a physical and informational conceptualization. As such, typically, the introduction of dot types coincides with an expansion of the set of basic types. For instance, Asher (2011) assumes, in addition to a general entity type e, primitive types such as p and i. Importantly, Asher (2011) also develops an account of subtyping that also applies to complex (e.g., functional) types. We return to a discussion of subtyping in Section 5.5.

The status of Type Compositional Logic (TCL) with respect to whether it counts as an STT is somewhat dependent on its implementation. If characterized as in the clause in item 8, along with clauses for other type constructors [Asher (2011) also assumes presuppositional, disjunctive (i.e., join), and quantificational type constructors], one could see the introduction of dot types as a conservative extension to a basic STT. However, Asher (2011) proposes an implementation of this theory within category theory (a discussion of which is outside the scope of this article) and also states that the theory could be provided within an RTT.[6] Both a category-theoretic implementation of TCL and an RTT implementation of TCL would constitute alternative views of propositions to one based on possible worlds, a topic we return to below in Section 5.3. For this reason, TCL's status as involving an alternative theory of propositions is underspecified in **Figure 1**.

5. FROM SIMPLE TYPE THEORIES TO RICH TYPE THEORIES

The move from STTs to RTTs is summarized succinctly by Ranta (1994, p. 18) as consisting primarily of two departures from STTs:

> [T]he syntax of predicate calculus does not make explicit everything that is semantically significant. We must look into the metalevel to reconstruct what has been left implicit. Having made such changes in the syntax of predicate calculus. . .Full Type Theory is then obtained by adding the propositions as types principle.

So, we may allow types to feature in the object language and then characterize propositions as types. The former is intrinsically linked to the notion of a judgment in RTTs; the latter constitutes an alternative to a possible-worlds-based characterization of propositions. I take these in turn. Subsequently, in Section 5.4, I briefly characterize two ways in which RTTs have been implemented, one which adds relatively few type constructors, but greatly expands the set of basic types, the other which keeps the number of basic types low, but greatly increases the number of type constructors.

5.1. Types as Object Language Entities and Reasoning in Context

As discussed by Ranta (1994), one thing that is made explicit in RTTs is whether types feature explicitly in the object language. For instance, take a standard definition of the existential quantifier (a clause in the definition of an interpretation function):

(9) If $\phi \in ME_t$ and u is in Var_a, then $[\![\exists u\phi]\!]^{M,g} = 1$ iff for some e in D_a, $[\![\exists u\phi]\!]^{M,g^e_u} = 1$ (Dowty et al. 1981, p. 92)

In the clause in item 9, types feature in the metalanguage as subscripts on sets. However, this means that although we are implicitly assuming that agents make judgments or assumptions regarding what is or is not, say, an individual (where such judgments are implicitly reflected in the model M as, say, what counts as part of the domain of type e), such a judgment cannot be expressed transparently in the object language of an STT.

A principal way in which types feature in the object languages of RTTs is by way of judgments. Canonically, a judgment contains some entity (understood broadly) on the left of a colon, and a

[6]In relation to RTT implementations of dot types to account for polysemy and copredication phenomena, see Luo (2012), Chatzikyriakidis & Luo (2015), and further discussion in Section 6.3.

type on the right. For instance, $a : e$ would be the judgment that the individual a is of the type e (this introduces an element into the type of e). As an example, $felix : Cat$ would be the judgment that some entity $felix$ is of the type Cat. Such judgments can also be specified relative to particular agents and contexts; for instance, given a context Γ in which agent A has made the above judgment, they make the judgment that $felix$ is feline, which can be represented in the following notation:

(10) $\Gamma, felix : Cat \vdash_A felix : Feline$

Permitting types in the object language allows one to reason explicitly about the types that we assume (i.e., about what is of a type given some other type judgments). The proof theory of Ranta's (1994) RTT includes type-theoretic variants of introduction and elimination rules (i.e., type introduction rules and type elimination rules). In addition, to say explicitly how types are constructed, an object-language equivalent of the STT recursive definitions of types is given in the form of type formation rules. As indicated above, judgments can occur in a context (i.e., be conditional on other type judgments). For instance, the following are the formation, introduction, and elimination rules for function types given by Chatzikyriakidis & Luo (2020, p. 9), where $\Gamma \vdash A\ type$ means that A is a type in the context Γ:

Type formation rule: a rule governing the formation of types

Type introduction rule: a rule governing the introduction of a term of the relevant type

Type elimination rule: a rule governing the elimination of a term of the relevant type

Dependent type: a type that depends on some value [e.g., if $Mother\_of(billie)$ is a type, it depends on the value $billie$]

$$\frac{\Gamma \vdash A\ type \quad \Gamma \vdash B\ type}{\Gamma \vdash A \to B\ type} \qquad \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x : A.b : A \to B} \qquad \frac{\Gamma \vdash f : A \to B \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B}$$

Formation Rule **Introduction Rule** **Elimination Rule**

The formation rule says that one can form a function type from any two types. The introduction and elimination rules allow us to make judgments about what is of a particular type. The introduction rule states that if $b : B$ in a context where $x : A$, then one can infer a function of type $A \to B$ from entities of type A to b. The functional type elimination rule equates to function application.

5.2. Dependent Types

An additional feature of making types explicit is that one can characterize dependent types—types that depend on some value. Take relational predicates such as *mother of*. In STT semantics, formulas such as $Mother\_of(alex)$ and $Mother\_of(billie)$ are interpreted as individuals, for instance, *charlie* (relative to some possible world). There is a sense, therefore, that the interpretation of the formulas $Mother\_of(alex)$ and $Mother\_of(billie)$ depend, respectively, on the interpretations of *alex* and *billie*. In RTT semantics, we have structured types, and entities can be of some type or not. For instance, $Mother\_of(alex)$ and $Mother\_of(billie)$ would be types such that it might be that *charlie* is of type $Mother\_of(alex)$ but not of type $Mother\_of(billie)$. Now, we can think of $Mother\_of$ as a type constructor such that the type $Mother\_of(alex)$ depends on the value *alex*, and the type $Mother\_of(billie)$ depends on the value *billie*.

In Ranta's system and many other developments, dependent types are used to characterize existential quantification via the construction of Σ-types (see also Section 6.2). The Σ-type constructor allows one to construct a type under certain conditions, namely, that if in a context in which something is of type A, we can infer that there is something of a type B, where the type B depends on the entity of type A, then $\Sigma x : A.B(x)$ is a type, where it is pairs of entities that are of this type. For example, the type in example 11a is the type of pairs (a, b), where a is of type Dog, and b is of type $Bark(a)$. In this sense, the second projection of each pair may be of a type that depends on the first projection. Using projection functions, an example of an instance of Σ-elimination is given in example 11b, in a context in which some pair is such that the first member of the pair is of type *Dog*, and the second member of the pair is of the relevant dependent barking type, and thus we can infer that there is a barking dog:

Propositions as types: a theory of propositions in which propositions are types: For each proposition, there is an inherently intensional type

(11a) $\Sigma x : Dog.Bark(x)$
(11b) $\Gamma, p : \Sigma x : Dog.Bark(x) \vdash \pi_2(p) : Bark(\pi_1(p))$

This reasoning pattern echoes that of ∃-elimination [$\exists x.P(x) \vdash P(a)$, for an arbitrary object a] except that the type that models the VP depends on the restrictor.

Ranta (1994) also uses Π-types, a type of function, to model universal quantification. Chatzikyriakidis & Luo (2020) use Π-types as dependent function types but maintain a distinct type for universal quantification. For instance, $\Pi x : Cat.Purr(x)$ is a type of functions from entities a of type *Cat* to something of type $Purr(a)$. If situations can be of type $Cat(x)$ for some value of x, a function from cats to situations in which that cat purrs will be of type $\Pi x : Cat.Purr(x)$.

5.3. Propositions as Types

In model-theoretic STT semantics, the recursive definitions in an interpretation function and the definitions of well-formed expressions allow for the compositional building of arbitrarily complex expressions of any type that are interpreted in set-theoretic terms.[7] For instance, $\lambda w.\exists x.Cat_w(x) \wedge (Sit_w(x) \vee Purr_w(x))$ is a complex expression that is interpreted as a set of worlds. In RTTs, by contrast, we compositionally build/construct—not expressions of some type, but types themselves. For instance, $\Sigma x : Cat.(Sit(x) \vee Purr(x))$ is a type, and, in this case, a pair of things may be of this type, that is, the type has truth-makers (is true if something is of this type). A natural move therefore is to model propositions as types, following the Curry–Howard Correspondence (Curry & Feys 1958, Howard 1980). Thus, we have a structured object, $\Sigma x : Cat.(Sit(x) \vee Purr(x))$, that can fill the role of propositions in RTTs instead of the unstructured sets of worlds in STTs. For the case in hand, suppose that $a : Cat$ and a situation s is of type $Sit(a) \vee Purr(a)$; then, (a, s) is of type $\Sigma x : Cat.(Sit(x) \vee Purr(x))$, and so $\Sigma x : Cat.(Sit(x) \vee Purr(x))$ is true insofar as it has a witness (i.e., there is something of this type).

Importantly, types can be seen as inherently intensional (see Section 6.1). As we have seen, propositional types, unlike Montagovian propositions, are structured, and their individuation conditions can be tied to this structure (or to the way in which it is constructed). That is, two types could have the same witnesses and still be distinguishable (for discussion, see Cooper 2023). Therefore, as we have seen with property theory above, modifying one's type theory can facilitate alternative views of propositions that can be used to address issues of hyperintensionality (see Section 6.1).

5.4. Two Sorts of Implementations of Rich Type Theories

There are two principal ways in which RTTs have been implemented. The first follows closely in the footsteps of Ranta (1994), albeit possibly with certain revisions, enrichments, and modifications (see, e.g., Luo 2012, Grudzińska & Zawadowski 2014, Chatzikyriakidis & Luo 2020). These theories retain the intuitionistic/constructivist foundations of Martin-Löf type theory (although this does not mean that they cannot be interpreted model-theoretically). The second tends more toward model theory, a principal example being TTR (e.g., Cooper 2011, 2023).

[7] In Montague Grammar, such expressions are interpreted as sets, individuals, etc. In representationalist frameworks such as Discourse Representation Theory (DRT; e.g., Kamp & Reyle 1993, Kamp et al. 2011), it is claimed that the representational object language layer is not disposable insofar as one cannot map directly from language to the set-interpretation level.

Type-theoretic semantics of the first sort tend to expand greatly the set of basic types, typically to include a basic type for all common nouns (see Mönnich 1985, Sundholm 1986), for instance, *Cat*, *Dog*. Types corresponding to VPs, on the other hand, are constructed as dependent types, for example, $Run(x)$, where x may be filled by an entity of some common-noun type (such as *Cat*). For instance, $\Sigma x : Cat.Run(x)$ is a type, any witnesses of which are pairs of individuals and, say, events or situations [e.g., (a, s) if $a : Cat$ and $s : Run(a)$].

In TTR, by contrast, Cooper (2012, 2023, among others) stays somewhat closer to the model-theoretic tradition in semantics, albeit in terms of a rich theory of types. In model-theoretic semantics, based on a simple theory of types (e.g., e, t), the domain of individuals is given in the model, and one can interpret predicates of individuals (e.g., constants of type $\langle e, t\rangle$) relative to the model. In a related vein, Cooper assumes a constrained set of basic types such as *Ind* (instead of e). Predicates in TTR, rather than being constants of some type, are instead dependent type constructors. For instance, where v is a variable over values, and x is a label, $\langle \lambda v : Ind.\mathrm{cat}(v)\langle r.\mathrm{x}\rangle\rangle$ is a type constructor with an arity $\langle Ind\rangle$, which, for a record r of type $[\mathrm{x} : Ind]$, returns the type $\mathrm{cat}(a)$ if a is the value that is labeled by x in r. The formula in example 12a provides a simple representation of the common noun *cat*. It is a function from records (situations) that contain an individual to a record type (proposition) that the individual is a cat. If the individual in question is Felix, then the proposition will be true if there is a record such as the one in example 12b, such that *felix* is of type *Ind* and the situation, *s*, is of type cat(*felix*):[8,9]

(12a) $\lambda r : [\mathrm{x} : Ind].\left[\mathrm{c_{cat}} : \mathrm{cat}(r.\mathrm{x})\right]$

(12b) $\begin{bmatrix} \mathrm{x} & = & felix \\ \mathrm{c_{cat}} & = & s \end{bmatrix}$

5.5. Subtyping

Subsumptive subtyping: if an entity a is of type σ that is a subtype of another type τ, then a is also of type τ

Coercive subtyping: if an entity a is of one type σ that is a subtype of another type τ, then there is an entity coercible from a that is of type τ

With a rich theory of types, an important element of reasoning about types concerns subtypes, the type theory analog of the subset relation in set theory. Intuitively, when one type is included in another, the former is a subtype of the latter. Subtyping is an account of reasoning about what type an entity is, given some judgment and a subtype relationship. One way of characterizing subtypes also mirrors that of the subset relation in set theory, subsumptive subtyping. Just as the member of any subset of a set is a member of that set, if an entity is of one type, σ, that is a subtype of another type, τ, then that entity is also of type τ:

(13) $a : \sigma, \sigma \leq \tau \vdash a : \tau$

However, as Luo (1997) and Luo et al. (2013) show, subsumptive subtyping is incompatible with a principle of RTTs that are more inferentialist, namely, canonicity (the principle that all meaningful statements can be reduced to canonical form where all types in this form have been constructed by their construction/formation rules). Luo (1997) proposes replacing subsumptive subtyping with coercive subtyping: "$A \leq B$ if there is a unique implicit coercion c from type A to type B, also written as $A \leq_c B$" (Chatzikyriakidis & Luo 2020, pp. 39–40; see also Luo 1999, Asher 2011, Cooper 2023). The intuitive difference here is that, if $\sigma \leq \tau$ and $a : \sigma$, although a itself is not of

[8]Note that the witnesses for Σ-types (see above) are ordered pairs of entities, but in TTR, witnesses for propositions are records that are unordered sets of labels (e.g., x, $\mathrm{c_{cat}}$) and values (e.g., *felix* and *s*).

[9]The predicate in example 12a is syntactically sugared insofar as it hides some details of how a type is constructed from the value of a label in a record, which in full would be $\langle \lambda v : Ind.\mathrm{cat}(v)\langle r.\mathrm{x}\rangle\rangle$ (see, e.g., Cooper 2012). Also, the record in example 12b is a matrix-based notational variant for a set of ordered label–value pairs, in this case, $\{\langle \mathrm{x}, felix\rangle, \langle \mathrm{c_{cat}}, s\rangle\}$.

type τ, it nonetheless has a counterpart, coercible from a ($c(a)$), that is of type τ. This solves the aforementioned technical problem with subsumptive subtyping. From a more philosophical point of view, arguably one way to think about this is that such coercions are akin to different modes of presentation of, or perspectives on, the same object.

Hyperintensionality: a problem arising when semantic representations are too coarsely grained to differentiate the meanings of, e.g., mathematical propositions

In more model-theoretic versions of RTTs such as TTR, record types are used to maintain a version of subsumptive subtyping. For instance, the relations in item 14 exemplify one way that a record type can be a subtype of another. (Note that $\sqsubseteq$ is used instead of $\leq$ in TTR.) If R_1 is a subtype of R_2, then R_1 has all of the fields that R_2 has (plus possibly some additional fields):[10]

$$(14)\qquad \begin{bmatrix} x & : & \mathit{Ind} \\ c_1 & : & \mathrm{cat}(x) \\ c_2 & : & \mathrm{cute}(x) \end{bmatrix} \sqsubseteq \begin{bmatrix} x & : & \mathit{Ind} \\ c_1 & : & \mathrm{cat}(x) \end{bmatrix}; \qquad \begin{bmatrix} x & : & \mathit{Ind} \\ c_1 & : & \mathrm{cat}(x) \end{bmatrix} \sqsubseteq \begin{bmatrix} x : \mathit{Ind} \end{bmatrix}$$

6. APPLICATIONS OF RICH TYPE THEORIES

6.1. Hyperintensionality and Propositional Attitude Constructions

A guiding intuition of simply typed semantics is that there is a correspondence between syntactic categories and semantic types. For instance, the interpretations of common nouns are commonly assumed to be of type $\langle s, \langle e, t\rangle\rangle$, and the interpretations of sentences (propositions) are of a type, a common choice being $\langle s, t\rangle$ (the type of sets of possible worlds/world–time pairs). So we have, for instance, propositions modeled as sets of possible worlds, each of which is of type $\langle s, t\rangle$. That is, each set corresponds to a mapping between the domain of type s (the set of possible worlds) and the domain of type t (the set of truth values 0 and 1). A well-known problem that arises is that of hyperintensionality. For instance, to take an example from Lewis (1970), although intuitively, "Grass is green or isn't" differs in meaning from "Snow is white or isn't," these two sentences have the same intension (are true in the same set of possible worlds). Likewise, all true mathematical statements share this feature.

The richly typed view of semantics retains the correspondence between syntactic categories and semantic types from simply typed semantics but also provides a means of addressing hyperintensionality. For instance, the type *Prop* (e.g., in MTT) is the type for all propositions. (If ϕ is a proposition, then ϕ : *Prop*, and likewise for the type *RecType*, the type of all record types in TTR. Insofar as propositions can be modeled as record types, *RecType* is the type of any such proposition.) However, given the propositions-as-types hypothesis, propositions are also types. Propositional types may have witnesses, for instance, records/situations in TTR, and two such types may even have the same witnesses, but that does not make them the same types. Indeed, types can be distinguished in how they are constructed and, relatedly, in terms of the structure they have.

The original proposal for analyzing attitudes (Ranta 1994) makes use of agent indexed contexts, that is, if Γ_A is the belief context for A, then A believes ϕ just in case $\Gamma_A \vdash \phi\ \mathit{True}$ (just in case ϕ can be inferred from Γ_A). Chatzikyriakidis & Luo (2020) refine this idea by allowing contexts to be intensional, thus averting problems of overgenerating beliefs, for instance, that *A believes that Oedipus intentionally married Jocasta* should not entail that *A believes that Oedipus intentionally married his mother*.

Also developing Ranta's (1994) original proposal, as well as his own earlier proposal (Cooper 2005), Cooper (2023) models an agent's belief context as a record type. Since propositions are also record types in TTR, belief is captured via subtyping (modulo some relabeling of fields in

[10]For further discussion and references, readers are referred to Chatzikyriakidis & Cooper (2018, section 2.4).

the record types), namely, that an agent, A, believes ϕ if and only if the record type modeling the agent's belief context is a subtype of the record type ϕ.

6.2. Quantification and Anaphora

Already in works by Sundholm (1986) and Ranta (1994), it has been shown that RTTs can capture donkey anaphora using dependent types (namely, Σ-types; see above) and Π-types. The witnesses of a Π-type are functions. Ranta (1994) uses Π-types for universal quantification [Chatzikyriakidis & Luo (2020) have Π-types, the witnesses of which are functions, but they use a distinct dependent type for universal quantification]. Below, some further developments are summarized.

Donkey anaphora: constructions in which anaphoric pronouns are dependent on antecedent material, as with *it* in *Every farmer who owns a donkey beats it*

Copredication: when multiple, prima facie incompatible predications are made based on a single polysemous antecedent expression

Bekki (2014) develops and extends Ranta's framework with a compositional discourse theory to provide a fully compositional account for anaphora and presupposition. Grudzińska & Zawadowski (2014, 2019, 2020) use a richly typed semantics and dependent types to model anaphora and a number of quantificational and anaphoric phenomena, including inverse linking constructions, possessive weak definites, and quantifier alternation. Piwek & Krahmer (2000) use RTT to analyze presupposition, its interaction with anaphoric pronouns, and bridging phenomena.

Cooper (2023) develops a rich notion of context [broadly speaking, mirroring Kaplan's (1978, 1989) proposal that expressions have characters and contents] with which he treats many phenomena, including quantification domain restriction and anaphora (but also puzzles relating to proper names and intensionality). Luo (2021) distinguishes between weak and strong forms of sum types to address puzzles arising from donkey anaphora in relation to counting.

6.3. Polysemy, Coercion, and Lexical Restrictions

Polysemous nouns have numerous interrelated senses. For instance, *lunch* can denote an eating eventuality or the food consumed. Furthermore, one can copredicate over many of these senses based on a single antecedent, as in, for instance, the sentence in example 15:

(15) Lunch was delicious but took forever.
(Asher & Pustejovsky 2013, p. 44)

When combined with quantifiers or numerals (e.g., *three informative books are heavy*), at least in some cases, we get a double distinctness interpretation (see, e.g., Gotham 2014), as discussed with regard to example 7 in Section 4.

Two principal puzzles arise for STT semantics in the light of polysemy and copredication: first, how to model terms that can denote entities of many distinct types, and second, how double distinctness interpretations can be captured.

Relatedly, we have cases of coercion (which arguably, unlike polysemy, are not lexicalized), in which expressions can be shifted, in context, to other interpretations, as we see in example 16, in which *novel* is coerced to mean *began reading/writing the novel*:

(16) Mary began the novel.
(Pustejovsky 1995, p. 32)

Asher (2011) develops a theory of both polysemy and copredication within TCL based on dot types. For instance, for a physical and informational book entity of type $p \bullet i$, modifiers can access aspects of this entity via the use of functions akin to the projection functions defined for product types in Section 4.1.

Retoré (2014) and Bassac et al. (2010) develop the Montagovian Generative Lexicon based on a multisorted, second-order λ-calculus, ΛTy_n. This proposal addresses phenomena such as coercion by using the expressiveness of the language to integrate lexical semantic information into a formal semantic framework. A second-order λ-calculus is also used, along with λ-DRT, to account for the counting and copredication puzzle in a paper by Mery et al. (2019).

Different accounts of polysemy have been developed within TTR. Cooper (2007, 2011) uses (dependent) predicate types to model entities having different aspects in the sense of, for instance, Asher's (2011) work. Sutton (2022) proposes that common nouns such as *lunch* denote situations that witness entities of different types and uses neo-Davidsonian-inspired thematic roles to constrain when predication across different senses is licensed [see also Ortega-Andrés & Vicente (2019) for related work based on Pustejovsky's (1995) Generative Lexicon].

Chatzikyriakidis & Luo (2015, 2018, 2020) develop an account of lexical semantics within MTT to deal with the above puzzles in which polysemous expressions characterize individuation criteria (see also Gotham 2014).

6.4. Probabilistic Semantics

Probabilistic versions of RTTs—specifically, probabilistic TTR (pTTR)—have been developed (Cooper et al. 2014, 2015). TTR is amenable to a bottom-up probabilistic interpretation, given that type judgments can be stored, updated, and then used to calculate probabilistic dependencies between types. In other words, one can model dynamic semantic learning in which newly encountered entities may be judged/classified as being of some type. This contrasts with top-down probabilistic models based on possible worlds (Lassiter 2011, van Eijck & Lappin 2012), in which probabilities are calculated over sets of possible worlds (and also sets of possible languages).

pTTR has been used to model vagueness (Larsson & Fernández 2014), to investigate the interaction between perceptual individuation and the count/mass distinction (Sutton & Filip 2017), and for classification based on visual information (Dobnik et al. 2013).

Probabilistic versions of TTR have also been integrated with Dynamic Syntax (DS) in the analysis of dialogue (see, e.g., Hough & Purver 2014, Hough et al. 2015). RTT approaches to dialogue are discussed below.

6.5. Dialogue

RTTs have also been employed to address issues arising from the analysis of dialogue, sometimes in combination with theories of the syntax–semantics interface. For instance, Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag 1994) has been integrated with TTR (see, e.g., Ginzburg 2012). Also, DS (Kempson et al. 2001) has been integrated with TTR, producing DS-TTR (see, e.g., Purver et al. 2010; Gregoromichelaki 2018; Gregoromichelaki et al. 2020, 2022; Eshghi et al. 2023) and probabilistic varieties thereof (Hough & Purver 2014, Hough et al. 2015). Such theories have provided insights regarding many phenomena including split utterances such as example 17, which pose a challenge for theories of syntax–semantics that assume complete sentences as a fundamental notion of communication and coordination, and underspecification as in example 18, where B's utterance has at least two readings as in examples 18a and b (see also Gregoromichelaki 2018 for an application to quotation):

(17a) A: I'm pretty sure that the
(17b) B: programmed visits?
(17c) A: programmed visits, yes, I think they'll have been debt inspections.
(Eshghi et al. 2023, from BNC KS1 789–791)

(18) Who does Bo admire? B: Bo?
(18a) Clausal confirmation: Are you asking who BO (of all people) admires?
(18b) Intended content: Who do you mean 'Bo'?
(Ginzburg 2012, p. 25)

One advantage offered by a rich theory of types with respect to these phenomena is that it offers one structure (types themselves can be structured objects) with which to characterize interactions between ever-evolving contexts and multimodal information sources, which can be integrated on a word-by-word basis with the addition of an incremental grammatical framework such as DS.

6.6. Selectional Restrictions and Modification

Type restrictions on modification are used in STTs. For instance, assuming disjoint basic types e and v, one can restrict adjectival modifiers to apply to eventuality-denoting predicates instead of predicates of physical entities. RTTs allow for more fine-grained distinctions to be made with respect to restrictions on modification without the need for meaning postulates. For instance, the type theory can be rich enough to restrict an adjective such as *friendly* to apply only to humans (or at least to animate entities). Thus, richer type theories shift more weight of explanation for selectional restrictions to the type theory (see, e.g., Chatzikyriakidis & Luo 2020, chapter 3).

One proposal for analyzing intersective modification is in terms of Σ-types (Mönnich 1985, Sundholm 1986, Ranta 1994). For instance, if *black* is of type *Object* → *Prop*, it can modify *cat*, since the type *Cat* is a subtype of the type *Object*. However, given the aforementioned complications relating to subsumptive subtyping, this picture was developed and completed by Chatzikyriakidis & Luo (2020) based on work by, for instance, Luo (1999, 2012) on coercive subtyping.

Chatzikyriakidis & Luo (2013, 2017, 2020) extend this account to subsective adjectives (e.g., *large*) by also making use of Π-types, such that subsective adjectives are type polymorphic (roughly, for any common-noun type σ, a subsective adjective may be of type $\sigma \rightarrow \tau$). (For further discussion of privative adjectives such as *fake* and noncommittal adjectives such as *alleged*, see Chatzikyriakidis & Luo 2020, chapter 3.)

The Partee Puzzle is a puzzle in semantics introduced by Barbara Partee regarding, for instance, why *The temperature is rising* and *The temperature is ninety* does not entail *Ninety is rising*. Cooper (2012, 2023) addresses this puzzle for constructions such as *the temperature is rising* in terms of a modification of record types but also in terms of distinction between common-noun predicates (recall that for Cooper, predicates are type constructors) with an arity of ⟨*Ind*⟩ and those with an arity of ⟨*Rec*⟩ (Record). For instance, the predicate *cat* constructs types from values of type *Ind*, and the predicate *temperature* constructs types from values of type *Rec*. That is, intuitively, *temperature* describes situations, while *cat* describes entities (albeit ones that are contained in a situation). For a discussion of Partee Puzzle cases in the context of a propositionalist approach to semantics, readers are referred to the article in this volume by Liefke (2024).

7. SUMMARY AND FUTURE DIRECTIONS

There are many ways in which Church–Montague semantics based on a simple system of types can be modified, extended, and adapted to the needs of those working in formal semantics. Within much work done with STTs, the default move for explaining some new phenomenon that cannot be captured within a type theory with basic types e, t, and s and one functional type constructor is to assume an additional basic type. Although there is nothing inherently wrong with this strategy, hopefully this review makes clear that there are many well-studied alternative strategies that could be explored (see also the sidebar titled Further Resources). As a simple example, given some propositional content of type $\langle s, t\rangle$, there may be reasons to assume an additional type i for informational object and then to provide a meaning postulate that connects any ϕ of type $\langle s, t\rangle$ to an individual in the domain of i. However, an equally valid but far less frequently considered alternative would be to construct a new type that depends on ϕ. As detailed above, adding new type constructors does not mean moving from an STT to an RTT.

FURTHER RESOURCES

1. For introductions to RTTs, the introductory chapters of Ranta (1994) and Chatzikyriakidis & Luo (2020), as well as the encyclopedia entry from Chatzikyriakidis & Cooper (2018), are recommended.
2. Cooper (2023) provides a thorough introduction to TTR, and MTTs are likewise introduced in the encyclopedia entry from Chatzikyriakidis & Cooper (2018).
3. An overview of the use of structure in semantics, including that based on RTTs, is forthcoming from Chatzikyriakidis et al. (2024).

Perhaps one of the more prominent motivations for using RTTs is a dissatisfaction with possible-worlds semantics exemplified by problems relating to hyperintensionality. As discussed in Sections 3 and 5, there are alternative routes available here, too. For instance, one can simplify one's type theory in the manner of property theory, or enrich it to an RTT. In choosing between these options, a difference between them relates to structure. Adding type constructors such as product types can give one a limited amount of structure with which one can, for instance, add lexical information (Section 4.1). RTTs are inherently structured insofar as one's semantic objects are structured types, which can differ both with respect to their structure and with respect to what is of that type. Structured types have been widely used to integrate lexical and compositional semantic information.

Much more work is to be done with RTTs, however—not least given the much larger STT research program in semantics. In addition, while probabilistic versions of more model-theoretic RTTs (namely, TTR) have been developed, it remains an open question how to develop, for instance, a probabilistic version of MTT. The tools and strategies developed in RTTs have nonetheless already proved valuable for addressing many semantic phenomena (Section 6) and, plausibly, could even be sources for innovation within the STT tradition.

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The author is very grateful to Stergios Chatzikyriakidis, Robin Cooper, and Eleni Gregoromichelaki for helpful discussions they had in preparation of a distinct work, without which this article would have been much impoverished. Thanks also to Stergios Chatzikyriakidis, Robin Cooper, and Louise McNally for their highly helpful comments during the preparation of this review.

P.S. received funding from the Beatriu de Pinós postdoctoral fellowships program, funded by the Secretary of Universities and Research (Government of Catalonia) and from the Horizon 2020 program of research and innovation of the European Union under the Marie Skłodowska-Curie grant agreement 801370.

LITERATURE CITED

Asher N. 2011. *Lexical Meaning in Context: A Web of Words*. Cambridge, UK: Cambridge Univ. Press

Asher N, Pustejovsky J. 2013. A type composition logic for generative lexicon. In *Advances in Generative Lexicon Theory*, ed. J Pustejovsky, P Bouillon, H Isahara, K Kanzaki, C Lee, pp. 39–66. Dordrecht, Neth.: Springer

Barsalou LW. 1992. Frames, concepts, and conceptual fields. In *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*, ed. E Kittay, A Lehrer, pp. 21–74. Hillsdale, NJ: Erlbaum

Bassac C, Mery B, Retoré C. 2010. Towards a type-theoretical account of lexical semantics. *J. Logic Lang. Inform.* 19(2):229–45

Bekki D. 2014. Representing anaphora with dependent types. In *Logical Aspects of Computational Linguistics*, ed. N Asher, S Soloviev, pp. 14–29. Berlin/Heidelberg, Ger.: Springer

Carpenter B. 1997. *Type-Logical Semantics*. Cambridge, MA: MIT Press

Chatzikyriakidis S, Cooper R. 2018. Type theory for natural language semantics. In *Oxford Research Encyclopedia of Linguistics*. Oxford, UK: Oxford Univ. Press. **https://doi.org/10.1093/acrefore/9780199384655.013.329**

Chatzikyriakidis S, Cooper R, Gregoromichelaki E, Sutton PR. 2024. *Theories of Types and the Structure of Meaning*. Cambridge Elements Series. Cambridge, UK: Cambridge Univ. Press. In press

Chatzikyriakidis S, Luo Z. 2013. Adjectives in a modern type-theoretical setting. In *Proceedings of Formal Grammar, 17th and 18th International Conferences (FG 2012/2013)*, ed. G Morrill, M Nederhof, pp. 159–74. Dordrecht, Neth.: Springer

Chatzikyriakidis S, Luo Z. 2015. Individuation criteria, dot-types and copredication: a view from modern type theories. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, ed. M Kuhlmann, M Kanazawa, GM Kobele, pp. 39–50. Stroudsburg, PA: Assoc. Comput. Linguist.

Chatzikyriakidis S, Luo Z. 2017. Adjectival and adverbial modification: the view from modern type theories. *J. Logic Lang. Inform.* 26:45–88

Chatzikyriakidis S, Luo Z. 2018. Identity criteria of common nouns and dot-types for copredication. *Oslo Stud. Lang.* 10(2):121–41

Chatzikyriakidis S, Luo Z. 2020. *Formal Semantics in Modern Type Theories*. London, UK/Hoboken, NJ: ISTE/Wiley

Chierchia G, Turner R. 1988. Semantics and property theory. *Linguist. Philos.* 11(3):261–302

Church A. 1940. A formulation of the simple theory of types. *J. Symb. Logic* 5(2):56–68

Cooper R. 2005. Austinian truth, attitudes and type theory. *Res. Lang. Comput.* 3:333–62

Cooper R. 2007. Copredication, dynamic generalized quantification and lexical innovation by coercion. In *Proceedings of GL 2007, Fourth International Workshop on Generative Approaches to the Lexicon*, ed. P Bouillon, L Danlos, K Kanzaki, pp. 143–84. Geneva, Switz.: Univ. Geneva

Cooper R. 2011. Copredication, quantification and frames. In *Proceedings of the 6th International Conference on Logical Aspects of Computational Linguistics (LACL 2011)*, ed. S Pogodalla, JP Prost, pp. 64–79. Berlin, Ger.: Springer

Cooper R. 2012. Type theory and semantics in flux. In *Philosophy of Linguistics*, ed. R Kempson, T Fernando, N Asher, pp. 271–323. Amsterdam, Neth.: Elsevier

Cooper R. 2023. *From Perception to Communication: A Theory of Types for Action and Meaning*. Oxford, UK: Oxford Univ. Press

Cooper R, Dobnik S, Lappin S, Larsson S. 2014. A probabilistic rich type theory for semantic interpretation. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pp. 72–79. Stroudsburg, PA: Assoc. Comput. Linguist.

Cooper R, Dobnik S, Larsson S, Lappin S. 2015. Probabilistic type theory and natural language semantics. *LiLT* 10. **https://doi.org/10.33011/lilt.v10i.1357**

Cresswell M. 1977. The semantics of degree. In *Montague Grammar*, ed. B Partee, pp. 261–92. New York, NY: Academic

Curry HB. 1942. The inconsistency of certain formal logics. *J. Symb. Logic* 7(3):115–17

Curry HB, Feys R. 1958. *Combinatory Logic*, Vol. 1. Amsterdam, Neth.: North-Holland

Dobnik S, Cooper R, Larsson S. 2013. Modelling language, action and perception in type theory with records. In *Proceedings of the 7th International Workshop on Constraint Solving and Language Processing (CSLP 2012)*, ed. D Duchier, Y Parmentier, pp. 70–91. Berlin, Ger.: Springer

Dowty DR, Wall RE, Peters S. 1981. *Introduction to Montague Semantics*. Dordrecht, Neth.: Kluwer

Eshghi A, Gregoromichelaki E, Howes C. 2023. Action coordination and learning in dialogue. In *Probabilistic Approaches to Linguistic Theory*, ed. J-P Bernardy, R Blanck, S Chatzikyriakidis, S Lappin, A Maskharashvili, pp. 357–418. Stanford, CA: CSLI

Filip H, Sutton PR. 2017. Singular count NPs in measure constructions. In *Proceedings of the 27th Conference on Semantics and Linguistic Theory (SALT 27)*, ed. D Burgdorf, J Collard, S Maspong, B Stefánsdóttir, pp. 340–57. Washington, DC: Linguist. Soc. Am.

Gallin D. 1975. *Intensional and Higher-Order Modal Logic*. Amsterdam, Neth.: North-Holland

Ginzburg J. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford, UK: Oxford Univ. Press

Gotham M. 2014. *Copredication, quantification and individuation*. PhD Thesis, Univ. College London, London, UK

Gotham M. 2017. Composing criteria of individuation in copredication. *J. Semant.* 34(2):333–71

Gotham M. 2021. Property inheritance, deferred reference and copredication. *J. Semant.* 39(1):87–116

Gregoromichelaki E. 2018. Quotation in dialogue. In *The Semantics and Pragmatics of Quotation*, ed. P Saka, M Johnson, pp. 195–255. Cham, Switz.: Springer

Gregoromichelaki E, Chatzikyriakidis S, Eshghi A, Hough J, Howes C, et al. 2020. Affordance competition in dialogue: the case of syntactic universals. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2020)*, ed. S Malamud, J Pustejovsky, J Ginzburg, pp. 204–19. Waltham, MA: Brandeis Univ.

Gregoromichelaki E, Eshghi A, Howes C, Mills GJ, Kempson R, et al. 2022. Language and cognition as distributed process interactions. In *Proceedings of the 26th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2022)*, ed. E Gregoromichelaki, J Hough, JD Kelleher, pp. 160–71. Dublin, Irel.: Technol. Univ. Dublin

Grudzińska J, Zawadowski M. 2014. System with generalized quantifiers on dependent types for anaphora. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pp. 10–18. Stroudsburg, PA: Assoc. Comput. Linguist.

Grudzińska J, Zawadowski M. 2019. Inverse linking, possessive weak definites and haddock descriptions: a unified dependent type account. *J. Logic Lang. Inform.* 28(2):239–60

Grudzińska J, Zawadowski M. 2020. A scope-taking system with dependent types and continuations. In *Proceedings of the Symposium on Logic and Algorithms in Computational Linguistics 2018 (LACompLing2018)*, ed. R Loukanova, pp. 155–76. Cham, Switz.: Springer

Hough J, Kennington C, Schlangen D, Ginzburg J. 2015. Incremental semantics for dialogue processing: requirements, and a comparison of two approaches. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, pp. 206–16. Stroudsburg, PA: Assoc. Comput. Linguist.

Hough J, Purver M. 2014. Probabilistic type theory for incremental dialogue processing. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pp. 80–88. Stroudsburg, PA: Assoc. Comput. Linguist.

Howard W. 1980. The formulae-as-types notion of construction. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, ed. R Hindley, JP Seldin, pp. 479–90. London, UK: Academic

Kamp H, Reyle U. 1993. *From Discourse to Logic*. Dordrecht, Neth.: Kluwer

Kamp H, van Genabith J, Reyle U. 2011. Discourse Representation Theory. In *Handbook of Philosophical Logic*, Vol. 15, ed. D Gabbay, F Guenthner, pp. 125–394. Dordrecht, Neth.: Springer

Kaplan D. 1978. On the logic of demonstratives. *J. Philos. Logic* 8:81–98

Kaplan D. 1989. Demonstratives: an essay on the semantics, logic, metaphysics and epistemology of demonstratives and other indexicals. In *Themes from Kaplan*, ed. J Almog, J Perry, H Wettstein, pp. 481–563. Oxford, UK: Oxford Univ. Press

Kempson R, Meyer-Viol W, Gabbay D. 2001. *Dynamic Syntax*. Oxford, UK: Blackwell

Kennedy C, McNally L. 2005. Scale structure and the semantic typology of gradable predicates. *Language* 81:345–81

Klein E. 1980. A semantics for positive and comparative adjectives. *Linguist. Philos.* 4:1–45

Kohlhase M. 1992. Unification in order-sorted type theory. In *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR'92)*, ed. A Voronkov, pp. 421–32. Berlin, Ger.: Springer

Kohlhase M. 1994. *A mechanization of sorted higher-order logic based on the resolution principle*. PhD Thesis, Saarland Univ., Saarbrücken, Ger.

Krifka M. 1989. Nominal reference, temporal constitution and quantification in event semantics. In *Semantics and Contextual Expression*, ed. R Bartsch, J van Benthem, P van Emde Boas, pp. 75–115. Dordrecht, Neth.: Foris

Krifka M. 1995. Common nouns: a contrastive analysis of English and Chinese. In *The Generic Book*, ed. G Carlson, FJ Pelletier, pp. 398–411. Chicago, IL: Univ. Chicago Press

Landman F. 2000. *Events and Plurality: The Jerusalem Lectures*. Dordrecht, Neth.: Kluwer

Landman F. 2011. Count nouns – mass nouns, neat nouns – mess nouns. In *The Baltic International Yearbook of Cognition, Logic and Communication*, Vol. 6: *Formal Semantics and Pragmatics: Discourse, Context and Models*, ed. BH Partee, M Glanzberg, J Skilters. Manhattan, KS: New Prairie. **https://doi.org/10.4148/biyclc.v6i0.1579**

Landman F. 2016. Iceberg semantics for count nouns and mass nouns: classifiers, measures and portions. In *The Baltic International Yearbook of Cognition, Logic and Communication*, Vol. 11: *Number: Cognitive, Semantic and Crosslinguistic Approaches*, ed. S Rothstein, J Šķilters. Manhattan, KS: New Prairie. **https://doi.org/10.4148/1944-3676.1107**

Larsson S, Fernández R. 2014. Vagueness and learning: a type-theoretic approach. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, ed. J Bos, A Frank, R Navigli, pp. 151–59. Stroudsburg, PA: Assoc. Comput. Linguist.

Lassiter D. 2011. Vagueness as probabilistic linguistic knowledge. In *Vagueness in Communication: Revised Selected Papers from the 2009 International Workshop on Vagueness in Communication (VIC 2009)*, ed. R Nouwen, R van Rooij, U Sauerland, HC Schmitz, pp. 127–50. Berlin, Ger.: Springer

Lewis D. 1970. General semantics. *Synthese* 22:18–67

Liefke K. 2014. *A single-type semantics for natural language*. PhD Thesis, Tilburg Univ., Tilburg, Neth.

Liefke K. 2024. Intensionality and propositionalism. *Annu. Rev. Linguist.* 10:85–105

Liefke K, Werning M. 2018. Evidence for single-type semantics? An alternative to *e*/*t*-based dual-type semantics. *J. Semant.* 35(4):639–85

Löb M. 1945. Solution of a problem of Leon Henkin. *J. Symb. Logic* 20(2):115–18

Löbner S. 2015. Functional concepts and frames. In *Meaning, Frames, and Conceptual Representation*, ed. T Gamerschlag, D Gerland, R Osswald, W Petersen, pp. 15–42. Düsseldorf, Ger.: Düsseldorf Univ. Press

Luo Z. 1997. Coercive subtyping in type theory. In *Proceedings of the 10th International Workshop on Computer Science Logic (CSL '96)*, ed. D van Dalen, M Bezem, pp. 275–96. Berlin/Heidelberg, Ger.: Springer

Luo Z. 1999. Coercive subtyping. *J. Logic Comput.* 9(1):105–30

Luo Z. 2012. Formal semantics in modern type theories with coercive subtyping. *Linguist. Philos.* 35(6):491–513

Luo Z. 2021. Donkey anaphora: type-theoretic semantics with both strong and weak sums. In *Proceedings of the ESSLLI 2021 Workshop on Computing Semantics with Types, Frames and Related Structures*, pp. 45–52. Stroudsburg, PA: Assoc. Comput. Linguist.

Luo Z, Soloviev S, Xue T. 2013. Coercive subtyping: theory and implementation. *Inform. Comput.* 223:18–42

Martin-Löf P. 1984. *Intuitionistic Type Theory*. Naples, Italy: Bibliopolis

Mery B, Moot R, Retoré C. 2019. Solving the individuation and counting puzzle with λ-DRT and MGL. In *New Frontiers in Artificial Intelligence*, ed. K Kojima, M Sakamoto, K Mineshima, K Satoh, pp. 298–312. Cham, Switz.: Springer

Mönnich U. 1985. *Untersuchungen zu einer konstruktiven Semantik fur ein Fragment des Englischen*. PhD Thesis, Univ. Tübingen, Tübingen, Ger.

Montague R. 1970. Universal grammar. *Theoria* 36:373–98

Montague R. 1973. The proper treatment of quantification in ordinary English. In *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, ed. J Hintikka, J Moravcsik, P Suppes, pp. 247–70. Dordrecht, Neth.: D. Reidel

Ortega-Andrés M, Vicente A. 2019. Polysemy and co-predication. *Glossa* 4(1):1

Parsons T. 1990. *Events in the Semantics of English*. Cambridge, MA: MIT Press

Partee B. 2007. *Type theory and natural language: Do we need two basic types?* Paper presented at the 100th Meeting of the Seminar: Mathematical Methods Applied to Linguistics, Moscow, Russ., Mar. 31

Petersen W. 2015. Representation of concepts as frames. In *Meaning, Frames, and Conceptual Representation*, ed. T Gamerschlag, D Gerland, R Osswald, W Petersen, pp. 43–67. Düsseldorf, Ger.: Düsseldorf Univ. Press

Piwek P, Krahmer E. 2000. Presuppositions in context: constructing bridges. In *Formal Aspects of Context*, ed. P Bonzon, M Cavalcanti, R Nossu, pp. 85–106. Dordrecht, Neth.: Springer

Pollard C, Sag IA. 1994. *Head-Driven Phrase Structure Grammar*. Stanford, CA: CSLI
Purver M, Gregoromichelaki E, Meyer-Viol W, Cann R. 2010. Splitting the 'I's and crossing the 'you's: context, speech acts and grammar. In *Proceedings of the 14th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2010)*, ed. P Łupkowski, M Purver, pp. 43–50. Poznań, Pol.: Pol. Soc. Cogn. Sci.
Pustejovsky J. 1994. Semantic typing and degrees of polymorphism. In *Current Issues in Mathematical Linguistics*, ed. C Martin-Vide, pp. 221–38. Amsterdam, Neth.: North-Holland
Pustejovsky J. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press
Pustejovsky J. 2013. Type theory and lexical decomposition. In *Advances in Generative Lexicon Theory*, ed. J Pustejovsky, P Bouillon, H Isahara, K Kanzaki, C Lee, pp. 9–38. Dordrecht, Neth.: Springer
Ranta A. 1994. *Type-Theoretical Grammar*. Oxford, UK: Clarendon
Retoré C. 2014. The Montagovian generative lexicon $\wedge Ty_n$: a type theoretical framework for natural language semantics. In *Proceedings of the 19th International Conference on Types for Proofs and Programs (TYPES 2013)*, ed. R Matthes, A Schubert, pp. 202–29. Wadern, Ger.: LIPIcs
Rothstein S. 2010. Counting and the mass/count distinction. *J. Semant.* 27(3):343–97
Rothstein S. 2017. *Semantics for Counting and Measuring*. Cambridge, UK: Cambridge Univ. Press
Seuren P. 1973. The comparative. In *Generative Grammar in Europe*, ed. F Keifer, N Ruwet, pp. 528–64. Dordrecht, Neth.: D Reidel
Sundholm G. 1986. Proof theory and meaning. In *Handbook of Philosophical Logic*, Vol. 3: *Alternatives to Classical Logic*, ed. D Gabbay, F Guenthner, pp. 471–506. Dordrecht, Neth.: Springer
Sutton PR. 2022. Restrictions on copredication: a situation theoretic approach. In *Proceedings of the 32nd Conference on Semantics and Linguistic Theory (SALT 32)*, ed. JR Starr, J Kim, B Öney, pp. 335–55. Washington, DC: Linguist. Soc. Am.
Sutton PR, Filip H. 2016a. Counting in context: count/mass variation and restrictions on coercion in collective artifact nouns. In *Proceedings of the 26th Conference on Semantics and Linguistic Theory (SALT 26)*, ed. M Moroney, C-R Little, J Collard, D Burgdorf, pp. 350–70. Washington, DC: Linguist. Soc. Am.
Sutton PR, Filip H. 2016b. Vagueness, overlap, and countability. In *Proceedings of Sinn und Bedeutung 20*, ed. N Bade, P Berezovskaya, A Schöller, pp. 730–47. Tübingen, Ger.: Univ. Tübingen
Sutton PR, Filip H. 2017. Individuation, reliability, and the mass/count distinction. *J. Lang. Model.* 5(2):303–56
Sutton PR, Filip H. 2018. Restrictions on subkind coercion in object mass nouns. In *Proceedings of Sinn und Bedeutung 21*, ed. R Truswell, C Cummins, C Heycock, B Rabern, H Rohde, pp. 1195–213. Edinburgh, UK: Univ. Edinburgh
Sutton PR, Filip H. 2020. Informational object nouns and the mass/count distinction. In *Proceedings of Sinn und Bedeutung 24*, Vol. 2, ed. M Franke, N Kompa, M Liu, JL Mueller, J Schwab, pp. 319–35. Osnabrüück/Berlin, Ger.: Osnabrüück Univ./Humboldt Univ. Berlin
van Eijck J, Lappin S. 2012. Probabilistic semantics for natural language. In *Logic and Interactive RAtionality (LIRA) Yearbook 2012*, Vol. 2, ed. Z Christoff, P Galeazzi, N Gierasimszuk, A Marcoci, S Smets, pp. 17–35. Amsterdam, Neth.: Univ. Amsterdam
Windhearn M. 2021. *Alternatives, exclusivity and underspecification*. PhD Thesis, Cornell Univ., Ithaca, NY
Zobel S. 2017. The sensitivity of natural language to the distinction between class nouns and role nouns. In *Proceedings of the 27th Conference on Semantics and Linguistic Theory (SALT 27)*, ed. D Burgdorf, J Collard, S Maspong, B Stefánsdóttir, pp. 438–58. Washington, DC: Linguist. Soc. Am.