A Single-Type Semantics for the PTQ*-Fragment

Kristina LIEFKE — *Munich Center for Mathematical Philosophy* (*MCMP*), *LMU Munich*

Abstract. In (Montague, 1970), Montague defines a formal theory of linguistic meaning which interprets a small fragment of English through the use of *two* basic types of objects: individuals and propositions. In this paper, I develop a comparable semantics which only uses *one* basic type of object (hence, *single-type semantics*). Such a semantics has been suggested by Partee (2009) as a 'minimality test' for the Montagovian type system, which challenges the need for a bi-partitioned ontology. The proposed semantics captures the propositional interpretation of proper names, unifies Montague's semantic ontology, and yields insight into the apparatus of types in formal semantics.

Keywords: foundations of formal semantics, natural language metaphysics, single-type hypothesis, type theory, unification.

1. Introduction

Natural languages presuppose a rich semantic ontology. To provide an interpretation for, e.g., English, we require the existence of individuals (e.g. Bill), propositions (Bill walks), properties of individuals (walk), relations between individuals (find), and many other kinds of objects. Theories of formal semantics (paradigmatically, Montague (1970, 1973)) tame this ontological 'zoo' by casting its members into a type structure, and generating objects of a more complex type from objects of a simpler type via a variant of Church's (1940) type-forming rule:

(CT) *If $\alpha_1, \ldots, \alpha_n$ and β are types, then $(\alpha_1 \ldots \alpha_n; \beta)$ is the type for functions from ordered n-tuples of objects of the types $\alpha_1 \ldots \alpha_n$ to objects of the type β.*

In this way, Montague (1970) reduces the referents of a small subset of English (hereafter, the *PTQ*-fragment*[1]) to constructions out of *two* basic types of objects: individuals (or *entities*, type e) and propositions (or functions from indices to truth-values, type $(s;t)$). Proper names (e.g. Bill) and sentences (Bill walks) are then interpreted as individuals, respectively propositions, intransitive verbs (walk) as functions from individuals to propositions (type $(e;(s;t))$), transitive verbs (find) as functions from pairs of individuals to propositions (type $(e\,e;(s;t))$), etc.

Montague's distinction between individuals and propositions (or between individuals, indices, and truth-values, cf. (Gallin, 1975)) has today become standard in formal semantics. This is due to the resulting semantics' modeling power, and the attendant possibility of explaining a wide range of syntactic and semantic phenomena. However, the question remains whether it is also possible to construct the ontological zoo from a *single* semantic basis, which *unifies* the types e and $(s;t)$.

[1] This name is justified by the similarity of this fragment to the fragment from Montague (1973) ('PTQ-fragment') without intensional nouns (e.g. temperature, price), intransitive verbs (rise, change), and prepositions (about). I show in (forthcoming) that, by coding individual concepts as type-$((s;t);e)$ objects, we can model the full PTQ-fragment.

The assumption behind the above question, i.e. that the PTQ*-fragment has an even simpler semantic basis than the one adopted in Montague (1970), has first been proposed by Barbara Partee. In particular, Partee (2009) makes the following suggestion about the linguistic type system:

Proposition 1 (Single-Type Hypothesis)**.** *The distinction between individuals and propositions is inessential for the construction of a rich linguistic ontology. The PTQ*-fragment can be modeled through the use of one basic type of object.*

Below, we will sometimes refer to Proposition 1 as *Partee's conjecture*. This conjecture suggests the possibility of obtaining all classes of PTQ*-referents from a single basic type (dubbed 'o'), whose objects encode the semantic content of individuals and propositions. From them, objects of more complex types are constructed via a variant, **ST** (for *single-type* rule), of the rule **CT**:

(ST) *If* $\alpha_1, \ldots, \alpha_n$ *and* β *are single-type types, then* $(\alpha_1 \ldots \alpha_n; \beta)$ *is a single-type type.*

In virtue of the neutrality of the type o between Montague's types e and $(s;t)$, any semantics which satisfies Proposition 1 (hereafter, *single-type semantics*[2]) will identify basic-type objects with the values of proper names (traditionally, type e) and of sentences and complement phrases (type $(s;t)$). As a result, it will also assign the same type, $(o;o)$, to common nouns (type $(e;(s;t))$) and to complementizers and sentence adverbs (type $((s;t);(s;t))$). The types of all other syntactic categories are obtained by replacing the labels 'e' and '$(s;t)$' by 'o' in these categories' Montague types.

Partee supports her conjecture by identifying a preliminary single-type object (i.e. extensional properties of Kratzer-style situations, type $(s;t)$; cf. (Kratzer, 1989)), and giving an $(s;t)$-based model for a miniature fragment of English. This model interprets the expressions you, a snake, and see into the single-type objects ⟦you⟧, ⟦a snake⟧, and ⟦see⟧, cf. (Partee, 2009, p. 40):

⟦you⟧	the property of (being) a minimal situation containing you;
⟦a snake⟧	the property of (being) a snake-containing situation;
⟦see⟧	a function from two situation properties p_1 and p_2 to a property p_3 which holds of a situation s_3 if s_3 contains two situations, s_1 and s_2, with the properties p_1, resp. p_2, where (something in) s_1 sees (something in) s_2.

The above interpretations enable the compositional interpretation of the sentence You see a snake:

⟦You see a snake⟧	the property of (being) a situation in which you see a snake (which is contained in the situation).

Partee's model supports the possibility of providing a type-neutral interpretation of proper names and sentences. At the same time, it suggests a strategy for the model's extension to larger PTQ-like fragments. However, the nature of her paper (a short *Festschrift* contribution) prevents a demonstration of the latter. A proof of workability is left to the semantic community.

[2]Since such semantics still assume a type-hierarchy over the basic type o (by the use of the rule **ST**), they should more correctly be referred to as '*single-base-type semantics*'. I owe this observation to Jim Pryor.

This paper takes up Partee's challenge. In particular, it develops a single-type semantics for the PTQ*-fragment which systematically extends Partee's formal evidence for Proposition 1. The paper is organized as follows: To show the possibility (or desirability) of a single-type semantics, Section 2 presents different kinds of support for this semantics. Section 3 discusses the difficulty of providing a single-type semantics with a primitive basic type, and identifies Partee's basic-type choice, $(s;t)$, as the simplest Montague type which answers this difficulty. Sections 4 and 5 demonstrate the ability of an $(s;t)$-based logic to model the PTQ*-fragment. Section 6 identifies a number of constraints on the resulting single-type semantics, and draws a number of interesting conclusions for the role of the Montagovian type system. The paper closes with an evaluation of the success of single-type semantics and pointers to future work.

2. Support for Single-Type Semantics

Partee's conjecture about the possibility of a single-type semantics suggests a 'minimality test' for the Montagovian type system: If we can formulate a single-type semantics *without* reference to Montagovian individuals or propositions, we will therewith *refute* the (commonly assumed) need for two distinct basic types. If our formulation of a single-type semantics *relies* on the availability of individuals or propositions, the semantics will *support* Montague's basic-type distinction.

But our interest in single-type semantics is also motivated by other considerations: These include *empirical* considerations (which regard the greater modeling power of single-type semantics w.r.t. traditional Montague semantics; cf. (Partee, 2009)), *formal* considerations (which regard the possibility of constructing single-type models; cf. Sect. 1), and other *methodological* considerations (besides minimality testing). To illustrate possible applications of a single-type semantics – and to prime the reader's intuitions about such a semantics –, we here focus on empirical considerations:

Empirical support for Partee's conjecture lies in a demonstration of the fact that single-type semantics improves upon the modeling power of traditional Montague semantics. This improvement is a consequence of the neutralization of the distinction between the types for proper names and sentences, and the resulting existence of *fewer* 'horizontal' constraints on semantic merging.[3] To illustrate the higher modeling power of single-type semantics, we identify a number of linguistic phenomena which can be accommodated in a single-type semantics, but which defy accommodation in traditional Montague semantics. Such phenomena occur in lexical syntax, the syntax of coordination, the semantics of specification, and nonsentential speech. They consist in the neutrality of certain classes of expressions between an NP- or a CP-complement, in the possibility of coordinating proper names with complement phrases (both, Bayer 1996), in the existence of specificational sentences with a postcopular CP (Potts, 2002), and in the use of names to assert a contextually salient *proposition* about their type-e referent (Merchant, 2008).

In particular, Montague semantics is unable to interpret (at least) one of the sentences from (1), and cannot interpret the sentences from (2) and (3):

[3] As a result, transitive verbs (traditionally, type $(e\,e;(s;t))$) can apply either to a proper name or to a CP.

(1) a. Mary remembered [$_{\text{NP}}$Bill].
    b. Mary remembered [$_{\text{CP}}$that Bill was waiting for her].

(2) Mary remembered [$_{\text{NP}}$Bill] and [$_{\text{CP}}$that he was waiting for her].

(3) [$_{\text{NP}}$The problem] is [$_{\text{CP}}$that Mary hates Bill].

The inability to interpret the above sentences in Montague semantics is due to its assumption of a *functional* relation between syntactic categories and semantic types, and its assignment of *different* types (i.e. the types e, resp. $(s;t)$) to proper names and complement phrases. In virtue of the former, Montague semantics cannot associate the different occurrences of the verb remember from (1) with the distinct types $(\underline{e}\, e; (s;t))$ and $(\underline{(s;t)}\, e; (s;t))$. However, in virtue of the latter, only this assignment enables the interpretation of both members of the sentence-pair from (1).[4] The impossibility of accommodating sentences (2) and (3) in traditional Montague semantics is further due to Montague's restriction of coordination and equation to same-type expressions. Since names (or NPs) and sentences are associated with distinct types, this restriction is not satisfied by (2) and (3).

Single-type semantics solves the above problems by cancelling the different-type assignents of names and CPs. In particular, since this semantics interprets all occurrences of names and CPs in the single basic type o, the pairs of arguments from (2) and (3) will satisfy Montague's coordinability and equatability requirements, such that we can interpret these two sentences in this semantics. Since the single-type type of sentence-complement verbs, $(o\,o; o)$, allows its expressions to take a CP *or a name* as its complement, it enables the interpretation of the two sentences from (1).

Beyond the above, the desirability of a single-type semantics is supported by the possibility of accommodating recent findings in nonsentential speech: These findings show that isolated occurrences of names in a context can be interpreted as the result of applying a contextually salient property to the name's type-e referent. Thus, the name Barbara Partee – when uttered as a woman is entering the room – is interpreted as the sentence from (4b) (or (4c)) (Merchant, 2008, pp. 9, 25–26):

(4) CONTEXT: A woman is entering the room. A linguist turns to her friend, gestures towards the door, and says (a).
    a. [$_{\text{NP}}$Barbara Partee]
    b. [$_{\text{NP}}$Barbara Partee] is (the woman) entering the room.
    c. [$_{\text{NP}}$Barbara Partee] is arriving.

Since Montague semantics does not interpret proper names in the semantic type for sentences, it is unable to model phenomena like (4). Single-type semantics, which assigns the type o to both names and sentences, enables the accommodation of these phenomena.

[4] Admittedly, one could obtain the required modeling power by introducing a different lexical entry for each of the occurrences of the verb remember from (1), by assigning the different entries the types $(e\,e; (s;t))$, resp. $((s;t)\,e; (s;t))$, and by connecting them by suitable meaning-relating postulates. However, since this differentiation of entries is not reflected in lexicographic research (cf., e.g., the OED entry for remember), we hesitate to adopt this strategy.

But the empirical scope of single-type semantics is not restricted to the sentence-type interpretation of proper names. The semantics further accommodates the propositional *behavior* of names, which cannot be modeled in Montague semantics: Our sketch of single-type semantics from the introduction of this section has suggested that proper names display the semantic behavior of sentences: If names receive an interpretation in the same domain as sentences, we expect that names – like sentences – can be evaluated as true or false with respect to a given set of contextual parameters, and that they may be related[5] by semantic equivalence. This is indeed the case: In particular, in the situation from (4), the announcement (4a) – when the new arrival is, in fact, Angelika Kratzer – is a false statement, rather than a mere misidentification, cf. (Stainton, 2006, pp. 8–10, 16).

In virtue of their truth- and falsity-conditions, names of the above form will, in a given situation, be equivalent to all true sentences in this situation which carry information about the names' type-e referent. For example, if the new arrival in the above situation is indeed Barbara Partee, the utterance of the name from (4a) will be equivalent to the sentence from (4b) (or (4c)) in that situation. The obtaining of equivalence relations between sententially interpreted names and sentences (or CPs) in a given context is supported by the assertion of an equivalence between the noun and complement phrases in the sentence from (3). This relation ensures that the replacement of an NP (or CP) by its CP- (or NP-)equivalent in the complement of an NP/CP-neutral verb does not change the truth-value of the original sentence. For the arguments from (3), this is demonstrated in (5):

(5) a. Chris noticed $[_{\text{NP}}$the problem$]$.
 b. Chris noticed $[_{\text{CP}}$that Mary hates Bil$]$.

Our expectations on the semantic behavior of proper names in a single-type semantics are summarized in Proposition 2:

Proposition 2 (Assertoric interpretation of names)**.** *In a single-type semantics, proper names have truth-conditions* (Prop. 2.i)*, and are equivalent to some contextually salient sentences* (Prop. 2.ii).

The above-cited phenomena illustrate the advantages of interpreting natural language in a single-type semantics. However, the reader is admonished to note that these phenomena can also be accommodated by dropping the assumption of a functional category/type relation (Alternative 1), or by explaining the assertoric behavior of proper names with reference to pragmatics (Alternative 2). The first alternative (adopted in semantic accounts of nonsentential speech, cf. (Merchant, 2008)) assumes that certain occurrences of proper names have a *non-standard semantic content*, which results from 'shifting' the names' standard interpretation (type e) to the standard interpretation of sentences (type $(s;t)$). The second alternative (adopted in pragmatic accounts of nonsentential speech, cf. (Stainton, 2006)) assumes that certain utterances of names have a *non-standard asserted content*, which results from attributing names the illocutionary act of making an assertion. Alternative 1 follows the approach of flexible Montague grammar, cf. (Partee, 1987).

[5]to other names, or to sentences.

The possibility of accommodating the above phenomena in a *small* extension of an *existing* generalization of Montague semantics (i.e. flexible Montague grammar) suggests the relative *weakness* of the presented empirical support for single-type semantics. *Stronger* support for single-type semantics comes from *methodological* considerations. These include the complete unification of Montague's semantic ontology and the identification of new representability relations between different types of objects. A detailed presentation of these considerations is given in (Liefke, forthcoming).

3. Motivating Partee's Single-Type Choice

Our empirical arguments for Proposition 1 support Partee's identification of the single basic type with the type for properties of situations (or *propositions*, type $(s;t)$): The interpretation of names and sentences (or CPs) in this type explains the neutrality of certain verbs between an NP- or a CP-complement (cf. (1), (5)), allows for the coordination or equation of noun and complement phrases under the satisfaction of Montague's coordinability resp. equatability requirements (cf. (2), (3)), and admits the propositional interpretation of isolated names in a given context (cf. (4)). The present section gives the rationale behind our single-type choice. To this aim, we first identify the problems of a single-type semantics with a primitive (i.e. unstructured) basic type (in Sect. 3.1). We then identify the type $(s;t)$ as the simplest Montague type which solves these problems (in Sect. 3.2).

3.1. Against a 'Primitive' Single-Type Semantics

The introduction to this paper has suggested a straightforward strategy for the provision of a single-type semantics. This strategy lies in the adoption of a single basic type, o, and the replacement of (terms or objects of) the types e and $(s;t)$ in Montague semantics by (terms and objects of) the type o. The characterization of type-o objects as semantic primitives (which cannot be obtained by the application of **ST** to objects of another type) obviates the further specification of o-based models.

But the apparent simplicity of the above approach is deceptive: Specifically, the identification of the type o with a non-Montagovian type (s.t., in particular, $o \neq t$) prevents the use of the familiar truth-functional connectives like *falsum* ($\perp$, type t) or the symbol for logical implication ($\Rightarrow$, type $(\alpha\,\alpha;t)$), and disables an easy truth-evaluation of basic-type terms. These problems can be remedied by introducing (non-logical) single-type stand-ins for these connectives, and by restricting the behavior of these stand-ins through the use of meta-level axioms. However, since the formulation of these axioms still requires the assumption of a designated truth-value type t, it must proceed at the level of a *multi*-typed metatheory. For the purposes of this paper, we identify the latter with the extension of an o-based logic via the truth-value type t.

The availability of the described metatheory facilitates the truth-evaluation of type-o terms. The latter proceeds via a consideration of the membership (or inclusion) of the referents of type-o terms in type-$(o;t)$ (resp. type-o) correspondents of indices. These evaluation strategies are derived from the evaluation of proposition-denoting formulas in Pollard's (2008) *constructed worlds theory* and in Fine's (1982) theory of *worlds as facts*. However, since these strategies still require the introduction of a new meta- *and* object theory, since the representation of indices in the types $(o;t)$, resp. o requires some complex coding machinery, and since 'primitive' single-type semantics prevent the easy identification of a name's sentential equivalents (cf. (4)), we refrain from their adoption.

The above observations motivate our attempt to identify the single basic type o with a particular Montague type. But the adoption of such a 'familiar' single-type type has many other advantages: For example, the adoption of an 'o-defining' Montague type will induce an algebraic structure on the basic-type domain (which will, in turn, facilitate the interpretation of linguistic connectives), and will enable a metalevel definition of the designated single-type constants. Beyond formal reasons, the interpretation of the type o as a concrete Montague type will lend our single-type semantics intuitive content, and will enable the identification of new representational relations between different types of Montagovian objects. We will identify some of these relations in Section 6. However, we first show the suitability of Partee's type $(s;t)$ as a single basic type (in Sect. 3.2).

3.2. Why the Type $(s;t)$?

The adequacy of the type $(s;t)$ as a single basic type for the modeling of the PTQ*-fragment lies in its satisfaction of the semantic requirements from Properties (i) to (iv):

(i) Familiarity *The basic type figures in the semantic analysis of some linguistic phenomenon.*

(ii) Conjoinability *The single-type domain has an algebraic structure.*

(iii) Representability *All Montagovian objects can be represented via single-type objects.*

(iv) Simplicity *Given its satisfaction of Properties (i) to (iii), the single basic type is obtained from the types e and $(s;t)$ through the least number of* **CT***-applications.*

Property (i) ensures the proximity of single-type semantics to mainstream formal semantics. Property (ii) allows the interpretation of linguistic connectives as algebraic operations. Property (iii) enables the bootstrapping of representations of all Montagovian objects from objects of the single basic type. Property (iv) guarantees the low semantic complexity of single-type objects.

Since the type $(s;t)$ is a common choice for the interpretation of sentences, it satisfies the Familiarity requirement from Property (i). Since there is an algebraic structure on the truth-value type t (s.t. it is possible to lift all algebraic operations to domains of some type $(\alpha_1 \dots \alpha_n;t)$), the type $(s;t)$ further satisfies the Conjoinability requirement from Property (ii).

That the type $(s;t)$ satisfies the Representability requirement from Property (iii) is ensured by its identity with Montague's type for propositions, and by the existence of an injective function from individuals to propositions (s.t. single-type representations of type-e objects preserve the distinctions between these objects). Every proposition φ can then be represented by itself (cf. (3.1)). Every individual a can be represented by (the characteristic function of) the set of all indices in which it exists (cf. (3.2)):

$$\{w_s \mid w \in \varphi\} \tag{3.1}$$

$$\{w_s \mid a \text{ exists in } w\} \tag{3.2}$$

In (3.2), an individual's *existence in* an index is understood as the individual's '*being in* an index' (i.e. as its *inhabitance of* that index). This understanding of existence corresponds to our pre-theoretical intuitions about existence, and to the understanding of *concreteness* in fixed-domain quantified modal logic, cf. (Linsky and Zalta, 1994). To ensure the injectivity of the individual-representations from (3.2), we assume that no two individuals exist exactly in the same indices. This assumption is common in Situation Semantics, cf. (Muskens, 1995, pp. 70–71), and in the semantics of quantified modal logic.

The representation of individuals in the semantic type for propositions enables the truth-evaluation of individual-designators (w.r.t. an index) and the identification of entailment and equivalence relations between pairs of individuals, and between an individual and a proposition. Thus, the designator of some individual a will be *true* (or *false*) at the current index @ in an $(s;t)$-based single-type semantics that employs the representation strategy from (3.2) if a exists (resp. does not exist) in @. The type-$(s;t)$ representation of the individual a will *entail* (resp. be entailed by) some proposition φ if the designator of φ (or a) is true at all indices at which a (φ) is true and if the designator of a (resp. φ) is false at all indices at which φ (or a) is false. The relations of single-type truth and (mutual) entailment will be formalized in Section 4.

Since the type for individuals, e, does not satisfy the Representability requirement from Property (iii),[6] and since the types e and $(s;t)$ are the *only* basic types in the semantic ontology from (Montague, 1970), the type $(s;t)$ is also the *simplest* suitable single-type type (cf. Property (iv)).

Its satisfaction of Properties (i) to (iv) identifies the type $(s;t)$ as the 'best' (or most suitable) single-type candidate. However, the adequate interpretation of natural language in an $(s;t)$-based semantics further requires a *partial* interpretation of the type $(s;t)$ (i.e. as *partial* sets of *situations*). This is due to our reference to an individual's *existence* in the representation from (3.2), and to our wish to preserve the standard behavior of negation in single-type semantics: Conservative semantics evaluate both the result, Fa, and the negation, $\neg Fa$, of the result of attributing a contextually salient property F to an individual a at an index w where a does not exist as '**F**'. For example, since Vulcan does not exist in the actual world, such semantics evaluate both the sentence Vulcan is a planet and the sentence Vulcan is not a planet as *false*. However, this violates the familiar axioms for negation.[7] Since the truth-combination **N** ('*neither-true-nor-false*') is uncomplemented (s.t. $-\mathbf{N} = \mathbf{N}$), the evaluation of both Fa and $\neg Fa$ at w as '**N**' preserves the familiar behavior of negation.

[6] This is due to the fact that there are commonly more propositions than individuals, s.t. there is not injective function from the former to the latter.

[7] According to the axiom of Top and Bottom, if $Fa(w) = \mathbf{F}$, then $\neg Fa(w) = \mathbf{T}$.

This completes our motivation of the type $(s;t)$ as an adequate single-type candidate. The next three sections incorporate the representational strategies from (3.1) and (3.2) into a formal single-type semantics. Our provision of this semantics uses Montague's (1973) method of indirect interpretation, which proceeds via the compositional translation of some subset of natural language (here, the PTQ*-fragment) into some logical language: Correspondingly, we will first define a general class of languages and models of the single-type logic STY_1^3 (in Sect. 4). We will then specify the translation rules which send logical forms of the PTQ*-fragment to STY_1^3 terms (in Sect. 5, 6).

4. The Single-Type Logic STY_1^3

Our previous considerations have suggested the identification of single-type semantics with a model of an $(s;t)$-based subsystem of an n-ary partial variant, TY_2^3, of Gallin's (1975) logic TY_2. This semantics constructs all of its objects from properties of situations (or *propositions*, type $(s;t)$).

The name of our single-type logic, 'STY_1^3', follows Gallin's naming convention for type logics. In particular, the subscript '1' is warranted by the construction of the lowest (or 'basic'[8]) STY_1^3 type $(s;t)$ from $1+t$ basic Gallin types. The letter 'S' (for *single-type*) distinguishes our theory from Church's (1940) Simple Theory of Types, TY_1. The superscript '3' indicates the partiality of the logic's models.

From the type $(s;t)$, all other types of the logic STY_1^3 are defined via the rule **ST** as follows:

Definition 1 (STY_1^3 types)**.** The set $\mathsf{1Type}$ of STY_1^3 *types* is the smallest set of strings such that, for $0 \leq n \in \mathbb{N}$, if $\alpha_1, \ldots, \alpha_n \in \mathsf{1Type}$, then $(\alpha_1 \ldots \alpha_n; (s;t)) \in \mathsf{1Type}$.

A language L for the logic STY_1^3 is a countable set $\cup_{\alpha \in \mathsf{1Type}} L_\alpha$ of uniquely typed non-logical constants. These include a constant for the *absurd* (or *impossible*) proposition, $\circled{\bot}$ (type $(s;t)$). For every STY_1^3 type α, we further assume a countable set $\mathcal{V}_\alpha$ of uniquely typed variables, with '$\cup_{\alpha \in \mathsf{1Type}} \mathcal{V}_\alpha$' abbreviated as '$\mathcal{V}$'. From these expressions, we form complex terms inductively with the help of functional application, abstraction, and the non-logical constant $\dot{\Rightarrow}$.

Definition 2 (STY_1^3 terms)**.** Let $\alpha_1, \ldots, \alpha_n, \beta \in \mathsf{1Type}$. The set T_α of STY_1^3 *terms* of the type α is defined as follows:

(i) $L_\alpha, \mathcal{V}_\alpha \subseteq T_\alpha$, $\circled{\bot} \in T_{(s;t)}$;
(ii) If $\boldsymbol{A} \in T_{(\beta\alpha_1 \ldots \alpha_n;(s;t))}$ and $\boldsymbol{B} \in T_\beta$, then $(\boldsymbol{A}(\boldsymbol{B})) \in T_{(\alpha_1 \ldots \alpha_n;(s;t))}$;
(iii) If $\boldsymbol{A} \in T_{(\alpha_1 \ldots \alpha_n;(s;t))}$ and $\boldsymbol{x} \in \mathcal{V}_\beta$, then $(\lambda \boldsymbol{x}.\boldsymbol{A}) \in T_{(\beta\alpha_1 \ldots \alpha_n;(s;t))}$;
(iv) If $\boldsymbol{A}, \boldsymbol{B} \in T_\alpha$, then $(\boldsymbol{A} \dot{\Rightarrow} \boldsymbol{B}) \in T_{(s;t)}$.

The constants $\circled{\bot}$ and $\dot{\Rightarrow}$ are single-type stand-ins for falsum ($\bot$, type t) and for logical implication ($\Rightarrow$, type $(\alpha\,\alpha;t)$), respectively. Their introduction is required by the unavailability of the TY_2^3 con-

[8]The description of the type $(s;t)$ as the 'basic' type of the logic STY_1^3 is, at best, unfortunate. Yet, since the two uses of the adjective *basic* are distinguished by their respective contexts, its ambiguity is harmless.

stants $\bot$ and $\Rightarrow$ in the logic STY_1^3 (by the absence of the type t; cf. Sect. 3.1), and by their need in a single-type theory. To ensure that $\textcircled{\bot}$ and $\dot{\Rightarrow}$ display the semantic behavior of $\bot$ and $\Rightarrow$, we will later define the former in terms of the latter (in Sect. 6).

From $\textcircled{\bot}$ and $\dot{\Rightarrow}$, single-type stand-ins of the remaining TY_2^3 connectives are obtained via (single-type variants of) the definitions from (Henkin, 1950). In particular, the STY_1^3 proxies $\textcircled{\top}$; $\boxdot$ and $\diamondsuit$; $\bigwedge$ and $\bigvee$; and $\doteq$, $\dot{\neq}$, $\dot{\rightarrow}$, and $\dot{\leftrightarrow}$ of TY_2^3 *verum* ($\top$), the modal box and diamond operators ($\Box$, $\Diamond$), the universal and existential quantifiers ($\forall$, $\exists$), and the symbols for equality ($=$), inequality ($\neq$), material implication ($\rightarrow$), and biimplication ($\leftrightarrow$) are non-logical constants of the types (s,t), $((s;t);(s;t))$, $(\alpha\,(s;t);(s;t))$, resp. $(\alpha\,\alpha;(s;t))$, where $\alpha \in \mathsf{1Type}$. Our use of the same symbol for TY_2^3 and STY_1^3 conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) is motivated by the availability of these connectives in the logic STY_1^3, s.t. they have their familiar type (i.e. $(\alpha\,\alpha;\alpha)$, resp. $(\alpha;\alpha)$).

STY_1^3 terms will sometimes be subscripted by their type. We adopt the usual conventions for binding, freedom, and closure. Substitution is standardly defined.

This completes our specification of STY_1^3 types and terms. We next turn to the definition of general STY_1^3 frames and models.

General STY_1^3 models are $\mathsf{1Type}$-restricted variants of general models for the logic TY_2^3, which consist of a frame F, an interpretation function I_F, and an assignment g_F. In particular, general STY_1^3 frames are defined as follows, where S is the TY_2^3 set of situations, and where $\mathbf{3}$ is the TY_2^3 set of the truth-combinations $\mathbf{T}, \mathbf{F}$, and $\mathbf{N}$:

Definition 3 (General STY_1^3 frames)**.** A *general frame* for STY_1^3 is a set $F = \{D_\alpha \mid \alpha \in \mathsf{1Type}\}$ of pairwise disjoint non-empty sets s.t. $D_{(\alpha_1 \ldots \alpha_n;(s;t))} \subseteq \{f \mid f : (D_{\alpha_1} \times \cdots \times D_{\alpha_n}) \rightarrow (S \rightarrow \mathbf{3})\}$ for all STY_1^3 types $\alpha_1, \ldots, \alpha_n$.

In virtue of our identification of the TY_2^3 domain $D_{(s;t)}$ with a subset of the space $(S \rightarrow \mathbf{3})$, the ground domain of the logic STY_1^3 will contain partial objects, which are ordered with respect to their degrees of truth and definedness. As a result, STY_1^3 objects will have the desired semantic properties from Section 3.2. To enable the recursive axiomatizability of STY_1^3 entailment, we associate STY_1^3 domains with subsets of their associated function spaces.

Interpretation functions $I_F : L \rightarrow F$ assign to each non-logical STY_1^3 constant $\boldsymbol{c}_\alpha$ a type-identical denotation in the frame F, s.t. $I_F(\boldsymbol{c}_\alpha) \in D_\alpha$. Assignments $g_F : \mathcal{V} \rightarrow F$ are analogously defined. Given an object $d \in D_\alpha$ and variables $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}_\alpha$, we define $g_F[d/\boldsymbol{x}]$ by letting $g_F[d/\boldsymbol{x}](\boldsymbol{x}) = d$ and $g_F[d/\boldsymbol{x}](\boldsymbol{y}) = g_F(\boldsymbol{y})$ if $\boldsymbol{x} \neq \boldsymbol{y}$, where $=$ and $\neq$ are symbols of the metalanguage (here, TY_2^3). The set of all assignments g_F with respect to a STY_1^3 frame F is denoted by '$\mathcal{G}_F$'.

On the basis of the above, general STY_1^3 models are defined as follows:

Definition 4 (General STY_1^3 models)**.** A *general model* for STY_1^3 is a triple $M_F = \langle F, I_F, V_F \rangle$, where each $D_\alpha \in F$ is the carrier of a *complete De Morgan algebra*, $\langle D_\alpha, \cap, \cup, -, 0, 1 \rangle$. The function $V_F : (\mathcal{G}_F \times \cup_\alpha T_\alpha) \to F$ is such that

(i) $V_F(g_F, \boldsymbol{c}) := I_F(\boldsymbol{c})$ if $\boldsymbol{c} \in L$, $V_F(g_F, \boldsymbol{x}) := g_F(\boldsymbol{x})$ if $\boldsymbol{x} \in \mathcal{V}$;

(ii) $V_F(g_F, \boldsymbol{A}(\boldsymbol{B})) := V_F(g_F, \boldsymbol{A})\big(V_F(g_F, \boldsymbol{B})\big)$;

(iii) $V_F(g_F, \lambda \boldsymbol{x}_\beta.\boldsymbol{A}) :=$ the fct. $f_{(\alpha_1 \ldots \alpha_n;(s;t))}$ s.t., $\forall d_\beta$, $f(d) = V_F(g_F[d/\boldsymbol{x}], \boldsymbol{A})$.

Clause (ii) comprises a definition of the interpretation of $\circledperp$- and $\Rrightarrow$-involving terms from Definition 2. The algebraic structure on STY_1^3 domains is a consequence of the De Morgan algebra on the set **3**, and of our definition of STY_1^3 types.

As desired, the logic STY_1^3 enables the truth-definition of its basic-type terms. This is due to the identification of the basic STY_1^3 type with the type for propositions $(s;t)$, and the definition of TY_2^3 truth and falsity for terms of this type. However, since the logic STY_1^3 does not command designated types for situations (s) or truth-combinations (t), the evaluation of the truth or falsity of basic STY_1^3 terms proceeds in models of the logic TY_2^3.

The truth (or falsity) of basic-type STY_1^3 terms is defined below. In the definition, an 'embedded' STY_1^3 model M_F and assignment function g_F of a general TY_2^3 model M_F (abbr. 'M_2') and assignment g_F (abbr. 'g_2') are understood as the result of restricting (the relevant constituents of) M_2 and g_2 to STY_1^3 terms and frames (s.t. $M_F = M^{2\restriction 1\mathrm{Type}}$ and $g_F = g^{2\restriction 1\mathrm{Type}}$).

Definition 5 (STY_1^3 truth)**.** An STY_1^3 term $\boldsymbol{A}_{(s;t)}$ is *true* (or *false*) at a situation w in an embedding TY_2^3 model, M^2, of a general STY_1^3 model M_F under an embedded assignment, g^2, of the assignment g_F iff $w \models_M \boldsymbol{A}$ (resp. $w =\!\!|_M \boldsymbol{A}$).

In the logic STY_1^3, entailment between basic-type terms is defined through the partial order, $\subseteq$, on the TY_2^3 set of truth-combinations as follows:

Definition 6 (STY_1^3 entailment)**.** A set of STY_1^3 terms $\Gamma := \{\gamma \mid \gamma \in T_{(s;t)}\}$ *entails* a set of STY_1^3 terms $\Delta := \{\delta \mid \delta \in T_{(s;t)}\}$, i.e. $\Gamma \models_g \Delta$, iff, for all general STY_1^3 models M_F and assignments g_F, $\bigcap_{\gamma \in \Gamma} V_F(g_F, \gamma) \subseteq \bigcup_{\delta \in \Delta} V_F(g_F, \delta)$.

The subscript 'g' of the entailment relation refers to the generality of STY_1^3 models. We call a term γ *g-valid* if $\models_g \gamma$ for every *general* STY_1^3 model M_F and g_F. Definition 6 allows the definition of semantic STY_1^3 equivalence in terms of mutual STY_1^3 entailment.

To enable a proof-theoretic characterization of STY_1^3 entailment, we use the TY_2^3 symbol for logical implication, $\Rightarrow$. Its behavior is characterized by single-type variants of the sequent rules from (Muskens, 1995). The logic STY_1^3 has the expected metamathematical properties (e.g. Soundness, Completeness, Compactness).

This completes our presentation of the single-type logic STY_1^3. We next show that a designated model of this logic interprets the PTQ*-fragment (cf. Prop. 1), accommodates the mentioned phenomena from lexical syntax, syntactic coordination, and specification (cf. Sect. 2), and accommodates the truth-evaluability of proper names (cf. Prop. 2).

5. STY_1^3-Based Single-Type Semantics

To identify the STY_1^3 interpretations of logical PTQ*-forms, we first specify the particular STY_1^3 language $\mathcal{L}$ and frame $\mathcal{F}$. The members of $\mathcal{L}$ are specified in Table 1. Our conventions for the use of STY_1^3 variables are introduced in Table 2. Since some of the designated STY_1^3 constants from Definition 2 will figure in our translation of logical PTQ*-forms, we assume their membership in $\mathcal{L}$. To enable the translation of the example sentences from (1) to (4), we extend the PTQ*-fragment via the lexical constituents of these sentences.[9] For better visibility, we sometimes replace round by square brackets in the notation for types.

CONSTANT	STY_1^3 TYPE
$\neg$	$[[\alpha_1 \ldots \alpha_n; (s;t)]\, \alpha_1 \ldots \alpha_n; (s;t)]$
$\wedge, \vee$	$[[\alpha_1 \ldots \alpha_n; (s;t)]\, [\alpha_1 \ldots \alpha_n; (s;t)]\, \alpha_1 \ldots \alpha_n; (s;t)]$
$\bigwedge, \bigvee$	$[\alpha\, (s;t); (s;t)]$
$\dot{\Rightarrow}, \doteq, \dot{\neq}, \dot{\rightarrow}, \dot{\leftrightarrow}$	$[\alpha\, \alpha; (s;t)]$
Ⓣ, Ⓛ, ***john***, ***mary***, ***bill***, ***partee***, ***w***	$(s;t)$
$\boxdot$, $\diamond$, ***man***, ***woman***, ***park***, ***fish***, ***pen***, ***unicorn***, ***problem***	$[(s;t); (s;t)]$
run, ***walk***, ***talk***, ***wait***, ***arrive***, ***E***	$[(s;t); (s;t)]$
find, ***lose***, ***eat***, ***love***, ***date***, ***remember***, ***hate***, ***believe***, ***assert***	$[(s;t)\, (s;t); (s;t)]$
rapidly, ***slowly***, ***voluntary***, ***allegedly***, ***try***, ***wish***	$[[(s;t); (s;t)]\, (s;t); (s;t)]$
in, ***for***	$[(s;t)\, [(s;t); (s;t)]\, (s;t); (s;t)]$
seek, ***conceive***	$[[[(s;t); (s;t)]; (s;t)]\, (s;t); (s;t)]$

Table 1: $\mathcal{L}$ constants.

VARIABLE	STY_1^3 TYPE	VARIABLE	STY_1^3 TYPE
$\boldsymbol{x}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}, \boldsymbol{z}$	$(s;t)$	$\boldsymbol{Q}, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$	$[[(s;t); (s;t)]; (s;t)]$
$\boldsymbol{p}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n, \boldsymbol{q}, \boldsymbol{r}$	$(s;t)$	$\boldsymbol{L}, \boldsymbol{L}_1, \ldots, \boldsymbol{L}_n$	$[[[(s;t); (s;t)]; (s;t)]\, (s;t); (s;t)]$
$\boldsymbol{P}, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$	$[(s;t); (s;t)]$	$\boldsymbol{R}, \boldsymbol{R}_1, \ldots, \boldsymbol{R}_n$	$[\alpha_1 \ldots \alpha_n; (s;t)]$

Table 2: STY_1^3 variables.

To give a general translation of expressions from the PTQ*-fragment, we let the frame $\mathcal{F}$ be very large, such that it contains possible values for all elements in $\mathcal{L}$. The function $\mathcal{I}_\mathcal{F} : \mathcal{L} \to \mathcal{F}$ respects the way in which different content words are conventionally related.[10] The specific role of the interpretation function $\mathcal{I}_\mathcal{F}$ will be discussed in Section 6.

[9] For convenience, we hereafter use the term '*PTQ*-fragment*' for the union of the constituents from (1) to (4) and the difference between the PTQ-fragment and the set of intensional nouns, intransitive verbs, and prepositions.

[10] Thus, $\mathcal{I}_\mathcal{F}$ is such that $\mathcal{I}_\mathcal{F}(\lambda \boldsymbol{x}.\boldsymbol{bill} \doteq \boldsymbol{x}) \subseteq \mathcal{I}_\mathcal{F}(\boldsymbol{man})$, where $\lambda \boldsymbol{x}.\boldsymbol{bill} \doteq \boldsymbol{x}$ and ***man*** are the TY_0 translations of the phrase be Bill and the common noun man, respectively.

We identify Logical Form (LF) with the component of syntactic representation that is interpreted in STY_1^3 models. Logical forms are translated into STY_1^3 terms via the process of type-driven translation, cf. (Klein and Sag, 1985). The latter proceeds in two steps, by first defining the translations of lexical elements (or *words*), and then defining the translations of non-lexical elements compositionally from the translation of their constituents.

Definition 7 (Basic STY_1^3 translations)**.** The base rule of type-driven translation translates the lexical PTQ*-elements into the following STY_1^3 terms, where $\vec{\boldsymbol{X}} = \boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ is a sequence of STY_1^3 variables of the types $\alpha_1, \dots, \alpha_n$. For reasons of space, we only translate some representative constants. Members of the same (sub-)category will receive an analogous translation:

Bill	$\rightsquigarrow$	$\boldsymbol{bill}$;	Mary	$\rightsquigarrow$	$\boldsymbol{mary}$;
Barbara Partee	$\rightsquigarrow$	$\boldsymbol{partee}$;	John	$\rightsquigarrow$	$\boldsymbol{john}$;
unicorn	$\rightsquigarrow$	$\boldsymbol{unicorn}$;	woman	$\rightsquigarrow$	$\boldsymbol{man}$;
problem	$\rightsquigarrow$	$\boldsymbol{problem}$;	waits	$\rightsquigarrow$	$\boldsymbol{wait}$;
arrives	$\rightsquigarrow$	$\boldsymbol{arrive}$;	hates	$\rightsquigarrow$	$\lambda \boldsymbol{Q} \lambda \boldsymbol{x}.\boldsymbol{Q}(\lambda \boldsymbol{y}.\boldsymbol{hate}(\boldsymbol{y}, \boldsymbol{x}))$;
exists	$\rightsquigarrow$	$\boldsymbol{E}$;	remembers	$\rightsquigarrow$	$\lambda \boldsymbol{Q} \lambda \boldsymbol{x}.\boldsymbol{Q}(\lambda \boldsymbol{y}.\boldsymbol{remember}(\boldsymbol{y}, \boldsymbol{x}))$;
seeks	$\rightsquigarrow$	$\boldsymbol{seek}$;	is	$\rightsquigarrow$	$\lambda \boldsymbol{Q} \lambda \boldsymbol{x}.\boldsymbol{Q}(\lambda \boldsymbol{y}.\boldsymbol{x} \doteq \boldsymbol{y})$;
that	$\rightsquigarrow$	$\lambda \boldsymbol{p}.\boldsymbol{p}$;	believes	$\rightsquigarrow$	$\lambda \boldsymbol{p} \lambda \boldsymbol{Q}.\lambda \boldsymbol{Q}(\lambda \boldsymbol{x}.\boldsymbol{believe}(\boldsymbol{p}, \boldsymbol{x}))$;
t_n/it$_n$	$\rightsquigarrow$	$\boldsymbol{x}_n$, for each n ;	for	$\rightsquigarrow$	$\lambda \boldsymbol{Q} \lambda \boldsymbol{P} \lambda \boldsymbol{x}.\boldsymbol{Q}(\lambda \boldsymbol{y}.\boldsymbol{for}(\boldsymbol{y}, \boldsymbol{P}, \boldsymbol{x}))$;
(s)he$_n$	$\rightsquigarrow$	$\boldsymbol{x}_n$, for each n ;	possibly	$\rightsquigarrow$	$\lambda \boldsymbol{p}.\dot{\Diamond} \boldsymbol{p}$;
rapidly	$\rightsquigarrow$	$\lambda \boldsymbol{P} \lambda \boldsymbol{x}.\boldsymbol{rapidly}(\boldsymbol{P}, \boldsymbol{x}) \wedge \boldsymbol{P}(\boldsymbol{x})$;	a	$\rightsquigarrow$	$\lambda \boldsymbol{P}_1 \lambda \boldsymbol{P} \bigvee \boldsymbol{x}.\boldsymbol{P}_1(\boldsymbol{x}) \wedge \boldsymbol{P}(\boldsymbol{x})$;
and	$\rightsquigarrow$	$\lambda \boldsymbol{R}_1 \lambda \boldsymbol{R}\, \lambda \vec{\boldsymbol{X}}.\boldsymbol{R}(\vec{\boldsymbol{X}}) \wedge \boldsymbol{R}_1(\vec{\boldsymbol{X}})$;	every	$\rightsquigarrow$	$\lambda \boldsymbol{P}_1 \lambda \boldsymbol{P} \bigwedge \boldsymbol{x}.\boldsymbol{P}_1(\boldsymbol{x}) \dot{\rightarrow} \boldsymbol{P}(\boldsymbol{x})$;
not	$\rightsquigarrow$	$\lambda \boldsymbol{R}\, \lambda \vec{\boldsymbol{X}}.\neg \boldsymbol{R}(\vec{\boldsymbol{X}})$;	or	$\rightsquigarrow$	$\lambda \boldsymbol{R}_1 \lambda \boldsymbol{R}\, \lambda \vec{\boldsymbol{X}}.\boldsymbol{R}(\vec{\boldsymbol{X}}) \vee \boldsymbol{R}_1(\vec{\boldsymbol{X}})$;
the	$\rightsquigarrow$	$\lambda \boldsymbol{P}_1 \lambda \boldsymbol{P} \bigvee \boldsymbol{x} \bigwedge \boldsymbol{y}.(\boldsymbol{P}_1(\boldsymbol{y}) \dot{\leftrightarrow} \boldsymbol{x} \doteq \boldsymbol{y}) \wedge \boldsymbol{P}(\boldsymbol{x})$			

Above, t_n represents the trace of a moved constituent in an LF that is translated as a free variable $\boldsymbol{x}_n$.

Definition 7 enables the single-type interpretation of all Logical Form-constituents of the PTQ*-fragment. Some example translations are given below. The reader will observe that the latter share the form of the sentences' translations from Montague (1973), cf. (Gallin, 1975).

$$[_{\mathrm{S}}[_{\mathrm{NP}}\text{Barbara Partee}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \boldsymbol{arrive}(\boldsymbol{partee}) \tag{5.1}$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}[_{\mathrm{DET}}\text{a}][_{\mathrm{N}}\text{woman}]][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \bigvee \boldsymbol{x}.\boldsymbol{woman}(\boldsymbol{x}) \wedge \boldsymbol{arrive}(\boldsymbol{x}) \tag{5.2}$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}[_{\mathrm{DET}}\text{every}][_{\mathrm{N}}\text{woman}]][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \bigwedge \boldsymbol{x}.\boldsymbol{woman}(\boldsymbol{x}) \dot{\rightarrow} \boldsymbol{arrive}(\boldsymbol{x}) \tag{5.3}$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}[_{\mathrm{DET}}\text{the}][_{\mathrm{N}}\text{woman}]][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \bigvee \boldsymbol{x} \bigwedge \boldsymbol{y}.(\boldsymbol{woman}(\boldsymbol{y}) \dot{\leftrightarrow} \boldsymbol{x} \doteq \boldsymbol{y}) \wedge \boldsymbol{arrive}(\boldsymbol{x}) \tag{5.4}$$

$$[_{\mathrm{S}}[\text{John}][_{\mathrm{VP}}[_{\mathrm{TV}}\text{seeks}][_{\mathrm{NP}}[\text{a}][_{\mathrm{N}}\text{unicorn}]]]] \rightsquigarrow \boldsymbol{seek}([\lambda \boldsymbol{P} \bigvee \boldsymbol{x}.\boldsymbol{unicorn}(\boldsymbol{x}) \wedge \boldsymbol{P}(\boldsymbol{x})], \boldsymbol{john}) \tag{5.5}$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}[\text{a}][_{\mathrm{N}}\text{unicorn}]]^0\, [_{\mathrm{S}}[\text{John}][_{\mathrm{VP}}[\text{seeks}]\ t_0]]] \rightsquigarrow \bigvee \boldsymbol{x}.\boldsymbol{unicorn}(\boldsymbol{x}) \wedge \boldsymbol{seek}([\lambda \boldsymbol{P}.\boldsymbol{P}(\boldsymbol{x})], \boldsymbol{john}) \tag{5.6}$$

Notably, in virtue of its same-type assignment to proper names and complement phrases, our STY_1^3-based single-type semantics enables the translation of both guises of NP/CP-complementneutral verbs (cf. (1a), (1b); in (5.7), (5.8)), and of NP/CP-coordinations (cf. (2); in (5.9)):[11]

$$[_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Mary}][_{\mathrm{VP}}[_{\mathrm{TV}}\textsf{remembers}][_{\mathrm{NP}}\textsf{Bill}]]] \rightsquigarrow \boldsymbol{remember}\,(\boldsymbol{bill}, \boldsymbol{mary}) \tag{5.7}$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Mary}]^1\ [_{\mathrm{S}}\ t_1\ [_{\mathrm{VP}}[_{\mathrm{TV}}\textsf{remembers}][_{\mathrm{CP}}[_{\mathrm{C}}\textsf{that}][_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Bill}][_{\mathrm{VP}}[\textsf{waits}][_{\mathrm{PP}}[_{\mathrm{P}}\textsf{for}][_{\mathrm{NP}}\textsf{she}_1]]]]]]]] \tag{5.8}$$
$$\rightsquigarrow \boldsymbol{remember}\,(\boldsymbol{for}\,(\boldsymbol{mary}, \boldsymbol{wait}, \boldsymbol{bill}), \boldsymbol{mary})$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Mary}]^1\ [_{\mathrm{S}}\ t_1\ [_{\mathrm{VP}}[_{\mathrm{TV}}\textsf{remembers}][_{\mathrm{NP}}\textsf{Bill}]]]] \tag{5.9}$$
$$[[_{\mathrm{CONJ}}\textsf{and}][_{\mathrm{CP}}[_{\mathrm{C}}\textsf{that}][_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Bill}][_{\mathrm{VP}}[_{\mathrm{IV}}\textsf{waits}][_{\mathrm{PP}}[_{\mathrm{P}}\textsf{for}][_{\mathrm{NP}}\textsf{she}_1]]]]]]$$
$$\rightsquigarrow \boldsymbol{remember}\,\big((\boldsymbol{bill} \wedge \boldsymbol{for}\,(\boldsymbol{mary}, \boldsymbol{wait}, \boldsymbol{bill})), \boldsymbol{mary}\big)$$

The identification of the CP-type with the type for objects in the quantificational domain of definite determiner phrases further enables the interpretation of CP-equatives (cf. (3); in (5.10)):

$$[_{\mathrm{S}}[_{\mathrm{NP}}[_{\mathrm{DET}}\textsf{the}][_{\mathrm{N}}\textsf{problem}]][_{\mathrm{VP}}[_{\mathrm{TV}}\textsf{is}][_{\mathrm{CP}}[_{\mathrm{C}}\textsf{that}][_{\mathrm{S}}[_{\mathrm{NP}}\textsf{Mary}][_{\mathrm{VP}}[_{\mathrm{TV}}\textsf{hates}][_{\mathrm{NP}}\textsf{Bill}]]]]]] \tag{5.10}$$
$$\rightsquigarrow \bigvee \boldsymbol{x} \bigwedge \boldsymbol{y}.\,(\boldsymbol{problem}\,(\boldsymbol{y}) \dot{\leftrightarrow} \boldsymbol{x} \doteq \boldsymbol{y}) \wedge \boldsymbol{x} \doteq \boldsymbol{hate}\,(\boldsymbol{bill}, \boldsymbol{mary})$$

The above examples suggest that our STY_1^3-based semantics is a conservative extension of traditional Montague semantics: Like Montague semantics, it enables the interpretation of the PTQ⋆- (or the PTQ-) fragment. Our semantics improves upon Montague semantics by allowing the interpretation of sentences of the form of (1) to (3). However, until now, the semantics has been unable to predict equivalence relations between proper names and sentences (cf. (4)). This is due to our restriction to an LF's semantic *type* (rather than to a particular object of that type). As a result, we can only predict equivalence (or entailment) relations between pairs of logical forms of same-category expressions whose members receive an interpretation as 'algebraically related' objects (e.g. between the forms Partee arrives and It is not the case that Partee does not arrive, and Partee and Partee and (Partee or Mary)). But our accommodation of Proposition 2.ii requires exactly the equivalence of 'algebraically unrelated' logical forms from different categories (for (4), the equivalence of the forms Partee and Partee arrives).

6. Constraints on STY_1^3-Based Single-Type Semantics

To identify equivalence relations between pairs of logical forms of different categories, we impose a number of constraints on the interpretation of primitive STY_1^3 constants (in Def. 8). These constraints specify, for every member of $\mathcal{L}$, which element in the 'embedding' TY_2^3 model it designates. From these constraints, constraints on the interpretation of the remaining STY_1^3 terms from Definition 7 are then obtained via a compositional definition.

For representative STY_1^3 terms from Table 1, these constraints are given in Definition 8. In this definition, we use the designated TY_2^3 constants from Table 3. Our typing conventions for TY_2^3 variables are given in Table 4. In Table 3, the predicate E applies to an individual- and a situation-denoting term to assert the existence of the individual at the situation. We assume that $\mathcal{L} \subseteq \mathcal{L}^2$ and $\mathcal{V} \subseteq \mathcal{V}^2$. The designated TY_2^3 frame $\mathcal{F}^2$ and function $\mathcal{I}_{\mathcal{F}}$ are s.t. $\mathcal{F} = \mathcal{F}^{2\restriction 1\mathsf{Type}}$ and $\mathcal{I}_{\mathcal{F}} = \mathcal{I}_{\mathcal{F}}^{\restriction 1\mathsf{Type}}$.

[11] Since it is not currently relevant, we neglect the tense and aspect of the original examples.

CONSTANT	TY_2^3 TYPE	CONSTANT	TY_2^3 TYPE
john, *mary*, *bill*, *partee*	e	*find*, *remember*, *hate*	$(e\,e;(s;t))$
believe, *assert*, ...	$[(s;t)\,e;(s;t)]$	*seek*, *conceive*	$[[(e;(s;t));(s;t)]\,e;(s;t)]$
rapidly, *allegedly*, ...	$[(e;(s;t))\,e;(s;t)]$	*in*, *for*	$[e\,(e;(s;t))\,e;(s;t)]$
man, *woman*, *unicorn*, *problem*, *wait*, *arrive*, E, ...			$(e;(s;t))$

Table 3: Non-logical $\mathcal{L}^2$-constants.

VARIABLE	TY_2^3 TYPE	VARIABLE	TY_2^3 TYPE
$i, j, k, k_1, \ldots, k_n$	s	$x, x_1, \ldots, x_n, y, z$	e
$p, p_1, \ldots, p_n, q, r$	$(s;t)$	$P, P_1, \ldots, P_n$	$(e;(s;t)$
$Q, Q_1, \ldots, Q_n$	$[[e;(s;t)];(s;t)]$	$L, L_1, \ldots, L_n$	$[[[(e;(s;t));(s;t)];(s;t)];(s;t)]$

Table 4: TY_2^3 variables.

Definition 8 (Definition of $\mathcal{L}$-constants)**.** The interpretations of the STY_1^3 constants from Table 1 obey the following semantic constraints: In (C8)–(C10), we let X abbreviate $\iota P.(\lambda k_1 \forall \boldsymbol{z}.\boldsymbol{P}(\boldsymbol{z})(k_1) = P([\iota z.\boldsymbol{z} = (\lambda k_2.E(z)(k_2))])(k_1))$:

$$\begin{array}{rlcl}
\text{(C1)} & \textcircled{\bot} & = & \lambda i.\bot; \qquad \text{(C2) } (\boldsymbol{B} \dot{\Rightarrow} \boldsymbol{C}) = \lambda i.\boldsymbol{B}(i) \Rightarrow \boldsymbol{C}(i); \\
\text{(C3)} & \textbf{\textit{partee}} & = & \lambda i.E(partee)(i); \qquad \text{(C4) } \textbf{\textit{woman}} = \lambda \boldsymbol{x}\, \lambda i.woman\big([\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i); \\
\text{(C5)} & \textbf{\textit{arrive}} & = & \lambda \boldsymbol{x}\, \lambda i.arrive\big([\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i); \\
\text{(C6)} & \textbf{\textit{believe}} & = & \lambda \boldsymbol{p}\, \lambda \boldsymbol{x}\, \lambda i.believe\big(\boldsymbol{p}, [\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i); \\
\text{(C7)} & \textbf{\textit{remember}} & = & \lambda \boldsymbol{y}\, \lambda \boldsymbol{x}\, \lambda i.remember\big([\iota y.\boldsymbol{y} = (\lambda j.E(y)(j))], [\iota x.\boldsymbol{x} = (\lambda k.E(x)(k))]\big)(i); \\
\text{(C8)} & \textbf{\textit{rapidly}} & = & \lambda \boldsymbol{P}\, \lambda \boldsymbol{x}\, \lambda i.rapidly\big(X, [\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i); \\
\text{(C9)} & \textbf{\textit{for}} & = & \lambda \boldsymbol{y}\, \lambda \boldsymbol{P}\, \lambda \boldsymbol{x}\, \lambda i.for\big([\iota y.\boldsymbol{y} = (\lambda j.E(y)(j))], X, [\iota x.\boldsymbol{x} = (\lambda k.E(x)(k))]\big)(i); \\
\text{(C10)} & \textbf{\textit{seek}} & = & \lambda \boldsymbol{Q}\, \lambda \boldsymbol{x}\, \lambda i.seek\big([\iota Q.(\forall \boldsymbol{P}.(\lambda k.\boldsymbol{Q}(\boldsymbol{P}, k)) = (\lambda k_3.Q(X, k_3)))] \\
 & & & \qquad [\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i)
\end{array}$$

The constraints (C1) and (C2) define the designated STY_1^3 constants $\textcircled{\bot}$ and $\dot{\Rightarrow}$ as the results of lifting the TY_2^3 connectives $\bot$ and $\Rightarrow$ to constructions out of the basic STY_1^3 type $(s;t)$.

In line with the type-$(s;t)$ representation of individuals from Section 3.2 (cf. (3.2)), the constraint (C3) defines the STY_1^3 constant ***partee*** as the designator of a function which sends situations to the truth-value of the proposition 'Barbara Partee exists' at those situations (i.e. as the designator of the characteristic function of the set of situations in which Barbara Partee exists).

The remaining constraints enable the definition of the STY_1^3 translations of sentential PTQ*-forms as (equivalents of) these forms' Montagovian translations. Thus, the constraints (C4) to (C10) contribute to the STY_1^3 representation of propositions along the lines of (3.1). In particular, the definition of the type-$((s;t);(s;t))$ term ***arrive*** as the designator of a function from propositions $\boldsymbol{x}$ to the set of situations at which the type-e correlate, $\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))$, of $\boldsymbol{x}$ arrives (cf. (C5)) enables the definition[12] of the STY_1^3 translation of the sentence Barbara Partee arrives (cf. (5.1)):

$$\begin{aligned} & [_{\text{S}}[_{\text{NP}}\text{Barbara Partee}][_{\text{VP}}[_{\text{IV}}\text{arrives}]]] \rightsquigarrow \boldsymbol{arrive}\,(\boldsymbol{partee}) \\ = \;& \lambda \boldsymbol{x}\, \lambda i.arrive\,\big([\iota x.\boldsymbol{x} = (\lambda j.E(x)(j))]\big)(i)\,[\lambda k.\,E(partee)(k)] \\ = \;& \lambda i.arrive\,\big([\iota x.[\lambda k.\,E(partee)(k)] = (\lambda j.E(x)(j))]\big)(i) \\ = \;& \lambda i.arrive\,\big([\iota x.partee = x]\big)(i) \quad = (\lambda i.\,arrive\,(partee)(i)) \end{aligned}$$

Since the definitions of the STY_1^3 translations of the PTQ*-forms from (5.2) to (5.6) are analogously obtained, we abstain from their statement. The definitions of the STY_1^3 terms from (5.7) to (5.10) are given below:

$$\boldsymbol{remember}\,(\boldsymbol{bill}, \boldsymbol{mary}) = \lambda i.remember\,(bill, mary)(i) \tag{6.1}$$

$$\begin{aligned} & \boldsymbol{remember}\,(\boldsymbol{for}\,(\boldsymbol{mary}, \boldsymbol{wait}, \boldsymbol{bill}), \boldsymbol{mary}) \\ = \;& \lambda i.remember\,\big(\underline{[\iota y.[\lambda k.for\,(mary, wait, bill)(k)] = (\lambda j.E(y)(j))]}, mary\big)(i) \end{aligned} \tag{6.2}$$

$$\begin{aligned} & \boldsymbol{remember}\,\big((\boldsymbol{bill} \wedge \boldsymbol{for}\,(\boldsymbol{mary}, \boldsymbol{wait}, \boldsymbol{bill})), \boldsymbol{mary}\big) \\ = \;& \boldsymbol{remember}\,(\boldsymbol{bill}, \boldsymbol{mary}) \wedge \boldsymbol{remember}\,(\boldsymbol{for}\,(\boldsymbol{mary}, \boldsymbol{wait}, \boldsymbol{bill}), \boldsymbol{mary}) \\ = \;& \lambda i.remember\,(bill, mary, i) \wedge \\ & \quad remember\,\big([\iota y.[\lambda k.for\,(pat, wait, bill, k)] = (\lambda j.E(y, j))], mary, i\big) \end{aligned} \tag{6.3}$$

$$\begin{aligned} & \bigvee \boldsymbol{x} \bigwedge \boldsymbol{y}.(\boldsymbol{problem}\,(\boldsymbol{y}) \dot{\leftrightarrow} \boldsymbol{x} \doteq \boldsymbol{y}) \wedge \boldsymbol{x} \doteq \boldsymbol{hate}\,(\boldsymbol{bill}, \boldsymbol{mary}) \\ = \;& \lambda i \exists x \forall y.(problem\,(y)(i) \leftrightarrow x = y) \wedge x = \underline{[\iota z.(\lambda k.hate(bill, mary)(k)) = (\lambda j.E(z)(j))]} \end{aligned} \tag{6.4}$$

The possibility of *defining* the STY_1^3 translations of [[TV][CP]]-structures in our single-type semantics is conditional on the existence of non-Montagovian individuals, which serve as type-e correlates of propositions: The TY_2^3 correlate, i.e. *remember*, of the STY_1^3 term ***remember*** restricts its first argument to TY_2^3 terms of the type e. To satisfy the typing constraints of the relevant TY_2^3 terms, we need to identify the individual which encodes the semantic information of the propositional argument. In the definitions of the STY_1^3 translations from (5.8) and (5.9), this is achieved by identifying the unique individual which exists exactly in the situations at which the formula $\lambda k.for\,(mary, wait, bill)(k)$ is true (cf. the underlined TY_2^3 term in (6.2), (6.3)). A similar observation holds for the definition (in (6.4)) of the STY_1^3 translation from (5.10).

Our presentation of the logic STY_1^3 has already established the possibility of evaluating the truth or falsity of basic-type terms in models of the metatheory TY_2^3 (cf. Def. 5). Since we know that the STY_1^3 translation of every logical PTQ*-form is defined through a term of the logic TY_2^3, we can evaluate the truth of logical PTQ*-forms via the truth of their translations' TY_2^3 definitions.

The identification of a STY_1^3 terms' referent in the designated model of the logic TY_2^3 enables the identification of equivalence relations between *proper names* and *sentences*. The semantic equiva-

[12]In the definition, the step from the third to the last line of step 3. is justified by our assumption of the unique reference of type-e constants and by the assumption that no two individuals exist in exactly the same situations (cf. Sect. 3.2). As a result, the interpretation of the term $\iota x.[\lambda k.E(partee)(k)] = (\lambda j.E(x)(j))$ will be defined in every model of the logic TY_2^3 which provides an interpretation for the constant *partee*.

lence of logical PTQ*-forms in our single-type semantics is defined below. In the definition, we let $\boldsymbol{A}_{(s;t)}$ and $\boldsymbol{B}_{(s;t)}$ be the STY_1^3 translations of the logical forms X, resp. Y, s.t. $X \rightsquigarrow \boldsymbol{A}$ and $Y \rightsquigarrow \boldsymbol{B}$. We let M^2 and $M_{\mathcal{F}}$ be the designated models of the logics TY_2^3, respectively STY_1^3, and let g^2 and $g = g^{2 \upharpoonright 1\mathsf{Type}}$ be their associated assignments.

Definition 9 (STY_1^3-based PTQ*-equivalence)**.** A logical form X is semantically *equivalent* to Y in M under g, i.e. $\textsc{means}_{M_{\mathcal{F}}}(Y, X)$, if $\models_g \boldsymbol{A} = \boldsymbol{B}$ in M^2 under g^2.

Definition 9 supports the equivalence of proper names with their simple containing simple existential sentences. For the name Barbara Partee, this is shown below:

$$\textsc{means}_{M_{\mathcal{F}}}\big([_{\mathrm{NP}}\text{Barbara Partee}],\ [_{\mathrm{S}}[_{\mathrm{NP}}\text{Barbara Partee}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{exists}]]]\big) \tag{6.5}$$
$$\text{iff} \quad \models_g \boldsymbol{partee} = \boldsymbol{E}\,(\boldsymbol{partee}) \ \text{ iff } \ \models_g (\lambda i.E\,(partee)(i)) = (\lambda i.E\,(partee)(i)) \ \text{ iff } \ \models_g \top$$

Significantly, because of our particular single-type choice, our STY_1^3-based single-type semantics fails to predict the attested equivalence relations between names and other contextually *salient* sentences besides existentials (e.g. the sentence Barbara Partee arrives) from Section 2. This is due to the fact that, for every individual and contextually salient (i.e. contingent) property, there will be situations in which the individual exists but does not have the property or at which it is undefined whether or not the individual has the property.

The satifaction of Proposition 2.ii requires the adoption of semantically 'richer' single-type objects, which provide different representations of Montagovian objects at different parameters. Functions from situations to propositions (type-$(s;(s;t))$) allow for this strategy: For example, these objects can represent individuals via functions from situations σ to the set of situations at which all true propositions at σ which carry information about the individual are true. We leave the development of this 'strong' single-type semantics for another occasion.

7. Conclusion

This paper has developed an $(s;t)$-based single-type semantics for the set of English logical forms from Montague (1970). The latter is a designated model for the logic STY_1^3, which interprets logical forms into constructions out of propositions. Objects of this type interpret proper names as (characteristic functions of) the set of situations in which the names' type-e referent exists, and interpret sentences and CPs as (characteristic functions of) the set of situations at which the sentence/CP is true. The semantics supports Partee's hypothesis from Proposition 1 (Partee, 2009), and accommodates the truth-evaluability of proper names from Proposition 2.i. However, the need to define STY_1^3 interpretations through the use of the 'lower' types e, s, and t suggests the need for a multi-typed metatheory, and the prominent role of Montague's (or Gallin's) original type system.

Future work will investigate 'stronger' single-type semantics (which further accommodate Prop. 2.ii), and the relationship of these semantics to Partee's original semantics. We hope that this research will give us further insight into the type system of natural language, and into the properties of minimal models in formal semantics.

References

Bayer, S. (1996). The coordination of unlike categories. *Language 72*(3), 579–616.

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

Fine, K. (1982). First-order modal theories iii. *Synthèse 53*, 43–122.

Gallin, D. (1975). *Intensional and Higher-Order Modal Logic with Applications to Montague Semantics*. North Holland.

Henkin, L. (1950). Completeness in the theory of types. *Journal of Symbolic Logic 15*, 81–91.

Klein, E. and I. A. Sag (1985). Type-driven translation. *Linguistics & Philosophy 8*, 163–201.

Kratzer, A. (1989). An investigation into the lumps of thought. *Linguistics & Philosophy 12*, 607–653.

Liefke, K. (forthcoming). *A Single-Type Semantics for Natural Language* (forthcoming PhD dissertation). Tilburg Center for Logic and Philosophy of Science.

Linsky, B. and E. N. Zalta (1994). In defense of the simplest quantified modal logic. *Philosophical Perspectives 8*, 431–458.

Merchant, J. (2008). Three types of ellipsis. In F. Recanati et al. (Ed.), *Context-Dependence, Perspective, and Relativity*, pp. 141–192. Berlin: Mouton de Gruyter.

Montague, R. (1970). English as a formal language. In R. Thomason (Ed.), *Formal Philosophy: Selected papers of Richard Montague*, pp. 188–221. New Haven: Yale University Press.

Montague, R. (1973). The proper treatment of quantification in ordinary english. In R. Thomason (Ed.), *Formal Philosophy*, pp. 247–270. New Haven: Yale University Press.

Muskens, R. (1995). *Meaning and Partiality*. CSLI Publications.

Partee, B. (1987). Noun phrase interpretation and type-shifting principles. *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, 115–143.

Partee, B. (2009). Do we need two basic types? *Snippets 20*, 37–41.

Pollard, C. (2008). Hyperintensions. *Journal of Logic and Computation 18*(2), 257–282.

Potts, C. (2002). The lexical semantics of parenthetical *-as* and appositive *-which*. *Syntax 5*(1), 55–88.

Stainton, R. J. (2006). *Words and Thoughts: Subsentences, ellipsis and the philosophy of language*. Oxford University Press.