Different usages of noun phrases appear to involve different semantic objects. In (1a) below the NP seems to refer to a concrete entity, in (1b) the NP looks like a unary predicate applying to an entity, while in (1c) the NP functions like a quantifier.

(1) a. Look at the trumpeter over there!
   b. Mary is a trumpeter.
   c. Every trumpeter likes Schoenberg.

In a notable step towards unifying the three usages of NPs, Montague’s classic article (Montague 1973, henceforth PTQ) avoided this traditional distinction, treating all NP occurrences as quantificational. Partee’s article Noun Phrase Interpretation and Type-Shifting Principles (Partee 1987, 2004) reconciles PTQ’s uniform quantificational strategy with the older distinction between three NP types. This three way distinction between NPs is encoded using the following semantic types:

- \( e \) entities
- \( (et)t \) one-place predicates, or functions from entities to truth-values
- \( (et)t \) quantifiers, or functions from one-place predicates to truth-values

On top of this distinction, Partee introduces operators that allow shifting the denotation of an NP to a different type than the one it is initially assigned. Using these type-shifters (TSs), one and the same NP may receive entity-referring, predicative and quantificational interpretations. In this way, Partee reintroduces the traditional distinction between NPs while retaining Montague’s quantificational treatment. In addition to this synthesis of previous approaches, Partee’s article contains a rather elaborate analysis of predicative NPs as in (1b), as well as insightful hints about the treatment of definite NPs, nominalization phenomena, plural, mass and generic NPs, and the general mathematical principles underlying type-shifting.

At a more general level, Partee’s article marks a methodological transition in formal semantics. Following the example of PTQ, most early works in the Montague tradition developed rigorous language fragments that explicitly treated certain linguistic facts, usually about English. This focus on small, well-defined fragments of English often obscured the wider linguistic relevance of semantic theory. In contrast, Partee’s article concentrates on general principles that are relevant to different languages and to different linguistic frameworks, and it is constructively removed from irrelevant technicalities of specific English fragments. This general account and the new ways it opened for semantic theory, together with the paper’s lucid and friendly style, have made Noun Phrase Interpretation and Type-Shifting Principles one of the modern classics in formal semantics.

This review is structured as follows. Section 1 gives necessary background on quantifier denotations and predicative NPs in PTQ. Section 2 reviews the main proposals in Partee’s article, and sections 3 and 4 comment on the work and its influence.

1 Quantificational and predicative NPs in PTQ

PTQ treats all NPs as quantifiers, i.e. functions from predicates to truth-values. This immediately accounts for the interpretation of NPs in subject positions. For example, in an extensional framework, the noun phrase a trumpeter denotes the following function:

\[
\lambda P_{et}. \exists x_e. \text{trumpeter}(x) \land P(x)
\]

In words, this is the function mapping a predicate \( P \) to true if and only if \( P \) holds of some trumpeter. For example, in the analysis (3b) of sentence (3a) below, the function (2) applies to the predicate smile. By simplifying the analysis, we get the intuitive result in (3c).

---

1 See chapter XXX-Zimmermann on PTQ, and chapter XXX-Keenan on Barwise & Cooper (1981), another classic article in semantic theory, which follows PTQ in analyzing all NPs as quantifiers.

2 All page references in the present article are to the reprinted version (Partee 2004).

3 Here and henceforth I use short notation for extensional types, where \( e \) and \( t \) are the types for entities and truth-values respectively, and \( (\tau, \sigma) \) is the type of functions from objects of type \( \tau \) to objects of type \( \sigma \). Suppressing outermost parentheses, we abbreviate this type as \( \tau \sigma \). Partee (1987) uses the \( (\tau, \sigma) \) notation for types of this form, hence, for instance, her notation \( ((e, t), t) \) for the quantifier type that I here denote \( (et)t \).

4 Like Partee, I ignore the intensional aspects of the analysis in PTQ.

5 Here and henceforth I refer by “simplification” to the \( \beta \)-conversion rule of the Lambda Calculus, which is used to simplify terms with function application. In general, a term \( \lambda x. \varphi (exp) \) describes a function \( x_\cdot \varphi \) applying to an expression \( exp \). The \( \beta \)-reduced term \( \varphi [x := exp] \), which is obtained by substituting \( exp \) for all free occurrences of \( x \) within \( \varphi \), denotes the same semantic object as the original term. For instance, the expression (3b) has the same denotation as (3c) since the latter is obtained by substituting the predicat smile for \( P \) in the term \( \exists x_e. \text{trumpeter}(x) \land P(x) \) within (3b).
(3) a. A trumpeter smiled.
   b. \((\lambda P_e.\exists x_e . \text{trumpeter}(x) \land P(x))(\text{smile}_e)\)
   c. \(\exists x_e . \text{trumpeter}(x) \land \text{smile}(x)\)

The quantificational analysis is applicable to all NPs in argument positions. Notably, even proper names, which intuitively denote simple entities, can be analyzed using quantificational operators of type \((et)t\). For example, the extensional version of PTQ uses the following quantifier as the denotation for the proper name Berg, relying on the entity berg:

(4) \(\lambda P_e. P(\text{berg})\)

In words, this is the \((et)t\) function that sends a predicate \(P\) to \(true\) if and only if \(P\) holds of the entity berg. In the sentence Berg smiled, this quantifier applies to the predicate smile, in formula:

(5) \((\lambda P_e. P(\text{berg}))(\text{smile})\)

After simplification, we get the intuitive analysis smile(berg).

Let us now consider NPs with a predicative use, as in Mary is a trumpeter \((=\text{(1b)})\). To analyze such sentences, PTQ treats the copula is as a function that takes quantifiers and turns them into predicates. Specifically, in sentence \((1b)\) the quantifier denotation \((2)\) of a trumpeter is mapped to a predicate. To see how this mapping works, let us consider that the function in \((2)\) can isomorphically be described as the set of all sets that contain a trumpeter.\(^6\)

Consider now the following situation:

(6) Let \(S\) be a situation with three entities: two trumpeters \(t_1\) and \(t_2\) and a non-trumpeter \(u\). In this situation, the sets of entities that contain a trumpeter are the following:

(7) \(\{t_1\} \quad \{t_1,u\} \quad \{t_2\} \quad \{t_2,u\} \quad \{t_1,t_2\} \quad \{t_1,t_2,u\}\)

With this description, we can present PTQ’s analysis of the predicate be a trumpeter using the following two steps:

**Step 1** Of the sets that the quantifier function describes, consider the singleton sets. In our example in \((7)\) these are the singletons \(\{t_1\}\) and \(\{t_2\}\).

**Step 2** Construct the predicate that holds of the entities in these singletons. In our example this is the predicate holding of \(t_1\) and \(t_2\) and nothing else.

Consequently, in the situation \(S\), sentence \((1b)\) is analyzed as claiming that Mary is either \(t_1\) or \(t_2\), i.e. one of the trumpeters in the situation. This is of course the desired result.

In more general terms, the extensional version of PTQ’s treatment of be is defined as the following function:

(8) \(BE = \lambda \psi(\text{et}), \lambda y_e . \psi(\lambda z_e . z = y)\)

In words, \(BE\) maps any quantifier \(\psi\) to the predicate holding of an entity \(y\) if and only if \(\psi\) holds of the function \(\lambda z_e . z = y\). The latter function is the predicate holding of \(y\) and nothing else, which in set notation can be described as the singleton \(\{y\}\). Accordingly, \(BE\) can be described as the function mapping every quantifier \(\psi\) to the predicate holding of \(y\) iff \(\psi\) holds of \(\{y\}\). Thus, for every entity \(y\), if the singleton \(\{y\}\) is one of the sets described by the quantifier \(\psi\) (as in \((7)\) above), we have \(y\) as one of the entities in the predicate assigned to \(\psi\) (as in Step 2).

There is an intuitive reason for Montague’s choice to use the \(BE\) function in his analysis of the English copula. Since PTQ only treats singular NPs, the \(BE\) operator only applies to denotations of NPs like Berg, a trumpeter, every trumpeter, etc. By only considering the singletons in such quantifiers, the \(BE\) function recovers the predicate denotation of the noun, at least in the cases where the copular sentence is clearly acceptable.\(^7\)

To see how this analysis works in lambda notation, let us consider what happens when we apply the \(BE\) operator to the quantifier in \((2)\):

\(^6\)The operator \((2)\) sends a function \(P\) of type \((et)\) to \(true\) if and only if the proposition \(\exists x_e . \text{trumpeter}(x) \land P(x)\) holds of \(P\). Of the entities in a given set \(E\), let \(S_P \subseteq E\) denote the set of entities \(\{x \in E \mid P(x)\}\) of which such a predicate \(P\) holds. Since \(S_P\) includes at least one trumpeter, we conclude that the set \(S_P \cap \{x \in E \mid \text{trumpeter}(x)\}\) is non-empty. Formally: the function \((2)\) is isomorphic to the set of sets \(\{S_P \subseteq E \mid S_P \cap \{x \in E \mid \text{trumpeter}(x)\} \neq \emptyset\}\). This presentation of a quantifier function as a set of sets follows from a general isomorphism between type-theoretical and set-theoretical perspectives on meaning. For more on this isomorphism and its use in formal semantics, see Winter (2016, ch.3-4).

\(^7\)On the vexing question of whether universal quantifier denotations of NPs like every trumpeter can ever appear as arguments of \(BE\) in predicate constructions, see section 3.
(9) \( BE(\lambda P_e. \exists x_e. \text{trumpeter}(x) \land P(x)) \)

This expression can be simplified as follows:
\[
= (\lambda P. \exists x. \text{trumpeter}(x) \land P(x))(\lambda z. z = y) \quad \text{by definition of } BE \\
= \lambda y. (\lambda P. \exists x. \text{trumpeter}(x) \land P(x))(\lambda z. z = y) \quad \text{simplification} \\
= \lambda y. \exists x. \text{trumpeter}(x) \land (\lambda z. z = y)(x) \quad \text{simplification} \\
= \lambda y. \exists x. \text{trumpeter}(x) \land x = y \quad \text{simplification} \\
= \lambda y. \text{trumpeter}(y) \quad \text{because for every predicate } P' \text{ and entity } y: \\
\exists x. P'(x) \land x = y \text{ is equivalent to } P'(y)
\]

This sequence of simplifications readily illustrates the complexity of applying the \( BE \) function to quantifiers. However, in the case of existential quantifiers like the denotation of the noun phrase \( \text{a trumpeter} \), it leads to a simple result: the expression \( \text{be a trumpeter} \) ends up denoting the same \( et \) predicate as the noun \( \text{trumpeter} \).

This happens because, in any situation, similarly to our example situation \( S \) above, the singletons \( \{y\} \) within the existential quantifier for \( \text{a trumpeter} \) are precisely the singletons \( \{y\} \) where \( y \) is a trumpeter.

2 Partee’s type-shifting strategy

Although Partee embraces the main foundational assumptions of PTQ, in one respect her proposal is substantially different from Montague’s. In PTQ, all NPs are treated as quantificational, and quantificational only. Thus, if predicates or entities come in handy for analyzing certain NP occurrences, this can only happen in PTQ by virtue of the meaning of the syntactic environment, and not due to the meaning of the NP itself. For instance, the PTQ analysis in (9) treats the expression \( \text{be a trumpeter} \) as denoting a predicate. This is due to the analysis of the English copula \( \text{be} \), but the “predicative” NP receives the same quantificational analysis of NPs in argument positions.

In contrast to PTQ, Partee lets NPs have entity and predicate denotations on top of their quantificational analysis. These different readings are derived using operators that Partee refers to as type-shifting principles—here, in short, type-shifters (TSs). For instance, in agreement with traditional accounts, Partee assumes that proper names initially denote entities. Thus, a sentence like Berg smiled immediately receives the intuitive analysis \( \text{smile(berg)} \), without PTQ’s intermediate step in (5). At the same time, following the proposal in (Partee & Rooth 1983), Partee allows all entity-denoting NPs to be shifted to quantifiers using the following operator:

(10) \( LIFT = \lambda x_e. \lambda P_e. P(x) \)

This function sends every entity \( x \) to the quantifier that holds of all predicates holding of \( x \). For instance, for the entity \( \text{berg} \) we have:

(11) \( LIFT(\text{berg}) = (\lambda x_e. \lambda P_e. P(x))(\text{berg}) = \lambda P_e. P(\text{berg}) \)

This is the same quantifier that the extensional PTQ analysis in (4) assumes as the lexical denotation of the proper name Berg. In this way, all proper names in Partee’s proposal are potentially ambiguous between entities and the quantifiers they denote in PTQ.

Type-shifting is also used in Partee’s analysis of predicative NPs. Following Williams (1983), Partee proposes that in \( \text{be} \) constructions like (1b), quantificational NPs are shifted to predicates. To do that, Partee still uses Montague’s function \( BE \), though not as the meaning of the English copula but rather as a TS that applies to the NP’s quantificational meaning and turns it into a predicate. Partee (2004, p.212) assumes, following Williams, that the copula \( \text{be} \) requires the expression following it to be of type \( et \). This triggers the application of \( BE \) in copular sentences like (1b). The resulting analysis is similar to the PTQ analysis (9), with \( BE \) as a type-shifter.

To summarize, we have seen two major elements in Partee’s analysis of NPs:

1. **Non-quantificational denotations**: in addition to quantifiers, NPs can also denote entities and predicates.

2. **Type-shifting**: an entity-denoting NP can be mapped to a quantifier using the type-shifter \( LIFT \); a quantifier-denoting NP can be mapped to a predicate using the type-shifter \( BE \).

\footnote{Formally: \( \lambda y. \text{trumpeter}(y) \) describes the same function as \( \text{trumpeter} \). In the Lambda Calculus this fact is generally captured using the \( \eta \)-conversion.}
This proposal modifies one of the central principles of Montague’s program: the matching between syntactic categories and semantic types. As a follower of the categorial tradition (Ajdukiewicz 1935, Bar-Hillel 1953, Lambek 1958), Montague assumed a close link between the syntactic category of an expression and its semantic role in the sentence. In the PTQ-style grammar architecture, we are always able to predict the semantic type of an expression from its syntactic category. Officially, we refer to this principle as follows:

(12) **Category-Type (C-T) Matching Principle:** If we assign a syntactic category $C$ to an expression $\alpha$, then the semantic type of $\alpha$ is predictable from $C$.

According to this principle, any occurrence of an expression of category $NP$ must have the same semantic type. This rigid assignment of types to categories gives compositional semantics a pleasingly restrictive architecture. Once we have decided on the workings of syntactic categories, we have little room to hesitate about the types of objects they denote. Specifically, according to the C-T Matching Principle, once we have decided to treat some NPs as quantificational, then all the expressions bearing the $NP$ category should receive a quantificational treatment.

Renouncing the C-T Matching Principle, Partee lets different NP occurrences have different types, and considers some empirical advantages of this approach over the PTQ analysis. First, as Partee (2004, p.207) mentions, proper names are among the NPs that license singular anaphoric expressions in discourse, as in (13a) below. By contrast, singular NPs like *every trumpeter* do not easily license such discourse anaphoric relations with singular pronouns (13b).

    b. Every composer came in. *He looked tired.

Following the treatments in (Kamp 1981, see ch. Geurts-XXX) and (Heim 1982), Partee assumes that only entity-denoting NPs contribute “discourse referents”. Thus, only entity-denoting NPs appear as antecedents of anaphors such as “he” in (13). In Partee’s proposal, proper names lexically denote entities, whereas NPs like *every composer* often fail to denote entities, even after type-shifting. Partee proposes that this accounts for contrasts as in (13). In addition, similarly to proper names as in (13a), definite and indefinite NPs like *the/a composer* also appear in discourse anaphora. Partee’s system also allows definites and indefinites to denote entities, which is used to account for this fact.

Another advantage of Partee’s proposal over PTQ concerns its treatment of predicative NPs. Montague’s work only treats predicative uses of NPs in copular sentences like (1b). However, Partee points out that other constructions also require predicative meanings of NPs. For instance, in sentence (14) below, the noun phrase *a trumpeter* functions as a predicate that holds of the entity denotation of *Mary*.

(14) Berg considers Mary a trumpeter.

In this kind of sentence, the verb *consider* combines with the small clause *Mary a trumpeter*. While PTQ does not analyze such sentences, Partee proposes to treat them by letting the verb *consider* select for a predicate of type *et* as one of its arguments. Partee’s *BE* operator shifts the quantificational denotation of *a trumpeter* in (14) to the required *et* type in the same way as in copular sentences like (1b). This leads to a $t$-type denotation of the small clause *Mary a trumpeter*, similarly to sentence (1b). Furthermore, Partee (2004, p.207) points out that indefinites like *a trumpeter* can appear in coordinations with adjectives and adjective phrases. This is illustrated by the following examples:

(15) a. John is competent in semantics and an authority on unicorns.
    b. Mary considers John competent in semantics and an authority on unicorns.

Coordinations as in (15) cannot be accounted for using the methods of PTQ. This is because adjectives are a prima facie case of predicates, and PTQ’s treatment of coordination requires the two conjuncts to be of the same type. Thus, cases like (15) are problematic for Montague’s uniform quantificational analysis of indefinite

---

8Partee’s *LOWER* type-shifter (see below) maps quantifiers to entities, but it is only defined for quantifiers that have single entities as their “basis”, i.e. quantifiers that can be described as the result of applying *LIFT* to some entity. This is not generally the case for NPs like *every composer*. Further, shifting *every composer* to a predicate is possible using *BE*, but that predicate cannot always be shifted further to an entity using Partee’s TSs, since the available strategy (the *IOTA* type-shifter, see below) requires the predicate to hold of only one entity.

9Partee does not fully analyze the status of discourses like (13b) in situations with only one composer. For some discussion of problems with universal NPs see section 3 below.

10Definites can be interpreted as entities using the *IOTA* type-shifter (see below). As for indefinites, (Partee 2004, p.216-7) tentatively proposes to replace PTQ’s treatment by free variables over entities as in (Kamp 1981) and (Heim 1982).

11For a review of small clauses and theoretical proposals for their syntactic treatment see Citko (2011).
NPs. Partee’s proposal easily deals with such coordinations using the predicative reading of indefinites, which is derived by the BE operator.

Importantly, these motivations for using entities and predicates as NP denotations do not undermine the quantificational treatment. As Partee mentions, Montague’s quantificational analysis of NPs is still required in conjunctions like King John and every peasant or a student I know and every teacher, where a proper name or an indefinite NP is conjoined with a universal NP. Since universal NPs can only be profitably analyzed using a quantificational type, the standard treatment of conjunction in PTQ requires that the other conjunct is also analyzed as a quantifier. With proper names like King John, this quantificational analysis is obtained using the LIFT operator from Partee & Rooth (1983). As for indefinites like a student I know, Partee considers two methods to derive their quantificational reading: either they initially denote quantifiers as in (2), or, following the proposals in (Kamp 1981) and (Heim 1982), they are treated as free variables ranging over entities (Partee 2004, p.216), which are shifted to quantifiers.

In addition to the LIFT and BE type-shifters, Partee introduces five other extensional operators between entities, predicates and quantifiers. Two of these TSs map predicates to quantifiers:

\[(16)\quad A = \lambda P_a.\lambda B_a.\exists x_e. P(x) \land B(x)\]

This operator maps every predicate \(P\) to the existential quantifier over this predicate, similarly to the PTQ analysis of a trumpeter in (2).

\[(17)\quad \text{THE} = \lambda P_e.\lambda B_e.\exists x_e. \forall y_e. (P(y) \leftrightarrow y=x) \land B(x)\]

This operator maps every predicate \(P\) to Montague’s quantificational analysis of definites. For instance, the sentence the trumpeter smiled is analyzed as follows:

\[
(\text{THE}(\text{trumpeter}))(\text{smile}) = ((\lambda P_e.\lambda B_e.\exists x_e. \forall y_e. (P(y) \leftrightarrow y=x) \land B(x))(\text{trumpeter}))(\text{smile}) \quad \text{by definition of THE}
\]

\[
= (\lambda B.\exists x. \forall y. (\text{trumpeter}(y) \leftrightarrow y=x) \land B(x))(\text{smile}) \quad \text{simplification}
\]

\[
= \exists x. \forall y. (\text{trumpeter}(y) \leftrightarrow y=x) \land \text{smile}(x) \quad \text{simplification}
\]

In words: there exists a unique trumpeter \(x\), and \(x\) smiled.

The \(A\) operator is the same function that generates the existential quantifier analysis of indefinites in (2). However, in addition to its use as the lexical meaning of the indefinite articles \(a\) and some in English, the \(A\) operator is also used as a TS. Similarly, Montague’s analysis of the English definite article is used as the THE type-shifter. Partee recognizes some mathematical properties of the type-shifting operators \(BE\), \(A\) and \(THE\), and claims that the “naturalness” of these properties helps to explain why in different languages, these functions “may be encoded either lexically or grammatically or not explicitly marked at all” (Partee 2004, p.204). This is an important part of her proposal, which deserves some elaboration.

Unlike English, in many languages there is no direct expression of (in)definiteness or predication similar to the English use of the words the, a and be. Consider for instance the following sentences in Polish and Hebrew:

\[(18)\quad \text{Anna je jablko}
\]

Anna eat apple

Polish: ‘Anna is eating the/an apple’ (Bielec 1998, p.270)

\[(19)\quad \text{dani ha-more le-matematika}
\]

Dani the-teacher to-Math

Hebrew: ‘Dani is the math teacher’ (Doron 1983, p.71)

Partee assumes that cross-linguistically, nouns and verbs are of the same predicate types as in English. Consequently, sentences as in (18) and (19) cannot be interpreted in Partee’s system without some additional “semantic glue”. For instance, the Polish noun jablko (‘apple’) in (18) must be shifted to an entity (using

\[13\] Other operators map predicates/quantifiers to entities (the IOTA/Lower operators, respectively, which are described below), and entities to predicates (the IDENT operator, which maps any entity \(x\) to the predicate describing the singleton \(\{x\}\)). Other TSs like NOM and LINK (see below) rely on intensional logic or further assumptions on the structure of semantic entities.

\[14\] This is the truth-conditional (aka. Russellian) analysis of definites. As mentioned below, Partee also uses a TS from predicates to entities, the IOTA operator, which models the presuppositional (aka. Strawsonian) analysis of definites. For a review of these two classic treatments in philosophy of language and formal semantics, see (Heim 2011).

\[15\] Additionally, many languages only optionally express concepts of (in)definiteness and predication, or express them using morphological processes rather than as single words. See (Dryer 2013a,b, Stassen 2013) for cross-linguistic overviews of these phenomena.
Indeed, some of Partee’s proposals have come under closer scrutiny. “the wealth of research these suggestions need to be tested against and integrated with” (Partee 2004, p.211). tee is of course aware of this unavoidable limitation of her contribution, and she duly makes caveats about that are not easy to test, or more concrete proposals that are quickly sketched without full development. Par-
anaphoric NPs and nominalized NPs. This broad ambition involves introducing general programmatic ideas singular and plural, definites and indefinites, genericity of NPs, quantificational NPs, presupposition of NPs, interpretation: the semantic distinction between argument NPs and predicate NPs, mass NPs and count NPs, reorganization of semantic theory, which has ramifications for some of the most central issues involving NP
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-
tic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and seman-

Partee’s article suggests that a principle of this sort can be extended to other areas in semantic theory, including the following:

1. **Nominalization operators:** Following Carlson (1977, see chapter XXX-Farkas) and Chierchia (1984), Partee proposes that bare plurals like dogs (as in dogs bark) and proper names like blue (as in blue is my favorite color) have entity denotations. These entities are derived by an operator NOM applying to the denotations of the common noun dogs and the adjective blue respectively. Importantly, NOM is an intensional TS: it must apply to a property denotation of the noun or adjective, rather than to a predicate of type et.

2. **Plurals:** Following Link (1983, see chapter XXX-Brasoveanu&Champollion), Partee assumes that plurals like the boys have entity denotations that represent “groups”. Partee proposes that such groups can be derived from quantifier denotations of plurals as in (Barwise & Cooper 1981) using a TS that she calls LINK. In this way, Partee aims to capture ambiguities as in the three boys ate a pizza, where the distributive interpretation (one pizza per boy) is derived by the quantificational denotation of the NP, and the collective interpretation (one pizza for all the boys together) is derived using its group reading.

3. **Definites:** In Partee’s account, all definite NPs like the king are ambiguous between the quantificational reading THE(king) obtained by the THE type-shifter, and the entity reading IOTA(king) obtained by the IOTA type-shifter. The quantificational reading is responsible for sentences where definites do not give rise to presuppositions: when there is no unique king, the definite quantifier THE(king) simply does not hold of any predicate. By contrast, the entity reading of definites is presuppositional: when there is no unique king, the denotation IOTA(king) is undefined. The IOTA operator is another TS that Partee considers as natural, hence potentially universal.

3 Critical comments

Partee’s paper modifies one of PTQ’s central principles: the matching between syntactic categories and semantic types (12). Developing the type-shifting strategy of (Partee & Rooth 1983), Partee proposes a comprehensive reorganization of semantic theory, which has ramifications for some of the most central issues involving NP interpretation: the semantic distinction between argument NPs and predicate NPs, mass NPs and count NPs, singular and plural, definites and indefinites, genericity of NPs, quantificational NPs, presupposition of NPs, anaphoric NPs and nominalized NPs. This broad ambition involves introducing general programmatic ideas that are not easy to test, or more concrete proposals that are quickly sketched without full development. Partee is of course aware of this unavoidable limitation of her contribution, and she duly makes caveats about “the wealth of research these suggestions need to be tested against and integrated with” (Partee 2004, p.211). Indeed, some of Partee’s proposals have come under closer scrutiny.

---

16This operator maps a predicate P to an entity c if c is the only entity of which P holds, otherwise it is undefined. Formally:

\[ IOTA = \lambda P_{et}.\begin{cases} c_e & \text{if } P = \lambda y. y = c \\ \text{undefined} & \text{otherwise} \end{cases} \]
One of the central ideas in Partee’s paper is that all NPs may in principle have predicative readings. The BE operator is defined for all quantifiers, and since Partee adopts Montague’s idea that all NPs have quantificational readings, we might expect all NPs to also have predicate denotations and appear in predicative positions. This expectation is not borne out. For instance, as illustrated by (21) below, universal NPs are often unacceptable in predicative positions:

\[(21) \ \text{?Berg is every composer.}\]

Partee’s system interprets (21) using the following predicative denotation of the noun phrase “every composer”:

\[
BE(\text{every(composer) = } \begin{cases} \text{composer} & \text{if } \text{composer} = (\lambda x.x = c) \text{ for some entity } c \\ \lambda x. \bot & \text{otherwise} \end{cases}
\]

The predicate in (22) corresponds to the singleton \{c\} if there is a unique composer c, otherwise it corresponds to the empty set. Consequently, Partee’s treatment expects sentence (21) to be true if Berg is the unique composer, and false otherwise. However, (21) is unacceptable, and does not seem to have any clear interpretation.

Addressing this problem, Partee (2004, pp.214-5) claims that with certain NPs, predicative readings are ruled out for pragmatic reasons. Partee argues that sentences like (21) are unacceptable because a sentence like “Berg is the (only) composer” would be a more natural alternative describing the same situations.

This pragmatic reasoning faces some problems. One problem is pointed out by Winter (2001, pp.143-5), considering cases like the following:

\[(23) \text{If Beethoven was the only pianist present at the premiere of his 4th Piano Concerto, then} \]

a. “every pianist at that premiere was among the performers.”

b. “the soloist was every pianist at that premiere.”

This example shows an asymmetry between argument positions (23a) and predicate positions (23b). Sentence (23a) is somewhat scholastic but acceptable, and avoids any implication that there was more than one pianist at the premiere. This is not the case for the predicative NP in (23b), which is as unacceptable as (21). If the unacceptability of (21) was only pragmatic, we should have expected this unacceptability to be pragmatically suspended in (23b) in the same way it is in (23a).

The syntactic-semantic restrictions on predicative NPs is another topic that is not accounted for in Partee’s paper. As shown in Doron’s (1983) study of Hebrew, there are important differences in distribution even among the NPs that easily appear in predicative positions. Specifically, unlike Hebrew definites as in (19), proper names cannot appear in predicative positions without a copula. Furthermore, Doron points out that English clauses that omit the copula (as in (15b) above), show similar distinctions to those in Hebrew. Consider for instance the following examples:

\[(24) \]

a. “I considered Mary a champion/the champion/*Sue.”

b. “I considered Mary to be a champion/the champion/Sue.”

The contrast in (24) shows a predicative environment where the be copula is required with proper names, but not with definite and indefinite NPs. Partee’s account does not distinguish proper names from other descriptions: denotations of all three kinds of NP can be mapped to predicates, e.g. using the BE operator. The unacceptability of “Sue” in (24a) shows that there are restrictions on the predicative readings of proper names that do not follow from Partee’s paradigm. More generally, Partee’s paper follows PTQ and analyzes

17I am ignoring here the use of universal NPs in case like “this house has been every color,” where the only valid interpretation is “every color x is such that the house has been (colored) x”. Partee (2004, p.219) refers to this kind of effect as the “Williams puzzle”, and treats it straightforwardly by letting the quantifier every color take sentential scope. The resulting predication over the entity-referring variable “(the color) x” is independent of the issues that I mention above, which involve Partee’s general mapping of quantifiers to predicates.

18Beethoven’s notable concert of 22 December 1808 featured the public premieres of his 4th Piano Concerto (with the composer as soloist), 5th and 6th Symphonies, and Choral Fantasy.

19Another potential problem for Partee’s shifting of quantifiers to predicates is observed in (Landman 2004, p.30), concerning examples with predicative plurals like “the people at the meeting were at most 30 deaf composers.” While such examples may perhaps not be fully acceptable, Landman shows that any theory that would like to treat them using TSs on quantifiers would face serious complications, since no function can derive sensible predicative meanings from standard quantificational denotations of NPs like at most 30 deaf composers.

20For instance, sentence (i) below is ungrammatical, and does not mean “Dani is Mr. Blum”:

(i) *dani adon blum
   Dani Mr. Blum
simple English sentences of the form \( NP_1 \) is \( NP_2 \) and small clauses as in (24) as involving two quantifiers (or entities) that are glued to each other using a logical operator (\( BE \) or \( IDENT \)). This approach avoids the traditional distinction between “predicative” usages of \( BE \) (\( Cicero \) \( is \) \( an \) \( orator \)) and “equative” usages of \( BE \) (\( Cicero \) \( is \) \( Tully \)). The observations by Doron, as well as by Geist (2007), Mikkelsen (2005) and others pose serious challenges for this uniform approach.

Another critical point about Partee’s proposal concerns the central role it gives to entity-denoting NPs. Partee motivates the use of entities as NP denotations by linguistic tradition, as well as by the special status of “referential NPs” (proper names, definites and simple indefinites) with discourse anaphora as in (13). However, while we may concede Partee’s claim that certain semantic operators on NPs need entities as their input, we should also note that under Partee’s other assumptions this is not conclusive motivation for introducing entities as NP denotations. Partee’s \( LOWER \) operator allows us to emulate any necessary operation on entities by adjusting it to PTQ-like quantifiers, leaving the result undefined when the quantifier cannot be associated with a unique entity.21 This shows a weakness of Partee’s argument for entity-denoting NPs. As opposed to her treatments of predicative and quantificational NPs, which are motivated by solid linguistic evidence, Partee’s proposal that “referential” NPs should denote entities is intuitive and fairly popular, but it is not formally necessary in her system.

4 Influence on further work

Partee’s paper has inspired much further research, and it would be impossible to summarize here all the important ideas in these contributions and the ways that they relate to Partee’s work. The non-exhaustive list below mentions some of the areas where Partee’s paper has been influential, with relevant references.


In addition to this direct influence of Partee’s article, her work was also one of the first general usages of the idea of type-shifting as introduced by Partee & Rooth (1983). As such, it helped to open the way for new usages of type-shifting in linguistics. For further references and a review of work on type-shifting from a psycholinguistic perspective, see Pyllkänen (2008). In relation to type-shifting in general, it should be mentioned that one of Partee’s TSs, the \( LIFT \) operator, adopted from Partee & Rooth (1983), comes out as a straightforward result in one of the most popular frameworks in type-logical categorial grammar: the Lambek Calculus (Lambek 1958, Moortgat 2011). Work on continuations in natural language (Barker 2001, de Groote 2001, Barker & Shan 2014) draws on this type-logical tradition as well as on the type-shifting analysis in Partee & Rooth (1983) and Partee (1987). Additionally, much work in Combinatory Categorical Grammar (Steedman 1997, Jacobson 2014) employs type-shifting principles.

Given this remarkable popularity of type-shifting as a tool in different grammatical frameworks, it is notable that there has not been much research on the theoretical foundations of Partee’s conception of type-shifting and its relation with categorial grammar and continuation-based semantics. Anticipating the linguistic importance

---

21The \( LOWER \) operator “retrieves” an entity from PTQ’s quantificational denotation of proper names and definites, as well as from Partee’s quantificational treatment of indefinites (Partee 2004, pp.216-7). Thus, if an operation \( op \) is defined on entities, we can instead use an operation \( OP \) on quantifiers by composing \( op \) with \( LOWER \). In formula, for any quantifier \( Q \):

\[
OP(Q) = op(LOWER(Q)) = \begin{cases} 
  op(c) & \text{if } Q = \lambda P. P(c) \\
  \text{undefined} & \text{otherwise}
\end{cases}
\]

22On Dynamic Semantics see chapters XXX-Geurts and XXX-Simons. Unifying dynamic approaches to meaning with Montague Semantics was not a central issue in Partee’s paper, but hers was one of the first published works to have proposed this direction, following an unpublished version of (Zeevat 1989).
of this topic, Partee (2004, p.224) claimed that this is one of the important areas for further research. I believe that, like many of Partee’s proposals in her seminal article, this claim still holds, more than 30 years after its first publication.
Acknowledgements: Work on this article was funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 742204). I’m grateful to Lucas Champollion, Edit Doron, Louise McNally, Eddy Ruys, Hanna de Vries, Joost Zwarts, and, especially, Barbara Partee, for their thoughtful remarks. All errors are mine.

References


