Type Shifting with Semantic Features: A Unified Perspective

Yoad Winter
Technion

1 Introduction

Since their introduction by Partee and Rooth (1983) into linguistic theory, type shifting principles have been extensively employed in various linguistic domains, including nominal predicates (Partee 1987), kind denoting NPs (Chierchia 1998), interrogatives (Groenendijk and Stokhof 1989), scrambled definites (De Hoop and Van der Does 1998) and plurals (Winter 2001,2002). Most of the accounts that use type shifting principles employ them as "last resort" mechanisms, which apply only when other compositional mechanisms fail. This failure is often sloppily referred to as *type mismatch*. The motivation for introducing type mismatch into the compositional mechanism is twofold: on the one hand it allows lexical items to be assigned the minimal types that are needed for describing their denotation; on the other hand, it has been argued that the "last resort" strategy of type shifting prevents derivation of undesired meanings.

The first goal of this paper is to define a simple notion of type mismatch, which will rather closely follow Partee and Rooth's original proposal but will be expressed within more familiar terms of categorial semantics. After introducing this implementation of traditional type mismatch, it will be argued that in fact, it covers only one possible kind of trigger for type shifting principles. Partee and Rooth's notion of mismatch is "external" in that the type of an expression is changed only when it combines with another type to which it cannot compose using the "normal" compositional mechanism. It will be argued that, within an appropriate type system, another notion of mismatch is also useful. This is the kind of mismatch in which the semantic type of an expression does not match its syntactic category. Two such cases will be explored: mismatch between morpho-syntactic number (singular or plural) and semantic number (a denotation ranging over atoms or sets), and mismatch between syntactic category (noun, DP, adjective etc.) and semantic role (predicate, quantifier, predicate modifier etc.). Both "external" and "internal" mismatch, which were proposed as triggers for type shifting in Winter (2001), will be formalized in one system where the definition of types matches more closely natural linguistic notions of semantic roles and semantic number.

2 Partee and Rooth's type fitting – twenty years later

The type shifting approach of Partee and Rooth (1983) assumes a traditional Montague Grammar, where expressions have syntactic categories (indicating their semantic types) and meaning composition is achieved using translation rules corresponding to syntactic rules. In later versions of Categorial Semantics (Van Benthem 1986,1991; Hendriks 1993), where a semantic *type calculus* is responsible for the composition of denotations and their types, translation rules are no longer necessary. Type shifting principles in Partee and Rooth's conception only applies when the grammar (or calculus) fails to compose types, or in Partee and Rooth's terms: when there is a *type mismatch*. To emphasize this "last resort" strategy for the application of type shifting principles, I will henceforth refer to this last resort resolution strategy as *type fitting*. Under this conception type mismatch is only a matter of *semantic* types, with no regard to the syntactic categories they originate from.

Partee and Rooth argue for type shifting as a last resort operation on the basis of the following examples.

(1) John caught and ate a fish.
$\not\Leftrightarrow$ John caught a fish and ate a fish.

(2) John hugged and kissed three women.
$\not\Leftrightarrow$ John hugged three women and kissed three women.

(3) John needed and bought a new coat.
$\Leftrightarrow$ John needed a new coat and bought a new coat.

(4) John wants and needs two secretaries.
$\Leftrightarrow$ John wants two secretaries and needs two secretaries.

In sentences (1) and (2), where two verbs that are extensional on their object argument are conjoined, the interpretation is not equivalent to "conjunction reduction" of the object argument. By contrast, when one of the verbs is intensional (as in (3)) or when both of them are (as in (4)), conjunction reduction leads to the correct paraphrase. Partee and Rooth propose to account for this phenomenon using the following three principles:

- Intensional verbs like *need, want* etc. have a "high" lexical type, as in Montague Grammar.
- Extensional verbs like *catch, eat, hug, kiss* and *buy* have the "high" type of intensional verbs, as well as a lower type $e(et)$.
- A general "last resort" strategy rules out a high type when a derivation using a lower type is available.

A conjunction of extensional verbs can be handled in type $e(et)$, and the resulting interpretation gives the object wide scope over the conjunction, as intuitively required in sentences (1) and (2). By contrast, when an intensional verb appears in a conjunction, the types of all the verbs in it must be of the higher type, and this leads to the "conjunction reduction" readings in (3) and (4). Hendriks, following Groenendijk and Stokhof (1989), implements Partee and Rooth's proposal without their "last resort" principle, and consequently type shifting in his system is free to occur whenever the resulting type sequence is derivable in the type calculus. Hendriks' system does not share with Partee and Rooth's the motivation to account for contrasts like the ones between (1)-(2) and (3)-(4). Ignoring the empirical debate – Hendriks explicitly argues against Partee and Rooth's judgements, and I will not try to resolve the empirical questions he raises in this paper – I will illustrate in this section how Partee and Rooth's "last resort" mechanism can be implemented using a simple fragment, which will be extended to deal with other phenomena in Section 4.

The simple fragment that will be introduced below consists of two calculi: a *syntactic calculus* $\mathcal{C}_{\mathbf{syn}}$ that manages expression *categories*; and a *semantic calculus* $\mathcal{C}_{\mathbf{sem}}$ that manages expression *types* and their respective denotations. Let $\mathsf{C}_1, \mathsf{C}_2, \ldots$ be the categories of a sequence of expressions, and let $\tau_1 : \varphi_1, \tau_2 : \varphi_2, \ldots$ be their respective types and denotations. Assume that a category C is derived in one derivation step of $\mathcal{C}_{\mathbf{syn}}$ from $\mathsf{C}_1, \mathsf{C}_2, \ldots$. Assume further that the type and denotation $\tau : \varphi$ are derived in $\mathcal{C}_{\mathbf{sem}}$ from $\tau_1 : \varphi_1, \tau_2 : \varphi_2, \ldots$. Then the following is a *derivation step* in the syntactic-semantic calculus $\mathcal{C}_{\mathbf{syn,sem}}$.

$$\frac{\mathsf{C}_1 : \tau_1 : \varphi_1 \quad \mathsf{C}_2 : \tau_2 : \varphi_2 \quad \ldots}{\mathsf{C} : \tau : \varphi}$$

This is a fairly standard view on the interactions between a syntactic and a semantic calculus (see e.g. Hendriks 1993). However, our definition of $\mathcal{C}_{\mathbf{syn,sem}}$ will also allow the application of type fitting principles when the syntactic calculus derives a category C from $\mathsf{C}_1, \mathsf{C}_2, \ldots$ but the semantic calculus fails to derive any type (and denotation) from the types $\tau_1, \tau_2, \ldots$ and the respective denotations. In such situations a sequence $\Pi = \pi_1, \pi_2, \ldots$ of type shifting operators is allowed to apply to the respective types and denotations in $\tau_1 : \varphi_1, \tau_2 : \varphi_2, \ldots$. If this derives the type and denotation $\tau : \varphi$ by $\mathcal{C}_{\mathbf{sem}}$, the following derivation step in $\mathcal{C}_{\mathbf{syn},\mathbf{sem}}$ is licensed:

$$\frac{\mathsf{C}_1 : \tau_1 : \varphi_1 \quad \mathsf{C}_2 : \tau_2 : \varphi_2 \quad ...}{\mathsf{C} : \tau : \varphi}\ \Pi$$

For simplicity, the fragment will employ only extensional types, using Definition 1 below. This definition is the standard definition of extensional types, with an additional type for coordinators. This special type is used here in order not to employ richer type systems with polymorphic coordination (see e.g. Emms 1991).

Definition 1 (extensional types) *Let* TYPE_1 *be the smallest set containing* e *(for* entities*) and* t *(for* truth values*), such that for every* $\tau, \sigma \in \mathrm{TYPE}_1$*:* $(\tau\sigma) \in \mathrm{TYPE}_1$*. The set of* extensional types *is* $\mathrm{TYPE}_1 \cup \{\mathsf{coor}\}$*, where* coor *is a special type for coordinators.*

Outermost parentheses of types are omitted. Each type in TYPE_1 classifies a domain according to the following standard definition.

Definition 2 (typed domains) D_e *is an arbitrary non-empty set.* $D_t = \{0, 1\}$*. If* τ *and* σ *are types then* $D_{\tau\sigma} = D_\sigma^{D_\tau}$*, the set of functions from* D_τ *to* D_σ*.*

The type coor for coordinators like *and* and *or* has no domain but is interpreted for *boolean* domains according to the following definitions of Partee and Rooth.

Definition 3 (boolean type) *An extensional type* τ *is* boolean *iff* $\tau = t$ *or* $\tau = \sigma_1\sigma_2$*, where* σ_2 *is a boolean type.*

Thus, boolean types are those types that are of the form $\sigma_1(\sigma_2(...(\sigma_n t)...))$, for some natural $n \geq 0$. For the corresponding domains conjunction is defined as follows, and similarly the other boolean operators.

Definition 4 (polymorphic conjunction) *Let* τ *be a boolean type. Let* $\wedge_{t(tt)}$ *be standard propositional conjunction. We denote:*

$$\mathbf{and}'_{\tau^c} \;=\; \begin{cases} \wedge_{t(tt)} & \text{if } \tau = t \\ \lambda X_\tau.\lambda Y_\tau.\lambda Z_{\sigma_1}.X(Z)\ \mathbf{and}'_{\sigma_2^c}\ Y(Z) & \text{if } \tau = \sigma_1\sigma_2 \end{cases}$$

The notation τ^c *is an abbreviation for type* $\tau(\tau\tau)$*, and the notation* $A\ \mathbf{and}'\ B$ *is of course a "sugaring" for* $(\mathbf{and}'(A))(B)$*.*

The set of categories is standardly defined as a closure of a set of *primitive* categories under the slash ('/') and backslash ('\') constructors. Formally:

Definition 5 (categories) *Let* CAT_0*, the set of* primitive categories*, be a finite non-empty set. The set* CAT_1 *is the smallest set containing* CAT_0 *that satisfies for every* X *and* $\mathsf{Y} \in \mathrm{CAT}_1$*:* $(\mathsf{X}/\mathsf{Y}) \in \mathrm{CAT}_1$ *and* $(\mathsf{X}\backslash\mathsf{Y}) \in \mathrm{CAT}_1$*. The set of* categories *is* $\mathrm{CAT}_1 \cup \{\&\}$*, where '&' is a special category for coordinators.*

In the grammar architecture that is used here, unlike traditional Montague Grammar, the semantic type of an expression is not predictable from its syntactic category. This allows type shifting not to affect the syntactic category and to introduce complex semantic operations (e.g. the composition of a transitive verb with its object) without complicating the syntactic categories that are assumed for expressions.

As a *syntactic calculus* for this toy grammar we need nothing more than the simple **AB** (Ajdukiewicz/Bar-Hillel) calculus, with its '/' and '\' elimination rules, augmented by a coordination rule. The resulting calculus, called '**AB**$^+$ calculus', is defined as follows.

Definition 6 (AB$^+$calculus) *For any categories* $\mathsf{X}, \mathsf{Y} \in \mathrm{CAT}_1$*:*

$$\frac{\mathsf{X}/\mathsf{Y} \quad \mathsf{Y}}{\mathsf{X}}\ /E \qquad \frac{\mathsf{Y} \quad \mathsf{Y}\backslash\mathsf{X}}{\mathsf{X}}\ \backslash E \qquad \frac{\mathsf{X} \quad \& \quad \mathsf{X}}{\mathsf{X}}\ C$$

	Category	Type	Denotation
John	NP	$(et)t$	$\mathbf{john}'$
a fish	NP	$(et)t$	$\mathbf{a\_fish}'$
catch	(NP\S)/NP	$e(et)$	$\mathbf{catch}'$
eat	(NP\S)/NP	$e(et)$	$\mathbf{eat}'$
need	(NP\S)/NP	$((et)t)(et)$	$\mathbf{need}'$
and	&	coor	$\mathbf{and}'$

Table 1: a toy P&R lexicon

For a *semantic calculus* in Partee and Rooth's framework we simply use the parallel of the $\mathbf{AB}^+$ calculus for undirected types with the appropriate semantics. We call this the '$\mathbf{A}^+$ calculus' (*Ajdukiewicz*$^+$ *calculus*):

Definition 7 ($\mathbf{A}^+$ calculus) *For any types $\tau, \sigma \in \mathrm{TYPE}_1$ and A, B, C denotations of the appropriate types:*

Function Application (APP): $$\frac{\tau\sigma : A \quad \tau : B}{\sigma : A(B)}$$

Permutation (PERM): $$\frac{\tau : B \quad \sigma : A}{\sigma : A \quad \tau : B}$$

Coordination (COOR): $$\frac{\tau : A \quad \mathsf{coor} : C \quad \tau : B}{\tau : (C_{\tau^c}(A))(B)}$$ (for any boolean type τ)

The toy lexicon we use for this grammar is given in Table 1. In this lexicon CAT_0, the set of primitive categories, is simply the set $\{\mathsf{NP}, \mathsf{S}\}$. The lexicon uses the following abbreviations, in addition to the non-logical constants $\mathbf{catch}'$, $\mathbf{eat}'$ and $\mathbf{need}'$.

$$\mathbf{john}' = \lambda P_{et}.P(\mathbf{j}')$$

$$\mathbf{a\_fish}' = \lambda P_{et}.\exists x[P(x) \wedge \mathbf{fish}'_{et}(x)]$$

The main insight of Partee and Rooth's proposal is in the application of "type correction" rules when "type mismatch" occurs. Formally, type mismatch is a situation where a string is syntactically well-formed according to the syntactic calculus, but semantically ill-formed according to the semantic type calculus. In such cases (only) type fitting operators are allowed to apply. In Partee and Rooth's original proposal, the only available type fitting rule is the following rule of *argument raising* (AR, cf. Hendriks 1993). The AR operator lifts the $e(et)$ type of transitive verbs to type $((et)t)(et)$, which composes with the quantifier type $(et)t$ as a first argument. Formally:

Argument Raising: $$\mathrm{AR}_{(e(et))(((et)t)(et))} \stackrel{def}{=} \lambda R_{e(et)}.\lambda Q_{(et)t}.\lambda y.Q(\lambda x.Q((R(x))(y)))$$

Since the AR operator is the only type fitting principle in the toy grammar we define, we use a singleton set Σ of type shifting principles: $\Sigma = \{\mathrm{AR}\}$.

The notions of type mismatch and its resolution are made explicit in the following definition of the syntactic-semantic calculus $\mathcal{C}_{\mathbf{syn},\mathbf{sem}}$.

Definition 8 ($\mathcal{C}_{\mathbf{syn},\mathbf{sem}}$ and type mismatch resolution) *Let $\mathcal{C}_{\mathbf{syn}}$ and $\mathcal{C}_{\mathbf{sem}}$ be syntactic and semantic calculi over given sets of categories and types. Let Σ, the set of* type shifting operations*, be a finite set of typed denotations. Let* exp_1, exp_2,... *be expressions of categories $\mathsf{C}_1, \mathsf{C}_2$, ..., types τ_1, τ_2, ... and denotations φ_1, φ_2, ..., respectively. Let Π be a sequence of operators op_1, op_2, ... where each operator op_i is of type $\tau_i\sigma_i$ s.t. either $op_i \in \Sigma$ or op_i is the identity function of type $\tau_i\tau_i$ ($\sigma_i = \tau_i$). A* derivation step *in $\mathcal{C}_{\mathbf{syn},\mathbf{sem}}$, the syntactic-semantic calculus over $\mathcal{C}_{\mathbf{syn}}$, $\mathcal{C}_{\mathbf{sem}}$ and Σ, ensues whenever the following* *hold:*

1. *Category* C *is derived from* $\mathsf{C}_1, \mathsf{C}_2, \ldots$ *in one derivation step of* $\mathcal{C}_{\mathbf{syn}}$.
2. *One of the following holds:*
   a. $\tau_1, \tau_2, \ldots \overset{\mathcal{C}_{\mathbf{sem}}}{\vdash} \tau$ *(type* τ *is derivable in* $\mathcal{C}_{\mathbf{sem}}$ *from* $\tau_1, \tau_2, \ldots$*); or*
   b. *There is no type* $\tau \in \mathrm{TYPE}$ *s.t.* $\tau_1, \tau_2, \ldots \overset{\mathcal{C}_{\mathbf{sem}}}{\vdash} \tau$ *(type mismatch),* *and* $\sigma_1 : op_1(\varphi_1), \sigma_2 : op_2(\varphi_2), \ldots \overset{\mathcal{C}_{\mathbf{sem}}}{\vdash} \tau : \varphi$ *(resolution).*

In the second case we say that $\mathsf{C} : \tau : \varphi$ *is* resolved *by* Π *from* $\mathsf{C}_1 : \tau_1 : \varphi_1, \mathsf{C}_2 : \tau_2 : \varphi_2, \ldots$. *The derivation step is denoted:*

$$\frac{\mathsf{C}_1 : \tau_1 : \varphi_1 \quad \mathsf{C}_2 : \tau_2 : \varphi_2 \quad \ldots}{\mathsf{C} : \tau : \varphi} \; (\Pi)$$

Appearance of the sequence Π *indicates type mismatch resolution using the operators in* Π *(case 2b above).*

In the $\mathbf{AB}^+$ and $\mathbf{A}^+$ calculi there are at most two items in a derivation step to which fitting operators can apply. Thus, instead of talking about resolution using a *sequence* of operators we talk about *left-* and/or *right-resolution* and use the following simpler notation.

$$\frac{\mathsf{C}_1 : \tau_1 : \varphi_1 \quad (\& : \mathsf{coor} : \varphi_c) \quad \mathsf{C}_2 : \tau_2 : \varphi_2}{\mathsf{C} : \tau : \varphi} \; op_1(l) \; op_2(r)$$

When this derivation step in $\mathcal{C}_{\mathbf{syn,sem}}$ involves a coordination, we of course have $\mathsf{C}_1 = \mathsf{C}_2 = \mathsf{C}$, $\tau_1 = \tau_2 = \tau$ (a boolean type) and $\varphi = (\varphi_c(\varphi_1))(\varphi_2)$. We omit the notation $op_1(l)$ (or $op_2(r)$) when op_1 (or op_2, respectively) is an identity function. This derivation characterizes the following scenario: a sequence of expressions forms an expression that is well-formed according to the syntactic calculus, but whose meaning is not derivable using the semantic calculus. This is referred to as a situation of *type mismatch*. A (pair of) type fitting operator(s) in Σ is said to (left/right) *resolve* the mismatch in case it can apply to the (leftward/rightward) type(s) in the given sequence and lead to a new sequence of types that can be composed using the semantic calculus. Note that in principle, we could have allowed recursive applications of type fitting principles until type resolution is achieved. However, for the sake of the analysis of the examples in this paper such a complication would not have any empirical advantages, and I am not aware of examples where it would. Moreover, the kind of type mismatch that is exemplified in Partee and Rooth's paper can always be resolved by applying AR to one of the types in the sequence (the predicate's type), and no other type shifting principle is available in the system. The situation will be different in Section 4.

In Figure 1 we see an illustration of the above system for Partee and Rooth's examples. In the case of *caught and ate a fish*, type mismatch appears only when the conjunction *caught and ate* composes with the object *a fish*. The AR operator is therefore allowed to apply (only) at this stage, which leads to the only attestable reading, entailing that there was a fish that was both caught and eaten. By contrast, in *needed and caught a fish*, type mismatch must apply already when the semantic calculus composes the denotations of *needed* and *caught*. The resolution of this type mismatch, by applying AR to the second conjunct, leads to the reading where a fish was needed, and some actual fish was caught.[1]

The general architecture of a grammar with type fitting *à la* Partee and Rooth creates situations where syntactically well-formed expressions are semantically "marked" since their meaning cannot be derived by the semantic calculus alone. There are two possible ways to view this mismatch between syntax and semantics. One way is to consider it a failure of the compositional system that *must* be resolved

[1]Typically, this is the kind of reading where a fish was needed *de dicto* (any fish, not a particular one). The other reading of the sentence, stating that there was a particular fish that was needed and caught, would be derived by any standard scope shifting principle (quantifying-in, quantifier raising etc.). Interestingly, the statement according to which one particular fish was needed (*de re*) and another one was caught, does not seem to be among the readings of the sentence, in agreement with what a system with scope shifting would expect.

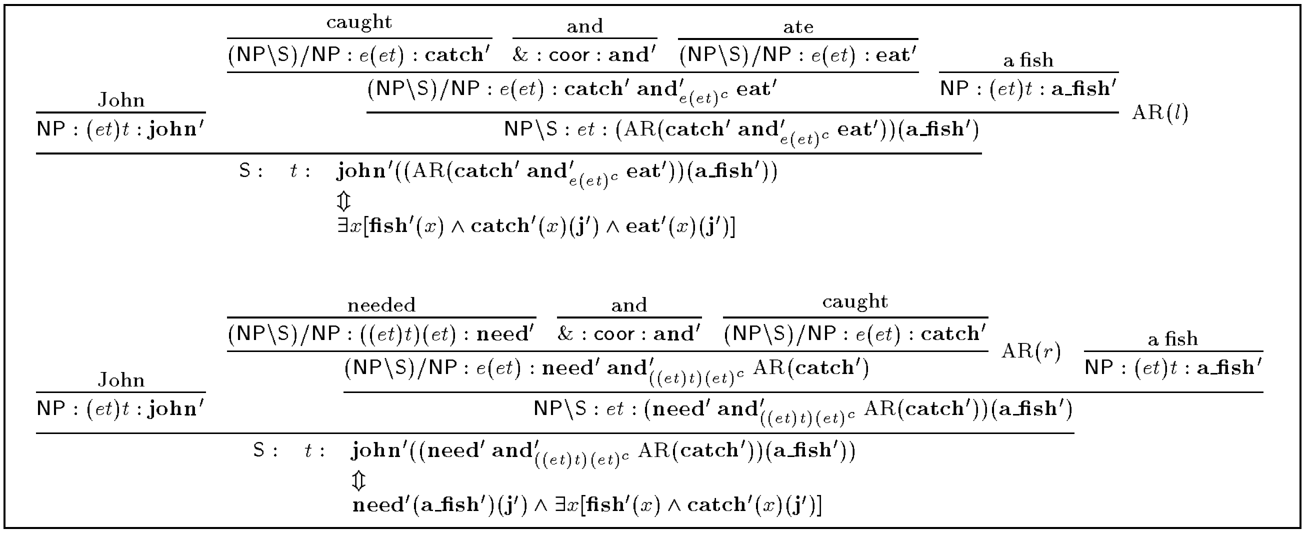

Figure 1: *caught and ate* vs. *needed and caught*

by some type shifting operator. Another way to look at the syntax-semantics mismatch is to consider it as a potential source of semantic anomaly. For instance, if a type system is developed in order to describe violations of selectional restrictions, type mismatch in syntactically well-formed expressions. In a standard type system like the present one, which deals only with denotational issues, I think the first option is more attractive since it puts a stronger restriction on the grammar as a whole (lexicon, syntax and semantics). Let me state this restriction as the following general hypothesis.

No type-anomalous expressions: *Whenever a given lexicon and given syntactic and semantic calculi lead to a type mismatch, there should be type shifting principles in* Σ *that resolve it.*

Cases that would go against this claim would be examples in natural language where the semantic type system would more naturally rule out a deviant expression than the syntactic system. Whether or not such cases exist seems to be an open problem.

3 Type fitting, category shifting and plurals

Should type fitting be the only strategy of meaning shift? In other words: are there situations that require to change meanings using covert operators, but in which Partee and Rooth's "last resort" strategy cannot apply? In Winter (2001) I argue that there are reasons to assume a "non-fitting" shifting strategy, which is however not based on shifting the type of expressions but on shifting their *semantic category* (e.g. from predicates to arguments or vice versa). In this section I will briefly review the reasons, mainly coming from the semantics of plurals, for assuming such a *category shifting* mechanism, which in the following section will be couched within a revision of Partee and Rooth's type fitting strategy.

In Bennett (1974), Scha (1981) and subsequent work, "plural individuals" are treated as elements of type et – standard one-place predicates, isomorphic to *sets* of atomic entities of type e. In Bennett's proposal, types no longer represent "semantic roles" in the sentence. For instance, type et is for both plural individuals (arguments of predicates) and predicates over singularities; type $(et)t$ is both for predicates over pluralities and for quantifiers over singularities, etc. This stands in opposition to the Link (1983) tradition, where "plural individuals" are of type e and the D_e domain is some algebraic structure (e.g. a lattice) over the set of atomic entities. Although the Link tradition is by far more popular in the literature than the Bennett/Scha tradition, both strategies have their advantages and disadvantages. The Bennett/Scha tradition complicates the types that are associated with natural language expressions, whereas the Link tradition complicates the ontology. In Winter (2001) I argue that some of the main operators for the semantics of plurals can be implemented as type fitting operators using Partee and Rooth's strategy. The three type fitting principles that are employed are defined below, in set theoretical notation:

Predicates	type:	$et \to (et)t$
	definition:	$pdist(A) \stackrel{def}{=} \wp(A) \setminus \{\emptyset\}$
	role:	a distributivity operator mapping predicates over singularities to "distributive" predicates over pluralities
Quantifiers	type	$(et)t \to ((et)t)t$
	definition:	$qfit(Q) \stackrel{def}{=} \{\mathcal{A} \subseteq \wp(E) : \{x \in E : \{x\} \in \mathcal{A}\} \in Q\}$
	role:	mapping quantifiers over singularities to "distributive" quantifiers over pluralities
Determiners	type:	$(et)((et)t) \to ((et)t)(((et)t)t)$
	definition:	$dfit(D) \stackrel{def}{=} \{\langle \mathcal{A}, \mathcal{B} \rangle \in \wp(\wp(E)) \times \wp(\wp(E)) : \langle \cup\mathcal{A}, \cup(\mathcal{A} \cap \mathcal{B}) \rangle \in D\}$
	role:	mapping determiners over singularities to "collective" determiners over pluralities

A typical example for the use of the *pdist* operator is when a collective predicate of type $(et)t$ and a "distributive" predicate of type et are conjoined, as in the following familiar kind of examples due to Dowty (1986) and Roberts (1987).

(5) The girls met in the bar and had a beer.

The *pdist* type fitting operator resolves the type mismatch in this case in a similar way to the type mismatch resolution by the AR operator in examples such as (3). Formally, this generates the following conjunction of type $(et)t$.

$$\mathbf{meet}'_{(et)t}\ \mathbf{and}'_{((et)t)^c}\ pdist(\mathbf{had\_beer}'_{et})$$

Another use that Winter (2001) makes in the above principles is in accounting for the difference between the determiners *every* and *all*. It is assumed that there is no denotational difference between these two determiners, which like all determiners are assumed to be of the "distributive" type $(et)((et)t)$. Both *every* and *all* denote the standard universal determiner, which is denoted '$\mathbf{every}'$'. However, the different plurality features of *every* and *all* affect the types of expressions in sentences they appear in. For instance, in sentence (6) below, the singular predicates for *student* and *met* (the set of students and the set of "meeting entities", respectively) are both of type et, and no type mismatch with the determiner *every* occurs. The result is the unacceptable reading of sentence (6) in (6a), claiming that each student is a "meeting entity".

(6) #Every student has met.

a. $\mathbf{every}'_{(et)((et)t)}(\mathbf{student\_sg}'_{et})(\mathbf{meet\_sg}'_{et})$

b. $*\ (dfit(\mathbf{every}'))(pdist(\mathbf{student\_sg}'))(pdist(\mathbf{meet\_sg}'))$

A statement using type shifting as in (6b) is ruled out since no type mismatch is present. By contrast, when plurals are involved, I assumed in Winter (2001) that the type of plural nominals (e.g. *students*) and verb phrases (e.g. *have met*) is $(et)t$. In sentence (7) below, the type mismatch that ensues with the determiner can only be resolved using the *dfit* operator, leading to the collective reading of (7) in (7b).

(7) All the students have met.

a. $\mathbf{every}'_{(et)((et)t)}\ \mathbf{students\_pl}'_{(et)t}\ \mathbf{meet\_pl}'_{(et)t}$ – type mismatch

b. $(dfit(\mathbf{every}'))(\mathbf{students\_pl}')(\mathbf{meet\_pl}')$

A similar contrast to the one between (6) and (7) exists between the following sentences.

(8) Every committee has met.

(9) All the committees have met.

While (8) is unambiguous and means that each of the committees had a separate meeting, sentence (9) is ambiguous between a ”collective” reading (a joint meeting of the committees) and a ”distributive” reading equivalent to (8). One complication that arises in the system of Winter (2001) is that in order to capture this ambiguity of (9), plural nominals like *students* or *committees* have to be treated as type ambiguous: the $(et)t$ type leads to the collective reading, whereas the et type leads to the distributive reading. The system proposed in the present paper avoids this type ambiguity.

In addition to type fitting principles *à la* Partee and Rooth, Winter (2001) proposes to use a different kind of covert semantic operators that are not triggered by type mismatch. These are called *category shifting* principles, and their triggers are purely syntactic: in the most straightforward implementation, category shifting principles are denotations of empty syntactic elements. These operators change *semantic features* of expressions but not necessarily their semantic type. Two kinds of semantic features are discussed in Winter (2001). One is the traditional distinction between predicates and quantifiers. Another semantic feature of denotations is whether they range over atoms (e type individuals) or over sets (et type ”plural individuals”). The two category shifting principles proposed in Winter (2001) change these two features:

1. A *Minimum operator* maps $(et)t$ quantifiers over atoms to $(et)t$ predicates over sets as follows:

   For each quantifier Q: $\text{MIN}(Q) \stackrel{def}{=} \{A \in Q : \forall B \subseteq A[B \in Q \rightarrow B = A]\}$ – the minimal sets in Q.

2. An *existential operator* – maps τt predicates over atoms ($\tau = e$) or over sets ($\tau = et$) to a $(\tau t)t$ quantifier:

   For each predicate P: $\text{E}(P) \stackrel{def}{=} \{A \subseteq D_\tau : A \cap P \neq \emptyset\}$ – the existential quantifier over P.

   (In fact, the proposed mechanism is a more complicated existential operation, involving *choice functions*, but this is immaterial for the present purposes.)

Let us consider some examples for the application of these two principles in Winter (2001). Sentence (10) is analyzed as in (10a), where the atomic quantifier denotation of the predicate nominal *John and Mary* is mapped to the predicate over sets $\{\{\mathbf{j'}, \mathbf{m'}\}\}$. The statement in (10a) claims that this predicate is among the sets in the quantifier over sets for *these people*.

(10) These people are John and Mary.

   a. $\mathbf{these\_people'}_{((et)t)t}(\text{MIN}((I_j \cap I_m)_{(et)t}))$

Sentence (11) is a simple example for the use of the E operator, which in (11a) maps the set of students to an existential quantifier.[2]

(11) A student arrived.

   a. $(\text{E}(\mathbf{student'}_{et}))_{(et)t}(\mathbf{arrived'}_{et})$

Combining the two category shifts is the basis for the account in Winter (2001) of the collective reading of NP coordination. Thus, for instance, sentence (12) is treated as in (12a), where the quantifier over atoms for *John and Mary* is first mapped to a predicate over sets as in (10a). The existential operator then maps this predicate over sets to a quantifier over sets, which captures the collective reading of *John and Mary*.

[2] In English, the same can be achieved by assigning the indefinite article a *denotation* of an existential determiner, but in languages that lack an indefinite article this is not a viable option.

(12) John and Mary met.

a. $(\mathrm{E}(\mathrm{MIN}((I_j \cap I_m)_{(et)t})))_{((et)t)t}(\mathbf{meet}'_{(et)t})$,
where $I_x = \{A \subseteq E : x \in A\}$

In this treatment, category shifting principles cannot be activated by the "last resort" principle of Partee and Rooth. To see why this is so, reconsider sentence (10). In this case the type of the predicate nominal does not change: it remains $(et)t$. Hence, type fitting is not necessary to begin with. Similarly, in (12), there is no need to apply the MIN operator, since the E operator itself could have resolved the type mismatch between the $(et)t$ subject and the $(et)t$ predicate (though this would lead to an undesired interpretation). In general, since MIN does not change the type of its argument, it is cannot be triggered by type mismatch.

A different example for the impossibility to use type mismatch as a trigger for category shifting in a compositional system comes from the following example, due to Hoeksema (1988).

(13) Dylan and Simon and Garfunkel wrote many hits in the 1960s.

a. $(\mathrm{E}(\mathrm{MIN}((I_d)_{(et)t})) \cap \mathrm{E}(\mathrm{MIN}((I_s \cap I_g)_{(et)t})))_{((et)t)t}(\mathbf{wrote\_many\_hits}'_{(et)t})$

To get the prominent reading (13a) of (13), where only *Simon and Garfunkel* are collectivized and not the whole subject,[3] category shifting has to apply *within* the subject, at a lower level to the one where type mismatch between the subject and predicate is detected. Applying category shifting at the level of the subject leads to an existing reading, though less prominent than (13a). According to this reading the three artists together wrote many hits in the 1960s.

A similar problem appears in the following example.

(14) A student and a teacher arrived.

a. $(\mathrm{E}(\mathbf{student}'_{et}) \cap \mathrm{E}(\mathbf{teacher}'_{et}))_{(et)t}(\mathbf{arrived}'_{et})$

For each of the indefinites in (14) to be existentially quantified separately, the E operator must apply twice within the subject, before the mismatch between the subject and predicate ensues. Applying E only at the level of the subject would lead to the existing but insufficient reading, according to which one person arrived, who is both a student and a teacher.

The conclusion in Winter (2001), as opposed to Partee (1987), is that although category shifting principles like E and MIN are useful for the derivation of meanings, they cannot be activated by type mismatch considerations *à la* Partee and Rooth (1983). However, as we shall see in the next section, revising the type system and the principles of "type mismatch" allows a unified mechanism that generalizes category shifting and type fitting.

4 Internal and external type fitting

This section introduces a unified perspective on type fitting and category shifting mechanisms. These two different modules that change different semantic resources – types as opposed to semantic categories – are replaced by one module for type change. To achieve that, the notion of type is redefined in such a way that represents both function-argument relations (as in standard types) and the semantic categories (predicate/quantifier, atom/set) of natural language expressions. Type change arises in this system as a result of two kinds of mismatch:

1. *External mismatch* – as in Partee and Rooth (1983) – a mismatch between typed denotations that cannot be composed by the type calculus.

[3] Of course, syntactic ambiguity in (13) leads to other readings as well, but due to obvious factors of world knowledge, they are not as prominent as the "collective Simon and Garfunkel" reading.

2. *Internal mismatch* – between the semantic type and the corresponding syntactic feature of one and the same expression.

The underlying intuition of this distinction is that external mismatch is a *failure* of the compositional mechanism, whereas internal mismatch is only an *unsteady state* of the syntax-semantics interface, which does not prevent interpretation but nevertheless sanctions application of type change operations. The correspondence between syntactic features and semantic types that this conception of internal mismatch employs is the following two kinds of correspondence:

Syntactic number (singular/plural) – Semantic number (atom/set);

Syntactic category (DP/NP) – Semantic role (quantifier/predicate).

As a result of this system, type change no longer involves empty syntactic categories as in Winter (2001), the notion of internal mismatch is justified by the natural correspondence assumed between the syntax and the semantics, and the lexical ambiguity of plural predicates in Winter (2001) is avoided.

The type system is first modified as follows. Instead of standardly having e and t as the primitive types of the system, we now have primitive types for quantifiers and predicates of all arities, with a semantic *number feature* (1 or 2) to denote semantic number. In addition to types, whose semantic number is unique in this definition, we add so-called *hyper-types*, which denote the type of operators that may change semantic number.

Definition 9 (Types and Hyper-types)
For any $n \in \{1, 2\}$, let TYPE_1 be the smallest set s.t.:

1. $\langle \mathsf{q}, n\rangle \in \mathrm{TYPE}_1$ (quantifiers)
2. $\langle \mathsf{p}m, n\rangle \in \mathrm{TYPE}_1$ for any natural number m (m-ary predicates)
3. If $\langle A, n\rangle$ and $\langle B, n\rangle$ are in TYPE_1 then $\langle (A \to B), n\rangle$ is in TYPE_1 (functions)

The set of *types* is $\mathrm{TYPE}_1 \cup \{\mathsf{coor}\}$, and the set of *hyper-types* is $\{(\tau \to \sigma) : \tau, \sigma \in \mathrm{TYPE}_1\}$.

Conventions: Types $\langle \mathsf{p}0, 1\rangle$ and $\langle \mathsf{p}0, 2\rangle$ (truth values) are both abbreviated 't'. In addition, for any type $\tau = \langle C, n\rangle$:

1. C is called the *semantic role* of τ: $C = \mathrm{SROL}(\tau)$;
2. n is called the *semantic number* of τ: $n = \mathrm{NUM}(\tau)$

The corresponding definition of set-theoretical domains (unlike the type-theoretical domains of Definition 2), employs two "basic domains": E_1 for atoms, and $\wp(E_1) \setminus \{\emptyset\}$ for non-empty sets of atoms. Officially:

Definition 10 (Domains) Let $E_1 \neq \emptyset$ be an arbitrary set, and $E_2 = \wp(E_1) \setminus \{\emptyset\}$.
The *domains of types* over $n \in \{1, 2\}$ are defined by:

1. $D_{\langle \mathsf{q}, n\rangle} = \wp(\wp(E_n))$.
2. $D_{\langle \mathsf{p}m, n\rangle} = \wp((E_n)^m)$, where $(E_n)^0 = \{\emptyset\}$.
3. $D_{\langle \tau \to \sigma, n\rangle} = D_{\langle \sigma, n\rangle}^{D_{\langle \tau, n\rangle}}$.

The *domains of hyper-types* are defined by:

$D_{\tau \to \sigma} = D_{\sigma}^{D_{\tau}}$.

Type	**Standardly**	**Denoting**
$\langle \mathsf{q}, 1\rangle$	$(et)t$	quantifiers over atoms
$\langle \mathsf{q}, 2\rangle$	$((et)t)t$	quantifiers over sets
$\langle \mathsf{p}1, 1\rangle$	et	one-place predicates over atoms
$\langle \mathsf{p}1, 2\rangle$	$(et)t$	one-place predicates over sets
$\langle \mathsf{p}2, 1\rangle$	$e(et)$	two-place predicates over atoms
$\langle \mathsf{p}2, 2\rangle$	$(et)((et)t)$	two-place predicates over sets
$\langle \mathsf{p}1 \to \mathsf{q}, 1\rangle$	$(et)((et)t)$	determiners over atoms
$\langle \mathsf{p}1 \to \mathsf{q}, 2\rangle$	$((et)t)(((et)t)t)$	determiners over sets
$\langle \mathsf{p}1 \to \mathsf{p}1, 1\rangle$	$(et)(et)$	modifiers of one-place predicates over atoms
$\langle \mathsf{p}1 \to \mathsf{p}1, 2\rangle$	$((et)t)((et)t)$	modifiers of one-place predicates over sets

Table 2: examples for some useful types

Type shifting operator	**Hyper-type**
AR	$\langle \mathsf{p}m, n\rangle \to \langle \mathsf{q} \to \mathsf{p}(m-1), n\rangle \quad (m \geq 1)$
pdist	$\langle \mathsf{p}1, 1\rangle \to \langle \mathsf{p}1, 2\rangle$
qfit	$\langle \mathsf{q}, 1\rangle \to \langle \mathsf{q}, 2\rangle$
dfit	$\langle \mathsf{p}1 \to \mathsf{q}, 1\rangle \to \langle \mathsf{p}1 \to \mathsf{q}, 2\rangle$
MIN	$\langle \mathsf{q}, 1\rangle \to \langle \mathsf{p}1, 2\rangle$
E	$\langle \mathsf{p}1, n\rangle \to \langle \mathsf{q}, n\rangle \quad (n = 1, 2)$

Table 3: type shifting operators and their hyper-types

Some examples for types under Definition 9 and for their standard parallels are given in Table 2. In this type system semantic number is a feature of a type. Hence, there are no types that mix semantic number. For instance: there is no type for functions like the *pdist* operator, from one place predicates over atoms to one place predicates over sets. Consequently, such functions that mix semantic number are described only using hyper-types. For instance: *pdist* is of hyper-type $\langle \mathsf{p}1, 1\rangle \to \langle \mathsf{p}1, 2\rangle$. This leads to the following hypothesis:

Hypothesis: *Types are sufficient for describing denotations of lexical entries. Hyper-types are needed only for type shifting operators.*

The set Σ of type shifting operators is given in Table 3. Note that the hyper-types for the AR and the E operators do not change semantic number and hence can also be represented using types.[4]

Note that all the primitive types in TYPE_1 have set theoretical domains, so conjunction can be defined for all types in TYPE_1 as follows (and similarly for the other boolean operators).

Definition 11 (polymorphic conjunction) *Let τ be a type in TYPE_1. We denote:*

$$\mathbf{and}'_{\tau^c} = \begin{cases} \cap & \text{if } \tau \text{ primitive (a quantifier or } m\text{-ary predicate)} \\ \lambda X_\tau . \lambda Y_\tau . \lambda Z_{\langle C_1, n\rangle} . X(Z) \ \mathbf{and}'_{\langle C_2, n\rangle^c} \ Y(Z) & \text{if } \tau = \langle C_1 \to \mathsf{C}_2, n\rangle \end{cases}$$

The notation $\langle C, n\rangle^c$ is an abbreviation for type $\langle C \to (C \to C), n\rangle$.

The Application rule of the $\mathbf{A}^+$ calculus in Definition 7 is redefined as follows, according to the new type system.

[4] Winter (2001) also argues for an E operator composed with distribution. This operator must be of a hyper-type $\langle \mathsf{p}1, 2\rangle \to \langle \mathsf{q}, 1\rangle$, which is not reducible to a type. For the sake of the discussion in this paper we will ignore this point.

$$\frac{\langle A \to B, n\rangle : f \quad \langle A, n\rangle : x}{\langle B, n\rangle : f(x)} \; n = 1, 2$$

For sake of completeness, we also add a rule that identifies the types $\langle \mathsf{p0}, 1\rangle$ and $\langle \mathsf{p0}, 2\rangle$ for truth values, which justifies the notation t for both of them.

$$\frac{\langle \mathsf{p0}, m\rangle : \varphi}{\langle \mathsf{p0}, n\rangle : \varphi} \; n, m = 1, 2$$

The set of categories is defined below using a slight revision of Definition 5, to allow number features on categories.

Definition 12 (categories) *Let* CAT_0*, the set of* primitive categories*, be a finite non-empty set. The set* CAT_1 *is the smallest set containing* $\mathrm{CAT}_0 \cup \{\mathsf{X}_n : \mathsf{X} \in \mathrm{CAT}_0, n \in \{1, 2\}\}$ *that satisfies for every* X *and* $\mathsf{Y} \in \mathrm{CAT}_1$*:* $\mathsf{X}/\mathsf{Y} \in \mathrm{CAT}_1$ *and* $\mathsf{Y}\backslash\mathsf{X} \in \mathrm{CAT}_1$*. The set of* categories *is* $\mathrm{CAT}_1 \cup \{\&\}$.

For the lexicon that will be introduced in the sequel we assume:

$$\mathrm{CAT}_0 \;=\; \{\mathsf{NP}, \mathsf{D'}, \mathsf{DP}, \mathsf{S}\}.$$

We use the following notation for semantic *features* of categories.

1. $\mathrm{NUM}(\mathsf{X}_n) = \mathrm{NUM}(\mathsf{X}_n/\mathsf{Y}_n) = \mathrm{NUM}(\mathsf{Y}_n\backslash\mathsf{X}_n) = n$
2. $\mathrm{SROL}(\mathsf{D'}_n) = \mathsf{q}$

Thus, for a category C and a feature FEAT, FEAT(C) is specified to be a value *val* if *val* is a preferred semantic feature of C, but not its only possible semantic feature. Hence, the preferred denotation of a singular (plural) expression is assumed to be based on atoms (sets), but this is not obligatory. The preferred denotation of $\mathsf{D'}$ is a quantifier, but it can also be a predicate. For DP (NP), the quantifier (predicate) denotation is obligatory, hence SROL(DP) and SROL(NP) are not defined.

As a syntactic calculus we still use the $\mathbf{AB}^+$ calculus of Definition 6, but in order to take care of number features within *and* conjunctions, the coordination rule is defined as follows when the conjoined categories include number features.

$$\frac{\mathsf{X}_l \quad \& \quad \mathsf{X}_m}{\mathsf{X}_k} \; C$$

where $l, m, k \in \{1, 2\}$ s.t. either $k = 2$ and ($\mathsf{X} = \mathsf{D'}$ or $\mathsf{X} = \mathsf{DP}$), or $l = m = k$

This condition on syntactic number reflects the general requirement of *and* conjunctions. With the exception of nominals, conjunction requires identity of the number feature on the conjuncts. See for instance, in English, the (un)acceptability of predicate conjunctions such as *smiles and dance/*dance* or *smile and dance/*dances*. However, with certain nominals, which here are classified as $\mathsf{D'}$, conjunctions are uniformly in the plural, independently of the number of the conjunct (cf. *John and Mary*, *some teacher and some author*). For more complex implications of this rule see example (15) below.

A toy lexicon that is used for illustration purposes is given in Table 4. The lexicon includes two empty elements (ϵ) that map a category N to $\mathsf{D'}$, and a $\mathsf{D'}$ to DP, without any change in type or denotation.

The definition of type mismatch and its resolution remain as in Definition 8. However, to model the triggers for the category shifting mechanisms of Winter (2001), we now also adopt the following definition of "internal" mismatch between type and category.

Definition 13 (internal mismatch resolution) *Let* exp *be an expression of category* C*, type* τ *and denotation* φ*. Let op be a type shifting operator of hyper-type* $\tau \to \tau'$*. Assume that the following hold for a feature* FEAT=NUM *(*FEAT=SROL*):*

1. $\mathrm{FEAT}(\mathsf{C}) \neq \mathrm{FEAT}(\tau)$

	Category	Type	Denotation
every	$\mathsf{DP}_1/\mathsf{NP}_1$	$\langle \mathsf{p1} \to \mathsf{q}, 1\rangle$	$\mathbf{every}' = \{\langle A, B\rangle \in \wp(E) \times \wp(E) : A \subseteq B\}$
all (the)	$\mathsf{DP}_2/\mathsf{NP}_2$	$\langle \mathsf{p1} \to \mathsf{q}, 1\rangle$	$\mathbf{every}'$
some	$\mathsf{D}'_n/\mathsf{NP}_n$	$\langle \mathsf{p1} \to \mathsf{q}, n\rangle,\ \ n = 1, 2$	$\mathrm{E} = \{\langle A, B\rangle \in \wp(E) \times \wp(E) : A \cap B \neq \emptyset\}$
a	$\mathsf{NP}_1/\mathsf{NP}_1$	$\langle \mathsf{p1} \to \mathsf{p1}, 1\rangle$	*id*
student	NP_1	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{student}'$
students	NP_2	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{student}'$
teacher	NP_1	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{teacher}'$
teachers	NP_2	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{teacher}'$
committee	NP_1	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{committee}'$
committees	NP_2	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{committee}'$
John	D'_1	$\langle \mathsf{q}, 1\rangle$	$I_{\mathbf{j}'} = \{A \subseteq E : \mathbf{j}' \in A\}$
Mary	D'_1	$\langle \mathsf{q}, 1\rangle$	$I_{\mathbf{m}'} = \{A \subseteq E : \mathbf{m}' \in A\}$
smiles	$\mathsf{DP}_1\backslash\mathsf{S}$	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{smile}'$
smile	$\mathsf{DP}_2\backslash\mathsf{S}$	$\langle \mathsf{p1}, 1\rangle$	$\mathbf{smile}'$
meets	$\mathsf{DP}_1\backslash\mathsf{S}$	$\langle \mathsf{p1}, 2\rangle$	$\mathbf{meet}'$
meet	$\mathsf{DP}_2\backslash\mathsf{S}$	$\langle \mathsf{p1}, 2\rangle$	$\mathbf{meet}'$
and	&	coor	$\mathbf{and}'$
ϵ	$\mathsf{D}'_n/\mathsf{NP}_n,\ \ n = 1, 2$	$\langle \mathsf{p1} \to \mathsf{p1}, m\rangle,\ \ m = 1, 2$	*id*
ϵ	$\mathsf{DP}_n/\mathsf{D}'_n,\ \ n = 1, 2$	$\langle \mathsf{q} \to \mathsf{q}, m\rangle,\ \ m = 1, 2$	*id*

Table 4: another toy lexicon

2. $\mathrm{FEAT}(\mathsf{C}) = \mathrm{FEAT}(\tau')$

Then we say that $\mathsf{C} : \tau' : op(\varphi)$ *is derived by op from* $\mathsf{C} : \tau : \varphi$ *due to N-mismatch (S-mismatch), and denote:*

$$\frac{\mathsf{C} : \tau : \varphi}{\mathsf{C} : \tau' : op(\varphi)}\ op(N/S)$$

This definition captures the situation where an expression has a category and a type with features that do not match. This can be a singular (plural) expression whose denotation ranges over atoms (sets, respectively), or an expression of a quantificational (predicative) category whose type is predicative (quantificational, respectively). When such a mismatch occurs and one of the type shifting principles in Σ can resolve it, then it is allowed to apply.

Consider for example Figure 2. In the first derivation, of the sentence *every committee meets*, the only internal mismatch is within the verb *meet*, between its singular number and set-based denotation. However, this internal mismatch cannot be resolved in the present system since no type shifting operator in Σ maps a denotation over sets to a denotation over atoms. The only mismatch that is resolved is an external mismatch, between the type of the predicate *meets* and the type of the subject. Two type shifting operators are needed to resolve this mismatch: the *qfit* operator changes the semantic number of the subject from atom to set; the AR operator allows the predicate to combine with a quantifier. The result is the derivation of the only reading of the sentence, talking about separate committee meetings. The situation is different in the sentence *all the committees meet*. In this case there are internal mismatches in both the determiner and the noun, which are resolved by the *dfit* and *pdist* operators, respectively. This leads to the collective reading. Note however that these resolutions are not obligatory in the derivation, and an additional derivation where they are not performed leads to another meaning, parallel to the ”separate meeting” reading of the sentence *every committee meets*. This eliminates the need for lexical ambiguity of plurals in Winter (2001).

Consider now the derivation in Figure 3 of the sentence *John and Mary met*. Due to the *and* rule, the conjunction *John and Mary* is of syntactic number 2 even though its denotation ranges over atoms. This creates an internal mismatch, which the MIN operator resolves. However, the MIN operator itself creates an internal mismatch between the semantic role of the subject category (quantifier) and its semantic type

$$\frac{\text{every}}{\mathsf{DP}_1/\mathsf{NP}_1 : \langle \mathsf{p1}\rightarrow \mathsf{q}, 1\rangle : \mathbf{every}'} \quad \frac{\text{committee}}{\mathsf{NP}_1 : \langle \mathsf{p1}, 1\rangle : \mathbf{committee}'}$$

$$\mathsf{DP}_1 : \langle \mathsf{q}, 1\rangle : \mathbf{every}'(\mathbf{committee}') \quad \frac{\text{meets}}{\mathsf{DP}_1\backslash\mathsf{S} : \langle \mathsf{p1}, 2\rangle : \mathbf{meet}'}$$

$$\mathsf{S} : \; t : \; (\mathit{qfit}(\mathbf{every}'(\mathbf{committee}')))(\mathbf{meet}') \quad \mathit{qfit}(l)\ \mathrm{AR}(r)$$

$$\Updownarrow$$

$$\forall x \in \mathbf{committee}'[\mathbf{meet}'(\{x\})]$$

$$\frac{\dfrac{\text{all the}}{\mathsf{DP}_2/\mathsf{NP}_2 : \langle \mathsf{p1}\rightarrow \mathsf{q}, 1\rangle : \mathbf{every}'}}{\mathsf{DP}_2/\mathsf{NP}_2 : \langle \mathsf{p1}\rightarrow \mathsf{q}, 2\rangle : \mathit{dfit}(\mathbf{every}')}\ \mathit{dfit}(N) \quad \frac{\dfrac{\text{committees}}{\mathsf{NP}_2 : \langle \mathsf{p1}, 1\rangle : \mathbf{committee}'}}{\mathsf{NP}_2 : \langle \mathsf{p1}, 2\rangle : \mathit{pdist}(\mathbf{committee}')}\ \mathit{pdist}(N)$$

$$\mathsf{DP}_2 : \langle \mathsf{q}, 2\rangle : (\mathit{dfit}(\mathbf{every}'))(\mathit{pdist}(\mathbf{committee}')) \quad \frac{\text{meet}}{\mathsf{DP}_2\backslash\mathsf{S} : \langle \mathsf{p1}, 2\rangle : \mathbf{meet}'}$$

$$\mathsf{S} : t : ((\mathit{dfit}(\mathbf{every}'))(\mathit{pdist}(\mathbf{committee}')))(\mathbf{meet}') \quad \mathrm{AR}(r)$$

Figure 2: *every* vs. *all*

$$\frac{\text{John}}{\mathsf{D}'_1 : \langle \mathsf{q}, 1\rangle : I_{\mathbf{j}'}} \quad \frac{\text{and}}{\& : \mathsf{coor} : \mathbf{and}'} \quad \frac{\text{Mary}}{\mathsf{D}'_1 : \langle \mathsf{q}, 1\rangle : I_{\mathbf{m}'}}$$

$$\mathsf{D}'_2 : \langle \mathsf{q}, 1\rangle : I_{\mathbf{j}'} \cap I_{\mathbf{m}'}$$

$$\mathsf{D}'_2 : \langle \mathsf{p1}, 2\rangle : \mathrm{MIN}(I_{\mathbf{j}'} \cap I_{\mathbf{m}'}) \quad \mathrm{MIN}(N)$$

$$\frac{\epsilon}{\mathsf{DP}_2/\mathsf{D}'_2 : \langle \mathsf{q}\rightarrow \mathsf{q}, 2\rangle : \mathit{id}} \quad \mathsf{D}'_2 : \langle \mathsf{q}, 2\rangle : \mathrm{E}(\mathrm{MIN}(I_{\mathbf{j}'} \cap I_{\mathbf{m}'})) \quad \mathrm{E}(S)$$

$$\mathsf{DP}_2 : \langle \mathsf{q}, 2\rangle : \mathrm{E}(\mathrm{MIN}(I_{\mathbf{j}'} \cap I_{\mathbf{m}'})) \quad \frac{\text{meet}}{\mathsf{DP}_2\backslash\mathsf{S} : \langle \mathsf{p1}, 2\rangle : \mathbf{meet}'}$$

$$\mathsf{S} : \; t : \; (\mathrm{E}(\mathrm{MIN}(I_{\mathbf{j}'} \cap I_{\mathbf{m}'})))(\mathbf{meet}') \quad \mathrm{AR}(r)$$

$$\Updownarrow$$

$$\mathbf{meet}'(\{\mathbf{j}', \mathbf{m}'\})$$

Figure 3: *John and Mary met*

(predicate). This is resolved by the E operator.[5]

Finally, let us consider the following sentences, and the present restatement of Winter's (2001) account of the effects they exemplify.

(15) a. A great author and a famous mathematician has passed away. (Hoeksema 1988)

b. A great author and a famous mathematician have passed away.

Sentence (15a) entails that one person has passed away who was both an author and a mathematician. By contrast, sentence (15b) does not have this entailment and moreover implies that two different people have passed away. In Figure 4 the phenomena that these two cases exemplify are analyzed in the given fragment. The singular reading of the conjunction *a student and a teacher* must be derived as an NP conjunction since the conjunction rule derives singularity of conjunctions within the DP only when singular NPs are conjoined. Therefore, the conjunction must denote an intersection of the sets for *student* and *teacher*, as intuitively required. By contrast, when the conjunction bears plural agreement it cannot be analyzed as an NP conjunction and must be analyzed as a D'(or DP) conjunction, in which case internal mismatch occurs between the derived $\mathsf{p}1$ semantic role and the q semantic role of the D' category. This allows application of the existential operator E and leads to the desired intersection of two existential quantifiers.

Consider now the following sentences, in contrast to the examples in (15).

(16) a. *Some great author and some famous mathematician has passed away.

b. Some great author and some famous mathematician have passed away.

[5] An additional analysis of the sentence is when the *qfit* operator applies to the subject without resolving the internal mismatch in it. This analysis however leads to the implausible statement "Mary met and John met".

Since *some* is analyzed as generating a D', these conjunctions must be plural, hence (16a) is syntactically ruled out and (16b) is analyzed parallel to the analysis of (15b) in Figure 4.

5 Conclusions

This paper started by reviewing Partee and Rooth's (1983) conception of type shifting as a strategy for type mismatch resolution. A formalization of Partee and Rooth's principle was given within categorial semantics. Winter's (2001) additional mechanism of category shifting was reviewed and was shown to be irreducible to Partee and Rooth's type fitting strategy with standard types. A non-standard typing system was developed, which describes not only function-argument relations but also other semantically interesting features like "semantic number" (atom/set) and "semantic role" (predicate/quantifier). It was shown that with this richer typing system it is possible to unify the principles of type fitting and category shifting. In the proposed system, type shifting is the only flexibility operation, and the only trigger for type shifting is mismatch. However, mismatch can be either between types of two expressions in a construction (Partee and Rooth's external type mismatch) or between a type and a category of the same expression (internal mismatch). It is important to note that both notions are in a sense independently motivated. External mismatch arises when using categorial semantic systems which are weaker than what the syntax requires for meaning composition. For instance, Partee and Rooth use the AR operator for composing the a binary relation with a quantifier, which the syntax requires but simple categorial semantics cannot derive. On the other hand, internal mismatch can only occur when there are natural correspondences between syntactic and semantic features. Thus, it is natural to assume that morpho-syntactic singularity (plurality) corresponds to quantification over atoms (sets), and that different layers within the DP correspond to different semantic roles (predicate, argument, quantifier, etc.). Unlike previous works, the present paper suggests an optional, rather than obligatory, correspondence between such syntactic and semantic features, which leads to "unsteady states" in the syntax-semantics interface. Whether there are more features that allow this kind of optional correspondence between syntax and semantics is currently under investigation.

Acknowledgements

Thanks to the participants of the workshop on Direct Compositionality, and especially to Pauline Jacobson and Chung-chieh Shan. I am grateful to Nissim Francez for his thorough remarks on an earlier draft.

References

Bennett, M. (1974). *Some Extensions of a Montague Fragment of English*. PhD thesis, University of California Los Angeles.

Chierchia, G. (1998). Reference to kinds across languages. *Natural Language Semantics*, 6:339–405.

De Hoop, H. and Van der Does, J. (1998). Type shifting and scrambled definites. *Journal of Semantics*, 15:393–416.

Dowty, D. (1986). Collective predicates, distributive predicates and *all*. In *Proceedings of the Eastern States Conference on Linguistics, ESCOL3*. Cascadilla Press.

Emms, M. (1991). Polymorphic quantifiers. In Barry, G. and Morrill, G., editors, *Studies in Categorial Grammar*. Centre for Cognitive Science, Edinburgh.

Groenendijk, J. and Stokhof, M. (1989). Type shifting rules and the semantics of interrogatives. In Chierchia, G., Partee, B., and Turner, R., editors, *Properties, Types and Meanings*, volume 2. Kluwer, Dordrecht.

Hendriks, H. (1993). *Studied Flexibility: categories and types in syntax and semantics*. PhD thesis, University of Amsterdam.

Hoeksema, J. (1988). The semantics of non-boolean *and*. *Journal of Semantics*, 6:19–40.

Link, G. (1983). The logical analysis of plurals and mass terms: a lattice theoretical approach. In Bauerle, R., Schwarze, C., and von Stechow, A., editors, *Meaning, Use and Interpretation of Language*. De Gruyter, Berlin.

Partee, B. (1987). Noun phrase interpretation and type shifting principles. In Groenendijk, J., de Jong, D., and Stokhof, M., editors, *Studies in Discourse Representation Theories and the Theory of Generalized Quantifiers*. Foris, Dordrecht.

Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In Bauerle, R., Schwarze, C., and von Stechow, A., editors, *Meaning, Use and Interpretation of Language*. De Gruyter, Berlin.

Roberts, C. (1987). *Modal Subordination, Anaphora, and Distributivity*. PhD thesis, University of Massachusetts at Amherst.

Scha, R. (1981). Distributive, collective and cumulative quantification. In Groenendijk, J., Stokhof, M., and Janssen, T. M. V., editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, Amsterdam.

Van Benthem, J. (1986). *Essays in Logical Semantics*. D. Reidel, Dordrecht.

Van Benthem, J. (1991). *Language in Action: categories, lambdas and dynamic logic*. North-Holland, Amsterdam.

Winter, Y. (2001). *Flexibility Principles in Boolean Semantics: coordination, plurality and scope in natural language*. MIT Press, Cambridge, Massachusetts.

Winter, Y. (2002). Atoms and sets: a characterization of semantic number. *Linguistic Inquiry*, 33:493–505.

Computer Science
Technion - Israel Institute of Technology
Haifa 32000
Israel
E-mail: `winter@cs.technion.ac.il`
WWW: `http://www.cs.technion.ac.il/~winter`

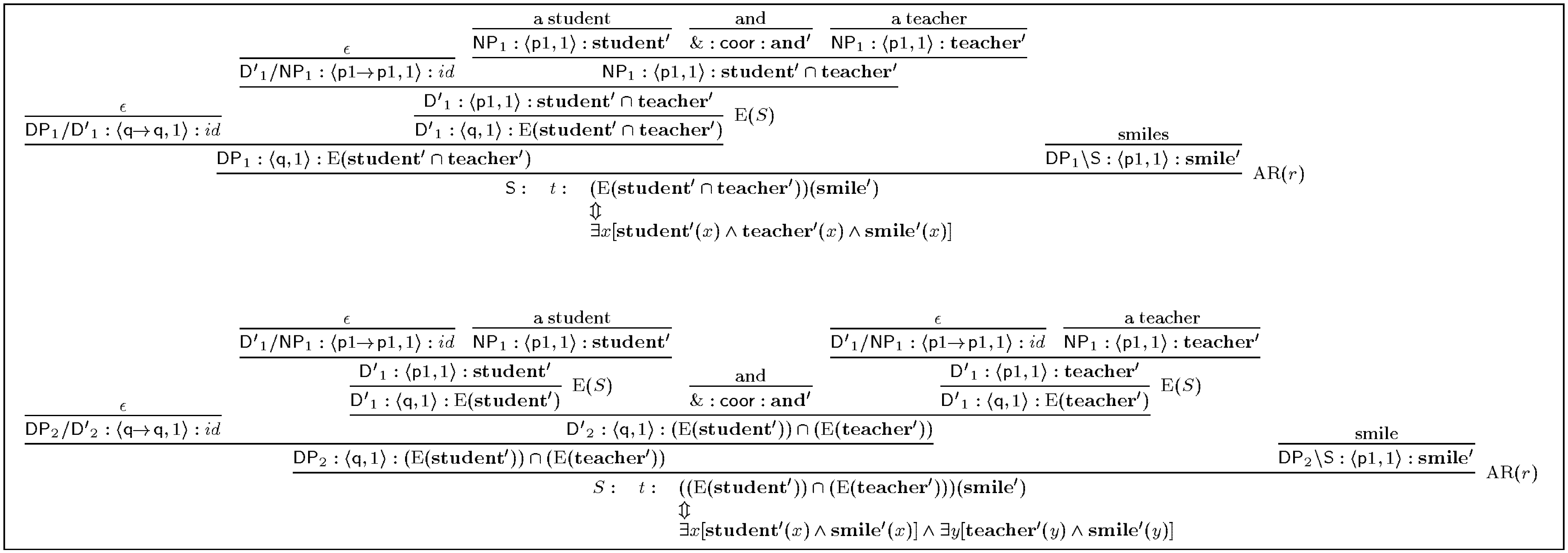

Figure 4: *a student and a teacher* – singular vs. plural