Plural Anaphora in Property-theoretic Discourse Theory

Chris Fox[*]

Abstract

It is possible to use a combination of classical logic and dependent types to represent natural language discourse and singular anaphora (Fox, 1994a). In this paper, these ideas are extended to account for some cases of plural anaphora. In the theory described, $\mathrm{PT}^{\mathcal{D}}$, universal quantification and conditionals give rise to a context in which singular referents within its scope are transformed into plurals. These ideas are implemented in axiomatic Property Theory (Turner, 1992) extended with plurals (Fox, 1993), giving a treatment of some examples of singular and plural anaphora in a highly intensional, weakly typed, classical, first-order logic.

Keywords: Plural Anaphora; Dependent Types; Property Theory

1 Introduction

The goal of the work described in this paper is to systematically translate natural language (NL) discourse into appropriate classical truth conditions within one theory, just using existing notions, such as the classical logical quantifiers, dependent types, and λ-calculus, all with their 'natural' interpretations.

A treatment of singular anaphora has already been implemented in an axiomatic Property Theory[1], PT (Turner, 1992), so giving a treatment of singular pronouns in an highly intensional, weakly typed, first-order logic (Fox, 1994a). This paper extends these ideas to give a treatment of plural pronouns in this framework. Constraints of space mean that just some the salient details of the treatment can be presented in the body of the paper: most of the Property-theoretic details are left to the appendices.

In a sense, the original treatment of singular pronouns—described in §3 and elsewhere (Fox, 1994a)—is a classical revision of the treatment of natural language (NL) discourse afforded by constructive theories (Sundholm, 1989; Ranta, 1991; Ahn and Kolb, 1990; Davila-Perez, 1994) but where a classical, non-constructive view of propositions is adopted. This seems a natural extension to existing work in the semantics of NL in Property Theory (for example (Chierchia, 1982; Chierchia and Turner, 1988; Kamareddine, 1988; Ramsay, 1990; Fox, 1993)), as some versions of Property Theory embody constructive type theories such as Martin-Löf's Type Theory (MLTT) without any alteration of the basic the theory (Smith, 1984; Turner, 1992). The treatment of plurals, given here, presents even more of a departure from the MLTT-like analysis of NL.

We are concerned with examples like:

> Every man walked in. They whistled.

although some mention will be made of how to treat anaphoric reference back to combined antecedents, as in:

> If a man and woman marry, they must be mad.

[*] Mathematical and Computing Sciences, Goldsmiths College, University of London, C.Fox@gold.ac.uk

[1] PT is presented in Appendix A.

Note that it is not the intention of this work to find a treatment of some of the more recalcitrant linguistic phenomena, such as dependent anaphora, and non-logical quantification. Nor is it the intention to provide a means of mechanically *selecting* the appropriate antecedents for singular and plural pronouns. Rather, it shows how cases plural anaphora which arise in simple discourse with conditionals and the universal 'every' can be dealt with entirely within an intensional, classically quantified theory.

The paper is perhaps more concerned with a foundational issue in semantics—that of determining the weakest formal system in which we can conduct NL semantics—rather than with obtaining a wide and accurate coverage of linguistic phenomena. Effectively, discourse is to be represented by increasing the *expressiveness* of a first order theory, without increasing its *power*.

In this proposal, the production of plural antecedents from universal and conditional discourse also accounts for the inaccessibility of singular antecedents in these settings.

In the body of this paper, we gloss over most of the fine technical details, and just give the flavour of the core intuitions. For example, a notion of felicitous discourse is characterised by the theory presented in the appendices. This can be used to avoid considering the truth conditions of discourse with inappropriate anaphoric reference. There is insufficient space to describe the details[2]. Before presenting the treatment of plural anaphora, we first give some details of the basic theory together with its treatment of singular anaphora.

2 The Basic Theory and Singular Pronouns

With the semantics of NL in MLTT, as adopted by Ranta and Davila-Perez for example, indefinites, sentence concatenation, conjunction and relative clauses are interpreted by the dependent sum operator Σ. Implication and universal quantification are represented by the dependent functional type operator Π (Ranta, 1991; Davila-Perez, 1994).[3]

Although we will make use of the dependent types that appear in constructive type theories, we do not adopt the constructive semantics programme; instead, we make use of the dependent types within an essentially classical theory. To do so requires that compositionality is explicitly weakened so that the interpretation of a determiner, for example, is context sensitive.

It is perhaps important to note that in MLTT propositions and types (or properties) are seen to belong to the same category: in some sense propositions are properties that are satisfied by objects corresponding with proofs of the appropriate proposition. It is this uniformity of the categories of semantic primitives in MLTT which allows so many constructs in NL semantics to be interpreted by just two type operators. To have a classical analysis, we can follow the constructive approach in so far as it use one operator to represent indefinites, concatenated sentences, conjunction and relative clauses and another for universals and implication, but then give a different interpretation to those operators depending upon their context.

In this theory, we shall use the two 'operators' ς, π where we would use Σ, Π in an MLTT based semantics of NL.[4] These operators are used in expressions of the form

[2] But see (Fox, 1994b; Fox, 1994a; Fox, 1993)

[3] The expression Σfg specifies a pair of objects $\langle a, b\rangle$ such that $a \varepsilon f$ and $b \varepsilon ga$. In the semantics of natural language this can be used to create nested pairs of discourse references. Combinations of selector functions operating on the specified pairs can then be used to extract appropriate individuals whenever a singular pronoun appears. The functional expression Πfg specifies a function h which takes elements $a \varepsilon f$ to elements of ga. In the setting of natural language semantics, these types effectively block singular anaphoric reference into the scope of a universal quantifier or a conditional: selector functions can only operate on nested pairs of individual, and not functions.

[4] We do not have to take ς, π to be primitive. For example, we could let them form a triple consisting of each of the three possible interpretations. Then when evaluating the truth conditions of a discourse we just *project* the appropriate interpretation. Here we will write $^{\kappa}(t)$ for the interpretation of t as a classical proposition, as in (1) below, and $^{\tau}(t)$ for either of the interpretations as a type, (2) and (3) below.

$\langle\varsigma,\tau,\varphi\rangle, \langle\pi,\tau,\varphi\rangle$. These expressions can be thought of a kind of *Generalised Quantifier* representation, with τ as the *domain*, φ as the *body* and ς or π as the quantifiers. Both of these operators have three different interpretations depending on whether:[5]

1. they have widest scope (or widest scope in the body φ), in which case they behave like the classical quantifiers $\exists, \forall$ (respectively);
2. they occur in the domain τ, but have no further ς, π expressions in their body in turn; in which case they behave like two new type operators not available in MLTT (namely, as a *separation type*, and a kind of identity function, respectively);
3. they occur in the domain τ, and, in turn, have a further ς, π expression in their body, in which case they behave like Σ, Π (respectively) in MLTT.

We shall exemplify these different interpretations, first for ς. The sentence:

A man laughed.

is represented by:

$$\langle\varsigma, \mathit{man}', \lambda x.\mathit{laughed}'x\rangle$$

Its interpretation as a classical proposition is given by:

$${}^{\kappa}\langle\varsigma, \mathit{man}', \lambda x.\mathit{laughed}'x\rangle$$

which leads to ς being interpreted as $\exists$, giving the sentence the truth conditions:[6]

$$\exists x \dot{\varepsilon} \mathit{man}'.\mathit{laughed}'x$$

In practice, the interpretation is a little richer that this. A ς expression as a proposition:

$${}^{\kappa}\langle\varsigma, f, g\rangle$$

is interpreted as:

$$\exists x \dot{\varepsilon}\, {}^{\tau}f.{}^{\kappa}(gx)$$

We can assume that if f is an atomic type then ${}^{\tau}f$ has the same extension as f, and if g is an atomic proposition, then ${}^{\kappa}g$ has the same truth conditions as g. During the rest of this section, the details of how ${}^{\kappa}$ and ${}^{\tau}$ are used to obtain the classical truth conditions will be glossed over. A more rigorous account is given in Appendix C.

The sentence:

A man owns a donkey.

is represented by:

$$\langle\varsigma, \mathit{man}', \lambda x.\langle\varsigma, \mathit{donkey}', \lambda y.\mathit{own}'xy\rangle\rangle$$

[5] In fact it is not quite as described: in practice we need to keep track on whether an expression is being interpreted as a *type* or as a classical *proposition*. These classifications do not correspond 1–1 with the notions *domain* and *body* but there is insufficient room to explain the full recursive definition. The behaviour as described is sufficiently accurate to cope with the examples in this paper.

[6] Note that here we use the notation $\dot{\varepsilon}$. In the full theory (presented in appendices) we need additional constraints on the terms over which quantified variables can range. For example, here we want x only to range over those terms that can be denoted by natural language. This is a technical trick that allows us to capture a notion of felicitous discourse. The notation is used to act as a reminder of these additional complexities in the full version of the theory.

Also, we just have $\mathit{laughed}'x$, where, in full-dress, this should really be $\mathrm{T}(\mathit{laughed}'x)$, so as to distinguish between the extensional and intensional levels of the theory. However, this is just a distracting Property-theoretic detail: the appendices give the full theory with all the correct typing constraints and extensional-intensional distinctions.

and when it is interpreted as a proposition it has the truth conditions:

$$\exists x \dot{\varepsilon} \mathit{man}'.\exists y \dot{\varepsilon} \mathit{donkey}'.\mathit{own}' xy$$

Note that as in (1) above, the first ς has wide scope, and the second has wide scope within the *body* of the first.

As an illustration of the second alternative, the sentence:

A man walked in. He whistled.

is represented by:

$$\langle \varsigma, \langle \varsigma \mathit{man}', \lambda y.\mathit{walked\text{-}in}' y\rangle, \lambda x.\mathit{whistled}' \mathit{he}\rangle$$

where the first ς represents sentence concatenation. In the truth conditions it will be interpreted as $\exists$. The second ς will be interpreted as a *separation type* of the form $\{x \varepsilon \tau.\varphi\}$. This gives the discourse the following truth conditions:

$$\exists x \dot{\varepsilon} \{y \varepsilon \mathit{man}'.\mathit{walked\text{-}in}' y\}.\mathit{whistled}' \mathit{he}$$

The separation type is not available in MLTT.

In effect, the proposition $\mathit{walked\text{-}in}' y$ is like a *test* in Dynamic Logic: it adds no more referents to the context (Groenendijk and Stokhof, 1990; Groenendijk and Stokhof, 1991).

The pronoun is resolved to "the man who walked in" by replacing it with x. It should be clear that these truth conditions are then equivalent to:

$$\exists y \dot{\varepsilon} \mathit{man}'.(\mathit{walked\text{-}in}' y \;\&\; \mathit{whistled}' \mathit{he})$$

The discourse:

A farmer owns a donkey. He beats it.

can be used to illustrate the final case. It has the representation:

$$\langle \varsigma, \langle \varsigma, \mathit{farmer}', \lambda f.\langle \varsigma, \mathit{donkey}', \lambda d.\mathit{own}' f d\rangle\rangle, \lambda x.\mathit{beats}'(\mathit{he})(\mathit{it})\rangle$$

The middle ς (used in the representation of "a farmer") has another ς expression in its body (used to represent "a donkey"). This means that it has to be interpreted as a Σ operator, so as to form a pair of objects, namely the farmer and the donkey:

$$\exists x \dot{\varepsilon} (\Sigma \mathit{farmer}' (\lambda f.(\{d \varepsilon \mathit{donkey}'.\mathit{own}'(f)(d)\}))).\mathit{beats}'(\mathit{he})(\mathit{it})$$

So the existentially quantified variable x will be a farmer-donkey pair. The pronouns *he*, *it* can be resolved by replacing them with the functions $\mathsf{fst}(x), \mathsf{snd}(x)$ respectively. These functions have the obvious behaviour when their argument is a pair.

This expression is then equivalent to:

$$\exists \langle f, d\rangle.f \dot{\varepsilon} \mathit{farmer}' \;\&\; d \dot{\varepsilon} \mathit{donkey}' \;\&\; \mathit{own}'(f)(d) \;\&\; \mathit{beats}'(f)(d)$$

At the top level of analysis, we use classical quantifiers to give a non-constructive interpretation of NL semantics. We use the Σ type when we need to form nested pairs of discourse referents, which can then be passed on to subsequent discourse. The separation type allows us to keep a classical notion of proposition for the propositional body when it contains no further ς or π expressions.

Clearly, the appropriate interpretation of an expression may change as a discourse progresses. Thus the truth conditions for a discourse must be re-evaluated if the discourse is extended. This extra step in the evaluation process is perhaps rather like the application of *existential closure* in Discourse Representation Theory (Kamp, 1981; Kamp and Reyle, 1993), and the closing off of the *discourse continuation* in some forms of dynamic semantics (Groenendijk and Stokhof, 1990).

In general, singular pronouns are resolved by replacing them with some selector function $\mathsf{sel}(x)$, defined as follows:

$$\mathsf{sel}(x) ::= x \mid \mathsf{fst}(\mathsf{sel}(x)) \mid \mathsf{snd}(\mathsf{sel}(x))$$

It is as if we have a version of Discourse Representation Theory where the discourse referents are put into some binary tree, and we select a referent by specifying the path to it from the root of the tree.[7]

3 Resolving Plural Anaphora

In this paper, we wish to introduce a treatment of plural anaphora, as they occur in examples such as:

> Every woman hit a man. They cried.

In effect, we are going to define an interpretation of π expressions that 'pluralises' constituent discourse referents in order to provide the antecedents of plural anaphora. Initially, we are interested in referring back to plural antecedent within the context of universals, which are represented by π. So, we must first discuss the nature of the interpretation of π in the various relevant contexts.

As with ς, when π appears with widest scope (or widest scope within the body of a ς or π expression), we interpret it as a quantifier—a universal quantifier in this case. The sentence:

> Every man walked.

is represented by:

$$\langle \pi, \mathit{man}', \lambda x.\mathit{walked}'x \rangle$$

which is interpreted as:

$$\forall x \dot{\varepsilon} \mathit{man}'.\mathit{walked}'x$$

Of course, at this top most level, there are no subsequent plural pronouns, so we do not need to worry about providing appropriate referents.

When a π expression appears in the domain of a ς or π expression as it has a further such expression in its body, then we could try to interpret it as a Π type.

> Every farmer owns a donkey. They have long ears.

would then be represented by:

$$\langle \varsigma, \langle \pi, \mathit{farmer}', \lambda f.\langle \varsigma, \mathit{donkey}', \lambda d.(\mathit{own}' f d) \rangle \rangle, \lambda x.(\mathit{long\text{-}ears}' \mathit{they}) \rangle$$

As before the outer-most ς represents sentence concatenation, and will be interpreted as existential quantification and the inner-most ς is interpreted as a separation type. Interpreting the π operator as a dependent functional type operator Π gives:

$$\exists x \dot{\varepsilon} (\Pi(\mathit{farmer}'))(\lambda f.\{d \varepsilon \mathit{donkey}'.\mathit{own}' f d\}).\mathit{long\text{-}ears}' \mathit{they}$$

To re-iterate, the meaning of the Π expression $\Pi f g$ is a type which holds of a term h if it is a function that takes elements a of type f to elements of type ga.

[7] The context in which a part of a discourse is analysed is then that of an expression within the scope of a classical quantifier that ranges over appropriate binary trees of discourse referents. It is in part the complexity and variety of objects over which the quantifiers can range that leads to the complex first-order 'typing' rules in Appendix C.

In this case, the Π expression specifies a function from farmers to donkeys that are owned by the farmer in question.

It might now be apparent how we could resolve the plural pronoun "they" in this example. The domain of the function x specified by the Π expression will be the set of farmers, and its codomain will be the set of owned donkeys. If we adopt a lattice-theoretic approach to the semantics of plural terms (Link, 1983), then the the pronoun can be resolved by replacing it by either the supremum (Gratzer, 1978) of the domain of x, to refer to the farmers, or by the supremum of the codomain of x to refer to the owned donkeys.

There are two objections to this proposal. First, the use of the supremum of just the domain or codomain of a function is not sufficient in all cases. In the representations of some discourses, we might have functions whose domains and codomains are nested pairs. This situation would arise in the discourse "If a farmer owns a donkey he beats it. They are really cruel." In this case we would have to select an appropriate type of individual from the domain or codomain.

Second, the domain and codomain of a function need not be worn 'on its sleeve': given an appropriate function we cannot necessarily obtain the relevant (co)domain. The function may belong to more than one type. We could try to get around this second criticism by using a revised interpretation of Π, where if $w\varepsilon(\Pi' fg)$ then $w = \langle h, \langle f, g\rangle\rangle$ where $h\varepsilon(\Pi fg)$. Then we can obtain the domain f and derive the codomain $\{z : \lceil \exists w\dot{\varepsilon} f.(z\varepsilon(gw))\rceil\}$ of h using selector functions.[8] Subsequent plural anaphora can refer to the supremum of these. However, this proposal does not address the first problem, where plural antecedents may have to be abstracted from within other structures.

There is another possibility where we abandon the Π interpretation of π and instead, when interpreting a π expression as a type we make it behave a bit like a ς expression, but where all the constituent witnesses to that type are replaced by their suprema, and if the type is inhabited, then the relevant universal condition must be satisfied. As an informal example: the representation of "every farmer owns a donkey" when interpreted as a type must be the set of pairs of the supremum of farmers, and the supremum of farmer owned donkeys. The set will be inhabited iff every farmer owns a donkey. (Of course, if it is inhabited it will be a singleton pair.)

Schematically, $\langle \pi, f, g\rangle$ interpreted as a type will be:

$$\{x : \lceil x = X \& \forall x \dot{\varepsilon} f.gx\rceil\}$$

In effect, X will be a 'pluralised' witness to $\langle \varsigma, f, g\rangle$ interpreted as a type (there will be only one such witness, which allows us to use equality here). This requires that we recurse through the latter type, and for each constituent in its witness we obtain an appropriate selector function which is then used to obtain an appropriate supremum in the corresponding position of the witness to the π expression. To obtain this result formally is a little involved, as these are *dependent* type expressions: to obtain the full range of witnesses that can occur at any particular position in the witness, we need to consider all the possible witnesses to the entire expression. We cannot consider any one part of the expression in isolation.

The term X can be found by way of some expression that depends upon the a 'pluralisation' of the corresponding ς expression: $\bigsqcup\langle \varsigma, f, g\rangle$. However, to define the appropriate equivalences here, we need to recurse through the structure of t, keeping track of the original t and building appropriate selectors to the individual witnesses to the type ${}^{T}t$. For this we need a richer notation:

$$\bigsqcup(t) = \bigsqcup_{s}^{t} t$$

[8] Note that we write $\lceil p \rceil$ when we really mean to have a Property-theoretic *intension* which, when a proposition, has the same truth conditions as p. This notation is used here for clarity. In the appendices, we use a distinct language of terms to express such intensions.

where we give the term $\bigsqcup_s^t t$ the effect of recursing through the type t in the argument, and building the suprema of witnesses to t using the selector s as part of the recursion. The original type t is recorded in the superscript. Initially, we need a *null* selector (an identity function) for s above. We can use $\lambda x.x$ for this, so X can be defined as follows:

$$X = \bigsqcup \langle \varsigma, f, g \rangle = \bigsqcup_{\lambda x.x}^{\langle \varsigma, f, g \rangle} \langle \varsigma, f, g \rangle$$

In the recursive case, where g contains the representation of further nominals, we want to form pairs of the 'pluralised' members, corresponding with the singular pairs that witness ${}^{\tau}\langle \varsigma, f, g \rangle$. So, in this case, $\bigsqcup_s^t \langle \varsigma, f, g \rangle$ will be equivalent to:

$$\langle \bigsqcup_{\mathsf{fst}(s)}^{t} f, \bigsqcup_{\mathsf{snd}(s)}^{t} g \rangle$$

There are two base cases, (1) where the argument is $\langle \varsigma, f, g \rangle$, and g does not include the representation of a nominal, ie. it is a *test*. In this case we want just to return the suprema of the witnesses to ${}^{\tau}\langle \varsigma, f, g \rangle$ in the context provided by t. We will use the notation $\sigma x \varphi$ to mean the supremum of terms for which φ holds. The relevant witnesses can be picked out from the witness to t with the selector s. In this case, $\bigsqcup_s^t t'$ is equivalent to:

$$\sigma x (\lceil \exists y \dot{\varepsilon}^{\tau} t y . s y = x \rceil)$$

Base case (2) is where the argument is is another 'pluralised' context $\langle \pi, f, g \rangle$. Here, we just return the witness to ${}^{\tau}\langle \pi, f, g \rangle$, in the context provided by t, so $\bigsqcup_s^t \langle \pi, f, g \rangle$ is equivalent to:

$$\sigma x (\lceil \exists y \dot{\varepsilon}^{\tau} t . s y = x \rceil)$$

Here, σ acts like Russell's ι.

Now, assuming that various constraints are met, the truth conditions of:

$$s \varepsilon^{\tau} \langle \pi, f, g \rangle$$

will be those of:

$$s \varepsilon \{ y : \lceil y = \bigsqcup \langle \varsigma, f, g \rangle \,\&\, \forall z \dot{\varepsilon} f . {}^{\kappa}(g z) \rceil \}$$

Plural anaphora can be made to be selector functions, as in the singular case. Felicity of anaphoric reference can be ensured by requiring that pronouns select the appropriate kind of object (singular or plural in this case) for the discourse to be 'meaningful'. This accounts for cases of plural anaphora where the antecedents arise from the occurrence of some form of universal quantification.

4 Combined Antecedents

We must generalise the use of selector functions to account for plural anaphoric reference to the denotations of combined nominals, as in:[9]

> If [a man]$_i$ and [a woman]$_j$ marry [they]$_{i+j}$ must be mad.
> [One student]$_i$ went to Paris. [Every lecturer]$_j$ went to Lyons. [They]$_{i+j}$ (all) had a good time.

If plural anaphora are always plural, then there are the following possible kinds of reference:

[9] There is not sufficient space here to describe the semantics of conjunction. Essentially it makes use of the ς operator. The semantics of disjunction can be implemented using π and negation, which in turn can be represented using π.

1. reference to two or more singulars;
2. reference to one or more plural and one or more singular;
3. reference to one or more plural antecedents.

We can cover these cases by allowing joins of selectors to be substituted for plural anaphora, such as $\lambda x.\mathsf{sel}(x) \sqcup \mathsf{sel}'(x)$. So singular pronouns are resolved by simple selectors:

$$\mathsf{sel}(x) ::= x \mid \mathsf{fst}(\mathsf{sel}(x)) \mid \mathsf{snd}(\mathsf{sel}(x))$$

and plural pronouns are resolved by plural selectors:

$$\mathsf{Psel}(x) ::= \mathsf{sel}(x) \mid \lambda x.(\mathsf{Psel}(x) \sqcup \mathsf{Psel}(x))$$

Although there is insufficient space to present the details of felicitous v. infelicitous discourse in this framework, it can be seen that 'coindexing' of a singular pronoun into the scope of a universal will infelicitous because no singular antecedents are made available by universals.[10]

5 Conclusions

We have sketched how we can treat singular and plural anaphora just using selector functions, dependent types, and pairs, all of which can be implemented in the λ-calculus, together with classical quantifiers, and a Link-style treatment of plurals. In this sense, we can see that existing classical machinery is adequate for obtaining the classical truth conditions of discourse.

In this theory, rather than stipulating that singular discourse referents are inaccessible when they occur within the scope of a universal or conditional, their inaccessibility is a consequence of the 'plural context' which is created by such expressions. It is the very act of creating appropriate antecedents for plural pronouns that leads to the loss of singular referents from such contexts: the singular referents are inaccessible because they have been 'transformed' into plurals.

It could be argued that the ς and π expressions used in the initial representation of discourse fall outside the domain of classical logic. Further, as they have perhaps a rather complex case analysis, they do not really satisfy the requirements of new logical constants. As was hinted in the text, however, these operators can be implemented as triples, in which case their interpretation as propositions and as types, produced by $^{\kappa}$ and $^{\tau}$, can be obtained by using projection functions, all of which can be implemented in λ-calculus.

Even given the systematic nature in which the theory obtains the classical truth conditions, there is a sense which the theory is not strictly compositional: the contribution that a particular expression makes to the final truth conditions depends upon its context. Of course, because of the nature of the phenomena, all theories that deal with anaphora must bend the rule of compositionality to some extent, especially when we consider going all the way from discourse to truth conditions expressed in classically quantified logic. In DRT, a 'box' at the top level effectively gives rise to existential quantification, whereas as an antecedent to a condition there is no existential quantification in the truth conditions (Kamp, 1981; Kamp and Reyle, 1993). Similarly in DPL, an 'existential' quantifier is sometimes interpreted as classically existential, but in the antecedent to a conditional it is interpreted as classically universal (Groenendijk and Stokhof, 1990; Groenendijk and Stokhof, 1991).

The ideas sketched in the body of the paper have been implemented in a first-order Property-theoretic framework, as given in the Appendices. This then gives a treatment

[10] In the Property-theoretic implementation, we equate 'felicity' with 'proposition-hood': only expressions that are provably propositions can have their truth conditions evaluated. This is much like the treatment of the logical paradoxes in PT. In the theory described in Appendices B and C, the representation of a discourse is only provably a proposition if pronouns therein refer to appropriate antecedents.

of discourse phenomena in an highly intensional logic with a semi-decidable proof theory. There may well be other ways in which these ideas can be implemented. For example, we could use a type theory with separation types, similar to one proposed by Feferman (Feferman, 1984).

This work shows that we can obtain classically quantified, first-order representations of the more straightforward examples of discourse without at any stage resorting to more powerful systems. More work is needed to see whether this approach can be extended to more recalcitrant discourse phenomena.

A PT (Property Theory)

The version of Property Theory presented here is Ray Turner's axiomatisation of Aczel's Frege Structures (Turner, 1990; Turner, 1992; Aczel, 1980), PT.

The Language of terms, basic vocabulary:

Individual variables:	$x, y, z, \ldots$
Individual constants:	$c, d, e, \ldots$
Logical constants:	$\vee, \wedge, \neg, \Rightarrow, \Xi, \Theta, \approx$

Inductive Definition of Terms:

1. Every variable or constant is a term.
2. If t is a term and x is a variable then $\lambda x.t$ is a term.
3. If t and t' are terms then $t(t')$ is a term.

The Language of Wff

1. If t and s are terms then $s = t, \mathrm{P}(t), \mathrm{T}(t)$ are atomic wff.
2. If φ and φ' are wff then $\varphi \,\&\, \varphi'$, $\varphi \,\mathrm{v}\, \varphi'$, $\varphi \rightarrow \varphi'$, $\sim\varphi$ are wff.
3. If φ is a wff and x a variable then $\exists x\varphi$ and $\forall x\varphi$ are wff.

The theory is governed by the following axioms:

Axioms of The $\lambda\beta$-Calculus

$$\begin{aligned} \lambda x.t &= \lambda y.t[y/x] \; y \text{ not free in } t \\ (\lambda x.t)t' &= t[t'/x] \end{aligned}$$

This defines the equivalence of terms.

The closure conditions for propositionhood are given by the following axioms:

Axioms of Propositions

1. $\mathrm{P}(t) \,\&\, \mathrm{P}(s) \rightarrow \mathrm{P}(t \wedge s)$
2. $\mathrm{P}(t) \,\&\, \mathrm{P}(s) \rightarrow \mathrm{P}(t \vee s)$
3. $\mathrm{P}(t) \,\&\, (\mathrm{T}(t) \rightarrow \mathrm{P}(s)) \rightarrow \mathrm{P}(t \Rightarrow s)$
4. $\mathrm{P}(t) \rightarrow \mathrm{P}(\neg t)$
5. $\forall x \mathrm{P}(t) \rightarrow \mathrm{P}(\Theta\lambda x.t)$
6. $\forall x \mathrm{P}(t) \rightarrow \mathrm{P}(\Xi\lambda x.t)$
7. $\mathrm{P}(s \approx t)$

Truth conditions can be given for those terms that are propositions:

Axioms of Truth

1. $\mathrm{P}(t) \,\&\, \mathrm{P}(s) \to (\mathrm{T}(t \wedge s) \leftrightarrow \mathrm{T}(t) \,\&\, \mathrm{T}(s))$
2. $\mathrm{P}(t) \,\&\, \mathrm{P}(s) \to (\mathrm{T}(t \vee s) \leftrightarrow \mathrm{T}(t) \,\mathrm{v}\, \mathrm{T}(s))$
3. $\mathrm{P}(t) \,\&\, (\mathrm{T}(t) \to \mathrm{P}(s)) \to (\mathrm{T}(t \Rightarrow s) \leftrightarrow \mathrm{T}(t) \to \mathrm{T}(s))$
4. $\mathrm{P}(t) \to (\mathrm{T}(\neg t) \leftrightarrow \ \sim\mathrm{T}(t))$
5. $\forall x \mathrm{P}(t) \to (\mathrm{T}(\Theta\lambda x.t) \leftrightarrow \forall x \mathrm{T}(t))$
6. $\forall x \mathrm{P}(t) \to (\mathrm{T}(\Xi\lambda x.t) \leftrightarrow \exists x \mathrm{T}(t))$
7. $\mathrm{T}(t \approx s) \leftrightarrow t = s$
8. $\mathrm{T}(t) \to \mathrm{P}(t)$

The last axiom states that only propositions may have truth conditions.

Note that the quantified propositions $\Theta\lambda x.t$, $\Xi\lambda x.t$ can be written as $\Theta x(t)$, $\Xi x(t)$, where the λ-abstraction is implicit.

Basic Types in PT The notions of n-place relations can be defined recursively:

1. $\mathit{Rel}_0(t) \leftrightarrow \mathrm{P}(t)$
2. $\mathit{Rel}_n(\lambda x.t) \leftrightarrow \mathit{Rel}_{n-1}(t)$

We can write $\mathit{Rel}_1(t)$ as $\mathrm{Pty}(t)$ and and $\lambda x.t$ as $\{x : t\}$. In keeping with this set-like notation, we can write $\mathrm{T}(tx)$ as $x\varepsilon t$, especially if t is a property.

Type Operators in PT We can give definitions for various type operators (Turner, 1992). Only intersection and disjoint union, and dependent functional and sum will be illustrated here:

$$\begin{array}{lll}
\cap & =_{\mathrm{def}} & \lambda f.\lambda g.\{x : fx \wedge gx\} \\
\oplus & =_{\mathrm{def}} & \lambda f.\lambda g.\{z : (\mathsf{fst}(z) \approx 0 \wedge f(\mathsf{snd}(z))) \vee (\mathsf{fst}(z) \approx 1 \wedge g(\mathsf{snd}(z)))\} \\
\Pi & =_{\mathrm{def}} & \lambda f.\lambda g.\{h : \Theta x(fx \Rightarrow gx(hx)\} \\
\Sigma & =_{\mathrm{def}} & \lambda f.\lambda g.\{h : f(\mathsf{fst}(h)) \wedge g(\mathsf{fst}(h))(\mathsf{snd}(h))\}
\end{array}$$

Pairs $\langle\cdot\rangle$ and $\mathsf{fst}, \mathsf{snd}$ have their usual definitions:

$$\begin{array}{lll}
\mathsf{fst} & =_{\mathrm{def}} & \lambda p.p\lambda xy.x \\
\mathsf{snd} & =_{\mathrm{def}} & \lambda p.p\lambda xy.y \\
\langle x, y\rangle & =_{\mathrm{def}} & \lambda z.z(x)(y)
\end{array}$$

B PT^p (PT with a mereology)

Language of mereological/plural terms To the basic vocabulary of terms is added:

Predicative constant: δ
Operative constants: $\sqcup, \sigma$

It is intended that δ will hold of natural language denotable terms. The term σ is the supremum operator, and the term $\sqcup$ is the summation operator.

Additional Axioms and Definitions

1. $a \sqcup b = b \sqcup a$ (Symmetry)
2. $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ (Associativity)
3. $a \sqcup a = a$ (Idempotence)
4. $t \leq s =_{\mathrm{def}} t \sqcup s = s$ (Definable ordering)
5. $\mathrm{P}(\delta t)$ (Denotability, internal)
6. $\Delta t =_{\mathrm{def}} \mathrm{T}(\delta t)$ (Denotability)
7. $\Delta a \,\&\, \Delta b \to \Delta(a \sqcup b)$ (Closure of Δ under $\sqcup$)

8. $\forall p((\mathrm{Pty}(p) \,\&\, \forall x(\mathrm{T}(px) \rightarrow$
   $\Delta(x)) \,\&\, \exists x(\mathrm{T}px)) \rightarrow \Delta(\sigma xpx))$ (Completeness of σ in Δ)
9. $\mathrm{Pty}_\Delta(p) =_{\mathrm{def}} \forall x(\Delta(x) \rightarrow \mathrm{P}(px))$ (Property of denotables)
10. $\forall_\Delta x\varphi =_{\mathrm{def}} \forall x(\Delta x \rightarrow \varphi)$ (Quantification over Δ)
11. $\exists_\Delta x\varphi =_{\mathrm{def}} \exists x(\Delta x \,\&\, \varphi)$ (Quantification over Δ)[11]
12. $\sigma_\delta x(px) =_{\mathrm{def}} \sigma x(\delta x \wedge px)$ (Supremum within Δ)
13. $\forall_\Delta xy(x \nleq y \rightarrow \exists_\Delta u(u \leq x \,\&\, u \nleq y))$ (Separability)
14. $\forall p\forall_\Delta y((\mathrm{Pty}_\Delta(p) \,\&\, \mathrm{T}(py)) \rightarrow y \leq \sigma_\delta xpx)$ (Upper bound)
15. $\forall p\forall_\Delta y((\mathrm{Pty}_\Delta(p) \,\&\, \forall_\Delta x(\mathrm{T}(px) \rightarrow x \leq y)) \rightarrow \sigma_\delta xpx \leq y)$ (Least upper bound)
16. $\forall p\forall_\Delta u((\mathrm{Pty}_\Delta(p) \,\&\, u \leq \sigma_\delta xpx) \rightarrow$
    $(\exists_\Delta z(pz \,\&\, u \leq z) \mathrm{\,v\,} \exists_\Delta z(pz \,\&\, z \leq u)))$ (Sup prime atoms)
17. $\top =_{\mathrm{def}} \sigma_\delta x(x \approx x)$ (Top)[12]

C PT$^{\mathcal{D}}$ (PT^{p} with discourse structures)

The expression $\mathsf{type}_t(s)$ says that s viewed as a type, ${}^{\tau}s$, can form propositions with elements of t. In the case of terms of the form $\langle d, f, g\rangle$ where g contains no further nominals (like a *test*), the expressions will have the same type as f. In the recursive cases, the 'type' of the expression will be the disjoint union of the 'types' of the constituents.

1. $\mathrm{Pty}_\Delta(f) \rightarrow \mathsf{type}_\delta(f)$
2. $\mathsf{type}_a(f) \,\&\, \forall x\varepsilon a.\mathrm{P}(gx) \rightarrow \mathsf{type}_a\langle d, f, g\rangle \,\&\, {\sim}\mathsf{dep}\langle d, f, g\rangle$
3. $\mathsf{type}_a(f) \,\&\, \forall x\varepsilon a.\mathsf{type}_b(gx) \rightarrow \mathsf{type}_{\Sigma ab}\langle d, f, g\rangle \,\&\, \mathsf{dep}\langle d, f, g\rangle$

Propositionhood for expressions of the form ${}^{\kappa}t$:

1. $\mathrm{P}(t) \rightarrow \mathrm{P}({}^{\kappa}t)$
2. $\exists t.\mathsf{type}_t\langle \varsigma, f, g\rangle \rightarrow \mathrm{P}^{\kappa}\langle \varsigma, f, g\rangle$
3. $\exists t.\mathsf{type}_t\langle \pi, f, g\rangle \rightarrow \mathrm{P}^{\kappa}\langle \pi, f, g\rangle$

Propositionhood for expressions of the form ${}^{\tau}t$:

1. $\mathrm{Pty}_\Delta(p) \,\&\, \Delta(x) \rightarrow \mathrm{P}({}^{\tau}px)$ (i.e. $\mathsf{type}_\delta(p) \rightarrow \forall x\varepsilon\delta.\mathrm{P}({}^{\tau}px)$)
2. $\mathsf{type}_t\langle \varsigma, f, g\rangle \rightarrow \forall x\varepsilon t.\mathrm{P}({}^{\tau}\langle \varsigma, f, g\rangle x)$
3. $\mathsf{type}_t\langle \pi, f, g\rangle \rightarrow \forall x\varepsilon t.\mathrm{P}({}^{\tau}\langle \pi, f, g\rangle x)$

Truth conditions for expressions of the form ${}^{\kappa}t$:

1. $\mathrm{P}(t) \rightarrow (\mathrm{T}^{\kappa}(p) \leftrightarrow \mathrm{T}(p))$
2. $\exists t.\mathsf{type}_t\langle \varsigma, f, g\rangle \,\&\, \mathsf{type}_a(f) \rightarrow \mathrm{T}^{\kappa}\langle \varsigma, f, g\rangle \leftrightarrow \exists x\varepsilon a.(\mathrm{T}({}^{\tau}fx) \,\&\, \mathrm{T}^{\kappa}(gx))$
3. $\exists t.\mathsf{type}_t\langle \pi, f, g\rangle \,\&\, \mathsf{type}_a(f) \rightarrow \mathrm{T}^{\kappa}\langle \pi, f, g\rangle \leftrightarrow \forall x\varepsilon a.(\mathrm{T}({}^{\tau}fx) \rightarrow \mathrm{T}^{\kappa}(gx))$

Truth conditions for expressions of the form ${}^{\tau}px$:

1. $\mathrm{Pty}_\Delta(p) \,\&\, \Delta(s) \rightarrow \mathrm{T}({}^{\tau}ps) \leftrightarrow \mathrm{T}(ps)$
2. ${\sim}\mathsf{dep}\langle \varsigma, f, g\rangle \,\&\, \exists t(\mathsf{type}_t\langle \varsigma, f, g\rangle \,\&\, s\varepsilon t) \rightarrow \mathrm{T}({}^{\tau}\langle \varsigma, f, g\rangle s) \leftrightarrow s\varepsilon\{y : {}^{\tau}fy \wedge {}^{\kappa}(gy)\}$
3. $\mathsf{dep}\langle \varsigma, f, g\rangle \,\&\, \exists t(\mathsf{type}_t\langle \varsigma, f, g\rangle \,\&\, s\varepsilon t) \rightarrow \mathrm{T}({}^{\tau}\langle \varsigma, f, g\rangle s) \leftrightarrow s\varepsilon\Sigma{}^{\tau}f(\lambda x.{}^{\tau}(gx))$
4. $\exists t.\mathsf{type}_t\langle \pi, f, g\rangle \,\&\, \mathsf{type}_a(f) \,\&\, s\varepsilon t \rightarrow$
   $\mathrm{T}({}^{\tau}\langle \pi, f, g\rangle s) \leftrightarrow s\varepsilon\{y : y \approx \bigsqcup\langle \varsigma, f, g\rangle \wedge \Theta z\varepsilon a({}^{\tau}fz \Rightarrow {}^{\kappa}(gz))\}$

Case (2) is where g does not introduce any more discourse referents, and case (3) is where g introduces more discourse referents. The final case (4) covers terms of the form ${}^{\tau}\langle \pi, f, g\rangle$. The meaning of $\bigsqcup\langle \varsigma, f, g\rangle$ is elaborated below.

[11] Restricting ourselves to these quantifiers, we achieve the effect of a free logic, where free variables range over all terms, and quantified variables range over denoting terms.

[12] A bottom element $\perp$ can be defined as $\overline{\top}$, but it can not be proven that $\Delta\perp$. This allows us to capture a notion of felicitous reference for pronouns and definites.

1. $\bigsqcup\langle\varsigma, f, g\rangle =_{\mathrm{def}} \bigsqcup_{\lambda x.x}^{\langle\varsigma,f,g\rangle} \langle\varsigma, f, g\rangle$
2. $\exists u.\mathsf{type}_u\langle\varsigma, f, g\rangle \mathbin{\&} \mathsf{dep}\langle\varsigma, f, g\rangle \rightarrow \bigsqcup_s^t \langle\varsigma, f, g\rangle = \langle\bigsqcup_{\mathsf{fst}(s)}^t f, \bigsqcup_{\mathsf{snd}(s)}^t g\rangle$
3. $\exists u.\mathsf{type}_u(t') \mathbin{\&} {\sim}\mathsf{dep}(t') \rightarrow \bigsqcup_s^t t' = \sigma x(\Xi y \varepsilon u.({}^{\tau}ty \wedge sy \approx x))$
4. $\exists u.\mathsf{type}_u\langle\pi, f, g\rangle \mathbin{\&} \mathsf{type}_a(f) \rightarrow \bigsqcup_s^t \langle\pi, f, g\rangle = \sigma x(\Xi y \varepsilon a.({}^{\tau}ty \wedge sy \approx x))$

References

Aczel, P. (1980). Frege structures and the notions of proposition, truth and set. In Barwise, Keisler, and Keenan, editors, *The Kleene Symposium*, North Holland Studies in Logic, pages 31–39. North Holland.

Ahn, R. and Kolb, H. (1990). Discourse representation theory meets constructive mathematics. ITK Research Report Number 16, Tilburg University.

Chierchia, G. (1982). Nominalisation and Montague grammar: a semantics without types for natural languages. *Linguistics and Philosophy*, 5:303–354.

Chierchia, G. and Turner, R. (1988). Semantics and property theory. *Linguistics and Philosophy*, 11:261–302.

Davila-Perez, R. (1994). Constructive type theory and natural language. Computer Science Memorandum 206, University of Essex.

Feferman, S. (1984). Towards useful type-free theories 1. *Journal of Symbolic Logic*, 49:61–102.

Fox, C. (1993). *Plurals and Mass Terms in Property Theory.* PhD thesis, University of Essex, Colchester, U.K.

Fox, C. (1994a). Discourse representation, type theory and property theory. In Bunt, H., Muskens, R., and Rentier, G., editors, *Proceedings of the International Workshop on Computational Semantics*, pages 71–80, ITK, Tilburg.

Fox, C. (1994b). Existence presuppositions and category mistakes. University of Essex. m.s.

Gratzer, G. (1978). *General Lattice Theory.* Academic Press, New York.

Groenendijk, J. and Stokhof, M. (1990). Dynamic Montague grammar. ITLI Prepublication Series LP–90–02.

Groenendijk, J. and Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases.* PhD thesis, University of Massachusetts, Amherst. Thesis Distributed by Graduate Linguistics Student Association.

Kamareddine, F. (1988). *Semantics in Frege Structures.* PhD thesis, University of Edinburgh.

Kamp, H. (1981). Theory of truth and semantic representation. In Groenendijk, J., Janssen, T., and Stokhof, M., editors, *Formal Methods in the Study of Language*, Mathematical Centre Tracts 135, pages 277–322. Amsterdam.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic.* Kluwer, Dordrecht.

Link, G. (1983). The logical analysis of plurals and mass terms: A lattice theoretic approach. In Bauerle, R., Schwarze, C., and von Stechow, A., editors, *Meaning, Use and Interpretation of Language*, pages 302–323. Walter de Gruyer, Berlin.

Ramsay, A. (1990). *The Logical Structure of English.* Pitman, London.

Ranta, A. (1991). Intuitionistic categorial grammar. *Linguistics and Philosophy*, 14:203–239.

Smith, J. M. (1984). An interpretation of Martin-Löf's Type Theory in a type-free theory of propositions. *Journal of Symbolic Logic*, 49.

Sundholm, G. (1989). Constructive generalised quantifiers. *Synthese*, 79:1–12.

Turner, R. (1990). *Truth and Modality for Knowledge Representation.* Pitman.

Turner, R. (1992). Properties, propositions and semantic theory. In Rosner, M. and Johnson, R., editors, *Computational Linguistics and Formal Semantics*, Studies in Natural Language Processing, pages 159–180. Cambridge University Press, Cambridge.