

Conservative Generalized Quantifiers and Presupposition

Tim Fernando

Trinity College Dublin

Abstract. Conservativity in generalized quantifiers is linked to presupposition filtering, under a *propositions-as-types* analysis extended with *dependent quantifiers*. That analysis is underpinned by model-theoretically interpretable proofs which inhabit propositions they prove, thereby providing objects for quantification and hooks for anaphora.

1. Introduction

A binary relation D on sets is *conservative* if for all sets A and B , D relates A to B precisely if it relates A to their intersection, $A \cap B$ (Keenan and Stavi 1986).

$$(1) \quad D(A, B) \text{ iff } D(A, A \cap B)$$

Widely assumed to hold for denotations of natural language determiners, conservativity is demonstrably true of some and all.

- (2) a. $\text{some}(A, B) \text{ iff } A \cap B \neq \emptyset$
b. Some ants bite iff some ants are ants that bite.
- (3) a. $\text{all}(A, B) \text{ iff } A \subseteq B$
b. All ants bite iff all ants are ants that bite.

In an extensional setting with exactly one world w , relative to which a true proposition is reduced to $\{w\}$ and a false proposition to \emptyset (these being the sets of worlds in which the respective propositions are true), the equivalences (2a) and (3a) characterize conjunction \wedge and material implication \supset , respectively. That is, some corresponds to \wedge , and all to \supset . These correspondences are not only preserved but arguably take on greater significance if a proposition is identified (instead) with the set of its *proofs* (rather than verifying worlds), under the *propositions-as-types* paradigm,¹ reviewed below. Such an identification provides an explanation not only for conservativity but also presupposition filtering in conjunction and implication, illustrated by (4) and (5).²

- (4) Buganda has a king and the king of Buganda is bald.
- (5) If Buganda has a king, the king of Buganda is bald.

That explanation is essentially the *admittance* account of Karttunen 1974, centered around the picture ' $\Gamma \triangleright A$ ' of a context Γ satisfying the presuppositions of a sentence A . Generalizing $A \wedge B$, $A \supset B$ and $D(A, B)$ to *dependent quantifiers* $(Qx:A)B$ equipped with a variable x to track dependencies on A ,

we can capture sequentiality in context growth by the equivalence (6), where (as in propositions-as-types) a context Γ is a sequence of variable:type declarations, and ' $\Gamma, x:A$ ' is the context obtained by appending to Γ the declaration ' $x:A$ ' that the variable x has type A .

$$(6) \quad \Gamma \triangleright (Qx:A)B \text{ iff } \Gamma \triangleright A \text{ and } \Gamma, x:A \triangleright B$$

The present work applies (6) to explain not only presupposition filtering in (4) and (5) but also the conservativity of determiners, (1).

Inasmuch as admittance \triangleright is a *prima facie* pragmatic notion, and Heim 1983 is lauded for deriving presupposition facts *semantically*, it may seem a curious twist to trace conservativity (surely a semantic property) to the 'pragmatic' condition (6). Twisted or not, it should be pointed out that

- (i) presupposition projection in (5) is, up to a point, independent of the precise semantic interpretation of *if*, be that material implication or some other conditional involving quantification, and
- (ii) whether or not (6) is deemed 'pragmatic', (6) can be understood within a perfectly respectable and rigorous logical system.

Moreover, that logical system offers a natural setting for combining quantification with anaphora. An obstacle facing any system that accounts for (1), (4) and (5) is that in (1), A and B range over sets of objects (e.g. denotations of common nouns), whereas to apply (6) to (4) and (5), A and B must denote propositions (relative to suitable contexts). For a uniform analysis under (6), the key is to treat proofs on a par with objects (in accordance with propositions-as-types) whilst embedding objects witnessing existential claims into proofs (following intuitionistic/constructive prescriptions) which can then be quantified selectively. The upshot is an account of adverbial quantification that is immune from the *proportion problem*, supporting *existential* as well as *universal* donkey readings. (The terminology here is as in, for example, Chierchia 1995.)

In fact, the account has much in common with *Discourse Representation Theory*, DRT, which in its later (*non-classical*) versions (from Kamp and Reyle 1993 on) allow for asymmetric quantification, E-type treatments of plural anaphora, and (in principle) existential donkeys alongside (classical) universal ones. What is distinctive about the present proposal (and behind its novel anaphoric take on conservativity) is its marriage of a model-theoretic conception of generalized quantification with a proof-theoretic approach to anaphora. That marriage rests on the understanding that — crudely put as a slogan —

objects are for quantification, and proofs for anaphora.

The role of proofs in anaphora has been argued at length in, for example,

Ranta 1994 (following Sundholm 1986) and Krahmer and Piwek 1999 (boosting the slogan in van der Sandt 1992 to “Presupposition projection as proof construction” with an eye to bridging and other inferences). To establish anaphoric dependencies, the usual model-theoretic account of quantification must be adjusted slightly, with conservativity falling out as a consequence.

1.1. *Outline of paper*

At the center of the present work is the notion of a dependent quantifier, written $(Qx : A)B$ in (6) above. Roughly put, a dependent quantifier is a generalized quantifier with conservativity and anaphora built in proof-theoretically. Section 2 focuses on dependencies, reviewing how these are introduced into Cartesian products and function spaces for proof-theoretic interpretations of \wedge/\exists and \supset/\forall respectively, and then relating these to the familiar set-theoretic equivalences (2a) and (3a). Section 3 turns to finer points about quantification, refining case quantification in Lewis 1975 to admit more sophisticated notions of eventuality. After a brief look at existential vs universal donkeys and at conservativity, the paper closes in section 4 with an even briefer discussion of the proof-theoretic shift advocated and the broader notion of conservativity at stake.

1.2. *So why read on?*

The formal details below are likely to try the patience of an empirically minded linguist, who may well ask what consequences the combination proposed of models with proofs has for semantic theory. One simple consequence is that in a sentence such as *some ants bite*, the verb phrase *bite* is interpreted as a proposition, relative to a context with a variable ranging over ants. From this interpretation, the set of ants that bite can be extracted; but beyond that, there are proofs that these ants bite, on which notions of eventuality can be based. Under this refined semantics for verb phrases, the intersection $A \cap B$ in conservativity (1) is re-analyzed in (6) as the subset $\{x : A|B\}$ of A satisfying B , where B is treated as a proposition relative to a context containing the typing $x : A$. (Indeed, $\{x : A|B\}$ is arguably closer to the gloss “ants that bite” in (2b) than $A \cap B$, the corresponding phrase for which, “are ants and bite,” is somewhat stilted.) Determiners can still be viewed as relations between sets; the difference lies rather in how the relevant sets are formed, the claim being that conservativity is a consequence of the context change underlying set formation, as well as presupposition satisfaction. Traditional formulations of semantic interpretation and of presupposition projection abstract away context dependence, which (6), in part, restores. The remainder of this paper is meant for the reader intrigued, albeit perhaps puzzled, by this quick sketch.

2. Dependencies

According to propositions-as-types, a proposition is the collection (or type) of

its proofs. To explain what its proofs are, it will be useful, for our purposes, to consider propositions alongside sets of objects (as they appear in the classical set-theoretic conception of generalized quantifiers), the idea being that

$$\frac{\text{object}}{\text{set}} \approx \frac{\text{proof}}{\text{proposition}} .$$

Objects and proofs combine to witness existential propositions, which can be employed to interpret indefinites, as in (7).

- (7) a. An aardvark belched.
 b. $(\exists x:\text{aardvark}) \text{ belched}(x)$

(7a) translates quite naturally to the existential proposition (7b), a proof of which is a pair $\langle a, p \rangle$ consisting of an aardvark a , and a proof p of $\text{belched}(a)$. That is, equating a proposition with its collection of proofs,

$$(\exists x:\text{aardvark}) \text{ belched}(x) = \{ \langle a, p \rangle \mid a:\text{aardvark} \text{ and } p:\text{belched}(a) \} .$$

The colon $:$ can be read as ordinary set membership when applied to sets such as aardvark , but is more intuitively read as *proves* when applied to propositions such as $\text{belched}(a)$. The variable x in the expression $\text{belched}(x)$ marks, in a pair $\langle a, p \rangle$ proving (7b), the dependence of p on a (i.e., the choice of p from $\text{belched}(a)$). This dependence is a crucial innovation over ordinary Cartesian products

$$A \times B = \{ \langle a, b \rangle \mid a:A \text{ and } b:B \} ,$$

the pairs $\langle a, b \rangle$ in which are assembled from independent components a and b . Part of the picture (6) of dependent quantifiers is the mediation of dependencies through a variable x , subject to a mechanism of context Γ and context growth (illustrated in (6) by the step to $\Gamma, x:A$). §2.1 reviews how that is achieved proof-theoretically, while §2.2 takes up the question of model-theoretic interpretation.

2.1. Proof-theoretic intuitions

Let us henceforth use the word *type* for a set, such as aardvark , or a proposition such as $\text{belched}(a)$. Generalizing from aardvark to A and $\text{belched}(x)$ to B , and turning \exists into Σ (for reasons that will become clear shortly), we get (8), where $B[x/a]$ is B with x replaced by a .³

- (8) $(\Sigma x:A)B = \{ \langle a, b \rangle \mid a:A \text{ and } b:B[x/a] \}$
 a. Call the left projection l , so $l\langle a, b \rangle = a$.
 b. Call the right projection r , so $r\langle a, b \rangle = b$.

Notice that the cross \times in the Cartesian product $A \times B$ is transformed into a variable x in $(\Sigma x:A)B$, which is often referred to as a *dependent product*. The

notation Σ as well as its use to interpret \exists (commonly construed semantically as a union) also suggest the label “dependent sum” — which can be confusing, as we will apply Σ to conjunction, not disjunction.

Passing to multi-sentential discourse, consider (9a) and the challenge of translating the occurrence in the second sentence of the pronoun *it*.

- (9) a. An aardvark belched. It was in distress.
- b. $(\Sigma z: (\Sigma x: \text{aardvark}) \text{belched}(x)) \text{ was-in-distress}(lz)$

(9b) translates (9a) by taking a proof z of the translation (7b) of (7a), and projecting its left component (named lz according to (8a)) to anaphorically bind *it*. The proof z is hypothetical inasmuch as it is a variable bound by Σ , the intuition behind (9) being that in order to interpret *it* (at least, under its anaphoric reading), we had better be prepared to take (7) for granted. (Readers familiar with DRT and/or related formalisms may complain that a translation of *it* into the term lz involves a bit more work than one into the variable x . Maybe so. But it should come as no surprise that in practice, that discrepancy is purely notational and easily resolvable.⁴).

Changing (9a) to an implication, consider (10a), the translation of which, (10b), appeals to the construct Π , characterized in (11).

- (10) a. If an aardvark belched it was in distress.
- b. $(\Pi z: (\Sigma x: \text{aardvark}) \text{belched}(x)) \text{ was-in-distress}(lz)$

$$(11) \quad (\Pi x: A)B = \{\text{functions mapping } a: A \text{ to } b: B[x/a]\}$$

Assuming x is a dummy variable (occurring in neither A nor B), the *dependent function space* $(\Pi x: A)B$ reduces to the usual function space $A \rightarrow B$, just as $(\Sigma x: A)B$ reduces to $A \times B$. In (10b), however, the dependence mediated by the variable z bound by Π is indispensable in linking an aardvark that belched with a proof that it was in distress. A function proving (10b) witnesses the claim in (10a) that every aardvark that belched was in distress, just as a pair proving (9b) witnesses (9a). Repeating the move from Σ to Π , we step from (7) to (12).

- (12) a. Every aardvark belched.
- b. $(\Pi x: \text{aardvark}) \text{belched}(x)$

Together (7), (9), (10) and (12) illustrate how to express the usual predicate logic constructs \exists, \wedge and \forall, \supset in terms of Σ and Π .

	<i>force</i>	
	Σ	Π
set	\exists	\forall
proposition	\wedge	\supset
<i>restrictor A</i>		

$(\exists/\forall x: A)B = (\Sigma/\Pi x: A)B$
 $A \wedge/\supset B = (\Sigma/\Pi _ : A)B$
 where $_$ is a dummy variable

The constructs \exists, \wedge, \forall and \supset are understood here in the abstract sense they are used in symbolic logic, and, as such, do not fully account for how their English counterparts are used. For example, (9a) can hardly be paraphrased as (9c), a better paraphrase being (9d).

- (9) c. ?An aardvark belched, and it was in distress.
 d. An aardvark belched, as it was in distress.

It is enough, for our present purposes, however, that \wedge/Σ capture *part* of what's going on in (9ad), and that that part explain the presupposition filtering in (4). Similarly for \supset/Π and (5).

Now that we have a feel for Σ and Π , let us inquire more closely into the notion of dependence at work. If the variable x in $(\Sigma x:A)B$ or $(\Pi x:A)B$ is to mediate non-vacuous dependence, it must occur in B . This raises the question: just what is (say) an expression $\text{belched}(x)$ with a variable x ? As an expression, $\text{belched}(x)$ expresses a type only in a context where x has an appropriate type. Which leads to the next question: what is a context? A *context* Γ is a finite sequence of variable typings, on the basis of which variables and more complicated terms such as lz can be typed. Examples are provided by (13a) and (13b), where a *sequent* $\Gamma \Rightarrow t:A$ can be read “ t has type A in the context Γ ” or “ Γ supports the typing $t:A$.”

- (13) a. $x:\text{aardvark} \Rightarrow x:\text{aardvark}$
 b. $z:(\Sigma x:\text{aardvark}) \text{belched}(x) \Rightarrow lz:\text{aardvark}$
 c. $\frac{\Gamma \Rightarrow t:\text{aardvark}}{\Gamma \triangleright \text{belched}(t)}$ d. $\frac{\Gamma \Rightarrow t:\text{aardvark}}{\Gamma \triangleright \text{was-in-distress}(t)}$

(13a) is an example of the simplest way to use an assumption (i.e., as a conclusion), while (13b) is suggested by our understanding (8) of Σ . (13c) and (13d) are rules for inferring a sequent at the bottom (the denominator of, as it were, the fraction), given sequents at the top (the numerator). The sequents $\Gamma \triangleright A$ at the bottom of (13c) and (13d) can be read “ A expresses a proposition in the context of Γ ” or, as mentioned in the introduction, “ Γ satisfies the presuppositions of A ,” shortened in Karttunen 1974 to “ Γ admits A .”⁵ We will justify (13c) and (13d) in §2.2; for now, note that (13a) and (13c) yield (14), while (13b) and (13d) yield (15).

- (14) $x:\text{aardvark} \triangleright \text{belched}(x)$
 (15) $z:(\Sigma x:\text{aardvark}) \text{belched}(x) \triangleright \text{was-in-distress}(lz)$

To relate (14) and (15) to presupposition satisfaction for (7b), (12b), (9b) and (10b), it is helpful to introduce a further form of assertion, *A type*, saying that A is a well-defined collection. A concrete example is (16a); more general uses are made in (16b), which characterizes presupposition satisfaction of Σ and Π , and (16c), which takes sequents $\Gamma \Rightarrow A$ *type* to be generalizations of $\Gamma \triangleright A$ (as A is, on this assumption, the collection of its proofs).⁶

- (16) a. $\epsilon \Rightarrow \text{aardvark type}$ (where ϵ is the empty sequence)

$$\begin{array}{l}
\text{b. } \frac{\Gamma \Rightarrow A \text{ type} \quad \Gamma, x:A \triangleright B}{\Gamma \triangleright (\Sigma x:A)B} \quad \frac{\Gamma \Rightarrow A \text{ type} \quad \Gamma, x:A \triangleright B}{\Gamma \triangleright (\Pi x:A)B} \\
\text{c. } \frac{\Gamma \triangleright A}{\Gamma \Rightarrow A \text{ type}}
\end{array}$$

Henceforth, we suppress the empty sequence ϵ when it occurs to the left of \Rightarrow or \triangleright , and write, for instance, \Rightarrow **aardvark type** for (16a). Instantiating the premisses of (16b) by (16a) and (14), we get \triangleright (7b) and \triangleright (12b) (dropping ϵ). \triangleright (7b) and (16c) then give \Rightarrow (7b) **type**, which together with (15) and (16b) yield \triangleright (9b) and \triangleright (10b).

Before moving on, let us pause to consider a slight variant of (9), replacing the pronoun *it* in (9a) by a definite description.

$$\begin{array}{l}
(17) \quad \text{a. } \quad \underline{\text{An aardvark belched.}} \quad \underline{\text{The aardvark was in distress.}} \\
\quad \quad \text{b. } \quad (\Sigma z: \quad (7b)) \quad \{lz:\text{aardvark}\} \text{ was-in-distress}(lz) \\
\quad \quad \text{c. } \quad \quad \quad z:(7b) \quad \triangleright \{lz:\text{aardvark}\} \text{ was-in-distress}(lz)
\end{array}$$

(17b) differs from (9b) only in introducing an expression $\{lz:\text{aardvark}\}$ for the definite description in (17a). Assuming (17c), we can derive \triangleright (17b) from \triangleright (7b) and (16b). But how do we get (17c)? (17c) follows from (13b), (15) and the general rule⁷

$$\frac{\Gamma \Rightarrow t:A \quad \Gamma \triangleright B}{\Gamma \triangleright \{t:A\}B}$$

If (9) has the variant (17), (10) has the variant (18).

$$\begin{array}{l}
(18) \quad \text{a. } \quad \text{If } \underline{\text{an aardvark belched}} \quad \underline{\text{the aardvark was in distress}} \\
\quad \quad \text{b. } \quad (\Pi z: \quad (7b)) \quad \{lz:\text{aardvark}\} \text{ was-in-distress}(lz)
\end{array}$$

Comparing $\{t:A\}B$ with $(\Sigma x:A)B$, we have in t vs x the *familiar* vs *novel* opposition in Heim 1982 between definites and indefinites. A question we have been ignoring since discussing (9) re-emerges: where does lz in (9b), (10b), (17b) or (18b) come from? Recognizing the need for a process to pick out lz , let us introduce a *meta-variable* v in the translation (19b) of (19a) and (19d) of (19c), the idea being that anaphora resolution turns v into lz .

$$\begin{array}{l}
(19) \quad \text{a. } \quad \text{It was in distress.} \quad \quad \text{c. } \quad \text{The aardvark was in distress.} \\
\quad \quad \text{b. } \quad \text{was-in-distress}(v) \quad \quad \text{d. } \quad \{v:\text{aardvark}\} \text{ was-in-distress}(v)
\end{array}$$

(19a/c) may, of course, occur within a discourse where v cannot be resolved to a term, or where v can be resolved to more than one term. This raises very interesting complications, which we pass over in silence.⁸

2.2. Formal implementation

Next, we construe \Rightarrow and \triangleright formally, relative to a relational signature L specifying relation symbols with arities. The arity of a relation symbol R is the number of arguments R takes, each of which is understood (for simplicity)

to have the same sort U . U can be equated with Montague's entity type e , and would, under Tarskian semantics, get interpreted by an L -model M as the universe U_M over which first-order quantification ranges, with n -ary L -relation symbols R interpreted as n -ary relations $R_M \subseteq U_M^n$ on U_M . Having drawn the distinction between expressions and denotations, we will often shorten "relation symbol" to "relation," as in the description (20) of a particular choice of L to account for the examples in §2.1.

- (20) a. U is a type, $\Rightarrow_L U$ type
 b. belched, was-in-distress and aavk are unary L -relations

$$\frac{\Gamma \Rightarrow_L t:U}{\Gamma \triangleright_L \text{belched}(t)} \quad \frac{\Gamma \Rightarrow_L t:U}{\Gamma \triangleright_L \text{was-in-distress}(t)} \quad \frac{\Gamma \Rightarrow_L t:U}{\Gamma \triangleright_L \text{aavk}(t)}$$

Returning to (7), repeated below as (21ab), (20ab) combine with the rules (16bc) to give \triangleright_L (21c).⁹

- (21) a. An aardvark belched. d. $\exists x(\text{aavk}(x) \wedge \text{belched}(x))$
 b. $(\Sigma x:\text{aardvark})\text{belched}(x)$
 c. $(\Sigma x:U) \underbrace{(\Sigma _:\text{aavk}(x))\text{belched}(x)}_{\text{aavk}(x) \wedge \text{belched}(x)}$ e.

x
$\text{aavk}(x)$
$\text{belched}(x)$

(21c) translates to the first-order formula (21d) and in DRT to (21e). Expressions such as (21b) containing *aardvark* can, as we will see presently, be formed from subtypes $\{x:A|B\}$, where

$\{x:A|B\}$ is read "the set of x 's in A satisfying B ".

More precisely, we take $\{x:A|B\}$ to abbreviate $l[(\Sigma x:A)B]$, where l is the left projection described in (8), and lifted to types in (22).¹⁰

- (22) a. $l[C]$ is the image $\{lc \mid c:C\}$ of C under l

$$\frac{\Gamma \Rightarrow_L C \text{ type}}{\Gamma \Rightarrow_L l[C] \text{ type}} \quad \frac{\Gamma \Rightarrow_L t:C}{\Gamma \Rightarrow_L lt:l[C]}$$

- b. $\{x:A|B\}$ is a subset of A

$$\frac{\Gamma, x:A \Rightarrow_L B \text{ type}}{\Gamma, y:\{x:A|B\} \Rightarrow_L y:A} \quad y \text{ not in } \Gamma$$

Thus, if *aardvark* is literally the subtype $\{x:U \mid \text{aavk}(x)\}$, then (22b) gives $x:\text{aardvark} \Rightarrow_L x:U$, which couples with (20) to produce (13cd). In general, a unary relation P induces the subtype $\{x:U \mid P(x)\}$.

Now, let us compare the characterizations (8) and (11) of Σ and Π with the classical accounts (2a) and (3a) of some and all. Recalling that, under propositions-as-types, a proposition A is true exactly if it is non-empty

(i.e., it has a proof), observe that

$$\begin{aligned} (\Sigma x:A)B \text{ is true} & \text{ iff } A \cap B \neq \emptyset \\ (\Pi x:A)B \text{ is true} & \text{ iff } A \subseteq B . \end{aligned}$$

Indeed, it is not even clear how to understand the right hand sides above, with the variable x missing. (22) provides the ingredients for fixing the equivalences: add x to the right hand sides, as in (23).

$$(23) \quad \begin{aligned} \text{a. } (\Sigma x:A)B \text{ is true} & \text{ iff } A \cap \{x:A|B\} \neq \emptyset \\ \text{b. } (\Pi x:A)B \text{ is true} & \text{ iff } A \subseteq \{x:A|B\} \end{aligned}$$

But just what are we to make of the equivalences in (23)? Let us start by restoring the distinction between expressions and denotations. In order to interpret say, $(\Sigma x:\text{aardvark}) \text{ belched}(x)$ as a collection of ordered pairs (in accordance with (8)), let us define an *L-proof-interpretation* $\llbracket \cdot \rrbracket$ to specify a set $\llbracket \mathbf{U} \rrbracket$ and for every n -ary *L*-relation symbol R and $(a_1 \dots a_n) \in \llbracket \mathbf{U} \rrbracket^n$, a set $\llbracket R, a_1 \dots a_n \rrbracket$ of what we will regard as proofs of $R(a_1 \dots a_n)$. (The use here of \in reflects the assumption that our meta-theory is some classical set theory, as is customary in model theory.¹¹) Exactly what the elements of $\llbracket R, a_1 \dots a_n \rrbracket$ are need not concern us any more than how to choose an *L*-model, any of which, M , induces an *L*-proof-interpretation $\mathcal{P}(M) = \llbracket \cdot \rrbracket$ with $\llbracket \mathbf{U} \rrbracket = \mathbf{U}_M$ and

$$\llbracket R, a_1 \dots a_n \rrbracket = \begin{cases} \{(R, a_1 \dots a_n)\} & \text{if } R_M(a_1 \dots a_n) \\ \emptyset & \text{otherwise.} \end{cases}$$

Conversely, an *L*-proof-interpretation $\llbracket \cdot \rrbracket$ reduces to the *L*-model $\mathcal{M}[\llbracket \cdot \rrbracket] = M$ where $\mathbf{U}_M = \llbracket \mathbf{U} \rrbracket$ and

$$R_M = \{(a_1 \dots a_n) \in \llbracket \mathbf{U} \rrbracket^n \mid \llbracket R, a_1 \dots a_n \rrbracket \neq \emptyset\} .$$

Clearly, $\mathcal{M}(\mathcal{P}(M)) = M$ although it is not always the case that $\mathcal{P}(\mathcal{M}[\llbracket \cdot \rrbracket]) = \llbracket \cdot \rrbracket$. Be that as it may, the point is that an *L*-interpretation provides the basis for an interpretation $\llbracket \cdot \rrbracket_1$ of first-order formulas, with, for instance,

$$\llbracket \forall x.(P(x) \supset \exists y.R(x, y)) \rrbracket_1 = (\Pi x:\llbracket \mathbf{U} \rrbracket) (\Pi _:\llbracket P, x \rrbracket) (\Sigma y:\llbracket \mathbf{U} \rrbracket) \llbracket R, x, y \rrbracket$$

or, for the reader familiar with DRT, an interpretation $\llbracket \cdot \rrbracket_2$, with

$$\llbracket \frac{x}{P(x)} \Rightarrow \frac{}{P(x)} \rrbracket_2 = (\Pi z:(\Sigma x:\llbracket \mathbf{U} \rrbracket) \llbracket P, x \rrbracket) \llbracket P, lz \rrbracket .$$

These interpretations can be obtained from translations $\rightsquigarrow_1, \rightsquigarrow_2$ to dependent type expressions over *L*

$$\begin{aligned} \forall x.(P(x) \supset \exists y.R(x, y)) & \rightsquigarrow_1 (\Pi x:\mathbf{U}) (\Pi _:\llbracket P(x) \rrbracket) (\Sigma y:\mathbf{U}) R(x, y) \\ \frac{x}{P(x)} \Rightarrow \frac{}{P(x)} & \rightsquigarrow_2 (\Pi z:(\Sigma x:\mathbf{U}) \llbracket P(x) \rrbracket) \llbracket P(lz) \rrbracket \end{aligned}$$

which are then fed into an interpretation $\llbracket \cdot \rrbracket'$ with

$$\begin{aligned} [(\Pi x:U)(\Pi _ : P(x))(\Sigma y:U)R(x,y)]' &= (\Pi x:[U])(\Pi _ : [P,x])(\Sigma y:[U])[R,x,y] \\ [(\Pi z:(\Sigma x:U)P(x))P(lz)]' &= (\Pi z:(\Sigma x:[U])[P,x])[P,lz] . \end{aligned}$$

That any L -proof-interpretation (from set theory) *can* be extended to interpret dependent types is clear from, for instance, Feferman 1975. (Details in Fernando 2001.) Returning to (23), the claim made there is that

$$\begin{aligned} [(\Sigma x:A)B]' \neq \emptyset &\text{ iff } [A]' \cap [\{x:A|B\}]' \neq \emptyset \\ [(\Pi x:A)B]' \neq \emptyset &\text{ iff } [A]' \subseteq [\{x:A|B\}]' \end{aligned}$$

for every L -proof-interpretation $[\cdot]$, extended canonically to $[\cdot]'$.

So what? What do we gain by piling a layer of language on to (23)? The pay-off lies in the possibility of expanding L with generalized quantifiers along the same lines as in the *weak logic* $L(Q)$ of Keisler 1970, where an L -model M is paired with a set-theoretic interpretation of a generalized quantifier symbol Q to form an $L(Q)$ -model. Keisler exploits the variety in $L(Q)$ -models to prove a completeness theorem for $L(Q)$, providing the basis for an axiomatic approach to Q . Now, turning to dependent types, the idea is to expand an L -proof-interpretation $[\cdot]$ to Q , by equipping $[\cdot]$ with a relational interpretation Q_r , much like any binary L -relation R is interpreted, except that the arguments of Q_r are subtypes of $[U]$. In fact, we will extend these arguments in section 3. But that can wait. The point to appreciate now is that (23) is positive evidence for the feasibility of the whole enterprise, establishing, as it does, the compatibility of dependent type interpretations of $Q = \exists$ and $Q = \forall$ with relational interpretations. Having noted the usefulness of the notions of a signature L and an L -proof-interpretation $[\cdot]$, we need not always mention them. Accordingly, when the choice of L and/or $[\cdot]$ does not matter, we will drop them (stripping, for instance, L from $\Rightarrow_L/\triangleright_L$, and $[\cdot]$ from $[A]$).

Getting down to details, how should a (binary) quantifier symbol Q feed into the dependent type expressions described above? On the assumption that smooth generalizations of Σ and Π are desirable, recall our rule (16b) for Σ and Π , repeated below.

$$(16) \quad \text{b.} \quad \frac{\Gamma \Rightarrow A \text{ type} \quad \Gamma, x:A \triangleright B}{\Gamma \triangleright (\Sigma x:A)B} \quad \frac{\Gamma \Rightarrow A \text{ type} \quad \Gamma, x:A \triangleright B}{\Gamma \triangleright (\Pi x:A)B}$$

(16b) is not quite in line with Karttunen's admittance account, formulated in the introduction as (6).

$$(6) \quad \Gamma \triangleright (Qx:A)B \text{ iff } \Gamma \triangleright A \text{ and } \Gamma, x:A \triangleright B$$

The sequent $\Gamma \triangleright A$ in (6) is weakened in (16b) to the premiss $\Gamma \Rightarrow A$ **type**. To say that A is a well-defined collection in Γ , $\Gamma \triangleright A$, falls short of saying that A makes a claim in Γ , $\Gamma \triangleright A$. A concrete example is the derivation of $\triangleright (\Sigma x:U) \text{ aavk}(x)$ [i.e., $\exists x \text{ aavk}(x)$] from $\Rightarrow U$ **type** and $x:U \triangleright \text{aavk}(x)$, leading (with the help of (16c) and (22a)) to $\Rightarrow \{x:U|\text{aavk}(x)\}$ **type**. In ordinary predicate logic, neither U nor $\{x:U|\text{aavk}(x)\}$ would qualify as a for-

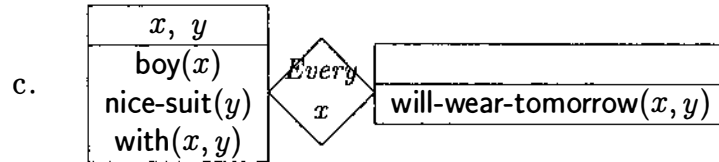
mula. A slightly different example, moving to verbal predicates, comes back to a point made in §1.2. The verb *bite* in *some ants bite* ought not to be interpreted too quickly as the set B of biters, as in (2), but rather as an expression $\text{bite}(x)$, the presuppositions of which are satisfied by any context Γ that supports the typing $x : U$. There is a variable x here worth recognizing before abstracting away. Behind the scenes lurk proofs $\in \llbracket \text{bite}, a \rrbracket$ for $a \in \llbracket U \rrbracket$ such that a bites in $\llbracket \cdot \rrbracket$ — proofs that get discarded when forming $\{x : U \mid \text{bite}(x)\}$.

As for (6) versus (16b), we will, nevertheless, see next a sense in which (6) will do for generalized quantifiers Q that more closely model English.

3. Quantification

In section 2, dependent types were employed to endow first-order formulas with anaphoric power. That said, whenever possible, let us adopt suggestive notation, writing \wedge or \exists instead of Σ , and \supset or \forall instead of Π . As hinted at the end of §2.2, it will be useful to refine \exists and \forall to model English sentences. An example is (24a), the translation (24b) of which can be pretty-printed as the DRT duplex condition (24c).¹²

- (24) a. Every boy with a nice suit will wear it tomorrow.
 b. $(\text{Every } z : C)D$ where C is $\exists x \exists y (\text{boy}(x) \wedge \text{nice-suit}(y) \wedge \text{with}(x, y))$, and D is $\text{will-wear-tomorrow}(lz, l(rz))$.



(24a) brings out a well-known complication (e.g. Kamp and Reyle 1993, Chierchia 1995); *every* binds *boy* without universally quantifying *suits*. Lest we assert that every nice suit owned by a boy be worn tomorrow, we had better not equate (24b) with $(\Pi z : C)D$. To pick out the quantified variable x (inside the diamond in (24c)), let us apply the left projection l ; by (22),

$$l[C] \equiv \{x : \text{boy} \mid (\exists y : \text{nice-suit}) \text{with}(x, y)\}$$

$$l[\{z : C \mid D\}] \equiv \{x : \text{boy} \mid (\exists y : \text{nice-suit})(\text{with}(x, y) \wedge \text{will-wear-tomorrow}(x, y))\}$$

where (extensional equivalence) \equiv holds between types that have the same objects, and $\underline{P} = \{x : U \mid P(x)\}$ for unary relations P . This leads to (25a), modifying the picture of all in (23b) (which, in turn, amends (3a)).

- (25) a. $(\text{Every } z : C)D$ iff $l[C] \subseteq l[\{z : C \mid D\}]$
 b. $(Qz : A)B$ iff $Q_r(l[A], l[\{z : A \mid B\}])$

(25b) generalizes (25a) to a quantifier Q with a binary relation Q_r which, under the set-theoretic conception of Q (in (1),(2) and (3)), interprets Q . In the present set-up, an equivalence ' T iff S ' between types T and S (as in (25ab)) means ' $\llbracket T \rrbracket \neq \emptyset$ iff $\llbracket S \rrbracket \neq \emptyset$ for all interpretations $\llbracket \cdot \rrbracket$ ' (see §2.2). A full-

the 'tuple of its participants; and these participants are the values of the variables that occur free in the open sentence modified by the adverb. [Lewis 1975, p.7]

- (28) Northerners seldom recognize southerners.
 'Few pairs of northerners n and southerners s are such that n recognizes s .'

After outlining a simple theory of case quantification, Lewis concedes that

matters are not quite that simple. . . . adverbs of quantification are not entirely unselective . . . we might prefer to treat the modified sentence as if it contained an event-variable (or even posit such a variable in underlying structure) and include an event-coordinate in the cases.

. . . not just any 'tuple of values of the free variables, plus perhaps a time- or event-coordinate, will be admissible as one of the cases quantified over. Various restrictions may be in force . . . [p.8]

The remainder of §3.1 is an attempt to work out these ideas in the present set-up.

We start by building up Lewis' admissible tuples to *generalized quantificational domains* (gqd's), focusing on the special case (29d) of the admittance/formation rule (6)'.¹³

- (29) a. $\Gamma \Rightarrow U$ gqd for every context Γ
 b.
$$\frac{\Gamma \Rightarrow T \text{ gqd} \quad \Gamma, x:T \Rightarrow T' \text{ gqd}}{\Gamma \Rightarrow (\exists x:T)T' \text{ gqd}}$$
 c.
$$\frac{\Gamma \Rightarrow T \text{ gqd}}{\Gamma \Rightarrow T \text{ type}}$$

 d.
$$\frac{\Gamma \Rightarrow T \text{ gqd} \quad \Gamma, x:T \triangleright A \quad \Gamma, z:(\exists x:T)A \triangleright B}{\Gamma \triangleright (Qz:(\exists x:T)A)B}$$

(29ab) throws U in among the gqd's, and closes them under tupling. (29c) says gqd's are types. In accordance with the intuition that $\Gamma \Rightarrow T$ gqd should *not* be conflated with $\Gamma \triangleright T$, we write \exists in (29bd) for Σ . To allow for dependencies which might become useful if further closure conditions on gqd's are introduced, we stop short in (29b) of writing Cartesian multiplication \times . For the present, however, \exists in (29b) can be simplified to \times . For example, (28) above translates to (30).

- (30) $(\text{Few } z : (\exists x : (U \times U)A)) \text{ recognize}(l(lz), r(lz))$ where A is $\text{northerner}(lx) \wedge \text{southerner}(rx)$.

The gloss in (28) can then be recovered by applying (25b) to (30). Identifying Lewis' admissibility conditions with A in (29d) and (30), notice that (29) goes

beyond Lewis 1975 in allowing A to quantify some objects away from the quantificational force of Q/few .

$$\begin{array}{l} \text{case} \\ \overbrace{(\exists x : (\mathbf{U} \times \mathbf{U})) R(lx, rx)} \quad - \quad \text{unselective/Lewis} \quad - \quad \text{e.g. (28)} \\ (\exists x : \mathbf{U}) \underbrace{(\exists y : \mathbf{U}) R(x, y)} \quad - \quad \text{selective/nominal} \end{array}$$

admissibility condition A

According to many (e.g. Peters and Westerståhl (to appear)), the latter selective reading fits nominal quantifications such as *Few northerners recognize southerners* better than (28). Clearly, we must impose suitable constraints on applications of (29d), if we are to rule out a selective reading of (28). But then, *unselective* readings are not always appropriate for adverbs either, as S. Berman's sentence (31) shows (Dekker 1997).

- (31) If a letter arrives for me, I am usually at home.
'For most times a letter arrives for me I am at home.'

For (29d) to capture (31), the admissibility condition A must internally bind the *letter*-variable (keeping it from the reach of *most*).

Indeed, selection complications infect already simple adverbial modifications, as in (32bc), where different sides of a proof z of the translation B of (32a) are selected.

- (32) a. Aldo belched \rightsquigarrow $\underbrace{(\exists t : \text{PastTime}) \text{belch}(\text{aldo}, t)}_B$
 b. Aldo belched yesterday \rightsquigarrow $(\Sigma z : B) \text{yesterday}(lz)$
 c. Aldo belched politely \rightsquigarrow $(\Sigma z : B) \text{polite}(rz)$

There is no question that translations indicated in (32) by \rightsquigarrow are problems which cannot be solved merely by studying the logical forms \rightsquigarrow points to.¹⁴ Thus, while (29) can hardly be the full story, that is no reason to reject it.

What can we say about labels such as rz in (32c) for $\text{belch}(\text{aldo}, lz)$? It is worth quoting Troelstra and Schwichtenberg 2000.

when are two proofs to be considered equal? We doubt ... this question will ever have a simple answer; it may well be that there are many answers, depending on aims and points of view. (p.xii)

Mysterious though their identity conditions may be, proofs can nevertheless be studied and used. And for the present linguistic applications, I suggest that a proposition $(\exists x : T)A$ formed from a *gqd* T (as in 29d) induces the following notion of identity on proofs:

$$z = z' \text{ mod } (\exists x : T)A \quad \text{iff} \quad z : (\exists x : T)A \text{ and } z' : (\exists x : T)A \text{ and } lz = lz' .$$

That is, proofs z and z' of $(\exists x:T)A$ are compared based on their left projections, these being tuples of objects with presumably non-problematic identity conditions. Accordingly, quantification in (25b) selects left components (giving us one half of the slogan in the introduction: objects are for quantification and proofs for anaphora). Precisely what the identity conditions on rz in (32c) are is sidestepped.

Turning now to Lewis' invocation of events, I propose (33), which falls *just* short of identifying proofs with eventualities.

- (33) *Constructive eventuality assumption* (CEA): associate eventualities with proofs, extracting objects for quantification.

For CEA to be a theory of eventualities, something must be said about the propositions $(\exists x:T)A$ defining eventuality-types, the instances of which form the left projection $\{x:T|A\}$. Leaving a development of CEA for some other occasion, I ought perhaps to point out that CEA is not so much an empirical claim as a methodological assumption. It can be satisfied trivially by reifying eventualities into T . The intent in CEA, however, is to base notions of eventuality also on the *right* side of $(\exists x:T)A$, as hinted by (32c).

3.2. Some refinements

Next, we extend (29) by (29e) and (29f), to consider various readings of (34), analyzed at length in Kanazawa 1994b, Chierchia 1995 and Krifka 1996.

$$(29) \quad \begin{array}{l} \text{e.} \quad \frac{\Gamma \Rightarrow T \text{ gqd} \quad \Gamma, x:T \triangleright A}{\Gamma \Rightarrow \{x:T|A\} \text{ gqd}} \\ \text{f.} \quad \frac{\Gamma \Rightarrow T \text{ gqd} \quad \Gamma, x:T \triangleright A \quad \Gamma, x:T, y:A \triangleright B}{\Gamma \triangleright (Qx:T, Q'y:A)B} \end{array}$$

- (34) Q farmer who owns a donkey beats it.
 $(Qx:\underline{\text{farmer}}, Q'y:(\exists v:\underline{\text{donkey}})\text{owns}(x, v)) \text{beats}(x, ly)$

Recalling that $\underline{P} = \{x : \cup P(x)\}$, (29e) is used to apply (29f) to (34), the point behind the double quantification $(Qx:T, Q'y:A)B$ being to separate the Q -quantified ‘‘farmer’’ variable x from the antecedent/indefinite ‘‘donkeys’’ in y , which are quantified (in the ‘scope’ B) by Q' . The precise semantics is (35).

$$(35) \quad (Qx:T, Q'y:A)B \text{ iff } Q_r(\{x:T|A\}, \{x:T|Q'_r(A, \{y:A|B\})\})$$

An immediate consequence of (35) is (36).

$$(36) \quad (Qx:T, \exists y:A)B \text{ iff } (Qz:(\exists x:T)A)B[x/lz, y/rz]$$

To match choices of Q' other than the weak reading \exists , let us build Q' into (25b) as shown in (37a), which uses (37b).

$$(37) \quad \begin{array}{l} \text{a.} \quad (Q_{Q'}z:C)D \text{ iff } Q_r(l[C], l[\{z:C|D\} -_{Q'} C]) \\ \text{b.} \quad C' -_{Q'} C = \{z:C' \mid Q'_r(\{y|\langle lz, y \rangle:C\}, \{y|\langle lz, y \rangle:C'\})\} \end{array}$$

(Alternatively, one might try expressing $(Qx:T, Q'y:A)B$ by iterations such

as $(Qx:T)(Q'y:A)B$; good luck!) Clearly, if $C' \subseteq C$ then $C' -_{\exists} C = C'$, from which (38) follows.

$$(38) \quad (Qz:C)D \text{ iff } (Q_{\exists z:C})D$$

Also, if C is $(\exists x:T)A$ and D is $B[x/lz, y/rz]$, then

$$\{z:C|D\} -_{Q'} C \equiv \{z:C|D \wedge Q'_r(A[x/lz], \{y:A[x/lz]|B[x/lz]\})\}$$

(where \equiv is extensional equivalence). Consequently, we have (39).

$$(39) \quad \text{For conservative } Q_r, \\ (Qx:T, Q'y:A)B \text{ iff } (Q_{Q'z}:(\exists x:T)A) B[x/lz, y/rz] .$$

To meet the assumption of conservativity in (39), observe the following.

$$(40) \quad \text{Given } Q, \text{ define } \tilde{Q} \text{ to have the relational interpretation}$$

$$\tilde{Q}_r(S, S') \text{ iff } Q_r(S, S \cap S') .$$

$$\text{Then for all } Q', (Q_{Q'}x:A)B \text{ iff } (\tilde{Q}_{Q'}x:A)B .$$

(40) says that in an expression $(Q_{Q'}x:A)B$, we can assume without loss of generality that Q_r is conservative (replacing, if necessary, Q by \tilde{Q}).

Having reformulated $(Qx:A)B$ and $(Qx:T, Q'y:A)B$ in terms of $Q_{Q'}$, what can we say now about existential vs universal donkeys? I am not sure. But rather than arguing (as Kanazawa 1994b and Chierchia 1995 do) that some dynamic version of conservativity be met, let us apply context change to explain conservativity in its static splendor. To simplify notation, we revert to $(Qx:A)B$, leaving the refinement $(Q_{Q'}x:A)B$ for the hardy reader to check. (Q' only clutters things up.)

3.3. Explaining conservativity

Inasmuch as the dependent quantifier $(Qx:A)B$ applies Q_r only to pairs $\langle S, S' \rangle$ such that $S' \subseteq S$, we can assume that Q_r is conservative (appealing to \tilde{Q} from §3.2). But surely it is not enough to say that Q_r is conservative because we *can assume* that it is. We could just as easily assume it is not! The relation $\{\langle S, S \cap S' \rangle | Q_r(S, S')\}$ is not in general conservative, but (like \tilde{Q}_r) induces a dependent quantifier truth-conditionally equivalent to $(Qx:A)B$. The blunt fact is that $(Qx:A)B$ is indifferent to what Q_r makes of pairs $\langle S, S' \rangle$ with $S' \not\subseteq S$. And to test Q_r 's conservativity, we need to apply Q_r to such pairs. Accordingly, consider (41), where \underline{P} is understood to be $\{x:U|P(x)\}$.

$$(41) \quad \begin{array}{ll} \text{a. } Q \text{ ants bite.} \\ \text{b. } Q_r(\underline{\text{ant}}, \underline{\text{bite}}) \\ \text{c. } Q_r(\underline{\text{ant}}, \{x:\underline{\text{ant}}|\underline{\text{bite}}(x)\}) & [\text{iff } Q_r(\underline{\text{ant}}, \underline{\text{ant}} \cap \underline{\text{bite}})] \end{array}$$

(41b) represents the truth conditions of (41a) under a traditional analysis, whereas (41c) comes from the present dependent quantifier interpretation. We can now explain conservativity.

The validity of the traditional analysis (41b) rests on its equivalence with that of dependent quantifiers (41c). *That equivalence is what it means for Q_r to be conservative.*

The twist in this simple explanation may well be hard to take. I have tried to prepare the ground for it by linking (41c) to presupposition and anaphora. Doing so is not altogether straightforward, as the failure of $(Qx : \underline{\text{ant}}) \text{bite}(x)$ to express (41c) attests.¹⁵

4. Discussion: updating Montague

Put in very general terms, the present work is an attempt to introduce a proof-theoretic interpretation of type theory to Montague semantics, while retaining model-theoretic interpretations. To connect presupposition satisfaction $\Gamma \triangleright A$ with truth, it is convenient to rewrite $\Gamma \triangleright A$ as $\Gamma \Rightarrow A$ wff (in accordance with the intuition that ‘ A ’s presuppositions are satisfied by Γ ’ precisely if ‘ A is a well-formed formula under Γ ’).¹⁶ The idea then is to boost Montague’s truth values t to wff, and to build generalized quantificational domains (gqd’s) from a universe U (Montague’s e), using Σ and l (rather than functions and λ ’s, which are implicit in the background signature L). Wff’s and gqd’s are all types, and, as such, are subject to the same rules on typings $t : A$, relative to a context Γ . That is to say, presupposition satisfaction and conservativity are subject to a common mechanism of context dependence/change. A context here is a sequence of variable-type declarations, familiar in typed functional programming languages. This makes the set-up attractive from a computational point of view. And, taking care to account for previous insights, perhaps also from a theoretical perspective.

Tim.Fernando@tcd.ie

Endnotes

*Talks on the work reported here were (in its early stages) given in London, Amsterdam and Dublin, not to mention New York. My thanks to the audiences there, including Paul Dekker for generously sharing some of his unpublished notes on the subject.

¹“Propositions-as-types” is referred to as *formulas-as-types* in BPT (Troelstra and Schwichtenberg 2000), and formulated elsewhere as the *Curry-Howard isomorphism* between a typed λ -calculus and a natural deduction system. The names Martin-Löf and de Bruijn (not to mention Girard) are also noteworthy in this connection; references can be found in BPT and Ranta 1994.

²The point of both (4) and (5) is that neither presupposes Buganda has a king. Although, in isolation, their second clause, *the king of Buganda is bald*, presupposes the existence of a Bugandan king, this presupposition is discharged by their first clause, *Buganda has a king*. (The presupposition that there is a thing called Buganda remains.)

³I am, for the sake of convenience, being a bit loose here with the distinction between syntactic expressions and their semantic denotations. In particular, the presence of the variable x in $\text{belched}(x)$ points to the need to relativize the notion of a type to a context that assigns variables such as x a type. Logical hygiene will come later.

⁴On co-indexing an anaphoric term with an antecedent x introduced by a dynamic existential quantifier $\mathcal{E}x$, Kanazawa 1994a is careful to point out that “the ‘binding’ relation between x and $\mathcal{E}x$ is an indirect one mediated by ; or \Rightarrow ” (page 224, footnote 10). By combining $\mathcal{E}x$ with ; for Σx , and $\mathcal{E}x$ with \Rightarrow for Πx (described in (11)), the proof-conditional approach encodes into its anaphoric terms the inferences required for binding — inferences that can be constrained or expanded as the user sees fit, by controlling term/proof construction. A detailed comparison between first-order fragments of dynamic and proof-conditional semantics is carried out in Fernando 2001.

⁵The use of the symbol \triangleright follows Beaver 1997.

⁶We would not go too far wrong were we to conflate the sequents $\Gamma \Rightarrow A$ **type** and $\Gamma \triangleright A$. Until, that is, the distinction is played up in section 3.

⁷I am indebted to Peter Krause (p.c.) for suggesting this treatment of the presupposition connective (see Beaver 1997 for some background). By contrast, in Fernando (to appear), the construct $\{\cdot\}B$ is analyzed before type evaluation, at which point the conditions \cdot in $\{\cdot\}B$ are assumed to be satisfied (and erased). The idea there is to borrow the distinction from programming language practice between compilation and execution, treating $\{\cdot\}$ as comments checked during compilation, and ignored at run-time. The formulation suggested by Krause does away with this distinction.

⁸The interested reader is referred to van der Sandt 1992, Beaver 1997, and, for a paper linking accommodation to leaps studied in proof theory, as well as a modal approach to ambiguity, Fernando (to appear).

⁹To demonstrate this rigorously, it helps to have $x : U \Rightarrow_L x : U$ and $x : U, y : \text{aavk}(x) \Rightarrow_L x : U$, both of which follow from (20), (16c) and the three rules

$$\frac{}{\epsilon \text{ context}} \quad \frac{\Gamma \Rightarrow A \text{ type}}{\Gamma, x : A \text{ context}} \quad x \text{ not in } \Gamma \quad \frac{\Gamma \text{ context}}{\Gamma \Rightarrow x : A} \quad 'x : A' \text{ in } \Gamma .$$

¹⁰Subtypes are a delicate matter in the propositions-as-types literature, the simple-minded treatment in (22) departing from the tenet that the type of an object is unique. For applications to linguistics, however, I am not aware

of any compelling reason to complicate (22) or reject the set-theoretic foundations nearly universally taken for granted in, for instance, model-theoretic semantics. Given the work on generalized quantifiers in linguistics (Keenan and Westerståhl 1997), surely we can be forgiven for being as conservative as possible in introducing useful proof-theoretic ideas to the set-theoretic conception of generalized quantifiers. In particular, as admirable as Sundholm 1989 might be, I am inclined to build on the weak completeness theorem in Keisler 1970 and the considerable body of work that has followed it.

¹¹See the preceding endnote.

¹²The reader unfamiliar with DRT can ignore the references to it, but may nevertheless find the boxes and diamonds useful for orientation. Speaking of which, the adequacy of (6) claimed at the end of §2.2 is not surprising from the perspective of Kamp and Reyle 1993 inasmuch as the treatment there of quantifiers respects (6).

¹³To justify (29d) on the basis of (6), it helps to have also (29c) and (16b). Also, (29ac) and ϵ context (from endnote 9) give (20a).

¹⁴Of the “three processes in natural language interpretation” referred to in the title of Fernando (to appear), translation \rightsquigarrow is the *central* process, and must (among other things) address the problem of anaphora resolution broached at the end of §2.1 above.

¹⁵Section 3 is all about the pressures to express (41c) instead through $(Qz : (\exists x : U)\text{ant}(x)) \text{bite}(lz)$. Handier notations are, of course, available. For user-friendly co-indexing, DRT boxes and diamonds are hard to beat. But the user unafraid of proof terms may wish to try out the inferential scope l, r and the λ -calculus afford. Fernando (to appear) develops the theory of presupposition with that in mind.

¹⁶I have opted for the labels ‘wff’ and ‘gqd’ rather than ‘proposition’ and ‘set’ to recognize the formal character of the notions at stake, which are understood to be relativized to a relational signature L (§2.2).

References

- David Beaver. Presupposition. In J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*. Elsevier, 1997.
- Gennaro Chierchia. *Dynamics of Meaning*. University of Chicago Press, 1995.
- Paul Dekker. Cases, adverbs, situations and events. In H. Kamp and B. Partee (eds.), *Proc. Workshop on Context Dependence*. Stuttgart, 1997.
- Solomon Feferman. A language and axioms for explicit mathematics. In J.N. Crossley (ed.), *Algebra and Logic*, LNM 450. Springer, Berlin, 1975.
- Tim Fernando. A type reduction from proof-conditional to dynamic semantics. *Journal of Philosophical Logic*, 30(2):121–153, 2001.
- Tim Fernando. Three processes in natural language interpretation. In *Fest-*

- schrift for Solomon Feferman*. To appear (ASL Lecture Notes series).
- Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. Dissertation, University of Massachusetts, Amherst, 1982. Published by Garland Press, New York, 1988.
- Irene Heim. On the projection problem for presuppositions: In M. Barlow, D. Flickinger, and M. Westcoat (eds.), *Proc. West Coast Conference on Formal Linguistics*, volume 2. Stanford, 1983.
- H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.
- Makoto Kanazawa. Dynamic generalized quantifiers and monotonicity. In M. Kanazawa and C. Piñón (eds.), *Dynamics, Polarity and Quantification*. CSLI Lecture Notes No. 48, Stanford, 1994a.
- Makoto Kanazawa. Weak vs. strong readings of donkey sentences and monotonicity inference in a dynamic setting. *Linguistics and Philosophy*, 27(2):109–158, 1994b.
- Lauri Karttunen. Presupposition and linguistic context. *Theoretical Linguistics*, pages 181–194, 1974.
- E. Keenan and J. Stavi. A semantic characterization of natural language determiners. *Linguistics and Philosophy*, 9(3):253–326, 1986.
- E. Keenan and D. Westerståhl. Generalized quantifiers in linguistics and logic. In J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*. Elsevier, 1997.
- H. Jerome Keisler. Logic with the quantifier ‘there exist uncountably many’. *Annals of Mathematical Logic*, 1:1–93, 1970.
- E. Krahmer and P. Piwek. Presupposition projection as proof construction. In H. Bunt and R. Muskens (eds.), *Computing Meaning*, vol 1. Kluwer, Dordrecht, 1999.
- Manfred Krifka. Pragmatic strengthening in plural predications and donkey sentences. In *Proc. Semantics and Linguistic Theory VI*. Cornell, 1996.
- David Lewis. Adverbs of quantification. In E. Keenan (ed.), *Formal Semantics of Natural Language*. Cambridge University Press, 1975.
- S. Peters and D. Westerståhl. Does English really have resumptive quantification and do ‘donkey’ sentences really express it? In D. Beaver and S. Kaufmann (eds.), *Proc. ‘Semfest’*. CSLI, Stanford, to appear.
- Aarne Ranta. *Type-Theoretical Grammar*. Oxford University Press, 1994.
- Rob A. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333–377, 1992.
- Göran Sundholm. Proof theory and meaning. In D. Gabbay and F. Guenther (eds.), *Handbook of Philosophical Logic*, vol 3. Reidel, Dordrecht, 1986.
- Göran Sundholm. Constructive generalised quantifiers. *Synthese*, 79, 1989.
- A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory* (Second edition). Cambridge University Press, 2000.