

System with Generalized Quantifiers on Dependent Types for Anaphora

Justyna Grudzińska

Institute of Philosophy
University of Warsaw

Krakowskie Przedmieście 3, 00-927 Warszawa
j.grudzinska@uw.edu.pl

Marek Zawadowski

Institute of Mathematics
University of Warsaw

Banacha 2, 02-097 Warszawa
zawado@mimuw.edu.pl

Abstract

We propose a system for the interpretation of anaphoric relationships between unbound pronouns and quantifiers. The main technical contribution of our proposal consists in combining generalized quantifiers with dependent types. Empirically, our system allows a uniform treatment of all types of unbound anaphora, including the notoriously difficult cases such as quantificational subordination, cumulative and branching continuations, and donkey anaphora.

1 Introduction

The phenomenon of unbound anaphora refers to instances where anaphoric pronouns occur outside the syntactic scopes (i.e. the c-command domain) of their quantifier antecedents. The main kinds of unbound anaphora are regular anaphora to quantifiers, quantificational subordination, and donkey anaphora, as exemplified by (1) to (3) respectively:

- (1) Most kids entered. They looked happy.
- (2) Every man loves a woman. They kiss them.
- (3) Every farmer who owns a donkey beats it.

Unbound anaphoric pronouns have been dealt with in two main semantic paradigms: dynamic semantic theories (Groenendijk and Stokhof, 1991); (Van den Berg, 1996); (Nouwen, 2003) and the E-type/D-type tradition (Evans, 1977); (Heim, 1990); (Elbourne, 2005). In the dynamic semantic theories pronouns are taken to be (syntactically free, but semantically bound) variables, and context serves as a medium supplying values for the variables. In the E-type/D-type tradition pronouns are treated as quantifiers. Our system combines aspects of both families of theories. As in the E-type/D-type tradition we treat unbound anaphoric

pronouns as quantifiers; as in the systems of dynamic semantics context is used as a medium supplying (possibly dependent) types as their potential quantificational domains. Like Dekker’s Predicate Logic with Anaphora and more recent multidimensional models (Dekker, 1994); (Dekker, 2008), our system lends itself to the compositional treatment of unbound anaphora, while keeping a classical, static notion of truth. The main novelty of our proposal consists in combining generalized quantifiers (Mostowski, 1957); (Lindström, 1966); (Barwise and Cooper, 1981) with dependent types (Martin-Löf, 1972); (Ranta, 1994).

The paper is organized as follows. In Section 2 we introduce informally the main features of our system. Section 3 sketches the process of English-to-formal language translation. Finally, sections 4 and 5 define the syntax and semantics of the system.

2 Elements of system

2.1 Context, types and dependent types

The variables of our system are always typed. We write $x : X$ to denote that the variable x is of type X and refer to this as a type specification of the variable x . Types, in this paper, are interpreted as sets. We write the interpretation of the type X as $\|X\|$.

Types can depend on variables of other types. Thus, if we already have a type specification $x : X$, then we can also have type $Y(x)$ depending on the variable x and we can declare a variable y of type Y by stating $y : Y(x)$. The fact that Y depends on X is modeled as a projection

$$\pi : \|Y\| \rightarrow \|X\|.$$

So that if the variable x of type X is interpreted as an element $a \in \|X\|$, $\|Y\|(a)$ is interpreted as the fiber of π over a (the preimage of $\{a\}$ under π)

$$\|Y\|(a) = \{b \in \|Y\| : \pi(b) = a\}.$$

One standard natural language example of such a dependence of types is that if m is a variable of the type of months M , there is a type $D(m)$ of the days of the month m . Such type dependencies can be nested, i.e., we can have a sequence of type specifications of the (individual) variables:

$$x : X, y : Y(x), z : Z(x, y).$$

Context for us is a partially ordered sequence of type specifications of the (individual) variables and it is interpreted as a parameter set, i.e. as a set of compatible n -tuples of the elements of the sets corresponding to the types involved (compatible wrt all projections).

2.2 Quantifiers, chains of quantifiers

Our system defines quantifiers and predicates polymorphically. A generalized quantifier Q is an association to every set Z a subset of the power set of Z . If we have a predicate P defined in a context Γ , then for any interpretation of the context $\|\Gamma\|$ it is interpreted as a subset of its parameter set. Quantifier phrases, e.g. *every man* or *some woman*, are interpreted as follows: $\|\text{every}_{m:man}\| = \{\|man\|\}$ and $\|\text{some}_{w:woman}\| = \{X \subseteq \|woman\| : X \neq \emptyset\}$.

The interpretation of quantifier phrases is further extended into the interpretation of chains of quantifiers. Consider an example in (2):

(2) Every man loves a woman. They kiss them.

Multi-quantifier sentences such as the first sentence in (2) are known to be ambiguous with different readings corresponding to how various quantifiers are semantically related in the sentence. To account for the readings available for such multi-quantifier sentences, we raise quantifier phrases to the front of a sentence to form (generalized) quantifier prefixes - chains of quantifiers. Chains of quantifiers are built from quantifier phrases using three chain-constructors: pack-formation rule $(?, \dots, ?)$, sequential composition $?|?$, and parallel composition $\frac{?}{?}$. The semantical operations that correspond to the chain-constructors (known as cumulation, iteration and branching) capture in a compositional manner cumulative, scope-dependent and branching readings, respectively.

The idea of chain-constructors and the corresponding semantical operations builds on Mostowski's notion of quantifier (Mostowski,

1957) further generalized by Lindström to a so-called polyadic quantifier (Lindström, 1966), see (Bellert and Zawadowski, 1989). To use a familiar example, a multi-quantifier prefix like $\forall_{m:M}|\exists_{w:W}$ is thought of as a single two-place quantifier obtained by an operation on the two single quantifiers, and it has as denotation:

$$\|\forall_{m:M}|\exists_{w:W}\| = \{R \subseteq \|M\| \times \|W\| : \{a \in \|M\| : \{b \in \|W\| : \langle a, b \rangle \in R\} \in \|\exists_{w:W}\|\} \in \|\forall_{m:M}\|\}.$$

In this paper we generalize the three chain-constructors and the corresponding semantical operations to (pre-) chains defined on dependent types.

2.3 Dynamic extensions of contexts

In our system language expressions are all defined in context. Thus the first sentence in (2) (on the most natural interpretation where *a woman* depends on *every man*) translates (via the process described in Section 3) into a sentence with a chain of quantifiers in a context:

$$\Gamma \vdash \forall_{m:M}|\exists_{w:W} \text{Love}(m, w),$$

and says that the set of pairs, a man and a woman he loves, has the following property: the set of those men that love some woman each is the set of all men. The way to understand the second sentence in (2) (i.e. the anaphoric continuation) is that every man kisses the women he loves rather than those loved by someone else. Thus the first sentence in (2) must deliver some internal relation between the types corresponding to the two quantifier phrases.

In our analysis, the first sentence in (2) extends the context Γ by adding new variable specifications on newly formed types for every quantifier phrase in the chain $Ch = \forall_{m:M}|\exists_{w:W}$ - for the purpose of the formation of such new types we introduce a new type constructor \mathbb{T} . That is, the first sentence in (2) (denoted as φ) extends the context by adding:

$$t_{\varphi, \forall_m} : \mathbb{T}_{\varphi, \forall_m: M}; t_{\varphi, \exists_w} : \mathbb{T}_{\varphi, \exists_w: W}(t_{\varphi, \forall_m})$$

The interpretations of types (that correspond to quantifier phrases in Ch) from the extended context Γ_φ are defined in a two-step procedure using the inductive clauses through which we define Ch but in the reverse direction.

Step 1. We define fibers of new types by inverse induction.

Basic step.

For the whole chain $Ch = \forall_{m:M} \exists_{w:W}$ we put:

$$\|\mathbb{T}_{\varphi, \forall_{m:M} \exists_{w:W}}\| := \|Love\|.$$

Inductive step.

$$\begin{aligned} \|\mathbb{T}_{\varphi, \forall_{m:M}}\| &= \{a \in \|M\| : \{b \in \|W\| : \\ &\langle a, b \rangle \in \|Love\|\} \in \|\exists_{w:W}\|\} \end{aligned}$$

and for $a \in \|M\|$

$$\|\mathbb{T}_{\varphi, \exists_{w:W}}\|(a) = \{b \in \|W\| : \langle a, b \rangle \in \|Love\|\}$$

Step 2. We build dependent types from fibers.

$$\begin{aligned} \|\mathbb{T}_{\varphi, \exists_{w:W}}\| &= \bigcup \{ \{a\} \times \|\mathbb{T}_{\varphi, \exists_{w:W}}\|(a) : \\ &a \in \|\mathbb{T}_{\varphi, \forall_{m:M}}\| \} \end{aligned}$$

Thus the first sentence in (2) extends the context Γ by adding the type $\mathbb{T}_{\varphi, \forall_{m:M}}$, interpreted as $\|\mathbb{T}_{\varphi, \forall_{m:M}}\|$ (i.e. the set of men who love some women), and the dependent type $\mathbb{T}_{\varphi, \exists_{w:W}}(t_{\varphi, \forall_{m:M}})$, interpreted for $a \in \|\mathbb{T}_{\varphi, \forall_{m:M}}\|$ as $\|\mathbb{T}_{\varphi, \exists_{w:W}}\|(a)$ (i.e. the set of women loved by the man a).

Unbound anaphoric pronouns are interpreted with reference to the context created by the foregoing text: they are treated as universal quantifiers and newly formed (possibly dependent) types incrementally added to the context serve as their potential quantificational domains. That is, unbound anaphoric pronouns $they_m$ and $them_w$ in the second sentence of (2) have the ability to pick up and quantify universally over the respective interpretations. The anaphoric continuation in (2) translates into:

$$\begin{aligned} \Gamma_{\varphi} \vdash \forall_{t_{\varphi, \forall_{m:M}} : \mathbb{T}_{\varphi, \forall_{m:M}}} \forall_{t_{\varphi, \exists_{w:W}} : \mathbb{T}_{\varphi, \exists_{w:W}}(t_{\varphi, \forall_{m:M}})} \\ Kiss(t_{\varphi, \forall_{m:M}}, t_{\varphi, \exists_{w:W}}), \end{aligned}$$

where:

$$\begin{aligned} \|\forall_{t_{\varphi, \forall_{m:M}} : \mathbb{T}_{\varphi, \forall_{m:M}}} \forall_{t_{\varphi, \exists_{w:W}} : \mathbb{T}_{\varphi, \exists_{w:W}}(t_{\varphi, \forall_{m:M}})}\| = \\ \{R \subseteq \|\mathbb{T}_{\varphi, \exists_{w:W}}\| : \{a \in \|\mathbb{T}_{\varphi, \forall_{m:M}}\| : \\ \{b \in \|\mathbb{T}_{\varphi, \exists_{w:W}}\|(a) : \langle a, b \rangle \in R\} \in \\ \|\forall_{t_{\varphi, \exists_{w:W}} : \mathbb{T}_{\varphi, \exists_{w:W}}(t_{\varphi, \forall_{m:M}})}\|(a)\} \in \|\forall_{t_{\varphi, \forall_{m:M}} : \mathbb{T}_{\varphi, \forall_{m:M}}}\|\}, \end{aligned}$$

yielding the correct truth conditions: *Every man kisses every woman he loves.*

Our system also handles intra-sentential anaphora, as exemplified in (3):

(3) Every farmer who owns a donkey beats it.

To account for the dynamic contribution of modified common nouns (in this case common nouns modified by relative clauses) we include in our system *-sentences (i.e. sentences with dummy quantifier phrases). The modified common noun gets translated into a *-sentence (with a dummy-quantifier phrase $f : F$):

$$\Gamma \vdash f : F \mid \exists_{d:D} Own(f, d)$$

and we extend the context by dropping the specifications of variables: ($f : F, d : D$) and adding new variable specifications on newly formed types for every (dummy-) quantifier phrase in the chain Ch^* :

$$t_{\varphi, f} : \mathbb{T}_{\varphi, f:F}; t_{\varphi, \exists_d} : \mathbb{T}_{\varphi, \exists_d:D}(t_{\varphi, f}),$$

The interpretations of types (that correspond to the quantifier phrases in the Ch^*) from the extended context Γ_{φ} are defined in our two-step procedure. Thus the *-sentence in (3) extends the context by adding the type $\mathbb{T}_{\varphi, f:F}$ interpreted as $\|\mathbb{T}_{\varphi, f:F}\|$ (i.e. the set of farmers who own some donkeys), and the dependent type $\mathbb{T}_{\varphi, \exists_d:D}(t_{\varphi, f})$, interpreted for $a \in \|\mathbb{T}_{\varphi, f:F}\|$ as $\|\mathbb{T}_{\varphi, \exists_d:D}\|(a)$ (i.e. the set of donkeys owned by the farmer a). The main clause translates into:

$$\begin{aligned} \Gamma_{\varphi} \vdash \forall_{t_{\varphi, f} : \mathbb{T}_{\varphi, f:F}} \forall_{t_{\varphi, \exists_d} : \mathbb{T}_{\varphi, \exists_d:D}(t_{\varphi, f})} \\ Beat(t_{\varphi, f}, t_{\varphi, \exists_d}), \end{aligned}$$

yielding the correct truth conditions *Every farmer who owns a donkey beats every donkey he owns.* Importantly, since we quantify over fibers (and not over $\langle farmer, donkey \rangle$ pairs), our solution does not run into the so-called ‘proportion problem’.

Dynamic extensions of contexts and their interpretation are also defined for cumulative and branching continuations. Consider a cumulative example in (4):

(4) Last year three scientists wrote (a total of) five articles (between them). They presented them at major conferences.

Interpreted cumulatively, the first sentence in (4) translates into a sentence:

$$\Gamma \vdash (Three_{s:S}, Five_{a:A}) Write(s, a).$$

The anaphoric continuation in (4) can be interpreted in what Krifka calls a ‘correspondence’

fashion (Krifka, 1996). For example, Dr. K wrote one article, co-authored two more with Dr. N, who co-authored two more with Dr. S, and the scientists that cooperated in writing one or more articles also cooperated in presenting these (and no other) articles at major conferences. In our system, the first sentence in (4) extends the context by adding the type corresponding to $(Three_{s:S}, Five_{a:A})$:

$$t_{\varphi, (Three_s, Five_a)} : T_{\varphi, (Three_s; Five_a)},$$

interpreted as a set of tuples

$$\|T_{\varphi, (Three_s, Five_a)}\| = \{\langle c, d \rangle : c \in \|S\| \ \& \ d \in \|A\| \ \& \ c \text{ wrote } d\}$$

The anaphoric continuation then quantifies universally over this type (i.e. a set of pairs), yielding the desired truth-conditions *The respective scientists cooperated in presenting at major conferences the respective articles that they cooperated in writing.*

3 English-to-formal language translation

We assume a two-step translation process.

Representation. The syntax of the representation language - for the English fragment considered in this paper - is as follows.

$$\begin{aligned} S &\rightarrow Prd^n(QP_1, \dots, QP_n); \\ MCN &\rightarrow Prd^n(QP_1, \dots, CN, \dots, QP_n); \\ MCN &\rightarrow CN; \\ QP &\rightarrow Det \ MCN; \\ Det &\rightarrow every, most, three, \dots; \\ CN &\rightarrow man, woman, \dots; \\ Prd^n &\rightarrow enter, love, \dots; \end{aligned}$$

Common nouns (CNs) are interpreted as types, and common nouns modified by relative clauses (MCNs) - as *-sentences determining some (possibly dependent) types.

Disambiguation. Sentences of English, contrary to sentences of our formal language, are often ambiguous. Hence one sentence representation can be associated with more than one sentence in our formal language. The next step thus involves disambiguation. We take quantifier phrases of a given representation, e.g.:

$$P(Q_1X_1, Q_2X_2, Q_3X_3)$$

and organize them into all possible chains of quantifiers in suitable contexts with some restrictions imposed on particular quantifiers concerning the places in prefixes at which they can occur (a detailed elaboration of the disambiguation process is left for another place):

$$\frac{Q_1x_1:X_1 \mid Q_2x_2:X_2}{Q_3x_3:X_3} P(x_1, x_2, x_3).$$

4 System - syntax

4.1 Alphabet

The alphabet consists of:

type variables: X, Y, Z, \dots ;

type constants: $M, men, women, \dots$;

type constructors: Σ, Π, \mathbb{T} ;

individual variables: x, y, z, \dots ;

predicates: P, P', P_1, \dots ;

quantifier symbols: $\exists, \forall, five, Q_1, Q_2, \dots$;

three chain constructors: $?|?, \frac{?}{?}, (?, \dots, ?)$.

4.2 Context

A context is a list of type specifications of (individual) variables. If we have a context

$$\Gamma = x_1 : X_1, \dots, x_n : X_n (\langle x_i \rangle_{i \in J_n})$$

then the judgement

$$\vdash \Gamma : \text{cxt}$$

expresses this fact. Having a context Γ as above, we can declare a type X_{n+1} in that context

$$\Gamma \vdash X_{n+1} (\langle x_i \rangle_{i \in J_{n+1}}) : \text{type}$$

where $J_{n+1} \subseteq \{1, \dots, n\}$ such that if $i \in J_{n+1}$, then $J_i \subseteq J_{n+1}$, $J_1 = \emptyset$. The type X_{n+1} depends on variables $\langle x_i \rangle_{i \in J_{n+1}}$. Now, we can declare a new variable of the type $X_{n+1} (\langle x_i \rangle_{i \in J_{n+1}})$ in the context Γ

$$\Gamma \vdash x_{n+1} : X_{n+1} (\langle x_i \rangle_{i \in J_{n+1}})$$

and extend the context Γ by adding this variable specification, i.e. we have

$$\vdash \Gamma, x_{n+1} : X_{n+1} (\langle x_i \rangle_{i \in J_{n+1}}) : \text{cxt}$$

Γ' is a *subcontext* of Γ if Γ' is a context and a sublist of Γ . Let Δ be a list of variable specifications from a context Γ , Δ' the least subcontext of Γ containing Δ . We say that Δ is *convex* iff $\Delta' - \Delta$ is again a context.

The variables the types depend on are always explicitly written down in specifications. We can think of a context as (a linearization of) a partially ordered set of declarations such that the declaration of a variable x (of type X) precedes the declaration of the variable y (of type Y) iff the type Y depends on the variable x .

The formation rules for both Σ - and Π -types are as usual.

4.3 Language

Quantifier-free formulas. Here, we need only predicates applied to variables. So we write

$$\Gamma \vdash P(x_1, \dots, x_n) : \mathbf{qf-f}$$

to express that P is an n -ary predicate and the specifications of the variables x_1, \dots, x_n form a subcontext of Γ .

Quantifier phrases. If we have a context $\Gamma, y : Y(\vec{x}), \Delta$ and quantifier symbol Q , then we can form a *quantifier phrase* $Q_{y:Y(\vec{x})}$ in that context. We write

$$\Gamma, y : Y(\vec{x}), \Delta \vdash Q_{y:Y(\vec{x})} : \mathbf{QP}$$

to express this fact. In a quantifier phrase $Q_{y:Y(\vec{x})}$: the variable y is the *binding variable* and the variables \vec{x} are *indexing variables*.

Packs of quantifiers. Quantifiers phrases can be grouped together to form a pack of quantifiers. The pack of quantifiers formation rule is as follows.

$$\frac{\Gamma \vdash Q_i y_i:Y_i(\vec{x}_i) : \mathbf{QP} \quad i = 1, \dots, k}{\Gamma \vdash (Q_1 y_1:Y_1(\vec{x}_1), \dots, Q_k y_k:Y_k(\vec{x}_k)) : \mathbf{pack}}$$

where, with $\vec{y} = y_1, \dots, y_k$ and $\vec{x} = \bigcup_{i=1}^k \vec{x}_i$, we have that $y_i \neq y_j$ for $i \neq j$ and $\vec{y} \cap \vec{x} = \emptyset$. In so constructed pack: the binding variables are \vec{y} and the indexing variables are \vec{x} . We can denote such a pack $Pc_{\vec{y}:\vec{Y}(\vec{x})}$ to indicate the variables involved. One-element pack will be denoted and treated as a quantifier phrase. This is why we denote such a pack as $Q_{y:Y(\vec{x})}$ rather than $(Q_{y:Y(\vec{x})})$.

Pre-chains and chains of quantifiers. Chains and pre-chains of quantifiers have binding variables and indexing variables. By $Ch_{\vec{y}:\vec{Y}(\vec{x})}$ we denote a pre-chain with binding variables \vec{y} and indexing variables \vec{x} so that the type of the variable y_i is $Y_i(\vec{x}_i)$ with $\bigcup_i \vec{x}_i = \vec{x}$. Chains of quantifiers are pre-chains in which all indexing variables are bound. Pre-chains of quantifiers arrange quantifier phrases into N -free pre-orders, subject to some binding conditions. Mutually comparable QPs in a pre-chain sit in one pack. Thus the pre-chains are built from packs via two chain-constructors of sequential $?$ and parallel composition $\overset{?}{\parallel}$. The chain formation rules are as follows.

1. *Packs of quantifiers.* Packs of quantifiers are pre-chains of quantifiers with the same binding variable and the same indexing variables, i.e.

$$\frac{\Gamma \vdash Pc_{\vec{y}:\vec{Y}(\vec{x})} : \mathbf{pack}}{\Gamma \vdash Pc_{\vec{y}:\vec{Y}(\vec{x})} : \mathbf{p-ch}}$$

2. Sequential composition of pre-chains

$$\frac{\Gamma \vdash Ch_1 \vec{y}_1:\vec{Y}_1(\vec{x}_1) : \mathbf{p-ch}, \Gamma \vdash Ch_2 \vec{y}_2:\vec{Y}_2(\vec{x}_2) : \mathbf{p-ch}}{\Gamma \vdash Ch_1 \vec{y}_1:\vec{Y}_1(\vec{x}_1) | Ch_2 \vec{y}_2:\vec{Y}_2(\vec{x}_2) : \mathbf{p-ch}}$$

provided $\vec{y}_2 \cap (\vec{y}_1 \cup \vec{x}_1) = \emptyset$; the specifications of the variables $(\vec{x}_1 \cup \vec{x}_2) - (\vec{y}_1 \cup \vec{y}_2)$ form a context, a subcontext of Γ . In so obtained pre-chain: the binding variables are $\vec{y}_1 \cup \vec{y}_2$ and the indexing variables are $\vec{x}_1 \cup \vec{x}_2$.

3. Parallel composition of pre-chains

$$\frac{\Gamma \vdash Ch_1 \vec{y}_1:\vec{Y}_1(\vec{x}_1) : \mathbf{p-ch}, \Gamma \vdash Ch_2 \vec{y}_2:\vec{Y}_2(\vec{x}_2) : \mathbf{p-ch}}{\Gamma \vdash \frac{Ch_1 \vec{y}_1:\vec{Y}_1(\vec{x}_1)}{Ch_2 \vec{y}_2:\vec{Y}_2(\vec{x}_2)} : \mathbf{p-ch}}$$

provided $\vec{y}_2 \cap (\vec{y}_1 \cup \vec{x}_1) = \emptyset = \vec{y}_1 \cap (\vec{y}_2 \cup \vec{x}_2)$. As above, in so obtained pre-chain: the binding variables are $\vec{y}_1 \cup \vec{y}_2$ and the indexing variables are $\vec{x}_1 \cup \vec{x}_2$.

A pre-chain of quantifiers $Ch_{\vec{y}:\vec{Y}(\vec{x})}$ is a *chain* iff $\vec{x} \subseteq \vec{y}$. The following

$$\Gamma \vdash Ch_{\vec{y}:\vec{Y}(\vec{x})} : \mathbf{chain}$$

expresses the fact that $Ch_{\vec{y}:\vec{Y}(\vec{x})}$ is a chain of quantifiers in the context Γ .

*Formulas, sentences and *-sentences.* The formulas have binding variables, indexing variables and argument variables. We write $\varphi_{\vec{y}:\vec{Y}(\vec{x})}(\vec{z})$ for a formula with binding variables \vec{y} , indexing variables \vec{x} and argument variables \vec{z} . We have the following formation rule for formulas

$$\frac{\Gamma \vdash A(\vec{z}) : \mathbf{qf-f}, \Gamma \vdash Ch_{\vec{y}:\vec{Y}(\vec{x})} : \mathbf{p-ch}}{\Gamma \vdash Ch_{\vec{y}:\vec{Y}(\vec{x})} A(\vec{z}) : \mathbf{formula}}$$

provided \vec{y} is final in \vec{z} , i.e. $\vec{y} \subseteq \vec{z}$ and variable specifications of $\vec{z} - \vec{y}$ form a subcontext of Γ . In so constructed formula: the binding variables are \vec{y} , the indexing variables are \vec{x} , and the argument variables are \vec{z} .

A formula $\varphi_{\vec{y}:\vec{Y}(\vec{x})}(\vec{z})$ is a *sentence* iff $\vec{z} \subseteq \vec{y}$ and $\vec{x} \subseteq \vec{y}$. So a sentence is a formula without free variables, neither individual nor indexing. The following

$$\Gamma \vdash \varphi_{\vec{y}:\vec{Y}(\vec{x})}(\vec{z}) : \mathbf{sentence}$$

expresses the fact that $\varphi_{\vec{y}:\vec{Y}(\vec{x})}(\vec{z})$ is a sentence formed in the context Γ .

We shall also consider some special formulas that we call **-sentences*. A formula $\varphi_{\vec{y}:\vec{Y}(\vec{x})}(\vec{z})$ is a **-sentence* if $\vec{x} \subseteq \vec{y} \cup \vec{z}$ but the set $\vec{z} - \vec{y}$ is possibly

not empty and moreover the type of each variable in $\vec{z} - \vec{y}$ is *constant*, i.e., it does not depend on variables of other types. In such case we consider the $\vec{z} - \vec{y}$ as a set of binding variables of an additional pack called a *dummy pack* that is placed in front of the whole chain Ch . The chain 'extended' by this dummy pack will be denoted by Ch^* . Clearly, if $\vec{z} - \vec{y}$ is empty there is no dummy pack and the chain Ch^* is Ch , i.e. sentences are *-sentences without dummy packs. We write

$$\Gamma \vdash \varphi_{\vec{y}:Y(\vec{x})}(\vec{z}) : \text{*sentence}$$

to express the fact that $\varphi_{\vec{y}:Y(\vec{x})}(\vec{z})$ is a *-sentence formed in the context Γ .

Having formed a *-sentence φ we can form a new context Γ_φ defined in the next section.

Notation. For semantics we need some notation for the variables in the *-sentence. Suppose we have a *-sentence

$$\Gamma \vdash Ch_{\vec{y}:Y(\vec{x})} P(\vec{z}) : \text{*sentence}$$

We define: (i) The environment of pre-chain Ch : $Env(Ch) = Env(Ch_{\vec{y}:Y(\vec{x})})$ - is the context defining variables $\vec{x} - \vec{y}$; (ii) The binding variables of pre-chain Ch : $Bv(Ch) = Bv(Ch_{\vec{y}:Y(\vec{x})})$ - is the convex set of declarations in Γ of the binding variables in \vec{y} ; (iii) $\mathbf{env}(Ch) = \mathbf{env}(Ch_{\vec{y}:Y(\vec{x})})$ - the set of variables in the environment of Ch , i.e. $\vec{x} - \vec{y}$; (iv) $\mathbf{bv}(Ch) = \mathbf{bv}(Ch_{\vec{y}:Y(\vec{x})})$ - the set of binding variables \vec{y} ; (v) The environment of a pre-chain Ch' in a *-sentence $\varphi = Ch_{\vec{y}:Y(\vec{x})} P(\vec{z})$, denoted $Env_\varphi(Ch')$, is the set of binding variables in all the packs in Ch^* that are $<_\varphi$ -smaller than all packs in Ch' . Note $Env(Ch') \subseteq Env_\varphi(Ch')$. If $Ch' = Ch_1 | Ch_2$ is a sub-pre-chain of the chain $Ch_{\vec{y}:Y(\vec{x})}$, then $Env_\varphi(Ch_2) = Env_\varphi(Ch_1) \cup Bv(Ch_1)$ and $Env_\varphi(Ch_1) = Env_\varphi(Ch')$.

4.4 Dynamic extensions

Suppose we have constructed a *-sentence in a context

$$\Gamma \vdash Ch_{\vec{y}:Y(\vec{x})} A(\vec{z}) : \text{*sentence}$$

We write φ for $Ch_{\vec{y}:Y(\vec{x})} A(\vec{z})$.

We form a context Γ_φ dropping the specifications of variables \vec{z} and adding one type and one variable specification for each pack in $Packs_{Ch^*}$.

Let $\check{\Gamma}$ denote the context Γ with the specifications of the variables \vec{z} deleted. Suppose $\Phi \in Packs_{Ch^*}$ and Γ' is an extension of the context

$\check{\Gamma}$ such that one variable specification $t_{\Phi',\varphi} : T_{\Phi',\varphi}$ was already added for each pack $\Phi' \in Packs_{Ch^*}$ such that $\Phi' <_{Ch^*} \Phi$ but not for Φ yet. Then we declare a type

$$\Gamma' \vdash T_{\Phi,\varphi}(\langle t_{\Phi',\varphi} \rangle_{\Phi' \in Packs_{Ch^*}, \Phi' <_{Ch^*} \Phi}) : \text{type}$$

and we extend the context Γ' by a specification of a variable $t_{\Phi,\varphi}$ of that type

$$\Gamma', t_{\Phi,\varphi} : T_{\Phi,\varphi}(\langle t_{\Phi',\varphi} \rangle_{\Phi' \in Packs_{Ch^*}, \Phi' <_{Ch^*} \Phi}) : \text{cxt}$$

The context obtained from $\check{\Gamma}$ by adding the new variables corresponding to all the packs $Packs_{Ch^*}$ as above will be denoted by

$$\Gamma_\varphi = \check{\Gamma} \cup \mathbb{T}(Ch_{\vec{y}:Y(\vec{x})} A(\vec{z})).$$

At the end we add another context formation rule

$$\frac{\Gamma \vdash Ch_{\vec{y}:Y(\vec{x})} A(\vec{z}) : \text{*sentence},}{\Gamma_\varphi : \text{cxt}}$$

Then we can build another formula starting in the context Γ_φ . This process can be iterated. Thus in this system sentence φ in a context Γ is constructed via *specifying sequence of formulas*, with the last formula being the sentence φ . However, for the lack of space we are going to describe here only one step of this process. That is, sentence φ in a context Γ can be constructed via *specifying *-sentence* ψ extending the context as follows

$$\Gamma \vdash \psi : \text{*sentence}$$

$$\Gamma_\psi \vdash \varphi : \text{sentence}$$

For short, we can write

$$\Gamma \vdash \Gamma_\psi \vdash \varphi : \text{sentence}$$

5 System - semantics

5.1 Interpretation of dependent types

The context Γ

$$\vdash x : X(\dots), \dots, z : Z(\dots, x, y, \dots) : \text{cxt}$$

gives rise to a dependence graph. A *dependence graph* $DG_\Gamma = (T_\Gamma, E_\Gamma)$ for the context Γ has types of Γ as vertices and an edge $\pi_{Y,x} : Y \rightarrow X$ for every variable specification $x : X(\dots)$ in Γ and every type $Y(\dots, x, \dots)$ occurring in Γ that depends on x .

The *dependence diagram* for the context Γ is an association $\| - \| : DG_\Gamma \rightarrow Set$ to every type X in T_Γ a set $\|X\|$ and every edge $\pi_{Y,x} : Y \rightarrow X$ in E_Γ a function $\|\pi_{Y,x}\| : \|Y\| \rightarrow \|X\|$, so that whenever we have a triangle of edges in E_Γ , $\pi_{Y,x}$ as before $\pi_{Z,y} : Z \rightarrow Y$, $\pi_{Z,x} : Z \rightarrow X$ we have $\|\pi_{Z,x}\| = \|\pi_{Y,x}\| \circ \|\pi_{Z,y}\|$.

The interpretation of the context Γ , the *parameter space* $\|\Gamma\|$, is the limit of the dependence diagram $\| - \| : DG_\Gamma \rightarrow Set$. More specifically,

$$\begin{aligned} \|\Gamma\| &= \|x : X(\dots), \dots, z : Z(\dots, x, y, \dots)\| = \\ &\{\vec{a} : \text{dom}(\vec{a}) = \mathbf{var}(\Gamma), \vec{a}(z) \in \|Z\|(\vec{a}[\mathbf{env}(Z)]), \\ &\|\pi_{Z,x}\|(\vec{a}(z)) = \vec{a}(x), \text{ for } z : Z \text{ in } \Gamma, x \in \mathbf{env}Z\} \end{aligned}$$

where $\mathbf{var}(\Gamma)$ denotes variables specified in Γ and $\mathbf{env}(Z)$ denotes indexing variables of the type Z .

The interpretation of the Σ - and Π -types are as usual.

5.2 Interpretation of language

Interpretation of predicates and quantifier symbols. Both predicates and quantifiers are interpreted polymorphically.

If we have a predicate P defined in a context Γ :

$$x_1 : X_1, \dots, x_n : X_n(\langle x_i \rangle_{i \in J_n}) \vdash P(\vec{x}) : \mathbf{qf}$$

then, for any interpretation of the context $\|\Gamma\|$, it is interpreted as a subset of its parameter set, i.e. $\|P\| \subseteq \|\Gamma\|$.

Quantifier symbol Q is interpreted as quantifier $\|Q\|$ i.e. an association to every¹ set Z a subset $\|Q\|(Z) \subseteq \mathcal{P}(Z)$.

Interpretation of pre-chains and chains of quantifiers. We interpret QP's, packs, pre-chains, and chains in the environment of a sentence Env_φ . This is the only case that is needed. We could interpret the aforementioned syntactic objects in their natural environment Env (i.e. independently of any given sentence) but it would unnecessarily complicate some definitions. Thus having a $(*)$ -sentence $\varphi = Ch_{\vec{y}:Y(\vec{x})} P(\vec{z})$ (defined in a context Γ) and a sub-pre-chain (QP, pack) Ch' , for $\vec{a} \in \|Env_\varphi(Ch')\|$ we define the meaning of

$$\|Ch'\|(\vec{a})$$

Notation. Let $\varphi = Ch_{\vec{y}:Y} P(\vec{y})$ be a $*$ -sentence built in a context Γ , Ch' a pre-chain used in the construction of the $(*)$ -chain Ch . Then

¹This association can be partial.

$Env_\varphi(Ch')$ is a sub-context of Γ disjoint from the convex set $Bv(Ch')$ and $Env_\varphi(Ch')$, $Bv(Ch')$ is a sub-context of Γ . For $\vec{a} \in \|Env_\varphi(Ch')\|$ we define $\|Bv(Ch')\|(\vec{a})$ to be the largest set such that

$$\{\vec{a}\} \times \|Bv(Ch')\|(\vec{a}) \subseteq \|Env_\varphi(Ch'), Bv(Ch')\|$$

Interpretation of quantifier phrases. If we have a quantifier phrase

$$\Gamma \vdash Q_{y:Y(\vec{x})} : \mathbf{QP}$$

and $\vec{a} \in \|Env_\varphi(Q_{y:Y(\vec{x})})\|$, then it is interpreted as $\|Q\|(\|Y\|(\vec{a})) \subseteq \mathcal{P}(\|Y\|(\vec{a}[\vec{x}]))$.

Interpretation of packs. If we have a pack of quantifiers in the sentence φ

$$Pc = (Q_{1y_1:Y_1(\vec{x}_1)}, \dots, Q_{ny_n:Y_n(\vec{x}_n)})$$

and $\vec{a} \in \|Env_\varphi(Pc)\|$, then its interpretation with the parameter \vec{a} is

$$\begin{aligned} \|Pc\|(\vec{a}) &= \|(Q_{1y_1:Y_1(\vec{x}_1)}, \dots, Q_{ny_n:Y_n(\vec{x}_n)})\|(\vec{a}) = \\ &\{A \subseteq \prod_{i=1}^n \|Y_i\|(\vec{a}[\vec{x}_i]) : \pi_i(A) \in \|Q_i\|(\|Y_i\|(\vec{a}[\vec{x}_i])), \\ &\text{for } i = 1, \dots, n\} \end{aligned}$$

where π_i is the i -th projection from the product.

Interpretation of chain constructors.

1. *Parallel composition.* For a pre-chain of quantifiers in the sentence φ

$$Ch' = \frac{Ch_{1\vec{y}_1:\vec{Y}_1(\vec{x}_1)}}{Ch_{2\vec{y}_2:\vec{Y}_2(\vec{x}_2)}}$$

and $\vec{a} \in \|Env_\varphi(Ch')\|$ we define

$$\left\| \frac{Ch_{1\vec{y}_1:\vec{Y}_1(\vec{x}_1)}}{Ch_{2\vec{y}_2:\vec{Y}_2(\vec{x}_2)}} \right\|(\vec{a}) = \{A \times B :$$

$$A \in \|Ch_{1\vec{y}_1:\vec{Y}_1(\vec{x}_1)}\|(\vec{a}[\vec{x}_1]) \text{ and}$$

$$B \in \|Ch_{2\vec{y}_2:\vec{Y}_2(\vec{x}_2)}\|(\vec{a}[\vec{x}_2])\}$$

2. *Sequential composition.* For a pre-chain of quantifiers in the sentence φ

$$Ch' = Ch_{1\vec{y}_1:\vec{Y}_1(\vec{x}_1)} | Ch_{2\vec{y}_2:\vec{Y}_2(\vec{x}_2)}$$

and $\vec{a} \in \|Env_\varphi(Ch')\|$ we define

$$\|Ch_{1\vec{y}_1:\vec{Y}_1(\vec{x}_1)} | Ch_{2\vec{y}_2:\vec{Y}_2(\vec{x}_2)}\|(\vec{a}) =$$

$$\{R \subseteq \|Bv(Ch')\|(\vec{a}) : \{\vec{b} \in \|Bv(Ch_1)\|(\vec{a}) :$$

$$\{\vec{c} \in \|Bv(Ch_2)\|(\vec{a}, \vec{b}) : \langle \vec{b}, \vec{c} \rangle \in R\} \in \|Ch_2\|(\vec{a}, \vec{b})$$

$$\|Ch_2\|(\vec{a}, \vec{b}) \in \|Ch_1\|(\vec{a})$$

Validity. A sentence

$$\vec{x} : \vec{X} \vdash Ch_{\vec{y}:\vec{Y}} P(\vec{y})$$

is true under the above interpretation iff

$$\|P\|(\|\vec{Y}\|) \in \|Ch_{\vec{y}:\vec{Y}}\|$$

5.3 Interpretation of dynamic extensions

Suppose we obtain a context Γ_φ from Γ by the following rule

$$\frac{\Gamma \vdash Ch_{\vec{y}:\vec{Y}(\vec{x})} A(\vec{z}) : * \text{-sentence,}}{\Gamma_\varphi : \text{cxt}}$$

where φ is $Ch_{\vec{y}:\vec{Y}(\vec{x})} A(\vec{z})$. Then

$$\Gamma_\varphi = \check{\Gamma} \cup \mathbb{T}(Ch_{\vec{y}:\vec{Y}(\vec{x})} A(\vec{z})).$$

From dependence diagram $\| - \|^\Gamma : DG_\Gamma \rightarrow Set$ we shall define another dependence diagram

$$\| - \| = \| - \|^\Gamma : DG_{\Gamma_\varphi} \rightarrow Set$$

Thus, for $\Phi \in Pack_{Ch^*}$ we need to define $\|\mathbb{T}_{\Phi, \varphi}\|^\Gamma$ and for $\Phi' <_{Ch^*} \Phi$ we need to define

$$\|\pi_{\mathbb{T}_{\Phi, \varphi}, t_{\Phi'}}\| : \|\mathbb{T}_{\Phi, \varphi}\| \longrightarrow \|\mathbb{T}_{\Phi', \varphi}\|$$

This will be done in two steps:

Step 1. (Fibers of new types defined by inverse induction.)

We shall define for the sub-prechains Ch' of Ch^* and $\vec{a} \in \|Env_\varphi(Ch')\|$ a set

$$\|\mathbb{T}_{\varphi, Ch'}\|(\vec{a}) \subseteq \|Bv(Ch')\|(\vec{a})$$

This is done using the inductive clauses through which we have defined Ch^* but in the reverse direction.

The *basic case* is when $Ch' = Ch^*$. We put

$$\|\mathbb{T}_{\varphi, Ch}\|(\emptyset) = \|P\|$$

The *inductive step*. Now assume that the set $\|\mathbb{T}_{\varphi, Ch'}\|(\vec{a})$ is defined for $\vec{a} \in \|Env_\varphi(Ch')\|$.

Parallel decomposition. If we have

$$Ch' = \frac{Ch_{\vec{y}_1:\vec{Y}_1(\vec{x}_1)}}{Ch_{\vec{y}_2:\vec{Y}_2(\vec{x}_2)}}$$

then we define sets

$$\|\mathbb{T}_{\varphi, Ch_i}\|(\vec{a}) \in \|Ch_i\|(\vec{a})$$

for $i = 1, 2$ so that

$$\|\mathbb{T}_{\varphi, Ch'}\|(\vec{a}) = \|\mathbb{T}_{\varphi, Ch_1}\|(\vec{a}) \times \|\mathbb{T}_{\varphi, Ch_2}\|(\vec{a})$$

if such sets exist, and these sets ($\|\mathbb{T}_{\varphi, Ch_i}\|(\vec{a})$) are undefined otherwise.

Sequential decomposition. If we have

$$Ch' = Ch_{\vec{y}_1:\vec{Y}_1(\vec{x}_1)} | Ch_{\vec{y}_2:\vec{Y}_2(\vec{x}_2)}$$

then we put

$$\|\mathbb{T}_{\varphi, Ch_1}\|(\vec{a}) = \{\vec{b} \in \|Bv(Ch_1)\|(\vec{a}) :$$

$$\{\vec{c} \in \|Bv(Ch_2)\|(\vec{a}, \vec{b}) : \langle \vec{b}, \vec{c} \rangle \in \|\mathbb{T}_{\varphi, Ch'}\|(\vec{a})\} \\ \in \|Ch_2\|(\vec{a}, \vec{b})\}$$

For $\vec{b} \in \|Bv(Ch_1)\|$ we put

$$\|\mathbb{T}_{\varphi, Ch_2}\|(\vec{a}, \vec{b}) = \{\vec{c} \in \|Bv(Ch_2)\|(\vec{a}, \vec{b}) :$$

$$\langle \vec{b}, \vec{c} \rangle \in \|\mathbb{T}_{\varphi, Ch'}\|(\vec{a})\}$$

Step 2. (Building dependent types from fibers.)

If Φ is a pack in Ch^* , $\vec{a} \in \|Env_\varphi(\Phi)\|$ then we put

$$\|\mathbb{T}_{\varphi, \Phi}\| = \bigcup \{\{\vec{a}\} \times \|\mathbb{T}_{\phi, \Phi}\|(\vec{a}) : \vec{a} \in \|Env_\varphi(\Phi)\|\},$$

$$\forall \Phi' <_{Ch^*} \Phi, (\vec{a} \upharpoonright \mathbf{env}_\varphi(\Phi')) \in \|\mathbb{T}_{\varphi, \Phi'}\|$$

It remains to define the projections between dependent types. If $\Phi' <_\varphi \Phi$ we define

$$\pi_{\mathbb{T}_{\varphi, \Phi}, t_{\Phi'}} : \|\mathbb{T}_{\varphi, \Phi}\| \longrightarrow \|\mathbb{T}_{\varphi, \Phi'}\|$$

so that $\vec{a} \mapsto \vec{a} \upharpoonright (\mathbf{env}_\varphi(\Phi') \cup \mathbf{bv}\Phi')$.

6 Conclusion

It was our intention in this paper to show that adopting a typed approach to generalized quantification allows a uniform treatment of a wide array of anaphoric data involving natural language quantification.

Acknowledgments

The work of Justyna Grudzińska was funded by the National Science Center on the basis of decision DEC-2012/07/B/HS1/00301. The authors would like to thank the anonymous reviewers for valuable comments on an earlier version of this paper.

References

- Barwise, Jon and Robin Cooper. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4: 159-219.
- Bellert, Irena and Marek Zawadowski. 1989. Formalization of the feature system in terms of pre-orders. In Irena Bellert *Feature System for Quantification Structures in Natural Language*. Foris Dordrecht. 155-172.
- Dekker, Paul. 1994. Predicate logic with anaphora. In Lynn Santelmann and Mandy Harvey (eds.), *Proceedings SALT IX*. Ithaca, NY: DMLL Publications, Cornell University. 79-95.
- Dekker, Paul. 2008. A multi-dimensional treatment of quantification in extraordinary English. *Linguistics and Philosophy* 31: 101-127.
- Elworthy, David A. H. 1995. A theory of anaphoric information. *Linguistics and Philosophy* 18: 297-332.
- Elbourne, Paul D. 2005. *Situations and Individuals*. Cambridge, MA: MIT Press.
- Evans, Gareth 1977. Pronouns, Quantifiers, and Relative Clauses (I). *Canadian Journal of Philosophy* 7: 467-536.
- Heim, Irene. 1990. E-type pronouns and donkey anaphora. *Linguistics and Philosophy* 13: 137-78.
- Groenendijk, Jeroen and Martin Stokhof. 1991. Dynamic Predicate Logic. *Linguistics and Philosophy* 14: 39-100.
- Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.
- Krifka, Manfred. 1996. Parametrized sum individuals for plural reference and partitive quantification. *Linguistics and Philosophy* 19: 555-598.
- Lindström, Per. 1966. First-order predicate logic with generalized quantifiers. *Theoria* 32: 186-95.
- Martin-Löf, Per. 1972. An intuitionistic theory of types. Technical Report, University of Stockholm.
- Mostowski, Andrzej. 1957. On a generalization of quantifiers. *Fundamenta Mathematicae* 44: 12-36.
- Nouwen, Rick. 2003. *Plural pronominal anaphora in context: dynamic aspects of quantification*. Ph.D. thesis, UiL-OTS, Utrecht, LOT dissertation series, No. 84.
- Ranta, Arne. 1994. *Type-Theoretical Grammar*. Oxford University Press, Oxford.
- Van den Berg, Martin H. 1996. *The Internal Structure of Discourse*. Ph.D. thesis, Universiteit van Amsterdam, Amsterdam.