A Remark on Collective Quantification

Juha Kontinen

Department of Mathematics and Statistics
University of Helsinki

Jakub Szymanik

Institute for Logic, Language and Computation
University of Amsterdam

June 21, 2007

Abstract

We consider collective quantification in natural language. For many years the common strategy in formalizing collective quantification has been to define the meanings of collective determiners, quantifying over collections, using certain type-shifting operations. These type-shifting operations, i.e., lifts, define the collective interpretations of collective determiners systematically from the standard meanings of quantifiers. All the lifts considered in the literature turn out to be definable in second-order logic. We argue that second-order definable quantifiers are probably not expressive enough to formalize all collective quantification in natural language.

Keywords: collective quantification, Lindström quantifiers, second-order generalized quantifiers, type-shifting, definability, computational complexity.

1 Introduction

Recently, there has been some interest in measuring the complexity of semantic constructions of natural language. These studies have been motivated by certain mathematical questions (see e.g. Hella et al., 1997) as well as cognitive considerations (see e.g. McMillan et al., 2005; Szymanik, 2007). As the complexity of the semantics of a language heavily depends on the expressive power of its quantifiers, most of the studies have focused on quantification. In particular, Mostowski and Wojtyniak (2004), followed by Sevenster (2006), study computational complexity of natural language quantifiers, and Mostowski and Szymanik (2005) search for semantic bounds of the so-called everyday fragment of natural language. In all of these studies only distributive reading of natural language determiners have been considered. In contrast — as the properties of plural objects are becoming more and more important in many areas (e.g. in game-theoretical investigations, where groups of agents are acting) — this paper is devoted to the collective readings of quantifiers. We mainly focus on definability issues, but we also discuss the connections to computational complexity. The rest of the introduction roughly presents the state of the art in formal semantics of collective plural noun phrases.

Already Bertrand Russell (1903) noticed that natural language contains quantification not only over objects, but also over collections of objects. The notion of a collective reading is a semantic one — as opposed to the grammatical notion of plurality — and it applies to the meanings of certain occurrences of plural noun phrases. The phenomenon is illustrated by the following sentences:

(1) Tikitu and Samson lifted the poker table together.

(2) The decks of cards on the table had different colors.

(3) Nina and Jonathan had flush together, but each of them alone had nothing.

(4) Most poker hands have no chance against an Ace and a Two.

(5) Most of the card combinations do not contain a picture card.

(6) Most of the PhD students play Texas Hold'em.

(7) Most groups of students have never played Hold'em together.

The question arises how should we model collective quantification in formal semantics. Many authors have proposed different mathematical accounts of collectivity in language (see Lønning, 1997, for an overview and references). None of the approaches is widely accepted, but, at least, all authors agree on the following three principles formulated by Link (1991):

Atomicity Each collection is constituted by all the individuals it contains.

Completeness Collections may be combined into new collections.

Atoms Individuals are collections consisting of only a single member.

In Link (1983) one finds the idea of replacing the domain of discourse, which consists of entities, with the structure of a complete join semilattice. The author focuses on the cumulative properties of mass nouns and observes that the same approach can be applied to cover plural nouns. The idea is to enrich the structure of models to account for cumulative references. The main advantage of this algebraic perspective is that it unifies the view on collective predication and predication involving mass nouns.

J. van der Does (1992) noticed that all that can be modeled with the algebraic models can be done as well within type theory. This alternative tradition, starting with the works of Bartsch (1973) and Bennett (1974), uses extensional type theory with the basic types: e (entities) and t (truth values), and compound types: $\alpha\beta$ (functions mapping type α objects onto type β objects). Together with the idea of type-shifting, introduced in the seminal paper of Partee and Rooth (1983), and then formally developed by J. van Benthem (1991), it gives a new approach to modeling collectivity in natural language. The strategy, introduced by Scha (1981) and later advocated and developed by J. van der Does (1992, 1996) and Winter (2001), is to lift first-order generalized quantifiers to a second-order setting. In the type theoretical terms the trick is to shift determiners of type $((et)((et)t))$, related to the distributive readings of quantifiers, into determiners of type $((et)(((et)t)t))$ which can be used to formalize collective readings of quantifiers.

In the next section we illustrate the idea of lifting first-order quantifiers by a few examples. Then we introduce, both, first-order and second-order generalized quantifiers, and show that the type theoretic approach can be redefined in terms of second-order generalized quantifiers. The idea of type-shifting turns out to be very closely related to the notion of definability which is central in generalized quantifier theory. We show that the type-shifting operations considered in the literature (i.e. lifts) are definable in second-order logic. This observation allows us to point out the restrictions of the type-shifting strategy used in the literature. In particular, we show that the collective meaning of the determiner Most can not be uniformly defined by any lift definable in second-order logic, unless the counting hierarchy collapses in computational complexity theory.

2 Lifting first-order determiners

Let us consider the following example sentences involving collective quantification.

(8) Five people lifted the table.

(9) Some students played poker together.

(10) All combinations of cards are losing in some situations.

The type-shifting strategy defines the collective readings of determiners by raising the types of the corresponding distributive determiners. In other words, the idea is to lift first-order generalized quantifiers — expressing properties of subsets of the discourse — into the second-order setting, in which it is possible to speak about the properties of collections of all subsets over a given domain. Let us analyze the examples.

The distributive reading of the sentence (8) claims that the total number of students who lifted the table on their own is exactly five. This statement can be formalized in elementary logic by the formula (11):

(11) $\exists^{=5}x[\mathrm{People}(x) \wedge \mathrm{Lift}(x)]$.

The collective interpretation of (8) claims that there was a collection of exactly five students who jointly lifted the table. This can be formalized by lifting the formula (11) to the second-order formula (12):

(12) $\exists X[\text{Card}(X) = 5 \wedge X \subseteq \text{People} \wedge \text{Lift}(X)]$.

In the similar way, by lifting the corresponding first-order determiners, we can express the collective readings of sentences (9)–(10) as follows:

(13) $\exists X[X \subseteq \text{Students} \wedge \text{Play}(X)]$.

(14) $\forall X[X \subseteq \text{Cards} \Rightarrow \text{Lose}(X)]$.

All the examples above can be described in terms of the uniform procedure of turning a determiner of type $((et)((et)t))$ into a determiner of type $((et)(((et)t)t))$ by the means of the type-shifting operator called *existential modifier*, $(\cdot)^{EM}$. Fix a universe of discourse U and take any $X \subseteq U$, and $Y \subseteq \mathcal{P}(U)$. Define the existential lift Q^{EM} of a quantifier Q in the following way:

$$\mathsf{Q}^{EM}(X, Y) \text{ is true } \iff \exists Z \subseteq X[\mathsf{Q}(X, Z) \wedge Z \in Y].$$

In the literature, lifts have been defined also for distributive and so-called neutral readings of sentences. For example, Ben Avi and Winter (2003) define the following lift — called dfit — for determiner fitting, to overcome some problems related to the monotonicity properties of the previous lifts considered in the literature. Note that the dfit operator turns a determiner of type $((et)((et)t))$ to a determiner of type $(((et)t)(((et)t)t))$, i.e., for all $X, Y \subseteq \mathcal{P}(U)$ we have that

$$\mathsf{Q}^{\text{dfit}}(X, Y) \text{ is true}$$

$$\iff$$

$$\mathsf{Q}[\cup X, \cup(X \cap Y)] \wedge [X \cap Y = \emptyset \vee \exists W \in X \cap Y \wedge \mathsf{Q}(\cup X, W)].$$

For us the most important observation is that all the lifts proposed in the literature (see Winter, 2001, for an overview) are definable by the means of second-order logic.

In the next section we recall the mathematical definitions of first-order and second-order generalized quantifiers. We also discuss the notion of definability which is a central concept in generalized quantifier theory. Then we show that collective determiners relate to second-order generalized quantifiers just like distributive determiners relate to first-order generalized quantifiers.

3 Generalized quantifiers

3.1 Lindström quantifiers

Let us first recall the definition of a first-order generalized quantifier formulated by Lindström (1966).

Let $s = (\ell_1, \ldots, \ell_r)$ be a tuple of positive integers. *A first-order generalized (Lindström) quantifier* of type s is a class Q of structures of vocabulary $\tau_s = \{P_1, \ldots, P_r\}$, such that P_i is ℓ_i-ary for $1 \leq i \leq r$, and Q is closed under isomorphisms.

To illustrate the notion let us look at some well-known examples of first-order generalized quantifiers.

$$\begin{aligned}
\forall &= \{(M, P) \mid P = M\}. \\
\exists &= \{(M, P) \mid P \subseteq M \text{ and } P \neq \emptyset\}. \\
Q_{\text{even}} &= \{(M, P) \mid P \subseteq M \text{ and } \operatorname{card}(P) \text{ is even}\}. \\
\mathsf{Most}^1 &= \{(M, P, S) \mid P, S \subseteq M \text{ and } \operatorname{card}(P \cap S) > \operatorname{card}(P \setminus S)\}. \\
\mathsf{Some} &= \{(M, P, S) \mid P, S \subseteq M \text{ and } P \cap S \neq \emptyset\}.
\end{aligned}$$

The first two examples are the standard first-order universal and existential quantifiers, both of type (1). The other examples are also familiar from natural language semantics. Their aim is to capture the truth-conditions of sentences of the form: "Even number of A's are B", "Most A's are B" and "Some A is B". Divisibility quantifier Q_{even} is of type (1), whereas the quantifiers Most^1 and Some are of type (1, 1).

First-order generalized quantifiers enable us to enrich the expressive power of first-order logic in a very controlled and minimal way. We define the extension, FO(Q), of first-order logic by a quantifier Q in the following way:

- The formula formation rules of FO are extended by the rule:

  if for $1 \leq i \leq r$, $\varphi_i(\overline{x}_i)$ is a formula and $\overline{x}_i$ is an ℓ_i-tuple of pairwise distinct variables, then $\mathsf{Q}\,\overline{x}_1, \ldots, \overline{x}_r\,(\varphi_1(\overline{x}_1), \ldots, \varphi_r(\overline{x}_r))$ is a formula.

- The satisfaction relation of FO is extended by the rule:

$$\mathbb{M} \models \mathsf{Q}\,\overline{x}_1, \ldots, \overline{x}_r\,(\varphi_1(\overline{x}_1), \ldots, \varphi_r(\overline{x}_r)) \text{ iff } (M, \varphi_1^{\mathbb{M}}, \ldots, \varphi_r^{\mathbb{M}}) \in \mathsf{Q},$$

  where $\varphi_i^{\mathbb{M}} = \{\overline{a} \in M^{\ell_i} \mid \mathbb{M} \models \varphi_i(\overline{a})\}$.

First-order generalized quantifiers have been used extensively in formal-semantics of natural language to model distributive determiners (see Westerståhl and Peters, 2006). However, they are not adequate in formalizing collective quantification. In the next section we present an intuitive and natural extension of Lindström quantifiers, second-order generalized quantifiers. They turn out to be a natural concept for interpreting the meanings of collective determiners in natural language. Moreover, this concept is consistent with the principles of atomicity, completeness, and atoms discussed in the introduction. We begin with the formal definitions.

3.2 Second-order generalized quantifiers

Second-order generalized quantifiers were first defined and applied in the context of descriptive complexity theory by Burtschick and Vollmer (1998). The general notion of a second-order generalized quantifier was later formulated by Andersson (2002). The following definition is a straightforward generalization from the first-order case. However, note that the types of second-order generalized quantifiers are more complicated than the types of first-order generalized quantifiers, since predicate variables can have different arities. Let $t = (s_1, \ldots, s_w)$, where $s_i = (\ell_1^i, \ldots, \ell_{r_i}^i)$, be a tuple of tuples of positive integers, for $1 \leq i \leq w$. A *second order structure* of type t is a structure of the form $(M, P_1, \ldots, P_w)$, where $P_i \subseteq \mathcal{P}(M^{\ell_1^i}) \times \cdots \times \mathcal{P}(M^{\ell_{r_i}^i})$. Below, we write $f[A]$ for the image of A under the function f.

A *second-order generalized quantifier* $\mathcal{Q}$ of type t is a class of structures of type t such that $\mathcal{Q}$ is closed under isomorphisms: if $(M, P_1, \ldots, P_w) \in \mathcal{Q}$ and $f\colon M \to N$ is a bijection such that $S_i = \{(f[A_1], \ldots, f[A_{r_i}]) \mid (A_1, \ldots, A_{r_i}) \in P_i\}$, for $1 \leq i \leq w$, then $(N, S_1, \ldots, S_w) \in \mathcal{Q}$.

The following examples show that second-order generalized quantifiers are a natural extension from the first-order case.

$$
\begin{aligned}
\exists^2 &= \{(M, P) \mid P \subseteq \mathcal{P}(M) \text{ and } P \neq \emptyset\}. \\
\mathsf{Even} &= \{(M, P) \mid P \subseteq \mathcal{P}(M) \text{ and } \operatorname{card}(P) \text{ is even}\}. \\
\mathsf{Even}' &= \{(M, P) \mid P \subseteq \mathcal{P}(M) \text{ and } \forall X \in P(\operatorname{card}(X) \text{ is even})\}. \\
\mathsf{Most}^2 &= \{(M, P, S) \mid P, S \subseteq \mathcal{P}(M) \text{ and } \operatorname{card}(P \cap S) > \operatorname{card}(P \setminus S)\}.
\end{aligned}
$$

The first example is the familiar unary second-order existential quantifier. The type of $\exists^2$ is $((1))$. The quantifier Even says that a formula holds for an even number of subsets of the universe. On the other hand, the quantifier Even' says that all the subsets satisfying a formula have an even number of elements. The quantifier Most^2 applies to two formulas ψ and φ and says that more than half of the subsets satisfying ψ also satisfy φ.

As in the first-order case, we define the extension, $\mathrm{FO}(\mathcal{Q})$, of FO by a second-order generalized quantifier $\mathcal{Q}$ in the following way:

- Second order variables are introduced to FO.
- The formula formation rules of FO are extended by the rule:

  if for $1 \leq i \leq w$, $\varphi_i(\overline{X}_i)$ is a formula and $\overline{X}_i = (X_{1,i}, \ldots, X_{r_i,i})$ is a tuple of pairwise distinct predicate variables, such that $\operatorname{arity}(X_{j,i}) = \ell^i_j$, for $1 \leq j \leq r_i$, then

  $$\mathcal{Q}\overline{X}_1, \ldots, \overline{X}_w\, (\varphi_1(\overline{X}_1), \ldots, \varphi_w(\overline{X}_w))$$

  is a formula.
- Satisfaction relation of FO is extended by the rule:

  $$\mathbb{M} \models \mathcal{Q}\overline{X}_1, \ldots, \overline{X}_w\, (\varphi_1, \ldots, \varphi_w) \text{ iff } (M, \varphi_1^{\mathbb{M}}, \ldots, \varphi_w^{\mathbb{M}}) \in \mathcal{Q},$$

  where $\varphi_i^{\mathbb{M}} = \{\overline{R} \in \mathcal{P}(M^{\ell^i_1}) \times \cdots \times \mathcal{P}(M^{\ell^i_{r_i}}) \mid \mathbb{M} \models \varphi_i(\overline{R})\}$.

3.3 Definability

The concept of *definability* is central in generalized quantifier theory. Informally, definability of a quantifier Q in a logic $\mathcal{L}$ means that there is a uniform way to express every formula of the form $\mathsf{Q}\,x\,\varphi$ in $\mathcal{L}$.

Formally, let Q be a first-order generalized quantifier of type s and $\mathcal{L}$ a logic. We say that the quantifier Q is *definable* in $\mathcal{L}$ if there is a sentence $\varphi \in \mathcal{L}$ of vocabulary τ_s such that for any τ_s-structure $\mathbb{M}$:

$$\mathbb{M} \models \varphi \Leftrightarrow \mathbb{M} \in \mathsf{Q}\,.$$

Let $\mathcal{L}$ and $\mathcal{L}'$ be logics. The logic $\mathcal{L}'$ is *at least as strong as* the logic $\mathcal{L}$ ($\mathcal{L} \leq \mathcal{L}'$) if for every sentence $\varphi \in \mathcal{L}$ over any vocabulary there exists a sentence $\psi \in \mathcal{L}'$ over the same vocabulary such that

$$\models \varphi \leftrightarrow \psi.$$

The logics $\mathcal{L}$ and $\mathcal{L}'$ are *equivalent* ($\mathcal{L} \equiv \mathcal{L}'$) if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$.

Below, we assume that the logic $\mathcal{L}$ has the so-called *Substitution Property*, i.e., that the logic $\mathcal{L}$ is closed under substituting predicates by formulas. The following fact is well-known for Lindström quantifiers.

Proposition 3.1. *Let Q be a first-order generalized quantifier and $\mathcal{L}$ a logic. The quantifier Q is definable in $\mathcal{L}$ iff*

$$\mathcal{L}(\mathsf{Q}) \equiv \mathcal{L}.$$

Proof. Since $\mathsf{Q} = \mathrm{Mod}(\varphi)$, where $\varphi = \mathsf{Q}\,\overline{x}_1, \ldots, \overline{x}_r\,(P_1(\overline{x}_1), \ldots, P_r(\overline{x}_r))$ the implication from right to left follows. For the other direction, we use recursively the fact that if φ is the formula which defines Q and $\psi_1(\overline{x}_1), \ldots, \psi_r(\overline{x}_r)$ are formulas of $\mathcal{L}$, then

$$\models \mathsf{Q}\,\overline{x}_1, \ldots, \overline{x}_r\,(\psi_1(\overline{x}_1), \ldots, \psi_r(\overline{x}_r)) \leftrightarrow \varphi(P_1/\psi_1, \ldots, P_r/\psi_r),$$

where the formula on the right is obtained by substituting every occurrence of $P_i(\overline{x}_i)$ in φ by $\psi_i(\overline{x}_i)$. □

In the second-order case, analogous notion of definability can be formulated. We do not give the formal definition here. However, things are not completely analogous to the first-order case. With second-order generalized quantifiers $\mathcal{L}(\mathcal{Q}) \equiv \mathcal{L}$ does not imply that the quantifier $\mathcal{Q}$ is definable in the logic $\mathcal{L}$. The converse implication is still valid.

Proposition 3.2 (Kontinen (2004)). *Let $\mathcal{Q}$ be a second-order generalized quantifier and $\mathcal{L}$ a logic. If the quantifier $\mathcal{Q}$ is definable in $\mathcal{L}$ then*

$$\mathcal{L}(\mathcal{Q}) \equiv \mathcal{L}.$$

Proof. The idea and the proof is analogous to the first-order case. Here we substitute second-order predicates by formulas having free second-order variables. □

In Kontinen (2002) it was shown that the extension $\mathcal{L}^*$ of FO by all first-order generalized quantifiers cannot define the monadic second-order existential quantifier. In other words, the logic $\mathcal{L}^*$, in which all properties of first-order structures can be defined, cannot express in a uniform way that a collection of subsets of the universe is non-empty. This observation can be used to argue for the fact that first-order generalized quantifiers alone are not adequate for formalizing all natural language quantification. For example, as quantifier $\exists^2$ is not definable in $\mathcal{L}^*$, the logic $\mathcal{L}^*$ cannot express the collective reading of sentences like (15).

(15) Some students gathered to play poker.

4 Defining collective determiners

In this section we show that collective determiners can be easily identified with certain second-order generalized quantifiers.

At first sight, there seem to be a problem with identifying the collective determiners with second-order generalized quantifiers; some of the collective determiners discussed have a mixed type $((et)(((et)t)t))$[1]. However, this is not a problem since it is straightforward to extend the definition to allow

[1]Note that the lift dfit of Ben Avi and Winter (2003) turns a first-order quantifier of type $(1,1)$ directly to a second-order quantifier of type $((1),(1))$.

also quantifiers with mixed types. Denote by Some^t the following quantifier of type $(1,(1))$

$$\{(M, P, G) \mid P \subseteq M;\ G \subseteq \mathcal{P}(M) :\ \exists Y \subseteq P(Y \neq \emptyset \text{ and } P \in G)\}.$$

Obviously, we can now express the collective meaning of sentence (15) by the formula (16).

(16) $\mathsf{Some}^t\, x, X[\text{Student}(x), \text{Play}(X)]$.

Analogously, we can define the corresponding second-order quantifier appearing in sentence (8), here as (17).

(17) Five people lifted the table.

We take $\mathsf{Five}^{\mathsf{t}}$ to be the second order-quantifier of type $(1,(1))$ denoting the class:

$$\{(M, P, G) \mid P \subseteq M;\ G \subseteq \mathcal{P}(M) :\ \exists Y \subseteq P(\text{card}(Y) = 5 \text{ and } P \in G)\}.$$

Now we can formalize the collective meaning of (17) by:

(18) $\mathsf{Five}^{\mathsf{t}} x, X[\text{Student}(x), \text{Lift}(X)]$.

Already these simple examples show that it is straightforward to associate with every lifted determiner a mixed second-order generalized quantifier. Also, it easy to see that for any first-order quantifier Q the lifted second-order quantifier Q^t can be uniformly expressed in second-order logic assuming the quantifier Q is also available. In fact, all the lifts discussed in Section 2., and, as far as we know, all proposed in the literature, are definable in second-order logic. This observation can be stated as follows.

Proposition 4.1. *Let* Q *be a first-order quantifier definable in* SO. *Then the second-order quantifiers* Q^{EM}, Q^{dfit} *and* Q^t *are definable in* SO.

Proof. Let us consider the case of Q^{EM}. Let $\psi(x)$ and $\phi(Y)$ be formulas. We want to express $\mathsf{Q}^{EM}\, x, Y(\psi(x), \phi(Y))$ in second-order logic. By the assumption, the quantifier Q can be defined by some sentence $\theta \in \text{SO}[\{P_1, P_2\}]$. We can now use the following formula:

$$\exists Z(\forall x(Z(x) \rightarrow \psi(x)) \wedge (\theta(P_1/\psi(x), P_2/Z) \wedge \phi(Y/Z)).$$

The other lifts can be defined analogously. □

Proposition 4.1 shows that the type shifting strategy cannot take us outside of second-order logic. In the next section we show that it is very unlikely that all collective determiners in natural language can be defined in second-order logic. Our argument is based on the close connection between second-order generalized quantifiers and certain complexity classes in computational complexity theory.

5 Lifting the determiner Most

Let us return to the example sentences (4)–(7) from the introduction. For readability, we repeat (7) here as (19).

(19) Most groups of students have never played Hold'em together.

It is easy to see that (19) can be formalized using the quantifier Most^2 by:

(20) $\mathsf{Most}^2\, X, Y[\text{Students}(X), \neg\text{Play}(Y)]$.

We assume above that the predicates Students(X) and Play(Y) are interpreted as collections of sets of atomic entities of the universe of discourse. Obviously, this is just one possible way of interpreting (19). However, it seems that something like Most^2 is needed in the formalization assuming that Students(X) and Play(Y) are interpreted as collective predicates.

For the sake of argument, let us assume that our formalization of sentence (19) is correct. It is easy to see that the lifts discussed before do not give the intended meaning when applied to the first-order quantifier Most^1. We shall next show that it is unlikely that *any* lift, which can be defined in second-order logic, can do the job. More precicely, we shall show (Theorem 5.1 below) that if the quantifier Most^2 can be lifted from the first-order Most^1 using a lift, which is definable in second-order logic, then something unexpected happens in computational complexity. This result indicates that the type-shifting strategy used to define the collective determiners in the literature is probably not general enough to cover all collective quantification in natural language.

We shall next discuss the complexity theoretic side of our argument. Recall that second-order logic corresponds in the complexity theoretic side to the polynomial hierarchy, PH, (see Stockmeyer, 1977). The polynomial hierarchy is an oracle hierarchy with NP as the building block. If we replace NP by probabilistic polynomial time (PP) in the definition of PH, then we arrive at a class called the counting hierarchy (CH). PP consists of languages L for which there is a polynomial time-bounded nondeterministic Turing machine N such that, for all inputs x, $x \in L$ iff more than half of the computations of N on input x end up accepting. The counting hierarchy can be defined now as follows in terms of oracle Turing machines

(1) $C_0P = \mathrm{PTIME}$,

(2) $C_{\mathrm{k}+1}P = \mathrm{PP}^{C_{\mathrm{k}}P}$,

(3) $\mathrm{CH} = \bigcup_{k \in \mathbb{N}} C_kP$.

It is known that PH is contained in the second level C_2P of CH (see Toda, 1991). The question whether $\mathrm{CH} \subseteq \mathrm{PH}$ is still open.

Now, we can turn to the theorem which is fundamental for our argumentation.

Theorem 5.1. *If the quantifier* Most^2 *is definable in second-order logic, then* $\mathrm{CH} = \mathrm{PH}$ *and* CH *collapses to its second level.*

Proof. The proof is based on the observation Kontinen and Niemistö (2006) that already the logic $\mathrm{FO}(\mathsf{Most}^2)$ can define complete problems for each level of the counting hierarchy. On the other hand, if the quantifier Most^2 is definable in second-order logic, then by Proposition 3.2 we would have that $\mathrm{FO}(\mathsf{Most}^2) \leq \mathrm{SO}$ and therefore SO would contain complete problems for each level of the counting hierarchy. This would imply that $\mathrm{CH} = \mathrm{PH}$ and furthermore that $\mathrm{CH} \subseteq \mathrm{PH} \subseteq C_2P$. □

Note that even if $\mathrm{CH} = \mathrm{PH}$ would be true, this does not automatically imply that the quantifier Most^2 can be defined in second-order logic. In fact, it would be very surprising if this would be the case.

6 Conclusion

We have showed that the higher-order approach to collective quantification in natural language can be formalized in terms of second-order generalized quantifiers. The previous attempts have relied implicitly on quantifiers which can be defined in second-order logic. We have presented an argument indicating that second-order definable quantifiers are probably not general enough to cover all collective quantification in natural language.

Acknowledgments

We would like to thank Johan van Benthem and Yoad Winter for valuable comments and suggestions. The first author was financially supported by the Academy of Finland, project 106300. The second author is a recipient of the 2007 Foundation for Polish Science Grant for Young Scientists and was supported by a Marie Curie Early Stage Research fellowship in the project GloRiClass (MEST-CT-2005-020841).

References

Andersson, A. (2002). On second-order generalized quantifiers and finite structures. *Annals of Pure Applied Logic*, 115(1–3):1–32.

Bartsch, R. (1973). The semantics and syntax of number and numbers. In Kimball, J., editor, *Syntax and Semantics*, pages 51–93. New York.

Ben Avi, G. and Winter, Y. (2003). Monotonicity and collective quantification. *Journal of Logic, Language and Information*, 12:127–151.

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

J. van Benthem (1991). *Language in action.* Amsterdam.

Burtschick, H.-J. and Vollmer, H. (1998). Lindström quantifiers and leaf language definability. *International Journal of Foundations of Computer Science*, 9(3):277–294.

J. van der Does (1992). *Applied quantifier logics.* PhD thesis, Universiteit van Amsterdam.

J. van der Does (1996). Sums and quantifiers. *Linguistics and Philosophy*, 16:509–550.

Hella, L., Väänänen, J., and Westerståhl, D. (1997). Definability of polyadic lifts of generalized quantifiers. 6:305–335.

Kontinen, J. (2002). Second-order generalized quantifiers and natural language. In Nissim, M., editor, *Proceedings of the Seventh ESSLLI Student Session*, pages 107–118.

Kontinen, J. (2004). Definability of second order generalized quantifiers. Technical Report 380, Department of Mathematics and Statistics, University of Helsinki. To appear in Archive for Mathematical Logic.

Kontinen, J. and Niemistö, H. (2006). Extensions of MSO and the monadic counting hierarchy. Manuscript available at http://www.helsinki.fi/ jkontine/.

Lindström, P. (1966). First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195.

Link, G. (1983). The logical analysis of plural and mass terms: a lattice–theoretical approach. In Bäuerle, R., Schwarze, C., and von Stechov, A., editors, *Meaning, use, and interpretation of language*, pages 301–323. Berlin.

Link, G. (1991). Plural. In Wunderlich, D. and von Stechov, A., editors, *Semantik. Semantics.* Walter de Gruter, Berlin.

Lønning, J. T. (1997). Plurals and collectivity. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 1009–1053. Elsevier.

McMillan, C., Clark, R., Moore, P., Devita, C., and Grossman, M. (2005). Neural basis for generalized quantifiers comprehension. *Neuropsychologia*, 43:1729–1737.

Mostowski, M. and Szymanik, J. (2005). Semantical bounds for everyday language. *ILLC Preprint Series PP-2006-40*, To appear in *Semiotica*.

Mostowski, M. and Wojtyniak, D. (2004). Computational complexity of the semantics of some natural language constructions. 127:219–227.

Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In Bäuerle, R., Schwarze, C., and von Stechov, A., editors, *Meaning, use, and interpretation of language*, pages 361–383. Berlin.

Russell, B. (1903). *The Principles of Mathematics.* London.

Scha, R. (1981). Distributive, collective and cumulative quantification. In Groenendijk, J., Janssen, T., and Stokhof, M., editors, *Formal methods in the study of language*, pages 483–512. Amsterdam.

Sevenster, M. (2006). *Branching Imperfect Information. Logic, Language, and Computation.* PhD thesis, Institute for Logic, Language, and Computation.

Stockmeyer, L. (1977). The polynomial–time hierarchy. *Theoretical Computer Science*, 3:535–555.

Szymanik, J. (2007). A comment on a neuroimaging study of natural language quantifier comprehension. *Neuropsychologia*, 45(9):2158–2160.

Toda, S. (1991). PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877.

Westerståhl, D. and Peters, S. (2006). *Quantifiers in Language And Logic.* Oxford University Press, Oxford.

Winter, Y. (2001). *Flexibility principles in Boolean semantics.* The MIT Press, London.