Quantifiers in TIME and SPACE : computational complexity of generalized quantifiers in natural language

Szymanik, J.K.

Publication date
2009

Chapter 5

Complexity of Collective Quantification

Most of the efforts in generalized quantifier theory focus on distributive readings of natural language determiners. In contrast — as properties of plural objects are becoming more and more important in many areas (e.g. in game-theoretical investigations, where groups of agents act in concert) — this chapter is devoted to collective readings of quantifiers. We focus mainly on definability issues, but we also discuss some connections with computational complexity.

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 determiners systematically from the standard meanings of quantifiers. We discuss the existential modifier, neutral modifier, and determiner fitting operators as examples of collective lifts considered in the literature. Then we show that all these lifts turn out to be definable in second-order logic.

Next, we turn to a discussion of so-called second order generalized quantifiers — an extension of Lindström quantifiers to second-order structures. We show possible applications of these quantifiers in capturing the semantics of collective determiners in natural language. We also observe that using second-order generalized quantifiers is an alternative to the type-shifting approach.

Then we study the collective reading of the proportional quantifier "most". We define a second-order generalized quantifier corresponding to this reading and show that it is probably not definable in second-order logic. If it were definable then the polynomial hierarchy in computational complexity theory would collapse; this is very unlikely and commonly believed to be false, although no proof is known.

Therefore, probably there is no second-order definable lift expressing the collective meaning of the quantifier "most". This is clearly a restriction of the type-shifting approach. One possible alternative would be to use second-order generalized quantifiers in the study of collective semantics. However, we notice that the computational complexity of such approach is excessive and hence it is not a plausible model (according to the Σ_1^1-thesis formulated in Section 1.8) of collective quantification in natural language. Hence, we suggest to turn in the direction of another well-known way of studying collective quantification, the many-sorted (algebraic) tradition. This tradition seems to overcome the weak points of the higher-order approach.

Another interpretation of our results might be that computational complexity restricts the expressive power of everyday language (see discussion in Section 1.8). Namely, even though natural language can in principle realize collective proportional quantifiers its everyday fragment does not contain such constructions due to their high complexity.

The main observations of this chapter are the result of joint work with Juha Kontinen (see Kontinen and Szymanik, 2008).

5.1 Collective Quantifiers

5.1.1 Collective Readings in Natural Language

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, discussed in letters between Frege and Russel in 1902 (see Frege, 1980):

(1) Bunsen and Kirchoff laid the foundations of spectral theory.

(2) The Romans conquered Gaul.

Sentence (1) does not say that Bunsen laid the foundations of spectral theory and that Kirchoff did also, even though they both contributed. It rather says that they did it together, that spectral theory was a result of a group (team) activity by Kirchoff and Bunsen. Similarly, sentence (2) claims that the Romans conquered Gaul together. For a contrast in meaning compare sentence (1) to (3) and sentence (2) to (4).

(3) Armstrong and Aldrin walked on the moon.

(4) Six hundred tourists visited the Colosseum.

Sentence (3) says that Armstrong walked on the moon and Aldrin walked on the moon. Similarly, sentence (4) is still true if six hundred tourists visited the Colosseum separately.

Notice that not all plural noun phrases are read collectively. Let us consider the following sentences:

(5) Some boys like to sing.

(6) All boys like to sing.

(7) $\exists^{\geq 2}x[\text{Boy}(x) \wedge \text{Sing}(x)]$.

(8) $\forall x[\text{Boy}(x) \implies \text{Sing}(x)]$.

The interpretations of sentences (5)–(6) can be expressed using standard first-order distributive quantifiers, by formulae (7) and (8) respectively.

Therefore, linguistic theory should determine what triggers collective readings. In fact this is to a large degree determined by a context. Noun phrases which can be read collectively may also be read distributively. Compare sentence (1) with proposition (9):

(1) Bunsen and Kirchoff laid the foundations of spectral theory.

(9) Cocke, Younger and Kasami discovered (independently) the algorithm which determines whether a string can be generated by a given context-free grammar.

However, there are so-called collective properties which entail some sort of collective reading; for example the phrases emphasized in the following sentences usually trigger a collective reading:

(10) All the Knights but King Arthur *met in secret*.

(11) Most climbers *are friends*.

(12) John and Mary *love each other*.

(13) The samurai *were twelve in number*.

(14) Many girls *gathered*.

(15) Soldiers *surrounded* the Alamo.

(16) Tikitu and Samson *lifted* the table.

5.1.2 Modelling Collectivity

The Algebraic Approach

When we have decided that a noun phrase has a collective reading the question arises how we should 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). According to many, a good semantic theory for collectives should obey the following intuitive principles:

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.

Among structures satisfying these requirements are many-sorted (algebraic) models, e.g. complete atomic join semilattices (see Link, 1983). The idea — often attributed to Frege and Leśniewski (see e.g. Lønning, 1997, pages 1028–1029) — is roughly to replace the domain of discourse, which consists of entities, with a structure containing also collections of entities. The intuition lying behind this way of thinking is as follows.

Consider the following question and two possible answers to it:

(17) Who played the game?

(18) John did.

(19) The girls did.

Both answers, (18) and (19), are possible. It suggests that a plural noun phrase like "the girls" should denote an object of the same type as "John", and a verb phrase like "played the game" should have one denotation which includes both individuals and collections. The algebraic approach to collectives satisfies this intuition. One of the advantage of this algebraic perspective is that it unifies the view on collective predication and predication involving mass nouns (see e.g. Lønning, 1997, Chapter 4.6).

Algebraic models come with formal languages, like many sorted first-order logic, i.e. a first-order language for plurals. The first sort corresponds to entities and the second one to collections. Such logics usually contain a pluralization operator turning individuals into collections (see e.g. Lønning, 1997, for more details). A similar approach is also adopted in Discourse Representation Theory to account not only for the meaning of collective quantification but also for anaphoric links (see Kamp and Reyle, 1993, Chapter 4).

The Higher-order Approach

From the extensional perspective all that can be modeled within algebraic models can be done in type theory as well (see e.g. van der Does, 1992). This tradition, starting with the works of Bartsch (1973) and Bennett (1974), uses extensional type-theory with two 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 independently by van Benthem (1983) (see also van Benthem, 1995) and Partee and Rooth (1983), this gives a way to model collectivity in natural language. The strategy — introduced by Scha (1981) and later advocated and developed by van der Does (1992, 1993) and Winter (2001) — is to lift first-order generalized quantifiers to a second-order setting. In type-theoretical terms the trick is to shift determiners of type $((et)((et)t))$ (corresponding to relations between sets), related to the distributive readings of quantifiers, into determiners of type $((et)(((et)t)t))$ (relations between sets and collections of sets) which can be used to formalize the collective readings of quantifiers.

In the next section we describe the type-shifting approach to collectivity in a more systematic and detailed way. Then we introduce 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.

5.2 Lifting First-order Determiners

5.2.1 Existential Modifier

Let us consider the following sentences involving collective quantification:

(20) At least five people lifted the table.

(21) Some students played poker together.

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

The distributive reading of sentence (20) claims that the total number of students who lifted the table on their own is at least five. This statement can be formalized in elementary logic by formula (23):

(23) $\exists^{\geq 5}x[\text{People}(x) \wedge \text{Lift-the-table}(x)]$.

The collective interpretation of sentence (20) claims that there was a collection of at least five students who jointly lifted the table. This can be formalized by shifting formula (23) to the second-order formula (24), where the predicate "Lift" has been shifted from individuals to sets:

(24) $\exists X[\text{Card}(X) = 5 \wedge X \subseteq \text{People} \wedge \text{Lift-the-table}(X)]$.

In a similar way, by lifting the corresponding first-order determiners, we can express the collective readings of sentences (21)–(22) as follows:

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

(26) $\forall X[X \subseteq \text{Cards} \implies \text{Lose-in-some-situation}(X)]$.

All the above examples can be described in terms of a uniform procedure of turning a determiner of type $((et)((et)t))$ into a determiner of type $((et)(((et)t)t))$ by means of a type-shifting operator introduced by van der Does (1992) and called the existential modifier, $(\cdot)^{EM}$.

5.2.1. Definition. Let us 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 type (1, 1) quantifier Q in the following way:

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

■

One can observe that the collective readings of sentences (20)–(22) discussed above agree with the interpretation predicted by the existential modifier.

We can now ask about the monotonicity of collective quantification defined via the existential lift. First of all, notice that the existential lift works properly only for right monotone increasing quantifiers. For instance, the sentence:

(27) No students met yesterday at the coffee shop.

with the ↓MON↓ quantifier No gets a strange interpretation under the existential modifier.

The existential modifier predicts that this sentence is true if and only if the empty set of students met yesterday at the coffee shop, which is clearly not what sentence (27) claims. This is because No^{EM} is ↑MON↑ but the collective interpretation of No should remain ↓MON↓, as sentence (27) entails both (28) and (29):

(28) No left-wing students met yesterday at the coffee shop.

(29) No students met yesterday at the "Che Guevara" coffee shop.

This is the so-called van Benthem problem for plural quantification (see van Benthem, 1986, pages 52–53): any general existential lift, like $(\cdot)^{EM}$, will be problematic as it turns any $((et)((et)t))$ determiner into a $((et)(((et)t)t))$ determiner that is upward monotone in the right argument. Obviously, this is problematic with non-upward monotone determiners.

5.2.2. Example. Consider the following sentence with a non-monotone quantifier and the reading obtained by applying the existential lift:

(30) Exactly 5 students drank a whole keg of beer together.

(31) $(\exists^{=5})^{EM}[\text{Student}, \text{Drink-a-whole-keg-of-beer}]$.

Formula (31) is true if and only if the following holds:

$$\exists A \subseteq \text{Student}[\text{card}(A) = 5 \wedge \text{Drink-a-whole-keg-of-beer}(A)].$$

This would yield truth as the logical value of sentence (30) even if there were actually six students drinking a keg of beer together. Therefore, it fails to take into account the total number of students who drank a keg of beer.

5.2.2 The Neutral Modifier

Aiming to solve problems with the collective reading of downward monotone quantifiers, like in sentence (27), van der Does (1992) proposed the so-called neutral modifier, $(\cdot)^N$.

5.2.3. Definition. Let U be a universe, $X \subseteq U$, $Y \subseteq \mathcal{P}(U)$, and Q a type (1, 1) quantifier. We define the *neutral modifier*:

$$\mathsf{Q}^N[X, Y] \text{ is true} \iff \mathsf{Q}\big[X, \bigcup(Y \cap \mathcal{P}(X))\big].$$

■

The neutral modifier can easily account for sentences with downward monotone quantifiers, like proposition (27). But what about non-monotone quantifiers, for example sentence (30)? Now we can express its meaning in the following way:

(32) $(\exists^{=5})^N[\text{Student}, \text{Drink-a-whole-keg-of-beer}]$.

This analysis requires that the total number of students in sets of students that drank a keg of beer together is five. Formula (32) is true whenever:

$$\text{card}\big(\{x | \exists A \subseteq \text{Student}[x \in A \wedge \text{Drink-a-whole-keg-of-beer}(A)]\}\big) = 5.$$

However, it does not require that there was one set of five students who drank a whole keg of beer together: in a situation where there were two groups of students, containing three and two members, sentence (30) would be true according to formula (32). Again, this is not something we would expect, because the collective reading triggered by "together" suggests that we are talking about one group of students.

In general, the following is true about monotonicity preservation under the neutral lift (see Ben-Avi and Winter, 2003, Fact 7):

5.2.4. Fact. Let Q be a distributive determiner. If Q belongs to one of the classes $\uparrow$MON$\uparrow$, $\downarrow$MON$\downarrow$, MON$\uparrow$, MON$\downarrow$, then the collective determiner Q^N belongs to the same class. Moreover, if Q is conservative and $\sim$MON (MON$\sim$), then Q^N is also $\sim$MON (MON$\sim$).

5.2.3 The Determiner Fitting Operator

To overcome problems with non-monotone quantifiers Winter (2001) combined the existential and the neutral modifiers into one type-shifting operator called dfit, abbreviating determiner fitting. The $(\cdot)^{dfit}$ operator turns a determiner of type $((et)((et)t))$ into a determiner of type $(((et)t)(((et)t)t))$.

5.2.5. Definition. For all $X, Y \subseteq \mathcal{P}(U)$ and a type (1, 1) quantifier Q we define the *determiner fitting operator*:

$$\mathsf{Q}^{dfit}[X, Y] \text{ is true}$$

$$\Longleftrightarrow$$

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

■

Using dfit we get the following interpretation of sentence (30):

(33) $(\exists^{=5})^{dfit}[\text{Student}, \text{Drink-a-whole-keg-of-beer}]$.

Formula (33) assigns a satisfactory meaning to sentence (30). It says that exactly five students participated in sets of students drinking a whole keg of beer together and moreover that there was a set of 5 students who drank a keg of beer together. It is true if and only if:

$$\begin{aligned}\text{card}\big(\{x \in A | A \subseteq \text{Student} \wedge \text{Drink-a-whole-keg-of-beer}(A)\}\big) = 5 \\ \wedge\, \exists W \subseteq \text{Student}[\text{Drink-a-whole-keg-of-beer}(W) \wedge \text{card}(W) = 5].\end{aligned}$$

Moreover, notice that the determiner fitting operator will assign the proper meaning also to downward and upward monotone sentences, like:

(27) No students met yesterday at the coffee shop.

(34) Less than 5 students ate pizza together.

(35) More than 5 students ate pizza together.

For sentence (27) the determiner fitting operator does not predict that the empty set of students met at the coffee shop as the existential modifier does. It simply does not demand existence of a witness set in cases when the intersection of arguments is empty. For the downward monotone sentence (34) the first conjunct of the determiner fitting lift counts the number of students in the appropriate collections and guarantees that they contain not more than five students. In the case of the upward monotone sentence (35) the second conjunct of dfit claims existence of the witness set. Table 5.1 adapted from (Ben-Avi and Winter, 2004) sums up the monotonicity behavior of determiner fitting.

Monotonicity of Q	Monotonicity of Q^{dfit}	Example
↑MON↑	↑MON↑	Some
↓MON↓	↓MON↓	Less than five
↓MON↑	∼MON↑	All
↑MON↓	∼MON↓	Not all
∼MON∼	∼MON∼	Exactly five
∼MON↓	∼MON↓	Not all and less than five
∼MON↑	∼MON↑	Most
↓MON∼	∼MON∼	All or less than five
↑MON∼	∼MON∼	Some but not all

Table 5.1: Monotonicity under the determiner fitting operator.

5.2.4 A Note on Collective Invariance Properties

We have briefly discussed the monotonicity properties of every lift (for more details see Ben-Avi and Winter, 2003, 2004). What about other invariance properties (see Section 2.2.5) in the collective setting?

Let us for example consider conservativity. Recall from Section 2.2.5 that a distributive determiner of type (1, 1) is conservative if and only if the following holds for all M and all $A, B \subseteq M$:

$$\mathsf{Q}_M[A, B] \iff Q_M[A, A \cap B].$$

In that sense every collective quantifier Q^{EM} trivially does not satisfy conservativity, as for every X, Y the intersection $X \cap \mathcal{P}(Y) = \emptyset$. Therefore, for every Z $Z \notin X \cap \mathcal{P}(Y)$.

In the case of the neutral lift and determiner fitting we can conclude the same because of a similar difficulty.

5.2.6. Fact. For every Q the collective quantifiers Q^{EM}, Q^{N}, and Q^{dfit} are not conservative.

The failure of this classical invariance is for technical reasons but still we feel that in the intuitive sense the collective quantifiers defined by these lifts satisfy conservativity. To account for this intuition let us simply reformulate the conservativity property in the collective setting (see also Chapter 5 in van der Does, 1992).

5.2.7. Definition. We say that a collective determiner Q of type $((et)(((et)t)t))$ satisfies *collective conservativity* iff the following holds for all M and all $A, B \subseteq M$:

$$\mathsf{Q}_M[A, B] \iff Q_M[A, \mathcal{P}(A) \cap B].$$

■

Do lifted quantifiers satisfy collective conservativity? The following fact gives a positive answer to this question.

5.2.8. Fact. For every Q the collective quantifiers Q^{EM}, Q^{N}, and Q^{dfit} satisfy collective conservativity.

Notice that collective conservativity is satisfied by our lifts no matter whether the distributed quantifier itself satisfies it. Therefore, it is fair to say that conservativity is incorporated into these lifts. We personally doubt that this is a desirable property of collective modelling.

Below we introduce an alternative method of grasping collectivity by means of extending Lindström quantifiers to second-order structures. Among other things our approach does not arbitrarily decide the invariance properties of collective determiners.

5.3 Second-order Generalized Quantifiers

Second-order generalized quantifiers (SOGQs) 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 of second-order generalized quantifiers is a straightforward generalization from the first-order case (see Definition 2.2.1). 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 $s_i = (\ell^i_1, \ldots, \ell^i_{r_i})$, be a tuple of tuples of positive integers. A *second order structure* of type t is a structure of the form $(M, P_1, \ldots, P_w)$, where $P_i \subseteq \mathcal{P}(M^{\ell^i_1}) \times \cdots \times \mathcal{P}(M^{\ell^i_{r_i}})$. Below, we write $f[A]$ for the image of A under the function f.

5.3.1. Definition. 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 $(N, S_1, \ldots, S_w) \in \mathcal{Q}$. ■

5.3.2. Convention. In what follows, second-order quantifiers are denoted $\mathcal{Q}$, whereas first-order quantifiers are denoted Q.

5.3.3. Example. The following examples show that second-order generalized quantifiers are a natural extension of the first-order case. While Lindström quantifiers are classes of first-order structures (a universe and its subsets), second-order generalized quantifiers are classes of second-order structures consisting not only of a universe and its subsets, but also of collections of these subsets.

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

The first quantifier is the unary second-order existential quantifier. The type of $\exists^2$ is $((1))$, i.e., it applies to one formula binding one unary second-order variable. The type of the quantifier Even is also $((1))$ and it expresses that a formula holds for an even number of subsets of the universe. On the other hand, the quantifier Even′ expresses that all the subsets satisfying a formula have an even number of elements. The type of the quantifier Most is $((1),(1))$ and it is the second-order analogue of the quantifier Most.

Examples of linguistic applications of second-order generalized quantifiers are given in the next section. However, already now we can notice that one can think about second-order generalized quantifiers as relations between collections of subsets of some fixed universe. Therefore, also from the descriptive perspective taken in linguistics the notion of second-order generalized quantifiers is a straightforward generalization of Lindström quantifiers.

5.3.4. Convention. Throughout the text we will write Most, Even, Some for Lindström quantifiers and Most, Even, Some for the corresponding second-order generalized quantifiers.

5.3.1 Definability for SOGQs

5.3.5. Definition. As in the first-order case, we define the extension, FO(Q), of FO by a second-order generalized quantifier Q of type $t = (s_1, \ldots, s_w)$, where $s_i = (\ell^i_1, \ldots, \ell^i_{r_i})$, in the following way:

- Second order variables are introduced to the FO language.
- The formula formation rules of FO-language 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

  $$\mathrm{Q}\overline{X}_1, \ldots, \overline{X}_w \left[\varphi_1(\overline{X}_1), \ldots, \varphi_w(\overline{X}_w)\right]$$

  is a formula.

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

$$\mathbb{M} \models \mathrm{Q}\overline{X}_1, \ldots, \overline{X}_w \left[\varphi_1, \ldots, \varphi_w\right] \text{ iff } (M, \varphi_1^{\mathbb{M}}, \ldots, \varphi_w^{\mathbb{M}}) \in \mathrm{Q},$$

where $\varphi_i^{\mathbb{M}} = \left\{\overline{R} \in \mathcal{P}(M^{\ell_1^i}) \times \cdots \times \mathcal{P}(M^{\ell_{r_i}^i}) \mid \mathbb{M} \models \varphi_i(\overline{R})\right\}$.

■

The notion of definability for second-order generalized quantifiers can be formulated as in the case of Lindström quantifiers (see Definition 2.2.10; see also Kontinen (2004) for technical details). However, things are not completely analogous to the first-order case. With second-order generalized quantifiers the equivalence of two logics $\mathcal{L}(\mathrm{Q}) \equiv \mathcal{L}$ does not imply that the quantifier Q is definable in the logic $\mathcal{L}$ (see the next paragraph for an example). The converse implication is still valid.

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

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

Proof The idea and the proof is analogous to the first-order case (see proof of the Proposition 2.2.12). Here we substitute second-order predicates by formulae having free second-order variables. □

Kontinen (2002) has shown that the extension $\mathcal{L}^*$ of first-order logic by all Lindström quantifiers cannot define the monadic second-order existential quantifier, $\exists^2$. 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 result also explains why we cannot add the second implication to the previous proposition. Namely, even though $\mathcal{L}^* \equiv \mathcal{L}^*(\exists^2)$ the quantifier $\exists^2$ is not definable in $\mathcal{L}^*$.

Moreover, 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 the quantifier $\exists^2$ is not definable in $\mathcal{L}^*$, the logic $\mathcal{L}^*$ cannot express the collective reading of sentences like:

(36) Some students gathered to play poker.

Therefore, we need to extend logic beyond $\mathcal{L}^*$ to capture the semantics of collective quantifiers in natural language.

Last but not least, let us notice that natural invariance properties for SOGQs have not been investigated. Studying invariance properties in the context of classical generalized quantifier theory led to many definability results of mathematical and linguistic value. In the case of SOGQs these questions are still waiting for systematic research.

In the next section we will show how to model collectivity by adding second-order generalized quantifiers to elementary logic.

5.4 Defining Collective Determiners by SOGQs

In this section we show that collective determiners can be easily identified with certain second-order generalized quantifiers. Thereby, we explain the notion of second-order generalized quantifiers a little bit more — this time with linguistic examples. We also observe that the second-order generalized quantifiers corresponding to lifted first-order determiners are definable in second-order logic.

Recall that the determiner fitting operator turns a first-order quantifier of type (1, 1) directly into a second-order quantifier of type $((1),(1))$. Nevertheless, at first sight, there seems to be a problem with identifying collective determiners with second-order generalized quantifiers, as the existential and neutral modifiers produce collective noun phrases of a mixed type $((et)(((et)t)t))$, while our Definition 5.3.1 talks only about quantifiers of uniform types. However, this is not a problem since it is straightforward to extend the definition to allow also quantifiers with mixed types. Below we define examples of second-order generalized quantifiers of mixed types which formalize collective determiners in natural language.

5.4.1. Definition. Denote by Some^{EM} the following quantifier of type $(1,(1))$

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

■

Obviously, we can now express the collective meaning of sentence (21), repeated here as sentence (37), by formula (38).

(37) Some students played poker together.

(38) $\mathsf{Some}^{EM}\, x, X[\text{Student}(x), \text{Played-poker}(X)]$.

Analogously, we can define the corresponding second-order quantifier appearing in sentence (20), here as (39).

(39) At least five people lifted the table.

5.4.2. Definition. We take five^{EM} to be the second order-quantifier of type $(1,(1))$ denoting the class:

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

■

Now we can formalize the collective meaning of (39) by:

(40) $\mathsf{five}^{EM} x, X[\mathrm{Student}(x), \mathrm{Lifted\text{-}the\text{-}table}(X)]$.

These simple examples show that it is straightforward to associate a mixed second-order generalized quantifier with every lifted determiner. Also, it is easy to see that for any first-order quantifier Q the lifted second-order quantifiers Q^{EM}, $\mathsf{Q}^{\mathrm{dfit}}$ and Q^{N} can be uniformly expressed in second-order logic assuming the quantifier Q is also available. In fact, all the lifts discussed in Section 5.2, and, as far as we know, all those proposed in the literature, are definable in second-order logic. This observation can be stated as follows.

5.4.3. Theorem. *Let* Q *be a Lindström quantifier definable in second-order logic. Then the second-order quantifiers* Q^{EM}, $\mathsf{Q}^{\mathrm{dfit}}$ *and* Q^{N} *are definable in second-order logic, too.*

Proof Let us consider the case of Q^{EM}. Let $\psi(x)$ and $\phi(Y)$ be formulae. 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 $\theta \in \mathrm{SO}[\{P_1, P_2\}]$. We can now use the following formula:

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

The other lifts can be defined analogously. □

For example, $\mathsf{Some}^{EM}, \mathsf{Most}^{N}, \mathsf{All}^{\mathrm{dfit}}$ are all definable in second-order logic.

Let us notice that the above theorem can be easily generalized to cover not only the three lifts we've discussed but all possible collective operators definable in second-order logic. Namely, using the same idea for the proof we can show the following:

5.4.4. Theorem. *Let us assume that the lift* $(\cdot)^{*}$ *and a Lindström quantifier* Q *are both definable in second-order logic. Then the second-order generalized quantifier* Q^{*} *is also definable in second-order logic.*

These theorems show that in the case of natural language determiners — which are obviously definable in second-order logic — the type-shifting strategy cannot take us outside 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.

5.5 Collective Majority

5.5.1 An Undefinability Result for SOGQ "MOST"

Consider the following sentences:

(41) In the preflop most poker hands have no chance against an Ace and a Two.

(42) Most of the PhD students played Hold'em together.

These sentences can be read collectively. For example, sentence (42) can be formalized using the second-order generalized quantifier Most (defined in Example 5.3.3) by the following formula:

(43) $\textsc{Most}\, X, Y[\text{PhD-Students}(X), \text{Played-Hold'em}(Y)]$.

Above we assume that the predicates PhD-Students and Play-Hold'em are interpreted as collections of sets of atomic entities of the universe. Obviously, this is just one possible way of interpreting sentence (42). In general, we are aware that when it comes to proportional determiners, like "most", it seems difficult to find contexts where they are definitely read collectively (see e.g. Lønning, 1997, p. 1016). On the other hand, we can not totally exclude the possibility that sentences (41)–(42) can be used in a setting where only a collective reading is possible. Anyway, it seems that Most is needed in the formalization, assuming that PhD-Students and Play-Hold'em are interpreted as collective predicates.

For the sake of argument, let us assume that our interpretation of sentence (42) is correct. We claim that the lifts discussed above do not give the intended meaning when applied to the first-order quantifier Most. Using them we could only obtain a reading saying something like "there was a group of students, containing most of the students, such that students from that group played Hold'em together". But what we are trying to account for is the meaning where both arguments are read collectively and which can be expressed as follows "most groups of students played Hold'em together".

We shall next show that it is unlikely that *any* lift which can be defined in second-order logic can do the job. More precisely, we show (Theorem 5.5.1 below) that if the quantifier Most can be lifted from the first-order Most 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 collective determiners in the literature is probably not general enough to cover all collective quantification in natural language.

Let us start by discussing the complexity theoretic side of our argument. Recall that second-order logic corresponds in complexity theory to the polynomial hierarchy, PH, (see Theorem 2.4.5 in the Prerequisites chapter, and Section 2.3.5).

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, (see Section 2.3.6).

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

5.5.1. THEOREM. *If the quantifier* MOST *is definable in second-order logic, then* CH = PH *and* CH *collapses to its second level.*

Proof The proof is based on the observation in Kontinen and Niemistö (2006) that already the extension of first-order logic by the unary second-order majority quantifier, MOST^I, of type ((1)), can define complete problems for each level of the counting hierarchy. The unary second-order majority quantifier is easily definable in terms of the quantifier MOST:

$$\text{MOST}^I[X]\psi \iff \text{MOST}\, X, X[X = X, \psi].$$

Hence, the logic FO(MOST) can define complete problems for each level of the counting hierarchy. On the other hand, if the quantifier MOST were definable in second-order logic, then by Proposition 5.3.6 we would have that FO(MOST) $\leq$ SO and therefore SO would contain complete problems for each level of the counting hierarchy. This would imply that CH = PH and furthermore that CH $\subseteq$ PH $\subseteq C_2P$ (see Toda, 1991). □

5.5.2. COROLLARY. *The type-shifting strategy is probably not general enough to cover all collective quantification in natural language.*

The following conjecture is a natural consequence of the theorem.

5.5.3. CONJECTURE. *The quantifier* MOST *is not definable in second-order logic.*

5.5.2 Consequences of Undefinability

Does SO Capture Natural Language?

Therefore, it is very likely that second-order logic is not expressive enough to capture natural language semantics. Recall that apart from second-order logic, collective quantification in natural language is also not expressible in $\mathcal{L}^*$ — first-order logic enriched by all Lindström quantifiers (see Kontinen, 2002, and Section 5.3.1). Then we have to look for alternative tools. As we have shown, the natural extension of first-order generalized quantifiers beyond elementary structures leads to the notion of second-order generalized quantifiers. We have outlined how one can account for collectivity using second-order generalized quantifiers. The question arises what kind of insight into language can be obtained by introducing second-order generalized quantifiers into semantics. For example, are there any new interesting generalizations or simplifications available in the theory? We can already notice that the van Benthem problem would disappear simply because we would not try to describe semantics of collective quantifiers in terms of some uniform operation but we would rather define separate second-order generalized quantifiers corresponding to their first-order counterparts. Moreover, with SOGQs we do not restrict ourselves to interpretations with already fixed invariance properties (see Section 5.2.4). We can enjoy the same freedom we have in distributive generalized quantifier theory.

Are Many-sorted Models More Plausible?

Theorem 5.5.1 also shows that the computational complexity of some collective sentences can be enormous when analyzed via second-order logic. In other words, such an approach to collective quantification violates the methodological Σ^1_1-thesis from Section 1.8. However, notice that all these claims are valid only if we restrict a semantic theory to universes containing nothing more than entities. But when we are dealing with collective sentences in our everyday communication we rather tend to interpret them in the universe which contains groups of people and combinations of cards as well as people and cards themselves. Theorem 5.5.1 explains this in terms of complexity — it would be too complicated to understand language, thinking about the world as containing only individual objects. From this point of view many-sorted approaches to natural language semantics seem to be closer to linguistic reality and our observation can be treated as an argument in favor of them. In the light of Theorem 5.5.1 we are inclined to believe that this approach is much more plausible than higher-order approaches. It would be interesting to design psychological experiments throwing light on the mental representation of collective quantification. We conjecture that the results of such experiments will show that subjects use some kind of many-sorted logical representation to comprehend collective interpretations of sentences. Experiments can also help to identify gaps in the semantic theory of collectives and motivate and direct the research effort to fill them in.

Does SOGQ "MOST" Belong to Everyday Language?

Last but not least, let us give an alternative interpretation of our result. As we mentioned, the collective meaning of proportional quantifiers like "most" in natural language is marginal at best. It is not completely clear that one can find situations where sentences like (42) have to be read collectively in the suggested way. It might be the case that everyday language does not realize proportional collective quantification, at all, among other reasons due to its extremely high computational complexity. Therefore, we can also argue that from a linguistic point of view there is no need to extend the higher-order approach to proportional quantifiers. Honestly, this is what we would expect, e.g. formulating Σ^1_1-thesis in Section 1.8, but at that point we can not exclude that possibility. However, if there are no such sentences in everyday English then we would say that we have just encountered an example where computational complexity restricts the expressibility of everyday language.

5.6 Summary

In this chapter we have studied the higher-order approach for collective determiners in natural language. In particular, we have considered type-shifting operators: the existential modifier, the neutral modifier and the determiner fitting operator. The research part of this chapter can be summed up in the following way:

- We observed that all these collective lifts are definable in second-order logic.
- Then we introduced second-order generalized quantifiers and proposed them as a tool for modeling collective quantification in natural language.
- Using second-order generalized quantifiers we considered the collective reading of majority quantifiers and proved that it is likely not definable in second-order logic due to its computational complexity. Hence, the type-shifting approach to collectivity in natural language does not obey the Σ_1^1-thesis formulated in Section 1.8.
- Therefore, the collective reading of quantifier "most" probably cannot be expressed using the type-shifting strategy applied to first-order quantifiers.
- In other words, the type-shifting strategy is not general enough to cover all instances of collective quantification in natural language.
- We see a viable alternative in many-sorted (algebraic) approaches which seem to be much easier from the computational complexity point of view and as a result much more psychologically plausible as a model of processing for collective determiners.
- Another possibility is that the collective reading of proportional quantifiers is not realized in everyday language and therefore there is no need for semantic theory to account for it. In that case we would say that computational complexity restricts everyday language expressibility.

Moreover, some natural research problems have appeared in this chapter. Let us mention a few of them:

5.6.1. Question. Does everyday language contain collective interpretations of proportional quantifiers?

5.6.2. QUESTION. What is the exact computational complexity of many-sorted approaches to collectivity in natural language?

5.6.3. QUESTION. What are the natural invariance properties for collective quantification? We noted in Section 5.2.4 that the standard definitions do not have to work properly in the collective context. After formulating empirically convincing invariance properties for collective quantifiers one may want to revise existing theories.

5.6.4. QUESTION. Moreover, the behavior of SOGQs under different invariance properties has not been studied enough. It might be that under some structural properties definability questions among SOGQs might be easier to solve. Obviously, studying definability questions for SOGQs is a natural enterprise from the perspective of collective semantics for natural language.

5.6.5. QUESTION. Finally, is there a purely semantic proof that the quantifier MOST is not definable in second-order logic?