A Treatment of Plurals and Plural Quantifications based on a Theory of Collections

Enrico Franconi
Istituto per la Ricerca Scientifica e Tecnologica (IRST), I-38050 Povo TN, Italy
(franconi@irst.it)

Abstract. Collective entities and collective relations play an important role in natural language. In order to capture the full meaning of sentences like "The Beatles sing 'Yesterday' ", a knowledge representation language should be able to express and reason about *plural entities* – like "the Beatles" – and their relationships – like "sing" – with any possible reading (cumulative, distributive or collective).

In this paper a way of including collections and collective relations within a concept language, chosen as the formalism for representing the semantics of sentences, is presented. A twofold extension of the $\mathcal{ALC}$ concept language is investigated : (1) special relations introduce collective entities either out of their components or out of other collective entities, (2) plural quantifiers on collective relations specify their possible reading. The formal syntax and semantics of the concept language is given, together with a sound and complete algorithm to compute satisfiability and subsumption of concepts, and to compute recognition of individuals.

An advantage of this formalism is the possibility of reasoning and stepwise refining in the presence of scoping ambiguities. Moreover, many phenomena covered by the Generalized Quantifiers Theory are easily captured within this framework. In the final part a way to include a theory of parts (mereology) is suggested, allowing for a lattice-theoretical approach to the treatment of plurals.
Key words: Plurals, Concept Languages, Generalized Quantifiers, Part-whole Relation

1. Introduction

In this paper it is shown how a concept language, i.e. a knowledge representation language of the KL-ONE family – also called Frame-Based Description Languages, Term Subsumption Languages, Terminological Logics, Taxonomic Logics or Description Logics (Woods and Schmolze, 1992) – can be extended in order to represent and reason about *collective entities* or *collections* (Allgayer, 1990; Franconi, 1992). The enriched concept language proposed here is intended to form the semantic and computational means for the representation of plurals and plural quantifiers in natural language; other approaches in the literature include (Allgayer and Franconi, 1992b; Lesmo *et al.*, 1988; Poesio, 1990; Quantz, 1992; Sowa, 1991; Tjan *et al.*, 1992). Although this work has been conceived for concept languages, it can be also applied to other knowledge representation formalisms, such as Conceptual Graphs (Sowa, 1991) and Intensional Propositional Semantic Networks SNePS (Shapiro and Rapaport, 1992).

The importance of concept languages to represent the logical form of a sentence has been elsewhere emphasized (see e.g. (Lavelli *et al.*, 1992)); concept languages have been successfully applied in many working natural language dialogue systems, like XTRA (Allgayer *et al.*, 1989), PENMAN (Bateman *et al.*, 1990), AlFresco (Stock, 1991), JANUS (Weischedel, 1989).

An analysis of plurals in natural language leads us to distinguish among two different categories of plural entities: classes and collections. Classes are involved in sentences like "Men are persons", where the NP "men" is represented by means of the class predicate MAN:

$$\forall x.\, MAN(x) \rightarrow PERSON(x).$$

On the other hand, collections are contingent aggregates of objects, and they should be represented as terms instead of predicates, i.e. they should be interpreted at the same level of individuals as single elements of the domain. For example, the logical form of the sentences "The Beatles are John, Paul, George and Ringo" and "John is the leader of the Beatles" is the following:

$$\ni(beatles,\ paul),\ \ni(beatles,\ john),\ \ni(beatles,\ ringo),\ \ni(beatles,\ george),$$
$$LED\text{-}BY(beatles, john).$$

The plural entity *Beatles* is interpreted as a collection, and in the logical form it does not appear as a predicate, but as a term, at the same level as the objects it is composed by.

In order to give a meaning to the terms denoting collections, a weakened form of Set Theory – called *Collection Theory* – is adopted, and the dangerous leap into a second order theory is avoided. It turns out that the collection theory is more adequate to represent plurals than set theory, because the extensionality principle does not hold. Moreover, interesting connections with the Generalized Quantifiers Theory (Barwise and Cooper, 1981) can be drawn.

Within the collection theory, *plural quantifiers* are introduced, in order to capture the different readings of a relation when applied to a collection. This approach allows for the representation of *ambiguous* readings, so that in the presence of incomplete information a complete reasoning can still be carried on.

A more radical departure from set theory to represent collections is proposed in the last part of this paper, introducing an *non-extensional mereology* (Simons, 1987). In this way, a lattice-theoretical approach for the treatment of plurals as in (Link, 1983) is possible.

The paper is organized as follows. At the beginning the Collection Theory and the Plural Quantifiers are introduced in a generic logical framework. Then it is presented how these theories can be merged into the propositionally complete concept language $\mathcal{ALC}$; many examples will clarify the expressive power of the newly obtained language $\mathcal{ALCS}$. The connections between $\mathcal{ALCS}$ and the Generalized Quantifiers Theory are investigated, and an example will show how the persistency property of symmetric determiners is a simple logical entailment. Then, some attention will be devoted on how the formalism can be used within a natural language dialogue system. Finally, the collection theory is abandoned in favour of a Mereology, i.e. a theory of the part-whole relation. In the appendix an account of the computational properties of $\mathcal{ALCS}$ is given, and a sound and complete decision procedure for a weaker form of the language is devised.

2. The Collection Theory

In this section a simple formal way to model collections of objects is introduced. A *collection* is formed by selecting certain objects, called *members* or *elements* of the collection.

Within this model, both objects and collections of objects are denoted by terms, i.e. they are interpreted as entities of the domain. Like in the standard set theory, a primitive *membership* binary relation (denoted by $\ni$) is introduced, to relate the collective entities with their elements. For example, the formulas

$$\ni(beatles,\ paul),\ \ni(beatles,\ john),\ \ni(beatles,\ ringo),\ \ni(beatles,\ george),$$

are intended to mean that the entities *john*, *paul*, *ringo* and *george* are elements of the collective entity named *beatles*; thus, their conjunction represents the meaning of the sentence "The Beatles are John, Paul, George and Ringo".

A collection can be related with other collections: it can share some components or it can include all the components of other collections. The *sub-collection* and *overlapping* relations between collections are defined by means of the $\ni$ primitive relation:

Definition 1 (Sub-collection and Overlapping)
$\subseteq(a, b)$ iff $\forall x. \ni(a, x) \rightarrow \ni(b, x)$
$\cap(a, b)$ iff $\exists x. \ni(a, x) \wedge \ni(b, x)$.

For example, the formulas

$$\subseteq(beatles,\ appleCharterMembers), \quad \cap(beatles,\ mostPopularSingers)$$

express that each of the *beatles* is also a founder of the Apple Records company, and that some of them are among the most popular singers.
The $\subseteq$ relation is defined as the usual *subset* of the ordinary set theory, whereas the $\cap$ relation is defined to be true for all the non-empty set-theoretic intersections $\cap^s$, i.e. $\cap(a, b)$ if and only if $\cap^s(a, b) \neq \emptyset$.
Since any other axiom for such relations is not introduced, it turns out that the $\subseteq$ and $\cap$ relations have weaker properties than their counterparts in set theory. For example, it follows from the definition that the $\subseteq$ relation is reflexive and transitive but not anti-symmetric – i.e. it is a quasi-ordering – since the extensionality axiom does not hold. In set theory, the extensionality axiom says that two sets are equal if and only if they have the same elements. In our framework, two collections having the same elements are not necessarily equal. So, from

$$\subseteq(appleCharterMembers,\ beatles), \quad \subseteq(beatles,\ appleCharterMembers)$$

does not follow that

$$appleCharterMembers = beatles,$$

because the entity *beatles* could have different attributes from the entity representing the group of people who founded the Apple Records company, like, for instance, legal liability or taxes to pay.

This simple framework will be referred as the *Collection Theory*. A richer theory would include also *negated* relations, such as "$\not\ni$", "$\not\cap$" (disjointness) and "$\not\subseteq$" – see (Wellman and Simmons, 1988); they are needed to properly cover many natural language phenomena. However, it is argued (though not yet proved) that, in the context of concept languages, negated relations may lead to undecidability. As a trivial consequence of this choice, it is worth noting that *paradoxes* – like the Russel paradox – are avoided and well-foundedness of the Collection Theory is guaranteed, if the negation of the $\ni$ relation can not be expressed.

3. Plural Quantifiers

Generic relations which apply to collective entities can be quantified in different ways. So, representational means are introduced to capture the semantics of the (possibly underdetermined) reading variants of NL expressions. As an example, take the possible readings involving the plural subject of the sentence "John, Paul, George and Ringo sing 'Yesterday' ". For the *collective reading* all the men together sing the song; in the case each man sings separately, we speak of the *distributive reading*; and, finally, the *cumulative reading* can describe the mixed situations in which, say, one sings alone, and separately the others sing together, with the proviso that all of them are involved in some action of singing. The $\lhd$ and $\unlhd$ (resp. $\rhd$ and $\unrhd$) operators – called *plural quantifiers* – introduce the left (resp. right) distributive and cumulative readings for generic binary relations having a collection as left (resp. right) argument.

Thus, relationships between collections have a more structured semantics than the standard one. A relation holds not only directly between the objects of predication, but may be distributed between the elements of such objects, if they are collections. This approach helps in delaying the decision for any of those variants of a sentence, allowing for the representation of the scope ambiguities in the logical form. The economy of representation makes it unnecessary for the system to compute all the disambiguated interpretations of the sentence before storing its meaning in the knowledge base (Poesio, 1991). In the case of underdetermined reading, a possible incremental growth of the knowledge base might rule out one or another reading.

In order to introduce the formalism, consider the following example:

"John is the leader of the Beatles"

$\mathit{LED\text{-}BY}(beatles, john)$,

"The Beatles were born in Liverpool"

$\lhd \mathit{BORN\text{-}IN}(beatles,\ liverpool)$,

"The Beatles sing 'Yesterday' "

$\unlhd \mathit{SING}(beatles,\ yesterday)$.

The relation $\mathit{LED\text{-}BY}$ has a "*collective*" reading – i.e. $john$ is the leader of the whole collection $beatles$. The relation $\mathit{BORN\text{-}IN}$ is "*left distributive*" over the components of the $beatles$ – i.e. each member was born in Liverpool:

$\mathit{BORN\text{-}IN}(john,\ liverpool),\ \mathit{BORN\text{-}IN}(paul,\ liverpool),$

$\mathit{BORN\text{-}IN}(george,\ liverpool),\ \mathit{BORN\text{-}IN}(ringo,\ liverpool).$

The relation $SING$ has a "*left cumulative*" reading with respect to the $beatles$ – i.e. it is possible that any collection of components of $beatles$ sings 'Yesterday', ranging from single individuals to the entire $beatles$ collection, with the proviso that the union of such collections should include at least all the $beatles$ members. For example, together with the collective interpretation

$SING(beatles,\ yesterday)$,

and the distributive interpretation

$SING(john,\ yesterday),\ SING(paul,\ yesterday),$

$SING(george,\ yesterday),\ SING(ringo,\ yesterday),$

a possible valid interpretation for the cumulative reading is the following:

$\ni(C_1,\ paul),\ \ni(C_1,\ john),\ \ni(C_1,\ elvis),$

$\ni(C_2,\ paul),\ \ni(C_2,\ george),\ \ni(C_2,\ ringo),$

$SING(C_1,\ yesterday),\ SING(C_2,\ yesterday).$

In this interpretation, the relation $SING$ holds between two collective entities C_1, C_2 and $yesterday$. The 'inclusion' condition of the cumulative reading is satisfied: each member of the $beatles$ belongs to at least one of the collections participating to the relation. The plural quantifiers are formally defined as follows.

Definition 2 (Plural Quantifiers)

$\lhd R(a,b)$ iff $\forall x.\ni(a,x) \rightarrow R(x,b)$

$\rhd R(a,b)$ iff $\forall x.\ni(b,x) \rightarrow R(a,x)$

$\unlhd R(a,b)$ iff $\forall x.\ni(a,x) \rightarrow \big(R(x,b) \vee (\exists s.\ni(s,x) \wedge R(s,b))\big)$

$\unrhd R(a,b)$ iff $\forall x.\ni(b,x) \rightarrow \big(R(a,x) \vee (\exists s.\ni(s,x) \wedge R(a,s))\big)$.

It is easy to check that the cumulative plural quantified expressions subsume the non-quantified – i.e. the collective – expressions and the distributive plural quantified ones:

$$\forall x, y.\ R(x,y) \rightarrow \unlhd R(x,y), \qquad \forall x, y.\ \lhd R(x,y) \rightarrow \unlhd R(x,y),$$
$$\forall x, y.\ R(x,y) \rightarrow \unrhd R(x,y), \qquad \forall x, y.\ \rhd R(x,y) \rightarrow \unrhd R(x,y).$$

In this way, the non-disambiguated logical form of a sentence with multiple interpretations can be represented:

"The Beatles sing 'Yesterday' "

$\unlhd SING(beatles,\ yesterday)$.

Information which comes later in the discourse can monotonically refine the knowledge about the quantifiers scoping. From the sentence "They sing the song all together" the system is able to conclude that *they* is anaphoric to *the Beatles* and *the song* to *'Yesterday'*, and will produce an interpretation which specializes the preceding underdetermined one:

$$SING(b,\ y), \quad SONG(y), \quad b = beatles, \quad y = yesterday,$$

On the other hand, if later in the discourse the sentence "Each one of them sings the song" appears, the system will add the formulas

$$\lhd SING(b,\ y), \quad SONG(y),$$

which are also a refinement with respect to the preceding cumulative interpretation.

This approach is in the direction of (Kempson and Cormack, 1981): it maintains the logical structure of the semantic representation of a sentence as close as possible to its syntactic structure.

Finally, let us consider the Massey's classical example about plurals:

"Tom, Dick and Harry carry the piano."

$\ni(c,\ tom), \quad \ni(c,\ dick), \quad \ni(c,\ harry), \quad CARRY(c,\ p), \quad PIANO(p)$.

"They play it."

$\lhd PLAY(c,\ p)$.

4. The $\mathcal{ALCS}$ concept language

The collection theory is now merged into a larger logic, in order to obtain an expressive, but still decidable, description language. A concept language framework has been chosen; in section 6 the reasons for this choice will be clear, by looking at the use of the concept language as the formalism for representing the meaning of utterances.

With respect to the formal apparatus, I will strictly follow the concept language formalism introduced by (Schmidt-Schauß and Smolka, 1991) and further elaborated for example by (Donini *et al.*, 1992; Donini *et al.*, 1991b;

C, D	$\rightarrow$	$A \mid$	(primitive concept)
		$\top \mid$	(top)
		$\bot \mid$	(bottom)
		$\neg C \mid$	(general complement)
		$C \sqcap D \mid$	(conjunction)
		$C \sqcup D \mid$	(disjunction)
		$\forall R.C \mid$	(universal quantifier)
		$\exists R.C$	(existential quantifier)

R	$\rightarrow$	$P \mid$	(primitive role)
		$\ni \mid$	(has element relation)
		$\subseteq \mid$	(sub-collection relation)
		$\cap \mid$	(overlaps relation)
		$\lhd R \mid$	(left distributive)
		$\rhd R \mid$	(right distributive)
		$\unlhd R \mid$	(left cumulative)
		$\unrhd R$	(right cumulative)

Figure 1. Syntax rules for the $\mathcal{ALCS}$ concept language.

Donini *et al.*, 1991a). Basic types of a concept language are *concepts*, *roles* and *individuals*. A concept is a description gathering the common properties among a collection of individuals; logically it is a unary predicate ranging on the domain of individuals. Properties are represented by means of roles, which are logically binary relations. In the following, we will consider the language $\mathcal{ALCS}$, which extends the propositionally complete concept language $\mathcal{ALC}$ (Schmidt-Schauß and Smolka, 1991), allowing a richer expressivity for roles: the $\ni$, $\subseteq$ and $\cap$ relations are the basic roles of the collection theory; the $\lhd$, $\rhd$, $\unlhd$ and $\unrhd$ plural quantifiers introduce the distributive and cumulative readings for generic roles.

According to the syntax rules of figure 4, $\mathcal{ALCS}$ *concepts* (denoted by the letters C and D) are built out of *primitive concepts* (denoted by the letter A) and *roles* (denoted by the letter R); roles are built out of *primitive roles* (denoted by the letter P).

Usually, concept and role expressions are called *TBox terms*. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta^{\mathcal{I}}$ of individuals (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$) that maps every concept to a subset of $\Delta^{\mathcal{I}}$ and every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that the equations of figure 4 are satisfied.

An interpretation $\mathcal{I}$ is a *model* for a concept C if $C^{\mathcal{I}} \neq \emptyset$. If a concept has a model, then it is *satisfiable*, otherwise it is *unsatisfiable*. A concept C is *subsumed* by a concept D (written $C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every interpretation $\mathcal{I}$. Subsumption can be reduced to satisfiability since C is subsumed by D if and only if $C \sqcap \neg D$ is not satisfiable.

According to the given semantics, the concept *non-empty collection*, denoting any collection having at least one element, and the concept *empty collection*, denoting any entity having no elements, can be defined as follows:

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.\, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.\, (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \\
\subseteq^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \forall x.\, (a,x) \in \ni^{\mathcal{I}} \rightarrow (b,x) \in \ni^{\mathcal{I}}\} \\
\cap^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists x.\, (a,x) \in \ni^{\mathcal{I}} \wedge (b,x) \in \ni^{\mathcal{I}}\} \\
(\lhd R)^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \forall x.\, (a,x) \in \ni^{\mathcal{I}} \rightarrow (x,b) \in R^{\mathcal{I}}\} \\
(\rhd R)^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \forall x.\, (b,x) \in \ni^{\mathcal{I}} \rightarrow (a,x) \in R^{\mathcal{I}}\} \\
(\unlhd R)^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \forall x.\, (a,x) \in \ni^{\mathcal{I}} \rightarrow ((\exists s.\, (s,x) \in \ni^{\mathcal{I}} \wedge (s,b) \in R^{\mathcal{I}}) \vee (x,b) \in R^{\mathcal{I}})\} \\
(\unrhd R)^{\mathcal{I}} &= \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \forall x.\, (b,x) \in \ni^{\mathcal{I}} \rightarrow ((\exists s.\, (s,x) \in \ni^{\mathcal{I}} \wedge (a,s) \in R^{\mathcal{I}}) \vee (a,x) \in R^{\mathcal{I}})\}
\end{aligned}
$$

Figure 2. The semantic interpretation for concepts and roles in $\mathcal{ALCS}$.

$$COLL \doteq \exists\ni.\top, \qquad ECOLL \doteq \forall\ni.\bot = \neg COLL.$$

It is easy to verify the validity of the following statements:

$$
\begin{aligned}
&\forall x, y.\ ECOLL(x) \rightarrow \subseteq(x,y), \\
&\forall x, y.\ (COLL(x) \wedge COLL(y)) \rightarrow (\subseteq(x,y) \rightarrow \cap(x,y)), \\
&\neg\exists x, y.\ \cap(x,y) \wedge (ECOLL(x) \vee ECOLL(y)).
\end{aligned}
$$

These statements match our intuitions about collections, and reflect the corresponding valid axioms in set theory, where $ECOLL$ is interpreted as the empty set and $COLL$ is interpreted as any non-empty set:

$$
\begin{aligned}
&\forall y.\ \subseteq^s(\emptyset, y), \\
&\forall x, y.\ (x \neq \emptyset \wedge y \neq \emptyset) \rightarrow (\subseteq^s(x,y) \rightarrow \cap^s(x,y) \neq \emptyset), \\
&\neg\exists x, y.\ \cap^s(x,y) \neq \emptyset \wedge (x = \emptyset \vee y = \emptyset).
\end{aligned}
$$

Let us introduce now some more complex concept definitions, in order to understand better the expressive power of $\mathcal{ALCS}$.

The concept $(\forall\cap.COLL)$ denotes any entity which possibly has an overlapping with some non-empty collection; the concept $(\forall\subseteq.COLL)$ denotes any entity which possibly contains less elements than some non-empty collection; finally, the concept $(\exists\cap.\top)$ denotes any collection having at least a common element with something else. It can be shown that the first concept is equivalent to the top concept, the second includes in its denotation all the non-empty collections, and the latter denotes only non empty collections:

$$\forall\cap.COLL \equiv \top, \qquad COLL \sqsubseteq \forall\subseteq.COLL, \qquad \exists\cap.\top \sqsubseteq COLL.$$

It is worth noting that expressions containing the $\lhd$ plural quantifier cannot be reformulated in terms of the $\ni$ role only. This is somehow counterintuitive, and surprisingly it can be proved that:

$$\exists(\lhd R).C \neq \forall\ni.(\exists R.C).$$

This can be understood by considering that the concept in the right hand side introduces for each element a possible new relation, whereas the concept in the left hand side introduces the same relation for each element; therefore, the former has a larger denotation. In fact, the best that can be proved is an inclusion relation:

$$\exists(\lhd R).C \sqsubseteq \forall\ni.(\exists R.C).$$

Let us consider now *assertions*, i.e. predications on individual objects; usually, they are referred to as *ABox statements*. Let $\mathcal{O}$ be the alphabet of symbols denoting *individuals*; an assertion is a statement of the form $C(a)$ or $R(a,b)$, where C is a concept, R is a role, and a and b denote individuals in $\mathcal{O}$. In order to assign a meaning to the assertions, the interpretation function $\cdot^{\mathcal{I}}$ is extended to individuals, so that $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ for each individual $a \in \mathcal{O}$, and $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (Unique Name Assumption). The semantics of the assertions is the following: $C(a)$ is satisfied by an interpretation $\mathcal{I}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and $R(a,b)$ is satisfied by $\mathcal{I}$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

A set Σ of assertions is called an *knowledge base*. An interpretation $\mathcal{I}$ is a *model* of Σ iff every assertion of Σ is satisfied by $\mathcal{I}$. If Σ has a model, then it is *satisfiable*. Σ *logically implies* an assertion α (written $\Sigma \models \alpha$) if α is satisfied by every model of Σ.

Given a knowledge base Σ, an individual a in $\mathcal{O}$ is said to be an *instance* of a concept C if $\Sigma \models C(a)$. The *instance recognition problem*, i.e. checking whether $\Sigma \models C(a)$, can be reduced to satisfiability since a is an instance of C with respect to a knowledge base Σ if and only if $\Sigma \cup \{\neg C(a)\}$ is unsatisfiable (Hollunder, 1990).

Coming back to our example regarding the Beatles, let us see how the concept representing any pop group can be defined using the $\mathcal{ALCS}$ language:

$$\begin{aligned} \mathit{POP\text{-}GROUP} \doteq\ & \forall\ni.\mathit{PERSON} \sqcap \forall \mathit{LED\text{-}BY}.\mathit{PERSON} \sqcap \\ & \forall\lhd \mathit{BORN\text{-}IN}.\mathit{CITY} \sqcap \forall\unlhd \mathit{SING}.\mathit{POP\text{-}SONG}. \end{aligned}$$

The definition states that a pop group is composed by persons, that the relation "*led by a person*" is inherently collective with respect to the group, that the relation "*born in a city*" inherently distributes to the single persons composing the group, and that the relation "*sing a pop song*" has a cumulative reading for the group.
From the definition and from the knowledge acquired during the discourse, the system is able to recognize the individual *beatles* as an instance of *POP-GROUP* and is able to classify the *POP-GROUP* as a collection:

$$\mathit{POP\text{-}GROUP}(\mathit{beatles}), \quad \mathit{POP\text{-}GROUP} \sqsubseteq \mathit{COLL}.$$

5. Generalized Quantifiers

So far the language $\mathcal{ALCS}$ has been considered; if qualified number restrictions are added to the language (Hollunder and Baader, 1991), some interesting connections with the Generalized Quantifiers Theory (GQT) (Barwise and Cooper, 1981) can be found. Such language is an extension of the concept language presented in (Allgayer and Franconi, 1992b), where the motivations for natural language applications are investigated. Joachim Quantz in (Quantz, 1992) presents a more general concept language where generalized quantifiers fit well; however, his approach is based on set theory.

The theory of Generalized Quantifiers extends the field of quantification to arbitrary quantitative ties between predications, in order to directly cover some properties of natural language quantifiers. The aim of GQT is to treat all noun phrases (NPs) in a unique way as *quantifiers*, where a quantifier is a pair composed by a determiner and a noun. Take for example the utterance "At most three persons sing 'Yesterday' ". The interpretation of the quantifier (i.e., the NP) "at most three persons" is seen as a set of collections – namely the set of all collections composed by at most three persons –, whereas the interpretation of the verb phrase (VP) "sing 'Yesterday' " is a set of entities – namely all 'Yesterday'-singers. The interpretation of the sentence S = (NP VP) then requires that the interpretation of VP be an element of the interpretation of the NP.

The NP's interpretation mainly depends – besides, of course, the noun – on the determiner used in constructing it. In this way, a determiner can be seen as a functor operating on sets of entities yielding a set of sets of entities. For example, the determiner "at most three" maps the interpretation of "persons", namely the set of all persons, onto the set of those sets of entities consisting of at most three persons. The membership of the set of 'Yesterday'-singers (being the interpretation of the VP of our example) in this collection of sets then decides the validity of the utterance.

Determiners are intrinsically characterized by certain properties; for our purposes only the properties of *symmetry* and *persistency* are introduced. Symmetry and persistency in this context are stated as follows:

Definition 3 (Symmetry and Persistency)

A *determiner* D is a function $D^{\mathcal{I}} : 2^{\Delta^{\mathcal{I}}} \mapsto 2^{2^{\Delta^{\mathcal{I}}}}$.

D is *symmetric* (aka *unary* or *adjectival*) if for all $A, B \subseteq \Delta^{\mathcal{I}}$ s.t. $B \in D^{\mathcal{I}}(A)$ then $A \in D^{\mathcal{I}}(B)$.

D is *persistent* (per↑) if for all $A, B, X \subseteq \Delta^{\mathcal{I}}$ s.t. $A \subseteq B$ and $X \in D^{\mathcal{I}}(A)$ then $X \in D^{\mathcal{I}}(B)$.

D is *anti-persistent* (per↓) if for all $A, B, X \subseteq \Delta^{\mathcal{I}}$ s.t. $A \subseteq B$ and $X \in D^{\mathcal{I}}(B)$ then $X \in D^{\mathcal{I}}(A)$.

We will call the property of being neither persistent nor anti-persistent per•; a determiner can not have the properties per↑ and per↓ at the same time. □

Determiner	Concept Expression	Persistency
some	$(\exists\ni.C)$	per$\uparrow$
no	$(\forall\ni.\neg C)$	per$\downarrow$
at least i	$(\geq i \ni C)$	per$\uparrow$
at most i	$(\leq i \ni C)$	per$\downarrow$
from i to j	$(\leq i \ni C) \sqcap (\geq j \ni C)$	per$\bullet$
exactly i	$(\leq i \ni C) \sqcap (\geq i \ni C)$	per$\bullet$

Figure 3. Persistency property of the logical symmetric natural language determiners and their mapping in the concept language.

In the following only symmetric determiners will be considered, like *some*, cardinal numbers, and the negative existential *no*; non-symmetric quantifiers include *every, each, all, both, neither*. To check if a determiner is symmetric, check the equivalence of sentences obtained by transposing the determiner. As an example, let us consider the sentence "All men are persons": this sentence is not equivalent to "All persons are men", thus *all* is not symmetric. On the other hand, the sentence "At most three persons sing 'Yesterday' " is equivalent to the sentence "At most three 'Yesterday'-singers are persons", making *at most three* a symmetric determiner.

The presence of a quantifier in a sentence can lead to some inferences which are driven by the persistency property of the determiner. According to GQT, by generalizing the NP (i.e., the quantifier) of a valid sentence S = (NP VP), we obtain a valid sentence. A persistent determiner requires a generalization of the noun to generalize the quantifier, whereas an anti-persistent determiner requires a specialization of the noun to generalize the quantifier. In the sentence "At most three persons sing 'Yesterday' ", because of the presence of an anti-persistent determiner (as is *at most three*), the NP "at most three persons" might be generalized to "at most three men", yielding the valid sentence "At most three men sing 'Yesterday' ". In fact, "man" specializes "person". Figure 5 shows the persistency property of the symmetric determiners.

Within the enriched language $\mathcal{ALCS}$ with qualified number restrictions, the property of persistency of symmetric determiners is reflected as a simple consequence. Qualified number restrictions extend the syntax and the semantics of the concept language in the following way:

$$C \;\rightarrow\; (\geq n\, R\, C) \mid (\leq n\, R\, C)$$
$$(\geq n\, R\, C)^{\mathcal{I}} \;=\; \{a \in \Delta^{\mathcal{I}} \mid \quad \| \{b \mid (a,b) \in R^{\mathcal{I}}\} \cap C^{\mathcal{I}} \| \quad \geq n\}$$
$$(\leq n\, R\, C)^{\mathcal{I}} \;=\; \{a \in \Delta^{\mathcal{I}} \mid \quad \| \{b \mid (a,b) \in R^{\mathcal{I}}\} \cap C^{\mathcal{I}} \| \quad \leq n\}$$

Considering the translation of quantifiers into concept expressions of figure 5, our example can be rewritten as

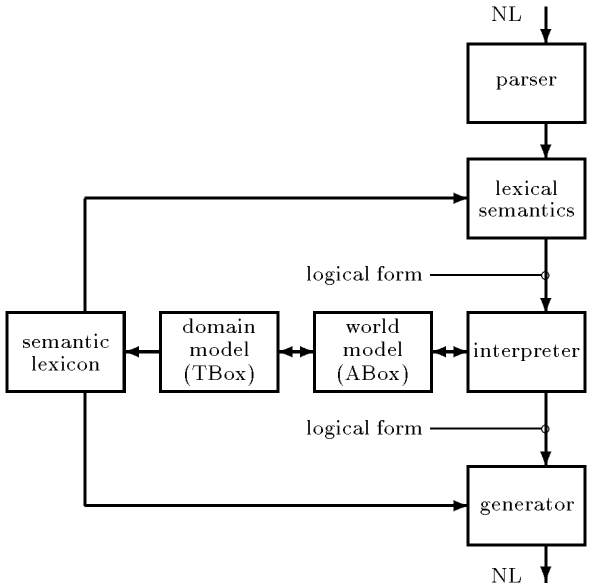

Figure 4. A Natural Language Dialogue Architecture.

$$MAN \sqsubseteq PERSON, \quad \big(\leq 3 \ni PERSON\big)(a), \quad \trianglelefteq SING(a, yesterday),$$

which trivially entails

$$\big(\leq 3 \ni MAN\big)(a);$$

i.e., validity of the sentence "At most three persons sing 'Yesterday' " implies the validity of the sentence "At most three men sing 'Yesterday' ". Moreover, this approach embodies not only the notion of persistency of GQT, but also parts of the notion of the 'Tree of Numbers' theory (cf. (van Benthem, 1986, Sec 2.2)); in fact, a valid entailment is also

$$\big(\leq 4 \ni MAN\big)(a);$$

i.e., validity of the sentence "At most three persons sing 'Yesterday' " implies the validity of the sentence "At most four men sing 'Yesterday' ".

6. Collection Theory and Natural Language Dialogue Systems

The representational framework outlined in this paper is particularly qualified for expressing the logical meaning of natural language utterances containing occurrences of plurals and plural quantifications. In this section I will devote more attention on how this formalism can be used within a natural language dialogue system. In the following the general architecture of figure 6 will be assumed (Lavelli *et al.*, 1992; Stock, 1991).

In this simplified view, the system is able to acquire information from a user and to answer her/his queries. Knowledge about the *abstract domain* – i.e. the meaning of the unary predicates (*types*) and binary predicates (*relations*) – is stored by means of TBox formulas, whereas knowledge about the actual *objects of the world* – the individuals – is stored by means of ABox formulas.

The *logical form*, denoting the meaning of an utterance, is based on ABox expressions. In this sense, the logical form includes unary and binary ground predicates – relating the objects appearing in the discourse – whose import is defined in the TBox domain model.

The meaning of informative sentences uttered by the user specializes and refines the knowledge already present in the world model, if it is consistent with it. The meaning of query sentences drives a reasoning process in order to find the answer matching the request; the newly deduced logical form is then translated into a natural language answer by the system.

An important role in such an architecture is played by the *lexicon*, which gives the appropriate semantic restrictions on the types, the relations and the objects involved in the natural language utterance.

For example, an informative sentence like

`U:` "The Beatles are John, Paul, George and Ringo."

is translated by the lexical semantics module into the logical form

$$\ni(beatles,\ john),\ \ni(beatles,\ paul),\ \ni(beatles,\ george),\ \ni(beatles,\ ringo)$$

which is then added to the world knowledge of the system.
The subsequent query sentence

`U:` "Who are the Beatles?"

is interpreted as

$$\ni(beatles,\ ?X).$$

By unification with the known world knowledge, this yields

$$\ni(beatles,\ john),\ \ni(beatles,\ paul),\ \ni(beatles,\ george),\ \ni(beatles,\ ringo)$$

which is finally translated by the generator into the natural language reply

`S:` "The Beatles are John, Paul, George and Ringo."

In the following a sample dialogue is given, highlighting the peculiarities of the collection theory in the representation of plurals and plural quantification. The possibility of handling plurals and plural quantifications in a *uniform* and *compositional* way is proper of the collection theory. I will not explicitly specify other aspects of meaning interpretation, such as lexical ambiguities,

anaphora resolution, multiple interpretations and so on, which are not specific with respect to the collection theory. Each `user` sentence (either informative or query) is followed by its internal simplified logical form; whereas each sentence generated by the `system` is preceded by its internal simplified logical form deduced from the world knowledge and the user request.
Please observe that objects with anaphoric reference are prefixed with an exclamation note. A topic module, not explicitly mentioned in the architecture, is responsible for anaphora resolution, i.e. for finding the equations denoting the referred objects. The topic module makes an intensive use of reasoning in the TBox and ABox (Samek and Strapparava, 1990; Zancanaro *et al.*, 1993).

`U`: "The Beatles are John, Paul, George and Ringo."
$\ni(beatles,\ john),\ \ni(beatles,\ paul),\ \ni(beatles,\ george),\ \ni(beatles,\ ringo)$

`U`: "Who are the Beatles?"
$\ni(beatles,\ ?X)$

$\ni(beatles,\ john),\ \ni(beatles,\ paul),\ \ni(beatles,\ george),\ \ni(beatles,\ ringo)$
`S`: "The Beatles are John, Paul, George and Ringo."

`U`: "John is the leader of the pop group."
$LED\text{-}BY(!PG, john),\ POP\text{-}GROUP(!PG),\ !PG = beatles$

`U`: "They were born in Liverpool."
$\lhd BORN\text{-}IN(!T,\ liverpool),\ !T = beatles$

`U`: "Where was the leader of the group born?"
$BORN\text{-}IN(?X,\ ?Y),\ LED\text{-}BY(!G, ?X),\ !G = beatles$

$BORN\text{-}IN(john,\ liverpool)$
`S`: "John was born in Liverpool."

`U`: "The Beatles sing 'Yesterday'."
$\unlhd SING(beatles,\ yesterday)$

`U`: "They sing the song all together."
$SING(!T,\ !S),\ SONG(!S),\ !T = beatles,\ !S = yesterday$

`U`: "Does John and somebody else sing 'Yesterday'?"
$\unlhd SING(!E,\ yesterday),\ \ni(!E,\ john),\ \ni(!E,\ ?X),\ PERSON(?X)$

$SING(!E,\ yesterday),\ \ni(!E,\ john),\ \ni(!E,\ paul),\ \ni(!E,\ george),\ \ni(!E,\ ringo)$

S: "John, Paul, George and Ringo sing 'Yesterday' all together."

U: "It is true that at most three persons are singing the song all together?"

$(\leq 3 \ni PERSON)(!F),\ SING(!F,\ !S),\ !S = yesterday$

$\bot \qquad (inconsistency)$

S: "No, it is false."

U: "How many persons at least are singing 'Yesterday'?"

$(\geq\ ?X \ni PERSON)(!Q),\ SING(!Q,\ yesterday),\ \ni(!Q,\ ?Y)$

$(\geq 4 \ni PERSON)(!Q),\ SING(!Q,\ yesterday),$
$\ni(!Q,\ john),\ \ni(!Q,\ paul),\ \ni(!Q,\ george),\ \ni(!Q,\ ringo)$

S: "There are at least four persons singing 'Yesterday': they are John, Paul, George and Ringo."

7. A Mereology

In this section the switching of the basic collection-forming operator from a quasi-ordering relation $\subseteq$ founded on a membership relation $\ni$, to a primitive *part-whole* partial-ordering relation "$\succeq$" (Simons, 1987) is proposed. The idea is as follows.

Let's have now the roles defined according to the syntax rule:

$$R \rightarrow\ P \mid \succeq \mid \lhd C.R \mid \unlhd C.R \mid \lll C.R \mid \rhd C.R \mid \unrhd C.R \mid \blacktriangleright C.R$$

The basic collection-forming relation is the partial-ordering binary relation $\succeq$ (to be read HAS-PART) on the $\Delta^{\mathcal{I}}$ set – it is a reflexive, anti-symmetric and transitive relation.

Definition 4 (Part-whole relation)

$\forall x. \succeq(x, x).$

$\forall x, y. \succeq(x, y) \wedge \succeq(y, x) \rightarrow x = y.$

$\forall x, y, z. \succeq(x, y) \wedge \succeq(y, z) \rightarrow \succeq(x, z).$

Again, the language embodies plural quantifiers which specify the reading of a binary relation applied to collections. Plural quantifiers are *qualified* in the sense that the elements of actual predications are selected by a qualification predicate C. The $\lhd$ and $\rhd$ quantifiers specify that the relation necessarily holds for all the parts of a certain type C, i.e. they express the left and right distributive readings. The $\unlhd$ and $\unrhd$ quantifiers specify that the relation necessarily holds for some parts including all the parts of a certain type C, i.e.

they express the left and right cumulative readings. Moreover, a new class of plural quantifiers can be easily introduced in the mereological framework: the $\lhd\!\!\!\lhd$ and $\rhd\!\!\!\rhd$ quantifiers. They allow us to represent the *group* reading of a relation – as in (Landman, 1989a; Landman, 1989b). The $\lhd\!\!\!\lhd$ and $\rhd\!\!\!\rhd$ quantifiers specify that the relation possibly holds for *some* part of a certain type C.

The semantics of the plural quantifiers is given by the following definition:

Definition 5 (Qualified Plural Quantifiers)

$\lhd C.R(a,b)$ iff $\forall x.\,(\succeq(a,x) \wedge C(x)) \rightarrow R(x,b)$

$\rhd C.R(a,b)$ iff $\forall x.\,(\succeq(b,x) \wedge C(x)) \rightarrow R(a,x)$

$\unlhd C.R(a,b)$ iff $\forall x.\,(\succeq(a,x) \wedge C(x)) \rightarrow (\exists s.\,\succeq(s,x) \wedge R(s,b))$

$\unrhd C.R(a,b)$ iff $\forall x.\,(\succeq(b,x) \wedge C(x)) \rightarrow (\exists s.\,\succeq(s,x) \wedge R(a,s))$

$\lhd\!\!\!\lhd C.R(a,b)$ iff $\exists x.\,\succeq(a,x) \wedge C(x) \wedge R(x,b)$

$\rhd\!\!\!\rhd C.R(a,b)$ iff $\exists x.\,\succeq(b,x) \wedge C(x) \wedge R(a,x)$

As it is expected, the cumulative plural quantified expressions are still more general than the unquantified (collective) expressions and the distributive plural quantified ones:

$$\forall x,y.\ R(x,y) \rightarrow \unlhd C.R(x,y), \qquad \forall x,y.\ \lhd C.R(x,y) \rightarrow \unlhd C.R(x,y),$$
$$\forall x,y.\ R(x,y) \rightarrow \unrhd C.R(x,y), \qquad \forall x,y.\ \rhd C.R(x,y) \rightarrow \unrhd C.R(x,y).$$

In this way, ambiguities in the readings can be preserved by using the cumulative operator.

In this mereological framework, a *plural* operator '$\star$' is defined as follows:

Definition 6 (Plural Operator)

$\star P(a)$ iff $\forall x.\succeq(a,x) \rightarrow \big(\exists y.P(y) \wedge \big(\succeq(x,y) \vee \succeq(y,x)\big)\big)$

The purpose of the plural operator is to allow the construction of plural collective entities having singular objects of a certain type as their parts. More specifically, for any subpart x of the plural entity, either x is a part of the given singular type or the latter is part of x. We can define, for example, the plural predicate PEOPLE from the singular predicate PERSON:

$$PEOPLE \doteq \star PERSON.$$

In this way, a *non-extensional Mereology* is reconstructed, which is a generalization of the simple Theory of Collections presented above. It is called *non-extensional* for the same reasons as in collection theory: the extensionality axiom is not valid within this mereology.

Elements of a collection are *parts* in the partial order: $\succeq(beatles,\ john)$.

As observed in (Sowa, 1991), the main advantage of this approach is its uniformity in treating elements and collections. The difference is evident if we look at the semantics of the cumulative plural quantifier: the Collection Theory distinguishes an individual x from a singleton collection s whose only element is x, such that $\ni(s, x)$: this is why an explicit disjunction was introduced in the semantics. On the other hand, within mereology the disjunction disappears, since the $\succeq$ relation is reflexive: $\succeq(john,\ john)$.

In order to recover the lost distinction between a collection and its elements, *qualification* on plural quantifiers is needed. The qualification predicate acts like a filter to select the correct level in the mereological partonomy. In this way, problems with *multi-level* plural entities (Link, 1993) can be solved. The examples for distributive and cumulative readings, using *qualified* plural quantifiers, are expressed in the mereological framework in the following way:

"The Beatles are born in Liverpool"

$\big(\lhd PERSON.BORN\text{-}IN\big)(beatles,\ liverpool)$,

"The Beatles sing 'Yesterday' "

$\big(\unlhd PERSON.SING\big)(beatles,\ yesterday)$.

The qualification predicates for the plural quantifiers are taken from the lexical definition of the involved relations: in this example, the qualification $PERSON$ comes from the lexical semantics entries of the relations $BORN\text{-}IN$ and $SING$.

The *group* reading is a sort of *weakened* collective reading; for example:

"The Beatles played in London"

$\big(\lll PEOPLE.PLAY\text{-}IN\big)(beatles, london)$,

represents the case in which it is unknown whether the group which played in London was composed actually by *all* the members of the Beatles. So, the relation $PLAY\text{-}IN$ holds for some collection of persons, i.e. an element of the class $PEOPLE$, which should be a part of the collective entity $beatles$.

A mereological version of the collection theory has also been applied to model the structure of *events* and *processes* (Bach, 1986) in the domain of tense and aspect in natural language, in order to properly account for perfective and imperfective sentences and for habituals by means of plural quantifiers ranging on collections of events (Franconi *et al.*, 1993a; Franconi *et al.*, 1993b). The basic assumption taken into consideration is that verbal morphology plays a crucial role in specifying the temporal meaning of a sentence.

Several issues still need to be considered within the mereological approach: the existence of atoms (entities which have no parts), the possible inclusion of several different $\succeq_i$ relations (Winston *et al.*, 1987), the introduction of a mass dissective predicate as in (Link, 1983), and, last but not least, the calculus.

This reformulation of the collection theory can be adopted for the logical analysis of plurals according to the lattice-theoretical approach pursued by (Link, 1983).

8. Conclusions

In this paper a Collection Theory has been presented, i.e. a formalism which is intended to give semantical and computational means to plurals and plural quantifications. This approach allows for complete reasoning even in the presence of scoping ambiguities.

The concept language $\mathcal{ALCS}$ has been studied, which embeds in a uniform and compositional way the collection theory, and a sound and complete algorithm to decide satisfiability, subsumption and instantiation for a slightly weaker variant of the language has been devised. Within this concept language, it is possible to capture many phenomena covered by the Generalized Quantifier Theory.

These extensions are planned to be added in the natural language dialogue system AlFresco (Stock, 1991), a system in which the logical analysis of the sentences is carried on using a concept language (Cattoni and Franconi, 1990; Franconi, 1991; Lavelli *et al.*, 1992).

Finally, a mereological framework has been suggested, and it has been argued that a theory of part-whole relation is more expressive and more adequate than an element-based collection theory. However, more work is needed to refine this mereological framework.

Acknowledgements

This paper is a substantially revised and extended version of a paper presented at the ECAI Workshop on the Theoretical Foundations of Knowledge Representation and Reasoning, Vienna, August 1992, and at the AAAI Fall Symposium “Issues in Description Logics: Users Meet Developers”, Cambridge MA, October 1992 . This work has been partially supported by the Italian National Research Council (CNR), project “Sistemi Informatici e Calcolo Parallelo”, and by the IRST MAIA project within the Natural Language Group headed by Oliviero Stock. The paper owes *a lot* to Jürgen Allgayer (IBM Germany), together with this work was started. I would like to thank also Alessandro Artale, Werner Nutt and Achille Varzi for the helpful and incisive discussions I had with them.

Appendix

A. The calculus for $\mathcal{ALCS}$

In this section an algorithm for deciding satisfiability of $\mathcal{ALCS}$-concepts is proposed. This algorithm can be used also to decide subsumption between two concepts and instance recognition between an individual and a concept. The soundness of the algorithm and its completeness with respect to $\mathcal{ALCS}^{\perp}$, which is a weaker variant of $\mathcal{ALCS}$, will be proved. The algorithm is sound but not complete for $\mathcal{ALCS}$.

In order to obtain $\mathcal{ALCS}^{\perp}$ from $\mathcal{ALCS}$, the Collection Theory is changed by relaxing the semantics of the $\subseteq$, $\cap$, ($\lhd$ R), ($\rhd$ R), ($\unlhd$ R) and ($\unrhd$ R) *collective roles*.

Definition 7 (Collective Roles for $\mathcal{ALCS}^{\perp}$)

$$\subseteq(a,b) \Rightarrow \forall x.\ni(a,x) \to \ni(b,x)$$

$$\cap(a,b) \Rightarrow \exists x.\ni(a,x) \wedge \ni(b,x)$$

$$\lhd R(a,b) \Rightarrow \forall x.\ni(a,x) \to R(x,b)$$

$$\rhd R(a,b) \Rightarrow \forall x.\ni(b,x) \to R(a,x)$$

$$\unlhd R(a,b) \Rightarrow \forall x.\ni(a,x) \to (\,(\exists s.\ni(s,x) \wedge R(s,b)) \vee R(x,b))$$

$$\unrhd R(a,b) \Rightarrow \forall x.\ni(b,x) \to (\,(\exists s.\ni(s,x) \wedge R(a,s)) \vee R(a,x)).$$

With respect to the definition of the collective roles for $\mathcal{ALCS}$, the *iff* '$\Leftrightarrow$' has been replaced with a simple implication '$\Rightarrow$'. In this way, the collective roles become *primitive*; nonetheless, they still induce a structure between the elements. Even if at first sight this simplification may give the impression that the obtained theory is too weak, we claim that this is not actually the case. There are two reasons: first, in a open world semantics – which is usually adopted for NLP semantics – the lost deductions that can be performed in $\mathcal{ALCS}$ from the elements to the collective roles are very few; second, from the natural language point of view such deductions are not the intuitive ones. As a matter of fact, in the dialogue given in section 6 there is only one missed inference, which has a minor impact in the flux of the conversation. In the following we will refer to the language $\mathcal{ALCS}^{\perp}$.

The rule-based calculus to decide the satisfiability of $\mathcal{ALCS}^{\perp}$ and $\mathcal{ALCS}$ concepts operates on constraints (Hollunder *et al.*, 1990). A constraint can be of the type $x : C$ or xRy, where C is a concept, R is a role, and x, y are variables belonging to a predefined alphabet of variable symbols.
The interpretation of constraints is defined as follows. Let $\mathcal{I}$ be an interpretation of the concept language. An *$\mathcal{I}$-assignment* α is a function that maps every variable to an element of $\Delta^{\mathcal{I}}$. We say that α *satisfies* $x : C$ if and only if $\alpha(x) \in C^{\mathcal{I}}$, and α *satisfies* xRy if and only if $(\alpha(x), \alpha(y)) \in R^{\mathcal{I}}$.
A *constraint system* S is a finite, non-empty set of constraints. An $\mathcal{I}$-assignment α *satisfies* a constraint system S if α satisfies every constraint in S. A constraint system S is *satisfiable* if there is an interpretation $\mathcal{I}$ and an $\mathcal{I}$-assignment α such that α satisfies S.
A *clash* is a system having one of the forms: $\{x : \perp\}, \{x : A, x : \neg A\}$.

Proposition 1. **(Reduction to a constraint system)** A $\mathcal{ALCS}^{\perp}$ concept C is satisfiable if and only if the constraint system {x:C} is satisfiable.

Proof. Follows from the definitions. □

For the calculus, we consider only *simple* $\mathcal{ALCS}^{\perp}$ concepts. A concept is called simple if it contains only complements of the form $\neg A$, where A is a primitive concept. An arbitrary $\mathcal{ALCS}^{\perp}$ ($\mathcal{ALCS}$) concept can be transformed in linear time into an equivalent simple concept by means of the following rewriting rules:

$$\neg\top \rightarrow \bot \qquad \neg(C \sqcap D) \rightarrow \neg C \sqcup \neg D \qquad \neg\forall R.C \rightarrow \exists R.\neg C$$
$$\neg\bot \rightarrow \top \qquad \neg(C \sqcup D) \rightarrow \neg C \sqcap \neg D \qquad \neg\exists R.C \rightarrow \forall R.\neg C$$
$$\neg\neg C \rightarrow C$$

Starting from the system $S = \{x : C\}$, the *propagation rules* are applied, until a contradiction is generated or a model of C is explicitly obtained: the propagation rules preserve satisfiability. It is important to remark that they are introduced in order to prove satisfiability, and that they are *not* intended to be used as deduction rules. We have the following rules:

$S \rightarrow_{\sqcap} \{x : C_1,\ x : C_2\} \cup S$
if $x : C_1 \sqcap C_2$ in S, and both $x : C_1$ and $x : C_2$ not in S

$S \rightarrow_{\sqcup} \{x : D\} \cup S$
if $x : C_1 \sqcup C_2$ in S, and neither $x : C_1$ nor $x : C_2$ in S, and $D = C_1$ or $D = C_2$

$S \rightarrow_{\exists} \{xRy,\ y : C\} \cup S$
if $x : \exists R.C$ in S, and there is no z s.t. both xRz and $z : C$ in S,
and y is a new variable

$S \rightarrow_{\forall} \{y : C\} \cup S$
if $x : \forall R.C$ and xRy in S, and $y : C$ not in S

$S \rightarrow_{\subseteq} \{y \ni z\} \cup S$
if $x \subseteq y$ and $x \ni z$ in S, and $y \ni z$ not in S

$S \rightarrow_{\cap} \{x \ni k,\ y \ni k\} \cup S$
if $x \cap y$ in S, and there is no z s.t. both $x \ni z$ and $y \ni z$ in S,
and k is a new variable

$S \rightarrow_{\lhd} \{zRy\} \cup S$
if $x(\lhd R)y$ and $x \ni z$ in S, and zRy not in S

$S \rightarrow_{\rhd} \{xRz\} \cup S$
if $x(\rhd R)y$ and $y \ni z$ in S, and xRz not in S

$S \rightarrow_{\unlhd} T \cup S$
if $x(\unlhd R)y$ and $x \ni z$ in S,
and there is no t s.t. both $t \ni z$ and tRy in S, and zRy not in S,
and s is a new variable, and $T = \{s \ni z,\ sRy\}$ or $T = \{zRy\}$

$S \rightarrow_{\unrhd} T \cup S$
if $x(\unrhd R)y$ and $y \ni z$ in S,
and there is no t s.t. both $t \ni z$ and xRt in S, and xRz not in S,
and s is a new variable, and $T = \{s \ni z,\ xRs\}$ or $T = \{xRz\}$

Propagation rules are either deterministic – they yield a uniquely determined constraint system – or nondeterministic ($\rightarrow_{\sqcup}, \rightarrow_{\unlhd}, \rightarrow_{\unrhd}$) – yielding several possible constraint systems.

Proposition 2. **(Local soundness and completeness)** Let S be a constraint system of $\mathcal{ALCS}^{\perp}$ concepts. If S' is obtained from S by applying a deterministic propagation rule, then S is satisfiable if and only if S' is satisfiable. If S' is obtained from S by applying a nondeterministic propagation rule, then S is satisfiable if S' is satisfiable; moreover, there is a way to apply the rule to S such that the obtained system is satisfiable if and only if S is satisfiable.

Proof. Easy by translating the constraint systems into logical formulas. □

A constraint system is said to be *complete* if no propagation rule is applicable to it. Because of the presence of nondeterministic rules, several complete systems can be derived from $\{x : C\}$.

Proposition 3. **(Termination)** Let C be a simple $\mathcal{ALCS}^{\perp}$ concept. Given the constraint system $\{x : C\}$, after a finite number of applications of the propagation rules one obtains a finite set of complete constraint systems.

Proof. (sketch) One should prove that the size of each obtained complete constraint system is finite – in particular by checking the number of the newly introduced variables –, and that the number of complete constraint systems generated by the propagation rules is finite – by observing that the application of the rules decreases the complexity of the constraint system, i.e. rules add to the system only *simpler* constraints for which no rule directly applies again. □

Proposition 4. **(Satisfiability)** A constraint system $S = \{x : C\}$ is satisfiable if and only if there exist a clash free complete constraint system which can be derived from S by applying the propagation rules.

Proof. (sketch) First prove that a complete system is satisfiable if and only if it contains no clash. Then the proposition follows from local soundness and completeness and the termination of the propagation rules, and from the independence of the meanings of the basic role expressions. □

Now, it is straightforward to put together the blocks and build up a sound and complete decision procedure to check satisfiability of $\mathcal{ALCS}^{\perp}$ concepts. One should collect all the complete constraint systems derivable from $\{x : C\}$ by applying the propagation rules. Such systems are, up to variable renaming, finitely many. If at least one of these systems is clash free, then C is satisfiable, otherwise it is unsatisfiable. So, the following holds:

Theorem 1. **(Decidability)** Satisfiability, subsumption and instance recognition problems in $\mathcal{ALCS}^{\perp}$ are decidable.

Corollary 1. **(Algorithm for $\mathcal{ALCS}^{\perp}$)** The proposed algorithm is a sound and complete decision procedure to check satisfiability, subsumption and instance recognition in $\mathcal{ALCS}^{\perp}$.

Corollary 2. **(Algorithm for $\mathcal{ALCS}$)** The proposed algorithm is a sound decision procedure to check satisfiability, subsumption and instance recognition in $\mathcal{ALCS}$.

From the computational point of view, the precise complexity of the satisfiability problem still need to be found. It is easy to see that the problem is PSPACE-hard – such a lower bound comes from $\mathcal{ALC}$. However, it is not known if a PSPACE-algorithm for checking the satisfiability of $\mathcal{ALCS}^{\perp}$ concepts exists. Moreover, we have not yet looked for a functional algorithm (Baader and Hollunder, 1991), which is more appropriate for implementing purposes. Finally, a less incomplete algorithm for $\mathcal{ALCS}$ is under study.

References

Allgayer, J. and Franconi, E. 1992a. Collective entities and relations in concept languages. In *Workshop Notes of the ECAI Workshop on Theoretical Foundations of Knowledge Representation and Reasoning*, Vienna, Austria. 1–5. Appears also in the Working Notes of the AAAI Fall Symposium "Issues in Description Logics: Users Meet Developer", Cambridge MA, October 1992.

Allgayer, J. and Franconi, E. 1992b. A semantic account of plural entities within a hybrid representation system. In *Proc. of the 5 th International Symposium on Knowledge Engineering*, Seville, Spain. 305–312.

Allgayer, J.; Jansen-Winkeln, R.; Reddig, C.; and Reithinger, N. 1989. Bidirectional use of knowledge in the multi-modal NL access system XTRA. In *Proc. of the 11 th IJCAI*, Detroit, MI.

Allgayer, J. 1990. L_{SB-ONE^+} – dealing with sets efficiently. In *Proc. of the 9 th ECAI*, Stockholm, Sweden. 13–18.

Baader, F. and Hollunder, B. 1991. A terminological knowledge representation system with complete inference algorithm. In *Proc. of the Workshop on Processing Declarative Knowledge*, Kaiserslautern, Germany. 67–86.

Bach, E. 1986. The algebra of events. *Linguistics and Philosophy* 9:5–16.

Barwise, J. and Cooper, R. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4:159–219.

Bateman, John A.; Kasper, Robert T.; Moore, Johanna D.; and Whitney, Richard A. 1990. A general organization of knowledge for natural language processing: the PENMAN upper model. Technical report, USC/Information Science Intitute, Marina del Rey CA.

Cattoni, R. and Franconi, E. 1990. Walking through the semantics of frame-based description languages: A case study. In *Proc. of the 5 th International Symposium on Methodologies for Intelligent Systems*, Knoxville, TN. 234–241.

Donini, F. M.; Lenzerini, M.; Nardi, D.; and Nutt, W. 1991a. The complexity of concept languages. In *Proc. of the 2 nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA. 151–162.

Donini, F. M.; Lenzerini, M.; Nardi, D.; and Nutt, W. 1991b. Tractable concept languages. In *Proc. of the 12 th IJCAI*, Sidney, Australia. 458–465.

Donini, F. M.; Hollunder, B.; Lenzerini, M.; Spaccamela, A. Marchetti; Nardi, D.; and Nutt, W. 1992. The complexity of existential quantification in concept languages. *Artificial Intelligence* 53:309–327.

Franconi, Enrico; Giorgi, Alessandra; and Pianesi, Fabio 1993a. A mereological approach to tense and aspect. In *Proc. of the International Conference on Mathematical Linguistics, ICML-93*, Barcelona, Spain. (Abstract).

Franconi, Enrico; Giorgi, Alessandra; and Pianesi, Fabio 1993b. Tense and aspect: a mereological approach. In *Proc. of the 13 th IJCAI*, Chambery, France 1222–1228.

Franconi, E. 1991. A short presentation of YAK, a hybrid knowledge representation system. In Peltason, C.; Luck, K.von; and Kindermann, C., editors 1991, *Proc. of the Terminological Logic Users Workshop*, Berlin, Germany. Department of Computer Science, Technische Universität Berlin. 174–177.

Franconi, E. 1992. Adding constraints inference to ABox reasoning. Abstract 9205-02, IRST, Povo TN, Italy.

Hollunder, B. and Baader, F. 1991. Qualifying number restriction in concept languages. In *Proc. of the 2 nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA. 335–346.

Hollunder, B.; Nutt, W.; and Schmidt-Schauß, M. 1990. Subsumption algorithms for concept description languages. In *Proc. of the 9 th ECAI*, Stockholm, Sweden. 348–353.

Hollunder, B. 1990. Hybrid inferences in KL-ONE-based knowledge representation systems. In *Proc. of the 14 th German Workshop on Artificial Intelligence*. Springer-Verlag.

Kempson, R. and Cormack, A. 1981. Ambiguity and quantification. *Linguistics and Philosophy* 4(2):259–310.

Landman, Fred 1989a. Groups, I. *Linguistics and Philosophy* 12:559–605.

Landman, Fred 1989b. Groups, II. *Linguistics and Philosophy* 12:723–744.

Lavelli, Alberto; Magnini, Bernardo; and Strapparava, Carlo 1992. An approach to multilevel semantics for applied systems. In *Proc. of the 3 rd ACL Conference on Applied Natural Language Processing*, Trento, Italy. 17–24.

Lesmo, L.; Berti, M.; and Terenziani, P. 1988. A network formalism for representing natural language quantifiers. In *Proc. of the 8 th ECAI*, Munich, Germany. 473–478.

Link, Godehard 1983. The logical analysis of plurals and mass terms: a lattice-theoretical approach. In Bauerle, R.; Schwarze, C.; and Stechow, A.von, editors 1983, *Meaning, Use and Interpretation of Language*. Walter de Gruyter. 302–323.

Link, Godehard 1993. Algebraic semantics for natural language: some philosophy, some applications. In Guarino, Nicola and Poli, Roberto, editors 1993, *Proc. of the International Workshop on Formal Ontology*, Padova, Italy. 19–49.

Poesio, M. 1990. Dialog-oriented ABoxing. In *Proc. of the 5 th International Symposium on Methodologies for Intelligent Systems*, Knoxville, TN. 277–288.

Poesio, Massimo 1991. Relational semantics and scope ambiguity. In Barwise, J.; Gawron, J. M.; Plotkin, G.; and Tutiya, S., editors 1991, *Situation Semantics and its Applications, vol.2*. CSLI, Stanford, CA. chapter 20, 469–497.

Quantz, J. 1992. How to fit generalized quantifiers into terminological logics. In *Proc. of the 10 th ECAI*, Vienna, Austria. 543–547.

Samek, Vieri Lodovici and Strapparava, Carlo 1990. Identifying noun phrase references: the topic module of the AlFresco system. In *Proc. of the 9 th ECAI*, Stockholm, Sweden. 573–578.

Schmidt-Schauß, M. and Smolka, G. 1991. Attributive concept descriptions with complements. *Artificial Intelligence* 48(1):1–26.

Shapiro, S. C. and Rapaport, W. J. 1992. The SNePS family. *Computer and Mathematics with Applications, special issue: Semantic Networks in Artificial Intelligence* 23(2-5):243–275.

Simons, P. 1987. *Parts: A Study in Ontology*. Clarendon Press, Oxford.

Sowa, J. F. 1991. Toward the expressive power of natural language. In Sowa, J. F., editor 1991, *Principles of Semantic Networks*. Morgan Kaufmann. 157–189.

Stock, Oliviero 1991. Natural language and exploration of an information space: the AlFresco interactive system. In *Proc. of the 12 th IJCAI*, Sidney, Australia. 972–978.

Tjan, Bosco S.; Gardiner, David A.; and Slagle, James R. 1992. Representing and reasoning with set referents and numerical quantifiers. In Nagle, T. E.; Nagle, J. A.; Gerholz, L. L.; and Eklund, P. W., editors 1992, *Conceptual Structures, current reasearch and practice*. Ellis Horwood. chapter 2, 53–66.

van Benthem, J. 1986. *Essays in Logical Semantics*. Studies in Linguistics and Philosophy. D. Reidel Publishing Company, Dordrecht, Holland.

Weischedel, R. M. 1989. A hybrid approach to representation in the Janus natural language processor. In *Proc. of ACL-89*, Vancouver, British Columbia.

Wellman, Michael P. and Simmons, Reid G. 1988. Mechanisms for reasoning about sets. In *Proc. of AAAI-88*, St. Paul, MN. 398–402.

Winston, Morton E.; Chaffin, Roger; and Herrmann, Douglas 1987. A taxonomy of part-whole relations. *Cognitive Science* 11:417–444.

Woods, William A. and Schmolze, James G. 1992. The KL-ONE family. *Computer and Mathematics with Applications, special issue: Semantic Networks in Artificial Intelligence* 23(2-5):133–177.

Zancanaro, Massimo; Stock, Oliviero; and Strapparava, Carlo 1993. Dialogue cohesion sharing and adjusting in an enhanced multimodal environment. In *Proc. of the 13 th IJCAI*, Chambery, France.