



**HAL**  
open science

## Accessibility for Plurals in Continuation Semantics

Sai Qian, Maxime Amblard

► **To cite this version:**

Sai Qian, Maxime Amblard. Accessibility for Plurals in Continuation Semantics. Motomura Yoichi; Butler Alastair; Bekki Daisuke. *New Frontiers in Artificial Intelligence - JSAI-isAI 2012 Workshops, LENLS, JURISIN, MiMI, Miyazaki, Japan, November 30 and December 1, 2012, Revised Selected Papers.*, 7856, Springer International Publishing, pp.53-68, 2013, Lecture Notes in Computer Science, 978-3-642-39931-2. 10.1007/978-3-642-39931-2\_5 . hal-00762203

**HAL Id: hal-00762203**

**<https://hal.science/hal-00762203>**

Submitted on 6 Dec 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Accessibility for Plurals in Continuation Semantics

Sai Qian<sup>1,2</sup>, Maxime Amblard<sup>1,2</sup>

<sup>1</sup> LORIA - Campus Scientifique - BP 239 - 54506 Vandoeuvre-lès-Nancy Cedex

<sup>2</sup> Université de Lorraine, 34 cours Léopold, CS 25233, 54052 Nancy Cedex - France  
`sai.qian@loria.fr` ; `maxime.amblard@loria.fr`

**Abstract.** This paper accounts for the semantics of plurality, in particular, the accessibility relation for plural anaphoras. Two types of discourse antecedents formations, inherited from the classical treatment [14], namely summation and abstraction, are studied in detail. Solutions for each phenomenon are provided respectively by introducing two new functions  $\mathfrak{S}\text{um}$  and  $\mathfrak{A}\text{bs}$ , for obtaining the semantic interpretations. The technical background in this paper is based on a recently proposed dynamic semantic framework [9], which pertains a style of the traditional Montague Grammar and the principle of compositionality, without adding any new concept to classical logic.

## 1 Plural Anaphors in Discourse

### 1.1 Overview

In linguistics, anaphora stands for the phenomenon where the interpretation of one expression (anaphor) depends on a preceding expression (antecedent). As a crucial type of reference, which ties pieces of discourse into a “unified whole”, anaphora plays an important role almost in every natural language. In computational linguistics, the task to determine which antecedent an anaphor refers to is called anaphora resolution, which is still challenging.

In formal semantics, despite its pervasive influence, Montague Grammar (MG) failed to explain a bunch of linguistic phenomena, including presupposition, donkey sentences, anaphora (intra-sentential and inter-sentential), etc. Under such background, natural language meaning transferred from truth conditions into “context change potential”, which was carried out by dynamic semantics, as opposed to the traditional static MG. As one of the representative works in dynamic semantics [11,13,8], Discourse Representation Theory (DRT) proposes the concept of discourse referent, which contains a list of potential antecedents for various accessible anaphors to choose from.

At the same time, the problem of plurality (particularity on noun phrases) has been constantly drawing linguists’ attention [6,17,15,10]. For the sake of simplicity, most semantic theories, when first established, would take examples involving only singularity into consideration. Intuitively, plurality could simply be regarded as a multiplication of singularity on the aspect of quantity. As a

result, it is absolutely natural to view plurality as a smooth extension of singularity. However, the semantics of plurality brings about a number of intricate problems in natural languages. It can be inferred from numerous linguistic examples that they are absolutely unparallel phenomena. One of the most prominent issues in the semantics of plurality is on **distributivity** and **collectivity**.

- (1) a. John and Mary lifted a piano.
- b. John and Mary went to school.
- c. John and Mary gathered in Paris.

For instance in the above example, (1-a) often derives two distinct readings. One in which there were two pianos, *John* and *Mary* each lifted one of them, this is the so-called distributive reading. While in the other reading, *John* and *Mary* lifted one single piano together, this corresponds to the so-called collective reading. Different from (1-a), (1-b) and (1-c) do not seem to create a similar ambiguity. (1-b) only means the “going-to-school” event by *John* and *Mary* individually are independent from each other, while in (1-c), *John* and *Mary* have to be set in a group in order to serve as the agent for the “gathering” event. There are many literatures on the topic of distributivity and collectivity [4,7,21], however, this will not be the most fundamental concern in this paper.

As exemplified above, plurality is a more complicated linguistic issue than the naive singular extension, anaphora is among the numerous aspects. Normally, A singular pronoun (*I, you, he, she, it*) is used to refer to individual referent, while a plural pronoun (*we, you, they*) is understood as referring to a group of referents. Further more, an individual referent needs to be explicitly indicated to become a potential antecedent, but it is not the case for plural antecedent. Here, we stick with the classical treatment of plural anaphora in dynamic semantics [14], where plural referents are supposed be formed through two types of constructions: **summation** and **abstraction**.

## 1.2 Two Plurality Phenomena

**Summation** In the discourse, we often find several explicitly specified individual referents. On one hand, singular pronouns could be used to refer to any accessible singular antecedents; on the other hand, plural pronouns could also be applied in the same discourse, even if the group of individuals is not designated. The process of constructing such groups out of explicit individuals is defined as **summation**.

- (2) a. John went to Bill’s party with Mary. They had a nice time.
- b. John loves Mary. Bill also loves Mary. They have to find a solution.

In Examples (2-a) and (2-b), there are various ways to construct the group of antecedents. For instance, any combination of the previously mentioned proper names (*John, Mary, and Bill*) can serve as a valid candidate for “*they*” in (2-a). One interpretation could be that all three of them were happy, or only *John* and *Mary* were, or some other possibilities. In (2-b), it is more likely that “*they*”

refers to *John* and *Bill*, since they are rival in love according to the context. But it is also possible that all three of them are involved in finding a solution for such a situation. The following is another example of summation:

- (3) Students were in vacation. John went to Paris. Bill and Mary went to Rome. They were enjoying their time.

In (3), “*they*” could refer to the group of *John*, *Bill* and *Mary*, or the group of only *Bill* and *Mary*. As indicated in [14], summation allows plural reference to any subset of currently accessible singular discourse referents. While from the above examples, we can infer that the potential applicants include not only the explicitly mentioned individual referents, but also the implicitly constructed non-individual ones (groups).

**Abstraction** Another sort of antecedent formation, namely **abstraction**, can be elucidated from the following example:

- (4) a. \*Every farmer owns a donkey. He is rich.  
b. Every farmer owns a donkey. They are rich.

From (4), we can see that “*every farmer*” creates a group of farmers who individually owns a donkey. This group can only serve as a valid antecedent for plural pronoun “*they*”, while not for singular one “*he*”, as shown in (4-a) and (4-b). Normally, the NPs suitable for abstraction operation have the following structure in common:

*quantifier + plural noun*

In particular, abstraction first builds a property out of a dual condition, by abstracting over position of quantified NPs (QNPs), then a set satisfying the property is constructed. The process is trivial since the quantifier itself represents a relation between two sets. In English, those quantifiers include *every*, *all*, *none*, *most*, *few*, etc. Moreover, they hold a special name in formal semantics: **generalized quantifier**. Sometimes, even a group is obviously indicated in the discourse, it might be the case that the complement group is the one to be referred to, such as in the following example:

- (5) Few students came on time. They were too lazy.

where “*they*” are normally considered referring to those students who did not come on time. As a result, for generalized NPs, not only the referential group be taken into consideration, but also the complement one.

## 2 Plurality in Continuation Semantics

### 2.1 A Motagovian Treatment of Dynamic Semantics

Solutions concerning plurals under dynamic semantics have been proposed in various literatures [5,3,15]. Also, there are respectively direct plural extensions

for DRT [14] and Dynamic Predicate Logic (DPL) [19]. This paper aims to provide a corresponding continuation semantic solution, while the theoretical background of our work comes from another recently proposed dynamic framework [9], which combines the traditional MG with a new concept of context.

Unlike other dynamic formalisms, [9] gets rid of all ad hoc definitions, only by introducing a concept of “right context” as a traditional continuation [22]. Thus a new atomic type  $\gamma$  for the “right context” is interpolated in Church’s simple type theory, where  $\iota$  denotes the type of individual,  $o$  denotes the type of proposition<sup>3</sup>. Assuming  $s$  and  $t$  are respectively the syntactic category for sentence and discourse, their semantic interpretations in the new type system are:

$$\llbracket s \rrbracket = \llbracket t \rrbracket = \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

In order to conjoin the meanings of sentences for obtaining the composed meaning of a discourse (the updating process), the following formula is proposed:

$$\llbracket D.S \rrbracket = \lambda e \phi. \llbracket D \rrbracket e (\lambda e'. \llbracket S \rrbracket e' \phi) \quad (1)$$

where  $D$  is the preceding discourse,  $S$  is the sentence currently being processed. As for the other two variables,  $e$  stands for the left context,  $\phi$  the right context. The updated context  $D.S$ , which has the potential to update the context, possesses the same semantic type as  $D$  and  $S$ . Turning to DRT, if we assume “ $x_1, x_2, \dots$ ” are reference markers, and “ $C_1, C_2, \dots$ ” are conditions, the corresponding  $\lambda$ -term for a general DRS in the new framework should be:

$$\lambda e \phi. \exists x_1 \dots x_n. C_1 \wedge \dots \wedge C_m \wedge \phi e'^4$$

To solve the problem of anaphoric reference, [9] introduced a special choice operator (selection function). The choice operator is represented by some oracles, such as  $sel_{he}, sel_{she}, \dots$ . It takes the left context as argument and returns a resolved individual element. In order to update the context, another operator “ $::$ ” is introduced, which adds new accessible variables to the processed discourse. For instance, term “ $a :: e$ ” actually is interpreted as “ $\{a\} \cup e$ ” mathematically. In other words, we can view the list as an updated version of the discourse referents in DRT, which keeps the accessible referents and discards those inaccessible ones for further processing. The compositional treatment of Example (6) could serve as a nice illustration for the new framework.

- (6)    a.    John kisses Mary.  
        b.    She smiles.

The following are the semantic entries of the lexicon:

<sup>3</sup> We stick with the original denotation in [9], but there is no great difference between  $\iota, o$  (Church’s denotation) and  $e, t$  (Montague’s denotation).

<sup>4</sup> Here, “ $e'$ ” is a left context made of “ $e$ ” and the variables “ $x_1, x_2, x_3 \dots$ ”. Its construction depends on the specific structure of the context, for more details see [9].

- $\llbracket \text{John/Mary} \rrbracket = \llbracket \text{John/Mary} \rrbracket = \lambda\psi e\phi.\psi j/m(j/m :: e)\phi^5$
- $\llbracket \text{she/they} \rrbracket = \lambda\psi e\phi.\psi(\text{sel}_{\text{she/they}}e)e\phi$
- $\llbracket \text{smiles} \rrbracket = \lambda s.s(\lambda x e\phi.\text{Smile}(x) \wedge \phi e)$
- $\llbracket \text{kisses} \rrbracket = \lambda o s.s(\lambda x o(\lambda y e\phi.\text{Kiss}(x, y) \wedge \phi e))$

Here are some remarks for the interpretations above: in “ $\llbracket \text{John/Mary} \rrbracket$ ”, the discourse referent “ $j/m$ ” is inserted in the left context explicitly with the list structure “ $(j/m :: e)$ ”; while for pronoun “ $\llbracket \text{she/they} \rrbracket$ ”, the selection function, denoted by the oracle “ $\text{sel}_{\text{she/they}}$ ”, is applied to the context which contains all accessible discourse referents; there is nothing special for predicate “ $\llbracket \text{smiles} \rrbracket$ ” and “ $\llbracket \text{kisses} \rrbracket$ ”, they are just the usual terms as in MG, but renewed with a proper treatment of context (e.g., the “ $\phi e$ ”).

With the entries above, we can obtain the semantic representation for (6-a) and (6-b) respectively, then finally the one for (6) by applying Formula 1 to the two components.

- $\llbracket (6\text{-a}) \rrbracket = (\llbracket \text{kisses} \rrbracket \llbracket \text{Mary} \rrbracket) \llbracket \text{John} \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi.\text{Kiss}(j, m) \wedge \phi(m :: j :: e)$
- $\llbracket (6\text{-b}) \rrbracket = \llbracket \text{smile} \rrbracket \llbracket \text{she} \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi.\text{Smile}(\text{sel}_{\text{she}}(e)) \wedge \phi(e)$
- $\llbracket (6) \rrbracket = \llbracket D.S \rrbracket = \llbracket (6\text{-a}).(6\text{-b}) \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi.(\text{Kiss}(j, m) \wedge \text{Smile}(\text{sel}_{\text{she}}(j :: m :: e)) \wedge \phi(j :: m :: e))$

Thus, assume the selection function “ $\text{sel}_{\text{she}}$ ” works correctly, it should pick up the appropriate element “ $m$ ” from the candidate list. At the same time, the updated list “ $(j :: m :: e)$ ” containing both entities will be passed to the continuation of the current discourse.

To sum up, this new framework sticks itself in the traditional Montagovian style, it successfully solves several compositionality problems one might encounter when accommodating DRT and Montague Semantics, for example, variable binding and presupposition.

## 2.2 Plurality in The New Framework

**Summation** Back to Example (3), as explained, “*they*” can either refer to the group of *Bill* and *Mary* ( $m \oplus b$ )<sup>6</sup> or the group of all three entities ( $j \oplus m \oplus b$ ). However, our first concern is whether a group could be broken down into individuals even if it is explicitly mentioned, like “*Bill and Mary*” in Example (3). Let’s look at the following examples:

- (7)    a. John went to Paris. Bill and Mary went to Rome.  
           b. She enjoyed the historical monuments very much.  
           c. They planned the whole trip without telling her.

<sup>5</sup> Proper names in the original paper was represented as “ $\lambda\psi e\phi.\psi j/me(\lambda e.\phi(j/m :: e))$ ”, while it has been recently updated by the author to the above form.

<sup>6</sup> The “ $\oplus$ ” operator conjoins two entities into a referential group, it is different from the set consisting of two entities, which denotes a property in mathematical logic.

Obviously, either (7-b) or (7-c) could serve as a valid continuation for (7-a). On the one hand, in (7-b), “*she*” refers to *Mary*, which is the individual element from the explicitly specified group “*Bill and Mary*”; on the other hand, “*they*” in (7-c) denotes the group consisting of the two men (*John* and *Bill*) who planned the trip, in which *Bill* also comes from the explicitly group in (7-a). As a consequence, we have the intuition that not only can an explicit group be decomposed, but also can all sub-groups made up of the accessible individual elements be potential antecedents for a referring expression. To justify this perception, let us refer to the following example:

- (8) a. John was in Paris. Bill was in Rome. Mary was in Barcelona.  
 b. They would come back to work after the vacation.  
 c. They avoided the bad weather in France/Italy/Spain.

Example (8) is a typical case of summation, where three individual entities are introduced in parallel. Theoretically, not only *John*, *Bill* and *Mary* can serve as potential antecedents, but also certain combinations among them. One of the most prevailing continuations involving a plural pronoun is (8-b), where “*they*” is most probably to denote the group of all three subjects. However, another possible continuation is (8-c), in which the denotation of “*they*” is ambiguous to some extent. Depending on which country is indicated at the end of the sentence. “*They*” could refer to the group of *Bill* and *Mary* if the country is *France*, *John* and *Mary* if the country is *Italy*, or *John* and *Bill* if the country is *Spain*. Consequently, all possible sub-groups consisting of arbitrary combination of the entities could be employed as antecedents for a upcoming referential expression (both singular and plural).

Based on the above analysis, we propose the following interpretation for proper names:

$$\llbracket \text{John} \rrbracket = \lambda \psi e \phi . \psi j \mathfrak{S}um(j :: e) \phi \quad (2)$$

The embedded summation function  $\mathfrak{S}um$  constructs all possible sub-groups from current accessible referents (namely the **power-group**, in accordance to the definition of **power-set**), offering the plural anaphor appropriate group antecedents. The following examples illustrate how  $\mathfrak{S}um$  performs in practice, and a formal definition will be given shortly.

- $\mathfrak{S}um(j :: e) \Rightarrow (j :: e)$
- $\mathfrak{S}um(m :: j :: e) = \mathfrak{S}um(m :: \mathfrak{S}um(j :: e))$   
 $\Rightarrow_{\mathfrak{S}um} (m :: j :: j \oplus m :: e)$
- $\mathfrak{S}um(b :: m :: j :: e) = \mathfrak{S}um(b :: \mathfrak{S}um(m :: j :: e))$   
 $\Rightarrow_{\mathfrak{S}um} \mathfrak{S}um(b :: (m :: j :: j \oplus m :: e))$   
 $\Rightarrow_{\mathfrak{S}um} (b :: m :: j :: b \oplus m :: b \oplus j :: m \oplus j :: b \oplus m \oplus j :: e)$
- ...

In order to give a formal definition for the summation function  $\mathfrak{S}um$ , we make use of two other concepts in functional programming, mainly for operating on

the data structure *list*: the **append** and **add** operation, which will be defined below in a recursive style<sup>7</sup>:

**Definition 1 The Append Function  $\mathfrak{App}$**

$\mathfrak{App}$  takes two lists  $l_1$  and  $l_2$  as arguments,  $\mathfrak{App}(l_1, l_2)$  will be:

- $l_2$ , if  $l_1 = []$  - the empty list;
- $head_1 :: \mathfrak{App}(tail_1, l_2)$ , in which  $head_1$  and  $tail_1$  denote the head and the tail of  $l_1$  respectively.

**Definition 2 The Add Function  $\mathfrak{Add}$**

$\mathfrak{Add}$  takes two arguments, an element  $a$  and a list  $l$ ,  $\mathfrak{Add}(a, l)$  will be:

- $[a]$  - list containing a single element  $a$ , if  $l = []$ ;
- $a \oplus head :: \mathfrak{Add}(a, tail)$ , in which  $head$  and  $tail$  denote the head and tail of  $l$  respectively.

Accordingly, we can give the definition for summation function  $\mathfrak{Sum}$ :

**Definition 3 The Summation Function  $\mathfrak{Sum}$**

$\mathfrak{Sum}$  takes a list  $l$  as argument,  $\mathfrak{Sum}(l)$  will be:

- $[]$  - the empty list, if  $l = []$ ;
- $\mathfrak{App}(\mathfrak{Add}(head, sum\_tail), sum\_tail)$ , in which  $head$  denotes the head of  $l$ ,  $sum\_tail$  denotes the result of  $\mathfrak{Sum}(tail)$  where  $tail$  denotes the tail of  $l$ .

In fact, the Append Function  $\mathfrak{App}$  is exactly the same as its usual meaning in other theories involving list operation. The Add Function  $\mathfrak{Add}$ , on the other hand, “inserts” a new element to each element of a list. However, the object (namely the group) created after the insertion is not a list any more, since the group formation operator “ $\oplus$ ” is applied, instead of the conventional union operator “ $\cup$ ”. In this way the group created by concatenating several individual elements with “ $\oplus$ ” possesses the same semantic type as an individual element, since both of them are served as legal candidates for the anaphora resolving process, and should be passed to the same oracle *sel* function. Further more, the definition of the summation function  $\mathfrak{Sum}$  is also quite similar to the classical recursive algorithm for power-set, differing only by replacing the “ $\oplus$ ” operator with the union operation. While this tiny difference results in the great distinction between the types of the two output data structures.

Now let us have a look at another example for illustrating the compositional ability of our framework, after taking  $\mathfrak{Sum}$  into consideration:

- (9)    a.    John and Mary went to Paris.  
        b.    Bill and Lucy went to Rome.

---

<sup>7</sup> The term *list* and *set* are used interchangeably in this paper, which means, order of elements in a list is indifferent here.



To obtain the semantic representation of (9), we still need the lexicon for conjunction “*and*”. Since the evaluation for “*and*” often involves different levels of semantic entities (NPs or VPs), special treatments are usually required [12]. Further more, we follow the idea that the distributive and collective distinction comes from the predicates [4]. As a result, we propose the following two representations for coordinator “*and*”, one distributive and another collective. The distributive *and* can be used for those distributive predicates such as “*rum*”, “*smile*”, “*eat*”, etc.; while the collective one will be assigned when collective predicates are presented, such as “*gather*”, “*form*”, etc.

1.  $\llbracket and \rrbracket_{dis} = \lambda AB\psi e\phi. A\psi e(\lambda e'. B\psi e'\phi)$
2.  $\llbracket and \rrbracket_{coll} = \lambda AB\psi e\phi. A(\lambda x. B(\lambda y. \psi(x \oplus y)))e\phi$

The distributive *and* aims to pass the prospective predicate to each of the NPs independently; while the collective *and* abstracts two individual variables from the conjoined NPs, then creates a integrated group “ $x \oplus y$ ” as the potential referent. Applying the above entries with the lexicon of proper names 2, we can obtain a pair of plural NPs as following:

1.  $\llbracket John\_and\_Mary \rrbracket_{dis} = (\llbracket and \rrbracket_{dis} \llbracket John \rrbracket) \llbracket Mary \rrbracket$   
 $\Rightarrow (\lambda AB\psi e\phi. A\psi e(\lambda e'. B\psi e'\phi)(\lambda \psi e\phi. \psi j \text{Sum}(j :: e)\phi)) \lambda \psi e\phi. \psi m \text{Sum}(m :: e)\phi$   
 $\Rightarrow_{\beta} \lambda \psi e\phi. (\psi j \text{Sum}(j :: e)(\lambda e'. (\psi m \text{Sum}(m :: e')\phi)))$
2.  $\llbracket John\_and\_Mary \rrbracket_{coll} = (\llbracket and \rrbracket_{coll} \llbracket John \rrbracket) \llbracket Mary \rrbracket$   
 $\Rightarrow (\lambda AB\psi e\phi. A(\lambda x. B(\lambda y. \psi(x \oplus y)))e\phi)(\lambda \psi e\phi. \psi j \text{Sum}(j :: e)\phi) \lambda \psi e\phi. \psi m \text{Sum}(m :: e)\phi$   
 $\Rightarrow_{\beta} \lambda \psi e\phi. \psi(j \oplus m) \text{Sum}(m :: \text{Sum}(j :: e))\phi$

Therefore, which of the two interpretations to use will depend on the specific predicate. In Example (9), we treat “*go to Paris/Rome*” as distributive. Then, the interpretation of Example (9) could be obtained by applying the corresponding semantic entries:

- $\llbracket (9-a) \rrbracket = \llbracket Go\_to\_Paris \rrbracket \llbracket John\_and\_Mary \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi. (Go\_Paris(j) \wedge Go\_Paris(m) \wedge \phi(\text{Sum}(j :: \text{Sum}(m :: e))))$   
 $\Rightarrow_{\text{Sum}} \lambda e\phi. (Go\_Paris(j) \wedge Go\_Paris(m) \wedge \phi(j :: m :: j \oplus m :: e))$
- $\llbracket (9-a) \rrbracket = \llbracket Go\_to\_Rome \rrbracket \llbracket Bill\_and\_Lucy \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi. (Go\_Rome(b) \wedge Go\_Rome(l) \wedge \phi(\text{Sum}(b :: \text{Sum}(l :: e))))$   
 $\Rightarrow_{\text{Sum}} \lambda e\phi. (Go\_Rome(b) \wedge Go\_Rome(l) \wedge \phi(b :: l :: b \oplus l :: e))$
- $\llbracket (9) \rrbracket = \llbracket D.S \rrbracket = \llbracket (9-a).(9-b) \rrbracket$   
 $\Rightarrow_{\beta} \lambda e\phi. (Go\_Paris(j) \wedge Go\_Paris(m) \wedge Go\_Rome(b) \wedge Go\_Rome(l) \wedge \phi(\text{Sum}(j :: \text{Sum}(m :: \text{Sum}(b :: \text{Sum}(l :: e))))))$   
 $\Rightarrow_{\text{Sum}} \lambda e\phi. (Go\_Paris(j) \wedge Go\_Paris(m) \wedge Go\_Rome(b) \wedge Go\_Rome(l) \wedge \phi(j :: m :: b :: l :: j \oplus m :: j \oplus b :: j \oplus l :: m \oplus b :: m \oplus l :: b \oplus l :: j \oplus m \oplus b :: j \oplus m \oplus l :: m \oplus b \oplus l :: j \oplus m \oplus b \oplus l :: e))$

In the last formula above, when the left context is extended completely, all combinations of the four initial individual entities are included in the list structure. A pronoun in the continuation could choose any element in the list, which is the exact task for the selection function. It might seem over-generation at first glance because of the size of the formula, however, we showed the necessity to contain the power-group with Example (8) already. More on this point will be discussed in the conclusion.

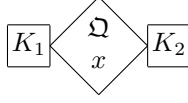
**Abstraction** In natural language discourse, antecedents are often accounted or constructed out of those explicitly mentioned referents, such as the cases for singular antecedents and the above mentioned group antecedents (summation). Nevertheless, there are also cases where plural anaphors refer back to some obscure referents in the context. The process of forming those concealed potential plural antecedents is called abstraction.

As described in the first Section, the universal quantifier “*every*”, which blocks singular pronouns, allows plural pronouns to refer back. We also stated that “*every*” is not the only quantifier which holds such property, there is actually a set of generalized quantifiers. Consider the following examples:

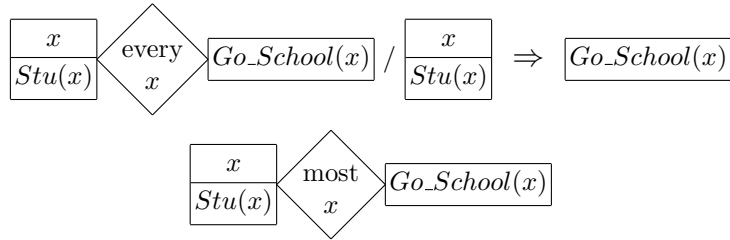
- (10)
- a. Every student went to school. They worked hard.
  - b. Most students went to school. They worked hard.
  - c. Two of five students went to school. They worked hard.
  - d. Two of five students went to school. They were at home.
  - e. Two of five students went to school. They had to hand in the homework by tomorrow.

All the subject-NPs above (e.g., *every student*, *most students*, *two of five students*) are composed of a generalized quantifier and a noun. This structure is used to denote a certain plural group antecedent. On the one hand, those QNPs have some special characters in creating referents. As remarked before, “*every student*” in (10-a) is singular in the morph-syntactic form, nevertheless, it only allows plural pronoun for a continuation. On the other hand, we notice that not all possible group antecedents are clearly marked, one such example can refer to (5). Likewise, in (10-c), (10-d) and (10-e), the plural anaphora “*they*” in the three discourses obviously denote distinct groups, despite all of them have the same first part. In (10-c), the students who did go to school is likely to be the referents; while in (10-d), “*they*” are more likely to denote to the group of students who did not show up; in the last one, the group of all students are preferable as the antecedent. On that account, the QNP “*two of five students*” must have introduced more than one potential plural antecedents.

According to [14], when abstraction applies to a sentence containing QNPs, it constructs a group of individuals, all of which satisfy the properties denoted by the QNPs. In DRT, this type of formation condition is called **duplex condition**, which is generally expressed with the following DRS:

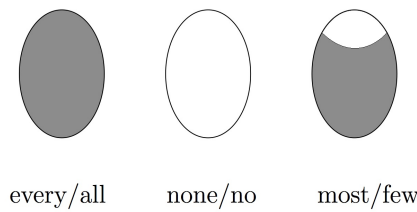


where  $\Omega$  is the interpretation of the specific generalized quantifier,  $K_1$  and  $K_2$  are the two referent-containing DRSs connected. Thus new group referents will be constructed through the function of  $\Omega$  on discourse referents in  $K_1$  and  $K_2$ . More details refers to Chapter 4 of [14]. Accordingly, the QNPs in (10-a) and (10-b), namely “*every student*” and “*most students*” could be interpreted respectively as:



The concept of generalized quantifier has been defined in literatures as “*a relation between two sets*” [18], e.g., the two sets of discourse referents in  $K_1$  and  $K_2$  in the general representation above. In fact, they also express the exact grouping conditions among a set of individuals. In English, the categorization of generalized quantifiers depends on a set of criteria, which will not be discussed in detail. Instead, we hold a exclusive view on quantifiers in this paper.

By and large, we consider all generalized quantifiers, including “*every*”, “*none*”, “*most*”, “*few*” and other complex ones such as “*two of five*”, always introduce three potential group referents when forming a QNP with a common noun. These three groups are named as **maximum group**, **reference group** and **complement group**<sup>8</sup>. The following diagram offers a comparison between the internal structure of three pairs of quantifiers.



**Fig. 1.** Structure Denoted by Generalized Quantifiers

<sup>8</sup> They are also sometimes referred as refset anaphora and compset anaphora in literature.

In the above figure, the filled part (gray) in the ellipse denotes the reference group, the wide part (white) denotes the complement group.

We can see that “*every/all*” introduces the reference group (equal to the maximum group) and the complement group. The former corresponds to the whole set, while the latter to the empty set; “*none/no*” works exactly in the opposite way, the reference group is the empty set, whereas the complement group (equal to the maximum group) is the whole set; “*most/few*” expresses the most general situation, which introduces a majority set, a minority set and a whole set, each corresponds exactly to the reference (or complement) group, complement (or reference) group and the maximum group. A similar analysis could be made for other complex proportional quantifiers (e.g., *two of five*). Hence, there are often more than one candidate antecedents simply because a generalized quantifier creates three (although it does not seem to be the case for “*every/all*” and “*none/no*”, their situations could be easily extended to the general one). Please note that all the three mentioned potential antecedents are group referents, which means none of them could serve as a proper antecedent for singular pronouns.

Therefore, we propose to unearth all possible groups that could serve as a plural referent for abstraction formation. That is to say, all three groups mentioned above, the maximum group, the reference group, the complement group, should be posted explicitly in the processing context. Based on the framework of [9], we assign the following interpretation for a generalized quantifier:

$$[[GQ]] = \lambda\psi ABe\phi.Quan(\psi)x.((Axe\lambda e.\top)Rel(\psi)(Bxe\lambda e.\top)) \wedge \phi((\mathfrak{Abs}(\psi, x) :: e)$$

Some remarks for the above formula: first of all, “ $\psi$ ” stands for the actual generalized quantifier being applied (e.g., *every*, *most*, *few*); secondly, “*Quan*” denotes a function that takes a verbatim quantifier as input, and yields a logic quantifier for the individual variables, “*Rel*” expects the same input as “*Quan*”, while it generates the corresponding logic connectives to link the two sets; in addition, the continuation passed to the two NPs in the formula is “ $\lambda e.\top$ ”, which is motivated by the treatment of universal quantifier from [9], it is originally used to limit the scope of quantifiers, but its function here is keep the logical part of the representation away from the continuation; lastly,  $\mathfrak{Abs}$  takes the quantifier and the variable as inputs, constructs all the group referents. Computationally, it is in fact not necessary to display all three groups in the left context, since anyone of them can be inferred from the rest two. As a result, we decide only to list explicitly the reference group and the complement group, and link them with co-indexing. In other words, the reference group and complement group which belong to the same set of individuals bears the same index. Following is the formal definition for the  $\mathfrak{Abs}$  function:

**Definition 4 The Abstraction Function  $\mathfrak{Abs}$**

$\mathfrak{Abs}$  takes two arguments: a generalized quantifier  $q$  and the related individual variable  $x$ . The output, namely  $\mathfrak{Abs}(q, x)$  will be a left context consisting of two group referents  $R_i$  and  $C_i$ :

- $R$ : the reference group of individuals denoted by the quantifier;
- $C$ : the complement group of individuals denoted by the quantifier;
- $i$ : the index that signifies the dependency of the two groups.

Note that besides “ $\mathfrak{Abs}$ ”, the other two functions mentioned above, “ $Quan$ ” and “ $Rel$ ”, are also quantifier-sensitive. That is to say, they return completely different output for different quantifiers as well. For instance, “ $Quan(every)$ ” creates “ $\forall$ ”, while “ $Quan(a)$ ” creates “ $\exists$ ” instead. This poses a dynamic view on generalized quantifiers.

More specifically, let us take the semantic interpretation of “ $every$ ” as a further illustration for the computability of  $\mathfrak{Abs}$ :

- $\llbracket every \rrbracket = \llbracket GQ \rrbracket(every)$   
 $\Rightarrow_{\beta} \lambda A B e \phi. Quan(every)x.(Axe\lambda e.\top Rel(every)Bxe\lambda e.\top) \wedge \phi(\mathfrak{Abs}(every, x) :: e)$   
 $\Rightarrow_{Quan, Rel} \lambda A B e \phi. \forall x.(Axe\lambda e.\top \rightarrow Bxe\lambda e.\top) \wedge \phi(\mathfrak{Abs}(every, x) :: e)$

Apply the above formula with the semantic entries for common noun “ $farmer$ ” and verb phrase “ $own\_a\_donkey$ ”, we can obtain the simplified representation of the classical “ $donkey$  sentence”:

- $\llbracket farmer \rrbracket = \lambda x e \phi.(Farmer(x) \wedge \phi e)$
- $\llbracket own\_a\_donkey \rrbracket = \lambda S.S(\lambda x e \phi. OD(x) \wedge \phi e)$
- $\llbracket every \rrbracket \llbracket farmer \rrbracket$   
 $\Rightarrow \lambda A B e \phi. \forall x.(Axe\lambda e.\top \rightarrow Bxe\lambda e.\top) \wedge \phi((R_1 :: C_1) :: e)(\lambda x' e' \phi'. (Farmer(x') \wedge \phi' e'))$   
 $\Rightarrow_{\beta} \lambda B e \phi. \forall x.(Farmer(x) \rightarrow Bxe\lambda e.\top) \wedge \phi(\mathfrak{Abs}(every, x) :: e)$
- $\llbracket own\_a\_donkey \rrbracket(\llbracket every \rrbracket \llbracket farmer \rrbracket)$   
 $\Rightarrow \lambda S.S(\lambda x' e' \phi'. od(x') \wedge \phi' e')(\lambda B e \phi. \forall x.(Farmer(x) \rightarrow Bxe\lambda e.\top) \wedge \phi(\mathfrak{Abs}(every, x) :: e))$   
 $\Rightarrow_{\beta} \lambda e \phi. \forall x.(Farmer(x) \rightarrow OD(x)) \wedge \phi(\mathfrak{Abs}(every, x) :: e)$   
 $\Rightarrow_{\mathfrak{Abs}} \lambda e \phi. \forall x.(Farmer(x) \rightarrow OD(x)) \wedge \phi(R_{far} :: C_{far} :: e)$

In this manner, the reference group “ $R_{far}$ ” and complement group “ $C_{far}$ ” of farmers, introduced by the universal quantifier “ $every$ ”, are inserted in the context for upcoming anaphors to choose from. Go back to Example (4), it might be clear why (4-a) is infelicitous, since there is no singular referent available. Assume the plural selection function “ $sel_{they}$ ” makes the good choice, the reference group should be picked for “ $they$ ” in (4-b). Actually, not only can “ $sel_{they}$ ” selects the appropriate candidate, it is also able generate the maximum group when necessary, such as for example (10-e). Moreover, it seems a little redundant for the complement group to exist in this example, because no continuation can access it, which might bring about the concern for over-generation again. However, we decide to keep the solution as general as possible, for cases involving other quantifiers such as “ $most$ ” and “ $few$ ”.

As a short remark, our proposal is not responsible for the complete task of anaphora resolution. That is, we only attempt to provide the possible list of

group referents for plural anaphors. Which referent to pick out finally requires other linguistic or computational mechanisms. Again in the above interpretation, this task is realized through the oracle selection function “ $sel_{he/they}$ ”.

### 3 Conclusion & Future Work

To sum up, first of all, we discussed the existing problems concerning plurality in the area of formal semantics in this paper. Further more, we briefly explained dynamic semantics, by presenting a recently proposed dynamic framework, which is based on the idea of continuation. Finally within the new framework, we introduced one of its extensions for handling plurality, particularly, the plural anaphora reference. The referent accessibility problem for plurals has been classified into summation and abstraction, depending on the different antecedent formation processes. Solution for each type has been provided respectively. In our proposal, plural group referent is the same semantic entity as singular referent. The group referent, which could serve as legal antecedent for plural anaphor, is distinct from the concept of set in a conventional mathematical or logic point of view.

One seeming concern of our framework is that both proposed functions,  $\mathfrak{S}\text{um}$  and  $\mathfrak{A}\text{bs}$ , might result in the problem of over-generation. However, we have shown with Example (8) and (10) it is possible that all sub-groups could serve as antecedents. We admit that the size of the context will grow exponentially once the number of accessible referents increases, which will make computation infeasible. A practical strategy is to strip off certain less possible referents from the left context at each processing stage, which could be further investigated and formalized in later works.

All in all, the accessibility problem addressed here is only a small part in the semantics of plurality. Future work could focus on the following aspects. In the first place, all examples in our paper are plain discourses without considering rhetorical relations (or discourse relations). However, it has been shown that the rhetorical relations play an important role in discourse referent accessibility [2,1], which is definitely worth being analyzed together with plurality in future. In addition, the interplay between plurality and eventuality might also be a potential topic [16,20], which would help to restrict the accessibility more precisely, especially for a deeper understanding of collectivity and distributivity. Finally, the mechanism for the new functions proposed in this paper:  $\mathfrak{S}\text{um}$  and  $\mathfrak{A}\text{bs}$ , needs more formal and detailed investigations, so as to adapt to other plurality phenomena such as independent plurals and release the framework from the concern of over-generation.

### References

1. Asher, N., Pogodalla, S.: Sdrt and continuation semantics. *New Frontiers in Artificial Intelligence* pp. 3–15 (2011)

2. Asher, N., Lascarides, A.: *Logics of conversation*. Cambridge University Press (2003)
3. van den Berg, M.: Dynamic generalized quantifiers. *Quantifiers, logic and language* pp. 63–94 (1996)
4. Dowty, D.: Collective predicates, distributive predicates and all. In: *Proceedings of the 3rd ESCOL*. pp. 97–115 (1987)
5. Elworthy, D.: A theory of anaphoric information. *Linguistics and Philosophy* 18(3), 297–332 (1995)
6. Gillon, B.S.: The readings of plural noun phrases in english. *Linguistics and Philosophy* 10(2), 199–219 (1987)
7. Gillon, B.: Collectivity and distributivity internal to english noun phrases. *Language Sciences* 18(1), 443–468 (1996)
8. Groenendijk, J., Stokhof, M.: Dynamic predicate logic. *Linguistics and Philosophy* 14(1), 39–100 (1991)
9. de Groote, P.: Towards a montagovian account of dynamics. *Proceedings of Semantics and Linguistic Theory XVI* (2006)
10. Hamm, F., Hinrichs, E.: *Plurality and quantification*, vol. 69. Kluwer Academic Publishers (1998)
11. Heim, I.: File change semantics and the familiarity theory of definiteness. In: Bäuerle, R., Schwarze, C., von Stechow, A. (eds.) *Meaning, Use, and Interpretation of Language*, pp. 164–189. Walter de Gruyter, Berlin (1983)
12. Hoeksema, J.: The semantics of non-boolean “and” i. *Journal of Semantics* 6(1), 19–40 (1988)
13. Kamp, H.: A theory of truth and semantic representation. In: Groenendijk, J., Janssen, T., Stokhof, M. (eds.) *Formal Methods in the Study of Language*, pp. 277–322. *Mathematical Centre Tracts 135*, Mathematisch Centrum, Amsterdam (1981)
14. Kamp, H., Reyle, U.: *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, vol. 42. Kluwer Academic Dordrecht, The Netherlands (1993)
15. Krifka, M.: Parametrized sum individuals for plural anaphora. *Linguistics and Philosophy* 19(6), 555–598 (1996)
16. Landman, F.: *Events and plurality: The Jerusalem lectures*, vol. 76. Kluwer Academic Pub (2000)
17. Lasersohn, P.: On the readings of plural noun phrases. *Linguistic Inquiry* 20(1), 130–134 (1989)
18. Mostowski, A.: On a generalization of quantifiers. *Fundamenta mathematicae* 44, 12–36 (1957)
19. Ogata, N.: Dynamic semantics of plurals dplq. *Electronic notes in theoretical computer science* 67, 263–283 (2002)
20. Qian, S., Amblard, M.: Event in compositional dynamic semantics. *Logical Aspects of Computational Linguistics* pp. 219–234 (2011)
21. Schwertel, U., Hess, M., Fuchs, N.: *Plural Semantics for Natural Language Understanding*. Ph.D. thesis, PhD thesis, Faculty of Arts–University of Zurich, 2005. Available at <http://www.ifi.unizh.ch/attempto/publications> (2003)
22. Strachey, C., Wadsworth, C.: Continuations: A mathematical semantics for handling full jumps. *Higher-order and symbolic computation* 13(1), 135–152 (2000)