On Plurals and Overlay[*]

Massimo Romanelli and **Tilman Becker** and **Jan Alexandersson**
DFKI GmbH
Stuhlsatzenhausweg 3, d-66123 Saarbrücken, Germany
{`romanell,becker,janal`}`@dfki.de`

Abstract

Using an ontology for the representation of user intentions in a multimodal dialogue system is a proven, flexible and powerful approach used in many implementations. Previous work has shown the advantages of representing instances of the ontology as typed feature structures and then using default unification to model the changes of the current user intentions. However, some scenarios require the use of sets to represent plurals and neither can typed feature structures naturally represent sets nor does standard default unification compute the intended results. To address this issue, we propose an extension of overlay by suggesting a representation of plurals, how to identify the intended set manipulations from the linguistic structure, extending the operational semantics for default unification and, finally, how to compute a score mirroring the success of the operation.

1 Introduction

The development of large ontologies makes it possible to use them for not only modeling the domain but also for the representation of the actions and intentions of a user and the system in real dialog systems. In previous work on using ontologies for this purpose, we show how, by simplifying the ontology to typed feature structures (TFS) it is possible to use unification-like operations for constructing new user intentions based on contextual—defeasible—information and new—strict—information (Alexandersson and Becker, 2003). This approach has been previously addressed by, e. g., (Grover et al., 1994) but also recently used in real multimodal systems, such as the MATCH system (Johnston et al., 2002).

However, the task of representing and manipulating plural-like objects is not addressed and we will focus on that in this paper. The treatment of plural-like objects is necessary in systems providing, for instance, the ability to book seats in a movie theater. Pluralities in general is a huge research topic, and we will, in this paper, restrict our focus on concrete, usually definite descriptions of sets.

The following example shows how sets of domain objects, here seats in a movie theater, need to be manipulated to represent the current state of the user intention.

(1) **U:**“I'd like to reserve these ↗[1] seats”.

(2) **S:** Is this OK?

(3) **U:** “No, two more seats here ↗”

The interpretation of the utterance in (3) is obviously something like “I'd like to reserve the seats “the system” understood plus these two I'm pointing at”.

In the following, we assume that meaning is analyzed on a pragmatic level, e.g., as instances of an appropriate ontology, that in turn is represented through typed feature structures (TFS) and that combining new (strict) information as in (3) with the previous context can be computed with default unification. Based on a lattice spanned by subsumption of TFS, (Carpenter, 1993) characterizes credulous default unification as the set of unifiers

---

[*] The research presented here is funded by the German BMBF under grants 01 IL 905 (SmartCom) and 01 IMD 01 (SmartWeb) by the EU under the grant FP6-506811 (AMI).

[1] Where “↗” stands for a pointing gesture, indicating the seats on a floor plan.

between the strict structure and the most special generalizations of the defeasible structure such that there exist a unifier between the strict structure and the generalized defeasible structure.

The representation of sets in TFS has been addressed by (Pollard and Sag, 1994) and recently by (Richter, 2004). The type hierarchy for sets proposed by (Richter, 2004) is shown in figure 1. Clearly, applying default unification on sets represented in this fashion will produce unpredictable results. There are two reasons for this. First, it is unclear how a lattice should be constructed such that the characterization of credulous default unification above produces the intended result. Second, applying our previous algorithm (see (Alexandersson and Becker, 2003)) the background will, in general, be overwritten, since in the subsumption hierarchy the sets are represented as nested sequences and default unification treats them blindly as such.

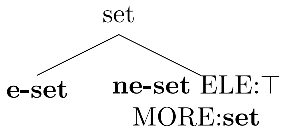

Figure 1: The type hierarchy for sets proposed in (Richter, 2004)

Within the semantic community, there is no generally accepted wide-coverage theory on how to represent and process plurals. The pioneer work on the formal representation and processing of plurals is due to (Link, 1983) where a formal model for the treatment of plurals is developed. His model has received extensions, e. g., (Landman, 1989) as well as critique, e. g., (Copestake, 1992). We will continue along the line of, e. g., (Kamp and Reyle, 1993; Schwarzschild, 1996) and use a union-based model for the representation and manipulation of plural entities.

The paper addresses the following points to include plurals into the previous approach: we identify surface forms relevant for the treatment of plurals and provide a mapping of their meaning to operations on the underlying representations; we extend the ontology and its representation as typed feature structures to represent sets and define the necessary operations on them. The paper provides an extension of the work in (Alexandersson and Becker, 2003), providing an in-context interpretation of plurals. This also requires an extension of scoring which is discussed in section 7.

Note, that this paper focuses on definite descriptions of sets that are changed in the discourse. Although constraints can be expressed as underspecified TFS, arbitrary constraints on a set, e.g., “I want ten or twelve seats somewhere in the front and not on the side.” cannot in general be treated in the same way.

2 Phenomena

We extend the dialogue scenarios set out in (Alexandersson and Becker, 2001) and present a number of examples motivating this work. In a typical dialogue, the user intention is composed from the interpretation of user contributions over several turns. The user adds or changes information either because the system has requested the information, for instance, in order to access a database or because the user has refined or even modified his intention.

By using default unification, it is possible to add information inconsistent with the context in a very natural way:

(4) **U:** “What is running on TV tonight?”

(5) **S:** “Here you see a list of broadcasts for tonight”

(6) **U:** “and tomorrow?” (→ *what is running on TV tomorrow*)[2]

Plurals occurs naturally for, e. g., the reservation of seats in a movie theater:

(7) **U:**“I’d like to reserve these ↗ four seats”.

but also for operating a video recorder:

(8) **U:**“I’d like to record these ↗ broadcasts”.

As soon as we introduce sets into the TFS, default unification will not do the intended thing: using the representation suggested in (Pollard and Sag, 1994; Richter, 2004), the results become unpredictable. For instance, the intended behavior for the following continuation of (7) is not to overwrite the set in the defeasible structure:

(9) **S:** “Is this OK?”

(10) **U:** “No, two more seats here ↗”

Instead, the two sets should rather be combined with set *union*. There are other manipulations possible, such as *subtract*:

(11) **U:** “No, not these ↗two.”

or *overwriting*:

---

[2] In this paper we skip the discourse processing required to resolve, for instance, ellipses. These tasks have been presented elsewhere, e. g., (Löckelt et al., 2002; Pfleger et al., 2003).

(12) **U:** “No, these ↗ instead.”

or even a combination of subtraction and adjoining, which we will call *substitution*:

(13) **U:** “This ↗ one instead of these ↗ two”

It is thus necessary to extend the default unification operation by detecting (and storing) the type of manipulation from the surface structure to the position in the representation where the actual set manipulation takes place, see section 6.

3 Model

In his analysis of plural and mass terms Link (Link, 1983) argues that the relations between individuals and groups in the discourse domain can be captured with two basic operations defining individual sum (**isum**,$\oplus$) and individual part (**ipart**,Π). He observes that $\oplus$ corresponds to the join operation and Π corresponds to a partial ordering relation in a lattice. Moreover, the analysis of Link provides two lattices for the representation of individual and material parts. (Kamp and Reyle, 1993) explicitely adopt Link's approach of a lattice model of plurality and restrict their interest to Link's individual lattice for the representation of count nouns. We follow this approach and retain only the individual lattice. This way the domain of individuals can be represented in a lattice, ordered under $\oplus$ like in figure 2.

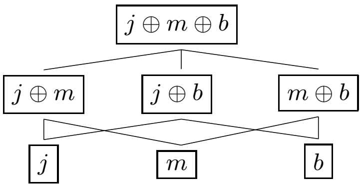

Figure 2: Link's complete join semi-lattice representing plural entities. j,b,m represent respectively the individuals John, Bill and Mari.

Landman (Landman, 1989) introduces an isometric set-theoretic model that captures the properties of the individual lattice in Link's approach. Landman argues that, given a set A of individuals, the power set of A without the empty set—$pow(A)\backslash\{0\}$—has the structure of a complete atomic join semi-lattice as in the Link's formalization of the individual lattice, where $\oplus$ corresponds to **union** and Π to **subset**.

(Schwarzschild, 1996) discusses whether there is a need for structured sets like in Landman's theory—requiring a higher order machinery—or, if it is sufficient, to use simple sets—called the union approach—and seems to advocate the latter. Schwarzschild discusses mostly problems of reference and intra-sentential cases, such as

(14) Tom and the boys reserved tickets.

Here, the question is how the resulting set is represented, and the union theory suggests—using Quine's innovation—that the semantics is $Tom \cup \{b_1, \ldots, b_n\} = \{Tom, b_1, \ldots, b_n\}$. Quite the same considerations on the opportunity of a structured set or simple set representation can be found in (Lasersohn, 1995), whereby the author tends to prefer the union theory solution. In our case, it is even simpler since every position in the ontology (represented as TFS) allows for either singular entities, such as one reservation, or sets of entities such as seats. Therefore, we always represent seats, no matter how many they are, as a set of one or more seats. Hence, we will never have the case that we have to combine a singular—represented as a single item—with a set, or structured sets with each other.

4 Approach

We make use of the representation of sets proposed by (Richter, 2004). Note, that in this representation, each position in the TFS is either of type set or not—it is not possible to represent either a singular or a plural. Also, the nodes of type set are not subject to (default) unification but rather to a special operation (e.g., union) as defined in figure 3. Thus, it is necessary in our approach to find the intended set-manipulation operation (see section 6) during analysis, which we then store to be able to perform the corresponding manipulation. We consider the following possible manipulations:

Union Utterances (7) and (10) are an example of union. The original set is enlarged with two more seats. Our implementation models this as *set union* of the original and the new set.

Difference Utterance (11) is an example of subtraction. This is implemented as *set difference.*

Overwrite See utterance (12). For the implementation, the set in background is discarded and replaced by the set in the covering.

Merge This is a variant of union. Union assumes that the new set is disjoint from that in the background, whereas merge allows for a non-empty intersection. In the implementation this is also realized as *set union.*

Substitute Utterance (13) is an example of substitution, assuming that the background set has more than two seats. Substitution requires that the user identifies *two* new sets: a subset of the set in background and its replacement. It is a combination of difference and union, where the set in context is subtracted by the first set using difference as described above. The result is then combined with the second set using union (set union).

5 Formalization

Our formalization presented here, differs slightly from that taken in (Alexandersson and Becker, 2003) in that we view atomic values as special cases of types, e. g., an integer, say, 42, is a feature-less sub-type of the feature-less type Integer. In this way, a type clash represents the unique source of failure for unification. Furthermore, we assume that there are no re-entrant structures.[3] Operations over set-TFS are specified over the member relation as in (Richter, 2004). Figure 3 gives the formal characterization for, e.g., the union operation. In our approach, testing for membership requires a definition of equality which we replace by unifyability.

Following (Alexandersson and Becker, 2001), the complete overlay consist of two steps:

Assimilation In order to guarantee that the two TFSs are in a direct subsumption relation (*BG* subsumes *CO* or vice versa) in a preprocessing step we perform an assimilation operation over the two TFS as shown in figure 4. The assimilation operation generalizes the *BG* to the least upper bound (**lub**) of the two TFS, unless one subsumes the other. Otherwise it returns the two TFS unchanged. The result of assimilation is passed to the overlay (proper), where the recursive call is performed as shown in definition 1.

Overlay At this point we distinguish two cases: If the assimilated TFS are sets, the *overlaySet* operation is invoked, otherwise we assign the greatest lower bound of *CO* and *BG* (**glb**), i. e., their unifier, as type for the result and recursively build the result TFS. In the *overlaySet* operation we assume a function—*assoc*—that computes the set operation that has to be performed. Thus, depending on the value of the *assoc* operation result, Union, Difference or Overwrite, we respectively perform *set union*, the *set difference*, or we assign *CO* as result for the *overlaySet*.

Definition 1 Overlay
Let $CO = \langle Q_{co}, \bar{q}_{co}, \theta_{co}, \delta_{co} \rangle$ and $BG = \langle Q_{bg}, \bar{q}_{bg}, \theta_{bg}, \delta_{bg} \rangle$[4] be two TFS (covering and background) such that the assimilated TFS are $CO' = \langle Q_{co'}, \bar{q}_{co'}, \theta_{co'}, \delta_{co'} \rangle$ and $BG' = \langle Q_{bg'}, \bar{q}_{bg'}, \theta_{bg'}, \delta_{bg'} \rangle$ f a feature, $Flag \in \{$`Union`, `Difference`, `Overwrite`$\}$ and $assoc(< TFS, Flag >)$ the operation denoting the set operation to be performed, then overlay(CO, BG) is defined as:

```
overlay$(CO, BG)$ :=

    if ($\bar{q}_{co'} \wedge \bar{q}_{bg'} \sqsupseteq$ Set )
    then $overlaySet(CO', BG')$;

    otherwise $overlay'(CO', BG')$;


overlay$'(CO', BG')$ :=    $\langle Q_o, \bar{q}_o, \theta_o, \delta_o \rangle$
        where

        $\bar{q}_o$    $:= \bar{q}_{co}$,
        $\theta o(\bar{q}_o)$    $:= glb(\theta co'(\bar{q}_{co'}), \theta bg'(\bar{q}_{bg'}))$
        $\delta_o(f, \bar{q}_o)$    :=

    if ($f$ in CO$'$ and BG$'$)
    then $overlay(\delta_{co'}(f, \bar{q}_{co'}), \delta_{bg'}(f, \bar{q}_{bg'}))$;

    else if ($f$ exists only in CO$'$)
    then $\delta_{co'}(f, \bar{q}_{co'})$;

    else if ($f$ exists only in BG$'$)
    then $\delta_{bg'}(f, \bar{q}_{bg'})$;

overlaySet$(CO', BG')$    $:= \langle Q_o, \bar{q}_o, \theta_o, \delta_o \rangle = Res$
        where

    if ($assoc(< CO', Flag >) =$ Union)
    then $Res := union(CO', BG')$;

    if ($assoc(< CO', Flag >) =$ Difference)
    then $Res := difference(BG', CO')$;

    if ($assoc(< CO', Flag >) =$ Overwrite)
    then $Res := CO'$;
```

□

Overlaying a set with a non-set TFS will always return the TFS in the cover, so that set-manipulation operations in overlay only apply if the two TFS are typed as set. In an extended definition, Quine's approach, see section 3, can be used to combine a non-set with a set, but only if warranted by the input, i. e., through the *assoc* function. Otherwise any combination of two non-sets

---

[3] We are currently working on an extension of our operational semantics for the correct treatment of such structures together with multiple inheritance as suggested in (Alexandersson and Becker, 2003).

[4] For details of the definition of TFS as graphs, see (Carpenter, 1992) and (Romanelli, 2005)

could be interpreted as union, countermanding our operational semantics, i. e., assimilation.

$$
\begin{aligned}
&\mathit{union}(x,y,z) \overset{\forall}{\Longleftarrow}\\
&\qquad \forall a(\mathit{member}(a,z) \leftrightarrow\\
&\qquad\qquad (\mathit{member}(a,x) \vee \mathit{member}(a,y)))\wedge\\
&\qquad \mathit{set\text{-}properties}({}^{x}[\mathit{set}])\wedge\\
&\qquad \mathit{set\text{-}properties}({}^{y}[\mathit{set}])\wedge\\
&\qquad \mathit{set\text{-}properties}({}^{z}[\mathit{set}])
\end{aligned}
$$

Figure 3: The union operation as in the work of Richter. The *set-properties* relation establishes non-cyclicity, finiteness and unicity.

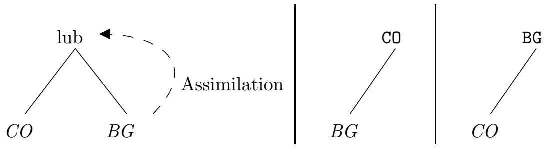

Figure 4: The three different cases for assimilation. lub is the least upper bound of the two TFS.

6 Linguistic Analysis

In this section we evidence the relation between a restricted number of uninflected words and modifications in the structure of plurals. With a lexical analysis that makes use of *lexical indicators* (see figure 5), we establish the relationship between the surface form and the semantic.

We ordered these lexical indicators depending on the operation they signal, union, difference, overwrite and substitute, as shown in table 1.

Some lexical indicators, e.g., “instead” are ambiguous wrt. their intended set operation. “Instead” can stand for a full replacement or only the substitution of a subset of the background. Further analysis is needed to disambiguate, which typically involves the discourse structure. In the case of “instead”, this can be a test whether a subset of the background has been brought into focus before, e.g. “I want these ten seats. OK, but I don’t like these two seats. I want these other two instead.”, or not, e.g. “I want these ten seats. No, I want them in the first row instead.” The former is a case of substitution, the latter of full replacement/overwrite.

We provide a sample grammar for German, see figure 6 based on the German word order analysis provided in (DUD, 1995). The grammar has only illustrative purpose and shows how the linguistic analysis could be performed. Note, that coordination is not covered by the grammar.

	Uni	**Diff**	**Mer**	**Ove**	**Sub**
and also	+	−	−	−	−
instead	−	+	−	+	−
rather	−	+	−	+	−
too	+	+	+	−	−
furthermore	+	−	−	−	−
additionally	+	−	−	−	−
further	+	−	−	−	−
further on	+	−	−	−	−
moreover	+	−	−	−	−
in addition	+	−	−	−	−
aside	+	−	−	−	−
in addition	+	−	−	−	−
instead	−	+	−	+	−
without	−	+	−	−	−
better	−	−	−	+	−
instead of	−	−	−	+	+
in place of	−	−	−	+	+
instead	−	−	−	+	+
alternatively	−	+	−	+	−
rather	−	+	−	+	−
another	+	−	+	−	−
only	−	−	−	+	−
not	−	−	−	+	−
too	+	+	+	−	−

Table 1: Lexical markers associated with set operations.

7 Scoring

The strength of overlay is the combination of default unification together with a scoring function (Pfleger et al., 2002; Alexandersson et al., 2004). The latter is necessary for actually using overlay in real dialogue systems where the analysis components produce multiple hypotheses. In the SmartKom system (Wahlster, 2003), their loci are indeed language and gesture recognition but also language interpretation produces multiple readings. Since default unification always succeeds, the scoring function makes it possible to choose the hypothesis that best fits the context. The score for overlay mirrors how similar two structures are by combining the amount of information stemming from cover or background together with type clashes and conflicting information.[5]

In the co-domain of $[1,-1]$, a positive score means roughly that the result is useful. However, when it comes to combining sets there is a new

---

[5] Note that, given the formalization presented in this paper, conflicting information will never occur. Instead this case is treated as a type clash.

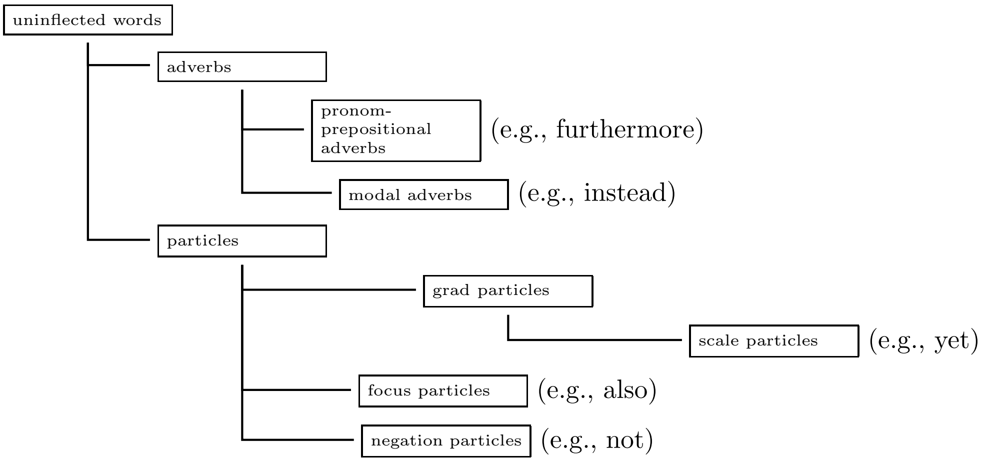

Figure 5: Word categories relevant to the modification of plural entities.

```
<S> ::= <frontfield>  <centerfield>  <endfield>

<frontfield> ::= [<np> <pf>]

<middlefield> ::= <pb> | <pa>

<endfield> ::= [<if>]

<pb> ::=   <particle> <deix> [<num>] [<obj>] [<gesture>]
        | <particle> <num> [<obj>] [<gesture>]
        | <particle> <gesture>

<pb> ::=   <deix> [<num>] [<obj>] [<gesture>] <particle>
        | <num> [<obj>] [<gesture>] <particle>
        | <gesture> <particle>

<np> ::= "ich"                                # I
<pf> ::= "moechte"                            # woud like
<particle> ::= "auch"                         # also
      | "ausserdem"                           # besides
      | "daneben"                             # aside
      | "dazu"                                # more
      | "desweiteren"                         # in addition
      | "hinzu"                               # to add
      | "ferner"                              # moreover
      | "noch"                                # another
      | "ueberdies"                           # too
      | "weiter"                              # further
      | "weiterhin"                           # further on
      | "zudem"                               # furthermore
      | "zusaetzlich"                         # additionally
<deix> ::= "diese"                            # these
<num> ::= "zwei"                              # two
<obj> ::= "plaetze"                           # seats
<gesture> ::= "hier"                          # here
<if> ::= "reservieren"                        # book
```

Figure 6: A grammar for the recognition of the Union case.

Op.	**ScoreFn**	**Cons.**	**Incons.**
Union:	$\|CO \cap BG\|$	2	1,4,5
Diff.:	$\|CO \setminus (CO \cap BG)\|$	4,5	1,2
Overw.:	$\|CO \cap BG\|$	2	1,4,5

Figure 7: The case analysis for the scoring operation for *overlaySet* depending on the relations between CO and BG, and on the set operation that has been performed. The **ScoreFn** column contains the operations used in the scoring functions. The numbers in the **Consistent** and **Inconsistent** columns refer to the cases depicted in figure 8, e. g., case 2 in figure 8 is consistent for **Union**.

dimension. A set operation, e.g., difference, can be more or less successful. Suppose that an interpretation of (10), see section 2, contains seats not present in the cover. This indicates either that the interpretation was suboptimal, or it can be the case that the user said something inconsistent. The general schema for the possible relations between two sets is shown in figure 8.

The type of relation has different influence, depending on the type of set operation. For instance, the second case (2. in figure 8) is consistent for the union operation but not for difference. Consistent relations for the latter operation are 4. and 5., which are inconsistent for the former. Figure 7 shows the relation between the cases listed in figure 8 and the operations presented in section 1.

We supply two scoring functions

setScore for the inclusion into the overall score. setScore computes a value in the co-domain of $[-1, 1]$.

conScore (consistency score) for indicating that the operation has was inconsistent. conScore computes a value in the co-domain of $[-1, 1]$, where everything that is below or equal to 0 is more or less inconsistent. Note that it is up to the dialogue manager to utilize this information.

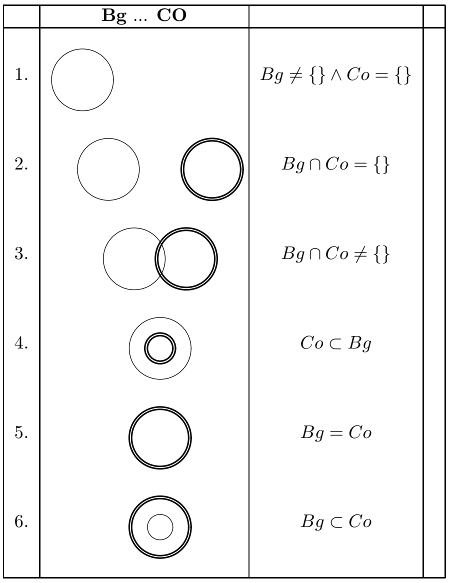

Figure 8: The different relations between the background and cover set respectively.

Definition 2 setScore

$$setScore(CO, BG, op) = \begin{cases} \frac{|CO|-(|CO\cap BG|*2)}{|CO|} & op = union \text{ or } overwrite \\ -1 & |CO| = 0 \\ \frac{|CO|-(|CO\setminus(CO\cap BG)|*2)}{|CO|} & \text{otherwise} \end{cases}$$

$\square$

Definition 3 conScore

$$conScore(CO, BG, op) = \begin{cases} \frac{2}{|CO\cap BG|+1} - 1 & op = union \text{ or } overwrite \\ -1 & |CO| = 0 \\ \frac{2}{|CO\setminus(CO\cap BG)|+1} - 1 & \text{otherwise} \end{cases}$$

$\square$

7.1 An example

To further highlight the behavior of our scoring functions, suppose the user has successfully selected five seats in a movie theater for reservation but suddenly remembers that the complete family should join and tries to add five more seats to the initial intention. The relation between the initial set and the new one will depend on the size of the intersection. The outcome of our scoring functions is depicted in figure 9.

For *conScore* we have 1 for the case that the intersection is empty and a value of 0 or less in case it is non-empty. *setScore* on the other hand is a linear function that uses the complete co-domain $[1, -1]$.

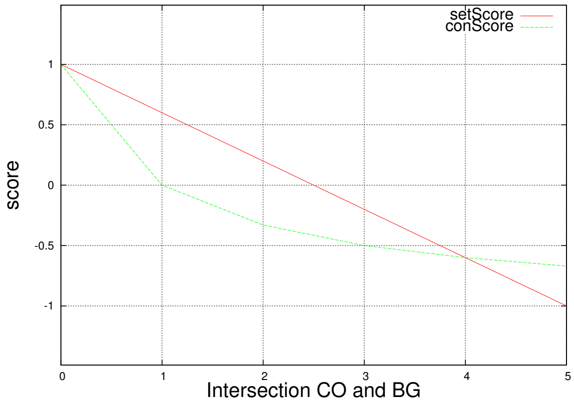

Figure 9: A plot representing the results of the different scoring operations depending on the size of the intersection.

Finally, it should be noted that the usage of the scoring functions, in particular, the conScore depends on the domain we are working with. If the task is to manipulate, say, grains on a plate, it might not be that inconsistent to make erroneously manipulations. For the selection of seats in a movie theater, however, every seat is important.

8 Conclusion

For the purpose of dialogue systems using a large ontology for the representation of user intentions, we have extended overlay (credulous default unification for TFS with scoring) to cope with sets. Sets are the natural modeling for plurals. We indicated how to extract information about the intended set manipulation from the surface structure. Additionally, we have sketched different possibilities to extend the scoring mechanism proposed in (Pfleger et al., 2002). Currently ongoing work is concerned with re-entrancy, the next step will be an extension of this work to general constraints on sets.

References

Jan Alexandersson and Tilman Becker. 2001. Overlay as the Basic Operation for Discourse Processing in a Multimodal Dialogue System. In *Workshop Notes of the IJCAI-01 Workshop on "Knowledge and Reasoning in Practical Dialogue Systems"*, Seattle, Washington, August.

Jan Alexandersson and Tilman Becker. 2003. The Formal Foundations Underlying Overlay. In *Proceedings of the Fifth International Workshop* *on Computational Semantics (IWCS-5)*, pages 22–36, Tilburg, The Netherlands, February.

Jan Alexandersson, Norbert Pfleger, and Tilman Becker. 2004. Scoring for overlay based on informational distance. In *KONVENS-04*, pages 1–4, Vienna,Austria, September 14 - 17.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures.* Cambridge University Press, Cambridge, England.

B. Carpenter. 1993. Skeptical and Credulous Default Unification with Applications to Templates and Inheritance. In T. Briscoe, V. de Paiva, and A. Copestake, editors, *Inheritance, Defaults, and the Lexicon*, pages 13–37. Cambridge University Press, Cambridge, CA.

A. Copestake. 1992. *The Representation of Lexical Semantic Information.* Doctoral dissertation, University of Sussex.

1995. *DUDEN - Die Grammatik*, volume 4 of *Der Duden in zwölf Bänden.* Dudenverlag, Mannheim.

Claire Grover, Chris Brew, Suresh Manandhar, and Marc Moens. 1994. Priority Union and Generalization in Discourse Grammars. In *Proceedings of the* 32^{nd} *Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Las Cruces, New Mexico.

Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. MATCH: An Architecture for Multimodal Dialogue Systems. In *Proceedings of the* 42^{th} *Annual Meeting of the Association for Computational Linguistics, ACL'02*, pages 376–383, Philadelphia.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publishers, London, Boston, Dordrecht.

Fred Landman. 1989. Groups I. *Linguistics and Philosophy*, 12:558–605.

Peter Lasersohn. 1995. *Plurality, Conjunctions and Events*, volume 55 of *Studies in Linguistics and Philosophy.* Kluwer.

Godehard Link. 1983. The Logical Analysis of Plurals and Mass Terms: A Lattice-theoretical Approach. In Rainer Bäuerle, Christoph Schwarze, and Arnim vov Stechow, editors, *Meaning, Use and Interpretation of Language*, pages 303–323. de Gruyter.

Markus Löckelt, Tilman Becker, Norbert Pfleger, and Jan Alexandersson. 2002. Making Sense of Partial. In *Proceedings of the sixth Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)*, pages 101–107, Edinburgh, UK, September.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2002. Scoring Functions for Overlay and their Application in Discourse Processing. In *KONVENS-02*, pages 139–146, Saarbrücken, September, 30–October, 2.

Norbert Pfleger, Ralf Engel, and Jan Alexandersson. 2003. Robust Multimodal Discourse Processing. In Kruijff-Korbayova and Kosny, editors, *Proceedings of Diabruck: 7th Workshop on the Semantics and Pragmatics of Dialogue*, Wallerfangen, Germany, September.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* Studies in Contemporary Linguistics. Chicago Uni Press, Chicago - London.

Frank Richter. 2004. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar.* Phil. dissertation, Eberhard-Karls-Universität Tübingen. Version of April 28th, 2000. Superseded by Richter 2004.

Massimo Romanelli. 2005. Ontology-based representation and processing of plurals for human-machine dialogue systems with unification-based operations. Master's thesis, University of Saarland.

Roger Schwarzschild. 1996. *Pluralities*, volume 61 of *Studies in Linguistics and Philosophy.* Kluwer.

Wolfgang Wahlster. 2003. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In R. Krahl and D. Günther, editors, *Proceedings of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin: DLR, June.