Efficient Generation of Descriptions in Context

Emiel Krahmer & Mariët Theune
IPO, Center for Research on User-System Interaction,
Eindhoven University of Technology
{e.j.krahmer/m.theune}@tue.nl

CONTENTS

1 Introduction **2**

2 The Incremental Algorithm (Dale & Reiter 1995) **3**

3 A Modification of the Algorithm Based on Salience **4**
3.1 Introduction 4
3.2 Preliminaries 5
3.3 Outline of the Modified Algorithm 6
3.4 Examples 8
3.5 Discussion 9

4 Determining Salience Weigths **10**
4.1 Hajičová: Hierarchical Focus Constraints 10
4.2 Grosz *et al.*: Centering 10
4.3 Examples, Predictions and Comparison 11
4.4 Discussion 13

5 Further extensions **13**
5.1 Pronominalization 13
5.2 Relational Descriptions 14
5.3 Bridging Descriptions 19

6 Concluding remarks **19**

1 INTRODUCTION

In their interesting 1995 paper, Dale & Reiter present various algorithms they developed alone or in tandem to determine the content of a *distinguishing description*. That is: a definite description which is an accurate characterization of the entity being referred to, but not of any other object in the current 'context set'. They argue that their *Incremental Algorithm*, discussed in more detail below, is the best one from a computational point of view (it has a low complexity and is fast) as well as from a psychological point of view (humans appear to do it in a similar way).

Even though Dale & Reiter (1995) primarily aimed at investigating the computational implications of obeying the Gricean maxims,[1] the Incremental Algorithm has become more or less accepted as the state of the art for generating descriptions. However, due to their original motivation, various other aspects of the generation of definites remained somewhat underdeveloped. In this paper, we flesh out a number of these aspects, *without losing sight of the attractive properties of the original algorithm* (speed, complexity, psychological plausibility).

The basic idea we want to pursue is that a definite description refers to the *most salient* element satisfying the descriptive content (Lewis 1979: 348-350, see Krahmer 1998 for a formalization in dynamic semantics). Lewis mentions the following example (due to McCawley):

(1) The dog got in a fight with another dog.

Lewis notes that this statement can only be true in a domain which contains at least two dogs, which entails that *the dog* cannot be a distinguishing description. According to Lewis, (1) means that the most salient dog got in a fight with some less salient dog. Lewis does not mention descriptions which refer to 'unique' objects, but it is readily seen that they can also be understood in terms of salience: if there is only one object with the relevant properties, it *has* to be the most salient one. Arguably, a notion of salience is implicit in Dale & Reiter's usage of context sets, and also extensions such as Horacek (1997:210) and Stone & Webber (1998:183) explicitly remark that some form of salience is important. None, however, specify how salience should be determined nor which repercussions the inclusion of a notion of salience has for the generation algorithm.[2]

In this paper we show that it is possible to integrate an explicit notion of salience into Dale & Reiter's Incremental Algorithm and that this paves the way for the context-sensitive generation of definite descriptions (section 3). In section 4, two ways of assigning salience weights to objects are discussed, one based on the

[1] For a good discussion of this aspect of their work, see Oberlander 1998.

[2] Horacek (1997) and Stone & Webber (1998) are closely related in spirit to the current work. Horacek (1997) makes various interesting observations about the limitations of the Incremental Algorithm and its ilk and proposes a new algorithm with 'flexible interfaces' to other modules. Many of the issues discussed by Horacek are also addressed in this paper, and some of his suggestions are taken over, in particular the integration of linguistic constraints during generation. This integration is also a central ingredient of Stone & Webber (1998), who go one step further and argue for the simultaneous inclusion of semantic and pragmatic information as well. While both approaches look promising, it is difficult to make a precise judgement of their respective performance and predictions since central components (e.g., selection of properties, salience determination) are left unspecified. As a consequence, the computational properties of their algorithms are not clear (see Horacek 1997:212 and Stone & Webber 1998:186).

hierarchical focusing constraints of Hajičová (1993), the other on the constraints of Centering Theory (Grosz *et al.* 1995). In section 5, a number of extensions of the modified Incremental Algorithm are described. To begin with, the presence of a principled way of salience weight assignment makes it possible to take the pronominalization decision within the algorithm (section 5.1). A more substantial extension is offered in section 5.2, where it is shown how the algorithm can be extended so as to be able to generate relational descriptions in an efficient and flexible way. Finally, in section 5.3 it is argued that the two previous extensions make it possible to generate bridging descriptions. In the concluding section we describe, among other things, some issues related to the implementation of the algorithm and its integration in a data-to-speech system (i.e., a system which converts structured, non-linguistic data into spoken language).

2 The Incremental Algorithm (Dale & Reiter 1995)

Suppose, for the sake of illustration, that we have a domain D_1 consisting of the following four entities $d_1 - d_4$.

d_1 ⟨ type, chihuahua ⟩, ⟨ size, small ⟩, ⟨ colour, black ⟩
d_2 ⟨ type, chihuahua ⟩, ⟨ size, large ⟩, ⟨ colour, white ⟩
d_3 ⟨ type, siamese cat ⟩, ⟨ size, small ⟩, ⟨ colour, black ⟩
d_4 ⟨ type, poodle ⟩, ⟨ size, small⟩, ⟨ colour, white ⟩

It should be stressed that the approaches described in this paper apply to *any* domain which meets the following criteria (Dale & Reiter 1995:254): (*i*) each entity in the domain is characterized by a list of attribute value pairs, or *properties*, (*ii*) each entity has at least an attribute *type* and (*iii*) there may be a subsumption hierarchy on the values of certain attributes. Here it is assumed that there is such a hierarchy for the attribute 'type': 'dog' subsumes 'chihuahua' and 'poodle', 'cat' subsumes 'siamese cat' and 'mammal' in its turn subsumes 'dog' and 'cat'. Additionally, there can be a *basic level value* (e.g., Rosch 1978) for a certain attribute. Following Dale & Reiter, we assume that the basic level values for 'type' are 'dog' and 'cat' respectively.[3]

The input for the Incremental Algorithm is an object r, a set C (the context set) consisting of alternative objects from which r has to be distinguished (the *distractors*) and, crucially, a list of preferred attributes. This list contains, in order of preference, the attributes that human speakers and hearers prefer for a particular domain. For instance, it seems likely that a human speaker would first try to describe an animal by its 'type' (is it a dog? is it a cat?), and if that doesn't help attributes like 'colour' and 'size' may be used. It is reasonable to assume that speakers have a general preference for *absolute* properties such as 'colour', which are easily observed without taking the other objects into account, over *relative* properties such as 'size',

[3] According to the categorization theories of Rosch and others, the basic levels are fixed. For instance, they are the first levels which children learn to understand and use. Additionally, and most relevant for current purposes, basic levels are normally used in neutral contexts. Lakoff (1987:42): "*For example,* There's a dog on the porch *can be used in a neutral context, whereas special contexts are needed for* There's a mammal on the porch *or* There's a wire-haired terrier on the porch. *(See Cruse 1977).*" From the perspective of generation, the basic level values might also be viewed as user-dependent (Dale p.c.). For instance, it seems likely that for a professional dog-breeder the 'basic level' is below 'dog' in the subsumption hierarchy.

which are less easily observed and always require inspection of the distractors.[4] Thus let us assume that the list of preferred attributes for the example domain is $\langle$ type, colour, size, …$\rangle$. Essentially, the Incremental Algorithm goes through the list of preferred attributes, and for each attribute it encounters it finds the *best value* of this property. The best value of a property is the value closest to the basic level value such that none of the values it subsumes rule out *more* objects in the domain. If adding this best value to the already selected properties has the effect of ruling out any of the remaining distractors, it is included in the list of properties to be used in the generation of the distinguishing description. The algorithm stops when the end of the list of preferred attributes is reached (failure), or when the list of distractors is empty (success). In the latter case, it is checked whether the 'type' property was included, and if not, its basic value is added to the selected list of properties.

Dale & Reiter (1995:247) argue that this algorithm has a polynomial complexity and the theoretical run time can be characterized as $n_d n_l$: the run time depends solely on the number of distractors n_d and the number of iterations (i.e., selected properties) n_l. This means that the Incremental Algorithm is the fastest algorithm discussed in Dale & Reiter (1995). One of the central features of the Incremental Algorithm is that there is no backtracking (hence the term 'incremental'): once a property p has been selected, it will be realised in the final description, even if a property which is added later would render the inclusion of p redundant with hindsight. This aspect is partly responsible for the efficiency of the algorithm, but Dale & Reiter additionally claim that this property is 'psychologically realistic' since human speakers also often include redundant modifiers in their referring expressions (see e.g., Pechmann 1989).[5]

3 A Modification of the Algorithm Based on Salience

3.1 Introduction

The contents of the description for an object which the incremental algorithm outputs are to a large extent determined by the context set. Nevertheless, Dale & Reiter do not address the question how such context sets are constructed nor how their contents can be updated during the generation process. They only write:

> We define the context set to be the set of entities that the hearer is currently assumed to be attending to; this is similar to the set of entities

[4] The literature on perception contains a wealth of empirical evidence for such general orderings, see, for instance, Pechmann (1989), Levelt (1989) and, more recently, Beun & Cremers (1998).

[5] Dale & Reiter (1995:248): *"For example, in a typical experiment a participant is shown a picture containing a white bird, a black cup, and a white cup and is asked to identify the white bird; in such cases, participants generally produce the referring expression* the white bird, *even though the simpler form* the bird *would have been sufficient."* Yet, it seems to us that the Incremental Algorithm would produce the description *the bird* in this situation: if we make the natural assumption that 'type' is the most preferred attribute, the property 'bird' will be the first one selected and immediately rules out the black and the white cup. In general, it should be noted that Pechmann's notion of incrementality refers to *speech production*; the fact that speakers, when describing an object, start uttering properties of that object without making sure whether these are actually distinctive or not. Dale & Reiter's incrementality refers to the lack of backtracking in property selection, but the order in which properties are selected is not related to the order in which properties are realized in speech. It is worth noting that full incrementality in the latter sense (each property is uttered as soon as it is selected), cannot be obtained without taking a certain amount of lookahead into account (Levelt 1989).

> in the focus spaces of the discourse focus stack in Grosz and Sidner's (1986) theory of discourse structure. (Dale & Reiter 1995:236)

Dale (1992:192-193) is somewhat more explicit and briefly describes a full and a partial order on focus spaces. However, he concludes that:

> [i]n the present domain [recipes, K&T], since the number of entities we are dealing with is relatively small, it is adequate to take the global working set to be the context.

(The 'working set' refers to the list of "identifiable distinct objects in the domain at any point in time," Dale 1992:56.) In this section our aim is to be explicit about the continuously changing contents of context sets and the repercussions this has for the Incremental Algorithm. It is instructive at this point to look at some of the options we have at our disposal. We assume that there always has to be a domain of discourse D: the set of objects which can be referred to (in a data-to-speech system this set is given as part of the data). Some people have suggested that a context set C may be a proper subset of D, containing those objects of D which have been referred to before. But, suppose for the sake of argument that D contains three objects: two dogs and a cat, and that the cat has just been mentioned. How should we restrict the context set? We cannot restrict C to the cat, because then we cannot refer to the dogs anymore, and for similar reasons we cannot restrict C to the dogs. This suggests that it is not feasible to dynamically reduce or enlarge the context set. Rather it should be a structured whole, containing precisely the objects in the domain and combined with a method to mark certain objects as more prominent than others. As our metric of prominence we shall use *salience weights*.

3.2 Preliminaries

The underlying idea of our modifications is the following:

> A definite description 'the $\overline{\text{N}}$' is a suitable description of an object d in a state s iff d is the most salient object with the property expressed by $\overline{\text{N}}$ in state s

Since the denotation of the $\overline{\text{N}}$ is an important factor, we use the notion of a *value set*. Let L be the list of properties expressed by some $\overline{\text{N}}$. The value set of L in some domain D (notation: $\mathsf{Val}_D(L)$) is the set of objects $d \in D$ which have the properties expressed by L.[6] More formally:

DEFINITION 1 (Value sets)
$\mathsf{Val}_D(\langle A, V\rangle) = \{d \in D \mid d \text{ has the property expressed by } V\}$,
$\mathsf{Val}_D(\{p_1, \ldots, p_n\}) = \mathsf{Val}_D(p_1) \cap \ldots \cap \mathsf{Val}_D(p_n)$, where $p_i = \langle A_i, V_i\rangle (1 \leq i \leq n)$.

Thus: $\mathsf{Val}_{D_1}(\{\langle \text{colour}, \text{white}\rangle, \langle \text{type}, \text{chihuahua}\rangle\}) = \mathsf{Val}_{D_1}(\langle \text{colour}, \text{white}\rangle) \cap \mathsf{Val}_{D_1}(\langle \text{type}, \text{chihuahua}\rangle) = \{d_2, d_4\} \cap \{d_1, d_2\} = \{d_2\}$. The domain subscript

[6] The use of value sets marks a minor deviation from Dale & Reiter. Whereas they use a function $\mathsf{RulesOut}$ which determines the objects which do *not* have a certain property p, we check which objects *do* have this property. This difference is akin to the difference between a cup which is half full and one which is half empty. We find the use of value sets somewhat more intuitive, as they provide a first connection with the semantics.

and the attributes are omitted when this does not lead to confusion; e.g., we write $\mathsf{Val}(\text{small, chihuahua})$. By definition, the value set of the empty list of properties is the entire domain ($\mathsf{Val}_D(\{\}) = D$). Following common practice, we use $|S|$ to denote the cardinality of a set S.

How can we model the salience of an entity? For that purpose we use a function variable sw (salience weight) which per state represents a function mapping elements in the domain to a natural number.[7] For the sake of simplicity, we shall assume that in the initial state (s_0), say the beginning of the generation process, all entities are minimally salient (represented as a zero salience level). Formally, $\forall d \in D : sw_0(d) = 0$. When this can be done without creating confusion, the index on sw_i is suppressed. Below, in section 4, we discuss and compare two methods for salience weight assignment. For the time being we simply assume that the salience weights are given. So let sw_s be the function assigning salience weights in state s, then we can define that an object r is the most salient object having certain properties L in a state s (notation: $\mathsf{MostSalient}(r, L, s)$) if, and only if, every object in $\mathsf{Val}(L)$ different from r has a lower salience weight in s than r itself.

DEFINITION 2 (Salience condition)
$\mathsf{MostSalient}(r, L, s) \Leftrightarrow \forall d \in \mathsf{Val}(L)(d \neq r \rightarrow sw_s(d) < sw_s(r))$

3.3 Outline of the Modified Algorithm

Figure 1 contains our proposal for a modified algorithm in pseudo-code. We have stuck as closely as possible to the algorithm from Dale & Reiter (1995:257) to ease comparison.[8] Below, we illustrate it with a number of examples. First, we give a general, somewhat informal overview.

The algorithm is called by $\mathsf{MakeReferringExpression}\ (r, P, s)$; that is, we try to generate a definite description for a referent r given some pre-defined list P of *preferred attributes* in a state s (where s is a pointer to the sw_s function). L is the list of properties which have been selected for inclusion in the expression generated and is initialized as the empty list. The variable *tree* contains the syntactic tree for the NP under construction which corresponds with the current list of properties L. Finally, *contrast* is a boolean variable which indicates whether the property under consideration is contrastive or not. As in the original version of the algorithm, the main loop iterates through the list P of preferred attributes. For each attribute A on this list, the best value V is sought (essentially in the same way as done in the Incremental Algorithm). Once the best value V is found, it is checked whether adding the property $\langle A, V\rangle$ to the list of already selected properties 'shrinks' the value set (and thus rules out one or more distractors). If this is so, the function $\mathsf{Contrastive}$ (r, A, V) checks whether the property under consideration is contrastive, i.e., if it serves to distinguish the object r in some linguistic context LC. LC is defined as the

[7] It is also possible to assign salience weights to *groups* of objects (such as focus spaces). Notice that the point-wise assignment can be mapped onto a group-wise assignment, but not vice versa. The additional information that point-wise assignments have is potentially useful for the sake of pronominalization. We do not believe, however, that there is a knock-down argument for either of the alternatives. For that the two are too similar.

[8] We also adopt the notation employed by Dale & Reiter, which is essentially the `WEB` style from Knuth (1986:viiff).

```
┌─────────────────────────────────────┐
│ MakeReferringExpression (r, P, s)   │
└─────────────────────────────────────┘
L ← {}, tree ← nil, contrast ← false
for each member A_i of list P do
    V ← FindBestValue (r, A_i, BasicLevelValue(r, A_i), s)
    if |Val(L ∪ {⟨A_i, V⟩})| < |Val(L)| ∧
      contrast ← Contrastive(r, A_i, V) ∧
      (tree ← UpdateTree(tree, V, contrast)) ≠ nil
    then L ← L ∪ {⟨A_i, V⟩}
    endif
    if MostSalient (r, L, s) = true
    then if ⟨type, X⟩ ∈ L for some X
            then tree ← AddDefDet (tree) ∧
                 return tree
            else V ← BasicLevelValue(r, type) ∧
                 (tree ← UpdateTree (tree, V, false)) ≠ nil ∧
                 tree ← AddDefDet (tree) ∧
                 return tree
            endif
   endif
return failure

┌──────────────────────────────────────┐
│ FindBestValue(r, A, initial-value, s) │
└──────────────────────────────────────┘
if UserKnows(r, ⟨A, initial-value⟩) = true
then value ← initial-value
else value ← nil
endif
if MostSalient(r, {⟨A, value⟩}, s) = false ∧
   (msv ← MoreSpecificValue(r, A, value)) ≠ nil ∧
   (new-value ← FindBestValue (r, A, msv, s)) ≠ nil  ∧
   |Val({⟨A, new-value⟩})| < |Val({⟨A, value⟩})|
then value ← new-value
endif
return value

┌────────────────────┐
│ MostSalient(r, L, s) │
└────────────────────┘
if ∀d ∈ Val(L)(d ≠ r ⇒ sw_s(d) < sw_s(r))
then return true
else return false
endif

┌─────────────────────┐
│ Contrastive(r, A, V) │
└─────────────────────┘
LC ← {d ∈ DR(PrevS ∪ CurrS) | d ≠ r ∧
                  Parent(BasicLevelValue(d, type)) =
                  Parent(BasicLevelValue(r, type))}
if ∃d ∈ LC : Value(d, A) ≠ V
then return true
else return false
endif
```

Figure 1: Full sketch of the modified algorithm.

set of objects/discourse referents (DRs) which are referred to in the previous sentence (PrevS) or in the sentence currently being generated (CurrS), and of which the basic level value of the attribute 'type' has the same parent as that of r. A property $\langle A, V \rangle$ of r is considered to be contrastive if there is an element $c \in LC$ which has a different value for the current attribute.[9] Subsequently, the algorithm tries to incorporate V in the NP under construction, using the function UpdateTree*(tree, V, contrast)*.[10] If this does not succeed (the lexical or syntactic restrictions of the generation module make it impossible to express the property), V is rejected. If it *does* succeed, the current property is added to the list of selected properties. Then it is checked whether the intended referent r is the most salient object in the current state of the discourse which satisfies L. If so, the algorithm succeeds. It realises the type in the constructed NP if it is not already there, after which the function AddDefDet inserts the determiner *the* to produce a full definite description.

3.4 Examples

First example: non-anaphoric description Reconsider our example domain D_1, and suppose that we start generating a monologue in the initial state s_0. Thus, by assumption, all elements of the domain are equally salient. Now we want to generate an expression for d_2. MakeReferringExpression (d_2, P, s_0) is called (assuming that P is $\langle$ type, colour, size, ...$\rangle$). The list of properties L is initialized as $\{\}$, *tree* as nil and *contrast* as **false**. We consider the first property of d_2, $\langle$ type, chihuahua $\rangle$. The best value for this attribute is 'chihuahua', since $|\mathsf{Val}(\text{chihuahua})| = 2 < |\mathsf{Val}(\text{dog})| = 3$. This property has sufficient descriptive content to be included in the description under construction: $|\mathsf{Val}(\text{chihuahua})| = 2 < |\mathsf{Val}(\{\})| = 4$. As a result the function UpdateTree is called which incorporates the type into the description under construction and returns tree (I) from figure 2. The value of L is now $\{\langle$ type, chihuahua $\rangle\}$. MostSalient$(d_2, \text{chihuahua}, s_0)$ fails because d_1 is also a chihuahua, and d_1 and d_2 are both minimally salient by assumption. So we proceed by taking the second property of d_2, $\langle$ colour, white $\rangle$. Now $|\mathsf{Val}(\text{white}, \text{chihuahua})| < |\mathsf{Val}(\text{chihuahua})|$; this property is discriminating and again we call the function UpdateTree which adds an AP for the property 'white' to the current NP tree (II, figure 2). Now, MostSalient$(d_2, \{\text{white, chihuahua}\}, s_0)$ *is* true: d_2 is the only white chihuahua in the domain, so it is by definition the most salient one. Since the type of d_2 is present in the constructed NP tree, the definite article is added, and the resulting tree (III, figure 2) is returned. When the description is conveyed, d_2 increases in salience, becoming more salient than the other objects in the domain.

---

[9]Thus, loosely speaking, the adjective *large* in the NP *the large dog* is marked as contrastive in the context of *a small cat* but not in the context of *a small car*. This treatment of contrast is closely related to the proposal of Prevost (1996), who presents an algorithm for deciding which properties should receive contrastive accent in a manner which is somewhat similar to the Incremental Algorithm. See Theune (1997) for some further discussion on Prevost's approach to contrast.

[10]In contrast with the rest of the algorithm, this function is largely domain and language dependent. Essentially, starting from a prototypical NP structure, the function attempts to integrate each new value V in the syntactic tree constructed so far. In general, the value of the 'type' attribute is realised as the head noun. Unary properties are added, in the order of selection, as prenominal AP modifiers. Relations (discussed in section 5.2) may be realized as postnominal PPs or relative clauses. If *contrast* is **true**, the expression of the value V is marked by a [+c] feature, which can be taken into account during the computation of prosody in a spoken language generation system. Following Horacek (1997),

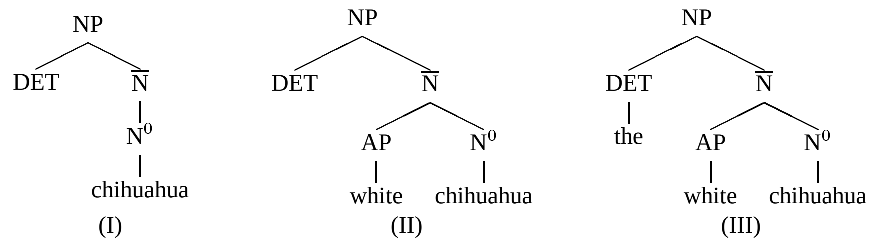


Figure 2: The three stages of generating the first example description.

Notice that when we assume that all entities in the domain are always equally salient (thus sw represents the constant function mapping each $d \in D$ to some n), the modified algorithm selects exactly the same properties as the original Incremental Algorithm. In other words, our version truly generalizes the original, which brings us to the next example.

Second example: anaphoric description An anaphoric description generally contains less information than its antecedent. This may be reflected by the omission of properties, by the use of a more general head noun, or by both, as this example shows. Suppose we want to refer to d_2 in a situation s' where d_2 is more salient than the other animals in the domain (e.g., because d_2 was referred to in the previous sentence). The BasicLevelValue for d_2 with attribute 'type' is 'dog', and since d_2 is currently the most salient dog in the domain, the modified Incremental Algorithm immediately succeeds and returns a tree for *the dog*.

Third example: contrast Now suppose that the preceding sentence referred to both d_1 and d_2 (e.g., '*the black chihuahua and the white chihuahua* ...') and that we call MakeReferringExpression (d_2, P, s). Then the first property added to the description under construction will be 'chihuahua' (as the reader may check for herself). The second property, 'white' is also added to the description, since it distinguishes d_2 from d_1. Moreover, this property is *contrastive*, since d_1 and d_2 have the same basic level value for the 'type' attribute and different values for the 'colour' attribute. The result is '*the white*$^{[+c]}$ *chihuahua*' (where [+c] is a feature indicating a contrastive relation).

3.5 Discussion

We have described a generalization of the Incremental Algorithm which extends the original in a number of ways. First and foremost, it explicitly takes the discourse context into account by treating context sets as a combination of a discourse domain with a salience function. This generalization entails that creating a referring expression for one object (here d_2 of domain D_1) can result in e.g., *the white chihuahua* (in the initial state), *the dog* (in a context like *the white chihuahua* ...), *the dog*$^{[+c]}$ (sample context: *the white chihuahua and the cat*) and *the white*$^{[+c]}$ *chihuahua* (context: *the black chihuahua and the white chihuahua* ...). This does not make the algorithm more complex: it still has a polynomial complexity, and a sim-

we allow each available slot in the prototypical NP structure to be filled only once.

ilar theoretical run time as the original algorithm ($n_d n_l$).[11]

4 Determining Salience Weigths

So far we did not discuss how salience weights are computed. The literature contains various methods to do so, such as Alshawi (1987), Hajičová (1993), Lappin & Leass (1994), and Grosz *et al*. (1995). In principle any of these methods will do; it is not our intention to argue in favour of any specific approach. Our main point here is that the modified algorithm can be associated with more than one principled way of determining salience weigths. To illustrate this we discuss the hierarchical focusing constraints of Hajičová (1993) and the Centering approach of Grosz *et al*. (1995). Throughout this section we let sw_i be a total function mapping the objects in a domain D to the set $\{0, \ldots, 10\}$, with the intuition that 0 represents complete non-salience and 10 maximal salience.[12]

4.1 *Hajičová: Hierarchical Focus Constraints*

The salience function given in definition 3 below closely corresponds to the rules given by Hajičová (1993), except that in our version entities are mapped to a *higher* number as they become more salient, with a maximum of 10, whereas in Hajičová (1993) entities are assigned a *lower* number as they become more salient, maximal salience being indicated by 0. As a consequence, the Praguian approach allows for an infinite decrease in salience weight, keeping track of objects which have faded from the discourse long ago. In our version, the salience weight of an entity cannot decrease below zero.

Definition 3 (Salience weight assignment based on Hajičová 1993)
Let U_k be a sentence uttered in the context of sw_i, and let $\mathsf{topic}(U_k) \subseteq D$ and $\mathsf{focus}(U_k) \subseteq D$ be the sets of entities which are referred to in the topic and the focus part of U_k respectively.[13] Then the new salience function sw_{i+1} is defined as:

$$sw_{i+1}(d) = \begin{cases} 10 & \text{if } d \in \mathsf{focus}(U_k) \\ 9 & \text{if } d \in \mathsf{topic}(U_k) \text{ and } d \text{ is referred to by a definite NP} \\ sw_i(d) & \text{if } d \in \mathsf{topic}(U_k) \text{ and } d \text{ is referred to by a pronoun} \\ \mathsf{max}(0, sw_i(d) - 1) & \text{if } d \notin \mathsf{topic}(U_k) \cup \mathsf{focus}(U_k) \text{ and } d \in \mathsf{topic}(U_l), l < k \\ \mathsf{max}(0, sw_i(d) - 2) & \text{if } d \notin \mathsf{topic}(U_k) \cup \mathsf{focus}(U_k) \text{ and } d \in \mathsf{focus}(U_l), l < k \end{cases}$$

4.2 *Grosz et al.: Centering*

An alternative definition of salience weight assignment can be based on the ranking of the so-called *forward looking centers* (C_f) of an utterance. According to Centering Theory (Grosz *et al.* 1995), the set of forward looking centers of an utterance contains the entities referred to in that utterance. This set is partially ordered to reflect the relative prominence of the referring expressions within the utterance.

[11] To see this, observe that the modified algorithm still requires as much iterations as properties realized in the description (n_l) and in each of the iterations has to inspect the set of distractors (n_d).

[12] Notice that we are not committed to an 11-point scale of salience *per se*. Any finite subset of the integers will do.

[13] Informally speaking, the topic of U is what sentence U is about, while the focus of U is what the sentence says about its topic (see e.g., Sgall *et al.* 1986). Notice that this notion of focus is very different from that used in e.g., Grosz & Sidner (1986) and Grosz *et al.* (1995).

Grammatical roles play a major factor in this, so that *subject* $>$ *object* $>$ *other*.

DEFINITION 4 (Salience weight assignment based on Centering Theory)
Let sw_i be some total function mapping the objects in a domain D to $\{0, \ldots, 10\}$. Let U_k be a sentence uttered in the context of sw_i, in which reference is made to $\{d_1, \ldots, d_n\} \subseteq D$. Let $C_f(U_k)$ (the forward looking centers of U_k) be a partial order defined over $\{d_1, \ldots, d_n\} \subseteq D$. Then the new salience function sw_{i+1} is defined as:

$$sw_{i+1}(d) = \begin{cases} 0 & \text{if } d \notin C_f(U_k) \\ n & \text{otherwise, where } n = \mathsf{level}(d, C_f(U_k)) \end{cases}$$

Here, $\mathsf{level}(d, C_f(U_k))$ refers to the level of the occurrence of d in the ordering $C_f(U_k)$, defined in such a way that the highest element(s) on the ordering are mapped to 10, the element(s) immediately below are mapped to 9, etc.

4.3 Examples, Predictions and Comparison

Here we discuss two illustrative examples in some detail, emphasizing the repercussions of choosing either way of assigning salience weights for the context sensitive, incremental generation of referring expressions. Suppose the domain of discourse is D_1. We use sw^h to indicate the salience weights as assigned by the Hajičová-based approach (definition 3), while sw^c gives the salience weights as based on the Centering approach (definition 4). We only keep track of the individuals in the domain and only list values which are not zero. Indices on words refer to entities in the domain. In these example sentences, the Praguian topic always coincides with the syntactic subject. Consider the following example.

(2) a. The$_2$ white chihuahua was angry.
$sw^h(d_2) = 9$
$sw^c(d_2) = 10$

b. It$_2$ viciously attacked the$_1$ black$^{[+c]}$ chihuahua.
$sw^h(d_2) = 9, sw^h(d_1) = 10$
$sw^c(d_2) = 10, sw^c(d_1) = 9$

c. { The$_1$ dog (H)/The$_1$ black$^{[+c]}$ dog (C) } barked loudly.
$sw^h(d_2) = 8, sw^h(d_1) = 10$
$sw^c(d_1) = 10$

First we take the Praguian point of view, using sw^h as defined above. In (2.a) the modified algorithm produces the description *the white chihuahua* to refer to d_2. Thereby, d_2 becomes the most salient object, with the highest-but-one degree of salience, as it is introduced in the topic of the sentence. In (2.b) d_2 is referred to by a pronoun occurring in the topic of the sentence, so its salience weight is unchanged.[14] A new object, d_1, is introduced to the discourse using the description *the black chihuahua*. Since this is done in the focus of the sentence, d_1 rises to maximal salience. When the modified algorithm generates a description for d_1 in (2.c), the description *the dog* is constructed, as d_1 is currently the most salient dog. This result,

[14] Below we address the problem of pronoun generation.

though marginally acceptable, is not what we want. A simple solution to this problem is to ignore small differences in activation degree between competitors (see also Kruijff-Korbayová & Hajičová 1997:41). This would amount to re-defining $\mathsf{MostSalient}(r, L, s)$ in such a way that this condition is met if, and only if, every object in $\mathsf{Val}(L)$ different from r has a salience weight in s which is at least, say, two points lower than the salience weight in s of r itself.[15] Let us now discuss example (2) from the Centering perspective. After the first sentence has been uttered, we find that C_f (2.a) is the singleton set containing d_2, and as a result this object is now the most salient object. In (2.b) we refer to both d_2 and d_1, and C_f (2.b) therefore contains both d_1 and d_2. Since d_2 is referred to in subject position, it is ranked higher than d_1. Consequently, to refer to d_1 in (2.c) the modified Incremental Algorithm correctly produces *the black dog*, where the adjective is marked as contrastive.

Now consider the following example. Suppose Joe went to a dog show (thus, the domain of discourse contains hundreds of dogs of all kinds, sizes, colours) and bought two dogs.

(3) a. Joe bought the$_{45}$ large black long-haired sausage dog and the$_{53}$ small grey pygmy poodle with the perm wave.
$sw^h(joe) = 9, sw^h(d_{45}) = 10, sw^h(d_{53}) = 10$
$sw^c(joe) = 10, sw^c(d_{45}) = 9, sw^c(d_{53}) = 9$

b. The$_{45}$ sausage dog was a bargain.
$sw^h(joe) = 8, sw^h(d_{45}) = 9, sw^h(d_{53}) = 8$
$sw^c(d_{45}) = 10$

c. { The$_{53}$ poodle (H) / The$_{53}$ small grey pygmy poodle with the perm wave (C) } was very expensive though.
$sw^h(joe) = 7, sw^h(d_{45}) = 8, sw^h(d_{53}) = 9$
$sw^c(d_{53}) = 10$

Comparing the two different salience weight assignments in this example presents the following picture. After generation of the first sentence, both approaches assign high salience weights to *joe* and the dogs d_{45} and d_{53}. The second sentence only contains a reference to d_{45} (*the sausage dog*). Using the Centering approach, this reduces the salience weight of *joe* and d_{53} to zero as they are not mentioned in (3.b), whereas using the Hajičová approach entails that their salience weights are reduced much less. This difference in salience weight reduction for d_{53} greatly influences its description in (3.c). Seen from a Praguian perspective, d_{53} is still the most salient poodle at this stage, and the generated description is *the poodle*. However, from the Centering perspective, d_{53} is not salient at all and the modified Incremental Algorithm again produces the description *the small grey pygmy poodle with the perm wave*, just as it did for the first-mention of this dog in (3.a). In our opinion, this shows that the Centering assumption that only objects mentioned in the previous sentence can have a non-zero salience weight, is too strong from the perspective of definite descriptions. Again, there is an obvious way to remedy this shortcoming: we have to take the structure of the discourse into account (see e.g., Walker 1997 for a proposal to this effect).

[15] In fact, Reiter (p.c.) suggested that the salience threshold might be dependent on text-genre. For instance, the salience threshold for legal texts seems to be very high (see e.g., Maes 1991, who shows that legal texts hardly contain anaphoric descriptions).

4.4 *Discussion*

As we have seen, there are at least two principled ways to determine salience weights that can be used in the modified algorithm. We have discussed two examples illustrating the differences. In the first case, the Centering approach yields better results, while in the second case, it is the Hajičová way of determining salience weights which pays off. A simple solution, which gives us the best of both worlds, would be to combine the ordering of definition 4 with the gradual decrease in salience offered by definition 3. In general, the determination of salience adds little computational overhead. To compute a new salience function only the values of the objects mentioned in the current clause and the objects with a non-zero salience weight have to be updated. Even for huge domains, the latter set is highly restricted, containing only the objects mentioned in the last few sentences.

5 FURTHER EXTENSIONS

5.1 *Pronominalization*

Reiter & Dale (1997:81) point out that a simple but surprisingly effective strategy for pronominalization is to "use a pronoun to refer to an entity if the entity was mentioned in the previous clause, and there is no other entity in the previous clause that the pronoun could possibly refer to". This is a fairly conservative strategy, which has the advantage that it will not often produce incorrect pronominalizations. On the other hand, it has been claimed that a pronoun should be used whenever this is possible.[16] The presence of salience weights in the modified algorithm makes it possible to employ a somewhat less conservative strategy, taking the pronominalization decision *within* the algorithm. This basic idea is as follows: if an object r is the most salient object with respect to the empty list of properties (thus: r is the single most salient object in the domain) and there is an antecedent for this object in the direct linguistic context, then r can be referred to using a pronoun.

$\boxed{\textsf{MakeReferringExpression}\ (r, P, s)}$

$L \leftarrow \{\}$, *tree* $\leftarrow$ nil, *contrast* $\leftarrow$ **false**

if MostSalient $(r, \{\}, s) =$ **true** $\wedge\ \exists c :$ Antecedent(c, r)

then *tree* $\leftarrow$ Pronominalize (*tree*, r) $\wedge$

  return *tree*

else for each member A_i of list P **do**

$\vdots$

Figure 3: Pronominalization within MakeReferringExpression. Remainder of the algorithm is as given in figure 1.

We certainly do not offer this as the final answer to the problem of generating pronouns. One obvious limitation is that it does not take the role of semantics and common sense into account (see e.g., Kameyama 1996, Passonneau 1996). The current approach is only concerned with the default approach to pronoun generation which is purely syntactically motivated, and does not address how semantic

[16] This is a specific instance of the DOAP principle from Williams 1997: "Don't Overlook Anaphoric Possibilities."

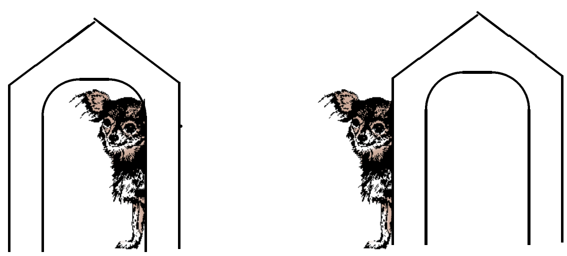
or pragmatic information can override the default. The success of this strategy depends fully on the adequacy of the underlying model of salience weight determination, but we contend that the approaches discussed in the previous section could serve as a starting point. Suppose, for example, that we determine salience weights as in definition 4 (based on Centering Theory). If sentence (4.a) has been uttered at the onset of a discourse, object d_2 is more salient than object d_3. If the modified algorithm subsequently generates a referring expression for d_2, it will produce *it*, while a subsequent reference to d_3 will yield a full, anaphoric description, which is intuitively right.

(4) a. The$_2$ white chihuahua was chasing the$_3$ cat.

b. $\{$It$_2$/The$_3$ cat$\}$ ran fast.

5.2 Relational Descriptions

Dale & Haddock (1991) offer an algorithm for the generation of relational descriptions, which is couched in terms of the "Greedy Heuristics algorithm" (Dale 1992). The basic claim we want to make here is that it is possible to generate relational descriptions in a computationally efficient way using a slightly adapted version of the (modified) Incremental Algorithm. Sticking to our continuing dogs and cats theme, let us consider the following situation:



More precisely, we focus on the following sample domain (D_2) where we introduce the attribute 'spatial' which has a relation (in (x, y)) as value:

d_1 ⟨ type, chihuahua (d_1) ⟩, ⟨ size, small (d_1) ⟩, ⟨ colour, black (d_1) ⟩
d_2 ⟨ type, chihuahua (d_2) ⟩, ⟨ size, small (d_2) ⟩, ⟨ colour, black (d_2) ⟩, ⟨ spatial, in (d_2, d_4) ⟩
d_3 ⟨ type, doghouse (d_3) ⟩ , ⟨ size, large (d_3) ⟩, ⟨ colour, white (d_3) ⟩
d_4 ⟨ type, doghouse (d_4) ⟩ , ⟨ size, large (d_4) ⟩, ⟨ colour, white (d_4) ⟩, ⟨ spatial, in (d_2, d_4) ⟩

When an object a is the intended referent and stands in a relation R to some object b, then —following Levelt (1989)— b is referred to as the *relatum* of a. Since we allow relations as properties, the definition of value sets has to be extended. A minor complication is that we now have to keep track of the object whose value set we want to determine (either the first or the second argument of a relation); therefore the intended referent is included as a subscript. This yields the following definition.

DEFINITION 5 (Value sets (modified))
$\mathsf{Val}_{a,D}(\langle A, P(a)\rangle) = \{d \in D \mid d$ has the property expressed by $P\}$
$\mathsf{Val}_{a,D}(\langle A, R(a,b)\rangle) = \{d \in D \mid \exists d' \in D : \langle d, d'\rangle$ stand in relation $R\}$

$\mathsf{Val}_{a,D}(\langle A, R(b,a)\rangle) = \{d \in D \mid \exists d' \in D : \langle d', d\rangle$ stand in relation $R\}$
$\mathsf{Val}_{a,D}(\{p_1,\ldots,p_n\}) = \mathsf{Val}_{a,D}(p_1) \cap \ldots \cap \mathsf{Val}_{a,D}(p_n)$.

For example: $\mathsf{Val}_{d_4,D_2}(\{\langle$ spatial, in (d_2,d_4) $\rangle\}) = \{d \in D_2 \mid \exists d' \in D_2 : d'$ is contained in $d\}$ (the set of objects which contain something) $= \{d_4\}$.[17] The list of preferred attributes needs some rethinking as well. It seems a valid assumption that people prefer to describe an object in terms of unary properties, and only shift to relations when unary descriptions do not suffice (i.e., one-place properties stand to relations as absolute properties stand to relative properties). This would follow from the omnipresent *principle of least effort* (e.g., Zipf 1949, Clark & Wilkes-Gibbs 1986): it takes less effort to consider and describe only one object. Similarly, we follow Horacek 1997:208 in assuming that human speakers and hearers have a preference for relatums which are close to the designated referent. Reasonable as such assumptions seem, we are not aware of any psycholinguistic research into these issues. For our example domain D_2, we simply assume that $P = \langle$ type, colour, size, spatial $\rangle$.

```
[MakeReferringExpression (r, P, L, tree, s)]
contrast ← false
for each member A_i of list P do
   V ← FindBestValue (r, A_i, BasicLevelValue(r, A_i), s)
   if (|Val(L ∪ {⟨A_i, V⟩})| < |Val(L)| ∨ A_i = type) ∧
     contrast ← Contrastive(r, A_i, V) ∧
     (tree ← UpdateTree(tree, V, contrast)) ≠ nil
   then L ← L ∪ {⟨A_i, V⟩}
   endif
   if V = R(r, r') ∨ V = R(r', r) for some relation R
   then tree ← MakeReferringExpression (r', P, {⟨A_i, V⟩}, tree, s)
   endif
   if MostSalient (r, L, s) = true
   then if ⟨type, X⟩ ∈ L for some X
          then tree ← AddDefDet (tree) ∧
                return tree
          else V ← BasicLevelValue(r, type) ∧
                (tree ← UpdateTree (tree, V, false)) ≠ nil ∧
                tree ← AddDefDet (tree) ∧
                return tree
          endif
  endif
return failure
```

Figure 4: Extension of the modified Incremental Algorithm which incorporates relational descriptions. Other functions as in figure 1.

[17] In certain situations the selected properties of the relatum (L_b) are also of interest. So a more general definition of the value set of a relation would be $\mathsf{Val}_{a,D}(\langle A, R(a,b)\rangle) = \{d \in D \mid \exists d' \in \mathsf{Val}_{b,D}(L_b) : \langle d, d'\rangle$ stand in relation $R\}$. For expository reasons, we stick to the simpler definition in the main text.

Figure 4 shows a version of the modified Incremental Algorithm which is suited for the generation of relational descriptions. The chief novelty is that the algorithm now allows for recursion: as soon as a relation R is included, the MakeReferring-Expression function is called again with as parameters the relatum, the list of preferred attributes, the relation (which already provides some information about the relatum!), the syntactic tree constructed so far, and the context. To enable this recursive call of MakeReferringExpression, the variables *tree* and L have been promoted to parameters.

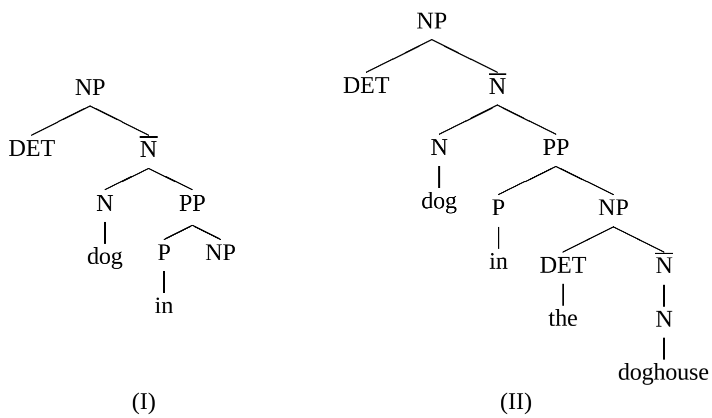


Figure 5: Two crucial stages in the generation of *the dog in the doghouse*.

Let us discuss an example: suppose we want to generate a description for object d_2 of example domain D_2 in the initial situation s_0 (all objects are equally non-salient). We call the function MakeReferringExpression $(d_2, P, \{\}, \text{nil}, s_0)$, where $P = \langle$ type, colour, size, spatial $\rangle$, $\{\}$ is the empty set of selected properties and nil is the empty tree. As before, we iterate through P. The first property we encounter is $\langle$ type, chihuahua (d_2) $\rangle$. The best value is 'dog', and including this property rules out the two doghouses. This property is realized as the N^0 in the NP tree under construction. The MostSalient condition is not true: d_1 is a dog as well. The second and third attributes ('colour' and 'size') fail to distinguish d_2 from d_1. The fourth item, $\langle$ spatial, in $(d_2, d_4)\rangle$, *does* rule out d_1, which is not inside something. This item is included in the tree under construction as the head of a PP. The resulting tree (I) is given in figure 5. Now we enter the recursion: the function MakeReferringExpression is called with as parameters d_4 (the relatum), the list of preferred attributes, the one property of d_4 already included ($\langle$ spatial, in $(d_2, d_4)\rangle$), the current tree (I), and the state s_0. The first element on the list of preferred attributes is 'type'. The type of d_4 is 'doghouse'. This property is not included since it does not rule out any of the objects not yet ruled out by the spatial property. Now, MostSalient $(d_4, \langle$ spatial, in $(d_2, d_4)\rangle, s_0)$ is true: d_4 is the most salient non-empty object in this situation. Before leaving this iteration, the basic 'type' value of d_4 (doghouse) is included after all[18] and the function AddDefDet inserts a definite determiner into the embedded NP. The resulting tree (II in figure 5) is returned and the initial call of

[18]It would be more efficient to simply include the 'type' immediately. However, it has been our intention to stick as close to Dale & Reiter (1995) as possible.

MakeReferringExpression continues. At this point, the selected properties of d_2 are $L_{d_2} = \{\langle$ spatial, in $(d_2, d_4)\rangle, \langle$ type, dog $(d_2)\rangle\}$. MostSalient (d_2, L_{d_2}, s_0) is now true as well. To wrap things up AddDefDet inserts a definite article into the main NP and the final tree is returned. Thus, the algorithm outputs *the dog in the doghouse*. The interesting thing about this description is that it is distinguishing, while neither *the dog* nor *the doghouse* in isolation are.

One common criticism of Dale & Haddock's algorithm is that it always produces 'embedded' instead of 'flat' descriptions (see e.g., Horacek 1997:212, Stone & Webber 1998:185). Consider the following situation: a room contains three tables t_1, t_2 and t_3, where t_1 supports a cup (c_1) and a glass (g_1, standing to the left of c_1), t_2 only a glass (g_2) and t_3 only a cup (c_2). A natural way to refer to t_1 would be something like *the table with the glass and the cup*. However, if t_1 would be fed to Dale & Haddock's algorithm, the algorithm would generate something like example (5), using a relatum (the cup) to fully specify the relatum (the glass) of the object to be described (the table).

(5) the table with the glass left of the cup

It is interesting to see what is wrong with Dale & Haddock's result. It is not that one cannot use a relatum to specify another relatum (think of: *the dog in the doghouse next to the garage*). Rather, in the scenario described above it is just not relevant that the glass is *left* of the cup. This information would only be relevant if the domain contained a fourth table (t_4) with a glass placed to the *right* of a cup. In that scenario it would be perfectly natural to describe t_1 as (5) (and notice that *left* would receive a pitch accent in that case).

The algorithm outlined in figure 4 can capture this insight by defining subsumption hierarchies on relations. Figure 6 contains (part of) such a subsumption hierarchy for spatial relations. Here σ stands for the underspecified spatial relation. We

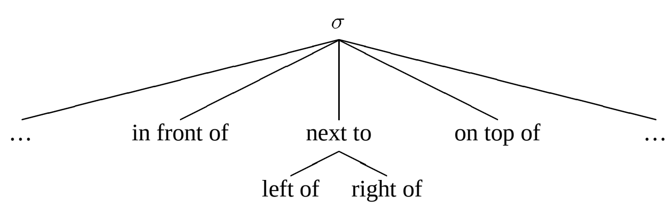


Figure 6: Part of the subsumption hierarchy on spatial relations

assume that σ is the 'basic level value' for spatial relations and that it is linguistically realised as a plain conjunction.[19] The value stored in the domain is the most specific one (just as for the animal types). Suppose we are generating a description for t_1, and the algorithm has decided to include the type of t_1 in this description (it is a table) as well as the fact that t_1 supports a glass g_1. At this point, t_1 is already distinguished from t_3 and the generated tree is given in figure 7.

[19] It is interesting to note that most of the work on categorization in cognitive science has been concerned with physical objects. A notable exception is Case study 2 in Lakoff (1987). Here Lakoff studies the interrelations between various senses of the preposition *over*, which leads —on a lower level— to a similar structure as 6. However, to the best of our knowledge the notion of basic level

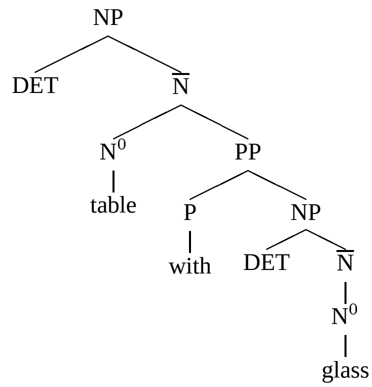


Figure 7: Intermediate tree for table t_1.

The algorithm is now in the recursion and tries to further specify the glass g_1. For this purpose it has only one attribute ('spatial') left, namely that this glass is located to the left of cup c_1.[20] The function FindBestValue is called with the basic level value of the relevant spatial relation, which we have taken to be σ. This basic level value turns out to be the best since $\sigma(g_1, c_1)$ rules out the remaining table t_2. By assumption, the underspecified spatial relation σ is linguistically realised as a conjunction. The resulting description will thus be *the table with the glass and the cup*. In contrast, if we would add table t_4 to the domain (which supports a glass to the right of a cup), then $\sigma(g_1, c_1)$ would not be the best value for the spatial relation as it would fail to distinguish the intended referent from the newly added table t_4. This would only be accomplished by *left of*. (We invite the reader to check what would happen if the glass on t_4 were placed on top of the cup.)

Let us take stock. We have shown that some simple modifications to the (modified) Incremental Algorithm suffice for the generation of descriptions involving relations. To begin with, this exercise shows how insights of Dale & Haddock (1991) can be incorporated in Dale & Reiter's Incremental Algorithm, which is more efficient than the Greedy Heuristics strategy used by Dale & Haddock. The resulting algorithm still has clear computational properties: the theoretical complexity remains polynomial, and the typical run time can be characterized as $n_r(n_d n_l)$ (where n_r is the number of objects described and $n_d n_l$ models the run time of the original Incremental Algorithm). Moreover, this algorithm is fully explicit about which properties should be tried in which order. The use of subsumption hierarchies on relations seems to offer an attractive and plausible means of obtaining some of the required flexibility. Finally, the possibility of generating relational descriptions in

---

values for certain classes of relations has not been studied, and we certainly do not intend to claim psychological reality for our assumptions here.

[20] Theoretically, *every* object in the domain stands in spatial relation to g_1. However, as said, we assume that human speakers and hearers have a preference for a relatum close to the designated referent. The problem of deciding which objects are 'close' may be likened to the problem of deciding which objects are 'large'. In the ideal case, both size and spatial distances are given in absolute measures (cm/in/pt) and a reasoning component determines which objects are indeed large or close by. It may well be that this kind of reasoning in its general form is rather complex and computationally expensive (see e.g., Lemon & Pratt 1997). For now, we simply assume that the input data are given.

a context-sensitive manner paves the way for the generation of another kind of nominals: bridging descriptions.

5.3 Bridging Descriptions

The two extensions of the modified Incremental Algorithm described above and displayed in figures 3 and 4 respectively can be combined in a straightforward way and this combination allows for the generation of bridging descriptions. From the current perspective, a bridging description is just a relational description with a highly salient relatum. To illustrate this, consider a domain of discourse which contains three objects d_1, d_2 and d_3: d_1 is a man, while d_2 and d_3 are chihuahua's of the same size and colour, the former being in the possession of d_1, the latter being a stray dog. Suppose that the man has just been mentioned ('A man is walking in the park') and thus is maximally salient. Now we attempt to generate a description referring to d_2. To begin with, the type ('dog') is included. The attributes 'size' and 'colour' are not included in the description since they fail to rule out the other chihuahua. Finally, the possessive relation is encountered: the fact that d_2 is in the possession of d_1 is included as this does rule out the stray chihuahua. At this point, the algorithm enters the recursion. Since d_1 is the single most salient object in the domain, we can pronominalize the reference to d_1 and a suitable pronoun is inserted in the current tree. Normally, this would result in (a tree for) *dog of him*, but following common practice (see e.g., Geurts 1995, Krahmer & van Deemter 1998) the function UpdateTree rewrites such descriptions using a possessive pronoun as determiner, with *his dog* as the net-result.[21]

6 CONCLUDING REMARKS

In this paper we have discussed a generalization of Dale & Reiter's Incremental Algorithm which extends the original algorithm in a number of respects. To begin with, we have made the notion of context sets more precise by adding salience weights. This makes it possible to generate descriptions in a fully context sensitive manner, without jeopardizing the attractive properties of the original algorithm. Additionally, the algorithm now immediately attempts to incorporate selected properties in the NP tree under construction and marks contrastive properties as such.

The modified Incremental Algorithm has been fully implemented and integrated in a data-to-speech system called LGM (see e.g., Klabbers *et al.* 1998 and the references cited therein). Klabbers *et al.* also describe how factors like newness and givenness, combined with contrastiveness (marked by the [+c] feature), determine the placement of pitch accents and intonational boundaries. We are currently evaluating the predictions of the modified Incremental Algorithm in an experiment.

Three related extensions of the modified algorithm have been described. First, a simple pronominalization decision within the algorithm was discussed. Second, we have shown that some modifications of the (modified) Incremental Algorithm allow for the generation of relational descriptions, again without losing the efficiency or speed of the original Incremental Algorithm. Finally, the combination of these two

[21] As a rule of thumb, we assume that this happens only if the relatum is animate. Thus, if a particular car c is highly salient and we want to refer to the motor of c, the resulting NP will not be *its motor* but *the motor*. It should be noted that this distinction is highly language dependent. In French, for example, it is common to refer to someone's hand as *le main* instead of *son main*.

extensions enables the generation of bridging descriptions.

Acknowledgments The modified algorithm was implemented and integrated in the LGM by John-Pierre Verhagen. Earlier versions of this work have been presented at ICSLP'98 (Sydney) and the Semantics Colloquium (Nijmegen, May '99). Thanks are due to the audiences at these occasions as well as to Janet Hitzeman, Jan Landsbergen, Jon Oberlander, Massimo Poesio, Ehud Reiter, Marc Swerts and Jacques Terken. We have greatly benefitted from Robert Dale's constructive and detailed comments on a previous version of this paper. Needless to say, we alone are responsible for the errors and opinions which this paper contains.

REFERENCES

[1] Alshawi, H. (1987), *Memory and Context for Language Interpretation.* Cambridge: Cambridge University Press.

[2] Beun, R.-J. and A. Cremers, (1998) Object Reference in a Shared Domain of Conversation, *Pragmatics & Cognition* **6**(1/2): 121-152.

[3] Clark, H. & D. Wilkes-Gibbs (1986), Referring as a Collaborative Process, *Cognition* 22:1-39.

[4] Cruse, D. (1977), The Pragmatics of Lexical Specificity, *Journal of Linguistics* **13**:153-164.

[5] Dale, R. (1992), *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*, MIT Press, Cambridge, Massachusetts.

[6] Dale, R. & N. Haddock (1991), Generating Referring Expressions Involving Relations, in: *Proceedings of EACL*, Berlin, 161-166.

[7] Dale, R. & E. Reiter (1995), Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions, *Cognitive Science* **18**: 233-263.

[8] Geurts, B. (1995), *Presupposing*, Ph.D. dissertation, University of Osnabrück.

[9] Grosz, B., A. Joshi & S. Weinstein (1995), Centering: A Framework for Modeling the Local Coherence of Discourse, *Computational Linguistics* **21**(2):203-225.

[10] Grosz, B. & Sidner, C. (1986), Attention, Intentions and the Structure of Discourse, *Computational Linguistics* **12**:175-204.

[11] Horacek, H. (1997), An Algorithm for Generating Referential Descriptions with Flexible Interfaces, *Proceedings of the 35th ACL/EACL*, Madrid, 206-213.

[12] Hajičová, E. (1993), Issues of Sentence Structure and Discourse Patterns. *Theoretical and Computational Linguistics*, Vol 2, Charles University, Prague.

[13] Kameyama, M. (1996), Indefeasible Semantics and Defeasible Pragmatics, in: *Quantifiers, Deduction and Context*, M. Kanazawa, C. Piñon, H. de Swart (eds.), CSLI Publications, Stanford, 111-138.

[14] Klabbers, E., Krahmer, E., & Theune, M. (1998), A Generic Algorithm for Generating Spoken Monologues, *Proceedings ICSLP'98*, Sydney, Australia, 2759-2862.

[15] Krahmer, E. (1998), *Presupposition and Anaphora*, CSLI Publications, Stanford.

[16] Krahmer, E. & K. van Deemter (1998), On the Interpretation of Anaphoric Noun Phrases, *Journal of Semantics* **15**(3/4), 355-392.

[17] Kruijff-Korbayová, I. & Hajičová, E. (1997), Topics and Centers: A Comparison of the Salience-based Approach and the Centering Theory, *Prague Bulletin of Mathematical Linguistics* **67**:25-50.

[18] Knuth, D. (1986), *Computers and Typesetting, Volume B*, Addison-Wesley Publishing Co., Reading, MA.

[19] Lakoff, G. (1987), *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, Chicago, The University of Chicago Press.

[20] Lappin, S. & H. Leass (1994), An Algorithm for Pronominal Anaphora Resolution, *Computational Linguistics* **20**(4): 535-561.

[21] Lemon, O. & I. Pratt (1997), Spatial Logic and the Complexity of Diagrammatic Reasoning, *Machine Graphics & Vision*, **6**(1):89-108.

[22] Levelt, W. (1989), *Speaking, from Intention to Articulation*, MIT Press, Cambridge, MA.

[23] Lewis, D. (1979), Scorekeeping in a Language Game, *Journal of Philosophical Logic* **8**:339-359.

[24] Maes, A. (1991), *Nominal Anaphors and the Coherence of Discourse*, Ph.D. dissertation, Tilburg University.

[25] Oberlander, J. (1998), Do the Right Thing, *Computational Linguistics*, **24**(3),501-507.

[26] Passonneau, R. (1996), Using Centering to Relax Gricean Informational Constraints on Discourse Anaphoric Noun Phrases, *Language and Speech* **39**(2-3):229-264.

[27] Pechmann, T. (1989), Incremental Speech Production and Referential Overspecification, *Linguistics* **27**:98–110.

[28] Prevost, S. (1996), An Information Structural Approach to Spoken Language Generation, *Proceedings of the 34th ACL*, Santa Cruz, 294-301.

[29] Reiter, E. & R. Dale (1997), Building Applied Natural Language Generation Systems, *Natural Language Engineering* **3**:57-87.

[30] Rosch, E. (1978), Principles of Categorization, *Cognition and Categorization*, E. Rosch and B. Lloyd (eds.), Erlbaum, Hillsdale NJ, 27-48.

[31] Sgall, P., E. Hajičová and J. Panevová (1986), *The Meaning of the Sentence in its Pragmatic Aspects.* Reidel, Dordrecht.

[32] Stone, M. & B. Webber (1998), Textual Economy Through Close Coupling of Syntax and Semantics, *Proceedings IWNLG*, Montreal, 178-187.

[33] Theune, M. (1997), Contrastive Accent in a Data-to-Speech System, *Proceedings of the 35th ACL/EACL*, Madrid, 519-521.

[34] Walker, M. (1997), Centering, Anaphora Resolution and Discourse Structure, in: *Centering in Discourse*, M. Walker *et al.* (eds.), Oxford University Press.

[35] Williams, E. (1997), Blocking and Anaphora, *Linguistic Inquiry* **28**:577-628.

[36] Zipf, G.K. (1949), *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Cambridge, MA.