Generating Bridging Definite Descriptions

Claire Gardent (gardent@loria.fr)
CNRS, Nancy, France

Kristina Striegnitz (kris@coli.uni-sb.de)
Dept. of Computational Linguistics, University of Saarbrücken, Germany

Abstract. In this paper, we focus on the role knowledge based reasoning plays in the generation of definite descriptions. Specifically, we propose an extension of Dale and Reiter's incremental algorithm which covers not only directly anaphoric descriptions but also indirect and associative anaphora. Starting from a formalism independent algorithm, we further show how this algorithm can be implemented using description logic.

1. Introduction

It has long been known that knowledge based reasoning is a crucial component of natural language processing (NLP). Yet the complexity involved in representing and using knowledge efficiently has led most NLP work to focus on more tractable aspects of language such as syntax, prosody or semantic construction.

The generation of definite descriptions (i.e., noun phrases with a definite article such as "the rabbit") is a case in point. The goal of this sub-task of natural language generation (which is the production of a text satisfying a given communicative goal) is to construct a noun phrase that allows the hearer to uniquely identify its referent in the context of utterance.

The standard algorithm for this task (on which most other proposals are based) is presented in (Dale and Reiter, 1995). But neither this algorithm nor the extensions proposed in (Horacek, 1997) and (Krahmer and Theune, 2001) take world knowledge into account when considering the context of utterance. For all these algorithms, the context is a *fixed* set of entities and relations intended to represent the current (linguistic and situational) context of utterance.

Yet many definite descriptions either refer to inferable entities (entities not explicitly mentioned or present in the context of utterance but inferable from it) or refer to contextually salient entities using inferred rather than explicitly mentioned relations (cf. Poesio and Vieira, 1998). For instance, the use of the definite article to refer to the patrons, the waitress, the busboys, etc. in (1) can only be explained by taking into account that world knowledge supports the assumption that these are somehow related to the restaurant.

(1) The young woman scans **the restaurant** with this new information. She sees all *the patrons* eating, lost in conversations. *The tired waitress*, taking orders. *The busboys* going through the motions, collecting dishes. *The manager* complaining to the cook about something.

One proposal which does integrate knowledge based reasoning in generation of definite descriptions is that presented in Stone, 1998. The philosophy underlying Stone's proposal is that knowledge based reasoning should be integrated with sentence planning to reason (i) about what the already generated text implies and (ii) about what the context entails. In this paper, we follow up on Stone's proposal and show how to integrate surface realization and inference into Dale and Reiter's algorithm to support the generation of such definite descriptions as illustrated in (1).

We start (Section 2) by presenting Dale and Reiter's base algorithm. Section 3 then summarizes the range of definite descriptions found in corpora while Section 4 focuses on cases involving knowledge based reasoning. In Section 5, we consider the two defining characteristics of definite descriptions, uniqueness and familiarity, and show how these can be defined to encompass not only directly coreferential, but also "inference-based definite descriptions". Section 6 presents the extended algorithm and an implementation of it based on description logic. Section 7 concludes and points to further research.

2. The Standard Algorithm for Generating Definite Descriptions

Most algorithms for generating definite descriptions that are described in the current literature have a common core based on Dale's greedy algorithm (Dale, 1989; Dale, 1992) and on the incremental algorithm proposed in (Dale and Reiter, 1995). We now describe this standard base algorithm.

2.1. The Task

The task is to find a description of an object (the *target entity*) that allows the hearer to uniquely identify that object in a given situation. This situation is known to both speaker and hearer and can be fully described by a set of facts, such as

Input:
R: a set of ground atomic formulas
t: target entity, $t \in \mathrm{terms}(R)$

Initialize:
1. *goals* $\leftarrow$ stack with only element t
2. $L \leftarrow \emptyset$

Check success:
3. <u>if</u> *goals* is empty <u>then</u> return L
4. *current_goal* $\leftarrow$ top(*goals*)
5. <u>if</u> *D(current_goal,L)* is a singleton <u>then</u> pop top(*goals*); goto 4

Extend description:
6. try to select an applicable $p \in R$
7. <u>if</u> there is no such p
<u>then</u> return fail
8. for each $o \in \mathrm{terms}(p) - \mathrm{terms}(L)$ push(o,*goals*)
9. $L \leftarrow L \cup \{p\}$
10. goto 4

Figure 1. Base algorithm for producing definite descriptions.

$$\{rabbit(r_1), rabbit(r_2), rabbit(r_3), hat(h_1), hat(h_2), hat(h_3),\\ white(r_1), black(r_2), white(r_3), in(r_1, h1), in(r_2, h_2)\}.$$

Let's assume that the goal is to describe the object r_1. The set $\{rabbit(r_1)\}$ would not be a uniquely identifying description in this case, as r_2 and r_3 also fit this description and are therefore *distractors* of r_1. In contrast, the description $\{rabbit(r_1), white(r_1), in(r_1, h_1), hat(h_1)\}$ would rule out all distractors and therefore achieve the given goal.

2.2. The Base Algorithm

The standard algorithm takes as input a target entity t and a set of facts. It either returns a set of facts that uniquely identifies t (the description) as output or fails in case such a description cannot be built. Further, to build a uniquely identifying description for a given target, the algorithm starts from an empty description and incrementally adds properties to it until all distractors are ruled out. Finally, if a property is chosen which is not unary but involves other objects, then the description must be extended until it uniquely identifies these other objects as well.

The pseudo-code of the algorithm is given in Figure 1. It uses a stack to keep track of the entities that have to be described. In the beginning the stack only contains the target entity (line 1) and the description is empty (line 2).

After the initialization, we enter the loop that adds properties to the description. This loop is controlled by the goal stack: as long as there are goals left on the stack, the algorithm tries to add information to the description. New goals are pushed onto the stack whenever adding a property introduces new objects into the description (line 8). Conversely, a goal is popped from the stack whenever the description built so far uniquely identifies the topmost object in the stack (line 5). This is checked by testing whether the distractor set of that object is a singleton, i.e., the set containing only that object. The distractor set of an object o is the set of all objects that fit the description given of o by L. Formally:

$$D(o, L) = \{o' | \exists \text{ substitution } s \text{ such that } s(o) = o' \text{ and } s(L) \subseteq R\}$$

So, if we can replace o by o' and replace all other objects mentioned in L in such a way that the resulting set is a subset of R, then o' is a distractor of o.

If the topmost goal cannot be removed from the stack, the algorithm tries to make the description more informative (lines 6–9) by *selecting* an *applicable* property from R and adding it to the description. Usually, all properties $p \in R$ that fulfill the following requirements are taken to be applicable:

- p mentions an object that is already part of the description, i.e.: $\text{terms}(p) \cap \text{terms}(L) \neq \emptyset$
- p is new to the description, i.e.: $p \notin L$
- adding p reduces the distractor set of at least one object mentioned in the description, i.e.: $D(o, L \cup \{p\}) \subset D(o, L)$ for an $o \in \text{terms}(L)$

Another restriction that has been used is to exclude all properties that, according to a given concept hierarchy, are more general than a concept that has already been included into the description.

In case there are several applicable properties the algorithm needs a strategy for deciding which one to select. This is where the different versions of the base algorithm differ most. Dale (1989), for instance, prefers those properties that rule out the most distractors. The "incremental" algorithm (Dale and Reiter, 1995), on the other hand, assumes a domain dependent ordering of properties according to their type (e.g shape > color > size) which determines the order in which properties are added.

2.3. Interleaving Content Determination and Surface Realization

The output of the base algorithm is a list of properties that uniquely identify the target entity. So, this base algorithm determines the semantic content a referring expression has to express in order to be successful, it does not, however, produce the surface form of such a referring expression. It is therefore possible that the base algorithm produces an output that cannot be verbalized.

The general idea for avoiding this problem is to interleave property selection with surface realization (see, e.g., Horacek, 1997). This allows us to immediately check that every selected property can be incorporated in the syntactic tree, and to make sure that the final description has no "holes" which have to be filled due to syntactic reasons. For example, "the red" is not a good noun phrase in English (or at least it requires a special context). To turn it into one we would have to include some property that can be realized as a head noun, even if it doesn't rule out any distractors.

2.4. An Example

To illustrate how the base algorithm works, we will now go through the example of Section 2.1.

The input to the algorithm is

$$R = \begin{array}{l}\{rabbit(r_1), rabbit(r_2), rabbit(r_3), hat(h_1), hat(h_2), hat(h_3), \\ \quad white(r_1), black(r_2), white(r_3), in(r_1, h1), in(r_2, h_2)\}\end{array}$$

and the target t is r_1.

The following table shows how the goal stack and the description L change with every pass through the main loop (lines 4–10 in Figure 1). The table furthermore shows the distractor set of the top most entity on the goal stack and the action of the algorithm.

Goals	L	Distractors	Action
$[r_1]$	$\emptyset$	terms(R)	extend L
$[r_1]$	$\{rabbit(r_1)\}$	$\{r_1, r_2, r_3\}$	extend L
$[r_1]$	$\{rabbit(r_1), white(r_1)$	$\{r_1, r_3\}$	extend L
$[h_1, r_1]$	$\{rabbit(r_1), white(r_1), in(r_1, h_1)\}$	$\{h_1\}$	pop goals
$[r_1]$	$\{rabbit(r_1), white(r_1), in(r_1, h_1)\}$	$\{r_1\}$	pop goals
[ ]	$\{rabbit(r_1), white(r_1), in(r_1, h_1)\}$		return L

In the beginning, only the target entity r_1 is on the goal stack, the description is empty, and all objects mentioned in R are distracting entities of r_1. Therefore, the condition in line 5 fails and the algorithm will try to eliminate distractors by extending the description. Let's assume that we are using a selection strategy which is following the incremental algorithm and picks properties in the following order: type > color > location. Adding $rabbit(r_1)$ to the description reduces the set of distractors. h_1, h_2, and h_3 all get ruled out, because substituting any of these objects for r_1 in $rabbit(r_1)$ won't give a property that is contained in R. However, there are still distractors, and even adding $white(r_1)$ doesn't eliminate all of them. Therefore, the algorithm adds $in(r_1, h_1)$ to the current description. This introduces a new entity h_1 into the description which is pushed onto the goal stack and becomes the new current goal. Now, the distractor set only contains h_1 itself as there is no other entity that contains a white rabbit. The goal is therefore popped off the stack. Similarly, r_1 can now be popped off the stack as there is no other white rabbit that is in something. Now, the goals stack is empty and the description is returned. Unfortunately, this description cannot be realized by a full noun phrase (it only contains information to yield "the rabbit in the"). Building the surface form while assembling the semantic content in the way described above would solve this problem.

3. Definite Descriptions in Real Texts

We now survey the types of definite descriptions that can be found in corpora thereby giving a list of the different cases that an algorithm for generating definite descriptions should be able to deal with.

Two properties are generally taken to characterize definite descriptions namely, *uniqueness* and *familiarity*. Roughly, uniqueness says that the referent denoted by the definite description must be the only referent satisfying the given description – this property is most prominently exposed in (Russell, 1906). Familiarity on the other hand, requires that this referent be known to the hearer – this is perhaps most strongly demonstrated in (Heim, 1982).

Indeed these two properties are the properties taken into account by the base algorithm: the set of relations it outputs must allow the hearer to uniquely identify the intended referent (uniqueness) and it must do so on the basis of shared knowledge about that referent (familiarity).

The familiarity/uniqueness explanation is a fairly high-level one however, and as shown by, e.g., Hawkins (1978) or Prince (1981), a finer grained examination of the phenomenon reveals a much more complex typology of possible uses. For a start, uniqueness is always relative to some restricted context. Here are some examples.

(2) b. If a rabbit sees a carrot, the rabbit eats the carrot.
c. There once was a doctor in London. The doctor was Welsh.

In (2a), uniqueness is relative to the quantification domain: for each rabbit and for each carrot that this rabbit sees, the rabbit eats the carrot that it sees. Similarly in (2b), uniqueness is relative to the domain of discourse: (2b) does not imply that there is a unique Welsh doctor in London but that there is a unique Welsh doctor in London *that the speaker is talking about.*

Although the base algorithm simply assumes an already restricted context, Krahmer and Theune (2001) show how it can be extended to deal with discourse domain restrictions. Interaction with quantification remains an open question and will probably remain so for a while as quantifiers have received little attention in the generation literature.

Moreover – and this is the main point of this paper – familiarity can be of different types, and only some of them are covered by the base algorithm. Following Poesio and Vieira (1998), for instance, we can identify four main familiarity classes: *coreferential* (direct or indirect), *bridging*, *larger situation*, and *unfamiliar* uses. In *coreferential uses*, the referent of the definite description is familiar in virtue of having been mentioned in the previous discourse (the referent is *discourse old* in Prince's (1981) terminology). In such cases, the hearer will know the intended referent either because the speaker uses the same description as was used in the previous discourse (*direct coreference*, cf. (3a)) or because she uses a description which on the basis of world or lexical knowledge, the hearer can infer to be true of the previously mentioned entity (*indirect coreference*, cf. (3b))

(3) a. *A woman* came in. *The woman* was wearing a beautiful hat.
b. *An actress* entered the stage. *The woman* was wearing a big hat.

In a *bridging use*, the referent of the definite description is discourse new but related by world knowledge to some previously mentioned, i.e., discourse old, entity. In (4) for instance, the referent of "the ceiling" is related by a *part-of* relation to the discourse old entity denoted by the NP "the room". The ceiling is not just any ceiling but the ceiling of the room that was mentioned in the previous sentence.

(4) A woman came into *the room. The ceiling* was very high.

Larger situation uses (e.g., (5)) are cases where the definite description denotes a discourse new but hearer old object: the described entity has not been mentioned in the previous discourse but is assumed by the speaker to be part of the hearer's general knowledge about the world (5a) and situation of utterance (5b).

(5) a. *The sun* is rising.
    b. Pass *the salt* please !

The *unfamiliar* class covers all remaining uses of definite descriptions; i.e., uses where the referent of the description is neither discourse/hearer old nor related by lexical knowledge to some discourse old entity. It encompasses definite descriptions with sentential complements (6a) and with modifiers relating the referent of the definite description to some either discourse or hearer old object (6b-c).

(6) a. Bill is amazed by *the fact that his father is black.*
    b. *The man John met yesterday* is interesting.
    c. *The Iran/Iraq war* is over.

In sum, a definite description can be familiar either because it refers to some known (i.e., either discourse or hearer old) entity (coreferential and situational use); or because it is related, either explicitly by the description (unfamiliar use) or implicitly by lexical knowledge (bridging), to some known entity.

The base algorithm, because it only resorts to information that is explicit in the context of utterance (either through the previous discourse or through the situation), can only account for directly coreferring or larger situation uses. Indirect coreferences and bridging uses cannot be dealt with as they require an interaction between generation and inference. In the next section, we look at these *inference base definite descriptions* in more detail to see what is needed to extend the base algorithm so that it can deal with them.

4. Definite Descriptions and Inference

Inference based definite descriptions, i.e., bridging and indirect coreferential uses, represent a non negligible proportion of uses in real text. An empirical study of the Wall Street Journal by Poesio and Vieira (1998) shows that out of 1412 definite descriptions being studied, 24% were "inference based" uses, with 9% bridging cases and 15% indirect coreference.

In both cases, processing requires reasoning based on world knowledge and the discourse context. With bridging uses, the hearer must be able to infer the implicit relation holding between the referent of the definite description and another discourse or hearer old entity. And in cases of indirect coreferential uses, the hearer must be able to infer that the properties used in the speaker's definite description, although not part of the common ground between speaker and hearer, hold of some discourse or hearer old entity.

We now consider these two cases in more detail using (here and in what follows) the following terminology. We call the referent of the definite description, the *target* and the (discourse or hearer old) entity with which it is either coreferential or related, the *anchor*.

4.1. Bridging

In a bridging use, a definite description relates the target entity to its anchor via some inferable relation, which we will call *bridging relation*. The term *bridging* was first introduced by Clark (1977), who identified several types of different bridging relations, such as the *part-of* relation, semantic roles of verbs, reasons, consequences. In this paper, we will concentrate on the *part-of* relation. Clark distinguishes three subcases of bridges involving the *part-of* relation, which are illustrated by the following examples.

(7) a. John entered *the room*. *The ceiling* was very high.
b. John entered *the room*. *The windows* looked out to the bay.
c. John entered *the room*. *The chandelier* was sparkling brightly.

In (7a), "the ceiling" is a part of the room mentioned in the previous sentence and it is a *necessary* part: to be a room, a room must have a ceiling. In contrast, "the windows" and "the chandelier" in (7b-c) are what Clark called *probable* and *inducible* parts. Rooms don't necessarily have windows, nor do they necessarily have a chandelier. But while most rooms have windows, rooms with chandeliers are actually rare. Nevertheless, it is plausible to link the chandelier in (7c) to the room as rooms typically have lamps and a chandelier is a type of lamp.

4.2. Indirect coreferential uses

Indirect coreferential uses are cases where the definite description refers to a discourse old entity using a hearer new description. So, in this case target and anchor are the same entity. Although the description used is not part of the common ground, the hearer can nevertheless identify the intended object because the description used by the speaker is compatible with the common ground and with world knowledge and allows the hearer to establish the link to the anchor. Here are some examples.

(8) a. *An actress* entered the stage. *The woman* was wearing a big hat.
b. I met *a man* yesterday. *The bastard* stole my money.
c. John bought *a new car. The Volvo* delights him.

In the first case, the property used in the definite description is entailed by world knowledge, as *woman* is a hypernym of *actress*. But as the other two examples show, this is not necessarily the case. Thus a man can, but need not, be a bastard – x *is a bastard* is a proposition that is compatible but not entailed by the proposition x *is a man*. Similarly a car can but need not be a Volvo – in that case, however, the two properties are not merely compatible, but stand in a hyponymic relation (Volvos are cars).

So, for both kinds of inference based definite descriptions, bridging uses and indirect coreferential uses, there are cases where the existence of an entity fitting the description is not logically entailed (Examples (7b,c) and (8b,c). The approach presented here focuses on a subset of these cases namely: bridging definite descriptions involving necessary, probable, and certain inducible parts; and indirect coreferential uses where the description of the target is a hypernym of some hearer known property of the anchor.

5. Familiarity and Uniqueness of Bridging Anaphora

As we saw in section 3, familiarity and uniqueness are two defining characteristics of definite descriptions. In this section, we show how to formulate these properties so as to encompass not only direct coreferential definite descriptions (as is done in the base algorithm) but also indirect coreferential and bridging uses. We start by presenting the structured context the extended algorithm is working with. We then go on to give an intuitive explanation of how uniqueness and familiarity differ in our algorithm from the way these are defined in the base algorithm. Finally, we present our definitions.

5.1. The Discourse Context

As seen in Section 2, a context in the base algorithm is an unstructured set of facts assumed to be shared by hearer and speaker. To deal with bridging anaphora, we need a slightly more sophisticated notion of context.

First, knowledge based reasoning must be possible – hence we assume that world and lexical knowledge (in the form of axioms) are part of the context.

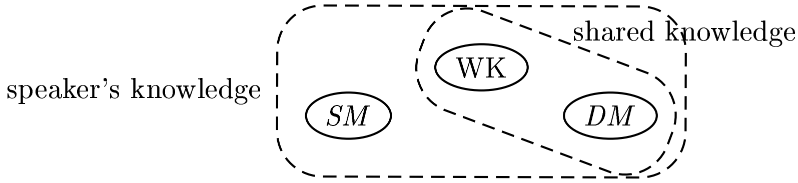

Figure 2. Modeling the discourse context.

Second, a distinction must be made between hearer's and speaker's knowledge to support the generation of examples such as (7c) where the description used by the speaker is hearer new but speaker old (in this example, the speaker knows that the room has a chandelier while the hearer doesn't).

To account for these observations, we assume that the context is structured into three sub-parts as follows:

Discourse Model (DM): a list of ground atomic formulas modeling the previous discourse
Ex: $DM = \{restaurant(r1), italian(r1), \ldots\}$

World Knowledge (WKL): shared rule-base background knowledge
Ex: $WKL = \{\forall x.restaurant(x) \rightarrow \exists y.cook(y) \wedge has(x,y), \ldots\}$
"restaurants have cooks"

Speaker Model (SM): a list of ground atomic formulas representing the additional knowledge of the speaker
Ex: $SM = \{has(r1, c1), cook(c1), excellent(c1), \ldots\}$

Discourse Model and World Knowledge are shared by speaker and hearer. As the discourse evolves, the Speaker Model (that is, that part of the speaker knowledge that is not shared with the hearer) shrinks while the Discourse Model grows. Figure 2 gives a graphical representation of how these three parts relate to each other.

5.2. Familiarity and uniqueness in the extended algorithm

In the base algorithm, the target is taken to be **hearer-old**, that is, present in the (linguistic or situational) context of utterance. Familiarity is thus taken as given and is restricted to coreferential and situational uses.

When extending the algorithm to bridging uses of definite descriptions, familiarity must be modified to include targets not only when the target itself is hearer old, but also when it is related to some hearer old entity by a bridging relation.

To encode this extended notion of familiarity, we introduce the notions of intended and potential anchors. In what follows, we give an intuitive description of these notions. These will be made precise in the next section.

Given a target t, the **intended anchors** of t (written $IA(t)$) are the entities which *the speaker* uses to "anchor" the target in the shared knowledge. That is, intended anchors are hearer-old entities which the speaker can relate either by identity or by some bridging relation to the target.

By contrast, the **potential anchors** of a target t given a description L (written $PA(t, L)$) are the entities which, given the description L, *the hearer* can relate to the target on the basis of the shared knowledge.

Given this, a description L for a target t is said to satisfy **familiarity** iff all intended anchors for t are potential anchors of t given L.

Since potential anchors are hearer old entities and since the set of intended anchors is by definition non empty, this definition ensures that a description that satisfies familiarity actually relates the target to some hearer old entity.

Further, it ensures that the set of anchors that are possible from the hearer's perspective is not smaller than the set of anchors that are possible from the speaker's perspective. Or in other words, that no speaker intended anchor is ruled out (from the hearer's perspective) by the description. The impact of this restriction is illustrated in section 5.5.4 below.

Now consider **uniqueness**. In the base algorithm, uniqueness is satisfied by a given description L iff the set of distractors for L is the singleton set containing the target, that is, if there is no other object than the target satisfying the given description.

Similarly, the extended algorithm requires that given a description L, the set of potential anchors equals the set of intended anchors. In essence, this relativizes the uniqueness condition of the base algorithm to anchors rather than targets. The singleton set containing the target now is the set of intended anchors and the description is used to eliminate elements not from the set of distractors (the set of elements in the context that fit the description) but from the set of potential anchors (i.e., from the set of elements in the context to which the target could be related either by identity or by some bridging relation).

This is not quite sufficient however to account for uniqueness in bridging uses and as we shall soon see in more detail, the extended algorithm further requires that it is plausible to assume that there is exactly one object that, given the description, can be related to the set of potential anchors.

5.3. Defining Intended and Potential Anchors

In the preceding section, we saw that for a given description, familiarity is satisfied iff all speaker's intended anchors are potential anchors of the description. To make this notion precise, we now define intended and potential anchors.

Recall that intended anchors are hearer old entities which the speaker knows to be related to the target either by identity or by some bridging relation. More specifically, intended anchors are all those discourse old entities for which the speaker can infer from her knowledge (*WKL*+ *DM*+ *SM*) that they are either identical to the target or related to the target via some bridging relation. (Here and in what follows, we write *bridge* to stand for some bridging relation.) The **intended anchors (*IA*) of a target** t are thus defined as follows:

$$IA(t) = \{o \in \text{terms}(DM) \mid WKL + DM + SM \models t = o \ \vee \ bridge(t, o)\}$$

To define the notion of potential anchors we start by defining what it means for a target t to be *familiar through* an entity a given a description L. Intuitively, t is *familiar through* an entity a given a description L if the hearer can infer on the basis of shared knowledge that a either fits the description given for t or that a is related to an entity that fits the description.

This can be formalized as follows. Let t be the target and let a be a discourse old entity. Furthermore, let $L(t, o_1, \ldots, o_n)$ be the conjunction of ground atomic formulas representing the description used by the speaker; Let $o_1, \ldots, o_n$ be the terms other than t occurring in L; and let $L(x, x_1, \ldots, x_n)$ be the formula obtained by substituting $x, x_1, \ldots, x_n$ for $t, o_1, \ldots, o_n$ in $L(t, o_1, \ldots, o_n)$. Finally, let R be the identity relation or a bridging relation (e.g., *part-of*), and let L' be either L or the generalization $L^{N'}$ of L as defined below.

Then **a target t is familiar through a given a description L of t** provided the following holds:

$$WKL + DM \models \exists x, x_1, \ldots, x_n . R(a, x) \wedge L'(x, x_1, \ldots, x_n)$$

This general condition specializes to different bridging uses of definite descriptions as follows.

Dir. Coref.	$DM \models \exists x, x_1, \ldots, x_n . a = x \wedge L(x, x_1, \ldots, x_n)$
Ind. Coref.	$WKL + DM \models \exists x, x_1, \ldots, x_n . a = x \wedge L(x, x_1, \ldots, x_n)$
Bridging (Nec. Parts)	$WKL + DM \models \exists x, x_1, \ldots, x_n . bridge(x, a) \wedge L(x, x_1, \ldots, x_n)$
Bridging (Prob. Parts)	$WKL + DM \models \exists x, x_1, \ldots, x_n . bridge(x, a) \wedge L^{N'}(x, x_1, \ldots, x_n)$

In a coreferential use, the target is familiar through an entity that is identical to the target, but while indirect coreference implies a resort to world knowledge, direct coreference only involves the discourse model.

In a bridging use involving necessary parts, the relation between the target and the entity through which it is familiar is a bridging relation and both discourse model and world knowledge are necessary for the inference to go through.

Bridging cases involving probable parts such as Example (7c) are more complex. In such cases it does not follow from the shared knowledge that a bridging relation holds between the target (the chandelier in Example (7c)) and the discourse old entity through which it is familiar (the room in Example (7c)). However, it does follow, that the discourse old entity is related via a bridging relation to an instance of a superconcept of the given description. A room does not necessarily include a chandelier but it includes furniture which is a superconcept of chandelier:

$$WKL \wedge room(a) \not\models \exists x \exists a.(part\_of(x,a) \wedge room(a) \wedge chandelier(x))$$
$$WKL \wedge room(a) \models \exists x \exists a.(part\_of(x,a) \wedge room(a) \wedge furniture(x))$$

To handle such cases, the condition is modified to involve not just the speaker's description (L) but also an adequately generalized version of the speaker's description ($L^{N'}$). More precisely, the relation N denoted by the head noun used in the speaker's description should be replaced by a relation N', such that

- N' subsumes N and
- the discourse old a is in a bridging relation *bridge* to elements of N' and there is no relation N'' such that N' subsumes N'' and a is in the same bridging relation *bridge* to elements of N''.

$L^{N'}$ is then the conjunction of atomic formulas representing the speaker's description where N' has been substituted for N.

Having defined what it means for a target to be familiar through an entity given a description, we can now define **the potential anchors of a target** t **given a description** L (written, $PA(t,L)$) to be all those entities a through which t is familiar:

$$PA(t,L) = \{a \in \text{terms}(DM) \mid t \textit{ is familiar through } a \textit{ given } L\}$$

5.4. Defining Familiarity and Uniqueness

Given the definition of intended and potential anchors introduced in the previous section, we can now state a Familiarity Condition and a Uniqueness Condition capturing the intuitions of Section 5.2.

The **Familiarity Condition** is stated as follows. A description L for a target t satisfies familiarity iff:

$$IA(t) \subseteq PA(t, L)$$

Uniqueness requires (i) that the sets of intended and potential anchors be equal and (ii) that it is plausible to assume that there is exactly one object that given the description can be related to the set of potential anchors.

The **Uniqueness Condition** therefore has the following two parts:

1. $PA(t, L) = IA(t)$
2. $\forall a \in IA(t), unique(t, a, L(t, o_1, \ldots, o_n)$:
   $WKL + DM + SM \not\models \exists y, x, x_1, \ldots, x_n (R(a, y) \wedge L(y, x_1, \ldots, x_n) \wedge$
   $R(a, x) \wedge L(x, x_1, \ldots, x_n) \wedge y \neq x)$
   with R being the identity or a bridging relation

The second condition says that it should not follow from world knowledge and the domain model that there exist two different objects which are related to the anchor and satisfy the speaker's description. Or in other words, that it must be plausible to assume that there is exactly one object that is related to the anchor and satisfies the speaker's description.

5.5. Examples

We now go trough some examples to illustrate the effect of our uniqueness and familiarity condition.

5.5.1. *An example satisfying both uniqueness and familiarity*

First, let us consider an example in which a restaurant has been mentioned and the speaker uses the description "the cook" to refer to the cook of that restaurant.

(9) a. $DM = \{restaurant(r)\}$
$WKL = \{\forall x(restaurant(x) \rightarrow \exists y(cook(y) \wedge part\_of(y, x)))\}$
$SM = \{cook(c), part\_of(c, r)\}$
$Target = c$
$Description = cook(c)$

b. John took Jim to the restaurant. The cook was wearing a white hat.

Given this context, the set of intended anchors for c is $IA(c) = \{r\}$ and the set of potential anchors for c given the description $cook(c)$ is $PA(c, cook(c)) = \{r\}$. Hence $IA(c) = PA(c, cook(c))$ and both familiarity and the first part of the uniqueness condition are satisfied. Since furthermore nothing in the given context entails that a restaurant must have more than one cook, it is consistent to assume that there is exactly one cook related to r hence the second part of the uniqueness condition is satisfied and the example satisfies both familiarity and uniqueness.

5.5.2. *An example violating familiarity*

Let us now consider a slightly modified version of the preceding example; one where the restaurant has been replaced by a zoo:

(10) a.
$$\begin{array}{lll} DM & = & \{zoo(z)\} \\ WKL & = & \{\} \\ SM & = & \{cook(c), part\_of(c, z)\} \\ Target & = & c \\ Description & = & cook(c) \end{array}$$

b. John took Jim to the zoo. *??? The cook* was wearing a white hat.

Now the set of potential anchors for the target c given the description $cook(c)$ is the empty set because world knowledge does not imply that zoos have cooks. In this case, the set of intended anchors is not a subset of the set of potential anchors. Hence familiarity is not satisfied and (10b) is odd.

5.5.3. *An example violating the first uniqueness clause*

Here is a situation with two restaurants in the discourse model (see 11a).

(11) a.
$$\begin{array}{lll} DM & = & \{restaurant(r_1), restaurant(r_2), italian(r_1), chinese(r_2)\} \\ WKL & = & \{\forall x(restaurant(x) \rightarrow \exists y(cook(y) \wedge part\_of(y, x)))\} \\ SM & = & \{cook(c), part\_of(c, r_1)\} \\ Target & = & c \\ Description & = & cook(c) \end{array}$$

b. There is an Italian restaurant around the corner and a Chinese one at the end of the street. *??? The cook* is very good.

In this case, given the description $cook(c)$ and the target c, the set of potential anchors $PA(c, cook(c)) = \{r_1, r_2\}$ whereas the set of intended anchors for c is $\{r_1\}$. Hence the first clause ($PA(t, L) = IA(t)$) of the uniqueness condition is not satisfied – which explains the oddity of (11b).

5.5.4. *An example violating the second clause of the uniqueness condition*

Finally, consider the following situation:

(12) a. $DM \;=\; \{book(b)\}$
$WKL = \;\{\forall x(book(x) \rightarrow \exists zy(page(y) \wedge page(z) \wedge part\_of(y,x) \wedge part\_of(z,x) \wedge z \neq y))\}$
$SM \;=\; \{page(p), part\_of(p,b)\}$
$Target \;=\; p$
$Description \;=\; page(p)$

b. John took the book back to the shop. ??? *The page* was missing.

In this case, the set of potential anchors for p given the description $page(p)$ is $\{b\}$, which is the same as the set of intended anchors. Hence both familiarity and the first clause of the uniqueness condition are satisfied. However, the second clause of the uniqueness condition is violated as it is inconsistent with world knowledge to assume that a book has exactly one page. Hence example (12b) is correctly ruled out by our definition of uniqueness.

6. An Algorithm for Generating Bridging Anaphora

In this section, we show how to extend the base algorithm introduced in Section 2 to bridging and indirect coreferential uses of definites. We illustrate the workings of the algorithm by means of some examples and describe an implementation of it using a description logic reasoning system for carrying out the necessary inferences on the background knowledge.

6.1. Extending the Standard Algorithm

We will now present an extension of the base algorithm that generates bridging definites in addition to coreferring definites. The main idea behind this extension is to use the relation between potential and intended anchors to control the algorithm. Thus the proposed algorithm starts with an empty description and extends it until the Uniqueness

Input:
WKL: a set of rules relating relations to each other
DM: a set of ground atomic formulas
SM: a set of ground atomic formulas
t: target entity, $t \in \text{terms}(DM \cup SM)$

Initialize:
1. *goals* $\leftarrow$ stack with only element t
2. $N \leftarrow$ initial syntactic structure with an open slot for a noun

Check success:
3. <u>if</u> *goals* is empty <u>then</u> return $\langle$uniquely identifying, $N\rangle$
4. *current_goal* $\leftarrow$ top(*goals*)
5. <u>if</u> IA(*current_goal*) $\not\subseteq$ *PA(current_goal,L(N))* <u>then</u> return $\langle$unfamiliar , $N\rangle$
6. <u>if</u> *PA(current_goal,L(N))* $=$ *IA(current_goal)*)
<u>and</u>
$\forall a \in IA(t)$ unique$(t, a, L(N))$
<u>then</u> pop top(*goals*); goto 4

Extend description:
7. <u>if</u> *current_goal* $\in$ terms(*DM*) <u>then</u> $R \leftarrow DM$ <u>else</u> $R \leftarrow DM + SM$
8. try to select an applicable atomic formula p s.t. $R + WKL \models p$
9. <u>if</u> there is no such p
<u>then</u> return$\langle$non uniquely identifying, $N\rangle$
10. for each $o \in \text{terms}(p) - \text{terms}(L(N))$ push(o,*goals*)
11. $N \leftarrow N'$ s.t. $L(N') = L(N) \cup \{p\}$
12. goto 4

Figure 3. Extension of the base algorithm to bridging cases.

and while the Familiarity Condition is satisfied. In terms of anchors this means that the algorithm proceeds until the sets of potential and intended anchors are identical and while the set of intended anchors is a subset of the set of potential anchors.

The algorithm is given in Figure 3. It takes a target entity and a representation of the discourse context (in the form described in the previous section) as input. It then simultaneously constructs the surface form and the semantic content of the referring expression (cf. Section 2.3). In Figure 3, we use N to refer to the (partial) syntactic tree. We assume that we have a way of accessing the open syntactic slots of N and that the function L gives us the set of properties that N is realizing. The output of the algorithm is a syntactic tree and a classification of this description as *uniquely identifying*, *non-uniquely identifying*, or *unfamiliar*.

The overall structure of the extended algorithm is similar to that of the base algorithm (cf. Figure 1). Again, we use a goal stack to keep track of the entities that have to be described. After some initializations (lines 1–2), we enter the main loop, which terminates successfully when the goal stack is empty (line 3). Otherwise, the algorithm examines the top entry of the goal stack (lines 4–6) and, if necessary, extends the description (7–12).

The main strategy of the *base* algorithm is to add information to the description until all distractors are ruled out. If it is not possible to rule out all distractors, no definite description can be constructed. The main strategy of the *extended* algorithm is to add information to the description until the Uniqueness Condition is satisfied. The description is extended until a) all potential anchors that are not intended anchors are ruled out, and b) it is consistent with the speaker's knowledge to assume that the target is unique wrt. the anchor (line 6).

The algorithm fails, returning the description and the classification *non uniquely identifying*, if the description, although it is not satisfying the Uniqueness Condition, cannot be specified any further by adding information (line 8–9). This is similar to the failure case of the base algorithm. But there is another way in which the extended algorithm can fail. And that is if the description doesn't satisfy the Familiarity Condition. In this case, the set of potential anchors doesn't include the intended anchors anymore (line 5).

As with the base algorithm, there is some freedom in defining what are applicable properties and the selection procedure for choosing among them. Sensible minimal requirements for a property p to be applicable are:

- $\text{terms}(p) \cap \text{terms}(L) \neq \emptyset$
- $WKL + DM + L(N) \not\models p$

These requirements are analogous to the first two of the three requirements that we used in the base algorithm. We furthermore suggest the following conditional restrictions on applicability to replace the third one.

- If there are syntactic holes, fill one of them (i.e, only those properties that can fill one of them are applicable).
- Only use a property that doesn't ensure that the Familiarity Condition holds if you have to fill a syntactic hole and there is no other way.

– Among the properties that ensure the satisfaction of the Familiarity Condition, prefer those that work toward the Uniqueness Condition by ruling out distracting entities. If there are no such properties, then choose a bridging relation that links the target to an entity $o \in \text{terms}(DM)$.

As in the base algorithm, we can now imagine various different strategies for selecting between applicable properties. We could, e.g., follow Dale and Reiter (1995) and assume a predefined order in which properties are considered, or we could always take the property that reduces the set of potential anchors most effectively (similar to Dale, 1992).

6.2. Examples

We will now illustrate how the extended algorithm deals with some of the examples that we saw in the previous sections.

6.2.1. *Example 1*

First, we will look at the case of a "normal" bridging anaphor. Let's assume that the speaker just uttered "There is a new restaurant next to the church", and that he now wants to say something about the cook of this restaurant. Let's furthermore assume that no other cooks or restaurants have been mentioned before. The relevant parts of the discourse context then look like this:

$$\begin{aligned}&WKL : \forall x(restaurant(x) \to \exists y(cook(y) \wedge part\_of(y, x)))\\&DM : restaurant(r), church(u)\\&SM : cook(c), part\_of(c, r)\end{aligned}$$

Together with the target (c) this will be the input to the algorithm.

The following table shows what the goal stack, the description, and the sets of potential and intended anchors of the current goal (i.e., the top most entity on the goal stack) look like in each pass through the main loop. We furthermore show the status of the Familiarity Condition and the Uniqueness Condition.

goals	description	*PA*	*IA*	
$[c]$	"..." $\emptyset$	terms(DM)	$\{r\}$	Fam.: ✓ Uniq.: ✗
$[c]$	"the cook" $\{cook(c)\}$	$\{r\}$	$\{r\}$	Fam.: ✓ Uniq.: ✓
$[\,]$	"the cook" $\{cook(c)\}$	⇒ uniquely identifying		

The algorithm starts with an empty description, which means that all objects mentioned in the discourse model are potential anchors. It then adds $cook(c)$, which fills the open syntactic slot while preserving the satisfaction of the Familiarity Condition, and cuts down the set of potential anchors. In fact, the set of potential anchors now contains only r which is the intended anchor of c, so that the Uniqueness condition is satisfied as well and the goal can be popped off the stack. The goal stack is empty then and the algorithm terminates successfully and returns the description.

6.2.2. *Example 2*

Now, we will look at the case of Example 11. Here bridging is possible, but the bridge has to be made explicit to pick out the correct anchor.

The input to the algorithm is as follows:

$WKL: \forall x(restaurant(x) \rightarrow \exists y(cook(y) \wedge part\_of(y,x)))$
$DM: restaurant(r_1), italian(r_1), restaurant(r_2), chinese(r_2), church(u)$
$SM: cook(c), part\_of(c, r_1)$
$target: c$

goals	description	*PA*	*IA*	
$[c]$	"..." $\emptyset$	terms(DM)	$\{r_1\}$	Fam.: ✓ Uniq.: ✗
$[c]$	"the cook" $\{cook(c)\}$	$\{r_1, r_2\}$	$\{r_1\}$	Fam.: ✓ Uniq.: ✗
$[r_1, c]$	"the cook of ..." $\{cook(c), part\_of(c, r_1)\}$	$\{r_1, r_2\}$	$\{r_1\}$	Fam.: ✓ Uniq.: ✗
$[r_1, c]$	"the cook of the restaurant" $\{cook(c), part\_of(c, r_1),$ $restaurant(r_1)\}$	$\{r_1, r_2\}$	$\{r_1\}$	Fam.: ✓ Uniq.: ✗
$[r_1, c]$	"the cook of the Italian restaurant" $\{cook(c), part\_of(c, r_1),$ $restaurant(r_1), italian(r_1)\}$	$\{r_1\}$	$\{r_1\}$	Fam.: ✓ Uniq.: ✓
$[c]$	"the cook of the Italian restaurant" $\{cook(c), part\_of(c, r_1),$ $restaurant(r_1), italian(r_1)\}$	$\{r_1\}$	$\{r_1\}$	Fam.: ✓ Uniq.: ✓
$[\,]$	"the cook of the Italian restaurant" $\{cook(c), part\_of(c, r_1),$ $restaurant(r_1), italian(r_1)\}$	⇒ uniquely identifying		

In this example, adding $cook(c)$ doesn't rule out enough distractors to satisfy the Uniqueness Condition. So, the algorithm next adds $part\_of(c, r_1)$, which doesn't rule out any distractors but links c to a discourse old entity. This adds a new goal to the goal stack. Then, *restaurant*(r_1) is added, which fills a syntactic hole. Adding *italian*(r_1) in the next round then rules out all distracting potential anchors and r_1 is removed from the goal stack. The potential anchors of c now don't contain any distracting anchors anymore either, as the shared knowledge doesn't imply that r_2 (the Chinese restaurant) has a cook which is part of an Italian restaurant.

6.2.3. *Example 3*

Finally, let's see what happens in cases where bridging is not possible, because the link between target and anchor is not implied by world knowledge.

The input to the algorithm is describing a situation where a zoo has been mentioned in the discourse and the speaker now wants to talk about its cook, which has not been mentioned before:

$$\begin{array}{l} \mathit{WKL}: \text{ (no relevant world knowledge)} \\ \mathit{DM}: \mathit{zoo}(\mathrm{z}) \\ \mathit{SM}: \mathit{cook}(\mathrm{c}),\ \mathit{part\_of}(\mathrm{c},\mathrm{z}) \\ \mathit{target}: \mathrm{c} \end{array}$$

The algorithm starts in the same way as in the other examples, but when the first property is added to the description, viz. *cook*(c), the set of potential anchors becomes empty, so that the Familiarity Condition is violated (cf. line 6 of Figure 3).

goals	description	*PA*	*IA*	
$[c]$	"..." $\emptyset$	terms(DM)	$\{z\}$	Fam.: ✓ Uniq.: ✗
$[c]$	"the cook" $\{cook(c)\}$	$\emptyset$	$\{z\}$	Fam.: ✗ Uniq.: ✗
		$\Rightarrow$ unfamiliar		

6.3. Implementation

There is a proof of concept implementation of the extended algorithm in the functional language Mozart Oz. It uses RACER, an automated reasoning system for description logics (DL), to carry out the necessary inferences on the discourse context. In this section, we will look at how these inferences are formulated as queries to the DL reasoning system. We will first give a brief introduction to description logics and describe how the different components of the discourse context can be represented as a DL knowledge base. Then, we will show how the functionality provided by typical DL reasoning systems nicely supports the reasoning tasks required by the extended algorithm.

6.3.1. *Description Logics*
Description logic (DL) is a family of logics in the tradition of knowledge representation formalisms such as KL-ONE (Woods and Schmolze, 1992). DL is a fragment of first-order logic which only allows unary and binary predicates (*concepts* and *roles*) and only very restricted quantification. A knowledge base consists of a *T-Box*, which contains axioms relating the concepts and roles, and one or more *A-Boxes*, which state that individuals belong to certain concepts, or are related by certain roles.

T-Box statements have the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept expressions. Concepts denote sets of individuals and the statement means that denotation of C_1 is a subset of the denotation of C_2, i.e., C_2 subsumes C_1. So, we can, for example, write $\mathit{cook} \sqsubseteq \mathit{human}$ to express *cooks are human.* Concepts can be combined by the boolean connectives $\sqcap$ (and), $\sqcup$ (or), and $\neg$. E.g., $\mathit{animal} \sqsubseteq \neg\mathit{human}$. Finally, we can use roles (binary predicates) and their inverses in combination with the quantifiers $\forall$ and $\exists$ to relate two concepts: $\mathit{restaurant} \sqsubseteq \exists \mathit{part\_of}^{-1}.\mathit{cook}$ expresses that every restaurant is related to a cook via an (inverse) *part_of* relation. More expressive DLs furthermore allow the number restrictions on roles such as $\mathit{horse} \sqsubseteq (= 4\ \mathit{part\_of}^{-1}).\mathit{leg}$, which means *horses have exactly four legs.*

That was the T-Box. The A-Box contains statements such as $\mathit{rabbit}(a)$ or $\mathit{love}(b, c)$ to express that the individual a is an instance of the concept rabbit, and the individual b and c are related through the role *love.*

We will represent the world knowledge as a T-Box and the discourse model and the speaker's model as A-Boxes.

Theorem provers for description logics support a range of different reasoning tasks. Among the basic reasoning tasks are, e.g., *subsumption checking* (Does one concept subsume another?), and *instance* and *relation checking* (Does a given individual belong to a certain concept?/Are two individuals related through a certain relation?). In addition, description logic systems usually provide some *retrieval functionality* which, e.g., allows to compute all atomic concepts that a given individual belongs to or all individuals that belong to a given concept. This will prove to be very useful for our purposes as it allows easy access to all properties of an object and provides an elegant way for computing the potential anchors.

There is a wide range of different description logics which add different extensions to a common core. Of course, the more expressive these extensions become, the more complex the reasoning problems are. In the last few years, new systems such as FaCT (Horrocks) and RACER (Haarslev and Möller) have shown that it is possible to achieve surprisingly good average-case performance for very expressive (but still decidable) logics. In this paper, we employ the RACER system because it allows for A-Box inferences.

6.3.2. *DL Reasoning for the Extended Algorithm*
The core reasoning task in the extended algorithm is to compute the set of potential anchors. In a DL setting, this is straightforwardly implemented by using the instance retrieval mechanism. Recall that this mechanism returns all instances of a given concept. We first create a DL concept which approximates the definition (cf. Section 5) of the potential anchors of a target t given a description L. That is, we create a concept which approximates the following formula

$$\lambda x \exists x_1, \ldots, x_n.L(x, x_1, \ldots, x_n)$$

Using instance retrieval, we can then gather all objects belonging to this concept or which are related to an instance of this concept via a bridging relation.

To construct the approximating DL concept from the set L of ground atomic formulas representing the semantic content of the definite description, we use the following strategy. Assuming that we want to compute the potential anchors of object o, we first collect all unary properties of o in L and conjoin the relations to form a concept expression. These properties are deleted from the set L. Then, we take one by one the binary properties relating o to some other object o' via a relation R, we (recursively) build a concept expression $C_{o'}$ for o' and conjoin $\exists R.C_{o'}$ with the previously constructed part. For example, for $L = \{cook(c), part\_of(c,r), restaurant(r), italian(r)\}$ and target c we will build the following DL concept:

$$cook \sqcap \exists part\_of.(restaurant \sqcap italian)$$

The resulting concept is an approximation of the corresponding first order formula. Due to the restricted expressive power of DL, the concept cannot capture reflexivity: $\lambda x(R(x,x))$ would be rendered as $\exists T.\top$. Similarly, if the same two objects are related in two different ways, this information is lost in the DL concept: $\lambda x(\exists y(R_1(x,y) \wedge R_2(x,y))$ becomes $(\exists R_1.\top) \wedge (\exists R_2.\top)$. As we only generate descriptions expressing a set of positive facts about an entity, however, these two are the only cases in which the DL concept is not equivalent to the first order formula corresponding to the description.

To check the second part of the Uniqueness Condition (cf. Section 5), we employ number restrictions and test that it is consistent with the shared knowledge to assume that the anchor is related to exactly one instance of the DL concept corresponding to the description. For example, to test whether it is *consistent* with world knowledge that an entity a has exactly one page, we test whether the *negation follows* from world knowledge. If so, it is not consistent, otherwise it is. To test the entailment, we send the query "is a an instance of the concept $\neg(= 1\ part\_of^{-1}).page$" to the DL prover.

Finally, we use RACER's functionality for retrieving properties (concepts, most specific concepts, roles) of a given instance to collect all potentially applicable properties.

The inferences that are necessary in our approach to generating definite descriptions could also be carried out by automated theorem provers for first order logic. As first order logic provers do not provide the kind of knowledge base management and retrieval functionality that comes with DL systems, however, one would need additional mechanisms for selecting applicable properties and for retrieving and maintaining the set of potential anchors.

7. Conclusion

In this paper, we have shown how the basic incremental algorithm for generating definite descriptions proposed by Dale and Reiter, can be extended to handle definite descriptions whose processing involves knowledge based reasoning. Specifically, we have shown how it can be integrated with reasoning to generate bridging and to a certain extent, indirect coreferential uses of definite descriptions.

But as seen in Section 3, bridging and coreferential uses do not exhaust the usage spectrum of definite descriptions. Larger situation and unfamiliar uses are also very frequent. Provided the context is extended to encode world and situational knowledge, the proposed algorithm naturally extends to larger situation uses – these are just uses where the entity is familiar because it is hearer old.

The unfamiliar class is more problematic. Recall that it includes definite descriptions with sentential complements (*the fact that John's father is black*) and containing inferables, i.e., entities that are familiar by virtue of being related to some discourse old entity (*the man John met yesterday, the Iran/Iraq war*). The first subclass (descriptions with sentential complements) can be viewed as a kind of event anaphora (the speaker is referring to John's father's blackness) and should probably be treated as such. The second case (containing inferables) raises the question of how familiarity should be defined. Conceivably, the familiarity notion used in our algorithm should be extended to encompass such cases and the algorithm modified to insist that all generated descriptions be both familiar and unique. This would not be difficult.

There is however a danger of overgeneration as other pragmatic factors seem to interact with the surface realization of containing inferables. It would therefore be important to first have a better understanding of the distribution and form of containing inferables i.e., of when explicit bridging is acceptable.

Another open empirical question we are currently investigating is that of the lexical knowledge involved in bridging uses of definite descriptions. In this paper, we have assumed a very simple and direct "part-of" relation between target and anchor. However it is known since at least (Clark, 1977) that bridging relations can be of various natures. We have therefore started a trilingual (English-French-German) corpus study which aims at assessing the relative prominence of the various bridging relations and to identify the precise lexical semantics involved in processing definite descriptions. This will hopefully also help to refine our treatment of bridging cases involving probable and inducible parts.

Acknowledgments

This work was partially supported by the Project InDiGen in SPP–Sprachproduktion, grant by the Deutsche Forschungsgemeinschaft to the University of Saarbrücken and by the Lorraine Region within the project "Ingénierie des langues, du document et de l'Information scientifique Technique et Culturelle" of the Plan Etat-Région 2000-2004.

References

Clark, H. H.: 1977, 'Bridging'. In: P. N. Johnson-Laird and P. C. Wason (eds.): *Thinking: Readings in Cognitive Science*. Cambridge University Press, Cambridge.

Dale, R.: 1989, 'Cooking up referring expressions'. In: *Proc. of the 27th ACL*. pp. 68–75.

Dale, R.: 1992, *Generating Referring Expressions: Building Descriptions in a Domain of Objects and Processes*. MIT Press.

Dale, R. and E. Reiter: 1995, 'Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions'. *Cognitive Science* **18**, 233–263.

Haarslev, V. and R. Möller, 'RACER: Renamed ABox and Concept Expression Reasoner'. `http://www.fh-wedel.de/~mo/racer/index.html`.

Hawkins, J. A.: 1978, *Definiteness and Indefiniteness*. London: Croom Helm.

Heim, I.: 1982, 'The Semantics of Definite and Indefinite Noun Phrases'. Ph.D. thesis, University of Massachusetts.

Horacek, H.: 1997, 'An Algorithm for Generating Referential Descriptions With Flexible Interfaces'. In: *Proc. of the 35th ACL*. pp. 206–213.

Horrocks, I., 'The FaCT System'. `http://www.cs.man.ac.uk/~horrocks/FaCT/`.

Krahmer, E. and M. Theune: 2001, 'Efficient context-sensitive generation of referring expressions'. In: K. van Deemter and R.Kibble (eds.): *Information Sharing: Givenness and Newness in Language Processing*. CSLI Publication.

Poesio, M. and R. Vieira: 1998, 'A Corpus-based Investigation of Definite Description Use'. *Computational Linguistics* **24**(2), 183–216.

Prince, E.: 1981, 'Towards a taxonomy of given-new information'. In: P. Cole (ed.): *Radical Pragmatics*. New York: Academic Press, pp. 223–256.

Russell, B.: 1906, 'On denoting'. *Mind* **14**, 479–493.

Stone, M.: 1998, 'Modality in Dialogue: Planning, Pragmatics and Computation'. Ph.D. thesis, Department of Computer & Information Science, University of Pennsylvania.

Woods, W. and J. Schmolze: 1992, 'The KL-ONE Family'. *Computer and Mathematics with Applications* **23**(2–5).