# THE COMPUTATIONAL COMPLEXITY OF AVOIDING CONVERSATIONAL IMPLICATURES

Ehud Reiter†
Aiken Computation Lab
Harvard University
Cambridge, Mass 02138

ABSTRACT

Referring expressions and other object descriptions should be maximal under the Local Brevity, No Unnecessary Components, and Lexical Preference preference rules; otherwise, they may lead hearers to infer unwanted conversational implicatures. These preference rules can be incorporated into a polynomial time generation algorithm, while some alternative formalizations of conversational implicature make the generation task NP-Hard.

## 1. Introduction

Natural language generation (NLG) systems should produce referring expressions and other object descriptions that are *free of false implicatures*, i.e., that do not cause the user of the system to infer incorrect and unwanted *conversational implicatures* (Grice 1975). The following utterances illustrate referring expressions that are and are not free of false implicatures:

1a) "Sit by *the table*"

1b) "Sit by *the brown wooden table*"

In a context where only one table was visible, and this table was brown and made of wood, utterances (1a) and (1b) would both fulfill the referring goal: a hearer who heard either utterance would have no trouble picking out the object being referred to. However, a hearer who heard utterance (1b) would probably assume that it was somehow important that the table was brown and made of wood, i.e., that the speaker was trying to do more than just identify the table. If the speaker did not have this intention, and only wished to tell the hearer where to sit, then this would be an incorrect conversational implicature, and could lead to problems later in the discourse. Accordingly, a speaker who only wished to identify the table should use utterance (1a) in this situation, and avoid utterance (1b).

Incorrect conversational implicatures may also arise from inappropriate attributive (informational) descriptions.[1] This is illustrated by the following utterances, which might be used by a salesman who wished to inform a customer of the color, material, and sleeve-length of a shirt:

2a) "I have *a red T-shirt*"

2b) "I have *a lightweight red cotton shirt with short sleeves*"

Utterances (2a) and (2b) both successfully inform the hearer of the relevant properties of the shirt, assuming the hearer has some domain knowledge about T-shirts. However, if the hearer has this domain knowledge, the use of utterance (2b) might incorrectly implicate that the object being described was *not* a T-shirt — because if it was, the hearer would reason, then the speaker would have used utterance (2a).

Therefore, in the above situations the speaker, whether a human or a computer NLG system, should use utterances (1a) and (2a), and should avoid utterances (1b) and (2b); utterances (1a) and (2a) are free of false implicatures, while the utterances (1b) and (2b) are not. This paper proposes a computational model for determining when an object description is free of false implicatures. Briefly, a description is considered free of false implicatures if it is maximal under the *Local Brevity*, *No Unnecessary Components*, and *Lexical Preference* preference rules. These preference rules were chosen on complexity-theoretic as well as linguistic criteria; descriptions that are maximal under these preference rules can be found in polynomial time, while some alternative formalizations of the free-of-false-implicatures constraint make the generation task NP-Hard.

† Currently at the Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland.

[1] The referring/attributive distinction follows Donnellan (1966): a referring expression is intended to identify an object in the current context, while an attributive description is intended to communicate information about an object.

This paper only addresses the problem of generating free-of-false-implicatures referring expressions, such as utterance (1a). Reiter (1990a,b) uses the same preference rules to formalize the task of generating free-of-false-implicatures attributive descriptions, such as utterance (2a).

## 2. Referring Expression Model

The referring-expression model used in this paper is a variant of Dale's (1989) model for full definite noun phrase referring expressions. Dale's model is applicable in situations in which the speaker intends to refer to an object that the speaker and hearer are mutually aware of, and the speaker has no other communicative goal besides identifying the referred-to object.[2] The model assumes that objects belong to a taxonomy class (e.g., *Chair*) and possess values for various attributes (e.g., *Color:Brown*).[3] Referring expressions are represented as a classification and a set of attribute-value pairs: the classification is syntactically realized as the head noun, while the attribute-value pairs are syntactically realized as NP modifiers. Successful referring expressions are required to be *distinguishing descriptions*, i.e., descriptions that contain a classification and a set of attributes that are true of the object being referred to, but not of any other object in the current discourse context.[4]

More formally, and using a somewhat different terminology from Dale, let a *component* be either a classification or an attribute-value pair. A classification component will be written *class:Class*; an attribute-value pair component will be written *Attribute:Value*. Then, given a target object, denoted *Target*, and a set of contrasting objects in the current discourse context, denoted *Excluded*, a set of components will represent a *successful referring expression* (a distinguishing description, in Dale's terminology) if the set, denoted *RE*, satisfies the following constraints:

1) Every component in RE applies to *Target*: that is, every component in RE is either a classification that subsumes *Target*, or an attribute-value pair that *Target* possesses.

2) For every member E of *Excluded*, there is at least one component in RE that does not apply to E.

*Example*: the current discourse context contains objects A, B, and C (and no other objects), and these objects have the following classifications and attributes (of which both the speaker and the hearer are aware):

A) *Table* with *Material:Wood* and *Color:Brown*.

B) *Chair* with *Material:Wood* and *Color:Brown*

C) *Chair* with *Material:Wood* and *Color:Black*

In this context, the referring expressions {*class:Table*} ("the table") and {*class:Table, Material:Wood, Color:Brown*} ("the brown wooden table") both successfully refer to object A, because they match object A but no other object. Similarly, the referring expressions {*class:Chair, Color:Brown*} ("the brown chair") and {*class:Chair, Material:Wood, Color:Brown*} ("the brown wooden chair") both successfully refer to object B, because they match object B, but no other object. The referring expression {*class:Chair*} ("the chair"), however, does not successfully refer to object B, because it also matches object C.

## 3. Conversational Implicature

### 3.1. Grice's Maxims and Their Interpretation

Grice (1975) proposed four maxims of conversation that speakers needed to obey: *Quality*, *Quantity*, *Relevance*, and *Manner*. For the task of generating referring expressions as formalized in Section 2, these maxims can be interpreted as follows:

*Quality*: The Quality maxim requires utterances to be truthful. In this context, it requires referring expressions to be factual descriptions of the referred-to object. This condition is already part of the definition of a successful referring expression, and does not need to be restated as a conversational implicature constraint.

[2] Appelt (1985) presented a more complex referring-expression model that covered situations where the hearer was not already aware of the referred-to object, and that allowed the speaker to have more complex communicative goals. A similar analysis to the one presented in this paper could in principle be done for Appelt's model, but it would be substantially more difficult, in part because the model is more complex, and in part because Appelt did not separate his 'content determination' subsystem from his planner and his surface-form generator.

[3] All attributes are assumed to be *predicative* (Kamp 1975).

[4] Dale also suggested that NLG systems should choose distinguishing descriptions of minimal cardinality; this is discussed in footnote 7.

*Quantity*: The Quality maxim requires utterances to contain enough information to fulfill the speaker's communicative goal, but not more information. In this context, it requires referring expressions to contain enough information to enable the hearer to identify the referred-to object, but not more information. Therefore, referring expressions should be successful (as defined in Section 2), but should not contain additional elements that are unnecessary for fulfilling the referring goal.

*Relevance*: The Relevance maxim requires utterances to be relevant to the discourse. In this context, where the speaker is assumed just to have the communicative goal of identifying an object to the hearer, the maxim prohibits referring expressions from containing elements that do not help distinguish the target object from other objects in the discourse context. Irrelevant elements are also unnecessary elements, so the Relevance maxim may be considered to be a special case of the Quantity maxim, at least for the referring-expression generation task as formalized in Section 2.

*Manner*: The Brevity submaxim of the Manner maxim requires a speaker to use short utterances if possible. In this context it requires the speaker to use a short referring expression if such a referring expression exists. The analysis of the other Manner submaxims is left for future work.

An additional source of conversational implicature was proposed by Cruse (1977) and Hirschberg (1985), who hypothesized that implicatures might arise from the failure to use *basic-level* classes (Rosch 1978) in an utterance. In this paper, such implicatures are generalized by assuming that there is a *lexical-preference hierarchy* among the *lexical classes* (classes that can be realized with single lexical units) known to the hearer, and that the use of a lexical class in an utterance implicates that no preferred lexical class could have been used in its place.

In summary, conversational implicature considerations require referring expressions to be brief, to not contain unnecessary elements, and to use lexically-preferred classes whenever possible. The following requests illustrate how violations of these principles in referring expressions may lead to unwanted conversational implicatures:

3a) "Wait for me by *the pine*."
({*class:Pine*})

3b) "Wait for me by *the tree that has pinecones*."
({*class:Tree, Seed-type:Pinecone*})

3c) "Wait for me by *the 50-foot-high pine*."
({*class:Pine, Height:50-feet*})

3d) "Wait for me by *the sugar pine*."
({*class:Sugar-pine*})

If there were only two trees in the hearer's immediate surroundings, a pine and an oak, then all of the above utterances would be successful referring expressions that enabled the hearer to pick out the object being referred to (assuming the hearer could recognize pines and oaks). In such a situation, however, utterance (3b) would violate the *brevity* principle, and thus would implicate that the tree could not be described as a "pine" (which might lead the hearer to infer that the tree was not a real pine, but some other tree that happened to have pinecones). Utterance (3c) would violate the *no-unnecessary-elements* principle, and thus would implicate that it was important that the tree was 50 feet tall (which might lead the hearer to infer that there was another pine tree in the area that had a different height). Utterance (3d) would violate the *lexical-preference* principle, and thus would implicate that the speaker wished to emphasize that the tree was a sugar pine and not some other kind of pine (which might lead the hearer to infer that the speaker was trying to impress her with his botanical knowledge). A speaker who only wished to tell the hearer where to wait, and did not want the hearer to make any of these implicatures, would need to use utterance (3a), and to avoid utterances (3b), (3c), and (3d).

### 3.2. Formalizing Conversational Implicature Through Preference Rules

The brevity, no-unnecessary-elements, and lexical-preference principles may be formalized by requiring a description to be a *maximal element* under a *preference function* of the set of successful referring expressions. More formally, let D be the set of successful referring expressions, and let $\gg$ be a preference function that prefers descriptions that are short, that do not contain unnecessary elements, and that use lexically preferred classes. Then, a referring expression is considered *free of false implicatures* if it is a maximal element of D with respect to $\gg$. In other words, a description B in D is free of false implicatures if there is no description A in D, such that $A \gg B$. This formalization is similar to the partially ordered sets that Hirschberg (1985) used to formalize scalar implicatures: D and $\gg$ together form a partially ordered set, and the assumption is that the use of an element in D carries the conversational implicature that no higher-ranked element in D could have been used.

The overall preference function $\gg$ will be decomposed into separate *preference rules* that cover each type of implicature: $\gg_B$ for brevity, $\gg_U$ for unnecessary elements, and $\gg_L$ for lexical preference. $\gg$ is then defined as the disjunction of these preference rules, i.e., A $\gg$ B if A $\gg_B$ B, A $\gg_U$ B, or A $\gg_L$ B. The assumption will be made in this paper that there are no conflicts between preference rules, i.e., that it is never the case that A is preferred over B by one preference rule, but B is preferred over A by another preference rule.[5] Therefore, $\gg$ will be a partial order if $\gg_B$, $\gg_U$, and $\gg_L$ are partial orders.

### 3.3. Computational Tractability

Computational complexity considerations are used in this paper to determine exactly how the no-unnecessary-elements, brevity, and lexical-preference principles should be formalized as preference rules. Sections 4, 5, and 6 examine various preference rules that might plausibly be used to formalize these implicatures, and reject preference rules that make the generation task NP-Hard. This is justified on the grounds that computer NLG systems should not be asked to solve NP-Hard problems.[6] Human speakers and hearers are also probably not very proficient at solving NP-Hard problems, which suggests that it is unlikely that NP-Hard preference rules have been incorporated into language.

## 4. Brevity

Grice's submaxim of brevity states that utterances should be kept brief. Many NLG researchers (e.g., Dale 1989; Appelt 1985: pages 117-118) have suggested that this means generation systems need to produce the *shortest* possible utterance. This will be called the *Full Brevity* preference rule. Unfortunately, it is NP-Hard to find the shortest successful referring expression (Section 4.1). *Local Brevity* (Section 4.2) is a weaker version of the brevity submaxim that can be incorporated into a polynomial-time algorithm for generating successful referring expressions.

[5] Section 7.2 discusses this assumption.

[6] Section 7.1 discusses the computational impact of NP-Hard preference rules.

### 4.1. Full Brevity

The Full Brevity preference rule requires the generation system to generate the shortest successful referring expression. Formally, A $\gg_{FB}$ B if length(A) < length(B). The task of finding a maximal element of $\gg_{FB}$, i.e., of finding the shortest successful referring expression, is NP-Hard. This result holds for all definitions of length the author has examined (number of open-class words, number of words, number of characters, number of components).

To prove this, let *Target-Components* denote those components (classifications and attribute-value pairs) of *Target* that are mutually known by the speaker and the hearer. For each $T_j$ in *Target-Components*, let *Rules-Out*($T_j$) be the members of *Excluded* that do *not* possess $T_j$ (so, the presence of $T_j$ in a referring expression 'rules out' these members). Then, consider a potential referring expression, RE = $\{C_1, \ldots, C_n\}$. RE will be a successful referring expression if and only if

a) Every $C_i$ is in *Target-Components*

b) The union of *Rules-Out*($C_i$), for all $C_i$ in RE, is equal to *Excluded*.

For example, if the task was referring to object B in the example context of Section 2, then *Target-Components* would be {*class:Chair, Material:Wood, Color:Brown*}, *Excluded* would be {A, C}, and

*Rules-Out*(*class:Chair*) = {A}
*Rules-Out*(*Material:Wood*) = empty set
*Rules-Out*(*Color:Brown*) = {C}

Therefore, {*class:Chair, Color:Brown*} (i.e., "the brown chair") would be a successful referring expression for object B in this context.

If description length is measured by number of components,[7] finding the minimal length referring expression is equivalent to solving a *minimum set cover* problem, where *Excluded* is the set being covered, and the *Rules-Out*($T_j$) are the covering sets. Unfortunately, finding a minimal set cover is an NP-

[7] Dale's (1989) *minimal distinguishing descriptions* are, in the terminology of this paper, successful referring expressions that are maximal under Full Brevity when number of components is used as the measure of description length. Therefore, finding a minimal distinguishing description is an NP-Hard problem. The algorithm Dale used was essentially equivalent to the *greedy heuristic* for minimal set cover (Johnson 1974); as such it ran quickly, but did not always find a true minimal distinguishing description.

Hard problem (Garey and Johnson 1979), and thus solving it is in general computationally intractable (assuming that $P \neq NP$).

Similar proofs will work for the other definitions of length mentioned above. On an intuitive level, the basic problem is that finding the shortest description requires searching for the global minimum of the length function, and this global minimum (like many global minima) may be very expensive to locate.

### 4.2. Local Brevity

The Local Brevity preference rule is a weaker interpretation of Grice's brevity submaxim. It states that it should not be possible to generate a shorter successful referring expression by replacing a set of components by a single new component. Formally, $\gg_{LB}$ is the transitive closure of $\gg_{LB'}$, where A $\gg_{LB'}$ B if size(components(A)-components(B)) = 1,[8] and length(A) < length(B). The best definition of length(A) is probably the number of open-class words in the surface realization of A.

Local brevity can be checked by selecting a potential new component, finding all minimal sets of old components whose combined length is greater than the length of the new component, performing the substitution, and checking if the result is a successful referring expression. This can be done in polynomial time if the number of minimal sets is polynomial in the length of the description, which will happen if (non-zero) upper and lower bounds are placed on the length of any individual component (e.g., the surface realization of every component must use at least one open-class word, but no more than some fixed number of open-class words).

## 5. No Unnecessary Elements

The Gricean maxims of Quantity and Relevance prohibit utterances from containing elements that are unnecessary for fulfilling the speaker's communicative goals. The undesirability of unnecessary elements is further supported by the observation that humans find *pleonasms* (Cruse 1986) such as "a female mother" and "an unmarried bachelor" to be anomalous. The computational tractability of the no-unnecessary-elements principle depends on how *element* is defined: detecting unnecessary *words* in referring expressions is NP-Hard (Section 5.1), but unnecessary *components* can always be found in polynomial time (Section 5.2).

### 5.1. No Unnecessary Words

The No Unnecessary Words preference rule forbids referring expressions from containing unnecessary words. Formally, A $\gg_{UW}$ B if A's surface form uses a subset of the words used by B's surface form. There are several variants, such as only considering open-class words, or requiring the words in B to be in the same order as the corresponding words in A. All of these variants make the generation problem NP-Hard.

The formal proofs are in Reiter (1990b). Intuitively, the basic problem is that any preference that is stated solely in terms of surface forms must deal with the possibility that new parses and semantic interpretations may arise when the surface form is modified. This means that the only way a generation system can *guarantee* that an utterance satisfies the No Unnecessary Words rule is to generate all possible subsets of the surface form, and then run each subset through a parser and semantic interpreter to check if it happens to be a successful referring expression. The number of subsets of the surface form is exponential in the size of the surface form, so this process will take exponential time.

To illustrate the 'new parse' problem, consider two possible referring expressions:

4a) "the child holding a pumpkin"

4b) "the child holding a slice of pumpkin pie"

If utterances (4a) and (4b) were both successful referring expressions (i.e., the child had a pumpkin in one hand, and a slice of pumpkin pie in the other), then (4a) $\gg_{UW}$ (4b) under any of the variants mentioned above. However, because utterance (4a) has a different syntactic structure than utterance (4b), the only way the generation system could discover that (4a) $\gg_{UW}$ (4b) would be by constructing utterance (4b)'s surface form, removing the words "slice," "of," and "pie" from it, and analyzing the reduced surface form.

This problem, of new parses and semantic interpretations being uncovered by modifications to the surface form, causes difficulties whenever a preference rule is stated solely in terms of the surface form. Accordingly, such preference rules should be avoided.

[8] This is a set formula, where "-" means set-difference and "size" means number of members. The formula requires A to have exactly one component that is not present in B; B can have an arbitrary number of components that are not present in A.

### 5.2. No Unnecessary Components

The No Unnecessary Components preference rule forbids referring expressions from containing unnecessary components. Formally, A $\gg_{UC}$ B if A uses a a subset of the components used by B.

Unnecessary components can be found in polynomial time by using a simple incremental algorithm that just removes each component in turn, and checks if what is left constitutes a successful referring expression.

The key algorithmic difference between No Unnecessary Components and No Unnecessary Words is that this simple incremental algorithm will *not* work for the No Unnecessary Words preference rule. This is because there are cases where removing any single word from an utterance's surface form will leave an unsuccessful (or incoherent) referring expression (e.g., imagine removing just "slice" from utterance (4b)), but removing several words will uncover a new parse that corresponds to a successful referring expression. In contrast, if B is a successful referring expression, and there exists another successful referring expression A that satisfies components(A) $\subset$ components(B) (and hence A is preferred over B under the No Unnecessary Components preference rule), then it will be the case that any referring expression C that satisfies components(A) $\subset$ components(C) $\subset$ components(B) will also be successful. This means that the simple algorithm can always produce A from B by incremental steps that remove a single component at a time, because the intermediate descriptions formed in this process will always be successful referring expressions. Therefore, the simple incremental algorithm will always find unnecessary components, but may not always find unnecessary words.

## 6. Lexical Preference

If the attribute values and classifications used in the description are members of a taxonomy, then they can be realized at different levels of specificity. For example, the object in the parking lot outside the author's window might be called "a vehicle," "a motor vehicle," "a car," "a sports car," or "a Porsche."

The Lexical Preference rule assumes there is a *lexical-preference hierarchy* among the taxonomy's lexical classes (classes that can be realized with single lexical units). The rule states that utterances should use preferred lexical classes whenever possible. Formally, A $\gg_L$ B if for every component in A, that is a component in B that has the same structure, and the lexical class used by the A component is equal to or lexically preferred over the lexical class used by the B component.

The lexical-preference hierarchy should, at minimum, incorporate the following preferences:

i) Lexical class A is preferred over lexical class B if A's realization uses a subset of the open-class words used in B's realization. For example, the class with realization "vehicle" is preferred over the class with realization "motor vehicle."

ii) Lexical class A is preferred over lexical class B if A is a basic-level class, and B is not. For example, if *car* was a basic-level class, then "a car" would be preferred over "a vehicle" or "a Porsche."[9]

In some cases these two preferences may conflict; this is discussed in Section 7.2.

Utterances that violate either preference (i) or preference (ii) may implicate unwanted implicatures. Preference rule (ii) has been discussed by Cruse (1977) and Hirschberg (1985). Preference rule (i) may be considered to be another application of the Gricean maxim of quantity, and is illustrated by the following utterances:

5a) "Wait for me by *my car*"

5b) "Wait for me by *my sports car*"

If utterances (5a) and (5b) were both successful referring expressions (e.g., if the speaker possessed only one car), then the use of utterance (5b) would implicate that the speaker wished to emphasize that his vehicle was a sports car, and not some other kind of car.

From an algorithmic point of view, referring expressions that are maximal under the lexical-preference criteria can be found in polynomial time if the following restriction is imposed on the lexical-preference hierarchy:

*Restriction*:

> If lexical class A is preferred over lexical class B, then A must either subsume B or be subsumed by B in the class taxonomy.

For example, it is acceptable for *car* to be preferred over *vehicle* or *Porsche*, but it is not acceptable for *car* to be preferred over *gift* (because *car* neither subsumes nor is subsumed by *gift*).

If the above restriction holds, a variant of the simple incremental algorithm of Section 5.2 may be used to implement lexical preference: the algorithm simply attempts each replacement that lexical preference suggests, and checks if this results in a successful referring expression. If the restriction does not hold, then the simple incremental algorithm may fail, and obeying the Lexical Preference rule is in fact NP-Hard (the formal proof is in Reiter (1990b)).

## 7. Issues

### 7.1. The Impact of NP-Hard Preference Rules

It is difficult to precisely determine the computational expense of generating referring expressions that are maximal under the Full Brevity or No Unnecessary Words preference rules. The most straightforward algorithm that obeys Full Brevity (a similar analysis can be done for No Unnecessary Words) simply does an exhaustive search: it first checks if any one-component referring expression is successful, then checks if any two-component referring expression is successful, and so forth. Let L be the number of components in the shortest referring expression, and let N be the number of components that are potentially useful in a description, i.e., the number of members of *Target-Components* that rule out at least one member of *Excluded*. The straightforward full-brevity algorithm will then need to examine the following number of descriptions before it finds a successful referring expression:

$$\sum_{i=1}^{i=L} \begin{pmatrix} i \\ N \end{pmatrix} = \sum_{i=1}^{i=L} \frac{N!}{i!(N-i)!}$$

For the problem of generating a referring expression that identifies object B in the example context presented in Section 2, N is 3 and L is 2, so the straightforward brevity algorithm will take only 6 steps to find the shortest description. This problem is artificially simple, however, because N, the number of potential description components, is so small. In a more realistic problem, one would expect *Target-Components* to include size, shape, orientation, position, and probably many other attribute-value pairs as well, which would mean that N would probably be at least 10 or 20. L, the number of attributes in the shortest possible referring expression, is probably fairly small in most realistic situations, but there are cases where it might be at least 3 or 4 (e.g., consider "the upside-down blue cup on the second shelf"). For some example values of L and N in this range, the straightforward brevity algorithm will need to examine the following number of descriptions:

L = 3, N = 10; 175 descriptions
L = 4, N = 20; over 6000 descriptions
L = 5, N = 50; over 2,000,000 descriptions

The straightforward full-brevity algorithm, then, seems prohibitively expensive in at least some circumstances. Because finding the shortest description is NP-Hard, it seems likely (existing complexity-theoretic techniques are too weak to prove such statements) that all algorithms for finding the shortest description will have similarly bad performance *in the worst case*. It is possible, however, that there exist algorithms that have acceptable performance in almost all 'realistic' cases. Any such proposed algorithm, however, should be carefully analyzed to determine in what circumstances it will fail to find the shortest description or will take exponential time to run.

### 7.2. Conflicts Between Preference Rules

The assumption has been made in this paper that the preference rules do not conflict, i.e., that it is never the case that description A is preferred over description B by one preference rule, while description B is preferred over description A by another preference rule. This means, in particular, that if lexical class $LC_1$ is preferred over lexical class $LC_2$, then $LC_1$'s realization must not contain more open-class words than $LC_2$'s realization; otherwise, the Lexical Preference and Local Brevity preference rules may conflict.[10] This can be supported by psychological and linguistic findings that basic-level classes are almost always realized with single words (Rosch 1978; Berlin, Breedlove, and Raven 1973). However, there are a few exceptions to this rule, i.e., there do exist a small number of basic-level categories that have realizations that require more than one open-class word. For example, *Washing-Machine* is a basic-level class for some people, and it has a realization that uses two open-class words. This leads to a conflict of the type mentioned above: basic-level *Washing-Machine* is preferred over non-

[10] This assumes that the Local Brevity preference rule uses number of open-class words as its measure of description length. If number of components or number of lexical units is used as the measure of description length, then Local Brevity will never conflict with Lexical Preference.

No other conflicts can occur between the No Unnecessary Components, Local Brevity, and Lexical Preference preference rules.

basic-level *Appliance*, but *Washing-Machine*'s realization contains more open-class words than *Appliance*'s.

The presence of a basic-level class with a multi-word realization can also cause a conflict to occur between the two lexical-preference principles given in Section 6 (such conflicts are otherwise impossible). For example, *Washing-Machine*'s realization contains a superset of the open-class words used in the realization of *Machine*, so the basic-level preference of Section 6 indicates that *Washing-Machine* should be lexically preferred over *Machine*, while the realization-subset preference indicates that *Machine* should be lexically preferred over *Washing-Machine*. The basic-level preference should take priority in such cases, so *Washing-Machine* is the true lexically-preferred class in this example.

### 7.3. Generalizability of Results

For the task of generating attributive descriptions as formalized in Reiter (1990a, 1990b), the Local Brevity, No Unnecessary Components, and Lexical Preference rules are effective at prohibiting utterances that carry unwanted conversational implicatures, and also can be incorporated into a polynomial-time generation algorithm, provided that some restrictions are imposed on the underlying knowledge base. The effectiveness and tractability of these preference rules for other generation tasks is an open problem that requires further investigation.

The Full Brevity and No Unnecessary Words preference rules are computationally intractable for the attributive description generation task (Reiter 1990b), and it seems likely that they will be intractable for most other generation tasks as well. Because global maxima are usually expensive to locate, finding the shortest acceptable utterance will probably be computationally expensive for most generation tasks. Because the 'new parse' problem arises whenever the preference function is stated solely in terms of the surface form, detecting unnecessary words will also probably be quite expensive in most situations.

## 8. Conclusion

Referring expressions and other object descriptions need to be brief, to avoid unnecessary elements, and to use lexically preferred classes; otherwise, they may carry unwanted and incorrect conversational implicatures. These principles can be formalized by requiring referring expressions to be maximal under the Local Brevity, No Unnecessary Components, and Lexical Preference preference rules. These preference rules can be incorporated into a polynomial-time algorithm for generating free-of-false-implicatures referring expressions, while some alternative preference rules (Full Brevity and No Unnecessary Words) make this generation task NP-Hard.

**Acknowledgements**

Many thanks to Robert Dale, Joyce Friedman, Barbara Grosz, Joe Marks, Warren Plath, Candy Sidner, Jeff Siskind, Bill Woods, and the anonymous reviewers for their help and suggestions. This work was partially supported by a National Science Foundation Graduate Fellowship, an IBM Graduate Fellowship, and a contract from U S WEST Advanced Technologies. Any opinions, findings, conclusions, or recommendations are those of the author and do not necessarily reflect the views of the National Science Foundation, IBM, or U S WEST Advanced Technologies.

**References**

Appelt, D. 1985 *Planning English Referring Expressions*. Cambridge University Press: New York.

Berlin, B.; Breedlove, D.; and Raven, P. 1973 General Principles of Classification and Nomenclature in Folk Biology. *American Anthropologist* 75:214-242.

Cruse, D. 1977 The pragmatics of lexical specificity. *Journal of Linguistics* 13:153-164.

Cruse, D. 1986 *Lexical Semantics*. Cambridge University Press: New York.

Dale, R. 1989 Cooking up Referring Expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*.

Donnellan, K. 1966 Reference and Definite Descriptions. *Philosophical Review* 75:281-304.

Garey, M. and Johnson, D. 1979 *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman: San Francisco.

Grice, H. 1975 Logic and conversation. In P. Cole and J. Morgan (Eds.), *Syntax and Semantics: Vol 3, Speech Acts*, pg 43-58. Academic Press: New York.

Hirschberg, J. 1985 *A Theory of Scalar Implicature*. Report MS-CIS-85-56, LINC LAB 21. Department of Computer and Information Science, University of Pennsylvania.

Johnson, D. 1974 Approximation algorithms for combinatorial problems. *Journal of Computer and Systems Sciences* 9:256-178.

Kamp, H. (1975) Two Theories about Adjectives. In E. Keenan (Ed.) *Formal Semantics of Natural Language*, pg 123-155. Cambridge University Press: New York.

Reiter, E. 1990a Generating Descriptions that Exploit a User's Domain Knowledge. To appear in R. Dale, C. Mellish, and M. Zock (Eds.), *Current Research in Natural Language Generation*. Academic Press: New York.

Reiter, E. 1990b *Generating Appropriate Natural Language Object Descriptions*. Ph.D thesis. Aiken Computation Lab, Harvard University: Cambridge, Mass.

Rosch, E. 1978 Principles of Categorization. In E. Rosch and B. Lloyd (Eds.), *Cognition and Categorization*. Lawrence Erlbaum: Hillsdale, NJ.