# Linear Encodings of Linguistic Analyses

Samuel S. Epstein
Bell Communications Research
445 South Street, 2Q-350
Morristown, NJ 07960-1910
USA
epstein@flash.bellcore.com

## 1. Introduction

Natural languages contain families of expressions such that the number of readings of an expression is an exponential function of the length of the expression. Two well-known cases involve prepositional phrase attachment and coordination. Other cases discussed below involve anaphora and relative operator scope. For example, in

> (1) John said that Bill said that ... that Harry said that he thinks that he thinks that ... that he thinks that it is raining.

each *he* can refer to any of *John, Bill, ..., Harry.*[1] Thus if (1) contains n names and m occurrences of *he,* this sentence has $n^m$ readings (assuming that all anaphoric relationships are intrasentential). We discuss below families of expressions whose ambiguities grow as various exponential functions (factorial, Fibonacci, Catalan) of expression length.

It appears, then, that exhaustive linear-time processing of natural language is impossible. An exponentially long answer cannot be produced in linear time.[2] On the other hand, human processing of natural language seems to be at least linearly fast. The ready explanation for this is that people do not recover all readings of ambiguous expressions. This is clearly correct, as far as it goes.

This paper shows how to encode in linear space the exponentially large numbers of readings associated with various families of expressions. The availability of these encodings removes an apparent obstacle to exhaustive analyses of these expressions in iinear time. The encodings may thus be useful for practical computational purposes. They may also provide a better basis than exponential-space encodings for explanations of how humans process language.

For each of the linguistic constructions discussed in this paper, there is a simple program that generates analyses of the construction. If there are no constraints on what counts as a linguistic analysis, then a specification of a program, which requires constant space, together with a specification of an input expression, which requires linear space, could count as a linear encoding of an analysis of the input. Intuitively, there is a vast qualitative divide between a (program,input) pair on one hand, and, for instance, a forest of constituent structure trees on the other hand. More generally, a question arises of how to distinguish analyses from procedures that yield analyses. This paper will not attempt to answer this question definitively. The analyses presented in Sections 2 - 4 all satisfy a notion of "legal" analysis that excludes (program,input) pairs. Sections 2 and 3 discuss polynomial space analyses. Section 4 adds a representational device to the repertory of Sections 2 and 3, so that linear space analyses are possible. Section 5 informally discusses a variety of issues, including the distinction between analysis and procedure.

## 2. Analyses in Conjunctive Normal Form

Assume that example (1) involves no ambiguities except for antecedents of pronouns. Assume further that the length of the analysis of (1), aside from the specification of antecedents of pronouns, grows linearly.[3] Let the proposition q comprise all aspects of the analysis of (1), aside from specifications of antecedents of pronouns. Let the proposition $p_{i,j}$ comprise the specification that the j-th name in (1) is the antecedent of the i-th occurrence of *he.* (For example, $p_{1,2}$ comprises the

1. (1) is of course highly unnatural in a sense. However, it effectively isolates for study a phenomenon that is intrinsic to natural language. Similar observations apply to the examples below.
2. It is of course also the case that an exponentially long answer cannot be produced in polynomial time. If the problem cannot be reformulated so that answers are not exponentially long, the question of tractability does not arise. See [Garey and Johnson 79] and [Barton, Berwick, and Ristad 87] for related discussions.
3. These assumptions, and similar assumptions for other examples below, permit a briefer discussion than would otherwise be possible. Reservations about these assumptions do not affect the substance of the discussion. Our concern with (1) focuses on exponentially growing possibilities for assigning antecedents to pronouns.

specification that *Bill* is the antecedent of the most shallowly embedded *he.)* Let n be the number of names in (1) and let m be the number of occurrences of *he*. Then an exhaustive analysis of (1) can take the following form:

$$(1\text{-a}) \quad (q \,\&\, p_{1,1} \,\&\, p_{2,1} \,\&\, \ldots \,\&\, p_{m,1}) \vee$$
$$(q \,\&\, p_{1,1} \,\&\, p_{2,1} \,\&\, \ldots \,\&\, p_{m\text{-}1,1} \,\&\, p_{m,2}) \vee$$
$$\vdots$$
$$(q \,\&\, p_{1,n} \,\&\, p_{2,n} \,\&\, \ldots \,\&\, p_{m,n})$$

(1-a), which contains $n^m$ disjuncts, is in Disjunctive Normal Form (DNF). Each disjunct fully specifies a possible interpretation of (1). It is an implicit assumption in much of the literature that the proper form for linguistic analyses is DNF. An analysis in DNF amounts to a listing of possible global interpretations.

(1-a) is logically equivalent to the following statement in Conjunctive Normal Form (CNF):

$$(1\text{-b}) \quad q \,\&\, (p_{1,1} \vee p_{1,2} \vee \ldots \vee p_{1,n}) \,\&$$
$$(p_{2,1} \vee p_{2,2} \vee \ldots \vee p_{2,n}) \,\&$$
$$\vdots$$
$$(p_{m,1} \vee p_{m,2} \vee \ldots \vee p_{m,n})$$

(1-b) contains m+1 conjuncts. The length of an exhaustive analysis of (1) is exponential in the number of pronouns in (1) when the analysis is given in DNF, but linear in the number of pronouns when the analysis is given in CNF. However, (1-b) is not linear in the length of (1), because each of m conjuncts contains n disjuncts, so that a total of m×n literals is required to specify anaphoric possibilities.

The following example has an analysis in DNF that grows as the factorial of the length of the input:

> (2) John told Bill that Tom told him that Fred told him that ... that Jim told him that Harry told him that it is raining.

The first occurrence of *him* can have *John* or *Bill* as antecedent. The second occurrence of *him* can have *John* or *Bill* or *Tom* as antecedent, and so on. (2) has an obvious analysis in CNF whose length is a quadratic function of the length of the input, namely

$$(2\text{-a}) \quad q \,\&\, (p_{1,1} \vee p_{1,2}) \,\&$$
$$(p_{2,1} \vee p_{2,2} \vee p_{2,3}) \,\&$$
$$\vdots$$
$$(p_{m,1} \vee p_{m,2} \vee \ldots \vee p_{m,m+1})$$

where the notation follows the same conventions as in (1-a,b).

The number of readings for the following noun phrase grows as the Catalan of the number of prepositional phrases:

> (3) the block in the box on the table ... in the kitchen

As [Church and Patil 82] discuss, examples like (3) are similar to other structures with systematic attachment ambiguities, such as coordination structures. While the number of readings of (3) is thus exponential in the length of (3), (3) has an $O(n^4)$ length analysis in CNF as follows:

$$(3\text{-a}) \quad q \,\&\, (p_{1,0}) \,\&$$

(3-a-2)
$$(p_{2,0} \vee p_{2,1}) \,\&$$

(3-a-3)
$$(p_{3,0} \vee p_{3,1} \vee p_{3,2}) \,\&$$
$$(p_{3,1} \supset p_{2,1}) \,\&$$

(3-a-4)
$$(p_{4,0} \vee p_{4,1} \vee p_{4,2} \vee p_{4,3}) \,\&$$
$$(p_{4,1} \supset p_{2,1}) \,\&$$
$$(p_{4,1} \supset (p_{3,1} \vee p_{3,2})) \,\&$$
$$(p_{4,2} \supset p_{3,2}) \,\&$$
$$\vdots$$

(3-a-k)
$$(p_{k,0} \vee p_{k,1} \vee \ldots \vee p_{k,k\text{-}1}) \,\&$$

(3-a-k,1)
$$(p_{k,1} \supset p_{2,1}) \,\&$$
$$(p_{k,1} \supset (p_{3,1} \vee p_{3,2})) \,\&$$
$$\vdots$$
$$(p_{k,1} \supset (p_{k\text{-}1,1} \vee p_{k\text{-}1,2} \vee \ldots \vee p_{k\text{-}1,k\text{-}2})) \,\&$$
$$\vdots$$

(3-a-k,m)
$$(p_{k,m} \supset p_{m+1,m}) \,\&$$
$$(p_{k,m} \supset (p_{m+2,m} \vee p_{m+2,m+1})) \,\&$$
$$\vdots$$
$$(p_{k,m} \supset (p_{k\text{-}1,m} \vee p_{k\text{-}1,m+1} \vee \ldots \vee p_{k\text{-}1,k\text{-}2})) \,\&$$
$$\vdots$$

In (3-a), $p_{i,j}$ comprises the specification that constituent i attaches to constituent j, where *the block* is constituent 0, *in the box* is constituent 1, *on the table* is constituent 2, and so on. Constituent k must attach to some constituent that lies to its left. If constituent k attaches to

constituent m, then the constituents between constituent m and constituent k cannot attach to constituents to the left of constituent m.[4]

For each pair (k,m), the number of atoms in (3-a-k,m) is $f_1(k,m) = \sum_{i=2}^{k-m} i$. $f_1(k,m)$ is quadratic in k. For each k, then, the number of atoms in (3-a-k) is $f_2(k) = k + \sum_{i=1}^{k-2} f_1(k,i)$, a cubic function in k. The number of atoms in (3-a) (excluding atoms hidden in q) is thus $\sum_{i=1}^{n} f_2(i)$, a quartic function in n. (3-a) is certainly not the most compressed CNF analysis of (3). It is, however, easy to describe.

Given an exhaustive analysis in DNF, choosing a global interpretation requires exactly one operation of selecting a disjunct. For (1-b) and (2-a), choosing a global interpretation requires a number of selections that is linear in the length of the input. I am aware of no other reason for preferring DNF to CNF for analyses of examples like (1) and (2). In favor of preferring CNF there is the practical advantage of polynomial-space output, with its implications for speed of processing. There is also the possibility of more accurate psycholinguistic modeling. It seems likely that people make decisions on antecedents for pronouns in examples like (1) and (2) locally, on a pronoun-by-pronoun basis, and that they do not choose among global analyses.[5] In contrast, the conjuncts of (3-a) clearly do not correspond one-to-one with processing decisions. Section 4 discusses an analysis of (3) whose components may correspond to local decisions on attachment sites.

## 3. Encodings with non-atomic propositional constants

It is possible to get a cubic length analysis of (3) by introducing constants for non-atomic propositions. For m<k, let $r_{k,m}$ be $(p_{k-1,m+1} \vee p_{k-1,m+2} \vee \ldots \vee p_{k-1,k-2})$. Then (3-a-k,m) is equivalent to:

(3-b-k,m) $(p_{k,m} \supset p_{m+1,m}) \;\&$
$(p_{k,m} \supset (p_{m+2,m} \vee p_{m+2,m+1})) \;\&$
.
.
.
$(p_{k,m} \supset (p_{k-2,m} \vee r_{k-1,m}))$
$(p_{k,m} \supset (p_{k-1,m} \vee r_{k,m}))$

Of course, the space required to define the $r_{k,m}$ must figure in the space required to encode an analysis of (3) along the lines of (3-b-k,m). $r_{k,m-1} \equiv (p_{k-1,m} \vee r_{k,m})$, so it requires quadratic space to define all the $r_{k,m}$. A revised version of (3) with (3-b-k,m) in place of (3-a-k,m) throughout requires cubic space.[6]

Tree representations of single readings for examples like (3) may be viewed as follows: edges correspond to atomic propositions that comprise specifications like "constituent i attaches to constituent j" or "constituent i projects to constituent j."[7] A non-terminal node A corresponds to a constituent, but also corresponds to the conjunction of the atomic propositions that correspond to edges that A dominates. Thus the root node of the tree corresponds to a proposition that comprises a full specification of constituent structure.

The situation is essentially the same for shared forests. ([Tomita 87] discusses shared forests and packed shared forests.) Edges in shared forests correspond to atomic propositions, and non-terminal nodes correspond to non-atomic propositions. To extend this perspective, shared forests compress the information in non-shared forests by exploiting the introduction of constants for non-atomic propositions. In a shared forest, the subtree that a node dominates is written only once. In effect, then, a constant is introduced that represents the conjunction that corresponds to the node. This constant is a constituent of the formulas that correspond to superior nodes. While shared forests are more compressed than unshared forests, the number of nodes in the shared forest representation of (3) is still exponential in the length of (3).

In a packed shared forest, a packed node that does not dominate any packed nodes corresponds to a disjunction of conjunctions of atomic propositions. Packed nodes that dominate other packed nodes correspond to disjunctions of conjunctions of atomic and non-atomic propositions. In effect, for each node (packed or non-packed), a constant is introduced that abbreviates the formula that corresponds to the node. Exploitation of constants for non-atomic propositions permits more significant compression for packed shared forests than for shared forests. The packed root node of a packed shared forest for (3) corresponds to a disjunction of conjunctions whose size in atoms is exponential in the length of (3). However, the number of nodes of a packed shared forest for (3) goes up as the square of the length of (3). The number of edges of the packed shared forest (a more authentic measure of the size of the forest) goes up as the cube of the length.

4. $(p_1 \supset (p_2 \vee \ldots \vee p_j))$ is equivalent to $(\neg p_1 \vee p_2 \vee \ldots \vee p_j)$, so that (3-a) is in CNF.

5. This is not to suggest that people produce an exhaustive analysis in CNF prior to choosing a reading. The hypothesis is rather that fragments of a CNF representation are produced (in some sense) during processing.

6. Further compression is possible if we allow quantification over subscript indices. However, quantification over artifacts of representation may uncontroversially involve crossing the divide between analysis and procedure.

7. Details of constituent structure are not relevant to the discussion here. For example, we will not distinguish "X attaches to V" from "X attaches to VP."

## 4. Encodings that introduce structural constants

A linear length encoding of an analysis of (1) is possible if we use the constant A = {John, Bill, ... , Harry} in the encoding as follows:

(1-c) q & (antecedent(pronoun$_1$) $\in$ A) &
(antecedent(pronoun$_2$) $\in$ A) &
.
.
.
(antecedent(pronoun$_m$) $\in$ A)

Note that "x $\in$ Y" is short-hand for the disjunction of the statements "x = y," where y ranges over Y, so that (1-c) is not very different from (1-b). Examples below involve freer use of constants that correspond to sets of linguistic entities. I will call such constants "structural."

A linear analysis of (2) is possible if we introduce constants $A_1$, ... , $A_m$, where $A_1$ = {John, Bill}, $A_2 = A_1 \cup$ {Tom}, $A_3 = A_2 \cup$ {Fred}, ... , $A_m = A_{m-1} \cup$ {Jim}:

(2-b) q &
(antecedent(pronoun$_1$) $\in A_1$) &
(antecedent(pronoun$_2$) $\in A_2$) &
.
.
.
(antecedent(pronoun$_m$) $\in A_m$)

Because $A_i$ can be defined in terms of $A_{i-1}$, only linear space is required to define these constants. It is convenient to mix definitions of constants with other aspects of the encoding of (2), as follows:

(2-c) q & $A_1$ = {John, Bill} &
(antecedent(pronoun$_1$) $\in A_1$ &
$A_2 = (A_1 \cup$ {Tom})) &
(antecedent(pronoun$_2$) $\in A_2$ &
$A_3 = (A_2 \cup$ {Fred})) &
.
.
.
(antecedent(pronoun$_{m-1}$) $\in A_{m-1}$ &
$A_m = (A_{m-1} \cup$ {Jim})) &
(antecedent(pronoun$_m$) $\in A_m$)

For (2), the introduction of structural constants permits a linear encoding. For the following example, the introduction of structural constants likewise permits a linear encoding:

> (4) Many teachers expect several students to expect many teachers to expect several students to ... to expect many teachers to expect several students to read some book.

Each quantifier in (4) can take scope over the quantifier to its immediate left (if any), and can take scope over the quantifier to its immediate right (if any). However, if a quantifier $Q_i$ takes scope over $Q_{i-1}$ to its immediate left, then the quantifier $Q_{i+1}$ to the immediate right of $Q_i$ cannot take scope over $Q_i$. (See [Epstein 88] for a discussion of relative operator scope.) It follows that the number of relative operator scope readings for (4) grows as the Fibonacci of the length of (4).[8] However, a linear encoding of an exhaustive analysis of (4) is as follows:

(4-a) q &

$[((Q_1 > Q_2) \& (L_1 = Q_2)) \vee$
$((Q_2 > Q_1) \& (L_1 = T))] \&$
$[Q_1 > Q_3] \&$
$[((L_1 > Q_3) \& (L_2 = Q_3)) \vee$
$((Q_3 > L_1) \& (L_2 = T))] \&$
.
.
.
$[Q_{k-2} > Q_{k+1}] \&$
$[Q_{k-1} > Q_{k+1}] \&$
$[((L_{k-1} > Q_{k+1}) \& (L_k = Q_{k+1})) \vee$
$((Q_{k+1} > L_{k-1}) \& (L_k = T))] \&$
.
.
.

Here q represents aspects of the analysis of (4) aside from the specification of relative operator scope, and $Q_i$ represents the i-th quantifier in (4), reading from the left. The $L_i$ are introduced constants corresponding to quantifiers that can have lower scope than some more deeply embedded quantifier. "$Q_i > Q_j$" means that $Q_i$ has higher scope than $Q_j$. For all Q, "$Q < T$" is true and "$Q > T$" is false.[9] Note that if we delete from (4-a) propositions that assign values to introduced constants, such as "$(L_1 = Q_2)$," the resulting statement is in CNF.

Section 3 discussed cubic length analyses of (3) with propositional constants. (3) has a linear analysis with structural constants as follows:

---

8. The most deeply embedded clause in (4) has 2 possible relative scope readings. The second most deeply embedded clause in (4) has 3 possible relative scope readings (many>several>some, many>some>several, several>many>some). Let $S_k$ be the k-th most deeply embedded clause in (4). ($S_k$ is immediately embedded in $S_{k+1}$.) Given that $S_k$ has a total of n (relative operator) scope readings, and that $S_{k-1}$ has a total of m scope readings, then the subject of $S_{k+1}$ can take scope over all the quantifiers in $S_k$, accounting for n global readings over $S_{k+1}$. Alternatively, the subject of $S_k$ can take scope over the subject of $S_{k+1}$. Then both these subjects take scope over all the quantifiers in $S_{k-1}$. The second alternative thus accounts for m additional global readings over $S_{k+1}$.
9. (4-a) does not explicitly state, for example, that $Q_1 > Q_5$, but this fact can be derived from (4-a) through application of the transitivity of relative operator scope. Generally speaking, linguistic representations don't explicitly include all their consequences.

(3-c) q &

$$[(ap(PP_1) = NP) \;\&\; (AP_1 = NP) \;\&\; (RE_1 = \{NP, PP_1\})] \;\&$$

$$[(ap(PP_2) \in RE_1) \;\&\; (AP_2 = ap(PP_2)) \;\&\; (RE_2 = (RE_1 \uparrow AP_2) \cup \{PP_2\})] \;\&$$

$$\vdots$$

$$[(ap(PP_k) \in RE_{k-1}) \;\&\; (AP_k = ap(PP_k)) \;\&\; (RE_k = (RE_{k-1} \uparrow AP_k) \cup \{PP_k\})] \;\&$$

$$\vdots$$

Here q represents aspects of the analysis of (3) aside from the specification of attachment points for the prepositional phrases. The desired solutions consist of specifications of attachment possibilities, stated in the form "$ap(PP_k) \in X$" ("attachment point of the k-th PP is one of the elements of X") in (3-c). The $AP_k$ and $RE_k$ are introduced constants. $AP_k$ is the attachment point of $PP_k$.[10] $RE_k$ represents the right edge of a constituent structure tree for the string consisting of *the block* and the first k PP's. (3-c) is in a sort of relaxed CNF, as discussed above in connection with (4-a), and in Section 5 below. "↑" in (3-c) is defined so that $RE \uparrow AP = \{AP\} \cup \{X \in RE \mid X \text{ precedes } AP\}$. (When $PP_k$ to the right of $PP_i$ attaches above $PP_i$, $PP_i$ is not in the right edge of the resulting structure, and is unavailable for attachment by material to the right of $PP_k$.)

As for (3), the number of readings of the following example (from [Church and Patil 82]) grows as the Catalan of the number of prepositional phrases:

> (5) Put the block in the box on the table ... in the kitchen.

However, there is an important difference between (3) and (5). In (3), any number of PP's can attach to *the block*, any number of PP's can attach to *the box*, and so on. No NP in (3) requires complements. (*In the box* must attach to *the block*, but only because *the block* is the only NP that lies to the left of *in the box*.) In (5), on the other hand, *put* requires one NP argument and one PP argument, and cannot accept any other complements.[11] An analysis of (5) along the lines of (3-c) would incorrectly include readings where more than one PP attaches to *put*, and readings where no PP attaches to *put*. A linear analysis of (5) is as follows:

(5-a) q &

$$[(ap(PP_1) \in \{VP, NP\}) \;\&\; (AP_1 = ap(PP_1)) \;\&\; (RE_1 = \{VP, NP\} \uparrow AP_1) \;\&\; (OG_1 = \{VP\} - \{AP_1\})] \;\&$$

$$[(ap(PP_2) \in RE_1) \;\&\; (AP_2 = ap(PP_2)) \;\&\; (RE_2 = (RE_1 \uparrow AP_2) \cup \{PP_2\}) \;\&\; (OG_2 = OG_1 - \{AP_2\})] \;\&$$

$$\vdots$$

$$[(ap(PP_k) \in RE_{k-1}) \;\&\; (AP_k = ap(PP_k)) \;\&\; (RE_k = (RE_{k-1} \uparrow AP_k) \cup \{PP_k\}) \;\&\; (OG_k = OG_{k-1} - \{AP_k\})] \;\&$$

$$\vdots$$

$$[(ap(PP_n) \in OG_{n-1} \square RE_{n-1})]$$

(5-a) is similar to (3-c), but includes the additional constants $OG_k$. $OG_k$ is the open (theta-)grid for the substructure corresponding to *put the block* followed by the first k PP's. $OG_k$ is either {VP}, if none of the first k PP's is attached to V, or is empty. Non-empty $OG_k$ indicates that for each constituent X in $OG_k$, some $PP_i$, $k<i\leq n$, must attach to X. "□" in (5-a) is defined so that $A \square B$ is equal to A if A is non-empty, and is otherwise equal to B. The final conjunct in (5-a) captures the requirement that if none of the first n-1 PP's attaches to *put*, then the final PP must attach to this verb.

## 5. Issues

The example constructions presented above illustrate a variety of abstract cases. In (1), local ambiguities are independent of each other. The assignment of an antecedent to a pronoun in (1) does not affect the possibilities of antecedent assignment for other pronouns in (1). An analysis of (1) in CNF need not include more than one appearance of any literal. (2) is similar to (1) in this respect. In (4), local ambiguities are interdependent, but local ambiguity possibilities depend only on ambiguity possibilities in neighboring clauses. There is thus a bound on how many ambiguities can interact. In (3), on the other hand, there is no such bound. Choosing an attachment site for $PP_j$ affects the attachment possibilities for $PP_{j+k}$, no matter how large k is. (5) is similar to (3), but also involves a global filter associated with the verb *put*. (3-c) and (5-a) employ a richer repertory of operators on structural constants (-, □, ↑) than is found in (1-c), (2-b), (2-c), and (4-a).

(1-b) may qualify relatively easily as an exhaustive analysis of (1), according to a common conception of what constitutes an analysis. (3-c), on the other hand, appears to have some of the earmarks of a procedure. The similarity of introduced constants to local variables is obvious. In particular, the constants $AP_i$ and $RE_i$ of

10. "$PP_i$ attaches to $PP_k$" really means that $PP_i$ attaches to the object of the preposition head of $PP_k$. This usage permits a briefer discussion than would otherwise be possible.

11. This characterization of *put* is not strictly speaking correct, but the necessary qualifications are irrelevant to the discussion here.

(3-c) could be replaced with two local variables AP and RE that receive destructive assignment. (3-c) also employs the operators "↑" and "∪", which might be regarded as corresponding to procedures. Whether (3-c) is an analysis or a procedure for computing analyses is ultimately a matter of selecting a definition for "linguistic analysis."

Criteria for a successful definition of "linguistic analysis" might appeal to psychological reality. One possible requirement is that components of analyses correspond to partial analyses built during human processing. When definitions of constants (assignments to local variables) are blended into what is otherwise a CNF formula, as in (2-c), (3-c), (4-a), and (5-a), the result might be called "relaxed CNF." Somewhat more precisely, suppose that a formula in "relaxed CNF" is a conjunction of "relaxed disjunctions," where a "relaxed disjunction" is the conjunction of a "generalized disjunction" with an "assignment formula." A "generalized disjunction" may be either a disjunction of atoms, or a statement of the form "x ∈ A." An "assignment formula" is a conjunction of statements that assign values to constants. Given such a relaxed CNF formula as an exhaustive analysis, obtaining an analysis of a single reading requires for each generalized disjunction the choice of a disjunct or the selection of an element. Such single-reading analyses may be produced by deterministic variants of non-deterministic processing models that produce exhaustive analyses in relaxed CNF. Relaxed CNF is compatible with a variety of processing models. For example, a component of the form "x ∈ A" might be produced before components that specify the contents of A.

Recent work on Kolmogorov complexity might provide alternative criteria for the definition of "linguistic analysis." ([Li and Vitányi 89] is a recent survey of work on Kolmogorov complexity.) In particular, notions of time-bounded algorithmic complexity, such as the "logical depth" of [Bennett 88], may be relevant. Following a third alternative, a satisfactory definition of "analysis" may involve a correspondence principle, along the following lines: every component of a "legal" analysis specifically mentions one or more components of the input. For this to work, "component" and "mention" themselves require appropriate definitions. Arbitrary fragments of analyses cannot count as "components." "Mention" should be transitive.

Analyses in relaxed CNF may be more compatible with principle-based grammars than are tree-based analyses. ([Chomsky 81] is the seminal work on principle-based grammars.) Constituent structure does not occupy as central a place in the principle-based paradigm as in other grammatical paradigms. Each generalized disjunction (or its deterministic counterpart) supplies a piece of information about the analyzed expression. Assignment formulas capture logical dependencies among these pieces. Each of the examples in this paper illustrates a single phenomenon. Relaxed CNF can also capture interactions among phenomena.

Analyses like (3-c) are probably less easily readable than packed shared forests. Full analyses that specify constituent structure information together with relative operator scope information, information on anaphora, and so on, will be even less readable. It may be possible to devise a more graphically oriented notation for linear encodings of linguistic analyses.

Whatever conception of "linguistic analysis" may ultimately prove most useful, it seems clear that working with relaxed Conjunctive Normal Form has advantages over working with Disjunctive Normal Form for computational applications. Relaxed CNF also appears to have advantages over DNF for psycholinguistic modeling. Introduced constants (local variables) have obvious utility in implementations. They may also play a role in human processing of language. In particular, as human processing proceeds, explicit details of previously encountered structure may recede into the background yet remain accessible.

## Acknowledgments

I am indebted for comments and discussions to Steven Abney, Yves Caseau, and Andrew Ogielski. Responsibility for errors is entirely mine.

## References

E. Barton, R. Berwick, and E. Ristad, *Computational Complexity and Natural Language*, MIT Press, Cambridge, Massachusetts, 1987.

C. Bennett, "Logical Depth and Physical Complexity," in R. Herken (ed.), *The Universal Turing Machine; A Half-Century Survey*, pp. 227-258, Oxford University Press, Oxford, 1988.

N. Chomsky, *Lectures on Government and Binding*, Foris Publications, Dordrecht, 1981.

K. Church and R. Patil, "Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table," *American Journal of Computational Linguistics*, 8:3-4, pp. 139-149, 1982.

S. Epstein, "Principle-Based Interpretation of Natural Language Quantifiers," *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pp. 718-723, 1988.

M. Garey and D. Johnson, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.

M. Li and P. Vitányi, *Kolmogorov Complexity and Its Applications (Revised Version)*, Report CS-R8901, Centre for Mathematics and Computer Science, Amsterdam, 1989.

M. Tomita, "An Efficient Augmented-Context-Free Parsing Algorithm," *Computational Linguistics*, 13:1-2, pp. 31-46, 1987.