An Implementable Semantics for Comparative Constructions
by
Manny Rayner and Amelie Banks

SICS R88018
Research Report
ISSN 0283-3638
AN IMPLEMENTABLE SEMANTICS
FOR COMPARATIVE CONSTRUCTIONS

Manny Rayner, Amelie Banks

Swedish Institute of Computer Science
Box 1263
S-164 28 KISTA
Sweden

Tel: +46 8 752 15 00

ABSTRACT

We describe a fairly comprehensive handling of the syntax and semantics of comparative constructions. The analysis is largely based on the theory developed by Pinkham, but we advance arguments to support a different handling of phrasal comparatives - in particular, we replace the use of C-ellipsis with a method of interpretation we call contrastive comparison. We explain the reasons for dividing comparative sentences into different categories, and for each category we give an example of the corresponding Montague semantics. The ideas have all been implemented within a large-scale grammar for Swedish, a "toy" version of which is presented, along with examples of the output.

Keywords: Comparatives, Semantics, Montague grammar

Note: Earlier versions of several of the results herein have already been published in [Banks 86], [Rayner & Banks 86], [Banks & Rayner 87] and [Rayner & Banks 88a]. Since a number of important changes have been made (especially in the treatment of contrastive comparatives, section 5), this report is to be taken as superseding all four.
# Table of Contents

1. Introduction .................................................. 3

2. Previous work on comparatives .............................. 4

3. Clausal comparatives .......................................... 5

4. Implementing a deletion approach to non-clausal comparison .......................... 7

5. Contrastive comparative: a nontransformational account of C-ellipsis ........... 9
   5.1. Basic ideas .................................................. 9
   5.2. Comparatives with "s-operators" ......................... 17
   5.3. "Parallel" phrasal comparatives ............................. 18
   5.4. Other contrastive constructions ..............................
      5.4.1 Vp-operators ........................................... 21
      5.4.2 Contrasted prepositions .............................. 21
   5.5. Ambiguities in contrastive constructions ................. 22

6. "Simple" phrasal comparatives ................................ 22

7. Other issues .................................................. 24
   7.1 Non-coreference of compared objects ....................... 25
   7.2 "Before" and "after" ......................................

8. Data from corpus analyses .................................... 26

9. Conclusions and further directions ........................ 28

Acknowledgements ............................................... 32

References ....................................................... 33

Appendix 1: Terminology ........................................ 35

Appendix 2: A logic grammar for comparatives ................. 36

Appendix 3: Examples ........................................... 54
1. INTRODUCTION

This paper is written with two distinct audiences in mind. On the practical side, we try to present a cookbook which the natural language interface implementor can use if he wishes to incorporate comparative constructions into his system's coverage. This is, we trust, interesting in itself; a quick glance at Table 1 should be enough to show that this construction is more common than is perhaps generally realized. We also try to give a usable classification of the various kinds of constructions generally lumped together under the blanket heading of "Comparative Ellipsis".

Examples of comparatives

1) John is taller than Mary.  
2) Few people run as fast as John.  
3) John bought more books than Mary.  
4) John was happier in New York than in London.  
5) John has more books than Mary has newspapers.  
6) John was born in the same city as Mary.  
7) Mary had more friends than John thought.  
8) John hit Mary harder than he meant to.  
9) More men than women bought the book.  
10) More people voted for the proposal than against.  
11) Mary seems brighter than most of the pupils.

Adjectival comparison  
Adverbal comparison with "as"  
Determiner comparison  
Comparison on PP  
Clausal comparison  
"Same" comparison  
"S-operator" comparison  
"VP-operator" comparison  
"CN-comparison"  
Preposition comparison  
"Simple" phrasal comparison

Table 1

On the theoretical side, we want to reexamine some fundamental questions concerning the nature of the comparative construction; we are going to argue that our practical work fairly strongly supports a hypothesis that has already appeared in several forms in the theoretical literature, namely that "comparative ellipsis" is a semantic rather than syntactic phenomenon. We expand more on this theme in section 2. In section 3 we present our handling of clausal comparison, which is a straightforward implementation of the theories of [Klein 80] and [Pinkham 85].

The next three sections cover non-clausal comparison, and constitute the main part of the paper. Section 4 describes our first implementation, which followed the traditional viewpoint: all comparison is, or can be derived from, clausal comparison. Thus for example 2) above would be regarded as derived from something like Few people run at a speed X such that John runs at speed X. In section 5 we first discuss some problems that arise from this treatment, and then suggest an alternative solution. This is not so easy to summarize succinctly, but its central idea is that the "duplication" of material is not a syntactic copying of parts of the parse-tree, but is instead a double application of a higher-order function constructed during the semantic analysis phase. We then demonstrate at length how this method can be used to handle a fairly wide variety of "elliptic constructions", some of which present problems for the syntactic approach. Section
6 deals with a special case of phrasal comparatives which we call "simple" phrasal comparatives, which we also handle by a direct interpretation of the phrase structure, rather than copying a constituent.

In section 7 we briefly discuss non-coreference of compared objects; we also look at sentences containing the words "before" and "after", pointing out some differences between these and comparative sentences. Our corpus analysis is presented in section 8, with some comments. In section 9 we summarize our results, and in particular address ourselves to the problem of justifying our classification of comparatives into separate categories instead of providing a unified interpretation. We also briefly discuss some other problems related to comparatives, which we consider to lie more or less outside of the main line of our research.

Both our solutions have been implemented within the framework of the SNACK-85 project, described in [Rayner & Banks 88b]. In Appendix 2 we give the complete program code for a "scaled-down" version of the system, which uses the second, "contrastive" solution to handle all the constructions described in the main body of the paper.

2. PREVIOUS WORK ON COMPARATIVES

Almost all authors are agreed on one point: the main problem in any theory which purports to give an explanation of the linguistic data is going to be the treatment of non-clausal comparatives1. The traditional viewpoint has been to explain these constructions by means of deletion rules, which in computational terms implies that the complete sentence will have to be reconstructed using syntactic methods; that is, rewriting rules will operate on the parse-tree to convert the elliptic form into a full clause. The first detailed account based on this idea was [Bresnan 73], which strongly influenced most work in the area during the following ten years.

Recently, however, other researchers have pointed out problems with Bresnan's approach; a very thorough and detailed criticism appears in [Pinkham 85]2, which has been our main theoretical source. Pinkham gives examples of a wide range of constructions which are difficult or impossible to explain in terms of deletion phenomena, and suggests instead an approach in which at least some comparative constructions are base-generated phrasal and then interpreted using a rule which she calls "distributive copying". The following example3 shows how the scheme works in practice. Sentence 1a) receives the logical form 1b):

1a) I invited more men than women
1b) I INVITED (MORE [q1 (q1 men), q2 (q2 women)])

---

1 An exception to this rule is Klein [80, 82, 83], who is primarily concerned with quite different problems, and only touches peripherally on those we will examine in this report.
2 Hereafter "Pinkham".
3 From Pinkham, p. 123
(The object of INVITED is the base generated phrasal). After distributive copying, this becomes 1c):

1c) MORE I q1 (INVITED q1 men), q2 (INVITED q2 women)]

This manoeuvre, replacing syntactic deletion rules with interpretative copying operations, seems to us very powerful, and (although we formulate it in a rather different way) is one of the central ideas in our own treatment of comparatives. We have in fact taken it even further than Pinkham, who keeps the verb deletion rule of "C-ellipsis" to explain some comparative constructions: in the account presented below in section 5, we get rid of the deletion rules completely and use only interpretative methods.

In this context, it is interesting to look at Levin's LFG-based work on sluicing constructions [Levin 82]. Levin presents a variety of arguments to support her claim that sluicing is not a c-structure phenomenon (i.e. not elliptic in nature), but rather explainable at f-structure level (i.e. in some sense related to a semantic copying operation). The differences between sluicing and comparative ellipsis are sufficiently great that this cannot in itself be said to prove anything, but it is none the less indicative of the way in which linguists are thinking about these problems.

In SNACK-85, which uses a framework based on that in [Pereira 83], we perform copying operations on "quant-trees", a level of structure which can loosely be compared with Chomskian logical form or LFG's f-structures. Viewed in this light, we claim that our treatment of non-clausal comparison (which at first glance might seem somewhat ad hoc) is in fact fairly well-related to current tendencies in theoretical linguistics.

3. CLAUSAL COMPARATIVES

Most authors are agreed that the case of clausal comparison is the simplest, and for this reason we tackle it first; despite this, it will be seen that there are a few tricky points. Our analysis is heavily based on Pinkham's and Klein's, and virtually amounts to an implementation of their work. In the interests of making the report self-contained, we start by summarizing what we see as the main ideas in the Pinkham/Klein treatment.

The fundamental notion in both Pinkham's and Klein's analyses is to assume that there is an implicit element present in a comparative clause, which is linked to the head of the comparison\(^1\) in a way similar to that in which a trace or gap is linked to its controller. This "trace" always contains a quantifier-like component. (We will adopt Pinkham's notation and symbolize this as Q). It may consist of just the Q on

---

\(^1\) We endeavour throughout this paper to keep our terminology as close as possible to that used by Pinkham, as summarized in the table in [Pinkham 85] and Appendix 1.
its own, or else be an implicit NP composed of the Q together with other material from the head of the comparison. (Parse-tree and logical form in example sentences 3a, b in appendix 3)

Pinkham argues that there are essentially three cases; these are exemplified in sentences 2a) - 2c). In the first of these, just the Q is extrapolated; in the second, a Q together with the CN\textsuperscript{1} books, taken from the head more books. If the head contains a comparative adjective, as in 2c), then the extra material, consisting of the adjective and the main noun from the head, is obligatory. For a justification, and an explanation of several apparent exceptions, we refer to Pinkham, p. 33 - 40.

2a) John bought more books than Mary bought (Q) records.
2b) John bought more books than Mary could carry (Q books).
2c) John bought a more expensive vase than Mary bought (a Q expensive vase).

A scheme of this kind can readily be implemented using any of the standard ways of handling traces. In our system, which is based on Extraposition Grammar [Pereira 83], we use the "extraposition list" to move the material from the head to the place in the comparative clause where it is going to appear; this corresponds to use of "slash categories" in a GPSG-like framework (Klein's solution), or to the employment of a HOLD register in an ATN.

Though this method appears to work well in practice, it gives rise to an interesting theoretical problem. If we implement the idea in its simplest form, using the same extraposition list (slash feature) both for "normal" and "comparative" traces, then it is possible to construct sentences containing crossing extrapositions. 3a) is an example of this (we write \( t_1, t_2 \) for the two traces).

3a) (Which pictures)\textsubscript{1} did John get more\textsubscript{2} for \( t_1 \) than Mary claimed he got \( t_2 \)?

We are not certain what attitude to adopt on this issue. The simplest solution would just be to rule such sentences inadmissible; certainly 3a) is very clumsy, and would normally be expressed as 3b), where the extrapositions no longer cross.

3b) (For which pictures)\textsubscript{1} did John get more\textsubscript{2} than Mary claimed he got \( t_2, t_1 \) ?

However, this does not appear satisfactory, given the presence of a sizable minority of speakers who judge sentences like 3a) as quite acceptable\textsuperscript{2}; if we are going to adopt a rigorous theoretical perspective, we must take the problem seriously. Whether it

---

\textsuperscript{1} By CN we mean CN in the "Montague" sense [Dowty et al 82 p. 183 - 190], i.e., a CN can be a noun with a modifier; an NP consists of a determiner and a CN.

\textsuperscript{2} To our amazement, a quick informal poll over ten of our colleagues actually discovered a speaker who preferred 3a) to 3b), though most of the others shared our opinion that the second was distinctly the better of the two.
would be better to do this by using separate extraposition lists for the two kinds of trace, or by allowing controlled crossing of extrapositions on a single list, is an issue which we regretfully postpone until a later paper; both approaches seem at least to have a \textit{prima facie} feasibility.

Another point of some interest arises from sentences like the well-known 4); the difficulty is obviously how to get the (normal) reading where the speaker expected the yacht to be longer than it actually turned out to be.

4) I thought your yacht was longer than it is.

Our suggested solution here is to treat the problem as a case of \textit{de dicto/de re} ambiguity. Thus following the analysis above, we have a basic compositional semantic analysis of 4) which corresponds to something like 4a):

4a) I thought your yacht was $Q$ long, $Q$ such that it is $Q'$ long, $Q'$ such that $Q > Q'$.

The different readings can now be produced by raising the scope of $Q'$, the normal one being obtained by moving it outside of the modal operator "I thought".

4. IMPLEMENTING A DELETION APPROACH TO NON-CLAUSAL COMPARISON\textsuperscript{2}

In this section, we will describe our first method, which is based on the conventional interpretation of comparatives: all comparatives are explicit or elliptic forms of clausal comparatives; making the analysis of comparison essentially a syntactic process.

This is done in three steps. The first is the recognition of a comparative clause which is triggered when a word marking comparison is parsed, e.g. \textit{than}, \textit{before} or \textit{as}. Secondly, the rest of the clause is parsed, and it is noted whether this is a complete sentence, adjective, prepositional phrase or nominal phrase. Thirdly, a modified copy of the clause containing the comparative marker is made using the comparand to replace some element in the comparative sentence (this is described below), and this copy is then annotated in the parse-tree as a modifier to the adjective or article. (This modified copy will be referred to as the \textbf{comp-s}).

An example of the kind of analysis we have in mind is shown in Diagram 2. The component marked \textbf{comp-s} is constructed from the main clause by replacing the agent \textit{John} with the "comparative object" \textit{me}. We use an LFG-like notation:

\begin{itemize}
  \item [1] Since writing this, we have discovered that this idea has recently been investigated in great detail by Richard Larson, in the context of Klein's framework [Larson 88].
  \item [2] For a more detailed description of the implementation, see [Banks 86]
\end{itemize}
Syntactic analysis of comparative

"John had this job before me"

s: verb: have
tense: imperfect
agent: "John"
obj: "this job"
time: mod-type: comparative
comp-s: verb: have
tense: imperfect
agent: "I"
obj: "this job"
time: mod-type: comparative-complement
comparison-type: before

Table 2

The simplest case is when the comparative object is a complete sentence, in which case this becomes the comp-s without having any changes made to it. If the comparative object is an adjective, as in

5a) John has more French books than English

the comp-s will consist of the comparand where the comparative object replaces the comparative adjective. If the comparative object is a prepositional phrase, it replaces that prepositional phrase in the comparand which has the same case as the comparative object. For example, in the sentence

5b) John saw more films by Woody Allen in London than in New York.

we want in New York to replace in London, (rather than by Woody Allen) as they are both locational.

The same idea is used if the comparative object is a nominal phrase, i.e., it replaces the object in the comparand which has the same case. The problem here is to decide which case the comparative object has. Consider, for example, the almost equivalent sentences

6) John has more books than Mary
7) John has more books than newspapers.

In 6) Mary obviously becomes the agent of the comp-s, analogous to I in the example above. In 7), however, it is equally obvious that newspapers should become the object of the comp-s. To distinguish between these two cases, the semantic properties of the comparative object must be considered. Our heuristic
here is to use the information stored in the semantic type to compare the objects in the original sentence to the comparative object. The case of the comparative object is then determined to be the same as that of the object of the original sentence whose semantic type is most similar to the semantic type of the comparative object.

As an example, we will show the analysis of sentence number 6). Firstly, having parsed the comparand, John has more books, we parse the comparison marker than. The second step is to parse a comparative object and remember what type of construction it is. We parse Mary and determine it to be a nominal phrase. Thirdly, we must decide the case of Mary in order to decide which object in the comparand sentence it should replace. To do this we compare the typelists of the agent John and the object books of the comparand with the typelist of Mary, and since the typelists of Mary and John are more similar than those of Mary and books, Mary is determined to have the same case as John, i.e., agent, in the comp-s. So the comp-s is constructed by making a copy of the comparand, and then replacing John with Mary to get Mary has X books.

5. CONTRASTIVE COMPARATIVE: A NON-TRANSFORMATIONAL ACCOUNT OF C-ELLPIS

5.1 Basic ideas

The syntactic solution described in the previous section is attractive, and seems to give a good explanation of the facts; indeed, many authors (e.g. [Bresnan73], [Klein 80]) appear to take for granted that any description of the comparative construction must be along these general lines.

More thought, however, shows that the exact formulation of the rule is somewhat problematic. We begin by noting the much remarked-on similarity between comparison and co-ordination. Now it has been known for some time that the analogous viewpoint for co-ordination is incorrect (see e.g. [Dowty et al 82, p. 271]); for example, although it is clear that 8a) and 8b) below are equivalent, it is equally clear that 9a) and 9b) are not.

8a) John and Mary were at the party.
8b) John was at the party and Mary was at the party.
9a) Everyone here is English or American.
9b) Everyone here is English or everyone here is American.

Given the above example, we can easily construct similar comparative sentences, such as 10a) and 10b). Here, if we assume a naive copying of the other parts of the main clause we would be forced to assume that 10a) had been derived from 10b). But this is actually a stronger statement, showing that the naive reconstruction is incorrect.

10a) Everyone spent more money in London than in New York
10b) Everyone spent more money in London than everyone spent in New York

The correct analysis is in fact 10c) which we (for reasons that will soon become apparent) rewrite as 10d)

10c) Everyone spent more money in London than they spent in New York
10d) Everyone has the property of being a P such that P spent an amount of money X in New York, where X is such that P spent an amount of money Y in London, where X > Y.

Either way, the "everyone" is not duplicated; this corresponds to the fact that spent more money in London than in New York is meaningful in itself, and is a property that can be predicated of people. So we have now altered our original standpoint: C-ellipsis works not on complete clauses, but on phrases. We can also get a new idea of the semantic structure of comparison by looking more closely at this example. Our suggestion is that 10d) be thought of as being built up out of the following components: the initial everyone, the contrasted elements in London and in New York, and the duplicated part, which could be rendered (roughly) as is a P such that P spent _ money _. The intention is that the first and second "gaps" in this last element should be filled respectively with a determiner and a PP. Clearly, there are going to be two "copies" of the duplicated part; in one of these, the PP slot will be filled by "in London", and in the other by "in New York". (Parse-tree and logical form in example sentences 4 a, b in appendix 3). What is not so obvious is the correct way to fill the determiner slots. Since the solution we are eventually going to propose may seem unintuitive, we attempt a step-by-step justification.

By analogy with the PTQ treatment of scoping phenomena1 ([Montague 72]; [Dowty et al 82], pp. 203-215), we suppose that there is a syntactic composition rule which allows us to replace a locative PP by a pronoun we call there; similarly, we allow a determiner to be another kind of pronoun, which we call so-many. The semantic rules will associate these pronouns with variables l_i, q_j, in the same way as Montague's he is associated with variables x_j. Then we define a syntactic rule2 for composition of phrasal comparatives like those in 10a) above, which takes as input the following elements:

i) A VP \( \Phi \) containing occurrences of there and so-many for some i and j.

ii) Two locative PPs \( \Lambda_1 \) and \( \Lambda_2 \).

iii) A comparative operator \( \Psi \) (more, less, as much etc.)

---

1Those familiar with Montague semantics will realize that the version we are using here is in several respects fairly different from that in PTQ. Most importantly, we more or less completely dispense with the mechanisms for dealing with intensionality; since this isn't our concern here, we feel that it would only confuse the issue. Readers who disapprove of these ideas should have no great difficulty in converting our derivations into more conventional ones.

2 Actually, Montague would call this a rule schema, parameterized by the two values i and j.
and produces as output the phrase \( \text{Comp}(\Phi, \Lambda_1, \Lambda_2, \Psi) \) constructed by performing the following operations:

a) Replace \textit{there} with the locative PP \( \Lambda_1 \).
b) Replace \textit{so-many} with the comparative operator \( \Psi \).
c) Concatenate than/as + \( \Lambda_2 \) to the end.

We assume that the associated type for a VP is \(<e,t>\), and let \( \Phi' \) be the denotation of \( \Phi \), \( \Psi' \) the denotation of \( \Psi \), etc. Since \( \Phi \) contains instances of the pronouns \textit{there} and \textit{so-many}, \( \Phi' \) will contain corresponding free variables \( i_1 \) and \( q_1 \). It will thus make sense to abstract over these: for convenience's sake, we will refer to \( \lambda i_1 \lambda q_1 \Phi' \) as \( \Phi'' \). The associated semantic rule will then define the denotation of \( \text{Comp}(\Phi, \Lambda_1, \Lambda_2, \Psi) \) to be the value of the expression

\[
\lambda x: \Phi''(\Lambda_1')[(N, N \text{ st } \Phi''(\Lambda_2')(N', N' \text{ st } \Psi''(N,N'))(x)](x)
\]

This formula is both central to the paper, and sufficiently obscure as to require some further explanation. To understand what it means in intuitive terms, we focus first on \( \Phi'' \). This can be viewed in the usual way as a predicate with three argument slots, which are respectively to be filled by

i) a locative PP denotation (\( \Lambda_1' \) for \( i = 1 \) or 2)
ii) a determiner denotation
iii) a variable, representing the abstracted contribution from the subject.

We now work "from the inside out"; thus we begin with the subformula \( (N', N' \text{ st } \Psi''(N,N')) \). This is intended to be a \textit{generalized determiner}, in the sense of [Barwise & Cooper 81]. For example, if \( \Psi \) has the value \textit{more}, it will be the determiner \( (N', N' \text{ st } \text{more}(N,N')) \), or to express this in more common terms (less than \( N \)). Working outward to the next level, the subformula \( \Phi''(\Lambda_2')(N', N' \text{ st } \Psi''(N,N'))(x) \) is that formed by substituting the locative denotation \( \Lambda_2' \) in the first argument place of \( \Phi'' \), the determiner denotation \( (N', N' \text{ st } \Psi''(N,N')) \) in the second, and the abstracted variable \( x \) in the third. It is thus a condition on \( N \), which means that \( (N, N \text{ st } \Phi''(\Lambda_2')(N', N' \text{ st } \Psi''(N,N'))(x)) \) is also a generalized determiner. The final formula is now constructed by once again substituting suitable values into the argument slots for \( \Phi'' \), (these now being \( \Lambda_1' \), \( (N, N \text{ st } \Phi''(\Lambda_2')(N', N' \text{ st } \Psi''(N,N'))(x)) \), and \( x \)), and then abstracting over the last of these.
To summarize: what we have described here is a rule which allows us to construct what could be called a comparativeVP, which contrasts two PP's. By making suitable changes we could just as easily have contrasted other constituents. Later in the paper, we will show in detail how this works, but it should be clear that the only thing that essentially needs to be done is to replace the locative proforms therei with other proforms ranging over the domain we intend to "contrast on", and make corresponding alterations in the type definitions for Λt. Similarly, we can construct comparative versions of other constituents than VP's, as long as they have associated types of a form in the series t, <e,t>, <e,<e,t>>, etc: this restriction is necessary so as to be able to substitute in the variables needed to reduce the constituent to a truth-value when building the outer determiner. (In practice, of course, we will primarily be interested in the types t and <e,t>, corresponding to clauses and VP's). Thus what we have seen is really an instance of a general schema for comparative interpretation rules.

After this explanation, it should not be too difficult to understand the following representation of a Montagovian derivation of a semantic form for sentence 10a) above:
Montagovian analysis of comparative

(spent(x,y,z) is to be read as "x spent y in the city z")

1. everyone  \( \lambda Q \forall x \) person(x) → Q(x)
2. New York  \( \lambda Q \exists z [z=\text{New York} \land Q(z)] \)
3. London  \( \lambda Q \exists z [z=\text{London} \land Q(z)] \)
4. spent  \( \lambda z \lambda y \lambda x \) spent(x,y,z)
5. spent so-many1 dollars there1  \( \lambda x : l_1(\lambda z : q_1(\lambda y : \text{dollar}(y)), \lambda y : \text{spent}(x,y,z)) \)

6. spent more dollars in London than in New York

\( \lambda x : \exists z : [z=\text{London}] \land \\
(N, N \text{ st} \\
\exists z': [z'=\text{New York}] \land \\
(N', N' \text{ st} \text{more}(N,N'))(\lambda y' : \text{dollar}(y'), \lambda y' : \text{spent}(x,y',z'))) \)

7. everyone spent more dollars in London than in New York

\( \forall x : \text{person}(x) \rightarrow \\
\exists z : [z=\text{London}] \land \\
(N, N \text{ st} \\
\exists z' : [z'=\text{New York}] \land \\
(N', N' \text{ st} \text{more}(N,N'))(\lambda y' : \text{dollar}(y'), \lambda y' : \text{spent}(x,y',z'))) \)

\( (\lambda y : \text{dollar}(y), \lambda y : \text{spent}(x,y,z)) \)

---

The analysis we have just presented solves the problems that arose in connection with the approach of section 4, but it is still not easily implementable as it stands. The reasons for this are exactly the same as those applying to the original PTQ treatment of scoping phenomena; essentially, the rules have insufficient syntactic motivation, making it difficult to construct a sensible parser. However, the solution (as far as scoping goes) is by now standard, and has been successfully implemented in a number of well-known systems, e.g. [Pereira 83], [Lesmo 85], [Grosz et. al. 86]).

---

1Remember that since \( q_1 \) is a variable ranging over determiner denotations it takes two arguments, which should both be of the type \(<e,t>\). Similarly, \( l_1 \) (an NP denotation) takes a single argument of the same type.
the methodology can originally be traced back to [Woods 78]. The trick is to split interpretation into two stages, the first linked to the syntax and the second to the semantics; mediating between these, we have an intermediate level of representation, which we call (following [Pereira 83]) a quant-tree. This is produced compositionally from the syntax, and then subjected to rewriting rules before being converted into the final logical form. Normally, these rewriting rules formalize scoping transformations; here, we will also use them to describe the interpretation of non-clausal comparison.

As our starting-point, we will use Pereira's formulation of the quant-tree level of structure. Here, there are three types of node: quants (representing generalized quantifiers), preds (representing clauses), and conjs (representing conjunctions of clauses). Rewriting can lift quant nodes higher in the tree, changing the scoping relationships.

We will use a modified version of Pereira's system; the exact reasons for our changes will become apparent in the course of the following discussion. First, we give the quant node a hierarchic structure: it now contains two components, the first corresponding to the NP's thematic role, and the second to its associated generalized quantifier. The generalized quantifier substructure is in turn composed of two parts, the first corresponding to the denotation of the determiner, and the second to that of the "CN" part of the NP. Thus to sum up, we have the following structure:

role_item (role-filler denotation; Pereira's quant)
  thematic role (denotation of preposition/argument position)
  quant (generalized quantifier; NP denotation)
    det (determiner denotation)
    cn_item (CN denotation)

Secondly, we introduce five new kinds of nodes beside the ones already named: we call these comparands, comparative-objects, comparisons, and Q- and C-placeholders. The various types of nodes interact as follows. (See diagram 1).

(Stage 1) At the syntactic level, we view the comparative object as a constituent in its associated comparative AP; when the parse-tree is transformed into the quant-tree, the AP gets turned into a comparand node, in which there is a comparative-object subnode representing the comparative object.

---

1 There is some disagreement between applied and theoretical linguists concerning the correct division of credit here. Although Woods and his colleagues were the first people to realize the idea in practice, they made no attempt to justify their work in formal terms; this was later achieved independently by Cooper [83], working within the Montague grammar school. Our work has to some extent been influenced by both these sources.

2 See [Pereira 83], p. 82. Pereira's system does not handle conjunctions of NP's which appear to need a fourth kind of node.
(Stage 2) Rewriting rules then move the comparative-object out of the comparand, leaving behind a Q- (for Quantifier-) placeholder. This is a pair consisting of the compared predicate (the adjective, adverb or whatever), and the logical variable (the "linking" variable), which corresponds to the lambda-bound variable Q in the Montagovian account above (p. 13).

(Stage 3) The "raised" comparative-object node is a 3-tuple. It consists of

- The variable Q (and is thus "linked" to the placeholder through it - hence the name),
- The comparison type (more than, less than, same as etc.)
- The subnode which represents the comparand NP or PP. We will call this the compared material.

The rewriting rules search the portion of the quant-tree under the comparative-object node until they find something compatible that it can be contrasted against; if nothing is found, the comparand node is moved "upwards" until a suitable such element is located. In the program code shown in appendix 3, the compatibility requirements are very simplistic, and do not amount to much more than demanding that contrasting be between elements of similar semantic type, and that only certain prepositions (e.g. for/against; in/on) may be contrasted. This could be improved upon; an obvious way would be to apply the ideas from section 4 above, and define preference heuristics which gave higher priority to comparisons between quant nodes whose variables are of similar type.

(Stage 4) The part of the node that is being contrasted against (in the simplest case, the whole of it) is replaced by another "placeholder" structure. We call this a C-placeholder ("contrast placeholder"), to distinguish it from the above-mentioned Q-placeholder, and we will refer to the thing it replaces as the contrasted material. An extra node, which we call a comparison, is inserted above the "compared" one; the comparison is a 5-tuple consisting of

- The comparison type.
- The "linking variable" Q, which functions as a pointer to the Q-placeholder.
- The compared material.
- The contrasted material.
- A pointer to the C-placeholder.

When the quant-tree is converted into logical form, there should thus be only comparison nodes and placeholder nodes left, with the placeholders "below" the comparisons. In the final stage, the portion of the quant-tree under the comparison node is duplicated twice, and the linking variables instantiated in each copy in the manner described above. So in the "inner" copy, the C-placeholder is instantiated to the compared material, and the Q-placeholder to a form \( \lambda N:\text{comp}(N,N') \) where \text{comp} is the type of comparison and \( N \) and \( N' \) are the degree variables in the inner
and outer determiners; in the "outer" copy, the C-placeholder is instantiated to the
contrasted material, and the Q-placeholder to the value of the inner form. The
above description should be compared with Table 4 and the program code in
Appendix 2.

The example sentence in the following diagram differs slightly from the one we
have been discussing, in that we have replaced dollars with money, and let more
modify the verb spend, instead of the noun dollars. The reason for this is simply to
save space - the box in the top right hand corner would otherwise become
enormous - and it doesn’t affect the way the sentence is treated.
Original quant tree

verb: spend

The comparand node is split, and the resulting comparative object node is moved up

verb: spend

Final quant tree. The compared material (the role_item node) in the comparative object node has been contrasted with the role_item node immediately beneath it, which has consequently been replaced by a C-placeholder.

INNER FORM = ex Y: [Y = London] & ex A1: more(A1,A) & ex S1: spend(S1) & agnt(S1,X) & loc(S1,Y) & amount(S1,A1)

OUTER FORM = ex Z: [Z = New York] & ex A: INNER FORM & ex S: spend(S) & agnt(S,X) & loc(S,Z) & amount(S,A)

TOTAL FORM = all X: person(X) implies OUTER FORM

Diagram 1
In the following subsections we will show, as promised, how different instantiations of the schema described in the beginning of this section can be used to assign a correct semantics to several other kinds of comparative construction, without any recourse to C-ellipsis.

5.2. COMPARATIVES WITH "S-OPERATORS"

In this section, we are going to examine comparative constructions like those in 11a), 11b) and 11c), which have a long and honourable history in the semantics literature:

11a) Mary had more friends than John had expected.
11b) Most people paid more than Mary said.
11c) John's yacht was longer than I thought.

In order to handle examples like these within our framework, we need a syntactic representation which does not involve ellipsis. Our solution is to introduce a syntactic constituent which we call an "s-operator": we define this implicitly by saying that an "s-operator" and a sentential complement combine to form a clause.\footnote{In a categorial grammar framework like HPSG [Pollard & Sag 88], we could simply identify an s-operator with a constituent of the form S/S-COMP. It is fairly straightforward to define s-operators in XG-grammar.} Thus the italicized portions of the sentences above are deemed to be s-operators, and in each of them the s-operator's missing complement is viewed as a kind of null pronoun. (Parse-tree and logical form in example sentences 5a, b in appendix 3).

Although this move may in English seem syntactically quite unmotivated, there are other languages where evidence can be found to support the claim that these pronouns really exist. In Russian, where comparative constructions very closely follow the English and Swedish patterns, they can optionally appear in the surface structure as the pronounesto. The following sentence illustrates this.

Он купил больше книг чем я это думал.
He bought more books than I esto thought.

Semantically, the analysis of such sentences is exactly parallel to that in the preceding subsection. Comparing 11b) with 10d), the "initial part" is most people, and the "contrasted elements" are the s-operator Mary said and an implicit trivial s-operator which we can write as (it is true that). The "duplicated part" is the predicate is a P such that P paid amount of money. Introducing a set of "proforms" attitude\textsuperscript{i} (with associated variables aj) ranging over S-operators, together with a suitable syntactic rule which permits their composition with an S-

\footnote{This was pointed out to us by Ilya Beder.}
complement\(^1\), we can sketch a Montagovian analysis similar to that in Table 4 and Diagram 1:

**Montagovian analysis of s-operator comparative**

\((\text{paid}(x,y))\) is to be read as "\(x\) paid \(y\) amount of money"

1. most people
   \(\lambda Q: \text{most}(\lambda x: \text{person}(x), Q)\)
2. Mary said
   \(\lambda Q: \text{said}(\text{Mary}, Q)\)
3. (it is true that)
   \(\lambda Q: Q\)
4. paid
   \(\lambda y \lambda x: \text{paid}(x,y)\)
5. attitude\(_1\) paid so-many\(_1\) dollars
   \(\lambda x: a_1(q_1(\lambda y: \text{dollar}(y), \lambda y: \text{paid}(x,y)))\)
6. (it is true that) paid more dollars than Mary said
   \(\lambda x: \exists y \text{paid}(x,y) \wedge \exists y' \text{said}(\text{Mary}, \text{paid}(x,y') \wedge \text{more}(y,y'))\)
7. most people paid more dollars than Mary said
   \(\text{most}(\lambda x: \text{person}(x), \lambda x: \exists y \text{paid}(x,y) \wedge \exists y' \text{said}(\text{Mary}, \text{paid}(x,y') \wedge \text{more}(y,y')))\)

*Table 4*

The implementation of this analysis in terms of quant-tree rewriting rules involves only a slight extension of the method described in section 5.1 above. The reader is referred to Appendix 3 for the concrete details.

### 5.3. "PARALLEL" PHRASAL COMPARATIVES

Comparative constructions of the type illustrated in 12a) have been the object of considerable controversy. The orthodox position was that they were "parallel" constructions: 12a) would thus be a reduced form of 12b).

12a) More women than men read "House and Garden".
12b) More women read "House and Garden" than men read "House and Garden".

Pinkham, however, gives good reasons for supposing that this is not the case, and that the construction is in some sense base generated phrasal (p.121-123). It will presumably not come as a revelation to hear that we agree with this idea, which it turns out can easily be accommodated in the theory we have developed above. More specifically, we allow a comparative complement of the form *than/as CN*, and then

---

\(^1\) Making this work correctly is non-trivial, since we have to be able to combine phrases of category A/B and B/C to yield phrases in A/C, with parallel semantic operations. Ideas of this kind have been explored in work by Ades and Steedman [81].
apply the analysis from section 5.1, contrasting the CN from the complement against that from the NP containing the comparative determiner. Exactly how this works out in practice is illustrated in table 5 and diagram 2 below, and in example sentences 6 a,b in appendix 3; as usual, we need a new set of pronouns, this time ranging over CN’s. Lacking a convenient word like the German solch, we name these of-that-kind, with corresponding variables kj.

Montagovian analysis of "parallel" phrasal comparative

(reads(x,y) is to be read as "x habitually reads y")

1. woman                  \[\lambda w: \text{woman}(w)\]
2. man                    \[\lambda z: \text{man}(z)\]
3. "House and Garden"     \[\lambda x: x = "H & G"\]
4. read "House and Garden" \[\lambda x: \text{read}(x,y) \land y = "H & G"\]
5. so-many₁ of-that-kind₁ read "House and Garden" \[q₁(k₁,\lambda x: \text{read}(x,y) \land y = "H & G")\]
6. more women than men read "House and Garden".
   \[[N, N st \text{[N’, N’ st N > N’]}]
    (\lambda z: \text{man}(z),[\lambda x: \text{read}(x,y) \land y = "H & G")])
    (\lambda w: \text{woman}(w),[\lambda x: \text{read}(x,y) \land y = "H & G")]])\]

Table 5

It is interesting to compare our treatment with that suggested in [Keenan & Stavi 86] (p.282-284) for comparative adjectival constructions like that in 13a); they argue convincingly that these are to be regarded as directly interpreted, rather than as "reduced forms" of sentences like 13b). It seems to us that their arguments can be adapted to support the analysis of "parallel" phrasals given above; so if we were to extend their example, we would have that 13b) in its turn was also to be interpreted directly, rather than considered a reduction of 13c).

13a) More male than female students passed the exam.
13b) More male students than female students passed the exam.
13c) More male students passed the exam than female students passed the exam.
Original quant tree. Note that the item corresponding to the object (House & Garden) has been scoped highest.

The comparand node is split, and the resulting comparative object node is moved up.

Final quant tree. This time, the compared material is a CN denotation and thus requires another CN denotation to be contrasted against. This is extracted from the agent node, as usual leaving behind a C-placeholder.

Construction of the logical form. \text{number}(X,N,P,Q)\text{ is to be read as: There are } N\text{ }X\text{'s satisfying } P\text{ which also satisfy } Q.\text{ }

\text{INNER FORM} = \text{number}(Z, N, L, \text{man}(Z))\text{, ex } R:\text{ read}(R1)\&\text{ agent}(R1,Z)\&\text{ object}(R1,Y))\&\text{ more}(N,N1)\text{, }

\text{OUTER FORM} = \text{number}(X, N, \text{woman}(X), \text{ex } R:\text{ read}(R)\&\text{ agent}(R,X)\&\text{ object}(R,Y))\&\text{ inner form}\text{, }

\text{TOTAL FOR M = ex } Y: [Y = \text{House }& \text{ Garden}] \& \text{ OUTER FORM}
5.4 OTHER CONTRASTIVE CONSTRUCTIONS

The three types of contrastive comparatives described above appear to account for
the vast majority of all constructions which occur in practice\(^1\). There are also a few
rare cases, two of which we here touch on briefly.

5.4.1 VP-OPERATORS

Analogously with the "S-operators" introduced in section 5.2 above, we define a
"VP-operator" to be a constituent of the syntactic category S/VP, i.e. something
which can combine with a VP complement to yield an S. This allows us to give a
similar account for sentences like those illustrated in 14a) to c):

14a) He scored more points than I usually can.
14b) He needs to get a better time than he's so far managed.
14c) He hit her harder than he'd meant to.

As usual, the Montagovian version will involve the introduction of a class of
pronouns which range over VP-operators, together with a combination rule for
phrasal comparative which gives them a complement of the form \(as/\)than + VP-
operator. Readers who wish to understand the implementation details for the
quant-tree version are referred to the relevant program code in Appendix 2. Parse-
tree and logical forms are given in example sentences 9a, b in Appendix 3.

5.4.2 CONTRASTED PREPOSITIONS

As a final example, we consider the case of contrasted prepositions, as shown in 15)
below\(^2\).

15) There are more points for the proposal than against.

These can be treated in a completely straightforward manner in the quant-tree
version; the key step will be the replacement of the thematic-role field of a suitable
NP denotation with a C-placeholder. (It should now be clear that we have
intentionally defined the structure for the NP denotation in such a way as to make
this manoeuvre possible). A Montagovian treatment of the construction is of course
somewhat problematic with the framework we have so far adopted, but a suitable
reworking along the lines recently popularized by linguists working within the
LFG/Situation semantics school would probably make it possible to solve this
problem (see eg. [Fenstad et al. 87], [Hirst 87]). (Parse-tree and logical form in
example sentences 11a, b in appendix 3).

---

\(^1\) Excluding "discourse comparatives": see sections 8 and 9.
\(^2\) We would like to thank Carol Friedmann for bringing this example to our attention.
5.5 AMBIGUITIES IN CONTRASTIVE CONSTRUCTIONS

The system we have been describing affords an elegant explanation of the ambiguities that can arise in connection with constructions of the type we have examined. Let us, for example, consider the sentences 16) and 17) below:

16) Nobody talked more to John about Carol than Mary.
17) Mary plans to have more children than Carol.

We claim that most speakers find both sentences ambiguous, the first in three ways and the second in two. We look first at 16), where we can paraphrase the various readings as 16a,b,c):

16a) Nobody talked more to John about Carol than Mary did. (*Mary talks a lot to John about Carol.*)
16b) Nobody talked more to John than to Mary about Carol. (*John is more popular than Mary when people want to hold conversations about Carol.*)
16c) Nobody talked more to John about Carol than they did about Mary. (*Everyone prefers discussing Carol to discussing Mary when they talk with John.*)

In terms of the ideas we have developed, ambiguity arises naturally as a result of the uncertainty concerning the identity of the element which "Mary" is intended to be compared with. By successively choosing the NP's "Nobody", "John" or "Carol", the reader can easily satisfy herself that each of the readings above can be derived.

Similar, though slightly more complex, considerations apply to sentence 17), where we represent the two readings as 17a) and 17b):

17a) Mary plans to have more children than Carol plans to have. (*Mary and Carol both plan to have specific numbers of children, that number being in Mary's case greater.*)
17b) Mary plans to have more children than Carol will have. (*Mary's plan is that, no matter how many children Carol has, she will have more.*)

To explain the divergence, we make the fairly uncontroversial hypothesis that the quant-tree contains an extra node which represents the virtual subject of "have more children". "Carol" can then be compared either with this, giving 17a), or with the real subject "Mary", giving 17b).

6. "SIMPLE" PHRASAL COMPARATIVES

We finally turn our attention to a third type of comparative construction, which does not properly seem to fit into any of the patterns given above. We start by giving in 18) - 20) some examples of the kind of sentence we have in mind.

18) Mary seems brighter than most pupils.
19) He ran faster than the world record.¹
20) John needs a bigger² spanner than the No. 4.

Pinkham uses constructions like these as her key examples when demonstrating the existence of base-generated phrasal comparatives. Looking for instance, at 20), we claim with Pinkham that the most natural solution is to treat bigger spanner than the No. 4 as a closed constituent with a semantic interpretation which does not involve the rest of the sentence. (Parse-tree and logical form in example sentences 7a, b in appendix 3).

It may not be obvious at first why this should be so, and we pause briefly to examine the possible alternatives. Firstly, suppose that we tried to use a reduction/predicate copying account. This would make 20) a form of 20a):

20a) John needs a (big to extent X) spanner, X such that John needs the (big to extent Y) No. 4. spanner, X > Y.

implying that John needs the No. 4. This is clearly wrong; the "needs" isn't copied in any way, and in fact the scope of any copying operation must be limited to the phrase bigger spanner than the No. 4. If we are absolutely bent on using copying, it appears to us that the only way in which it can be done is to treat 20) as derived from 20c) through 20b)

20b) John needs a spanner which is bigger than the No. 4.
20c) John needs a spanner which is (big to extent X), X such that the No. 4 is (big to extent Y), X > Y.

To be honest, we can't completely discount this approach. However, since it makes bigger than the No. 4 into a constituent in the intermediate 20b), we think it simpler to interpret the phrase structure directly, as is illustrated in the following Montagovian analysis.

¹Pinkham's example 124a, p. 136
² We will treat "bigger" as though it were actually "more big" for the usual reasons
Montagovian analysis of "simple" phrasal comparative

(needs(x,y) to be read as "x needs something of which the predicate y holds")

1. John                   \(\lambda x: x = \text{John}\)
2. needs                  \(\lambda x, y: \text{needs}(x, y)\)
3. No. 4                  \(\lambda x: \text{type_of}(x, \text{No. 4})\)
4. big                    \(\lambda x, y: \text{big}(x, y)\)
5. spanner                \(\lambda x: \text{spanner}(x)\)
6. the                    \(\lambda P \lambda Q: \text{the}(P, Q)\)
7. more                   \(\lambda P \lambda Q \lambda R: (\lambda x: \exists y: P(x, y) \land R(Q, \lambda z: \exists y': P(z, y') \land \text{more}(y, y')))\)
8. more big than the No. 4 \(\lambda x: \exists y: \text{big}(x, y) \land \text{the}(\lambda z: \text{type_of}(x, \text{No. 4}), \lambda z: \exists y': \text{big}(z, y') \land \text{more}(y, y')))\)
9. a bigger spanner than the No. 4 \(\lambda x: \text{spanner}(x) \land \exists y: \text{big}(x, y) \land \text{the}(\lambda z: \text{type_of}(x, \text{No. 4}), \lambda z: \exists y': \text{big}(z, y') \land \text{more}(y, y')))\)
10. John needs a bigger spanner than the No. 4 \(\text{needs}(\text{John}, \lambda x: \text{spanner}(x) \land \exists y: \text{big}(x, y) \land \text{the}(\lambda z: \text{type_of}(x, \text{No. 4}), \lambda z: \exists y': \text{big}(z, y') \land \text{more}(y, y')))\)

Table 6

It will be apparent that bigger than the No. 4 turns up as a constituent here too, and thus our solution is in a sense equivalent with the alternate one proposed above. This is a striking illustration of the difficulties that can attend any efforts to make rigorous comparisons between different syntactic-semantic analyses of natural-language constructions.

7. OTHER ISSUES

This concludes our analysis, with the exception of some minor points which we have postponed till now in the interests of expositional clarity. The first concerns non-coreference of compared objects, the second the question of whether "before" and "after" can be regarded as comparatives.
7.1. NON-COREFERENCE OF COMPARED OBJECTS

Although our tests show that speaker judgments are not completely unanimous, there is a strong tendency to assume that comparison is always between different objects. This point becomes important when answering questions like 21).

21) Has any king ruled as long as Gustav V?

Most people find "Yes" meaning "Yes, Gustav V did" extremely misleading. To correct this, we make a slight adjustment in the analysis we have so far been using, so that an appropriate inequality is added when the logical form is produced from the reshaped quant tree. With this alteration 22) may be judged true even if John is present, and 23) may correctly be said of one of Mary's articles.

22) Nobody here has read as many books as John.
23) Mary has never written an article that is as bad as this one.

7.2. "BEFORE" AND "AFTER"

In English, the words before and after can display several of the features associated with comparatives. For example, sentences like 24a) and 25a) would appear to have interpretations which could be represented as 24b) and 25b):

24a) John arrived before me.
24b) John arrived at a time T, T such that I arrived at a time T', T before T'.

25a) Henry VIII married Anne Boleyn after Catherine of Aragon.
25b) Henry VIII married Anne Boleyn at a time T, T such that he married Catherine of Aragon at a time T', T after T'.

Given examples like these, it tempting to conclude that before and after are the "comparative forms" of adverbials which could be represented as "at-early-time" and "at-late-time". (Indeed, we made exactly this proposal in an earlier paper). However, it must be noted that, even if this analysis is correct, these adverbials would appear to be subject to certain restrictions with regard to the comparative complements they can take. For example, examples 26a) - 27a) are at best dubious, and should be contrasted with the correct 26b) - 27b):

26a) *John arrived before I expected.
26b) John arrived earlier than I expected.

27a) ?Mary left before she needed to.
27b) Mary left earlier than she needed to.

It thus seems uncertain whether these words should be regarded as a special sort of comparative, or as belonging to a separate class of their own. The best way to
resolve this question might perhaps be to consider cross-linguistic data; if it turns out that there are other languages which allow a full range of comparative complements to the analogous words, it would presumably make sense to hypothesize that this was originally the case for English, and that the missing constructions have simply fallen into disuse. Lacking at the moment any such evidence, we refrain from further speculation on this subject.

8. DATA FROM CORPUS ANALYSES

In order to give our claims concerning relative frequencies of the various constructions some substance, we here give the results of a small corpus analysis. We took the texts of one English and one Swedish novel (Agatha Christie's *And then there were none*; Selma Lagerlöf's *Kejsaren av Portugalien*), and manually extracted all sentences containing comparatives. The results are summarized in table 7.
<table>
<thead>
<tr>
<th>Type of construction</th>
<th>No. of examples</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Discourse comparative:</td>
<td>37</td>
<td>33.9%</td>
</tr>
<tr>
<td>Contrastive:</td>
<td>30</td>
<td>27.5%</td>
</tr>
<tr>
<td>Simple</td>
<td>7</td>
<td>6.4%</td>
</tr>
<tr>
<td>2. Clausal:</td>
<td>7</td>
<td>6.4%</td>
</tr>
<tr>
<td>3. Contrastive:</td>
<td>34</td>
<td>31.2%</td>
</tr>
<tr>
<td>NP:</td>
<td>12</td>
<td>11.1%</td>
</tr>
<tr>
<td>NP + &quot;anaphoric verb&quot; (^1):</td>
<td>4</td>
<td>3.7%</td>
</tr>
<tr>
<td>PP/Adverbial:</td>
<td>10</td>
<td>9.1%</td>
</tr>
<tr>
<td>S-operator:</td>
<td>5</td>
<td>4.6%</td>
</tr>
<tr>
<td>VP-operator:</td>
<td>2</td>
<td>1.8%</td>
</tr>
<tr>
<td>Other:</td>
<td>1</td>
<td>0.9%</td>
</tr>
<tr>
<td>4. Simple:</td>
<td>11</td>
<td>10.1%</td>
</tr>
<tr>
<td>5. Determiner/Phrasal(^2):</td>
<td>6</td>
<td>5.5%</td>
</tr>
<tr>
<td>6. Verb of change + comparative(^3)</td>
<td>6</td>
<td>5.5%</td>
</tr>
<tr>
<td>7. Inte comparative än att...(^4)</td>
<td>4</td>
<td>3.7%</td>
</tr>
<tr>
<td>8. Other:</td>
<td>4</td>
<td>3.7%</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>109</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 7

One striking fact is immediately apparent; the commonest type of construction (No. 1, "discourse comparative") is actually one that we fail to cover! In our defense, however, we can at any rate claim that none of the other authors we have quoted appear to do so either. By *discourse comparatives*, we mean here comparatives where the associated comparative clause (in English normally introduced by "than" or "as") is completely absent, and must be inferred from the context. Typical examples (taken from the Agatha Christie novel) follow below in sentences 28) - 30); we will discuss these briefly in the next section.

28) The abandoned creature ... committed a still graver sin. (p. 73)
29) At eight o'clock the wind was blowing more strongly. (p.118)
30) I should be colder if I were dead. (p.168)

---

\(^1\) By this we mean verbs like "do" or "be", as in "bought more books than John did" or "is involved in more deals than Mary is". We view elements like these as making no contribution beyond that of forcing the associated NP to be contrasted against a preceding subject.

\(^2\) Constructions like "more than five", or "at least ten". These are well-known in the literature (Pinkham treats them in considerable detail), and easily implementable.

\(^3\) Constructions involving verbs like "grow" or "become" together with a comparative, as in "The room grew more cheerful" or "The road became harder". The problems involved in giving a correct semantics to such sentences derive in our opinion primarily from the verbs, and not from the comparatives.

\(^4\) A construction peculiar to Swedish, as in *Det var inte tynge än att han kunde lyfta det.* (Lit: "It was not heavier than that he could carry it").
If the discourse comparatives are excluded, however, we are clearly capable of dealing with the vast majority of those left. Thus we feel fairly well-justified in claiming that our treatment covers most of the common cases of comparison which occur in practice.

9. CONCLUSIONS AND FURTHER DIRECTIONS

We have presented a method for syntactic and semantic interpretation of comparative sentences. This has been done by dividing our material into three different groups, each of which are treated differently:

- Clausal comparatives (section 4), which are handled by extraposing a constituent containing a Q, following Pinkham's theoretical analysis.

- Contrastive comparatives (section 5), treated by direct interpretation using "predicate copying".

- "Simple" phrasals (section 6), handled by a different direct interpretation method.

We do not claim that this classification is the only way to explain the facts; as we have said above, it would be possible to rewrite simple phrasal comparatives into contrastive comparatives, and also to rewrite contrastive comparatives as clausal comparatives. We think, however, that this manoeuvre would give us nothing in the form of real gains; even though a unified solution might seem theoretically more elegant, the syntactic transformations needed are more complicated than the use of different categories. Thus our first argument against a unified approach is the practical one: we need do less work as implementors if we adopt the classification described here. Despite this, we suspect that many readers (especially those more theoretically than practically inclined) would find it comforting to have some direct evidence that supports our point of view. In this connection we think that we can point to at least three strongly suggestive cases: extraction from contrastive NP complements in English, simple comparatives in Russian, and the distribution of the particle vad in Swedish. We now examine each of these in order.

The first point is well-known, and was first described in detail in [Hankamer 71]: in English (and in fact a good many other languages), it is possible to extract out of contrastive NP complements, but not from clausal comparative complements. For example, compare 31a) with 31b):

31a) Who does John run faster than ti?
31b) *Who does John run faster than ti runs?

The relevance of this observation is obvious: if contrastive NP complements really are elliptic foms of clausal complements, we would appear to need some
mechanism by which something originally an extraction island could cease to be one when a reduction is performed. This would be a very unusual phenomenon, which it would certainly be preferable to avoid introducing into a syntactic theory; rather than do this, it seems much simpler to treat the two constructions as fundamentally different, as we have done.

Secondly, we consider the case of "simple" comparatives in Russian. A sentence like 32a) can be translated into Russian in two distinct ways, illustrated in 32b) and 32c):

32a) Ivan is bigger than me.
32b) Иван больше чем я.
   Ivan-NOM (is) bigger than I-NOM.
32c) Иван больше меня.
   Ivan-NOM (is) bigger I-GEN.
32d) Иван купил больше дому меня.
   Ivan-NOM bought (a) bigger house-ACC I-GEN.

What is interesting is that, while 32b) has the same syntactic structure as the English version, 32c) lacks the comparative marker (чём = "than") and puts the comparative complement into the genitive. Moreover, this extra possibility is only available with the verb "to be"; thus for example a sentence like 32d) would be ungrammatical. These apparently anomalous facts can be neatly explained if we hypothesize that the contrastive and "simple" comparative constructions are formally distinguished in Russian, 32b) being contrastive and 32c) simple.

Thirdly, we look at Swedish, where comparative constructions are on the whole virtually identical to the corresponding ones in English. One significant difference, however, is the distribution of the relative pronoun vad ("what"); this can optionally be inserted after the comparative marker in some constructions, as shown in 33) and 34).

33) Johan köpte fler böcker än (vad) Maria gjorde.
    John bought more books than (what) Mary did.

34) Johan har ett dyrare hus än (vad) jag har.
    John has a more expensive house than (what) I have.

Given the correspondences between clausal comparison and relative clauses described in section 3, it is very tempting to account for the "vad" as a relative

---

1The data in this paragraph is due to Mats Carlsson.
pronoun realizing the normally null Q. If we are prepared to accept this, it then appears significant that "vad" may not be used in most phrasal comparatives, as shown in 35) and 36). This would seem problematic for a transformational account, but is quite natural if phrasal comparatives are treated as contrastive; there isn't any Q, so it can't be realized as a "vad".

35) Johan köpte fler böcker än (*vad) Maria.
   John bought more books than (*what) Mary.

36) Fler kvinnor än (*vad) män läser "Hänt i Veckan".
   More women than (*what) men read "News of the World".

There is, however, one exception to the rule: "vad" may appear in the "s-operator" constructions from section 5.1 above, as shown in 37).

37) Johan köpte fler böcker än (vad) Maria trodde.
   John bought more books than (what) Mary thought.

We are not certain how to explain this, and leave the reader to judge the facts for himself; but despite this irregularity, we think the other data gives our theory a considerable amount of concrete backing.

Looking ahead, our next goal is to extend the theory to cover Discourse Comparatives, mentioned in section 8, which turned out to be by far the most frequent construction. A cursory examination of the example sentences would suggest that most of the missing comparative complements are of one of the following: "than previously", "than the one just mentioned" or "than is the case". For example, take another look at sentences 28 - 30 from section 8:

28) The abandoned creature ... committed a still graver sin (than the one just mentioned)
29) At eight o'clock the wind was blowing more strongly (than previously)
30) I should be colder (than is the case) if I were dead.

To determine the nature of the comparand, it seems clear that the use of pragmatic information is needed; how this might be done is a question which we postpone until a later paper.

Another problem we propose to address is a correct explanation of the interaction between comparatives and coordination as in sentences 38a) - d)

38a) No one could have a safer and more promising future than John.
38b) Mary is more attractive than Jane and Sarah.
38c) John visits the US more often than Mary, but less frequently than Carol.
38d) The results were better than John claimed, but worse than we had hoped.

Certain technical difficulties arise here if a direct application of most theories of conjunction known to us is used; however, sentences of this kind seem to be very rare (we only found one such example in two full-length novels), and the problem, although of considerable theoretical interest, appears to be fairly minor one from the practical point of view.

ACKNOWLEDGEMENTS

We would like to thank three anonymous referees for valuable comments and criticism.
REFERENCES


[Barwise & Cooper 81] Barwise, J. and Cooper, R., Generalized Quantifiers and Natural Language, Linguistics and Philosophy 4, p. 159-219

[Bowen & Carvalho 88] Bowen, J. and Carvalho, A., A Logic Grammar Formalism for Pronoun Endophora, unpublished manuscript, 1988


[Cooper 83] Cooper, R. Quantification and Syntactic Theory, D. Reidel, 1983


[Hankamer 71] Hankamer, J., Why there are two "than"s in English, CLS 9, p. 179-192


[Lesmo & Torasso 85] Lesmo, L. and Torasso, P., Analysis of Conjunction in a Rule-Based Parser, in Proc. 23rd ACL, Chicago


[Pereira 83] Pereira, F.N.C., Logic for Natural Language Analysis, SRI Technical Note No 275, 1983


[Rayner & Banks 88a] Rayner, M. and Banks, A., Parsing and Interpreting Comparatives, in Proc. 26th ACL, Buffalo, 1988

33

Appendix 1: Terminology

Comparative Clause: the clause introduced by the comparison marker

Compared Element: the largest constituent in the main or the comparative clause
the leftmost element of which is more or the comparative quantifier $Q$

Comparison Marker: words like than, as, before, after.

Head of the Comparison: refers to the compared element in the main clause

Phrasal Comparative: a comparative complement which appears to be the reduced
form of a comparative clause. This may be a remnant of the application of
Comparative Ellipsis to a comparative clause, or it may be base generated.

$Q$: An (implicit or explicit) quantifier which is extraposed in the interpretation of
clausal comparatives.
Appendix 2: A logic grammar for comparatives

The syntax

% The syntactic part of the system, written as an XG-grammar [Pereira 83]
% Internal XG stuff, used for extraposing constituents.

open_x_bracket ... close_x_bracket --> [].
extrapose(s_comp(S)) ... s_comp(S) --> [].
extrapose(vp_comp(Vp,Controller)) ... vp_comp(Vp,Controller) --> [].
extrapose(np(A,B)) ... np(np(A,B)) --> [].
extrapose(article(A,B,C)) ... article(A,B,C) --> [].
extrapose(adv(A,B)) ... adv(adv(A,B)) --> [].

% Clauses

question(y_n_q(S)) -->
  extraposed_verb,
  s(S).

question(wh_q(s(subject(Subject,Controller),Vp))) -->
question_np(Subject),
vp(Vp,Controller).

% Swedish is a verb-second language, so a fronted WHQ forces the
% next thing to be a verb.

question(wh_q(S)) -->
  extraposed_question_element,
  extraposed_verb,
  s(S).

main_s(S) --> s(S).

main_s(S) -->
  extraposed_adv,
  extraposed_verb,
  s(S).

% Subjects are distinguished from Roles and NP's. The second component is
% a Control variable, i.e. a pointer to the variable in the associated VP
% which will at some stage be unified with that in the Subject's NP. This
% may seem perverse (why not unify them at once?), but is actually a sensible
% way of treating subject contrasting (and also conjoined subject NP's,
% though we don't do that here).

s(s(subject(Subject,Controller),Vp)) -->
np(Subject),
vp(Vp,Controller).

extraposed_verb ... verb(Verb,Verb_var,Mood,Voice,Frame_pattern) -->
  verb(Verb,Verb_var,Mood,Voice,Frame_pattern).

extraposed_question_element ... np(Question_element) -->
question_np(Question_element).
extraposed_question_element ... preposition(Prep), np(Question_element) -->
preposition(Prep),
question_np(Question_element).

extraposed_adv ... adv(Adv) -->
adv(Adv).

% The XG-pre-processor has been modified slightly so as to give a simple
% transparent treatment of bound anaphora. (This idea was borrowed from
% [Bowen & Carvalho 88]). The form with extra_anaphor(Anaphor, Constituents)
% means that Constituents are to be parsed with Anaphor available as an
% anaphoric referent. This is used to treat verbs in comparative complements
% like "do" and "is", which (we claim, at any rate), refer back to some
% verbal antecedent, e.g. "Which kings reigned longer than Charles I did?"

% The structure of a VP is vp(Verb_var, Mood, Frame), where
% Verb_var is a "typed variable" [Dahl 77], representing the verb-event.
% Mood is the mood (finite/infinite)
% Frame is a list of argument and complement structures.

vp(vp(Verb_var, Mood, Frame), trace_np(Controller)) -->
verb(Verb, Verb_var, Mood, Voice, Frame_pattern),
with_extra_anaphor(verb(Verb, Voice),
(verb_args(Frame_pattern, Frame, Mods, trace_np(Controller), Voice),
verb_mads(Mods, []))).

% NPs. As explained in the paper (p. 6), NP's consist of a det + a CN.

question_article(Article) :-
  member(Article, [which, which_pl, what, who, how_many(_)]).

question_np(Np) -->
np(Np),
{Np = np(det([Article, _, _]), cn(Head, Type, Mods)),
 (question_article(Article);
  member(role(_, np(det([Article_1, _, _]), Ch)), Mods),
  question_article(Article_1)).

np(np(det([Art, Det, Num]),
  cn(Noun, Type, Mods))) -->
det(det([Art, Det, Num]),
  cn(Noun, Type, Mods), Det, Num),
  np_constraints(Art, Det, Num),
optionally_signal_comparative(np(det([Art, Det, Num]),
  cn(Noun, Type, Mods)),
  [Art|Mods]),
  optional_comparand.

np(np(det([[], det, sing]),
  cn(Name, Type, [name(Name)])]) -->
name(Name, Type).

np(np(det([virtual, det, sing]),
  cn(Pronoun, Type, []))) -->
pronoun(Pronoun, Type).
det(det([Art,Det,Num])) -->
article(Art,Det,Num).

det(det([],_,_)) --> [].

cn(cn(Noun,Type,Mods),Det,Num) -->
optional_adj(Mod),
optional_noun(Mod,Noun,Det,Num,Type).

% No undetermined NP's without articles

np_constraints(Art,Det,Num) :-
  dif(Art,[]).

np_constraints([],_,_).

% If there's an adjective, then we don't necessarily need a noun.

optional_noun(Mod,Noun,Det,Num,Type) -->
noun(Noun,Det,Num,Type).

optional_noun([Adj],omitted_noun,_,_,Type) --> [].

optionally_signal_comparative(Np,Potential_comparatives) ...
optional_adj([Adj]) → adj(Adj).
optional_adj([]) → [].

adj(adj(Adj,X,[])) → adjective(Adj).

adj(comparative(Adj, Degree, X, adj(Adj, Y, [comp_degree(Degree, X)]), Complementizer, Comp)) → comparative_adjective(Adj, Degree, Complementizer).

%Verb args

%Base case. If we've parsed all the frame, we're finished
verb_args([], Mods, Mods, _, _) → [].

verb_args([Pattern|Rest_pattern], [Arg|Next_args], Out_args, Control, Voice) → 
verb_arg(Pattern, Arg, Control, Voice),
verb_args(Rest_pattern, Next_args, Out_args, Control, Voice).

verb_arg(subject, role(subject, Controller), 
Control, active) → [].

verb_arg(predicate, role(predicate, Predicate_np), 
Controller, active) → predicate(Predicate_np).

verb_arg(agent(Type), role(agent, Controller), Controller, active) → 
{check_np_type(Controller, Type)}.

%With a passive verb, the agent is omitted.
verb_arg(agent(_), role(agent, null), _, passive) → [].

%With an active verb, we parse an object by parsing an NP
verb_arg(object(Type), role(object, Np), _, active) → 
np(Np), {check_np_type(Np, Type)}.

%But with a passive verb, the surface subject is the object
verb_arg(object(Type), role(object, Controller), Controller, passive) → 
{check_np_type(Controller, Type)}.

verb_arg(prep_object(Prep, Type), role(prep_object, Np), _, _, _) → 
preposition(Prep),
np(Np), {check_np_type(Np, Type)}.

verb_arg(relational_object, role(Head, np(Det, cn(Head, Type, Mods))), _, active) → 
np(np(Det, cn(Head, Type, Mods))).

verb_arg(s_comp, role(s_comp, S), _, active) → 
s_comp(S).

verb_arg(inf_comp, role(inf_comp, s(subject(Controller, Controller_1), Vp)), 
Controller, active) → 
vp_comp(Vp, Controller_1).
s_comp(S) -->
    [att], s(S).

vp_comp(Vp, Controller) -->
    Vp(Vp, Controller).

predicate(np(det([]), undet, sing),
    cn(predicate, Type, [Phrasal_comp])) -->
    phrasal_comparative(Phrasal_comp).

phrasal_comparative(phrasal_comparative(Adj, Degree, Np)) -->
    comparative_adjective(Adj, Degree, Complementizer),
    complementizer(Complementizer),
    np(Np).

check_np_type(trace_np(_, _).

check_cn_type(cn(Head, Np_type, Mods), Type) :-
    can_unify(Np_type, Type).

check_np_type(np(Det, cn(Head, Np_type, Mods)), Type) :-
    check_cn_type(cn(Head, Np_type, Mods), Type).

check_np_type(comp_np(Det, Cn1, Cn2), Type) :-
    check_cn_type(Cn1, Type),
    check_cn_type(Cn2, Type).

% A complement to a comparative. This can only be parsed if we have
% earlier detected a comparative marker, which has made its presence
% known by putting a comparative_signalled element on the extraposition
% list.

optional_comparand -->
    comparison_signalled(Complementizer, Extraposed, Complement),
    complementizer(Complementizer),
    comparand_complement(Complementizer, Extraposed, Complement).

% First case: an NP or PP comparand, the simplest (and commonest)
% kind of contrastive comparative. Section 5.1. We may have an
% "anaphoric verb" with an NP, which is going to be available
% from the Bowen-Carvalho anaphor stack (see comments above, p. 37).

comparand_complement(Complementizer, _, subject_np(Np)) -->
    optional_vad(Complementizer),
    np(Np),
    anaphoric_verb.

comparand_complement(_, _, Np) -->
    np(Np).

comparand_complement(_, _, Role) -->
    pp(Role).

% Second case: an "s-operator" comparand, as in section 5.2

comparand_complement(Complementizer, _, s_operator(Inner_s, S)) -->
    optional_vad(Complementizer),

40
open_x_bracket,
extrapose(s_comp(trace_s(Inner_s))),
s(S),
close_x_bracket.

% Third case: CN-comparison (section 5.3)
comparand_complement(_,_,_,Cn) -->
   cn(Cn,Det,Num).

% Fourth case: VP-operator comparison (section 5.4.1)
comparand_complement(Complementizer,_,vp_operator(Inner_vp,Controller,S)) -->
   optional_vad(Complementizer),
   open_x_bracket,
   extrapose(vp_comp(trace_vp(Inner_vp),trace_np(Controller))),
   s(S),
   close_x_bracket.

% Fifth case: contrasted prepositions (section 5.4.2)
comparand_complement(_,_,_,role_marker(Prep,Role)) -->
   preposition(Prep).

% Sixth case: clausal comparison (section 3)
comparand_complement(Complementizer,Extraposed,S) -->
   optional_vad(Complementizer),
   open_x_bracket,
   extrapose(Extraposed),
   s(S),
   close_x_bracket.

% Of course, the optional comparand will usually not be present!
% However, if a comparison has been signalled we have to pick it up
% somewhere, otherwise it will still be hanging around on the extraposition
% list when we reach the end of the sentence.
optional_comparand --> [].
complementizer(null) --> [],
complementizer(Comp) --> [Comp],{Comp \== null}.

% The "vad" phenomenon in Swedish (section 9, p. 30).
optional_vad('n') --> [vad].
optional_vad('n') --> [].
optional_vad(Other) -->
   {dif(Other,'n')}.% Getting an 'anaphoric verb' - see comments on page 37.
anaphoric verb -->
   verb(Verb,_,_,_,_),
anaphor_stack(Potential_referents),
   {verb_anaphora_antecedent(Verb,Potential_referents)}.
verb_anaphora_antecedent(Verb,Potential_referents) :-
member(Ref,Potential_referents),
verb_anaphora_match(Verb,Ref).

verb_anaphora_match(Verb,verb(Ref_verb,active)) :-
   verb_anaphora_match_1(Verb,Ref_verb,active).

verb_anaphora_match(Verb,verb(Ref_verb,passive)) :-
   verb_anaphora_match_1(Verb,Ref_verb,passive).

% Rules for verb anaphora (grossly oversimplified - this is not
% the main subject of the paper!)
verb_anaphora_match_1('do',_,active).
verb_anaphora_match_1('have',have,active).
verb_anaphora_match_1('be_passive','_',passive).
verb_anaphora_match_1('be','be',active).

% Verb mods
verb.mods(Mods,Mods) --> [].
verb.mods([Mod|Mods],mods) -->
   verb.mod(Mod),
   optional_comparand.

verb.mod(Pp) --> pp(Pp).
verb.mod(Adv) --> adv(Adv).

pp(role(Role,Np)) -->
   preposition(Prep),
   np(Np),
   {Np = np(_,cn(_,Type,_))},
   interpret_preposition(Prep,Role,Type). % A call to the case-grammar
   % expert

% Adverbs
adv(comparative(Adverb,Degree,X,adv(Adverb,Y,[comp_degree(Degree,X)]),
   Complementizer,Complement)) ... .
   comparison_signalled(Complementizer,adverb(Adverb,X),Complement) -->
   (adverb(Adverb,comp,Complementizer),
   {Degree = comp};
   adjective_modifier(Degree,Complementizer),
   adverb(Adverb,pos,[_]).

adv(adv(Adverb,X)) -->
   adverb(Adverb,pos,[]).

adv(verbals_adv(Adverb)) -->
   adverb(Adverb,verbal,[]).

adjective(Adj) -->
   [Word],[word(Word,Adj,adjective)].

adjective_modifier(Mod,Complementizer) -->
   [Word],[word(Word,Mod,adjective_modifier(Complementizer))].

adverb(Adverb,Degree,Complementizer) -->
   [Word],[word(Word,Adverb,adverb(Degree,Complementizer))].
article(Comp_type,Comp_art,X,article(comp_article(Comp_type,Comp_art,X),Det,Num),Complementizer,Complement),
    Det,Num) -->
comparative_article(Comp_art,Comp_type,Complementizer,Det,Num).

comparative_adjective(Adj,comp,'n') -->
    [Word],{word(Word,Adj,comparative_adjective)}.

comparative_adjective(Adj,Mod,Complementizer) -->
    adjective_modifier(Mod,Complementizer),
    [Word],{word(Word,Adj,adjective)}.

comparative_article(Comp_art,Comp_type,Complementizer,Det,Num) -->
    [Word],
    {word(Word,Comp_art,article(comparative,Comp_type,Complementizer,Det,Num))}.

name(Name,Type) -->
    [Word],{word(Word,Name,name(Type))}.

noun(Noun,Det,Num,Type) -->
    [Word],{word(Word,Noun,noun(Det,Num,Type))}.

preposition(Prep) -->
    [Word],{word(Word,Prep,preposition)}.

pronoun(Pronoun,Type) -->
    [Word],{word(Word,Pronoun,pronoun(Type))}.

verb(Verb,Verb_var,Mood,Voice,Frame_pattern) -->
    [Word],{word(Word,Verb,verb(Voice,Mood,Verb_var,Frame_pattern))}. 

43
The semantics

% Top level

top_sem(y_n_q(S),y_n_q(Sem)) :- !,
   s_form(S,Sem).

top_sem(wh_q(S),wh_q(Sem)) :- !,
   s_form(S,Sem).

top_sem(S,Sem) :-
   s_form(S,Sem).

% s_form first converts S into a quant_tree S_item, and then reduces
% it (i.e. the quant_tree) into a logical Form.

s_form(S,Form) :-
   sem(S,S_item),
   reduce_S_item(S_item,Form).

reduce_S_item(s_item(Quant_tree),Form) :-
   reduce(Quant_Tree,Form,[],[],sentential).

% sem is the first stage of the interpretation process, and converts a
% syntax tree (first argument) into a quant_tree (second argument) in a
% compositional fashion.

sem_for_list([],[]).

sem_for_list([First_mod|Rest_mods],[First_sem|Rest_sems]) :-
   sem(First_mod,First_sem),
   sem_for_list(Rest_mods,Rest_sems).

sem(name(Name),relation(name_of,[Name])).

sem(comparative(Relation,Comp_type,Comp_var,_,_),s(Subj,Vp)),
   clausal_comparand_item(Relation,Comp_var,Complement_sem)) :-
   s_form(s(Subj,Vp),Complement_sem).

sem(comparative(Relation,Comp_type,_,_,Complement),
   comparison_item(Relation,Q,comparand_item(Q,Comp_type,Compl_sem))) :-
   non_clausal_comparative_complement(Complement),
   sem(Complement,Compl_sem).

non_clausal_comparative_complement(Complement) :-
   (Complement = role(_,_)),
   Complement = np(_,_),
   Complement = subject_np(_),
   Complement = cn(_,_),
   Complement = s_operator(_,_),
   Complement = vp_operator(_,_,_),
   Complement = role_marker(_,_)).

sem(phrasal_comparative(Adj,Comp_type,np(Det, cn(Head, Type, Mods))),Sem) :-
   sem(adj(Adj, Prop_1, [role(null,np(Det,cn(Head, Type, Mods)))],Sem).
[adj(Adj,Prop_2,[comp_degree(Comp_type,Prop_1)]|Mods))]),
    Sem).
sem(adj(Adj,Prop,Mods),
    role_item(Adj,quant([property, det, sing], cn_item(Prop, Subitems)))) :-
    sem_for_list(Mods, Subitems).
sem(comp_degree(Comp_type, Var), inverse_relation(Comp_type, [Var])).
sem(verbal_adv(Adv), relation(Adv, [])).
% This is for "s_operators"
sem(trace_s(S_var), trace_s_item(S_var)).
% And this is for VP_operators.
sem(trace_vp(Vp_var), trace_vp_item(Vp_var)).
sem(role(Role,Arg), role_item(Role,Arg_item)) :-
    sem(Arg,Arg_item).
sem(subject(Np, trace_np(Control)), subject_item(Np_sem, Control)) :-
    sem(Np,Np_sem).
% Subject np's (possibly not a very good name) are NP's occurring in
% comparative complements which are forced to contrast against a subject, due
% to being followed by an "anaphoric verb" (e.g. "He likes Carol more than
% *Mary does* - because of the *does*, Mary can't be contrasted with Carol.
% See page 37.
sem(subject_np(Np), subject_np_item(Np_sem)) :-
    sem(Np,Np_sem).
sem(det([[comparative(Type,Deg,Ext,Comp_var,Compl,Comp),D,Num]]),
    [Comp_sem,D,Num]) :-
    sem(comparative(Type,Deg,Ext,Comp_var,Compl,Comp),Comp_sem).
sem(det([[Det,D,N]]),[Det,D,N]) :-
    functor(Det,comparative, ).
sem(cn(., Np_var, Mods), cn_item(Np_var, Body)) :-
    sem_for_list(Mods, Body).
sem(np(Det,Cn), quant(Det_sem,Cn_sem)) :-
    sem(Det,Det_sem),
    sem(Cn,Cn_sem).
sem(trace_np(Var), quant(virtual_det, cn_item(Var, [])).
sem(role_marker(Prep, Role), role_marker_item(Prep, Role)).
sem(s_operator(Var,S), s_operator_item(Var, S_operator_sem)) :-
    sem(S, S_operator_sem).
sem(vp_operator(Vp_var, Control_var, S),
    vp_operator_item(Vp_var, Control_var, S_sem)) :-
sem(S,S_sem).
sem(s(Subject,Vp),s_item(Quant_tree)) :-
    sem(Subject,Subj_sem),
    sem(Vp,Vp_sem),
    perform_raising([Subj_sem,Vp_sem],Quant_tree).
sem(vp(Verb_var,_,Frame),
      vp_item(Verb_var,Quant_tree)) :-
    remove_empty_agent(Frame,Frame_1),
    sem_for_list(Frame_1,Quant_tree).
remove_empty_agent([],[]).
remove_empty_agent([role(agent,null)|Rest],Rest) :- !.
remove_empty_agent([First|Rest],[First|Rest_1]) :-
    remove_empty_agent(Rest,Rest_1).

% perform_raising is what actually does the quant_tree rewriting, moving the
% comparand nodes around and contrasting them against suitable things to make
% comparisons.
perform_raising(Quant_tree_0,Quant_tree) :-
    perform_raising_1(Quant_tree_0,Quant_tree,[]).
perform_raising_1([],[],[]).
perform_raising_1([First|Rest],[First_1|Rest],Raised_item) :-
    raise_comparative(First,First_1,Raised_item).

% We need to introduce perform_raising_2 so as not to get an infinite
% recursion. perform_raising_2 can only move things up, and can't merge.
perform_raising_1(Rest,[Comparison,C_slot_operator|Rest_1],[]) :-
    perform_raising_2(Rest,Rest_1,Raised_item),
    s_operator_comparison(Raised_item,C_slot_operator,Comparison).

s_operator_comparison(comparand_item(Q,Comp_type,Comp_sem),
                      C_slot_operator,
                      comparison(Comp_type,
                      Q,
                      Comp_sem,
                      assert_operator,
                      C_slot_operator)) :-
    Comp_sem = s_operator_item(_,_).
perform_raising_1(Q_tree,[Comparison|Q_tree_2],[]) :-
    perform_raising_2(Q_tree,Q_tree_1,comparand_item),
    Comparand_item = comparand_item(_,_),
    compatible_for_comparison_0(Comparand_item,Q_tree_1,Q_tree_2,
    Contrasted_item,C_slot),
    comparison_item(Comparand_item,Contrasted_item,C_slot,Comparison).
perform_raising_1([First|Rest],[First|Rest_1],Compared_item) :-
    perform_raising_1(Rest,Rest_1,Compared_item).
perform_raising_2([First|Rest],[First|Rest],Raised_item) :-
    raise_comparative(First,First_1,Raised_item).
perform_raising_2([First|Rest],[First|Rest_1],Comparand_item) :-
    perform_raising_2(Rest,Rest_1,Comparand_item).

% Raise_comparative lifts a comparand out of a constituent, leaving a
% Q-placeholder behind.
raise_comparative(comparison_item(Relation,Q,Raised_comparative),
    comparison_placeholder(Relation,Q),
    Raised_comparative).
raise_comparative(quant([Det0,D,N],Cn_item),
    quant([Det1,D,N],Cn_item),
    Raised_comparative) :-
    raise_comparative(Det0,Det1,Raised_comparative).
raise_comparative(quant(Det,cn_item(Var,[Comp0])),
    quant(Det,cn_item(Var,[Comp1])),
    Raised_comparative) :-
    raise_comparative(Comp0,Comp1,Raised_comparative).
raise_comparative(role_item(Role,Quant0),role_item(Role,Quant1),
    Raised_comparative) :-
    raise_comparative(Quant0,Quant1,Raised_comparative).
raise_comparative(subject_item(Quant0,Var),subject_item(Quant1,Var),
    Raised_comparative) :-
    raise_comparative(Quant0,Quant1,Raised_comparative).
raise_comparative(vp_item(Verb_var,Quant_tree_0),
    vp_item(Verb_var,Quant_tree),
    Raised_comparative) :-
    perform_raising_1(Quant_tree_0,Quant_tree,Raised_comparative).

% comparison_item constructs a comparison node from the various
% bits and pieces that go into it - the type, the Q pointer, the
% compared and contrasted items, and the C-slot. See page 15.
comparison_item(comparand_item(Q,Comp_type,Compared_item),
    Contrast_set_item,C_slot,
    comparison(Comp_type,
        Q,
        Compared_item,
        Contrast_set_item,
        C_slot)).

% compatible_for_comparison checks a comparand_item against something that
% might be a contrastable item (or contain one). The last three arguments
% are: what the thing is replaced by (either the C-slot, or something
% containing the C-slot), the extracted contrasted item, and a pointer to the
% C-slot.
compatible_for_comparison_0(Comparand,[F|R],[F_1|R],Contrast,C_slot) :-
    compatible_for_comparison_1(Comparand,F,F_1,Contrast,C_slot).
compatible_for_comparison_0(Comparand,[F|R],[F|R_1],Contrast,C_slot) :-
    compatible_for_comparison_0(Comparand,R,R_1,Contrast,C_slot).
compatible_for_comparison_1(Comparand,Item,Modified_item,Contrast,C_slot) :-

compatible_for_comparison(Comparand, Item, Modified_item, Contrast, C_slot),
\+contains_placeholder(Contrast).

% First case: We're looking at a role_item, and the comparand also contains a
% role_item. (John has more books in the bedroom than in the kitchen)

compatible_for_comparison(comparand_item(Q, Comp_type,
    role_item(Role, Comp_quant)),
    role_item(Role, Quant),
    C_slot, role_item(Role, Quant), C_slot).

% Second case: we're looking at a role_item, and comparing with an NP.
% (John has more books in the bedroom than the kitchen).

compatible_for_comparison(comparand_item(Q, Comp_type, quant(Det, Np)),
    role_item(Role, Quant), role_item(Role, C_slot), Quant, C_slot).

% Third case: we're looking at a role_item, and comparing with a
role_marker_item
% (Carol Friedmann, "more women voted for the proposal than against").

compatible_for_comparison(comparand_item(Q, Comp_type,
    role_marker_item(Prep, Comp_role)),
    role_item(Role, Quant),
    role_item(C_slot, Quant), Role, C_slot) :-
    Quant = quant(Det, cn_item(Type, _)),
    interpret_preposition(Prep, Comp_role, Type),
    contrastable_roles(Comp_role, Role).

contrastable_roles(X, Y) :-
    (contrastable_roles_1(X, Y);
     contrastable_roles_1(Y, X)).

contrastable_roles_1(X, X).
contrastable_roles_1(for, against).

% Fourth case: we're looking at a subject_item, and comparing with an NP
% (John has more books in the kitchen than Mary).

compatible_for_comparison(comparand_item(Q, Comp_type, quant(Det, Np)),
    subject_item(Quant, Control),
    subject_item(C_slot, Control), Quant, C_slot).

% Fifth case: we're looking at a subject_item, and comparing with a
% subject_np. (John has more books in the kitchen than Mary does).

compatible_for_comparison(comparand_item(Q, Comp_type, subject_np_item(_)),
    subject_item(Quant, Control),
    subject_item(C_slot, Control), Quant, C_slot).

% Sixth case: we're looking at a vp_operator_item, and comparing with a
% subject_item. (John has more books in the kitchen than Mary is able to).

compatible_for_comparison(comparand_item(Q, Comp_type, vp_operator_item(_, _, _)),
    subject_item(Quant, Control),
    subject_item(C_slot, Control), Quant, C_slot).
% Seventh case: we're looking at a subject_item, and comparing with a
% cn_item. (More men have books in the kitchen than women).

compatible_for_comparison(comparand_item(Q,Comp_type,cn_item(_,_)),
subject_item(qquant(Det,Cn_item),Control),
subject_item(qquant(Det,C_slot),Control),
Cn_item,C_slot).

% Eighth case: we're looking at a vp_item, and comparing with anything. We
% just pass it down into the vp_item's frame:

compatible_for_comparison(Comparand_item,
vp_item(Var,Frame),
vp_item(Var,Frame_1),
Contrasted,C_slot).

compatible_for_comparison_0(Comparand_item,Frame,Frame_1,Contrasted,C_slot).

% Reduce turns the quant_tree into a logical form by recursively going
% through it.
% The arguments are:
% 1: Quant_tree
% 2: Piece of logical form
% 3: the variable for the "head" - the verb if this is a verb structure, the
% noun if it's a nominal
% 4: an alist of relations already picked up, i.e. thematic roles relating to
% the head object (arg 3).
% 5: whether this is nominal or verbal (affects whether or not we quantify
% over
% the head variable when we get to the innermost level.)

reduce([],[],_,_,sentential).

reduce([],ex_event(Typed_var,Body),Typed_var,Pre_relations,verbal) :-
build_relations(Pre_relations,Typed_var,Relations),
join(Join_body,Body,Relations).

reduce([],Body,Typed_var,Pre_relations,nominal) :-
build_relations(Pre_relations,Typed_var,Relations),
join(Join_body,Body,Relations).

reduce([subject_item(Q,Control_var)|Rest],
Structure,Var,Pre_relations,V or n) :-
(Quant = qquant(Det,cn_item(Typed_var,Subitems));
Quant = subject_np_item(qquant(Det,cn_item(Typed_var,Subitems))));
Typed_var = Control_var,
reduce(Subitems,Restrictions,Typed_var,[],nominal),
reduce(Rest,Attributes,Var,Pre_relations,V or n),
det_sem(Det,Structure,Restrictions,Typed_var,Attributes).

% Definitely a bit tricky. The vp_operator_item's third element is an
% S denotation with two slots, which the other two components point to.
% Vp_var is the missing VP denotation, which needs to be filled in from
% the stuff under the subject node, i.e. what we get in Attributes.
% Vp_control_var is the control variable for the subject in the vp_operator,
% and thus has to be unified with the subject's own control variable.

reduce([subject_item(Vp_operator,Control_var)|Rest],
Structure,Var,Pre_relations,V or n) :-
Vp operator = vp_operator_item(Vp_var, Vp_control_var, S_item),
reduce(Rest, Attributes, Var, Pre_relations, V_or_n),
Vp_control_var = Control_var,
Vp_var = Attributes,
reduce_s_item(S_item, Structure).

reduce([vp_operator_item(Vp_var, Control_var, S_sem)|Rest], Form, Var,
Pre_relations, V_or_n) :-
reduce(Rest, Inner_form, Var, Pre_relations, V_or_n),
reduce_s_item(S_sem, Form),
S_var = Inner_form.

reduce([vp_item(Verb_var, Quant_tree)]|Structure, _, _, _) :-
reduce(Quant_tree, Structure, Verb_var, [], verbal).

reduce([trace_vp_item(Structure)]|Structure, _, _, _).

reduce([role_item(Role, quant(Det, cn_item(Typed_var, Subitems)))|Rest],
Structure, Var, Pre_relations, V_or_n) :-
reduce(Subitems, Restrictions, Typed_var, [], nominal),
reduce(Rest, Attributes, Var, [relation(Role, [Typed_var])|Pre_relations],
V_or_n),
det_sem(Det, Structure, Restrictions, Typed_var, Attributes).

reduce([s_item(Quant_tree)|Rest], Structure, Var, Pre_relations, V_or_n) :-
reduce_s_item(s_item(Quant_tree), S_sem),
reduce(Rest, Rest_sem, Var, Pre_relations, V_or_n),
conjoin(Structure, [S_sem, Rest_sem]).

reduce([role_item(Role, s_item(Quant_tree))|Rest],
Structure, Var, Pre_relations, V_or_n) :-
reduce_s_item(s_item(Quant_tree), S_sem),
reduce(Rest, Structure, Var, [relation(Role, [S_sem])|Pre_relations],
V_or_n).

reduce([role_item(Role, trace_s_item(S))|Rest],
Structure, Var, Pre_relations, V_or_n) :-
reduce(Rest, Structure, Var, [relation(Role, [S])|Pre_relations],
V_or_n).

reduce([clausal_comparand_item(Rel, Prop_var, S_sem)|Rest],
Structure, Var, Pre_relations, V_or_n) :-
Prop_var = [property|_],
reduce(Rest, Rest_structure, Var,
[relation(Rel, [Prop_var])|Pre_relations], V_or_n),
conjoin(Structure, [S_sem, Rest_structure]).

reduce([relation(Rel, Args)|Rest], Structure, Var, Pre_relations, V_or_n) :-
reduce(Rest, Structure, Var, [relation(Rel, Args)|Pre_relations], V_or_n).

reduce([inverse_relation(Rel, Args)|Rest],
Structure, Var, Pre_relations, V_or_n) :-
reduce(Rest, Structure, Var,
[inverse_relation(Rel, Args)|Pre_relations], V_or_n).

reduce([assert_operator|Rest], Form, Var, Pre_relations, V_or_n) :-
reduce(Rest, Form, Var, Pre_relations, V_or_n).
reduce([s_operator_item(S_var,S_sem)|Rest],Form,Var,Pre_relations,V_or_n) :-
    reduce(Rest,Inner_form,Var,Pre_relations,V_or_n),
    reduce_s_item(S_sem,Form),
    S_var = Inner_form.

reduce([comparison(C_type,Q,Compared,Contrasted,C_slot)|Rest],
        Form,Var,Pre_relations,V_or_n) :-
    add_inequality_to_compared(Contrasted,Compared,Modified_compared),
    reduce_comparison(C_type,Q,Modified_compared,Contrasted,C_slot,Rest,
                      Form,Var,Pre_relations,V_or_n).

reduce([direct_comparison_item(Comp_art,NL,NG,Item_1,Item_2)|Rest],
        Form,Var,Pre_relations,V_or_n) :-
    prepare_unique_form(Rest,Var,Rest_2,Var_2,_,
                        Rest_1,Var,Pre_relations,V_or_n),
    reduce([Item_1|Rest_1],Form_1,Var,Pre_relations,V_or_n),
    reduce([Item_2|Rest_2],Form_2,Var_2,Pre_relations,V_or_n),
    Comparison = .. [Comp_art,NL,NG],
    conjoin(Form,Form_1,Form_2,Comparison).

reduce([comparison_placeholder(Adj_or_adv,q(C_var,Condition))]|Rest),
        ex_prop(C_var,Body),Var,Pre_relations,V_or_n) :-
    C_var = [property|_],
    reduce(Rest,Attributes,Var,
            [relation(Adj_or_adv,[C_var])|Pre_relations],V_or_n),
    conjoin(Body,[Condition,Attributes]).

% This is the key clause, and corresponds almost exactly to the
% formula on page 11.

reduce_comparison(C_type,Q,Compared,Contrasted,C_slot,Rest,
                    Form,Var,Pre_relations,V_or_n) :-
    prepare_unique_form(Rest,Var,Inner_rest,Inner_var,Inner_sub,_,
                        q(Inner_q),
                        q(C_slot,Inner_sub,[Inner_c_slot]),
                        Q = q(NL,NL_condition),
                        Inner_q = q(NL,NL_condition),
                        NL_condition = .. [C_type,NL,NG],
                        C_slot = Contrasted,
                        Inner_c_slot = Compared,
                        reduce(Inner_rest,NL_condition,Inner_var,Pre_relations,V_or_n),
                        reduce(Rest,Form,Var,Pre_relations,V_or_n)).

prepare_unique_form(Rest_items,Var,Own_rest_items,Own_var,Subs) :-
    copy_term_with_subsr([Rest_items,Var],[Own_rest_items,Own_var],Subs).

% add_inequality_to_compared(Contrasted,Compared,Modified_compared)
% As explained in section 7.1, comparison between quantified things seems
% to assume that they are non-coreferential.

add_inequality_to_compared(role_item(_,Quant1),
                            role_item(Role,Quant2),
                            role_item(Role,Modified_quant2)) :-
    !,add_inequality_to_compared(Quant1,Quant2,Modified_quant2).

add_inequality_to_compared(quant(_,Cn1),
                           quant(Det,Cn2),
                           quant(Det,Modified_cn2)) :-
    !,add_inequality_to_compared(Cn1,Cn2,Modified_cn2).

51
add_inequality_to_compared(cn_item(Var1,Body1),
   cn_item(Var2,Body2),
   cn_item(Var2,
       [relation(not_equal,[Var1]|Body2)]) :- !.

add_inequality_to_compared(_,Y,X).

build_relations([],_,[]).

build_relations([relation(null,Args)|Pre_relations],Var,Relations) :-
   build_relations(Pre_relations,Var,Relations).

build_relations([relation(Role,Args)|Pre_relations],
   Var,[Relation|Relations]) :-
   Relation \== null,
   (Role = role_marker_item(_,Actual_role);
    atom(Role), Role = Actual_role),
   Relation \== [Actual_role,Var|Args],
   build_relations(Pre_relations,Var,Relations).

build_relations([inverse_relation(Role,Arg)|Pre_relations],
   Var,[Relation|Relations]) :-
   (Role = role_marker_item(_,Actual_role);
    atom(Role), Role = Actual_role),
   Relation \== [Actual_role,Arg,Var],
   build_relations(Pre_relations,Var,Relations).

det_sem(virtual_det,Range,R,X,A) :- !,conjoin(Range,[R,A]).
det_sem([a,undet,sing],ex(X,R,A),R,X,A) :- !.
det_sem(['some',undet,sing],ex(X,R,A),R,X,A) :- !.
det_sem([],undet,sing],ex(X,R,A),R,X,A) :- !.
det_sem([],det,sing],the(X,R,A),R,X,A) :- !.
det_sem([property,det,sing],ex_prop(X,R,A),R,X,A) :- !,X = [property|_].
det_sem([which,_,_],wh(X,R,A),R,X,A) :- !.
det_sem([which_pl,_,_],wh_pl(X,R,A),R,X,A) :- !.
det_sem([comparison_placeholder(number,q(N,Comparison)),_],
   [and,number(N,X,R,A),Comparison],
   R,X,A) :- !,N = [number|_].
det_sem([comparison_placeholder(identity,q(X,Comparison)),_],
   ex(X,Range,A),
   R,X,A) :- !,
   conjoin(Range,[R,Comparison]).
det_sem([number(N),_],number(N,X,R,A),R,X,A) :- !,N = [number|_].
det_sem([clausal_comparand_item(number,N,Complement_sem),_],
   [and,number(N,X,R,A),Complement_sem],
   R,X,A) :- !,N = [number|_].
det_sem([comp_article(number,Comparator,Nl),_],
   [and,number(N,X,R,A),Comparator],
   R,X,A) :- !,N = [number|_],
   Comparator \== [Comparator,Nl,Nl].

copy_typed_variable(X,_):- var(X),!.
copy_typed_variable([X|Rest],[X|Rest_1]) :-
   copy_typed_variable(Rest,Rest_1).

contains_placeholder(Placeholder) :-
   functor(Placeholder,placeholder,_)!.
contains_placeholder(Term) :-
    functor(Term, _, N),
    contains_placeholder_1(Term, 1, N).

contains_placeholder_1(Term, N, N) :-
    arg(N, Term, Last_arg),
    contains_placeholder(Last_arg).

contains_placeholder_1(Term, I) :-
    arg(I, Term, Arg),
    (contains_placeholder(Arg), !; 
     I1 is I + 1, 
     contains_placeholder_1(Term, I1)).
Appendix 3: Examples

This appendix contains examples of the output produced by the program in the preceding section; we have chosen eleven sentences (representing between them all the various types of construction dealt with in the paper), and for each one give the program's syntactic and semantic analysis. To make these more intelligible for readers unfamiliar with Swedish, we have replaced all content words with their English equivalents. In addition, three points about the notation should be borne in mind:

1. Typed variables are represented in the form type/identifier. Thus for example king/A should be read as "Variable A, of type king". Types are arranged in a simple hierarchy, and two variables can only unify if their types are compatible, i.e. if one is a super-type of the other (see [Dahl 77]).

2. It has been convenient to treat the empty list [] as being a null conjunction, and thus equivalent with the value true. This is primarily of importance in connection with the next point:

3. the, a, etc. are to be read as generalized quantifiers in the sense of [Barwise & Cooper 81], and are not shorthand for the more familiar \( \forall, \exists \), etc.; thus for example \( a(x, P, Q) \) (where \( P \) and \( Q \) are forms containing free occurrences of \( x \)) means "At least one object \( x \), of which \( P \) holds, is such that \( Q \) also holds of it". In the (fairly common) case where \( P \) is equal to \( [] (=true) \), this reduces to ordinary existential quantification.
Syntax trees

1a: [vilka,kungar,regерade,längre,än,Gustav III]

which kings reigned longer than Gustav III?

wh_q(s(subject(np(det([which_pl
    undet
    plur]))
    cn(king
    king//A
    []))
    trace_np(anything//B))
vp(reign//C
    finite
    [role(agent
        trace_np(anything//B))
    comparative(long
        comp
        anything//D
        adv(long
            anything//E
            [comp_degree(comp
                anything//D)])
    än
    np(det([[]
        det
        sing]))
    cn(Gustav III
        king//F
        [name(Gustav III)])))])}
2a: [vilka, kungar, flades, under, samma, Århundrade, som, Karl XII]

which kings were-born during the same century as Karl XII?

wh_q(s(subject(np(det([which_pl
   undet
   plur]))
   cn(king
   king//A
   []))
   trace_np(anything//B))
vp(be born//C
   finite
   [role(agent
      null)
   role(object
      trace_np(anything//B))
   role(during
      np(det([comparative(identity
         same
         anything//D
         article(comp_article(identity
            same
            anything//D)
            undet
            sing)
         som
         np(det([[]
            det
            sing])
            cn(Karl XII
            king//E
            [name(Kar: XII)]))
            undet
            sing))
            cn(century
            century//F
            []))))))))

56
Charlie bought more books than (what) Mary could carry.

s(subject(np(det([[[
  det
  sing])
  cn(Charlie
  man/A
  [name(Charlie)]))
  trace_np(anything//B))
vp(buy//C
  finite
  [role(agent
    trace_np(anything//B))
  role(object
    np(det([comparative(number
      more
      anything//D
      article(comp_article:number
        more
        anything//D)
        undet
        plur)
      än
      s(subject(np(det([[[
        det
        sing])
        cn(Mary
        woman//E
        [name(Mary)]))
        trace_np(anything//F))
      vp(be able to//G
        finite
        [role(agent
          trace_np(anything//F))
        role(inf_comp
          s(subject:trace_np(anything//F)
            trace_np(anything//H))
          vp(carry//I
            infinite
            [role(agent
              trace_np(anything//H))
            role(object
              np(det([comp_article(
                number
                more
                anything//D)
                undet
                plur])
              cn(book
                book//J
                []))))))))))))})))
4a: [vilka,turister,spenderade,fler,kronor,i,London,än,i,New York]

which tourists spent more kronor in London than in New York?

wh_q(s(subject(np(det([which_pl
   undet
   plur]))
   cn(tourist
tourist//A
[]))
trace_np(anything//B))
vp(spend//C
  finite
  [role(agent
       trace_np(anything//B))
   role(object
       np(det([comparative(number
       more
       anything//D
       article(comp_article(number
       more
       anything//D)
       undet
       plur)
   än
   role(location
       np(det([]
       det
       sing))
       cn(New "York
city,//E
   [name(New York)]))))
   un
   role(location
       np(det([]
       det
       sing))
       cn(London
city,//G
   [name(London)]))))

58
for which paintings received Charlie more kronor than (what) Mary claimed?

wh_q(s{subject(np(det([]
    det
    sing)))
    cn(Charlie
    man//A
    [name(Charlie)])
    trace_np(anything//B))
vp(receive//C
    finite
    [role(agent
    trace_np(anything//B))
    role(object
    np(det([comparative(number
    more
    anything//D
    article(comp_article(number
    more
    anything//D))
    undet
    plur)
  än
  s_op(anything//E):
  s(subject(
    np(det([]
    det
    sing)))
    cn(Mary
    woman//F
    :name(Mary)])
    trace_np(anything//G))
vp(claim//H
    finite
    [role(
    agent
    trace_np(anything//G))
    role(
    s_comp
    trace_s(anything//E)))))

  cn(krona
  krona//I
  [])))
role(prep_object
np(det([which_pl
  undet
  plur])
    cn(picture
    picture//J
    [])))))))

59
6a: [fler, kvinnor, än, män, läser, Hånt i Veckan]

more women than men read 'News of the World'

s(subject(np(det[[comparative(number

more

anything//A

article(comp_article(number

more

anything//A)

undet

plur)

än

cn(män

man//B

[]))

undet

plur])

cn(woman

woman//C

[]))

trace_np(anything//D))

vp(read//E

finite

[role(agent

trace_np(anything//D))

role(object

np(det([[

det

sing])

cn(News of the World

newspaper//F

[name(News of the World)]))))])}
7a: [nu, verkar, Marie, längre, än, Kalle]

now appears Mary taller than Charlie.

s(subject np(det([]
    det
    sing))
  cn(Mary
    woman/A
    [name(Mary)])
  trace_np(anything/B))
vp(appear/C finite
  [role(subject
    trace_np(anything/B))
  role(predicate
    np(det([]
      undet
      sing))
    cn(predicate
      anything/D
      [phrasal_comparative(tall
        comp
        np(det([], det, sing))
      cn(Charlie
        man/E
        [name(Charlie)]))])])

verbal_adv(now))))}
Charlie owns more English books than French (ones).
9a: [Kalle, springer, fortare, än, vad, Marie, kan]

Charlie runs faster than Mary can.

s(subject(np(det([]
  det
  sing))
  cn(Charlie
  man/A
  [name(Charlie)]))
  trace_np(anything/B))
vp(run/C
  finite
  [role(agent
  trace_np(anything/B))
  comparative(fast
  comp
  anything/D
  adv(fast
    anything/E
    [comp_degree(comp
      anything/D))])
  än
  vp_operator(anything/F
  anything/G
  s(subject(np(det([]
    det
    sing))
    cn(Mary
    woman/H
    [name(Mary)]))
    trace_np(anything/I))
  vp(be able to/J
  finite
  [role(agent
  trace_np(anything/I))
  role(inf_comp
    s(subject(t:trace_np(anything/I)
        trace_np(anything/G))
        trace_vp(anything/F)))))))))))
10a: [vilka, kungar, regerade, längre, än, vad, Gustav II:ː, gjorde]

which kings reigned longer than (what) Gustav II:ː did?

wh_q(s(subject(np(det{[which_pl
    undet
    plur})))
    cn(king
    king//A
    [[]])
    trace_np(anything//B))
vp(reign//C
    finite
    [role(agent
       trace_np(anything//B))
    comparative(long
      comp
      anything//D
      adv(long
        anything//E
        [comp_degree(comp
          anything//D)])
      än
      subject_np(np(det{[[]
         det
         sing})))
    cn(Gustav III
        king//F
        [name(Gustav III)]))])})})})
more women voted for the-suggestion than against.
Logical forms

lb: [vilka, kungar, regerade, längre, än, Gustav III]

which kings reigned longer than Gustav III?

wh_q(wh_pl(king//A
  [  
ex_prop(property//B
      [and
      the(king//C
      [and
      name_of(king//C
      Gustav III)
      not_equal(king//C
      king//A)])
    ex_prop(property//D
        [and
        comp(property//B
        property//D)
        ex_event(reign//E
            [and
            long(reign//E
            property//D)
            agent(reign//E
            king//C)]))]
  ex_event(reign//F
    [and
    long(reign//F
    property//B)
    agent(reign//F
    king//A)])))}
2b: [vilka, kungar, föddes, under, samma, århundrade, som, Karl XII]

which kings were-born during the same century as Karl XII?

wh_q(wh_pl(king//A
    []
    ex(century//B
        the(king//C
            [and
                name_of(king//C
                    Karl XII)
                not_equal(king//C
                    king//A)]
            ex(century//D
                same(century//B
                    century//D)
            ex_event(be born//E
                [and
                    during(be born//E
                        century//B)
                    object(be born//E
                        king//C)])]
        ex_event(be born//F
            [and
                during(be born//F
                    century//B)
                object(be born//F
                    king//A)])]})}
Charlie bought more books than (what) Mary could carry.

```text
the(man//A
   name_of(man//A
      Charlie)
   [and
    number(number//B
      book//C
      []
     ex_event(buy//D
        [and
         object(buy//D
            book//C)
         agent(buy//D
            man//A))]))
the(woman//E
   name_of(woman//E
      Mary)
   ex_event(be able to//F
    [and
     inf_comp(be able to//F
      [and
       number(number//G
         book//H
         []
        ex_event(carry//I
          [and
           object(carry//I
             book//H)
           agent(carry//I
              woman//E))]))
      more(number//B
         number//G)]
    agent(be able to//F
       woman//E)))]])
```
which tourists spent more kronor in London than in New York?

wh_q(wh_pl(tourist//A
    []
    the(city//B
        name_of(city//B
            London)
    [and
        number(number//C
            krona//D
        []
        ex_event(spend//E
            [and
                object(spend//E
                    krona//D)
                location(spend//E
                    city//B)
                agent(spend//E
                    tourist//A)])
    the(city//F
        [and
            name_of(city//F
                New York)
            not_equal(city//F
                city//B)]
        [and
            number(number//G
                krona//H
            []
            ex_event(spend//I
                [and
                    object(spend//I
                        krona//H)
                    location(spend//I
                        city//F)
                    agent(spend//I:
                        tourist//A)])
            more(number//C
                number//G)]))))}
for which paintings received Charlie more kronor than (what) Mary claimed?

wh_q(the(man//A
  name_of(man//A
    Charlie)
  wh_pl(picture//B
    []
    [and
    number(number//C
      krona//D
      []
      ex_event(receive//E
        [and
        object(receive//E
          krona//D)
        prep_object(receive//E
          picture//B)
        agent(receive//E
          man//A))])
  the(woman//F
    name_of(woman//F
      Mary)
    ex_event(claim//G
      [and
      s_comp(claim//G
        the(man//H
          name_of(man//H
            Charlie)
        wh_pl(picture//I
          []
          [and
          number(number//J
            krona//K
            [:
            ex_event(receive//L
              [and
              object(receive//L
                krona//K)
            prep_object(receive//L
              picture//I)
            agent(receive//L
              man//H))])
          more(number//C
            number//J)]))])
  agent(claim//G
    woman//F)])))))))
more women than men read 'News of the World'

[and
  number(number//A
    woman//B
    [])
  the(newspaper//C
    name_of(newspaper//C
      News of the World)
    ex_event(read//D
      [and
        object(read//D
          newspaper//C)
        agent(read//D
          woman//B)]))
  [and
    number(number//E
      man//F
      not_equal(man//F
        woman//B)
    the(newspaper//G
      name_of(newspaper//G
        News of the World)
    ex_event(read//H
      [and
        object(read//H
          newspaper//G)
        agent(read//H
          man//F)]))
    more(number//A
      number//E)]]
7b: [nu, verkar, Marie, längre, än, Kalle]

now appears Mary taller than Charlie.

the (woman // A
    name_of (woman // A
        Mary)
    ex (anything // B
        ex_prop (property // C
            the (man // D
                ex_prop (property // E
                    comp (property // C
                        property // E)
                    [and
                        name_of (man // D
                            Charlie)
                        tall (man // D
                            property // E)]))
        [[]]
        tall (anything // B
            property // C))
    ex_event (appear // F
        [and
            now (appear // F)
            predicate (appear // F
                anything // B)
            subject (appear // F
                woman // A)]))}
8b: [Kalle, äger, fler, engelska, böcker, än, franska]

Charlie owns more English books than French (ones).
9b: [Kalle, springer, fortare, än, vad, Marie, kan]

Charlie runs faster than Mary can.

the (man//A
   name_of (man//A
       Charlie)
   ex_prop (property//B
       [and
       the (woman//C
           name_of (woman//C
               Mary)
           ex_event (be able to//D
               [and
               inf_comp (be able to//D
                   ex_prop (property//E
                       [and
                       comp (property//B
                           property//E)
                       ex_event (run//F
                           [and
                           fast (run//F
                               property//E)
                           agent (run//F
                               woman//C))]
                       agent (be able to//D
                           woman//C))]
           ex_event (run//G
               [and
               fast (run//G
                   property//B)
               agent (run//G
                   man//A))])])})
10b: [vilka, kungar, regerade, längre, än, vad, Gustav II:., gjorde]

which kings reigned longer than (what) Gustav II did?

```
wh_q(wh_pl(king//A
 []
 ex_prop(property//B
 [and
 the(king//C
    name_of(king//C
       Gustav III)
 ex_prop(property//D
 [and
    comp(property//B
       property//D)
 ex_event(reign//E
 [and
    long(reign//E
       property//D)
    agent(reign//E
       king//C)])])
 ex_event(reign//F
 [and
    long(reign//F
       property//B)
    agent(reign//F
       king//A)])])])
```
more women voted for the-suggestion than against.

more(number/A
  woman/B
  []
  the(suggestion/C
    []
    ex_event(vote/D
      [and
        for(vote/D
          suggestion/C)
        agent(vote/D
          woman/B)]))
  [and
    number(number/E
      woman/F
      []
      the(suggestion/G
        []
        ex_event(vote/H
          [and
            against(vote/H
              suggestion/G)
            agent(vote/H
              woman/F)])
          more(number/A
            number/E)])}