COMPARATIVES AND ELLIPSIS

S. G. Pulman
SRI International, and University of Cambridge Computer Laboratory

SRI International Cambridge Computer Science Research Centre
23 Miller's Yard, Cambridge CB2 1RQ
sgp@cam.sri.com

ABSTRACT

This paper analyses the syntax and semantics of English comparatives, and some types of ellipsis. It improves on other recent analyses in the computational linguistics literature in three respects: (i) it uses no tree- or logical-form rewriting devices in building meaning representations (ii) this results in a fully reversible linguistic description, equally suited for analysis or generation (iii) the analysis extends to types of elliptical comparative not elsewhere treated.

INTRODUCTION

Many treatments of the English comparative construction have been advanced recently in the computational linguistics literature (e.g. Rayner and Banks, 1989; Ballard, 1988). This interest reflects the importance of the construction for many natural language applications, especially those concerning access to databases, where it is natural to require information about quantitative differences and limits which are most naturally expressed in terms of comparatives and superlatives.

However, all of these analyses have their defects (as no doubt does this one). The most pervasive of these defects is one of principle: they all place a high reliance on non-compositional methods (tree or formula rewriting) for assembling the logical forms of comparatives even in cases that might be thought to be straightforwardly compositional. These devices mean that the grammatical descriptions involved lack, to varying extents, the important property of reversibility: they can only be used to analyse, not to generate, expressions of comparison. This is a serious restriction on the practical usefulness of such analyses.

The analysis presented here of the syntax and compositional semantics of the main instances of the English comparative and superlative is intended to provide a fairly theory-neutral 'off the shelf' treatment which can be translated into a range of current grammatical theories. The main theoretical claim is that by factoring out the compositional properties of the construction from the various types of ellipsis also involved, a cleaner treatment can be arrived at which does not need any machinery specific to this construction. A semantics in terms of generalised quantifiers is proposed.

SYNTAX

Intuitively, a sentence like:

John owns more horses than Bill owns

seems to consist of two sentences ascribing ownership of horses, together with a comparison of them, where some material has been omitted. Despite appearances, however, this pre-theoretical intuition is almost wholly wrong, both syntactically, and, as we shall see, semantically. The sequence 'more horses than Bill owns' is in fact an NP, and a constituent of the main clause, as can be seen from the fact that it can appear as a syntactic subject, and be conjoined with other simple NPs:

> [More horses than Bill owns] are sold every day
> John, Mary, and [more linguists than they could cope with] arrived at the party

In order to accommodate example like these we must analyse the whole sequence as an NP, with some internal structure approximately as follows. (We use a simple unification grammar formalism for illustration, with some obvious notational abbreviations).

```
NP[-comp] -> NP[+comp,postp=P,<feats=R>]
             S'[+comp,postp=P,<feats=R>]
```

A [+comp] NP is one like:

> a nicer horse, a less nice horse, less nice a horse, several horses more several more horses, as many horses, at least 3 more horses, etc.

We will not go into details of the internal structure of these NPs, other than to require that whether the comparative element is a determiner or an adjective, the dominating NP carries feature values which characterise it as a comparative NP, and which enforce 'agreement' between comparative pre- and post-particles (-er/than,as/as, etc.) via the variable 'P'. We assume that NPs marked as comparative in this way are not permitted elsewhere in the grammar.

In the case of the [+comp] S' constituent, there are several possibilities. Some forms of comparative can be regarded as straightforward examples of unbounded dependency constructions:

> ... more horses than Bill ever dreamed he would own _
> ... more horses than Bill wanted _ to run in the race

These involve Wh-movement of NPs. The second type involving a missing determiner dependency:

> John owns more horses than Bill owns _ sheep
> There were more horses in the field than there were _ sheep.

Rules of the following form will generate [+comp] sentences of this type, using 'gap-threading' to capture the unbounded dependency:

```
S'[+comp,postp=P, <feats=R>] ->
 COMP[form=P]
 S[-comp,gapIn=[CAT[<feats=R>]],gapOut=[]]
(where CAT is either NP or Det)
```

As well as these 'movement' comparatives are those involving ellipsis:

> John owns more horses than Bill/Bill does/does Bill/sheep.
> Name a linguist with more publications than Chomsky.
> He looks more intelligent with his glasses off than on.

It is noteworthy that sentences like the second of these demonstrate that the appropriate level at which ellipsis is recovered is not syntactic, but semantic: there is *no* syntactic constituent in the first portion of the sentence that could form an appropriate antecedent. We therefore do not attempt to provide a syntactic mechanism for these cases, but rather regard them as containing another instantiation of an `S'[+comp]` introduced by a rule:

```
S'[+comp] -> COMP S[+ellipsis, -comp]
```

An elliptical sentence is not a constituent required solely for comparatives, but is needed to account for sentence fragments of various kinds:

> John, Which house?, Inside, On the table, Difficult to do,
> John doesn't, He might not want to,
> etc.

All of these, as well as more complex sequences of fragments (e.g. 'IBM, tomorrow' in response to 'Where and when are you going?') need to be accommodated in a grammar.

Very many cases of this type of ellipsis can be analysed by allowing an elliptical S to consist of one or more phrases (NP, PP, AdjP, AdvP) or their corresponding lexical categories. Most other commonly occurring patterns can be catered for by allowing verbs which subcategorise for a non-finite VP (modals, auxiliary 'do', 'to') to appear without one, and by adding a special lexical entry for a main verb 'do' which allows it to constitute a complete VP. Depending, of course, on other details of the grammar in question the latter two moves will allow all of the following to be analysed:

> Will John?, John won't, He may do,
> He may not want to, Is he going to?
> etc.

With this treatment of ellipsis, our syntax will be able to analyse all the examples of comparatives above, and many more. It will also, however, accept examples like:

> John owns more horses than inside.
> Bill is happier than John won't.

for there is no syntactic connection between the main clause and the elliptical sentence. We assume that some of these examples may actually be interpretable given the right context: at any rate, it is not the business of syntax to stigmatise them as unacceptable.

Comparatives with adjectives and adverbial phrases, are, *mutatis mutandis*, exactly analogous to those with NPs, and we omit discussion of them here.

SEMANTICS

In the interests of familiarity the analysis will be presented as far as possible in an 'intensionless Montague' framework: a typed higher order logic.

Firstly, we need the notion of a generalised quantifier. It is well known that most, if not all, complex natural language quantifiers can be expressed as relations between sets. Specifically (Barwise and Cooper, 1981) a quantifier with a restriction R and a body B can be expressed as a relation on the sizes of the set satisfying R, and the set which represents the intersection of the sets satisfying R and B. A quantifier like 'all' can be represented using the relation =, and so a sentence like 'all men are mortal', in a convenient notation, will translate as:

quant(λnm.n=m,λx.man(x),λx.mortal(x))

(In logical forms, lower case variables will be of type e, and upper case variables will be of type e→t unless indicated otherwise. All functions are 'curried': thus λxy.P is equivalent to λxλy.P. Read expressions like 'quant(Q,R,B)' as 'the relation Q holds between the size of the set denoted by R, and the size of the set denoted by λx.Rx&Bx'. This latter is the intersection set.

The important thing to note at this point is that the relation Q can be arbitrarily complex, as it needs to be in order to accommodate determiners like 'at least 4 but not more than 7'. The second important thing to notice is that for many quantifiers, we are only interested in the size of the intersection set, and thus the first lambda variable in Q will be vacuous. Thus 'some' can be expressed as the relation 'λnm.m $\geq$ 1', as in 'some men snore':

quant(λnm.m $\geq$ 1, λx.man(x),λx.snore(x))

In the case of the movement types of comparative we can give the semantics in a wholly compositional way by building up generalised quantifiers which contain the comparison. Informally, the gist of the analysis is that in a sentence like 'John owns more horses than Bill owns', there is a generalised quantifier characterising the set of horses that John owns as being greater than the set of horses that Bill owns. Informally, we can think of the complement of a comparative NP as a complex determiner:

John owns [more than Bill owns] horses

(In this respect, as in the use of generalised quantifiers, this analysis yields logical forms somewhat similar to those of Rayner and Banks, 1989).

To build these quantifiers we assume that the various relations signalled by the comparative construction are part of the quantifier. Thus the final analysis of the example sentence is:

quant(λnm.more(m,
λx.horse(x)& own(Bill,x)),
λy.horse(y),λz.own(John,z))

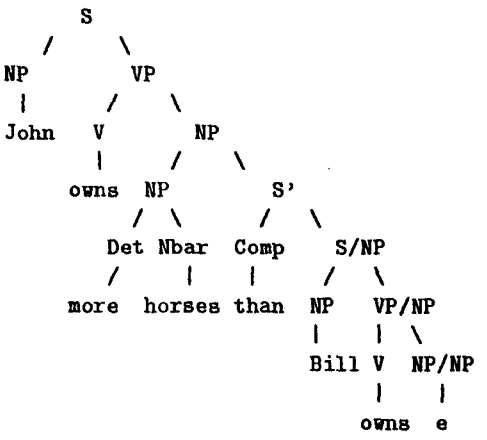
'More' (or 'less' or 'as') is the relation used to build the quantifier. To avoid notational clutter we can assume that 'more' is 'overloaded', and can take as its arguments either a number, or an expression of type e→t, in which case it is interpreted as taking the cardinality of the set denoted by that expression. 'More' in fact takes a third argument, which is another quantifier relation. Thus the meaning of a sentence like 'john owns at least 3 more horses than Bill owns' would get a logical form like

quant(λnm.more(m,λab.b$\geq$3,
λx.horse(x)&own(Bill,x)),
λy.horse(y),λz.own(john,z))

The way to read this is 'the relation of being more (by a number greater than or equal to 3) than the size of the set of horses owned by Bill, holds of the set of horses owned by John'. Where this extra argument to 'more' is not explicit, we assume it defaults to 'greater than 0'. However, we shall ignore this refinement in the illustrations that follow).

Note that this quantifier is only interested in the intersection set: this is always true of comparative quantifiers.

We now give the meanings of each constituent involved in a couple of examples, along with the relevant rules, in skeletal form. We indicate the trail of gap threading using the 'slash' notation. For the purposes of this illustration we use the analysis of the semantics of unbounded dependencies from Gazdar, Klein, Pullum and Sag (1985): a constituent C containing a gap of category X is of type X→C. So given a tree of the form [A [B C]] which might normally have as the interpretation of A as B applied to C, the interpretation of a tree [A/X [B C/X]] would be 'λX.B(C(X))'. Since gaps themselves are analysed as identity functions this will have the right type.



The relevant rules and sense entries in schematic form are:

S → NP VP : NP(VP)
VP → V NP : V(NP)
NP → NP[+comp] S' :NP(S)
S' → Comp S/NP : λx.S(λP.P(x))
S' → Comp S/Det : λx.S(λPQ.P(x) & Q(x))
S/Gap → NP VP/Gap : λG.NP(VP(G))
VP/Gap → V NP/Gap : λG.V(NP(G))
NP/NP → ϵ : λN.N
NP/Det → Nbar : λD.D(Nbar)
NP → bill : λP.P(bill)
NP → Det Nbar : Det(Nbar)
Det → more :
λPQR.quant(λnm.more(m,
λx.Px & Qx),λy.Py,λz.Rz)
Nbar → horses : λx.horse(x)
V → owns : λNx.N(λy.owns(x,y))

'Gap' abbreviates either NP[-comp] or Det, and G is a variable of the appropriate type for that constituent. N is an NP type variable; D a Det type variable, as are their primed versions. Notice that comparative determiners and their NPs are of higher type than non-comparative NPs, at least for those analyses which analyse relative clauses as modifiers of Nbar rather than NP. Constituent meanings are assembled by the rules above as follows:

[NP+comp more horses]:
λQR.quant(λnm.more(m,
λx.horse(x)& Q(x)),
λy.horse(y),λz.R(x))

[VP/NP owns ϵ]:
λG.[λNx.N(λy.owns(x,y))]([λN'.N'] G)
= λG.λx.G(λy.owns(x,y))

[S/NP Bill owns ϵ]:
λG'.[λP.P(bill)]([λG.λx.G(λy.owns(x,y))] G')
= λG'.G'(λy.owns(bill,y))

[S' than Bill owns ϵ]:
= λx.[λG'.G'(λy.owns(bill,y)](λP.P(x))
= λx.owns(bill,x)

[NP [more horses][S' than Bill owns ϵ]:
λR.quant(λnm.more(m,
λx.horse(x) & own(bill,x)),
λy.horse(y),λz.R(z))

The remainder of the sentence is straightforward. The second example for illustration is:

John owns more horses than Bill owns _ sheep.

For the subdeletion cases, a fully compositional treatment demands a separate sense entry for 'more', since the Nbar of the NP in which 'more' appears does not occur inside the comparative quantifier:

λPQR.quant(λnm.more(m, λx.Qx),
λy.Py,λz.Rz)

We do not have to multiply syntactic ambiguities: the appropriate sense entry can be selected by passing down into the NP a syntactic feature value indicating whether the following S' contains an NP or a Det gap. Constituents are assembled as follows: remember that D has the type of ordinary determiners: (e→t)→((e→t)→t).

[NP/Det ϵ sheep]: λD.D(λs.sheep(s))

[VP/Det owns ϵ sheep]:
λD'.[λNx.N(λy.owns(x,y))]([λD.D(λs.sheep(s))]D')
= λD'.λx.[D'(λs.sheep(s))](λy.owns(x,y))

[S/Det Bill owns ϵ sheep]:
λD'.([D'(λs.sheep(s))](λy.owns(bill,y)))

[S' than Bill owns ϵ sheep]:
λx.[λD'.([D'(λs.sheep(s))](λy.owns(bill,y)))]
(λPQ.P(x) & Q(x))
= λx.sheep(x) & owns(bill,x)

[NP+comp more horses]:
λQR.quant(λnm.more(m, λx.Qx),
λy.horse(y),λz.R(z))

[NP more horses
than Bill owns ϵ sheep]:
λQR.[quant(λnm.more(m, λx.Qx),
λy.horse(y),λz.R(z))]
(λx.sheep(x) & owns(bill,x))
= λR.quant(λnm.more(m,
λx.sheep(x) & owns(bill,x)),
λy.horse(y),λz.R(z))

The final logical form for the whole sentence is:

quant(λnm.more(m,
λx.sheep(x) & owns(bill,x)),
λy.horse(y),λz.own(john,z))

ELLIPSIS

In order to explain our treatment of ellipsis, we need more about the actual logical forms produced compositionally for sentences. These are the 'quasi logical forms' (QLF) of Alshawi and van Eijck (1989), differing from 'resolved logical forms' (RLF) in several respects: they contain 'a_terms' representing the meanings of pronouns and other contextually dependent NPs; 'a_forms' (anaphoric formula) representing the meanings of sentences containing contextually determined predicates (possessives, compound nominals, 'have' 'do' etc); and 'q_terms' representing the meaning of other quantified NPs before the later explicit quantifier scoping phase (see Moran 1988). QLFs are fleshed out to RLFs via a process of contextually guided inference (Alshawi, 1990). Since ellipsis is clearly a contextually determined aspect of interpretation we extend the 'a_form' construct to provide a QLF for elliptical sentences, and treat the process of interpretation as akin to reference resolution for pronouns.

Take a sequence like (A) 'Who came?' (B) 'John'. We represent the meaning of the 'missing' constituent by an 'a_form' binding a variable of the appropriate type to combine with the meaning of the 'present' constituents to form an expression of the appropriate type for the S' constituent containing the ellipsis. Thus the meaning of the two utterances will be represented as:

past(come(who))
a_form(P,P(john))

One can think of 'a_form' as asserting that there is such a P: resolution finds that P. For consistency with the Montague notation we are using we will indicate an 'a_form' variable as a free variable: 'P(john)'.

The ellipsis resolution method uses a technique which is formally a restricted type of higher-order unification (Huet 1975). Ellipsis resolution proceeds in three steps. Firstly, we have to find a 'context', which in the case of intersentential ellipsis is the logical form of the preceding utterance. Next, one or more 'parallel' elements are found in this context. In the example above, it would be 'who'. This step is somewhat analogous to the establishing of pronoun antecedents, and may be similarly sensitive to properties like agreement, focus, sortal restrictions, etc. When the parallel element(s) have been found, the next step abstracts over the position(s) of the element(s), and suggests the result as a candidate for P. In this example the only possibility is that P = λx.past(come(x)). Thus the meaning of the elliptical sentence after resolution is:

[λx.past(come(x))](john)
= past(come(john))

The theoretical advantages of higher-order unification in the interpretation of ellipsis are amply documented in Dalrymple, Shieber, and Pereira (forthcoming). More details of our own treatment are in Alshawi et al. (forthcoming).

This analysis of inter-sentential ellipsis generalises cleanly to intra-sentential ellipsis, in particular the comparative cases discussed above: the only difference is that location of the 'context' is not trivial, since the ellipsis is, as it were, contained in the logical form that yields the context. As an example, the NP in 'Name a linguist with [more publications than John]' will have a structure:

[NP [NP more publications] [S' than
[S+elliptical [NP John]]]]

The meaning of the elliptical S will be as above, but the appropriate version of the semantics for the S' rule will (as was the case with the analysis of the movement comparatives given earlier) have to arrange things so that the type of the whole elliptical S' expression is e→t. Thus the variable representing the ellipsis will be of type e→(e→t), assuming that 'john' in this context is of type e. Omitting some of the details, the meaning of the entire NP will then be:

λR.quant(λnm.more(m,
λx.publications(x) & [P(john)](x)),
λy.publications(y), λz.R(z))

where the meaning of the elliptical S' [P(john)] figures in the second term of the comparison after beta-reduction. The meaning for the whole sentence, again taking some short cuts will be:

name(hearer,linguist) &
quant(λnm.more(m,
λx.publications(x) & [P(john)](x)),
λy.publications(y), λz.have(linguist,z))

We now have to find a suitable context for ellipsis resolution. The only candidate expression with an element parallel to 'john' is 'λz.have(linguist,z)'. Abstracting over the parallel element gives us 'λlz.have(l,z)', which is an appropriate candidate for P. After substituting and reducing the final meaning of the whole sentence will be:

name(hearer,linguist) &
quant(λnm.more(m,
λx.publications(x) & have(john,x)),
λy.publications(y), λz.have(linguist,z))

In reality, of course, the details are more complex than this, but this semi-formal reconstruction should convey the basic principles. Now we have succeeded in analysing all the types of comparative so far discussed using either purely compositional means, or a non-compositional device for contextual interpretation of ellipsis whose main properties, however, are motivated on grounds other than its use for comparatives. Furthermore, once we have this type of ellipsis mechanism in place, it is a simple matter to extend it to account for comparatives in which the whole comparison is missing:

John has 2 more horses.
There are at least as many sheep.

As Rayner and Banks somewhat ruefully note, these are in many texts by far the most commonly encountered form of comparative, although their analysis, in common with others, fails to handle them.

Syntactically, what we do is to give the various comparative morphemes an analysis in which they are marked as [-comparative]. Thus a phrase like 'at least as many sheep' will be analysed as either a + or - comparative NP. In the first case, the syntax will only permit it to occur with an explicit complement, as detailed above, and in the second case the syntax will prevent an explicit complement occurring. Semantically, however, the second contains an elliptical comparison. Thus the meaning of 'more' in this type of comparative will be:

λPQ.quant(λnm.more(m,
λx. P(x) & R(x)),
λy.P(y),λz.(Q(z))

where R represents the meaning of the missing constituent. In a context where 'John has more horses' follows a sentence like 'Bill has some horses', R should be resolved to 'λa.have(bill,a)'. Notice that it may be necessary to provide interpretations for 'more' in these contexts corresponding to both the NP-gap and the Det-gap cases: the elliptical portion is different depending on whether the preceding sentence was 'Bill has some horses' or 'Bill has many sheep': the latter is like the Det-gap type of explicit comparison.

IMPLEMENTATION STATUS

Morphology, syntax and compositional semantics for NP, AdjP and AdvP comparatives of both movement and ellipsis types have been fully implemented, as well as some other common types of comparative not mentioned here (e.g. Nbar comparatives like 'more men than women'). Ellipsis resolution has been implemented for the inter-sentential cases, but not, at the time of writing, for the intra-sentential cases. However, we foresee no problem here, as this is an extension of existing mechanisms.

ACKNOWLEDGEMENTS

This work was supported by the CLARE consortium: BT, BP, the Information Engineering Directorate of the DTI, RSRE Malvern, and SRI International. I thank Hiyan Alshawi for his many substantial contributions to the analyses described here, and Jan van Eijck and Manny Rayner for comments on an earlier draft.

REFERENCES

Alshawi, H. (et al.) forthcoming 'The Core Language Engine', MIT Press.
Alshawi, H. (1990) Resolving Quasi-Logical Forms, Computational Linguistics 16.
Alshawi, H. and van Eijck, J. (1989) Logical forms in the Core Language Engine, Proceedings of 27th ACL, Vancouver: ACL
Ballard, B. (1988) A General Computational Treatment of Comparatives for Natural Language Question Answering, in Proceedings of 26th ACL, Buffalo: ACL
Barwise, J. and Cooper, R. 1981 Generalised Quantifiers and Natural Language, Linguistics and Philosophy, 4, 159-219
Gazdar, G., Klein, E., Pullum. G. and Sag, I. (1985) Generalised Phrase Structure Grammar, Oxford: Basil Blackwell
Huet, G. (1975) A Unification Algorithm for Typed Lambda Calculus, Jl. Theoretical Computer Science, 1.1, 27-57.
Moran, D. B. (1988) Quantifier Scoping in the Core Language Engine, in Proceedings of 26th ACL, Buffalo: ACL
Rayner, M. and Banks, A (1989) An Implementable Semantics for Comparative Constructions, Computational Linguistics, 16.2, 86-112
Dalrymple, M., Shieber, S., and Pereira, F. (forthcoming) Ellipsis and Higher Order Unification, Linguistics and Philosophy.