UCL Working Papers in Linguistics 8 (1996)

S-dependency[*]

AND ROSTA

Abstract

Of the grammatical relations (dependencies) that relate words syntagmatically, some — in 'movement' constructions — but not all determine the order of the relata. These are **s-dependencies**, dependencies determining sequence; they also have additional functions, such as regulating what can extract and where it can extract to, and defining subordinacy. From some but — in the case of coordination and pied piping — not all s-dependencies is built the surface constituency structure that effects continuity of phrases.

1 Syntax

This paper outlines and motivates certain developments in Word Grammar's theory of syntax.[1] The grammatical locales we call in on on the way include personal pronoun 'case' forms, extraction and extractability, prepositional passives, coordination and pied piping. But my goal is not so much to delve into data as to extend the basic apparatus of the theory.

---

[*]Thanks to Dick Hudson for comments on this and earlier drafts. Author's email address: a.rosta@ucl.ac.uk

[1]On Word Grammar, see Hudson 1990, 1994a. As far as syntax goes, the key tenets of Word Grammar (WG) are (i)-(iv).

(i) Grammar is a list of constraints on potential utterances.
(ii) Syntax is monostratal.
(iii) All nodes are terminal. (Hudson's (1984, 1988, 1990) version of WG allows non-terminal nodes solely to deal with the grammar of coordination. But as I explain in §4, this is unnecessary.)
(iv) Grammatical relations are basic (rather than derived).

(i) and (ii) are important in defining the tenor of the theory. (iii) is a constraint whose particular importance is in defining WG's salient differences from other theories, since most other theories still reject (iii) or at least have not volubly embraced it. (iv) is what the bulk of the syntax component of the grammar is concerned with: most rules state which words can be related to which words, by which relationships.

In this section I introduce three kinds of relationship, each of such a generality that every word (in a syntagm of more than one word) is an argument of each kind of relationship. They are 'associacy', 'dependency' and 's-dependency': they are relationships such that one of their arguments is, respectively, 'associate', 'dependent' and 's-dependent' of the other.[2] Associacy, we'll define with extreme generality as any grammatically significant syntagmatic relationship between words.[3] To avoid unnecessary distractions, I shan't define 'dependent'[4] and instead will use it to mean 'subject or object or adjunct or extractee or one of various other kinds of grammatical relation'.

To s-dependency, the rest of the paper is devoted. The 's' of s-dependency can be considered an abbreviation of a variety of terms – **surface** dependency, **subordinating** dependency, **sequential** dependency, **structural** dependency, **scaffolding** dependency, **skeletal** dependency... — the appropriacy of each of which I'll explain.

To start with, a good way to get an initial intuitive notion of the role of s-dependency in syntax is to imagine that WG was polystratal: dependencies that are s-dependencies would obtain at 'surface structure' and dependencies that are not s-dependencies would not. That gives us 's' for 'surface'.

Second, it is s-dependency that gives us the grammatical notion of subordination (the next 's'). This is because s-dependency imposes upon the syntagm a hierarchical structure: that is, if in diagrams (I–V) the arrows indicate s-dependencies (from s-head to s-dependent), then (I) and (II) and, as explained in §4, (III) are possible structures, but (IV)

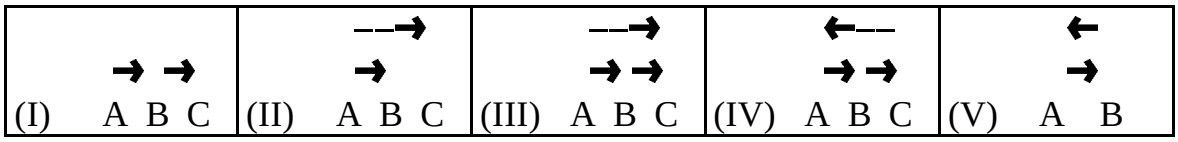
---

[2]The converse of 'Dependent' is 'Head' and converse of 'S-dependent' is 'S-head'. Some recent work, e.g. Hudson (1995b, 1995c), has established 'Parent' and 'Anchor' as synonyms of 'Head', the term 'Head' being disfavoured because of its significantly different meaning in Phrase Structure Grammar. The present 'Dependent' corresponds in practise quite closely to Hudson (1990)'s 'Dependent', while the 'Dependent' of Dependency Grammar is more like a conflation of Dependent and 'Niece' (on which, see below). The 'Dependent' of Rosta (1994) has bifurcated into S-dependent and Niece. Rosta (1994)'s 'Grammatical Relation' ('GR') has bifurcated into Associate and Dependent. Rosta (1994)'s allocation of terms had some unfortunate consequences. First, it disagreed with Hudson's usage and precedent: he has always used 'dependency' and 'grammatical relation' pretty much synonymously. Second, my use of 'GR' to denote not the relationship but the words related by it was a source of mild confusion. It is for these reasons that I have executed a terminological about turn.

[3]Associacy may or may not be the same as Hudson (1990)'s Companion relationship. If X is head of Y or X is dependent of Y then X is companion of Y, which is compatible with Companionship being Associacy, but it is not clear whether if X is companion of Y then X is head of Y or X is dependent of Y, which is not compatible with Companionship being Associacy.

[4]Dependency still remains undefined in WG.

and (V) are not, while if the arrows indicate dependencies, then each of (I–V) is possible.[5] An example of the need for subordination is discussed in §3.5.



Third, s-dependency is above all a relationship of linear sequence (the 's' for 'sequential'). Every word takes its position from each of the words it is s-dependent of. For example, *it* in *She tried to dispose of it* takes its position from and is s-dependent of *of*, while *what* in *What did she try to dispose of?* takes its position from and is s-dependent of not *of* but *did*, and *what* in *He wondered what she tried to dispose of* takes its position from and is s-dependent of *wondered*. This is discussed in §3.1. A further role of s-dependency in word order is that from some but not necessarily all s-dependencies is built the skeletal tree responsible for continuity (of phrases), and in part for processing complexity and for intonation phrasing. This is discussed in §2 and again in §4. It is these functions of s-dependency that give us 's' for 'skeletal', 'structural', 'scaffolding'.

S-dependency also provides the means of stipulating (i) linear precedence restrictions (§3.1, as already mentioned), (ii) that dependencies cannot unconstrainedly hold between any pair of words that in themselves are eligible to be linked by a dependency (§3.2), (iii) when words must 'remain in situ' and when they are 'movable' (§3.3), (iv) permitted 'landing sites' for extraction (§3.4), and (v) that 'movement' is 'upward' (§3.5).

2 Skeletal trees

2.0 Outline

§2.1 explains how every sentence must contain a skeletal tree. §2.2 explains how the tree matches up with s-dependencies. §2.3 considers the role of the skeletal tree in grammar.

[5]Compare how, in a family tree, (I) shows A as parent of B and B as parent of C, (II) shows A as parent of B and of C, and (III) shows A as parent of B, B as parent of C and A as parent of C, which is possible though incestuous. In contrast, no family, however unorthdox, can have structure like (IV) or (V).

2.1 Projectivity and skeletal constituency

In (1a), *the* precedes *went*, as subjects should, and *children* follows *the*, as complements should. In (1b), *book* properly follows *the*, and *the* and *tomorrow* properly follow *read*.[6] Yet the word order of (1a–b) is ungrammatical. Why?

```
          S   of
          of      C
(1)  a. * The went children.

          of   O
          of      A
          .    of          C
     b. * Read the tomorrow books.
```

The solution, put forward in Hudson (1994b) and Rosta (1994), is that for every sentence it must be possible to construct a 'skeletal tree' such that to every branch in the tree there corresponds a dependency, with the lower node being dependent of the higher. The nature of this correspondence is described in §2.2. In this section we'll look at other properties of skeletal trees, namely (2a–d). (2a) is merely an assumption; it could be abandoned without consequence to the other restrictions on skeletal trees.

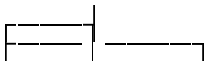
(2) a. There is one node per word. (In the rectilinear diagrams I'm using, a vertical line is equivalent to a node, and taller verticals are superordinate to shorter verticals.)
b. There is exactly one tree per sentence.
c. Branches don't converge.
d. Branches don't cross.

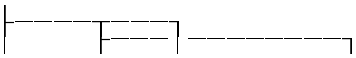
(2c–d) are a statement of the principle of projectivity, which most versions of Dependency Grammar adopt in some form or other. (2d) is a requirement of continuity; it is equivalent to a prohibition against discontinuous constituency – a point which will

[6]The diagrams show *the* as subject ('s') of *went* and *children* as complement ('c') of *the* in (1a), and *the* as object ('o') of *read*, *books* as complement of *the* and *tomorrow* as adjunct ('a') of *read* in (1b). Other abbreviations used in other diagrams are 'x' for 'xcomp' (predicative complement), 'e' for 'extractee'/'extraposee', 'k' for clausal (see Rosta 1994 on clausals), and 'd' for 'dependent'.

be made use of shortly.

(1a–b) yield the skeletal trees shown in (3a–b). The branches of these trees cross, and hence the sentences are ungrammatical.

(3) a. 
* The went children.

b. 
* Read the tomorrow books.

Let us proceed, then, to formalize projectivity. Constraint (2b) is formalizable as rule (4), the effect of which is that for any two words linked by a dependency, one is subordinate of the other, or both are subordinates of a third (that is, node *z* dominates nodes *x* and *y*, or node *z* is one of nodes *x* and *y* and dominates the other);[7] it is assumed that every word in a sentence is head or dependent of another word in the sentence, and not head or dependent of any word not in the same sentence.

(4) **One tree per sentence.**
If *x* is dependent of *y* then ∃*z* *z* is unsubordinate of *x* and *z* is unsubordinate of *y*. (An unsubordinate of W is W or a superordinate of W. A superordinate of W is an s-head of W or a superordinate of an s-head of W.)

And then we need rules (5a–b). (5b) ensures that for every word except the root of the tree, there is exactly one word that is its aunt; this will be made use of in §2.2. (5a) gives us the ban against convergence.

(5) a. Every word is daughter of exactly one phrase.
b. Every phrase contains exactly one word and zero or more phrases.

As for the ban on crossing, i.e. the requirement of continuity, I would argue that the use of constituency means we need no stipulation of continuity specifically for syntax. That skeletal constituent structures can be derived from a projective relational structure, and vice versa, is hardly news. It has long been held[8] that a projective dependency

[7]Provisionally, we can understand a subordinate of W as a word dominated by W in the skeletal tree. Formally, a subordinate of W is an s-dependent of W or a subordinate of an s-dependent of W.

grammar and a phrase structure grammar are equivalent, in the sense that each can be derived from the other.[9] However, that constituent structures and projective dependency structures are each derivable from the other does not mean they are equivalent in all respects, and nor does it mean there is no point in building constituent structures off dependency structures. With constituency, continuity comes for free: continuity may reasonably be considered inherent (at least by default) in phrase structure. This is because the rule that parts and wholes, of which phrases are a subtype, are continuous applies throughout the grammar,[10] and in fact could be inherited from extragrammatical knowledge – a leg, for example, could not contain a thigh and a shin but not a knee, and a collection of a finger, a nose and a buttock would not be conceptualized as a single body part. Continuity of wholes must ultimately be stipulated, but only at a broader level of generality than syntax. Thus, if one wished to permit discontinuity (as, for example, McCawley (1982, 1989) does), then either one must stipulate construction-specific possibility of discontinuity, or one must reinterpret constituency as based not on the part–whole relationship but instead on the member–sequence relationship and accordingly be obliged to stipulate default continuity. Without recourse to constituency, continuity of projective dependency structure must be stipulated. This can be confirmed by considering extralinguistic analogues. Imagine a gathering of men, where each man is either father or son of another man present. There would therefore be a non-looping convergenceless structure of relationships, but it would certainly not necessarily be the case that they form a continuous structure such that each man is physically separated from his father by none but his brothers or his sons.

---

[8] At least since Gaifman (1965), Robinson (1970), but probably earlier by anyone who gave the matter a moment's thought. See Fraser (1989).

[9] The equivalence holds only if the dependency structures are projective. The dependency structures of WG certainly aren't equivalent to phrase structures. Projective dependency structure is equivalent to unlabelled bracketing if a phrase can't be mother of more than one word, and is equivalent to headed phrase structure if a phrase can contain more than one word. See the discussion in Fraser (1989, 1990). It is the mistaken assumption that dependency grammars necessarily impose the requirement of projectivity, or at least prohibit bicipitality (double headedness) and interdependency that is responsible for the myth that DGs and PSGs are 'notational variants'.

[10] There are possible exceptions, such as complex Semitic-type nonconcatenative morphology, where morphemes are intercalated. But all that shows is that constituency (understood as based on part–whole relationships rather than member–sequence relationships) is not the proper basis for analysing such phenomena.

From here on, then, 'skeletal tree' will be taken to be synonymous with 'constituency structure' and 'phrase structure'.

2.2 Correspondence between dependency, s-dependency and constituency

In this section we will see how the skeletal tree — i.e. the rudimentary constituent structure — corresponds to dependency structure. Recall that (5b) stated in effect that every word either is the root of the skeletal tree or is niece of exactly one word. The exceptionless rules (6a–c) define the basic correspondence between constituency structure (i.e. the skeletal tree), s-dependency structure and dependency structure: s-dependency mediates between constituency structure and dependency structure, in that all the aunt–niece relatronships must correspond to some of the s-dependency structure, and some of the s-dependency structure must correspond to some of the dependency structure.

(6) a. Without exception, if *x* is niece of *y* and *x* is instance of Word and *y* is instance of Word then *x* is s-dependent of *y*.[11]
b. Without exception, if *x* is s-dependent of *y* then *x* is dependent of *y*.
c. Therefore: without exception, if *x* is niece of *y* and *x* is instance of Word and *y* is instance of Word then *x* is dependent of *y*.

The default rules are (7a–c), which require dependency structure, s-dependency structure and constituency structure (again construed as aunt–niece relationships) to be the same.[12]

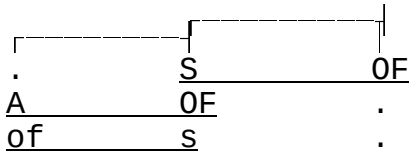
[11]X is niece of Y in bracketings [[X] Y] and [Y [X]].

[12]In ancestral forms of the present analysis I did not employ these default rules. I assumed that the only constraint upon the correspondences between constituency, s-dependency and dependency are those stated by the exceptionless rules. Thus any constituency structure at all was possible for a sentence, so long as for each word the word that is its aunt was its s-head, and any s-dependency structure at all was possible, so long as each s-dependency corresponded to a dependency. Similarly, any dependency structure was okay if it yielded an okay s-dependency structure (and if the individual dependencies satisfy the constraints that most of the grammar is taken up by stating). The best evidence against this view and in favour of there being the default rules is the prohibition against 'promiscuous dependency', discussed in §3.2.

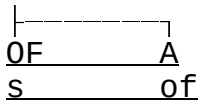
(7) a. By default, if *x* is dependent of *y* then *x* is s-dependent of *y*.
b. By default, if *x* is s-dependent of *y* then *x* is niece of *y* and *x* is instance of Word and *y* is instance of Word.
c. Therefore: by default, if *x* is dependent of *y* then *x* is niece of *y* and *x* is instance of Word and *y* is instance of Word.

The exceptions to (7a–b) are construction-specific and licensed by rule. The exceptions to (7b) are discussed in §4, and are stated there in just one rule, which states exactly when a word can have more than one s-head and how in such cases all but one of its s-heads are not its aunt. But until we reach §4 we'll be examining only s-dependents of W that are also nieces of W.

The exceptions to (7a) are greater in number, though still few. They allow dependencies that exceptionally aren't s-dependencies to occur in constructions involving adjuncts, as in (8a–b), raising, (8c–d), extraction, (8e), extraposition, (8f), and passive, (8g).[13] Dependencies that correspond to s-dependencies are capitalized. The constituency structures are shown by (rectilinearized) stemma and by bracketing.

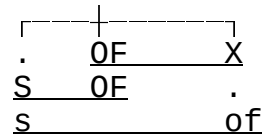


(8) a. [[[Smiling] students] waved].

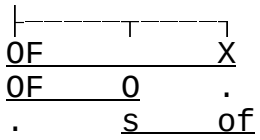


b. [Anyone [smiling]] will be reprimanded.

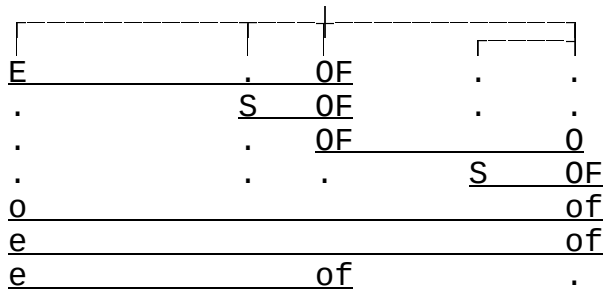
[13]This list is not necessarily complete, though for English there aren't any obvious additions to be made. A possible candidate is preposed adjuncts, as in *On Tuesday she may have been visiting him*. Outside English there are further constructions, such as clitic climbing, as in Italian *Io lo voglio leggere* ('I it want to read' – 'I want to read it'), and, in partial VP-fronting in German, what Hudson (1995b) has called 'universal raising'.



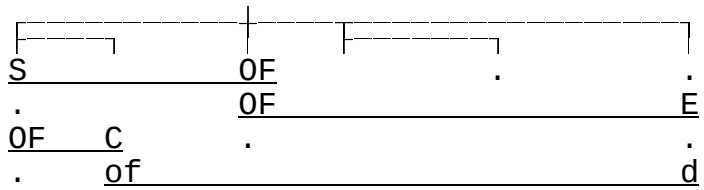
c. [[It] keeps [raining]].



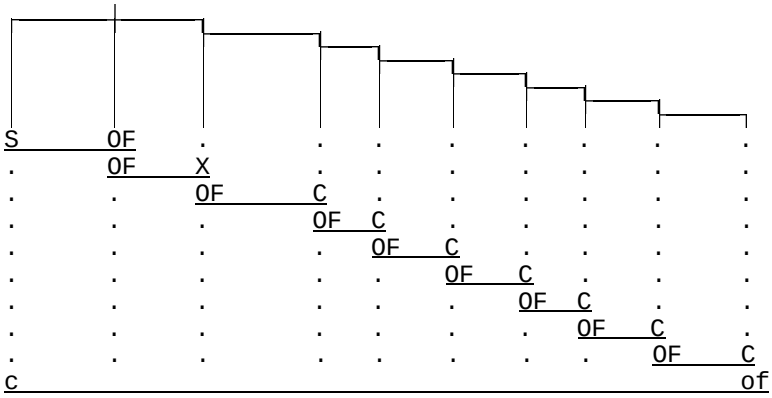
d. [Make [it] [rain]].



e. [[*Sordello*], [he] knows [[she] read]].



f. [[The [word]] had [passed [around]] [that [the colt from old Regret had got away]]].

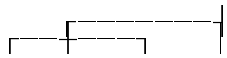


g. [[Doors] were [marked [on [the [top [of [the [left [of]]]]]]]]].

2.3 The role of skeletal trees

What job does phrase structure do? Most obviously, it corresponds to s-dependency structure and thence to dependency structure, and it effects contintuity. But apart from this hugely important function, it appears to do nothing else in syntax in particular and very little else in grammar in general. It appears that to some significant extent it is the skeletal tree that determines intonation phrasing (cf. Taglicht 1994, 1995), but the nature of the formal rules for this is not clear. Also, it may be that skeletal branches figure in rules for the form of instances of the lexemes ME, US, HIM, HER and THEM.[14] For example, instances of the lexeme ME, must be pronounced *I* when sole subject of a finite: *I will, *Me will*. But when the ME is one of two or more subjects, it can have the form *me*: *Me and him will, Him and me will*.[15] There are several ways to formally describe these facts. For example, the key factor could be whether the pronoun is a dependent of a conjunction: the rule would be that by default every instance of ME must be pronounced *I*, but, exceptionally, not every ME that is dependent of a conjunction need be pronounced *I*. But there is a much tidier description available, so long as we can refer to dependencies that are branches in the skeletal tree, and so long as the skeletal tree shown in (9) obtains (as indeed I assume it does – see §4.1): a ME can be pronounced *I* when subject of a finite, and must be pronounced *I* when it is subject of and niece of a finite.

(9)



But outside the grammar, skeletal trees again seem to play an important role, in usage – in matters of style and processing. Hudson (1995c) has shown that two primary determinants of the processing difficulty of a sentence are (i) the number of words that a branch passes above (Hudson's 'dependency distance'), and (ii) the number of branches that pass above a word (Hudson's 'dependency density'). The more words that a branch crosses, and the more branches that cross over a word, the harder the sentence is to process.

---

[14]See Hudson (1992, 1995a) on these lexemes and English caselessness.

[15]There is of course tons of idiolectal and dialectal variation in the grammar of pronoun forms. I'm here reporting a bit of my own lect, which is not unusual in this respect.

3 S-dependency

3.0 Introduction

The function of s-dependency that we've seen so far is that it mediates between dependency structure and phrase structure. This has been described in §2. This section outlines further functions of s-dependency. §3.1 describes s-dependency's function in linear precedence rules. §3.2 explains the role of s-dependency in limiting which pairs of words a dependency can potentially hold between. A further function of s-dependency is that it constitutes a grammatical phenomenon, 'remaining in situ', as an alternative to and on a par with extraction, extraposition, promotion, and so on: §3.3 provides a partial illustration of this by showing how the extractability of dependents of W can follow from whether the dependent must be s-dependent of W. §3.4 discusses the role of s-dependency in statements of rules concerning where extracted words end up slotting into the skeletal tree. §3.5 shows, in the context of a constraint on prepositional passives, how s-dependency allows us to capture notions that we might describe, in transformational metaphor, as 'movement must be upwards'. Of all these functions attributed to s-dependency in §§2–3, none can be handled in WG without recourse to s-dependency, except perhaps in rules pertaining to unextractability, such as we meet in §3.3.

3.1 Linear precedence

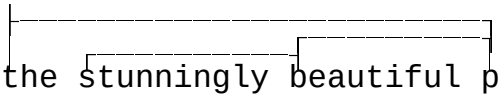
The sorts of word order restrictions that continuity accounts for are holistic in nature. Even though syntax consists mostly of pairwise relationships between words, no sentence in which every word but one has exactly one head can be discontinuous solely because of the relative position of a head–dependent pair. In such a structure, discontinuity is due to the relative position of at least two head–dependent pairs. This is of course not all there is to the grammar of word order. Besides continuity, we need rules governing the relative order of codependents and rules governing the relative order of head–dependent pairs. Examples of codependents whose order is fixed are the italicized words in (10–11) (on examples like (11), cf. Crystal (1971: 126 ff.)).

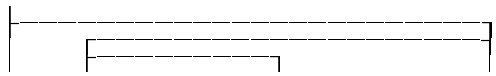
(10) a. He bet *her five* pounds *that* it would.
b. *He bet *her that* it would *five* pounds.

(11) a. all the very same *favourite great big old red Dutch cardboard picture* books
b. *all the very same *Dutch red old big great favourite picture cardboard* books
c. *all the very same *cardboard picture favourite great big old red Dutch* books

It remains a matter of some uncertainty what sort of rules we need in order to effect these orderings, and whether we need to stipulate the orderings explicitly. The other sort of ordering rules, pertaining to head–dependent pairs, will be the focus of this section.

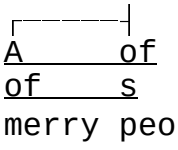
The grammaticality difference between (12a–b) cannot be explained in terms of continuity — both are continuous, and it makes no difference to the structure's continuity whether *stunningly* precedes or follows *beautiful*.

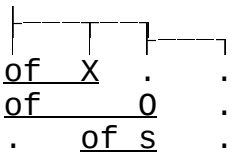
(12) a.


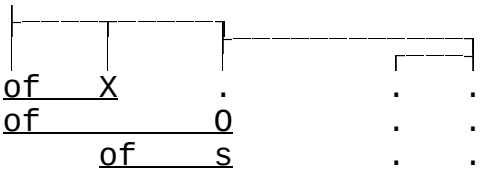
b. *


In order to account for the ungrammaticality of (12b), we need a rule requiring *stunningly* to precede *beautiful*, rather than vice versa. There are many such rules in Hudson (1990), stating, for example, that a complement of W follows W, a subject of W precedes W, and so on.

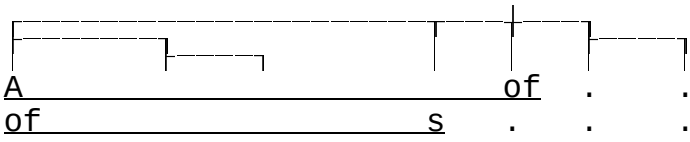
The problem with such rules is that they clash. Consider various exceptions, as in (13a–d), to the generalization that the subject of W precedes W. (13a) obeys the rule that attributive adjectives precede the noun; (13b–c) obey the rule that complements of W follow W. (13d) obeys the rule that absolute adjuncts of W may (must?) precede W.

(13) a.


b.




c. Make happy everyone you meet.



d. Pleased with herself, she ate some chocolate.

In WG hitherto, one would have had to override the default subject order rule for each of these constructions. That strategy amounts merely to stipulating the ordering for every possible construction. But note that these exceptional subjects in (13a–d) are also not s-dependents of the word they're subject of. This is the key to the problem.

S-dependency, I maintain, is crucially involved in the grammar of head–dependent orderings. The grammar is sensitive to the relative order of two words only if one is s-dependent of the other. This is stated in (14), using 'Precessor' and 'Successor' to denote relationships of precedence and succession. (Note that Precessor and Successor are not converses: it is not possible for X to be precessor of Y and Y to be successor of X, since that would require each to be s-dependent of the other, which is prohibited by constraints on s-dependency loops stated in §4.1.)

(14) Without exception, *x* is s-dependent of *y* iff *x* is precessor of *y* or *x* is successor of *y*.

The intention is that Precessor and Successor are the only relationships of sequence that the grammar has access to, at least in so far as their arguments are words.

(13a–d) turn out not to be exceptions to the word order rules pertaining to subjects, since the actual rule is (15): the reason (13a–d) are not exceptions is that although *people, the* and *everyone* are subject of, respectively, *merry, on* and *happy*, they are not s-dependent of the word that they're subject of.

(15) If *x* is subject of *y* and *x* is s-dependent of *y* then *x* is precessor of *y*.

The word order in (13a–d) will therefore follow from the rules that pertain to the s-dependencies in these sentences; in (13a), for instance, *merry* precedes *people* because *merry*, as an attributive adjunct and s-dependent of *people*, is precessor of *people*.

I am not aware of any exception to (15), save perhaps for *the more fool she* and a handful of similar examples.[16] There is not, however, a general presumption that such rules will be exceptionless. Indeed, by stating (16) as a default rule and accepting that there will be exceptions to it, we actually end up requiring very very few word order rules.

(16) By default, if *x* is s-dependent of *y* then *x* is successor of *y*.

The only word order rules (for head–dependent pairs) we'll need are those that are exceptions to (16) (or those that are exceptions to the exceptions, if there are any). We therefore need just a small handful of rules to cover orderings where dependents are precessors: subjects, extractees, preposed adjuncts, the relative sequence of *the* and *'s* in *the boy's*, and the position of 'quantitatives' (a.k.a. 'measure modifiers') of non-verbs, as in *three miles wide*, *three feet under*, *three years ago*. A further consequence of (16) is that virtually all clashes between word order requirements are resolved automatically by default overriding (i.e. the elsewhere principle). For example, in *be here*, *here* follows *be* because it is s-dependent and therefore, by default, also successor of *be*, while in *here am I* the rule requiring an extractee and s-dependent of W to be precessor of W automatically overrides (16) by virtue of its conditions being a subset of (16)'s. Any need for further explicit rules to resolve conflicts between word-order rules would arise only in a construction in which an s-dependent that is normally a precessor is actually a successor. At present I'm not aware of any clear examples of that happening.[17]

3.2 No promiscuous dependency

What prohibits the asterisked dependencies in (17a–b)?[18]

---

[16]In inversions like *Did she* and *Here shall remain every man*, I take *she* and *every* to be s-dependent but not subject of the auxiliaries.

[17]So-called subject-auxiliary inversion is the absence of object-to-subject raising, not postposing of subjects or preposing of auxiliaries.

[18]This question was brought to my attention by Hazel Clark.

```
                a     of
               *a                          of
(17)  a.  The naked minstrel endeavoured to play a saraband.

                                    of             a
               *of                                 a
      b.  The minstrel endeavoured to play a saraband naked.
```

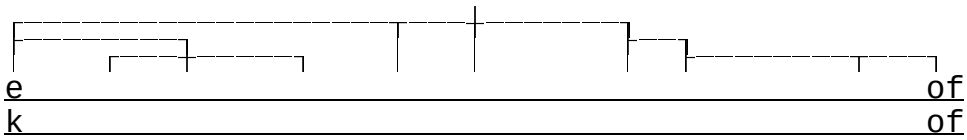
To the rescue comes rule (7a), which says that by default, every dependent of W must be s-dependent of W. Exceptions are permitted for certain constructions, such as extraction (for examples, see §3.3), but there are no rules stating exceptions that would allow examples like (17a–b). The effect of (7a) is to prevent dependencies from rambling freely throughout the sentence, engineering liaisons between any old pair of words. Instead, all dependencies must be s-dependencies (and therefore, by default, skeletal branches) unless permission is expressly granted for a dependency to be unaccompanied by an s-dependency.

3.3 Extractability

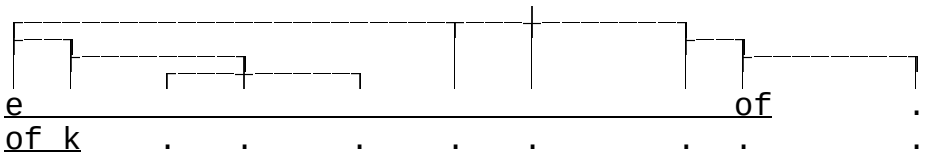
As we have seen in §2.2, by default if X is dependent of Y then X is s-dependent of Y. But rules permit various exceptions to that rule: if X is dependent of Y, then potentially there are a range of further relationships that may also hold between X and Y, and in general one of these relationships must apply. Simplifying, the options are as follows. X can also be extractee of Y – i.e. it is extracted. Or X can be extraposee of Y — i.e. it is extraposed. Or X can also be 'promotee' of Y — i.e. it is raised or passivized. Or, if none of these other options apply, X can be s-dependent of Y. Depending on the word class of X and Y, and the type of dependency that holds between them, only a subset of these options are available. For example, if X is adjunct of Y, and Y is a verb, then X will be extractee or s-dependent of Y, while if Y is not a verb, X will be its s-dependent. In this section I'll show how this model affords a means to deal with extractability.

I deal first with the unusual case of obligatory extraction, illustrated by the data in (18–20);[19] clausal complements (labelled 'k') of prepositions must extract.

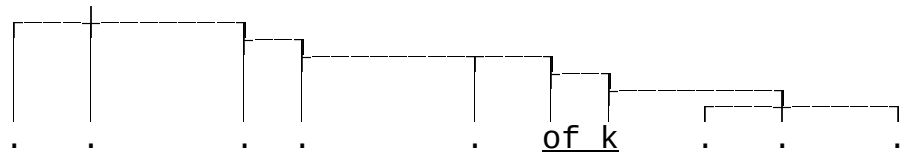
[19]The judgements (which I agree with) are from Kaplan & Bresnan (1982) and Postal (1994).



(18) a. That she loved him, she managed to convince him of.
b. That she loved him, she sought to casually inform him of.



(19) a. * Of that she loved him, she managed to convince him.
b. * Of that she loved him, she sought to casually inform him.



(20) a. * She managed to convince him of that she loved him.
b. * She sought to casually inform him of that she loved him.

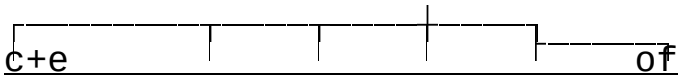
This obligatory extraction arises from rule (21), which allows only for (18a–b), in which *that* is clausal and extractee of *of*, and not for (19–20), in which *that* is clausal and s-dependent of *of*.

(21) If *x* is clausal of *y* and *y* is instance of Preposition then *x* is extractee of *y*.

Examples of obligatory unextractability (and more generally, 'remaining in situ') and optional extractability are given in (22–23) respectively. In (22a) *ticket* is complement and s-dependent of *the*, and in (22b) *ticket* is complement but not s-dependent of *the*. In (23a) *it* is complement and s-dependent of *read* and in (23b) *it* is complement and extractee and not s-dependent of *read*.

of c

(22) a. She just can't find the ticket.

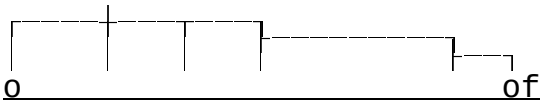


c+e of

b. Ticket, she just can't find the.

. . . of o

(23) a. She endeavoured to read it.



o of

e of

b. What did she endeavour to read?

(22a–b) could follow from (7a), the default whereby all dependencies are s-dependencies., in which case for (23a–b) we need (24a), which states an exception to (7a). (24b) is the rule that comes from combining (7a) and (24a).

(24) a. If *x* is complement of *y* and *y* is instance of Verb then *x* is s-dependent of *y* or *x* is extractee of *y*.

b. If *x* is dependent of *y* then *x* is s-dependent of *y* or [*x* is complement of *y* and *y* is instance of Verb and *x* is extractee of *y*].

Alternatively, it may be that in general complements in general are extractable (which is the position taken in Hudson (1990)). In this case, (25a), which allows for both (23a–b), is a fairly general exception to (7a), and we then need (25b), to prevent (22b), as a more specific exception to (25a). (25c) is the rule that combines (7a) and (25a–b).

(25) a. If *x* is complement of *y* then *x* is s-dependent of *y* or *x* is extractee of *y*.
b. If *x* is complement of *y* and *y* is instance of Pronoun then *x* is s-dependent of *y*.
c. If *x* is dependent of *y* then *x* is s-dependent of *y* or [*x* is complement of *y* and *y* is not instance of Pronoun and *x* is extractee of *y*].

S-dependency provides a basis for explaining the well-known extractability contrast (26a–b).

(26) a. Who do you reckon went?
b. *Who do you reckon that went?

The explanation requires us to follow Hudson (1984, 1990) in taking *that* in (27) to be complement of *know*, and *is* to be complement of *that*.

(27)
```
        of   c
        .    of      c
        know that it is
```

According to Hudson's analysis of extraction, a word can only be extractable by virtue of its subjecthood if it is the subject of a complement of a verb. In (26a) *who* is a subject of a complement (*went*) of a verb (*reckon*), so *who* is extractable. Assuming the dependency analysis in (27), in (26b) *who* is a subject of a complement (*went*) of a non-verb (*that*), so it is not extractable.

Hudson's explanation is in fact correct in its essence but wrong in its details. In cases where the subject in a finite clause is also a complement of a word within the clause Hudson wrongly (and inadvertently) predicts that the subject should extract. For example, *she* in *He knows that she will* is both subject and object of *will* and by virtue of being an object it should be able to extract. Even if we restrict ourselves to complementation patterns countenanced by Hudson (1990) we have examples like *He knows that she was seen*, where *she* is object of *seen* and so should be extractable.

The solution to this problem follows from rules such as (25a–c). The default rule of this sort that applies to subjects is (28). It rules out, e.g., **Who do you doubt that was seen* (which Hudson (1990) overgenerates) because *who* is subject but not s-dependent of finite *was*.

(28) By default, if *x* is subject of *y* and *y* is instance of Finite then *x* is s-dependent of *y*.

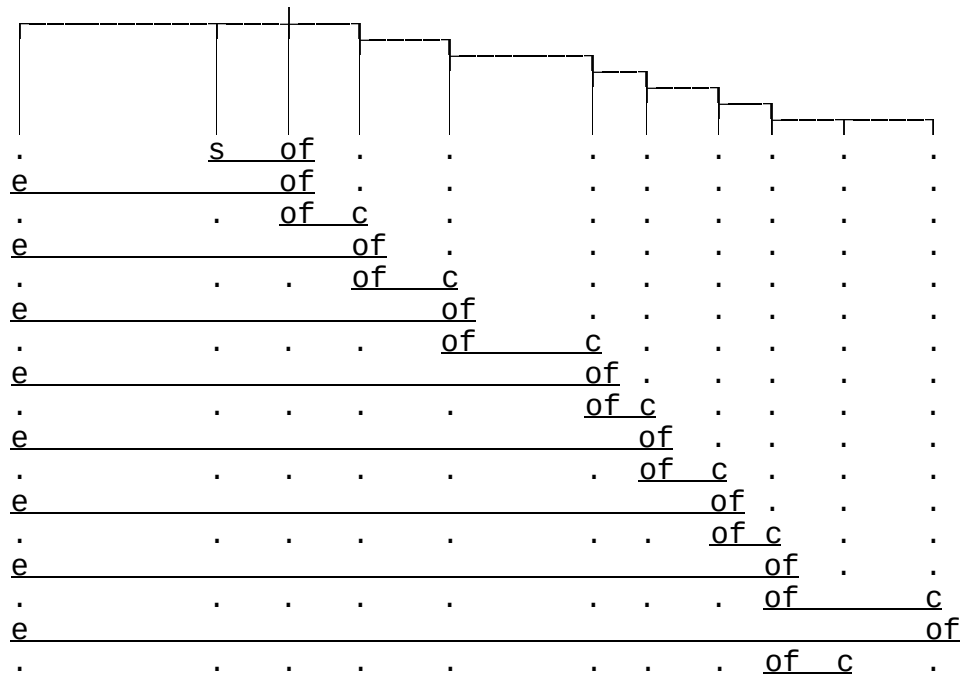
(28) is overridden by the more specific rule (29a) (which is more specific because finites that are complements of verbs are a subset of finites in general). (28) and (29a) combined give the exceptionless rule (29b).

(29) a. If *x* is subject of *y* and *y* is instance of Finite and *y* is complement of *z* and *z* is instance of Verb then *x* is s-dependent of *y* or *x* is extractee of *y*.
b. Without exception, if *x* is subject of *y* and *y* is instance of Finite then *x* is s-dependent of *y* or ∃ *z* [*y* is complement of *z* and *z* is instance of Verb then *x* is extractee of *y*]

(29a) allows *Who do you doubt went*, because *went* is complement of a verb, *doubt*, so its subject can extract – that is, *who* can be extractee and not s-dependent of *went*. But (29a) won't let *Who do you doubt that went* through, because if we assume the dependency structure in (27) then here *went* is not complement of a verb so there is no exemption from (28) and who must be s-dependent of *went*.

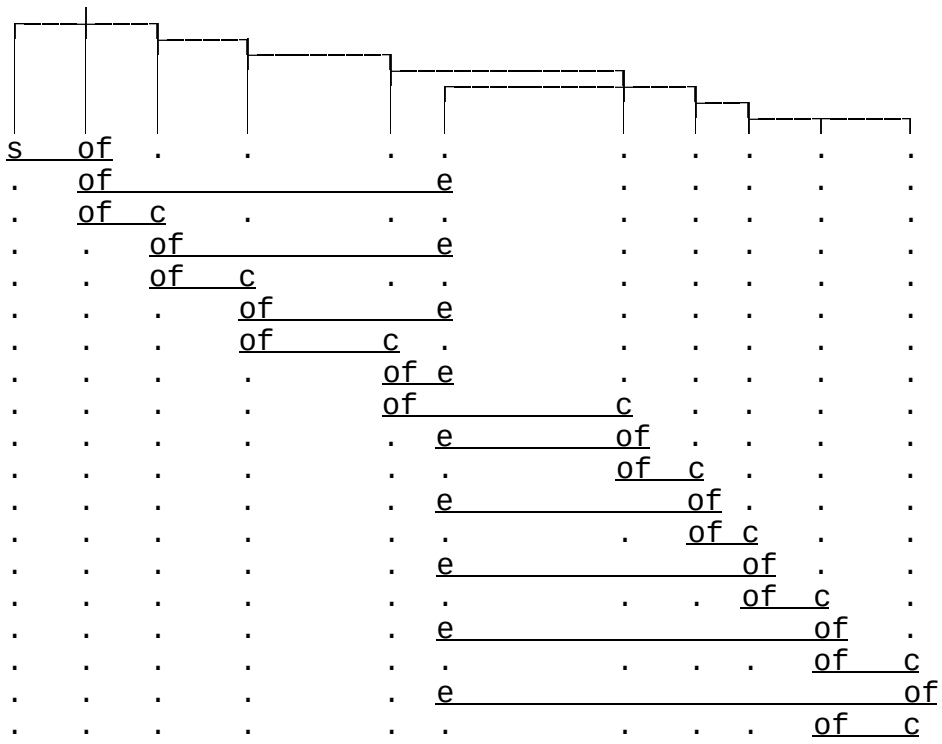
3.4 Landing sites for extraction

Word Grammar has a stepping-stone analysis of extraction. In (30), for example, *chocolate* is extractee not only of *on* and of *has* but also of all the words in the s-dependency chain between them, as shown in the diagram. (The decision to treat *on* as dependent of *cut* rather than *down* is gratuitous.)

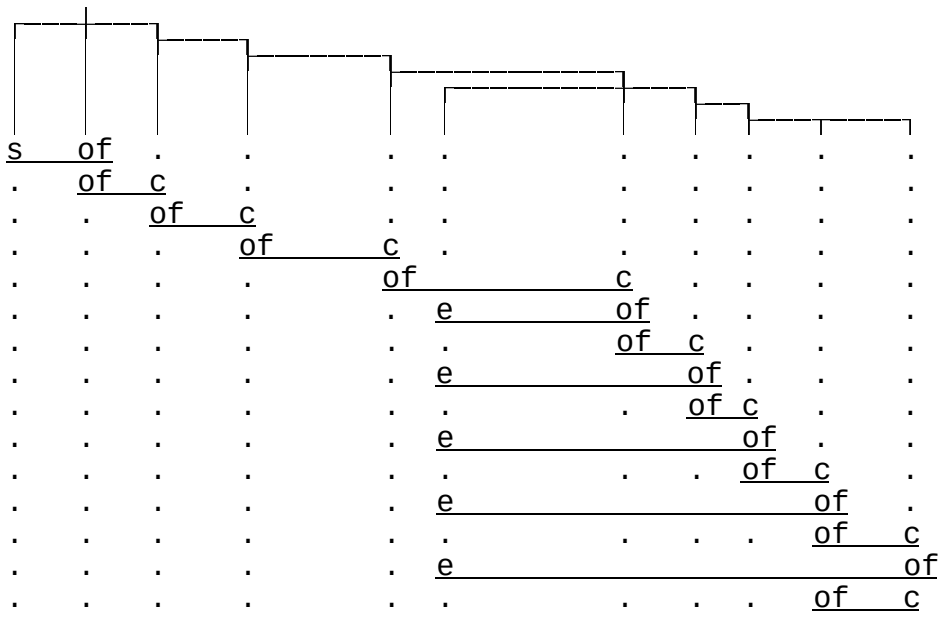


(30) Chocolate, she has been meaning to try to cut down on.

Having *chocolate* be extractee of all these words has various benefits, notably the benefit of accounting for a number of island constraints (see Hudson (1990)), but it runs into a problem: how come *chocolate* can't come to rest at some point between *has* and *on*? What rules out either of (31a–b) as possible structures?



(31) a. * She has been meaning to chocolate try to cut down on.



b. * She has been meaning to chocolate try to cut down on.

The solution to this problem is rule (32), which says that if X is extractee of its s-head, its s-head must be finite.[20] The effect of the rule is that an extractee can come to rest at only a subset of the words it can extract across. (Note that this rule cannot be formulated without reference to s-dependency.)

(32) If *x* is extractee of *y* and *x* is s-dependent of *y* then *y* is instance of Finite.

(31a–b) are both excluded by (32) because *chocolate* is extractee and s-dependent of *try*, but *try* is not instance of Finite.

3.5 A constraint on prepositional passives

In this section we see how s-dependency allows us to capture the constraint that a word can promote (NP-move) only upwards.

(33a–b) are of course acceptable, as are, if suitably intoned, their counterparts (34a–b), with the preposition extracted. (35a–b), prepositional passive counterparts of (33a–b) are fine too. But (36a–b), counterparts of (35a–b), with the preposition extracted, are ungrammatical.

(33) a. She slept in the bed.
b. She wrote on one side of the paper.

(34) a. In the bed, she slept.
b. On one side of the paper she wrote.

(35) a. The bed was slept in.
b. The paper was written on one side of.

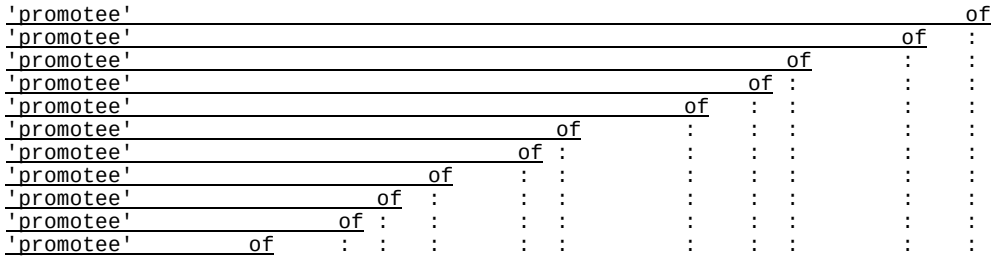
(36) a. *In, the bed was slept.
b. *On which side of was the paper written?

How come the complement of the prepositions in (35a–b) can promote, but the complement of the prepositions in (36a–b) can't?

[20]There are exceptions to the rule, e.g. to accommodate *Where to?, Who with?*, etc.

Prepositional passives seem to work by means of a severely constrained recursive mechanism the details and full nature of which remain unclear. But I will sketch an analysis that is probably broadly along the right lines. There seems to be no limit in principle to the length of the promotion path, as (37) suggests, though each step in the promotion path is tightly restricted – it's hard to come up with examples that differ very much from (37). I therefore assume that (37) involves something akin to the partial associacy structure shown.

(37)

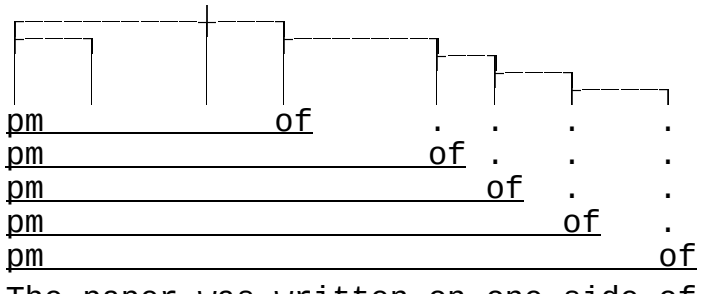


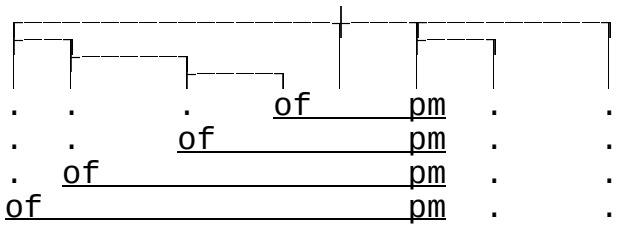
One of the constraints on 'promotees' is, I suggest, something like (38), the effect of which is to allow promoted complements to proceed only upwards.

(38) If *x* is promotee of *y* then ∃*z* *y* is unsuperordinate of *z* and *z* is instance of Passive. (An unsuperordinate of W is W or a subordinate of W.)

This has the effect of allowing (39a)/(35b), because *the* can get to be promotee of *written* and once it is promotee of *written* it can be subject of *written*. But (39b)/ (36b) is excluded, because although *the* could get to be promotee of *on*, it cannot go the further step of becoming promotee of, and hence subject of, *written*, because *on* is not unsuperordinate of *written*; (38) is not satisfied, because there is a word (*on, which, side*, of) that *the* is promotee of but that is not unsuperordinate of a passive. ('Pm' = 'promotee'.)

(39) a.





b. * On which side of was the paper written?

4 S-dependency loops

4.0 Outline

This section explains how some s-dependencies do not form branches of the skeletal tree. In such cases, a word has more than one s-head, but only one of its s-heads is its aunt. This happens in coordination, §4.1, and pied piping, §4.2.

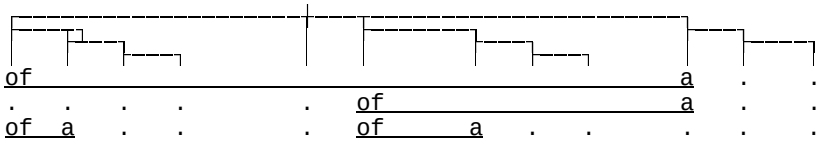
4.1 Coordination

Hudson's (1984, 1988, 1989, 1990) WG analyses of coordination have always taken the view that coordinate structures, and no others, are built by constituency rather than dependency. It would be better if coordination were by dependency, firstly because this would reduce the range of structural devices employed in syntax (from dependency plus constituency, to just dependency), and secondly because, as any moderately comprehensive survey of coordination data will reveal, there is no sharp divide between coordinate and non-coordinate structures – there is a gradient of constructions with incrementally fewer and fewer properties of canonical coordination. Surprisingly, Hudson has never really justified his rejection of coordination by dependency. In Hudson (1990: 405) the objection to coordination by dependency is fleshed out only to 'no one word stands out as the root of the whole coordination. Any one of the conjuncts could in general occur in place of the coordination [...] so each of their heads would qualify equally well as the head of the whole coordination. Nor could we take the conjunction AND as the head of the coordination because its distributional properties have nothing at all to do with those of the whole coordination (which follows the ordinary rules for nouns, verbs and adwords rather than those for conjunctions).' It's curious that he takes this as necessitating a rejection of dependency, since it is an argument not so much

against dependency as in favour of dependency loops, which WG, exceptionally among theories of Dependency Grammar, allows. In Hudson (1984: 212) all that is said is 'for simplicity I shall simply assume that they are not dependency structures, and leave it to others to prove the contrary'. Since dependency is in principle preferable to constituency, the job for these others is merely to show that dependency does the job. Here there is not enough space to provide a full-blown analysis of coordination by dependency, but since there is not even any prior reason to believe dependency can't do the job I shall (offering a promissory note to whoever wants one) assume it can.

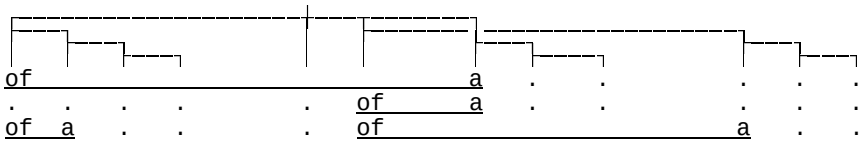
Considerations of continuity lead us to conclude that the conjunction is the root of the coordination. Conjuncts are niece of the conjunction. So are words that are dependents of each of the conjuncts: this can be seen from contrasting (40a–b). *Who (was French)* is adjunct only of *another* and is niece of *another*, while *who (were embracing)* is adjunct of both *one* and *another* and is niece of *and*. If the order is as in (40b), branches cross, and the phrase is ungrammatical.

(40) a.



one who was Japanese and another who was French who were embracing

b. *



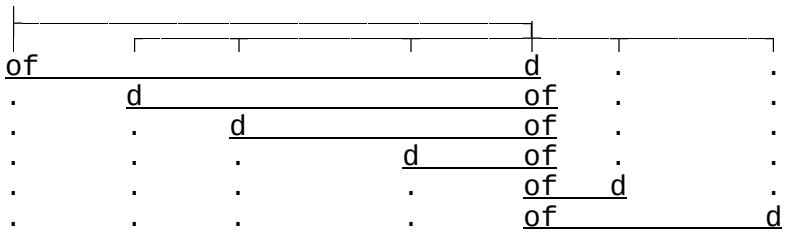
one who was Japanese and another who were embracing who was French

(41) therefore must, for the reasons just given, have the constituency shown in the diagram by stemma and by bracketing.

(41)

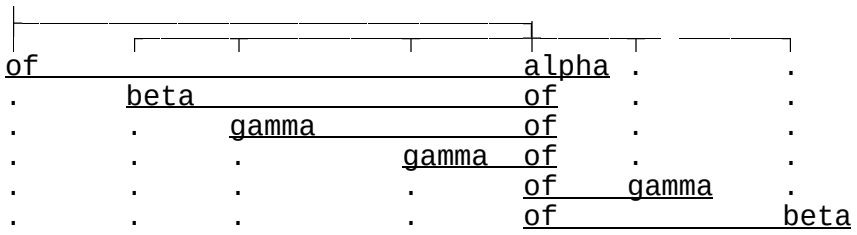
[that [[she] [washed], [cored] and [peeled] [apples]]].

Since every branch must ultimately correspond to a dependency, this means that we need the dependencies shown in (42).



(42) [that [[she] [washed], [cored] and [peeled] [apples]]].

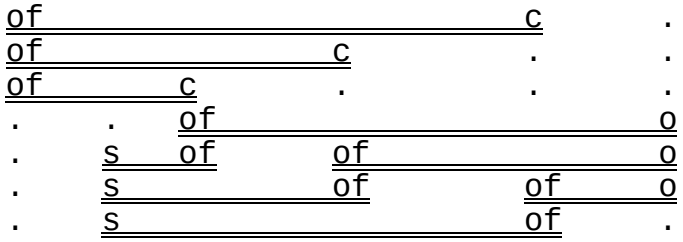
These dependencies are of three types, none of which are previously known to us. I'll arbitrarily label the three types 'alpha', 'beta' and 'gamma':



(43) [that [[she] [washed], [cored] and [peeled] [apples]]].

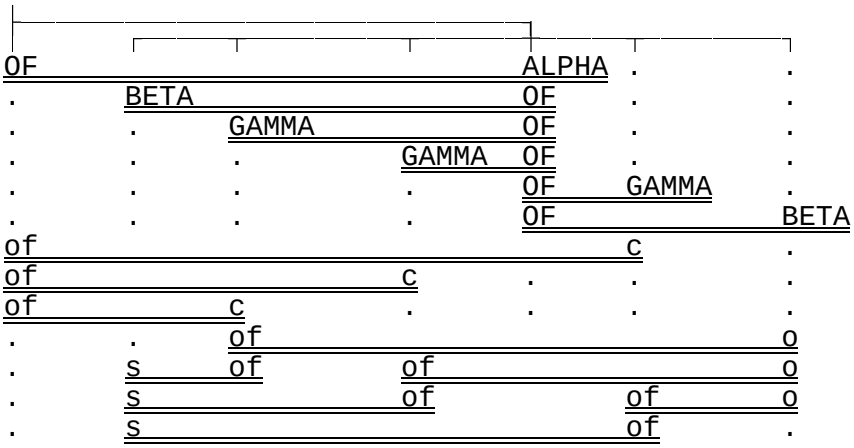
'Gamma' dependencies hold between conjuncts and conjunction. An 'alpha' dependency holds between *and* and *that*. A 'beta' dependency holds between *she* and *and* and between *apples* and *and*. A beta dependent of a conjunction is a dependent of each of the conjunction's gamma dependents; for example, *apples*, a beta dependent of *and*, is object of each of the gamma dependents of *and – washed, cored* and *peeled*.

But in addition to these s-dependencies shown in (41–43), we also require the additional s-dependencies shown in (44) by double underlining. *She* and *apples* must be s-dependent of *washed, cored* and *peeled* because that is what by default they must be, unless they are extracted (or otherwise moved), which they are not, and because *she* must precede — i.e. be precessor of — *washed, cored* and *peeled* and *apples* must follow them – it must be their successor.



(44) that she washed, cored and peeled apples.

So, to get the full structure, we must combine (43) and (44), which gives us the result shown in (45). Constituency is shown by stemma and bracketing. S-dependencies are shown by double underlining. S-dependencies corresponding to skeletal branches are capitalized.



(45) [that [[she] [washed], [cored] and [peeled] [apples]]]

As we can see, there are s-dependency loops here: *she, washed, cored, peeled* and *apples* are all s-dependent of more than one word. They're all s-dependent and niece of *and*; *she* and *apples* are also s-dependent (but not niece) of *washed, cored* and *peeled*, which in turn are each also s-dependent (but not niece) of *that*. The conjunction's beta dependent is s-dependent of each of the conjunction's gamma dependents, and each of the gamma dependents is s-dependent of the word the conjunction is alpha dependent of; see rule (49) below.

As a final step prior to formalizing this phenomenon of s-dependency loops, I'll define an associacy type, 'Legate', that has the special property of licensing s-dependency loops. Only legates can be alpha dependents and have beta dependents. Rule (46) tells us where to find legates, and it gives us the legate associacies shown in (47), where italics indicates associacies that aren't dependencies.

(46) If *x* is gamma dependent of *y* then *y* is legate of *x*.

```
OF                                  ALPHA    .        .
.     BETA                          OF       .        .
.     .     GAMMA                   OF       .        .
.     .     of                      legate   .        .
.     .     .          GAMMA        OF       .        .
.     .     .          of           legate   .        .
.     .     .          .            OF       GAMMA    .
.     .     .          .            legate   of       .
.     .     .          .            OF                BETA
```

(47) `[that [[she] [washed], [cored] and [peeled] [apples]]]`

We can now replace the default rule (7b), repeated here as (48), by the exceptionless (49), which also includes in it the earlier exceptionless (6a).

(48) By default, if *x* is s-dependent of *y* then *x* is niece of *y* and *x* is instance of Word and *y* is instance of Word.

(49) Without exception, *x* is s-dependent of *y* iff
[*x* is niece of *y* and *x* is instance of Word and *y* is instance of Word,
or ∃*z* *z* is legate of *x* and *z* is alpha dependent of *y*,
or ∃*z* *z* is legate of *y* and *x* is beta dependent of *z*]

4.2 Pied piping

For pied piping, Rosta (1994) motivated the partial associacy structure shown in (50a–b) ('a' = adjunct, 'k' = clausal, 'o' = object, 'p' = proxy). We will take this to be correct.

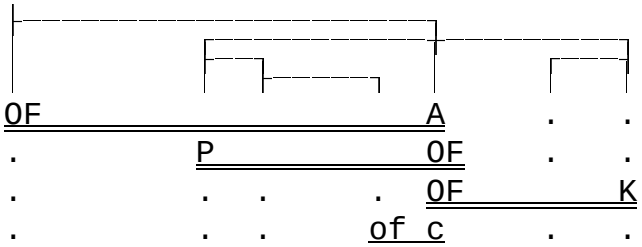
```
of                    a
.         p           of
.         .  .     .  of       k
```

(50) a. `libraries in nooks of which she dozed`

```
of        o
.      p  of
.         of              k
```

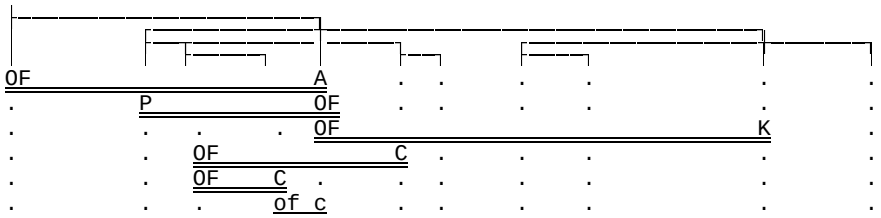
b. `wonder in which nooks she dozed`

Rosta (1994) took the additional and incorrect step of treating proxy associacies as (in terms of the present analysis) dependents and potential s-dependents, giving the the analysis shown in (51).



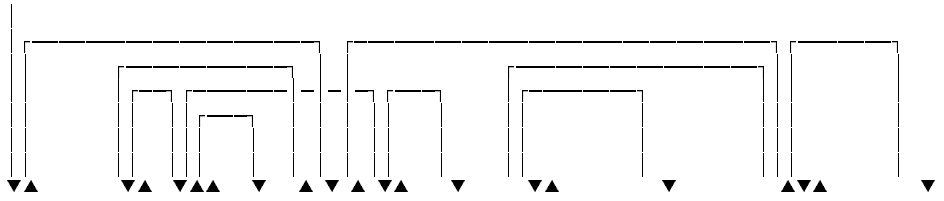
(51) libraries in nooks of which she dozed

There are a number of objections to this analysis, the most crushing of which is that applied to (52), it leads to tangling branches.



(52) flowers, her gift of which to Edgar many commentators thought improper

In (52) the branch from *gift* to *to* crosses the branch from *flowers* to *which*. If we use Hudson's (1994b) notation for the same structure, as in (53), there is apparently even more tangling. (Due to software limitations his arcs are here rendered rectilinearly.)

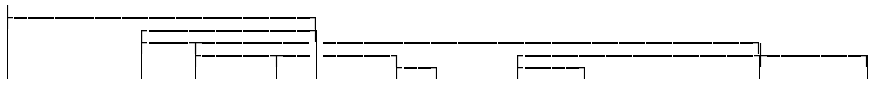


(53) flowers, her gift of which to Edgar many commentators thought improper

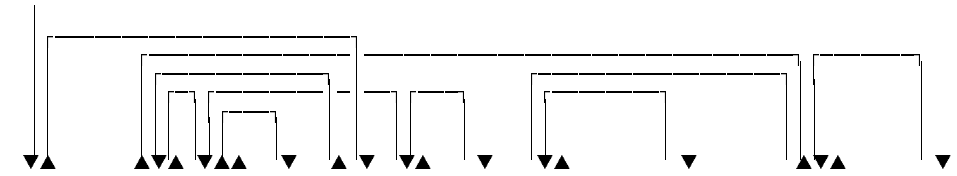
Here we see the *gift–to* s-dependency tangling not only with the *which–thought* but also with the *flowers–which* and *which–her*.

The skeletal tree assumed by Rosta (1994) therefore cannot be correct. What should the tree be, then? Suppose *thought* were niece of *her* rather than of *which*, as in (54a–b) (the same trees shown with the two notations). On its own, this change doesn't help: there is no longer a tangle of *which–thought* with *gift–to*, but *flowers–which* still tangles with *gift–to* and now tangles with *her–thought*.

(54) a.

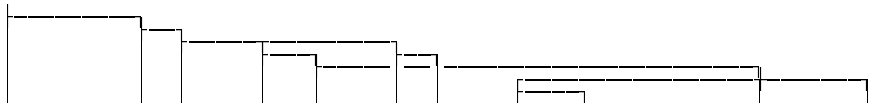


b.

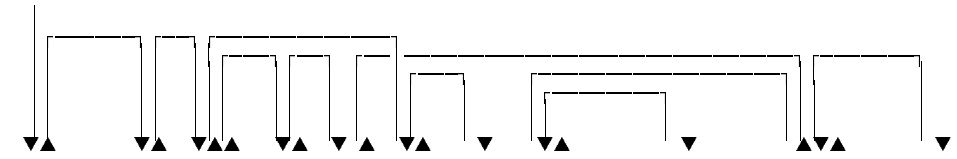


Similarly, if we adapt (52)/(53) by making *her* rather than *which* s-dependent of *flowers*, tangling remains, as (55a–b) show. We lose the tangling of *gift–to* with *flowers–which* and *which–her*, but *gift–to* continues to tangle with *which–thought*.

(55) a.

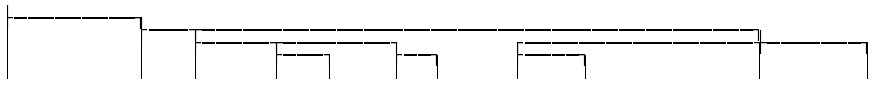


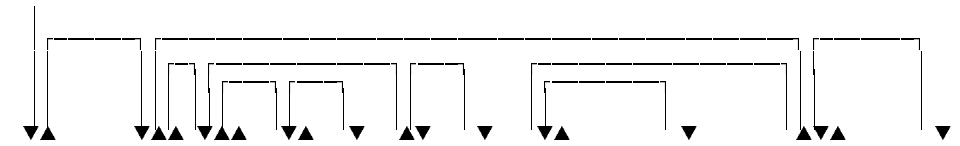
b.



However, if we combine the revisions leading to (54) and (55) this will give us the tangle-free structure in (56a–b).

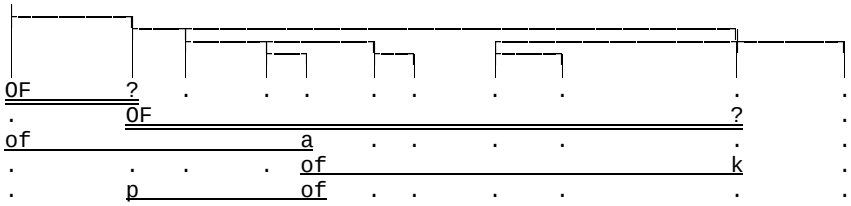
(56) a.





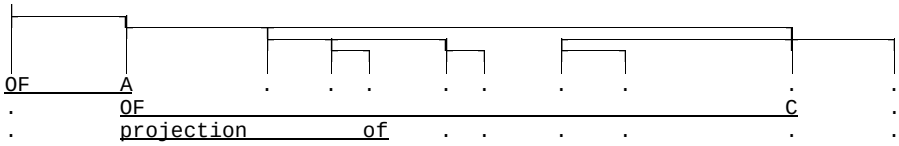
b. flowers, her gift of which to Edgar many commentators thought improper

The skeletal tree of (56a–b) requires the presence of additional dependencies that have not yet been independently motivated. These are shown in (57) labelled '?'.



(57) flowers, her gift of which to Edgar many commentators thought improper

What are these mystery dependencies? I will address this question shortly, though at this point we can note that salvation won't come from Hudson's (1990) analysis of pied piping, even though his analysis would give the dependency/associacy structure shown in (58), with *her* as adjunct of *flowers*, and *thought* as complement of *her*. That analysis is to be rejected, for the reasons given in Rosta (1994)'s criticisms of it (which I won't repeat here).



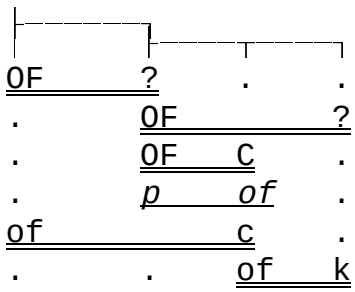
(58) flowers, her gift of which to Edgar many commentators thought improper

A further objection to (51) that is relevant here is that it is an exception to the otherwise exceptionless rule that complements of a preposition P are either s-dependent of P or are promoted or extracted. In (51) *which* is complement of *of* but is neither s-dependent of *of* nor extracted or promoted. In accordance with (7a), the rule that by default dependents of W are s-dependents of W, we expect at least the partial s-dependency structure shown in (59a–b).

```
        of                      sd
        .         of sd         .
        .         .  of    sd   .
        .         .  .     of sd
        .         .  .        of        sd
(59) a. libraries in nooks of which she dozed

        of        sd
        .      of sd
        .      .  of    sd
        .      .  of              sd
     b. wonder in which nooks she dozed
```

We have established that pied piping involves the associacy structure shown in (60a), and the constituency structure shown by stemma in (60b). We also require the s-dependencies shown in (60c): they will preserve the generalizations that dependents of W are s-dependents of W unless they are 'moved' (extraposed, extracted, promoted, etc.). Adding together all of these gives us the overall pattern in (60d).

```
        of     ?
        .      of        ?
        .      of   c    .
        .      p    of   .
        of          c    .
        .      .    of   k
(60) a. wonder with whom to dine

        ┌──────┐┌─────┬──┐
     b. wonder with whom to dine

        of          sd
        .      .    of   sd
     c. wonder with whom to dine
```

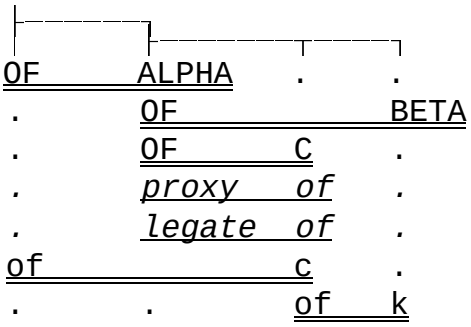


d. wonder with whom to dine

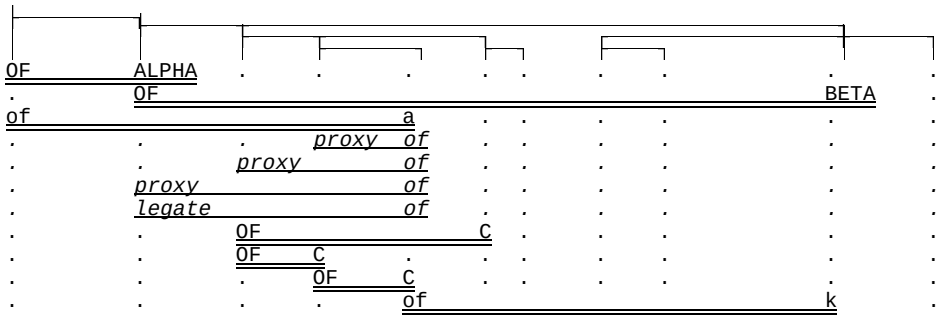
The only remaining step is to complete the associacy structure (60d) in such a way as to yield the requisite s-dependency structure. The single addition we need is provided by rule (61).

(61) If *x* is a wh-pronoun, and *y* is clausal of *x*, and *z* is proxy of *x*, and [*z* is subject of *y* or *z* is extractee of *y*] then *z* is legate of *x*.

We end up with the associacy and s-dependency structure illustrated in (62a–b). In the diagram labels of dependents of W that are also s-dependents of W are italicized and labels of (s-)dependents of W that are also nieces of W are capitalized. The single lines in the stemma represent the part of s-dependency structure that corresponds to constituent structure, and the double lines represent s-dependencies that do not correspond to constituent structure.



(62) a. wonder with whom to dine



b. flowers, her gift of which to Edgar many commentators thought improper

5 Conclusion

This paper has introduced and motivated s-dependency, which is essentially a relation of linear precedence, being equivalent to 'precessor or successor' (§3.1), but is considerably more than a means of stipulating word order (§§3.2–5).

Every s-dependent of W is dependent of W. Usually, dependents of W are s-dependents of W, but in some constructions, typically those requiring 'movement', a word can have dependents that aren't its s-dependent (§2.2).

Sentences have a skeletal constituency structure. Every word is daughter of a phrase. Every phrase is mother of one word and zero or more phrases (§2.1). Every word is s-dependent of its aunt (§2.2). In certain constructions, including coordination and pied piping, a word can have more than one s-head, in which case it is niece of one of its s-heads but not of the others (§4).

References

Baltin, M.R. & Kroch, A.S. (1989). *Alternative conceptions of phrase structure.* Chicago: University of Chicago Press.

Bresnan, J. (ed.) (1982). *The mental representation of grammatical relations.* Cambridge, Massachusetts: MIT Press.

Crystal, D. (1971). *Linguistics.* Harmondsworth: Penguin.

Fraser, N.M. (1989). 'Parsing and dependency grammar.' in R. Carston (ed.) *UCL Working Papers in Linguistics* 1. 296–320.

Fraser, N.M. (1990). 'Prolegomena to a formal theory of dependency grammar.' in J. Harris (ed.) *UCL Working Papers in Linguistics* 2. 298–319.

Gaifman, H. (1965). 'Dependency systems and phrase-structure systems.' *Information and Control* 2. 304–337.
Hudson, R.A. (1984). *Word Grammar*. Oxford: Blackwell.
Hudson, R.A. (1988). 'Coordination and grammatical relations.' *Journal of Linguistics* 24. 303–342.
Hudson, R.A. (1989). 'Gapping and grammatical relations.' *Journal of Linguistics* 25. 57–94.
Hudson, R.A. (1990). *English Word Grammar*. Oxford: Blackwell.
Hudson, R.A. (1992). 'The case against Case'. in H. van de Koot (ed.) *UCL Working Papers in Linguistics* 4. 301–326.
Hudson, R.A. (1994a). 'Word Grammar.' In Asher, R. (ed.) *Encyclopedia of Language and Linguistics*. Oxford: Pergamon Press. 4990–4993.
Hudson, R.A. (1994b). 'Discontinuous phrases in dependency grammar.' In J. Harris (ed.) *UCL Working Papers in Linguistics* 6. 89–120.
Hudson, R.A. (1995a). 'Does English really have case?' *Journal of Linguistics* 31. 375–392.
Hudson, R.A. (1995b). 'HPSG without PS.' Ms.
Hudson, R. (1995c). 'Measuring syntactic complexity.' Paper given at the September Linguistics Association of Great Britain conference at the University of Essex.
Kaplan, R.M. & Bresnan, J. (1982). 'Lexical-Functional Grammar: a formal system for grammatical representation.' in Bresnan (ed.) (1982). 173–281.
McCawley, J.D. (1982). 'Parentheticals and discontinuous constituent structure.' *Linguistic Inquiry* 13. 91–106.
McCawley, J.D. (1989). 'Individuation in and of syntactic structures.' In Baltin & Kroch (eds.) (1989). 117–138.
Postal, P.M. (1994). 'Parasitic and pseudoparasitic gaps.' *Linguistic Inquiry* 25. 63–117.
Robinson, J.J. (1970). 'Dependency structures and transformational rules.' *Language* 46. 259–285.
Rosta, A. (1994). 'Dependency and grammatical relations.' In J. Harris (ed.) *UCL Working Papers in Linguistics* 6. 219–258.
Taglicht, J. (1994). 'Syntactic structure and intonational phrasing.' Ms.
Taglicht, J. (1995). 'Syntactic constraints on intonational phrasing in English.' Paper given at the September Linguistics Association of Great Britain conference at the University of Essex.