Gap Licensing in *tough*- and Similar Constructions

Hee-Rahk Chae

In this paper[1], we will examine several different types of *tough*- and similar constructions. They have A-positon fillers and show unbounded dependency between filler and gap. Problems with the rule-based approach of GKPS arise when the triggering element of the construction concerned is not the head of the local structure and when the same lexical items in different constructions carry their gap-licensing properties with them. I propose a new gap licensing mechanism, which introduces a FOOT feature [⟨GAP α⟩ LICENSOR], which is specified in the lexicon as a part of the syntactic information of those lexical items which trigger the construction. We will see that the present analysis can handle all the problems posed by the GKPS approach and can account for some recalcitrant data, which are too tough to handle in other approaches.

0. Introduction

In this paper, I propose that a set of (unbounded) A-position filler-gap constructions can be dealt with by a single mechanism based on a new feature

[1] An abridged version of this paper was presented at the 9th West Coast Conference on Formal Linguistics held at Stanford University on March 3, 1990 and it appears in the proceedings thereof. My deepest thanks go to Arnold Zwicky and Robert Levine, whose comments and suggestions on the earlier versions of this paper were indispensable for the development of my original rough ideas. Comments from David Dowty, Pauline Jacobson, Carl Pollard and John Nerbonne were also very helpful. In addition, I am grateful to my colleague Mark Libucha for helping me out with the grammaticality judgments and with other suggestions.

which I refer to as a GAP LICENSOR[2]. The basic idea behind this analysis is that each lexical item which triggers the construction has this feature as a part of its syntactic information. We will see that this lexically oriented approach is superior to the rule-based approach of the GKPS framework (Gazdar, Klein, Pullum and Sag 1985).

Tough- and similar constructions are examined in section 1. Even though not all of these constructions have exactly the same characteristics, they are similar in the sense that they show (unbounded) dependency between a filler NP and an accusative NP gap. In section 2, I provide an analysis of these constructions within the GKPS framework, pointing out certain problems that arise in this analysis. These problems and some other motivations lead us to a new analysis, which employs the GAP LICENSOR feature and a principle which defines its role, in section 3. In this new approach, no separate rules are necessary to account for the constructions which are discussed in section 1.

1. *tough-* and Similar Constructions

1.1. *easy*-type Constructions

The following sentences represent typical examples of the *tough-* construction:

(1) a. Bob is hard to convince.
    b. Jack is easy to fool.
    c. This paper was tough for me to finish in a week.

We can see the characteristics of unbounded dependency between filler and gap:

[2] My original term for this feature was SLASH LICENSOR. But I adopted Hukari and Livine's (1987) proposal that SLASH (/) and GAP (//) have different characteristics and hence should be distinguished. From a technical point of view, there is also a problem of the SLASH LICENSOR approach in the following case:

(i) Which violins are those sonatas too complex to play on?

Here the VP node which dominates *to play on* has two /NP's. Then, it would be very difficult to make explicit the fact that the LICENSOR feature licenses the /NP which is related to the gap after *play* rather than that after *on*. However, under the GAP LICENSOR approach, the VP has two different kinds of features: GAP(//) NP for the gap after *play* and SLASH(/) NP for the gap after *on*.

(2) a. This new kind of box was hard for me to persuade Harry to retool his factory to produce.
b. Robin isn't easy for me to keep Dana from criticizing.
c. This paper was tough for me to try to finish in a week.

The *easy*-type construction (3a) has a parallel construction with the expletive *it* (3b) or the infinitive clause (3c) as its subject:

(3) a. Bob is hard to convince.
b. It is hard to convince Bob.
c. To convince Bob is hard.

But we do not have a parallel left dislocation construction, which has a resumptive pronoun in the place of the gap:

(4) *Bob is hard to convince him.

As the sentences in (3b-c) indicate, there seems to be no thematic relationship between the filler and the matrix predicate in this construction[3].

The following list presents some other adjectives which trigger the *easy*-type construction:

(5) awkward, convenient, impossible, tricky, difficult, nice, unpleasant, pleasant (Quirk, Greenbaum, Leech and Svartvik 1985 (henceforth QGLS): 1229)[4], delightful, exciting, interesting, boring, awful.

(6) a. The book is impossible for Tom to read.
b. Those darts are tricky to use.
c. Mozart is delightful to play on this violin.
d. The backyard would be exciting for Koala bears to be in.
e. Vienna is nice for us to visit.

[3] However, the existence of the corresponding expletive *it* construction does not guarantee that there is no thematic relation between the filler and the matrix predicate:

(i) (?)Robin is obnoxious to try to get to talk to.
(ii) It is obnoxious to try to get to talk to Robin.

The truth conditions for these two sentences are not the same (Robert Levine, personal communication).

[4] QGLS say that adjectives in this construction refer to "degrees of ease or comfort." However, this phrase does not seem to characterize all the adjectives concerned.

1.2. *breeze*-type Construction

There is a group of nouns and noun phrases which triggers largely the same type of construction as the *easy*-type construction:

(7) a. The game was a breeze for her to win.
b. John is a pain to deal with.
(8) a. The game was a breeze for Tom to convince her to win.
b. John is a pain to try to deal with.

This construction shows the characteristics of unbounded dependency and other characteristics of the *easy*-type construction:

(9) a. The game was a cinch for her to win.
b. It was a cinch for her to win the game.
c. For her to win the game was a cinch.
(10) *The game was a cinch for her to win it.

There is no thematic relationship between the filler and the matrix predicate.

The following nouns and noun phrases also occur in the *breeze*-type construction:

(11) snap, bear, bitch, nuisance, blast, joy, pain (in the neck), a waste of time, a piece of cake.
(12) a. John is a bear to get along with.
b. Lee is a bitch to make friends with.
c. Tom is fun for us to give presents to.
d. Mary is a pleasure to teach.
e. Hebrew is a waste of time for us to learn.
f. Mark is a piece of cake for me to beat.
g. John is a pain in the neck to have to deal with.

1.3. *ready*-type Construction

Let us observe the following sentences:

(13) a. The house is ready for Jim to buy.
b. These books are available for visitors to borrow.
(14) a. The house is ready for Tom to force Jim to buy.
b. These books are available for the principal to allow visitors to borrow.

Sentences of this type also show the filler-gap relationship and this relationship can be unbounded (14).

But this construction has different characteristics from those which we have observed thus far:

(15) a. The house is ready for Jim to buy.
b. *It is ready for Jim to buy the house.
c. *For Jim to buy the house is ready.
(16) The house is ready for Jim to buy it.

The matrix predicate does not allow *it* or the infinitive clause as its subject as we can see in (15b-c). This fact implies that there is a thematic relation between the filler subject and the matrix predicate. And this type of construction has a parallel left dislocation construction.

Other adjectives inducing this type of construction are as follows:

(17) fit, free, sufficient (QGLS, p. 1229).
(18) a. These books are free for visitors to borrow.
b. The cloth is sufficient to make a dress out of.

1.4. *too* Construction

The phenomenon of unbounded filler-gap relationship can also be found in some of the *too* ··· *to* sentences:

(19) a. John is too nasty to make friends with.
b. John is too nasty to ask Mary to make friends with.

From the following pair of ungrammatical sentences, we can see that the element which makes the above sentences have the characteristics of unbounded dependency is not the matrix adjective but *too*:

(20) a. *John is famous for you to meet.
b. *John is very famous for you to meet.
c. John is too famous for you to meet.

Sentences of this *too* construction have similar characteristics to those of the *ready*-type construction:

(21) a. John is too nasty for me to make friends with.
b. *It is too nasty for me to make friends with John.
c. *For me to make friends with John is too nasty.
(22) John is too nasty for me to make friends with him.

There is a thematic relationship between the filler subject and the matrix predicate. And the resumptive pronoun is allowed in this construction.

1.5. *enough* Construction

There seem to be no major differences between this construction and the *too* construction except the position of *too* and *enough* with reference to the adjective in the matrix predicate:

(23) a. John is tall enough for us to see.
b. John is tall enough to imagine my little son could have seen.
(24) a. *John is tall for us to see.
b. *John is very tall for us to see.
(25) a. *It is tall enough for us to see John.
b. *For us to see John is tall enough.
(26) John is tall enough for us to see him.

1.6. [*easy person*]-type Construction

There is another (unbounded) filler-gap construction which is induced by the same set of adjectives as that for the *easy*-type construction:

(27) a. John is a tough man to argue with.
b. (?) John is a tough man for me to stop Mary from arguing with.

The NP in the matrix predicate consists of one of the *easy*-type adjectives (cf. section 1.1) and a noun (some kind of "indefinite" noun, in the sense that its denotation is a superset of the denotation of the subject NP).

This construction is different from the following construction of the infinitival relative clause:

(28) a. John has a doll for Mary to play with.
b. (?)John has a doll to tell Mary to play with.

(29) John met a tough man to argue with.

First, in (27), the gap is more naturally interpreted with *John* as its filler, which is separated from the infinitive clause, rather than with *man*. But it is interpreted with *doll* as its filler in (28a), which comes just before the infinitival clause, as it is in (29) where the gap is interpreted with *man* as its filler. Second, the special (semantic) relation between NP *a tough man* and the infinitive clause in (27a) is retained in (27b). But the relation between NP *a doll* and the infinitive clause in (28a) is not preserved in (28b).

The [*easy person*]-type construction shows its own characteristics:

(30) a. John is an easy person to please.
b. *It is an easy person to please John.
c. *John is an easy person to please him.

Sentence (30b) shows that this construction is similar to those constructions which we have examined in (1.3-5) in the sense that there is a thematic relationship between the filler subject and the matrix predicate. But sentence (30c) indicates that it is also similar to those in (1.1-2) in that it does not allow the resumptive pronoun.

1.7. *take*-type Construction

The last construction to be considered in this paper is concerned with the following sentences:

(31) a. New York took me eight hours to drive to.
b. The cake[5] took Mary all day to bake.
(32) a. Minneapolis will cost me $300 to fly to.
b. The shirt cost John three dollars to buy.
(33) a. (?) Dana took me exactly four minutes to talk Robin out of marrying.
b. (?) Minneapolis will cost me $300 to arrange for Mary to fly to.

We can see that *take* and *cost* also trigger to some extent a (unbounded) filler-gap construction.

[5] David Dowty (personal communication) pointed out to me that the most appropriate NP which can occur in this position is one which indicates the result of the process represented by the gapped VP. That is, *cake* is the result of *baking*. However, the other sentences in (32) and (33) are also admitted by many people.

This construction presents characteristics which are similar to those of the *easy*-type construction:

(34) a. The shirt cost John three dollars to buy.
b. It cost John three dollars to buy the shirt.
c. *The shirt cost John three dollars to but it.

There is no thematic relationship between the filler subject and the matrix predicate. And the resumptive pronoun is not allowed.

2. A GKPS Analysis

We have observed seven different types of contructions which can be regarded as *tough*- or similar constructions in the sense that all show (unbounded) dependency between filler and gap. We found that the first two types (1.1-2) and the last type (1.7) share the following characteristics: i) they have parallel constructions with the expletive *it* as their subjects, ii) they do not have parallel left dislocation costructions. Three other types (1.3-5) group together in that they show the charateristics opposite to those of the first group. And [*easy person*]-type construction (1.6) forms a separate group. However, all of the seven types deserve a unified account because they share one of the most significant characteristics of syntactic phenomena, i.e. (unbounded) dependency between the filler NP, which usually appears as the matrix subject NP, and the accusative gap NP, which is usually[6] in the most deeply embedded clause.

In this section, I construct rules for the seven types of constructions under the GKPS framework. These rules will license the occurrence of the gap in them in cooperation with the FOOT Feature Principle (FFP). And the Control Agreement Principle (CAP) is responsible for the account of the filler-gap association.

GKPS (p. 150) has the following rule for the *easy*-type construction:

(35) $A^1 \rightarrow H[42], V^2[INF]/NP[-NOM]$

This is the only rule which is provided in GKPS to analyze "Missing Object Constructions." Besides the problem related to the difference between the wh-gap and the *tough*-gap (cf. footnote 2 and footnote 11), there is anoth-

[6] The following sentence contains a gap in a clause which is not the deepest:

Robin was easy to persuade to go to leave.

er problem with this rule in the part of /NP[-NOM] as was pointed out to me by Arnold Zwicky (personal Communication). [-NOM] cannot pick out the appropriate set of NP's:

(36) a. *Yesterday was easy for John to go.
cf. What day did you go?
b. *Here is hard for trains to travel.

Bare NP adverbials seem to be [-NOM]. But we cannot get these sentences. So, /NP[ACC] would be better for the rule. But this is not a serious problem with rule (35).

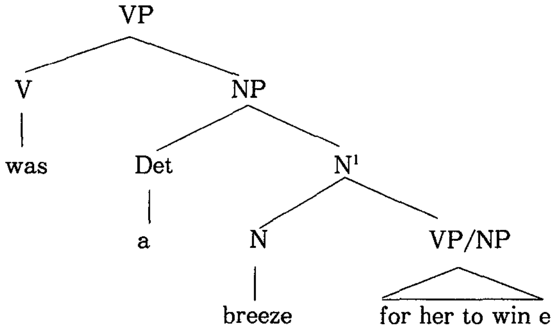
Let us consider what kind of rules would be constructed for the other types of constructions. I repeat the following sentences for convenience of reference:

(37) a. The game was a breeze for her to win. (*breeze*-type)
b. The house is ready for Jim to buy. (*ready*-type)
c. John is too short for you to see. (*too*)
d. John is tall enough for us to see. (*enough*)
e. John is a tough man to argue with. ([*easy person*]-type)
f. The cake took Mary all day to bake. (*take*-type)

Sentence (37b) can be handled with exactly the same kind of rule as (35) except that the rule would have a different SUBCAT value of the head. For the other five constructions, we need to formulate new rules.

Assuming the following tree for sentence (37a) (*breeze*-type construction),

(38)



we can formulate the rule as follows:

(39) $N^1 \rightarrow H[101], V^2[INF]/NP[-NOM]$

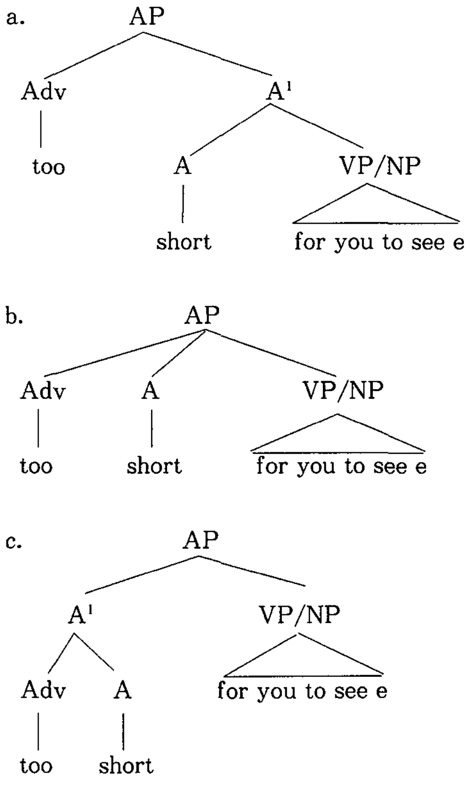
A difficulty arises when the triggering element is an N^1 rather than an N, as in the case of *waste of time* or *piece of cake* (cf. (12e-g)). As an N^1 cannot have a SUBCAT feature, it would be difficult to write a rule for these expressions. One might propose that we can reanalyze these N^1's as N's in the lexicon. However, this seems to be not a very good idea, especially for the *waste* examples, as Arnold Zwicky (personal communication) has pointed out. *Waste of time* is not a fully fixed expression:

(40) a. waste of (my/your/…) time/money/work/effort/…
b. pain in the neck/ass/butt/you know what/…

Therefore, we cannot treat these expressions as lexicalized items.

Second, let us look at sentence (37c) (*too* construction). There would be three different ways of analyzing it:

(41)



It would be difficult to write a rule for structure (c) because the trigger of the construction, i.e. *too*, and the constituent which contains the gap are not sister nodes. Such a rule would violate the locality principle of GPSG (Generalized Phrase Structure Grammar).

Let us consider structures (a) and (b). If we allow a lexical item which is not the head of a rule to have a SUBCAT feature, the rules for them would be as (42). If we do not, we need another feature of AdvFORM, one of whose value is *too*. Then, we can write the rules as (43):

(42) a. A^2→Adv[102], H^1/NP[-NOM]
b. A^2→Adv[102], H[140], V^2[INF]/NP[-NOM]

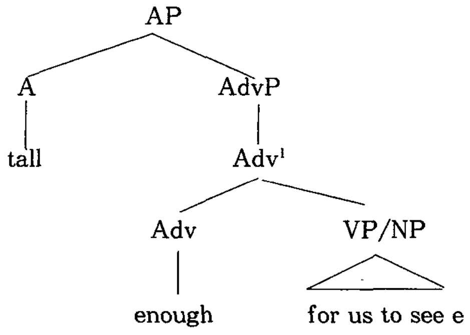
(43) a. A^2→Adv[too], H^1/NP[-NOM]
b. A^2→Adv[too], H[140], V^2[INF]/NP[-NOM]

Whichever type of rules we may choose, we need one more mechanism in the system of the whole grammar. That is, we must allow non-head lexical items to have SUBCAT features or we need another feature of AdvFORM.

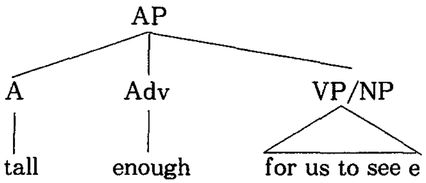
Furthermore, in the case of (42b) and (43b), the SUBCAT feature of the head has nothing to do with the characteristics of the construction. The adjectives which occur in these rules are not the elements which trigger the construction. From a different point of view, the problem is that the SUBCAT class 140 comprises all the adjectives which can occur in the predicate position in English. There is no motivation for using the SUBCAT feature. We will see the same problem in the rules of (46) and (50) below.

Third, the possible structures of sentence (37d) (*enough* construction) would be as follows:

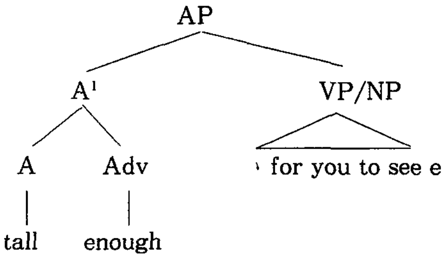
(44) a.



b.



c.



We cannot write a rule for structure (c) because the trigger of the construction and the gap are not sister nodes.

If we assume structure (a), the rule would be as follows:

(45) $Adv^1 \rightarrow H[103], V^2/NP[-NOM]$

In this case, we cannot write a rule with the new feature AdvFORM because the head of a lexical ID rule should have the SUBCAT feature. Thus, if we are going to specify the same kind of adverbs, i.e. *too* and *enough*, in the same way, we cannot use those rules in (43) for the *too* construction.

When we assume structure (b), we can write the following rules:

(46) a. $A^2 \rightarrow H[150], Adv[103], V^2[INF]/NP[-NOM]$
b. $A^2 \rightarrow H[150], Adv[enough], V^2[INF]/NP[-NOM]$

Both of these rules are not appropriate in the sense that there is no intrinsic relation between the construction concerned and the lexical head of the rule. That is, there is no connection between the gap in V^2 and the lexical item which can occur as the head of the rule. The gap is triggered by adverb *enough*, not by the head of the rule. And the SUBCAT class 150 should contain all the possible predicate position adjectives, which weakens the original motivation for the SUBCAT feature.

Hukari and Levine (1987) propose an analysis for the *too* and *enough* constructions, which avoids the above-mentioned problems and captures the fact that *too* and *enough* function in a parallel fashion. They introduce a head feature DEGFORM, which takes either [*too*] or [*enough*] as its value (p. 100). The rule for the construction is introduced by the following metarule:

(47) Degree Metarule

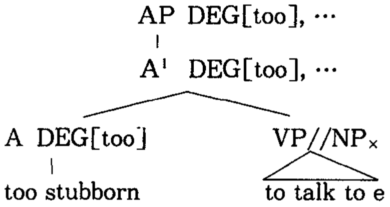
$A^1 \rightarrow W$

$\Downarrow$

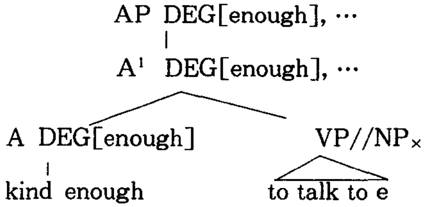
$A^1[\text{DEG}[\alpha]] \rightarrow W, V^2(//\text{NP})$, where α = *too, enough*

Based on these new mechanisms, they give the following structures for the two constructions:

(48) a.



b.

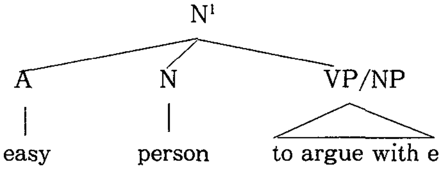


However, we have reason to disagree with their analysis. First, we need independent motivations for the DEGFORM feature and the Degree Metarule from other sources. Second, it is a little too abstract in the sense that *too* and *enough* are regarded not as words but as feature values, at least at the syntactic level. We would need to set up some independent criteria for determining which lexical items can be analyzed as features at the syntactic level. Another, related, abstractness problem is that A[DEGFORM α] is analyzed as a compound adjective in the lexicon. That is, [*too* +A] and [A+*enough*] are treated as compound adjectives, as sort of lexicalized phrases. But this does not appear to be a good idea because the [*too*+A] and [A+*enough*] expressions are not confined to some limited number of adjectives. We can reanalyze phrases as lexical items only when they are frozen expressions. However, almost all the predicate adjectives can combine with *too* or *enough* in the given constructions.

Before moving to the next filler-gap construction, let us consider which structures can be assigned for the *too* and *enough* constructions among those given in (41) and (44). We have seen that we could not write good rules for these constructions based on any of the possible structures under the GKPS framework. The issue of which structures are the right ones for these cosntructions will not affect my argumentation in the next section except when we choose structure (41a) for the *too* construction (cf. footnote 16). But it would be necessary and convenient for us to assign particular structures for them. I propose that we should use the structures in (c), i.e. [[*too* A] VP] and [[A *enough*] VP],[7] in view of such examples as (74), which have multiple modifiers of the adjective. By assuming these structures we can easily capture the fact that the modifying phrases form a constituent (cf. Gazdar 1988: 78).

Fourth, let us observe sentence (37e) ([*easy person*]-type constructions). The structure for this type of construction would be as follows[8]:

(49)



[7] From a purely semantic point of view, [*too*+VP/NP[-NOM]] and [*enough*+VP/NP[-NOM]] seem to form constituents. In that case, we might need a special operation such as Right Wrapping in Categorial Grammar for the *too* construction. However, semantic constituency does not necessarily match with syntactic constituency.

[8] There are two more possible structures for this construction: [[A N] VP/NP] and [A [N VP/NP]]. However, these structures are not likely to be appropriate ones. The first structure says that the relation between *easy* and *person* is the same as an ordinary [A N] phrase such as *easy question*, where *easy* modifies *question*. But the *easy* in sentence (37e) and that here are different lexical items. The former has the gap-licensing property and the latter does not. The second structure is the same as that for an infinitival relative. Under this structure, if (i) is ungrammatical, then (ii) should be too:

(i) *Robin is an easy man.
(ii) Robin is an easy man to argue with.

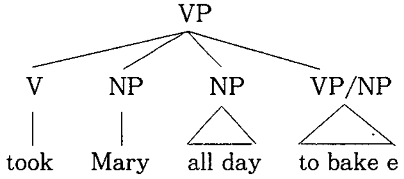
But sentence (ii) is perfect.

(50) $N^1 \rightarrow$ Adj[42][9], H[151], V^2/NP[-NOM]

Rule (50) has the same kind of problem as that for the *too* and *enough* consturction rules. The construction which it is going to describe has nothing to do with the SUBCAT value of the H. The gap here is triggered by one of the adjectives in Adj[42], not by a noun in H[151]. This is a serious problem because the fact that a particular lexical item is the triggering element of the construction is represented by the SUBCAT value of the head in the GKPS framework. Thus, the idea that some particular set of lexical items is responsible for the construction cannot be effectively implemented within the rule.

Finally, the tree structure of sentence (37f) (*take*-type construction) and the rule for it would be as follows:

(51)



(52) $V^2 \rightarrow$ H[104], NP, NP, V^2/NP[-NOM]

There seem to be no special problems for this rule.

Thus far, we have seen that there are some difficulties with the formulation of those rules for the constructions which we have observed in section 1 whithin the theoretical framework of GKPS. The following set of data seems to raise a more serious problem to the framework:

(53) a. John is too easy to make friends with.
b. It is too easy to make friends with John.
c. *John is too easy to make friends with him.

[9] The Adj here seems to have the same SUBCAT feature as the H of the *easy*-type construction rule because the same set of adjectives trigger the two constructions. The following data may be problematic for this view:

(i) The book is impossible for Tom to read. (*easy*-type)
(ii) ?* The book is an impossible thing (stuff) for Tom to read.
(iii) ?* That is an impossible book for Tom to read.

However, the ungrammaticality of (ii - iii) is due to the difficulties of the combination of *impossible* and some nouns. Therefore, the ungrammaticality of them should be explained by an independent principle.

(54) a. John is easy enough to make friends with.
b. It is easy enough to make friends with John.
c. *John is easy enough to make friends with him.

There seem to be two elements which trigger the unbounded filler-gap construction in both (53a) (*too* and *easy*) and (54a) (*easy* and *enough*).

In (53a), the *too* construction rules in (42) or (43) will apply under the GKPS analysis. That is, the sentence will be licensed by the *too* construction rule. But it shows the characteristics of the *easy*-type construction as we can see in (53b-c). Thus, I think, sentence (53a) should be licensed by the *easy*-type construction rule (35). The situation is the same with sentence (54a). It can only be licensed by the *enough*-type construction rule (45) or (46) because rule (35) cannot apply to it. But, again, it shows the characteristics of the *easy*-type construction.

Under the approach of Hukari and Levine (1987), one may say that the structures for sentences (53a) and (54a) are manifestations of the same rule as that for the structures in (48), i.e. the output of metarule (47), but the option (i.e. the parentheses in the metarule) is not taken in the former rule and is taken in the latter rule (Robert Levine, personal communication). A problem with this approach is that we cannot capture the relationship between simple sentences of the *easy*-type construction and the sentences in (53a) and (54a) because they are licensed by two different rules. We have observed that the latter sentences show the same characteristics as those for the former sentences.

We have a piece of evidence for determining the structures of the sentences in (53) and (54):

(55) How easy is John to make friends with?

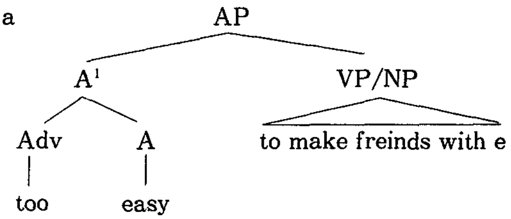
This example shows that [*too* + A] and [A + *enough*] form constituents because only constituents can be extracted out[10]:

[10] Extrapositon might be involved in sentence (55) as was suggested to me by Pauline Jacobson (personal communication):

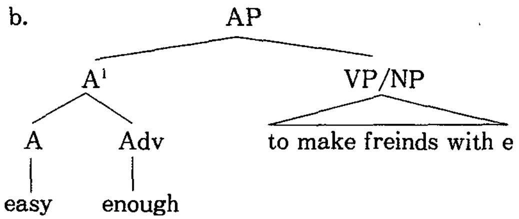
How easy to make friends with is John?

Even in that case, it would be better to assume that the major syntactic break comes in between *how easy* and *to make friends with* rather than to assume that it is in some other place.

(56) a



b.



Here we are assuming the same structures as those for the *too/enough* constructions. Then, the issue is how we can account for the fact that the sentences here show the characteristics of the *easy*-type construction rather than the *too/enough* constructions. We will see in the next section that our approach can handle this problem very naturally.

The most serious problem with the rule-based approach of GKPS framework is that it fails to express the fact that the same lexical items in different constructions carry their gap-licensing properties with them. In section 1, we have seen that the *easy*-type construction and the [*easy person*]-type construction are licensed by the same set of adjectives:

(57) a. John is tough to argue with.
b. John is a tough man to argue with.

The lexical item *tough* is associated with both of these sentences, which are manifestations of different filler-gap constructions. It seems to be true that *tough* triggers both of these constructions. Otherwise, we cannot explain why the two constructions are related with the same set of adjectives. It would be very difficult for the GKPS framework to capture the relationship between the two constructions. Each of the sentences in (57) is licensed by separate ID rules. So, we would have to relate the two rules in some way or other.

We have the following sets of sentences which show the same situation (Robert Levine, personal communication):

(58) a. Robin is too kind to take advantage of.
b. Robin is too kind a person to take advantage of.
c. Robin is too much of an unknown quantity to trust.
(59) a. Robin is kind enough to take advantage of.
b. Robin is enough of a question mark to trust.
(60) a. Robin isn't worth *(it) for you to talk to.
b. Robin isn't worth (*it) talking to.

Here we have seven distinct constructions. But the lexical item involved is the same and the gap is of the same kind in each set. Notice that all the occurrences of *too* in (58) induce exactly the same characteristics in all the three constructions: (i) a thematic relationship between the filler subject and the matrix predicate, (ii) allowance of a resumptive pronoun. The same is true with *enough* in (59). The sentences in (60) are manifestations of two different constructions because we need the pleonastic *it* for the (a) construction but not for the (b) construction. Yet both have *worth* and both contain the same kind of gap, whose filler is the subject of *worth*.

3. An Alternative Approach

Besides the task of overcoming those problems which we have observed in section 2, there seem to be other motivations for a new analysis. One of the most important characteristics of the seven constructions is that they are lexically governed. The number of the lexical items which can trigger those constructions is very small. In the cases of the *too* and *enough* constructions, there is only one such lexical item in each construction. And there are only two such items in the *take*-type construction, as far as I know. Therefore, the rules in section 2 should be different from the following rules in GKPS, which are not lexically governed but structurally governed:

(61) a. S → X^2, H/X^2
b. VP[+it] → H[44], X^2, S[FIN]/X^2 (be)

These rules are different from the *tough*-construction rules in the sense that the former contain both the filler and the gap in the same rule while the latter contain only the gap in the rule (cf. linked gap vs. free gap (Jacobson 1984)).

There are a few works which argue that we need more than one SLASH-like feature in GPSG. Zwicky (1987) introduces BSLASH (\) to account for passives, and Hukari and Levine (1987) introduce GAP (//) to analyze Missing Object Constructions. The present analysis also captures the fact that the gap in *tough*-constructions shows characteristics different than that in (61), not only by positing a different SLASH-like feature (i.e., I adopt the GAP analysis[11]) but also by eliminating all the rules for *tough*-constructions, as we will see below.

Based on the above reasoning, I here introduce a new feature [GAP LICENSOR][12] (GL), which can be generalized as [α LICENSOR]. In this particular case of the *tough*- and similar constructions, it would be [⟨GAP NP [ACC]⟩ LICENSOR]. Each lexical item which triggers one of these constructions has GL in a stack as a part of its syntactic information in the lexicon.[13] GL is a kind of FOOT feature. Thus, its behavior is similar to that of other FOOT features in its spreading throughout the tree. But it is different from ordinary FOOT features in the sense that it is a stacked feature. It moves around the tree by way of a stack. Its behavior is regulated by the following principle, which can be stated as a part of the FOOT Feature Principle (FFP) or as a separate principle:

(62) [⟨GAP XP⟩ LICENSOR] in the stack of a node (pops out of the stack and) licenses the FOOT feature ⟨GAP XP⟩ on its VP daughter.[14]

[11] Robert Levine (personal communication, cf. Hukuri and Levine (1991)) pointed out to me the following differences between the wh-gaps and the *tough*-gaps: i) free linkage through finite clauses (yes for wh, no for very many speakers for *tough*), ii) case connectivity (yes for wh, no for *tough*), iii) ability to tolerate wh-gaps as sisters (no for wh, yes for *tough*), and iv) ability to sustain gaps in the filler category itself (no for wh, yes for *tough*).

[12] LICENSOR is a feature and GAP is its value. So, [LICENSOR GAP] would be a better representation under the GKPS system.

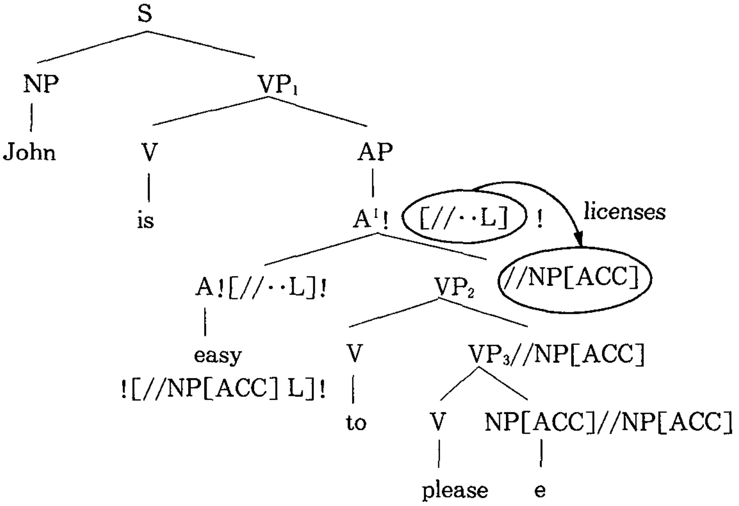
[13] An anonymous *Language Research* referee expressed a concern about the role of lexical items in the present approach. His point is that this approach is "too lexical" in accounting for the syntactic phenomenon of gaps. However, it should be noticed that lexical items provide only base information for the operation of syntactic mechanisms in our framework. Whatever framework we may adopt, we need a mechanism to implement the fact that only a limited number of lexical items induce the construction concerned. For example, *possible* does not occur in the *easy*-type construction while *impossible* does.

[14] Here I assume that there is no syntactic connectivity between filler and gap in *tough*- and similar constructions, following Jacobson (1984), Dowty and Jacobson (1989), and Carl Pollard (personal communication). If it turns out to be

The key idea here is that GL is a property of the 'top' node, which has a VP node as its daughter. However, that property is due to some particular element which is located in various different parts of different constructions. With the above principle and GL, we can connect that element and the top node very effectively. We can see a related view in Zwicky (1987). He introduces BSLAH and the construction feature PAS to describe English passives. The role of his PAS feature is similar to that of GL in the present analysis in some respect. Even though ordinary verbs do not have the feature in the syntactic part of their lexical entries, some exceptional verbs such as *resemble* and *rumor* contain [-PAS] or [+PAS] as a part of their syntactic information (p. 663).

To see how GL and principle (62) work, let us consider the following sentence of the *easy*-type construction:

(63)



[14] continued
true that there is syntactic connectivity between filler and gap in these constructions, we can implement this idea by saying that i) [⟨GAP XP⟩ LICENSOR] is an ordinary FOOT feature, and ii) it terminates when it meets XP as its sister node. This termintion mechanism will assume the role of accounting for the filler-gap association.

I use !X! to represent that X is in a stack. Principle (62) operates when !GL! is instantiated on A^1, and when //NP is instantiated on VP_2. Then, //NP[ACC] on VP_2 will be licensed by the principle. If //NP[ACC] is instantiated on A^1, it will be ruled out by a modified version of the FFP[15] because the licensed //NP [ACC] on VP_2, and the unlicensed //NP[ACC] on A^1 are different. In other words, the //NP[ACC] specification on VP_2 is required by the !GL! on A^1 (and/or by principle (62)) as opposed to free instantiation. So, //NP [ACC] instantiated above VP_2 will not terminate properly at the upper end.

Then, let us consider how we can control the propagation of the !GL! feature. According to principle (62), the GL comes out of the stack when it licenses the GAP on its VP daughter node. Its upward propagation stops at the moment when it comes out of the stack because it propagates only through a stack. Any downward propagation of !GL! would become problematic at the point of 'lexical insertion' by the fact that lexical items other than those which trigger the construction do not have that feature. In (63), if !GL! is instantiated on VP_2, the verb *to* and/or VP_3 should have that feature according to the FFP. However, the instantiation would not be allowed because none of the lexical items under the VP_2 node have the !GL! feature in the lexicon.

When we were talking about the seven different types of filler-gap constructions, we saw that some constructions do not allow resumptive pronouns while others do allow. Therefore, there seems to be more than one kind of GL involved:

(64) a. John is easy for you to please.
b. *John is easy for you to please him.
c. *John is easy for you to please his dog.

(65) a. John is too short for you to see.
b. John is too short for you to see him.
c. John is too short for you to see his eyes.

[15] The FFP as is stated in GKPS says that only inheritance from an ID rule stops the upward flow of FOOT features. But we can easily modify the FFP to the effect that not only an inherited feature but also a feature which is required by another feature has a different function from a freely instantiated feature. Under this revised version of the FFP, we can say that a feature specification which is forced to appear on a category can also stop the upward propagation of FOOT features.

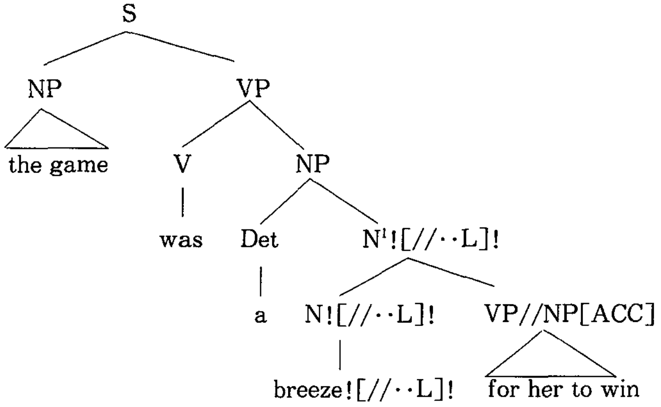
In the *easy*-type construction there must be a gap while it is not obligatory in the *too* construction.

Accordingly, we can distinguish two different types of GL: obligatory GL (we may call it GL_1) and optional GL (GL_2). The problem is how to implement the idea of obligatoriness and optionality formally. The easiest way would be using parentheses to indicate optionality. Then we can say that *too* has !(GL)! and *easy* has !GL! in the lexicon as parts of their syntactic information.

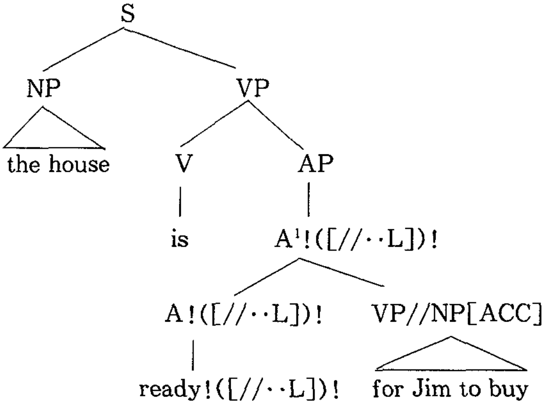
However, there is perhaps a more principled way of expressing the optionality because there seems to be a close relationship between the filler, the gap and the trigger of the çonstruction. Let us observe the data in (65). We can see that a gap exists when there is a filler which has the same information or specification as the gap, as in sentence (a). But there is no gap when there is no such filler. Based on this observation, one might propose that [⟨GAP α⟩ L_1] requires the foot feature ⟨GAP α⟩ on its VP daughter. And [⟨GAP α⟩ L_2] requires the feature only when it terminates properly. But this kind of mechanism violates the locality principle. It needs information from outside of the local tree. A better analysis may be found within a framework which directly connects the subject NP (i.e. a potential filler) and the predicate (which contains the trigger) of the construction as a kind of a compatibility rule as was suggested by Arnold Zwicky (personal communication). Here I do not want to pursue this issue further. We will simply follow the tradition of using parentheses to indicate optionality.

Under the present mechanism, we can analyze *tough*- and similar constructions without any special rules for them. No separate rules are necessary for them. Consequently, we can overcome all the difficulties which are caused by the formulation of the rules. First of all, let us see how basic sentences of each of the seven constructions can be handled. As we have already seen how the mechanism works with the *easy*-type construction, we will proceed to the other six constructions:

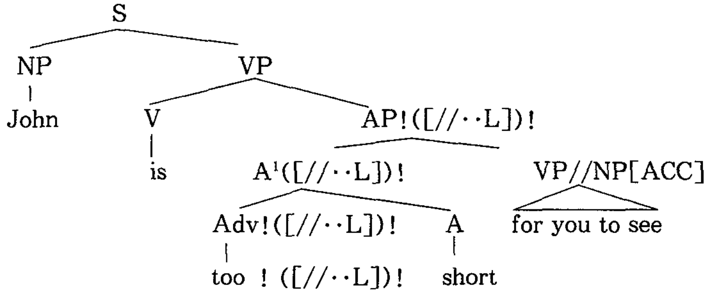
(66) a. *breeze*-type construction



b. *ready*-type construction

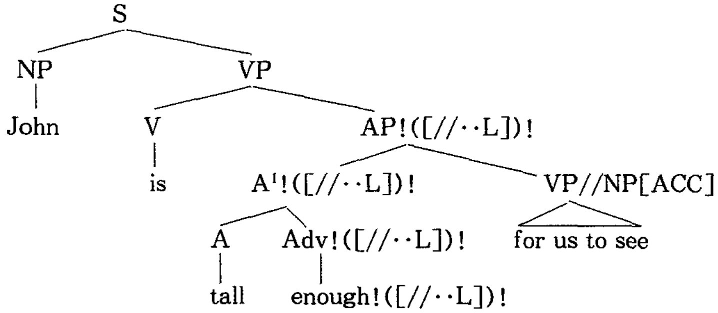


c. *too* construction[16]

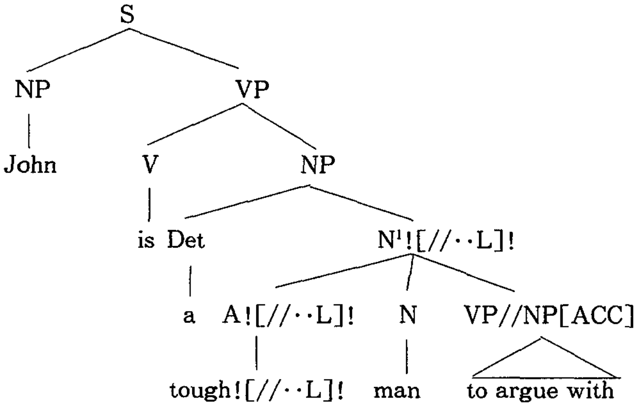


[16] In this paper, we have assumed the (c) structures in (41) and (44) for the *too* and *enough* constructions. However, this analysis is compatible with any of the structures in (41) and (44) except (41a) for the *too* construction. But (41a) is the least

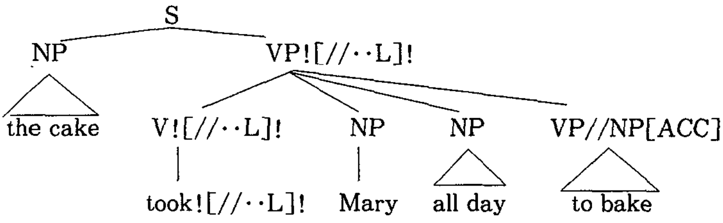
d. *enough* construction



e. [*easy person*]-type construction



f. *take*-type construction



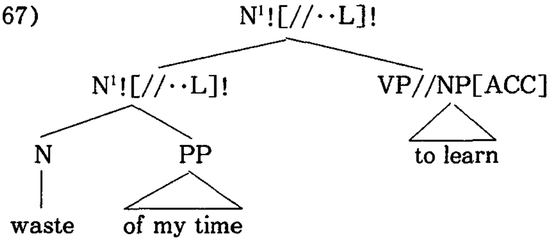
There seem to be no apparent problems in accounting for all of these sen-

[16] continued
likely structure for the construction among the three possible structures. In the worst case, we can accommodate even this structure by modifying principle (62) to the effect that the GL feature licenses the GAP feature on its VP or A¹ daughter node.

tences within our framework.

Now let us consider how other problems for the GKPS-style treatment can be solved in our framework. First, we have seen in section 2 that such expressions as *waste of time* (cf. (40)) cause a problem for the *breeze*-type construction rule (39). Under our approach, we can simply stipulate that these expressions themselves have the !GL! feature as a property of idiomatic expressions. Then they need not be reanalyzed as zero-bar categories:

(67)



As we have seen in (40a), the *waste of* ··· expression is somewhat productive. So, we cannot treat such expressions as zero-bar lexical items. But these expressions should be listed in the lexicon because they have the characteristics of triggering a filler-gap construction.

Second, the problems which occur in sentences with *too easy* (53a) and *easy enough* (54a) can also be solved under the present approach. In each of these sentences there are two lexical items which have the !GL! feature. But they show the characteristics of the *easy*-type construction rather than the *too/enough* constructions. This means that the GAP is licensed by the GL which originates from *easy* rather than that from *too/enough*. Notice that the gap is obligatory in the *easy*-type construction and it is optional in the *too/enough* constructions. In other words, the LICENSOR feature of *easy* must license a GAP to derive a grammatical sentence. It is an obligatory feature, !GL!. But the feature of *too/enough* need not be used obligatorily, i.e., it is an optional feature, !(GL)!.

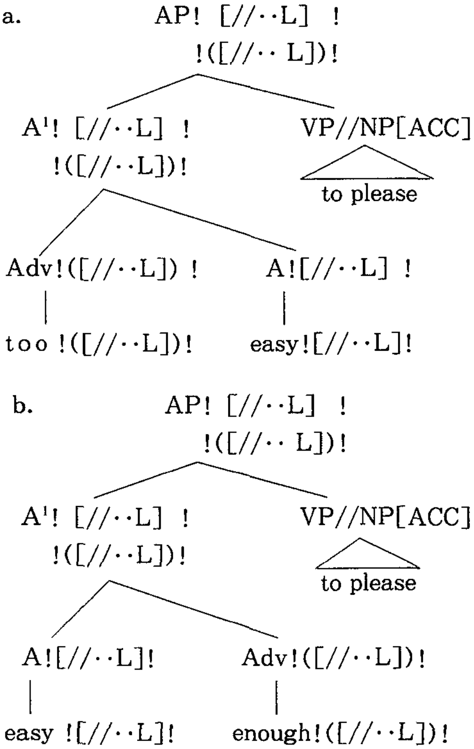
We can implement the fact here by positing a principle of the following sort:

(68) When there is more than one LICENSOR feature among the daughters of a node, an obligatory LICENSOR feature goes into the stack later than an optional LICENSOR feature.

Features which are put into the stack later come out of it earlier, an arrangement that uses the stack to capture the precedence of obligatory elements of grammar over optional ones. Obligatory-over-optional precedence ensures that no obligatory features are left unused (cf. Hastings (1976)).

Then we can analyze the sentences in (53a) and (54a) as follows:

(69) a.



b.

In these structures, the stack of the AP node has two LICENSOR features. But the obligatory feature, which originates from *easy*, is always put on the top of the stack due to priciple (68). The GAP can be licensed only by this obligatory feature, and the optional LICENSOR feature will not be used here. Thus, we can provide a natural account for the fact that sentences (53a) and (54a) have the characteristics of the *easy*-type construction rather than the *too/enough* constructions.

In section 2, we noticed that one of the most serious problems with the rule-based approach of GKPS is that it fails to capture the fact that the same lexical items induce the same kind of gapping in different constructions, with reference to the data in (57-60). The strongest argument for the present GL approach lies in dealing with these data. Our approach is based on the idea that each lexical item is the source of gap-licensing property in an A-positon filler-gap construction. We need not worry about accounting for the relationship between different constructions which have the same gap-licensing lexical item. They show the same gapping characteristics simply because they have the same lexical item, which triggers the constructions concerned.

Further support for our approach comes from the analysis of the following examples:

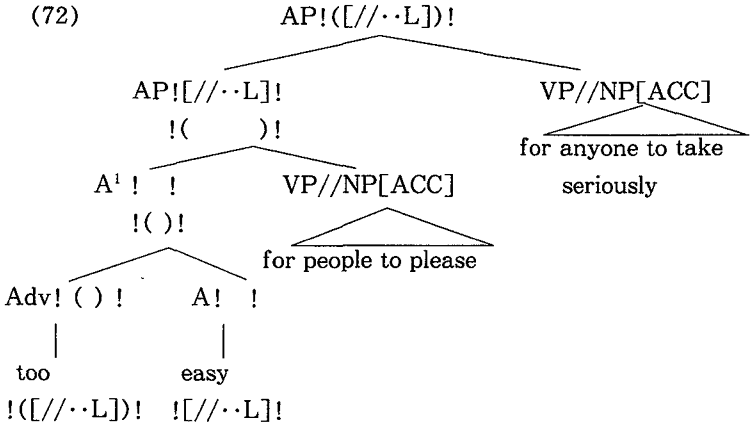
(70) a. *Robin and Pat are too anxious for everyone to like *e* to be comfortable around *e*.
b. Robin and Pat are too anxious for everyone to like them to be comfortable around *e*.

(71) a. John is too easy for people to please *e* for anyone to take *e* seriously.
b. *John is too easy for people to please him for anyone to take *e* seriously.

In (70) we need an object NP of *like* because *anxious* is not a gap-licensing lexical item. But there must be a gap NP after *please* in (71) because *easy* is such a lexical item.

Let us consider how we can license each of the two gaps in sentence (71a). The structure for it would be as follows:

(72)



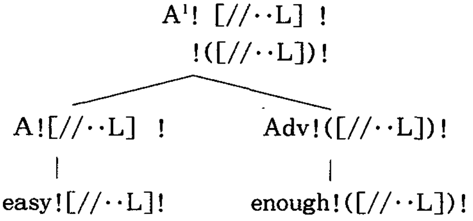
The gap after *please* is licensed by the obligatory GL from *easy* and the gap after *take* by the optional GL from *too.*

Under this analysis, we can account for the following sentence without any extra cost:

(73) Robin isn't easy enough to talk to e_1 to approach e_2 directly on this problem.

This sentence shows a kind of crossed dependency. The first gap (e_1) is licensed by *easy* and the second one (e_2) by *enough.* Under the approach here, we need not assume such expensive mechanisms as 'flattening' or 'liberation'. The structure for this sentence is the same as (72) except for the following part:

(74)



Principle (68) ensures that the obligatory [GL] comes on the top of the stack on A^1 regardless of the linear order of its daughters (cf. (69)).

We might be able to generalize [⟨GAP XP⟩ LICENSOR] AS [α LICENSOR][17]. Let us consider the following sentence (Gazdar 1988: 77):

(74) This fence is |so much|$_c$|too much|$_b$ high|er|$_a$ |than|$_a$ that one |for|$_b$ me to even consider climbing it |that|$_c$ it's simply incomprehensible to me that Mary would try to get me to do it.

In this sentence *so* (*much*) is related to a *that*-clause, *too* (*much*) is related to a *for*-clause, and *-er* is related to a *than*-phrase. We can then say that the *that*-clause, *for*-clause and *than*-phrase are licensed by *so* (*much*), *too* (*much*) and *-er*, respectively. That is, *so* (*much*), *too*

[17] Under this generalization, *easy* and *too/enough* have ![VP//XP LICENSOR]! and !([VP(//XP) LICENSOR])!, respectively. We should also revise principle (62) to accommodate this change. Notice that, in the case of the feature for *too/enough,* not only the //XP but the whole feature is optional, because there are expressions with these words which do not have any VPs, e.g. *it's too hot/hot enough.*

(*much*) and *-er* have [*that*-clause LICENSOR], [*for*-clause LICENSOR] and [*than*-phrase LICENSOR], respectively, in the lexicon[18]. With the left-branching tree structure proposed in Gazdar (1988: 78), we can effectively account for, not only these dependencies but also the fact that the string *so much too much higher* forms a constituent in this sentence.

4. Conclusion

We have observed seven different constructions which show (unbounded) dependency between filler and gap. If we are going to account for all of these constructions within the GKPS framework, we may need several additional mechanisms, which cannot be always independently motivated. One of the problems of this approach is caused by the fact that the triggering element of the construction concerned is not always the head of the local structure. In cases where it is not the head of the structure, the idea that some particular set of lexical items is related to the construction cannot be effectively implemented within the rule because that fact is represented through the SUBCAT feature value of the head. A more serious problem is that it fails to express the fact that the same lexical items in different constructions carry their gap-licensing properties with them.

I have proposed a new gap licensing mechanism, which introduces a stacked FOOT feature [⟨GAP XP⟩ LICENSOR], and principle (62), which regulates its behavior. Even though the source of the LICENSOR feature is a particular lexical item, the (unbounded) dependency between the filler NP and the accusative gap NP is accounted for by the characteristics of such FOOT features as GAP and LICENSOR, and by such syntactic mechanisms as the FFP and Principle (62). This system provides a unified account for all the varieties of lexically triggered fillergap dependency constructions, which do not necessarily have structural similarities. The lexical origin of the LICENSOR feature is the main innovation for an effective account of the data which are problematic for a rule-based approach.

[18] John Nerbonne told me that he has a similar idea as the present program of licensing mechanisms. Specifically, he pointed out to me that the comparative morpheme has the property of licensing/requiring a *than*-phrase in the comparative construction.

As an appendix to this paper, I will briefly consider the analyses of *tough*- and similar constructions in Browning (1987), a Government and Binding (GB) framework. She provides an account of "Null Operator Constructions," which comprise not only lexically triggered constructions but also structurally oriented constructions such as relatives and clefts. Null operators play an important role in accounting for these constructions. Three types of constraints are proposed to regulate their behavior : "constraints governing the licensing of the null operator itself, the licensing of the variable created by null operator movement and the licensing of the null operator chain as a whole (p. 2)." The most important motivations for the null operator movement approach are that these constructions obey Subjacency and license parasitic gaps.

However, there seem to be several problems which cannot be easily solved in this framework. First, in Browning's approach, the ultimate source of gap licensing in a *tough*- construction is a special thematic role assignment pattern of the *tough*- adjective. She assumes that this adjective has only an internal theta-role to assign and hence the matrix subject is not licensed by theta-role assignment (p. 147). However, this approach predicts that the following sentence would be grammatical :

(75) *John is possible to talk to.

The adjective *possible* cannot occur in a *tough*-construction even though *impossible* can (cf. (5) and footnote 13). But *possible* seems to have the same theta-role assignment pattern as other *tough*-adjectives :

(76) a. To talk to John is possible
b. it is possible to talk to John.

From these data, we can see that there is no thematic relation between *possible* and the matrix subject.

Second, let us consider the data in (57)-(60). These data show that the same lexical items in different constructions carry their gap-licensing properties with them. There do not seem to be straightforward ways of capturing similarities among these constructions. For example, in the following set of sentences,

(57) a. John is tough to argue with.
b. John is a tough man to argue with.

the adjective *tough* in (57b) is not even a theta-role assigner because it is a prenominal adjective, while this adjective being a theta-role assigner is crucial in (57a).

From the discussions for the data in (53)-(54), we have seen that the following sentences,

(53) a. John is too easy to make friends with.
(54) a. John is easy enough to make friends with.

show the characteristics of *tough*-constructions rather than *too/enough* constructions. However, under Browning's approach, these sentences can only be analyzed as *too/enough* constructions because the Deg(ree) words *too* and *enough* are the (obligatory) cue words for assigning special structures for these constructions.

Lastly, let us look at the following sentences which contain two gap-licensing elements and two gaps :

(71) a. John is too easy for people to please *e* for anyone to take *e* seriously.
(73) Robin isn't easy enough to talk to *e* to approach *e* directly on this problem.

Each of the two gaps is licensed by one of the two gap-licensing lexical items. One might argue that parasitic gaps are involved here. But this is not true :

(77) *John is very easy for people to please *e* for anyone to take *e* seriously.
(70) a. *Robin and pat are too anxious for everyone to like *e* to be comfortable around *e*.

If one of the two gaps is a parasitic gap, these sentences would be grammatical because there is only one gap-licensing element in a parasitic gap construction. These data indicate that the two gaps in (71) and (73) are licensed by different lexical items.

Therefore, the problem is how to combine the structure for *tough*-constructions and that for *too* constructions to get the structures for the sentences in (71) and (73). It seems to be almost impossible to get such a structure for sentence (73) because the dependencies between triggers and gaps are crossed rather than nested. The only possible structure for sentence (71) would be as follows :

(78) …[too [easy for people to please *e*] for anyone to take *e* seriously]

However this structure cannot capture the fact that *too easy* forms a constituent (cf. (55), (56) and (74)). Another problem for this structure is that we need to revise the mechanisms to license the "agreement chain" to ensure the connection between the inner gap (i. e. the gap after *please*) and the matrix subject. Notice that the clause which contains this gap is separated from the matrix subject by another clause (i. e. the *too* clause).

References

Browning, Marguerite (1987) *Null Operator Constructions*, Ph. D. Dissertation, MIT.

Chomsky, Noam (1977) 'On wh-Movement,' *Formal Syntax*, ed. by P. Culicover, T. Wasow and A. Akmajian, 71-132, Academic press.

Dowty, David and Pauline Jacobson (1989) 'Agreement as a Semantic Phenomenon,' *Proceedings of the Fifth Eastern States Conference on Linguistics*, 1-17, The Ohio State University.

Gazdar, Gerald (1988) 'Applicability of Indexed Grammars to Natural Languages,' *Natural Language Parsing and Linguistic Theory*, ed. by U. Reyle and C. Rohrer, 69-94, D. Reidel Publishing Company.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum and Ivan Sag (1985) *Generalized Phrase Structure Grammar*, Blackwell.

Hastings, Ashley (1976) 'On the Obligatory-Optional Principle,' *The Application and Ordering of Grammatical Rules*, ed. by Andreas Koutsoudas, 294-319, Mouton.

Hukuri, Thomas and Robert Levine (1987) 'Rethinking Connectivity in Unbounded Dependency Constructions,' *Proceedings of the Sixth Annual West Coast Conference on Formal Linguistics*, 91-102, Stanford University.

Hukuri, Thomas and Robert Levine (1991) 'On the Disunity of Unbounded Dependency Constructions,' *Natural Language and Linguistic Theory* 9, 97-144.

Jacobson, Pauline (1984) 'Connectivity in Phrase Structure Grammar,' *Natural Language and Linguistic Theory* 1, 535-81.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech and Jan Svartvik (1985) *A Comprehensive Grammar of the English Language*, Longman.

Zwicky, Arnold (1987) 'Slashes in the Passive,' *Linguistics* 25, 639-69.

Department of Linguistics
The Ohio State University
1841 Millikin Road #204
Columbus, OH 43210
U. S. A.