Disjunction without Tears

Allan Ramsay

Department of Computer Science
University College Dublin
Belfield, DUBLIN 4,
Ireland

It is now common practice to use structured sets of features to describe syntactic structures, and to compare such structured sets by some form of unification. It is extremely tempting to try to encode disjunctive information within this framework; unfortunately, general unification of disjunctive structures is NP-complete, which is not a desirable property for the basic operation involved in comparing descriptions. We note that certain kinds of disjunctive specifications can be converted to conjunctive ones. This enables us to describe a restricted set of disjunctive phenomena without incurring the costs associated with general disjunctive unification.

1 Unification and Syntactic Description

We assume that the use of lattices and directed acyclic graphs (DAGs) for describing properties of syntactic structures, and of unification for combining such descriptions, is too well known to require justification at this point. A wide variety of notations have been used in this context—PROLOG terms, PATR-II specifications, FUG functional descriptors, GPSG category specifications, and so on (Pereira and Warren 1980; Shieber 1986; Kay 1985; Gazdar et al. 1985). The discussion below is couched in a FUG-like notation for concreteness, but the argument is equally applicable to any of the other standard formalisms (except perhaps GPSG).

We can use our notation for characterizing objects in terms of their properties. We will generally be talking about linguistic objects such as words or phrases, though there is nothing that restricts this kind of description to such objects. We start by considering the word *providing.* We know various things about this word, including the fact that it is a present participle. We can represent this specific piece of information with the descriptor shown in Figure 1. We have included some *negative* information in this description, namely the fact the word in question is not tensed, is not an infinitive, and is not a past participle or a passive one. This information is generally omitted from FUG specifications, since it seems to be redundant. In fact it is not redundant unless we have specified elsewhere that being a participle, being tensed, and being an infinitive are mutually exclusive, and similarly for present, past, and passive participles. We will continue to include this kind of information locally within specifications, rather than appealing to some externally specified set of constraints. In subsequent descriptors we will generally write $\neg TENSED$ for $TENSED = \bot$, though we will continue to spell out positive specifications such as $PRESENT = T$ in full.

2 Two uses of Disjunction

The kind of representation in Figure 1 is appropriate when some specific piece of information about some item is known—in Figure 1, for instance, the fact that the word in question is a present participle. It often happens, however, that we know that some item can be described in several ways, but that we are not sure which is correct in the present circumstances. Consider for instance the word *provided.* This might be a past tense form, or a past participle, or a passive participle. There is nothing about the word itself that can help us decide which it is, though in any actual context only one of these descriptions will be appropriate.

(1) He provided us with everything we needed.
(2) He has provided us with everything we needed.
(3) Everything we needed has been provided.

We could produce three descriptions of the kind in Figure 1, one for each case. If we did this, however, we would find ourselves repeating all sorts of information—the fact that it's a verb, for instance, plus whatever we know about its subcategorization frame, and so on. It is therefore tempting to try to adapt our notation so that it allows disjunctive specifications for feature values, as shown in Figure 2. Figure 2 represents a description of an item that is either a past tense verb or a past or passive participle, with the curly bracket {used to indicate a range of options. This kind of

$$SYNTAX = \left[\begin{array}{l} MAJOR = \left[\begin{array}{l} CAT = VERB \\ BAR = 0 \end{array}\right. \\ MINOR = \left[HEAD = \left[FORM = \left[\begin{array}{l} TENSED = \bot \\ PARTICIPLE = \left[\begin{array}{l} PRESENT = T \\ PAST = \bot \\ PASSIVE = \bot \end{array}\right. \\ INFINITIVE = \bot \end{array}\right.\right.\right. \end{array}\right.$$

Figure 1 Present Participle, e.g. *providing.*

disjunctive specification is widespread in unification grammer—the curly bracket, for instance, is standard notation in FUG, and most other notations provide some way of talking about disjunction. Kasper and Rounds (1986), among others, have taken up the question of exactly what such notations mean. We are more interested here in investigating the circumstances under which they are really necessary, and in trying to remove them wherever we can.

Much the same sort of issue arises when we consider syntactic rules, particularly when we consider rules representing information about subcategorization frames. Consider, for instance, the interpretation of the verb *be* as an auxiliary. *Be,* as an auxiliary, can be combined with either a VP whose main verb is a present participle or one whose main verb is a passive participle. We might try to represent this information with the rule shown in Figure 3. Figures 2 and 3 are very perspicuous. Figure 2 describes a word that is a past tense verb, a past participle, or a passive participle. Figure 3 describes a grammatical constraint, namely that *be* may be followed by a VP whose main verb is either a present participle or a passive one. The placeholder $?H$ is used to indicate that the form of the VP that results from combining an instance of *be* with a suitable complement has the same *HEAD* features as the instance of *be.* Unfortunately, the introduction of disjunctions into our descriptions has drastic effects on the computational properties of unification, particularly when it is combined with the use of placeholders or other ways of specifying reentrance. To see this, suppose we want to see whether some VP whose main verb has the properties ascribed to *provided* fits the constraints imposed by *be* (in other words, we are trying to

$$SYNTAX = \left[\begin{array}{l} MAJOR = \left[\begin{array}{l} CAT = VERB \\ BAR = 0 \end{array}\right. \\ MINOR = \left[HEAD = \left[FORM = \left\{\begin{array}{l} \left[\begin{array}{l} TENSED = \left[\begin{array}{l} \neg PRESENT \\ PAST = T \end{array}\right. \\ \neg PARTICIPLE \\ \neg INFINITIVE \end{array}\right. \\ \left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ PAST = T \\ \neg PASSIVE \end{array}\right. \\ \neg INFINITIVE \end{array}\right. \\ \left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \\ PASSIVE = T \end{array}\right. \\ \neg INFINITIVE \end{array}\right. \end{array}\right.\right.\right. \end{array}\right.$$

Figure 2 Underdetermined Verb Form, e.g. *provided.*

$$SYNTAX = \left[\begin{array}{l} MAJOR = \left[\begin{array}{l} CAT = VERB \\ BAR = 2 \end{array}\right. \\ MINOR = \left[HEAD = ?H \right. \end{array}\right.$$

$$\Longrightarrow$$

$$SYNTAX = \left[\begin{array}{l} MAJOR = \left[\begin{array}{l} CAT = VERB \\ BAR = 0 \end{array}\right. \\ MINOR = \left[HEAD = ?H \right. \end{array}\right.$$

$$SYNTAX = \left[\begin{array}{l} MAJOR = \left[\begin{array}{l} CAT = VERB \\ BAR = 2 \end{array}\right. \\ MINOR = \left[HEAD = \left[FORM = \left\{\begin{array}{l} \left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} PRESENT = T \\ \neg PAST \\ \neg PASSIVE \end{array}\right. \\ \neg INFINITIVE \end{array}\right. \\ \left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \\ PASSIVE = T \end{array}\right. \\ \neg INFINITIVE \end{array}\right. \end{array}\right.\right.\right. \end{array}\right.$$

Figure 3 Underdetermined Rule—Subcategorization Frame for *be.*

parse (3)). At some point we will have to compare

$$\left[\begin{array}{l} TENSED = \left[\begin{array}{l} \neg PRESENT \\ PAST = T \end{array}\right. \\ \neg PARTICIPLE \\ \neg INFINITIVE \end{array}\right.$$
$$\left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ PAST = T \\ \neg PASSIVE \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$
$$\left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \\ PASSIVE = T \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$

with

$$\left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} PRESENT = T \\ \neg PAST \\ \neg PASSIVE \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$
$$\left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \\ PASSIVE = T \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$

We will have to try various possibilities—is

$$\left[\begin{array}{l} TENSED = \left[\begin{array}{l} \neg PRESENT \\ PAST = T \end{array}\right. \\ \neg PARTICIPLE \\ \neg INFINITIVE \end{array}\right.$$

the same as

$$\left[\begin{array}{l} \neg TENSED \\ PARTICIPLE = \left[\begin{array}{l} PRESENT = T \\ \neg PAST \\ \neg PASSIVE \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$

and so on. Eventually we will compare the part of the description of *solved* that says it might be a passive participle with the part of the rule that says that a VP whose main verb is a passive participle is acceptable here. At this point we will realize that the given text fits the rule, but only after trying out numerous options that led nowhere.

Worse than this, there may be several locally compatible sets of options, only one of which may lead to a globally coherent description of the complete text being examined. If this is a possibility then the process of unifying two structures turns out to be NP-complete, a most undesirable consequence of our decision to allow disjunctive feature descriptions.

3 Extra Constraints

If we look again at the descriptions in Figures 2 and 3 we see that we know rather more about the *FORM* part of these descriptions than is explicitly stated. In particular, we know that the *FORM* of any verb whatsoever is drawn from the range of options shown in Figure 4. Given this extra information, we see that a disjunctive description such as the one we have been using for *provided* can be replaced by a conjunctive one containing nothing but negative information. The descriptions of the *FORM* of the lexical item *provided* and the complement of *be,* for instance, can be replaced by the following purely *conjunctive* descriptions:

$$\left[\begin{array}{l} TENSED = [\neg PRESENT \\ PARTICIPLE = [\neg PRESENT \\ \neg INFINITIVE \end{array}\right.$$

and

$$\left[\begin{array}{l} TENSED = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \end{array}\right. \\ PARTICIPLE = \neg PAST \\ \neg INFINITIVE \end{array}\right.$$

The equivalence depends on the fact that in any specific case *FORM* has exactly one of the values given in Figure 4.

$TENSED = [PRESENT = T$ $\quad PARTICIPLE = [PRESENT = T$

$TENSED = [PAST = T$ $\quad PARTICIPLE = [PAST = T$

$INFINITIVE = T$ $\quad PARTICIPLE = [PASSIVE = T$

Figure 4 Possible Values for FORM for Verbs.

If we know what values it doesn't have, we can infer the range that the value it does have must be drawn from. When we attempt to unify these two specifications, we find that they lead to the following more precise description:

$$\left[\begin{array}{l} TENSED = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \end{array}\right. \\ PARTICIPLE = \left[\begin{array}{l} \neg PRESENT \\ \neg PAST \end{array}\right. \\ \neg INFINITIVE \end{array}\right.$$

The only way for this to be compatible with the general constraint that the value of *FORM* must be drawn from the values in Figure 4 is if it is in fact a passive participle. We have obtained the required effect without complicating our unification algorithm, simply by making use of the extra information that the value in question must be drawn from a known finite range. Note that we do not need to refer explicitly to the information in Figure 4 when we want to know whether two specifications for *FORM* are compatible. Rather we have used this information to construct our specifications, which can be compared directly using ordinary unification.

Many of the situations that seem to call for disjunctive descriptions can be dealt with this way. The NP *the sheep* could be either third person singular or third person plural? Then describe it as not first person singular or first person plural or second person singular or second person plural. The pronoun *he* is nominative, whereas *it* may be either nominative or accusative? Then describe *he* as not accusative, and say nothing about *it.* When we can replace disjunctive descriptions by ones that embody a conjunction of negations, we can save a considerable amount of work, since our unification algorithm no longer needs to plod through a series of possible analyses, keeping track of the options that have been tried and possibly backtracking when some analysis that looked plausible leads to a dead end. We cannot hope to eliminate disjunction from our grammar entirely, since if we could then parsing would become a trivial deterministic task, which it does not look like becoming. We can, however, eliminate it in a lot of places where it looks as though it would be useful; which doesn't make parsing a trivial task, but it does mean that we can avoid doing more work than we really need.

4 Limitations

The example discussed above shows how we can replace disjunctive descriptions by conjunctive ones in two specific cases, namely in the description of the *FORMs* of lexical items and VP complements. We further suggested that this technique might be extendable to other uses of disjunctive specifications, such as the agreement properties of the NP *the sheep* or the case marking of the pronoun *it.* We have not, however, banished disjunction from the grammar of

English. We could hardly expect to, given that in some versions of unification grammar, e.g. FUG, the entire grammar is specified by a single disjunctive descriptor, which is required to unify with the description of any legal sentence. Just what are the limits of our technique?

The following two restrictions seem crucial. (i) The elements of the disjunction must pick out points in some finite partition of the space of possibilities, preferably a small finite partition. (ii) The disjunction must not contain any placeholders (reentrance conditions). Consider for instance the word *changes*. For the sake of argument we will assume that this item has exactly two interpretations, as a plural noun and as a third person singular verb. We could represent the fact that it can be seen either as a noun or as a verb as follows:

$$SYNTAX = \begin{bmatrix} MAJOR = \begin{bmatrix} CAT = \begin{bmatrix} \neg ADJECTIVE \\ \neg ADVERB \\ \ldots \\ \neg PREPOSITION \end{bmatrix} \\ BAR = 0 \end{bmatrix} \\ MINOR \end{bmatrix}$$

This indicates that we can specify the major category of *changes* in terms of what it is not, though it does seem that this may be too cumbersome to be worthwhile. The real problem, however, is that the *MINOR* features will be expected to specify some value for *AGR;* but the value of *AGR* for this word depends on whether it is in fact a noun or a verb—it is plural as a noun, but singular as a verb. We simply cannot deal with this within our framework, and if we try to adapt the framework to accommodate it, then all the computational complexity will return.

This is particularly significant when we consider information about subcategorization. It is widely, and very sensibly, argued that as much information as possible about constituent structure should be associated directly with lexical items. FUG, for instance, permits the description of a lexical item to contain a pattern that describes its complements, and UCG (Calder 1988) even goes so far as to specify that for many lexical items the major category *is* the subcategorization frame. It is clear that the range of possible subcategorization frames cannot effectively be expressed in terms of a small finite set; and even if it could, our technique would be inapplicable because of the complex dependencies between the choice of subcategorization frame and the values of other features. This is the critical restriction on our technique—that it cannot be used to eliminate the disjunctions that arise from different subcategorization frames and different phrase structure rules. Our aim is to replace *unnecessary* disjunctions, not to eliminate them from syntactic description entirely.

We end by returning to our earlier remark that "the argument is equally applicable to any of the other standard formalisms (*except perhaps GPSG*)." Our analysis of our main example depended on the fact that we have general knowledge about the range of possible values for the feature *FORM,* which we used to modify local descriptions of this feature. In GPSG, however, local information about feature structures is not complete, since there is a full default logic specifying possible restrictions on what values are possible or mandatory in various circumstances. It seems likely that the interactions between this logic and our use of general knowledge about features will interact in ways that are at best hard to predict, and may even make our approach completely infeasible.

REFERENCES

Calder J., Klein, E., and Zeevat, H. 1988 Unification Categorial Grammar: A Concise, Extendable Grammar for Natural Language Processing. *Proceedings of the 12th International Conference on Computational Linguistics,* 83–86.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. 1985 *Generalised Phrase Structure Grammar.* Basil Blackwell, Oxford, U.K.

Kasper, R. and Rounds, W. 1986 A Logical Semantics for Feature Structures. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics* 24, 257–266.

Kay, M. 1985 Parsing in Functional Unification Grammar. In: Dowty, D.R., Karttunen, L., and Zwicky, A.M. (eds.) *Natural Language Parsing.* Cambridge University Press, Cambridge.

Pereira, F.C.N. and Warren, D.H.D. 1980 Definite Clause Grammars for Language Analysis—A Survey of the Formalism and a Comparison with ATNs. *Artificial Intelligence* 13(3): 231–278.

Shieber, S. M. 1986 *An Introduction to Unification-Based Approaches to Grammar.* University of Chicago Press, Chicago.