Functional Constraints in Dependency Grammar

Tom B.Y. LAI[1] HUANG Changning[2]

We follow the tradition of projective syntax in Dependency Grammar and emulate dependency rules with re-write rules in which head daughters are identified (in terms of unification) with the parents. Constraints defined in terms of grammatical functions are then used to take care of non-projective grammatical restrictions. Binary rules in this formalism are not a mechanism to introduce binary-branching phrasal structures, but license dependents of a governing head one after another in linear order. While lexical subcategorization information takes care of complements of a head word, global dependency rules generate adjunct dependents.

Projective Dependency Grammar

In Dependency Grammar (TESNIÈRE 1959), words contract binary, asymmetrical governor-dependent relationships as shown below for the sentence *That man lives in London*:

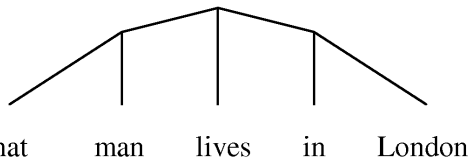

Figure 1

ROBINSON (1970) has posited four axioms of well-formedness to constraint syntactic dependency structures:

(1a) one and only one element is independent;

(1b) all others depend directly on some element;

---

[1] City University of Hong Kong and Tsinghua University, Beijing.
[2] Tsinghua University, Beijing, and Microsoft, Beijing.

(1c) no element depends directly on more than one other;

(1d) if A depends directly on B and some element C intervenes between them (in linear order of string), then C depends directly on A or on B or some other intervening element.

These axioms require that syntactic dependency should be *projective* in the sense that dependency structures like Fig. 2, in which branches cross one another (violating (1d)) and a word depends on more than one governor (violating (1c)), are not allowed.

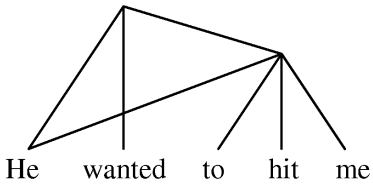

Figure 2

HAYS (1964) has used *dependency rules* to generate dependency structures with these properties:

(2a) X(A,B,C, ... ,H, * ,Y, ... ,Z)

(2b) X(*)

(2c) * (X)

Rule (2a) states that the governing *auxiliary alphabet* (i.e. *symbol* or *category*) X has dependents A, B, C, ..., H, Y, ..., Z (in this order) and that X itself (the governor, denoted by *) is situated between H and Y. Rule (2b) says that the *terminal alphabet* X occurs without any dependents. In (2c), X occurs without any governor, i.e. it is the *main* or *central* element.

Compared to Robinson's axioms, Hays' rules are more restrictive about the linear order of dependents of a word. GAIFMAN (1965) has established that a *projective* Dependency Grammar obtained in this way is 'equivalent' to a phrase structure grammar in the sense that every structure attributed to a string by one grammar has a corresponding structure attributed by the other.

Functional Annotation in Dependency Rules

Gaifman, Hays and Robinson's approach to syntactic dependency require that there should be no multiple governors, no mutual dependency and no crossing branches. It is however known that there are indeed non-projective grammatical phenomena in human languages. To account for these phenomena, followers of Dependency Grammar like HUDSON (1982), MEL'CUK (1988), STAROSTA (1988) and HAJICOVA (1991), have allowed multiple-headedness and non-projectivity in their frameworks.

Many computational linguists have however based their work on projective syntactic dependency (e.g. COURTIN & GENTHIAL 1998, NAGAO 1993, YUAN & HUANG 1992). Indeed, working on Chinese, we (LAI & HUANG 1994, 1995, 1998a, 1998b) have tried to maintain single-headedness and projectivity in the syntactic component by leaving morphological considerations to another level of analysis and by introducing grammatical constraints on projective syntactic dependency structures (see also BOURDON et al. 1998).

Dependency Grammarians have put labels like 'subject' and 'object' on dependency links between words in a linguistic expression. We (LAI & HUANG 1984) note that such functional information can be added to Hays' dependency rules (cf. KAPLAN & BRESNAN 1982):

(3a) X(A(fa), B(fb), ..., * , ... Z(fz))

or, in an equivalent notation:

(3b) X(A, B, …, X, …, Z)
X.fa = A
……..
X.fz = Z

In the first line of (3b), the X outside the brackets is like the left hand side of a re-write rule, and (A, B, …, X, …, Z) is like the right hand side. However, there is one important difference between the two kinds of rules. In a phrase-structure re-write rule, if two identical symbols are found on both sides of the rule, they are different tokens of the same *type*. For a dependency rule like (3b), however, the two occurrences of X are one and the same *token*.

We emulate these rules with re-write rules delimited by unification-based constraints. Adapting the PATR formalism (SHIEBER 1986; GAZDAR & MELLISH 1989) for our use, syntactic structures like the following are produced:

(4) [tv, [[n, [john]], saw, [n, [mary]]]]

In these dependency structures, there are no 'intermediate phrasal' nodes. And it is in this sense that our formalism is dependency-based rather than phrase-structure-based.

A functional structure accompanies the syntactic structure in (4). It specifies that *John* and *Mary* are the 'subject' and 'object', respectively, of the head word *saw*.

In our more recent work (LAI & HUANG 1998b, 1999a), rules like (3b) have been made 'binary' as follows (with 'functional annotations'):

(5a) X(X, Y)

(5b) X(Y, X)

These very general rules will obviously over-generate. To avoid this, lexical entries are equipped with subcategorization lists (cf. POLLARD & SAG 1994). For example:

(6) '*give*'
subcat.left = [subj::n]
subcat.right = [iobj::n, obj::n]

Complements licensed by a governing head word are divided between two subcategorization lists depending on whether they occur to the left or to the right of the word. Elements in these lists are arranged so that consecutive firing of the 'binary' rules generates the complements in their normal word order.

Details of the implementation are described in LAI & HUANG (1998b, 1999b). For languages with relatively fixed word order like English, complements subcategorized by a governing word are generated one after another according to the order in which they are specified in the subcategorization lists. For languages with free word order, the way in which the subcategorization lists are accessed can be manipulated (LAI & HUANG 1999a).

Adjunct dependents are provided by global rules:

(7a) X(X, Y)
X.adjunct = Y

(7b) X(Y, X)
X.adjunct = Y

which contain additional constraints to make sure that the adjuncts of the correct syntactic categories are attached to a head.

With subcategorization information residing in the lexicon, this approach to Dependency Grammar is lexicalist in nature. It is not a phrase-structure grammar (e.g. KAPLAN & BRESNAN 1982; POLLARD & SAG 1994) in that no phrasal projections are postulated. We use binary rules, but the 'domain' of a governing head word is not a tree with an excessive number of binary-branching nodes. It is a flat constellation of the head word and its dependents. And, element order in the subcategorization lists is more or less a reflection of 'normal' word order, which is also different from POLLARD & SAG (1994).

Non-projectivity and 'Movement'

A discussion of treatment of non-projectivity, often the result of 'movements' in transformation-based formalisms, is in order.

Complements of a governing word may take up 'unusual' positions in certain syntactic constructions. For example, the English verb *give* can be used in different syntactic patterns like:

(8a) *They give John a book.*

(8b) *They gave a book to John.*

The usage of *give* in (8a) can be accounted for by the subcategorization information given in (6). The usage in (8b) (apart from the morphological variation between the two word forms *give* and *gave*) requires another subcategorization frame:

(9) '*give*'
subcat.left = [subj::n]
subcat.right = [obj::n, pobj::p(to)]

This kind of subcategorization property variation within the domain of a head word involves 'movement', but does not involve non-project-

ivity. In our computational treatment of this, multiple subcategorization lists are used.

Satisfaction of the subcategorization properties of a word outside of its immediate 'domain' does pose a threat to non-projectivity. 'Control' verbs provide a good example.

(10) *He wanted to hit me*.

In (10), the subject of the verb *hit* is not found in its immediate domain, but in the domain of the matrix verb *wanted*. The point here is that the subcategorization property of the higher control verb *wanted* determines that the subcategorization requirements of the lower verb *hit* should be left unsatisfied within its immediate domain.

In our treatment of control verbs, control information is added to elements of the subcategorization lists. The right subcategorization list of *wanted* is:

(11) [obj::n::place_holder, comp::v(to)::ctr(subj,subj)]

The dependency rule to work with this is:

(12) H(H, X)
X.cat = Cat
H.subcat.right = List
{topstack(Dep, List)}
Dep = F::Cat::Ctr
control(Ctr, H, X)
H.fs.f(F) = X

In addition to 'functional' constraints, procedural actions, indicated by curly braces, can also be attached to a dependency rule. *topstack* is the action that manipulates subcategorization lists. *control* takes action according to the value of the indicator *Ctr*. If *Ctr* is a place-holder indicating that the governor is not a control verb, nothing is done. Otherwise, according to the information contained in *Ctr*, appropriate action is taken on the subcategorization list of the embedded verb and the functional relation between the controller and the controllee is established.

While control is a lexically-motivated syntactic phenomenon, topicalization (e.g. *Linguistics I know he likes*) is non-lexical by nature. In our Dependency Grammar model, this is taken care of by a global rule. The

topicalization rule is rather similar to the adjunct rules. Either the topicalized element or the highest predicate in the utterance can be considered the governor. In either case, a feature is instantiated at the highest predicate and passed down to its own dependent predicates as necessary and appropriate, and the complement-generating dependency rules are further modified accordingly so that an appropriate dependent of the predicate can be gapped or made pronominal. This treatment of topicalization also illustrates how distant-dependency (some of which, unlike topicalization, are lexically-motivated, e.g. by relative pronouns) is dealt with.

A brief look at clitic proforms is also interesting. Consider the clitic proform *en* in the following French examples:

(13a) J'*en* connais la fin. (adapted from MEL'CUK 1988, 37)
I EN knew the end
I knew the end of it

(13b) … qui *en* ont demandé … (SARTRE 1938, 21)
… who EN have asked …
… who have asked for it …

(13c) … je regrette d'*en* avoir tant dit. (SARTRE 1938, 161)
… I regret about EN have so said
… I regret that I have said so about it.

(13d) Il y a rien à *en* dire. (SARTRE 1938, 13)
There-is nothing to EN say
There is nothing to say about it

(13e) Je ne peux plus *en* douter. (SARTRE 1938, 17)
I NEG can NEG EN doubt
I cannot not have doubt about it

In (13a), *en* (proform for the preposition *de* and its object) stands for something that is dependent on the dependent *la fin* of the predicate *ont demandé*, and in (13b), *en* is a direct dependent of the predicate. (13c) to (13e) show that *en* is not raised further up to the domain of a higher predicate. This is thus a *local* phenomenon that does not involve long-distance dependency.

Pronoun clitic ‘movement’ in Czech is non-local:

(14) Tuto knihu jsem *se mu* rozhodl dát k narozeninám.
This book I-have refl him decided give to birthday
(HOLAN et al. 1998)

In (14), the pronoun clitic *mu* is not attached to the predicate *dát* that subcategorizes for it, but is raised to attach to the higher predicate *rozhodl*. This tells us that pronoun clitics in Czech is a non-local phenomenon that involves long-distance dependency.

Proform clitics in French and in Czech can both be dealt with using global rules like topicalization, though they are different, in the two languages, with respect to the local versus long-distance criterion. Further study of their treatment taking into account further contextual and positional constraints should be interesting.

Conclusion

We work with a Dependency Grammar model with annotated complement and adjunct rules that generate dependents of a governing word one after another. In this lexicalist model, we make use of multiple subcategorization lists and global dependency rules to generate dependent elements realized within and outside the immediate domain of the subcategorizing word. Computational implementation has been done using a convenient context-free grammar emulation.

We have been working on Chinese, a language with relatively fixed word order like English. After having started with analysing a small corpus of Chinese sentences, we are planning to scale up our efforts.

We would like our approach to be applicable not only to Chinese, but also to other languages in the world. So, we have also looked at some linguistic issues in several other languages. In this paper, the examples quoted are mainly from English.

References

BOURDON et al. (1998): M. B., L. DA SYLVA, M. GAGNON, A. KHARRAT, S. KNOLL and A. MACLACHLAN, A Case Study in Implementing Dependency-Based Grammars. In: *Proceedings of COLING-ACL '98 Workshop on Processing of Dependency-Based Grammars*, Montreal, 15 August 1998, 78-87.

COURTIN, J. and GENTHIAL, D. (1998): Parsing with Dependency Relations and Robust Parsing. In: *Proceedings of COLING-ACL '98 Workshop on Processing of Dependency-Based Grammars*, Montreal, 15 August 1998, 88-94.

COVINGTON, M.A. (1990): Parsing Discontinuous Constituents in Dependency Grammar. In: *Computational Linguistics* 16/4, 234-236.

GAIFMAN, H. (1965): Dependency Systems and Phrase-Structure Systems. In: *Information and Control* 8, 304-337.

GAZDAR, G. and MELLISH, C. (1989): Natural Language Processing in Prolog. Wokingham, UK: Addision Wesley.

HAJICOVA, E. (1991): Free Word Order Described Without Unnecessary Complexity. In: *Theoretical Linguistics* 17, 99-106.

HAYS, D.G. (1964): Dependency Theory: A. Formalism and Some Observations. In: *Language* 40, 511-525.

HELLWIG, P. (1986): Dependency Unification Grammar. In: *Proceedings of COLING '86*, 1986, 195-199.

HOLAN et al. (1998): T. H., V. KUBON, K. OLIVA and M. PLATEK, Two Useful Measures of Word Order Complexity. In: *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, Montreal, 15 August 1998, 21-28.

HUANG et al. (1992): H. C.N., YUAN C.F. and PAN S.M. (1992): Yuliaoku, Zhishi Huoqu He Jufa Fenxi (Corpora, Knowledge Acquisition and Syntactic Parsing). In: *Journal of Chinese Information Processing* 6/3, 1-6.

HUDSON, R. (1984): Word Grammar. Oxford: Blackwell.

— (1990): English Word Grammar. Oxford: Blackwell.

— (1994): Discontinuous Phrases in Dependency Grammr. In: *University College London Working Papers in Linguistics* 6, 89-124.

KAPLAN, R.M. and BRESNAN, J. (1982): Lexical-Functional Grammar: A. Formal System for Grammatical Representation. In: J.W. BRESNAN, (ed.), The Mental Representation of Grammatical Relations, Cambridge, MA: MIT Press, 173-281.

LAI, B.Y.T. and HUANG, C.N. (1994): Dependency Grammar and the Parsing of Chinese Sentences. In: *Proceedings of 1994 Kyoto Conference (Joint ACLIC8 and PAC FoCoL2)*, Kyoto, 10-11 Aug. 1994, 63-71.

— (1995): Single-Headedness and Projectivity for Syntactic Dependency. Paper presented at Linguistics Association of Great Britain Spring Conference, University of Newcastle, 10-12 August, 1995.

LAI, T.B.Y. and HUANG, C.N. (1998a): An Approach to Dependency Grammar for Chinese. In: Y. GU, (ed.), Studies in Chinese Linguistics, Hong Kong: Linguistic Society of Hong Kong, 143-163.

— (1998b): Complements and Adjuncts in Dependency Grammar Parsing Emulated by a Constrained Context-Free Grammar. In: *Proceedings of COLING-ACL '98 Workshop on Processing of Dependency-Based Grammars*, Montreal, 15 August 1998, 102-108.

— (1999a): Free Word Order in a Constraint-based Implementation of Dependency Grammar. In: *Proceedings of PACLIC13*, Taipei, 10-11 Feb. 1999, 161-168.

— (1999b): Unification-based Parsing Using Annotated Dependency Rules. Paper to be presented at NLPRS '99, Beijing, November 5-7, 1999, and to appear in the proceedings.

MAXWELL, D. (ms.): Unification Dependency Grammar.

MEL'CUK, I.A. (1998): Dependency Syntax: Theory and Practice. New York: State University of New York Press.

NAGAO, M. (1993): Current Status and Further Trends of Natural Language Processing. In: *Proceedings of KB&KS'93*, Tokyo, December 1993, 31-40.

POLLARD, C. and SAG, I. (1994): Head-Driven Phrase Structure Grammar. Chicago: University of Chicago Press.

ROBINSON, J.J. (1970): Dependency Structures and Transformation Rules. In: *Language* 46, 259-285.

SARTRE, J.-P. (1938): La nausée. Paris: Gallimard (Folio).

SHIEBER, S.M. (1986): An Introduction to Unification-Based Approach to Grammar. Chicago: Chicago University Press.

STAROSTA, S. (1988): The Case for Lexicase. London: Pinter.

TESNIÈRE, L. (1959): Eléments de syntaxe structurale. Paris: Klincksieck.

YUAN, C.F. and HUANG, C.N. (1992): Knowledge Acquisition and Chinese Parsing Based on Corpus. In: *Proceedings of COLING '92*, Nantes, France, 23-28 August 1992, 13000-13004.

ZHOU, M. and HUANG, C.N. (1994): An Efficient Syntactic Tagging Tool for Corpora. In: *Proceedings of COLING '94*, Kyoto, 5-9 August 1994, 949-955.