UNIVERSAL GUIDES AND FINITENESS AND SYMMETRY OF GRAMMAR PROCESSING ALGORITHMS

Miroslav Martinović

Courant Institute of Mathematical Sciences, New York University
martin@cs.nyu.edu

ABSTRACT

This paper presents a novel technique called "universal guides" which explores inherent properties of logic grammars (changing variable binding status) in order to characterize formal criteria for termination in a derivation process. The notion of universal guides also offers a new framework in which both parsing and generation can be viewed merely as two different instances of the same generic process: guide consumption. This technique generalizes and exemplifies a new and original use of an existing concept of "proper guides" recently proposed in literature for controlling top-down left-to-right (TDLR) execution in logic programs. We show that universal guides are independent of a particular grammar evaluation strategy. Also, unlike proper guides they can be specified in the same manner for any given algorithm without knowing in advance whether the algorithm is a parsing or a generation algorithm. Their introduction into a grammar prevents as well the occurrence of certain grammar rules an infinite number of times during a derivation process.

1. INTRODUCTION AND MOTIVATION

This research interacted with a Japanese-English machine translation project at New York University. The results reported herein are part of an attempt to establish an evaluation system for grammar processing algorithms (parsing and generation algorithms). The need for evaluation of various competing approaches presently available for parser and generator design has been felt strongly in both Theoretical and Computational Linguistics. Both fields have been thus far predominantly empirical, so that the measuring of actual progress has become very difficult. Here, we introduce the notion of universal guides in order to discuss two of the most relevant criteria for the comparison of different parsing and generation algorithms: finiteness and symmetry. Other criteria such as completeness, soundness, efficiency, etc., although equally significant and interesting, are outside of the scope of this paper and are addressed in [M92].

There is a natural appeal to the problem of characterizing parsing and generation within the same framework and in a symmetrical way. The reversibility is by its nature symmetrical: parsing is retrieving a semantic content from a phonological one, and generation, a phonological from a semantic content. Several papers ([S88], [N89], [SNMP89], [DI88], [DI90], and [DIP90]) have recognized parsing and generation as instances of a single paradigm and have pointed out the correspondence between certain parsing problems and techniques (left-recursion, linking, Early deduction) and their correlates in generation.

It has also been long noticed that adopting a certain technique for a derivation process can lead to termination problems (sometimes referred to as infinite derivations). Perhaps the best known example of this is using the TDLR derivation for left-recursive rules. Consequently, to specify conditions on grammars, whose fulfillment is necessary and sufficient to guarantee finite derivations under a given evaluation strategy, posed another problem, one that has been given serious attention recently ([D90], [DIP90]). These conditions are usually referred to as the finiteness criteria and are often given in the form of sufficient though not necessary conditions ("worst case" analysis of the finiteness of an algorithm).

What we propose here is to abstract the notion of string index in parsing to the notion of a *universal guide*. A similar proposal was made in [DIP90] for *guides* (here called *proper guides* to distinguish them from *universal guides*). Using the new concept, both parsing and generation can be seen as two instances of the same generic process: universal guide consumption. Universal guides prove to be more general than proper guides because they can be used under any evaluation strategy and not only under TDLR technique as must proper guides. They achieve symmetry in treating parsing and generation but need not be instantiated differently in either case, unlike proper guides. Universal guides can be consumed anywhere during a derivation, as opposed to proper guides which may be consumed only by the application of lexical rules, therefore restricting the class of grammars for which they can be used. Moreover, we show here that proper guides can be viewed as a special case of our universal guides concept. The introduction of universal guides into a grammar also prevents the infinite repetition of certain grammar rules (i.e. those that do not instantiate any grammar variable and cause infinite loops).

2. PRIOR RELATED WORK

Some of the most significant findings with respect to characterizing finiteness and symmetry of grammar processing algorithms have been published by Dymetman, Isabelle and Perrault in [DIP90]. The authors pointed out the symmetrical nature of parsing and generation by introducing the notion of (proper) guides. A guide structure is a partially ordered set G that respects the descending chain condition, i.e. the condition that in G all strictly decreasing chains ($l_1 > l_2 > ... > l_i > ...$) are finite.

Guides were introduced into each of a logic grammar's non-terminals in the form of new, so called *guide* variables. These variables added some redundancy which could be exploited for tighter control of the computational process. After the guide variables were added and left recursion was eliminated (by performing the usual transformation as indicated in [DIP90]), the creation of a new grammar, equivalent to the original one, was completed. Then, a set of conditions was specified that, if the new grammar satisfied it, guaranteed a finite derivation for any given goal. The conditions are: the guide consumption condition (GCC) and the no-chain condition (NCC). The guide consumption condition states that the values for guide variables must initially be finite and must also be consumed (decreased) each time a **lexical** predicate is expanded. The no-chain condition prohibits the exclusive appearance of predicates like $T \to U$ on the right-hand side of a rule. It was shown that if both GCC and NCC held, all derivations in the grammar would be finite. The notion of guides is applicable to both parsing and generation but it is instantiated differently in each case (for parsing, the guide variable represents the list of words awaiting to be analyzed, and for generation, the list of semantics of subcategorized constituents remaining to be generated). The authors of [DIP90] also demonstrated an application of their main result to the class of lexical grammars.

The following improvements look desirable with respect to the main result from [DIP90]:

(i) The guides should be specified before the details of the algorithm (parsing or generation) and the underlying grammar (lexical or other kind) are available. They should not be dependent on these details;

(ii) The main result (concerning finite derivations) should be stated with respect to any grammar evaluation strategy and not only with respect to the top-down, left-to-right algorithm;

(iii) The consumption of guides should be allowed at any level, not only lexical; and

(iv) The very introduction of guides (with no additional grammar transformations) should prevent a certain kind of infinite derivations from happening (i.e. those due to left recursive rules).

We show here that the guides' approach by Dymetman et al. can be viewed as a special case of the universal guides approach that we introduce in this paper. We also demonstrate that universal guides realize the desired improvements.

3. UNIVERSAL GUIDES

We motivate our introduction of universal guides around the idea that a derivation can be perceived as a process of discovering the set of all variables that participate in it. In other words, a derivation is finding all logic grammar symbols that are uninstantiated at the moment when they appear in the derivation for the first time, and keeping track of changing of their binding status. The logic grammar symbols enter a derivation by applying a production rule in which they participate either as bound, or partially or totally unbound. The non-bound ones may or may not get instantiated during the derivation and their number can be finite (in the case of a finite derivation) or infinite (as with an infinite derivation). The term complete derivation we use for the derivation in which the set of uninstantiated variables eventually gets reduced to an empty set.

The set (finite or infinite) of all these variables has properties similar to the guides as defined by Dymetman, et al. in [DIP90] (only the descending chain condition is not guaranteed) and will as such be a major component of our notion of universal guides. The comparison relation for this partially ordered structure consisting of sets will be based on the relation "being a subset" $\subset$. We formalize the previous discussion by the following definitions.

DEFINITION 3.1. (A USEFUL PARTIALLY ORDERED RELATION)
Let S and S' be two sets and N and N' two non-negative integers. We say that ordered pair (S,N) is *greater than or equal to* ordered pair (S',N') and write $(S,N) \geq (S',N')$ iff ($S=S'$ & $N=N'$) or ($S \supset S'$ or ($S=S'$ & $N>N'$)$_\blacksquare$

$\geq$ is obviously a reflexive, anti-symmetrical and transitive relation and therefore a relation of partial order.

DEFINITION 3.2. (UNIVERSAL GUIDES)
Let Ξ be a collection of all subsets of a set ξ ($\Xi = \mathcal{P}(\xi)$) and $I^{\geq 0}$ set of all non-negative integers. A universal guide structure is a partially ordered structure $(P, \geq)$, where $P = \{ (S,N) \,/\, S \in \Xi \text{ \& } N \in I^{\geq 0} \}$, and $\geq$ is from the definition 3.1.$_\blacksquare$

The presence of a special kind of universal guides in a grammar is always only implicit, but for the sake of being able to prove facts about them formally, it can be made explicit. The universal guide structure in a logic grammar is based on the set of variables still uninstantiated at a given moment of a derivation process. Expansion of a production rule may or may not instantiate (consume) some of them. For instance, by adding two special extra arguments to each symbol in the following rules:

(1) noun_phrase (Num, NP_Str, NP_Rest) --> det(Num,NP_Str,D_Rest),noun(Num,D_Rest,NP_Rest).
(2) det (sing, [a|D_Rest], D_Rest).
(3) noun (sing, [dog|N_Rest], N_Rest).

the convergence or divergence of a derivation using the rules becomes explicit. The new arguments are: a set of currently uninstantiated variables (the so called guide's set component) and a non-negative integer (the guide's numeric component). The following simple derivation of the phrase *a dog*:

noun_phrase (Num, NP_Str, NP_Rest) -->
  det (Num, NP_Str, D_Rest),
  noun (Num, D_Rest, NP_Rest) --> *(1)*
  det (sing, [a|D_Rest], D_Rest),
  noun (sing, D_Rest, NP_Rest) --> *(2)*
  det (sing, [a,dog|NP_Rest], [dog|NP_Rest]),
  noun (sing, [dog|NP_Rest], NP_Rest). *(3)*

(assuming there are ρ rules in the grammar) becomes:

noun_phrase (Num, NP_Str, NP_Rest,
    {Num, NP_Str, NP_Rest, D_Rest}, *ρ*) -->
  det (Num, NP_Str, D_Rest,
    {Num, NP_Str, NP_Rest, D_Rest}, *ρ*-1),
  noun (Num, D_Rest, NP_Rest,
    {Num, NP_Str, NP_Rest, D_Rest}, *ρ*-1) --> *(1)*
  det (sing, [a|D_Rest], D_Rest,
    {NP_Rest, D_Rest}, *ρ*),
  noun (sing, D_Rest, NP_Rest,
    {NP_Rest, D_Rest}, *ρ*) --> *(2)*
  det (sing, [a,dog|NP_Rest], [dog|NP_Rest],
    {NP_Rest}, *ρ*),
  noun (sing, [dog|NP_Rest], NP_Rest,
    {NP_Rest}, *ρ*). *(3).*

The new arguments are given in bold case. When a rule to be expanded (partially or completely) instantiates some variables (rules (2) and (3) in the previous example), the guide's set component is reduced by those variables and the guide's numeric component is reset to ρ (the number of rules in the grammar). On the other hand, when a rule does not instantiate any variable, the set component stays unchanged and the numeric component is decreased by one (rule (1)).

The numeric component actually counts (down) the number of consecutive occurrences of rules that do not instantiate any variable. If that number is larger than the total number of rules in the grammar (ρ), then a grammar rule must have been repeated in its unchanged form, and a potentially infinite derivation is caught. Such a rule is always failed. For example, *np --> np pp* rule will in a TDLR derivation cause an infinite loop. However, adding universal guides to it will always, when this rule is applied, decrease the guide's numeric component by one (as no variable gets instantiated). Since the numeric component is always initialized and reset (after a rule that instantiates some variable(s)) to ρ (number of rules in the grammar), it will eventually go down to 0, which in turn will fail this rule, as in the following sequence:

np(Vars,ρ) --> np(Vars,ρ-1) pp(Vars,ρ-1)
--> np(Vars,ρ-2) pp(Vars,ρ-2)pp(Vars,ρ-1)
--> np(Vars,ρ-3) pp(Vars,ρ-3)pp(Vars,ρ-2)pp(Vars, ρ-1)
  ....
--> np(Vars,0) pp(Vars,0)pp(Vars,1) ... pp(Vars,ρ-1)
--> fail.

All details (additional arguments for original predicates, additional predicates, transformation of the original rules into equivalent ones containing guides) of the procedure for introducing universal guides into a grammar and their handling can be found in [M92].

Thus, the universal guide structure is represented in the new grammar by the pairs (UnIn,Num) which stand for the set of all currently uninstantiated variables and the numeric guide's component, respectively. The new grammar is equivalent to the original one. By the introduction of an additional grammar predicate at the end of each rule (called *decrease* in [M92]) the guide consumption condition is demonstrated to hold for any finite and complete derivation. If a variable gets bound then the guide gets strictly smaller because the set of still uninstantiated variables participating in the derivation has lost one member. If no variable gets instantiated, the decrease of the number component is there to ensure that the guide itself strictly decreases (by the definition 3.2.). By failing, this new (*decrease*) predicate will stop any derivation that contains a sequence of more than ρ consecutive applications of rules that do not instantiate any variable (because the further decrease of the guide's numeric component would make it negative). Otherwise, a production rule would be repeated in the same manner without instantiating any of the present variables which would in turn cause an infinite derivation to take place. Thus, because of the way the universal guides are introduced for any derivation in the new grammar, guide consumption condition holds. The difference between finite and infinite derivations is isolated and solely characterized by the set component of the universal guides being either finite or infinite initially.

The following theorem establishes a correlation between *proper* and *universal guides*.

Theorem 3.1.: If there is a proper guide structure for a class of logic grammars (in its form from [DIP90]) satisfying guide consumption and no-chain conditions (GCC and NCC) under the TDLR grammar evaluation algorithm, then the universal guide structure (under the same algorithm) is a proper guide structure satisfying both GCC and NCC.■
Proof.

The existence of the proper guides satisfying GCC and NCC under the TDLR algorithm guarantees that all derivations will be finite for the given class of grammars (main theorem from [DIP90]).

Since the derivations are finite, the universal guides will initially assume a finite value (its set component will be set to all variables taking part in the derivation and its number component will be assigned value ρ (number of different production rules in the grammar). The universal guides are defined to always satisfy the guide consumption condition, and since NCC is assumed as well, it only remains to show that the descending chain condition is respected.

As every strictly descending chain of universal guides with a finite initial value must be finite (moreover, we know that its length is always less than or equal to $\rho * u$ (u being number of variables taking part in the derivation), the universal guide structure has all properties of a proper guide structure■

Thus, whenever proper guides can be used to establish the finiteness of an algorithm, the universal guides approach may likewise be used.

Also, the notion of universal guides proves to be more general than the notion of guides in the sense that it does not assume any particular algorithm under which a grammar will be processed. It is applicable to any algorithm, and to apply it would mean to specify conditions on grammars that would (for a given algorithm) guarantee finiteness of the forementioned sets (set components of the universal guides). Of course, the character of the conditions will depend on the nature of the grammar processing algorithm. Proper guides as proposed by Dymetman, et al. guarantee finiteness of one specific (TDLR) algorithm if the grammar satisfies GCC and NCC.

The following example describing a *wh-question* (here used for the generation of the sentence *who wrote this* from the given semantics *wrote(who,this)*) could be helpful to illustrate the applicability of the universal guides approach where the proper guides would not work:

(1) whques
(WhQues_Sem, WhQues_Sen, WhQues_SenRest) -->
    whsubj
(Number, WhSubj_Sem, WhQues_Sen, Rest1),
    whpred
(Number, Tense, [WhSubj_Sem, WhObj_Sem], WhQues_Sem, Rest1, Rest2),
    whobj (WhObj_Sem, Rest2, WhQues_SenRest).
(2) whsubj (X, who, [who|WhSubjRest], WhSubjRest).
(3) whsubj (X, what, [what|WhSubjRest], WhSubjRest).
(4) whpred (sing, perf, [Subj,Obj], wrote(who,this),
    [wrote|WhPredRest], WhPredRest).
(5) whobj (this, [this|WhObjRest], WhObjRest).

Fig. 3.1.: A Sample Grammar.

When the semantic-head-driven generation algorithm (see [SNMP89]) is used, the order in which different rules are applied can best be described by the analysis tree from the Fig. 3.2.. The numbering for edges indicates the order in which the grammar rules were used. Thus, rule (4) was used first, then rule (1), followed by rules (2) and (5) respectively. The variables that were introduced as uninstantiated are *WhQues_Sen* (by invoking the topmost predicate), *Subj, Obj,* and *WhPredRest* (by the application of rule (4)). The application of rule (1) (expanded second) unified *number* components of *whsubj* and *whpred* (already instantiated by rule (4)), *semantic* components of *whsubj* and *whobj* with variables *Subj* and *Obj* from *whpred,*

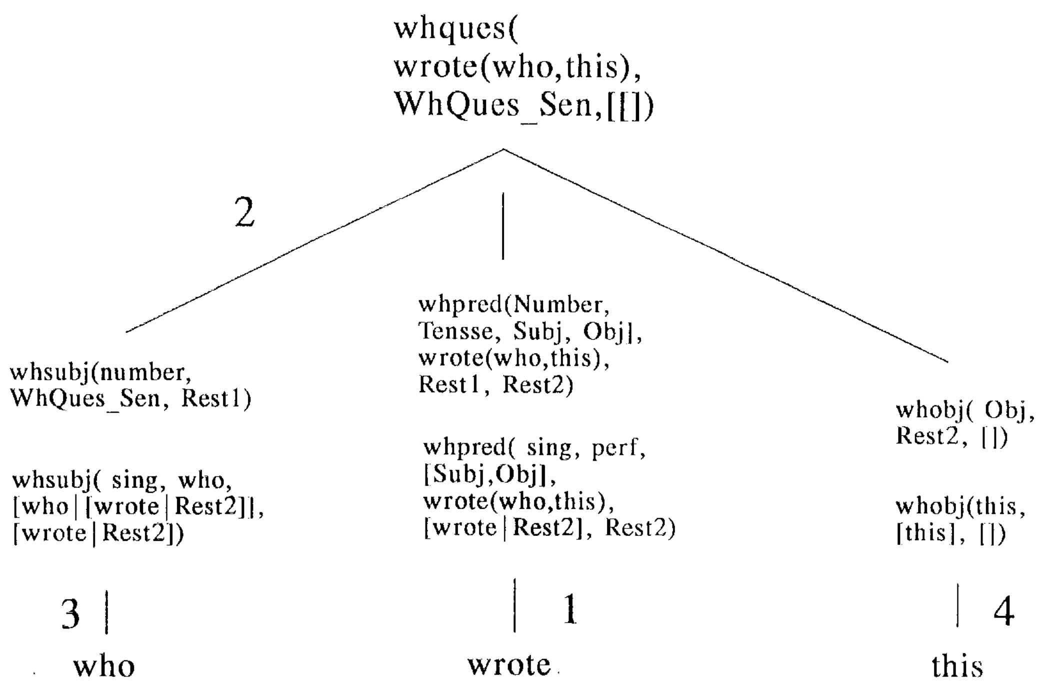

Fig. 3.2.: A Derivation Tree.

respectively, *WhSubjRest* component with *WhPred_Sen* component of *whpred* (already partially instantiated by rule (4)), as well as *WhPredRest* with *WhObj_Sen*. It also unified *WhSubj_Sen* with *WhQues_Sen* and *[]* with *WhObjRest* and therefore did not introduce any new uninstantiated variables. The application of rules (2) and (4) did not introduce new variables neither. Thus, the set of all uninstantiated variables participating in this derivation is { WhQues_Sen, Subj, Obj, WhPredRest }. *Subj* gets instantiated by the application of rule (2), *Obj* and *WhPredRest* by rule (5), and the instantiation of the variable *WhQues_Sen* is partially done by rules (4), (2), and eventually completed by rule (5). Thus, set component of the universal guide variable { WhQues_Sen, Subj, Obj, WhPredRest } is first consumed by the application of rule (4) (part of *WhQues_Sen*), then by rule (2) (another part of *WhQues_Sen*, plus *Subj*), and eventually reduced to an empty set (represented here as []) by rule (5) (final ingredient of *WhQues_Sen*, plus *Obj* and *WhPredRest*).

This semantic-head-driven derivation with the universal guides included (and under assumption that the grammar has ρ different production rules) can be described by the following steps:

-- whques (wrote(who, this), WhQues_Sen, [], **{WhQues_Sen, Subj, Obj, WhPredRest}, *ρ***). -?-
-- whpred (sing, perf, [Subj,Obj], wrote(Subj,Obj), [wrote|WhPredRest], WhPredRest, **{WhQues_Sen, Subj, Obj, WhPredRest}, *ρ***). rule (4)
-- whques (wrote(who,this), WhQues_Sen, [], **{WhQues_Sen, Subj, Obj, WhPredRest}, *ρ*-1**) -->
whsubj (sing, Subj, WhQues_Sen, [wrote|WhPredRest], **{WhQues_Sen, Subj, Obj, WhPredRest}, *ρ*-1**),
whpred (sing, perf, [Subj,Obj], wrote(Subj,Obj), [wrote|WhPredRest], WhPredRest, **{WhQues_Sen,Subj,Obj,WhPredRest}, *ρ*-1**),
whobj (Obj, WhPredRest, [], **{WhQues_Sen,Subj,Obj,WhPredRest}, *ρ*-1**) rule (1)
-- whsubj (sing, who, [who,wrote|WhPredRest], [wrote|WhPredRest],**{WhQues_Sen,Obj,WhPredRest}, *ρ***) rule (2)
-- whobj (this, [this], [], {}, ***ρ***) rule (5)

Unlike universal guides proper guides require a specific (TDLR) grammar evaluation strategy and therefore this approach is not applicable at all for the semantic-head-driven generation algorithm since this algorithm assumes a grammar evaluation strategy different from TDLR.

Generally and for any evaluation strategy, the formal link between the universal guide consumption and termination can be expressed by the following claim:

Theorem 3.2.: Let G be a logic grammar and G' its equivalent after the universal guides were introduced into G. If the guide consumption condition is fulfilled for a derivation in G' and initial value of the guide structure is finite, then the derivation in question will be finite too$_\blacksquare$

The proof of this theorem as well as a detailed specification of how to introduce universal guides into a grammar can be found in [M92].

Moreover, the common essence of the parsing and generation process as merely different instances of the same generic process of consuming the universal guides becomes obvious after making the appearance of universal guides explicit. Universal guide variables do not necessarily have different meaning for parsing and generation as do proper guides. Even under an evaluation strategy (TDLR) assumed in advance proper guides (as in the case of lexical grammars from [DIP90]) represent different entities for parsing and generation. For a parsing algorithm guides are difference lists of words remaining to be analyzed, and for a generation they are lists of subcategorized semantics to be generated next. Unlike proper guides, universal guides exposed the common substance of the two processes. They are always (for parsing as well as for generation) instantiated as sets of all currently uninstantiated variables.

Another feature of the universal guides that gives them an advantage over proper guides is that they do not impose the restriction that the guides can be consumed only at the level of lexical predicates. Thus, the class of grammars for which this approach can be used is broader than for that of proper guides.

Also, we in effect presented here a class of grammars the recursivity of which can be proven by induction.

4. CONCLUSION

This paper addressed finiteness and symmetry of parsing and generation algorithms using a novel *universal guides approach.* We pointed out some deficiencies of *proper guides approach* as advocated in some earlier research. These included the applicability of proper guides only when the evaluation strategy is TDLR, and when it is also known whether a parsing or a generation algorithm is in question. Also, the consumption of proper guides was allowed only at the lexical level. By the introduction of *universal guides* all of these deficiencies are eliminated and a true symmetry is achieved in treating the parsing and generation problem. Unlike *proper guides*, *universal guides* do not need to be constructed and instantiated differently for parsing and for generation, and no additional grammar transformation (i.e. left recursion elimination) is needed for them to be applicable.

5. ACKNOWLEDGMENTS

This paper is based upon work supported by the Defense Advance Research Project Agency under Contract N00014-90-J-1851 from the Office of Naval Research, the National Science Foundation under Grant IRI-93-02615 and the Canadian Institute for Robotics and Intelligent Systems (IRIS). Additional support was

provided by Wagner College, Staten Island and Pace University, New York.

6. REFERENCES

[DI88] DYMETMAN, M. and ISABELLE, P.
1988. "Reversible Logic Grammars for Machine Translation".
Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages,
Carnegie-Mellon University, Pittsburgh, PA.

[DI90] DYMETMAN, M. and ISABELLE, P.
1990. "Grammar Bidirectionality through Controlled Backward Deduction".
Logic and Logic Grammars for Language Processing.
eds. Saint Dizier, P. and S. Szpakowicz, Chichester, England: Ellis Horwood.

[DIP90] DYMETMAN, M., ISABELLE, P. and PERRAULT, F.
1990. "A Symmetrical Approach to Parsing and Generation".
Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)
Helsinki, Finland, Vol. 3., pp. 90-96.

[DM78] DERSHOWITZ, N., MANNA, Z.
1978. "Proving Termination with Multiset Orderings".
Technical Report, Stanford Uni., March 1978.

[D87] DERSHOWITZ, N.
1987. "Termination of Rewriting".
Journal of Symbolic Computation
Vol. 3, 1987.

[M92] MARTINOVIĆ, M.
1992. *An Evaluation System for Parsing and Generation Algorithm.*
October 1992, Belgrade University Thesis Research Report.

[MS92] MARTINOVIĆ, M. and STRZALKOWSKI, T.
1992. "Comparing Two Grammar-Based Generation Algorithms: A Case Study".
Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics,
July 1992, Newark, Delaware.

[N89] VAN NOORD, J.
1989. "BUG: A Directed Bottom-Up Generator for Unification Based Formalisms".
Working Papers in Natural Language Processing No. 4.
Utrecht, Holland: RUU, Department of Linguistics.

[S88] SHIEBER, S. M.
1988. "A Uniform Architecture for Parsing and Generation".
Proceedings of the 12th International Conference on Computational Linguistics.
pp. 614-619., Budapest, Hungary, August 1988.

[SNMP89] SHIEBER, S.M., VAN NOORD, G., MOORE, R., PEREIRA, F.
1989. "A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms".
Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics,
pp. 7-17, Vancouver, B.C., Canada, June 1989.