Spang-Hanssen, Henning. 1983. 'La notion de verbe auxiliaire'. In Michael Herslund, Ole Mørdrup & Finn Sørensen (eds.), *Analyses grammaticales du français*, 5-16. København: Akademisk Forlag.
Stojanov, Stojan (gl. red.). 1983. *Gramatika na săvremennija bălgarski knižoven ezik. Tom II. Morfologija.* Sofija: Izdatelstvo na Bălgarskata Akademija na Naukite.
Tedeschi, Philip J. & Annie Zaenen (eds.). 1981. *Syntax and semantics 14: Tense and aspect.* New York: Academic Press.
Thieroff, Rolf. 1992. *Das finite Verb im Deutschen. Tempus – Modus – Distanz* (=Studien zur deutschen Grammatik 40). Tübingen: Gunter Narr.

Lund University, Dept. of Linguistics
Working Papers 42 (1994), 197-208

Appending X-bar Grammar (AXG) for Syllables

Bengt Sigurd

Introduction and background

Syllable structure, consonant clusters and word phonotactics have been studied in many languages, particularly during the heyday of structural linguistics. The phonotactic structure of Swedish monosyllables and words is analyzed in great detail in Sigurd 1965, a book which constitutes the background of this paper. A great number of regularities and constraints are noted and formulated there and many of these have been utilized in the rules presented in this paper. One of the main observations is that consonants occur in a certain order in the clusters and can be said to show a varying tendency to occur close to the vowel (sonority, Jespersen 1897; vowel adherence, Sigurd 1955). Thus, when *r* and *l* are combined (only finally in *-rl)*, *r* has the greatest vowel adherence as it must occur closest to the vowel. When *l* and *k* occur, *l* must occur close to the vowel, which is witnessed finally in *-lk* and initially in *kl-*.

The order constraints may be expressed by a partial rank order between the consonants involved, e.g.

r < l < j, v < m < n < b, p, g, k, f < d, t < s

There is an ongoing discussion about the universality of such an ordering and other combinatory phonotactic restrictions (see Basbøll 1977, Clements 1990). It is possible to formulate very general rules based on features, e.g. the rule that a voiceless consonant must always occur outside a voiced consonant, a stop must always occur outside liquids and nasals, if two consonants of the same type combine finally (in Swedish) the dental must always occur last *(-rl, -mn, -gd, -pt, -kt. -fs).* Some of these generalizations will be utilized in the rules presented below. The other main observation to be made is that certain consonants combine, others do not, although their order follows the sonority scale. Thus *t* and *d* have clearly less sonority than *l*, but there is no initial *tl-* or *dl-*. A number of restrictions other than order are clearly at work. It is natural to try to express the combinatory and order restrictions in features and this will also be done in this paper.

Appending X-bar Grammar (AXG; Sigurd 1994, Sigurd & Lastow 1993) is a system characterized by rules which append constituents to head constituents resulting in new constituents with higher bar values according to X-bar theory (see Jackendoff 1977 and various modern grammatical theories). In syntax, one may for instance append an adjective (e.g. *nice*) to a noun head (e.g. *dog*) with bar value 1 producing a noun (phrase) with bar value 2 *(nice dog)*. To this noun one may in turn append an article resulting in a noun with bar value 3 *(a nice dog)*. Verbs are expanded in the same way and eventually a subject noun (phrase) is appended to a verb (phrase) resulting in a sentence, i.e. a verb (phrase) with a certain high bar value. AXG is a computer oriented grammar written directly in Prolog either using the built-in predicate *append* or Definite Clause Grammar (DCG).

This paper applies the AXG approach to consonant clusters and syllables. Simple consonants, such as *l*, may thus be seen as consonants with bar value 1. The addition of another consonant before *l*, such as *p*, results in a complex consonant *(pl)* with bar value 2 and the addition of a further consonant, e.g. *s* results in a consonant with bar value 3 *(spl)*. This cluster may then be appended to a following vowel with following consonants to make a syllable.

Overview, alternative descriptions

It is natural to treat the vowel as the head of the syllable, just as the verb is treated as the head of the sentence. There are then several possibilities. One (Approach I) may be illustrated by the following rules, where the initial (inc3) and final (finc3) clusters are treated as separate constituents (generated by AXG rules) and added as equals to the vowel (v). The initial clusters are described as a successive addition of consonants. The final clusters are to be described in the same way.

Approach I

```
1. v2 --> inc3, v1, finc3     % spj+u+t
2. v1 --> v                   % u
3. inc3 --> c3, inc2          % s+pj
4. inc2 --> c2, inc1          % p+j
5. inc1 --> c                 % j
...
```


A second alternative (Approach II) assumes that the final clusters are more closely related to the vowel than the initial ones, which is reasonable in Swedish, the language under analysis. There are some restrictions on the combination of vowels and following consonants. Only simple consonants can generally occur after long vowels, *j* cannot occur after *i*, the vowel must be short before *j* and *ng*. There are no such restrictions on the combination of the initial consonants and the following vowel. Approach II agrees with the structure often assumed for the syllable: Syllable --> Onset+Rhyme, Rhyme --> Nucleus+Coda. After studies of coarticulation some phoneticians believe, however, that the connection between the initial consonants and the vowel is the closest (Wood 1994). Approach II may be demonstrated by the following rules.

Approach II

```
1. v3 --> inc3,v2        % spj+ut
2. v2 --> v1,finc3       % u+t
3. v1 --> v              % u
4. inc3 --> c3, inc2     % s+pj
5. finc3 --> ...
...
```

This solution is reminiscent of the standard Chomskyan view of syntactic structure which can be traced back to his view of subject and object reflected in the well-known rules in Syntactic Structures: S --> Np + Vp, Vp --> V, Np. The Np of the first rule has later been called *specifier* and the Np of the second rule *complement* and one may use these terms for the constituents of the syllables in our rules as well.

In Approach II the final consonants are generated as a separate constituent. A third solution (III) takes each consonant added finally as a separate layer added to the vowel head. This approach is illustrated by the following rules.

Approach III

```
v5 --> inc3, v4      % t+orsk
v4 --> v3, c3        % ors+k
v3 --> v2, c2        % or+s
v2 --> v1, c1        % o+r
v1 --> v             % o
```

We will present detailed rules for Swedish syllables illustrating approach I and III below. We will also describe two different ways of generating final clusters, one by stating which consonants can combine, the other by stating which consonants cannot combine. But we will first present the feature analysis of the Swedish consonants and vowels which we will use.

Distinctive features of Swedish phonemes

The feature set-up of the consonant phonemes is registered in the frame *p(cons, Type, Place, Voice, Bar)*, where the constant *p* denotes phoneme and *Type, Place, Voice, Bar* are variables (Variables are always spelled with initial capital letters in Prolog). *Type* varies over: *liq(uid), nas(al), fric(ative), klus(il)* ('stop'). *Place* varies over *lab(ial), dent(al), pal(atal), vel(ar), phar(yngeal). Voice* varies over *tonl* (voiceless), *ton* (voiced); some Swedish words have been used in the names of constants or variables in the program. The first rule below states that the phoneme *s* has the features *cons(onant), fric(ative), dent(al),* or equivalently that those features are manifested as *s*. The consonants *ç, sh* (due to restrictions in the Prolog program we do not use (IPA) symbols) and *h* occur only initially and cannot enter into combinations, which is why they have been given the bar value 3 (the phoneme *sh* may occasionally be combined finally as in *lunch, match*, but we disregard these foreign words in the general rules below). The feature set-ups of *sh* and *h* are not clear; we have given *sh* the place feature *vel(ar)* and *h* the feature *phar(yngeal)* to make them distinctive. We are not going to discuss the merits of our analysis. It is sufficient for our present purpose. The phoneme *ng* has not been included and length has been disregarded.

Consonants

```
p(cons,fric,dent,tonl,1) --> [s].
p(cons,fric,lab,tonl,1) --> [f].
p(cons,fric,lab,ton,1) --> [v].
p(cons,fric,pal,ton,1) --> [j].
p(cons,fric,pal,tonl,3) --> [ç]. % only initially
p(cons,fric,vel,tonl,3) --> [sh]. % initially
p(cons,fric,phar,tonl,3) --> [h]. % only initially
p(cons,klus,lab,tonl,1) --> [p].
p(cons,klus,pal,tonl,1) --> [k].
p(cons,klus,dent,tonl,1) --> [t].
p(cons,klus,lab,ton,1) --> [b].
p(cons,klus,pal,ton,1) --> [g].
p(cons,klus,dent,ton,1) --> [d].
p(cons,liq,dent,ton,1) --> [l].
p(cons,liq,pal,ton,1) --> [r].
```



```
p(cons,nas,lab,ton,1) --> [m].
p(cons,nas,dent,ton,1) --> [n].
```

The Swedish vowels are analyzed as follows, using the frame *p(Height, Front, Round, Bar). Height* varies over: *low, middle, lowmid, high. Front(ness)* varies over: *front, back. Round(edness)* varies over *round(ed), unround(ed).* The set of features needed for the Swedish vowels has been debated for many years and our analysis is not meant to contribute to that discussion. It is satisfactory for the purpose of this paper and we will not discuss the problems involved.

Vowels

```
p(voc,low,back,unround,1) --> [a].
p(voc,middle,front,unround,1) --> [e].
p(voc,high,front,unround,1) --> [i].
p(voc,middle,front,round,1) --> [y].
p(voc,middle,back,round,1) --> [o].
p(voc,high,front,unround,1) --> [u].
p(voc,lowmid,front,unround,1) --> [ä].
p(voc,middle,front,round,1) --> [ö].
```

Approach I

According to approach I we treat vowels and initial and final clusters separately. The following AXG rules generate initial clusters *(inc)* according to this approach.

Initial clusters

The list of initial clusters in Swedish can be seen in the demo below.

```
inc(cons,Typ,Place,Ton,3) --> p(cons,Typ,Place,Ton,3). % h, ç, sh
```

The above rule states that a phoneme with bar level 3 can make up an initial consonant. Such phonemes are: *h, ç* and *sh* as defined above. By assigning them the bar value 3, they cannot be combined (pronouns are assigned the bar level 3 for the same reason in Sigurd 1994). The following rules state that an *inc* level 3 may be an *inc* level 2, and that an *inc* level 2 may be an *inc* level 1. Note that the features *(Type, Place, Ton)* of the consonant are carried over (percolated) to the higher constituent.

```
inc(cons,Type,Place,Ton,3) --> inc(cons,Type,Place,Ton,2). % all 2 are 3
inc(cons,Type,Place,Ton,2) --> inc(cons,Type,Place,Ton,1). % all 1 are 2
```

The following rules state that any consonant bar 1 may be an initial constituent.

```
inc(cons,Type,Place,Ton,1) --> p(cons,Type,Place,Ton,1).
```

The features of the added consonant are percolated to inc2 to be used in order to restrict further combinations. A number of conditions may be stated using the features of the added consonant (Type1, Place1, Ton1) and the following consonant (Type2, Place2, Ton2). One condition is that the voice of the consonant closest to the vowel (Ton2) should be more vowel adherent than the voice of the preceeding consonant (Ton1). This may be expressed by *vadher(Ton2,Ton1)*, which is defined as follows using a semicolon ';' to separate alternatives within a parenthesis as is done in Prolog.

```
vadher(X,Y) :- (X=ton, Y=tonl); % X voiced, Y voiceless
               X=ton, Y= ton; % both voiced
               X=tonl,Y=tonl). % both voiceless
```

The rule states that a consonant X is considered more vowel adherent than another (Y) if it is voiced while the other is voiceless, if both are voiced or if both are voiceless. The rule prohibits voiced consonants from occurring outside voiceless consonants, which is a general rule in Swedish. The formulation of the rule allows a voiced consonant to occur closer to the vowel than another voiced one, a voiceless consonant to occur closer than a voiceless one and a voiced consonant closer than a voiceless one. There are seemingly exceptions in words such as *spasm, rhythm*, but the *m* is is fact pronounced voicelessly. The following is the rule covering all initial 2-member clusters such as indicated by the comments.

```
inc(cons,Type1,Place1,Ton1,2) --> p(cons,Type1,Place1,Ton1,1), % first
    p(cons,Type2,Place2,Ton2,1), % second consonant
    {vadher(Ton2,Ton1)}, % vowel adherence condition
    {(Type1=nas,(Place1=dent;Place1=lab),Ton1=ton,Type2=fric,
    Place2=pal, Ton2=ton; %mn+j
    Type1=fric,Place1=lab,Ton1=ton,
    Type2=liq,Place2=pal,Ton2=ton;    %vr
    Type1=klus,Place1=dent,(Ton1=tonl;Ton1=ton),
    (Type2=fric,Place2=lab,Ton2=ton; % tv,dv
    Type2=liq,Place2=pal,Ton2=ton); % tr,dr
    Type1=klus,Place1=pal,(Ton1=ton;Ton1=tonl), % k,g
    (Type2=liq,(Place2=pal;Place2=dent),Ton2=ton; % kr,gr,kl,kr
    Type2=nas,Place2=dent,Ton2=ton); % kn,gn
    Type1=klus,Place1=pal,Ton1=tonl,
    Type2=fric,Place2=lab,Ton2=ton); % kv
    Type1=klus,Place1=lab,(Ton1=tonl;Ton1=ton),
    (Type2=liq,(Place2=pal;Place2=dent),Ton2=ton; %pr,pl,br,bl
    Type2=fric,Place2=pal,Ton2=ton); % pj,bj
    Type1=fric,Place1=lab,Ton1=tonl,
    (Type2=liq,(Place2=pal;Place2=dent),Ton2=ton; %fr,fl
    Type2=fric,Place2=pal,Ton2=ton; % fj
    Type2=nas,Place2=dent,Ton2=ton))}. % fn
```


The conditions (within { }) may be rendered in various ways and there may be more economic (shorter) ways, but the initial clusters include various irregularities which have to be treated individually. The voiced and the voiceless member of a pair do not always behave in the same way. There is *kv-*, but not *gv-*. Similarly, *v* and *f* do not behave in the same way before *r* and *l*; *fr-* and *fl-* occur but not *vl-*, only *vr-*. There is *fn-* but not *vn-*. It is complicated to include *fj-* in a rule with *bj-* and *pj-*, as *vj-* does not occur.

The following rule states that an *s* may precede a 2-consonant cluster beginning by a voiceless stop *(p, t, k)*. This rule will in fact overgenerate and has to be restricted. It generates *stv* and *skn* as *tv* and *kn* exist. There is no way to restrict it using the features available to the rule, but one may filter out the two clusters directly from the output (see below).

```
inc(cons,klus,Place,tonl,3) --> [s],inc(cons,klus,Place,tonl,2).
```

Final clusters

The following rule states that all consonants with bar value 1 (thus not *h, ç* and *sh*) are accepted as *finc* with bar value 1.

```
finc(cons,Type,Place,Ton,1) --> p(cons,Type,Place,Ton,1). % all c 1 are inc
```

The following rule is similar to the initial rule generating 2-member clusters. It states that one may add another consonant after a *finc* with bar value 1 provided the conditions on the features are met. There are a great number of conditions or restrictions, some of which seem regular, others idiosyncratic.

```
finc(cons,Type2,Place2,Ton2,2) --> p(cons,Type1,Place1,Ton1,1),
      p(cons,Type2,Place2,Ton2,1),
      {vadher(Ton1,Ton2)}, % the vowel adherence condition
      {(Type1=liq,Place1=pal,Ton1=ton,not((Type2=liq,Place2=pal));
      Type1=liq,Place1=dent,Ton1=ton, Type2\=liq; % all, not lr
      Type1=nas,Place1=lab,Ton1=ton, % m combinations
      (Type2=nas,Place2=dent,Ton2=ton; % mn
      Type2=fric,Place2=pal,Ton2=ton; % mj
      Type2=klus,Place2=dent,(Ton2=ton;Ton2=tonl); % md,mt
      Type2=klus,Place2=lab,(Ton2=ton;Ton2=tonl); % mb, mp
      Type2=fric,Ton2=tonl); % mf,ms
      Type1=nas,Place1=dent,Ton1=ton, % n combinations
      (Type2=fric,Place2=pal,Ton2=ton; % nj
      Type2=klus,Place2=dent,(Ton2=ton;Ton2=tonl)); % nd,nt
      Type1=fric,Place1=lab,(Ton1=ton;Ton1=tonl), %v,f combonations
      Type2=klus,Place2=dent,(Ton2=ton;Ton2=tonl); % vd, vt, ft
      Type1=klus,Place1=pal,(Ton1=ton;Ton1=tonl), % g,k combinations
      Type2=klus,Place2=dent,(Ton2=ton;Ton2=tonl); % gd,gt,kt
      Type1=klus,Place1=lab,(Ton1=ton;Ton1=tonl), % b,p combinations
      Type2=klus,Place2=dent,Ton2=tonl; % bt,pt
      Type1=fric,Place1=dent,Ton1=tonl, % s combinations
      Type2=klus,(Place2=lab;Place2=pal;Place2=dent),Ton2=tonl)}.
      % sp,sk,st
```

Again, one may find alternative ways of formulating these restrictions. We will show how they can instead be formulated in terms of filters below.

It is not clear exactly which final clusters are to be accepted in Swedish. As a consequence it is not clear which rules should be set up. One problem is *-mj* and *-nj*. These clusters only occur in morphologically marked imperative forms *(tämj, tänj)* and they do not agree with the order established beteen *m, n* and *j* initially, where *j* occurs closest to the vowel *(mjuk, njuta)*. The final *-jm* and *-nj* would have been more natural and they are in fact found in some loanwords such as *dajm, sejn(fall)*. Another problem is that certain clusters containing *s*, *k* and *t* may prolong the syllables in a somewhat abnormal way, due to morphological pressure. There is an inflectional suffix *-sk* which may be added to all words except those ending in *s*, resulting in final sequences such as *rdsk (Sigurdsk), Herbstsk*. Such adjectives may be inflected for neuter by adding a *t*, resulting in clusters such as *-rdskt, -rbstskt* and such neuter adjectives may be treated as nouns allowing the addition of an *s*, resulting in clusters such as *-rbstskts*. But in normal words no more than three consonants are allowed. The following rule allows the addition of a final *s*, unless there is one already.


```
finc(cons,fric,dent,tonl,3) --> finc(cons,Type1,Place1,Ton1,2),
            {not((Type1=fric,Place1=dent,Ton1=tonl))},
            [s]. % s may be added, if the previous cons is not s
```

The following rule allows the addition of a *k* to a final *s*.

```
finc(cons,klus,dent,tonl,3) --> finc(cons,fric,dent,tonl,3),[k]. % k to s
```

The following three rules give all Swedish initial clusters, all final clusters, all syllables, respectively. The rule for initial clusters filters out *skn-* and *stv-*. The third rule combines all the initial consonants with all all vowels and all final clusters. The number of vowels is 8. The number of initial clusters is 47, the number of final clusters is (about) 160. The number of possible monosyllables in Swedish is thus about 60 000. It takes some time to generate and print out all of these according to our program!

```
incluster(P) :- inc(cons,Type1,Place1,Ton1,3,P,[]),P\=[s,k,n],P\=[s,t,v].

fincluster(P) :- finc(cons,Type2,Place2,Ton2,3,P,[]).

fullstavelse(V) :- inc(cons,Type1,Place1,Ton1,3,P,[]),P\=[s,k,n],P\=[s,t,v],
            p(voc,Type2,Place2,Ton2,1,F,[]),append(P,F,S),
            finc(cons,Type3,Place3,Ton3,3,N,[]),
            append(S,N,V),
            print(V),nl. % gives in + vowels + final= all syllables
```

Approach III

The following rules start from the vowel to which final consonants may successively be added resulting in higher bar values. Finally the vowel bar value 4 may append an initial cluster resulting in a vowel bar value 5, i.e. a full syllable. We use the category syllable *(s)* in the following rules. The features of the last consonant are added after the vowel features in the extended frame as can be seen.

The following rule states that a syllable may consist of one vowel only.

```
s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,1) -->
      p(voc,Height,Front,Round,1). % e.g. a, i,u
```

The following rule states that a syllable bar 2 may consist of a vowel and a following consonant. Note that the features of the added consonant are percolated to the slots 6,7,8 which allows the formulation of constraints.

```
s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,2) -->
      s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,1),
      p(cons,Type2,Place2,Ton2,1). % e.g. al, ak
```

The following is the important rule which shows which 2-member final clusters are allowed. Note that the rules are now formulated as filters using the built-in predicate *not.*

```
s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,3) -->
      s(voc, Height, Front, Round, cons,Type1,Place1,Ton1,2),
      p(cons,Type2,Place2,Ton2,1),
      {vadher(Ton1,Ton2), % general vowel adherence constraint
      not((Type1=Type2,Place1=dent)), % no dent first if same type
      not((Type1=Type2,Place1=Place2,Ton2=tonl)), % not vf, dt...
      not((Place1=lab,Place2=pal)), % not pk, bg
      not((Place1=pal,Place2=lab)), % not kp, gb
      not((Type1=fric,Type2=liq)), % not vl,jl
      not((Type1=klus,Type2=liq)), % not kr, tl, gr...
      not((Type1=klus,Type2=nas)), % not kn, gn, tm, dm...
      not((Type1=fric,Type2=nas)), % not jn, jm
      not((Type1=nas,Type2=liq)), % not ml, nl
      not((Type1=nas,Place1=dent,Place2=lab)), % not nb, np
      not((Type1=nas,Place1=lab,
      Type2=fric,Place2=lab,Ton2=ton)), % not mv
      not((Type1=klus,Type2=fric,Place2\=dent)), % not dj,dv
      not((Type1=klus,Place1=lab,Place2=dent,Ton2=ton)), % not bd
      not((Type1=fric,Place1=lab,Type2=klus,Place2=lab))}.%not vb,vp,fp
```

Again, one may formulate these constraints in several ways. The following is the rule which allows the addition of *s,* except to *s.*

```
s(voc, Height, Front, Round, cons,fric,dent,tonl,4) -->
      s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,3),
      p(cons,fric,dent,tonl,1), % s
      {not((Type2=fric,Place2=dent,Ton2=tonl))}. % if not s
```

The following rule allows the addition of *k.*

```
s(voc, Height, Front, Round, cons,klus,pal,tonl,4) -->
      s(voc, Height, Front, Round, cons,fric,dent,tonl,3), % s
      p(cons,klus,pal,tonl,1). % k
```

The following rule (not in DCG) appends the initial part to the vowel with its consonants generating full syllables, i.e. syllables bar value 5. We have filtered out the initial clusters *stv, skn.*


```
s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,5,S,[]) :-
      inc(cons,Type1,Place1,Ton1,3,P,[]),P\=[s,k,n],P\=[s,t,v],
      s(voc, Height, Front, Round, cons,Type2,Place2,Ton2,4,Q,[]),
           append(P,Q,S). % initial clusters plus syllables bar 4
```

Conclusions

AXG provides compact rules for generating consonant clusters and syllables. Such rules can be written for any language. All relevant generalizations can be formalized conveniently on the basis of the features. There are generally several approaches available as illustrated. Different approaches may be chosen for different languages. There are also several ways of expressing what can be generated and what cannot. One should ideally be able to refer to universal constraints, but it is not clear which these are. The evaluation of the different approaches must take additional facts and theoretical points of view into account.

The rules described may be used to generate syllables for practical use in devising trade marks and names for instance, but it is then easier to list the initial and final clusters and have the computer combine them with vowels. Such a program will certainly run faster than the one demonstrated here.

Acknowledgement

I am indebted to Birgitta Lastow for comments on this paper.

Demo

The following is a print-out of syllables generated as initial clusters followed by *a.*

```
:- instavelser(X)

ca sha ha sa fa va ja pa ka ta ba ga da la ra ma na fja fla fra fna vra
pja pla pra kva kla kra kna tva tra bja bla bra gla gra gna dva dra mja
nja spa ska sta spja spla spra skva skla skra stra no
```

References

Basbøll, Hans. 1977. 'The structure of the syllable and a proposed hierarchy of distinctive features'. In W. Dressler & O. Pfeiffer (eds.), *Phonologica 1976*, 143-148. Innsbruck.

Chomsky, Noam. 1957. *Syntactic structures.* The Hague: Mouton.

Clements, George. 1990. 'The sonority cycle and syllable organization'. In W. Dressler, H. Luschützky, O. Pfeiffer & J. Rennison (eds.), *Phonologica 1988*, 283-333. Cambridge University Press.

Jackendoff, Ray. 1977. *X-bar Syntax. A study of phrase structure.* Cambridge (Mass): MIT Press.

Jespersen, Otto. 1897-99. *Fonetik – en systematisk framstællning af læren om sproglyd.* København: Schubotheske Forlag.

Sigurd, Bengt. 1955. 'Rank order of consonants established by distributional criteria'. *Studia Linguistica* 9, 8-20.

Sigurd, Bengt. 1965. *Phonotactic structures in Swedish.* Lund: Uniskol.

Sigurd, Bengt. 1994. 'Fragments of a Swedish X-bar grammar (AXG)'. In Christer Platzack et al. (ed.), *Språkbruk, grammatik och språkförändring. Festskrift till Ulf Teleman*, 219 - 228.

Sigurd, Bengt & Birgitta Lastow. 1993. 'Fragments of a Japanese Appending X-bar Grammar (AXG)'. *Proc. Natural Language Processing Pacific Rim Symposium.* Fukuoka Institute of Technology, Japan.

Wood, Sidney. 1994. 'Syllable structure and the timing of speech gestures. An analysis of speech gestures from an X-ray motion film of Bulgarian speech'. *Proc. Third Congr. International Clinical Linguistics and Phonetics Association.* Dept of phonetics, University of Helsinki.

Lund University, Dept. of Linguistics
Working Papers 42 (1994), 209-224

Nexus Grammar (NEXG) for Swedish and English

Bengt Sigurd and Barbara Gawrońska

Introduction

Nexus Grammar (NEXG) is characterized by its focus on the syntactic unit consisting of the subject and the finite verb and since Jespersen 1924 known as the sentence nexus. The importance of the sentence nexus is clear from the fact that typological studies have shown that agreement between subject and predicate is the basic kind of agreement in languages. Other evidence of the importance of the nexus is the fact that some languages treat the nexus as a combinatory unit which is found in one straight (normal, right) form and one form where the order between the subject and the finite verb is inverted. NEXG is designed to treat languages for which a sentence nexus can be identified, e.g. Swedish and English. NEXG captures the function of word order as a marker of mode and topic. It also sheds some interesting light on the restricted use of inverted nexus in English where an auxiliary *(do,* the equivalent of Swedish *göra*, if no other) is always required.

NEXG is inspired by ideas found in Diderichsen's field grammar (Diderichsen 1946), in particular the division of sentences into three parts Thus, NEXG can be regarded as an implementation of a variant of his grammar. Diderichsen divides all sentences into three parts: the initial part called the fundament *(fundamentet)*, the *nexus* part *(neksusledet)* consisting of the subject and the finite verb with an optional nexus (sentence) adverbial and a content part *(inneholdsledet)* which includes the remainder of the sentence, e.g. objects, predicatives, infinitives, participles, verb particles and certain adverbs. Diderichsen's basic idea is that the ('underlying, deep structure') order of the parts in the nexus is fixed (in Danish as in Swedish): finite verb + subject np + nexus adverb (*vsa* in his notation) and he identifies a corresponding order *(VSA)* in the content part of the sentence. This (inverted) nexus word order is found ('as the surface order') when an adverb or an object occurs before the nexus and he explains this as the