Implementing Basque Grammar on the Computer

ARTHUR HOLMER*
BENGT SIGURD*

INTRODUCTION

With the arrival of computers, texts (corpora) could be investigated much faster in order to establish dictionaries, concordances or word frequency lists. Such projects have developed into a special branch of linguistics, Corpus Linguistics (see Aijmer & Altenberg, 1991). But computers also offer possibilities to translate automatically between languages and this was in fact one of the first projects suggested after the computer had been invented (see Locke & Booth, 1955). A number of machine translation (MT) projects were started in the 1950s. Due to the practical potential of machine translation Russian-English, French-English, Japanese-English, Chinese-English, etc. various computer systems have been constructed and some commercial systems are offered although they cannot produce perfect translation, except within restricted domains. The MT projects required ways of implementing grammar on the computer beside lexicons and stimulated linguists to develop more formal methods of description, formal grammar, mainly following the works of Noam Chomsky.

This paper presents a computerized formal grammar of Basque which may be used for simple translation between Basque and other languages. We will give examples of translation from English to Basque and from Basque to Chinese. But the computerized grammar has a value also as a precise description of Basque. Basque is known for its relatively free word order and rich verb morphology. The free word order with complex agreement between the

* Department of Linguistics, Lund University.

verb and the associated noun phrases (subject, direct and indirect objects) offers interesting challenges to grammarians (see references).

We will describe the basic features of Basque grammar using a computerized grammar called Permutational Grammar (PG; see, Eeg-Olofsson & Sigurd, 2000). It is a variant of Swetra Grammar which has been presented in various publications, notably Sigurd (ed.) *Computational Grammars for Analysis and Machine Translation* (1994).

BASQUE SYNTAX PATTERNS

The following sentence illustrates possible positions of adverbials, as shown by the adverb *gaur* 'today'.

1.	Peruk	(gaur)	Amaia	(gaur)	ikusi	du	(gaur).
	Peru-erg	(today)	Amaia-abs	(today)	seen	has	(today)

'Peru has seen Amaia today'.

Other variants with different word orders include: *Peruk (gaur) ikusi du (gaur) Amaia (gaur)*, *Amaia (gaur) Peruk (gaur) ikusi du (gaur)* and *Amaia (gaur) ikusi du (gaur) Peruk (gaur)*. In all these examples *Peruk*, the ergative (ERG) form of the proper name *Peru,* is the subject. *Amaia* which is in absolutive Case (henceforth ABS) is the object. The word order changes thus have no effect on the functional interpretation of the noun phrases.

While the pattern in (1), with SOV word order, is the most common (cf De Rijk 1969), many others are acceptable, given a suitable context. The most important factor governing Basque word order is information structure — the focussed element (such as a wh-word in a wh-question) is obligatorily placed immediately before the main verb.

Another constraint concerns the relative ordering of main verb and auxiliary. In affirmative clauses, the main verb precedes the auxiliary as in *ikusi du* 'seen has3sA-3sE' ('has seen'; third person singular absolutive and third person singular ergative). In negated clauses, on the other hand, the auxiliary obligatorily occurs immediately after the negator *ez*, whereas the main verb is positioned to the right of this cluster, not necessarily immediately.

The subject of an intransitive clause is realized in ABS case (the same case as that of an object of a transitive construction). If there is an auxiliary in the sentence it is inflected for person and number agreement with the subject, direct object and indirect object (if these arguments are present in the argument structure of the verb).

Another interesting feature of Basque, in keeping with the head-final nature of the language, is its placement of the relative clauses in front of its head (the correlate), cf examples (2a - c). The relative marker is formed by suffixing the relativizer *-en* to the auxiliary verb.

2a.	Peru	ikusi	du-en	neska	joan	da.
	Peru	see	have-3sA-3sE.PRES-REL	girl	go	is-3sA

'The girl who has seen Peru has gone'.

b.	Peru	ikusi	zuen	neska	joan	da.
	Peru	see	have-3sA-3sE.PRET-(REL)	girl	go	is-3sA

'The girl who saw Peru has gone'.

c. Peru-k ikusi du-en neska joan da.
Peru-ERG see have-3sA-3sE.PRES-REL girl go is-3sA
'The girl whom Peru has seen has gone'.

We will present the formal rules for some of these sentences. We will not, however, discuss all the complexities and subtleties of Basque syntax. For more details, the reader is referred to the references given in the bibliography.

A RULE FOR TRANSITIVE SENTENCES WITH ORDER VARIATIONS

Basque word order patterns have to be distinguished on the basis of the argument structure of the verb and the mode of the sentence. Negated sentences have to be treated separately. Furthermore, patterns with synthetic (main verb) inflection have to be distinguished from cases with periphrastic (auxiliary) inflection. Let us start with a transitive sentence with the object in focus (3).

3. Peru-k gaur Amaia ikusi du.
Peru-ERG today Amaia-ABS seen have-3sA-3sE
'Peru has seen Amaia today'.

In Basque the word order is very free and it is therefore convenient to describe Basque sentences by a rule which reorders (permutes) a basic order of noun phrases, verbs and adverbials in all possible ways with certain restrictions. This is the idea behind Permutational Grammar used here.

The other problem in Basque is the agreement which implies different verb forms for different sets of subject and object noun phrases. It is generally stated - with Latin as a pertinent example - that the agreement markings make it possible to have free word order in a language as the functional roles of the constituents can be identified anyway by the agreement (case) endings.

The agreement relations between the verb and the noun phrases can be handled by variables on the constituents, variables which occur both at the verb and the noun phrases involved denoting that the subject noun phrase form has to agree with the verb form, i.e. have the same (or combinable) values. The description of a language requires a great number of variables as will be obvious from the rules below.

The Permutational Grammar rules are generative rewriting (arrow) rules (following Chomsky 1957) which are written in the programming language Prolog (using the formalism Definite Clause Grammar, DCG) and can be run directly on the computer. The rule below states that there is a Basque sentence type (bs), which covers certain declarative sentences (Mode:d(eclarative)), the d is placed in the first slot after bs, the focussed element is registered in Fo, and the functional representation in the third slot. Note that the linear order in the functional representation does not reflect surface word order, which is instead brought about by the operation permute after the arrow (—>). The predicate permute reorders the list of phrases, bnp(Agr1,N1), badvp(A1), etc into a great number of other lists represented by the variable M. Capital letters denote variables in Prolog and comments are preceded by %.

```
4.  bs(d,Fo,[subj(N1),pred([V,As,T]),obj(N2),advl(A1),
    advl(A2),advl(A3),co(N1,Mo,C,F2)]) -->
    {permute([bnp(Agr1,N1),badvp(A1),bnp(Agr2,N2),
    badvp(A2),bvt([V,As]),bauxt(Agr1,Agr2,T),badvp(A3),
    btail(C,F2)],M)},
    {imbefore(bvt([V,As]),bauxt(Agr1,Agr2,T),M)}, % v before aux
    {imbefore(Fo,bvt([V,As]),M)}, % focus before main verb
    {Fo  badvp([])}, {last(btail(C,F2),M)}, surf(M).
```

The functional representation of this sentence type is what is included in the bracket immediately to the left of the arrow, i.e. (5).

```
5.  [subj(N1),pred([V,As,T]),obj(N2),
    advl(A1),advl(A2),advl(A3),
    co(N1,Mo,C,F2)].
```

This functional representation shows the meaning (value) of the functional roles when all rules including phrase rules, morphological rules and lexical rules are applied and the variables are instantiated. The meaning (value) of the subject is denoted by the variable N1 and when instatitiated it may be m(peru,prop), which is the semantic representation of the given name *Peru*, regardless of whether it be realized in ergative, absolutive or dative case. The value of the predicate is shown by [V,As,T] – the main verb value (V), the Aspect value (As) and the tense value (T), the latter two of which are given by the auxiliary. When instatitiated by a main verb form and an auxiliary it may be realized as the following (6).

```
6.  [see,perf,pres].
```

The bracket including the functional roles (5) includes three adverbial slots (with the variables A1, A2, A3), which is generally enough for ordinary sentences. There is also a last constituent named co. This constituent (which we will not treat in detail here) has to do with possible coordination and the use of full stop or question mark in other cases. Its simplified surface equivalent is btail. The category labels should otherwise be self-explanatory.

The realization of the functional representation is thus shown to the right of the generative arrow in a bracket parenthesis. The Basque categories have the prefix b to distinguish them from e.g. English categories which have the prefix e when several grammars are run simultaneously as in machine translation. A possible adverbial badvp(A1) is placed next in the surface list of phrases and then a noun phrase bnp(Agr2,N2), corresponding to the object (N2) in the functional representation. The variable N1 is the value of the subject as given by the lexical words and the agreement variables Agr1 and Agr2, also occurring on the auxiliary (bauxt), cover complex values of number, person and case. Thus the formula agr(sg,p3,erg) represents third person singular ergative. Number is marked in the first slot, person in the second and case in the third. If a noun phrase has the variable Agr1 and the verb also carries the variable Agr1 this means that the phrases have the same agreement features, or more correctly: features which can be unified. The checking of agreement is handled by the unification mechanism in Prolog. By this different markers (e.g. erg and abs) cannot unify, but identical markers can. If one (or both) markers are uninstantiated unification is also possible. Uninstantiated values are written as underscore (_). Thus agr(sg,_,erg) and agr(_,_,erg) can unify, but agr(sg,_,abs) and agr(pl,_,_abs) cannot. Uninstantiated values get numbers when the program is run.

The square bracket parenthesis after the arrow shows one basic order, but the preceeding Prolog predicate permute allows this order to be changed in all possible ways (permutations) in order to generate other word orders. It is the use of the permutation predicate which gives this variant of Swetra grammar its great power (for a definition of permute, see Eeg-Olofsson & Sigurd 2000). The following (7) illustrates the lexical listing of some Basque transitive verbs (bvt).

7. bvt([see,perf]) —> [ikusi].
   bvt([see,imp]) —> [ikusten].
   bvt([see,fut]) —> [ikusiko].
   bvt([learn,imp]) —> [ikasten].
   bvt([learn,fut]) —> [ikasiko].
   bvt([write,perf]) —> [idatzi].
   bvt([write,imp]) —> [idatzen].

WORD ORDER CONSTRAINTS

If no constraint was introduced on the sequence of phrases in the list M, all logically possible permutations would be permitted. However, it is a characteristic of Basque that the main verb must occur immediately before the auxiliary (except in negated sentences) and this condition is implemented by the code within the curly condition brackets used in Prolog.

8. {imbefore(bvt([V,As]),bauxt(Agr1,Agr2,T),M)}

This code states that the constituent bvt([V,As]) occurring in any of the permuted lists (M) must occur immediately before the auxilliary constituent bauxt(Agr1,Agr2,T). Note that it is the auxiliary that carries the agreement features Agr1, (also to be found in the first NP representing the subject) and Agr2 (also to be found in the NP representing the object). The main verb carries the main meaning (V) and the aspectual meaning denoted by As. The auxiliary contributes the tense meaning denoted by the variable T.

A futher condition is implemented by the next curly brackets in the rule. It states that the focus constituent represented by the variable Fo must occur immediately before the main verb in all permutations.

9. {imbefore(Fo,bvt([V,As]),M)}

The final predicate surf(M) in rule 4 instructs the generator to realize the permuted list (M) as a linear string of words.

The syntactic Prolog rule above can generate examples such as the following which are all acceptable in Basque. The translations given are only some of the possible equivalents (10).

10a. Peruk gaur Amaia ikusi du. (SAOV)
"Today Peru has seen Amaia."

b. Peruk hemen Amaia ikusi du gaur. (SA$_1$OVA$_2$)
"Here Peru has seen Amaia today."

c. Gaur Amaia ikusi du Peruk hemen. (A$_1$OVSA$_2$)
"Today Peru has seen Amaia here".

The use of the predicate imbefore excludes ungrammatical examples such as (11).

11a.	*Amaia	Peruk	ikusi	gaur	du.
	Amaia	Peru-ERG	see	today	have-3sA-3sE
	(*V-AUX not adjacent)				

b.	*Peruk	ikusi	Amaia	du.
	Peru-ERG	see	Amaia	have-3sA-3sE
	(*V-AUX not adjacent)			

One may test the grammar by parsing a Basque sentence. This is shown by the following call for the analysis of [gaur, ‘Peruk’, ‘Amaia’, ikusi, du, hemen], where we get the solution (12) that the mode is d (declarative), the object is in focus, and the functional representation is as shown in F.

```
12.  bs(M, Fo, F, [gaur, ‘Peruk’, ‘Amaia’, ikusi, du, hemen], [])
     No.1 : M = d,
     Fo = bnp(agr(sg, p3, abs), m(amaia, prop)),
     F = [subj(m(peru, prop)),
     pred([see, perf, pres]), obj(m(amaia, prop)),
     advl(m(today, _94197)), advl([]), advl(m(here, _93507)),
     co(m(peru, prop), _10191, [.], [])]
```

Alternatively one may ask the grammar to produce a sentence given certain facts, the mode required, what is to be the focus, what is to be the subject, the predicate, etc.

The grammar also requires lexical rule (entries), which for an adverbial phrase may take the form as in (13).

```
13.  badvp([]) --> []. % empty, no adverb
     badvp(m(today,_)) --> [gaur].
     badvp(m(here,_)) --> [hemen].
```

The first item in the list gives an empty list indicating an optional adverbial. The syntactic rule given above (4) allows three adverbial positions which may then be empty.

PROGRAMMING NOUN PHRASES

The syntactic rules use the category noun phrase bnp, representing the various functional roles subject, object and dative object. The noun phrase carries different agreement values which fit the requirements of the verb.

Noun phrases may include relative clauses, determiners, adjectives and numbers. In order to handle these cases, successively smaller noun phrases of different hierarchical order (bar values) are used (bnp3, bnp2, bnp1). The rule in (14) is rather trivial in that it indicates that a bnp may consist simply of a phrase of level bnp3. (A further alternative rule defines another possible bnp, namely a relative clause followed by a bnp3, and the purpose of this is to allow for noun phrases with relative clauses). Thus, this rule simply allows for a bnp which does not contain a relative clause.

```
14.  bnp(Agr,N) —> bnp3(Agr,N). % only np3 (no rel clause)
```

The rule in (15) allows the formation of a bnp3 which consists simply of a lexical head bnlex, provided that this lexical head is a proper noun (prop).

The consequence of this is that a proper noun such as a name may occur as a bnp in its own right, or may occur together with a relative clause (by virtue of forming a bnp3 which in turn may be combined with a relative clause to form a bnp.

15.
```
bnp3(Agr,N) —> bnlex(Agr,N), {N=m(_,prop)}.
```

Examples of lexical entries of proper nouns are given in (16).

16.
```
bnlex(agr(sg,p3,abs),m(amaia,prop)) —> ['Amaia'].
bnlex(agr(sg,p3,abs),m(peru,prop)) —> ['Peru'].
```

ARTICLES

In Basque the definite marker *(a)* is suffixed to the noun phrase (if an NP contains several words, the article is suffixed to the last word). The following rule (17) displays the affixation of the article to an NP containing only one word. This is the first step in our derivation of nominal morphology. It affixes the definite determiner *-a* to a bnp2 to create a bnp3, e.g. *gizona* ('the man') from *gizon* ('man'), *ardoa,* 'the wine', from *ardo*, 'wine'.

17.
```
bnp3(Agr,[the,B],[F1]) :-
bnp2(Agr,B,[F],[]),B=m(B1,P),P  prop,
(ccat(R,a,F),F1=F ; % don't add if ending in a
ccat(F,a,F1)). % add def art a
```

The operator :- denotes implication in Prolog in such a way that what comes before the :- operator is true if what comes after the :- is true (in somewhat simplified terms). This rule can be verbalized such that if there is a bnp2 with the agreement features Agr, the meaning B and the form F and it is not a proper name (P prop), a corresponding bnp3 may be formed by concatenating (ccat - a variant of the built-in predicate concat) an *a* to the form F, giving a new form F1.

The first section of the condition (ccat(R,a,F), F1=F) states that if the form F already ends in *-a*, no extra suffix is added (F1 = F). The marker the, meaning definite, is added to the meaning B as seen in the square brackets [the,B]. If there are several words in F, e.g. if there is a following adjective, the *-a* has to be concatenated to the last item. We will not present this variant of the rule here.

A further alternative to forming a bnp3 is the addition of the indefinite article *bat* ('one'). This is illustrated in (18). The ensuing meaning representation includes the indefinite article *a* (from English). Note that this is only possible if bnp2 is not a proper name (P prop). We have not listed the definite nor the indefinite article in Basque as separate categories.

18.
```
bnp3(Agr,[a,B]) —>
bnp2(Agr,B), {B=m(B1,P),P  prop},[bat].
```

DEMONSTRATIVES AND ADJECTIVES

As an alternative to affixing of the definite article *-a*, a demonstrative may be added to a bnp2 to yield a bnp3. Such elements include the demonstratives *hau* (this) and *hori* (that). Example (19) illustrates a rule which derives bnp3 from bnp2 followed by a demonstrative (bdet). The lexical items for the two demonstratives are also given. The meaning of the demonstrative item (D) is placed before the meaning of np2 in brackets [D,B].

```
19.  bnp3(Agr,[D,B]) —> bnp2(Agr,B),bdem(D).
     bdem(this) —> [hau].
     bdem(that) —> [hori].
```

A grammatical bnp2 may consist of a bnp1 followed by an adjective. This case is covered by rule (20). One possible adjective, *handi* 'big', is listed in (20).

```
20.  bnp2(Agr,B) --> bnp1(Agr,B).
     bnp2(Agr,[A,B]) --> bnp1(Agr,B), ba(A). % with adj
     ba(big) --> [handi].
```

A bnp1 consists of a simple lexical item and this case is covered by the following rule, which uses lexical items as those listed below.

```
21.  bnp1(Agr,B) --> bnlex(Agr,B).
```

The following (22) is a list of lexical items with agreement features and meaning representations. Word meanings are written in the format m(L,G), where L is the lexical meaning and G the grammatical meaning (e.g. sg, pl, prop, pres, past). We will show below how the absolutive form can be used as a basis for generating the other cases.

```
22.  bnlex(agr(sg,p3,abs),m(book,sg)) --> [liburu].
     bnlex(agr(sg,p3,abs),m(girl,sg)) --> [neska].
     bnlex(agr(sg,p3,abs),m(basquelanguage,_)) --> [euskara].
     bnlex(agr(sg,p3,abs),m(wine,_)) --> [ardo].
     bnlex(agr(sg,p3,abs),m(boy,sg)) --> [mutil].
     bnlex(agr(sg,p3,abs),m(person,sg)) --> [gizon].
     bnlex(agr(sg,p3,abs),m(letter,sg)) --> [eskutitza].
     bnlex(agr(sg,p3,abs),m(money,_)) --> [diru].
     bnlex(agr(sg,p3,abs),m(book,sg)) --> [liburu].
     bnlex(agr(sg,p3,abs),m(table,sg)) --> [mahai].
     bnlex(agr(sg,p3,abs),m(house,sg)) --> [etxe].
```

The above rules are sufficient to generate noun phrases up to the level of bnp3 (i.e. noun phrases without relative clauses).

NUMBER ATTRIBUTES

Alternatively, a numeral may be added to bnp2 to create a bnp3. This case is given by (23). Numerals do not occur with proper names, as is stated by P prop. Some numerals are also listed.

```
23.  bnp3(Agr,[N,B]) -->
     bnum(N),bnp2(Agr,B), {B=m(B1,P),P  prop}. % with num
     bnum(2) --> [bi].
     bnum(3) --> [hiru].
     bnum(4) --> [lau].
     bnum(5) --> [bost].
```

NOUN PHRASES WITH RELATIVE CLAUSES

The category bnp may incude a relative clause as is shown in the following example (24).

24. Peru ikusi duen Amaia
Peru-(ABS) seen 3sA-3sE-AUX-REL Amaia
'Amaia who has seen Peru'.

The relative clause in Basque precedes the noun (in our system bnp3). The agreement features of the head should be carried over to the whole noun phrase. The meaning of the relative clause is represented by R in the square brackets [N,R], where N is the meaning (value) of bnp3.

25.
```
bnp(Agr,[N,R]) -->
bs(rel,N,R),bnp3(Agr,N).
```

The relative clause is defined as the other clauses (sentences) but is primarily distinguished by lacking an element (the relativized noun phrase). To distinguish it from other clauses it has the word rel in the mode slot: bs(rel,N,R). The N in the focus slot is a copy of the correlate which is percolated down into the relative clause to the functional role which is relativized.

The following is a rule for a relative clause where the subject is relativized. The subject noun phrase is therefore missing in the string of categories but N occurs in the functional representation as it is percolated down from the correlate, the N in bnp3(Agr,N).

26.
```
bs(rel,N,[subj(N),pred([V,Asp,T]),obj(N2),
advl(A1),advl(A2),advl(A3),co(N,Mo,C,F2)]) -->
{permute([badvp(A1), bvt([V,As]), badvp(A2),
bnp3(Agr2,N2),badvp(A3),
{bnlex(Agr1,N,F,[])}, % get Agr features from correlate
rel(Agr1,Agr2,T)],M)}, % special auxiliary in rel clause
{imbefore(bvt([V,As]),rel(Agr1,Agr2,T),M)},
{last(rel(Agr1,Agr2,T),M)}, % relative auxiliary last
surf(M).
```

Note that the noun phrase in the relative clause is a bnp3 (which may not contain a relative clause). Thus, a relative clause may not contain another relative clause in this grammar. The purpose of this in our system is to avoid recursion problems. (It also reflects performance difficulties).

The rule further makes use of a separate set of auxiliaries termed rel (syntactically a combination of an indicative auxiliary such as *du* '3sA-3sE-AUX-PRES' and the relativizer itself, which is the suffix *-en*. These are derived in the morphological component (see below).

The word order constraint requires the main verb to occur before the relative auxiliary and the relative auxiliary to occur last.

The following (27) illustrates the generation of an ergative noun phrase which incorporates subject relativization.

27.
```
bnp(Agr,F,X,[])
Agr = agr(sg, p3, erg), F = [[subj(m(amaia, prop)),
pred([learn, fut, pres]), obj(m(basquelanguage,_)), advl([]),
advl([]), advl([]), co(m(amaia, prop), _5415,
_5418, _5421)], m(amaia, prop)],
X = [euskara, ikasiko, duen, 'Amaiak']
```

euskara ikasiko duen Amaiak
Basque-(ABS) learn-FUT 3sA-3sE-AUX-PRES-REL Amaia-ERG
'Amaia, who will learn Basque...'

PROGRAMMING INTRANSITIVE SENTENCES

Basque has two types of sentence patterns without an object. One is the type headed by what is known in generative circles as unergative verbs, which are semantically intransitive but agentive and pattern grammatically with transitive verbs. In traditional work on Basque, they are often treated as a subgroup of transitive verbs. To prevent overgeneration, such verbs are treated in our model as a separate class, termed *semi-transitives*, which combine transitive morphology with an intransitive sentence pattern (28).

28. Amaia-k gaur kantatu du.
Amaia-ERG today sing have-3sA-3sE
'Amaia has sung today'.

The verbs in these sentences have been denoted by bvs. The auxiliary takes double agreement although there is no overt object. In the rule below we have denoted the object by the empty set [] in the functional representation. There is no object NP in the surface sequence of categories. The agreement value of the auxiliary is set at agr(sg,p3,abs), which represents a default value. Note that this sentence type uses a transitive auxiliary (bauxt).

29.
```
bs(d,Fo,[subj(N1),pred([V,As,T]),obj([]),advl(A1),
advl(A2),advl(A3),co(N2,Mo,C,F2)]) -->
{permute([badvp(A1),bnp(Agr1,N1),bvs([V,As]),
bauxt(Agr1,agr(sg,p3,abs),T), badvp(A2),
badvp(A3),btail(C,F2)],M)},
{imbefore(bvs([V,As]),bauxt(Agr1,agr(sg,p3,abs),T),M)},
{imbefore(Fo,bvs([V,As]),M)},
{Fo  badvp([])}, {last(btail(C,F2),M)}, surf(M).
```

Some verb forms pertaining to the class of bvs are listed in (30).

30.
```
bvs([dance,perf]) -->[dantzatu].
bvs([dance,imp]) --> [dantzatzen].
bvs([dance,fut]) --> [dantzatuko].
```

An example of a clause generated by the above rule is given below (31).

31.
```
bs(M, Fo, F, X, [])
No.1 : M = d, Fo = bnp(agr(sg, p3, erg), m(peru,prop)),
F = [subj(m(peru,prop)), pred([dance, perf, pres]), obj([]),
advl([]), advl([]), advl([]), co(_76644, _64297, [.], [])],
X = ['Peruk', dantzatu, du]
```

Peru-k dantzatu du.
Peru-ERG danced 3sA-3sE-AUX
'Peru has danced'.

UNACCUSATIVE SENTENCES

The second type of intransitive verb is that which is referred to in generative work as unaccusative. This is the type which is termed intransitive in traditional work on Basque. The main verb has been denoted by bvi in the present paper. An unaccusative verb requires the use of the intransitive auxiliary ‘to be’ rather than ‘to have’. This intransitive auxiliary is denoted by bauxi. The sole argument of such a clause is a bnp which carries the absolutive case feature and the auxiliary agrees with this. For reasons of uniformity an object slot has been included in the functional representation but set at [].

```
32.  bs(d,Fo,[subj(N),pred([V,As,T]),obj([]),advl(A1),
     advl(A2),advl(A3),co(N,Mo,C,F2)]) —>
     {permute([badvp(A1),bnp(Agr1,N),
     bvi([V,As]),bauxi(Agr1,T),badvp(A2),
     badvp(A3),btail(C,F2)],M)},
     {imbefore(bvi([V,As]),bauxi(Agr1,T),M)},
     {imbefore(Fo,bvi([V,As]),M)},
     {Fo  badvp([])}, {last(btail(C,F2),M)}, surf(M).
```

Such rules require bvi lexical entries (33).

```
33.  bvi([go,perf]) --> [joan].
     bvi([go,imp]) --> [joaten].
     bvi([go,fut]) --> [joango].
     bvi([come,perf]) --> [etorri].
     bvi([come,imp]) --> [etortzen].
     bvi([come,fut]) --> [etorriko].
```

SYNTHETIC INFLECTION

Certain Basque verbs can be inflected directly, without using an auxiliary. In order to demonstrate synthetic inflection we include the following rule (34). Note that no auxiliary is included in the pattern, and that the main verb is denoted by bfvt (“finite transitive verb”).

```
34.  bs(d,Fo,[subj(N1),pred([V,As,T]),obj(N2),advl(A1),
     advl(A2),advl(A3),co(N1,Mo,C,F2)]) —>
     {permute([badvp(A1),bnp(Agr1,N1), badvp(A2),
     bnp(Agr2,N2),bfvt(Agr1,Agr2,[V,As,T]),badvp(A3),btail(C,F2)],
     M)}, {imbefore(Fo,bfvt(Agr1,Agr2,[V,As,T]),M)},
     {Fo  badvp([])}, {last(btail(C,F2),M)},surf(M).
```

As with auxiliary verbs, the inflection of the main verb can be derived productively in the morphological component. In the present version, however, inflected main verbs are listed lexically. The following (35) illustrates some lexical items with bfvt verbs.

```
35.  bfvt(agr(sg,_,erg),agr(sg,_,abs),[see,imp,pres]) --> [dakus].
     bfvt(agr(pl,_,erg),agr(sg,_,abs),[see,imp,pres]) --> [dakuste].
```

DITRANSITIVE SENTENCES

Ditransitive verbs such as *eman* ‘give’ take, in addition to the agent in ERG, two object arguments, one in ABS (the object given) and one in DAT (the recipient). The auxiliary agrees in person and number with all three arguments of the clause. This is exemplified below.

36. Peruk liburua Amaiari eman dio.
Peru-ERG book-(ABS) Amaia-DAT give 3sA-3sE-3sD-AUX
'Peru has given the book to Amaia.'

This type of clause requires a special syntactic pattern, including a ditransitive main verb (bvd) and a ditransitive auxiliary (bauxd). The following demonstrates the rule for such a syntactic pattern (37).

```
37. bs(d,Fo,[subj(N1),pred([V,As,T]),obj(N2),dobj(N3),
    advl(A1),advl(A2),advl(A3), co(N1,Mo,C,F2)]) —>
    {permute([badvp(A1),bnp(Agr1,N1), badvp(A2),
    bnp(Agr3,N3), bnp(Agr2,N2),bvd([V,As]),
    bauxd(Agr1,Agr2,Agr3,T),badvp(A3), btail(C,F2)],M)},
    {imbefore(bvd([V,As]),bauxd(Agr1,Agr2,Agr3,T),M)},
    {imbefore(Fo,bvd([V,As]),M)}, % focus before verb
    {Fo  badvp([])}, {last(btail(C,F2),M)}, surf(M).
```

Some verbs required by this pattern are listed in (38).

```
38. bvd([give,perf]) --> [eman].
    bvd([give,imp]) --> [ematen].
    bvd([give,fut]) --> [emango].
```

SOME MORPHOLOGICAL RULES DERIVING CASE FORMS

A characteristic fact of Basque is that the case endings are suffixed to whole noun phrases including possible articles and attributives. The typical ergative ending *-k* is thus not only suffixed to the name *Peru* to give *Peruk* but also to *gizona* (the definite form of *gizon* 'man') to give *gizonak* (man-DET-ERG*)* to *gizon bat* ('a man'), to give the ergative form *gizon batek.* The ergative form of *gizon handi bat,* 'a big man' is *gizon handi batek.*

Following the hierarchy of noun phrases presented earlier it is convenient to add case suffixes to bnp3 deriving other forms of bnp3, to ensure that the case suffix is added after the definite article or demonstrative. Given that the absolutive case form is morphologically unmarked, it is natural to take this as the basis as shown in the following rule deriving ergative forms for singular nouns. The rule states (using the operator :-) that the existence of bnp3(agr(sg,person,abs), B,[F],[]) implies the existence of bnp3(agr(sg,person,erg), B,[F1|X],X) provided that *-k* is added to the absolutive form F unless the latter ends in a consonant, in which case *-ek* is added (39). The use of |X here serves to map the result of the derivation into a DCG format, and will not be commented no further.

```
39. bnp3(agr(sg,Pers,erg),B,[F1|X],X) :-
    bnp3(agr(sg,Pers,abs),B,[F],[]),
    (ccat(W,C,F),cons(C),ccat(F,ek,F1); % ek after consonant
    ccat(F,k,F1)). % else add k
```

This rule derives *Peruk* from *Peru.* The form *Peru* is listed in the lexicon as an absolutive form and being a proper name it is also a bnp3. The rule can also derive the ergative definite and indefinite forms, but the situation is more complicated when the ergative ending is to be added to a bnp3 consisting of several words. This situation can also be handled in Prolog, but we will not present these rules here. The result of such rules would give ergative forms such as e.g. *gizon batek* 'a man-ERG', and *gizon handi batek* 'a big man-ERG'.

Analogously, dative forms can be derived according to the following rule (40), which gives *Amaiari* ('Amaia-DAT') from *Amaia* ('Amaia-ABS'), *gizon bati* ('one man-DAT') from *gizon bat* ('one man-ABS'). Here, too, noun phrases including several words would require more complex implication rules.

```
40.  bnp(agr(sg,Pers,dat),B,[F1|X],X) :-
     bnp(agr(sg,Pers,abs),B,[F],[]),
     (ccat(W,C,F),cons(C), ccat(F,i,F1); % if ending in consonant
     ccat(F,ri,F1)). % else add ri
```

The use of these and similar rules can derive the entire case morphology of Basque productively. In the present version, we have only presented rules deriving ergative and dative case.

DERIVING AUXILIARY MORPHOLOGY

Since Basque displays multiple verb agreement, a typical verb paradigm involves a two-dimensional table rather than a simple list. This is illustrated below for the present tense of the transitive auxiliary 'to have' (4). Further verb tables are required for various modes and tenses, as well as for ditransitive verbs, displaying verbal systems of enormous complexity as shown below (Fig 1; 2pol denotes polite second person)

		ERG						
		SG				PL		
ABS		1	2	2pol	3	1	2	3
SG	1	---	nauk/-n	nauzu	nau	---	nauzue	naute
	2	haut	---	---	hau	haugu	---	haute
	2pol	zaitut	---	---	zaitu	zaitugu	---	zaituzte
	3	dut	duk/-n	duzu	du	dugu	duzue	dute
PL	1	---	gaituk/-n	gaituzu	gaitu	---	gaituzue	gaituzte
	2	zaizuztet	---	---	zaituzte	zaituztegu	---	zaituztete
	3	ditut	dituk/-n	dituzu	ditu	ditugu	dituzue	dituzte

Fig 1. Inflection of 'to have' (PRES)

Furthermore, since certain subordinators in Basque (e.g. *-la* 'that', *-n* 'REL') are realized as verbal suffixes, this adds substantially to the complexity of the verbal paradigm. For any given person agreement configuration, there are 18 different mood/tense/complementizer forms of the auxiliary. For any given mood/tense, there are 184 different possible combinations of person agreement. This implies that for the auxiliary verb *izan* 'to be, have', the total number of combinations of person, mood and tense is 3312 forms. This makes it preferable to derive these forms in a morphological component rather than listing them lexically.

To make full use of the productive sections of the verb morphology, we have chosen to design a morphological component which is capable of deriving all of these forms rather than simply the subset of forms which is required by the syntactic component shown (i.e. indicative mood). Thus, the present morphological component is ready to be applied to a much more advanced syntactic component than the one being demonstrated here. The following sections outline the principal stages of this derivation.

The general outline is as follows. We define various different levels of the form of each verb (expressed as a predicate taking various person, number, mood and tense features as arguments). Each level serves as the basis for constructing a new level. After the basic underlying forms have been generated, a new level of morphology is introduced for each slot in the morphology (absolutive agreement, ergative agreement etc).

We begin by defining a predicate bvur (Basque auxiliary verb underlying representation) which combines the different person agreement features which can be represented by the Basque auxiliary.

41.
```
bvur([Agra,Agre,Agrd]) :-
real(Agra),megareal(Agre),megareal(Agrd),
noncoref(Agra,Agre),noncoref(Agra,Agrd),
noncoref(Agre,Agrd).
```

Note that the predicate bvur does not define any form. It simply defines possible combinations of person agreement. It specifies that Agra, agreement with an absolutive argument, must refer to one of the possible 8 person/number forms, whereas Agre and Agrd (ergative and dative agreement respectively) must refer to one of the 8 person/number forms or null (which is used to indicate that no agreement takes place). The predicates real and megareal are consequently defined as follows:

42.
```
real(A):-
member(A,[s1,f2,m2,s2,s3,p1,p2,p3]).

megareal(X) :-
member(X,[null,s1,m2,f2,s3,p1,s2,p2,p3]).
```

The predicate noncoref ensures that different agreement categories do not refer to the same person. Thus it prevents a form which for instance would take a first person subject and a first person object, a configuration which is impossible in Basque (reflexivity is instead coded by a reflexive phrase which is grammatically third person, although it may have non-third person reference: e.g. *nere burua* lit. 'my head'). At the same time noncoref allows a form to have the same person features for more than one agreement category if and only if that person is third person (as in *Paul has seen Mary*), or if either of the agreement categories is null. We do not define noncoref in detail here.

There is another problem with Basque agreement, however. In Basque, a ditransitive verb may not take an absolutive argument which is not third person. Thus, the following example would be ungrammatical in Basque.

43. *Zuk ni harakinari eman "naiozu".
2s-ERG 1s-ABS butcher-DAT give AUX-1sA-3sD-2sE
'You have given me to the butcher.' (Laka 1995)

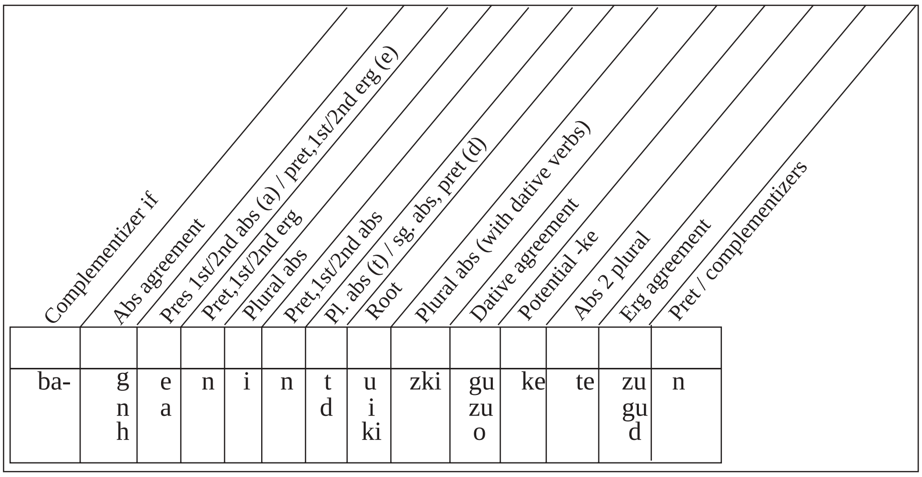
This phenomenon is termed *paradigm trimming* in the literature (Addis 1993). The present system filters out forms which do not obey this restriction by defining a new level of underlying representation for the verb, bvurpt. The filter states, somewhat simplified, that a bvurpt exists if there is a bvur, and either if the absolutive argument is third person (with both other arguments having real reference) or if either the ergative or the dative argument is null.

The next level introduces the possible combinations of the seven moods and two tenses. Due to cooccurrence restrictions, only ten different mood/tense forms actually exist, rather than the fourteen expected if conccurrence were free. We illustrate below the most productive of these rules (other moods can only cooccur with either past or present tense, but not with both).

```
44.  bvura(Agr,[Mood,Tense]) :-
     member(Mood,[ind,pot,subjunc]),
     member(Tense,[pres,pret]),
     bvurpt(Agr).
```

So far, we have still not taken any account of the form of the verb, only its grammatical features. In the next level we shall start generating underlying verb forms.

First, a few remarks should be made on the structure of the Basque verb. Maximally, a Basque auxiliary contains 14 morpheme slots (not all of which are used simultaneously, however). They are illustrated in Fig 2.



Complementizer if	Abs agreement	Pres 1st/2nd abs (a) / pret,1st/2nd erg (e)	Pret,1st/2nd erg	Plural abs	Pret,1st/2nd abs	Pl. abs (t) / sg. abs, pret (d)	Root	Plural abs (with dative verbs)	Dative agreement	Potential -ke	Abs 2 plural	Erg agreement	Pret / complementizers
ba-	g n h	e a	n	i	n	t d	u i ki	zki	gu zu o	ke	te	zu gu d	n

Fig 2. The slots of the Basque auxiliary

For ease of exposition, the slots are marked with Greek letters starting from the root and covering first the prefixes, then the suffixes, and finally the complementizer *ba-* 'if'). Slots a through e host portmanteau morphemes which indicate certain configurations of person, number and tense.

Due to the complex specifications of the morphemes in slots a through e, it is unnecessarily costly in terms of processing capacity to derive these morphemes productively. For this reason we have chosen to fuse slots a - e with the root and list the ensuing strings lexically.

Given that the shape of the root and the behaviour of slots a - e vary depending on the transitivity of the auxiliary, and given that slots h and q are only relevant for verbs with dative agreement, the derivation splits at this stage into parallel lines of development, which will be reunified at a later stage. This is illustrated in Fig. 3. The purpose is to avoid having to lead every single type of verb through various different types of morphological process which may be irrelevant to it.

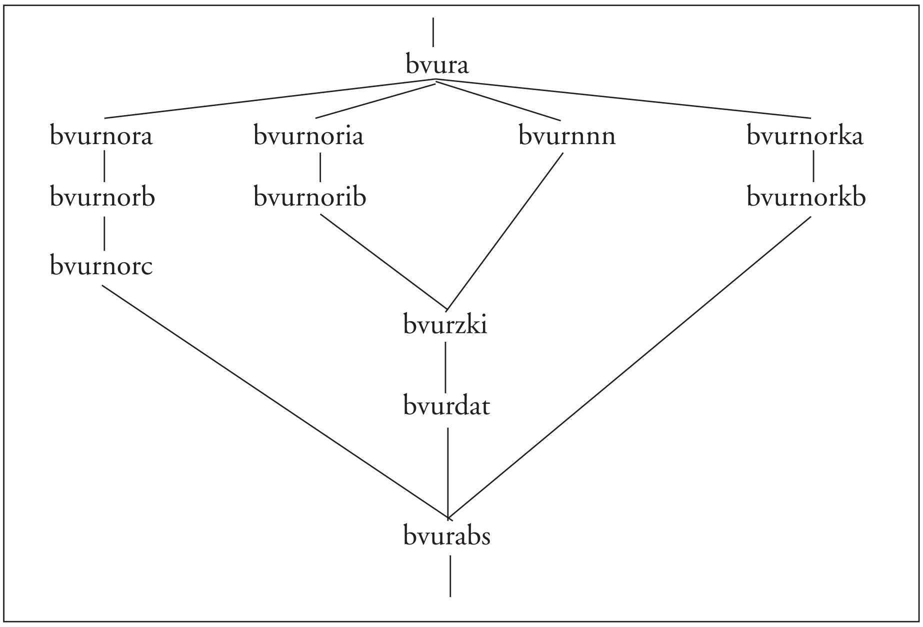


Fig. 3 Parallel derivations of transitives and intransitives

The bvurnora - bvurnorb - bvurnorc line defines processes which only concern intransitive verbs (referred to in the literature as *nor*-verbs). The bvurnoria - bvurnorib line defines processes which only hold for verbs which have an absolutive and a dative (*nori*) argument, but no ergative argument. The bvurnnn level defines processes the form of ditransitive verbs, and the bvurnorka - bvurnorkb line defines the forms and processes of transitive (*nork*) verbs. The rules defining each level vary in complexity.

The bvurzki level takes as its base any form which has a dative argument (i.e. either bvurnorib or bvurnnn), and suffixes *-zki-* to the root if the absolutive agreement is plural. The relevant rule is illustrated below:

```
45. bvurzki([A,B,C],[M,T],G) :-
    ((plform(A),Zkisuffix=[z,k,i]);
    (sgform(A),Zkisuffix=[])),
    (bvurnorib([A,B,C],[M,T],F);
    bvurnnn([A,B,C],[M,T],F)),
    join(F,Zkisuffix,G).
```

The bvurzki level serves as a base for dative agreement, leading to the level bvurdat. The predicate defining bvurdat is given below:

```
46. bvurdat([A,B,C],[M,T],G) :-
    sufagr(C,Suffix),
    bvurzki([A,B,C],[M,T],F),
    join(F,Suffix,G).
```

This level requires defining the relation between certain person features and corresponding morphemes. This is done with the predicate sufagr, as shown below.

```
47.  sufagr(s1,[c]).
     sufagr(f2,[y]).
     sufagr(m2,[q]).
     sufagr(s2,[z,u]).
```

Note that the morphemes used for 1st person singular, 2nd person familiar feminine and 2nd person familiar masculine are in themselves underlying representations ([c], [y] and [q] respectively), which will be filtered to the correct form before the derivation is finished. This is due to the fact that they are realized differently depending on whether they are word-final or followed by another morpheme (e.g. 1st person singular is realized as *-t* word-finally and as *-da-* word-internally).

The next level, bvurabs, takes as its base bvurnorc, bvurdat and bvurnorkb, and effectively reunites the lines of derivation. Any process taking place after this stage holds for any of the four types of person configuration. The predicate bvurabs ensures that absolutive agreement is correctly added to the stem. It also caters for prefixed ergative agreement in the past tense if the absolutive argument is third person (the so-called ergative displacement phenomenon). Thus, it is not concerned primarily with a specific category of agreement, but rather with the prefixed agreement slot in itself.

The remainder of the Basque verb can effectively be divided into a set of suffixed slots which each is responsible for a certain type of affix. Each level from bvurabs on will therefore add exactly one slot to the form and generate the correct morphology for it. For reasons of space, we shall not define each predicate in detail, but simply list the next three levels, and briefly describe what each does.

1) bvurke: adds *-ke* in potentials and apodosis conditionals
2) bvurte: adds *-te* for 2nd person plural absolutive
3) bvurerg: adds suffixed ergative agreement

After bvurerg, a new category is introduced, *type*. The predicate bvuren takes bvurerg as its base, and defines different kinds of usage of the auxiliary: aux for simple auxiliary (in a matrix clause), rel for a relative form in a relative clause, comp for a verb in a complement clause, subjla and subjen for two different kinds of complement clauses with subjunctive verbs (headed by the suffixes *-la* and *-n*, respectively) and finally auxif for clauses headed by the subordinator *ba-* 'if'). This is illustrated schematically below.

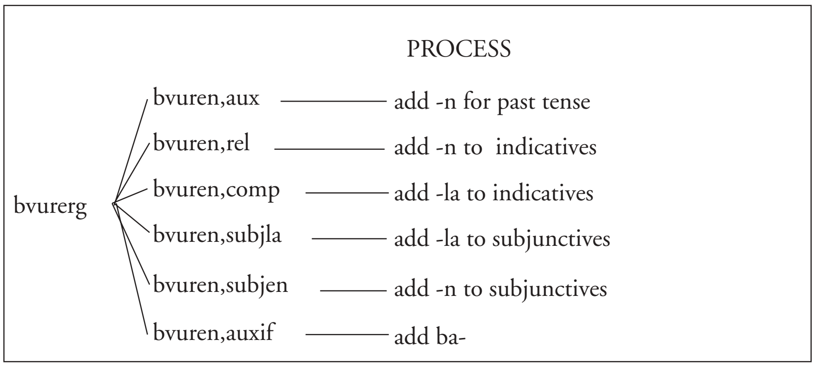


Fig 4. Past tense morpheme and complementizer affixation

The final predicate required by the morphological component is the one which delivers the surface form, baux. It, in turn, calls the predicate filter-form, which has two important characteristics. Firstly, it filters underlying morpheme representations and returns the surface representation by replacing certain sequences of atoms in the list by other sequences. Secondly, it converts the ensuing list to a word by fusing the atoms in the list.

The predicate baux thus delivers forms such as the following.

48. No.1: Agr = [p2, s1, null], Type = aux,
Moodtense = [pot, pres], Form = zaitzaketet

The parsing system makes use of the same rules. However, it requires a separate (and more complex) filter to derive an underlying representation from a surface form.

Finally the output of baux must be translated into a format which can be applied by the DCG grammar used in the syntactic component. This implies two processes: firstly, the features must be expressed as they are in the syntactic component, and secondly the form itself must be made compatible with a DCG format. For our present purposes we only need to translate indicative forms to a DCG formalism.

AUTOMATIC TRANSLATIONS

The function of the entire system is demonstrated in (49). Having loaded the grammars for English and Basque, we make a complex call consisting of three sections. Firstly, we call for an English sentence (es) having a functional representation F and the form given by the call (*Peru did not see Amaia*). Secondly, we subject the functional representation F to a transfer rule (ebtransf), giving the functional representation F1 with an expanded aspectual specification ([see,perf,past]). Thirdly, we call for a Basque sentence with the functional representation F1 and with the form Y, which is the Basque translation of the English sentence (further calls would generate all possible word orders in Basque).

49. **CALLS:**
es(M, Fo, F, [‘Peru’, did, not, see, ‘Amaia’], []),
ebtransf(F, F1), bs(_,_, F1, Y, [])

RESULTS:
No.1 : M = d, Fo = m(peru, prop),
F = [subj(m(peru, prop)), pred(m(see,past)),
obj(m(amaia,prop)), advl([]), advl(m(nix, _75618)), advl([]),
co(m(peru, prop), _73320, [], [])],
F1 = [subj(m(peru, prop)), pred([see,perf, past]), obj(m(amaia,
prop)), advl([]), advl(m(nix, _75618)),advl([]),
co(m(peru, prop), _73320, [], [])],
Y = [‘Peruk’, ‘Amaia’, ez,zuen, ikusi]

As a final illustration, we show that the same system can be used to translate Basque into Chinese (50). Note that this example of translation between Basque and Chinese does not require the use of a transfer rule. This is due to the fact that the aspectual systems of Basque and Chinese have been coded so similarly as to allow a sharing of the functional representation F1. This does not, however, imply any typological similarity between Basque and Chinese. In this case it is necessary to specify that the result be given in Romanized transcription (pinyin) rather than in Chinese characters. The glosses of the examples are given after the demo.

50. **CALLS:**

```
bs(_, _, F1, [eskutitza,ikasleak,idatzi,du], []), % from Basque
assert(lex(p)), % Use Pinyin transcription
cs(_, _, F1, X, []). % generate Chinese
```

RESULTS:

```
F1 = [subj(m(student, sg)), pred([write, perf, pres]),
obj(m(letter, sg)), advl([]), advl([]),
advl([]), co(m(student, sg), _73325, [], [])], % interlingua

X = [xue2sheng1, yi3jing1, xie3, xin4, le, .] % Chinese text
```

Eskutitza	ikasleak	idatzi	du.
letter-(ABS)	student-ERG	write	AUX-3sA-3sE

'The student has written the letter.'

Xuesheng	yijing	xie	xin	le.
student already	write	letter	PRF	

'The student has written the letter.'

The program demonstrated here can be extended to include a larger lexicon and more complex sentence patterns. In the present version, it is primarily a demonstration illustrating that Swetra grammar, when incorporating Permutational Grammar and an advanced morphological component, is fully capable of addressing two of the most tenacious problems of Basque grammar, namely the relatively free word order and the complex verbal system.

Acknowledgements

We are grateful to Itziar Laka for sharing with us her native intuitions of Basque as well as her linguistic insight. Any mistakes are ours and ours alone.

REFERENCES

Addis, Kristin. 1993. 'Paradigm trimming in Basque'. Linguistics 31 (1993), 431 - 474

Aijmer, Karin & Altenberg, Bengt (eds). 1991. English Corpus Linguistics. London. Longman

Chomsky, Noam. 1957. Syntactic Structures. The Hague. Janua Linguarum

De Rijk, Rudolf P. 1969. 'Is Basque an S.O.V. language?'. Fontes Linguæ Vasconum I-3: 319-351

Eeg-Olofsson, Mats & Sigurd, Bengt. 2000. 'Permutational Grammar for Free Word Order Languages'. Working Papers. Dept of Linguistics, Lund University. 48. 2000

Holmer, Arthur. 1999. 'An active analysis of Basque ergativity'. Fontes Linguæ Vasconum 81, 189-225. Pamplona. Gobierno de Navarra

Ortiz de Urbina, Jon. 1989. Parameters in the grammar of Basque - GB approach to Basque syntax. Dordrecht. Foris

Laka, Itziar. 1995. A brief grammar of Euskara, the Basque language. http://www.ehu.es/grammar

LOCKE, William & BOOTH, Donald. 1955. Machine Translation of Languages. New York. John Wiley & Sons.

SALTARELLI, Mario. 1988. Basque. Croom Helm Descriptive Grammar Series. New York. Croom Helm

SIGURD, Bengt (ed) 1994. Computerized grammars for analysis and machine translation. Lund. Lund University Press.

SIGURD, Bengt & GAO, Hong. 1999. 'Outline of a computerized Chinese grammar enabling English and Swedish translation'. Working Papers. Dept of Linguistics, Lund University. 47.1999:181-200

LABURPENA

Euskal itzulpenak elektronika bidez egiten hasteko lehen saio bat aurkezten du lanak. Informatika eta hizkuntzalaritza elkarrekin txertatzeko ahalegina izan nahi du. Esparru mugatua da oraingoz; aditz trinkoaren erabilpena eta hitzei dagozkien ergatiboa eta datiboko deklinabidea bakarrik entseatu dira. Ez da merkaturako prest dagoen programa, helburu batzuen behin-behineko saioa baizik. Horrek ez du esan nahi, ikerlan zabalago baten ondorioz hitzen multzo handia eta morfologia osoa horrela balia ez daitezkeenik; hizkuntzaren aldetik ez legoke eragozpenik.

RESUMEN

El trabajo presenta los primeros resultados de un ensayo de traducción electrónica del euskera. Pretende interrelacionar la electrónica y la lingüística. De momento, la labor se limita al campo reducido de la conjugación perifrástica, así como al ergativo y al dativo en la declinación de las palabras. Viene a ser un prototipo simplificado que no trata de ser un programa comercial, aunque no existen problemas de carácter lingüístico para incluir un vocabulario muy amplio y la morfología completa.

RÉSUMÉ

Le travail présente les premiers résultats d'un essai de traduction électronique de l'euskera. Il prétend présenter une relation entre l'électronique et la linguistique. Pour l'instant, le labeur se limite au domaine réduit de la conjugaison périphrastique, ainsi qu'à l'ergatif et au datif dans la déclinaison des mots. C'est un prototype simplifié qui n'essaie pas d'être un programme commercial, bien qu'il n'existe pas de problème de caractère linguistique pour inclure un vocabulaire plus étendu et la morphologie complète.

ABSTRACT

This study presents the first results from a trial of an electronic translation of Euskara. At present, the work limits itself to the syntax and to the specialised field of periphrastic conjugation and the ergative and the dative in word declensions. Although no major problems of a linguistic nature remain when it comes to including an extremely wide vocabulary and the complete morphology, it is only a simplified prototype which does not claim to be a commercial programme.