What is Alignment Syntax?

Tamás Csontos
Teacher Training Faculty, Kecskemét College, HUNGARY

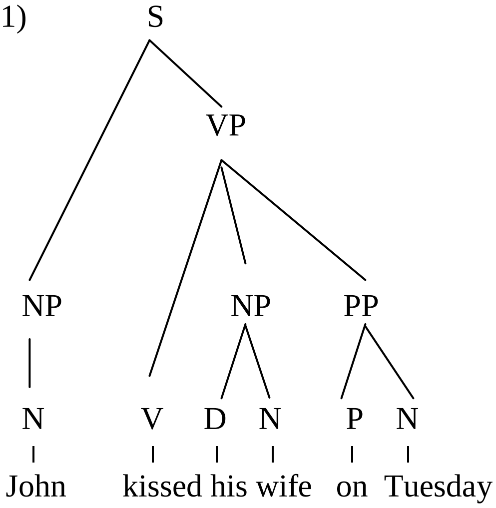
Abstract: I am going to introduce a relatively new syntactic model, called Alignment Syntax, which denies the existence of any kinds of constituents. I will start with the basic assumptions and notions, including conceptual units, domains, constraints and late lexical insertion. Then I would like to demonstrate with a concrete example how Alignment Syntax can account for certain phenomena.

Keywords: Alignment Syntax, CUs, constraints, domains, optimal candidate, late vocabulary insertion

1 Introduction

According to traditional syntactic theories (e.g. X-bar theory, the Minimalist Programme, Optimality Theory etc.), sentences are structured sequences of words and they have a hierarchical constituent structure. Words are grouped together into phrases (NPs, VPs, PPs), and phrases into sentences. Originally, it was assumed that in (1), for example, the determiner (*his*) and the noun (*wife*) constitute an NP. The verb (*kissed*) and the following NP and PP make up a VP. This VP and the subject NP, together, form the sentence. [1]

(1)



In the framework I am going to adopt (i.e. Alignment Syntax), however, it is assumed that there are no phrases, and constituents are not structured hierarchically. It is claimed that elements are ordered linearly. In the present article I will discuss the most important background assumptions and try to illustrate how Alignment Syntax works. The aim of this paper is to give a short introduction to the framework, as due to lack of space I will not be able to explore everything in depth.

---

[1] Later, during the development of X-bar theory and with the emergence other theories, most of the categories and phrases have been relabelled.

2 Basic assumptions and notions

The foundations of Alignment Syntax were laid down by Mark Newson in his article *Deforestation in syntax* in 2004. This was followed by other papers published in ELTE's *Even Yearbook* (see the reference section).

Alignment Syntax, which is a restricted Optimality Theoretic grammar, utilises a set of conflicting constraints. They evaluate the candidate set of possible expressions which is generated for every input. Then the optimal candidate will be spelled out by the best fitting vocabulary items from the lexicon. A number of questions arise: what are the input elements and the constraints and what determines which are the best fitting vocabulary items? I am going to answer these questions soon.

Input elements are taken from a universal stock of basic units, which are referred to as Conceptual Units (CUs). They are the abstract 'building blocks' that the syntax manipulates. There are two types of CUs: *roots* which have descriptive semantic content and *functional units* which carry more functional content, e.g. tense and aspect. Let us take a look at a concrete example, namely *John gave Sarah money.* The input elements for this sentence are listed in (2):

(2) √MONEYTH, √JOHNA , √SARAHG , √GIVE , [past]

John is the agent (hence √JOHNA); *money* is the theme (hence √MONEYTH) and *Sarah* is the goal (hence √SARAHG).[2] Note that these elements are not actual lexical elements/words: they are just abstract mental representations. So √MONEY means 'something you can pay with'; while √GIVE, for instance, means 'you do it when you hand something to somebody'. When lexical insertion takes place, √MONEY, for instance, is spelled out as *money* in English, *Geld* in German and *pénz* in Hungarian. In (2) all the input elements are roots, except [past], which is a functional unit. Finally, it has to be mentioned that in Alignment Syntax the input carries all the information necessary for the interpretation of expressions, so it is the input where semantic interpretation takes place.

The generator of the candidate expressions, GEN, imposes linear orderings on the input elements. These orderings constitute the candidate set. For example, if the input elements are: a, b and c, the candidate set is as in (3):

(3)

1. a b c	9. a c
2. a c b	10. c a
3. b a c	11. b c
4. b c a	12. c b
5. c a b	13, a
6. c b a	14. b
7. a b	15. c
8. b a	16. ø

---

[2] According to Newson (2013) arguments are related to event structure by specific relating elements, called relators. Arguments which are related to the first sub-event of a complex sub-event are referred to as argument 1. Thus the input element for *Sarah* would be √SARAH and the argument feature [arg1]. For demonstrative purposes, however, I do not adopt this more recent analysis.

As we can see the candidate set is limited, as we are not allowed to insert elements which are not present in the input. On the other hand, there may be input elements which are missing from the candidate set. This is possible, but in this case a faithfulness constraint is violated (introduced below). This is not problematic, however, as, in Optimality Theory, low ranked constraints may be violated if this allows higher ranked ones to be satisfied.

Before turning to the constraints, we have to discuss *domains*, as they play an important role in our model. Domains are sets of elements which share some properties determined by the input. For example, arguments which are related to a specific root predicate can constitute a domain, called the argument domain (**DA**). √MONEYTH , √JOHNA and √SARAHG are members of **DA**. A domain is not necessarily a contiguous string, because there may be other elements between its members, e.g. the verb in English.

In Alignment Syntax we distinguish between 2 types of constraints: faithfulness and alignment constraints. As we have said, they evaluate the candidate set. Faithfulness constraints are violated if an element which is part of the input is missing from the output. In (3) above, for example, candidate (9) violates one faithfulness constraint, while (13) violates two.

The alignment constraints can align two single elements to each other. The possible alignment constraints can be seen in (4):

(4)

aPb ‘ a precedes b’	violated by b...a order
aFb ‘ a follows b’	violated by a...b order
aAb ‘a adjacent to b’	violated by every CU which intercedes between a and b

Let us assume that the input elements are *a, b* and *c* and let us further assume that the constraints are aPb, aPc, aAb and aAc.

(5)

	aPb	aPc	aAb	aAc
b a c	*(!)			
b c a	* (!)	*	*	
→ a b c				*
a c b			* (!)	
c b a	* (!)	*		*
c a b	* (!)	*		

The table shows that the third candidate is the winner although it does violate a constraint. The second candidate, for instance, is out, because it violates the highest ranked constraint, as *a* does not precede *b;* it also violates the second constraint, as *a* does not precede *c* and it also violates the lowest ranked constraint, as *a* is not adjacent to *b.*

The alignment constraint can align a single element to a domain as well:

(6)

aPDy ’ a precedes domain y’	violated by every member of domain y which precedes a
aFDy ’ a follows domain y’	violated by every member of domain y which follows a
aADy ’ a is adjacent to domain y’	violated if a does not appear at the edges of domain y

If the members of D_x are x, a, b and c and the constraints are aPD_x, xPb, bPc and bAc, we can see that the second candidate will be the optimal candidate. For example, the fourth one is out, because there are three members of the domain which precede *a,* so the highest ranked constraint is violated three times.

(7)

	aPD_x	xPb	bPc	bAc
x a b c	*(!)			
→ a x b c				
x a c b	*(!)		*	
x b c a	***(!)			
x c a b	**(!)		*	*
a b x c		*(!)		*
a b c x		*(!)		

The constraints aPb, aFb and aADx are non-gradient, because they cannot be violated gradually: they are either violated or not. The others are gradient constraints; this means that they can be violated to a different extent. In addition to the above, there are also anti-alignment constraints with respect to a domain. For example, $a{*}PD_x$ says that *a* cannot precede domain x. Similarly, the constraint a*FDx requires that *a* cannot follow domain x. Consequently, the former anti-alignment constraint is violated if *a* precedes all the members of domain x. If we combine an anti-precedence and a precedence constraint, this interaction can yield the so called second position phenomenon. Let us assume the members of Dx are *a, b* and *c*.

(8)

	$a{*}PD_x$	aPD_x
a b c	*(!)	
→ *b a c*		*
c a b		**(!)

The first candidate violates **$a{*}PD_x$**, as *a* does precede all the members of the domain, the third candidate violates the second constraint twice, because two elements of the domain precede *a*, while the second candidate violates it only once.

The assumptions that we have made can be summarized in the following diagram:

(9) ***input → GEN → candidate set → constraints → optimal candidate***

When we have the optimal candidate, the conceptual units will be spelled out by phonological exponents from the vocabulary containing lexical entries: phonological forms and the features associated with them. For example, √GIVE [past] will be spelled out as *gave,* because the lexical entry for GIVE [past] is *gave*: √GIVE [past]↔ gave. *Gave* is the phonological form and √GIVE and [past] are the associated features. These data are stored in the vocabulary.

There are four principles which determine what can spell out a given string of conceptual units if there is no exact match between that string and the vocabulary item. The first principle is called the Superset Principle, which says that the best fitting match for a sequence of features is that vocabulary item which is associated with all the features which can be found in that sequence. It is not a problem if that vocabulary item contains other features as well which cannot be found in the sequence. For instance, if the sequence to be spelled out is <a,b,c> and the vocabulary items that can possibly spell it out associated with features <a,b>, <a,b,c,d> <a,b,d>, the best fitting match will be <a,b,c,d> despite the fact that it is associated with an extra <d> feature: it contains all the features of the sequence <a,b,c> .

Secondly, it is a basic condition that only contiguous sequences can be spelled out by a single vocabulary item. In addition, it is also assumed that vocabulary insertion is ‘root centric’, which means that the process starts with the roots, spelling these out with those contiguous functional units that the vocabulary entry allows for. Remaining functional units are spelled out separately.

The last principle is the principle of Minimal Vocabulary Access. It says that if you can spell out a sequence of features with one vocabulary item instead of two, you have to spell it out with one item and not with two. If we assume the following lexical entries for the items below, we will see why √GIVE [past] will be spelled out as *gave*, and not as **gived.*

(10)

√GIVE ↔ give
√GIVE [past] ↔ gave

[past] ↔ ed
√LIVE ↔ live

According to the principle of Minimal Vocabulary Access, it is better to spell out the sequence √GIVE [past] with one item (i.e. *gave*). This, however, is not possible with the sequence √LIVE [past], as there is no single item in the lexicon which is associated with both of these CUs. Therefore, √LIVE and [past] will be spelled out separately as *live* and *ed* (i.e. by two vocabulary items) yielding the form *lived.*

3 Alignment Syntax in action

In this section I demonstrate how certain phenomena (e.g. the order of arguments and the verb-second phenomenon in English) can be explained in the light of Alignment Syntax. Let our example be the sentence *John gave Sarah money.* As I have already claimed, the input elements are √MONEYTH , √JOHNA , √SARAHG , [past] , √GIVE.

First of all, we have to account for the order of arguments in English, which is agent>goal>theme. *Sarah gave John money* does not mean the same as the example above, because *Sarah* in *Sarah gave John money* cannot be interpreted as the goal, because in English goals cannot precede agents. The agent, the goal and the theme constitute the *argument domain* (DA). The correct order can be achieved if we postulate the following constraints:

(11) [agent]PDA >[3] [goal]PDA > [theme]PDA.

Secondly, the observation is that the verb must follow the first argument in a sentence and thus occupy the second position. The sentence *John Sarah gave money* is ungrammatical, because the verb does not follow the first argument. The constraints which are responsible for this phenomenon are:

---

[3] The symbol ’>’ means that the constraint which precedes it is ranked higher than the one which follows it.

(12) √*PDA > √PDA

The first of these constraints requires that the verb should not precede all the members of DA (an anti-alignment constraint), while the second says that it must precede DA. These conflicting constraints guarantee that the verb will take the second position in the sentence (see also (8) above).

Lastly, the past tense morpheme must immediately follow the verb, which is the reason why * *John ed play with Bill* and * *John play with ed Bill* are unacceptable. The relevant constraints are:

(13) [tense]F√ > [tense]A√ [4]

As table (14) illustrates, the constraints introduced in (11), (12) and (13) and their order yield the desired result:

(14)

	√*PDA	√PDA	[a]PDA	[g]PDA	[th]PDA	[tense]F√	[tense]A√
√GIVE[past] √MONEYTH √JOHNA √SARAHG	*(!)		*	* *			
√MONEYTH √GIVE [past] √JOHNA √SARAHG		*	*(!)	* *			
√JOHNA √SARAHG √GIVE [past] √MONEYTH		**(!)		*	**		
√JOHNA √SARAHG √MONEYTH√GIVE [past]		***(!)		*	**		
√JOHNA [past] √GIVE √MONEYTH √SARAHG		*		* *(!)	*	*	
√JOHNA √GIVE √MONEYTH [past] √SARAHG		*		* *(!)	*		*
→√JOHNA√GIVE[past] √SARAHG √MONEYTH		*		*	**		

Finally, the following vocabulary items spell out the CUs of the winning candidate, respecting the principle of Minimal Vocabulary Access:

(15)

√JOHNA	√GIVE [past]	√SARAHG	√MONEYTH
John	*gave*	*Sarah*	*money*

4 Conclusion

In my article I tried to illustrate how the present theory can explain basic word order in English utilising only alignment constraints. Admittedly, I have been scratching only at the surface. There are much more complex phenomena that have been already accounted for within this framework (e.g. the use of dummy auxiliaries in English – Newson and Szécsényi (2012)) but there are many other issues that need to be addressed in the future as well.

---

[4] I am simplifying the analysis here for demonstrative purposes again. Newson (2010) takes not only tense but also aspect CUs into consideration and thus gives a more complex but a more general explanation for the order of these elements.

References

[1] Newson, Mark: Deforestation in syntax, In: László Varga (ed.). The Even Yearbook 6. Eötvös Loránd University, 2004. (Available at: http://seas3.elte.hu/delg/publications/even/2004.html#ne)

[2] Newson, Mark: Syntax First, Words After: A possible consequence of doing Alignment Syntax without a lexicon, In: László Varga (ed.). The Even Yearbook 9. Eötvös Loránd University, 2010. (Available at: http://seas3.elte.hu/delg/publications/even/2010.html#nm.)

[3] Newson, Mark and Szécsényi, Krisztina: Dummy Auxiliaries and Late Vocabulary Insertion, In: László Varga (ed.). The Even Yearbook 10. Eötvös Loránd University, 2012. (Available at: http://seas3.elte.hu/delg/publications/even/2012.html#nm.)

[4] Newson, Mark: What you can do with a domain and why they are not phrases, In: Péter Szigetvári (ed.). *VLlxxx*. Eötvös Loránd University, 2013. (Available at: http://seas3.elte.hu/tmp/vlfs/newson.html)

Author data

Tamás Csontos: Teacher Training Faculty, Kecskemét College
6000 Kecskemét, Kaszap u. 6-14. E-mail: csontostamas82@freemail.hu