Restrictive relative clauses in Alignment Syntax[1]

Márton Kucsera

Introduction

This paper examines the status of English restrictive relative clauses within the theory of Syntax First Alignment (SFA) developed by Newson (2000). This a more restricted version of standard Optimality Theory (OT). In SFA, all syntactic phenomena are reduced to the interaction of alignment and faithfulness constraints, and syntactic structure is eradicated from the theory. The aim of this article is to show that there are a number of descriptive facts about relative clauses that fall out from an SFA account without any further stipulation. Here, an analysis is introduced that does not have to rely on the Doubly Filled COMP Filter (introduced in Chomsky and Lasnik, 1977) to ban the co-occurrence of *that* and a *wh*-element, and also excludes subject extraction from zero relatives (as in (1-c)) wihout any further assumption.

(1) a. the man who lives next door
b. the man that lives next door
c. *the man lives next door

In the first section, it is discussed to what extent the basic assumptions about the input in SFA differ from those in standard OT. In particular, it is shown that certain OT analyses, such as the analysis of relative clauses by Keer and Baković (1997) are unavailable in an SFA system because of these differences. Then, the second section introduces the input hypothesized for restrictive relatives. The third section presents the relevant constraints, and the fourth section details in what ways these can interact. Finally, the fifth section describes the rankings that can account for the English data.

[1]This article is a shortened version of the MA thesis entitled "English restrictive relative clauses in Optimality Theory" which was submitted in April, 2014. There are a number of people without whom writing this thesis could not have been possible. I would like to thank Marianna Hordós, my advisor, whose comments and insights helped this work a great deal. I am no less grateful to Mark Newson, who refereed my thesis, for his review and pointing out the directions in which the analysis can be furthered. Also, I want to thank the members of the TDK Research Group who listened to my presentation of this work for their questions and comments. Needless to say, any error that might remain in the text is mine.

1 Relative clauses in standard Optimality Theory

1.1 The input in Optimality Theory

In standard OT, it is customary to view the input as a set of lexical items (Kager, 1999). Thus, the input does not contain any reference to syntactic structure. Following Grimshaw's model (1997), Kager gives the contents of the input for syntactic structures in (2).

(2) *Specifications in the input:*

a. lexical heads and their argument structure
b. assignment of arguments
c. tense and semantically meaningful auxiliaries

(Kager, 1999)

There is however significant variation among different authors as far as the nature of the input is concerned. While for Grimshaw the input seems to be a dependency relation of lexical items and certain features, others may include more or less information and structure in the input. Legendre (2001), to all intents and purposes, admits that there is little agreement on the nature of the input, and it is up to individual authors to decide whether or not to include, for instance, *wh*-scope. On the other hand, there are certain works that assume that even more is part of the input than that. For example, Broihier (1995) follows Pesetsky in claiming that the input is an s-structure or an LF.[2] Here, the framework devised by Grimshaw will be discussed, as not only is that kind of input the most common (Legendre, 2001), but it does not make much sense to examine cases where the input is already a structure in itself while the paper works within a framework that does not recognise any kind of syntactic structure.

This "standard" type of input places semantic interpretation after evaluation. In other words, a winning candidate is selected first; then, it is interpreted. This means that the candidates generated from the same input might actually have different meanings; that is, if a feature of the winner is deleted by GEN, that feature will not be interpreted. Two problems are pointed out in connection with that approach. First, it requires semantic information to be accessed repeatedly, which can be seen as a problem for the economy of the grammar. Second, it could be argued that interpretation after selecting a winner is not a feature of OT syntax itself, but it

[2]Although this issue is not in the scope of the present paper, such an approach is presumably an attempt to make the implicit constraint interactions in GB or Minimalism explicit. As Grimshaw (1998) points out, referring to "last resort" or "default" is actually bringing constraint interaction into the system. Until such instances or constraint interaction are not formalised in some way, they cannot be integrated into any theory (e.g. one cannot know whether constraint domination is transitive.)

is used to conform to other theories that the analysis incorporates.

First, part of the semantic interpretation can arguably be done on the input. For instance, the meaning of the lexical items, relations, and features listed in (2) can be read off the input. However, there are a number of constraints that are grounded in semantics; therefore, part of the meaning cannot not come from the input, but, in EVAL, there exist constraints that refer to the semantic interpretation of certain structures. An example for that is Grimshaw's (1997) OPSPEC constraint which requires operators to always occupy a Specifier position. The reasoning goes that operators must take scope over their domain, and that is only possible if they are in a Specifier position. Thus, the interpretation of scope relations clearly does not take place before the evaluation. In other words, it must be the winning candidate that gets interpreted.

A more intriguing picture emerges if one examines the FULLINT constraint. This constraint assigns a violation mark for each element in the structure whose semantic content is ignored;[3] i.e., it does not contribute to the meaning of the sentence (e.g. dummy auxiliaries and expletive subjects in English). These "semantically empty" elements are always inserted by GEN as the input only contains semantically relevant lexical items. Notice, however, that this assumption can also significantly alter the way EVAL is conceptualised.

The question that could arise in connection with FULLINT is whether EVAL is sensitive to semantic content. When calculating the violations for FULLINT there must obviously be a way to check whether or not a given element has any semantic content. Therefore EVAL must be able to access semantic specifications, too. As far as economy is concerned this model does not fare particularly well since semantic information (or the lexicon) is accessed at least three times for every single input.

One might perhaps try to escape this problem by claiming that what EVAL has access to is only a set of possible dummy elements, so it cannot actually access the lexicon. This argument can be refuted, however, on the basis of the fact that every dummy element has a counterpart that is a full-fledged lexical item, so EVAL still has to differentiate between those two. Another alternative would be to claim that FULLINT is actually similar to the DEPIO constraint developed for phonology by Prince and Smolensky(1993/2002). In their framework, DEPIO is a faithfulness constraint that assures that no input elements are deleted. Basically, it checks that every element in the input has a corresponding element in the output. This explanation would only lead to a further problem, though: it would require every tense, aspect and modality specification to be expressed in the form of a lexical item in the input; otherwise there would be a violation of this constraint.

[3]Grimshaw (1998) explains that this constraint is basically a faithfulness constraint as semantic content can be understood as a bundle of features associated with an element in the input; hence, it can be underparsed.

Second, Newson (2014) points out that placing semantic interpretation after the selection of the optimal candidate might also be motivated by conformity to other theories. As McCarthy(2008) explains, OT is a fairly general theory. In fact, it is not even a theory in its own right, but only a decision-making mechanism. Therefore, an OT analysis must have some other theory about language, in which the constraints of the particular analysis are grounded. As Grimshaw's paper is written in the spirit of Government and Binding Theory, it is not surprising that her assumptions about the place of semantic interpretation are compatible with GB, where Logical Form (the structure from which meaning is read off) is also derived after syntactic processes have taken place.

1.2 The analysis by Keer and Baković(1997)

As is shown above, the notion of the input is not always clear-cut in standard OT, nor is it the only source feeding semantic interpretation. Therefore, it is possible to argue that the three types of English restrictive relatives, while their meaning is the same, are related to three different kinds of input. In fact, this is the assumption that the analysis of Keer and Baković (1997) makes.

Their line of argumentation is as follows. If there is an operator (*wh* or empty), GEN builds a CP projection over the IP of the clause to host it. The structure of the CP is regulated by the following three constraints in (3).

(3) *Constraints on CP-structure*:

a. SPECLFT: The specifier is on the left edge of a phrase
b. HDLFT: The head is on the left edge of a phrase
c. OBHD: Phrases have heads

If the ranking of these three is SPECLFT >> HDLFT >> OBHD, then they basically achieve what the Doubly Filled COMP Filter of Chomsky and Lasnik (1977) does. This is shown in detail in the tableaux below.

(4) Three types of relative clauses

	Input: *wh*-operator	SPECLFT	HDLFT	OBHD
☞	[$_{CP}$ wh_i [$_{IP}$ … V t_i …]]			*
	[$_{CP}$ wh_i that [$_{IP}$ … V t_i …]]		*!	
	[$_{CP}$ that wh_i [$_{IP}$ … V t_i …]]	*!		

	Input: empty operator	SPECLFT	HDLFT	OBHD
	[$_{CP}$ op_i [$_{IP}$ . . . V t_i . . .]]			*!
☞	[$_{CP}$ op_i that [$_{IP}$ . . . V t_i . . .]]			

	Input: no operator	SPECLFT	HDLFT	OBHD
☞	[$_{IP}$ . . . V t_i . . .]			
☞	[$_{CP}$ that [$_{IP}$ . . . V t_i . . .]]			

(Keer and Baković, 1997:6)

As the tableaux show, the Doubly Filled COMP Filter falls out from the analysis as the result of the interaction between the three constraints. Since an overt specifier and an overt head violate either SPECLFT or HDLFT, and since OBHD is ranked below both, the latter will be violated. This is not the case, however, if the operator has no phonological content, then, apparently, both SPECLFT and HDLFT are satisfied.

In addition, the analysis needs a faithfulness constraint to select a winner when there is no operator. Therefore a fourth constraint is introduced: FAITHOP, which penalises each operator that is not included in the input. As a result, a clause introduced by *that* emerges as a winner in the case when there is no operator. Thus, the final ranking is shown in (5).

(5) SPECLFT >>{HDLFT, FATIHOP} >>OBHD

While this analysis is straightforward, it comes with a certain number of problems. First, it is not at all clear what necessitates the three different kinds of input. In particular, the existence of an input with no operator seems quite troubling. It seems to raise the question why operators are included in the input if the system can work without them. That is, the grammar that sometimes includes an item in the input that is not necessary to build a grammatical structure is problematic as far as economy is concerned. In addition, there is a further problem because of which such an analysis cannot be adopted in the framework of Alignment Syntax. If we assume that meaning is read off the input, then three constructions with the same meaning cannot be assigned three different inputs. Therefore, as long as one adheres to the assumptions formulated by Newson (2004); i.e. that the input is what feeds the semantic component of the grammar, any account that relies on different inputs to explain variation is out of question for the analysis .

An additional problem with the analysis surfaces when one attempts to account for a phenomenon which involves a subset of the data and seems to exhibit additional variation: pied-piping and preposition stranding. The data in itself is not too complex. In the case of *wh*-relatives, when the operator is also the complement of a preposition, the preposition itself may be situated at the left edge of the clause along with the operator (pied-piping), or it may stay behind (preposition stranding). With the other two types of relatives, only preposition stranding is possible. This is summarised in (6).

(6) a. *wh*-relatives:
  (i) the film which the children talked about
  (ii) the film about which the children talked
 b. *that*-relatives:
  (i) the film that the children talked about
  (ii) *the film about that the children talked
 c. zero relatives:
  (i) the film the children talked about
  (ii) *the film about the children talked

If the assumption that variation stems from the input is adopted it means that the class of *wh*-operators must be split into two. There must be a different element in the input for the “stranded” and for the “pied-piped” operator. Also, notice that, in the case of pied-piping, the constraint OP-SPEC is violated as in this case the operator occupies a complement position as is shown in (7).

(7) $[_{CP}$ $[_{PP}$ $[_{P'}$ about $[_{NP}$ which]]] $[_{C'}$ $\emptyset$ $[_{IP}$ the children talked t]]]

Thus, to solve this problem further constraints have to be devised. As a result, each case of variation increases the set of possible input elements and might result in further constraints added to the system.

2 The input for relative clauses in Alignment Syntax

While the analysis discussed above can function in a system where the input is not the sole input of semantic interpretation, an SFA account must make drastically different assumptions concerning the input for two reasons. First, SFA is based on the idea that it is only the input that receives interpretation; thus, anything that the generator does to it is invisible to the semantic component of the grammar (Newson, 2004). Therefore, the assumption that different types of relatives have different inputs is untenable since their identical meaning proves that they are interpreted similarly. Secondly, the SFA grammar does not manipulate lexical items, but abstract conceptual units (CU) and lexical insertion takes place after a winning candidate is selected. Thus, this section explores what kinds of functional CUs are part of the input for restrictive relative clauses.

Here, it is argued that there is a functional CU that is responsible for the interpretation of relatives as restrictive modifiers. This CU is named |restr|. While this paper does not investigate the nature of that CU in depth, it seems reasonable to assume that it has something in common with determiners primarily because a number of languages use actual determiners to introduce relative clauses (in fact, *that* in English can also appear as a determiner). Consider the example from Kalaallisut (West Greenlandic) in (8). There, the relative clause is introduced by *tassani* which is a determiner with a locative suffix.

(8) ikerasanngua.mut ilaga.akka tassa.ni naapipp.akkit
little-sound.ALL/S accompany.IND/1S/3P this.LOC/S meet.IND/1S/2S
"I accompanied them to the little sound at which I met you"

(Sadock, 2003: 28)

Similarly, it is argued that there is a single relative operator, $|\text{op}_{\text{rel}}|$, which is present in the input of every relative clause. In this section, the details of such an input are worked out and it is shown how it can be spelled out in two different ways: as *that*, as a *wh*-element, or not spelled out at all.

To begin with, it is clear that both the complementiser and the operator must be included in the input since GEN cannot insert anything into the candidate forms[4]. Therefore, both of them must be present in the form of functional CUs. Accordingly, I assume that the input for the relative clauses in (9) consists of the elements shown in (10).

(9) a. the book that Bill read
b. the book which Bill read
c. the book Bill read

(10) $\sqrt{BOOK}$
$|\text{restr}|_i$
$\sqrt{READ}$ (|agent|, |theme|)
$|\text{agent}| = \sqrt{BILL}$
$|\text{theme}| = |\text{op}_{\text{rel}}|_i(\sqrt{BOOK})$

[4]In Alignment Syntax, the capabilities of GEN are limited to ordering and deleting elements. Hence, the problem of dealing with infinitely large candidate sets due to insertion can be avoided (Newson, 2004)

The example shows that the input consists of two kinds of units: root (marked with $\sqrt{}$) and functional CUs (marked with ||). Roots are more closely associated with "content words" (although they are not specified for category). Functional CUs, on the other hand, express grammatical relations. They appear in a dependency structure, which is shown by the indentation in the example. As can be seen in (10), it is proposed that in the input |restr| appears as a dependent of the root element it modifies. This way, it can be assured that clausal modifiers are treated differently from other modifiers (e.g. adjectives), which appear as root CUs in the input, not as functional ones. Also, |restr| always has a root CU as dependent; that root will be realised as a verb (the predicate of the relative clause). In this configuration, it is very easy to define what a relative domain consists of: it contains |restr| and everything depending on it (with dependency being treated as a transitive relation here; i.e. if *a* depends on *b*, and *b* depends on *c*, then *a* depends on *c*, as well).

Also, it is argued that the relative operator is a function-like element that takes the CU the |restr| is dependent on as its argument and basically returns a reference to it. Thus, from the point of view of the verb, the operator behaves like a regular argument. Most importantly, the constraints that regulate the position of the verb and its arguments remain sensitive to the position of $|op_{rel}|$. It must be emphasized that every $|op_{rel}|$ CU must be related to a |restr| (that is shown by the matching indices in (10)), and maximally one $|op_{rel}|$ might belong to a |restr|.[5] Otherwise, $|op_{rel}|$ would become far too general a tool in the grammar; for instance, if its appearance would not be constrained it could easily be used to replace pronouns.

[5]While Hungarian seems to have relative clauses with multiple operators (Lipták, 2000), that is not necessarily a counterexample as in those Hungarian relatives the relative pronouns are interchangeable with interrogative ones.

(11) aki amit talált, megette
REL-P.NOM REL-P.ACC find.PAST-3S eat.PAST-3S
"everyone ate whatever he found"

Thus the above example is the same as the one below.

(12) ki mit talált, megette
WHO.NOM WHAT.ACC find.PAST-3S eat.PAST-3S

Normally, relative pronouns cannot be replaced with their interrogative counterparts.

(13) megették, amit találtak
eat.PAST-3P REL-P.ACC find.PAST.3P
"they ate what they found"

(14) *megették, mit találtak
eat.PAST-3P WHAT.ACC find.PAST.3P

Thus, this case seems more complex as there are probably different elements in the input as well.

Furthermore, the operator in relative clauses must be different from the operator in questions for three main reasons. First, the operators in questions do not show variation (i.e. they always surface as *wh*-elements). Second, interrogative operators do not get reference from another element in the sentence. Third, as a consequence of the one to one relationship between |restr| and $|\text{op}_{\text{rel}}|$, relative clauses cannot have multiple operators while interrogatives certainly can.[6]

Having discussed the nature of the two functional CUs which are typical of relative clauses, what still remains is the question of lexical insertion. As this section has already established that the input always contains the same elements the difference between the three types of relative clause must come down to inserting different lexical items corresponding to |restr| and $|\text{op}_{\text{rel}}|$. In this paper, it is argued that there are three different outcomes of lexical insertion depending in the configuration of these two CUs.[7]

As a result, it is the order of these two elements that yield the three types of relative clauses: *wh*-, *that*-, and zero relatives. First, let us discuss the insertion of *wh*-elements. Here, the analysis hinges on the assumption that English *wh*-elements incorporate both the $|\text{op}_{\text{rel}}|$ and |restr| CUs. Thus, in the light of the minimal vocabulary access principle[8] , a *wh*-element is always inserted to replace a |restr| $|\text{op}_{\text{rel}}|$ sequence. A further assumption is that when not next ot |restr|, $|\text{op}_{\text{rel}}|$ itself is specified as empty (that is, lacking phonetic content). A single |restr| element, in turn is replaced wit *that*.

By adopting this approach, the grammar outlined in this paper accounts for the effects of the Doubly Filled COMP Filter without any further stipulation. Considering that every adjacent |restr|, $|\text{op}_{\text{rel}}|$sequence is replaced with a *wh*-element, there obviously cannot exist a *wh*, *that* sequence as there is no remaining CU which could be replaced by *that*. Thus, the way to account for the difference between languages with and without the DFCF is to assume that different lexical entries belong to *wh*-elements in those languages. In languages, where the DFCF apparently does not hold, such as Alemanic (cf. Bayer and Brandner, 2008), *wh*-words are inserted to replace $|\text{op}_{\text{rel}}|$ only leaving |restr| in the sequence which can, in turn, be replaced with a complementiser.

[6]A further argument for this position might come from cross-linguistic data. If one claims that CUs are universal (as, for instance, Newson (2010) does), then the two kinds of operators must be differentiated as they are realised as different lexical items in several languages. As an example, consider Hungarian, where the interrogative operator *ki* (who.NOM) has its relative counterpart *aki*.

[7]Naturally, if deletion is allowed (as this paper assumes it is), one would expect five possible orderings of the elements A *A* and *B*: *AB*, *BA*, *A*, *B*, and ∅. The reason for this apparent lack of two of the possibilities is only outlined here and will be elaborated on in the following sections.

[8]Newson (2010) argues that, during vocabulary insertion, a single vocabulary item that can spell out a larger number of CUs is favored over multiple words that spell out the CUs in question individually

Based on the above, it follows that *that* is inserted only when there is no $|\text{op}_{\text{rel}}|$ adjacent to the |restr| CU. This can happen in two cases: (i) $|\text{op}_{\text{rel}}|$ and |restr| are separated by other elements, or (ii) $|\text{op}_{\text{rel}}|$ was deleted in the winner.

Finally, zero-relatives can be the result of, again, two possible configurations: (i) |restr| is deleted in the output, (ii) both |restr| and $|\text{op}_{\text{rel}}|$ are deleted in the output. Thus, we can see that although there are more possible configurations of the two CUs (if deletion is also a possibility), but, in the end, these amount only to three different cases of lexical insertion as some of the individual configurations lead to the same result. This approach could account for the fact that *wh*-relatives pattern with *that*-relatives with respect to dislocation. The cases where |restr|is preserved in the winner can be located further away from the root which they refer while without the |restr| CU the relative clause must be adjacent to the noun it modifies.

(15) a. I saw the child yesterday [who Mary adores].
b. I saw the child yesterday [that Mary adores].
c. *I saw the child yesterday [Mary adores].

To summarise, this section argues that the input for relative clauses contains two special functional CUs: $|\text{op}_{\text{rel}}|$ and |restr|. While the former is responsible for appearing as an element co-referent with the head noun within the relative clause, the latter provides restrictive interpretation for the clause.

3 The relevant constraints for English restrictive relatives

First, as the present paper is concerned with relative clauses, the constraints that regulate the respective positions of the verb and its arguments appear in a very simplified form. With adapting the notion of Late Lexical Insertion, constraints that refer to categories or word classes should not be used at all as syntax only manipulates conceptual units. Therefore, the constraints used in earlier works in Alignment Syntax, such as s*P*v (the subject precedes the verb) or arg*F*v (arguments follow the verb), are rather difficult if not impossible to interpret in a system where syntax operates before lexical insertion.

Throughout the analysis, however, the above two constraints are going to be used for the sake of simplicity. Normally, one would have to define the argument domain, a ranking among the arguments and domain constraints that place the (verbal) root element in the argument domain. This would require minimally the constraints shown in (16).

(16) $\sqrt{\ }{*}P\text{D}_{\text{ARG}} >> \sqrt{\ }P\text{D}_{\text{ARG}} >> |\text{agent}|P\text{D}_{\text{ARG}} >> |\text{goal}|P\text{D}_{\text{ARG}} >> |\text{theme}|P\text{D}_{\text{ARG}}$

If one considers how the ranking in (16) selects a winner with the input being a transitive verb and its two arguments (e.g. *Jim hit Bill*), it is easy to see that this ranking works just like the ranking s*P*v >> arg*F*v. The tableaux comparing the two are given in (17) and (18).

(17)

	Input: $\sqrt{HIT}$, |ag|, |th|	$\sqrt{}$*$P\mathrm{D}_{\mathrm{ARG}}$	$\sqrt{}P\mathrm{D}_{\mathrm{ARG}}$	|ag|$P\mathrm{D}_{\mathrm{ARG}}$	|th|$P\mathrm{D}_{\mathrm{ARG}}$
	$\sqrt{HIT}$ |ag| |th|	*!		*	**
	|ag| |th| $\sqrt{HIT}$		**!		*
☞	|ag| $\sqrt{HIT}$ |th|		*		**
	|th| $\sqrt{HIT}$ |ag|		*	**!	

Here, the highest ranking constraint requires the root to be second in the argument domain (i.e. that exactly one member of the domain precede it). At the same time, members of the argument domain want to precede the domain, and there is a constant hierarchy among them; thus, the highest argument in the hierarchy precedes the root while the others follow it.

(18)

	Input: hit (Jim, Bill)	s*P*v	arg*F*v
	hit Jim Bill	*!	
	Jim Bill hit		**!
☞	Jim hit Bill		*
	Bill hit Jim	*!	*

Therefore, while this paper assumes that the ranking used in (16) and in the above tableaux is, in fact, correct, the two constraints of the latter are going to be used as a shorthand for those for the sake of simplicity. Thus, in the following chapter, only s*P*v and arg*F*v are going to be used.

Having discussed these notational issues, the constraints that this paper assumes to be specific to relative clauses are introduced. First, two alignment constraints are discussed; then, two faithfulness constraints are shown.

The alignment constraints that are relevant to relative clauses only are the ones that regulate the position of the two functional Conceptual Units introduced previously: |restr|and $|\mathrm{op}_{\mathrm{rel}}|$. Also, it was argued that all the elements that depend on |restr|in the input are in a domain called the relative domain $\mathrm{D}_{\mathrm{REL}}$.

The first alignment constraint this paper operates with prefers |restr|to be situated at the left edge of the relative clause. This can be achieved by positing the following constraint.

(19) $|\text{restr}|P\text{D}_{\text{REL}}$: assign one violation mark for each member of D_{REL} that precedes |restr|

There is, at least, another constraint that must be posited to account for the emergence of the |restr| $|\text{op}_{\text{rel}}|$ sequences that can emerge as *wh*-elements. Here, there are two possible solutions. One could argue that there is both a precedence- and an adjacency type constraint in this case that can be formulated as follows.

(20) a. $|\text{op}_{\text{rel}}|F|\text{restr}|$: $|\text{op}_{\text{rel}}|$ follows |restr|
b. $|\text{op}_{\text{rel}}|A|\text{restr}|$: assign one violation mark for each element between $|\text{op}_{\text{rel}}|$ and |restr|

In this case, however, (20-a) must always be ranked above (20-b) to avoid side-switching as lexical insertion is, in fact, sensitive to the order of CUs (cf. Newson & Szécsényi's (2012) analysis of English dummy auxiliaries). Thus, it seems an easier solution – if some ranking must be posited – to rank (19) above (20-a). Notice that this ranking necessarily involves obeying (20-a) as the tableau below shows. Thus, the analysis is going to be worked out without taking $|\text{op}_{\text{rel}}|F|\text{restr}|$ into account. This, however, does not mean that this constraint does not exist, only that with the other two being active, its effect cannot be detected.

(21)

		$|\text{restr}|P\text{D}_{\text{REL}}$	$|\text{op}_{\text{rel}}|A|\text{restr}|$
	$|\text{op}_{\text{rel}}||\text{restr}|$	*!	
☞	$|\text{restr}||\text{op}_{\text{rel}}|$		
	$|\text{restr}|$ x $|\text{op}_{\text{rel}}|$		*!

Although it seems an equally good solution to define the position of $|\text{op}_{\text{rel}}|$ not with respect to |restr| but instead to D_{REL}, it is argued that this is not the case. Admittedly a result the same as (21) would be produced by ranking $|\text{restr}|P\text{D}_{\text{REL}}$ above a – hypothetical – $|\text{op}_{\text{rel}}|P\text{D}_{\text{REL}}$ constraint. It would, naturally, mean little problem that, in that case, the operator would always violate that constraint, but a more serious problem emerges with this approach when one examines the phenomenon of pied-piping as will be demonstrated in section 5.1. Therefore, the two alignment constraints that are assumed to be active in the case of relative clauses are $|\text{restr}|P\text{D}_{\text{REL}}$ and $|\text{op}_{\text{rel}}|A|\text{restr}|$.

After discussing the alignment constraints, the necessary faithfulness constraints are discussed. This matter is fairly simple because of the way GEN creates candidate sets. If GEN can delete anything from the input, then there must be a faithfulness constraint against the deletion of each of these elements. In particular, a faithfulness constraint must be posited that prohibits the deletion of both of the assumed functional CUs. Accordingly, the analysis is going to work under the assumption that there is a PARSE(|restr|) and a PARSE(|op_{rel}|) constraint. In addition, there is another constraint of the PARSE-type that needs to be mentioned although it is not, strictly speaking, specific to relative clauses. Recall that GEN is allowed to delete elements and features from the input, and domain membership is seen as an index (basically, a feature) specified in the input. Therefore, there must be a constraint that works towards all the domain indexes surfacing in the output. This constraint is PARSE(di) (standing for domain index).

To summarise, this section has proposed a number of constraints for the analysis of relative clauses. Apart from s*P*v and arg*F*v – which are going to be used instead of the constraints referring to the argument domain – five constraints were proposed, and these are listed together in (22).

(22)
a. |restr|*P*D_{REL}
b. |op_{rel}|*A*|restr|
c. PARSE(|restr|)
d. PARSE(|op_{rel}|)
e. PARSE(di)

The first two are alignment constraints. (22-a) requires the |restr| element to precede every member of D_{REL} while (22-b) prefers |op_{rel}| to be adjacent to |restr|. The last three are faithfulness constraints which penalise the deletion of |restr|, |op_{rel}|, and domain indexes, respectively. Their interactions in the different kinds of restrictive relatives are discussed in 5.

4 Conflicting constraints and resolution strategies

Every Optimality Theoretic analyses must rely on the interaction of different constraints, and the observed grammatical sentences are viewed as the result of that interaction. Before one can move on to coming up with actual rankings to account for any given phenomenon it must be seen clearly in what ways the proposed constraints can possibly interact. In addition, OT constraints are argued to be universal; that is, only their ranking differs across languages. Therefore, it must also be shown that the interactions of the set of constraints that the analysis relies on are able to account for data from other languages. Doing so is what this section aims at. First it is discussed what kind of output forms can result from the interaction of the constraint set introduced in section 3. Then, a brief cross-linguistic survey of relative clauses follows. In that, relative clauses from Hungarian, Mandarin, Japanese, and Tagalog are examined, and it is argued that the proposed system can account for all the data presented. The analyses proposed here for these four languages are hardly more than sketches, nevertheless, and they must be subject to extensive further research.

First, notice that the constraints proposed earlier and given in (22), which is repeated here in (23) , do not inherently conflict in themselves.

(23) a. $|\text{restr}|P\text{D}_{\text{REL}}$
b. $|\text{op}_{\text{rel}}|A|\text{restr}|$
c. PARSE(|restr|)
d. PARSE($|\text{op}_{\text{rel}}|$)
e. PARSE(di)

That is, there is no pair of constraints A and B where constraint A can only be obeyed if and only if B is violated (cf. $|\text{agent}|P\text{D}_{\text{ARG}}$ and $|\text{theme}|P\text{D}_{\text{ARG}}$ for such a case).[9] This means that other constraints must affect the functional CUs present in relatives for constraint conflict to arise. This paper assumes that the effect of other constraints can influence |restr| and $|\text{op}_{\text{rel}}|$ in two possible ways. First, $|\text{op}_{\text{rel}}|$ can be affected by by other constraints since it assumes the role of another CU (possibly, a root). Thus, it is quite easy to imagine that another constraint would require $|\text{op}_{\text{rel}}|$ in a place further away from |restr| hence violating $|\text{op}_{\text{rel}}|A|\text{restr}|$. The other possibility stems from the fact that the relative domain coincides with another domain, which is called the propositional domain in this paper, and which includes the root element, everything depending on it, and (if applicable) whatever the root depends on. [10] Therefore, if another constraint requires some element to precede the propositional domain, that requirement will necessarily involve a conflict with $|\text{restr}|P\text{D}_{\text{REL}}$.

Here, the former case is discussed as the latter seems to have less relevance to the English data (although it is going to be referred to with respect to Hungarian). Therefore, suppose, for instance, an alignment constraint in the form of $|\text{op}_{\text{rel}}|F\text{x}$, which requires $|\text{op}_{\text{rel}}|$ to be on the right side of *x* conflicting with $|\text{op}_{\text{rel}}|A|\text{restr}|$. Five strategies are discussed here; three of those involve faithful output forms while the remaining two leads to underparsing. Underparsing will occur if PARSE($|\text{op}_{\text{rel}}|$) or PARSE(|restr|) is ranked lower than both of the conflicting alignment constraints. If $|\text{op}_{\text{rel}}|$ or |restr| is deleted, then the alignment constraints will be satisfied vacuously. The cases involving the deletion of |restr| and $|\text{op}_{\text{rel}}|$ are given in (24) and (25) respectively. The candidate set for each of the examples comprises of sequences that can emerge as winners with other rankings; that is the losers show the different ways of resolving constraint conflict in this cases. Note that the rankings used in these cases are termed to be "example" rankings as they have no relevance to the analysis of the English data; in fact, each of them is only ad hoc ranking whose only point is to show the possible results of constraint

[9]This is, however, only true when both constraints are active, and neither of them is satisfied vacuously.

[10]As Alignment Syntax does not recognise grammatical structure this domain is needed to be able to grasp generalisations which would otherwise involve the notion of a "clause" as a unit that belongs together and is manipulated as such by the grammar.

interaction.

(24) example ranking: |op$_{rel}$|*A*|restr| >> |op$_{rel}$|*F*x >> PARSE(|op$_{rel}$|) >> |restr|*P*D$_{REL}$ >> PARSE(|restr|)

	|op$_{rel}$|*A*|restr|	|op$_{rel}$|*F*x	PARSE(|op$_{rel}$|)	|restr|*P*D$_{REL}$	PARSE(|restr|)
|restr| |op$_{rel}$| x		*!			
|restr| x |op$_{rel}$|	*!				
x |restr| |op$_{rel}$|				*!	
☞ x |op$_{rel}$|					*
|restr| x			*!		

(25) example ranking: |op$_{rel}$|*A*|restr| >> |op$_{rel}$|*F*x >> PARSE(|restr|) >> |restr|*P*D$_{REL}$ >> PARSE(|op$_{rel}$|)

	|op$_{rel}$|*A*|restr|	|op$_{rel}$|*F*x	PARSE(|restr|)	|restr|*P*D$_{REL}$	PARSE(|op$_{rel}$|)
|restr| |op$_{rel}$| x		*!			
|restr| x |op$_{rel}$|	*!				
x |restr| |op$_{rel}$|				*!	
x |op$_{rel}$|			*!		
☞ |restr| x					*

As far as faithful candidates are concerned, there will always be one alignment constraint that is violated. Accordingly, there are three possible outcomes. First, if |op$_{rel}$|*F*x is violated, |restr| precedes the relative domain, and |op$_{rel}$| follows it immediately. Second, if |op$_{rel}$|*A*|restr| is violated, both |op$_{rel}$| is the relative domain following *x*. Third, if |restr|*P*D$_{REL}$ is violated, |restr| is situated inside the relative domain, as well. These three cases are shown below in this order.

(26) example ranking: PARSE(|op$_{rel}$|) >> PARSE(|restr|) >> |restr|*P*D$_{REL}$ >> |op$_{rel}$|*A*|restr| >> |op$_{rel}$|*F*x

	PARSE($\vert\text{op}_{\text{rel}}\vert$)	PARSE($\vert\text{restr}\vert$)	$\vert\text{op}_{\text{rel}}\vert A\vert\text{restr}\vert$	$\vert\text{op}_{\text{rel}}\vert F\text{x}$	$\vert\text{restr}\vert PD_{\text{REL}}$
$\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$ x				*!	
$\vert\text{restr}\vert$ x $\vert\text{op}_{\text{rel}}\vert$			*!		
☞ x $\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$					*
x $\vert\text{op}_{\text{rel}}\vert$		*!			
$\vert\text{restr}\vert$ x	*!				

	PARSE($\vert\text{op}_{\text{rel}}\vert$)	PARSE($\vert\text{restr}\vert$)	$\vert\text{restr}\vert PD_{\text{REL}}$	$\vert\text{op}_{\text{rel}}\vert A\vert\text{restr}\vert$	$\vert\text{op}_{\text{rel}}\vert F\text{x}$
☞ $\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$ x					*
$\vert\text{restr}\vert$ x $\vert\text{op}_{\text{rel}}\vert$				*!	
x $\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$			*!		
x $\vert\text{op}_{\text{rel}}\vert$		*!			
$\vert\text{restr}\vert$ x	*!				

(27) example ranking: PARSE($|\text{op}_{\text{rel}}|$) >> PARSE($|\text{restr}|$) >> $|\text{restr}|PD_{\text{REL}}$ >> $|\text{op}_{\text{rel}}|F\text{x}$ >> $|\text{op}_{\text{rel}}|A|\text{restr}|$

	PARSE($\vert\text{op}_{\text{rel}}\vert$)	PARSE($\vert\text{restr}\vert$)	$\vert\text{restr}\vert PD_{\text{REL}}$	$\vert\text{op}_{\text{rel}}\vert F\text{x}$	$\vert\text{op}_{\text{rel}}\vert A\vert\text{restr}\vert$
$\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$ x				*!	
☞ $\vert\text{restr}\vert$ x $\vert\text{op}_{\text{rel}}\vert$					*
x $\vert\text{restr}\vert\ \vert\text{op}_{\text{rel}}\vert$			*!		
x $\vert\text{op}_{\text{rel}}\vert$		*!			
$\vert\text{restr}\vert$ x	*!				

(28) example ranking: PARSE($|\text{op}_{\text{rel}}|$) >> PARSE($|\text{restr}|$) >> $|\text{op}_{\text{rel}}|A|\text{restr}|$ >> $|\text{op}_{\text{rel}}|F\text{x}$ >> $|\text{restr}|PD_{\text{REL}}$

Having seen the different outcomes of constraint conflict, cross-linguistic data can be now brought forward to demonstrate how these appear in natural languages. In these cases, the interfering alignment constraint (generally stated as $|\text{op}_{\text{rel}}|F\text{x}$ in the example) is the alignment constraint responsible for the placement of arguments. First, Hungarian is shown where the $|\text{restr}||\text{op}_{\text{rel}}|$ sequence is usually found preceding the relative domain, but another case is also described when a higher ranking domain-precedence constraint is able to push this sequence further in the domain. Next, Japanese examples are shown where the winner is always an unfaithful candidate. After that, examples from Mandarin Chinese demonstrate the case where only the $|\text{restr}|$ element is located on the edge of the relative domain. Finally, Tagalog is shown as a rather extreme case where, it seems, relativisation can trigger more significant changes in the sentence.

4.1 Hungarian

In Hungarian, relative pronouns are usually found preceding the relative clause as (29) shows.

(29) a könyv, amit János olvas
the book rel-pron.ACC John read.3s
"the book John is reading"

The relative pronoun is taken to be a |restr| $|op_{rel}|$ sequence as it bears the accusative case. The reason for that is that the functional CU that is spelled out as the accusative suffix must be associated with the Theme argument of the verb, which is the operator in this case. Also, the constraint that aligns |theme| with the element that bears that thematic role must be very high-ranked if not undominated. Therefore, the presence of the accusative morpheme at the left edge of the clause must indicate that $|op_{rel}|$is situated there. Normally, however the Theme argument (with neutral interpretation) would have a place following the subject.

(30) János egy könyvet olvas
John a book.ACC read.3s
"John is reading a book"

The relative pronoun cannot appear in that place, though. (31) is ungrammatical regardless of what interpretation the preceding element (*János*) might have.

(31) *a könyv, János amit olvas

This leads to the conclusion that the $|op_{rel}|$ element must satisfy a higher-ranked constraint by appearing at the front the clause. In other words, these relative clauses seem to be the result of the constraint interaction shown by (26).

There is, however, another relative construction documented by Nádasdy (2011) where the relative clause precedes the modified noun instead of following it.[11] And, only in this case, it is possible for a focused or a topicalised element to precede the relative pronoun as is shown in (32).

[11]Note that this construction must be a very recent development in Hungarian as neither Kenesei (1992), nor Kenesei, Vago & Fenyvesi (1998) mention it, and some speakers may even consider them questionable

(32) JÁNOS amit olvas könyv

This seems to indicate that there is an even higher-ranked constraint at work here, which makes it possible that elements of the relative domain precede the relative pronoun similarly to (28). Nevertheless, the side-switching phenomenon (i.e. that the relative clause where the relative pronoun is not the first in the domain appears preceding the modified noun while it follows the noun, otherwise) is not explained by this and is subject to future research.

4.2 Japanese

Relative clauses in Japanese seem to be a fairly straightforward issue with respect to the system outlined above. It provides an example for the case where the winner is an unfaithful candidate. Japanese relative clauses precede the modified noun and contain no element that could be identified as a lexical item corresponding to either |restr| or $|op_{rel}|$ as can be seen from (33).

(33) Tokyo kara kita Iguchi
Tokyo come perf. Iguchi
"Iguchi who has come from Tokyo""

(Kaiser et al., 2013: 519)

The question naturally is which CU is deleted in the winner. Without further research it is hardly possible to give a definite answer. Still, it would probably be easier to argue for the deletion of |restr| as it seems more straightforward to assume that $|op_{rel}|$ is associated with the empty word than the other way around. This way Japanese relative clauses can be described by a ranking similar to (24)

4.3 Mandarin

Similarly to Japanese, Mandarin relative clauses seem to be unproblematic, as well. They constitute an example for the operator left in place and the |restr| element getting replaced with a lexical element in itself. That lexical item is *de* as the examples show.

(34) zhong shuiguo de nongren
grow fruit REL farmer
"the farmer who grows fruit"

(35) tamen zhong de shuiguo
they grow REL fruit
"the fruit that they grow"

(Li & Thompson, 1981: 580-1)

Mandarin relatives are also underparsed ones, but here there is on overt element; most probably |restr|; thus, the ranking shown in (25) could be used to account for it.

4.4 Tagalog

Finally, the picture offered by Tagalog is far more intricate than the three previous ones. It seems that many types of alignment constraints must be quite highly-ranked in Tagalog as the winning candidate seems rather unfaithful to the input. For one thing, there is no relative pronoun. But what is far more intriguing than that is the fact that only nominative arguments can be relativised. If the operator would be a non-nominative argument of the verb, then the clause is passivised so that the operator will end up as a nominative argument.

(36) isda=ng i-b-in-igay ng=lalake sa=bata
fish=LNK IV-PERF-give GEN=man DAT=child
"the fish which was given to the child by the man"

(37) bata=ng b-in-igy-an ng=lalake ng=isda
child=LNK PERF-give-DV GEN=man DAT=fish
"the child which was given fish by the man"

(Kroeger, 1993: 23-4)

Therefore, it seems that in Tagalog even CUs which mark argument relations or voice can be deleted or re-ordered. to satisfy the alignment constraints stating the position of $|op_{rel}|$.

5 An alignment based analysis of English restrictive relatives

This section offers three different constraint rankings that are able to account for the three different kinds of relative clauses in English. First *wh-* relatives are discussed, and also a possible analysis of pied-piping. Then, *that*-relatives, and finally zero relatives.

There is one feature that is common to all of them: the asymmetry between the treatment of subjects and other arguments. The reason for this can be found in the nature of the constraints that regulate the order of arguments. Primarily, constraint conflict arises because arg*F*v and $|op_{rel}|A|restr|$ conflict when the operator is a non-subject argument. In the case of subjects, however, the s*P*v constraint is always present, and it outranks arg*F*v. Probably, the situation is similar with adverbs and other modifiers, which are not arguments. However, as their representation in Alignment Syntax is not so elaborated as that of arguments, they are not included in this part of the analysis. Furthermore, as long as there is a constraint that requires them to be further away from the first element of the domain, their behaviour should be similar to the behaviour of arguments.

Another assumption that is generally made here about the ranking of the constraints proposed in section 3 is that, among them, |restr|PD$_{REL}$ is always the highest ranking one. The grounding for this position is that if it were possible to solve constraint conflict by moving |restr| next to |op$_{rel}$| somewhere inside the domain, then one would find clause-internal *wh*-elements. As a result, PARSE(di) seems not to be active in most cases since the deletion of a domain index is a possible solution only when a number of elements compete for preceding the domain, and the deletion of an index can avoid the violation of a domain-precedence constraint. However, in the case of English, only |restr| wants to precede the domain. The exception from this is the case of pied-piping, and the question of underparsing domain indexes is addressed in the analysis of that phenomenon.

5.1 Ranking for *wh*-relatives

For *wh*-relatives, the proposed ranking is shown in (38).

(38) |restr|PD$_{REL}$ >> PARSE(|restr|) >> sPv >> PARSE(|op|) >> |op|A|restr|>> arg$F$v

This ranking results in winners where |op$_{rel}$| is always precedes the verb. Notice that if the operator is the subject, then this ranking has little effect as sPv forces the subject to appear at the right position, adjacent to |restr|. Also faithfulness constraints must also be relatively high-ranked so that no underparsed candidate beat the faithful one. The results of this ranking are shown in the tableau.

(39)

	Input: |restr|, V(|op|, O)	|restr|PD$_{REL}$	PARSE(dt)	sPv	PARSE(|op|)	|op|A|restr|	arg$F$v
☞	|restr||op| V O						*
	|restr|V S |op|					**!	
	|restr|V O				*!		*
	|op| V O		*!				*

On the other hand, when the operator is a non-subject argument, it is |op$_{rel}$|A|restr| that will be crucial because it is ranked above arg$F$v. Therefore, the operator is forced to be immediately following |restr| as (40).

(40)

	Input: |restr|, V(S, |op|)	|restr|PD_{REL}	PARSE(|restr|)	s*P*v	PARSE(|op|)	|op|*A*|restr|	arg*F*v
☞	|restr||op| S V						**
	|restr| S V |op|					**!	*
	|restr| SV				*!		*
	S V |op|		*!				*

5.1.1 Pied-piping and preposition stranding

As far as preposition stranding is concerned, this paper argues that, again, variation hinges on the relative ranking of two constraints. One is termed |prep|*P*n (prepositions precede the noun they are associated with), the other |prep|*F*v (prepositions follow the verb). Then, it already makes sense why pied-piping is specific to *wh*-relatives: only in this case does the noun ($|op_{rel}|$) a preposition is associated with end up in a position where the above two conflict. It seems enough at first sight to claim that if the ranking is |prep|*P*n >> |prep|*F*v, then pied piping occurs and preposition stranding happens in the opposite situation. However, notice that pied-piping in this case yields ungrammatical results as the following tableau shows.

(41)

		|restr|PD_{REL}	PARSE(|restr|)	PARSE($\|op_{rel}\|$)	|prep|*P*n	|prep|*F*v	$\|op_{rel}\|$*A*|restr|
	|prep| |restr| $\|op_{rel}\|$ V	*!				*	
☞	|restr| |prep| $\|op_{rel}\|$ V					*	*
	|prep| $\|op_{rel}\|$ |restr| V	**!				*	
	|restr| $\|op_{rel}\|$ V |prep|				*!		
	|restr|V |prep|			*!			

With ranking like this the following sentence is a winning candidate.

(42) *the film that about we talked

It is easy to see that the candidate that we expect to win instantly dies on the first constraint as |restr|is there is an element from the relative domain that precedes |restr|. The question arises whether that violation could be avoided. The answer is yes, it can be avoided by underparsing the domain feature of |prep|. Although that is a violation of PARSE(di), if it is ranked lower than $|op_{rel}|$*A*|restr| then the violations change in the following way (the element with the underparsed domain feature is marked with *x*).

(43)

<table>
<tr><th></th><th>|restr|PDREL</th><th>PARSE(|restr|)</th><th>PARSE(|op$_{rel}$|)</th><th>|prep|Pn</th><th>|prep|Fv</th><th>|op$_{rel}$|A|restr|</th><th>PARSE(di)</th></tr>
<tr><td>|prep| |restr| |op$_{rel}$| V</td><td>*!</td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>|restr| |prep| |op$_{rel}$| V</td><td></td><td></td><td></td><td></td><td>*</td><td>!*</td><td></td></tr>
<tr><td>|prep| |op$_{rel}$| |restr| V</td><td>**!</td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>|restr| |op$_{rel}$| V |prep|</td><td></td><td></td><td></td><td>*!</td><td></td><td></td><td></td></tr>
<tr><td>|restr|V |prep|</td><td></td><td></td><td>*!</td><td></td><td></td><td></td><td></td></tr>
<tr><td>☞ |prep|$_x$ |restr| |op$_{rel}$| V</td><td></td><td></td><td></td><td></td><td>*</td><td></td><td>*</td></tr>
</table>

Thus, by including underparsed candidates, the phenomenon of pied piping can be accounted for. That is pied-piping is analysed as the underparsing of the domain index of preposition that allows it to precede the noun it is associated with while not violating |restr| textitPD$_{REL}$.

5.2 Ranking for *that*-relatives

For *that*-relatives the following ranking is proposed.

(44) |restr|*P*D$_{REL}$ >> PARSE(|restr|) >> s*P*v >> arg*F*v >> PARSE(|op|) >> |op|*A*|restr|

What is quite interesting to note here is that the winner is a different kind of output depending on whether the operator is subject or some other argument. If it is a subject, then the winner is always a sequence with the |op$_{rel}$| CU underparsed as in (45). On the other hand, if it is not a subject, then |op$_{rel}$| is simply situated behind the verb avoiding a violation of arg*F*v. This can be seen in (46).

(45)

<table>
<tr><th>Input: |restr|, V(|op|, O)</th><th>|restr|PD$_{REL}$</th><th>PARSE(|restr|)</th><th>sPv</th><th>argFv</th><th>PARSE(|op|)</th><th>|op|A|restr|</th></tr>
<tr><td>|restr||op|V O</td><td></td><td></td><td></td><td>*!</td><td></td><td></td></tr>
<tr><td>|restr| V|op| O</td><td></td><td></td><td>*!</td><td></td><td></td><td>**</td></tr>
<tr><td>☞ |restr| V O</td><td></td><td></td><td></td><td></td><td>*</td><td></td></tr>
<tr><td>|op| V O</td><td></td><td>*!</td><td></td><td>*</td><td></td><td></td></tr>
</table>

(46)

<table>
<tr><th>Input: |restr|, V(S, |op|)</th><th>|restr|PD$_{REL}$</th><th>PARSE(|restr|)</th><th>sPv</th><th>argFv</th><th>PARSE(|op|)</th><th>|op|A|restr|</th></tr>
<tr><td>|restr||op| S V</td><td></td><td></td><td></td><td>*!</td><td></td><td></td></tr>
<tr><td>☞ |restr| S V |op|</td><td></td><td></td><td></td><td></td><td></td><td>**</td></tr>
<tr><td>|restr| S V</td><td></td><td></td><td></td><td></td><td>*!</td><td></td></tr>
<tr><td>S V |op|</td><td></td><td>*!</td><td></td><td></td><td></td><td></td></tr>
</table>

To summarise, the sequences this ranking produces are actually of two different kinds. In the case of subjects, an underparsed candidate wins (much like in Mandarin) while other arguments follow the verb as they would in any other type of clause.

5.3 Ranking for zero relatives

Finally, it is argued that zero relatives have the following ranking.

(47) |restr|*P*D$_{\text{REL}}$ >> s*P*v >> PARSE(|op|) >> |op|*A*|restr| >> arg*F*v >> PARSE(|restr|)

Again there is a significant difference between subject and object operators. If the operator is an object, |restr| is underparsed in the output. The case of the subject operator is more intriguing. though. Notice that there the candidate with deleted |restr| is actually harmonically bound by the winner. In other words, no matter how the constraints are ranked the former is always going to lose against the latter. That is quite an appealing point of the analysis as no further generalisations have to be made about the ungrammaticality of zero relatives with subject operators. Instead, those are always mapped onto *wh*-relatives. Below, the tableaux for the two different relatives are given. In the first, the there is a subject operator, and, in the second, there is an object operator.

(48)

	Input: |restr|, V(|op|, O)	|restr|*P*D$_{\text{REL}}$	s*P*v	PARSE(|op|)	|op|*A*|restr|	arg*F*v	PARSE(|restr|)
☞	|restr| |op| V O					*	
	|restr|V |op| O		*!		*		
	|restr|V O			*!			
	|op| V O					*	*!

(49)

	Input: |restr|, V(S, |op|)	|restr|*P*D$_{\text{REL}}$	s*P*v	PARSE(|op|)	|op|*A*|restr|	arg*F*v	PARSE(|restr|)
	|restr| |op| S V					**!	
	|restr| S V |op|				*!	*	
	|restr|S V			*!		*	
☞	S V |op|					*	*

In the case of non-subject arguments, underparsing occurs as the faithfulness constraint is ranked lower than any of the alignment constraints (which resembles the Japanese example). In the case, of subjects the conflict is resolved by a higher ranked alignment constraint, s*P*v.

To summarise, this section introduced three rankings that can produce the three types of English restrictive relatives, and most of the rankings resolved constraint conflict in a ways that was shown by the cross-linguistic examples. In *wh*-relatives is $|op_{rel}|$ always adjacent to |restr|, much like Hungarian relative pronouns. In *that*-relatives underparsing is present only in the case of subject operators while non-subject ones behave like normal arguments. Finally, zero relatives are underparsed if the operator is not the subject, but if it is, then a harmonically binding *wh*-form always beats the underparsed one.

Conclusion

This paper have presented an SFA-based analysis of English restrictive relatives. First, the standard OT assumptions about the input were reviewed and a brief survey was provided of an analysis that makes use of different input structures. Then, the second section introduced a uniform input for all types of relatives compatible with the assumptions of SFA. It was argued that the input contain to specific functional CUs: $|op_{rel}|$ and |restr|. The third section introduced the relevant constraints and the fourth showed in what way these can interact. Cross-linguistic examples were also mentioned that exemplify this kind of interaction. Finally, the fifth section presented the actual rankings that can produce the range of data observable in English without the need of the DFCF or any stipulation banning subject extraction in zero relatives.

References

Bayer, Josef and Ellen Brandner (2008). "On wh-head-movement and the Doubly-Filled-Comp-Filter". In: Charles B. Chang and Hannah J. Haynie (eds.) *Proceedings of the 26th West Coast Conference on Formal Linguistics*. (87-95). Somerville, MA: Cascadilla Proceedings Project.

Chomsky, Noam and Howard Lasnik (1977). "Filters and Control". *Linguistic Inquiry, 8*(3), 425-504.

Gáspár, Miklós (2005). "Coordination in Optimality Theory". PhD dissertation, Budapest: ELTE.

Grimshaw, Jane (1997). "Projections, Heads, and Optimality". *Linguistic Inquiry 28*(3).

Grimshaw, Jane (1998). "Constraints on constraints in Optimality Theoretic syntax". ms. Rutgers.

Hayes, Bruce and Colin Wilson (2008). "A Maximum Entropy model of phonotactics and phonotactic learning" *Linguistic Inquiry, 39*(3), 379-440.

Kager, René (1999). *Optimality Theory*. Cambridge, UK: CUP.

Kaiser, Stefan, Yasuko Ichikawa, Noriko Kobayashi, and Hilofumi Yamamoto (2001). *Japanese: a comprehensive grammar*. London: Routledge.

Keer, Edward and Eric Baković (1997). "Have FAITH in syntax". ROA Report #200, available at: `http://roa.rutgers.edu/files/200-0697/roa-200-keer-4.pdf`

Kenesei, István (1992) "Az alárendelt mondatok szerkezete" [The structure of subordinate clauses]. In: Frenec Kiefer (ed.) *Strukturális magyar nyelvtan*[Hungarian structural grammar] Vol. 1. (529-714). Budapest: Akadémiai Kiadó.

Kenesei, István, Robert Michael Vago, and Anna Fenyvesi, (1998). *Hungarian*. London: Routledge.

Kroeger, Paul (1993): *Phrase structure and grammatical relations in Tagalog*. Stanford: CSLI Publications.

Legendre, Geraldine (2001). "Introduction to Optimality Theory in syntax". In: Geraldine Legendre, Jane Grimshaw and Sten Vikner (eds.) Optimality Theoretic Syntax. Cambrdige, MA: MIT Press.

Li, Charles N., and Sandra A. Thompson (1981). *Mandarin Chinese: a functional reference grammar*. Berkeley: University of California Press.

Lipták, Anikó (2000). "Multiple relatives as relatives of question" In: Alberti, G. & Kenesei, I. (eds): *Approaches to Hungarian 7*. Szeged: JATE Press.

McCarthy, John (2008). *Doing Optimality Theory: applying theory to data*. Malden: Blackwell.

Nádasdy, Ádám (2011) "Balról korlátozó mellékmondatok" [Restrictive relative clauses on the left] Lecture handout. Institute of Linguistics, Hungarian Academy of Sciences.

Newson, Mark (2000). "A restrictive theory of alignment and some of its consequences". ms. Eötvös Loránd University

Newson, Mark (2004). "Deforestation in syntax". In: Varga, L. (ed.) *The Even Yearbook 6*. (135-148). Budapest: ELTE SEAS.

Newson, Mark (2010). "Syntax first, words after: A possible consequence of doing Alignment Syntax without a lexicon" In: Varga, L. (ed.) *The Even Yearbook 9*. Budapest: ELTE SEAS.

Newson, Mark (2013). "What you can do with a domain and why they are not phrases". In: Szigetvári, P. (ed.) *Papers Presented to László Varga on his 70th birthday*. Budapest: ELTE SEAS.

Newson, Mark (2014). "Alignment Syntax". Course taught at Eötvös Loránd University, Budapest.

Newson, Mark and Krisztina Szécsényi (2012). "Dummy auxiliaries and late vocabulary insertion". In: Varga, L. (ed.) *The Even Yearbook 10*. Budapest: ELTE SEAS.

Prince, Alan and Paul Smolensky (1993/2002) "Optimality Theory: Constraint interaction in generative grammar." ROA Report #537 available at: `http://roa.rutgers.edu/files/537-0802/537-0802-PRINCE-0-0.PDF`

Sadock, Jerrold M. (2003). *A grammar of Kalaallisut (West Greenlandic Inuttut)* München: Lincom Verlag.