DIPLOMAMUNKA

MA THESIS

Kucsera Márton
Anglisztika MA
Elméleti nyelvészet szakirány

2014

CERTIFICATE OF RESEARCH

By my signature below, I certify that my ELTE MA thesis, entitled

ENGLISH RESTRICTIVE RELATIVE CLAUSES IN OPTIMALITY THEORY

is entirely the result of my own work, and that no degree has previously been conferred upon me for this work. In my thesis I have cited all the sources (printed, electronic or oral) I have used faithfully and have always indicated their origin. The electronic version of my thesis (in PDF format) is a true representation (identical copy) of this printed version. If this pledge is found to be false, I realize that I will be subject to penalties up to and including the forfeiture of the degree earned by my thesis.

Date: April 13, 2014

Signed: ....

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Bölcsészettudományi Kar

DIPLOMAMUNKA

MA THESIS

Angol korlátozó értelmű vonatkozó mellékmondatok

az optimalitáselméletben

English restrictive relative clauses in Optimality Theory

Témavezető:

Dr. Hordós Marianna
egyetemi adjunktus

Készítette:

Kucsera Márton
Anglisztika MA
Elméleti nyelvészet szakirány

2014

Abstract

This thesis discusses the treatment of English restrictive relative clauses in Optimality Theory (OT). In particular, it addresses the question of how they can be represented in Alignment Syntax. It is assumed that the input of restrictive relatives contains two specific functional Conceptual Units (CU). Then, it is shown how the alignment and faithfulness constraints targeting these CUs interact with the other constraints that regulate the order of arguments in a sentence. Also, the issue of handling variation in OT is discussed. This paper argues — in compliance with the view that Alignment Syntax has about the nature of the input — that multiple constraint rankings are responsible for variation. Therefore, different rankings are worked out for the three types of English restrictive relatives: *wh*-, *that*-, and zero relatives. Finally, it is suggested that an analysis with multiple rankings be integrated into a probabilistic grammar.

Contents

Introduction

The present paper discusses the status of English restrictive relative clauses in Optimality Theory (OT). In particular, the problem addressed here is how they can be integrated into a more restricted version of OT, Alignment Syntax. This theory was first proposed by Newson (2000) and, in essence, it reduces syntax to the interaction of faithfulness constraints and alignment constraints, which regulate the linear arrangement of elements within the sentence. As a result, syntactic structure as such is eradicated from Alignment Syntax.[1] In addition, Alignment Syntax sets strict requirements about the nature of the input, which make it impossible to account for variation phenomena to be explained in terms of differing inputs. Thus, the aim of this paper is to show that the analysis of restrictive relative clauses can be integrated into this framework.

The first chapter discusses the nature of the input. First, it is shown that standard OT works with a very loose definition of what the input is. Then, the standpoint of Alignment Syntax is explained concerning the input. The idea of Late Lexical Insertion is introduced, which allows the syntax to manipulate abstract entities (Conceptual Units). Finally, the assumptions about the nature of the input for relative clauses are elaborated on. It is argued that they involve two specific types of Conceptual Units: |restr| and $|\mathrm{op}_{\mathrm{rel}}|$.

The second chapter is about the nature of constraints that the Evaluator component

[1] An important note on terminology: while the paper attempts to avoid using expressions that implicitly assume the existence of phrases and syntactic structure altogether, such as *structure* itself, or *construction*, in certain cases such expressions cannot be replaced. Therefore, the reader is advised not take the appearance of words like *clause*, *head*, or *antecedent* to be evidence that these are actual entities manipulated by the grammar.

(EVAL) consists of in OT and Alignment Syntax, and the capabilities of the Generator component (GEN). This chapter is organised much like the first. It starts with a review of standard OT's view on the general nature of constraints. After that, the two constraint types of Alignment Syntax are reviewed. Finally, the constraints that are necessary to account for restrictive relative clauses are introduced.

The third chapter shows how the different rankings of the proposed constraints can account for the three main types of restrictive relatives shown in (1).

(1) a. the man who we know
 b. the man that we know
 c. the man we know

Also the problem of pied-piping and preposition stranding are given a possible analysis. In this chapter, the three types of relatives are introduced independently of each other, and it is one of the aims of the final chapter to show how those different rankings can be reconciled in a single grammar.

The fourth and final chapter has a twofold purpose. First, it addresses additional problems that are not part of the analysis proposed in chapter 3. Then, it reviews models of grammar which allow multiple rankings to exist simultaneously; thus, making possible for the variation exhibited by relative clauses to be part of an Alignment Syntactic grammar.

Chapter 1

The input

This section examines the nature of the input. First, the assumptions of Standard Optimality Theory are examined, and certain problems are pointed out in connection with them. Next, the input of Alignment Syntax is introduced with its two main features: being the sole locus of semantic interpretation and consisting of abstract Conceptual Units instead of lexical items. Finally, it is shown what kind of abstract input can be assumed for restrictive relative clauses, and how the selection of the input elements can account for the Doubly Filled COMP Filter.

1.1 The input in standard OT

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

[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.)

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.

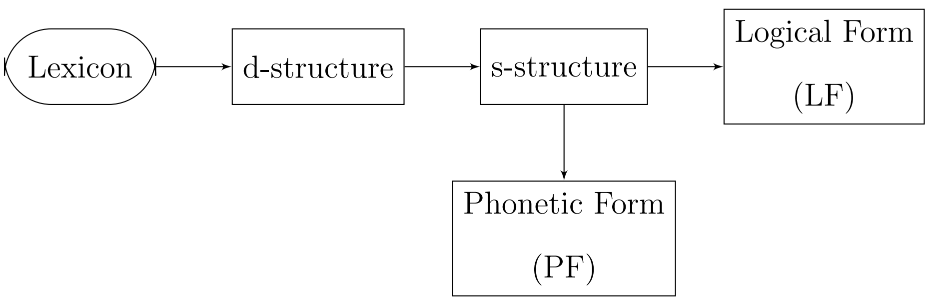
One might perhaps try to escape this problem by claiming that what EVAL has access to is only a set of possible dummy elements, so it cannot actually access the lexicon. This argument can be refuted, however, on the basis of the fact that every dummy element has a counterpart that is a full-fledged lexical item, so EVAL still has to differentiate between

[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.

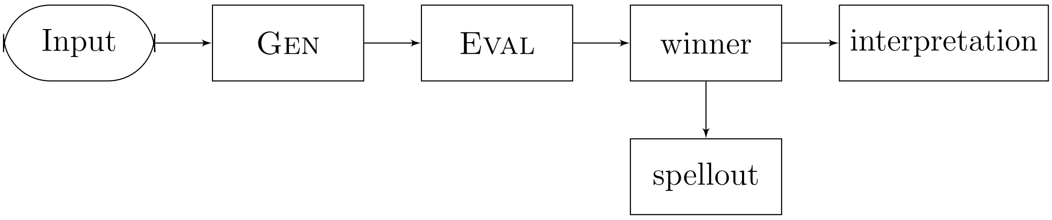
those two. Another alternative would be to claim that FullInt is actually similar to the DepIO constraint developed for phonology by Prince and Smolensky(1993/2002). In their framework, DepIO is a faithfulness constraint that assures that no input elements are deleted. Basically, it checks that every 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.

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. In light of this, the similarities between (3) (based on Haegeman, 1994) and (4) are fairly understandable.

(3) *A model of a standard GB grammar*



(4) *A possible model of Grimshaw's OT grammar*



1.2 The input in Alignment Syntax

The question of the interpretation of the input is handled differently by Alignment Syntax. Newson (2004) proposes that only the input serves as the basis for semantic interpretation. Thus, once meaning is read off the input, candidates can be generated and evaluated without respect to meaning. In other words, the members of the candidate set for a given input will not have different meanings; the difference lies either in the linear order of the elements or in the underparsing of the input.

This approach obviously assumes that semantic relationships are reflected in the input; for example, the arguments of predicates are specified, and, in the case of relative clauses, scope relations are also marked. Therefore, it will not be necessary to put operators into a position where they can take scope over their domain as scope, along with every other meaning relationship is already interpreted in the input before the candidate set is generated and evaluated. Thus, the input in Alignment Syntax includes individual elements as well as the meaning relationships that hold between them.

There is an issue in connection with the semantic specification of the input that must be addressed here. One might perhaps argue that this kind of input is, in fact, a way of smuggling back bits of structure into the alignment-based grammar. It must be emphasised, therefore, that the kinds of relations present in the input are not at all like syntactic structure in other theories. The simplest way to see this is to consider that the trees in Phrase Structure Grammars always express two different relations: dominance and precedence (Partee et al., 1987). On the other hand, the input in Alignment Syntax is completely void of a precedence relation. Also, it would be hard to argue that involves a dominance relation either since the specification of the input (even if one were to represent it in the form of a dominance relation) would hardly conform to the tree diagrams of other generative theories as it is not based on immediate constituent structure. Therefore, it is safe to argue that specifying semantic relationships in the input does not equal to including structure in Alignment Syntax.

1.2.1 Late Lexical Insertion

Another question that must be answered in connection with the nature of the input is what kind of elements it contains. Newson's (2010) proposal about the nature of the input is adopted here. He claims that the input is not made up of lexical items but of abstract, universal bits of meaning, and such an element is called a Conceptual Unit (CU). This also means that the syntax has no access to the lexical items; it only manipulates CUs. It is only after the selection of the most harmonic candidate that the abstract CUs get replaced with particular lexical items, through the process of Late Lexical Insertion.

What this process involves is actually replacing the sequence of CUs with elements from the lexicon whose meaning involves the particular CUs. This process is fairly similar to the lexical insertion process of Nanosyntax, described by Starke (2009). There is, however, a significant difference between the two as the latter works with bits of structure; i.e. each lexical item is a sub-tree comprising a number of CUs. During the insertion process part of the syntactic structure corresponding to the sub-tree associated with a lexical item is replaced with that lexical item.

As Alignment Syntax does not have any kind of structure the main criterion is that only *contiguous elements* can be replaced with a single lexical item. For instance, suppose that there is a lexical element A that corresponds to the CUs |x| and |y|. In the winner, A can only replace both |x| and |y| only if those two are adjacent and in the order specified by A. A further criterion is *minimal vocabulary access*, which means that the least possible number of lexical elements must be inserted. In other words, if two CUs correspond to a single lexical item, they should be replaced with that instead of inserting more lexical items. For example, consider the following three correspondences between lexical elements and CUs: $A \leftrightarrow$ |x||y|, $B \leftrightarrow$|x|, $C \leftrightarrow$|y|. Minimal vocabulary access means that an |x||y| sequence is always replaced with A instead of B C since the latter option would require accessing the lexicon multiple times.

Finally, like Nanosyntax, a superset-approach is adopted towards lexical insertion in Alignment Syntax, as well; that is, the inserted lexical item must minimally include the CUs specified by the syntax, and it can include further elements, if there is no exact match. Newson (2010) also mentions the possibility of a different approach, where the CUs of the inserted lexical item are, instead of a superset, a subset of the CUs specified by the syntax if there is no exact match in the lexicon. This subset approach is typical in Distributed Morphology (developed by Halle and Marantz, 1993). Alignment Syntax, however, is argued to operate with the subset approach; otherwise, further methods would be needed for lexical insertion such as, Fusion, Fission, and Impoverishment in Distributed Morphology.

1.3 The input for relative clauses

This section discusses what elements are in the input of relative clauses. Since it was discussed earlier (in section 1.2) that in Alignment Syntax it is only the input that is interpreted, one must define very carefully what is part of the input. 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 as it primarily because a number of languages use actual determiners to introduce relative clauses (in fact, *that* in English is also appear as a determiner). Consider the example from Kalaallisut (West Greenlandic) in (5). There, the relative clause is introduced by *tassani* which is a determiner with a locative suffix.

(5) 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, as argued in section 2.2. Therefore, both of them must be present in the form of functional CUs. Accordingly, I assume that the input for the relative clauses in (6) consists of the elements shown in (7).

(6) a. the book that Bill read

b. the book which Bill read

c. the book Bill read

(7) $\sqrt{BOOK}$

$|\text{restr}|_i$

$\sqrt{READ}$ (|agent|, |theme|)

$|\text{agent}| = \sqrt{BILL}$

$|\text{theme}| = |\text{op}_{\text{rel}}|_i(\sqrt{BOOK})$

The example shows that the input consists of two kinds of units: root (marked with $\surd$) 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 (7), 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 $|\mathrm{op}_{\mathrm{rel}}|$. It must be emphasized that every $|\mathrm{op}_{\mathrm{rel}}|$ CU must be related to a |restr|(that is shown by the matching indices in (7)), and maximally one $|\mathrm{op}_{\mathrm{rel}}|$ might belong to a |restr|.[4] Otherwise, $|\mathrm{op}_{\mathrm{rel}}|$ would become far too general a tool in the grammar; for instance, if its appearance would not be constrained it could

[4]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.

(8)	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.

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

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

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

easily be used to replace pronouns.

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 |op_{rel}|, relative clauses cannot have multiple operators while interrogatives certainly can.[5]

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 |op_{rel}|. In this paper, it is argued that there are three different outcomes of lexical insertion depending in the configuration of these two CUs.[6] 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 |op_{rel}| and |restr| CUs. Thus, in the light of the minimal vocabulary access principle formulated in section 1.2.1, a *wh*-element is always inserted to replace a |restr| |op_{rel}| sequence. A further assumption is that when not next ot |restr|, |op_{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

(11) *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.

[5]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*.

[6]Naturally, if deletion is allowed (as this paper assumes it is; cf. section2.2), one would expect five possible orderings of the elements A A and B: AB, BA, A, B, and $\emptyset$. The reason for this apparent lack of two of the possibilities is only outlined here and will be elaborated on in the following sections.

that every adjacent |restr|, |op_{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 |op_{rel}|only leaving |restr|in the sequence which can, in turn, be replaced with a complementiser.

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

(12) 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].

In sum, this chapter argued for adopting the assumptions in Newson (2010) for the analysis of relative clauses. In particular, it was claimed that, first, the only place for semantic interpretation is the input. Then, following other theories, like Distributed Morphology and Nanosyntax, it was argued that lexical items are inserted only after syntactic operations, and they are inserted on a best-fit basis. Finally, the input of

English restrictive relative clauses was discussed and it was shown that they contain two kinds of functional CUs: |restr| and $|\mathrm{op}_{\mathrm{rel}}|\dot{\mathrm{T}}$he hypothesis is that a |restr|, $|\mathrm{op}_{\mathrm{rel}}|$ sequence is spelled out as a *wh*-word, a single |restr| as *that* while a single $|\mathrm{op}_{\mathrm{rel}}|$ is associated with an empty word. Therefore, the analysis is able to account for the Doubly Filled COMP Filter without any further assumptions.

Chapter 2

The generator and possible constraints

This chapter discusses the nature of possible constraints and GEN. The first section examines the nature of those in standard OT. Two problems with the approach OT takes are described. First, it is pointed out that constraints that can have any possible form or are dependent on the apparatus of another theory actually prohibit OT to constitute a theory of language on its own. Secondly, the problem of infinite candidate sets is discussed. The next section introduces the basic constraint types Alignment Syntax uses, and gives a definition of precedence, subsequence, adjacency and domain-based constraints. The final section discusses what constraints are relevant in the examination of relative clauses and introduces a purely notational simplification in the representation of argument structure that is going to be used throughout this paper.

2.1 Possible constraints and the generator in standard OT

There is actually quite little to be said about the nature of OT constraints in general as there can be significant differences among authors concerning the set of possible constraints. This section shows that the reason for that is the very general nature of OT, which makes a very (perhaps undesirably) large range of constraints possible. This section shows what works in OT generally sate about the nature of constraints and about the capabilities of GEN.

First, the nature of constraints largely depends on what linguistic theory is adopted besides OT. While this issue is seldom made explicit in OT analyses, it must be borne in mind that OT in itself is not really more than a very general device that can select the "most harmonic" element from a set based on certain criteria. For instance, McCarthy (2008) makes it clear that OT can accommodate a wide range of phonological hypotheses (concerning, for example, syllable structure, the nature of segments etc.). The same can be said about OT syntax; that is, the theory adopted will partially predetermine the constraints available for the analysis. An example for that might be the constraint FAITH; this constraint makes sense only if the analysis uses a theoretical framework that allows transformation by movement, like Grimshaw's (1997) seminal paper does.

Second, authors working in an OT framework rarely define what exactly counts as a constraint. In syntax, a notable exception is Grimshaw, 1998; however, even her restrictions are not particularly strong, and she admits that it is possible for the proposed system to be changed or expanded. In phonology, de Lacy (2011) proposes a Constraint Definition Language (CDL), but some aspects of his system are not made clear, and it cannot handle gradient constraints (constraints that can be violated to a greater or a smaller extent) easily. On the other hand, McCarthy (2008) basically claims that anything can be a constraint that can be given in the form "assign one violation mark for every X that...", however, the reader is warned that the most plausible constraints have some additional grounding.

As a result, the only standard requirement for OT constraints is to be universal. This is basically the only restriction that Legendre (2001) makes in connection with possible constraints. In sum, one can argue that the only common trait of constraints in standard OT is that their effect must be shown in a number of languages using factorial typology.

More or less the same can be said about Gen. There is not much to describe its nature as it is completely determined by the constraints in Eval. That is, one must assume that the generator is only capable of such operations that are constrained against in the evaluation. For instance, it is not hard to see that Gen is allowed to move elements if and only if the position of the elements are subject to constraints. Otherwise, every permutation of the same elements would be considered equally optimal.

A related issue is whether Gen is allowed to insert and delete input elements. Prince and Smolenksy (1993/2002) allow both; in other words, they have constraints against both deletion (MaxIO) and insertion (DepIO). This view is carried over to syntax by Grimshaw (1997), who renamed these two constraints Parse and Fill respectively. While it is common to assume that insertion is possible, this approach also creates a significant problem, namely, that allowing insertion makes the candidate set infinite. In other words, searching for the optimal candidate can take an infinite amount of time. While it is possible to claim that there is always a finite subset of the candidate set that harmonically binds the rest of the candidates, the infinite number of candidates still creates a problem for standard OT analyses (Kager, 1999).

2.2 Alignment Syntax

This section offers solutions to the two problems discussed above within the framework of Alignment Syntax. First, it is discussed how one can formulate a restrictive theory of language within OT by strictly defining constraints. In particular, the system developed by Newson (2004, 2010) makes use of two basic constraint types: alignment and faithfulness. In this section, it is shown what forms of these will be used later on. Then, a more restricted generator is shown that can generate finite candidate sets only.

To begin with, alignment constraints are sensitive to the linear position of elements within a sequence. Every alignment constraint specifies a relation between two elements: a very general form of such constraints is given in (13). It is important to see that alignment always involves two elements: constraints specify the position of one element called the target (x, in the example) with respect to another one called the host (y, in the example).

(13) xRy: violated if the relation R does not hold between x and y

This assumption has two important consequences for the analysis. First, the positions "first" and "last" are not defined in this framework. That is, constraints in the form of (14) cannot be used in the analysis.

(14) First(x): x is the first element of the sequence (assign one violation mark for every element that precedes x)

Although similar constraints are not unprecedented in alignment-based accounts (cf. Gáspár, 2004), it is argued here that all phenomena that seem to involve the notion of 'first' and 'last' can, in fact, be accounted for with the use of domain-alignment.

Secondly, positing the same general form for all alignment constraints means that, with a number of relations defined, one can describe the range of possible alignment constraints. This paper adopts the relations defined by Newson (2010) as shown in (15).

(15) Possible alignment relations:

a. precedence/subsequence

b. adjacency

These two types basically describe two different ways to align the target to the host. Precedence and subsequence specify which side of the host the target should be on. These relations are marked with P (precedes) and F (follows), respectively. They are, however not sensitive to the distance (i.e. intervening elements) between the target and the host. This is shown by the small tableau in (16).

(16)

	xPy
☞ xy	
yx	*
☞ xay	
yabx	*

Adjacency, on the other hand, complements precedence and subsequence as it is only violated by elements between the target and the host and is insensitive to the relative order of these two. In other words, adjacency-constraints want the target to be as close to the host as possible. Notice how the winning candidates change in the candidate set of (16) if the precedence-constraint is replaced with an adjacency-constraint (marked with A).

(17)

	xAy
☞ xy	
☞ yx	
xay	*
yabx	**

The tableaux above show an additional difference between the two types of constraints that must be mentioned. Notice that the precedence-type can either be satisfied or violated only once in every sequence. [7] That is to say, precedence-type constraints are categorical: they are sensitive to a configuration that can be expressed as a binary opposition: element x either precedes y or it does not. Adjacency-constraints, on the

[7] There are cases where precedence constraints have more than one violation, but in those cases there are multiple targets. Therefore, multiple violations can occur when the constraint is evaluated more than once with different targets. For instance, if there are two elements x following element y, then xPy will have two violation marks.

other hand, can be violated more than once by a single target if there are a number of elements between it and its host as can be seen in the case of the last candidate in (17). That is, this constraint-type is gradient: it can be violated to any degree; it "counts" the number of intervening elements.

Another important property of this system that has to be pointed out is that it abandons the idea of different violation types. Originally, Newson (2004) proposed that the adjacency relation is integrated into the precedence/subsequence constraints; this idea is present in earlier works in Alignment Syntax (cf. Kántor, 2003 and Gáspár, 2004). This means that a single type of constraints is responsible for prohibiting targets on the wrong side of the host (side violations) and targets too far away from their respective hosts (edge violations). The assumption was that a side violation (marked with ☆) is more serious than any number of edge violations (marked with *).

Although abandoning this proposal leads to a larger constraint inventory, a system with only one type of violation is more desirable for two reasons. First, selecting the winning candidate becomes a simpler task computationally as different types of violation do not have to be compared. That is, constraints become much simpler as the ones with two types of violations must include (at least implicitly) a condition to decide which one to assign. (18) shows a possible way of making that condition explicit.

(18) xRy:

a. IF R does not hold between x and y: assign ☆

b. ELSE: assign * for each element between x and y

It must be borne in mind, however, that it is generally undesirable to have constraints in such form. Both Grimshaw (1998) and McCarthy (2008) agree that, as a consequence of violability, OT constraints should be maximally general, without conditions or provisos of any kind. That provides a further argument for replacing these kinds of constraints with simpler ones.

Second, the dominance of precedence/subsequence over adjacency is not hard-wired into the system thereby allowing it to account for side-switching phenomena more easily (cf. Newson, 2010 on Hungarian verbal prefixes). Therefore, this paper does not use constraints which can be violated in different ways.

The next question this sections considers is the nature of constraints that make reference to domains. As discussed in section 1.2, it is assumed here that a number of elements in the input can constitute a domain. It is natural, therefore, to ask how the syntax can work with these domains. First and foremost, domains only appear as hosts in any constraint since, this way, the grammar does not actually manipulate them as some kind of structure (Newson, 2014). In other words, while single elements can be aligned with respect to a domain, a domain in itself cannot be aligned to anything.

Another question that arises is what meaning the alignment relations shown above have when applied to a domain and a single element instead of a pair of elements. The answer to how precedence and subsequence can be applied to domains seems fairly straightforward. If an element precedes a domain it must precede everything in the domain; if it follows the domain, it follows all its members. An example is shown in (19).

(19) $\mathrm{x}P\ \mathrm{D_Y}$: assign one violation mark for each element in $\mathrm{D_Y}$ that precedes x

The notion of being adjacent to a domain is less obvious intuitively, but Newson and Maunula (2006) define it as follows. According to their assumptions element x is adjacent to a domain $\mathrm{D_Y}$ if all its members either follow or precede x. This constraint is categorical; i.e. for every target, one violation can be assigned. However, the analysis proposed in this paper does not use domain adjacency; thus, it will not be further mentioned.

Finally, it is discussed how it is possible to have a candidate set that is finite in all cases. This question is actually simpler than it might look at first glance. As described in section 2.1, theoretically, Gen is allowed to perform only such operations that are constrained against. This basically means that, if no Fill constraint is included in the system (i.e. insertion is not constrained), then Gen cannot insert anything. This assumption comes with two consequences. First, the fact that all candidates contain only input elements could make it harder to account for meaningless "dummy" elements. Second, this assumption makes sure that the candidate set will always be finite, which, in turn, means that searching for the optimal candidate can never take an infinite amount of time. In fact, the candidate set comprises of all the possible orderings over all the subsets of the input. The number of candidates, then, is the sum of all subsets of the input multiplied by the factorial of their cardinality as is shown by the formula below.

(20) Cardinality of the candidate set for an input with n elements:

$$\sum_{i=0}^{n} i = \binom{n}{i} i!$$

While this number can be fairly large, it will never be infinite, meaning that finding the most harmonic candidate will always be a finite task even without using any kind of special searching algorithm.

2.3 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 (section 1.2.1), 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 sPv (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 (21).

(21) $\sqrt{}{}^{*}P\mathrm{D}_{\mathrm{ARG}}$ >> $\sqrt{}P\mathrm{D}_{\mathrm{ARG}}$ >> |agent|$P\mathrm{D}_{\mathrm{ARG}}$ >> |goal|$P\mathrm{D}_{\mathrm{ARG}}$ >> |theme|$P\mathrm{D}_{\mathrm{ARG}}$

If one considers how the ranking in (21) 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 sPv >> arg$F$v. The tableaux comparing the two are given in (22) and (23).

(22)

	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.

(23)

	Input: hit (Jim, Bill)	sPv	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 (21) 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 sPv 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 in section 1.3: |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.

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

There is, at least, another constraint that must be posited to account for the emergence of the |restr| $|\mathrm{op}_{\mathrm{rel}}|$ 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.

(25) a. $|\mathrm{op}_{\mathrm{rel}}|F|\mathrm{restr}|$: $|\mathrm{op}_{\mathrm{rel}}|$ follows |restr|

b. $|\mathrm{op}_{\mathrm{rel}}|A|\mathrm{restr}|$: assign one violation mark for each element between $|\mathrm{op}_{\mathrm{rel}}|$ and |restr|

In this case, however, (25-a) must always be ranked above (25-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 (24) above (25-a). Notice that this ranking necessarily involves obeying (25-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.

(26)

	$\|\text{restr}\|P\text{D}_{\text{REL}}$	$\|\text{op}_{\text{rel}}\|A\|\text{restr}\|$
$\|\text{op}_{\text{rel}}\|\|\text{restr}\|$	*!	
☞ $\|\text{restr}\|\|\text{op}_{\text{rel}}\|$		
$\|\text{restr}\|$ 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 (26) 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 3.3.1. 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($|\text{op}_{\text{rel}}|$) 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 sPv 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 (27).

(27)
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)

The first two are alignment constraints. (27-a) requires the |restr| element to precede every member of D_{REL}while (27-b) prefers $|\text{op}_{\text{rel}}|$ to be adjacent to |restr|. The last three are faithfulness constraints which penalise the deletion of |restr|, $|\text{op}_{\text{rel}}|$, and domain indexes, respectively. Their interactions in the different kinds of restrictive relatives are discussed in the next chapter.

Chapter 3

Analysis of English restrictive relative clauses

3.1 Dealing with variation: an analysis by Keer and Baković

As mentioned before, it is not entirely straightforward how to handle variation in Optimality Theoretic grammars as they always assign a most harmonic candidate to a given input. As the different forms of relative clauses clearly represent a case of variation one has to ask the question how the grammar should be modified to be able to handle this issue. There are two possible answers to this problem. First, it can be assumed that what seems to be variation on the surface is actually not as there are different inputs for each of the different outputs. The other possibility is that ranking of the constraints in Eval is somehow subject to change; therefore, it is possible to have different winners for the same input if it is evaluated more than once. This section shows why the first possibility is theoretically undesirable and untenable in Alignment Syntax. Thus, it is assumed that the second approach is preferable although the details are going to be worked out only in section 4.2.

First, an analysis that uses multiple inputs to account for variation is reviewed. Keer and Baković (1997) analyse English relative clauses using three different kinds of input. Their analysis is briefly presented, then problems with such an approach are discussed. The analysis basically claims that relative clauses can come with three different operators: *wh*-operators, empty operators (i.e. a phonologically null element that, in many respects behaves like a *wh*-operator), and with no operator at all.

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 (28).

(28) *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.

(29) 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
☞ $[_{IP} \ldots V\ t_i \ldots]$			
☞ $[_{CP}$ that $[_{IP} \ldots V\ t_i \ldots]]$			

(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 (30).

(30) 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 in section 1.2, 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 (31).

(31) 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 (32).

(32) $[_{CP}\ [_{PP}\ [_{P'}$ 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.

On the other hand, the approach of this paper can be extended to the phenomena described above. In OT, a different winner is selected if either (i) the input is different or (ii) the constraints are ranked differently in EVAL. The way to proceed then is to assume that the constraint hierarchy of EVAL is not constant; thus, it can yield different winners for the same input. How such a grammar is supposed to work will be discussed in detail in chapter 4.2. This chapter is going to show what rankings of the constraints introduced in section 2.3 can result in different kinds of relative clauses. The issue of how an OT grammar without a consistent EVAL can function is addressed in the final chapter.

3.2 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 2.3. 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 (27), which is repeated here in (33) , do not inherently conflict in themselves.

(33) a. $|\text{restr}|P\text{D}_{\text{REL}}$

b. $|\text{op}_{\text{rel}}|A|\text{restr}|$

c. $\textsc{Parse}(|\text{restr}|)$

d. $\textsc{Parse}(|\text{op}_{\text{rel}}|)$

e. $\textsc{Parse}(\text{di})$

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).[8] 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, as it was discussed in section 1.3, $|\text{op}_{\text{rel}}|$ 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. [9] 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

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

[9] 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.

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 (34) and (35) 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 interaction.

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

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

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

	$|\text{op}_{\text{rel}}|A|\text{restr}|$	$|\text{op}_{\text{rel}}|F\text{x}$	PARSE(|restr|)	$|\text{restr}|PD_{\text{REL}}$	PARSE($|\text{op}_{\text{rel}}|$)
$|\text{restr}|\ |\text{op}_{\text{rel}}|$ x		*!			
$|\text{restr}|$ x $|\text{op}_{\text{rel}}|$	*!				
x $|\text{restr}|\ |\text{op}_{\text{rel}}|$				*!	
x $|\text{op}_{\text{rel}}|$			*!		
☞ $|\text{restr}|$ 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 $|\mathrm{op_{rel}}|F\mathrm{x}$ is violated, |restr| precedes the relative domain, and $|\mathrm{op_{rel}}|$ follows it immediately. Second, if $|\mathrm{op_{rel}}|A|\mathrm{restr}|$ is violated, both $|\mathrm{op_{rel}}|$ is the relative domain following x. Third, if $|\mathrm{restr}|P\mathrm{D_{REL}}$ is violated, |restr| is situated inside the relative domain, as well. These three cases are shown below in this order.

(36) example ranking: PARSE($|\mathrm{op_{rel}}|$) >> PARSE(|restr|) >> $|\mathrm{restr}|P\mathrm{D_{REL}}$ >> $|\mathrm{op_{rel}}|A|\mathrm{restr}|$ >> $|\mathrm{op_{rel}}|F\mathrm{x}$

	PARSE($\|\mathrm{op_{rel}}\|$)	PARSE(|restr|)	$\|\mathrm{restr}\|P\mathrm{D_{REL}}$	$\|\mathrm{op_{rel}}\|A\|\mathrm{restr}\|$	$\|\mathrm{op_{rel}}\|F\mathrm{x}$
☞ |restr| $\|\mathrm{op_{rel}}\|$ x					*
|restr| x $\|\mathrm{op_{rel}}\|$				*!	
x |restr| $\|\mathrm{op_{rel}}\|$			*!		
x $\|\mathrm{op_{rel}}\|$		*!			
|restr| x	*!				

(37) example ranking: PARSE($|\mathrm{op_{rel}}|$) >> PARSE(|restr|) >> $|\mathrm{restr}|P\mathrm{D_{REL}}$ >> $|\mathrm{op_{rel}}|F\mathrm{x}$ >> $|\mathrm{op_{rel}}|A|\mathrm{restr}|$

	PARSE($\|\mathrm{op_{rel}}\|$)	PARSE(|restr|)	$\|\mathrm{restr}\|P\mathrm{D_{REL}}$	$\|\mathrm{op_{rel}}\|F\mathrm{x}$	$\|\mathrm{op_{rel}}\|A\|\mathrm{restr}\|$
|restr| $\|\mathrm{op_{rel}}\|$ x				*!	
☞ |restr| x $\|\mathrm{op_{rel}}\|$					*
x |restr| $\|\mathrm{op_{rel}}\|$			*!		
x $\|\mathrm{op_{rel}}\|$		*!			
|restr| x	*!				

(38) example ranking: $\textsc{Parse}(|\mathrm{op_{rel}}|) >> \textsc{Parse}(|\mathrm{restr}|) >> |\mathrm{op_{rel}}|A|\mathrm{restr}| >> |\mathrm{op_{rel}}|F\mathrm{x} >> |\mathrm{restr}|P\mathrm{D_{REL}}$

	$\textsc{Parse}(\lvert\mathrm{op_{rel}}\rvert)$	$\textsc{Parse}(\lvert\mathrm{restr}\rvert)$	$\lvert\mathrm{op_{rel}}\rvert A\lvert\mathrm{restr}\rvert$	$\lvert\mathrm{op_{rel}}\rvert F\mathrm{x}$	$\lvert\mathrm{restr}\rvert P\mathrm{D_{REL}}$
$\lvert\mathrm{restr}\rvert\ \lvert\mathrm{op_{rel}}\rvert$ x				*!	
$\lvert\mathrm{restr}\rvert$ x $\lvert\mathrm{op_{rel}}\rvert$			*!		
☞ x $\lvert\mathrm{restr}\rvert\ \lvert\mathrm{op_{rel}}\rvert$					*
x $\lvert\mathrm{op_{rel}}\rvert$		*!			
$\lvert\mathrm{restr}\rvert$ x	*!				

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 $|\mathrm{op_{rel}}|F\mathrm{x}$ in the example) is the alignment constraint responsible for the placement of arguments. First, Hungarian is shown where the $|\mathrm{restr}||\mathrm{op_{rel}}|$ 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 |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.

3.2.1 Hungarian

In Hungarian, relative pronouns are usually found preceding the relative clause as (39) shows.

(39)
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| $|\text{op}_{\text{rel}}|$ 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 $|\text{op}_{\text{rel}}|$is situated there. Normally, however the Theme argument (with neutral interpretation) would have a place following the subject.

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

The relative pronoun cannot appear in that place, though. (41) is ungrammatical regardless of what interpretation the preceding element (*János*) might have.

(41) *a könyv, János amit olvas

This leads to the conclusion that the $|\text{op}_{\text{rel}}|$ element must satisfy a higher-ranked constraint by appearing at the front the clause. In other words, this relative clauses seem to be the result of the constraint interaction shown by (36).

There is, however, another relative construction documented by Nádasdy (2011) where the relative clause precedes the modified noun instead of following it.[10] And, only in this case, it is possible for a focused or a topicalised element to precede the relative pronoun as is shown in (42).

(42) JÁNOS amit olvas könyv

[10]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

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 (38). 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.

3.2.2 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 $|\text{op}_{\text{rel}}|$ as can be seen from (43).

(43) 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 $|\text{op}_{\text{rel}}|$ is associated with the empty word than the other way around. This way Japanese relative clauses can be described by a ranking similar to (34)

3.2.3 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.

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

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

Mandarin relatives are also underparsed ones, but here there is on overt element; most probably |restr|; thus, the ranking shown in (35) could be used to account for it.

(Li & Thompson, 1981: 580-1)

3.2.4 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.

(46) 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"

(47) 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

$|\mathrm{op}_{\mathrm{rel}}|$.

3.3 Possible constraint rankings in English

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 $|\mathrm{op}_{\mathrm{rel}}|A|\mathrm{restr}|$ conflict when the operator is a non-subject argument. In the case of subjects, however, the sPv 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 2.3 is that, among them, $|\mathrm{restr}|P\mathrm{D}_{\mathrm{REL}}$ 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 $|\mathrm{op}_{\mathrm{rel}}|$ 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.

3.3.1 Ranking for *wh*-relatives

For *wh*-relatives, the proposed ranking is shown in (48).

(48) |restr|PD_{REL} >> PARSE(|restr|) >> sPv >> PARSE(|op|) >> |op|A|restr|>> arg$F$v

This ranking functions much like the example shown in (36). 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.

(49)

	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 (50).

(50)

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

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|Pn (prepositions precede the noun they are associated with), the other |prep|Fv (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|Pn >> |prep|Fv, 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.

(51)

		|restr|PD_{REL}	PARSE(|restr|)	PARSE(|op_{rel}|)	|prep|Pn	|prep|Fv	|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.

(52) *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).

(53)

	$\|restr\|P\text{D}_{\text{REL}}$	Parse($\|$restr$\|$)	Parse($\|\text{op}_{\text{rel}}\|$)	$\|$prep$\|P$n	$\|$prep$\|F$v	$\|\text{op}_{\text{rel}}\|A\|$restr$\|$	Parse(di)
$\|$prep$\|$ $\|$restr$\|$ $\|\text{op}_{\text{rel}}\|$ V	*!				*		
$\|$restr$\|$ $\|$prep$\|$ $\|\text{op}_{\text{rel}}\|$ V					*	!*	
$\|$prep$\|$ $\|\text{op}_{\text{rel}}\|$ $\|$restr$\|$ V	**!				*		
$\|$restr$\|$ $\|\text{op}_{\text{rel}}\|$ V $\|$prep$\|$				*!			
$\|$restr$\|$V $\|$prep$\|$			*!				
☞ $\|\text{prep}\|_x$ $\|$restr$\|$ $\|\text{op}_{\text{rel}}\|$ V					*		*

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$_{\text{REL}}$.

3.3.2 Ranking for *that*-relatives

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

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

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 $|\text{op}_{\text{rel}}|$ CU underparsed as in (55). On the other hand, if it is not a subject, then $|\text{op}_{\text{rel}}|$ is simply situated behind the verb avoiding a violation of arg$F$v. This can be seen in (56).

(55)

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

(56)

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

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.

3.3.3 Ranking for zero relatives

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

(57) |restr|PD$_{\text{REL}}$ >> sPv >> 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.

(58)

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

(59)

	Input: |restr|, V(S, |op|)	|restr|PD$_{\text{REL}}$	sPv	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, sPv.

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 $|\text{op}_{\text{rel}}|$ 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.

Chapter 4

Further issues

This chapter addresses questions that remain unanswered by the general analysis presented in the previous chapter. The aim of this chapter is not to provide exhaustive explanation of those issues, but rather to point out how they can possibly be integrated into the proposed framework and to pinpoint the areas in which further research is needed. These questions fall into two large categories. First additional types of relatives are discussed that show a varied picture with respect to what degree of variation they allow. Then, it is shown how a system involving multiple rankings might be integrated into a grammar. There, a number of theoretical devices and models are reviewed to find a model which is the best suited for the analysis presented here.

4.1 Other kinds of relative clauses

This section introduces three kinds of restrictive relative clauses whose analysis seems to require additional assumptions than the ones made above. First headless relatives are introduced where there is no noun the clause modifies. Second, clauses with a possessive *wh*-element are examined. Third, clauses with other *wh*-elements (e.g. *where, when*) are shown.

4.1.1 Headless relatives

Headless relatives lack a head noun to which the relative clause refers to as in (60).

(60) Harry accepted what the others said.

Following Quirk et al. (1985), it is assumed here that not only clauses introduced with *what* are seen as headless relatives, but other *wh*-clauses with no antecedent noun, as well.

(61) a. We were standing on the platform when the train arrived.

b. There are great pubs where Bob lives.

Relatives like this are interesting for the analysis as they do not show any kind of variation; a *wh*-element always has to precede the rest of the relative as shown by (62).

(62) *Harry accepted (that) the others said.

In terms of the proposed analysis, that means that the input for headless relatives must perform in such a way on each of the proposed rankings that the same sequence emerges as winner in all cases. In other words, there must be a particular configuration of CUs specific to this case that reduces the possible outcomes to one.

It is argued here that the deciding factor here is that |restr| is aligned with respect to the verbal root in the main clause, and everything else in the relative domain is aligned with respect to |restr|. For instance, |restr| in the example is an argument of the main verb; therefore, can be topicalised while other kinds of relatives cannot.

(63) a. What the others said, Harry accepted.

b. *That the others said, Harry accepted the things.

Another issue that needs answering is where $|\mathrm{op_{rel}}|$ can take its reference from. Assume that in the cases when there is no root element to refer to, $|\mathrm{op_{rel}}|$ is applied to |restr|. Also keep in mind that the $|\mathrm{op_{rel}}|$ element is visible to the constraints that apply to the CU $|\mathrm{op_{rel}}|$ refers to. Thismeans that if $|\mathrm{op_{rel}}|$ refers to |restr|, then $|\mathrm{op_{rel}}|$ has to satisfy $|\mathrm{restr}|P\mathrm{D_{REL}}$, as well. Thus, just like in the case of pied-piping, the winning candidate will have an element with an underparsed domain index preceding |restr|. Therefore, |restr| and $|\mathrm{op_{rel}}|$ always precede the relative domain no matter how the other constraints are ranked. Also an additional functional CU that expresses thematic is aligned with respect to |restr|; for instance, it is |theme| in the previous example. It is argued here that the sequence of these three functional CUs is what is replaced with a *wh*-word.

To summarise the two key elements in the case of headless relatives are $|\mathrm{op_{rel}}|$ referring to |restr|, and |restr|being associated with the verbal root of the main clause.

4.1.2 Possessive relatives

The case of possessive relatives might seem similar to that of headless ones since they also have a compulsory *wh*-element as in (64).

(64) a. the man whose car I borrowed
b. *the man that car I borrowed
c. *the man car I borrowed

As far as the analysis is concerned, however, there is a significant difference. Namely, that here the relative is not related to the main verb as it was in the case of headless relatives. To account for this phenomenon at least two assumptions have to made. First, one has to assume that $|\mathrm{op_{rel}}|$ in the input is actually applied to a sequence of a root element and a |poss| CU that has the meaning "possessor" and is in most cases replaced with *'s* when adjacent to a root. This also means that $|\mathrm{op_{rel}}|$ is visible to constraints which that target the |poss| CU. Therefore, it is understandable, why the possessed element (*car* in the example) is situated as close to the left edge of the relative domain as possible. The explanation for this is that there must be both an precedence and an adjacency constraint that require the possessor both to precede the possessed and to be as close to is as possible. It is less obvious, though, why cannot the sequence of these two elements be left following the verb. In fact, this issue probably requires the introduction of a further constraint that is sensitive only to a proper subset of $|op_{rel}|$elements. Thus, the thorough investigation of this matter is subject of further research.

4.1.3 Other *wh*-elements

As it was already demonstrated in section 4.1.1, headless relatives appear with a number of *wh*-words. In that respect, relative clauses with a nominal antecedent are no different as (65) shows.

(65) a. the house where Bob was born
b. the day when Ann arrived
c. the reason why they became angry
d. the way how we approach the problem

Quirk et al. (1985) demonstrate, there are actually eight patterns in which sequences like above can appear.

(66) a. the day on which she arrived
b. the day which she arrived on
c. the day that she arrived on
d. the day she arrived on

e. the day when she arrived

f. the day she arrived

g. the day that she arrived

h. when she arrived

(Quirk et al., 1985: 1254)

Note that the first four types were already discussed in section 3.3. Also note that these four correspond to the following three kinds (from (66-e) to (66-g)). The reason why this correspondence is not one-to-one is the fact that both (66-a) and (66-b) correspond to (66-e) as the former ones include a preposition which can either be stranded or pied-piped. Apart from that, one can conclude that, because the same pattern is seen here, this phenomenon can be accounted for in the same way as other relatives.

It is assumed here that the only difference lies in what $|\mathrm{op}_{\mathrm{rel}}|$ refers to in the input. Without going into details into how the prepositions associated with adverbials are represented in the input, suppose that it involves a single |adv| element (which is a general representation for the different kinds of adverbials) which is replaced with a preposition during lexical insertion. Thus, in the relatives in the example, the operator might be present in two different configurations in the input. First, it can be applied only to the root element the relative refers to. Second, it can be applied to a sequence of the root and |adv|; thus, it will correspond to a lexical item which also incorporates the meaning of |adv|.

In conclusion, this section demonstrated that relatives involving a wider range of *wh*-words actually conform to the analysis presented in section 3.3, the only difference is what $|\mathrm{op}_{\mathrm{rel}}|$ creates a reference to.

4.2 Implementation of the alignment-based analysis

The final question one has to consider is how an Optimality Theoretic Grammar with multiple rankings can function. The first chapter has shown that in the framework of Alignment Syntax the only choice is to work with multiple rankings when accounting for variation. This section provides a brief review of three models: the Multiple Grammars Model, Stochastic Optimality Theory, and Maximum Entropy Grammars.

First, the assumption that the theory of Multiple Grammars makes is rather straightforward. According to McCarthy (2008), this theory assumes that the speaker of a language has multiple Eval components. More precisely Eval has access to all the possible rankings of constraints. That is, there is a probability value associated with the rankings. The problem with this approach, as Kager (1999) points out, is that of learnability. In other words, it would be a complicated task to account for how a learner is able to acquire all the possible rankings.

A very similar approach is formulated by Stochastic Optimality Theory. This model was designed to address the problem that OT's learning algorithm, Recursive Constraint Demotion, fails when given data that involves variation (Boersma, 1999). In Stochastic OT, constraints are not ordered, but each constraint has a rank, which is a positive real number. During the evaluation, the rank of each constraint is multiplied with a noise factor; a random number that is normally distributed between zero and one, and the ranking for the evaluation will be based on the result. With the inclusion of this probabilistic element, it is possible to have different rankings for each different evaluation. However, since the noise factor is selected from a normal distribution, the probability of a constraint with a very low rank ending up above a constraint with a very high rank is fairly small. A problem with Stochastic OT is that its learning algorithm is not guaranteed converge even when there is a possible ranking to account for the data set (Jäger, 2003).

The final model introduced here is Maximum Entropy Grammars. This model is not, strictly speaking, Optimality Theoretic as it operates only with markedness constraints.

In this model, the probability of certain forms appearing is calculated. Again, there are no ordered constraints; constraints only have a weight-value that is taken into account in calculating the probability: the higher the weight of the violated constraint, the less probable is the given surface form. This also means that multiple violations of a constraint with low weight can produce the same result as the single violation of a constraint with high weight; that is, weaker constraints can "gang up" on stronger ones. The biggest advantage of this model is that it can produce the closest correlation with data collected from native speakers (Hayes & Wilson, 2008) and has a learning algorithm that always converges (Jäger, 2003). However, it seems less appropriate to be used with alignment-based constraints as the number of violations of the low-ranked adjacency constraints may depend on the length of the sentence. That is, if the sentence has more modifiers, and, as a result, there are more intervening elements between hosts and target, the adjacency constraints become stronger and stronger because of the larger number of violations. Naturally, it would be undesirable for the grammar to handle shorter sentences differently form longer ones; therefore the usefulness of this model with an alignment-based grammar is questionable.

In conclusion, it seems that the Multiple Grammars and Stochastic OT models are better suited to integrate the analysis proposed by this paper. Among those two, it is Stochastic OT that seems to be a more usable and psychologically more valid option. Therefore, further research is needed to investigate whether the assumptions made here can be integrated into a stochastic analysis.

Conclusion

This paper has explored the possibility of including restrictive relative clauses in an Alignment Syntactic grammar of English. In the first chapter, the nature of the input was discussed, and it was shown what kind of input elements are needed for relative clauses. The two most important criteria were that (i) the input must be the same for all kinds of relatives as it is the locus of semantic interpretation, and (ii) the input must consist of abstract CUs as syntax operates before the insertion of lexical items. It was proposed that the input for relatives includes two specific CUs: $|\mathrm{op}_{\mathrm{rel}}|$ and |restr|, which, depending on their configuration can be spelled out as a *wh*-element, *that*, or not spelled out at all.

The second chapter showed the types of constraints that Alignment Syntax uses. Also, the alignment and faithfulness constraints that are active only in relative clauses were introduced. The third chapter made use of these constraints when proposing three different rankings to account for the three different types of relatives. Also, a brief cross-linguistic typology was provided to show that the three ways of constraint conflict resolution used in English are active in other languages, as well.

Finally, the fourth chapter discussed some of the more problematic cases such as headless and possessive relatives and the appearance of a range of *wh*-words. Moreover, this chapter introduced three possible ways to integrate multiple rankings within the same grammar, and it concluded that Stochastic OT is probably the best fit for the alignment-based analysis of English restrictive relative clauses.

References

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

Boersma, P. (1999): "Optimality-Theoretic learning in the Praat program". ROA Report #380 available at: `http://roa.rutgers.edu/files/380-0200/roa-380-boersma-1.pdf`

Broihier, K. (1995): "Optimality Theoretic rankings with tied constraints: Slavic relatives, resumptive pronouns and learnability". ROA Report #46 available at: `http://roa.rutgers.edu/files/46-0295/46-0295-BROIHIER-0-0.PDF`

Chomsky, N., & Lasnik, H. (1977). "Filters and Control". *Linguistic Inquiry, 8*(3), 425-504.

de Lacy, P. (2011). "Markedness and faithfulness". In: van Oostendorp, M., Ewen, C. J., Hume, E. & Rice, K. (eds) *The Blackwell Companion to Phonology.* (1491-1512). Malden: Blackwell.

Gáspár, M. (2004). "Topic universality in OT inputs". In László Varga (ed.) *The Even Yearbook 6.* (60-73). Budapest: ELTE SEAS.

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

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

Grimshaw, J. (1998). "Constraints on constraints in Optimality Theoretic syntax".

ms. Rutgers.

Haegeman, L. (1994) *Introduction to Government and Binding Theory.* Malden: Blackwell.

Halle, M.& Marantz, A. (1993). “Distributed Morphology and the Pieces of Inflection”. In: Hale, K. & Keyser, S. J. (eds) *The View from Building 20.* (111-176). Cambridge, MA: MIT Press.

Hayes, B. & Wilson, C. (2008). “A Maximum Entropy model of phonotactics and phonotactic learning” *Linguistic Inquiry, 39*(3), 379-440.

Jäger, G. (2003). ”Maximum Entropy models and Stochastic Optimality Theory”. ROA Report #625 available at: `http://roa.rutgers.edu/files/625-1003/625-JAEGER-0-0.PDF`

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

Kaiser, S., Ichikawa, Y., Kobayashi, N. & Yamamoto, H. (2001). *Japanese: a comprehensive grammar.* London: Routledge.

Kántor, G. (2003). “Sentence structure versus no structure”. In: Kiss, Z. (ed) *The Odd Yearbook 7.* Budapest: ELTE SEAS.

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

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

Kenesei, I., Vago, R. M. & Fenyvesi, A. (1998) *Hungarian.* London: Routledge.

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

Legendre, G. (2001). “Introduction to Optimality Theory in syntax”. In: Legendre, G., Grimshaw, J. & Vikner, S. (eds) Optimality Theoretic Syntax. Cambrdige, MA: MIT Press.

Li, C. N. & Thompson, S. A. (1981): *Mandarin Chinese: a functional reference* *grammar*. Berkeley: University of California Press.

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

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

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

Newson, M. (2000). “A restrictive theory of alignment and some of its consequences”. ms. Eötvös Loránd University

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

Newson, M. (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, M. (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, M. (2014). “Alignment Syntax”. Course taught at Eötvös Loránd University, Budapest.

Newson, M. & Maunula, V. (2006). “Word order in Finnish: Whose side is the focus on?”. In: Varga, L. (ed.) *The Even Yearbook 7*. Budapest: ELTE SEAS.

Newson, M. & Szécsényi, K. (2012). “Dummy auxiliaries and late vocabulary insertion”. In: Varga, L. (ed.) *The Even Yearbook 10*. Budapest: ELTE SEAS.

Partee, B. H., Meulen, A. T., Wall, R. E. (1987). *Mathematical methods in linguistics*. Dodrecht: Kluwer.

Prince, A. and Smolensky, P. (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`

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A comprehensive grammar of the English language*

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

Starke, M. (2009): "Nanosyntax – A short primer to a new approach to language". In: Svenonius, P., Ramchand, G., Starke, M., & Tarald, K. (eds): Nordlyd 36.1