

Abduction and Preferences in Linguistics

Kathrin Konczak and Ralf Vogel

{Institut für Informatik, Institut für Linguistik},
Universität Potsdam,
Postfach 90 03 27, D-14439 Potsdam,
{konczak@cs.uni-potsdam.de rvogel@ling.uni-potsdam.de }

Abstract. We associate optimality theory with abduction and preference handling. We present linguistic problems that appear in the study of dialects as new application of abduction and preference handling. Differences in German dialects originate from different rankings of linguistic constraints which determine the well-formedness of expressions within a language. We introduce a framework for analyzing differences in German dialects by abduction of preferences. More precisely, we will take the perspective of a linguist and reconstruct dialectal variation as abduction problem: Given an observation that a sentence is found as grammatically correct, abduct the underlying constraint ranking. For this, we give a new definition for the determination of optimal candidates for total orders with indifference. Additionally, we give an encoding for the diagnosis front-end of the DLV system.

1 Introduction

During the last decade, Answer Set Programming (ASP) [9] has become an increasingly acknowledged tool for knowledge representation and reasoning. A main advantage of ASP is that it is based on solid theoretical foundations, while being able to model different techniques from knowledge representation, e.g. abduction and preference handling, in an arguably satisfactory way.

The role of abduction [3, 11] has been demonstrated in a variety of applications, e.g. in diagnosis of electrical circuitry [5], air crew-assignment [3], and database updates [12]. One of the most widely-used definitions of abduction is the generalized stable model semantics [13]. Given a logic program Π , a set of atoms A standing for abducibles, and an observation \mathcal{O} , an explanation Δ is a subset of A such that there is an answer set of $\Pi \cup A$ which satisfies \mathcal{O} .

The modeling of preferences is another line in ASP research, which has been intensively studied in the last years, cf. [2]. Strongly rooted in the research of non-monotonic formalisms, the ability to specify preferences is acknowledged to be particularly beneficial to ASP, since they constitute a very natural and effective way of resolving indeterminate solutions. For example, preferences have been successfully used for timetabling, auctioning, configuration and other domains.

In this work we want to find preference structures in a linguistic framework as explanations of an abduction problem. A linguistic grammar is a model of the implicit knowledge that guides linguistic behavior. This knowledge is usually conceived as a system of

rules and/or well-formedness constraints which determine for a given language which expressions are well-formed and which are not. Language particular knowledge (grammars of individual languages) thereby has to be distinguished from knowledge about languages in general (universal grammar). In the grammar theoretical framework of Optimality Theory (OT), the set of well-formedness constraints is universal, while their importance varies from language to language.

OT constraints are violable and ranked relative to each other. The effect of a constraint violation depends on the rank of the constraint. OT models grammar as a competition of candidate expressions on the constraint hierarchy – relative to a given input which determines what is to be expressed. The candidate that performs best on the constraint hierarchy is the grammatical expression, all others are losers and therefore impossible as expressions for the input. Hence, learning a language can be understood as inferring the underlying constraint ranking from observations.

The linguistic example that we will be dealing with is dialectal variation in the word order possibilities of German 3-verb clusters, as discussed in [14] within the OT framework. An example of a 3-verb cluster is the group of verbs underlined in the German sentence below:

Maria glaubt, dass sie das Lied singen müssen wird.

Maria thinks that she the song sing must will.

Standard German and Swiss German variants differ in their default orders for verb clusters of this type (as well as further non-default ordering possibilities which will briefly be discussed in Section 4):

	Default verb cluster orders
Standard German:	<i>singen müssen wird</i>
Swiss German:	<i>wird müssen singen</i>

In this paper, we will take the perspective of the linguist, and reconstruct the dialectal variation in German as abduction problem: Given the observation of a particular verb order, abduct the underlying constraint ranking.

In Section 2, we recall basic definitions of answer sets, abduction, order relations, and linguistics. In Section 3, we reconsider optimal candidates and elaborate on the above example, for which we then give an implementation within ASP in Section 4. Furthermore, we develop in Section 4 a new definition for optimal candidates for orders with indifferences. After presenting the results on our example, we draw conclusions and discuss further research issues in Section 5.

2 Background

Answer Sets. A (normal) logic program is a finite set of rules such as $p_0 \leftarrow p_1, \dots, p_m, \text{not } p_{m+1}, \dots, \text{not } p_n$, where $n \geq m \geq 0$, and each p_i ($0 \leq i \leq n$) is an atom. For such a rule r , we let $\text{head}(r)$ denote the head, p_0 , of r and $\text{body}(r)$ the body, $\{p_1, \dots, p_m, \text{not } p_{m+1}, \dots, \text{not } p_n\}$, of r . Let $\text{body}^+(r) = \{p_1, \dots, p_m\}$ and $\text{body}^-(r) = \{p_{m+1}, \dots, p_n\}$. For a set of rules Π , we write $\text{head}(\Pi) = \{\text{head}(r) \mid r \in \Pi\}$. A program is basic if $\text{body}^-(r) = \emptyset$ for all its rules. The reduct, Π^X , of a program Π relative to a set X of atoms is defined by $\Pi^X = \{\text{head}(r) \leftarrow \text{body}^+(r) \mid$

$r \in \Pi$, $body^-(r) \cap X = \emptyset$. A set of atoms X is *closed under* a basic program Π if for any $r \in \Pi$, $head(r) \in X$ if $body^+(r) \subseteq X$. The smallest set of atoms being closed under a basic program Π is denoted by $Cn(\Pi)$. Then, a set X of atoms is an *answer set* of a program Π if $Cn(\Pi^X) = X$. The set of all answer sets of Π is denoted by $AS(\Pi)$.

Abduction (within answer set programming.) An abductive framework [8, 3, 6] is a triple $\langle \Pi, H, O \rangle$, where Π is a logic program, H is a set of facts, referred to as *hypotheses*, and O is a set of atoms, referred to as *observations*. A set $\Delta \subseteq H$ is an *explanation* of O wrt Π if there is an answer sets of $\Pi \cup \Delta$ (brave reasoning) which contains O . Note that in the literature some definitions of abduction are based on cautious reasoning, i.e. O must be contained in *all* answer sets of $\Pi \cup \Delta$. In this work, we concentrate on brave reasoning, since we use the diagnosis front-end of the DLV system [4, 5].

Order relations. Let R be a binary relation on a set \mathcal{C} . We say that R is reflexive if $R(x, x)$ holds for all $x \in \mathcal{C}$. R is transitive if $R(x, y)$ and $R(y, z)$ implies $R(x, z)$ for all $x, y, z \in \mathcal{C}$. R is antisymmetric if $R(x, y)$ and $R(y, x)$ implies $x = y$ for all $x, y \in \mathcal{C}$. R is asymmetric if for all $x, y \in \mathcal{C}$ we do not have both $R(x, y)$ and $R(y, x)$. R is total (or complete) if we have either $R(x, y)$ or $R(y, x)$ for all $x, y \in \mathcal{C}$. We say that R is a *total order* if R is reflexive, transitive, total, and antisymmetric. Additionally, R is a *strict total order* if R is asymmetric, transitive, and total. Whenever R is a total order we use the notation $y \preceq x$ instead of $R(x, y)$, where $y \preceq x$ means that x is higher or equal preferred to y . Whenever R is a strict total order we use the notation $y \prec x$ instead of $R(x, y)$, where $y \prec x$ means that x is (strictly) higher preferred than y .

Linguistic problems. Linguists use observations of two kinds to find out the set of well-formed expressions of a given language. One method is corpus-based, structures which appear more than rarely can be assumed to be well-formed. The other method is explicit elicitation, speakers of a language are asked to give a well-formedness judgment on particular constructed examples.

The task of the linguist is similar to the one of the language learner: figure out the underlying system of constraints for a language, based on observations.

Our example is the verb order in 3-verb clusters of German dialects given in Section 1 with the following different ordering strategies for the verbs:

Default verb cluster orders	
Standard German:	<i>singen müssen wird</i>
Swiss German:	<i>wird müssen singen</i>

The relative order of object noun phrases and their governing verbs does not differ, however: the object follows the verb in all German dialects, contrary to, e.g., English:

Default object-verb orders	
German:	<i>ein Lied singen</i>
English:	<i>to sing a song</i>

Syntactic structures are composed recursively by *complementation*. The object, here: “*ein Lied*” is the complement of its governing head, here: the predicative verb “*singen*”. This verb phrase, “*ein Lied singen*”, is the complement of the modal verb “*müssen*”, and this modal verb phrase, in Standard German: “*ein Lied singen müssen*”, is the comple-

ment of the temporal auxiliary verb, “*werden*”. The differences between the languages and variants can now be described in terms of complement-head order:

	Default complement-head orders
Standard German:	All complements precede their heads: “ <i>ein Lied singen müssen wird</i> ”
Swiss German:	Noun complements precede their heads, verbal complements follow them: “ <i>wird müssen ein Lied singen</i> ”
English	All complements follow their heads: “ <i>will have to sing a song</i> ”

The differences between the three languages can be reconstructed within OT using the following three constraints:

H-Comp A complement follows its head.

Comp-H A complement precedes its head.

H-VComp A verbal complement follows its head.

Constraint rankings are indicated with “ \gg ”, meaning “has higher priority than”. The three rankings that conform to the observations are the following:

Standard German: $Comp-H \gg H-Comp (H-VComp)$

Swiss German: $H-VComp \gg Comp-H \gg H-Comp$

English: $H-Comp \gg Comp-H (H-VComp)$

The exact rank of $H-VComp$ can only be determined in Swiss German. While its effects are completely subsumed by the high rank of $H-Comp$ in English, in Standard German all that is necessary is that $Comp-H$ has highest priority, while the relative order of $H-Comp$ and $H-VComp$ is irrelevant because of the low rank of these two constraints. Grammars are usually, but not necessarily, strict total orders of constraints. The rankings given here are only the crucial ones. For those which are left open, any order will be compatible with the observations.

3 Optimal candidates

A linguistic grammar predicts the well-formedness of expressions. OT grammars do so by establishing a competition between different candidate expressions which are evaluated on a hierarchy of well-formedness constraints like $H-Comp$, $Comp-H$, and $H-VComp$. An OT grammar is an input-output mapping. The input defines what is to be expressed. We then have a set of candidate output expressions. The candidates incur different constraint violations. Each candidate is evaluated on the basis of the constraint hierarchy, and the candidate that performs best in this evaluation is the winner, the optimal, hence, grammatical expression.

In the following, we will consider the determination of optimal candidates wrt well-formedness of expressions. For this, let \mathcal{X} be a set of candidates (sentences), \mathcal{C} be a set of constraints, $\delta : \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{N}$ be a violation function, where $\delta(x, c)$ denotes the degree of violation of $x \in \mathcal{X}$ wrt $c \in \mathcal{C}$, and \ll be a strict total order on \mathcal{C} . Then, we call $\mathcal{L} = (\mathcal{X}, \mathcal{C}, \delta, \ll)$ a linguistic framework.

In Section 1 and 2, we have taken an example for the dialectal variation in German 3-verb clusters. In the following we will elaborate this example. For the 3-verb clus-

ter with the verbs $\{wird, m\u00fcssen, singen\}$ (“will, must, sing”), we have the following possible word orders:¹

Maria glaubt, dass sie das Lied ...

(321) singen m\u00fcssen wird. (231) m\u00fcssen singen wird.

(123) wird m\u00fcssen singen. (132) wird singen m\u00fcssen.

(312) singen wird m\u00fcssen. (213) m\u00fcssen wird singen.

Our set of candidates $\mathcal{X} = \{321, 231, 123, 132, 312, 213\}$ is constituted by these possible word orders.² $\mathcal{C} = \{H-VComp, Comp-H, H-Comp\}$ is our set of constraints. The degree of violation denotes how well a sentence fulfills a constraint. E.g. for constraint $H-VComp$ the best order is where the auxiliary verb (“wird”) precedes the modal verb (“m\u00fcssen”), which precedes the predicative verb (“singen”). Candidate 123 has this order. Candidate 132 violates this constraint once, since “m\u00fcssen” comes after “singen”, and candidate 321 violates $H-VComp$ three times, since the verbs are in the reverse order. The violation degrees for all constraints and candidates are given in the following table, where the degree of violation is represented by the number of asterisks *.

	$H-VComp$	$Comp-H$	$H-Comp$
321	***		**
231	**	*	*
123		**	
132	*	*	*
312	**	*	**
213	*	**	*

Next, we want to clarify when a sentence is a best candidate wrt a given constraint ranking [1].

Definition 1. Let $\mathcal{L} = (\mathcal{X}, \mathcal{C}, \delta, \ll)$ be a linguistic framework, where \ll is a strict total order on \mathcal{C} .

Then, candidate $x \in \mathcal{X}$ is a winner if there does not exist a candidate $y \in \mathcal{X}$, $x \neq y$ such that there exists a constraint $c \in \mathcal{C}$ where

1. for all $c' \in \mathcal{C}$ where $c \ll c'$ we have that $\delta(x, c') \geq \delta(y, c')$,³ and
2. $\delta(x, c) > \delta(y, c)$.

Candidate x is a winner, also called *optimal* candidate, if there does not exist another candidate y who is better than x . That is the case when there exists a constraint, where y has a lower degree of violation than x (Condition 2) and when y behaves better for all higher constraints (Condition 1).

In our example, for the constraint order $H-VComp \gg Comp-H \gg H-Comp$ (Swiss German), we get that 123 is a winner. There, for the highest constraint $H-VComp$,

¹ The numbers signal the hierarchical position of the verb. Verb 1 is the temporal auxiliary (*werden*), verb 2 the modal verb (*m\u00fcssen*), and verb 3 the predicative verb (*singen*).

² Due to OT we have to consider all possibilities.

³ For a better understanding we use the condition $\delta(x, c') \geq \delta(y, c')$ instead of $\delta(x, c') = \delta(y, c')$. Requiring $\delta(x, c') = \delta(y, c')$ instead of $\delta(x, c') \geq \delta(y, c')$ would select the crucial constraint c , which is not requested here, and is also sufficient for characterizing optimal candidates.

all other candidates have a higher degree of violation and are hence worse than 123. For constraint ranking $Comp-H \gg H-VComp \gg H-Comp$ (observed in Standard German), candidate 321 is a winner, since for the highest constraint $Comp-H$ all other candidates are worse than 321. Hence, we get the verb cluster *singen müssen wird* as winner for the Standard German dialect and *wird müssen singen* as winner for the Swiss German dialect.

4 Abduction of constraint rankings

In Optimality theoretic terms, linguists observe that a candidate is determined by a speaker as a *winner*, expressing that the sentence is grammatically correct. The problem is that the observer does not know which underlying constraint ranking the speaker has. Here, abduction comes into play.

Given a linguistic framework $\mathcal{L} = (\mathcal{X}, \mathcal{C}, \delta)$ with an unknown constraint ranking \ll and an observation that candidate $x \in \mathcal{X}$ is a winner, we want to abduct the constraint ranking \ll which explains x . For this, our set of hypotheses is the set of all possible (pairwise) constraint rankings. Then, the explanations give us possible strict total orders such that x is optimal. In the following we will give an encoding for this.

The set of candidates is given by rules (1) $cd(x) \leftarrow$ for each $x \in \mathcal{X}$, the set of constraints by (2) $cst(c) \leftarrow$ for each $c \in \mathcal{C}$, and the violation degrees by (3) $viol(x, c, \delta(x, c)) \leftarrow$ where $\delta(x, c)$ is the degree of violation of $x \in \mathcal{X}$ wrt $c \in \mathcal{C}$.

According to Definition 1, a winner, can be determined by the following rules:

- (4) $winner(X) \leftarrow cd(X), not\ defeated(X)$
- (5) $defeated(X) \leftarrow cd(X), cd(Y), Y \neq X, better(Y, X)$
- (6) $better(Y, X) \leftarrow cd(X), cd(Y), Y \neq X, cst(C),$
 $wins(Y, X, C), not\ hp(X, Y, C)$
- (7) $hp(X, Y, C) \leftarrow cd(X), cd(Y), Y \neq X, cst(C), pref(C_1, C), wins(X, Y, C_1)$
- (8) $wins(X, Y, C) \leftarrow cd(X), cd(Y), cst(C),$
 $viol(X, C, NX), viol(Y, C, NY), NX < NY$
- (9) $pref(X, Z) \leftarrow pref(X, Y), pref(Y, Z)$
- (10) $\leftarrow pref(C, C), cst(C)$
- (11) $\leftarrow cst(C_1), cst(C_2), unranked(C_1, C_2), C_1 \neq C_2$
- (12) $unranked(C_1, C_2) \leftarrow not\ pref(C_1, C_2), not\ pref(C_2, C_1), cst(C_1), cst(C_2)$

Rules (4)–(8) encode Definition 1, where $winner(x)$ is derived whenever x is not defeated, i.e. there exists no candidate $y \neq x$ which is better than x . The relation that y is better than x (Condition 1 and 2 in Definition 1) is encoded by Rules (6)–(7). Rule (8) only encodes the expression $\delta(x, c) < \delta(y, c)$. Rule (9) is inserted for encoding the transitivity relation⁴ of preferences and Rules (10)–(12) ensure that the preference ordering is strict and total. Our logic program Π consists of the rules (1)–(12). An observation is that exactly one candidate is observed as a winner, but the other candidates are not. Hence, for $x \in \mathcal{X}$ we have

$$O(x) = \left\{ \begin{array}{l} winner(x) \leftarrow \text{for } x \in \mathcal{X} \\ defeated(y) \leftarrow \text{for all } y \in \mathcal{X}, y \neq x \end{array} \right\}$$

⁴ Note that this is useful as simplification for formulating the hypotheses.

This is due by the fact that we want to observe unique optimal winners [1]. Our hypotheses are the set of all possible pairwise preference relations:

$$H = \{pref(c, c') \leftarrow \mid c, c' \in \mathcal{C}, c \neq c'\}.$$

Then, an explanation $\Delta \subseteq H$ for the abduction problem $\langle \Pi, H, O \rangle$ gives us a possible strict total order among the constraints such that $\Pi \cup \Delta$ explains O . More precisely, $\Delta \subseteq H$ is an explanation if $O(x) \subseteq S$ for some $S \in AS(\Pi \cup \Delta)$.

In our linguistic example, we observe that a sentence is a winner and want to abduct possible rankings among the set of constraints. Additionally, we have the background knowledge that constraint *Comp-H* is strictly higher preferred than *H-Comp*. This is due to the observation that object noun phrases precede their governing verbs in all German dialects. Hence, we have additionally in Π the fact

$$pref(comp_h, hcomp) \leftarrow .$$

Then, our hypotheses are

$$H = \left\{ \begin{array}{ll} pref(comp_h, hvcomp) \leftarrow & pref(hvcomp, hcomp) \leftarrow \\ pref(hvcomp, comp_h) \leftarrow & pref(hcomp, hvcomp) \leftarrow \end{array} \right\}$$

which gives us together with $pref(comp_h, hcomp) \leftarrow$ all possible constraint rankings. For computing explanations, we use the DLV system [4, 5] with the command `-FD` for abductive diagnosis.

x	Explanations
321	$\Delta_1 = \{Comp-H \gg H-Comp \gg H-VComp\}$ $\Delta_2 = \{Comp-H \gg H-VComp \gg H-Comp\}$
231	no explanations
123	$\Delta = \{H-VComp \gg Comp-H \gg H-Comp\}$
132	no explanations
312	no explanations
213	no explanations

Table 1. Explanations for $Comp-H \gg H-Comp$ and strict total orders

Table 1 lists all possible explanations (strict total orders of the set of constraints) for observation $O(x)$. Whenever a candidate has no explanation, this means that the constraints are not sufficient for explaining these candidates as a winner. In [14], further, more special cases were considered, e.g. where one of the verbs is focused, i.e., emphasized by stress. This opens additional ordering possibilities. Experiments for focus dependent orders will follow later in this section. In [14], observations lead to the conclusion that for Standard German *Comp-H* is ranked strictly higher than *H-VComp*. By additional analysis, the authors identified candidate 321 as the winner in Standard German with the underlying constraint ranking $Comp-H \gg H-VComp \gg H-Comp$. This is also found out by the method of abduction, i.e. by explanation Δ_2 for observation $O(321)$. Analogously, candidate 123 is the winner in Swiss German with the

underlying constraint ranking $H-VComp \gg Comp-H \gg H-Comp$ as stated by explanation Δ for $O(123)$. Interestingly, $O(321)$ is also explained by another constraint ranking, namely $\Delta_1 = \{Comp-H \gg H-Comp \gg H-VComp\}$. This confirms the fact that the relative ordering of $H-Comp$ and $H-VComp$ is irrelevant.

In the example, we have found out that observing candidate 321 yields two explanations, $H-Comp \gg H-VComp$ and $H-VComp \gg H-Comp$. This supposes that $H-Comp$ and $H-VComp$ can be ranked equally. For this reason, we want to abduct total orders (not necessarily strict). Since Definition 1 is only valid for strict total orders, we have to extend it for total orders.

Definition 2. Let $\mathcal{L} = (\mathcal{X}, \mathcal{C}, \delta, \preceq)$ be a linguistic framework, where \preceq is a total order on \mathcal{C} .

Then, candidate $x \in \mathcal{X}$ is a winner if there exists no $y \neq x$ such that there exists a $c \in \mathcal{C}$ such that

1. for all $c' \neq c$ such that $c' \approx c$ or $c' \succ c$ we have $\delta(c', x) \geq \delta(c', y)$, and
2. $\delta(c, y) < \delta(c, x)$.

This definition coincides for strict total orders with Definition 1. Condition 1 has been extended to handle equally ranked constraints. Now, a candidate y is better than x if there exists a constraint c such that for all equally ranked and for all higher ranked constraints, y behaves at least as good as x (Condition 1) and y is wrt c strictly better than x (Condition 2). Hence, candidate x is a winner if there exists no other candidate which is better than x . One major motivation for allowing equally ranked constraints in Optimality Theory is the observation of non-unique winners. Often, more than one expression is possible for a given input. That two candidates do not differ at a single constraint, is very unlikely. So under strict total constraint ranking, one should win over the other. Non-unique winners occur if two candidates differ with respect to two constraints where one constraint favors one, and the other constraint the other candidate. These two constraints must be ranked equally.

For example, let x, y, z be candidates, c_1, c_2 be constraints where $c_1 \approx c_2$ and let be given the following constellations for violation degrees:

Case 1:	Case 2:	Case 3:																														
<table style="margin: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 10px;"></td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_1</td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_2</td> </tr> <tr> <td style="padding: 0 10px;">x</td> <td style="padding: 0 10px; text-align: center;">*</td> <td style="padding: 0 10px; text-align: center;">**</td> </tr> <tr> <td style="padding: 0 10px;">y</td> <td style="padding: 0 10px; text-align: center;">**</td> <td style="padding: 0 10px; text-align: center;">*</td> </tr> </table>		c_1	c_2	x	*	**	y	**	*	<table style="margin: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 10px;"></td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_1</td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_2</td> </tr> <tr> <td style="padding: 0 10px;">x</td> <td style="padding: 0 10px; text-align: center;">*</td> <td style="padding: 0 10px; text-align: center;">**</td> </tr> <tr> <td style="padding: 0 10px;">y</td> <td style="padding: 0 10px; text-align: center;">*</td> <td style="padding: 0 10px; text-align: center;">*</td> </tr> </table>		c_1	c_2	x	*	**	y	*	*	<table style="margin: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 10px;"></td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_1</td> <td style="padding: 0 10px; border-bottom: 1px solid black;">c_2</td> </tr> <tr> <td style="padding: 0 10px;">x</td> <td style="padding: 0 10px; text-align: center;">**</td> <td style="padding: 0 10px;"></td> </tr> <tr> <td style="padding: 0 10px;">y</td> <td style="padding: 0 10px; text-align: center;">*</td> <td style="padding: 0 10px; text-align: center;">*</td> </tr> <tr> <td style="padding: 0 10px;">z</td> <td style="padding: 0 10px; text-align: center;">**</td> <td style="padding: 0 10px;"></td> </tr> </table>		c_1	c_2	x	**		y	*	*	z	**	
	c_1	c_2																														
x	*	**																														
y	**	*																														
	c_1	c_2																														
x	*	**																														
y	*	*																														
	c_1	c_2																														
x	**																															
y	*	*																														
z	**																															

In case 1 we get x and y as a winner. In case 2 we get only y as a winner and in case 3 we get x, y and z as a winner.

In the following, we will give an encoding of Definition 2 in ASP.

- (13) $winner(X) \leftarrow cd(X), not\ defeated(X)$
(14) $defeated(X) \leftarrow cd(X), cd(Y), Y \neq X, better(Y, X)$
(15) $better(Y, X) \leftarrow cd(X), cd(Y), Y \neq X, cst(C),$
 $wins(Y, X, C), not\ hp(X, Y, C)$
(16) $hp(X, Y, C) \leftarrow cd(X), cd(Y), Y \neq X, cst(C), C \neq C_1,$
 $prefeq(C_1, C), wins(X, Y, C_1)$
(17) $hp(X, Y, C) \leftarrow cd(X), cd(Y), Y \neq X, cst(C), pref(C_1, C), wins(X, Y, C_1)$
(18) $wins(X, Y, C) \leftarrow cd(X), cd(Y), cst(C),$
 $viol(X, C, NX), viol(Y, C, NY), NX < NY$
(19) $pref(X, Z) \leftarrow pref(X, Y), pref(Y, Z)$
(20) $pref(X, Z) \leftarrow pref(X, Y), prefeq(Y, Z)$
(21) $pref(X, Z) \leftarrow prefeq(X, Y), pref(Y, Z)$
(22) $prefeq(X, Z) \leftarrow prefeq(X, Y), prefeq(Y, Z)$
(23) $prefeq(X, Y) \leftarrow prefeq(Y, X)$
(24) $\leftarrow pref(C, C), cst(C)$
(25) $\leftarrow cst(C_1), cst(C_2), unrkcd(C_1, C_2), C_1 \neq C_2$
(26) $unrkcd(C_1, C_2) \leftarrow cst(C_1), cst(C_2),$
 $not\ pref(C_1, C_2), not\ pref(C_2, C_1), not\ prefeq(C_1, C_2)$

Rule (13)–(18) encode Definition 2. Rule (16) has been added compared to the encoding for Definition 1, which addresses the handling of equally ranked constraints. Rules (19)–(26) make, analogously to Rules (9)–(12), the preference relation, which is transitive and total explicit, where $prefeq(X, Y)$ denotes that the constraints X and Y are ranked equally and $pref(X, Y)$ expresses that x is ranked strictly higher than y . Hence, our logic program Π for our abduction problem now consists of the rules (1)–(3) and (13)–(26). Our hypotheses are now the set of all pairwise strict preference relations and the set of all possible indifferences.

$$H = \{pref(c, c') \leftarrow c, c' \in \mathcal{C}, c \neq c'\} \cup \{prefeq(c, c') \leftarrow c, c' \in \mathcal{C}, c \neq c'\}$$

From this set of hypotheses, all possible total orders are constructible. Coming back to our example, Table 2 shows us all explanations for observation $O(x)$ when abducting total orders (not necessarily strict) with the given background knowledge $pref(comp, hcomp) \leftarrow$. For candidate 321, we additionally get the supposed explanation that $H-VComp \approx H-Comp$.

Next, we want to extend our example to stress dependent non-default orders. While 123 is the default order in the St. Gallen (Swiss German) dialect, further orders are possible with particular stress patterns. In particular, order 213 (*müssen wird singen*, “must will sing”) is possible with main stress on verb 2, “*müssen*”, and order 312 is possible with stress on verb 3, “*singen*”. Along with these stress patterns comes a meaning change, contrastive focus. These expressions are only usable for particular purposes, e.g., where one wants to emphasize that she HAD TO sing the song, though she didn’t want to (order 213 with stress on verb 2), or that she had to SING the song, though she only wanted to sum it (order 312 with stress on verb 3).

In the following we will concentrate exemplarily on the St. Gallen (Swiss German) dialect. For our 3 verb cluster, the focus can either be on the modal, on the auxiliary, or

x	Explanations
321	$\Delta_1 = \{Comp-H \gg H-Comp \gg H-VComp\}$ $\Delta_2 = \{Comp-H \gg H-VComp \gg H-Comp\}$ $\Delta_3 = \{Comp-H \gg H-VComp \approx H-Comp\}$
231	no explanations
123	$\Delta = \{H-VComp \gg Comp-H \gg H-Comp\}$
132	no explanations
312	no explanations
213	no explanations

Table 2. Explanations for $Comp-H \gg H-Comp$ and total orders

on the predicative verb. The word order effect of focus is that the focused verb may be leftmost in the verb cluster. This is required by the focus constraints that we use here. Table 3 shows the violation degrees for the focus constraints [14], where $Foc(V)$ denotes stress on the predicative verb, $Foc(Mod)$ stress on the modal verb, and $Foc(Aux)$ stress on the auxiliary verb. For focus considerations the constraint $H-Comp$ is irrelevant.

	$Foc(V)$	$Foc(Mod)$	$Foc(Aux)$
321		*	*
231	*		*
123	*	*	
132	*	*	
312		*	*
213	*		*

Table 3. Violations of narrow focus for St. Gallen Swiss German

evant. Hence, we only have to consider the constraints $H-VComp$, $Comp-H$, and a constraint $Foc(\cdot)$ which evaluates whether the focus is leftmost. Additionally, we have the knowledge that $H-VComp \gg Comp-H$ holds, since we are in the Swiss German dialect.

Applying our abduction framework to this case yields the following results: 321, 231, and 132 have no explanations. They cannot be obtained as a winner for any focus considered here. 123 has $H-VComp \gg Foc(i) \approx Comp-H$ as explanation for all three foci, where $i \in \{V, Mod, Aux\}$. Additionally, 123 has the explanation $Foc(Aux) \approx H-VComp \gg Comp-H$ for focus on the auxiliary verb. Candidate 312 has an explanation only for focus on the predicative verb, i.e. $Foc(V) \gg H-VComp \gg Comp-H$. Candidate 213 has an explanation only for focus on the modal verb. That is $Foc(Mod) \gg H-VComp \gg Comp-H$.

Instead of abducting preference structures, one can abduct which dialect and underlying focus the speaker has. More precisely, we have knowledge about the constraint ranking for the different foci within Standard and Swiss German. Then, we observe a sentence as a winner. The question is then, which focus and which underlying dialect the native speaker has. The abduction problem $\langle \Pi, H, O \rangle$ is defined as follows: Our

hypotheses are the set of matters

$$H = \left\{ \begin{array}{ll} \text{matter}(\text{focVSG}) \leftarrow & \text{matter}(\text{focVStG}) \leftarrow \\ \text{matter}(\text{focModSG}) \leftarrow & \text{matter}(\text{focModStG}) \leftarrow \\ \text{matter}(\text{focAuxSG}) \leftarrow & \text{matter}(\text{focAuxStG}) \leftarrow \end{array} \right\}$$

where $\text{matter}(\text{focVSG})$ denotes that the focus is on the verb in Standard German, $\text{matter}(\text{focModSG})$ denotes focus on modal verb in Standard German, . . . , and $\text{matter}(\text{focAuxStG})$ denotes the focus on the auxiliary verb in St. Gallen Swiss German. Π contains information about preference orderings, which depend on stress and the dialect. We have taken for St. Gallen Swiss German the ranking $\text{Foc}(i) \gg H\text{-VComp} \gg \text{Comp-H}$, where $i \in \{V, \text{Mod}, \text{Aux}\}$. Analogously, for Standard German we have taken the ranking $\text{IF}(i) \gg \text{Comp-H} \gg H\text{-VComp}$, where IF (“Ideal Focus”) denotes the focus placement constraint that is at work in Standard German, cf. [14] – roughly, it requires focused verbs to be “isolated” from the next higher verb at an edge of the cluster, e.g., 312 is ideal for focus on verb 3, and 132 is ideal for focus on verb 1 or 2. Furthermore, Π contains the slightly modified rules (1)–(9), where the preference relations and the constraints depend on the abduct matter. Again, we have used DLV as diagnosis system. Since we are interested in abducting single explanations⁵ for this example, we have used the DLV option `-FDsingle`.

As explanation for $O(321)$ we get that the focus is on the auxiliary verb in Standard German, for $O(123)$ focus on auxiliary in St. Gallen dialect, for $O(132)$ focus on modal in Standard German, for $O(312)$ we get two explanations, one where the focus is on the predicative verb in Standard German, and the other where it is on the predicative verb in St. Gallen German. $O(213)$ is explained by focus on the modal in St. Gallen German. $O(231)$ has no explanations. These results are in line with the empirical findings in [14].

5 Discussion and Further Work

In this work, we have associated optimality theory with abduction and preference handling. Abduction and preference handling were studied in many other issues, e.g in [10] to derive intended conclusions. But, as far as the authors know, abduction and preferences were not yet linked to optimality theory before.

We have shown that abduction within ASP is a useful knowledge reasoning tool for linguistic problems, here: dialectic studies, where the abducted explanations match with the empirical results found out by the linguists. ASP has also successfully been used for research in historical linguistics [7].

We have taken the perspective of a linguist and have reconstructed dialectal variation as abduction problem: Given an observation that a sentence is found as grammatically correct (well-formed), abduct the underlying constraint ranking of the dialect. Furthermore, we have provided an encoding (within ASP) for the diagnosis front-end of the DLV system.

⁵ An explanation Δ is *single*, if $|\Delta| = 1$ holds.

Regarding linguistic studies, there is an ongoing debate within linguistics about how unique the rule systems of language are in human cognition, as well as in biology in a very broad sense. The reconstruction of grammatical regularities with abduction and preference handling has consequences for this debate: if grammars can be modeled this way, then they share core properties with other non-linguistic rule systems. This supports a position that does not make special assumptions about the nature of linguistic rule systems.

Regarding well-formed expressions, optimal candidates were defined only for strict total orders of the underlying constraints before [1]. In this work, we have extended the definition for optimal candidates to total orders, where it is allowed that constraints can be ranked equally. We want to note that there are several possibilities to model the determination of winners in case of equally ranked constraints. Another possibility would be to “count” all constraint violations of candidates which are ranked equally and then apply Definition 1 where equally ranked constraints are understood as one constraint. But this may lead to unintuitive results. For example, let x, y, z be candidates, c_1, c_2 be constraints where $c_1 \approx c_2$ and let be given the following violation degrees:

	c_1	c_2
x	* * *	
y	*	*
z	* * *	

Definition 2 yields x, y and z as optimal. Counting the violation degrees of all equally ranked constraints would lead to y as a winner. Then, x and z are not optimal candidates which is not desired.

We have abducted constraint orderings for dialects in German. Furthermore, we have exemplarily studied abduction for focus dependent orders. The results confirm the empirical findings. Optimality Theory can thus successfully be modeled with abduction and preference handling. For the practicing linguist, this tool can be quite helpful. When considering a linguistic phenomenon, one usually starts with exploring it within a few languages. A constraint set is hypothesized that is held responsible for the observed patterns. However, as it is one premise of OT that *any* constraint ranking is a possible grammar, it is not enough to show that a set of constraints can be used to explain a particular phenomenon in one language. All possible constraint rankings and all possible candidates have to be considered. Every OT analysis leads to a proposal about *possible* languages. The calculation of these predictions can easily go beyond what one can handle without computational tools. The abductive framework introduced here can therefore aid the evaluation of linguistic theories.

An interesting further research topic is to study the outcome of observing non-unique optimal candidates. That is, we have observations like $winner(x)$, where it is possible that other candidates $y \neq x$ can also be obtained optimal. First experiments have shown that this leads to different results for the focus contexts. There, it happens that for special rankings among the constraints more than one candidate is observed as a winner. Furthermore, candidate 132 can be explained in dialect considerations, cf. Table 2, when we consider non-unique optimal candidates. We will leave it to further linguistic studies to interpret these behaviors. Another line would be to observe candi-

dates as non-optimal. That is, the explanations give possible constraint rankings such that a candidate could not be a winner. Also, one can study the abduction of partial preference relations, which has not been considered in this paper. Although optimality theory excludes partial orders, we can ask here whether partial orders (not total) may lead to “optimal” candidates. More precisely, under which conditions and for which examples, is a partial order among constraints an explanation, but no total extension of that partial ordering leads to an explanation for observing a candidate as winner. For example, candidate 132 violates each of our word order constraints once and represents a kind of “compromise candidate”, which could be obtained by partial orders. Interestingly, this candidate can be observed in German dialects very frequently, it even appears to be a second default order sometimes. In [14], this is captured by assuming additional constraints. Introducing new constraints is a delicate issue, as these require independent substantial justification which is not always easy to find. An explanation of the occurrence of the 132 order as default with partial constraint ranking would have the advantage of avoiding this consequence.

Acknowledgments. The first author was supported by the German Science Foundation (DFG) under grant SCHA 550/6, TP C and by the EC under project IST-2001-37004 WASP. The second author was supported by the DFG under grant FOR-375/2-A3.

References

1. P. Besnard, G. Fanselow, and T. Schaub. Optimality theory as a family of cumulative logics. *Journal of Logic, Language and Information*, 12(2):153–182, April 2003.
2. J. Delgrande, T. Schaub, H. Tompits, and K. Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 20(2):308–334, 2004.
3. M. Denecker and A. Kakas. Abduction in logic programming. In A. Kakas and F. Sadri, editors, *Computational Logic. Logic Programming and Beyond*, volume 2407 of *Lecture Notes in Computer Science*, pages 402–436. Springer-Verlag, 2002.
4. DLV. <http://www.dbai.tuwien.ac.at/proj/dlv/>, 2002.
5. T. Eiter, W. Faber, N. Leone, and G. Pfeifer. The diagnosis frontend of the dlvs system. *AI Communications*, 12(1-2):99–111, 1999.
6. T. Eiter, G. Gottlob, and N. Leone. Abduction from logic programs: semantics and complexity. *Theoretical Computer Science*, 189(1–2):129–177, 1997.
7. E. Erdem, V. Lifschitz, L. Nakhleh, and D. Ringe. Reconstructing the evolutionary history of indo-european languages using answer set programming. In *Practical Aspects of Declarative Languages, 5th International Symposium, PADL 2003*, volume 2562 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2003.
8. P. Flach and A. Kakas. *Abduction and Induction: Essays on their Relation and Integration*. Applied Logic Series. Kluwer Academic Publishers, 2000.
9. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the International Conference on Logic Programming*, pages 1070–1080. The MIT Press, 1988.
10. K. Inoue and C. Sakama. Abducing priorities to derive intended conclusions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 44–49. Morgan Kaufmann Publishers Inc., 1999.

11. A. Kakas, R. Kowalski, and F. Toni. The role of abduction in logic programming. In *Handbook of logic in Artificial Intelligence and Logic Programming*, volume 5, pages 235–324. Oxford University Press, 1998.
12. A. C. Kakas and P. Mancarella. Database updates through abduction. In D. McLeod, R. Sacks-Davis, and H. Schek, editors, *Proceedings of the 16th International Conference on very large databases (VLDB'90)*, pages 650–661. Morgan Kaufmann, 1990.
13. Antonis C. Kakas and Paolo Mancarella. Generalized stable models: A semantics for abduction. In *9th European Conference on Artificial Intelligence, ECAI '90*, pages 385–391, 1990.
14. T. Schmid and R. Vogel. Dialectal variation in german 3-verb-clusters. a surface-oriented optimality theoretic account. *Journal of Comparative Germanic Linguistics*, 7(3):235–274, 2004.