Elimination of spurious explanations

Gerhard Friedrich [1]

Abstract.

The generation of explanations is considered as a main asset of knowledge-based systems. In this paper we show that current approaches of generating explanations for constraint satisfaction problems fall short. These approaches can lead to spurious explanations with respect to a proposed (or selected) solution. We introduce an extension of current explanation principles such that all explanations of properties regarding a proposed solution are well-founded explanations.

1 Introduction

The generation of explanations has a long tradition in expert systems. For example in the area of sales supporting systems [7] a knowledge-based software application helps the customer to find the right product (configuration) for her needs. In sales support systems as well as configuration systems the customer has to answer various questions and finally one product or a manageable number of products is presented. Constraint-based methods became a key technology in this area because they offer enough expressive power to represent the relevant knowledge. In addition a huge library of concepts and algorithms is available to efficiently solve various reasoning tasks.

In such applications a solution represents a product or a service a customer can purchase. Explanations are generated in order to give feedback. For example because you, as a customer, told us that comfortable cars are important to you we included a special sensor system in our offer that helps you to park your car easily. Such explanations are exploited by customers in various ways, e.g., to increase the confidence in a solution or to facilitate trade-off decisions [4].

In this paper we focus on the question which (customer) decisions led to certain properties of a solution (product). Current approaches employ the abduction principle in order to generate minimal explanations for a set of assertions. We show that these approaches output explanations which are based on spurious solutions, thus leading to wrong reasons why certain customer choices result in particular features of a specific product. Based on this observation we propose a new concept of (minimal) well-founded explanations which eliminates spurious explanations. For the construction of an algorithm we explore important properties of this new concept. The new algorithm can be easily integrated with existing constraint satisfaction problem solvers with acceptable additional computation costs.

In Section 2 we introduce a working example which is exploited in Section 3 to show the problems with current approaches. In Section 4 we present an analysis of these problems and motivate our new concept. Section 5 introduces well-founded explanations. An algorithm for the computation of well-founded explanations is given in Section 6 followed by a discussion of related work.

[1] Universitaet Klagenfurt, Universitaetsstrasse 65, 9020 Klagenfurt, Austria, email: gerhard.friedrich@ifit.uni-klu.ac.at

2 Example

For the introduction of our concept we take an example from a car configuration domain. Let us assume the customers can select various packages for a car. We distinguish between a *business package* and a *recreation package*. The recreation package includes a video-camera on the rear side of the car which allows watching the distance to an obstacle behind the car. This technical device supports the customer-oriented product function *easy parking*. Note, that customers may not be interested in the technical details a priori but may request them for justifying their confidence in a solution. The business package includes a sensor system in the back bumper which also supports easy parking. However, this sensor system is incompatible with the recreation package for technical reasons (a tow-bar comes along with the recreation package preventing the assembly of the sensor system). Note, that from a customer point of view, the video-camera and the sensor system support the *same* product functionality. In addition, the business package includes a radio with a GSM-telephone (GSM-radio) which supports hands-free mobile communication.

This domain can be modeled as a constraint satisfaction problem by the following variables and constraints. The set of variables $\mathcal{V}$ is {*biz-pack*, *rec-pack*, *video*, *sensor*, *GSM-radio*, *easy-parking*, *free-com*}. Unary constraints define the domains of the variables. For simplicity we assume for each variable the domain $\{y, n\}$.

Further constraints are specified by the following tables:

$c_{r,v}$:[2] If *rec-pack* is chosen then *video* must be assembled (and vice versa).

$c_{b,r,s}$: If *biz-pack* is chosen and *rec-pack* is not chosen then *sensor* is assembled. *rec-pack* and *sensor* are incompatible.

$c_{r,v}$:

rec-pack	*video*
y	y
n	n

$c_{b,r,s}$:

biz-pack	*rec-pack*	*sensor*
y	y	n
y	n	y
n	y	n
n	n	n
n	n	y

$c_{v,s,e}$: If *video* or *sensor* is assembled then *easy-parking* is supported (and vice versa).

$c_{v,s,e}$:

video	*sensor*	*easy-parking*
n	n	n
y	n	y
n	y	y
y	y	y

The constraint connecting the variables *biz-pack* and *GSM-radio* is called $c_{b,g}$. $c_{g,f}$ connects *GSM-radio* and *free-com*. The tables for these two constraints are identical to the table of $c_{r,v}$.

Let us assume that a customer chooses the business package and

[2] Variables are abbreviated by their first letter(s).

the recreation package. Consequently, the configured car includes a video-camera and a GSM-radio. Functions supported by such a car are easy-parking and hands-free mobile communication. More formally, if the customer sets $\{$*biz-pack* $= y$, *rec-pack* $= y\}$, then the solution to the constraints $C = \{c_{r,v}, c_{b,r,s}, c_{v,s,e}, c_{b,g}, c_{g,f}\}$ representing the configured car assigns *video* $= y$, *sensor* $= n$, *GSM-radio=y*, *easy-parking* $= y$, *free-com* $= y$.

Let us assume that this solution is presented to the customer. If the customer asks which choices led to these functions of the *specific configured* car clearly the following answer must be provided. Easy parking is supported because the car comes with a video camera. This video camera is included because it is included in the recreation package. Note, that business package is not responsible for easy parking in our case because the sensor system cannot be included.

Our goal is to provide concepts and algorithms which are able to compute such explanations automatically. In particular we must identify those parts of the user input and the knowledge base which explain features of a given solution. In the following we will review the standard approach for generating explanations which turns out to allow problematic outputs.

3 Problems with the current approach

Abduction is the widely accepted concept for defining explanation [3, 8, 11]. The basic idea of these proposals are to use entailment ($\models$) to explain outputs of a problem solving process. Following [8, 11] we base our concept on constraint satisfaction.

More formally, a constraint satisfaction problem (CSP) $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ [8] is defined by a set of variables $\mathcal{V}$, a set of constraints $\mathcal{C}$, and a global domain $\mathcal{D}$. Each constraint has the form $c(x_i, \ldots, x_j)$ where $x_i, \ldots, x_j$ are n variables in $\mathcal{V}$ and c is an n-ary constraint predicate. Each n-ary constraint predicate has an associated n-ary relation $R(c) \subseteq \mathcal{D}^n$. A mapping $v : \mathcal{V} \rightarrow \mathcal{D}$ of variables to values represented by a set of values associated to variables $\{(x_k = v_{x_k}) | x_k \in \mathcal{V} \wedge v_{x_k} = v(x_k)\}$ satisfies a constraint $c(x_i, \ldots, x_j)$ iff $(v_{x_i}, \ldots, v_{x_j}) \in R(c)$. Such a mapping v is a solution of the CSP iff it satisfies all constraints in $\mathcal{C}$.

A set of constraints C is satisfiable iff the CSP with variables $V(C)$ and constraints C has a solution. A CSP $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ is trivially satisfied if $\mathcal{V}$ or $\mathcal{C}$ is empty. We also write $C \models \bot$ in the case where $\bot$ is a unary constraint with arbitrary variable and empty relation, i.e., the CSP is not satisfiable. A mapping $v : \mathcal{V} \rightarrow \mathcal{D}$ is a solution of $(C, \mathcal{V}, \mathcal{D})$ iff $C \cup \{(x_k = v_{x_k}) | x_k \in V(C) \wedge v_{x_k} = v(x_k)\}$ is satisfied. Consequently, finding a solution of a CSP is mapped to the problem of checking the consistency of a set of constraints.

Entailment is defined as usual:

Definition 1 *(Junker) A constraint ϕ is a (logical) consequence of a set of constraints C with variables $V(C)$ iff all solutions of the CSP with variables $V(C \cup \phi)$ and constraints C also satisfy the constraint ϕ. We write $C \models \phi$ in this case.*

Junker [8] gives the following definition of an explanation (rsp. argument) based on a propagation operator Π.

Definition 2 *(Junker) Let Π be a propagation operator, $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ a consistent CSP, and ϕ a constraint.*

A subset C of $\mathcal{C}$ is called a Π-argument for ϕ and $\mathcal{C}$ iff $\phi \in \Pi(C)$. C is a minimal Π-argument for ϕ and $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ iff no proper subset of C is a Π-argument for ϕ and $(\mathcal{C}, \mathcal{V}, \mathcal{D})$.

$\Pi(C)$ corresponds to the unique minimal fixpoint that contains C. In case Π is correct and complete the test $\phi \in \Pi(C)$ can be formally replaced by $C \models \phi$, i.e., every solution of C is a solution of ϕ. Consequently, Junker follows the abduction principle for defining explanations (arguments)[3].

Example continued: There are two minimal explanations (arguments) for *easy-parking* $= y$[4]:

EXP1: $\{$*rec-pack* $= y,\ c_{r,v},\ c_{v,s,e}\} \models$ *easy-parking* $= y$, which is intended. However, there is a second minimal explanation:

EXP2: $\{$*biz-pack* $= y,\ c_{b,r,s},\ c_{r,v},\ c_{v,s,e}\} \models$ *easy-parking* $= y$.

Consequently, on one hand *rec-pack* $= y$ is part of a minimal explanation for *easy-parking* $= y$ but on the other hand *biz-pack* $= y$ is also part of an alternative minimal explanation for *easy-parking* $= y$. Note, that the original solution (for which we are generating explanations) includes a video camera. Clearly, the second explanation is not correct with respect to the original solution since easy parking is provided by a video-camera and the video-camera is included by the recreation package and not by the business package.

The reason for this spurious explanation lies in the fact that if we choose the business package then either the sensor or the video-camera is assembled, depending on the choice regarding the recreation package. In both variants of such a car easy parking is supported. However, the customer *has chosen* the recreation package which implied the video-camera. An explanation of a consequence of user inputs must be based on the solution implied by these user inputs. Reducing the set of user inputs in order to find minimal explanations allows potentially more solutions (logical models) which may lead to a wrong argumentation. Consequently, the second explanation says that if the business package is chosen then all solutions will provide easy parking, regardless of other customer choices and solutions presented to the customer.

4 Analysis and outline of the new concept

In the following we will elaborate our argumentation in more detail and outline the basic ideas of our new concept for eliminating spurious explanations. We will apply the concept of projection which is defined according to relational algebra in databases. Let constraint c have variables $x_i, \ldots, x_j$ (and possibly others). The projection of constraint c on $x_i, \ldots, x_j$ (written as $c\{x_i, \ldots, x_j\}$) is a constraint with variables derived from the variables of c by removing all variables not mentioned in $x_i, \ldots, x_j$, and the allowed tuples of c are defined by a relation $R(c\{x_i, \ldots, x_j\})$ consisting of all tuples $(v_{x_i}, \ldots, v_{x_j})$ such that a tuple appears in $R(c)$ with x_i value v_{x_i}, ..., x_j value v_{x_j}. A constraint with no variables is trivially satisfied.

Subsequently, we compare the solutions of the original problem and of the two minimal explanations EXP1 and EXP2. The only solution based on the set of all constraints C and all user inputs *rec-pack* $= y$ and *biz-pack* $= y$ (the original CSP) is:

rec	biz	vid	sen	eas	gsm	fre
y	y	y	n	y	y	y

For the explanation of *easy-parking* $= y$ where we use EXP1 (i.e., the user input *rec-pack* $= y$ and $c_{r,v},\ c_{v,s,e}$), the solutions implied for the original variables $\mathcal{V}$ are

[3] Junker partitions the set of constraints in explainers and a background theory. In order to simplify our presentation we omit this partitioning. This partitioning can be introduced in our concepts easily.

[4] We consider user inputs as additional constraints.

solution	rec	biz	vid	sen	eas	gsm	fre
1	y	y	y	n	y	y	y
2	y	n	y	n	y	n	n

In both solutions *easy-parking* is y. Solution 1 is identical to the solution of the original CSP. However, Solution 2 differs in the variables {*biz-pack*, *GSM-radio*, *free-com*} from the original CSP. So we might argue that an explanation eventually exploits values of variables which are out of the scope of the original solution and therefore might lead to a spurious explanation. However, in order to derive *easy-parking* $= y$ we only need the constraints $c_{r,v},\ c_{v,s,e}$. Consequently, variables {*biz-pack*, *GSM-radio*, *free-com*} are superfluous to derive *easy-parking* $= y$. Even not all variables in $c_{r,v},\ c_{v,s,e}$ are necessary for the derivation. If we analyze $c_{v,s,e}$ then we recognize that setting *video* to y implies *easy-parking* $= y$, regardless of the value of *sensor*. Consequently, the relevant variables in our case are *rec-pack*, *video*, and *easy-parking*. The solutions of $\{$*rec-pack* $= y,\ c_{r,v},\ c_{v,s,e}\}$ and the solutions of the original CSP projected on these relevant variables are identical.

For the explanation of *easy-parking* $= y$ where we use EXP2 (i.e., the user input *biz-pack* $= y$ and the constraints $c_{b,r,s},\ c_{r,v},\ c_{v,s,e}$), the solutions implied for the original variables $\mathcal{V}$ are

solution	rec	biz	vid	sen	eas	gsm	fre
1	y	y	y	n	y	y	y
2	n	y	n	y	y	y	y

Since we only need $c_{b,r,s},\ c_{r,v},\ c_{v,s,e}$ for the explanation, the variables not included in these constraints are irrelevant for this explanation. All other variables in these three constraints (i.e., *biz-pack*, *rec-pack*, *video*, *sensor*, *easy-parking*) are needed. For example if we delete the variable *video* then $\{$*biz-pack* $= y,\ c_{b,r,s},\ c_{r,v}\{r\},\ c_{v,s,e}\{s,e\}\} \not\models$ *easy-parking* $= y$.

The solutions w.r.t. these relevant variables are

solution	rec	biz	vid	sen	eas
1	y	y	y	n	y
2	n	y	n	y	y

The argumentation for *easy-parking* $= y$ must show that in both solutions (in both logical models) *easy-parking* $= y$ is contained. In particular, the constraint $c_{b,r,s}$, and the user input *biz-pack* $= y$ imply either *sensor* $= y$ or *rec-pack* $= y$. *rec-pack* $= y$ implies *video* $= y$. In both cases *easy-parking* $= y$ is implied by $c_{v,s,e}$. Please note, that this argumentation uses variable settings which are not contained in the original solution, e.g., the car we are talking about comes with a video and not with a sensor equipment. We consider such an argumentation as spurious because it argues with a possible world which is apparently not possible under the current settings.

The principal idea of our approach is, that an explanation based on constraints C for a constraint ϕ and for a specific solution $\int$ must imply ϕ and the possible solutions of the explanation must be consistent with the specific solution $\int$ (w.r.t. the relevant variables).

5 Well-founded explanations

The following definitions of explanation will lead to a more concise explanation compared to previous approaches [8]. We not only consider the relevance of constraints but also investigate the relevance of variables. The goal is to compute a minimal explanation consisting of constraints and variables needed to deduce a certain property. Constraints and variables in such an explanation are exploited to construct an *understandable* argumentation chain for the user [1]. For the introduction of our concepts we need the application of projection on a set of constraints. $C\{V\}$ is defined by applying the projection on $V \subseteq \mathcal{V}$ to all $c \in C$, i.e., $C\{V\} = \{c\{V \cap V(c)\} | c \in C\}$. $V(c)$ are the variables of c.

Definition 3 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, ϕ a constraint.*

A tuple (C, V) where $C \subseteq \mathcal{C}$ and $V \subseteq \mathcal{V}$ is an explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ iff $C\{V\} \models \phi$.

(C, V) is a minimal explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ iff there exists no $C' \subset C$ and no $V' \subset V$ s.t. (C', V) or (C, V') or (C', V') is an explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$.

Example cont.: ($\{$*rec-pack* $= y,\ c_{r,v},\ c_{v,s,e}\}$, {*rec-pack*, *video*, *easy-parking*}) is a minimal explanation for *easy-parking* $= y$.

For the computation of minimal explanations we employ the following monotonicity property.

Remark 1 *If $C\{V\} \not\models \phi$ then for all $V' \subseteq V$ holds $C\{V'\} \not\models \phi$. The same applies for deleting constraints. However, it could be the case that (C', V) and (C, V') are minimal explanations for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ and $C' \subset C$ and $V' \subset V$.*

We employ a CSP to find a solution for a user. Such a solution is described by a set of *solution relevant* variables S which comprise all or a subset of variables of the CSP. We make the reasonable assumption, that there is enough information provided by the user (or about the user) such that the CSP unambiguously defines the values of variables S. More formally, $\int = \{(x_k = v_{x_k}) | x_k \in S \wedge (\mathcal{C} \models x_k = v_{x_k})\}$. For example in car configuration the user has to provide enough information such that a car is well-defined. Information gathering is the task of an elicitation process [10, 1]. Our approach deals with the generation of explanations of properties of a (possible) solution for a user. Consequently, a user can explore various solutions and can ask for an explanation regarding the relation between user decisions and properties of a specific solution.

Subsequently, the projection $\int\{V\}$ of a solution $\int$ on variables V is defined as $\{(x_k = v_{x_k}) | x_k \in V \wedge (x_k = v_{x_k}) \in \int\}$.

The definition of well-founded explanations for a property ϕ w.r.t. a user solution $\int$ is based on the following idea. First, an explanation (C, V) for ϕ, i.e., $C\{V\} \models \phi$ must show that every solution (logical model or sometimes called possible world model) of $C\{V\}$ is a solution (a model) for ϕ. Second, if $C\{V\}$ allows some possible world models with value assignments of solution relevant variables S other than those assigned in $\int$ then the explanation of ϕ is based on possible world models which are in conflict to the original solution $\int$ (which was presented to the user). Therefore we must assure that every possible world (solution) of $C\{V\}$ is consistent with the variable assignment of $\int$.

Definition 4 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S, (C, V) an explanation for ϕ.*

A tuple (C, V) is a well-founded (WF) explanation for ϕ w.r.t. $\int$ iff every solution $s\{S\}$ of $(C\{V\}, V, \mathcal{D})$ is a part of $\int$ (i.e., $s\{S\} \subseteq \int$).

(C, V) is a minimal well-founded (MWF) explanation for ϕ w.r.t. $\int$ iff there exists no $C' \subset C$ and no $V' \subset V$ s.t. (C', V) or (C, V') or (C', V') is a WF explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ w.r.t. $\int$.

Remark 2 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, (C, V) an explanation for ϕ and $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S.*

1. *An explanation (C, V) is a well-founded explanation for $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ w.r.t. $\int$ iff $C\{V\} \models \int\{V\}$.*
2. *If $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ is satisfiable and $\mathcal{C} \models \phi$ then there always exists a well-founded explanation for ϕ.*

3. *It could be the case that for a satisfiable $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ and a ϕ s.t. $\mathcal{C} \models \phi$ and $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for S there exists no minimal explanation of ϕ which is also well-founded.*

By applying Definitions 3, 4 and Remark 2.1 the subsequent corollary follows immediately which characterizes well-founded explanations based on logical entailment.

Corollary 1 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP and $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S.*

A tuple (C, V) where $C \subseteq \mathcal{C}$ and $V \subseteq \mathcal{V}$ is a well-founded explanation for ϕ w.r.t. $\int$ iff $C\{V\} \models \phi \wedge \int\{V\}$.

Example cont.: Let a car be characterized by the solution relevant variables *video*, *sensor*, *GSM-radio* which describe the configuration needed for manufacturing. ($\{$*rec-pack* $= y$, $c_{r,v}$, $c_{v,s,e}\}$, $\{$*rec-pack*, *video*, *easy-parking*$\}$) is a MWF explanation for *easy-parking* $= y$ w.r.t. the solution (car configuration) *video* $= y$, *sensor* $= n$, *GSM-radio* $= y$. It entails *easy-parking* $= y$ and *video* $= y$. ($\{$*biz-pack* $= y$, $c_{b,r,s}$, $c_{r,v}$, $c_{v,s,e}\}$, $\{$*biz-pack*, *rec-pack*, *video*, *sensor*, *easy-parking*$\}$) is a minimal explanation for *easy-parking* $= y$ but it is not well-founded since it does not entail *video* $= y$, *sensor* $= n$.

6 Computing minimal well-founded explanations

The following remarks are exploited for the design of an algorithm that computes MWF explanations. The truth value of $C\{V\} \models \int\{V\}$ is non-monotonic w.r.t. the addition or deletion of variables of V. For example although $C\{V\} \not\models \int\{V\}$ there could exist a non empty $V' \subset V$ s.t. $C\{V'\} \models \int\{V'\}$. Reducing (or adding) variables reduces (adds) constraints on both sides of the entailment relation.

For the computation we employ a propagation operator Π as described above and following [8]. Additions and deletions of constraints are performed by an add and delete operator. Let $\Pi(\emptyset) := \emptyset$ and $\Pi(C) := \text{add}(\Pi(C'), c)$ for sets $C' \subset C \subseteq \mathcal{C}$ and $C = C' \cup \{c\}$. $\Pi(C') := \text{delete}(\Pi(C), c)$. Deleting a variable x in a set of constraints corresponds to deletion of $n-1$ equality constraints where n is the number of constraints c where x is a variable, i.e., $x \in V(c)$.

Remark 3.1. and 3.2. say that if we lost the well-founded property because $C\{V\} \not\models \phi$ by deleting constraints or variables this property cannot be restored by additional deletions. Remark 3.3. states that if we lost the property of $C\{V\} \models \int\{V\}$ (e.g., by variable deletions) and this property can be restored by additional variable deletions, then there is a maximal set of variables $V_m \subset V$ s.t. $C\{V_m\} \models \int\{V_m\}$. Remark 3.4. and 3.5. tell us that if we have minimized the variables w.r.t. to ϕ or $\int$ then this minimality property is preserved if we delete constraints. Consequently, we can first minimize variables and then constraints.

Remark 3 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S, and ϕ a constraint.*

1. *if $C\{V\} \not\models \phi$ then there is no well-founded explanation (C, V') for $(\mathcal{C}, \mathcal{V}, \mathcal{D})$, $\int$, ϕ with variables $V' \subseteq V$.*
2. *if $C\{V\} \not\models \phi$ then there is no well-founded explanation (C', V) for $(\mathcal{C}, \mathcal{V}, \mathcal{D})$, $\int$, ϕ with constraints $C' \subseteq C$.*
3. *Let $C \subseteq \mathcal{C}$ and $V \subseteq \mathcal{V}$ s.t. $C\{V\} \not\models \int\{V\}$.*
   If there exists $V' \subseteq V$ and $V' \neq \emptyset$ s.t. $C\{V'\} \models \int\{V'\}$ then there is a unique maximal $V_m \subset V$ s.t. for all V'' where $V_m \subset V'' \subseteq V : C\{V''\} \not\models \int\{V''\}$.
4. *Let (C, V) be minimal for V for a $C \subseteq \mathcal{C}$ s.t. $C\{V\} \models \phi$ but for all $V' \subset V : C\{V'\} \not\models \phi$.*
   For all $C' \subseteq C$ where $C'\{V\} \models \phi$, it follows that for all $V' \subset V : C'\{V'\} \not\models \phi$.
5. *Let (C, V) be minimal for V for a $C \subseteq \mathcal{C}$ s.t. $C\{V\} \models \int\{V\}$ but for all $V' \subset V : C\{V'\} \not\models \int\{V'\}$.*
   For all $C' \subseteq C$ where $C'\{V\} \models \int\{V\}$, it follows that for all $V' \subset V : C'\{V'\} \not\models \int\{V'\}$.

The proposed algorithm for computing MWF explanations comprises four functions. Function *MWF-explain* (depicted in Figure 1) is called for a CSP $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ with $\mathcal{C}$, $\mathcal{V}$, ϕ, and $\int$ as input parameters. MWF-explain first minimizes the set of variables and then the set of constraints. For the minimization of constraints the function *reduce-constraints* is employed using a standard explanation algorithm [8]. The minimization of variables is realized by the Function *reduce-variables* depicted in Figure 2. This function starts with a check if $\mathcal{C} \models \phi$. Then it investigates the variables of $\mathcal{V}$. The function holds two sets of variables. V corresponds to a working set, X corresponds to the variables necessary for a MWF explanation. If a selected variable x of V is necessary s.t. $(V - x) \cup X$ does not contain the variables of at least one well-founded explanation then x is included in X. Otherwise, x is deleted from V. In the investigation of the necessity of variable x the Function *maxV* (depicted in Figure 3) is called. This function returns a maximal set of variables V' of $V - x$ s.t. $C\{V' \cup X\} \models \int\{V' \cup X\}$ if such a V' exists. Otherwise, it returns 'no'. In addition it returns the propagation state via an in/out parameter.

function MWF-explain$(C, V, \phi, \int)$
% C set of constraints, V set of variables, ϕ a constraint
% $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables
$V :=$ reduce-variables$(C, V, \phi, \int)$;
$C :=$ reduce-constraints$(C, V, \phi, \int)$;
return (C,V) endfunction

Figure 1. Computation of a MWF explanation

function reduce-variables(inout $C, V, \phi, \int$)
% C input and output: a set of constraints and propagation state
% X output: a minimal set of variables needed for a MWF explanation
$C := \Pi(C\{V\})$;
If $\phi \notin C$ then throw exception 'no explanation';
$X := \emptyset$;
while $V \neq \emptyset$ do
  $x :=$ an-element-of(V);
  $C' := C$;
  $V' :=$ maxV$(C', V, x, X, \int)$;
  if $V' \neq$ 'no' then
    if $\phi \in C'$ then $V := V'$; $C := C'$
    else $X := X \cup \{x\}$; $V := V - \{x\}$ endif
  else $X := X \cup \{x\}$; $V := V - \{x\}$ endif endwhile;
return X endfunction

Figure 2. Reduction of variables

Theorem 1 *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a CSP, ϕ a constraint to be explained w.r.t. the solution $\int$ for the solution relevant variables S.*

The function MWF-explain$(\mathcal{C}, \mathcal{V}, \phi, \int)$ always terminates. If $\mathcal{C} \not\models \phi$ then MWF-explain terminates with an exception. Otherwise, it returns a minimal well-founded explanation (C,V) of ϕ with respect to the solution $\int$ for the solution relevant variables S.

function maxV(inout $C, V, x, X, \int$)
% outputs a $V' \subseteq (V - \{x\})$ s.t. V' is a maximal set of
% variables where $C\{V' \cup X\} \models \int\{V' \cup X\}$
% if such a set exists, otherwise return 'no'
% additional output is C, the current propagation state
diff := $\{x\}$;
$V' := V -$ diff;
loop $C := \Pi(C\{X \cup (V -$ diff$)\})$;
forall $(y = v) \in \int\{X \cup (V -$ diff$)\}$ do
if $(y = v) \notin C$ then
if $y \in V'$ then $V' := V' - y$
else return 'no' endif endif endforall;
if $V' = V -$ diff then return V'
else diff $:= V - V'$ endif endloop endfunction

Figure 3. Maximize variables s.t. solution is entailed.

The computational costs can be estimated as follows. Let n_c be the number of constraints and n_v the number of variables in a CSP $(\mathcal{C}, \mathcal{V}, \mathcal{D})$. The complexity of Function *reduce-variables* is $O(n_c)$ add-constraint operations and $O(n_v^2)$ delete-variable operations. The function *reduce-constraints* can be implemented by a standard explanation algorithm [8] which has a complexity of $O(n_c^2)$ add-constraint operations in the worst case[5]. Consequently the total complexity of finding a MWF explanation is $O(n_c^2)$ add-constraint operations plus $O(n_v^2)$ delete-variable operations. Of course the overall computational costs are dominated by the costs of the propagation operator Π for a specific knowledge base.

We have deployed various sales advisory systems (for digital cameras, financial products, skis, cigars etc.) where the large knowledge bases are typically represented by 40 variables and 60 constraints. Products are characterized in the average by 15 variables. User inputs are covered by 15 to 20 variables. The domains of the variables have a typical size of 5 to 10 possible values. Because of this rather small number of variables the additional overhead should be acceptable for similar real world applications as it is in our domain.

7 Related work

The presented concepts are based on the ideas of Quickxplain [8] which showed excellent results in practical settings. Our concepts avoid spurious explanations and output minimal well-founded explanation w.r.t. a solution.

The work in [5, 12] follows the idea to exploit inferences to compute explanation trees. As their basic definition of explanation is based on the usual abduction principle their method allows spurious explanations. However, one of their goals is to compute concise inferences which are comprehensible. We plan to integrate this approach to facilitate the communication process after identifying a MWF.

Rochart et al. [11] presents methods for implementing explanations within global constraints. Similar to the other approaches their algorithm can lead to spurious explanations. In addition, our intention was to provide a non-intrusive explanation facility in order to re-use easily existing constraint solver and database systems.

The generation of explanations is a very active research area in the field of causal theories [9, 2, 6]. Their concepts are based on causal models described by exogenous and endogenous variables which are connected by functions. In contrast to this approach we do not presume a functional property of constraints. Therefore, we allow a more general form of knowledge. It is interesting to note that our method solves the standard rock-throwing example [9] correctly although our intention was not to provide causal explanations. The investigation of relationships between causal explanations and our proposal will be the subject of future work.

[5] Note, that in [8] the complexity of computing an explanation is expressed under the assumption that the deletions of constraints is done in the reverse order of additions. Every deletion corresponds to restoring the last saved propagation state and by adding constraints as needed.

8 Conclusions

In this paper we have shown that current approaches for the generation of explanations fall short. These approaches may compute spurious explanations which are incompatible to a solution proposed. Based on an analysis of these problems we developed the new concept of well-founded explanations for CSPs. We showed some important properties of this concept which were exploited to develop a non-intrusive algorithm for the computation of minimal well-founded explanations with acceptable additional computation costs.

ACKNOWLEDGEMENTS

I thank Dietmar Jannach and Ulrich Junker for helpful comments and discussions as well as anonymous referees for valuable remarks.

The research project is funded partly by grants from the Austrian Central Bank, OeNB, (No. 9706), from the Kaerntner Wirtschaftsfoerderungsfonds, and from the EU (-20-REG-1025/12-2003).

REFERENCES

[1] L. Ardissono, A. Felfernig, G. Friedrich, A. Goy, D. Jannach, G. Petrone, R. Schäfer, and M. Zanker, 'A framework for the development of personalized, distributed web-based configuration systems', *AI Magazine*, **24**(3), (2003).

[2] Hana Chockler and Joseph Y. Halpern, 'Responsibility and blame: A structural-model approach', in *IJCAI'03*, Acapulco, Mexico, (August 2003).

[3] Luca Console, Daniele Theseider Dupre, and Pietro Torasso, 'On the relationship between abduction and deduction', *Journal of Logic and Computation*, **1**(5), (1991).

[4] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Stumptner, 'Consistency-based diagnosis of configuration knowledge bases', *Artificial Intelligence*, **152**(2), (2004).

[5] Eugene C. Freuder, Chavalit Likitvivatanavong, and Richard J. Wallace, 'Deriving explanations and implications for constraint satisfaction problems', in *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *LNCS*, Paphos, Cyprus, (November 2001). Springer.

[6] M. Hopkins and J. Pearl, 'Clarifying the usage of structural models for commonsense causal reasoning', in *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford, CA, USA, (March 2003). AAAI.

[7] Dietmar Jannach, 'Advisor suite - a knowledge-based sales advisory system', in *European Conference on Artificial Intelligence - PAIS 2004*, Valencia, Spain, (2004).

[8] Ulrich Junker, 'Quickxplain: Conflict detection for arbitrary constraint propagation algorithms', in *IJCAI'01 Workshop on Modelling and Solving problems with constraints (CONS-1)*, Seattle, WA, USA, (August 2001).

[9] James D. Park, 'Causes and explanations revisited', in *IJCAI'03*, Acapulco, Mexico, (August 2003).

[10] Pearl Pu, Boi Faltings, and Pratyush Kumar, 'User-involved tradeoff analysis in configuration tasks', in *CP03 Workshop on User-Interaction in Constraint Satisfaction*, (2003).

[11] Guillaume Rochart, Narendra Jussien, and François Laburthe, 'Challenging explanations for global constraints', in *CP03 Workshop on User-Interaction in Constraint Satisfaction*, (2003).

[12] Mohammed H. Sqalli and Eugene C. Freuder, 'Inference-based constraint satisfaction supports explanation', in *Proceedings of the Thirteenth National Conference on Artificial Intelligence Conference (AAAI 96)*, Portland, Oregon, (August 4-8 1996). AAAI Press / The MIT Press.