Preferred Explanations for Quantified Constraint Satisfaction Problems

Deepak Mehta
4C, UCC, Ireland
d.mehta@4c.ucc.ie

Barry O'Sullivan
4C, UCC, Ireland
b.osullivan@4c.ucc.ie

Luis Quesada
4C, UCC, Ireland
l.quesada@4c.ucc.ie

***Abstract*—The Quantified Constraint Satisfaction Problem (QCSP) is a generalization of the classical constraint satisfaction problem in which some variables can be universally quantified. This additional expressiveness can help model problems in which a subset of the variables take value assignments that are outside the control of the decision maker. Typical examples of such domains are game-playing, conformant planning and reasoning under uncertainty. In these domains decision makers need explanations when a QCSP does not admit a winning strategy. We present an approach to defining preferences amongst the requirements of a QCSP, and an approach to finding most preferred explanations of inconsistency based on preferences over relaxations of quantifiers and constraints. This paper unifies work from the fields of constraint satisfaction, explanation generation, and reasoning under preferences and uncertainty.**

I. Introduction

Uncertainty is ubiquitous in real-world decision-making. However, quantifying the nature of the uncertainty can be very difficult, if not impossible, in many settings. Domain experts can usually provide qualitative statements of which risks are more important to consider than others, and which outcomes are more likely than others. In this paper we report on the formal underpinnings of an approach to risk-aware decision-making that is based on an extension of the classic Constraint Satisfaction Problem (CSP) known as the Quantified Constraint Satisfaction Problem (QCSP) [1]. Parameters under the control of the decision maker are modelled as existentially quantified variables since a value (a decision) must be assigned (made) to these variables. All uncertain variables are universally quantified so that decision makers must consider how to preempt every possible assignment to those variables. Of course, such a formulation means that it will be seldom possible for a decision maker to satisfy the constraints of the QCSP since it is likely that some values of the universal (uncertain) variables cannot be preempted. Therefore, we assist the decision maker by abstracting their decision problem so the specific reasons for infeasibility can be focused upon.

Example 1 (Weekend planning): Assume that John wants to prepare a plan for Saturday and Sunday on Friday evening. He is interested in two activities: rowing (*row*) and watching movie (*mov*). Also, assume that there are two weather possibilities: sun (s) and rain (r). Each activity should be carried out on a different day. If the activity is rowing then the weather should be sunny. Let *Asat* and *Asun* denote the activities performed on Saturday and Sunday, respectively. Let *Wsat* and *Wsun* denote the weather on Saturday and Sunday, respectively. The basic formulation of this problem is as follows:

$$\exists Asat, Asun \in \{row, mov\} : \forall Wsat, Wsun \in \{s, r\} :$$
$$\{(Asat \neq Asun), (Asat = row \Rightarrow Wsat = s), (Asun = row \Rightarrow Wsun = s)\}$$

There is no decision that can be made in this case that properly responds to the risk. This is because for any assignment to *Asat* and *Asun* there is at least one assignment to *Wsat* and *Wsun* that is inconsistent with it. Many relaxations, giving rise to risk responses, of this problem are possible. For example, one relaxation could be to restrict the domain of *Wsat* to $\{s\}$ and another could be to restrict *Wsun* to $\{s\}$. However, if John knows that on Saturday it is less likely to rain, then the former would be preferred over the latter. The QCSP obtained by removing a less likely value r from *Wsat* is as follows:

$$\exists Asat, Asun \in \{row, mov\} : \forall Wsat \in \{s\} : Wsun \in \{s, r\} :$$
$$\{(Asat \neq Asun), (Asat = row \Rightarrow Wsat = s), (Asun = row \Rightarrow Wsun = s)\}$$

This QCSP is satisfiable, i.e. there is an appropriate risk response in this setting. This is because there exist assignments to the existential variables, $Asat = row$ and $Asun = mov$, such that for any assignment to the uncertain/universal variables, *Wsat* and *Wsun*, the constraints are satisfied. ▲

In this paper we present a framework for generating preferred explanations in a QCSP setting. An advantage of the framework is that recent developments in QCSP modelling and solving can be applied directly to qualitative risk management [2]. We present an explanation generation algorithm that takes a preference (or likelihood) ordering into account in order to generate the most preferred (most likely) explanation in a given context.

II. Preliminaries

Definition 1 (Constraint Satisfaction Problem): A constraint satisfaction problem (CSP) is a 3-tuple $P \hat{=} \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ where $\mathcal{X}$ is a finite set of variables $\mathcal{X} \hat{=} \{x_1, \ldots, x_n\}$, $\mathcal{D}$ is a set of finite domains $\mathcal{D} \hat{=} \{D(x_1), \ldots, D(x_n)\}$ where the domain $D(x_i)$ is the finite set of values that variable x_i can take, and a set of constraints $\mathcal{C} \hat{=} \{c_1, \ldots, c_m\}$. Each constraint c_i is defined by the ordered set $var(c_i)$ of the variables it involves, and a set $sol(c_i)$ of allowed combinations of values. An assignment of values to the variables in $var(c_i)$ *satisfies* c_i if it belongs to

$sol(c_i)$. A *solution* to a CSP is an assignment to each variable by a value from its domain such that every constraint in $\mathcal{C}$ is satisfied.

Definition 2 (Quantified CSP): A QCSP, ϕ, has the form

$$\mathcal{Q}.\mathcal{C} = Q_1 x_1 \in D(x_1) \cdots Q_n x_n \in D(x_n).\mathcal{C}(x_1, \ldots, x_n)$$

where $\mathcal{C}$ is a set of constraints (see Definition 1) defined over the variables $x_1, \ldots, x_n$, and $\mathcal{Q}$ is a sequence of quantifiers over the variables $x_1, \ldots, x_n$ where each Q_i $(1 \leq i \leq n)$ is either an existential, $\exists$, or a universal, $\forall$, quantifier.

The expression $\exists x_i.c$ means that "there exists a value $a \in D(x_i)$ such that the assignment (x_i, a) satisfies c". Similarly, the expression $\forall x_i.c$ means that "for every value $a \in D(x_i)$, (x_i, a) satisfies c". When the variable and the domain of the variable is clear from context we often write Q_i rather than $Q_i x_i \in D(x_i)$ in the quantifier sequence. When the position of a universal quantifier, Q_i, in the sequence $\mathcal{Q}$ is j such that $j \neq i$ we write Q_i^j, where $1 \leq j \leq n$, otherwise we simply write Q_i.

III. Relaxations of Requirements

Requirements correspond to either a constraint in the QCSP, or the scope of a universal quantifier, or the position of a universal quantifier. The requirements of an input QCSP are called *original requirements*. When the input QCSP is inconsistent, we seek the closest QCSP by relaxing one or more original requirements. For example, an extensional constraint could be relaxed by adding more allowed tuples, the scope of a universal quantifier could be relaxed by restricting its scope to a subset of the domain of the universally quantified variable, and the position of a universal quantifier could be relaxed by moving it to the left in the sequence of quantifiers. Notice that a universal quantifier could be relaxed by either relaxing its scope or relaxing its position. However, we treat them separately for the purpose of clarity. We frame relaxation of each as instances of *requirement relaxation*, over a partial order defined for that purpose.

Definition 3 (Substitution of a Requirement): Given a QCSP ϕ, the substitution of a requirement r in ϕ results in a new QCSP $\phi[r]$.

- If the requirement $r \equiv Q_i x_i \in D(x_i)$ of type *scope of universal quantifier* is to be substituted by $Q_i x_i \in D'(x_i)$ then $Q_1 x_1 \ldots Q_i x_i \in D(x_i) \ldots Q_n x_n.\mathcal{C}[Q_i x_i \in D'(x_i)]$ results in $Q_1 x_1 \ldots Q_i x_i \in D'(x_i) \ldots Q_n x_n.\mathcal{C}$.
- If the requirement $r \equiv Q_i$ of type *position of universal quantifier* is to be substituted by Q_i^k, where $k < i$ then $Q_1 \ldots Q_k \ldots Q_{i-1} Q_i \ldots Q_n.\mathcal{C}[Q_i^k]$ results in positioning Q_i in k and moving the other quantifiers accordingly, i.e., $Q_1 \ldots Q_i^k Q_k^{k+1} \ldots Q_{i-1}^i \ldots Q_n.\mathcal{C}$.
- If the requirement $r \equiv c_j$ of type *constraint* is to be substituted by another constraint c'_j then $\mathcal{Q}.(c_1 \ldots c_j \ldots c_m)[c'_j]$ results in $\mathcal{Q}.(c_1 \ldots c'_j \ldots c_m)$.

The notion of requirement substitution can be lifted to work on a set of requirements R: $\phi[\emptyset] = \phi$, $\phi[\{r\} \cup R] = (\phi[r])[R]$.

Definition 4 (Ordering over Requirement Relaxations): Let R be the set of possible relaxations of a requirement r_0 and let $r_1 \in R$ and $r_2 \in R$ be two relaxations of r_0. We say that r_2 is a relaxation[1] of r_1, denoted by $r_1 \sqsubseteq r_2$, if and only if for any QCSP ϕ if $\phi[r_1]$ is satisfiable then $\phi[r_2]$ is also satisfiable. We say that r_2 is a strict relaxation of r_1, denoted by $r_1 \sqsubset r_2$, if and only if $r_1 \sqsubseteq r_2$ is true and the converse is not true.

We require that the partial order $\sqsubseteq$ also be a *meet-semilattice*, i.e., *greatest lower bounds* are guaranteed to exist: if $r_1, r_2 \in R$, then $r_1 \sqcap r_2$ is well-defined in which case $r_1 \sqcap r_2 \sqsubseteq r_1$ and $r_1 \sqcap r_2 \sqsubseteq r_2$ hold.

Definition 5 (Relaxation of a QCSP): Given a requirement r of a QCSP ϕ, and a requirement relaxation r' such that $r \sqsubseteq r'$, $\phi[r']$ is a relaxation of ϕ.

Example 2 (Relaxation of a QCSP): Consider a QCSP defined on the variables x_1 and x_2 such that $D(x_1) = \{3, 5\}$ and $D(x_2) = \{6, 9, 10\}$ as follows: $\exists x_1 \in \{3, 5\} \forall x_2 \in \{6, 9, 10\}.\{x_2 \bmod x_1 = 0\}$. This QCSP is false. This is because for any value for variable x_1 there is at least one value in the domain of x_2 that is inconsistent with it.

If we relax the constraint requirement $(x_2 \bmod x_1 = 0)$ to $(x_2 \bmod x_1 < 2)$ the resulting QCSP $\exists x_1 \in \{3, 5\} \forall x_2 \in \{6, 9, 10\}.\{x_2 \bmod x_1 < 2\}$ becomes true. If we relax the scope of the domain of the universally quantified variable x_2 to $\{6, 9\}$ then the resulting QCSP $\exists x_1 \in \{3, 5\} \forall x_2 \in \{6, 9\}.\{x_2 \bmod x_1 = 0\}$ is true. If we relax the position of the universal quantifier from 2 to 1 the resulting QCSP $\forall x_2 \in \{6, 9, 10\} \exists x_1 \in \{3, 5\}.\{x_2 \bmod x_1 = 0\}$ is true. ▲

IV. Preferred Conflicts as Explanations

Given an unsatisfiable QCSP (a conflict) we compute explanations of this unsatisfiability by relaxing a subset of its requirements to the point where any further relaxation would yield a satisfiable QCSP (a minimal conflict). Let ϕ be a QCSP defined over the set of original requirements including those that can be relaxed and those that cannot. An original requirement that cannot be relaxed is also called a mandatory requirement. We use Υ to denote a set of original requirements of ϕ that can be relaxed. $\mathcal{R}$ is a relaxation function on Υ that maps each original requirement in Υ to its set of possible requirement relaxations, i.e., $\forall r_i \in \Upsilon$, $\mathcal{R}_i$ is the set of possible requirement relaxations of r_i.

For each $r_i \in \Upsilon$, we use $\dagger_i$ to denote its full relaxation (or *bottom relaxation*). If a requirement r is a constraint c then its bottom relaxation is the Cartesian product of the domains of the variables involved in the constraint c, i.e., $\dagger_r = \Pi_{x \in var(c)} D(x)$. If a requirement r is either a scope of a universal quantifier or a position of a universal quantifier Q_i then $\dagger_r = \forall x_i \in \emptyset$. Throughout the paper we assume that each $r_i \in \Upsilon$ can be fully relaxed, i.e., $\forall r_i \in \Upsilon$, $\dagger_i \in \mathcal{R}_i$.

We say that $\mathcal{I} \in \prod \mathcal{R}_i$ is an instance of $\mathcal{R}$ if and only if $\forall r_i \in \Upsilon$, $\mathcal{I}_i$ is an element of $\mathcal{R}_i$. Let $\mathcal{I}$ and $\mathcal{I}'$ be two instances of $\mathcal{R}$. We say that $\mathcal{I}'$ is a strict relaxation of $\mathcal{I}$, denoted $\mathcal{I} \sqsubset \mathcal{I}'$, if and only if there exists a requirement $r_i \in \Upsilon$ such that $\mathcal{I}_i \sqsubset \mathcal{I}'_i$ and for all the other requirements

[1] A relaxation of a requirement is also a requirement.

$r_j \in \Upsilon$, $\mathcal{I}_j \sqsubseteq \mathcal{I}'_j$. We use $\bigsqcap(\mathcal{R})$ to denote the *top instance* of $\mathcal{R}$, i.e., if $\mathcal{I} = \bigsqcap(\mathcal{R})$ then there does not exist any other instance $\mathcal{I}'$ of $\mathcal{R}$ such that $\mathcal{I}' \sqsubset \mathcal{I}$. We use $\bigsqcup(\mathcal{R})$ to denote the *bottom* (or a most relaxed) instance of $\mathcal{R}$, i.e., if $\mathcal{I} = \bigsqcup(\mathcal{R})$ then there does not exist any other instance $\mathcal{I}'$ of $\mathcal{R}$ such that $\mathcal{I} \sqsubset \mathcal{I}'$. The former is well-defined when there is a unique minimal relaxation, and the latter one is well-defined when there is a unique maximal relaxation, for each requirement.

We say that a *conflict* is an instance of $\mathcal{R}$ that makes ϕ inconsistent. When confronted with an inconsistent QCSP a user is generally interested in resolving the conflicts. To allow a user to resolve a conflict by relaxing at most one requirement it is important to ensure the minimality of the conflict. We define the notion of minimal conflict with respect to a (typically incomplete) consistency propagation method Π, such as QAC [3], in a similar way to Junker [4]. In what follows, the consistency of a QCSP is defined in terms of Π so *consistency* means Π-*consistency*. Using an incomplete operator is perfectly reasonable since it only means that the conflict computed is minimal with respect to the consistency operator. Furthermore, some interesting classes of QCSP may be easy to solve in practice despite the worst-case theoretical complexity, e.g., the QCSPs solved in [5].

Definition 6 (Minimal Conflict): Given a set of original requirements Υ that can be relaxed, and a consistency propagator Π, a minimal conflict $\mathcal{I}$ of a QCSP ϕ is an instance of $\mathcal{R}$ such that $\phi[\mathcal{I}]$ is inconsistent and there does not exist any $\mathcal{I} \sqsubset \mathcal{I}'$ such that $\phi[\mathcal{I}']$ is inconsistent.

If $\mathcal{I}$ is a minimal conflict of ϕ under $\mathcal{R}$ then $\phi[\mathcal{I}]$ corresponds to a maximally relaxed explanation of ϕ [6].

Now we define the notion of preference over conflicts of a quantified CSP building upon the notion of preference over conflicts of a CSP [4]. Given two conflicts I and I' of a quantified CSP, we say that I is more important than I' if resolving I involves relaxing a more important requirement. As the user is supposed to resolve all the conflicts, it is better to present him/her first with those conflicts that involve more critical decisions, i.e., with those conflicts that involve relaxing more important requirements.

Definition 7 (Anti-lex Ordering): Let $\prec$ be a total order in terms of importance on the set of original requirements Υ. Here, $r_i \prec r_j$ means that r_i is more important than r_j. Let $\mathcal{I}$ and $\mathcal{I}'$ be two instances of a relaxation function $\mathcal{R}$. We say that $\mathcal{I} \prec_{antilex} \mathcal{I}'$ if and only if r_i is the least important original requirement such that $\mathcal{I}_i \neq \dagger_i \wedge \mathcal{I}'_i = \dagger_i$, r_j is the least important original requirement such that $\mathcal{I}'_j \neq \dagger_j \wedge \mathcal{I}_j = \dagger_j$, and $r_i \prec r_j$.

Many conflicts may exist so we focus on the preferred one. If $\mathcal{I}$ and $\mathcal{I}'$ are two minimal conflicts of $\mathcal{R}$ and $\mathcal{I} \prec_{antilex} \mathcal{I}'$ then it means that $\mathcal{I}$ is more important than $\mathcal{I}'$.

Definition 8 (Preferred Conflict): Given a total order $\prec$ in terms of importance on set of requirements Υ, a minimal conflict $\mathcal{I}$ of a QCSP ϕ is a preferred conflict if and only if there is no other minimal conflict $\mathcal{I}'$ of ϕ such that $\mathcal{I}' \prec_{antilex} \mathcal{I}$.

Example 3 (Antilex Ordering on Instances of $\mathcal{R}$): Consider an unsatisfiable QCSP defined on variables x_1, x_2 and x_3 such that $D(x_1) = \{1,2\}$, $D(x_2) = \{1,2,3\}$ and $D(x_3) = \{2,3\}$ as follows: $\exists x_1 \forall x_2 \exists x_3.\{x_1 < x_2, x_2 < x_3\}$. Let $\Upsilon = \{r_1, r_2, r_3\}$ be the set of original requirements that can be relaxed, where $r_1 \equiv \forall x_2 \in \{1,2,3\}$, $r_2 \equiv x_1 < x_2$, and $r_3 \equiv x_2 < x_3$. Let us assume that $r_1 \prec r_2 \prec r_3$ is the order of importance on the requirements. The relaxation function $\mathcal{R}$ is defined as follows: $\mathcal{R}_1 = \{\forall x_2 \in \{1,2,3\}, \forall x_2 \in \emptyset\}$, $\mathcal{R}_2 = \{x_1 < x_2, true\}$, and $\mathcal{R}_3 = \{x_2 < x_3, true\}$. Here $\dagger_1 \equiv \forall x_2 \in \emptyset$, $\dagger_2 \equiv true$, and $\dagger_3 \equiv true$. From the definition of minimal conflict it follows that $\mathcal{I} = \{\forall x_2 \in \{1,2,3\}, x_1 < x_2, true\}$ and $\mathcal{I}' = \{\forall x_2 \in \{1,2,3\}, true, x_2 < x_3\}$ are the only minimal conflicts of $\mathcal{R}$. The least important requirements that need to be relaxed for resolving the conflicts $\mathcal{I}$ and $\mathcal{I}'$ are r_2 and r_3 respectively, and since r_2 is more important than r_3, $\mathcal{I} \prec_{antilex} \mathcal{I}'$. Since there are only two minimal conflicts, $\mathcal{I}$ is also the preferred conflict.

▲

V. Two-Point Relaxation Functions

We present an algorithm for computing a preferred conflict of ϕ under the two-point relaxation function $\mathcal{R}$, where for every original requirement $r_i \in \Upsilon$, $\mathcal{R}_i = \{\dagger_i, r_i\}$. If $\dagger_i$ is in $\mathcal{R}_i$ then r_i is allowed to relax fully. Notice that any pair of instances, say $\mathcal{I}$ and $\mathcal{I}'$, can only be different if there exists at least one $r_j \in \Upsilon$ such that $\mathcal{I}_j \neq \mathcal{I}'_j$, and that would imply that either $\mathcal{I}_j = \dagger_j$ or $\mathcal{I}'_j = \dagger_j$ holds in a two-point relaxation function. Therefore, any pair of instances of the two-point relaxation function $\mathcal{R}$ are comparable and hence they are totally ordered with respect to $\prec_{antilex}$.

The following proposition shows how to compute a preferred conflict by decomposing a given two-point relaxation function defined on a given set of original requirements, which will form the basis for Algorithm 2.

Proposition 1: Let $\Upsilon = \{r_1, \ldots, r_m\}$ be an original set of requirements of a QCSP ϕ and let $\mathcal{R} = \{\{\dagger_1, r_1\}, \ldots, \{\dagger_m, r_m\}\}$ be a relaxation function on Υ. Suppose that $\Upsilon^1 = \{r_1, \ldots, r_k\}$ and $\Upsilon^2 = \{r_{k+1}, \ldots, r_m\}$ are disjoint sets of requirements of ϕ and that no requirement of Υ^2 is preferred to a requirement of Υ^1. Let $\mathcal{I}^2$ be the preferred conflict of ϕ under $\mathcal{R}^2 = \{\{r_1\}, \ldots, \{r_k\}, \{\dagger_{k+1}, r_{k+1}\}, \ldots, \{\dagger_m, r_m\}\}$. Let $\mathcal{I}^1$ be the preferred conflict of ϕ under $\mathcal{R}^1 = \{\{\dagger_1, r_1\}, \ldots, \{\dagger_k, r_k\}, \{\mathcal{I}^2_{k+1}\}, \ldots, \{\mathcal{I}^2_m\}\}$. If $\mathcal{I}^1$ is the preferred conflict of ϕ under $\mathcal{R}^1$ and $\mathcal{I}$ is the preferred conflict of ϕ under $\mathcal{R}$ then $\mathcal{I} = \mathcal{I}^1$.

Proof: To prove that $\mathcal{I} = \mathcal{I}^1$, i.e., $\mathcal{I}^1$ is the preferred conflict of ϕ under $\mathcal{R}$, we prove that any instance of $\mathcal{R}$ that is not in $\mathcal{R}^1$ cannot be the preferred conflict of $\mathcal{R}$. From the definition of $\mathcal{R}^1$, this is equivalent to proving that the projection of Υ^2 on $\mathcal{I}$ i.e., $\mathcal{I}_{\Downarrow \Upsilon^2}$, is equal to $\mathcal{I}^2_{\Downarrow \Upsilon^2}$. We prove this by contradiction. If we assume that $\mathcal{I}_{\Downarrow \Upsilon^2} \neq \mathcal{I}^2_{\Downarrow \Upsilon^2}$ then either $\mathcal{I}_{\Downarrow \Upsilon^2} \prec_{antilex} \mathcal{I}^2_{\Downarrow \Upsilon^2}$ or $\mathcal{I}_{\Downarrow \Upsilon^2} \succ_{antilex} \mathcal{I}^2_{\Downarrow \Upsilon^2}$. If $\mathcal{I}_{\Downarrow \Upsilon^2} \prec_{antilex} \mathcal{I}^2_{\Downarrow \Upsilon^2}$ then it means that there exists a conflict $\mathcal{I}'$ under $\mathcal{R}^2$ such that $\mathcal{I}'_{\Downarrow \Upsilon^1} = \mathcal{I}^2_{\Downarrow \Upsilon^1}$ and $\mathcal{I}'_{\Downarrow \Upsilon^2} = \mathcal{I}_{\Downarrow \Upsilon^2}$. This would imply that $\mathcal{I}' \prec_{antilex} \mathcal{I}^2$, which contradicts the assumption that $\mathcal{I}^2$ is the preferred conflict of $\mathcal{R}^2$. If $\mathcal{I}_{\Downarrow\Upsilon^2} \succ_{antilex} \mathcal{I}^2_{\Downarrow\Upsilon^2}$ then $\mathcal{I} \succ_{antilex} \mathcal{I}^2$. This would imply $\mathcal{I}$ is not the preferred conflict under $\mathcal{R}$, which also contradicts the assumption. ∎

Let $\Upsilon = \{r_1, \ldots, r_m\}$ be an original set of requirements of ϕ that can be relaxed and let $\mathcal{R} = \{\{\dagger_1, r_1\}, \ldots, \{\dagger_m, r_m\}\}$ be a relaxation function on Υ. The algorithm QUICKQCSPXPLAIN for computing a preferred conflict is depicted in Algorithm 1. If the input QCSP, ϕ, is consistent then there is no conflict in which case the algorithm raises an exception. Otherwise, the algorithm QUICKQCSPXPLAIN′ (Algorithm 2) is invoked, which computes the preferred conflict $\mathcal{I}$ of ϕ under $\mathcal{R}$ on the set of requirements Υ.

Algorithm 1 QUICKQCSPXPLAIN($\phi, \Upsilon, \mathcal{R}, \prec$)

Require: : A QCSP ϕ; $\forall r_i \in \Upsilon$, $\mathcal{R}_i = \{\dagger_i, r_i\}$.
Ensure: : A preferred conflict of ϕ.
1: **if** $\bot \notin \Pi(\phi)$ **then**
2: **return** exception "no conflict"
3: $\mathcal{I} \leftarrow$ QUICKQCSPXPLAIN′(ϕ, `true`, $\Upsilon, \mathcal{R}, \prec$)
4: **return** $\mathcal{I}$

The invariant of QUICKQCSPXPLAIN′ is that ϕ under the top instance of $\mathcal{R}$ is inconsistent. If it is not the case then it means that ϕ is consistent under $\mathcal{R}$. One of the parameters of the algorithm is Δ, which is a Boolean variable. It is `true` if it is unknown that ϕ is inconsistent under the bottom instance $\mathcal{B} = \bigsqcup(\mathcal{R})$. If ϕ is inconsistent under $\mathcal{B}$, then the preferred conflict of ϕ under $\mathcal{R}$ is $\mathcal{B}$ (Line 1-2). If $|\Upsilon| = 1$ then it means that there exists only one requirement with two possible relaxations. As the bottom instance is already known to be consistent from Line 1, the top instance of $\mathcal{R}$ has to be inconsistent and the preferred conflict is $\bigsqcap(\mathcal{R})$.

If the cardinality of the set of the original requirements is greater than one, it is ordered in decreasing order of importance with respect to $\prec$. To find the preferred conflict the ordered set of original requirements is divided into two sets, $\Upsilon_1 = \{r_1, \ldots, r_k\}$ and $\Upsilon_2 = \{r_{k+1}, \ldots, r_m\}$, such that no requirement of Υ_2 is more important than one of Υ_1. First, a relaxation function $\mathcal{R}^2$ is obtained from $\mathcal{R}$ by enforcing that each requirement in Υ^1 cannot be relaxed (Line 8-9). If $\mathcal{I}^2$ is the preferred conflict of ϕ under relaxation function $\mathcal{R}^2$ then, from Proposition 1, the preferred conflict of $\mathcal{R}$ is the preferred conflict of $\mathcal{R}^1$, obtained from $\mathcal{R}$ by setting each $\mathcal{R}_r$ for each $r \in \Upsilon_2$ to the corresponding one in $\mathcal{I}^2$ (Line 10-12).

QUICKQCSPXPLAIN′ avoids unnecessary consistency checks in cases where it is known that ϕ is consistent under the bottom instance. More precisely, if all the requirements in Υ^2 are set to their bottom relaxation in $\mathcal{I}^2$ then if all the requirements in Υ^1 are set to their bottom relaxation in $\mathcal{I}^1$ then this would imply that ϕ is consistent under the bottom instance of $\mathcal{R}$, which is a contradiction. Therefore, whenever all the requirements in Υ^2 are set to their bottom relaxation in $\mathcal{I}^2$, Δ_2 is set to `false` (Line 13) to avoid the consistency check in Line 1 when computing $\mathcal{I}^1$.

Algorithm 2 QUICKQCSPXPLAIN′($\phi, \Delta, \Upsilon, \mathcal{R}, \prec$)

1: **if** Δ and $\bot \in \Pi(\phi[\bigsqcup(\mathcal{R})])$ **then**
2: **return** $\bigsqcup(\mathcal{R})$
3: **if** $|\Upsilon| = 1$ **then**
4: **return** $\bigsqcap(\mathcal{R})$
5: let $r_1, \ldots, r_m$ be an enumeration of Υ that respects $\prec$
6: let $k = \lfloor(1+m)/2\rfloor$ where $1 \leq k < m$
7: $\Upsilon^1 \leftarrow \{r_1, \ldots, r_k\}$ and $\Upsilon^2 \leftarrow \{r_{k+1}, \ldots, r_m\}$
8: $\mathcal{R}^2 \leftarrow \mathcal{R}$
9: $\forall r \in \Upsilon^1, \mathcal{R}^2_r \leftarrow \{\sqcap(\mathcal{R}_r)\}$
10: $\mathcal{I}^2 \leftarrow$ QUICKQCSPXPLAIN′(ϕ, `true`, $\Upsilon^2, \mathcal{R}^2, \prec$)
11: $\mathcal{R}^1 \leftarrow \mathcal{R}$
12: $\forall r \in \Upsilon^2, \mathcal{R}^1_r \leftarrow \{\mathcal{I}^2_r\}$
13: $\Delta_2 \equiv \left(((\mathcal{I}^2)_{\Downarrow\Upsilon^2}) \neq \left(\bigsqcup(\mathcal{R}^2)_{\Downarrow\Upsilon^2}\right)\right)$
14: $\mathcal{I}^1 \leftarrow$ QUICKQCSPXPLAIN′($\phi, \Delta_2, \Upsilon^1, \mathcal{R}^1, \prec$)
15: **return** $\mathcal{I}^1$

QUICKQCSPXPLAIN is a reformulation of QUICKXPLAIN [4] in terms of relaxations, and thereby generalising it to QCSP with at most one distinct relaxation available for each of the original requirements, i.e., a requirement is either present or fully relaxed. In the worst-case, QUICKQCSPXPLAIN will perform $\mathcal{O}(k \log \frac{n}{k})$ number of consistency checks, where n is the number of original requirements and k is the number of original requirements in the preferred conflict that are not fully relaxed. Here consistency checks refers to the number of times consistency of a QCSP is checked using Π.

VI. CONCLUSIONS

In this paper we presented a framework for generating most preferred explanations for the inconsistency of a QCSP. The additional expressiveness of the QCSP can help model problems in which a subset of the variables take value assignments that are outside the control of the decision maker. We presented an approach to representing preferences, and a corresponding algorithm for computing preferred explanations based on the notion of conflict.

ACKNOWLEDGMENT

This material is based upon work supported by the Science Foundation Ireland under Grant No. 05/IN/I886 and Grant No. 08/CE/I1523.

REFERENCES

[1] Chen, H.: The Computational Complexity of Quantified Constraint Satisfaction. PhD thesis, Cornell (August 2004)
[2] Gent, I.P., Nightingale, P., Stergiou, K.: Qcsp-solve: A solver for quantified constraint satisfaction problems. In: Procs of IJCAI. (2005) 138–143
[3] Bordeaux, L., Monfroy, E.: Beyond NP: Arc-consistency for quantified constraints. In: Procs of CP 2002. (2002) 371–386
[4] Junker, U.: Quickxplain: Preferred explanations and relaxations for over-constrained problems. In: Procs of AAAI 2004. (2004) 167–172
[5] Stynes, D., Brown, K.N.: Realtime online solving of quantified csps. In Gent, I.P., ed.: CP. Volume 5732 of Lecture Notes in Computer Science., Springer (2009) 771–786
[6] Ferguson, A., O'Sullivan, B.: Quantified constraint satisfaction problems: from relaxations to explanations. In: Procs of IJCAI-2007. (2007) 74–79