Quantified Constraint Satisfaction Problems: From Relaxations to Explanations

Alex Ferguson and **Barry O'Sullivan**
Cork Constraint Computation Centre, University College Cork, Ireland
and Centre for Telecommunications Value-Chain Research
{a.ferguson|b.osullivan}@4c.ucc.ie

Abstract

The Quantified Constraint Satisfaction Problem (QCSP) is a generalisation of the classical CSP in which some of variables can be universally quantified. In this paper, we extend two well-known concepts in classical constraint satisfaction to the quantified case: problem relaxation and explanation of inconsistency. We show that the generality of the QCSP allows for a number of different forms of relaxation not available in classical CSP. We further present an algorithm for computing a generalisation of conflict-based explanations of inconsistency for the QCSP.

1 Introduction

One of the disadvantages of the classical CSP framework is that it lacks sufficient expressive power for modelling particular aspects of real world problems, for example, uncertainty or other forms of uncontrollability in the environment in which a set of constraints must be satisfied.

In this paper we are concerned with the Quantified CSP, a generalisation of the classical CSP that allows some of the variables to be universally quantified [Chen, 2004]. The semantics of universal quantification over a variable is that the set of constraints must be satisfiable for any assignment to it. This is in contrast with classical CSP, where all variables are existentially quantified, i.e. any assignment to an existentially quantified variable that satisfies the constraints is satisfactory. While the classical CSP is known to be NP-complete in general, the QCSP is PSPACE-complete [Chen, 2004]. We consider the problem of relaxing an instance of the QCSP when it is, for example, unsatisfiable. We propose several novel forms of problem relaxation for the QCSP and present an algorithm for generating conflict-based explanations of inconsistency.

Our motivation comes from problems in conformant planning and supply-chain management. We are interested in using constraints to support the local decision-making processes of a company that must supply products to a set of customers, while managing a complex network of its own suppliers. Classical CSP is not sufficiently expressive to model this type of problem concisely, since some of the variables are not under the control of the decision-maker, but we can model the problem as a Quantified CSP. The parallels between adversarial games and QCSP are natural: we can model variables under our control using existential quantifiers and those variables outside our control using universal quantifiers [Chen, 2004].

The remainder of this paper is organised as follows. In Section 2 we present a formal definition of the fundamental concepts in constraint satisfaction and the Quantified CSP. We present several new forms of problem relaxation in Section 3 that have previously not been considered due to the literature's focus on classical CSP. We show how these forms of relaxation can be captured using the notion of *requirement relaxation*. Section 4 presents an approach to generating explanations of conflict in Quantified CSP based on requirement relaxation. We show how an existing explanation generation algorithm for classical CSPs can be extended to the quantified case. We review the most related work in Section 5. A number of concluding remarks are made in Section 6.

2 Preliminaries

Definition 1 (Classical 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 of a value from its domain to each variable such that every constraint in $\mathcal{C}$ is satisfied.*

A fundamental notion used in reasoning about classical CSPs is that of arc consistency [Mackworth, 1977]. Due to logical conjunction, any assignment to the variables that is locally inconsistent with a constraint is guaranteed not to be part of any consistent solution.

In the classical CSP we can regard all variables as being *existentially quantified*: we wish to show that there exist assignments for each of the variables that satisfy all constraints simultaneously. However, we can generalise further by also allowing variables to be *universally quantified*.

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*[1]*. 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*".*

Definition 3 (Semantics of QCSP) *(based on [Gent* et al.*, 2005]) We define the semantics of the* QCSP*,* $\mathcal{Q}.\mathcal{C}$*, recursively as follows. If* $\mathcal{C}$ *is empty then the problem is* true*. If* $\mathcal{Q}$ *is of the form* $\exists x_1 Q_2 x_2 \ldots Q_n x_n$ *then* $\mathcal{Q}.\mathcal{C}$ *is* true *iff there exists a value* $a \in D(x_1)$ *such that* $Q_2 x_2 \ldots Q_n x_n.(\mathcal{C} \cup \{x_1 = a\})$ *is* true*. If* $\mathcal{Q}$ *is of the form* $\forall x_1 Q_2 x_2 \ldots Q_n x_n$ *then* $\mathcal{Q}.\mathcal{C}$ *is* true *iff for every value* $a \in D(x_1)$ *such that* $Q_2 x_2 \ldots Q_n x_n.(\mathcal{C} \cup \{x_1 = a\})$ *is* true*. Otherwise,* $\mathcal{Q}.\mathcal{C}$ *is* false*.*

Example 1 (Quantified CSP) *Consider a* QCSP *defined on the variables* x_1 *and* x_2 *such that* $D(x_1) = D(x_2) = \{1, 2, 3\}$ *as follows:* $\exists x_1 \forall x_2.\{x_1 < x_2\}$*. 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.* ▲

The above deals only with the equivalent of the decision problem for QCSPs; the analogous treatment for exemplification requires consideration of a *winning strategy* generalising the idea of a solution [Chen, 2004]. Of course, the order of the quantifiers in a QCSP is critical. Consider the following QCSP: $\exists x_1 \in \{0, 1\} \forall x_2 \in \{0, 1\}.\{x_1 \neq x_2\}$ is false, while $\forall x_2 \in \{0, 1\} \exists x_1 \in \{0, 1\}.\{x_1 \neq x_2\}$ is true.

3 Relaxation

Our motivation for studying relaxations of the QCSP relates to the observation that many real world problems are over-constrained. Accordingly, we are often interested in relaxing the constraints of the problem so that we find a solution that is at least relatively satisfactory. In this scenario, problem relaxation can be seen as a way of exploring the space of alternative models for a given problem. Furthermore, problem relaxations give us a language with which we can *explain* the over-constrainedness of a problem. Moving from classical CSPs to QCSPs provides us with a richer class of relaxations, and hence, a more expressive language for explanations.

One of the most well known frameworks for reasoning about problem relaxation in classical CSP is Partial Constraint Satisfaction [Freuder and Wallace, 1992]. One can identify four possible ways of relaxing a classical CSP: the domain of a variable can be enlarged; the relation of a constraint (the set of allowed tuples) can be enlarged; a constraint can be removed from the problem; or, a variable can be removed from the problem. Note that all of these strategies can be regarded as enlarging domain or constraint relations. (And domains can themselves be formulated as unary constraints.)

[1] When the domain of a variable is clear from context we often write $Q_i x_i$ rather than $Q_i x_i \in D(x_i)$ in the quantifier sequence.

While all of these forms of relaxation are available to us in the QCSP, the additional expressiveness we have access to within the QCSP gives rise to new forms of relaxation that are not available in classical CSP.

3.1 Relaxation for the Quantified CSP

A useful viewpoint to adopt when studying the QCSP is to consider an instance of the problem as an adversarial game between two players: a universal player who sets the universally quantified variables, and an existential player, who sets the existentially quantified variables [Chen, 2004]. In these terms, we can regard relaxations of the QCSP as modifications to the problem that make it easier for the existential player to win. From the perspective of the universal player, a relaxation of the QCSP may restrict the ways in which this player can falsify the formula. For example, we can restrict the domain of values of the universally quantified variables. From the perspective of the existential player, a relaxation increases the set of responses that can be played to counteract the choices of the universal player. We proceed by discussing five classes of relaxation that are available in the context of the QCSP. While the first two of these are familiar from CSP, the remaining three are particular to the more expressive framework, and have no direct equivalents in the unquantified case.

Single Constraint Relaxation. This is the type of relaxation equivalent to that normally considered in classical CSPs. For example this is the classical approach in the PCSP framework in which allowed tuples are added to an extensionally represented constraint. In principle, this may be replacing a constraint with any logically weaker one. As mentioned earlier, complete removal of a constraint may be considered as a special case of this type of relaxation.

Example 2 (Single Constraint Relaxation) *Consider the following* QCSP*, which is* false*:*

$$\exists x_1 \in \{1, 2\} \forall x_2 \in \{1, 2, 3\}.\{x_1 < x_2\}.$$

If we relax the constraint between x_1 *and* x_2 *from* $<$ *to* $\leq$ *to get the following:*

$$\exists x_1 \in \{1, 2\} \forall x_2 \in \{1, 2, 3\}.\{x_1 \leq x_2\},$$

the QCSP *becomes* true*. This is because if the existential player sets* x_1 *to 1, the universal can no longer find an assignment that falsifies the* QCSP*.* ▲

Relaxation of Existentially Quantified Domains. Enlarging the domain of an existentially quantified variable is the direct equivalent of relaxing domains in the classical framework.

Example 3 (Relaxation of Existential Domains) *We revisit the* QCSP *presented in Example 2. This is* false *because the scenario in which the universal player sets* x_2 *to 1 is available. To counter this, the existential player must be able to play a value that is smaller than 1. A suitable relaxation is the following, which adds the value 0 to the domain of* x_1*:*

$$\exists x_1 \in \{0, 1, 2\} \forall x_2 \in \{1, 2, 3\}.\{x_1 < x_2\};$$

the relaxed QCSP *is* true *since the existential player can set* x_1 *to 0 and the universal player cannot falsify the formula.* ▲

Relaxation of Universally Quantified Domains. A class of relaxation not corresponding to any in classical CSPs is to remove values from the domains of universally quantified variables. Doing so reduces the number of scenarios under which the existential player must find satisfying assignments for the existential variables, leading to a more satisfiable problem. The rationale for this may be an analysis of which 'problem values' it may be permissible to ignore, or the narrowing of a range with a probabilistic interpretation.

Example 4 (Relaxation of Universal Domains) *Returning to the* QCSP *of Example 2, which is* false*, due to the availability of the assignment of x_2 to 1, for which the existential player has no satisfying counter-assignment. However, if we relax the domain of the universally quantified variable x_2 so that it no longer contains the value 1 to get the following:*

$$\exists x_1 \in \{1,2\} \forall x_2 \in \{2,3\}.\{x_1 < x_2\};$$

the relaxed QCSP *is* true *since the existential player can set x_1 to 1 and the universal player cannot falsify the formula.* ▲

Quantifier Relaxation. A fourth form of relaxation, also with no direct equivalent in classical CSP, is to *reverse* the quantification of a variable from universal to existential. This relaxation corresponds to achieving control over a factor previously considered to be environmental. This further relaxes the universal domain relaxation.

Example 5 (Quantifier Relaxation) *We revisit the* QCSP *presented in Example 2, which is* false*. If relax the universal quantifier on variable x_2 to be existential to get:*

$$\exists x_1 \in \{1,2\} \exists x_2 \in \{1,2,3\}.\{x_1 < x_2\};$$

the relaxed QCSP *is* true*.* ▲

Quantifier Moving. A fifth class of problem relaxation corresponds to moving a universally quantified variable to the left in the sequence of quantifiers. Equivalently, we can consider moving an existentially quantified variable to the right in the sequence of quantifiers. This corresponds to analysis or oracular knowledge of the value of an environment variable, or of delaying a decision until better information is available.

Example 6 (Quantifier Moving) *We consider an example, based on one presented earlier in this paper. The following* QCSP *is* false*:*

$$\exists x_1 \in \{0,1\} \forall x_2 \in \{0,1\}.\{x_1 \neq x_2\}$$

We relax this problem by moving the universally quantified variable to the left to give the following QCSP *which is* true*:*

$$\forall x_2 \in \{0,1\} \exists x_1 \in \{0,1\}.\{x_1 \neq x_2\}$$

Informally, the relaxed QCSP *gives the existential player the opportunity to react to the actions of the universal player.* ▲

It should be noted that *whether* each of these relaxations is sensibly possible for a given application is a modelling or knowledge engineering question. In each case, a given constraint, domain, or quantifier may be inherent to the problem, or "background", and thus not susceptible to relaxation.

We claim that this set of relaxations is comprehensive (unless reformulation of the original problem is considered in conjunction with relaxation). It is also orthogonal, if we apply quantifier relaxation only in the case where the domain is a singleton, in the sense that all possible single-step relaxations result in problems that are mutually incomparable.

3.2 QCSP Relaxation as Requirement Relaxation

We now present a uniform treatment of the relaxation of both quantifiers and constraints, that is, the first four classes of relaxation we identify above. *Requirements* correspond to either a constraint in the QCSP, or a universal quantifier, and we frame relaxation of each as instances of *requirement relaxation*, over a partial order defined for that purpose.

Definition 4 (Relaxation of a QCSP) *We define the relaxation $\phi[r]$ of a* QCSP *ϕ to a relaxed version of a single requirement to a new value, r, as follows:*

$$\begin{aligned} &Q_1 x_1 \ldots Q_i x_i \in D(x_i) \ldots Q_n x_n.\mathcal{C}[Q'_i x_i \in D'(x_i)] \\ &\qquad \hat{=}\; Q_1 x_1 \ldots Q'_i x_i \in D'(x_i) \ldots Q_n x_n.\mathcal{C}; \\ &\mathcal{Q}.(c_1 \ldots c_j \ldots c_m)[c'_j] \;\hat{=}\; \mathcal{Q}.(c_1 \ldots c'_j \ldots c_m). \end{aligned}$$

This can then be extended to a set of relaxations R: $\phi[\emptyset] = \phi$, $\phi[\{r\} \cup R] = (\phi[r])[R]$.

Definition 5 (Ordering over Requirement Relaxations) *Given the set of possible relaxations, $\mathcal{R}(R)$, of a requirement R, we say that $(r_1 \in \mathcal{R}(R)) \sqsubseteq (r_2 \in \mathcal{R}(R))$ iff for any problem ϕ, if $\phi[r_1]$ is satisfiable then $\phi[r_2]$ is, necessarily. We further require that this partial order also be a* meet-semilattice*, i.e. that* greatest lower bounds *are guaranteed to exist: if $r_1, r_2 \in \mathcal{R}(R)$, then $r_1 \sqcap r_2$ is well-defined.*

The ordering operator and property we use to make formal the notion of relaxation *per se*: if $r_1 \sqsubset r_2$, then we say that r_2 is a (strict) relaxation of r_1. The greatest lower bound corresponds to the unique requirement relaxation which is as constraining as both of its arguments, but no more (that is, the greatest such element of the relaxation space). The lattice property follows naturally from the relaxation spaces we shall define, and is motivated by the observation that we progressively tighten approximations downwards, and that our relaxation spaces must have a unique least element. For clarity in that respect, we will define the meet operator directly. The partial order may be defined as $r_1 \sqsubseteq r_2 \;\hat{=}\; r_1 = (r_1 \sqcap r_2)$, and we note that in each case it is straightforward to derive a closed form for the comparison.

We now define the space of relaxations for constraints, the first form of relaxation we identified above. Essentially, a relaxation of a constraint involves adding additional allowed tuples into its relation.

Definition 6 (Requirement Relaxation for Constraints) *Given a constraint $c \;\hat{=}\; \langle var(c), sol(c) \rangle$ we define its relaxations in terms of adding additional allowed tuples to $sol(c)$ as follows:*

$$\mathcal{R}(c) \;\hat{=}\; \{sol'(c) : sol(c) \subseteq sol'(c) \subseteq \Pi_{x \in var(c)} D(x)\}.$$

The elements of $\mathcal{R}(c)$ form the usual lattice using intersection, that is, $\sqcap \;\hat{=}\; \cap$.

We now consider universal quantifier relaxation. Informally, the space of possible relaxations for a universal quantifier corresponds to restricting the quantifier to any one of the exponentially many subsets of the domain of the quantified variable, narrowing to a single choice of value, and thereafter widening to *existentially* quantified subsets. It therefore corresponds to the second, third and fourth forms of relaxation identified in Section 3.1. Of course, in practice, one can limit this relaxation space to a subset of tractable size.

Definition 7 (Requirement Relaxation for Universals) *Given a requirement, r, on a universally quantified variable x, i.e.* $r \mathrel{\hat{=}} \forall x \in D(x)$*, the set of relaxations* $\mathcal{R}(\forall x \in D(x))$ *is defined as:*

$$\begin{array}{rcl} \mathcal{R}(r) & \hat{=} & \{(\forall x \in D'(x)) : \emptyset \subseteq D'(x) \subseteq D(x)\} \\ & \cup & \{(\exists x \in D'(x)) : \emptyset \subseteq D'(x) \subseteq D(x)\}. \end{array}$$

The elements of $\mathcal{R}(\forall x \in D(x))$ *form the following meet-semilattice:*

$$\begin{array}{rcl} (\forall x \in D(x)) & \sqcap & (\forall x \in D'(x)) \\ & \hat{=} & (\forall x \in (D(x) \cup D'(x))); \\ (\exists x \in D(x)) & \sqcap & (\exists x \in D'(x)) \\ & \hat{=} & (\exists x \in (D(x) \cap D'(x))); \\ (\forall x \in D(x)) & \sqcap & (\exists x \in D'(x)) \\ & \hat{=} & (\forall x \in D(x)), \textit{if}\, D(x) \cap D'(x) \neq \emptyset; \\ (\forall x \in D(x)) & \sqcap & (\exists x \in D'(x)) \\ & \hat{=} & (\exists x \in \emptyset), \textit{if}\, D(x) \cap D'(x) = \emptyset. \end{array}$$

Note the existence of unique top and bottom points, corresponding to trivially- and un-satisfiable quantifications: $\top = (\forall x \in \emptyset), \bot = (\exists x \in \emptyset)$.

We can now define the space of possible relaxations of a given QCSP in terms of its constituent requirements in the natural way, in terms of the cross-product of the available relaxations for each, and similarly the associated comparison and meet operations.

Before we consider how explanations of falsity can be computed using relaxations, we consider two special cases of relaxation: namely, 1-point and 2-point relaxation spaces. The former, which we denote by $\mathcal{R}_1$, contains only the original constraint itself, and is useful when we wish to ensure that some constraints or quantifiers cannot be relaxed. The latter, $\mathcal{R}_2$, contains both the original constraint, and $\top$, the maximal relaxation; it is useful when we wish to model the case where we either maintain the requirement or completely relax it.

4 From Relaxations to Explanations

We now consider explanations for the QCSP. A familiar notion of explanation is that based on *minimal conflicts* [Junker, 2004]. As we have considered quantification, which is not readily presentable in terms of elements of a conflict *set*, and further, more general relaxations of constraints than their removal, it is necessary to generalise this notion to that of a *minimally conflicting explanation*, or alternatively, a *maximally relaxed conflict-based explanation*. We will henceforth refer to these as maximal, following the relaxation ordering previously defined. We define this notion with respect to a (typically incomplete) consistency propagation method Π, such as QAC [Bordeaux and Monfroy, 2002], in a similar way to Junker [Junker, 2004].

Definition 8 (Maximally Relaxed Explanation) *Given a consistency propagator* Π*, a* maximally relaxed (Π-conflict-based) explanation *of a* Π*-inconsistent* QCSP ϕ*, is a maximally relaxed* QCSP*,* X*, that is inconsistent with respect to* Π*; i.e. such that* $\phi \sqsubseteq X$*,* $\bot \in \Pi(X)$ *and for all* X' *such that* $X \sqsubset X', \bot \notin \Pi(X')$.

4.1 Explanation Algorithms

We adopt a scheme similar to that of the QUICKXPLAIN 'family' of algorithms [Junker, 2001; 2004], modified and extended in the following way. We replace the removal (respectively, addition) of a constraint with a *specified requirement relaxation* (resp. tightening). The specified relaxations form a semilattice, as described in the previous section. This could, in principle, be *every* possible relaxation, but at a minimum we generally wish to consider the specified relaxations corresponding to the removal (resp. imposition) of each constraint, and the relaxation of a universal quantifier to an existential one: that is, corresponding to relaxation space $\mathcal{R}_2$. We may further restrict the set of specified relaxations on the basis that some constraints, and some variables, are intrinsically environmental, and that they should participate in every explanation; this corresponds to using relaxation space $\mathcal{R}_1$.

As consideration of every possible relaxation is in general completely intractable, it will generally be desirable to specify a much smaller set. This may be done in broadly two ways. Firstly, as part of the modelling process, an expert specifying the class of problem instances under consideration can give the sets of possible requirement relaxations that are meaningful. Secondly, we can consider a user-driven refinement step, after initial exploration of the available explanations, based on a base step choice of available relaxations, when these have been found to be unsatisfactory.

4.2 The SIMPLEQUANTIFIEDXPLAIN Algorithm

When the set of specified relaxations are binary, i.e. requirements are either in place or fully relaxed, we can use the equivalent of Junker's algorithms, modified simply to use, instead of the imposition and removal of a constraint, the choice between the original and relaxed requirement throughout. We firstly reformulate Junker's REPLAYXPLAIN [Junker, 2001] in terms of relaxations in Algorithm 1, thereby generalising it to QCSPs with (at most) one distinct relaxation available for each of the original requirements, i.e. a requirement is either present, or fully relaxed. We call this modified procedure SIMPLEQUANTIFIEDXPLAIN. We provide as arguments ϕ, a QCSP, and R, a set of available relaxations, which here should only contain the greatest relaxations allowed for each requirement, r_i, that is relaxable in the specified relaxation $\mathcal{R}_s$, i.e. $R = \{\top_i : i \in [1, m+n], \top_i = \bigsqcup \mathcal{R}_s(r_i), |\mathcal{R}_s(r_i)| > 1\}$. (This is only well-defined for spaces with a unique maximal relaxation.) We begin with a maximally relaxed problem, using all of the available relaxations R, then build a working set W of relaxations we then iteratively *omit*, until a $\Pi-$inconsistency occurs. At that point we conclude that the relaxation r last added to W (that is, last removed from the subproblem) can be added to a set X corresponding to a partial conflict, and that no *further* relaxations need to be considered, as some subset of W must be minimal, and still yield an inconsistency. We iteratively continue doing this until the set of possible further relaxations is exhausted, at which point X can be used to relax the original problem ϕ to yield a maximally relaxed inconsistent problem, $\phi[R - X]$, which is our explanation.

Algorithm 1: SIMPLEQUANTIFIEDXPLAIN(ϕ, R)

Input : A QCSP ϕ; a set of available single-step relaxations $R \subseteq \{\top_1, \ldots, \top_{n+m}\}$.
Output: A maximally relaxed conflict-based explanation for ϕ.

```
if ⊥ ∉ Π(φ) then return exception "no conflict";
if R = ∅ then return φ;
X ← ∅;
R0 ← R;
while R0 ≠ ∅ do
    W ← ∅;
    while ⊥ ∉ Π(φ[R − (X ∪ W)]) do
        select (any) r from R0;
        W ← W ∪ {r};
        R0 ← R0 − {r};
    X ← X ∪ {r};
    R0 ← W − {r};
return φ[R − X];
```

Note that we do not, as with REPLAYXPLAIN and Junker's other explanation procedures, split the input into background and user constraints, and nor do we restrict the output to only user constraints, since the syntax of quantified problems precludes us from doing so. We are instead able to designate some constraints and some quantified variables as *not relaxable*, which then can be regarded as playing the same role as background constraints. Thus implicitly, user requirements must have at least one available relaxation. One difference in the form of this algorithm is that the 'background' constraints are included in the explanation. Background constraints may be removed from the output after the fact if desired. Observe that the two are equivalent in this sense: if REPLAYXPLAIN (C, U) = X, and SIMPLEQUANTIFIEDXPLAIN ($C \cup U$, $\{\top_i : c_i \in U\}$) = X_q, and $X_q = X \cup C$. (Assuming similar ordering of relaxations (constraints) in each case.) The remaining algorithms presented by Junker may be reformulated in terms of relaxations similarly. (Discussion of the correctness and complexity of this algorithm is deferred until after the presentation of QUANTIFIEDXPLAIN, of which it is a special case.)

Example 7 (SIMPLEQUANTIFIEDXPLAIN) *Consider the execution of* SIMPLEQUANTIFIEDXPLAIN$(\phi, \mathcal{R})$*, where* $\phi \mathrel{\hat{=}} \exists x_1 \in \{1,2\} \forall x_2 \in \{1,2,3\}.\{x_1 < x_2\}$ *and* $R \mathrel{\hat{=}} \{x_2 = \top, c = \top\}$*, a trace of the progress of which is shown in Figure 1. Note that in this case, the explanation is unique, and identical to the original problem, as no relaxation is possible without introducing consistency. Had different choices of relaxations been made, a different trace would have occurred, but the same final result.* ▲

4.3 The QUANTIFIEDXPLAIN Algorithm

We now consider QUANTIFIEDXPLAIN (Algorithm 2), a more general algorithm allowing for requirement relaxation lattices of arbitrary size. Once again we follow the basic structure of REPLAYEXPLAIN, but now the key difference is that instead of successively adding constraints to a conflict set (or as in SIMPLEQUANTIFIEDXPLAIN, removing available relaxations), we instead consider a current candidate relaxation comprising one element from each relaxation space, $r_1 \ldots r_{m+n}$, and use $\sqsubset$ to obtain successive approximations to a maximally relaxed explanation. We begin with a maximal relaxation of *each* requirement, and then progressively tighten these one at a time: we select an i such that r_i may be assigned a new value strictly less than the old one, tightening by a minimal amount at each step to ensure maximality of the final relaxation of the whole problem. When an inconsistency is detected, we eliminate all relaxations tighter than or incomparable to the current approximation from future consideration, as unnecessary for a maximally relaxed explanation. At the same time, we restrict the relaxation space for the last-relaxed requirement, $\mathcal{R}_i$, to ensure that that requirement may not be as relaxed as the earlier value, r'_i, that did not produce an inconsistency, as we have guaranteed that that value cannot take part in the explanation currently under construction. We then repeat this process with the relaxation spaces so restricted, until eventually only one possibility remains for each relaxation, thus fully determining the chosen explanation. The complexity of this algorithm (and that of SIMPLEQUANTIFIEDXPLAIN) is equivalent to that of Junker's REPLAYXPLAIN, given a consistency operator of similar complexity, and similar numbers of available relaxations. However, in the QCSP case, there is scope for each of those to be significantly larger.

$\phi_0 = \phi[R - (X \cup W)]$	$\perp \in \Pi(\phi_0)$?
$\exists x_1 \in \{1,2\} \exists x_2 \in \{1,2,3\}.\emptyset$	no
$\exists x_1 \in \{1,2\} \exists x_2 \in \{1,2,3\}.\{x_1 < x_2\}$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1,2,3\}.\{x_1 < x_2\}$	yes
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1,2,3\}.\emptyset$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1,2,3\}.\{x_1 < x_2\}$	yes

Figure 1: SIMPLEQUANTIFIEDXPLAIN: trace on Example 7.

Algorithm 2: QUANTIFIEDXPLAIN$(\phi, \mathcal{R})$

Input : A QCSP ϕ; a set of relaxation spaces for each quantifier and constraint, $\mathcal{R} \subseteq \mathcal{R}(\phi)$.
Output: A maximally relaxed conflict-based explanation for ϕ.

```
if ⊥ ∉ Π(φ) then return exception "no conflict";
enumerate 𝓡 as 𝓡1 ... 𝓡m+n;
if ∀i ∈ [1, m+n] |𝓡i| = 1 then return φ;
while ∃i : |𝓡i| > 1 do
    foreach 𝓡i do choose an ri from maxima(𝓡i);
    if ⊥ ∈ Π(φ' = φ[{r1, ..., rm+n}]) then return φ';
    while ⊥ ∉ Π(φ[{r1, ..., rm+n}]) do
        choose an i s.t. ri ≠ ⨅(𝓡i);
        r'i ← ri;
        choose an ri from maxima{r : r ∈ 𝓡i, r ⊏ ri};
    𝓡i ← {r : r ∈ 𝓡i, r'i ⋢ r};
    foreach 𝓡j do 𝓡j ← {r : r ∈ 𝓡j, rj ⊑ r};
return φ[{r1, ..., rm+n}];
```

Example 8 (QUANTIFIEDXPLAIN) *Consider the* QCSP $\phi \hat{=} \exists x_1 \in \{1,2\} \forall x_2 \in \{1,2,3\}.\{x_1 < x_2\}$ *and the execution of* QUANTIFIEDXPLAIN *on* ϕ *with the following relaxation spaces:* $\mathcal{R}_1(\exists x_1 \in \{1,2\})$*;* $\mathcal{R}_8(\forall x_2 \in D(x_2)) = \{(\forall x_2 \in D'(x_2) : \emptyset \subset D'(x_2) \subseteq D(x_2)\} \cup \{\exists x_2 \in D(x_2)\}$*; and* $\mathcal{R}_3(x_1 < x_2) = \{x_1 < x_2, x_1 \leq x_2, \top\}$*. We give a trace of* QUANTIFIEDXPLAIN *in Figure 2.* ▲

$\phi_0 = \phi[\{r_1, \ldots, r_{m+n}\}]$	$\perp \in \Pi(\phi_0)$
$\exists x_1 \in \{1,2\} \exists x_2 \in \{1,2,3\}.\emptyset$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1\}.\emptyset$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1\}.\{x_1 \leq x_2\}$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1\}.\{x_1 < x_2\}$	yes
$\exists x_1 \in \{1,2\} \exists x_2 \in \{1,2,3\}.\{x_1 < x_2\}$	no
$\exists x_1 \in \{1,2\} \forall x_2 \in \{1\}.\{x_1 < x_2\}$	yes

Figure 2: QUANTIFIEDXPLAIN: trace on Example 8.

The proof of correctness of QUANTIFIEDXPLAIN relies on the following properties, analogous to [Junker, 2001]:

Property 1 *If each relaxation space has a unique top element, and* $\perp \in \Pi(\phi[\{\top_1, \ldots, \top_{m+n}\}])$*, then the only maximally relaxed explanation is* $\phi[\{\top_1, \ldots, \top_{m+n}\}]$*.*

Property 2 *If* $\perp \notin \Pi(\phi[\{r_1, \ldots, r_{m+n}\}])$*, for all maximal* r_i*, then no conflict exists, and thus no explanation.*

Property 3 *If* $\perp \in \Pi(\phi[r_i])$*, and* X *is a maximally relaxed explanation for* ϕ*, given the relaxation spaces* $\mathcal{R}_1 \ldots \mathcal{R}'_i \ldots \mathcal{R}_{m+n}$*, then* X *is also a maximally relaxed explanation for* ϕ *given the relaxation spaces* $\mathcal{R}_1 \ldots \mathcal{R}_i \ldots \mathcal{R}_{m+n}$*, where* $\mathcal{R}'_i = \{r \in \mathcal{R}_i, r_i \sqsubseteq r\}$*.*

Property 4 *If* $\perp \in \Pi(\phi[r_i])$ *and* $\perp \notin \Pi(\phi[r'_i])$*, where* $r_i \sqsubset r'_i$*, then if* X *is a maximally relaxed explanation for* ϕ*, given the relaxation spaces* $\mathcal{R}_1 \ldots \mathcal{R}'_i \ldots \mathcal{R}_{m+n}$*, then* X *is also a maximally relaxed explanation for* ϕ *given the relaxation spaces* $\mathcal{R}_1 \ldots \mathcal{R}_i \ldots \mathcal{R}_{m+n}$*, where* $\mathcal{R}'_i = \{r \in \mathcal{R}_i, r'_i \not\sqsubseteq r\}$*.*

Theorem 1 (Maximally Relaxed Explanations) *If* $\perp \in \Pi(\phi)$*, and* $\mathcal{R}$ *is a set of available relaxations of some, all or none of the requirements of* ϕ*, and* QUANTIFIEDXPLAIN *(*ϕ*,* $\mathcal{R}$*) =* X*, then:* $\perp \in \Pi(X)$*, and if* $X' = X[r]$ *for some* $r \in \mathcal{R}$ *such that* $X \sqsubset X'$*, then* $\perp \notin \Pi(X')$*.*

5 Related Work

Benhamou and Goualard studied quantified interval constraints, with engineering applications in mind [Benhamou and Goualard, 2000]. More recently, researchers have begun to study how classical CSP concepts such as arc-consistency, satisfiability and interchangeability can be extended to the QCSP [Bordeaux *et al.*, 2005]. A number of techniques for solving the QCSP have been proposed based on encodings to QBF [Gent *et al.*, 2004], as well as using a systematic QCSP solver, called QCSP-Solve, directly [Gent *et al.*, 2005]. A variety of repair-based methods have also been proposed [Stergiou, 2005]. A deep study of the complexity of the QCSP is also available [Chen, 2004]. Our work on relaxation and explanations is the first on these important aspects of reasoning about QCSPs.

6 Conclusions and Future Work

We have extended the concepts of problem relaxation and explanation of inconsistency to the QCSP. We showed that the generality of the QCSP allows for the definition of a number of different forms of relaxation not available in classical CSP. We presented an algorithm called QUANTIFIEDXPLAIN for computing minimally conflicting explanations for QCSPs. Our work on relaxation and explanations for the QCSP is the first study on these interesting and practical aspects of quantified constraint reasoning. We plan to extend the explanation algorithm to deal uniformly with the final class of relaxation, that of quantifier moving, and to establish complexity comparable to Junker's QUICKXPLAIN algorithm. Another important area is that of *preferred* explanations, with which it would be possible to improve the situation presented here, where an arbitrary explanation is generated from those possible.

Acknowledgements. This work was supported by Science Foundation Ireland (Grant 03/CE3/I405). We thank Hubie Chen, Helene Fargier and Nic Wilson for their comments.

References

[Benhamou and Goualard, 2000] F. Benhamou and F. Goualard. Universally quantified interval constraints. In *CP*, pages 67–82, 2000.

[Bordeaux and Monfroy, 2002] L. Bordeaux and E. Monfroy. Beyond NP: Arc-consistency for quantified constraints. In *CP*, pages 371–386, 2002.

[Bordeaux *et al.*, 2005] L. Bordeaux, M. Cadoli, and T. Mancini. CSP properties for quantified constraints: Definitions and complexity. In *AAAI*, pages 360–365, 2005.

[Chen, 2004] H. Chen. *The Computational Complexity of Quantified Constraint Satisfaction.* PhD thesis, Cornell University, August 2004.

[Freuder and Wallace, 1992] E.C. Freuder and R.J. Wallace. Partial constraint satisfaction. *Artif. Intell.*, 58(1-3):21–70, 1992.

[Gent *et al.*, 2004] I.P. Gent, P. Nightingale, and A. Rowley. Encoding quantified CSPs as quantified boolean formulae. In *ECAI*, pages 176–180, 2004.

[Gent *et al.*, 2005] I.P. Gent, P. Nightingale, and K. Stergiou. QCSP-solve: A solver for quantified constraint satisfaction problems. In *IJCAI*, pages 138–143, 2005.

[Junker, 2001] U. Junker. QUICKXPLAIN: Conflict detection for arbitrary constraint propagation algorithms. In *IJCAI'01 Workshop on Modelling and Solving problems with constraints*, 2001.

[Junker, 2004] U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *AAAI*, pages 167–172, 2004.

[Mackworth, 1977] A.K. Mackworth. Consistency in networks of relations. *Artif. Intell.*, 8(1):99–118, 1977.

[Stergiou, 2005] K. Stergiou. Repair-based methods for quantified CSPs. In *CP*, pages 652–666, 2005.