Disjunctions, Independence, Refinements

Mathias Broxvall and Peter Jonsson
Department of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden
email: {matbr, petej}@ida.liu.se

Jochen Renz
Institut für Informationssysteme
Technische Universität Wien
Favoritenstrasse 9
A-1040 Wien, Austria
email: renz@dbai.tuwien.ac.at

Abstract

An important question in constraint satisfaction is how to restrict the problem to ensure tractability (since the general problem is NP-hard). The use of disjunctions has proven to be a useful method for constructing tractable constraint classes from existing classes; the well-known 'max-closed' and 'ORD-Horn' constraints are examples of tractable classes that can be constructed this way. Three sufficient conditions (the guaranteed satisfaction property, 1-independence and 2-independence) that each ensure the tractability of constraints combined by disjunctions have been proposed in the literature. We show that these conditions are both necessary and sufficient for tractability in three different natural classes of disjunctive constraints. This suggests that deciding this kind of property is a very important task when dealing with disjunctive constraints. We provide a simple, automatic method for checking the 1-independence property—this method is applicable whenever the consistency of the constraints under consideration can be decided by path-consistency. Our method builds on a connection between independence and refinements (which is a way of reducing one constraint satisfaction problem to another.)

Keywords: Constraint satisfaction, disjunctive constraints, tractability

1 Introduction

The constraint satisfaction problem provides a natural framework for expressing many combinatorial problems in computer science. Since the general problem is NP-hard [14], an important question is how to restrict the problem to ensure tractability. This research has mainly followed two different paths: restricting the scope of the constraints [10, 11], i.e. which variables may be constrained with other variables, or restricting the constraints [9, 12, 19], i.e. the allowed values for mutually constrained variables. In this paper, we will only consider problems where the constraints are restricted.

Quite a large number of tractable subclasses of the CSP problem have been identified in the literature. Due to the lack of systematicity in this search, it is of considerable interest to investigate how tractable constraint types may be combined in order to yield more general problems which are still tractable. Cohen *et al.* [8] have studied so-called 'disjunctive constraints', i.e. constraints which have the form of the disjunction of two constraints of specified types. They identified certain properties which allow for new tractable constraint classes to be constructed from existing classes. Several important classes of tractable constraints can be obtained by their method such as the Horn and Krom fragments of propositional logic, the ORD-Horn class [15] and the classes of max-closed and connected row-convex constraints [12, 9].

The investigation of disjunctive constraints was continued in Broxvall & Jonsson [5] where all tractable disjunctive classes for reasoning about partially and totally ordered time were identified. Somewhat surprisingly, all of these tractable classes can be obtained by using 1-independence. This observation raised the question whether tractable disjunctive constraints can be completely characterised by these kind of properties. We partially answer this question in this paper.

We consider three different properties, known as the guaranteed satisfaction (GS) property, 1-independence and 2-independence [8]. Let Γ and Δ be two sets of relations such that the CSP problem over $\Gamma \cup \Delta$ is tractable. In short, we prove the following:

- Let the set Δ^* contain all possible disjunctive relations over Δ. The CSP problem for this set is tractable if and only if Δ has the GS property.

- Let the set $\Gamma \overset{\times}{\vee} \Delta^*$ contain all disjunctive relations over $\Gamma \cup \Delta$ where relations in Γ are allowed to appear at most once in a disjunction (compare with the Horn fragment of propositional logic). The CSP problem for this set is tractable if and only if Δ is 1-independent of Γ.

- Consider the set $\Gamma \cup \Delta^2$ where Δ^2 contains all disjunctive relations over Δ containing at most two disjuncts (compare with the Krom fragment of propositional logic). The CSP problem for this set is tractable if and only if Δ is 2-independent of Γ.

Our results are obtained by using the definition of the disjunction combinator $\overset{\times}{\vee}$ proposed in [5] instead of the original definition in [8]. This change makes the result cleaner since we do not have to take care of a number of pathological special cases. This issue is discussed in greater detail in the paper.

These results suggest that automatic methods for checking these properties may be very useful when working with disjunctive constraints. Also, it is hardly surprising that deciding these properties is a highly non-trivial task in many cases. For classes of binary constraints where satisfiability can be decided by checking path-consistency, we present a fairly simple method for verifying the 1-independence property. This method builds on a somewhat surprising connection between 1-independence and *refinements* [16]. Loosely speaking, a refinement is a way of reducing one CSP problem to another and it has the property that if the second problem can be decided by path-consistency, then path-consistency decides the first problem, too. Refinements were successful in proving tractability of large subsets of RCC-8 as well as Allen's Interval Algebra [16]. One important aspect of refinements is that their correctness can be easily checked by a computer-assisted analysis which implies that 1-independence can be automatically checked in many cases. To demonstrate the usefulness of our method, we show that all previously known independence results for the time point algebras for partially and totally ordered time [5] can be derived automatically. This raises the question whether our method is complete or not—unfortunately, we are not able to answer this question in its full generality.

The paper is organized as follows: In Section 2 we give an overview of the basic definitions concerning CSPs, disjunctions and refinements. Section 3 contains the main complexity results for combining constraints with disjunctions. In Section 4 we relate refinements and 1-independence and prove the connection between them. We also exemplify how the method can be used for identifying tractable disjunctive constraints. Finally, the last section contains some discussions and conclusions of the results presented earlier. Some of the results in Section 4 have previously been presented in a conference paper [6].

2 Preliminaries

This section consists of three parts where we define the constraint satisfaction problem, provide some background material concerning disjunctions and describe the refinement method.

2.1 The constraint satisfaction problem

Let $\mathcal{S}$ be a set of relations over some domain D (of *values*) and let V be a set of variables. The relations in $\mathcal{S}$ may be of arbitrary arity and the domain D is not necessarily finite. Let $R \in \mathcal{S}$ be a relation of arity a and $x \in V^a$ (where V^a denotes the a-fold cartesian product of V). We write $R(x)$ (a *constraint*) to denote that the variables in x are related by R. This definition allows the use of repeated variables in the scope of a constraint, e.g. $R(x, y, x)$. For any constraint $c = R(x)$, let $\mathrm{Rel}(c) = R$. The consistency problem $\mathrm{CspSat}(\mathcal{S})$ is defined as follows:

Instance: A tuple (V, C) where V is a set of variables and C is a finite set of constraints over V, where for each $c \in C$, $\mathrm{Rel}(c) \in \mathcal{S}$.

Question: Is there a *satisfying instantiation* of the variables, i.e. a total function $f : V \to D$ such that for all $R(x_1, \ldots, x_a) \in C$, $(f(x_1), \ldots, f(x_a)) \in R$?

Given an instance Θ of $\mathrm{CspSat}(\mathcal{S})$, let $\mathrm{Mods}(\Theta)$ denote the class of models of Θ (i.e. the satisfying instantiations) and $\mathrm{Vars}(\Theta)$ the variables appearing in Θ. Let $\perp$ denote the empty relation (of arbitrary arity).

2.2 Basics of disjunctions

We begin by introducing operators for combining relations with disjunctions.

Definition 1 Let R_1, R_2 be relations of arity i, j and define the disjunction $R_1 \vee R_2$ of arity $i + j$ as follows:

$$\begin{aligned} R_1 \vee R_2 \;=\; & \{(x_1, \ldots, x_{i+j}) \in D^{i+j} | (x_1, \ldots, x_i) \in R_1 \vee \\ & (x_{i+1}, \ldots, x_{i+j}) \in R_2\} \end{aligned}$$

Thus, the disjunction of two relations with arity i, j is the relation with arity $i+j$ satisfying either of the two relations.

To give a concrete example, let $D = \{0, 1\}$ and let the relations $\mathsf{and} = \{\langle 1, 1\rangle\}$ and $\mathsf{xor} = \{\langle 0, 1\rangle, \langle 1, 0\rangle\}$ be given. The disjunction of and and xor is given by:

$$\mathsf{and} \vee \mathsf{xor} = \left\{ \begin{array}{l} \langle 0,0,0,1\rangle, \langle 0,1,0,1\rangle, \langle 1,0,0,1\rangle, \langle 1,1,0,1\rangle, \\ \langle 0,0,1,0\rangle, \langle 0,1,1,0\rangle, \langle 1,0,1,0\rangle, \langle 1,1,1,0\rangle, \\ \langle 1,1,0,0\rangle, \langle 1,1,0,1\rangle, \langle 1,1,1,0\rangle, \langle 1,1,1,1\rangle \end{array} \right\}$$

We see that the constraint $(\mathsf{and} \vee \mathsf{xor})(x, y, x, z)$ is satisfiable when x, y and z has, for instance, been instantiated to $1, 0, 0$, respectively.

The definition of disjunction can easily be extended to sets of relations.

Definition 2 Let Γ_1, Γ_2 be sets of relations and define the disjunction $\Gamma_1 \overset{\times}{\vee} \Gamma_2$ as follows:

$$\Gamma_1 \overset{\times}{\vee} \Gamma_2 = \Gamma_1 \cup \Gamma_2 \cup \{R_1 \vee R_2 \mid R_1 \in \Gamma_1, R_2 \in \Gamma_2\}$$

The disjunction of two sets of relations $\Gamma_1 \stackrel{\times}{\vee} \Gamma_2$ is the set of disjunctions of each pair of relations in Γ_1, Γ_2 plus the sets Γ_1, Γ_2. It seems sensible to include Γ_1 and Γ_2 since one wants to have the choice of using the disjunction or not. Thus, our definition of $\stackrel{\times}{\vee}$ differs slightly from the definition given by Cohen *et al.* [8]; they define $\Gamma_1 \stackrel{\times}{\vee} \Gamma_2$ as $\{R_1 \vee R_2 \mid R_1 \in \Gamma_1, R_2 \in \Gamma_2\}$. The two definitions coincide if $\perp$ is included in both Γ_1 and Γ_2. Otherwise, the definitions are different and the implications of this are pointed out in Subsection 3.1. We will tacitly assume that $\perp$ is not a member of any set of relations that we consider. Note that $\textsc{CspSat}(\Gamma)$ has the same complexity as $\textsc{CspSat}(\Gamma \cup \{\perp\})$ up to polynomial-time reductions.

In many cases we shall be concerned with constraints that are specified by disjunctions of an arbitrary number of relations. Thus, we make the following definition: for any set of relations, Δ, define $\Delta^* = \bigcup_{i=0}^{\infty} \Delta^i$ where $\Delta^0 = \Delta$ and $\Delta^{i+1} = \Delta^i \stackrel{\times}{\vee} \Delta$.

For proving tractability of disjunctive constraints, a number of properties have been proposed in [8]:

Definition 3 Let Δ be a set of relations. If every instance $\textsc{CspSat}(\Delta)$ is satisfiable, then we say that Δ has the *guaranteed satisfaction* (GS) property.

$\textsc{CspSat}(\Delta^*)$ is clearly tractable if Δ has the GS property.

Definition 4 For any sets of relations Γ and Δ, define $\textsc{CspSat}_{\Delta \leq k}(\Gamma \cup \Delta)$ to be the subproblem of $\textsc{CspSat}(\Gamma \cup \Delta)$ consisting of all instances containing at most k constraints over the relations in Δ. We say that Δ is *k-independent* of Γ if the following condition holds: any set of constraints C in $\textsc{CspSat}(\Gamma \cup \Delta)$ has a solution provided every subset of C belonging to $\textsc{CspSat}_{\Delta \leq k}(\Gamma \cup \Delta)$ has a solution.

It is easy to see that if Δ is k-independent of Γ, then Δ is $k+1$-independent of Γ, too. The following result by Cohen *et al.* [8] demonstrates the usefulness of the independence property.

Theorem 5 Let Γ and Δ be sets of relations such that $\textsc{CspSat}(\Gamma \cup \Delta)$ is tractable. If Δ is 1-independent of Γ, then $\textsc{CspSat}(\Gamma \stackrel{\times}{\vee} \Delta^*)$ is tractable. If Γ is 2-independent of $\emptyset$, then $\textsc{CspSat}(\Gamma \stackrel{\times}{\vee} \Gamma)$ is tractable.

The notion of 1-independence can alternatively (but equivalently) be defined as follows: Let $C = \{c_1, \ldots, c_k\}$ and $D = \{d_1, \ldots, d_n\}$ be arbitrary finite sets of constraints over Γ and Δ, respectively. Then, Δ is 1-independent of Γ iff for every possible choice of C and D, the following holds: if $C \cup \{d_i\}$, $1 \leq i \leq n$, is satisfiable, then $C \cup D$ is satisfiable. Also note that Γ is 2-independent of $\emptyset$ if and only if for every constraint problem C over Γ having no solution, there exists a pair of constraints $c_i, c_j \in C$ such that $\{c_i, c_j\}$ has no solution.

2.3 Basics of the refinement method

We review the *refinement method* as introduced by Renz [16] in this subsection. For proofs and additional results, see [16] or its forthcoming journal version for more details [17].

So far the refinement method has been introduced for binary CSPs only. So, although we deal with n-ary constraints in this paper, the parts dealing with refinements apply only to binary constraints.

Let $\mathcal{A}$ be a finite set of jointly exhaustive and pairwise disjoint binary relations, also called *basic* relations, and $\mathcal{S} \subseteq 2^{\mathcal{A}}$. We denote the standard operations composition, intersection and converse by $\circ$, $\cap$ and $\cdot^{-1}$, respectively. Furthermore, we define the unary operation $\neg$ such that $\neg R = \mathcal{A} \backslash R$ for all relations $R \subseteq \mathcal{A}$ and let eq denote the binary equality relation.

A set of constraints is path-consistent if for any consistent assignment of two variables, there exists an assignment for every third variable such that the three assignments taken together are consistent. Path-consistency can be enforced by iteratively applying the following operation to every pair of variables x_i, x_j, until a fixed point is reached (R_{ij} specifies the relation between x_i and x_j):

$$\forall k : R_{ij} := R_{ij} \cap (R_{ik} \circ R_{kj}).$$

If the empty relation occurs during this process, the set is inconsistent, otherwise the resulting set is path-consistent.

A *refinement* of a constraint xRy is a constraint $xR'y$ such that $R' \subseteq R$. A refinement of a set of constraints Θ is a set of constraints Θ' such that every constraint of Θ' is a refinement of a constraint of Θ. It is clear that if Θ' has a solution, then also Θ has a solution.

In order to handle different refinements, a *refinement matrix* is used that contains for every relation $S \in \mathcal{S}$ all specified refinements.

Definition 6 A *refinement matrix* M of $\mathcal{S}$ has $|\mathcal{S}| \times 2^{|\mathcal{A}|}$ boolean entries such that for $S \in \mathcal{S}$, $R \in 2^{\mathcal{A}}$, $M[S][R] = true$ only if $R \subseteq S$, i.e. R is a refinement of S.

Definition 7 Let $\Delta \subseteq \mathcal{S}$. M^{Δ} is the Δ-*refinement matrix* of $\mathcal{S}$ if for every $S \in \mathcal{S}$, $M^{\Delta}[S][S'] = \mathsf{true}$ iff

1. there exists a relation $R \in \Delta$ such that $S' = S \cap R$ and $S' \neq \emptyset$; or
2. $S' = S$.

The basic idea of the refinement method [16] is to exploit that the path-consistency algorithm only looks at triples of constraints and that refinements of constraints are passed from triple to triple. Thus, the possible number of different triples over a set of relations $\mathcal{S}$ as well as the number of refinements of these triples is

Algorithm: Check-Refinements
Input: A set $\mathcal{S}$ and a refinement matrix M of $\mathcal{S}$.
Output: fail if the refinements specified in M can make a path-consistent triple of constraints over $\mathcal{S}$ inconsistent; succeed otherwise.

1. $changes \leftarrow$ true
2. while $changes$ do
3. $\quad oldM \leftarrow M$
4. $\quad$ for every path-consistent triple $T = (R_{12}, R_{23}, R_{13})$ of relations over $\mathcal{S}$ do
5. $\quad\quad$ for every refinement $T' = (R'_{12}, R'_{23}, R'_{13})$ of T with $oldM[R_{12}][R'_{12}] = oldM[R_{23}][R'_{23}] = oldM[R_{13}][R'_{13}] =$true do
6. $\quad\quad\quad T'' \leftarrow$ Path-Consistency(T')
7. $\quad\quad\quad$ if $T'' = (R''_{12}, R''_{23}, R''_{13})$ contains the empty relation then return fail
8. $\quad\quad\quad$ else do $M[R_{12}][R''_{12}] \leftarrow$ true, $M[R_{23}][R''_{23}] \leftarrow$ true, $M[R_{13}][R''_{13}] \leftarrow$ true
9. $\quad$ if $M = oldM$ then $changes \leftarrow$ false
10. return succeed

Figure 1: Algorithm Check-Refinements [16]

limited, although there is an infinite number of different sets of constraints Θ over $\mathcal{S}$. Therefore, it is possible to extract properties of a set of relations $\mathcal{S}$ by just analyzing a limited number of triples of constraints over $\mathcal{S}$. This is done by the algorithm Check-Refinements (see Figure 1) which takes as input a set of relations $\mathcal{S}$ and a refinement matrix M of $\mathcal{S}$ and either succeeds or fails. A *triple* (R, S, T) of relations denotes the following CSP problem on three variables: $\{xRz, xSy, yTz\}$. Since $\mathcal{A}$ is a finite set of relations, M can be changed only a finite number of times, so the algorithm always terminates.

If Check-Refinements$(\mathcal{S}, M)$ returns succeed, we have checked *all* possible refinements of *every* path-consistent triple of variables as given by the refinement matrix M. Thus, applying these refinements to a path-consistent set of constraints can *never* result in an inconsistency when enforcing path-consistency. This is stated in the following theorem.

Theorem 8 (Renz [16]) Let $\mathcal{S}$ be a set of relations that can be decided by path-consistency, M a refinement matrix of $\mathcal{S}$ and assume that Check-Refinements$(\mathcal{S}, M)$ returns succeed. For every path-consistent set Θ of constraints over $\mathcal{S}$, the following holds: for every refinement Θ' of Θ such that $x_i R' x_j \in \Theta'$ only if $x_i R x_j \in \Theta$ and $M[R][R'] = \mathsf{true}$, Θ' has a solution.

The refinement method, thus, simply consists of running the algorithm Check-Refinements on a set of relations $\mathcal{S}$ and a refinement matrix M. We say that $\mathcal{S}$ can be *refined* by M, if Check-Refinements$(\mathcal{S}, M)$ returns succeed.

Renz [16] used the refinement method in a different way, namely, for showing that path-consistency decides a set of relations $\mathcal{S}$: Assume that path-consistency decides consistency for a set of relations $\mathcal{T}$. If Check-Refinements$(\mathcal{S}, M)$ returns succeed and if the resulting refinement matrix M' contains for each relation $S \in \mathcal{S}$ a relation $T_S \in T$, i.e. $M'[S][T_S] = true$, then path-consistency decides consistency of $\mathcal{S}$. It turned out that by using the refinement matrix $M^{\neq}$ it was possible to prove tractability for all maximal tractable subsets of RCC-8 and the Interval Algebra which contain all basic relations.

3 Tractable Disjunctions

We shall now show the close connections between tractable disjunctive constraints and the GS/independence properties. Our main results are the following: Let Γ and Δ be two sets of relations such that CspSat$(\Gamma \cup \Delta)$ is tractable. Then,

(1) CspSat(Δ^*) is tractable iff Δ has the GS property;

(2) CspSat$(\Gamma \overset{\times}{\vee} \Delta^*)$ is tractable iff Δ is 1-independent of Γ; and

(3) CspSat$(\Gamma \cup \Delta^2)$ is tractable iff Δ is 2-independent of Γ.

If these conditions are not met, then CspSat(Δ^*), CspSat$(\Gamma \vee \Delta^*)$ and/or CspSat$(\Gamma \cup \Delta^2)$ are NP-complete. The proofs of (1)–(3) can be found in Subsections 3.1–3.3, respectively. An interesting question is whether (3) can be strengthened to ensure tractability of CspSat$(\Gamma \overset{\times}{\vee} \Delta)$. We demonstrate that this does not hold in general at the end of Subsection 3.3.

The NP-completeness results are based on reductions from the following two NP-complete problems:

3-Sat
Instance: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$.
Question: Is there a satisfying truth assignment for C?

3-Colourability
Instance: Undirected graph $G = (V, E)$.
Question: Does there exist a function $f : V \rightarrow \{0, 1, 2\}$ such that $f(u) \neq f(v)$

whenever $\{u, v\} \in E$?

Before we proceed, we need to prove that the problems we consider are members of NP.

Lemma 9 Assume that $\mathcal{S}$ is a tractable set of relations. For any set $\mathcal{S}'$ of relations constructed using $\overset{\times}{\vee}$ and the relations in $\mathcal{S}$, CSPSAT($\mathcal{S}'$) is in NP.

Proof: Non-deterministically choose one atomic constraint from every disjunctive constraint (we assume, without loss of generality, that there exists polynomial-time computable decomposition operators for the disjunctive constraints) and show that the resulting set of constraints is satisfiable. Since CSPSAT($\mathcal{S}$) is tractable[1], CSPSAT($\mathcal{S}'$) is in NP. □

3.1 The guaranteed satisfaction property

We begin by studying the (admittedly trivial) GS property. The proof idea will, however, turn out to be very useful for proving results about the independence properties.

Theorem 10 The following statements are equivalent:

1. Δ has the GS property;
2. Δ is 1-independent of $\emptyset$;
3. CSPSAT(Δ^*) is tractable;
4. CSPSAT(Δ^3) is tractable;

Otherwise, CSPSAT(Δ^3) and CSPSAT(Δ^*) are NP-complete.

Proof: We show that $(1) \Rightarrow (3) \Rightarrow (4) \Rightarrow (1)$ and $(1) \Leftrightarrow (2)$.

The implication $(1) \Rightarrow (3)$ is trivial and $(3) \Rightarrow (4)$ follows from the fact that $\Delta^3 \subseteq \Delta^*$. To show that $(4) \Rightarrow (1)$, we assume the opposite, i.e. CSPSAT(Δ^3) is tractable but Δ does not have the GS property. This implies that there exists a set of constraints $H = \{h_1, \ldots, h_n\}$ over Δ such that H is not satisfiable. Note that $|H| > 1$ since we do not allow the relation $\perp$. We choose H to be minimal; i.e. $|H|$ is as small as possible. This implies that every strict subset $H' \subset H$ is satisfiable. Finally, consider the set $\mathcal{H} = \{h_1 \vee h_2\} \cup (H - \{h_1, h_2\})$ and note that in any model of $\mathcal{H}$, either h_1 or h_2 hold, but not both.

To prove NP-hardness, we show that 3-SAT can be transformed to CSPSAT(Δ^3) in polynomial time; membership in NP follows from Lemma 9. Arbitrarily choose

[1]It is actually sufficient that CSPSAT($\mathcal{S}$) is in NP.

a 3-SAT formula $F = c_1 \wedge \ldots \wedge c_n$ over the variables $p_1, \ldots, p_m$. We incrementally construct an instance of CSPSAT(Δ^3) that is satisfiable iff F is satisfiable.

For each variable p_i, introduce a fresh copy of the set $\mathcal{H}$ (i.e. the copies of $\mathcal{H}$ are over disjoint sets of variables) where we denote the 'important' relations h_1 and h_2 as h_t^i and h_f^i, respectively. As we noted earlier, this will force either h_t^i or h_f^i to hold in any model but not both. We interpret h_t^i as 'p_i is true' and h_f^i as 'p_i is false'.

For each clause c_i, it is now easy to add a disjunction corresponding to the clause: for instance, $(p_i \vee \neg p_j \vee p_k)$ is translated to $h_t^i \vee h_f^j \vee h_t^k$. Obviously, the resulting set of constraints (which trivially can be computed in polynomial time) is an instance of CSPSAT(Δ^3) and is satisfiable iff F is satisfiable.

Finally, we show that (1) $\Leftrightarrow$ (2). The only-if direction is obvious so we prove the other direction. Assume to the contrary that there exists a set of constraints H over Δ such that H is not satisfiable. Since Δ is 1-independent of $\emptyset$, this implies that there must be a single constraint $h \in H$ that is not satisfiable—in other words, $\mathrm{Rel}(h)$ is the empty relation and we have a contradiction. $\Box$

This result does not hold if the original definition of $\overset{\times}{\vee}$ [8] is used (see Section 2 for the exact definition). Assume that Δ has the GS property. Then, $\perp \notin \Delta$. Assume furthermore that Γ is an arbitrary set of relations (we do not even require that CSPSAT(Γ) is tractable). Then, $\Gamma \overset{\times}{\vee} \Delta$ is tractable! This follows from the fact that $\Gamma \not\subseteq \Gamma \overset{\times}{\vee} \Delta$; every possible member of $\Gamma \overset{\times}{\vee} \Delta$ is either of the form $R(x_1, \ldots, x_{arity(R)})$ where $R \in \Delta$ or $R(x_1, \ldots, x_{arity(R)}) \vee S(y_1, \ldots, y_{arity(S)})$ where $R \in \Delta$ and $S \in \Gamma$. Hence, the GS property ensures that every instance of CSPSAT($\Gamma \overset{\times}{\vee} \Delta$) is satisfiable. It seems counter-intutitive that $\Gamma \cup \Delta$ can be a computationally harder problem than $\Gamma \overset{\times}{\vee} \Delta$ which explains why we have modified the definition of $\overset{\times}{\vee}$.

There are also technical reasons for defining $\overset{\times}{\vee}$ the way we have done. For instance, the result in the next section would be very different. It simply states that CSPSAT($\Gamma \overset{\times}{\vee} \Delta^*$) is tractable iff CSPSAT($\Gamma \cup \Delta$) is tractable and Δ is 1-independent of Γ. With the original definition of $\overset{\times}{\vee}$, we would need to take care of several cases; one of them is that CSPSAT($\Gamma \overset{\times}{\vee} \Delta^*$) is tractable if Δ has the GS property but Δ is not 1-independent of Γ (which once again is a 'strange' case where CSPSAT($\Gamma \cup \Delta$) may be computationally harder than CSPSAT($\Gamma \overset{\times}{\vee} \Delta^*$).

3.2 1-Independence

The proof presented here is a slight variation of the proof of Theorem 10 so we only sketch the proof.

Theorem 11 The following statements are equivalent:

1. Δ is 1-independent of Γ;

2. CspSat($\Gamma \overset{\times}{\vee} \Delta^*$) is tractable;

3. CspSat($\Gamma \cup \Delta^3$) is tractable;

Otherwise, CspSat($\Gamma \cup \Delta^3$) and CspSat($\Gamma \overset{\times}{\vee} \Delta^*$) are NP-complete.

Proof: The implications (1) $\Rightarrow$ (2) follows from Theorem 5 and (2) $\Rightarrow$ (3) is trivial since $\Gamma \cup \Delta^3 \subseteq \Gamma \overset{\times}{\vee} \Delta^*$. To show that (3) $\Rightarrow$ (1), we assume the opposite, i.e. CspSat($\Gamma \cup \Delta^3$) is tractable but Δ is not 1-independent of Γ. This implies that there exists a set of constraints X over Γ and a set $H = \{h_1, \ldots, h_n\}$ over Δ such that $X \cup \{h_i\}$ is satisfiable for every $1 \leq i \leq n$ but $X \cup H$ is not satisfiable. Choose X and H such that $|H|$ is as small as possible and note that $|H| \geq 2$. The existence of a set $H' \subset H$ such that $X \cup H'$ is not satisfiable contradicts the minimality of H so $X \cup H'$ is satisfiable for all $H' \subset H$. Finally, consider the set $\mathcal{X} = X \cup \{h_1 \vee h_2\} \cup (H - \{h_1, h_2\})$ and note that in any model of $\mathcal{X}$, either h_1 or h_2 hold, but not both. The result can now easily be shown by a reduction from 3-Sat that is analogous to the reduction employed in the proof of Theorem 10. $\Box$

By combining Theorems 10 and 11, we see that whenever Δ is 1-independent of Γ, Δ must have the GS property—this observation can significantly simplify the search for sets of 1-independent relations.

3.3 2-Independence

The proof of this case consists of two parts; the first part strengthens a tractability result by Cohen *et al.* [8] while the second part is a hardness result in the style of Theorems 10 and 11. The reduction is quite different, though, and is based on 3-Colourability instead of 3-Sat. In the end of this subsection (Theorem 14), we complement this positive result with a negative result showing that 2-independence is not sufficient for ensuring tractability of CspSat($\Gamma \overset{\times}{\vee} \Delta$).

Theorem 12 CspSat($\Gamma \cup \Delta^2$) is tractable iff Δ is 2-independent of Γ. Otherwise, CspSat($\Gamma \cup \Delta^2$) is NP-complete.

Cohen *et al.* have shown that CspSat(Δ^2) is tractable if Δ is 2-independent of $\emptyset$; i.e. an instance I of CspSat(Δ^2) has a solution if every $I' \subseteq I$ such that $|I'| = 2$ has a solution. We begin by generalising this result.

Lemma 13 If Δ is 2-independent of Γ, then CspSat($\Gamma \cup \Delta^2$) is tractable.

Proof: We show that the algorithm 2Ind-Solvable defined in Figure 2 succeeds when applied to C if and only if C has a solution.

only-if: Assume that 2IND-SOLVABLE returns **succeed**. This implies that there exists a satisfying truth assignment, μ, for $A \cup A' \cup A''$. Define the set of constraints C' as follows:

$$C' = \{c \mid \mu(q_c) = \textsf{true}\}.$$

We first show that C' has a solution. If C' has no solution, there exists $c_1, \ldots, c_k \in C'$ such that $\mathrm{Rel}(c_1), \ldots, \mathrm{Rel}(c_k) \in \Delta$ and $Q_\Gamma \cup \{c_1, \ldots, c_k\}$ is not satisfiable. We know that Q_Γ has a solution since the algorithm did not fail in line 4. Hence, the fact that μ satisfies A and A'' implies that $Q_\Gamma \cup \{c_i, c_j\}$ is satisfiable for $1 \leq i, j \leq k$ so $Q_\Gamma \cup \{c_1, \ldots, c_k\}$ is satisfiable since Δ is 2-independent of Γ. So C' does indeed have a solution.

Now, let f be a model of C'. For each disjunctive constraint in C we know that at least one of its disjuncts is a member of C', because μ satisfies the formulae in A'. We also know that every non-disjunctive constraint is a member of C' since μ satisfies the formulae in A''. Taken together, this means that C has a model.

if: Assume that C has a model f. Define the truth assignment $\mu : \{q_c \mid c \in P \cup Q_\Gamma \cup Q_\Delta\} \to \{\textsf{true}, \textsf{false}\}$ as follows:

$$\mu(q_c) = \textsf{true} \text{ iff } c \text{ is satisfied by } f.$$

We show that μ is a satisfying truth assignment of $A \cup A' \cup A''$ by considering the elements of A, A' and A'' in turn.

(1) For each formula $(\neg q_{c'} \vee \neg q_{c''}) \in A$, we know that $Q_\Gamma \cup \{c', c''\}$ has no model. Hence, it cannot be the case that $\mu(c') = \mu(c'') = \textsf{true}$, which means that $(\neg q_{c'} \vee \neg q_{c''})$ is satisfied by μ.

(2) For each formula $(q_{c'} \vee q_{c''}) \in A'$ we know that there is a constraint $c \in C$ of the form $c = c' \vee c''$. Since f is a model of C, f satisfies at least one of c' and c'' which means that $(q_{c'} \vee q_{c''})$ is satisfied by μ.

(3) For each formula $q_c \in A''$ there exists a constraint $c \in C$ that is not a disjunction. Consequently, f must satisfy c and μ satisfies q_c.

Finally, we have to show that the algorithm 2IND-SOLVABLE runs in polynomial time. This follows directly from the observation that line 5 can be computed in polynomial time (since CSPSAT($\Gamma \cup \Delta$) is tractable) and that the test in line 8 can be performed in polynomial time by using some tractable algorithm for showing the satisfiability of 2CNF formulae (such as the algorithm by Aspvall *et al.* [1]). □

Proof: (of Theorem 12) The if direction follows from Lemma 13. To show the other direction we assume to the contrary that CSPSAT($\Gamma \cup \Delta^2$) is tractable but Δ is not 2-independent of Γ.

This implies that there exists a set of constraints X over Γ and a set $H = \{h_1, \ldots, h_n\}$ over Δ such that $X \cup \{h_i, h_j\}$ is satisfiable for every $1 \leq i, j \leq n$ but $X \cup H$ is not satisfiable. Choose X and H such that $|H|$ is as small as possible and note that $|H| \geq 3$. The existence of a set $H' \subset H$ such that $X \cup H'$ is not satisfiable contradicts the minimality of H so $X \cup H'$ is satisfiable for all $H' \subset H$. Thus, we can define the satisfiable set $\mathcal{X} = X \cup \{h_1 \vee h_2, h_1 \vee h_3, h_2 \vee h_3, h_4, \ldots, h_n\}$ which have the following property: In every model of $\mathcal{X}$, exactly one of h_1, h_2, h_3 is not satisfied.

To prove the result, we show that 3-Colourability can be transformed to CspSat($\Gamma \cup \Delta^2$) in polynomial time. Arbitrarily choose an undirected graph $G = (V, E)$ such that $V = \{v_1, \ldots, v_k\}$. We will construct an instance of CspSat($\Gamma \cup \Delta^2$) that is satisfiable iff G is colourable with three colours.

For each vertex v_i, introduce a fresh copy of the set $\mathcal{X}$ where we denote the constraints h_1, h_2, h_3 as h_1^i, h_2^i, h_3^i, respectively. As we have already noted, this will force exactly one of h_1^i, h_2^i, h_3^i not to hold in every model. We interpret that h_j^i does not hold as 'vertex v_i has colour j'.

For each edge $(v_i, v_j) \in E$, we add the disjunctions $h_1^i \vee h_1^j$, $h_2^i \vee h_2^j$ and $h_3^i \vee h_3^j$ which ensures that v_i and v_j are not assigned the same colour. The resulting set of constraints can be computed in polynomial time, it is an instance of CspSat($\Gamma \cup \Delta^2$) and is satisfiable iff G is 3-colourable which concludes the proof. $\square$

Theorem 14 There exist sets of unary relations Γ, Δ such that CspSat($\Gamma \cup \Delta$) is tractable, Δ is 2-independent of Γ but CspSat($\Gamma \overset{\times}{\vee} \Delta$) is NP-complete.

Proof: Consider the domain $D = \{0, 1, 2\}$. Define unary relations $\mathsf{neq}_i \subseteq D$, $0 \leq i \leq 2$, such that $\mathsf{neq}_i(x)$ holds iff $i \neq x$ and define $\mathsf{eq}_i \subseteq D$, $0 \leq i \leq 1$, such that $\mathsf{eq}_i(x)$ holds iff $i = x$. Let $\Gamma = \{\mathsf{neq}_0, \mathsf{neq}_1, \mathsf{neq}_2\}$ and $\Delta = \{\mathsf{eq}_0, \mathsf{eq}_1\}$. Proving the tractability of CspSat($\Gamma \cup \Delta$) and that Δ is 2-independent of Γ are routine verifications.

We show that CspSat($\Gamma \overset{\times}{\vee} \Delta$) is NP-complete by a polynomial-time reduction from 3-Colourability. Let $G = (V, E)$ be an arbitrary undirected graph. We will construct an instance X of CspSat($\Gamma \overset{\times}{\vee} \Delta$) that is satisfiable iff G can be 3-coloured.

Assume $V = \{v_1, \ldots, v_m\}$. To simplify our description of the reduction, we will only consider edges $e = (v_i, v_j)$ in E such that $i < j$. Obviously, we can do this without loss of generality since G is undirected. For each vertex $v \in V$, introduce a variable $\hat{v}$ and for each edge $(v, w) \in E$, introduce three variables e_{vw}^i, $0 \leq i \leq 2$. Finally, for each edge $(v, w) \in E$, add the following six constraints to X:

(1) $\mathsf{neq}_0(\hat{v}) \vee \mathsf{eq}_0(e_{vw}^0)$ (2) $\mathsf{neq}_0(\hat{w}) \vee \mathsf{eq}_1(e_{vw}^0)$
(3) $\mathsf{neq}_1(\hat{v}) \vee \mathsf{eq}_0(e_{vw}^1)$ (4) $\mathsf{neq}_1(\hat{w}) \vee \mathsf{eq}_1(e_{vw}^1)$
(5) $\mathsf{neq}_2(\hat{v}) \vee \mathsf{eq}_0(e_{vw}^2)$ (6) $\mathsf{neq}_2(\hat{w}) \vee \mathsf{eq}_1(e_{vw}^2)$

Algorithm: 2IND-SOLVABLE
Input: A finite set C of constraints over $\Gamma \cup \Delta^2$.
Output: succeed if C is satisfiable; fail otherwise.

1. $P \leftarrow \{c_1, c_2 \mid c \in C \text{ and } c = c_1 \vee c_2\}$
2. $Q_\Delta \leftarrow \{c \in C \mid c \text{ is not a disjunction and } \mathrm{Rel}(c) \in \Delta\}$
3. $Q_\Gamma \leftarrow \{c \in C \mid c \text{ is not a disjunction and } \mathrm{Rel}(c) \in \Gamma \setminus \Delta\}$
4. if Q_Γ has no solution then return fail
5. define a set of boolean variables $\{q_c \mid c \in P \cup Q_\Gamma \cup Q_\Delta\}$
6. $A \leftarrow \{(\neg q_{c'} \vee \neg q_{c''}) \mid c', c'' \in P \cup Q_\Delta \text{ and } Q_\Gamma \cup \{c', c''\} \text{ not satisfiable}\}$
7. $A' \leftarrow \{(q_{c'} \vee q_{c''}) \mid \exists c \in C \text{ such that } c = c' \vee c''\}$
8. $A'' \leftarrow \{q_c \mid c \in Q_\Gamma \cup Q_\Delta\}$
9. if $A \cup A' \cup A''$ is satisfiable
   then return succeed
   else return fail

Figure 2: Algorithm 2IND-SOLVABLE

The value of variables $\hat{v}$ will equal the colour of the corresponding vertex and variable e^i_{vw} is to be interpreted as follows: if $e^i_{vw} = 0$, then variable $\hat{w}$ does not have the value i; otherwise, $\hat{w}$ equals i. Now, consider constraint (1). It tells us that either $\hat{v}$ is not equal to 0 *or* the variable $\hat{w}$ is not equal to 0. Hence, the constraints (1), (3) and (5) ensure that adjacent vertices are not assigned the same colour. For this to work, it must also be true that a variable $\hat{w}$ cannot have a value i and at the same time $e^i_{vw} = 0$. This is guaranteed by constraints (2), (4) and (6).

We can now show that X is satisfiable iff G is 3-colourable.

only-if: Let M be a model of X. We show that $M(\hat{v}) \neq M(\hat{w})$ whenever there is an edge between v and w in G. Since the range of M is $\{0, 1, 2\}$, M can easily be modified into a three-colouring of G.

Assume to the contrary that X has a model M such that $M(\hat{v}) = M(\hat{w}) = 0$ (the other two cases are analogous) and $(v, w) \in E$. Constraints (1) and (2) implies that both $\mathsf{eq}_0(e^0_{vw})$ and $\mathsf{eq}_1(e^0_{vw})$ hold which leads to a contradiction.

if: Let $f : V \rightarrow \{0, 1, 2\}$ be a 3-colouring of G. Construct a model M of X as follows:

$M(\hat{v}) = f(v)$;

$M(e^i_{vw}) = 0$ if $f(w) \neq i$

$M(e^i_{vw}) = 1$ if $f(w) = i$

To see that M is a model of X, arbitrarily choose a constraint c in X. Assume first that c is on the form (1) $\mathsf{neq}_0(\hat{v}) \vee \mathsf{eq}_0(e^0_{vw})$. This constraint is not satisfied iff $M(\hat{v}) = 0$ and $M(e^0_{vw}) = 1$. By the construction of M, it follows that $f(v) = 0$ and $f(w) = 0$ which contradicts the fact that f is a 3-colouring of G.

Assume c is on the form (2) $\mathsf{neq}_0(\hat{w}) \vee \mathsf{eq}_1(e^0_{vw})$ instead. This constraint is not satisfied iff $M(\hat{w}) = 0$ and $M(e^0_{vw}) = 0$. By the construction of M, it follows that $f(w) = 0$ and $f(w) \neq 0$ at the same time. □

4 1-Independence and Refinements

In the previous section we have shown that the 1-independence property is a necessary and sufficient condition for tractability of a natural class of disjunctive constraints. However, it is often quite difficult to prove that this property holds for a certain class, and this has to be proven for each class anew. Recently, Renz [16] proposed a general method for proving tractability of classes of relations which is comprised by running a simple algorithm. This refinement method, which is described in Section 2.3, seems to be related to the 1-independence property in the following (simplified) way:

The 1-independence property specifies when a constraint can be added to a set of constraints without changing consistency, while by the refinement method it can be shown if a relation can be removed from a disjunctive constraint without changing consistency. Actually, removing a relation R from a disjunctive constraint xSy is the same as adding the constraint $x \neg R y$. In Subsection 4.1, we try to elaborate this similarity and show under which conditions the 1-independence property corresponds to the refinement method and vice versa. Some successful examples for using the refinement method for proving 1-independence property are presented in Subsection 4.2. We stress once again that the results in this section are only applicable when considering binary relations.

4.1 Connections between 1-Independence and Refinements

We will now show how the refinement method can be used for proving 1-independence. Let $\mathcal{A}$ be a set of basic relations and choose $\mathcal{S} \subseteq 2^{\mathcal{A}}$ such that $\mathcal{S}$ can be decided by path-consistency. Let Δ be a subset of $\mathcal{S}$. We make the following additional assumptions about $\mathcal{S}$ and Δ:

1. $\mathsf{eq} \in \mathcal{S}$;

2. Δ is closed under intersection.

These restrictions can be imposed without loss of generality: First note that since $\textsc{CspSat}(\mathcal{S})$ is tractable, the problem $\textsc{CspSat}(\mathcal{S} \cup \{\mathsf{eq}\})$ is also tractable and can trivially be reduced to the first problem (by contracting any two variables related by eq to a single variable). The fact that Δ can be assumed to be closed under intersection follows from the next lemma.

Lemma 15 Let Γ, Δ be sets of relations such that Δ is 1-independent of Γ, then the closure of Δ under intersection is also 1-independent of Γ.

Proof: Let Θ be a set of constraints over Γ and $H = \{h_1, \ldots, h_n\}$ a set of constraints over $\Delta \cup \{R \cap S\}$ for some $R, S \in \Delta$. Assume that $\Theta \cup \{h_i\}$, $1 \leq i \leq n$ is satisfiable. Construct the set

$$H' = (H - \{(R \cap S)(x) \in H\}) \cup \{R(x), S(x) \mid (R \cap S)(x) \in H\}$$

and note that $\Theta \cup \{h'\}$ is satisfiable for all $h' \in H'$. The constraints in H' are all based on the relations in Δ so $\Theta \cup H'$ is satisfiable by 1-independence. It follows from the construction of H' that $\Theta \cup H$ is also satisfiable and $\Delta \cup \{R \cap S\}$ is 1-independent of Γ. $\Box$

From now on, we assume that all relations encountered are members of $\mathcal{S}$. We need a couple of lemmata before we can establish the main result.

Lemma 16 A triple (R, S, T) is satisfiable iff $R \cap (S \circ T) \neq \emptyset$.

Proof: The only-if direction is obvious. We show the other direction by choosing some basic relation $K \in R \cap (S \circ T)$ and arbitrarily picking two values a and c such that aKc. The fact that $K \in S \circ T$ implies that for all possible choices of a and c, there exists a value b such that aSb and bTc. By making the assignments $x = a$, $y = b$ and $z = c$, we have shown that (R, S, T) is satisfiable. $\Box$

Lemma 17 Assume that $\mathcal{S}$ can be refined by M^{Δ}, let R be a relation in $\mathcal{S}$ and $K_1, \ldots, K_n \in \Delta$. If $R \cap K_i \neq \emptyset$, $1 \leq i \leq n$, then $R \cap \bigcap_{i=1}^{n} K_i \neq \emptyset$.

Proof: Induction over n. The lemma obviously holds for $n = 1$ so we assume that it holds for $n = k$, $k \geq 1$. We show that the claim holds for $n = k + 1$. The induction hypothesis tells us that $R \cap \bigcap_{i=1}^{k} K_i \neq \emptyset$ and we know that $\bigcap_{i=1}^{k} K_i \in \Delta$ since Δ is closed under intersection. Consider the triple (R, R, eq) and note that it is path-consistent since $R = R \cap (R \circ \mathsf{eq})$ and $\mathsf{eq} = \mathsf{eq} \cap (R \circ R^{-1})$.

Since $R \cap K_{k+1} \neq \emptyset$, the fact that $\mathcal{S}$ can be refined by M^{Δ} implies that $(R \cap K_{k+1}, R \cap \bigcap_{i=1}^{k} K_i, \mathsf{eq})$ is satisfiable. By Lemma 16, this is equivalent with $(R \cap K_{k+1}) \cap ((R \cap \bigcap_{i=1}^{k} K_i) \circ \mathsf{eq}) \neq \emptyset$ so $(R \cap K_{k+1}) \cap (R \cap \bigcap_{i=1}^{k} K_i) = (R \cap \bigcap_{i=1}^{k+1} K_i) \neq \emptyset$ which concludes the induction. $\Box$

Theorem 18 If $\mathcal{S}$ can be refined by M^{Δ}, then Δ is 1-independent of $\mathcal{S}$.

Proof: Let Θ be a set of constraints over $\mathcal{S}$ and $H = \{h_1, \ldots, h_n\}$ a set of constraints over Δ. Let Θ' be the set Θ after enforcing path-consistency and assume that $\Theta \cup \{h_i\}$, $1 \leq i \leq n$, has a solution.

Arbitrarily choose i and assume that $h_i = xR'y$. Since $\Theta \cup \{h_i\}$ has a solution, $\Theta' \cup \{h_i\}$ also has a solution and there is a non-empty relation R that relates x and y in Θ'. Note that adding h_i to Θ' is the same thing as refining the relation $xRy \in \Theta'$ to $x(R \cap R')y$. Certainly, $R \cap R' \neq \emptyset$ since $\Theta' \cup \{h_i\}$ would not have a solution otherwise. Consequently, $M^{\Delta}[R][R \cap R'] = \mathsf{true}$ since $R' \in \Delta$ and by Lemma 17, we know that the relation R cannot be refined to the empty relation by adding more constraints from H. Thus, adding the constraints in H to Θ are all refinements according to M^{Δ}.

Since $\mathsf{Check\text{-}Refinements}(\mathcal{S}, M^{\Delta})$ succeeds, Theorem 8 tells us that such refinements can be made without making Θ' inconsistent, i.e. $\Theta' \cup H$ has a solution which trivially implies that $\Theta \cup H$ has a solution. We have thus shown that Δ is 1-independent of $\mathcal{S}$ since Θ and H were arbitrarily chosen. □

This theorem gives us the possibility to prove 1-independence of Δ with respect to $\mathcal{S}$ automatically by simply running $\textsc{Check-Refinements}(\mathcal{S}, M^{\Delta})$. If the algorithm returns `succeed`, we know that Δ is independent of $\mathcal{S}$. In order to make use of a negative answer of the algorithm, we also have to prove the opposite direction, i.e. independence of Δ with respect to a set $\mathcal{S}$ implies that $\textsc{Check-Refinements}(\mathcal{S}, M^{\Delta})$ returns `succeed`. Although this is a highly desirable property, we have not been able to prove this nor did we find a counterexample. There are, however, many examples for which this conjecture holds. As we will see in Subsection 4.2, this includes all 1-independence results for the point algebras for partially and totally ordered time. We give a proof of a slightly limited version of this conjecture.

Definition 19 Let $\mathcal{S} \subseteq 2^{\mathcal{A}}$ and $R \in \mathcal{S}$. We say that *path-consistency makes R explicit* iff for every path-consistent instance Θ of $\textsc{CspSat}(\mathcal{S})$, the following holds: if $M(x)RM(y)$ for every $M \in \text{Mods}(\Theta)$, then $xSy \in \Theta$ and $S \subseteq R$.

Theorem 20 Let $\mathcal{S} \subseteq 2^{\mathcal{A}}$ and assume that Δ is independent of $\mathcal{S}$. Then, $\textsc{Check-Refinements}(\mathcal{S}, M^{\Delta})$ returns `succeed` if and only if path-consistency makes $\neg R$ explicit for every $R \in \Delta$.

Proof: *only-if:* Assume to the contrary that there exists a path-consistent instance Θ of $\textsc{CspSat}(\mathcal{S})$, $x, y \in \text{Vars}(\Theta)$ and relations $R \in \Delta$, $S \in \mathcal{S}$ such that:

1. $xSy \in \Theta$;
2. for all $M \in \text{Mods}(\Theta)$, $M(x)\neg RM(y)$; and

3. $S \cap R \neq \emptyset$.

Since $R \in \Delta$ and Check-Refinements$(\mathcal{S}, M^{\Delta})$ returns `succeed`, the instance

$$\Theta' = \Theta \cup \{uRv \mid uTv \in \Theta \text{ and } T \cap R \neq \emptyset\}$$

has a solution. However, $S \cap R \neq \emptyset$ so $xRy \in \Theta'$. We know that all models M of Θ have the property $M(x) \neg R M(y)$ so every model M' of Θ' must also have this property. This contradicts the fact that Θ' has a model and, consequently, $S \cap R = \emptyset$ and $S \subseteq \neg R$. We have thus shown that path-consistency makes $\neg R$ explicit.

if: Let Θ be a path-consistent instance of CspSat$(\mathcal{S})$ and arbitrarily choose a constraint $xSy \in \Theta$ such that $S \cap R \neq \emptyset$ for some $R \in \Delta$. The fact that path-consistency makes $\neg R$ explicit gives that $\Theta \cup \{xRy\}$ has a solution and, by independence, $\Theta' = \Theta \cup \{uRv \mid R \in \Delta,\ uTv \in \Theta \text{ and } T \cap R \neq \emptyset\}$ has a solution. However, Θ' is equivalent to Θ refined by the matrix M^{Δ} so Check-Refinements$(\mathcal{S}, M^{R})$ returns `succeed` by Theorem 8 □

Corollary 21 Given a set of relations $\mathcal{S} \subseteq 2^{\mathcal{A}}$ for which path-consistency computes minimal labels and a refinement matrix M^{Δ}, Check-Refinements$(\mathcal{S}, M^{\mathcal{R}})$ returns `succeed` if and only if Δ is independent of $\mathcal{S}$.

Proof: If path-consistency computes minimal labels, then it makes $\neg R$ explicit for every $R \in \Delta$. □

Examples of when path-consistency computes minimal labels can, for instance, be found in Deville *et al.* [9] and Bessière *et al.* [3].

4.2 Computational experience

We will demonstrate that many 1-independence results can be obtained by using the refinement method. We shall show that all independence results for the point algebras for partially and totally ordered time can be derived using refinements. This is possible since we know *every* maximal tractable set of disjunctions of relations for partially ordered time [5]. This, of course, requires a definition of a maximal tractable set of disjunctions of relations. Let B be a set of basic relations and $\Gamma \subseteq B^*$ such that CspSat(Γ) is tractable. We say that Γ is *maximal tractable* (with respect to B) iff for every $R \in B^*$ such that $R \notin \Gamma$, $\Gamma \cup \{R\}$ is not tractable.

The point algebra for partially ordered time is based on the notion of *relations* between pairs of variables interpreted over a partially-ordered set. We consider four basic relations which we denote by $<, >, =$ and $\|$. If x, y are points in a partial order $\langle T, \leq \rangle$ then we define these relations in terms of the partial ordering $\leq$ as follows:

	Γ_A	Δ_A	Γ_B	Δ_B	Γ_C	Δ_C	Δ_D
$<$	$\bullet$		$\bullet$		$\bullet$		
$\leq$	$\bullet$		$\bullet$				$\bullet$
$<>$			$\bullet$	$\bullet$			
$<=>$			$\bullet$	$\bullet$			$\bullet$
$\|$	$\bullet$	$\bullet$			$\bullet$	$\bullet$	
$\| =$	$\bullet$				$\bullet$	$\bullet$	$\bullet$
$=$	$\bullet$		$\bullet$		$\bullet$		$\bullet$
$\neq$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	
$< \|$	$\bullet$	$\bullet$			$\bullet$	$\bullet$	
$\leq \|$	$\bullet$				$\bullet$	$\bullet$	$\bullet$

Table 1: Tractable classes of the point algebra for partially ordered time [5].

1. $x < y$ iff $x \leq y$ and not $y \leq x$
2. $x > y$ iff $y \leq x$ and not $x \leq y$
3. $x = y$ iff $x \leq y$ and $y \leq x$
4. $x \| y$ iff neither $x \leq y$ nor $y \leq x$

The point algebra for partially ordered time has been throughly investigated earlier and a total classification with respect to tractability has been given in Broxvall and Jonsson [4]. In Broxvall and Jonsson [5] the sets of relations in Table 1 are defined and it is proven that $\Gamma_A \overset{\times}{\vee} \Delta_A^*$, $\Gamma_B \overset{\times}{\vee} \Delta_B^*$, $\Gamma_C \overset{\times}{\vee} \Delta_C^*$ and Δ_D^* are the unique maximal tractable disjunctive classes of relations for partially ordered time. The proofs of tractability for those sets relied on a series of handmade independence proofs. We will now derive these independence results using the refinement method.

To do so, we need to show that the classes $\Gamma_A, \Gamma_B, \Gamma_C$ and Δ_D are decidable by path-consistency. We begin by proving a useful connection (Lemma 22) between RCC-5 and the point algebra for partially ordered time which in turn will be needed to prove that path-consistency decides Γ_A and Γ_B.

RCC-5 [2] is based on the notions of regions and binary relations on them. A region p is a regular open set of a topological space. Regions themselves do not have to be internally connected, i.e. a region may consist of different disconnected pieces.

Given two regions, their relation can be described by exactly one of the elements of the set $\mathbf{B}$ of five *basic RCC-5 relations.* The definition of these relations can be found in Table 2.

$X\{\mathtt{DR}\}Y$	iff	$X \cap Y = \emptyset$
$X\{\mathtt{PO}\}Y$	iff	$\exists a, b, c : a \in X, a \notin Y, b \in X, b \in Y, c \notin X, c \in Y$
$X\{\mathtt{PP}\}Y$	iff	$X \subset Y$
$X\{\mathtt{PPI}\}Y$	iff	$X \supset Y$
$X\{\mathtt{EQ}\}Y$	iff	$X = Y$

Table 2: The five basic relations of RCC-5.

Lemma 22 Let Γ be a set of relations in the point algebra for partially ordered time and define the function σ such that

1. $\sigma(<) = \mathsf{PP}$;
2. $\sigma(>) = \mathsf{PP}^{-1}$;
3. $\sigma(=) = \mathsf{EQ}$; and
4. $\sigma(||) = (\mathsf{DR\ PO})$.

Then, Γ can be decided by path-consistency if the set

$$\Gamma' = \{\bigcup_{r \in R} \sigma(r) \mid R \in \Gamma\}$$

of RCC-5 relations can be decided by path-consistency.

Proof: Let Π be an arbitrary CSP instance over the relations in Γ. Define the set Σ of RCC-5 formulae as follows: for each $x_i\ R\ x_j \in \Pi$, add the formula $x_i \bigcup_{r \in R} \sigma(r)\ x_j$. Note that Σ is a CSP instance over Γ' that, by assumption, can be decided by path-consistency.

We begin by comparing the composition tables for partially-ordered time and the RCC-5 relations (PP), (PP^{-1}), (EQ) and $(\mathsf{DR\ PO})$:

	$<$	$>$	$=$	$\|\|$
$<$	$\{<\}$	$\top$	$\{<\}$	$\{\|\| <\}$
$>$	$\top$	$\{>\}$	$\{>\}$	$\{> \|\|\}$
$=$	$\{<\}$	$\{>\}$	$\{=\}$	$\{\|\|\}$
$\|\|$	$\{\|\| <\}$	$\{> \|\|\}$	$\{\|\|\}$	$\top$

	(PP)	(PP^{-1})	(EQ)	$(\mathsf{DR\ PO})$
(PP)	(PP)	$\top$	(PP)	$(\mathsf{PP\ DR\ PO})$
(PP^{-1})	$\top$	(PP^{-1})	(PP^{-1})	$(\mathsf{PP}^{-1}\ \mathsf{DR\ PO})$
(EQ)	(PP)	(PP^{-1})	(EQ)	$(\mathsf{DR\ PO})$
$(\mathsf{DR\ PO})$	$(\mathsf{PP\ DR\ PO})$	$(\mathsf{PP}^{-1}\ \mathsf{DR\ PO})$	$(\mathsf{DR\ PO})$	$\top$

By also noting that $<^{-1} = >$, $>^{-1} = <$, $\mathsf{PP}^{-1} = \mathsf{PPI}$, $\mathsf{PPI}^{-1} = \mathsf{PP}$ and that all other relations are invariant under $\cdot^{-1}$, it is obvious that the empty relation can be derived from Π if and only if it can be derived from Σ. Thus, we only have to show that whenever Σ has a model, Π also has a model.

Let M be a model that assigns regions to the variables $x_1, \ldots, x_n$ that appear in Σ. We define an interpretation N from the variables in Π to the partial order $\langle \{M(x_i) \mid 1 \leq i \leq n\}, \subseteq \rangle$ as follows: $N(x_i) = M(x_i)$ for $1 \leq i \leq n$. To conclude the proof, we pick an arbitrary constraint $x_i R x_j$ in Σ and show that it is satisfied by the interpretation N. Assume now, for instance, that $M(x_i)$ (PP) $M(x_j)$. By the definition of σ, we know that $\{<\} \subseteq R$ and it follows immediately that $N(x_i) < N(x_j)$ and the constraint $x_i R x_j$ is satisfied. The remaining cases can be proved analogously. $\Box$

Theorem 23 Path-consistency decides consistency for Γ_A, Γ_B, Γ_C and Δ_D.

Proof: Let $\Gamma' = \{\bigcup_{r \in R} \sigma(r) \mid R \in \Gamma_A\}$ (where σ is defined as in Lemma 22) and note that $\Gamma' \subseteq R_5^{28}$ [13]. Since R_5^{28} can be decided by path-consistency [18], Lemma 22 implies that path-consistency decides Γ_A.

Similarly, we can verify that path-consistency decides Γ_B; in this case, $\Gamma'' = \{\bigcup_{r \in R} \sigma(r) \mid R \in \Gamma_B\} \subseteq R_5^{14}$ [13]. Showing that R_5^{14} is decided by path-consistency is straightforward and left as an exercise (hint: compare R_5^{14} and the point algebra for totally ordered time and recall that the latter is decided by path consistency).

For Γ_C the result follows from the fact that it is a subset of Γ_A and Δ_D is trivially decided by path-consistency. $\Box$

By using the algorithm Check-Refinements, we can automatically verify that $\Delta_A, \Delta_B, \Delta_C$ and Δ_D are valid refinements of $\Gamma_A, \Gamma_B, \Gamma_C$ and Δ_D, respectively. Theorem 23 now gives that $\Delta_A, \Delta_B, \Delta_C$ and Δ_D are independent of $\Gamma_A, \Gamma_B, \Gamma_C$ and Δ_D, respectively, so we have proven tractability of all maximal tractable sets of disjunctions of relations for the point algebra for partially ordered time.

In Broxvall and Jonsson [5] the point algebra for totally ordered time is also investigated and the following two classes are defined:

$$\mathcal{X}_1 = \{(<), (<=), (<>), (=)\} \overset{\times}{\vee} \{(<>)\}$$

$$\mathcal{X}_2 = \{(<=), (=)\}^*$$

These two classes are the only two maximal tractable sets of disjunctions of relations. It is well-known that path-consistency decides the point algebra for totally ordered time and the independence result can easily be verified using the refinement algorithm.

5 Conclusions and Open Questions

We have studied the complexity of reasoning with disjunctive constraints. We have shown that three previously presented properties are necessary and sufficient for tractability of $\textsc{CspSat}(\Delta^*)$, $\textsc{CspSat}(\Gamma \overset{\times}{\vee} \Delta^*)$ and $\textsc{CspSat}(\Gamma \cup \Delta^2)$. There is at least one interesting case that is not covered by our results so we pose the following problem:

Open question 1. Assume $\textsc{CspSat}(\Gamma \cup \Delta)$ is tractable. What is a necessary and sufficient condition for tractability of $\textsc{CspSat}(\Gamma \overset{\times}{\vee} \Delta)$.

Ideas taken from Cohen *et al.* [7] can probably be used for answering this question if we restrict Γ and Δ to be relations over disjoint domains.

We have provided a method for automatically deciding the 1-independence property based on refinements. The only requirement for applying this method is the sufficiency of path-consistency for deciding consistency in the class of constraints under consideration. In many cases this can, however, also be shown by using refinement techniques. We have demonstrated that this method is complete in two cases (the point algebras for totally-ordered and partially-ordered time) but we have not been able to prove this in general. We ask the following:

Open question 2. Assume path-consistency decides $\textsc{CspSat}(\Gamma)$. Is it true that $\Delta \subseteq \Gamma$ is 1-independent of Γ if and only if $\textsc{Check-Refinements}(\Gamma, M^{\Delta})$ returns $\mathsf{succeed}$?

Even if it turns out that the answer to the previous question is 'yes', there is still room for improving the refinement method since (1) it is restricted to binary relations only; and (2) path-consistency must decide the underlying $\textsc{CspSat}$ problem.

Open question 3. Given arbitrary sets Γ,Δ of relations, is there an algorithm for deciding whether Δ is 1-independent of Γ or not?

The previous questions naturally suggest our final question.

Open question 4. Given arbitrary sets Γ,Δ of relations, is there an algorithm for deciding whether Δ is 2-independent of Γ or not?

Acknowledgements

We would like to thank Andrei Bulatov, David Cohen, Peter Jeavons, Andrei Krokhin and the anonymous reviewers for valuable comments. Mathias Broxvall has been supported by the ECSEL graduate student program. Peter Jonsson has been supported by the Swedish Research Council for the Engineering Sciences (TFR) under grant 97-301 and the Swedish Research Counncil (VR) under grant 221-2000-361. Jochen Renz has been partially supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the project FAST-QUAL-SPACE which is part of the DFG special research effort on 'Spatial Cognition', by the Wallenberg foundation and by a Marie Curie Fellowship of the European Community programme "Improving Human Potential" under contract number HPMF-CT-2000-00667.

References

[1] B. Aspvall, M. F. Plass and R. E. Tarjan, A linear time algorithm for testing the truth of certain quantified Boolean formulas, Information Processing Letters 8 (1979) 121–123.

[2] B. Bennett, Spatial reasoning with propositional logics, in: Proceedings of the 4th International Conference on Principles on Knowledge Representation and Reasoning (KR-94), Bonn, Germany, 1994, pp. 165–176.

[3] C. Bessière, A. Isli and G. Ligozat. Global consistency in interval algebra networks: tractable subclasses, in: W. Wahlster, editor, Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96), Budapest, Hungary, 1996, pp. 3–7.

[4] M. Broxvall and P. Jonsson, Towards a complete classification of tractability in point algebras for nonlinear time, in: Proceedings of the 5th International Conference on Principles and Practice of Constraint Programming (CP-99), Alexandria, VA, USA, 1999, pp. 448–454.

[5] M. Broxvall and P. Jonsson, Disjunctive temporal reasoning in partially ordered time structures, in: Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, USA, 2000, pp. 464–469.

[6] M. Broxvall, P. Jonsson and J. Renz, Refinements and independence: A simple method for identifying tractable disjunctive constraints, in: Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP-2000), Singapore, 2000, pp. 114–127.

[7] D. Cohen, P. Jeavons and R. Gault, New tractable classes from old, in: Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP-2000), Singapore, 2000, pp. 160–171.

[8] D. Cohen, P. Jeavons, P. Jonsson and M. Koubarakis, Building tractable disjunctive constraints, Journal of the ACM 47 (5) (2001) 826–853.

[9] Y. Deville, O. Barette and P. van Hentenryck, Constraint satisfaction over connected row convex constraints, Artificial Intelligence 109 (1–2) (1999) 243–271.

[10] E. C. Freuder. A sufficient condition for backtrack-bounded search. Journal of the ACM 32 (1985) 755–761.

[11] G. Gottlob, N. Leone and F. Scarcello, A comparison of structural CSP decomposition methods, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 1999, pp. 394–399.

[12] P. G. Jeavons and M. C. Cooper, Tractable constraints on ordered domains, Artificial Intelligence 79 (1996) 327–339.

[13] P. Jonsson and T. Drakengren, A complete classification of tractability in RCC-5, Journal of Artificial Intelligence Research 6 (1997) 211–221.

[14] A. K. Mackworth, Consistency in networks of relations, Artificial Intelligence 8 (1977) 99–118.

[15] B. Nebel and H.-J. Bürckert, Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra, Journal of the ACM 42 (1) (1995) 43–66.

[16] J. Renz, Maximal tractable fragments of the region connection calculus: A complete analysis, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 1999, pp. 448–454w.

[17] J. Renz, On the complexity of binary relations, Technical Report, Institut für Informationssysteme, Technische Universität Wien, forthcoming.

[18] J. Renz and B. Nebel, On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus, Artificial Intelligence 108 (1–2) (1999) 69–123.

[19] P. van Beek and R. Dechter, On the minimality and decomposability of row-convex constraint networks, Journal of the ACM 42 (3) (1995) 543–561.