Reasoning About Set Constraints Applied to Tractable Inference in Intuitionistic Logic

Thomas Drakengren, thodr@ida.liu.se and
Peter Jonsson, petej@ida.liu.se
Department of Computer and Information Science,
Linkping University
S-581 83 Linkping, Sweden

September 4, 1998

Abstract

Automated reasoning about sets has received a considerable amount of interest in the literature. Techniques for such reasoning have been used in, for instance, analyses of programming languages, terminological logics and spatial reasoning. In this paper, we identify a new class of set constraints where checking satisfiability is tractable (i.e. polynomial-time). We show how to use this tractability result for constructing a new tractable fragment of intuitionistic logic. Furthermore, we prove NP-completeness of several other cases of reasoning about sets.

1 Introduction

There has been considerable interest in formalisms for describing and reasoning about sets. We begin by describing some of these. The most well-studied class of set constraints is, probably, *Herbrand set constraints*. Such have been suggested as a formalism for describing relationships between sets of terms of a free algebra. A *positive* set constraint has the form $X \subseteq Y$, where X and Y are *set expressions*. Examples of set expressions are $\varnothing$ (the empty set), α (a set-valued variable), $c(X,Y)$ (a constructor application), and the union, intersection, or complement of set expressions. The computational problem then is to decide whether there exists an assignment that satisfies all the given set constraints. This is the classical definition of a set constraint as stated in, for example, the paper by Heintze and Jaffar [17]. However, certain researchers have also studied *negative* set constraints [26, 3], i.e. set constraints of the form $X \not\subseteq Y$.

There are a number of important applications for automated reasoning about Herbrand set constraints. These include analyses of programming languages [4, 17, 31], logic programming [2, 18] and the study of finite automata [8]. On the theoretical side, rapid progress has been made in understanding the algorithms for and complexity of solving various classes of set constraints. One of the most important results is that the satisfiability problem for set constraint is decidable even if we allow negative constraints [1, 14]. On the practical side, several systems have been implemented for reasoning based on solving systems of set constraints [4, 16].

A variant of set constraints are the *Tarskian set constraints* [15, 21, 22]. Syntactically, Tarskian set constraints are very similar to Herbrand set constraints. The difference is that Tarskian set constraints are interpreted relative to a first-order structure whereas Herbrand set constraints are always interpreted over the Herbrand universe generated by a specified set of constructor functions. This difference fundamentally changes the nature of the satisfiability problem.

Reasoning about sets has also received attention in the artificial intelligence community. One example is so-called "concept languages" or "terminological languages" [32, 7]. The set expressions of concept languages are constructed from set variables and relation variables using a variety of compositional mechanisms. For instance, the expression $\forall R.C$ where R is a relation expression and C is a set expression denotes the set

$$\{x \mid \forall y.R(x, y) \rightarrow y \in C\}.$$

Many of these languages can be viewed as highly restricted fragments of propositional dynamic logic (PDL) [10]. Thus, many of them have satisfiability problems in P, NP or PSPACE which is significantly better than full PDL which is EXPTIME-complete [9].

Reasoning about sets has also appeared in connection with other types of logics, such as modal logics. These logics involve formulae which are true or false of possible worlds in Kripke structures. It is natural to view a Kripke structure as an ordinary first order structure whose domain is the set of possible worlds. In this view each formula denotes a set of worlds, which is nothing more than a subset of the domain of the structure in which the formula is interpreted.

In this paper, we study a variation of reasoning about Tarskian set constraints. More specifically, we continue the study of the set relations by Jonsson and Drakengren [25], restricting the set expressions to only allow for set-valued variables, but allowing them to be related by relations other than $\subseteq$: variables are related with subset ($\subseteq$), disjointness (disj) and non-equality ($\neq$), and we allow disjunctions of such set relations. As a concrete example, consider the following three relations:

$$Z \subseteq X, Z \subseteq Y, X \mathbin{\mathsf{disj}} Y.$$

It is not hard to see that we can satisfy these relations by setting $X = Y = Z = \varnothing$. However, if we disallow the use of the empty set, we cannot satisfy the relations. The case investigated by Jonsson and Drakengren [25] was the case where the empty set is disallowed, whereas the emphasis of this paper is on the other case, allowing the empty set as well. We shall see that this choice has a strong impact on the complexity of the satisfiability problem in certain cases.

The results in this paper complement the results of Jonsson and Drakengren [25], and show the complexity of the satisfiability problem for five types of formulae, where the empty set is allowed in models of set contraints: general formulae, Horn formulae of two kinds, Krom formulae and atomic formulae. The results show that the problem is polynomial for atomic formulae and NP-complete for Krom and general formulae. For one kind of Horn formulae, the problem is polynomial if we disallow the empty set and NP-complete otherwise, but for the other kind both problems are polynomial.

Even though the problem that we are about to study appears to be severely restricted, it appears to be useful. To demonstrate this, we show how it can be used in *inference in intuitionistic logic*. We exhibit a subclass of intuitionistic propositional logic (see, for example, Heyting [19] or Dragalin [11] for an introduction), where computing entailment can be done in polynomial time. This is an improvement of the general case, since the full problem has been shown to be PSPACE-complete [33]. Furthermore, intuitionistic logic and spatial reasoning are intimately connected: it is used for that purpose by Bennett [6], and indeed, Nebel [27] found a tractable subclass of intuitionistic logic, which aided in finding a tractable subclass of spatial reasoning. In this paper, we show that our tractable class is incomparable to that of Nebel [27].

The structure of the paper follows: In Section 2 we introduce the formalism to be used, the *disjunctive set relations*, DSRs. Its computational properties are investigated for different restrictions on the DSRs in Section 3, Section 4 and Section 5. Finally, we apply these results to intuitionistic logic in Section 6, after which a discussion and conclusions conclude the paper.

2 Disjunctive Set Relations

Our approach to reasoning about sets is based on that of Jonsson and Drakengren [25], extended with concepts for handling also empty sets in models: There are *set variables* and *relations* on these. We assume that we have a fixed universe of variable names for sets. Then, an $S_\varnothing$*-interpretation* is a function that maps set variables to (possibly empty) sets whereas an S*-interpretation* is a function that maps set variables to nonempty sets. The case of S-interpretations has already been investigated by Jonsson and Drakengren [25].

We will relate set variables by the following three relations: subset ($\subseteq$), nonequality ($\neq$) and disjointness (disj) having the usual definitions. A formula of this type is called an *atomic relation.* A *disjunctive set relation* (DSR) is a set (disjunction) of atomic relations.

An atomic relation XRY is satisfied by an $S_\varnothing$-interpretation (S-interpretation) I iff $I(X)RI(Y)$. A DSR $\gamma = \{\gamma_1, \ldots, \gamma_n\}$, which is more conveniently written as a disjunction $\gamma = \gamma_1 \vee \ldots \vee \gamma_n$, is said to be $S_\varnothing$-satisfiable (S-satisfiable) iff there exists an $S_\varnothing$-interpretation (S-interpretation) that satisfies at least one member of each γ_i, $1 \leq i \leq n$. Such a satisfying $S_\varnothing$-interpretation (S-interpretation) is said to be an $S_\varnothing$-model (S-model) of γ. We also allow conjunctions of two or more DSRs $\alpha_1, \ldots, \alpha_n$ written as $\alpha_1 \wedge \ldots \wedge \alpha_n$, which are interpreted in the obvious way.

For DSRs, we have the following natural reasoning problem which we will denote DSRSAT$_\varnothing$ and DSRSAT, respectively.

INSTANCE: A finite set Γ of DSRs.
QUESTION: Does there exist an $S_\varnothing$-model (S-model) of Γ?

Relation	DSR-form
$X = \varnothing$	$X \,\mathsf{disj}\, X$
$X \neq \varnothing$	$E \,\mathsf{disj}\, E \wedge X \neq E$
$X = Y$	$X \subseteq Y \wedge Y \subseteq X$
$X \subset Y$	$X \subseteq Y \wedge X \neq Y$
$X \not\subseteq Y$	$F \neq \varnothing \wedge F \subseteq X \wedge F \,\mathsf{disj}\, Y$
$X \cap Y \neq \varnothing$	$F \neq \varnothing \wedge F \subseteq X \wedge F \subseteq Y$
$X \subseteq Y \cap Z$	$X \subseteq Y \wedge X \subseteq Z$
$Y \cup Z \subseteq X$	$Y \subseteq X \wedge Z \subseteq X$

Table 1: Some relations that can be expressed as DSRs. E and F denote fresh set variables.

We will see that DSRSAT$_\varnothing$ is NP-complete later on. DSRSAT was proved to be NP-complete by Jonsson and Drakengren [25]. To define tractable subclasses of these problems, we need to name some special classes of DSRs.

Definition 1 Let γ be a DSR. Then we let $\mathcal{S}(\gamma)$ denote the set of subset relations in γ, $NE(\gamma)$ the set of non-equality relations, $\mathcal{D}(\gamma)$ the set of disjointness relations, $NS(\gamma) = NE(\gamma) \cup \mathcal{D}(\gamma)$, and $SD(\gamma) = \mathcal{S}(\gamma) \cup \mathcal{D}(\gamma)$. We say that γ is a *subset relation* iff $|NS(\gamma)| = 0$, γ is a *non-subset relation* iff $|\mathcal{S}(\gamma)| = 0$, γ is an *SD-relation* iff $|NE(\gamma)| = 0$, and γ is a *non-SD-relation* iff $|SD(\gamma)| = 0$. If γ is a subset or a non-subset relation we say that γ is *homogeneous* and otherwise we say that γ is *heterogeneous*. If γ is an SD or a non-SD-relation we say that γ is *2-homogeneous* and otherwise we say that γ is *2-heterogeneous*. If $|\gamma| = 1$, then γ is *atomic*, if $|\gamma| \leq 2$, then γ is *Krom*, if $|\mathcal{S}(\gamma)| \leq 1$, then γ is *Horn*, and if $|SD(\gamma)| \leq 1$, then γ is *2-Horn*. Thus if γ is 2-Horn, then γ is also Horn. We extend these definitions to sets of relations in the obvious way. For example, if Γ is a set of DSRs and all $\gamma \in \Gamma$ are 2-Horn, then Γ is 2-Horn. □

Sometimes we blur the distinction between an atomic DSR γ and the atomic relation $\delta \in \gamma$, so that we can talk about the *DSR* δ.
Some examples of DSRs follow.

Example 2 Let

$$\gamma \equiv X \neq Y \vee Y \,\mathsf{disj}\, Z \vee W \subseteq X \vee X \subseteq Y.$$

Now

$$\mathcal{S}(\gamma) = W \subseteq X \vee X \subseteq Y,$$

$$NE(\gamma) = X \neq Y,$$

$$\mathcal{D}(\gamma) = Y \,\mathsf{disj}\, Z,$$

$$NS(\gamma) = X \neq Y \vee Y \,\mathsf{disj}\, Z,$$

and

$$SD(\gamma) = W \subseteq X \vee X \subseteq Y \vee Y \,\mathsf{disj}\, Z.$$

We see that γ is not Horn, and also not 2-Horn or Krom. The formula

$$X \subseteq Y \vee Y \subseteq Z$$

is a Krom DSR,

$$X \subseteq Y \vee W \neq Z \vee W \,\mathsf{disj}\, X$$

is a Horn DSR, but not a 2-Horn DSR, and

$$X \subseteq Y \vee W \neq Z \vee Z \neq V$$

and

$$X \,\mathsf{disj}\, Y \vee X \neq Z \vee Y \neq W$$

are 2-Horn DSRs. □

From the relations $\subseteq$, disj and $\neq$, several other set relations can be constructed. Table 1 contains a few examples.

3 The Atomic Case

The first reasoning problem that we will study is AtomDSRSat$_{\varnothing}$:

Instance: A finite set Γ of atomic DSRs.
Question: Does there exist an $S_{\varnothing}$-model of Γ?

We claim that Algorithm 3 correctly solves AtomDSRSat$_{\varnothing}$, and furthermore, does so in polynomial time. The corresponding algorithm for S-interpretations was proved correct and polynomial by Jonsson and Drakengren [25].

Algorithm 3 (Atom-SAT$_{\varnothing}(G)$)

input instance $G = \langle V, E \rangle$ of AtomDSRSat$_{\varnothing}$

```
Extend G with a new node ∅ and a new edge ∅ disj ∅
while there exists a null-defined node Z ∈ V different from ∅
    collapse the nodes Z and ∅ in G
if there exist inconsistent nodes X, Y ∈ V then
    reject
accept
```

□

To understand the algorithm Atom-SAT$_{\varnothing}$, a few things need to be explained. First, note that the input to the algorithm does not consist of a finite set of atomic DSRs, but of a directed, labelled graph (DLG). This is no loss of generality since any AtomDSRSat$_{\varnothing}$ instance can be represented by a DLG, where the nodes are set variables and the arcs are labelled by the relations $\subseteq$, $\neq$ and disj. From now on, let Γ be an arbitrary instance of AtomDSRSat$_{\varnothing}$ and $G = \langle V, E \rangle$ the DLG representing it.

We need the following auxiliary definitions.

Definition 4 Let $H = \langle W, F \rangle$ be a DLG, $x, y \in W$, and H' the graph obtained from H by removing all edges not labelled by $\subseteq$. Now, a directed path from x to y in H' is said to be a $\subseteq$*-path* in H. $\square$

Next, we define the concepts of *null-defined* and *inconsistent* nodes.

Definition 5 A node $Z \in V$ is *null-defined* iff there exist $X, Y \in V$ such that

- there exists an arc $e \in E$ from X to Y labelled disj in G
- there exists a $\subseteq$-path from Z to X in G
- there exists a $\subseteq$-path from Z to Y in G.

$\square$

Note that the case when we have $X \,\mathsf{disj}\, X$ is covered by this definition (we allow zero-length paths).

Definition 6 Two nodes $X, Y \in V$ are *inconsistent* iff

- there exists an arc $e \in E$ from X to Y labelled $\neq$
- there exists a $\subseteq$-path from X to Y in G
- there exists a $\subseteq$-path from Y to X in G.

$\square$

The following lemma is immediate and left without proof.

Lemma 7 If Z is a null-defined node in G, then Z must be assigned $\varnothing$ by any $S_\varnothing$-model of G, and if X and Y are inconsistent in G, then G has no $S_\varnothing$-model. $\square$

We can now show the completeness of Algorithm 3.

Lemma 8 Let Γ be a set of atomic DSRs. If Atom-SAT$_\varnothing(\Gamma)$ rejects, then Γ is not $S_\varnothing$-satisfiable.

Proof First note that the collapsing done in lines 2-3 preserves soundness, i.e. it does not remove any models, by Lemma 7.

Suppose that the algorithm rejects, and let G_* be obtained from G by performing the collapsing done in lines 2-3. Then there are two inconsistent nodes in G_*. Now, by Lemma 7, G_* cannot have an $S_\varnothing$-model. Since lines 2-3 preserve soundness, G cannot have an $S_\varnothing$-model.

Next, we prove the soundness of the algorithm.

Lemma 9 Let Γ be a set of atomic DSRs. If Atom-SAT$_\varnothing(\Gamma)$ accepts, then Γ is $S_\varnothing$-satisfiable.

Proof Suppose that the algorithm accepts. Let G_* be the graph obtained from G by performing lines 2-3 of Algorithm 3. Note that if G_* has an $S_\varnothing$-model, then G will have an $S_\varnothing$-model, by assigning the empty set to all nodes collapsed with the node $\varnothing$.

We show how to construct an $S_\varnothing$-model I of G_* over the set $\mathbb{N}$ of natural numbers.

Let G' be the graph obtained from G_* by removing arcs which are not labelled $\subseteq$. Assume $V = \{v_\varnothing, v_1, \ldots, v_n\}$ and define the function $I : V \to 2^{\mathbb{N}}$ such that

$$I(v_\varnothing) = \varnothing$$

and for $i \neq \varnothing$

$$I(v_i) = \{j \in \mathbb{N} \mid \text{there exists a path from } v_j \text{ to } v_i \text{ in } G'\}$$

for $1 \leq i \leq n$. We will show that I is a model of G_* by considering the three possible ways of relating two variables v_i and v_j in G_*, for $i, j \in \mathbb{N} \cup \{\varnothing\}$.

1. $v_i \subseteq v_j$. Then there exists a path from v_i to v_j in G' which implies $I(v_i) \subseteq I(v_j)$ by the construction of I.

2. $v_i \neq v_j$. First note that $i \neq j$, since otherwise the algorithm would have rejected.

   Suppose that one of v_i or v_j is $v_\varnothing$. Since $i \neq j$, and the only variable which is assigned $\varnothing$ in I is $v_\varnothing$, $I(v_i) \neq I(v_j)$ holds.

   Now suppose that $i, j \in \mathbb{N}$, and assume $i \in I(v_j)$. Then there is a path in G' from v_i to v_j, by the definition of I. But since v_i and v_j are not inconsistent, we must have $j \notin I(v_i)$. Similarly, if $j \in I(v_i)$, then $i \notin I(v_j)$. Thus $I(v_i) \neq I(v_j)$ holds.

3. $v_i \,\mathsf{disj}\, v_j$. If either of v_i or v_j is $v_\varnothing$, the relation holds trivially in I. Thus suppose $i, j \in \mathbb{N}$, and let $k \in I(v_i) \cap I(v_j)$. By the construction of I, there exists a path from v_k to v_i and from v_k to v_j in G'. Thus v_k is null-defined. But since all null-defined nodes have been collapsed with $\varnothing$ in lines 2-3, we must have $v_k = v_\varnothing$, contradicting our assumption. Thus there is no such k, and thus $I(v_i) \,\mathsf{disj}\, I(v_j)$.

Theorem 10 AtomDSRSat$_\varnothing$ is polynomial.

Proof Correctness of algorithm Atom-SAT follows from Lemma 8 and Lemma 9. The number of iterations in lines 2-3 is bounded from above by the number of nodes in the given instance, and the tests can easily be carried out in polynomial time. Thus, the theorem follows.

We conclude this section by showing that DSRSat$_\varnothing$ and DSRSat are members of NP, which will be needed later on.

Lemma 11 DSRSat$_\varnothing$ and DSRSat are in NP.

Proof Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a finite set of DSRs. If Γ is $S_\varnothing$-satisfiable, then for each i, $1 \leq i \leq n$, there exists an atomic set relation $\alpha_i \in \gamma_i$ such that $A = \{\alpha_1, \ldots, \alpha_n\}$ is $S_\varnothing$-satisfiable. Checking the $S_\varnothing$-satisfiability of A is polynomial by Theorem 10. Thus, given Γ we can let A be the non-deterministic guess which can be verified in polynomial time. As a consequence, DSRSat$_\varnothing$ is in NP.

The proof for DSRSat is analogous.

4 The Horn Case

The reasoning problems we shall study in this section extends the class of atomic DSRs, as was done for S-interpretations in Jonsson and Drakengren [25]. They are defined as follows:

HORNDSRSAT$_\varnothing$:
INSTANCE: A finite set Γ of Horn DSRs.
QUESTION: Does there exist an $S_\varnothing$-model of Γ?

HORNDSRSAT:
INSTANCE: A finite set Γ of Horn DSRs.
QUESTION: Does there exist an S-model of Γ?

2HORNDSRSAT:
INSTANCE: A finite set Γ of 2-Horn DSRs.
QUESTION: Does there exist an $S_\varnothing$-model of Γ?

2HORNDSRSAT$_\varnothing$:
INSTANCE: A finite set Γ of 2-Horn DSRs.
QUESTION: Does there exist an S-model of Γ?

The problem HORNDSRSAT was proved to be polynomial by Jonsson and Drakengren [25], and thus 2HORNDSRSAT is also polynomial.

Somewhat surprisingly, including the empty set makes the corresponding problem harder: We now show that HORNDSRSAT$_\varnothing$ is NP-complete.

Theorem 12 HORNDSRSAT$_\varnothing$ is NP-complete, even if we do not allow $\subseteq$ to appear in the formulae.

Proof Membership in NP follows from Lemma 11. Hardness for NP is shown by a polynomial-time reduction from 3SAT [13]. The problem has the following definition:

INSTANCE: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$.
QUESTION: Is C satisfiable?

Let $L = \langle U, C\rangle$ be an instance of 3SAT with $U = \{u_1, \ldots, u_m\}$ and $C = \{c_1, \ldots, c_n\}$. We will construct an instance of HORNDSRSAT$_\varnothing$ that is $S_\varnothing$-satisfiable iff L is satisfiable.

We begin by introducing the empty set by the following axiom: $E \,\mathsf{disj}\, E$. Clearly, $E = \varnothing$ in any model. For each u_i in U, we introduce a set variable x_i. We will interpret x_i as follows: u_i is false if $x_i = \varnothing$ and u_i is true if $x_i \neq \varnothing$. Consider the following set of Horn DSRs:

$$\Gamma = \bigcup_{c \in C} \{ \bigvee_{u_p \in c} x_p \neq E \vee \bigvee_{\neg u_p \in c} x_p \,\mathsf{disj}\, x_p \}.$$

Clearly, Γ has an $S_\varnothing$-model iff L is satisfiable. Also note that the relation $\subseteq$ is not used anywhere in Γ. Hence, the theorem follows.

As a simple consequence of this result and Lemma 11, DSRSAT$_\varnothing$ is also NP-complete.

Next, we show the tractability of 2HORNDSRSAT. The proof is very similar to how the tractability of HORNDSRSAT was proved [25]. The idea of constructing Horn classes in this way originated in the paper by Jonsson and Bäckström [23].

Definition 13 Let A be an $S_\varnothing$-satisfiable set of atomic DSRs and let γ be a DSR. We say that γ ***2-blocks*** A iff for every $d \in NE(\gamma)$, $A \cup \{d\}$ is not $S_\varnothing$-satisfiable. $\square$

Clearly, we can use the ATOMDSRSAT$_\varnothing$ algorithm to decide in polynomial time whether a DSR γ 2-blocks a set A of atomic DSRs. The next lemma relates satisfiability of a set of 2-Horn DSRs to the satisfiability of a set of atomic DSRs.

We claim that Algorithm 14 correctly solves 2HORNDSRSAT in polynomial time. The next three lemmata prove that the algorithm is sound.

Algorithm 14 (2-Horn-SAT(Γ))

input instance Γ of 2HORNDSRSAT

$A \leftarrow \bigcup\{\gamma \in \Gamma \mid |NE(\gamma)| = 0\}$
if not Atom-SAT$_\varnothing(A)$ **then**
    reject
if $\exists \gamma \in \Gamma$ that 2-blocks A and is a non-SD DSR **then**
    reject
if $\exists \gamma \in \Gamma$ that 2-blocks A and is 2-heterogeneous **then**
    2-Horn-SAT$((\Gamma - \{\gamma\}) \cup SD(\gamma)$
accept

$\square$

Lemma 15 Let Γ be a set of 2-Horn DSRs. Let $A \subseteq \Gamma$ be the set of *SD* DSRs (which are then atomic, since they are also 2-Horn) in Γ and let $D = \{D_1, \ldots, D_k\} \subseteq \Gamma$ be the set of DSRs that are not *SD* DSRs. If A is $S_\varnothing$-satisfiable and D_i does not 2-block A for any $1 \leq i \leq k$, then Γ is $S_\varnothing$-satisfiable.

Proof Pick one non-*SD* relation d_i out of every D_i such that $A \cup \{d_i\}$ is $S_\varnothing$-satisfiable. This is possible since no D_i 2-blocks A. We shall show that $\Gamma' = A \cup \{d_1, \ldots, d_k\}$ (which we note is a set of atomic relations) is $S_\varnothing$-satisfiable and, hence, Γ is $S_\varnothing$-satisfiable.

Suppose to the contrary that Γ' is not $S_\varnothing$-satisfiable, and let G be the equivalent graph representation of Γ'. Denote by G_* the graph obtained from G by performing lines 1-3 of Algorithm 3. By the correctness of Algorithm 3, we know that there exist nodes X, Y in G_* which are inconsistent, and thus are connected by an $\neq$ arc in G_*. Since lines 1-3 in the algorithm do not add any $\neq$-relations, there have to exist nodes X', Y' in G connected by $\neq$, which after the collapsing correspond to nodes X and Y

in G_*. This relation has to be a d_i for some i. Consider the graph G' obtained from G by removing all $\neq$-relations except for this d_i. Now run Algorithm 3 on G'. Since the collapsings performed in lines 1-3 do not depend on any of the relations removed from G, X and Y will still be inconsistent in the resulting graph G'_*, and thus, $A \cup \{d_i\}$ is unsatisfiable, contradicting the assumption. Thus Γ' is $S_\varnothing$-satisfiable, and hence the same holds for Γ.

Lemma 16 Let Γ be a set of 2-Horn DSRs and let $C \subseteq \Gamma$ be the set of *SD* DSRs in Γ. If there exists a 2-heterogeneous DSR $\gamma \in \Gamma$ such that γ 2-blocks C, then Γ is $S_\varnothing$-satisfiable iff $(\Gamma - \{\gamma\}) \cup SD(\gamma)$ is $S_\varnothing$-satisfiable.

Proof $\Leftarrow$) Trivial.
$\Rightarrow$) If Γ is $S_\varnothing$-satisfiable, then C has to be $S_\varnothing$-satisfiable. Since γ 2-blocks C, $SD(\gamma)$ must be $S_\varnothing$-satisfied in any solution of Γ.

Lemma 17 Let Γ be a set of 2-Horn DSRs. If 2-Horn-SAT(Γ) accepts, then Γ is $S_\varnothing$-satisfiable.

Proof Induction over n, the number of 2-heterogeneous DSRs in Γ.

Suppose $n = 0$, and that the algorithm accepts, which has to be in line 8. Then the formulae in A are $S_\varnothing$-satisfiable, and there does not exist any $\gamma \in \Gamma$ that 2-blocks A. Now, by Lemma 15, Γ is $S_\varnothing$-satisfiable.
Suppose Γ contains $n+1$ 2-heterogeneous DSRs, assuming that the result holds for n. First suppose that the algorithm accepts in line 7 (that is, within that call). Then $(\Gamma - \{\gamma\}) \cup SD(\gamma)$, containing n 2-heterogeneous DSRs, is $S_\varnothing$-satisfiable by the induction hypothesis. By Lemma 16, this is equivalent to Γ being $S_\varnothing$-satisfiable. Now instead suppose that the algorithm accepts in line 8. Then there does not exist any $\gamma \in \Gamma$ with $|NE(\gamma)| > 0$ which 2-blocks A. By Lemma 15, this means that Γ is $S_\varnothing$-satisfiable.

The completeness of 2-Horn-SAT is shown in the following two lemmata.

Lemma 18 Let Γ be a set of 2-Horn DSRs. Let $C \subseteq \Gamma$ be the set of *SD* DSRs in Γ. If there exists a non-*SD* DSR $\gamma \in \Gamma$ that 2-blocks C, then Γ is not $S_\varnothing$-satisfiable.

Proof In any solution to Γ, the relations in $C \cup \{\gamma\}$ have to be $S_\varnothing$-satisfied. Since γ is a non-*SD* DSR and 2-blocks C, this is not possible, and the lemma follows.

Lemma 19 Let Γ be a set of 2-Horn DSRs. If 2-Horn-SAT(Γ) rejects, then Γ is not $S_\varnothing$-satisfiable.

Proof Induction over n, the number of 2-heterogeneous DSRs in Γ.
If $n = 0$ then, 2-Horn-SAT can reject in lines 3 or 5. If 2-Horn-SAT rejects in line 3, then, trivially, Γ is not $S_\varnothing$-satisfiable. If 2-Horn-SAT rejects in line 5, then there exists a non-*SD* DSR $\gamma \in \Gamma$ that 2-blocks A. Hence, Γ is not $S_\varnothing$-satisfiable by Lemma 18.
Suppose that Γ contains $n+1$ 2-heterogeneous DSRs, assuming that the result holds for n. If 2-Horn-SAT rejects in line 3, then clearly Γ is not $S_\varnothing$-satisfiable. If 2-Horn-SAT rejects in line 5, then Γ is not $S_\varnothing$-satisfiable by Lemma 18. If 2-Horn-SAT rejects in line 7, then $(\Gamma - \{\gamma\}) \cup SD(\gamma)$, which contains n 2-heterogeneous DSRs, is not $S_\varnothing$-satisfiable by the induction hypothesis. By Lemma 16, this is equivalent with Γ not being $S_\varnothing$-satisfiable.

Finally, we can show that 2-Horn-SAT is a polynomial-time algorithm and, thus, show that 2HornDSRSat is polynomial.

Theorem 20 2HornDSRSat is polynomial.

Proof By Lemma 17 and Lemma 19, it is sufficient to show that 2-Horn-SAT is polynomial. The number of recursive calls is bounded by the number of 2-heterogeneous DSRs in the given input. By Theorem 10, we can in polynomial time decide whether a set of atomic DSRs is $S_\varnothing$-satisfiable. Since we need only check a polynomial number of such systems in each recursion, the theorem follows.

5 The Krom Case

In the case of propositional logic, satisfiability can be checked in polynomial time for so-called *Krom formulae* (i.e. CNF formulae with two literals or less per clause). Therefore, in this section we study the corresponding restriction on DSRs in order to check whether the restriction to Krom formulae yields a tractable satisfiability problem. The answer unfortunately turns out to be negative for both the case of $S_\varnothing$-models and S-models. Recall that the class of Krom formulae is not comparable to the class of Horn DSRs. The satisfiability problems are defined as follows:

KromDSRSat$_\varnothing$:
Instance: A finite set Γ of Krom DSRs.
Question: Does there exist an $S_\varnothing$-model of Γ?

KromDSRSat:
Instance: A finite set Γ of Krom DSRs.
Question: Does there exist an S-model of Γ?

Neither of these restrictions were investigated by Jonsson and Drakengren [25].

We shall show that both of these problems are NP-complete.

Theorem 21 KromDSRSat is NP-complete.

Proof DSRSat is in NP, by Lemma 11, so KromDSRSat is in NP.

We show hardness for NP by a polynomial-time reduction from Monotone 3Sat, which is defined as follows.

Instance: Propositional logic formula $\alpha \equiv \bigwedge_i C_i$ over some set $\mathcal{P}$ of propositional symbols, where either $C_i \equiv p \vee q \vee r$ or $C_i \equiv \neg p \vee \neg q \vee \neg r$, for propositional symbols $p, q, r \in \mathcal{P}$.
Question: Is α satisfiable?

First, we show that the following problem is also NP-complete, by reducing Monotone 3Sat to it:

Instance: Propositional logic formula $\alpha \equiv \bigwedge_i C_i$ over some set $\mathcal{P}$ of propositional symbols, where either $C_i \equiv p \vee q \vee r$ or $C_i \equiv \neg p \vee \neg q$, for propositional symbols

$p, q, r \in \mathcal{P}$.
QUESTION: Is α satisfiable?

Let $\alpha = \bigwedge_i C_i$ be an instance of MONOTONE 3SAT, and construct an instance of the above problem as follows: For each $C_i \equiv \neg p \vee \neg q \vee \neg r$, introduce three new variables s_1, s_2 and s_3, and replace C_i by the following four clauses:

$$\neg p \vee \neg s_1, \quad \neg q \vee \neg s_2, \quad \neg r \vee \neg s_3, \quad s_1 \vee s_2 \vee s_3.$$

Denote the formula obtained by performing these replacements by α'. It is easy to see that α is satisfiable iff α' is, and that the reduction is polynomial.

Now, from an instance α of the above NP-complete problem, construct an instance Γ of KROMDSRSAT as follows. First, introduce a new set variable E, and for each propositional symbol p in α, introduce a set variable P. For each such variable P, add a relation $E \subseteq P$. We intend to model that P is false in a model iff $P = E$ holds in that model.

For each $C_i \equiv p \vee q \vee r$, add a relation $P \neq Q \vee R \neq E$, and for each $C_i \equiv \neg p \vee \neg r$, add a relation $P \subseteq E \vee R \subseteq E$. Collect all these formulae in Γ. It is easy to see that Γ has an S-interpretation iff α is satisfiable, and that the reduction is polynomial. The result follows.

Theorem 22 KROMDSRSAT$_\varnothing$ is NP-complete.

Proof Membership in NP follows from Lemma 11.

Let Γ be an instance of KROMDSRSAT, E a fresh variable, $\mathcal{P}$ the set of variables in Γ, and set

$$\Gamma' = \Gamma \cup \{E \,\mathsf{disj}\, E\} \cup \{P \neq E | P \in \mathcal{P}\}.$$

Now Γ' forces every variable not to take the value $\varnothing$ in any model, and thus Γ' has an $S_\varnothing$-model iff Γ has an S-model. The transformation is polynomial, and NP-completeness follows.

We can now summarize the complexity results for the different restricted version of DSRs, from the proofs in this paper and from Jonsson and Drakengren [25]. The result can be found in Table 2.

	DSRSAT	DSRSAT$_\varnothing$
Atomic	P	P
2-Horn	P	P
Horn	P	NPC
Krom	NPC	NPC
General	NPC	NPC

Table 2: The complexity of reasoning about DSRs.

6 Application: Tractable Inference in Intuitionistic Logic

In this section we exploit a connection between topology and intuitionistic logic found by Tarski [34], namely that entailment in intuitionistic logic corresponds to entailment using set constraints and topological models, in order to restrict the general entailment problem of intuitionistic logic to one that is computable in polynomial time. By the fact that the general problem is PSPACE-complete [33, 20], this represents an improvement, but of course at the cost of a more restricted language. To our knowledge the only such known tractable subclass is that of Nebel [27], restricting queries of the form $\phi_1, \ldots, \phi_n \vdash_I \phi_{n+1}$ so that every ϕ_i is a member of the set $\mathcal{I}_b$, the set of *binary clauses*, containing formulae of the following kinds:

$$\neg x, \neg x \vee \neg y, \neg(x \wedge y), \neg x \vee y, x \rightarrow y,$$

$$x, x \vee y, x \wedge y, \neg x \wedge y, \neg x \wedge \neg y.$$

Later, we shall see that our class is incomparable with Nebel's.

First, we need a few concepts.

Definition 23 A *topological space* is a pair $\langle \mathcal{U}, \mathcal{T} \rangle$, where $\{\varnothing, \mathcal{U}\} \subseteq \mathcal{T} \subseteq 2^{\mathcal{U}}$, and $\mathcal{T}$ is closed under finite intersections and arbitrary unions. The elements of $\mathcal{T}$ are called the *open sets*.

Given any set $Y \subseteq \mathcal{U}$, we define its *interior* $i(Y)$ as the union of all open sets contained in Y, i.e.

$$i(Y) = \bigcup_{t \in \mathcal{T}, t \subseteq Y} t.$$

Note that $i(Y)$ is always open, and that Y is open iff $i(Y) = Y$, so any topology is determined by $\langle \mathcal{U}, i \rangle$. $\square$

For a concise introduction to topology, see Arkhangelskii and Fedorchuk [5].

Much of the following notation is taken from Bennett [6].

Definition 24 Let $\mathcal{P}$ be an enumerable set of propositional symbols. Then α is a formula in propositional intuitionistic logic (or an $\mathcal{I}_0$ formula) over $\mathcal{P}$ iff α is constructed from $\mathcal{P}$ and the connectives $\wedge$, $\vee$, $\neg$ and $\rightarrow$, as usual. We use $\vdash_I$ to denote entailment in $\mathcal{I}_0$.

A *topological interpretation* of $\mathcal{I}_0$ is a tuple $\langle \mathcal{U}, \mathcal{T}, \mathcal{P}, d \rangle$, where $\langle \mathcal{U}, \mathcal{T} \rangle$ is a topological space (inducing the interior mapping i), $\mathcal{P}$ is an enumerable set of propositional symbols, and d is a denotation function assigning each $p \in \mathcal{P}$ a set in $\mathcal{T}$. The domain of d is extended to all $\mathcal{I}_0$ formulae formed from these variables by defining

- $d(\neg\alpha) = i(\overline{d(\alpha)})$,
- $d(\alpha \wedge \beta) = d(\alpha) \cap d(\beta)$,
- $d(\alpha \vee \beta) = d(\alpha) \cup d(\beta)$, and
- $d(\alpha \rightarrow \beta) = i(\overline{d(\alpha)} \cup d(\beta))$.

We use $\models_T$ to denote topological entailment, i.e. entailment between set constraints which may contain the interior operator i, using topological interpretations. $\Box$

Tarski [34] has established the following connection between topology and intuitionistic logic.

Theorem 25 Let α be an $\mathcal{I}_0$ formula. Then $\vdash_I \alpha$ iff $\models_T d(\alpha) = \mathcal{U}$. $\Box$

The following generalisation was made by Bennett [6].

Theorem 26 Let $\alpha_1, \ldots, \alpha_n, \beta$ be $\mathcal{I}_0$ formulae. Then

$$d(\alpha_1) = \mathcal{U} \wedge \ldots \wedge d(\alpha_n) = \mathcal{U} \models_T d(\beta) = \mathcal{U}$$

iff

$$\alpha_1, \ldots, \alpha_n \vdash_I \beta.$$

$\Box$

Next, a consequence of Theorem 26, needed in order to use a satisfiability algorithm for checking entailment.

Corollary 27 Let $\alpha_1, \ldots, \alpha_n, \beta$ be $\mathcal{I}_0$ formulae. Then

$$\alpha_1, \ldots, \alpha_n \vdash_I \beta$$

iff

$$d(\alpha_1) = \mathcal{U} \wedge \ldots \wedge d(\alpha_n) = \mathcal{U} \wedge d(\beta) \neq \mathcal{U}$$

does not have a topological model.

Proof Trivial.

We shall use the 2-Horn-SAT algorithm for reasoning with the set constraints imposed by the $\mathcal{I}_0$ formulae through the d function. For this, we need a few results.

Proposition 28 Let Γ be a set of $\mathcal{I}_0$ formulae and α_i, $i \in \{1 \ldots n\}$, $\mathcal{I}_0$ formulae. Then $\Gamma \vdash_I \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$ iff $\Gamma \vdash_I \alpha_i$ for every $i \in \{1 \ldots n\}$.

Proof Easily obtained from the axioms of intuitionistic logic. For a list of these, see, for instance, Dragalin [11].

Proposition 29 Let Γ be a set of $\mathcal{I}_0$ formulae and α, β $\mathcal{I}_0$ formulae. Then $\Gamma \vdash_I \alpha \rightarrow \beta$ iff $\Gamma, \alpha \vdash_I \beta$.

Proof See e.g. Dragalin [11].

The following theorem makes it possible to reason about sets alone, bypassing the requirement of topological spaces for the connection with intuitionistic logic stated in Theorem 25.

Theorem 30 Let Γ be a set of DSRs where the set variable symbols (including $\mathcal{U}$ and $\varnothing$) are taken from a set $\mathcal{X}$, and construct the set Γ' as

$$\Gamma' = \Gamma \cup \{X \subseteq \mathcal{U} | X \in \mathcal{X}\} \cup \{\varnothing \operatorname{\mathsf{disj}} \varnothing\}.$$

If there exists an $S_\varnothing$-model M for Γ', then there exists a topological space $\langle A, \mathcal{T}\rangle$ such that $M(\mathcal{U}) = A$, $M(\varnothing) = \varnothing$ and for each $X \in \mathcal{X}$, $M(\mathcal{X}) \in \mathcal{T}$.

Proof Take an $S_\varnothing$-model M for Γ', set $\mathcal{B}' = \{M(X)|X \in \mathcal{X}\}$, and let $\mathcal{T}$ be obtained from $\mathcal{B}'$ by closing it under finite intersections and arbitrary unions ($\mathcal{B}'$ is a *subbasis* for $\mathcal{T}$). The relations in Γ' guarantee that every set in $\mathcal{T}$ is a subset of $M(\mathcal{U})$, so we can set $A = M(\mathcal{U})$. We also have $M(\varnothing) = \varnothing$, by the relation $\varnothing \operatorname{\mathsf{disj}} \varnothing$ included in Γ'.

The next result is crucial for translating in intuitionistic formulae into set constraints.

Proposition 31 First abbreviate "X is satisfiable iff Y is satisfiable" as "X siff Y", X and Y being set constraints, and "satisfiable" meaning having a topological model.

- $d(\alpha \wedge \beta) = \varnothing$ iff $d(\alpha) \operatorname{\mathsf{disj}} d(\beta)$
- $d(\alpha_1 \vee \ldots \vee \alpha_n) = \varnothing$ iff $d(\alpha_1) = \varnothing \wedge \ldots \wedge d(\alpha_n) = \varnothing$
- $d(\neg\alpha) = \mathcal{U}$ iff $d(\alpha) = \varnothing$
- $d(\alpha_1 \wedge \ldots \wedge \alpha_n) = \mathcal{U}$ iff $d(\alpha_1) = \mathcal{U} \wedge \ldots \wedge d(\alpha_n) = \mathcal{U}$
- $d(\alpha \to \beta) = \mathcal{U}$ iff $d(\alpha) \subseteq d(\beta)$
- $d(\alpha \to \neg\beta) = \mathcal{U}$ iff $d(\alpha) \operatorname{\mathsf{disj}} d(\beta)$
- $d(\alpha_1 \wedge \ldots \wedge \alpha_n) \neq \varnothing$ siff
  $X \neq \varnothing \wedge X \subseteq d(\alpha_1) \wedge \ldots \wedge X \subseteq d(\alpha_n)$,
  for a fresh variable X
- $d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \varnothing$ iff $d(\alpha_1) \neq \varnothing \vee \ldots \vee d(\alpha_n) \neq \varnothing$
- $d(\neg\alpha) \neq \mathcal{U}$ iff $d(\alpha) \neq \varnothing$
- $d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \mathcal{U}$ siff
  $X \neq \varnothing \wedge X \operatorname{\mathsf{disj}} d(\alpha_1) \wedge \ldots \wedge X \operatorname{\mathsf{disj}} d(\alpha_n)$,
  for a fresh variable X

Proof

- $d(\alpha \wedge \beta) = \varnothing$ iff $d(\alpha) \cap d(\beta) = \varnothing$ iff $d(\alpha) \operatorname{\mathsf{disj}} d(\beta)$
- $d(\alpha_1 \vee \ldots \vee \alpha_n) = \varnothing$ iff
  $d(\alpha_1) \cup \ldots \cup d(\alpha_n) = \varnothing$ iff $d(\alpha_1) = \varnothing \wedge \ldots \wedge d(\alpha_n) = \varnothing$
- $d(\neg\alpha) = \mathcal{U}$ iff $i(\overline{d(\alpha)}) = \mathcal{U}$ iff $\overline{d(\alpha)} = \mathcal{U}$ iff $d(\alpha) = \varnothing$
- $d(\alpha_1 \wedge \ldots \wedge \alpha_n) = \mathcal{U}$ iff $d(\alpha_1) \cap \ldots \cap d(\alpha_n) = \mathcal{U}$ iff
  $d(\alpha_1) = \mathcal{U} \wedge \ldots \wedge d(\alpha_n) = \mathcal{U}$
- $d(\alpha \to \beta) = \mathcal{U}$ iff $i(\overline{d(\alpha)} \cup d(\beta)) = \mathcal{U}$ iff
  $\overline{d(\alpha)} \cup d(\beta) = \mathcal{U}$ iff $d(\alpha) \subseteq d(\beta)$

- $d(\alpha \to \neg\beta) = \mathcal{U}$ iff $i(\overline{d(\alpha)} \cup d(\neg\beta)) = \mathcal{U}$ iff $\overline{d(\alpha)} \cup d(\neg\beta) = \mathcal{U}$ iff $d(\alpha) \subseteq d(\neg\beta)$ iff $d(\alpha) \subseteq i(\overline{d(\beta)})$ iff $d(\alpha) \subseteq \overline{d(\beta)}$ iff $d(\alpha)\,\mathsf{disj}\,d(\beta)$

- $d(\alpha_1 \wedge \ldots \wedge \alpha_n) \neq \varnothing$ iff $d(\alpha_1) \cap \ldots \cap d(\alpha_n) \neq \varnothing$ siff $X \neq \varnothing \wedge X \subseteq d(\alpha_1) \wedge \ldots \wedge X \subseteq d(\alpha_n)$, for a fresh variable X

- $d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \varnothing$ iff $d(\alpha_1) \cup \ldots \cup d(\alpha_n) \neq \varnothing$ iff $d(\alpha_1) \neq \varnothing \vee \ldots \vee d(\alpha_n) \neq \varnothing$

- $d(\neg\alpha) \neq \mathcal{U}$ iff $i(\overline{d(\alpha)}) \neq \mathcal{U}$ iff $\overline{d(\alpha)} \neq \mathcal{U}$ iff $d(\alpha) \neq \varnothing$

- $d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \mathcal{U}$ iff $d(\alpha_1) \cup \ldots \cup d(\alpha_n) \neq \mathcal{U}$ siff $X \neq \varnothing \wedge X\,\mathsf{disj}\,d(\alpha_1) \wedge \ldots \wedge X\,\mathsf{disj}\,d(\alpha_n)$, for a fresh variable X

The actual translation procedure is carried out using a rewrite relation, which is defined next. The motivation for how the relation is defined is that given formulae on a certain form, we always obtain a certain kind of set constraints, for which checking satisfiability is a polynomial-time problem.

Definition 32 Define the relations $\rightsquigarrow_i$ on set constraints[1], by the following, where α, α_i and β are $\mathcal{I}_0$ formulae, and each new variable introduced is assumed to be unique.

$$
\begin{array}{lll}
d(\alpha \wedge \beta) = \varnothing & \rightsquigarrow_1 & d(\alpha)\,\mathsf{disj}\,d(\beta) \\
d(\alpha_1 \vee \ldots \vee \alpha_n) = \varnothing & \rightsquigarrow_1 & d(\alpha_1) = \varnothing \wedge \ldots \wedge d(\alpha_n) = \varnothing \\
\\
d(\neg\alpha) = \mathcal{U} & \rightsquigarrow_2 & d(\alpha) = \varnothing \\
d(\alpha_1 \wedge \ldots \wedge \alpha_n) = \mathcal{U} & \rightsquigarrow_2 & d(\alpha_1) = \mathcal{U} \wedge \ldots \wedge d(\alpha_n) = \mathcal{U} \\
d(\alpha \to \beta) = \mathcal{U} & \rightsquigarrow_2 & d(\alpha) \subseteq d(\beta) \\
d(\alpha \to \neg\beta) = \mathcal{U} & \rightsquigarrow_2 & d(\alpha)\,\mathsf{disj}\,d(\beta) \\
\\
d(\alpha_1 \wedge \ldots \wedge \alpha_n) \neq \varnothing & \rightsquigarrow_3 & X \neq \varnothing \wedge X \subseteq d(\alpha_1) \wedge \ldots \wedge X \subseteq d(\alpha_n) \\
d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \varnothing & \rightsquigarrow_3 & d(\alpha_1) \neq \varnothing \vee \ldots \vee d(\alpha_n) \neq \varnothing \\
\\
d(\neg\alpha) \neq \mathcal{U} & \rightsquigarrow_4 & d(\alpha) \neq \varnothing \\
d(\alpha_1 \vee \ldots \vee \alpha_n) \neq \mathcal{U} & \rightsquigarrow_4 & X \neq \varnothing \wedge X\,\mathsf{disj}\,d(\alpha_1) \wedge \ldots \wedge X\,\mathsf{disj}\,d(\alpha_n)
\end{array}
$$

Also define $\rightsquigarrow = \bigcup_i \rightsquigarrow_i$, and denote by $\Delta \rightsquigarrow^* \Delta'$ that there exists a sequence $\Delta \rightsquigarrow \ldots \rightsquigarrow \Delta'$, that is *maximal*, i.e. it is impossible to further rewrite Δ' with $\rightsquigarrow$. □

Corollary 33 Let Δ be a set constraint. Then there exists a set constraint Δ' such that $\Delta \rightsquigarrow^* \Delta'$, and for no other $\Delta'' \neq \Delta'$, we have $\Delta \rightsquigarrow^* \Delta''$. Furthermore, the rewriting from Δ to Δ' can be done in polynomial time.

[1]We are satisfied by viewing the constraints as mathematical expressions, and refrain from introducing the machinery of a formal language for set expressions. Fortunately, the notion of a mathematical expression is well-defined enough.

Proof Since the $\rightsquigarrow_i$ relations can only rewrite each atomic relation in exactly one way, and in each step one redex is removed (a *redex* is a possibility to rewrite), Δ' will be uniquely determined (modulo the choices of fresh variables) by Δ. Obviously the rewriting terminates, and in polynomial time.

Corollary 34 Let Δ be a set constraint, and suppose that $\Delta \rightsquigarrow^* \Delta'$. Then Δ is satisfiable iff Δ' is satisfiable.

Proof Directly from Proposition 31.

Now, we can define the languages we need for specifying the tractable class for intuitionistic logic.

Definition 35 Let $\mathcal{P}$ be a set of propositional symbols. Define the sets $\mathcal{L}_i$, for $1 \leq i \leq 5$ in BNF as follows, letting p, q, p_j denote elements in $\mathcal{P}$, and α^i, α^i_j elements of $\mathcal{L}_i$.

$$\mathcal{L}_1 ::= \; p \mid p_1 \wedge p_2 \mid \alpha^1_1 \vee \ldots \vee \alpha^1_n$$
$$\mathcal{L}_2 ::= \; p \mid \neg\alpha^1 \mid \alpha^2_1 \wedge \ldots \wedge \alpha^2_n \mid p \to q \mid p \to \neg q$$
$$\mathcal{L}_3 ::= \; p \mid p_1 \wedge \ldots \wedge p_n \mid p_1 \vee \ldots \vee p_n$$
$$\mathcal{L}_4 ::= \; p \mid \neg\alpha^3 \mid p_1 \vee \ldots \vee p_n$$
$$\mathcal{L}_5 ::= \; \alpha^4_1 \wedge \ldots \wedge \alpha^4_n \mid (\alpha^2_1 \wedge \ldots \wedge \alpha^2_n) \to \alpha^5$$

□

We immediately see that the BNF definitions of the languages $\mathcal{L}_i$ follow the definitions of the rewrite relations $\rightsquigarrow_i$, except for $\mathcal{L}_5$. For example, taking $p \wedge q \in \mathcal{L}_1$, $d(p \wedge q) = \varnothing \rightsquigarrow_1 d(p) \,\mathsf{disj}\, d(q)$.

Proposition 36 Let $\alpha \in \mathcal{L}_2$ and $\beta \in \mathcal{L}_4$. Then $d(\alpha) = \mathcal{U} \rightsquigarrow^* \Delta$ and $d(\beta) \neq \mathcal{U} \rightsquigarrow^* \Delta'$, where Δ and Δ' are conjunctions of 2-Horn DSRs, satisfying that whenever an expression $d(\alpha)$ occurs in them, we have $\alpha \in \mathcal{P}$.

Proof We first prove the result for $\alpha \in \mathcal{L}_2$ by induction, by also proving the stronger result that for $\beta \in \mathcal{L}_1$, the corresponding result holds for $d(\beta) = \varnothing$.

We begin with the basis steps:

- For $\beta \equiv p$, the result is trivial.
- For $\beta \equiv p_1 \wedge p_2$, the result follows by the definition of $\rightsquigarrow_1$.
- For $\alpha \equiv p$, the result is trivial.
- For $\alpha \equiv p \to q$, the result is immediate by the definition of $\rightsquigarrow_2$.
- For $\alpha \equiv p \to \neg q$, the result is immediate by the definition of $\rightsquigarrow_2$.

Next, the induction steps:

- For $\beta \equiv \alpha^1_1 \vee \ldots \vee \alpha^1_n \in \mathcal{L}_1$, we get $d(\beta) = \varnothing \rightsquigarrow_1 d(\alpha^1_1) = \varnothing \wedge \ldots \wedge d(\alpha^1_n) = \varnothing$, and the result follows by induction.

- For $\alpha \equiv \neg\alpha^1$, we get $d(\neg\alpha^1) = \mathcal{U} \rightsquigarrow_2 d(\alpha^1) = \varnothing$, and the result follows by induction.
- For $\alpha \equiv \alpha_1^2 \wedge \ldots \wedge \alpha_n^2$, we get $d(\alpha_1^2 \wedge \ldots \wedge \alpha_n^2) = \mathcal{U} \rightsquigarrow_2 d(\alpha_1^2) = \mathcal{U} \wedge \ldots \wedge d(\alpha_1^2) = \mathcal{U}$, and the result follows by induction.

For $\beta \in \mathcal{L}_4$, no induction is needed. However, we first need to prove the result that for $\alpha \in \mathcal{L}_3$, the same holds for $d(\alpha) \neq \varnothing$.

- For $\alpha \equiv p$, the result is trivial.
- For $\alpha \equiv p_1 \wedge \ldots \wedge p_n$, the result follows from the definition of $\rightsquigarrow_3$.
- For $\alpha \equiv p_1 \vee \ldots \vee p_n$, the result follows from the definition of $\rightsquigarrow_3$.

Now the results for $\beta \in \mathcal{L}_4$:

- For $\beta \equiv p$, the result is trivial.
- For $\beta \equiv p_1 \vee \ldots \vee p_n$, the result follows from the definition of $\rightsquigarrow_4$.
- For $\beta \equiv \neg\alpha^3$, we get $d(\neg\alpha^3) \neq \mathcal{U} \rightsquigarrow_4 d(\alpha^3) \neq \varnothing$, and the result follows by the property of $\mathcal{L}_3$.

Algorithm 37 (Entailment(Γ, β))

input A finite set $\Gamma = \{\alpha_1, \ldots, \alpha_m\} \subseteq \mathcal{L}_2$, $\beta \in \mathcal{L}_5$.
1 **if** $\beta = \bigwedge_{i=1}^{n} \beta_i$ and $n > 1$ **then**
2   **for** each $\beta_i \in \beta$
3   **if** not Entailment(Γ, β_i) **then**
4   **reject**
5   **accept**
6 **elsif** $\beta \equiv \gamma \rightarrow \delta$ **then**
7   Entailment($\Gamma \cup \{\gamma\}, \delta$)
8 **else**
9   Compute $d(\bigwedge \Gamma) = \mathcal{U} \wedge d(\beta) \neq \mathcal{U} \rightsquigarrow^* \Delta$
10   2-Horn-SAT($\{\Delta\} \cup \{d(p) \subseteq \mathcal{U} | p \in \mathcal{P}\} \cup \{\varnothing \;\mathsf{disj}\; \varnothing\}$)
11 **endif**

□

Theorem 38 Let $\Gamma \subseteq \mathcal{L}_2$ and $\beta \in \mathcal{L}_5$. Then Algorithm 37 decides whether $\Gamma \vdash_I \beta$, and furthermore, it does so in polynomial time.

Proof First, if $\beta \equiv \bigwedge_i \beta_i$, then by Proposition 28 it is necessary and sufficient to check entailment for the β_is separately, which is done in lines 1-5. Then, as long as

$\beta \equiv \gamma \to \delta$, we may move γ to the left of $\vdash_I$, and keep δ, by Proposition 29, which is done in lines 6-7. Then we are left at line 9 with a problem of the form

$$\Gamma \vdash_I \beta,$$

where $\Gamma \subseteq \mathcal{L}_2$ and $\beta \in \mathcal{L}_4$, by the construction of $\mathcal{L}_5$. Since $\mathcal{L}_2$ is closed under finite conjunction, $\bigwedge \Gamma \in \mathcal{L}_2$. Thus, by Proposition 36 we can rewrite the constraint

$$d(\bigwedge \Gamma) = \mathcal{U} \wedge d(\beta) \neq \mathcal{U}$$

using $\rightsquigarrow^*$ into a set constraint Δ where all occurrences of $d(\alpha)$ satisfy $\alpha \in \mathcal{P}$. Furthermore, if we expand each usage of the $=$ relation into two $\subseteq$ relations, this constraint is a conjunction of 2-Horn DSRs in the variables $d(p)$ (since the values of propositional symbols in a model are independent, we can consider the expressions $d(p)$ to be variables), and it is satisfiable iff the original constraint is satisfiable, by Corollary 34. By Corollary 27, Δ is satisfiable by a topological model iff $\Gamma \vdash_I \beta$.

Let $\Omega = \{\Delta\} \cup \{d(p) \subseteq \mathcal{U} | p \in \mathcal{P}\} \cup \{\varnothing \,\mathsf{disj}\, \varnothing\}$. If now Δ has a topological model, then it is certainly an $S_\varnothing$-model of Ω. Conversely, if Ω has an $S_\varnothing$-model, then by Theorem 30, Δ has a topological model, and correctness follows.

Polynomiality follows from the polynomiality of 2-Horn-SAT, since the rewritings in line 9 using $\rightsquigarrow^*$ are clearly polynomial.

Note that this class does not subsume that of Nebel [27], since, for example, formulae on the form $\neg x \vee \neg y$ are not included here. However, Nebel's class does not subsume this class either, since e.g. $(x \to y) \to (w \to z)$ is not included therein.

7 Discussion

Having identified a tractable fragment of intuitionistic logic, several interesting questions arise. Nebel [27] uses his tractable class in order to obtain a tractable class for spatial reasoning, in the so-called RCC-8 spatial algebra [28, 29]. This class was later extended by Renz and Nebel [30] to a maximal tractable subclass of that spatial algebra. Thus, since this class is incomparable to Nebel's, is it possible to use it to obtain other tractable subclasses of RCC-8, by reducing their satisfiability problem to that of intuitionistic logic? For the case of the RCC-5 spatial algebra, all possible cases of tractable subclasses have already been characterised [30, 24].

Another relevant question is whether we can use our tractable class of set constraints for tractable inference in other logical systems. For instance, Renz and Nebel [30] use classes of modal logics in order to prove classes of the RCC-5 and RCC-8 spatial algebras tractable, so there is obviously a connection between these formalisms, and our reasoning about sets. It would thus be interesting to investigate whether we can find tractable classes of these logics, and related formalisms.

8 Conclusions

We have found a new class of set constraints, the *2-Horn DSRs*, for which reasoning is tractable (i.e. polynomial). This class is used for constructing a new class in intuitionistic logic, where tractable inference can be performed. In order to investigate the borderline between tractable and intractable reasoning with set constraints, we prove that a class of set constraints previously proved tractable, the *Horn DSRs*, becomes intractable if the empty set is allowed as values for set variables. In addition, we prove that the restriction to *Krom formulae*, which is known to be polynomial for propositional logic, is still NP-complete for set constraints, whether or not the empty set is allowed for set variables.

Acknowledgements

Thanks to the anonymous reviewers for constructive comments.

References

[1] A. Aiken, D. Kozen, and E. Wimmers. Decidability of systems of set constraints with negative constraints. Research Report RS-94-32, BRICS, Department of Computer Science, Århus University, October 1994.

[2] A. Aiken and T. K. Lakshman. Directional type checking of logic programs. In *Proceedings of the 1st International Static Analysis Symposium*, pages 163–173, 1994.

[3] A. Aiken and B. Murphy. Implementing regular tree expressions. In *Proceedings of the 1991 Conference on Functional Programming Languages and Computer Architecture*, pages 427–447, August 1991.

[4] A. Aiken, E. Wimmers, and T. K. Lakshman. Soft typing with conditional types. In *Proceedings of the 21st annual ACM Symposium on Principles of Programming Languages*, pages 163–173, 1994.

[5] A.V. Arkhangelskii and V. V. Fedorchuk. The basic concepts and constructions of general topology. In A. V. Arkhangelskii and L. S. Pontryagin, editors, *General Topology I*, volume 17 of *Encyclopaedia of Mathematical Sciences*, chapter 1. Springer-Verlag, 1990.

[6] B. Bennett. Spatial reasoning with propositional logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR '94)*, pages 165–176, Bonn, Germany, May 1994. Morgan Kaufmann.

[7] R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[8] J. A. Brzozowski and E. Leiss. On equations for regular languages, finite automata and sequential networks. *Theoretical Computer Science*, 10:19–35, 1980.

[9] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamics logics. In *Proceedings of the 12th (US) National Conference on Artificial Intelligence (AAAI '94)*, Seattle, WA, USA, July–August 1994. American Association for Artificial Intelligence.

[10] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR '91)*, Cambridge, MA, USA, 1991.

[11] A. G. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*, volume 67 of *Translations of Mathematical Monographs*. American Mathematical Society, 1988.

[12] Thomas Drakengren. *Algorithms and Complexity for Temporal and Spatial Formalisms*. Doctoral dissertation, Linköping University, October 1997. Available from `http://www.ida.liu.se/publications/t-phd/1997/0498/`.

[13] Michael Garey and David Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[14] R. Gilleron, S. Tison, and M. Tommasi. Solving systems of set constraints with negated subset relationships. In *Proceedings of the 34th Annual IEEE Symposium on the Foundations of Computer Science*, pages 372–380. IEEE Computer Society, November 1993.

[15] R. Givan, D. Kozen, D. McAllester, and C. Witty. Tarskian set constraints. In *Proceedings of the 11th annual IEEE symposium on logic in computer science*, New Jersey, USA, July 1996.

[16] N. Heintze. *Set Based Program Analysis*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1992.

[17] N. Heintze and J. Jaffar. A decision procedure for a class of set constraints. In *Proceedings of the 5th Symposium on Logic in Computer Science*, pages 42–51, June 1990.

[18] N. Heintze and J. Jaffar. An engine for logic program analysis. In *Symposium on Logic in Computer Science*, pages 318–328, June 1992.

[19] Arend Heyting. *Intuitionism: An Introduction*. North-Holland, 3rd edition, 1971.

[20] J. Hudelmaier. An $O(nlogn)$-space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.

[21] B. Jónsson and A. Tarski. Boolean algebras with operators. Part I. *American Journal of Mathematics*, 73:891–939, 1951.

[22] B. Jónsson and A. Tarski. Boolean algebras with operators. Part II. *American Journal of Mathematics*, 74:127–167, 1952.

[23] Peter Jonsson and Christer Bäckström. A linear-programming approach to temporal reasoning. In *Proceedings of the 13th (US) National Conference on Artificial Intelligence (AAAI '96)*, pages 1235–1240, Portland, OR, USA, August 1996. American Association for Artificial Intelligence.

[24] Peter Jonsson and Thomas Drakengren. A complete classification of tractability in RCC-5. *Journal of Artificial Intelligence Research*, 6:211–221, 1997.

[25] Peter Jonsson and Thomas Drakengren. Qualitative reasoning about sets applied to spatial reasoning. In [12], pages 195–209, 1997.

[26] K. Marriott and M. Odersky. Systems of negative Boolean constraints. Research Report YALEU/DCS/RR-900, Computer Science Department, Yale University, April 1992.

[27] Bernhard Nebel. Computational properties of qualitative spatial reasoning: First results. In I. Wachsmuth, C-R. Rollinger, and W. Brauer, editors, *KI '95: Advances in Artificial Intelligence*, pages 233–244, Bielefeld, Germany, September 1995. Springer-Verlag.

[28] D. A. Randell and A. G. Cohn. Modelling topological and metrical properties of physical processes. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR '89)*, pages 55–66, Toronto, ON, Canada, May 1989. Morgan Kaufmann.

[29] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In Bill Swartout and Bernhard Nebel, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR '92)*, pages 165–176, Cambridge, MA, USA, October 1992. Morgan Kaufmann.

[30] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connected calculus. In Martha E. Pollack, editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI '97)*, Nagoya, Japan, August 1997. Morgan Kaufmann.

[31] J. C. Reynolds. Automatic computation of data set definitions. *Information Processing Letters*, 68:456–461, 1969.

[32] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[33] R. Statman. Intuitionistic logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, 1979.

[34] Alfred Tarski. Sentential calculus and topology. In *Logic, Semantics, Metamathematics*, chapter 17. Oxford Clarendon Press, 1956.