Logical Aspects of Set Constraints*

Dexter Kozen

TR 94-1421
May 1994

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

* Proc. 1993 Conf. Computer Science Logic (CSL `93), Eur. Assoc. Comput. Sci. Logic, September 1993.

Logical Aspects of Set Constraints*

Dexter Kozen

Computer Science Department
Cornell University
Ithaca, New York 14853, USA
kozen@cs.cornell.edu

Abstract. Set constraints are inclusion relations between sets of ground terms over a ranked alphabet. They have been used extensively in program analysis and type inference. Here we present an equational axiomatization of the algebra of set constraints. Models of these axioms are called *termset algebras*. They are related to the Boolean algebras with operators of Jónsson and Tarski. We also define a family of combinatorial models called *topological term automata*, which are essentially the term automata studied by Kozen, Palsberg, and Schwartzbach endowed with a topology such that all relevant operations are continuous. These models are similar to Kripke frames for modal or dynamic logic. We establish a Stone duality between termset algebras and topological term automata, and use this to derive a completeness theorem for a related multidimensional modal logic. Finally, we prove a small model property by filtration, and argue that this result contains the essence of several algorithms appearing in the literature on set constraints.

1 Introduction

Set constraints are inclusion relations between sets of ground terms over a ranked alphabet. They have been used extensively in program analysis and type inference [29, 21, 27, 28, 32, 20, 3, 4].

Several algorithms for solving general systems of set constraints have appeared [5, 1, 2, 7, 14, 15, 30, 9]. These algorithms use a variety of interesting techniques and expose a rich structure touching on monadic second-order logic, automata on infinite trees, and combinatorics on hypergraphs. Although these several approaches may appear to differ radically, there are common threads that underlie them all.

When working with set constraints, it is soon apparent that many basic properties follow from a few simple algebraic laws, and that a large part of the basic theory can be developed from a purely algebraic standpoint, independent of the standard set-theoretic interpretation. In the process of developing this theory, one discovers that some constructions in the recent literature on set constraints

* Proc. 1993 Conf. Computer Science Logic (CSL'93), Eur. Assoc. Comput. Sci. Logic, September, 1993.

can be recast in more classical contexts. One rather unexpected connection is that the algebraic models presented here and the term automata of [25], which were developed independently in a completely different context, turn out to be Stone duals.

In this paper, we present an equational axiomatization of the algebra of sets of ground terms over a ranked alphabet. We call the models of these axioms *termset algebras*. These models form an equational variety and are related to the Boolean algebras with operators and complex algebras (algebras of subsets of another algebra) introduced by Jónsson and Tarski [22, 23]; see also [16]. We also define a family of topological models called *topological term automata*, which are essentially the term automata of [25] endowed with a topology such that all relevant operations are continuous. We show that these two classes of structures are Stone duals (see [18, 17]).

We also identify a subfamily of term automata in which the topology is induced by labelings on the states. These models are quite similar to Kripke frames for modal or dynamic logic (see [10, 31, 17, 26]) and provide a semantics for a kind of multidimensional modal logic (*cf.* [31]). We give a completeness result for this logic. Finally, we prove a small model property by filtration, a classical technique of modal logic, and argue that this result contains the essence of [1, Theorem 5.1], [14, Proposition 14], and the proof of decidability of the Monadic Class given in [13, §2.1] on which the algorithm of [7] is based.

2 Termset Algebras

Let $\mathbf{B}$ denote the usual signature of Boolean algebra consisting of symbols $+$ (join), $\cdot$ (meet), $\neg$ (negation), 0 (bottom), and 1 (top). The operators $\rightarrow$ (implication), $-$ (difference), and $\oplus$ (symmetric difference) are defined as usual:

$$\begin{aligned} x \rightarrow y &= \neg x + y \\ x - y &= \neg(x \rightarrow y) = x \cdot \neg y \\ x \oplus y &= (x - y) + (y - x)\ . \end{aligned}$$

The expressions $x \leq y$ and xy are used to abbreviate $x + y = y$ and $x \cdot y$, respectively.

Let Σ be a finite ranked alphabet disjoint from $\mathbf{B}$ consisting of various function symbols $f, g, h, \ldots$, each with an associated finite *arity*. *Constants* are symbols of arity 0 and are denoted $a, b, c, \ldots$ In general, the use of any expression of the form $f(x_1, \ldots, x_n)$ carries the implicit assumption that f is of arity n.

Definition 1. A *(Σ-)termset algebra* is a structure $\mathcal{A}$ of signature $\Sigma + \mathbf{B}$ such that $\mathcal{A}$ is a Boolean algebra with respect to the operators $\mathbf{B}$ and satisfies

$$f(\ldots, x + y, \ldots) = f(\ldots, x, \ldots) + f(\ldots, y, \ldots) \tag{1}$$

$$f(\ldots, x - y, \ldots) = f(\ldots, x, \ldots) - f(\ldots, y, \ldots) \tag{2}$$

$$\sum_{f \in \Sigma} f(1, \ldots, 1) = 1 \tag{3}$$

$$f(1, \ldots, 1) \cdot g(1, \ldots, 1) = 0\ , \quad f \neq g\ . \tag{4}$$

The ellipses in (1) and (2) indicate that the explicitly given arguments occur in corresponding places, and that implicit arguments in corresponding places agree.

These axioms define an equational variety. Some immediate consequences are

$$f(\ldots,0,\ldots) = 0 \tag{5}$$

$$f(\ldots,\neg x,\ldots) = f(\ldots,1,\ldots) - f(\ldots,x,\ldots) \tag{6}$$

$$f(\ldots,xy,\ldots) = f(\ldots,x,\ldots) \cdot f(\ldots,y,\ldots) \tag{7}$$

$$f(\ldots,x \oplus y,\ldots) = f(\ldots,x,\ldots) \oplus f(\ldots,y,\ldots) \tag{8}$$

$$x \leq y \Rightarrow f(\ldots,x,\ldots) \leq f(\ldots,y,\ldots)\ . \tag{9}$$

2.1 Examples

The standard interpretation of set expressions found in the literature on set constraints (see *e.g.* [5, 1]) is a model of these axioms. We call this model the *standard termset algebra*. Elements are the subsets of T_Σ, the set of ground terms over Σ. The Boolean operators have their usual set-theoretic interpretations, and

$$f(A_1,\ldots,A_n) = \{f(t_1,\ldots,t_n) \in T_\Sigma \mid t_i \in A_i,\ 1 \leq i \leq n\}\ . \tag{10}$$

The set $f(A_1,\ldots,A_n)$ can be viewed as a marked direct product of the A_i; elements are n-tuples marked with f.

We can also define similar termset algebras consisting of sets of finite and infinite terms or sets of regular terms [12]. We discuss these models further in §4.2.

There are other examples of termset algebras that have no representation as sets of terms. We will see several of these examples in the sequel.

2.2 Nondeterministic Termset Algebras

Later in the course of this exposition we will consider a weaker axiomatization in which equation (2) is replaced by the inequality

$$f(\ldots,x-y,\ldots) \geq f(\ldots,x,\ldots) - f(\ldots,y,\ldots) \tag{11}$$

and (5), which no longer follows, is postulated as an axiom. Algebraic models of these axioms are called *nondeterministic termset algebras.*

2.3 Boolean Algebras with Operators

Termset algebras are a special case of the *Boolean algebras with operators* introduced by Jónsson and Tarski [22, 23]. These are Boolean algebras satisfying (1) and (5). An example of a Boolean algebra with operators is the *complex algebra* or algebra of subsets of another algebra [16]. We need not postulate (5) as an axiom for termset algebras, since it follows from (2).

The two chief features that distinguish termset algebras from Boolean algebras with operators and complex algebras are characterized by axiom (2) and axioms (3) and (4). We discuss the former in §3 below. To explain the latter, note that in the standard termset algebra, the expression $f(1,\ldots,1)$ denotes the set of all ground terms with head symbol f. Axioms (3) and (4) then capture the intuition that every term has exactly one head symbol.

2.4 Homomorphisms, Ideals, Ultrafilters

A *(termset algebra) homomorphism* is a Boolean algebra homomorphism $h : \mathcal{A} \rightarrow \mathcal{B}$ such that for all $f \in \Sigma$,

$$h(f^{\mathcal{A}}(x_1,\ldots,x_n)) = f^{\mathcal{B}}(h(x_1),\ldots,h(x_n)) \ .$$

An *(termset algebra) ideal* in $\mathcal{A}$ is a Boolean algebra ideal I with the extra property

$$x \in I \Rightarrow f(\ldots,x,\ldots) \in I \ . \tag{12}$$

Ideals are kernels of homomorphisms. Maximal ideals are kernels of homomorphisms into *simple* termset algebras, *i.e.* those with no nontrivial ideals. A *dual ideal* in $\mathcal{A}$ is a set of the form $\{\neg x \mid x \in I\}$, where I is a termset algebra ideal.

An *filter* in $\mathcal{A}$ is a Boolean algebra filter (dual Boolean algebra ideal). An *ultrafilter* is a maximal filter.

To avoid confusion, we always use the terms *filter* and *ultrafilter* in the Boolean algebra sense and *dual ideal* and *maximal dual ideal* in the termset algebra sense. They are not the same: for example, there is an ultrafilter containing $f(a)$, but no dual ideal contains $f(a)$ (for the same reason that the set constraint $f(a) = 1$ is not satisfiable, although here one can give a short algebraic proof using (12)).

Every filter extends to an ultrafilter; this is a standard application of Zorn's Lemma [18]. By the same technique one can show that every dual ideal extends to a maximal dual ideal.

It follows from axioms (3) and (4) that for every ultrafilter u there is exactly one $f \in \Sigma$ with $f(1,\ldots,1) \in u$.

3 Connections with Modal Logic

The operator f in a termset algebra behaves like a deterministic modal possibility operator in each place. If we fix $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$ and define

$$\begin{aligned} \Diamond y &= f(x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n) \\ \Box y &= \neg\Diamond\neg y \ , \end{aligned}$$

then $\Diamond$ and $\Box$ satisfy the usual laws of the minimal normal modal logic K [10, 17]:

$$\Diamond(x+y) = \Diamond x + \Diamond y \tag{13}$$

$$\Diamond 0 = 0 \tag{14}$$

$$(\Diamond x)\cdot(\Box y) \leq \Diamond(xy) \tag{15}$$

$$\Box(x \rightarrow y) \leq \Box x \rightarrow \Box y \tag{16}$$

In addition, because of axiom (2), $\Diamond$ is a *deterministic* modality in the sense used in dynamic logic to model deterministic computation (see [26]):

$$\Diamond x \leq \Box x \tag{17}$$

$$(\Diamond x) \cdot (\Box y) \geq \Diamond(xy) \tag{18}$$

$$\Box(x \rightarrow y) \geq \Box x \rightarrow \Box y \; . \tag{19}$$

The laws (17)–(19) are equivalent in the sense that they are interderivable in the presence of (13)–(16).

4 Term Automata

4.1 Infinite Terms

Infinite terms and regular terms are useful in program logic, program specification and type theory. Regular terms are commonly used to represent recursive types [6, 8, 24, 25]. The following definition is from Courcelle [12].

Definition 2. Let ω denote the set of natural numbers and let Σ be a ranked alphabet. A *(Σ-)term* is a partial function $t : \omega^* \rightarrow \Sigma$ whose domain is nonempty, prefix-closed, and respects arities in the sense that if $t(\alpha)$ is defined then

$$\{i \mid t(\alpha i) \text{ is defined}\} = \{1, 2, \ldots, \text{arity}(t(\alpha))\} \; .$$

A term is *regular* if it has only finitely many subterms up to isomorphism.

Example 1. The finite term $f(g(a), f(a, g(b)))$ is formally a partial map t with domain $\{\epsilon, 1, 2, 11, 21, 22, 221\}$ such that $t(\epsilon) = t(2) = f$, $t(1) = t(22) = g$, $t(11) = t(21) = a$, and $t(221) = b$. The infinite term $f(a, f(a, f(a, \ldots)))$ is formally a map s whose domain is the infinite set described by the regular expression $2^* + 2^*1$ such that $s(\alpha) = f$ for $\alpha \in 2^*$ and $s(\alpha) = a$ for $\alpha \in 2^*1$. The infinite term s is regular since it has two subterms up to isomorphism, namely s and a.

4.2 Term Automata

It is well known that an infinite regular term can be represented by a finite labeled graph such that the infinite term is obtained by "unwinding" the graph (see [12, 11]). We use the automata-theoretic formulation introduced in [24] of this idea.

Definition 3. A *(Σ-)term automaton* is a tuple

$$M = (Q,\ \Sigma,\ \ell,\ \delta)$$

where:

- Q is a set of *states* (not necessarily finite)

– Σ is a ranked alphabet
– $\ell : Q \to \Sigma$ is a *labeling*
– $\delta : Q \times \omega \to Q$ is a partial function such that for all $q \in Q$,

$$\{i \mid \delta(q,i) \text{ is defined}\} = \{1, 2, \ldots, \text{arity}(\ell(q))\} .$$

The function δ extends uniquely to a partial function $\widehat{\delta} : Q \times \omega^* \to Q$ according to the inductive definition

$$\widehat{\delta}(q, \epsilon) = q$$
$$\widehat{\delta}(q, \alpha i) = \delta(\widehat{\delta}(q, \alpha), i) ,$$

with the understanding that δ is strict (undefined if one of its arguments is undefined). For each $q \in Q$, the partial function

$$t_q = \lambda\alpha.\ell(\widehat{\delta}(q, \alpha))$$

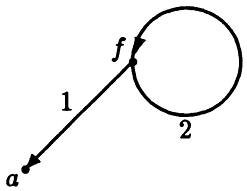
is a Σ-term in the sense of Definition 2.

Every term in the sense of Definition 2 is t_q for some state q of some term automaton. In fact, $t = t_t$ in the syntactic term automaton

$$I_\Sigma = (\{\Sigma\text{-terms}\},\ \Sigma,\ \ell,\ \delta)$$

where $\ell(t) = t(\epsilon)$ and $\delta(t, i) = \lambda\alpha.t(i\alpha)$, $1 \le i \le \text{arity}(\ell(t))$. In this sense the notion of term automaton (Definition 3) is a generalization of the notion of term (Definition 2).

A term is regular iff it is t_q for some state q of some finite term automaton [25, Lemma 8]. For example, if q is the state labeled f in the term automaton



then t_q is the infinite regular term s of Example 1 of §4.1.

The syntactic term automaton I_Σ defined above has a subautomaton R_Σ consisting of the regular terms, which in turn has a subautomaton T_Σ consisting of the finite terms.

4.3 Term Automata and Set-Theoretic Termset Algebras

Let $M = (Q,\ \Sigma,\ \ell,\ \delta)$ be a term automaton. For $f \in \Sigma$, define the partial function $R_f^M : Q \to Q^n$ and the set-theoretic function $f^M : (2^Q)^n \to 2^Q$ by

$$R_f^M(q) = \begin{cases} (\delta(q,1), \ldots, \delta(q,n)) \text{ , if } \ell(q) = f \\ \text{undefined ,} \qquad\qquad \text{otherwise.} \end{cases} \tag{20}$$

$$\begin{aligned} f^M(A_1, \ldots, A_n) &= \{q \in Q \mid \ell(q) = f \text{ and } \delta(q,i) \in A_i,\ 1 \le i \le n\} \\ &= (R_f^M)^{-1}(A_1 \times \cdots \times A_n) . \end{aligned} \tag{21}$$

The family of subsets of Q can be endowed with a termset algebra structure by giving the Boolean operators their usual set-theoretic interpretations and interpreting f as f^M. One can show that this gives a termset algebra. Such an algebra, or a subalgebra of such an algebra, is called a *set-theoretic termset algebra.*

4.4 Topological Term Automata

Let Σ have the discrete topology. A *topological term automaton* is a term automaton

$$M = (Q,\ \Sigma,\ \ell,\ \delta)$$

endowed with a topology on Q such that the functions ℓ and $\lambda q.\delta(q, i)$ are continuous; equivalently, such that the partial maps R_f^M defined in (20) are continuous in the product topology on Q^n.

The topological term automaton M gives rise to a particular set-theoretic termset algebra $\mathbf{Cl}\, M$ whose elements are the clopen (closed and open) subsets of Q. These sets form a Boolean algebra, and the continuity of ℓ and δ insure that the operations $f^{\mathbf{Cl}\, M}$ defined by (21) preserve clopen sets.

Morphisms of topological term automata are continuous maps $h : M_1 \rightarrow M_2$ preserving ℓ and δ in the sense that

$$\begin{aligned} \ell_1(q) &= \ell_2(h(q)) \\ h(\delta_1(q, i)) &= \delta_2(h(q), i)\ . \end{aligned}$$

5 Stone Representation and Duality

5.1 A Representation Theorem

The following representation theorem says that every termset algebra is represented by a topological term automaton. The proof is a standard ultrafilter construction (see [18, 17]).

Theorem 4. *Every termset algebra is isomorphic to a set-theoretic termset algebra.*

Proof. Given a termset algebra $\mathcal{A}$, define the term automaton

$$\mathbf{St}\, \mathcal{A} = (U,\ \Sigma,\ \ell,\ \delta)$$

where U is the set of ultrafilters of $\mathcal{A}$, $\ell(u)$ is the unique $f \in \Sigma$ such that $f^{\mathcal{A}}(1, \ldots, 1) \in u$, and

$$\delta(u, i) = \{x \in \mathcal{A} \mid f^{\mathcal{A}}(\underbrace{1, \ldots, 1}_{i-1}, x, \underbrace{1, \ldots, 1}_{n-i}) \in u\}\ .$$

One must show that the set $\delta(u,i)$ is an ultrafilter; this follows from the elementary properties of termset algebras. Now for $x \in \mathcal{A}$, define

$$h(x) = \{u \in U \mid x \in u\}$$

and let the sets $h(x)$ generate a topology on $\mathbf{St}\,\mathcal{A}$. Then $\mathbf{Cl}\,\mathbf{St}\,\mathcal{A}$ forms a set-theoretic termset algebra as described in §4.4 and $h : \mathcal{A} \to \mathbf{Cl}\,\mathbf{St}\,\mathcal{A}$ is a termset algebra isomorphism.

We argue explicitly that h is a homomorphism with respect to $f \in \Sigma$:

$$\begin{aligned}
&f^{\mathbf{Cl}\,\mathbf{St}\,\mathcal{A}}(h(x_1),\ldots,h(x_n))\\
&= \{u \mid \ell(u) = f \text{ and } \delta(u,i) \in h(x_i),\ 1 \le i \le n\}\\
&= \{u \mid f^{\mathcal{A}}(1,\ldots,1) \in u \text{ and } x_i \in \delta(u,i),\ 1 \le i \le n\}\\
&= \{u \mid f^{\mathcal{A}}(1,\ldots,1) \in u \text{ and } f^{\mathcal{A}}(\underbrace{1,\ldots,1}_{i-1},x_i,\underbrace{1,\ldots,1}_{n-i}) \in u,\ 1 \le i \le n\}\\
&= \{u \mid f^{\mathcal{A}}(x_1,\ldots,x_n) \in u\}\\
&= h(f^{\mathcal{A}}(x_1,\ldots,x_n))\ .
\end{aligned}$$

That h is one-to-one follows from the fact that every filter extends to an ultrafilter.

5.2 Duality

Let $\mathcal{A}$ be a termset algebra and let $\mathbf{St}\,\mathcal{A}$ be its associated topological term automaton as constructed in §5.1. As a topological space, $\mathbf{St}\,\mathcal{A}$ is compact, Hausdorff, and has a base of clopen sets (namely $\{h(x) \mid x \in \mathcal{A}\}$). A topological term automaton with these properties is called *Stone*.

Let **STA** denote the category of Stone automata and continuous maps preserving ℓ and δ, and let **TSA** denote the category of termset algebras and termset algebra homomorphisms. Defining $\mathbf{St}\,h = h^{-1}$ for a termset algebra homomorphism h, the construction **St** becomes a contravariant functor **TSA** $\to$ **STA**. Similarly, defining $\mathbf{Cl}\,g = g^{-1}$ for a morphism $g : M \to N$ of topological term automata, the construction **Cl** becomes a contravariant functor **STA** $\to$ **TSA**. Moreover, these functors are bijections on the homsets of **TSA** and **STA**, and are inverses up to isomorphism, thus constitute a Stone duality (see [18]).

6 Completeness

6.1 Annotated Term Automata

Definition 5. Let $X = \{P, Q, \ldots\}$ be a set of propositional letters. An ***annotated term automaton*** is a tuple

$$M = (Q,\ \Sigma,\ \ell,\ \delta,\ X,\ \rho)$$

where $(Q,\ \Sigma,\ \ell,\ \delta)$ is a term automaton and $\rho : X \to 2^Q$. We topologize M by taking the weakest topology such that all $\rho(P)$ are clopen and ℓ and δ are continuous.

Annotated term automata provide a Kripke frame semantics for the multidimensional propositional modal logic discussed in §3. Syntactically, we may view formulas φ as ground terms over the ranked alphabet $\Sigma \cup X$, where the propositional letters X have arity 0. From a technical standpoint, however, it will be more convenient to use a more abstract syntax in which formulas are elements of $\mathcal{F}_X$, the free termset algebra on generators X, and work implicitly modulo the axioms of termset algebra.

To define satisfaction over M, extend ρ uniquely by induction to a termset algebra homomorphism $\rho : \mathcal{F}_X \rightarrow \mathbf{Cl}\, M$. We write $M, q \models \varphi$ and say φ is *satisfied* at q in M if $q \in \rho(\varphi)$. We write $M \models \varphi$ and say that φ is *realized* in M if $\rho(\varphi) = Q$. Define Th M, the *theory of* M, to be the set of formulas realized by M. This is a dual ideal $\{\neg\varphi \mid \varphi \in \ker \rho\}$. Since ρ is onto, $\mathcal{F}_X/\mathrm{Th}\ M \cong \mathbf{Cl}\, M$.

The free termset algebra $\mathcal{F}_X$ on generators X naturally gives rise to an annotated term automaton $(\mathbf{St}\,\mathcal{F}_X,\ X,\ \rho)$, where $\rho(P) = \{u \mid P \in u\}$. For this structure, $\rho(\varphi) = \{u \mid \varphi \in u\}$, thus ρ is just the Stone isomorphism $\rho : \mathcal{F}_X \rightarrow \mathbf{Cl}\,\mathbf{St}\,\mathcal{F}_X$.

6.2 Completeness

We consider a proof system consisting of the axioms of termset algebra and the usual rules of modal logic, namely modus ponens and modal generalization. The latter rule takes the form

$$\frac{\varphi}{\neg f(\ldots,\neg\varphi,\ldots)} \tag{22}$$

in this context. A set of formulas is *consistent* if its deductive closure does not contain 0. The following lemma and theorem establish deductive completeness of this system over annotated term automata.

Lemma 6. *A set of formulas is consistent and deductively closed if and only if it is a dual ideal in* $\mathcal{F}_X$.

In the proof of this lemma, the rule of modal generalization (22) corresponds directly to the property §2.4(12) of termset algebra ideals.

If Φ is consistent, let $\mathcal{F}_X/\Phi$ denote the quotient of $\mathcal{F}_X$ modulo the deductive closure of Φ, which is a dual ideal by Lemma 6.

Theorem 7. *A set of formulas is consistent if and only if it is realizable.*

Proof. Let Φ be a given consistent set of formulas. The Stone dual $\mathbf{St}\,\mathcal{F}_X/\Phi$ of the quotient $\mathcal{F}_X/\Phi$ with annotations

$$\rho(P) = \{u \mid P \in u\}$$

is the desired annotated term automaton realizing Φ. Conversely, any Th M is a dual ideal and thus consistent by Lemma 6.

7 Solutions of Set Constraints

In [19, 5, 1, 2, 7, 9, 14, 15, 30], various techniques for the solution of systems of set constraints are given. Here we introduce yet another approach.

Let Φ be a set of formulas φ (representing the set constraints $\varphi = 1$ in the sense of [1]) and let X be the set of propositional letters occurring in Φ. A *standard solution* of the constraints Φ is an annotation of the finite syntactic term automaton T_Σ defined in §4.2 such that the resulting annotated term automaton realizes Φ.

Definition 8. A term automaton $M = (Q,\ \Sigma,\ \ell,\ \delta)$ is said to be *closed* if for all $q_1, \ldots, q_n \in Q$ and $f \in \Sigma$ of arity n, $f^M(\{q_1\}, \ldots, \{q_n\})$ is nonempty; *i.e.*, there exists $q \in Q$ such that $\ell(q) = f$ and $\delta(q, i) = q_i$, $1 \le i \le n$.

This notion of closure corresponds directly to the notion of closure in hypergraphs defined in [1]. We can characterize this property algebraically:

Definition 9. A termset algebra is *closed* provided it satisfies the property

$$f(x_1, \ldots, x_n) = 0 \Rightarrow \bigvee_{i=1}^{n} (x_i = 0)\ . \tag{23}$$

A dual ideal Φ is *closed* provided

$$\neg f(x_1, \ldots, x_n) \in \Phi \Rightarrow \bigvee_{i=1}^{n} (\neg x_i \in \Phi)\ . \tag{24}$$

Thus the quotient $\mathcal{A}/\Phi$ is closed iff the dual ideal Φ is. The following lemma says that the algebraic and combinatorial notions of closure defined above coincide under Stone duality.

Lemma 10. *A termset algebra $\mathcal{A}$ is closed in the sense of Definition 9 iff its Stone dual* $\mathbf{St}\ \mathcal{A}$ *is closed in the sense of Definition 8. For any closed topological term automaton M, Stone or not,* $\mathbf{Cl}\ M$ *is closed.*

Theorem 11. *Let Φ be a set of formulas. The following statements are equivalent:*

(i) Φ has a standard solution
(ii) Φ has a consistent extension $\widehat{\Phi}$ satisfying (24)
(iii) $\mathbf{St}\ \mathcal{F}_X/\Phi$ *has a closed subautomaton.*

Proof. (i) $\rightarrow$ (ii): Let ρ be an annotation of T_Σ realizing Φ, and let M be the resulting annotated term automaton. Then Th M is a dual ideal extending Φ, and satisfies (24) by Lemma 10.

(ii) $\rightarrow$ (iii): By Stone duality, $\mathbf{St}\ \mathcal{F}_X/\widehat{\Phi}$ embeds into $\mathbf{St}\ \mathcal{F}_X/\Phi$, and by Lemma 10, $\mathbf{St}\ \mathcal{F}_X/\widehat{\Phi}$ is closed.

(iii) $\to$ (i): Let M be a minimal closed subautomaton of $\mathbf{St}\,\mathcal{F}_X/\Phi$. One can construct by induction a map $\sigma : T_\Sigma \to M$ such that

$$\delta(\sigma(f(t_1,\ldots,t_n)),i) = \sigma(t_i),\ 1 \le i \le n\ .$$

The image of σ is a closed subautomaton of M, and since M is minimal, σ is onto. Also, σ is one-to-one, since $\sigma(t)$ satisfies the formula t, and these formulas are pairwise inconsistent. The annotation of T_Σ realizing Φ is inherited from M under the bijection σ.

7.1 Filtration

In [1], an exponential-size hypergraph H for a given finite collection of set constraints Φ was constructed by *ad hoc* means. The hypergraph H is described by Boolean formulas obtained from Φ. It was proved that Φ has a solution iff H has a closed induced subhypergraph [1, Theorem 5.1].

The graph H is essentially a filtrate of $\mathcal{F}_X/\Phi$ and can be obtained by a standard filtration construction of modal logic (see [10, 17, 26]), which we outline here.

Determinacy is not preserved under the filtration construction. Thus we must work with nondeterministic termset algebras as defined in §2.2.

Definition 12. A *nondeterministic term automaton* is a frame

$$N = (Q,\ \Sigma,\ \ell,\ R)$$

such that for each $f \in \Sigma$, R gives a total map $R_f : Q \to 2^{Q^n}$ (instead of a partial map $Q \to Q^n$ as with deterministic term automata). Analogous to (21), for $A_1,\ldots,A_n \subseteq Q$ we define

$$f^N(A_1,\ldots,A_n) = \{q \in Q \mid R_f(q) \cap A_1 \times \cdots \times A_n \ne \emptyset\}\ .$$

Lemma 13. *The frame N gives a set-theoretic nondeterministic termset algebra under the construction of §4.3.*

Given an annotation $\rho : X \to 2^Q$, we define satisfaction as in §6.1, except here we must consider formulas as elements of the free nondeterministic termset algebra instead of $\mathcal{F}_X$.

Let Φ be a set of formulas over atomic formulas X. Let $\overline{\Phi}$ be the smallest set of formulas containing Φ such that

- $f(1,\ldots,1) \in \overline{\Phi}$ for all $f \in \Sigma$
- if $f(\varphi_1,\ldots,\varphi_n) \in \overline{\Phi}$, then $f(\underbrace{1,\ldots,1}_{i-1},\neg\varphi_i,\underbrace{1,\ldots,1}_{n-i}) \in \overline{\Phi}$, $1 \le i \le n$
- any subformula of a formula in $\overline{\Phi}$ is in $\overline{\Phi}$.

Let $M = (Q^M,\ \Sigma,\ \ell^M,\ \delta^M,\ X,\ \rho^M)$ be an annotated term automaton. The standard filtration construction of modal logic applied to M yields an annotated nondeterministic term automaton $N = (Q^N,\ \Sigma,\ \ell^N,\ R^N,\ X,\ \rho^N)$, where

- $[q] = \{p \mid \text{for all } \varphi \in \overline{\Phi},\ p \in \rho^M(\varphi) \Longleftrightarrow q \in \rho^M(\varphi)\}$
- $Q^N = \{[q] \mid q \in Q^M\}$
- $\ell^N([q]) = \ell^M(q)$
- $R_f^N([p]) = \{([q_1], \ldots, [q_n]) \mid \exists q \in Q^M\ [q] = [p],\ (q_1, \ldots, q_n) \in R_f^M(q)\}$
- $\rho^N(P) = \{[q] \mid q \in \rho^M(P)\}$.

Lemma 14. *For $\varphi \in \overline{\Phi}$, $[q] \in \rho^N(\varphi)$ if and only if $q \in \rho^M(\varphi)$.*

Proof. Induction on the structure of φ (see [17, 10]).

The following theorem is a restatement of [1, Theorem 5.1].

Theorem 15. *A set Φ of formulas has a standard solution if and only if the filtrate N of $\mathbf{St}\ \mathcal{F}_X/\Phi$ by $\overline{\Phi}$ has a closed subautomaton.*

Proof. By Lemma 11, Φ has a standard solution iff $\mathbf{St}\ \mathcal{F}_X/\Phi$ has a closed subautomaton. The image of any closed subautomaton of $\mathbf{St}\ \mathcal{F}_X/\Phi$ under the filtration map $q \mapsto [q]$ is a closed subautomaton of N. Conversely, if N has a closed subautomaton, as in the proof of Theorem 11 we can define a map $\sigma : T_\Sigma \to N$ by induction such that

$$\sigma(f(t_1, \ldots, t_n)) \in f^N(\{\sigma(t_1)\}, \ldots, \{\sigma(t_n)\}) .$$

The annotation of T_Σ realizing Φ is inherited from N under σ. Let

$$\rho^{T_\Sigma}(P) = \{t \mid \sigma(t) \in \rho^N(P)\} .$$

One can argue inductively that for all $t \in T_\Sigma$ and for all $\varphi \in \overline{\Phi}$,

$$t \in \rho^{T_\Sigma}(\varphi) \Longleftrightarrow \sigma(t) \in \rho^N(\varphi) . \tag{25}$$

The argument is straightforward except for the case $\varphi = f(\varphi_1, \ldots, \varphi_n)$, which we argue explicitly. For the direction ($\Rightarrow$),

$$\begin{aligned}
& t \in \rho^{T_\Sigma}(f(\varphi_1, \ldots, \varphi_n)) \\
& \Longleftrightarrow t = f(t_1, \ldots, t_n) \wedge t_i \in \rho^{T_\Sigma}(\varphi_i),\ 1 \le i \le n \\
& \Longleftrightarrow t = f(t_1, \ldots, t_n) \wedge \sigma(t_i) \in \rho^N(\varphi_i),\ 1 \le i \le n \qquad \text{(induction hypothesis)} \\
& \Rightarrow\ \sigma(t) \in \rho^N(f(\varphi_1, \ldots, \varphi_n)) .
\end{aligned}$$

Conversely,

$$\begin{aligned}
& t \notin \rho^{T_\Sigma}(f(\varphi_1, \ldots, \varphi_n)) \\
& \Longleftrightarrow t \in \rho^{T_\Sigma}(\sum_{g \neq f} g(1, \ldots, 1) + \sum_{i=1}^{n} f(\underbrace{1, \ldots, 1}_{i-1}, \neg\varphi_i, \underbrace{1, \ldots, 1}_{n-i})) \\
& \Rightarrow\ \sigma(t) \in \rho^N(\sum_{g \neq f} g(1, \ldots, 1) + \sum_{i=1}^{n} f(\underbrace{1, \ldots, 1}_{i-1}, \neg\varphi_i, \underbrace{1, \ldots, 1}_{n-i}))
\end{aligned}$$

by the preceding argument. But then $\sigma(t) \notin \rho^N(f(\varphi_1, \ldots, \varphi_n))$, since

$$\rho^N(\sum_{g \neq f} g(1, \ldots, 1) + \sum_{i=1}^{n} f(\underbrace{1, \ldots, 1}_{i-1}, \neg\varphi_i, \underbrace{1, \ldots, 1}_{n-i})) \cap \rho^N(f(\varphi_1, \ldots, \varphi_n)) = \emptyset ,$$

because by Lemma 14 this set is the image under the filtration map $u \mapsto [u]$ of

$$\rho^{\mathbf{St}\,\mathcal{F}_X/\Phi}((\sum_{g \neq f} g(1, \ldots, 1) + \sum_{i=1}^{n} f(\underbrace{1, \ldots, 1}_{i-1}, \neg\varphi_i, \underbrace{1, \ldots, 1}_{n-i})) \cdot f(\varphi_1, \ldots, \varphi_n)) = \emptyset .$$

Since $\mathbf{St}\ \mathcal{F}_X/\Phi$ realizes Φ, so does N by Lemma 14, and so does T_Σ by (25).

References

1. A. Aiken, D. Kozen, M. Vardi, and E. Wimmers, *The complexity of set constraints.* This volume.
2. A. Aiken, D. Kozen, and E. Wimmers, *Decidability of systems of set constraints with negative constraints*, Tech. Rep. 93-1362, Computer Science Department, Cornell University, June 1993.
3. A. Aiken and B. Murphy, *Implementing regular tree expressions*, in Proc. 1991 Conf. Functional Programming Languages and Computer Architecture, August 1991, pp. 427–447.
4. ———, *Static type inference in a dynamically typed language*, in Proc. 18th Symp. Principles of Programming Languages, ACM, January 1991, pp. 279–290.
5. A. Aiken and E. Wimmers, *Solving systems of set constraints*, in Proc. 7th Symp. Logic in Computer Science, IEEE, June 1992, pp. 329–340.
6. R. M. Amadio and L. Cardelli, *Subtyping recursive types*, in Proc. 18th Symp. Princip. Programming Lang., ACM, January 1991, pp. 104–118.
7. L. Bachmair, H. Ganzinger, and U. Waldmann, *Set constraints are the monadic class*, in Proc. 8th Symp. Logic in Computer Science, IEEE, June 1993, pp. 75–83.
8. L. Cardelli and P. Wegner, *On understanding types, data abstraction, and polymorphism*, Computing Surveys, 17:4 (1985), pp. 471–522.
9. W. Charatonik and L. Pacholski, *Negative set constraints with equality*, in Proc. 9th Symp. Logic in Computer Science, IEEE, July 1994. To appear. Also, Max-Planck-Institut für Informatik Technical Report MPI-I-93-265.
10. B. F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, 1980.
11. A. Colmerauer, *PROLOG and infinite trees*, in Logic Programming, S.-A. Tärnlund and K. L. Clark, eds., Academic Press, January 1982, pp. 231–251.
12. B. Courcelle, *Fundamental properties of infinite trees*, Theor. Comput. Sci., 25 (1983), pp. 95–169.
13. B. Dreben and W. D. Goldfarb, *The Decision Problem: Solvable Classes of Quantificational formulas*, Addison Wesley, 1979.
14. R. Gilleron, S. Tison, and M. Tommasi, *Solving systems of set constraints using tree automata*, in Proc. Symp. Theor. Aspects of Comput. Sci., vol. 665, Springer-Verlag Lect. Notes in Comput. Sci., February 1993, pp. 505–514.
15. ———, *Solving systems of set constraints with negated subset relationships*, in Proc. 34th Symp. Foundations of Comput. Sci., IEEE, November 1993, pp. 372–380.

16. R. GOLDBLATT, *Varieties of complex algebras*, Annals of Pure and Applied Logic, 44 (1989), pp. 173–242.
17. ———, *Mathematics of Modality*, vol. 43 of CSLI Lecture Notes, Center for the Study of Language and Information, 1993.
18. P. R. HALMOS, *Lectures on Boolean algebras*, Springer-Verlag, 1974.
19. N. HEINTZE AND J. JAFFAR, *A decision procedure for a class of set constraints*, in Proc. 5th Symp. Logic in Computer Science, IEEE, June 1990, pp. 42–51.
20. ———, *A finite presentation theorem for approximating logic programs*, in Proc. 17th Symp. Principles of Programming Languages, ACM, January 1990, pp. 197–209.
21. N. D. JONES AND S. S. MUCHNICK, *Flow analysis and optimization of LISP-like structures*, in Proc. 6th Symp. Principles of Programming Languages, ACM, January 1979, pp. 244–256.
22. B. JÓNSSON AND A. TARSKI, *Boolean algebras with operators*, Amer. J. Math., 73 (1951), pp. 891–939.
23. ———, *Boolean algebras with operators*, Amer. J. Math., 74 (1952), pp. 127–162.
24. D. KOZEN, J. PALSBERG, AND M. I. SCHWARTZBACH, *Efficient inference of partial types*, in Proc. 33rd Symp. Found. Comput. Sci., IEEE, October 1992, pp. 363–371.
25. ———, *Efficient recursive subtyping*, in Proc. 20th Symp. Princip. Programming Lang., ACM, January 1993, pp. 419–428.
26. D. KOZEN AND J. TIURYN, *Logics of programs*, in Handbook of Theoretical Computer Science, van Leeuwen, ed., vol. B, North Holland, 1990, pp. 789–840.
27. P. MISHRA, *Towards a theory of types in PROLOG*, in Proc. 1st Symp. Logic Programming, IEEE, 1984, pp. 289–298.
28. P. MISHRA AND U. REDDY, *Declaration-free type checking*, in Proc. 12th Symp. Principles of Programming Languages, ACM, 1985, pp. 7–21.
29. J. C. REYNOLDS, *Automatic computation of data set definitions*, in Information Processing 68, North-Holland, 1969, pp. 456–461.
30. K. STEFÁNSSON, *Systems of set constraints with negative constraints are NEXPTIME-complete*, in Proc. 9th Symp. Logic in Computer Science, IEEE, June 1994. To appear. Also Cornell University TR93-1380, August 1993.
31. Y. VENEMA, *Many-Dimensional Modal Logic*, PhD thesis, Universiteit van Amsterdam, January 1992.
32. J. YOUNG AND P. O'KEEFE, *Experience with a type evaluator*, in Partial Evaluation and Mixed Computation, D. Bjørner, A. P. Ershov, and N. D. Jones, eds., North-Holland, 1988, pp. 573–581.

This article was processed using the LaTeX macro package with LLNCS style