ELSEVIER

Theoretical Computer Science 212 (1999) 3–27

Theoretical Computer Science

Some complexity bounds for subtype inequalities [1]

Marcin Benke *

Institute of Informatics, Warsaw University, ul. Banacha 2, 02-097 Warsaw, Poland

Abstract

We study complexity of type reconstruction with subtypes. As proved recently, this problem is polynomially equivalent to checking satisfiability of systems of inequalities. Therefore we concentrate on the latter problem and prove that for TC-feasible posets it is in P. Further we propose alternation as a framework suitable for presenting and explaining the aforementioned complexity for various classes of underlying subtype relation.

Keywords: Type reconstruction; Subtyping; Complexity

0. Introduction

This article discusses various aspects of a single decision problem: satisfiability of subtype inequalities (abbreviated SSI). This problem, by itself interesting and presenting numerous challenges, is very closely related to many type reconstruction problems. The reader is referred to John Mitchell's papers [13, 14] for introduction to that area as well as the basic reduction of the original problem of type reconstruction to the problem of poset satisfiability.

Recent results of Hoang and Mitchell [10] show that the problem of Type Reconstruction with Subtyping (TRS) is polynomial-time equivalent to the problem of Satisfiability of Subtype inequalities (SSI). So now the latter problem, as the only known algebraic equivalent of the former, gains importance in the study of foundations of programming languages involving subtyping.

In connection with SSI problem, its special case called FLAT-SSI was considered by many authors [3, 12, 17, 23, 25]. The latter is equivalent to the retractability problem, known from the theory of partial orders [7, 15]. The purpose of the research was to provide some kind of 'taxonomy' amongst posets, having in mind the complexity of

* E-mail: marcin.benke@mimuw.edu.pl.

[1] This work has been partially supported by Polish KBN grants 2 P301 031 06 and 8 T11C 034 10 as well as ESPRIT BRA "Gentzen".

0304-3975/99/$ – see front matter
PII: S0304-3975(98)00134-0

satisfiability-checking. Even though a lot of research have been going on in the area of retractability [18–22] and structure theory of partial orders in general [6–8, 11], there is so far no such classification. The problem of FLAT-SSI attracted research interests mainly as an 'attack route' towards the general SSI problem, and thus towards the problem of type reconstruction with subtyping. The aim of this paper is to establish further links between SSI and FLAT-SSI. Sections 2 and 3 show that for posets for which feasibility of FLAT-SSI is witnessed by formulae of transitive closure logic, SSI is feasible too. Section 4 shows that for posets for which FLAT-SSI is NP-complete (wrt some class of reductions), SSI is PSPACE complete. It also proposes alternation as the framework within which relations between complexity of FLAT-SSI and SSI can be explained.

1. Preliminaries

Assuming we have already defined a subtype ordering, the simplest way to extend simple-typed lambda-calculus with subtyping is just to add the subsumption rule to the original system:

$$E \cup \{x : \tau\} \vdash x : \tau$$

$$\frac{E \vdash M : \tau \to \rho \quad E \vdash N : \tau}{E \vdash (MN) : \rho}$$

$$\frac{E \cup \{x : \tau\} \vdash M : \rho}{A \vdash (\lambda x.M) : \tau \to \rho}$$

$$\frac{E \vdash M : \tau \vdash \tau \leqslant \rho}{E \vdash M : \rho}$$

John C. Mitchell in his seminal paper [13] presented a reduction of typability in this system to SSI, thus showing its decidability (it is easy to construct a naive algorithm solving SSI in nondeterministic exponential time). Later, Hoang and Mitchell [10] showed that typability in this system is equivalent to SSI. The remainder of this section presents the latter problem as well as notions and problems pertinent to it.

1.1. Subtype inequalities

Let Q be a finite poset. The elements of Q are constant symbols of the signature which in addition contains a binary operation symbol $\to$. Let $\mathscr{T}_Q$ be the term algebra over this signature. The carrier of $\mathscr{T}_Q$ is partially ordered by extending the order from Q to all terms by the rule

$$\frac{r_1 \leqslant t_1 \quad t_2 \leqslant r_2}{(t_1 \to t_2) \leqslant (r_1 \to r_2)}$$

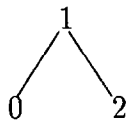

Fig. 1(a). Poset Q_1.

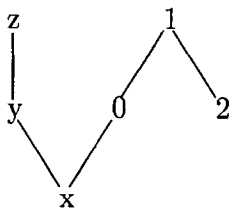

Fig. 1(b). Poset R.

A *system* Σ of *inequalities* is a finite set of formulas of the form

$$\Sigma = \{\tau_1 \leqslant \rho_1, \ldots, \tau_n \leqslant \rho_n\},$$

where τ's and ρ's are terms over the above signature with variables from a set V. Σ is said to be *flat* if every term in Σ is of size 1, i.e. it is either a constant symbol or a variable. Σ is said to be *satisfiable* in $\mathcal{T}_Q$ if there is a valuation $v : V \to \mathcal{T}_Q$ such that $\tau_i[v] \leqslant \rho_i[v]$ holds in $\mathcal{T}_Q$ for all i.

Satisfiability of Subtype Inequalities (SSI) is the following problem: given a system of inequalities Σ, decide whether it is satisfiable (the poset Q is considered to be fixed, rather then a part of the problem).

Similarly, FLAT-SSI is the problem of deciding whether given flat system of inequalities is decidable.

For example take $Q = Q_1$ as in Fig. 1(a), and consider the inequalities $x \leqslant 0$, $x \leqslant y$, $y \leqslant z$. Assigning either 1 or 2 to x would falsify $x \leqslant 0$, whence $x = 0$ in any satisfying assignment. Since $0 \leqslant 2$ does not hold in Q, y and z must each be either 0 or 1. Of these four possibilities, $y \leqslant z$ rules out $y = 1, z = 0$, and the remaining three assignments are all satisfying assignments. Hence, this set of inequalities is Q-satisfiable, in three ways.

1.2. Retractions and obstacles

Let Q and R be posets. We say that R extends Q if Q is a subposet of R. We say that R retracts to Q ($R \rhd Q$) if there exists an order preserving and idempotent (i.e. such that $f \circ f = f$) map $f : R \to Q$.

The problem of *Q-retractability* is defined as follows: given $R \supseteq Q$, does R retract to Q?

The example above of Q-satisfiability has an evident reformulation as a Q-retractability problem. We extend Q_1 to R by adjoining to Q_1 the variables x, y, z treated as new points, ordered as in the inequalities, as shown in Fig. 1(b).

The following theorem, due to Pratt and Tiuryn, relates retractability and satisfiability:

Theorem 1 (Pratt and Tiuryn [12]). *Q-FLAT-SSI[2] is polynomial-time equivalent to the P-retractibility problem.*

Proof. To reduce the Q-retractibility problem to Q-FLAT-SSI, translate the given extension R of Q to a set of inequalities by taking the set of variables to be $R - Q$ and taking the set of inequalities to be the graph of R, i.e. all $q \leqslant q'$ holding in R. Then R retracts to Q if and only if the set of inequalities is simultaneously satisfiable in Q.

To reduce Q-FLAT-SSI to Q-retractibility, translate the given set of inequalities to an extension R of Q whose non-Q elements are the variables appearing in the inequalities, ordered according to the reflexive transitive closure of the given inequalities. R is a preordered set: reflexive and transitive but not necessarily antisymmetric. Identify all equivalent elements, those pairs x, y such that $x \leqslant y \leqslant x$. (This extension might not be *conservative*, in the sense that for some $p \neq q \in Q$, $p \leqslant q$ might hold in R but not in Q, in which case R cannot retract to Q.) The given inequalities are then satisfiable in Q if and only if R retracts to Q.

In the same paper, Pratt and Tiuryn introduce the notion of an *obstacle* to retractability – a property of a larger poset which prevents it from retracting onto another one. An obstacle is called complete for Q if R retracts to Q whenever R does not satisfy it. For example, let Q extend P_1 (depicted on the Fig. 1a). Q retracts to P_1 iff $\{0,2\}$ has no lower bound in Q. This obstacle can be expressed with the formula

$$\exists x.(x \leqslant 0 \wedge x \leqslant 2).$$

In the mentioned paper, they discuss a class of posets (which they call TC-feasible) for which complete obstacles can be expressed by formulae of logic with a transitive closure operator. In the first part of this article we show that for such posets SSI can be decided in polynomial time.

This is a generalization of [3], where we discussed other kind of obstacles, complete for the class of Helly posets, which we consider useful in connection with inheritance.[3]

The notion of retraction and the retractability problem can be generalized to the case when R is a preorder in an obvious manner. As a flat system of inequalities can be naturally viewed as a preorder (modulo transitive closure, that is), we find the preorder formulation more convenient for our application. The obstacles for preorder retraction are the same as for poset retraction.

1.3. Intractable posets

An n-crown is a poset with $2n$ elements $0, 1, \ldots, 2n-1$ ordered in such a way that $2i \leqslant (2i \pm 1) \bmod 2n$.

[2] This problem is actually called Q-SAT in [12] but we choose to stick to the notation from [23].

[3] The notion of a Helly poset is well known in order thoery. As we do not use it in this article, we shall omit its definition, which is quite technical and can be found e.g. in [15, 18].

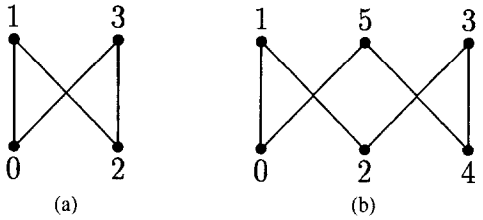

Fig. 2. (a) 2-crown; (b) 3-crown.

Pratt and Tiuryn [12] show that for n-crowns ($n \geqslant 2$), FLAT-SSI is NP-complete. Moreover, in [23] it is shown that for these posets SSI is PSPACE-hard. In Section 5 we show how this result can be generalized, proving that with some restriction on reductions, for every poset for which FLAT-SSI is NP-complete, the general problem of SSI is PSPACE-complete (We deal only with the "hardness" part as Frey has recently proved that SSI is in PSPACE [9]).

1.4. Shapes and weak satisfiability

The set $\mathscr{T}_\star$ of *shapes* is the set of terms without variables over the signature $\Sigma = \langle 0, \rightarrow \rangle$.

We shall use the canonical map $(\cdot)_\star : \mathscr{T}_Q(V) \rightarrow \mathscr{T}_\star(V)$

$$(c)_\star = 0 \text{ for } c \in Q, \qquad (v)_\star = v \text{ for } v \in V \qquad (t \rightarrow u)_\star = (t)_\star \rightarrow (u)_\star$$

and call $(t)_\star$ the shape of t if t is a term without variables.

Note that the subtype order on $\mathscr{T}_Q$ is stratified, i.e. only terms of the same shape are comparable. In the sequel we shall operate on strata of this ordering, defined as follows:

$$Q_0 = Q,$$

$$Q_{\sigma\rightarrow\tau} = \{t \rightarrow u : t \in Q_\sigma, u \in Q_\tau\}.$$

A system of inequalities $\Sigma = \{\tau_1 \leqslant \rho_1, \ldots, \tau_n \leqslant \rho_n\}$ is said to be *weakly satisfiable* if $\Sigma_\star = \{(\tau_1)_\star = (\rho_1)_\star, \ldots, (\tau_n)_\star = (\rho_n)_\star\}$ is satisfiable in $\mathscr{T}_\star$.

Weak satisfiability is clearly a necessary condition for satisfiability. It is decidable in (and in fact complete for) polynomial time since it is an instance of the unification problem [5, 23].

In the sequel, we shall deal only with weakly satisfiable sytems. In some places we shall assume (for the sake of proofs, not algorithms) that all inequalities of the system are annotated with proper shape and use the notation

$$t \leqslant_\sigma u$$

for an inequality in shape σ.

1.5. Complexity classes

In naming complexity classes we generally follow the conventions of [1, 2]. By $DTM(s,t)$ we understand a class of problems decidable by an s-space bounded and t-time bounded deterministic Turing machine. Expressions $NTM(s,t)$ and $ATM(s,t)$ denote corresponding nondeterministic and alternating classes.

Beside obvious complexity classes, we use the following abbreviations (ATIME and ASPACE denote alternating time and space respectively, cf. e.g. [2, 16]):

$$\text{NLOGSPACE} = \text{NSPACE}(\log n), \tag{1}$$

$$\text{ALOGSPACE} = \text{ASPACE}(\log n), \tag{2}$$

$$\text{AP} = \bigcup_{c>0} \text{ATIME}(n^c), \tag{3}$$

The correspondence between alternating and deterministic complexity classes is established by the following:

$$\text{ASPACE}(s(n)) = \bigcup_{c>0} \text{DTIME}(c^{s(n)}), \tag{4}$$

$$\text{ATIME}(t(n)) \subseteq \text{DSPACE}(t^2(n)); \tag{5}$$

in particular, we have

$$\text{ALOGSPACE} = \text{P}, \tag{6}$$

$$\text{AP} = \text{PSPACE}. \tag{7}$$

2. Transitive closure logic for subtype inequalities

In this section we introduce a variant of first order logic with transitive closure operator. Syntactically, the main difference from the logic proposed in [17] is that since the models we work with are stratified according to shapes, in our logic variables are annotated with shapes.

2.1. Syntax

Let Q be a finite poset, X a set of variables and $\sigma, \sigma_1, \sigma_2 \ldots$ be shapes. First we define the set of σ-shaped terms over Q with variables from a set X, as the smallest set $\mathcal{T}_Q^\sigma(X)$ satisfying the following conditions:

- if $x \in X$ then $x^\sigma \in \mathcal{T}_Q^\sigma(X)$,
- if $q \in Q$ then $q \in \mathcal{T}_Q^0(X)$,
- if $t_1 \in \mathcal{T}_Q^{\sigma_1}(X)$ and $t_2 \in \mathcal{T}_Q^{\sigma_2}(X)$ then $t_1 \to t_2 \in \mathcal{T}_Q^{\sigma_1 \to \sigma_2}(X)$.

Usually, we will assume that Q and X are fixed and use a shorthand $t : \sigma$ to mean that t is a term of shape σ.

The set of *annotated TC-formulae over Q* (or, short: ATC-formulae) is the least set ATC_Q such that

- Every atomic formula $t \leqslant_\sigma u$, where $t, u : \sigma$, is in ATC_Q.
- If φ and ψ are in ATC_Q, and every variable x free in φ and ψ has identical annotations in both formulae, then

  $$(\varphi \vee \psi),\ (\varphi \wedge \psi)$$

  are in ATC_Q.
- If φ is in ATC_Q, and every free occurrence of x is annotated by σ then

  $$(\exists x^\sigma.\varphi)$$

  is in ATC_Q.
- if φ is in ATC_Q, $\boldsymbol{\sigma} = \sigma_1, \ldots, \sigma_n$, then

  $$TC(\lambda \boldsymbol{x}^{\boldsymbol{\sigma}}, \boldsymbol{y}^{\boldsymbol{\sigma}}.\varphi)(\boldsymbol{t}, \boldsymbol{u})$$

  is in ATC_Q, where $\boldsymbol{x}, \boldsymbol{y}$ are n-vectors of individual variables, $\boldsymbol{t}, \boldsymbol{u}$ are n-vectors of terms such that $t_i, u_i : \sigma_i$.

We shall say that a formula is *flat* if it contains only 0-shaped terms and all its bound variables are annotated with 0. In such a case the annotations are of no consequence and we can safely omit them.

A formula will be called *balanced* if every inequality in it is in the same shape. From now on, we shall deal only with balanced formulae.

2.2. Free variables

Given an ATC-formula φ (or a term t), the set of its *free variables*, $FV(\varphi)$ is defined as usual. It should be stressed that λ in the TC operator is also a binder, so that

$$FV(TC(\lambda \boldsymbol{x}^{\boldsymbol{\sigma}}, \boldsymbol{y}^{\boldsymbol{\sigma}}.\varphi)(\boldsymbol{t}, \boldsymbol{u}) = (FV(\varphi) \setminus \{\boldsymbol{x}, \boldsymbol{y}\}) \cup FV(\boldsymbol{t}) \cup FV(\boldsymbol{u}).$$

2.3. Lonely variables

An occurrence of a variable shall be called *lonely* in φ, if it is free and not inside a term. Formally, given an ATC-formula φ (or a term t), we define the set of its *lonely variables*, $LV(\varphi)$ as follows:

$$\begin{aligned}
LV(x) &= \{x\},\\
LV(t \to u) &= \emptyset,\\
LV(t \leqslant u) &= LV(t) \cup LV(u),\\
LV(\varphi \wedge \psi) &= LV(\varphi) \cup LV(\psi),\\
LV(\varphi \vee \psi) &= LV(\varphi) \cup LV(\psi),\\
LV(\exists x.\varphi) &= LV(\varphi) \setminus \{x\},\\
LV(TC(\lambda \boldsymbol{x}^{\boldsymbol{\sigma}}, \boldsymbol{y}^{\boldsymbol{\sigma}}.\varphi)(\boldsymbol{t}, \boldsymbol{u}) &= (LV(\varphi) \setminus \{\boldsymbol{x}, \boldsymbol{y}\}) \cup LV(\boldsymbol{t}) \cup LV(\boldsymbol{u}).
\end{aligned}$$

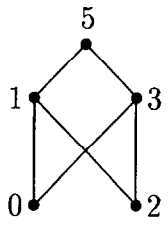

Fig. 3. A poset which is TC-feasible but not Helly.

For example, in the formula (skipping the annotations for brevity)

$$\exists x.x \leqslant (y \to z) \wedge t \leqslant x,$$

the variable t is lonely, while x, y, z are not (x is bound and y, z occur inside a term).

Note that for balanced formulae, all free occurrences of a lonely variable are lonely.

2.4. Semantics

First we define a semantics for flat formulae. A Q-model is any poset R of which Q is a subposet. A valuation v assigns to each variable x an element $v(x) \in R$. Now we can define when a flat formula is satisfied in R by a valuation v ($R \models \varphi[v]$):[4]

- $R \models (t_1 \leqslant_0 t_2)[v]$ iff $v(t_1) \leqslant_R v(t_2)$;
- $R \models (\exists x^0.\varphi)[v]$ iff $R \models (\varphi[r/x])[v]$ for some $r \in R$;
- $R \models TC(\lambda \boldsymbol{x}, \boldsymbol{y}.\varphi)(\boldsymbol{t}, \boldsymbol{u})[v]$ iff there exists a positive integer k and vectors of elements of R $\boldsymbol{t} = \boldsymbol{r}_0, \boldsymbol{r}_1, \ldots \boldsymbol{r}_k = \boldsymbol{u}$ such that $R \models (\varphi[\boldsymbol{r}_i/\boldsymbol{x}, \boldsymbol{r}_{i+1}/\boldsymbol{y}])[v]$ for $i = 0, 1, \ldots, k-1$.

For closed formulae, we shall omit the valuation and write simply $R \models \varphi$.

Flat formulae can be used to express obstacles for retractibility. For example, any poset R extending the poset depicted in Fig. 3, retracts to it if and only if there is no path in R which connects 0 and 2 and whose all points are bounded above by 1 and 3. This can be expressed in our logic as follows:

$$R \not\models TC(\lambda x, y.((x \leqslant y \vee y \leqslant x) \wedge x \leqslant 1 \wedge x \leqslant 3 \wedge y \leqslant 1 \wedge y \leqslant 3))(0, 2).$$

Definition 2. Poset Q is called TC-*feasible* if there exists a flat TC-formula φ_Q such that for every R extending Q,

$$R \rhd Q \Leftrightarrow R \not\models \varphi_Q.$$

Such φ_Q is called a *complete obstacle* for Q.

Theorem 3. *Every absolute retract (and hence every Helly poset) is TC-feasible.*

[4] For brevity, obvious clauses for conjunction and disjunction have been omitted in this definition and in the definition of the semantics of general formulae overleaf. See [15] and [3] for a discussion of absolute retracts and Helly posets.

Proof. If Q is an absolute retract then a complete obstacle for it is a formula stating that some hole in Q is not separated, i.e. a disjunction over all holes[5] in Q of formulae stating that a particular hole is not separated. If

$$\mathscr{H} = \{(v_t, \delta_t^{+1}, \delta_t^{-1}) : t \in T\}$$

is a hole in Q, the fact that it is not separated in Q can be expressed as follows:

$$\exists x. \bigwedge \{d^{\varepsilon}(v_t, x) \leqslant \delta^{\varepsilon} \mid t \in T,\ \varepsilon = \pm 1\}$$

Distance judgements are also easy to express in our logic, e.g. we can rewrite

$$d^{+1}(p,q) \leqslant n,$$

as

$$\exists x_1 \ldots \exists x_{n-1}.\ p \leqslant x_1 \wedge x_1 \geqslant x_2 \wedge \cdots \wedge x_{n-2} \leqslant x_{n-1} \wedge x_{n-1} \geqslant q.$$

From the above proof it follows that obstacles for absolute retracts can be expressed by existential formulae, i.e. without transitive closure operator. The poset depicted in Fig. 3 can also serve as an evidence that adding this operator really increases expressive power and that the class of TC-feasible posets is wider than Helly posets [17].

Before we present the semantics of arbitrary ATC-formulae, let us recall that variables are annotated with shapes which define how they should be valuated.

Let R be a poset, $v : X \to \mathscr{T}_R$. We say that v is compatible with φ if for every variable x free in φ, the shape of $v(x)$ corresponds to the annotation of x in φ. In what follows we shall consider only compatible valuations.

- $\mathscr{T}_R \models (t_1 \leqslant_\sigma t_2)[v]$ iff $v(t_1) \leqslant_{R_\sigma} v(t_2)$;
- $\mathscr{T}_R \models (\exists x^\sigma.\varphi)[v]$ iff $\mathscr{T}_R \models (\varphi[r/x])[v]$ for some $r \in R_\sigma$;
- $\mathscr{T}_R \models TC(\lambda \boldsymbol{x}^{\boldsymbol{\sigma}}, \boldsymbol{y}^{\boldsymbol{\sigma}}.\varphi)(\boldsymbol{t}, \boldsymbol{u})[v]$ iff there exists a positive integer k and vectors of elements from $\mathscr{T}_R$ $\boldsymbol{t} = \boldsymbol{r}^0, \boldsymbol{r}^1, \ldots \boldsymbol{r}^k = \boldsymbol{u}$ such that $r_i^j \in R_{\sigma_i}$ for all relevant i,j and $\mathscr{T}_R \models (\varphi[\boldsymbol{r}^j/\boldsymbol{x}, \boldsymbol{r}^{j+1}/\boldsymbol{y}])[v]$ for $j = 0, 1, \ldots, k-1$.

Proposition 4. *For flat φ we have*

$$\mathscr{T}_R \models \varphi \quad \textit{iff} \quad R \models \varphi.$$

Proof. For atomic φ, by definition we have

$$\mathscr{T}_R \models t \leqslant_0 u \quad \text{iff } t \leqslant_R u \quad \text{iff } R \models t \leqslant u.$$

Other cases follow by easy induction, basing on the fact that if $v : X \to \mathscr{T}_R$ is compatible with a flat formula φ then $v(x) \in R$ for every $x \in FV(\varphi)$. □

Definition 5. Let Q be a finite poset and Σ a system of inequalities over Q. We say that φ is a *semantical consequence* of Σ ($\Sigma \models \varphi$) if for every valuation v being a solution of Σ, $\mathscr{T}_Q \models \varphi[v]$.

[5] Since Q is finite, the set of holes is also finite.

2.5. Projections

First we define projections on shapes:

$$0 \downarrow i = 0, \qquad (\sigma_1 \to \sigma_2) \downarrow i = \sigma_i, \quad i = 1,2.$$

Next we define projections on terms:

$$c \downarrow i = c, \quad x^\sigma \downarrow i = x^{\sigma\downarrow i}, \ i = 1,2,$$

$$(t_1 \to t_2) \downarrow i = t_i, \ i = 1,2.$$

Now we define projections of ATC-formulae: $(\cdot)\downarrow 1, (\cdot)\downarrow 2 : ATC_Q \to ATC_Q$:

$$(t \leqslant_0 u) \downarrow i = t \leqslant_0 u,$$

$$(t \leqslant_{\sigma_1\to\sigma_2} u) \downarrow 1 = (u \downarrow 1) \leqslant_{\sigma_1} (t \downarrow 1),$$

$$(t \leqslant_{\sigma_1\to\sigma_2} u) \downarrow 2 = (t \downarrow 2) \leqslant_{\sigma_2} (u \downarrow 2),$$

$$(\varphi \wedge \psi) \downarrow i = (\varphi \downarrow i) \wedge (\psi \downarrow i),$$

$$(\varphi \vee \psi) \downarrow i = (\varphi \downarrow i) \vee (\psi \downarrow i),$$

$$(\exists x^\sigma.\varphi) \downarrow i = \begin{cases} \exists x^{\sigma\downarrow i}.\varphi \downarrow i & \text{if } x \in LV(\varphi), \\ \exists x^\sigma.\varphi \downarrow i & \text{otherwise,} \end{cases}$$

$$(TC(\lambda \boldsymbol{x}^\sigma, \boldsymbol{y}^\sigma.\varphi)(t,u)) \downarrow i = TC(\lambda x^{\sigma'}, y^{\sigma'}.(\varphi \downarrow i))(t \downarrow i, u \downarrow i),$$

where

$$\sigma'_j = \begin{cases} \sigma_j \downarrow i & \text{if } x_j \in LV(\varphi), \\ \sigma_j & \text{otherwise.} \end{cases}$$

For $\pi = p_1 p_2 \ldots p_n \in \{1,2\}^*$ we shall write $\varphi \downarrow \pi$ for $(\ldots((\varphi \downarrow p_1) \downarrow p_2)\ldots) \downarrow p_n$.

Lemma 6. *If φ is balanced, $LV(\varphi) = \emptyset$ and v is compatible with φ then $\mathscr{T}_R \models \varphi[v]$ iff $\mathscr{T}_R \vdash (\varphi \downarrow i)[v \downarrow i]$ for $i = 1,2$.*

Proof. by induction on φ. The case when $\varphi \equiv t \leqslant_\sigma u$ is trivial for $\sigma = 0$ and follows from definition of subtype order and projections for complex shapes (since $LV(\varphi) = \emptyset$, we have $t : t_1 \to t_2$ and $u : u_1 \to u_2$). The case when $\varphi \equiv \exists x^\sigma.\varphi_1$ is again trivial if $\sigma = 0$ and follows from the induction hypothesis if $x \notin LV(\varphi_1)$. Otherwise, if $\sigma = \sigma_1 \to \sigma_2$ then $\mathscr{T}_R \models \varphi[v]$ iff there exist $r_1 \in R_{\sigma_1}$, $r_2 \in R_{\sigma_2}$ such that $\mathscr{T}_R \models (\varphi_1[r_1 \to r_2/x])[v]$. Applying the induction hypothesis to the latter formula yields the thesis for this case. Conjunction and disjunction are obvious and TC can be handled similarly to $\exists$.

2.6. Closures

Let $t \preceq u$ denote the formula $TC(\lambda x^\sigma, y^\sigma . x \leqslant y)(t,u)$, The closure of a formula φ (denoted $\overline{\varphi}$) is defined as follows:

$$\begin{aligned}
\overline{t \leqslant u} &= t \preceq u \\
\overline{\varphi \wedge \psi} &= \overline{\varphi} \wedge \overline{\psi} \\
\overline{\varphi \vee \psi} &= \overline{\varphi} \vee \overline{\psi} \\
\overline{\exists x^\sigma . \varphi} &= \exists x^\sigma . \overline{\varphi} \\
\overline{TC(\lambda x^\sigma, y^\sigma . \varphi)(t,u)} &= TC(\lambda x^\sigma, y^\sigma . (\overline{\varphi})(t,u)
\end{aligned}$$

3. A proof system for ATC-formulae

As observed in [17], to decide satisfiability of a flat system Σ of inequalities over a poset Q for which we know a complete obstacle φ_Q, it is sufficient to check whether Σ satisfies an obstacle formula. This can be done in NLOGSPACE. Unfortunately, this result cannot be easily carried over to general systems, since there is no efficient (i.e. one that can be realized in polynomial time) method for checking satisfaction of a formula by a general system (as minimal solutions of such system can have exponential size).

In this section we introduce an inference system for ATC-formulae and prove its soundness. For the reasons outlined above, it is not complete. In fact it is designed so that for a fixed formula, we can check in polynomial time, whether it is derivable from a given system of inequalities. On the other hand we show the system is strong enough to be useful in deciding SSI for TC-feasible posets.

3.1. Inference rules

Let Σ be a weakly satisfiable system of inequalities over Q and let σ be the most general unifier of $\Sigma_\star$. We annotate every variable x occurring in Σ with the shape $\sigma(x)$. Consider the inference system depicted in Fig. 4.

Lemma 7. *For every ATC-formula φ, if $\Sigma \vdash \varphi$ then $\Sigma \models \varphi$.*

Proof. By induction on the derivation of φ. The cases when the last rule was axiom, alternative or conjunction are obvious. If the last rule used was $(\downarrow)$, then the thesis follows directly from Lemma 6.

Corollary 8. *If φ is a complete obstacle for Q and $\Sigma \vdash \varphi$ then Σ is not satisfiable.*

Theorem 9. *For any fixed, flat ATC-formula φ, one can check in time polynomial in $|\Sigma|$, whether $\Sigma \vdash \varphi$.*

$$\Sigma \vdash t \leqslant u \quad \text{for } t \leqslant u \in \Sigma$$

$$\frac{\Sigma \vdash \varphi}{\Sigma \vdash \varphi \downarrow i}(\downarrow) \qquad LV(\varphi) = \emptyset$$

$$\frac{\Sigma \vdash \varphi \quad \Sigma \vdash \psi}{\Sigma \vdash \varphi \wedge \psi}(\wedge)$$

$$\frac{\Sigma \vdash \varphi_i}{\Sigma \vdash \varphi_1 \vee \varphi_2}(\vee)$$

$$\frac{\Sigma \vdash \varphi[t/x^\tau]}{\Sigma \vdash \exists x^\tau.\varphi}(\exists) \quad t : \tau$$

$$\frac{\Sigma \vdash \varphi[\boldsymbol{t}/\boldsymbol{x}^\sigma, \boldsymbol{u}/\boldsymbol{y}^\sigma]}{\Sigma \vdash TC(\lambda \boldsymbol{x}^\sigma, \boldsymbol{y}^\sigma.\varphi)(\boldsymbol{t}, \boldsymbol{u})}(TC_0)$$

$$\frac{\Sigma \vdash TC(\lambda \boldsymbol{x}^\sigma, \boldsymbol{y}^\sigma.\varphi)(\boldsymbol{t}, \boldsymbol{s}) \quad \Sigma \vdash TC(\lambda \boldsymbol{x}^\sigma, \boldsymbol{y}^\sigma.\varphi)(\boldsymbol{s}, \boldsymbol{u})}{\Sigma \vdash TC(\lambda \boldsymbol{x}^\sigma, \boldsymbol{y}^\sigma.\varphi)(\boldsymbol{t}, \boldsymbol{u})}(TC_S)$$

Fig. 4. An inference system for ATC-formulae.

Proof. First, observe that, if we erase annotations, then the only formulae which may occur in a derivation of $\Sigma \vdash \varphi$ are subformulae of φ with free variables instantiated by subterms of terms occurring in Σ. Hence, the number of such formulae is polynomial in $|\Sigma|$. On the other hand, the number of distinct shapes that may occur in such derivation is bounded by the size of Σ. Thus the number of formulae that may occur in the derivation is polynomial and we may check systematically for each of them (proceeding from bigger to smaller terms and from simpler to more complicated formulae), whether it is derivable from Σ.

3.2. *Simple formulae and restricted derivations*

An annotated formula is called simple, if it contains no occurrences of TC or disjunction. Obviously no derivation of a simple formula can use rules for such constructs.

Lemma 10. *If* $\Sigma \vdash \exists x^\sigma.\varphi$, *then there exist*: *a shape* $\hat{\sigma}$, *a term* $t : \hat{\sigma}$, *a formula* $\hat{\varphi}$ *and a path* $\pi \in \{1,2\}^*$ *such that*

$$\sigma = \hat{\sigma} \downarrow \pi,$$

$$\varphi = \hat{\varphi} \downarrow \pi,$$

$$\Sigma \vdash \hat{\varphi}[\hat{t}/x^{\hat{\sigma}}].$$

Proof. This lemma is easily proved by induction on derivations.

Lemma 11. *The following rules are admissible:*

$$\frac{\Sigma \vdash \exists y.\exists x.\varphi}{\Sigma \vdash \exists x.\exists y.\varphi}$$

$$\frac{\Sigma \vdash \exists \boldsymbol{z}.\exists y.\exists x.\varphi}{\Sigma \vdash \exists \boldsymbol{z}.\exists x.\exists y.\varphi}$$

$$\frac{\Sigma \vdash \varphi[t/z] \quad \Sigma \vdash \exists \boldsymbol{y}.\psi[t/z]}{\Sigma \vdash \exists z.\exists \boldsymbol{y}.(\varphi \wedge \psi)} \quad \boldsymbol{y} \notin FV(\varphi)$$

$$\frac{\Sigma \vdash \exists \boldsymbol{x}.\varphi[t/z] \quad \Sigma \vdash \exists \boldsymbol{y}.\psi[t/z]}{\Sigma \vdash \exists z.\exists \boldsymbol{y}\exists \boldsymbol{x}.(\varphi \wedge \psi)} \quad \begin{array}{l} \boldsymbol{y} \notin FV(\varphi) \\ \boldsymbol{x} \notin FV(\psi) \end{array}$$

Proof. This lemma can be proved by induction on derivations, using the previous lemma. □

As we shall see in the next section, there is a close correspondence between simple formulae and systems of inequalities, which we shall use in the later proof. However, to deal with all formulae of our logic we introduce the notion of a *simplification set* of a formula φ – a possibly infinite set of simple formulae such that φ is derivable if and only if some formula from its simplification set is derivable.

Definition 12. The *simplification set* of formula φ is the set of simple formulae, defined as follows:

$$[t \leqslant u] = \{t \leqslant u\}$$

$$[\varphi_1 \vee \varphi_2] = [\varphi_1] \cup [\varphi_2]$$

$$[\varphi_1 \wedge \varphi_2] = \{\psi_1 \wedge \psi_2 : \psi_1 \in [\varphi_1], \psi_2 \in [\varphi_2]\}$$

$$[\exists x.\varphi] = \{\exists x.\psi : \psi \in [\varphi]\}$$

$$[TC(\lambda \boldsymbol{x}^{\sigma}, \boldsymbol{y}^{\sigma}.\varphi)(\boldsymbol{t}, \boldsymbol{u})] = \bigcup_{k \in \omega} \{\exists \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k.\psi[\boldsymbol{t}/\boldsymbol{y}, \boldsymbol{z}_1/\boldsymbol{y}] \wedge \psi[\boldsymbol{z}_1/\boldsymbol{x}, \boldsymbol{z}_2/\boldsymbol{y}] \wedge \ldots \wedge \psi[\boldsymbol{z}_k/\boldsymbol{y}, \boldsymbol{u}/\boldsymbol{y}] : \psi \in [\varphi]\}\}.$$

Lemma 13. *For every formula φ and $\psi \in [\varphi]$, we have* (i) $FV(\psi) \subseteq FV(\varphi)$ *and* (ii) $LV(\psi) \subseteq LV(\varphi)$.

Proof. (i) follows by an easy induction over φ; (ii) follows from (i) and the fact that simplification does not change terms occuring in the formula (though some may be omitted, but it can only decrease LV).

Lemma 14. *For every ATC-formula φ, $\Sigma \vdash \varphi$ iff there exists a simple formula ψ belonging to the simplification set of φ such that $\Sigma \vdash \psi$.*

Proof. This lemma is easily proved by structural induction over formulae, using Lemma 11.

3.3. Canonical form of simple formulae

By canonical form of a simple formula we mean its prenex form. A formula in this form may be treated as a system of inequalities. Namely, for the formula $\varphi \equiv \exists \boldsymbol{x}.(t_1 \leqslant u_1 \wedge \cdots \wedge t_n \leqslant u_n)$, the corresponding system of inequalities is $\Delta(\varphi) = \{t_1 \leqslant u_1, \ldots, t_n \leqslant u_n\}$.

If Γ and Δ are systems of inequalities, we say that $\Gamma \vdash \Delta$ if there is a formula φ such that $\Gamma \vdash \varphi$ and $\Delta = \Delta(\varphi)$.

3.4. Flat systems

Let Σ be a flat system of inequalities over Q. We shall write $Q \cup \Sigma$ as a shorthand for

$$\Sigma \cup \{t \leqslant u \mid Q \models t \leqslant u, \quad t, u \in Q\}.$$

Consider the set $Q_\Sigma = Q \cup var(\Sigma)$, preordered by the relation $\preceq$ defined as follows:

$$t \preceq u \quad \text{iff} \quad Q \cup \Sigma \vdash t \preceq u.$$

Lemma 15. *Σ is satisfiable iff $\langle Q_\Sigma, \preceq \rangle$ retracts to $\langle Q, \leqslant \rangle$.*

Proof. Let v be a solution of Σ. We will show that $v \cup id_Q$ is a retraction. The idempotence is obvious, so it only remains to prove monotonicity.

If $Q \cup \Sigma \vdash t \preceq u$ then there exist $t = r_0, r_1, \ldots, r_k = u \in var(\Sigma) \cup Q$ such that

$$Q \cup \Sigma \vdash r_i \leqslant r_{i+1}, \quad i = 0, 1, \ldots, k-1,$$

hence

$$Q \cup \Sigma \ni r_i \leqslant r_{i+1}, \quad i = 0, 1, \ldots, k-1.$$

Since v is a solution of Σ, we have

$$v(r_i) \leqslant_Q v(r_{i+1}), \quad i = 0, 1, \ldots, k-1;$$

thus

$$v(t) \leqslant_Q v(u).$$

It is easy to see that any retraction $v : Q_\Sigma \to Q$ is a solution of Σ.

For a given, finite Q we shall construct a formula $NGC(Q)$ such that $\Sigma \vdash NGC(Q)$ iff Σ is not ground consistent:

$$NGC(Q) \equiv \bigvee \{c \preceq d \mid Q \not\models c \leqslant d\}.$$

Lemma 16. *Let φ be a complete obstacle for Q. If $\Sigma \vdash NGC(Q)$ then Σ is not satisfiable. Otherwise Q_Σ retracts to Q iff $Q_\Sigma \not\models \varphi$.*

Proof. From Lemma 7 it follows that if $\Sigma \vdash NGC(Q)$ then every solution of Σ must satisfy $NGC(Q)$; this is possible only if Σ has no solutions. If $\Sigma \nvdash NGC(Q)$ then Q_Σ is an extension of Q and Q_Σ retracts to Q iff $Q_\Sigma \not\models \varphi$ since φ is a complete obstacle for Q.

Lemma 17. *For every ATC-formula ψ,*

$$Q_\Sigma \models \psi \Leftrightarrow Q \cup \Sigma \vdash \bar{\psi}$$

(where $\bar{\psi}$ denotes the closure of ψ defined in section 2.6*).*

Proof. Note that, by definition of ordering on Q_Σ, for every $t, u \in Q_\Sigma$

$$Q_\Sigma \models t \leqslant u \Leftrightarrow Q \cup \Sigma \vdash t \preceq u.$$

Now, the thesis follows by an easy induction on ψ.

Lemma 18. *Let φ be a complete obstacle for Q. For every flat system of inequalities Σ, it is satisfiable iff*

$$Q \cup \Sigma \nvdash \bar{\varphi} \vee NGC(Q).$$

Proof. This lemma is a simple consequence of previous three lemmas.

3.5. Single-shaped systems and formulae

A system of inequalities is called *σ-shaped* if all its variables and inequalities are of the shape σ. Similarly, a formula is called *σ-shaped* if it is balanced and all its variables (free and bound) as well as all inequalities are annotated with σ. A system (formula) is called *single-shaped* if it is σ-shaped for some σ.

These notions will serve us as an intermediate level between flat and general systems of inequalities and will facilitate the proof of the main theorem.

Definition 19. Let $\sigma = \sigma_1 \to \sigma_2$ and let $\Sigma = \{t_1 \leqslant u_1, \ldots, t_n \leqslant u_n\}$ be a σ-shaped system of inequalities. For $i = 1, 2$, we define

$$\Sigma \Downarrow i = \{(t_1 \leqslant u_1) \downarrow i, \ldots, (t_n \leqslant u_n) \downarrow i\}.$$

Lemma 20. *Let φ be a complete obstacle for Q and Σ be single-shaped. Σ is satisfiable iff*

$$Q \cup \Sigma \nvdash \varphi \vee NGC(Q).$$

Proof. Before we derive into technicalities, let us explain the intuition behind this lemma. A σ-shaped system of inequalities over Q may be viewed as a flat system over

Q_σ, which is again a TC-feasible poset. Even though φ is not a complete obstacle for Q_σ, one can easily construct a formula φ^σ which is such an obstacle, and having the property that $\Sigma \vdash \varphi$ iff $\Sigma \vdash \varphi^\sigma$. But as this line of proof is technically more complicated and needs several technical lemmas similar to 11, we shall prove this lemma in a slightly different way.

If $\Sigma \vdash \varphi$ then obviously Σ is not satisfiable (see Corollary 8). Thus it remains to prove that if Σ is not satisfiable then $\Sigma \vdash \varphi$. Let Σ be σ-shaped. We shall proceed by induction on σ. If Σ is flat then the thesis follows from the Lemma 18. On the other hand, if $\sigma = \sigma_1 \to \sigma_2$, Σ is unsatisfiable iff either $\Sigma \Downarrow 1$ or $\Sigma \Downarrow 2$ is.

Let us assume that $Q \not\models \Sigma \Downarrow 2$ (the other case is handled dually). By completeness of φ, we have that $\Sigma \Downarrow 2 \vdash \varphi$. We shall show that this implies $\Sigma \vdash \varphi$. First we shall prove that for every formula ψ derivable from $\Sigma \Downarrow 2$ without using $(\downarrow)$, there exists a formula $\psi \uparrow 2$ such that

(1) $\Sigma \vdash \psi \uparrow 2$,
(2) $(\psi \uparrow 2) \downarrow 2 = \psi$,
(3) $LV(\psi \uparrow 2) = \emptyset$ iff $LV(\psi) = \emptyset$.

- If $\psi \equiv t \leqslant u$ then $t \leqslant u \in \Sigma \Downarrow 2$, since the derivation does not contain $(\downarrow)$. Hence there is $t' \leqslant u'$ in Σ such that $(t' \leqslant u') \downarrow 2 = t \leqslant u$.
- If $\psi \equiv \exists x^\tau.\psi'$ and $\Sigma \Downarrow 2 \vdash \psi'[t/x]$ then $\Sigma \vdash (\psi'[t/x]) \uparrow 2$. Let ψ'' be such that

$$\psi''[t/x] = (\psi'[t/x]) \uparrow 2$$

and let

$$\psi \uparrow 2 = \begin{cases} \exists x^{\sigma_1 \to \sigma_2}.(\psi'') & \text{if } \tau = \sigma_2, \\ \exists x^\tau.\psi'' & \text{otherwise.} \end{cases}$$

The correctness of the above definition follows from the fact that, since $\Sigma \Downarrow 2$ is σ_2-shaped, $x \in LV(\psi')$ iff $\tau = \sigma_2$.

- The TC case is handled very similarly (cf. the definition of projection in Section 2.5.
- The cases of conjunction and disjunction are trivial.

If the derivation of the obstacle φ from $\Sigma \Downarrow 2$ does not contain $(\downarrow)$, the thesis follows immediately. Otherwise, consider a subderivation ending with an application of the rule $(\downarrow)$

$$\dfrac{\dfrac{\vdots}{\Sigma \Downarrow 2 \vdash \psi}}{\Sigma \Downarrow\ 2 \vdash \psi \downarrow i}$$

and such that it contains no other application of this rule. We have

(1) $\Sigma \vdash \psi \uparrow 2$,
(2) $(\psi \uparrow 2) \downarrow 2 = \psi$,
(3) $LV(\psi) = \emptyset$,
(4) $LV(\psi \uparrow 2) = \emptyset$.

Hence

$$\dfrac{\dfrac{\vdots}{\Sigma \vdash \psi \uparrow 2}}{\Sigma \vdash \psi \downarrow i}$$

Thus we have proved the desired thesis. □

Definition 21. A system $\hat{\Sigma}$ is called a σ-view of Σ if the following conditions hold:
(1) $\hat{\Sigma}$ is σ-shaped.
(2) $\Sigma \vdash \hat{\Sigma}$.
(3) For every σ-shaped Δ, if $\Sigma \vdash \Delta$ then $\hat{\Sigma} \vdash \Delta$.

Lemma 22. *For every system Σ and shape σ minimal in Σ there exists a σ-view of Σ.*

Proof. Let σ be a shape minimal in Σ, ρ be a shape of some inequality in Σ and π be a path such that $\rho \downarrow \pi = \sigma$. Further, let Σ_ρ denote the set of ρ-shaped inequalities in Σ and ψ_ρ^Σ be the formula

$$\psi_\rho^\Sigma = \exists \boldsymbol{x}. \bigwedge_{t \leqslant u \in \Sigma_\rho} t \leqslant u,$$

where the quantification is over all variables occurring in Σ_ρ which have shape different from σ. Obviously, the formula $(\psi_\rho^\Sigma \downarrow \pi)$ is σ-shaped, and because of minimality of σ it is derivable from Σ.

Now it is easily seen that

$$\Delta_\sigma^\Sigma = \Delta(\bigwedge\{(\psi_\rho^\Sigma \downarrow \pi) \mid \rho \downarrow \pi = \sigma\})$$

is a σ-view of Σ: the condition (1) is obvious, (2) follows from the derivability of $\psi_\rho^\Sigma \downarrow \pi$, and (3) can be proved by an easy induction on derivations.

3.6. General systems

Theorem 23. *Let φ be the complete obstacle for Q. For every system of inequalities Σ, Σ is satisfiable iff it is weakly satisfiable and*

$$Q \cup \Sigma \nvdash \varphi \vee NGC(Q).$$

Proof. The ($\Rightarrow$) implication is obvious. The opposite implication is proved by induction on the number of equivalence classes of $\sim$ defined on $var(\Sigma)$ as follows:

$$x \sim y \text{ iff } \Sigma_* \models x = y.$$

Suppose first that the quotient set $var(\Sigma)/_\sim$ has only one element σ. Then every inequality in Σ is either σ-shaped, or of the form $t_1 \to t_2 \leqslant u_1 \to u_2$, or $p \leqslant q$ with

$p, q \in Q$. Thus it is easy to construct (by decomposition of complex inequalities and removing inequalites between constants from Q) a system Σ' which is σ-shaped and equivalent (in the sense of satisfiability as well as derivability of flat formulae) to Σ. Satisfiability of Σ' (and hence Σ) follows from Lemma 20.

Now let us assume that $var(\Sigma)/_\sim$ has $n+1$ elements and let $y \in var(\Sigma)$ be such that for no $z \in var(\Sigma)$ there is a term τ with $|\tau| > 1, z \in var(\tau)$ and $\Sigma_* \models \tau = y$. If there is no such y, then one easily finds a term τ such that $|\tau| > 1, z \in var(\tau)$ and $\Sigma_* \models \tau = z$. This would contradict weak satisfiability of Σ.

Let

$$[y] = \{z \in var(\Sigma) | z \sim y\} = \{y_1, \ldots, y_k\}.$$

Let σ be the shape assigned to y by the most general unufier of $\Sigma_\star$, and let $\hat{\Sigma}$ be a σ-view of Σ.

Again, the satisfiability of $\hat{\Sigma}$ follows from the Lemma 20. Let $\hat{v} : [y] \to \mathscr{T}_\sigma$ be a solution of $\hat{\Sigma}$ and let

$$\Sigma_1 = \hat{v}(\Sigma) = \{\hat{v}(\tau) \leqslant \hat{v}(\rho) : \tau \leqslant \rho \in \Sigma\}.$$

In the above definition $\hat{v}$ acts as identity on variables other than those in $[y]$. Let $\sim_1$ be the equivalence relation associated with Σ_1. One can prove that

$$|var(\Sigma_1)/_{\sim_1}| = n.$$

Σ_1 is weakly satisfiable

To complete the proof we need to prove that

$$\Sigma_1 \not\vdash \varphi \vee NGC(Q).$$

To do this we shall prove that for every flat Δ derivable from Σ_1, $Q \models \Delta$. On the other hand we have that

$$Q \not\models \varphi \vee NGC(Q).$$

For every flat Δ derivable from Σ_1 there exists a σ-shaped Δ' derivable from Σ such that Δ is derivable from $\hat{v}(\Delta')$. Since $\hat{\Sigma}$ is a σ-view of Σ, we also have $\hat{\Sigma} \vdash \Delta'$. Thus $\hat{v}(\hat{\Sigma}) \vdash \hat{v}(\Delta')$. But since $\hat{v}$ is a solution of $\hat{\Sigma}$, we have $Q \models \Delta'$, which we wanted to prove.

Corollary 24. *For any TC-feasible Q and Σ – a system of inequalities over Q one can check in time polynomial in $|\Sigma|$, whether Σ is satisfiable.*

Proof. By Theorem 23 there is a flat ATC formula φ_Q depending only on Q and such that Σ is satisfiable iff φ is not derivable from Q, which by Theorem 9 can be checked in polynomial time.

4. Subtyping and alternation

The aim of this section is to establish further links between SSI and FLAT-SSI, providing some evidence in favor of the following conjecture:

Conjecture 25. *Given a poset Q such that Q-FLAT-SSI is complete for NTM(s,t), Q-SSI is complete for ATM(s,t).*

In our opinion, the 'nondeterminism vs. alternation' concept constitutes a framework within which various complexity phenomena bound with subtyping can be explained. Sure enough, there is still a lot of open questions and gaps to be filled, but we present it with hope that it will encourage further research in this area. One example would be the apparent 'gap' in the poset hierarchy. So far we know no posets for which SSI is NP-complete or FLAT-SSI – P-complete. Within our framework, the explanation for this gap is provided by the fact that (unless P=NP or NP=PSPACE) NP is not an alternating complexity class and (unless P=NL or P=NP), P is not a nondeterministic complexity class.

4.1. Motivating examples

First let us look at several examples known so far that supporting the thesis that arrows in the systems of inequalities correspond on the complexity level exactly to the transition from nondeterministic classes to corresponding alternating classes. This is at the same time a resume of current knowledge about the complexity of SSI:

(1) If Q is discrete, then
- Q-FLAT-SSI is in NLOGSPACE;[6]
- Q-SSI is equivalent to the unification, and hence AL-complete.

(2) If Q is a disjoint union of lattices (but not discrete), then
- Q-FLAT-SSI is NLOGSPACE-complete [4];
- Q-SSI is ALOGSPACE-complete [23].

(3) If Q is a non-discrete Helly poset, then
- Q-FLAT-SSI is NLOGSPACE-complete [3, 4];
- Q-SSI is ALOGSPACE-complete [3].

(4) If Q is a non-discrete TC-feasible poset, then
- Q-FLAT-SSI is NLOGSPACE-complete [17];
- Q-SSI is ALOGSPACE-complete (Corollary 24).

(5) If Q is an n-crown ($n > 1$), then
- Q-FLAT-SSI is NP-complete [17];
- Q-SSI is AP-complete [9, 23].

[6] The problem whether it is NLOGSPACE-hard is equivalent to a known open problem in complexity, whether SYMLOGSPACE=NLOGSPACE.

4.2. Encoding alternation

In this section we show that the result of [23] (AP-hardness of SSI for crowns) can be generalized stating that for all posets for which FLAT-SSI is NP-hard, SSI is AP-hard. To this end, we construct an encoding for QBF[7] as an SSI, given encoding of SAT[8] as FLAT-SSI.

We shall first recall the construction from [23] and explain its most important elements. Then we shall show how this idea can be generalized by abstracting out the essential conditions on the poset and state our main theorem in this section. Finally we go on with (admittedly intricate) details of the proof.

Tiuryn's encoding is based upon the encoding of SAT presented in [17], which for given boolean formula φ constructs a flat system of inequalities Σ^{φ} such that φ is satisfiable if and only if Σ^{φ} is. Its parts important for the encoding of QBF are: the encoding of truth values and the mechanism that simulates negation. On the other hand parts responsible for conjunction and disjunction are irrelevant here.

The crucial tool used in [17] was an extension of crown called double crown. For every propositional variable two copies of such double crown were used. The negation was simulated by "locking" together copies denoting *true* and *false*. What made such locking mechanism possible was an antimonotonic bijection of a crown onto itself.

The observation that enabled Tiuryn to extend the encoding of SAT to an encoding of QBF was that the locking mechanism described above, when combined with the arrow operator (which is antimonotonic in its first and monotonic in its second argument) can be used to express quantifiers.

Let us now summarize the elements that we shall use (albeit in a generalized form) in our encoding:

- a system of inequalities with distinguished corresponding to truth values of propositional variables occurring in the formula,
- an antimonotonous bijection (used to simulate negation).

Now let us formulate assumptions about encodings of instances of SAT as systems of inequalities. Intuitively, these assumptions express the requirement that whenever there exists a simulation of NTM, there exists one which is "regular" enough to be transformed to a simulation of an ATM. This intuition is formalized in the following.

Definition 26. Let $\varphi = \varphi(\boldsymbol{x})$ be a 3-CNF[9] propositional formula with variables $\boldsymbol{x} = x_1, \ldots x_n$ (and no other)

We say that a flat system of inequalities Σ_{φ} encodes φ if there exist variables $z_1, \ldots, z_n$ and constants $\boldsymbol{c}$ such that for every $p_1, \ldots, p_n \in \{0, 1\}$

$$\models \varphi[\boldsymbol{p}/\boldsymbol{x}] \Longleftrightarrow \Sigma_{\varphi}[\boldsymbol{c}/\boldsymbol{z}] \text{ is satisfiable}$$

[7] That is the problem of checking satisfiability of Quantified Boolean Formulae.

[8] Satisfiability of boolean formulae in conjunctive normal form.

[9] 3-CNF means a conjunctive normal form with 3 disjuncts in each clause.

We say the encoding is *symmetric*, if there exists an antimonotonic bijection $f : Q \to Q$ that extends to an antimonotonic and idempotent[10] bijection of (the poset corresponding to) Σ_φ onto itself and such that $c_i^1 = f(c_i^0)$ for $i = 1,\ldots,n$.

Theorem 27. *Let Q be a poset such that Q-FLAT-SSI is complete for NP under* symmetric *reductions. Then Q-SSI is complete for AP.*

Proof. Since [9] presents an AP-algorithm for deciding SSI for an arbitrary finite poset, we need to prove hardness only. Let

$$\forall x_n \exists y_n \ldots \forall x_1 \exists y_1 \ \varphi$$

be an instance of QBF, φ contains no quantifiers and is in 3-CNF.

Let Σ_φ be a symmetric encoding of φ. We show how to construct a system of inequalities Σ_k such that

$$\psi_k \text{ holds } \iff \Sigma_k \text{ is satisfiable,}$$

where

$$\psi_k = \exists x_n \exists y_n \ldots \exists x_{k+1} \exists y_{k+1} \forall x_k \exists y_k \ldots \forall x_1 \exists y_1 \ \varphi.$$

The construction of Σ_k is by induction on k, the number of quantifier alternations in ψ_k.

Let us recall that f is idempotent, i.e. $f \circ f = id$.

In what follows we use a with sub- or super-scripts. These are new variables. We will also use new variables $u_k^{i,j}$, where $0 \leqslant k \leqslant n$, $i,j \in Q$ and u is a propositional variable of φ. The variable $u_k^{i,j}$ is a version of $u^{i,j}$, lifted to level k. The variable a_k^i, which we use below, represents constant i lifted to level k.

Let us first define sets Δ_k, for $0 \leqslant k \leqslant n$:

$$\Delta_0 = \{ a_{0,0}^{i,j} = a_0^j \mid i,j \in P \} \cup \{ a_0^i = i \mid i \in P \}.$$

For $k < n$, Δ_{k+1} is Δ_k plus Eqns. (8)–(11) below, with i,j ranging over Q:

$$a_{k+1}^i = a_k^{f(i)} \to a_k^i. \tag{8}$$

For $k+1 < p \leqslant n$ and $z_p \in \{x_p, y_p\}$,

$$z_{p,k+1}^{i,j} = z_{p,k}^{f(j),f(i)} \to z_{p,k}^{i,j}. \tag{9}$$

[10] The idempotence assumption is introduced just to simplify the presentation; it could be dropped since Q is finite and for every antimonotonic bijection g there exists an integer k such that g^k is idempotent.

For $1 \leqslant p \leqslant k$,

$$a^{i,j}_{p,k+1} = a^{f(j),f(i)}_{p,k} \to a^{i,j}_{p,k}, \tag{10}$$

$$a^{i,j}_{k+1,k+1} = a^{j}_{k+1}. \tag{11}$$

For every $k \geqslant 0$, let $\hat{\Sigma}_k$ be the system of inequalities obtained from $\hat{\Sigma}$ by replacing every variable $[u]^{i,j}$ of $\hat{\Sigma}$ by $[u]^{i,j}_k$, and replacing the constant $i \in Q$ by a (new) variable a^i_k. Hence, there are no constants in $\hat{\Sigma}_k$.

Finally, we set $\Sigma_{k+1} = \Delta_{k+1} \cup \hat{\Sigma}_{k+1}$ plus Eq. (12) with i,j ranging over Q and $1 \leqslant p \leqslant k+1$:

$$z_{p,k+1} = a^{c^0_p, c^1_p}_{p,k+1}. \tag{12}$$

The thesis follows from the following lemmas:

Lemma 28. *Let* $V_k = \{x_{k+1}, y_{k+1}, \ldots, x_n, y_n\}$. *For all* $k \geqslant 0$, *and for every function* $\xi : V_k \to \{0,1\}$, $\Sigma_{k+1} \cup \{ z_k = a_k^{c_k^{\xi(v)}} \mid v \in V_{k+1} \}$ *is satisfiable iff for every* $i \in \{0,1\}$, $\Sigma_k \cup \{ z_k = a_k^{c_k^{\xi(v)}} \mid v \in V_k \} \cup \{ z_{k+1,k} = a_k^{c_i} \}$ *is satisfiable.*

Proof. Take Σ_{k+1}. Let u be one of $z_1, \ldots, z_n$. The inequalities in $\hat{\Sigma}_{k+1}$ compare u_{k+1} with some a^l_{k+1}, hence by (8), the former has to be expanded introducing two new variables. We use a special naming convention for the new variables introduced by this expansion, so that it will be easier to follow the proof. Let us choose the substitution

$$u_{k+1} = f(u^0_k) \to u^1_k. \tag{13}$$

First, we shall show that $\hat{\Sigma}_{k+1}$ is equivalent to two copies of $\hat{\Sigma}_k$, one for u^0's and the other for u^1's. Indeed, $\hat{\Sigma}_{k+1}$ is equivalent to

$$f(\hat{\Sigma}_k[z^0_k/z_k]) \cup \hat{\Sigma}_k[z^1_k/z_k]$$

For $k+1 < p \leqslant n$, by (9) and (13) we get

$$x^1_{p,k} = x_{p,k} \tag{14}$$

and

$$f(x^0_{p,k}) = f(x_{p,k}) \tag{15}$$

Since f is a bijection, it follows that the variables x^0_p and x^1_p are equated for $k+1 < p \leqslant n$ and we can assume that we are dealing just with one copy x_p. A similar statement holds for $y_{k+2}, \ldots, y_n$.

By (12) (for $p = k+1$), (13), (11) and (8) we obtain

$$f(x^1_{k+1,k}) = f(a_k^{c^1_p}) \tag{16}$$

and

$$f^2(x^0_{k+1,k}) = f(a_k^{f(c^0_p)}). \tag{17}$$

Bearing in mind that f is idempotent, we get

$$x^1_{k,k+1} = a_k^{c^1_{k+1}}$$

and

$$x^0_{k+1,k} = a^{c^0_p}.$$

Putting these two together we obtain for $i = 0, 1$,

$$x^i_{k,k+1} = a_k^{c^i_{k+1}}. \tag{18}$$

By (12), (10) and (13) we can conclude that for $l = 0, 1$, $1 \leqslant p \leqslant k$,

$$f(x^l_{p,k}) = f(a_k^{c^1_p})$$

and by idempotence of f

$$x^l_{p,k} = a_k^{c^1_p}.$$

Thus, we have shown that Σ_{k+1} is equivalent to (18) plus two copies of Σ_k, one copy in which for every $1 \leqslant p \leqslant k$, every x_p and every y_p has been replaced by x^0_p and y^0_p, respectively; and the other in which x_p and every y_p has been replaced by $(x_p)_1$ and $(y_p)_1$. This completes the proof of the lemma. □

For $0 \leqslant k \leqslant n$ let

$$\varphi_k = \forall x_k \exists y_k \ldots \forall x_1 \exists y_1 \ \varphi$$

Hence, free variables of φ_k are among $V_k = \{x_{k+1}, y_{k+1}, \ldots, x_n, y_n\}$. The following result shows correctness of the choice of Σ_k.

Lemma 29. *For every $0 \leqslant k \leqslant n$ and for every valuation $\xi : V_k \to \{0,1\}$, ξ satisfies φ_k iff $\Sigma_k \cup \{z_j = a_k^{c_j^{\xi(z_j)}} \mid z_j \in V_k\}$ is satisfiable.*

Proof. The proof is by induction on k. For $k = 0$, it is enough to observe that φ_0 is φ and the statement follows from the proof of NP-hardness of the flat case.

Now, in order to complete the proof let us take any truth assignment $\xi : V_{k+1} \to \{0,1\}$, and for $i, j \in \{0,1\}$ let $\xi_{i,j} : V_k \to \{0,1\}$ be an extension of ξ such that $\xi_{i,j}(x_{k+1}) = i$ and $\xi_{i,j}(y_{k+1}) = j$. Then we have

$$\xi \text{ satisfies } \varphi_{k+1} \tag{19}$$

iff

$$\forall i \in \{0,1\} \exists j \in \{0,1\} \xi_{i,j} \text{ satisfies } \varphi_k \tag{20}$$

iff $\forall i \in \{0,1\} \exists j \in \{0,1\}$

$$\Sigma_k \cup \{ v_k = a_k^{c_k^{\xi(v)}} \mid v \in V_k \} \cup \{ x_{k+1,k} = a_k^{c_{k+1}^i},\ y_{k+1,k} = a_k^{c_{k+1}^j} \} \text{ is satisfiable} \tag{21}$$

iff $\forall i \in \{0,1\}$

$$\Sigma_k \cup \{ v_k = a_k^{c_k^{\xi(v)}} \mid v \in V_k \} \cup \{ x_{k+1,k} = a_k^{c_{k+1}^i} \} \text{ is satisfiable} \tag{22}$$

iff

$$\Sigma_{k+1} \cup \{ v_k = a_k^{c_k^{\xi(v)}} \mid v \in V_{k+1} \} \text{ is satisfiable.} \tag{23}$$

Equivalence of (20) and (21) follows from the induction assumption. Equivalence of (22) and (23) follows from Proposition 28. □

Acknowledgements

This paper would be never written without the continous advice, encouragement and patience of Professor Jerzy Tiuryn. Many thanks go also to Damian Niwiński for the fruitful discussions and suggestions.

The author would also like to express his gratitude towards anonymous referees, whose constructive critique and many helpful comments led to considerable improvement of this article.

References

[1] J.L. Balcazar, J. Diaz, J. Gabarro, Structural Complexity II, Springer, Berlin, 1990.
[2] J.L. Balcazar, J. Diaz, J. Gabarro, Structural Complexity I, 2nd ed., Springer, Berlin, 1995.
[3] M. Benke, Efficient type reconstruction in the presence of inheritance (extended abstract), in: Proc. Int. Symp. MFCS 1993, Springer, Berlin, 1993.
[4] M. Benke, Efficient Type reconstruction in the presence of inheritance, Tech. Report TR94-10(199), Institute of Informatics, Warsaw University, December 1994.
[5] C. Dwork, P. Kanellakis, J.C. Mitchell, On the sequential nature of unification, J. Logic Programming 1, (1984) 35–50.
[6] D. Duffus, W. Poguntke, I. Rival, Retracts and the fixed point problem for finite partially ordered sets, Canad. Math. Bull. 23 (1980) 231–236.
[7] D. Duffus, I. Rival, Retracts of partially ordered sets. J. Austral. Math. Soc. (Ser. A), 27 (1979) 495–506.
[8] D. Duffus, I. Rival, A structure theory for ordered sets. Discrete Math. 35 (1981) 53–118.

[9] A. Frey, Satisfying subtype inequalities in polynomial space, in: P. van Hentenryck, (ed.), Proc. 4th Internat. Symp. on Static Analysis (SAS'97), number 1302 in Lecture Notes in Computer Science, Paris, France, September 1997. Springer, Berlin, pp. 265–277.
[10] M. Hoang, J.C. Mitchell, Lower bounds on type inference with subtypes, in: Conf. Rec. ACM Symp. on Principles of Programming Languages, ACM Press, San Francisco, 1995.
[11] D. Kelly, I. Rival, Crowns, fences and dismantlable lattices, Canad J. Math. 26 (1974) 1257–1271.
[12] P. Lincoln, J.C. Mitchell, Algorithmic aspects of type inference with subtypes, in: Conf. Rec. ACM Symp. Principles of Programming Languages, 1992, pp. 293–304.
[13] J.C. Mitchell, Coercion and type inference, in: Conf. Rec. ACM Symp. Principles of Programming Languages, 1984, pp. 175–185.
[14] J.C. Mitchell, Type inference with simple subtypes, J. Funct. Programming 1 (3) (1991) 245–285.
[15] P. Nevermann, I. Rival, Holes in ordered sets, Graphs and Combin. 1 (1985) 339–350.
[16] C.H. Papadmitriou, Computational complexity, Addison-Wesley, Reading, MA, 1994.
[17] V. Pratt, J. Tiuryn, Satisfiability of inequalities in a poset. Fundam. Inform. 28 (1,2) (1996) 165–182.
[18] A. Quillot, An application of the Helly property to the partially ordered sets, J. Combin. Theory (A) 35 (1983) 185–198.
[19] A. Quillot, On the Helly property working as a compactness criterion for graphs, J. Combin. Theory (A) 40 (1985), 186–193.
[20] M.S. Roddy, Cores and retracts, Order 11 (1) (1994) 1–10.
[21] M.S. Roddy, Fixed points and products, Order 11 (1) (1994) 11–14.
[22] A. Rutkowski, B.S. Schröder, Retractability and the fixed point property for products, Order 11(4) (1994) 353–359.
[23] J. Tiuryn, Subtype inequalities, in: Proc. 7th IEEE Symp. on Logic in Computer Science, Springer, Berlin, pp. 1992, 308–315.
[24] J. Tiuryn, M. Wand, Type reconstruction with recursive types and atomic subtyping, in: M.-C. Gaudel, J.-P. Jouannaud (Eds.), TAPSOFT'93: Theory and Practice of Software Development, Proc. 4th Internat. Joint Conf. CAAP/FASE, Lecture Notes in Computer Science, vol. 668, Springer, Berlin, 1993, pp. 686–701.
[25] M. Wand, P. O'Keefe, On the complexity of type inference with coercion, in: Proc. ACM Conf. Functional Programming and Computer Architecture, 1989.