Contents

Some Complexity Bounds for Subtype Inequalities

Marcin Benke*
Institute of Informatics
Warsaw University
ul. Banacha 2
02-097 Warsaw, POLAND
e-mail: Marcin.Benke@mimuw.edu.pl

December, 1995

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.

Introduction

Recent results of Hoang and Mitchell [HM95] 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 [WO89, PT95, Tiu92, LM92, Ben94]. The latter is equivalent to the retractability problem, known from the theory of partial orders [NR85]. The purpose of the research was to provide some kind of 'taxonomy' amongst posets, having in mind the complexity of satisfiability-checking. 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 through 4 show that for posets for which feasibility of FLAT-SSI is witnessed by formulae of transitive closure logic, SSI is feasible too. Section 5 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.

*This work has been partially supported by Polish KBN grant 2 P301 031 06 and ESPRIT BRA "Gentzen".

1 Preliminaries

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 $\mathcal{T}_Q$ be the term algebra over this signature. The carrier of $\mathcal{T}_Q$ is partially ordered by extending the order from Q to all terms by the rule

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

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

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

where τ's and ρ's are terms over the above signature with variables from 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 $\mathcal{T}_C$ if there is a valuation $v : V \to \mathcal{T}_C$ such that $\tau_i[v] \leq \rho_i[v]$ holds in $\mathcal{T}_C$ for all i.

1.2 Shapes and weak satisfiability

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

The shape of a term $t \in \mathcal{T}_Q$ (without variables) is defined as follows:

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

Note that the subtype order on $\mathcal{T}_Q$ 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:

$$\begin{aligned} Q_0 &= Q \\ Q_{\sigma \to \tau} &= \{t \to u : t \in Q_\sigma, u \in Q_\tau\} \end{aligned}$$

A system of inequalities $\Sigma = \{\tau_1 \leq \rho_1, \ldots, \tau_n \leq \rho_n\}$ 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 $\mathcal{T}_\star$. The most general unifier of $\Sigma_\star$ will be denoted by $mgu(\Sigma_\star)$

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.

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

$$t \leq_\sigma u$$

for an inequality in shape σ.

1.3 Retractions and obstacles

We say that $R \supseteq Q$ 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. For every Q, Q-FLAT-SSI is logspace-equivalent to q-retractability. Henceforth we shall identify flat systems of inequalities over Q with corresponding extensions of Q.

V. Pratt and J. Tiuryn [PT95] 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. The reader is referred to this paper for an in-depth explanation of this concept.

In the mentioned paper, its authors discuss classes of posets for which complete obstacles can be expressed by TC-formulae. This subject is explored further in sections 2 through 4 . This is a generalization of [Ben94], where we discussed other kind of obstacles, complete for the class of Helly posets, which we consider useful in connection with inheritance. Figure 1 shows an example of a poset that is TC-feasible but not Helly.

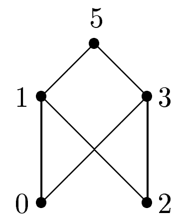

Figure 1: A poset which is TC-feasible but not Helly [PT95].

In fact, TC-feasible posets encompass even more than the class of absolute retracts [NR85, Ben94], tractability of which we conjectured in [Ben94].

1.4 Intractable posets

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

V. Pratt and J. Tiuryn [PT95] show that for n-crowns ($n \geq 2$), FLAT-SSI is NP-complete. Moreover, in [Tiu92] it is shown that for these posets SSI is PSPACE-complete. In section 5 we show how this result can be generalized.

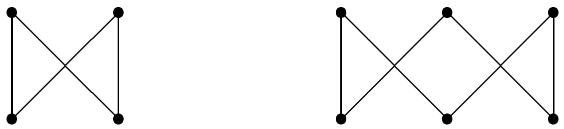

Figure 2: (a) 2-crown (b) 3-crown

1.5 Complexity classes

Beside obvious complexity classes, we use the following abbreviations:

$$NL = NSPACE(\log n) \tag{1}$$

$$AL = ASPACE(\log n) \tag{2}$$
$$AP = \bigcup_{c>0} ATIME(n^c) \tag{3}$$

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

$$ASPACE(s(n)) = \bigcup_{c>0} DTIME(c^{s(n)}) \tag{4}$$
$$ATIME(t(n)) \subseteq DSPACE(t^2(n)) \tag{5}$$

in particular, we have

$$AL = P \tag{6}$$
$$AP = PSPACE \tag{7}$$

2 Transitive closure logic for subtype inequalities

2.1 Syntax

Let $\sigma, \sigma_1, \sigma_2 \ldots$ be shapes. The set of *annotated TC-formulas over Q* is the least set ATC_Q such that

- Every atomic formula $t_1 \leq_\sigma t_2$, where t_1, t_2 are terms from $\mathcal{T}_Q(X)$ is in ATC_Q.
- If φ and ψ are in TC_Q, and every variable x free in φ and ψ has identical annotations in both formulae, then
$$(\varphi \vee \psi),\ (\varphi \wedge \psi)$$
- If φ is in TC_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, $\vec{\sigma} = \sigma_1, \ldots, \sigma_n$, then
$$TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}} . \varphi)(\vec{t_1}, \vec{t_2})$$
is in ATC_Q, where $\vec{x}, \vec{y}$ are n-vectors of individual variables, $\vec{t_1}, \vec{t_2}$ are n-vectors of Q-terms, and $\vec{t}$ denotes the vector $t_1, \ldots, t_n$

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

2.2 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 ($R \models \varphi[v]$):

- $R \models (t_1 \leq_0 t_2)[v]$ iff $v(t_1) \leq_R v(t_2)$;
- $R \models (\varphi \wedge \psi)[v]$ iff $R \models \varphi[v]$ and $R \models \psi[v]$
- $R \models (\varphi \vee \psi)[v]$ iff $R \models \varphi[v]$ or $R \models \psi[v]$
- $R \models (\exists x^0 . \varphi)[v]$ iff $R \models (\varphi[r/x])[v]$ for some $r \in R$;
- $R \models TC(\lambda \vec{x}, \vec{y} . \varphi)(\vec{t}, \vec{u})$ iff there exists a positive integer k and vectors of elements from R $\vec{t} = \vec{r}_0, \vec{r}_1, \ldots \vec{r}_k = \vec{u}$ such that $R \models (\varphi[\vec{r}_i/\vec{x}, \vec{r}_{i+1}/\vec{y}])[v]$ for $i = 0, 1, \ldots, k-1$.

For formulas without free variables, 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. 1, retracts to it if and only if

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

Now the semantics of general formulae:

- $\mathcal{T}_R \models (t_1 \leq_\sigma t_2)[v]$ iff $v(t_1) \leq_{R_\sigma} v(t_2)$;
- $\mathcal{T}_R \models (\varphi \wedge \psi)[v]$ iff $\mathcal{T}_R \models \varphi[v]$ and $\mathcal{T}_R \models \psi[v]$
- $\mathcal{T}_R \models (\varphi \vee \psi)[v]$ iff $\mathcal{T}_R \models \varphi[v]$ or $\mathcal{T}_R \models \psi[v]$
- $\mathcal{T}_R \models (\exists x^\sigma.\varphi)[v]$ iff $\mathcal{T}_R \models (\varphi[r/x])[v]$ for some $r \in R_\sigma$;
- $\mathcal{T}_R \models TC(\lambda \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u})$ iff there exists a positive integer k and vectors of elements from $\mathcal{T}_R$ $\vec{t} = \vec{r}_0, \vec{r}_1, \ldots \vec{r}_k = \vec{u}$ such that $\mathcal{T}_R \models (\varphi[\vec{r}_i/\vec{x}, \vec{r}_{i+1}/\vec{y}])[v]$ for $i = 0, 1, \ldots, k-1$.

Fact 2.1
For flat φ we have

$$\mathcal{T}_R \models \varphi \text{ iff } R \models \varphi$$

We say that φ is a *semantical consequence* of Σ ($\Sigma \models \varphi$) if for every valuation v being a solution of Σ, $\mathcal{T}_Q \models \varphi[v]$.

2.3 Projections

First we define projections on shapes:

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

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: $(\cdot) \downarrow 1, (\cdot) \downarrow 2 : ATC_Q \to ATC_Q$

$$\begin{aligned}
(t \leq_0 u) \downarrow i &= t \leq_0 u \\
(t \leq_{\sigma_1 \to \sigma_2} u) \downarrow 1 &= (u \downarrow 1) \leq_{\sigma_1} (t \downarrow 1) \\
(t \leq_{\sigma_1 \to \sigma_2} u) \downarrow 2 &= (t \downarrow 2) \leq_{\sigma_2} (t \downarrow 2) \\
(\exists x^\sigma.\varphi) \downarrow i &= \exists x^{\sigma \downarrow i}.\varphi[x^{\sigma\downarrow i}/x^\sigma], \; i = 1,2 \\
(TC(\lambda \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(t,u)) \downarrow i &= TC(\lambda x^{\vec{\sigma}\downarrow i}, y^{\vec{\sigma}\downarrow i}.(\varphi \downarrow i))(t \downarrow i, u \downarrow i)
\end{aligned}$$

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

Let R be a poset, $v : X \to \mathcal{T}_R$. We say that v is compatible with (the annotation of) φ if for every variable x free in φ, the shape of $v(x)$ corresponds to the annotation of x in φ.

3 The proof system

3.1 Free Variables

Given an ATC-formula φ (or a term t), we define the set of its *free variables*, $FV(\varphi)$ as follows:

$$\begin{array}{rcl} FV(x) & = & \{x\} \\ FV(t \to u) & = & FV(t) \cup FV(u) \\ FV(t \leq u) & = & FV(t) \cup FV(u) \\ FV(\varphi \wedge \psi) & = & FV(\varphi) \cup FV(\psi) \\ FV(\varphi \vee \psi) & = & FV(\varphi) \cup FV(\psi) \\ FV(\exists x.\varphi) & = & FV(\varphi) \setminus \{x\} \\ FV(TC(\lambda \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u}) & = & (FV(\varphi) \setminus \{\vec{x}, \vec{y}\}) \cup FV(\vec{t}) \cup FV(\vec{u}) \end{array}$$

3.2 Lonely Variables

Given an ATC-formula φ (or a term t), we define the set of its *lonely variables*, $LV(\varphi)$ as follows:

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

Lemma 3.1
If $LV(\varphi) = \emptyset$ and v is compatible with φ then $\mathcal{T}_R \models \varphi[v]$ iff $\mathcal{T}_R \models (\varphi \downarrow i)[v \downarrow i]$ for $i = 1, 2$.

Proof: by induction on φ. The case when $\varphi \equiv t \leq_\sigma u$ is trivial for $\sigma = 0$ and follows from definition of subtype order and projections for complex shapes. The case when $\varphi \equiv \exists x^\sigma.\varphi_1$ is again trivial for $\sigma = 0$. If $\sigma = \sigma_1 \to \sigma_2$ then $\mathcal{T}_R \models \varphi[v]$ iff there exist $r_1 \in R_{\sigma_1}$, $r_2 \in R_{\sigma_2}$ such that $\mathcal{T}_R \models (\varphi_1[r_1 \to r_2/x])[v]$. 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 $\exists$. ∎

3.3 Closures

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

$$\begin{array}{rcl} \overline{t \leq u} & = & t \preceq u \\ \overline{\varphi \wedge \psi} & = & \overline{\varphi} \wedge \overline{\psi} \end{array}$$

$$\overline{\varphi \vee \psi} = \overline{\varphi} \vee \overline{\psi}$$
$$\overline{\exists x^\sigma . \varphi} = \exists x^\sigma . \overline{\varphi}$$

3.4 Inference rules

Let Σ be weakly satisfiable and all its variables be annotated according to $mgu(\Sigma_\star)$. Consider the inference system depicted in Fig. 3

$$\Sigma \vdash t \leq u \text{ for } t \leq u \in \Sigma$$

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

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

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

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

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

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

Figure 3: An inference sytem for ATC-formulae

Lemma 3.2
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 the Lemma 3.1. ∎

3.5 Normal derivations

We shall say that a derivation is in *normal form* if all the applications of the rule ($\downarrow$) are made as early as possible. Now it is easy to observe, that

In the normal derivation of	the last rule is
$\varphi_1 \wedge \varphi_2$	$(\wedge)$
$\varphi_1 \vee \varphi_2$	$(\vee)$
$TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u})$	TC_0 or TC_S

Fact 3.3
Any derivation from a flat system Σ is normal, and the last rule is always an introduction of the main connective.

Fact 3.4
Any normal derivation of $TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u})$ always ends either with single use of TC_0 or like

$$\dfrac{\Sigma \vdash \varphi[\vec{t}/\vec{x}, \vec{r_1}/\vec{y}]}{\Sigma \vdash TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{r_1})} \quad \cdots \quad \dfrac{\Sigma \vdash \varphi[\vec{r_k}/\vec{x}, \vec{u}/\vec{y}]}{\Sigma \vdash TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{r_k}, \vec{u})}$$
$$\vdots$$
$$\dfrac{\Sigma \vdash TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{r_i}) \quad \Sigma \vdash TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{r_i}, \vec{u})}{TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u})}$$

Fact 3.5
For fixed φ, one can check in time polynomial in $|\Sigma|$, whether $\Sigma \vdash \varphi$.

4 Results

4.1 Simple formulas 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.

A *simplification set* of formula φ is the set of simple formulas, defined as follows:

$$\begin{aligned}
[t \leq u] &= \{t \leq u\} \\
[\varphi_1 \vee \varphi_2] &= [\varphi_1] \cup [\varphi_2] \\
[\varphi_1 \wedge \varphi_2] &= \{\psi_1 \wedge \psi_2 : \psi_1 \in [\varphi_1], \psi_2 \in [\varphi_2]\} \\
[\exists x.\varphi] &= \{\exists x.\psi : \psi \in [\varphi]\} \\
[TC(\lambda\, \vec{x}^{\vec{\sigma}}, \vec{y}^{\vec{\sigma}}.\varphi)(\vec{t}, \vec{u})] &= \bigcup_{k \in \omega} \{\exists \vec{z}_1, \ldots, \vec{z}_k.\psi[\vec{t}/\vec{x}, \vec{z}_1/\vec{y}] \wedge \psi[\vec{z}_1/\vec{x}, \vec{z}_2/\vec{y}] \wedge \\
&\quad \ldots \wedge \psi[\vec{z}_k/\vec{x}, \vec{u}/\vec{y}] : \psi \in [\varphi]\}\}
\end{aligned}$$

Lemma 4.1
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 has no bearing on LV). ∎

Lemma 4.2
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 formulas, taking into account normal derivations defined in the section 3.5 ∎

4.2 Canonical form of simple formulas

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 $\varphi \equiv \exists\vec{x}.(t_1 \leq u_1 \wedge \ldots t_n \leq u_n)$, its corresponding system of inequalities is $\Sigma(\varphi) = \{t_1 \leq u_1, \ldots t_n \leq u_n\}$

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 $\Delta = \Sigma(\varphi)$.

4.3 Flat systems

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

$$\Sigma \cup \{t \leq u : Q \models t \leq u\}.$$

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

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

Lemma 4.3
Σ is satisfiable iff $\langle Q \cup \Sigma, \preceq\rangle$ retracts to $\langle Q, \leq\rangle$.

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 by Fact 3.4, there exist $t = r_0, r_1, \ldots, r_k = u \in var(\Sigma) \cup Q$ such that

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

hence

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

Since v is a solution of Σ, we have

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

thus

$$v(t) \leq_Q v(u)$$

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

Let

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

Lemma 4.4
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 3.2 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 $\Sigma \not\vdash NGC(Q)$ then Q_Σ is a partial order and Q_Σ retracts to Q iff $Q_\Sigma \not\models \varphi$ since φ is a complete obstacle for Q. ∎

Lemma 4.5
For every TC-formula ψ,

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

(where $\bar{\varphi}$ denotes the closure of φ defined in section 3.3).

Proof: This lemma is easily proved by structural induction over formulas, taking into account normal derivations defined in the section 3.5 ∎

Lemma 4.6
Let φ be the complete obstacle for Q. For every flat system of inequalities Σ, Σ is satisfiable iff

$$Q \cup \Sigma \not\vdash \bar{\varphi} \vee NGC(Q)$$

Proof: This lemma is actually a conjecture from previous three lemmas. ∎

4.4 Single-shaped systems and formulas

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

Lemma 4.7
Let φ be the complete obstacle for Q and Σ be single-shaped. Σ is satisfiable iff

$$Q \cup \Sigma \not\vdash \varphi \vee NGC(Q)$$

Proof: 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 (see Proposition 12 and Theorem 14 in [PT95]). Then the thesis follows from Lemma 4.6. ∎

A system $\hat{\Sigma}$ is called a σ-view of Σ if the following conditions hold:

1. $\Sigma \vdash \hat{\Sigma}$
2. $\hat{\Sigma}$ is σ-shaped
3. For every σ-shaped Δ, if $\Sigma \vdash \Delta$ then $\hat{\Sigma} \vdash \Delta$.

Lemma 4.8
For every system Σ and shape σ minimal in Σ there exists a σ-view of Σ.

Proof: Using the notation from section 1.2, an example of a σ-view of Σ is

$$E_\sigma(\Sigma) = \{(t \leq_\rho u) \downarrow \pi : \rho \downarrow \pi = \sigma\}$$

∎

4.5 General systems

Theorem 4.9
Let φ be the complete obstacle for Q. For every system of inequalities Σ, Σ is satisfiable iff it is weakly satisfiable and

$$Q \cup \Sigma \not\models \varphi \vee NGC(Q)$$

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

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

Suppose first that the quotient set $var(\Sigma)/\sim$ has only one element. Then satisfiability of Σ follows from the Lemma 4.7.

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 $\Sigma_* \models \tau = z$. 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 $mgu(\Sigma_\star)$, and let $\hat{\Sigma}$ be a σ-view of Σ.

Again, the satisfiability of $\hat{\Sigma}$ follows from the Lemma 4.7. Let $\hat{v} : [y] \to \mathcal{T}_\sigma$ be a solution of $\hat{\Sigma}$ and let

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

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

$$|var(\Sigma_1)/\sim_1| = n$$

$$\Sigma_1 \text{ is weakly satisfiable}$$

To complete the proof we need to prove that

$$\Sigma_1 \not\models \varphi \vee NGC(Q)$$

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

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 4.10
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.

5 Subtyping and Alternation

The aim of this chapter is to establish further links between SSI and FLAT-SSI, claiming the following:

Thesis 5.1
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.

5.1 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 P is discrete, then
   - P-FLAT-SSI is in NL[1];
   - P-SSI is equivalent to the unification, and hence AL-complete.
2. If P is a disjoint union of lattices (but not discrete), then
   - P-FLAT-SSI is NL-complete [Ben94];
   - P-SSI is AL-complete [Tiu92].
3. If P is a non-discrete Helly poset, then
   - P-FLAT-SSI is NL-complete [Ben94];
   - P-SSI is AL-complete [Ben93].
4. If P is a non-discrete TC-feasible poset, then
   - P-FLAT-SSI is NL-complete [PT95];
   - P-SSI is AL-complete (Corollary 4.10).
5. If P is an n-crown ($n > 1$), then
   - P-FLAT-SSI is NP-complete [PT95];
   - P-SSI is AP-hard [Tiu92].

[1] the problem whether it is NL-hard is equivalent to a known open problem in complexity, whether SYMLOGSPACE=NLOGSPACE

5.2 Encoding alternation

In this section we show that the result of Tiuryn 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 as an SSI, given encoding of SAT as FLAT-SSI.

First let us make some assumptions about encodings of instances of SAT as systems of inequalities. Later we show how these assumptions can be either removed or replaced. 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 5.2
Let $\varphi = \varphi(\vec{x})$ be a 3-CNF propositional formula with variables $\vec{x} = x_1, \dots x_n$ (and no other)

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

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

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

Theorem 5.3
Let P be a poset such that P-FLAT-SSI is complete for NP under symmetric *reductions. Then P-SSI is complete for AP.*

Proof:

Let

$$\forall x_1 \exists y_1 \dots \forall x_{p(n)} \exists y_{p(n)} \varphi$$

be an instance of QBF, φ contains no quantifiers

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

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

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

Let q be the smallest positive integer such that $f^q = id$ (such q must exist since Σ is finite, moreover it can't be greater than $|\Sigma|$).

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 $0 \le k \le n$, $i, j \in P$ 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 $0 \le k \le n$.

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

For $k < n$, Δ_{k+1} is Δ_k plus the equations (8–11), with i, j ranging over P.

$$a^i_{k+1} = a^{f(i)}_k \to a^i_k \tag{8}$$

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

$$f^i(z_{p,k+1}) = f^{i+1}(z_{p,k}) \to f^i(z_{p,k}) \quad \text{for } i = 0, \dots q-1 \tag{9}$$

For $1 \leq p \leq k$,

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

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

For every $k \geq 0$, 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 $i \in P$ 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 the equation (12) with i, j ranging over P and $1 \leq p \leq k+1$.

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

The thesis follows from the following lemmas, proved in the next section.

Lemma 5.4
Let $V_k = \{x_{k+1}, y_{k+1}, \dots, x_n, y_n\}$. For all $k \geq 0$, and for every function $\xi : V_k \to \{0,1\}$, $\Sigma_{k+1} \cup \{\, z_k = a^{c^{\xi(v)}_k}_k \mid v \in V_{k+1} \,\}$ is satisfiable iff for every $i \in \{0,1\}$, $\Sigma_k \cup \{\, z_k = a^{c^{\xi(v)}_k}_k \mid v \in V_k \,\} \cup \{z_{k+1,k} = a^{c_i}_k\}$ is satisfiable.

For $0 \leq k \leq n$ let

$$\varphi_k = \forall x_k \exists y_k \dots \forall x_1 \exists y_1 \; \varphi$$

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

Lemma 5.5
For every $0 \leq k \leq n$ and for every valuation $\xi : V_k \to \{0,1\}$, ξ satisfies φ_k iff $\Sigma_k \cup \{z_j = a^{c^{\xi(z_j)}_j}_k \mid z_j \in V_k\}$ is satisfiable.

∎

5.3 Proofs of the lemmas

Proof of lemma 5.4: Take Σ_{k+1}. Let u be one of $z_1, \dots, z_n$. 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

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

For $k+1 < p \leq n$, 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 $k+1 < p \leq n$ 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

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

and

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

Substituting $i = 0,\ j = 1$ in (16) yields $x^1_{k,k+1}]{=}a_k^{c^1_{k+1}}$. Similarly, substituting $i = 0,\ j = q-1$ in (17) we get $x^0_{k+1,k} = a^{c^0_p}$. 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 $l = 0, 1$, $1 \leq p \leq k$ and $0 < i \leq q$,

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

Thus we have shown that Σ_{k+1} is equivalent to (18) plus two copies of Σ_k, one copy in which for every $1 \leq p \leq k$, 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. ∎

Proof of lemma 5.5: 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 $\forall i \in \{0,1\} \exists j \in \{0,1\}$

$$\begin{aligned} \Sigma_k \quad & \cup \quad \{\, v_k = a_k^{c_k^{\xi(v)}} \mid v \in V_k \,\} \\ & \cup \quad \{x_{k+1,k} = a_k^{c^i_{k+1}} \ ,\ y_{k+1,k} = a_k^{c^j_{k+1}}\} \\ & \quad \text{is satisfiable} \end{aligned} \tag{21}$$

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

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

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 5.4. ∎

Acknowledgments

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 for the fruitful discussions and suggestions.

References

[Ben93] Marcin Benke. Efficient type reconstruction in the presence of inheritance (extended abstract). In *Proc. Int. Symp. MFCS 1993*, 1993.

[Ben94] Marcin Benke. Efficient type reconstruction in the presence of inheritance. Technical Report TR94-10(199), Institute of Informatics, Warsaw University, December 1994.

[HM95] My Hoang and John C. Mitchell. Lower bounds on type inference with subtypes. In *Conf. Rec. ACM Symp. Principles of Programming Languages*, 1995.

[LM92] Patrick Lincoln and John C. Mitchell. Algorithmic aspects of type inference with subtypes. In *Conf. Rec. ACM Symp. Principles of Programming Languages*, pages 293–304, 1992.

[Mit84] John C. Mitchell. Coercion and type inference. In *Conf. Rec. ACM Symp. Principles of Programming Languages*, pages 175–185, 1984.

[NR85] P. Nevermann and I. Rival. Holes in ordered sets. *Graphs and Combinatorics*, (1):339–350, 1985.

[PT95] V. Pratt and J. Tiuryn. Satisfiability of inequalities in a poset. Technical Report TR 95-15(215), Institute of Informatics, Warsaw University, 1995.

[Tiu92] Jerzy Tiuryn. Subtype inequalities. In *Proc. 7th IEEE Symp. Logic in Computer Science*, pages 308–315, 1992.

[TW93] Jerzy Tiuryn and Mitchell Wand. Type reconstruction with recursive types and atomic subtyping. In *M.-C. Gaudel and J.-P. Jouannaud (Eds.) TAPSOFT'93: Theory and Practice of Software Development, Proc. 4th Intern. Joint Conf. CAAP/FASE, Springer-Verlag LNCS 668*, pages 686–701, 1993.

[WO89] M. Wand and Patrick O'Keefe. On the complexity of type inference with coercion. In *Proc. ACM Conf. Functional Programming and Computer Architecture*, 1989.