A minimal system of disjunctive properties for strictness analysis

Grzegorz Grudziński

gsg@mimuw.edu.pl

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

Phone: +48 22 658 31 65
Fax: +48 22 658 31 64

Abstract

In this paper we study a simple disjunctive system for strictness analysis. The paper gives a definition of the system (an abstract interpretation and an equivalent inference system) and investigates its properties. This system is minimal in a certain sense and has several important features (we prove polymorphic invariance, optimal interpretations of constants; these are the first proofs of these properties for a system with disjunctions). Also it is more tractable than other methods for disjunctive analysis of programs which were proposed so far. Moreover the way we define the system is similar enough to more traditional abstract interpretations that the standard algorithms for computing program properties can be used here, which does not seem to be the case with the other, more complicated techniques.

`Keywords:`

intersection types, union types, program analysis, strictness analysis, abstract interpretation

1 Introduction

Most traditional abstract interpretation frameworks for strictness analysis (and other kinds of static analysis) which have a characterization in terms of an inference system, correspond to systems with intersections of properties, but without disjunctions. The disjunctive systems are of interest for two main reasons: they can be stronger than similar conjunctive systems and they are very natural extensions of systems with only conjunctive properties. However the field of disjunctive systems has received only moderate interest as far as the number of published papers is concerned. The most important work in this field is still restricted to PhD theses by Jensen (see [1]) and Benton (see [2]).

Jensen in his PhD thesis ([2]) defines an inference system and an equivalent abstract interpretation. However he does not investigate such important properties of the abstract interpretation as correctness with respect to the standard semantics of the language or the problem of optimal constant interpretations for his analysis.

Benton defines only an inference system without corresponding abstract interpretation. Actually it is unclear whether one can define a compositional abstract interpretation which would be sound and complete with regard to his inference system. Benton proves that his system is correct. He also investigates a problem of polymorphic invariance for a traditional (i.e. “conjunctive”) abstract interpretation. Benton does not formulate a similar theorem for his disjunctive analysis. Instead he conjectures that the property does not hold in that case. However this is just the property of the system presented by Benton and not a property of disjunctive systems in general.

Both disjunctive analyses are a bit involved and lack proper semantic tools to investigate some of the essential properties of disjunctive abstract interpretations (like correctness wrt standard denotational semantics or polymorphic invariance). Also there are no algorithms at present for inference of properties in these systems and the abstract interpretation defined by Jensen is different enough from "conjunctive" abstract interpretations that the traditional algorithms of computing abstract properties are not immediately applicable to it.

In this paper we define a minimal system of disjunctive strictness analysis. The system is minimal in the sense that the disjunctions can only occur at the top level of formulas. It is weaker then Jensen's system, but has some attractive properties of the more sophisticated disjunctive systems (it is stronger than the conjunctive version) and furthermore is more tractable: we prove that it is correct, has the polymorphic invariance property and that optimal abstract interpretations of the constants of the language exist in this system. We define an abstract interpretation and an equivalent inference system. The abstract interpretation is defined so that the traditional methods of computing abstract properties can be immediately applied. Also the abstract domains are much smaller than in the case of Jensen's system, thus significantly decreasing the complexity of computation of properties of programs.

2 Abstract interpretation and its properties

This section gives the definition of a minimal disjunctive abstract interpretation framework. It is built upon a traditional abstract interpretation, which we assume is given.

Assume that the base abstract interpretation we are given has the abstract domains defined as in Figure 1.

$$\begin{aligned} A_\iota &= \text{some finite lattice of properties} \\ A_{\sigma\to\tau} &= A_\sigma \to A_\tau \end{aligned}$$

Figure 1: Abstract domains for a simple disjunctive abstract interpretation

By $\leq^A$ we denote the partial order relation for the conjunctive abstract domains. Also by $\mathbf{f}_\sigma$ and $\mathbf{t}_\sigma$ we denote the least and the greatest element of the abstract domain for type σ, respectively.

Figure 2 defines the abstract disjunctive domains (ordered by set inclusion). We write $\mathbb{P}_H(A)$ to denote the lattice of downwards closed subsets of the domain A and we write $\downarrow X$ for the downward closure of the set X.

$$\widehat{A}_\sigma = \mathbb{P}_H(A_\sigma)$$

Figure 2: Abstract domains for a minimal disjunctive abstract interpretation

Figure 3 (see p. 3) gives a definition of an abstract semantics which uses these domains.

Definition 2.1 (disjunctive environment)
We say that an environment ρ is a *disjunctive environment* if for each variable x^σ in the domain of ρ the value $\rho(x^\sigma)$ belongs to $\widehat{A}_\sigma$ and this value is a principal ideal (i.e. the downward closure of a single element).

□

We assume that for the disjunctive abstract interpretation we work only with disjunctive environments.

Definition 2.2 (set application)
For any $F \in \widehat{A}_{\sigma\to\tau}$ and $a \in \widehat{A}_\sigma$ we define

$$F \odot a = \bigcup_{f \in F} \vec{f}(a)$$

□

$$\begin{array}{lcl}
[\![x^\sigma]\!]_{\widehat{A}}\,\rho & = & \rho(x^\sigma) \\[1ex]
[\![c^\sigma]\!]_{\widehat{A}}\,\rho & = & \text{an element of } \widehat{A}_\sigma \\[1ex]
[\![e_1^{\sigma\to\tau} e_2^\sigma]\!]_{\widehat{A}}\,\rho & = & ([\![e_1^{\sigma\to\tau}]\!]_{\widehat{A}}\,\rho) \odot ([\![e_2^\sigma]\!]_{\widehat{A}}\,\rho) \\[1ex]
[\![\lambda x^\sigma . e^\tau]\!]_{\widehat{A}}\,\rho & = & \{g \in A_{\sigma\to\tau} \mid \forall v \in A_\sigma\ (g(v) \in [\![e^\tau]\!]_{\widehat{A}}\,\rho[x \mapsto \downarrow\{v\}])\} \\[1ex]
\multicolumn{3}{l}{[\![\texttt{if}\ e_1\ \texttt{then}\ e_2\ \texttt{else}\ e_3\ \texttt{fi}]\!]_{\widehat{A}}\,\rho} \\
 & = & \begin{cases} \{\mathbf{f}_\sigma\} & \text{if } [\![e_1]\!]_{\widehat{A}}\,\rho = \{\mathbf{f}\} \\ ([\![e_2^\sigma]\!]_{\widehat{A}}\,\rho) \cup ([\![e_3^\sigma]\!]_{\widehat{A}}\,\rho) & \text{otherwise} \end{cases} \\[1ex]
[\![\texttt{fix}]\!]_{\widehat{A}}\,\rho & = & \text{the least fixpoint operator}
\end{array}$$

Figure 3: A simple disjunctive abstract semantics

Fact 2.3
For any term e and any disjunctive environment ρ, the value $[\![e]\!]_{\widehat{A}}\,\rho$ is downward closed.

2.1 Correctness of the abstract interpretation

The correctness of the abstract interpretation is established using logical relations. Our aim is to define a framework which can be used for different base domains and different types of analyses (though all our examples concern a particular strictness analysis) rather than a particular analysis. For this purpose we assume that the definition of the logical relation for base types is already given and we only have to define it for compound types.

We define the type-indexed family $\{\,\widehat{\mathcal{S}}^\sigma\,\}_{\sigma\in\mathrm{Types}}$ of relations (where for each type σ the relation $\widehat{\mathcal{S}}^\sigma$ is a subset of $D_\sigma \times \widehat{A}_\sigma$) as follows:

$$\begin{array}{rcl}
d\ \widehat{\mathcal{S}}^\iota\ e & \Leftrightarrow & \text{according to the particular analysis} \\
f\ \widehat{\mathcal{S}}^{\sigma\to\tau}\ g & \Leftrightarrow & \forall d \in D_\sigma\ \forall a \in \widehat{A}_\sigma\ (d\ \widehat{\mathcal{S}}^\sigma\ a\ \Rightarrow\ (fd)\ \widehat{\mathcal{S}}^\tau\ (g \odot a))
\end{array}$$

We extend the definition of logical relations to environments.

Definition 2.4
Assume that ρ is a standard environment (a finite function on variable names which ranges over standard domains) and $\widehat{\rho}$ is an abstract environment (i.e. it ranges over abstract domains). Assume also that $dom(\rho) = dom(\widehat{\rho})$. We write $\rho\ \mathcal{S}\ \widehat{\rho}$ to mean that for each variable $x^\sigma \in dom(\rho)$ we have $\rho(x^\sigma)\ \widehat{\mathcal{S}}^\sigma\ \widehat{\rho}(x^\sigma)$.

□

In the following we need two simple lemmas.

Lemma 2.5
Let us take an arbitrary logical relation $\{\ \widehat{\mathcal{S}}^\sigma\ \}_{\sigma\in\text{Types}}$. Assume that for any base type ι and for any $v \in D_\iota$ and $a_1, a_2 \in \widehat{A}_\iota$ such that $v\ \widehat{\mathcal{S}}^\iota\ a_1$ and $v\ \widehat{\mathcal{S}}^\iota\ a_2$, we have $v\ \widehat{\mathcal{S}}^\iota\ (a_1 \cup a_2)$.

Then for any type σ and for any $v \in D_\sigma$ and $a_1, a_2 \in \widehat{A}_\sigma$ such that $v\ \widehat{\mathcal{S}}^\sigma\ a_1$ and $v\ \widehat{\mathcal{S}}^\sigma\ a_2$ we also have $v\ \widehat{\mathcal{S}}^\sigma\ (a_1 \cup a_2)$.

Lemma 2.6
For any types σ and τ, any terms u^τ, t^σ and any disjunctive environment $\widehat{\rho}$, the following equation is true:

$$[\![(\lambda x^\sigma.u^\tau)t^\sigma]\!]_{\widehat{A}}\ \widehat{\rho} = \bigcup_{v\in[\![t^\sigma]\!]_{\widehat{A}}\ \widehat{\rho}} [\![u^\tau]\!]_{\widehat{A}}\ (\widehat{\rho}[x \mapsto \downarrow\{v\}])$$

Now we state and prove the main property of logical relations for our semantics.

Theorem 2.7
For each type σ, each term e^σ, and all environments $\rho, \widehat{\rho}$ such that $\rho\ \widehat{\mathcal{S}}\ \widehat{\rho}$ we have:

$$([\![e^\sigma]\!]_D\ \rho)\ \widehat{\mathcal{S}}^\sigma\ ([\![e^\sigma]\!]_{\widehat{A}}\ \widehat{\rho})$$

Proof.
The proof is by induction over the structure of e. The cases for constants and variables are easy.

Consider the case of application. Assume that $e \equiv u^{\sigma\to\tau}v^\tau$. By the induction hypothesis we know that $[\![u]\!]_D\ \rho\ \widehat{\mathcal{S}}^{\sigma\to\tau}\ [\![u]\!]_{\widehat{A}}\ \widehat{\rho}$ and $[\![v]\!]_D\ \rho\ \widehat{\mathcal{S}}^\sigma\ [\![v]\!]_{\widehat{A}}\ \widehat{\rho}$. Using the definition of the abstract semantics we get $\forall d \in D_\sigma \forall a \in \widehat{A}_\sigma (d\ \widehat{\mathcal{S}}^\sigma\ a\ \Rightarrow\ (([\![u]\!]_D\ \rho)d)\ \widehat{\mathcal{S}}^\tau\ (([\![u]\!]_{\widehat{A}}\ \widehat{\rho}) \odot a)$.

By the assumption we get $([\![u]\!]_D\ \rho)([\![v]\!]_D\ \rho)\ \widehat{\mathcal{S}}^\tau\ (([\![u]\!]_{\widehat{A}}\ \widehat{\rho}) \odot ([\![v]\!]_{\widehat{A}}\ \widehat{\rho}))$, which is exactly $([\![uv]\!]_D\ \rho)\ \widehat{\mathcal{S}}^\tau\ ([\![uv]\!]_{\widehat{A}}\ \widehat{\rho})$.

Now consider the case of abstraction. Assume that $e \equiv \lambda x^\sigma.u^\tau$. By the induction hypothesis we know that for each $d \in D_\sigma, a \in \widehat{A}_\sigma$ such that a is a downward closure of a single element and $d\ \widehat{\mathcal{S}}^\sigma\ a$ we have $([\![u]\!]_D\ (\rho[x \mapsto d]))\ \widehat{\mathcal{S}}^\tau\ ([\![u]\!]_{\widehat{A}}\ (\widehat{\rho}[x \mapsto a]))$. By Lemma 2.5 for each $d \in D_\sigma$ and $a \in \widehat{A}_\sigma$ such that $d\ \widehat{\mathcal{S}}^\sigma\ a$ we have $([\![u]\!]_D\ (\rho[x \mapsto d]))\ \widehat{\mathcal{S}}^\tau\ \bigcup_{v\in a}([\![u]\!]_{\widehat{A}}\ (\widehat{\rho}[x \mapsto \downarrow\{v\}]))$. By Lemma 2.6 this concludes the proof.

□

Assuming that the logical relation on base types is suitably defined, we conclude that the disjunctive interpretation we have defined is correct with regard to the standard semantics.

2.2 Optimal constant interpretations

This section is devoted to the proof of the property that the abstract interpretation defined in this paper has optimal interpretations for constants of the language. This means that for each constant in the language there exists an element of the abstract domain (for the same type) which is the least (in the domain ordering) sound interpretation of this constant (with respect to its standard semantics).

In order to state this property in a formal way, we need to introduce abstraction and concretisation functions for the abstract semantics. First we need to define the notion of a Galois connection.

Definition 2.8 (Galois connection)
Let A and B be complete lattices and let $f : A \to B$ and $g : B \to A$ be monotone maps. Then f and g are said to form a *Galois connection* with left (or lower) component f and right (or upper) component g if for all $a \in A$ and $b \in B$:

$$f(a) \sqsubseteq b \iff a \sqsubseteq g(b)$$

□

With the base abstract interpretation we inherit the concretisation functions $\gamma_\sigma : A_\sigma \to \mathbb{P}_H(D_\sigma)$ and the abstraction functions $\alpha_\sigma : \mathbb{P}_H(D_\sigma) \to A_\sigma$. We assume that the concretisation functions preserve meets and that the abstraction and the concretisation functions form a Galois connection. Thus we also assume that the given abstract interpretation has optimal interpretation for constants (in the sense defined at the beginning of this section). In this section we prove that the disjunctive interpretation also has the same property.

Definition 2.9 (abstraction and concretisation functions)
For each type σ the *disjunctive concretisation function* $\widehat{\gamma}_\sigma : \widehat{A}_\sigma \to \mathbb{P}_H(D_\sigma)$ is defined as

$$\widehat{\gamma}_\sigma(a) = \bigcup(\vec{\gamma_\sigma}(a))$$

□

We need a few lemmas concerning the properties of the abstraction and concretisation functions.

Lemma 2.10 (downwards closure)
For each type σ and each element $a \in \widehat{A}_\sigma$, the set $\widehat{\gamma}_\sigma(a)$ is downwards closed.

Lemma 2.11 (meet preservation)
The abstraction functions preserve meets, that is for each type σ and each two elements $a, b \in A_\sigma$ we have the following equality

$$\widehat{\gamma}_\sigma(a \cap b) = \widehat{\gamma}_\sigma(a) \cap \widehat{\gamma}_\sigma(b)$$

Theorem 2.12 (optimal constant interpretations)
For each constant c_σ of the language there is a *least* $a \in \widehat{A}_\sigma$ such that

$$[\![c_\sigma]\!]_D \in \widehat{\gamma}_\sigma(a)$$

Proof.
Take any constant c_σ and assume that there is no least interpretation for it, i.e. there are two different minimal, incomparable elements $a, b \in A_\sigma$ such that $[\![c_\sigma]\!]_D \in \widehat{\gamma}_\sigma(a)$ and $[\![c_\sigma]\!]_D \in \widehat{\gamma}_\sigma(b)$, which means that $[\![c_\sigma]\!]_D \in \widehat{\gamma}_\sigma(a) \cap \widehat{\gamma}_\sigma(b)$. But we know that $\widehat{\gamma}_\sigma$ preserves meets, so we have that $[\![c_\sigma]\!]_D \in \widehat{\gamma}_\sigma(a \cap b)$, which contradicts the assumption that a and b are minimal and incomparable.

□

2.3 Polymorphic invariance of the abstract interpretation

In this section we prove a polymorphic invariance property for our abstract interpretation. The definitions and proofs are obviously inspired by those given by Benton. However, this is the first paper (according to the best knowledge of the author) to consider the polymorphic invariance problem for an abstract interpretation with disjunctive properties (and the particular disjunctive abstract interpretation considered in this paper is also new).

First we need to extend the programming language to include type variables, which are ranged over with α, β:

$$\sigma \ ::= \ \alpha \mid \iota \mid \sigma \to \sigma$$

Then we define a *type substitution* to be a function from type variables to types, and extend it to function from types to types in the natural way. We also define the action of type substitutions on terms: if t_σ is a term and S is a type substitution, then $S(t_\sigma)$ is defined in the obvious way by defining:

$$\begin{array}{lcl} S(x_\sigma) & = & x_{S(\sigma)} \\ S(\lambda x_\sigma.e) & = & \lambda x_{S(\sigma)}.S(e) \\ S(e_1 e_2) & = & (S(e_1))(S(e_2)) \end{array}$$

We assume that the substitution is defined in such a way that it does appropriate renaming where name conflicts would arise otherwise.

We extend both abstract interpretations to this new language by assuming that A_α is the two–element lattice consisting of $\mathbf{f}_\alpha$ and $\mathbf{t}_\alpha$, for any type variable α.

For the following we assume that we are given an arbitrary type substitution S.

We define the maps $\epsilon_\sigma : \widehat{A}_\sigma \to \widehat{A}_{S(\sigma)}$ and $\kappa_\sigma : \widehat{A}_{S(\sigma)} \to \widehat{A}_\sigma$ inductively as follows:

$$\begin{array}{ll} \epsilon_\iota = id_{\widehat{A}_\iota} & \kappa_\iota = id_{\widehat{A}_\iota} \\ \epsilon_\alpha(\mathbf{t}_\alpha) = \mathbf{t}_{S(\alpha)} & \epsilon_\alpha(\mathbf{f}_\alpha) = \mathbf{f}_{S(\alpha)} \\ \kappa_\alpha(x) = \mathbf{t}_{S(\alpha)} \ (x \neq \mathbf{f}) & \kappa_\alpha(\mathbf{f}_{S(\alpha)}) = \mathbf{f}_{S(\alpha)} \\ \epsilon_{\sigma\to\tau}(f) = \epsilon_\tau \circ f \circ \kappa_\sigma & \kappa_{\sigma\to\tau}(g) = \kappa_\tau \circ g \circ \epsilon_\sigma \end{array}$$

By induction on types we can prove that ϵ_σ and κ_σ form a Galois connection for each type σ. Therefore ϵ_σ preserves meets and κ_σ preserves joins.

In the following we use a slightly non-standard notion of a logical relation. Namely, by a logical relation we mean not only a type-indexed family of relations defined on pairs of domains for the same type, but also a type-indexed family of relations defined on such pairs of domains that the type of the second is a function of the type of the first (the standard notion is a special case where this function is just identity).

However it is similar enough in spirit to the standard definition of this notion that we do not want to introduce a special name for it.

Now we define a logical relation $\{\ \mathcal{S}^\sigma\ \}_{\sigma\in\text{Types}}$, where $\mathcal{S}^\sigma \ \subseteq \widehat{A}_\sigma \times \widehat{A}_{S(\sigma)}$:

$$d\ \mathcal{S}^\iota\ a \equiv d = e$$

$$d\ \mathcal{S}^\alpha\ a \equiv d = \kappa_\alpha(a)$$

$$f\ \mathcal{S}^{\sigma\to\tau}\ g \equiv \forall d, a\, (d\ \mathcal{S}^\sigma\ a \ \Rightarrow \ (f \odot d)\ \mathcal{S}^\tau\ (g \odot a))$$

This relation satisfies the following properties:

1. for any σ and any $d \in \widehat{A}_\sigma$ we have $d\ \mathcal{S}^\sigma\ \epsilon_\sigma(d)$;
2. for any σ, $d \in \widehat{A}\sigma$ and $a \in \widehat{A}_{S(\sigma)}$, if $d\ \mathcal{S}^\sigma\ a$ then $d = \kappa_\sigma(a)$.

Theorem 2.13 (polymorphic invariance)
Assume that e_σ is a term and $\rho_1^{\widehat{A}}, \rho_2^{\widehat{A}}$ are abstract environments such that

- $FV(e) \subseteq dom(\rho_1^{\widehat{A}})$,
- for all $x^\sigma \in dom(\rho_1^{\widehat{A}})$ we have $x^{S(\sigma)} \in dom(\rho_2^{\widehat{A}})$,
- for all $x^\sigma \in dom(\rho_1^{\widehat{A}})$ we have $\rho_1^{\widehat{A}}(x^\sigma)\ \mathcal{S}^\sigma\ \rho_2^{\widehat{A}}(x^{S(\sigma)})$.

Then

$$[\![e]\!]_{\widehat{A}}\ \rho_1^{\widehat{A}}\ \mathcal{S}^\sigma\ [\![S(e)]\!]_{\widehat{A}}\ \rho_2^{\widehat{A}}$$

Proof.
By induction on the structure of terms.

□

3 Simple disjunctive properties

Figure 4 (see p. 7) gives the formation rules for the disjunctive properties. In that figure, the family $\{\mathrm{L}_\sigma\}_{\sigma\in\mathbf{Types}}$ of sets of properties refers to the standard conjunctive property system, while $\{\widehat{\mathrm{L}}_\sigma\}_{\sigma\in\mathbf{Types}}$ is the family of disjunctive properties.

$$\mathbf{f}_\sigma \in \mathrm{L}_\sigma \qquad\qquad \mathbf{t}_\sigma \in \mathrm{L}_\sigma$$

$$\frac{\phi,\psi \in \widehat{\mathrm{L}}_\sigma}{\phi\wedge\psi \in \mathrm{L}_\sigma} \qquad\qquad \frac{\phi \in \mathrm{L}_\sigma \quad \psi \in \mathrm{L}_\tau}{(\phi\to\psi) \in \mathrm{L}_{\sigma\to\tau}}$$

$$\frac{\phi \in \mathrm{L}_\sigma}{\phi \in \widehat{\mathrm{L}}_\sigma} \qquad\qquad \frac{\phi,\psi \in \widehat{\mathrm{L}}_\sigma}{\phi\vee\psi \in \widehat{\mathrm{L}}_\sigma}$$

Figure 4: Formation rules for simple disjunctive strictness properties

Figure 5 (see p. 8) defines the entailment relation on strictness properties. The semantics of properties is given in Fig. 6 (see p. 8).

Since we are dealing with a framework rather than a particular abstract interpretation, we assume that we are given interpretation of constant properties and that we only need to give interpretation to compound properties. Of the interpretations of constant properties we require the following:

1. for each base type ι, each element a of $\widehat{A}_\iota$ is definable (i.e. there is a property $\phi_a \in \widehat{L}_\iota$ such that $a = [\![\phi_a]\!]_\iota$);

Lemma 3.1 (normal form for properties)
For each type σ and property ϕ_σ there exists a property ψ_σ such that ψ_σ is a disjunction of disjunction–free properties and $\phi_\sigma \le \psi_\sigma$ and $\psi_\sigma \le \phi_\sigma$.

Proof.
Induction on the structure of ϕ_σ.

□

Now we need to prove a number of properties concerning the relationship between the (syntactic) properties and their semantic interpretations.

$$\phi \leq \phi \text{ [refl]} \qquad \phi \leq \mathbf{t}_\sigma \text{ [t]} \qquad \mathbf{f}_\sigma \leq \phi \text{ [f]}$$

$$\frac{\phi \leq \gamma \quad \gamma \leq \psi}{\phi \leq \psi} \text{ [trans]}$$

$$\phi \wedge \psi \leq \phi \text{ [}\wedge\text{E-1]} \qquad \phi \wedge \psi \leq \psi \text{ [}\wedge\text{E-2]}$$

$$\phi \leq \phi \vee \psi \text{ [}\vee\text{I-1]} \qquad \psi \leq \phi \vee \psi \text{ [}\vee\text{I-2]}$$

$$\frac{\phi \leq \psi_1 \quad \phi \leq \psi_2}{\phi \leq \psi_1 \wedge \psi_2} \text{ [}\wedge\text{I]} \qquad \frac{\phi_1 \leq \psi \quad \phi_2 \leq \psi}{\phi_1 \vee \phi_2 \leq \psi} \text{ [}\vee\text{E]}$$

$$\mathbf{t}_{\sigma\to\tau} \leq \mathbf{t}_\sigma \to \mathbf{t}_\tau \text{ [t}\to\text{]} \qquad \mathbf{t}_\sigma \to \mathbf{f}_\tau \leq \mathbf{f}_{\sigma\to\tau} \text{ [f}\to\text{]}$$

$$\frac{\phi_2 \leq \phi_1 \quad \psi_1 \leq \psi_2}{(\phi_1 \to \psi_1) \leq (\phi_2 \to \psi_2)} \text{ [}\to\text{]}$$

$$(\phi \to \psi_1) \wedge (\phi \to \psi_2) \leq (\phi \to \psi_1 \wedge \psi_2) \text{ [}\to\wedge\text{]}$$

$$\phi \wedge (\psi_1 \vee \psi_2) \leq (\phi \wedge \psi_1) \vee (\phi \wedge \psi_2) \text{ [dist]}$$

Figure 5: Entailment relation for simple disjunctive strictness properties

Let $\phi, \phi_1, \phi_2 \in \widehat{L}_\sigma$ and $\psi \in \widehat{L}_\tau$

$[\![]\!]_\sigma : \widehat{L}_\sigma \to \widehat{A}_\sigma$

$$\begin{array}{lcl}
[\![\phi]\!]_\sigma & = & \text{some element of } \widehat{A}_\sigma \text{ (for } \phi \text{ which is a constant property)} \\
[\![\phi_1 \wedge \phi_2]\!]_\sigma & = & [\![\phi_1]\!]_\sigma \cap [\![\phi_2]\!]_\sigma \\
[\![\phi_1 \vee \phi_2]\!]_\sigma & = & [\![\phi_1]\!]_\sigma \cup [\![\phi_2]\!]_\sigma \\
[\![\phi \to \psi]\!]_{\sigma\to\tau} & = & \{f \in A_{\sigma\to\tau} \mid \vec{f}([\![\phi]\!]_\sigma) \subseteq [\![\psi]\!]_\tau\}
\end{array}$$

Figure 6: Semantics of properties

Lemma 3.2 (definability)
For each type σ and each element $a \in \widehat{A}_\sigma$ there is a property $\phi_a \in \widehat{L}_\sigma$ such that

$$a = [\![\phi_a]\!]_\sigma$$

Proof.
Induction on the structure of σ.

$\square$

Lemma 3.3
For each type σ and each property ϕ_σ which does not contain disjunctions, the semantic interpretation $[\![\phi_\sigma]\!]_{\widehat{A}}$ is a principal ideal.

Theorem 3.4 (soundness of entailment)
For all types σ and all properties $\phi, \psi \in \widehat{L}_\sigma$

$$\phi \leq \psi \;\Rightarrow\; [\![\phi]\!]_\sigma \subseteq [\![\psi]\!]_\sigma$$

Proof.
By checking validity of each axiom and inference rule.

$\square$

Theorem 3.5 (completeness of entailment)
For all types σ and all properties $\phi, \psi \in \widehat{L}_\sigma$

$$[\![\phi]\!]_\sigma \subseteq [\![\psi]\!]_\sigma \;\Rightarrow\; \phi \leq \psi$$

Proof.
Induction on the structure of the properties.

$\square$

4 The simple disjunctive inference system

The inference system for simple disjunctive strictness properties presented in Fig. 7 (see p. 10) is inspired by the inference system given by Jensen in [1]. However it is quite different. The syntactic restriction on the properties requires completely different inference rules for abstraction and application (these rules are not based on any presented in the literature). The resulting system is sufficiently simple that it may be possible to invent a property inference algorithm for it.

Also the corresponding abstract semantics is very different from the one given by Jensen and much easier to work with by using traditional techniques of proof. Also, unlike Jensen's approach, it is possible to use standard abstract interpretation algorithms for computing the properties.

5 Equivalence of the logic and the abstract interpretation

In this section we prove that the abstract interpretation of Sec. 4 (see p. 9) is sound and complete with respect to the abstract interpretation of Sec. 2 (see p. 2).

Theorem 5.1 (soundness of logic)
Let e be a term of type σ and assume that Γ is an environment mapping free variables of e to properties. Let $\widehat{\rho}_\Gamma$ be defined as $\widehat{\rho}_\Gamma(x) = [\![\Gamma(x)]\!]$ for all free variables x of term e. Then for any property $\phi \in \widehat{L}_\sigma$ we have

$$\Gamma \vdash e : \phi \;\Rightarrow\; [\![e]\!]_{\widehat{A}}\, \widehat{\rho}_\Gamma \subseteq [\![\phi]\!]$$

$$\Gamma_x, x^\sigma : \phi^\sigma \vdash x^\sigma : \phi^\sigma \text{ [var]}$$

$$\frac{\Gamma_x, x^\sigma : \phi_1^\sigma \vdash t^\tau : \psi_{1,1}^\tau \vee \ldots \vee \psi_{m,1}^\tau \quad \ldots \quad \Gamma_x, x^\sigma : \phi_n^\sigma \vdash t^\tau : \psi_{1,n}^\tau \vee \ldots \vee \psi_{m,n}^\tau}{\Gamma_x \vdash (\lambda x^\sigma . t^\tau) : \bigvee_{p \in \{1,\ldots,m\}^{\{1,\ldots,n\}}} \phi_1 \to \psi_{p(1),1} \wedge \ldots \wedge \phi_n \to \psi_{p(n),n}} \text{ [abs]}$$

$$\frac{\begin{array}{c}\Gamma \vdash t^{\sigma\to\tau} : (\phi_1^\sigma \to \psi_{1,1}^\tau \wedge \ldots \wedge \phi_n^\sigma \to \psi_{1,n}^\tau) \vee \ldots \vee (\phi_1^\sigma \to \psi_{m,1}^\tau \wedge \ldots \wedge \phi_n^\sigma \to \psi_{m,n}^\tau) \\ \Gamma \vdash s^\sigma : \phi_1^\sigma \vee \ldots \vee \phi_n^\sigma\end{array}}{\Gamma \vdash (t^{\sigma\to\tau} s^\sigma)^\tau : \psi_{1,1}^\tau \vee \ldots \vee \psi_{1,n}^\tau \vee \ldots \vee \psi_{m,1}^\tau \vee \ldots \vee \psi_{m,n}^\tau} \text{ [app]}$$

$$\frac{\Gamma \vdash t^\sigma : \phi^\sigma \quad \Gamma \vdash t^\sigma : \psi^\sigma}{\Gamma \vdash t^\sigma : \phi^\sigma \wedge \psi^\sigma} \text{ [conj]}$$

$$\frac{\Gamma \vdash t^\sigma : \phi^\sigma \quad \phi^\sigma \leq_\sigma \psi^\sigma}{\Gamma \vdash t^\sigma : \psi^\sigma} \text{ [sub]}$$

$$\Gamma \vdash \underline{n}^\iota : \mathbf{t}_\iota$$

$$\frac{\Gamma \vdash s^\iota : \phi^\iota \quad \Gamma \vdash s^\iota : \psi^\iota}{\Gamma \vdash \mathtt{plus}^{\iota\to\iota\to\iota}\, s^\iota\, t^\iota : \phi^\iota \wedge \psi^\iota} \text{ [arith]}$$

$$\frac{\Gamma \vdash s^\iota : \mathbf{f}_\iota}{\Gamma \vdash \mathtt{if}^{\iota\to\sigma\to\sigma\to\sigma}\ s^\iota\ \mathtt{then}\ t_1^\sigma\ \mathtt{else}\ t_2^\sigma\ \mathtt{fi} : \mathbf{f}_\sigma} \text{ [cond1]}$$

$$\frac{\Gamma \vdash t_1^\sigma : \phi_1^\sigma \quad \Gamma \vdash t_2^\sigma : \phi_2^\sigma}{\Gamma \vdash \mathtt{if}^{\iota\to\sigma\to\sigma\to\sigma}\ s^\iota\ \mathtt{then}\ t_1^\sigma\ \mathtt{else}\ t_2^\sigma\ \mathtt{fi} : \phi_1^\sigma \vee \phi_2^\sigma} \text{ [cond2]}$$

$$\frac{\Gamma \vdash \lambda x^\sigma . s^\sigma : \phi^\sigma \to \phi^\sigma}{\Gamma \vdash \mathtt{fix}(\lambda x^\sigma . s^\sigma) : \phi^\sigma} \text{ [fix]}$$

Figure 7: Inference system for simple disjunctive strictness properties

Proof.
By induction on the structure of the derivation of the assertion $\Gamma \vdash e : \phi$ in the program logic. This requires checking correctness of the inference rules.

The validity of the rules [**var**], [**conj**], [**tau**] are straightforward. We can also assume that sound interpretations of constants have been chosen. We are left with the task of verifying the rules for abstraction and application.

Consider the abstraction rule:

$$\frac{\Gamma_x, x^\sigma : \phi_1^\sigma \vdash t^\tau : \psi_{1,1}^\tau \vee \ldots \vee \psi_{m,1}^\tau \quad \ldots \quad \Gamma_x, x^\sigma : \phi_n^\sigma \vdash t^\tau : \psi_{1,n}^\tau \vee \ldots \vee \psi_{m,n}^\tau}{\Gamma_x \vdash (\lambda x^\sigma . t^\tau) : \bigvee_{p \in \{1,\ldots,m\}^{\{1,\ldots,n\}}} \phi_1 \to \psi_{p(1),1} \wedge \ldots \wedge \phi_n \to \psi_{p(n),n}} \text{[abs]}$$

For each of the premises (i.e. for each $i \in \{1, \ldots, n\}$) the induction assumption is that:

$$[\![t]\!]_{\widehat{A}} \, (\widehat{\rho}_\Gamma[x \mapsto [\![\phi_i]\!]]) \subseteq [\![\psi_{1,i} \vee \ldots \vee \psi_{m,i}]\!]$$

Let F mean $[\![\lambda x.t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma$. We need to prove that

$$F \subseteq [\![\bigvee \{\phi_1 \to \psi_{p(1),1} \wedge \phi_n \to \psi_{p(n),n} \mid p \in \{1, \ldots, m\}^{\{1,\ldots,n\}}\}]\!]$$

By the definition $F = \{g \in A_{\sigma\to\tau} | \forall v \in A_\sigma (g(v) \in [\![t]\!]_{\widehat{A}} \, (\widehat{\rho}_\Gamma[x \mapsto \downarrow\{v\}]))\}$. Thus it is enough to prove that for each $f \in F$ there exists $p \in \{1, \ldots, m\}^{\{1,\ldots,n\}}$ such that $f \in [\![\phi_1 \to \psi_{p(1),1} \wedge \ldots \wedge \phi_n \to \psi_{p(n),n}]\!]$.

Take an arbitrary $f \in F$. We need to show that there is $p \in \{1, \ldots, m\}^{\{1,\ldots,n\}}$ such that $\vec{f}([\![\phi_i]\!]) \subseteq [\![\psi_{p(i),i}]\!]$, for all $i \in \{1, \ldots, n\}$. Select any $i \in \{1, \ldots, n\}$. By Lemma 3.3 we know that there exists $v_i \in A_\sigma$ such that $[\![\phi_i]\!] = \downarrow\{v_i\}$. By the definition of F we know that $f(v_i) \in [\![t]\!]_{\widehat{A}} \, (\widehat{\rho}_\Gamma[x \mapsto \downarrow\{v_i\}])$. By the induction hypothesis (and monotonicity of f) it follows that $f(v_i) \in [\![\psi_{1,i} \vee \ldots \vee \psi_{m,i}]\!]$. This means that there exists $j_i \in \{1, \ldots, m\}$ such that $f(v_i) \in [\![\psi_{j_i,i}]\!]$.

In the way thus described we can construct a function p_i such that $p_i(k) = j_i$, for all $k \in \{1, \ldots, n\}$. This concludes the proof for the abstraction rule.

For the application case consider the rule

$$\frac{\begin{array}{c}\Gamma \vdash t^{\sigma\to\tau} : (\phi_1^\sigma \to \psi_{1,1}^\tau \wedge \ldots \wedge \phi_n^\sigma \to \psi_{1,n}^\tau) \vee \ldots \vee (\phi_1^\sigma \to \psi_{m,1}^\tau \wedge \ldots \wedge \phi_n^\sigma \to \psi_{m,n}^\tau) \\ \Gamma \vdash s^\sigma : \phi_1^\sigma \vee \ldots \vee \phi_n^\sigma\end{array}}{\Gamma \vdash (t^{\sigma\to\tau} s^\sigma)^\tau : \psi_{1,1}^\tau \vee \ldots \vee \psi_{1,n}^\tau \vee \ldots \vee \psi_{m,1}^\tau \vee \ldots \vee \psi_{m,n}^\tau} \text{[app]}$$

where all properties $\phi_1, \ldots, \phi_n$ and $\psi_{1,1}, \ldots, \psi_{m,n}$ are disjunction–free. The induction hypothesis for the premises of the rule is that

$$[\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma \subseteq [\![(\phi_1 \to \psi_{1,1} \wedge \ldots \wedge \phi_n \to \psi_{1,n}) \vee \ldots \vee (\phi_1 \to \psi_{m,1} \wedge \ldots \wedge \phi_n \to \psi_{m,n})]\!]$$

and

$$[\![s]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma \subseteq [\![\phi_1 \vee \ldots \vee \psi_n]\!]$$

We have to prove that $L = [\![ts]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma \subseteq [\![\psi_{1,1} \vee \ldots \vee \psi_{m,n}]\!] = R$. By the definition of the abstract semantics we know that $L = ([\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma) \odot ([\![s]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma) = \bigcup\{\vec{f}([\![s]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma) : f \in [\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma\}$. Thus it is enough to prove that for any $f \in [\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma$ we have $\vec{f}([\![s]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma) \subseteq R$.

In order to prove this take an arbitrary $f \in [\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma$. We know that $f \in [\![t]\!]_{\widehat{A}} \, \widehat{\rho}_\Gamma \subseteq [\![(\phi_1 \to \psi_{1,1} \wedge \ldots \wedge \phi_n \to \psi_{1,n}) \vee \ldots \vee (\phi_1 \to \psi_{m,1} \wedge \ldots \wedge \phi_n \to \psi_{m,n})]\!]$. This implies that for some k $f \in [\![\phi_1 \to \psi_{k,1} \wedge \ldots \wedge \phi_n \to \psi_{k,n}]\!]$. Thus for all $j \in \{1, \ldots, n\}$ we have $\vec{(f)} \subseteq [\![\psi_{k,j}]\!]$. But then $\forall j \in \{1, \ldots, n\} \vec{f} \subseteq [\![\psi_{1,1} \vee \ldots \vee \psi_{m,n}]\!]$. Therefore $\vec{f}([\![\phi_1 \vee \ldots \vee \psi_n]\!]) \subseteq [\![\psi_{1,1} \vee \ldots \vee \psi_{m,n}]\!]$. By the monotonicity of f we conclude that $\vec{f}([\![s]\!]_{\widehat{A}} \, \widehat{\rho_\Gamma}) \subseteq [\![\psi_{1,1} \vee \ldots \vee \psi_{m,n}]\!]$. This concludes the proof.

□

Theorem 5.2 (completeness of logic)
Let e be a term of type σ. Let $\widehat{\rho}$ be an environment mapping variables into values (from abstract domains) and Γ — an environment mapping variables into properties. Assume that for each variable x which is free in e we have $\widehat{\rho}(x) = [\![\Gamma(x)]\!]$. Then for any property $\phi \in \widehat{L}_\sigma$

$$[\![e]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![\phi]\!] \;\Rightarrow\; \Gamma \vdash e : \phi$$

Proof.
The proof is by induction on the structure of the term e. It is similar to the proof of completeness which Jensen gives for his system (see [1]).

The case of variable follows by assumption, for constants it is easy to verify.

Consider the case of application, i.e. let $e \equiv st$. By the induction hypothesis we know that for any properties Θ', Δ'

$$[\![s]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![\Theta']\!] \;\Rightarrow\; \Gamma \vdash s : \Theta'$$

and

$$[\![t]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![\Delta']\!] \;\Rightarrow\; \Gamma \vdash t : \Delta'$$

By assumption we also know that $[\![st]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![\varrho_1 \vee \ldots \vee \varrho_l]\!]$. We want to prove that $\Gamma \vdash st : \rho_1 \vee \ldots \vee \rho_l$.

By the definition of the abstract semantics we know that $[\![st]\!]_{\widehat{A}}\, \widehat{\rho} = ([\![s]\!]_{\widehat{A}}\, \widehat{\rho}) \odot ([\![t]\!]_{\widehat{A}}\, \widehat{\rho}) \subseteq [\![\varrho_1 \vee \ldots \vee \varrho_l]\!]$. By the definition of the semantics of properties we have

$$[\![s]\!]_{\widehat{A}}\, \widehat{\rho} = \downarrow\!\{f_1, \ldots, f_n\}$$

for certain functions $f_1, \ldots, f_n$. By Lemma 3.2 there exist properties $\theta_1, \ldots, \theta_n$ such that for any $i \in \{1, \ldots, n\}$ we have $[\![\theta_i]\!] = \downarrow\{f_i\}$. We can assume that each of θ_i is of the form $\phi_{1,i} \to \psi_{1,i} \wedge \ldots \wedge \phi_{n,i} \to \psi_{n,i}$ (to make the proof more readable we additionally assume that each property consists of the same number of conjunctions). Also there exists a property $\gamma_1 \vee \ldots \vee \gamma_m$ such that $[\![t]\!]_{\widehat{A}}\, \widehat{\rho} = [\![\gamma_1 \vee \ldots \vee \gamma_m]\!]$. By the assumption we know $[\![\gamma_i]\!] = \downarrow\{v_i\}$ that for all $i \in \{1, \ldots, m\}$ and also $\forall i \in \{1, \ldots, n\} \forall j \in \{1, \ldots, m\}\; f_i(v_j) \in [\![\varrho_1 \vee \ldots \vee \varrho_l]\!]$. This implies that $[\![s]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![(\gamma_1 \to \varrho_{p_1(1)} \wedge \ldots \wedge \gamma_m \to \varrho_{p_1(m)}) \vee \ldots \vee (\gamma_1 \to \varrho_{p_n(1)} \wedge \ldots \wedge \gamma_m \to \varrho_{p_n(m)})]\!]$, which, by induction hypothesis, concludes this case.

Consider the case of abstraction, i.e. $e \equiv \lambda x.s$ and suppose that $[\![\lambda x.s]\!]_{\widehat{A}}\, \widehat{\rho} \subseteq [\![\phi]\!]$. By the definition of the syntax and semantics of the properties we can assume that $[\![\phi]\!]$ is of the form $[\![(\phi_1 \to \psi_{1,1} \wedge \ldots \wedge \phi_n \to \psi_{1,n}) \vee \ldots \vee (\phi_1 \to \psi_{m,1} \wedge \ldots \wedge \phi_n \to \psi_{m,n})]\!]$, that is we have

$$\begin{aligned}
&[\![\lambda x.s]\!]_{\widehat{A}}\, \widehat{\rho} = \downarrow\!\{g \in A_{\sigma\to\tau} | \forall v \in A_\sigma g(v) \in [\![s]\!]_{\widehat{A}}\, (\widehat{\rho}[x \mapsto \downarrow\!\{v\}])\} \\
&\subseteq [\![(\phi_1 \to \psi_{1,1} \wedge \ldots \wedge \phi_n \to \psi_{1,n}) \vee \ldots \vee (\phi_1 \to \psi_{m,1} \wedge \ldots \wedge \phi_n \to \psi_{m,n})]\!]
\end{aligned}$$

Let us write $F = \downarrow\!\{f_1, \ldots, f_k\}$ for $[\![\lambda x.s]\!]_{\widehat{A}}\, \widehat{\rho}$ and Ψ for $\psi_{i,1} \vee \ldots \vee \psi_{i,n}$.

To conclude this case by the induction hypothesis we only need to prove that the inclusion $[\![s]\!]_{\widehat{A}}\, (\widehat{\rho}[x \mapsto [\![\phi_i]\!]]) \subseteq [\![\Psi]\!]$ holds for all $i \in \{1, \ldots, n\}$.

By the assumption we know that there is j such that $F \subseteq [\![\phi_1 \to \psi_{1,j} \wedge \ldots \wedge \phi_n \to \psi_{n,j}]\!]$. This implies that for all $f_i \in F$ and each $l \in \{1, \ldots, n\}$ we have $\vec{f_i}([\![\phi_l]\!]) \subseteq [\![\psi_{l,j}]\!] \subseteq [\![\Psi]\!]$. It is very easy to prove that for all $v \in A_\sigma$ the equality $F \odot (\downarrow\!\{v\}) = \downarrow\!\{f_1 v, \ldots, f_n v\} = [\![s]\!]_{\widehat{A}}\, (\widehat{\rho}[x \mapsto \downarrow\!\{v\}])$ holds, which implies that $F \odot [\![\phi_l]\!] \subseteq [\![\Psi]\!]$ for all $l \in \{1, \ldots, n\}$. But then also $[\![s]\!]_{\widehat{A}}\, (\widehat{\rho}[x \mapsto [\![\phi_l]\!]]) \subseteq [\![\Psi]\!]$ for all $l \in \{1, \ldots, n\}$. By the induction hypothesis this concludes the proof.

□

References

[1] T. P. Jensen, *Abstract Interpretation in Logical Form.* PhD thesis, Department of Computing, Imperial College, London, November 1992.

[2] P. N. Benton, *Strictness Analysis of Lazy Functional Programs.* PhD thesis, Computer Laboratory, University of Cambridge, August 1993. Also Technical Report 309.

[3] F. Barbanera, M. Dezani-Ciancaglini, U. De'Liguoro, *Intersection and union types: Syntax and semantics* Inf. Comput. 119, 202–230.

[4] P. N. Benton, *Strictness logic and polymorphic invariance*, in *Proceedings of 2nd International Symposium on Logical Foundations for Computer Science*, LNCS 620, Springer Verlag, Berlin 1992.

[5] S. Hunt, *Abstract interpretation of functional languages: From theory to practice*, PhD thesis, Imperial College, London, 1991.