Rank 3 Inhabitation of Intersection Types Revisited

Andrej Dudenhefner   Jan Bessai   Boris Düdder   Jakob Rehof

Technical University of Dortmund, Germany

May 20, 2016

Contents

Intersection Type System (BCD)

- Characterizes normalization/strong normalization in λ-calculus [Pot80]
- Characterizes finite function tables [Sal+12]
- Framework for the study of semantic domains for the λ-calculus

- Undecidable type checking
  (does the given term have the given type)
- Undecidable typability (w/o rule (ω))
  (does the given term have any type)
- Undecidable inhabitation [Urz99]
  (is there any term having the given type)

Intersection Type System (BCD)

Definition (Intersection Types $\mathbb{T}$)

$$\mathbb{T} \ni \sigma, \tau, \rho \ ::= \ a \mid \omega \mid \sigma \to \tau \mid \sigma \cap \tau \quad \text{where } a \in \mathbb{A}$$

Definition (Subtyping $\leq$)

Least preorder (reflexive and transitive relation) over $\mathbb{T}$ such that

$$\sigma \leq \omega, \quad \omega \leq \omega \to \omega, \quad \sigma \cap \tau \leq \sigma, \quad \sigma \cap \tau \leq \tau,$$
$$(\sigma \to \tau_1) \cap (\sigma \to \tau_2) \leq \sigma \to \tau_1 \cap \tau_2,$$
$$\text{if } \sigma \leq \tau_1 \text{ and } \sigma \leq \tau_2 \text{ then } \sigma \leq \tau_1 \cap \tau_2,$$
$$\text{if } \sigma_2 \leq \sigma_1 \text{ and } \tau_1 \leq \tau_2 \text{ then } \sigma_1 \to \tau_1 \leq \sigma_2 \to \tau_2$$

Intersection Type System (BCD)

Definition (Type Assignment)

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ (\mathrm{Ax}) \qquad \frac{}{\Gamma \vdash e : \omega}\ (\omega)$$

$$\frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \lambda x.e : \sigma \to \tau}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash e : \sigma \qquad \Gamma \vdash e : \tau}{\Gamma \vdash e : \sigma \cap \tau}\ (\cap\mathrm{I})$$

$$\frac{\Gamma \vdash e : \sigma \to \tau \qquad \Gamma \vdash e' : \sigma}{\Gamma \vdash (e\ e') : \tau}\ (\to\mathrm{E}) \qquad \frac{\Gamma \vdash e : \sigma \qquad \sigma \leq \tau}{\Gamma \vdash e : \tau}\ (\leq)$$

Intersection Type Inhabitation

Definition ($\vdash ? : \tau$)

Given a type τ is there a λ-term e such that $\vdash e : \tau$?

Definition (Rank [Lei83])

$$\begin{aligned}
\mathbf{rank}(\tau) &= 0 \text{ if } \tau \text{ is a simple type} \\
\mathbf{rank}(\sigma \to \tau) &= \max(\mathbf{rank}(\sigma) + 1, \mathbf{rank}(\tau)) \\
\mathbf{rank}(\sigma \cap \tau) &= \max(1, \mathbf{rank}(\sigma), \mathbf{rank}(\tau))
\end{aligned}$$

- $\vdash ? : \tau$ with $\mathbf{rank}(\tau) \leq 2$ is EXPSPACE-complete [Urz09]
- $\vdash ? : \tau$ with $\mathbf{rank}(\tau) \geq 3$ is undecidable [Urz09]

- Which features of BCD contribute to undecidability of inhabitation?
- Can BCD proof search simulate a Turing machine *directly*?
- Are there particular inhabitation instances which are hard to decide?

Approach in [BDS13] (Lambda Calculus with Types)

$\text{EQA} \leq \text{ETW} \leq \text{WTG} \leq \text{IHP}$ (15 pages w/o EQA theory)

- **EQA** Emptiness problem for queue automata
- **ETW** Emptiness problem for typewriter automata
- **WTG** Problem of winning a “tree game”
- **IHP** Intersection type inhabitation problem

[BDS13] Barendregt, Dekkers and Statman. “Lambda calculus with types”. Cambridge University Press, 2013.

Approach in [Sal+12; Loa01]

$$\text{WSTS} \leq \text{LDF} \leq \text{IHP} \quad \text{(7+3 pages)}$$

WSTS Word problem in semi-Thue systems

LDF λ-definability problem

IHP Intersection type inhabitation problem

[Loa01] Loader. "The undecidability of λ-definability". Logic, Meaning and Computation. Springer Netherlands, 2001.

[Sal+12] Salvati et al. "Loader and Urzyczyn are logically related". Automata, Languages, and Programming. Springer Berlin Heidelberg, 2012.

Approach in [Urz09]

$$\text{ELBA} \leq \text{SSTS1} \leq \text{HETM} \leq \text{IHP} \quad \text{(6 pages)}$$

ELBA Emptiness problem for linear bounded automata

SSTS1 Problem of deciding whether there is a word that can be rewritten to 1s in a simple semi-Thue system

HETM Halting problem for expanding tape machines

IHP Intersection type inhabitation problem

[Urz09] Urzyczyn. “Inhabitation of low-rank intersection types”. Typed Lambda Calculi and Applications. Springer Berlin Heidelberg, 2009.

Problematic aspects

- Introduced machinery is highly specialized
- Multiple degrees of non-determinism, alternation, parallelism
- Instructions create new instructions (higher order memory)
- λ-definability requires model theory
- Difficult to pinpoint necessary aspects

Goal

$$\mathrm{HTM} \leq \mathrm{IHP}$$

Intuition

$$\text{SSTS01} \leq \text{IHP}$$

Definition (Simple semi-Thue System, SSTS)

A semi-Thue system over an alphabet $\boldsymbol{\Sigma}$ is *simple*, if each rule has the form $\boldsymbol{ab} \Rightarrow \boldsymbol{cd}$ for some $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d} \in \boldsymbol{\Sigma}$.

Lemma (SSTS01)

Given a simple semi-Thue system over $\boldsymbol{\Sigma}$, it is undecidable whether $\exists \boldsymbol{n} \in \mathbb{N}.\boldsymbol{0}^{\boldsymbol{n}} \twoheadrightarrow \boldsymbol{1}^{\boldsymbol{n}}$

Simultaneous Set of Judgments

Proof search algorithm [Bun08] uses **Simultaneous Set of Judgments**[1]

$$\Gamma_1 \vdash ? : \tau_1, \ldots, \Gamma_n \vdash ? : \tau_n \quad \text{where } \mathbf{dom}(\Gamma_1) = \ldots = \mathbf{dom}(\Gamma_n)$$

with transformations such as

$\Gamma_1 \vdash ? : \tau_1, \quad \Gamma_2 \vdash ? : \sigma \cap \tau$

$\rightsquigarrow \Gamma_1 \vdash ? : \tau_1, \quad \Gamma_2 \vdash ? : \sigma, \quad \Gamma_2 \vdash ? : \tau$

$\Gamma_1 \vdash ? : \sigma_1 \rightarrow \tau_1, \quad \Gamma_2 \vdash ? : \sigma_2 \rightarrow \tau_2$

$\rightsquigarrow \Gamma_1 \cup \{x : \sigma_1\} \vdash ? : \tau_1, \quad \Gamma_2 \cup \{x : \sigma_2\} \vdash ? : \tau_2$ where x is fresh

$\Gamma_1 \vdash ? : \tau_1, \quad \Gamma_2 \vdash ? : \tau_2$

where $x : \sigma_1^1 \rightarrow \sigma_1^2 \rightarrow \tau_1 \in \Gamma_1$ and $x : \sigma_2^1 \rightarrow \sigma_2^2 \rightarrow \tau_2 \in \Gamma_2$

$\rightsquigarrow \Gamma_1 \vdash ? : \sigma_1^1, \quad \Gamma_2 \vdash ? : \sigma_2^1$ and $\Gamma_1 \vdash ? : \sigma_1^2, \quad \Gamma_2 \vdash ? : \sigma_2^2$

[1] logically same as Intersection Synchronous Logic [PRR12]

Simple semi-Thue System Simulation

Fix SSTS S over Σ with $l, r, \bullet \notin \Sigma$.

Let $\Gamma = \{z : 1\} \cup \{x_{ab \Rightarrow cd} : \sigma_{ab \Rightarrow cd} \mid ab \Rightarrow cd \in S\}$ where
$\sigma_{ab \Rightarrow cd} = (l \to c \to a) \cap (r \to d \to b) \cap \bigcap_{e \in \Sigma} (\bullet \to e \to e)$

Let

$\Gamma_1 = \Gamma,$	$\Gamma_2 = \Gamma,$	$\Gamma_3 = \Gamma,$	$\dots$	$\Gamma_{n-2} = \Gamma,$	$\Gamma_{n-1} = \Gamma,$	$\Gamma_n = \Gamma,$
$y_1 : l$	$y_1 : r$	$y_1 : \bullet$	$\dots$	$y_1 : \bullet$	$y_1 : \bullet$	$y_1 : \bullet$
$y_2 : \bullet$	$y_2 : l$	$y_2 : r$	$\dots$	$y_2 : \bullet$	$y_2 : \bullet$	$y_2 : \bullet$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$y_{n-2} : \bullet$	$y_{n-2} : \bullet$	$y_{n-2} : \bullet$	$\dots$	$y_{n-2} : l$	$y_{n-2} : r$	$y_{n-2} : \bullet$
$y_{n-1} : \bullet$	$y_{n-1} : \bullet$	$y_{n-1} : \bullet$	$\dots$	$y_{n-1} : \bullet$	$y_{n-1} : l$	$y_{n-1} : r$

Intuitively: $y : l$, $y : r$ in neighboring environments; $y : \bullet$ otherwise.

Simple semi-Thue System Simulation

$tabu \overset{ab\Rightarrow cd}{\Rightarrow} tcdu$ for $n = 4$ is simulated by

$\Gamma_1 \vdash ? : t, \quad \Gamma_2 \vdash ? : a, \quad \Gamma_3 \vdash ? : b, \quad \Gamma_4 \vdash ? : u$

using $x_{ab\Rightarrow cd} : (l \to c \to a) \cap (r \to d \to b) \cap \bigcap_{e \in \Sigma} (\bullet \to e \to e)$

$\Gamma_1 \vdash ? : t, \quad \Gamma_2 \vdash ? : c, \quad \Gamma_3 \vdash ? : d, \quad \Gamma_4 \vdash ? : u$

and $\Gamma_1 \vdash ? : \bullet, \quad \Gamma_2 \vdash ? : l, \quad \Gamma_3 \vdash ? : r, \quad \Gamma_4 \vdash ? : \bullet$

The second condition is satisfied iff l, r are inhabited in exactly the neighboring contexts.

Intuitively: type environments encode rewrite rule and order information; inhabited atoms encode current string.

Simple semi-Thue System Simulation

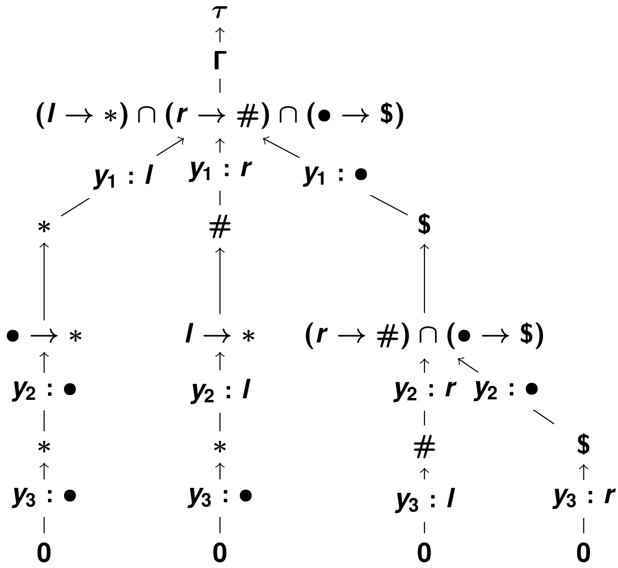
$\Gamma_1 \vdash ? : 1, \ldots, \Gamma_n \vdash ? : 1$ is satisfied since $z : 1 \in \Gamma_i$ for $1 \leq i \leq n$

Lemma

We have $0^n \twoheadrightarrow 1^n$ *iff* $\Gamma_1 \vdash ? : 0, \ldots, \Gamma_n \vdash ? : 0$ *is satisfied.*

Next: construct $\Gamma_1 \vdash ? : 0, \ldots, \Gamma_n \vdash ? : 0$ for arbitrary/unknown n

$$\sigma_* = ((\bullet \to *) \to *) \cap ((l \to *) \to \#) \cap ((r \to \#) \cap (\bullet \to \$) \to \$)$$
$$\sigma_0 = ((\bullet \to 0) \to *) \cap ((l \to 0) \to \#) \cap ((r \to 0) \to \$)$$
$$\tau = \sigma_* \to \sigma_0 \to 1 \to \sigma_{t_1} \to \ldots \to \sigma_{t_k} \to (l \to *) \cap (r \to \#) \cap (\bullet \to \$)$$



Relative Tags

- l left
- r right
- $\bullet$ other

Absolute Tags

- \$ last
- # next to last
- $*$ other

Back to the Goal

$$\text{HTM} \leq \text{IHP}$$

HTM $\leq$ IHP

Fix a TM $\boldsymbol{M} = (\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{q_0}, \boldsymbol{q_f}, \delta)$ where

- $\boldsymbol{\Sigma}$: finite set of tape symbols with $\sqcup \in \boldsymbol{\Sigma}$
- $\boldsymbol{Q}$: finite set of states with $\boldsymbol{q_0}, \boldsymbol{q_f} \in \boldsymbol{Q}$
- $\boldsymbol{q_0}$: initial state
- $\boldsymbol{q_f}$: final state
- $\delta : \boldsymbol{Q} \times \boldsymbol{\Sigma} \to \boldsymbol{Q} \times \boldsymbol{\Sigma} \times \{+1, -1\}$: transition function

Let $\mathbb{A} = \boldsymbol{\Sigma} \dot{\cup} \{l, r, \bullet\} \dot{\cup} \{\langle q, a \rangle \mid q \in Q, a \in \Sigma\} \dot{\cup} \{\circ, *, \#, \$\}$ The

configuration $(q, 3, abcd\sqcup\sqcup)$ is represented as $ab\langle q, c \rangle d\sqcup\sqcup$

TM simulation

TM simulation using most n tape cells by

$$\Gamma_1 \vdash ? : \langle q_0, \text{␣} \rangle, \quad \Gamma_2 \vdash ? : \text{␣}, \quad \ldots, \quad \Gamma_n \vdash ? : \text{␣}$$

where

$$\sigma_f = \bigcap_{a \in \Sigma} a \cap \bigcap_{a \in \Sigma} \langle q_f, a \rangle$$

for $t = ((q, c) \mapsto (q', c', +1)) \in \delta$

$$\sigma_t = \bigcap_{a \in \Sigma} (\bullet \to a \to a) \cap (l \to c' \to \langle q, c \rangle) \cap \bigcap_{a \in \Sigma} (r \to \langle q', a \rangle \to a)$$

for $t = ((q, c) \mapsto (q', c', -1)) \in \delta$

$$\sigma_t = \bigcap_{a \in \Sigma} (\bullet \to a \to a) \cap (r \to c' \to \langle q, c \rangle) \cap \bigcap_{a \in \Sigma} (l \to \langle q', a \rangle \to a)$$

$\Gamma_1, \ldots, \Gamma_n$ Initialization

$$\begin{aligned}
\sigma_* &= ((\bullet \to \circ) \to \circ) \cap ((\bullet \to *) \to *) \\
&\quad \cap ((l \to *) \to \#) \cap ((r \to \#) \cap (\bullet \to \$) \to \$) \\
\sigma_0 &= ((\bullet \to \langle q_0, \llcorner\!\lrcorner \rangle) \to \circ) \cap ((\bullet \to \llcorner\!\lrcorner) \to *) \\
&\quad \cap ((l \to \llcorner\!\lrcorner) \to \#) \cap ((r \to \llcorner\!\lrcorner) \to \$) \\
\tau_\star &= \sigma_0 \to \sigma_* \to \sigma_f \to \sigma_{t_1} \to \ldots \to \sigma_{t_k} \\
&\quad \to (l \to \circ) \cap (r \to \#) \cap (\bullet \to \$) \\
&\quad \text{where } \delta = \{t_1, \ldots, t_k\}
\end{aligned}$$

- $\circ$ marks the first symbol to be initialized to $\langle q_0, \llcorner\!\lrcorner \rangle$

Lemma

M halts starting with the empty tape iff there exists a λ-term e such that $\emptyset \vdash e : \tau_\star$

Insights

- “Neighboring” judgments recognized using $\boldsymbol{y} : \boldsymbol{l}$ and $\boldsymbol{y} : \boldsymbol{r}$
- TM simulation with fixed number of cells in rank 2 and order 2
- Inhabitant directly encodes computation
- Initialization requires only one $\boldsymbol{a} \cap \boldsymbol{b} \rightarrow \boldsymbol{c}$ type in the environment to increase the number of simultaneous judgments
- $\tau_\star$ is of rank 3 and order 3
- SSTS01 is convenient

Bibliography I

H.P. Barendregt, W. Dekkers, and R. Statman. *Lambda Calculus with Types*. Perspectives in Logic, Cambridge University Press, 2013.

Martin W. Bunder. "The Inhabitation Problem for Intersection Types." In: *Theory of Computing 2008. Proc. Fourteenth Computing: The Australasian Theory Symposium (CATS 2008), Wollongong, NSW, Australia, January 22-25, 2008. Proceedings*. Ed. by James Harland and Prabhu Manyem. Vol. 77. CRPIT. Australian Computer Society, 2008, pp. 7–14. ISBN: 978-1-920682-58-3. URL: http://crpit.com/abstracts/CRPITV77Bunder.html.

D. Leivant. "Polymorphic Type Inference." In: *Proc. 10th ACM Symp. on Principles of Programming Languages*. ACM. 1983, pp. 88–98.

Bibliography II

Ralph Loader. "The undecidability of λ-definability." In: *Logic, Meaning and Computation*. Springer, 2001, pp. 331–342.

G. Pottinger. "A Type Assignment for the Strongly Normalizable Lambda-Terms." In: *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Ed. by J. Hindley and J. Seldin. Academic Press, 1980, pp. 561–577.

Elaine Pimentel, Simona Ronchi Della Rocca, and Luca Roversi. "Intersection Types from a Proof-theoretic Perspective." In: *Fundam. Inform.* 121.1-4 (2012), pp. 253–274. DOI: 10.3233/FI-2012-778. URL: http://dx.doi.org/10.3233/FI-2012-778.

S. Salvati et al. "Urzyczyn and Loader are logically related." In: *Proceedings of ICALP 2012*. Vol. 7392. LNCS. Springer, 2012, pp. 364–376.

Bibliography III

- P. Urzyczyn. "Inhabitation of Low-Rank Intersection Types." In: *Proceedings of TLCA'09*. Vol. 5608. LNCS. Springer, 2009, pp. 356–370.
- P. Urzyczyn. "The Emptiness Problem for Intersection Types." In: *Journal of Symbolic Logic* 64.3 (1999), pp. 1195–1215.