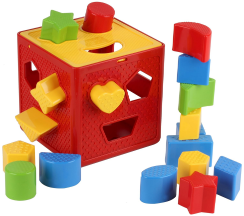
Extending the Quantitative Pattern-Matching Paradigm

Sandra Alves **Miguel Ramos** Delia Kesner

APLAS'24, Kyoto, Japan

Pattern-Matching



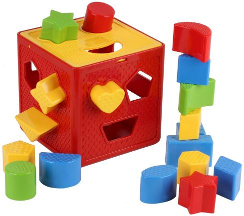
Efficient way of decomposing and processing data.

Available in most programming languages and proof assistants.

The λ-calculus is a good tool to study programming languages [Landin, 1965], however...

- Hard to encode pattern-matching into the (pure) λ-calculus.
- Not easy to generalize properties of the λ-calculus to pattern-matching.

Pattern-Matching



It is necessary to study pattern-matching directly:

Operationally [Huet and Lévy, 1991, Sekar and Ramakrishnan, 1993, Cirstea and Kirchner, 2001, Kahl, 2004, Jay and Kesner, 2009, Khasidashvili, 1990].

Denotationally [Cirstea and Kirchner, 2001, Kahl, 2004, Jay and Kesner, 2009, Bucciarelli et al., 2015, Accattoli and Barras, 2017, Alves et al., 2018, Barenbaum et al., 2018, Alves et al., 2020, Bucciarelli et al., 2021].

Quantitative Semantics

vs.

Quantitative semantics allow us to reason quantitatively about programs:

- λ-Calculi [Accattoli and Guerrieri, 2018, Accattoli et al., 2020, Accattoli and Guerrieri, 2022, de Carvalho, 2018, Kesner and Viso, 2022].
- Classical Calculi [Kesner and Vial, 2020, Santo et al., 2024].
- Effects [Lago et al., 2021, Alves et al., 2023].
- Pattern-Matching [Bucciarelli et al., 2015, Alves et al., 2020, Bucciarelli et al., 2021].

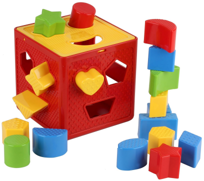
Related Work

In previous works:

Pattern-matching only over pairs
[Bucciarelli et al., 2015, Alves et al., 2020, Bucciarelli et al., 2021].

No quantitative semantics [Cirstea and Kirchner, 2001, Kahl, 2004, Jay and Kesner, 2009, Accattoli and Barras, 2017, Barenbaum et al., 2018].

In this work...



and

...we provide quantitative semantics for pattern-matching over data.

Pattern-Matching in Haskell

```haskell
data TProd a b = P a b -- Product Types

fst :: TProd a b -> a
fst (P x y) = x

data TSum a b = L a | R b -- Sum Types

swap :: TSum a b -> TSum b a
swap s = case s of L x -> R x
                   R x -> L x
```

The Pattern-Matching Calculus

Patterns p, q

Variables	$x, y, \ldots$
Data	$\mathsf{c}(p_1, \ldots, p_n)$

Terms t, u

Variables	$x, y, \ldots$
λ-abstractions	$\lambda p.t$
Applications	$t\ u$
Matching	$t[p \setminus u]$
Cases	$\texttt{case}\ t\ \texttt{of}\ (p_1.u_1, \ldots, p_n.u_n)$
Data	$\mathsf{c}(t_1, \ldots, t_n)$

The Pattern-Matching Calculus

Patterns p, q

Variables	$x, y, \ldots$
Data	$\mathtt{c}(p_1, \ldots, p_n)$

Terms t, u

Variables	$x, y, \ldots$
λ-abstractions	$\lambda p.t$
Applications	$t\ u$
Matching	$t[p \setminus u]$
Cases	$\mathtt{case}\ t\ \mathtt{of}\ (p_1.u_1, \ldots, p_n.u_n)$
Data	$\mathtt{c}(t_1, \ldots, t_n)$

$$\mathtt{fst} \equiv \lambda \mathtt{P}(x, y).x$$
$$\mathtt{swap} \equiv \lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))$$

Operational Semantics

Weak Head Evaluation*

$$\underline{(\lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})}$$

Operational Semantics

Weak Head Evaluation*

$$\underline{(\lambda s.\texttt{case}\ s\ \texttt{of}\ (\texttt{L}(x).\texttt{R}(x), \texttt{R}(x).\texttt{L}(x)))}\ \underline{\texttt{R}(\texttt{Id})} \quad \texttt{-- } \textit{beta}$$

$$\to \quad \texttt{case}\ \underline{s}\ \texttt{of}\ (\texttt{L}(x).\texttt{R}(x), \texttt{R}(x).\texttt{L}(x))\underline{[s \setminus \texttt{R}(\texttt{Id})]}$$

Operational Semantics

Weak Head Evaluation*

$$
\begin{array}{ll}
& \underline{(\lambda s.\texttt{case}\ s\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})} \quad \texttt{-- beta} \\
\to & \texttt{case}\ \underline{s}\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))\underline{[s \setminus \mathtt{R}(\mathtt{Id})]} \quad \texttt{-- substitute} \\
\to & \texttt{case}\ \underline{\mathtt{R}(\mathtt{Id})}\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \underline{\mathtt{R}(x)}.\mathtt{L}(x))
\end{array}
$$

Operational Semantics

Weak Head Evaluation*

$$
\begin{aligned}
& \underline{(\lambda s.\texttt{case}\ s\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})} && \text{-- beta} \\
\to\ & \texttt{case}\ \underline{s}\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))\underline{[s \setminus \mathtt{R}(\mathtt{Id})]} && \text{-- substitute} \\
\to\ & \texttt{case}\ \underline{\mathtt{R}(\mathtt{Id})}\ \texttt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \underline{\mathtt{R}(x)}.\mathtt{L}(x)) && \text{-- match} \\
\to\ & \mathtt{L}(\underline{x})\underline{[x \setminus \mathtt{Id}]}
\end{aligned}
$$

Operational Semantics

Weak Head Evaluation*

$$
\begin{aligned}
& \underline{(\lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})} && \text{-- beta} \\
\rightarrow\ & \mathtt{case}\ \underline{s}\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))\underline{[s \setminus \mathtt{R}(\mathtt{Id})]} && \text{-- substitute} \\
\rightarrow\ & \mathtt{case}\ \underline{\mathtt{R}(\mathtt{Id})}\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \underline{\mathtt{R}(x)}.\mathtt{L}(x)) && \text{-- match} \\
\rightarrow\ & \mathtt{L}(\underline{x})\underline{[x \setminus \mathtt{Id}]} && \text{-- substitute} \\
\rightarrow\ & \mathtt{L}(\mathtt{Id})
\end{aligned}
$$

Operational Semantics

Weak Head Evaluation*

$$
\begin{aligned}
& \underline{(\lambda s.\mathsf{case}\ s\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})} && \texttt{-- } \textit{beta} \\
\to\ & \mathsf{case}\ \underline{s}\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))\underline{[s \setminus \mathtt{R}(\mathtt{Id})]} && \texttt{-- } \textit{substitute} \\
\to\ & \mathsf{case}\ \underline{\mathtt{R}(\mathtt{Id})}\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \underline{\mathtt{R}(x)}.\mathtt{L}(x)) && \texttt{-- } \textit{match} \\
\to\ & \mathtt{L}(\underline{x})\underline{[x \setminus \mathtt{Id}]} && \texttt{-- } \textit{substitute} \\
\to\ & \mathtt{L}(\mathtt{Id}) && \texttt{-- } \textit{done!}
\end{aligned}
$$

Operational Semantics

Weak Head Evaluation*

$$
\begin{array}{lll}
 & \underline{(\lambda s.\mathsf{case}\ s\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))}\ \underline{\mathtt{R}(\mathtt{Id})} & \texttt{-- beta} \\
\to & \mathsf{case}\ \underline{s}\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x))\underline{[s \setminus \mathtt{R}(\mathtt{Id})]} & \texttt{-- substitute} \\
\to & \mathsf{case}\ \underline{\mathtt{R}(\mathtt{Id})}\ \mathsf{of}\ (\mathtt{L}(x).\mathtt{R}(x), \underline{\mathtt{R}(x)}.\mathtt{L}(x)) & \texttt{-- match} \\
\to & \mathtt{L}(\underline{x})\underline{[x \setminus \mathtt{Id}]} & \texttt{-- substitute} \\
\to & \mathtt{L}(\mathtt{Id}) & \texttt{-- done!}
\end{array}
$$

*CBV-like with respect to data patterns and CBN-like with respect variable patterns.

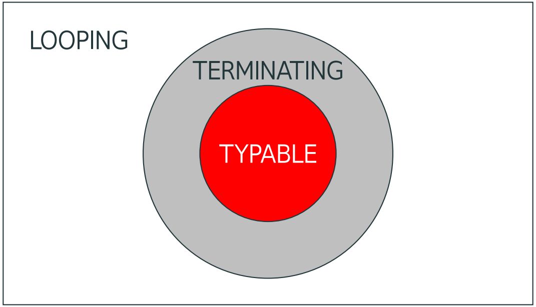
Operational Semantics

We are able to encode the (weak) CBV and CBN version of the λ-calculus.

Types: Simple vs. Intersection

Simple Types

$$A, B ::= b \mid A \to B$$

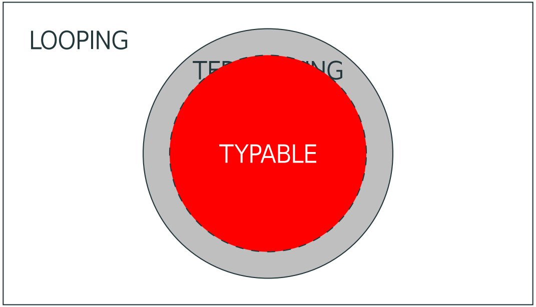


$\vdash \lambda x.x : A \to A$ $\qquad$ $\nvdash \lambda x.xx$ $\qquad$ $\nvdash (\lambda x.xx)(\lambda x.xx)$

Types: Simple vs. Intersection

Intersection Types

$$A, B ::= b \mid A \to B \mid A \cap B$$



$\vdash \lambda x.x : A \to A$

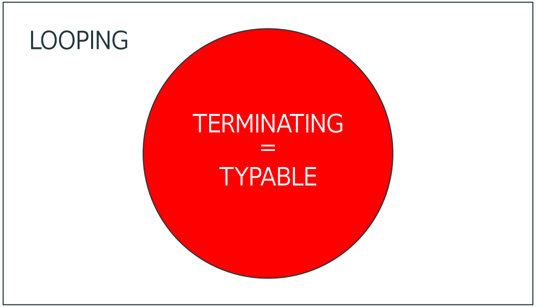
$\nvdash \lambda x.xx$

$\nvdash (\lambda x.xx)(\lambda x.xx)$

Types: Simple vs. Intersection

Intersection Types

$$A, B ::= b \mid A \to B \mid A \cap B$$



$\vdash \lambda x.x : A \to A$

$\vdash \lambda x.xx : ((A \to B) \cap A) \to B$

$\nvdash (\lambda x.xx)(\lambda x.xx)$

Intersection Types

Idempotent vs. Non-Idempotent

Intersection Types

Idempotent	vs.	Non-Idempotent
[Coppo and Dezani-Ciancaglini, 1978]		[Gardner, 1994, Kfoury and Wells, 1999] A flavor of [Girard, 1987]'s Linear Logic

Intersection Types

Idempotent	vs.	Non-Idempotent
[Coppo and Dezani-Ciancaglini, 1978]		[Gardner, 1994, Kfoury and Wells, 1999] A flavor of [Girard, 1987]'s Linear Logic
Associativity, Commutativity and $A \cap A \sim A$		Associativity, Commutativity but $A \cap A \not\sim A$

Intersection Types

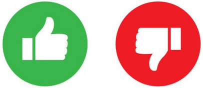
Idempotent	vs.	Non-Idempotent
[Coppo and Dezani-Ciancaglini, 1978]		[Gardner, 1994, Kfoury and Wells, 1999] A flavor of [Girard, 1987]'s Linear Logic
Associativity, Commutativity and $A \cap A \sim A$		Associativity, Commutativity but $A \cap A \not\sim A$
$A \cap A \cap B$ is set $\{A, B\}$		$A \cap A \cap B$ is multiset $[A, A, B]$

Intersection Types

Idempotent	vs.	Non-Idempotent
[Coppo and Dezani-Ciancaglini, 1978]		[Gardner, 1994, Kfoury and Wells, 1999] A flavor of [Girard, 1987]'s Linear Logic
Associativity, Commutativity and $A \cap A \sim A$		Associativity, Commutativity but $A \cap A \not\sim A$
$A \cap A \cap B$ is set $\{A, B\}$		$A \cap A \cap B$ is multiset $[A, A, B]$
"$1 + 1 = 1$"		"$1 + 1 = 2$"

Intersection Types

Idempotent	vs.	Non-Idempotent
[Coppo and Dezani-Ciancaglini, 1978]		[Gardner, 1994, Kfoury and Wells, 1999] A flavor of [Girard, 1987]'s Linear Logic
Associativity, Commutativity and $A \cap A \sim A$		Associativity, Commutativity but $A \cap A \not\sim A$
$A \cap A \cap B$ is set $\{A, B\}$		$A \cap A \cap B$ is multiset $[A, A, B]$
"$1 + 1 = 1$"		"$1 + 1 = 2$"
Qualitative Type Systems		Quantitative Type Systems



Yes or No

Bounds and Exact Measures

[De Carvalho, 2007, de Carvalho, 2018]

Intersection Types

Idempotent vs. Non-Idempotent

[(

REMARK

These type systems are NOT for programming!

They are equivalent to models of computation.

$\Downarrow$

Completely syntactical tools for reasoning about the denotation of terms in those models.

Yes or No

Bounds and Exact Measures

[De Carvalho, 2007, de Carvalho, 2018]

Main Result

Quantitative Characterization of Weak Head-Termination

WEAK HEAD-TERMINATION in n-steps $\Leftrightarrow$ TYPABILITY with derivation of at least size n

Main Result

Quantitative Characterization of Weak Head-Termination

WEAK HEAD-TERMINATION in n-steps $\Leftrightarrow$ TYPABILITY with derivation of at least size n

Proof

Relies on quantitative versions of the usual subject reduction and subject expansion lemmas.

Example

Recall that...

$$(\lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))\ \mathtt{R}(\mathtt{Id}) \to^4 \mathtt{L}(\mathtt{Id})$$

...weak head-terminates in 4 steps.

Example

We can build the following type derivation for it…

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{}{s{:}[\mathtt{R}([\mathtt{abs}])] \vdash s{:}\mathtt{R}([\mathtt{abs}])}\,(\mathtt{var})
    }{s{:}[\mathtt{R}([\mathtt{abs}])] \vdash s{:}[\mathtt{R}([\mathtt{abs}])]}
    \quad
    \dfrac{
      \dfrac{}{x{:}[\mathtt{abs}] \Vdash x{:}[\mathtt{abs}]}\,((\mathtt{var}))
    }{x{:}[\mathtt{abs}] \Vdash \mathtt{R}(x){:}[\mathtt{R}([\mathtt{abs}])]}\,((\mathtt{data}))
    \quad
    \dfrac{
      \dfrac{
        \dfrac{}{x{:}[\mathtt{abs}] \vdash x{:}\mathtt{abs}}\,(\mathtt{var})
      }{x{:}[\mathtt{abs}] \vdash x{:}[\mathtt{abs}]}
    }{x{:}[\mathtt{abs}] \vdash \mathtt{L}(x){:}\mathtt{L}([\mathtt{abs}])}\,(\mathtt{data})
  }{s{:}[\mathtt{R}([\mathtt{abs}])] \vdash \mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x),\mathtt{R}(x).\mathtt{L}(x)){:}\mathtt{L}([\mathtt{abs}])}\,(\mathtt{case})
  }{\vdash \lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x),\mathtt{R}(x).\mathtt{L}(x)){:}[\mathtt{R}([\mathtt{abs}])] \to \mathtt{L}([\mathtt{abs}])}\,(\mathtt{abs})
  \quad
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{\vdash \mathtt{Id}{:}\mathtt{abs}}\,(\mathtt{abs}^\star)
      }{\vdash \mathtt{R}(\mathtt{Id}){:}[\mathtt{abs}]}
    }{\vdash \mathtt{R}(\mathtt{Id}){:}\mathtt{R}([\mathtt{abs}])}\,(\mathtt{data})
  }{\vdash \mathtt{R}(\mathtt{Id}){:}[\mathtt{R}([\mathtt{abs}])]}
}{\Phi \triangleright \vdash (\lambda s.\mathtt{case}\ s\ \mathtt{of}\ (\mathtt{L}(x).\mathtt{R}(x), \mathtt{R}(x).\mathtt{L}(x)))\ \mathtt{R}(\mathtt{Id}) : \mathtt{L}([\mathtt{abs}])}\,(\mathtt{app})
$$

…of size 10.

(the number of rules)

Future Work

This work is (mostly) foundational, so there are a lot of ideas left to explore…

- Obtaining exact measures
- Adding a fixpoint operator
- Considering CBNeed operational semantics
- Allowing nondeterministic matching
- Allowing free variables and considering strong evaluation
- Considering applications to study algebraic effects and handlers

The End

References i

- Accattoli, B. and Barras, B. (2017).
  The negligible and yet subtle cost of pattern matching.
  In Chang, B. E., editor, *Programming Languages and Systems - 15th Asian Symposium, APLAS 2017, Suzhou, China, November 27-29, 2017, Proceedings*, volume 10695 of *Lecture Notes in Computer Science*, pages 426–447. Springer.
- Accattoli, B., Graham-Lengrand, S., and Kesner, D. (2020).
  Tight typings and split bounds, fully developed.
  J. Funct. Program., 30:e14.

References ii

- Accattoli, B. and Guerrieri, G. (2018).
  Types of fireballs.
  In Ryu, S., editor, *Programming Languages and Systems - 16th Asian Symposium, APLAS 2018, Wellington, New Zealand, December 2-6, 2018, Proceedings*, volume 11275 of *Lecture Notes in Computer Science*, pages 45–66. Springer.
- Accattoli, B. and Guerrieri, G. (2022).
  The theory of call-by-value solvability.
  Proc. ACM Program. Lang., 6(ICFP):855–885.
- Alves, S., Dundua, B., Florido, M., and Kutsia, T. (2018).
  Pattern-based calculi with finitary matching.
  Log. J. IGPL, 26(2):203–243.

References iii

- Alves, S., Kesner, D., and Ramos, M. (2023).
  Quantitative global memory.
  In Hansen, H. H., Scedrov, A., and de Queiroz, R. J. G. B., editors, *Logic, Language, Information, and Computation - 29th International Workshop, WoLLIC 2023, Halifax, NS, Canada, July 11-14, 2023, Proceedings*, volume 13923 of *Lecture Notes in Computer Science*, pages 53–68. Springer.
- Alves, S., Kesner, D., and Ventura, D. (2020).
  A Quantitative Understanding of Pattern Matching.
  In Bezem, M. and Mahboubi, A., editors, *25th International Conference on Types for Proofs and Programs (TYPES 2019)*, volume 175 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:36, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

References iv

- Barenbaum, P., Bonelli, E., and Mohamed, K. (2018).
  Pattern Matching and Fixed Points: Resource Types and Strong Call-By-Need: Extended Abstract.
  In *Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming*, pages 1–12, Frankfurt am Main Germany. ACM.
- Bucciarelli, A., Kesner, D., and Rocca, S. R. D. (2015).
  Observability for pair pattern calculi.
  In Altenkirch, T., editor, *13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015, July 1-3, 2015, Warsaw, Poland*, volume 38 of *LIPIcs*, pages 123–137. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Bucciarelli, A., Kesner, D., and Rocca, S. R. D. (2021).
  Solvability = typability + inhabitation.
  Log. Methods Comput. Sci., 17(1).

References v

- Cirstea, H. and Kirchner, C. (2001).
  The rewriting calculus - part I.
  Log. J. IGPL, 9(3):339–375.
- Coppo, M. and Dezani-Ciancaglini, M. (1978).
  A new type assignment for λ-terms.
  Arch. Math. Log., 19(1):139–156.
- De Carvalho, D. (2007).
  ***Sémantiques de la logique linéaire et temps de calcul*.**
  PhD thesis, Aix-Marseille 2.
- de Carvalho, D. (2018).
  Execution time of λ-terms via denotational semantics and intersection types.
  Math. Struct. Comput. Sci., 28(7):1169–1203.

References vi

- Gardner, P. (1994).
  Discovering needed reductions using type theory.
  In *TACS*.
- Girard, J.-Y. (1987).
  Linear logic.
  Theoretical Computer Science, 50(1):1–101.
- Huet, G. and Lévy, J.-J. (1991).
  Computations in orthogonal rewriting systems - Part I and Part II.
  In Lassez, J.-L. and Plotkin, G., editors, *Computational Logic, Essays in Honor of Alan Robinson*, pages 395–443. MIT Press.
- Jay, C. B. and Kesner, D. (2009).
  First-class patterns.
  J. Funct. Program., 19(2):191–225.

References vii

- Kahl, W. (2004).
  Basic pattern matching calculi: a fresh view on matching failure.
  In Kameyama, Y. and Stuckey, P. J., editors, *Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Nara, Japan, April 7-9, 2004, Proceedings*, volume 2998 of *Lecture Notes in Computer Science*, pages 276–290. Springer.
- Kesner, D. and Vial, P. (2020).
  Consuming and persistent types for classical logic.
  In Hermanns, H., Zhang, L., Kobayashi, N., and Miller, D., editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 619–632. ACM.

References viii

- Kesner, D. and Viso, A. (2022).
  Encoding tight typing in a unified framework.
  In Manea, F. and Simpson, A., editors, *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, volume 216 of *LIPIcs*, pages 27:1–27:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Kfoury, A. J. and Wells, J. B. (1999).
  Principality and decidable type inference for finite-rank intersection types.
  In Appel, A. W. and Aiken, A., editors, *POPL '99, Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Antonio, TX, USA, January 20-22, 1999*, pages 161–174. ACM.
- Khasidashvili, Z. (1990).
  Expression reduction systems.
  In *Proceedings of IN Vekua Institute of Applied Mathematics*, volume 36, Tbilisi.

References ix

- Lago, U. D., Faggian, C., and Rocca, S. R. D. (2021).
  Intersection types and (positive) almost-sure termination.
  Proc. ACM Program. Lang., 5(POPL):1–32.
- Landin, P. J. (1965).
  A correspondence between ALGOL 60 and church's lambda-notations: Part II.
  Commun. ACM, 8(3):158–167.
- Santo, J. E., Kesner, D., and Peyrot, L. (2024).
  A faithful and quantitative notion of distant reduction for the lambda-calculus with generalized applications.
  Log. Methods Comput. Sci., 20(3).
- Sekar, R. C. and Ramakrishnan, I. V. (1993).
  Programming in equational logic: Beyond strong sequentiality.
  Inf. Comput., 104(1):78–109.

References x