(Re)Painting (and Polarizing) Arrow Types

Adrienne Lancelot, Giulio Manzonetto, Guy McCusker, and Gabriele Vanoni

The relational semantics of linear logic may be presented on λ-terms via intersection type systems. Interpretation in these models induces a λ-theory where terms which can be typed in all of the same typing context are equivalent:

$$t \equiv^{\mathrm{typ}} u \text{ if } \forall(\Gamma, \mathsf{T}),\ \Gamma \vdash t : \mathsf{T} \iff \Gamma \vdash u : \mathsf{T}$$

Extensional Interpretation. Interpreting programs in denotational semantics often abstracts away the implementation details and focuses on the actual meaning of program phrases. Relational semantics may appear to follow this extensional collapse, as equivalent programs are exactly those which are interchangeable in type derivations.

This *extensionality* in program equivalences is made formal via observational equivalence: two programs are observationally equivalent if they cannot be distinguished by putting them in an execution environment. Typically, one *observes* termination (or non-termination) for some evaluation strategy, which matches well with intersection types, where typability may characterize termination.

Soudness of semantics induced by intersection type systems are shown by the two following properties:

1. *Adequacy:* typability is equivalent to observability (usually termination);
2. *Compositionality:* if $t \equiv^{\mathrm{typ}} u$ then $C\langle t\rangle \equiv^{\mathrm{typ}} C\langle u\rangle$ for any context C.

Still, relational semantics are often not complete, hence not *fully abstract* for observational equivalence: some observationally equivalent programs are separated by a typing context. This can be understood as the fact that intersection types allow for some non-determinism while pure contexts do not.

Intensional Information. The mismatch between relational semantics and observational equivalence can also be understood as the fact that some intensional or quantitative information is present in relational semantics. Indeed, non-idempotent intersection type systems are able to provide complexity analyses of λ-terms, yielding evaluation time bounds, as pioneered by de Carvalho [8] and later refined by Accattoli et al. [2], as well as execution space bounds [1]. This quantitative information, for the most part, only appears at the level of type derivations—the construction of the denotation of a term—and are invisible in the final denotation and hence in the (in)equational theory[1]. However, as this equational theory in general does not coincide with the observational equivalence, the following questions arise naturally:

What is the (in)equational theory induced by these semantics?
And is it cost-aware as the type system suggests?

[1] We are interested in program equivalences and preorders, *i.e.* equational and inequational theories.

We answer these questions in recent works introducing a cost-aware observational equivalence [3] and refine the analysis to a cost-aware observational improvement preorder [15]. In this talk, we will focus on the second contribution of a cost-aware observational improvement preorder, which characterizes the semantic preorder induced by head intersection type systems as well as the syntactic Böhm tree preorder up to η-reductions. The objective of the talk will be to describe a new compositionality proof technique we developed for a variant of relational semantics.

Interaction Equivalence and Improvement. Briefly, two programs are **interaction equivalent** if they cannot be distinguished by putting them in an execution environment, neither in terms of termination nor in the number of computation steps between the program and the environment. **Interaction improvement** is the asymmetric variant which allows a program to *improve* another one by either terminating in more environments or terminating using less communication with the environment.

Defining interaction equivalence (and improvement) requires to separate internal steps (local computation within the program or within the environment) from interaction ones (communication between the program and its execution environment).

To achieve the distinction between those interaction steps and internal steps, we use the *checkers calculus* of [3] which is a bichromatic version of the untyped λ-calculus. Checkers terms are color-annotated λ-terms, where the abstraction and the application constructors are tagged by a color, either black or white. Intuitively, programs are players uniformly painted black, and contexts are opponents uniformly painted white. As a term is executed in a context, the black and white parts meet and mix. This allows us to define notions of quantitative observational equivalence, preorder and *interaction improvement*, akin to Sands's improvement [20], but counting only computation steps that involve a white constructor interacting with a black one.

Characterizing The Costs at Play Behind Relational Semantics. Interaction improvement makes explicit, in particular, the quantitative aspects underlying the preorder induced by the relational model of [11]. The interpretation in this model can be described via a type system based on multi types, also known as non-idempotent intersection types [17,7]. The preorder induced by this model has been characterized as the Böhm tree preorder up to η-reductions [5]. However, the execution costs involved in this relational model have so far remained unclear.

In this work, we introduce a colored relational semantics for the checkers calculus and define an ordering on interpretations that compares their elements in a refined manner which is both qualitative and quantitative. Our main contribution is to show that, when restricted to regular λ-terms, this ordering coincides with both the relational preorder—equivalently, with the Böhm tree preorder up to η-reductions—and interaction improvement.

It turns out to be quite challenging to show that the ordering we introduce on interpretations is *compositional*. Though denotational approaches are typically compositional by definition, the fact that it presents difficulties here is perhaps not surprising given the intensional information we are tracking: similar challenges arise in intensional models such as game semantics [12,13] and operational approaches such as applicative and normal form bisimulations [4,10] (or so-called operational game semantics [16,14]). Here we establish compositionality using a novel technique which bears comparison to game semantics.

Our relational semantics annotates the ordinary relational model in two ways: numerical annotations track the steps needed to reduce a term, while color annotations on the types track the colorings of terms and their potential contexts. An unannotated entry in the relational semantics of a term may have several valid annotations in our system. *Improvements* result not only from lower numerical annotations, but also from better matching of color annotations between a term and its context. The delicate interplay between colors and counts requires a careful analysis. Once the appropriate preorder has been defined (Definition 6), we will explain in the talk how to prove that it is *compositional*. To prove this property, we treat typings dynamically: we study how modifications to the context colorings of a typing affect the corresponding typings of a term—a process we call *repainting*. When an improved term is placed in a context, these color changes propagate between the term and its context, in a manner reminiscent of the composition of strategies in game semantics. Once this propagation stabilizes, a new typing emerges, and the preorder is preserved.

Talk. This document presents the intersection type system; explaining the compositionality proof of the preorder of Definition 6 will be the goal of the talk.

1 The Checkers Calculus

We briefly present the *checkers calculus* from [3], a bichromatic variant of the λ-calculus designed to count the reductions that occur during the interaction between a λ-term (the *player*) and a context (the *opponent*).

Operational Semantics. In the checkers calculus, each application $\cdot^{\mathsf{c}}$ and each abstraction λ_{c} is assigned a color $\mathsf{c} \in \{\circ, \bullet\}$, either white or black, depending on whether the constructor 'belongs' to the player or the opponent. We use $\mathsf{c}^{\perp}$ to denote the opposite color, namely $\circ^{\perp} := \bullet$ and $\bullet^{\perp} := \circ$.

Definition 1. *The set $\Lambda_{\circ\bullet}$ of* checkers terms *is inductively defined as follows:*

$$\Lambda_{\circ\bullet} \quad \ni \quad t, u, s ::= x \mid \lambda_{\bullet} x.t \mid \lambda_{\circ} x.t \mid t \bullet u \mid t \circ u$$

There are two kinds of colored β-redexes $(\lambda_{\mathsf{c}} x.t) \cdot^{\mathsf{d}} u$:

$$\begin{array}{rl} \textsc{Silent} & (\lambda_{\mathsf{c}} x.t) \cdot^{\mathsf{c}} u \mapsto_{\beta_{\tau}} t\{x := u\} \\ \textsc{Interaction} & (\lambda_{\mathsf{c}} x.t) \cdot^{\mathsf{c}^{\perp}} u \mapsto_{\beta_{◐}} t\{x := u\} \end{array}$$

Silent redexes are β-redexes where the color c of the abstraction matches the color d of the application. Intuitively, these steps are internal to each player's world. In interaction redexes, instead, the color of the abstraction and the color of the application are different, *i.e.* $\mathsf{c} \neq \mathsf{d}$.

Given $t \in \Lambda_{\circ\bullet}$, we write $\Downarrow^{\LEFTcircle k}_{\mathtt{h}_{\circ\bullet}}$ whenever t has a head normal form h, and the head reduction sequence $t \to_{\mathtt{h}_{\circ\bullet}} \cdots \to_{\mathtt{h}_{\circ\bullet}} h$ contains k interaction steps $\to_{\mathtt{h}\LEFTcircle}$.

Definition 2 (Checkers interaction improvement and equivalence [3]). *The* interaction improvement (preorder) $\sqsubseteq^{\mathrm{imp}}_{\LEFTcircle} \subseteq \Lambda_{\circ\bullet} \times \Lambda_{\circ\bullet}$ *is defined as:*

$$t \sqsubseteq^{\mathrm{imp}}_{\LEFTcircle} u \textit{ if } \forall C \in \mathcal{C}_{\circ\bullet}, k \in \mathbb{N}\,.\,[\,C\langle t\rangle \Downarrow^{\LEFTcircle k}_{\mathtt{h}_{\circ\bullet}} \Rightarrow C\langle u\rangle \Downarrow^{\LEFTcircle k'}_{\mathtt{h}_{\circ\bullet}} \ \textit{ for some } k' \leq k].$$

Example 1. Consider $\mathtt{I}_\bullet \coloneqq \lambda_\bullet x.x$ and $\mathtt{1}_\bullet \coloneqq \lambda_\bullet x.\lambda_\bullet y.x \bullet y$. We have that $\mathtt{1}_\bullet \not\sqsubseteq^{\mathrm{int}}_{\LEFTcircle} \mathtt{I}_\bullet$, as they are separated by $C = \langle\cdot\rangle \circ z \circ w$, but $\mathtt{1}_\bullet \sqsubseteq^{\mathrm{imp}}_{\LEFTcircle} \mathtt{I}_\bullet$ holds.

2 Checkers Multi Types

Our multi type system is an annotated version of the relational semantics of the λ-calculus [6,5], in which arrows are decorated with a color as in $\lambda_{\mathsf{c}} x.t : \mathsf{M} \xrightarrow{\mathsf{c}} \mathsf{L}$.

Definition 3. *1.* Linear types L, L' *and* multi types M, N *are generated by:*

$$\begin{array}{lll} \textsc{Linear types} & \mathsf{L}, \mathsf{L}' ::= \mathsf{A} \mid \mathsf{M} \xrightarrow{\bullet} \mathsf{L} \mid \mathsf{M} \xrightarrow{\circ} \mathsf{L} & \textit{where } \mathsf{A} \textit{ is an atom} \\ \textsc{Multi types} & \mathsf{M}, \mathsf{N} ::= [\mathsf{L}_1, \ldots, \mathsf{L}_n] & n \geq 0 \end{array}$$

2. Typing derivations $\pi : \Gamma \vdash^{k}_{\LEFTcircle} t : \mathsf{T}$*, are obtained by applying the rules:*

$$\frac{}{x : [\mathsf{L}] \vdash^{0}_{\LEFTcircle} x : \mathsf{L}}\ \mathsf{ax} \qquad \frac{(\Gamma_i \vdash^{k_i}_{\LEFTcircle} t : \mathsf{L}_i)_{i \in I} \quad I \textit{ finite}}{\uplus_{i \in I} \Gamma_i \vdash^{\sum_{i \in I} k_i}_{\LEFTcircle} t : [\mathsf{L}_i]_{i \in I}}\ \mathsf{many} \qquad \frac{\Gamma, x : \mathsf{M} \vdash^{k}_{\LEFTcircle} t : \mathsf{L}}{\Gamma \vdash^{k}_{\LEFTcircle} \lambda_{\mathsf{c}} x.t : \mathsf{M} \xrightarrow{\mathsf{c}} \mathsf{L}}\ \lambda$$

$$\frac{\Gamma \vdash^{k_1}_{\LEFTcircle} t : \mathsf{M} \xrightarrow{\mathsf{c}} \mathsf{L} \quad \Delta \vdash^{k_2}_{\LEFTcircle} u : \mathsf{M}}{\Gamma \uplus \Delta \vdash^{k_1 + k_2}_{\LEFTcircle} t \cdot^{\mathsf{c}} u : \mathsf{L}}\ @_{\tau} \qquad \frac{\Gamma \vdash^{k_1}_{\LEFTcircle} t : \mathsf{M} \xrightarrow{\mathsf{c}} \mathsf{L} \quad \Delta \vdash^{k_2}_{\LEFTcircle} u : \mathsf{M}}{\Gamma \uplus \Delta \vdash^{k_1 + k_2 + 1}_{\LEFTcircle} t \cdot^{\mathsf{c}^{\perp}} u : \mathsf{L}}\ @_{\LEFTcircle}$$

As is standard, we obtain (by subject reduction/expansion) that a checkers term is typable $\Gamma \vdash^{k}_{\LEFTcircle} t : \mathsf{L}$ exactly when it is head normalizable, and the index k gives an upper bound on the number of interaction steps from t to its $\mathtt{h}_{\circ\bullet}$-nf.

We study the interpretation of terms in this colored relational model.

Definition 4. *The* colored interpretation *of a checkers term $t \in \Lambda_{\circ\bullet}$ is given by:*

$$[\![t]\!]^{\LEFTcircle} \coloneqq \{(\Gamma, \mathsf{L}, k) \mid \exists\, \pi \rhd \Gamma \vdash^{k}_{\LEFTcircle} t : \mathsf{L}\}.$$

There are several possible ways of comparing these semantic interpretations. The most natural one is set theoretical inclusion $[\![t]\!]^{\LEFTcircle} \subseteq [\![u]\!]^{\LEFTcircle}$, which leads to the checkers type preorder studied in [3] but does not capture the interaction improvement $\sqsubseteq^{\mathrm{imp}}_{\LEFTcircle}$. Instead, one has to check that t and u have the same typings, but allowing the index k' associated with u to be smaller:

$$t \leq u \text{ if } \forall \Gamma,\ \mathsf{L},\ k,\ \Gamma \vdash^{k}_{\LEFTcircle} t : \mathsf{L} \implies \exists k' \leq k,\ \Gamma \vdash^{k'}_{\LEFTcircle} u : \mathsf{L}$$

Unfortunately, this definition does not work, since $\sqsubseteq^{\mathrm{imp}}_{\LEFTcircle}$ validates η-reduction on monochromatic checkers terms (cf. Example 1), while this comparison suffers from the same issue as the inclusion. Consider the black η-expansion of x:

$$\dfrac{\dfrac{\dfrac{}{x:[\mathbf{0}\xrightarrow{\circ}\mathsf{L}]\vdash^0_{\LEFTcircle} x:\mathbf{0}\xrightarrow{\circ}\mathsf{L}}\,\mathsf{ax} \qquad \dfrac{}{\vdash^0_{\LEFTcircle} y:\mathbf{0}}\,\mathsf{many}}{x:[\mathbf{0}\xrightarrow{\circ}\mathsf{L}]\vdash^1_{\LEFTcircle} x\bullet y:\mathsf{L}}\,@_{\LEFTcircle}}{x:[\mathbf{0}\xrightarrow{\circ}\mathsf{L}]\vdash^1_{\LEFTcircle}\lambda_\bullet y.x\bullet y:\mathbf{0}\xrightarrow{\bullet}\mathsf{L}}\,\lambda$$

and note that $\lambda_\bullet y.x\bullet y \sqsubseteq^{\mathrm{imp}}_{\LEFTcircle} x$ holds, whereas $x:[\mathbf{0}\xrightarrow{\circ}\mathsf{L}]\vdash^{k'}_{\LEFTcircle} x:\mathbf{0}\xrightarrow{\bullet}\mathsf{L}$ cannot hold, because of the mismatch between the colors. Indeed, the rule ax requires that both occurrences of $\mathbf{0}\to\mathsf{L}$ share the same color.

Whitening Types. The above example suggests that, if we wish to define a preorder $\sqsubseteq^{\mathrm{pwc}}_{\LEFTcircle}$ that coincides with $\sqsubseteq^{\mathrm{imp}}_{\LEFTcircle}$ on the interpretations of black terms, we need a comparison between typings that can reduce the cost of derivations and whiten certain arrows in the types. Let us analyze the example further:

$$(x:[\mathsf{M}\xrightarrow{\mathsf{c}}\mathsf{L}],\mathsf{M}\xrightarrow{\mathsf{d}}\mathsf{L},k)\in[\![\lambda_\bullet y.x\bullet y]\!]^{\LEFTcircle}$$

iff $\mathsf{d}=\bullet$ (forced by the $\lambda_\bullet y$) and either $\mathsf{c}=\bullet$ and $k=0$, or $\mathsf{c}=\circ$ and $k=1$. A triple of this kind[2] belongs to the interpretation of x whenever $\mathsf{c}=\mathsf{d}$ and $k=0$. In particular, there is a whiter and cheaper typing $(x:[\mathsf{M}\xrightarrow{\circ}\mathsf{L}],\mathsf{M}\xrightarrow{\circ}\mathsf{L},0)\in[\![x]\!]^{\LEFTcircle}$.

This can be generalized: whenever $\Gamma\vdash^k_{\LEFTcircle} t:\mathsf{L}$ and t is a black η-expansion of u, we can derive $\Gamma'\vdash^{k'}_{\LEFTcircle} u:\mathsf{L}'$ where (Γ',L') is a whiter version of (Γ,L) and $k'\leq k$, yielding a derivation that is both whiter and cheaper. Moreover, whitening may only be applied to arrows of positive polarity, *i.e.* functions from the environment.

Definition 5 (Polarized Whitening). *Given a* polarity $\mathsf{p}\in\{+,-\}$ *we denote by* $\neg\mathsf{p}$ *the opposite polarity.*

(i) For all $k\in\mathbb{N}$, we define $\mathsf{T}'\leq^+_k\mathsf{T}$ and $\mathsf{T}'\leq^-_k\mathsf{T}$ by:

$$\dfrac{\mathsf{p}\in\{+,-\}}{\mathsf{A}\leq^{\mathsf{p}}_0\mathsf{A}}\qquad \dfrac{\mathsf{M}'\leq^-_{k_1}\mathsf{M}\quad \mathsf{L}'\leq^+_{k_2}\mathsf{L}}{\mathsf{M}'\xrightarrow{\circ}\mathsf{L}'\leq^+_{k_1+k_2+1}\mathsf{M}\xrightarrow{\bullet}\mathsf{L}}\qquad \dfrac{\mathsf{M}'\leq^{\neg\mathsf{p}}_{k_1}\mathsf{M}\quad \mathsf{L}'\leq^{\mathsf{p}}_{k_2}\mathsf{L}}{\mathsf{M}'\xrightarrow{\mathsf{c}}\mathsf{L}'\leq^{\mathsf{p}}_{k_1+k_2}\mathsf{M}\xrightarrow{\mathsf{c}}\mathsf{L}}$$

$$\dfrac{\mathsf{L}'_1\leq^{\mathsf{p}}_{k_1}\mathsf{L}_1\quad\cdots\quad\mathsf{L}'_n\leq^{\mathsf{p}}_{k_n}\mathsf{L}_n}{[\mathsf{L}'_1,\ldots,\mathsf{L}'_n]\leq^{\mathsf{p}}_{k_1+\cdots+k_n}[\mathsf{L}_1,\ldots,\mathsf{L}_n]}$$

(ii) The relations above extend to environments Γ and pairs $\langle\Gamma,\mathsf{L}\rangle$ as follows:

$$\dfrac{}{\emptyset\leq^{\mathsf{p}}_0\emptyset}\qquad \dfrac{\Gamma'\leq^{\mathsf{p}}_{k_1}\Gamma\quad \mathsf{M}'\leq^{\mathsf{p}}_{k_2}\mathsf{M}}{\Gamma',x:\mathsf{M}'\leq^{\mathsf{p}}_{k_1+k_2}\Gamma,x:\mathsf{M}}\qquad \dfrac{\Gamma'\leq^{\neg\mathsf{p}}_{k_1}\Gamma\quad \mathsf{L}'\leq^{\mathsf{p}}_{k_2}\mathsf{L}}{\langle\Gamma',\mathsf{L}'\rangle\leq^{\mathsf{p}}_{k_1+k_2}\langle\Gamma,\mathsf{L}\rangle}$$

Intuitively, $\mathsf{T}'\leq^{\mathsf{p}}_k\mathsf{T}$ holds if T' is obtained from T by whitening k arrows occurring with polarity p. In particular, the underlying uncolored type must be the same.

Everything is now in place to introduce the preorder $\sqsubseteq^{\mathrm{pwc}}_{\LEFTcircle}$ on checkers terms.

Definition 6 (Polarized Whiter-Cheaper Improvement). *For all checkers terms t, u we define $t\sqsubseteq^{\mathrm{pwc}}_{\LEFTcircle}u$ if and only if*

$$\forall(\Gamma,\mathsf{L},k)\in[\![t]\!]^{\LEFTcircle},\ \exists(\Gamma',\mathsf{L}',k')\in[\![u]\!]^{\LEFTcircle},\exists d,\ \langle\Gamma',\mathsf{L}'\rangle\leq^+_d\langle\Gamma,\mathsf{L}\rangle \text{ such that } k\geq k'+d$$

[2] Note that $(x:[\mathsf{A}],\mathsf{A},0)\in[\![x]\!]^{\LEFTcircle}-[\![\lambda_\bullet y.x\bullet y]\!]^{\LEFTcircle}$. This is expected as $x\not\sqsubseteq^{\mathrm{imp}}_{\LEFTcircle}\lambda_\bullet y.x\bullet y$.

References

1. Accattoli, B., Dal Lago, U., Vanoni, G.: Multi types and reasonable space. Proc. ACM Program. Lang. **6**(ICFP), 799–825 (2022). https://doi.org/10.1145/3547650
2. Accattoli, B., Graham-Lengrand, S., Kesner, D.: Tight typings and split bounds, fully developed. J. Funct. Program. **30**, e14 (2020). https://doi.org/10.1017/S095679682000012X
3. Accattoli, B., Lancelot, A., Manzonetto, G., Vanoni, G.: Interaction equivalence. Proc. ACM Program. Lang. **9**(POPL) (Jan 2025). https://doi.org/10.1145/3704891, https://doi.org/10.1145/3704891
4. Biernacki, D., Lenglet, S., Polesiuk, P.: Proving soundness of extensional normal-form bisimilarities. Electronic Notes in Theoretical Computer Science **336**, 41–56 (2018). https://doi.org/https://doi.org/10.1016/j.entcs.2018.03.015, https://www.sciencedirect.com/science/article/pii/S1571066118300185, the Thirty-third Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIII)
5. Breuvart, F., Manzonetto, G., Ruoppolo, D.: Relational graph models at work. Log. Methods Comput. Sci. **14**(3) (2018). https://doi.org/10.23638/LMCS-14(3:2)2018, https://doi.org/10.23638/LMCS-14(3:2)2018
6. Bucciarelli, A., Ehrhard, T., Manzonetto, G.: Not enough points is enough. In: Duparc, J., Henzinger, T.A. (eds.) Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4646, pp. 298–312. Springer (2007). https://doi.org/10.1007/978-3-540-74915-8_24, https://doi.org/10.1007/978-3-540-74915-8_24
7. Bucciarelli, A., Kesner, D., Ventura, D.: Non-idempotent intersection types for the lambda-calculus. Log. J. IGPL **25**(4), 431–464 (2017). https://doi.org/10.1093/JIGPAL/JZX018
8. de Carvalho, D.: Sémantiques de la logique linéaire et temps de calcul. Thèse de doctorat, Université Aix-Marseille II (2007)
9. de Carvalho, D.: Execution time of λ-terms via denotational semantics and intersection types. Math. Struct. Comput. Sci. **28**(7), 1169–1203 (2018). https://doi.org/10.1017/S0960129516000396
10. Dal Lago, U., Gavazzo, F.: Effectful Normal Form Bisimulation. In: ESOP 2019 - European Symposium on Programming. Prague, Czech Republic (Apr 2019), https://hal.inria.fr/hal-02386004
11. Hyland, M., Nagayama, M., Power, J., Rosolini, G.: A category theoretic formulation for engeler-style models of the untyped lambda. In: Seda, A.K., Hurley, T., Schellekens, M.P., an Airchinnigh, M.M., Strong, G. (eds.) Proceedings of the Third Irish Conference on the Mathematical Foundations of Computer Science and Information Technology, MFCSIT 2004, Dublin, Ireland, July 22-23, 2004. Electronic Notes in Theoretical Computer Science, vol. 161, pp. 43–57. Elsevier (2004). https://doi.org/10.1016/J.ENTCS.2006.04.024, https://doi.org/10.1016/j.entcs.2006.04.024
12. Ker, A.D., Nickau, H., Ong, C.L.: Innocent game models of untyped lambda-calculus. Theor. Comput. Sci. **272**(1-2), 247–292 (2002). https://doi.org/10.1016/S0304-3975(00)00353-4, https://doi.org/10.1016/S0304-3975(00)00353-4
13. Ker, A.D., Nickau, H., Ong, C.L.: Adapting innocent game models for the böhm tree λ-theory. Theor. Comput. Sci. **308**(1-3), 333–366 (2003). https://doi.org/10.1016/S0304-3975(02)00849-6

14. Koutavas, V., Lin, Y.Y., Tzevelekos, N.: Fully abstract normal form bisimulation for call-by-value pcf. In: 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). pp. 1–13 (2023). https://doi.org/10.1109/LICS56636.2023.10175778
15. Lancelot, A., Manzonetto, G., McCusker, G., Vanoni, G.: Interaction improvement. In: Bertrand, N., Milius, S. (eds.) Foundations of Software Science and Computation Structures. pp. 372–395. Springer Nature Switzerland, Cham (2026)
16. Levy, P.B., Staton, S.: Transition systems over games. In: Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). CSL-LICS '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2603088.2603150, https://doi.org/10.1145/2603088.2603150
17. Paolini, L., Piccolo, M., Ronchi Della Rocca, S.: Essential and relational models. Math. Struct. Comput. Sci. **27**(5), 626–650 (2017). https://doi.org/10.1017/S0960129515000316, https://doi.org/10.1017/S0960129515000316
18. Sands, D.: Proving the correctness of recursion-based automatic program transformations. Theor. Comput. Sci. **167**(1&2), 193–233 (1996). https://doi.org/10.1016/0304-3975(96)00074-6
19. Sands, D.: Total correctness by local improvement in the transformation of functional programs. ACM Trans. Program. Lang. Syst. **18**(2), 175–234 (mar 1996). https://doi.org/10.1145/227699.227716
20. Sands, D.: Improvement theory and its applications, p. 275–306. Cambridge University Press, USA (1999)
21. Wadsworth, C.P.: The Relation Between Computational and Denotational Properties for Scott's $\mathcal{D}_\infty$-Models of the Lambda-Calculus. SIAM J. Comput. **5**(3), 488–521 (1976). https://doi.org/10.1137/0205036