Theories of Names by A. Pitts and M. Gabbay and Semantics of References in Programming Languages

Andrei V. Klimov

Keldysh Institute of Applied Mathematics of Russian Academy of Sciences

Moscow, Russia

A selfish purpose of this talk:

I am looking for experts in Category Theory
who are interested in its applications
to semantics of programming languages

Main message of this talk

1. **We have: Semantics of functional languages** extended by the **concept of a name or reference**
   - Theories by **Andrew M. Pitts** and **Murdoch J. Gabbay**

2. **Cornerstone: The concept of contextual equivalence, Leibniz equivalence**
   - "Equivalent are those entities that we cannot distinguish“

3. **Category Theory** is the most promising one among the theories by A. Pitts and M. Gabbay
   - Computer scientist need to master it

4. **Wanted: A theory of object-oriented languages**
   - with adequate semantics of references and object

5. **Purpose**: To move **from functional to object-oriented programming**, **preserving the beauty** of the theory of functional languages:
   - it is possible to go quite far
   - **removing the main restriction** on functional languages: **trees → graphs**
   - many valuable **properties are preserved**, including the **determinacy of parallel computations**

Purely functional language *vs*. references as data

- **Purely functional languages**
  - **Data** = first-order values (with equality) + closures (without equality)
  - **Program** = a set of function definitions; no primary operations with side effects; for example:
    - *factorial*$(n) =$ ***if*** $n = 0$ ***then*** 1 ***else*** $n * $ *factorial*$(n-1)$
  - **Semantics** is defined **without the order of computation** (denotational, based on set theory)
    - $h(f(x), g(y))$ – calls to functions f, g, h can be evalulated in any order
  - Hence, programs are trivially **parallelized**
  - and have **referential transparency:**:
    - For all functions in the language that return a value with equality, the following property holds

$$f(x) = f(x)$$

- **Differences between functional and object-oriented languages**
  - The first thing that comes to mind:
    - **no mutable objects**; no changing states; no side effects
  - But even if we give up mutable objects, we still won't get a functional language
  - **Reference generator** ***new*** = generator of new objects without an object with a state
    - has **no referential transparency**

$$\textit{\textbf{new}} \neq \textit{\textbf{new}}$$

  - Hence, the **classical detonation semantics** based on set theory does **not work**
    - **we need foundations of mathematics (logics) without referential transparency**

Purely functional language *with* references as data

- Consider a **purely functional language** extended by
  - **references** in the data domain
  - **reference generator *new*** among the primary operations
- **Does it have any good property to build a good theory?**
- **Leibniz equivalence, contextual equivalence, observational equivalence**

$$x \approx y \coloneqq (\forall C : D \to Bool)\ C(x) = C(y)$$

- **Statement**
  - In a functional **language** with the ***new*** reference generator,

    the results of **evaluation of copies of a term are contextually equivalent**

$$f(x) \approx f(x)$$

  - Moreover, the values of functions from contextually **equivalent arguments** are equivalent :

$$x \approx y \Rightarrow f(x) \approx f(y)$$

- Such **modified referential transparency** with contextual equivalence instead of the usual equality
  - **is the basis for building good mathematical theories**

Nominal set Theory by Andrew Pitts and Mordach Gabbay

- **2019 ACM Alonzo Church Award for Outstanding Contributions to Logic and Computation** is given jointly to **Murdoch J. Gabbay** and **Andrew M. Pitts** for **their ground-breaking work introducing the theory of nominal representations** http://siglog.org/winners-of-the-2019-alonzo-church-award

- Andrew Pitts and his disciples conducted research for almost thirty years. Selected papers:

1993 Andrew M. Pitts, Ian D.B. Stark. **Observable properties of higher order functions that dynamically create local names, or: What's new?** *Lecture Notes in Computer Science*, vol 711. DOI: 10.1007/3-540-57182-5_8

2013 Andrew M. Pitts. **Nominal Sets: Names and Symmetry in Computer Science**. *Cambridge Tracts in Theor. Comp. Sci.*, vol. 57. DOI: 10.1017/CBO9781139084673

2016 Andrew M. Pitts. **Nominal techniques**. *ACM SIGLOG News* 3, 1, 57–72. DOI: 10.1145/2893582.2893594

- **Theories of functional languages with a reference (name) generator are constructed based on**
  - Set Theory (Fraenkel-Mostowski permutation model, 1922-1938)
  - Category Theory (Schanuel topos) 🡸 looks most promising
  - operational approach

- **We have not yet found** works on the theory and **complete semantics of object-oriented languages**
  - with **formal semantics of references** along with parallel computation

Our publications on deterministic parallel programming and connections with the theories by A. Pitts and M. Gabbay

On deterministic parallel programming

- А.И. Адамович, Анд.В. Климов. **Как создавать параллельные программы, детерминированные по построению? Постановка проблемы и обзор работ** // *Программные системы: теория и приложения*. 2017. Т. 8. № 4 (35). С. 221–244. doi:10.25209/2079-3316-2017-8-4-221-244
- А.И. Адамович, Анд.В. Климов. **Подход к построению системы детерминированного параллельного программирования на основе монотонных объектов** // *Вестник СибГУТИ*. 2019. № 3. С. 14–26. URL: http://vestnik.sibsutis.ru/uploads/1570089084_1278.pdf
- A.I. Adamovich, And.V. KLimov. **Building Cyclic Data in a Functional-Like Language Extended with Monotonic Objects** // *X Workshop PSSV: Program Semantics, Specification and Verification: Theory and Applications : Abstracts*. Novosibirsk : A.P. Ershov IIS, 2019. P. 11–19. URL: https://persons.iis.nsk.su/files/persons/pages/tezisy_seminara_pssv.pdf.
- А.И. Адамович, Анд.В. Климов. **О детерминированной параллельной реализации метода ветвей и границ на монотонных объектах** // *Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции*. М. : ИПМ им. М.В. Келдыша, 2019. С. 3–18. — doi:10.20948/abrau-2019-61

On A. PItts theories

- А.И. Адамович, Анд.В. Климов. **О теориях имен и ссылок в формальных языках и последствиях для функционального и объектно-ориентированного программирования** // *Научный сервис в сети Интернет: труды XXIII Всероссийской научной конференции* (20-23 сентября 2021 г.). — М. : ИПМ им. М.В.Келдыша, 2021. С. 3-21. — doi:10.20948/abrau-2021-30.
- And.V. Klimov. **On Semantics of Names in Formulas and References in Object-Oriented Languages** // *Computer Algebra: 4th International Conference Materials. Moscow, June 28–29, 2021. —* Moscow : MAKS Press, 2021. P. 73–76. — URL: http://www.ccas.ru/ca/_media/ca-2021.pdf.

Conclusion 2: State of art

- Many ideas of the theory of computation and programming languages are based on the concept of
  - **contextual equivalence (observational, Leibniz equivalence)**

  which should be used **instead of the equality of values**
- **Andrew Pitts' group** developed theories of functional languages extended by notions of:
  - **references** (names) in the data domain
  - **reference generation *new*** among primary operation

  having **referential transparency with contextual equivalence** instead of the equality
- **They have developed** and implemented
  - **programming language FreshML** with these properties
  - **package Nominal for proof assistant Coq** for proving properties of programs in FreshML
- **We are not aware of applications of this theory to object-oriented languages**
  - **Open problem**: Building a theory of object-oriented languages with complete semantics of references and objects, , including parallel and concurrent computation
  - Such a theory is needed for the verification of parallel object-oriented programs
- **Our contribution: It is possible to impose restrictions on operations on objects** in an object-oriented language so that **referential transparency with contextual equivalence** holds, **deterministic parallelism** and other good properties are preserved
  - we are working on such a system of deterministic parallel programming
    (together with **Alexei Adamovich**, Ailamazyan Program Systems Institute of RAS)

Conclusion 2: Questions

- **Is Category Theory taught in some university in Russian?**
  - It must be taught for computer scientists (semantics of languages, type theory, verification)
  - CT concepts and statements look simple, but habits should be developed from youth
- **Are there in Russia experts in Category Theory working on / interested in semantics of lang's?**
  - Applications of CT to type theory are well-known from 1990s
  - –"– to semantics (functional, denotational, operational) are not so wide-spread (Am I right?)
- **Am I right by saying that there is no complete semantics of object-oriented languages:**
  - especially pointing to formal semantics of references?
  - and considering parallelism?
- **Are these topics interesting to the community of this seminar?**
  - semantics of programming languages
  - problems of using traditional methods of semantics (like set-theoretic denotational one)
  - alternative approaches to semantics
- **Position: Without precisely formally defining a semantics of a languages we can't speak about:**
  - program verification and correctness of various analyses
  - equivalence program transformations and metacomputation in general