様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　５ 年　 ６ 月 １３ 日現在

機関番号：１１３０１
研究種目：基盤研究(C)（一般）
研究期間：2017～2022
課題番号：１７Ｋ００９１

研究課題名（和文）Tagless-final DSL embedding: how to keep extending the language and be sure it will still work

研究課題名（英文）Tagless-final DSL embedding: how to keep extending the language and be sure it will still work

研究代表者
Ｋｉｓｅｌｙｏｖ　Ｏｌｅｇ（KISELYOV, Oleg）

東北大学・情報科学研究科・助教

研究者番号：５０７５４６０２

交付決定額（研究期間全体）：（直接経費）　　3,500,000円

研究成果の概要（和文）：tagless-finalとは、内部DSL(埋め込みドメーイン領域特化言語)の埋め込み法であり、効率が高い、拡張の容易で、型安全性を持つDSL実装へと繋っている．本研究には、筆者の先行研究に提案されたtagless-final法がより一層発展され、データベースクエリと高性能ストリーム処理から自然言語の文法と意味論解析まで幅広い範囲で適用できた．複数企業でも利用されてきた．

研究成果の学術的意義や社会的意義
Tagless-final was used in the conduct of Personalized Genomic Cancer Vaccine clinical trial NCT02721043, Mount Sinai - Hammer Lab, 2016-2019. Tagless-final has been used in several companies and large-scale research projects, and mentioned in job ads. I had no involvement with these applications.

研究成果の概要（英文）：Tagless-final approach is a method for embedding domain-specific languages (DSLs) in a programming language. It addresses the principal challenge: implementing embedded DSLs that are (1) efficient; (2) easily extensible for new domain knowledge; (3) statically assured to "do no wrong" no matter how they are modified. Proposed in the author's prior research, the present research has developed the approach to a practical extent: a wide variety of DSLs, from database query and highest-performance stream processing to natural language grammar and semantics. The results are taught at summer schools, presented at scientific meetings, published in journal articles and a monograph. The approach underlies the new undergraduate class `Compilers' that I have developed and taught. The tagless-final approach has been seemingly successfully used in other academic and commercial projects, and explained in several books and taught in many tutorials, by people with no affiliation with me.

研究分野：プログラミング言語

キーワード：DSL　Domain-specific language　tagless-final　denotational semantics　algebra

１．研究開始当初の背景

Domain-specific languages (DSLs) help us deal with the increasingly more complex and pervasive software systems by giving notation, analysis, verification, optimization and tooling specialized to an application domain. Examples include: scientific computing, e.g., Firedrake for solving partial differential equations using finite element method (used by Rolls-Royce for the simulation of turbomachinery components of jet engines); Halide to generate sharpening, blurring etc. filters for Adobe photoshop and smartphone camera software; the highly-configurable operating system MirageOS. DSLs are pervasive in the financial industry.

This research concerns a large, practically-significant class of DSLs: agile, easily extensible and embedded in a host language. An example is DSLs for representing (database) relations in a programming language. The domain, of relational algebra, is fixed. Yet we frequently need to extend the language, for a different data store (in-memory data, SNS feed, etc.) or to let the users tell some features of data or query that may help improve the performance. Expressing domain-specific knowledge is a driving force for extensibility.

The agility – adding features and adjusting the implementation – brings in the danger of inadvertently breaking something in the process. It is often mentioned that many bugs, in prose or programs, occur during editing. A common way to assure a degree of correctness and guard against silly mistakes is types. Static typing is especially important for DSLs used by non-professional programmers. Since agile DSLs are meant to be frequently extended and adjusted, we want to statically (before running) assure that their implementation “does no wrong”: it is consistent with DSL types at all times, no matter how we modify it. Since many DSL adjustments take the form of program transformations, we want them to preserve DSL types (and hence be typed themselves). Easily and assuredly implementing static analyses (of which typing is one) was stated as the main challenge at the Dagstuhl seminar on DSL.

To fulfill the desiderata, we have proposed a so-called tagless-final approach for embedding of typed DSLs in a typed programming language. Since only well-typed DSL terms are representable, the well-typedness of DSL expressions is assured by construction. The approach hence represents not only the syntax and the denotational semantics of a DSL but also its type system, making it easy to ascertain the soundness of the latter. The approach is extensible: a
DSL expression, once written, can be interpreted in a variety of ways (to evaluate, to pretty-print, to transform and compile), with more interpreters can be added at any time. Not only the set of interpreters is extensible; the DSL itself is extensible, too, with new language forms, constants and operations. As we enrich the syntax of the DSL, we still reuse the old interpreters as they are. We have used the approach to implement several extensible DSLs in the domain of language-integrated query, probabilistic programming, delimited continuations, hardware description languages, generation of specialized numerical kernels and natural language semantics. For some implementations, performance mattered. A DSL expression had to be analyzed and converted to the form which can be efficiently evaluated. These optimization passes are, by construction, scope-safe and type-preserving.

２．研究の目的

We solve a principal challenge of domain-specific languages (DSL): implementing embedded DSLs that are (1) efficient; (2) easily extensible for new domain knowledge by domain experts; (3) statically assured to "do no wrong" no matter how they are modified. All three requirements are satisfied simultaneously. In prior research we have developed a tagless-final DSL embedding that had shown great promise to meet the challenge, on a few simple examples.

Yet the generality of the approach is not yet clear. We have not yet tried complex, global optimizations. We have not yet tried embedding languages with complex type systems, such as those in natural language semantics (type-logical grammars) or program verification. Some even doubt if the typed final approach is capable of doing these tasks.

This research is to develop the potential, extending the tagless-final approach to address the open challenges:

- performing global transformations used in optimizing compilers, but in a type- and scope- safe way;
- embedding DSLs with complex, resource-sensitive (substructural) type systems.

３．研究の方法

To answer the challenges, we chose to tackle three practically useful extensible embedded DSL: for generating highest-performance numeric kernels, probabilistic programming for graphical models, and for prototyping (substructural) type calculi. The DSLs are deliberately chosen from very distinct domains.

- Embedded linear algebra DSL for specifying compositions of BLAS Level 3 operations in a Matlab-like way and generating the corresponding highest-performing numerical kernels. At first we target conventional superscalar processors; in further research, we add more backends (GPGPU, supercomputers, FPGA). The language may be considered a clone of Spiral's SPL, but with types, embedded rather than standalone, and extensible. The implementation certainly needs complex global optimizations.

- Probabilistic programming for graphical models

- Embedded DSL for specifying typed calculi, used in prototyping of functional languages and their type systems and in theoretical linguistics. The language is similar to the Logical Framework (LF) but again, embedded rather than standalone. The language should be capable of expressing Type Logical Grammars and resource-sensitive calculi (such as the full linear logic and its variations).

Not only these three DSLs serve as proof of concept of the proposed framework, they should be useful on their own (and their further development is subject of further research, of the author and the collaborators). Since these DSLs are embedded, they require less maintenance and hopefully enjoy longer life than the standalone DSLs (no longer maintained) by which they are inspired.

４．研究成果

All planned DSLs have been implemented, with the results presented at scientific meetings and published in proceedings and journals. The DSLs are used in education: in the course on compilers that I have been developing at Tohoku University; in the course on probabilistic
programming (2016 Formosan Summer School on Logic, Language, and Computation, Taipei, Taiwan); at the summer school on meta-programming (Dagstuhl, 2019).

In addition, while working on the problem of performing global optimizations on DSLs embedded in the tagless-final style, I have developed the approach of using normalization-by-evaluation – rather than term re-writing – to transform a program to the normal form. The approach has proved successful. First, I was able to reproduce, in a much simpler way, the previous work in optimizing language-integrated database queries and compiling them to SQL. Mainly, the approach was readily extended to queries that involved ordering and subranging operations. For the first time we have proposed the compositional semantics of these query operations. The developed language-integrated query system is the first that supports ordering and subranging in a portable and consistent way. The results have been published.

The tagless-final approach and the normalization-by-evaluation has been the foundation for Strymonas - the highest performance single-thread stream processing library - whose version 2.0 was released in 2022. We have demonstrated that using the same DSL we can generate OCaml, Scala and even C code.

I have also worked on the reasoning about tagless-final DSLs. I have investigated abstract interpretation, and denotational semantics of DSL programs with computational effects such as mutation and non-determinism, whose results are published in two journal papers (EPTCS 285, 2018) and (EPTCS 294, 2019).

I have used the tagless-final approach in writing code to check and assist in theoretical work in lambda-calculus, whose results are published as a journal paper (J. Functional Programming, v30, 2020, e7). The tagless-final approach was instrumental for prototyping and constructive proofs of correctness in solving a long-standing problem in heterogeneous metaprogramming: general offshoring with mutable variables.

I, together with a master student, accomplished the second main project goal: using tagless-final to embed Logical-framework-like DSL for specifying typed calculi, for prototyping of functional languages and their type systems and, in particular, natural language semantic theories. We have implemented a DSL for describing derivations in AB grammars, Lambek grammars, and Kubota and Levin's Hybrid Type-Logical grammars. Not only do we mechanically check that derivations are well-formed, but we also format them as LaTeX figures according to the common theoretical-linguistic conventions.

The above research has lead to an unexpected result in the field of Formal Grammar: an algebraic presentation of Lambek Grammars and the demonstration that under a natural restriction (satisfied in natural languages), Lambek grammars are strongly equivalent to context-free grammars, strengthening the existing solutions to this long-standing problem.

I have applied the tagless-final DSL approach to the completely different domain: transforming terms representing natural language sentences into logical formulas. This work is a part of a large project on natural language comprehension. The results have been published in a series of papers in LNCS.

I have published the monograph:

Reconciling Abstraction with High Performance: A MetaOCaml approach.
nowpublishers, Foundations and Trends® in Programming Languages Series ISBN: 978-1-68083-436-9
1-112 pp., 2018.
https://www.amazon.co.jp/Reconciling-Abstraction-High-Performance-Foundations/dp/1680834363

which presents tagless-final, in combination with staging, as the main method for optimizing and compiling DSLs.

As part of disseminating research results and making them more accessible and applicable, I have delivered a three-lecture course at the International Summer School on Metaprogramming (Dagstuhl, Germany, 2019), on the tagless-final approach and its algebraic semantics.
https://www.cl.cam.ac.uk/events/metaprog2019/index.html

Tagless-final approach has been used, seemingly successfully, by other people, with whom I have no affiliation or even contact:

Tagless-final approach was used in a conduct of a cancer treatment trial

Personalized Genomic Vaccine clinical trial NCT02721043,
Mount Sinai - Hammer Lab, 2016-1019. See
Sebastien Mondet: Bioinformatics, The Typed Tagless Final Way
https://icfp17.sigplan.org/event/ocaml-2017-papers-bioinformatics-the-typed-tagless-final-way

Tagless-final is used in Carnegie Mellon University's Binary Analysis Platform.
https://discuss.ocaml.org/t/ann-bap-2-0-release/4719
https://icfp21.sigplan.org/details/ocaml-2021-papers/10/Binary-Analysis-Framework-BAP-Using-Universal-Algebra-and-Tagless-Final-Style-for-

Tagless-final is expounded in textbooks:

Mastering Functional Programming
by Anatolii Kmetiuk. Packt Publishing, 2018
http://www.foxebook.net/mastering-functional-programming

Functional Programming for Mortals
Sam Halliday, LeanPub, 2020
https://leanpub.com/fpmortals/read

and many articles and lectures, for details, see
https://github.com/atapin/awesome-tagless-final
https://awesomeopensource.com/projects/tagless-final

５．主な発表論文等

〔雑誌論文〕　計18件（うち査読付論文　17件／うち国際共著　17件／うちオープンアクセス　4件）

１．著者名 Kiselyov Oleg	４．巻 12758
２．論文標題 Polynomial Event Semantics: Negation	５．発行年 2021年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 82～95
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-79942-7_6	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Kiselyov Oleg	４．巻 13215
２．論文標題 Generating C	５．発行年 2022年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 75～93
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-99461-7_5	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 KISELYOV OLEG、MU SHIN-CHENG、SABRY AMR	４．巻 31
２．論文標題 Not by equations alone: Reasoning with extensible effects	５．発行年 2021年
３．雑誌名 Journal of Functional Programming	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） 10.1017/S0956796820000271	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Kiselyov Oleg、Imai Keigo	４．巻 12073
２．論文標題 Session Types Without Sophistry	５．発行年 2020年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 66～87
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-59025-3_5	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg、Hoshino Yuya	4．巻 12331
2．論文標題 Lambek Grammars as Second-Order Abstract Categorial Grammars	5．発行年 2020年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 231～243
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-58790-1_15	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 小林 友明, Oleg Kiselyov	4．巻 -
2．論文標題 より清浄なStream Fusion	5．発行年 2020年
3．雑誌名 日本ソフトウェア科学会第37回大会, 2020	6．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

1．著者名 Kiselyov Oleg	4．巻 294
2．論文標題 Effects Without Monads: Non-determinism ? Back to the Meta Language	5．発行年 2019年
3．雑誌名 Electronic Proceedings in Theoretical Computer Science	6．最初と最後の頁 15～40
掲載論文のDOI（デジタルオブジェクト識別子） 10.4204/EPTCS.294.2	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

1．著者名 Kiselyov Oleg	4．巻 11717
2．論文標題 Polynomial Event Semantics	5．発行年 2019年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 313～324
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-31605-1_23	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 KISELYOV OLEG	4．巻 30
2．論文標題 Many more predecessors: A representation workout	5．発行年 2020年
3．雑誌名 Journal of Functional Programming	6．最初と最後の頁 e7
掲載論文のDOI（デジタルオブジェクト識別子） 10.1017/S095679682000009X	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg、Hoshino Yuya	4．巻 12331
2．論文標題 Lambek Grammars as Second-Order Abstract Categorial Grammars	5．発行年 2020年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 231～243
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-58790-1_15	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg、Sivaramakrishnan KC	4．巻 285
2．論文標題 Eff Directly in OCaml	5．発行年 2018年
3．雑誌名 Electronic Proceedings in Theoretical Computer Science	6．最初と最後の頁 23～58
掲載論文のDOI（デジタルオブジェクト識別子） 10.4204/EPTCS.285.2	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

1．著者名 Kiselyov Oleg	4．巻 10818
2．論文標題 $$\lambda $$λ to SKI, Semantically	5．発行年 2018年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 33～50
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-90686-7_3	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg	4．巻 10838
2．論文標題 Transformational Semantics on a Tree Bank	5．発行年 2018年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 241～252
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-93794-6_17	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg、Katsushima Tatsuya	4．巻 10695
2．論文標題 Sound and Efficient Language-Integrated Query	5．発行年 2017年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 364～383
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-71237-6_18	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kiselyov Oleg	4．巻 10838
2．論文標題 Transformational Semantics on a Tree Bank	5．発行年 2018年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 241～252
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-93794-6_17	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Oleg Kiselyov	4．巻 na
2．論文標題 Higher-order Programming is an Effect	5．発行年 2017年
3．雑誌名 Informal Proc. HOPE 2017 at ICFP 2017. Oxford, UK, Sep 3, 2017	6．最初と最後の頁 online
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

1．著者名 Oleg Kiselyov	4．巻 2211.04107
2．論文標題 Do Mutable Variables Have Reference Types?	5．発行年 2022年
3．雑誌名 ACM SIGPLAN ML Family Workshop 2022/ArXiv	6．最初と最後の頁 online
掲載論文のDOI（デジタルオブジェクト識別子） 10.48550/arXiv.2211.04107	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

1．著者名 Oleg Kiselyov and Haruki Watanabe	4．巻 in print
2．論文標題 Events and Relative Clauses	5．発行年 2023年
3．雑誌名 Lecture Notes in Computer Science	6．最初と最後の頁 in print
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕　計20件（うち招待講演　3件／うち国際学会　15件）

1．発表者名 Oleg Kiselyov
2．発表標題 QNP Textual Entailment with Polynomial Event Semantics
3．学会等名 LENLS18 (Logic and Engineering of Natural Language Semantics 18)（国際学会）
4．発表年 2021年

1．発表者名 庄司諭, Oleg Kiselyov
2．発表標題 配列言語の長所を分かりやすく味わう: OCaml上の埋め込み配列言語（ポスター・デモ）
3．学会等名 第23回プログラミングおよびプログラミング言語ワークショップ（PPL 2021）
4．発表年 2021年

１．発表者名
Oleg Kiselyov

２．発表標題
Polynomial Event Semantics: Negation

３．学会等名
Logic and Engineering of Natural Language Semantics 17 (LENLS17)（国際学会）

４．発表年
2020年

１．発表者名
小林 友明

２．発表標題
より清浄なStream Fusion

３．学会等名
日本ソフトウェア科学会第37回大会

４．発表年
2020年

１．発表者名
Oleg Kiselyov

２．発表標題
Lambek Grammars as Second-order Abstract Categorial Grammars

３．学会等名
Logic and Engineering of Natural Language Semantics, LENLS16（国際学会）

４．発表年
2019年

１．発表者名
Oleg Kiselyov

２．発表標題
From the tagless-final cookbook: simple hardware description language and optimization-by-evaluation

３．学会等名
Second International Summer School on Metaprogramming, Schloss Dagstuhl（招待講演）（国際学会）

４．発表年
2019年

１．発表者名
Oleg Kiselyov

２．発表標題
lambda to SKI, Semantically. Declarative Pearl

３．学会等名
Fourteenth International Symposium on Functional and Logic Programming（国際学会）

４．発表年
2018年

１．発表者名
Oleg Kiselyov

２．発表標題
Lambda to SKI

３．学会等名
IFIP Working Group 2.1, 77th meeting (Brandenburg, Germany)（国際学会）

４．発表年
2018年

１．発表者名
Oleg Kiselyov

２．発表標題
Polynomial event Semantics: Non-Montagovian proper treatment of quantifiers.

３．学会等名
Logic and Engineering of Natural Language Semantics, LENLS15（国際学会）

４．発表年
2018年

１．発表者名
Oleg Kiselyov

２．発表標題
tagless-final・extensible-effectsから機械自然言語理解まで [C3（ポスター・デモ）]

３．学会等名
第20回プログラミングおよびプログラミング言語ワークショップ PPL 2018

４．発表年
2018年

１．発表者名
Oleg Kiselyov

２．発表標題
Systematic Generation of Optimal Code

３．学会等名
Tutorial at IFL 2017（招待講演）（国際学会）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Sound and Efficient Language-Integrated Query: Maintaining the ORDER

３．学会等名
EPFL, IC Colloquium (EPFL, Lausanne, Switzerland)（招待講演）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Higher-order Programming is an Effect

３．学会等名
HOPE 2017 at ICFP 2017（国際学会）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Effects without monads: non-determinism

３．学会等名
ML Family workshop at ICFP 2017（国際学会）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Sound and Efficient Language-Integrated Query -- Maintaining the ORDER

３．学会等名
APLAS 2017（国際学会）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Transformational Semantics (TS) on a Tree Bank

３．学会等名
LENLS 14（国際学会）

４．発表年
2017年

１．発表者名
Oleg Kiselyov

２．発表標題
Do Mutable Variables Have Reference Types?

３．学会等名
ACM SIGPLAN ML 2022 Workshop（国際学会）

４．発表年
2022年

１．発表者名
Oleg Kiselyov

２．発表標題
Generating C

３．学会等名
FLOPS 2022（国際学会）

４．発表年
2022年

１．発表者名
Oleg Kiselyov

２．発表標題
Events and Relative Clauses

３．学会等名
LENLS (Logic and Engineering of Natural Language Semantics 19)（国際学会）

４．発表年
2022年

１．発表者名
Oleg Kiselyov

２．発表標題
Lambek Grammars and a New Look to Context-Free Grammars (half-tutorial)

３．学会等名
AiDL 2022 Workshop

４．発表年
2022年

〔図書〕　計1件

１．著者名	４．発行年
Oleg Kiselyov	2018年
２．出版社	５．総ページ数
Now Publishers	112
３．書名	
Reconciling Abstraction with High Performance: A MetaOCaml approach	

〔産業財産権〕

〔その他〕

Lambek Grammars as an embedded DSL
http://okmij.org/ftp/gengo/LG-algebra.html
Higher-kinded bounded polymorphism without ...
http://okmij.org/ftp/ML/higher-kind-poly.html
Better than shell pipes
http://okmij.org/ftp/ML/myawk/index.html
Elementary Tutorial on Normalization-by-Evaluation
http://okmij.org/ftp/tagless-final/NBE.html
Embedding of lambda calculus with De Bruijn Levels
http://okmij.org/ftp/tagless-final/cookbook.html#dblevels
Lambda calc embedded in OCaml and normalization
http://okmij.org/ftp/Computation/lambda-calc.html#lambda-normalizer-ocaml
Executable direct denot sem of ... delim. control
http://okmij.org/ftp/continuations/implementations.html#denot
Algebras
http://okmij.org/ftp/tagless-final/Algebra.html
Evaluators, Normalizers, Reducers
http://okmij.org/ftp/tagless-final/semantics.html
Tagless-final operational semantics
http://okmij.org/ftp/tagless-final/cookbook.html#reducer
Non-determinism: a sublanguage rather than a monad
http://okmij.org/ftp/tagless-final/nondet-effect.html
Typed final (tagless final) style

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関