

Istoricul domeniului interpretării monadice, așa cum se poate reconstitui de pe Internet, dintr-o serie de surse liber accesibile (selectiv).

1988 .: Eugenio Moggi își susține teza sa de doctorat “The Partial Lambda Calculus” la Univ. Edinburgh. El se va ocupa ulterior, printre altele, de modelarea programării imperative prin concepte ale limbajelor funcționale și va introduce monadele ca instrumente de modelare a noțiunilor din limbajele imperative.

1989 .: Eugenio Moggi ține cursul CS359 la Stanford University, sub titlul “An Abstract View of Programming Languages”. În Cap 4 (pg. 25) sunt prezentate teoretic semanticile denotaționale și monadele.

1990 .: Nu am găsit nici o lucrare din domeniul “monadic programming”, liber disponibilă prin Internet.

1991 martie. : Stephen Robert Adams își susține teza de doctorat “Modular Grammars for Language Prototyping” (Univ. Southampton). La pagina 59 apare termenul “partial parser”. Stephen R. A. studiază gramaticile modulare cu atribute.

1992 – Aflat în anul al III-lea, Dan Popa se înscrie la cursul opțional de Semantică. Acesta se dovedește un curs teoretic, clasic, fără semantici monadice, însoțit de seminarii și laboratoare de ML.

1992 – ACM Sigplan 92 : Un volum în care există 3 articole despre continuări (eng. Continuations) inclusiv: “Three Monads For Continuations”. O semantică este dată ca exemplu de utilizare a monadelor (pg. 46). Rezultatele sunt teoretice.

1992 Philip Wadler publică în articolul “The essence of functional programming” [Wad-92b] o serie de back-end-uri monadice de interpretoare ilustrând ideea de plug-in monadic aplicat aceleiași descrieri a semanticii. Implementarea este realizată în Gofer iar monadele sunt folosite prin intermediul operațiilor “bindM” și “unitM”. Publică și alte lucrări: [Wad-92a],[Wad-92c].

1993 ian. : Graham Hutton și Errik Meijer publică “Monadic Parsing in Haskell” în JFP 1(1) . Practic este prezentată o bibliotecă de parsere modulare și combinatori de parsere asemănătoare ParseLib-ului. [Hut-93].

1993 – Ca urmare a unor discuții purtate prin internet Marck P.Jones și Luc Duponcheel prezintă lucrarea “Composing monads”, [Jon-93]. Monadele sunt introduse destul de sofisticat, folosindu-se *map*, *unit* și *join* împreună cu șapte legi algebrice (axiome). Compunerea monadelor era la acea dată o problemă dificilă, nerezolvată.

1994 ian. : “Building Interpreters by Composing Monads” ([Ste-94] de Guy. L. Steele Jr.)
Lucrarea propune un set de “Pseudomonade” și o stivă de tipuri ca suport pentru realizarea interpretoarelor, o soluție complicată și criticabilă. (ACM SIGPLAN-SIGACT Symp.)

1995 – Philip Wadler publică “How to Declare an Imperative” , [Wad95] o lucrare amplă despre monade pe care le introduce drept generalizări ale monoidului . Lucrarea oferă argumente pentru utilizarea monadelor ca mijloc de implementare a limbajelor imperative (atunci când se lucrează într-un limbaj funcțional pur.) Monadele fuseseră propuse – ca instrumente teoretice pentru reprezentarea limbajelor – de E. Moggi.

1995 : Sheng Liang , Paul Hudak, Mark P. Jones publică “Monads Transformers and Modular Interpreters” argumentând importanța morfismelor de monade implementate sub formă de transformatori de monade (eng. “Monad Transformers”).

1995 : David Espinosa își prezintă celebra teză “Semantic Lego” [Esp-95] argumentând posibilitatea implementării semanticii modulare monadice în limbajul Scheme. Lipsa do-notației (care va fi introdusă în Haskell 98) face ca acest demers să fie cumva înaintea vremurilor sale. Pentru a înlocui do-notația și a implementa proiecțiile operațiilor în monadă, D.Espinosa folosește un set de funcții de “lifting monadic”. Do notația din Haskell le va elimina, fiind mai practică.

1995 nov: Luc Duponcheel publică “Using catamorphism, subtypes and monad transformers for writing modular functional interpreters” [Dup-95] (Utrecht Univ.).

Catamorfismele vor fi și motiv de ironie pe pagina de web “Evolution of a Haskell Programmer” .

1996 – La Walla Walla College , Anthony A. Aaby pregătește și publică un tutorial de Haskell [Aab-96]. Vor mai apare și altele, în diverse părți ale lumii ulterior (nu le-am inclus în acest istoric).

1997 april. : William L. Harrison și Samuel N. Kamin (Dept. Comp. Sci, Univ. Illinois.) prezintă lucrarea : “Compilation as Partial Evaluation of Functor Category Semantics” [Har-97].

1997 Simon Marlow și Andy Gill lucrează la dezvoltarea unei implementări de generator de parsere pentru gramatici LALR, în Haskell. Programul a fost lansat sub numele de Happy.

1998 iul. : La Univ. Bristol își susține Christopher Bentley Dornan teza de doctorat cu titlul “Type Secure Meta – Programming” . Teza folosește Haskell-ul și monada de I/O , prezintă o viziune asupra compilării (pg. 79), noțiuni de programare modulară (pg. 93) iar drept aplicație oferă un mic file server (pg. 66). Anexele conțin: Dynamic semantics, do – notația , monadele (incl. Monada cu Stări și manipularea erorilor - “Error Handling State Monad”), o serie de reguli de tipizare, un generator de cod (pg. 87), un compilator (pg. 80).

1999 (aprox.) E publicată lucrarea “Fast, Error Correcting Parser Combinators – A Short Tutorial” de S.Doaitse Swierstra și Pablo R. Azero Akocer (Utrecht. Univ. Olanda resp. Univ. Major de San Simon, Cochabamba , Bolivia). Autorii explorează un set aparte de combinatori de parsere cu scopul realizării unui parser corector de erori.

2000 iun. : Apare “Recursive Monadic Bindings” de Levent Erkök și John Launchbury (Oregon. Grad. Institute of Sci. and Tech.) Dintre monade, sunt folosite: Monada Identitate , Monada Maybe (pg.28), Monada Listelor, (pg. 33) , Monada Stărilor (pg. 39). Autorii folosesc monadele la modelarea circuitelor electronice, nu la construcția interpretoarelor limbajelor.

2000 iul. : Apare "Writing High performance Server Applications in Haskell – Case Study : A Haskell Web Server" de Simon Marlow (MS. Research Limited Cambridge). Un web server este de fapt un interpretor pentru cereri trimise sub forma URL-urilor. Acestea formează, însă un limbaj de tip 3 (din ierarhia Chomsky) nu unul de tip 2 cum sunt majoritatea limbajelor de programare de care ne-am ocupat.

2001 Apare volumul "Grammars and Parsing" de Johan Jeuring și Doaitse Swierstra. Această carte ar fi trebuit, opiniei noastre, recomandată fiecărui doctorand care studiază domeniul limbajelor modulare, lucru imposibil deoarece doctorandul este adesea obligat să-și scrie singur bibliografia examenelor. Lucrarea prezintă printre altele semanticile algebrice implementate în Haskell (pp.121) și analiza sintactică a expresiilor folosind combinatori de parsere.

2001 Daan Leijen publică "Parsec – a fast combinator parser library " . Totodată oferă biblioteca Parsec, o bibliotecă de combinatori de parsere prevăzută cu un mecanism de semnalare a erorilor.

2002 Apare lucrarea "Tagless Staged Interpreters for Typed Languages" de Emir Pašalić, Walid Taha, Tim Sheard. Ea tratează probleme ale interpretării, folosind ca limbaj de implementare Meta-ML-ul. În lucrare este descris un mic limbaj, numit λH .

2002 martie. : Apare teza "Pure Functional Parsing – an advanced tutorial, de Peter Ljunglöf . Lucrarea conține și un capitol destinat unor combinatori de parsere cu backtracking (cap.3) și o serie de algoritmi clasici, implementați în Haskell. Nu există, prin natura temei alese, elementele de semantică monadică modulară de care ne ocupăm noi.

2002 oct. : "Composing Monads Using Coproducts" de Christoph Lüth, Neil Ghani. Abordează spinoasa problemă teoretică a compunerii monadelor (rezolvată negativ). Autorii folosesc noțiunea de coprodus.

2002 Apare "Monads and Modularity" de Christoph Lüth și Neil Ghani, (Springer.)

Aplicațiile propuse folosesc Monada Excepțiilor, Monada Stărilor , Monada Nedeterminismului.

2002 Tot în acest an apare “Some notes on Monads” de Andrea Schalk. Am utilizat o idee din această lucrare – cea a functorului dat ca exemplu la început – pentru a realiza lucrarea “An Assembler in a Nutshell”, despre asamblarele modulare și obținerea lor dintr-un functor, [Pop-05].

2003 nov. :Apare draft-ul volumului “Computational semantics and Type Theory” de Jan van Eijck. Capitolul 8.2 prezintă un set de combinatori de parsere notați cu paranteze unghiulare “<.>” în stilul de notație care este folosit de biblioteca Parsec.

2003 Apare lucrarea “Monads for Programming Languages”, [Shu03], de John N. Shutt, de la Worcester Polytechnic Institute.

2004 sept. : Graham Hutton publică manualul “Programming in Haskell” - [Hut-04], într-o primă formă (draft).

2004 Volumul “Higher Dimensional Categories – an illustrated guide book, Eugenia Cheng, Aaron Lauda este oferit prin internet.

2005 martie : Dan Popa începe să scrie “Introducere în Haskell 98 prin exemple”. - primul manual de Haskell din România, tipărit de o editură acreditată CNCSIS. Primele exemple din carte datează din 25 martie. În volum este prezentat limbajul Haskell și bazele construirii limbajelor folosind șablonul monadic de programare. Un studiu de caz despre interpretarea directă a expresiilor folosind parsere modulare din ParseLib este inclus în finalul volumului.

2005 aprilie – mai : Un capitol intitulat “Monad Transformers and Modular Interpreters” - tratând despre semantica interpretării modulare bazată pe o mașină virtuală înlocuită cu o monadă compusă (din “monad transformers” - implementări de morfisme de monade) este prezentat cursanților de la Stocholm University, Suedia, curs ținut de Johan Glimming. Titlul cursului: “Advanced Functional Programming”. În cursul al 12-lea sunt

citare "State Monad Transformer" și "Error Monad Transformer".

2005 mai. : Apare lucrarea de referință "Tackling The Awkward Squad: monadic input/output, concurrency, exceptions and foreign language call in Haskell" de Simon Peyton Jones, la capătul a cinci ani de cercetări și revizuirii. Lucrare de referință.

2005 : Apare "An Assembler in a Nutshell", în volumul - "Proceedings of the 30th annual Congress of The American – Romanian Academy of Art and Sciences , ASEM Moldova, pg. 196-198. [Pop-05]

2005 sept.: La Tallin, în Estonia, în cadrul ACM Haskell '05, Autrijus Tang prezintă lucrarea "Bootstrapping Perl 6 with Haskell". Nu rezolvă problema arborilor modulari dar folosește un fel de arbori cu variante, în ideea că printre variante se vor găsi și structurile necesare (binare, ternare, quaternare etc).

2005 oct : Lennart Anderson de la Univ. Lund, publică "Parsing with Haskell". [And-05]

2005 - Începe Anthony A. Aaby să scrie "The Lambda Calculus" [Aab-05].

2005 – Apare "The Haskell Programmer's Guide to the I/O Monad – Don't Panic" de Stefan Klinger. Este o dovadă a interesului pentru șablonul de programare monadic.

2005 Theodore Norvell publică "Monads For The Working Haskell Programmers" - a short tutorial (versiunea scrisă pentru programatorii în Haskell 98). http://www.engr.mun.ca/~theo/misc/haskell_and_monads

2006 – Apare lucrarea "Open Data Type and Open Functions" semnată de Andres Löh și Ralf Hinze (univ. Bonn). Deoarece arborii declarați cu data în Haskell nu oferă modularitatea așteptată de autori, aceștia propun să se extindă limbajul Haskell. Am abordat și noi această problemă dar am rezolvat-o în cadrul Haskell-ului, prin înlocuirea constructorilor de tip cu alte funcții utilizator. [Pop-08b]

2006 nov. : Apare un draft (neterminat) al volumului "Practical Foundations for

Programming Languages” de Robert Harper de la Carnegie Mellon Univ. Volumul este un tratat dezvoltat de sintaxă, semantică, teoria tipurilor, type checking, polimorfism, control, continuări, logică, paralelism. Capitolul despre “I/O monad” - monada de I/O – nu este scris, fiind format doar dintr-o pagină de titlu.

2007 ian. : La conferința HOPL III (“History of Programming Languages Conference”) se primește articolul “A History of Haskell” semnat de Paul Hudak, John Hughes, Simon Peyton Jones.

2007 ian. : Se tipărește și se distribuie cu ajutorul Editurii Edusoft – Bacău (Editură acreditată CNCSIS) “Introducere în Haskell 98 prin exemple” [Pop-07] . Cartea cuprinde capitole și subcapitole speciale, cu rolul expres de a introduce cititorul în tehnicile de construire a parserelor modulare monadice și a semanticilor monadice transcrise în notație. (ISBN 978-973-8934-48-1)

2007 : O echipă a proiectului Wikibooks grupată în jurul lui Jonathan Tang publică : “Write Yourself a Scheme in 48 Hours”, cu un subtitlu inspirat de versiunea engleză a titlului cărții subsemnatului: “An Introduction to Haskell through Example”. Volumul conține o implementare de Scheme , cu parsere modulare bazate pe ParseSec. Acest proiect își are începutul în niște lucrări ale lui Jonathan Tang din anul 2006. http://en.wikibooks.org/wiki/Write_Yourself_a_Scheme_in_48_Hours.

2008 april. : Torben Ægidius Mogensen (re)publică într-o ediție revizuită “Basic of Compiler Design” . Volumul nu conține nici o prezentare a tehnicilor de “monadic parsing” sau “monadic semantics”. Volumul apare la Dept.Comp.Sci. - Univ. Copenhaga. Deși noi considerăm că sunt tehnici care vor fi considerate în viitor fundamentale, neincluderea lor în volum o interpretăm ca pe o dovadă a modernității lor.

2008 august. : Este lansat limbajul didactic Rodin implementat în Haskell folosind șablonul monadic, parsere modulare, semantică monadică și arbori modulari. <Http://www.haskell.org/haskellwiki/Rodin>

2008 oct : La a XII-a Conferință anuală a SȘMR Dan Popa prezintă lucrarea “Modular

evaluation and interpreters using monads and type classes” în care prezintă printre altele și o soluție la problema modularității arborilor sintactici, fără să mai fie nevoie să extindă limbajul Haskell.

2008 oct : Limbajul Rodin (un DSL didactic scris în Haskell folosind tehnicile de mai sus) este folosit ca material didactic la Univ. Bacău, la secția de Matematică.

2008 oct-nov. : Dan Popa publică la Editura MatrixRom “Practica interpretării monadice”, cu o mică întârziere față de estimarea inițială.

Notă: O serie de articole publicate sub egida: Springer Verlag, ACM , Elsevier Science nefiind accesibile decât contra cost nu sunt incluse. De asemenea nu sunt incluse cercetările INRIA (din Franța).