

Towards a Mathematical Operational Semantics

Daniele Turi Gordon Plotkin

~ 1997

Gordon Plotkin



Figure : <http://wcms.inf.ed.ac.uk/lfcs/people>

Gordon Plotkin

≡ Google Scholar

 Gordon Plotkin

Laboratory for Foundations of Computer Science, School of Informatics, University of Edinburgh
Verified email at inf.ed.ac.uk

[FOLLOW](#) [GET MY OWN PROFILE](#)

TITLE	CITED BY	YEAR
A structural approach to operational semantics GD Plotkin Computer Science Department, Aarhus University	4562	1981
A framework for defining logics R Harper, F Honsell, G Plotkin Journal of the ACM (JACM) 40 (1), 143-184	1901	1993
Call-by-name, call-by-value and the λ -calculus GD Plotkin Theoretical computer science 1 (2), 125-159	1657	1975
LCF considered as a programming language GD Plotkin Theoretical computer science 5 (3), 223-255	1445	1977
A note on inductive generalization GD Plotkin Machine intelligence 5 (1), 153-163	1436	1970
Petri nets, event structures and domains, part I M Nielsen, G Plotkin, G Winskel Theoretical Computer Science 13 (1), 85-108	1347 *	1981
A powerdomain construction GD Plotkin SIAM Journal on Computing 5 (3), 452-487	842	1976
A calculus for access control in distributed systems M Abadi, M Burrows, B Lampson, G Plotkin ACM Transactions on Programming Languages and Systems (TOPLAS) 15 (4), 706-734	774	1993
The category-theoretic solution of recursive domain equations MB Smyth, GD Plotkin SIAM Journal on Computing 11 (4), 761-783	713	1982
Abstract types have existential type JC Mitchell, GD Plotkin ACM Transactions on Programming Languages and Systems (TOPLAS) 10 (3), 470-502	712	1988

Cited by [VIEW ALL](#)

	All	Since 2015
Citations	27677	5162
h-index	62	36
i10-index	145	81



Co-authors [VIEW ALL](#)

 Martin Abadi Principal Scientist, Google, and ...
 John Power Reader in Computer Science, Un...
 Glynn Winskel University of Cambridge
 Mogene Nielsen Professor of Computer Science, ...
 Luca Cardelli University of Oxford
 Marcelo Fiore Professor in Mathematical Found...
 Martin Hyland Professor of Mathematical Logic,...



Daniele Turi



Figure : <http://www.dcs.ed.ac.uk/home/dt/ellie.html>

Daniele Turi

≡ Google Scholar

SEARCH

SIGN IN



Daniele Turi

CTO, AnyJunk Ltd, London, UK

Verified email at anyjunk.co.uk

Semantics Semantic web Personalisation Machine Learning

FOLLOW

GET MY OWN PROFILE

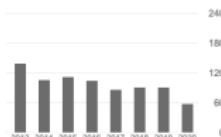
Cited by

VIEW ALL

All

Since 2015

Citations	2980	539
h-index	23	12
i10-index	25	12



TITLE

CITED BY

YEAR

Abstract syntax and variable binding

M Fiore, G Plotkin, D Turi

Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158 ...)

437

1999

Towards a mathematical operational semantics

D Turi, G Plotkin

Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science ...

423

1997

The first provability challenge

L Moreau, B Lüdke, I Attali, RS Barge, S Bowers, S Callahan, ...

Concurrency and computation: practice and experience 20 (5), 409-418

269

2008

The instance store: DL reasoning with large numbers of individuals

I Horrocks, LLL Turi, S Bechhofer

Proc. of the 2004 Description Logic Workshop (DL 2004), 31-40

208

2004

Mining Taverna's semantic web of provenance

J Zhao, C Cobb, R Stevens, D Turi

Concurrency and Computation: Practice and Experience 20 (5), 463-472

154

2008

Initial algebra and final coalgebra semantics for concurrency

J Rutten, D Turi

Workshop/School/Symposium of the REX Project (Research and Education in ...)

153

1993

Semantics of name and value passing

M Fiore, D Turi

Proceedings 18th Annual IEEE Symposium on Logic in Computer Science, 93-104

152

2001

The foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces

J Rutten, D Turi

Mathematical Structures in Computer Science 8 (5), 481-540

130

1998

Taverna^{my}Grid: Aligning a Workflow System with the Life Sciences Community

T Oinn, P Li, DB Kell, C Goble, A Goderis, M Greenwood, D Hull, ...

Workflows for e-Science, 300-319

128

2007

On the foundations of final semantics: non-standard sets, metric spaces, partial orders

I Huai-Duc Turi, D Turi

123

1992

Semantics

Syntax Domain
 $\xrightarrow{\text{denotational, operational}}$

Example 1: Regular Expressions (not in the paper)

$$\text{RExp}_A^1 \xrightarrow{\text{denotational, operational}} \mathcal{P}(A^*)$$

¹ $e, f \in \text{RExp}_A ::= a \in A \mid 0 \mid 1 \mid e \cdot f \mid e + f \mid e^*$

Example 1: Regular Expressions (denotational)

For $L_1, L_2 \subseteq A^*$ define

$$L_1 \cdot L_2 \stackrel{\text{def}}{=} \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

$$L_1 + L_2 \stackrel{\text{def}}{=} L_1 \cup L_2$$

$$L^* \stackrel{\text{def}}{=} \bigcup_{n \geq 0} L^n = \{w_1 \cdot \dots \cdot w_n \mid n \geq 0, w_i \in L, 1 \leq i \leq n\}$$

where $L^0 \stackrel{\text{def}}{=} \{\varepsilon\}$ and $L^{n+1} \stackrel{\text{def}}{=} L \cdot L^n$.

The *standard interpretation* is given by the unique homomorphism

$$\text{de} : (\text{RExp}_A, a_{a \in A}, 0, 1, \cdot, +, *) \rightarrow (\mathcal{P}(A^*), \{a\}_{a \in A}, \emptyset, \{\varepsilon\}, \cdot, +, *).$$

Example 1: Regular Expressions (operational)

Define

$$E : \text{RExp}_A \rightarrow 2 \quad D_a : \text{RExp}_A \rightarrow \text{RExp}_A, \quad a \in A$$

as follows:

$$E(b) \stackrel{\text{def}}{=} 0, \quad b \in A$$

$$D_a(b) \stackrel{\text{def}}{=} [a = b], \quad b \in A$$

$$E(0) \stackrel{\text{def}}{=} 0$$

$$D_a(0) \stackrel{\text{def}}{=} 0$$

$$E(1) \stackrel{\text{def}}{=} 1$$

$$D_a(1) \stackrel{\text{def}}{=} 0$$

$$E(e \cdot f) \stackrel{\text{def}}{=} E(e) \cdot E(f)$$

$$D_a(e \cdot f) \stackrel{\text{def}}{=} D_a(e) \cdot f + E(e) \cdot D_a(f)$$

$$E(e + f) \stackrel{\text{def}}{=} E(e) + E(f)$$

$$D_a(e + f) \stackrel{\text{def}}{=} D_a(e) + D_a(f)$$

$$E(e^*) \stackrel{\text{def}}{=} 1$$

$$D_a(e^*) \stackrel{\text{def}}{=} D_a(e) \cdot e^*.$$

The *language interpretation* is given by the unique homomorphism

$$\text{op} : (\text{RExp}_A, E, (D_a)_{a \in A}) \rightarrow (\mathcal{P}(A^*), \varepsilon, (\delta_a)_{a \in A})^2.$$

$${}^2\varepsilon(L) = [\varepsilon \in L], \quad \delta_a(L) = \{w \mid a \cdot w \in L\}, \quad a \in A$$

Example 1: Regular Expressions

We have established

$$\text{op} : (\text{RExp}_A, E, (D_a)_{a \in A}) \rightarrow (\mathcal{P}(A^*), \varepsilon, (\delta_a)_{a \in A})$$

$$\text{de} : (\text{RExp}_A, a_{a \in A}, 0, 1, \cdot, +, *) \rightarrow (\mathcal{P}(A^*), \{a\}_{a \in A}, \emptyset, \{\varepsilon\}, \cdot, +, *).$$

Some questions:

- ▶ Is the equality $\text{op} = \text{de}$ given?
- ▶ Are op and de also homomorphisms of the respective other type?
- ▶ In general, how can we ensure such properties?

Main idea

To derive an *adequate semantics*:

Relate syntax and semantics by a distributive law.

Example 1: Regular Expressions

Let $\Sigma, B : \text{Set} \rightarrow \text{Set}$ satisfy

$$\Sigma X = A + 1 + 1 + X^2 + X^2 + X \quad BX = X^A \times 2.$$

Then one observes

$(\text{RExp}_A, a_{a \in A}, 0, 1, \cdot, +, *)$: initial Σ -algebra

$(\mathcal{P}(A^*), \varepsilon, (\delta_a)_{a \in A})$: final B -coalgebra.

We defined **morphisms** and obtained

$$\begin{array}{ccc} \Sigma \text{RExp}_A & \xrightarrow{\Sigma \text{de}} & \Sigma \mathcal{P}(A^*) \\ \downarrow & & \downarrow \\ \text{RExp}_A & \xrightarrow{\text{de}} & \mathcal{P}(A^*) \end{array} \quad \begin{array}{ccc} \text{RExp}_A & \xrightarrow{\text{op}} & \mathcal{P}(A^*) \\ \downarrow & & \downarrow \\ B \text{RExp}_A & \xrightarrow{B \text{op}} & B \mathcal{P}(A^*) \end{array} .$$

Instead of *defining* above **morphisms**, we can *deduce* them from $\lambda : \Sigma B \Rightarrow B \Sigma$.

Full picture (simple)

Let $\Sigma, B : \mathbf{C} \rightarrow \mathbf{C}$ be endofunctors.

Let $(\mu\Sigma, h)$ denote the initial Σ -algebra, and let $(\nu B, k)$ denote the final B -coalgebra.

Let $\lambda : \Sigma B \rightarrow B\Sigma$ be a natural transformation.

By initiality and finality there are unique morphisms

$$\begin{array}{ccc} \Sigma(\mu\Sigma) & \dashrightarrow^{\Sigma!} & \Sigma B(\mu\Sigma) \\ h \downarrow & & \downarrow Bh \circ \lambda_{\mu\Sigma} \\ \mu\Sigma & \dashrightarrow^{\text{!}} & B(\mu\Sigma) \end{array} \quad \begin{array}{ccc} \Sigma(\nu B) & \dashrightarrow^{\text{!}} & \nu B \\ \lambda_{\nu B} \circ \Sigma k \downarrow & & \downarrow k \\ B\Sigma(\nu B) & \dashrightarrow^{B\text{!}} & B(\nu B). \end{array}$$

Full picture (simple)

One observes that

$$\langle \Sigma(\mu\Sigma) \xrightarrow{h} \mu\Sigma \xrightarrow{!} B(\mu\Sigma) \rangle \quad \langle \Sigma(\nu B) \xrightarrow{!} \nu B \xrightarrow{k} B(\nu B) \rangle$$

are initial and final, respectively, along tuples of algebra and coalgebra morphisms related by λ .

Thus we find

$$\begin{array}{ccc} \Sigma(\mu\Sigma) & \xrightarrow{\Sigma^{\text{op}}=\Sigma^{\text{de}}} & \Sigma(\nu B) \\ h \downarrow & & \downarrow ! \\ \mu\Sigma & \xrightarrow{\text{op}= \text{de}} & \nu B \\ ! \downarrow & & \downarrow k \\ B(\mu\Sigma) & \xrightarrow{B^{\text{op}}=B^{\text{de}}} & B(\nu B) \end{array}$$

Full picture (hard)

The picture complicates if one allows variations in the sense of

$$e \in \text{RExp}_A X ::= x \in X \mid e \in \text{RExp}_A.$$

Mathematically, we move from endofunctors to monads and comonads.

Every endofunctor freely generates a monad and a comonad.

Freely generated monads/comonads

Let $\Sigma, B : \mathbf{C} \rightarrow \mathbf{C}$ be endofunctors.

The following definitions induce a monad T_Σ and a comonad D_B on \mathbf{C} ,

$$T_\Sigma X = \mu(\lambda Y. X + \Sigma Y) \quad D_B X = \nu(\lambda Y. X \times BY).$$

There exist equivalences

$$\text{Alg}(\Sigma) \simeq \text{Alg}(T_\Sigma) \quad \text{Coalg}(B) \simeq \text{Coalg}(D_B)$$

satisfying

$$(\mu\Sigma, h) \mapsto (T_\Sigma 0, \mu_0) \quad (\nu B, k) \mapsto (D_B 1, \delta_1).$$

Distributive laws

A *distributive law* between a monad $\langle T, \eta, \mu \rangle$ and a comonad $\langle D, \varepsilon, \delta \rangle$ is a natural transformation

$$\lambda : TD \Rightarrow DT$$

satisfying the laws

$$\lambda \circ \eta_D = D\eta \quad \lambda \mu_D = D\mu \circ \lambda_T \circ T\lambda$$

and their dual

$$T\varepsilon = \varepsilon_T \circ \lambda \quad D\lambda \circ \lambda_D \circ T\delta = \delta_T \circ \lambda.$$

Distributive laws

There exist liftings T_λ and D_λ

$$\begin{array}{ccc} \text{Coalg}(D) & \xrightarrow{T_\lambda} & \text{Coalg}(D) \\ U_D \downarrow & & \downarrow U_D \\ C & \xrightarrow{T} & C \end{array} \quad \begin{array}{ccc} \text{Alg}(T) & \xrightarrow{D_\lambda} & \text{Alg}(T) \\ U_T \downarrow & & \downarrow U_T \\ C & \xrightarrow{D} & C \end{array}$$

satisfying

$$T_\lambda(X \xrightarrow{k} DX) = TX \xrightarrow{\lambda_X \circ T k} DTX$$

$$D_\lambda(TX \xrightarrow{h} X) = TDX \xrightarrow{D h \circ \lambda_X} DX.$$

In fact, liftings of T to $\text{Coalg}(D)$, liftings of D to $\text{Alg}(T)$, and distributive laws coincide.

Bialgebras

A λ -bialgebra is an object with both a T -algebra and a D -coalgebra structure

$$\langle TX \xrightarrow{h} X \xrightarrow{k} DX \rangle,$$

satisfying

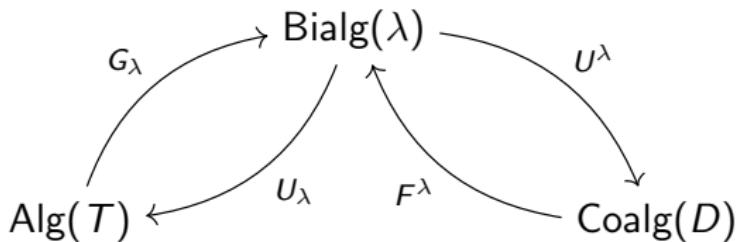
$$\begin{array}{ccc} TX & \xrightarrow{Tk} & TDX \\ h \downarrow & & \downarrow \lambda_X \\ X & \xrightarrow{k} & DX \end{array} .$$
$$\downarrow Dh$$

There exist the following equivalences of categories

$$\text{Alg}(T_\lambda) \simeq \text{Bialg}(\lambda) \simeq \text{Coalg}(D_\lambda).$$

Bialgebras

There are free-forgetful adjunctions $F^\lambda \dashv U^\lambda$ and $U_\lambda \dashv G_\lambda$



where

$$G_\lambda(TX \xrightarrow{h} X) = \langle TD\!X \xrightarrow{D_\lambda(h)} D\!X \xrightarrow{\delta_X} D^2\!X \rangle$$

$$F^\lambda(X \xrightarrow{k} DX) = \langle T^2\!X \xrightarrow{\mu_X} TX \xrightarrow{T_\lambda(k)} DT\!X \rangle.$$

$F^\lambda(0 \xrightarrow{!_{D^0}} D0)$ is initial, and $G_\lambda(T1 \xrightarrow{!_{T1}} 1)$ is final.

Adequate semantics

By initiality of $F^\lambda(0 \xrightarrow{!D_0} D_0)$ and finality of $G_\lambda(T1 \xrightarrow{!T_1} 1)$:

$$\begin{array}{ccc} T^20 & \xrightarrow{T_{\text{op}}=T_{\text{de}}} & TD1 \\ \mu_0 \downarrow & & \downarrow D_\lambda(!T_1) \\ T0 & \xrightarrow{\text{op}=de} & D1 \\ T_\lambda(!D_0) \downarrow & & \downarrow \delta_1 \\ DT0 & \xrightarrow{D_{\text{op}}=D_{\text{de}}} & D^21. \end{array}$$

In particular, note

$$T_\Sigma 0 = \mu \Sigma \quad D_B 1 = \nu B,$$

and compare above with the simpler picture before.

Example 2: GSOS

Given a finite set A , consider the following language

$$e \in T_{\Sigma}X ::= x \in X \mid 0 \mid a.e \mid e \parallel e$$

with operational semantics

$$\mathcal{R} = \left\{ \frac{}{a.e \xrightarrow{a} e}, \quad \frac{e \xrightarrow{a} f}{e \parallel g \xrightarrow{a} f \parallel g}, \quad \frac{e \xrightarrow{a} f}{g \parallel e \xrightarrow{a} g \parallel f} \right\}.$$

Above rules generate the smallest relation $R \subseteq T_{\Sigma}X \times A \times T_{\Sigma}X$ satisfying

$$(e, a, f) \in R \Leftrightarrow \exists \text{ Proof : } \frac{\cdots}{e \xrightarrow{a} f}.$$

Example 2: GSOS

In other words, if

$$\Sigma X = 1 + A \times X + X \times X \quad BX = (\mathcal{P}_{\text{fi}} X)^A = (1 + \mathcal{P}_f X)^A$$

then $(T_\Sigma X, R)$ is a B -coalgebra.

Same question as before:

How do we ensure an adequate semantics?

Typically answered by:

Constraints on the format of \mathcal{R} , such as GSOS.

Example 2: GSOS \Rightarrow DL (very short)

$$\begin{array}{c} \mathcal{R} \\ \downarrow \\ \rho : \Sigma(\text{Id} \times B) \Rightarrow BT_\Sigma \\ \downarrow \\ \varrho : \Sigma(T_\Sigma \times BT_\Sigma) \Rightarrow BT_\Sigma \\ \downarrow \\ (\varrho : T_\Sigma)^\varrho : \text{Coalg}(B) \rightarrow \text{Coalg}(B) \\ \downarrow \\ \lambda : T_\Sigma D_B \Rightarrow D_B T_\Sigma. \end{array}$$