# Determinization and Bialgebraic Semantics*

**Luís Monteiro**

**NOVA-LINCS and Departamento de Informática**
**Faculdade de Ciências e Tecnologia, 2829-516 Monte de Caparica, Portugal**
`lm@fct.unl.pt`

Two main subjects, determinization [9, 6] and bialgebraic semantics [10, 11], are combined in this paper to produce both operational models and adequate denotational semantics for process calculi in a coalgebraic setting. The idea is to extend bialgebraic semantics to capture behaviours that do not constitute a final coalgebra for the given behaviour functor but are still sufficiently "well-behaved" to be tractable, like the behaviours that result from determinization.

Bialgebraic semantics is based on functorial notions of syntax and behaviour, leading to an operational semantics and a canonical adequate denotational semantics. This is basically the work of Turi and Plotkin, appearing first as Turi's Ph.D. thesis [10] and soon afterwards in a joint publication [11]. An "abstract programming language" is defined by a signature $\Sigma$ for the operators of the language, also seen as an endofunctor on **Set** in the usual way. The language defined by $\Sigma$ is $T0$, where $(T, \eta^T, \mu^T)$ is the free monad generated by $\Sigma$ and 0 stands for the empty set. The operational meaning of the language is described in two steps. First, an endofunctor $B$ on **Set** defines the category of $B$-coalgebras where the intended behaviours take place. Second, $T0$ is turned into a $B$-coalgebra $(T0, \psi_0)$ by specifying operational (SOS) rules for the operators of the language. For rules in the GSOS format [3], Turi and Plotkin have shown that they can be described by an appropriate distributive law involving the functors $T$ and $B$. Assuming there is a final $B$-coalgebra $(Z, \zeta)$, the operational semantics of the language $T0$ is the unique $B$-morphism from $(T0, \psi_0)$ to $(Z, \zeta)$. Now $T0$ is the carrier of an initial $T$-algebra $(T0, \mu_0^T)$. The operational rules of the language, on the other hand, allow once again to define a $B$-coalgebra $(TZ, \psi_Z)$, and the only $B$-morphism $\beta$ to $(Z, \zeta)$ gives rise to another $T$-algebra $(Z, \beta)$. The only $T$-homomorphism from $(T0, \mu_0^T)$ to $(Z, \beta)$ is then the denotational semantics of $T0$ derived from the operationality of the language. It turns out that the operational and the denotational semantics coincide; in other words, the denotational semantics is adequate and the operational semantics is compositional. This semantics is called "bialgebraic" because $T0$ and $Z$ are bialgebras, that is, are simultaneously $B$-coalgebras and $T$-algebras. This type of situation is typical in Computer Science and a number of publications have exploited some of its applications or developed further the theory [2, 5, 7, 1].

One limitation of the bialgebraic machinery is that it is restricted to the behaviours that constitute the final coalgebra of the behaviour functor. Several proposals to tackle other types of behaviours coalgebraically have been pursued for some time now [4, 8, 7, 6, 9], and in particular the work in [7] is conducted in the bialgebraic setting. Given the variety of behaviours that have been obtained through determinization [9], the extension of the bialgebraic techniques to the determinization setting is only natural and is the main goal of this paper.

Determinization is a generalization to the coalgebraic framework of the familiar powerset construction that builds a deterministic automaton from a nondeterministic one. It consists in assuming that the "behaviour" functor $B$ has a decomposition $B = FP$, where $F$ is a functor for which a final $F$-coalgebra $\mathbb{Z}$ exists and $P$ is (the underlying functor of) a monad;

* This work was partially supported by project QAIS, PTDC/EIA-CCO/122240/2010, FCT, Portugal.


Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

furthermore, there is a distributive law $\lambda^{PF} : PF \Rightarrow FP$ connecting both functors. The distributive law allows to lift an $FP$-coalgebra $\mathbb{X}$ with state space $X$ to an $F$-coalgebra $P^\sharp\mathbb{X}$ with state space $PX$; the unique $F$-morphism $[\![\cdot]\!]_{P^\sharp\mathbb{X}}$ from $P^\sharp\mathbb{X}$ to the final $F$-coalgebra $\mathbb{Z}$, composed with the unit of the monad $P$, gives what is called in [6] the "extension" (operational) semantics $\beta_{\mathbb{X}} = [\![\cdot]\!]_{P^\sharp\mathbb{X}} \circ \eta^P_X$ of $\mathbb{X}$; we also call $\beta_{\mathbb{X}}$ the "$F$-behaviour" of $\mathbb{X}$.

To adapt bialgebraic semantics to the determinization setting we proceed as follows. As before, given $\Sigma$ and operational rules for its operators, there is an $FP$-coalgebra $(T0, \psi_0)$, and $\beta_{T0}$ is the operational semantics of $T0$ in terms of $F$-behaviours. For the denotational semantics, a distributive law $\lambda^{TP} : TP \Rightarrow PT$ is assumed. The monads $P$ and $T$ compose to give a monad $PT$, and any $PT$-algebra gives rise to a $T$-algebra on the same carrier. The crucial step is to define a $PT$-algebra on $Z$, and therefore also a $T$-algebra, by the previous remark; it turns out that the operation of this algebra is $\beta_{TZ} : TZ \to Z$, the $F$-behaviour derived from the determinization of $(TZ, \psi_Z)$. The denotational semantics of $(T0, \mu^T_0)$ is then the unique $T$-homomorphism to $(Z, \beta_{TZ})$. We prove—under the hypothesis that $\lambda^{TP}$ is a solution of a certain guarded recursive equation, in the terminology of [2]—that this $T$-homomorphism is precisely $\beta_{T0}$, thus establishing the adequacy of the denotational semantics with respect to the extension or $F$-behaviour semantics. We illustrate the approach with an outline of (strong) failure semantics for a CSP-like fragment without $\tau$-transitions.

## References


1. Adriana Balan and Alexander Kurz. On coalgebras over algebras. *Theoretical Computer Science*, 412(38):4989–5005, 2011.
2. Falk Bartels. *On Generalised Coinduction and Probabilistic Specification Formats.* PhD thesis, Free Univ. Amsterdam, 2004.
3. Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.
4. Michele Boreale and Fabio Gadducci. Processes as formal power series: A coinductive approach to denotational semantics. *Theor. Comput. Sci.*, 360(1-3):440–458, 2006.
5. Bart Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, pages 375–404, 2006.
6. Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. In *Proceedings of the 11th International Workshop on Coalgebraic Methods in Computer Science (CMCS 2012)*, volume 7399 of *Lecture Notes in Computer Science*, pages 109–129, 2012.
7. Bartek Klin. Bialgebraic methods and modal logic in structural operational semantics. *Inf. Comput.*, 207(2):237–257, 2009.
8. Luís Monteiro. A coalgebraic characterization of behaviours in the linear time - branching time spectrum. In Andrea Corradini and Ugo Montanari, editors, *Recent Trends in Algebraic Development Techniques (WADT 2008)*, volume 5486 of *Lecture Notes in Computer Science*, pages 250–264. Springer, 2009.
9. Alexandra Silva, Filippo Bonchi, Marcello Bonsangue, and Jan Rutten. Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science*, 9(1):1–27, 2013.
10. Daniele Turi. *Functorial Operational Semantics and its Denotational Dual.* PhD thesis, Free Univ. Amsterdam, 1996.
11. Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *Proc. 12th Annual IEEE Symposium on Logic in Computer Science*, pages 280–291. IEEE Computer Society Press, 1997.