Generic Expressions from Predicate Liftings*

Ulrich Dorsch, Stefan Milius, and Lutz Schröder

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

This work contributes to the coalgebraic study of generic expression languages for specifying the behaviour of finite coalgebras. The goals of this line of work are a generic Kleene-type theorem, sound and complete axiomatizations of behavioural equivalence and complexity bounds for equivalence checking.

Previous work by Silva et al. (e.g. [5]) gives generic expression calculi as well as sound and complete axiomatizations for an inductively defined class of set functors, and Myers [4] bases his generic expression calculi on presentations of functors on locally finite varieties.

Our idea is to use the framework of coalgebraic modal logic, and at the moment we restrict ourselves to endofunctors on **Set**. To define the semantics of our generic expressions we use predicate liftings; recall that an n-ary *predicate lifting* for a **Set**-functor T is a natural transformation

$$\lambda : \mathcal{Q}^n \Rightarrow \mathcal{Q}T^{op}$$

where $\mathcal{Q}$ is the contravariant powerset functor. The expressions themselves are formed using a simple grammar that permits fixed point constructions and modalities which are supposed to capture the 1-step behaviour of states in coalgebras.

▶ **Definition 1** (Expression Language)**.** The *expression language* $\mathcal{E}(\mathcal{L})$ is defined by the grammar

$$\varphi ::= z \mid \nu z.\varphi \mid \mathrm{L}\left(\varphi_1, \dots, \varphi_{\mathrm{ar(L)}}\right)$$

with the greatest fixed point operator ν, z and L ranging over a set of fixed point variables and a set $\mathcal{L}$ of modalities respectively.

We will only consider the *closed* (all fixed point variables bound) and *guarded* (all fixed point variables separated by at least one modality from their binding operator) expressions $\mathcal{E}_0(\mathcal{L})$ for technical reasons.

As in coalgebraic modal logics, the semantics of expressions is defined by giving for each modality $\mathrm{L} \in \mathcal{L}$ a predicate lifting $[\![L]\!] \in \Lambda$ with matching arity. At this point we require the predicate liftings to be *monotone* so that Tarski's fixed point theorem guarantees the existence of the greatest fixed point.

Another way how to define the semantics of expressions is the one from [5]: One defines a T-coalgebra structure on $\varepsilon : \mathcal{E}_0(\mathcal{L}) \to T\mathcal{E}_0(\mathcal{L})$:

$$\varepsilon\left(\mathrm{L}\left(\varphi_1, \dots, \varphi_{\mathrm{ar(L)}}\right)\right) \in [\![\mathrm{L}]\!]_{\mathcal{E}_0(\mathcal{L})}\left(\{\varphi_1\}, \dots, \{\varphi_{\mathrm{ar(L)}}\}\right), \qquad \varepsilon\left(\nu z.\varphi\right) = \varepsilon\left(\varphi\left[\nu z.\varphi/z\right]\right).$$

For this to be well-defined we need an assignment of *singleton preserving* predicate liftings $[\![\mathrm{L}]\!]$ to modalities as above. The semantics of expressions is now defined via behavioural equivalence.

* This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under projects MI 717/5-1 and SCHR 1118/11-1

Calco Early Ideas.
Editors: Billy Editor and Bill Editors; pp. 1–2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We can prove that the two definitions of semantics coincide in case the predicate liftings employed are both monotone and singleton preserving; both properties are also required for our main result.

As a first step we show that our expression language captures behavioural equivalence.

▶ **Theorem 2.** *Let Λ be a set of description preserving monotone predicate liftings for a functor T. Then every expression $\varphi \in \mathcal{E}_0(\mathcal{L})$, describes exactly one behavioural equivalence class in any T-coalgebra.*

If we additionally require the set Λ to be *separating*, i.e. every $t \in TX$ is uniquely determined by the set $\left\{(\lambda, A) \mid \lambda \in \Lambda, A \in X^{\mathrm{ar}(\lambda)}, t \in \lambda(A)\right\}$, then every expression describes exactly one behavioural equivalence class (cf. Λ-bisimulation in [2]).

For our main result we need another property, stronger than separating, for our set of predicate liftings: A set Λ of predicate liftings for a functor T is called *strongly expressive* if

$$\forall t \in TX \,.\, \exists \lambda \in \Lambda, \left(x_1, \ldots x_{\mathrm{ar}(\lambda)}\right) \in X^{\mathrm{ar}(\lambda)} \,.\, \{t\} = \lambda\left(\{x_1\}, \ldots, \left\{x_{\mathrm{ar}(\lambda)}\right\}\right).$$

We can now state our Kleene-type theorem, generic in nature although we have to restrict ourselves to functors that permit sets of predicate liftings with the above properties.

▶ **Theorem 3.** *Let Λ be a strongly expressive set of description preserving monotone predicate liftings for a functor T and (X, ξ) a T-coalgebra with finite carrier. Then for all $x \in X$, there exists a $\varphi \in \mathcal{E}_0(\mathcal{L})$ with*

$$\forall y \in X \,.\, y \in [\![\varphi]\!] \iff y \sim x.$$

We have also investigated connections between predicate liftings with the above properties and functor presentations: The direction from strongly expressive sets of predicate liftings to functor presentations is straightforward; conversely, the Moss liftings from the work of Marti and Venema [3] already have the required properties. Therefore we can rely on several results from Myers [4] which are based on functor presentations, especially with regard to a sound and complete axiomatization of behavioural equivalence.

As far as complexity goes, we do not have any concrete results yet. But since our expression language is basically the coalgebraic μ-calculus without propositional symbols and any kind of boolean connectives we have EXPTIME as upper bound for equivalence checking, see Cirstea et al. [1]. Because we have only a fragment of coalgebraic μ-calculus and the properties that our predicate liftings need to have are rather strong we conjecture that equivalence checking for our language is in PTIME.

References

1 Corina Cîrstea, Clemens Kupke, and Dirk Pattinson. EXPTIME tableaux for the coalgebraic mu-calculus. *Logical Methods in Computer Science*, 7(3), 2011.

2 Daniel Gorín and Lutz Schröder. Simulations and bisimulations for coalgebraic modal logics. In *Algebra and Coalgebra in Computer Science – 5th International Conference, CALCO 2013, Warsaw, Poland, September 3-6, 2013. Proceedings*, pages 253–266, 2013.

3 Johannes Marti and Yde Venema. Lax extensions of coalgebra functors and their logic. *J. Comput. Syst. Sci.*, 81(5):880–900, 2015.

4 Robert S. R. Myers. *Rational Coalgebraic Machines in Varieties: Languages, Completeness and Automatic Proofs.* PhD thesis, Imperial College London, 2013.

5 Alexandra Silva, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Non-deterministic Kleene coalgebras. *Logical Methods in Computer Science*, 6(3), 2010.