# Quantitative regular behaviours

Alexandra Silva[2], joint work with Filippo Bonchi[2], Marcello Bonsangue[1,2], and Jan Rutten[2,3]

[1] LIACS - Leiden University
[2] Centrum voor Wiskunde en Informatica (CWI)
[3] Vrije Universiteit Amsterdam (VUA)
{ams,bonchi,janr}@cwi.nl
marcello@liacs.nl

We present a systematic way to generate (1) languages of (generalised) regular expressions, and (2) sound and complete axiomatizations thereof, for a wide variety of quantitative systems. Our quantitative systems include weighted versions of automata and transition systems, in which transitions are assigned a value in a monoid that represents cost, duration, probability, etc. Such systems are represented as coalgebras and (1) and (2) above are derived in a modular fashion from the underlying (functor) type of these coalgebras.

In previous work [1, 2], we applied a similar approach to a class of systems (without weights), coalgebras for the so-called Kripke-polynomial functors, that generalizes both the results of Kleene (on rational languages and DFA's) and Milner (on regular behaviours and finite LTS's), and includes many other systems such as Mealy and Moore machines.

We now enlarge this framework to deal with quantitative systems. To this end, we give a non-trivial extension of the class of Kripke-polynomial functors by adding a functor type that allows the transitions of our systems to take values in a *monoid* structure of quantitative values. This new class, which we shall call quantitative functors, now includes weighted automata and several probabilistic systems. At the same time, we extend our earlier approach to the new setting. This will allow us to derive for each functor in our new extended class everything we were after: a language of regular expressions; a corresponding Kleene Theorem; and a sound and complete axiomatization for the corresponding notion of behavioural equivalence.

As a consequence, our results now include languages and axiomatizations, both existing and new ones, for weighted automata and many different kinds of probabilistic systems. Table 1 shows three examples of derived languages and axiomatizations.

## References

1. M. Bonsangue, J. Rutten, and A. Silva. A Kleene theorem for polynomial coalgebras. In *FoSSaCS*, 2009. To appear.
2. M. Bonsangue, J. Rutten, and A. Silva. An algebra for Kripke polynomial coalgebras. In *LICS*, 2009. To appear.

<u>Weighted automata</u> – $\mathbb{S} \times (\mathbb{S}^{Id})^A$

$\varepsilon ::= \emptyset \mid \varepsilon \oplus \varepsilon \mid \mu x.\varepsilon \mid x \mid s \mid a(s \cdot \varepsilon)$ where $s \in \mathbb{S}$ and $a \in A$

$$(\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \equiv \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) \qquad \varepsilon_1 \oplus \varepsilon_2 \equiv \varepsilon_2 \oplus \varepsilon_1 \qquad \varepsilon \oplus \emptyset \equiv \varepsilon$$
$$a(s \cdot \varepsilon) \oplus a(s' \cdot \varepsilon) \equiv a((s + s') \cdot \varepsilon) \qquad s \oplus s' \equiv s + s' \qquad a(0 \cdot \varepsilon) \equiv \emptyset$$
$$\varepsilon[\mu x.\varepsilon/x] \equiv \mu x.\varepsilon \qquad \gamma[\varepsilon/x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon \qquad 0 \equiv \emptyset$$

---

<u>Stratified systems</u> – $D_\omega(Id) + (B \times Id) + 1$

$\varepsilon ::= \mu x.\varepsilon \mid x \mid \langle b, \varepsilon \rangle \mid \bigoplus_{i \in 1 \cdots n} p_i \cdot \varepsilon_i \mid \downarrow$ where $b \in B$, $p_i \in (0,1]$ and $\sum_{i \in 1 \dots n} p_i = 1$

$$(\varepsilon_1 \oplus \varepsilon_2) \oplus \varepsilon_3 \equiv \varepsilon_1 \oplus (\varepsilon_2 \oplus \varepsilon_3) \qquad \varepsilon_1 \oplus \varepsilon_2 \equiv \varepsilon_2 \oplus \varepsilon_1 \qquad (p_1 \cdot \varepsilon) \oplus (p_2 \cdot \varepsilon) \equiv (p_1 + p_2) \cdot \varepsilon$$
$$\varepsilon[\mu x.\varepsilon/x] \equiv \mu x.\varepsilon \qquad \gamma[\varepsilon/x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon$$

---

<u>Segala systems</u> – $\mathcal{P}_\omega(D_\omega(Id))^A$

$\varepsilon ::= \emptyset \mid \varepsilon \boxplus \varepsilon \mid \mu x.\varepsilon \mid x \mid a(\{\varepsilon'\})$ where $a \in A$, $p_i \in (0,1]$ and $\sum_{i \in 1 \dots n} p_i = 1$
$\varepsilon' ::= \bigoplus_{i \in 1 \cdots n} p_i \cdot \varepsilon_i$

$$(\varepsilon_1 \boxplus \varepsilon_2) \boxplus \varepsilon_3 \equiv \varepsilon_1 \boxplus (\varepsilon_2 \boxplus \varepsilon_3) \qquad \varepsilon_1 \boxplus \varepsilon_2 \equiv \varepsilon_2 \boxplus \varepsilon_1 \qquad \varepsilon \boxplus \emptyset \equiv \varepsilon \qquad \varepsilon \boxplus \varepsilon \equiv \varepsilon$$
$$(\varepsilon'_1 \oplus \varepsilon'_2) \oplus \varepsilon'_3 \equiv \varepsilon'_1 \oplus (\varepsilon'_2 \oplus \varepsilon'_3) \qquad \varepsilon'_1 \oplus \varepsilon'_2 \equiv \varepsilon'_2 \oplus \varepsilon'_1 \qquad (p_1 \cdot \varepsilon) \oplus (p_2 \cdot \varepsilon) \equiv (p_1 + p_2) \cdot \varepsilon$$
$$\varepsilon[\mu x.\varepsilon/x] \equiv \mu x.\varepsilon \qquad \gamma[\varepsilon/x] \equiv \varepsilon \Rightarrow \mu x.\gamma \equiv \varepsilon$$

Table 1: All the expressions are closed and guarded. The congruence and the $\alpha$-equivalence axioms are implicitly assumed for all the systems. The symbols 0 and $+$ denote, in the case of weighted automata, the empty element and the binary operator of the commutative monoid $\mathbb{S}$ while, for the other systems, denote the ordinary 0 and sum of real numbers. With a slight abuse of notation, we write $\bigoplus_{i \in 1 \cdots n} p_i \cdot \varepsilon_i$ for $p_1 \cdot \varepsilon_1 \oplus \cdots \oplus p_n \cdot \varepsilon_n$.