Coalgebra at the CWI: a brief overview

Alexandra Silva

Centrum Wiskunde & Informatica, The Netherlands

CWI post-doc scientific meeting, February 2011
The team

Jan Rutten  Marcello Bonsangue  Alexandra Silva  Joost Winter

Alexandra Silva  (CWI)
What is Coalgebra?

- Mathematical framework to reason about several dynamical systems and models of computation

- det. automata
- trees
- linear systems/stream circuits
How do we do it?

In order to study all these systems uniformly we think of them as

$$(S, g: S \rightarrow G(S))$$

Example:
How do we do it?

- In order to study all these systems uniformly we think of them as
  \[(S, g: S \to G(S))\]

**Example:**

\[(S, g: S \to 2 \times S^A)\]
How do we do it?

In order to study all these systems uniformly we think of them as

\[(S, g: S \rightarrow G(S))\]

Example:

\[(S, g: S \rightarrow 2 \times S^A)\]
How do we do it?

- In order to study all these systems uniformly we think of them as
  \[(S, g: S \rightarrow G(S))\]

Example:

- **Universal coalgebra**: the type \(G\) is rich enough to determine a notion of behaviour and equivalence

\[(S, g: S \rightarrow 2 \times S^A)\]
How do we do it?

- In order to study all these systems uniformly we think of them as

\[(S, g: S \rightarrow G(S))\]

**Example:**

\[(S, g: S \rightarrow 2 \times S^A)\]

- **Universal coalgebra:** the type \(G\) is rich enough to determine a notion of behaviour and equivalence

**Example:** Languages \((A^*)\) and language equivalence
Some of our recent achievements

The results above can be extended uniformly to a larger class of systems ($G$-coalgebras), including: labelled transition systems, infinite trees, Mealy machines, probabilistic automata, weighted automata, etc.
Some of our recent achievements

The results above can be extended uniformly to a larger class of systems ($G$-coalgebras), including: labelled transition systems, infinite trees, Mealy machines, probabilistic automata, weighted automata, etc.
What does this yield concretely?

- Languages, axiomatizations and algorithms to reason about equivalence for a large class of models
- From the general framework we recover known results (e.g. for LTS Milner’s language and axiomatization), but also...
- ...new results, impact in the concurrency community (cf. our CONCUR paper–Bonchi, Bonsangue, Rutten & Silva 2009)
What does this yield concretely?

- Languages, axiomatizations and algorithms to reason about equivalence for a large class of models
- From the general framework we recover known results (e.g. for LTS Milner’s language and axiomatization), but also...
- ...new results, impact in the concurrency community (cf. our CONCUR paper–Bonchi, Bonsangue, Rutten & Silva 2009)
What does this yield concretely?

- Languages, axiomatizations and algorithms to reason about equivalence for a large class of models
- From the general framework we recover known results (e.g. for LTS Milner’s language and axiomatization), but also...
- ...new results, impact in the concurrency community (cf. our CONCUR paper–Bonchi, Bonsangue, Rutten & Silva 2009)
What we are looking at now

Context Free languages and grammars
- Is it possible to generalize CFL to other models?

Automation
- Coinduction is very suitable for automation
- Can we provide automatic reasoning on equivalence of models?
- Applications in program schematology, compiler certification, etc
What we are looking at now

Context Free languages and grammars
- Is it possible to generalize CFL to other models?

Automation
- Coinduction is very suitable for automation
- Can we provide automatic reasoning on equivalence of models?
- Applications in program schematology, compiler certification, etc
What we are looking at now

Algebraic proof long and requires ingenuity
Coinductive proof fully automatic
Thank you for your attention!