From Algebras and Coalgebras to Dialgebras

Erik Poll

Joint work with Jan Zwanenburg

University of Nijmegen

CMCS'2001

Algebra vs Coalgebra

Interesting **duality** between algebra and coalgebra:

constructor : $IN(X) \to X$	$\longleftrightarrow$	destructor : $X \to OUT(X)$
construction	$\longleftrightarrow$	observation
initial algebra	$\longleftrightarrow$	final coalgebra
induction, recursion	$\longleftrightarrow$	coinduction, corecursion
quotient-, sub-algebra	$\longleftrightarrow$	sub-, quotient-coalgebra
	$\vdots$	
WADT	$\longleftrightarrow$	CMCS

This focuses on **differences** between algebra and coalgebra.

What do algebra and coalgebra have in **common** *?*

Algebras and Coalgebras

An $\left\{ \begin{array}{l} \text{algebra} \\ \text{coalgebra} \end{array} \right.$ is a pair $(X, \vec{f})$ consisting of

a set X with functions $\left\{ \begin{array}{lrcl} f_i : & F_i(X) & \to & X \\ f_i : & X & \to & F_i(X) \end{array} \right.$

with the F_i (polynomial) functors, giving the $\left\{ \begin{array}{l} \text{input} \\ \text{output} \end{array} \right.$ types

There's an obvious way to "unify" these definitions ...

Dialgebras [Hagino87]

A **dialgebra** $(X, \vec{f})$ is a pair $(X, \vec{f})$ consisting of a set X with functions

$$f_i : IN_i(X) \to OUT_i(X)$$

with IN_i and OUT_i polynomial functors, giving input and output types.

The **signature** of the dialgebra (X, h) is

$$\Sigma(X) = (IN_1(X) \to OUT_1(X)) \times \ldots \times (IN_n(X) \to OUT_n(X))$$

NB such a dialgebraic signature is not a functor!

NB polynomial functors without constant exponents $C \to F(X)$!

Running example: sets of natural numbers

Signature Σ_{SET} defined as

$$\begin{array}{rlcll}
\Sigma_{SET}(X) = & & X & & \texttt{empty} \\
\times & (X \times X & \rightarrow & X) & \texttt{union} \\
\times & (\mathbb{N} \times X & \rightarrow & X) & \texttt{add} \\
\times & (X \times \mathbb{N} & \rightarrow & Bool) & \texttt{contains?} \\
\times & (X & \rightarrow & 1 + \mathbb{N} \times X) & \texttt{removeMin} \\
\times & (X & \rightarrow & X \times X) & \texttt{split} \\
\times & (X \times X & \rightarrow & X \times X) & \texttt{symdiff}
\end{array}$$

E.g. $\mathcal{SET} = (\mathcal{P}(\mathbb{N}), (\emptyset, \cup, \ldots))$ and $\mathcal{LIST} = (List_{\mathbb{N}}, (nil, +\!\!+, \ldots))$ are Σ_{SET}-dialgebras.

Removing some operations yields an algebra or coalgebra.

More examples of dialgebras:

- Take a coalgebra, and add an initial state $init : 1 \rightarrow X$.

  NB an initial state makes sense for many coalgebras!

- Classes with **constructors**, e.g. $new : C \rightarrow X$, and with **binary methods**, e.g. $op : X \times X \rightarrow X$.

  Viewing coalgebras as OO classes does not allow constuctors or binary methods.

Obvious question: which (co)algebraic notions and properties can be generalised to dialgebra, and which cannot ?

Homomorphism between Σ-dialgebras

A **homomorphism** h from (A, f) to (B, g) is a function $h : A \rightarrow B$ that "preserves" the operations, i.e.

$$\begin{tikzcd}
IN_i(A) \arrow[r, "f_i"] \arrow[d, "IN_i(h)"'] & OUT_i(A) \arrow[d, "OUT_i(h)"] \\
IN_i(B) \arrow[r, "g_i"] & OUT_i(B)
\end{tikzcd}$$

for all i.

The function $\mathcal{A} : List_{\mathbb{N}} \rightarrow \mathcal{P}(\mathbb{N})$ which maps $[l_1, \ldots, l_n]$ to $\{l_1, \ldots, l_n\}$ is a homomorphism from the dialgebra $\mathcal{LIST}$ to $\mathcal{SET}$.

Predicate and relation lifting

For *any* signature Σ

$$\Sigma(X) ::= X \mid C \mid \Sigma_1 + \Sigma_2 \mid \Sigma_1 \times \Sigma_2 \mid \Sigma_1 \rightarrow \Sigma_2$$

– *not just dialgebraic ones!* – we can define **predicate** and **relation lifting**:

- $\boldsymbol{\Sigma^{pred}(P)} \subseteq \boldsymbol{\Sigma(X)}$ for any **predicate** $P \subseteq X$
- $\boldsymbol{\Sigma^{rel}(\sim)} \subseteq \boldsymbol{\Sigma(X)} \times \boldsymbol{\Sigma(Y)}$ for any **relation** $\sim \subseteq X \times Y$

Invariants and bisimulations for dialgebras

An **invariant** for (A, f) is a predicate P on A such that $\boldsymbol{f \in \Sigma^{pred}(P)}$, *ie. the operations "preserve"* P.

A **bisimulation** between (A, f) and (B, g) is a relation $\sim\ \subseteq A \times B$ such that $\boldsymbol{(f, g) \in \Sigma^{rel}(\sim)}$, *ie. the operations "respect"* $\sim$.

A **(partial) congruence** on (A, f) is a *(partial) equivalence relation* that is a *bisimulation*.

An *invariant* P on (A, f) yields a **sub-dialgebra** (P, f).

A *(partial) congruence* $\sim$ on (A, f) yields a **quotient-dialgebra** $(A, f)/\sim$

These are all generalisations of the existing (co)algebraic notions.

Examples

Recall $\mathcal{SET} = (\mathcal{P}(\mathbb{N}), (\emptyset, \cup, \ldots))$ and $\mathcal{LIST} = (List_{\mathbb{N}}, (nil, +\!\!+, \ldots))$

Examples of **invariants** for $\mathcal{LIST}$:

- being sorted, not containing duplicates

Examples of **bisimulations**:

- $graph(\mathcal{A})$ is a bisimulation between $\mathcal{LIST}$ and $\mathcal{SET}$
- $\sim_{PERM}$, which relates lists that are permutations, is a bisimulation between $\mathcal{LIST}$ and itself, and a congruence

Properties of invariants and bisimulations

Closure of invariants and bisimulations under $\cap$ and $\cup$:

	inv & bisim	
	$\cap$	$\cup$
algebras	yes	no
dialgebras	yes	no
coalgebras	yes	yes[1]

[1]For invariants this yields the weakest invariant Φ_{inv} such that $\Phi_{inv} \Rightarrow \Phi$, for any Φ.

[1]For bisims this yields the greatest bisimulation between two dialgebras.

Counterexample: union of bisims not a bisim

Consider the Σ-(di)algebras

$$(List_{Nat}, (+\!\!+, length)) \qquad\qquad (\mathbb{Z}, (+, abs))$$

with $\Sigma(X) = (X \times X \to X) \times (X \to \mathbb{N})$.

Then $\sim_1$ and $\sim_2$ are bisims between these dialgebras

$$l \sim_1 length(l) \qquad\qquad l \sim_2 -length(l)$$

Then $\sim_1 \cup \sim_2$ is not a bisim: e.g. $[k, l] \sim_1 2$ and $[m, n] \sim_2 -2$ but $[k, l] +\!\!+ [m, n] \not\sim_i 0$ since only $[k, l, m, n] \sim_i \pm 4$

The problem with $\cup$: binary methods

Dialgebras **without binary methods** are dialgebras with *at most one occurrence of X in all $IN_i(X)$*. Eg. no union : $X \times X \to X$

These are better-behaved:

	inv & bisim	
	$\cap$	$\cup$
algebras	yes	**no**
dialgebras	yes	**no**
dialgebras without binary	yes	**yes**
coalgebras	yes	**yes**

Properties of congruences

For general dialgebras, bisimulations are not closed under union :-(

This means there is no canonical notion of equivalence between elements of different dialgebras – given by the largest bisimulation :-(

However, all is not lost:

> **Congruences are closed under "union"**, ie.
> if $\sim_1$ and $\sim_2$ are congruences then so is $(\sim_1 \cup \sim_2)^*$.

Here we use the restriction to polynomial functors <u>without</u> constant exponents $C \to F(X)$.

Properties of invariants, bisimulations and congruences

Closure of invariants and bisimulations and congruences under $\cap$ and $\cup$:

	inv & bisim		**congr.**
	$\cap$	$\cup$	$\cup$
algebras	yes	no	**yes**
dialgebras	yes	no	**yes**[2]
coalgebras	yes	yes[1]	**yes**

[1]Closure of *bisim's* under $\cup$ yields a unique *greatest bisimulation <u>between two dialgebras</u>* .

[2]Closure of *congruences* under $\cup$ yields a unique *greatest congruence <u>on a single dialgebra</u>*.

Observational equivalence

Beware of different notions of "observational equivalence", namely

1. between individual elements of a dialgebra, if they are related by the greatest congruence. *E.g.* $[1,2,3] \sim_{perm} [3,2,1]$, *both from $\mathcal{LIST}$*

2. between individual elements of distinct dialgebras if they are related by some bisimulation. *E.g.* $[1,2,3] \sim \{1,2,3\}$, *from $\mathcal{LIST}$ and $\mathcal{SET}$.*

3. between whole dialgebras, if there is a (total) bisimulation between them. *Eg.* $\mathcal{LIST} = (List, (nil, +\!\!+, \ldots)) \simeq_{obs} \mathcal{SET} = (\mathcal{P}(\mathbb{N}), (\emptyset, \cup, \ldots))$

2. only makes sense for coalgebras and dialgebras without binary methods.

Dialgebraic specification: $SPEC_{=}$ vs $SPEC_{\sim}$

Possible **dialgebraic specifications** for Σ_{SET}:

- $SPEC_{=}$

  $\texttt{union}(x, y) = \texttt{union}(y, x) \ \wedge \ \ldots$

- $SPEC_{\sim}$

  $\texttt{union}(x, y) \sim \texttt{union}(y, x) \wedge \ldots$ for some (partial) congruence $\sim$

$SPEC_{=}$ looks nicer than $SPEC_{\sim}$, but $SPEC_{\sim}$ is weaker ...

Fortunately, models of $SPEC_{=}$ and $SPEC_{\sim}$ are observationally equivalent!

Dialgebraic specification: behavioural satisfaction

Models of $SPEC_{\sim}$ **behaviourally satisfy** $SPEC_{=}$:

Theorem

$$\mathcal{A} \models SPEC_{\sim} \iff \mathcal{A}/{\sim} \models SPEC_{=}$$
$$\iff \exists \mathcal{A}'.\ \mathcal{A}' \models SPEC_{=} \ \wedge \ \mathcal{A}' \simeq_{obs} \mathcal{A}$$

(For algebras [BidoitHennickerWirsing'95][HofmannSannella'96])

The following proof rule is sound

$$\frac{\mathcal{A} \models SPEC_{\sim} \qquad \models SPEC_{=} \Rightarrow \Phi}{\mathcal{A} \models \Phi}$$

Moral: choose $SPEC_{\sim}$ or $SPEC_{=}$, depending on which is easiest

Related Work

Recent work on extending coalgebra / combining algebra with coalgebra:

- **algebra-coalgebra pairs** aka **bialgebras** [KurzHenniger'99] [Cirstea'00]

  Pairs(f, g) with f an F-algebra and g a G-coalgebra.

  Clearly, any bialgebra is a dialgebra.

- **coalgebras with binary methods** [Tews'00]

  Coalgebras for higher-order polynomial functors.

  This suggests further generalisation of dialgebra.

Related Work: abstract data types in type theory

Dialgebras – and hence both algebras and coalgebras – are special cases of **abstract data types (ADT's)**.

ADT's described in type theory using existential types [MitchellPlotkin88]. Associated logic theory using parametricity [PlotkinAbadi93] can be used for behavioural specification of ADT's [(Poll)Zwanenburg'99][Hannay'99]

⇒ notion of "**observability**" make sense for coalgebras *and* algebras.

⇒ notion of "**behavioural satisfaction**" make sense for algebraic *and* coalgebraic specifications.

The type-theoretic setting allows much wilder signatures than dialgebraic ones.

Conclusions

- **Dialgebras provide a smooth generalisation of (co)algebras**
- Dialgebras without binary methods are better behaved than arbitrary dialgebras.

Extra properties of (co)algebras over general dialgebras

- For **(co)algebras** we have **initial/final** models.
  But for specifications we usually want loose models, so absence of initial/final models not that important ?
- For **coalgebras** (and dialgebras without binary methods):
  1. invariants/bisims closed under union
  2. observational equivalence of individual 'objects', from different coalgebras
  3. subtyping - substitutability of individual 'objects'

2,3 illustrate that : $\dfrac{\text{coalgebra}}{\text{dialgebra}} = \dfrac{\text{OO}}{\text{ADT}}$

Future Work

Notion of dialgebra is fairly ad-hoc and very syntactical.

- Less ad-hoc and more semantical notion of dialgebra ?

  (dyad, nested sketches, . . . ?)
- Relaxing restrictions on (polynomial functors in) dialgebraic signatures?