Isomorphism of Finitary Inductive Types

Christian Sattler
joint work (in progress) with Nicolai Kraus

University of Nottingham

May 2014

Motivating Example

Generic data may be represented in a multitude of ways.

Motivating Example

Generic data may be represented in a multitude of ways.

Consider generic binary trees with generic data at nodes and leafs:

```
data Tree (X Y : Set) : Set where
  leaf : X → Tree X Y
  node : Y → Tree X Y → Tree X Y → Tree X Y
```

Motivating Example

Unravelling the left-most branch yields an alternative presentation as *spine trees*:

```
data Spine (X Y : Set) : Set where
  nil  : Spine X Y
  cons : Y → Tree X Y → Spine X Y → Spine X Y

data SpineTree (X Y : Set) : Set where
  spine : X → Spine X Y → SpineTree X Y
```

Motivating Example

Unravelling the left-most branch yields an alternative presentation as *spine trees*:

$$
\begin{array}{l}
\mathsf{data}\ \mathsf{Spine}\ (X\ Y : \mathsf{Set}) : \mathsf{Set}\ \mathsf{where} \\
\quad \mathsf{nil}\ \ \ : \mathsf{Spine}\ X\ Y \\
\quad \mathsf{cons} : Y \to \mathsf{Tree}\ X\ Y \to \mathsf{Spine}\ X\ Y \to \mathsf{Spine}\ X\ Y
\end{array}
$$

$$
\begin{array}{l}
\mathsf{data}\ \mathsf{SpineTree}\ (X\ Y : \mathsf{Set}) : \mathsf{Set}\ \mathsf{where} \\
\quad \mathsf{spine} : X \to \mathsf{Spine}\ X\ Y \to \mathsf{SpineTree}\ X\ Y
\end{array}
$$

Altenkirch et al. (2005): can we decide whether two such definitions (i.e., parametric finitary inductive types) are generically isomorphic?

Regular Functors

A modular description of parametric finitary inductive types is given by *regular functors*. They are composed of (codes for)

- finite products,
- finite sums,
- (parametric) initial algebra formation.

Regular Functors

A modular description of parametric finitary inductive types is given by *regular functors*. They are composed of (codes for)

- finite products,
- finite sums,
- (parametric) initial algebra formation.

Recall that mutual inductive definitions may be transformed into non-mutual "nested" definitions, e.g.

$$\text{SpineTree} = \mu A.\, X \times (\mu B.\, 1 + Y \times A \times B)$$

Regular Functors

A modular description of parametric finitary inductive types is given by *regular functors*. They are composed of (codes for)

- finite products,
- finite sums,
- (parametric) initial algebra formation.

Recall that mutual inductive definitions may be transformed into non-mutual "nested" definitions, e.g.

$$\text{SpineTree} = \mu A.\, X \times (\mu B.\, 1 + Y \times A \times B)$$

Thus we ask: is isomorphism of regular functors decidable?

Isomorphism: Clarification Needed

But what does *isomorphic* mean?

Isomorphism: Clarification Needed

But what does *isomorphic* mean?

Most general sensible notion of model for parametric finitary inductive types: bicartesian-closed categories with initial algebras for regular functors.

Isomorphism: Clarification Needed

But what does *isomorphic* mean?

Most general sensible notion of model for parametric finitary inductive types: bicartesian-closed categories with initial algebras for regular functors.

Possible choices:

- Isomorphism in the standard set model, type theory, or any other locally cartesian-closed category with sufficient colimits.
- Syntactic isomorphism (isomorphism in all models): closed λ-terms $f : A \to B$ and $g : B \to A$ such that $g \circ f =_{\beta\eta} \mathrm{id}_A$ and $f \circ g =_{\beta\eta} \mathrm{id}_B$ with conversion rules for extensionality of sums and uniqueness of recursors.
- ...

The Set Model

Altenkirch et al. (2005) observe that isomorphism in the set model boils down to equivalence of context-free grammars with

- commuting letters (instead of derived words one considers multisets of letters),
- multiplicity of derivation (taking into account the number of possible derivations of a given multiset).

Here, letters correspond to type parameters.

The Set Model

Altenkirch et al. (2005) observe that isomorphism in the set model boils down to equivalence of context-free grammars with

- commuting letters (instead of derived words one considers multisets of letters),
- multiplicity of derivation (taking into account the number of possible derivations of a given multiset).

Here, letters correspond to type parameters.

(The terminology of *regular* functors is slightly misleading in this context as they more closely resemble *context-free* grammars.)

The Set Model: Power Series

Lists:

$$\mathsf{List}(X) = 1 + X + X^2 + \ldots \in \mathbb{N}[\![X]\!]$$

The Set Model: Power Series

Lists:

$$\mathrm{List}(X) = 1 + X + X^2 + \ldots \in \mathbb{N}[\![X]\!]$$

Fibonacci sequence:

$$\begin{aligned} & A = 1 + (X + X^2) \times A \\ \Longrightarrow\ & A = 1 + X + 2X^2 + 3X^3 + 5X^4 + 8X^5 + \ldots \in \mathbb{N}[\![X]\!] \end{aligned}$$

The Set Model: Power Series

Lists:

$$\mathsf{List}(X) = 1 + X + X^2 + \ldots \in \mathbb{N}[\![X]\!]$$

Fibonacci sequence:

$$\begin{aligned} & A = 1 + (X + X^2) \times A \\ \implies & A = 1 + X + 2X^2 + 3X^3 + 5X^4 + 8X^5 + \ldots \in \mathbb{N}[\![X]\!] \end{aligned}$$

Binary trees:

$$\begin{aligned} & \mathsf{Tree} = X + Y \times \mathsf{Tree} \times \mathsf{Tree} \\ \implies & \mathsf{Tree} = \frac{1 - \sqrt{1 - 4XY}}{2Y} \\ \implies & \mathsf{Tree} = X + X^2Y + 2X^3Y^2 + 5X^4Y^3 + 15X^5Y^4 + \ldots \in \mathbb{N}[\![X, Y]\!] \end{aligned}$$

The Set Model: Guardedness

Call a regular functor *guarded* if it has a representation as a power series with finite coefficients.

The Set Model: Guardedness

Call a regular functor *guarded* if it has a representation as a power series with finite coefficients.

Via syntactic analysis, any regular functor F can be decomposed into the sum $F = G + H$ of a guarded regular functor G and a "purely unguarded" fuctor H fulfilling $\mathbf{N} \times H = H$.

The Set Model: Guardedness

Call a regular functor *guarded* if it has a representation as a power series with finite coefficients.

Via syntactic analysis, any regular functor F can be decomposed into the sum $F = G + H$ of a guarded regular functor G and a "purely unguarded" fuctor H fulfilling $\mathbf{N} \times H = H$.

The primordial purely unguarded type is

$$\mathbf{N} = \mu X.\, 1 + X$$

The Set Model: Guardedness

Call a regular functor *guarded* if it has a representation as a power series with finite coefficients.

Via syntactic analysis, any regular functor F can be decomposed into the sum $F = G + H$ of a guarded regular functor G and a "purely unguarded" fuctor H fulfilling $\mathbf{N} \times H = H$.

The primordial purely unguarded type is

$$\mathbf{N} = \mu X.\, 1 + X$$

Warning: only the purely unguarded part of the decomposition is uniquely determined.

The Set Model: Guardedness

Consider regular functors with named parameters $X_1, \ldots, X_n$.

The Set Model: Guardedness

Consider regular functors with named parameters $X_1, \ldots, X_n$.

Closing guarded regular functors under negation and inversion, we obtain a subfield

$$K \subseteq \mathbb{Q}(\!(X_1, \ldots, X_n)\!)$$

of formal Laurent series.

The Set Model: Guardedness

Consider regular functors with named parameters $X_1, \ldots, X_n$.

Closing guarded regular functors under negation and inversion, we obtain a subfield

$$K \subseteq \mathbb{Q}(\!(X_1, \ldots, X_n)\!)$$

of formal Laurent series. By structural induction, we see that the field extension

$$\mathbb{Q}(X_1, \ldots, X_n) \subseteq K$$

is algebraic.

The Set Model: Guardedness

Consider regular functors with named parameters $X_1, \dots, X_n$.

Closing guarded regular functors under negation and inversion, we obtain a subfield

$$K \subseteq \mathbb{Q}(\!(X_1, \dots, X_n)\!)$$

of formal Laurent series. By structural induction, we see that the field extension

$$\mathbb{Q}(X_1, \dots, X_n) \subseteq K$$

is algebraic.

The minimal polynomial of a guarded regular functor, together with a bounded prefix of its list of coefficients, yields a *finitary description* of its semantics.

The Set Model

Decidability of isomorphism of purely unguarded regular functors is dealt with by Parikh's theorem (1961).

The Set Model

Decidability of isomorphism of purely unguarded regular functors is dealt with by Parikh's theorem (1961).

However, we not yet know how to combine these results. A complication is provided by the fact that the diagonal of an algebraic power series need not be algebraic.

The Set Model

Decidability of isomorphism of purely unguarded regular functors is dealt with by Parikh's theorem (1961).

However, we not yet know how to combine these results. A complication is provided by the fact that the diagonal of an algebraic power series need not be algebraic.

(However, we note that the interesting examples of generic datatypes tend to be guarded.)

The Term Model: An Outline

Main part of our contribution: internalization of the guardedness argument in the equational theory of the initial model.

The Term Model: An Outline

Main part of our contribution: internalization of the guardedness argument in the equational theory of the initial model.

Warning: initial algebras in the term model **cannot** be constructed as colimits of chains. In particular, we do not have any way to properly induct over natural numbers. Uniqueness of the recursor provides for a very weak substitute of induction over identities.

The Term Model: An Outline

Main part of our contribution: internalization of the guardedness argument in the equational theory of the initial model.

Warning: initial algebras in the term model **cannot** be constructed as colimits of chains. In particular, we do not have any way to properly induct over natural numbers. Uniqueness of the recursor provides for a very weak substitute of induction over identities.

This provides the capital complication.

The Term Model: An Outline

The main ingredients are:

- utilizing internal booleans for internal propositional logic,
- an internal version of induction over internal propositions,
- a version of traversability generalized to multiple argument functors, together with a derivation of traversability of regular functors in arbitrary bicccs — the abstract concept of traversability keeps the development from getting overly syntactic,
- internal (generalized structural) equality predicates defined using traversability,
- internal representations of algebraic structures such as rings and fields, polynomials and power series,

The Term Model: An Outline

(cont.)

- internal enumerative listings of degree–sorted values of guarded regular types to serve as lookup tables for indexing functions, with the listings again utilizing traversals,
- polymorphic injection and extraction of data of regular functors akin to the concept of shapely functors, but in a weaker setting (our category is not *extensive*).

Conclusion

Core results:

- Syntactic isomorphism of guarded regular functors is decidable.
- The set model is complete for isomorphism of guarded regular functors

Further questions:

- What about mixed guarded–unguarded types?
- Do we have a similar result for coinductive types?