A canonical representation for free left-dioids

M. Jaskelioff, E. Postan, **E. Rivas**

Universidad Nacional de Rosario

CIFASIS - CONICET

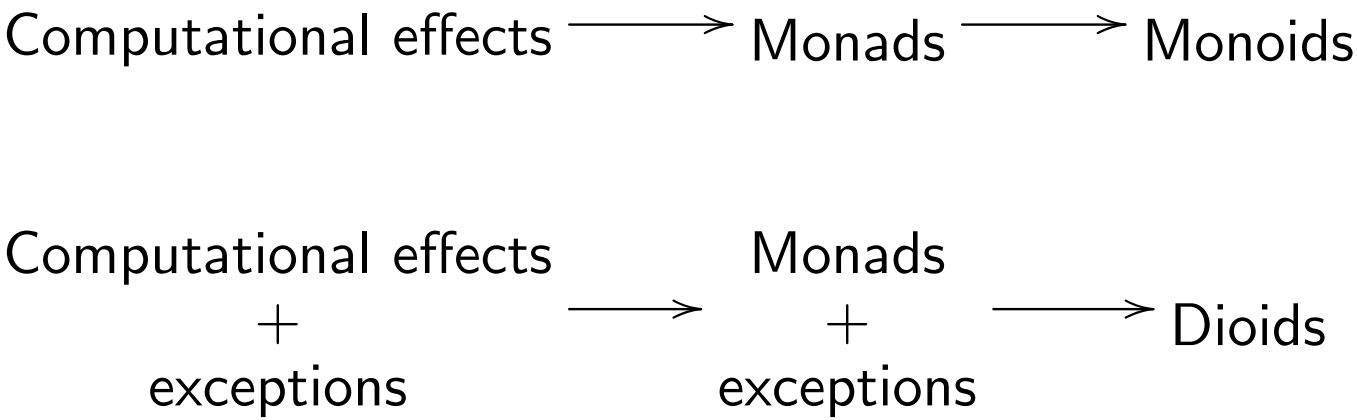
XIV Congreso Dr. Antonio Monteiro
2017, Bahía Blanca.

Motivation

Computational effects $\longrightarrow$ Monads $\longrightarrow$ Monoids

Free structures give syntactic programs modulo expected laws of behavior: we want implementations.

Motivation



Free structures give syntactic programs modulo expected laws of behavior: we want implementations.

Left-dioids

Left-dioid

A *(left-)dioid* is a set tuple $(D, \otimes, \oplus, 1_D, 0_D)$, where

- D is a set,
- $\otimes, \oplus : D \times D \to D$ are binary operations,
- $1_D, 0_D \in D$ are constants,

such that the following is satisfied:

- $(D, \otimes, 1_D)$ is a monoid,
- $(D, \oplus, 0_D)$ is a monoid,
- $0_D \otimes x = 0_D$ for all $x \in D$,
- $1_D \oplus x = 1_D$ for all $x \in D$.

Examples of dioids

Bounded lattices

Every bounded lattice $\langle L, \vee, \wedge, 0, 1 \rangle$ is a dioid.

Examples of dioids

Bounded lattices

Every bounded lattice $\langle L, \vee, \wedge, 0, 1 \rangle$ is a dioid.

Binary operations

Let S be any set. Then $\langle S \times S \to S, \diamond, *, \pi_1, \pi_2 \rangle$ is a dioid, where $(f \diamond g)(x, y) = f(g(x, y), y)$ and $(f * g)(x, y) = f(x, g(x, y))$.

Examples of dioids

Bounded lattices

Every bounded lattice $\langle L, \vee, \wedge, 0, 1 \rangle$ is a dioid.

Binary operations

Let S be any set. Then $\langle S \times S \to S, \diamond, *, \pi_1, \pi_2 \rangle$ is a dioid, where $(f \diamond g)(x, y) = f(g(x, y), y)$ and $(f * g)(x, y) = f(x, g(x, y))$.

Real unit interval

The real interval $[0, 1]$ has a dioid structure $\langle [0, 1], *, \cdot, 0, 1 \rangle$ where $x \cdot y$ is the usual multiplication and $x * y = x + y - xy$.

Free objects

Considering the forgetful functor $U : \mathrm{Did} \to \mathrm{Set}$, the free construction is given by the left adjoint F to U.

Free objects

Considering the forgetful functor $U : \mathrm{Did} \to \mathrm{Set}$, the free construction is given by the left adjoint F to U.

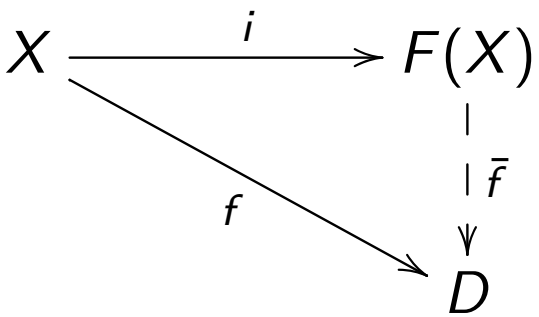
Explicitly, the free dioid over X is a dioid $F(X)$ and a function $i : X \to F(X)$ which are universal.

Free objects

Considering the forgetful functor $U : \mathrm{Did} \to \mathrm{Set}$, the free construction is given by the left adjoint F to U.

Explicitly, the free dioid over X is a dioid $F(X)$ and a function $i : X \to F(X)$ which are universal.

Universality: for any dioid D and function $f : X \to D$, there exists a unique dioid homomorphism $\bar{f} : F(X) \to D$ which make



commute.

Constructing the free algebras

Mathematically, we can construct F as:

$$F(X) = T(X)/\theta_0$$

Note that elements of $F(X)$ are actually sets.

Constructing the free algebras

Mathematically, we can construct F as:

$$F(X) = T(X)/\theta_0$$

Note that elements of $F(X)$ are actually sets.

But it cannot be implemented in most programming languages! They lack of quotients!

Constructing the free algebras

Mathematically, we can construct F as:

$$F(X) = T(X)/\theta_0$$

Note that elements of $F(X)$ are actually sets.

But it cannot be implemented in most programming languages! They lack of quotients!

Alternative plan

We try to find canonical elements for each set, in an uniform manner, obtaining a concrete representation.

An illustrative example: monoids

A concrete representation the free monoid over X is given by the least solution to the following recursive equation of sets:

$$F(X) \cong \{*\} \sqcup X \times F(X)$$

An illustrative example: monoids

A concrete representation the free monoid over X is given by the least solution to the following recursive equation of sets:

$$F(X) \cong \{*\} \sqcup X \times F(X)$$

A uniform choice: we choose the fully-right-associated term with the unit at the end.

An illustrative example: monoids

A concrete representation the free monoid over X is given by the least solution to the following recursive equation of sets:

$$F(X) \cong \{*\} \sqcup X \times F(X)$$

A uniform choice: we choose the fully-right-associated term with the unit at the end.

$$\{(xy)z, (xe)(yz), e(x(y(ze))), x(yz), \ldots\} \mapsto x(y(ze))$$

This construction can be generalised to obtain free monoids in monoidal categories.

A concrete construction for dioids

It is not trivial to find normal forms for dioid expressions.

A concrete construction for dioids

It is not trivial to find normal forms for dioid expressions.

Ezequiel found a representation of the free dioid over X, it is the least solution to the following recursive equations of sets:

$$F(X) \cong \{*\} \sqcup \{*\} \sqcup T \tag{1}$$
$$T \cong X \sqcup (S \times \{*\}) \sqcup (S \times T) \sqcup (M \times \{*\}) \sqcup (M \times T) \tag{2}$$
$$S \cong X \sqcup (M \times \{*\}) \sqcup (M \times T) \tag{3}$$
$$M \cong X \sqcup (S \times \{*\}) \sqcup (S \times T) \tag{4}$$

A concrete construction for dioids

It is not trivial to find normal forms for dioid expressions.

Ezequiel found a representation of the free dioid over X, it is the least solution to the following recursive equations of sets:

$$F(X) \cong \{*\} \sqcup \{*\} \sqcup T \tag{1}$$
$$T \cong X \sqcup (S \times \{*\}) \sqcup (S \times T) \sqcup (M \times \{*\}) \sqcup (M \times T) \tag{2}$$
$$S \cong X \sqcup (M \times \{*\}) \sqcup (M \times T) \tag{3}$$
$$M \cong X \sqcup (S \times \{*\}) \sqcup (S \times T) \tag{4}$$

Don't worry, we have a proof in the proof assistant Agda!

Conclusion and further work

Main conclusion: we have an explicit construction for free dioids which can be generalised to other categories.

We are now mainly interested in finding:

- Another representation for free dioids.
- A Cayley-like representation for dioids.

Conclusion and further work

Main conclusion: we have an explicit construction for free dioids which can be generalised to other categories.

We are now mainly interested in finding:

- Another representation for free dioids.
- A Cayley-like representation for dioids.

Thank you!