DSLM: Presenting Mathematical Analysis Using Functional Programming

Cezar Ionescu
cezar@chalmers.se

Patrik Jansson
patrikj@chalmers.se

Paper + talk: https://github.com/DSLsofMath/tfpie2015

Style example

∀ ε ∈ ℝ. (ε > 0) ⇒ ∃ a ∈ A. (|a − sup A| < ε)
Domain-Specific Languages of Mathematics [Ionescu and Jansson, 2015]: is a course currently developed at Chalmers in response to difficulties faced by third-year students in learning and applying classical mathematics (mainly real and complex analysis)
Main idea: encourage students to approach mathematical domains from a functional programming perspective (similar to Wells [1995]).

“... ideally, the course would improve the mathematical education of computer scientists and the computer science education of mathematicians.”
Introduction

- make functions and types explicit
- use types as carriers of semantic information, not just variable names
- introduce functions and types for implicit operations such as the power series interpretation of a sequence
- use a calculational style for proofs
- organize the types and functions in DSLs

Not working code, rather working understanding of concepts
We begin by defining the symbol $i$, called the **imaginary unit**, to have the property

$$i^2 = -1$$

Thus, we could also call $i$ the square root of $-1$ and denote it $\sqrt{-1}$. Of course, $i$ is not a real number; no real number has a negative square.

(Adams and Essex [2010], Appendix I)
We begin by defining the symbol $i$, called the **imaginary unit**, to have the property

$$i^2 = -1$$

Thus, we could also call $i$ the square root of $-1$ and denote it $\sqrt{-1}$. Of course, $i$ is not a real number; no real number has a negative square.

(Adams and Essex [2010], Appendix I)

```plaintext
data l = i
```
Definition: A complex number is an expression of the form

\[ a + bi \quad \text{or} \quad a + ib, \]

where \( a \) and \( b \) are real numbers, and \( i \) is the imaginary unit.
Complex numbers

**Definition:** A complex number is an expression of the form

\[ a + bi \quad \text{or} \quad a + ib, \]

where \( a \) and \( b \) are real numbers, and \( i \) is the imaginary unit.

```hs
data Complex = Plus1 R R I |
               Plus2 R I R

show :: Complex -> String
show (Plus1 x y i) = show x ++ " + " ++ show y ++ "i"
show (Plus2 x i y) = show x ++ " + " ++ "i" ++ show y
```
Definition: A complex number is an expression of the form

\[ a + bi \quad \text{or} \quad a + ib, \]

where \( a \) and \( b \) are real numbers, and \( i \) is the imaginary unit.

For example, \( 3 + 2i, \ \frac{7}{2} - \frac{2}{3}i, \ i\pi = 0 + i\pi, \) and \( -3 = -3 + 0i \) are all complex numbers. The last of these examples shows that every real number can be regarded as a complex number.
For example, $3 + 2i$, $\frac{7}{2} - \frac{2}{3}i$, $i\pi = 0 + i\pi$, and $-3 = -3 + 0i$ are all complex numbers. The last of these examples shows that every real number can be regarded as a complex number.

```
data Complex = Plus1 R R I |
              | Plus2 R I R
```

toComplex : $\mathbb{R} \rightarrow$ Complex
toComplex $x = Plus1 x 0 i$

- what about $i$ by itself?
- what about, say, $2i$?
(We will normally use $a + bi$ unless $b$ is a complicated expression, in which case we will write $a + ib$ instead. Either form is acceptable.)

\[
\text{data } \text{Complex} \ = \ Plus \ R \ R \ I
\]

\[
\text{data } \text{Complex} \ = \ Plusl \ R \ R
\]
It is often convenient to represent a complex number by a single letter; $w$ and $z$ are frequently used for this purpose. If $a$, $b$, $x$, and $y$ are real numbers, and $w = a + bi$ and $z = x + yi$, then we can refer to the complex numbers $w$ and $z$. Note that $w = z$ if and only if $a = x$ and $b = y$.

\texttt{newtype Complex = C (R, R)}
Definition: If $z = x + yi$ is a complex number (where $x$ and $y$ are real), we call $x$ the real part of $z$ and denote it $\text{Re} (z)$. We call $y$ the imaginary part of $z$ and denote it $\text{Im} (z)$:

\[
\begin{align*}
\text{Re} (z) &= \text{Re} (x + yi) = x \\
\text{Im} (z) &= \text{Im} (x + yi) = y
\end{align*}
\]

$\text{Re} : \text{Complex} \rightarrow \mathbb{R}$
$\text{Re} z @ (C (x, y)) = x$

$\text{Im} : \text{Complex} \rightarrow \mathbb{R}$
$\text{Im} z @ (C (x, y)) = y$
Shallow vs. deep embeddings

The sum and difference of complex numbers
If \( w = a + bi \) and \( z = x + yi \), where \( a, b, x, \) and \( y \) are real numbers, then

\[
\begin{align*}
w + z &= (a + x) + (b + y)i \\
w - z &= (a - x) + (b - y)i
\end{align*}
\]

Shallow embedding:

\[
(+) : \text{Complex} \rightarrow \text{Complex} \rightarrow \text{Complex} \\
(C(a, b)) + (C(x, y)) = C((a + x), (b + y)) \\
\text{newtype Complex} = C(\mathbb{R}, \mathbb{R})
\]
The sum and difference of complex numbers
If $w = a + bi$ and $z = x + yi$, where $a$, $b$, $x$, and $y$ are real numbers, then

$$w + z = (a + x) + (b + y)i$$
$$w - z = (a - x) + (b - y)i$$

Deep embedding (buggy):

$$(+) : \text{Complex} \rightarrow \text{Complex} \rightarrow \text{Complex}$$

$$(+) = \text{Plus}$$

**data** \text{ComplexDeep} = i

- $\text{ToComplex} \, \mathbb{R}$
- $\text{Plus} \, \text{Complex} \, \text{Complex}$
- $\text{Times} \, \text{Complex} \, \text{Complex}$
- ...

The sum and difference of complex numbers
If $w = a + bi$ and $z = x + yi$, where $a$, $b$, $x$, and $y$ are real numbers, then

\[
\begin{align*}
  w + z &= (a + x) + (b + y) \, i \\
  w - z &= (a - x) + (b - y) \, i
\end{align*}
\]

Deep embedding:

\[
(+) : \text{Complex} \rightarrow \text{Complex} \rightarrow \text{Complex} \\
(+) = \text{Plus} \\
\text{data Complex} = i \\
| \quad \text{ToComplex} \, \mathbb{R} \\
| \quad \text{Plus} \, \text{Complex Complex} \\
| \quad \text{Times} \, \text{Complex Complex} \\
| \quad \ldots
\]
Next: start from a more “mathematical” quote from the book:

The *completeness* property of the real number system is more subtle and difficult to understand. One way to state it is as follows: if $A$ is any set of real numbers having at least one number in it, and if there exists a real number $y$ with the property that $x \leq y$ for every $x \in A$ (such a number $y$ is called an *upper bound* for $A$), then there exists a smallest such number, called the *least upper bound* or *supremum* of $A$, and denoted $\text{sup}(A)$. Roughly speaking, this says that there can be no holes or gaps on the real line—every point corresponds to a real number.

(Adams and Essex [2010], page 4)
Min ("smallest such number")

Specification (not implementation)

\[
\begin{align*}
\text{min} & : \mathcal{P}(\mathbb{R}) \rightarrow \mathbb{R} \\
\text{min} \ A & = x \iff x \in A \land (\forall a \in A. \ x \leq a)
\end{align*}
\]

Example consequence (which will be used later):

*If \( y < \text{min} \ A \), then \( y \notin A \).*
Upper bounds

\[\text{ubs} : \mathcal{P} \mathbb{R} \rightarrow \mathcal{P} \mathbb{R}\]

\[\text{ubs } A = \{x \mid x \in \mathbb{R}, \text{x upper bound of } A\}\]
\[= \{x \mid x \in \mathbb{R}, \forall a \in A. \ a \leq x\}\]

The completeness axiom can be stated as

Assume an \(A : \mathcal{P}^+ \mathbb{R}\) with an upper bound \(u \in \text{ubs } A\).

Then \(s = \sup A = \min (\text{ubs } A)\) exists.

where

\[\sup : \mathcal{P}^+ \mathbb{R} \rightarrow \mathbb{R}\]

\[\sup = \min \circ \text{ubs}\]
Assume an $A: \mathcal{P}^+ \rightarrow \mathbb{R}$ with an upper bound $u \in \text{ubs } A$. Then $s = \sup A = \min (\text{ubs } A)$ exists.

But $s$ need not be in $A$ — could there be a “gap”? 
Completeness and “gaps”

Assume an $A : \mathcal{P}^+ \mathbb{R}$ with an upper bound $u \in \text{ubs } A$.

Then $s = \sup A = \min (\text{ubs } A)$ exists.

But $s$ need not be in $A$ — could there be a “gap”?

With “gap” = “an $\epsilon$-neighbourhood between $A$ and $s$” we can show there is no “gap”.
A proof: Completeness implications step-by-step

\[ \epsilon > 0 \]
\[ \Rightarrow \{ \text{arithmetic} \} \]
\[ s - \epsilon < s \]
A proof: Completeness implications step-by-step

\[ \epsilon > 0 \]
\[ \Rightarrow \{ \text{arithmetic} \} \]
\[ s - \epsilon < s \]
\[ \Rightarrow \{ s = \min(\text{ubs } A), \text{property of } \min \} \]
\[ s - \epsilon \notin \text{ubs } A \]
A proof: Completeness implications step-by-step

$\epsilon > 0$

$\Rightarrow \{ \text{arithmetic} \}$

$s - \epsilon < s$

$\Rightarrow \{ s = \min(\text{ubs} \ A), \text{property of } \min \}$

$s - \epsilon \notin \text{ubs} \ A$

$\Rightarrow \{ \text{set membership} \}$

$\neg \forall a \in A. \ a \leq s - \epsilon$
A proof: Completeness implications step-by-step

\[ \varepsilon > 0 \]
\[ \Rightarrow \text{ arithmetic } \]
\[ s - \varepsilon < s \]
\[ \Rightarrow \text{ property of } \min \]
\[ s - \varepsilon \notin \text{ubs } A \]
\[ \Rightarrow \text{ set membership } \]
\[ \neg \forall a \in A. \ a \leq s - \varepsilon \]
\[ \Rightarrow \text{ quantifier negation } \]
\[ \exists a \in A. \ s - \varepsilon < a \]
A proof: Completeness implications step-by-step

\[ \epsilon > 0 \]

\[ \Rightarrow \quad \{ \text{arithmetic} \} \]

\[ s - \epsilon < s \]

\[ \Rightarrow \quad \{ s = \min (\text{ubs } A), \text{property of } \min \} \]

\[ s - \epsilon \notin \text{ubs } A \]

\[ \Rightarrow \quad \{ \text{set membership} \} \]

\[ \neg \forall a \in A. \ a \leq s - \epsilon \]

\[ \Rightarrow \quad \{ \text{quantifier negation} \} \]

\[ \exists a \in A. \ s - \epsilon < a \]

\[ \Rightarrow \quad \{ \text{definition of upper bound} \} \]

\[ \exists a \in A. \ s - \epsilon < a \leq s \]
A proof: Completeness implications step-by-step

\[ \epsilon > 0 \]
\[ \Rightarrow \{ \text{arithmetic} \} \]
\[ s - \epsilon < s \]
\[ \Rightarrow \{ s = \min(\text{ubs } A), \text{property of } \min \} \]
\[ s - \epsilon \notin \text{ubs } A \]
\[ \Rightarrow \{ \text{set membership} \} \]
\[ \neg \forall a \in A. \ a \leq s - \epsilon \]
\[ \Rightarrow \{ \text{quantifier negation} \} \]
\[ \exists a \in A. \ s - \epsilon < a \]
\[ \Rightarrow \{ \text{definition of upper bound} \} \]
\[ \exists a \in A. \ s - \epsilon < a \leq s \]
\[ \Rightarrow \{ \text{absolute value} \} \]
\[ \exists a \in A. \ (|a - s| < \epsilon) \]
To sum up the proof says that the completeness axiom implies:

\[ \forall \epsilon \in \mathbb{R}. \ (\epsilon > 0) \implies \exists a \in A. \ (|a - \sup A| < \epsilon) \]
Completeness: proof interpretation ("no gaps")

To sum up the proof says that the completeness axiom implies:

\[ \forall \epsilon \in \mathbb{R}. \ (\epsilon > 0) \Rightarrow \exists a \in A. \ (|a - \sup A| < \epsilon) \]

More detail:
Assume a non-empty \( A : \mathcal{P} \mathbb{R} \) with an upper bound \( u \in \text{ubs } A \).
Then \( s = \sup A = \min (\text{ubs } A) \) exists.
We know that \( s \) need not be in \( A \) — could there be a "gap"?
Completeness: proof interpretation ("no gaps")

To sum up the proof says that the completeness axiom implies:

\[
\text{proof} : \forall \epsilon \in \mathbb{R}. \ (\epsilon > 0) \Rightarrow \exists a \in A. \ (|a - \sup A| < \epsilon)
\]

More detail:
Assume a non-empty \( A \) : \( \mathcal{P} \mathbb{R} \) with an upper bound \( u \in \text{ubs} \ A \).
Then \( s = \sup A = \min (\text{ubs} A) \) exists.
We know that \( s \) need not be in \( A \) — could there be a “gap”?
No, \( \text{proof} \) will give us an \( a \in A \) arbitrarily close to the supremum.
So, there is no “gap".
Conclusions

- make functions and types explicit: \( \text{Re} : \text{Complex} \rightarrow \mathbb{R} \), \( \text{min} : \mathcal{P}^+ \mathbb{R} \rightarrow \mathbb{R} \)
- use types as carriers of semantic information, not just variable names
- introduce functions and types for implicit operations such as \( \text{toComplex} : \mathbb{R} \rightarrow \text{Complex} \)
- use a calculational style for proofs
- organize the types and functions in DSLs (for \text{Complex}, limits, power series, etc.)
Future work

Partial implementation in Agda:

- errors caught by formalization (but no “royal road”)
  - ComplexDeep
  - choice function
- subsets and coercions
  - $\epsilon : \mathbb{R}_{>0}$, different type from $\mathbb{R}_{\geq 0}$ and $\mathbb{R}$ and $\mathbb{C}$
  - what is the type of $|\cdot|$? ($\mathbb{C} \to \mathbb{R}_{\geq 0}$?)
  - other subsets of $\mathbb{R}$ or $\mathbb{C}$ are common, but closure properties unclear
