Syntax for function expressions

& derivatives

Patrik Jansson

FunExp

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

```
der : F → F
der (Add f g) = Add (der f) (der g)
der (Mul f g) ≟ mulD (der f) (der g)
der X         = C 1
der (C c)     = C 0
```

$\exists$ mulD. H_2(der, Mul, mulD) $\wedge$?

"der implements D"

DSL → δσλ
DSLsofMath

Syntax for function expressions & derivatives

Patrik Jansson

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

$der : F \to F$

$$\begin{tikzcd} F \arrow[r, "der"] \arrow[d, "eval"'] \arrow[dr, dashed] & F \arrow[d, "eval"] \\ S \arrow[r, "D"] & S \end{tikzcd}$$

$D : S \to S$

$\forall f : F.\ \underline{D\,(eval\,f) == eval\,(der\,f)}$

where $S = \mathbb{R} \to \mathbb{R}$

$\exists\, muld.\ \underbrace{H_2(der, Mul, muld)}_{\text{"der implements D"}} \land\ ?$

$= H_1(eval, der, D)$

Syntax for function expressions

Patrik Jansson

& derivatives

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

$\exists mulD.\ H_2(der, Mul, mulD) \wedge$?

$eval \circ der = D \circ eval$

Assume mulD exists, look for $\bot$.

$der\ (Mul\ f\ g) = mulD\ (der\ f)\ (der\ g)$

Remember isPrime: we used $x = y = 2$ & $\bar{x} = 3$

such that $isP\ x = isP\ \bar{x}$ but $isP(x+y) \neq isP(\bar{x}+y)$

$$isP(x+y) = isP\ x \otimes isP\ y$$
$$isP(\bar{x}+y) = isP\ \bar{x} \otimes isP\ y$$

$$isP\ 4 \neq isP\ 5$$
$$F \neq T$$

DSL → δσλ
DSLsofMath

Syntax for function expressions

Patrik Jansson

& derivatives

```haskell
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

$\exists\, muld.\ H_2(der, Mul, muld) \wedge$
$\quad eval \circ der = D \circ eval$?

Assume $muld$ exists, look for $\bot$.

$der\,(Mul\ f\ g) = muld\ (der\ f)\,(der\ g)$

Can we find $f, \bar{f}, g$ s.t. $der\ f = der\ \bar{f}$ but
$der\,(Mul\ f\ g) \neq der\,(Mul\ \bar{f}\ g)$

Syntax for function expressions

Patrik Jansson

& derivatives

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

$\exists$ muld. $H_2(\text{der}, \text{Mul}, \text{muld}) \wedge$

eval ∘ der = D ∘ eval

Z

Assume muld exists, look for $\bot$.

der (Mul f g) = muld (der f) (der g)

Can we find $f, \bar{f}, g$ s.t. der f = der $\bar{f}$ but

der (Mul f g) $\neq$ der (Mul $\bar{f}$ g)

let f = Add X (C 0)

$\bar{f}$ = Add X (C 1)

g = X

der ("f*g") = der ("x^2") = "2*x"

der ("$\bar{f}$*g") = der ("x^2+x") = "2*x+1"

DSL → δσλ

DSLsofMath

Syntax for function expressions

Patrik Jansson

& derivatives

```haskell
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

$\exists\, muld.\ H_2(der, Mul, muld) \land$
$eval \circ der = D \circ eval$

?

Assume muld exists, look for $\bot$.

$der\ (Mul\ f\ g) = muld\ (der\ f)\ (der\ g)$

Can we find $f, \bar{f}, g$ s.t. $der\ f = der\ \bar{f}$ but
$der\ (Mul\ f\ g) \neq der\ (Mul\ \bar{f}\ g)$

let $f = Add\ X\ (C\ 0)$
$\bar{f} = Add\ X\ (C\ 1)$
$g = X$

$der\ f = der\ \bar{f} = Add\ (C\ 1)\ (C\ 0)$

DSL → δσλ DSLsofMath

Syntax for function expressions

Patrik Jansson

$f = \mathrm{Add}\ X\ (C\ 0)$

$\bar{f} = \mathrm{Add}\ X\ (C\ 1)$

$g = X$

$\mathrm{der}\ f = \mathrm{der}\ \bar{f} = \mathrm{Add}\ (C\ 1)\ (C\ 0)$

$h = \mathrm{der}\ (\mathrm{Mul}\ f\ g) = \mathrm{muld}\ (\mathrm{der}\ f)\ (\mathrm{der}\ g)$

$\quad = \mathrm{muld}\ (\quad)\ (C\ 1) = k =$ for fun.

$\bar{h} = \mathrm{der}\ (\mathrm{Mul}\ \bar{f}\ g) = \mathrm{muld}\ (\mathrm{der}\ \bar{f})\ (\mathrm{der}\ g)$

$\mathrm{eval}\ h = \mathrm{eval}\ (\mathrm{der}\ (\mathrm{Mul}\ f\ g)) = D\ (\mathrm{eval}\ (\mathrm{Mul}\ f\ g)) = D\ (\mathrm{eval}\ f * \mathrm{eval}\ g)$

$= D\ (id * id) = D\ (\backslash x \to x^2) = (2\cdot)$

spec. $H_1(\mathrm{eval}, \mathrm{der}, D)$

$\mathrm{eval}\ \bar{h} = \mathrm{eval}\ (\mathrm{der}\ (\mathrm{Mul}\ \bar{f}\ g)) = D\ (\mathrm{eval}\ (\mathrm{Mul}\ \bar{f}\ g)) = D\ (\mathrm{eval}\ \bar{f} * \mathrm{eval}\ g)$

$= D\ ((\backslash x \to x+1) * id) = D\ (\backslash x \to x^2 + x) = \backslash x \to 2\cdot x + 1$

$\mathrm{eval}\ h\ x = \mathrm{eval}\ \bar{h}\ x \Rightarrow 2\cdot x = 2\cdot x + 1 \Rightarrow 0 = 1$!

DSL → δσλ
DSLsofMath

Syntax for function expressions

Patrik Jansson

let $f = \text{Add}\ X\ (C\ 0)$

$\bar{f} = \text{Add}\ X\ (C\ 1)$

$g = X$

$\text{der}\ f = \text{der}\ \bar{f} = \text{Add}\ (C\ 1)\ (C\ 0)$

let $h = \text{muld}\ (\text{der}\ f)\ (\text{der}\ g)$

$= \text{der}\ (\text{Mul}\ f\ g)$

$\bar{h} = \text{der}\ (\text{Mul}\ \bar{f}\ g) = \text{muld}\ (\text{der}\ \bar{f})(\text{der}\ g)$

$= h$

$\text{eval}\ (\text{der}\ (\text{Mul}\ f\ g)) = D\ (\text{eval}\ (\text{Mul}\ f\ g)) =$

$D\ (\text{eval}\ f * \text{eval}\ g) = D\ (id * id) = D\ ({}^\wedge 2) = (2*) = \text{eval}\ h$

$\text{eval}\ (\text{der}\ (\text{Mul}\ \bar{f}\ g)) = \dots = D\ (\backslash x \to (x+1) * x) = \backslash x \to 2*x+1$

Thus $\text{eval}\ h\ x = 2*x \neq 2*x+1 = \text{eval}\ \bar{h}\ x$

Thus $h \neq \bar{h}$ but also $h = \bar{h}$. Contradiction.

DSL → δσλ
DSLsofMath

Syntax for function expressions & derivatives

Patrik Jansson

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

```
der : F → F
der (Add f g) = Add (der f) (der g)
der (Mul f g) =? muld (der f) (der g)
der X         = C 1
der (C c)     = C 0
```

No!

$\cancel{\exists\, muld}.\ H_2(der, Mul, muld) \wedge$
$eval \circ der = D \circ eval$

No such muld can ∃!

Thus der is not a homomorphism!

Syntax for function expressions

Patrik Jansson

& derivatives

```
data F where
  Add :: F → F → F
  Mul :: F → F → F
  X   :: F
  C   :: ℝ → F
```

```
der : F → F
der (Add f g) = Add (der f) (der g)
der (Mul f g) = m f g (der f) (der g)
der X         = C 1
der (C c)     = C 0
```

was muld

$m : F \to F \to F \to F \to F$

$m\ f\ g\ f'\ g' = \text{Add}\ (\text{Mul}\ f'\ g)\ (\text{Mul}\ f\ g')$

Thus der can be defined, but not as a homomorphism

DSLsofMath

Tupling to the rescue

$S = \mathbb{R} \to \mathbb{R}$

Patrik Jansson

Trig

$$\begin{tikzcd} F \arrow[r, "der2"] \arrow[dr, dashed, "evalD"'] & F \times F \arrow[d] \\ & S \times S \end{tikzcd}$$

f $\quad f'$

mulDer : $F \times F \to F \times F \to F \times F$

mulDer (f, f') (g, g') = (Mul f g, Add (Mul f' g) (Mul f g'))

mulD : $S \times S \to S \times S \to S \times S$

mulD (f, f') (g, g') = $(f * g, (f' * g) + (f * g'))$

Tupling to the rescue

$S = \mathbb{R} \to \mathbb{R}$

Patrik Jansson

$$mulD : (S, S) \to (S, S) \to (S, S)$$
$$mulD\,(f, f')\;(g, g') = (\,f*g\;,\;f'*g + f*g'\,)$$

$$mulDer : (F, F) \to (F, F) \to (F, F)$$
$$mulDer\,(f, f')\;(g, g') =$$
$$(Mul\;f\;g\;,\;Add\;(Mul\;f'\;g)\;(Mul\;f\;g')\;)$$

Tupling to the rescue

Patrik Jansson

$$\mathit{adder} : (F, F) \to (F, F) \to (F, F)$$
$$\mathit{adder}\ (f, f')\ (g, g') = (\mathit{Add}\ f\ g\ , \mathit{Add}\ f'\ g')$$

$$\mathit{mulder} : (F, F) \to (F, F) \to (F, F)$$

$$\mathit{der2} : F \to (F, F)$$
$$\mathit{der2}\ (\mathit{Add}\ f\ g) = \mathit{adder}\ \overbrace{(\mathit{der2}\ f)}^{(f, f')}\ \overbrace{(\mathit{der2}\ g)}^{(g, g')}$$
$$\mathit{der2}\ (\mathit{Mul}\ f\ g) = \mathit{mulder}\ (\mathit{der2}\ f)\ (\mathit{der2}\ g)$$
$$\mathit{der2}\ X = (X, C\ 1)$$
$$\mathit{der2}\ (C\ c) = (C\ c, C\ 0)$$

$$H_2(\mathit{der2}, \mathit{Add}, \mathit{adder})$$
$$H_2(\mathit{der2}, \mathit{Mul}, \mathit{mulder})$$

Tupling to the rescue

Patrik Jansson

$$\text{adder} : (F,F) \to (F,F) \to (F,F)$$
$$\text{adder}\ (f,f')\ (g,g') = (\text{Add}\ f\ g\ ,\ \text{Add}\ f'\ g')$$

$$\text{mulder} : (F,F) \to (F,F) \to (F,F)$$

$$\text{der2} : F \to (F,F)$$
$$\text{der2}\ (\text{Add}\ f\ g) = \text{adder}\ \overbrace{(\text{der2}\ f)}^{(f,f')}\ \overbrace{(\text{der2}\ g)}^{(g,g')}$$
$$\text{der2}\ (\text{Mul}\ f\ g) = \text{mulder}\ (\text{der2}\ f)\ (\text{der2}\ g)$$
$$\text{der2}\ X = (X,\ C\ 1)$$
$$\text{der2}\ (C\ c) = (C\ c,\ C\ 0)$$

der2 specification

Patrik Jansson

$f, f', fe : F$

$$\text{let } (f, f') = \text{der2 } fe$$
$$\text{in } (\underbrace{\text{eval } f}_{} == \underbrace{\text{eval } fe}_{}) \wedge (\underbrace{D\,(\text{eval } f)}_{\mathbb{R}\to\mathbb{R}} == \underbrace{\text{eval } f'}_{\mathbb{R}\to\mathbb{R}})$$

eval fe
||

$$\boxed{\text{der} = \text{snd} \circ \text{der2}}$$

```
der2 : F → (F, F)
der2 (Add f g) = addder (der2 f) (der2 g)
der2 (Mul f g) = mulder (der2 f) (der2 g)
der2 X         = (X, C 1)
der2 (C c)     = (C c, C 0)
```