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) ≟ 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" ?

Syntax for function expressions & derivatives

Patrik Jansson

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

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

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

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

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

$$\underbrace{\text{"der implements D"}}_{H_1(eval,\, der,\, D)}$$

Syntax for function expressions & derivatives

Patrik Jansson

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

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

$\text{eval} \circ \text{der} = D \circ \text{eval}$

Assume muld exists, look for $\bot$.

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

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

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

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

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

$$\begin{array}{ccc} \text{isP}(x+y) & \neq & \text{isP}(\bar{x}+y) \\ \| & & \| \\ \text{isP}\ 4 & \neq & \text{isP}\ 5 \\ \| & & \| \\ F & \neq & T \end{array}$$

Syntax for function expressions & derivatives

Patrik Jansson

```
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 $\perp$.

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

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 & derivatives

Patrik Jansson

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

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

$\lightning$

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

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

Syntax for function expressions

Patrik Jansson

& derivatives

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

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

$\mathit{eval} \circ \mathit{der} = D \circ \mathit{eval}$

Assume mulD exists, look for $\bot$.

$\mathit{der}\,(\mathit{Mul}\ f\ g) = \mathit{mulD}\ (\mathit{der}\ f)\ (\mathit{der}\ g)$

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

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

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

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

$g = X$

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

Syntax for function expressions

Patrik Jansson

```
let f = Add X (C 0)
    f̄ = Add X (C 1)
    g = X
```

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

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

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

eval (der (Mul f g)) = D (eval (Mul f g)) =

D (eval f * eval g) = D (id * id) = D (^2) = (2*) = eval h

eval (der (Mul $\bar{f}$ g)) = … = D (\x -> (x+1)*x) = \x -> 2*x+1

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

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

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 & 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) = m f g (der f) (der g)
der X         = C 1
der (C c)     = C 0
```

```
m : F → F → F → F → F
m f g f' g' = Add (Mul f' g) (Mul f g')
```

Thus der can be defined, but not as a homomorphism

Tupling to the rescue

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

Patrik Jansson

$$\text{mulD} : (S, S) \to (S, S) \to (S, S)$$

$$\text{mulD}\,(f, f')\;(g, g') = (\underbrace{f * g}_{\backslash x \to f x \cdot g x},\; f' * g + f * g')$$

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

$$\text{mulDe}\,(f, f')\;(g, g') =$$

$$(\text{Mul}\;f\;g,\; \text{Add}\;(\text{Mul}\;f'\;g)\;(\text{Mul}\;f\;g'))$$

Tupling to the rescue

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

Patrik Jansson

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

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

Tupling to the rescue

Patrik Jansson

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

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

Tupling to the rescue

Patrik Jansson

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

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

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

der2 specification

$f, f', fe : F$

Patrik Jansson

let $(f, f') = \text{der2}\ fe$

in $(\underbrace{\text{eval}\ f}\ \texttt{==}\ \underbrace{\text{eval}\ fe}) \land (\underbrace{D\ (\text{eval}\ f)}_{\mathbb{R}\to\mathbb{R}}\ \texttt{==}\ \underbrace{\text{eval}\ f'}_{\mathbb{R}\to\mathbb{R}})$

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

```
der2 : F → (F, F)
der2 (Add f g) = adder (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)
```