Linear Algebra

Patrik Jansson

DSL of this week:

- vectors
- linear transforms
- matrices

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

```haskell
class (Field s, AddGroup v) => VectorSpace v s where
    scale :: s -> v -> v
```

Two types: s for "scalars"

v for "vectors"

```haskell
data VE s = Zero | Add (VE s) (VE s)
          | Negate (VE s)  | Scale s (VE s)
          | Var String
```

Linear Algebra

Patrik Jansson

```
class (Field s, AddGroup v) => VectorSpace v s where
    scale :: s -> v -> v
```

Laws:

```
scale s zero    = zero
scale s (x+y)   = (scale s x) + (scale s y)
scale s (-x)    = -(scale s x)
```

$\forall s : s$
$\forall x, y : v$

```
scale zero x    = zero
scale (a+b) x   = (scale a x) + (scale b x)
scale (-s) x    = -(scale s x)
```

$\forall a, b, s : s$
$\forall x : v$

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

Laws:

$$\left.\begin{aligned} &\text{scale } s \text{ zero} = \text{zero} \\ &\text{scale } s\ (x+y) = (\text{scale } s\ x) + (\text{scale } s\ y) \\ &\text{scale } s\ (-x) = -(\text{scale } s\ x) \end{aligned}\right\} \begin{aligned} &\forall s : S \\ &\forall x, y : V \end{aligned}$$

$$\left.\begin{aligned} &\text{scale zero } x = \text{zero} \\ &\text{scale } (a+b)\ x = (\text{scale } a\ x) + (\text{scale } b\ x) \\ &\text{scale } (-s)\ x = -(\text{scale } s\ x) \end{aligned}\right\} \begin{aligned} &\forall a, b, s : S \\ &\forall x : V \end{aligned}$$

$$\left.\begin{aligned} &\text{scale one} = id \\ &\underbrace{\text{scale } (a*b)}_{: V \to V} = (\text{scale } a) \circ (\text{scale } b) \end{aligned}\right\} \forall a, b : S$$

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

Laws:

$$\text{scale } s \text{ zero} = \text{zero}$$
$$\text{scale } s\ (x+y) = (\text{scale } s\ x) + (\text{scale } s\ y)$$
$$\text{scale } s\ (-x) = -(\text{scale } s\ x)$$

$$\forall s.\ H_0(\text{scale } s, \text{zero}, \text{zero}) \wedge H_2(\text{scale } s, (+), (+)) \wedge H_1(\text{scale } s, \text{neg}, \text{neg})$$

$$\text{scale zero } x = \text{zero}$$
$$\text{scale } (a+b)\ x = (\text{scale } a\ x) + (\text{scale } b\ x)$$
$$\text{scale } (-s)\ x = -(\text{scale } s\ x)$$

$$H_0(\text{scale}, \text{zero}, \text{zero}_F) \wedge H_2(\text{scale}, (+), (+_F)) \wedge H_1(\text{scale}, \text{neg}, \text{neg}_F)$$

$$\text{scale one} = id$$
$$\text{scale } (a*b) = (\text{scale } a) \circ (\text{scale } b)$$

$$H_0(\text{scale}, \text{one}, id) \wedge H_2(\text{scale}, (*), (\circ))$$

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

instance Field s => VectorSpace s s where
  scale = (*)   -- :: $s \to s \to s \equiv s \to v \to v$

All scalars can be seen as "1-dimensional vectors".

$$\text{scale } s \; 1 = s * 1 = s$$

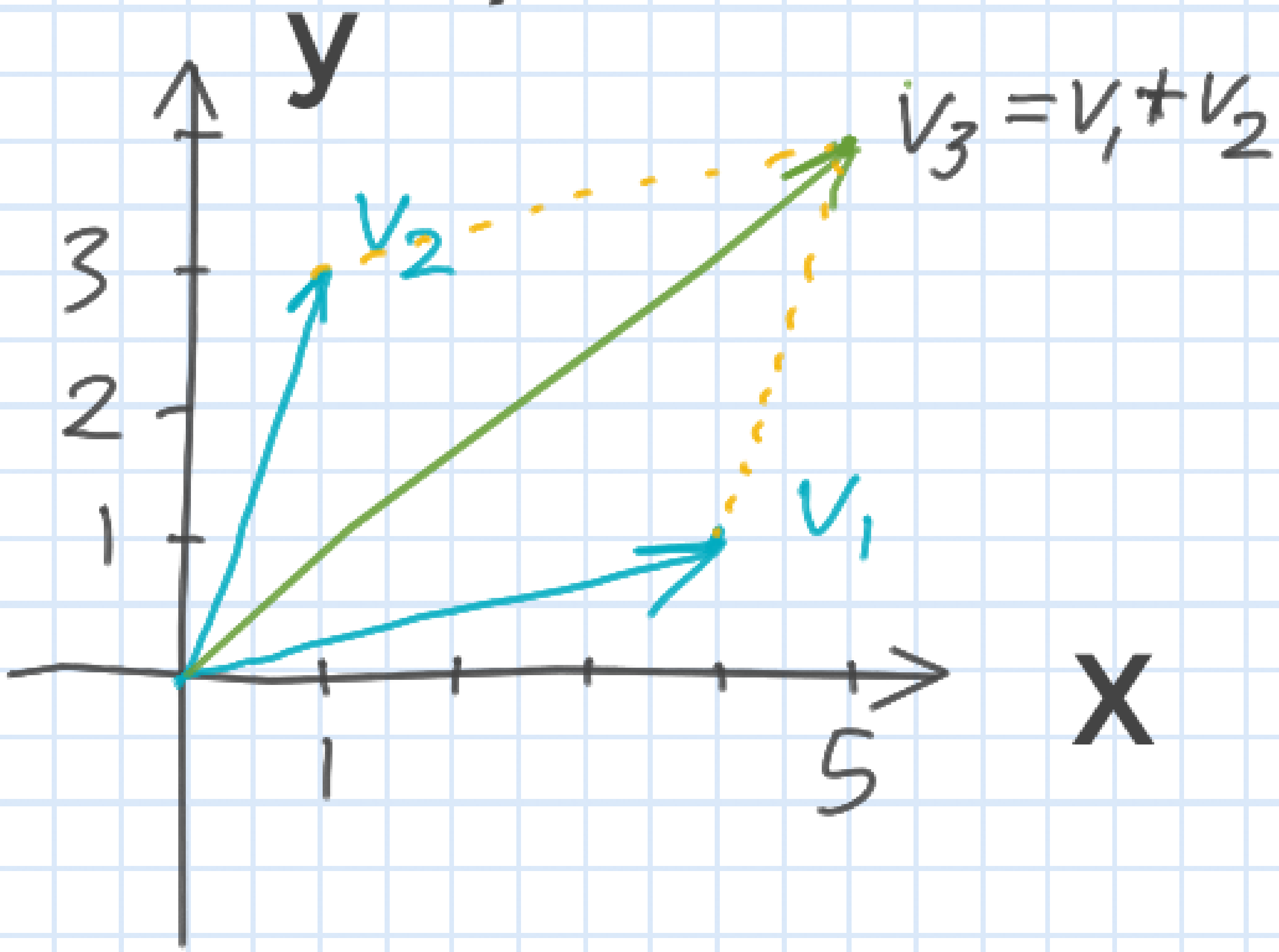
DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

```
class (Field s, AddGroup v) => VectorSpace v s where
    scale :: s -> v -> v

Example: data A = X | Y
         type TwoD = A -> ℝ
         v1 X = 4 ; v1 Y = 1 ;
         v2 X = 1 ; v2 Y = 3 ;
         v3 X = 5 ; v3 Y = 4 ;
```



DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

```
class (Field s, AddGroup v) => VectorSpace v s where
    scale :: s -> v -> v
```

Example:
```
data A = X | Y
type TwoD = A -> ℝ

instance VectorSpace TwoD ℝ where scale = scaleF
scaleF :: ℝ -> TwoD -> TwoD
scaleF s v = w = \a -> s * (v a) = (s*) ∘ v
    where w a = s * (v a)
```

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

```
instance Field s => VectorSpace (a -> s) s where
    scale = scaleF
scaleF :: Field s => s -> (a -> s) -> (a -> s)
scaleF s v = \i -> s * v i
```

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

linear combinations: $\sum_i a_i \cdot v_i = \sum_i \text{scale}\ a_i\ v_i$

```
linComb : (Finite g, VectorSpace v s) => (g -> s) -> (g -> v) -> v
```

linComb a v = $\sum_i$ scale (a i) (v i)

```
            = sum (map (\i -> scale (a i) (v i))
                       finiteDomain)
```

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

linear combinations: $\sum_i a_i \cdot v_i = \sum_i \text{scale } a_i\, v_i$

$\text{linComb} : (\text{Finite } g, \text{VectorSpace } v\ s) \Rightarrow (g \to s) \to (g \to v) \to v$

$\text{linComb } a\ v = \sum_i \text{scale } (a\ i)\ (v\ i)$

A collection of vectors $\{v_0, \dots v_n\}$ is $\boxed{\text{linearly independent}}$

iff $\forall a : \{0..n\} \to s.\quad (\text{linComb } a\ v = 0) \Leftrightarrow (\forall i.\ a_i = 0)$

$\left\{ e_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ e_Y = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

linear combinations: $\sum_i a_i \cdot v_i = \sum_i \text{scale } a_i \; v_i$

```
linComb : (Finite g, VectorSpace v s) => (g -> s) -> (g -> v) -> v
linComb a v = Σ_i scale (a i) (v i)
```

$e : G \to G \to s$

Basis for "vectors a functions": $G = \{0..n\}$

$$e_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad e_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad \ldots \quad e_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

pos. k

$e_i : G \to s$

$e_i \; j = \text{one}, \quad \text{if } i = j$

$ = \text{zero}, \quad \text{if } i \neq j$

Linear Algebra & Homomorphisms

Patrik Jansson

$$h : V \longrightarrow W \qquad \text{"linear transform"}$$

$$\begin{aligned}
LinTran(h, V, W) = \quad & H_0(h, zero_V, zero_W) \\
\wedge\ & H_2(h, (+_V), (+_W)) \\
\wedge\ & \forall s.\ H_1(h, scale_V\, s, scale_W\, s)
\end{aligned}$$

Cont. in DSLsofMath lecture 7.1b https://jamboard.google.com/d/1Kx-uI4J8zi4GejuNuSP-SxxydeqgHguychnbvpUiCxM/edit?usp=sharing

$DSL \to \delta\sigma\lambda$
DSLsofMath