Linear Algebra

Patrik Jansson

DSL of this week:

- vectors
- linear transforms
- matrices

DSL → δσλ

DSLsofMath

Linear Algebra

Patrik Jansson

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

Two types: s for "scalars"
v for "vectors"

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

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

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

Laws:

$$\left.\begin{array}{l} \text{scale } c \text{ zero} = \text{zero} \\ \text{scale } c\ (x+y) = (\text{scale } c\ x) + (\text{scale } c\ y) \\ \text{scale } c\ (-x) = -(\text{scale } c\ x) \end{array}\right\} \begin{array}{l} \forall c : s \\ \forall x, y : v \end{array}$$

$$\left.\begin{array}{l} \text{scale zero } x = \text{zero} \\ \text{scale } (a+b)\ x = (\text{scale } a\ x) + (\text{scale } b\ x) \\ \text{scale } (-c)\ x = -(\text{scale } c\ x) \end{array}\right\} \begin{array}{l} \forall a, b, c : s \\ \forall x : v \end{array}$$

Linear Algebra

Patrik Jansson

Laws:

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

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

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

Linear Algebra

Patrik Jansson

Laws:

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

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

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

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

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

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

DSL → δσλ
DSLsofMath

Linear Algebra

Patrik Jansson

```
instance Field s => VectorSpace s s where
  scale = (*)      -- :: s -> s -> s ≡ s -> v -> v
```

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

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

Linear Algebra

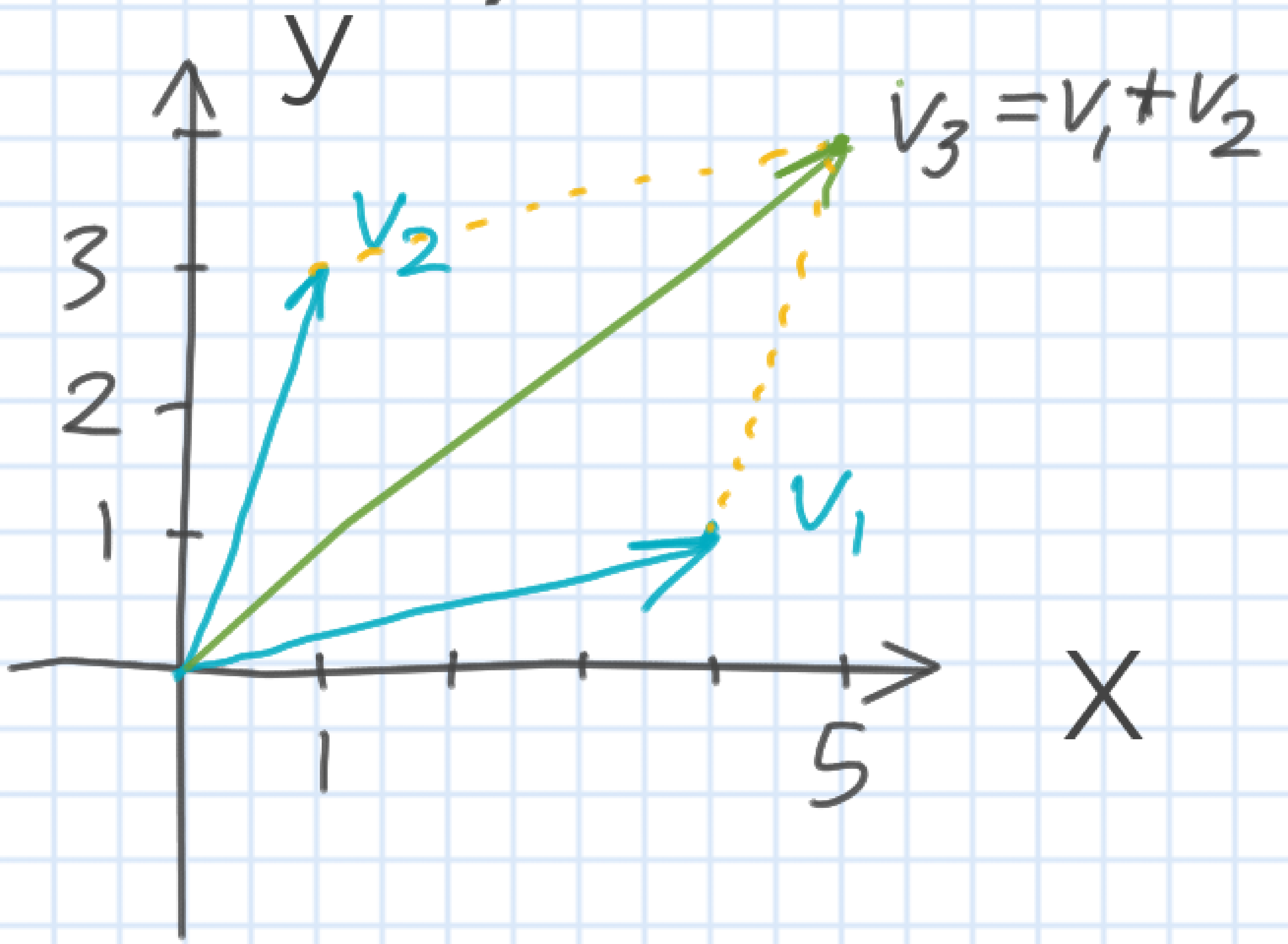
Patrik Jansson

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

Example:

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

v1 X = ; v1 Y = ;
v2 X = ; v2 Y = ;
v3 X = ; v3 Y = ;
```



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
```

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
```

Linear Algebra

Patrik Jansson

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

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

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

$$= \text{sum } (\text{map } (\backslash i \to \text{scale } (a\ i)\ (v\ i))\ \text{finiteDomain})$$

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 $(v_0, \ldots, v_n)$ is $\boxed{\text{linearly independent}}$

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

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)

$e : G \to G \to s$

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

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

pos. k

$e_i : G \to s$

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

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

Linear Algebra & Homomorphisms

Patrik Jansson

$$h : V \longrightarrow W$$ "linear transform"

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

Cont. in DSLsofMath lecture 7.1b https://jamboard.google.com/d/1Kx-uI4J8zi4GejuNuSP-SxxydeqgHguychnbvpUiCxM