axle

Adam Pingel
May 23, 2015

Algebra: MacLane & Birkhoff

Monoids

Scalaz's Nick Partridge derives
Monoid, Foldable, and more at
vimeo.com/10482466

learnyouahaskell.com

Post-Monoid

Nov 12, 2012 Oscar Boykin and Sam Ritchie took questions about Algebird at [Cascading meetup](#) (Twitter)

February 2013 "Life After Monoids" Tom Switzer (NE Scala) youtube.com/watch?v=xO9AoZNSOH4

August 2014 "Reasoning with Types" Adelbert Chang (Scala by the Bay) youtube.com/watch?v=TbASMeJSh9s

Structure	Composed of	Examples
Semigroup	an associative binary operator	Integer +, string concat
Monoid	a semigroup with identity element	(+, 0), (*, 1), (concat, "")
Group	a set of elements together with an operation that combines any two elements to form a third element. The operation satisfies closure, associativity, identity and invertibility.	Integer + Rubik's Cube group
Abelian Group	a group in which the result of applying the group operation to two group elements does not depend on their order	Integer +
Ring	an abelian group with a second binary operation that is associative, is distributive over the abelian group operation and has an identity element	2x2 matrix + and *
Field	a nonzero commutative ring that contains a multiplicative inverse for every nonzero element	(+, -, *, /) for $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{Q}$
Vector Space	a collection of objects called **vectors**, which may be added together and multiplied ("scaled") by numbers, called *scalars* in this context	$\mathbf{R}^2$

Diverse Algorithms

Units of Measurement	Algebra
Machine Learning	Games
Visualization	Statistics
Probabilistic Graphical Models	Bioinformatics
Natural Language Processing	...

Diseño Platform
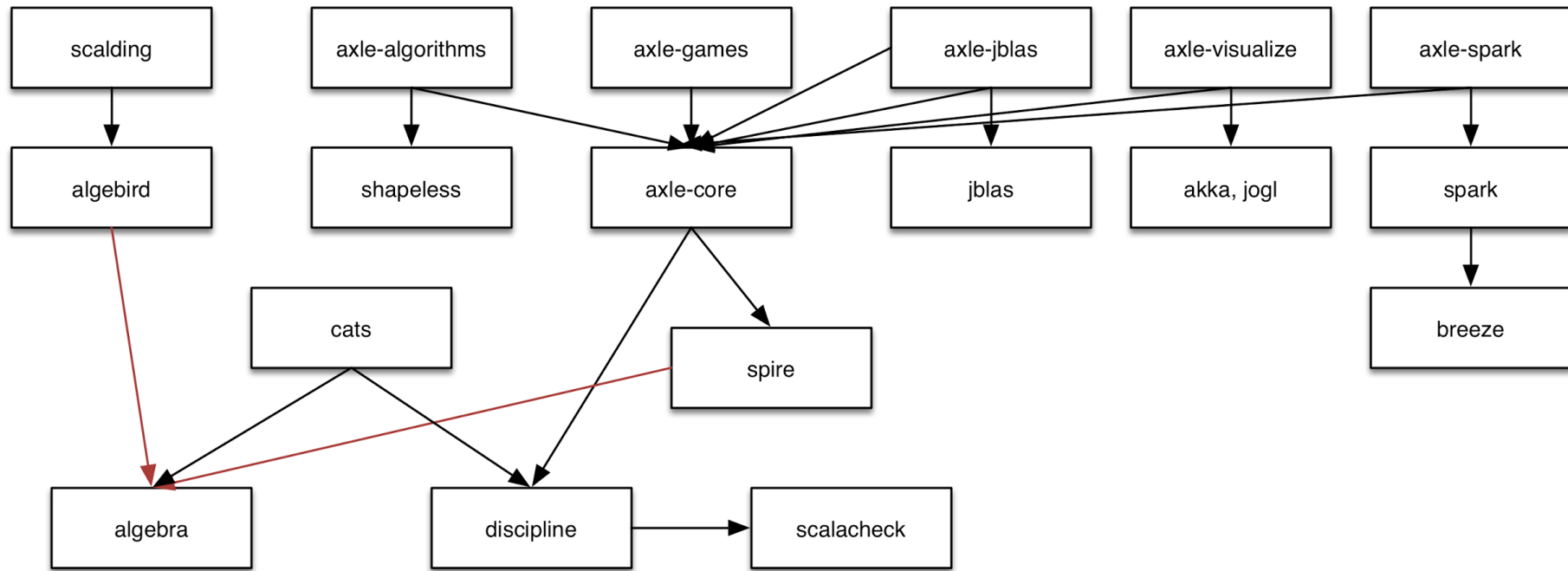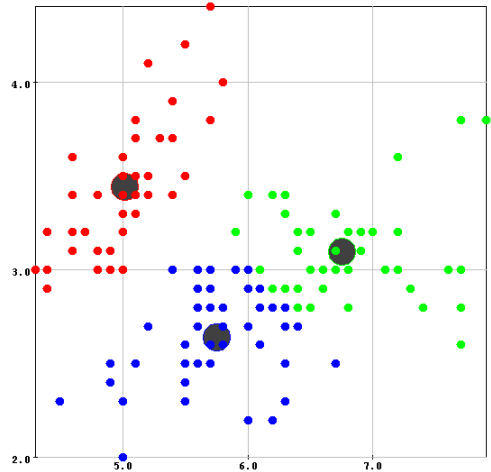
Docs (axle-lang.org)

```
6   val classifier = KMeans[Iris, List, DoubleMatrix](
7       irises,
8       2,
9       (iris: Iris) => List((iris.sepalLength in cm).magnitude.toDouble, (iris.sepalWidth in cm).magnitude.
10      (PCAFeatureNormalizer[DoubleMatrix] _).curried.apply(0.98),
11      (features: Seq[Double]) => Iris(1 *: cm, 1 *: cm, 1 *: cm, 1 *: cm, ""),
12      K = 3,
13      iterations = 20)
```

Produce a "confusion matrix"

```
1   import axle.ml.ConfusionMatrix
2   import spire.implicits.IntAlgebra
3
4   val confusion = ConfusionMatrix[Iris, Int, String, Vector, DoubleMatrix](classifier, irises.toVector, _
5
6   string(confusion)
```

```
 1  49   0 :  50 Iris-setosa
34   0  16 :  50 Iris-versicolor
16   0  34 :  50 Iris-virginica


51  49  50
```

… and …

Correct	Some ScalaCheck properties (to be organized with Discipline)
Complete	20k lines of code, but lots of "???" yet
Fast	No benchmarking yet
Diverse Data	A few small datasets available in axle.data

Feedback welcome!

@axlelang   @pingel

axle-lang.org

gitter.im/axlelang/axle

github.com/axlelang/axle

```
"org.axle-lang" %% "axle-core" % "0.2.0-SNAPSHOT"
```