



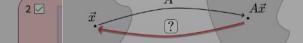
2

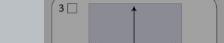
Linear Algebra

1 Vector $c_1 = \langle a, b \rangle$
2 Vector $c_2 = \langle c, d \rangle$
3 Matrix $M = \text{columns}(c_1, c_2)$
4 LinearlyDependent(c_1, c_2)

Select diagrams that correspond to the
LinearlyDependent(\vec{a})

1 

2 

3 

4 

Correct Answer
 A^{-1}

Designing Declarative Language Tutorials:
a Guided and Individualized Approach

Anael Kuperwajs Cohen, Wode Ni, Joshua Sunshine

15



**Defining Visual Mathematical Narratives
Declaratively**

Max Krieger, Wode Ni, Joshua Sunshine

28



What is Penrose?

Wode "Nimo" Ni

Math notations are great!

Figure 3.7 The projection p of b onto a , with $\cos \theta = \frac{a^T b}{\|a\| \|b\|}$

be negative:

$$\left\| b - \frac{a^T b}{a^T a} a \right\|^2 = b^T b - 2 \frac{(a^T b)^2}{a^T a} + \left(\frac{a^T b}{a^T a} \right)^2$$

This tells us that $(b^T b)(a^T a) \geq (a^T b)^2$ —and therefore

3i All vectors a and b satisfy the **Schwarz inequality**

$$a^T b \leq \|a\| \|b\|$$

According to formula (2), the ratio between $a^T b$ and $\|b\|$ all cosines lie in the interval $-1 \leq \cos \theta \leq 1$, thus the Schwarz inequality is the same as $|\cos \theta| \leq 1$ understood proof, because cosines are so familiar. Notice that ours came directly from the calculation when we introduce new possibilities for the length Cauchy is also attached to this inequality $|a^T b| \leq \|a\| \|b\|$ the Cauchy-Schwarz-Buniakowsky inequality! Now that Buniakowsky's claim is genuine.

One final observation about $|a^T b| \leq \|a\| \|b\|$ is that it is identical with its projection p , the distance

Example 1 Project $b = (1, 2, 3)$ onto the line through $a = (1, 1, 1)$.

$$\hat{x} = \frac{a^T b}{a^T a} = \frac{6}{3} = 2$$

The projection is $p = \hat{x}a = (2, 2, 2)$. The angle

$$\cos \theta = \frac{\|p\|}{\|b\|} = \frac{\sqrt{12}}{\sqrt{14}} \quad \text{and also } \cos \theta = \frac{a^T b}{\|a\| \|b\|}$$

The Schwarz inequality $|a^T b| \leq \|a\| \|b\|$ is $6 \leq \sqrt{36} \leq \sqrt{42}$. The cosine is less than

Dem.—On AB describe the square $ABDE$. Join BE . Through C draw CG parallel to AE , intersecting BE in F . Through F draw HK parallel to AB .

Now the square AD is equal to the three figures AK , FD , and GH : to each add the square CK , and we have the sum of the squares AD , CK equal to the sum of the three figures AK , CD , GH ; but CD is equal to AK ; therefore the sum of the squares AD , CK is equal to twice the figure AK , together with the figure GH . Now AK is the rectangle AB , BK ; but BK is equal to BC ; therefore AK is equal to the rectangle AB , BC , and AD is the square on AB ; CK the square on CB ; and GH square on HF , and therefore equal to the square on AC . Hence the sum of the squares on AB and BC is equal to twice the rectangle AB , BC , together with the square on AC .

Or thus: On AC describe the square $ACDE$. Produce the sides CD , DE , EA , and make each produced part equal to CB . Join BF , FG , GH , HB . Then the figure $BFGH$ is a square [I. XLVI, Ex. 3], and it is equal to the square on AC , together with the four equal triangles HAB , BCF , FDG , GEH . Now [I. XLVII], the figure $BFGH$ is equal to the sum of the squares on AB , AH —that is, equal to the sum of the squares on AB , BC ; and the sum of the four triangles is equal to twice the rectangle AB , BC , for each triangle is equal to half the rectangle AB , BC . Hence the sum of the squares on AB , BC is equal to twice the rectangle AB , BC , together with the square on AC .

Or thus:

therefore

therefore

$$AC = AB - BC;$$

$$AC^2 = AB^2 - 2AB \cdot BC + BC^2;$$

$$AC^2 + 2AB \cdot BC = AB^2 + BC^2.$$

Comparison of iv. and viii.

By iv., square on sum = sum of squares + twice rectangle.

By viii., square on difference = sum of squares-twice rectangle.

Cors. from iv. and vii.

1. Square on the sum, the sum of the squares, and the square on the difference of any two lines, are in arithmetical progression.

2. Square on the sum + square on the difference of any two lines = the sum of the squares on the lines (Props. ix. and x.).

3. The square on the sum – the square on the difference of any two lines = four times the rectangle under lines (Prop. viii.).

Again, since CB is equal to CG , the rectangle $AC \cdot CB$ is equal to the rectangle $AC \cdot CG$; but $AC \cdot CG$ is the figure AG (Def. iv.). Therefore the rectangle $AC \cdot CB$ is equal to the figure AG . Now the figures AG , GD are equal [I. XLIII.], being the complements about the diagonal of the parallelogram AD . Hence the parallelograms AG , GD are together equal to twice the rectangle $AC \cdot CB$. Again, the figure HF is the square on HG , and HG is equal to AC . Therefore HF is equal to the square on AC , and CI is the square on CB ; but the whole figure AD , which is the square on AB , is the sum of the four figures HF , CI , AG , GD . Therefore the square on AB is equal to the sum of the squares on AC , CB , and twice the rectangle $AC \cdot CB$.

Or thus: On AB describe the square $ABDE$, and cut off AH , EG , DF each equal to CB . Join CF , FG , GH , HC . Now the four $\triangle ACH$, CBF , FDG , GEH are evidently equal; therefore their sum is equal to four times the $\triangle ACH$; but the $\triangle ACH$ is half the rectangle $AC \cdot AH$ (I. Cor. 2)—that is, equal to half the rectangle $AC \cdot CB$. Therefore the sum of the four triangles is equal to $2AC \cdot CB$.

Again, the figure $CFGH$ is a square [I. XLVI, Cor. 3], and equal to $AC^2 + AH^2$ [I. XI. VIII.]—that is, equal to $AC^2 + CB^2$. Hence the whole figure $ABDE = AC^2 + CB^2 + 2AC \cdot CB$.

Or thus: $AB = AC + CB$.

Squaring, we get

$$AB^2 = AC^2 + 2AC \cdot CB + CB^2.$$

Cor. 1.—The parallelograms about the diagonal of a square are squares.

Cor. 2.—The square on a line is equal to four times the square on its half.

For let $AB = 2AC$, then $AB^2 = 4AC^2$.

This Cor. may be proved by the First Book thus: Erect CD at right angles to AB , and make $CD = AC$ or CB . Join AD , DB .

$$\text{Then } AD^2 = AC^2 + CD^2 = 2AC^2$$

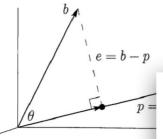
$$\text{In like manner, } DB^2 = 2CB^2;$$

$$\text{therefore } AD^2 + DB^2 = 2AC^2 + 2CB^2 = 4AC^2.$$

But since the angle ADB is right, $AD^2 + DB^2 = AB^2$;

$$\text{therefore } AB^2 = 4AC^2.$$

Cor. 3.—If a line be divided into any number of parts, the square on the whole is equal to the sum of the squares on all the parts, together with twice the sum of the rectangles contained by the several distinct pairs of parts.



Journal of Health Politics, Policy and Law

10 of 10

-3-

Digitized by srujanika@gmail.com

Glossary

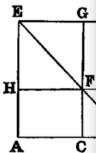
the first time in history, the world's
population has reached 7 billion.

Section 2: Reporting to the Board - 1

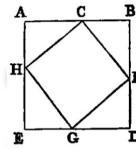
第二部分：基础与应用

The Journal of Neuroscience, July 1, 2009 • 29(27):8533–8543 • 8539

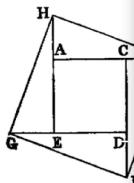
Digitized by srujanika@gmail.com



These data will be used to fit the model of $\phi(t)$ to a curve to be compared with the data of $\phi(t)$ in the theory of the state $\phi(t)$. Then the energy $E(t)$ is used to compare with the data of $E(t)$. From the theory of the state $\phi(t)$, we expect to obtain the following results.



This is the information about the library.
This is the space in which it used to be.
This is what it used to hold.
This is all that has ever been.
This is what it used to look like.
This is how it used to be used.



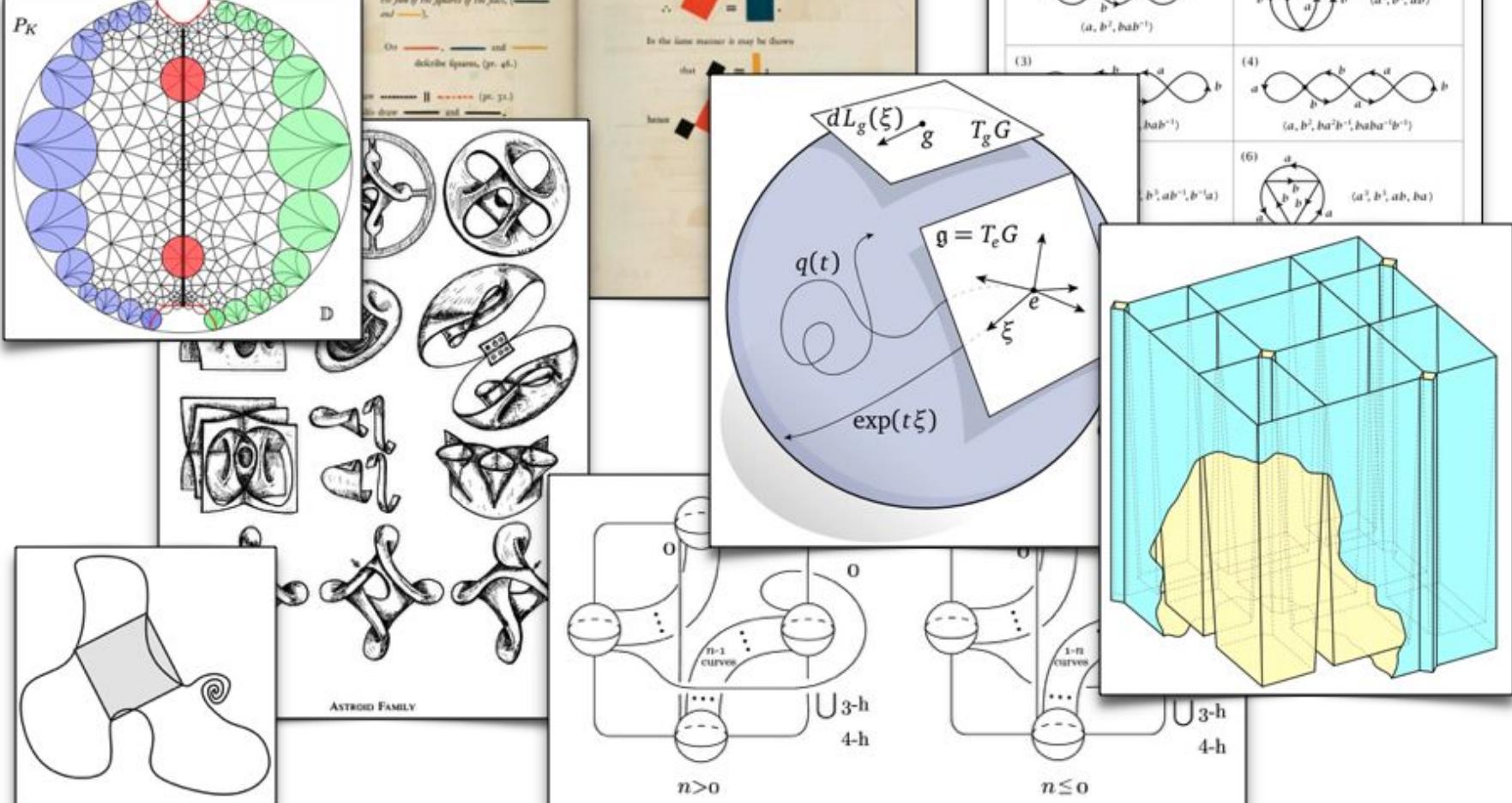
A diagram showing a triangle with vertices labeled A, B, and C. Vertex D is located above vertex B. A vertical line segment BD connects vertex B to vertex D. This segment is labeled "altitude".

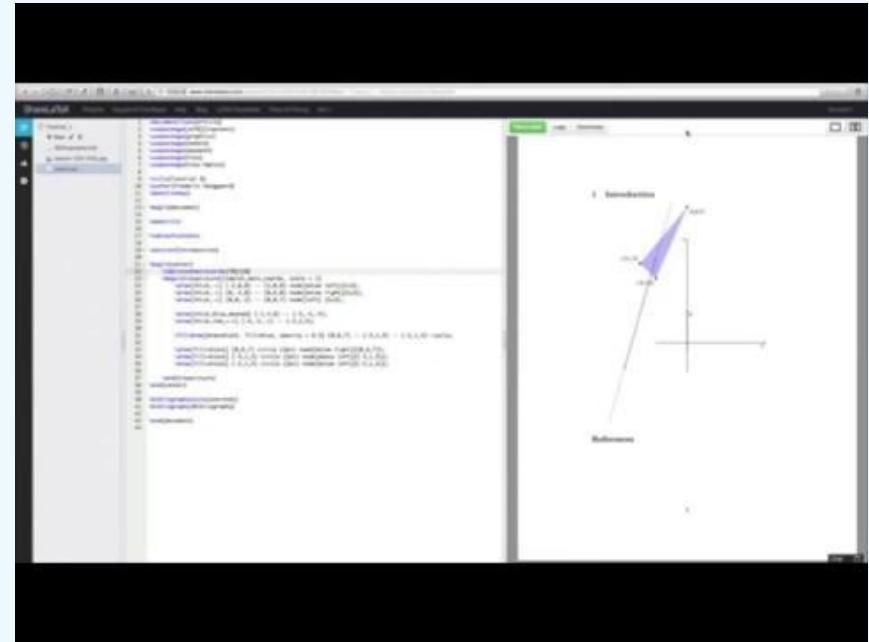
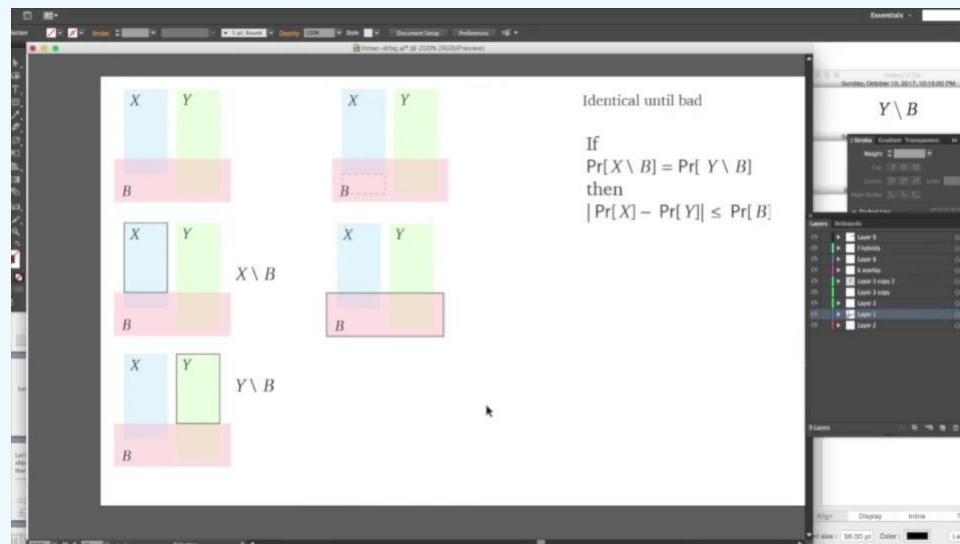
None	$10^7 \times 10^7 = 10^{14}$
In the center	$10^9 \times 10^9 = 10^{18}$
Outside	$10^7 \times 10^7 \times 10^7 \times 10^7 = 10^{28}$
One year the angle $360^\circ \times \pi \approx 10^7 \times 10^7 = 10^{14}$	
Outside	$10^7 \times 10^7$

Fig. 3. If a line is drawn from one vertex of a triangle, the angle at the point where it meets the base of the triangle will be less than the angle at the other vertex.

But that's not the whole story!

But that's not the whole story picture!





Let's make a diagram

Manipulation of low-level attributes
Meaning of the diagram is **lost**

Let's make a diagram (painfully)

Domain

Defines all possible notation in a domain to be visualized

set theory \cap

Real Analysis \mathbb{R}

Ray Tracing 

Linear Algebra \vec{v}

Neural Networks 

...

Substance

Declare objects and relationships with high-level notation

```
Set A, B  
A ⊓ B
```

Style

Map the objects and relationships to a concrete visual *representation*

```
Set X { shape = Circle{} }  
X ⊓ Y {  
    ensure X.shape contains Y.shape  
}
```

Introducing Penrose



set theory

venn

run

sign in

larger venn

sub sty dsl

editing fill

```
1 Set A, B, C, D, E, F, G
2
3 IsSubset(B, A)
4 IsSubset(C, A)
5 IsSubset(D, B)
6 IsSubset(E, B)
7 IsSubset(F, C)
8 IsSubset(G, C)
9
10 NotIntersecting(E, D)
11 NotIntersecting(F, G)
12 NotIntersecting(B, C)
13
14 AutoLabel All
15
```



Click run to render your diagram.

resample

autostep (off)

download

Let's make a diagram again



set theory

venn

run

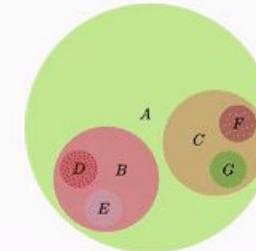
sign in

larger venn

sub sty dsl

editing fill

```
1 Set A, B, C, D, E, F, G
2
3 IsSubset(B, A)
4 IsSubset(C, A)
5 IsSubset(D, B)
6 IsSubset(E, B)
7 IsSubset(F, C)
8 IsSubset(G, C)
9
10 NotIntersecting(E, D)
11 NotIntersecting(F, G)
12 NotIntersecting(B, C)
13
14 AutoLabel All
15
```



resample



autostep (on)

download

Buy 1 get n free



set theory

venn

run

sign in

venn 3d

sub

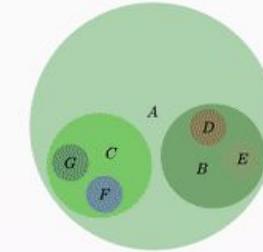
sty

dsl

editing

fill

```
1 Set A, B, C, D, E, F, G  
2  
3 IsSubset(B, A)  
4 IsSubset(C, A)  
5 IsSubset(D, B)  
6 IsSubset(E, B)  
7 IsSubset(F, C)  
8 IsSubset(G, C)  
9  
10 NotIntersecting(E, D)  
11 NotIntersecting(F, G)  
12 NotIntersecting(B, C)  
13  
14 AutoLabel All  
15
```



resample

autostep (on)

download



Diagramming with style(s)

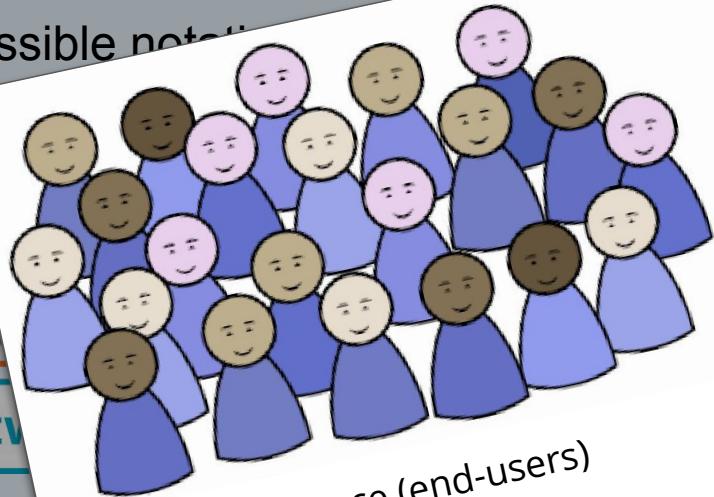
Domain

Defines all possible notations
to be visualized

set theory

Ray Tracing

Neural Networks



Substance (end-users)



Domain/Style (experts)

ensure X.shape contains Y.shape

}

So many languages!

Domain

Defines all possible notation in a domain
to be visualized

set theory \cap

Real Analysis \mathbb{R}

Ray Tracing 

Linear Algebra \vec{v}

Neural Networks 

• • •

Substance

Declare objects and relationships
with high-level notation

```
Set A, B  
A ⊂ B
```

Style

Map the objects and relationships to a
concrete visual *representation*

```
Set X { shape = Circle{} }  
X ⊂ Y {  
    ensure X.shape contains Y.shape  
}
```

What's in the Domain?

Domain

Defines all possible notation in a domain
to be visualized

set theory \cap

```
type Set
predicate IsSubset : Set s1 * Set s2
notation "A ⊆ B" ~ "IsSubset(A, B)"
```

Substance

Declare objects and relationships
with high-level notation

```
Set A, B
A ∩ B
```

Style

Map the objects and relationships to a
concrete visual *representation*

```
Set X { shape = Circle{} }
X ∩ Y {
    ensure X.shape contains Y.shape
}
```

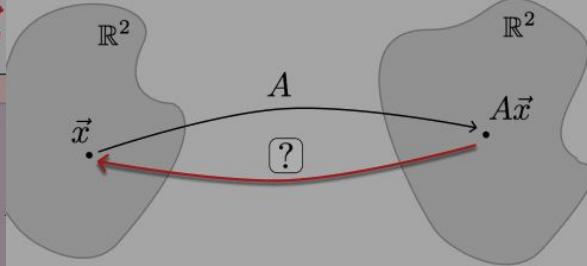
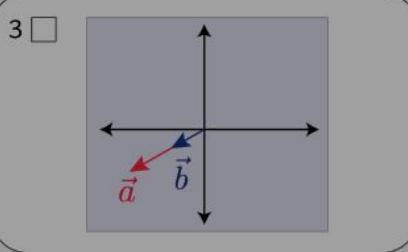
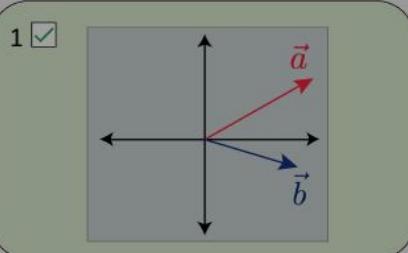
What's in the Domain?

Linear Algebra \vec{v}

Cartesian

```
1 Vector c1 = <a, b>
2 Vector c2 = <c, d>
3 Matrix M = columns(c1, c2)
4 LinearlyDependent(c1, c2)
```

Select diagrams that correspond to the

LinearlyDependent(\vec{a})

Correct Answer
 A^{-1}

Designing Declarative Language Tutorials: a Guided and Individualized Approach

Anael Kuperwajs Cohen, Wode Ni, Joshua Sunshine



set theory

venn-3d

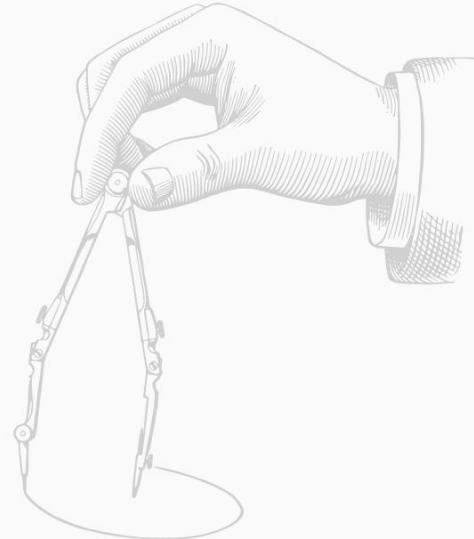
run share

My Diagram

1

editing

fill



Click run to render your diagram.

resample

autostep (off)

download

So...what now?

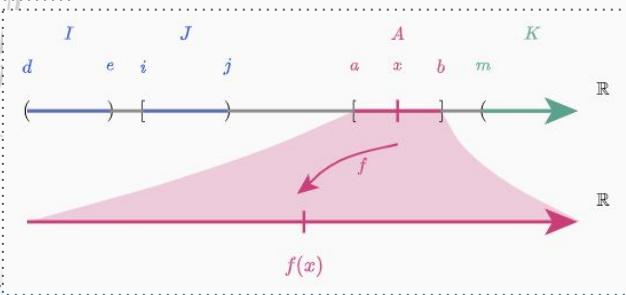
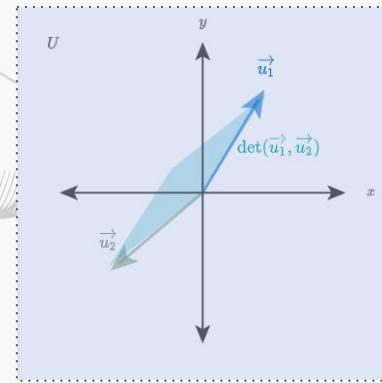
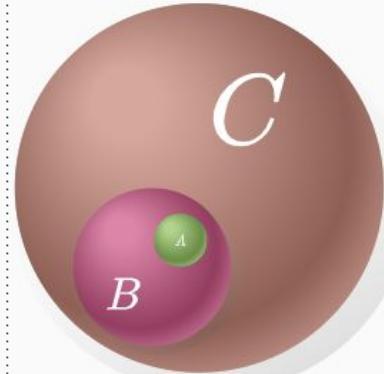


set theory

run share

My Diagram

1

Set A, B, C IsSubset(A, B)IsSubset(B, C)VectorSpace U Vector $u_1 \in U$ Vector $u_2 \in U$ Scalar $c := \det_R(u_1, u_2)$ $A := [a, b] \subseteq R$ Real $l \in A$ $f : A \rightarrow R$ Continuous(f)Real f_l $f_l := f(l)$ Real d, e, i, j $I := (d, e) \subseteq R$ R $J := [i, j] \subseteq R$ 

resample

download

Too many DSLs!

Higher-order Functions

Scala allows the definition of higher-order functions. These are functions that *take other functions as parameters*, or whose *result is a function*. Here is a function `apply` which takes another function `f` and a value `v` and applies function `f` to `v`:

```
1. def apply(f: Int => String, v: Int) = f(v)
```

Note: methods are automatically coerced to functions if the context requires this.

Here is another example:

```
1. class Decorator(left: String, right: String) {
2.   def layout[A](x: A) = left + x.toString() + right
3. }
4.
5. object FunTest extends App {
6.   def apply(f: Int => String, v: Int) = f(v)
7.   val decorator = new Decorator("[", "]")
8.   println(apply(decorator.layout, 7))
9. }
```

made up of packages.

in package `main`.

the packages with import paths "fmt"

ake name is the same as the last
ath. For instance, the "math/rand"
s that begin with the statement `package`

t in which these programs are executed
h time you run the example program
he same number.

ber, seed the number generator; see
stant in the playground, so you will
else as the seed.)



Here's some simple arithmetic.

```
ghci> 2 * 15
17
ghci> 49 * 100
4900
ghci> 1892 - 1472
420
ghci> 5 / 2
2.5
ghci>
```

This is pretty self-explanatory. We can also use several operators on one line and all the usual precedence rules are obeyed. We can use parentheses to make the precedence explicit or to change it.

```
ghci> (50 * 100) - 4999
1
ghci> 50 * 100 - 4999
1
ghci> 50 * (100 - 4999)
-244950
```

Pretty cool, huh? Yeah, I know it's not but bear with me. A little pitfall to watch out for here is negating numbers. If we want to have a negative number, it's always best to surround it with parentheses. Doing `5 * -3` will make GHCI yell at you but doing `5 * (-3)` will work just fine.

Boolean algebra is also pretty straightforward. As you probably know, `&&` means a boolean and, `||` means a boolean or. `not` negates a `True` or a `False`.

```
ghci> True && False
False
ghci> True && True
True
ghci> False || True
True
ghci> not False
True
ghci> not (True && True)
False
```

Testing for equality is done like so.

```
ghci> 5 === 5
True
ghci> 1 === 0
False
ghci> 5 /= 5
False
ghci> 5 /= 4
True
ghci> "hello" === "hello"
True
```

What about doing `5 + "llama"` or `5 == True`? Well, if we try the first snippet, we get a big scary error message!

```
No instance for (Num [Char])
arising from a use of '+' at <interactive>:1:0-9
Possible fix: add an instance declaration for (Num [Char])
in the expression: 5 + "llama"
```

Building your first TypeScript file =

In your editor, type the following JavaScript code in `greeter.ts`:

```
function greeter(person) {
  return "Hello, " + person;
}

let user = "Jane User";

document.body.textContent = greeter(user);
```

Compiling your code =

We used a `.ts` extension, but this code is just JavaScript. You could have copy/pasted this straight out of an existing JavaScript app.

At the command line, run the TypeScript compiler:

```
tsc greeter.ts
```

The result will be a file `greeter.js` which contains the same JavaScript that you fed in. We're up and running using TypeScript in our JavaScript app!

Now we can start taking advantage of some of the new tools TypeScript offers. Add a `: string` type annotation to the 'person' function argument as shown here:

```
function greeter(person: string) {
  return "Hello, " + person;
}

let user = "Jane User";
```

Higher-order Functions

Scala allows the definition of higher-order functions. These are functions that *take other functions as parameters*, or whose *result is a function*. Here is a function `apply` which takes another function `f` and a value `v` and applies function `f` to `v`:

```
1. def apply(f: Int => String, v: Int) = f(v)
```

Note: methods are also functions, so you can pass them around like any other value.

Here is another example:

```
1. class Decorator(left: String, right: String) {
2.   def layout(x: String) = left + x + right
3. }
4.
5. object FunTest extends App {
6.   def apply(f: Int => String, v: Int) = f(v)
7.   val decorator = new Decorator("[", "]")
8.   println(apply(decorator.layout, 7))
9. }
```

made up of packages.

in package `main`.

the packages with import paths "fmt"

package name is the same as the last path. For instance, the "math/rand" packages that begin with the statement `package`

in which these programs are executed each time you run the example program the same number.

number, seed the number generator; see instant in the playground, so you will else as the seed.)



Language tutorials are hard to follow!

here's some simple arithmetic.

```
ghci> 2 + 15
17
ghci> 49 * 100
4900
ghci> 1892 - 1472
420
ghci> 5 / 2
2.5
```

This is pretty easy to follow. In a language with operators, however, all the usual precedence rules are obeyed. We can use parentheses to make the precedence explicit or to change it.

```
ghci> (50 * 100) - 244
1
ghci> 50 * 100 - 244
244
```

pretty cool, huh? Yeah, I know it's not bear with me. A little pitfall to watch out for here is negating numbers. If we want to save a negative number, it's always best to surround it with parentheses. Doing `5 * -3` will make GHCi yell at you but doing `* (-3)` will work just fine.

boolean algebra is also pretty straightforward. As you probably know, `&&` means a boolean and, `||` means a boolean or, `not` negates a `True` or a `False`.

```
ghci> True && False
False
ghci> True && True
True
ghci> False || True
True
ghci> not False
True
ghci> not (True && True)
False
```

Testing for equality is done like so.

```
ghci> 5 == 5
True
ghci> 1 == 0
False
ghci> 5 /= 5
False
ghci> 5 /= 4
True
ghci> "hello" == "hello"
True
```

What about doing `5 + "llama"` or `5 == True`? Well, if we try the first snippet, we get a big scary error message!

```
No instance for (Num [Char])
arising from a use of '+' at <interactive>:1:0-9
Possible fix: add an instance declaration for (Num [Char])
In the expression: 5 + "llama"
```

Building your first TypeScript file

In your editor, type the following JavaScript code in `greeter.ts`:

```
function greeter(person) {
  return "Hello, " + person;
}
```

```
let user = "Jane User";
```

```
document.body.textContent = greeter(user);
```

Compiling your code

We used a `.ts` extension, but this code is just JavaScript. You could have copy/pasted this straight out of an existing JavaScript app.

At the command line, run the TypeScript compiler:

```
tsc greeter.ts
```

The result will be a file `greeter.js` which contains the same JavaScript that you fed in. We're up and running using TypeScript in our JavaScript app!

Now we can start taking advantage of some of the new tools TypeScript offers. Add a `: string` type annotation to the `'person'` function argument as shown here:

```
function greeter(person: string) {
  return "Hello, " + person;
}
```

```
let user = "Jane User";
```



Video game tutorials?

Objective



Question 1 of 5

Show two linearly dependent vectors

```
1 Autolabel All  
2 VectorSpace U  
3 In(A, U)  
4 In(B, U)
```

```
6 Vector A  
7 Vector B
```

```
8 Vector C  LinearilyDependent(A, B)  
Scalere s LinearilyDependent(B, C)
```

Objective

Show two linearly dependent vectors

```
1 Autolabel all  
2 VectorSpace U  
3 In(A, U)  
4 In(B, U)  
5  
6 Vector A  
7 Vector B  
8  
9
```

Diagram - Click 'Run'

Add linear dependency, type "LinearlyDependent(A, B)"



Run

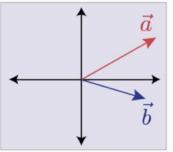
Check

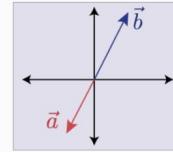
A guided tour of Penrose

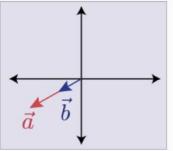
Recognition: Recognizing the associations between program outputs and source programs

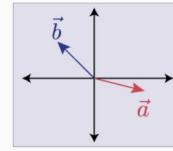
Question 1 of 5

Select diagrams that correspond to the following statement
 \vec{a} and \vec{b} are linearly independent

1 

2 

3 

4 

check

Individualized practice

Question 1 of 5

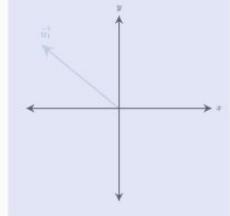
Diagram - Click 'Run'

editing fill

Show two linearly dependent vectors, \vec{u}_1 and \vec{u}_2

1 VectorSpace U
2 Vector $u1 \in U$
3 Label $u1 \ $\\vec{u_{1}}\$$

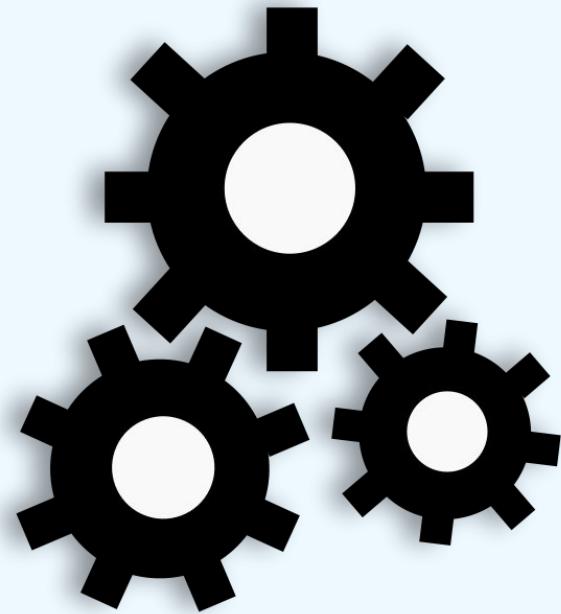
Untitled



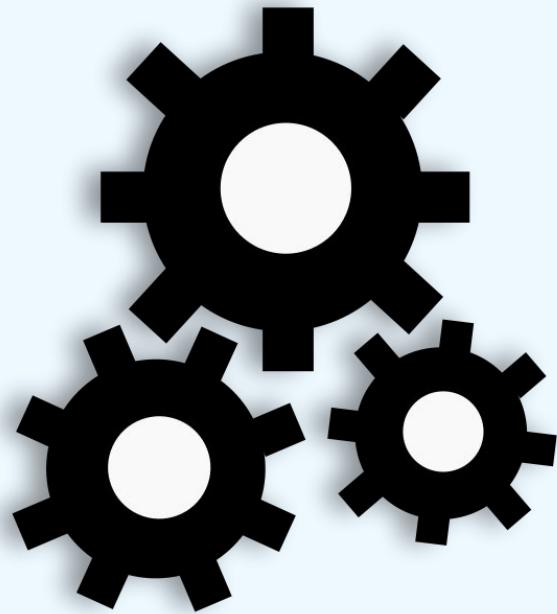
Help run check

Recall: Recalling language constructs of the DSL

Individualized practice



Future Work



Future Work

Questions?

Question 1 of 5

Objective	Diagram - Click 'Run'
Show two linearly dependent vectors	
1 Autolabel all 2 VectorSpace U 3 In(A, U) 4 In(B, U) 5 6 Vector A 7 Vector B 8 9	<p>Add linear dependency, type "LinearlyDependent(A, B)"</p>

Run **Check**

Question 1 of 5

Objective	Diagram - Click 'Run'
Show two linearly dependent vectors	
1 Autolabel all 2 VectorSpace U 3 In(A, U) 4 In(B, U) 5 6 Vector A 7 Vector B 8 LinearlyDependent(A, B) 9	<p>Click 'Run'</p>

Run **Check**

Question 1 of 5

Select diagrams that correspond to the following statement
 \vec{a} and \vec{b} are linearly independent

1 <input type="checkbox"/>	2 <input type="checkbox"/>
3 <input type="checkbox"/>	4 <input type="checkbox"/>

check

Question 1 of 5

Show two linearly dependent vectors, \vec{u}_1 and \vec{u}_2	Diagram - Click 'Run'
1 VectorSpace U 2 Vector $u_1 \in U$ 3 Label $u_1 \$\backslash vec\{u_{\{1\}}\$$	<p>editing Untitled</p>

Help **run** **check**

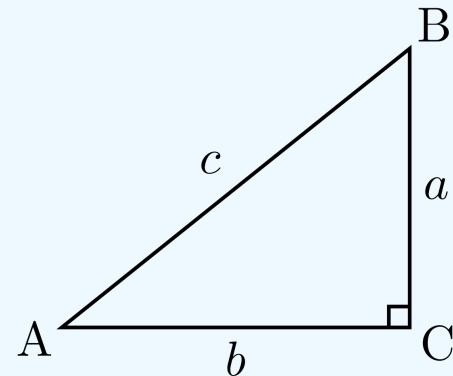


Defining Visual Mathematical Narratives Declaratively

Max Krieger, Wode Ni, Joshua Sunshine

Let's prove

$$a^2 + b^2 = c^2$$



(Pythagorean Theorem)



A textual proof?



Decomposability

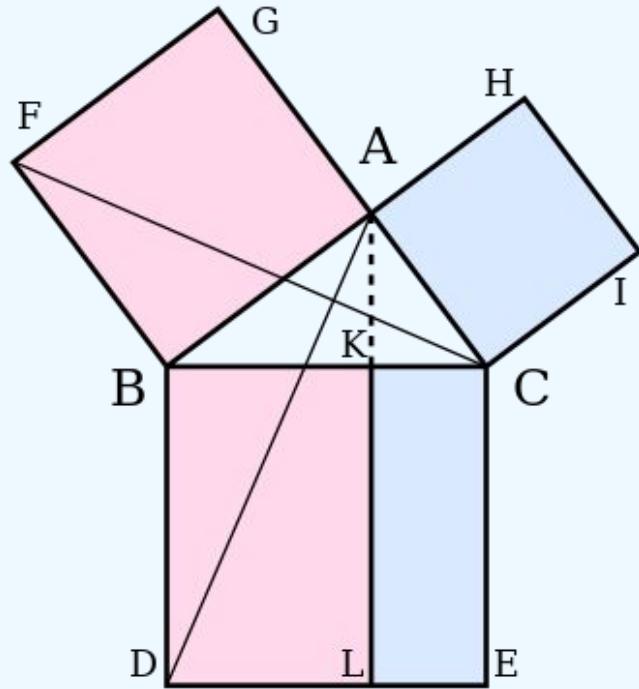


Structure

The proof is as follows:

1. Let ACB be a right-angled triangle with right angle CAB .
2. On each of the sides BC , AB , and CA , squares are drawn, $CBDE$, $BAGF$, and $ACIH$, in that order. The construction of squares requires the immediately preceding theorems in Euclid, and depends upon the parallel postulate.^[9]
3. From A , draw a line parallel to BD and CE . It will perpendicularly intersect BC and DE at K and L , respectively.
4. Join CF and AD , to form the triangles BCF and BDA .
5. Angles CAB and BAG are both right angles; therefore C , A , and G are collinear. Similarly for B , A , and H .
6. Angles CBD and FBA are both right angles; therefore angle ABD equals angle FBC , since both are the sum of a right angle and angle ABC .
7. Since AB is equal to FB and BD is equal to BC , triangle ABD must be congruent to triangle FBC .
8. Since $A-K-L$ is a straight line, parallel to BD , then rectangle $BDLK$ has twice the area of triangle ABD because they share the base BD and have the same altitude BK , i.e., a line normal to their common base, connecting the parallel lines BD and AL . (lemma 2)
9. Since C is collinear with A and G , square $BAGF$ must be twice in area to triangle FBC .
10. Therefore, rectangle $BDLK$ must have the same area as square $BAGF$ = AB^2 .
11. Similarly, it can be shown that rectangle $CKLE$ must have the same area as square $ACIH$ = AC^2 .
12. Adding these two results, $AB^2 + AC^2 = BD \times BK + KL \times KC$
13. Since $BD = KL$, $BD \times BK + KL \times KC = BD(BK + KC) = BD \times BC$
14. Therefore, $AB^2 + AC^2 = BC^2$, since $CBDE$ is a square.

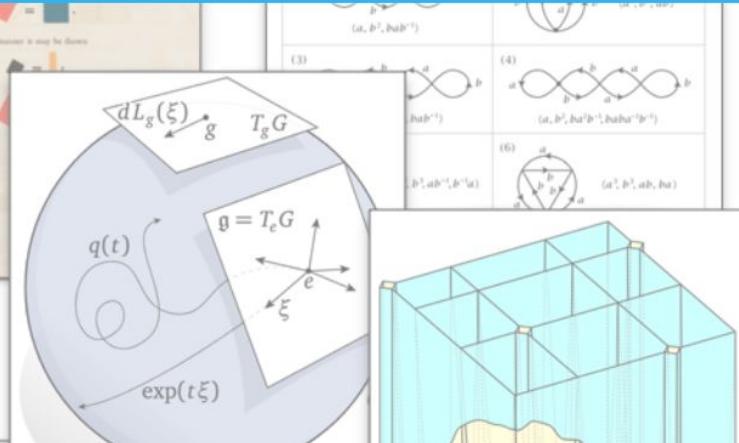
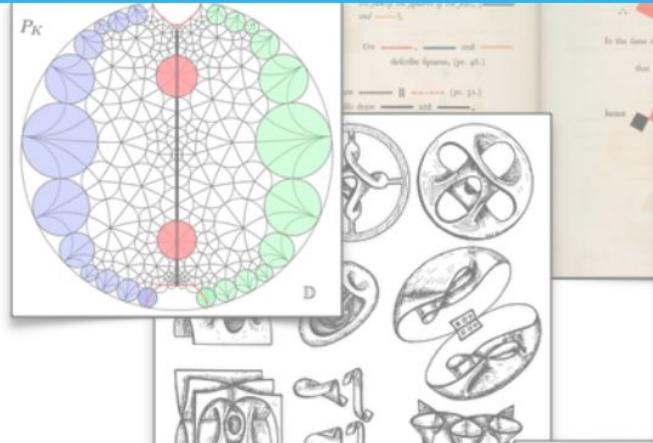
A textual proof?



A visual proof?

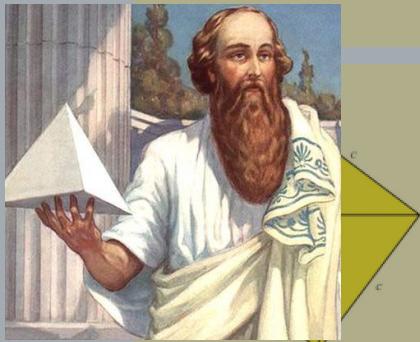
G

And that's not the whole story **or** picture!

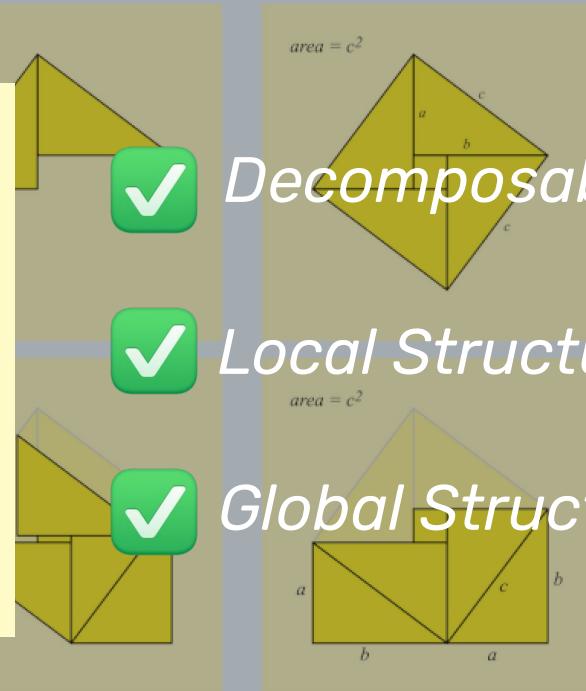
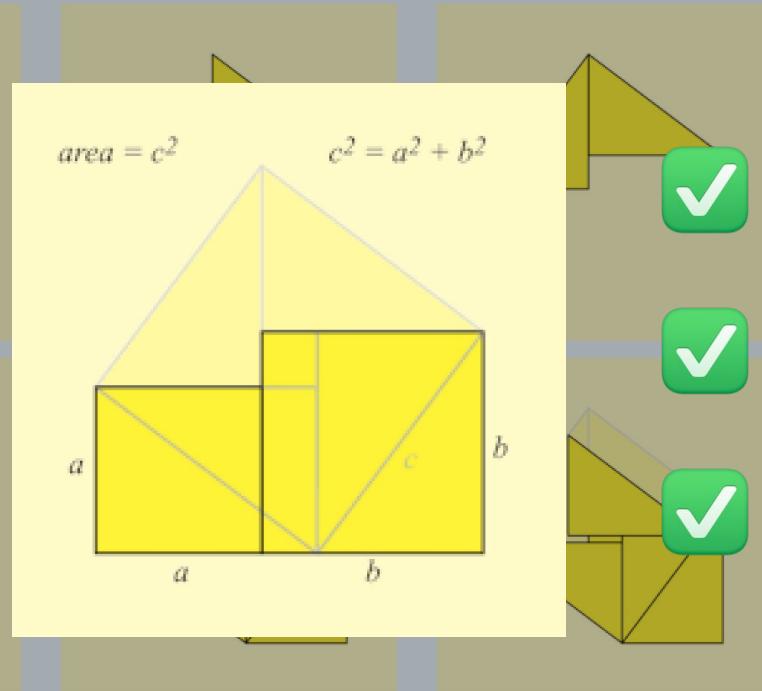


D L E

A visual proof?



“QED, I GUESS”
~ Mr. Pythagoras



A visual narrative

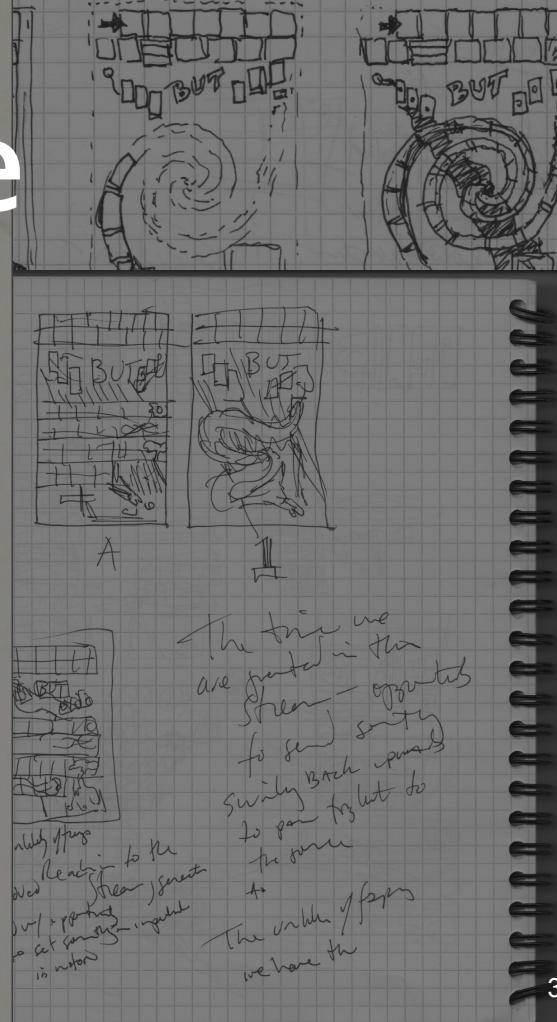
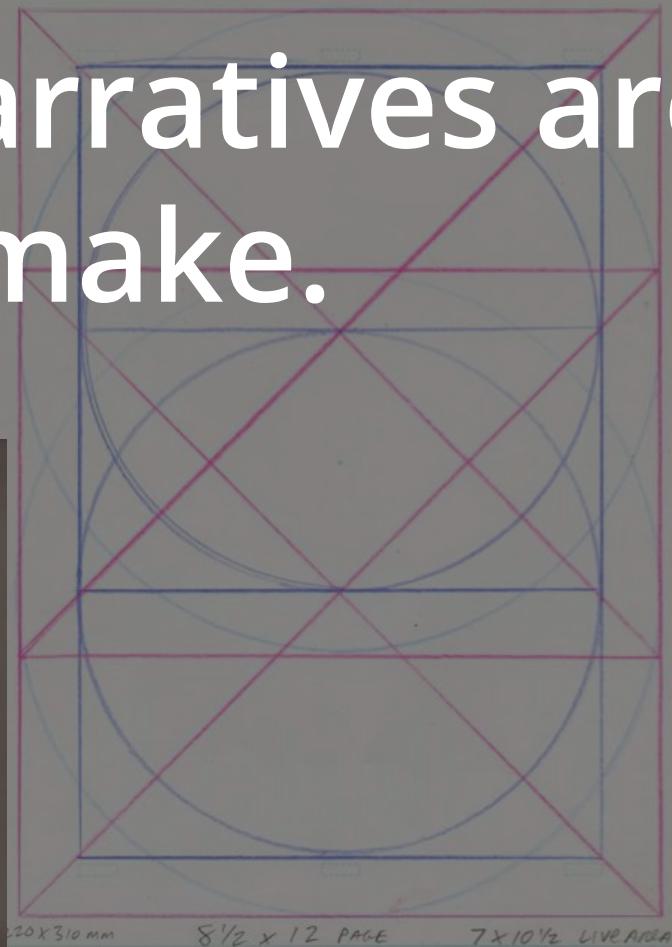
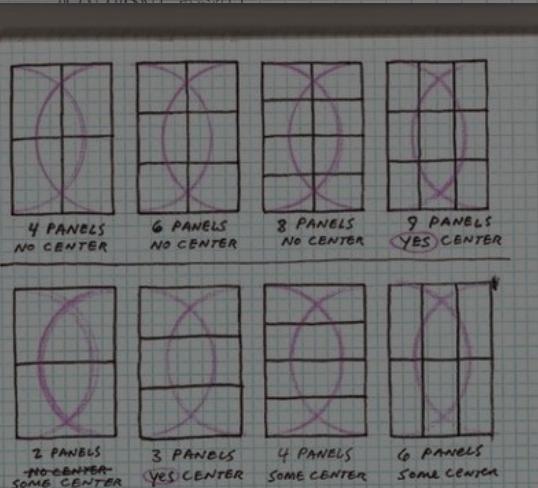
Visual Narrative

Noun. A story told through pictures.

aka Comics, Manga, Cave Paintings...



Visual narratives are hard to make.



We've talked about creating diagrams declaratively with *Penrose*

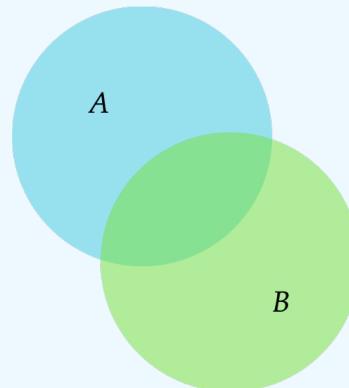
Could we also make diagrammatic
narratives?*



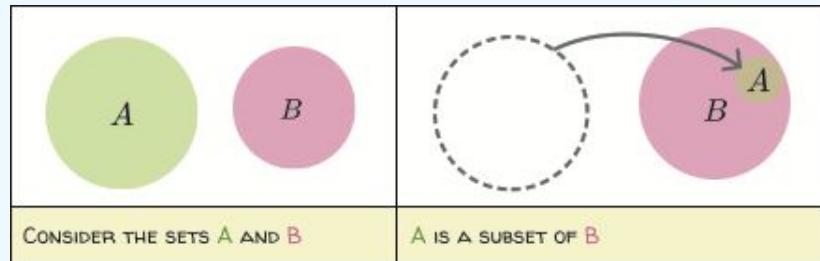
*We haven't implemented any of this yet. *This is just inquiry!*

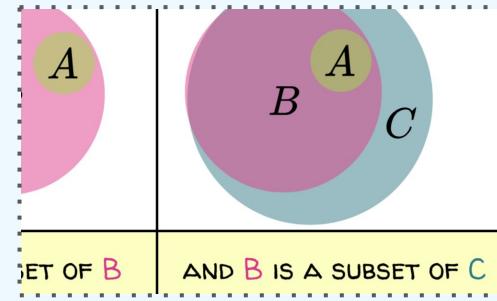
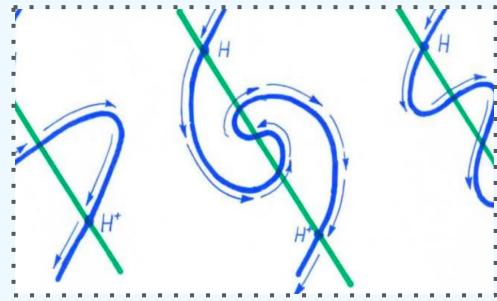
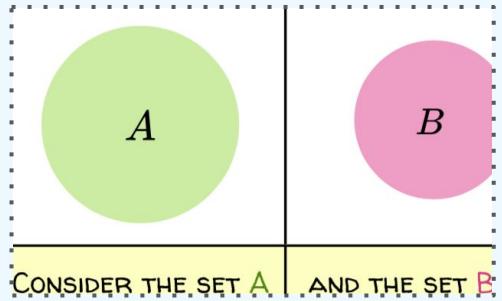
What does the language look like?

Set A, B
 $A \cap B$



Set A, B
 $A \cap B$

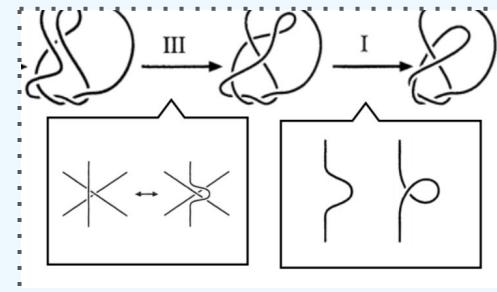
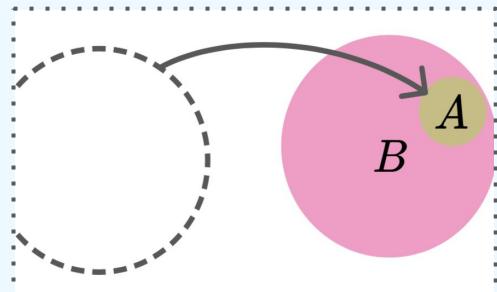
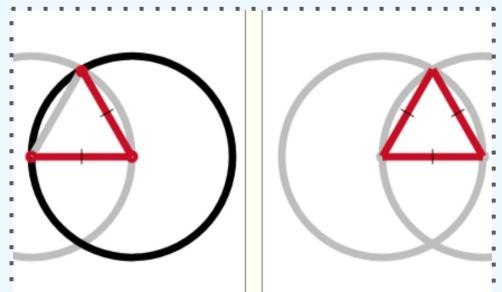




Conjunction

Multiples

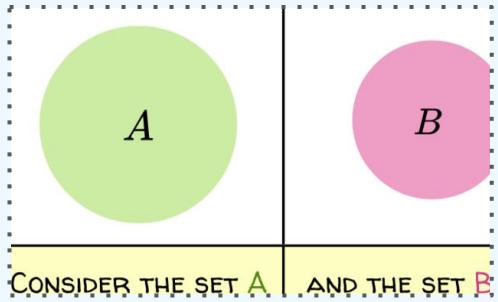
Predicates



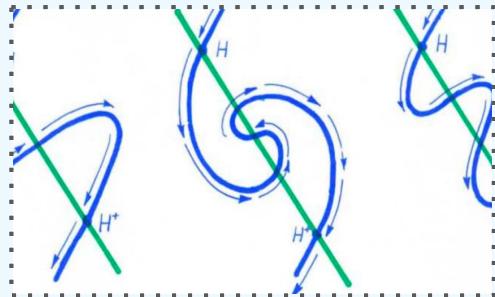
Foci

Emanata

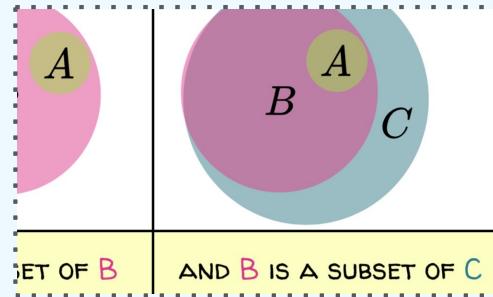
Nesting



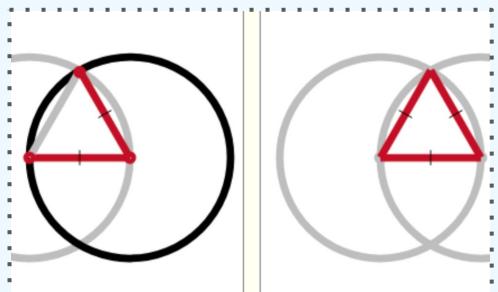
Conjunction



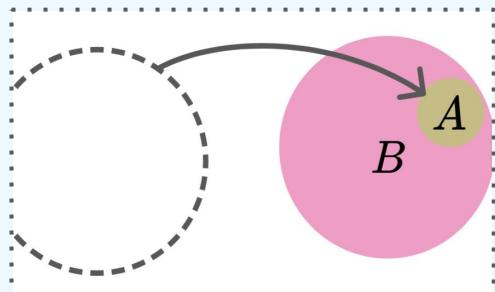
Multiples



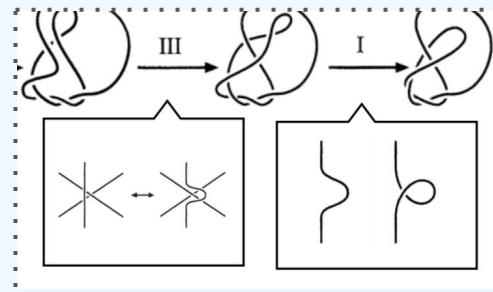
Predicates



Foci

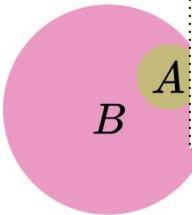
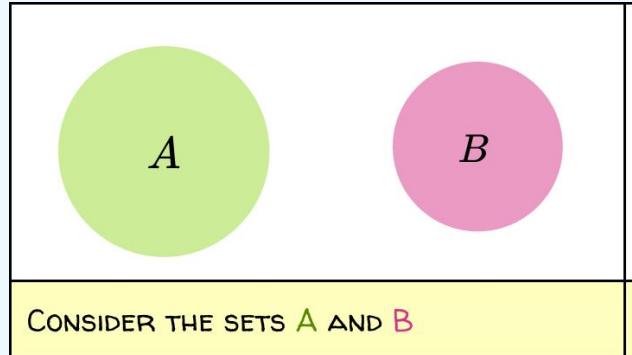


Emanata

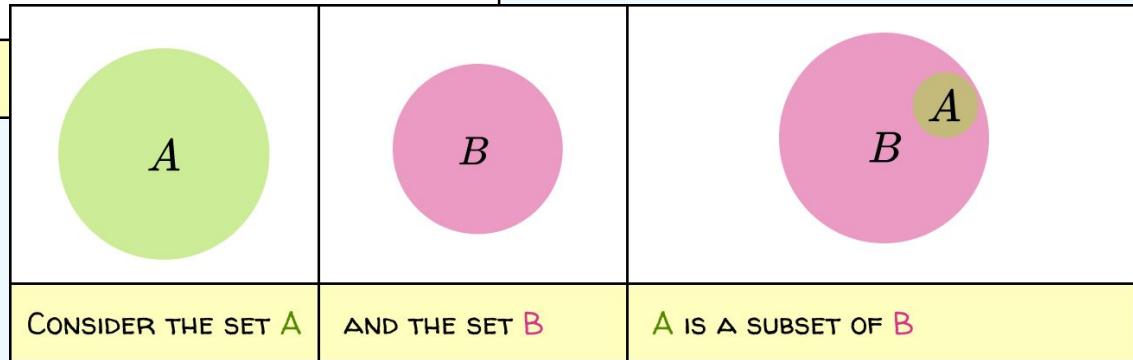


Nesting

Panel ([A, B],
"Consider the sets [A] and [B]")

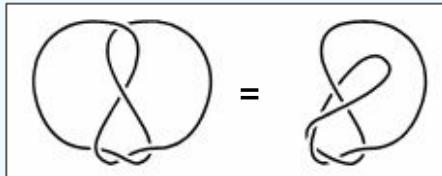


Panel ([A], "Consider the set [A]")
Panel ([B], "and the set [B]")



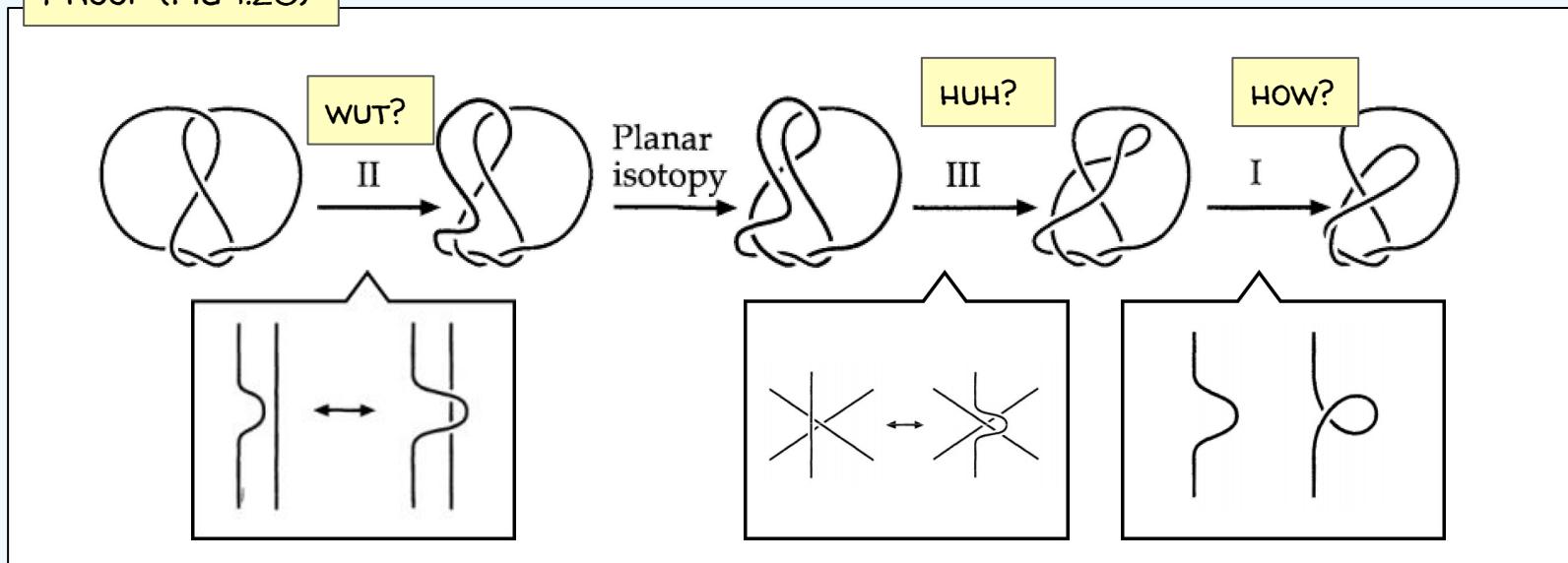
Conjunction

FIG 1.25, THE KNOT BOOK



NestPanels (p1, [(thm1, t1),
(thm2, t2), (lemma1, t3)])

PROOF (FIG 1.26)

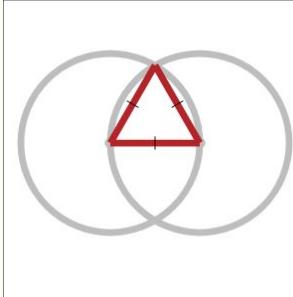
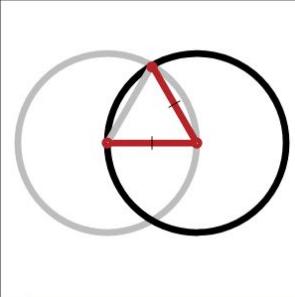
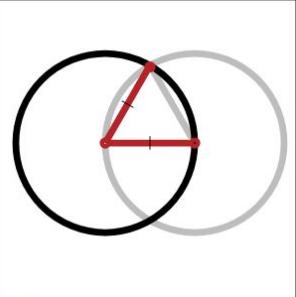


Nesting

These lines are both radius of the same circle and so equal.

These lines are both radius of the other circle and so equal too.

So we have a \triangle triangle with all lines equal to one another. Boom.



1.4

1.5

◦ Circle definition.

1.6

Equal to the same, equal to each other.

ELI PARRA'S ELEMENTS

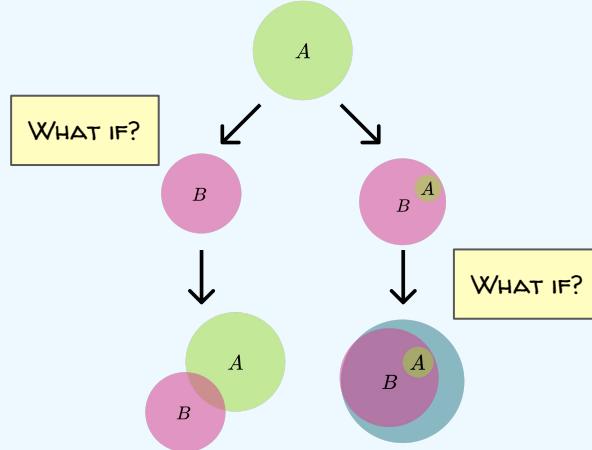
Circle C,D
{ Focus(C)
Panel ([C,D]) }

SuppletPanel ({ IsSubset(A,B) },
[A,B], "[A] is a subset of [B]")

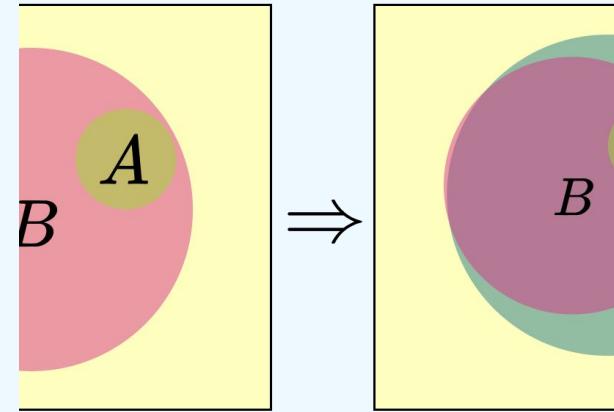
CONSIDER THE SETS A AND B	A IS A SUBSET OF B

Foci & Emanata

PARALLEL COUNTERFACTUALS



GUTTER AS IMPLICATION



FUTURE WORK: SAMPLED LAYOUT, INTERACTIVITY

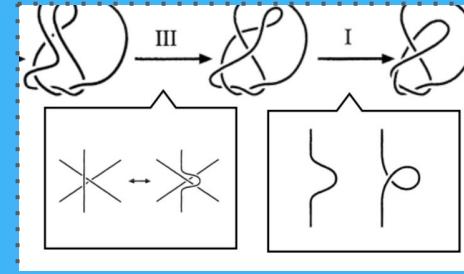
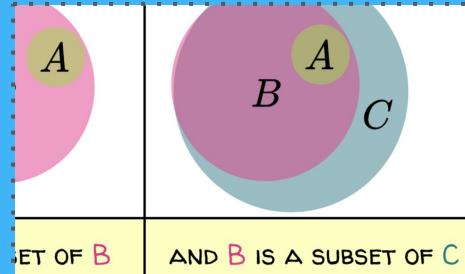
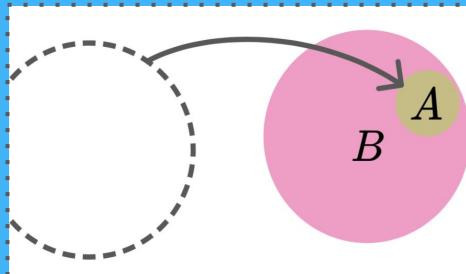
What emerges?

Thank you!

Acknowledgements

Co-Authors **Nimo Ni** and **Joshua Sunshine**

References via **Katherine Ye**, **Neil Cohn**, **Eli Parra**



QUESTIONS?

