

Formalización de las matemáticas con Lean. Un caso de estudio: Geometría euclídea plana.

Facultad de Ciencias Matemáticas.
Trabajo dirigido por Jorge Carmona Ruber.

Adrián Lattes Grassi

18 de septiembre de 2023

Proceso

- Aprendizaje de Lean

Proceso

- Aprendizaje de Lean
- Estudio y formalización de la geometría de Hilbert

Proceso

- Aprendizaje de Lean
- Estudio y formalización de la geometría de Hilbert
- Lectura de trabajos relacionados

Proceso

- Aprendizaje de Lean
- Estudio y formalización de la geometría de Hilbert
- Lectura de trabajos relacionados
- Independencia del axioma de las paralelas

Formalización asistida por computadores

- Digitalización de definiciones y enunciados

Formalización asistida por computadores

- Digitalización de definiciones y enunciados
- Comprobación mecanizada de demostraciones

Formalización asistida por computadores

- Digitalización de definiciones y enunciados
- Comprobación mecanizada de demostraciones
- Uso en docencia

Formalización asistida por computadores

- Digitalización de definiciones y enunciados
- Comprobación mecanizada de demostraciones
- Uso en docencia
- Demostración automatizada

Formalizando matemáticas en Lean

- Lean implementa el *Cálculo de construcciones inductivas*

Formalizando matemáticas en Lean

- Lean implementa el *Cálculo de construcciones inductivas*
- Correspondencia de Curry-Howard

Formalizando matemáticas en Lean

- Lean implementa el *Cálculo de construcciones inductivas*
- Correspondencia de Curry-Howard
- *Proposiciones como tipos*

P : Prop

Formalizando matemáticas en Lean

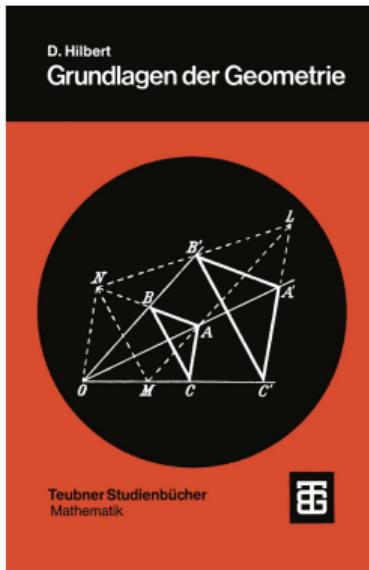
- Lean implementa el *Cálculo de construcciones inductivas*
- Correspondencia de Curry-Howard
- *Proposiciones como tipos*
 $P : \text{Prop}$
- *Demostraciones como términos*
 $p : P : \text{Prop}$

Formalizando matemáticas en Lean

- Lean implementa el *Cálculo de construcciones inductivas*
- Correspondencia de Curry-Howard
- *Proposiciones como tipos*
 $P : \text{Prop}$
- *Demostraciones como términos*
 $p : P : \text{Prop}$
- Modo táctico

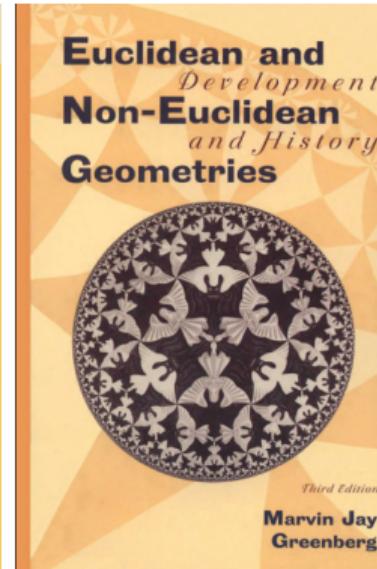
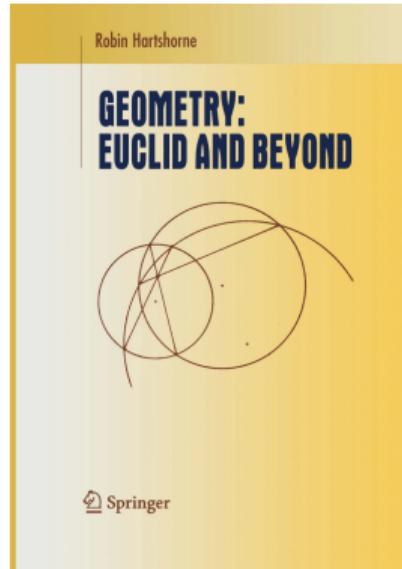
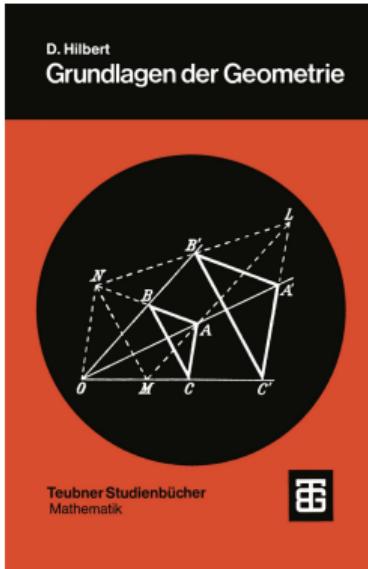
Geometría euclídea plana

Axiomatización de Hilbert



Geometría euclídea plana

Axiomatización de Hilbert



Geometría euclídea plana

Axiomatización de Hilbert

- *Puntos.* A, B, C, \dots
- *Líneas.* l, m, n, \dots
- *incidencia.* $A \sim l$
- *orden.* $A * B * C$
- *congruencia de segmentos.* $\overline{AB} \cong \overline{CD}$
- *congruencia de ángulos.* $\angle ABC \cong \angle CDE$

Geometría de incidencia

```
class incidence_geometry (Point Line : Type*) :=  
(lies_on : Point → Line → Prop)  
(infix ` ~ ` : 50 := lies_on)
```

Geometría de incidencia

```
class incidence_geometry (Point Line : Type*) :=
(lies_on : Point → Line → Prop)
(infix ` ~ ` : 50 := lies_on)
(I1 {A B : Point} (h : A ≠ B) : ∃! l : Line, A ~ l ∧ B ~ l)
```

Geometría de incidencia

```
class incidence_geometry (Point Line : Type*) :=
(lies_on : Point → Line → Prop)
(infix ` ~ ` : 50 := lies_on)
(I1 {A B : Point} (h : A ≠ B) : ∃! l : Line, A ~ l ∧ B ~ l)
(I2 (l : Line) : ∃ A B : Point, A ≠ B ∧ A ~ l ∧ B ~ l)
```

Geometría de incidencia

```
class incidence_geometry (Point Line : Type*) :=
(lies_on : Point → Line → Prop)
(infix ` ~ ` : 50 := lies_on)
(I1 {A B : Point} (h : A ≠ B) : ∃! l : Line, A ~ l ∧ B ~ l)
(I2 (l : Line) : ∃ A B : Point, A ≠ B ∧ A ~ l ∧ B ~ l)
(I3 : ∃ A B C : Point, neq3 A B C ∧ ¬ ∃ l : Line, A ~ l ∧ B ~ l ∧ C ~ l)
```

Geometría de incidencia

Proposición

Dos líneas distintas pueden tener como mucho un punto en común.

Geometría de incidencia

Proposición

Dos líneas distintas pueden tener como mucho un punto en común.

```
def is_common_point
{Point Line : Type*} [incidence_geometry Point Line]
(A : Point) (l m : Line) :=
A ~ l ∧ A ~ m

def have_common_point
(Point : Type*) {Line : Type*} [incidence_geometry Point Line]
(l m : Line) :=
∃ A : Point, is_common_point A l m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
(Point : Type*) {Line : Type*}
[ig : incidence_geometry Point Line] :
 $\forall l m : \text{Line}, l \neq m \rightarrow$ 
 $(\exists! A : \text{Point}, \text{is_common_point } A l m)$ 
 $\vee \neg \text{have_common_point Point } l m :=$ 
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=  
begin
  intro hneq,
  intro hneq,
  intro hneq,
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (∃! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m
```

```
Point Line : Type u
ig : incidence_geometry Point Line
  ⊢ ∀ (l m : Line), l ≠ m →
  (∃! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
```

begin

```
intros l m,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
```

```
      ⊢ l ≠ m →
      (exists! A : Point, is_common_point A l m)
      ∨ ¬ have_common_point Point l m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (∃! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line

⊢ l ≠ m →
  (∃! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line

⊢ ¬((exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m) →
  l ≠ m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line

      ⊢ ¬((exists! A : Point, is_common_point A l m)
           ∨ ¬ have_common_point Point l m) →
           l ≠ m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
      ⊢ (¬exists! A : Point, is_common_point A l m)
      ∧ have_common_point Point l m →
      l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
  ⊢ (¬exists! A : Point, is_common_point A l m)
    ∧ have_common_point Point l m →
      l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
not_unique : ¬exists! A : Point, is_common_point A l m
hlm : have_common_point Point l m
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (∃! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ¬∃! A : Point, is_common_point A l m
hlm: have_common_point Point l m
          ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ¬∃ A : Point,
             is_common_point A l m
             ∧ ∀ B : Point, is_common_point B l m → B = A
hlm: have_common_point Point l m

⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ¬∃ A : Point,
             is_common_point A l m
             ∧ ∀ B : Point, is_common_point B l m → B = A
hlm: have_common_point Point l m

⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ∀ A : Point, is_common_point A l m →
            (exists B : Point, is_common_point B l m ∧ B ≠ A)
hlm: have_common_point Point l m
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
    (exists! A : Point, is_common_point A l m)
    ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ∀ A : Point, is_common_point A l m →
  (exists B : Point, is_common_point B l m ∧ B ≠ A)
hlm: have_common_point Point l m

⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
    (exists! A : Point, is_common_point A l m)
    ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ∀ A : Point, is_common_point A l m →
  (exists! B : Point, is_common_point B l m ∧ B ≠ A)
A: Point
hA: is_common_point A l m
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
not_unique: ∀ A : Point, is_common_point A l m →
  (exists! B : Point, is_common_point B l m ∧ B ≠ A)
A: Point
hA: is_common_point A l m
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
A B : Point
hA : is_common_point A l m
hB : is_common_point B l m
hAB : B ≠ A
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩,
  rw ne_comm at hAB
```

```
Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
A B: Point
hA: is_common_point A l m
hB: is_common_point B l m
hAB: B ≠ A
      ⊢ l = m
```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩,
  rw ne_comm at hAB,
```

```
Point Line : Type u
ig : incidence_geometry Point Line
l m : Line
A B : Point
hA : is_common_point A l m
hB : is_common_point B l m
hAB : A ≠ B
      ⊢ l = m
```

Geometría de incidencia

```

lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) [Line : Type*]
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
    (exists! A : Point, is_common_point A l m)
    ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩,
  rw ne_comm at hAB,
  exact unique_of_exists_unique (ig.I1 hAB) ⟨hA.1, hB.1⟩ ⟨hA.2, hB.2⟩

```

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point
  (Point : Type*) [Line : Type*]
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
  (exists! A : Point, is_common_point A l m)
  ∨ ¬ have_common_point Point l m :=

begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩,
  rw ne_comm at hAB,
  exact unique_of_exists_unique (ig.I1 hAB) ⟨hA.1, hB.1⟩ ⟨hA.2, hB.2⟩

ig.I1 hAB : ∃! l : Line, A ~ l ∧ B ~ l
```

Point Line: Type u
ig: incidence_geometry Point Line
l m : Line
A B: Point
hA: is_common_point A l m
hB: is_common_point B l m
hAB: A ≠ B
 |- l = m

Geometría de incidencia

```
lemma neq_lines_have_at_most_one_common_point          goals accomplished ✓
  (Point : Type*) {Line : Type*}
  [ig : incidence_geometry Point Line] :
  ∀ l m : Line, l ≠ m →
    (exists! A : Point, is_common_point A l m)
    ∨ ¬ have_common_point Point l m :=
begin
  intros l m,
  contrapose,
  push_neg,
  rintro ⟨not_unique, hlm⟩,
  rw exists_unique at not_unique,
  push_neg at not_unique,
  cases hlm with A hA,
  rcases not_unique A hA with ⟨B, ⟨hB, hAB⟩⟩,
  rw ne_comm at hAB,
  exact unique_of_exists_unique (ig.I1 hAB) ⟨hA.1, hB.1⟩ ⟨hA.2, hB.2⟩
end
```

Geometría del orden

Definición

Dados dos puntos distintos A, B el **segmento** \overline{AB} es el *conjunto de puntos* que contiene a A, B y a todos los puntos que están entre ellos.

Geometría del orden

Definición

Dos puntos distintos A, B determinan el **segmento** \overline{AB} .

```
structure Seg (Point : Type*) := {A B : Point} (neq : A ≠ B)
```

Geometría del orden

Definición

Dos puntos distintos A, B determinan el **segmento** \overline{AB} .

```
structure Seg (Point : Type*) := {A B : Point} (neq : A ≠ B)
```

Definición

Un punto C **pertenece** al segmento \overline{AB} si coincide con A o B o está entre ellos ($A * C * B$).

```
def Seg.in (seg : Seg Point) (Line : Type*)
  [og : order_geometry Point Line] (P : Point) : Prop :=
  P = seg.A ∨ P = seg.B ∨ (og.between seg.A P seg.B)
```

Geometría del orden

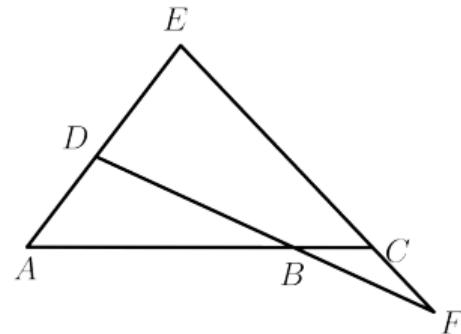
Teorema

*Dados dos puntos distintos A y C existe un tercer punto B que se encuentra entre ellos:
A * B * C.*

Geometría del orden

Teorema

*Dados dos puntos distintos A y C existe un tercer punto B que se encuentra entre ellos: $A * B * C$.*



*Formalizing Hilbert's Grundlagen in Isabelle/Isar,
Laura I. Meikle and Jacques D. Fleuriot.*

Independencia del axioma de las paralelas

```
def parallel (Point : Type*) {Line : Type*} [ig : incidence_geometry Point Line]
(l m : Line) : Prop :=  $\neg \exists P : Point, is\_common\_point P l m$ 
```

Independencia del axioma de las paralelas

```
def parallel (Point : Type*) {Line : Type*} [ig : incidence_geometry Point Line]
(l m : Line) : Prop :=  $\neg \exists P : Point, is\_common\_point P l m$ 

def P (Point Line : Type*) [ig : incidence_geometry Point Line] :=
 $\forall (l : Line) (A : Point), \neg A \sim l \rightarrow \exists! m : Line, A \sim m \wedge parallel Point l m$ 
```

Independencia del axioma de las paralelas

```
def parallel (Point : Type*) {Line : Type*} [ig : incidence_geometry Point Line]
(l m : Line) : Prop :=  $\neg \exists P : Point, is\_common\_point P l m$ 

def P (Point Line : Type*) [ig : incidence_geometry Point Line] :=
 $\forall (l : Line) (A : Point), \neg A \sim l \rightarrow \exists! m : Line, A \sim m \wedge parallel Point l m$ 

theorem parallels_independence
(Point Line : Type*) :
 $\neg \forall plane : hilbert\_plane Point Line,$ 
 $P Point Line :=$ 
begin
  sorry
end
```

Independencia del axioma de las paralelas

```
def parallel (Point : Type*) {Line : Type*} [ig : incidence_geometry Point Line]
(l m : Line) : Prop :=  $\neg \exists P : Point, is\_common\_point P l m$ 

def P (Point Line : Type*) [ig : incidence_geometry Point Line] :=
 $\forall (l : Line) (A : Point), \neg A \sim l \rightarrow \exists! m : Line, A \sim m \wedge parallel Point l m$ 

theorem parallels_independence
(Point Line : Type*) :
 $\neg \forall plane : hilbert\_plane Point Line,$ 
 $P Point Line :=$ 
begin
  sorry
end
```

Point: Type u_1
Line: Type u_2
 $\vdash \neg \forall plane : hilbert_plane Point Line,$
 $P Point Line$

Independencia del axioma de las paralelas

```
def parallel (Point : Type*) {Line : Type*} [ig : incidence_geometry Point Line]
(l m : Line) : Prop :=  $\neg \exists P : Point, is\_common\_point P l m$ 

def P (Point Line : Type*) [ig : incidence_geometry Point Line] :=
 $\forall (l : Line) (A : Point), \neg A \sim l \rightarrow \exists! m : Line, A \sim m \wedge parallel Point l m$ 

theorem parallels_independence
(Point Line : Type*) :
 $\neg \forall plane : hilbert\_plane Point Line,$ 
 $P Point Line :=$ 
begin
  push_neg,
  sorry
end
```

Point: Type u_1
Line: Type u_2
 $\vdash \exists plane : hilbert_plane Point Line,$
 $\neg P Point Line$

Conclusiones

- Cumplimiento del proyecto de formalización de Hilbert.

Conclusiones

- Cumplimiento del proyecto de formalización de Hilbert.
- Lean como herramienta de estudio.

Conclusiones

- Cumplimiento del proyecto de formalización de Hilbert.
- Lean como herramienta de estudio.
- Dificultades con *mathlib*, en particular al intentar definir modelos.

Conclusiones

- Cumplimiento del proyecto de formalización de Hilbert.
- Lean como herramienta de estudio.
- Dificultades con *mathlib*, en particular al intentar definir modelos.
- Queda mucha geometría por formalizar en Lean.

¡Gracias!



Detalle de *La escuela de Atenas*, de Rafael Sanzio, donde vemos a Euclides enseñando a sus alumnos.