Elementos de matemáticas formalizados en Isabelle/HOL

Facultad de Matemáticas
Departamento de Ciencias de la Computación e Inteligencia Artificial
Trabajo Fin de Grado

Carlos Núñez Fernández

El presente Trabajo Fin de Grado se ha realizado en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Sevilla y ha sido supervisado por José Antonio Alonso Jiménez y María José Hidalgo Doblado.

Índice

Sumario

La finalidad de este trabajo es la formalización de teoremas de diferentes teorías de las matemáticas. Para ello, se han elegido una serie de teoremas de la teoría de números, teoría de conjuntos, teoría de funciones, teoría de retículos y teoría de geometría. Una vez formalizados los mismos en el sistema de pruebas automáticas Isabelle/HOL, se han utilizado teorías ya predefinidas en él y comprobado la similitud existente entre la demostración en lenguaje natural y la demostración formal.

The purpose of this project is the formalization of theorems from differents mathematic's theories. For it, some theorems have been choosen from number theory, set theory, function theory, lattice theory and geometry theory. Once they have been formalized by the automatic proof system Isabelle/HOl, theories have already been used predefined in it and checking the existing similarity between proof in natural language and formal proof.

Introducción

Actualmente el gran interés de la formalización matemática reside en la capacidad de la verificación de demostraciones mediante un ordenador. Para ello, se hace preciso poder expresar las definiciones, teoremas y pruebas en un lenguaje generado por una gramática que permita verificar de forma mecánica las pruebas. También hay que dotar al ordenador de una información previa al teorema de forma que junto con una orientación humana, se pueda llegar a validar la demostración de los teoremas. La importancia que se le atribuye a la formalización es la capacidad de cálculos y razonamientos que puede realizar un ordenador a la vez, incluso demostrando teoremas muy difíciles como el teorema de los Knaster–Tarski como se expondrá en el Capítulo 5. En [6] pueden encontrarse una lista de 100 teoremas formalizados junto con el programa usado.

Dentro de los sistemas de pruebas automáticas los más usados, como se aprecia en [7] son HOL, Mizar, PVS, Coq, Isabelle/Isar entre otros. Isabelle suministra una estructura genérica para construir sistemas deductivos con un especial foco en la prueba de teoremas de lógica de orden superior. Sin embargo, Isar proporciona un entorno de lenguaje propio, diseñado específicamente para el desarrollo de pruebas y teorias. En su conjunto Isabelle/Isar es un marco de referencia para el desarrollo formal de documentos matemáticos, incluida una comprobación completa de pruebas, en el que las definiciones y pruebas se organizan como teorias. En el presente análisis, el sistema de pruebas automático utilizado es Isabelle/HOL que es una especialización de Isabelle/Isar con lógica de orden superior(HOL).

El objetivo general de este trabajo es mostrar como se elaboran demostraciones formales y estructuradas en Isabelle/HOL, manifestando la capacidad que tiene este sistema de pruebas automáticas en los diferentes aspectos de las matemáticas.

El objetivo específico es estudiar la similitud que hallada entre las demostraciones en lenguaje natural y las de lenguaje formal en Isabelle/HOL de aspectos básicos de las diferentes teorías de las matemáticas.

La metodología utilizada para este trabajo ha sido seleccionar y probar formalmente teoremas en las distintas áreas de las matemáticas, de forma que se muestre la capacidad de Isabelle/HOl en los diferentes ámbitos. Una vez escogido el teorema para

formalizar se ha seguido siempre el mismo esquema:

1. Enunciado del teorema en lenguaje natural.
2. Demostración del teorema en lenguaje natural.
3. Especificación en Isabelle/HOL.
4. Demostración en Isabelle/HOL que visiblemente muestra la similitud con la prueba en lenguaje natural.

Tanto para la demostración formal como para su anterior especificación ha sido necesario usar diferentes teorías ya predefinidas en Isabelle/HOl, e incluso, en algunos casos introducir una serie de lemas y definiciones para poder llegar a entender y demostrar formalmente el teorema.

La descripción de los capítulos es la siguiente.

Dentro de la gran cantidad de teorías matemáticas, se han elegido cinco de ellas con las que trataremos.

En el capítulo 1 se muestran dos teoremas de la teoría de números. El primero de ellos es un resultado clásico sobre números impares. La decisión de escoger este teorema se debe a su demostración, ya que se realiza por inducción sobre los números naturales, pudiendo mostrar el esquema inductivo predefinido en Isabelle y observar la gran similitud existente entre la demostración en lenguaje natural y formal. También se manifiesta la capacidad de Isabelle/HOL dando pruebas menos explícitas e incluso automáticas. El segundo teorema es una propiedad sobre los conjuntos finitos de números naturales que, al igual que en el anterior, se demuestra de manera inductiva; pero esta vez una inducción sobre conjuntos finitos, mostrando también el esquema ya predefinido en Isabelle/HOL de inducción sobre conjuntos finitos.

En el capítulo 2 se analizan dos resultados de la teoría de funciones. Estos son una caracterización sobre funciones inyectivas y sobreyectivas respectivamente. La importancía de estos teoremas es exponer la capacidad de Isabelle/HOL para trabajar con tipos; es decir, con los dominios y codominios de las funciones. En la demostración de ambos teoremas se hace preciso especificar los tipos tanto de las funciones como de los elementos, debido a que, en el caso de no especificarlo, Isabelle/HOL toma el tipo más general posible y no admite la prueba. Se ha de descatar la necesidad de usar definiciones de la teoría ya predefinida Fun.thy e introducir unas nuevas para conseguir la similitud con la especificación y demostración formal.

En el capítulo 3 se muestra el teorema de Cantor, un teorema importante de la teoría de conjuntos. La elección del mismo se debe a su demostración formal y automática, ya que está es idéntica a la demostración en lenguaje natural y, en cuanto a la automática, cabe destacar la capacidad de Isabelle/HOL para realizarlo automáticamente.

En el capítulo 4 se estudia el teorema del punto fijo de Knaster–Tarski un resultado de la teoría de retículos. La elección de este surge por el interés en la teoría de retículos, ya que es una teoría importada en Isabelle/HOL y para su plena comprensión tiene una notación específica. Además subrayar la peculiaridad de los retículos y de los retículos completos que se definen en Isabelle/HOL como clases.

En el capítulo 5, se formaliza la teoría de la geometría, en la que diferenciamos tres tipos: geometría simple, geometría no proyectiva y geometría proyectiva. Cada tipo de geometría la declararemos en un entorno local, definiendo los axiomas propios de ella y, dentro de él, demostrando una serie de lemas y dando, por último, una interpretación del mínimo modelo con que se puede formar verificando los axiomas.

En el apéndice A, basándonos en el libro de HOL se indican todas las reglas y lemas usados en el trabajo agrupadas en diferentes secciones.

Por último cabe destacar que el trabajo realizado se encuentra en GitHub

Capítulo 1

Teoría de números

En este capítulo se van a analizar dos teoremas sobre la teoría de números. El primero es un resultado sobre los números impares y el segundo un resultado sobre conjuntos finitos, la importancia que se le ortoga a estos, es debido a, su demostración inductiva. El primero se demuestra por una inducción sobre números naturales y el segundo por una inducción sobre conjuntos finitos, pudiendo mostrar un esquema inductivo predefinido en Isabelle/HOL de ambas demostraciones. También se mostrará la capacidad de Isabelle/HOL dando pruebas explícitas y automáticas.

1.1 Suma de los primeros números impares

1.1.1 Demostración en lenguaje natural

El primer teorema es una propiedad de los números naturales.

Teorema 1.1.1 *La suma de los n primeros números impares es* n^2.

Demostración: La demostración la haremos por inducción sobre n.

(Base de la inducción) El caso $n = 0$ es trivial.

(Paso de la inducción) Supongamos que la propiedad se verifica para n y veamos que también se verifica para $n + 1$.

Tenemos que demostrar que $\sum_{j=1}^{n+1} k_j = (n+1)^2$ donde k_j el j–ésimo impar; es decir, $k_j = 2j - 1$.

$$\begin{aligned}
&\sum_{j=1}^{n+1} k_j \\
&= k_{n+1} + \sum_{j=1}^{n} k_j \\
&= k_{n+1} + n^2 \quad (\textit{Hipótesis inducción}) \\
&= 2(n+1) - 1 + n^2 \\
&= n^2 + 2n + 1 \\
&= (n+1)^2
\end{aligned}$$

□

1.1.2 Especificación en Isabelle/HOL

Para especificar el teorema en Isabelle, se comienza definiendo la función *suma-impares* tal que *suma-impares n* es la suma de los *n* primeros números impares

fun *suma-impares* :: *nat* ⇒ *nat* **where**
suma-impares 0 = 0
| *suma-impares* (*Suc n*) = (2∗(*Suc n*) − *1*) + *suma-impares n*

El enunciado del teorema es el siguiente:

lemma *suma-impares n* = *n* ∗ *n*
oops

En la demostración se usará la táctica *induct* que hace uso del esquema de inducción sobre los naturales:

$$\frac{P\ 0 \qquad \bigwedge nat.\ \dfrac{P\ nat}{P\ (Suc\ nat)}}{P\ nat} \qquad (nat.induct)$$

Vamos a presentar distintas demostraciones del teorema. La primera es la demostración automática.

1.1.3 Demostración automática

La correspondiente demostración automática es

lemma *suma-impares n* = *n* ∗ *n*
by (*induct n*) *simp-all*

1.1.4 Demostración estructurada

La demostración estructurada y detallada del lema anterior es:

```
lemma suma-impares n = n * n
proof (induct n)
 have suma-impares 0 = 0
  by (simp only: suma-impares.simps(1))
 also have ... = 0 * 0
  by (simp only:  mult-0)
 finally show suma-impares 0 = 0 * 0
  by (simp only: mult-0-right)
next
 fix n
 assume HI: suma-impares n = n * n
 have suma-impares (Suc n) = (2 * (Suc n) - 1) + suma-impares n
  by (simp only: suma-impares.simps(2))
 also have ... = (2 * (Suc n) - 1) + n * n
  by (simp only: HI)
 also have ... = n * n + 2 * n + 1
  by (simp only: mult-Suc-right)
 also have ... = (Suc n) * (Suc n)
  by (simp only: mult-Suc mult-Suc-right)
 finally show suma-impares (Suc n) = (Suc n) * (Suc n)
  by this
qed
```

En la demostración anterior se pueden ocultar detalles.

```
lemma suma-impares n = n * n
proof (induct n)
 show suma-impares 0 = 0 * 0 by simp
next
 fix n
 assume HI: suma-impares n = n * n
 have suma-impares (Suc n) = (2 * (Suc n) - 1) + suma-impares n
  by simp
 also have ... = (2 * (Suc n) - 1) + n * n
  using HI by simp
 also have ... = (Suc n) * (Suc n)
  by simp
 finally show suma-impares (Suc n) = (Suc n) * (Suc n)
```

```
  by simp
qed
```

1.1.5 Demostración con patrones

La demostración anterior se puede simplificar usando patrones.

```
lemma suma-impares n = n * n (is ?P n = ?Q n)
proof (induct n)
 show ?P 0 = ?Q 0 by simp
next
 fix n
 assume HI: ?P n = ?Q n
 have ?P (Suc n) = (2 * (Suc n) - 1) + suma-impares n
  by simp
 also have ... = (2 * (Suc n) - 1) + n * n using HI by simp
 also have ... = ?Q (Suc n) by simp
 finally show ?P (Suc n) = ?Q (Suc n) by simp
qed
```

La demostración usando otro patrón es

```
lemma suma-impares n = n * n (is ?P n)
proof (induct n)
 show ?P 0 by simp
next
 fix n
 assume ?P n
 then show ?P (Suc n) by simp
qed
```

1.2 Propiedad de los conjuntos finitos de número naturales

1.2.1 Demostración en lenguaje natural

El siguiente teorema es una propiedad que verifican todos los conjuntos finitos de números naturales. Se ha estudiado en el tema 10 de la asignatura de LMF de tercer curso del grado en Matemáticas. Su enunciado es el siguiente:

Teorema 1.2.1 *Sea S un conjunto finito de números naturales. Entonces todos los elementos de S son menores o iguales que la suma de los elementos de S; es decir,*

$$\forall m \in S \Longrightarrow m \leq \sum S$$

donde $\sum S$ denota la suma de todos los elementos de S.

Primero se debe notar que podemos dar una definición inductiva de conjunto finito, lo que conlleva un esquema de inducción asociado.

Definición 1.2.2 *La definición inductiva de un conjunto finito es:*

- *$\emptyset$ es finito.*
- *Si A es un conjunto finito y x un elemento entonces $A \cup \{x\}$ es un conjunto finito.*

De esta construcción se obtiene un esquema de inducción. Para ello sea φ una propiedad sobre conjuntos finitos. El esquema de inducción viene dado por:

Si se verifica:

1. $\varphi(\emptyset)$.
2. $\forall A$ finito tal que $\varphi(A)$ y $\forall x$ entonces $\varphi(A \cup \{x\})$.

Entonces $\forall A$ finito se verifica $\varphi(A)$.

Demostración: La demostración del teorema la haremos por inducción sobre conjuntos finitos.

(Base de la inducción) El caso $S = \emptyset$ es trivial.

(Paso de la inducción) Supongamos que se verifica el teorema para un conjunto finito de números naturales, que se denotará por S y sea a un elemento. Vamos a demostrarlo para $S \cup \{a\}$.

Sea $a \in \mathbb{N}$ tal que $a \notin S$, ya que si $a \in S$ se tendría probado el teorema. Luego hay que probar que:

$$\forall n \in S \cup \{a\} \Longrightarrow n \leq \sum(S \cup \{a\})$$

Distingamos dos casos ahora:

Caso 1: $n = a$.

Si $n = a$, se tiene que:

$$n = a \leq a + \sum S = \sum(S \cup \{a\}).$$

Caso 2: $n \neq a$.

Si $n \neq a$, tenemos que $n \in S$, luego usando la hipótesis de inducción:

$$n \leq \sum S \leq \sum S + a = \sum(S \cup \{a\}).$$

□

En la demostración del teorema hemos usado un resultado, que vamos a probar en Isabelle después de la especificación del teorema; el resultado es $\sum S + a = \sum(S \cup \{a\})$.

1.2.2 Especificación en Isabelle/HOL

Para la especificación del teorema en Isabelle, primero consideremos la definición de conjunto finito ya definida en Isabelle.

```
    inductive finite :: "'a set ⇒ bool"
where
emptyI [simp, intro!]: "finite {}"
| insertI [simp, intro!]: "finite A ⟹ finite (insert a A)"
```

Esta definición de conjunto finito es una definición inductiva en Isabelle, equivalente a la Definición 1.2.2 en lenguaje natural. Esta definición genera automáticamente el siguiente esquema de inducción en Isabelle:

$$[\![\mathit{finite}\ x;\ P\ \emptyset;\ \bigwedge A\ a.\ \mathit{finite}\ A \wedge P\ A \Longrightarrow P\ (\{a\} \cup A)]\!] \Longrightarrow P\ x \qquad (\mathit{finite.induct})$$

También se debe notar que *finite S* indica que un conjunto S es finito y definir la función *sumaConj* tal que *sumaConj n* es la suma de todos los elementos de S.

definition *sumaConj* :: *nat set* ⇒ *nat* **where**

sumaConj S $\equiv \sum S$

Donde $\sum$ ya se encuentra definido en Isabelle, pero se renombra de la siguiente forma:

abbreviation Sum ("$\sum$")
where "$\sum \equiv$ sum $(\lambda x.x)$"

El enunciado del teorema es el siguiente :

lemma *finite S* $\Longrightarrow \forall x \in S.\ x \leq$ *sumaConj S*
oops

Vamos a demostrar primero el lema enunciado anteriormente

lemma *aux-propiedad-conjuntos-finitos*:
assumes *finite S*
$x \notin S$
shows $x +$ *sumaConj S* $=$ *sumaConj* (*insert x S*)
proof −
have $x +$ *sumaConj S* $= x + \sum S$
by (*simp only*: *sumaConj-def*)
also have ... $=$ *sum* $(\lambda x.\ x)$ (*insert x S*)
using *assms*
by (*rule sum.insert*[*THEN sym*])
also have ... $=$ *sumaConj* (*insert x S*)
by (*simp only*: *sumaConj-def*)
finally show *?thesis*
by *this*
qed

En la demostración del lema anterior se ha usado *sumConj-def*, que hace referencia a la definición sumaConj que hemos hecho anteriormente.

Vamos a presentar diferentes formas de demostración:

1.2.3 Demostración automática

La demostración automática es:

lemma *finite S* $\Longrightarrow \forall x \in S.\ x \leq$ *sumaConj S*
by (*induct rule*: *finite-induct*)
(*auto simp add*: *sumaConj-def*)

1.2.4 Demostración detallada

La demostración declarativa es:

lemma *sumaConj-acota*:
finite S $\Longrightarrow \forall\, x{\in}S.\ x \leq$ *sumaConj S*
proof (*induct rule*: *finite-induct*)
show $\forall\, x \in \{\}.\ x \leq$ *sumaConj* $\{\}$
by (*simp only*: *ball-empty*)
next
fix *x* **and** *F*
assume *fF*: *finite F*
and *xF*: $x \notin F$
and *HI*: $\forall\ x{\in}F.\ x \leq$ *sumaConj F*
show $\forall\, y \in$ *insert x F*. $y \leq$ *sumaConj* (*insert x F*)
proof
fix *y*
assume $y \in$ *insert x F*
then have $y = x \vee y \in F$
by (*simp only*: *insert-iff*)
then show $y \leq$ *sumaConj* (*insert x F*)
proof
assume $y = x$
then have $y \leq x +$ (*sumaConj F*)
by (*simp only*: *le-add-same-cancel1*)
also have $\ldots =$ *sumaConj* (*insert x F*)
using *fF xF*
by (*rule aux-propiedad-conjuntos-finitos*)
finally show *?thesis*
by *this*
next
assume $y \in F$
then have $y \leq$ *sumaConj F*
using *HI*
by (*simp only*: *HI*)
also have $\ldots \leq x +$ (*sumaConj F*)
by (*simp only*: *le-add-same-cancel2*)
also have $\ldots =$ *sumaConj* (*insert x F*)
using *fF xF*
by (*rule aux-propiedad-conjuntos-finitos*)
finally show *?thesis*

```
      by this
    qed
  qed
qed
```

Capítulo 2

Teoría de funciones

Este capítulo muestra dos resultados de la teoría de funciones. Estos resultado son la caracterización de funciones inyectivas y sobreyectivas respectivamente. Estos teoremas muestran la capacidad de Isabelle/HOL de trabajar con tipos, debido a que, en la demostración de ambos teoremas es necesario especificar los dominios y codominios de las funciones.

2.1 Cancelación de funciones inyectivas

2.1.1 Demostración en lenguaje natural

El siguiente teorema que se va a probar es una caracterización de las funciones inyectivas. Primero se definirá el significado de inyectividad de una función y la propiedad de ser cancelativa por la izquierda.

Definición 2.1.1 *Una función $f : A \longrightarrow B$ es inyectiva si*

$$\forall x, y \in A : f(x) = f(y) \Longrightarrow x = y.$$

Antes de definir la propiedad de ser cancelativa por la izquierda se va a probar la siguiente propiedad.

Lema 2.1.2 *Sea f una función $f : A \longrightarrow B$. Las siguientes condiciones son equivalentes:*

1. $\forall C : (\forall g, h : C \longrightarrow A) : f \circ g = f \circ h \Longrightarrow g = h.$

2. $\forall g, h : \{0,1\} \longrightarrow A : f \circ g = f \circ h \Longrightarrow g = h$.

Demostración:

1 $\Longrightarrow$ 2 Trivial, ya que tomando en particular el conjunto $C = \{0,1\}$ se tiene probado.

2 $\Longrightarrow$ 1 La prueba se va a realizar por reducción al absurdo, es decir, supongamos que $\exists C : \exists g, h : C \longrightarrow A : f \circ g = f \circ h$ y $g \neq h$. Como $g \neq h$ esto implica que $\exists c \in C$ tal que $g(c) \neq h(c)$. Consideremos ahora $r : \{0,1\} \longrightarrow A$ tal que

$$r(x) = \begin{cases} c & si \quad x = 0 \\ c & si \quad x = 1 \end{cases}$$

Definamos entonces $g' = g \circ r$ y $h' = h \circ r$. Luego se tiene que

$$\begin{aligned} &(f \circ g')(x) \\ &= (f \circ g \circ r)(x) \\ &= ((f \circ g) \circ r))(x) \\ &= ((f \circ h) \circ r)(x)(\textit{Hipótesis inducción}) \\ &= (f \circ h \circ r)(x) \\ &= (f \circ h')(x) \end{aligned}$$

Entonces $f \circ g' = f \circ h'$. Luego, usando la hipótesis $g' = h'$.

Por otra parte

$$\begin{aligned} &g'(0) && h'(0) \\ &= (g \circ r)(0) && = (h \circ r)(0) \\ &= g(r(0)) && = h(r(0)) \\ &= g(c) && = h(c) \end{aligned}$$

Luego $g'(0) \neq h'(0)$ por lo que hemos llegado a un absurdo.

□

Puesto que tenemos esta equivalencia, la definición de la propiedad de ser cancelativa por la izquierda será:

Definición 2.1.3 (Cancelativa izquierda) *Una función $f : A \longrightarrow B$ es cancelativa por la izquierda si*

$$\forall g, h : \{0,1\} \longrightarrow A : f \circ g = f \circ h \Longrightarrow g = h.$$

Nota 2.1.4 *Es adecuado usar esta definición, ya que a la hora de especificar la definición en Isabelle no se puede cuantificar tipos y esta definición no lo requiere. Sin embargo, la otra sí.*

El teorema es el siguiente:

Teorema 2.1.5 *Una función f es inyectiva si y solo si es cancelativa por la izquierda.*

Vamos a hacer dos lemas previos, ya que se descompone la doble implicación en dos implicaciones y se va a demostrar cada una de ellos por separado.

Lema 2.1.6 (Condición necesaria) *Si f es una función inyectiva, entonces f es cancelativa por la izquierda.*

Demostración: Hay que probar que $\forall g, h : \{0,1\} \longrightarrow A : f \circ g = f \circ h$ esto implica que $g = h$. Luego, sean g, h tales que $f \circ g = f \circ h$, veamos que $\forall x.g(x) = h(x)$.

Se tiene que:

$$f \circ g = f \circ h \Longrightarrow (f \circ g)(x) = (f \circ h)(x) \overset{def.}{\Longrightarrow} f(g(x)) = f(h(x)) \overset{f\ inyect.}{\Longrightarrow} g(x) = h(x)$$

□

Lema 2.1.7 (Condición suficiente) *Si f es cancelativa por la izquierda, entonces f es inyectiva.*

Demostración: Sea $f : A \longrightarrow B$. Si el dominio de la función f fuese vacío, f es inyectiva. Supongamos que el dominio de la función f es distinto del vacío y que f verifica la propiedad de ser cancelativa por la izquierda. Hay que demostrar que $\forall a, b$ tales que $f(a) = f(b)$, esto implica que $a = b$.

Sean a, b tales que $f(a) = f(b)$.

Consideremos las funciones constantes $g : \{0,1\} \longrightarrow A$ tal que $g(x) = a$, $\forall x$ y $h : \{0,1\} \longrightarrow A$ tal que $h(x) = b$, $\forall x$. Veamos que $f \circ g = f \circ h$. En efecto, $\forall x$

$$\begin{array}{ll} (f \circ g)(x) & (f \circ h)(x) \\ = f(g(x)) & = f(h(x)) \\ = f(a) & = f(b) \end{array}$$

Por hipótesis se tiene que $f(a) = f(b)$ luego $f \circ g = f \circ h$. Por hipótesis se tiene que f es cancelativa por la izquierda, por lo tanto, esto implica que $g = h$. Es decir, que $\forall x, g(x) = h(x)$. Y, por tanto, $a = b$.

□

2.1.2 Especificación en Isabelle/Hol

Previamente a la especificación del teorema, vamos a definir en Isabelle la propiedad de que una función sea cancelativa por la izquierda.

definition *cancelativaIzquierda* :: $('a \Rightarrow 'b) \Rightarrow bool$ **where**
cancelativaIzquierda f =
$(\forall (g :: bool \Rightarrow 'a)\ h.\ (f \circ g = f \circ h \longrightarrow g = h))$

Nota 2.1.8 *En esta definición en Isabelle hemos definido las funciones g y h en los booleanos*$(\{0,1\})$*, aunque solo haría falta en un conjunto con ,al menos, 2 elementos, ya que realiza el mismo papel. Pero como el tipo booleano ya está predefinido en Isabelle utilizamos este.*

La especificación del teorema es la siguiente:

theorem *caracterizacion-funcion-inyecctiva*:
inj f $\longleftrightarrow$ *cancelativaIzquierda f*
oops

Al igual que en la demostración a mano, se va a demostrar a través de dos lemas asociados a cada implicación. Son los siguientes:

lemma *cancelativaIzquierda f* $\Longrightarrow$ *inj f*
oops

lemma *inj f* $\Longrightarrow$ *cancelativaIzquierda f*
oops

En la especificación anterior, *inj f* es una abreviatura de *inj-on f* definida en la teoría Fun.thy. Además, contiene la definición de *inj-on*

$$inj\text{-}on\ f\ A = (\forall\, x{\in}A.\ \forall\, y{\in}A.\ f\,x = f\,y \longrightarrow x = y) \qquad (inj\text{-}on\text{-}def)$$

Presentaremos distintas demostraciones de los lemas.

2.1.3 Demostración estructurada de los lemas

Las demostraciones declarativas son las siguientes:

lemma *condicion-necesaria-detallada*:
assumes *inj f*
shows *cancelativaIzquierda f*

```
proof −
 have ∀ (g :: bool ⇒ 'a) h. (f ∘ g = f ∘ h ⟶ g = h)
 proof (intro allI impI)
  fix g h :: bool ⇒ 'a
  assume f ∘ g = f ∘ h
  show g = h
  proof (rule ext)
   fix x
   have (f ∘ g)(x) = (f ∘ h)(x)
    by (simp only: ‹f ∘ g = f ∘ h›)
   then have f(g(x)) = f(h(x))
    by (simp only: comp-apply)
   then show g(x) = h(x)
    using ‹inj f›
    by (simp only: injD)
  qed
 qed
 then show cancelativaIzquierda f
  by (simp only: cancelativaIzquierda-def)
qed
```

```
lemma condicion-suficiente-detallada:
 assumes cancelativaIzquierda f
 shows inj f
proof (rule injI)
 fix a b
 assume f a = f b
 let ?g = λx :: bool. a
 let ?h = λx :: bool. b
 have ∀ (g :: bool ⇒ 'a) h. (f ∘ g = f ∘ h ⟶ g = h)
  using assms
  by (simp only: cancelativaIzquierda-def)
 then have ∀ h. (f ∘ ?g = f ∘ h ⟶ ?g = h)
  by (rule allE)
 then have (f ∘ ?g = f ∘ ?h ⟶ ?g = ?h)
  by (rule allE)
 moreover
 have f ∘ ?g = f ∘ ?h
```

```
 proof (rule ext)
  fix x :: bool
  have (f ∘ (λx :: bool. a)) x = f(a)
   by (simp only: comp-apply)
  also have ... = f(b)
   by (simp only: ⟨f a = f b⟩)
  also have ... =  (f ∘ (λx :: bool. b)) x
   by (simp only: comp-apply)
  finally show (f ∘ (λx. a)) x =  (f ∘ (λx. b)) x
   by this
 qed
 ultimately have ?g = ?h
  by (rule mp)
 then show  a = b
  by (rule fun-cong)
qed
```

Nota 2.1.9 *En la demostración de condición suficiente detallada, es necesario especificar los tipos tanto de las funciones como de los elementos. Ya que en caso de no especificarlo toma el tipo más general posible y no se puede demostrar.*

Otras demostraciones declarativas no detalladas usando demostradores automáticos metis, auto y blast son:

```
lemma condicion-necesaria-1:
 assumes inj f
 shows cancelativaIzquierda f
proof −
 have ∀ (g :: bool ⇒ 'a) h. (f ∘ g = f ∘ h ⟶ g = h)
 proof (intro allI impI)
 fix g h :: bool ⇒ 'a
 assume f ∘ g = f ∘ h
 then show g = h
  using ⟨inj f⟩
  by (simp add: inj-on-def fun-eq-iff)
 qed
 then show cancelativaIzquierda f
  by (simp only: cancelativaIzquierda-def)
qed
```

2.1.4 Demostración del teorema en Isabelle/Hol

En consecuencia, la demostración de nuestro teorema:

theorem *caracterizacion-inyectividad*:
inj f $\longleftrightarrow$ *cancelativaIzquierda f*
proof (*rule iffI*)
show *inj f* $\Longrightarrow$ *cancelativaIzquierda f*
by (*rule condicion-necesaria-detallada*)
next
show *cancelativaIzquierda f* $\Longrightarrow$ *inj f*
by (*simp only*: *condicion-suficiente-detallada*)
qed

Su demostración automática es

theorem *inj f* $\longleftrightarrow$ *cancelativaIzquierda f*
using *condicion-necesaria-detallada*
condicion-suficiente-detallada
by *auto*

2.2 Cancelación de las funciones sobreyectivas

2.2.1 Demostración en lenguaje natural

El siguiente teorema prueba una caracterización de las funciones sobreyectivas. Primero se definirá el significado de la sobreyectividad de una función y de la propiedad de ser cancelativa por la derecha.

Definición 2.2.1 *Una función* $f : A \longrightarrow B$ *es sobreyectiva si*

$$\forall y \in B : \exists x \in A : f(x) = y$$

Antes de definir la propiedad de ser cancelativa por la derecha se va a probar la siguiente propiedad.

Lema 2.2.2 *Sea* f *una función* $f : A \longrightarrow B$. *Las siguientes condiciones son equivalentes:*

1. $\forall C : (\forall g, h : B \longrightarrow C) : g \circ f = h \circ f \Longrightarrow g = h$

2. $\forall g, h : B \longrightarrow \{0,1\} : g \circ f = h \circ f \Longrightarrow g = h$.

Demostración:

$1 \Longrightarrow 2$ Trivial, ya que tomando en particular $C = \{0,1\}$ se tiene probado.

$2 \Longrightarrow 1$ La demostración se realizará por reducción al absurdo, es decir, supongamos que $\exists C$ y $\exists g, h$ tales que $g \circ f = h \circ f$ y $g \neq h$. Como $g \neq h$ esto implica que $\exists b \in B$ tal que $g(b) \neq h(b)$.

Definamos $r : C \longrightarrow \{0,1\}$ como:

$$r(x) = \begin{cases} 0 & si \quad x \neq g(b) \\ 1 & si \quad x = g(b) \end{cases}$$

Consideremos $g' = r \circ g$ y $h' = r \circ h$. Luego se tiene que:

$$\begin{array}{l} (g' \circ f)(x) \\ = (r \circ g \circ f)(x) \\ = (r \circ (g \circ f))(x) \\ = (r \circ (h \circ f))(x) H.I \\ = (r \circ h \circ f)(x) = (h' \circ f)(x). \end{array}$$

Por lo tanto, $g' \circ f = h' \circ f$. Luego por hipótesis $g' = h'$.

Veamos por otro lado que:

$$\begin{array}{ll} g'(b) & h'(b) \\ = r(g(b)) & = r(h(b)) \\ = 1 & = 0 \end{array}$$

Por lo que hemos llegado a un absurdo.

□

Puesto que tenemos esta equivalencia, la definición de la propiedad de ser cancelativa por la derecha será:

Definición 2.2.3 (Cancelativa derecha) *Una función $f : A \longrightarrow B$ tiene la propiedad de ser cancelativa por la derecha si:*

$$\forall g, h : B \longrightarrow \{0,1\} : g \circ f = h \circ f \Longrightarrow g = h.$$

Nota 2.2.4 *Es adecuado usar esta definición, ya que a la hora de especificar la definición en Isabelle no se puede cuantificar tipos y esta definición no lo requiere. Sin embargo, la otra sí.*

El teorema es el siguiente:

Teorema 2.2.5 *Una función f es sobreyectiva si y solo si es cancelativa por la derecha.*

El teorema se puede dividir en dos lemas, ya que se demuestra por una doble implicación.

Lema 2.2.6 (Condición necesaria) *Si f es sobreyectiva, entonces f es cancelativa por la derecha.*

Demostración: Sea $f : A \longrightarrow B$ sobreyectiva. Veamos que f es cancelativa por la derecha, es decir, sean $g, h : B \longrightarrow \{0,1\}$ tales que $g \circ f = h \circ f$ hay que probar que $g = h$. Usando la definición de sobreyectividad $(\forall y \in Y, \exists x | y = f(x))$ y la hipótesis, tenemos que:

$$\begin{aligned} & g(y) \\ &= g(f(x)) \\ &= (g \circ f)(x) \\ &= (h \circ f)(x) \\ &= h(f(x)) \\ &= h(y). \end{aligned}$$

□

Lema 2.2.7 (Condición suficiente) *Si f es cancelativa por la derecha entonces f es sobreyectiva.*

Demostración:

La prueba se va a realizar por reducción al absurdo. Luego supongamos que nuestra función $f : A \longrightarrow B$ no es sobreyectiva, es decir, $\exists y_1 \in B$ *tal que* $\nexists x \in A : f(x) = y_1$.

Definamos ahora las funciones $g, h : B \longrightarrow \{0,1\}$

$$g(y) = 0 \; \forall y \in B$$

$$h(y) = \begin{cases} 0 & si \quad y \neq y_1 \\ 1 & si \quad y = y_1 \end{cases}$$

Entonces $g(y) \neq h(y)$. Sin embargo, por hipótesis se tiene que si $g \circ f = h \circ f$ entonces $h = g$. En efecto,

$$\begin{aligned} & (g \circ f)(x) \\ &= g(f(x)) \\ &= 0 \\ &= h(f(x)) \\ &= (h \circ f)(x). \end{aligned}$$

Por lo que $g(y) = h(y)$, con lo que hemos llegado a una contradicción.

□

2.2.2 Especificación en Isabelle/Hol

Su especificación es la siguiente, que se dividirá en dos al igual que en la demostración a mano:

definition *cancelativaDerecha* :: $('a \Rightarrow 'b) \Rightarrow bool$ **where**
cancelativaDerecha f =
$(\forall (g :: 'b \Rightarrow bool)\ h.\ (g \circ f = h \circ f \longrightarrow g = h))$

theorem *caracterizacion-funciones-sobreyectivas*:
surj f $\longleftrightarrow$ *cancelativaDerecha f*
oops

lemma *condicion-suficiente*:
surj f $\Longrightarrow$ *cancelativaDerecha f*
oops

lemma *condicion-necesaria*:
cancelativaDerecha f $\Longrightarrow$ *surj f*
oops

En la especificación anterior, *surj f* es una abreviatura de *range f* = *UNIV*, donde *range f* es el rango o imagen de la función f y *UNIV* es el conjunto universal definido en la teoría Set.thy como una abreviatura de *top* que, a su vez está definido en la teoría Orderings.thy mediante la siguiente propiedad

$$\frac{\textit{ordering-top less-eq less top}}{\textit{less-eq a top}} \qquad (\textit{ordering-top.extremum})$$

Además queda añadir que la teoría donde se encuentra definido *surj f* es en Fun.thy. Esta teoría contiene la definicion *surj-def*.

$$\textit{surj}\, f = (\forall y.\ \exists x.\ y = f\, x) \qquad (\textit{surj--def})$$

2.2.3 Demostración estructurada

Presentaremos distintas demostraciones de los lemas. Las primeras son las detalladas:

```
lemma condicion-suficiente-detallada:
  assumes surj f
  shows cancelativaDerecha f
proof −
  have ∀ (g :: 'a ⇒ bool) h. (g ∘ f = h ∘ f ⟶ g = h)
  proof (intro allI impI)
    fix g h :: 'a ⇒ bool
    assume g ∘ f = h ∘ f
    show g = h
    proof  (rule ext)
      fix x
      have ∃ y. x = f y
        using assms
        by (simp only: surjD)
      then obtain y where x = f y
        by (rule exE)
      then have g x = g (f y)
        by (simp only: ‹x = f y›)
      also have ... = (g ∘ f) y
        by (simp only: comp-apply)
      also have ... = (h ∘ f) y
        by (simp only: ‹g ∘ f = h ∘ f›)
      also have ... = h (f y)
        by (simp only: comp-apply)
      also have ... = h x
        by (simp only: ‹x = f y›)
      finally show g x = h x
        by this
    qed
  qed
  then show cancelativaDerecha f
    by (simp only: cancelativaDerecha-def)
qed

lemma condicion-necesaria-detallada-l1:
  assumes ∄ x. y = f x
  shows g ∘ f = g(y := z) ∘ f
proof (rule ext)
  fix a
  show (g ∘ f) a = (g(y := z) ∘ f) a
```

```
  proof −
   have ∀ x. y ≠ f x
    using assms
    by (rule Meson.not-exD)
   then have y ≠ f a
    by (rule allE)
   then have f a ≠ y
    by (rule not-sym)
   have (g ∘ f) a = g (f a)
    by (simp only: o-apply)
   also have ... = (g(y := z)) (f a)
    using ⟨f a ≠ y⟩
    by (rule fun-upd-other [THEN sym])
   also have ... = (g(y := z) ∘ f) a
    by (simp only: o-apply)
   finally show ?thesis
    by this
  qed
qed

lemma condicion-necesaria-detallada-l2:
  assumes (λx. a) = (λx. a)(y := b)
  shows a = b
proof −
  have a = ((λx. a)(y := b)) y
   using assms
   by (rule fun-cong)
  also have ... = b
   by (rule fun-upd-same)
  finally show a = b
   by this
qed

lemma condicion-necesaria-detallada:
  assumes cancelativaDerecha f
  shows surj f
proof −
  have ∀ y. ∃ x. y = f x
  proof (rule ccontr)
   assume ¬ (∀ y. ∃ x. y = f x)
```

```
  then have ∃ y. ∄ x. y = f x
    by (rule Meson.not-allD)
  then obtain y0 where ∄ x. y0 = f x
    by (rule exE)
  then have ∀ x. y0 ≠ f x
    by (rule Meson.not-exD)
  let ?g = (λx. True) :: 'b ⇒ bool
  let ?h = ?g(y0 := False)
  have ∀ (g :: 'b ⇒ bool) h . g ∘ f = h ∘ f ⟶ g = h
    using assms by (simp only: cancelativaDerecha-def)
  then have ∀ h .(?g ∘ f = h ∘ f) ⟶ (?g = h)
    by (rule allE)
  then have (?g ∘ f = ?h ∘ f) ⟶ (?g = ?h)
    by (rule allE)
  moreover
  have (?g ∘ f = ?h ∘ f)
    using ⟨∄ x. y0 = f x⟩
    by (rule condicion-necesaria-detallada-l1)
  ultimately have (?g = ?h)
    by (rule mp)
  then have True = False
    by (rule condicion-necesaria-detallada-l2)
  with True-not-False show False
    by (rule notE)
 qed
 then show surj f
  using surj-def
  by (rule rev-iffD2)
qed
```

2.2.4 Demostración teorema

En consecuencia, la demostración del teorema es

```
theorem caracterizacion-funciones-sobreyectivas:
 surj f ⟷ cancelativaDerecha f
proof (rule iffI)
 show surj f ⟹ cancelativaDerecha f
  by (rule condicion-suficiente-detallada)
next
 show cancelativaDerecha f ⟹ surj f
```

by (*rule condicion-necesaria-detallada*)
qed

2.2.5 Demostración automática del teorema

theorem *surj f* $\longleftrightarrow$ *cancelativaDerecha f*
by (*auto simp add*: *condicion-suficiente-detallada*
condicion-necesaria-detallada)

Capítulo 3

Teoría de conjuntos

En este capítulo se analiza un teorema importante de la teoría de conjuntos, el teorema de Cantor; en el que la demostración formal y en lenguaje natural es prácticamente idéntica.

3.1 Teorema de Cantor

3.1.1 Demostración en lenguaje natural

El siguiente teorema, denominado teorema de Cantor por el matemático Georg Cantor, es un resultado importante de la teoría de conjuntos.

Para la exposición del teorema vamos a definir una serie de conceptos:

Definición 3.1.1 (Conjunto Potencia) *El conjunto potencia de un conjunto A ($\mathcal{P}(A)$) es el conjunto formado por todos los subconjuntos de A.*

Definición 3.1.2 (Cardinal) *El cardinal de un conjunto A (denotado $\#A$) es el número de elementos del propio conjunto.*

El enunciado original del teorema es el siguiente :

Teorema 3.1.3 *El cardinal del conjunto potencia de cualquier conjunto A es estrictamente mayor que el cardinal de A, o lo que es lo mismo, $\#\mathcal{P}(A) > \#A$.*

El enunciado del teorema lo podemos reformular como sigue:

Teorema 3.1.4 *Dado un conjunto A,* $\nexists f : A \longrightarrow \mathcal{P}(A)$ *sobreyectiva.*

Demostración: La prueba se va a realizar por reducción al absurdo.

Supongamos que $\exists f : A \longrightarrow \mathcal{P}(A)$ sobreyectiva, es decir, $\forall B \in \mathcal{P}(A), \exists x \in A$ tal que $B = f(x)$.

En particular, tomemos el conjunto

$$B = \{x \in A : x \notin f(x)\}$$

y supongamos que $\exists a \in A : B = f(a)$, ya que B es un subconjunto de A. Luego podemos distinguir dos casos :

1. Si $a \in B$. Entonces por definición del conjunto B se tiene que $a \notin B$, luego se llega a una contradicción.

2. Si $a \notin B$. Entonces por definición de B se tiene que $a \in B$, luego se ha llegado a otra contradicción.

En los dos casos se ha llegado a una contradicción, por lo que no existe a y f no es sobreyectiva.

□

3.1.2 Especificación en Isabelle/HOL

Para la especificación del teorema en Isabelle, primero debemos notar que

$$f :: 'a \Rightarrow 'a\ set$$

significa que es una función de tipos, donde $'a$ significa un tipo y para poder denotar el conjunto potencia tenemos que poner $'a\ set$ que significa que es de un tipo formado por conjuntos del tipo $'a$.

El enunciado del teorema es el siguiente :

theorem *Cantor*: $\nexists f :: 'a \Rightarrow 'a\ set. \forall A. \exists x. A = f\ x$
oops

A continuación presentaremos diferentes formas de demostración del teorema.

3.1.3 Demostración detallada

La primera es la demostración detallada del teorema:

theorem

```
  ∄f :: 'a ⇒ 'a set. ∀B. ∃x. B = f x
proof (rule notI)
  assume ∃f :: 'a ⇒ 'a set. ∀A. ∃x. A = f x
  then obtain f :: 'a ⇒ 'a set where Hf: ∀A. ∃x. A = f x
    by (rule exE)
  let ?B = {x. x ∉ f x}
  from Hf obtain ∃x. ?B = f x
    by (rule allE)
  then obtain a where Ha: ?B = f a
    by (rule exE)
  show False
  proof (cases a ∈ ?B)
    assume a ∈ ?B
    then have a ∉ f a
      by (rule CollectD)
    moreover
    have a ∈ f a
      using ⟨a ∈ ?B⟩
      by (simp only: Ha)
    ultimately show False
      by (rule notE)
  next
    assume a ∉ ?B
    with Ha have a ∉ f a
      by (rule subst)
    moreover
    have a ∈ f a
    proof (rule ccontr)
      assume a ∉ f a
      then have a ∈ ?B
        by (rule CollectI)
      with ⟨a ∉ ?B⟩ show False
        by (rule notE)
    qed
    ultimately show False
      by (rule notE)
  qed
qed
```

3.1.4 Demostración automática

La demostración automática del teorema es:

theorem *Cantor*:
$\nexists f :: 'a \Rightarrow 'a\ set.\ \forall B.\ \exists x.\ B = f\ x$
by *best*

Capítulo 4

Teoría de retículos

En este capítulo se muestra un teorema de la teoría de retículos, el teorema de Knaster-Tarski. En Isabelle/HOL la teoría de retículos ya se encuentra predefinida y tanto los retículos como los retículos completos se definen como clases.

4.1 Teorema de Knaster Tarski

4.1.1 Demostración en lenguaje natural

El siguiente teorema, denominado teorema de Knaster-Tarski del punto fijo, es un teorema de la teoría de retículos y lleva el nombre de los matemáticos Bronislaw Knaster y Alfred Tarski.

Para la exposición y demostración del teorema es necesario definir una serie de conceptos previos.

Definición 4.1.1 *Sea L un conjunto. Un orden parcial sobre L es una relación binaria $\leq$ sobre L, tal que $\forall x, y, z \in L$ se verifica:*

1. *$x \leq x$ (Propiedad reflexiva).*
2. *Si $x \leq y$ e $y \leq x$ entonces $x = y$ (Propiedad antisimétrica).*
3. *Si $x \leq y$ e $y \leq z$ entonces que $x \leq z$ (Propiedad transitiva).*

Definición 4.1.2 *Un conjunto L con un relación de orden $(\leq)$ se denomina conjunto parcialmente ordenado y se denota $(L, \leq)$.*

Definición 4.1.3 *Dado un subconjunto S de un conjunto $(L, \leq)$ parcialmente ordenado, se define el supremo de S(sup S), si existe, al mínimo elemento de S que mayor o igual que cada elementos de S.*

Definición 4.1.4 *Dado un subconjunto S de un conjunto $(L, \leq)$ parcialmente ordenado, se define el ínfimo de S(inf S), si existe, al máximo elemento de S que menor o igual que cada elementos de S.*

Definición 4.1.5 *Sea $(L, \leq)$ un conjunto no vacío parcialmente ordenado. Se dirá que L es un retículo si:*

1. *Si $\forall a, b \in L$ existe $sup(\{a, b\})$.*
2. *Si $\forall a, b \in L$ existe $inf(\{a, b\})$.*

Definición 4.1.6 *Sea $(L, \leq)$ un conjunto parcialmente ordenado no vacío. Se dirá que L es un retículo completo si $\forall S \subset L$ existe $sup(S)$ e $inf(S)$.*

Ejemplo 4.1.7 *Ahora vamos a dar una serie de ejemplos de conjuntos que son retículos, retículos completos y algunos que no son retículos.*

- *Los subconjuntos de un conjunto dado con la relación de orden la inclusión es un retículo y el supremo está dado por la unión y el ínfimo por la intersección.*
- *Los enteros no negativos, con la relación de orden la divisibilidad es un retículo con el ínfimo el mínimo común múltiplo y el supremo el máximo común divisor.*
- *Los enteros no negativos, con la relación de orden la divisibilidad es un retículo completo siendo el ínfimo de este conjunto el 1 ya que divide a cualquier número y siendo el supremo el 0 ya que es divisible por cualquier número.*
- *Un ejemplo de un conjunto paricalmente ordenado, pero que no es un retículo es el conjunto $\{\emptyset, \{0\}, \{1\}\}$ con la relación de orden la inclusión. No es un retículo ya que el $\{0\}$ y el $\{1\}$ no tienen supremo.*

Definición 4.1.8 *Una función $f : L \longrightarrow R$ entre dos conjuntos parcialmente ordenados, $(L, \leq)$ y $(R, \leq')$ respectivamente. Se dirá que es monótona si conserva el orden, es decir, si $x \leq y$ implica $f(x) \leq' f(y)$.*

Definición 4.1.9 *Sea $(L, \leq)$ un conjunto parcialmente ordenado y $f : L \longrightarrow L$. Diremos que x es un punto fijo de una función si y solo si $f(x) = x$.*

El enunciado del teorema es el siguiente:

Teorema 4.1.10 *Sea L un retículo completo y $f : L \longrightarrow L$ una función monótona. Entonces $\exists a \in L$ punto fijo de f.*

Demostración: Hay que probar que $\exists a \in L$ tal que $f(a) = a$.
Sea $H = \{x \in L | f(x) \leq x\}$. Como L, por hipótesis, es un retículo completo tenemos que $\exists a = inf H$, luego como a es una cota inferior se tiene que $\forall x \in H\, a \leq x$. Como f es una función monótona, $f(a) \leq f(x) \leq x\, \forall x \in H$. Luego se obtiene que $f(a)$ es una cota inferior de H, por ser cota inferior llegamos a que $f(a) \leq a$. Ahora veamos que $f(a) \geq a$ y ya se tendría probado el teorema. En efecto, como f es monótona, $f(f(a)) \leq f(a)$. Esto implica que $f(a) \in H$ y como a es cota inferior de H entonces $a \leq f(a)$.

□

4.1.2 Especificación en Isabelle/HOL

Para la comprensión de la especificación vamos a notar una serie de definiciones y notación que se encuentran en la teoría de retículos y retículos completos importada en Isabelle, Lattice.thy y LatticeComplete.thy respectivamente.

La notación requerida para la comprensión de la especificación y demostración del teorema que viene importada de Lattice-Syntax.thy es:

notation
bot ($\bot$) and
top ($\top$) and
inf (infixl $\sqcap$ 70) and
sup (infixl $\sqcup$ 65) and
Inf ($\sqcap$– [900] 900) and
Sup ($\sqcup$– [900] 900)

Tanto los retículos, como los retículos completos se definen en Isabelle como clases:

Retículo:
class lattice =
assumes ex-inf : $\exists$ inf. is-inf x y inf.
assumes ex-sup : $\exists$ sup. is-sup x y sup.

Retículo completo:
class complete-lattice = lattice + Inf + Sup + bot + top +
assumes Inf-lower : $x \in A \Longrightarrow A \leq x$

and Inf-greatest : $(\bigwedge x.x \in A \Longrightarrow z \leq x) \Longrightarrow z \leq \sqcap A$.
and Sup-upper : $x \in A \Longrightarrow x \leq \sqcup A$
and Sup-least : $(\bigwedge x.x \in A \Longrightarrow x \leq z) \Longrightarrow \sqcup A \leq z$
and Inf-empty [simp] : $\sqcap\{\} = \top$
and Sup-empty [simp] : $\sqcup\{\} = \bot$

Para la especificiación del teorema también debemos notar que:

$$f :: "'a :: \textit{complete-latices} \Rightarrow' a"$$

significa que f es una función cuyo dominio y codominio es un retículo completo.

La especifiación del teorema es:

theorem *Knaster-Tarski*:
fixes $f :: 'a :: \textit{complete-lattice} \Rightarrow 'a$
assumes *mono f*
shows $\exists a. f\, a = a$
oops

A continuación presentaremos distintas demostraciones del teorema.

4.1.3 Demostración detallada

theorem *Knaster-Tarski-detallada*:
fixes $f :: 'a :: \textit{complete-lattice} \Rightarrow 'a$
assumes *mono f*
shows $\exists a. f\, a = a$
proof
let $?H = \{u. f\, u \leq u\}$
let $?a = \bigsqcap ?H$
show $f\, ?a = ?a$
proof −
have $f\, ?a \leq ?a$
proof (*rule Inf-greatest*)
fix x
assume $x \in ?H$
then have $?a \leq x$
by (*rule Inf-lower*)
with ⟨*mono f*⟩ **have** $f\, ?a \leq f\, x$
by (*rule monoD*)

```
    also have f x ≤ x
     using ⟨x ∈ ?H⟩
     by (rule CollectE)
    finally show f ?a ≤ x
     by this
   qed
   from ⟨mono f⟩ and ⟨f ?a ≤ ?a⟩ have f (f ?a) ≤ f ?a
    by (rule monoD)
   then have f ?a ∈ ?H
    by (rule CollectI)
   then have ?a ≤ f ?a
    by (rule Inf-lower)
   show ?thesis
    using ⟨f ?a ≤ ?a⟩
          ⟨?a ≤ f ?a⟩
    by (rule order-antisym)
  qed
qed
```

4.1.4 Demostración automática

```
theorem Knaster-Tarski-automatica:
 fixes f :: 'a :: complete-lattice ⇒ 'a
 assumes mono f
 shows ∃ a. f a = a
proof
 let ?H = {u. f u ≤ u}
 let ?a = ⨅ ?H
 show f ?a = ?a
 proof −
  have f ?a ≤ ?a
  proof (rule Inf-greatest)
   fix x
   assume x ∈ ?H
   then have ?a ≤ x
    by (simp add: Inf-lower)
   then show f ?a ≤ x
    by (metis (mono-tags, lifting) ⟨x ∈ ?H⟩ assms le-INF-iff
       mem-Collect-eq mono-Inf order.trans)
  qed
  then show f ?a = ?a
```

```
    by (meson CollectI Inf-lower antisym assms mono-def)
  qed
qed
```

Capítulo 5

Teoría de geometría

En este capítulo se formalizará la teoría de la geometría, dividiendola en tres tipos: geometría simple, geometría no proyectiva y geometría proyectiva. También se mostrará la interpretación del mínimo modelo de cada geometría en Isabelle/HOL.

5.1 Introducción a la geometría

La geometría posee una larga de historia de estar presentada y representada por sistemas axiomáticos; es decir, mediante conjuntos de axiomas a partir del cual se pueden derivar lógicamente teoremas. Un axioma es una declaración que se considera verdadera, que sirve como punto de partida para razonamientos y argumentos adicionales.

Por ello, vamos a representar la geometria simple, que la entenderemos definiendo el plano como un conjunto de puntos y las líneas como conjuntos de puntos, la geometría no proyectiva añadiéndole un axioma a la simple y por último, la geometria proyectiva añadiendole 3 axiomas a la simple.

Todo esto se definirá en Isabelle/HOL como un entorno local. Un entorno local o declaración local consiste en secuencia de elementos que declararán parámetros(**fixed**) y suposiciones (**assumption**).

También de cada tipo de geometría se dará el modelo mínimo que posee cada una, esto se hará mediante el comando **interpretation**. El comando **interpretation** como su nombre indica consiste en interpretar los comandos locales; es decir, dar un modelo (que en este caso será el mínimo que ofrece cada entorno local) y probar todos los axiomas que este tenga.

5.2 Geometría simple

5.2.1 Entorno local

La geometría simple, como ya se ha dicho anteriormente, posee tres elementos fundamentales. Los puntos, el plano, que es el conjunto de todos ellos, y las rectas, que son conjuntos de puntos. Esta geometría posee 5 axiomas:

1. El plano es no vacío.
2. Toda línea es un subconjunto no vacío del plano.
3. Para cualquier par de puntos en el plano, existe una línea que contiene a ambos.
4. Dos líneas diferentes se cortan en no más de un punto.
5. Para cada línea, existe un punto del plano que no pertenece a ella.

Se ha declarado un entorno local, denotado **Simple−Geometry**, con un par de constantes (**lines** y **plane**) junto con los 5 axiomas anteriores.

locale *Simple-Geometry* =
  fixes *plane* :: *'a set*
  fixes *lines* :: (*'a set*) *set*
  assumes *A1*: $plane \neq \{\}$
    and *A2*: $\forall l \in lines.\ l \subseteq plane \wedge l \neq \{\}$
    and *A3*: $\forall p \in plane.\ \forall q \in plane.\ \exists l \in lines.\ \{p,q\} \subseteq l$
    and *A4*: $\forall l \in lines.\ \forall r \in lines.$
        $l \neq r \longrightarrow l \cap r = \{\} \vee (\exists q \in plane.\ l \cap r = \{q\})$
    and *A5*: $\forall l \in lines.\ \exists q \in plane.\ q \notin l$

A pesar de la definición del anterior entorno local con 5 axiomas, no en todas las demostraciones, se van a usar todos ellos. Sin embargo, al haber definido tanto las líneas como el plano como conjuntos tenemos todas las funciones definidas en Isabelle/HOL de la teoría de conjuntos Set.thy.

5.2.2 Proposiciones de geometría simple

A continuación vamos a presentar una serie de lemas que vamos a demostrar dentro del entorno de la geometría simple.

El primer lema es el siguiente:

Lema 5.2.1 *Existe al menos una línea.*

Demostración: Vamos a demostrar que el conjunto de líneas es no vacío. Para ello, supongamos en primer lugar, por el axioma A1, que q es un punto del plano. Entonces, por el axioma A3, tenemos que existe una línea l tal que $\{q, q\} \subseteq l$. Luego, ya hemos probado que existe una línea.

□

La formalización del lema y su demostración en Isabelle/HOl es la siguiente:

```
lemma (in Simple-Geometry) one-line-exists:
 ∃l. l ∈ lines
proof −
 have ∃q. q ∈ plane  using A1 by auto
 then obtain q1 where q1 ∈ plane by (rule exE)
 then obtain ∃l ∈ lines. {q1, q1} ⊆ l using A3 by auto
 then show ?thesis by auto
qed
```

El segundo lema es el siguiente

Lema 5.2.2 *Existen al menos dos puntos que son diferentes en el plano*

Demostración: Para la demostración del lema, usando el lema anterior, tenemos que existe una línea l. Además, por el axioma A2, sabemos que $l \neq \emptyset$ lo que implica que existe un punto q en l. Por otro lado, por el axioma A5, sabemos que existe un punto p que no está en l. Luego ya tenemos probada la existencia de dos puntos. A parte, como $p \notin l$ y $q \in l$ tienen que ser distintos.

□

La especificación y demostración del lema en Isabelle/HOL es la siguiente:

```
lemma (in Simple-Geometry) two-points-exist:
 ∃p1 p2. p1 ≠ p2 ∧ {p1, p2} ⊆ plane
proof −
 obtain l1 where l1 ∈ lines
  using one-line-exists by (rule exE)
 then obtain l1 ⊆ plane ∧ l1 ≠ {}
  using A2 by auto
 then have ∃q. q ∈ l1 ∧ q ∈ plane
  by auto
 then obtain p1 where p1 ∈ l1 ∧ p1 ∈ plane
  by (rule exE)
 moreover obtain p2 where p2 ∈ plane ∧ p2 ∉ l1
```

```
  using ⟨l1 ∈ lines⟩ A5 by auto
 ultimately show ?thesis
  by force
qed
```

El siguiente lema es el siguiente:

Lema 5.2.3 *Existen al menos tres puntos diferentes en el plano.*

Demostración: Para la demostración del lema vamos a usar el lema anterior; es decir, tenemos que existen dos puntos distintos p y q. Por el axioma A3, se tiene que existe una línea l que pasa por esos dos puntos. Usando el axioma A5, sabemos que existe un punto r que no pertenece a l. Veamos que son diferentes; es decir, como hemos tomado $p \neq q$ simplemente tenemos que probar que $r \neq q$ y $r \neq p$. Como $r \notin l$ ya se tiene la prueba.

□

La especificación y demostración del lema en Isabelle/HOL es la siguiente:

```
lemma (in Simple-Geometry) three-points-exist:
 ∃ p1 p2 p3. distinct [p1, p2, p3] ∧ {p1, p2, p3} ⊆ plane
proof −
 obtain p1 p2 where p1 ≠ p2 ∧ {p1, p2} ⊆ plane
  using two-points-exist by auto
 moreover then obtain l1 where l1 ∈ lines ∧ {p1, p2} ⊆ l1
  using A3 by auto
 moreover then obtain p3 where p3 ∈ plane ∧ p3 ∉ l1
  using A5 by auto
 ultimately have distinct [p1, p2, p3] ∧ {p1, p2, p3} ⊆ plane
  by auto
 then show ?thesis
  by (intro exI)
qed
```

El siguiente lema es una consecuencia inmediata del lema anterior.

Lema 5.2.4 *Si el plano es finito, entonces la cardinalidad del plano es mayor o igual que 3.*

La especificación y demostración del lema en Isabelle/HOL es la siguiente:

```
lemma (in Simple-Geometry) card-of-plane-greater:
 assumes finite plane
 shows card plane ≥ 3
```

proof −
obtain *p1 p2 p3* **where**
distinct [*p1*, *p2*, *p3*] ∧ {*p1*, *p2*, *p3*} ⊆ *plane*
using *three-points-exist* **by** *auto*
moreover then have {*p1*, *p2*, *p3*} ⊆ *plane*
by (*rule conjE*)
then have *card* {*p1*, *p2*, *p3*} ≤ *card plane*
using *assms* **by** (*simp add*: *card-mono*)
ultimately show *?thesis*
by *auto*
qed

Lema 5.2.5 *Sean a y b dos puntos distintos, l una línea que pasa por ellos y p un punto fuera de l. Sea n una línea que pasa por a y p y m una línea que pasa b y p. Entonces, $m \neq n$.*

Demostración: La demostración se hará por reducción al absurdo; es decir, supongamos que $m = n$ y se llegará a un absurdo. Primero notemos que $m \neq l$ ya que $p \notin l$ pero $p \in m$, luego podemos aplicar el axioma A4 a las líneas m y l. Al aplicarlo resulta que tenemos que $l \cap m = \emptyset$ o existe un punto q tal que $l \cap m = \{q\}$.

Primero supongamos que $l \cap m = \emptyset$, sin embargo $b \in l$ y $b \in m$ luego hemos llegado a n absurdo.

Segundo supongamos que sea q el punto tal que $l \cap m = q$, sin embargo al principio se ha supuesto que $m = n$. Por lo tanto, se tiene que $\{a, b\} \subseteq \{q\}$ con lo que se ha llegado a un absurdo ya que $a \neq b$.

Por los dos casos se ha llegado a un absurdo luego, $m \neq n$.

□

Para tener una visión geométrica de la demostración incluimos la figura 5.4.6.

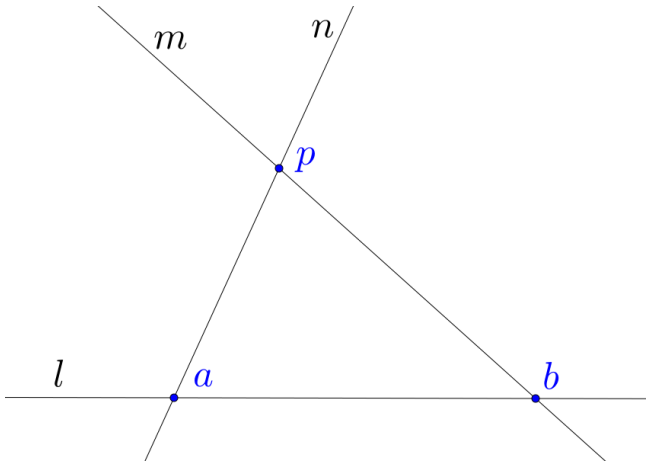

Figura 5.1: Visión geométrica de la demostración de líneas diferentes

La especificación y demostración del lema en Isabelle/HOL es la siguiente:

lemma (**in** *Simple-Geometry*) *how-to-produce-different-lines*:
assumes
$l \in lines$
$\{a, b\} \subseteq l$ $a \neq b$
$p \notin l$
$n \in lines$ $\{a, p\} \subseteq n$
$m \in lines$ $\{b, p\} \subseteq m$
shows $m \neq n$
proof (*rule notI*)
assume $m = n$
show *False*
proof −
have $m \neq l$
using *assms*(4, 8) **by** *auto*
moreover have $l \neq m \longrightarrow l \cap m = \{\} \vee (\exists q \in plane.\ l \cap m = \{q\})$
using *assms*(1, 7) *A4* **by** *auto*
ultimately have $l \cap m = \{\} \vee (\exists q \in plane.\ l \cap m = \{q\})$
by *auto*
then show *False*
proof (*rule disjE*)
assume $l \cap m = \{\}$
then show *False*
using *assms*(2, 6) ⟨$m = n$⟩ **by** *auto*
next
assume $\exists q \in plane.\ l \cap m = \{q\}$

```
    then obtain q where q ∈ plane ∧ l ∩ m = {q}
     by auto
    then have l ∩ m = {q}
     by (rule conjE)
    then have {a, b} ⊆ {q}
     using assms(2, 6, 8) ⟨m = n⟩ by auto
    then show False
     using assms(3) by auto
   qed
  qed
qed
```

Lema 5.2.6 *Sea l una línea tal que existen dos puntos $\{a, b\} \subseteq l$ con $a \neq b$, un punto p tal que $p \notin l$. Sea n una línea tal que $\{a, p\} \subseteq n$ y m otra línea tal que $\{b, p\} \subseteq m$. Supongamos además que existen otros dos puntos c, d tales que pertenecen a n y m respectivamente y $c \neq p$. Entonces $c \neq d$.*

Demostración: La demostración se hará por reducción al absurdo, es decir, supongamos que $c = d$ y llegaremos a una contradicción. Tenemos todas las hipótesis del lema anterior, luego $m \neq n$, por lo que podemos aplicar el axioma A4 a las líneas m y n. Se tiene por lo tanto que $m \cap n = \emptyset$ o existe un punto q tal que $m \cap n = \{q\}$.

Primero supongamos que $m \cap n = \emptyset$, sin embargo por hipótesis se tiene que $p \in m$ y $p \in n$ luego hemos llegado a una contradicción.

Segundo sea q el punto tal que $m \cap n = \{q\}$. Como se ha supuesto que $c = d$ se tiene que $c, p \subseteq \{q\}$, pero por hipótesis se tiene que $c \neq p$ luego se ha llegado a una contradicción.

En los dos caso se ha llegado a una contradicción, por lo que $c \neq d$.

□

Para entender mejor la demostración se puede ver geométricamente en la siguiente figura 5.4.8

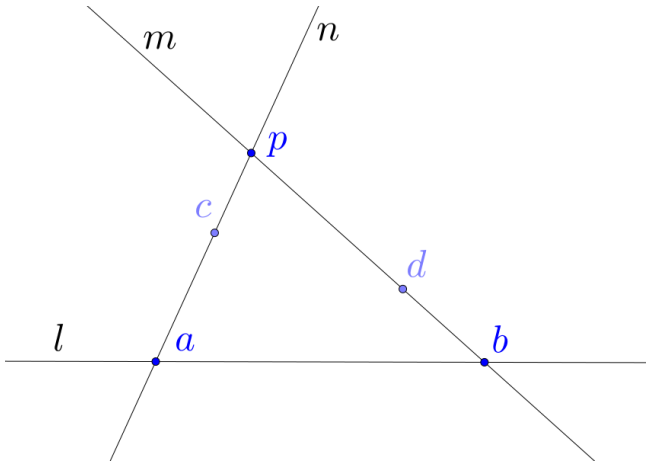

Figura 5.2: Visión geométrica de la demostración de puntos diferentes

La especificación y demostración del lema en Isabelle/HOL es la siguiente:

lemma (**in** *Simple-Geometry*) *how-to-produce-different-points*:
assumes
$l \in lines$
$\{a, b\} \subseteq l$ $a \neq b$
$p \notin l$
$n \in lines$ $\{a, p, c\} \subseteq n$
$m \in lines$ $\{b, p, d\} \subseteq m$
$p \neq c$
shows $c \neq d$
proof
assume $c = d$
show *False*
proof −
have $m \neq n$
using *assms how-to-produce-different-lines* **by** *simp*
moreover have $n \neq m \longrightarrow m \cap n = \{\} \vee (\exists q \in plane.\ m \cap n = \{q\})$
using *assms*(5,7) *A4* **by** *auto*
ultimately have $m \cap n = \{\} \vee (\exists q \in plane.\ m \cap n = \{q\})$
by *auto*
then show *False*
proof (*rule disjE*)
assume $m \cap n = \{\}$
then show *False*
using *assms*(6, 8) **by** *auto*
next

assume $\exists q \in plane.\ m \cap n = \{q\}$
then obtain q **where** $q \in plane \wedge m \cap n = \{q\}$
by *auto*
then have $\{p,d\} \subseteq \{q\}$
using $\langle c = d \rangle$ *assms* **by** *auto*
then show *False*
using $\langle c = d \rangle$ *assms*(9) **by** *auto*
qed
qed
qed

5.2.3 Interpretación mínimo modelo geometría simple

El mínimo modelo que tiene la geometría simple es considerar el plano como el conjunto formado por tres números $\{a, b, c\}$, ya pueden ser enteros,naturales etc y con ellos formar únicamente 3 líneas. En este caso serían las combinaciones que se pueden hacer de 2 elementos de un conjunto de 2, es decir, 3.

Para ello se va a dar el la definicion del **planes-3** que es el plano de 3 elementos y **lines-3** que es el conjunto formado por 3 líneas.

definition *plane-3* $\equiv \{1::nat,2,3\}$

definition *lines-3* $\equiv \{\{1,2\},\{2,3\},\{1,3\}\}$

interpretation *Simple-Geometry-smallest-model*:
Simple-Geometry plane-3 lines-3
apply *standard*
apply (*simp add*: *plane-3-def lines-3-def*)+
done

5.3 Geometría no proyectiva

5.3.1 Entorno local

La geometría no proyectiva es un tipo de geometría en el que asumimos paralelismo, en nuestro caso entre rectas.

Definición 5.3.1 *El paralelismo es una relación que se establece entre dos rectas cualesquiera del plano, esta relación dice que dos rectas son paralelas si bien son la misma recta o no comparten ningún punto, es decir, su intersección es vacía.*

Gracias a esta relación entre rectas, podemos definir un nuevo entorno local añadiendo al ya definido **Simple-Geometry** un nuevo axioma, el axioma de la existencia del paralelismo.

Parallels-Ex: sea p un punto del plano y l una línea. Si $p \notin l$ entonces debe existir una línea m tal que $p \in m$ y $l \cap m = \emptyset$.

Al nuevo entorno local lo denotaremos como **Non-Projective-Geometry**.

locale *Non-Projective-Geometry* =
Simple-Geometry +
assumes *parallels-Ex*:
$\forall p \in plane.\ \forall l \in lines.\ p \notin l \longrightarrow (\exists m \in lines.\ p \in m \wedge m \cap l = \{\})$

5.3.2 Proposiciones de geometría no proyectiva

A continuación vamos a presentar un lema sobre geometría no proyectiva:

Lema 5.3.2 *Es falso que todo par de líneas se cortan.*

Demostración: La demostración se hará por reducción al absurso. Es decir, supongamos que todo par de líneas se cortan.

Sea ahora $l1$ una línea obtenida por el el lema 5.2.1. Porr el axioma A5 obtenemos un punto $q1$ tal que $q1 \notin l1$. Usando el axioma **Parallels-Ex** aplicado al punto $q1$ y a la línea $l1$ obtenemos que existe una línea m tal que $q1 \in m$ y $m \cap l = \emptyset$. Por lo tanto, hemos llegado a una contradicción ya que se ha demostrado que existen dos líneas cuya intersección es vacía.

□

La formalización y demostración en Isabelle/Hol es la siguiente:

lemma (**in** *Non-Projective-Geometry*) *non-projective*:
$\neg(\forall r \in lines.\ \forall s \in lines.\ r \cap s \neq \{\})$
proof
assume *4*: $\forall r{\in}lines.\ \forall s{\in}lines.\ r \cap s \neq \{\}$
show *False*
proof −
obtain *l1* **where** *1*: $l1 \in lines$
using *one-line-exists* **by** *auto*
then obtain *q1* **where** *2*: $q1 \in plane \wedge q1 \notin l1$
using *A5* **by** *auto*
then have $q1 \notin l1 \longrightarrow (\exists m \in lines.\ q1 \in m \wedge m \cap l1 = \{\})$
using *1 parallels-Ex* **by** *simp*

then obtain *m1* **where** *3*: $m1 \in lines \wedge q1 \in m1 \wedge m1 \cap l1 = \{\}$
using *2* **by** *auto*
then obtain $m1 \cap l1 \neq \{\}$ **using** *1 4* **by** *auto*
then show *?thesis* **using** *3* **by** *auto*
qed
qed

5.3.3 Interpretacion modelo geometría no proyectiva

El mínimo modelo de la geometría no proyectiva es considerar que el plano tiene 4 elementos; es decir, considerar el plano como $\{a, b, c, d\}$ siendo estos números enteros,naturales etc. Con estos 4 elementos para que sea un modelo de la geometría no proyectiva hay que formar como mínimo 6 rectas.

Para ello vamos a dar la definicion **plane-4** que es el plano formado por 4 elementos y **lines-4** que son las líneas asociadas a estos elementos.

definition *plane-4* $\equiv \{1::nat, 2, 3, 4\}$

definition *lines-4* $\equiv \{\{1,2\},\{2,3\},\{1,3\},\{1,4\},\{2,4\},\{3,4\}\}$

interpretation *Non-projective-geometry-card-4*:
Non-Projective-Geometry plane-4 lines-4
apply *standard*
apply (*simp add*: *plane-4-def lines-4-def*)+
done

5.4 Geometría proyectiva

5.4.1 Entorno local

La geometría proyectiva es un tipo de geometría que se basa en que dado cualquier par de rectas su intersección siempre es un punto.

Para ello vamos a definir un nuevo entorno local **Projective-Geometry** tal que se basa en el entorno local ya definido **Simple-Geometry** añadiéndole dos axiomas más. Estos axiomas son los siguientes:

1. Cualquier par de líneas se cortan.
2. Toda línea tiene al menos 3 puntos.

El nuevo entorno local es el siguiente:

locale *Projective-Geometry* =
Simple-Geometry +
assumes *A6*: $\forall\, l \in lines.\ \forall\, m \in lines.\ \exists\, p \in plane.\ p \in l \wedge p \in m$
and *A7*: $\forall\, l \in lines.\ \exists\, x.\ card\ x = 3 \wedge x \subseteq l$

5.4.2 Proposiciones de geometría proyectiva

A continuación vamos a demostrar una serie de lemas dentro del entorno **Projective-Geometry**. Antes de los lemas vamos a demostrar en Isabelle que si un conjunto x tiene cardinalidad 3, entonces está formado por 3 puntos distintos. Este pequeño lema nos ayudará en las demostraciones de los siguientes.

lemma *construct-set-of-card3*:
$card\ x = 3 \Longrightarrow \exists$ *p1 p2 p3*. *distinct* [*p1*,*p2*,*p3*] $\wedge\ x = \{p1,p2,p3\}$
by (*metis card-eq-SucD distinct.simps*(2)
distinct-singleton list.set(1) *list.set*(2) *numeral-3-eq-3*)

Los dos primeros lemas que vamos a demostrar son versiones equivalentes al axioma A7 ya definido y en los dos se utilizará el dicho axioma.

Lema 5.4.1 *Para todo línea l, existen $p1$, $p2$, $p3$ tales que $\{p1, p2, p3\} \subseteq l$ y son distintos entre sí.*

Demostración: Sea l una línea cualquiera. Por el axioma A7 obtenemos que existe x tal que cardinalidad $x = 3$ y que $x \subseteq l$. Por el lema definido anteriormente, se obtiene que existen $p1$, $p2$, $p3$ distintos entre sí y tales que $x = \{p1, p2, p3\}$ y que $p1 \neq p2 \neq p3$.

□

La formalización y demostración en Isabelle/HOL es la siguiente:

lemma (**in** *Projective-Geometry*) *A7a*:
$\forall\, l \in lines.\ \exists\, p1\ p2\ p3.\ \{p1, p2, p3\} \subseteq plane \wedge$
$distinct\ [p1, p2, p3] \wedge$
$\{p1, p2, p3\} \subseteq l$
proof
fix *l*
assume *1*: $l \in lines$
show $\exists\, p1\ p2\ p3.\ \{p1, p2, p3\} \subseteq plane \wedge$
$distinct\ [p1, p2, p3] \wedge$
$\{p1, p2, p3\} \subseteq l$
proof −

```
  obtain x where 2: card x = 3 ∧ x ⊆ l
    using 1 A7 by auto
  then have 3: card x = 3
    by (rule conjE)
  have ∃ p1 p2 p3. distinct [p1, p2, p3] ∧ x = {p1, p2, p3}
    using 3 by (rule construct-set-of-card3)
  then obtain p1 p2 p3
    where 4 :distinct [p1,p2,p3] ∧ x = {p1, p2, p3}
    by auto
  obtain l ⊆ plane ∧ l ≠ {}
    using 1 A2  by auto
  then have
    {p1, p2, p3} ⊆ plane ∧ distinct [p1, p2, p3] ∧ {p1, p2, p3} ⊆ l
    using 4 2 by auto
  then show ?thesis
    by auto
 qed
qed
```

Lema 5.4.2 *Sea l una linea y p, q dos puntos de l. Entonces existe un punto r tal que $r \neq p$, $r \neq q$ y $r \in l$.*

Demostración: Sea l una linea y p, q dos puntos tales que $\{p, q\} \subseteq l$. Por el axioma $A7$ se tiene que existe x tal que la cardinalidad $x = 3$ y $x \subseteq l$. Por el lema demostrado anteriormente, se tiene que existen $p1$, $p2$, $p3$ distintos entre sí y tales que $\{p1, p2, p3\} \subseteq l$. Luego, usando las hipótesis se tiene 3 posibilidades:

1. Si $p1 \notin p, q$ entonces ya tendríamos probado el lema.
2. Si $p2 \notin p, q$ entonces ya tendríamos probado el lema.
3. Si $p3 \notin p, q$ entonces ya tendríamos probado el lema.

En cualquiera de los 3 casos ya se tendría probado el lema.
□ La formalización y demostración en Isabelle/HOL es la siguiente:

```
lemma (in Projective-Geometry) A7b:
 assumes l ∈ lines
   {p, q} ⊆ l
 shows  ∃ r ∈ plane. r ∉ {p, q} ∧ r ∈ l
proof −
```

```
 obtain x where 1: card x = 3 ∧ x ⊆ l
  using assms A7 by auto
 then have card x = 3
  by (rule conjE)
 then have ∃ p1 p2 p3. distinct [p1,p2,p3] ∧ x = {p1,p2,p3}
  by (rule construct-set-of-card3)
 then obtain p1 p2 p3
  where 2: distinct [p1,p2,p3] ∧ x = {p1,p2,p3}
  by auto
 have l ⊆ plane ∧ l ≠ {}
  using A2 assms by auto
 then have 3: x ⊆ plane
  using 1 by auto
 then have p1 ∉ {p,q} ∨ p2 ∉ {p,q} ∨ p3 ∉ {p,q}
  using 2 by auto
 then show ?thesis
  using 1 2 3 by auto
qed
```

Lema 5.4.3 *Para todo punto del plano existen dos líneas distintas que pasan por él.*

Demostración: Sea p un punto del plano cualquiera. Por el axioma $A3$ obtenemos que existe una línea l tal que $\{p, p\} \subseteq l$. Luego, por el axioma $A5$, se obtiene un punto r tal que $r \notin l$. Por lo tanto, por el axioma $A3$ de nuevo, se obtiene otra recta m tal que $\{p, r\} \subseteq m$. Ya se tiene probada la existencia de las dos rectas que pasan por el punto p, para probar que son diferentes simplemente se usa que $r \in m$ y $r \notin l$.

□

La formalización y demostración en Isabelle/HOL es la siguiente:

```
lemma (in Projective-Geometry) two-lines-per-point:
 ∀ p ∈ plane. ∃ l ∈ lines. ∃ m ∈ lines. l ≠ m ∧ p ∈ l ∩ m
proof
 fix p
 assume 1: p ∈ plane
 show ∃ l ∈ lines. ∃ m ∈ lines. l ≠ m ∧ p ∈ l ∩ m
 proof −
  obtain l where 2: l ∈ lines ∧ {p,p} ⊆ l
   using A3 1 by auto
  then obtain r where 3: r ∉ l ∧ r ∈ plane
   using A5 by auto
```

```
  then obtain m where 4: m ∈ lines ∧ {p,r} ⊆ m
   using A3 1  by auto
  then have l ≠ m ∧ p ∈ l ∩ m
   using  2 3 by auto
  then show ?thesis
   using 2 4 by auto
 qed
qed
```

Para el próximo lema se va a usar el siguiente lema auxiliar.

Lema 5.4.4 *[Lema auxiliar 1] Sea l una línea y r, s dos puntos tales que $\{r, s\} \subseteq l$. Sea también $l2$ otra línea y p otro punto tal que $\{p, r\} \subseteq l2$. Entonces, si $p \neq r$ y $s \notin l2$ se tiene que $p \notin l$.*

Demostración: La demostración se hará por reducción al absurdo; es decir, supongamos $p \in l$ y se llegará a una contradicción.

Supongamos que $p \in l$. Primero obtenemos que como $s \in l$ y $s \notin l2$ entonces $l \neq l2$. Usando esto último, obtenemos del axioma $A4$ que $l \cap l2 = \emptyset$ o $\exists q$ punto tal que $l \cap l2 = \{q\}$

1. Supongamos que $l \cap l2 = \emptyset$, pero por hipótesis se tiene que $r \in l$ y $r \in l2$. Luego se llega a una contradicción.

2. Supongamos que existe q tal que $l \cap l2 = \{q, \}$. Sin embargo, como hemos supuesto que $p \in l$ y, además, $r \in l$, $r \in l2$, $p \in l2$ y $r \neq p$ se llega a una contradicción.

En los dos casos hemos llegado a una contradicción luego se tiene que $p \notin l$.

□

La formalización y demostración en Isabelle/HOL del lema auxiliar es la siguiente:

```
lemma (in Projective-Geometry) punto-no-pertenece:
 assumes l2 ∈ lines ∧ {p,r} ⊆ l2
      l ∈ lines ∧  {r,s} ⊆ l
      p ≠ r
      s ∉ l2
     shows p ∉ l
proof
 assume 1:p ∈ l
 have l ∩ l2 = {} ∨ (∃ q ∈ plane. l ∩ l2 = {q})
  using A4 assms(1,2,4) by auto
 then show False
```

```
 proof
  assume l ∩ l2 = {}
  then show False
   using assms(1,2) by auto
 next
  assume  ∃q∈plane. l ∩ l2 = {q}
  then obtain t where l ∩ l2 = {t}
   by auto
  then have {p,r} ⊆ {t}
   using assms(1,2) 1  by auto
  then show False
   using assms(3) by auto
 qed
qed
```

El lema a demostrar es el siguiente:

Lema 5.4.5 *Para todo punto p existe una línea l tal que* $p \notin l$.

Demostración: Sea p un punto cualquiera, por el axioma $A3$ obtenemos una línea $l1$ tal que $\{p, p\} \subseteq l1$. Usando el axioma $A5$ se obtiene un punto r tal que $r \notin l1$. De nuevo usando el axioma $A3$ obtenemos una línea $l2$ tal que $\{p, r\} \subseteq l2$. Repitiendo el mismo razonamiento, usamos el axioma $A5$ para obtener un punto s tal que $s \notin l2$ y por el axioma $A3$ una línea l tal que $\{r, s\} \subseteq l$. Por último, usando que $r \notin l1$ se tiene que $p \neq r$ y, por lo tanto, se tienen todas las hipótesis del lema auxiliar 5.4.4, luego se ha demostrado que existe una línea l tal que $p \notin l$.

□

La formalizacion y demostración en Isabelle/HOL es la siguiente:

```
lemma (in Projective-Geometry) external-line:
 ∀p ∈ plane. ∃l ∈ lines. p ∉ l
proof
 fix p
 assume 1: p ∈ plane
 show ∃l ∈ lines. p ∉ l
 proof −
  obtain l1 where 2: l1 ∈ lines ∧ {p,p} ⊆ l1
   using 1 A3 by auto
  then obtain r where 3: r ∈ plane ∧ r ∉ l1
   using A5 by auto
  obtain l2 where 4: l2 ∈ lines ∧ {p,r} ⊆ l2
```

```
    using 1 3 A3 by auto
  then obtain s where 5: s ∈ plane ∧ s ∉ l2
    using A5 3 by auto
  obtain l where 6: l ∈ lines ∧ {r,s} ⊆ l
    using 3 5 A3 by auto
  have p ≠ r using 2 3 by auto
  then have p ∉ l
    using 4 6 5 punto-no-pertenece [of l2 p r l s] by simp
  then show ?thesis
    using 6 by auto
 qed
qed
```

Para el próximo lema, se va a usar el siguiente lema auxiliar:

Lema 5.4.6 *Sean $l, l1, l2$ líneas tales que existen puntos p, q, r tal que $\{p, r\} \subseteq l$, $\{p, q\} \subseteq l1$ $\{r, q\} \subseteq l2$ y, además, $l \neq l1$ y $p \neq r$. Entonces se tiene que $l1 \neq l2$.*

Demostración: La demostración se hará por reducción al absurdo, es decir, supongamos que $l1 = l2$ y se llegará a una contradicción.

Supongamos que $l1 = l2$. Como por hipótesis se tiene que $l \neq l1$ entonces usando el axioma $A4$ obtenemos que $l \cap l1 = \emptyset$ o existe un punto tal que $l \cap l1 = \emptyset$. Veamos los dos casos.

1. Supongamos que $l \cap l1 = \emptyset$. Como por hipótesis se tiene que $p \in l$ y $p \in l1$ entonces se llega a un absurdo.
2. Supongamos que existe un punto t tal que $l \cap l1 = \{t\}$. Como se había supuesto que $l1 = l2$ se tiene que, usando las hipótesis, $\{p, r\} \subseteq \{t, \}$. Sin embargo, como $p \neq r$ entonces se llega a un absurdo.

En ambos casos hemos llegado a un absurdo, luego $l1 \neq l2$.

□

Su demostración y formalización en Isabelle/HOL es la siguiente:

```
lemma (in Projective-Geometry) lineas-diferentes:
 assumes l ∈ lines ∧ {p,r} ⊆ l
      l1 ∈ lines ∧ {p,q} ⊆ l1
      l2 ∈ lines ∧ {r,q} ⊆ l2
      l1 ≠ l
      p ≠ r
```

```
 shows  l1 ≠ l2
proof
 assume 1:l1 = l2
 have l ∩ l1 = {} ∨ (∃ q ∈ plane. l ∩ l1 = {q})
  using A4 assms(1,2,4) by auto
 then show False
 proof
  assume l ∩ l1 = {}
  then show False
   using assms(1,2) by auto
 next
  assume ∃ q∈plane. l ∩ l1 = {q}
  then obtain t where l ∩ l1 = {t}
   by auto
  then have {p,r} ⊆ {t}
   using assms(1,2,3) 1 by auto
  then show False
   using assms(5) by auto
 qed
qed
```

Lema 5.4.7 *Para todo punto p en el plano, existen al menos tres líneas que pasan por p.*

Demostración: Sea p un punto del plano, usando el lema 5.4.5 se obitene que una línea h tal que $p \notin h$. Usando la definición equivalente del axioma $A7$ (lema 5.4.1) se obtienen tres puntos a, b, c distintos entre sí y tales que $\{a, b, c\} \subseteq h$. Por lo tanto, usando el axioma $A3$ obtenemos de forma equivalente tres líneas l, m, n tales que $\{a, p\} \subseteq l$, $\{b, p\} \subseteq m$, $\{c, p\} \subseteq n$. Ya hemos probado que existen 3 líneas que verifican las condiciones, lo único que queda por probar es que sean diferente. Usando el lema auxiliar 5.4.6 se concluye la prueba.

□

La siguiente figura 5.3 muestra una visión geométrica de la demostración anterior.

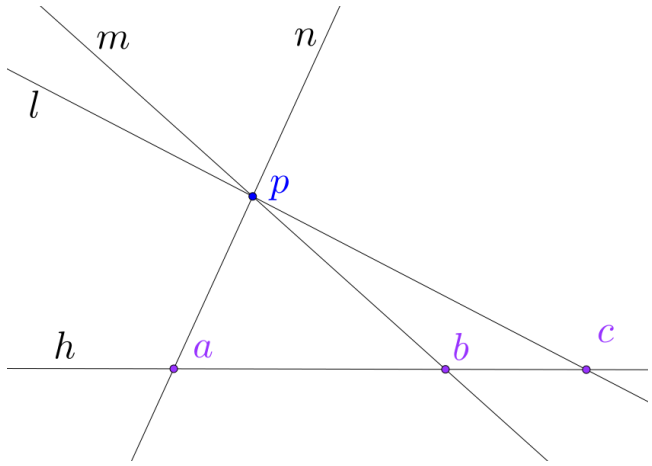

Figura 5.3: Visión geométrica de la demostración del lema 5.3

La formalización y demostración en Isabelle/HOL es la siguiente:

lemma (**in** *Projective-Geometry*) *three-lines-per-point*:
 $\forall p \in plane. \exists l\ m\ n.$
  distinct [*l*,*m*,*n*] $\wedge$ {*l*,*m*,*n*} $\subseteq$ *lines* $\wedge$ $p \in l \cap m \cap n$
proof
 fix *p*
 assume *1*: $p \in plane$
 show $\exists l\ m\ n.$ *distinct* [*l*,*m*,*n*] $\wedge$ {*l*,*m*,*n*} $\subseteq$ *lines* $\wedge$ $p \in l \cap m \cap n$
 proof −
  obtain *h* **where** *2*: $h \in lines \wedge p \notin h$
   using *1 external-line* **by** *auto*
  then obtain *a b c*
   where *3*: {*a*,*b*,*c*} $\subseteq$ *plane* $\wedge$ *distinct* [*a*,*b*,*c*] $\wedge$ {*a*,*b*,*c*} $\subseteq$ *h*
   using *A7a* **by** *auto*
  then obtain *l* **where** *4*: $l \in lines \wedge$ {*a*,*p*} $\subseteq$ *l*
   using *1 A3* **by** *auto*
  obtain *m* **where** *5*: $m \in lines \wedge$ {*b*,*p*} $\subseteq$ *m*
   using *1 3 A3* **by** *auto*
  obtain *n* **where** *6*: $n \in lines \wedge$ {*c*,*p*} $\subseteq$ *n*
   using *1 3 A3* **by** *auto*
  have *7*:$h \neq l$
   using *4 2* **by** *auto*
  have $a \neq b$
   using *3* **by** *auto*
  then have *9*: $m \neq l$
   using *3 4 5 2 7 lineas-diferentes* [*of h a b l p m*] **by** *simp*

```
  have 8: h ≠ m
   using 5 2 by auto
  have  b ≠ c
   using 3 by auto
  then have 10: m ≠ n
   using 6 5 3 2 8 lineas-diferentes [of h b c m p n ] by simp
  have a ≠ c
   using 3 by auto
  then  have 11: l ≠ n
   using 2 3 4 6 7 lineas-diferentes [of h a c l p n ] by simp
  show ?thesis
   using  4 5 6 9 10 11 by auto
 qed
qed
```

Para el siguiente lema se va a usar el siguiente lema auxiliar:

Lema 5.4.8 *Sea l y $l1$ líneas tales que $l \neq l1$ y existen puntos p, q, c tales que $\{p, c\} \subseteq l$ y $\{q, c\} \subseteq l1$ con $c \neq p$. Entonces $p \neq q$*

Demostración: La demostración se hará por reducción al absurdo, es decir, supongamos que $p = q$ y se llegará a un absurdo.

Supongamos que $p = q$. Entonces usando la hipótesis $l \neq l1$ y el axioma $A4$ se obtiene que $l \cap l1 = \emptyset$ o existe un punto tal que $l \cap l1 = \{q\}$. Veamos los dos casos por separado.

1. Supongamos que $l \cap l1 = \emptyset$. Como por hipótesis se tiene que $c \in l$ y $c \in l1$ entonces se llega a un contradicción.

2. Supongamos que existe t tal que $l \cap l1 = \{t\}$. Sin embargo, como hemos supuesto que $p = q$, se tiene que $\{p, c\} \subseteq \{t\}$. Pero como $p \neq c$ se llega a una contradicción.

En los dos casos se ha llegado a una contradicción luego se tiene que $p \neq q$.

□

La formalización y demostración en Isabelle/HOL es la siguiente:

```
lemma (in Projective-Geometry) puntos-diferentes:
 assumes l ∈ lines
      l1 ∈ lines
      {p,c} ⊆ l
      {q,c} ⊆ l1
```

```
      l ≠ l1
      c ≠ p
    shows p ≠ q
proof
 assume 1: p = q
 have l ∩ l1 = {} ∨ (∃ q ∈ plane. l ∩ l1 = {q})
  using assms(1,2,5) A4 by auto
 then show False
 proof
  assume l ∩ l1 = {}
  then show False
   using assms(3,4) by auto
 next
  assume ∃ q∈plane. l ∩ l1 = {q}
  then obtain t where l ∩ l1 = {t}
   by auto
  then have {p,c} ⊆ {t}
   using assms(3,4) 1 by auto
  then show False
   using assms(6) by auto
 qed
qed
```

Lema 5.4.9 *Existen al menos 7 puntos diferenetes en el plano.*

Demostración: Primero sea l una línea que se obtiene usando el lema 5.2.1, usando el lema equivalente al axioma $A7$ (lema 5.4.1) se obtienen 3 puntos $p1, p2, p3$ distintos entre sí tales que $\{p1, p2, p3\} \subseteq l$. Ahora usando el axioma $A5$ se obtiene que existe q tal que $q \notin l$, luego ya se tienen probado que existen 4 puntos diferentes, ya que q es diferente del resto porque $q \notin l$. Consideremos ahora tres líneas $l1, l2, l3$ obtenidas por el axioma $A3$ tales que $\{p1, q\} \subseteq l1$, $\{p2, q\} \subseteq l2$, $\{p3, q\} \subseteq l3$. Ahora vamos a obtener los tres puntos restantes, usando que $l \neq l1 \neq l2 \neq l3$ gracias al lema auxiliar 5.4.6:

1. Obtenemos $p4$ tal que $p4 \in l1$ y $p4 \notin \{p1, q\}$ usando el lema equivalente al axioma $A7$ (lema 5.4.2). Entonces veamos que $p4$ es diferente al resto de los puntos. Se tiene que $p4 \notin \{p1, q\}$, luego veamos que $p4 \neq p2$ y $p4 \neq p3$. Usando el lema auxiliar 5.4.8 se tiene probado. Luego hemos probado que $p4$ es diferente al resto de los puntos.

2. Obtenemos $p5$ tal que $p5 \in l2$ y $p5 \notin \{p2, q\}$ usando el lema equivalente al axioma $A7$ (lema 5.4.2). Entonces veamos que $p5$ es diferente al resto de los puntos, ya se

tiene que $p5 \notin \{p2, q\}$ luego falta por probar que $p5 \notin \{p1, p3, p4\}$. Sin embargo es inmediato comprobar que se verifica usando el lema auxiliar **??**, luego ya hemos probado que $p5$ es diferente al resto de los puntos.

3. Obtenemos $p6$ tal que $p6 \in l3$ y $p6 \notin \{p3, q\}$ usando el lema equivalente al axioma $A7$ (lema 5.4.2). Entonces veamos que $p6$ es diferente al resto de los puntos, ya se tiene que $p6 \notin \{p3, q\}$ luego falta por probar que $p6 \notin \{p1, p2, p4, p5\}$. Sin embargo es inmediato comprobar que se verifica usando el lema auxiliar **??**, luego ya hemos probado que $p6$ es diferente al resto de los puntos.

Por lo tanto, ya tenemos probado la existencia y la disparidad de 7 puntos: $\{p1, p2, p3, p4, p5, p6,$ □

La siguiente figura 5.4 muestra una intuición geométrica de la demostración del lema 5.4.9.

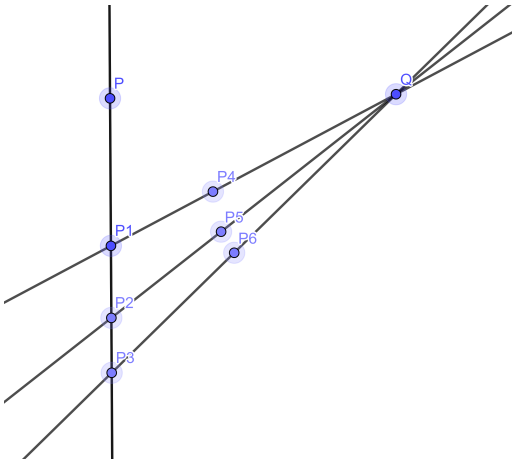

Figura 5.4: Visión geométrica de la demostración del lema 5.4.9

La formalización y demostración del lema en Isabelle/HOL es la siguiente:

lemma (**in** *Projective-Geometry*) *at-least-seven-points*:
 ∃ *p1 p2 p3 p4 p5 p6 p7*.
  distinct [*p1*,*p2*,*p3*,*p4*,*p5*,*p6*,*p7*] ∧ {*p1*,*p2*,*p3*,*p4*,*p5*,*p6*,*p7*} ⊆ *plane*
proof −
 obtain *l* **where** *1*: *l* ∈ *lines*
  using *one-line-exists* **by** *auto*
 then obtain *x* **where** *2*: *card x* = *3* ∧ *x* ⊆ *l*
  using *A7* **by** *auto*
 then have *card x* = *3*
  by (*rule conjE*)

then obtain *p1 p2 p3*
where *3*: *distinct* [*p1,p2,p3*] $\wedge$ $x = \{p1,p2,p3\}$
using *construct-set-of-card3* [*of x*] **by** *auto*
then have *4*: $\{p1,p2,p3\} \subseteq l$
using *2* **by** *auto*
then have *5*: $\{p1,p2,p3\} \subseteq plane$
using *A2 1* **by** *auto*
obtain *q* **where** *6*: $q \in plane \wedge q \notin l$
using *A5 1* **by** *auto*
then have *7*: *distinct* [*p1,p2,p3,q*]
using *3 4* **by** *auto*
obtain *l1* **where** *8*: $l1 \in lines \wedge \{p1,q\} \subseteq l1$
using *5 6 A3* **by** *auto*
then have *9*: $l1 \neq l$
using *6* **by** *auto*
obtain *p4* **where** *10*: $p4 \notin \{p1,q\} \wedge p4 \in l1$
using *A7b* [*of l1 p1 q*] *8* **by** *auto*
have *11*: $p4 \neq p2$
using *3 4 1 6 2 10 8 puntos-diferentes* [*of l1 l p4 p1 p2*] **by** *auto*
have *12*: $p4 \neq p3$
using *7 1 9 4 10 8 puntos-diferentes* [*of l1 l p4 p1 p3*] **by** *auto*
obtain *l2* **where** *13*: $l2 \in lines \wedge \{p2,q\} \subseteq l2$
using *5 6 A3* **by** *auto*
then obtain *p5* **where** *14*: $p5 \notin \{p2,q\} \wedge p5 \in l2$
using *A7b* [*of l2 p2 q*] *7* **by** *auto*
have *15*: $l2 \neq l$
using *6 13* **by** *auto*
have *16*: $p5 \neq p1$
using *1 13 14 4 15 puntos-diferentes* [*of l l2 p1 p2 p5*]
by *auto*
have *17*: $p5 \neq p3$
using *1 13 14 4 15 puntos-diferentes* [*of l l2 p3 p2 p5*]
by *auto*
have *20*: $l1 \neq l2$
using *1 9 13 4 8 7 lineas-diferentes* [*of l p1 p2 l1 q l2*]
by *simp*
have *21*: $p4 \neq p5$
using *13 8 14 4 10 20 puntos-diferentes* [*of l1 l2 p4 q p5*]
by *auto*
obtain *l3* **where** *22*: $l3 \in lines \wedge \{p3,q\} \subseteq l3$

```
  using A3 5 6 by auto
 then obtain p6 where 23: p6 ∉ {p3,q} ∧ p6 ∈ l3
  using A7b by metis
 have 25: p6 ≠ p1
  using 1 22 6 4  23 puntos-diferentes [of l3 l p1 p3 p6]
  by auto
 have 26: p6 ≠ p2
  using 1 22 6 23 4 puntos-diferentes [of l3 l p2 p3 p6]
  by auto
 have 29: l1 ≠ l3
  using 1 4 9 22 8 7 lineas-diferentes [of l p1 p3 l1 q l3]
  by simp
 have 31: p6 ≠ p4
  using 22 8 10 23  29 puntos-diferentes [of l1 l3 p4 q p6]
  by auto
 have 34: l2 ≠ l3
  using 1 4 13 22 15 7 lineas-diferentes [of l p2 p3 l2 q l3]
  by simp
 have 35: p6 ≠ p5
  using 22 13 23 14 34 puntos-diferentes [of l2 l3 p5 q p6]
  by auto
 moreover have distinct [p1,p2,p3,p4,p5,p6,q]
  using 7 10 11 12 14 16 17 21 23 25 26 31 7 35 by auto
 moreover have {p1,p2,p3,p4,p5,p6,q} ⊆ plane
  using 6 5  A2 10 8 14 13 22 23 by auto
 ultimately show ?thesis
  by blast
qed
```

5.4.3 Interpretación modelo geometría proyectiva

El mínimo modelo que presenta la Geometria Proyectiva es considerar que el plano tiene 7 puntos y con ellos formar como mínimo 7 líneas. Este modelo se conoce como el **plano de Fano** que es el plano proyectivo con el menor número de puntos y líneas necesarios para que se verifiquen todos los axiomas. Para ello vamos a dar la definición en Isabelle del plano de 7 elementos **plane-7** y la definición **lines-7** asociado a sus 7 líneas.

La siguiente figura 5.5 muestra una visión del **plano de Fano**:

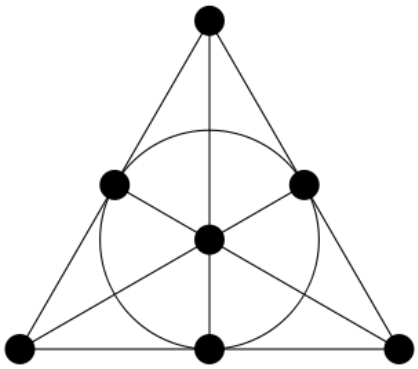

Figura 5.5: Visión geométrica del plano de Fano

definition *plane-7* $\equiv \{1::nat,2,3,4,5,6,7\}$

definition *lines-7* $\equiv \{\{1,2,3\},\{1,6,5\},\{3,4,5\},\{5,7,2\},\{3,7,6\},$
$\{1,4,7\},\{2,4,6\}\}$

Para poder demostrar la existencia de este modelo mínimo con el comando **interpretation** es necesario definir los siguientes lemas auxiliares:

lemma *aux1a*: $card\ \{Suc\ 0, 2, 3\} = 3$
by *auto*

lemma *aux1*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{Suc\ 0, 2, 3\}$
using *aux1a* **by** *blast*

lemma *aux2a*: $card\ \{Suc\ 0, 6, 5\} = 3$
by *auto*

lemma *aux2*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{Suc\ 0, 6, 5\}$
using *aux2a* **by** *blast*

lemma *aux3a*: $card\ \{3::nat, 4, 5\} = 3$
by *auto*

lemma *aux3*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{3::nat, 4, 5\}$
using *aux3a* **by** *blast*

lemma *aux4a*: $card\ \{5::nat, 7, 2\} = 3$

by *auto*

lemma *aux4*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{5::nat, 7, 2\}$
using *aux4a* **by** *blast*

lemma *aux5a*: $card\ \{3::nat, 7, 6\} = 3$
by *auto*

lemma *aux5*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{3::nat, 7, 6\}$
using *aux5a* **by** *blast*

lemma *aux6a*: $card\ \{Suc\ 0, 4, 7\} = 3$
by *auto*

lemma *aux6*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{Suc\ 0, 4, 7\}$
using *aux6a* **by** *blast*

lemma *aux7a*: $card\ \{2::nat, 4, 6\} = 3$
by *auto*

lemma *aux7*: $\exists\, x.\ card\ x = 3 \wedge x \subseteq \{2::nat, 4, 6\}$
using *aux7a* **by** *blast*

interpretation *Projective-Geometry-smallest-model*:
Projective-Geometry plane-7 lines-7
apply *standard*
apply (*simp add*: *plane-7-def lines-7-def*)+
apply (*intro conjI*)
apply (*rule aux1*)
apply (*rule aux2*)
apply (*rule aux3*)
apply (*rule aux4*)
apply (*rule aux5*)
apply (*rule aux6*)
apply (*rule aux7*)
done

end

Apéndice A

Lemas de HOL usados

En este apéndice se recogen la lista de los lemas usados en el trabajo indicando la página del libro de HOL donde se encuentra.

A.1 Las bases de lógica de orden superior (2)

A.1.1 Lógica primitiva (2.1)

A.1.1.1 Axiomas y definiciones básicas (2.1.4)

- (p. 36) $\dfrac{(P \longrightarrow Q) \wedge P}{Q}$ (*mp*)
- (p. 36) $\dfrac{\dfrac{P}{Q}}{P \longrightarrow Q}$ (*impI*)
- (p. 36) $\dfrac{\bigwedge x.\, f\, x = g\, x}{f = g}$ (*ext*)
- (p. 36) $\dfrac{s = t \wedge P\, s}{P\, t}$ (*subst*)

A.1.2 Reglas fundamentales (2.2)

A.1.2.1 Reglas de congruencia para aplicaciones (2.2.2)

- (p. 37) $\dfrac{f = g}{f\,x = g\,x}$ (*fun-cong*)

A.1.2.2 Igualdades de booleanos (2.2.3)

- (p.38) $\dfrac{Q \wedge P = Q}{P}$ (*rev-iffD2*)

A.1.2.3 Cuantificadores universales(1) (2.2.5)

- (p. 38) $\dfrac{\forall\, x.\, P\,x \qquad \dfrac{P\,x}{R}}{R}$ (*allE*)

A.1.2.4 Negación (2.2.7)

- (p. 39) $\dfrac{\neg\, P \wedge P}{R}$ (*notE*)

A.1.2.5 Implicación (2.2.8)

- (p. 40) $\dfrac{t \neq s}{s \neq t}$ (*not-sym*)

A.1.2.6 Derivación de iffI (2.2.10)

- (p. 40) $\dfrac{\dfrac{P}{Q} \qquad \dfrac{Q}{P}}{P = Q}$ (*iffI*)

A.1.2.7 Cuantificadores universales(2) (2.2.12)

- (p. 41) $\dfrac{\bigwedge x.\ P\ x}{\forall\, x.\ P\ x}$ (*allI*)

A.1.2.8 Cuantificadores existenciales (2.2.13)

- (p. 41) $\dfrac{\exists\, x.\ P\ x \qquad \bigwedge x.\ \dfrac{P\ x}{Q}}{Q}$ (*exE*)

A.1.2.9 Conjunciones (2.2.14)

- (p. 42) $\dfrac{\dfrac{\neg\ P}{False}}{P}$ (*ccontr*)

A.2 Órdenes abstractos (4)

A.2.1 Monotonicidad (4.9)

- (p. 95) $\dfrac{mono\, f \wedge x \leq y}{f\, x \leq f\, y}$ (*monoD*)

A.2.2 Nombres duplicados (4.17)

- (p. 107) $\dfrac{x \leq y \wedge y \leq x}{x = y}$ (*order-antisym*)

A.3 Grupos (5)

A.3.1 Soporte para razonar sobre signos (5.7)

- (p.204) $(a \leq a + b) = ((0::'a) \leq b)$ (*le-add-same-cancel1*)
- (p. 204) $(a \leq b + a) = ((0::'a) \leq b)$ (*le-add-same-cancel2*)

A.4 Retículos abstractos (6)

A.4.1 Retículos concretos (6.3)

- (p. 140) $\dfrac{\bigwedge x.\ \dfrac{x \in A}{z \leq x}}{z \leq Inf\ A}$ (*Inf-greatest*)

A.5 Teoría de conjuntos para lógica de orden superior(7)

A.5.1 Conjuntos como prediados

- (p. 157) $\dfrac{a \in \{x \mid P\ x\}}{P\ a}$ (*CollectD*)
- (p. 157) $\dfrac{P\ a}{a \in \{x \mid P\ x\}}$ (*CollectI*)
- (p. 157) $\dfrac{a \in \{x \mid P\ x\} \qquad \dfrac{P\ a}{PROP\ W}}{PROP\ W}$ (*CollectE*)

A.5.2 Operaciones básicas (7.3)

A.5.2.1 Conjunto vacío (7.3.3

- (p. 206) $Ball\ \emptyset\ P = True$ (*ball-empty*)

A.5.2.2 Aumentando un conjunto (7.3.10)

- (p. 170) $(a \in \{b\} \cup A) = (a = b \vee a \in A)$ (*insert-iff*)

A.6 Nociones sobre funciones (9)

A.6.1 El operador composición (9.2)

- (p. 199) $(f \circ g)\ x = f\ (g\ x)$ (*comp-apply*)

A.6.2 Inyectividad y biyectividad (9.5)

- (p. 213) $\dfrac{inj\, f \wedge f\, x = f\, y}{x = y}$ (*injD*)

A.6.3 Actualización de funciones (9.6)

- (p. 213) $\dfrac{z \neq x}{(f(x := y))\ z = f\, z}$ (*fun-upd-other*)
- (p. 213) $(f(x := y))\ x = y$ (*fun-upd-same*)

A.7 Retículos completos (10)

A.7.1 Retículos completos abstractos (10.3)

- (p. 220) $\dfrac{x \in A}{Inf\, A \leq x}$ (*Inf-lower*)

A.8 Números naturales (16)

A.8.1 Operaciones aritméticas (16.3)

- (p. 348) $0 * n = 0$ (*mult-0*)
- (p. 348) $Suc\ m * n = n + m * n$ (*mult-Suc*)
- (p. 348) $m * Suc\ n = m + m * n$ (*mult-Suc-right*)
- (p. 348) $m * 0 = 0$ (*mult-0-right*)

A.9 Conjuntos finitos(18)

A.9.1 Predicados para conjuntos finitos

- (p. 419) $\dfrac{finite\ F \qquad P\ \emptyset \qquad \bigwedge x\ F.\ \dfrac{finite\ F \wedge x \notin F \wedge P\ F}{P\ (\{x\} \cup F)}}{P\ F}$ (*finite-induct*)

A.10 Método de prueba Meson (37)

A.10.1 Forma de negación normal (37.1)

- (p. 740) $\dfrac{\nexists x.\ P\ x}{\forall x.\ \neg\ P\ x}$ (*Meson.not-exD*)
- (p. 740) $\dfrac{\neg\ (\forall x.\ P\ x)}{\exists x.\ \neg\ P\ x}$ (*Meson.not-allD*)

Bibliografía

[1] *Isabelle/HOL — Higher-Order Logic*. 2019. En http://isabelle.in.tum.de/website-Isabelle2019/dist/library/HOL/HOL/document.pdf.

[2] Reuben Albert, A. Adrian; Sandler. *An introduction to Finite Projective Planes*. New York: Holt, Rinehart and Winston, 1968.

[3] José A. Alonso. Temas de "Lógica matemática y fundamentos (2018–19)". Technical report, Univ. de Sevilla, 2019. En https://www.cs.us.es/~jalonso/cursos/lmf-18/temas.php.

[4] H. A. Priestly B.A. Davey. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

[5] Allan Clark. *Elements of abstract algebra*. Dover New York, 1984.

[6] F.Wiedijk. *Formalizing 100 theorems*. En http://www.cs.ru.nl/~freek/100/.

[7] F.Wiedijk. *The Seventeen Provers of the World*. 2006. En http://cs.ru.nl/~freek/comparison/comparison.pdf.

[8] Paul Halmos. *Naive Set Theory*. Van Nostrand, 1960. Reprinted by Springer-Verlag, Undergraduate Texts in Mathematics, 1974.

[9] J.W.P. Hirshfeld. *Projective Geometries Over Finite Fields*. Clarendon Press, 1998.

[10] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A proof assistant for Higher–Order Logic*. Lecture Notes in Computer Science, Vol. 2283, Springer–Verlag, 2019. En https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/Isabelle2019/doc/tutorial.pdf.

[11] Michael Rathejn and Wilfried Sieg. *Proof Theory*. The Stanford Encyclopedia, 2018. En https://plato.stanford.edu/archives/fall2018/entries/proof-theory/.

[12] H. Jerome Keisler; Joel Robbin. *Mathematical Logic and Computability*. McGraw-Hill, 1996.

[13] Steven Roman. *Lattices and Ordered Sets*. Springer Science

[14] Makarius Wenzel. *The Isabelle/Isar Reference Manual*. 2020. En http://isabelle.in.tum/doc/isar-ref.pdf.

[15] Harold E. Wolfe. *Introduction to Non-Euclidean Geometry*. Courier Corporation, 2013.