1 SISTEMAS FORMALES / LÓGICOS

Un *sistema* es una realidad en la que ciertos elementos se agrupan por alguna razón que depende de un observador. Por ejemplo, se habla del sistema solar, del sistema bancario, de sistemas de ecuaciones, etc. Cuando se trata de entender una realidad, se puede experimentar directamente para descubrir cómo se interactúa con ella. Muchas acciones culturales, como la lengua materna, caminar, nadar, etc., son aprendidas y entendidas así. Sin embargo, cuando se pretende entender una realidad, de manera que se pueda explicar y razonar sobre ella, es necesario establecer un lenguaje que permita denotar sus objetos y sus eventuales cambios.

Al hablar de un sistema de manera coloquial o informal se puede caer en equivocaciones debidas a una mala comprensión de lo que constituye el sistema y de su dinamismo. Para evitar esas eventuales fallas de comprensión o de uso de los sistemas se requiere un tratamiento *formal* (donde la forma importa), con lo cual se determina con claridad qué hace parte del sistema y qué no, y se entiende claramente cómo cambia lo que puede cambiar.

Las matemáticas pueden considerarse como un lenguaje de descripción y de discusión para hablar sobre sistemas. De hecho, las matemáticas sirven para definir *sistemas formales*. Se verá que un sistema formal no es otra cosa que un conjunto de elementos simbólicos que pueden manipularse de manera adecuada de suerte que reflejen una realidad deseada.

1.1 SISTEMAS FORMALES

Tras de un sistema formal hay, siempre, una realidad que se modela. La pretensión es que el formalismo se comporte a imagen y semejanza de la realidad modelada.

Una manera rápida de explicar el concepto de *sistema formal* es mediante la "ecuación" [Woo1988]

Sistema formal = Lenguaje formal + Aparato deductivo.

Con esto se quiere significar que un sistema formal consta de dos partes bien definidas:

- un *lenguaje formal*, con el que se denotan los elementos de la realidad modelada, y
- un *aparato deductivo*¸ que sirve para establecer elementos de la realidad que tienen alguna cualidad interesante (v.gr., valor numérico, ser o no verdaderos, etc.).

1.1.1 Lenguaje formal

Un *lenguaje formal* se puede establecer al definir dos componentes:

- un *alfabeto*, que especifica qué *símbolos* se utilizan en el lenguaje, y
- una *sintaxis*¸ que indica cómo los símbolos pueden juntarse para constituir elementos del lenguaje.

El *alfabeto* A es un conjunto finito de símbolos, no vacío, usualmente entendido con alguna notación intuitiva[1].

---

[1] Se hablará informalmente de conjuntos, como colecciones. Más adelante, los conjuntos en sí se pueden tratar de manera formal. Por otra parte, el alfabeto es un conjunto no vacío, de modo que siempre hay letras para armar palabras con él.

Ejemplo A: Alfabetos

Se pueden considerar los siguientes alfabetos, apropiados para los lenguajes que se mencionan:

- para simbolizar números reales no negativos, v.gr., `3.14159`, o para denotar secciones de un libro, v.gr., `1.2.2`, considérese el alfabeto que incluye los diez dígitos decimales más el símbolo `'.'`:
  `{0,1,2,3,4,5,6,7,8,9,.}`
- para expresar notas musicales, v.gr., `Do#` :
  `{Do,Re,Mi,Fa,Sol,La,Si,#,b}`
- para representar cartas de una baraja inglesa, v.gr., A ♥:
  `{♣,♠,♥,♦,A,2,3,4,5,6,7,8,9,10,J,Q,K}`

etc.

§

Una *fórmula* o *palabra* es una secuencia de 0 o más símbolos del alfabeto. El conjunto de todas las palabras posibles del alfabeto se denomina `A*`. La cadena vacía de símbolos se representa con el símbolo ε.

Algunas fórmulas o palabras se consideran *bien formadas*. Éstas son las que satisfacen las *reglas* establecidas por la *sintaxis*. Dicho de otro modo, la sintaxis establece qué palabras están bien formadas y, de paso, se entiende que el lenguaje está constituido, precisamente, por las fórmulas. En adelante se hablará de *fórmulas* para referirse a fórmulas bien formadas y solo se añadirá un calificativo de bien o mal formada si la situación lo requiere.

Por ejemplo, en el lenguaje de los números reales no negativos arriba mencionado, las siguientes palabras son bien formadas:

`0, 1.2, 1234.94, 6.28`

También, en el lenguaje de las secciones de libros, son bien formadas:

`0, 1.2, 12.3.4, 6.28`

Nótese que la palabra `12.3.4` puede denotar una sección de un libro, pero no un número real no negativo. En este caso, las reglas sintácticas que definen el lenguaje de las secciones deben dar como bien formada una palabra que no es considerada así en el lenguaje de los números reales no negativos.

El lenguaje es, entonces, un subconjunto de `A*`, constituido por las fórmulas que la sintaxis establezca. Para expresar las reglas sintácticas se debe usar un *metalenguaje*[2].

Ejemplo B: Un lenguaje sencillo

Considérese el lenguaje `L`, que usa el alfabeto `A = {x,y}`, cuya sintaxis expresada en lenguaje natural está dada por la siguiente regla:
"*una fórmula es cualquier cadena finita de `0` o más símbolos `x`, seguidos por uno a tres símbolos `y`, o una cadena de uno o más símbolos `x`*".

Las siguientes palabras están en el lenguaje `L`:

```
y
xy
xyyy
xxxxxxxxxxxxx
```

Las siguientes palabras no están en `L`

[2] Un *metalenguaje* es un lenguaje entendido, con el que se puede hablar del otro lenguaje. Podría ser el español, simbología matemática diseñada para ese fin, etc.

xyyyyy
xyxy
xxxxy

§

En informática es usual establecer las reglas sintácticas de un lenguaje mediante una *gramática*, que no es otra cosa que un formalismo para, dado un alfabeto y unas reglas de construcción, precisar qué palabras que se construyan con las letras del alfabeto se consideran parte del lenguaje. Una posibilidad de hacer esto es mediante las llamadas gramáticas *BNF* (*Backus-Naur Form)*.

Una gramática BNF para L es una cuádrupla $G=(A,N,\Sigma,P)$, donde

- A es el alfabeto del lenguaje L. Se dice que éstos son los símbolos *terminales*.
- N es un alfabeto de símbolos auxiliares, diferentes de los de A. Se dice que éstos son los símbolos *no terminales*.
- Σ es un elemento distinguido dentro de N. Se llama el símbolo *inicial*.
- P es un conjunto de *producciones*.

$V = A \cup N$ es el alfabeto que reúne los símbolos terminales con los no terminales.

Una *producción* es una regla sintáctica de la forma

$$\alpha \rightarrow \beta$$

donde α, β son palabras en V^*, pero α no es una palabra en A^*. Lo que se quiere entender es que siempre que se encuentre el patrón α dentro de una cadena de símbolos, éste se puede cambiar por la cadena β.

Una *fórmula (fórmula bien formada, fbf)*, para una gramática BNF, es una palabra constituida por solo símbolos terminales y derivable del símbolo inicial usando un número finito de producciones[3].

Para denotar las producciones de una manera estructurada se usan notaciones abreviadas que usan al lado derecho expresiones formadas por palabras separadas por el carácter |:

$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_m$

Esta es una manera corta de representar el conjunto de m reglas

$\alpha \rightarrow \beta_1$
$\alpha \rightarrow \beta_2$
…
$\alpha \rightarrow \beta_m$

Por otro lado, se permiten extensiones de la notación BNF para denotar de una manera simple y compacta algunos tipos de reglas, v.gr.

- ε es la palabra vacía.
- Se usan paréntesis simples para agrupar expresiones.
- Elementos opcionales son denotados entre paréntesis cuadrados. Por ejemplo:

  $$\alpha \rightarrow \beta[\gamma]$$

  es una representación abreviada de las reglas
  $\alpha \rightarrow \beta$
  $\alpha \rightarrow \beta\gamma$

[3] Es importante observar que los elementos de un lenguaje –sus palabras, sus fórmulas- se definen en muchos casos como resultado de un proceso recurrente. Por la misma razón, cuando se trate de hablar de propiedades de los elementos de un lenguaje, es normal que estas se establezcan también de manera recurrente.

- Elementos repetidos n veces son denotados con un exponente que indica el número de repeticiones.

$$\alpha \to \gamma^n$$

Por ejemplo,

$$\alpha \to (\beta | \delta)^2$$

es una representación abreviada de la regla

$$\alpha \to (\beta | \delta)(\beta | \delta)$$

- Elementos repetidos `0` o más veces son denotados con un `*`. Así,

$$\alpha \to \gamma^*$$

es una representación abreviada equivalente a la de las reglas

$\alpha \to \varepsilon$

$\alpha \to \gamma\alpha$

- Elementos repetidos `1` o más veces son denotados con un $^+$. Así,

$$\alpha \to \gamma^+$$

es una representación abreviada equivalente a la de las reglas

$\alpha \to \gamma$

$\alpha \to \gamma\alpha$

Usualmente, los símbolos no terminales diferentes de Σ se distinguen porque se enmarcan en paréntesis angulares $\langle\rangle$. Esto permite manejar como símbolos cadenas de caracteres complejas que, en la teoría de lenguajes formales, representan sublenguajes conocidos como *categorías sintácticas.*

La distinción sintáctica entre símbolos no terminales y terminales hace que la gramática se entienda correctamente con la sola mención de las reglas de producción correspondientes. Así mismo, las categorías sintácticas se denotan con nombres suficientemente descriptivos para entender la intención de quien diseña el lenguaje.

Ejemplo C: Gramáticas

El lenguaje `L` del Ejemplo B en 1.1.1 se podría expresar con la siguiente gramática

$L \to x^*y \mid x^*yy \mid x^*yyy \mid x^+$

El lenguaje ⟨db⟩ denota los dígitos binarios:

⟨db⟩ → 0|1

El lenguaje ⟨digito⟩ denota los dígitos decimales:

⟨digito⟩ → 0|1|2|3|4|5|6|7|8|9

Nótese que también pudo haberse definido:

⟨digito⟩ → ⟨db⟩|2|3|4|5|6|7|8|9

El siguiente lenguaje define cadenas de 4 dígitos decimales:

⟨4dig⟩ → ⟨digito⟩⟨digito⟩⟨digito⟩⟨digito⟩

Y se habría podido definir, de manera equivalente:

$\langle 4dig\rangle \to \langle digito\rangle^4$

El siguiente lenguaje denota números reales no negativos, expresados en notación decimal[4]:

⟨real no_neg⟩ → ⟨entero sin signo⟩ | ⟨fraccion decimal⟩ |

---

[4] En rigor, solo sirve para denotar números reales racionales.

$\langle$entero sin signo$\rangle\langle$fraccion decimal$\rangle$
$\langle$fraccion decimal$\rangle \rightarrow$.$\langle$entero sin signo$\rangle$
$\langle$entero sin signo$\rangle \rightarrow$ $\langle$digito$\rangle^+$

§

Ejemplo D: `Nat` ***y*** `SNat`

Con el alfabeto `A1 = {S,0}`, se puede definir un lenguaje que represente los números naturales (enteros no negativos). Obsérvese que la notación que se busca no es la usual (`0,1,2,3,…`), ya que el alfabeto ni siquiera tiene símbolos para los dígitos que no son `0`.

El lenguaje `Nat` se puede definir con la gramática BNF
$\langle$Nat$\rangle \rightarrow$ 0 | S$\langle$Nat$\rangle$

Los elementos del lenguaje son `{0,S0,SS0,SSS0,…}`. La idea, que se desarrollará mejor más adelante, es entender que el número `k` se representa con un `0` antecedido por `k S`'s.

También se puede definir un lenguaje `SNat` que contenga todas las palabras de `Nat` y, además, denote sumas de números naturales (en la notación de `Nat`). El alfabeto será ahora ahora `A2 = {S,0,+}`. y una gramática para `SNat` puede ser:
$\langle$SNat$\rangle \rightarrow \langle$Nat$\rangle$ | $\langle$SNat$\rangle$ + $\langle$Nat$\rangle$

Así, las palabras de `Nat` también están en `SNat`, pero también hay palabras como
`0+SSS0, SSS0+0, SS0+SSS0, SSS0+SS0, SSS0+SSS0+0+SS0, …`

que se espera que denoten sumas.

§

Ejemplo E: Números binarios

Un ejemplo más de lo que puede ser una semántica lo constituyen las notaciones para números y sus significados correspondientes. Por ejemplo, los números binarios se pueden representar con cadenas de símbolos en el alfabeto `B = {0,1}`. Se pueden definir con la gramática siguiente:
$\langle$db$\rangle \rightarrow$ 0 | 1
$\langle$num_bin$\rangle \rightarrow \langle$db$\rangle$ | $\langle$num_bin$\rangle\langle$db$\rangle$.

§

Ejercicios 1.1.1

1 Dado un alfabeto `A`, ¿es el conjunto `A*` necesariamente infinito? Explique su respuesta.

2 Dado el alfabeto `A={a,b,c}`, exprese en lenguaje natural una descripción sintáctica posible, de tal forma que `abc`, `abcab`, `bcac`, `bbabbc` sean formulas bien formadas, pero `ab`, `abba`, `bcb`, `c`, `aaaaacca` no lo sean.

3 Sea `L` el lenguaje de las palabras que comienzan con el símbolo `a` y terminan con el símbolo `b`, en el alfabeto `A = {a,b,c}`. Diseñe una gramática BNF que describa a `L`.

4 Sea `num_dec` el lenguaje de los números enteros no negativos, expresados en base `10`. Defina un alfabeto y una gramática que generen esta notación. Explique cómo se puede generalizar esto para notaciones de los mismos números en una base entera positiva mayor que `1`.

1.1.2 Semántica

La semántica permite darle significado a las palabras de un lenguaje. En los ejemplos de la sección anterior, es claro que los lenguajes que se construyen para representar números pueden significar, de manera natural, los números que culturalmente se leen con ellos. Así, `1` debería significar el número "uno", `2.3` el decimal "dos coma tres", etc. Sin embargo, puede no ser claro lo que significan las representaciones bien formadas de notas musicales o de cartas de la baraja.

Una *semántica* para un lenguaje es una asignación que hace corresponder objetos de la realidad modelada con palabras del lenguaje. Todo objeto de la realidad, que fuera motivo de interés por parte de quien la observa, debería tener una palabra que lo representara; de lo contrario, el lenguaje no sería *suficientemente expresivo*. Es normal que la semántica se explique recursivamente, porque las palabras de los lenguajes se definen por este método. De hecho, la recursión que define la semántica se apoya, usualmente, en los casos recursivos que definen la gramática.

Ejemplo A: Una semántica simple

El lenguaje `L` del Ejemplo B en 1.1.2 se podría expresar con la siguiente gramática
$\langle L \rangle \rightarrow$ `x*y | x*yy | x*yyy |` x^+

Ahora, el lenguaje `L` podría interpretarse con `I`, una asignación de valores numéricos, de la siguiente forma:

- `x` denota el número 3
- `y` denota el número 2
- la yuxtaposición de un símbolo al lado de otro denota la suma de su interpretación.

Así, las fórmulas del ejemplo se interpretan como

```
I(y) = 2
I(xy) = 5
I(xyyy) = 9
I(xxxxxxxxxxxxx) = 39.
```

§

Ejemplo B: Semántica para `Nat` ***y*** `SNat`

Considérese el lenguaje `Nat`, del Ejemplo D en 1.1.1. Las fórmulas de `Nat` son de la forma S^k0, donde `k` es un entero no negativo. La semántica que se quiere usar es sencilla: la fórmula S^k0 se interpreta como el entero `k`, i.e., los llamados números naturales. Nótese cómo todo número natural `k` tiene una fórmula en el lenguaje formal que se interpreta como `k`. Más formalmente, se puede pensar en una semántica `J` para `Nat`, tal que:

$J(0) = 0$
$J(S\alpha) = 1 + J(\alpha)$

Así, `SSSSS0`, `SS0` son fórmulas de `Nat`, con las que se quieren denotar los números `5` y `2`. Por ejemplo:

```
  J(SS0)
= 1 + J(S0)
= 1 + 1 + J(0)
= 1 + 1 + 0
= 2
```

Obsérvese cómo la semántica se define para toda posible fórmula bien formada del lenguaje. De esta manera, toda fórmula bien formada tiene una interpretación. Si esto no es posible, la interpretación no está bien definida.

En el mismo Ejemplo D de 1.1.1 se define el lenguaje `SNat` para denotar sumas de dos números naturales. La idea del párrafo anterior se puede extender para definir una gramática para `SNat`. Además de cadenas de la forma S^k0, en `SNat` hay cadenas de la forma S^k0+S^j0. La semántica que se quiere para esta cadena es que una tal cadena denote la suma `k+j`.

Por ejemplo, las cadenas `S0+SS0`, `S0+SS0+SSSS0` se pueden entender como notaciones para las sumas `1+2` y `1+2+4`.

Como en `Nat`, `SSSSS0`, `SS0` son fórmulas de `SNat` con las que se quieren denotar los números `5` y `2`. Pero también lo son `S0+SS0`, `S0+SS0+SSSS0`. Éstas deberían denotar las sumas `1+2` y `1+2+4`. Esta idea de "querer denotar" es lo que se realiza a través de una semántica.

La siguiente semántica sirve para llevar a cabo la idea pretendida[5]:

```
H(0) = 0
H(Sα) = 1 + H(α)
H(0+β) = H(β)
H(Sα+β) = H(Sα) + H(β)
```

Por ejemplo:

```
H(SS0+0+SSS0)     = H(SS0+0) + H(SSS0)
                  = H(SS0) + H(0) + H(SSS0)
                  = …
                  = H(0) + 1 + 1 + H(0) + H(0) + 1 + 1 + 1
                  = 0 + 1 + 1 + 0 + 0 + 1 + 1 + 1
                  = 5
```

§

Ejemplo C: Semántica para números binarios

Se quiere definir una semántica para `num_bin` que corresponda a la forma usual en que se interpretan los números binarios, con el lenguaje definido en el Ejemplo E de 1.1.1.

Si la semántica se define como[6]:

```
B(0) = 0
B(1) = 1
B(αd) = 2*B(α) + B(d)
```

un número binario tiene como significado el valor entero que se le asigna usualmente. Por ejemplo:

```
   B(1001)  = 2*B(100) + B(1)
            = 2*(2*B(10) + B (0)) + B(1)
            = 2*(2*(2*B(1) + B(0)) + B (0)) + B(1)
            = 2*(2*(2*1 + 0) + 0) + 1
            = 2*(2*(2*1 + 0) + 0) + 1
            = 9.
```

§

Ejercicios 1.1.2

1 Considere el alfabeto `A = {a,b,c,d,e}`, y el lenguaje `L` definido por:

`L` → a⟨T⟩`*c`

---

[5] Obsérvese que hay una "sobrecarga" de significado para el símbolo +. Dependiendo del contexto, se ve como un símbolo del lenguaje o como el operador de suma usual. Leyendo y escribiendo con cuidado no hay lugar a confusión.

[6] Nótese cómo se puede aceptar y entender una definición de semántica recursiva, en donde el significado de un objeto complejo se apoya en los significados de partes que lo componen.

$\langle T \rangle \rightarrow$ b|c|d|e

Si L se interpreta de manera que

- los símbolos a,b,c,d y e denotan los números 1,2,3,4 y 5 respectivamente
- la yuxtaposición de un símbolo al lado de otro denota el producto de sus interpretaciones

Describa el conjunto de valores que el lenguaje representa.

2 Defina un lenguaje para denotar cadenas de números con dígitos en el alfabeto T = {0,1,2}. ¿Cómo puede definir una semántica que interprete estas cadenas de dígitos como números en base 3?

3 Sea MSNat un lenguaje que contenga expresiones de números naturales, sumas y multiplicaciones. Extienda la gramática indicada para SNat en el Ejemplo B.

1.1.3 Aparato deductivo

Un sistema formal puede modelar una realidad, en cierto sentido, estática, como en los ejemplos de sistemas numéricos de las secciones anteriores. De esta manera se puede pensar en que el sistema tiene un lenguaje para denotar objetos de la realidad.

Los objetos denotados por un lenguaje formal pueden, a su vez, resultar interesantes por tener alguna cualidad que dependa de la realidad. El interés por ciertas fórmulas se puede establecer con mecanismos de transformación similares a los usados para definir los lenguajes, aunque algo más elaborados que una gramática.

Más precisamente, se define un conjunto básico de fórmulas interesantes, que se llaman *axiomas*, y unas *reglas de inferencia*, que permiten definir como interesantes otras fórmulas a partir de fórmulas interesantes que satisfagan ciertas condiciones. Los axiomas y las reglas constituyen un *aparato* o *cálculo deductivo*, que se usa para inferir, derivar o deducir (estos verbos se utilizan indistintamente) fórmulas 'interesantes' a partir de unas que ya lo sean. La noción de interesante, no sobra repetirlo, es una cuestión de modelaje y depende de la intención de un observador, si bien hay sistemas formales comúnmente aceptados y utilizados.

La *deducción*, entendida como el uso de un aparato deductivo, es un proceso recursivo que sirve para develar fórmulas interesantes. El nombre técnico para algo interesante es *teorema.* Un teorema es un axioma o se deriva de otros teoremas mediante reglas de inferencia.

Antes de dar una definición más rigurosa de lo que significan los conceptos anteriores, se tratará de explicarlos intuitivamente con un ejemplo.

***Ejemplo A: Inferencias sobre* Nat *y* SNat**

Considérese el lenguaje SNat del Ejemplo D de 1.1.1 y la semántica establecida en el Ejemplo B de 1.1.2. Ahora se puede dar una regla de inferencia que sirva para efectuar, sobre la notación, cálculos de sumas.

Cada número natural es representable en SNat. Y cada suma de números naturales también lo es. Una forma de inferencia que parece interesante es la que calcula las sumas de números naturales dando como resultado números naturales. Las reglas siguientes lo harían:

$$[+1] \qquad \frac{0+x}{x}$$

$$[+2] \qquad \frac{Sx+y}{x+Sy}$$

[+1] y [+2] son ejemplos de reglas de inferencia. Las variables x, y representan elementos del lenguaje SNat.

En [+1] se quiere entender que, a partir de una fórmula de la forma 0+x (i.e., 0+0, 0+S0, 0+SS0, …) se puede derivar la correspondiente fórmula x (i.e., 0,S0,SS0,…). Es como si la regla explicara lo que significa sumar 0 a x: debe dar x como resultado.

En [+2] se quiere entender que, a partir de una fórmula de la forma Sx+y se puede derivar la correspondiente fórmula x+Sy. Aquí es más difícil imaginar todos los ejemplos, pero la idea es que de una suma como SS0+SSS0 se pueda derivar la suma S0+SSSS0.

Si se usa una flecha '→' para denotar un paso de derivación, entonces se pueden encadenar pasos para conformar derivaciones. Por ejemplo:

```
   SS0+SSS0
→      ⟨[+2]⟩
   S0+SSSS0
→      ⟨[+2]⟩
   0+SSSSS0
→      ⟨[+1]⟩
   SSSSS0
```

Lo 'interesante', en este ejemplo, es el valor de las expresiones. Las reglas están diseñadas para que una expresión derive en otra con el mismo valor que, al final, será un número natural. Es decir, las reglas sirven para *calcular* valores de números expresiones que representan sumas de números naturales.

Curiosamente, en este ejemplo no se ha hablado de axiomas. La idea es que cualquier suma de la forma S^k0+S^j0 se puede pensar como 'interesante', i.e., se podrían definir como axiomas esta clase de fórmulas. Las fórmulas que se derivan son elementos de Nat u otras expresiones de sumas que parecen 'más simples'. Como los valores se conservan, todas las sumas derivadas tienen un mismo valor que, como lo muestra el cálculo, equivale a un único valor de Nat.

§

En el caso más general, una regla de inferencia tiene la forma

$$\langle\text{Identificador}\rangle \quad \frac{\text{Hip}_1, \ldots, \text{Hip}_k}{\text{C}} \Bigg| \quad \langle\text{Condiciones}\rangle$$

donde ⟨Identificador⟩ es una etiqueta para denotar la regla, y ⟨Condiciones⟩ es un conjunto de restricciones para poder aplicar la regla (cuando no hay restricciones, no se escribe nada). Las fórmulas $\text{Hip}_1,\ldots, \text{Hip}_k$ se conocen como *hipótesis* y la fórmula C como *conclusión* de la regla.

Hay una notación horizontal para reglas de inferencia, cuando no hay condiciones de aplicación:

$$\langle\text{Identificador}\rangle\text{:}\quad \text{Hip}_1, \ldots, \text{Hip}_k \vdash \text{C}$$

Cuando hay un aparato deductivo asociado a un sistema formal, un *teorema* es un axioma o la conclusión de una regla aplicada a un conjunto de hipótesis que ya sean teoremas. Una *demostración* de un teorema C es una secuencia de *deducciones* (también llamadas *inferencias*, aplicaciones de reglas de inferencia a hipótesis) que termina con C como última conclusión.

Si las reglas de inferencia reflejan la forma en que la realidad se comporta (en cuanto a lo que se quiere ver como interesante), se dice que son *correctas* o *adecuadas* (inglés: *sound*). Así, una regla correcta permite deducir fórmulas interesantes de fórmulas interesantes. El cálculo deductivo como tal se dice *correcto* o *adecuado* si todas sus reglas lo son.

Para un aparato deductivo, la propiedad de corrección es necesaria para que pueda considerarse útil. De lo contrario, podrá deducir –y considerar como interesantes- fórmulas que, en la realidad, no lo sean. Es decir, estará modelando mal la realidad, en cuanto a lo que interesa.

Además de correcto, un aparato deductivo podría deducir solo algunas de las fórmulas interesantes que describen la realidad, de modo que podría haber fórmulas interesantes que podrían no ser demostrables. En este caso, el aparato deductivo sería *incompleto*. La propiedad de *completitud*, muy deseable en general, no es siempre alcanzable de manera razonable o práctica.

Ejemplo B: Lenguajes formales como resultado de deducciones

La manera en que se ha definido un lenguaje formal, como resultado de la aplicación de unas reglas gramaticales, se puede explicar en términos de un cálculo deductivo.

De hecho, dado un alfabeto `A`, sea `G =(V,A,N,`Σ`,P)` una gramática según la definición de 1.1. Considérese un cálculo deductivo definido así:

- Axioma 1: Σ
- Para cada producción $\alpha \to \beta$, se define una regla de inferencia:

$$\langle\alpha\to\beta\rangle \qquad \frac{\alpha}{\beta}$$

Entonces, el lenguaje es el conjunto de palabras en `A*`, derivables desde Σ mediante la aplicación de las reglas de inferencia (las producciones!).

§

Ejemplo C: El sistema MIU

Considérese el sistema `MIU` [Hof1979], definido con las reglas:

`[r0]` `MI.`
`[r1]` α`I.` $\to$ α`IU.`
`[r2]` `M`α`.` $\to$ `M`$\alpha\alpha$`.`
`[r3]` α`III`β $\to$ α`U`β
`[r4]` α`UU`β $\to$ $\alpha\beta$

Esta es una aplicación del Ejemplo B para un lenguaje específico. La regla `[r0]` es, en realidad, un axioma. Las reglas `[r1],…,[r4]` dan lugar a reglas de inferencia.

El alfabeto es `{M,I,U,.}`. En [Hof1979] se plantea la pregunta de descubrir si `MU.` es derivable (de `MI.`) mediante estas reglas. Esto puede refrasearse como preguntar si `MU.` está en el lenguaje que las producciones definen o, en términos de un cálculo deductivo, decidir si `MU.` es un teorema.

El sistema `MIU` es un sistema formal artificial. No está modelando verdades, como en las discusiones que motivan las definiciones, pero sí habla de cierta clase de fórmulas que son interesantes: las derivables desde `MI.`, con las reglas dadas. La corrección de las reglas no es cuestionable, debido a lo artificial del sistema ("se define así"). La completitud del sistema corresponde a la construcción de una demostración de `MU.`, en caso de que ella exista.

§

Ejercicios 1.1.3

1 Extienda las ideas del Ejemplo A para definir reglas de inferencia adecuadas para el lenguaje `MSNat` del Ejercicio 1.1.2.3.

2 Para resolver el acertijo `MU` (descubrir si `MU.` es un teorema del sistema `MIU` del Ejemplo C) se puede notar que el número de `Is` en las palabras del lenguaje nunca es divisible por `3`. Explique por qué esto es así y cómo ayuda esto a resolver el acertijo.

1.2 SISTEMAS LÓGICOS

Una clase importante de sistemas formales son los *sistemas lógicos*, los cuales modelan realidades en las que hay afirmaciones, y éstas pueden ser ciertas o falsas. En un sistema lógico las afirmaciones ciertas son interesantes, i.e., los teoremas son afirmaciones que deben ser ciertas. Los axiomas de un sistema lógico son fórmulas que representan afirmaciones que se consideran verdaderas. Las reglas de inferencia constituyen un *aparato deductivo* que permite explicar que ciertas fórmulas son ciertas cuando algunas otras lo son.

Un sistema lógico clásico maneja dos valores de verdad posibles: verdadero y falso. Es usual incluir, como parte del lenguaje, dos símbolos que actúan como constantes, `true` y `false` y que, al momento de definir su semántica se entienden como fórmulas que son verdadera y falsa, respectivamente. También se incluyen símbolos como =, ∧, ∨, ⇒, ⇐, ≡, ¬, …, para construir fórmulas que representan afirmaciones de diferentes formas (igualdad, conjunción, disyunción, implicación, consecuencia, equivalencia, negación, ...).

Para fijar ideas, supóngase un sistema lógico cuyas fórmulas sean afirmaciones sobre la igualdad entre objetos. En él puede haber axiomas que modelen las llamadas *reflexividad* y *simetría* de la igualdad:

```
Ax 1:       x=x                 // reflexividad
Ax 2:       x=y  ⇒  y=x         // simetría
```

En este sistema, además de símbolos como = y ⇒, los cuales son parte del alfabeto, las letras `x,y,z` representan fórmulas (lo que se escribe al final de los renglones como `//…` son comentarios explicativos, al estilo de lo usado en los lenguajes de programación). Los significados de los símbolos son informalmente entendidos (¡más tarde serán formalmente definidos, en todo caso!), así:

`Ax 1`: "Cualquier cosa es igual a sí misma"

`Ax 2`: "Si una cosa es igual a otra, entonces la otra es igual a la primera"

Como ejemplo de regla de inferencia para esta clase de sistemas formales se puede citar una muy conocida (desde Aristóteles), llamada *Modus Ponens*, que se puede expresar de la siguiente manera:

$$\textit{Modus Ponens:}\quad \frac{p \Rightarrow q,\ p}{q}$$

Para usar una regla como la anterior, se entiende que p y q son fórmulas. Claramente, p ⇒ q también debería ser una fórmula. Y la regla permite afirmar que, si p ⇒ q y p son teoremas, también q debe serlo. Por ejemplo, cuando p se interpreta como "llueve" y q se interpreta como "el campo se moja", la regla dice: "si es cierto que cuando llueve, el campo se moja y es cierto que llueve, entonces es cierto que el campo se moja".

Otro ejemplo importante de regla de inferencia es la llamada *Regla de Leibniz*. Esta regla deduce conclusiones mediante el 'remplazo de iguales por iguales'. Es decir, si se tiene un concepto de igualdad (o de identidad), entre objetos, es factible remplazar un objeto por otro igual, en una afirmación cierta, para obtener otra afirmación cierta. La regla tien la siguiente forma:

$$\textit{Leibniz:}\quad \frac{x=y}{E[x]=E[y]}$$

La notación `E[z]` se refiere a una expresión `E` que tiene como parámetro una subexpresión `z`. De nuevo, la Regla de Leibniz afirma que si dos expresiones `x`, `y` son iguales, cualquier expresión `E[x]` tiene el mismo valor que la expresión `E[y]`.

La siguiente terminología se usa para referirse a ciertas fórmulas en sistemas lógicos (aunque puede extenderse a las derivaciones en general):

Axioma : verdad asumida.
Teorema : verdad demostrada.
Lema : teorema de importancia menor.
Corolario : aserción que se deduce fácilmente de un teorema.
Proposición : teorema, lema.

Conjetura : afirmación que se adivina cierta. Si se prueba, es un teorema.

1.2.1 Verdad

Ya se indicó que los sistemas lógicos tratan con la verdad de las afirmaciones. Pero, ¿qué es la verdad?

Según [Mey2010], la verdad puede ser legal (decidida por un jurado, un juez, etc.), autoritaria (declarada por una autoridad), científica (confirmada por un experimento), probable (establecida por observaciones estadísticas), filosófica (establecida mediante argumentación), matemática (demostrable mediante una prueba formal). Estas posibilidades muestran que la comprensión de la verdad es una cuestión cultural y que depende, incluso, del uso que se le da.

En términos de informática, la verdad tiene que ver más con un enfoque matemático o, en términos de 1.1, de sistemas formales que manejan el concepto de verdad o falsedad[7]. El lenguaje formal que el sistema lógico maneje puede tener más o menos expresividad (y entonces, semántica más o menos complicada) y la determinación de la verdad de una fórmula específica puede ser también más o menos complicada por esta razón.

En los capítulos posteriores se estudiarán lógicas que van subiendo en complejidad de sus fórmulas, así como en sus posibilidades semánticas. Una fórmula puede ser verdadera en toda realidad (como lo son esos axiomas para las lógicas más sencillas) o verdadera en unos casos pero no en otros. La verdad debe establecerse siempre con respecto a la realidad que se pretende modelar, pero una de las cosas deseables con los sistemas lógicos es que éstos sean herramientas operacionales para el cálculo de la verdad de las fórmulas. Es decir, el sistema lógico se constituye en una plataforma de manipulación simbólica que permite dilucidar si algo es cierto o no en la realidad, de acuerdo con que su representación en un sistema lógico se pueda demostrar que es verdadera con el aparato deductivo correspondiente.

1.2.2 Pruebas formales

Las pruebas formales son derivaciones en un sistema lógico. En su forma básica, una prueba parte de teoremas (verdades conocidas) y, mediante el uso de reglas de inferencia, se llega a lo que se ha querido probar.

En el Cap. 2 se explicará con más detalle una notación para pruebas formales. Sin embargo, de manera general, una prueba tiene pasos de deducción en los que reglas como

$$\mathtt{r:}\ \ \mathtt{Hip}_1, \ldots, \mathtt{Hip}_k \vdash \mathtt{C}$$

se pueden aplicar. La aplicación es posible cuando se dispone de teoremas $\mathtt{Teo}_1, \ldots, \mathtt{Teo}_k$ que son instancias correspondientes de las hipótesis $\mathtt{Hip}_1, \ldots, \mathtt{Hip}_k$, i.e., hay una manera de remplazar las variables que aparecen en los $\mathtt{Hip}_j$ de manera que el remplazo resulte en los $\mathtt{Teo}_j$ correspondientes. El resultado de la aplicación de la regla es una `fórmula C'`, que es una versión instanciada de `C` con los mismos remplazos sobre `C` que sirvieron para hacer corresponder los $\mathtt{Teo}_j$ con los $\mathtt{Hip}_j$.

En los sistemas lógicos, siendo la verdad la cualidad interesante para calcular, una regla de inferencia como la de arriba tiene una interpretación pretendida; de hecho, se quiere entender que debe ser cierto que la fórmula

$$\mathtt{Hip}_1 \land \ldots \land \mathtt{Hip}_k \Rightarrow \mathtt{C}$$

---

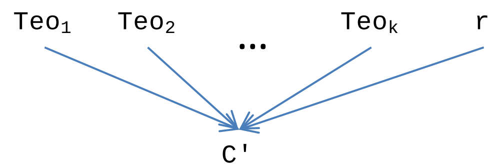
[7] Hay sistemas lógicos en los que se maneja más de un valor de verdad posible e, incluso, puede concebirse una verdad con infinitos valores posibles. Aquí se seguirá trabajando con lógica de dos valores, lo que es interesante y suficiente para muchísimas aplicaciones.

es verdadera[8], para cualquier forma en que se remplacen las variables que aparezcan en ella. Por otro lado, si se contara con un teorema de la forma

$$\mathtt{Hip}_1 \wedge \ldots \wedge \mathtt{Hip}_k \Rightarrow \mathtt{C}$$

para efectos de calcular se podría incluir una regla de inferencia como `r`.

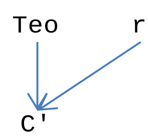
Un paso de deducción como el arriba descrito se puede representar gráficamente así:



Nótese que, teniendo en cuenta el comentario anterior, `r` puede ser una regla de inferencia o un teorema ya probado.

Por otro lado, es importante señalar que los arcos en estos grafos tienen orientación: la verdad de una fórmula en un nodo depende, para su prueba, de que todas las fórmulas en los nodos de los que salen arcos hacia ella sean verdaderas.

Otra forma de deducción corresponde al hecho de que una fórmula `C'` sea una *instanciación*, es decir, el resultado de remplazar unas variables de un teorema por una fórmula cualquiera. Como se supone que los teoremas son verdaderos, también lo son sus instanciaciones. Los remplazos se denotan con una notación similar a la usada en asignaciones de un lenguaje de programación (aunque se entiende que los remplazos se hacen simultáneamente). Por ejemplo, un remplazo simultáneo de las variables `x`, `y` por las fórmulas `x+1` y `a+z` se denota como `x,y:= x+1,a+z`. Gráficamente, una instanciación se representa así (`Teo` es el teorema que se instancia, `r` el remplazo, `C'` el resultado de la instanciación):



En general, en un sistema lógico se prueban teoremas de la forma

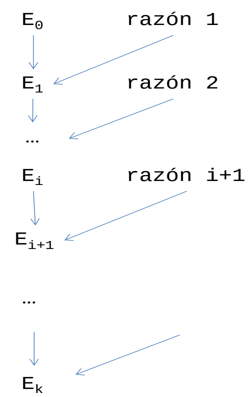
$$\alpha_1 \wedge \ldots \wedge \alpha_s \Rightarrow \beta$$

donde $\alpha_1,\ldots,\alpha_s$, β son fórmulas. Una prueba completa de un teorema así puede representarse gráficamente con un grafo dirigido acíclico en el que en los nodos se incluyen fórmulas y los arcos que entran a un nodo siguen un patrón como los arriba descritos para `C'`. La excepción son esos nodos que no requieren demostración para considerarse verdaderos: a ellos no llega ningún arco. Estos nodos contienen fórmulas que son axiomas o alguna α_j. Para que la demostración sea completa, el grafo debe contener un nodo con la fórmula β.

En 2.6 se verá que las pruebas que allí se estudian tiene una representación gráfica sencilla, siguiendo un esquema de la forma

---

[8] El significado de los símbolos $\wedge$ y $\Rightarrow$ se entiende -por ahora- intuitivamente: $\wedge$ es la conjunción "y", $\Rightarrow$ se lee "implica". En 1.3.1 se discute qué puede querer decir esto.



donde $\texttt{razon}_j$ es una regla de inferencia o un teorema ya demostrado. Si la demostración de este teorema auxiliar (también llamado *lema*) se requiere, la demostración correspondiente hace parte integral de la prueba principal.

1.2.3 El método axiomático

La forma en que los sistemas lógicos modelan la verdad y las deducciones se conoce como el *método axiomático* de hacer lógica. Se entiende que la verdad de las aserciones es deducida a partir de verdades asumidas como tales y reglas de inferencia.

1.2.3.1 Qué debería cumplir un axioma

Un axioma no se demuestra, pero sí se justifica, i.e., se argumenta para afirmar que es una verdad universal en la realidad que interesa. Si se logra evidenciar que una aserción no es verdad en cierto caso, ésta no puede ser un axioma.

Al definir axiomas hay dos problemas posibles:

- Axiomas falsos. Contra esto, tratar de asegurarse -intuitivamente- de que lo que se pretende que sea un axioma se satisface en varios casos. Esto no basta muchas veces, pero es mejor que nada[9].
- Axiomas que son teoremas demostrables. En realidad, esto no causa problemas en cuanto a hacer las cosas correctamente ("modelar bien"). Pero un sistema formal será más fácil de entender si tiene menos axiomas porque, al fin y al cabo, éstos se aceptan con "actos de fe". Mejor tener menos axiomas pero, por otro lado, es bueno tener sentido común para, por razones prácticas, decidir si algo se debe demostrar o no.

1.2.3.2 Qué debería cumplir una regla de inferencia

Las reglas de inferencia abstraen leyes de deducción que son universalmente correctas en la realidad modelada. La corrección de una regla corresponde al hecho de que, siempre que pueda aplicarse sobre hipótesis que corresponden a teoremas, da lugar a un resultado que también debería ser un teorema. En el caso de sistemas lógicos, si una regla de inferencia se aplica a fórmulas verdaderas, la conclusión debe ser verdadera.

Como los axiomas, si una pretendida regla no es correcta en un caso, no se puede asumir como parte del sistema de inferencia, porque esto dará lugar a un modelaje errado de la realidad y, usualmente incoherente o absurdo. Pero, en general, no se puede demostrar la corrección de las reglas de inferencia y, al definirlas, pueden darse problemas equivalentes a los ya mencionados para los axiomas.

---

[9] Por ejemplo, hay dificultades, a veces insalvables, cuando el número de casos es infinito.

Más adelante, se verá que axiomas y reglas de inferencia se pueden intercambiar sin afectar lo que con ellos se puede demostrar.

Ejemplo A: Ecuaciones sobre `SNat`

Sea `SNatE` un lenguaje basado en `SNat` (Ejemplo D de 1.1) que permite modelar ecuaciones de expresiones de tipo `SNat.` El alfabeto tiene un símbolo nuevo '=' y se permiten fórmulas con la producción
⟨SNatE⟩ → ⟨SNat⟩ = ⟨SNat⟩

La semántica pretendida para una expresión de esta clase debería ser "el lado derecho de la ecuación tiene el mismo valor que el lado izquierdo".

Entonces se pueden incluir los axiomas

```
Ax 1:       x=x               // reflexividad
Ax 2:       x=y  ⇒  y=x       // simetría
```

que describen propiedades características de una igualdad. Esto quiere decir que, si las variables que aparecen en un axioma se remplazan por fórmulas de `SNat`, se espera que lo denotado por la fórmula resultante sea verdadero. Por ejemplo, si en `Ax 1` la variable `x` se remplaza por `SS0`, se obtiene la afirmación

`SS0 = SS0`

que debería interpretarse como que `2=2`, lo cual es verdadero (en la realidad de la aritmética, que es donde se usa este lenguaje) .

En cambio, si se incluyera un axioma de la forma

```
Ax 3:       x+x=x
```

habría problemas. Obsérvese que si se remplaza `x` por `0` en `Ax 3` el resultado es

`0+0=0`

y esto es verdadero con la semántica que se ha definido para `SNat`. Sin embargo, cuando `x` se remplaza por `S0` se obtiene

`S0+S0=S0`

lo cual, de acuerdo con la semántica, equivaldría a afirmar que `1+1=1`, o bien, que `2=1`. Este es un caso de una fórmula que no puede ser axioma porque no representa una verdad universal.

Por otro lado, si se incluyera un axioma como

```
Ax 4: Sx+0=0+Sx+0
```

se puede ver que lo que se afirma es redundante, ya que se puede probar con los otros axiomas:

```
      ⟨Ax 1⟩
   x=x
→     ⟨x:= x+0⟩
   x+0=x+0
→     ⟨[+1]⟩
   0+x+0=x+0
→     ⟨x:= Sx⟩
   0+Sx+0=Sx+0
→     ⟨Ax 2⟩
   Sx+0=0+Sx+0.
```

Como un ejemplo de regla de inferencia incorrecta

$$ri: \quad \frac{\texttt{x=x}}{\texttt{0=Sx}}$$

La regla `ri` afirma que, si `x=x` es verdadero, se puede concluir que `0=Sx` también lo es. Lo primero es siempre cierto, ya que es un axioma. De manera que si se considera el caso en que `x` es , `0=S0` (que se interpreta como `0=1`) también debería ser cierto. Como esto no es así, la regla `ri` es incorrecta.

§

1.2.4 Proposiciones y predicados

El lenguaje natural (español, inglés, etc.) se utiliza enunciando frases que pretenden ser ciertas. Es muy usual que quien habla o escribe enuncie frases conformando un discurso que debería 'tener sentido', i.e., debería ser coherente y comunicar algo. Los sistemas lógicos no hacen otra cosa que modelar la forma en que los discursos del lenguaje natural se construyen y se entienden.

Los razonamientos del lenguaje natural pueden ser más o menos complicados de abstraer en términos de lenguajes formales y sistemas lógicos. La *lógica matemática* se ocupa de estos asuntos y su desarrollo y comprensión actuales corresponde a explicar formalmente cómo se razona en sistemas lógicos que van aumentando en complejidad según la de lo que se pretenda modelar en la realidad. Esta sección pretende ser una introducción sencilla a la forma en que se usan sistemas lógicos para capturar la esencia de los razonamientos en el plano del lenguaje natural.

Una *proposición* es una frase declarativa. Como tal, es una afirmación sobre algo, y puede o no ser verdad. En lógica, una proposición se concibe de manera atómica, de manera que no tiene partes que la compongan. Por eso, para representar proposiciones se usan letras o palabras que actúan como identificadores. Una *lógica proposicional* es un sistema lógico que representa frases declarativas con proposiciones (atómicas).

En lenguaje natural, una proposición tiene una estructura gramatical reconocible ("sujeto + verbo + predicado"). Se verá, más adelante, que si bien la lógica proposicional sirve para demostrar la verdad de muchas afirmaciones declarativas, hay otras para las cuales no es suficiente. Entonces, hay sistemas lógicos que se apoyan en la estructura gramatical de lo que se afirma, e incluso -por mencionar un ejemplo- en reconocer que lo que se afirma sea cierto para una colección de individuos ("todos", "algunos", ...). Esto da lugar a lo que se llamará *lógica de predicados.*

Para referirse indistintamente a proposiciones o predicados se puede usar la palabra *aserción*, entendida como una afirmación cualquiera.

Ejemplo A: Notaciones y significado

Los siguientes ejemplos de aserciones se expresan en una notación de la forma
`⟨denotación de aserción⟩ ≡ ⟨significado⟩`

Una denotación para una aserción es una representación simbólica de ella. Debe incluir símbolos que reflejen las partes destacadas de lo que se afirma. El significado se entenderá con una frase o fórmula en un lenguaje conocido, que podría ser una frase en lenguaje natural, un enunciado matemático, etc.:

```
p       ≡ "está lloviendo"
q       ≡ "Adán es el padre de Abel"
m(A,E)  ≡ "la madre de A es E"
s       ≡ (∀n| n≥0 : 1+2+…+n = n(n+1)/2)
r(k)    ≡ (∀n| n≥k : primo(n^2+n+41))
mu      ≡ "MU no es generable por el sistema MIU"
```

En términos más sencillos, la parte izquierda de cada renglón es una notación formal para la parte derecha, que no es otra cosa que lo que significa la notación. De esta manera:

- `p` y `q` representan frases completas, sin especificar sus partes (sujeto, …). Son proposiciones.
- `m(A,E)` es una frase en la que se destacan su sujeto y su complemento, A, E. Son predicados.

- s es una frase que se lee: "para todo n, n mayor o igual a 0, la suma de los números desde 1 hasta n es igual a n(n+1)/2". Es un predicado. Nótese que hay notación especial para expresar una afirmación sobre una colección.
- r(k) es similar en forma, pero usa una notación primo(x) para afirmar que x tiene la propiedad de ser primo (un número entero mayor que 1 que solo es divisible por 1 y por él mismo). Es un predicado.
- mu es una proposición que se refiere a que cierta palabra esté en un lenguaje, el del sistema MIU, explicado en el Ejemplo A de 1.1.3.

¿Cuáles de ellas son verdad?

p	Depende de una realidad que se modela.
q	Como p. Si el modelo es el Génesis, es verdad.
m(A,E)	El mismo comentario si, además, A representa a Abel y E representa a Eva.
s	Parece cierto … (y lo es, pero se muestra después).
r(k)	Parece cierto … (y no lo es: con k=0 y n=41 es falso).
mu	Es cierto. Hay pruebas ingeniosas, pero puede no ser sencillo verlas.

§

Ejemplo B: La verdad no es fácil de decidir

[Mey2010] presenta ejemplos como los siguientes, donde la verdad no es fácil de dilucidar. El significado intuitivo es el que se conoce de las Matemáticas.

B1 Considérese el enunciado

$$(\forall a,b,c,d: \mathbf{int} \mid a>0 \wedge b>0 \wedge c>0 \wedge d>0: a^4 + b^4 + c^4 \neq d^4)$$

La notación matemática puede ser algo novedosa pero no tan difícil de entender. Se está afirmando que, para toda forma de elegir cuatro números enteros a, b, c, d, que sean mayores que 0, la suma de las potencias cuartas de los tres primeros siempre es diferente de la potencia cuarta del cuarto.

Compárese con un enunciado como

$$(\forall a,b,c: \mathbf{int} \mid a>0 \wedge b>0 \wedge c>0: a^2 + b^2 \neq c^2)$$

que se puede leer como que, siempre que se elijan tres números enteros cualesquiera, la suma de los cuadrados de los dos primeros debe ser diferente al cuadrado del tercero. Esto es falso y, de hecho, hay muchos ejemplos de lo contrario (*contraejemplos*), v.gr., a = 3, b = 4, c = 5.

La afirmación con las potencias cuartas es una conjetura enunciada por Euler en 1769. Más de dos siglos después se comprobó que la conjetura es falsa, con el contraejemplo

$$a = 95800,\ b = 217519,\ c = 414560,\ d = 422481.$$

B2 La ecuación

$$313*(x^3 + y^3) = z^3$$

no tiene solución para x,y,z enteros positivos. Los contraejemplos –soluciones de la ecuación- son números con más de 1000 dígitos decimales.

B3 El Teorema de los 4 colores afirma que todo mapa puede ser coloreado con 4 colores, sin usar el mismo color para regiones adyacentes. Este resultado fue conjeturado en 1852, pero solo pudo ser demostrado como verdadero hasta 1976, después de muchos intentos fallidos que conllevaron desarrollos matemáticos importantes. La demostración de 1976 tuvo detractores que no consideraron aceptable una prueba soportada en los resultados de un programa de computador.

B4 En 1742, Goldbach enunció la conjetura de que todo entero par mayor que 2 es suma de dos primos. Todavía no se sabe si es correcta. Según Wikipedia, ha sido comprobada para n menor que un trillón. Claramente, nunca será una prueba definitiva el comprobar que el enunciado vale para números menores que una cota, por grande que esta sea. Algo que puede ser más sorprendente es que también es posible que la afirmación, aunque sea cierta no se pueda demostrar (i.e., no hay argumentos razonablemente aceptables para conseguir afirmar su verdad).

§

1.3 MÁS SOBRE PRUEBAS

Los conceptos de pruebas, inferencias, deducciones, etc., que se han introducido en las secciones anteriores tienen un sentido eminentemente formal. Una prueba, en lo cotidiano, es una argumentación en la que quien demuestra pretende convencer a un interlocutor sobre la verdad de un enunciado. Naturalmente, una tal argumentación no tiene que ser formal: la realidad de lo cotidiano no es un mundo formal (de alfabetos y fórmulas) y, más allá de esto, no hay axiomas ni reglas de inferencia que describan qué debe ser verdad.

Como ya se notó, la lógica matemática usa el método axiomático para hacer sus pruebas. Cuando se basa en axiomas y en reglas de inferencia que reflejan la realidad, una prueba formal es lo mejor que se puede tener. No obstante, el formalismo puede resultar demasiado exigente para quien argumenta o para quien lee la demostración. Una salida formal para esta dificultad consiste en admitir otras formas de demostraciones que, se pueden considerar como variantes lícitas de las demostraciones básicas (v.gr., pruebas por contradicción, por casos, etc.). Se verá en capítulos posteriores cómo esto es posible.

Otra salida al problema al "exceso de rigor" de las pruebas formales es aceptar argumentos informales como parte de las demostraciones. Aquí hay que tener cuidado para aceptar cada argumento, porque en la informalidad se pueden colar conclusiones erradas que se demuestren verdaderas. Un sistema deductivo que no distinga claramente entre lo que es cierto y lo que es falso es completamente inútil.

En este sentido, aquí se pretende tener un nivel de rigor lo más formal posible, teniendo de presente un sentido común que permita "bajar la guardia" de manera más bien controlada. En otras palabras, pretendiendo saber qué se puede perder cuando se permite un argumento informal.

1.3.1 Inferencias lógicas

En 1.1.3 se ha introducido el concepto de 'regla de inferencia' como parte de un 'aparato deductivo'. En un sistema lógico, las deducciones o inferencias que interesan tratan de calcular la verdad de alguna fórmula. Esta verdad se ha referido a lo que un observador decide que es cierto en una realidad, para así mismo definir axiomas y reglas de inferencia que le permitan calcular esa verdad de manera coherente con su realidad.

El uso de reglas de inferencia conlleva, en general, el poder afirmar que cuando las hipótesis de una regla son verdaderas, la conclusión también debe serlo. Y que esto pase siempre. En términos coloquiales, se dice que la conclusión $\mathtt{C}$ es una consecuencia de las hipótesis $\mathtt{Hip}_1$, …, $\mathtt{Hip}_k$, y se escribe, como en 1.2.2,

$$\mathtt{Hip}_1 \wedge \dots \wedge \mathtt{Hip}_k \Rightarrow \mathtt{C}$$

El símbolo $\wedge$ es el de la conjunción "y". Una fórmula como $\mathtt{Hip}_1 \wedge \dots \wedge \mathtt{Hip}_k$ se va a entender como verdadera si todos los $\mathtt{Hip}_j$ son verdaderos.

Una implicación denota la posibilidad de una inferencia. En general, es una fórmula de la forma

$$\mathtt{p} \Rightarrow \mathtt{q}$$

donde p, q son fórmulas. La fórmula a la izquierda del símbolo $\Rightarrow$, p, se denomina el *antecedente*, mientras que q, la del lado derecho, se llama el *consecuente*.

Una de las primeras cuestiones que hay que dejar en claro, para evitar confusiones, es el valor de verdad de una implicación. En primer lugar, se debe entender que el antecedente y/o el consecuente pueden ser verdaderos o no y, en cada caso, hay que saber si la implicación es verdadera o no. De hecho, la verdad de una implicación se define en términos de la verdad de sus partes. La siguiente tabla, que más adelante se llamará una *tabla de verdad*, establece tal definición:

p	q	$p \Rightarrow q$	
`false`	`false`	`true`	`(1)`
`false`	`true`	`true`	`(2)`
`true`	`false`	`false`	`(3)`
`true`	`true`	`true`	`(4)`

En la tabla, se usan las constantes `true` y `false` para denotar los casos en que las partes sean verdaderas o falsas, respectivamente. Solo hay cuatro casos, como puede verse. Enseguida se discutirá por qué esta tabla se define de esta manera.

El caso `(4)` parece fácil de entender y aceptar. De hecho, así se ha explicado la forma en que una prueba formal sirve para afirmar que algo que se prueba es verdadero.

El caso `(3)` se entiende también fácilmente. No puede ser verdadera una implicación cuando de una verdad se concluye una falsedad.

Los otros dos casos, `(1)` y `(2)`, reflejan el hecho de entender que, de una falsedad se puede concluir tanto otra falsedad como una verdad. En términos prácticos, si en una prueba se llegara a demostrar que α es falso, de allí se podría concluir tanto `true` como `false`. Es decir, en un tal sistema lógico sería tan verdadero `true` como `false`. Si uno quiere que esto pase, lo verdadero no se va a distinguir de lo falso, en el plano de lo formal; en otras palabras, el sistema lógico no sirve para reflejar la realidad (que sí distingue dos valores de verdad).

La idea de que p implique q es la de establecer cuándo está bien decir que el valor de verdad de q es coherente con el de p. La coherencia de que $p \Rightarrow q$ quiere decir que, en la realidad, se pueden observar los casos que se han señalado como verdaderos. Para ilustrarlo con un ejemplo, supónganse los significados
$p \equiv$ "Pedro es almirante"
$q \equiv$ "Pedro es un marino"

Entonces, en la realidad puede darse que (casos en que $p \Rightarrow q$ es verdadero):

- Pedro no es almirante y Pedro no es un marino
- Pedro no es almirante y Pedro es un marino
- Pedro es almirante y Pedro es un marino

Sin embargo, no es coherente pensar que (el caso en que $p \Rightarrow q$ es falso):

- Pedro es almirante y Pedro no es un marino.

Al afirmar que $p\Rightarrow q$ es verdadero, se está afirmando que uno de los tres casos correspondientes, en la tabla de verdad, se dé, i.e., p falso, o p y q verdaderos. Y que no se da el caso de que p sea verdadero y q sea falso. Nótese que si, por otra parte, se puede afirmar que p es verdadero, entonces q debe ser verdadero. Esta discusión sirve, además, como justificación para asumir como correcta la regla de *Modus Ponens*, mencionada al principio de 1.2.3.

Como ayuda mnemotécnica para recordar cuándo una implicación es cierta, se puede pensar que $p\Rightarrow q$ es verdadero cuando p es menos o igual de verdadero que q (entendiendo que lo falso es menos cierto que lo verdadero).

1.3.2 Patrones de prueba

Como ya se advirtió, las pruebas formales hasta ahora definidas pueden resultar difíciles de manipular e, incluso, de entender. La práctica de demostrar da lugar a clasificar las pruebas de acuerdo con estrategias de demostración específicas, lo que da lugar a poder hablar de *patrones de pruebas*. Estos son esquemas lícitos de pruebas, que se pueden justificar como atajos notacionales de pruebas formales básicas.

Se enuncian someramente algunos de los más usuales patrones de prueba, pero después se revisarán de manera formal:

- Prueba *directa*: mostrar un resultado desde axiomas, usando reglas de inferencia.
- Prueba por *contradicción*: mostrar que lo contrario de lo que se quiere probar conlleva a un absurdo (la afirmación de algo flagrantemente falso).
- Prueba por *hipótesis*: cuando se quiere mostrar una implicación, v.gr., $p \Rightarrow q$, se supone que p es cierto (como si fuera un axioma más, "local" a la prueba) y se demuestra q. Más generalmente, si se quiere probar que $p_1 \wedge p_2 \wedge \ldots \wedge p_r \Rightarrow q$, se pueden suponer como ciertos $p_1, p_2, \ldots, p_r$ y se prueba q.
- Prueba por *contrarrecíproca* o *contrapositiva*: si se quiere mostrar que $p \Rightarrow q$, esto es equivalente a mostrar que $\neg q \Rightarrow \neg p$.
- Prueba por *casos*: si se identifican casos que cubran todas las posibilidades y, en cada caso, se logra mostrar que un resultado es verdadero, se puede afirmar que el resultado es siempre verdadero.

1.3.3 Esquema de pruebas

Para las pruebas formales, como deducciones o derivaciones en un sistema lógico, se usa una notación que aquí se quiere introducir con un ejemplo y que, más adelante, se irá refinando y completando para hacer cómodas tanto la escritura como la lectura de las pruebas. En especial, se incluirán variantes para escribir pruebas que sigan patrones como los mencionados en la sección anterior.

Ejemplo A: Una prueba y su notación

El siguiente es un ejemplo de prueba formal en un sistema lógico. Para empezar, el sistema consta de un axioma y tres reglas de inferencia, así[10]:

```
(Ax) 0<1
(R1) a<b       ⊢  a+c<b+c
(R2) a<b, b<c  ⊢  a<c
(R3) a = b     ⊢  E[a]=E[b]
```

La idea es interpretar los símbolos como un lenguaje para denotar fórmulas aritméticas. El `Ax 1` parece plausible. La regla `R1` afirma que si un número `a` es menor que un número `b`, también el número `a+c` siempre será menor al número `b+c`. La regla `R2` enuncia que la relación "ser menor" es transitiva. `R3` es la Regla de Leibniz, ya mencionada en 1.2.

Supóngase que se quiere demostrar que es un teorema 2<3. Una prueba formal:

```
Teo: 2<3
Dem:
          ⟨Ax⟩
   0<1
⇒         ⟨R1, c:= 1⟩
   0+1<1+1
⇒         ⟨Aritmética⟩
   1<2
⇒         ⟨R1, c:= 1⟩
   1+1<2+1
```

[10] El lenguaje formal se supone conocido.

```
⇒         ⟨Aritmética⟩
    2<3
                                                                 []
```

§

El ejemplo ilustra un esquema de pruebas que será utilizado en adelante, con algunas variantes. Conviene destacar algunos elementos del esquema:

- **`Teo`** es un indicador de que sigue la demostración de un teorema.
- `2<3` es el teorema que se demuestra.
- Cada paso de la prueba se escribe en un renglón aparte.
- Cada paso está precedido por una razón. Las razones se ewscriben entre paréntesis angulares ⟨ … ⟩ pueden mencionar axiomas (que son verdaderos por definición), reglas de inferencia o teoremas.
- Otra posibilidad de escribir una razón, no formal, es dar un argumento de sentido común como `⟨Aritmética⟩`. En este caso se espera que el interlocutor acepte esta clase de razones, en aras de simplificar el entendimiento de la prueba.
- Una razón puede tener indicado, como en los usos de `R1`, el remplazo que debe hacerse sobre las variables de la regla, del teorema o de la instanciación que sirven como razón. Además, se puede señalar en qué partes de la fórmula se aplica la razón.
- El símbolo [] se utiliza para señalar que la prueba terminó.

Ahora, supóngase que se quisiera mostrar que `1<3`. Siguiendo la forma de trabajar ya explicada, se podría mostrar un resultado intermedio, auxiliar o lateral, como `1<2`. Teniendo éste, y el teorema anterior, la regla `R2` permite afirmar que también debe valer `1<3`. En esta situación, los resultados auxiliares son llamados *lemas*, i.e., los lemas son teoremas menores que sirven para la demostración de otro “más importante”.

La demostración formal podría ser así:

```
Teo’: 1<3
Dem:
          ⟨Ax⟩
    0<1
⇒         ⟨R1, c:= 1⟩
    0+1<1+1
⇒         ⟨Aritmética⟩
    1<2
⇒         ⟨R2, Teo: 2<3⟩
    1<3
                                                                 []
```

1.3.4 Otras clases de pruebas

La noción de prueba o demostración que aquí se ha dado tienen particularidades que no tiene por qué ser universalmente utilizadas. Para empezar, son pruebas formales y, como tales, dependen del sistema lógico que se usa y de su notación, así como la convención de esquema de pruebas que se adopte.

Ejemplo A: Otro formalismo de prueba

A manera de ilustración, considérese un esquema de prueba diferente del explicado en la sección anterior, para mostrar `1<3`:

```
Teo: 1<3
```

```
Dem:
(1) 0<1          , Axioma
(2) 0+1<1+1      , R1 (1)
(3)  1<2         , R3 (2)
(4) 1+1<2+1      , R1 (3)
(5) 2<3          , R3 (4)
(6) 1<3          , R2 (3,5)
                                                          □
```

Una notación similar a esta se usa, por ejemplo, en [Ros2007]. El esquema escogido tiene algunas ventajas menores que, a medida que el esquema se complete y se haga más expresivo, podrán evidenciarse.

La demostración con esta notación parece más compacta que la del Ejemplo A. Sin embargo, en la notación de ese ejemplo se evidencia una práctica que obliga a usar lo último que se dedujo en la próxima deducción (con ayuda de los lemas, si se requieren). Esta es una razón para preferir esta clase de notaciones para las pruebas que se usan en este texto.

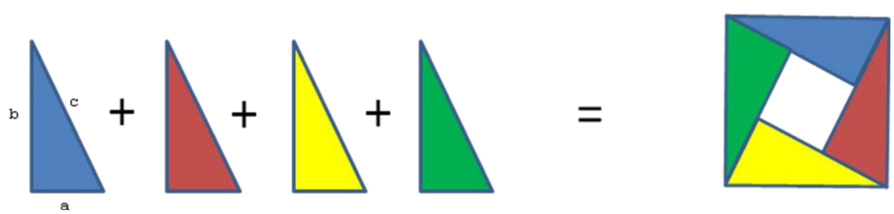
§

Ejemplo B: *Una prueba gráfica*

El Teorema de Pitágoras es conocido por relacionar el tamaño de los cuadrados de los lados de un triángulo rectángulo. Así, si los lados que forman ángulo recto en el triángulo (i.e., los catetos) miden a y b, y el lado restante (la hipotenusa) mide c, se quiere probar que

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

El siguiente dibujo muestra cuatro triángulos rectángulos iguales, con catetos de longitudes a y b e hipotenusas de longitud c. Con los triángulos se arma un cuadrado de lado c, cuya área es igual a la de los cuatro triángulos, más la de un cuadrado de lado b-a.



Es decir:

⟨área del cuadrado externo⟩
c^2
= ⟨4 triángulos de área a*b/2 + cuadrado de lado a-b⟩
$4*(a*b/2) + (a-b)^2$
= ⟨Aritmética⟩
$2*a*b + a^2 - 2*a*b + b^2$
= ⟨Aritmética⟩
$a^2 + b^2$

Aunque muy elegante, esta es una demostración informal del Teorema de Pitágoras. Hay muchas afirmaciones que se aceptan como evidentes que deberían justificarse mejor, si se quisiera una prueba formal. Por ejemplo:

- ¿por qué los triángulos con catetos a y b deben tener hipotenusas iguales?
- ¿por qué se pueden acomodar formando un cuadrado?
- ¿por qué la figura interna es también un cuadrado?
- ¿por qué la suma de las áreas internas es igual a la del cuadrado mayor?
- …

Si se quisiera hacer en este momento una prueba formal de este teorema, se requeriría establecer un sistema formal que modelara propiedades geométricas de triángulos y cuadrados y, en el mejor de los casos, que aprovechara la idea que sugiere la figura. Esto podría ser demasiado dispendioso y, en situaciones como esta, parecería suficiente la figura y la argumentación indicada para convencerse de que el resultado es verdadero. De todas formas, hay que tener consciencia de que la corrección de una argumentación depende de tener claro lo que se acepta como verdadero (los axiomas) y la forma en que se deduzca (las reglas de inferencia).

§

1.3.5 Buenas prácticas al construir pruebas

A continuación se presenta una lista de buenas prácticas para construir pruebas, basada en ideas expuestas en [Mey2010]:

- *Establecer lo que se va a probar*. No dejar de señalar qué es lo que se va a demostrar.
- *Definir un plan de prueba*. Anunciar qué tipo de patrón de prueba se va a usar. Más adelante se podrá anunciar, por ejemplo, que la prueba es por casos, por contradicción, etc.
- *Mantener un flujo lineal de argumentación*. El esquema básico de 1.3.3 promueve esta buena práctica. En las pruebas que así se escriben, cada paso usa una razón para transformar la última fórmula que se pudo demostrar. Esto es ideal, pero no siempre posible. Lo que sí es posible es que se prueben –aparte- teoremas auxiliares (lemas) que sigan el patrón lineal (véase la figura al final de 1.2.2) o usen otros lemas.
- *No olvidar las razones*. Las razones parte esencial de una prueba. Solo deberían poderse omitir si fueran verdasderamente obvias (aunque la obviedad es cuestión de sentido común …).
- *Cuidado con el simbolismo excesivo*. De nuevo, es de sentido común buscar que el interlocutor entienda lo que deba ser la prueba. A veces una explicación informal en lenguaje natural es suficiente para ello.
- *Revisar y simplificar*. Las mejores pruebas son las que se entienden fácilmente. Y esto tiene que ver con que sean cortas y simples.
- *Introducir notación de manera inteligente*. La notación bien escogida para manipular conceptos y resultados puede ser de gran ayuda para la comprensión de las pruebas y de sus resultados. Hay que evitar los excesos de tener que exigir del interlocutor que asimile una gran cantidad de notaciones nuevas que sean complicadas de recordar.
- *Estructurar las pruebas largas*.Esta idea tiene que ver con la de pensar en un plan de trabajo. Una prueba larga y plana es terreno abonado para cometer errores en su construcción o para que el interlocutor se pierda en las argumentaciones. Estructurar la prueba en etapas da lugar a una comprensión de lo global a lo particular que aclara complejidades innecesarias.
- *Usar lemas*. Los lemas son teoremas auxiliares que sirven para apoyar las demostraciones. Deben ser demostrardos, como teoremas que son. Además de servir para mantener el flujo lineal de las pruebas, los lemas sirven para acortar pruebas que repiten una y otra vez el mismo tipo de argumento.
- *Cuidado con lo obvio*. Lo obvio pude dejarse sin razón, lo cual lo marcaría como tal. Así debe hacerse, si hay seguridad de que cualquier interlocutor va a saber por qué una argumentación es obvia. Si esto no es así, la demostración no debe ser considerada tal.
- *Marcar el final*. No dejar de señalar que la demostración se terminó.

1.3.6 Pruebas erradas

Una prueba de un teorema puede contener errores (y, entonces, no ser una prueba) si

- *Se parte de hechos falsos*. Cualquier paso de inferencia va a suponer la verdad de lo que supone, por ejemplo, una regla de inferencia en sus hipótesis. Si esto es así, las inferencias puede llevar a cualquier cosa (cf. 1.3.1)[11].
- *Se usa mal una regla de inferencia*. Si una regla no es aplicable (porque alguna hipótesis no es verdadera, o porque hay condiciones de aplicación que dejan de cumplirse) resultados errados pueden llegar a ser considerados como correctos.

En ocasiones es difícil descubrir dónde está el problema. Esto es más así en la medida que la prueba sea menos formal. En una prueba informal puede no tenerse claro el lenguaje, las reglas de inferencia, los esquemas de prueba lícitos, etc.

Cuando hay una prueba errada, puede pasar cualquier cosa: se puede llegar, porque se tiene mucha suerte, a un resultado verdadero; pero -muy frecuentemente- se termina en resultados absurdos con respecto a la realidad. En los siguientes ejemplos se muestran pruebas que contienen errores. Para dramatizar la situación, los resultados son afirmaciones extrañas a la intuición. Pero no debe olvidarse que, aun terminando en un resultado verdadero, una prueba puede ser incorrecta.

Ejemplo A: Un error aritmético

Teo A: 1/8 > 1/4
Dem:

$$
\begin{array}{ll}
 & \qquad \langle \text{Aritmética} \rangle \\
 & 3 > 2 \\
\Rightarrow & \qquad \langle\ x>y \Rightarrow x*a>y*a\ \rangle \\
 & 3*\log_{10}(1/2) > 2*\log_{10}(1/2) \\
\Rightarrow & \qquad \langle\ u*\log_{10}v = \log_{10} u^{v} \text{, dos veces}\ \rangle \\
 & \log_{10}(1/2)^{3} > \log_{10}(1/2)^{2} \\
\Rightarrow & \qquad \langle\ \log_{10} x > \log_{10} y \ \Rightarrow\ x>y\ \rangle \\
 & (1/2)^{3} > (1/2)^{2} \\
\Rightarrow & \qquad \langle \text{Aritmética} \rangle \\
 & 1/8 > 1/4
\end{array}
$$

[]

¿Qué está mal en esta prueba? Algo debe estar mal, porque no es cierto que 1/8 sea mayor que 1/4. Al revisar con cuidado cada paso, se descubre un problema en la primera inferencia. La razón que aquí se da es la verdad de un teorema sobre números reales que afirme

$$x>y \ \Rightarrow\ x*a>y*a$$

lo cual parece plausible. También hubiera podido argumentarse lo mismo si se tuviera una regla de inferencia como

$$x>y \ \vdash\ x*a>y*a$$

Sin embargo, una regla así no es correcta. Basta mostrar un contraejemplo para ello. Se observa que la implicación

$$2>1 \ \Rightarrow\ 2*(-1)>1*(-1)$$

no es verdadera, porque afirma algo que es falso en los números reales $2>1 \ \Rightarrow\ -2>-1$.

---

[11] Una excepción a lo anterior es el caso de las pruebas por contradicción. En una tal prueba, se supone que lo contrario de o que se quiere demostrar es verdadero y se llega a mostrar que `false` es verdadero. Comon esto no puede ser, se aduce que lo que se quería probar no puede ser flaso, luego debe ser verdadero.

Una variante de la regla sí es corrrecta cuando lo que multiplica a los dos lados de la desigualdad del antecedente es un número mayor que 0. La regla correcta podría enunciarse así

$$\textit{Monotonía *:}\quad \frac{x>y}{x*a>y*a}\ \Big|\ a>0$$

En la demostración del 'Teo A' la regla de Monotonía* no es aplicable, porque $\log_{10}(1/2)<0$.

§

Ejemplo B: Dimensiones

La siguiente 'demostración' comprueba que un centavo (¢1) es igual a un peso ($1). Dentro de la argumentación se usa el hecho de que cien centavos equivalgan a un peso.

Teo B: ¢1 = $1
Dem:

¢1 = $0.01 = ($0.1)^2 = (¢10)^2 = ¢100 = $1

[]

En este caso se trata de una mala aplicación de una regla que debería convertir pesos en centavos. Lo que debería ser correcto tendría la forma:
¢1 = ¢1 * ($1/¢100) = $0.01 = $(0.1)^2 = …

Y tampoco es cierto que (¢10)^2 = ¢100. De hecho, sí es cierto que (¢10)^2 = ¢^{2}100. Este último error es raro de entender, si se trata de imaginar qué es ¢^{2}1; pero si se cree que esto es igual a ¢1, es lo mismo que confundir un metro lineal con un metro cuadrado.

Nótese cómo el evitar enunciar las razones facilita el cometer errores.

§

Ejemplo C: Un error algebraico

El siguiente ejemplo usa una argumentación algebraica para mostrar que si hay dos números iguales, uno de ellos debe ser 0.

Teo C: $a = b \Rightarrow a = 0$
Dem:

$$
\begin{array}{ll}
 & \langle \text{Hipótesis} \rangle \\
 & a = b \\
\Rightarrow & \langle\ x = y \ \Rightarrow\ u*x = u*y;\ a*a = a^2\ \rangle \\
 & a^2 = a*b \\
\Rightarrow & \langle\ x = y \ \Rightarrow\ x-u = y-u\ \rangle \\
 & a^2-b^2 = a*b-b^2 \\
\Rightarrow & \langle\ a^2-b^2 = (a-b)*(a+b);\ \ a*b-b^2 = (a-b)*b\ \rangle \\
 & (a-b)*(a+b) = (a-b)*b \\
\Rightarrow & \langle\ x*y = x*z \ \Rightarrow\ y = z\ \rangle \\
 & a+b = b \\
\Rightarrow & \langle\ x+c = c \ \Rightarrow\ x = 0\ \rangle \\
 & a = 0
\end{array}
$$

[]

La cancelación del factor a-b corresponde a una división por 0. De nuevo, el error está en aplicar una regla en condiciones que no debe hacerse.

§

Ejemplo D: ¿Qué está mal?

La siguiente prueba informal es conocida, entre otros nombres, como la *paradoja del examen inesperado*[12].

Teo D: "No se puede dar un *quiz* que se ha anunciado como sorpresivo"

El profesor de una clase que se dicta de lunes a viernes anuncia un viernes que algún día de la próxima semana hará un *quiz* sorpresivo. Suponiendo que el profesor siempre dice la verdad, los estudiantes demuestran que el quiz no puede llegar a darse.

Dem:

El quiz no puede tener lugar el viernes, porque no sería sorpresivo, dado que -si no se hizo entre lunes y jueves- por fuerza, tendría lugar el viernes, y no sería sorpresivo. Sabiendo que no es el viernes, tampoco puede ser el jueves, porque si no sucedió entre lunes y miércoles, debería ser el jueves ... Y así, hasta llegar a la conclusión de que no puede ser el lunes.

[]

En este caso no hay una demostración formal, pero hay una argumentación que puede ser o no correcta. La verdad es que el profesor puede llegar un día, por ejemplo, el lunes o el miércoles, y hacer el *quiz*, sin que se pierda la calidad de sorpresivo. ¿Qué está mal en el razonamiento de los estudiantes?

§

[12] Hay muchas variantes y refraseos. Véase, por ejemplo, `http://es.wikipedia.org/wiki/Paradoja_del_examen_sorpresa` .