LOS TEOREMAS DE INCOMPLETITUD DE GÖDEL

ALEXANDRE MIQUEL

En este artículo se presenta la demostración de los dos teoremas de incompletitud de Gödel, que establecen los límites fundamentales del razonamiento formal. Aunque este resultado se cumpla para una clase grande de sistemas formales (incluyendo las teorías de conjuntos ZF y ZFC), sólo vamos a presentar la demostración de ambos teoremas para el caso fundamental de la *Aritmética de Peano* (PA), que ya contiene toda la sustancia del argumento de Gödel:

> **Primer teorema de incompletitud** — *Si* PA *es consistente, entonces* PA *es incompleta, en el sentido en que existe en el lenguaje de* PA *una fórmula cerrada G que no se puede ni demostrar ni refutar en* PA*:* $\mathrm{PA} \nvdash G$ *y* $\mathrm{PA} \nvdash \neg G$.

> **Segundo teorema de incompletitud** — *Si* PA *es consistente, entonces la fórmula cerrada Cons*PA *que expresa la consistencia de* PA *en el lenguaje de* PA *no se puede demostrar en* PA*:* $\mathrm{PA} \nvdash \mathit{Cons}\mathrm{PA}$.

En la sección 6, veremos que la demostración de estos dos teoremas se generaliza naturalmente a cualquier teoría recursiva que contiene la Aritmética de Peano.

Índice

1. La Aritmética de Peano (PA) 2
1.1. Lenguaje de la Aritmética 2
1.2. Sustitución y α-equivalencia 4
1.3. Sistema de deducción 5
1.4. Axiomas y teoremas 8
1.5. Consistencia y completitud 10
1.6. Expresividad de PA 11
1.7. El modelo estándar de PA 13
1.8. Propiedades de completitud de PA 15
2. Funciones recursivas 19
2.1. Definiciones y propiedades 20
2.2. Conjuntos recursivos 22
2.3. El teorema de representación 23
2.4. Teorema chino del resto y función β de Gödel 24
3. Digitalización de PA 27
3.1. Abismación 27
3.2. Emparejamiento 27
3.3. Codificación de las listas finitas 28
3.4. Digitalización de los términos 29
3.5. Digitalización de las fórmulas 30
3.6. Digitalización de las derivaciones 32
3.7. Algunos conjuntos recursivos y no recursivos 34
4. El primer teorema de incompletitud 35
4.1. Construcción de fórmulas autorreferenciales 35
4.2. Una fórmula G no demostrable pero verdadera 36

4.3. La hipótesis de 1-consistencia 37
4.4. El truco de Rosser 38
4.5. La noción de demostrabilidad no es recursiva 39
4.6. La noción de verdad no es definible en PA 40
4.7. Conclusión 41
5. El segundo teorema de incompletitud 41
5.1. El argumento de Gödel 41
5.2. Las condiciones de derivabilidad de Hilbert-Bernays 42
5.3. El teorema de Löb 44
6. Generalización del los teoremas de incompletitud 44
6.1. Lenguaje y teorías de primer orden 45
6.2. La Aritmética de Robinson (Q) 46
6.3. Inmersión de teorías 47
6.4. La Aritmética de Presburger (P) 48
Referencias 48

1. La Aritmética de Peano (PA)

La Aritmética de Peano (notación: PA) es una teoría de primer orden[1] definida —como toda teoría de primer orden— por su lenguaje (de primer orden) y su sistema de axiomas.

1.1. Lenguaje de la Aritmética. El lenguaje de PA distingue dos tipos de expresiones:

- Los *términos* (notación: t, u, v, etc.), que representan enteros naturales.
- Las *fórmulas* (notación: ϕ, ψ, χ, etc.), que representan aserciones sobre enteros naturales.

Los términos y las fórmulas del lenguaje de PA son expresiones finitas construidas a partir de los siguientes símbolos primitivos:

- Las variables (notación: x, y, z, etc.) En lo siguiente, se supone dado una cantidad infinita numerable de variables indizadas por los enteros naturales: $\mathtt{x}_0$, $\mathtt{x}_1$, $\mathtt{x}_2$, $\mathtt{x}_3$, etc.
- El símbolo de constante 0 (« cero »).
- El símbolo de función unaria s (« sucesor »).
- Los símbolos de funciones binarias $+$ (« adición ») y $\times$ (« multiplicación »).
- El símbolo de relación binaria $=$ (« igualdad »).
- La constante proposicional $\perp$ (« fórmula absurda »).
- La conectiva binaria $\Rightarrow$ (« implicación »).
- El cuantificador $\forall$ (« cuantificador universal »).

Veremos más abajo (Notaciones 1.4) que las otras construcciones usuales de la lógica —negación, conjunción, disyunción, equivalencia lógica, cuantificador existencial— se pueden definir a partir de estos símbolos primitivos mediante de las leyes de De Morgan.

Definición 1.1 (Términos)**.** — Los *términos* (notación: t, u, v, etc.) son las expresiones construidas por aplicación finita de las siguientes reglas:

(1) Si x es una variable, entonces x es un término.
(2) El símbolo de constante 0 es un término.
(3) Si t es un término, entonces $s(t)$ (« sucesor de t ») es un término.
(4) Si t, u son términos, entonces $t + u$ (« t más u ») es un término.
(5) Si t, u son términos, entonces $t \times u$ (« t por u ») es un término.

[1] Una definición general de la noción de teoría de primer orden es dada en la Sección 6.1.

El *conjunto de las variables libres* de un término t, escrito $FV(t)$, es definido por inducción estructural sobre t por las ecuaciones:

$$\begin{array}{rcl} FV(x) & = & \{x\} \\ FV(0) & = & \varnothing \\ FV(s(t)) & = & FV(t) \\ FV(t+u) & = & FV(t) \cup FV(u) \\ FV(t \times u) & = & FV(t) \cup FV(u) \end{array}$$

Se observa que en un término t, todas las ocurrencias de variables son libres, de tal modo que el conjunto (finito) $FV(t)$ contiene todas las variables que ocurren al menos una vez en t. Se dice que un término t es *cerrado* cuando $FV(t) = \varnothing$, y que es *abierto* si no.

En lo siguiente, se utiliza la notación $t \equiv u$ (antes de $t = u$) para expresar que dos términos t y u son idénticos, afín de no confundir con la fórmula $t = u$ construida a partir de t y u mediante del símbolo de igualdad = (véase Def. 1.3 más abajo).

Notación 1.2 (Enteros de Peano)**.** — En lo siguiente, se utilizan las abreviaturas $1 \equiv s(0)$, $2 \equiv s(1)$, $3 \equiv s(2)$, etc. Más generalmente, a cada entero natural n se asocia el término cerrado $\overline{n}$ llamado el *entero de Peano asociado a n* y definido por:

$$\overline{n} \equiv \underbrace{s(\cdots s}_{n \text{ veces}}(0)\cdots)$$

Intuitivamente, el entero de Peano $\overline{n}$ es la representación del entero natural n en el lenguaje de PA, como término cerrado.

Definición 1.3 (Fórmulas)**.** — Las *fórmulas* (notación: ϕ, ψ, χ, etc.) son las expresiones construidas por aplicación finita de las siguientes reglas:

(1) $\bot$ (« el absurdo ») es una fórmula.
(2) Si t, u son términos, entonces $t = u$ (« t es igual a u ») es una fórmula.
(3) Si ϕ, ψ son fórmulas, entonces $\phi \Rightarrow \psi$ (« ϕ implica ψ ») es una fórmula.
(4) Si x es una variable y ϕ una fórmula, entonces $\forall x\phi$ (« para todo x, ϕ ») es una fórmula.

Como para los términos, se utiliza la notación $\phi \equiv \psi$ (antes de $\phi = \psi$) para expresar que dos fórmulas ϕ y ψ son idénticas.

Notaciones 1.4. (Construcciones definidas) — En este lenguaje minimal, las otras conectivas y la cuantificación existencial se pueden definir por las leyes de De Morgan:

$$\begin{array}{rcll} \top & \equiv & \bot \Rightarrow \bot & \text{(Fórmula trivial)} \\ \neg\phi & \equiv & \phi \Rightarrow \bot & \text{(Negación)} \\ \phi \wedge \psi & \equiv & \neg(\phi \Rightarrow \psi \Rightarrow \bot) & \text{(Conjunción)} \\ \phi \vee \psi & \equiv & \neg\phi \Rightarrow \neg\psi \Rightarrow \bot & \text{(Disyunción)} \\ \phi \Leftrightarrow \psi & \equiv & (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi) & \text{(Equivalencia lógica)} \\ \exists x\,\phi & \equiv & \neg\forall x\,\neg\phi & \text{(Cuantificación existencial)} \end{array}$$

(Como de costumbre, se considera que el símbolo de implicación $\Rightarrow$ se asocia implícitamente a la derecha, leyendo $\phi \Rightarrow \psi \Rightarrow \chi$ como $\phi \Rightarrow (\psi \Rightarrow \chi)$.)

Contrario a los términos, las fórmulas contienen dos tipos de ocurrencias de variables: las ocurrencias *ligadas*, que se encuentran bajo una cuantificación con el mismo nombre de variable, y las otras ocurrencias, que son dichas *libres*. Así, el *conjunto de las variables libres* de una fórmula ϕ, escrito $FV(\phi)$, es definido por inducción estructural sobre ϕ por las ecuaciones:

$$\begin{array}{rcl} FV(\bot) & = & \varnothing \\ FV(t = u) & = & FV(t) \cup FV(u) \\ FV(\phi \Rightarrow \psi) & = & FV(\phi) \cup FV(\psi) \\ FV(\forall x\, \phi) & = & FV(\phi) \setminus \{x\} \end{array}$$

Se dice que una fórmula ϕ es *cerrada* cuando $FV(\phi) = \varnothing$, y que es *abierta* si no.

1.2. Sustitución y α-equivalencia. Cada variable libre x que ocurre en un término t o en una fórmula ϕ representa un entero incógnito que se puede sustituir por un entero concreto (un entero de Peano), y más generalmente por cualquier término u, cerrado o abierto. En el lenguaje de PA (como en cualquier lenguaje de primer orden), esta noción de sustitución se formaliza mediante de dos operaciones:

- Una operación de *sustitución en los términos*, escrita $t\{x := u\}$, que consiste en sustituir en el término t cada ocurrencia libre de la variable x por el término u.
- Una operación de *sustitución en las fórmulas*, escrita $\phi\{x := u\}$, que consiste en sustituir en la fórmula ϕ cada ocurrencia libre de la variable x por el término u.

Como los términos sólo contienen ocurrencias libres de variables, la operación de sustitución $t\{x := u\}$ en los términos se define inmediatamente por inducción estructural sobre t:

Definición 1.5 (Sustitución en un término)**.** — Dado dos término t, u y una variable x, se escribe $t\{x := u\}$ el término definido por inducción estructural sobre t por las ecuaciones:

$$\begin{array}{rcll} x\{x := u\} & \equiv & u & \\ y\{x := u\} & \equiv & y & \qquad (\text{si } y \not\equiv x) \\ 0\{x := u\} & \equiv & 0 & \\ (s(t))\{x := u\} & \equiv & s(t\{x := u\}) & \\ (t_1 + t_2)\{x := u\} & \equiv & t_1\{x := u\} + t_2\{x := u\} & \\ (t_1 \times t_2)\{x := u\} & \equiv & t_1\{x := u\} \times t_2\{x := u\} & \end{array}$$

La definición de la operación de sustitución $\phi\{x := u\}$ en las fórmulas es más delicada que en los términos, debido a la presencia de cuantificaciones que inducen un riesgo de *captura de variable*. Este problema se puede entender con el siguiente ejemplo.

Ejemplo 1.6. — Consideremos la fórmula

$$\phi \ \equiv\ \exists y\,(x = 2 \times y) \qquad\qquad (\equiv \neg\forall y \neg(x = 2 \times y))$$

que sólo depende de la variable libre x, y que expresa que « x es par ». Si se remplaza ingenuamente la variable x por el término $u \equiv y + 3$, se obtiene una fórmula

$$\phi' \ \equiv\ \exists y\,(y + 3 = 2 \times y)$$

que no expresa que « $y + 3$ es par », pero que tiene un sentido muy diferente. (De hecho, esta fórmula ya no depende de y.) Aquí, el problema viene de que durante la sustitución, la variable y fue capturada por el cuantificador $\exists$, induciendo una confusión entre la variable cuantificada y la variable y que viene del término u, la cual refiere a algún objeto afuera de la cuantificación. Para evitar tal captura, se necesita cambiar el nombre de la variable ligada (por una nueva variable z) antes de remplazar las ocurrencias de x, de tal modo que se obtenga la fórmula

$$\phi\{x := y + 3\} \ \equiv\ \exists z\,(y + 3 = 2 \times z)$$

que tiene el sentido deseado (« $y + 3$ es par »).

Ahora se puede dar la definición formal de la operación $\phi\{x := u\}$.

Definición 1.7 (Sustitución en una fórmula)**.** — Dado una fórmula ϕ, una variable x y un término u, se escribe $\phi\{x := u\}$ la fórmula definida por inducción sobre el número de conectivas y de cuantificadores en ϕ por las siguientes ecuaciones:

$$\begin{array}{rcll}
\bot\{x := u\} & \equiv & \bot & \\
(t_1 = t_2)\{x := u\} & \equiv & t_1\{x := u\} = t_2\{x := u\} & \\
(\phi \Rightarrow \psi)\{x := u\} & \equiv & \phi\{x := u\} \Rightarrow \psi\{x := u\} & \\
(\forall x\,\phi)\{x := u\} & \equiv & \forall x\,\phi & \\
(\forall y\,\phi)\{x := u\} & \equiv & \forall x\,\phi\{x := u\} & (\text{si } y \not\equiv x,\ y \notin FV(u)) \\
(\forall y\,\phi)\{x := u\} & \equiv & \forall z\,\phi\{y := z\}\{x := u\} & (\text{si } y \not\equiv x,\ y \in FV(u))
\end{array}$$

eligiendo la nueva variable z (en el último caso) de tal modo que $z \notin FV(\phi)$ y $z \notin FV(u)$.

Se observa que para definir la sustitución adentro de una fórmula de la forma $\forall y\,\phi$ (cuantificación universal), se necesita distinguir tres casos:

- El caso donde $y \equiv x$ (la variable cuantificada coincide con la variable sustituida). En este caso, la variable x no es libre en $\forall x\,\phi$, de tal modo que la sustitución es inoperante.
- El caso donde $y \not\equiv x$ y $y \notin FV(u)$. En este caso, no hay ningún riesgo de captura; por lo tanto se puede hacer el remplazo directamente.
- El caso donde $y \not\equiv x$ y $y \in FV(u)$. Es el caso problemático, donde se necesita cambiar la variable ligada y por otra variable z (que no es libre ni en ϕ ni en u —se dice que z es una variable fresca), antes de remplazar las ocurrencias de x por u.

En virtud del cambio de variable ligada en el último caso, la definición de la operación de sustitución $\phi\{x := u\}$ en las fórmulas es definida al menos de los nombres de variables ligadas. En otro lado, es claro que dos fórmulas que sólo difieren por los nombres de variables ligadas (por ejemplo: las fórmulas $\exists y\,(x = 2{\times}y)$ y $\exists z\,(x = 2{\times}z)$) tienen el mismo sentido (« x es par ») y son intercambiables. Afín de identificar tales fórmulas, se define la relación de *α-equivalencia*, que expresa que dos fórmulas sólo difieren por los nombres de variables ligadas:

Definición 1.8 (α-equivalencia)**.** — La relación $\phi \equiv_\alpha \psi$ de *α-equivalencia* entre dos fórmulas ϕ y ψ es definida por inducción sobre el número de conectivas y de cuantificadores en las dos fórmulas ϕ y ψ a partir de las siguientes reglas:

- Si $\phi \equiv \bot$ y $\psi \equiv \bot$, entonces $\phi \equiv_\alpha \psi$.
- Si $\phi \equiv (t_1 = t_2)$ y $\psi \equiv (u_1 = u_2)$, entonces $\phi \equiv_\alpha \psi$ si y sólo si $t_1 \equiv u_1$ y $t_2 \equiv u_2$.
- Si $\phi \equiv (\phi_1 \Rightarrow \phi_2)$ y $\psi \equiv (\psi_1 \Rightarrow \psi_2)$, entonces $\phi \equiv_\alpha \psi$ si y sólo si $\phi_1 \equiv_\alpha \psi_1$ y $\phi_2 \equiv_\alpha \psi_2$.
- Si $\phi \equiv \forall x\,\phi_1$ y $\psi \equiv \forall y\,\psi_1$, entonces $\phi \equiv_\alpha \psi$ si y sólo si $\phi_1\{x := z\} \equiv_\alpha \psi_1\{y := z\}$, donde z es cualquier variable tal que $z \notin FV(\phi)$ y $z \notin FV(\psi)$ ("variable fresca").
- En todos los otros casos, la relación $\phi \equiv_\alpha \psi$ no se cumple.

1.3. Sistema de deducción. La noción de deducción (en lógica clásica) se puede definir a través de muchos sistemas formales (sistemas de Hilbert, deducción natural, cálculo de secuentes de Gentzen, etc.) que últimamente definen la misma noción de *consecuencia lógica*. Aquí, se adopta un sistema de *deducción natural* (en lógica clásica) con secuentes explícitos, restringido a nuestro lenguaje minimal de fórmulas. Formalmente:

Definición 1.9 (Secuentes)**.** — Se llama un *secuente* todo par (Γ, ϕ) escrito

$$\Gamma \vdash \phi \qquad\qquad (\text{“}\Gamma \text{ lleva a } \phi\text{”})$$

donde Γ es una lista finita de fórmulas (posiblemente vacía) y ϕ una fórmula.

Intuitivamente, un secuente $\Gamma \vdash \phi$ representa una etapa particular de razonamiento, en la cual se trata de establecer ϕ (el *consecuente*) a partir de las hipótesis en Γ (el *antecedente*).

Axioma:	$\overline{\Gamma \vdash \phi}$	si $\phi \in \Gamma$
Debilitamiento:	$\dfrac{\Gamma \vdash \phi}{\Gamma' \vdash \phi}$	si $\Gamma \subseteq \Gamma'$
Introducción de $\Rightarrow$	$\dfrac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \Rightarrow \psi}$	
Eliminación de $\Rightarrow$ (*modus ponens*)	$\dfrac{\Gamma \vdash \phi \Rightarrow \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}$	
Introducción de $\forall$ (**generalización**)	$\dfrac{\Gamma \vdash \phi\{x := y\}}{\Gamma \vdash \forall x\, \phi}$	si $y \notin FV(\Gamma, \forall x\, \phi)$
Eliminación de $\forall$ (**instanciación**)	$\dfrac{\Gamma \vdash \forall x\, \phi}{\Gamma \vdash \phi\{x := t\}}$	
Introducción de $=$ (**principio de identidad**)	$\overline{\Gamma \vdash t = t}$	
Eliminación de $=$ (**remplazo de los iguales**)	$\dfrac{\Gamma \vdash t = u \qquad \Gamma \vdash \phi\{x := t\}}{\Gamma \vdash \phi\{x := u\}}$	
Razonamiento por el absurdo	$\dfrac{\Gamma, \neg\phi \vdash \bot}{\Gamma \vdash \phi}$	

Cuadro 1. Reglas de deducción

En lo siguiente, las listas finitas de fórmulas —igualmente llamadas *contextos*— se escriben $\Gamma \equiv \phi_1, \ldots, \phi_n$ (fórmulas separadas por comas), y la concatenación de dos listas Γ y Δ se escribe Γ, Δ. Un secuente con un antecedente vacío se escribe simplemente $\vdash \phi$ (antes de $\emptyset \vdash \phi$). Dado listas finitas de fórmulas Γ y Δ, se escribe $\Gamma \subseteq \Delta$ cuando cada fórmula que ocurre en Γ también ocurre en Δ (sin tener en cuenta ni el orden ni las repeticiones). Las notaciones $FV(\Gamma)$ (variables libres) y $\Gamma\{x := u\}$ (sustitución) se extienden a las listas finitas de fórmulas del modo obvio: si $\Gamma \equiv \phi_1, \ldots, \phi_n$, entonces

$$\begin{array}{rcl} FV(\Gamma) & = & FV(\phi_1) \cup \cdots \cup FV(\phi_n) \\ \Gamma\{x := u\} & \equiv & \phi_1\{x := u\}, \ldots, \phi_n\{x := u\} \end{array}$$

El sistema de deducción que utilizaremos aquí es definido a partir de las 9 *reglas de deducción* dadas en el Cuadro 1. Cada (instancia de una) regla de deducción es de la forma

$$\frac{\Gamma_1 \vdash \phi_1 \quad \cdots \quad \Gamma_p \vdash \phi_p}{\Gamma \vdash \phi}$$

donde los secuentes $\Gamma_1 \vdash \phi_1, \ldots, \Gamma_p \vdash \phi_p$ se llaman las *premisas* de la regla, y el secuente $\Gamma \vdash \phi$ la *conclusión* de la regla. Intuitivamente, tal regla expresa un paso de deducción que permite establecer la validez de la conclusión a partir de la validez de las premisas, es decir:

como: ϕ_1 *se cumple (bajo las hipótesis* Γ_1*), y*
$\vdots$
ϕ_p *se cumple (bajo las hipótesis* Γ_p*);*
luego: ϕ *se cumple (bajo las hipótesis* Γ*)*

Algunas reglas tienen una *condición de borde* (a la derecha), que restringe el uso de la regla a elementos sintácticos con una forma particular. Por ejemplo, la regla de introducción de $\forall$ (generalización) restringe el uso de la regla a los Γ, ϕ tales que la variable y no es libre ni en Γ ni en $\forall x\,\phi$. Intuitivamente, esta condición expresa que la variable y es fresca respecto a la lista de hipótesis Γ (y a la fórmula $\forall x\,\phi$), de tal modo que dicha variable pueda representar un entero "cualquiera" —es decir: un entero sobre el cual no se plantea ninguna hipótesis. (Además, la condición de frescura $y \notin FV(\Gamma, \forall x\,\phi)$ sobre la variable y es formulada de tal modo que se pueda tomar $y \equiv x$ cuando la condición de frescura $x \notin FV(\Gamma)$ ya se cumple para x.)

Las reglas de deducción son los ladrillos elementales del razonamiento formal, y se pueden agregar hasta formar *árboles de derivación*, o *derivaciones*. Formalmente:

Definición 1.10 (Derivaciones)**.** — Se llama una *derivación* todo árbol finito de secuentes construido por aplicación finita de la siguiente regla de agregación:

- Si

  $$\begin{array}{ccc} \vdots\, d_1 & & \vdots\, d_p \\ \Gamma_1 \vdash \phi_1 & \cdots & \Gamma_p \vdash \phi_p \end{array}$$

  son derivaciones de los secuentes $\Gamma_1 \vdash \phi_1, \ldots, \Gamma_p \vdash \phi_p$ (respectivamente), y si

  $$\frac{\Gamma_1 \vdash \phi_1 \quad \cdots \quad \Gamma_p \vdash \phi_p}{\Gamma \vdash \phi}$$

  es (una instancia de) una de las 9 reglas del Cuadro 1, entonces el árbol

  $$d \equiv \left\{ \frac{\begin{array}{ccc} \vdots\, d_1 & & \vdots\, d_p \\ \Gamma_1 \vdash \phi_1 & \cdots & \Gamma_p \vdash \phi_p \end{array}}{\Gamma \vdash \phi} \right.$$

  es una derivación del secuente $\Gamma \vdash \phi$.

La relación « d es una derivación del secuente $\Gamma \vdash \phi$ » se escribe $d : (\Gamma \vdash \phi)$. Cuando tal derivación existe, se dice que el secuente $\Gamma \vdash \phi$ es *derivable*.

Analogía 1.11. — Se puede ver un sistema de deducción como un juego de construcción en el cual cada regla de deducción define una familia (infinita) de ladrillos. En este juego, cada ladrillo tiene un único enchufe abajo (la conclusión) y cero o más tomas arriba (las premisas), la forma de los enchufes y de las tomas siendo determinada por las fórmulas correspondientes. Según esta analogía, una derivación es un montaje de ladrillos en el cual sólo se queda un enchufe libre (abajo): la conclusión de la derivación.

Ejemplo 1.12. — El siguiente árbol

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{x = y \vdash x = y}\ \text{(axioma)} \qquad \cfrac{}{x = y \vdash x = x}\ \text{(=-intro)}}{x = y \vdash y = x}\ \text{(=-elim, con } \phi(z) \equiv z{=}x)}{\vdash x = y \Rightarrow y = x}\ \text{(}\Rightarrow\text{-intro)}}{\vdash \forall y\,(x = y \Rightarrow y = x)}\ \text{(}\forall\text{-intro)}}{\vdash \forall x\,\forall y\,(x = y \Rightarrow y = x)}\ \text{(}\forall\text{-intro)}$$

es una derivación del secuente $\vdash \forall x\,\forall y\,(x = y \Rightarrow y = x)$ (sin hipótesis).

Ejemplo 1.13. — A partir de dos derivaciones $d_1 : (\Gamma \vdash \phi)$ y $d_2 : (\Gamma \vdash \psi)$ (de los secuentes $\Gamma \vdash \phi$ y $\Gamma \vdash \psi$), se puede construir la siguiente derivación

$$
\cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma, \phi \Rightarrow \psi \Rightarrow \bot \vdash \phi \Rightarrow \psi \Rightarrow \bot}\ (\text{ax.}) \qquad \cfrac{\begin{array}{c}\vdots\; d_1\\ \Gamma \vdash \phi\end{array}}{\Gamma, \phi \Rightarrow \psi \Rightarrow \bot \vdash \phi}\ (\text{debil.})}{\Gamma, \phi \Rightarrow \psi \Rightarrow \bot \vdash \psi \Rightarrow \bot}\ (\Rightarrow\text{-intro}) \qquad \cfrac{\begin{array}{c}\vdots\; d_2\\ \Gamma \vdash \psi\end{array}}{\Gamma, \phi \Rightarrow \psi \Rightarrow \bot \vdash \psi}\ (\text{debil.})}{\Gamma, \phi \Rightarrow \psi \Rightarrow \bot \vdash \bot}\ (\Rightarrow\text{-elim})}{\Gamma \vdash \underbrace{\neg(\phi \Rightarrow \psi \Rightarrow \bot)}_{\phi \wedge \psi}}\ (\Rightarrow\text{-intro})
$$

del secuente $\Gamma \vdash \phi \wedge \psi$. El trozo de derivación entre las derivaciones d_1, d_2 y la conclusión $\Gamma \vdash \phi \wedge \psi$ se puede ver como una *regla compuesta* o *macroregla*

$$
\begin{array}{c}
\begin{array}{c}\vdots\; d_1\\ \Gamma \vdash \phi\end{array} \qquad \begin{array}{c}\vdots\; d_2\\ \Gamma \vdash \psi\end{array} \\
\hline\hline
\Gamma \vdash \phi \wedge \psi
\end{array}
$$

que construye una derivación de $\Gamma \vdash \phi \wedge \psi$ a partir de derivaciones de $\Gamma \vdash \phi$ y de $\Gamma \vdash \psi$.

El ejemplo anterior sugiere que las reglas de deducción se pueden agregar para formar *reglas compuestas*, igualmente llamadas *reglas derivables*. Formalmente, se dice que una regla

$$
\begin{array}{c}
\Gamma_1 \vdash \phi_1 \quad \cdots \quad \Gamma_p \vdash \phi_p \\
\hline\hline
\Gamma \vdash \phi
\end{array}
$$

es *derivable* (lo que se indica con una doble raya de inferencia) cuando existe una derivación incompleta del secuente $\Gamma \vdash \phi$ en la cual los únicos secuentes no justificados (las "tomas" libres) son los secuentes $\Gamma_1 \vdash \phi_1, \ldots, \Gamma_p \vdash \phi_p$. Es claro que tal derivación incompleta siempre se puede completar, enchufando derivaciones (completas) $d_1, \ldots, d_p$ de las premisas $\Gamma_1 \vdash \phi_1, \ldots, \Gamma_p \vdash \phi_p$ en las "tomas" correspondientes —cuando tales derivaciones existen.

Las reglas primitivas del Cuadro 1 sólo conciernen las construcciones del lenguaje minimal: $\bot$, $\Rightarrow$, $\forall$. Sin embargo, el mecanismo de agregación de reglas descrito anteriormente permite derivar las reglas de deducción para las construcciones definidas: $\neg$, $\wedge$, $\vee$ y $\exists$.

Proposición 1.14. — *Las reglas de deducción dadas en el Cuadro 2 son derivables.*

Demostración. Ejercicio. □

1.4. Axiomas y teoremas. El sistema de deducción presentado en la sección 1.3 es un sistema muy general que define el sentido de las construcciones lógicas (primitivas y definidas) y de la igualdad en cualquier teoría de primer orden. (No tiene nada específico para la Aritmética.) Para finalizar la definición de la Aritmética de Peano (PA), se necesita dar sus axiomas —los *axiomas de Peano*— que sirven a definir los objetos a los cuales se refieren implícitamente las variables (aquí: los enteros naturales), así como el sentido de los símbolos específicos de PA: 0, s, $+$, $\times$. Formalmente, los axiomas de Peano son las siguientes fórmulas cerradas:

(A_1) $\forall y\,(0 + y = y)$ $\qquad$ (A_2) $\forall x\,\forall y\,(s(x) + y = s(x + y))$

(A_3) $\forall y\,(0 \times y = 0)$ $\qquad$ (A_4) $\forall x\,\forall y\,(s(x) \times y = (x \times y) + y)$

(A_5) $\forall x\,(s(x) \neq 0)$ $\qquad$ (A_6) $\forall x\,\forall y\,(s(x) = s(y) \Rightarrow x = y)$

(A_7) $\forall z_1 \cdots \forall z_k\,(\phi\{x := 0\} \wedge \forall x\,(\phi \Rightarrow \phi\{x := s(x)\}) \Rightarrow \forall x\,\phi)$
para cada fórmula ϕ con las variables libres $x, z_1, \ldots, z_k$

Introducción de $\top$	$\overline{\overline{\Gamma \vdash \top}}$	
Eliminación de $\bot$ (***ex falso quod libet***)	$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash \phi}$	
Introducción de $\neg$	$\dfrac{\Gamma, \phi \vdash \bot}{\Gamma \vdash \neg\phi}$	
Eliminación de $\neg$ (**contradicción**)	$\dfrac{\Gamma \vdash \phi \quad \Gamma \vdash \neg\phi}{\Gamma \vdash \bot}$	
Introducción de $\wedge$	$\dfrac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}$	
Eliminación de $\wedge$ (**izquierda y derecha**)	$\dfrac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \qquad \dfrac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}$	
Introducción de $\vee$ (**izquierda y derecha**)	$\dfrac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \qquad \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}$	
Eleminación de $\vee$ (**razonamiento por casos**)	$\dfrac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \chi \quad \Gamma, \psi \vdash \chi}{\Gamma \vdash \chi}$	
Introducción de $\Leftrightarrow$	$\dfrac{\Gamma, \phi \vdash \psi \quad \Gamma, \psi \vdash \phi}{\Gamma \vdash \phi \Leftrightarrow \psi}$	
Eliminación de $\Leftrightarrow$ (**directa y recíproca**)	$\dfrac{\Gamma \vdash \phi \Leftrightarrow \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi} \qquad \dfrac{\Gamma \vdash \phi \Leftrightarrow \psi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi}$	
Introducción de $\exists$	$\dfrac{\Gamma \vdash \phi\{x := t\}}{\Gamma \vdash \exists x\, \phi}$	
Eliminación de $\exists$	$\dfrac{\Gamma \vdash \exists x\, \phi \quad \Gamma, \phi\{x := y\} \vdash \psi}{\Gamma \vdash \psi}$	si $y \notin FV(\Gamma, \exists x\, \phi, \psi)$
Symetría de $=$	$\dfrac{\Gamma \vdash t = u}{\Gamma \vdash u = t}$	
Transitividad de $=$	$\dfrac{\Gamma \vdash t = u \quad \Gamma \vdash u = v}{\Gamma \vdash t = v}$	

CUADRO 2. Reglas derivadas

Observaciones 1.15. — Los axiomas (A_1)–(A_4) expresan las ecuaciones recursivas que definen la adición y la multiplicación sobre los enteros naturales. Como observó Hilbert, estos axiomas son « definiciones escondidas » de los símbolos $+$ y $\times$.

Los axiomas (A_5) y (A_6) expresan que 0 no es el sucesor de ningún objeto, y que la función sucesor es inyectiva. Dicho de otra manera, estos dos axiomas expresan que el « universo del discurso » de la Aritmética de Peano (PA) es infinito, en el sentido de Dedekind[2].

[2] Se recuerda que un conjunto X es infinito (en el sentido de Dedekind) si existe una función $f : X \to X$ que es inyectiva pero no sobreyectiva. Es exactamente lo que expresan los axiomas (A_5) y (A_6).

Sucesor	$\dfrac{\mathrm{PA},\Gamma \vdash s(t) = s(u)}{\mathrm{PA},\Gamma \vdash t = u}$	$\dfrac{\mathrm{PA},\Gamma \vdash s(t) = 0}{\mathrm{PA},\Gamma \vdash \bot}$	
Adición	$\overline{\mathrm{PA},\Gamma \vdash 0 + u = u}$	$\overline{\mathrm{PA},\Gamma \vdash s(t) + u = s(t + u)}$	
Multiplicación	$\overline{\mathrm{PA},\Gamma \vdash 0 \times u = 0}$	$\overline{\mathrm{PA},\Gamma \vdash s(t) \times u = (t \times u) + u}$	
Inducción	$\dfrac{\mathrm{PA},\Gamma \vdash \phi\{x := 0\} \quad \mathrm{PA},\Gamma,\phi\{x := y\} \vdash \phi\{x := s(y)\}}{\mathrm{PA},\Gamma \vdash \phi\{x := t\}}$		si $y \notin FV(\Gamma, \forall x\, \phi)$

CUADRO 3. Reglas aritméticas

Finalmente, el axioma (A_7) —el *principio de inducción*— no es estrictamente hablando un axioma, pero un *esquema de axiomas*, que define un axioma de inducción particular para cada fórmula ϕ con las variables libres $x, z_1, \ldots, z_k$. Así, en razón de la presencia de los axiomas de inducción, el conjunto de los axiomas de PA es infinito numerable:

$$\underbrace{(A_1)\ \cdots\ (A_6)}_{6\ \text{axiomas}} + \underbrace{(A_7)}_{\omega\ \text{axiomas}}$$

En lo siguiente, se escribe Ax(PA) el conjunto (infinito) de los axiomas de PA.

Definición 1.16 (Demostración en PA). — Sea ϕ una fórmula cerrada del lenguaje de PA. Se llama una *demostración* de ϕ en PA cualquier derivación d de un secuente con la forma $\Delta \vdash \phi$, donde Δ es una lista finita de axiomas de Peano. La relación « d es una demostración de ϕ en PA » se escribe $d : (\mathrm{PA} \vdash \phi)$. Cuando tal demostración existe, se dice que la fórmula ϕ es *demostrable en* PA —o que ϕ es *un teorema de* PA— y se escribe $\mathrm{PA} \vdash \phi$.

Observación 1.17. — Es importante recordar que en cualquier teoría de primer orden, una demostración es una estructura finita que sólo utiliza un número finito de axiomas, incluso cuando el sistema de axiomas subyacente es infinito. En el caso de PA, esto significa que una demostración sólo utiliza un número finito de axiomas de inducción (A_7). Por lo tanto, cada fórmula ϕ que es demostrable en PA es demostrable en una subteoría finita de PA, es decir: en una subteoría con un sistema finito de axiomas (propiedad de *compacidad sintáctica*).

Notación 1.18 (Pseudo-secuentes). — Durante la construcción de una demostración en PA, es cómodo utilizar pseudo-secuentes de la forma $\mathrm{PA}, \Gamma \vdash \phi$, donde el prefijo « PA, » representa la lista (finita) de todos los axiomas utilizados efectivamente por la demostración —la cual sólo se puede conocer *a fortiori*, una vez que la construcción de la demostración ha sido acabada. En práctica, los pseudo-secuentes $\mathrm{PA}, \Gamma \vdash \phi$ se derivan con las mismas reglas de deducción (Cuadros 1 y 2) como los secuentes usuales, añadiendo el prefijo « PA, » en las premisas/conclusiones, y modificando la regla « Axioma » (Cuadro 1) por:

Axioma $\qquad \overline{\mathrm{PA},\Gamma \vdash \phi} \qquad$ si $\phi \in \Gamma \cup \mathrm{Ax(PA)}$

La posibilidad de utilizar los axiomas de PA como hipótesis suplementarias en los pseudo-secuentes permite derivar reglas específicas para la Aritmética, dadas en el Cuadro 3.

1.5. Consistencia y completitud. El principal interés de un sistema formal tal que PA es que él permite distinguir dos clases de fórmulas cerradas ϕ: las que son *demostrables* (tales que $\mathrm{PA} \vdash \phi$), consideradas como verdaderas, y las que son *refutables* (tales que $\mathrm{PA} \vdash \neg\phi$), consideradas como falsas. Tal distinción naturalmente plantea dos problemas:

El problema de la consistencia: Se dice que PA es *consistente* si PA $\nvdash \perp$, es decir: si no hay ninguna demostración formal de la fórmula absurda $\perp$ a partir de los axiomas de Peano. Es claro que si PA es consistente, entonces la clase de las fórmulas demostrables y la clase de las fórmulas refutables son desyuntadas (por la regla de contradicción, véase Cuadro 2 p. 9). En otro lado, si PA es inconsistente (es decir: si PA $\vdash \perp$), entonces toda fórmula cerrada ϕ es un teorema de PA (por la regla del *ex falso quod libet*, véase Cuadro 2) así como su negación $\neg\phi$, de tal modo que cada una de las dos clases de fórmulas (demostrables y refutables) se confunde con la clase de todas las fórmulas cerradas —dicho de otra manera: tal sistema formal no sirve para nada.

El problema de la completitud: Se dice que una fórmula cerrada ϕ es *decidible* (en PA) cuando es demostrable (PA $\vdash \phi$) o refutable (PA $\vdash \neg\phi$), y que PA es *completa* si todas las fórmulas cerradas de su lenguaje son decidibles. De modo equivalente, PA es completa si la clase de todas la fórmulas cerradas es la unión de ambas clases de fórmulas, demostrables y refutables. Es claro que un sistema formal que es a la vez consistente y completo induce nociones de demostrabilidad y refutabilidad ideales que forman una *partición* de la clase de las fórmulas cerradas, de tal modo que estas nociones se pueden confundir con las nociones de verdad y de falsedad[3]. En otro lado, si PA es incompleta, la existencia de fórmulas cerradas indecidibles (es decir: ni demostrables ni refutables) crea un "tercer caso" que prohíbe identificar las nociones de demostrabilidad y de refutabilidad con las nociones de verdad y de falsedad (por el principio del tercero excluido). En este caso, sólo se puede ver la demostrabilidad y la refutabilidad como *aproximaciones* (por defecto) de las nociones de verdad y de falsedad.

El primer teorema de incompletitud de Gödel (Sección 4) muestra que si PA es consistente, entonces es incompleta. Dicho de otra manera, sólo hay dos casos posibles:

- O bien PA es inconsistente y completa (trivialmente). En este caso, todas las fórmulas cerradas son demostrables *y* refutables, pero el sistema formal no sirve para nada.
- O bien PA es consistente e incompleta. En este caso, las fórmulas demostrables y refutables forman clases desyuntadas, pero existe fórmulas cerradas —las fórmulas indecidibles— que permanecen afuera de esta clasificación.

1.6. Expresividad de PA. A pesar de su lenguaje minimalista, la Aritmética de Peano es un formalismo muy expresivo, que permite expresar y demostrar todas las propiedades básicas sobre los enteros naturales. Por ejemplo las propiedades algebraicas de + y ×:

Proposición 1.19 (Propiedades algebraicas)**.** — *Las siguientes fórmulas son teoremas de* PA*:*

1. $\forall x\,(x + 0 = x)$ *(0 es neutro para +)*
2. $\forall x\,\forall y\,(x + y = y + x)$ *(conmutatividad de +)*
3. $\forall x\,\forall y\,\forall z\,((x + y) + z = x + (y + z))$ *(asociatividad de +)*
4. $\forall x\,\forall y\,\forall z\,(x + z = y + z \Rightarrow x = y)$ *(regularidad de +)*
5. $\forall x\,(x \times 0 = 0)$ *(0 es absorbente para ×)*
6. $\forall x\,(x \times 1 = x)$ *(1 es neutro para ×)*
7. $\forall x\,\forall y\,(x \times y = y \times x)$ *(conmutatividad de ×)*
8. $\forall x\,\forall y\,\forall z\,((x \times y) \times z = x \times (y \times z))$ *(asociatividad de ×)*
9. $\forall x\,\forall y\,\forall z\,(z \neq 0 \land x \times z = y \times z \Rightarrow x = y)$ *(regularidad de ×)*
10. $\forall x\,\forall y\,\forall z\,((x + y) \times z = x \times z + y \times z)$ *(distributividad de × respecto a +)*

[3] Del punto de vista de la teoría de modelos, la identificación de la demostrabilidad con la verdad (y de la refutabilidad con la falsedad) se justifica en razón de que para todo modelo $\mathscr{M}$ (en el sentido de Tarski) de una teoría $\mathscr{T}$ de primer orden que es a la vez consistente y completa, tenemos la equivalencia $\mathscr{T} \vdash \phi$ (« ϕ es demostrable en $\mathscr{T}$ ») sii $\mathscr{M} \models \phi$ (« ϕ es verdadera en $\mathscr{M}$ ») para toda fórmula cerrada ϕ. De hecho, todos los modelos de una teoría consistente y completa son elementalmente equivalentes (aunque puedan ser no isomorfos).

(utilizando la abreviatura $t \neq u$ para $\neg(t = u)$).

Además, el orden (estricto y no estricto) se puede definir en el lenguaje de PA mediante de las abreviaturas:

$$\begin{array}{rcll} t \leq u & \equiv & \exists z\,(t + z = u) & (z \notin FV(t),\ z \notin FV(u)) \\ t < u & \equiv & s(t) \leq u \ \equiv\ \exists z\,(t + z = u) & (z \notin FV(t),\ z \notin FV(u)) \\ t \nleq u & \equiv & \neg(t \leq u) & \\ t \nless u & \equiv & \neg(t < u) & \end{array}$$

De nuevo:

Proposición 1.20 (Propiedades del orden). — *Las siguientes fórmulas son teoremas de* PA*:*

1. $\forall x\,(0 \leq x)$ *(0 es mínimo para $\leq$)*
2. $\forall x\,(x \leq x)$ *(reflexividad de $\leq$)*
3. $\forall x\,\forall y\,(x \leq y \land y \leq x \Rightarrow x = y)$ *(antisimetría de $\leq$)*
4. $\forall x\,\forall y\,\forall z\,(x \leq y \land y \leq z \Rightarrow x \leq z)$ *(transitividad de $\leq$)*
5. $\forall x\,\forall y\,(x \leq y \lor y \leq x)$ *(totalidad de $\leq$)*
6. $\forall x\,\forall y\,(x \leq y \Leftrightarrow s(x) \leq s(y))$
7. $\forall x\,\forall y\,\forall z\,(x \leq y \Leftrightarrow x + z \leq y + z)$
8. $\forall x\,\forall y\,\forall z\,(z \neq 0 \Rightarrow (x \leq y \Leftrightarrow x \times z \leq y \times z))$
9. $\forall x\,\neg(x < x)$ *(irreflexividad de $<$)*
10. $\forall x\,\forall y\,\forall z\,(x < y \land y < z \Rightarrow x < z)$ *(transitividad de $<$)*
11. $\forall x\,\forall y\,(x < y \lor x = y \lor y < x)$ *(principio de tricotomía)*
12. $\forall x\,\forall y\,(x < y \Leftrightarrow s(x) < s(y))$
13. $\forall x\,\forall y\,\forall z\,(x < y \Leftrightarrow x + z < y + z)$
14. $\forall x\,\forall y\,\forall z\,(z \neq 0 \Rightarrow (x < y \Leftrightarrow x \times z < y \times z))$
15. $\forall x\,\forall y\,(x \leq y \Leftrightarrow x < y \lor x = y)$ *(caracterización de $\leq$)*
16. $\forall x\,\forall y\,(x < y \Leftrightarrow x \leq y \land x \neq y)$ *(caracterización de $<$)*
17. $\forall \vec{z}\,[\forall x\,(\forall y\,(y < x \Rightarrow \phi(y, \vec{z})) \Rightarrow \phi(x, \vec{z})) \ \Rightarrow\ \forall x\,\phi(x, \vec{z})]$ *(principio de inducción fuerte)*
para cada fórmula $\phi(x, \vec{z})$ con las variables libres x y $\vec{z} \equiv z_1, \ldots, z_k$ ($k \geq 0$).
18. $\forall \vec{z}\,[\exists x\,\phi(x, \vec{z}) \ \Rightarrow\ \exists x_0\,(\phi(x_0, \vec{z}) \land \forall x\,(\phi(x, \vec{z}) \Rightarrow x_0 \leq x))]$ *(principio de buen orden)*
para cada fórmula $\phi(x, \vec{z})$ con las variables libres x y $\vec{z} \equiv z_1, \ldots, z_k$ ($k \geq 0$).

Más generalmente, se puede formalizar y demostrar en PA las propiedades básicas relacionadas a la noción de divisibilidad y a los números primos, mediante de las abreviaturas:

$$\begin{array}{rcll} t \mid u & \equiv & \exists z\,(t \times z = u) & (z \notin FV(t),\ z \notin FV(u)) \\ \mathrm{Prim}(t) & \equiv & t \neq 1 \land \forall z\,(z \mid t \Rightarrow z = 1 \lor z = t) & (z \notin FV(t)) \end{array}$$

Aunque el lenguaje de PA sólo pueda expresar —estrictamente hablando— propiedades sobre los enteros naturales, se puede formalizar y demostrar en PA algunas propiedades de ciertos conjuntos (finitos e infinitos), como por ejemplo el conjunto de los números primos:

Proposición 1.21 (Teorema de Euclides). — *La fórmula*

$$\forall x\,\exists y\,(y > x\ \land\ \mathrm{Prim}(y))$$

(« el conjunto de los números primos es infinito ») es un teorema de PA.

En la Sección 2.4, veremos que el lenguaje PA es suficientemente expresivo para representar todas las sucesiones finitas de enteros naturales —y más generalmente todas las estructuras finitas —como enteros naturales, mediante de una codificación adecuada. Y que su sistema de axiomas es bastante potente para razonar sobre las estructuras finitas, à través de dicha codificación. Esta observación —según la cual la Aritmética de Peano no es sólo la teoría de los enteros naturales, pero más generalmente la teoría de *todas las estructuras finitas*— es una de las ideas claves de la demostración de los teoremas de incompletitud de Gödel.

1.7. El modelo estándar de PA. Se llama el *modelo estándar de la aritmética* el modelo (en el sentido de Tarski) cuyo dominio de interpretación es el conjunto $\mathbb{N}$ de los enteros naturales, y en el cual cada símbolo del lenguaje de PA es interpretado del modo obvio: el símbolo 0 por el entero 0, el símbolo s por la función sucesor, etc. Por abuso de lenguaje, el modelo estándar (formado a partir del conjunto $\mathbb{N}$) también se escribe $\mathbb{N}$. Formalmente:

Definición 1.22 (Valor de un término cerrado)**.** — A cada término cerrado t del lenguaje de PA, se asocia un entero natural escrito $[\![t]\!]$ y llamado el *valor* de t. Este valor es definido por recursión sobre la estructura de t a partir de las siguientes ecuaciones:

$$\begin{array}{rclcrcl} [\![0]\!] & = & 0 & \qquad & [\![t+u]\!] & = & [\![t]\!] + [\![u]\!] \\ [\![s(t)]\!] & = & [\![t]\!] + 1 & \qquad & [\![t \times u]\!] & = & [\![t]\!] \cdot [\![u]\!] \end{array}$$

(dando a los símbolos 0, 1, +, $\cdot$ su significación usual en $\mathbb{N}$ en los miembros derechos).

En particular, el valor del entero de Peano $\overline{n}$ (Def. 1.2) es $[\![\overline{n}]\!] = n$ para todo $n \in \mathbb{N}$.

Luego se define la *relación de satisfacción* $\mathbb{N} \models \phi$ (donde ϕ es una fórmula cerrada), que expresa que la fórmula ϕ es verdadera en el modelo estándar:

Definición 1.23 (Satisfacción de una fórmula cerrada)**.** — La *relación de satisfacción* $\mathbb{N} \models \phi$ (donde ϕ es una fórmula cerrada) es definida por inducción sobre el número de conectivas y de cuantificadores en ϕ por las siguientes equivalencias:

1. $\mathbb{N} \not\models \bot$ (es decir: $\mathbb{N} \models \bot$ no se cumple, por definición);
2. $\mathbb{N} \models t = u$ sii $[\![t]\!] = [\![u]\!]$;
3. $\mathbb{N} \models \phi \Rightarrow \psi$ sii ($\mathbb{N} \models \phi$ implica $\mathbb{N} \models \psi$);
4. $\mathbb{N} \models \forall x\, \phi$ sii $\mathbb{N} \models \phi\{x := \overline{n}\}$ para todo $n \in \mathbb{N}$.

Las ecuaciones anteriores se extienden inmediatamente a las construcciones definidas $\top$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \Leftrightarrow \psi$ y $\exists x\, \phi$ (Notaciones 1.4 p. 3) por:

5. $\mathbb{N} \models \top$;
6. $\mathbb{N} \models \neg\phi$ sii $\mathbb{N} \not\models \phi$;
7. $\mathbb{N} \models \phi \wedge \psi$ sii $\mathbb{N} \models \phi$ y $\mathbb{N} \models \psi$;
8. $\mathbb{N} \models \phi \vee \psi$ sii $\mathbb{N} \models \phi$ o $\mathbb{N} \models \psi$;
9. $\mathbb{N} \models \phi \Leftrightarrow \psi$ sii ($\mathbb{N} \models \phi$ sii $\mathbb{N} \models \psi$);
10. $\mathbb{N} \models \exists x\, \phi$ sii $\mathbb{N} \models \phi\{x := \overline{n}\}$ para algún $n \in \mathbb{N}$.

Cuando se cumple la aserción $\mathbb{N} \models \phi$, se dice que la fórmula cerrada ϕ es *satisfecha*, o *verdadera*, en el modelo estándar. Es claro que todas las reglas de deducción (Cuadro 1 p. 6) preservan la noción de verdad en el modelo estándar, de tal modo que:

Proposición 1.24 (Adecuación)**.** — *Si un secuente $\phi_1, \ldots, \phi_p \vdash \psi$ (con variables libres $\vec{x} \equiv x_1, \ldots, x_k$) es derivable, entonces para todos $\vec{n} = n_1, \ldots, n_k \in \mathbb{N}$:*

$$\textit{Si}\ \ \mathbb{N} \models \phi_1\{\vec{x} := \vec{n}\}\ \ \textit{y}\ \ \ldots\ \ \textit{y}\ \ \mathbb{N} \models \phi_p\{\vec{x} := \vec{n}\}, \quad \textit{entonces} \quad \mathbb{N} \models \psi\{\vec{x} := \vec{n}\}.$$

Además, se verifica fácilmente que:

Proposición 1.25 (Satisfacción de los axiomas)**.** — *Todo axioma de Peano ϕ (A_1–A_7) es satisfecho en el modelo estándar:* $\mathbb{N} \models \phi$.

Por lo tanto:

Teorema 1.26 (Corrección)**.** — *Si* PA $\vdash \phi$, *entonces* $\mathbb{N} \models \phi$.

Una definición infinitaria Por construcción, la relación de satisfacción $\mathbb{N} \models \phi$ efectúa una traducción literal de la fórmula cerrada ϕ en el lenguaje de la « teoría ambiente »[4], dando a los símbolos que aparecen en ϕ su sentido usual en el conjunto de los enteros naturales —y más generalmente en la matemática. Por ejemplo, la aserción

$$\mathbb{N} \models \forall x\, \exists y_1\, \exists y_2\, (\mathrm{Prim}(y_1) \wedge \mathrm{Prim}(y_2) \wedge y_1 + y_2 = 2 \times x + 4)$$

es por definición equivalente (en la teoría ambiente) a la aserción

Todo entero par superior a 2 *es suma de dos números primos.*

Así, para determinar si la fórmula cerrada

$$\forall x\, \exists y_1\, \exists y_2\, (\mathrm{Prim}(y_1) \wedge \mathrm{Prim}(y_2) \wedge y_1 + y_2 = 2 \times x + 4)$$

es satisfecha en el modelo estándar, habrá que resolver (positivamente o negativamente) la conjetura de Goldbach, ni más ni menos. Este ejemplo ya muestra que la definición de la relación de satisfacción $\mathbb{N} \models \phi$ no se reduce sólo a un cálculo (contrario a todas las nociones que introdujimos anteriormente), pero constituye una definición infinitaria auténtica.

Técnicamente, la definición de la relación de satisfacción $\mathbb{N} \models \phi$ es una definición recursiva que se basa implícitamente sobre un orden bien fundado en el conjunto de las fórmulas cerradas. Según esta definición, la satisfacción de la fórmula $\forall x\, \phi(x)$ se puede definir *solamente después* de haber definido la satisfacción de *todas* las fórmulas $\phi(\overline{n})$, para todos los enteros naturales $n \in \mathbb{N}$. Tal definición es posible en práctica sólo si la teoría ambiente posee mecanismos que permiten construir objetos infinitarios (por ejemplo conjuntos infinitos) y razonar sobre tales objetos. Por ejemplo en la teoría de conjuntos (tomada aquí como teoría ambiente), se puede definir la relación de satisfacción $\mathbb{N} \models \phi$ de la siguiente manera:

1. Para cada fórmula ϕ, se escribe $|\phi|$ el tamaño de la fórmula ϕ, es decir, el número de conectivas y cuantificadores en A (sin contar las igualdades), y para cada entero $k \in \mathbb{N}$, se escribe $\mathscr{L}_k$ el conjunto de las fórmulas cerradas de tamaño k.
2. Para cada entero natural k, se define un conjunto $\mathscr{V}_k \subseteq \mathscr{L}_k$ cuyos elementos son las fórmulas cerradas de tamaño k que son satisfechas en el modelo estándar. El conjunto $\mathscr{V}_k \subseteq \mathscr{L}_k$ se define por recursión sobre k escribiendo:
$$\begin{array}{rcll} \mathscr{V}_0 & = & \{(t = u) \in \mathscr{L}_0 \ : \ [\![t]\!] = [\![u]\!]\} & \\ \mathscr{V}_k & = & \{(\phi \Rightarrow \psi) \in \mathscr{L}_k \ : \ \phi \in \mathscr{L}_{|\phi|} - \mathscr{V}_{|\phi|} \ \text{ o } \ \psi \in \mathscr{V}_{|\psi|}\} \quad \cup & \\ & & \{(\forall x\, \phi) \in \mathscr{L}_k \ : \ \text{para todo } n \in \mathbb{N},\ \ \phi\{x := \overline{n}\} \in \mathscr{V}_{|\phi|}\} & (k \geq 1) \end{array}$$
Se observa que esta definición es bien formada, pues cada conjunto $\mathscr{V}_k$ ($k \geq 0$) es definido solamente a partir de los conjuntos $\mathscr{V}_0, \ldots, \mathscr{V}_{k-1}$.
3. Finalmente, se escribe $\mathbb{N} \models \phi$ cuando $\phi \in \mathscr{V}_{|\phi|}$. (Se invita al lector verificar que esta definición cumple todas las equivalencias de la Def. 1.23.)

Se puede observar que la construcción por inducción de la sucesión de conjuntos *infinitos* $\mathscr{V}_k \subseteq \mathscr{L}_k$ es la sola parte auténticamente infinitaria de la definición anterior. En otro lado, la definición del valor $[\![t]\!]$ asociado a un término cerrado t es una definición finitaria, que sólo utiliza una cantidad finita de información sobre los subtérminos de t.

En cualquier caso, cada vez que la teoría ambiente es suficientemente expresiva para permitir la construcción del modelo estándar (es decir, para permitir la definición de la relación de satisfacción $\mathbb{N} \models \phi$), es fácil verificar que:

[4]Es decir la teoría (en el sentido intuitivo) en la cual se ubica para razonar sobre los términos, las fórmulas, las derivaciones, etc. de la teoría formal de la aritmética (PA), por oposición a dicha teoría *formal*, que es el *objeto* de nuestro razonamiento. La teoría ambiente (se trata obviamente de una noción intuitiva, no definida) también se llama la *meta-teoría*, o la *teoría externa*; ella corresponde al nivel de razonamiento usual en matemática.

Proposición 1.27 (Consistencia de PA). — *La Aritmética de Peano es consistente:* $\quad \mathrm{PA} \nvdash \bot$.

Demostración. Como la fórmula $\bot$ no es satisfecha en el modelo estándar (por definición), no se puede demostrarla en PA, por el Teorema 1.26 (por contraposición). □

Razonamiento en un marco finitario Obviamente, la demostración de consistencia anterior sólo tiene sentido cuando la teoría ambiente permite construir objetos infinitarios (tales que los conjuntos $\mathscr{V}_k$) y razonar sobre dichos objetos. En cambio, cuando se trabaja en una teoría ambiente *finitaria* —es decir en una teoría ambiente el la cual sólo se puede razonar sobre objetos finitos[5]— siempre es posible razonar sobre los enteros, los términos, las fórmulas o las derivaciones, y definir operaciones sintácticas tales que la sustitución en las fórmulas o el cálculo del valor de un término. Sin embargo, ya no se puede definir el modelo estándar de la Aritmética, y la cuestión de la consistencia de PA permanece abierta.

El lector podrá verificar que la mayoría de los razonamientos que efectuaremos en las siguientes páginas son razonamientos puramente finitarios, que se pueden formalizar plenamente (por ejemplo) en la teoría de conjuntos hereditariamente finitos —es decir: en la teoría de conjuntos el la cual el axioma del infinito es remplazado por su negación (« todos los conjuntos son finitos »)[6]. Por regla general, trabajaremos con mayor frecuencia en un marco finitario, sin suponer que la Aritmética de Peano (PA) es consistente —salvo, por supuesto, en los casos donde mencionaremos explícitamente el modelo estándar.

1.8. Propiedades de completitud de PA.

Antes de demostrar que PA es incompleta (primer teorema de incompletitud), es interesante recordar algunos resultados de completitud de PA respecto a ciertas clases de fórmulas cerradas con baja complejidad lógica.

[**NB:** Los resultados técnicos presentados aquí se pueden saltar en primera lectura.]

1.8.1. Decidibilidad de las fórmulas atómicas.

La siguiente proposición muestra que todas las igualdades y desigualdades verdaderas entre enteros de Peano son demostrables en PA:

Proposición 1.28. — *Para todos enteros naturales n, m, tenemos que:*

$$\begin{cases} \mathrm{PA} \vdash \overline{n} = \overline{m} & si\ n = m \\ \mathrm{PA} \vdash \overline{n} \neq \overline{m} & si\ n \neq m \end{cases}$$

Demostración. En el caso donde $n = m$, tenemos

$$\frac{}{\mathrm{PA} \vdash \overline{n} = \overline{m}} \text{ (refl.)}$$

por la regla de reflexividad (Cuadro 3) —los términos $\overline{n}$ y $\overline{m}$ siendo idénticos. En el caso donde $n \neq m$, se distinguen dos subcasos según que $n > m$ o $n < m$. En cada de los dos subcasos, la derivación se construye utilizando $|n - m| - 1$ veces la regla de inyectividad del sucesor

[5]Por supuesto, esto no implica que el « universo del discurso » (formado por todos los objetos sobre los cuales se permite razonar) sea finito, ni que toda forma de razonamiento por inducción sea prohibida. La Aritmética de Peano es el arquetipo de las teorías finitarias (aunque su « universo del discurso » sea infinito), pero existe teorías finitarias muchas más débiles, como por ejemplo la Aritmética de Robinson Q (véase Sección ???).

[6]De hecho, la teoría *formal* de los conjuntos hereditariamente finitos es equiconsistente a PA.

(Cuadro 3) antes de concluir con la regla « $s \neq 0$ » (*idem*):

$$\dfrac{\dfrac{}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^n(0) = s^m(0)}\,(\text{ax.})}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^{n-1}(0) = s^{m-1}(0)}\,(s\text{-iny.})$$
$$\vdots$$
$$\dfrac{\dfrac{\dfrac{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^{n-m+1}(0) = s(0)}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^{n-m}(0) = 0}\,(s\text{-iny.})}{\mathrm{PA}, \overline{n} = \overline{m} \vdash \bot}\,(s{\neq}0)}{\mathrm{PA} \vdash \overline{n} \neq \overline{m}}\,(\Rightarrow\text{-intro})$$

(subcaso donde $n > m$)

$$\dfrac{\dfrac{}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^n(0) = s^m(0)}\,(\text{ax.})}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^{n-1}(0) = s^{m-1}(0)}\,(s\text{-iny.})$$
$$\vdots$$
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{PA}, \overline{n} = \overline{m} \vdash s(0) = s^{m-n+1}(0)}{\mathrm{PA}, \overline{n} = \overline{m} \vdash 0 = s^{m-n}(0)}\,(s\text{-iny.})}{\mathrm{PA}, \overline{n} = \overline{m} \vdash s^{m-n}(0) = 0}\,(=\text{-sim.})}{\mathrm{PA}, \overline{n} = \overline{m} \vdash \bot}\,(s{\neq}0)}{\mathrm{PA} \vdash \overline{n} \neq \overline{m}}\,(\Rightarrow\text{-intro})$$

(subcaso donde $n < m$) □

Además, los axiomas de PA permiten justificar el cálculo des las operaciones $+$ y $\times$:

Proposición 1.29 (Cálculo de $+$, $\times$)**.** — *Para todos enteros naturales n, m, tenemos que:*

$$\mathrm{PA} \vdash \overline{n} + \overline{m} = \overline{n+m} \qquad y \qquad \mathrm{PA} \vdash \overline{n} \times \overline{m} = \overline{nm}\,.$$

Demostración. (Caso de la adición) La derivación de $\mathrm{PA} \vdash \overline{n} + \overline{m}$ se construye por inducción ambiente (o meta-inducción) sobre n:

- Caso de base ($n = 0$). Tenemos que

$$\dfrac{}{\mathrm{PA} \vdash \overline{0} + \overline{m} = \overline{m}}$$

  por la regla « $0 + u = u$ » (Cuadro 3).
- Paso de inducción ($n \to n + 1$). Tenemos que

$$\dfrac{\dfrac{}{\mathrm{PA} \vdash s(\overline{n}) + \overline{m} = s(\overline{n} + \overline{m})} \qquad \dfrac{\begin{array}{c}\text{(hipótesis de inducción)}\\ \vdots \\ \mathrm{PA} \vdash \overline{n} + \overline{m} = \overline{n+m}\end{array} \qquad \dfrac{}{\mathrm{PA} \vdash s(\overline{n} + \overline{m}) = s(\overline{n} + \overline{m})}\,(=\text{-refl.})}{\mathrm{PA} \vdash s(\overline{n} + \overline{m}) = s(\overline{n+m})}\,(=\text{-elim})}{\mathrm{PA} \vdash \underbrace{s(\overline{n}) + \overline{m}}_{\overline{n+1}+\overline{m}} = \underbrace{s(\overline{n+m})}_{\overline{n+m+1}}}\,(=\text{-trans.})$$

(Caso de la multiplicación) Mismo método como para $+$. □

Proposición 1.30 (Valor de un término cerrado)**.** — *Para todo término cerrado t:*

$$\mathrm{PA} \vdash t = \overline{n} \qquad\qquad (\textit{donde } n = [\![t]\!])$$

Demostración. Por inducción estructural sobre t, utilizando la Prop. 1.29. □

Combinando las Prop. 1.28 y 1.30 arriba, se deduce más generalmente que todas las igualdades cerradas son decidibles en PA:

Proposición 1.31 (Decidabilidad de las igualdades)**.** — *Para todos términos cerrados t, u:*

$$\begin{cases} \mathrm{PA} \vdash t = u & si\ [\![t]\!] = [\![u]\!] \\ \mathrm{PA} \vdash t \neq u & si\ [\![t]\!] \neq [\![u]\!] \end{cases}$$

1.8.2. Decidibilidad del orden. La proposición anterior se puede generalizar a las relaciones de orden $t \leq u$ y $t < u$. Para ello, se comienza por establecer la siguiente proposición:

Proposición 1.32. — *Para todos enteros naturales n, m, tenemos que:*

$$\begin{cases} \mathrm{PA} \vdash \overline{n} \leq \overline{m} & si\ n \leq m \\ \mathrm{PA} \vdash \overline{n} \not\leq \overline{m} & si\ n > m \end{cases} \qquad y \qquad \begin{cases} \mathrm{PA} \vdash \overline{n} < \overline{m} & si\ n < m \\ \mathrm{PA} \vdash \overline{n} \not< \overline{m} & si\ n \geq m \end{cases}$$

Demostración. Mismo método como para la Prop. 1.28, construyendo las derivaciones a partir de derivaciones de las siguientes fórmulas

$$\begin{array}{lll} \forall x\,(0 \le x) & \forall x\,(s(x) \not\le 0) & \forall x\,\forall y\,(x \le y \Leftrightarrow s(x) \le s(y)) \\ \forall x\,(0 < s(x)) & \forall x\,\neg(x \not< 0) & \forall x\,\forall y\,(x < y \Leftrightarrow s(x) < s(y)) \end{array}$$

que son teoremas de PA. □

Combinando la proposición anterior con la Prop. 1.30, se deduce que todas las desigualdades cerradas son decidibles en PA:

Proposición 1.33 (Decidabilidad de las desigualdades). — *Para todos términos cerrados t, u:*

$$\begin{cases} \mathrm{PA} \vdash t \le u & si\ [\![t]\!] \le [\![u]\!] \\ \mathrm{PA} \vdash t \not\le u & si\ [\![t]\!] > [\![u]\!] \end{cases} \qquad y \qquad \begin{cases} \mathrm{PA} \vdash t < u & si\ [\![t]\!] < [\![u]\!] \\ \mathrm{PA} \vdash t \not< u & si\ [\![t]\!] \ge [\![u]\!] \end{cases}$$

1.8.3. Decidibilidad de las fórmulas sin cuantificadores.

Definición 1.34 (Fórmulas sin cuantificadores). — Se llama una *fórmula sin cuantificadores* toda fórmula que se puede construir por aplicación finita de las siguientes reglas:

(1) La fórmula $\perp$ es una fórmula sin cuantificadores.
(2) Si t, u son términos, entonces las tres fórmulas $t = u$, $t \le u$ y $t < u$ son fórmulas sin cuantificadores.
(3) Si ϕ, ψ son fórmulas sin cuantificadores, entonces la fórmula $\phi \Rightarrow \psi$ es una fórmula sin cuantificadores.

Observaciones 1.35. — (1) Aquí, la terminología "sin cuantificadores" es abusiva, pues las dos fórmulas "sin cuantificadores" $t \le u \equiv \exists z\,(t + z = u)$ y $t < u \equiv \exists z\,(s(t) + z = u)$ (construidas por la regla (2)) ya contienen cuantificadores. Intuitivamente, las fórmulas $t \le u$ y $t < u$ se pueden considerar como fórmulas atómicas (similares a la fórmula atómica $t = u$), olvidando las cuantificaciones que sirven a definirlas.

(2) La regla (3) se generaliza obviamente a las conectivas definidas (Def. 1.4 p. 3), utilizando las reglas (1) y (3): si ϕ, ψ son fórmulas sin cuantificadores, entonces las fórmulas $\top$, $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$ y $\phi \Leftrightarrow \psi$ también son fórmulas sin cuantificadores.

Proposición 1.36 (Completitud respecto a las fórmulas sin cuantificadores). — *Toda fórmula cerrada ϕ sin cuantificadores es decidible en* PA*:* $\mathrm{PA} \vdash \phi$ *o* $\mathrm{PA} \vdash \neg\phi$.

Demostración. — La propiedad se demuestra por inducción estructural sobre ϕ, distinguiendo los casos según la última regla aplicada.

(1) ϕ es $\perp$. Obvio, pues $\mathrm{PA} \vdash \neg\perp$.
(2) ϕ es una de las tres fórmulas cerradas $t = u$, $t \le u$ o $t < u$. Sigue de las Prop. 1.31, 1.33.
(3) ϕ es $\phi_1 \Rightarrow \phi_2$, donde ϕ_1 y ϕ_2 son fórmulas cerradas sin cuantificadores. Por hipótesis de inducción, las dos fórmulas ϕ_1, ϕ_2 son decidibles. Se distinguen los siguientes casos:
   1. $\mathrm{PA} \vdash \neg\phi_1$ ($\equiv \phi \Rightarrow \perp$). En este caso, tenemos $\mathrm{PA} \vdash \phi$ por

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\mathrm{PA} \vdash \neg\phi_1}}{\mathrm{PA}, \phi_1 \vdash \neg\phi_1}\,\text{(deb.)} \qquad \dfrac{}{\mathrm{PA}, \phi_1 \vdash \phi_1}\,\text{(ax.)}}{\dfrac{\mathrm{PA}, \phi_1 \vdash \perp}{\mathrm{PA}, \phi_1 \vdash \phi_2}\,(\perp\text{-elim})}\,(\Rightarrow\text{-elim})}{\mathrm{PA} \vdash \underbrace{\phi_1 \Rightarrow \phi_2}_{\phi}}\,(\Rightarrow\text{-intro})$$

   2. $\mathrm{PA} \vdash \phi_1$. En este caso se distinguen los dos siguientes subcasos.

2.1. $\mathrm{PA} \vdash \phi_2$. En este caso, tenemos $\mathrm{PA} \vdash \phi$ por

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \mathrm{PA} \vdash \phi_2\end{array}}{\mathrm{PA}, \phi_1 \vdash \phi_2}\ \text{(deb.)}}{\mathrm{PA} \vdash \underbrace{\phi_1 \Rightarrow \phi_2}_{\phi}}\ (\Rightarrow\text{-intro})$$

2.2. $\mathrm{PA} \vdash \neg\phi_2$. En este caso, tenemos $\mathrm{PA} \vdash \neg\phi$ por

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \mathrm{PA} \vdash \neg\phi_2\end{array}}{\mathrm{PA}, \phi_1 \Rightarrow \phi_2 \vdash \neg\phi_2}\ \text{(deb.)} \qquad \dfrac{\dfrac{}{\mathrm{PA}, \phi_1 \Rightarrow \phi_2 \vdash \phi_1 \Rightarrow \phi_2}\ \text{(ax.)} \quad \dfrac{\begin{array}{c}\vdots\\ \mathrm{PA} \vdash \phi_1\end{array}}{\mathrm{PA}, \phi_1 \Rightarrow \phi_2 \vdash \phi_1}\ \text{(deb.)}}{\mathrm{PA}, \phi_1 \Rightarrow \phi_2 \vdash \phi_2}\ (\Rightarrow\text{-elim})}{\mathrm{PA}, \phi_1 \Rightarrow \phi_2 \vdash \bot}\ (\Rightarrow\text{-elim})}{\mathrm{PA} \vdash \underbrace{\neg(\phi_1 \Rightarrow \phi_2)}_{\neg\phi}}\ (\Rightarrow\text{-intro})$$

□

Observación 1.37. — El lector habrá observado que la demostración de la proposición anterior sigue exactamente el cálculo de la tabla de verdad de la fórmula cerrada ϕ (particularmente en el caso (3) correspondiente a la implicación). De hecho, la propiedad anterior se puede ver como una propiedad de completitud respecto a las tablas de verdad: toda fórmula cerrada que se puede *verificar* con una tabla de verdad finita se puede *demostrar* en PA[7]. En la sección 1.8.5, veremos que esta propiedad se extiende a las fórmulas con cuantificaciones acotadas.

1.8.4. Desarrollo del orden y de las cuantificaciones acotadas.

Notaciones 1.38. — Dado una sucesión finita de fórmulas $\phi_1, \ldots, \phi_n$ ($n \geq 0$), se escribe

$$\bigwedge_{i=1..n} \phi_i \ \equiv\ \phi_1 \wedge \cdots \wedge \phi_n \qquad \text{y} \qquad \bigvee_{i=1..n} \phi_i \ \equiv\ \phi_1 \vee \cdots \vee \phi_n$$

con la convención que $\bigwedge_{i=1..n} \phi_i \equiv \top$ y $\bigvee_{i=1..n} \phi_i \equiv \bot$ cuando $n = 0$ (sucesión vacía).

La siguiente proposición muestra que para todo término cerrado t, las dos fórmulas abiertas $x \leq t$ y $x < t$ se pueden desarrollar en disyunciones finitas de igualdades:

Proposición 1.39 (Desarrollo del orden)**.** — *Para todo término cerrado t:*

$$\mathrm{PA} \ \vdash\ \forall x \Big(x \leq t \ \Leftrightarrow \bigvee_{n \leq [\![t]\!]} x = \overline{n} \Big) \qquad y \qquad \mathrm{PA} \ \vdash\ \forall x \Big(x < t \ \Leftrightarrow \bigvee_{n < [\![t]\!]} x = \overline{n} \Big)$$

Demostración. Por inducción (ambiente) sobre $[\![t]\!]$, utilizando la Prop. 1.30 y construyendo las derivaciones a partir de derivaciones de las fórmulas

$$\begin{array}{ll} \forall x\,(x \leq 0 \Leftrightarrow x = 0) & \qquad \forall x\, \forall y\,(x \leq s(y) \Leftrightarrow x \leq y \vee x = s(y)) \\ \forall x\,(x < 0 \Leftrightarrow \bot) & \qquad \forall x\, \forall y\,(x < s(y) \Leftrightarrow x < y \vee x = y) \end{array}$$

que son teoremas de PA. □

[7]En otro lado, las fórmulas con cuantificadores no se pueden verificar (en general) con tablas de verdad finitas (véase la discusión de la Sección 1.7). Es precisamente la razón de ser de la noción de demostración: capturar un cálculo de verdad infinito en una estructura de datos finita —el precio a pagar siendo la incompletitud.

Notaciones 1.40 (Cuantificaciones acotadas)**.** — Dado una fórmula ϕ, una variable x y un término t tal que $x \notin FV(t)$, se definen las *cuantificaciones acotadas* $(\forall x \leq t)\phi$, $(\forall x < t)\phi$, $(\exists x \leq t)\phi$ y $(\exists x < t)\phi$ mediante de las siguientes abreviaturas

$$\begin{array}{rclcrcl}(\forall x \leq t)\,\phi & \equiv & \forall x\,(x \leq t \Rightarrow \phi) & \qquad & (\forall x < t)\,\phi & \equiv & \forall x\,(x < t \Rightarrow \phi)\\ (\exists x \leq t)\,\phi & \equiv & \neg(\forall x \leq t)\,\neg\phi & & (\exists x < t)\,\phi & \equiv & \neg(\forall x < t)\,\neg\phi\end{array}$$

Sigue inmediatamente de la Prop. 1.39 que:

Proposición 1.41 (Desarrollo de las cuantificaciones acotadas)**.** — *Para toda fórmula $\phi \equiv \phi(x)$ (dependiendo de x) y para todo término cerrado t, tenemos que:*

$$\begin{array}{lcl} \mathrm{PA} \vdash (\forall x \leq t)\phi(x) \Leftrightarrow \bigwedge_{n \leq [\![t]\!]} \phi(\overline{n}) & \qquad & \mathrm{PA} \vdash (\forall x < t)\phi(x) \Leftrightarrow \bigwedge_{n < [\![t]\!]} \phi(\overline{n}) \\ \mathrm{PA} \vdash (\exists x \leq t)\phi(x) \Leftrightarrow \bigvee_{n \leq [\![t]\!]} \phi(\overline{n}) & & \mathrm{PA} \vdash (\exists x < t)\phi(x) \Leftrightarrow \bigvee_{n < [\![t]\!]} \phi(\overline{n}) \end{array}$$

1.8.5. Decidibilidad de las fórmulas con cuantificaciones acotadas.

Definición 1.42 (Fórmulas con cuantificaciones acotadas, o FCA)**.** — Se llama una *fórmula con cuantificaciones acotadas* (FCA) toda fórmula que se puede construir por aplicación finita de las siguientes reglas:

(1) La fórmula $\bot$ es una FCA.
(2) Si t, u son términos, entonces las tres fórmulas $t = u$, $t \leq u$ y $t < u$ son FCA.
(3) Si ϕ, ψ son FCA, entonces la fórmula $\phi \Rightarrow \psi$ es una FCA.
(4) Si ϕ es una FCA y t un término tal que $x \notin FV(t)$, entonces las dos fórmulas $(\forall x \leq t)\,\phi$ y $(\forall x < t)\,\phi$ son FCA.

Proposición 1.43 (Eliminación de las cuantificaciones acotadas)**.** — *Toda fórmula cerrada ϕ con cuantificaciones acotadas es equivalente en* PA *a alguna fórmula cerrada ϕ^* sin cuantificadores:* $\mathrm{PA} \vdash \phi \Leftrightarrow \phi^*$.

Demostración. La fórmula cerrada ϕ^* (sin cuantificadores) asociada a la fórmula cerrada ϕ (con cuantificaciones acotadas) es definida por inducción sobre el número de conectivas y de cuantificaciones acotadas en ϕ utilizando las ecuaciones

$$\begin{array}{rclcrcl} \bot^* & \equiv & \bot & & & & \\ (t = u)^* & \equiv & t = u & \qquad & (\phi \Rightarrow \psi)^* & \equiv & \phi^* \Rightarrow \psi^* \\ (t \leq u)^* & \equiv & t \leq u & & (t < u)^* & \equiv & t < u \\ ((\forall x \leq t)\,\phi)^* & \equiv & \bigwedge_{n \leq [\![t]\!]} (\phi\{x := \overline{n}\})^* & & ((\forall x < t)\,\phi)^* & \equiv & \bigwedge_{n < [\![t]\!]} (\phi\{x := \overline{n}\})^* \end{array}$$

y observando que para toda cuantificación acotada cerrada $(\forall x \leq t)\,\phi$ o $(\forall x < t)\,\phi$, el término t es cerrado, de tal modo que se pueda desarrollar dicha cuantificación (Prop. 1.41). Luego se verifica fácilmente por inducción que: $\mathrm{PA} \vdash \phi \Leftrightarrow \phi^*$. □

Combinando la proposición anterior con la Prop. 1.36, se deduce inmediatamente que todas las fórmulas cerradas con cuantificaciones acotadas son decidibles en PA:

Corolario 1.44 (Decidibilidad de las fórmulas con cuantificaciones acotadas)**.** — *Toda fórmula cerrada ϕ con cuantificaciones acotadas es decidible en* PA*:* $\mathrm{PA} \vdash \phi$ *o* $\mathrm{PA} \vdash \neg\phi$.

2. Funciones recursivas

Esta sección es dedicada a las funciones recursivas, que constituyen uno de los ingredientes principales en la demostración de los teoremas de incompletitud de Gödel. Aquí se recuerdan las definiciones y propiedades principales de la teoría de la recursión (Secciones 2.1 y 2.2), antes de establecer el *teorema de representación* en PA (Secciones 2.3 y 2.4).

2.1. Definiciones y propiedades. En lo siguiente, se escribe $\mathbb{N}^k \rightharpoonup \mathbb{N}$ el conjunto de las *funciones parciales* de $\mathbb{N}^k$ a $\mathbb{N}$ ($k \geq 1$), y dado una función parcial $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$, se escribe $\mathrm{dom}(f)$ ($\subseteq \mathbb{N}^k$) su *dominio*. Se recuerda que el conjunto $\mathbb{N}^k \to \mathbb{N}$ de las funciones *totales* de $\mathbb{N}^k$ a $\mathbb{N}$ es un subconjunto del conjunto de las funciones parciales: $(\mathbb{N}^k \to \mathbb{N}) \subset (\mathbb{N}^k \rightharpoonup \mathbb{N})$.

Definición 2.1 (Funciones recursivas). — Se llama una *función recursiva* toda función parcial $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ ($k \geq 1$) que se puede construir por aplicación finita de las siguientes reglas:

Función nula: La función $\mathtt{cero} : \mathbb{N} \to \mathbb{N}$ definida por

$$\mathtt{cero}(n) = 0$$

para todo $n \in \mathbb{N}$ (función total) es recursiva.

Función sucesor: La función $\mathtt{suc} : \mathbb{N} \to \mathbb{N}$ definida por

$$\mathtt{suc}(n) = n + 1$$

para todo $n \in \mathbb{N}$ (función total) es recursiva.

Proyecciones: Para todos $k \geq i \geq 1$, la función $\mathtt{proy}_i^k : \mathbb{N}^k \to \mathbb{N}$ definida por

$$\mathtt{proy}_i^k(n_1, \ldots, n_k) = n_i$$

para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ (función total) es recursiva.

Esquema de composición: Si las funciones $f_1, \ldots, f_p : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^p \rightharpoonup \mathbb{N}$ ($k, p \geq 1$) son recursivas, entonces la función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ definida por

$$h(n_1, \ldots, n_k) = g(f_1(n_1, \ldots, n_k), \ldots, f_p(n_1, \ldots, n_k))$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho es definido) es recursiva.

Nota Bene: dado $(n_1, \ldots, n_k) \in \mathbb{N}^k$, el valor de $h(n_1, \ldots, n_k)$ no es definido cuando:
- o bien $f_i(n_1, \ldots, n_k)$ no es definido para algún $i \in [1..p]$;
- o bien $f_i(n_1, \ldots, n_k)$ es definido para todo $i \in [1..p]$, pero $g(f_1(n_1, \ldots, n_k), \ldots, f_p(n_1, \ldots, n_k))$ no es definido.

Esquema de recursión: Si las funciones $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$ (con $k \geq 1$) son recursivas, entonces la función $h : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ definida por recursión sobre $n \in \mathbb{N}$ por

$$\begin{aligned} h(0, n_1, \ldots, n_k) &= f(n_1, \ldots, n_k) \\ h(n+1, n_1, \ldots, n_k) &= g(n, h(n, n_1, \ldots, n_k), n_1, \ldots, n_k) \end{aligned}$$

(para todo $(n, n_1, \ldots, n_k) \in \mathbb{N}^{k+1}$ donde el lado derecho es definido) es recursiva.

Nota Bene: dado $(n, n_1, \ldots, n_k) \in \mathbb{N}^{k+1}$, el valor de $h(n, n_1, \ldots, n_k)$ no es definido cuando:
- o bien $n = 0$ y $f(n_1, \ldots, n_k)$ no es definido;
- o bien $n \geq 1$ y $h(n-1, n_1, \ldots, n_k)$ no es definido;
- o bien $n \geq 1$ y $h(n-1, n_1, \ldots, n_k)$ es definido, pero $g(n-1, h(n-1, n_1, \ldots, n_k), n_1, \ldots, n_k)$ no es definido.

Esquema de minimización: Si la función $f : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ ($k \geq 1$) es recursiva, entonces la función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ definida por:

$$\begin{aligned} h(n_1, \ldots, n_k) &= \mu n\,.\, f(n, n_1, \ldots, n_k) > 0 \\ &= \text{el único } n \in \mathbb{N} \text{ (si existe) tal que} \\ &\quad\ f(n, n_1, \ldots, n_k) > 0 \ \wedge\ (\forall m < n)\, f(m, n_1, \ldots, n_k) = 0 \end{aligned}$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho es definido) es recursiva.

Nota Bene: dado $(n_1, \ldots, n_k) \in \mathbb{N}^k$, el valor de $h(n_1, \ldots, n_k)$ no es definido cuando:
- o bien existe $n \in \mathbb{N}$ tal que $f(n, n_1, \ldots, n_k)$ no es definido, mientras $f(m, n_1, \ldots, n_k) = 0$ (definido) para todo $m < n$;
- o bien $f(n, n_1, \ldots, n_k) = 0$ (definido) para todo $n \in \mathbb{N}$.

Observaciones 2.2. — (1) Es importante observar que:

- Todas las funciones *iniciales* —es decir: las funciones `cero`, `suc` : $\mathbb{N} \to \mathbb{N}$ y las proyecciones $\mathtt{proy}_i^k : \mathbb{N}^k \to \mathbb{N}$ ($k \geq i \geq 1$)— son funciones recursivas totales.
- Si $f_1, \ldots, f_p : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^p \rightharpoonup \mathbb{N}$ ($k, p \geq 1$) son funciones recursivas totales, entonces la función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ construida a partir de $f_1, \ldots, f_p, g$ por el ***esquema de composición*** es una función recursiva total también.
- Si $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ ($k \geq 1$) son funciones recursivas totales, entonces la función $h : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ construida a partir de f y g por el ***esquema de recursión*** es una función recursiva total también.

Luego, la única regla que permite construir una función recursiva $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ que sea realmente parcial es el ***esquema de minimización***. Esto pasa típicamente cuando la función recursiva $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ es definida por minimización

$$h(n_1, \ldots, n_k) = \mu n \,.\, f(n, n_1, \ldots, n_k) > 0$$

a partir de una función total $f : \mathbb{N}^{k+1} \to \mathbb{N}$ para la cual existe $(n_1, \ldots, n_k) \in \mathbb{N}^k$ tal que $f(n, n_1, \ldots, n_k) = 0$ para todo $n \in \mathbb{N}$, de tal modo que el valor de $h(n_1, \ldots, n_k)$ es indefinido. Intuitivamente, la función h intenta hallar el más pequeño $n \in \mathbb{N}$ tal que $f(n, n_1, \ldots, n_k) > 0$, calculando sucesivamente los valores de $f(m, n_1, \ldots, n_k)$ para todos los enteros $m \in \mathbb{N}$, hasta encontrar un $m \in \mathbb{N}$ tal que $f(m, n_1, \ldots, n_k) > 0$ —pero como tal m no existe, el cálculo de $h(n_1, \ldots, n_k)$ entra en un bucle infinito.

(2) La discusión anterior muestra más generalmente que si $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ es una función recursiva parcial, los argumentos $(n_1, \ldots, n_k) \in \mathbb{N}^k$ para los cuales el valor de $f(n_1, \ldots, n_k)$ es indefinido son precisamente los para los cuales el cálculo efectivo de $f(n_1, \ldots, n_k)$ entra en un bucle infinito. Por lo tanto, escribiremos en lo siguiente:

- $f(n_1, \ldots, f_k)\downarrow$ cuando $(n_1, \ldots, n_k) \in \mathrm{dom}(f)$ (« $f(n_1, \ldots, n_k)$ converge »)
- $f(n_1, \ldots, f_k)\uparrow$ cuando $(n_1, \ldots, n_k) \notin \mathrm{dom}(f)$ (« $f(n_1, \ldots, n_k)$ diverge »)

(3) La noción de función recursiva intenta captura la noción intuitiva de « función calculable ». De hecho, se puede demostrar que la clase de las *funciones recursivas* (Gödel, 1931) coincide con la clase de las *funciones λ-definibles* (Church, 1936), la clase de las *funciones Turing-computables* (Turing, 1936) y más generalmente, con la clase de todas las funciones que se pueden implementar en cualquier lenguaje de programación razonable. En lo siguiente, describiremos a menudo las funciones recursivas con *pseudo-código*, dado que tal pseudo-código se puede siempre compilar en un función recursiva según la definición oficial.

(4) En términos de cardinalidad, se puede demostrar que el conjunto de todas las funciones recursivas parciales y totales de aridad $k \geq 1$ es infinito numerable ($= \aleph_0 = \mathrm{card}(\mathbb{N})$), mientras el conjunto $\mathbb{N}^k \rightharpoonup \mathbb{N}$ de todas las funciones parciales (así como el conjunto $\mathbb{N}^k \to \mathbb{N}$ de todas las funciones totales) tiene la *potencia del continuo* ($= 2^{\aleph_0} = \mathrm{card}(\mathbb{R})$). Este argumento ya muestra que la gran mayoría de las funciones $f : \mathbb{N}^k \to \mathbb{N}$ no son recursivas.

Definición 2.3 (Función recursiva primitiva)**.** — Se llama una *función recursiva primitiva* toda función recursiva que se puede construir a partir de las funciones iniciales (cero, sucesor y proyecciones) sólo utilizando el esquema de composición y el esquema de recursión.

Según la Observación 2.2 (1), es claro que:

Proposición 2.4. — *Toda función recursiva primitiva es una función total.*

Observaciones 2.5. — (1) Todas las funciones aritméticas usuales son recursivas primitivas: $+$ (adición), $\times$ (multiplicación), $\dot{-}$ (sustracción truncada[8]), $\div$ (cociente de la división euclidiana), mód (resto de la división euclidiana), ! (factorial), $\uparrow$ (potencia), etc.

[8]La substracción truncada es definida para todos $n, m \in \mathbb{N}$ por $n \dot{-} m = \mathrm{máx}(n - m, 0)$ ($\in \mathbb{N}$).

(2) Todas las funciones recursivas primitivas son totales, pero existen funciones recursivas totales que no son recursivas primitivas. Un contra-ejemplo es dado por la función recursiva total $\mathtt{ack} : \mathbb{N}^2 \to \mathbb{N}$ (*función de Ackermann*) definida por las ecuaciones:

$$\begin{array}{rcl} \mathtt{ack}(0,m) & = & n+1 \\ \mathtt{ack}(n+1,0) & = & \mathtt{ack}(n,1) \\ \mathtt{ack}(n+1,m+1) & = & \mathtt{ack}(n,\mathtt{ack}(n+1,m)) \end{array} \qquad (n,m \in \mathbb{N})$$

En efecto, se puede demostrar que la función « diagonal » $(n \mapsto \mathtt{ack}(n,n)) : \mathbb{N} \to \mathbb{N}$ crece más rápidamente que *cualquier* función recursiva primitiva $f : \mathbb{N} \to \mathbb{N}$, en el sentido en que

$$\text{existe } n_f \in \mathbb{N} \text{ tal que para todo } n \geq n_f\text{:} \quad \mathtt{ack}(n,n) > f(n).$$

Obviamente, esta propiedad implica que la función de Ackermann no es recursiva primitiva.

(3) Por definición, las funciones recursivas primitivas no se pueden construir por el ***esquema de minimización***. Sin embargo, la clase de las funciones recursivas primitivas es estable por una restricción de este esquema: el ***esquema de minimización acotada***.

Proposición 2.6 (Esquema de minimización acotada)**.** — *Si $f : \mathbb{N}^{k+1} \to \mathbb{N}$ y $g : \mathbb{N}^k \to \mathbb{N}$ ($k \geq 1$) son funciones recursivas primitivas, entonces la función $h : \mathbb{N}^k \to \mathbb{N}$ definida por*

$$\begin{aligned} h(n_1,\ldots,n_k) &= \mu\, n < g(n_1,\ldots,n_k)\,.\, f(n,n_1,\ldots,n_k) > 0 \\ &= \min(\{n < g(n_1,\ldots,n_k) : f(n,n_1,\ldots,n_k) > 0\} \cup \{g(n_1,\ldots,n_k)\}) \end{aligned}$$

para todo $(n_1,\ldots,n_k) \in \mathbb{N}^k$ es recursiva primitiva.

2.2. Conjuntos recursivos.

Definición 2.7 (Conjuntos recursivos, recursivos primitivos)**.** — Dado un entero $k \geq 1$, se dice que un conjunto $A \subseteq \mathbb{N}^k$ es *recursivo* (resp. *recursivo primitivo*) cuando su función característica $\mathbb{1}_A : \mathbb{N}^k \to \mathbb{N}$, definida por

$$\mathbb{1}_A(n_1,\ldots,n_k) = \begin{cases} 1 & \text{si } (n_1,\ldots,n_k) \in A \\ 0 & \text{si } (n_1,\ldots,n_k) \notin A, \end{cases} \qquad ((n_1,\ldots,n_k) \in \mathbb{N}^k)$$

es una función recursiva (resp. recursiva primitiva).

Proposición 2.8. — *Dado $k \geq 1$:*

(1) *Todo conjunto recursivo primitivo $A \subseteq \mathbb{N}^k$ es recursivo.*
(2) *Todo conjunto finito $A \subseteq \mathbb{N}^k$ es recursivo primitivo (y luego: recursivo).*
(3) *Si dos conjuntos $A, B \subseteq \mathbb{N}^k$ son recursivos (resp. recursivos primitivos), entonces los conjuntos A^c ($= \mathbb{N}^k - A$), $A \cup B$ y $A \cap B$ son recursivos (resp. recursivos primitivos).*
(4) *Una función total $f : \mathbb{N}^k \to \mathbb{N}$ es recursiva (resp. recursiva primitiva) si y sólo si su grafo*

$$\{(n_1,\ldots,n_k,m) : f(n_1,\ldots,n_k) = m\} \qquad (\subseteq \mathbb{N}^{k+1})$$

es un conjunto recursivo (resp. recursivo primitivo).

Definición 2.9 (Conjuntos recursivamente enumerables)**.** — Dado $k \geq 1$, se dice que un conjunto $A \subseteq \mathbb{N}^k$ es *recursivamente enumerable* cuando $A = \mathrm{dom}(f)$ para alguna función recursiva (parcial) $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$. De modo equivalente, un conjunto $A \subseteq \mathbb{N}^k$ es recursivamente enumerable cuando su *función característica parcial* $\mathbb{1}_A^{\uparrow} : \mathbb{N}^k \rightharpoonup \mathbb{N}$, definida por

$$\mathbb{1}_A^{\uparrow}(n_1,\ldots,n_k) = \begin{cases} 1 & \text{si } (n_1,\ldots,n_k) \in A \\ \uparrow \text{ (indefinido)} & \text{si } (n_1,\ldots,n_k) \notin A, \end{cases} \qquad ((n_1,\ldots,n_k) \in \mathbb{N}^k)$$

es una función recursiva.

Proposición 2.10. — *Dado $k \geq 1$:*

(1) *Todo conjunto recursivo* $A \subseteq \mathbb{N}^k$ *es recursivamente enumerable.*
(2) *Si dos conjuntos* $A, B \subseteq \mathbb{N}^k$ *son recursivamente enumerables, entonces los conjuntos* $A \cup B$ *y* $A \cap B$ *son recursivamente enumerables.*
(3) *En general, el complementario* $A^c = \mathbb{N}^k - A$ *de un conjunto recursivamente enumerable* $A \subseteq \mathbb{N}^k$ *no es recursivamente enumerable. Sin embargo:*
(4) *Un conjunto* $A \subseteq \mathbb{N}^k$ *es recursivo si y sólo si* A *y su complementario* $A^c = \mathbb{N}^k - A$ *son recursivamente enumerables.*

2.3. El teorema de representación. El objetivo de esta sección es demostrar que toda función recursiva $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ ($k \geq 1$) se puede representar en el lenguaje de la Aritmética de Peano (PA) por cierta fórmula ϕ_f que describe el cálculo de la función f. Es claro que tal posibilidad de representación viene del carácter *numerable* de las funciones recursivas (no hubiera suficientes fórmulas en el lenguaje de PA para representar *todas* las funciones $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$) y de su carácter *calculable* (que permite tal descripción).

Definición 2.11 (Representación de una función parcial)**.** — Sea $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ una función parcial de aridad $k \geq 1$. Se dice que una fórmula aritmética $\phi_f(x_1, \ldots, x_k, y)$ (con las variables libres $x_1, \ldots, x_k, y$) *representa la función* f *en* PA si para todo $(n_1, \ldots, n_k) \in \mathrm{dom}(f)$:

$$\mathrm{PA} \ \vdash \ \forall y \Big(\phi_f(\overline{n}_1, \ldots, \overline{n}_k, y) \ \Leftrightarrow \ y = \overline{f(n_1, \ldots, n_k)} \Big) .$$

Cuando tal fórmula existe, se dice que la función (parcial) f es *representable en* PA.

Se puede observar que esta noción de representación no especifica nada sobre la relación $\phi_f(\bar{n}_1, \ldots, \bar{n}_k, y)$ para las k-uplas $(n_1, \ldots, n_k)$ afuera del dominio de f. Del mismo modo se define la noción de representación para un subconjunto de $\mathbb{N}^k$:

Definición 2.12 (Representación de un conjunto)**.** — Sea $A \subseteq \mathbb{N}^k$ un conjunto ($k \geq 1$). Se dice que una fórmula aritmética $\phi_A(x_1, \ldots, x_k)$ (con las variables libres $x_1, \ldots, x_k$) *representa el conjunto* $A \subseteq \mathbb{N}^k$ *en* PA si para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$:

$$\begin{cases} \mathrm{PA} \ \vdash \ \phi_A(\overline{n}_1, \ldots, \overline{n}_k) & \text{si} \quad (n_1, \ldots, n_k) \in A \\ \mathrm{PA} \ \vdash \ \neg\phi_A(\overline{n}_1, \ldots, \overline{n}_k) & \text{si} \quad (n_1, \ldots, n_k) \notin A \end{cases}$$

Cuando tal fórmula existe, se dice que el conjunto $A \subseteq \mathbb{N}^k$ es *representable en* PA.

Lema 2.13 (Caracterización)**.** — *Un conjunto* $A \subseteq \mathbb{N}^k$ *es representable en* PA *si y sólo si su función característica* $\mathbb{1}_R : \mathbb{N}^k \to \mathbb{N}$ *es representable en* PA.

Demostración. Si el conjunto $A \subseteq \mathbb{N}^k$ es representado en PA por una fórmula $\phi_A(x_1, \ldots, x_k)$, entonces su función característica $\mathbb{1}_A : \mathbb{N}^k \to \mathbb{N}$ es representada en PA por la fórmula

$$\phi_{\mathbb{1}_A}(x_1, \ldots, x_k, y) \ \equiv \ (\phi_A(x_1, \ldots, x_k) \land y = 1) \lor (\neg\phi_A(x_1, \ldots, x_k) \land y = 0) .$$

Recíprocamente, si la función característica $\mathbb{1}_A : \mathbb{N}^k \to \mathbb{N}$ es representada en PA por una fórmula $\phi_{\mathbb{1}_A}(x_1, \ldots, x_n, y)$, entonces el conjunto $A \subseteq \mathbb{N}^k$ es representado en PA por la fórmula

$$\phi_A(x_1, \ldots, x_k) \ \equiv \ \phi_{\mathbb{1}_A}(x_1, \ldots, x_k, 1) . \qquad \square$$

Teorema 2.14 (Representación)**.** — *Toda función recursiva es representable en* PA.

Corolario 2.15. — *Todo conjunto recursivo es representable en* PA.

Demostración del Teorema 2.14. La demostración se hace por inducción sobre la construcción de la función recursiva f, distinguiendo los casos según la regla de construcción considerada. (En cada caso, se da solamente la fórmula aritmética que representa la función considerada; la justificación correspondiente es dejada en ejercicio al lector.)

Función nula: La función $\mathtt{cero} : \mathbb{N} \to \mathbb{N}$ definida por

$$\mathtt{cero}(n) = 0 \qquad (n \in \mathbb{N})$$

es representada por la fórmula aritmética $\phi(x, y) \equiv y = 0$.

Función sucesor: La función $\mathtt{suc} : \mathbb{N} \to \mathbb{N}$ definida por

$$\mathtt{suc}(n) = n + 1 \qquad (n \in \mathbb{N})$$

es representada por la fórmula aritmética $\phi(x, y) \equiv y = s(x)$.

Proyecciones: Para todos $k \geq i \geq 1$, la función $\mathtt{proy}_i^k : \mathbb{N}^k \to \mathbb{N}$ definida por

$$\mathtt{proy}_i^k(n_1, \ldots, n_k) = n_i \qquad (n_1, \ldots, n_k) \in \mathbb{N}^k$$

es representada por la fórmula aritmética $\phi(x_1, \ldots, x_k, y) \equiv y = x_i$.

Esquema de composición: Sean $f_1, \ldots, f_p : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^p \rightharpoonup \mathbb{N}$ funciones parciales representadas respectivamente por unas fórmulas $\phi_{f_1}(x_1, \ldots, x_k, y), \ldots, \phi_{f_p}(x_1, \ldots, x_k, y)$ y $\phi_g(x_1, \ldots, x_p, y)$. Entonces, la función parcial $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ definida por

$$h(n_1, \ldots, n_k) = g(f_1(n_1, \ldots, n_k), \ldots, f_p(n_1, \ldots, n_k))$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho es definido) es representada por la fórmula aritmética

$$\begin{array}{rcl} \phi_h(x_1, \ldots, x_k, y) & \equiv & \exists z_1 \cdots \exists z_p\ [\phi_{f_1}(x_1, \ldots, x_k, z_1)\ \land \\ & & \qquad\qquad\qquad \vdots \\ & & \qquad\qquad \phi_{f_p}(x_1, \ldots, x_k, z_p)\ \land \\ & & \qquad\qquad \phi_g(z_1, \ldots, z_p, y) \quad] \end{array}$$

Esquema de minimización: Sea $f : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ una función parcial representada por una fórmula $\phi_f(x_1, \ldots, x_{k+1}, y)$. Entonces, la función parcial $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ definida por:

$$\begin{array}{rl} h(n_1, \ldots, n_k) & = \mu n \,.\, f(n, n_1, \ldots, n_k) > 0 \\ & = \text{el único } n \in \mathbb{N} \text{ (si existe) tal que} \\ & \quad f(n, n_1, \ldots, n_k) > 0\ \land\ (\forall m < n)\, f(m, n_1, \ldots, n_k) = 0 \end{array}$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho es definido) es representada por la fórmula aritmética

$$\begin{array}{rcl} \phi_h(x_1, \ldots, x_k, y) & \equiv & (\exists z > 0)\ \phi_f(y, x_1, \ldots, x_k, z)\ \land \\ & & (\forall y' < y)\ \phi_f(y', x_1, \ldots, x_k, 0) \end{array}$$

Esquema de recursión: Este caso es el objeto de la sección siguiente (2.4). □

2.4. Teorema chino del resto y función β de Gödel. La representación en PA de una función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ definida por el esquema de recursión

$$\begin{array}{rcl} h(0, \vec{p}) & = & f(\vec{p}) \\ h(n + 1, \vec{p}) & = & g(n, h(n, \vec{p}), \vec{p}) \end{array} \qquad (n \in \mathbb{N},\ \vec{p} \in \mathbb{N}^k)$$

es un problema delicado. Aquí, la dificultad viene de que para describir el valor devuelto por h para un entero n dado, se necesita describir la sucesión finita $r_0, \ldots, r_n$ de todos los valores devueltos por h para los enteros $m \leq n$:

$$\begin{array}{rcl} r_0 & = & f(\vec{p}) \\ r_1 & = & g(0, r_0, \vec{p}) \\ \vdots & & \vdots \\ r_n & = & g(n - 1, r_{n-1}, \vec{p}) \end{array}$$

Además, no se puede utilizar una variable distinta para cada elemento de la sucesión finita $r_0, \ldots, r_n$, pues el tamaño de dicha sucesión depende del entero (variable) n. Por lo tanto, se necesita definir una codificación de las sucesiones finitas de enteros naturales en los enteros naturales, y, sobre todo, una codificación que se pueda expresar en el lenguaje de PA.

El truco de Gödel consiste en utilizar el teorema chino del resto[9] para representar cada sucesión finita $r_1, \ldots, r_k$ por un entero natural n (en general muy largo) tal que

$$r_1 = n \text{ mód } p_1, \quad r_2 = n \text{ mód } p_2, \quad \cdots, \quad r_k = n \text{ mód } p_k$$

donde n mód p indica el resto de la división euclidiana de n por p, y donde $p_1, \ldots, p_k$ es una sucesión finita elegida juiciosamente (veremos abajo cómo).

El teorema chino del resto se enuncia así:

Teorema 2.16 (Teorema chino del resto). — *Sea $p_1, \ldots, p_k$ ($k \geq 1$) una sucesión finita de enteros positivos coprimos a pares. Entonces para toda sucesión de k enteros $r_1, \ldots, r_k$, existe un entero natural n tal que $n \equiv r_i \pmod{p_i}$ para todo $i \in [1..k]$.*

Demostración. Basta mostrar la existencia de un entero relativo $n \in \mathbb{Z}$ tal que $n \equiv r_i \pmod{p_i}$ para todo $i \in [1..k]$; para obtener un entero natural con la misma propiedad, se tomará $n + qp_1 \cdots p_k$ con q suficientemente grande. El teorema se demuestra por inducción sobre $k \geq 1$:

- $k = 1$. Inmediato.
- $k \geq 2$. Por hipótesis de inducción, existe un entero $m \in \mathbb{Z}$ tal que $m \equiv r_i \pmod{p_i}$ para todo $i \in [1..k-1]$. Como los enteros $p_1, \ldots, p_{k-1}, p_k$ son coprimos a pares, los dos enteros $p_1 \cdots p_{k-1}$ y p_k son coprimos. Por el lema de Bézout, existe enteros $a, b \in \mathbb{Z}$ tales que $ap_1 \cdots p_{k-1} + bp_k = 1$. Se puede tomar $n = r_k a p_1 \cdots p_{k-1} + mbp_k$, verificando que:
  - $n = r_k a p_1 \cdots p_{k-1} + mbp_k = r_k a p_1 \cdots p_{k-1} + m(1 - ap_1 \cdots p_{k-1})$
    $\equiv m \pmod{p_i} \equiv r_i \pmod{p_i}$ para todo $i \in [1..k-1]$
  - $n = r_k a p_1 \cdots p_{k-1} + mbp_k = r_k(1 - bp_k) + mbp_k \equiv r_k \pmod{p_k}$. □

En los cursos de Álgebra, este teorema se formula en general de la siguiente manera:

Corolario 2.17 (Teorema chino del resto, versión algebraica). — *Para toda sucesión finita $p_1, \ldots, p_k$ ($k \geq 1$) de enteros no nulos coprimos a pares, tenemos el isomorfismo de anillos:*

$$\mathbb{Z}/p_1 \cdots p_k\mathbb{Z} \;\cong\; (\mathbb{Z}/p_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/p_k\mathbb{Z})\,.$$

Demostración. Se considera el homomorfismo de anillos

$$\mathbb{Z}/p_1 \cdots p_k\mathbb{Z} \;\to (\mathbb{Z}/p_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/p_k\mathbb{Z})$$

definido por $f(n \text{ mód } p_1 \cdots p_k) = (n \text{ mód } p_1, \ldots, n \text{ mód } p_k)$ para todo $n \in \mathbb{N}$. Según el Teorema 2.16, este homomorfismo es sobreyectivo, y como los conjuntos de partida y de llegada tienen mismo cardinal, es un isomorfismo. □

A fin de representar una sucesión finita de enteros naturales $r_1, \ldots, r_k$ con el teorema chino del resto, tenemos que construir una sucesión de enteros $p_1, \ldots, p_k \neq 0$ coprimos a pares y tales que $p_i > r_i$ para todo $i \in [1..k]$. De nuevo, el truco consiste en elegir una sucesión aritmética finita cuyos términos cumplen las propiedades deseadas, utilizando el siguiente lema:

Lema 2.18. — *Para todos enteros $k \geq 0$ y $p \geq 1$, si p es divisible por todos los enteros entre 1 y k, entonces los $k + 1$ enteros*

$$p + 1, \quad 2p + 1, \quad \cdots, \quad (k + 1)p + 1$$

son coprimos a pares.

[9] La primera mención del teorema se encuentra en un libro del matemático chino Sun Tzu (siglo III).

Demostración. Dado dos índices i, j tales que $1 \leq i < j \leq k+1$, supongamos (por contradicción) que los dos enteros $ip+1$ y $jp+1$ no son coprimos, es decir, supongamos que existe un número primo q tal que $q|(ip+1)$ y $q|(jp+1)$. Por sustracción, tenemos que $q|((j-i)p)$. Se distinguen dos casos, según que $q|p$ o $q|(j-i)$.

- Caso donde $q|p$. Como $q|(ip+1)$, tenemos que $q|((ip+1)-ip)$ por sustracción, es decir: tenemos que $q|1$. Este caso es imposible.
- Caso donde $q|(j-i)$. Como $1 \leq j-i \leq k$, tenemos que $q \leq k$. Según nuestra hipótesis sobre p, así tenemos que $q|p$. Pero ya vimos en el caso anterior que es imposible.

Los dos casos llevan a una contradicción; luego los enteros $ip+1$ y $jp+1$ son coprimos. □

Definición 2.19 (Función β de Gödel)**.** — Se llama la *función β de Gödel* la función recursiva primitiva $\beta : \mathbb{N}^3 \to \mathbb{N}$ definida para todo $(n, p, i) \in \mathbb{N}^3$ por

$$\beta(n, p, i) \; = \; n \text{ mód } ((i+1)p+1)$$

donde n mód q indica el resto de la división euclidiana de n por q (con $q > 0$).

Combinando el lema anterior con el teorema chino del resto, tenemos que:

Lema 2.20. — *Para todo entero $k \geq 0$ y para toda sucesión finita $r_0, \ldots, r_k \in \mathbb{N}$, existe dos enteros n y p tales que $\beta(n, p, i) = r_i$ para todo $i \in [0..k]$.*

(Intuitivamente, los enteros n y p definen una codificación de la sucesión finita $r_0, \ldots, r_k$, en la medida en que la función β permite extraer todos los r_i a partir de n y p.)

Demostración. Se elige un entero $p > 0$ tal que $i|p$ para todo $i \in [1..k]$, y suficientemente grande para que $r_i \leq p$ para todo $i \in [0..k]$. Según el Lema 2.18, los enteros de la forma $(i+1)p+1$ (con $i \in [0..k]$) son coprimos en pares. Luego basta tomar un entero natural n tal que n mód $((i+1)p+1) = r_i$ para todo $i \in [0..k]$ (por el teorema chino del resto). □

Se verifica fácilmente que

Proposición 2.21 (Representación de la función β)**.** — *La función β de Gödel es representada en* PA *por la fórmula $\phi_\beta(u, v, x, y)$ definida por:*

$$\phi_\beta(u, v, x, y) \; \equiv \; \exists z \, (u = s(s(x) \times v) \times z + y \land y < s(s(x) \times v)) \, .$$

(En esta fórmula, las variables u, v, x representan los enteros n, p, i del lema anterior.)

Ahora se puede terminar la demostración del teorema de representación:

Fin de la demostración del Teorema 2.14 (Representación).

- **Esquema de recursión:** Sean $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$ funciones parciales representadas respectivamente por unas fórmulas $\phi_f(x_1, \ldots, x_k, y)$ y $\phi_g(x_1, \ldots, x_{k+1}, y)$. Entonces, la función parcial $h : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ definida por recursión sobre $n \in \mathbb{N}$ por:

$$\begin{array}{rcl} h(0, n_1, \ldots, n_k) & = & f(n_1, \ldots, n_k) \\ h(n+1, n_1, \ldots, n_k) & = & g(n, h(n, n_1, \ldots, n_k), n_1, \ldots, n_k) \end{array}$$

(para todo $(n, n_1, \ldots, n_k) \in \mathbb{N}^{k+1}$ donde el lado derecho es definido) es representada por la fórmula aritmética

$$\begin{array}{l} \phi_h(x, x_1, \ldots, x_k, y) \; \equiv \\ \quad \exists u \, \exists v \, \Big(\phi_\beta(u, v, x, y) \hfill \land \\ \qquad\qquad \exists z_0 \, (\phi_\beta(u, v, 0, z_0) \land \phi_f(x_1, \ldots, x_k, z_0)) \hfill \land \\ \qquad\qquad (\forall x' < x) \, \exists z \, \exists z' \, (\phi_g(x, z, x_1, \ldots, x_k, z') \land \\ \qquad\qquad\qquad\qquad \phi_\beta(u, v, x, z) \land \phi_\beta(u, v, s(x), z')) \Big) \end{array}$$

(La justificación de que esta fórmula representa efectivamente la función h en PA —en el sentido de la Def. 2.11— es un ejercicio muy técnico, que dejamos al lector.) $\square$

Informalmente, la fórmula $\phi_h(x, x_1, \ldots, x_k, y)$ dice que:

« Existe u y v tales que: *(existencia de una sucesión finita)*
- $\beta(u, v, x) = y$ *(último término de la sucesión)*
- $\beta(u, v, 0) = f(x_1, \ldots, x_k)$ *(primer término de la sucesión)*
- para todo $x' < x$: *(relación de recurrencia)*
  $\beta(u, v, x' + 1) = g(x, \beta(u, v, x), x_1, \ldots, x_k)$ »

3. Digitalización de PA

3.1. Abismación. El primer teorema de incompletitud se basa sobre la observación que todas las expresiones sintácticas de la Aritmética de Peano —los términos, las fórmulas y las derivaciones— son estructuras de datos finitas que se pueden representar por enteros naturales, a través de una codificación adecuada. El interés de tal *digitalización* de las estructuras sintácticas de la aritmética es que permite expresar propiedades formales —y razonar— sobre estas estructuras en el lenguaje mismo de la Aritmética de Peano, a través de la representación con números naturales. Así, la demostración del primer teorema de Gödel es una construcción en abismo, en la cual se construyen fórmulas (sintácticas) que hablan de fórmulas (numéricas) y demostraciones (sintácticas) que establecen propiedades sobre las demostraciones (numéricas). En práctica, la *abismación* de la teoría PA en el lenguaje de PA se hace en dos etapas:

1. A cada expresión sintáctica e de PA (un término, una fórmula o una demostración) se asocia un número natural $\ulcorner e \urcorner \in \mathbb{N}$, llamado el *código de Gödel* de e. Técnicamente, las inyecciones $t \mapsto \ulcorner t \urcorner$ (codificación de los términos), $\phi \mapsto \ulcorner \phi \urcorner$ (codificación de las fórmulas) y $d \mapsto \ulcorner d \urcorner$ (codificación de las derivaciones) son definidas a partir de una función recursiva de *emparejamiento* (Sección 3.2), de tal modo que todas las operaciones sintácticas de PA (sobre los términos, las fórmulas y las derivaciones) se puedan expresar por funciones recursivas sobre los códigos de Gödel correspondientes.
2. Luego, como todas las funciones que expresan las operaciones sintácticas de PA à través de la codificación son funciones recursivas (primitivas), se puede aplicar el teorema de representación (Sección 2.3) a dichas funciones, con el fin de representarlas en el lenguaje de PA por fórmulas adecuadas (en general muy largas).

3.2. Emparejamiento. La codificación de las expresiones sintácticas de PA es definida a partir de una inyección recursiva $\langle \_, \_ \rangle : \mathbb{N}^2 \to \mathbb{N}$ de *emparejamiento*, que asocia a cada par de enteros naturales n y m un entero natural $\langle n, m \rangle$. En lo siguiente, se considera la biyección de $\mathbb{N}^2$ sobre $\mathbb{N}$ que corresponde a la enumeración diagonal de los pares de enteros naturales

$$\begin{array}{c|ccccccccccc}
\cdots & \cdots \\
4 & 14 & & \cdots \\
 & & \nwarrow \\
3 & 9 & & 13 & & \cdots \\
 & & \nwarrow & & \nwarrow \\
2 & 5 & & 8 & & 12 & & \cdots \\
 & & \nwarrow & & \nwarrow & & \nwarrow \\
1 & 2 & & 4 & & 7 & & 11 & & \cdots \\
 & & \nwarrow & & \nwarrow & & \nwarrow & & \nwarrow \\
0 & 0 & & 1 & & 3 & & 6 & & 10 & & \cdots \\
\hline
{}^{n}{}_{m} & 0 & & 1 & & 2 & & 3 & & 4 & & \cdots
\end{array}$$

y que es definida para todos $m, n \in \mathbb{N}$ por:

$$\langle n, m\rangle = (n+m)(n+m+1)/2 + n\,.$$

Proposición 3.1 (Propiedades de la función de emparejamiento)**.**

(1) *La función* $\langle\_,\_\rangle : \mathbb{N}^2 \to \mathbb{N}$ *es una biyección recursiva primitiva.*

(2) *Las dos proyecciones* $\pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$ *definidas por* $\pi_1(\langle n, m\rangle) = n$ *y* $\pi_2(\langle n, m\rangle) = m$ *(para todos* $n, m \in \mathbb{N}$*) son recursivas primitivas también.*

(3) *Para todos* $n, m \in \mathbb{N}$*:* $\langle n, m\rangle \geq \text{máx}(n, m)$.
Además, si $n \geq 1$*, entonces:* $\langle n, m\rangle > \text{máx}(n, m)$.

Demostración. (1) El carácter recursivo primitivo de la función $(n, m) \mapsto \langle n, m\rangle$ es obvio según la definición. Para demostrar que la función es biyectiva, vamos a construir explícitamente las dos proyecciones $\pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$. Sea $f : \mathbb{N} \to \mathbb{N}$ la función (recursiva primitiva) definida por $f(k) = k(k+1)/2$ para todo $k \in \mathbb{N}$. Por construcción, tenemos que

$$f(n+m) \;\leq\; f(n+m) + n \;=\; \langle n, m\rangle \;<\; f(n+m+1)$$

para todos $n, m \in \mathbb{N}$. Para invertir la función de emparejamiento, se observa que la función $f : \mathbb{N} \to \mathbb{N}$ es estrictamente creciente y empieza a partir de $f(0) = 0$. Entonces, para todo $p \in \mathbb{N}$, existe un único $k \in \mathbb{N}$ tal que $f(k) \leq p < f(k+1)$. Sea $g : \mathbb{N} \to \mathbb{N}$ la función que calcula k en función de p, la cual se puede definir por minimización acotada, escribiendo

$$g(p) = \mu k \leq p\,.\,[p < f(k+1)] \qquad (p \in \mathbb{N})$$

Ahora se pueden definir las funciones $\pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$ para todo $p \in \mathbb{N}$, escribiendo

$$\pi_1(p) \;=\; p - f(g(p)) \qquad \text{y} \qquad \pi_2(p) \;=\; g(p) - \pi_1(p)$$

Para todos $n, m \in \mathbb{N}$, se verifica inmediatamente que

$$\begin{aligned} g(\langle n, m\rangle) &= n+m & \qquad \pi_1(\langle n, m\rangle) &= \langle n, m\rangle - f(g(\langle n, m\rangle)) = n \\ f(g(\langle n, m\rangle)) &= (n+m)(n+m+1)/2 & \qquad \pi_2(\langle n, m\rangle) &= g(\langle n, m\rangle) - \pi_1(\langle n, m\rangle) = m \end{aligned}$$

las dos igualdades a la derecha implicando que la función $\langle\_,\_\rangle : \mathbb{N}^2 \to \mathbb{N}$ es inyectiva. Para demostrar que también es sobreyectiva, se observa que para todo $p \in \mathbb{N}$, tenemos que:

$$\begin{aligned} \langle \pi_1(p), \pi_2(p)\rangle &= (\pi_1(p) + \pi_2(p))(\pi_1(p) + \pi_2(p) + 1)/2 + \pi_1(p) && (\text{def. de } \langle\_,\_\rangle) \\ &= g(p)(g(p)+1)/2 + \pi_1(p) && (\text{def. de } \pi_2) \\ &= f(g(p)) + \pi_1(p) \;=\; p && (\text{def. de } f \text{ y de } \pi_1) \end{aligned}$$

(2) Obvio según la construcción del ítem (1).

(3) Como la función $f(k) = k(k+1)/2$ es estrictamente creciente (véase (1)), tenemos que $f(k) \geq k$ para todo $k \in \mathbb{N}$. Entonces, para todos $n, m \in \mathbb{N}$ tenemos que:

$$\langle n, m\rangle \;=\; f(n+m) + n \;\geq\; (n+m) + n \;\geq\; \text{máx}(n, m)\,,$$

la última desigualdad siendo estricta cuando $n \geq 1$. □

3.3. Codificación de las listas finitas. La función recursiva primitiva $\langle\_,\_\rangle : \mathbb{N}^2 \to \mathbb{N}$ de emparejamiento permite representar más generalmente todas las listas finitas de enteros con enteros. Tal codificación de las listas finitas se construye a partir de:

- el entero `nil` $= 0$, que representa la lista vacía;
- la construcción `cons`$(l, n) = \langle l, n\rangle + 1$, que representa la lista obtenida añadiendo a (la derecha[10] de) la lista finita representada por el entero l el entero n.

[10] Aquí se introducen sucesivamente los elementos por la derecha, afín de quedarse consistente con la política de introducción de las hipótesis en los contextos de los secuentes (Sección 1.3).

Así, cada lista finita de enteros naturales $n_1, n_2, \ldots, n_k$ se puede representar por el entero natural $[n_1, n_2, \ldots, n_k]$ definido por:

$$[n_1, n_2, \ldots, n_k] \;=\; \mathtt{cons}(\ldots\mathtt{cons}(\mathtt{cons}(\mathtt{nil}, n_1), n_2)\ldots, n_k)\,.$$

Proposición 3.2 (Biyección entre $\mathbb{N}^{<\omega}$ y $\mathbb{N}$). — *La función $n_1, \ldots, n_k \mapsto [n_1, \ldots, n_k]$ es una biyección entre el conjunto $\mathbb{N}^{<\omega}$ de las listas finitas de enteros naturales y $\mathbb{N}$.*

En lo siguiente, utilizaremos esta codificación de las listas finitas para representar los contextos de hipótesis en los secuentes.

3.4. Digitalización de los términos. A cada término t del lenguaje de PA se asocia un *número de Gödel* $\ulcorner t \urcorner \in \mathbb{N}$, definido recursivamente por las ecuaciones

$$\begin{array}{rcl@{\qquad\qquad}rcl} \ulcorner x \urcorner & = & \langle 0, \#x \rangle & \ulcorner t + u \urcorner & = & \langle 2, \langle \ulcorner t \urcorner, \ulcorner u \urcorner \rangle \rangle \\ \ulcorner s(t) \urcorner & = & \langle 1, \ulcorner t \urcorner \rangle & \ulcorner t \times u \urcorner & = & \langle 3, \langle \ulcorner t \urcorner, \ulcorner u \urcorner \rangle \rangle \end{array}$$

donde $x \mapsto \#x$ es una biyección fijada entre el conjunto (numerable) de las variables y $\mathbb{N}$.

Observación 3.3. — Utilizando la Prop. 3.1 (3), se verifica fácilmente que para todos términos t, u del lenguaje de PA, tenemos que:

$$\ulcorner s(t) \urcorner \;>\; \ulcorner t \urcorner \qquad\quad \ulcorner t + u \urcorner \;>\; \text{máx}(\ulcorner t \urcorner, \ulcorner u \urcorner) \qquad\quad \ulcorner t \times u \urcorner \;>\; \text{máx}(\ulcorner t \urcorner, \ulcorner u \urcorner)\,.$$

Por consecuencia, la codificación $t \mapsto \ulcorner t \urcorner$ respeta el orden de los subtérminos, en el sentido en que si u es un subtérmino estricto de t, entonces $\ulcorner u \urcorner < \ulcorner t \urcorner$.

Esta observación permite demostrar lo siguiente:

Proposición 3.4. — *La codificación de los términos $t \mapsto \ulcorner t \urcorner$ es inyectiva.*

Demostración. Es suficiente demostrar que la función de codificación tiene una función inversa (a la izquierda), es decir: una función parcial $\mathsf{T} : \mathbb{N} \rightharpoonup \mathscr{L}_T(\mathrm{PA})$ (de los naturales hasta los términos de PA) tal que para todo término t del lenguaje de PA, el término $\mathsf{T}(\ulcorner t \urcorner)$ sea definido y idéntico a t: $\mathsf{T}(\ulcorner t \urcorner) \equiv t$. Esta función inversa $\mathsf{T} : \mathbb{N} \rightharpoonup \mathscr{L}_T(\mathrm{PA})$ es definida por inducción fuerte sobre $\mathbb{N}$, utilizando las ecuaciones:

$$\begin{array}{rcll} \mathsf{T}(\langle 0, m \rangle) & \equiv & \text{la variable } x \text{ tal que } n = \#x & \\ \mathsf{T}(\langle 1, m \rangle) & \equiv & s(\mathsf{T}(m)) & \text{(si } \mathsf{T}(m) \text{ es definido)} \\ \mathsf{T}(\langle 2, m \rangle) & \equiv & \mathsf{T}(\pi_1(m)) + \mathsf{T}(\pi_2(m)) & \text{(si } \mathsf{T}(\pi_1(m)) \text{ y } \mathsf{T}(\pi_2(m)) \text{ son definidos)} \\ \mathsf{T}(\langle 3, m \rangle) & \equiv & \mathsf{T}(\pi_1(m)) \times \mathsf{T}(\pi_2(m)) & \text{(si } \mathsf{T}(\pi_1(m)) \text{ y } \mathsf{T}(\pi_2(m)) \text{ son definidos)} \end{array}$$

(el valor de $\mathsf{T}(\langle n, m \rangle)$ siendo indefinido en todos los otros casos). La corrección (buena fundación) de la definición de la función parcial T sigue de la Prop. 3.1 (3). □

Además:

Proposición 3.5. — *El conjunto de los códigos de términos*

$$\mathtt{Term} \;=\; \{\ulcorner t \urcorner \;:\; t \textit{ término de } \mathrm{PA}\} \qquad\qquad (\subset \mathbb{N})$$

es recursivo primitivo.

Demostración. La función característica del conjunto $\mathtt{Term} \subseteq \mathbb{N}$ se puede calcular por el siguiente algoritmo recursivo:

$$
\begin{array}{l}
\mathtt{check\_term}(n) \ := \\
\quad \text{sea } t := \pi_1(n) \\
\quad \text{sea } m := \pi_2(n) \\
\quad \text{si } t = 0: \quad \text{devolver } 1 \\
\quad \text{si } t = 1: \quad \text{devolver } \mathtt{check\_term}(m) \\
\quad \text{si } t = 2: \quad \text{devolver } \mathtt{check\_term}(\pi_1(m)) \times \mathtt{check\_term}(\pi_2(m)) \\
\quad \text{si } t = 3: \quad \text{devolver } \mathtt{check\_term}(\pi_1(m)) \times \mathtt{check\_term}(\pi_2(m)) \\
\quad \text{si no}: \quad \text{devolver } 0
\end{array}
$$

El carácter recursivo primitivo (y total) de este algoritmo sigue de que cada llamada recursiva a la función `check_term` se hace con un argumento estrictamente menor que n. □

Asimismo:

Proposición 3.6. — *El conjunto de los códigos de términos cerrados*

$$\mathtt{Term}_0 \ = \ \{\ulcorner t \urcorner \ : \ t \text{ término de PA}\} \qquad (\subset \mathtt{Term})$$

es recursivo primitivo.

Se verifica más generalmente que las operaciones usuales sobre los términos (test de ocurrencia de variable, sustitución, etc.) corresponden, a través de esta codificación, a funciones recursivas. En lo siguiente, se utilizará la función $\mathtt{pnum} : \mathbb{N} \to \mathbb{N}$ que a cada $n \in \mathbb{N}$ asocia el código del término $\overline{n} = s^n(0)$, y que es definida por recursión primitiva por:

$$
\begin{aligned}
\mathtt{pnum}(0) &= \langle 0, 0 \rangle \\
\mathtt{pnum}(n+1) &= \langle 2, \mathtt{pnum}(n) \rangle
\end{aligned}
$$

de tal modo que:

$$\mathtt{pnum}(n) \ = \ \ulcorner \overline{n} \urcorner \ = \ \ulcorner \underbrace{s(\cdots s}_{n}(0)\cdots) \urcorner \qquad (n \in \mathbb{N})$$

3.5. Digitalización de las fórmulas. A cada fórmula ϕ del lenguaje de PA se asocia un *número de Gödel* $\ulcorner \phi \urcorner \in \mathbb{N}$, definido recursivamente por las ecuaciones

$$
\begin{aligned}
\ulcorner \bot \urcorner &= \langle 0, 0 \rangle & \ulcorner \phi \Rightarrow \psi \urcorner &= \langle 2, \langle \ulcorner \phi \urcorner, \ulcorner \psi \urcorner \rangle \rangle \\
\ulcorner t = u \urcorner &= \langle 1, \langle \ulcorner t \urcorner, \ulcorner u \urcorner \rangle \rangle & \ulcorner \forall x\, \phi \urcorner &= \langle 3, \langle \#x, \ulcorner \phi \urcorner \rangle \rangle
\end{aligned}
$$

Observación 3.7. — Utilizando la Prop. 3.1 (3), se verifica que para todas fórmulas ϕ, ψ del lenguaje de PA, tenemos que:

$$\ulcorner \phi \Rightarrow \psi \urcorner \ > \ \text{máx}(\ulcorner \phi \urcorner, \ulcorner \psi \urcorner) \qquad\qquad \ulcorner \forall x\, \phi \urcorner \ > \ \ulcorner \phi \urcorner$$

Por consecuencia, la codificación $\phi \mapsto \ulcorner \phi \urcorner$ respeta el orden de las subfórmulas, en el sentido en que si ϕ es una subfórmula estricta de ψ, entonces $\ulcorner \phi \urcorner < \ulcorner \psi \urcorner$.

Como para los términos, esta observación permite demostrar los siguientes resultados:

Proposición 3.8. — *La codificación de las fórmulas $\phi \mapsto \ulcorner \phi \urcorner$ es inyectiva.*

Demostración. Mismo método como para los términos. □

Observación 3.9. — Se puede observar que las dos funciones de codificación $t \mapsto \ulcorner t \urcorner$ (para los términos) y $\phi \mapsto \ulcorner \phi \urcorner$ (para las fórmulas) no tienen imágenes disjuntas, de tal modo que un entero natural puede ser a la vez un código de término y un código de fórmula. Esto no induce ninguna confusión en práctica, porque los términos y las fórmulas no se encuentran en los mismos contextos. La misma observación también vale para los códigos $\#x$ de variables

—de hecho *todo* entero natural es un código de variable— y los códigos $\ulcorner d \urcorner$ de derivaciones que definiremos en la sección 3.6.

Proposición 3.10. — *El conjunto de los códigos de fórmulas*

$$\mathtt{Form} \ = \ \{\ulcorner\phi\urcorner \ : \ \phi \textit{ fórmula de } \mathrm{PA}\} \qquad (\subset \mathbb{N})$$

es recursivo primitivo.

Demostración. La función característica del conjunto $\mathtt{Form} \subseteq \mathbb{N}$ se puede calcular por el siguiente algoritmo recursivo:

$\mathtt{check\_form}(n) \ :=$
  sea $t := \pi_1(n)$
  sea $m := \pi_2(n)$
  si $t = 0 \wedge m = 0$: devolver 1
  si $t = 1$: devolver $\mathtt{check\_term}(\pi_1(m)) \times \mathtt{check\_term}(\pi_2(m))$
  si $t = 2$: devolver $\mathtt{check\_form}(\pi_1(m)) \times \mathtt{check\_form}(\pi_2(m))$
  si $t = 3$: devolver $\mathtt{check\_form}(\pi_2(m))$
  si no : devolver 0

El carácter recursivo primitivo (y total) de este algoritmo sigue de que cada llamada recursiva a la función check_form se hace con un argumento estrictamente menor que n. □

Asimismo:

Proposición 3.11. — *El conjunto de los códigos de fórmulas cerradas*

$$\mathtt{Form}_0 \ = \ \{\ulcorner\phi\urcorner \ : \ \phi \textit{ fórmula cerrada de } \mathrm{PA}\} \qquad (\subset \mathtt{Form})$$

es recursivo primitivo.

Más generalmente, las operaciones sintácticas usuales sobre las fórmulas (test de ocurrencia libre/ligada de variable, sustitución, test de α-conversión, etc.) corresponden, a través de la codificación $\phi \mapsto \ulcorner\phi\urcorner$, a funciones recursivas primitivas. En lo siguiente, se utilizarán las siguientes funciones recursivas primitivas:

- La función $\mathtt{fneg} : \mathbb{N} \to \mathbb{N}$ de negación, definida para todo $n \in \mathbb{N}$ por:

$$\mathtt{fneg}(n) \ = \ \langle 2, \langle n, \langle 0, 0\rangle\rangle\rangle$$

  de tal modo que $\mathtt{fneg}(\ulcorner\phi\urcorner) = \ulcorner\phi \Rightarrow \bot\urcorner = \ulcorner\neg\phi\urcorner$ para toda fórmula ϕ.
- La función $\mathtt{fsubst} : \mathbb{N}^3 \to \mathbb{N}$ de sustitución, definida por

$$\mathtt{fsubst}(\ulcorner\phi\urcorner, \#x, \ulcorner t\urcorner) \ = \ \ulcorner\phi\{x := t\}\urcorner$$

  y completada por la ecuación $\mathtt{fsubst}(n, v, m) = 0$ en el caso donde n no es un código de fórmula o m no es un código de término.

Además, es importante observar que:

Proposición 3.12. — *El conjunto de los códigos de los axiomas de* PA *(A_1–A_7)*

$$\mathtt{Axiom} \ = \ \{\ulcorner\phi\urcorner \ : \ \phi \in \mathrm{Ax}(\mathrm{PA})\} \qquad (\subset \mathtt{Form}_0)$$

es recursivo primitivo.

En lo siguiente, utilizaremos la función recursiva primitiva $\mathtt{check\_axiom\_list} : \mathbb{N} \to \mathbb{N}$ que verifica si su argumento es una lista de códigos de Gödel de axiomas de PA:

$$\mathtt{check\_axiom\_list}([n_1, \ldots, n_k]) \ = \ \begin{cases} 1 & \text{si } l = [n_1, \ldots, n_k] \text{ con} \\ & \quad n_1, \ldots, n_k \in \mathtt{Axiom} \\ 0 & \text{si no} \end{cases}$$

Esta función se implementa fácilmente a partir de la función característica del conjunto `Axiom` (escrita aquí `check_axiom`) con el siguiente algoritmo:

> `check_axiom_list`(l) :=
>   *{ verificar recursivamente las fórmulas en l }*
>   mientras $l \neq$ `nil` :
>   *{ verificar la última fórmula en l}*
>   si `check_axiom`($\pi_2(l-1)$) = 0 :   devolver 0
>   si no : $l := \pi_1(l-1)$ *{ sacar la última fórmula }*
>   devolver 1   *{ éxito }*

(***N.B.*:** El entero $\pi_2(l-1)$ representa el último elemento de la lista finita l, mientras $\pi_1(l-1)$ representa la lista l privada de su último elemento, véase Sección 3.3.)

3.6. Digitalización de las derivaciones. La digitalización de las derivaciones (Sección 1.3) se hace en tres etapas:

- A cada contexto de hipótesis $\Gamma \equiv \phi_1, \ldots, \phi_n$ se asocia el número de Gödel

$$\ulcorner\Gamma\urcorner \;=\; [\ulcorner\phi_1\urcorner, \ldots, \ulcorner\phi_n\urcorner]\,,$$

  utilizando la codificación de las listas finitas (Sección 3.3).
- A cada secuente $\Gamma \vdash \phi$ se asocia el número de Gödel

$$\ulcorner\Gamma \vdash \phi\urcorner \;=\; \langle\ulcorner\Gamma\urcorner, \ulcorner\phi\urcorner\rangle\,.$$

- Finalmente, a cada derivación

$$d \;\equiv\; \left\{\dfrac{\begin{matrix}\vdots\, d_1 & & \vdots\, d_n \\ \Gamma_1 \vdash \phi_1 & \cdots & \Gamma_n \vdash \phi_n\end{matrix}}{\Gamma \vdash \phi}\right.$$

  (construida por aplicación finita de las reglas del Cuadro 1 p. 6), se asocia el número de Gödel $\ulcorner d\urcorner$ definido por recursión sobre la estructura arborescente de d por

$$\ulcorner d\urcorner \;=\; \langle\ulcorner\Gamma \vdash \phi\urcorner, \langle r, [\ulcorner d_1\urcorner, \ldots, \ulcorner d_n\urcorner]\rangle\rangle\,, \qquad (n \in \mathbb{N})$$

  donde r es un entero entre 0 y 8 que indica la última regla utilizada[11] para construir la derivación d (0 = axioma, 1 = debilitamiento, ..., 8 = razonamiento por el absurdo).

Proposición 3.13. — *El conjunto de los códigos de derivaciones*

$$\texttt{Deriv} \;=\; \{\ulcorner d\urcorner \;:\; d \textit{ derivación}\} \qquad (\subset \mathbb{N})$$

es recursivo primitivo.

[11] Estrictamente hablando, no se necesita introducir tal número en la codificación, en la medida en que la última regla utilizada para construir d siempre se puede deducir a partir de la forma de la conclusión y de las premisas. Sin embargo, el conocimiento de dicha regla facilita mucho el diseño del algoritmo de verificación.

Demostración. La función característica del conjunto $\mathtt{Deriv} \subseteq \mathbb{N}$ se puede calcular por el siguiente algoritmo recursivo:

$\mathtt{check\_deriv}(d)$:=
  sea $c := \pi_1(d)$ *{ conclusión de d }*
  sea $r := \pi_1(\pi_2(d))$ *{ última regla utilizada }*
  sea $l := \pi_2(\pi_2(d))$ *{ derivaciones de las premisas }*
  { extraer la lista de las premisas }
  sea $l' := \mathtt{map\_pi1}(l)$
  { verificar la buena formación de la regla utilizada }
  si $\mathtt{check\_rule}(c, r, l') = 1$:
    { verificar recursivamente las derivaciones en l }
    mientras $l \neq \mathtt{nil}$:
      { verificar la última derivación en l }
      si $\mathtt{check\_deriv}(\pi_2(l-1)) = 0$:   devolver 0
      si no: $l := \pi_1(l-1)$ *{ sacar la última derivación }*
    devolver 1   *{ éxito }*
  si no: devolver 0   *{ regla incorrecta }*

Este algoritmo utiliza dos funciones recursivas primitivas auxiliares, cuya implementación es dejada en ejercicio al lector:

- Una función $\mathtt{map\_pi1}(l) : \mathbb{N} \to \mathbb{N}$ tal que

$$\mathtt{map\_pi1}([n_1, \ldots, n_k]) = [\pi_1(n_1), \ldots, \pi_1(n_k)]$$

  para todo $[n_1, \ldots, n_k] \in \mathbb{N}$.
- Una función $\mathtt{check\_rule} : \mathbb{N}^3 \to \mathbb{N}$ tal que

$$\begin{cases} \mathtt{check\_rule}(\ulcorner\Gamma \vdash \phi\urcorner, r, [\ulcorner\Gamma_1 \vdash \phi_1\urcorner, \ldots, \ulcorner\Gamma_n \vdash \phi_n\urcorner]) \;=\; 1 \\ \qquad \text{si} \quad \Gamma_1 \vdash \phi_1, \ldots, \Gamma_n \vdash \phi_n \;/\; \Gamma \vdash \phi \quad \text{es una instancia de la regla } r \in [0..8] \\ \mathtt{check\_rule}(c, r, l) \;=\; 0 \quad \text{si no} \end{cases}$$

El carácter recursivo primitivo (y total) del algoritmo sigue de que cada llamada recursiva a la función $\mathtt{check\_deriv}$ se hace con un argumento estrictamente menor que d. □

Finalmente:

Corolario 3.14. — *El conjunto*

$$\mathtt{Proof} \;=\; \{(\ulcorner d\urcorner, \ulcorner\phi\urcorner) \;:\; d \text{ es una demostración de } \phi \text{ en PA}\} \qquad (\subset \mathbb{N})$$

es recursivo primitivo.

Demostración. La función característica del conjunto $\mathtt{Proof} \subseteq \mathbb{N}^2$ se calcula por:

$\mathtt{check\_proof}(d, n)$:=
  devolver $\mathtt{check\_deriv}(d)\ \times$
      $\mathtt{check\_form0}(n)\ \times$
      $\mathtt{check\_equal}(\pi_2(\pi_1(d)), n)\ \times$
      $\mathtt{check\_axiom\_list}(\pi_1(\pi_1(d)))$

donde $\mathtt{check\_form0} : \mathbb{N} \to \mathbb{N}$ es la función característica del conjunto $\mathtt{Form}_0 \subseteq \mathbb{N}$ (códigos de las fórmulas cerradas), y $\mathtt{check\_equal} : \mathbb{N}^2 \to \mathbb{N}$ la función de test de igualdad. □

3.7. Algunos conjuntos recursivos y no recursivos. En las secciones 3.4–3.6, vimos que los siguientes conjuntos

$$\begin{array}{rcll}
\mathtt{Term} & = & \{\ulcorner t \urcorner \ :\ t \text{ término de PA}\} & (\subset \mathbb{N}) \\
\mathtt{Term}_0 & = & \{\ulcorner t \urcorner \ :\ t \text{ término cerrado de PA}\} & (\subset \mathtt{Term}) \\
\mathtt{Form} & = & \{\ulcorner \phi \urcorner \ :\ \phi \text{ fórmula de PA}\} & (\subset \mathbb{N}) \\
\mathtt{Form}_0 & = & \{\ulcorner \phi \urcorner \ :\ \phi \text{ fórmula cerrada de PA}\} & (\subset \mathtt{Form}) \\
\mathtt{Axiom} & = & \{\ulcorner \phi \urcorner \ :\ \phi \text{ axioma de PA}\} & (\subset \mathtt{Form}_0) \\
\mathtt{Deriv} & = & \{\ulcorner d \urcorner \ :\ \phi \text{ derivación bien formada}\} & (\subset \mathbb{N}) \\
\mathtt{Proof} & = & \{(\ulcorner d \urcorner, \ulcorner \phi \urcorner) \ :\ d \text{ demostración de } \phi \text{ en PA}\} & (\subset \mathbb{N}^2)
\end{array}$$

son recursivos primitivos. También se verifica que el conjunto

$$\begin{array}{rcll}
\mathtt{Proof}^\neg & = & \{(\ulcorner d \urcorner, \ulcorner \phi \urcorner) \ :\ d \text{ demostración de } \neg\phi \text{ en PA}\} & (\subset \mathbb{N}) \\
 & = & \{(m,n) \in \mathbb{N}^2 \ :\ (m, \mathtt{fneg}(n)) \in \mathtt{Proof}\} & (\subset \mathbb{N}^2)
\end{array}$$

(conjunto de los "códigos de refutaciones") es primitivo recursivo.

Códigos de fórmulas demostrables y refutables A partir de los conjuntos $\mathtt{Proof}$ y $\mathtt{Proof}^\neg$ ($\subseteq \mathbb{N}^2$), se definen los conjuntos

$$\begin{array}{rcll}
\mathtt{Dem} & = & \{n \in \mathbb{N} \ :\ (m,n) \in \mathtt{Proof} \text{ para algún } m \in \mathbb{N}\} & (\subset \mathtt{Form}_0) \\
\mathtt{Dem}^\neg & = & \{n \in \mathbb{N} \ :\ (m,n) \in \mathtt{Proof}^\neg \text{ para algún } m \in \mathbb{N}\} & \\
 & = & \{n \in \mathbb{N} \ :\ \mathtt{fneg}(n) \in \mathtt{Dem}\} & (\subset \mathtt{Form}_0)
\end{array}$$

cuyos elementos son los códigos de fórmulas demostrables y refutables en PA. Es claro que:

- Si PA es consistente, entonces: $\mathtt{Dem} \cup \mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ y $\mathtt{Dem} \cap \mathtt{Dem}^\neg = \varnothing$.
- Si PA es inconsistente, entonces: $\mathtt{Dem} = \mathtt{Dem}^\neg = \mathtt{Form}_0$.

Proposición 3.15. — *Los dos conjuntos* $\mathtt{Dem}, \mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ *son recursivamente enumerables.*

Demostración. El conjunto $\mathtt{Dem} \subseteq \mathtt{Form}_0$ es recursivamente enumerable (Def. 2.9) pues es el dominio de la función recursiva parcial $\mathtt{search\_proof} : \mathbb{N} \rightharpoonup \mathbb{N}$ dada por el algoritmo:

$\mathtt{search\_proof}(n)$:=
  sea $m := 0$
  mientras $\mathtt{check\_proof}(m,n) \neq 1$: $m := m + 1$
  { se sale del bucle anterior si y sólo si se encontró
   una demostración de la fórmula de código n }
  devolver 1

Asimismo, el conjunto $\mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ es recursivamente enumerable, pues es el dominio de la función recursiva parcial $\mathtt{search\_proof} \circ \mathtt{fneg} : \mathbb{N} \rightharpoonup \mathbb{N}$. □

En la sección 4.5 p. 39, veremos que ninguno de los dos conjuntos $\mathtt{Dem}, \mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ es recursivo (bajo la hipótesis de consistencia de PA).

Observación 3.16. — Según un punto de vista más algebraico, se puede ver los dos conjuntos $\mathtt{Dem}, \mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ como las imágenes de los conjuntos $\mathtt{Proof}, \mathtt{Proof}^\neg \subseteq \mathbb{N}^2$ por la función $\pi_2 : \mathbb{N}^2 \to \mathbb{N}$ (« segunda proyección ») definida por $\pi_2(m,n) = n$ para todo $(m,n) \in \mathbb{N}^2$:

$$\mathtt{Dem} = \pi_2(\mathtt{Proof}) \qquad \text{y} \qquad \mathtt{Dem}^\neg = \pi_2(\mathtt{Proof}^\neg).$$

Según un resultado estándar de la teoría de la recursión, la imagen de cualquier conjunto recursivamente enumerable por una función recursiva también es un conjunto recursivamente enumerable. Este argumento brinda otra demostración (más abstracta) de que los dos conjuntos $\mathtt{Dem}, \mathtt{Dem}^\neg \subseteq \mathtt{Form}_0$ son recursivamente enumerables.

Códigos de fórmulas verdaderas y falsas Cuando la teoría ambiente permite definir el modelo estándar (véase la discusión de la Sección 1.7), es posible definir los conjuntos de los códigos de fórmulas verdaderas y falsas:

$$\begin{aligned} \mathtt{True} &= \{\ulcorner\phi\urcorner : \phi \text{ cerrada y } \mathbb{N} \models \phi\} && (\subset \mathtt{Form}_0) \\ \mathtt{False} &= \{\ulcorner\phi\urcorner : \phi \text{ cerrada y } \mathbb{N} \not\models \phi\} && (\subset \mathtt{Form}_0) \end{aligned}$$

Por construcción, los dos conjuntos $\mathtt{True}$ y $\mathtt{False}$ forman una partición del conjunto $\mathtt{Form}_0$ de los códigos de fórmulas cerradas:

$$\mathtt{True} \uplus \mathtt{False} = \mathtt{Form}_0 .$$

Además, el teorema de corrección (Teorema 1.26) implica las inclusiones

$$\mathtt{Dem} \subseteq \mathtt{True} \qquad \text{y} \qquad \mathtt{Dem}^\neg \subseteq \mathtt{False} .$$

En la sección 4.6 p. 40, veremos que ninguno de los dos conjuntos $\mathtt{True}, \mathtt{False} \subset \mathtt{Form}_0$ es recursivamente enumerable (bajo la hipótesis de existencia del modelo estándar).

4. El primer teorema de incompletitud

4.1. Construcción de fórmulas autorreferenciales. Así como lo veremos en la Sección 4.2, la demostración del primer teorema de incompletitud es esencialmente una adaptación de la paradoja del mentiroso (« esta oración es falsa ») al marco formal de la Aritmética de Peano, basada sobre la construcción de una fórmula G que dice: « no soy demostrable ». Para construir tal fórmula autorreferencial, necesitaremos el siguiente teorema:

Teorema 4.1 (Punto fijo)**.** — *Para cada fórmula aritmética $\psi \equiv \psi(x)$ con una única variable libre x, existe una fórmula aritmética cerrada $\psi^\bullet$ tal que*

$$\mathrm{PA} \vdash \psi^\bullet \Leftrightarrow \psi\Big(\overline{\ulcorner\psi^\bullet\urcorner}\Big),$$

donde $\ulcorner\psi^\bullet\urcorner$ es el código de Gödel de la fórmula $\psi^\bullet$.

Intuitivamente, la fórmula cerrada $\psi^\bullet$ dice:

« mi propio código de Gödel satisface la propiedad $\psi(x)$ ».

En lo siguiente, se dice que tal fórmula —que expresa una propiedad sobre su propio código de Gödel— es una fórmula *autorreferencial*. Otro punto de vista consiste en ver la fórmula $\psi^\bullet$ como un punto fijo de la propiedad $\psi(x)$, a través de la codificación de Gödel. Así, el teorema expresa que cada propiedad tiene un punto fijo en este sentido.

Demostración. Sea $h : \mathbb{N} \to \mathbb{N}$ la función recursiva primitiva definida para todo $n \in \mathbb{N}$ por $h(n) = \mathtt{fsubst}(n, \#x, \mathrm{pnum}(n))$. Por construcción, tenemos que

$$h(\ulcorner\chi\urcorner) = \mathtt{fsubst}(\ulcorner\chi\urcorner, \#x, \mathrm{pnum}(\ulcorner\chi\urcorner)) = \ulcorner\chi\{x := \overline{\ulcorner\chi\urcorner}\}\urcorner$$

para toda fórmula aritmética χ. Según el Teorema 2.14, existe una fórmula $\phi_h(x, y)$ tal que

$$\mathrm{PA} \vdash \forall y \Big(\phi_h(\overline{n}, y) \Leftrightarrow y = \overline{h(n)}\Big)$$

para todo $n \in \mathbb{N}$. En particular, tenemos que

$$(*) \qquad \mathrm{PA} \vdash \forall y \Big(\phi_h\Big(\overline{\ulcorner\chi\urcorner}, y\Big) \Leftrightarrow y = \overline{\ulcorner\chi\{x := \overline{\ulcorner\chi\urcorner}\}\urcorner}\Big)$$

para toda fórmula aritmética χ. Ahora, se definen las fórmulas $\chi \equiv \exists y\,(\psi(y) \wedge \phi_h(x,y))$ (con única variable libre x) y $\psi^\bullet \equiv \chi\{x := \overline{\ulcorner\chi\urcorner}\}$ (cerrada). Por construcción, tenemos que:

$$\begin{aligned}
\text{PA} \ \vdash\ \psi^\bullet \ &\Leftrightarrow\ \chi\{x := \overline{\ulcorner\chi\urcorner}\} && \text{(def. de } \psi^\bullet\text{)}\\
&\Leftrightarrow\ \exists y\Big(\psi(y) \wedge \phi_h\big(\overline{\ulcorner\chi\urcorner}, y\big)\Big) && \text{(def. de } \chi\text{)}\\
&\Leftrightarrow\ \exists y\Big(\psi(y) \wedge y = \overline{\ulcorner\chi\{x := \overline{\ulcorner\chi\urcorner}\}\urcorner}\Big) && \text{(por } (*)\text{)}\\
&\Leftrightarrow\ \exists y\Big(\psi(y) \wedge y = \overline{\ulcorner\psi^\bullet\urcorner}\Big) && \text{(def. de } \psi^\bullet\text{)}\\
&\Leftrightarrow\ \psi\big(\overline{\ulcorner\psi^\bullet\urcorner}\big) && \text{(tautología)}
\end{aligned}$$

□

4.2. Una fórmula G no demostrable pero verdadera. En la sección 3.6, vimos que el conjunto `Proof` $\subseteq \mathbb{N}^2$ definido por

$$\texttt{Proof} = \{(\ulcorner d\urcorner, \ulcorner\phi\urcorner)\ :\ d \text{ es una demostración de } \phi \text{ en PA}\} \qquad (\subset \mathbb{N}^2)$$

es recursivo primitivo. Por el teorema de representación, este conjunto es representado en PA por una fórmula aritmética $Proof(y,x)$ (con dos variables libres x, y) que satisface:

- PA $\vdash$ $Proof(\ulcorner d\urcorner, \ulcorner\phi\urcorner)$ si d es una demostración de ϕ en PA,
- PA $\vdash$ $\neg Proof(m,n)$ en todos los otros casos.

La demostración histórica del primer teorema de incompletitud se basa sobre la fórmula (cerrada) G construida por el Teorema 4.1 como punto fijo de

$$\text{PA} \ \vdash\ G \ \Leftrightarrow\ \neg Dem(\overline{\ulcorner G\urcorner}),$$

donde $Dem(x)$ es la fórmula (con la única variable libre x) definida por:

$$Dem(x) \ \equiv\ \exists y\, Proof(y,x)\,.$$

Intuitivamente, la fórmula (cerrada) G dice: « no soy demostrable ».

Proposición 4.2 (G no es demostrable)**.** — *Si* PA *es consistente (*PA $\nvdash \bot$*), entonces la fórmula cerrada G no es demostrable en* PA*:* PA $\nvdash G$.

(En otro lado, es claro que si PA es inconsistente, entonces la fórmula G es demostrable en PA, como todas la fórmulas cerradas.)

Demostración. Bajo la hipótesis de consistencia de PA, se supone que G tiene una demostración d en PA. Entonces:

1. $(\ulcorner d\urcorner, \ulcorner G\urcorner) \in$ `Proof` (definición del conjunto `Proof` $\subseteq \mathbb{N}^2$)
2. PA $\vdash Proof(\overline{\ulcorner d\urcorner}, \overline{\ulcorner G\urcorner})$ (por representación)
3. PA $\vdash Dem(\overline{\ulcorner G\urcorner})$ ($\equiv \exists y\, Proof(y, \overline{\ulcorner G\urcorner})$) (regla de introducción de $\exists$, Cuadro 2 p. 9)
4. PA $\vdash \neg G$ (pues PA $\vdash G \Leftrightarrow \neg Dem(\overline{\ulcorner G\urcorner})$)
5. PA $\vdash \bot$ (pues PA $\vdash G$ y PA $\vdash \neg G$)

La última aserción es imposible, pues PA es consistente. Luego PA $\nvdash G$. □

En otro lado, la fórmula G es verdadera en el modelo estándar (suponiendo su existencia):

Proposición 4.3 (G es verdadera)**.** — $\mathbb{N} \models G$.

Demostración. Supongamos la existencia del modelo estándar (véase la discusión de la Sección 1.7). Entonces, la Aritmética de Peano es consistente (Prop. 1.27), y la fórmula G no es demostrable en PA por la Prop. 4.2. Así tenemos que:

1. $(m, \ulcorner G\urcorner) \notin$ `Proof` para todo $m \in \mathbb{N}$ (pues G no es demostrable)
2. PA $\vdash \neg Proof(\overline{m}, \overline{\ulcorner G\urcorner})$ para todo $m \in \mathbb{N}$ (por representación)
3. $\mathbb{N} \models \neg Proof(\overline{m}, \overline{\ulcorner G\urcorner})$ para todo $m \in \mathbb{N}$ (por el Teorema 1.26)

4. $\mathbb{N} \models \forall x\, \neg Proof(x, \overline{\ulcorner G \urcorner})$ (definición de $\mathbb{N} \models \forall x\, \phi(x)$)
5. $\mathbb{N} \models \neg \exists x\, Proof(x, \overline{\ulcorner G \urcorner})$ (tautología)
6. $\mathbb{N} \models \neg Dem(\overline{\ulcorner G \urcorner})$ (definición de la fórmula $Dem(x)$)
7. $\mathbb{N} \models G \Leftrightarrow \neg Dem(\overline{\ulcorner G \urcorner})$ (Teorema 1.26 aplicado a la fórmula $G \Leftrightarrow \neg Dem(\overline{\ulcorner G \urcorner})$)
8. $\mathbb{N} \models G$ (por 6. y 7.) □

Puede parecer extraño que la fórmula cerrada G sea verdadera (en el modelo estándar) sin ser demostrable en PA. Esta paradoja reside en que la aserción PA $\models G$ (la cual presupone la existencia del modelo estándar) sólo se puede establecer en una teoría ambiente infinitaria. En otro lado, la aritmética de Peano (PA) es una teoría finitaria, cuyos objetos (los enteros naturales) son todos finitos. Por lo tanto, el razonamiento infinitario de la Prop. 4.3 no se puede formalizar en PA, lo que explica que la fórmula G no sea demostrable en PA.

4.3. La hipótesis de 1-consistencia. Vimos en la Prop. 4.2 que la fórmula G de Gödel no es demostrable, bajo la hipótesis de consistencia de PA. Desgraciadamente, la hipótesis de consistencia es demasiada débil para establecer que la *negación* de la fórmula G tampoco es demostrable en PA. Para ello, se necesita introducir la hipótesis de 1-*consistencia*[12], que es estrictamente más fuerte que la consistencia (igualmente llamada 0-consistencia):

Definición 4.4 (1-consistencia)**.** — Se dice que PA es 1-*consistente* si para toda función recursiva primitiva $f : \mathbb{N} \to \mathbb{N}$ tal que

$$\mathrm{PA} \ \vdash \ \exists x\, \phi_f(x, 0)$$

(donde la fórmula $\phi_f(x, y)$ es una representación de f en PA), existe $n \in \mathbb{N}$ tal que $f(n) = 0$.

Intuitivamente, la hipótesis de 1-consistencia expresa que toda función recursiva primitiva que tiene *demostrablemente* una raíz (en el sistema formal de PA) tiene *realmente* una raíz (en el sentido de la teoría ambiente). De modo equivalente, esta hipótesis expresa que todo conjunto recursivo primitivo que es demostrablemente no vació (en PA) es realmente no vacío.

Es claro que si PA es inconsistente, entonces no puede ser 1-consistente, pues la fórmula $\exists x\, s(x) = 0$ es demostrable aunque la función sucesor no tenga ninguna raíz. Por contraposición, la 1-consistencia de PA implica la (0-)consistencia de PA.

Ahora se puede establecer la segunda componente de la indecidibilidad de G:

Proposición 4.5 ($\neg G$ no es demostrable)**.** — *Si* PA *es* 1-*consistente, entonces la fórmula cerrada* $\neg G$ *no es demostrable en* PA*:* PA $\nvdash \neg G$.

Demostración. Bajo la hipótesis de 1-consistencia de PA, se supone que

1. PA $\vdash \neg G$. Entonces:
2. PA $\vdash \exists y\, Proof(y, \overline{\ulcorner G \urcorner})$ ($\equiv Dem(\overline{\ulcorner G \urcorner})$) (por la equivalencia $G \Leftrightarrow \neg Dem(\overline{\ulcorner G \urcorner})$)
3. Existe $m \in \mathbb{N}$ tal que $(m, \ulcorner G \urcorner) \in \mathtt{Proof}$ (por 1-consistencia, con el conjunto `Proof`)
4. PA $\vdash G$ (por definición del conjunto `Proof`)
5. PA $\vdash \bot$ (por 1. y 4.)

La última aserción es imposible, pues PA es consistente. Luego PA $\nvdash \neg G$. □

En consecuencia (Prop. 4.2 y 4.5):

Teorema 4.6 (Incompletitud, formulación histórica)**.** — *Si* PA *es* 1-*consistente, entonces la fórmula cerrada G es indecidible en* PA*:* PA $\nvdash G$ *y* PA $\nvdash \neg G$.

[12]En el artículo de Gödel, la 1-consistencia era llamada *ω-consistencia*.

4.4. El truco de Rosser. El defecto de la demostración histórica del primer teorema de incompletitud reside en que utiliza la hipótesis de 1-consistencia, que se puede remplazar por la hipótesis más débil de (0-)consistencia, como demostró Rosser. El truco de Rosser consiste en definir (todavía por el Teorema 4.1) otra fórmula cerrada G' como punto fijo de

$$\mathrm{PA} \ \vdash \ G' \Leftrightarrow \neg \mathit{Dem}'(\overline{\ulcorner G' \urcorner})$$

donde $\mathit{Dem}'(x)$ es la fórmula definida por

$$\mathit{Dem}'(x) \ \equiv \ \exists y\, (\mathit{Proof}(y,x) \wedge (\forall y' < y)\, \neg \mathit{Proof}^{\neg}(y',x))$$

y donde la fórmula $\mathit{Proof}^{\neg}(y,x)$ es una representación (en PA) del conjunto $\mathtt{Proof}^{\neg} \subseteq \mathbb{N}^2$ de las refutaciones (véase Sección 3.7). Intuitivamente, la fórmula $\mathit{Dem}'(x)$ expresa que x es el código de una fórmula cerrada que tiene una demostración de código y, tal que no existe ninguna refutación de dicha fórmula con un código $y' < y$.

Proposición 4.7 (G' no es demostrable)**.** — *Si* PA *es consistente (*$\mathrm{PA} \nvdash \bot$*), entonces la fórmula cerrada* G' *no es demostrable en* PA*:* $\mathrm{PA} \nvdash G'$.

Demostración. Bajo la hipótesis de consistencia de PA, se supone que G' tiene una demostración d en PA. Entonces:

1. $(\ulcorner d \urcorner, \ulcorner G' \urcorner) \in \mathtt{Proof}$ (definición del conjunto $\mathtt{Proof} \subseteq \mathbb{N}^2$)
2. $\mathrm{PA} \vdash \mathit{Proof}(\overline{\ulcorner d \urcorner}, \overline{\ulcorner G' \urcorner})$ (por representación)
3. $(m, \ulcorner G' \urcorner) \notin \mathtt{Proof}^{\neg}$ para todo $m < \ulcorner d \urcorner$, (consistencia de PA)
4. $\mathrm{PA} \vdash \bigwedge_{m < \ulcorner d \urcorner} \neg \mathit{Proof}^{\neg}(\overline{m}, \overline{\ulcorner G' \urcorner})$ (por representación)
5. $\mathrm{PA} \vdash (\forall y' < \overline{\ulcorner d \urcorner})\, \neg \mathit{Proof}^{\neg}(y', \overline{\ulcorner G' \urcorner})$ (Prop. 1.41)
6. $\mathrm{PA} \vdash \mathit{Proof}(\overline{\ulcorner d \urcorner}, \overline{\ulcorner G' \urcorner}) \wedge (\forall y' < \overline{\ulcorner d \urcorner})\, \neg \mathit{Proof}^{\neg}(y', \overline{\ulcorner G' \urcorner})$ (por 2. y 5.)
7. $\mathrm{PA} \vdash \exists y \big(\mathit{Proof}(y, \overline{\ulcorner G' \urcorner}) \wedge (\forall y' < y)\, \neg \mathit{Proof}^{\neg}(y', \overline{\ulcorner G' \urcorner})\big)$ (por la regla $\exists$-intro.)
8. $\mathrm{PA} \vdash \mathit{Dem}'(\overline{\ulcorner G' \urcorner})$ (Def. de la fórmula $\mathit{Dem}'(x)$)
9. $\mathrm{PA} \vdash \neg G'$ (pues $\mathrm{PA} \vdash G' \Leftrightarrow \neg \mathit{Dem}'(\overline{\ulcorner G' \urcorner})$)
10. $\mathrm{PA} \vdash \bot$ (pues $\mathrm{PA} \vdash G'$ y $\mathrm{PA} \vdash \neg G'$)

La última aserción es imposible, pues PA es consistente. Luego $\mathrm{PA} \nvdash G$. □

Proposición 4.8 ($\neg G'$ no es demostrable)**.** — *Si* PA *es consistente (*$\mathrm{PA} \nvdash \bot$*), entonces la fórmula cerrada* $\neg G'$ *no es demostrable en* PA*:* $\mathrm{PA} \nvdash \neg G'$.

Demostración. Bajo la hipótesis de consistencia de PA, se supone que $\neg G'$ tiene una demostración d en PA. Entonces:

1. $(\ulcorner d \urcorner, \ulcorner G' \urcorner) \in \mathtt{Proof}^{\neg}$ (definición del conjunto $\mathtt{Proof}^{\neg} \subseteq \mathbb{N}^2$)
2. $\mathrm{PA} \vdash \mathit{Proof}^{\neg}(\overline{\ulcorner d \urcorner}, \overline{\ulcorner G' \urcorner})$ (por representación)
3. $\mathrm{PA} \vdash (\forall y > \overline{\ulcorner d \urcorner})\, (\exists y' < y)\, \mathit{Proof}^{\neg}(y', \overline{\ulcorner G' \urcorner})$ (razonamiento aritmético elemental)
4. $(m, \ulcorner G \urcorner') \notin \mathtt{Proof}$ para todo $m \leq \ulcorner d \urcorner$ (consistencia de PA)
5. $\mathrm{PA} \vdash \bigwedge_{m \leq \ulcorner d \urcorner} \neg \mathit{Proof}(\overline{m}, \overline{\ulcorner G' \urcorner})$ (por representación)
6. $\mathrm{PA} \vdash (\forall y \leq \overline{\ulcorner d \urcorner})\, \neg \mathit{Proof}(y, \overline{\ulcorner G' \urcorner})$ (Prop. 1.41)
7. $\mathrm{PA} \vdash \forall y\, (\neg \mathit{Proof}(y, \overline{\ulcorner G' \urcorner}) \vee (\exists y' < y)\, \mathit{Proof}^{\neg}(y', \overline{\ulcorner G' \urcorner}))$ (combinando 3. y 6.)
8. $\mathrm{PA} \vdash \neg \mathit{Dem}'(\overline{\ulcorner G' \urcorner})$ (definición del predicado $\mathit{Dem}'(x)$)
9. $\mathrm{PA} \vdash G'$ (pues $\mathrm{PA} \vdash G' \Leftrightarrow \neg \mathit{Dem}'(\overline{\ulcorner G' \urcorner})$)
10. $\mathrm{PA} \vdash \bot$ (pues $\mathrm{PA} \vdash G'$ y $\mathrm{PA} \vdash \neg G'$)

La última aserción es imposible, pues PA es consistente. Luego $\mathrm{PA} \nvdash \neg G'$. □

En consecuencia (Prop. 4.7 y 4.8):

Teorema 4.9 (Primer teorema de incompletitud). — *Si* PA *es consistente, entonces la fórmula cerrada G' es indecidible en* PA*:* PA $\nvdash G'$ *y* PA $\nvdash \neg G'$.

4.5. La noción de demostrabilidad no es recursiva. En la sección 3.7, vimos que los dos conjuntos $\mathtt{Dem}, \mathtt{Dem}^{\neg} \subseteq \mathtt{Form}_0$ definidos por

$$\begin{aligned} \mathtt{Dem} &= \pi_2(\mathtt{Proof}) &= \{\ulcorner\phi\urcorner : \mathrm{PA} \vdash \phi\} && (\subset \mathtt{Form}_0) \\ \mathtt{Dem}^{\neg} &= \pi_2(\mathtt{Proof}^{\neg}) &= \{\ulcorner\phi\urcorner : \mathrm{PA} \vdash \neg\phi\} && (\subset \mathtt{Form}_0) \end{aligned}$$

(cuyos elementos son los códigos de las fórmulas demostrables y refutables) son recursivamente enumerables (Prop. 3.15). Ahora se puede demostrar que:

Proposición 4.10 (La demostrabilidad no es recursiva). — *Si* PA *es consistente, entonces ninguno de los dos conjuntos* $\mathtt{Dem}, \mathtt{Dem}^{\neg} \subseteq \mathtt{Form}_0$ *es recursivo.*

(En otro lado, si PA es inconsistente, es claro que $\mathtt{Dem} = \mathtt{Dem}^{\neg} = \mathtt{Form}_0$, de tal modo que ambos conjuntos son obviamente recursivos.)

Demostración. Supongamos PA es consistente y que el conjunto $\mathtt{Dem} \subseteq \mathbb{N}$ es recursivo. Por el Teorema de representación, este conjunto es representado por una fórmula $Dem''(x)$ dependiendo de la única variable libre x, de tal modo que para toda fórmula cerrada ϕ tengamos:

$$\begin{cases} \mathrm{PA} \vdash Dem''(\overline{\ulcorner\phi\urcorner}) & \text{si} \quad \mathrm{PA} \vdash \phi \\ \mathrm{PA} \vdash \neg Dem''(\overline{\ulcorner\phi\urcorner}) & \text{si} \quad \mathrm{PA} \nvdash \phi \end{cases}$$

Sea G'' la fórmula cerrada definida (por el Teorema 4.1) como punto fijo de $\neg Dem''(x)$:

$$(*) \qquad \mathrm{PA} \ \vdash \ G'' \Leftrightarrow \neg Dem''(\overline{\ulcorner G''\urcorner})$$

Se distinguen los dos casos siguientes, según que PA $\vdash G''$ o PA $\nvdash G''$:

- PA $\vdash G''$. En este caso, tenemos que $\ulcorner G'\urcorner \in \mathtt{Dem}$, de tal modo que PA $\vdash Dem''(\overline{\ulcorner G''\urcorner})$ (por representación), es decir PA $\vdash \neg G''$ (por $(*)$). Es imposible, pues PA es consistente.
- PA $\nvdash G''$. En este caso, tenemos que $\ulcorner G'\urcorner \notin \mathtt{Dem}$, de tal modo que PA $\vdash \neg Dem''(\overline{\ulcorner G''\urcorner})$ (por representación), es decir PA $\vdash G''$ (por $(*)$): contradicción.

Luego, el conjunto $\mathtt{Dem} \subseteq \mathbb{N}$ no es recursivo. Del mismo modo se demuestra que el conjunto $\mathtt{Dem}^{\neg} \subseteq \mathbb{N}$ no es recursivo tampoco. □

Observaciones 4.11. — (1) Intuitivamente, este resultado expresa que no existe ningún algoritmo que permita determinar (en tiempo finito) si una fórmula cerrada ϕ es demostrable o no en PA. En otro lado, vimos que el conjunto $\mathtt{Dem} \subseteq \mathbb{N}$ es recursivamente enumerable, lo que significa que existe un *semi-algoritmo* (el semi-algoritmo de *búsqueda de demostración*) que termina sólo cuando la fórmula cerrada ϕ es demostrable en PA (después de haber encontrado una demostración), pero que no termina cuando la fórmula ϕ no tiene ninguna demostración en PA (porque dicho semi-algoritmo sigue enumerar las demostraciones en bucle infinito).

(2) Es claro que si la aritmética fuera consistente y completa, sería posible determinar en tiempo finito si una fórmula cerrada o no es demostrable: bastaría enumerar todas las demostraciones de PA, hasta encontrar o bien una demostración de ϕ (en cuyo caso PA $\vdash \phi$), o bien una demostración de $\neg\phi$ (en cuyo caso PA $\nvdash \phi$, por consistencia). Por contraposición, la Proposición anterior (Prop. 4.10) implica que PA es incompleta (bajo la hipótesis de consistencia), y como su demostración no depende del primer teorema de incompletitud, esto nos da una demostración alternativa (pero no constructiva) del primer teorema de incompletitud.

(3) Según un punto de vista más filosófico, la Prop. 4.10 (que implica el primer teorema de incompletitud) expresa que la ausencia de demostración —que corresponde filosóficamente

a una *ausencia de información*— no se puede traducir algorítmicamente en una *información negativa* —es decir: una respuesta « no », dada en tiempo finito. Luego, una interpretación filosófica posible del primer teorema de incompletitud es la siguiente:

$$\text{Ausencia de información} \quad \neq \quad \text{Información negativa.}$$

(La ausencia de demostración constituyendo un caso auténtico de ausencia de información, que no se puede transformar en información negativa por ningún procedimiento algorítmico.)

4.6. La noción de verdad no es definible en PA**.** En esta sección se supone la existencia del modelo estándar (Sección 1.7). Bajo esta hipótesis:

Proposición 4.12 (La verdad no es definible)**.** — *No existe ninguna fórmula aritmética $V(x)$ (dependiendo de la variable x) tal que para toda fórmula cerrada ϕ:*

$$\begin{cases} \mathrm{PA} \vdash V(\overline{\ulcorner\phi\urcorner}) & si \quad \mathbb{N} \models \phi \\ \mathrm{PA} \vdash \neg V(\overline{\ulcorner\phi\urcorner}) & si \quad \mathbb{N} \not\models \phi \end{cases}$$

Demostración. Supongamos que tal fórmula $V(x)$ existe, y consideremos la fórmula cerrada H construida (por el Teorema 4.1) como punto fijo de

$$(*) \qquad \mathrm{PA} \ \vdash \ H \ \Leftrightarrow \ \neg V(\overline{\ulcorner H\urcorner})$$

Se distinguen los dos casos siguientes, según que $\mathbb{N} \models H$ o $\mathbb{N} \not\models H$:

- $\mathbb{N} \models H$. En este caso, tenemos que $\mathrm{PA} \vdash V(\overline{\ulcorner H\urcorner})$, entonces $\mathrm{PA} \vdash \neg H$ (por $(*)$), lo que implica (por el Teorema 1.26) que $\mathbb{N} \models \neg H$, es decir: $\mathbb{N} \not\models H$: contradicción.
- $\mathbb{N} \not\models H$. En este caso, tenemos que $\mathrm{PA} \vdash \neg V(\overline{\ulcorner H\urcorner})$, entonces $\mathrm{PA} \vdash H$ (por $(*)$), lo que implica (por el Teorema 1.26) que $\mathbb{N} \models H$: contradicción.

Los dos casos llevan a una contradicción, luego tal fórmula $V(x)$ no puede existir. □

Observación 4.13 (Paradoja de Richard)**.** — El resultado anterior (debido a Tarski) es fuertemente relacionado con la paradoja de Richard, que parte de la siguiente observación:

Para todo entero $n \in \mathbb{N}$, la cantidad de frases que se pueden formar (en español) con menos de n palabras es finita. Luego, las frases con menos de n palabras sólo pueden definir (es decir: caracterizar) una cantidad finita de enteros naturales, de tal modo que para todo $n \in \mathbb{N}$, existe una cantidad infinita de enteros naturales que no se pueden definir utilizando menos de n palabras. La paradoja de Richard se obtiene considerando el entero natural definido como

El entero natural más pequeño que no se puede definir con menos de dieciséis palabras

y observando que tal definición sólo utiliza... quince palabras.

Por supuesto, tal paradoja se resuelve observando que el predicado de *definibilidad*

« m es definible con menos de n palabras »

(obviamente relacionado con el predicado de *verdad*) no se puede definir adentro de un lenguaje formal tal que PA —aunque se pueda definir en la teoría ambiente (remplazando el "número de palabras" por el "número de símbolos"). De tal modo que la "definición" sobre la cual se basa la paradoja, de hecho, no es una definición bien formada.

En la sección 3.7, definimos los dos conjuntos

$$\begin{aligned} \mathtt{True} &= \{\ulcorner\phi\urcorner \ : \ \phi \text{ cerrada y } \mathbb{N} \models \phi\} && (\subset \mathtt{Form}_0) \\ \mathtt{False} &= \{\ulcorner\phi\urcorner \ : \ \phi \text{ cerrada y } \mathbb{N} \not\models \phi\} && (\subset \mathtt{Form}_0) \end{aligned}$$

que forman una partición del conjunto $\mathtt{Form}_0$ de los códigos de fórmulas cerradas:

$$\mathtt{True} \uplus \mathtt{False} = \mathtt{Form}_0\,.$$

Corolario 4.14. — *Los conjuntos* $\mathtt{True}, \mathtt{False} \subseteq \mathbb{N}$ *no son recursivamente enumerables.*

Demostración. Por definición de la relación de satisfacción $\mathbb{N} \models \phi$, es claro que

$$\mathtt{True} = \{n \in \mathbb{N} : \mathtt{fneg}(n) \in \mathtt{False}\} \qquad \text{y} \qquad \mathtt{False} = \{n \in \mathbb{N} : \mathtt{fneg}(n) \in \mathtt{True}\}$$

de tal modo que el conjunto $\mathtt{True}$ es recursivamente enumerable si y sólo si el conjunto $\mathtt{False}$ es recursivamente enumerable. Por contradicción, supongamos que ambos conjuntos $\mathtt{True}, \mathtt{False} \subseteq \mathbb{N}$ son recursivamente enumerables. Como estos dos conjuntos forman una partición del conjunto $\mathtt{Form}_0 \subseteq \mathbb{N}$, tenemos que

$$\mathtt{True}^c = (\mathtt{Form}_0 - \mathtt{False})^c = \underbrace{(\mathtt{Form}_0)^c}_{\text{rec. prim.}} \cup \underbrace{\mathtt{False}}_{\text{rec. enum.}},$$

lo que implica que el complementario $\mathtt{True}^c = \mathbb{N} - \mathtt{True}$ del conjunto $\mathtt{True} \subseteq \mathbb{N}$ también es recursivamente enumerable. Entonces, el conjunto $\mathtt{True}$ es recursivo (Prop. 2.10 (4) p. 22) y luego representable en PA (Corolario 2.15 p. 23). Pero esto implica trivialmente que la verdad es definible en PA, lo que contradice la Prop. 4.12. Luego, la hipótesis era absurda, y ninguno de los dos conjuntos $\mathtt{True}, \mathtt{False} \subseteq \mathbb{N}$ es recursivamente enumerable. □

4.7. Conclusión. La situación relativa de los conjuntos $\mathtt{Proof}, \mathtt{Proof}^{\neg} \subseteq \mathbb{N}^2$ (recursivos primitivos), $\mathtt{Dem}, \mathtt{Dem}^{\neg} \subseteq \mathtt{Form}_0$ (recursivamente enumerables y no recursivos) y $\mathtt{True}, \mathtt{False} \subseteq \mathtt{Form}_0$ (no recursivamente enumerables) es resumida en la Fig. 1 abajo.

Demostraciones		**Demostrabilidad**		**Verdad**
(Conjuntos recursivos primitivos)		(Conjuntos recursivamente enumerables)		(Conjuntos no recursivamente enumerables)
$\mathtt{Proof} \subset \mathbb{N}^2$	$\xrightarrow{\pi_2}$	$\mathtt{Dem} = \pi_2(\mathtt{Proof})$	$\subsetneq$	$\mathtt{True} \subset \mathbb{N}$
$\mathtt{Proof}^{\neg} \subset \mathbb{N}^2$	$\xrightarrow{\pi_2}$	$\mathtt{Dem}^{\neg} = \pi_2(\mathtt{Proof}^{\neg})$	$\subsetneq$	$\mathtt{False} \subset \mathbb{N}$
		$\mathtt{Dem} \uplus \mathtt{Dem}^{\neg} \subsetneq \mathtt{Form}_0$ (incompletitud sintáctica)		$\mathtt{True} \uplus \mathtt{False} = \mathtt{Form}_0$ (completitud semántica)

FIGURA 1. Situación relativa de los conjuntos de códigos notables

5. El segundo teorema de incompletitud

5.1. El argumento de Gödel. Vimos (Sección 4.2) que la demostración histórica del primer teorema de incompletitud se apoya en la construcción de una fórmula aritmética cerrada G tal que $\mathrm{PA} \vdash G \Leftrightarrow \neg\mathtt{Dem}(\overline{\ulcorner G \urcorner})$, de tal modo que:

$$\textit{Si } \mathrm{PA} \textit{ es consistente, entonces } \mathrm{PA} \nvdash G \qquad \text{(Prop. 4.2)}$$

Analizando la estructura de la demostración de este resultado, Gödel observa que

(1) La implicación anterior, formulada en el meta-lenguaje, se puede internalizar adentro del lenguaje de PA a través de la codificación de las fórmulas y de las demostraciones con números naturales. Naturalmente, las aserciones « PA es consistente » y « G no es demostrable » son remplazadas por la fórmulas aritméticas $\mathit{Cons}\mathrm{PA} \equiv \neg\mathit{Dem}(\overline{\ulcorner \bot \urcorner})$ y $\neg\mathit{Dem}(\overline{\ulcorner G \urcorner})$, y el enunciado de la Prop. 4.2 se convierte en la fórmula aritmética

$$\mathit{Cons}\mathrm{PA} \Rightarrow \neg\mathit{Dem}(\overline{\ulcorner G \urcorner}).$$

(2) La (meta-)demostración de la implicación « *Si* PA *es consistente, entonces* PA $\nvdash G$ » sólo utiliza principios de razonamiento finitarios, de tal modo que se puede internalizar en su totalidad adentro de PA, convirtiéndose en una demostración *formal* de la fórmula aritmética $\mathit{Cons}\mathrm{PA} \Rightarrow \neg \mathit{Dem}(\overline{\ulcorner G \urcorner})$. Así:

$$\mathrm{PA} \ \vdash \ \mathit{Cons}\mathrm{PA} \Rightarrow \neg \mathit{Dem}(\overline{\ulcorner G \urcorner})\,.$$

Y como $\mathrm{PA} \vdash G \Leftrightarrow \neg \mathit{Dem}(\overline{\ulcorner G \urcorner})$, se obtiene que

$$\mathrm{PA} \ \vdash \ \mathit{Cons}\mathrm{PA} \Rightarrow G\,.$$

(3) Pero como PA $\nvdash G$ (suponiendo PA consistente), se deduce que PA $\nvdash \mathit{Cons}\mathrm{PA}$ (bajo la misma hipótesis). Dicho de otra manera, si PA es consistente, entonces PA no puede demostrar la fórmula $\mathit{Cons}\mathrm{PA} \equiv \neg \mathit{Dem}(\overline{\ulcorner \bot \urcorner})$ que expresa su propia consistencia.

Internalización de la demostración de la Prop. 4.2 Obviamente, la parte más técnica de la demostración esbozada anteriormente (que otramente no tiene ninguna dificultad conceptual) es la construcción efectiva de la demostración (formal) de la implicación $\mathit{Cons}\mathrm{PA} \Rightarrow \neg \mathit{Dem}(\overline{\ulcorner G \urcorner})$ en PA, internalizando la demostración de la Prop. 4.2 adentro de PA.

Para ello, se puede observar que no se necesita internalizar el teorema de representación (Teorema 2.14), pues sólo sirve para construir algunas fórmulas cruciales en el enunciado del primer teorema de incompletitud. Se podrá tomar estas fórmulas tal cuales, asegurándose que verifican (en PA) las propiedades deseadas. Por ejemplo, habrá que demostrar (entre otras cosas) que la sustitución es compatible con la relación de α-equivalencia, es decir, en PA:

$$\begin{aligned}&\forall x\,\forall x'\,\forall v\,\forall y\,\forall z\,\forall z'\\ &\quad(\mathit{Form}(x) \wedge \mathit{Form}(x') \wedge \mathit{Term}(y) \wedge\\ &\quad\ \mathit{Subst}(x,v,y,z) \wedge \mathit{Subst}(x',v,y,z') \wedge \mathit{Equiv}(x,x') \ \Rightarrow \ \mathit{Equiv}(z,z'))\end{aligned}$$

(donde $\mathit{Form}(x)$ expresa que x es un código de fórmula, $\mathit{Term}(y)$ que y es un código de término, $\mathit{Equiv}(y,y')$ que las fórmulas de códigos y, y' son α-equivalentes, etc.)

Sin embargo, la demostración formal de la implicación $\mathit{Cons}\mathrm{PA} \Rightarrow \neg \mathit{Dem}(\overline{\ulcorner G \urcorner})$ en PA sigue siendo un ejercicio técnicamente difícil, y es la razón por la cual puede ser útil considerar el problema con una forma un poquito más abstracta, afín de extraer los ingredientes cruciales de tal demostración. Esto nos conduce a interesarnos en las condiciones de derivabilidad introducidas por Hilbert y Bernays.

5.2. Las condiciones de derivabilidad de Hilbert-Bernays. Sea $\mathscr{T}$ una teoría de primer orden cuyo lenguaje contiene el lenguaje de la Aritmética, dada con:

- una codificación $\phi \mapsto \ulcorner \phi \urcorner$ de las fórmulas cerradas del lenguaje de $\mathscr{T}$ en $\mathbb{N}$;
- una fórmula $\mathit{Dem}(x)$ que sólo depende de la variable x.

Se dice que la teoría $\mathscr{T}$ satisface las *condiciones de derivabilidad de Hilbert-Bernays* si para todas las fórmulas cerradas ϕ, ψ del lenguaje de $\mathscr{T}$ se cumplen las tres condiciones siguientes:

(HB1) Si $\mathscr{T} \vdash \phi$, entonces $\mathscr{T} \vdash \mathit{Dem}(\overline{\ulcorner \phi \urcorner})$

(HB2) $\mathscr{T} \ \vdash \ \mathit{Dem}(\overline{\ulcorner \phi \Rightarrow \psi \urcorner}) \Rightarrow \big(\mathit{Dem}(\overline{\ulcorner \phi \urcorner}) \Rightarrow \mathit{Dem}(\overline{\ulcorner \psi \urcorner})\big)$

(HB3) $\mathscr{T} \ \vdash \ \mathit{Dem}(\overline{\ulcorner \phi \urcorner}) \Rightarrow \mathit{Dem}\Big(\overline{\ulcorner \mathit{Dem}(\overline{\ulcorner \phi \urcorner}) \urcorner}\Big)$

Para manipular más fácilmente estas tres condiciones, se utiliza la notación

$$\Box\phi \ \equiv \ \mathit{Dem}(\overline{\ulcorner \phi \urcorner}) \qquad\qquad (\text{« } \phi \text{ es demostrable »})$$

para cada fórmula cerrada ϕ del lenguaje de $\mathscr{T}$, gracias a la cual las tres condiciones de derivabilidad de Hilbert-Bernays se escriben más simplemente[13]:

$$\begin{array}{ll} \text{(HB1)} & \text{Si} \quad \mathscr{T} \vdash \phi, \quad \text{entonces} \quad \mathscr{T} \vdash \Box\phi \\ \text{(HB2)} & \mathscr{T} \vdash \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi) \\ \text{(HB3)} & \mathscr{T} \vdash \Box\phi \Rightarrow \Box\Box\phi \end{array}$$

(donde ϕ, ψ son fórmulas cerradas cualesquiera del lenguaje de $\mathscr{T}$).

La condición (HB1) dice que si ϕ es demostrable en $\mathscr{T}$, entonces la fórmula $\Box\phi$ también es demostrable en $\mathscr{T}$[14]. Es la condición que expresa que la fórmula $\mathit{Dem}(x)$ es la internalización de la noción de demostrabilidad (en el sentido de $\mathscr{T}$) adentro de $\mathscr{T}$. En PA, ya vimos que la fórmula $\mathit{Dem}(x) \equiv \exists y\, \mathit{Proof}(y, x)$ (Sección 4.2) satisface esta condición.

La condición (HB2) expresa en la teoría $\mathscr{T}$ que la demostrabilidad de una implicación y la demostrabilidad de su miembro izquierdo implican la demostrabilidad de su miembro derecho. Esta condición es nada más que la regla de *modus ponens*

$$\text{Si} \quad \mathscr{T} \vdash \phi \Rightarrow \psi \quad \text{y} \quad \mathscr{T} \vdash \phi, \quad \text{entonces} \quad \mathscr{T} \vdash \psi$$

internalizada adentro de la teoría $\mathscr{T}$ gracias a la modalidad $\Box\phi$:

$$\mathscr{T} \vdash \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi).$$

La condición (HB3) sigue el espíritu de las dos otras condiciones, y es nada más que la condición (HB1)

$$\text{Si} \quad \mathscr{T} \vdash \phi, \quad \text{entonces} \quad \mathscr{T} \vdash \Box\phi$$

internalizada adentro de la teoría $\mathscr{T}$:

$$\mathscr{T} \vdash \Box\phi \Rightarrow \Box\Box\phi.$$

Veremos en la Sección 5.3 que las condiciones de Hilbert-Bernays capturan las propiedades necesarias para demostrar el segundo teorema de incompletitud. Para utilizar estas condiciones en la Aritmética de Peano, se necesita verificar que:

Proposición 5.1. — *La Aritmética de Peano (*PA*) satisface las condiciones de Hilbert-Bernays, con la fórmula $\mathit{Dem}(x) \equiv \exists y\, \mathit{Proof}(y, x)$ definida en la sección 4.2.*

Demostración. (HB1) Ya vimos que la primera condición se cumple en PA, por definición de la fórmula $\mathit{Dem}(x) \equiv \exists y\, \mathit{Proof}(y, x)$ (véase Sección 4.2).

(HB2) La segunda condición se deduce de que:

$$\mathrm{PA} \ \vdash \ \forall x\, \forall y\, \forall z\, (\mathit{Imp}(x, y, z) \land \mathit{Dem}(z) \land \mathit{Dem}(x) \Rightarrow \mathit{Dem}(y))\,,$$

donde $\mathit{Imp}(x, y, z)$ es la fórmula que representa en PA la función de codificación de la implicación $\mathtt{imp} : \mathbb{N}^2 \to \mathbb{N}$, definida por $\mathtt{imp}(n, m) = \langle 2, \langle n, m\rangle\rangle$ para todo $(n, m) \in \mathbb{N}^2$ (véase Sección 3.6). No se construirá aquí una demostración formal (en PA) de la fórmula anterior, que es un ejercicio técnico de escaso interés conceptual.

(HB3) La tercera condición se deduce de que:

$$\mathrm{PA} \ \vdash \ \forall y\, \forall z\, (\mathit{FSubstNum}(\overline{\ulcorner \mathit{Dem}(x_0) \urcorner}, \overline{\# x_0}, y, z) \land \mathit{Dem}(y) \Rightarrow \mathit{Dem}(z))\,,$$

donde $\mathit{FSubstNum}(x, v, y, z)$ es una fórmula aritmética que representa en PA la función recursiva primitiva $\mathtt{fsubstnum} : \mathbb{N}^3 \to \mathbb{N}$ definida por $\mathtt{fsubstnum}(n, v, m) = \mathtt{fsubst}(n, v, \mathtt{pnum}(m))$

[13]Se reconocen tres principios estándar de la lógica modal, es decir: la regla de necesitación (HB1), la ley de distributividad (HB2) y la ley de transitividad, restringidas aquí a las fórmulas cerradas de $\mathscr{T}$.

[14]¡Cuidado! Esta condición no dice que la implicación $\phi \Rightarrow \Box\phi$ es demostrable en $\mathscr{T}$ (lo que es más fuerte).

para todo $(n, v, m) \in \mathbb{N}^3$. De nuevo, la construcción de una demostración formal (en PA) de la fórmula anterior es un ejercicio técnico de escaso interés conceptual que no haremos aquí. □

Observación 5.2. — El esbozo de demostración anterior ya permite extraer las dos fórmulas cerradas que se necesita demostrar formalmente en PA, es decir:

1. $\forall x \, \forall y \, \forall z \, (\mathit{Imp}(x, y, z) \wedge \mathit{Dem}(z) \wedge \mathit{Dem}(x) \Rightarrow \mathit{Dem}(y))$
2. $\forall y \, \forall z \, (\mathit{FSubstNum}(\overline{\ulcorner \mathit{Dem}(x_0) \urcorner}, \overline{\# x_0}, y, z) \wedge \mathit{Dem}(y) \Rightarrow \mathit{Dem}(z))$

Por supuesto, la demostración formal de estas dos fórmulas (en la cual reside toda la dificultad técnica del segundo teorema de incompletitud) dependerá fuertemente de la fórmula $\mathit{Proof}(y, z)$ elegida para representar el conjunto recursivo primitivo `Proof` $\subseteq \mathbb{N}^2$.

5.3. El teorema de Löb. Se comienza por demostrar una versión abstracta del segundo teorema de incompletitud, debida a Löb:

Teorema 5.3 (Löb). — *Sea $\mathscr{T}$ una teoría de primer orden en la cual todas las funciones recursivas son representables, y que satisface las condiciones de Hilbert-Bernays. Dada una fórmula ϕ cerrada del lenguaje de $\mathscr{T}$, si $\mathscr{T} \vdash \mathit{Dem}(\overline{\ulcorner \phi \urcorner}) \Rightarrow \phi$, entonces $\mathscr{T} \vdash \phi$.*

Demostración. Sea ψ una fórmula cerrada tal que $\mathscr{T} \vdash \psi \Leftrightarrow (\mathit{Dem}(\overline{\ulcorner \psi \urcorner}) \Rightarrow \phi)$, cuya existencia es dada por el Teorema del punto fijo (Teorema 4.1) adaptado a la teoría $\mathscr{T}$[15] y aplicado a la fórmula $\psi(y) \equiv (\mathit{Dem}(y) \Rightarrow \phi)$. Usando la notación $\Box\phi \equiv \mathit{Dem}(\overline{\ulcorner \phi \urcorner})$, se verifica que:

1. $\mathscr{T} \vdash \Box\phi \Rightarrow \phi$ — por hipótesis
2. $\mathscr{T} \vdash \psi \Leftrightarrow (\Box\psi \Rightarrow \phi)$ — por construcción de ψ
3. $\mathscr{T} \vdash \psi \Rightarrow (\Box\psi \Rightarrow \phi)$ — por 2.
4. $\mathscr{T} \vdash \Box(\psi \Rightarrow (\Box\psi \Rightarrow \phi))$ — por 3. y (HB1)
5. $\mathscr{T} \vdash \Box\psi \Rightarrow \Box(\Box\psi \Rightarrow \phi)$ — por 4. y (HB2)
6. $\mathscr{T} \vdash \Box\psi \Rightarrow (\Box\Box\psi \Rightarrow \Box\phi)$ — por 5. y (HB2)
7. $\mathscr{T} \vdash \Box\psi \Rightarrow \Box\Box\psi$ — por (HB3)
8. $\mathscr{T} \vdash \Box\psi \Rightarrow \Box\phi$ — por 6. y 7.
9. $\mathscr{T} \vdash \Box\psi \Rightarrow \phi$ — por 1. y 8.
10. $\mathscr{T} \vdash \psi$ — por 2. y 9.
11. $\mathscr{T} \vdash \Box\psi$ — por 10. y (HB1)
12. $\mathscr{T} \vdash \phi$ — por 9. y 11. □

Combinando este resultado con la Prop. 5.1, se deduce que:

Teorema 5.4 (Segundo teorema de incompletitud). — *Si* PA *es consistente, entonces la fórmula*

$$\mathit{Cons}\mathrm{PA} \;\equiv\; \neg \mathit{Dem}(\overline{\ulcorner \bot \urcorner}) \;\equiv\; \neg \exists y \, \mathit{Proof}(y, \overline{\ulcorner \bot \urcorner}) \qquad (\text{« PA } \textit{es consistente} \text{ »})$$

no es demostrable en PA.

Demostración. Razonando por contraposición, se supone que PA $\vdash$ *Cons*PA, es decir que PA $\vdash$ $\mathit{Dem}(\overline{\ulcorner \bot \urcorner}) \Rightarrow \bot$. Por el teorema de Löb (con la fórmula $\phi \equiv \bot$), se deduce que PA $\vdash \bot$. □

6. Generalización del los teoremas de incompletitud

En las Secciones 4 y 5 demostramos los dos teoremas de incompletitud de Gödel en el caso particular (pero esencial) de la Aritmética de Peano (PA). Sin embargo, los dos teoremas se aplican a una clase mucho más grande de teorías, y en esta última sección, proponemos formular los dos teoremas en un marco más general.

[15]Se verifica fácilmente que el teorema del punto fijo se puede establecer en cualquier teoría $\mathscr{T}$ en la cual todas las funciones recursivas son representables.

6.1. Lenguaje y teorías de primer orden. Los teoremas de incompletitud de Gödel expresan propiedades de sistemas formales llamados *teorías*. Hay muchas nociones de teoría en lógica, pero la gran mayoría de ellas[16] se pueden reducir a la noción fundamental de teoría de primer orden, la cual es basada en la noción fundamental de lenguaje de primer orden:

Definición 6.1 (Lenguaje de primer orden)**.** — Un *lenguaje de primer orden* es un lenguaje de *términos* y de *fórmulas* definido a partir de un *vocabulario* que contiene:

- *símbolos de función* (notación: f, g, h, etc.),
- *símbolos de predicado* (notación: p, q, r, etc.);

cada símbolo de función o de predicado siendo dado con un entero natural, llamado su *aridad*. Los símbolos de función de aridad 0 también se llaman *símbolos de constante*. Dado un vocabulario, se definen los *términos* y las *fórmulas* del lenguaje correspondiente como las expresiones sintácticas construidas por aplicación finita de las siguientes reglas:

- Reglas de construcción de los términos:
  (1) Si x es una variable, entonces x es un término.
  (2) Si f es un símbolo de función de aridad k, y si $t_1, \ldots, t_k$ son términos, entonces $f(t_1, \ldots, t_k)$ es un término (cuando $k = 0$, este término se escribe f antes de $f()$).
- Reglas de construcción de las fórmulas:
  (1) $\bot$ es una fórmula.
  (2) Si t, u son términos, entonces $t = u$ es una fórmula.
  (3) Si p es un símbolo de predicado de aridad k, y si $t_1, \ldots, t_k$ son términos, entonces $p(t_1, \ldots, t_k)$ es una fórmula (cuando $k = 0$, esta fórmula se escribe p antes de $p()$).
  (4) Si ϕ, ψ son fórmulas, entonces $\phi \Rightarrow \psi$ es una fórmula.
  (5) Si x es una variable y ϕ una fórmula, entonces $\forall x\, \phi$ es una fórmula.

Se utilizan las abreviaturas usuales para definir las otras construcciones lógicas: $\neg\phi \equiv \phi \Rightarrow \bot$, $\top \equiv \neg\bot$, $\phi \wedge \psi \equiv \neg(\phi \Rightarrow \psi \Rightarrow \bot)$, $\phi \vee \psi \equiv \neg\phi \Rightarrow \neg\psi \Rightarrow \bot$, $\exists x\, \phi \equiv \neg\forall x \neg\phi$, etc.

Definición 6.2 (Teorías de primer orden)**.** — Una *teoría de primer orden* $\mathscr{T}$ es definida por:

- Un lenguaje de primer orden, llamado *el lenguaje de* $\mathscr{T}$.
- Un conjunto de fórmulas cerradas del lenguaje de $\mathscr{T}$, llamadas los *axiomas de* $\mathscr{T}$.

Se dice que una fórmula cerrada ϕ del lenguaje de $\mathscr{T}$ es un *teorema* de $\mathscr{T}$ (notación: $\mathscr{T} \vdash \phi$) cuando existe una lista finita Δ de axiomas de la teoría $\mathscr{T}$ tal que el secuente $\Delta \vdash \phi$ es derivable en el sistema de deducción de la Sección 1.3 (adaptado al lenguaje de $\mathscr{T}$).

Se dice que una teoría $\mathscr{T}$ es *finita* cuando su vocabulario y su conjunto de axiomas son finitos. Dado dos teorías $\mathscr{T}$, $\mathscr{T}'$, se dice que $\mathscr{T}$ es una *subteoría* de $\mathscr{T}'$ (notación: $\mathscr{T} \subseteq \mathscr{T}'$), o que $\mathscr{T}'$ es una *extensión* de $\mathscr{T}$ (notación: $\mathscr{T}' \supseteq \mathscr{T}$), cuando el lenguaje de $\mathscr{T}$ está incluido en el lenguaje de $\mathscr{T}'$, y cuando todo teorema de $\mathscr{T}$ también es un teorema de $\mathscr{T}'$. (Se verifica fácilmente que la última condición se cumple cuando todo axioma de $\mathscr{T}$ es un teorema de $\mathscr{T}'$.)

Definición 6.3 (Teorías recursivas)**.** — Se dice que una teoría $\mathscr{T}$ es *recursiva* cuando:

(1) El lenguaje (de términos y de fórmulas) de $\mathscr{T}$ es numerable.
(2) La teoría $\mathscr{T}$ es dada con dos funciones de codificación $t \mapsto \ulcorner t \urcorner$ y $\phi \mapsto \ulcorner \phi \urcorner$ de los términos y de las fórmulas en el conjunto $\mathbb{N}$, tales que todas las operaciones de construcción de términos y fórmulas corresponden a funciones recursivas a través de esta codificación. Esto implica que las otras operaciones sintácticas —por ejemplo las dos operaciones de sustitución $t\{x := u\}$ (en los términos) y $\phi\{x := u\}$ (en las fórmulas)— también corresponden a funciones recursivas a través de esta codificación.

[16] Incluso las teorías dichas de segundo orden o de orden superior.

(3) El conjunto $\texttt{Axiom}_{\mathscr{T}} \subseteq \mathbb{N}$ de los códigos de axiomas de $\mathscr{T}$ es recursivo. Esto implica que el conjunto de los códigos de demostraciones de $\mathscr{T}$ también es recursivo.

(Se verifica fácilmente que todas las teorías finitas son recursivas.)

La propiedad de recursividad es esencial para asegurar la existencia de un algoritmo que permita decir si una demostración es correcta o no, y es la razón por la cual todas las teorías usuales en matemática son recursivas: PA (Aritmética de Peano), ZF (teoría de conjuntos de Zermelo-Fraenkel), ZFC (ZF extendida con el axioma de la elección), etc. Sin la propiedad de recursividad, el razonamiento matemático pierde su carácter *objetivo* —es decir, aquí: la posibilidad de ser verificado por un *objeto* (una computadora, por ejemplo).

En otro lado, se pueden considerar teorías de primer orden no recursivas (incluso teorías cuyo lenguaje no es numerable); tales teorías desempeñan un papel importante en la teoría de modelos, pero no tienen ningún interés práctico.

6.2. La Aritmética de Robinson (Q). Vimos que la demostración de los teoremas de incompletitud se basa fuertemente en la posibilidad de representar las funciones recursivas (y los conjuntos recursivos) en PA, por fórmulas que describen el cálculo subyacente.

Sin embargo, se puede observar que la demostración del teorema de representación (Teorema 2.14) y más generalmente la demostración del primer teorema de incompletitud nunca utilizan el principio de inducción de PA —aunque utilicen fuertemente el razonamiento por inducción en la teoría ambiente, que es cosa muy diferente. Por lo tanto, el teorema de representación y el primer teorema de incompletitud se cumplen en teorías mucho más débiles que la Aritmética de Peano, tales que la Aritmética de Robinson.

Formalmente, se llama la *Aritmética de Robinson* (notación: Q) la teoría de primer orden cuyo lenguaje es el lenguaje de PA y cuyos axiomas son los axiomas $(A_1) - (A_6)$ de PA (Sección 1.4) más el nuevo axioma

$$(A_7^-) \quad \forall x\,(x = 0 \lor \exists y\,(x = s(y)))$$

Dicho de otra manera, la Aritmética de Robinson es la teoría obtenida a partir de la Aritmética de Peano eliminando todos los axiomas de inducción, y remplazándolos por el único axioma (A_7^-) que expresa que todo entero natural es o bien cero, o bien el sucesor de otro entero natural.

Se verifica fácilmente que el axioma (A_7^-) es tautológicamente equivalente al axioma de inducción con la fórmula $\phi(x) \equiv x = 0 \lor \exists y\,(x = s(y))$, de tal modo que PA $\vdash (A_7)^-$. Luego:

Proposición 6.4. — Q *es una subteoría finita de* PA *(con el mismo lenguaje).*

Por supuesto, Q es mucho más débil que PA, en razón de la ausencia de los axiomas de inducción (salvo uno). Por ejemplo, ni se puede demostrar en Q que la adición es conmutativa:

$$\mathrm{Q} \nvdash \forall x \forall y\,(x + y = y + x)\,.$$

Sin embargo, la Aritmética de Robinson Q es bastante expresiva para demostrar/refutar todas las (des)igualdades verdaderas/falsas, y satisface más generalmente todas las propiedades de completitud computacional presentadas en la Sección 1.8:

Proposición 6.5. — *Todos los resultados presentados en la Sección 1.8 ('Propiedades de completitud de* PA*') se cumplen en* Q*, y más generalmente en cualquier extensión de* Q.

(En particular, el axioma (A_7^-) sirve para establecer las propiedades de desarrollo del orden.)

De tal modo que:

Teorema 6.6 (Representación). — *Todas las funciones recursivas (y todos los conjuntos recursivos) son representables en* Q*, y más generalmente en cualquier extensión de* Q.

6.3. Inmersión de teorías. La noción de subteoría definida en la Sección 6.1 es demasiada débil para capturar la intuición que una teoría « contiene la Aritmética de Peano (o la de Robinson) ». (Por ejemplo, la teoría de conjuntos ZF no es formalmente una extensión de PA, aunque contenga intuitivamente la Aritmética de Peano a través de la representación de los enteros naturales por los ordinales finitos.) Para dar un contenido formal a esta noción intuitiva, se necesita definir la noción mucho más general de *inmersión de teorías*:

Definición 6.7 (Inmersión de teorías). — Sea $\mathscr{T}$, $\mathscr{T}'$ dos teorías de primer orden. Se llama una *inmersión* de $\mathscr{T}$ adentro de $\mathscr{T}'$ cualquier par $(\mathsf{T}, D(x))$ donde:

- $D(x)$ es una fórmula del lenguaje de $\mathscr{T}'$ que sólo depende de la variable x, llamada la *imagen* de la inmersión $(\mathsf{T}, D(x))$ adentro de la teoría $\mathscr{T}$;
- T es una función que asocia a cada fórmula ϕ del lenguaje de $\mathscr{T}$ una fórmula $\mathsf{T}(\phi)$ del lenguaje de $\mathscr{T}'$, tal que para todas las fórmulas ϕ, ψ del lenguaje de $\mathscr{T}$:

$$\begin{array}{ll} (1) & FV(\mathsf{T}(\phi)) \subseteq FV(\phi) \\ (2) & \mathscr{T}' \vdash \mathsf{T}(\bot) \Leftrightarrow \bot \\ (3) & \mathscr{T}' \vdash \mathsf{T}(\phi \Rightarrow \psi) \Leftrightarrow (\mathsf{T}(\phi) \Rightarrow \mathsf{T}(\psi)) \\ (4) & \mathscr{T}' \vdash \mathsf{T}(\forall x\, \phi) \Leftrightarrow \forall x\, (D(x) \Rightarrow \mathsf{T}(\phi)) \end{array}$$

  (Los ítem (3) y (4) se generalizan inmediatamente a las conectivas $\bot$, $\neg$, $\wedge$, $\vee$, $\Leftrightarrow$ y al cuantificador existencial $\exists$.)

Además, cuando ambas teorías $\mathscr{T}$, $\mathscr{T}'$ son recursivas, se dice que la inmersión $(\mathsf{T}, D(x))$ de $\mathscr{T}$ adentro de $\mathscr{T}'$ es *recursiva* cuando existe una función recursiva $h : \mathbb{N} \to \mathbb{N}$ tal que

$$\ulcorner \mathsf{T}(\phi) \urcorner = h(\ulcorner \phi \urcorner)$$

para toda fórmula ϕ del lenguaje de $\mathscr{T}$.

Ahora, se puede decir que una teoría recursiva $\mathscr{T}$ *contiene* PA *(o* Q*)* cuando existe una inmersión recursiva de PA (o de Q) adentro de la teoría $\mathscr{T}$. Esto nos permite formular los dos teoremas de incompletitud de Gödel en el marco más general:

Teorema 6.8 (Primer teorema de incompletitud). — *Sea $\mathscr{T}$ una teoría de primer orden, recursiva y conteniendo la Aritmética de Robinson* Q. *Si $\mathscr{T}$ es consistente, entonces $\mathscr{T}$ es incompleta en el sentido en que existe en el lenguaje de $\mathscr{T}$ una fórmula cerrada G que no se puede ni demostrar ni refutar en la teoría $\mathscr{T}$: $\quad \mathscr{T} \nvdash G \quad$ y $\quad \mathscr{T} \nvdash \neg G$.*

Teorema 6.9 (Segundo teorema de incompletitud). — *Sea $\mathscr{T}$ una teoría de primer orden, recursiva y conteniendo la Aritmética de Peano* PA. *Si $\mathscr{T}$ es consistente, entonces la fórmula cerrada $Cons\mathscr{T}$ que expresa la consistencia de $\mathscr{T}$ en el lenguaje de $\mathscr{T}$ no se puede demostrar en la teoría $\mathscr{T}$: $\quad \mathscr{T} \nvdash Cons\mathscr{T}$.*

(La demostración de los dos teoremas en el caso general sigue el mismo método como en el caso particular de PA, las fórmulas G y $Cons\mathscr{T}$ siendo definidas como imágenes de fórmulas aritméticas adecuadas a través de la inmersión de Q/PA adentro de la teoría $\mathscr{T}$.)

En particular, los dos teoremas de incompletitud se cumplen para una clase muy grande de teorías de primer orden, que va de la Aritmética de Peano (PA) a la teoría de conjuntos de Zermelo-Fraenkel (ZF) y sus extensiones usuales (con el axioma de la elección, el axioma de constructibilidad, la hipótesis generalizada del continuo o los axiomas de cardinales grandes, etc.), pasando por la Aritmética de orden n (PAn) para todo $n \geq 2$, la Aritmética de orden superior (PAω) y la teoría de conjuntos de Zermelo (Z) y sus extensiones usuales.

6.4. La Aritmética de Presburger (P). En otro lado, los teoremas de incompletitud no se cumplen para algunas teorías recursivas que no contienen Q (en el sentido de la existencia de una inmersión). Es el caso por ejemplo de la *Aritmética de Presburger* (notación: P), que es la subteoría de PA obtenida eliminando el símbolo $\times$ de su lenguaje así como todos los axiomas que refieren a este símbolo (pero manteniendo todos los axiomas de inducción que no refieren a este símbolo). En efecto, se puede demostrar que:

Teorema 6.10 (Completidud de P)**.** — *La Aritmética de Presburger* P *es completa, en el sentido en que para toda fórmula cerrada ϕ de su lenguaje:* $\mathrm{P} \vdash \phi$ *o* $\mathrm{P} \vdash \neg\phi$.

(La demostración utiliza la técnica de *eliminación de cuantificadores*, que permite mostrar que toda fórmula del lenguaje de P es equivalente a una fórmula sin cuantificadores.)

Además, ni se puede formular el segundo teorema de incompletitud para P, en la medida en que el lenguaje de P no es bastante expresivo para definir la fórmula *Cons*P.

Referencias

[1] G. Boolos, J. Burgess, R. Jeffrey. *Computability and Logic: Fourth Edition*, Cambridge University Press, Cambridge, UK. 2002.

[2] K. Gödel. *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I.* Monatshefte für Math. und Physik 38:173–98. 1931

[3] E. Mendelson. *Introduction to Mathematical Logic, 4th Edition.* Chapman & Hall, 1997.

[4] R. I. Soare. *Recursively enumerable sets and degrees: a study of computable functions and computably generated sets.* Perspectives in Mathematical Logic. Springer-Verlag, 1987.

Instituto de Matemática y Estadística Prof. Ing. Rafael Laguardia (IMERL) – Facultad de Ingeniería (Fing), Universidad de la República (UdelaR) – Julio Herrera y Reissig 565 – Montevideo C.P. 11300 – Uruguay

E-mail address: amiquel@fing.edu.uy