Concurso de monografías 2023

Unión Matemática Argentina

El problema de la palabra para grupos

Lautaro Ludueña

Vicente R. Schkolnik

Universidad Nacional de Córdoba
Facultad de Matemática, Astronomía, Física y Computación

Introducción

El problema de la palabra ha atravesado diagonalmente un importante grueso de historia matemática del siglo XX: desde su origen en la topología algebraica y la teoría combinatoria de grupos, hasta tiempos más maduros en que la temprana teoría de la computabilidad y la lógica supieron precisar y responder negatiavamente al problema. Así, este y otros dos problemas hermanos, el del isomorfismo y el de la conjugación, han habitado pesadamente la intersección de dichas áreas. Supieron entonces pasar por manos de matemáticos como Max Dehn, Alan Turing, Alonzo Church, Pyotr Novikov o Michael Rabin, entre muchos otros. En todo esto, hemos encontrado una maravillosa excusa para presentar y recorrer sectores de la matemática que creemos fascinantes, con ánimo de dar una respuesta concreta al asunto planteado, pero también de invitar al lector a introducirse y ahondar en ellos. Puntualmente, hablamos de la teoría combinatoria de grupos y de la computabilidad.

No se espera conocimiento alguno en teoría de la computabilidad, ni más que una familiaridad básica con la de grupos. Sin embargo, aún para esto último presentaremos definiciones elementales en un primer y breve capítulo preeliminar, que el lector allegado a estos conceptos puede eludir perfectamente. En el segundo capítulo definiremos grupo libre y presentaciones, las cuales servirán para motivar y plantear de forma más precisa el problema en cuestión:

> *Dado un grupo generado por un conjunto finito de generadores que cumplen ciertas relaciones (identidades), y dado un elemento expresado como producto de dichos generadores y sus inversos ¿Existe un método para deducir, a partir de las relaciones, si tal elemento es igual a la identidad del grupo?*

Posteriormente, un tercer capítulo independiente del desarrollo matemático en el trabajo ahondará la historia del problema, partiendo ya de un planteo lo suficientemente acabado del mismo. Sin embargo, para precisar lo que hemos llamado "método", en el capítulo 3 se brindará un recorrido veloz por teoría básica de computabilidad. Se hablará ahí de lenguajes, máquinas de Turing y algoritmos. Luego, en el capítulo 4 serán trabajadas las herramientas algebraicas técnicas que servirán para construir los ambientes en los que se prueban, ya en el quinto capítulo, los resultados finales.

Sin miedo a estropear sorpresa alguna, podemos adelantar esta cuestión: tal método que protagoniza la pregunta antes planteada **no existe** en general; hablamos de un problema **indecidible**. Más aún, sus dos problemas hermanos también lo son, junto a una espectacular familia de problemas análogos y que en el trabajo detallaremos. Tales problemas consisten siempre en decidir si un grupo satisface o no una propiedad, la cual debe apenas cumplir condiciones desgraciadamente simples y deseables.

Índice general

Capítulo 1

Preeliminares

> Las estructuras son las armas del matemático.
>
> *Nicolas Bourbaki*

1.1. Grupos

El ambiente más natural en nuestro recorrido consiste en la elemental y abundante estructura algebraica de grupo $(G, +)$. Aquí, G es un conjunto no vacío y $+ : G \times G \to G$, tal que $(g, h) \mapsto g + h$, es una operación binaria que satisface:

- $\forall a,\ b,\ c \in G : (a + b) + c = a + (b + c)$ (Asociatividad)
- $\exists e \in G : \forall a \in G,\ a + e = e + a = a$ (Existencia de elemento neutro)
- $\forall a \in G\ \exists a' \in G :\ a + a' = e$ (Existencia de inversos)

Solemos denominar $a' = -a,\ e = 0$, y en notación multiplicativa $a' = a^{-1},\ e = 1$. Ejemplos abundan: $(\mathbb{Z}, +),\ (\mathbb{Z}_n, +),\ (Gl(n, \mathbb{F}), \circ),\ (\mathbb{F}^n, +)$ (donde $\mathbb{F}$ es un cuerpo), entre muchos otros.

1.2. Subgrupos y coclases

Dado $(G, +)$ un grupo, un subconjunto $H \subset G$ será un subgrupo de G, denotándose $H < G$, si $(H, +)$ es un grupo. De los axiomas se deriva fácilmente que e debe pertenecer a H y actuar también de neutro. Quizás los ejemplos más representativos en un comienzo consistan en $(\mathbb{Z}, +)$, y los subgrupos $(m\mathbb{Z}, +)$ (de ahora en adelante obviaremos la operación, refieriéndonos solo al conjunto soporte).

Es fácil ver que la intersección de subgrupos será también un subgrupo. En particular, si G es un grupo y X es un subconjunto de este, la intersección de todos los subgrupos de G que contienen a X será un grupo, minimal con esta

característica. Lo denominamos el *subgrupo generado por X*, denotándolo $\langle X \rangle$. En el caso $X = \{g_1, ..., g_n\}$ (i.e. es finito), entonces se denota $\langle g_1, ..., g_n \rangle$, denominándolo subgrupo generado por $g_1, ..., g_n$. Cuando este último subgrupo es igual a G, diremos que tales elementos son generadores de G. No es difícil ver que $\langle X \rangle = \{h_1^{k_1}, ..., h_n^{k_n} |\ n \in \mathbb{N},\ h_1, ..., h_n \in X,\ k_1, ..., k_n \in \mathbb{Z}\ \}$. En particular, si $g \in G$, denotamos por $\langle g \rangle$ al subgrupo cíclico de G generado por g. El cardinal del conjunto soporte de un grupo se denomina *orden* del mismo.

En aritmética modular nos interesa analizar a los enteros módulo m, i.e. según su resto en la división por m. Tenemos así en cuenta los primeros m residuos positivos como representantes. La generalización natural de esta noción para grupos consiste en tomar $H < G$ y considerar la relación de equivalencia:

$$g \equiv_l h (mod\ H) \iff h^{-1}g \in H,$$

denominada relación de congruencia a izquierda. La congruencia a derecha es análoga, exigiendo que $gh^{-1} \in H$ y denotada por $\equiv_r$. Las clases de equivalencia de estas relaciones son llamadas coclases a izquierda o derecha y denotadas $[g]_l = gH,\ [g]_r = Hg$, respectivamente.

1.3. Cocientes

Como sucede con los residuos módulo m en $\mathbb{Z}$, nos interesa que las coclases en G módulo H puedan "heredar" la operación de G, y constituir un grupo por si mismas. Para que esto ocurra, H debe cumplir con la propiedad de ser **normal**: para todo $h \in H$ y para todo $g \in G$, $ghg^{-1} \in H$. Esto equivale a que las clases laterales coincidan: $gH = Hg,\ \forall g \in G$. En tal caso, denotamos $H \triangleleft G$, y sucederá que la operación $g_1H * g_2H = g_1g_2H$ esta bien definida, cumpliendo con los axiomas de grupo. Al grupo de coclases lo denominamos entonces **cociente** de G entre H, denotándolo G/H. Notemos que todo subgrupo de un grupo abeliano (esto es, tal que $gh = hg$ para todo $g, h \in G$) es normal; en particular, el conjunto de residuos módulo m en los enteros resulta, via la operación $[a] + [b] = [a + b]$, en el cociente $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$.

Debemos recordar también que, dados dos grupos $(G, *_G)$ $(H, *_H)$, una función $\phi : G \to H$ se dice *homomorfismo* si sucede que $\phi(g *_G h) = \phi(g) *_H \phi(h)$. Un homomorfismo se llamará *monomorfismo* si es inyectivo, *epimorfismo* si es sobreyectivo, e *isomorfismo* si cumple las últimas dos condiciones. En este caso, como la intuición indica, dos grupos isomorfos son equivalentes como grupos. Particularmente, si $H \triangleleft G$ y $\pi : G \to G/H$ es la proyección (la función basada en el mapa $g \mapsto gH$), esta se trata de un epimorfismo. De igual modo, la inclusión $\iota : H \hookrightarrow G$ es un monomorfismo (solo se necesita que H sea un subgrupo). Si $\phi : G \to K$ es un monomorfismo, diremos que G se incrusta en K, y cuando quieramos enfatizar esta situación utilizaremos «↪». En general, notemos que $Ker(\phi) = \{g \in G :\ \phi(g) = e_H\}$ es un subgrupo normal de G, e $Im(\phi)$ es un

subgrupo de K, para todo homomorfismo ϕ.

Por último, recordaremos el primero de los tres teoremas de isomorfía, el cual vamos a utilizar con frecuencia:

Teorema 1.3.1. *Si $\phi : G \to K$ es un homomorfismo y H es un subgrupo normal de G contenido en $Ker(\phi)$, entonces existe un único homomorfismo $\phi' : G/H \to K$ tal que $\phi'(\pi(g)) = \phi(g)$, siendo $\pi : G \to G/H$ la proyección. Más aún, ϕ' será un isomorfismo si y solo si ϕ es un epimorfismo y $H = Ker(\phi)$.*

La demostración de este teorema, junto al desarrollo más extendido y profundo de todas las ideas hasta aquí presentadas, pueden encontrarse en el libro de Thomas W. Hungerford, *Algebra*, en las secciones 1-7, Capítulo 1 [1].

Ponemos fin así a los preeliminares del trabajo. La noción de subgrupo generado y de generadores esta por ahora sujeta a un grupo-ambiente G por defecto. En él tomamos elementos concretos para generar subgrupos. Sin embargo, en abstracto, esto puede realizarse considerando elementos formales como generadores. Este asunto nos ocupará próximamente.

Capítulo 2

Presentaciones de grupos

Avanzaremos a las definiciones que dan lugar inmediato al problema de la palabra. Tener multitud de ejemplos concretos de grupos, provenientes de diversos rincones de la matemática, no debe desviar la atención de un hecho no menor: podemos construir una estructura formal de grupo, "aislada" de alguna manera, partiendo meramente de un conjunto primitivo. De hecho, el procedimiento de esta construcción revela que añadir *restricciones* a la forma en que operamos con los elementos del grupo, reconfigura la estructura general del mismo. Más aún, veremos que es posible con tales modificaciones alcanzar la estructura exacta de cualquier otro grupo.

2.1. Grupo Libre

Consideremos un conjunto arbitrario X que, intuitivamente, contendrá a los generadores del grupo que pretendemos formar. A este último denotaremos $F(X)$. Si $X = \emptyset$, definimos $F(X) = \{e\}$. Si $X \neq \emptyset$, consideramos X^{-1} un conjunto de igual cardinal. Dada entonces una biyección $X \to X^{-1}$, para cada $x \in X$ denotamos x^{-1} a su imágen. Por último, tendremos en cuenta un conjunto con un solo elemento, disjunto a $X \cup X^{-1}$; a tal elemento denotaremos 1.

Una palabra en X será una sucesión $(a_1, a_2, ...)$ de elementos en $X \cup X^{-1} \cup \{1\}$. En el conjunto de palabras en X, $(X \cup X^{-1} \cup \{1\})^*$, diremos que una palabra es reducida si satisface:

- Para todo $i \in \mathbb{N}$, si $a_i = x$, luego $a_{i+1} \neq x^{-1}$, y si $a_i = x^{-1}$ entonces $a_{i+1} \neq x$.
- Si $a_i = 1$ para algún i, entonces $a_j = 1$ para todo $j > i$.

En particular, la palabra $(1, 1, 1, ...)$ es reducida. Lamaremos a esta última *palabra vacía*, y la denotaremos por 1 (esta ambiguedad en la notación no debe afligir, pues no lleva a confusiones). Entonces, definiremos $F(X)$ como el conjunto de todas las palabras reducidas en X. Así, toda palabra en $F(X)$ es una sucesión $(x_1^{\lambda_1}, ..., x_k^{\lambda_k}, 1, 1, 1, ...)$ con $k \in \mathbb{N}$ y $\lambda_i \in \{-1, 1\}$, donde convenimos que x^1 denote a x, para todo $x \in X$; luego, la representaremos como $x_1^{\lambda_1} \dots x_k^{\lambda_k}$. Por construcción,

si $x_1^{\lambda_1} \dots x_k^{\lambda_k} = y_1^{\epsilon_1} \dots y_m^{\epsilon_k}$, entonces $m = k$ y $\lambda_i = \epsilon_i$, $x_i = y_i$, para cada i. Por lo tanto, el mapa $x \mapsto x^1$ de X en $F(X)$ es inyectivo, con lo que podemos considerar a X como un subconjunto de $F(X)$. Cuando X se sobreentienda, escribiremos $F = F(X)$

Nuestra intención es definir una operación en F. Es natural pensar que esta deba ser la yuxtaposición de palabras. Sin embargo, es rápido de advertir que esto puede producir palabras no reducidas; por lo tanto debemos refinar esta cuestión. Dadas $x_1^{\lambda_1} \dots x_k^{\lambda_k}$, $y_1^{\epsilon_1} \dots y_m^{\epsilon_k}$ dos palabras reducidas en X con $k \leq m$, sea t el mayor natural tal que $x_{k-i}^{\lambda_{k-i}} = y_{m-i}^{\epsilon_{k-i}}$, para todo $i \in \{1, 2, ..., k-1\}$ $(0 \leq t \leq k)$. Entonces, definimos:

$$(x_1^{\lambda_1} \dots x_k^{\lambda_k})(y_1^{\epsilon_1} \dots y_m^{\epsilon_k}) = \begin{cases} x_1^{\lambda_1} \dots x_{k-t}^{\lambda_{k-t}} y_{t+1}^{\epsilon_{t+1}} \dots y_m^{\epsilon_k} & si \quad t < m \\ y_{k+1}^{\epsilon_{k+1}} \dots y_m^{\epsilon_k} & si \quad t = k < m \\ 1 & si \quad t = k = m \end{cases}$$

En el caso en que $k > m$, la definición es análoga. Este producto consiste en yuxtaponer dos palabras y luego "reducirlas", con lo cual resulta en una operación cerrada dentro de F.

Teorema 2.1.1. *$F = F(X)$ forma un grupo con la operación antes definida. Además, $F = \langle X \rangle$.*

Demostración. Es claro que 1 es neutro por izquierda y por derecha, y que la palabra inversa a $x_1^{\lambda_1} \dots x_k^{\lambda_k}$ es $x_k^{-\lambda_k} \dots x_1^{-\lambda_1}$. Así, debemos nada más probar la asociatividad. Para esto, a cada $\lambda = 1, -1$ y a cada $x \in X$ asociamos el mapa $f_{x^\lambda} : F \to F$ tal que $1 \mapsto x^\lambda$ y:

$$f_{x^\lambda}(x_1^{\lambda_1} \dots x_k^{\lambda_k}) = \begin{cases} x^\lambda x_1^{\lambda_1} \dots x_k^{\lambda_k} & si \quad x^\lambda \neq x_1^{-\lambda_1} \\ x_2^{\lambda_2} \dots x_k^{\lambda_k} & si \quad x^\lambda = x_1^{-\lambda_1} \end{cases}$$

Puesto que $f_{x^\lambda} \circ f_{x^{-\lambda}} = f_{x^{-\lambda}} \circ f_{x^\lambda} = id$, estos mapas son biyecciones en F. Sean entonces $(\mathbb{S}(F), \circ)$ el grupo de biyecciones en F, $F' < \mathbb{S}(F)$ el subgrupo generado por $\{f_{x^\lambda} : \ x \in X, \ \lambda = 1\}$, y $\psi : F \to F'$ el mapa tal que $1 \mapsto id$ y $x_1^{\lambda_1} \dots x_k^{\lambda_k} \mapsto f_{x_1^{\lambda_1}} \circ \dots \circ f_{x_k^{\lambda_k}}$. Entonces, por definición $\psi(gh) = \psi(g)\psi(h)$, para cualesquiera $g, h \in F$. Más aún, dado que $f_{x_1^{\lambda_1}} \circ \dots \circ f_{x_k^{\lambda_k}}$ asigna $1 \mapsto x_1^{\lambda_1} \dots x_k^{\lambda_k}$, se tiene que:

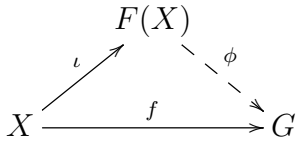
$$\psi(x_1^{\lambda_1} \dots x_k^{\lambda_k}) = \psi(y_1^{\epsilon_1} \dots y_m^{\epsilon_m}) \iff x_1^{\lambda_1} \dots x_k^{\lambda_k} = y_1^{\epsilon_1} \dots y_m^{\epsilon_m}.$$

Luego ψ es inyectivo y, por otro lado, debido a su definición, se sigue rápidamente que también es suryectivo. De este modo, al tratarse de una biyección, la estructura de grupo de F' implica que F satisface asociatividad y que ψ es un isomorfismo.

Finalmente, puesto que todo elemento de F se representa como producto de elementos en X, sus inversos, o es 1, es inmediato que $F = \langle X \rangle$. □

El grupo F, denominado **grupo libre** sobre X, satisface una propiedad universal expresada en el próximo teorema. Esta propiedad, que además lo asegura único en relación a X (salvo isomorfismo), permite describir de manera sencilla a los homomorfismos que parten de F hacia cualquier otro grupo, al mismo estilo en que describimos transformaciones lineales entre espacios vectoriales indicando la asignación sobre alguna base del espacio dominio. De hecho, a un nivel categórico, se trata del mismo fenómeno: $F(X)$ es el objeto libre sobre X en la categoría de grupos; también así, lo es un $\mathbb{K}$-espacio vectorial sobre alguna (cualquiera) de sus bases. No ahondaremos en el significado de estas últimas afirmaciones, que pueden revisarse también en el capítulo 1, sección 7 de [1].

Teorema 2.1.2. *Sean G un grupo, X un conjunto y $\iota : X \to F(X)$ la incrustación de X en el grupo libre. Entonces, para toda función $f : X \to G$ existe un único homomorfismo $\phi : F(X) \to G$ tal que $\phi \circ \iota = f$:*



La demostración puede ser algo extensa en sus detalles, pero esencialmente es sencilla y natural. Daremos por ello una idea de la misma.

Idea de la demostración. Vamos a construir el homomorfismo que extiende de naturalmente al mapa f. Comenzamos definiendo $\phi(1) = e$ y, para cada palabra en X reducida $w = x_1^{\lambda_1} \dots x_k^{\lambda_k}$ con k natural y $\lambda_i = 1, -1$, definimos $\phi(w) = f(x_1)^{\lambda_1} \dots f(x_k)^{\lambda_k}$. Este es un elemento bien definido de G, y así se deriva claramente que ϕ debe ser un homomorfismo tal que $\phi \circ \iota = f$. Más aún, si ϕ' es otro homomorfismo en iguales condiciones a ϕ, al cumplirse que $\phi'(x^1) = f(x) = \phi(x^1)$ para cada $x \in X$, se tiene que $\phi(x_1^{\lambda_1} \dots x_k^{\lambda_k}) = f(x_1)^{\lambda_1} \dots f(x_k)^{\lambda_k} = \phi'(x_1^{\lambda_1} \dots x_k^{\lambda_k})$, para toda $x_1^{\lambda_1} \dots x_k^{\lambda_k} \in F$. Luego $\phi = \phi'$. □

Notemos que, dados dos conjuntos X_1 y X_2, de este último teorema es fácil deducir que si $|X_1| = |X_2|$ entonces $F(X) \cong F(Y)$. Basta utilizar en lugar de f a las respectivas incrustaciones $\iota_1 : X_1 \to F(X_1)$, $\iota_2 : X_2 \to F(X_2)$, compuestas con la biyección que deberá existir entre X_1 y X_2; luego, los homomorfismos inducidos compondrán la identidad. Del mismo modo, es inmediato que todo otro grupo con la misma característica de F respecto a X que describe el teorema, será isomorfo a F. Un corolario todavía más interesante en lo próximo es el siguiente:

Corolario 2.1.1. *Todo grupo es la imágen, por vía de un homomorfismo, de algún grupo libre.*

Demostración. Basta convencerse de que dado un grupo G, existe un conjunto X de generadores del mismo (en particular $\langle G \rangle = G$). Luego, considerando en lugar de la función f del teorema a la inclusión $X \hookrightarrow G$, al generar X a G el homomorfismo $\phi : F(X) \to G$ inducido será suryectivo. Por lo tanto, $G = Im(\phi)$. □

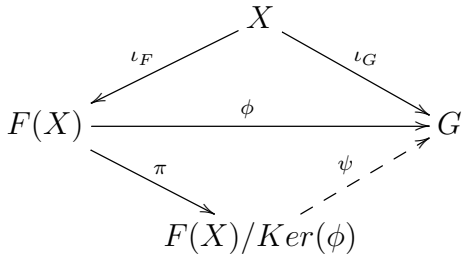
La intuición detrás del término *libre* es clara a esta altura: F no responde a ninguna restricción en su estructura. Cualquier combinación simbólica bien formada (reducida) de elementos en $X \cup X^{-1}$ es un elemento único e irreducible. Consecuencia de esto es la evidente infinitud de F si $|X| \geq 1$, o que el orden de todo elemento no trivial sea infinito también. Para más claridad, basta tomar el caso en que $|X| = 1$, donde $F \cong \mathbb{Z}$ (podemos pensar en $X = \{1\}$), y compararlo con $\mathbb{Z}_n$, también cíclico pero finito. En particular, el generador de este último grupo cumple la relación:

$$\underbrace{1 + \cdots + 1}_{n \ veces} = 0.$$

Con *relaciones*, aludimos a vínculos entre elementos del grupo, que notablemente harán cambiar la estructura del mismo. La próxima sección nos ocupa en delimitar rigurosamente estas nociones.

2.2. Presentaciones de grupos

Nuestro último corolario, sumado al primer teorema de isomorfía, permiten afirmar que todo grupo G es isomorfo a un cociente F/N, donde F es el grupo libre en un conjunto X de generadores de G, y $N \lhd F$ es el núcleo del epimorfismo ϕ del corolario:



Por ende, para describir cualquier grupo G, salvo isomorfismo, es suficiente con explicitar X y N, pues F se deriva en forma única a partir del conjunto de generadores. A la vez, N es determinado por cualquier subconjunto de F que lo genere. En particular, si $g = x_1^{k_1} \dots x_m^{k_m} \in F$ es un generador de N, luego en G se tendrá que $x_1^{k_1} \dots x_m^{k_m} = \iota_G(x_1)^{k_1} \dots \iota_G(x_m)^{k_m} = \phi(g) = e$. A la ecuación $x_1^{k_1} \dots x_m^{k_m} = e$ en G la denominamos **relación** en los generadores x_i. Entonces, con lo que hemos visto, cualquier grupo puede especificarse dando un conjunto X de generadores y otro conjunto R de relaciones en dichos generadores. Las mismas actúan como restricciones en la estructura del grupo libre, las cuales ocurren en G.

De modo recíproco, podemos preguntarnos si para un conjunto X y un conjunto R de palabras reducidas en X, existe un grupo G generado por X en donde las relaciones de R sean válidas. Por ejemplo, si $X = \{a\}$ y $R = \{a^n\}$ —donde a^n denota la yuxtaposición de a n veces—, entonces es fácil advertir que hablamos de la estructura de $\mathbb{Z}_n \cong F(\{a\})/\langle a^n \rangle$. En general, el procedimiento consistirá en cocientar el grupo libre generado por X, F, entre el subgrupo normal generado por R. Este último es la intersección de todos los subgrupos normales de F que contienen a R (es inmediato de probar que también será normal).

Definición 2.2.1. *Dado un conjunto X, sean R un conjunto de palabras reducidas sobre X, F el grupo libre en X y $N \lhd F$ el subgrupo normal generado por R. Decimos que un grupo G es definido por los generadores $x \in X$ y las relaciones $r = e$, con $r \in R$, si $G \cong F/N$. Además, diremos que $\langle X|R \rangle$ es una presentación de G.*

Decimos que un grupo es **finitamente presentado** si admite una presentación $\langle X|R \rangle$ con X y R finitos. En general, una presentación para un grupo dado no es única, y serán preferibles las más económicas en términos de los generadores y relaciones que utilicemos. Veamos a continuación algunos ejemplos para ilustrar mejor lo que hasta ahora vimos:

1. Ya hemos mencionado a $\mathbb{Z}_n \cong \langle a \mid a^n \rangle \cong \langle a, b \mid a^n,\ b \rangle$.
2. $\mathbb{Z} \oplus \mathbb{Z} \cong \langle a, b \mid aba^{-1}b^{-1} \rangle$.
3. El grupo diedral de reflexiones y rotaciones que mantienen invariante a un n-agono regluar, D_n, puede presentarse como $\langle x, y \mid x^n,\ yx = x^{-1}y \rangle$, donde x "interpreta" a la rotación en sentido horario de π/n rad, e y a la reflexión respecto a la mediatriz (fijada) de alguno de sus lados.
4. El grupo de permutaciones sobre un conjunto de 3 elementos, $\mathbb{S}_3$, admite la presentación $\langle a, b \mid a^2,\ b^3,\ (ab)^3 \rangle$. Aquí, a y b interpretan respectivamente a las trasposiciones (12) y (23).

2.3. El problema de la palabra

Como ya hemos sugerido, las relaciones de una presentación $\langle X|R \rangle$ pueden interpretarse como las "reglas" que determinan la estructura del grupo, en término de los generadores. Muchas preguntas son posibles de hacerse en este ambiente. Algunas podrían ser:

1. ¿Existen en R relaciones redundantes? Más precisamente, puede ocurrir que exista $R' \subsetneq R$ tal que $F(X)/\langle R' \rangle_N \cong F(X)/\langle R \rangle_N$ [1]. Ejemplos sencillos son $\mathbb{S}_3 = \langle a, b \mid a^2,\ b^2,\ (ab)^3,\ (aba)^2 \rangle$, como también $\mathbb{Z}_n = \langle a \mid a^{kn}\ (k \in \mathbb{Z}) \rangle$. En tal caso —y especialmente si R y X son finitos—, ¿cuál es el menor sistema de relaciones en R no redundantes que preserven a la presentación?

[1] Con el subíndice N denotamos al subgrupo normal generado.

2. Si consideramos una ecuación $x_1^{\lambda_1} \dots x_k^{\lambda_k} = y_1^{\delta_1} \dots y_m^{\delta_m}$ escrita en términos de los elementos de X (generadores), ¿es válida dentro de la presentación $\langle X|R\rangle$? Esto, formalmente, significa preguntarse:

$$\pi(x_1^{\lambda_1} \dots x_k^{\lambda_k}) \stackrel{?}{=} \pi(y_1^{\delta_1} \dots y_m^{\delta_m}),$$

donde $\pi : F(X) \to \langle X|R\rangle$ es la proyección al cociente. Equivalentemente:

$$\pi(x_1)^{\lambda_1} \dots \pi(x_k)^{\lambda_k} (\pi(y_1)^{\delta_1} \dots \pi(y_m)^{\delta_m})^{-1} \stackrel{?}{=} e$$

El tono de estas preguntas puede sugerir pensar a las presentaciones como estructuras formales con la relación de igualdad, donde las expresiones válidas —es decir, identidades— se consiguen con la manipulación simbólica de las palabras reducidas a partir de los axiomas de grupo sobre la operación interna, y las relaciones de R. Desde este punto de vista, la pregunta última consiste en determinar si una ecuación escrita en términos de los generadores se **deduce** de las relaciones. De modo implícito, este será el enfoque que usaremos en el capítulo 4.

Si se presta atención, nos daremos cuenta que ambas preguntas planteadas son resolubles si, dada una palabra reducida $w \in F$, somos capaces de decidir la validez de $\pi(w) = e$ (*i.e.* $wR = R$) Más aún, nos interesa mostrar un método para esto. Informalmente, usamos método para referirnos a un procedimiento rigurosamente especificado, que consiga indefectiblemente la solución: un **si** o un **no** como respuestas. Con más precisión, hablamos de un **algoritmo**. Debemos esperar al capítulo 4 para aclarar esta noción; sin embargo, ya hemos llegado al problema central de este trabajo:

> **Problema de la palabra.** Dado un grupo G finitamente presentado por $\langle X|R\rangle$ ¿Podemos decidir si, dada una palabra reducida en X cualquiera, esta es trivial en G? (más precisamente, la proyección de dicha palabra reducida).

Junto a este problema, Max Dehn publicó en 1911 [3] otros dos problemas vinculados estrechamente, y de importancia troncal en la teoría combinatoria de grupos:

> **Problema del isomofismo.** Dados G y G' grupos finitamente presentados por $\langle X|R\rangle$ y $\langle X'|R'\rangle$, respectivamente. ¿Podemos decidir si G y G' son isomorfos en base a sus presentaciones?

> **Problema de la conjugación.** Dado G, un grupo finitamente presentado por $\langle X|R\rangle$, ¿Podemos decidir si dos palabras reducidas cualesquiera en X son conjugadas en G?

Aunque nuestro claro protagonista es el problema de la palabra, terminaremos dando respuestas mucho más generales, como hemos adelantado en la introducción. Por otro lado, antes de seguir, conviene hacer una observación: el problema

de la palabra no depende realmente de la presentación del grupo G, si no de este último en si mismo. Efectivamente, si existe un procedimiento como en el problema para cierta presentación finita $\langle X|R\rangle$ de G, entonces en cualquier otra presentación finita $\langle X'|R'\rangle$ solo tendremos que traducir la palabra dada a una expresión escrita en los elementos de X. Para esto, es suficiente con traducir los elementos de X' que intervienen en la escritura de la palabra.

Definición 2.3.1. *Sea G un grupo finitiamente presentado por $\langle X|R\rangle$. Diremos que en G el problema de la palabra es resoluble (o que G tiene el problema de la palabra resoluble) si existe un algoritmo que, dada una palabra reducida cualquiera en X, decide si es trivial en G* [2].

Veamos algunos ejemplos de grupos donde el problema de la palabra es resoluble:

Ejemplo 2.3.1. *El problema de la palabra es resoluble para grupos libres finitamente generados. En efecto, dada una palabra reducida en X, basta comprobar su longitud para saber si es igual o no a 1 (pensando a la presentación como $\langle X|\ \rangle \cong F(X)$).*

Ejemplo 2.3.2. *El problema de la palabra es resoluble en $\langle a \mid a^n\rangle$ ($n \in \mathbb{N}$). Notemos que toda palabra reducida en $\{a\}$ será una concatenación finita del elemento a, o de a^1, o será 1. Entonces, si la palabra brindada es distinta de 1, solo debemos examinar la longitud de la misma y devolver **V** (verdadero; correcto) si es múltiplo de n, o **F** (falso; incorrecto) en caso contrario.*

Definición 2.3.2. *Un grupo G se dice residualmente finito si, para todo $g \in G$ tal que $g \neq e$, existe un sugbrupo normal $N \triangleleft G$ tal que $g \notin N$ y G/N es finito.*

Por el primer teorema de isomorfía, esta propiedad equivale a que para todo elemento no trivial $g \in G$, deben existir un grupo finito F y un homomorfismo $\xi : G \to F$ tales que $\xi(g) \neq e_F$.

La familia de grupos residualmente finitos es amplia: contiene a todos los grupos finitos, abelianos finitamente generados, entre otros. De hecho, todo grupo libre es residualmente finito. Un boceto de la prueba, para $m \geq 2$, consistiría en utilizar la propiedad universal del grupo libre, dada una palabra $x_1^{\epsilon_1} \dots x_m^{\epsilon_m}$ no trivial, para inducir un homomorfismo $\xi : F(X) \to \mathbb{S}_{m+1}$ —el grupo de permutaciones sobre $\{1, ..., m+1\}$—, a partir del mapa $x_i \mapsto (i\ i+1)$. Una vez realizado esto, es claro que $\xi(x_1^{\epsilon_1} \dots x_m^{\epsilon_m}) = (1\ 2\ 3 \dots m\ m{+}1) \neq 1$. Para $m = 1$, hablaríamos de $\mathbb{Z}$; dado $g = n$ en este caso, basta con tomar $N = m\mathbb{Z}$, donde m no sea un divisor de n.

Como era de imaginarnos por el sitio de esta última definición, el problema de la palabra es resoluble en todo grupo residualmente finito. Un algoritmo posible puede revisarse en la sección 2.2 de *A Course in the Theory of Groups* de

[2] Desde una perspectiva computacional, solo nos interesa la expresión formal simbólica de la palabra. Con más rigurosidad algebraica, nos interesa decidir la trivialidad de **la proyección** en G de la palabra reducida.

Robinson [2]. Pese al optimismo de nuestros últimos resultados, el problema de la palabra aún permanece entre grandes signos de interrogación en su formulación general. Esto es, en el ámbito de un grupo cualquiera.

Hasta aquí, hemos requerido introducir conceptos y resultados elementales, aunque no por ello menos complejos o abstractos, como es típico en la teoría de grupos. Dicha introducción, abarcada en los últimos dos capítulos, fue solo necesaria para poder formular de manera más o menos precisa el problema central. Lo que nos falta solamente, como lo hemos dicho varias veces ya, es delimitar y entender qué es un procedimiento efectivo. Tal tarea nos ocupará luego de una historización breve del problema.

Capítulo 3

Historia del problema

> Las Matemáticas tienen una historia continua de 5000 años de desarrollo con la corriente más importante de la cultura, en una actividad colectiva de hombres de talento inusual que atravesaron límites espaciales y temporales y crearon una de las maravillas del mundo.
>
> *Richard Mankiewicz*

Tanto en la introducción como en la presentación del problema en cuestión, hemos indicado que el mismo fue planteado en el marco de la topología algebraica. Ahora, sin embargo, ahodaremos más en sus recovecos históricos. A estos les hemos dedicado un breve capítulo independiente del transcurso puramente matemático del trabajo, pues consideramos que la agitada historia del problema, además de ser tan atrapante como su planteo y motivación, da cuenta de su importancia y magnitud. Además, el final de esta historia hace honor a la envergadura del problema, resultando en uno de los primeros resultados de insolubilidad fuera de contextos teóricos de la lógica.

Un problema en la intersección entre el álgebra y la topología

La historia del problema comienza con el nacimiento de la teoría combinatoría de grupos, de parte de Walther von Dyck, quien fue alumno de Felix Klein. Su objetivo era estudiar grupos discretos de isometrías del espacio hiperbólico. El artículo de 1882 *Gruppentheoretische Studie* contiene la primera aparición de una presentación de grupo como la conocemos. Asume (sin justificación rigurosa) la existencia de un grupo libre para cualquier conjunto finito y da como resulta-

do (otra vez sin una prueba rigurosa) que todo grupo de N generadores puede obtenerse del grupo libre sobre N elementos, sumando relaciones.

Heinrich Tietze publicó el artículo *Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten* en 1908. La parte topológica de este artículo se basó en la noción de grupo fundamental introducida por Poincaré en 1895. En el artículo de 1908, Tietze definió el grupo fundamental de una variedad (un tipo particular de espacio topologico) y demostró que era un invariante topológico. Para ello introdujo las transformaciones de Tietze. Demostró que dos presentaciones finitas cualesquiera para el mismo grupo pueden transformarse entre sí aplicando un número finito de transformaciones de Tietze. El mismo escribió en [5]:

> ≪Se observa de inmediato que puede ocurrir que dos grupos sean isomorfos aunque se presenten utilizando diferentes sistemas de generadores y relaciones definitorias. ... Sin embargo, no se ha resuelto ni el problema general de caracterizar la totalidad de formas abstractas de generar un grupo dado, ni siquiera el problema especial de encontrar un método para decidir si dos grupos dados por sus presentaciones son isomorfos.≫

En 1910 Max Dehn publicó *Über die Topologie des dreidimensionalen Raumes*. En este artículo, consideró el problema de cuándo dos nudos son equivalentes, basándose en el trabajo de Poincaré. Aquí se ocupó del grupo fundamental como una invariante clave, el cual es obtenido naturalmente como dado por una presentación. Rápidamente se dio cuenta de que los problemas de la teoría de nudos eran casos especiales de preguntas mucho más generales sobre grupos finitamente presentados. Hizo explícitos estos problemas, ya propios de la teoría de grupos, en su artículo de 1911 *Über unendliche diskontinuierliche Gruppen* [3]. Y al año siguiente, en su articulo *Transformationen der Kurven auf zweiseitigen Flächen* [4], presentó un algoritmo que resolvía un caso particular del problema de la palabra y el de la conjugación. Hoy este es conocido como *Algoritmo de Dehn*. Dehn sabía que el problema de la palabra era difícil y planteaba un tipo de pregunta completamente nuevo en matemáticas.

Aplicación de la lógica matemática

A la hora de intentar abarcar el problema surgieron muchas preguntas de naturaleza vital, no solo para la resolución del mismo sino para su correcta formulación. Por ejemplo, ¿cómo podría alguien probar que tal algoritmo no existía?; ¿cómo definimos matemáticamente lo que es un algoritmo?, etc. Se requirió la teoría de la computabilidad y desarrollos en lógica matemática para hacer que los interrogantes de Dehn fueran matemáticamente concretas. Mas aún, estas áreas no solo proporcionarían preguntas precisas, sino que también terminarían dando respuestas a las mismas.

Así, la primer pregunta a responder era ¿qué es precisamente un algoritmo? En la década de 1930, Kurt Gödel investigó cómo la manipulación simbólica en la

lógica formal podía simularse mediante funciones sobre los números naturales (hoy llamadas funciones recursivas). En ellas, cada par *entrada-salida* podía conseguirse mediante un procedimiento "mecánico" o computacional. Entonces, funciones con estas características en genreal serían denominadas *funciones computables*, y demarcarlas con precisición sería el camino a seguir para definir lo que es un procedimiento efectivo, o algoritmo. Independientemente de Gödel, Alonzo Church desarrollaba el cálculo λ, diseñado para aclarar cuestiones sobre los fundamentos de las matemáticas. En 1933, Church propuso que la intuición detrás de las funciones computables era capturada por su definición rigurosa de l-definibilidad (no profundizaremos en dicha definición), dentro del marco del cálculo λ. Para 1936, por otra parte, Alan Turing habría publicado su idea de una **Máquina de Turing** [15] (que él llamaba L.C.M, siglas de "logical computing machine"), proponiendo que una "función computable" era toda aquella que pudiera ser computada por una máquina de Turing. Rápidamente se demostró que una función era l-definible precisamente cuando podía ser calculada por una máquina de Turing. Así, ambos conceptos demostraban ser en verdad equivalentes. Esto, sumado al poder computacional de los dos paradigmas, fue suficiente para que casi toda la comunidad matemática aceptara la *Tesis de Church-Turing*, la cual afirma que ambos modelos computacionales (equivalentes) encarnan rigurosamente el concepto intuitivo de algoritmo, o función computable. Por fuera de una cuestión histórica, en el siguiente capítulo profundizaremos y ordenaremos algunas de estas definiciones e ideas, dedicándole también algún comentario a la tesis antes mencionada.

Ahora, una vez aclarado qué era exactamente un algoritmo o procedimiento efectivo, se podía hablar de lo que significaba un problema no resoluble (algoritmicamente). Informalmente, un problema P puede ser visto como un conjunto infinito de preguntas Q_i; entonces diremos que P es **decidible**, o **algortimicamente soluble**, si podemos computar la función $f : P \to \{Y, N\}$ con una máquina de Turing, donde $f(Q_i) = Y$ si la respuesta a Q_i es SÍ, y $f(Q_i) = N$ si la respuesta a Q_i es NO. Si podemos demostrar encambio que tal función f no puede ser computada, entonces el problema P es **indecidible** o **algoritmicamente insoluble**.

La quimera entre el álgebra y la lógica muestra su riqueza

En 1938, Church propuso que el problema de la palabra para grupos debería demostrarse insoluble utilizando las nuevas definiciones formales de computabilidad. En 1946 y 1947 Emil Post hizo el primer avance. Sin embargo, Post había tenido una suerte increíble: había producido muchos de los conceptos desarrollados por Gödel y Turing antes que ellos, pero su trabajo no se había considerado publicable [6]. Tal rechazo hizo que Post considerara que sus ideas requerían de un análisis profundo para ser aceptadas, por lo que esperó más de quince años para volver a presentar al mundo matemático sus ideas revolucionarias. Suficiente

tiempo para que Gödel, quien trabajo de forma independiente de Post, publicara sus resultados. Respecto a Turing, a ambos se les ocurrió la idea de una máquina de cómputos lógicos casi al mismo tiempo. Pero el artículo de Post [7] vio la luz cinco años después de Turing. Sin embargo, en él dio a conocer lo que hoy llamamos *el problema de la correspondencia de Post*, el cual demostró insoluble. Church vio el trabajo de Post y le sugirió que tratara de demostrar que el problema de la palabra para semigrupos era insoluble [8]. Tuvo éxito y publicó una prueba de tal resultado en 1947 [9]. Sin embargo, ese mismo año Andrey Markov Jr., quien había estado trabajando independientemente de Post, publicó una prueba del mismo resultado [8].

Turing aprendió sobre el problema de la palabra para grupos después de leer el artículo de Post. Pensó en el problema durante diez días y luego declaró que lo había resuelto. Hizo arreglos para dar un seminario describiendo la prueba, pero justo antes del seminario encontró un error [10] [11].

En 1956, Bill Boone, quien fue estudiante de docotrado de Church, finalmente demostró que el problema de la palabra para grupos era insoluble. Sin embargo, el matemático soviético Petr Sergeevich Novikov había estado trabajando en el problema y anunció en 1952 que tenía una prueba de la insolubilidad del problema de la palabra para grupos. Pocos le creyeron porque, aunque era un matemático muy conocido, tenía 51 años y había publicado principalmente en física matemática hasta ese momento. Pese a esto, en 1957 su trabajo le valió el premio Lenin [12]. Actualmente el teorema que afirma la insolubulidad del problema lleva el nombre de ambos: **Teorema de Novikov-Boone**.

Finalmente se había demostrado la insolubilidad algorítmica del problema, pero la historia aún no acababa. Un enfoque diferente resultó ser muy significativo. Los matemáticos Graham Higman, Bernhard Neumann y Hanna Neumann introdujeron en 1949 lo que ahora conocemos como una **extensión HNN** (por las inciales de sus creadores), un concepto algebraico que permitiria explotar el resultado de Novikov-Boone. El poder de esta última construcción se vio de dos formas. La primera es que brindaba una prueba diferente del resultado de Novikov-Boone. La segunda es que tal enfoque permitía aplicar este último a otros problemas de decisión de la teoría de grupos. El mayor ejemplo de esto es el teorema de Adian-Rabin, el cual afirma que una familia muy grande de problemas de decisión se pueden reducir al problema de la palabra y que, por tanto, son algorítmicamente insolubles. Dichos problemas consisten en determinar si un grupo cumple una propiedad P invariante por isomorfismos, tal que existe al menos un grupo que la cumple, y tal que existe al menos un grupo que no se incrusta en ningún grupo que si la cumpla. Este último resultado, de vasta generalidad, será también tratado en el final del trabajo, donde mencionaremos además algunas propiedades deseables y comunes que entran en la clase de propiedades descriptas antes.

Capítulo 4

Computabilidad

> Un cálculo es una forma especial de argumento matemático. A uno se le da un conjunto de instrucciones, y se supone que los pasos en el cómputo se siguen —se siguen deductivamente— de las instrucciones dadas. *Entonces, un cálculo es solo otra deducción matemática, aunque de una forma muy especializada.*
>
> *Saul Kripke*

Comenzaremos dando nociones elementales de la Teoría de la Computabilidad. En ella, entre otras cosas, se busca capturar formalmente la idea intuitiva de resolver algorítmicamente un problema matemático bien definido. Será esencial para esto contar, en primer lugar, con un soporte sobre el cual *codificar* los elementos importantes del problema. En segundo lugar, queremos que tal codificación sea interpretada y ejecutada por lo que llamaremos un *programa*. Estas dos nociones primitivas vendrán asociadas a los lenguajes formales y los autómatas. El lector dispuesto a profundizar en las ideas que presentaremos, puede recurrir al libro de Sipster, *Introduction to the Theory of Computation* [13], o al libro en español de Alfonseca-Alfonseca-Moriyón, *Teoría de Autómatas y Lenguajes Formales* [14].

De la siguiente primera sección, más allá de las definiciones de alfabeto, concatenación y lenguajes, no necesitaremos prácticamente nada más para secciones siguientes. Se incluyen sin embargo algunos aspectos algebraicos elementales de los lenguajes sobre un alfabeto fijado, que pueden ser de interés general.

4.1. Lenguajes

Definición 4.1.1. *Llamaremos **alfabeto** a un conjunto no vacío y finito.*

Los elementos de un alfabeto serán llamados **símbolos** o **letras**. Usualmente se denota a los alfabetos con una letra griega mayúscula, como Σ. Fijado un alfabeto, las palabras en él serán concatenaciones (i.e. secuencias finitas) de letras del mismo. De aquí en adelante, también las llamaremos **cadenas** de símbolos (usualmente referidas también como *strings*, del inglés). Más formalmente:

Definición 4.1.2. *Dado un alfabeto Σ, definimos las **palabras en** Σ con las siguientes cláusulas:*

1. *$\emptyset$ es una palabra en Σ. La denotaremos con el símbolo e y la llamaremos **palabra vacía**.*
2. *Todo $a \in \Sigma$ es una palabra en Σ.*
3. *Si λ es una palabra y $a \in \Sigma$, la concatenación formal λa es una palabra en Σ. Definimos además que si λ es una palabra, $e\lambda = \lambda e = \lambda$*
4. *Nada mas es una palabra en Σ.*

Evidentemente, lo anterior define un conjunto numerable, denotado Σ^* y denominado **universo del discurso** o **lenguaje universal** de Σ. Por ejemplo, dado $\Sigma = \{a\}$, $\Sigma^* = \{e, a, aa, aaa, ...\}$. En el lenguaje universal de un alfabeto, podremos definir una operación cerrada que extiende a la concatenación de letras. El tercer ítem de la última definición exhibe una función $(\lambda\ , a) \mapsto \lambda a$, nombrada por $C : \Sigma^* \times \Sigma \longrightarrow \Sigma^*$. De este modo, tenemos la siguiente definición recursiva:

Definición 4.1.3. *Sean Σ un alfabeto, $m \in \mathbb{N}$ y λ, $\sigma = a_1 a_2 ... a_m$ palabras en Σ. Definimos la concatenación $\lambda \cdot \sigma$, denotada simplemente $\lambda\sigma$, como:*

1. *$C(\lambda, a_1)$ si $m = 1$.*
2. *$C(\lambda \cdot a_1 ... a_{m-1}, a_m)$ si $m > 1$.*

Por definición entonces, la concatenación de palabras es cerrada en Σ^*. Más aún, es fácil comprobar que es asociativa y tiene a la palabra e como neutro a izquierda y derecha. Luego, $(\Sigma^*, \cdot)$ es un monoide.

Definición 4.1.4. *Se llama **lenguaje** a un subconjunto de Σ^**

Nótese que no deben confundirse los lenguajes $\emptyset$ con $\{\emptyset\} = \{e\}$. Al primero denotaremos por 0, y al segundo por 1. Nuevamente, podremos definir una estuctura algebraica sobre el conjunto de lenguajes en un alfabeto dado.

Definición 4.1.5. *Si L_1 y L_2 son lenguajes en un alfabeto, definimos:*

1. *$L_1 \vee L_2 = L_1 \cup L_2$*

2. $L_2 \wedge L_2 = \{\lambda\sigma : \lambda \in L_1 \wedge \sigma \in L_2\}$

Además, si L es un lenguaje, definimos L^n como el lenguaje 1 si $n = 0$, o como $L^{n-1} \wedge L$, si $n > 0$.

Denotaremos $\mathbb{L} = \mathcal{P}(\Sigma^*)$. El último operador que definimos en $\mathbb{L}$ es la clausura de Kleene.

Definición 4.1.6. *Sea L un lenguaje. Definimos su clausura (o clausura de Kleene) como $L^* = \bigvee_{n=0}^{\infty} L^n$.*

Finalmente, es fácil comprobar que $(\mathbb{L}, \vee)$ y $(\mathbb{L}, \wedge)$ forman monoides con 0 y 1 como neutros respectivos. Más aún, $L_1 \vee (L_2 \wedge L_3) = (L_1 \wedge L_2) \vee (L_1 \wedge L_3)$, con lo cual $(\mathbb{L}, \vee, \wedge)$ es un binoide distributivo. Se lo denomina **binoide libre generado por** Σ.

4.2. Maquinas de Turing

Alan Turing introdujo en 1936 la idea que acabaría acoplando su nombre al de *máquina*, y sentando las bases de la computación teórica. La misma abstrae las nociones —que discutiremos brevemente más adelante— de algoritmo y de cómputo, ligadas ambas a la de los problemas resolubles de forma algorítmica.

Informalmente, una máquina de Turing puede pensarse como un sistema automático y programable. Consta de una cinta con infinitas celdas dispuestas de forma horizontal, sobre las cuales podemos escribir símbolos en un cierto alfabeto e ingresar cadenas de caracteres a la máquina. Esta posee un cabezal capaz de desplazarse por la cinta, leer los símbolos y modificarlos, cambiando de estado en cada una de estas acciones. Un estado concreto del cabezal, junto al símbolo leído por este, determinan qué acción realiza.

Definición 4.2.1. *Una máquina de Turing es una 7-úpla $(\Sigma, \Gamma, \sqcup, Q, F, q_0, \delta)$ donde :*

- *$\Sigma \subsetneq \Gamma$ son alfabetos y $\sqcup \in \Gamma - \Sigma$ es un símbolo distinguido, denominado símbolo blanco.*
- *Q es un conjunto finito de estados, con $q_0 \in Q$ un símbolo distinguido llamado **estado inicial** y F un subconjunto de estados llamados **finales**.*
- *$\delta : D_\delta \longrightarrow Q \times \Gamma \times \{=, +, -\}$, con $D_\delta \subseteq (Q - F) \times \Gamma$, es la **función de transición**.*

En el alfabeto Σ (alfabeto de lectura) se ingresarán las palabras sobre la cinta, la cual la máquina es capaz de leer, y con los símbolos de Γ la máquina escribe sobre esta. Los infinitos espacios de la cinta no ocupados por las cadenas de caracteres ingresadas serán ocupadas por el símbolo blanco.

El funcionamiento de la máquina, informalmente, es como sigue: se ingresa una cadena de símbolos en Σ y, al comienzo de la ejecución, el estado inicial q_0 se situará sobre el primer símbolo de la cadena. En cada paso, la máquina consulta a la función de transición $\delta(q, \sigma)$, donde q es el estado actual y σ el símbolo en la celda de la cinta donde el cabezal se ubica. Si $\delta(q, \sigma)$ no esta definida, la máquina se detiene (en inglés, *halts*). En caso contrario, se siguen las instruciones de $\delta(q, \sigma)$: si $\delta(q, \sigma) = (q', \sigma', L,)$, con $L \in \{=, +, -\}$,

- cambiar en la celda el símbolo σ por σ';
- cambiar el estado del cabezal de q a q';
- si L es $=$, quedarse en la celda actual; si L es $+$ avanzar a la celda derecha; si L es $-$ retroceder a la celda izquierda.

Si acaso el estado al que llagamos por vía de δ pertenece a F (es decir, es un estado final), la máquina se dentendrá y habrá terminado el cómputo exitosamente. Nótese que según la función de transición y la cadena ingresada, la máquina podría entrar en un bucle (en inglés, *loops*) y nunca detenerse.

Al ser Q y Γ finitos, la función de transición puede especificarse mediante una **tabla de transición**. Otra representación usual para las máquinas de Turing y autómatas en general es la de **diagramas de estados**. Estos consisten de grafos en los que cada nodo representa un estado en Q, las aristas representan transiciones y cada una de ellas es acompañada de una especificación. Esto es, se anota el símbolo leído, el símbolo que se sobreescribe y el caracter $L \in \{=, +, -\}$. Los nodos de estados finales se escriben con alguna notación especial, para destacarlos.

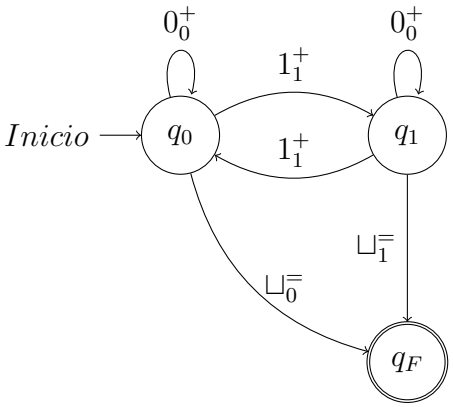
Ejemplo 4.2.1. *Consideremos la máquina de Turing dada por:*

$$M = (\Sigma = \{0, 1\}, \Gamma = \{0, 1, \sqcup\}, \sqcup, Q = \{q_0, q_1, q_F\}, F = \{q_F\}, \delta),$$

con la función de transición definida por la tabla:

δ	0	1	$\sqcup$
$\to q_0$	$(q_0, 0, +)$	$(q_1, 1, +)$	$(q_F, 0, =)$
q_1	$(q_1, 0, +)$	$(q_0, 1, +)$	$(q_F, 1, =)$
q_F			

Puede comprobarse con relativa facilidad el comportamiento de esta máquina: siempre avanza hacia la derecha en la cinta, saltando entre los estados q_0 y q_1 cuando lee el 1, y dejando sin alterar los símbolos 0 y 1. De esta forma, calcula la paridad de unos en la cadena ingresada, anotando al final de esta (al leer el símbolo blanco) dicha paridad. En general, resulta más sencillo comprender el funcionamiento de las máquinas y autómatas viendo sus diagramas de estados:



Vamos a representar más formalmente el funcionamiento de una máquina de Turing. Asumimos M $= (\Sigma, \Gamma, \sqcup, Q, F, q_0, \delta)$.

Definición 4.2.2. *Una configuración de una máquina de Turing M es una terna (v, q, w), con v, $w \in \Gamma^*$, $q \in Q$, que representa una situación en la que M se encuentra en un punto dado. Esto es, la máquina se encuentra en el estado q, la cinta tiene los símbolos de la cadena vw y el cabezal se encuentra en el primer símbolo de la subcadena w. La representamos por vqw.*

Dada una configuración $C = vqw$ de M, si w_0 es el primer símbolo de la cadena w, sobre la cual está el cabezal, y δ está definida sobre (q, w_0), entonces naturalmente la máquina pasará a una nueva configuración.

Definición 4.2.3. *Un **movimiento** o **transición** de M es el paso vía δ de una configuración, digamos C_1, a otra, C_2 (cuando esté bien definida). En este caso denotaremos $C_1 \vdash C_2$.*

Si existe una sucesión $C_0, ..., C_n$ tal que $C_{i-1} \vdash C_i$, $i = 1, ..., n$, entonces escribimos $C_0 \vdash^* C_n$.

Definición 4.2.4. *Sea M una máquina de Turing.*

- *Diremos que M **acepta** una entrada w si existen u, $v \in \Gamma^*$ tales que $q_0 w \vdash^* uq_F v$, con $q_F \in F$ (nótese que esto obliga a que el cómputo se detenga en $uq_F v$).*
- *Diremos que M **rechaza** una entrada w si existen u, $v \in \Gamma^*$ tales que $q_0 w \vdash^* uqv$, con $q \in Q - F$ tal que si v_0 es el primer símbolo de v, $(q, v_0) \notin D_\delta$ (es decir, δ no esta definida para esta configuración y por ende el cómputo se detiene, mas el estado último no es final).*

Una máquina de Turing puede usarse para operar de forma *mecánica* sobre las cadenas de símbolos entregadas y devolver una salida deseada. Esto, como ya hemos sugestivamente indicado, es abstraer la solución algorítmica a un problema,

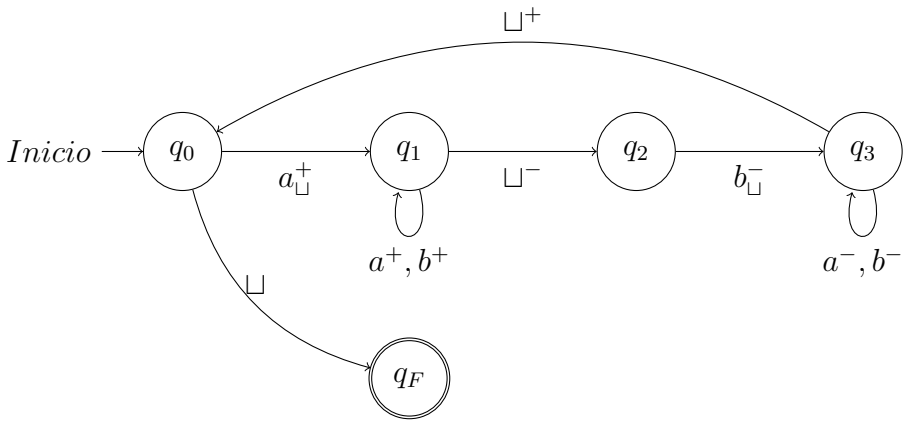
siempre que este sea traducible a un lenguaje sobre cierto alfabeto, en el cual logre definirse una máquina M que lo compute (de esto hablaremos en la sección siguiente). Por ejemplo, el lector podría intentar diagramar una máquina de Turing a la que puedan ingresarse dos números enteros en binario y devuelva la suma. Sin embargo, las últimas definiciones sugieren una relación (que es en verdad íntima) entre los lenguajes y las máquinas de Turing. La misma se extiende a otros tipos de autómatas y sirve para jerarquizar los lenguajes. Usaremos la definición de Sipster [13], capítulo 3.

Definición 4.2.5. *Sea M una máquina de Turing y L un lenguaje.*

- *Diremos que M* ***reconoce*** *a L si para toda $w \in L$, M acepta a w. Un lenguaje arbitrario se dirá* ***Turing-reconocible*** *si existe una máquina de Turing que lo reconozca.*
- *Diremos que M* ***decide*** *a L si para toda $w \in L$, M acepta a w, y si $w \notin L$ la rechaza. Un lenguaje arbitrario se dirá* ***decidible*** *si existe una máquina de Turing que lo decida.*

Observemos que, equivalentemente, una máquina decide cierto lenguaje si acepta únicamente las palabras de este, y se detiene para toda cadena ingresada. En el caso de un lenguaje Turing-reconocible, no necesariamente la máquina que lo reconoce debe detenerse para una entrada arbitraria. Obviamente, todo lenguaje decidible es Turing-reconocible. Por último, cabe decir que en muchas bibliografías, la primer clase de lenguajes es denominada **recursivamente enumerables**, mientras los segundos **enumerables**.

Ejemplo 4.2.2. *La siguiente máquina diagramada decide el lenguaje $L = \{a^n b^n : n \in \mathbb{N}\} \subset \{a, b\}^*$:*



Nótese que el comportamiento de esta máquina es el siguiente: borra la primera a en la cadena, para luego ir hasta la última b y borrarla también, continuando en zig-zag el proceso. Si el orden de las letras era correcto, y habían tantas letras a como b, la máquia borrará todas y finalizará exitosamente al leer el símbolo blanco. Puede verse que la máquina efectivamente decide a L: acepta sus palabras y rechaza cualquier otra.

4.3. Algoritmos y codificación

Hemos dicho ya que el formalismo de Turing condensa la intuición de algoritmo, o procedimiento efectivo. Solo nos queda aclarar dos cuestiones no menores, incluso en la dirección al principal problema abordado —el problema de la palabra—:

- Qué se entiende, finalmente, por algoritmo;
- cómo traducir problemas a entradas (cadenas) para ser ingresadas a una máquina de Turing.

Sobre la primera cuestión, debe decirse que en 1936, de manera independiente, Alan Turing [15] y Alonzo Church [16] demuestran la insolubilidad del *Entscheidungsproblem* de Hilbert. Tal problema consistía en determinar si era algorítmicamente decidible la validez o teoremicidad de una fórmula de primer orden. El primero implementó en su artículo las conocidas máquinas ya trabajadas; el segundo, Church, trabajó otro formalismo en apariencia muy distinto y menos evidente: el cálculo lambda. Sin embargo, demostraron ser modelos computacionales equivalentes. En cualquier caso, ambos trabajos debían primero enfrentarse a esclarecer lo que significaba *algorítmicamente decidible*. La equivalencia entre los dos modelos propuestos y su poder computacional sugieren la siguiente afirmación, matemáticamente indemostrable pero aceptada en muy gran medida: *todo algoritmo es equivalente a una máquina de Turing* (**tesis de Church-Turing**). Cabe decir que valen múltiples equivalencias a la anterior proposición, que reemplazan el modelo computacional de Turing por otros, como el ya mencionado cálculo lambda de Church, equivalentes todos entre si.

Si bien es claro que todo procedimiento o cómputo de una máquina de Turing es de naturaleza álgoritmica, radica en la parte recíproca de la tesis lo filosóficamente sustancial: de existir un procedimiento matemático que se pueda decir algorítmico (o pensarse como tal), es Turing-computable. No debe confundirse esto con las equivalencias entre modelos computacionales, las cuales están matemáticamente probadas.

Respecto al segundo ítem: para resolver un problema de forma algorítmica, o para probar que dicha labor es imposible, requerimos traducir los elementos del problema (o asumir que tal cosa puede hacerse) a un lenguaje determinado. Esto es, se debe **codificar**. La codificación dependerá del problema o del objeto matemático en cuestión. Dado un objeto O, denotaremos su codificación por $\langle O \rangle$. Debemos asumir que diversos objetos como grafos, autómatas (a nuestros efectos, máquinas de Turing), polinomios, o combinaciones de todos estos, son codificables.

Por ejemplo, de querer codificar grafos finitos no dirigidos a un determinado lenguaje, basta convencerse primero de que cada vértice es representable como un número natural, y cada arista como un par ordenado de vértices. Podemos así escribir en una cadena a todo grafo colocando primero los vértices numerados, y luego las aristas, en algún orden elegido. Lo que falta para expresarlos en un lenguaje, es asegurarnos de que las cadenas anteriores pueden escribirse sobre un alfabeto finito (de lo contrario, tal como hicimos hasta ahora, nuestro alfabeto sería numerable). Para esto, representamos cada número natural con un alfabeto unario $\{1, |\}$, denotando la separación entre símbolos originales con una barra y cada número como una cadena de unos:

$$\langle 123 \rangle = 1|11|111$$

Entonces, podemos escribir todo grafo finito no dirigido en el lenguaje $\{1, |, (,)\}$ (los paréntesis sirven para distinguir los pares que representan aristas, aunque podrían cambiarse por alguna cantidad establecida de barras).

En general, la descripción que se hace de una máquina de Turing es de alto nivel, confiando en que la descripción más detallada, de más bajo nivel (véase: especificar el lenguaje usado, los estados, la función de transición, entre otros) es realizable. Si por ejemplo, se quisiera probar la decibilidad del lenguaje

$$L = \{\langle G \rangle : \ G \text{ es un grafo finito, no dirigido y } \textbf{conexo}\},$$

basta definir la máquina M tal que, ingresado un grafo codificado $\langle G \rangle$, siga las instrucciones:

- Marcar el primer vértice.
- Repetir el siguiente procedimiento hasta que no queden nuevos vértices para marcar:
    - A cada vértice de G, marcarlo si esta vinculado por una arista a algún vértice antes marcado.
- Chequear todos los vértices. Si alguno no fue marcado, **rechazar**. En caso contrario (todos fueron marcados) **aceptar**.

El lector puede pensar la forma de implementar dicha máquina especificando sus partes y funciones. Lo importante es convencerse de que tal construcción es perfectamente posible. En las siguientes demostraciones, este será el nivel de descripción que haremos sobre las máquinas de Turing necesarias.

4.4. Decidibilidad

Vamos ahora a demostrar la existencia de lenguajes indecidibles; más aún, de lenguajes no Turing-reconocibles. Por la posibilidad de codificación que presentamos en la sección previa, y ateniéndonos a la tesis de Church-Turing, lo último implicará la existencia de problemas matemáticos no resolubles computacionalmente. Finalmente, estos resultados serán los necesarios para abordar el problema de la palabra.

Definición 4.4.1. *Utilizando una de las equivalencias de la tesis de Church-Turing, diremos que una función es* ***computable*** *si existe alguna máquina de Turing T que la compute; esto es, cada par entrada-salida de f, es un par entrada-salida de T (módulo codificación en el lenguaje que esta admite).*

Puesto que todo conjunto numerable puede biyectarse a un subconjunto de los naturales, y este a su vez es codificable en un lenguaje sobre un alfabeto unario $\{1, |\}$, podemos extender la noción de decibilidad de lenguajes a subconjuntos de conjuntos numerables arbitrarios utilizando la función característica del mismo. Esta función es, para un subconjunto $S \subseteq X$:

$$\chi_S(x) = \begin{cases} 1 & si \quad x \in S \\ \\ 0 & si \quad x \notin S \end{cases}$$

Definición 4.4.2. *Un subconjunto S de un conjunto numerable X es decidible si su función característica χ_S es computable*

Como trabajaremos más adelante, los problemas de decibilidad —es decir, los de tipo: *¿existe un algoritmo que decida si ...?*—, son reescribibles en términos de la decibilidad de conjuntos; a su vez estos, en los de la decibilidad de lenguajes. Veremos algunos resultados cardinales que darán pie a determinar lenguajes y conjuntos indecidibles.

Teorema 4.4.1. *Dado un alfabeto Σ, el conjunto de todos los lenguajes en él, $\mathbb{L}$, es no numerable.*

Demostración. Pensemos primero que Σ tiene un solo elemento, digamos a. Luego $\Sigma^* = \{e, a, aa, aaa, ...\} \approx \mathbb{N}$. Pero así $\mathbb{L} \approx \mathcal{P}(\mathbb{N})$, que es no numerable. Para ver con mas detalle esta afirmación, numeremos las palabras del lenguaje univeral $\Sigma^* = \{a^n\}_{n=0}^{\infty}$ e identifiquemos cada lenguaje $L \in \mathbb{L}$ con una sucesión $s_n \in \{0, 1\}^{\mathbb{N}}$ del siguiente modo:

$$s_n(L) = \begin{cases} 1 & si \quad a^n \in L \\ \\ 0 & si \quad a^n \notin L \end{cases}$$

Es fácil comprobar que esta asignación es biyectiva. Luego, de ser $\mathbb{L}$ numerable, digamos igual a $\{L_n\}_{n=0}^{\infty}$, consideremos el lenguaje $\mathcal{L}$ asociado a la sucesión:

$$S_n(\mathcal{L}) = \begin{cases} 1 & si \quad s_n(L_n) = 0 \\ \\ 0 & si \quad s_n(L_n) = 1 \end{cases}$$

(esto es, $\mathcal{L}$ es el lenguaje que contiene solo las palabras indexadas por valores naturales donde la sucesión S_n es igual a 1). Por definición entonces, $\mathcal{L} \notin \{L_n\}_{n=0}^{\infty} = \mathbb{L}$, absurdo. El caso general, donde $|\Sigma| > 1$, se sigue inmediatamente de que

$$\{a\} \hookrightarrow \Sigma, \text{ implica } \{a\}^* \hookrightarrow \Sigma^*, \text{ implica } \mathcal{P}(\{a\}^*) \hookrightarrow \mathcal{P}(\Sigma^*)$$

sumado a lo que hemos visto recién. □

Continuamos con un teorema sobre la cardinalidad del conjunto de máquinas de Turing, que da pie a los primeros resultados de indecibilidad.

Teorema 4.4.2. *El conjunto de las máquinas de Turing es numerable.*

Idea de la demostración. Biyectaremos cada alfabeto con un subconjunto finito de $\mathbb{N}$. Tal familia es numerable. Luego, por cada forma de elegir una 2-upla de subconjuntos naturales finitos (Γ, Q), $(|\Gamma| > 1)$, nos quedan solo finitas maneras de elegir subconjuntos $\emptyset \neq \Sigma \subsetneq \Gamma$, $F \subseteq Q$, funciones $\delta \subseteq (Q - F \times \Gamma) \times (Q \times \Gamma \times \{+, -, =\})$ y símbolos $\sqcup \in \Gamma - \Sigma$, $q_0 \in Q$. Cada una de estas elecciones determinan una máquina de Turing. Entonces, el conjunto de máquinas es coordinable a un producto cartesiano finito de conjuntos contables, y por lo tanto es contable (numerable, más aún). □

Usamos el hecho de que los alfabetos son mapeables a subconjuntos naturales finitos, que bien podrían ser secciones (i.e., de la forma $\{1, ..., n\}$). A su vez, los demás componentes de la máquina se piensan también como símbolos de alfabetos encriptados en números. Hay que notar que esto permite traducir entonces cualquier alfabeto (y así todo lenguaje) a un solo universo sobre un alfabeto unario, que puede constar de un símbolo especial para denotar separaciones u otros caracteres especiales: $\{1, |\}$. Lo realizamos vía la ya utilizada asignación

$$n \mapsto \underbrace{1..,1}_{\text{n}}.$$

Cómodos con esta idea, se nos presenta un corolario del anterior teorema:

Corolario 4.4.1. *Existen lenguajes no Turing-reconocibles (y por ende indecidibles).*

Demostración. Por cada lenguaje L Turing-reconocible, existe al menos una máquina de Turing T que lo reconoce; esto define un mapa $L \mapsto T$. Más aún se trata de una inyección: si $L \neq L'$, existe $w \in L - L'$ y por ende la máquina T' asociada a L' rechaza a w o no detiene su cómputo (*loopea*) en ella. Sin embargo, la máquina T asociada a L la acepta, por definición. Entonces $T \neq T'$. Por lo tanto hay a lo sumo numerables lenguajes Turing-computables (de hecho, de manera recíproca, cada máquina de Turing define un lenguaje reconocible maximal: el de las cadenas que acepta). Luego, al haber no numerables lenguajes, existen lenguajes no Turing-reconocibles. □

Veamos al respecto del universo $\{1, |\}^*$, y de modo esquemático, un ejemplo importante sobre el cual debe depositarse confianza para algunos de los siguientes teoremas. Describimos una máquina de Turing T en el propuesto alfabeto de este modo:

- Los estados de T se numeran en notación unaria:

$$1, 11, 111, ..., 1^{|Q_T|}$$

- Los símbolos del alfabeto también se pueden representar de la misma manera:

$$1, 11, 111, ..., 1^{|\Gamma_T|}$$

- Las direcciones del cabezal se pueden describir con tres símbolos: $1, 11, 111$.

- Cada transición

$$\delta_T(q, a) = (q', a', Dir)$$

  se puede codificar como

$$q|a|q'|a'|Dir$$

- Finalmente, para codificar por completo la máquina T, hay que indicar cuales son el estado inicial, los estados finales y cada una de las transiciones posibles. Esto podría hacerse así:

$$\langle q_0\rangle||\langle q_{F1}\rangle|\langle q_{F2}\rangle|...|\langle q_{Fk}\rangle|||\langle\delta_1\rangle||...||\langle\delta_j\rangle$$

  en donde $\langle q\rangle$ representa la codificación del estado q, y $\langle\delta\rangle$ la codificación de la transición δ (y en general $\langle.\rangle$ representa codificación).

No debe alarmar que distintos símbolos adquieran igual representación. La misión de esta codificación es construir máquinas de Turing capaces de **interpretar** cualquier otra máquina y **simular** su cómputo. En este sentido, la codificación descripta alcanza, y solo reside en los detalles de programación de tales máquinas el cómo distinguir entre sí los símbolos y sus funciones (por ejemplo, mediante ordenamientos). Una máquina en este lenguaje programada para simular cualquier otra máquina de Turing, es llamada **máquina universal** y la denotamos U. El

último aspecto sobre la encriptación anterior, es que para escribir una cadena que codifique un par *máquina-entrada* (M, w), podemos ingresar

$$\langle M\rangle||||\langle w\rangle,$$

denotándola como $\langle M, w\rangle$.

4.5. *Halting Problem* e Indecibilidad

Con las herramientas que obtuvimos, enfrentamos ahora dos teoremas de los más importantes filosóficamente en lógica y computación: existen problemas algorítmicamente irresolubles. A esta familia se unirá el problema de la palabra más adelante. En primer lugar, consideremos la siguiente pregunta: *¿Es posible decidir si, dada una máquina de Turing M y una entrada w, M se detiene en w?*. Pues tal problema indecidible.

Teorema 4.5.1. *El lenguaje*

$$A_{MT} = \{\langle M, w\rangle : M \text{ es una máquina de Turing y acepta a } w\}$$

es indecidible.

Demostración. Supongamos que fuera decidible. Entonces, existe una máquina D que decide al lenguaje. Esquemáticamente:

$$D(\langle M, w\rangle) = \begin{cases} \text{acepta} & \text{si } M \text{ acepta } w \\ \\ \text{rechaza} & \text{si } M \text{ no acepta } w \end{cases}$$

Contruimos ahora una máquina nueva R, la cual recibe una máquina encriptada $\langle M\rangle$ y simula D con la entrada $\langle M, \langle M\rangle\rangle$, pero devuelve lo contrario: si D acepta $\langle M, \langle M\rangle\rangle$, entonces R rechaza, y si D la rechaza, acepta. En resumen:

$$R(\langle M\rangle) = \begin{cases} \text{acepta} & \text{si } M \text{ no acepta } \langle M\rangle \\ \\ \text{rechaza} & \text{si } M \text{ acepta } \langle M\rangle \end{cases}$$

Pero esto significa que:

$$R(\langle R\rangle) = \begin{cases} \text{acepta} & \text{si } R \text{ no acepta } \langle R\rangle \\ \\ \text{rechaza} & \text{si } R \text{ acepta } \langle R\rangle \end{cases}$$

lo cual es absurdo. Tal contradicción provino de suponer que una máquina D decisora de A_{MT} existe. □

Aún podría guardarse esperanza sobre la solubilidad computacional de otro problema más débil: decidir si una máquina de Turing dada detiene su cómputo en determinada entrada. Popularmente, este problema se conoce como **halting problem**, o problema de parada. En el mismo artículo de 1936 antes citado, Alan Turing prueba que tal problema es también indecidible. La siguiente demostración resulta como corolario de nuestro último teorema.

Teorema 4.5.2. *El lenguaje*

$$HALT_{MT} = \{\langle M, w\rangle : M \text{ es una máquina de Turing y se detiene en } w\}$$

es indecidible.

Demostración. Supongamos que, por el contrario, existe una máquina D que decide $HALT_{MT}$. Procederemos construyendo una máquina H que ejecute a D como subrutina. Dada una entrada $\langle M, w\rangle$:

- Simular D en $\langle M, w\rangle$;
- si D rechaza $\langle M, w\rangle$, **rechazar**;
- si D acepta $\langle M, w\rangle$:
  - Simular M en w;
  - si M acepta w, **aceptar**;
  - si M rechaza w, **rechazar**.

Esta máquina, cuya existencia se debe a la de D, decide al lenguaje A_{MT}. Por esta contradicción, $HALT_{MT}$ es indecidible. □

Corolario 4.5.1. *Existen lenguajes Turing-reconocibles pero no decidibles.*

Demostración. Consideremos al propio $HALT_{MT}$, que como vimos es indecidible. Tomemos en cuenta la máquina universal de Turing U, y hagamos que dado un par máquina-string $\langle M, w\rangle$, U simule $M(w)$. De este modo $U(\langle M, w\rangle)$ se detiene si y solo si $M(w)$ lo hace. Es decir, U reconoce a $HALT_{MT}$. □

Hacemos una última observación, que se ya se habrá advertido desde un comienzo. Los objetos más elementales en esta sección fueron alfabetos, lenguajes y palabras. Es claro que en el caso de grupos libres, productos libres —parecidos en espíritu, como veremos— y presentaciones, se manifiestan los mismos objetos con nombres distintos. Sin ir más lejos, el criterio que establece qué palabras sobre ciertos generadores son reducidas, cuando el sistema de generadores es finito, determina la gramática formal para un lenguaje concreto: el grupo libre en dados generadores. Este paralelismo hace más clara la intuición detrás del cómo la teoría de la computablilidad opera orgánicamente dentro de este sector del álgebra y la teoría de grupos en particular.

Capítulo 5

Producto libre y extensiones HNN

> El álgebra invierte la importancia relativa de los factores en el lenguaje ordinario. Es esencialmente un lenguaje escrito, y se esfuerza por ejemplificar en sus estructuras escritas los patrones que tiene como propósito transmitir.
>
> *Alfred North Whitehead*

Volvemos al terreno del álgebra. Revisaremos las últimas herramientas necesarias para los resultados finales. Estos últimos volverán a requerir las nociones de computabilidad. Los conceptos que vamos a ver son el producto libre, el producto libre amalgamado y las extensiones HNN. De los tres, los primeros dos son necesarios para demostrar la insolubilidad del problema de la palabra; el último será necesario para probar un resultado de indecidibilidad aún más general.

5.1. Producto Libre

Dada $\{G_i\}_{i\in I}$ una familia de grupos, que podemos asumir disjuntos, haremos una construcción similar a la del grupo libre sobre un conjunto de generadores. Sea $X = \bigcup_{i\in I} G_i$ y sea $\{1\}$ un conjunto de un solo elemento, disjunto con X. Una palabra en X es un sucesión de la forma $(a_1, a_2, \cdots)$ donde $a_n \in X \cup \{1\}$ y existe algún $m \in \mathbb{N}$ tal que $a_n = 1$, para todo $n > m$. Esto es, cada palabra en X puede pensarse como una secuencia de $(X \cup \{1\})^{<\omega}$, o igualmente como una cadena en $(X \cup \{1\})^*$, desde el punto de vista de los lenguajes.

Diremos que una palabra en X está reducida si:

1. ningún $a_n \in X$ es el elemento neutro del grupo respectivo;

2. para todo $i \in I$, elementos yuxtapuestos $a_i,\ a_{i+1}$ no están en el mismo grupo;

3. si $a_n = 1$, entonces $a_m = 1$ para todo $n \leq m$.

Así, podemos escribir cualquier palabra reducida de forma unica como

$$(a_1, \cdots, a_s, 1, 1, 1, \dots) = a_1 \cdots a_s.$$

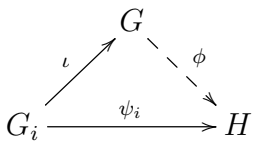
El procedimiento para transformar una palabra en X a su forma reducida es removiendo toda instancia de los elementos neutros $e_i \in G_i$ de cada grupo, y reemplazando cada subpalabra $g_1 g_2$, con $g_1,\ g_2 \in G_i$ para algún i, por su producto (palabra de longitud uno) en G_i. Nótese además que la palabra $1 = (1, 1, 1, ...)$ es reducida.

Es relativamente fácil comprobar que el conjunto de palabras reducidas en X es un grupo con la siguiente operación binaria: dadas $a_1 \cdots a_n$ y $b_1 \cdots b_k$, definimos su producto como la palabra reducida obtenida por la yuxtaposición y posterior reducción de ambas. Es decir, a la palabra $a_1 \cdots a_n b_1 \cdots b_k$ se le aplica el procedimiento antes explicado para reducirla. Por ejemplo, si $a_i,\ b_i \in G_i$ con $i \in \{1, 2, 3\}$, entonces $a_1 a_2 a_3 \cdot a_3^{-1} b_2 b_3 = a_1 c_2 b_3$, donde $c_2 = a_2 b_2 \in G_2$. Queda también claro que la palabra 1 actúa de neutro.

Al producto libre lo denotamos como $\Pi^*_{i\in I} G_i$. Observemos que para cada G_i existe un monomorfismo de grupos canónico $\iota_i : G_i \to \Pi^*_{i\in I} G_i$ dado por los mapas $e_i \mapsto 1$ y $g \mapsto (g, 1, \cdots)$, que por simplicidad notacional y por la universalidad de su función se denota como ι simplemente, omitiendo su índice. Con esto, el producto libre satisface una propiedad universal dual al producto directo, que además asegura su unicidad, elevado a isomorfismo. Su demostración es rutinaria, por lo que solo esbozamos la idea.

Teorema 5.1.1. *Sean $\{G_i\}_{i\in I}$ una familia de grupos y $\Pi^*_{i\in I} G_i$ su producto libre. Si $\psi_i : G_i \to H$ es una familia de homomorfismos de grupos, entonces existe un único homomorfismo $\phi : \Pi^*_{i\in I} G_i \to H$ tal que $\phi\iota = \psi_i$, para todo $i \in I$. Más aún, todo grupo con esta propiedad es isomorfo a $\Pi^*_{i\in I} G_i$.*

Idea de la demostración. Sea $G = \Pi^*_{i\in I} G_i$. Para cada $i \in I$, debemos establecer ϕ tal que conmute el diagrama:



El mismo es inducido naturalmente a través del mapa $G \to H$ dado por $\iota(g) = (g, 1, 1, ...) \mapsto \psi_i(g)$, con $g \in G_i$. Extendiendo esta asignación a todo G, obenemos:

$$\psi(\iota(g_1)\iota(g_2)\dots\iota(g_n)) \doteq \psi_{i_1}(g_1)\psi_{i_2}\dots\psi_{i_n}(g_n),$$

donde $g_k \in G_{i_k}$ para cada $k \in \{1, ..., n\}$. Se demuestra tal función bien definida, homomorfismo y única. Por último, si existiese $G^{'}$ con la misma propiedad, aplicamos el teorema dos veces, primero a $H = G$ y luego a $H = G^{'}$. Obtenemos respectivamente $\phi_1 : G \to G^{'}$ y $\phi_2 : G^{'} \to G$ tales que $\phi_2\phi_1 = id_G$, $\phi_1\phi_2 = id_{G^{'}}$. Entonces $G \cong G^{'}$.

□

Nos queda ver la forma del producto libre en término de las presentaciones de los grupos involucrados. Más de una vez abusaremos de notación, escribiendo $=$ en vez de $\cong$, como será el caso para la caracterización del producto libre en la siguiente afirmación.

Proposición 5.1.1. *Sea $\{G_i\}_{i\in I}$ una familia de grupos tal que $G_i = \langle X_i | R_i\rangle$. Entonces $\Pi^*_{i\in I} G_i = \langle \bigcup_{i\in I} X_i | \bigcup_{i\in I} R_i\rangle$*

Idea de la demostración. Solo hace falta ver que el grupo $G = \langle \bigcup_{i\in I} X_i | \bigcup_{i\in I} R_i\rangle$ cumple la propiedad del producto libre del anterior teorema. Primero necesitamos una familia de monomorfismos $\iota_i : G_i \to G$. Para cada i, ι_i es el extensión inducida por el mapa $x\langle R_i\rangle_N \mapsto x\langle \bigcup_{j\in I} R_j\rangle_N$ tal que:

$$\iota_i(x_1^{k_1}\dots x_n^{k_n}\langle R_i\rangle_N) \doteq x_1^{k_1}\dots x_n^{k_n}\langle \bigcup_{j\in I} R_j\rangle_N,$$

con $n \in \mathbb{N}$, $k_1 \dots k_n \in \mathbb{Z}$ y $x_1 \dots x_n \in X_i$. Tal extensión resulta bien definida, inyectiva y un homomorfismo. Ahora, sea $K = \langle \bigcup_{i\in I} X_i | \bigcup_{i\in I} R_i\rangle$. Dados H un grupo y $\psi_i : G_i \to H$ una familia de homomorfismos, definimos $\phi : K \to H$ extendiendo al mapa $x\langle \bigcup_{j\in I} R_j\rangle_N \mapsto \psi_i(x)$, con $x \in X_i$ para cada $i \in I$, de modo que:

$$\phi(x_{i_1}^{k_1}\dots x_{i_n}^{k_n}\langle \bigcup_{j\in I} R_j\rangle_N) = \psi_{i_1}(x_{i_1})^{k_1}\dots\psi_{i_n}(x_{i_n})^{k_n},$$

con $x_{i_m} \in X_{i_m}$, para $m \in 1, ..., n$. Tal ϕ se demuestra bien definido, homomorfismo y único. Entonces $K \cong \Pi^*_{i\in I} G_i$.

□

Ejemplo 5.1.1. *Se tiene que:*

- $\mathbb{Z} * \mathbb{Z}$ *tiene por presentación* $\langle a, b \mid \rangle$.
- *El producto libre* $\mathbb{Z}_2 * \mathbb{Z}_3 = \langle a, b \mid a^2, b^3\rangle$ *es infinito y no abeliano. Algunos de sus elementos son, por ejemplo,* $(1_2, 2_3, 1_3)$, $(2_3, 1_2)$, *donde el subíndice*

*$i = 2, 3$ inidica que el elemento es de $\mathbb{Z}_i$. Por una parte, el grupo contiene el conjunto $\{(1_2), (1_2, 1_3), (1_2, 1_3, 1_2), \dots\}$, con lo cual es infito. Por otra, los elementos $a = (1_2), b = (1_2, 2_3)$ no conmutan, pues $ab = (1_2)(1_2, 2_3) = (1_2 + 1_2, 2_3) = (0_2, 2_3) = (2_3)$ y $ba = (1_2, 2_3)(1_2) = (1_2, 2_3, 1_2)$. A este grupo se le conoce como **Grupo modular** y es muy importante en varias ramas de las matemáticas.*

- *De la última proposición vista se sigue que el producto libre de grupos libres es otro grupo libre. Más todavía, el producto libre de los grupos libres $F(X_i)$, donde X_i son conjuntos, es el grupo libre $F(\bigcup_{i\in I} X_i)$.*
- *El grupo dado por la presentación $\langle a, b | a^2, b^2\rangle$ es llamado **grupo diedral infinito**, por la ultima proposición se tiene que el grupo $\mathbb{Z}_2 * \mathbb{Z}_2$ satisface dicha presentación.*

5.2. Producto libre amalgamado

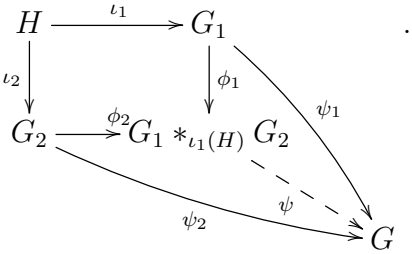
Sean G_1 y G_2 grupos y $A_1 < G_1$ y $A_2 < G_2$ subgrupos isomorfos entre si, donde $\psi : A_1 \to A_2$ es un isomorfismo.

Definición 5.2.1. *El producto **libre amalgamado** es el cociente del producto libre $G_1 * G_2$ entre la clausura normal del conjunto $\{\psi(a)a^{-1} | a \in A_1\}$.*

Nótese que, para cada $j = 1,\ 2$, la composición de la inclusión $i_j : G_j \hookrightarrow G_1 * G_2$ con la proyección canónica π de $G_1 * G_2$ sobre el produto libre amalgamado induce un homomorfismo $\phi_j = \pi i_j$ de G_j sobre el producto libre amalgamado. Si se sobrentiende que $A < G$ es isomorfo por ψ a $\psi(A) < H$, se denota al producto libre amalgamado (como en la definición anterior, con $A_1 = A,\ A_2 = \psi(A)$ y $H = G_2$) como $G *_A$.

La idea detrás del producto libre amalgamado es que, teniendo dos grupos con subgrupos isomorfos entre si, forcemos a estos últimos a ser "iguales" en el producto libre (es decir, se amalgaman). El producto libre amalgamado viene a satisfacer una propiedad universal que lo asegura único, *módulo isomorfismo.*

Teorema 5.2.1. *Sean H, G_1, G_2 grupos tales que $\iota_1 : H \to G_1$ y $\iota_2 : H \to G_2$ son monomorfismos. Además, sean G un grupo y $\psi_1 : G_1 \to G$ y $\psi_2 : G_2 \to G$ homomorfismos. Entonces, existe un unico homomorfismo $\psi : G_1 *_{\iota_1(H)} G_2 \to G$ tal que el siguiente diagrama conmuta:*



Más aún, el producto libre amalgamado $G_1 *_{\iota_1(H)} G_2$ es único con esta propiedad, salvo isomorfismo.

La demostración es rutinaria pero extensa. El mismo tampoco será necesario en lo que sigue.

5.3. Extensiones HNN

Con el producto libre presentaremos una última herramienta: extensiones HNN (Higman/Neumann/Neumann). De las mismas se derivará el lema de Britton, de importancia general en el problema.

Sean G un grupo y A, B dos subgrupos de G isomorfos entre si por vía de $\psi : A \to B$, un isomorfismo. Sea t un elemento tal que $\{t\} \cap G = \emptyset$ y sea $\langle t \rangle$ el grupo libre generado por $\{t\}$ (isomorfo a $\mathbb{Z}$).

Definición 5.3.1. *La **extención HNN** de G relativa a A, B, ψ, denotada como $\langle G, t \mid t^{-1}at = \psi(a),\ a \in A \rangle$, se define como el cociente del producto libre $G * \langle t \rangle$ entre la clausura normal del conjunto $\{t^{-1}at\psi(a)^{-1} : a \in A\}$. A t se le llama letra estable.*

Si ψ se sobreentiende, entonces denotamos la extensión como G^*. La idea detrás de esto es tomar dos subgrupos isomorfos de G y forzarlos a ser conjungados dentro de un nuevo grupo-ambiente. Como la extensión HNN es un cociente de grupos, tendremos la proyección canonica $\pi : G * \langle t \rangle \to G^*$, que envía cada elemento a su clase de equivalencia. Usaremos π para denotar dicha proyección en lo que resta del texto.

En algunas bibliografías las relaciones que se utilizan para definir el cociente son $tat^{-1}\psi(a)^{-1}$. Ambas definiciones son equivalentes: se considera el isomorfismo $\phi : G * \langle t \rangle \to G * \langle t \rangle$ inducido por el mapa $t \mapsto t^{-1}$. Entonces el epimorfismo $\pi \circ \phi : G * \langle t \rangle \to G^*$ satisface $\phi(\langle \{tat^{-1}\psi(a)^{-1} : a \in A\} \rangle_N) = Ker\ \pi$. Luego $\langle \{tat^{-1}\psi(a)^{-1} : a \in A\} \rangle_N = Ker\ \pi \circ \phi$. Por el primer teorema de isomorfía, se induce el isomorfismo $\xi : \frac{G*\langle t \rangle}{Ker\ \pi \circ \phi} \to G^*$ que hace conmutar el diagrama:

$$\begin{array}{ccc} G * \langle t \rangle & \xrightarrow{\pi} & G^* \\ \uparrow{\scriptstyle \phi} & & \uparrow{\scriptstyle \xi} \\ G * \langle t \rangle & \xrightarrow{\pi'} & \frac{G*\langle t \rangle}{ker\ \pi \circ \phi} \end{array}$$

A continuación veamos, a modo de curiosidad y sin trascendencia en lo que nos resta, dos ejemplos:

Ejemplo 5.3.1. *Si G es un grupo, $A = G$ y $\psi \in Aut(G)$, entonces $G^* \cong G \rtimes_\theta \mathbb{Z}$, donde $\theta : \mathbb{Z} \to Aut(G)$ es un homomorfismo tal que $\theta(1) = \psi$. Esto nos permite ver la extensión HNN como una generalización del producto semidirecto por $\mathbb{Z}$. Para ver esto, observemos que existe un homomorfismo $\phi : G * \langle t \rangle \to G \rtimes_\theta \mathbb{Z}$ tal que el siguiente diagrama conmuta:*

$$\begin{array}{ccccc} G * \langle t \rangle \cong G * \mathbb{Z} & & \xleftarrow{\iota_G} & & G \\ \uparrow{\scriptstyle \iota_{\mathbb{Z}}} & \overset{\phi}{\searrow} & & & \downarrow{\scriptstyle \sigma_G} \\ \mathbb{Z} & & \xrightarrow{\sigma_{\mathbb{Z}}} & & G \rtimes_\theta \mathbb{Z} \end{array},$$

donde $\iota_G : G \to G * \mathbb{Z}$, $\iota_{\mathbb{Z}} : \mathbb{Z} \to G * \mathbb{Z}$, $\sigma_G : G \to G \rtimes_\theta \mathbb{Z}$, $\sigma_{\mathbb{Z}} : \mathbb{Z} \to G \rtimes_\theta \mathbb{Z}$ son los monomorfismos canónicos. Es fácil ver que ϕ satisface que $g \mapsto (g, 0)$, para todo $g \in G$, y que $t^k \mapsto (e_G, k)$ para todo $k \in \mathbb{Z}$ (por la conmutatividad del diagrama). De esto se sigue que ϕ es un epimorfismo. Además $\langle tgt^{-1}\psi(g)^{-1} : g \in G \rangle_N \subset Ker(\phi)$, ya que:

$$\phi(tgt^{-1}\psi(g)^{-1}) = \phi(t)\phi(g)\phi(t)^{-1}\phi(\psi(g))^{-1} =$$

$$(e_G, 1)(g, 0)(e_G, -1)(\psi(g)^{-1}, 0) = (\psi(g), 1)(g^{-1}, -1) = (e_G, 0).$$

Por ende, queda bien definido el epimorfismo $\phi' : G^* \to G \rtimes_\theta \mathbb{Z}$ tal que $\phi'(\pi(\omega)) = \phi(\omega)$, y solo falta probar su inyectividad. Sea $g_0 t^{n_1} g_1 \cdots t^{n_m} g_m$ un elemento de $G * \langle t \rangle$ tal que $\phi'(\pi(g_0 t^{n_1} g_1 \cdots t^{n_m} g_m)) = (e_g, 0)$. Usando las relaciones que definen al cociente G^*, podemos reexpresar esto como $\phi'(gt^n) = (e_G, 0)$, donde $g \in G$ y $n \in \mathbb{Z}$. Luego $\phi(gt^n) = (e_g, 0) \Rightarrow (g, n) = (e_G, 0) \Rightarrow g = e_G, n = 0$, y por lo tanto $\pi(g_0 t^{n_1} g_1 \cdots t^{n_m} g_m) = e_{G^*}$. Entonces ϕ' es un isomorfismo.

Ejemplo 5.3.2. *La extensión HNN de $\mathbb{Z}$ respecto a los subgrupos $n\mathbb{Z}, m\mathbb{Z}$, es el grupo de Baumslag-Solitar $BS(m, n) = \langle a, b \mid ba^n = a^m b \rangle$, con $n, m \in \mathbb{N}$. Para ver esto, observar que el grupo $\mathbb{Z} * \langle b \rangle$ tiene por presentación $\langle a, b | \ \rangle \cong F(\{a, b\})$. Por otro lado, los grupos $n\mathbb{Z}$ y $m\mathbb{Z}$ son subgrupos de $\mathbb{Z}$ isomorfos por $f : n\mathbb{Z} \to m\mathbb{Z}$, definido por $f(nk) = mk$, $k \in \mathbb{Z}$. Luego el cociente de $\mathbb{Z} * \langle b \rangle$ entre la clausura normal de $\{ba^n b^{-1} a^{-m} : a \in \mathbb{Z}\}$, es por definición $\langle a, b \mid ba^n = a^m b \rangle$, lo que queríamos probar. En particular, notar que $B(1, 1)$ es isomorfo a $\mathbb{Z} \oplus \mathbb{Z}$ y, por último, algo no tan obvio es que $BS(1, -1)$ es isomorfo al grupo fundamental de la botella de Klein.*

La extensión HNN de G relativa a A, B, ψ es el grupo al que se hace referencia en el siguiente teorema:

Teorema 5.3.1. *(Higman/Neumann/Neumann)*
Sea un G un grupo y sean A, B subgrupos de G isomorfos entre si por vía de $\psi : A \to B$. Entonces existe un grupo G' tal que G se incrusta en G', y existe $t \in G'$ tal que $t^{-1}at = \psi(a)$, para todo $a \in A$.

Lo no trivial en la construcción del grupo G' es la incrustación en este de G, lo cual será probado más adelante en la sección como corolario del lema de Britton.

Nos toca introducir una última definición, respectiva a una forma canónica de escritura en las extensiones HNN. Para ello, consideremos la extensión HNN G^* de G respecto A, B, ψ. Elegimos un conjunto de representantes de las clases laterales derechas de A en G, T_A y de las clases laterales derechas de B en G, T_B —donde $1 \in T_A$ (respectivamente T_B) es el representante de la clase A (respectivamente B)—.

Definición 5.3.2. *Una forma normal es una sucesión finita* $(g_0, t^{\epsilon_1}, g_1, \cdots, t^{\epsilon_n}, g_n)$, *donde* $g_i \in G$, g_0 *es un elemento cualquiera del grupo,* $\epsilon_i \in \{-1, 1\}$, *y además:*

1. *Si* $\epsilon_i = -1$, *entonces* $g_i \in T_A$.
2. *Si* $\epsilon_i = 1$, *entonces* $g_i \in T_B$.
3. *No tiene una subsucesión de la forma* $(t^{\epsilon}, 1, t^{-\epsilon})$.

En primer lugar, notemos que una forma normal es una palabra reducida en $G \cup \{t\}$. Ahora, en la extensión HNN, se cumplen las relaciones:

$$t^{-1}at = \psi(a) \ (1).$$

De esto se obtiene $t^{-1}a = \psi(a)t^{-1}$. Como ϕ es isomorfismo, conjungando ambos lados de (1): $tbt^{-1} = \psi^{-1}(b)$, donde $b = \psi(a)$. Por lo tanto:

$$tb = \psi^{-1}(b)t$$

Esta "pseudoconmutatividad" es útil para el siguiente resultado, que nos permite dar una escritura única a los elementos de G^* con el concepto de forma normal.

Proposición 5.3.1. *Sea* $\pi : G * \langle t \rangle \to G^*$ *la proyección canónica. Entonces todo* $g \in G^*$ *puede escribirse de forma única como* $g = \pi(g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n)$, *donde* $(g_0, t^{\epsilon_1}, g_1 \cdots, t^{\epsilon_n}, g_n)$ *es una forma normal.*

Por la definición de forma normal se observa que $g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n$ es un elemento de $G * \langle t \rangle$ (si sucede que $g_0 = 1_G$ en la forma normal, entonces se considera que el elemento es $t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n \in G * \langle t \rangle$).

Idea de la demostración. Sean $g \in G^*$ y $g' \in \pi^{-1}(g)$. Primero, veamos que puede suponerse g' igual a $g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n$, donde $g_i \in G$ y $\epsilon_i \in \{-1, 1\}$. En principio, la forma general de g' será $g'_0 t^{k_1} g'_1 \cdots t^{k_n} g'_{n'}$, con $k_i \in \mathbb{Z}$. Sin embargo, si en la expresión de tal preimagen hubiera un $k_i > 1$, basta suponer $k_i = 2$ (el caso general es inductivo) y observar que

$$\pi(t^2) = \pi(t(t^{-1}at\psi(a)^{-1})t) = \pi(at\psi(a)^{-1}t), \ a \in A.$$

Esta igualdad es debida a las relaciones por las cuales cocientamos. De igual modo, si $k_1 < -1$, alcanza observar que:

$$\pi(t^{-2}) = \pi(t^{-1}\psi(a)t^{-1}a^{-1}), \ a \in A.$$

Al ser π un homomorfismo, se consigue que $\pi(g'_0 t^{k_1} g'_1 \cdots t^{k_n} g'_{n'}) = \pi(g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n)$, con $\epsilon_i \in \{-1, 1\}$ tal como queríamos. A su vez:

$$\pi(g') = \pi(g_0)\pi(t)^{\epsilon_1}\pi(g_1)\cdots\pi(t)^{\epsilon_n}\pi(g_n)$$

Ahora, implementamos un procedimiento que recorre de derecha a izquierda la palabra. Comenzamos con g_n: si $\epsilon_n = -1$, tomamos al representante en T_A de su clase lateral en G/A, g_n^A. Sabemos que existe $a_n \in A$ tal que $g_n(g_n^A)^{-1} = a_n$. De igual modo hacemos si $\epsilon_n = 1$: se tienen los elementos $g_n^B \in T_B$ y $b_n \in B$ tales que $g_n(g_n^B)^{-1} = b_n \in B$. Suponiendo el primer caso planteado ($\epsilon_n = -1$) cambiamos g_n por $a_n g_n^A$ en la expresión. Ya que $\pi(g_n) = \pi(a_n g_n^A) = \pi(a_n)\pi(g_n^A)$, se obtiene que:

$$\pi(g') = \pi(g_0)\pi(t)^{\epsilon_1}\pi(g_1)\cdots\pi(t)^{\epsilon_{n-1}}\pi(g_{n-1})\pi(t)^{-1}\pi(a_n)\pi(g_n^A)$$

Notar que $\pi(t)^{-1}\pi(a_n) = \pi(t^{-1}a_n) = \pi(\psi(a_n)t^{-1})$. Entonces:

$$\pi(g') = \pi(g_0)\pi(t)^{\epsilon_1}\pi(g_1)\cdots\pi(t)^{\epsilon_{n-1}}\pi(g_{n-1}\psi(a_n))\pi(t)^{-1}\pi(g_n^A)$$

Ahora, se revisa el valor de ϵ_{n-1} y luego realizamos el mismo procedimiento para $g_{n-1}\psi(a_n)$, según ϵ_{n-1} sea 1 o -1. Continuando de esta forma, llegaremos a una palabra en forma normal dentro de la preimagen de g, donde en su primer caracter (el símbolo libre de la definición de forma normal) se han acumulado los "residuos"que van quedando del proceso (como fue $\psi(a_n)$ para g_n). Si en algún punto del procedimiento nos encontramos con un caracter de A o B y lo debemos cambiar por su representante, tendríamos que colocar el 1. Como esto no puede dejarse así (o nos saldríamos del producto libre) se hacen colapsar t^{-1} y t^1, que quedan a la par, y todo lo que sigua y se deba cancelar. Después, reiniciamos el proceso.

Puesto que T_A, T_B se consideran fijados, una vez construida una preimagen de g en forma normal, la unicidad se desprenderá inmediatamente de la unicidad de los representantes elegidos. □

La **escritura canónica** (así la llamaremos a partir de ahora) recién presentada de cada elemento de la extensión nos servirá esencialmente para el próximo resultado:

Lema 5.3.2. *(Lema de Britton) Supongamos que una palabra $\omega = g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n$ en $G * \langle t \rangle$, con $n > 1$ y $\epsilon_i \in \{1, -1\}$, no tiene subpalabras de la forma $t^{-1}at$, o tbt^{-1}, donde $a \in A$, $b \in B$. Entonces $\pi(\omega) \neq 1_{G^*}$.*

Antes de la demostración vamos a ver algunas propiedades importantes.
La unicidad dada por la propsición anterior nos permite definir la longitud de $\pi(\omega)$, donde ω es una palabra en $G * \langle t \rangle$, como la longitud de la forma normal que define a la escritura canónica. Ahora, supongamos que $\omega = h_0 t^{\epsilon_1} h_1 \cdots t^{\epsilon_n} h_n$, $h_i \in G$, $n > 1$, $\epsilon_i \in \{1, -1\}$, es una palabra en $G * \langle t \rangle$ de longitud $2n + 1$ con subpalabras de la forma $t^{-1}at$ o tbt^{-1} ($a \in A$, $b \in B$). Entonces la escritura

canónica de $\pi(\omega)$ tiene una longitud menor a $2n+1$. En efecto, si $\omega = gt^{-1}ath$ con $g, h \in G * \langle t \rangle$ y $a \in A$, luego:

$$\pi(gt^{-1}ath) = \pi(g)\pi(t^{-1}at)\pi(h) = \pi(g)\pi(\psi(a))\pi(h) = \pi(g\psi(a)h).$$

Esto significa que podemos encontrar una preimagen de $\pi(\omega)$ con una longitud menor a la de ω pero escrita también en forma $h'_0 t^{\epsilon_1} h_1 \cdots t^{\epsilon_n} h'_{n'}$, $h'_i \in G$, $n' > 1$, $\epsilon_i \in \{1, -1\}$. Ahora, debemos notar en la demostración de la proposición 5.3.1 que aplicar el proceso que convierte una expresión como la anterior a una forma normal con igual imágen según π, hace que la longitud de la forma normal que devuelve sea menor o igual a $2n'+1$. Por ende, como hemos visto que $n' < n$, la forma normal asociada a la expresión canónica de $\pi(\omega)$ tendrá una longitud menor a $2n+1$.

Recíprocamente, si ω cumple las condiciones del lema, entonces la longitud de la escritura canonica de $\pi(\omega)$ es igual a la de ω. Esto es debido a que en el proceso esbozado en la proposición 5.3.1, al no haber $g_i \in A$ o $g_i \in B$, nunca se alcanza una instancia en que algún símbolo de G a intercambiar por su representante en T_A o T_B sea tal que se deba colocar 1. Luego, la longitud de la palabra no se recorta.

Demostración. Como $\pi(\omega)$ es un elemento de G^*, se pude expresar en su forma normal, $\pi(w) = \pi(g'_0 t^{\epsilon_1} g'_1 \cdots t^{\epsilon_{n'}} g'_{n'})$. Por las observaciones anteriores,$\pi(\omega)$ tiene longitud $2n+1$; sin embargo 1_{G^*} tiene una escritura canónica de longitud 1, que es $\pi(1)$. Por la unicidad de la formal normal, $\pi(\omega) \neq 1_{G^*}$. □

El siguiente corolario del lema de Britton es una propiedad basica de las extensiones HNN y es lo que nos faltaba probar del teorema 4.3.1.

Corolario 5.3.1. *El homomorfismo canónico $\phi : G \to G^*$ que lleva cada elemento de G a su clase de equivalencia en G^*, es inyectivo.*

Demostración. Tomando la función $\iota : G \to G * \langle t \rangle$ tal que $g \mapsto (g, 1, 1, \dots)$ y π la proyección canónica, luego $\phi(g) = \pi(\iota(g))$. Ahora, si $g \in G$ es no trivial, entonces $\psi(g) = \pi((g, 1, 1, ...))$ y esa es la expresión de la proposición 5.3.1, la cual no tiene subpalabras de la forma $\pi(t)^{-1}\pi(a)\pi(t)$, con $a \in A$ o $\pi(t)\pi(b)\pi(t)^{-1}$, con $b \in B$. Por el lema de Britton, $\phi(g) \neq 1_{G^*}$. Luego $ker\phi = \{1_{G^*}\}$. □

Capítulo 6

Indecidibilidad del problema

Resolver el problema de la palabra para todos los grupos puede ser tan imposible como resolver todos los problemas matemáticos.

Max Dehn

Finalmente, esta sección culminará el trayecto hasta aquí transitado, justificando las herramientas que fuimos desarrollando. En efecto, demostraremos la existencia de un grupo para el cual el problema de la palabra es indecidible. Por ende, el problema de la palabra, en general, lo es.

6.1. Teorema de Novikov-Boone

Para probar la insolubilidad, explotaremos nuestro conocimiento sobre problemas no decidibles ya conocidos. Construiremos un grupo tal que el problema de la palabra en él sea equivalente a la decidibilidad de algún conjunto indecidible (no recursivo).

Definición 6.1.1. *Sea G un grupo con una presentación $\langle S|R\rangle$. Decimos que G es:*

1. *Finitamente generado (denotaremos f.g.) si el conjunto S es finito.*
2. *Finitamente presentado (denotaremos f.p.) S y R son finitos.*
3. *Recursivamente generado si S es Turing-reconocible.*
4. *Recursivamente presentado si S y R son Turing-reconocibles.*

Teorema 6.1.1 (Teorema de la incrustación de Higman)**.** *Un grupo finitamente generado G es recursivamente presentado si y sólo si se incrusta en algún grupo finitamente presentado.*

No presentaremos una demostración de este teorema, pues pese a ser necesario en lo que sigue, la prueba es técnica y excede por mucho las definiciones brindadas aquí. Puede verse en [18] y [19].

Por el teorema 4.5.2 y el corolario 4.5.1, el conjunto $HALT_{MT}$ es indecidible pero Turing reconocible. La idea es usar tales propiedades a nuestro favor, codificándolo primero en un subconjunto de $\mathbb{N}$:

1. Podemos pensar a $HALT_{MT}$ ya expresado en el alfabeto $\{1, |\}$ (ver la observación posterior al corolario 4.4.2). Luego, cada elemento de $HALT_{MT}$ es una cadena de la forma $1^{n_1}|^{n_2}\cdots|^{n_{m-1}}1^{n_m}$, donde $s^k = s\cdots s$ k-veces, siendo $s \in \{1, |\}$.

2. Ahora consideramos la asignación $1^{n_1}|^{n_2}\cdots|^{n_{m-1}}1^{n_m} \mapsto \Pi_{i=1}^{m} p_i^{n_i}$, donde p_i es el i-ésimo primo. Tal codificación esta bien definda, y es una inyección.

Denotaremos $\langle HALT_{MT}\rangle$ al conjunto $HALT_{MT}$ codificado.

Finalmente, vamos a enunciar y probar el que quizás sea el teorema principal del texto. El mismo fue demostrado en 1955 por el matemático sovietico Pyort Novikov, seguido por el matemático estadounidense William Boone, quien presentó una demostración diferente en 1958.

Teorema 6.1.2 (Teorema de Novikov-Boone)**.** *Existe un grupo finitamente presentado tal que el problema de la palabra en él es indecidible.*

Presentaremos una demostración que utiliza el teorema de incrustación de Higman, el cual fue demostrado (por el mismo Higman) en 1961 [20]. Por ende, dicha demostración es diferente y posterior a la de Boone.

Demostración. Tomamos los grupos libres sobre los conjuntos disjuntos $\{a, b\}$ y $\{c, d\}$. Ambos grupos resultan isomorfos, vía el isomorfismo inducido por el mapa $a \to c, b \to d$. Ahora consideremos G, el producto libre $F(a, b) * F(c, d) \cong F(a, b, c, d)$ con la amalgamación de los subgrupos $\langle\{a^{-s}ba^s | s \in \langle HALT_{MT}\rangle\}\rangle$ y $\langle\{c^{-s}dc^s | s \in \langle HALT_{MT}\rangle\}\rangle$ (estos claramente isomorfos por la restricción del isomorfimo antes referido). Notemos que por la definición de producto libre amalgamado, G tiene la siguiente presentación:

$$\langle a, b, c, d | a^{-s}ba^s = c^{-s}dc^s, s \in \langle HALT_{MT}\rangle\rangle$$

Luego, vamos a considerar la siguiente palabra en $\langle a, b\rangle * \langle c, d\rangle$: $\omega = a^{-t}ba^tc^{-t}d^{-1}c^t$. Observemos que:

$$\pi(\omega) = 1_G \Leftrightarrow \pi(a^{-t}ba^tc^{-t}d^{-1}c^t) = 1_G \Leftrightarrow \pi(a^{-t}ba^t) = \pi((c^{-t}d^{-1}c^t)^{-1}) = \pi(c^{-t}dc^t)$$

$$\Leftrightarrow t \in \langle HALT_{MT} \rangle \ (1)$$

Por ende, de existir un máquina de Turing M que decida al conjunto $\{\omega \textit{ palabra en } \langle a, b\rangle * \langle c, d\rangle \mid \pi(\omega) = 1_G\}$, la misma operaría como sigue:

$$M(\langle\omega\rangle) = \begin{cases} \text{acepta} & \text{si } \pi(\omega) = 1_G \\ \\ \text{rechaza} & \text{si } \pi(\omega) \neq 1_G \end{cases}$$

Pero entonces, definimos la máquina M' tal que:

$$M'(\langle n\rangle) = \begin{cases} \text{acepta} & \text{si } M \text{ acepta } \langle a^{-n}ba^nc^{-n}d^{-1}c^n\rangle \\ \\ \text{rechaza} & \text{si } M \text{ no acepta } \langle a^{-n}ba^nc^{-n}d^{-1}c^n\rangle \end{cases}$$

Luego, por (1), M' decide el conjunto $\langle HALT_{MT}\rangle$ lo cual es un absurdo. Por lo tanto, el problema de la palabra en G es algoritmicamente indecidible.

Notar que G no es finitamente presentado, pero como $\langle HALT_{MT}\rangle$ es Turing-reconocible (i.e. recursivamente enumerable), G es recursivamente presentado. Entonces, por el teorema de la incustación de Higman (6.1.1), existe un grupo finitamente presentado con una copia isomorfa a G (la incrustación de este último). Por lo tanto, dicho grupo también cumple con que el problema de la palabra sobre él es indecidible. □

En la demostración anterior se construyó un grupo con la caracterisitica deseada utilizando la indecidibilidad del conjunto $HALT_{MT}$, pero podemos dar un ejemplo mucho más concreto de un grupo finitamente generado con un problema de la palabra indecidible. El siguiente ejemplo fue presentado por el matemático Donald J. Collins en 1986:

Ejemplo 6.1.1. *El grupo G con la siguiente presentación:*

$$\begin{array}{l} \langle a, b, c, d, e, p, q, r, t, k| \\ \end{array}$$
$$\begin{array}{lll} p^{10}a = ap, & pacqr = rpcaq & ra = ar \\ p^{10}b = bp, & p^2adq^2r = rp^2daq^2, & rb = br, \\ p^{10}c = cp, & p^3bcq^3r = rp^3cbq^3, & rc = cr, \\ p^{10}d = dp, & p^4bdq^4r = rp^4dbq^4, & rd = dr, \\ p^{10}e = ep, & p^5ceq^5r = rp^5ecaq^5, & re = er, \\ aq^{10} = qa, & p^6deq^6r = rp^6edbq^6, & pt = tp, \\ bq^{10} = qb, & p^7cdcq^7r = rp^7cdceq^7, & qt = tq, \\ cq^{10} = qc, & p^8ca^3q^8r = rp^8a^3q^8, & \\ dq^{10} = qd, & p^9da^3q^9r = rp9a^3q^9, & \\ eq^{10} = qe, & a^{-3}ta^3k = ka^{-3}ta^3s & \rangle \end{array}$$

cumple con que el problema de la palabra en él es indecidible. La demostración puede verse en [21].

6.2. Propiedades de Markov

La insolubilidad del problema de la palabra es la semilla de una gran familia de resultados de indecidibilidad en estructuras algebraicas. Ya hemos estado advirtiendo, desde la introducción misma, que una familia grande y laxa de propiedades en grupos finitamente presentados —propiedades que casi siempre deseamos decidir sobre un grupo dado—, conforma otro resultado de indecibilidad. Con más precisión, el problema de poder contestar afirmativa o negativamente si alguna de estas propiedades vale en un grupo dado, es en general indecidible.

En esta sección revisaremos brevemente el teorema que demuestra estas afirmaciones, derrumbando por fin la esperanza de obtener un procedimiento efectivo para determinar la validez de, al menos, un importante cúmulo de propiedades interesante en la teoría de grupos.

Definición 6.2.1. *Una propiedad de Markov P, para grupos finitamente presentados, es tal que:*

1. *Se preserva bajo isomorfismo. Esto es, si un grupo G es isomorfo a otro H y tiene la propiedad P, entonces H también tiene la propiedad P.*
2. *Existe un grupo f.p. que tiene la propiedad P.*
3. *Existe un grupo f.p. que no puede incrustarse como subgrupo en nigún grupo f.p. con la propiedad P.*

Aunque pueda no parecer a simple vista, muchas de las propiedades elementales que nos interesan de los grupos f.p. (y de los grupos en general) son propiedades de Markov. Por solo mencionar algunas:

1. Ser trivial.
2. Ser finito (pues se conserva por isomorfismo, $\mathbb{Z}_2$ es f.p y finito, pero $\mathbb{Z}$ es f.p y no se incrusta en ningún grupo finito).
3. Ser abeliano.
4. Ser f.g. y libre.
5. Ser tal que el problema de la palabra sea decidible. En efecto, el teorema de Novikov-Boone asegura la existencia de un grupo f.g. sin esta propiedad. Por ende, no puede incrustarse en ningún otro que si la tenga. Además, la propiedad claramente se preserva por isomorfismo y, como hemos probado en el segundo capítulo, numerosos grupos f.g. la cumplen.

Ahora, dada P un propiedad de Markov y G un grupo f.p. es posible hacerse la misma pregunta que encomendó todo nuestro trabajo previo con el problema de la palabra. La demostración del siguiente teorema es constructiva e inteligente, y utiliza prácticamente todo lo que hemos definido en este trabajo. La respuesta que brinda es sorprendentemente negativa.

Teorema 6.2.1. *(Adian-Robin) Sea P una propiedad de Markov. Entonces, el conjunto de todos los grupos finitamente presentados que satisfacen P es indecidible. Es decir, dadas una presentación finita $\langle X, R\rangle$ y una propiedad de Markov P, decidir si la presentación tiene la propiedad P es un problema algorítmicamente insoluble.*

Demostración. Por definición, existen grupos G y H f.p. tales que G satisface P, y H no se incrusta en ningún grupo f.p. que cumpla P. Sea K un grupo f.p. tal que el problema de la palabra en él es indecidible. Entonces $K * H$ es f.p. y el problema de la palabra sobre él también resulta indecidible; supongamos que $\langle x_1, \dots, x_n | R(x_1, \dots, x_n)\rangle$ es una presentación de $K * H$, y sea $\langle p\rangle$ un grupo ciclico infinito. Consideremos los siguientes grupos:

1. $G_1 = (K * H) * \langle p\rangle = \langle q_0 = p, q_1 = px_1, \dots, q_n = px_n | R'\rangle$, donde $R' = R(p^{-1}q_0, \dots, p^{-1}q_n)$ (recordar que $\langle X_1|R_1\rangle * \langle X_2|R_2\rangle = \langle X_1 \sqcup X_2 | R_1 \sqcup R_2\rangle$).

2. Ahora vamos a tomar las $n+1$ extensiones HNN sucesivas de G_1 relativas a los subgrupos $\langle q_0\rangle, \dots, \langle q_n\rangle$ y $\langle q_0^2\rangle, \dots, \langle q_n^2\rangle$, haciéndoles corresponder las letras estables $r_0, \dots, r_n$. Procedemos de forma sucesiva, finalizando con el grupo $G_2 = \langle G_1, r_0, \dots, r_n | r_i q_i^2 r_i^{-1} = q_i : i \in \{1, \dots, n\}\rangle$.

3. Aplicando el lema de Britton (Lema 4.3.2) a G_2 obtenemos que los subgrupos $\langle r_0, \dots, r_n\rangle$ y $\langle r_0^2, \dots, r_n^2\rangle$ son libres sobre los generadores. En efecto, de suceder que alguna palabra en $\{r_0, \dots r_n\}$ tenga su proyección en G_2 igual a 1, por el lema de Britton tal expresión debería tener un caracter formal de $\langle q_n\rangle$; pero este no puede formarse con los símbolos $r_0, \dots, r_n$. Por ende, son isomorfos. Tomamos así la letra estable s y construimos las extensión HNN de G_2 respecto a los subgrupos anteriores: $G_3 = \langle G_2, s | s r_i s^{-1} = r_i^2 : i \in \{1, \dots, n\}\rangle$ (notar que las relaciones descriptas son suficientes).

4. Ahora elegiremos el grupo cíclico infinito $\langle a\rangle$ y su extensión HNN con respecto a él mismo y a $\langle a^2\rangle$, con la letra estable b. Obtenemos $G_4 = \langle a, b | bab^{-1} = a^2\rangle$.

5. Tomamos la extensión HNN de G_4 respecto a los subgrupos $\langle b\rangle$ y $\langle b^2, \rangle$, con la letra estable c. Esto resulta en el grupo $G_5 = \langle G, c | cbc^{-1} = b^2\rangle$.

6. Vamos a definir el penúltimo grupo: dada ω una palabra en K, definimos el grupo $G_6 = \langle G_3 * G_5 | sa^{-1}, \omega q_0 \omega^{-1} q_0^{-1} c^{-1}\rangle$.

7. El último grupo será $G_7 = G_6 * G$

Si la palabra ω en K es trivial, entonces de las relaciones de G_6 se seguría que cada generador es trivial: $\omega = 1 \Rightarrow c = 1 \Rightarrow b = 1 \Rightarrow a = 1 \Rightarrow s = 1 \Rightarrow \forall i,\ r_i = 1 \Rightarrow \forall i,\ q_i = 1$ (todo esto en G_7). Por lo tanto, G_6 sería trivial. Luego $G_7 \cong G$, con lo cual G_7 tiene la propiedad P. Si ω no es trivial en K, entonces $a, b, \omega q_0 \omega^{-1} q_0^{-1}$ tienen orden infinito en G_5. Entonces los grupos $\langle a, c\rangle$ y $\langle s, \omega q_0 \omega^{-1} q_0^{-1}\rangle$ son copias

del grupo libre sobre dos elementos en G_5 y G_3 respectivamente. Por lo tanto, son isomorfos por vía de ψ, fruto de extender los mapas : $a \mapsto s,\ c \mapsto \omega q_0 \omega^{-1} q_0^{-1}$. Así $G_6 \cong G_3 *_\psi G_5$. Es decir, podemos ver a G_6 como el producto amalgmado de G_3 y G_5. De este modo G_3 se inscrusta en G_6, y este último lo hace también en G_7. A la vez, H se inscrusta en G_3; por ende, también en G_7. Esto significa que, por construcción, G_7 tampoco la cumple.

Por lo tanto, ω es trivial en K si y solo si G_7 satisface P. De existir así una máquina de Turing que decida si G_7 tiene la propiedad P o no, entonces esa máquina decide el problema de la palabra en K, absurdo. □

Habiendo logrado probar este teorema, cuya demostración (como dijimos recientemente) justifica casi todo nuestro esfuerzo, brindaremos un último y breve resultado. Poco hemos hablado, más que en los primeros capítulos, de uno de los problemas hermanos al de la palabra. Este es el problema del isomorfismo, descripto también por Dehn en 1911, cuya suerte probablemente ya nos esperamos.

Corolario 6.2.1. *El problema del isomorfismo no es decidible.*

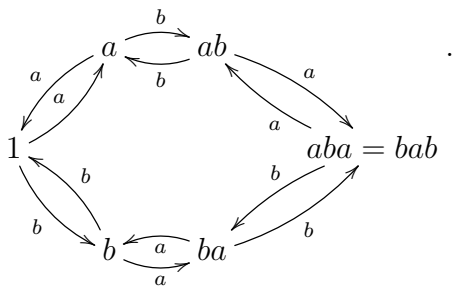
Demostración. Ciertamente, dado un grupo G finitamente presentado, la propiedad $P(G)$ de *ser isomorfo a* G es de Markov. Luego, por el teorema de Adian-Robin, existirá un par de grupos finitamente presentados $G,\ H$, tales que $P(G)$ en H sea indecidible. □

Capítulo 7

Conclusiones

Hemos indicado ya que el motivo de este trabajo es, por una parte, plantear sólidamente lo que fue (y aún es) un problema significativo e influyente matemática e históricamente, para ir luego tejiendo el camino hacia su *solución* —sin ánimos de ironía—. Por otra parte, y a la par, se pretendía dar un acercamiento a los temas necesarios para esto último, particularmente los relacionados a la teoría de computabilidad, dando la posibilidad al lector de profundizarlos por su cuenta. Hicimos énfasis además en el modo en que la historia del problema atraviesa y requiere del desarrollo de la computabilidad, tomando así una ubicación especialmente importante.

Sin embargo, también destacamos que el ambiente natural de este problema induce a pensar la matemática desde un lugar algo distinto al que generalmente se adopta cuando pensamos en objetos concretos de la misma. Con más precisión, nos referimos en esto último a la diferencia de índole filosófica (y discutible) entre los «objetos matemáticos» en si mismos, que nos interesan entender y describir, y las expresiones simbólicas formales con las que los representamos. Estas últimas, desprovistas de carácter semántico y solo regidas por ciertos axiomas, son objetos fundamentales en la lógica. La distinción que planteamos puede parecer una cuestión meramente filosófica y de poco interés matemático; sin embargo, puede interponerse no solo en nuestro entendimiento (y más aún, desarrollo) de la materia, si no en el ejercicio matemático común. Podríamos demostrar que la presentación $\langle a,\ b \mid a^2,\ b^2,\ (ab)^3 \rangle$ del grupo simétrico $\mathbb{S}_3$ es finita y acotada en su cardinal por 6, mediante la manipulación formal de sus elementos; por ejemplo, dando un algoritmo que recorra sus palabras, partiendo de 1, añadiendo generadores y reduciendo conforme indiquen las relaciones:



.

Alguien, por otra parte, puede visualizar al grupo actuando como permutaciones, y deducir tras un breve conteo la respuesta. No queda claro que estemos hablado de lo mismo, sin embargo. La segunda solución implica interpretar a la estructura de la presentación como el grupo concreto en cuestión; la segunda es una prueba sobre la propia estructura en si misma. Asuntos de este estilo rodean el ejercicio matemático en estructuras formales —al menos para quienes encontramos interesante plantearlos—, dando una cierta sensación de vértigo.

En otro sentido, la severidad de afirmar que existen problemas indecidibles ha causado temblores bien justificados a niveles filosóficos y matemáticos, y como ya hemos dicho, el problema de la palabra encarna justamente eso. Ni siquiera es fácil pensar en lo que significa la existencia de un problema no resoluble de manera algorítmica. Casi parece que todo ejercicio lógico-matemático humano puede capturarse mediante un procedimiento efectivo (al menos en lo que concierne a una solución matemática por si misma, por fuera del proceso mental que implica), y en esta intuición radica también la fuerte evidencia a favor de la tesis de Church-Turing.

Todo lo anterior sumado al especial sabor del álgebra, protagonista en el problema, hacen de este no solo entretenido e interesante, si no misterioso —como muchas cuestiones que entran en el terreno de la lógica matemática, en nuestra opinión—. De este modo, en última instancia, más allá de informar (y en el mejor de los casos, enseñar), esperamos haber capturado y transmitido ese interés, y ese misterio.

Bibliografía

[1] Thomas W. Hungerford. *Algebra*. Springer-Verlag New York, Inc. 1974.

[2] Derek J.S. Robinson. *A Course in the Theory of Groups*. 2da edición. Springer-Verlag New York, Inc. 1996.

[3] Max Dehn. *Über unendliche diskontinueierliche Gruppen*. Mathematische Annalen 71. 116-144.

[4] .Dehn, M *Transformation der Kurven auf zweiseitigen Flächen*. Math. Ann. 72, 413–421. 1912.

[5] Heinrich Tietze. *Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten*. Monatshefte für Mathematik und Physik, 19, pp. 1–118. 1908.

[6] John Stillwell. *Emil Post and His Anticipation of Gödel and Turing*. Mathematics Magazine, 77 (1): 3–14. 2004.

[7] Emil L. Post. *A variant of a recursively unsolvable problem*. Bull. Amer. Math. Soc. 52 (4): 264–269. 1946.

[8] Martin Davis. *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*. pag 292, Raven Press, New York. 1965.

[9] Emil L. Post. *Recursive unsolvability of a problem of Thue*. The Journal of Symbolic Logic, Vol. 12, No. 1. pp. 1-11. 1947.

[10] Alan M. Turing. *The Word Problem in Semi-Groups With Cancellation*. The Annals of Mathematics 52 (2): 491–505. 1950.

[11] Charles F. Miller. *Turing machines to word problems*. Turing's Legacy: 330.

[12] S. I. Adian. *Mathematical logic, the theory of algorithms and the theory of sets*. AMS Bookstore, 1977.

[13] Michael Sipster. *Introduction to the Theory of Computation*. 3era edición, Cengage Learning. 2013.

[14] E. Alfonseca Cubero - M. Alfonseca Moreno - R. Moriyón Salomon. *Teoría de Autómatas y Lenguajes Formales*. 1era edición, McGraw-Hill. 2007.

[15] Alan M. Turing. *On Computable Numbers, with an Aplication to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, s2-42, 230-265. 1936.

[16] Alonzo Church. *A note on the Entscheidungsproblem*. Journal of Symbolic Logic, 40-41. 1936.

[17] Alonzo Church. *Review of Turing 1936*. Journal of Symbolic Logic, 42-43. 1937.

[18] Alexander Burka y Andrew H. Wallace. *Higman's Embedding Theorem and Decision Problems*. 2019.

[19] Stal Aanderaa. *A Proof of Higman's Embedding Theorem, Using Britton Exensions of Groups.*. North Holland Publishing Company. 1973.

[20] Graham Higman. *Subgroups of finitely presented groups*. Proceedings of the Royal Society. Series A. Mathematical and Physical Sciences. vol. 262 pp. 455-475. 1961.

[21] Donald J. Collins. *A simple presentation of a group with unsolvable word problem.* Illinois Journal of Mathematics, 1986.