



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO DE CIENCIAS
MATEMÁTICAS

ESCUELA DE CIENCIAS

Variedades, Grupos y Máquinas de Turing

T E S I S

QUE PARA OBTENER EL GRADO ACADÉMICO DE

MAESTRO EN CIENCIAS

P R E S E N T A :

RODOLFO CONDE MARTÍNEZ

DIRECTOR DE TESIS:

DR. MARCELO ARMENTO AGUILAR GONZÁLEZ DE LA VEGA

MÉXICO, D. F.

ABRIL 2008



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS
MATEMÁTICAS
FACULTAD DE CIENCIAS

Variedades, Grupos y Máquinas de Turing

T E S I S

QUE PARA OBTENER EL GRADO ACADÉMICO DE

MAESTRO EN CIENCIAS

P R E S E N T A :

RODOLFO CONDE MARTÍNEZ

DIRECTOR DE TESIS:
DR. MARCELO ALBERTO AGUILAR GONZALEZ DE LA VEGA

MÉXICO, D.F.

ABRIL 2008.

Variedades, Grupos y Máquinas de Turing

Rodolfo Conde Martínez
rcm@gmx.co.uk

Abril del 2008

A Dios Padre, Mi Señor Jesucristo, y el Espíritu Santo. Porque mi Dios me dió la fuerza para atravesar este largo y duro camino, y me ha permitido llegar aquí de nuevo.

Para mi Papa, que me apoya siempre en las decisiones que tomo, y me ha alentado a cambiar mi rumbo hacia el emocionante camino de la ciencia.

Para mi Mama, que no aparta su apoyo de mi y aún me sigue esperando cada día.

Por siempre...Gracias.

*Te haré entender, y te enseñaré el
camino en que debes andar;
Sobre ti fijaré mis ojos.*

Salmo 32.8

*Encomienda a Jehová tu camino,
Y confía en él; y él hará.*

Salmo 37.5

En Dios siempre hay esperanza.

Agradecimientos

A Dios mi Señor, por encima de todo y todos. Él sabe lo mucho que me gusta todo esto, y me ha dado la oportunidad de seguir haciéndolo, y fuerzas para soportar los momentos difíciles.

A mis padres, porque siguen apoyándome en muchas formas, y su comprensión no se aparta de mí. Siempre puedo contar con ellos.

A mi abuelita Aurita (Áurea del Águila), que me cuidó durante mi niñez. Siempre preocupada porque comiera bien, me levantara para ir a la primaria, me lavara los dientes, me acostara temprano...en fin. Y mi Tío Pipe (José Guadalupe Conde del Águila), en todo momento me mostró cariño, me compraba juguetes, me abrazaba, me besaba....Nunca olvidaré el tiempo que pasé junto a ellos en Teapa, Tabasco.

A Chendy, por el gran apoyo que me brindó cuando buscaba entrar a la UNAM, acompañándome a la inscripción del examen de admisión cuando no sabía como moverme en el D.F., y porque siempre me recibe con cariño y buen trato cuando voy a Villahermosa a visitar a mi familia.

A toda mi familia, mis herman@s (Perla, Thalía, Viridiana y Carlos). Mis tí@s, prim@s, padrinos, madrinas, uff...son tantos...tod@s me han apoyado en una forma u otra, en alguna etapa de mi vida...verdaderamente !!!

A Karla Verónica García, por todos los momentos y experiencias inolvidables que pasamos juntos.

A mis amig@s, desde la primaria hasta el día de hoy. Con tod@s ell@s, siempre he tenido amistad, comprensión, diversión, tristezas, solidaridad, unión...De todo !!! Mis hermanos (Mike, Marcos, Carlos, Paco, los clásicos, y la mas reciente adición: Emmanuel, con quien estoy muy agradecido por ayudarme con unas imágenes en 3D para el último capítulo.).

A mi tutor, el Dr. Marcelo Aguilar, por la enorme paciencia y comprensión que me tuvo. Porque se aventuró a recibir como su estudiante a un computólogo, deseoso de aprender matemáticas (en especial topología).

A mis sinodales, por el valioso tiempo que me dedicaron a mi y a esta tesis, sus atenciones, consejos y correcciones.

A los matemáticos, porque, a pesar de mi ignorancia y mis deficiencias, me recibieron con los brazos abiertos, jamás fui rechazado, al contrario, me ayudaron y me enseñaron. Todo lo contrario de cuando un matemático, va de la misma forma, al mundo de los computólogos (y esto es muy triste).

A toda la banda de SaitoSoft: Gaby (aka Dhi), Sarai, Emmanuel (aka Galphord), Sergio (aka Checo), Germán (aka Germen), Elvia (aka Elvis), Henry, Walter, Juan Carlos. En ellos encontré, aparte de decididos compañeros de trabajo, entrañables nuevos amigos. Tengo que darle gracias especiales a Galphord, por sus consejos que llegaron en un momento de varios caminos en mi vida; y a Germen, por ayudarme con un asunto secreto que pocos conocen jajaja.

A Luis Humberto Alva Martínez y Mateo Saito Hata, de la empresa SaitoSoft, quienes fueron los primeros (y últimos) en darme la oportunidad de trabajar medio tiempo, para sustentarme mientras estudiaba mi maestría (gracias por la laptop).

A Enrique Landa Sotelo y Miguel Torres Pineda, de la empresa BCM, que continuaron dándome la oportunidad de trabajar mientras terminaba de escribir esta tesis. Es muy importante que las empresas concedan este tipo de oportunidades, para que la gente se anime a seguir su preparación, para aspirar a lograr mejorar su vida.

A todas las personas, alrededor del mundo entero, que colaboran en los sitios web de Wikipedia, *la enciclopedia libre* (<http://www.wikipedia.org/>; libre, de libertad de expresión, no de cerveza gratis), Mathworld (<http://mathworld.wolfram.com/>) y PlanetMath (<http://planetmath.org/>). Tres excelentes sitios web, donde siempre encontré información y referencias iniciales de muchas cosas cubiertas en esta obra. Son un excelente punto de inicio para buscar un concepto que uno no conoce. Los recomiendo ampliamente, e invito a todos a cooperar con ellos, para que mejoren aún más.

A mi Alma Mater, la Universidad Nacional Autónoma de México, el Instituto de Matemáticas, y la Facultad de Ciencias, por ser mis segundos hogares durante mi maestría. Como en la licenciatura, fue un rato inolvidable, de lujo.

A todos ustedes, que han hecho algo por mi, de verdad les digo...

Muchas Gracias.

Índice general

Introducción	IX
Índice general	XV
Índice de figuras	XVII
1. Teoría de la computación	1
1.1. Antecedentes históricos	1
1.2. Definiciones básicas	5
1.3. Modelos equivalentes	20
1.4. Universalidad, autorreferencia e indecidibilidad	26
1.4.1. La Máquina Universal de Turing	26
1.4.2. Problemas no computables	29
1.5. Grados de Turing: Clasificando lo imposible	34
1.5.1. Máquinas oráculo	34
1.5.2. El universo de Turing	37
1.5.3. Grados de Turing computables enumerablemente	38
2. Álgebra	41
2.1. Presentaciones de grupos	41
2.1.1. La relación vacía de Markov	44
2.1.2. Operaciones en presentaciones	45
2.2. Computabilidad en grupos	51
2.3. El método de decisión de Tarski	54
3. Topología	57
3.1. Topología combinatoria	57
3.1.1. Variedades combinatorias	58

3.1.2.	La conjetura Hauptvermutung	60
3.1.3.	Variedades diferenciables y/o combinatorias	60
3.2.	Asas y cirugía de Morse	61
3.2.1.	Descomposición en asas	62
3.2.2.	Cirugía de Morse	63
3.3.	Computabilidad en espacios	64
4.	Presentación finita de variedades	89
4.1.	Variedades C^∞ -algebraicas	90
4.2.	Codificación de variedades	92
5.	Insolubilidad de la equivalencia	103
5.1.	Vecindades de asa	104
5.2.	Construcción de F_n	109
5.3.	Resultados finales	117
	Bibliografía	125

Índice de figuras

1.1. Máquina de Turing determinística. El símbolo “+” indica a la máquina que no puede seguir avanzando a la izquierda, mientras que los símbolos después del último 0 indica que la celda correspondiente está vacía.	6
1.2. Máquina de Turing de cintas múltiples.	21
1.3. Máquina de Turing de cinta infinita en ambos lados.	22
1.4. Ejecución de una Máquina de Turing no determinística.	23
1.5. Máquina Enumeradora. Representamos la cinta de salida, donde se enumeran las palabras, por un monitor de computadora (es muy usual representarla por una impresora).	24
1.6. La Máquina Universal de Turing. Nótese el uso de los metasímbolos $M\#x$ y x , para denotar la codificación en el alfabeto $\{0, 1\}$ de la máquina M y la palabra x	29
3.1. Complejo simplicial orientado.	65
3.2. Aplicación del algoritmo del Corolario 3.3.15. Por simplicidad, en (3) y (4), hemos omitido los vértices y las aristas interiores en los triángulos sombreados.	82
3.3. Árbol máximo del complejo K , para calcular una presentación finita de $\pi_1(K)$, usando el algoritmo dado por el Corolario 3.3.13.	84
5.1. Vecindad de asa del 1-simplejo estándar (la línea verde).	106
5.2. Vecindad de asa de un complejo simplicial, indicado por las líneas de color rojo.	107

Introducción

La Topología y la Teoría de la Computación, son dos ramas de la matemática, que aparentan ser muy distantes, debido a que son de naturaleza distinta. Mientras que la topología se aplica ampliamente en el ámbito continuo, la teoría de la computación vive dentro de lo discreto, lo combinatorio; parecería que no hay forma de que estas ramas estrechen lazos y se combinen, para encontrar nuevos conocimientos matemáticos.

Sin embargo, en los últimos años, este mito parece empezar a desvanecerse por completo, y decimos que es un mito, porque desde mediados del siglo XX, ya se están dando conexiones importantes entre las ramas mencionadas. Primero, la topología se ha servido de la computación, para probar cuales invariantes y propiedades topológicas se pueden calcular, y cuales no. Y en tiempos más recientes, la computación se ha servido de la topología, para obtener resultados importantes en varias áreas (como por ejemplo, Computación Distribuida). Pero a pesar de todo esto, aún muchos especialistas de las dos ramas, se resisten al cambio; no ven la forma en que pudiéramos llevar las cosas de computación al ámbito continuo de la topología, y viceversa; incluso, muchos no le ven la utilidad. Es muy probable que esta resistencia sea originada, en buena parte, porque desde la formación de cada científico, topólogo y computólogo, se desarrolla una como “alergia”, hacia la otra parte. Es muy común que los matemáticos que gustan de la topología (y también de otras ramas del ámbito continuo, como el análisis, la geometría diferencial, etc.) no gusten de aventurarse demasiado dentro de la lógica (que es donde históricamente, nace la teoría de la computación); y el computólogo, simplemente evita a toda costa todo lo que tenga que ver con áreas como cálculo, análisis y topología, pues siente que no es lo suyo, no es muy hábil en ellas (más aún, muchas veces cuestiona el porque alguien como él, tiene que llevar materias como cálculo o análisis). Así que de entrada, podemos decir que el

topólogo y el computólogo, simplemente, ni se hablan.

Con esta obra, queremos intentar contribuir a la creencia de que esta tendencia no debe seguir más (al menos, no en lo científico), pues, como veremos, se descubren cosas muy interesantes cuando se combinan dos áreas matemáticas tan distintas. Expondremos resultados computacionales que son de gran importancia para quienes se dedican a la topología, en particular, a las variedades (de todos los tipos). Estos resultados fueron presentados por primera vez en 1968, por W. W. Boone, W. Haken y V. Poénaru en [BHP68], con un sabor “muy lógico”, y ciertamente, sufriendo del problema de la notación “extraña” de la lógica. A pesar del esfuerzo de los autores de intentar escribir los resultados, de tal forma que fueran accesibles para los especialistas en cada área, siguen sin ser ampliamente conocidos. En particular, muchos especialistas de variedades no los conocen, o si han escuchado acerca de ellos, no tienen totalmente claro como se han obtenido, ni las fuertes implicaciones que tienen.

Por ello, el relato que haremos en este trabajo, esta dirigido en forma particular, a los especialistas en topología (aunque un computólogo, o especialista en lógica, teniendo un conocimiento básico de los conceptos topológicos usados, también lo puede leer), intentaremos llevarlos cuidadosamente por el camino que es necesario cruzar, para poder llegar a las conclusiones de nuestra referencia principal [BHP68]. Queremos darles un relato agradable y entretenido, mostrándoles como se van combinando bellamente resultados computacionales y topológicos (usando como puente resultados algebraicos de computabilidad), para mostrarnos un resultado por demás trascendente en las matemáticas. Así, la contribución que hacemos con esta obra es precisamente realizar un desarrollo más profundo de todos los pasos necesarios para poder llegar a las conclusiones finales, comenzando con definir el modelo de computo usado (es decir, dar el objeto matemático que será nuestra “computadora”), obtener una forma de representar en forma finita a las variedades diferenciables para que sean manipuladas en programas (sí, esos hechos con lenguajes como C, C++, Java, etc.) y muchas cosas más, hasta finalmente mostrar que la clasificación de variedades está mas alla de nuestras capacidades de cálculo. No cabe duda, un emocionante camino nos está esperando.

¿Y de qué nos habla este resultado? Pues ni mas ni menos, que de la imposibilidad de resolver, en general, los problemas de clasificación topológica,

por difeomorfismo, homeomorfismo, equivalencia homotópica y combinatoria, para variedades diferenciables, topológicas y combinatorias, de dimensión $n \geq 4$. Aún más, veremos, que no existe algún proceso, por medio del cual, pudiéramos decidir si una variedad dada, es simplemente conexa, o no !! El objetivo final, es mostrar que estos problemas topológicos (transformados en problemas de “decisión”) son equivalentes a dos problemas computacionales, que nacen en el álgebra, (para ser precisos, los problema de isomorfismo y de trivialidad, entre grupos finitamente presentados), para los cuales sabemos, no existe algoritmo que pueda resolverlos (Boone, [Bo68]), y por lo tanto, tampoco puede existir algoritmo alguno que resuelva el problema de clasificación de variedades. Para poder definir los problemas de clasificación en forma computacional, será necesario restringir nuestra atención a variedades que se pueden “representar de manera finita”. Este concepto, específicamente en la forma de “presentación finita de una variedad diferenciable”, requiere una discusión detallada que daremos en el Capítulo 4. Definiremos lo que llamaremos una “presentación (finita) algebraica de atlas” (que es esencialmente, una matriz de números racionales), que describe una n -variedad topológica cerrada, con una triangulación (representando la estructura combinatoria) y un atlas (representando la estructura diferenciable). Así, mostraremos que en esta clase de variedades, los problemas mencionados, son imposibles de resolver.

En el Capítulo 1, hacemos una breve introducción al mundo de la Teoría de la computación, iniciando con una reseña histórica de sus inicios y quiénes fueron sus mentores; a continuación introducimos las palabras y los lenguajes, (objetos básicos para realizar cálculos), nuestro modelo formal de computo: la Máquina de Turing (es decir, nuestra computadora), con lo que formalizamos la idea intuitiva de lo que es un algoritmo. Hablaremos de la existencia de la Máquina Universal de Turing, capaz de ejecutar cualquier programa que se le presente, y de como su existencia nos lleva inevitablemente al descubrimiento de problemas matemáticos que no podemos resolver (algunos de estos, ocurriendo en las matemáticas cotidianas). Después, definiremos las Máquinas de Turing con oráculos, las cuales nos permiten ver como se relaciona un problema con otro, con respecto a la dificultad para resolverlos; para finalmente, introducir los grados de Turing, que son una clasificación de todo el universo de posibles problemas; con ellos, seremos capaces de decir exactamente cual es la dificultad de los problemas topológicos que enfrentaremos.

En el Capítulo 2, hablaremos de varios resultados computacionales rela-

cionados con el álgebra, particularmente, con la teoría de grupos. Fue justamente en estos objetos, tan comunes en el quehacer matemático diario, donde se dieron los primeros resultados de imposibilidad computacional, que no surgieron en la lógica o en la computación. Discutiremos acerca del famoso *problema de la palabra*, el cual no tiene solución algorítmica en general [No55], y cómo este hecho implica que tampoco hay un procedimiento finito con el que podamos decir si dos elementos de un grupo son conjugados, o si el grupo tiene orden 1, o es finito, o es libre de torsión, etc.; para rematar con la fuerte insolubilidad de los problemas de isomorfismo y trivialidad, en el sentido de que estos problemas pueden ser “tan difíciles como queramos”. Para poder llevar a cabo la discusión, introduciremos el concepto de presentación (finita) de un grupo y algunas de sus propiedades. La conexión que existe entre la topología y el álgebra (a través de la topología algebraica), es lo que nos permite descubrir que la imposibilidad, también asoma su horrible rostro en las variedades.

En el Capítulo 3 nos movemos hacia la topología. Primeramente, introducimos los conceptos de variedad combinatoria (llamadas también variedades LP), el pegado de asas (desarrollado por Smale) y cirugía de Morse (estos últimos, a un nivel muy básico), pues serán muy necesarios en los siguientes capítulos. Después, presentaremos varios resultados de computabilidad en topología. Veremos como los complejos simpliciales han dado la cara en lo que a cálculos se refiere (gracias a su naturaleza combinatoria), siendo que para estos, existen algoritmos para calcular muchos invariantes topológicos, como los grupos de homología, los números de Betti y presentaciones finitas del grupo fundamental. Pero también hay resultados negativos (aparte de los que expondremos) como el de Markov, sobre la insolubilidad del *problema del homeomorfismo* [Ma58] (este es otro resultado trascendente de imposibilidad en topología, que no es muy conocido). Al final del capítulo, definiremos formalmente los problemas de decisión que se originan de la clasificación de variedades y de la propiedad topológica de ser simplemente conexa.

En el Capítulo 4, nos dedicamos por entero a desarrollar el concepto de *presentación finita de variedades*, que como hemos mencionado, en esencia es una matriz de números racionales. Gracias a una conexión entre la geometría algebraica y la topología diferencial desarrollada por Nash [Na52], es posible representar los sistemas coordenados de toda variedad diferenciable cerrada, por medio de un conjunto de ecuaciones, dadas por polinomios con coeficientes en \mathbb{Q} . De esta forma, nos será posible dotar a las presentaciones finitas de variedades, con elementos para que guarden la información importante

acerca de la estructura diferenciable y/o combinatoria de las variedades que representan.

Finalmente, en el Capítulo 5 combinamos todo lo desarrollado anteriormente, para poder demostrar la fuerte imposibilidad algorítmica (en general) de la propiedad de ser simplemente conexa y de la clasificación de variedades, lo cual nos llevará a concluir, que la clasificación (topológica, diferenciable, combinatoria y homotópica) de variedades, es imposible en general. Sin embargo, también obtenemos otros dos resultados importantes que nos hablan de la existencia de ciertas variedades diferenciables en las cuales sí existen algoritmos para resolver los problemas de clasificación mencionados.

Agradeciendo de antemano a todos las personas que se atrevan emprender el camino en este (esperamos ameno, a pesar de lo largo) relato, el autor espera que les guste y lo disfruten tanto, como el disfruto el hacerlo. Bien...comencemos....

*Si X es un matemático, entonces X es un programador.
Anónimo.*

Capítulo 1

Teoría de la computación

Iniciamos esta obra dando una breve introducción a lo que es la teoría de la computación. Primero damos un resumen histórico de cómo ésta ha evolucionado a través de los años, después introducimos los conceptos básicos, como lo que es una Máquina de Turing, la Máquina Universal, problemas computables, no computables, y finalmente, definiremos los grados de Turing, mencionando algunas de sus muchas propiedades.

1.1. Antecedentes históricos

Muchas personas piensan que las ciencias de la computación se dedican casi de manera exclusiva al estudio de las máquinas modernas, que son precisamente llamadas “computadoras”, junto con los lenguajes de programación, sus aplicaciones, hacer libros que explican cómo manejar procesadores de texto, hojas de cálculo, Internet, etc. Y creen que la ciencia de la computación ha nacido por todo esto. Esta idea es una terrible equivocación, y es, como nos dice Dijkstra (y lo menciona Galaviz [Ga03]), “equivalente a afirmar que el objeto de estudio de la astronomía son los telescopios”. Antes de intentar determinar cuál es el objeto de estudio de la ciencia de la computación, debemos definir primero que es computar, y esto significa, en un sentido general, pasar una colección de datos a través de un proceso que finalmente nos proporciona la solución de un problema. Sin embargo, muchas veces realizar el proceso para solucionar un problema dado puede convertirse en algo muy pesado para una persona.

La historia de la computación ha tenido como motor principal a la pereza. El ser humano siempre ha buscado, desde tiempos antiguos, ideas y mecanismos que le permitan no tener que hacerse cargo de tareas largas, repetitivas, pesadas y aburridas, las cuales no traen consigo alguna retribución espiritual (por ejemplo, hacer cuentas aritméticas) y ha hecho uso de todo su ingenio para inventar artefactos y máquinas que le hagan la vida más cómoda, y le permitan llevar a cabo este tipo de tareas de una forma sencilla y rápida. Muchos de estos mecanismos han tenido como objetivo, hacernos ejecutar los procesos de forma automática, sin pensar en lo que estamos haciendo (tablas de multiplicar). Entre menos haya que pensar para hacer un cálculo, mejor. En otras palabras, el dichoso proceso debe constar de una serie de pasos simples que, al aplicarlos en un orden apropiado, siempre nos proporcionan la solución buscada. Esta búsqueda del ser humano por facilitarse la vida data desde el siglo XII, y los buscadores han sido, entre muchos, ilustres matemáticos. Leonardo de Pisa (Fibonacci) con su libro *Liber Abaci* que versa sobre la notación indo-arábiga y los métodos que se usaban en ella para hacer cálculos; John Neper con los logaritmos y sus tablillas, para hacer multiplicaciones de una forma más sencilla; Blaise Pascal inventó un dispositivo para realizar sumas y restas, y Leibniz, quien creía que todo es aritmética y que los complicados cálculos matemáticos podían ser encargados a “esclavos” confiables que nunca fallaran, inventó una máquina para multiplicar, utilizando ruedas dentadas. Poco a poco, los esfuerzos en encontrar formas sencillas de hacer cálculos se enfocaron en gran medida a facilitar la realización de operaciones aritméticas por medio de máquinas.

Pero también había muchos intentos por automatizar el razonamiento humano, por mecanizar el pensamiento, y hacer sencilla la tarea de deducir la verdad o falsedad de un enunciado a partir de ciertas premisas. Pruebas de esto son las máquinas lógicas de Morland y Jevons, y el famoso trabajo de George Boole: *Investigación sobre las leyes del pensamiento* (1854), donde se sientan las bases de lo que conocemos como álgebra de Boole. Años después, en 1879, Gottlob Frege, un matemático alemán, en un intento por expresar la totalidad de las matemáticas sobre la lógica, publica un folleto llamado *Begriffsschrift* (“Escritura de conceptos” o “Lenguaje de conceptos”), donde asentaba muchos de los fundamentos de la lógica simbólica. A mediados del siglo XIX, Charles Babbage terminó el diseño de su “máquina analítica”, un artefacto programable por medio de tarjetas perforadas, la cual no se hizo realidad durante la vida de Babbage, sino hasta 1906, cuando su hijo, Henry

Babbage, finalizó la construcción de la máquina de su padre y calculó el número π con 29 decimales de precisión. A finales del siglo XIX y principios del XX, otras máquinas surgieron después de la de Babbage, que eran capaces de efectuar la multiplicación de forma directa, y no por medio de sumas repetidas.

Es a principios del siglo XX cuando se alcanzó uno de los logros intelectuales más importantes de la matemática, de la lógica, y de las ciencias de la computación, aún antes de que estas existieran. David Hilbert, en 1900, propone su programa de 23 problemas de lo que él consideraba debía ser el desarrollo de las matemáticas en los próximos años. El segundo de los problemas decía así:

Demostrar la consistencia de los axiomas de la aritmética.

Es decir, probar que no es posible, a partir de dichos axiomas, demostrar al mismo tiempo una proposición y su negación. Había que mostrar que la maquinaria aritmética estaba bien construida, que era infalible. Casi al mismo tiempo, Whitehead y Russell, con su obra titulada *Principia Mathematica* [WR10-13], intentaban reducir las matemáticas a la manipulación simbólica, independiente de alguna semántica. Esto era en parte para entender y evitar las paradojas conjuntistas descubiertas por Russell y otros. Con todo esto, lo que se buscaba era demostrar que el engranaje formal que se había construido a lo largo de la historia era perfecto, una fría máquina, perfecta por su forma y su construcción, no por su significado. Un sistema formal perfecto.

Esta forma de ver el desarrollo de la matemática es altamente computacional, ya que existe una clara analogía entre sistema formal y máquina. En un sistema formal existen ciertos postulados (axiomas) a partir de los cuales todo el resto es construido. El sistema es todo aquello que puede deducirse a partir de los postulados (el conjunto de todos los teoremas) aplicando ciertas reglas de inferencia en una forma puramente mecánica. Una máquina posee ciertos componentes, que efectúan movimientos elementales, regidos por las leyes de la física, que al interactuar, definen todo lo que se puede hacer con la máquina. Hilbert y todos los matemáticos y filósofos de ese tiempo estaban totalmente convencidos de que su intento de mecanizar el quehacer matemático era posible, sin lugar a dudas, tarde o temprano encontrarían la forma de hacerlo. Y de pronto, en 1931, un matemático de nombre

Kurt Gödel, hace ver a todos los matemáticos de la época lo equivocados que estaban, al demostrar sus teoremas de incompletez, los cuales dicen que un sistema formal que contiene a la aritmética es incompleto (habrá verdades que no se pueden demostrar) o bien es inconsistente (se puede demostrar simultaneamente una proposición y su negación), con lo cual, el programa de Hilbert quedó hecho pedazos.

El trabajo de Gödel abrió el espectro de lo que las ciencias de la computación estudian. Sabiendo que hay cosas que no se pueden deducir en un sistema formal, y que la aritmética que utilizamos para hacer cálculos es incompleta, surgen nuevas preguntas, como

- ¿Qué cosas se pueden calcular y cuales no ?
- ¿Son mas las cosas que se pueden calcular o las que no ?
- ¿Qué significa precisamente el que algo sea calculable ?
- ¿Es posible clasificar las cosas que se pueden calcular de acuerdo al grado de dificultad para calcularlas ?

Y muchas otras. De este modo, se presenta una necesidad de formalizar el concepto de algoritmo, de proceso efectivo (del cual se tenía una “idea intuitiva”) para resolver todas estas interrogantes relacionadas con *el problema de la decidibilidad* (Entscheidungsproblem). Así fue como se desarrollaron, casi al mismo tiempo, varias formas de definir lo que es un algoritmo y la computabilidad efectiva. Entre otros, resaltan los siguientes:

- Máquinas de Turing (Alan Turing [Tu36]).
- Sistemas Post (Emile Post [Po36, Po43]).
- Funciones μ -recursivas (Kurt Gödel [Gö65]).
- Cálculo λ (Alonzo Church [Ch33]).

Estos sistemas fueron desarrollados mucho tiempo antes de que las computadoras existieran. Pero lo más impresionante fue que a pesar de que superficialmente se ven tan diferentes, todos estos formalismos puedan simularse uno al otro, y por lo tanto, son equivalentes en poder computacional. Este

hecho no podía ser mera coincidencia, por lo que se pensó que la noción, elusiva por tanto tiempo, de computabilidad efectiva había sido atrapada por fin. Alonzo Church [Ch36] fue el primero en hacer esta declaración, y desde entonces ha sido conocida como la *Tesis de Church* (o la *Tesis de Church-Turing*). No se trata de un teorema, sino mas bien de la creencia de que los sistemas mencionados capturan la idea precisa de lo que es *computabilidad efectiva*, ni mas ni menos. Si usamos el modelo de máquina de Turing, podemos enunciar la *Tesis de Church-Turing* así:

Para todo aquello que es computable, existe una máquina de Turing que lo calcula.

Es a partir de esta hipótesis que se construye toda la Teoría de la Computación, la cual nos servirá para lograr el objetivo principal de esta obra.

Con esto terminamos nuestro resumen de la historia de las ciencias de la computación, el cual ha sido muy apretado. No hemos dado muchos otros datos interesantes, y no mencionamos la evolución tecnológica de la computación. Para el lector interesado en disfrutar un relato muy ameno y con más contenido histórico, le sugerimos consultar [Ga03].

1.2. Definiciones básicas

Probablemente, el modelo de cómputo que ha logrado que la Tesis de Church-Turing sea aceptada ampliamente, ha sido la máquina de Turing. Fue el primer modelo que mostró la característica de ser programable, y es muy parecido en funcionamiento a las computadoras que estamos acostumbrados. Alan Turing usó sus máquinas para demostrar que no era posible resolver algorítmicamente el problema de la decidibilidad [Tu36]. Escogemos, pues, a la máquina de Turing como nuestro modelo formal de cómputo para demostrar la imposibilidad de encontrar una forma de clasificar las variedades diferenciables y/o combinatorias. A continuación, introducimos todos los conceptos esenciales de la teoría de la computación que estaremos usando a lo largo de esta obra. Para profundizar más en el tema, el lector puede consultar [Pa94, Ko97, Co04].

Para programar, no es necesario tener una computadora, ni conocer la sintaxis y la semántica de un lenguaje de programación, como C, Fortran,

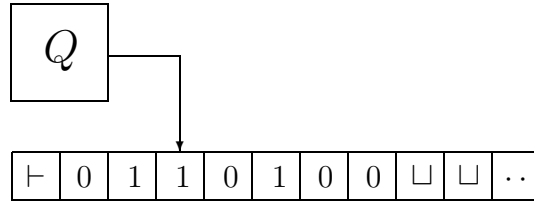


Figura 1.1: Máquina de Turing determinística. El símbolo “ \vdash ” indica a la máquina que no puede seguir avanzando a la izquierda, mientras que los símbolos después del último 0 indica que la celda correspondiente está vacía.

Java, etc. Todo lo que necesitamos es uno de los modelos de computo descritos en la sección anterior (o algún otro de muchos modelos que no mencionamos, pero que existen), unas hojas de papel y lápiz. Demos primero una descripción informal de lo que es una máquina de Turing (Figura 1.1). Intuitivamente, la máquina tiene una cinta semi-infinita dividida en celdas, que es el espacio de trabajo, y una cabeza de lectura y escritura, que puede moverse a la derecha e izquierda, leer y escribir símbolos. En la cinta, la máquina recibe una palabra de símbolos, que representa los datos de entrada para el programa. Con la cabeza lectora, la máquina puede ver los símbolos escritos en la cinta y modificarlos. Al iniciar la ejecución del programa, la máquina se encuentra en un estado “inicial” s , viendo con la cabeza lectora el primer símbolo de la palabra de entrada. En cada paso, dependiendo de qué símbolo se encuentre en la celda que está bajo la mira de la cabeza lectora, y el estado actual de la máquina, esta escribe un nuevo símbolo en la celda, mueve la cabeza lectora a la derecha ó izquierda, y cambia a un nuevo estado. El programa es especificado a través de una función de transición. Cuando la máquina entra en un estado especial t , llamado de “aceptación”, se detiene y decimos que acepta a la palabra de entrada; si entra en otro estado especial r , llamado de “rechazo”, se detiene y decimos que rechaza a la palabra de entrada; y si nunca entra en alguno de estos estados, decimos que la máquina entra en un ciclo infinito. Ahora pasemos a las definiciones de estas ideas.

Definición 1.2.1. Un alfabeto es cualquier conjunto finito (no vacío). Si Σ es un alfabeto, a sus elementos les llamaremos símbolos o letras. A una sucesión finita de símbolos la llamamos una palabra ó palabra sobre el alfabeto Σ . El conjunto de todas las palabras sobre Σ lo denotamos por Σ^* . A la

palabra vacía, que no contiene algún símbolo, la llamamos λ . Un lenguaje¹ es cualquier subconjunto de Σ^* .

Definición 1.2.2. Una Máquina de Turing (MT) determinística de una sola cinta es una séptupla

$$M = (Q, \Sigma, \Gamma, \delta, s, t, r),$$

donde

- Q es un conjunto finito no vacío (los estados);
- Σ es el alfabeto de entrada;
- Γ es el alfabeto de la cinta, donde $\Sigma \subseteq \Gamma$;
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$, la función de transición (el programa de M);
- $s \in Q$, el estado de inicio;
- $t \in Q$, el estado de aceptación;
- $r \in Q$, el estado de rechazo, $r \neq t$.

Intuitivamente, $\delta(p, a) = (q, b, D)$ significa que “Estando en el estado p y teniendo la cabeza lectora en la celda de la cinta con el símbolo a , escribe a b en esa celda, mueve la cabeza lectora en la dirección D , y cambia el estado actual a q . Cuando M entra en alguno de los estados especiales t o r , no permitimos que se hagan transiciones que cambien estos estados. En Γ siempre se encuentra un símbolo especial “ \sqcup ”, llamado el *blanco* y sirve para indicar que la correspondiente celda está vacía. Los símbolos R, L significan derecha e izquierda, respectivamente. Al conjunto de estados y la función de transición los llamamos el *control finito* de M . Más adelante veremos varios ejemplos de MTs, pero primero, daremos otras definiciones importantes.

Definición 1.2.3. Sea $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$ una MT sobre un alfabeto Σ . Definimos una configuración de M , como un elemento del conjunto $Q \times \Gamma^* \times \mathbb{N}$.

¹En teoría de la computación, los términos lenguaje, conjunto y problema se usan indistintamente. Cuando esto no da lugar a ambigüedades, haremos lo mismo a lo largo de la obra.

Una configuración nos da el estado del cálculo de M , en un momento del tiempo. Si (q, w, n) es una configuración, esta nos dice que M se encuentra en el estado q , con la palabra w escrita en la cinta, viendo en su cabeza lectora el símbolo que está en la posición n , dentro de w . En la literatura, es muy usual denotar esto por

$$a_1 \cdots a_{n-1} q a_n \cdots a_m, \quad (w = a_1 \cdots a_m).$$

La *configuración inicial* de M en la entrada w es $(s, w, 1)$. Podemos definir una relación de “siguiente configuración”, en el conjunto $Q \times \Gamma^* \times \mathbb{N}$. Si $(q, w, t), (u, x, p)$ son configuraciones de M , donde $w = a_1 \cdots a_m$, decimos que (q, w, t) nos lleva a (u, x, p) en un paso, denotado por $(q, w, t) \xrightarrow{1}_M (u, x, p)$, si $x = a_1 \cdots a_{t-1} b a_{t+1} \cdots a_m$ y ocurre uno de los siguientes casos:

- M tiene la transición $\delta(q, a_t) = (u, b, R)$ y $p = t + 1$, o
- M tiene la transición $\delta(q, a_t) = (u, b, L)$ y $p = t - 1$.

Usando la notación alterna de configuraciones, esto lo vemos así:

- Si $\delta(q, a_t) = (u, b, R)$ entonces

$$a_1 \cdots a_{t-1} q a_t \cdots a_m \xrightarrow{1}_M a_1 \cdots a_{t-1} b u a_{t+1} \cdots a_m.$$

- Y si $\delta(q, a_t) = (u, b, L)$ entonces

$$a_1 \cdots a_{t-1} q a_t \cdots a_m \xrightarrow{1}_M a_1 \cdots a_{t-2} u a_{t-1} b \cdots a_m.$$

Sean α, β configuraciones. Definimos la cerradura reflexiva y transitiva $\xrightarrow{*}_M$, de la relación $\xrightarrow{1}_M$, en forma inductiva, como sigue:

- $\alpha \xrightarrow{0}_M \alpha$,
- $\alpha \xrightarrow{n+1}_M \beta$ si y sólo si $\alpha \xrightarrow{n}_M \gamma \xrightarrow{1}_M \beta$, para alguna configuración γ , y
- $\alpha \xrightarrow{*}_M \beta$ si y sólo si $\alpha \xrightarrow{n}_M \beta$ para algún $n \geq 0$.

Tenemos todo lo necesario para poder dar definiciones precisas de cuando una MT M acepta o rechaza una palabra, así como del lenguaje aceptado por M .

Definición 1.2.4. Sea M una MT sobre un alfabeto Σ , $x \in \Sigma^*$. Decimos que M acepta a x , si $(s, x, 1) \xrightarrow[M]{*} (t, y, n)$, para algunos $y \in \Sigma^*$, $n \in \mathbb{N}$. M rechaza a x si $(s, x, 1) \xrightarrow[M]{*} (r, z, m)$, donde $z \in \Sigma^*$, $m \in \mathbb{N}$. El lenguaje aceptado por M , $L(M)$ es

$$L(M) = \{x \in \Sigma^* \mid (s, x, 1) \xrightarrow[M]{*} (t, y, n)\}.$$

Dada una MT M y una palabra $x \in \Sigma^*$, decimos que M se *detiene* en x , si M acepta ó rechaza a x . Es posible, por supuesto, que M no acepte ni rechace a x , en cuyo caso decimos que M entra en ciclo con x . Esta observación es muy importante, ya que estamos interesados en formalizar la idea intuitiva de algoritmo, lo cual corresponde a un proceso que siempre se debe de detener en cualquier palabra de entrada, por lo que buscamos máquinas que siempre se detengan en todas las entradas. Esto queda asentado en la siguiente

Definición 1.2.5. Una MT M es total, si para toda palabra $x \in \Sigma^*$, M se detiene en x . Si $A \subseteq \Sigma^*$ es un lenguaje, llamamos a A

- computable enumerablemente (c.e.), si $A = L(M)$ para alguna MT M ,
- co-computable enumerablemente (co-c.e.) si el complemento, $\Sigma^* - A$ es c.e. y
- computable², si $A = L(M)$ para alguna MT total M .

Un conjunto A c.e. es tal que existe una MT M que *sólamete* puede reconocer a los elementos de A . Si $x \in \Sigma^*$ es tal que $x \in A$, entonces al ejecutar a la máquina M en x , sabremos en tiempo finito con certeza que $x \in A$. Pero si por otro lado $x \notin A$ y ejecutamos a M con x como entrada, es posible que M se detenga rechazando a x , con lo que nos daremos cuenta que $x \notin A$; pero también es posible que M nunca se detenga en x . En otras palabras, cuando $x \notin A$, no lo podremos saber con certeza en tiempo finito. El término

²En la mayoría de los libros, se usan los términos recursivo enumerablemente (recursively enumerable) y recursivo (recursive) en lugar de computable enumerablemente y computable, los cuales han sido adoptados en los últimos años, pues reflejan más claramente el concepto que representan.

computable enumerablemente es un poco difícil de manejar al principio, pero afortunadamente tiene una forma equivalente que es más sencilla. Llamamos a un lenguaje $B \subseteq \Sigma^*$ *listable* si existe un programa que imprime a los elementos de B . Si B es infinito, el programa se ejecuta para siempre. Aunque el concepto de listable parece ser muy distinto al de computable enumerablemente, se puede demostrar que son equivalentes. Hablaremos de esto en la Sección 1.3, cuando presentemos un modelo de cómputo equivalente en poder computacional a la MT: La máquina enumeradora. Ahora veamos algunas relaciones entre conjuntos c.e. y computables.

Proposición 1.2.6. *Sea Σ un alfabeto y $A \subseteq \Sigma^*$.*

(i) *Si A es computable, entonces $\Sigma^* - A$ es computable.*

(ii) *Si A es computable, entonces A es c.e.*

(iii) *A y $\Sigma^* - A$ son c.e. si y sólo si A es computable.*

Demostración. (i). Si A es computable, entonces existe M , una MT total tal que $L(M) = A$. Intercambiando los estados de aceptación y rechazo de M , obtenemos una MT M' total, con $L(M') = \Sigma^* - A$.

(ii). En particular, tenemos que $L(M) = A$, y por lo tanto, A es c.e.

(iii). \Rightarrow) Como A y $\Sigma^* - A$ son c.e., entonces existen MTs $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, s_1, t_1, r_1)$ y $M_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, s_2, t_2, r_2)$ tales que $A = L(M_1)$ y $\Sigma^* - A = L(M_2)$. Queremos construir una MT M total con la propiedad de que $L(M) = A$. M no puede ser simplemente una copia de M_1 , ya que esta no necesariamente se detiene en todas las entradas. Tampoco podemos intercambiar los estados de aceptación y rechazo de M_2 e intentar usar a esta nueva máquina, pues tiene el mismo problema que M_1 ; pero dada cualquier palabra $x \in \Sigma^*$, necesariamente al menos una de las dos máquinas M_1 o M_2 se debe detener en x , pues $A \cup (\Sigma^* - A) = \Sigma^*$. Lo que necesitamos es ejecutar al mismo tiempo tanto a M_1 como a M_2 y esperar a que una de ellas se detenga, tomando las decisiones apropiadas en cada caso. Pero nuestras máquinas son *determinísticas*, lo cual implica que solo pueden ejecutar una instrucción a la vez, y no pueden hacerlo en paralelo. La solución a este problema será que la máquina M simule un paso de M_1 , guarde su estado y a continuación simule un paso de M_2 , guarde su estado y entonces verifique en que estados están cada una de las máquinas, y si alguna entra en estado de rechazo o aceptación, tome la decisión apropiada. He aquí el programa de M en pseudocódigo.

M = “En la entrada x :

1. Recupera el estado y el símbolo que M_1 tiene en su control finito.
2. Simula el siguiente paso de M_1 en x y guarda su estado actual.
3. Recupera el estado y el símbolo que M_2 tiene en su control finito.
4. Simula el siguiente paso de M_2 en x y guarda su estado actual.
5. Si M_1 entro en su estado de aceptación, entonces *acepta*.
6. Si M_1 entro en su estado de rechazo, entonces *rechaza*.
7. Si M_2 entro en su estado de aceptación, entonces *rechaza*.
8. Si M_2 entro en su estado de rechazo, entonces *acepta*.
9. Si no se da ninguno de los casos anteriores, regresa al paso 1.”

Veamos ahora como implementar este código. Sea $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, donde

$$Q = \{p_0, p_1, q_0, q_1\} \cup \{o, o', o_1, o_2, u, u_1, s, t, r\} \cup \{p_a\}_{a \in \Sigma} \cup \{o_1^q\}_{q \in Q_1} \cup \{o_2^p\}_{p \in Q_2},$$

$$\Gamma = \Sigma \cup \left\{ \binom{a}{b} \mid a \in \Gamma_1, b \in \Gamma_2 \right\} \cup \left\{ \binom{a_q}{b_p} \mid a \in \Gamma_1, q \in Q_1, b \in \Gamma_2, p \in Q_2 \right\}.$$

La idea detras del alfabeto Γ , es que dividiremos el espacio de trabajo de M en dos secciones, en la parte de arriba, M simulará el programa de M_1 y en la de abajo, el de M_2 . A la función δ la iremos describiendo junto con el comportamiento de M (siguiendo el pseudocódigo). Dada una entrada $x \in \Sigma^*$, la primera tarea de M es recorrer una celda a la derecha la palabra x , lo cual requiere las transiciones

$$\delta(s, a) = (p_0, a, R),$$

$$\delta(p_0, a) = (p_0, a, R),$$

$$\delta(p_0, \sqcup) = (p_1, \sqcup, L),$$

$$\delta(p_1, a) = (p_a, \sqcup, R),$$

$$\delta(p_1, \sqcup) = (p_1, \sqcup, L),$$

$$\delta(p_a, \sqcup) = (p_1, a, L),$$

para todo $a \in \Sigma$. Cuando M se tope con el inicio de la cinta, debe de dividir en dos partes su espacio de trabajo y colocar una copia de x en cada parte.

$$\delta(p_1, \vdash) = (q_0, \vdash, R),$$

$$\delta(q_0, \sqcup) = (q_0, \binom{\vdash}{\vdash}, R),$$

$$\delta(q_0, a) = (q_0, \binom{a}{a}, R), \quad a \in \Sigma.$$

Al terminar la duplicación de x , M regresa su cabeza lectora al inicio del espacio de trabajo dividido (es decir, hasta que se topa con el símbolo $\binom{\vdash}{\vdash}$).

$$\delta(q_0, \sqcup) = (q_1, \binom{\sqcup}{\sqcup}, L),$$

$$\delta(q_1, \binom{a}{a}) = (q_1, \binom{a}{a}, L), \quad a \in \Sigma.$$

$$\delta(q_1, \binom{\vdash}{\vdash}) = (q_0, \binom{\vdash}{\vdash}, R),$$

y marca el primer símbolo de x (en la parte de arriba de la cinta), indicando que M_1 esta viendo a esta letra en su cabeza lectora, y se encuentra en su estado inicial s_1 ; lo mismo para M_2 (en la parte de abajo)

$$\delta(q_0, \binom{a}{a}) = (o, \binom{a_{s_1}}{a_{s_2}}, R), \quad a \in \Sigma.$$

Es aquí, en el estado o , donde M inicia la simulación de “ejecutar al mismo tiempo” a M_1 y M_2 . M recorre la cabeza lectora a la izquierda, buscando el símbolo $\binom{\vdash}{\vdash}$,

$$\delta(o, \binom{a}{b}) = (o, \binom{a}{b}, L), \quad \binom{a}{b} \in \Gamma - \Sigma,$$

$$\delta(o, \binom{\vdash}{\vdash}) = (o_1, \binom{\vdash}{\vdash}, R),$$

al encontrarlo, inicia un recorrido hacia la derecha buscando en la parte de arriba de su cinta dividida, un símbolo de la forma b_q , donde $b \in \Gamma_1$ y $q \in Q_1$,

$$\delta(o_1, \binom{c}{d}) = (o_1, \binom{c}{d}, R), \quad c \in \Gamma_1, d \in C_2,$$

donde $C_2 = \Gamma_2 \cup \{e_p \mid e \in \Gamma_2, p \in Q_2\}$. De esta forma, M sabe en que estado y símbolo se quedo la ejecución de M_1 y procede a efectuar la siguiente transición, segun el programa δ_1 . Así, δ contiene transiciones

$$\delta(o_1, \binom{c_q}{d}) = (o_1^g, \binom{a}{d}, D), \quad d \in C_2,$$

por cada transición de M_1 de la forma $\delta_1(q, c) = (g, a, D)$ ($D \in \{R, L\}$). Finalmente, M debe guardar el contenido (estado actual y símbolo bajo la cabeza lectora) del control finito de M_1 . Esto requiere las transiciones

$$\delta(o_1^g, \binom{c}{d}) = (o', \binom{c_g}{d}, R), \quad g \in Q_1, c \in \Gamma_1, d \in C_2.$$

El estado o' le indica a M que ahora debe simular a M_2 , para ello, agregamos transiciones similares a las que tenemos para la ejecución de M_1 (sea $C_1 = \Gamma_1 \cup \{f_q \mid f \in \Gamma_1, q \in Q_1\}$).

$$\delta(o', \binom{a}{b}) = (o', \binom{a}{b}, L), \quad \binom{a}{b} \in \Gamma - \Sigma,$$

$$\delta(o', \binom{\vdash}{\vdash}) = (o_2, \binom{\vdash}{\vdash}, R),$$

$$\delta(o_2, \binom{c}{d}) = (o_2, \binom{c}{d}, R), \quad c \in C_1, d \in \Gamma_2,$$

$$\delta(o_2, \binom{c}{d_p}) = (o_2^h, \binom{c}{b}, F), \text{ tal que } \delta_2(p, d) = (h, b, F), F \in \{R, L\} \text{ y } c \in C_1,$$

$$\delta(o_2^h, \binom{c}{d}) = (u, \binom{c}{d_h}, R), \quad h \in Q_2, c \in C_1, d \in \Gamma_2.$$

Cuando M pasa al estado u , es el momento de ver en que estados se encuentran M_1 y M_2 . Para ello, M recorre su cabeza lectora a la izquierda hasta encontrar el símbolo $\binom{\vdash}{\vdash}$, luego inicia la lectura de las palabras en las cintas de M_1 y M_2 , buscando sus estados de aceptación o rechazo, y M debe aceptar o rechazar de acuerdo a lo que especificamos en el pseudocódigo que dimos. Este comportamiento lo logramos con las siguientes transiciones

$$\delta(u, \binom{a}{b}) = (u, \binom{a}{b}, L),$$

$$\delta(u, \binom{\vdash}{\vdash}) = (u_1, \binom{\vdash}{\vdash}, R),$$

$$\delta(u_1, \binom{a}{b}) = (u_1, \binom{a}{b}, R),$$

$$\delta(u_1, \binom{c_{t_1}}{d}) = (t, \binom{c_{t_1}}{d}, R),$$

$$\delta(u_1, \binom{c_{r_1}}{d}) = (r, \binom{c_{r_1}}{d}, R),$$

$$\delta(u_1, \binom{c}{d_{t_2}}) = (r, \binom{c}{d_{t_2}}, R),$$

$$\delta(u_1, \binom{c}{d_{r_2}}) = (t, \binom{c}{d_{r_2}}, R),$$

para cada $a, b \in \Gamma - \Sigma, c \in \Gamma_1, d \in \Gamma_2$. Si M no encuentra a M_1 y M_2 en sus estados de aceptación o rechazo, entonces moverá su cabeza lectora a hacia la derecha hasta encontrarse con el símbolo blanco \sqcup de su cinta y debe regresar a simular el siguiente paso de la ejecución de cada máquina, lo cual logramos así:

$$\delta(u_1, \sqcup) = (o, \sqcup, L).$$

Ya casi completamos la especificación completa de M , sólo nos falta tratar un caso especial: Puede ocurrir que las palabras en las cintas de trabajo de M_1 y M_2 crezcan con una longitud más grande que la que tiene x . Entonces en medio de una simulación de los pasos de M_1 ó de M_2 se puede presentar la necesidad de hacer mas grande el espacio de trabajo de alguna de estas máquinas, lo que se traduce en la cinta de M en que nos encontraremos con un blanco (el de M , que es \sqcup). Esto lo resolvemos aumentando el espacio de trabajo dividido para las dos máquinas, lo cual resolvemos colocando las transiciones

$$\delta(o_1^q, \sqcup) = (o', \left(\begin{smallmatrix} \sqcup_q \\ \sqcup \end{smallmatrix}\right), L), \quad q \in Q_1,$$

$$\delta(o_2^p, \sqcup) = (u, \left(\begin{smallmatrix} \sqcup \\ \sqcup_p \end{smallmatrix}\right), L), \quad p \in Q_2.$$

Y esta es la especificación completa de M . Para concluir, sólo debemos argumentar que en efecto, M es una máquina total y además $L(M) = A$. Que M es total se sigue del hecho de que A y $\Sigma^* - A$ son una partición de Σ^* y por ello, cualquier palabra x estará en $L(M_1)$ ó en $L(M_2)$. Esto significa que, al correr M a M_1 y M_2 en x , alguna de ellas eventualmente se detendra, aceptando o rechazando a x , por lo que M entrará en su estado de aceptación o de rechazo, es decir, M se detendra en x . Ya que x es cualquier palabra en Σ^* , M es una MT total.

Probemos ahora que $L(M) = A$. Si $x \in L(M)$, esto significa que M acepto a x , y esto solo puede pasar si en la ejecución de M_1 y M_2 que realiza M sucede uno de dos posibles casos:

1. M_1 entro en su estado de aceptación,
2. M_2 entro en su estado de rechazo.

Si ocurre el caso 1, entonces tenemos que $x \in L(M_1) = A$. Si sucede el caso 2, entonces $x \notin L(M_2) = \Sigma^* - A$, lo cual implica que $x \in A$. Supongamos ahora que $x \in A$. Si alimentamos a M con x y la ejecutamos, entonces como $x \in A$, en la simulación que M hace de M_1 y M_2 , llegará un momento en el que M_1 entrará en su estado de aceptación, lo cual hará que M acepte a x ; también podría suceder que antes que M_1 acepte, M_2 rechace a x , pero de acuerdo con la especificación de M , esto también hará que M acepte a x .

En cualquier caso, M se detiene aceptando, por lo cual $x \in L(M)$. Hemos demostrado que

$$x \in L(M) \Leftrightarrow x \in A,$$

con lo que tenemos que $L(M) = A$.

\Leftarrow) Por *i*), $\Sigma^* - A$ es computable y por *ii*), A y $\Sigma^* - A$ son c.e. □

Aunque los conjuntos computables son cerrados bajo complemento, los conjuntos c.e. no lo son. También, la converso de la Proposición 1.2.6 inciso *ii*) no siempre es cierta, es decir, no toda MT es equivalente a una MT total (algoritmo), por lo que hay lenguajes de los que solamente podemos reconocer a sus elementos; si la máquina es alimentada con una palabra que no esta en el lenguaje, no sabemos que pueda pasar, podría detenerse y rechazar, ó podría entrar en un ciclo infinito. Volveremos a estos temas mas adelante, en la Sección 1.4, cuando hablemos de la Máquina Universal de Turing.

Decidibilidad y semidecidibilidad

Una propiedad P de palabras sobre un alfabeto Σ es *decidible*, si el conjunto que consiste de todas las palabras que tienen la propiedad P es computable. Si este conjunto es solamente c.e., entonces P es *semidecidible*. Muchas veces, los especialistas en lógica y teoría de la computación usan los términos decidible, semidecidible, computable y c.e. de manera indistinta tanto para conjuntos como propiedades. Este abuso está justificado, ya que estos términos son equivalentes,

$$\begin{aligned} P \text{ es decidible} &\Leftrightarrow \{x \in \Sigma^* \mid P(x)\} \text{ es computable,} \\ A \text{ es computable} &\Leftrightarrow "x \in A" \text{ es decidible.} \end{aligned}$$

$$\begin{aligned} P \text{ es semidecidible} &\Leftrightarrow \{x \in \Sigma^* \mid P(x)\} \text{ es c.e.,} \\ A \text{ es c.e.} &\Leftrightarrow "x \in A" \text{ es semidecidible.} \end{aligned}$$

Y no solo estaremos interesados en poder decidir la pertenencia de palabras en lenguajes, también queremos calcular funciones. Necesitamos pues, máquinas de Turing que puedan dar una salida que no sea un simple "sí" y "no".

Definición 1.2.7. Sea Σ un alfabeto y $f: D \subseteq \Sigma^* \rightarrow \Sigma^*$ una función de palabras. Decimos que f es parcialmente computable (p.c.), si existe una MT

$M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, que calcula la función f , es decir, si en la entrada $x \in D$, M termina su cómputo (al entrar en su estado t) con la palabra $f(x)$ escrita en la cinta. Esto lo denotamos por $M(x) = f(x)$. Si además, tenemos que $D = \Sigma^*$, entonces decimos que f es computable.

En el caso de las funciones, el término *parcialmente computable* se refiere al hecho de que existe un algoritmo que nos dice claramente cómo calcular la función, siempre que se le de como entrada un elemento del dominio de ésta. Pero este procedimiento no está obligado a dar una salida cuando le es alimentada una palabra que no está en el dominio de la función (de aquí el prefijo *parcialmente*), por lo que en este caso, la máquina podría detenerse en su estado de rechazo o podría quedarse en un ciclo infinito. Por ello, cuando la función está definida en todo Σ^* , decimos que es *computable*.

Con el marco de trabajo que definen todos estos conceptos, podemos empezar a codificar muchos objetos de uso diario en las matemáticas, y preguntarnos acerca de la computabilidad de muchas funciones conocidas. Veamos algunos ejemplos.

Ejemplo 1.2.8. Mostremos codificaciones para algunos conjuntos conocidos. Los números naturales \mathbb{N} , pueden ser codificados usando el alfabeto $\Sigma = \{1\}$, donde la palabra 1^n representa al número n . Para los enteros \mathbb{Z} , usemos el alfabeto

$$\Sigma = \{0, 1, +, -\},$$

y si tomamos $m \in \mathbb{Z}$ tal que $m > 0$, lo representamos con la palabra $+1^m$; si $m < 0$, con la palabra -1^m ; el cero es representado por el símbolo “0”. También podemos usar el alfabeto $\Sigma = \{0, 1\}$ y utilizar codificación binaria, como en las computadoras. Si ahora deseamos encontrar una representación en palabras de los racionales \mathbb{Q} , fijamos como alfabeto a $\Sigma = \{0, 1, +, -, /\}$ y, reutilizando la codificación de enteros, la palabra $+111/-1111111$ representaría al racional $\frac{3}{7}$. El 0 puede ser representado por 0/1.

Ejemplo 1.2.9. Encontremos una codificación en palabras para el conjunto $\mathbb{Q}[x]$, de polinomios con coeficientes en \mathbb{Q} . Tomemos a $\Sigma = \{0, 1, +, -, a, /\}$. Para los coeficientes, que son racionales, reutilizamos la codificación dada en el Ejemplo 1.2.8. Para codificar el grado de cada término, usamos palabras en el símbolo “ a ”. Así, si tenemos el polinomio $2x^3 + \frac{1}{5}x^2 - x + 4$, este tiene la representación

$$+11/1aaa + 1/11111aa - 1/1a + 1111/1.$$

Para no confundir al polinomio cero con el racional cero, lo podemos representar con la palabra $0/1a$. De esta forma, podemos ver a cualquier polinomio como una palabra en el alfabeto dado.

Ejemplo 1.2.10. Construyamos una MT que sume números naturales. Sea $\Sigma = \{0, 1\}$, y reutilicemos la codificación de \mathbb{N} que dimos en el Ejemplo 1.2.8. Intuitivamente, lo que deseamos es que la máquina reciba como entrada las palabras 1^n y 1^m (representando los naturales n, m) y que al terminar su cómputo, la cinta contenga la palabra 1^{n+m} , es decir, queremos “pegar” las palabras iniciales. Podemos usar el símbolo 0 para separar a las palabras que representan los número a sumar, y la tarea de la máquina sera borrar este cero y juntar ambas palabras de 1’s en una sola. Así, queremos calcular la función de palabras

$$f: \{1^n 0 1^m \mid n, m \in \mathbb{N}\} \subseteq \Sigma^* \rightarrow \Sigma^*,$$

tal que $f(1^n 0 1^m) = 1^{n+m}$. En palabras que no están en el dominio de f , el comportamiento de la máquina estará indefinido. Sea $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$ una MT tal que

$$Q = \{s, q_1, q_2, q_3, t, r\},$$

$$\Sigma = \{0, 1\},$$

$$\Gamma = \{0, 1, \sqcup\}.$$

La función δ viene especificada en la siguiente tabla.

δ	0	1	\sqcup
s	(r, \cdot, \cdot)	$(q_1, 1, R)$	(r, \cdot, \cdot)
q_1	$(q_2, 1, R)$	$(q_1, 1, R)$	(r, \cdot, \cdot)
q_2	(r, \cdot, \cdot)	$(q_2, 1, R)$	(q_3, \sqcup, L)
q_3	\cdot	(t, \sqcup, R)	\cdot
t	\cdot	\cdot	\cdot
r	\cdot	\cdot	\cdot

Las partes con el símbolo “ \cdot ” en la tabla significan que podemos poner cualquier transición o símbolo. En la entrada 11101111 tenemos la siguiente sucesión de configuraciones:

$$\begin{aligned} s11101111 &\xrightarrow[M]{1} 1q_11101111 \xrightarrow[M]{*} 111q_101111 \xrightarrow[M]{1} 1111q_21111 \\ &\xrightarrow[M]{*} 11111111q_2\sqcup \xrightarrow[M]{1} 11111111q_31 \xrightarrow[M]{1} 11111111\sqcup t\sqcup. \end{aligned}$$

Y así, $M(11101111) = 1111111$.

Ejemplo 1.2.11. Construyamos una MT total que decide si la palabra de entrada es un polinomio con coeficientes racionales de grado 3. La máquina primero verifica que la palabra es, en efecto, un polinomio válido (reutilizamos la codificación del Ejemplo 1.2.9), después de esto, analiza al polinomio buscando que alguno de los términos sea de grado 3. Si lo encuentra y en efecto este es el mayor grado, acepta la entrada, si no lo encuentra o hay al menos un término de grado mayor a 3, rechaza la palabra. Formalmente, $M_3 = (Q, \Sigma, \Gamma, \delta, s, t, r)$ donde,

$$Q = \{s, q_1, q_z, q_u, q_d, q_{uu}, q_a, q_b, p, p_1, p_2, p_3, g, g_1, g_2, g_3, r, t\},$$

$$\Sigma = \{+, -, 0, 1, a, /\},$$

$$\Gamma = \{+, -, 0, 1, a, /, \sqcup\},$$

y δ está dada en la siguiente tabla.

δ	+	-	0	1	a	\sqcup	/	\vdash
s	$(q_1, +, R)$	$(q_1, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	.
q1	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_z, 0, R)$	$(q_u, 1, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	.
qz	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_d, /, R)$.
qu	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_u, 1, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_d, /, R)$.
qd	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_{uu}, 1, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	.
quu	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(q_{uu}, 1, R)$	(q_a, a, R)	(q_b, \sqcup, L)	(r, \cdot, \cdot)	.
qa	$(q_1, +, R)$	$(q_1, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(q_a, a, R)	(q_b, \sqcup, L)	(r, \cdot, \cdot)	.
qb	$(q_b, +, L)$	$(q_b, -, L)$	$(q_b, 0, L)$	$(q_b, 1, L)$	(q_b, a, L)	(r, \cdot, \cdot)	$(q_b, /, L)$	(p, \vdash, R)
p	$(p, +, R)$	$(p, -, R)$	$(p, 0, R)$	$(p, 1, R)$	(p_1, a, R)	(r, \cdot, \cdot)	$(p, /, R)$.
p1	$(p, +, R)$	$(p, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(p_2, a, R)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	.
p2	$(p, +, R)$	$(p, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(p_3, a, R)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	.
p3	$(g, +, R)$	$(g, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(t, \cdot, \cdot)	(r, \cdot, \cdot)	.
g	(r, \cdot, \cdot)	(r, \cdot, \cdot)	$(g, 0, R)$	$(g, 1, R)$	(g_1, a, R)	(t, \cdot, \cdot)	$(g, /, R)$.
g1	$(g, +, R)$	$(g, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(g_2, a, R)	(t, \cdot, \cdot)	(r, \cdot, \cdot)	.
g2	$(g, +, R)$	$(g, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(g_3, a, R)	(t, \cdot, \cdot)	(r, \cdot, \cdot)	.
g3	$(g, +, R)$	$(g, -, R)$	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(r, \cdot, \cdot)	(t, \cdot, \cdot)	(r, \cdot, \cdot)	.
t
r

Efectuemos algunos cálculos con M_3 . En la entrada $-11/111$ tenemos,

$$\begin{aligned}
s - 11/111 &\xrightarrow{M_3} -q_1 11/111 \xrightarrow{M_3} -1q_u 1/111 \xrightarrow{M_3} -11q_u/111 \\
&\xrightarrow{M_3} -11/q_d 111 \xrightarrow{M_3} -11/1q_{uu} 11 \xrightarrow{M_3}^* -11/111q_{uu} \sqcup \\
&\xrightarrow{M_3} -11/11q_b 1 \xrightarrow{M_3}^* q_b \vdash -11/111 \xrightarrow{M_3} p - 11/111 \\
&\xrightarrow{M_3}^* -11/111p \sqcup \xrightarrow{M_3} -11/11r 1.
\end{aligned}$$

Y por lo tanto, M_3 rechaza a $-11/111$. En la entrada $+1/1aa - 1/1$ M_3 ejecuta lo siguiente,

$$\begin{aligned}
& s + 1/1aa - 1/1 \xrightarrow{M_3} +q_1 1/1aa - 1/1 \xrightarrow{M_3} +1q_u/1aa - 1/1 \xrightarrow{M_3} +1/q_d 1aa - 1/1 \\
& \xrightarrow{M_3} +1/1q_{uu}aa - 1/1 \xrightarrow{M_3} +1/1aq_aa - 1/1 \xrightarrow{M_3} +1/1aaq_a - 1/1 \\
& \xrightarrow{M_3} +1/1aa - q_1 1/1 \xrightarrow{M_3} +1/1aa - 1q_u/1 \xrightarrow{M_3} +1/1aa - 1/q_d 1 \\
& \xrightarrow{M_3} +1/1aa - 1/1q_{uu} \sqcup \xrightarrow{M_3} +1/1aa - 1/q_b 1 \xrightarrow{M_3} q_b \vdash +1/1aa - 1/1 \\
& \xrightarrow{M_3} p + 1/1aa - 1/1 \xrightarrow{M_3} +1/1paa - 1/1 \xrightarrow{M_3} +1/1ap_1a - 1/1 \\
& \xrightarrow{M_3} +1/1aap_2 - 1/1 \xrightarrow{M_3} +1/1aa - p1/1 \xrightarrow{M_3} +1/1aa - 1/1p \sqcup \\
& \xrightarrow{M_3} +1/1aa - 1/r1.
\end{aligned}$$

Por lo que el polinomio $x^2 - 1$ es rechazado por M_3 . Finalmente, veamos que pasa con la palabra $+11/1aaa + 1/11111aa - 1/1a$,

$$\begin{aligned}
& s + 11/1aaa + 1/11111aa - 1/1a \xrightarrow{M_3} +11/1aaa + 1/11111aa - 1/1aa \sqcup \\
& \xrightarrow{M_3} +11/1aaa + 1/11111aa - 1/1q_b a \xrightarrow{M_3} q_b \vdash +11/1aaa + 1/11111aa - 1/1a \\
& \xrightarrow{M_3} p + 11/1aaa + 1/11111aa - 1/1a \xrightarrow{M_3} +11/1paaa + 1/11111aa - 1/1a \\
& \xrightarrow{M_3} +11/1aaap_3 + 1/11111aa - 1/1a \xrightarrow{M_3} +11/1aaa + g1/11111aa - 1/1a \\
& \xrightarrow{M_3} +11/1aaa + 1/11111gaa - 1/1a \xrightarrow{M_3} +11/1aaa + 1/11111aag_2 - 1/1a \\
& \xrightarrow{M_3} +11/1aaa + 1/11111aa - g1/1a \xrightarrow{M_3} +11/1aaa + 1/11111aa - 1/1ga \\
& \xrightarrow{M_3} +11/1aaa + 1/11111aa - 1/ag_1 \sqcup \xrightarrow{M_3} +11/1aaa + 1/11111aa - 1/1ta.
\end{aligned}$$

Así, M_3 acepta al polinomio $2x^3 + \frac{1}{5}x^2 - x$.

De forma similar a estos ejemplos, podríamos seguir encontrando codificaciones para todo tipo de objeto que podamos representar en un lenguaje finito (gráficas combinatorias, funciones, etc.) y preguntarnos acerca de la computabilidad de sus propiedades, operaciones y demás. En particular, todas las

operaciones básicas y propiedades de polinomios con coeficientes racionales, son fácilmente llevadas a cabo por MTs. Esto es de gran importancia, ya que los racionales y su densidad topológica son uno de los ingredientes que permiten que podamos realizar cuestiones de cómputo en el ámbito de lo continuo.

Cuando hacemos, o vemos ejemplos concretos de implementaciones de algoritmos en MTs, pronto nos damos cuenta de que dar la especificación completa de una MT puede ser una tarea bastante larga y engorrosa, muchas veces, poco o nada ilustrativa³. Por ello, para no hacer tediosas nuestras demostraciones y argumentos en donde intervienen algoritmos, sólo daremos una descripción corta, en pseudocódigo (como hicimos en la demostración del inciso *iii*) de la Proposición 1.2.6); y a veces, nos conformaremos con la idea principal del algoritmo. Esto es algo usual en la teoría de la computación, y la justificación para hacerlo, es que estamos respaldados por la tesis de Church-Turing. También, de aquí en adelante, fijaremos nuestro alfabeto de trabajo a $\Sigma = \{0, 1\}$, esto no nos hará perder generalidad en nuestros resultados, ya que se puede ver que siempre es posible codificar palabras de un alfabeto cualquiera a palabras en el alfabeto $\{0, 1\}$ y por lo tanto, podemos manejar los problemas que atacaremos con este conjunto. Esto es algo que no debe sorprendernos, ya que Σ es precisamente el alfabeto de uso de las computadoras de hoy día. Todos los programas son transformados por los compiladores a palabras de 0's y 1's, y de esta forma son manipulados por nuestras computadoras.

1.3. Modelos equivalentes

Ahora veremos que el concepto de computabilidad que encierra la MT es muy robusto. Para ello, presentaremos diferentes variaciones en el modelo de la MT, que parecerían aumentar o disminuir su poder de cómputo, y sin embargo resulta que no es así, todos son equivalentes al modelo básico introducido en la Definición 1.2.2. Cabe mencionar que existen muchos de estos modelos equivalentes, pero solo mostraremos algunos. Nos interesará, en forma particular, el concepto de *máquinas enumeradoras*, el cual usaremos en

³En términos prácticos de programación, puede ser más complicado que hacer un programa completo en lenguaje ensamblador (de máquina).

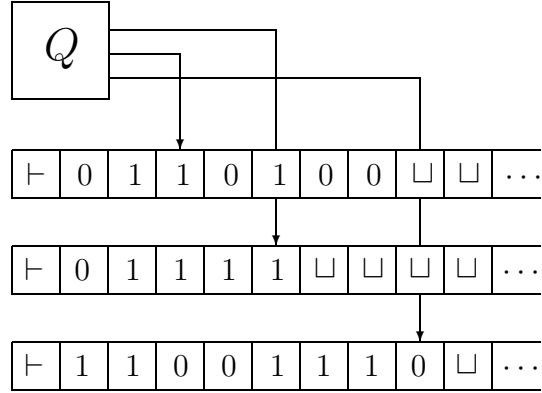


Figura 1.2: Máquina de Turing de cintas múltiples.

varios resultados necesarios para mostrar la imposibilidad de la clasificación de las variedades.

Máquinas de cintas múltiples

Una máquina de Turing con k -cintas es muy similar a una MT estándar, solo que la primera tiene k cintas semi-infinitas a la derecha y k cabezas de lectura y escritura independientes una de la otra. Al iniciar un cómputo, la palabra de entrada esta en la primera cinta y las otras están en blanco. En cada paso, la máquina lee un símbolo de cada cinta y basándose en esta información y su estado actual, escribe un símbolo nuevo en cada cinta, mueve sus cabezas (no se mueven necesariamente en la misma dirección) y cambia el estado. La función de transición es de la forma

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k,$$

y una transición de la forma $\delta(q, a_{i1}, a_{i2}, \dots, a_{ik}) = (p, b_{i1}, b_{i2}, \dots, b_{ik}, D_1, D_2, \dots, D_k)$ significa “Estando en el estado q , viendo en la primera cinta el símbolo a_{i1} , en la segunda el símbolo a_{i2}, \dots , en la k -ésima cinta el símbolo a_{ik} , cambia el estado actual a p , escribe en la primera cinta el símbolo b_{i1} , en la segunda el símbolo b_{i2}, \dots , en la k -ésima cinta el símbolo b_{ik} y mueve la cabeza lectora en la primera cinta a la dirección D_1 , en la segunda a la dirección D_2, \dots , en la k -ésima a la dirección D_k ($D_i = L$ ó R)”. Cuando la máquina inicia su funcionamiento, la primera transición que se ejecuta es con la $(k+1)$ tupla $(s, a, \sqcup, \dots, \sqcup)$, donde a es e primer símbolo de la palabra de

entrada, y todas las demás cintas están vacías.

Una ventaja de trabajar con este modelo es que da mayor claridad a los programas, ya que la información se puede separar en varias cintas, por este motivo es muy usado en los libros de algoritmos y complejidad. En la Figura 1.2, podemos ver una MT con 3-cintas.

Máquinas de cinta infinita en ambas direcciones

Una máquina de Turing con cinta infinita a ambos lados (Figura 1.3) se define y se comporta exactamente de la misma manera que nuestro modelo estándar, la única diferencia es que se puede mover hacia la izquierda tanto como sea necesario. Salvo esto, no tiene alguna ventaja en particular.

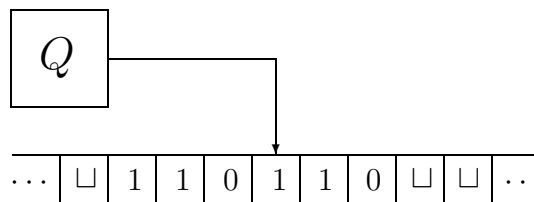


Figura 1.3: Máquina de Turing de cinta infinita en ambos lados.

Máquinas no deterministas

Las máquinas de Turing no deterministas encierran el poder de “adivinar”. En un momento dado, dado un cierto estado de la máquina, esta puede tomar al mismo tiempo más de una decisión y en cada una, seguir cálculos distintos. Este modelo de cómputo es irreal, en el sentido de que no es posible llevarlo a cabo en forma práctica. Sin embargo, también es equivalente al modelo de MT introducido en la Definición 1.2.2. Una Máquina de Turing no determinística es una séptupla $N = (Q, \Sigma, \Gamma, \Delta, s, t, r)$, donde Q, Σ, Γ, s, t y r son como en una MT estándar. Para reflejar el hecho de que N ya no tiene necesariamente una única siguiente elección en cada paso, sino varias, Δ no es una función, sino una relación $\Delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}$. Es decir, para cada combinación *estado-símbolo*, puede haber más de un siguiente paso, o ninguno. En la Figura 1.4 podemos visualizar una posible ejecución de N en una palabra dada. Intuitivamente, N acepta una palabra de entrada, si en el “árbol de ejecución”, existe un camino desde la raíz hasta un nodo hoja, en donde N se encuentre en su estado de aceptación.

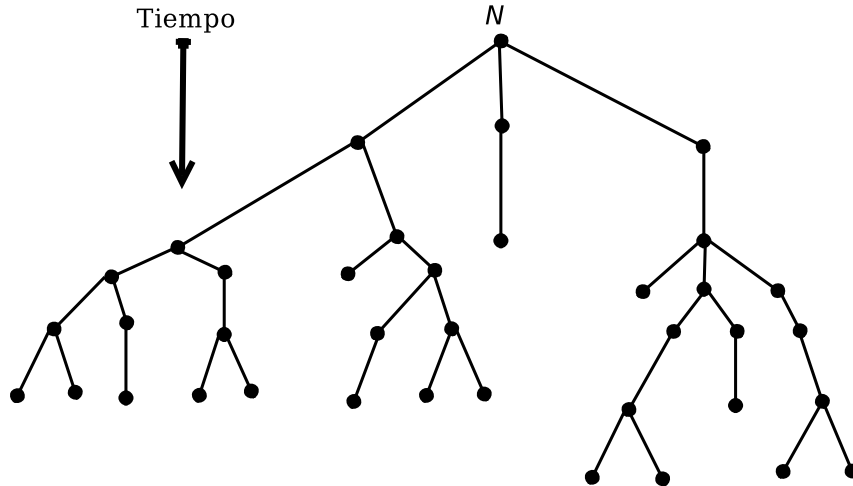


Figura 1.4: Ejecución de una Máquina de Turing no determinística.

Actualmente, la mejor forma que se conoce para simular una máquina de Turing no determinística, es usando una MT estándar con un número exponencial de pasos, en el peor de los casos. Esto, por supuesto, no es una solución práctica. Buscar una manera más eficiente de llevar a cabo esta simulación es uno de los problemas abiertos⁴ más importantes de la Teoría de la computación y de la matemática misma [Si05, Pa94].

Máquinas enumeradoras

En la Sección 1.2 definimos los conjuntos computables enumerablemente (c.e.), como aquellos conjuntos que son aceptados por máquinas de Turing.

⁴El famoso problema $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. El enunciado formal de esto requiere la introducción de muchos conceptos de *complejidad computacional*, pero afortunadamente hay muchas formas equivalentes para describirlo. Consideremos el siguiente problema numérico: Sean C un subconjunto finito de naturales, y $W \in \mathbb{N}$. Decidir si existe un subconjunto $D \subseteq C$ tal que

$$\sum_{d \in D} d = W.$$

Este problema de decisión es conocido como *SUBCONJUNTO SUMA*. Entonces, el problema $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ es equivalente a demostrar si existe una MT total que decida a *SUBCONJUNTO SUMA* y que se ejecute en tiempo proporcional a un polinomio $p(n)$, donde $n = |C|$ (el tamaño de la entrada).

El termino *computable enumerablemente* nació de un formalismo diferente (pero equivalente) que encierra la idea de que los elementos de un conjunto c.e. pueden ser *enumerados* en una lista (posiblemente infinita) de una forma mecánica [Ko97, Co04]. Esto es precisamente lo que hacen las máquinas enumeradoras, las cuales fueron inventadas por Alan Turing [Tu36], de hecho, esta es la forma en la que nacieron las MTs, ya que Turing estaba interesado en enumerar las expansiones decimales de números y funciones reales computables.

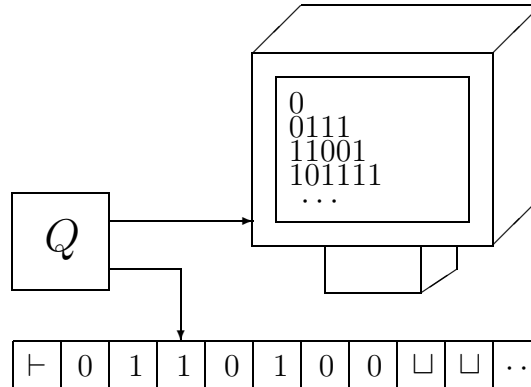


Figura 1.5: Máquina Enumeradora. Representamos la cinta de salida, donde se enumeran las palabras, por un monitor de computadora (es muy usual representarla por una impresora).

La idea intuitiva detras de una máquina enumeradora esta dada en la Figura 1.5. La máquina tiene un control finito (cabeza de lectura, escritura, conjunto de estados y función de transición), y dos cintas: una de lectura y escritura, llamada la cinta de trabajo; y otra solo de escritura, llamada la cinta de salida. La cabeza en la cinta de trabajo se puede mover en cualquier dirección, escribiendo y leyendo elementos del alfabeto de las cintas, Γ . La cabeza en la cinta de salida se mueve solamente a la derecha, cada vez que escribe un símbolo del alfabeto de salida $\Sigma \subseteq \Gamma$. La máquina no tiene estados de rechazo ni de aceptación, y tampoco recibe palabras de entrada. Al iniciar el cómputo, esta empieza en un estado inicial, con ambas cintas en blanco, moviendo sus cabezas de acuerdo al programa (función de transición), ocasionalmente escribiendo símbolos en la cinta de salida. La máquina tiene un estado distinguido, llamado el *estado de enumeración*; y cuando entra en este estado, decimos que la palabra que esta escrita en la cinta de salida ha sido

enumerada. Despues de esto, la cinta de salida es automáticamente borrada y la cabeza lectora movida al inicio de la cinta, y la maquina continua sus cálculos, ejecutando el programa para siempre (la misma palabra podría ser enumerada mas de una vez). Si E es una máquina enumeradora, definimos el lenguaje enumerado por E , $L(E)$, como el conjunto de palabras en Σ^* que son enumeradas por E .

Definición 1.3.1. Una máquina enumeradora es una séxtupla $E = (Q, \Sigma, \Gamma, \delta, s, e)$, donde

- Q, Σ, Γ y s son como en la Definición 1.2.2;
- $\delta: Q \times \Gamma^2 \rightarrow Q \times \Gamma^2 \times \{R, L\}^2$ es la función de transición (E es una máquina de dos cintas);
- $e \in Q$ es un estado especial, llamado el estado de enumeración.

Cada vez que E entra en su estado de enumeración, decimos que la palabra que esta en la segunda cinta es enumerada, y a continuación se debe de borrar el contenido de la cinta, es decir, E siempre contiene las transiciones

$$\delta(e, (a_i, \sqcup)) = (q_b, (a_i, \sqcup), (D, L)),$$

$$\delta(q_b, (a_i, a_j)) = (q_b, (a_i, \sqcup), (F, L)),$$

para cada $a_i, a_j \in \Sigma$, y algunos⁵ $D, F \in \{R, L\}$, $q_b \in Q$.

Esta es la definición formal que esta detras del concepto de *conjunto listable* que dimos despues de definir los conjuntos c.e. Veamos algunos ejemplos.

Ejemplo. El conjunto $A = \{a \in \mathbb{Z} \mid \exists x, y, z \in \mathbb{Z}(a = x^3 + y^3 + z^3)\}$ es listable. Nuestro programa (usando la codificación para \mathbb{Z} que dimos en el Ejemplo 1.2.8) para listar a A primero imprime todos los posibles enteros de la forma $x^3 + y^3 + z^3$ para todos $|x|, |y|, |z| \leq 10$; despues imprime $x^3 + y^3 + z^3$ para $|x|, |y|, |z| \leq 100$ y así sucesivamente.

Ejemplo. El conjunto $P = \{p \in \mathbb{N} \mid p \text{ es primo}\}$ es listable. Utilizando un programa que implemente la criba de Eratóstenes, podemos ir probando uno por uno todos los naturales, y cada vez que detectamos un número primo, imprimirlo.

⁵ D, F y q_b dependen de la máquina en particular.

Ejemplo. Todo conjunto computable $C \subseteq \Sigma^*$ es listable (esto es lo que probamos en el inciso *ii*) de la Proposición 1.2.6). Dada una MT total M para decidir a C , podemos construir un programa que genere en orden lexicográfico todas las palabras en Σ^* , y pruebe cada palabra con la máquina M , y cada vez que está acepte una palabra, imprimirla.

Como hemos mencionado, las Máquinas enumeradoras y las MTs tiene la misma capacidad computacional.

Teorema 1.3.2. *La familia de conjuntos enumerados (listables) por máquinas enumeradoras es precisamente la familia de conjuntos c.e. En otras palabras, un conjunto L es $L(E)$ para alguna máquina enumeradora E si y sólo si L es $L(M)$ para alguna MT M .*

Así pues, todos estos modelos de máquinas son equivalentes al modelo dado en la Definición 1.2.2. El lector interesado en las demostraciones de estos hechos puede consultar [Ko97, Si05, Co04, Pa94].

1.4. Universalidad, autorreferencia e indecidibilidad

Hemos dicho que el concepto de MT (total) es nuestra definición formal de lo que entendemos intuitivamente como algoritmo, pero, también hemos dicho, que la MT es equivalente en poder computacional a una computadora estándar. Hasta el momento, el lector podría, y con toda razón, rechazar esta última afirmación. El motivo es que, en la forma en que definimos una MT, sólo es capaz de ejecutar una tarea específica, resolver un problema dado, y no más. Pero en esta sección, veremos que existe una MT, la *Máquina Universal de Turing*, que es capaz de recibir como entrada la especificación de cualquier otra MT, y ejecutarla (de la misma forma en que una computadora recibe un programa para ejecutarlo). La existencia de esta máquina nos lleva inmediatamente a descubrir problemas, sencillos en su especificación, pero imposibles de resolver en general.

1.4.1. La Máquina Universal de Turing

El artículo de Turing [Tu36] en 1936 es importante, no tanto por el hecho de haber inventado la MT, ya había muchos formalismos del concepto de

1.4. UNIVERSALIDAD, AUTORREFERENCIA E INDECIDIBILIDAD 27

algoritmo. Lo interesante fue la descripción de una *máquina universal*, que era capaz de llevar a cabo cualquier algoritmo sin necesidad de algún objeto adicional. Esta máquina ha jugado un papel fundamental en el desarrollo de la computadora estándar como la conocemos.

Para mostrar la existencia de la Máquina Universal de Turing (MUT), primero necesitamos un alfabeto y una codificación en palabras de todas las MTs. Escojamos el conjunto $\{0, 1\}$ como alfabeto. Sea $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$ una MT cualquiera, la codificación debe ser suficientemente simple para que la máquina universal pueda leerla, y debe especificar toda la información asociada con M (estados, alfabetos, estado de aceptación, rechazo e inicial). Supongamos sin pérdida de generalidad, que los conjuntos Q, Σ y Γ de M están dados en términos de números naturales, de la siguiente forma:

$$Q = \{1, 2, \dots, n\},$$

$$\Gamma = \{1, 2, \dots, m\},$$

$$\Sigma = \{1, 2, \dots, k\} \quad (k \leq m).$$

Con esto, s, t y r también son números naturales y haciendo $R = 1, L = 2$, la función δ se vuelve un subconjunto finito de \mathbb{N}^5 . En esta forma, es sencillo encontrar una codificación en el alfabeto $\{0, 1\}$ para M . La séxtupla $(Q, \Sigma, \Gamma, s, t, r)$ está totalmente determinada en la palabra

$$1^n 0 1^k 0 1^m 0 1^s 0 1^t 0 1^r 0$$

la cual nos indica que M tiene n estados, representados por los números 1 a n ; tiene un alfabeto de cinta con m símbolos, representados por los números 1 a m , de los cuales, los primeros k representan los símbolos del alfabeto de entrada; los estados de inicio, aceptación y rechazo son representados por los números s, t y r , respectivamente (Nótese que estamos usando al símbolo 0 como un separador de nuestras palabras de 1's, las cuales representan los números n, k, m, \dots etc.). Aún nos falta especificar lo más importante: el programa δ , pero hacerlo es sencillo, el resto de la palabra que codifica a M consiste de una sucesión de subpalabras que especifican las transiciones de δ . Por ejemplo, la subpalabra $1^p 0 1^a 0 1^q 0 1^b 0 1$ diría que δ contiene la transición $\delta(p, a) = (q, b, R)$ ($R = 1, L = 11$). De esta forma, M está completamente codificada en una palabra del alfabeto $\{0, 1\}$.

Ejemplo. Usemos la codificación de MTs propuesta para representar en el alfabeto $\{0, 1\}$ a la MT del Ejemplo 1.2.10. Lo primero que debemos hacer es renombrar a los elementos de los conjuntos Q , Σ y Γ como números naturales, lo cual hacemos de la siguiente forma: Para el conjunto de estados Q hacemos la asignación

$$\begin{aligned} s &\rightarrow 1, \\ q_1 &\rightarrow 2, \\ q_2 &\rightarrow 3, \\ q_3 &\rightarrow 4, \\ t &\rightarrow 5, \\ r &\rightarrow 6. \end{aligned}$$

Al conjunto Γ lo transformamos en

$$\begin{aligned} 0 &\rightarrow 1, \\ 1 &\rightarrow 2, \\ \sqcup &\rightarrow 3. \end{aligned}$$

Con esto tenemos que $\Sigma = \{1, 2\}$. El programa δ queda así (en este caso, para propósitos del ejemplo, colocamos transiciones en los espacios que tenían “.”, pero no es necesario que la función δ sea total):

δ	1	2	3
1	(6, 1, 1)	(2, 2, 1)	(6, 1, 1)
2	(3, 2, 1)	(2, 2, 1)	(6, 1, 1)
3	(6, 1, 1)	(3, 2, 1)	(4, 3, 2)
4	(1, 1, 1)	(5, 3, 1)	(1, 1, 1)
5	(1, 1, 1)	(1, 1, 1)	(1, 1, 1)
6	(1, 1, 1)	(1, 1, 1)	(1, 1, 1)

Por lo que M viene dada por la palabra

$1^6 01^2 01^3 0101^5 01^6 010101^6 01010101^2 01^2 01^2 010101^3 01^6 010101^2 0101^3 01^2 010$
 $1^2 01^2 01^2 01^2 0101^2 01^3 01^6 010101^3 0101^6 010101^3 01^2 01^3 01^2 0101^3 01^3 01^4 01^3 01^2$
 $01^4 0101010101^4 01^2 01^5 01^3 0101^4 01^3 01010101^5 0101010101^5 01^2 01010101^5 01^3$
 $01010101^6 0101010101^6 01^2 01010101^6 01^3 010101.$

Esta palabra es pues, el resultado de “compilar” nuestra MT (programa) a el lenguaje que entiende la computadora (la MUT) para que lo pueda ejecutar.

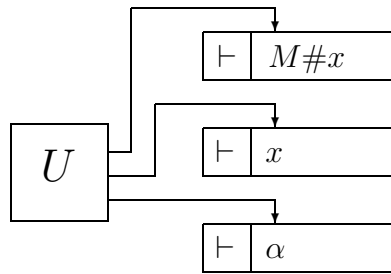


Figura 1.6: La Máquina Universal de Turing. Nótese el uso de los metasímbolos $M\#x$ y x , para denotar la codificación en el alfabeto $\{0, 1\}$ de la máquina M y la palabra x .

Una vez que hemos escogido la codificación de máquinas de Turing, podemos construir la MUT U , como una máquina de 3-cintas (Figura 1.6). En la primera cinta, U guarda su palabra de entrada, que es una descripción de una MT M y una palabra x (codificada en el alfabeto $\{0, 1\}$) sobre el alfabeto de entrada de M ; la segunda cinta sirve para mantener el contenido actual de la cinta de M y en la tercer cinta, U guarda cálculos temporales para ejecutar a M en x . Podemos mostrar que el lenguaje aceptado por U es

$$L(U) = \{M\#x \in \{0, 1, \#\}^* \mid x \in L(M)\}.$$

El símbolo “#” es un elemento en el alfabeto de U , distinto de 0 y 1, que usamos para separar a M y x .

Con la existencia de la MUT, se fortalece nuestra confianza en la Tesis de Church-Turing, y nuestra creencia de que las MTs son equivalentes a las computadoras modernas.

1.4.2. Problemas no computables

Con un argumento sencillo de conteo, es fácil ver que, ya que existen más problemas que soluciones (MTs), entonces no existen algoritmos para *casi todos* los problemas. Esto no sería preocupante, si los conjuntos que no son computables encerrarán objetos y propiedades que no tienen importancia en

las matemáticas. Veremos ahora que varios de estos conjuntos están altamente conectados con las matemáticas de cada día.

El problema de detención

La existencia de la Máquina Universal de Turing hace que descubramos muchos conjuntos relacionados con MTs que no son computables. El primero de estos, es el conocido como *El problema de detención* (halting problem) para máquinas de Turing, el cual nos pregunta, dados una MT M cualquiera y una palabra $x \in \Sigma^*$, si existe un algoritmo con el que podamos decidir si M se detiene en x (con aceptación ó rechazo). Dicho en palabras más conocidas: Dado un programa P (en nuestro lenguaje de programación favorito) y una posible entrada (la palabra x), ¿Existe otro programa que nos de como salida “sí”, si P se va a ejecutar por un tiempo finito al darle como entrada los datos representados por x , y “no”, si P entra en un ciclo infinito con x ? Por supuesto, el programa que buscamos no debe ejecutar a P en x , pues así no obtendríamos un algoritmo. Formalmente, buscamos una MT total para decidir el conjunto

$$\text{HP} = \{M\#x \in \{0, 1, \#\}^* \mid M \text{ se detiene en } x\}.$$

Turing [Tu36] demostró que HP no es computable, usando la MUT y una versión modificada de la técnica de diagonalización, inventada por Cantor a finales del siglo XIX, para demostrar que no existe una función biyectiva entre los números naturales y los reales. Sin embargo, no es difícil probar que HP es c.e., simplemente usamos la máquina U para simular a M en x , aceptando si M se detiene en x , y si M entra en un ciclo con x , U también se quedará en un ciclo, pero esto no importa, ya que sólo nos interesa aceptar las palabras $M\#x$ que están en HP. Sin embargo, el siguiente conjunto

$$\text{HP}_T = \{M\#x\#t \in \{0, 1, \#\}^* \mid M \text{ se detiene en } x \text{ a lo más en } t \text{ pasos}\},$$

es computable, la máquina U simplemente simula a M en x durante t pasos, aceptando si dentro de ese tiempo M se detuvo en x , y rechazando en caso contrario.

Otro problema muy parecido a HP, es el llamado *problema de pertenencia* (membership problem) MP, el cual pregunta si es posible, dada una MT M

y una palabra x , decidir si x es aceptada por M . Formalmente, nos preguntamos si el conjunto $MP = \{M\#x \in \{0, 1, \#\}^* \mid x \in L(M)\}$ es computable. Se puede mostrar que, si MP es computable, entonces HP es computable, lo cual contradice lo que hemos dicho de HP , por lo que MP también es no computable. A partir de la indecidibilidad de HP y MP , es posible demostrar que otras propiedades relacionadas con MTs también son indecidibles. Por ejemplo, dada una MT, es indecidible si

- acepta a la palabra vacía,
- acepta cualquier palabra,
- acepta un conjunto finito,
- es equivalente a otra MT con una descripción mas corta (esto es, determinar, dado un programa en Pascal, C o Java, si existe otro programa, en el mismo lenguaje de programación, más chico y equivalente al primero).

El problema de la decidibilidad en lógica

Kurt Gödel demostró el Teorema de incompletez al construir, para cualquier sistema formal que contiene a la aritmética, una proposición que afirmaba su propia no demostrabilidad. Era una proposición que hablaba de sí misma, hacía referencia a ella misma. Turing, en su artículo de 1936 [Tu36], hizo una demostración del Teorema de incompletez, utilizando sus máquinas; la clave está en que el poder de hacer autorreferencia se encuentra presente en las máquinas de Turing (y en todos los lenguajes de programación). Pongamos el problema en los términos en los que hemos estado hablando: Dividamos primero todas las proposiciones de la Teoría de números en dos subconjuntos disjuntos, las que son ciertas y las que son falsas. Dentro de las que son ciertas, encontramos al conjunto de proposiciones que se pueden demostrar, es decir, los teoremas. Entonces, es posible demostrar que

- i) el conjunto de teoremas es c.e., pero
- ii) el conjunto de todas las proposiciones que son ciertas no lo es,

por lo tanto, los dos conjuntos no pueden ser iguales, y el sistema formal no puede ser completo. De esta forma, Turing [Tu36] mostró que el problema de

decidibilidad (Entscheidungsproblem) no es computable⁶. Pero si agregamos una variante al problema, específicamente, si nos dan una proposición P , junto con ciertos axiomas y un natural t , y nos preguntamos si existe una demostración de P , de a lo más t símbolos, usando los axiomas dados, esto sí lo podemos decidir. Notamos la gran similitud entre este problema y el conjunto HP_T , que es también una variante decidible de HP.

El problema de las ecuaciones diofantinas

Una ecuación *diofantina* es una ecuación polinomial (igualada a cero) en una o más variables, con coeficientes enteros. Consideremos ahora el siguiente problema: Dada una ecuación diofantina $p(x_1, \dots, x_n) = 0$, encontrar una solución (a_1, \dots, a_n) tal que cada a_i es un entero. Este es uno de los problemas más antiguos de la matemática, mencionado en el libro *Aritmética* del matemático griego Diofantus, y es la base para uno de los problemas propuestos por Hilbert en su programa, el décimo problema de Hilbert, que dice así:

Encontrar una forma general para decidir efectivamente si una ecuación diofantina tiene una solución ó no.

Para contestarlo, se necesitaron matemáticas del siglo XX, y la respuesta no fue la que Hilbert esperaba. Aunque hay algoritmos para decidir si ecuaciones de la forma $ax + by = c$ y varias formas cuadráticas tienen soluciones enteras, no existe un algoritmo general que funcione en todos los casos.

La estrategia para demostrar que el décimo problema de Hilbert no tenía una solución afirmativa, fue resultado de un trabajo conjunto de muchos matemáticos. Primero hay que traducir el problema al ámbito de conjuntos de números enteros (que ya vimos, los podemos codificar en palabras en $\{0, 1\}^*$), definiendo lo que es un *conjunto diofantino*.

Definición 1.4.1. Un conjunto $A \subseteq \mathbb{Z}$ es diofantino si existe un polinomio $p(t, x_1, \dots, x_n) \in \mathbb{Z}[t, x_1, \dots, x_n]$ tal que

$$A = \{a \in \mathbb{Z} \mid (\exists x_1, \dots, x_n \in \mathbb{Z}) p(a, x_1, \dots, x_n) = 0\}.$$

⁶El lector puede encontrar una versión muy amigable de la demostración de Turing en [Ko97, p287-291].

1.4. UNIVERSALIDAD, AUTORREFERENCIA E INDECIDIBILIDAD 33

Por ejemplo, el conjunto de números naturales $\mathbb{N} \subseteq \mathbb{Z}$ es diofantino, pues dado $a \in \mathbb{Z}$, por el *Teorema de los cuatro cuadrados* de Lagrange tenemos que

$$a \in \mathbb{N} \Leftrightarrow (\exists x_1, x_2, x_3, x_4 \in \mathbb{Z}) x_1^2 + x_2^2 + x_3^2 + x_4^2 = a.$$

No es difícil ver que todo conjunto diofantino A es c.e., pues en la entrada $a \in \mathbb{Z}$ (codificado en una palabra en $\{0, 1\}$, por supuesto) enumeramos todas las tuplas de enteros (x_1, \dots, x_n) y probamos una por una si satisfacen la correspondiente ecuación $p(t, x_1, \dots, x_n) = 0$ que caracteriza a A . Si en efecto, $a \in A$, entonces encontraremos la tupla que satisfaga la ecuación. La solución negativa al décimo problema de Hilbert se deriva de un hecho muy importante que fue demostrado en varias etapas por los matemáticos Martin Davis, Hilary Putnam, Julia Robinson y Yuri Matiyasevich.

Teorema 1.4.2 (Davis, Putman, Robinson, Matiyasevich 1970). *Sea $A \subseteq \mathbb{Z}$. Entonces A es c.e. si y sólo si A es diofantino.*

Este notable resultado es conocido como el teorema DPRM. Como hay conjuntos c.e. que no son computables, se obtiene como consecuencia que el problema de las ecuaciones diofantinas es indecible. La historia completa y todos los detalles de este famoso resultado pueden ser encontrados en [Ma93, Re86]. Pero las consecuencias del Teorema 1.4.2 no terminan aquí. Podemos generalizar el problema de las ecuaciones diofantinas y plantearlo sobre otros anillos, como $\mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{R}(t)$ y muchos otros, y preguntarnos si existe un algoritmo para resolver el décimo problema de Hilbert sobre el anillo dado. Las respuestas encontradas hasta ahora son sorprendentes y muy variadas. Se tienen soluciones positivas en \mathbb{R} y \mathbb{C} ; negativas en $\mathbb{R}(t)$; y aun no se sabe una respuesta en \mathbb{Q} . Para más detalles de esto y una exposición muy amigable, vease [Poo08]⁷.

Veremos más problemas no computables que ocurren en las matemáticas de todos los días en los siguientes capítulos, particularmente en álgebra, y como consecuencia de esto, en topología.

⁷Hasta se está realizando una película respecto a este problema histórico para las matemáticas. Los detalles en <http://www.zalafilms.com/>.

1.5. Grados de Turing: Clasificando lo imposible

Hasta este momento, tenemos una clasificación de problemas computacionales que los divide en 2 tipos: Los problemas que se pueden resolver (los computables), y los que no tienen algoritmo que los resuelva (los no computables). En complejidad computacional, una rama de la teoría de la computación, se busca desarrollar una clasificación de los problemas que son computables [Pa94, Si05]. ¿Qué hay de los conjuntos no computables? Sabemos que no existe algoritmo que los pueda resolver, pero no nos conformamos solo con esto. Nos gustaría saber, si es posible, si todos tienen la misma dificultad, es decir, si son “igual de imposibles”, o algunos son “menos imposibles” que otros. Preguntas de este tipo, es posible contestarlas usando los conceptos de computación relativa y grados de Turing, que presentaremos a continuación.

1.5.1. Máquinas oráculo

Introducimos las máquinas de Turing con oráculos, las cuales fueron definidas por primera vez por Turing [Tu39], quien las tomó como un tema secundario, y que sin embargo, han sido fundamentales en el estudio de la computación.

Definición 1.5.1. Sea $A \subseteq \Sigma^*$. Una Máquina de Turing con Oráculo (MTO) A , denotada por M^A , es una sucesión de diez términos $M^A = (Q, \Sigma, \Gamma, \delta, s, t, r, q_O, q_s, q_n)$, donde

- Q, Σ, Γ, s, t y r son como en la Definición 1.2.2;
- $\delta: Q \times \Gamma^2 \rightarrow Q \times \Gamma^2 \times \{R, L\}^2$ es la función de transición (M^A tiene dos cintas);
- $q_O \in Q$ es el estado de petición al oráculo;
- $q_s, q_n \in Q$ son los estados de aceptación y rechazo del oráculo.

Cada vez que la máquina entra en su estado q_O de petición al oráculo A , cambia solamente a alguno de los estados q_s ó q_n (dependiendo de la respuesta de A), es decir, δ contiene transiciones de la forma

$$\delta(q_O, (a, b)) = (q_s, (a, b), (D_1, D_2)),$$

$$\delta(q_O, (a, b)) = (q_n, (a, b), (D'_1, D'_2)),$$

para algunos $a, b \in \Gamma$ y $D_1, D_2, D'_1, D'_2 \in \{R, L\}$. Estas son las únicas transiciones que involucran cambios de q_O hacia otros estados.

Intuitivamente, una Máquina de Turing con Oráculo A es una MT estándar, que además viene equipada con una cinta especial, llamada la cinta de consulta. Cada vez que la máquina escribe una palabra en esta cinta, entra en su estado especial de petición, y a continuación obtiene de forma instantánea una respuesta del oráculo A , la cual es reflejada en la transición a uno de los estados de aceptación ó rechazo del oráculo. No hay algo misterioso acerca de las MTOs, se comportan exactamente igual que las MTs ordinarias, la única diferencia radica en el uso del oráculo. Una máquina con oráculo $A = \emptyset$ es equivalente a una MT ordinaria. La siguiente definición es la versión relativizada de la Definición 1.2.5.

Definición 1.5.2. Sean $A, B \subseteq \Sigma^*$. Decimos que A es computable enumerablemente en B , si existe una MTO M^B , tal que $A = L(M^B)$. Si además, M^B es total, entonces decimos que A es computable en B o que A es Turing reducible a B , y lo denotamos por $A \leq_T B$.

Proposición 1.5.3. La relación \leq_T en el conjunto 2^{Σ^*} es reflexiva y transitiva.

Demostración. Sean $A, B, C \subseteq \Sigma^*$ lenguajes. Reflexividad: Construimos una MT con oráculo M^A tal que, en la entrada $x \in \Sigma^*$, de inmediato pregunta al oráculo si $x \in A$, aceptando si la respuesta es afirmativa, y rechazando en caso contrario.

Transitividad: Supongamos que $A \leq_T B$ y $B \leq_T C$, entonces existen MTOs M^B y N^C , tal que M^B puede decidir si una palabra x esta en A con un número finito de preguntas a su oráculo de la forma ¿Está y_1 en B ?, ..., ¿Está y_n en B ?. Similarmente, N^C decide si x esta en B con un número finito de preguntas a su oráculo de la forma ¿Está w_1 en C ?, ..., ¿Está w_m en C ? Queremos construir una MTO K^C , tal que decida al conjunto A con un número finito de preguntas al oráculo C . La máquina K^C puede ser construida reutilizando el programa de la máquina M^B , reemplazando las rutinas de llamada al oráculo B , con el programa de la máquina N^C . Así, en la entrada $x \in \Sigma^*$, K^C (siguiendo el programa de M^B) haría cálculos hasta generar la palabra y_1 , y a continuación, ejecuta como subrutina el programa de N^C , con lo que usando al oráculo C , determina si y_1 está en B ,

dependiendo de este resultado, hace cálculos para generar a y_2 , y nuevamente usa la subrutina del programa de N^C para saber si $y_2 \in B$. Repitiendo este proceso con las palabras y_3, \dots, y_n , K^C finalmente podrá determinar si $x \in A$ haciendo un número finito de preguntas al oráculo C . Como M^B y N^C son MTO totales, K^C también es total, con lo que tenemos que $A \leq_T C$. \square

Podemos construir, como en el caso de las MTs estándar, una máquina universal, que puede ejecutar cualquier máquina con oráculo, y podemos formar la versión relativizada de HP y MP, y no obtendríamos resultados distintos a los que sabemos en el caso normal. La indecidibilidad también se hace presente en el mundo de la computación relativa. Veamos algunos resultados en este ámbito.

Proposición 1.5.4. *Sea Σ un alfabeto.*

- i) Si $A \subseteq \Sigma^*$ es c.e. entonces $A \leq_T HP$.*
- ii) $HP \leq_T MP$.*
- iii) Para cada $B \subseteq \Sigma^*$, tenemos que $B \leq_T \Sigma^* - B$.*

Demostración. *i).* Sea $A \subseteq \Sigma^*$ un conjunto c.e. y sea M una MT tal que $A = L(M)$ y $x \in \Sigma^*$, entonces podemos modificar a M de tal manera que, cuando entre en su estado de rechazo r , M entre en un ciclo trivial (agregamos también un nuevo estado de rechazo, r' , inaccesible). Llamemos a esta nueva máquina M' ; luego, preguntamos al oráculo HP si M' se detiene en x . Si la respuesta es afirmativa, entonces M acepta a x y por lo tanto, $x \in A$. Si la respuesta es negativa, entonces M rechaza o entra en un ciclo con x y así $x \notin A$. Concluimos que $A \leq_T HP$.

ii). En efecto, podemos construir una MTO tal que decide al conjunto HP haciendo un número finito de preguntas al oráculo MP, he aquí como: Dada la palabra $M\#x$, primero preguntamos al oráculo MP si M acepta a x . Si la respuesta es afirmativa, entonces aceptamos; en caso contrario, intercambiamos los estados de aceptación y rechazo de M para obtener una nueva máquina M'' , y preguntamos al oráculo si M'' acepta a x . Si la respuesta es sí, entonces M rechaza a x , y por lo tanto, se detiene en x , así que aceptamos. Si la respuesta es no, entonces M no acepta ni rechaza a x , por lo tanto entra

en ciclo con x , por lo que rechazamos. Esto demuestra que $HP \leq_T MP$.

iii). Sea $B \subseteq \Sigma^*$ cualquier lenguaje. Construimos una MT con oráculo B tal que, en la entrada x , preguntamos al oráculo si $x \in B$, aceptando si no está, y rechazando en caso contrario. Ya que $x \in B$ ó $x \in \Sigma^* - B$, la máquina construida de esta forma es total y por lo tanto $B \leq_T \Sigma^* - B$. \square

1.5.2. El universo de Turing

Con las máquinas oráculo, estamos en posición de introducir la herramienta básica que nos permitirá dar una clasificación de los problemas no decidibles. La reducibilidad de Turing nos permite entrar en lo que llamamos el *Universo de Turing*.

Definición 1.5.5. Sean $A, B \subseteq \Sigma^*$. Decimos que A es Turing equivalente a B , si $A \leq_T B$ y $B \leq_T A$, y lo denotamos por $A \equiv_T B$. Esta es una relación de equivalencia. El grado de Turing (o grado de insolubilidad) de A es su clase de equivalencia, $\deg A = [A]$, bajo \equiv_T . Al conjunto $2^{\Sigma^*} / \equiv_T$, de todos los grados de Turing, lo llamamos \mathcal{D} .

La noción de grado de insolubilidad fue introducida por vez primera por Emil Post en 1944 [Po44] y desde entonces, ha sido un área de la teoría de la computación de intensa investigación, para tratar de entender mejor su estructura. Hablaremos brevemente acerca de algunas de sus propiedades, ya que el tema es muy extenso y podría alargar demasiado la obra. Siguiendo la nomenclatura acostumbrada en la literatura, usamos letras minúsculas en negritas (**a**, **b**, **c**, ...) para denotar grados de insolubilidad. El conjunto \mathcal{D} tiene las siguientes propiedades básicas [Co04]⁸:

- Todo grado de Turing contiene exactamente \aleph_0 elementos.
- El cardinal de \mathcal{D} es 2^{\aleph_0} .

El grado de Turing del conjunto vacío $\deg \emptyset = \{B \subseteq \Sigma^* \mid B \equiv_T \emptyset\}$ lo denotamos por $\mathbf{0}$. Este grado es precisamente la colección de todos los conjuntos computables, ya que, una MT M que usa el oráculo vacío, no tiene

⁸El mencionar algunas de las tantas propiedades de los conjuntos \mathcal{D} y \mathcal{E} (mencionado al final del capítulo) es únicamente con propósitos informativos, no usaremos nada de esto a lo largo de la obra.

información adicional, pero ya que se detiene en todas las posibles entradas, debe ser una MT total, y así $L(M)$ es computable. Al grado del problema de detención, HP, lo denotamos por $\mathbf{0}'$. Este también es un grado con problemas muy especiales. Mencionamos que, todo problema c.e. se Turing reduce a HP, por lo tanto, cualquier problema Turing equivalente a HP tiene esta propiedad (MP, por la Proposición 1.5.4 incisos *i*) y *ii*); las teorías incompletas de Gödel, entre otros). A estos problemas les decimos completos con respecto a la Turing reducción en la clase de los lenguajes c.e., ya que si existiera un algoritmo para resolver alguno de ellos, entonces todos los conjuntos c.e. serían computables (lo cual, ya sabemos que no es posible, pues al ser HP indecidible, todos los problemas Turing equivalentes a el deben ser indecibles). Una técnica muy usual para demostrar que un problema dado A es indecidible, es mostrar que HP es Turing reducible a A .

La colección de los grados de Turing, \mathcal{D} , tiene una relación de orden, \leq , definida por $\mathbf{a} \leq \mathbf{b} \Leftrightarrow A \leq_T B$, donde $\deg A = \mathbf{a}$ y $\deg B = \mathbf{b}$ (por supuesto, esta relación esta bien definida, $B_0 \leq_T A_0 \Leftrightarrow B_1 \leq_T A_1$, si $A_0 \equiv_T A_1$ y $B_0 \equiv_T B_1$), que convierte a \mathcal{D} en un conjunto parcialmente ordenado (\mathcal{D}, \leq) , que tiene entre otras, las siguientes propiedades [AF06, Co04]:

- El grado $\mathbf{0}$ es el más pequeño de todos los grados de Turing.
- Existen en \mathcal{D} grados mínimos. Un grado \mathbf{a} es mínimo si $\mathbf{a} \neq \mathbf{0}$ y no existe algún grado entre $\mathbf{0}$ y \mathbf{a} .
- Existen parejas de grados de Turing que no tienen cota inferior más grande, es decir, \mathcal{D} no es una retícula.
- Toda pareja de grados de Turing tiene cota superior más pequeña, es decir, \mathcal{D} es una semi-retícula superior.

Dados estos hechos, concluimos que la estructura de (\mathcal{D}, \leq) es muy complicada.

1.5.3. Grados de Turing computables enumerablemente

Los problemas c.e. juegan un papel muy importante en las matemáticas. La mayoría de los problemas indecibles que se han encontrado fuera de la

teoría de la computación, son computables enumerablemente. En el campo de estudio de los grados de Turing, se ha puesto especial atención a los grados que contienen conjuntos c.e.

Definición 1.5.6. Un grado de Turing $\mathbf{a} \in \mathcal{D}$ es computable enumerablemente (c.e.), si contiene un lenguaje c.e. Denotamos por \mathcal{E} a la colección de todos los grados de Turing c.e.

Los grados $\mathbf{0}$ y $\mathbf{0}'$ que mencionamos anteriormente, son grados c.e. Por supuesto, no son los únicos, existen muchos más. Algunas propiedades de \mathcal{E} son [AF06, Co04]:

- El conjunto \mathcal{E} es denso, es decir, entre cualesquiera dos grados de Turing c.e., siempre existe un tercer grado de Turing c.e.
- Todo grado c.e. es menor o igual a $\mathbf{0}'$ (aunque, no todo grado menor a $\mathbf{0}'$ es c.e.).
- Existe una pareja de grados c.e. que no tienen cota inferior más grande (\mathcal{E} no es una retícula).
- \mathcal{E} es una semi-retícula superior.
- Toda retícula distributiva finita puede ser encajada en \mathcal{E} .

Post [Po44] introdujo el problema de determinar si existe un conjunto c.e. E , tal que $\mathbf{0} < \deg E < \mathbf{0}'$ (es decir, un problema E c.e. tal que E no es computable y además HP no es Turing reducible a E). Este ha sido uno de los problemas más influyentes y fundamentales en el estudio de los grados de Turing, y es conocido como *el Problema de Post*, y fue resuelto en los 50's de manera independiente por Friedberg [Fr57] y Mučnik [Mu56]. Es interesante hacer notar, que ambos desarrollaron la misma técnica para construir el conjunto c.e. buscado. Esta técnica es conocida en el mundo de la computación como *el método de prioridad*, y es una de las técnicas principales para establecer resultados sobre conjuntos y grados c.e.

La discusión empieza a rebasar el alcance de este capítulo, así que debemos detenernos aquí. El lector interesado en profundizar más en las máquinas oráculo y los grados de Turing, puede consultar [AF06, Co04]. En el resto de la obra, trataremos exclusivamente con los grados de Turing c.e., pues es precisamente dentro de \mathcal{E} donde encontraremos clasificados los problemas topológicos de los que hablaremos.

Capítulo 2

Álgebra

Después de haber introducido el marco de trabajo necesario para desarrollar resultados sobre teoría de la computación, es tiempo de que presentemos la semilla con la que podremos cosechar la indecidibilidad de la clasificación de las variedades. Esta semilla es dada por el álgebra, en particular, por la teoría de grupos; para esto, nos será necesario hablar de conceptos como las presentaciones de grupos, las cuales nos permitirán representar grupos, de tal forma que los podamos manipular computacionalmente, y hablar de la decidibilidad de varias de sus propiedades. La notación que usamos es totalmente estándar, es encontrada en libros como [Fr88, Ro95], y asumimos del lector un conocimiento básico en esta teoría. Aunque no supondremos que los grupos tratados son abelianos, denotaremos por 1 a la identidad y además usaremos notación multiplicativa.

2.1. Presentaciones de grupos

La idea de *presentación de grupo*, es formarlo a partir de un conjunto de generadores, los cuales estarán sujetos a ciertas ecuaciones o relaciones. Intuitivamente, el grupo presentado debe ser “tan libre” como sea posible en los generadores, sujetos a las relaciones dadas.

Definición 2.1.1. Sea G un grupo. Una presentación de G es una pareja $\mu = (S: R)$, tal que

- a) S es el conjunto de generadores de μ .

- b) R es el conjunto de relaciones de μ . Cada $r \in R$ es una sucesión finita¹ $x_1^{\varepsilon_1} x_2^{\varepsilon_2} \cdots x_n^{\varepsilon_n}$, donde $x_i \in S$ y $\varepsilon_i = \pm 1$.
- c) Sea $F(\mu)$ el grupo libre generado por los elementos de S , $N(\mu)$ el subgrupo normal generado por R . Entonces se debe cumplir que

$$G \cong F(\mu)/N(\mu).$$

Al grupo G , presentado por μ , lo denotamos por $G(\mu)$. Definimos a $r(\mu)$ y $p(\mu)$ como los cardinales de S y R respectivamente.

Muchas veces, se denotan a las relaciones r , de una presentación de grupo como ecuaciones $r = 1$. Un grupo puede tener más de una presentación (Para ver esto, tomamos $G = \mathbb{Z}_6$ y las presentaciones $(\{a\} : \{a^6\})$ y $(\{a, b\} : \{a^2, b^3, aba^{-1}b^{-1}\})$). En la siguiente tabla, mostramos varios ejemplos de presentaciones para algunos grupos conocidos.

Grupo	Presentación
El grupo trivial	$(\emptyset : \emptyset)$
El grupo libre en S	$(S : \emptyset)$
El grupo abeliano libre en S	$(S : \{xyx^{-1}y^{-1} \mid x, y \in S\})$
El grupo cíclico de orden n	$(\{a\} : \{a^n\})$
$\mathbb{Z} \times \mathbb{Z}$	$(\{x, y\} : \{xyx^{-1}y^{-1}\})$
$\mathbb{Z}_m \times \mathbb{Z}_n$	$(\{x, y\} : \{x^m, y^n, xyx^{-1}y^{-1}\})$
El grupo de Cuaterniones	$(\{i, j\} : \{i^4, ijij^{-1}, i^2j^2\})$
$GL_2(\mathbb{Z})$	$(\{a, b, j\} : \{aba(bab)^{-1}, (aba)^4, j^2, (ja)^2, (jb)^2\})$

Tabla 1: Ejemplos de presentaciones de grupos.

Decimos que una presentación es *finitamente generada*, si el conjunto de generadores es finito y *finitamente relacionada*, si el conjunto de relaciones

¹Por conveniencia, no requerimos como necesario listar las llamadas “relaciones triviales”, $x_i^{-1}x_i = 1$, en los elementos de R .

es finito. En caso de que una presentación sea finitamente generada y finitamente relacionada, entonces hablamos de una *presentación finita*. Estamos interesados en grupos, los cuales tienen presentaciones finitas, a los que llamamos grupos finitamente presentados. Surge entonces la pregunta, de si todo grupo tiene una presentación (finita).

Proposición 2.1.2. *Sea G un grupo. Entonces*

(a) *G tiene una presentación.*

(b) *Si G es finito, entonces G tiene una presentación finita.*

Demostración. Sea G un grupo. (a) Consideremos a $F(G)$, el grupo libre en G . Por la propiedad universal de grupos libres, existe un único homomorfismo $\varphi: F(G) \rightarrow G$, tal que $\varphi \circ i_G = 1_G$; claramente, φ es suprayectiva. Sea $K = \ker \varphi$, entonces, por el primer Teorema del isomorfismo, tenemos que $F(G)/K \cong G$, y por lo tanto, G tiene la presentación $(G: K)$. (b) Si $|G| < \infty$, entonces podemos usar su tabla de multiplicación para obtener una presentación finita. Tomamos como conjunto generador a G y como conjunto de relaciones todas las palabras de la forma $g_i g_j g_k^{-1}$, donde $g_i g_j = g_k$ es una entrada de la tabla de G . Es sencillo ver que estos conjuntos forman una presentación finita de G (usando un argumento totalmente análogo al anterior, o considerando el isomorfismo $\iota: G \rightarrow G(\mu)$, dado por $\iota(g) = gN(\mu)$), por lo tanto G , es finitamente presentado. \square

Sin embargo, existen grupos que no son finitamente presentados, el siguiente ejemplo de ello fue encontrado por W. W. Boone [Bo95]: Sea F el grupo libre con base $\{a, b\}$ y F' su subgrupo conmutador con base $\{\omega_n\}_{n \in \mathbb{N}}$. Entonces el grupo G con presentación

$$(\{a, b, p\}: \{p^{-1}\omega_n p = \omega_n \mid n \geq 1\})$$

es un grupo finitamente generado, pero no es finitamente presentado. Este es uno de varios resultados negativos que tendremos para grupos en lo que respecta a computación, pero antes de verlos, debemos introducir un par de conceptos. Ya mencionamos que si un grupo tiene una presentación, esta no necesariamente es única. Esto induce la siguiente

Definición 2.1.3. Sean μ, ν presentaciones de grupos. Decimos que μ y ν son presentaciones isomorfas, si $G(\mu) \cong G(\nu)$. Esto lo denotamos por $\mu \cong \nu$.

Observamos que, si consideramos a las presentaciones finitas de grupos como cadenas sobre el alfabeto $\Sigma(x_1, \dots, x_n) = \{x_1, \dots, x_n, x_1^{-1}, \dots, x_n^{-1}\}$, (donde los x_i 's son los generadores, y el conjunto de cadenas lo denotaremos por $\Sigma^*(x_1, \dots, x_n)$), entonces la propiedad de isomorfismo de presentaciones se convierte en una *propiedad de palabras*, la cual es equivalente a la propiedad de isomorfismo de los grupos presentados, por tanto, podemos preguntarnos de la decidibilidad de esta. Hay un concepto de equivalencia de presentaciones de grupos más fuerte que el de isomorfismo.

Definición 2.1.4. Sean μ, ν presentaciones de grupos. Decimos que μ y ν son *congruentes*, si son idénticas, salvo notación, es decir, si existe una correspondencia uno a uno entre los generadores de μ y los de ν que lleva las relaciones de μ en las de ν .

Ejemplo. Las presentaciones $(\{c\} : \{c^6\})$ y $(\{a, b\} : \{a^2, b^3, aba^{-1}b^{-1}\})$ del grupo \mathbb{Z}_6 son isomorfas, pero no congruentes, mientras que las presentaciones $(\{a, b, j\} : \{aba(bab)^{-1}, (aba)^4, j^2, (ja)^2, (jb)^2\})$ y $(\{x, y, z\} : \{xyx(yxy)^{-1}, (xyx)^4, z^2, (zx)^2, (zy)^2\})$ de $GL_2(\mathbb{Z})$ son congruentes (e isomorfas).

2.1.1. La relación vacía de Markov

Presentamos una pequeña herramienta que necesitaremos más adelante, cuando empecemos a hablar de computabilidad y topología. Mas que una herramienta, deberíamos decir, un “truco”, ya que esa es la manera en la que se le conoce. La idea es, dada una presentación finita de grupo μ , incrementar el número de relaciones de μ , de tal forma que no alteremos el grupo $G(\mu)$. Para esto, usaremos lo que se conoce como “la relación vacía”, el cual denotamos por $*$. Podríamos definir esta relación como $\lambda = 1$, que denota que la cadena vacía, λ , que es la identidad en $F(\mu)$, es equivalente a la identidad en $G(\mu)$ (lo cual siempre pasa), con lo que la relación vacía $*$ no tiene ningún efecto cuando la colocamos en alguna palabra de $F(\mu)$. Si $\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_m\})$, y $t \in \mathbb{N}$, definimos una nueva presentación, $\mu * t$, como

$$\mu * t = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_m, *^t\}),$$

donde $*^t$ significa que repetimos $*$ t veces. Como ya mencionamos, con la relación vacía no alteramos el grupo representado por μ , por lo que $\mu \cong \mu * t$ para toda t , entonces ¿Para qué nos sirve esta relación vacía? Markov² [Ma58] introdujo a la relación vacía³ para poder obtener un resultado por demás importante de computabilidad en topología; volveremos a tocar este tema en el Capítulo 3.

2.1.2. Operaciones elementales en presentaciones de grupos

En esta sección introducimos operaciones [Ma58] sobre presentaciones finitas de grupos. Las necesitaremos más adelante en el Capítulo 5, donde jugarán un papel muy importante en varias demostraciones.

Definición 2.1.5. Sea μ una presentación finita de grupos. Definimos las siguientes operaciones elementales:

Op_1 : La presentación

$$\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i \equiv r' r'', r_{i+1}, \dots, r_m\})$$

es reemplazada por la presentación

$$\mu' = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r' x_j^\varepsilon x_j^{-\varepsilon} r'', r_{i+1}, \dots, r_m\}),$$

para alguna $j \in \{1, \dots, n\}$ y $\varepsilon = \pm 1$.

Op_1^{-1} : La operación inversa de Op_1 (borrar una subpalabra $x_j^\varepsilon x_j^{-\varepsilon}$ en una relación).

Op_2 : La presentación $\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i, r_{i+1}, \dots, r_m\})$ es reemplazada por la presentación

$$\mu' = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r'_i, r_{i+1}, \dots, r_m\}),$$

donde la palabra r'_i es una permutación cíclica de la palabra r_i .

²A. A. Markov ha sido un personaje también muy importante en el mundo de los algoritmos, aunque usualmente no es muy mencionado en los libros de Teoría de la computación. Dio un modelo de computo (que por supuesto, es equivalente a los modelos que dimos en el Capítulo 1) muy parecido al que dio Post, y probó la existencia de un *Algoritmo Universal* que podía ejecutar cualquier programa. Estos y otros resultados que obtuvo se pueden encontrar en su libro de teoría de algoritmos [Ma77].

³También conocido como “El truco de Markov”.

Op_3 : La presentación $\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i, r_{i+1}, \dots, r_m\})$ es reemplazada por $\mu' = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i^{-1}, r_{i+1}, \dots, r_m\})$.

Op_4 : La presentación $\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i, r_{i+1}, \dots, r_m\})$ es reemplazada por $\mu' = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_{i-1}, r_i r_j, r_{i+1}, \dots, r_m\})$, donde $j \in \{1, \dots, m\}$ y $j \neq i$.

Op_5 : La presentación $\mu = (\{x_1, \dots, x_n\} : \{r_1, \dots, r_m\})$ es reemplazada por $\mu' = (\{x_1, \dots, x_n, x_{n+1}\} : \{r_1, \dots, r_m, x_{n+1}r\})$, donde $x_{n+1} \neq x_i$ para toda $i \in \{1, \dots, n\}$, y $r \in \Sigma^*(x_1, \dots, x_n)$.

Op_5^{-1} : La operación inversa de Op_5 (borrar en μ a un generador x_j y una relación $r_i = x_j r$, donde las palabras $r_1, \dots, r_{i-1}, r, r_{i+1}, \dots, r_m$ no contienen alguna de las letras x_j o x_j^{-1}).

No es difícil ver que estas operaciones preservan la clase de isomorfismo de μ . Más aún, $Op_1^{\pm 1}$, Op_2 , Op_3 y Op_4 no cambian el alfabeto usado para μ , y por tanto tenemos que $F(\mu) = F(\mu')$ y $N(\mu) = N(\mu')$. En el caso de $Op_5^{\pm 1}$, observemos que la relación (agregada con Op_5 y borrada con Op_5^{-1}) $x_{n+1}r = 1$ implica que $r = x_{n+1}^{-1}$ en $G(\mu')$, por lo tanto, esta relación es equivalente a la relación trivial $x_{n+1}x_{n+1}^{-1} = 1$, con lo cual $G(\mu) \cong G(\mu')$ por medio del isomorfismo $f: G(\mu') \rightarrow G(\mu)$, dado por

$$f(x_i N(\mu')) = \begin{cases} x_i N(\mu) & \text{si } 1 \leq i \leq n, \\ r^{-1} N(\mu) & \text{si } i = n + 1. \end{cases}$$

Ahora probaremos un resultado respecto a estas operaciones y la relación vacía de Markov, que más adelante nos será de gran utilidad.

Teorema 2.1.6 (W. W. Boone, W. Haken y V. Poénaru). Sean $\mu = (\{x_1, \dots, x_r\} : \{\alpha_1, \dots, \alpha_p\})$ y $\mu' = (\{y_1, \dots, y_{r'}\} : \{\beta_1, \dots, \beta_{p'}\})$ dos presentaciones de grupos isomorfas, y sean $t, t' \in \mathbb{N}$, tales que $p + t - r = p' + t' - r'$, $t \geq p + r'$, y $t' \geq p' + r$. Entonces la presentación $\mu * t$ se puede transformar con una sucesión finita de operaciones $Op_1^{\pm 1}$, Op_2 , Op_3 , Op_4 , $Op_5^{\pm 1}$ en la presentación $\mu' * t'$.

Demostración. Sean μ, μ', t, t' con las condiciones del enunciado. Asumimos, sin pérdida de generalidad, que $p + r' \geq p' + r$. Sea $g: G(\mu) \rightarrow G(\mu')$ un isomorfismo de $G(\mu)$ sobre $G(\mu')$. El primer paso es transformar a $\mu * t$

y $\mu' * t'$ en las presentaciones $\mu_{\#}$ y $\mu'_{\#}$, ambas generadas por un mismo conjunto, $z_1, \dots, z_{r_{\#}}$, donde $r_{\#} = r + r'$, y z_1, \dots, z_r están “en correspondencia” con x_1, \dots, x_r y lo mismo para $z_{r+1}, \dots, z_{r_{\#}}$ y $y_1, \dots, y_{r'}$.

Sean $\xi_1, \dots, \xi_{r'} \in \Sigma^*(x_1, \dots, x_r)$, tales que los elementos $\xi_j N(\mu) \in G(\mu)$ satisfacen

$$\xi_j N(\mu) = g^{-1}(y_j N(\mu')), \quad 1 \leq j \leq r';$$

similarmente, sean $\eta_1, \dots, \eta_r \in \Sigma^*(y_1, \dots, y_{r'})$, tales que $\eta_j N(\mu') = g(x_j N(\mu))$, con $1 \leq j \leq r$. Transformamos a $\mu * t$ con $r_{\#} = r + r'$ operaciones Op_5 en la presentación

$$\mu_1 = (\{x_1, \dots, x_r, z_1, \dots, z_{r_{\#}}\} : \{\alpha_1, \dots, \alpha_p, z_1 x_1^{-1}, \dots, z_r x_r^{-1}, z_{r+1} \xi_1^{-1}, \dots, z_{r_{\#}} \xi_{r'}^{-1}, *^t\});$$

de la misma forma, transformamos a $\mu' * t'$ en la presentación

$$\mu'_1 = (\{y_1, \dots, y_{r'}, z_1, \dots, z_{r_{\#}}\} : \{\beta_1, \dots, \beta_{p'}, z_1 \eta_1^{-1}, \dots, z_r \eta_r^{-1}, z_{r+1} y_1^{-1}, \dots, z_{r_{\#}} y_{r'}^{-1}, *^{t'}\}).$$

A continuación, en μ_1 reemplazaremos una por una, todas las x_i 's en las palabras $\alpha_1, \dots, \alpha_p, \xi_1^{-1}, \dots, \xi_{r'}^{-1}$ con las z_i 's. Para lograrlo, primero aplicamos la operación Op_3 en cada relación $z_i x_i^{-1}$, para obtener la nueva relación $x_i z_i^{-1}$; despues, a las otras relaciones, distintos de los anteriores, que contengan alguna letra $x_i^{\pm 1}$, les “movemos” esta letra hacia la derecha (usando Op_2 y, posiblemente, Op_3), transformándolas en relaciones de la forma τx_i^{-1} ; luego, con Op_4 , obtenemos cadenas de la forma $\tau x_i^{-1} x_i z_i^{-1}$, y con Op_1^{-1} , τz_i^{-1} . Repetimos este proceso tantas veces como sea necesario (claramente es finito) hasta que eliminemos de estas relaciones a las x_i 's y sus inversas. De la misma manera, en μ'_1 , reemplazamos todas las y_j 's en $\beta_1, \dots, \beta_{p'}$ y en $\eta_1^{-1}, \dots, \eta_r^{-1}$ con las z_{r+j} . Ahora, con r (r') operaciones Op_5^{-1} , borramos los generadores x_i (y_j) y las relaciones $x_i z_i^{-1}$ ($y_j z_{r+j}^{-1}$) de μ_1 (μ'_1), y así obtenemos como resultado las presentaciones

$$\begin{aligned} \mu_{\#} &= (\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, *^t\}), \\ \mu'_{\#} &= (\{z_1, \dots, z_{r_{\#}}\} : \{\delta_1, \dots, \delta_{p_{\#}}, *^{t'}\}), \quad p_{\#} = p + r', \end{aligned}$$

donde $\gamma_1, \dots, \gamma_{p_{\#}}$ son obtenidas a partir de $\alpha_1, \dots, \alpha_p, z_{r+1} \xi_1^{-1}, \dots, z_{r_{\#}} \xi_{r'}^{-1}$, respectivamente, al reemplazar x_i con z_i , y $\delta_1, \dots, \delta_{p'+r}$ son obtenidas a partir

de $\beta_1, \dots, \beta_{p'}, z_1\eta_1^{-1}, \dots, z_r\eta_r^{-1}$ respectivamente, al reemplazar y_j por z_{r+j} ; y $\delta_{p'+r+1}, \dots, \delta_{p\#}$ son palabras vacías (o podríamos decir, más relaciones vacías, recordemos nuestra suposición inicial de que $p + r' \geq p' + r$).

Lo que sigue, es demostrar que en el grupo libre $F(\mu_\#) = F(\mu'_\#)$ generado por $z_1, \dots, z_{r\#}$, las δ_i 's generan el mismo subgrupo normal que las γ_i 's ($N(\mu_\#) = N(\mu'_\#)$). Para ver esto, consideremos los siguientes isomorfismos obvios:

$$\begin{aligned} g_1: G(\mu) &\rightarrow G(\mu_1), \text{ tal que } g_1(x_i N(\mu)) = x_i N(\mu_1), & 1 \leq i \leq r; \\ g'_1: G(\mu') &\rightarrow G(\mu'_1), \text{ tal que } g'_1(y_j N(\mu')) = y_j N(\mu'_1), & 1 \leq j \leq r'; \\ g_2: G(\mu_\#) &\rightarrow G(\mu_1), \text{ tal que } g_2(z_k N(\mu_\#)) = z_k N(\mu_1), & 1 \leq k \leq r\#; \\ g'_2: G(\mu'_\#) &\rightarrow G(\mu'_1), \text{ tal que } g'_2(z_k N(\mu'_\#)) = z_k N(\mu'_1), & 1 \leq k \leq r\#. \end{aligned}$$

Entonces la composición $g_\# = g_2^{-1}g'_1gg_1^{-1}g_2: G(\mu_\#) \rightarrow G(\mu'_\#)$ es un isomorfismo, y es tal que $g_\#(z_k N(\mu_\#)) = z_k N(\mu'_\#)$, pues, si $k \leq r$, tenemos que

$$\begin{aligned} g_2^{-1}g'_1gg_1^{-1}g_2(z_k N(\mu_\#)) &= g_2^{-1}g'_1gg_1^{-1}(z_k N(\mu_1)) \\ &= g_2^{-1}g'_1gg_1^{-1}(x_k N(\mu_1)) \\ &= g_2^{-1}g'_1g(x_k N(\mu)) \\ &= g_2^{-1}g'_1(\eta_k N(\mu')) \\ &= g_2^{-1}(\eta_k N(\mu'_1)) \\ &= g_2^{-1}(z_k N(\mu'_1)) \\ &= z_k N(\mu'_\#). \end{aligned}$$

Similarmente, si $k > r$, entonces

$$\begin{aligned} g_2^{-1}g'_1gg_1^{-1}g_2(z_k N(\mu_\#)) &= g_2^{-1}g'_1gg_1^{-1}(z_k N(\mu_1)) \\ &= g_2^{-1}g'_1gg_1^{-1}(\xi_{k-r} N(\mu_1)) \\ &= g_2^{-1}g'_1g(\xi_{k-r} N(\mu)) \\ &= g_2^{-1}g'_1(y_{k-r} N(\mu')) \\ &= g_2^{-1}(y_{k-r} N(\mu'_1)) \\ &= g_2^{-1}(z_k N(\mu'_1)) \\ &= z_k N(\mu'_\#). \end{aligned}$$

Por lo tanto, $g_\#(\tau N(\mu_\#)) = \tau N(\mu'_\#)$, donde $\tau \in \Sigma^*(z_1, \dots, z_{r\#})$, y así tenemos que $g_\#(N(\mu'_\#)N(\mu_\#)) = N(\mu'_\#)N(\mu'_\#) = N(\mu'_\#)$. Ya que también

$g_{\#}(N(\mu_{\#})) = N(\mu'_{\#})$ (haciendo τ igual a la identidad), entonces $N(\mu'_{\#})$ $N(\mu_{\#}) = N(\mu_{\#})$; por lo que $N(\mu'_{\#}) \subseteq N(\mu_{\#})$. Usando a $g_{\#}^{-1}$ e invirtiendo los papeles de $N(\mu_{\#})$ y $N(\mu'_{\#})$, obtenemos la otra inclusión, con lo que $N(\mu_{\#}) = N(\mu'_{\#})$. Esto nos da como consecuencia, que si $i = 1, \dots, p_{\#}$, entonces

$$\begin{aligned} \delta_i &= \prod_{k=1}^{n_i} T_{ik} \gamma_{u_{ik}}^{\varepsilon_{ik}} T_{ik}^{-1}, \\ \gamma_i &= \prod_{k=1}^{m_i} S_{ik} \delta_{v_{ik}}^{\zeta_{ik}} S_{ik}^{-1}, \end{aligned} \tag{2.1}$$

para algunas $T_{ik}, S_{ik} \in \Sigma^*(z_1, \dots, z_{r_{\#}})$, y $\varepsilon_{ik}, \zeta_{ik} = \pm 1$. Con esto, mostraremos que podemos transformar a la presentación $\mu_{\#}$, en la presentación $\mu_{\natural} = (\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, \delta_1, \dots, \delta_{p_{\#}}, *^{t-p_{\#}}\})$, usando las operaciones $Op_1^{\pm 1}, Op_2, Op_3$ y Op_4 (Notamos que la expresión $*^{t-p_{\#}}$ en μ_{\natural} tiene sentido, ya que por hipótesis $t - p_{\#} \geq 0$). De la misma forma, es posible transformar a $\mu'_{\#}$ en μ_{\natural} . Esto completará la demostración.

Transformemos primero a $\mu_{\#}$ en μ_{\natural} . Primero, observemos que la operación

$$\begin{aligned} \widehat{Op}: (\{x_1, \dots, x_r\} : \{\alpha_1, \dots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}\alpha_p\}) \\ \mapsto (\{x_1, \dots, x_r\} : \{\alpha_1, \dots, \alpha_{i-1}, \omega \alpha_i^{\varepsilon} \omega^{-1}, \alpha_{i+1}\alpha_p\}), \end{aligned}$$

donde $\omega \in \Sigma^*(x_1, \dots, x_r)$ y $\varepsilon = \pm 1$, puede ser llevada a cabo por una sucesión de operaciones Op_1, Op_2 y Op_3 . La operación inversa \widehat{Op}^{-1} , esta similarmente compuesta por operaciones Op_1^{-1}, Op_2 y Op_3 . Transformamos a $\mu_{\#}$ con \widehat{Op} en la presentación

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{u_{11}-1}, T_{11} \gamma_{u_{11}}^{\varepsilon_{11}} T_{11}^{-1}, \gamma_{u_{11}+1}, \dots, \gamma_{p_{\#}}, *^t\});$$

entonces, usando Op_4 en las relaciones $T_{11} \gamma_{u_{11}}^{\varepsilon_{11}} T_{11}^{-1}$ y $*$, tenemos

$$\begin{aligned} (\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{u_{11}-1}, T_{11} \gamma_{u_{11}}^{\varepsilon_{11}} T_{11}^{-1}, \gamma_{u_{11}+1}, \dots, \gamma_{p_{\#}}, \\ T_{11} \gamma_{u_{11}}^{\varepsilon_{11}} T_{11}^{-1}, *^{t-1}\}); \end{aligned}$$

despues, aplicamos \widehat{Op}^{-1} y obtenemos

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{u_{11}-1}, \gamma_{u_{11}}, \gamma_{u_{11}+1}, \dots, \gamma_{p_{\#}}, T_{11} \gamma_{u_{11}}^{\varepsilon_{11}} T_{11}^{-1}, *^{t-1}\}).$$

Repitiendo el mismo conjunto de operaciones con las relaciones apropiadas, podemos llegar a la presentación

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, T_{11}\gamma_{u_{11}}^{\varepsilon_{11}}T_{11}^{-1}T_{12}\gamma_{u_{12}}^{\varepsilon_{12}}T_{12}^{-1}, *^{t-1}\}).$$

Continuamos de la misma forma, aplicando las mismas operaciones $n_1 - 2$ veces (recordemos las igualdades en (2.1)) y esto nos da como resultado

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, \prod_{k=1}^{n_1} T_{1k}\gamma_{u_{1k}}^{\varepsilon_{1k}}T_{1k}^{-1} = \delta_1, *^{t-1}\}).$$

En forma similar, podemos obtener la presentación

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, \delta_1, \delta_2, *^{t-2}\}),$$

y finalmente, llegamos a

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\gamma_1, \dots, \gamma_{p_{\#}}, \delta_1, \dots, \delta_{p_{\#}}, *^{t-p_{\#}}\}) = \mu_{\natural}.$$

En el caso de la transformación de μ'_{\natural} en μ_{\natural} , aplicamos el mismo proceso anterior para llegar a la presentación

$$(\{z_1, \dots, z_{r_{\#}}\} : \{\delta_1, \dots, \delta_{p_{\#}}, \gamma_1, \dots, \gamma_{p_{\#}}, *^{t'-p_{\#}}\}).$$

Luego, usando nuestras hipótesis, tenemos que

$$\begin{aligned} t' - p_{\#} &= t' - p + r' \\ &= t' + t - r - p' - t' \\ &= t - r - p' \\ &\geq t - p_{\#}. \end{aligned}$$

Con lo que deducimos que $t' - p_{\#} \geq 0$, por lo que la presentación anterior tiene sentido, y también vemos que puede tener relaciones vacías de mas. Pero esto no es ningún problema, simplemente las removemos hasta que nos quedamos con $t - p_{\#}$, y así obtenemos la presentación μ_{\natural} . Esto completa la demostración. □

2.2. Computabilidad en grupos

Ahora veremos varios resultados acerca de lo que podemos hacer y no podemos hacer en grupos. Entre los problemas de los que hablaremos, están dos que son los ingredientes básicos para deducir la no computabilidad de la clasificación de las variedades. Ya sabemos que no todo grupo puede ser finitamente presentado, pero ahí no termina la historia.

El problema de la palabra

A través de la historia, los calculos prácticos en teoría de grupos han sido llevados a cabo usando varias formas normales. Estas usualmente resolvían de manera implícita el siguiente problema: Dadas dos palabras en los generadores del grupo, ¿Representan estas palabras al mismo elemento? En 1911, Max Dehn [De11], afirmó que este era un problema muy importante en grupos, por lo que sería muy útil tener un método para poder resolverlo. A este problema algorítmico lo llamamos *el problema de la palabra*. Aunque es muy común hablar del “problema de la palabra para el grupo G ”, en realidad nos referimos al problema con respecto a una presentación de G . Formalmente, dada μ , una presentación de grupo, y dos palabras $v, w \in \Sigma^*(S)$ (donde S es el conjunto de generadores de μ), buscamos un algoritmo el cual nos de como salida “sí”, si $v = w$ en $G(\mu)$, y “no” en otro caso. El problema de la palabra no depende de la presentación de grupo dada, ya que si μ, ν son dos presentaciones isomorfas, entonces existe un algoritmo para resolver el problema de la palabra en μ si y sólo sí existe un algoritmo para resolver el problema de la palabra en ν . Se conocen algoritmos para resolver este problema computacional en varios casos particulares, como por ejemplo:

- Grupos finitos.
- El grupo $ISO(n)$ de isometrías de \mathbb{R}^n (también llamado grupo euclidiano).
- Grupos simples finitamente presentados.
- Grupos residualmente finitos⁴ y finitamente presentados.
- Grupos fundamentales de 2-variedades orientables cerradas.

⁴Un grupo G es residualmente finito (o finitamente aproximable) si para cada elemento $g \in G$ tal que $g \neq 1$, existe un homomorfismo $h: G \rightarrow H$, donde $|H| < \infty$ y $h(g) \neq 1$.

Y otros mas, sin embargo, este problema, aunque muy sencillo en su forma, es indecible en general.

Teorema 2.2.1 (P. S. Novikov [No55]). *Existe un grupo finitamente presentado, para el cual el problema de la palabra no es computable⁵.*

Así, este fue uno de los primeros ejemplos de un problema no computable, que no surgió en la lógica o en la teoría de la computación, sino en una de las ramas más antiguas e importantes de las matemáticas, el álgebra. Pero el problema de la palabra no es lo único no computable en grupos, hay muchos otros problemas y propiedades de grupos que resultaron ser también indecibles. Dada una presentación finita μ , los siguientes problemas relacionados a propiedades de grupos son indecibles:

- Dadas dos palabras $u, w \in F(\mu)$, decidir si u, w representan elementos conjugados en $G(\mu)$.
- Decidir si $G(\mu)$ es de orden 1.
- Decidir si $G(\mu)$ es finito.
- Decidir si $G(\mu)$ es un p -grupo.
- Decidir si $G(\mu)$ es simple.
- Decidir si $G(\mu)$ es de torsión (o libre de torsión).

Entre varios otros problemas, véase [Ro95, p. 469, Corolario 12.34] para mas detalles.

Los problemas de isomorfismo y trivialidad

Teniendo una forma de representar grupos en una computadora, y siendo capaz de formular algoritmos para que resuelvan cuestiones sobre ellos, no podía faltar alguien que se preguntara si será posible reconocer de forma algorítmica si dos grupos son isomorfos. Ya tenemos una propiedad de cadenas equivalente a esto, que dimos en la Definición 2.1.3, por lo que basta con ver

⁵El método con el cual Novikov obtuvo este resultado fue uno muy largo y complicado, en el cual utilizó un resultado de Alan Turing [Tu50], donde se demostraba que el problema de la palabra para semi-grupos con cancelación no es computable. Años después, Boone [Bo59] obtuvo una demostración más sencilla.

si existe un algoritmo para decidir esta propiedad. Si \mathcal{K} es una colección de presentaciones finitas de grupos, definimos el conjunto

$$\text{ISOMORFISMO}(\mathcal{K}) = \{(\mu, \nu) \in \mathcal{K} \times \mathcal{K} \mid \mu \cong \nu\},$$

como el problema del isomorfismo para \mathcal{K} . También es útil muchas veces, el poder decir si un grupo dado es el grupo trivial. De manera similar a lo anterior, definimos el conjunto

$$\text{TRIVIAL}(\mathcal{K}) = \{\mu \in \mathcal{K} \mid \mu \cong (\emptyset: \emptyset)\},$$

como el problema de trivialidad para \mathcal{K} . Disponer de algoritmos para decidir ambas propiedades sería algo por demás útil en muchas ramas de la matemática (y también fuera de ella), como es el caso de la Topología algebraica.

Desafortunadamente, la indecidibilidad asoma su rostro nuevamente; no es posible, en general, determinar si dos presentaciones dadas son isomorfas, o si alguna de ellas representa al grupo trivial. Pero esta vez hay más, es posible, dentro del ámbito de los conjuntos c.e., escoger que tan “insoluble” queremos que sea el problema. Los siguientes resultados fueron obtenidos por W. W. Boone [Bo68], y son las semillas de indecidibilidad que usaremos para probar la imposibilidad de la clasificación de las variedades en el Capítulo 5. Para poder enunciarlos, antes debemos dar una pequeña definición.

Definición 2.2.2. Sea \mathcal{K} un conjunto de presentaciones finitas de grupos. \mathcal{K} es (completamente) computable, si existe una MT total M , tal que, al recibir como entrada una presentación finita μ , M acepta a μ si existe $\nu \in \mathcal{K}$, tal que μ es (congruente) isomorfa a ν , y la rechaza en caso contrario.

Teorema 2.2.3 (Boone [Bo68]). *Para cada grado de Turing $\mathbf{e} \in \mathcal{E}$, existe un conjunto completamente computable $\mathcal{Q}(\mathbf{e})$, de presentaciones finitas de grupos, tal que,*

G1 $\text{ISOMORFISMO}(\mathcal{Q}(\mathbf{e})) \in \mathbf{e}$;

G2 $\text{TRIVIAL}(\mathcal{Q}(\mathbf{e})) \in \mathbf{e}$;

G3 $\mathcal{Q}(\mathbf{e})$ contiene a la presentación del grupo trivial $\mu_0 = (\emptyset: \emptyset)$;

G4 Para todas $\mu, \nu \in \mathcal{Q}(\mathbf{e}) - \{\mu_0\}$, tenemos que $r(\mu) = r(\nu)$ y $p(\mu) = p(\nu)$.

El Teorema 2.2.3 es esencialmente el Resultado 4 de Boone [Bo68]. Al agregar la presentación μ_0 a $\mathcal{Q}(\mathbf{e})$, no cambiamos el grado de insolubilidad de ninguno de los conjuntos considerados. También, todas las presentaciones de $\mathcal{Q}(\mathbf{e})$ tienen el mismo conjunto de generadores, por lo que “conjunto computable” implica “conjunto completamente computable”.

En vista de todos los fuertes resultados negativos de computabilidad en teoría de grupos, estaríamos tentados a decir que las presentaciones de grupos no tienen suficiente información de los grupos que representan, no podemos manipular de manera práctica muchas propiedades importantes de estos. Sin embargo, hasta hoy día, las presentaciones de grupos son lo mejor que se tiene para manejar grupos de manera algorítmica. Hay muchas satisfacciones que la matemática ha obtenido de la Teoría combinatoria de grupos. Incluso existen programas de computadora, como **MAGMA**⁶ y **GAP**⁷. El lector interesado en profundizar puede consultar [SE], como una introducción al tema.

2.3. El método de decisión de Tarski para el álgebra elemental de los reales

Como parte de los ingredientes algebraicos y computacionales que necesitaremos en los capítulos posteriores, vamos a enunciar un resultado muy famoso de Tarski [Ta67]. Para ello, necesitamos definir lo que es una *expresión elemental*⁸ en el conjunto de números reales.

Definición 2.3.1. Una expresión elemental en el álgebra (elemental) de los números reales, es una expresión compuesta de los siguientes objetos:

1. Variables sobre los números reales.
2. Constantes que denotan números naturales.
3. Los símbolos $+$, $-$, \cdot , \div , que expresan suma, resta, multiplicación y división de números reales.

⁶Disponible en <http://magma.maths.usyd.edu.au/>.

⁷Disponible en <http://www.gap-system.org/>.

⁸Para los expertos en lógica: Una expresión elemental es una fórmula de la lógica de predicados de primer orden con signatura $\{+, -, \cdot, \div\} \cup \mathbb{N} \cup \{>, =\}$.

4. Los símbolos $>$, $=$, que denotan a las relaciones de números reales “mayor que” e “igual”.
5. Los conectivos lógicos \vee (disyunción), \wedge (conjunción), \neg (negación) e \Rightarrow (implicación).
6. Los cuantificadores \forall (universal) y \exists (existencial) sobre variables.

Es importante notar que con expresiones elementales, no es posible, en general, hablar acerca de conjuntos de números reales. Aunque podemos hablar de algunos conjuntos finitos, como por ejemplo, el conjunto $\{5, 7, 10000\}$ que viene dado por la expresión elemental “ $x = 5 \vee x = 7 \vee x = 10000$ ”, pero no es posible hablar del conjunto de todos los enteros, es decir, no es posible escribir con los elementos de una expresión elemental el enunciado “ x es un entero”, por lo que la expresión “ $x \in \mathbb{Z}$ ” no es elemental. Pero todas las propiedades básicas de campo y de orden de \mathbb{R} pueden ser dadas con expresiones elementales, como por ejemplo

“Las operaciones binarias $+$ y \cdot del campo \mathbb{R} son conmutativas.”

$$\forall x, y (x + y = y + x \wedge x \cdot y = y \cdot x)$$

“Existe en el campo \mathbb{R} un elemento neutro para la operación $+$.”

$$\exists y \forall x (x + y = x)$$

“Para cada $x, y, z \in \mathbb{R}$, si $x \geq y$ entonces $x + z \geq y + z$.”

$$\forall x \forall y \forall z ((x > y \vee x = y) \Rightarrow (x + z > y + z \vee x + z = y + z))$$

Así como las propiedades de polinomios de grado n , con n fijo.

“El inverso en la suma del polinomio $x + 1$ es $-x - 1$.”

$$x + 1 - x - 1 = 0$$

“La suma de los polinomios $x^2 + 1$ y $x^4 + 3x^3 + 10x + 2$ no es igual a

$$x^2 + x + 1.”$$

$$\neg((x^2 + 1) + (x^4 + 3x^3 + 10x + 2) = x^2 + x + 1)$$

La razón por la que la expresión anterior es elemental, es que a cada monomio lo podemos ver como una constante (el coeficiente) multiplicada por variables sobre los reales (la indeterminada x), por ejemplo, $3x^3$ como expresión elemental es $3 \cdot x \cdot x \cdot x$. Estamos listos para enunciar el resultado de Tarski.

Teorema 2.3.2 (Tarski [Ta67]). *Existe un algoritmo para determinar, dada una expresión elemental en el álgebra elemental de los números reales, si esta es verdadera o falsa.*

Con este resultado, tenemos todos los ingredientes algebraicos que necesitamos, por lo que damos por concluido el capítulo.

Capítulo 3

Topología

Ahora pondremos nuestra atención en la topología, introduciendo todos los conceptos y resultados necesarios para poder proceder con nuestro objetivo en los siguientes capítulos. Ya que los resultados que deseamos probar tienen que ver con los complejos simpliciales y las variedades (diferenciables y/o combinatorias), la discusión estará ampliamente enfocada en estos objetos. Seguiremos una estructura bastante parecida a la del Capítulo 2. Primero, introducimos los conceptos (no muy conocidos) de variedades combinatorias y equivalencia combinatoria. Hablaremos también del pegado de asas y cirugía de Morse; y finalmente, discutiremos los resultados de computabilidad que se tienen en espacios, tanto propiedades computables, como no computables. Lo que asumimos del lector en este capítulo, es un conocimiento básico sobre topología general, topología algebraica (específicamente, sobre el grupo fundamental y los grupos de homología de un espacio), complejos simpliciales y topología diferencial. Nuestra notación es totalmente estándar. Las referencias recomendadas para estos temas son [HFFA03, GP75, Hi76, Ko93, Sp66, Ro88].

3.1. Topología combinatoria

Desde principios del siglo XX, la disciplina de la topología combinatoria ha desarrollado y hecho uso de técnicas combinatorias, aplicándolas en la topología (lo cual dio origen al surgimiento de la topología algebraica). Es precisamente dentro de esta disciplina donde surge los conceptos de complejo simplicial abstracto y el de variedades combinatorias, que tienen un papel

protagónico en la topología PL (Piecewise Linear, es decir, lineal por pedazos). Ya que serán de gran importancia en capítulos posteriores, daremos una breve introducción del tema, el lector puede encontrar muchos mas detalles de la topología PL en [Hu69, RS72].

3.1.1. Variedades combinatorias

Introducimos los conceptos de variedades combinatorias (también conocidas como variedades PL, como veremos mas adelante), y equivalencia combinatoria entre complejos.

Con $\text{Vert}(K)$ denotamos al conjunto de vértices del complejo simplicial K . Si $v \in \text{Vert}(K)$, la estrella cerrada de v en K es la unión de todos los simplejos que contienen a v y sus caras; este es un subcomplejo de K y lo llamamos $\text{St}_K(v)$. A la realización geométrica de K la denotamos por $|K|$. Una subdivisión de K es un complejo simplicial K^* que satisface: 1) $\text{Vert}(K^*) \subset |K|$; 2) para cada simplejo $s^* \in K^*$, existe un simplejo $s \in K$ tal que $s^* \subset |s|$; 3) la función inducida $\varphi: |K^*| \rightarrow |K|$ es un homeomorfismo. El n -simplejo estándar Δ_n es el complejo simplicial cuyos vértices son $\text{Vert}(\Delta_n) = \{e_0, e_1, \dots, e_n\}$ y los simplejos son todos los subconjuntos (no vacíos) de vértices. Como es usual, el interior y la frontera de una variedad M los denotamos por $\text{Int } M$ y ∂M .

Definición 3.1.1. Dos complejos simpliciales K, L son equivalentes combinatoriamente, si existen subdivisiones, K^*, L^* , de K y L respectivamente, que son isomorfas. Decimos que K es una n -bola combinatoria, si K es equivalente combinatoriamente al n -simplejo estándar Δ_n .

Cuando K y L son equivalentes combinatoriamente, existe pues, un isomorfismo simplicial, $j: K^* \rightarrow L^*$, por lo tanto, existe un homeomorfismo $f: |K| \rightarrow |L|$, con la propiedad de que $f(s^*) = j(s^*)$ para todo simplejo $s^* \in K^*$ (f lleva los simplejos de K^* linealmente en los simplejos de L^*); es decir, el siguiente diagrama

$$\begin{array}{ccc} |K^*| & \xrightarrow[\cong]{|j|} & |L^*| \\ \cong \downarrow & & \downarrow \cong \\ |K| & \xrightarrow[\cong]{f} & |L| \end{array}$$

conmuta. Llamamos a f un homeomorfismo PL (o también homeomorfismo semilineal); éstos son la equivalencia principal de la topología PL, y las pro-

propiedades que son preservadas por ellos son llamadas invariantes PL. Ahora podemos dar los conceptos de variedad PL y variedad combinatoria.

Definición 3.1.2. Una n -variedad PL es un espacio metrizable M con un atlas PL máximo, donde un atlas PL es un atlas Φ para M como n -variedad topológica, y es tal que los cambios de coordenadas son homeomorfismos PL.

Definición 3.1.3. Una n -variedad combinatoria es un espacio M que tiene una triangulación K , tal que para cada vértice $p \in \text{Vert}(K)$ la estrella cerrada $\text{St}_K(p)$ es equivalente combinatoriamente al n -simplejo estándar Δ_n (es decir, $\text{St}_K(p)$ es una n -bola combinatoria).

Aunque las dos definiciones anteriores son muy diferentes superficialmente, en realidad son equivalentes [Hu69].

Teorema 3.1.4. *Sea M un espacio. Entonces M es una variedad PL si y sólo si M es una variedad combinatoria.*

Por lo que con este resultado, estamos justificados para usar los dos conceptos en forma indistinta. En particular, estaremos usando a lo largo de la obra la definición de variedad combinatoria, pues se acomoda mejor a nuestros propósitos. Cuando M es una variedad combinatoria (PL), a una triangulación K que cumple con las condiciones de la Definición 3.1.3, le llamamos una *estructura combinatoria en M* . Básicamente, todas las variedades conocidas ($\mathbb{R}^n, \mathbb{S}^n, D^n, \mathbb{R}P^n$, etc.) son ejemplos de variedades PL.

Uno de los problemas más famosos de la topología es la llamada *Conjetura de Poincaré*¹, que dice que toda 3-variedad simplemente conexa (triangulada) es homeomorfa a la esfera \mathbb{S}^3 . La *Conjetura generalizada de Poincaré*, dice que toda n -variedad cerrada, que es del mismo tipo de homotopía que \mathbb{S}^n , es homeomorfa a \mathbb{S}^n . En el caso de las variedades PL, este problema ha sido resuelto en forma positiva, para $n \geq 5$.

Teorema 3.1.5. *Sea M^n una variedad PL cerrada, que es del mismo tipo de homotopía que \mathbb{S}^n , $n \geq 5$. Entonces M^n es homeomorfa a \mathbb{S}^n .*

¹Después de casi un siglo de esfuerzos, esta conjetura ha sido resuelta, por Grigori Perelman (siguiendo el programa de Richard Hamilton). La Conjetura de Poincaré, era uno de los siete problemas del Milenio, por los que el *Clay Mathematics Institute*, ofrece un premio de \$ 1,000,000 USD. Otro de estos problemas, es el principal problema abierto de la computación, que mencionamos en un pie de página del Capítulo 1: $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. vease <http://www.claymath.org/millennium/>.

Este teorema fue probado por Smale [Sm61], usando técnicas de pegado de asas (de lo cual hablaremos un poco más adelante), para variedades PL y para variedades diferenciables. Para $n \geq 6$, el Teorema 3.1.5² puede ser deducido del teorema de h -cobordismo. Un cobordismo (W^m, M_0, M_1) consiste de una variedad compacta W , con ∂W difeomorfa a la unión disjunta de M_0 y M_1 . (W, M_0, M_1) es un h -cobordismo, si las inclusiones, $i_q: M_q \hookrightarrow W$ ($q = 0, 1$), son equivalencias homotópicas.

Teorema 3.1.6. *Sea (W^m, M_0, M_1) un h -cobordismo, con W una variedad simplemente conexa, y $m \geq 6$. Entonces W es homeomorfa a $M_0 \times [0, 1]$. Por lo tanto, M_0 es homeomorfa a M_1 .*

3.1.2. La conjetura Hauptvermutung

Otra conjetura muy conocida en topología, es la llamada *Hauptvermutung*, que es la abreviación para *die Hauptvermutung der kombinatorischen Topologie*, lo cual es el alemán para el enunciado “la conjetura principal de la topología combinatoria”. Esta afirma que dos complejos simpliciales K, L , son equivalentes combinatoriamente, si éstos son homeomorfos (es decir, tienen esencialmente una única triangulación). La converso es siempre cierta.

Es sabido que esta conjetura es falsa en general, lo cual fue probado por John Milnor en 1961 [Mi61]. En los años sesentas, hubo mucha actividad intentando probar o refutar la versión restringida a las variedades PL (tratando de imitar la prueba de Milnor), primero probando que es cierta en casos especiales, y luego verificando que es falsa en general. En variedades de dimensión ≤ 3 , la conjetura es cierta, pero en dimensión ≥ 4 , resultó ser falsa. Todos estos hechos pueden ser consultados en [RCSARC96].

3.1.3. Variedades diferenciables y/o combinatorias

A las variedades con las que trabajaremos de aquí en adelante, les pediremos que sean variedades PL, o diferenciables, o ambas cosas, y en el último caso, les pediremos que cumplan una condición adicional: que la estructura combinatoria sea compatible con la estructura diferenciable. Esto significa lo siguiente: Sea M^n una C^r - n -variedad ($r \in \mathbb{N} \cup \{\infty\}$). Decimos que una

²J. Stallings obtuvo una prueba de este teorema para $n \geq 7$, usando diferentes métodos [Sta60].

estructura combinatoria K en M es compatible con la C^r -estructura de M , si cada q -simplejo $s \in K$ es un C^r -simplejo, es decir, existe un sistema de coordenadas $h: V \rightarrow \mathbb{R}^n$ en M , tal que $|s| \subseteq V$ y $h(|s|)$ es un q -simplejo rectilíneo en \mathbb{R}^n . Recordemos algunos resultados importantes acerca de las variedades diferenciables y combinatorias:

- i) Toda variedad con una estructura de clase C^1 , tiene una estructura de clase C^∞ y, más aún, tiene una estructura de clase C^ω (analítica) [Wh36].
- ii) Toda variedad diferenciable admite una estructura combinatoria compatible [Ca35, Ca61-2, Whi40].
- iii) Existen variedades PL, que no admiten una estructura diferenciable [Ke60].
- iv) Para $n \geq 7$, la esfera \mathbb{S}^n admite varias estructuras diferenciables (no difeomorfos) [Mi56].

Notación. Para evitar escribir de manera repetitiva “ M una variedad diferenciable y/o combinatoria”, abreviaremos esta frase con “ M una variedad C^∞ -PL” (y lo mismo con una función diferenciable y/o PL). Por supuesto, cuando M sea diferenciable y combinatoria, asumimos que la estructura combinatoria es compatible con la estructura diferenciable.

3.2. Asas y cirugía de Morse

En esta sección, recordamos los conceptos básicos sobre el pegado de asas y la cirugía de Morse. Ambos han sido muy importantes en la obtención de muchos resultados trascendentes en lo que respecta a variedades. Nosotros sólo necesitaremos un manejo elemental de ellos. Daremos las definiciones principales y también mencionaremos algunos resultados, pero antes de ello, debemos mencionar un hecho importante que usaremos de forma implícita. Si tenemos dos variedades diferenciables M, N , donde ambas tienen frontera, no podemos dar una estructura diferenciable al producto $M \times N$ en la forma en que lo haríamos si M y N no tuvieran frontera, ya que en $M \times N$ se presentan esquinas, lo cual nos da problemas al hablar de la derivabilidad. Afortunadamente, Milnor solucionó este problema al desarrollar el método del “Enderezamiento del ángulo”. De este modo, el producto de variedades diferenciables

con frontera tiene una estructura diferenciable única salvo difeomorfismo. La descripción de este método puede ser encontrada en [DH, Ma92].

3.2.1. Descomposición en asas

Un asa es un disco, el cual puede ser adherido (en una parte o en toda su frontera) a una variedad. Formalmente, un λ -asa de dimensión n es un disco, $D^n \subseteq \mathbb{R}^n$, visto como un producto, $D^n = D^\lambda \times D^{n-\lambda}$, de los discos de dimensión λ y $n - \lambda$, respectivamente. Sea M^n una variedad compacta con frontera. Consideremos a una $(n - 1)$ -subvariedad, $N^{n-1} \subseteq \partial M$, y tomemos p asas de dimensión n ,

$$D_1^n, \dots, D_p^n.$$

Supongamos que tenemos p encajes³ C^∞ -PL

$$\varphi_i: \mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda} \subseteq \partial D_i^n \rightarrow \text{Int } N, \quad 1 \leq i \leq p,$$

tales que, para todas $1 \leq l \neq k \leq p$, se cumpla que $\text{Im } \varphi_l \cap \text{Im } \varphi_k = \emptyset$. Formamos el espacio cociente

$$(M^n \cup D_1^n \cup \dots \cup D_p^n) / \sim,$$

donde $x \sim y$ si y sólo si $x \in \mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda}$ y $\varphi_i(x) = y \in N \subseteq \partial M$. El espacio topológico obtenido es una n -variedad, ya que tiene una estructura C^∞ -PL natural (gracias al método del “Enderezamiento del ángulo” de Milnor, véase [Sm61]; para nuestros propósitos, sólo necesitaremos los casos $\lambda = 1$ o 2 , con $n \geq 5$). Denotamos a esta n -variedad por

$$\chi(M^n, N^{n-1}; \varphi_1, \dots, \varphi_p, \lambda).$$

Y decimos que este es el resultado de agregar p asas de índice λ a la variedad M^n , sobre la subvariedad N^{n-1} .

Ejemplo. Veamos que variedad podemos obtener al pegar al 2-disco D^2 una 1-asa de dimensión 2 sobre la subvariedad $N = \{0\} \times I \cup \{1\} \times I \subseteq \partial D^2$. Si $\varphi: \mathbb{S}^0 \times D^1 \rightarrow \text{Int } N$ es el encaje para el pegado de la asa y es tal que envía las dos componentes de $\mathbb{S}^0 \times D^1$ con la misma orientación sobre las componentes de N , entonces $\chi(D^2, N; \varphi, 1)$ es $\mathbb{S}^1 \times I$. Si por otro lado, φ lleva a cada componente de $\mathbb{S}^0 \times D^1$ con orientaciones opuestas sobre las componentes de N , entonces $\chi(D^2, N; \varphi, 1)$ es la banda de Móbius.

³En el caso de que φ_i sea PL, le pedimos que lo sea sobre la estructura combinatoria de D^n y la de M^n .

Uno de los resultados más importantes sobre el pegado de asas, es el que dice que cualquier n -variedad M C^∞ -PL, puede ser obtenida a partir del n -disco, pegando sucesivamente asas de índices $\lambda = 1, \dots, n$ [Sm61]. A la lista de los encajes correspondientes, φ_i , le llamamos una *presentación por asas de M* . Por lo tanto, existe una descomposición en asas de M , de la forma

$$D^n = M_0 \subset M_1 \subset \dots \subset M_n = M,$$

donde cada M_i es obtenida de M_{i-1} al pegarle un n_i -asa. Las descomposiciones en asas no son únicas. El siguiente, es la versión diferenciable del Teorema 3.1.5.

Teorema 3.2.1 (Smale, [Sm61]). *Sea M^n una variedad diferenciable cerrada, que es del mismo tipo de homotopía que \mathbb{S}^n , $n \geq 5$. Entonces M^n es homeomorfa a \mathbb{S}^n .*

3.2.2. Cirugía de Morse

Veamos ahora la definición de cirugía de Morse [Mi61-2]. Sea N^{n-1} una $(n-1)$ -variedad compacta. Consideremos p copias de la $(n-1)$ -esfera $\mathbb{S}_1^{n-1}, \dots, \mathbb{S}_p^{n-1}$. A cada una la podemos ver de la siguiente forma,

$$\mathbb{S}_i^{n-1} = (\mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda}) \cup (D_i^\lambda \times \mathbb{S}_i^{n-\lambda-1}),$$

donde

$$\partial(\mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda}) = \partial(D_i^\lambda \times \mathbb{S}_i^{n-\lambda-1}).$$

Tomemos, nuevamente, p encajes³ C^∞ -PL

$$\varphi_i: \mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda} \rightarrow \text{Int } N, \quad 1 \leq i \leq p,$$

tales que $\text{Im } \varphi_l \cap \text{Im } \varphi_k \neq \emptyset$ ($l \neq k$). Sea C la cerradura del subespacio $N - \bigcup_{i=1}^p \varphi_i(\mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda})$. Tenemos que

$$\partial C = \partial N \cup \bigcup_{i=1}^p \varphi_i(\mathbb{S}_i^{\lambda-1} \times \mathbb{S}_i^{n-\lambda-1}),$$

donde

$$\partial(\mathbb{S}_i^{\lambda-1} \times D_i^{n-\lambda}) = \mathbb{S}_i^{\lambda-1} \times \mathbb{S}_i^{n-\lambda-1}.$$

Finalmente, formamos el espacio cociente

$$(C \cup \bigcup_{i=1}^p D_i^\lambda \times \mathbb{S}_i^{n-\lambda-1}) / \sim,$$

donde \sim identifica a $x \in \mathbb{S}_i^{\lambda-1} \times \mathbb{S}_i^{n-\lambda-1}$ con $\varphi_i(x) \in \partial C$. El espacio resultante es una n -variedad, que denotamos por

$$v(N^{n-1}; \varphi_1, \dots, \varphi_p, \lambda).$$

A esto le llamamos el resultado de p cirugías de Morse de índice λ , aplicadas a M .

Ambos conceptos, el pegado de asas y la cirugía de Morse aparecieron de manera independiente, sin aparente relación; sin embargo, podemos notar que el agregar asas a una variedad, induce cirugía de Morse en la subvariedad en donde se hace el pegado de las asas. Más aún, si identificamos las esferas \mathbb{S}_i^{n-1} de la definición de cirugía de Morse con las fronteras de los discos, ∂D_i^n , en la definición del pegado de asas, y además consideramos iguales a $N, D_i^n, D_i^{n-\lambda}, \mathbb{S}_i^{\lambda-1}, \mathbb{S}_i^{n-\lambda-1}$ y φ_i , entonces

$$v(N^{n-1}; \varphi_1, \dots, \varphi_p, \lambda) \subseteq \partial\chi(M^n, N^{n-1}; \varphi_1, \dots, \varphi_p, \lambda);$$

y en caso de que $N = \partial M$, entonces $v = \partial\chi$, a la cual le decimos la “ $(n-1)$ -variedad construida en ∂M , por la cirugía de Morse inducida por el pegado de asas a M ”.

3.3. Computabilidad en espacios

Pasemos ahora al tema de computabilidad en espacios topológicos. Veremos que aunque ha habido grandes logros en lo que a cálculos se refiere, aún falta mucho por hacer. Mencionaremos solo lo que es más relevante para nuestro objetivo final.

Codificación de espacios

La clase de espacios más agraciados, en cuanto a tener una forma finita de representarlos en una computadora, han sido sin lugar a dudas, los complejos simpliciales finitos, ya que su esquema puede ser especificado como

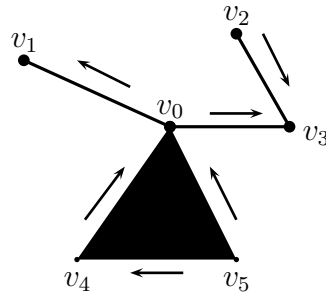


Figura 3.1: Complejo simplicial orientado.

una palabra de símbolos. Si $\Sigma = \{v_1, \dots, v_n, (,)\}$ es un alfabeto, entonces el complejo abstracto dado en la figura 3.1, está totalmente especificado por la palabra

$$(v_0)(v_1)(v_2)(v_3)(v_4)(v_5)(v_0v_1)(v_0v_3)(v_3v_2)(v_0v_5)(v_0v_4)(v_4v_5)(v_0v_5v_4), \quad (3.1)$$

que codifica en primer lugar a los 0-simplejos (vértices) dados por las subpalabras de la forma (v_i) , seguidos de los 1-simplejos (v_iv_j) (aristas) y finalmente un 2-simplejo $(v_0v_5v_4)$ (el triángulo). Más aún, podemos construir una MT total M tal que, en la entrada $w \in \Sigma^*$, M acepta a w si y sólo si w representa a un complejo simplicial abstracto. Esta representación de complejos es una muy satisfactoria, con la que se pueden realizar muchos cálculos.

Otra forma de representar complejos finitos es utilizando matrices de incidencia. Sea K un n -complejo; entonces a K lo podemos representar por medio de una colección de matrices $\{E^q\}_{q=0}^{n-1}$, donde la matriz E^q codifica la relación de incidencia de todos los q -simplejos de K con todos los $(q+1)$ -simplejos; la entrada e_{ij}^q de E^q es igual a 1 (-1), si el q -simplejo orientado $s_i \in K$ es adyacente⁴ al $(q+1)$ -simplejo $s_j \in K$, y (no) presenta la orientación inducida por este último; e_{ij}^q es igual a 0 si s_i no es adyacente a s_j . Por ejemplo, las matrices de incidencia del complejo de la Figura 3.1 son (omitimos los parentesis, para mayor claridad)

⁴En un complejo K , dos simplejos s_i, s_j son *adyacentes* si tienen una cara en común.

E^0	v_0v_1	v_0v_3	v_3v_2	v_0v_5	v_0v_4	v_4v_5
v_0	-1	-1	0	1	1	0
v_1	1	0	0	0	0	0
v_2	0	0	-1	0	0	0
v_3	0	1	1	0	0	0
v_4	0	0	0	0	-1	1
v_5	0	0	0	-1	0	-1

E^1	$v_0v_4v_5$
v_0v_1	0
v_0v_3	0
v_3v_2	0
v_0v_5	-1
v_0v_4	1
v_4v_5	1

Con esta representación de complejos, también se pueden hacer varios cálculos importantes (que por supuesto, no dependen de la orientación escogida), pero estos pueden ser muy largos, ya que las matrices de incidencia tienden a crecer mucho, con respecto al tamaño del complejo. Para resolver esta cuestión, se inventaron las matrices de bloques, las cuales se obtienen al considerar un número finito de cadenas simpliciales, llamadas cadenas de bloques, que cumplen ciertos requerimientos. De estas cadenas, se puede obtener un conjunto de matrices semejantes a las de incidencia, pero con la ventaja de que son mucho más pequeñas, así que los cálculos se vuelven más fáciles y prácticos⁵. Todos los detalles de estas representaciones de complejos se pueden encontrar en [ST51, p. 90-103] y [Ro88, p. 155-156].

Pasemos al caso de las variedades. Podemos decir que a esta clase de espacios no les ha ido tan bien como a los complejos. Sabemos que toda variedad diferenciable es triangulable, lo cual nos dice que, existen variedades que tiene un esquema de complejo simplicial asociado, pero con esta codificación, solo tenemos información acerca de la estructura de la variedad como complejo,

⁵Como solo estamos tratando cuestiones de decidibilidad, no nos importa cuánto tiempo se tarden los algoritmos que formulemos, mientras tomen una cantidad finita de tiempo, es suficiente para nuestros fines. Por supuesto, en la práctica, un algoritmo que tarda un tiempo de 2^n , donde n es el “tamaño” de la entrada, no es útil, ya que con una entrada de tamaño $n = 100$, tardaría miles de años en darnos la respuesta.

no nos dice nada de la estructura de variedad. Si tenemos una variedad PL, estaríamos tentados a pensar que en este caso, la situación es mas favorable, por la estructura combinatoria, pero no es así; un esquema de complejo simplicial no es suficiente para codificar la información importante acerca de una variedad PL (Veremos mas adelante el porqué de esto). ¿Qué hay de la presentación por asas, nos podría servir ? No, ya que necesitaríamos representar a las funciones φ_i en forma finita, lo cual en general, no parece posible.

Pero a pesar de este panorama poco alentador, sí existe una forma de representar en una computadora, una clase bastante amplia de variedades (C^∞ -PL) con toda la información importante sobre la estructura diferenciable y/o combinatoria. Esto lo discutiremos en el Capítulo 4, donde probaremos que *toda variedad C^∞ -PL cerrada, tiene una presentación finita de variedad*; por el momento, supondremos la existencia de dichas presentaciones, para poder introducir los problemas computacionales (originados a partir de propiedades de variedades) que enfrentaremos en el Capítulo 5. A estas presentaciones las denotaremos con las letras \mathfrak{M} , \mathfrak{N} , \mathfrak{S} , etc.

Ahora vamos a discutir la computabilidad de varias propiedades e invariantes en los espacios.

Los grupos de homología y números de Betti

Los grupos de homología de un espacio son uno de los invariantes topológicos mas importantes que se conocen, ya que, aunque no tienen toda la información que logran guardar los grupos de homotopía, son mucho más fáciles de calcular que estos últimos. Afortunadamente, en el caso de los complejos simpliciales, la teoría de homología singular (con coeficientes en \mathbb{Z}) es equivalente a la teoría de homología simplicial, y esto abre la puerta a métodos computacionales para poder calcular de forma mecánica los grupos de homología de un complejo simplicial finito. La discusión de estos temas está basada en [Mu00].

Para un complejo simplicial K , se sabe que el n -ésimo grupo de homología (simplicial) $H_n(K)$, es un grupo abeliano finitamente generado (f.g.). Por el Teorema de clasificación de estos grupos, tenemos que

$$H_n(K) \cong \underbrace{\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}}_{b_n} \oplus \mathbb{Z}_{d_{n,1}} \oplus \cdots \oplus \mathbb{Z}_{d_{n,l}},$$

donde $d_{ni-1} \mid d_{ni}$, para toda $i = 2, \dots, l$. El número de sumandos directos de \mathbb{Z} , b_n , y los d_{ni} 's, son invariantes numéricos del complejo K , son conocidos como el n -ésimo número de Betti⁶ y los *coeficientes de torsión de la dimensión n de K* , respectivamente. Estos, determinan por completo a $H_n(K)$, por lo que si hubiese una forma de calcularlos, entonces podríamos calcular los grupos de homología de cualquier complejo finito. Y en efecto la hay, pero para poder entender como es posible hacer este cálculo, primero debemos ver algunos conceptos y resultados preliminares.

Definición 3.3.1. Sean G, G' grupos abelianos libres con bases a_1, \dots, a_n y b_1, \dots, b_m respectivamente. Si $f: G \rightarrow G'$ es un homomorfismo, entonces tenemos que

$$f(a_j) = \sum_{i=1}^m \lambda_{ij} b_i$$

para algunos enteros únicos λ_{ij} . A la matriz (λ_{ij}) se le conoce como la matriz de f con respecto a las bases dadas para G y G' .

Si $A \in M_{n \times m}(\mathbb{Z})$, consideremos las siguientes “operaciones elementales de renglones” en A :

- (1) Intercambiar los renglones i y k .
- (2) Multiplicar al renglón i por -1 .
- (3) Reemplazar al renglón i por (renglón i) + q (renglón k), con $q \in \mathbb{Z}$ e $i \neq k$.

Cuando A es la matriz de un homomorfismo de grupos abelianos libres, como en la Definición 3.3.1, cada una de estas operaciones corresponden a un cambio de base en G' . La primera corresponde a intercambiar a b_i y b_k ; la segunda corresponde al reemplazo de b_i por $-b_i$; y la tercera corresponde al reemplazo de b_k por $b_k - qb_i$. Existen tres “operaciones elementales de columnas”, totalmente similares a estas que hemos definido sobre renglones, y que corresponden a cambios de base de G .

⁶En general, el n -ésimo número de Betti de un espacio X , se puede definir como la dimensión del espacio vectorial $H_n(X, \mathbb{Q})$. El término “números de Betti” fue usado por primera vez por Henri Poincaré, en honor de Enrico Betti.

¿Qué tienen que ver estas matrices sobre los enteros con los grupos de homología de un complejo simplicial finito K ? Mucho, estas matrices son la herramienta básica para poder encontrar formas de calcular el número de Betti y los coeficientes de torsión de $H_n(K)$. Nuestro siguiente paso es definir una “forma normal” para matrices enteras.

Definición 3.3.2. Sea $B \in M_{n \times m}(\mathbb{Z})$. Decimos que B esta en forma normal de Smith (FNS), si tenemos que

$$B = \begin{pmatrix} b_1 & 0 & 0 & \cdots & 0 \\ 0 & b_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & & 0 \\ \vdots & & & b_l & 0 \\ & & & & 0 \\ & & & & \ddots \\ 0 & \cdots & & & 0 \end{pmatrix},$$

donde $b_i \geq 1$ y $b_1 \mid b_2 \mid \cdots \mid b_l$.

Existe un algoritmo por medio del cual, es posible llevar cualquier matriz A sobre los enteros a una matriz en FNS. El procedimiento por medio del cual se hace esto utiliza las operaciones elementales por renglones y columnas que hemos definido y es muy parecido al algoritmo de Gauss para diagonalizar matrices sobre los reales o los complejos. Para nuestros propósitos, no necesitamos ver los detalles del algoritmo, nos basta con saber que existe⁷. Gracias a estos hechos, si G, G' y f son como en la Definición 3.3.1, entonces la matriz (λ_{ij}) del homomorfismo f puede ser llevada a una matriz B en FNS, y como las operaciones elementales de renglones (columnas) corresponden a cambios de base en G' (G), entonces B es la matriz de f con respecto a ciertas bases de G y G' .

Para poder encontrar algoritmos para calcular $H_n(K)$, nos vamos a fijar en las matrices de los operadores frontera $\partial_n: C_n(K) \rightarrow C_{n-1}(K)$ con respecto a las bases de los grupos $C_n(K)$, que obtenemos al orientar todos los n -simplejos de K y considerarlos como n -cadenas elementales. De hecho, las matrices de los operadores frontera con respecto a estas bases coinciden con

⁷El lector puede consultar los detalles del algoritmo para llegar a la FNS en [Mu00, p. 56-57] y también en http://en.wikipedia.org/wiki/Smith_normal_form.

las matrices de incidencia de K que mencionamos al inicio de esta sección. Así, las matrices E^0, E^1 que representan al complejo simplicial de la Figura 3.1 son precisamente las matrices de los operadores ∂_1, ∂_2 , con respecto a las bases inducidas por la orientación escogida en el complejo. Ya casi estamos listos para enunciar el teorema que nos permitirá calcular grupos de homología de forma mecánica, sólo nos falta dar un resultado sobre complejos de cadena que necesitaremos.

Teorema 3.3.3. *Sea $\mathcal{C} = \{C_*, \partial_*\}$ un complejo de cadenas; supongamos que cada grupo C_p es libre y de rango finito. Entonces, para cada p , existen subgrupos U_p, V_p y W_p de C_p tales que*

$$C_p = U_p \oplus V_p \oplus W_p,$$

donde $\partial_p(U_p) \subseteq W_{p-1}$, $\partial_p(V_p) = 0$ y $\partial_p(W_p) = 0$. Más aún, existen bases para U_p y W_{p-1} con respecto a las cuales el homomorfismo $\partial_p: U_p \rightarrow W_{p-1}$ tiene una matriz de la forma

$$B = \begin{pmatrix} b_1 & & 0 \\ & \ddots & \\ 0 & & b_l \end{pmatrix},$$

donde $b_i \geq 1$ y $b_1 \mid b_2 \mid \cdots \mid b_l$.

Demostración. *Paso 1.* Sean

$$Z_p = \ker \partial_p \quad \text{y} \quad B_p = \text{Im } \partial_{p+1}.$$

Y sea $W_p = \{c_p \in C_p \mid \exists m \in \mathbb{Z}(m \neq 0 \wedge mc_p \in B_p)\}$. W_p es un subgrupo de C_p , es llamado el grupo de fronteras débiles. Claramente

$$B_p \subset W_p \subset Z_p \subset C_p.$$

(En la segunda inclusión usamos el hecho de que C_p es libre de torsión, por lo que la ecuación $mc_p = \partial_{p+1}d_{p+1}$ implica que $\partial_p c_p = 0$). mostraremos que W_p es un sumando directo de Z_p . Consideremos la proyección natural

$$Z_p \rightarrow H_p(\mathcal{C}) \rightarrow H_p(\mathcal{C})/T_p(\mathcal{C}),$$

donde $T_p(\mathcal{C})$ es el subgrupo de torsión de $H_p(\mathcal{C})$. El núcleo de esta proyección es W_p ; por lo tanto, $Z_p/W_p \cong H_p(\mathcal{C})/T_p(\mathcal{C})$. Este grupo es f.g. y libre

de torsión, es decir, es un grupo libre. Si $c_1 + W_p, \dots, c_k + W_p$ es una base para Z_p/W_p , y d_1, \dots, d_l es una base para W_p , entonces es sencillo ver que $c_1, \dots, c_k, d_1, \dots, d_l$ es una base para Z_p . Por lo tanto $Z_p = V_p \oplus W_p$, donde V_p es el grupo generado por la base c_1, \dots, c_k .

Paso 2. Supongamos que escogemos bases e_1, \dots, e_n y e'_1, \dots, e'_m para C_p y C_{p-1} respectivamente, con respecto a las cuales la matriz de $\partial_p: C_p \rightarrow C_{p-1}$ tiene la FNS

$$\begin{array}{c} e'_1 \\ e'_2 \\ \vdots \\ e'_l \\ e'_{l+1} \\ \vdots \\ e'_m \end{array} \begin{pmatrix} e_1 & e_2 & \cdots & e_l & e_{l+1} & \cdots & e_n \\ b_1 & 0 & 0 & & \cdots & & 0 \\ 0 & b_2 & 0 & & \cdots & & 0 \\ & 0 & 0 & \ddots & & & 0 \\ \vdots & & & & & & \\ & \vdots & & & b_l & & 0 \\ & & & & & 0 & \\ & & & & & & \ddots \\ 0 & & & \cdots & & & 0 \end{pmatrix}$$

donde $b_i \geq 1$ y $b_1 \mid b_2 \mid \cdots \mid b_l$. Afirmamos que:

- (1) e_{l+1}, \dots, e_n es una base de Z_p .
- (2) e'_1, \dots, e'_l es una base de W_{p-1} .
- (3) $b_1 e'_1, \dots, b_l e'_l$ es una base de B_{p-1} .

Probemos que en efecto, estos hechos son ciertos. Sea c_p una p -cadena y calculemos su frontera; si

$$c_p = \sum_{i=1}^n a_i e_i, \quad \text{entonces} \quad \partial_p c_p = \sum_{i=1}^l a_i b_i e'_i.$$

Para probar (1), notemos que como $b_i \neq 0$, la p -cadena c_p es un ciclo si y sólo si $a_i = 0$ para $i = 1, \dots, l$. Para (3), vemos que cualquier $p-1$ frontera $\partial_p c_p$ está en el grupo generado por $b_1 e'_1, \dots, b_l e'_l$; ya que $b_i \neq 0$, estos elementos son independientes. Finalmente, probamos (2) de la siguiente forma: Cada uno de los elementos del conjunto $\{e_1, \dots, e_l\}$ pertenece al grupo W_{p-1} , pues $b_i e'_i = \partial_p e_i$. Ahora sea

$$c_{p-1} = \sum_{i=1}^m d_i e'_i$$

una $(p-1)$ -cadena y supongamos que $c_{p-1} \in W_{p-1}$. Entonces c_{p-1} satisface una ecuación de la forma

$$\alpha c_{p-1} = \partial_p c_p = \sum_{i=1}^l a_i b_i e'_i$$

para algún $\alpha \neq 0$. Igualando los coeficientes, vemos que $\alpha d_i = 0$ para $i > l$, lo que nos dice que $d_i = 0$ para $i > l$. Concluimos que e'_1, \dots, e'_l es una base de W_{p-1} .

Paso 3. Con lo hecho en los pasos 1 y 2, finalmente podemos probar el teorema. Escojamos bases para C_p y C_{p-1} de la misma forma que lo hicimos en el Paso 2. Definamos a U_p como el grupo generado por e_1, \dots, e_l ; entonces

$$C_p = U_p \oplus Z_p.$$

Usando el Paso 1, escogemos a V_p tal que $Z_p = V_p \oplus W_p$. Entonces los grupos U_p, V_p y W_p nos dan una descomposición de C_p tal que $\partial_p(V_p) = 0$ y $\partial_p(W_p) = 0$. La existencia de las bases para U_p y W_{p-1} que dan origen a la matriz B del homomorfismo $\partial_p: U_p \rightarrow W_{p-1}$ se sigue del Paso 2. \square

Notemos que W_p y $Z_p = V_p \oplus W_p$ son subgrupos de C_p que están determinados de manera única, mientras que los subgrupos U_p y V_p no lo están.

Teorema 3.3.4. *Sea K un complejo simplicial finito. Entonces los grupos de homología $H_p(K)$ son computables.*

Demostración. Sea K un complejo finito. Por el Teorema 3.3.3, tenemos una descomposición

$$C_p(K) = U_p \oplus V_p \oplus W_p,$$

donde $Z_p = V_p \oplus W_p$ es el grupo de p -ciclos y W_p es el grupo de p -fronteras debiles. Ahora

$$H_p(K) = Z_p/B_p \cong V_p \oplus (W_p/B_p) \cong (Z_p/W_p) \oplus (W_p/B_p).$$

El grupo Z_p/W_p es libre y el grupo W_p/B_p es un grupo de torsión; por lo que calcular $H_p(K)$ se reduce a calcular estos dos grupos.

A continuación, escojamos bases para los grupos de cadenas $C_p(K)$, orientando a todos los simplejos de K . Luego, consideremos la matriz A^p del operador frontera $\partial_p: C_p(K) \rightarrow C_{p-1}(K)$ con respecto a esta elección de bases (esta es la matriz de incidencia de los $(p-1)$ -simplejos con los p -simplejos de K); entonces obtenemos una matriz N^p en FNS a partir de la matriz A^p . Examinando el paso 2 de la prueba del Teorema 3.3.3, podemos concluir los siguientes hechos sobre la matriz N^p :

- (1) El rango de Z_p es igual al número de columnas de ceros de N^p .
- (2) El rango de W_{p-1} es igual al número de renglones no cero de N^p .
- (3) Si b_1, \dots, b_l son las entradas distintas de cero de N^p , entonces existe un isomorfismo

$$W_{p-1}/B_{p-1} \cong \mathbb{Z}_{b_1} \oplus \mathbb{Z}_{b_2} \oplus \dots \oplus \mathbb{Z}_{b_l}.$$

Por lo tanto, la matriz N^p que es la FNS de la matriz A^p del operador ∂_p nos da los coeficientes de torsión de K en dimensión $p-1$; son precisamente las entradas de la matriz mayores a 1. De esta matriz también podemos deducir el rango de Z_p . Por otro lado, la matriz en FNS N^{p+1} de $\partial_{p+1}: C_{p+1}(K) \rightarrow C_p(K)$ nos da el rango de W_p . La diferencia de estos dos últimos números es el rango del grupo Z_p/W_p es decir, el número de Betti de K en dimensión p .

Ya solo nos falta dar el algoritmo preciso con el cual, a partir del complejo K , calcular el p -ésimo grupo de homología $H_p(K)$. Por lo que hemos dicho anteriormente, podemos representar complejos simpliciales finitos con palabras como la que mostramos en (3.1), y por la forma de los grupos abelianos f.g., que están totalmente determinados por el número de Betti y los coeficientes de torsión, que son números enteros, los podemos representar con palabras de la forma

$$b\#c_1\#c_2\#\dots\#c_n$$

donde b representa al número de Betti y c_1, c_2, \dots, c_n son los coeficientes de torsión. El símbolo $\#$ nos sirve para separar a las palabras que representan los enteros. Con todo esto, escogiendo un alfabeto adecuado Σ , podemos formar los siguientes conjuntos de palabras

$$\mathcal{CS} = \{K \in \Sigma^* \mid K \text{ es un complejo simplicial finito}\},$$

$$\mathcal{AB}_{fg} = \{G \in \Sigma^* \mid G \text{ es un grupo abeliano f.g.}\}^8.$$

Definamos $h_p: \Sigma^* \rightarrow \Sigma^*$, una función de palabras dada por

$$h_p(K) = \begin{cases} H_p(K) & \text{si } K \in \mathcal{CS}, \\ \lambda & \text{en otro caso.} \end{cases}$$

Es decir, h_p es la función de palabras que (bajo la codificación que tenemos de complejos y grupos) es igual al p grupo de homología $H_p(K) \in \mathcal{AB}_{fg}$, si la palabra K representa un complejo finito, y en otro caso, h_p es igual a la cadena vacía λ . Lo que deseamos es mostrar que h_p es una función computable, en el sentido de la Definición 1.2.7, por lo que debemos dar un algoritmo que siempre se detenga en cualquier entrada w , y nos de como resultado al terminar $h_p(w)$. Sea M_p una MT que ejecuta el siguiente programa:

$M_p =$ “En la entrada K :

1. Verificar que K representa un complejo; si no es así, borra todo el contenido de la cinta y al terminar ve al paso **13**.
2. Orienta a todos los p -simplejos de K .
3. Orienta a todos los $(p + 1)$ -simplejos de K .
4. Orienta a todos los $(p - 1)$ -simplejos de K .
5. Calcula la matriz A^p del operador ∂_p .
6. Calcula la matriz A^{p+1} del operador ∂_{p+1} .
7. Calcula las FNS N^p, N^{p+1} de las matrices A^p y A^{p+1} .
8. Calcula $z_p = (\# \text{ columnas de } N^p) - \text{rango } N^p$.
9. Calcula $w_p = \text{rango } N^{p+1}$.
10. Calcula $\beta_p = z_p - w_p$.
11. Encuentra las entradas b_1, \dots, b_l mayores a 1 en la matriz N^{p+1} .
12. Escribe en la cinta la palabra $\beta_p \# b_1 \# \dots \# b_l$.
13. Entra en el estado de *aceptación*.”

Veamos que el programa dado siempre termina. En el paso **1** M_p recorre toda su cadena de entrada, verificando que es en efecto una representación de un complejo simplicial, esencialmente basta con ver que los parentesis de la palabra están bien apareados (recordemos la palabra dada en (3.1))

⁸Aquí, los símbolos K y G , tienen dos significados. Por un lado, representan una palabra de símbolos en Σ^* ; por otro, representan el objeto matemático en sí (es decir, un complejo y un grupo). Este doble uso de los símbolos es común en computación y lo que se quiere decir se deduce del contexto en el que se encuentran.

y que los simplejos dados cumplen las condiciones de la definición de un complejo simplicial abstracto, lo cual lo podemos hacer en tiempo finito. Los pasos **2-4** son fácilmente llevados a cabo en tiempo finito (podemos tomar, por ejemplo, el orden en el que vienen especificados los vértices de los simplejos en la palabra que representa a K). Los pasos **5** y **6** son posibles debido a que podemos calcular las matrices de incidencia correspondientes. El paso **7** es posible debido a la existencia del algoritmo para calcular la FNS de cualquier matriz con entradas en \mathbb{Z} . Los pasos **8** y **9** los podemos llevar a cabo examinando las matrices en FNS N^p, N^{p+1} para determinar las columnas distintas de cero y el total de columnas de cada matriz. El paso **10** es trivialmente posible. Para el paso **11**, examinamos de nuevo a la matriz N^{p+1} para encontrar las entradas b_1, \dots, b_l mayores a 1. Los dos últimos pasos son fácilmente llevados a cabo en tiempo finito, con lo que tenemos que la máquina M_p siempre se detiene.

Por los resultados obtenidos en esta demostración, el algoritmo dado es correcto. Por lo tanto, cuando la cadena de entrada K representa a un complejo finito, la cadena que M_p escribe en su cinta al finalizar su cómputo

$$\beta_p \# b_1 \# \dots \# b_l$$

nos dice que K tiene número de Betti β_p y coeficientes de torsión b_1, \dots, b_l en dimensión p , con lo cual tenemos totalmente identificado al grupo $H_p(K)$. Y si K no representa a un complejo, entonces M_p borra todo el contenido de su cinta y se detiene, es decir, escribe la palabra vacía λ . Así, $M_p(K) = h_p(K)$ por lo que tenemos que h_p es una función computable. Esto demuestra el teorema. \square

Como mencionamos con anterioridad, nosotros sólo necesitamos un algoritmo para calcular el segundo número de Betti de un complejo finito. Analizando la prueba del teorema anterior, podemos deducir el siguiente

Corolario 3.3.5. *Sea K un complejo simplicial finito. Si A^p y A^{p+1} son las matrices de los homomorfismos ∂_p y ∂_{p+1} respectivamente, entonces*

$$\beta_p(K) = \text{rango } C_p(K) - \text{rango } A^p - \text{rango } A^{p+1}.$$

Demostración. Sabemos de la prueba del Teorema 3.3.4 que

$$\beta_p(K) = (\# \text{ columnas de } N^p) - \text{rango } N^p - \text{rango } N^{p+1}, \quad (3.2)$$

donde N^p, N^{p+1} son las matrices en FNS correspondientes a las matrices A^p y A^{p+1} . El número de columnas de N^p corresponde al cardinal de la base del grupo $C_p(K)$ con respecto a la cual, se calculó la matriz A^p de ∂_p . Como en el caso de las matrices sobre un campo, es posible ver que $\text{rango } N^p = \text{rango } A^p$. Combinando estos hechos, tenemos que (3.2) se convierte en

$$\beta_p(K) = \text{rango } C_p(K) - \text{rango } A^p - \text{rango } A^{p+1},$$

que es lo que queríamos demostrar. □

Corolario 3.3.6. *Existe un algoritmo que, al recibir como entrada un complejo simplicial finito K , calcula el número de Betti $\beta_p(K)$ (sin utilizar el algoritmo para obtener la FNS de una matriz con entradas en \mathbb{Z}).*

Demostración. Sea $f_p: \Sigma^* \rightarrow \Sigma^*$, dada por

$$f_p(K) = \begin{cases} \beta_p(K) & \text{si } K \in \mathcal{CS}, \\ \lambda & \text{en otro caso.} \end{cases}$$

Entonces f_p es una función de palabras computable. En efecto, construimos una MT M_{β_p} , con el siguiente programa:

M_{β_p} = “En la entrada K :

1. Verificar que K representa un complejo; si no es así, borra todo el contenido de la cinta y al terminar ve al paso **12**.
2. Orienta a todos los p -simplejos de K .
3. Orienta a todos los $(p+1)$ -simplejos de K .
4. Orienta a todos los $(p-1)$ -simplejos de K .
5. Calcula la matriz A^p del operador ∂_p .
6. Calcula la matriz A^{p+1} del operador ∂_{p+1} .
7. Calcula $\alpha_p = (\# \text{ } p\text{-simplejos de } K)$.
8. Calcula $r_p = \text{rango } A^p$.
9. Calcula $r_{p+1} = \text{rango } A^{p+1}$.
10. Calcula $\beta_p = \alpha_p - r_p - r_{p+1}$.
11. Escribe en la cinta la palabra β_p .
12. Entra en el estado de *aceptación*.”

Por el Corolario 3.3.5, este proceso es correcto. Para ver que M_{β_p} siempre se detiene, sólo necesitamos verificar los pasos **7**, **8** y **9**. El paso **7** es posible, ya que M_{β_p} simplemente cuenta todos los p -simplejos (que son precisamente la base que escogemos para el grupo $C_p(K)$) especificados en la palabra de entrada que describe a K . Para calcular el rango de las matrices A^p y A^{p+1} , usamos un algoritmo análogo al que tenemos para encontrar el rango de una matriz con entradas en \mathbb{R} . Nuestro algoritmo para matrices con entradas en \mathbb{Z} solo usa las siguientes operaciones elementales:

- (1) Intercambiar columnas.
- (2) Multiplicar la columna i por -1 .
- (3) Reemplazar la columna i por la nueva columna (columna i) ± 1 (columna k), donde $i \neq k$.

Con lo que los pasos **8** y **9** de M_{β_p} son realizables en tiempo finito. Así, M_{β_p} siempre se detiene, y además $M_{\beta_p}(K) = f_p(K)$, con lo que f_p es computable, y así obtenemos el corolario. □

Del Corolario 3.3.6, nos quedamos con la máquina M_{β_2} , con la cual tendremos la capacidad de calcular el segundo número de Betti $\beta_2(K)$, de un complejo dado K . Esto será pieza fundamental para nuestros resultados finales.

El grupo fundamental

Partiendo de las clases de homotopía de los caminos cerrados en un punto x de un espacio X , podemos formar otro invariante muy interesante, el llamado grupo fundamental (el primer grupo de homotopía) $\pi_1(X, x)$. Este invariante contiene mucha información acerca de la estructura de hoyos (de dimensión 1) del espacio X , es un grupo muy bien estudiado, que ha dado muchos frutos a la topología.

Y hay resultados muy interesantes para grupos fundamentales de los complejos simpliciales en cuanto a computabilidad. El desarrollo de estos resultados puede ser encontrado con todos los detalles en [Ro88, p. 164-179] y (a gran sorpresa del autor) en [Ro95, p. 367-401].

Definición 3.3.7. Sea K un n -complejo. Una arista en K es una pareja ordenada (u, v) , de vértices (no necesariamente distintos), tal que ambos están en un simplejo de K . Dos aristas $(u_1, v_1), (u_2, v_2)$ son adyacentes si $v_1 = u_2$. Si $u, v \in \text{Vert}(K)$, entonces un camino de u a v es una sucesión de aristas

$$\alpha = (u, v_1)(v_1, v_2) \cdots (v_{n-2}, v_{n-1})(v_{n-1}, v),$$

donde u y v son el origen y destino de α , respectivamente.

Usando este concepto, podemos (de manera totalmente análoga a las gráficas combinatorias) definir cuando un complejo simplicial es conexo, y sus componentes conexas (que se puede probar, son subcomplejos conexos máximos de K). Pero lo más importante, es que podemos definir, de una manera combinatoria, el producto y la relación de homotopía entre caminos de aristas (que fungen como los caminos continuos usuales) en K . El producto de dos caminos $\alpha = e_1 e_2 \cdots e_n, \beta = f_1 f_2 \cdots f_m$, tales que el origen de la arista f_1 es igual al destino de la arista e_n , es el camino

$$\alpha\beta = e_1 \cdots e_n f_1 \cdots f_m.$$

Y la relación de homotopía entre caminos de aristas se define así: Tenemos dos tipos de operaciones elementales en un camino α en K . La primera, reemplaza un par de aristas adyacentes de la forma $(u, v)(v, w)$ en α , por la arista (u, w) , si el 2-simplejo $\{u, v, w\}$ está en K . La segunda operación es la inversa de la primera, reemplazando a (u, w) por $(u, v)(v, w)$. Dos caminos α, β en K son *homotópicos*, si uno puede ser obtenido a partir del otro, por medio de una sucesión finita de operaciones elementales. Es sencillo ver que esta relación (combinatoria) de homotopía es de equivalencia, y que el producto de clases de caminos de aristas está bien definido. Entonces, tomando un vértice base, $v \in \text{Vert}(K)$, caminos cerrados en v , y definiendo el camino trivial en v como (v, v) , tenemos el conjunto de clases de equivalencia

$$\pi(K, v) = \{[\alpha] \mid \alpha \text{ es un camino cerrado en } v\}.$$

Teorema 3.3.8. (i) Para todo vértice v en el complejo K , $\pi(K, v)$ es un grupo, con identidad la clase del camino trivial en v .

(ii) Si L es la componente conexa de K que contiene a v , entonces $\pi(K, v) = \pi(L, v)$.

(iii) Si K es conexo, con vértices base v y u , entonces

$$\pi(K, v) \cong \pi(K, u).$$

(iv) Si $\varphi: K \rightarrow L$ es una función simplicial, entonces $\varphi_*: \pi(K, v) \rightarrow \pi(L, \varphi(v))$, definida por $\varphi_*([\alpha]) = [\varphi(\alpha)]$, es un homomorfismo de grupos.

(v) Sea K^2 el 2-esqueleto de K . Entonces la inclusión $i_2: K^2 \hookrightarrow K$, induce un isomorfismo $i_{2*}: \pi(K^2, v) \xrightarrow{\cong} \pi(K, v)$.

Al grupo $\pi(K, v)$, se le conoce como *el grupo de caminos cerrados de aristas del complejo K* (con vértice base, v). Por el inciso (iii), cuando K es conexo, podemos escribir simplemente $\pi(K)$. Notamos de inmediato la gran similitud de las propiedades del grupo fundamental $\pi_1(|K|, v)$ con las de $\pi(K, v)$. Esto no es mera coincidencia.

Teorema 3.3.9. *Para todo $v \in \text{Vert}(K)$, $\pi(K, v) \cong \pi_1(|K|, v)$.*

Como consecuencia de este teorema y del anterior, vemos que $\pi_1(|K|, v)$ depende solamente de la estructura del 2-esqueleto de K . Hasta este momento, la situación va por buen camino. Tenemos, para un complejo K , un grupo definido de manera combinatoria, que es isomorfo al grupo fundamental de $|K|$. Entonces, si conociéramos un método, por medio del cual, calcular el grupo de aristas, podríamos usarlo para calcular el grupo fundamental. Lo que veremos a continuación, es que podemos, a partir de un complejo dado, calcular una presentación del grupo de aristas. Necesitamos otras definiciones y resultados.

Definición 3.3.10. Un árbol es un complejo simplicial conexo, de dimensión a lo mas 1, tal que no contiene ciclos⁹ (el único árbol de dimensión 0, tiene sólo un vértice). Un subcomplejo $T \subseteq K$, es un árbol máximo, si T es un árbol que no está contenido en otro árbol más grande.

Lema 3.3.11. *Si K es conexo, entonces un árbol T es máximo $\Leftrightarrow \text{Vert}(T) = \text{Vert}(K)$.*

⁹En un complejo simplicial K , un camino $\alpha = e_1 \cdots e_n$ es *reducido*, si α es el camino trivial o ninguna arista $e_i = (u, v)$, es adyacente a su inversa y tampoco es el camino trivial. Un *ciclo* es un camino cerrado reducido. Se puede ver que todo camino es homotópico a un camino reducido o a un camino trivial.

Los árboles de los complejos son exactamente como los árboles definidos en las gráficas combinatorias. Todo complejo K tiene un árbol máximo (esto es sencillo de ver si K es finito, y una aplicación del Lema de Zorn muestra que es cierto en general). Usualmente, un complejo tendrá más de un árbol máximo.

Sea T un árbol máximo de K . Consideremos el grupo, $G(K, T)$, dado por la siguiente presentación:

- generadores:** Todas las aristas de K .
relaciones: Tipo a): $(u, v) = 1$, si (u, v) es una arista de T .
 Tipo b): $(u, v)(v, w) = (u, w)$, si $\{u, v, w\} \in K$.

El siguiente resultado relaciona al grupo $\pi_1(|K|)$ con $G(K, T)$ (a través de $\pi(K)$), en la forma que buscamos, dándonos una presentación para el grupo fundamental, llamado Teorema de Tietze.

Teorema 3.3.12 (H. Tietze, 1908). *Si K es un complejo conexo y T es un árbol máximo en K , entonces*

$$\pi(K) \cong G(K, T).$$

Corolario 3.3.13. *Existe un algoritmo que, al darle como entrada un complejo simplicial finito y conexo, $K \in \mathcal{CS}$, calcula una presentación finita de grupo μ , tal que $G(\mu) \cong \pi_1(|K|)$.*

Demostración. Construimos una MT M tal que, al recibir como entrada un complejo $K \in \mathcal{CS}$, esta primero verifica que K es conexo, si esta prueba falla, M rechaza de inmediato. Luego, M calcula un árbol máximo $T \subseteq K$, y con este, construye una presentación finita μ del grupo $G(K, T)$. Este proceso siempre termina, pues K es finito. Por los Teoremas 3.3.9 y 3.3.12, $G(\mu) \cong G(K, T) \cong \pi(K) \cong \pi_1(|K|)$, con lo que el algoritmo ejecutado por M es correcto. □

Observación. El algoritmo dado por el corolario anterior calcula solamente una presentación finita del grupo $\pi_1(|K|)$.

Así que, a partir de un complejo finito, podemos calcular de forma mecánica, una presentación de su grupo fundamental. Pero esto no termina ahí.

También podemos hacer la operación inversa, es decir, a partir de una presentación (finita) de grupo ν , construir un 2-complejo simplicial (finito), tal que su grupo fundamental, es precisamente $G(\nu)$ ¹⁰.

Teorema 3.3.14. *Dado un grupo G , existe un 2-complejo simplicial conexo K , tal que $G \cong \pi_1(|K|)$.*

Demostración. Ver [Ro88, p. 178-179] o [Ro95, p. 399-400]. □

Corolario 3.3.15. *Existe un algoritmo tal que, al recibir como entrada una presentación finita de grupo, μ , calcula un 2-complejo finito y conexo, K , tal que $G(\mu) \cong \pi_1(|K|)$.*

Demostración. La demostración del teorema anterior es de manera constructiva, así que al aplicarla a un grupo finitamente presentado, nos da un proceso que siempre termina, y calcula un complejo simplicial finito, con las propiedades requeridas. □

Corolario 3.3.16. *Un grupo G es finitamente presentado si y sólo sí, existe un 2-complejo finito y conexo K , tal que $G \cong \pi_1(|K|)$.*

Demostración. Inmediata de los Corolarios 3.3.15 y 3.3.13. □

Mostramos ahora como funcionan los algoritmos dados por los Corolarios 3.3.13 y 3.3.15. Sea $\mu = (\{a, b, c, e\} : \{a, e\})$, entonces $G(\mu)$ es un grupo libre de rango 2, ya que las relaciones $a = 1 = e$ nos dicen que a y e son iguales a la identidad en $G(\mu)$ (Equivalentemente, podemos obtener la presentación $(\{b, c\} : \emptyset)$ a partir de μ , aplicando dos operaciones Op_5^{-1}). Calculemos un 2-complejo simplicial, con grupo fundamental $G(\mu)$. Los pasos a seguir son los siguientes:

¹⁰La prueba de este importante resultado hace uso del Teorema de Seifert-van Kamper. Para no seguir introduciendo más conceptos, hemos optado por sólo enunciar el resultado que necesitamos e ilustrar con ejemplos el algoritmo con el cual construimos el complejo, a partir de la presentación de grupo.

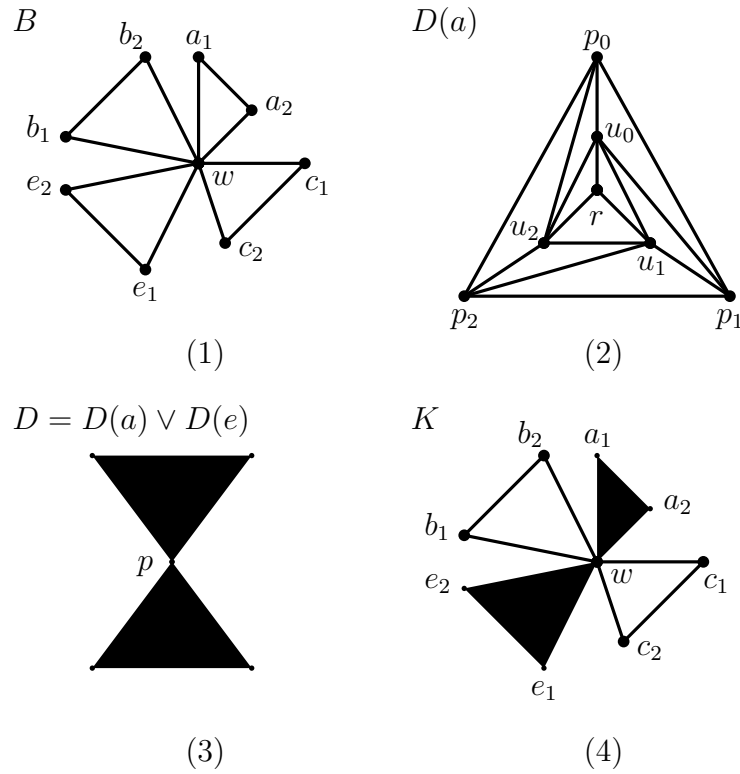


Figura 3.2: Aplicación del algoritmo del Corolario 3.3.15. Por simplicidad, en (3) y (4), hemos omitido los vértices y las aristas interiores en los triángulos sombreados.

1. Construimos el 1-complejo B , mostrador en la Figura 3.2 (1); una cuña de triángulos “huecos”, uno por cada generador de μ . De esta forma, hacemos corresponder a cada generador, un camino cerrado, y a cada relación le corresponde un camino de aristas en B . En este caso, a e le corresponde el camino $\alpha_e = (w, e_1)(e_1, e_2)(e_2, w)$ y para a tenemos el camino $\alpha_a = (w, a_1)(a_1, a_2)(a_2, w)$ (Si tuviéramos la relación ab , le correspondería el camino cerrado $\alpha_{ab} = (w, a_1)(a_1, a_2)(a_2, w)(w, b_1)(a_1, b_2)(b_2, w)$).
2. Para cada uno de las relaciones a, e , construimos los 2-complejos $D(a), D(e)$ (En la Figura 3.2 (2), mostramos a $D(a)$, $D(e)$ es totalmente similar. Estos son llamados “polígonos triangulados”). Cada complejo tiene

tantos vértices p_i 's y u_i 's, como tienen aristas los caminos asociados con las relaciones de μ . También definimos funciones $f_a: \partial D(a) \rightarrow B$, $f_a(p_i) = a_i$ (f_e es análoga, y hacemos $a_0 = e_0 = w$).

3. El siguiente paso es formar el 2-complejo $D = D(a) \vee D(e)$ (Figura 3.2 (3)), identificamos a cada p_0 en un vértice común, p . Si definimos a ∂D , como el subcomplejo inducido por los vértices p_i 's, entonces existe una función simplicial $f: \partial D \rightarrow B$, tal que $f|_{\partial D(x)} = f_x$, para $x \in \{a, e\}$.
4. Finalmente, construimos el complejo cociente K , a partir de la unión disjunta $B \cup D$, identificando a a_i con $f_a(p_i)$, y e_i con $f_e(p_i)$ (Figura 3.2 (4)). K es el complejo buscado. El algoritmo termina.

Ahora aplicamos al complejo K el algoritmo para construir una presentación de $\pi_1(|K|)$. El primer paso es obtener un árbol máximo, T , de K , el cual podemos ver en la Figura 3.3. De aquí, aplicamos las reglas para construir la presentación del grupo $G(K, T)$, y el resultado es una presentación ν , con más de 30 generadores y 20 relaciones. Sin embargo, es sencillo ver que muchos de los generadores, son equivalentes a la identidad en $G(\nu)$. Por ejemplo, tenemos los generadores (r, q_1) , (q_1, q_2) y (q_2, r) , y las relaciones $(r, q_1) = (q_2, r) = 1$ y $(r, q_1)(q_1, q_2) = (q_2, r)$, pues $\{r, q_1, q_2\} \in K$. Combinando las ecuaciones mencionadas, vemos que $(q_1, q_2) = 1$ en $G(\nu)$. Lo mismo pasa con todas las aristas que están en T , siendo las aristas (b_1, b_2) , (c_1, c_2) las únicas que no se pueden eliminar. De este modo, la presentación ν es equivalente a

$$(\{(b_1, b_2), (c_1, c_2)\}: \emptyset),$$

que es isomorfa a μ , y por lo tanto, $\pi_1(|K|)$ es un grupo libre de rango 2, tal cual es $G(\mu)$.

Terminamos nuestra discusión de la computabilidad del grupo fundamental con dos lemas que son consecuencia sencilla de los resultados mencionados. Serán importantes en el capítulo final.

Lema 3.3.17. *Existe un algoritmo para decidir, dado un complejo simplicial finito y conexo, K ,*

- i) *Si K es un complejo de dimensión 2.*
- ii) *Si K tiene dimensión 2, determinar todas las clases de congruencias de presentaciones de grupo μ_1, \dots, μ_n , tales que $G(\mu_i) \cong \pi_1(|K|)$.*

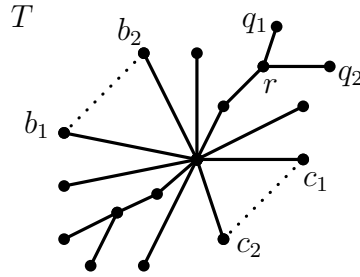


Figura 3.3: Árbol máximo del complejo K , para calcular una presentación finita de $\pi_1(|K|)$, usando el algoritmo dado por el Corolario 3.3.13.

Demostración. Es sencilla. Primero, examinamos a K para determinar si es un complejo de dimensión 2 (si no lo es, rechazamos de inmediato y el algoritmo termina). Después, aplicamos el Corolario 3.3.13, para obtener una presentación μ , de $\pi_1(|K|)$. Finalmente, generamos todas las permutaciones del conjunto de generadores de μ , y las aplicamos en la presentación μ , para obtener las presentaciones $\mu_1 = \mu, \dots, \mu_n$, ($n \in \mathbb{N}$), que son representantes de las clases de congruencia buscadas. \square

Lema 3.3.18. Si μ es una presentación de grupo y $n \geq 4$, entonces existe un complejo simplicial rectilíneo (con vértices racionales) $K_2 \subseteq \mathbb{R}^{n+1}$, tal que $G(\mu) \cong \pi_1(|K_2|)$.

Demostración. Sea μ una presentación de grupo y $n \geq 4$. Usando el Corolario 3.3.15, encontramos a θ_2 , un complejo simplicial abstracto (finito y conexo) de dimensión a lo más 2, tal que $G(\mu) \cong \pi_1(|\theta_2|)$. Luego, como $n + 1 \geq 5$, aplicamos el Teorema de encaje de complejos simpliciales en \mathbb{R}^{n+1} ([ST51, p. 57-58] y [Sp66, p. 120-121]). Al examinar la demostración de este, vemos que el complejo $K_2 \subseteq \mathbb{R}^{n+1}$, isomorfo a θ_2 , puede ser construido con vértices, vectores con coordenadas en \mathbb{Q} , y además, $\pi_1(|\theta_2|) \cong \pi_1(|K_2|)$, con lo que obtenemos el lema. \square

El problema del homeomorfismo

A. A. Markov, fue el primero en utilizar las técnicas y resultados de indecidibilidad en grupos para mostrar resultados computacionales en topología [Ma58]. Usando el hecho de que el problema de la palabra en grupos es indecidible, Markov fue capaz de construir una variedad M^n , en cada dimensión $n > 3$, tal que el problema de decidir si alguna variedad dada es homeomorfa a M^n , es no computable. Fue precisamente para esto que Markov introdujo el truco de la “relación vacía” (Sección 2.1.1) en las presentaciones de grupos, ya que necesitaba que diferentes presentaciones de un grupo dado, correspondieran a la misma variedad (al menos, que este fuera el caso para el grupo trivial). El resultado de Markov fue la chispa que encendió la llama, para que surgieran otros resultados sobre decidibilidad en topología, como el de Novikov¹¹, sobre la imposibilidad de poder reconocer a la esfera \mathbb{S}^n , $n \geq 5$, de una sucesión dada de n -variedades PL, y los resultados más generales obtenidos por Boone y compañía [BHP68]. Así, estos hechos muestran que la clasificación topológica de espacios es, en general, imposible.

Variedades simplemente conexas

Una variedad M es *simplemente conexa*, si $\pi_1(M) = 1$. Esta propiedad ha jugado siempre un papel importante en la topología, ya que es hipótesis fundamental en varios teoremas. Muchos resultados sobre la estructura de las variedades, obtenidos por Smale [Sm62], tienen como hipótesis a esta propiedad. Sería, pues, muy útil contar con un procedimiento finito para poder decidir, dada una variedad M (en la codificación finita de variedades que hemos asumido, existe), si M es o no simplemente conexa. Este es uno de los problemas de decisión que enfrentaremos en el último capítulo, por lo que debemos dar las definiciones pertinentes en el ámbito de conjuntos de palabras.

Definición 3.3.19. Decimos que la presentación finita \mathfrak{M} es simplemente conexa, si presenta una n -variedad simplemente conexa.

¹¹Una breve exposición de este (poco conocido) importante resultado aparece en un apéndice de [VKF77]. Es este resultado la razón por la cual, un esquema de complejo simplicial no es suficiente para poder representar a una variedad PL. No existe un algoritmo que pueda decirnos si un complejo dado, representa una variedad PL, ya que si existiera, el resultado de Novikov sería falso.

Para estudiar computacionalmente a esta propiedad de palabras, introducimos el lenguaje

$$\text{SIMPLEMENTE CONEXA}(C) = \{\mathfrak{M} \in C \mid \mathfrak{M} \text{ es simplemente conexa}\},$$

donde C es un conjunto de presentaciones finitas de variedades. Veremos que, desafortunadamente, no existe un algoritmo que en general nos permita decidir esta propiedad.

Los problemas de clasificación de variedades

Como hemos dicho, el problema del homeomorfismo en variedades es indecidible; pero, ¿Que hay de otros tipos de clasificaciones ?, ¿Será posible clasificar a todas las variedades diferenciables (combinatorias), de un dimensión dada, bajo difeomorfismo (equivalencia combinatoria) ? Consideremos pues, la clasificación de este tipo de variedades por difeomorfismo, homeomorfismo, equivalencia combinatoria y homotópica, denotadas por los símbolos $\approx_d, \approx_t, \approx_c, \approx_h$, respectivamente. Ya que tenemos presentaciones finitas de variedades C^∞ -PL cerradas, para manipularlas de forma algorítmica, podemos definir (en términos de propiedades de palabras) los problemas computacionales relacionados a las equivalencias de variedades dadas (difeomorfismo, homeomorfismo, equivalencia combinatoria y homotópica). De forma análoga a la Definición 2.1.3, tenemos

Definición 3.3.20. Dadas presentaciones finitas de variedades $\mathfrak{M}, \mathfrak{M}'$, decimos que son x -equivalentes ($x \in \{d, t, c, h\}$), y lo denotamos por $\mathfrak{M} \approx_x \mathfrak{M}'$, si presentan n -variedades que son x -equivalentes.

Dado un conjunto C , de presentaciones finitas de variedades, definimos el problema de x -equivalencia para C , como

$$x\text{-EQUIVALENCIA}(C) = \{(\mathfrak{M}, \mathfrak{M}') \in C \times C \mid \mathfrak{M} \approx_x \mathfrak{M}'\}.$$

Si $\mathfrak{N} \in C$, el problema de x -equivalencia con \mathfrak{N} es

$$x\text{-EQUIVALENCIA}(C, \mathfrak{N}) = \{\mathfrak{M} \in C \mid \mathfrak{M} \approx_x \mathfrak{N}\}.$$

Recordemos que, en dimensión ≤ 2 , tenemos una clasificación de variedades (de todos los tipos). En dimensión 3, la clasificación aún no está resuelta en

todos los casos, pero el conocimiento de hoy día, parece indicar que el problema ha sido finalmente resuelto¹. Sin embargo, no conocemos algoritmos que reciban como entrada, una presentación finita de variedad \mathfrak{M} , que represente una variedad de dimensión ≤ 3 , y nos diga que variedad es, usando las clasificaciones conocidas. Estos algoritmos deben de existir, pero parece ser que nadie se ha dado a la tarea de encontrarlos.

Nuestro objetivo principal es mostrar la imposibilidad algorítmica (en general) de la propiedad ser simplemente conexa y de las diferentes equivalencias de variedades (es decir, del problema de la x -equivalencia, con $x \in \{d, t, c, h\}$), en el fuerte sentido, de que podemos encontrar conjuntos de presentaciones de variedades tales que, los problemas mencionados tienen un grado arbitrario de insolubilidad, dentro del ámbito de los lenguajes c.e. (listables).

Todos estos son algunos de los resultados mas importantes que se conocen en cuanto a la computabilidad de propiedades y problemas topológicos, en las referencias al final de la obra, el lector puede encontrar mayores detalles de estos hechos y otros que no mencionamos. Aunque ha habido varios logros en cuanto a formas de representar de forma finita algunas clases de espacios topológicos, no se sabe hasta el momento, como podríamos codificar espacios un poco más generales. El encontrar algo como una “Presentación de espacio”, en el sentido en que se encontraron las presentaciones de grupos, y a partir de esto, encontrar presentaciones finitas de espacios, podría abrir el espectro para avanzar mucho más nuestros conocimientos computacionales sobre la topología.

La conexión entre la topología y la teoría de la computación es algo muy interesante y en cierta forma, extraño para muchos, pues se necesita combinar dos puntos de vista matemáticos muy diferentes: El continuo y el discreto; pero de esta combinación han surgido cosas muy interesantes¹². Indudablemente, esta conexión se debe seguir explotando a su máximo.

¹²Una de las conexiones mas recientes, ha sido el uso de la topología algebraica para probar resultados de Computación Distribuida [HS99, HR00, GR05].

Capítulo 4

Presentación finita de variedades diferenciables y combinatorias

Teniendo todas las herramientas preliminares de Computación, Álgebra y Topología que necesitamos, es tiempo de desarrollar el concepto de *presentación finita de variedades C^∞ -PL*. Como ya mencionamos, no tenemos un concepto tan general, como en el caso de grupos, donde sabemos que todo grupo tiene una presentación (no necesariamente finita), sin embargo, la clase de variedades que podremos representar en una computadora, es bastante amplia. Después de dar la definición de presentación finita de variedades, probaremos que toda variedad C^∞ -PL cerrada tiene una presentación. Ya que nuestros datos deben ser finitos, para alimentarlos a una computadora, no es sorpresa que nuestras variedades sean compactas. En el desarrollo del concepto de presentación finita de variedades, hace su aparición otra rama muy importante de las matemáticas: La Geometría Algebraica, que tiene una importante conexión con la topología diferencial, desarrollada por John Nash [Na52]. Hablaremos brevemente de esto. Algo que veremos con las presentaciones finitas de variedades, es que su definición es muy grande, y debe cumplir con muchas condiciones no triviales, lo cual es muy probable que haya desalentado a los matemáticos de usarlas para conocer más sobre ellas.

4.1. Variedades C^∞ -algebraicas

Antes de introducir el concepto de presentación finita de variedades, debemos entender (al menos, en un nivel básico) la conexión existente entre la geometría algebraica y la geometría diferencial. Resaltaremos los puntos más importantes, que son piezas clave para nuestro objetivo.

En 1952, se publicó un artículo de John Nash [Na52], en donde se desarrollan varias conexiones entre topología diferencial y geometría algebraica sobre los reales, y se refieren a las formas geométricas que puede asumir una variedad algebraica real. La idea de Nash era ver cuando una variedad diferenciable tiene una “representación algebraica”, lo cual lo llevo a definir objetos que llamaremos “*variedades C^∞ -algebraicas*”¹, que son esencialmente pedazos de variedades algebraicas que tienen una estructura diferenciable. He aquí algunos ejemplos de variedades C^∞ -algebraicas, que son del uso cotidiano de muchos matemáticos:

- La esfera $\mathbb{S}^n = \{(x_1, \dots, x_{n+1}) \in \mathbb{R}^{n+1} \mid x_1^2 + \dots + x_{n+1}^2 - 1 = 0\}$, y el disco $D^n = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_1^2 + \dots + x_n^2 - 1 \leq 0\}$.
- La esfera de Riemann.
- Curvas elípticas.
- Grassmanianas.

Como ya mencionamos, en general una representación algebraica de una variedad diferenciable será una porción de una variedad algebraica, a la que llamaremos una *componente*.

Definición 4.1.1. Sea V una variedad algebraica real. Una componente de V es un subconjunto $\mathcal{B} \subseteq V$, que satisface las siguientes propiedades:

- a) Dados $x, y \in \mathcal{B}$, existe un camino analítico entre x y y , enteramente contenido en \mathcal{B} .
- b) \mathcal{B} debe ser máximo entre los subconjuntos de V que gozan de la propiedad anterior.

¹En ingles, el término usado es *Algebraic Manifolds*.

c) Existe un punto de \mathcal{B} , el cual tiene una vecindad N , tal que $N \cap V \subseteq \mathcal{B}$.

Los resultados obtenidos por Nash se centran precisamente en las variedades diferenciables cerradas. Tres de ellos, son de gran importancia para nosotros, en nuestra tarea de desarrollar una forma de representar finitamente a las variedades diferenciables. Los dos primeros, son resultados técnicos, usados en la demostración del Teorema 1 de [Na52].

Lema 4.1.2. *Si M es una variedad encajada analíticamente en \mathbb{R}^n , entonces M tiene una vecindad $\mathcal{N} \subseteq \mathbb{R}^n$, en la cual existe un único punto $z \in M$, para cada punto $x \in \mathcal{N}$, y es tal que z depende analíticamente de x .*

Lema 4.1.3. *Sea f una función real, analítica en un abierto $\mathcal{C} \subseteq \mathbb{R}^n$, tal que \mathcal{C} contiene una región cerrada \mathcal{A} , acotada por una variedad analítica cerrada \mathcal{B} . Entonces existe una sucesión de polinomios (en las coordenadas de \mathbb{R}^n) $\{p_v\}$, que converge uniformemente a f en \mathcal{A} . Más aún, para cualquier tipo de derivación D , con respecto a las coordenadas de \mathbb{R}^n , tenemos que $\{Dp_v\}$ converge uniformemente a Df en \mathcal{A} .*

Este último lema es de gran importancia, ya que permite aproximar funciones reales analíticas, y sus derivadas, por medio de sucesiones de polinomios, que pueden ser escogidos de tal forma que tengan coeficientes en \mathbb{Q} (y por lo tanto, serán manipulables por medio de algoritmos). El teorema principal de [Na52] nos dice que toda variedad diferenciable cerrada tiene una representación algebraica está basado en aproximaciones de los varios objetos que se utilizan (funciones, operadores lineales, sus derivadas), es decir, las afirmaciones dadas en la demostración son ciertas, si las aproximaciones son lo suficientemente buenas.

Teorema 4.1.4 (John Nash [Na52]). *Sea \mathcal{D} un encaje diferenciable en \mathbb{R}^n de una variedad diferenciable cerrada M . Entonces \mathcal{D} puede ser aproximado por una representación algebraica de la variedad M .*

Ilustremos con un ejemplo, las ideas principales de la demostración del Teorema 4.1.4. Supongamos que M es una 5-variedad en \mathbb{R}^8 . Lo que hacemos primero, es encajar a M analíticamente, usando un resultado de Whitney [Hi76], sea \mathcal{D} tal encaje. A continuación, trabajamos en una pequeña vecindad \mathcal{N} de \mathcal{D} , en donde cada punto $x \in \mathbb{R}^8$, tiene un único punto $y(x)$, en \mathcal{D} (Lema 4.1.2). Con esto, y usando el Lema 4.1.3, obtenemos un vector función, $\mathbf{u}(x)$, el cual tiene componentes polinomiales, y aproxima al vector $y(x) - x$.

Usando funciones algebraicas en las coordenadas de \mathbb{R}^8 , describimos un 3-plano ($3 = 8 - 5$) $P(x)$, que contiene a x , y es aproximadamente igual al 3-plano $K(x)$, normal a \mathcal{D} en $y(x)$. A $P(x)$ lo describimos por medio de una matriz simétrica $\mathbf{L}(x)$, con entradas polinomiales, que aproxima a la matriz $\mathbf{K}(x)$, que describe al plano $K(x)$. La condición que define a la variedad algebraicamente encajada, es que la función \mathbf{u} debe ser normal al plano $P(x)$. En este ejemplo, esto se reduce a 3 condiciones algebraicas, pues $\dim P = 3$, por lo que debe ser posible en una 5-variedad dentro de \mathbb{R}^8 . Sin embargo, para que tengamos una buena aproximación algebraica a \mathcal{D} , aparte de que \mathbf{u} aproxime al vector $y(x) - x$, también debemos tener que \mathbf{u} tenga derivadas aproximando las de $y(x) - x$.

Sea $\alpha(\lambda)$ el polinomio característico de \mathbf{L} . Como \mathbf{L} aproxima a \mathbf{K} , entonces $\alpha(\lambda)$ aproxima al polinomio $b(\lambda) = \lambda^3(\lambda - 1)^5$, y por lo tanto, las raíces de α aproximan a las de b . Sea $\beta(\lambda)$ el factor de α que contiene a las 3 raíces “pequeñas” (es decir, a las que se aproximan a cero), y sea $\mathbf{P} = \beta(\mathbf{L})$. Se puede ver que como \mathbf{L} , \mathbf{P} también aproxima a \mathbf{K} . Entonces el sistema dado por

$$\mathbf{P}\mathbf{u} = 0,$$

determina una componente \mathcal{B} , de una variedad algebraica. \mathcal{B} es la aproximación algebraica a \mathcal{D} . Hay muchos detalles interesantes que hemos omitido, y pueden ser encontrados en [Na52]. Dos de los objetos que más nos interesan para construir presentaciones finitas de variedades, son precisamente las matrices \mathbf{L} , \mathbf{u} , pues es a partir de estas que podemos calcular (usando aproximaciones) el atlas de una variedad diferenciable, por lo que estas matrices son la esencia de las presentaciones que en la siguiente sección, mostraremos que existen.

4.2. Codificación de variedades

La tarea de desarrollar un concepto funcional de “presentación finita de una variedad” es algo no trivial. Cuando deseamos desarrollar una codificación para algún objeto (matemático, o de la vida real) es necesario que esta guarde la información importante sobre el objeto, y debe proporcionar métodos para que un programa (MT) pueda obtener esta información y manipularla. Básicamente, lo que describe a una variedad C^∞ -PL, es su atlas, así que necesitamos una forma de poder acceder a los homeomorfismos C^∞ -PL. Como ya mencionamos en el Capítulo 3, para dimensión mayor a 3,

un esquema de complejo simplicial abstracto no basta para poder representar una variedad LP, pues no existe algoritmo que nos diga si un complejo dado, representa una variedad combinatoria (pues tal algoritmo implicaría un algoritmo para el problema de equivalencia combinatoria con la n -esfera [VKF77]) Y lo mismo sucede con una representación por medio de matrices de incidencia o de bloques. Necesitamos algo más elaborado.

Para comenzar, pediremos que cualquier definición que demos de “presentación finita \mathfrak{M} , para construir una variedad M ” satisfaga las siguientes condiciones [BHP68]:

- (a) \mathfrak{M} esta dada como una sucesión finita de símbolos en algún alfabeto, Σ .
- (b) Existe un algoritmo para determinar si alguna palabra de símbolos en Σ es una presentación finita.
- (c) Para cada presentación finita \mathfrak{M} , existe una única n -variedad M , que es presentada por \mathfrak{M} .
- (d) \mathfrak{M} describe a M en una forma “natural”.

La manera de interpretar la condición (d), es que una presentación finita \mathfrak{M} de una variedad diferenciable y combinatoria, debe describir una triangulación Δ y un C^∞ -atlas Φ de M .

Ahora definiremos la presentación finita \mathfrak{M} de tal forma que nos proporcione todos los ingredientes necesarios para construir a la variedad diferenciable que representa. Necesitamos que \mathfrak{M} describa:

- (I) Un espacio euclidiano $\mathbb{E}_q(\mathfrak{M}) = \mathbb{R}^q$. Esto lo lograremos colocando en \mathfrak{M} q letras x_1, \dots, x_q que representarán las coordenadas del espacio \mathbb{R}^q .
- (II) Un complejo simplicial rectilíneo de dimensión n , $\Delta(\mathfrak{M}) \subseteq \mathbb{E}_q(\mathfrak{M})$ con vértices racionales. \mathfrak{M} tendrá especificados un conjunto p_1, \dots, p_s de puntos de \mathbb{R}^q con coordenadas racionales, que codifican los vértices de un n -complejo simplicial θ , que representa a $\Delta(\mathfrak{M})$.

- (III) Para cada estrella cerrada de $\Delta(\mathfrak{M})$, un homeomorfismo semilineal en un n -subespacio de $\mathbb{E}_q(\mathfrak{M})$ (Lo cual hace evidente que $\Delta(\mathfrak{M})$ es una n -variedad combinatoria). Dado un vértice p_k , colocaremos en \mathfrak{M} una tupla de n enteros $i_{1k} < i_{2k} < \dots < i_{nk} \leq q$, que nos indica en que subespacio de dimensión n de \mathbb{R}^q se proyecta la estrella cerrada de p_k . En otras palabras, esta tupla de enteros nos indica que la proyección $(x_1, \dots, x_q) \mapsto (x_{i_{1k}}, \dots, x_{i_{nk}})$ induce un homeomorfismo semilineal de $\text{St}_{\Delta(\mathfrak{M})}(p_k)$ en \mathbb{R}^n .
- (IV) Para cada estrella abierta de $|\Delta(\mathfrak{M})|$, un homeomorfismo en un n -subespacio de $\mathbb{E}_q(\mathfrak{M})$ tal que estos homeomorfismos forman un C^∞ -atlas $\Phi(\mathfrak{M})$ en $|\Delta(\mathfrak{M})|$ y son descritos con un conjunto de ecuaciones algebraicas, que serán deducidas de la matriz \mathbf{L} y el vector \mathbf{u} de la demostración del Teorema 4.1.4, que tienen como componentes a polinomios en variables que representan las coordenadas x_1, \dots, x_q de $\mathbb{E}_q(\mathfrak{M})$, esto se logra usando las técnicas desarrolladas por Nash [Na52].

Definición 4.2.1. Una presentación algebraica de atlas \mathfrak{M} de una n -variedad cerrada con una estructura combinatoria y una estructura diferenciable compatible, es una colección ordenada

$$\mathfrak{M} = (x_1, \dots, x_q; p_1, \dots, p_s; \theta; \mathbf{i}_1, \dots, \mathbf{i}_s; \mathbf{L}; \mathbf{u}; \delta, \epsilon, D)$$

que cumple las siguientes propiedades:

- (I) x_1, \dots, x_q son símbolos, llamados variables coordenadas o simplemente coordenadas. Denotamos por $\mathbb{E}_q(\mathfrak{M})$ el espacio euclidiano \mathbb{R}^q con coordenadas x_1, \dots, x_q presentado por \mathfrak{M} .
- (II) Los vectores columnas $p_1, \dots, p_s \in M_{q \times 1}(\mathbb{Q})$ son todos diferentes entre sí; θ es un complejo simplicial de dimensión n con vértices p_1, \dots, p_s .

Requerimos que las p_i 's y θ satisfagan:

- (IIa) Si $\{p_{j_0}, \dots, p_{j_m}\} \in \theta$, entonces p_{j_0}, \dots, p_{j_m} están en posición general, es decir, existe un m -simplejo (rectilíneo) σ_m en $\mathbb{E}_q(\mathfrak{M})$ con vértices p_{j_0}, \dots, p_{j_m} .
- (IIb) El conjunto de todos los simplejos que corresponden a elementos de θ en el sentido de (IIa), forma un complejo simplicial rectilíneo en

$\mathbb{E}_q(\mathfrak{M})$, “El complejo simplicial presentado por \mathfrak{M} ” y que denotamos por $\Delta(\mathfrak{M})$; además, el complejo frontera de $\Delta(\mathfrak{M})$ es vacío.

(III) Cada \mathbf{i}_k ($k \in \{1, \dots, s\}$) es una n -tupla de enteros positivos $i_{1k} < i_{2k} < \dots < i_{nk} \leq q$ tal que la estrella cerrada (compacta) $\text{St}_{\Delta(\mathfrak{M})}(p_k)$ de p_k en $\Delta(\mathfrak{M})$, se proyecta 1-1 en el espacio coordinado $\mathbb{E}_n(\mathbf{i}_k)$, con coordenadas $x_{i_{1k}}, \dots, x_{i_{nk}}$. Es decir, la función $\iota_k: \mathbb{E}_q(\mathfrak{M}) \rightarrow \mathbb{E}_n(\mathbf{i}_k)$ que lleva un punto con coordenadas x_1^*, \dots, x_q^* en el punto con coordenadas $x_{i_{1k}}^*, \dots, x_{i_{nk}}^*$ induce un homeomorfismo semilineal de $\text{St}_{\Delta(\mathfrak{M})}(p_k)$ en $\mathbb{E}_n(\mathbf{i}_k)$.

(IV) \mathbf{L} es una matriz simétrica de tamaño $q \times q$ y \mathbf{u} es un vector de q coordenadas, ambos con entradas en $\mathbb{Q}[x_1, \dots, x_q]$. $\delta, \epsilon, D \in \mathbb{Q}^+$, son tales que $\epsilon < \frac{1}{2n}$; $D < 1$. Sea B_k la q -bola en $\mathbb{E}_q(\mathfrak{M})$ ($k = 1, \dots, s$) con radio δ y con centro p_k . Requerimos que se cumplan las siguientes propiedades:

(IVa) Para cada punto en el conjunto $\bigcup_{k=1}^s B_k$, \mathbf{L} posee n eigenvalores a_1, \dots, a_n tales que

$$|a_i| < \epsilon/n \quad (1 \leq i \leq n),$$

y $q - n$ eigenvalores b_1, \dots, b_{q-n} tales que

$$|b_j - 1| < \epsilon/n \quad (1 \leq j \leq q - n).$$

Esto es equivalente a la condición de que los coeficientes del polinomio característico $\alpha(\lambda)$ de \mathbf{L} (con el coeficiente de grado mayor normalizado a 1) estén suficientemente próximos a los coeficientes de $\lambda^n(\lambda - 1)^{q-n}$.

(IVb) Ningún 1-simplejo de $\Delta(\mathfrak{M})$ tiene una longitud mayor a $\frac{1}{2}\delta$.

Sea $\beta(\lambda) = \lambda^n + \beta_{n-1}\lambda^{n-1} + \dots + \beta_1\lambda + \beta_0$ el factor de $\alpha(\lambda)$ que tiene a los n eigenvalores que se aproximan a cero (con coeficiente de grado mayor normalizado a 1). Por Nash [Na52, p. 412], los coeficientes de $\beta(\lambda)$ son funciones reales, analíticas en las x_i 's, y por (IVa), tenemos que $|\beta_k| < \epsilon$ para cada $k = 0, \dots, n - 1$. Más aún, sean $\mathbf{P} = \beta(\mathbf{L})$, $\Psi = \mathbf{P}\mathbf{u}$ y Ψ_1, \dots, Ψ_q las componentes de Ψ . Finalmente, sean $i'_{1k}, \dots, i'_{q-nk} \in \{1, \dots, q\} - \mathbf{i}_k$. Entonces, para cada $k = 1, \dots, s$ pedimos que se cumplan las siguientes propiedades:

(IVc) El valor absoluto del Jacobiano

$$\frac{\partial(\Psi_{i'_{1k}}, \dots, \Psi_{i'_{q-nk}})}{\partial(x_{i'_{1k}}, \dots, x_{i'_{q-nk}})} \text{ es mas grande que } D \text{ en } B_k.$$

(IVd) El sistema de $2q - n$ ecuaciones

$$(G_k) \begin{cases} \Psi_{i'_{1k}} = \dots = \Psi_{i'_{q-nk}} = 0 \\ \mathfrak{X}^* - \mathfrak{X} = \eta_1 \nabla \Psi_{i'_{1k}} + \dots + \eta_{q-n} \nabla \Psi_{i'_{q-nk}}, \end{cases}$$

junto con la desigualdad $|\mathfrak{X}^* - \mathfrak{X}| < \frac{1}{2}\delta$, donde $\mathfrak{X}^*, \mathfrak{X}$ son matrices de tamaño $q \times 1$ con componentes x_1, \dots, x_q y x_1^*, \dots, x_q^* respectivamente, definen una función 1-1, digamos $g_k: \text{Int St}_{\Delta(\mathfrak{M})}(p_k) \rightarrow \mathbb{E}_n(i_k)$ si las matrices coordenadas de todos los puntos de $\text{Int St}_{\Delta(\mathfrak{M})}(p_k)$ son tomados como el rango de \mathfrak{X}^* . En particular, si $\{p_k, p_{k_1}, \dots, p_{k_n}\}$ es un n -simplejo de θ y si $\mathfrak{X}^* = \zeta_1 p_{k_1} + \dots + \zeta_n p_{k_n} + (1 - \zeta_1 - \dots - \zeta_n)p_k$, entonces los jacobianos del sistema (G_k) (en las $2q$ variables $x_1, \dots, x_q, \eta_1, \dots, \eta_{q-n}, \zeta_1, \dots, \zeta_n$) con respecto a $x_{i'_{1k}}, \dots, x_{i'_{q-nk}}, \eta_1, \dots, \eta_{q-n}, \zeta_1, \dots, \zeta_n$ y con respecto a $x_1, \dots, x_q, \eta_1, \dots, \eta_{q-n}$, respectivamente, ambos tienen valores absolutos mayores a D si $|\mathfrak{X}^* - p_k| \leq \frac{1}{2}\delta$ y $|\mathfrak{X}^* - \mathfrak{X}| < \frac{1}{2}\delta$. Además, la función $g_k \circ g_l^{-1}|_{g_l(\text{Int St}_{\Delta(\mathfrak{M})}(p_k) \cap \text{Int St}_{\Delta(\mathfrak{M})}(p_l))}$, tiene jacobiano distinto de cero para toda l , tal que $p_l \in \partial \text{St}_{\Delta(\mathfrak{M})}(p_k)$.

Sea \mathbb{E}_n el subespacio de $\mathbb{E}_q(\mathfrak{M})$ con coordenadas x_1, \dots, x_n , y sea $\kappa_k: \mathbb{E}_n(i_k) \rightarrow \mathbb{E}_n$ el homeomorfismo lineal que mapea el eje $x_{i_{lk}}$ en el eje x_l ($l = 1, \dots, n$). Entonces el atlas $\Phi(\mathfrak{M})$, presentado por \mathfrak{M} , es la colección de cartas $h_k = \kappa_k \circ g_k$ ($k = 1, \dots, s$). La n -variedad diferenciable y combinatoria $M(\mathfrak{M})$ presentada por \mathfrak{M} es el espacio $|\Delta(\mathfrak{M})|$ junto con la estructura combinatoria $\Delta(\mathfrak{M})$ y la estructura diferenciable $\Phi(\mathfrak{M})$.

El siguiente resultado, nos da la seguridad de que la definición de presentación algebraica de atlas, aunque muy larga, cumple en gran medida con los criterios de una buena presentación finita de una n -variedad C^∞ -PL [BHP68].

Teorema 4.2.2. *Para toda n -variedad diferenciable cerrada \hat{M} , existe una presentación algebraica (finita) de atlas \mathfrak{M} tal que la variedad $M(\mathfrak{M})$ presentada por \mathfrak{M} es difeomorfa a \hat{M} .*

Más aún, el concepto de presentación algebraica finita de atlas satisface los requerimientos **(a)**, **(b)** y **(c)** mencionados al inicio de esta sección, y con respecto a **(d)**, el siguiente **(d')**: Dada una presentación algebraica de atlas \mathfrak{M} , el C^∞ -atlas $\Phi(\mathfrak{M})$, presentado por \mathfrak{M} puede ser aproximado de forma computable. Lo que esto quiere decir es (usando la notación de la Definición 4.2.1) que existe un algoritmo para resolver el siguiente problema: Dado un número racional positivo τ y las componentes de un punto racional

$$p \in \text{Int St}_{\Delta(\mathfrak{M})}(p_k)$$

determinar las componentes de un punto racional $h_k^\#(p) \in \mathbb{R}^n$ tal que

$$|h_k^\#(p) - h_k(p)| < \tau.$$

Observación. Este hecho hace un fuerte contraste con la existencia de grupos, los cuales no son finitamente presentables, y números reales cuyas expansiones decimales no son computables.

Demostración. Sea \hat{M} una variedad diferenciable cerrada. Primero debemos encontrar a todos los elementos que necesitamos para formar una presentación algebraica finita de atlas \mathfrak{M} para \hat{M} . Por un resultado de Whitney [Hi76, Teorema 4.7.1, p. 118], existe una variedad \mathcal{D} , difeomorfa a \hat{M} , y encajada analíticamente en un euclidiano \mathbb{R}^q (para q suficientemente grande). Tomamos las variables coordenadas x_1, \dots, x_q , como las letras que son parte del alfabeto con el que construiremos la presentación \mathfrak{M} .

Por el Lema 4.1.2, existe una vecindad \mathcal{N} de \mathcal{D} , en \mathbb{R}^q , con la propiedad de que cada punto $p \in \mathcal{N}$ tiene un único punto $d \in \mathcal{D}$, que es el más cercano; y dado $\epsilon \in \mathbb{Q}^+$, existen matrices \mathbf{L} y \mathbf{u} con componentes polinomiales, tal que \mathbf{u} aproxima al vector $d - p$ y \mathbf{L} representa un plano que es aproximadamente igual al plano K normal a \mathcal{D} en d . Además, el polinomio característico $\alpha(\lambda)$ de \mathbf{L} tiene n raíces aproximadamente igual a cero y $q - n$ raíces aproximadamente igual a 1 (pues recordemos que \mathbf{L} aproxima a la matriz que representa al plano K , y por lo tanto $\alpha(\lambda)$ aproxima al polinomio $\lambda^n(\lambda - 1)^{q-n}$). Si $\beta(\lambda)$ es el factor de $\alpha(\lambda)$ que tiene las n raíces pequeñas y si $\mathbf{P} = \beta(\mathbf{L})$, $\Psi = \mathbf{P}\mathbf{u}$, las ecuaciones dadas por $\Psi = 0$, determinan una componente $\mathcal{B} \subseteq \mathcal{N}$, de una variedad algebraica que aproxima a \mathcal{D} . Y en [Na52], se demuestra que existe un homeomorfismo analítico, de la vecindad \mathcal{N} de \mathcal{D} , sobre una vecindad \mathcal{N}^* de \mathcal{B} , el cual lleva a \mathcal{D} en \mathcal{B} , de tal forma que cada punto en \mathcal{B} , tiene como preimagen a su único punto más cercano en \mathcal{D} (más aún, cada punto

de \mathcal{N} se mueve dentro de un plano normal a \mathcal{D}). Este homeomorfismo induce una estructura diferenciable en \mathcal{B} , a partir de la de \mathcal{D} . Por lo tanto, \mathcal{B} es difeomorfa a \hat{M} y está encajada analíticamente en \mathbb{R}^q . de ahora en adelante, asumiremos que \mathcal{B} tiene una estructura diferenciable, que será precisamente la que le es inducida por \mathcal{D} , a partir del homeomorfismo analítico. De la prueba de Nash del Lema 4.1.3, se puede ver que los coeficientes de los polinomios de las matrices \mathbf{L} y \mathbf{u} pueden ser escogidos como números racionales. Así que podemos tomar a estas matrices y el número ϵ , como los elementos \mathbf{L} , \mathbf{u} y ϵ de la presentación \mathfrak{M} .

Para cada $p \in \mathcal{N}$, Ψ aproxima a la matriz de coeficientes del vector de distancia de p hacia su punto más cercano \hat{p} en \mathcal{D} . Por lo tanto, si Ω denota el plano de dimensión $q-n$, que es normal a \mathcal{B} en \hat{p} , el vector gradiente $\nabla\Psi_i$ ($i = 1, \dots, q$) yace aproximadamente en la dirección de la proyección del eje x_i en Ω ; y $|\nabla\Psi_i|$ es aproximadamente igual al coseno del ángulo entre el eje x_i y su proyección en Ω . Dado esto, si $i'_1 < \dots < i'_{q-n} \leq q$, son enteros positivos, tales que Ω se proyecta 1-1 en el espacio \mathbb{R}^{q-n} (con coordenadas $x_{i'_1}, \dots, x_{i'_{q-n}}$), entonces los vectores $\nabla\Psi_{i'_1}, \dots, \nabla\Psi_{i'_{q-n}}$ son linealmente independientes, y por tanto, también sus proyecciones en \mathbb{R}^{q-n} , es decir,

$$\frac{\partial(\Psi_{i'_1}, \dots, \Psi_{i'_{q-n}})}{\partial(x_{i'_1}, \dots, x_{i'_{q-n}})} \neq 0.$$

Ahora, existen números positivos $\delta, D \in \mathbb{Q}$, tales que (i) para cada punto de \mathcal{D} , su vecindad de radio 2δ , esta contenida en \mathcal{N} , y (ii) para cada q -bola B de radio δ en \mathcal{N} , existe una $(q-n)$ -tupla $i'_1 < \dots < i'_{q-n}$ de enteros tal que

$$\left| \frac{\partial(\Psi_{i'_1}, \dots, \Psi_{i'_{q-n}})}{\partial(x_{i'_1}, \dots, x_{i'_{q-n}})} \right| > D. \quad (4.1)$$

Así, para todo punto $p \in B$, tomamos δ y D como los elementos requeridos en \mathfrak{M} .

Por un resultado de Cairns [Ca35, Ca61-2], existe una triangulación $\hat{\Delta}$ de \mathcal{B} , (compatible con la estructura diferenciable de \mathcal{B}). Para cada $\hat{p}_k \in \text{Vert}(\hat{\Delta})$, escogemos un punto cercano $p_k \in \mathcal{N}$, con coordenadas en \mathbb{Q} , con lo que podemos usar a estos puntos como las matrices p_1, \dots, p_s en \mathfrak{M} . Y como el complejo abstracto θ , tomamos el inducido por los vértices p_1, \dots, p_s y el esquema de $\hat{\Delta}$. El complejo simplicial rectilíneo, isomorfo a $\hat{\Delta}$, lo denotamos por $\Delta(\mathfrak{M})$; y a la δ -vecindad de p_k en \mathbb{R}^q , por B_k . Como cada tupla de

enteros \mathbf{i}_k ($k \in \{1, \dots, s\}$), tomamos el conjunto $\{1, \dots, q\} - \{i'_{1k}, \dots, i'_{q-nk}\}$, donde $i'_{1k}, \dots, i'_{q-nk}$, son los enteros i'_1, \dots, i'_{q-n} correspondientes a B_k , en el sentido dado en el párrafo anterior. Hemos encontrado a todos los elementos requeridos para la presentación \mathfrak{M} , y ahora debemos ver que estos cumplen con todas los requisitos de la Definición 4.2.1.

(I), (II), (IIa), (IIb). Es claro que los elementos de \mathfrak{M} cumplen con estas propiedades.

(III). Si escogemos a δ lo suficientemente pequeño, la estrella cerrada $\text{St}_{\Delta(\mathfrak{M})}(p_k)$ estará muy cercana a una vecindad de \hat{p}_k en el n -plano Π , tangente a \mathcal{B} en \hat{p}_k , de modo que se proyectará (en la forma en que Π lo hace) 1-1 en el n -espacio $\mathbb{E}_n(\mathbf{i}_k)$, con lo que se satisface está condición.

(IV). Las matrices \mathbf{L} y \mathbf{u} tienen los tamaños y las entradas requeridos por esta condición y como para cada punto $p \in \mathcal{N}$, \mathbf{L} representa un plano que es aproximadamente igual al plano K normal a \mathcal{D} en el único punto d más cercano a p , entonces es claro que \mathbf{L} es una matriz simétrica, pues aproxima al operador lineal que proyecta cada punto $x \in \mathbb{R}^q$ en sus componentes que son paralelas a K ; y si ϵ, D son lo suficientemente pequeños, tendremos que $\epsilon < \frac{1}{2n}$ y $D < 1$, con lo que cumplimos con esta propiedad.

(IVa). \mathbf{L} puede satisfacer esta condición siempre que $\bigcup_{k=1}^s B_k \subseteq \mathcal{N}$.

(IVb). La triangulación $\hat{\Delta}$ puede ser escogida tan fina, de tal forma que ninguno de sus 1-simplejos tenga longitud mayor a $\frac{1}{2}\delta$, y esta propiedad será preservada en los 1-simplejos de $\Delta(\mathfrak{M})$.

(IVc). Ya vimos que para cada q -bola B de radio δ en \mathcal{N} existe una tupla de $q - n$ enteros tales que se cumple la desigualdad dada en (4.1). En particular, si tomamos a la bola B_k alrededor de p_k ($k = 1, \dots, s$), tenemos que los enteros $\{1, \dots, q\} - \mathbf{i}_k = \{i'_{1k}, \dots, i'_{q-nk}\}$ son tales que el valor absoluto del Jacobiano de la transformación $(x_{i'_{1k}}, \dots, x_{i'_{q-nk}}) \mapsto (\Psi_{i'_{1k}}, \dots, \Psi_{i'_{q-nk}})$ es mas grande que D en B_k , y con esto se cumple **(IVc)**.

(IVd). Sea $\mathfrak{X} \in \mathcal{B}$; si Ω es el plano normal a \mathcal{B} en \mathfrak{X} , entonces una vecindad de radio $\frac{1}{2}\delta$ de \mathfrak{X} en Ω , intersectará a $|\Delta(\mathfrak{M})|$ precisamente en un punto, \mathfrak{X}^* . Con esto, las ecuaciones del sistema (G_k) y la desigualdad $|\mathfrak{X}^* - \mathfrak{X}| < \frac{1}{2}\delta$ de la condición **(IVd)** se satisfacen precisamente en una elección de $\eta_1, \dots, \eta_{q-n}$, pues los vectores $\nabla \Psi_{i'_{1k}}, \dots, \nabla \Psi_{i'_{q-nk}}$, tomados en \mathfrak{X} , generan el plano de dimensión $q - n$ Ω . Notemos que en B_k , del hecho de que $\Psi_{i'_{1k}} = \dots = \Psi_{i'_{q-nk}} = 0$, se sigue que $\Psi = 0$, ya que por la condición **(IVc)**, estas $q - n$ ecuaciones son independientes, y en $\Psi = \mathbf{P}\mathbf{u}$, la matriz \mathbf{P} tiene rango $q - n$. Con todo esto, podemos definir una función $f: |\Delta(\mathfrak{M})| \rightarrow \mathcal{B}$

dada por

$$f(\mathfrak{X}^*) = \mathfrak{X}^* - (\eta_1 \nabla \Psi_{i'_{1k}}(\mathfrak{X}^*) + \cdots + \eta_{q-n} \nabla \Psi_{i'_{q-nk}}(\mathfrak{X}^*)),$$

que es continua y su inversa también goza de esta propiedad. Entonces f es un homeomorfismo y con este, podemos definir la función $g_k: \text{Int St}_{\Delta(\mathfrak{M})}(p_k) \rightarrow \mathbb{E}_n(i_k)$ como $g_k = \iota_k \circ f|_{\text{Int St}_{\Delta(\mathfrak{M})}(p_k)}$, donde $\iota_k: \mathbb{R}^q \rightarrow \mathbb{E}_n(i_k)$ es la proyección. Es inmediato verificar que g_k es inyectiva.

Como ya mencionamos, \mathcal{B} tiene una estructura diferenciable que le es inducida a partir del atlas de \mathcal{D} . Luego, usando a f , podemos inducir un C^∞ -atlas Φ_f en $|\Delta(\mathfrak{M})|$ a partir del atlas de \mathcal{B} , y si consideramos a la colección de cartas $\Phi(\mathfrak{M}) = \{(h_k, \text{Int St}_{\Delta(\mathfrak{M})}(p_k))\}_{k=1}^s$ ($h_k = \kappa_k \circ g_k$), y tomamos una carta $\phi \in \Phi_f$, entonces es sencillo ver que cada h_k es de clase C^∞ y además C^∞ -relacionado con ϕ . Así, la colección de cartas $\Phi(\mathfrak{M})$ es un C^∞ -atlas en $|\Delta(\mathfrak{M})|$, donde la triangulación $\Delta(\mathfrak{M})$ es compatible con $\Phi(\mathfrak{M})$, y así $M(\mathfrak{M})$ es difeomorfa a \hat{M} . Y si escogemos a D lo suficientemente pequeño, se cumple la condición **(IVd)**. Esto muestra que \mathfrak{M} es una presentación finita algebraica de atlas para \hat{M} .

Para concluir, nos falta ver que el concepto de presentación algebraica de atlas satisface los requerimientos **(a)**-**(c)** y **(d')**, lo cual hacemos a continuación.

(a). Es trivial.

(b). En este caso, necesitamos usar el Teorema 2.3.2. Todas las condiciones **(I)** a **(IV)** en la Definición 4.2.1 pueden ser consideradas como expresiones elementales, ya que sólo consideramos sistemas de ecuaciones y desigualdades de polinomios con coeficientes en \mathbb{Q} . En la forma en que los hemos encontrado, los polinomios de la matriz Ψ , podría tener coeficientes irracionales, obtenidos a partir de operaciones racionales, sobre los coeficientes de $\beta(\lambda)$. Pero podemos encargarnos de esta desafortunada posibilidad, si consideramos los coeficientes de $\beta(\lambda)$ (excepto el de mayor grado, que es 1) como variables adicionales, las cuales deben satisfacer la ecuación $\beta(\lambda)\gamma(\lambda) = \alpha(\lambda)$, donde también consideramos como variables adicionales a los coeficientes del factor $\gamma(\lambda)$. Específicamente, escribimos

$$\begin{aligned} \beta(\lambda) &= \lambda^n + \beta_{n-1}\lambda^{n-1} + \cdots + \beta_1\lambda + \beta_0 \\ \gamma(\lambda) &= \lambda^{q-n} + \gamma_{q-n-1}\lambda^{q-n-1} + \cdots + \gamma_1\lambda + \gamma_0 \\ \alpha(\lambda) &= \lambda^q + \alpha_{q-1}\lambda^{q-1} + \cdots + \alpha_1\lambda + \alpha_0, \end{aligned}$$

donde los α_i 's son racionales obtenidos de \mathbf{L} , y a los β_j 's y γ_k 's los consideramos como variables. Necesitamos satisfacer la ecuación $\beta(\lambda)\gamma(\lambda) = \alpha(\lambda)$ para todo valor de λ , es decir, queremos que para toda $m = 0, \dots, q - 1$ el coeficiente de λ^m en $\beta(\lambda)\gamma(\lambda)$ sea igual a α_m . Esto nos lleva a obtener q ecuaciones con coeficientes en \mathbb{Q} , en las q variables β_j, γ_k . Sabemos (Ver el párrafo que precede el enunciado de la condición **(IVc)**) que estas ecuaciones tienen soluciones reales, las cuales cumplen que $|\beta_j| < \epsilon$, ($j = 0, \dots, n - 1$). Entonces, podemos alargar el sistema de ecuaciones de la condición **(IV)** con las q ecuaciones obtenidas aquí, y las n desigualdades $|\beta_j| < \epsilon$, donde consideramos las β_j y las γ_k como variables. Por lo que todas las ecuaciones polinomiales que tenemos que considerar tienen coeficientes en \mathbb{Q} (Las derivadas parciales que también debemos considerar, son obtenidas a partir de operaciones racionales en polinomios). Con todo esto, es un trabajo de rutina el verificar que todas las condiciones que la presentación de variedad debe cumplir son expresiones elementales. Por el Teorema 2.3.2, se satisface el inciso **(b)**.

(c). Si \hat{M} y \hat{N} son dos n -variedades que tiene la misma presentación finita \mathfrak{M} , entonces ambas son difeomorfos a $|\Delta(\mathfrak{M})|$, así que \hat{M} y \hat{N} también son difeomorfos, es decir, son esencialmente la misma n -variedad, por lo que en efecto \mathfrak{M} presenta una única variedad (salvo difeomorfismo).

(d'). Para calcular el atlas $\Phi(\mathfrak{M})$ con cierta precisión dada, tenemos que resolver sistemas de ecuaciones del tipo considerado en la demostración del inciso **(b)**, con una tolerancia de error dada τ . Si las componentes (racionales) de un punto $p \in \text{Int St}_{\Delta(\mathfrak{M})}(p_k)$ son dadas, sabemos que la imagen de p , $h_k(p)$, yace en una vecindad N , de $\kappa_k \circ \iota_k(p)$, en \mathbb{R}^n . Podemos subdividir una vecindad de N en hipercubos con aristas (paralelas a los ejes coordenados) de longitud τ y todos tienen como vértice común al punto $\kappa_k \circ \iota_k(p)$. Si formamos la expresión elemental que describe al punto $h_k(p)$ (usando la expresión que tenemos para g_k) y combinándola con las demás expresiones que tenemos, podemos usar el Teorema 2.3.2 para determinar el cubo que contiene a $h_k(p)$, y tomar su punto medio como una solución aproximada $h_k^\#(p)$. Esto muestra que la condición **(d')** se satisface y así obtenemos el teorema. □

Con esta demostración, terminamos el objetivo principal de este capítulo, y todos los problemas planteados en el capítulo 3 quedan bien definidos. El concepto de presentación finita de variedad es “bastante complicado”, tiene

muchos elementos que deben cumplir muchas condiciones complicadas. Esta es quizá una de las razones por las que, fuera de los resultados que daremos en el capítulo 5, no se tienen otros que se obtengan usando las presentaciones de variedades. Sería interesante encontrar una forma más “compacta” de representar variedades C^∞ -PL (cerradas), lo cual puede abrir la puerta para que mas investigadores se interesen en estudiar estas presentaciones (tanto topólogos, como computólogos).

Capítulo 5

Insolubilidad del problema de equivalencia entre variedades

Estamos entrando en la recta final de nuestro camino para probar la imposibilidad algorítmica (en general) de los problemas de clasificación de variedades (bajo difeomorfismo, homeomorfismo, equivalencia combinatoria y homotópica). Para poder lograrlo, combinaremos todo lo desarrollado en los capítulos anteriores y mostraremos que para cada dimensión $n \geq 4$, existe una función de palabras computable, F_n , que recibe como entrada, presentaciones finitas de grupos, y nos da como salida, presentaciones finitas de variedades C^∞ -PL, que gozan de la propiedad de que, cualesquiera dos variedades presentadas, son equivalentes si y sólo si, sus grupo fundamentales y los segundos números de Betti son los mismos. Así, F_n se volverá una pieza clave para poder construir reducciones de Turing entre los problemas de isomorfismo de grupos y x -equivalencia¹, y los problemas de decidibilidad de las propiedades de ser el grupo trivial y de ser simplemente conexa. Ya que podemos escoger los problemas de grupos mencionados, de forma que sean tan difíciles como queramos, podremos hacer lo mismo con los problemas topológicos que enfrentamos. Por supuesto, en todo esto, hay muchos detalles que aún debemos llenar.

¹Recordemos que en el Capítulo 3, en la Definición 3.3.20, definimos los problemas de x -equivalencia ($x = d, t, c, h$) relacionados con las clasificaciones de variedades bajo difeomorfismo, homeomorfismo, equivalencia combinatoria y homotópica respectivamente.

5.1. Vecindades de asa

El propósito de esta sección, es definir el concepto de vecindad de asa (rectilínea normada) de un complejo simplicial de dimensión a lo más 2. Estas vecindades, son definidas de tal manera, que se amoldan al complejo dado, como si estuvieran “engrosando” sus vértices, aristas y triángulos. A partir de esto, definiremos un $(n + 1)$ -complejo simplicial, que aproxima una de estas vecindades de asa. La ventaja de esta aproximación, es que podremos hacer manipulaciones computacionales, lo cual es pieza clave en la construcción de la función computable F_n .

Definición 5.1.1. Sea K_2 un complejo simplicial rectilíneo de dimensión a lo más 2 en \mathbb{R}^{n+1} , con vértices racionales, $n \geq 4$; sean P^i, K^j, T^k ($i = 1, \dots, u_0; j = 1, \dots, u_1; k = 1, \dots, u_2$) los vértices, aristas, y triángulos, respectivamente, de K_2 . Definimos la vecindad de asa esférica de K_2 en \mathbb{R}^{n+1} , con radios $\rho_0 > \rho_1 > \rho_2$, como la unión N de $(n + 1)$ -bolas $P_{n+1}^i, K_{n+1}^j, T_{n+1}^k$, con las siguientes propiedades, para todos i, j, k :

- (P) P_{n+1}^i es la $(n + 1)$ -bola en \mathbb{R}^{n+1} con radio ρ_0 y centro P^i .
- (K) K_{n+1}^j es el $(n + 1)$ -cilindro en $\mathbb{R}^{n+1} - \bigcup_{i=1}^{u_0} \text{Int } P_{n+1}^i$ con radio ρ_1 y eje K^j (es decir, el conjunto de todos los puntos en $\mathbb{R}^{n+1} - \bigcup_{i=1}^{u_0} \text{Int } P_{n+1}^i$ tales que su distancia a K^j es menor o igual a ρ_1); ρ_1 debe ser tan pequeño, que los K_{n+1}^j 's son disjuntos a pares.
- (T) T_{n+1}^k es el $(n + 1)$ -cilindro en $\mathbb{R}^{n+1} - \text{Int}(\bigcup_{i=1}^{u_0} P_{n+1}^i \cup \bigcup_{j=1}^{u_1} K_{n+1}^j)$ con radio ρ_2 y eje T^k ; ρ_2 debe ser tan pequeño que los T_{n+1}^k 's son disjuntos a pares.

Y más aún, definimos la vecindad de asa rectilínea normada de K_2 en \mathbb{R}^{n+1} , como un complejo rectilíneo N^* de dimensión $n + 1$, con vértices racionales, que contiene subcomplejos

$$P_{n+1}^{*i}, K_{n+1}^{*j}, T_{n+1}^{*k}, \quad \text{con } |N^*| = \bigcup_{i,j,k=1}^{u_0, u_1, u_2} |P_{n+1}^{*i}| \cup |K_{n+1}^{*j}| \cup |T_{n+1}^{*k}|,$$

con las siguientes propiedades:

- (δ^*) Existen números racionales positivos, ρ_0, ρ_1, ρ_2 , y δ^* , $2\delta^* < \rho_2 < \rho_1 < \rho_0$, tal que $|N^*| \subset N$, donde N es la vecindad de asa esférica de K_2 en \mathbb{R}^{n+1} , con radios ρ_0, ρ_1, ρ_2 ; y (con la notación de la primera definición) cada punto de $\partial P_{n+1}^{*i} \cap \partial N^*$, $\partial K_{n+1}^{*j} \cap \partial N^*$, $\partial T_{n+1}^{*k} \cap \partial N^*$ esta mas cerca que δ^* a un punto en $\partial P_{n+1}^i \cap \partial N$, $\partial K_{n+1}^j \cap \partial N$, $\partial T_{n+1}^k \cap \partial N$, respectivamente.
- (**P***) P_{n+1}^{*i} es una estrella cerrada de P^i en \mathbb{R}^{n+1} .
- (**K***) K_{n+1}^{*j} es una estrella cerrada vecindad de

$$\tilde{K}^j = K^j \cap (\mathbb{R}^{n+1} - \bigcup_{i=1}^{u_0} \text{Int} |P_{n+1}^{*i}|), \quad \text{en } \mathbb{R}^{n+1} - \bigcup_{i=1}^{u_0} \text{Int} |P_{n+1}^{*i}|.$$

- (**T***) T_{n+1}^{*k} es una estrella cerrada vecindad de

$$\begin{aligned} \tilde{T}^k &= T^k \cap (\mathbb{R}^{n+1} - \text{Int}(\bigcup_{i=1}^{u_0} |P_{n+1}^{*i}| \cup \bigcup_{j=1}^{u_1} |K_{n+1}^{*j}|)), \\ &\text{en } \mathbb{R}^{n+1} - \text{Int}(\bigcup_{i=1}^{u_0} |P_{n+1}^{*i}| \cup \bigcup_{j=1}^{u_1} |K_{n+1}^{*j}|). \end{aligned}$$

- (σ) Si σ es un n -simplejo de $\partial P_{n+1}^{*i} \cap \partial N^*$ y l es la línea en \mathbb{R}^{n+1} que es normal a σ y contiene el punto medio de σ , entonces l contiene a P^i ; similarmente, si σ esta en $\partial K_{n+1}^{*j} \cap \partial N^*$, entonces l interseca a \tilde{K}^j en un punto; finalmente, si σ esta en $\partial T_{n+1}^{*k} \cap \partial N^*$, entonces l interseca a \tilde{T}^k en un punto.

Así pues, una vecindad de asa rectilínea normada, es un complejo aproximando una vecindad de asa esférica. En las figuras² 5.1 y 5.2, podemos ver ejemplos de vecindades de asa para el 1-simplejo estándar y un complejo simplicial de dimensión 1 (una cuña de un triangulo hueco con dos 1-simplejos), asumiendo que tomamos las vecindades dentro de \mathbb{R}^3 ($n + 1 = 3$ en la Definición 5.1.1). Las partes de las figuras en donde se dibujan a las vecindades de asa con líneas y cuadrados, no representan una triangulación

²Estas imagenes fueron generadas con 3DStudio MAX y son cortesía de Juan Emmanuel Barrera García.

de las vecindades, su propósito es mostrar como se ve el complejo al que están cubriendo. Los resultados siguientes son dos lemas que necesitaremos más adelante. El primero, es un resultado de computabilidad; el segundo nos asegura la existencia de presentaciones finitas de variedades, tales que las variedades presentadas, son complejos frontera de vecindades de asa rectilíneas normadas, de complejos simpliciales de dimensión a lo más 2.

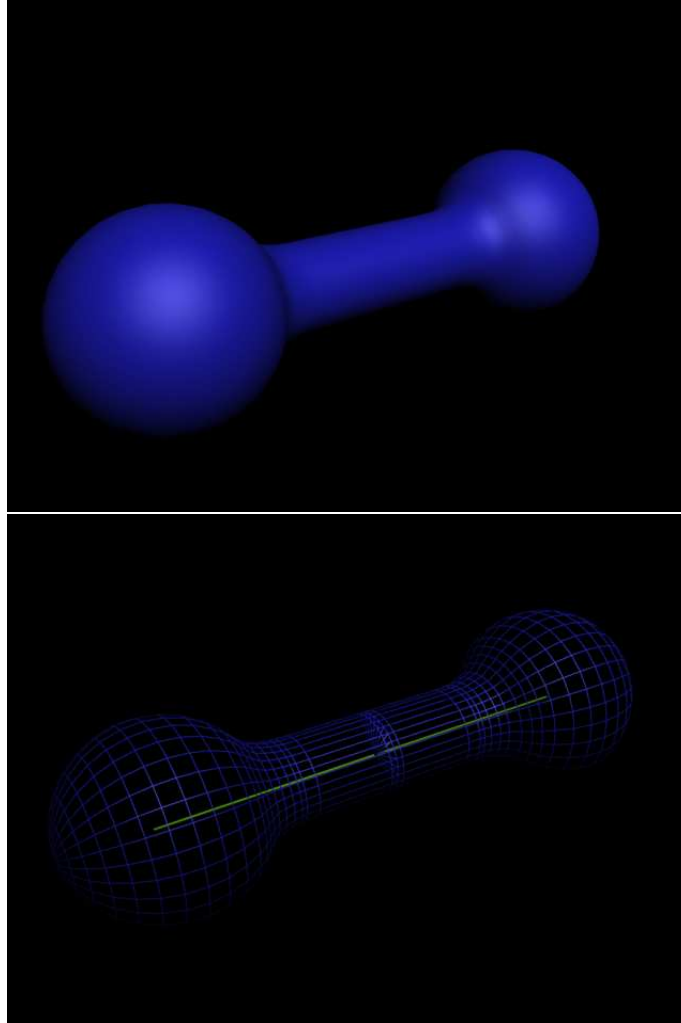


Figura 5.1: Vecindad de asa del 1-simplejo estándar (la línea verde).

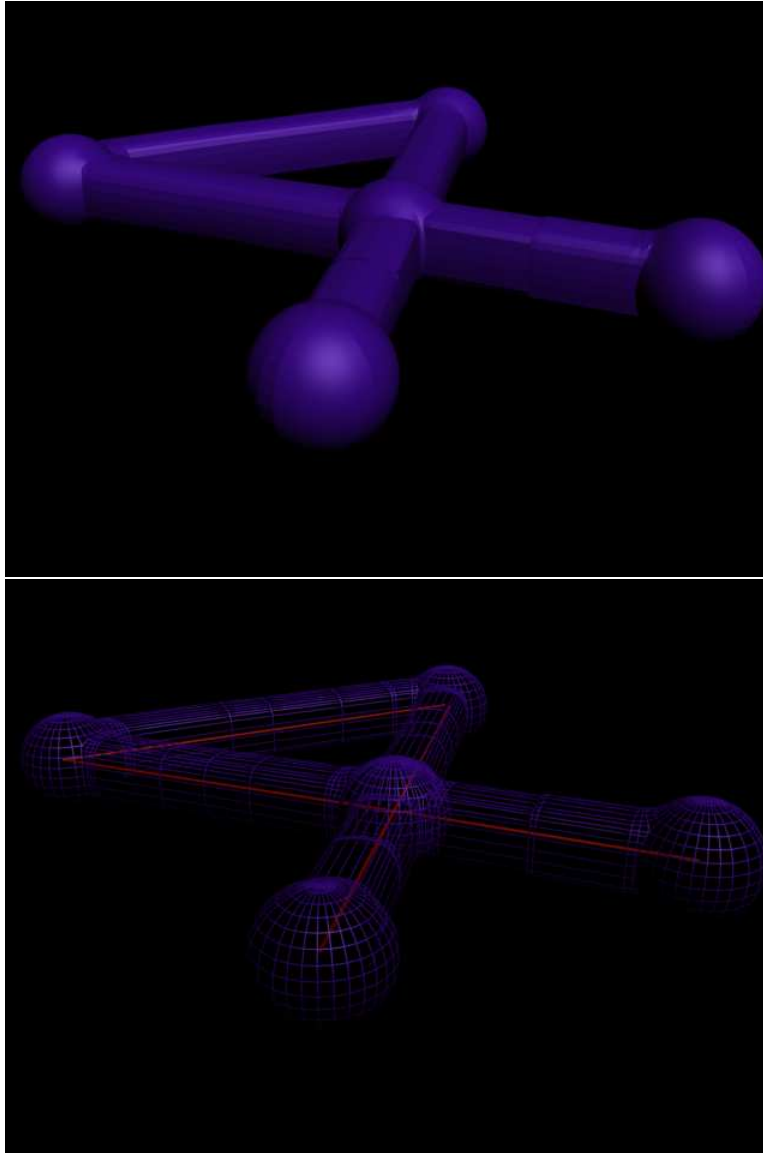


Figura 5.2: Vecindad de asa de un complejo simplicial, indicado por las líneas de color rojo.

Lema 5.1.2. *Existe un algoritmo para decidir, dado un complejo simplicial rectilíneo de dimensión n , $\theta_n \subseteq \mathbb{R}^{n+1}$ con vértices racionales, $n \geq 4$,*

i) Si θ_n es el complejo frontera de una vecindad de asa rectilínea normada

de algún complejo K_2 (rectilíneo, de dimensión a lo más 2) en \mathbb{R}^{n+1} .

ii) Si la respuesta a i) es afirmativa, determinar todos los complejos K_2 en \mathbb{R}^{n+1} , de los cuales θ_n es frontera de una vecindad de asa.

Demostración. Dado un n -complejo $\theta_n \subseteq \mathbb{R}^{n+1}$, examinamos todos los posibles subcomplejos de dimensión n de θ_n , y determinamos si pueden ser expresados como conjuntos de la forma

$$\{\partial P_{n+1}^{*i} \cap \partial N^*, \partial K_{n+1}^{*j} \cap \partial N^*, \partial I_{n+1}^{*k} \cap \partial N^* \mid i = 1, \dots, u_0; j = 1, \dots, u_1; \\ k = 1, \dots, u_2\},$$

(usamos la misma notación que en la Definición 5.1.1). Esto lo podemos calcular, ya que las propiedades de intersección de las líneas normales a través de los centros de los n -simplejos puede ser decidida (omitimos los detalles por simplicidad). Siempre que encontremos un conjunto de subcomplejos con estas propiedades, el 2-complejo K_2 puede ser determinado. (Los vértices de K_2 son las intersecciones de las líneas normales a través de los centros de los n -simplejos de los “conjuntos” $\partial P_{n+1}^{*i} \cap \partial N^*$, $i = 1, \dots, u_0$; etc.) Esto completa la demostración. \square

Lema 5.1.3. *Sea K_2 un complejo simplicial rectilíneo de dimensión a lo más 2 en \mathbb{R}^{n+1} , con vértices racionales, $n \geq 4$. Entonces existe una presentación algebraica de atlas \mathfrak{M} , de una n -variedad diferenciable, tal que $\mathbb{E}_q(\mathfrak{M}) = \mathbb{R}^{n+1}$ y $\Delta(\mathfrak{M})$ es el complejo frontera de una vecindad de asa rectilínea normada N^* de K_2 en \mathbb{R}^{n+1} .*

Demostración. Comencemos con una vecindad de asa esférica N de K_2 en \mathbb{R}^{n+1} . Entonces, para algún $\delta^* \in \mathbb{Q}^+$ suficientemente pequeño, existe una vecindad de asa rectilínea normada N^* de K_2 en \mathbb{R}^{n+1} que aproxima a N de la forma descrita en la Definición 5.1.1. Por otro lado, sabemos también que existe una n -variedad diferenciable \mathcal{V} , encajada en \mathbb{R}^{n+1} , que aproxima a ∂N , es decir, \mathcal{V} es tal que para algún $\delta^\# > 0$ y para cada $p \in \mathcal{V}$, la vecindad B de radio $\delta^\#$ alrededor de p en la línea normal a \mathcal{V} a través de p interseca a ∂N en un sólo punto, $q(p)$, de tal forma que la función $\varphi: \mathcal{V} \rightarrow \partial N$ dada por $\varphi(p) = q(p)$, es un homeomorfismo. La variedad \mathcal{V} puede ser obtenida de ∂N si suavizamos las esquinas en las cuales las fronteras de las regiones

cilíndricas (o esféricas) $\partial P_{n+1}^{*i} \cap \partial N$, $\partial K_{n+1}^{*j} \cap \partial N$, $\partial T_{n+1}^{*k} \cap \partial N$ se juntan; para ver ejemplos de esto, puede consultarse a Cairns [Ca61].

Por Whitney [Hi76] existe una n -variedad analítica $\mathcal{D} \subseteq \mathbb{R}^{n+1}$ que aproxima a \mathcal{V} . Desde este punto, la construcción de la presentación algebraica de atlas \mathfrak{M} , la realizamos de la misma forma que en la demostración de la primera afirmación del Teorema 4.2.2, con $q = n + 1$, y tomamos como el complejo representado por \mathfrak{M} , $\Delta(\mathfrak{M})$, al complejo ∂N^* , lo cual podemos hacerlo, siempre que hayamos escogido a N^* lo suficientemente fino (es decir, subdividido) y que todas las aproximaciones dadas sean lo suficiente cercanas. Con esto aseguramos que las vecindades de radio $\frac{1}{2}\delta$ de los puntos de \mathcal{B} (La componente de la variedad algebraica de la demostración del Teorema 4.2.2) en las líneas normales, \mathcal{Q} a \mathcal{B} , estarán lo suficientemente cercanas a los intervalos normales, \mathfrak{q} a \mathcal{V} , y como consecuencia, nos darán correspondencias uno a uno entre los puntos de \mathcal{B} y los de ∂N y también con los de ∂N^* . Así, ∂N^* quedará dentro de la vecindad \mathcal{N}^* de \mathcal{B} , como en la demostración del Teorema 4.2.2. Esto concluye la demostración. \square

5.2. Construcción de F_n

Usando los Lemas 3.3.17, 3.3.18, 5.1.2 y 5.1.3, probaremos que existe una función computable (en el sentido de la Definición 1.2.7) que asocia a cada presentación finita de grupo μ , una presentación algebraica de atlas \mathfrak{M} , de una n -variedad diferenciable cerrada tal que

- (i) El espacio euclidiano presentado por \mathfrak{M} tiene dimensión $n + 1$;
- (ii) $\Delta(\mathfrak{M})$ es el complejo frontera de una vecindad de asa rectilínea normada N^* , de un complejo rectilíneo de dimensión a lo más 2, digamos K_2 , en \mathbb{R}^{n+1} ;
- (iii) Los grupos $\pi_1(|K_2|)$ y $G(\mu)$ son isomorfos.

Lema 5.2.1. *Sea $n \geq 4$. Existe una función computable F_n , la cual asocia a cada presentación finita de grupo μ , una presentación algebraica (finita) de atlas, correspondiente a una n -variedad diferenciable cerrada (con una estructura combinatoria compatible), $F_n(\mu)$ tal que*

- i) La función multivaluada F_n^{-1} es computable, en el sentido de que existe un algoritmo para determinar, dada una presentación algebraica de atlas \mathfrak{M} : (a) Si \mathfrak{M} pertenece o no al rango de F_n y (b) Si la respuesta a (a) es afirmativa, determinar todas las clases de congruencia en $F_n^{-1}(\mathfrak{M})$.
- ii) $G(\mu) \cong \pi_1(F_n(\mu))$.
- iii) Obtenemos las siguientes ecuaciones para el segundo número de Betti:

$$\beta_2(F_n(\mu * 1)) = \begin{cases} \beta_2(F_n(\mu)) + 2 & \text{si } n = 4, \\ \beta_2(F_n(\mu)) + 1 & \text{si } n > 4. \end{cases}$$

- iv) Si la presentación μ' puede derivarse de la presentación μ , por una operación elemental $Op_1^{\pm 1}, Op_2, Op_3, Op_4, Op_5^{\pm 1}$, entonces $F_n(\mu) \approx_x F_n(\mu')$ para $x = d, t, c, h$.

Demostración. Primero hacemos notar el hecho de que dado $n \geq 4$, existe una máquina enumeradora E , que enlista todas las presentaciones algebraicas de atlas de n -variedades diferenciables cerradas, que tienen a x_1, \dots, x_{n+1} como variables coordenadas. Si μ es una presentación finita de grupo, definimos a F_n como la función computada por la MT N_n , es decir, $F_n(\mu) = N_n(\mu)$; donde N_n ejecuta el siguiente programa:

$N_n =$ "En la entrada μ :

1. Verifica que μ es una presentación de grupo válida; si no lo es, borra todo el contenido de la cinta, escribe la palabra *EINVAL* y entra en el estado de *aceptación*.
2. Simula a la máquina E hasta que enumere a la siguiente presentación \mathfrak{M}_m . Guarda el estado actual de E .
 - 2.1. Verifica si $\Delta(\mathfrak{M}_m)$ es el complejo frontera de una vecindad de asa rectilínea N^* . En caso afirmativo, genera la lista $S_2 = \{K_{21}, \dots, K_{2p}\}$ de todos los complejos de dimensión a lo más 2 de los cuales N^* es vecindad; en caso contrario, ve al paso 2.3.
 - 2.2. Para cada complejo $K_2 \in S_2$:
 - 2.2.1. Genera representantes ν_1, \dots, ν_r de las clases de congruencia de presentaciones finitas de grupos

tales que $G(\nu_i) \cong \pi_1(|K_2|)$, con $1 \leq i \leq r$.

2.2.2. Verifica si μ es congruente con alguna de las ν_i 's. En caso de que lo sea, ve al paso **3**; en caso contrario, continua con el siguiente complejo en S_2 .

2.3. Si alguna de las propiedades verificadas en los pasos **2.1** y **2.2.** fallaron, ve al paso **2**.

3. Escribe en la cinta la palabra \mathfrak{M}_m .

4. Entra en el estado de *aceptación*."

La idea detras de la máquina N_n , es ir revisando una por una, todas las presentaciones finitas de n -variedades diferenciables cerradas, buscando una que satisfaga los incisos (i), (ii) y (iii) del párrafo al inicio de esta sección. Por los Lemas 3.3.18 y 5.1.3, $N_n(\mu)$ siempre existe para cualquier presentación de grupo μ . Además, todos los pasos del programa de N_n son realizables en tiempo finito, ya que podemos usar los algoritmos dados por los Lemas 3.3.17 y 5.1.2 para que N_n lleve a cabo los ciclos indicados en los pasos **2** y **2.2**. Con todo esto, tenemos que N_n siempre se detiene en cualquier entrada, por lo que la función que calcula es computable. Sea pues, $F_n(\mu) = N_n(\mu)$. Debemos mostrar que F_n cumple con las condiciones **i**)-**iv**):

i) El algoritmo que requerimos para la relación F_n^{-1} está dado por la siguiente³ MT N_n^{-1} :

N_n^{-1} = "En la entrada \mathfrak{M} :

1. Verifica que \mathfrak{M} es una presentación de variedad válida; si no lo es, ve al paso **7**.

2. Verifica que las variables coordenadas x_1, \dots, x_q de \mathfrak{M} sean precisamente $n + 1$. En caso contrario, ve al paso **7**.

3. Verifica si $\Delta(\mathfrak{M})$ es el complejo frontera de una vecindad de asa rectilínea N^* . En caso afirmativo, genera la lista $S_2 = \{K_{21}, \dots, K_{2p}\}$ de todos los complejos de dimensión a lo más 2 de los cuales N^* es vecindad; en caso contrario, ve al paso **7**.

4. Escoge un complejo $K'_2 \in S_2$ y genera la lista $L_2 = \{\nu_1, \dots, \nu_r\}$ de las clases de congruencia de presentaciones finitas de grupos tales que $G(\nu_i) \cong \pi_1(|K'_2|)$, con $1 \leq i \leq r$.

³En el programa de N_n^{-1} , la expresión " $L_2 = L_2 \cup \{\mu\}$ " de la línea **5.2.1** no debe ser tomada como una expresión matemática. Es usual entre programadores al especificar algoritmos, el denotar de esta forma la instrucción "Agregar el objeto μ a L_2 ".

5. Para cada complejo $K_2 \in S_2 - \{K'_2\}$:
 - 5.1. Genera la lista de representantes $O_2 = \{\mu_1, \dots, \mu_t\}$ de las clases de congruencia de presentaciones finitas de grupos tales que $G(\mu_j) \cong \pi_1(|K_2|)$, con $1 \leq j \leq t$.
 - 5.2. Para cada presentación $\mu \in O_2$:
 - 5.2.1. Si μ no es congruente con alguna de las presentaciones en L_2 , entonces $L_2 = L_2 \cup \{\mu\}$.
 - 5.2.2. Continúa con la siguiente presentación en O_2 .
 - 5.3. Continúa con el siguiente complejo en $S_2 - \{K'_2\}$.
6. Escribe la lista L_2 en la cinta y entra en el estado de *aceptación*.
7. Borra todo el contenido de la cinta, escribe la palabra *ERROR* y entra en el estado de *rechazo*.”

Primero probamos que N_n^{-1} es una MT total. Dada una entrada \mathfrak{M} , los pasos **1** y **2** son fácilmente realizables en tiempo finito. El paso **3** también se puede hacer en tiempo finito, por la existencia del algoritmo dado en el Lema 5.1.2; lo mismo sucede para el paso **4** usando el Lema 3.3.17. Los pasos internos especificados en **5** son realizables por N_n^{-1} también por el Lema 3.3.17, y los ciclos dados en **5** y **5.2** se ejecutan solamente un número finito de veces, pues las listas S_2 , L_2 y O_2 siempre son de cardinal finito. Los pasos **6** y **7** son fácilmente llevados a cabo, por lo que N_n^{-1} es una MT total.

Ahora veamos que el algoritmo de N_n^{-1} es correcto. Sea \mathfrak{M} una posible entrada para N_n^{-1} y supongamos primero que \mathfrak{M} no pertenece al rango de F_n . Esto sólo puede ocurrir si se da alguno de los siguientes casos:

- (a) \mathfrak{M} no es una presentación finita de n -variedad válida.
- (b) El número de variables coordenadas de \mathfrak{M} es distinto a $n + 1$.
- (c) $\Delta(\mathfrak{M})$ no es el complejo frontera de una vecindad de asa rectilínea normada.

El caso (a) es detectado por N_n^{-1} en el paso **1**, que llevará a la máquina al paso **7**, con lo cual rechazará; si ocurre (b), la verificación del paso **2** fallará, lo que provocará que N_n^{-1} rechace; finalmente, el caso (c) es detectado en el paso **3** con lo que se ejecutará **7** y por lo tanto, N_n^{-1} rechaza. Si \mathfrak{M} está en el rango de F_n , entonces las verificaciones de los pasos **1**, **2** y **3** serán exitosas, y N_n^{-1} procederá a generar la lista de todos los complejos simpliciales de dimensión a lo más 2 relacionados con la vecindad de asa de la cual $M(\mathfrak{M})$ es

frontera; luego, N_n^{-1} ejecutará el código de los pasos **4** a **6**, terminando con la lista L_2 de los representantes de las clases de congruencia de presentaciones de grupos tales que su imagen bajo F_n es precisamente \mathfrak{M} . Concluimos que el algoritmo ejecutado por N_n^{-1} es correcto, cumpliendo con lo requerido por **i**).

ii) $\Delta(F_n(\mu))$ es el complejo frontera de una vecindad de asa rectilínea normada N^* de un 2-complejo K_2 , tal que $\pi_1(|K_2|) \cong G(\mu)$. Ya que $n \geq 4$, tenemos que $\pi_1(|\Delta(F_n(\mu))|) \cong \pi_1(|K_2|)$ debido al siguiente argumento: Toda trayectoria cerrada C en K_2 la podemos deformar adentro de N^* en una trayectoria en $\partial N^* = \Delta(F_n(\mu))$ (Con una pequeña deformación, transformamos a C en una curva en N^* , separada de K_2 ; y esta curva la movemos dentro de ∂N^*), y viceversa. Similarmente, si la trayectoria C acota un disco singular (una imagen continua de un disco), digamos D , en K_2 , entonces la trayectoria deformada en ∂N^* acota un disco singular (obtenido a partir de D , por deformación) en ∂N^* , y viceversa. Esto concluye el argumento.

Para probar **iii**) y **iv**) necesitamos los siguientes resultados:

Lema 5.2.2. *La vecindad de asa rectilínea normada N^* de K_2 , con $\pi_1(|K_2|) \cong G(\mu)$ y $\Delta(F_n(\mu)) = \partial N^*$ puede ser obtenida a partir de un $(n+1)$ -disco agregando (de manera combinatoria) $r(\mu)$ asas de grado 1 y $p(\mu)$ asas de grado 2. Más aún, la estructura diferenciable de $M(F_n(\mu))$ puede ser extendida a una estructura diferenciable de N^* .*

Demostración. Recordemos en este momento el proceso (dado por el Corolario 3.3.15) por medio del cual obtenemos, a partir de una presentación de grupo μ , un 2-complejo conexo K_2 . Esencialmente, construimos primero una cuña B , de $r(\mu)$ triángulos huecos, con vértice común w , después construimos los $p(\mu)$ polígonos triangulados $D(r_j)$ (uno por cada relación r_j de μ) y los pegamos en los triángulos huecos de B de acuerdo a los generadores que tengan cada r_j (ver el ejemplo del Capítulo 3 y la figura 3.2).

En detalle, podemos obtener a N^* de la siguiente manera: Comenzamos con la $(n+1)$ -bola D_0^{n+1} con centro en el vértice w ; entonces agregamos asas de grado 1, $D_1^{n+1}, \dots, D_{r(\mu)}^{n+1}$, usando los encajes

$$\psi_i: \mathbb{S}_i^0 \times D_i^n \rightarrow \text{Int } \mathbb{S}_0^n \quad (i = 1, \dots, r(\mu)).$$

Este proceso de agregado de 1-asas nos da como resultado una variedad $M' = \chi(D_0^{n+1}, \mathbb{S}^n; \psi_1, \dots, \psi_{r(\mu)}, 1)$, que cubre al vértice w con la $(n+1)$ -bola

D_0^{n+1} y cada 1-asa D_i^{n+1} ($1 \leq i \leq r(\mu)$) cubre un triángulo hueco de B que corresponde al generador i -ésimo de μ . Es decir, hemos construido y pegado los subcomplejos P_{n+1}^* 's y K_{n+1}^* 's de N^* , que cubren los vértices y aristas de K_2 . A continuación, agregamos a M' asas de grado 2, $D_1^{n+1}, \dots, D_{p(\mu)}^{n+1}$, con encajes

$$\varphi_j: \mathbb{S}_j^1 \times D_j^{n-1} \rightarrow \text{Int } \partial M' \quad (j = 1, \dots, p(\mu)),$$

tal que con φ_j pegamos a D_j^{n+1} en los huecos de M' en donde se encuentran los triángulos de K_2 , dados por los polígonos triangulados $D(r_j)$. Con este proceso de agregado de 2-asas obtenemos una variedad $M = \chi(M', \partial M'; \varphi_1, \dots, \varphi_{p(\mu)}, 2)$, de tal forma que hemos construido y pegado los subcomplejos T_{n+1}^* 's de N^* que cubren los triángulos de K_2 . Así, M es precisamente la vecindad de asa N^* . Esto demuestra el primer enunciado del lema.

La estructura diferenciable de la variedad \mathcal{B} , considerada en la prueba del Lema 5.1.3, la cual es difeomorfa a $M(F_n(\mu))$, y está encajada diferenciablemente en \mathbb{R}^{n+1} , es representada por un atlas que obtenemos al proyectar ciertas estrellas de \mathcal{B} en planos coordenadas de dimensión n de \mathbb{R}^{n+1} . Sea $\tilde{N} \subseteq \mathbb{R}^{n+1}$ la $(n+1)$ -variedad de la cual \mathcal{B} es frontera. Por lo tanto, \tilde{N} aproxima a N^* . Las proyecciones de las estrellas de \mathcal{B} , pueden ser extendidas a difeomorfismos de $(n+1)$ -bolas en \tilde{N} sobre hemiespacios coordenados de \mathbb{R}^{n+1} . Como resultado de esto, obtenemos un atlas de una vecindad de \mathcal{B} en \tilde{N} . Este atlas puede ser extendido a uno de \tilde{N} , usando las funciones identidad en las $(n+1)$ -bolas del interior de \tilde{N} .

Existe un homeomorfismo $\varphi: \tilde{N} \rightarrow N^*$, el cual es igual a la identidad afuera de una vecindad de \mathcal{B} , y además $\varphi|_{\mathcal{B}}$ transforma la estructura diferenciable de \mathcal{B} en la de $M(F_n(\mu))$. Por esto, φ transforma la estructura diferenciable de \tilde{N} en una estructura diferenciable de N^* , que extiende a la de $M(F_n(\mu))$, y con esto terminamos la demostración. □

Lema 5.2.3. *Si $\Delta(F_n(\mu)) = \partial N^*$ y $\Delta(F_n(\mu')) = \partial N^{*'}$, donde N^* y $N^{*'}$, son vecindades de asa rectilíneas normadas de 2-complejos K_2 y K_2' , respectivamente, y si N^* y $N^{*'}$ son equivalentes combinatoriamente, entonces*

$$M(F_n(\mu)) \approx_x M(F_n(\mu')) \quad (x = d, t, c, h).$$

Demostración. Es inmediato para los casos de homeomorfismo, equivalencia combinatoria y homotópica. El caso de difeomorfismo se sigue del Teorema 6.5 de Munkres [Mu60], el cual nos dice en particular, que la equivalencia combinatoria implica difeomorfismo, si los grupos de homología $H_m(N^*)$ y $H_m(N^{*'})$, son triviales para $m \geq 3$. Y en efecto, N^* y $N^{*'}$ tienen homologías triviales para $m \geq 3$, pues son vecindades de complejos de dimensión a lo más 2 (y por lo tanto, del mismo tipo de homotopía que complejos de dimensión a lo más 2). □

iii) Sea $\Delta(F_n(\mu)) = \partial N^*$, entonces $\Delta(F_n(\mu * 1))$ es la frontera de una vecindad, digamos $N^{*'}$, de un 2-complejo K'_2 , que tiene una realización homeomorfa a la unión de $|K_2|$ con un disco abierto $D_{p(\mu)+1}$, con frontera el vértice w , tal que $|K_2| \cap D_{p(\mu)+1} = \emptyset$. Podemos escoger al disco abierto $D_{p(\mu)+1}$ en \mathbb{R}^{n+1} , de tal forma que intersecte al cerrado $\mathbb{R}^{n+1} - \text{Int } N^*$ en un disco, digamos D . Sea A_{n+1} una vecindad (triangulada) de D en $\mathbb{R}^{n+1} - \text{Int } N^*$. Entonces la variedad M , obtenida al pegar a N^* la 2-asa A_{n+1} , es equivalente combinatoriamente a $N^{*'}$. Así,

$$\partial M = (\partial N^* - \text{Int}(\partial N^* \cap \partial A_{n+1})) \cup \partial(A_{n+1} - \partial N^* \cap \partial A_{n+1}).$$

(El complejo ∂M se obtiene con cirugía de Morse a partir de ∂N^*). En esta ecuación, $\partial N^* \cap \partial A_{n+1}$ es homeomorfo a $\mathbb{S}^1 \times D^{n-1}$ y $\partial A_{n+1} - \text{Int}(\partial N^* \cap \partial A_{n+1})$ es homeomorfo a $D^2 \times \mathbb{S}^{n-2}$; Más aún, $\partial N^* \cap \partial A_{n+1}$ esta dentro de una n -bola en ∂N^* . Ahora, tenemos que

$$\beta_2(\partial N^* - \text{Int}(\partial N^* \cap \partial A_{n+1})) = \begin{cases} \beta_2(\partial N^*) + 1 & \text{si } n = 4, \\ \beta_2(\partial N^*) & \text{si } n > 4. \end{cases}$$

Y si agregamos $\partial A_{n+1} - \partial N^* \cap \partial A_{n+1}$, β_2 se incrementa en 1 en ambos casos. Con un poco de esfuerzo, es posible verificar todo esto con los métodos para calcular grupos de homología de los que hablamos en el Capítulo 3, que son descritos en [Mu00, Ro88]. Esto concluye **iii**).

iv) Sean $\Delta(F_n(\mu)) = \partial N^*$ y $\Delta(F_n(\mu')) = \partial N^{*'}$, donde N^* , $N^{*'}$ son vecindades de asa rectilíneas normadas de K_2 y K'_2 , tales que $\pi_1(|K_2|) \cong G(\mu)$ y $\pi_1(|K'_2|) \cong G(\mu')$, respectivamente. Por el Lema 5.2.3, es suficiente demostrar que N^* y $N^{*'}$ son equivalentes combinatoriamente.

Caso 1. La presentación μ' se obtiene de μ por medio de las operaciones Op_2 o Op_3 . Entonces K_2 y K'_2 son equivalentes combinatoriamente, pues la única diferencia entre las presentaciones μ y μ' es el orden de los símbolos en una de sus relaciones, por lo que el algoritmo que construye un 2-complejo a partir de una presentación de grupo (dado en el Corolario 3.3.15), nos da esencialmente el mismo resultado al aplicarlo a μ y μ' . Ya que K_2 y K'_2 son equivalentes combinatoriamente, también lo son sus vecindades N^* y $N^{*'}$.

Caso 2. La presentación μ' se obtiene de μ por medio de las operaciones $Op_1^{\pm 1}$ o Op_4 , digamos, al reemplazar α_i por α'_i . Sea μ'' la presentación de grupo que obtenemos de μ al borrar α_i . Sea K''_2 un 2-complejo tal que $\pi_1(|K''_2|) \cong G(\mu'')$ y sea $N^{*''}$ una vecindad de asa rectilínea normada de K''_2 en \mathbb{R}^{n+1} . Entonces podemos obtener un complejo con realización homeomorfa a $|K_2|$ ($|K'_2|$), al agregar a K''_2 un 2-disco abierto D_i (D'_i) que corresponde a α_i (α'_i , es decir, D_i y D'_i son los interiores de las realizaciones de los polígonos triangulados $D(\alpha_i)$ y $D(\alpha'_i)$ respectivamente). Escogemos a D_i y D'_i en \mathbb{R}^{n+1} de tal manera que $D_i \cap (\mathbb{R}^{n+1} - \text{Int } N^{*''})$ y $D'_i \cap (\mathbb{R}^{n+1} - \text{Int } N^{*''})$ sean 2-discos, digamos D y D' , con la condición de que $\partial D \cap \partial D' = \emptyset$. Sean A_{n+1} y A'_{n+1} $(n+1)$ -bolas, vecindades (trianguladas) de D y D' respectivamente, en $\mathbb{R}^{n+1} - \text{Int } N^{*''}$. Entonces la variedad M , obtenida al pegar a $N^{*''}$ la 2-asa A_{n+1} y la variedad M' , obtenida al pegar a $N^{*''}$ la 2-asa A'_{n+1} , son equivalentes combinatoriamente a N^* y $N^{*'}$ respectivamente. Ahora probaremos que M y M' son equivalentes combinatoriamente una con la otra. Para esto, usamos el método del “deslizado de asas” [Sm61, Po66] que lo podemos describir para nuestro caso como sigue:

Las curvas ∂D y $\partial D'$ tienen el mismo tipo de homotopía en $|K''_2|$; por lo tanto, tienen el mismo tipo de homotopía en $N^{*''}$, es decir, existe un anillo singular de dimensión 2 en $N^{*''}$, el cual tiene como curvas fronteras a ∂D y $\partial D'$. Ya que $n \geq 4$, podemos deformar a este anillo singular en uno no-singular, digamos $J \subseteq \partial N^{*''}$ con $\partial J = \partial D \cup \partial D'$. Como consecuencia de todo esto, podemos deformar a ∂D sobre J en $\partial D'$. Por tanto, existe un homeomorfismo semilineal de $N^{*''}$ en si mismo que es igual a la identidad afuera de alguna vecindad de J y que lleva a la vecindad $\partial A_{n+1} \cap \partial N^{*''}$ de ∂D (en $\partial N^{*''}$), en la vecindad $\partial A'_{n+1} \cap \partial N^{*''}$ de $\partial D'$ (en $\partial N^{*''}$). Finalmente, podemos extender este homeomorfismo a uno semilineal de M sobre M' y así, estas son

equivalentes combinatoriamente.

Caso 3. La presentación μ' se obtiene de μ por medio de la operación Op_5 . Nuevamente, Sean $\Delta(F_n(\mu)) = \partial N^*$, $\Delta(F_n(\mu')) = \partial N^{*'}$, N^* y $N^{*'}$ como en los casos anteriores. Entonces obtenemos un complejo equivalente combinatoriamente a K'_2 , agregando un arco abierto $Y_{r(\mu)+1}$ (correspondiente al triangulo hueco del nuevo generador) y un 2-disco abierto $D_{p(\mu)+1}$ (correspondiente al polígono triangulado de la nueva relación), tales que $\partial D_{p(\mu)+1}$ contiene a $Y_{r(\mu)+1}$ solamente una vez. Escogemos a $Y_{r(\mu)+1}$ y $D_{p(\mu)+1}$ en \mathbb{R}^{n+1} de tal manera que $Y_{r(\mu)+1} \cap (\mathbb{R}^{n+1} - \text{Int } N^*)$ es un arco, digamos Y , y $(D_{p(\mu)+1} \cup Y_{r(\mu)+1}) \cap (\mathbb{R}^{n+1} - \text{Int } N^*)$ es un 2-disco, digamos D . Sea A_{n+1} una $(n+1)$ -bola, vecindad (triangulada) de D en $\mathbb{R}^{n+1} - \text{Int } N^*$. Entonces la variedad M , obtenida al agregar a N^* la 2-asa A_{n+1} , es equivalente combinatoriamente a $N^{*'}$. Pero por otro lado, como A_{n+1} es una $(n+1)$ -bola, tenemos que $\partial A_{n+1} \cap \partial N^*$ es una n -bola (vecindad del arco $\partial D \cap \partial N^* = \partial D - \text{Int } Y$). Así, M es equivalente combinatoriamente a N^* .

Caso 4. La presentación μ' se obtiene de μ por medio de la operación Op_5^{-1} . Intercambiando μ y μ' en el Caso 3, obtenemos este, y con esto concluimos la demostración del Lema. □

5.3. Resultados finales

Teniendo el Lema 5.2.1, nuestra meta está muy cerca. Ahora probaremos un teorema que nos permitirá construir reducciones de Turing entre los problemas computacionales algebraicos de isomorfismo y trivialidad, y los de clasificación de variedades y la verificación de la propiedad simplemente conexa. Combinando estas reducciones con el hecho de que los problemas algebraicos mencionados, pueden ser escogidos con un grado de dificultad arbitrario (Teorema 2.2.3), obtendremos el resultado final.

Teorema 5.3.1. *Para toda $n \geq 4$ y todo conjunto completamente computable⁴ \mathcal{Q} de presentaciones finitas de grupos, existe un conjunto computable*

⁴Para recordar lo que significa que un conjunto de presentaciones finitas de grupos sea (completamente) computable, vease la Definición 2.2.2

$K(n, \mathcal{Q})$ de presentaciones finitas de n -variedades, equipadas con estructuras diferenciables (compatibles combinatoriamente) tal que se cumplen las siguientes condiciones:

K1 $K(n, \mathcal{Q})$ es el rango de una función computable F_n , del conjunto $\overline{\mathcal{Q}}$ de presentaciones de grupos de la forma

$$\bar{\mu} = \mu * (4p(\mu) + 4r(\mu) + t), \quad \mu \in \mathcal{Q}, \quad t = 0, 1, \dots$$

en el conjunto de todas las presentaciones finitas de n -variedades. La función (posiblemente multivaluada) F_n^{-1} es computable, en el sentido de que existe un algoritmo para determinar, dada $\mathfrak{M} \in K(n, \mathcal{Q})$, todas las clases de congruencia de $F_n^{-1}(\mathfrak{M})$ (es decir, determinar un conjunto de presentaciones de grupos $\mu_1, \dots, \mu_q \in F_n^{-1}(\mathfrak{M})$, tal que toda $\bar{\mu} \in F_n^{-1}(\mathfrak{M})$ es congruente a alguna μ_1, \dots, μ_q).

K2 Si $\bar{\mu} \in \overline{\mathcal{Q}}$, entonces se tiene que $G(\bar{\mu}) \cong \pi_1(F_n(\bar{\mu}))$.

K3 Sean $\mu, \mu' \in \overline{\mathcal{Q}}$, tales que $G(\mu) \cong G(\mu')$. Si $c_{\mu, \mu'} = p(\mu') - p(\mu) + r(\mu) - r(\mu')$, entonces tenemos que

$$\beta'_2(F_n(\mu')) - \beta_2(F_n(\mu)) = \begin{cases} 2c_{\mu, \mu'} & \text{si } n = 4, \\ c_{\mu, \mu'} & \text{si } n > 4. \end{cases} \quad (5.1)$$

K4 $F_n(\mu) \approx_x F_n(\mu')$ ($x \in \{d, t, c, h\}$) si y sólo si $G(\mu) \cong G(\mu')$ y $\beta'_2(F_n(\mu')) = \beta_2(F_n(\mu))$.

Demostración. Sean $n \geq 4$ y \mathcal{Q} un conjunto completamente computable de presentaciones finitas de grupos.

K1. Sea F_n la función del Lema 5.2.1, con dominio restringido a $\overline{\mathcal{Q}}$; el conjunto $K(n, \mathcal{Q}) = F_n(\overline{\mathcal{Q}})$ es computable. Si $\mathfrak{M} \in K(n, \mathcal{Q})$, entonces por **i**), inciso (b) del Lema 5.2.1, existe un algoritmo para determinar todas las clases de congruencia de $F_n^{-1}(\mathfrak{M})$.

K2. Es inmediato de **ii**) del Lema 5.2.1.

K3. Sean $\mu, \mu' \in \overline{\mathcal{Q}}$, tales que $G(\mu) \cong G(\mu')$. Tomemos a $\tilde{\mu}, \tilde{\mu}' \in \mathcal{Q}$ y t, t' tales que

$$\begin{aligned} \mu &= \tilde{\mu} * t, \\ \mu' &= \tilde{\mu}' * t', \\ t &\geq 4p(\tilde{\mu}) + 4r(\tilde{\mu}), \quad t' \geq 4p(\tilde{\mu}') + 4r(\tilde{\mu}'). \end{aligned}$$

Supongamos, sin pérdida de generalidad, que $t \geq t'$, lo que implica que $t \geq p(\tilde{\mu}) + r(\tilde{\mu}')$. Ahora sea $t'' = t - p(\tilde{\mu}') + r(\tilde{\mu}') + p(\tilde{\mu}) - r(\tilde{\mu})$. Entonces $t'' \geq p(\tilde{\mu}') + r(\tilde{\mu})$ y por la definición de t'' , $p(\tilde{\mu}) + t - r(\tilde{\mu}) = p(\tilde{\mu}') + t'' - r(\tilde{\mu}')$. Por el Teorema 2.1.6, $\mu = \tilde{\mu} * t$ se puede transformar con una sucesión finita de operaciones $Op_1^{\pm 1}, Op_2, Op_3, Op_4, Op_5^{\pm 1}$ en la presentación $\tilde{\mu}' * t''$. Por **iv**) del Lema 5.2.1

$$F_n(\mu) \approx_x F_n(\tilde{\mu}' * t''). \quad (5.2)$$

Y se sigue de **iii**) del Lema 5.2.1 que

$$\beta_2(F_n(\mu)) - \beta_2(F_n(\tilde{\mu}')) = \begin{cases} 2t'' & \text{si } n = 4, \\ t'' & \text{si } n > 4, \end{cases}$$

y

$$\beta_2(F_n(\mu')) - \beta_2(F_n(\tilde{\mu}')) = \begin{cases} 2t' & \text{si } n = 4, \\ t' & \text{si } n > 4, \end{cases}$$

por lo cual

$$\beta_2(F_n(\mu')) - \beta_2(F_n(\mu)) = \begin{cases} 2(t' - t'') & \text{si } n = 4, \\ t' - t'' & \text{si } n > 4. \end{cases} \quad (5.3)$$

Pero

$$\begin{aligned} t - t'' &= p(\tilde{\mu}') - r(\tilde{\mu}') - p(\tilde{\mu}) + r(\tilde{\mu}) \\ &= p(\tilde{\mu}') + t' - t' - r(\tilde{\mu}') - p(\tilde{\mu}) - t + t + r(\tilde{\mu}) \\ &= p(\mu') - t' - r(\mu') - p(\mu) + t + r(\mu), \end{aligned}$$

por lo tanto, $t' - t'' = p(\mu') - r(\mu') - p(\mu) + r(\mu)$. Combinando este hecho con (5.3), obtenemos (5.1).

K4. (\Rightarrow) Si $F_n(\mu) \approx_x F_n(\mu')$, entonces $G(\mu) \cong \pi_1(F_n(\mu)) \cong \pi_1(F_n(\mu')) \cong G(\mu')$ y $\beta_2(F_n(\mu)) = \beta_2(F_n(\mu'))$. (\Leftarrow) Si $\beta_2(F_n(\mu)) = \beta_2(F_n(\mu'))$, entonces usando (5.3), $t' = t''$, con lo que $\tilde{\mu}' * t'' = \mu'$; por (5.2), tenemos que $F_n(\mu) \approx_x F_n(\mu')$, y esto concluye la demostración. \square

Del Teorema 5.3.1 obtenemos los siguientes corolarios:

Corolario 5.3.2. *Dado un conjunto \mathcal{Q} completamente computable de presentaciones finitas de grupos, se tiene que $ISOMORFISMO(\mathcal{Q}) \equiv_T x-EQUIVALENCIA(K(n, \mathcal{Q}))$ ($x \in \{d, t, c, h\}$).*

Demostración. Sea \mathcal{Q} un conjunto completamente computable de presentaciones finitas de grupos. Mostraremos primero que existe una MTO M_x^A , donde $A = \text{ISOMORFISMO}(\mathcal{Q})$, que es capaz de decidir a x -EQUIVALENCIA($K(n, \mathcal{Q})$). He aquí el algoritmo⁵:

$M_x^A =$ “En la entrada $(\mathfrak{M}, \mathfrak{M}')$:

1. Calcula $\bar{\mu}, \bar{\mu}' \in \overline{\mathcal{Q}}$ tales que $\mathfrak{M} = F_n(\bar{\mu}), \mathfrak{M}' = F_n(\bar{\mu}')$.
2. Calcula los segundos números de Betti, β_2, β_2' de $\mathfrak{M}, \mathfrak{M}'$.
3. Si $\beta_2 \neq \beta_2'$, entonces *rechaza*.
4. Si $\beta_2 = \beta_2'$, calcula $\mu, \mu' \in \mathcal{Q}$ tales que $\bar{\mu} = \mu * t, \bar{\mu}' = \mu' * t'$.
5. Pregunta al oráculo A si $\mu \cong \mu'$.
6. Si la respuesta es afirmativa, *acepta*, y *rechaza* en otro caso.”

El algoritmo claramente termina, sólo necesitamos verificar los pasos **1**, **2** y **4**. El paso **1** es posible, ya que F_n^{-1} es computable en el sentido descrito en el inciso **K1** del Teorema 5.3.1. Como $\mathfrak{M}, \mathfrak{M}'$ tienen triangulaciones de las variedades que representan, los segundos números de Betti de estas son computables y así el paso **2** es posible. Por la definición de $\overline{\mathcal{Q}}$, el calcular $\mu, \mu' \in \mathcal{Q}$ a partir de $\bar{\mu}, \bar{\mu}'$ en el paso **4** es posible, y así, por el inciso **K4** del Teorema 5.3.1, hemos Turing reducido el problema de x -equivalencia para el conjunto $K(n, \mathcal{Q})$ al problema de isomorfismo para el conjunto \mathcal{Q} .

Ahora mostramos la MTO M^{B_x} , donde $B_x = x$ -EQUIVALENCIA($K(n, \mathcal{Q})$), que decide a $\text{ISOMORFISMO}(\mathcal{Q})$:

$M^{B_x} =$ “En la entrada (μ, μ') :

1. Calcula $\bar{\mu}, \bar{\mu}' \in \overline{\mathcal{Q}}$ y $t_{\mu, \mu'} \in \mathbb{N}$ tales que

$$\begin{aligned} \bar{\mu} &= \mu * (4p(\mu) + 4r(\mu)), \\ \bar{\mu}' &= \mu' * (4p(\mu') + 4r(\mu')), \\ t_{\mu, \mu'} &= p(\bar{\mu}') - r(\bar{\mu}') - p(\bar{\mu}) + r(\bar{\mu}). \end{aligned}$$
2. Calcula $\mathfrak{M} = F_n(\bar{\mu} * t_{\mu, \mu'}), \mathfrak{M}' = F_n(\bar{\mu}')$.
3. Pregunta al oráculo B_x si $\mathfrak{M} \approx_x \mathfrak{M}'$.
4. Si la respuesta es afirmativa, *acepta*, y *rechaza* en otro caso.”

Nuevamente, el algoritmo siempre termina, y cada paso de este es fácilmente realizable. Mostremos que es correcto. Primero notamos que, si $\mu \cong \mu'$, entonces $\bar{\mu} * t_{\mu, \mu'} \cong \bar{\mu}'$ y combinando el inciso **K3** del Teorema 5.3.1 con una cuenta sencilla, tenemos que $\beta_2(F_n(\bar{\mu} * t_{\mu, \mu'})) = \beta_2(F_n(\bar{\mu}'))$. Supongamos que

⁵De aquí en adelante, omitiremos en todos los algoritmos los pasos que verifican si la entrada es válida y asumiremos que ésta siempre es correcta.

$G(\mu) \cong G(\mu')$, entonces por lo dicho anteriormente y el inciso **K4** del Teorema 5.3.1, $F_n(\bar{\mu} * t_{\mu, \mu'}) \approx_x F_n(\bar{\mu}')$, por lo que en el paso **3**, la respuesta del oráculo B_x sera afirmativa, y en el paso **4**, M^{B_x} aceptará. Si $G(\mu) \not\cong G(\mu')$, entonces $G(\bar{\mu} * t_{\mu, \mu'}) \not\cong G(\bar{\mu}')$, y nuevamente, por **K4** del Teorema 5.3.1, $F_n(\bar{\mu} * t_{\mu, \mu'})$ no es x -equivalente a $F_n(\bar{\mu}')$ y en el paso **3**, B_x dará una respuesta negativa, con lo que M^{B_x} rechazará en el paso **4**. De este modo, concluimos que el algoritmo es correcto. Así, los problemas ISOMORFISMO(\mathcal{Q}) y x -EQUIVALENCIA($K(n, \mathcal{Q})$) son Turing reducibles uno al otro, como queríamos mostrar. \square

Corolario 5.3.3. *Si el conjunto de presentaciones \mathcal{Q} tiene a la presentación vacía $\mu_0 = (\emptyset, \emptyset)$ del grupo trivial y, si para algún $W \in \mathbb{N}$, $5p(\mu) + 4r(\mu) \leq W$ para toda $\mu \in \mathcal{Q}$, entonces el problema de x -equivalencia a $\overline{\mathfrak{M}} = F_n(\mu_0 * W)$ en $K(n, \mathcal{Q})$ es Turing equivalente al problema de trivialidad para \mathcal{Q} .*

Demostración. Nuevamente, mostramos los algoritmos requeridos. Para Turing reducir x -EQUIVALENCIA($K(n, \mathcal{Q}), \overline{\mathfrak{M}}$) a $C = \text{TRIVIAL}(\mathcal{Q})$, construimos la MTO K_x^C , con el siguiente programa:

$K_x^C =$ “En la entrada \mathfrak{M} :

1. Calcula $\bar{\mu} \in \overline{\mathcal{Q}}$ tal que $\mathfrak{M} = F_n(\bar{\mu})$.
2. Calcula los números de Betti, $\beta_2, \bar{\beta}_2$ de $\mathfrak{M}, \overline{\mathfrak{M}}$.
3. Si $\beta_2 \neq \bar{\beta}_2$, entonces *rechaza*.
4. Si $\beta_2 = \bar{\beta}_2$, calcula $\mu \in \mathcal{Q}$ tal que $\bar{\mu} = \mu * t_\mu$, con $t_\mu \geq 4p(\mu) + 4r(\mu)$.
5. Pregunta al oráculo C si $G(\mu)$ es el grupo trivial.
6. Si la respuesta es afirmativa, *acepta*, y *rechaza* en otro caso.”

El argumento para verificar que este algoritmo es correcto es análogo al de la máquina M_x^A del corolario anterior, por lo que lo omitimos.

El algoritmo K^{D_x} para Turing reducir $\text{TRIVIAL}(\mathcal{Q})$ a $D_x = x$ -EQUIVALENCIA($K(n, \mathcal{Q}), \overline{\mathfrak{M}}$) es el siguiente:

$K^{D_x} =$ “En la entrada μ :

1. Calcula $t_\mu = W - p(\mu) + r(\mu)$.
2. Calcula $\mathfrak{M} = F_n(\mu * t_\mu)$.
3. Pregunta al oráculo D_x si $\mathfrak{M} \approx_x \overline{\mathfrak{M}}$.
4. Si la respuesta es afirmativa, *acepta*, y *rechaza* en otro caso.”

Es sencillo ver que el algoritmo siempre termina, veamos que es correcto. Como por hipótesis, $5p(\mu) + 4r(\mu) \leq W$, entonces $t_\mu \geq 4p(\mu) + 4r(\mu)$, por lo que $\mu * t_\mu \in \overline{\mathcal{Q}}$. Supongamos que μ presenta el grupo trivial, entonces $G(\mu * t_\mu) \cong G(\mu) \cong G(\mu_0)$ y por el inciso **K3** del Teorema 5.3.1, $\beta_2(F_n(\mu * t_\mu)) = \beta_2(F_n(\mu_0)) = \beta_2(\overline{\mathfrak{M}})$. Por el inciso **K4** del mismo teorema, $F_n(\mu * t_\mu) \approx_x \overline{\mathfrak{M}}$; así, después de ejecutarse el paso **3**, en el paso **4** K^{D_x} aceptará y la salida es correcta. Si μ no presenta el grupo trivial, entonces por el inciso **K4** del Teorema 5.3.1, $F_n(\mu * t_\mu)$ no es x -equivalente a $\overline{\mathfrak{M}}$ y en el paso **4**, K^{D_x} rechazará, por lo que la salida es correcta, y así concluimos la demostración. □

Combinando estos corolarios con los resultados de Boone, dados en el Teorema 2.2.3, obtenemos finalmente el resultado principal.

Teorema 5.3.4. *Para cada dimensión $n \geq 4$ y cada grado de Turing $\mathbf{e} \in \mathcal{E}$, existe un conjunto computable $C(n, \mathbf{e})$ de presentaciones finitas de n -variedades, equipadas con una estructura diferenciable y una estructura combinatoria compatible, tales que, para cada $x = d, t, c, h$:*

T1 x -EQUIVALENCIA($C(n, \mathbf{e})$) $\in \mathbf{e}$.

T2 $C(n, \mathbf{e})$ contiene una cierta presentación $\overline{\mathfrak{M}}$ de una n -variedad simplemente conexa, tal que x -EQUIVALENCIA($C(n, \mathbf{e}), \overline{\mathfrak{M}}$) $\in \mathbf{e}$.

T3 SIMPLEMENTE CONEXA($C(n, \mathbf{e})$) $\in \mathbf{e}$.

Demostración. Sean $n \geq 4$ y $\mathbf{e} \in \mathcal{E}$. Por el Teorema 2.2.3, existe un conjunto computable $\mathcal{Q}(\mathbf{e})$ de presentaciones finitas de grupos, tal que los conjuntos ISOMORFISMO($\mathcal{Q}(\mathbf{e})$) y TRIVIAL($\mathcal{Q}(\mathbf{e})$) tienen grado de Turing \mathbf{e} . El conjunto $\mathcal{Q}(\mathbf{e})$ tiene a todos sus elementos con el mismo conjunto de generadores y el mismo número de relaciones, por lo que en este caso, computable implica completamente computable.

T1. Por el Teorema 5.3.1, existe un conjunto computable $K(n, \mathcal{Q}(\mathbf{e}))$, de presentaciones finitas de n -variedades; hagamos $C(n, \mathbf{e}) = K(n, \mathcal{Q}(\mathbf{e}))$. Entonces por el Corolario 5.3.2, los conjuntos ISOMORFISMO($\mathcal{Q}(\mathbf{e})$) y x -EQUIVALENCIA($C(n, \mathbf{e})$) son Turing equivalentes, con lo cual tenemos que x -EQUIVALENCIA($C(n, \mathbf{e})$) $\in \mathbf{e}$.

T2. Al tener todos los elementos de $\mathcal{Q}(\mathbf{e})$ el mismo número de generadores y relaciones, es claro que existe $W \in \mathbb{N}$ tal que $5p(\mu) + 4r(\mu) \leq W$, para toda μ ; además, la presentación vacía del grupo trivial $\mu_0 = (\emptyset, \emptyset)$ esta en $\mathcal{Q}(\mathbf{e})$. Aplicando el Corolario 5.3.3, tenemos que

$$x\text{-EQUIVALENCIA}(C(n, \mathbf{e}), \overline{\mathfrak{M}}) \equiv_T \text{TRIVIAL}(\mathcal{Q}(\mathbf{e})),$$

donde $\overline{\mathfrak{M}} = F_n(\mu_0 * W)$, con lo que obtenemos que $x\text{-EQUIVALENCIA}(C(n, \mathbf{e}), \overline{\mathfrak{M}}) \in \mathbf{e}$.

T3. En este caso, es sencillo ver que $x\text{-EQUIVALENCIA}(C(n, \mathbf{e}), \overline{\mathfrak{M}}) \leq_T \text{SIMPLEMENTE CONEXA}(C(n, \mathbf{e})) \leq_T \text{TRIVIAL}(\mathcal{Q}(\mathbf{e}))$, por lo cual $\text{SIMPLEMENTE CONEXA}(C(n, \mathbf{e})) \in \mathbf{e}$. Esto termina la demostración. \square

Este es el resultado buscado a lo largo de esta obra, que nos dice que podemos escoger variedades $C^\infty\text{-PL}$, tales que los problemas topológicos representados por los lenguajes $x\text{-EQUIVALENCIA}(C(n, \mathbf{e}))$ ($x = d, t, c, h$) y $\text{SIMPLEMENTE CONEXA}(C(n, \mathbf{e}))$, sean “tan imposibles” como queramos. Como consecuencia de esto, logramos el objetivo principal de esta obra y además podemos obtener otros resultados también muy importantes.

Corolario 5.3.5. *Para cada dimensión $n \geq 4$, existen variedades $C^\infty\text{-PL}$, tales que los problemas relacionados con decidir si cualesquiera dos variedades son difeomorfas, homeomorfas, equivalentes combinatoriamente u homotópicas, son no computables. Más aún, el problema de decidibilidad de la propiedad de ser simplemente conexa en estas variedades es no computable.*

Demostración. Tomando a $\mathbf{e} = \mathbf{0}'$, en el Teorema 5.3.4, vemos que los lenguajes $d\text{-EQUIVALENCIA}(C(n, \mathbf{0}'))$, $t\text{-EQUIVALENCIA}(C(n, \mathbf{0}'))$, $c\text{-EQUIVALENCIA}(C(n, \mathbf{0}'))$, $h\text{-EQUIVALENCIA}(C(n, \mathbf{0}'))$ y $\text{SIMPLEMENTE CONEXA}(C(n, \mathbf{0}'))$, son Turing equivalentes al problema de detención de máquinas de Turing HP (que como sabemos, esta en $\mathbf{0}'$). Ya que HP no es computable, entonces los lenguajes mencionados no son computables. Como consecuencia, existen variedades $C^\infty\text{-PL}$, que no pueden ser clasificadas, y tampoco es posible decidir si son simplemente conexas o no. \square

Corolario 5.3.6. *Para cada dimensión $n \geq 4$, existen variedades C^∞ -PL, tales que los problemas relacionados con decidir si cualesquiera dos variedades son difeomorfas, homeomorfas, equivalentes combinatoriamente u homotópicas, y el problema de decidibilidad de la propiedad de ser simplemente conexa en estas variedades son computables.*

Demostración. Aplicamos el Teorema 5.3.4, usando el grado de Turing 0 de los lenguajes computables. □

Corolario 5.3.7. *Para cada dimensión $n \geq 4$, existen variedades C^∞ -PL, tales que los problemas relacionados con decidir si cualesquiera dos variedades son difeomorfas, homeomorfas, equivalentes combinatoriamente u homotópicas, y el problema de decidibilidad de la propiedad de ser simplemente conexa en estas variedades son listables (c.e.).*

Demostración. Inmediata al combinar el Corolario anterior y la Proposición 1.2.6, inciso (ii). □

Hemos llegado al final de este trabajo. Esperamos que hayamos logrado nuestros objetivos, que han sido dar un relato entendible y agradable y mostrar que ramas matemáticas como la topología y la computación, no deben estar separadas, muchos beneficios pueden obtener una de la otra y juntas, descubrir cosas que pueden cambiar el rumbo de la matemática misma. Pensamos que, así como la topología es una de los fundamentos de la matemática, la ciencia de la computación también lo es, ya que esta nos dicta que es lo que podemos hacer, y que no. Y en el caso de poder hacerlo, cuanto nos vamos a tardar.

Con esto concluimos esta obra.

Bibliografía

- [AF06] K. Ambus-Spies, P. A. Fejer: Degrees of Unsolvability. No publicado. Disponible en http://www.cs.umb.edu/~fejer/articles/History_of_Degrees.pdf. 2006.
- [BHP68] W. W. Boone, W. Haken, V. Poénaru: On recursively unsolvable problems in topology and their clasification. Contributions to Math. Logic (Colloquium, Hannover, 1966). North-Holland Publishing Company, Amsterdam, 1968.
- [Bo68] W. W. Boone: Decision problems about algebraic and logical systems as a whole and recursively enumerable degrees of unsolvability. Contributions to Math. Logic (Colloquium, Hannover, 1966). North-Holland Publishing Company, Amsterdam, 1968.
- [Bo59] W. W. Boone: The word problem. Ann. Math., 70. p207-265. 1959.
- [Ca35] S. S. Cairns: Triangulation of the manifold of class one. Bull. Amer. Math. Soc. 41. p549-552, 1935.
- [Ca61] S. S. Cairns: The manifold smoothing problem. Bull. Amer. Math. Soc. 67. p237-238, 1961.
- [Ca61-2] S. S. Cairns: A simple triangulation method for smooth manifolds. Bull. Amer. Math. Soc. 67. p389-390, 1961.
- [Ch36] A. Church: An unsolvable problem of elementary number theory. Amer. J. Math., 58. p345-363. 1936.
- [Ch33] A. Church: A set of postulates for the foundation of logic. Ann. Math., 33-34. p346-366, 839-864. 1933.

- [Co04] S. Barry Cooper: *Computability Theory*. USA, Chapman & Hall/CRC mathematics. 2004.
- [De11] M. Dehn: Über unendliche diskontinuierliche gruppen. *Math. Ann.*, 71. p73-77. 1911.
- [DH] A. Douady, L. Héroult: Arrondissement des variétés à coins. *Commentarii mathematici Helvetici*. p484-491.
- [Fr88] J. B. Fraleigh: *Álgebra abstracta*. Wilmington. Delaware, E.U.A., Addison-Wesley Iberoamericana, 1988.
- [Fr57] R. M. Friedberg: Two recursively enumerable sets of incomparable degrees of unsolvability (solution to Post's problem, 1944). *Proc. Nat. Acad. Sci. U.S.A.*, 43. p236-238. 1957.
- [GR05] E. Gafni, S. Rajsbaum: Musical Benches. *Proc. 19th Int. Symposium on Distributed Computing (DISC'05)*. Springer Verlag LNCS To Appear. Cracow (Poland) 2005.
- [Ga03] J. Galaviz: *Elogio de la pereza. La ciencia de la computación en una perspectiva histórica*. Facultad de Ciencias, UNAM, México, D.F. 2003.
- [Gö65] K. Gödel: On undecidable propositions of formal mathematical systems. Hewlitt, New York, M. Davis, ed., Raven Press. 1965.
- [GP75] V. Guillemin, A. Pollack: *Differential Topology*. Pearson Education, Inc. (Prentice Hall). 1975.
- [HS99] M. Herlihy, Nir Shavit: The topological structure of asynchronous computability. *J. of the ACM*, 46(6). p858-923. 1999.
- [HR00] M. Herlihy, S. Rajsbaum: Algebraic Spans. *Mathematical Structures in Computer Science*. 10(4). p549-573. 2000.
- [HFFA03] D. Hinrichsen, J. L. Fernández Muñoz, A. Fraguera Collar, A. Álvarez Prieto: *Topología general*. México. Sociedad Matemática Mexicana. 2003.
- [Hi76] M. W. Hirsch: *Differential topology*. New York, Springer. 1976.

- [Hu69] J. F. P. Hudson: Piecewise linear topology. New York: W. A. Benjamin. 1969.
- [Ke60] M. A. Kervaire: A manifold which does not admit any differentiable structure. *Commentarii Mathematici Helvetici*. 34. p257-270, 1960.
- [Ko93] A. A. Kosinski: Differential manifolds. Boston Academy Press. 1993.
- [Ko97] D. C. Kozen: Automata and Computability. New York, Springer-Verlag. 1997.
- [Ma92] J. Margalef Roig: Differential topology. Amsterdam : North-holland. 1992.
- [Ma58] A. A. Markov: Insolubility of the problem of homeomorphy. Proc. Intern. Congress of Mathematicians. p300-306, Cambridge University Press. 1958.
- [Ma77] A. A. Markov, N. M. Nagorny: The Theory of algorithms. Kluwer Academic Publishers. 1977.
- [Ma93] Y. Matiyasevich: Hilbert's Tenth Problem. Cambridge, Mass., London, MIT Press, 1993.
- [Mi56] J. Milnor: On manifolds homeomorphic to the 7-sphere. *Ann. Math.* 64. p399-405, 1956.
- [Mi61] J. Milnor: Two complexes wich are homeomorphic but combinatorially distinct. *Ann. Math.*, 74. p575-590. 1961.
- [Mi61-2] J. Milnor: A procedure for killing the homotopy groups of differentiable manifolds. *Symposia in Pure Mathematics*. Am. Math. Soc., Vol III. p39-55. 1961.
- [Mu56] A. A. Mučnik: On the unsolvability of the problem of reducibility in the theory of algorithms. *Dokl. Akad. Nauk SSSR (N.S.)*, 108. p194-197. 1956.
- [Mu60] J. Munkres: Obstructions to the smoothing of piecewise-differentiable homeomorphisms. *Ann. Math.* 72. p521-554, 1960.

- [Mu00] J. Munkres: *Topology*. 2nd Edition. Upper Saddle River, New Jersey; México. Prentice Hall. c2000.
- [Na52] J. Nash: Real algebraic manifolds. *Ann. Math.* 56. p405-421, 1952.
- [No55] P. S. Novikov: On the algorithmic unsolvability of the word problem in group theory. *Trudy Math. Inst. Steklov*, 44. p1-143. 1955. (en Ruso)
- [Pa94] C. H. Papadimitriou: *Computational Complexity*. University of California, San Diego, Addison-Wesley. 1994.
- [Po66] V. Poénaru: Sur la theorie des immersions. *Topology* 1. p81-100. 1966.
- [Poo08] B. Poonen: Undecidability in Number Theory. *Notices of the AMS*. 55, Number 3. p344-350. 2008.
- [Po43] E. Post: Formal reductions of the general combinatorial decision problem. *Amer. J. Math.*, 65. p197-215. 1943.
- [Po36] E. Post: Finite combinatory processes-formulation, I. *J. Symbolic Logic*, 1. p103-105. 1936.
- [Po44] E. Post: Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50. p284-316. 1944.
- [RCSARC96] A. A. Ranick, A. J. Casson, D. P. Sullivan, M. A. Armstrong, C. P. Rourke, G. E. Cooke: *The Hauptvermutung book. A collection of papers on the topology of manifolds*. K-Monographs in Mathematics 1. Kluwer. 1996.
- [Re86] C. Reid: *Hilbert*. Berlin, Heidelberg, New York, London, Paris, Tokyo, Springer-Verlag, 1986.
- [Ro95] J. J. Rotman: *An introduction to the theory of groups*. New York. Springer-Verlag. 1995.
- [Ro88] J. J. Rotman: *An introduction to Algebraic Topology*. New York, Berlin, Heidelberg, London, Paris, Tokyo. Springer-Verlag. 1988.
- [RS72] C. P. Rourke, B. J. Sanderson: *Introduction to Piecewise-Linear Topology*. New York, Heidelberg, Berlin. Springer-Verlag. 1972.

- [ST51] H. Seifer, W. Threlfall: Lecciones de Topología. Publicaciones del Instituto Jorge Juan de matemáticas (Traducido del alemán), Madrid, 1951.
- [SE] Ákos Seress: An introduction to computational group theory. Ohio State University, Columbus.
- [Si05] M. Sipser: Introduction to the theory of Computation. MIT, Boston Massachusetts, Thomson Course Technology. 2005.
- [Sm61] S. Smale: Generalized Poincaré conjecture in dimensions greater than four. *Ann. Math.* 74. p391-406, 1961.
- [Sm62] S. Smale: On the structure of manifolds. *Amer. J. Math.* 84. p387-399, 1962.
- [Sp66] E. H. Spanier: Algebraic Topology. New York. Springer-Verlag. 1966.
- [Sta60] J. Stallings: Polyhedral homotopy spheres. *Bull. Amer. Math. Soc.* 66. p485-488, 1960.
- [Ta67] A. Tarski: A decision method for elementary algebra and geometry. (Santa Monica, Rand, 1948) Paris, Institut Blaise Pascal. 1967.
- [Tu50] A. M. Turing: The word problem in semi-groups with cancellation. *Ann. Math.*, 52. p491-505. 1950.
- [Tu39] A. M. Turing: Systems of logic based on ordinals. *Proc. London Math. Soc.* 45. p161-228. 1939.
- [Tu36] A. M. Turing: On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* 42. p230-265. 1936.
- [VKF77] I. A. Volodin, V. E. Kuznetsov, A. T. Fomenko: On the recognition problem of the standard 3-sphere. *Uspekhi Mat. Nauk*, 32 (5). p71-168. 1977. (en Ruso)
- [WR10-13] A. N. Whitehead, B. Russell: Principia Mathematica. Cambridge, Cambridge University Press. Tres volúmenes, 1910-1913.
- [Whi40] J. H. C. Whitehead: On C^1 -complexes. *Ann. Math.* 41. p809-824, 1940.

- [Wh36] H. Whitney: Differentiable manifolds. Ann. Math. 37. p645-680, 1936.