



UNIVERSIDAD
DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y
Telecomunicación
Facultad de Ciencias

DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

Teoría de nudos

Presentado por:

Iñaki Melguizo Marcos

Tutor:

Rafael Ortega Ríos

Departamento de Matemática Aplicada

Pedro A. García Sánchez

Departamento de Álgebra

Curso académico 2021-2022

Teoría de nudos

Iñaki Melguizo Marcos

Iñaki Melguizo Marcos *Teoría de nudos*.
Trabajo de fin de Grado. Curso académico 2021-2022.

**Responsable de
tutorización**

Rafael Ortega Ríos
Departamento de Matemática Aplicada

Pedro A. García Sánchez
Departamento de Álgebra

Doble Grado en Ingeniería
Informática y Matemáticas

Escuela Técnica Superior
de Ingenierías Informática
y Telecomunicación
Facultad de Ciencias

Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D./Dña. Iñaki Melguizo Marcos

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2021-2022, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 31 de diciembre de 2021

Fdo: Iñaki Melguizo Marcos

Índice general

Resumen	IX
Summary	XI
Introducción	XIII
1 Motivación	XIV
2 Objetivos	XIV
I Teoría de nudos	1
1 Síntesis	3
2 Nudos y Tipos de nudos	11
2.1 Nudo como objeto matemático	11
2.2 Nudos equivalentes	13
2.3 Nudos admisibles y no admisibles	14
2.4 Proyecciones de nudos	15
2.5 Invariantes de nudos	27
3 Grupo de un nudo	29
3.1 Presentación superior e inferior del grupo de un nudo	29
3.1.1 Arcos superiores e inferiores	29
3.1.2 Grupo libre y presentación de un grupo	30
3.1.3 Cálculo de la presentación superior e inferior	35
3.2 Presentación superior e inferior del grupo de un nudo, continuación	49
3.2.1 Caracterización de la presentación inferior del grupo de un nudo	49
3.2.2 Grupo fundamental de un espacio topológico	51
3.2.3 Cálculo de la presentación superior del grupo de un nudo	53
3.3 La presentación Wirtinger	68
3.4 Ejemplos de presentaciones	70
3.4.1 Nudo trivial	71
3.4.2 Nudo trébol	72
3.4.3 Nudo de ocho	73
3.5 Existencia de tipos de nudos no triviales	75

II	Parte Informática	77
4	Codificación de nudos	79
4.1	Diagrama de un nudo	79
4.2	Formas de codificar el diagrama de un nudo	81
4.2.1	Código de un diagrama etiquetado	81
4.2.2	Código de Gauss	88
4.2.3	Código de Gauss Extendido	90
4.2.4	Notación Dowker de un nudo	98
5	Estudio computacional de un nudo mediante su notación Dowker	107
5.1	Introducción	107
5.2	Tutorial del funcionamiento de la clase Nudo	108
5.3	Conclusiones y trabajos futuros	121
A	Documentación de la clase Nudo	123
	Bibliografía	143

Resumen

En este trabajo empezaremos estudiando teóricamente un nudo como objeto matemático. Comenzaremos por la definición de este, continuando por establecer el criterio de equivalencia entre dos nudos, y finalizaremos con el cálculo de dos presentaciones del grupo fundamental del complemento de un nudo, al que llamaremos grupo de un nudo. El grupo de un nudo es un invariante de nudos, lo que implica que si comprobamos que los grupos de dos nudos no son isomorfos, habremos demostrado que dichos nudos no son equivalentes. En este hecho radica la relevancia de dicho cálculo.

Una vez hayamos finalizado el estudio matemático de los nudos, estudiaremos diferentes codificaciones de los nudos, que serán necesarias para que un programa informático pueda obtener la información asociada a un nudo concreto mediante un conjunto de datos. Por último, crearemos una clase en *Python* que tome como entrada la notación Dowker de un nudo, una de esas codificaciones de nudos mencionadas anteriormente. Dicha clase proveerá métodos para calcular la presentación superior e inferior del grupo del nudo asociado a dicha notación Dowker, siguiendo el algoritmo visto en la parte matemática. Además, dicha clase contendrá métodos que, utilizando el módulo *Turtle* de *Python*, dibujarán diferentes diagramas asociados a ese nudo, ayudando de esa forma a comprender mejor el proceso de cálculo de la presentación superior e inferior del grupo de dicho nudo.

PALABRAS CLAVE: nudo, presentación, grupo, notación Dowker, proyección.

Summary

Knots have always been an element present in our lives, and everyone has come face to face with a knot, such as the knot that forms the cord of a pair of trainers, or the knot that forms the cord of a mobile phone charger when it gets tangled. We want to be able to classify these knots, however, if we consider knots as we know them, any knot can be transformed into any other knot, since we can untie it, and once untied, tie it to obtain any other knot. Therefore, to study them mathematically, we will define knots as sets of \mathbb{R}^3 homeomorphic to a circumference, that is to say, to obtain a mathematical knot we would have to join the ends of the physical knot.

Subsequently, we will define the concept of knot equivalence, since it is a non-trivial problem to study whether two knots are equivalent or not. Moreover, the main objective of knot theory is the classification of knots, being two knots of the same type if they are equivalent. To try to help solve the problem of determining the equivalence or non-equivalence of two knots, a series of properties or data associated with knots were discovered, which fulfil the property that all knots of the same type have equivalent properties or associated data. These properties or data are called knot invariants. Thus, if two knots have a non-equivalent value for a given knot invariant, these knots will not be equivalent.

Among the knot invariants we find the group of a knot, which is the fundamental group of the complement of the knot. This invariant is very powerful, that is, it allows us to distinguish many types of knots by checking that the fundamental groups of two knots are not isomorphic. However, it can sometimes be quite complicated to prove whether two groups are isomorphic or not. In one way or another, in this project we will study this invariant, for which our main objective in the mathematical part of the work will be to calculate two presentations of the group of a knot, which we will call the over and under presentations of the group of the knot.

The knots that we will study will be the equivalent knots to polygonal knots, which we will call tame knots. The basic idea for calculating the over and under presentations of a polygonal knot is as follows, bearing in mind that every knot will have a certain orientation associated with it.

First, we will project the knot in the plane $\mathbb{R}^2 \times \{0\}$ by means of the orthogonal projection on that plane, which we will call \mathcal{P} . We can consider, by a theorem seen in the project, that every point of the projection of the knot will only be the image under the projection \mathcal{P} of one or two points. The points that have two preimages will be called double points of the projection of the knot, the one with the larger z -coordinate we will call it, the overcrossing point, and the other is the corresponding undercrossing point. We will divide the knot into two classes of closed, connected segmented arcs, so that every undercrossing point is contained in a underpass and every overcrossing point in a overpass. We will label the overpasses and underpasses, and after projecting the knot already divided into overpasses

Summary

and underpasses, for each underpass [overpass] we will define a path in $\mathbb{R}^2 \times \{0\}$ that goes through a Jordan curve surrounding the projection of the underpass [overpass]. The Jordan curves associated with the underpasses [overpasses] arcs must not intersect each other. The relators of the over [under] presentation of the knot group will be determined by the overpasses with which that path intersects as it traverses its associated Jordan curve in a counterclockwise [clockwise] direction and the sign of those intersections, which will also be introduced in the project.

In the second part of the project, we will create a Python class that allows us to visualise what we will call the diagram of a knot, and to calculate the over and under presentations of the knot's group. In order for this class or any program to be able to return information about a knot or represent its diagram, it has to receive information from the knot that makes it uniquely identifiable. For this purpose, different knot codes and notations were created.

Firstly, we will describe the concept of diagram, which in short is the projection of a given knot with information about, at each crossing of this projection, which strand is the one that passes over and which one passes under it. Once we have defined the concept of a knot, we will study four encodings of knot diagrams, passing from the diagram of a knot to its associated notation and viceversa for each one of the four encodings. Also, we will execute functions from the *Sage Knot* class and the *Link* class from the *SnapPy Python* package that will draw the knot diagrams, as well as passing from one notation to another.

Finally, we will choose the Dowker notation as the input for the constructor of the class we will create in Python. This class will use the *Turtle* module to draw the knot associated with the Dowker notation, divide the knot into overpasses and underpasses and draw its diagram using blue to represent the overpasses and red to represent the underpasses. It will also represent the Jordan curves associated to the underpasses [overpasses] and will help us to calculate the over [under] presentation of the knot group associated to that Dowker notation. In addition, this program will have two methods called *get_upper_presentation* and *get_lower_presentation* that will return, as their names indicate, the over and under presentations of the group. Documentation of the most relevant methods and examples of their use can be found in the last chapter of the project.

KEY WORDS: knot, presentation, group, Dowker notation, projection.

Introducción

Durante siglos, los nudos han sido herramientas imprescindibles para diferentes profesiones y trabajos. Entre los marineros, desde épocas históricas remotas, se desarrolló todo un conjunto de tipos y variedades de nudos según su poder de amarre y resistencia. Persiste aún su uso y diversidad para el manejo de embarcaciones y se mantiene su función primordial: facilitar la atadura de cabos, permitir firmeza y evitar el desanudado. Históricamente la técnica de nudos también ha sido utilizada por artesanos de todas las culturas (sirviéndose de ataduras, lazadas o enlaces con diversos materiales) para trabajos artísticos o menestrales.

Para el estudio matemático de los nudos, llamado teoría de nudos, se ha denominado nudo matemático a un nudo físico cuyos extremos están unidos. De esta forma, si consideramos un nudo como una cuerda cuyos extremos se encuentran unidos, ya no se podría deformar un nudo hasta llegar a cualquier otro sin romper dicha cuerda y volver a pegar sus extremos. En consecuencia, esta definición matemática de nudo nos permitire establecer un criterio de equivalencia entre los mismos.

Aunque científicos como Carl Friedrich Gauss comenzaron a estudiar los nudos desde una óptica matemática, el interés por esta perspectiva aumentó notablemente desde finales del siglo XIX. Lord Kelvin ya propuso en ese periodo que los átomos eran nudos formados por pequeños vórtices. Esta teoría propició un enorme interés en buscar modelos de clasificación de los nudos, interpretando que si se lograba tabular todos los nudos posibles se podría llegar a explicar como los átomos absorben y emiten luz. Con posterioridad, en el apogeo de las teorías atómicas, numerosos científicos como el físico Peter Guthrie Tait comenzaron a tabular los nudos pensando que podrían estar creando una tabla de elementos.

Una vez descartada la teoría atómica de Kelvin se perdió momentáneamente el interés por la clasificación de los nudos. No es hasta la década de los años veinte del pasado siglo cuando, gracias a las aportaciones de Kurt Reidemeister, Max Dehn y J. W. Alexander, se produjeron notables avances en la compleja tarea de clasificar los nudos.

En realidad, la tarea de clasificación de los nudos fue y sigue siendo el problema central de la teoría de nudos, ya que no existen propiedades o datos asociados a un determinado tipo de nudo que sean únicos para nudos de dicho tipo. En cambio, lo que sí que existen son unas determinadas propiedades, tales que si dos nudos son equivalentes dichas propiedades serán también equivalentes. A dichas propiedades las llamaremos invariantes de nudos, de los cuales, en este trabajo nos centraremos en el llamado **grupo de un nudo**, que es el grupo fundamental del complemento de dicho nudo. Al tratarse de un invariante de nudos, si podemos comprobar que los grupos de dos nudos no son isomorfos, podremos asegurar que dichos nudos no son equivalentes.

1. Motivación

La teoría de nudos es un campo en pleno auge en la actualidad. Se están encontrando aplicaciones prácticas en la teoría de cuerdas, en la replicación del ADN o en criptología. Esto está provocando que un número considerable de investigadores estén trabajando en este área del conocimiento, y por ello, nos pareció muy interesante centrar nuestro trabajo en la teoría de nudos.

De los invariantes de nudos existentes, decidimos centrarnos en el grupo de un nudo, ya que para su estudio se necesitan tanto conceptos topológicos como algebraicos, lo que lo convierte en un proyecto atractivo al permitirnos trabajar con dos ramas de las matemáticas. Además, de los invariantes de nudos, el grupo de un nudo es uno de los más potentes, es decir, es uno de los que permite distinguir más nudos no equivalentes, razón por la cual pensamos que el trabajo puede ser de gran utilidad práctica.

Dado que el método que seguiremos para el cálculo de dicho invariante implica dibujar la proyección del nudo y hallar una serie de conjuntos y espacios asociados a esta, consideramos que la informática sería de gran utilidad para calcular computacionalmente la presentación superior e inferior del grupo de un nudo, y poder visualizar la proyección de dicho nudo y los espacios asociados a este, necesarios para calcular las presentaciones del grupo del nudo.

El conjunto de razones expuestas en esta sección fueron las que nos decantaron a realizar este interesante trabajo.

2. Objetivos

Los objetivos matemáticos fijados inicialmente en la propuesta del trabajo estaban relacionados con el estudio teórico de los nudos matemáticos. En el capítulo 2 estudiaremos el concepto de nudo, equivalencia de nudos, proyección de un nudo y el de invariante de un nudo. En el capítulo 3, analizaremos el concepto de grupo de un nudo, y describiremos el proceso para el cálculo de dos presentaciones del grupo de un nudo, a las que llamaremos presentación superior e inferior del grupo de un nudo. También, justificaremos el motivo por el cual dichas presentaciones, son presentaciones del grupo del nudo analizado. Todos estos conceptos que estudiaremos en la parte matemática del trabajo se sintetizarán en el capítulo 1, que será de utilidad para comprender la idea general del trabajo y para ser utilizado como esquema a la hora de implementar el programa que calcule los relatores de la presentación superior e inferior del grupo de un nudo. Además, en la Sección 2.4 se incluye una demostración personal de un teorema [Teorema 2.2] obtenido del libro *Introduction to Knot Theory* [CF77].

Por tanto, se han cumplido los objetivos matemáticos fijados en la propuesta del trabajo y para cumplirlos nos hemos basado principalmente en el libro *Introduction to Knot Theory* [CF77], sustentado por conceptos topológicos obtenidos mayoritariamente de los apuntes *Topología II, Apuntes de clase* [LF19].

Los objetivos informáticos fijados en la propuesta del trabajo estaban orientados a calcular

computacionalmente las presentaciones superior e inferior del grupo de un nudo arbitrario, y a representar dichos nudos, para posteriormente, debido a que el grupo de un nudo es un invariante de nudos, poder determinar la no equivalencia de dos nudos si dos presentaciones de los grupos de dichos nudos generan grupos no isomorfos. Para esto último, ya existen programas que realizan dicha funcionalidad, por lo que no la hemos incluido en nuestro programa.

Para cumplir con los objetivos fijados, en el capítulo 4, describiremos distintas formas de codificar un nudo, puesto que para que un ordenador identifique un determinado nudo, tenemos que asociarle a dicho nudo un conjunto de datos que lo distingan. En el capítulo 5, describiremos los métodos de una clase que hemos implementado en *Python*, llamada *Nudo*, la cual permite dibujar el diagrama de un nudo cualquiera mediante el módulo *Turtle* y calcular las presentaciones superior e inferior del grupo de dicho nudo, además de otras muchas funcionalidades. Dicha clase recibirá por parámetro la notación Dowker de un nudo, que es una de las codificaciones de nudos que estudiaremos en el capítulo 4.

En consecuencia, consideramos que se han cumplido también los objetivos informáticos fijados en la propuesta inicial. La información acerca de las diferentes codificaciones de los nudos se ha obtenido fundamentalmente del libro *The Knot Book* [Ada94] y del artículo *What are Gauss and Dowker-Thistlethwaite codes?* [Mil19].

Parte I.

Teoría de nudos

En esta primera parte realizaremos el estudio teórico de los nudos, vistos como objeto matemático. En el capítulo 1 haremos un resumen del proceso que seguiremos en los capítulos 2 y 3 hasta calcular la presentación superior e inferior del grupo de un nudo. Seguidamente, en el capítulo 2 definiremos más en profundidad los conceptos básicos relacionados con la teoría de nudos, para finalmente en el capítulo 3, poder calcular la presentación superior e inferior del grupo de un nudo, que al ser el grupo de un nudo un invariante de nudos, nos ayudarán a determinar si dos nudos son equivalentes o no.

1. Síntesis

En este capítulo definiremos los conceptos básicos de la teoría de nudos, hasta llegar a calcular lo que llamaremos la presentación superior del grupo de un nudo. El objetivo de este capítulo no será profundizar en los conceptos que se definan, ni justificar o demostrar los razonamientos utilizados hasta llegar al cálculo de dicha presentación, ya que este será el objetivo perseguido en los capítulos 2 y 3. En consecuencia, la finalidad de este capítulo será sintetizar el proceso de cálculo de dicha presentación, que además de servirnos como esquema del trabajo, será útil para la implementación informática que haremos en la siguiente parte de este.

El proceso que seguiremos para el cálculo de la presentación superior del grupo de un nudo se ha obtenido del libro *Introduction to Knot Theory* [CF77], así como muchos de los conceptos utilizados.

Como bien dice el título del trabajo, nos centraremos en el estudio matemático de los nudos. Obviamente, todo nudo (visto como objeto físico) puede transformarse en otro si las operaciones de atar y desatar están permitidas, de ahí que estas operaciones deban estar prohibidas porque si no, todos los nudos serían equivalentes. Una opción para evitar dichas operaciones es unir los extremos del nudo, y esto es lo que haremos a la hora de definir un nudo como objeto matemático. Por tanto, un nudo K es un subconjunto de \mathbb{R}^3 tal que existe un homeomorfismo de la circunferencia unidad $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$ a K . Por ejemplo, un nudo sería el conjunto representado en la Figura 1.1, también llamado nudo de ocho.

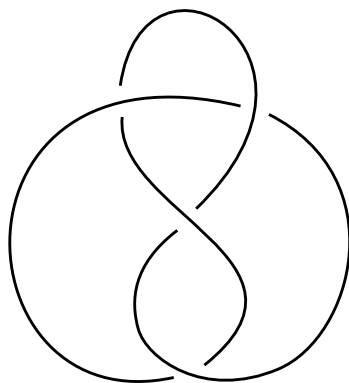


Figura 1.1.: Nudo de ocho.

1. Síntesis

Dados dos nudos K_1 y K_2 , diremos que son equivalentes si existe un homeomorfismo de \mathbb{R}^3 en sí mismo que aplique K_1 en K_2 . Teniendo ya la noción de equivalencia, si dos nudos son equivalentes van a existir determinadas propiedades que permanecerán igual para nudos equivalentes. A dichas propiedades las llamaremos invariantes de nudos, y dentro de dichos invariantes nos centraremos en el llamado grupo de un nudo (lo definiremos más adelante). La idea central del trabajo será estudiar este invariante de nudos, calculando las presentaciones del grupo de diversos nudos, que nos servirán para asegurar, en caso de que los grupos de dos nudos no sean isomorfos, que dichos nudos no son equivalentes.

Diferenciaremos los nudos en nudos admisibles (en los que nos centraremos) y no admisibles, siendo un nudo admisible un nudo equivalente a un nudo poligonal, que cuyo nombre indica, es aquel que está formado por la unión de un número finito de segmentos de recta, cuyos extremos son los vértices del nudo, como el de la [Figura 1.2](#). Un teorema nos dirá que, aparte de los nudos poligonales, si un nudo admite una parametrización regular de clase C^1 será admisible, lo que refleja que los nudos usuales serán admisibles, llegando a ser hasta complicado demostrar la existencia de los nudos no admisibles.

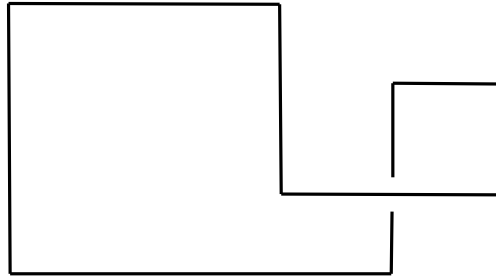


Figura 1.2.: Nudo poligonal.

Primeramente, asociaremos a un nudo K una determinada proyección ortogonal sobre un plano, que será crucial para el estudio de estos que haremos posteriormente. Dada una proyección ortogonal sobre un plano $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, diremos que un punto de la imagen, $p \in \mathcal{P}K$, es un punto múltiple [doble] si la imagen inversa, $\mathcal{P}^{-1}p$, contiene más de un punto [dos puntos] de K . Diremos que un nudo poligonal K está en posición regular respecto a una proyección ortogonal sobre un plano si los únicos puntos múltiples de $\mathcal{P}K$ son puntos dobles, habiendo un número finito de estos y si ningún punto doble es imagen de un vértice de K . Y así llegamos a un resultado interesante que nos dice que todo nudo poligonal K es equivalente, bajo una rotación cercana a la identidad de \mathbb{R}^3 a un nudo poligonal en posición regular respecto a una proyección ortogonal arbitraria.

Por ejemplo, dadas las proyecciones de dos nudos representadas en la [Figura 1.3](#), tenemos que el nudo de la [Figura 1.3a](#) está en posición regular respecto a dicha proyección, ya que solo contiene un punto múltiple y es un punto doble (representado con color rojo). En cambio, el nudo de la [Figura 1.3b](#) no está en posición regular respecto a dicha proyección ya que contiene un punto triple (representado con color verde).

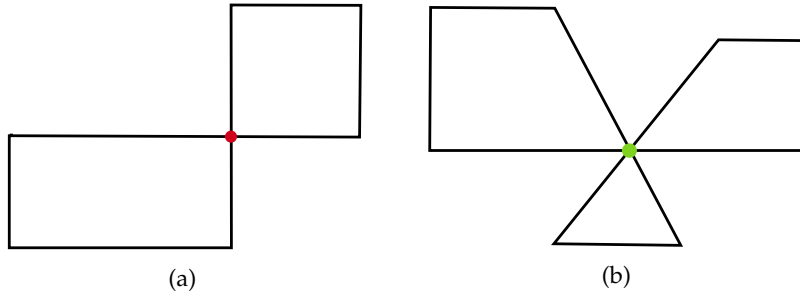


Figura 1.3.: (a) Nudo que está en posición regular. (b) Nudo que no está en posición regular.

Dado cualquier nudo admisible K , considerando un punto p_0 en $\mathbb{R}^3 \setminus K$ (que veremos que será irrelevante), llamaremos grupo de K al grupo fundamental $\Pi(\mathbb{R}^3 \setminus K, p_0)$. Durante el trabajo, trataremos de hallar dos presentaciones de dicho grupo, a las que llamaremos presentación superior e inferior, que nos servirán para determinar si los grupos de dos nudos admisibles son o no isomorfos. Para calcular dichas presentaciones del grupo de un nudo arbitrario K tendremos que seguir una serie de pasos. Veremos los pasos que hay que seguir para calcular la presentación superior, y será análogo el proceso para calcular la presentación inferior.

Dado que todo nudo poligonal K es equivalente, bajo una rotación cercana a la identidad de \mathbb{R}^3 , a un nudo poligonal en posición regular respecto a una proyección ortogonal arbitraria, consideraremos sin pérdida de generalidad que todo nudo poligonal está en posición regular respecto a la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ tal que $\mathcal{P}(x, y, z) = (x, y, 0)$.

En primer lugar, dado que habíamos dicho que todo nudo poligonal en posición regular respecto a una proyección, si tenía puntos múltiples tenían que ser puntos dobles, llamaremos punto superior a la preimagen por la aplicación \mathcal{P} , definida en el párrafo anterior, de un punto doble con coordenada z mayor, y punto inferior a la preimagen de dicho punto doble con coordenada z menor. Ahora, para algún número positivo n , seleccionamos un subconjunto Q de K conteniendo exactamente $2n$ puntos, de forma que divida el nudo poligonal K en dos clases de arcos segmentados y cerrados, los arcos superiores (que denotaremos por A_1, \dots, A_n , y su unión por A), y los inferiores (que denotaremos por B_1, \dots, B_n , y su unión por B). Esta subdivisión se debe hacer de forma que ningún arco superior contenga un punto inferior y ningún arco inferior contenga un punto superior, cumpliendo que el número de puntos de Q sea mínimo. Una vez dividido el nudo en arcos superiores e inferiores, tenemos que existe un homeomorfismo de \mathbb{R}^3 en sí mismo de forma que la imagen de $A \setminus Q$ se encuentra sobre el plano $\mathbb{R}^2 \times \{0\}$ y la imagen de $B \setminus Q$ se encuentra bajo dicho plano. En consecuencia, durante todo el trabajo consideraremos que $A \setminus Q$ está en $\{(x, y, z) \in \mathbb{R}^3: z > 0\}$, $B \setminus Q$ se encuentra en $\{(x, y, z) \in \mathbb{R}^3: z < 0\}$, y por tanto, $Q \subset \mathbb{R}^2 \times \{0\}$.

Por ejemplo, para el nudo de la [Figura 1.3a](#), que hemos dicho que está en posición regular respecto de dicha proyección, una posible división en arcos superiores e inferiores podría ser la realizada en la [Figura 1.4](#), donde los puntos de Q están señalados con color verde, el arco superior con color azul y el arco inferior con color rojo.

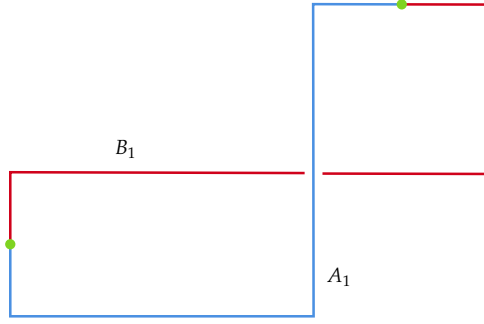


Figura 1.4.: Nudo dividido en arcos superiores e inferiores.

Dado un nudo poligonal K , una vez dividido en n arcos superiores y n inferiores para un número natural $n \in \mathbb{N}$, tomamos $x = \{x_1, \dots, x_n\}$, un conjunto formado por n símbolos que se encontrarán en biyección con los n arcos superiores del nudo poligonal, A_1, \dots, A_n . Además, para un nudo arbitrario K , cada presentación está hecha con respecto a una orientación de K y de \mathbb{R}^3 , por lo que una de las dos direcciones a través del nudo K es escogida como positiva y dibujaremos una flecha en K para representarla.

Dado un camino $a: [0, 1] \rightarrow \mathbb{R}^2 \times \{0\}$ y un nudo poligonal en posición regular K , diremos que a es simple si:

1. Es poligonal.
2. Ni el punto inicial del camino, $a(0)$, ni el final, $a(1)$, pertenecen a $\mathcal{P}K$.
3. El camino a intersecta a $\mathcal{P}K$ en tan solo un número finito de puntos, ninguno de los cuales es un vértice de a o un vértice de $\mathcal{P}K$, por lo que los cortes de dicho camino con $\mathcal{P}K$ deben ser transversales.

Definiremos el grupo libre $F[x]$ que tendrá como base libre las clases de equivalencia de x_1, \dots, x_n , y a cada camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ le asignaremos un elemento $a^\#$ de $F[x]$ tal que

$$a^\# = x_{i_1}^{\epsilon_1} \cdots x_{i_l}^{\epsilon_l},$$

donde las proyecciones de los arcos superiores que se cortan con el camino a son, en orden, $\mathcal{P}A_{i_1}, \dots, \mathcal{P}A_{i_l}$ (aquí se ve la biyección existente entre los símbolos x_i y los arcos superiores A_i , $i = 1, \dots, n$), y donde el valor de ϵ_k dependerá de la regla del tornillo que se introdujo para la orientación, por mantener la visión espacial:

- $\epsilon_k = 1$ si A_{i_k} y el camino a forman un tornillo de la mano derecha (orientación horaria).



Figura 1.5.: Tornillo de la mano derecha.

- $\epsilon_k = -1$ si A_{i_k} y el camino a forman un tornillo de la mano izquierda (orientación antihoraria).

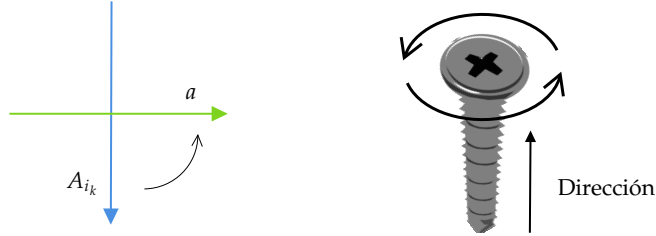


Figura 1.6.: Tornillo de la mano izquierda.

En la **Figura 1.7**, podemos ver un nudo donde con color azul están señalados los arcos superiores y con rojo los inferiores. También tenemos un camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{PB}$ representado con color verde, al que le vamos a asignar un elemento del grupo $F[x]$ según los criterios vistos anteriormente.

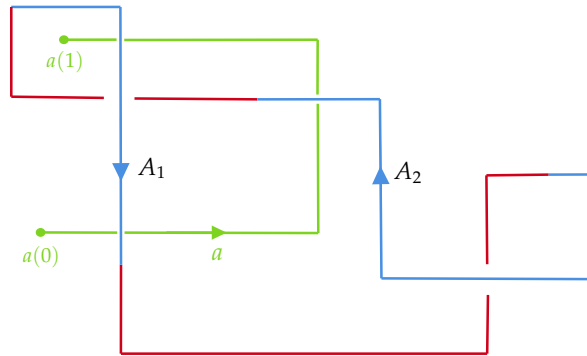


Figura 1.7.: Camino simple a .

En este caso $a^\# = x_1^{-1}x_2x_1$.

1. Síntesis

Para continuar con el estudio de los nudos, tomamos un punto $p_0 = (0, 0, z_0)$ para un número positivo z_0 que sea mayor que la coordenada z , en valor absoluto, de todos los puntos del nudo K . Dado un punto $p \in \mathbb{R}^2 \times \{0\}$, llamamos \bar{p} al camino que recorre el segmento que une el punto p_0 con el punto justo encima de p a la altura de p_0 , y después el segmento que une dicho punto al punto p . Una vez definido el camino \bar{p} para cualquier punto $p \in \mathbb{R}^2 \times \{0\}$, dado un camino simple a en $\mathbb{R}^2 \times \{0\}$, definimos el lazo

$$*a = \overline{a(0)} \cdot a \cdot \overline{a(1)}^{-1}.$$

Dado el nudo y el camino de la Figura 1.7, y tomando un punto p_0 verificando las condiciones mencionadas en el párrafo anterior, el lazo $*a$ sería el representado en la Figura 1.8.

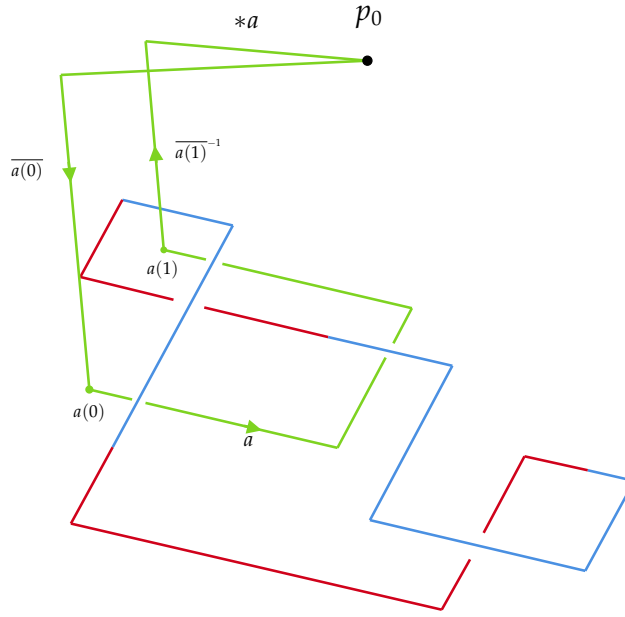


Figura 1.8.: Lazo $*a$.

Dado el grupo $F[x]$ mencionado anteriormente, estamos ya en posición de definir el homomorfismo $\phi: F[x] \rightarrow \Pi(\mathbb{R}^3 \setminus K, p_0)$. Para cada $j = 1, \dots, n$, tomamos a_j un camino simple cualquiera en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ que cumpla que $a_j^\# = x_j$. Definimos

$$\phi(x_j) = [*a_j] \quad \text{para todo } j = 1, \dots, n,$$

donde los corchetes indican la clase de equivalencia en $\mathbb{R}^3 \setminus K$ del lazo con base $p_0, *a_j$. El homomorfismo ϕ se puede extender a todo el grupo $F[x]$, siendo

$$\phi(a^\#) = [*a]$$

para cualquier camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$, y probaremos que dicho homomorfismo es sobreyectivo, o en otras palabras, que $\phi(x_1), \dots, \phi(x_n)$ genera $\Pi(\mathbb{R}^3 \setminus K, p_0)$.

Ahora vamos a seleccionar un punto $q_0 \in (\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}K$, que será de gran importancia

a la hora de calcular los relatores de la presentación superior del grupo del nudo. Seleccionamos también espacios disjuntos, abiertos y simplemente conexos V_1, \dots, V_n en $\mathbb{R}^2 \times \{0\}$, tal que $\mathcal{P}B_i \subset V_i$ para todo $i = 1, \dots, n$, y cuyos bordes son curvas de Jordan lineales a trozos, que serán recorridas en sentido antihorario (visto desde arriba) por los caminos v_1, \dots, v_n , que son simples en el sentido antes definido. Además, dichos espacios V_1, \dots, V_n , deben ser tomados de forma que q_0 no se encuentre en ninguno de ellos. Escogemos también caminos simples c_1, \dots, c_n tales que cada c_i tenga punto inicial q_0 y punto final $v_i(0)$, cumpliendo que $c_i(t) \in (\mathbb{R}^2 \times \{0\}) \setminus \bigcup_{k=1}^n \overline{V_k}$ excepto para $t = 1$ (ya que un camino está definido entre 0 y 1).

Para el nudo de la Figura 1.7 y de la Figura 1.8, los elementos definidos en el párrafo anterior podrían ser los señalados en la Figura 1.9.

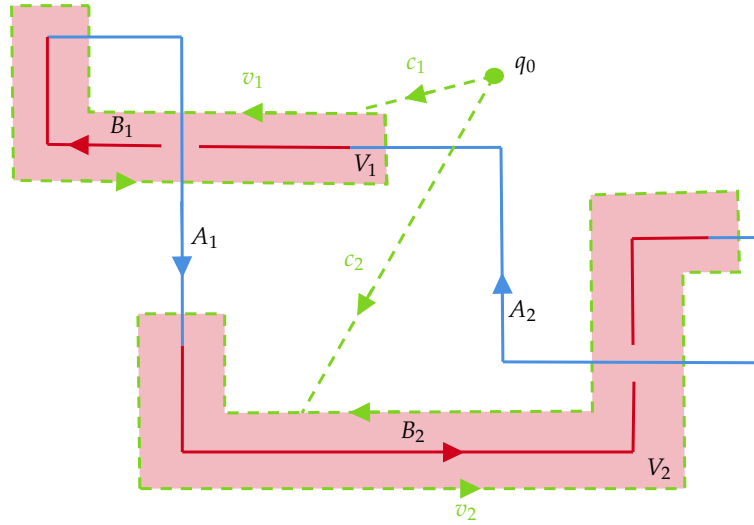


Figura 1.9.: Espacios V_1 y V_2 y caminos c_1, c_2, v_1 y v_2 .

Una vez ya definido el homomorfismo $\phi: F[x] \rightarrow \Pi(\mathbb{R}^3 \setminus K, p_0)$, los caminos simples c_1, \dots, c_n y los lazos v_1, \dots, v_n , tras varios pasos seremos capaces de obtener la presentación superior del grupo de un nudo arbitrario, que será

$$\langle x_1, \dots, x_n : r_1, \dots, r_n \rangle_\phi, \quad (1.1)$$

donde los relatores son de la forma $r_i = (c_i \cdot v_i \cdot c_i^{-1})^\#$ para todo $i = 1, \dots, n$, y ϕ es el homomorfismo definido anteriormente.

Por tanto, si consideramos el nudo K y los caminos c_1, c_2, v_1 y v_2 señalados en la Figura 1.9, los relatores de la presentación superior del grupo de K serían

$$\begin{aligned} r_1 &= x_1 x_1^{-1} x_1^{-1} x_2 = x_1^{-1} x_2, \\ r_2 &= x_2^{-1} (x_1 x_2 x_2^{-1} x_2^{-1}) x_2 = x_2^{-1} x_1. \end{aligned}$$

1. Síntesis

Por ello, la presentación superior del grupo de K sería

$$\langle x_1, x_2 : x_1^{-1}x_2, x_2^{-1}x_1 \rangle_\phi.$$

Sin embargo, un teorema nos dirá que cualquiera de los n relatores r_1, \dots, r_n se puede obtener de los $n - 1$ restantes, por lo que cualquiera de los relatores podrá ser borrado. Por esta razón, tenemos que una presentación del grupo de un nudo será

$$\langle x_1, \dots, x_n : r_1, \dots, \hat{r}_k, \dots, r_n \rangle_\phi,$$

donde \hat{r}_k indica el borrado del k -ésimo relator r_k , siendo k un entero cualquiera entre 1 y n .

En consecuencia, una presentación del grupo del nudo de la **Figura 1.9** será

$$\langle x_1, x_2 : x_1^{-1}x_2 \rangle_\phi,$$

dado que r_1 se puede obtener de r_2 o viceversa.

Por último, probaremos que la presentación

$$\langle x_1, \dots, x_n : v_1^\#, \dots, v_n^\# \rangle_\phi \tag{1.2}$$

es equivalente a la presentación (1.1) porque definen grupos isomorfos.

La ventaja principal respecto a la presentación superior (1.1) es que no hace falta calcular $c_1^\#, \dots, c_n^\#$, y también tenemos que cualquiera de los relatores $v_1^\#, \dots, v_n^\#$ es consecuencia de los $n - 1$ restantes de forma que uno de ellos puede ser omitido.

Por ejemplo, para el nudo de la **Figura 1.9**, los relatores de la presentación (1.2) del grupo de dicho nudo serán

$$v_1^\# = x_1 x_1^{-1} x_1^{-1} x_2 = x_1^{-1} x_2,$$

$$v_2^\# = x_1 x_2 x_2^{-1} x_2^{-1} = x_1 x_2^{-1}.$$

En consecuencia, la presentación (1.2) del grupo de dicho nudo será

$$\langle x_1, x_2 : x_1^{-1}x_2, x_1x_2^{-1} \rangle_\phi,$$

donde se puede eliminar uno de los dos relatores.

2. Nudos y Tipos de nudos

En este capítulo presentaremos con más profundidad que en el capítulo 1 los conceptos fundamentales de la teoría de nudos. Primeramente, definiremos matemáticamente un nudo y describiremos el concepto de equivalencia de dos nudos, que permitirá la clasificación de estos. Posteriormente, definiremos los nudos de cuyo estudio nos encargaremos, los nudos admisibles, y veremos que un nudo, normalmente, viene especificado por una determinada proyección ortogonal sobre un plano. Por último, estudiaremos unos invariantes básicos por los que se distinguen los diferentes tipos de nudos, siendo uno de ellos el grupo de un nudo, de cuyo estudio nos encargaremos en el capítulo 3.

Los conceptos definidos en este capítulo se han basado en el libro *Introduction to Knot Theory* [CF77], a excepción de una demostración en la Sección 2.4 que ha sido realizada de manera personal, y de la Sección 2.5.

2.1. Nudo como objeto matemático

Prácticamente todo el mundo se ha encontrado cara a cara con ciertos tipos de nudos, como el que forma el cordón de una zapatilla cuando lo atas o el que forma un cable cuando se enreda. El problema que se nos plantea es mostrar matemáticamente si dos nudos son distintos entre sí, o si por el contrario, son equivalentes.

Para probar matemáticamente lo anterior, previamente debemos definir un nudo como objeto matemático, y no como objeto físico que es como lo solemos conocer. En primer lugar, proporcionaremos la definición matemática de nudo que debe definir objetos matemáticos que se aproximen a los objetos físicos que conocemos. Posteriormente, explicaremos cuando dos nudos se pueden considerar el mismo, es decir, cuando son equivalentes.

Obviamente, si consideramos dos nudos arbitrarios tal y como los conocemos, se podrían transformar el uno en el otro si las operaciones de atar y desatar estuvieran permitidas. Por ejemplo, los llamados nudo simple y nudo de ocho, si estuvieran permitidas dichas operaciones, se podrían transformar el uno en el otro, como se ve en la Figura 2.1.

2. Nudos y Tipos de nudos

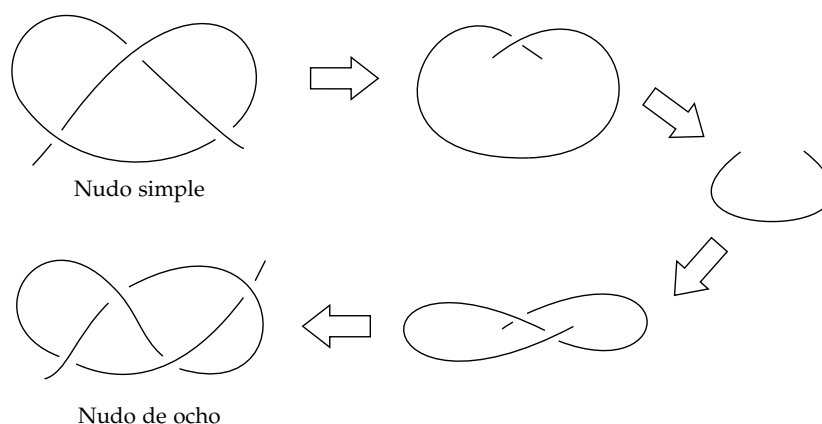


Figura 2.1.: Transformación del nudo simple en el nudo de ocho.

Sin embargo, veremos posteriormente que resultarán ser nudos distintos. Por lo tanto, tenemos dos opciones, considerar estas operaciones como prohibidas a la hora de determinar si dos nudos son equivalentes, o establecer una definición de nudo que no permita las acciones previas. Optaremos por la segunda opción, uniendo los extremos del nudo entre sí, ya que posteriormente veremos que la distinción matemática de los nudos será más sencilla con esta nueva definición.

Para la definición de nudo utilizaremos el concepto de homeomorfismo.

Definición 2.1. Un **homeomorfismo** es una aplicación entre dos espacios topológicos que cumple con ser una aplicación biyectiva, y que tanto ella como su inversa son continuas. Consecuentemente, se dice que dos espacios topológicos son homeomorfos si existe un homeomorfismo entre ellos.

De tal forma, ya estamos en posición de definirlo formalmente.

Definición 2.2. Un **nudo** K es un subconjunto de \mathbb{R}^3 tal que existe un homeomorfismo de la circunferencia unidad C a K , considerando la circunferencia unidad $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$.

En consecuencia, tenemos una amplia gama de subconjuntos de \mathbb{R}^3 que serán nudos, aunque muchos de ellos serán difíciles de describir matemáticamente, como el representado en la [Figura 2.2](#)

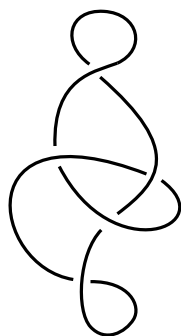


Figura 2.2.: Nudo.

2.2. Nudos equivalentes

Ya que hemos definido formalmente un nudo, nos planteamos la cuestión acerca de cuando dos nudos, K_1 y K_2 , deben considerarse el mismo, es decir, cuando son equivalentes. Vemos que no se puede confundir con ser o no homeomorfos, ya que, por la definición de nudo, dos nudos arbitrarios son homeomorfos a la circunferencia unidad, y por tanto, son homeomorfos entre sí. Por ejemplo, tenemos que por el hecho de ser nudos, tanto el nudo trébol como el nudo de ocho son homeomorfos a la circunferencia unidad, como se muestra en la Figura 2.3a y en la Figura 2.3b respectivamente.

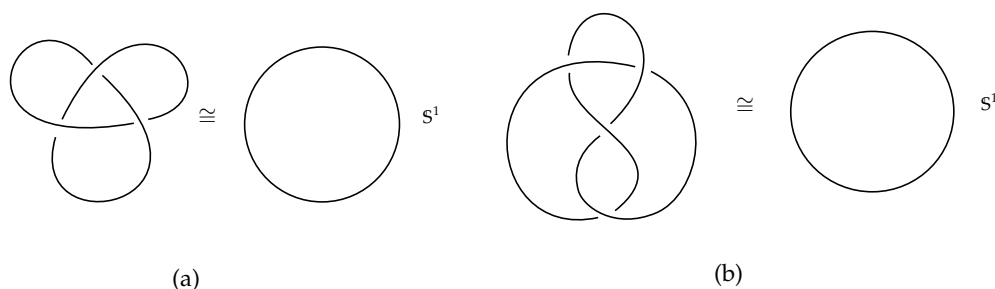


Figura 2.3.: (a) Nudo trébol. (b) Nudo de ocho.

En consecuencia, obtenemos que el nudo trébol y el nudo de ocho son homeomorfos, por lo que no podemos establecer como criterio para determinar la equivalencia de dos nudos el hecho de que estos sean homeomorfos o no.

Para la distinción de nudos, por lo visto anteriormente, no utilizaremos las propiedades topológicas del espacio formado por los puntos del nudo, sino que utilizaremos la forma en la que dicho espacio es embebido en \mathbb{R}^3 .

Definición 2.3. Dos nudos K_1 y K_2 son **equivalentes** si existe un homeomorfismo de \mathbb{R}^3 en sí mismo que aplica K_1 en K_2 .

La relación de “ser equivalente a” es una relación de equivalencia, dos nudos equivalentes se dice que son del mismo *tipo* y cada clase de equivalencia es un *tipo de nudo*. Aquellos nudos

2. Nudos y Tipos de nudos

que son equivalentes a la circunferencia $C = \{(x, y, 0) : x^2 + y^2 = 1\}$ se llaman *triviales* y pertenecen al tipo trivial. Un ejemplo de nudo trivial sería el nudo representado en la [Figura 2.4](#), ya que existe un homeomorfismo de \mathbb{R}^3 en sí mismo, que aplica la circunferencia C en ese nudo.

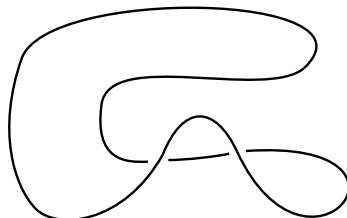


Figura 2.4.: Nudo trivial.

2.3. Nudos admisibles y no admisibles

Una vez estudiada la definición de nudo y la relación de equivalencia entre estos, nos centraremos en los nudos en los cuales focalizaremos nuestro estudio, los nudos admisibles, debido a los invariantes del tipo de nudo que tienen dichos nudos, además de que representan a los nudos usuales, como veremos más adelante. Previamente, definiremos el concepto de nudo poligonal, mediante el cual podremos definir el ya mencionado concepto de nudo admisible.

Definición 2.4. Un nudo **poligonal** es aquel que está formado por la unión de un número finito de segmentos de recta llamados aristas, cuyos extremos son los vértices del nudo.

Un nudo poligonal podría ser el representado en la [Figura 2.5](#).

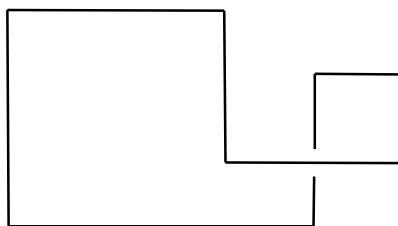


Figura 2.5.: Nudo poligonal.

Dada la definición anterior ya podemos definir el concepto de nudo admisible.

Definición 2.5. Un nudo se dice que es **admisible** si es equivalente a un nudo poligonal. En caso contrario diremos que el nudo es no admisible.

Por lo tanto debemos preguntarnos qué nudos, aparte de los poligonales, son admisibles. Gran parte de la respuesta nos la da el teorema siguiente.

Teorema 2.1. Si un nudo admite una parametrización regular de clase C^1 , es admisible.

Este teorema refleja que los nudos usuales serán admisibles, llegando a ser hasta complicado demostrar la existencia de los nudos no admisibles.

2.4. Proyecciones de nudos

Normalmente, un nudo K está especificado por una determinada proyección ortogonal sobre un plano. Por ejemplo, la proyección ortogonal sobre el plano $\mathbb{R}^2 \times \{0\}$ sería la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ definida por

$$\mathcal{P}(x, y, z) = (x, y, 0).$$

Dada una proyección ortogonal sobre un plano, $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, un punto de la imagen $p \in \mathcal{P}K$ es llamado *punto múltiple* si la imagen inversa $\mathcal{P}^{-1}p$ contiene más de un punto de K , y se le llama *orden* de $p \in \mathcal{P}K$ a la cardinalidad de $(\mathcal{P}^{-1}p) \cap K$. Por lo tanto, llamaremos punto doble a un punto múltiple de orden 2, punto triple a un punto múltiple de orden 3, etc. Antes de centrarnos en las utilidades de estos conceptos, vamos a ver algunos ejemplos.

El nudo de la [Figura 2.5](#) tiene un único punto múltiple, concretamente de orden dos. En cambio, el nudo de la [Figura 2.6](#) tiene dos puntos múltiples, uno de orden 2 (representado con color rojo) y otro de orden 3 (representado con color verde).

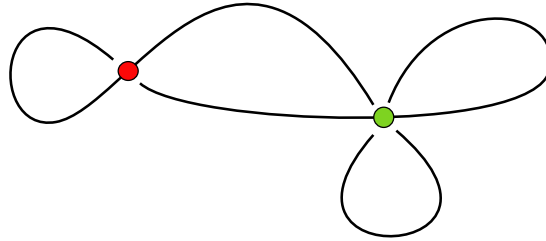


Figura 2.6.: Nudo con dos puntos múltiples.

Normalmente, la imagen $\mathcal{P}K$ puede ser complicada en cuanto al número y tipo de puntos múltiples, sin embargo, es posible que ese nudo K sea equivalente a otro cuya proyección sea más simple. Para la proyección de un nudo poligonal arbitrario, el criterio para ser considerada simple es que, el nudo esté en lo que se considera una posición regular respecto a una proyección ortogonal sobre un plano, que se define a continuación.

Definición 2.6. Un nudo poligonal está en **posición regular** respecto a una proyección ortogonal sobre un plano si:

- I. Los únicos puntos múltiples de K son puntos dobles, y hay un número finito de estos.
- II. Ningún punto doble es imagen de un vértice de K .

2. Nudos y Tipos de nudos

Por tanto, dada la proyección ortogonal del nudo de la [Figura 2.5](#), ese nudo poligonal está en posición regular respecto a dicha proyección, ya que solo tiene un punto múltiple que es de orden dos, y además, ese punto doble no es imagen de un vértice. Por otro lado, dadas las proyecciones de los nudos representadas en la [Figura 2.7](#), el nudo de la [Figura 2.7a](#) no está en posición regular respecto a dicha proyección, ya que contiene un punto triple (representado con color verde), y el nudo de la [Figura 2.7b](#) tampoco está en posición regular respecto a dicha proyección ya que el único punto doble que contiene (representado con color rojo) es imagen de un vértice del nudo K .

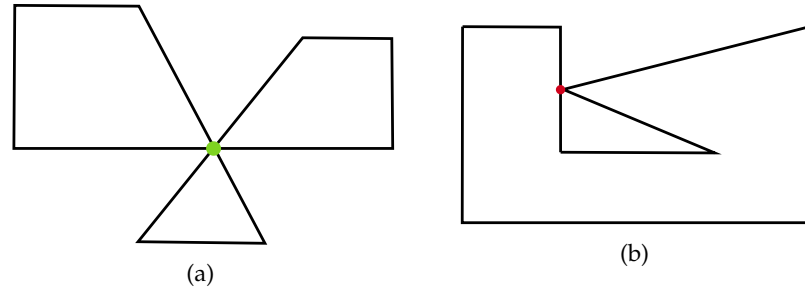


Figura 2.7.: (a) Punto triple. (b) Punto doble imagen de un vértice.

Cada punto doble de la imagen de la proyección de un nudo poligonal en posición regular es la imagen de dos puntos del nudo. Aquel punto con coordenada z mayor se llamará **punto superior**, y el otro, con coordenada z menor, es el correspondiente **punto inferior**. Veremos la importancia de estos conceptos en el capítulo siguiente.

Vamos a enunciar un teorema transcendental en el estudio de los nudos, que provocará que para estudiar los nudos admisibles, que son el principal objeto de estudio en este trabajo, nos sea suficiente con estudiar los nudos poligonales en posición regular respecto a una proyección ortogonal arbitraria sobre un plano. Para la demostración de dicho teorema se ha seguido un enfoque distinto al que se sigue en el libro *Introduction to Knot Theory* [CF77], donde nos hemos ayudado del homeomorfismo existente entre el plano proyectivo \mathbb{P}^2 y el espacio $Proj$ de todas las proyecciones ortogonales sobre planos que pasan por el origen para demostrar dicho teorema.

Teorema 2.2. *Todo nudo poligonal K es equivalente, bajo una rotación cercana a la identidad de \mathbb{R}^3 , a un nudo poligonal en posición regular respecto a una proyección ortogonal sobre un plano arbitraria.*

Demostración. La idea geométrica es mantener K fijo y mover la proyección. Cada conjunto de rectas paralelas en \mathbb{R}^3 determina una única proyección ortogonal de \mathbb{R}^3 sobre el plano perpendicular al conjunto de rectas paralelas, que pasa por el origen (el plano).

Por ejemplo, en la [Figura 2.8](#) se puede ver la proyección de un punto $q \in \mathbb{R}^3$ sobre un plano perpendicular al conjunto de rectas con vector director v que contiene al origen.

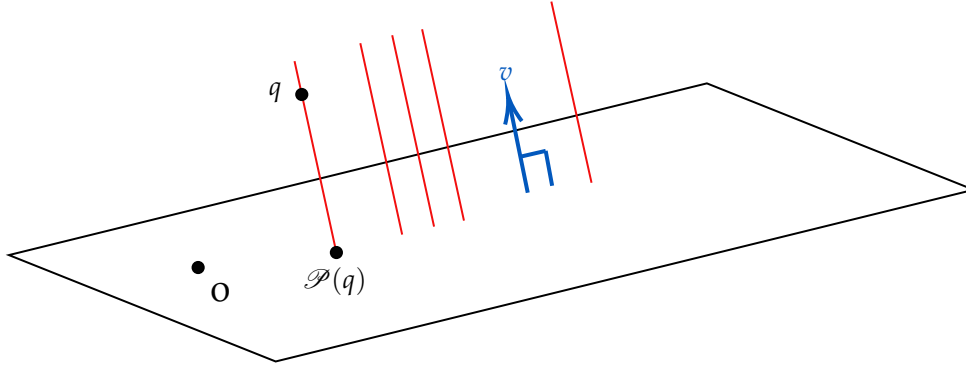


Figura 2.8.: Proyección ortogonal sobre un plano.

Si consideramos la aplicación $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ definida como

$$\mathcal{P}(p) = p - \langle p, v \rangle v, \quad v \in S^2,$$

siendo v el vector director unitario del haz de rectas, la proyección de un punto cualquiera $q \in \mathbb{R}^3$ será $\mathcal{P}(q)$. Vemos que la aplicación \mathcal{P} queda definida completamente por el vector $v \in S^2$, produciendo los vectores v y $-v$ la misma proyección. Denotaremos al espacio de todas las proyecciones ortogonales sobre planos que pasan por el origen como $Proy$. Tenemos que el espacio $Proy$ es un espacio topológico, por ser un subconjunto de los endomorfismos de \mathbb{R}^3 , es decir, $Proy \subset End(\mathbb{R}^3)$. Gracias a estas últimas observaciones, podemos afirmar que la regla

$$\mathbb{P}^2 \rightarrow Proj, \quad [v] = \{v, -v\} \mapsto \mathcal{P}$$

define un homeomorfismo entre el plano proyectivo obtenido por la identificación de los puntos antípoda de la esfera unidad S^2 y el espacio $Proj$ de todas las proyecciones ortogonales sobre planos que pasan por el origen. Es decir, hay una asociación biyectiva y bicontinua entre las proyecciones descritas anteriormente ($Proj$) y el plano proyectivo \mathbb{P}^2 . Para simplificar la demostración, trabajaremos en el espacio \mathbb{P}^2 .

Para esta demostración utilizaremos el concepto de conjunto diseminado. Un subconjunto A de un espacio topológico X se dice **denso en ninguna parte**, o diseminado en X si el interior de su clausura es vacío. Es decir, A es diseminado en X si y solo si $\overset{\circ}{\bar{A}} = \emptyset$ en X . La idea de la demostración será ver que el conjunto de los puntos del proyectivo que tengan asociados vectores directores para los cuales el nudo K no esté en posición regular con respecto a la proyección ortogonal asociada a ese vector director sea un subconjunto diseminado de \mathbb{P}^2 . De esta forma, el complemento de ese subconjunto será un conjunto denso en \mathbb{P}^2 , por lo que si un nudo K no está en posición regular respecto a una determinada proyección, bajo una rotación cercana a la identidad de \mathbb{R}^3 dicho nudo ya sí estará en posición regular respecto a esa proyección.

Sea Q el conjunto de puntos de \mathbb{P}^2 que corresponden a proyecciones con respecto a las cuales K no está en posición regular. A continuación, vamos a demostrar que dicho conjunto Q no es denso en ninguna parte de \mathbb{P}^2 , y de esta forma deduciremos que hay una proyección \mathcal{P}_0 respecto de la cual, K está en posición regular y está arbitrariamente cerca de la

2. Nudos y Tipos de nudos

proyección original \mathcal{P} . Cualquier rotación de \mathbb{R}^3 que aplique la recta $\mathcal{P}_0^{-1}(0,0,0)$ en la recta que pasa por el origen cuyo vector director es el vector normal del plano sobre el que se proyecta, será suficiente para completar la demostración.

Para probar que Q no es denso en ninguna parte de \mathbb{P}^2 , consideramos primero el conjunto de todas las rectas que pasan por un vértice y por los puntos de una arista de K . Si tomamos p y σ , un vértice y una arista respectivamente de K , tenemos dos posibilidades:

- $p \in \sigma$: En este caso, será única la recta que pase por el vértice p y por los puntos de la arista σ y dicha recta estará definida por el vector director de la arista σ . Dicha recta determinará una única proyección ortogonal, que en el plano proyectivo \mathbb{P}^2 será un punto.



Figura 2.9.: $p \in \sigma$.

- $p \notin \sigma$: En esta situación, tendremos un haz de rectas que pasan por el vértice p y por los puntos de la arista σ , como se puede ver en la [Figura 2.10](#). Los vectores directores de ese haz de rectas determinarán un haz de proyecciones, que en el plano proyectivo \mathbb{P}^2 será un arco cerrado.

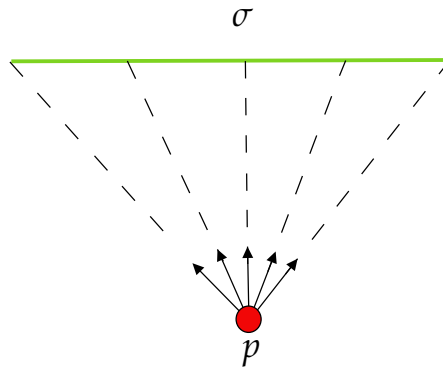


Figura 2.10.: $p \notin \sigma$.

A esa unión finita de arcos y puntos en \mathbb{P}^2 la denotaremos por Q_1 , siendo este número de puntos y rectas finito, ya que al ser K un nudo poligonal tiene un número finito de vértices, y por tanto, un número finito de aristas.

Observamos que, como las dos situaciones anteriores son los dos únicos escenarios posibles, en \mathbb{P}^2 estamos eliminando conjuntos homeomorfos a arcos cerrados o puntos, por tanto, la imagen de dichos arcos y puntos en $Proj$ será compacta y de interior vacío, por lo que la imagen de $\mathbb{P}^2 \setminus Q_1$ será abierta y densa como queríamos.

Ya que el plano proyectivo \mathbb{P}^2 puede ser descrito por ser los puntos de la esfera S^2 donde cada punto y su antípoda pertenecen a la misma clase de equivalencia (son indistinguibles),

en la **Figura 2.11** se puede ver un ejemplo de dichos puntos y arcos de \mathbb{P}^2 descritos anteriormente representados en la esfera S^2 , y se aprecia de manera más intuitiva que $\mathbb{P}^2 \setminus Q_1$ es denso.

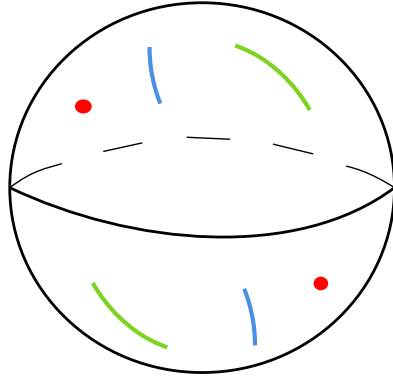


Figura 2.11.: Puntos y arcos en \mathbb{P}^2 .

En consecuencia, cualquier proyección asociada a un punto de $\mathbb{P}^2 \setminus Q_1$ debe satisfacer la condición (II) de la definición de nudo en posición regular.

Queda por probar que los puntos múltiples de orden $n \geq 3$ pueden ser “evitados”, que se prueba como sigue. Consideremos tres rectas r, s y t cada una de las cuales contiene una arista de K y vamos a hallar las rectas que se intersectan con estas tres. Por ejemplo, si tres aristas arbitrarias de un nudo K son los conjuntos señalados con color negro en la **Figura 2.12**, tomamos las siguientes rectas r, s y t .

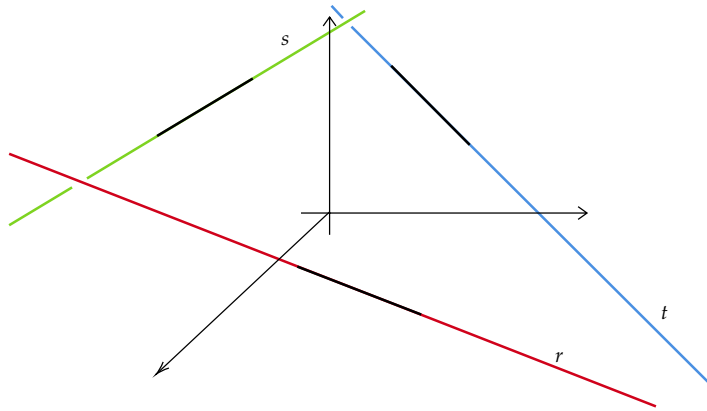


Figura 2.12.: Tres rectas conteniendo a tres aristas del nudo K .

Dado que la posición relativa de las tres rectas puede ser variable en función de la disposición de las aristas, se puede dar una gran variedad de posibilidades, que debemos estudiar por separado. Las diferentes posiciones relativas de tres rectas en \mathbb{R}^3 que se pueden dar son:

1. Dos rectas paralelas y
 - a) la otra paralela a ellas encontrándose las tres en el mismo plano.

2. Nudos y Tipos de nudos

- b) la otra paralela a ellas no encontrándose las tres en el mismo plano.
- c) la otra recta intersectando a una de ellas.
- d) la otra recta intersectando a las dos rectas paralelas.
- e) la otra cruzándose con ellas, siendo esta recta, paralela al plano que contiene a las otras dos rectas.
- f) la otra cruzándose con ellas, siendo esta recta, secante al plano que contiene a las otras dos rectas.

2. Dos rectas se intersecten y

- a) la otra se cruce con ellas, siendo esta recta, paralela al plano que contiene a las otras dos rectas.
- b) la otra se cruce con ellas, siendo esta recta, secante al plano que contiene a las otras dos rectas .
- c) la otra recta se intersecte con las dos primeras.
- d) la otra recta se intersecte con solo una de ellas.

3. Las tres rectas se crucen.

Estos son todos los posibles escenarios que se pueden dar y de los cuales queremos obtener el conjunto de vectores directores de las rectas que intersectan a tres rectas cualesquiera en dichas posiciones relativas. Aunque veremos que algunos de los casos se pueden estudiar de forma conjunta, en otros el conjunto de vectores directores de las rectas que intersectan a las tres rectas r, s y t se obtendrá de una forma particular.

(1) En el caso en que dos rectas sean paralelas, por ejemplo t y s , existirá un plano que contenga a dichas rectas. Por tanto, todos los vectores directores de las rectas que intersecten a dichas rectas paralelas t y s estarán en un mismo plano vectorial de \mathbb{R}^3 , por lo que estarán contenidos en una recta proyectiva, que es un conjunto cerrado de interior vacío en \mathbb{P}^2 . En consecuencia, al añadir una recta más, en este caso la recta r , es independiente su posición relativa con respecto a las rectas s y t , ya que el conjunto de los vectores directores de las rectas que intersecten a las rectas r, t y s , estará obviamente contenido en el conjunto de los vectores directores de las rectas que intersectan a las rectas t y s , ya que el segundo conjunto es menos restrictivo. Vamos a analizarlo geométricamente viendo algunos de los casos posibles del escenario (1).

(1a) En este caso tenemos tres rectas paralelas, r, s y t cada una de las cuales contiene una arista del nudo K y se encuentran en el mismo plano, al que llamaremos $\Pi_{s,t}$, como podemos ver en la [Figura 2.13](#).

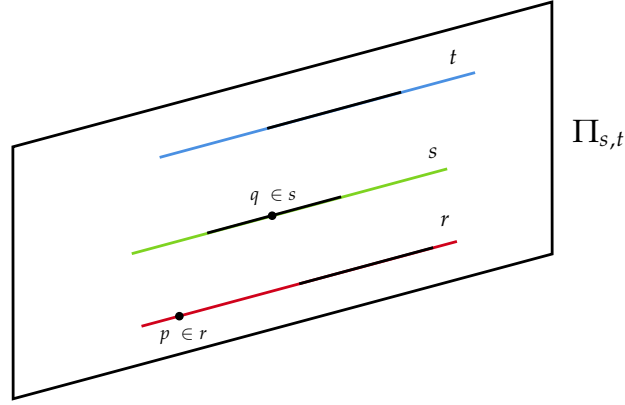


Figura 2.13.: Tres rectas paralelas en el mismo plano.

Los vectores directores unitarios de las rectas que intersectan a r, s y t serán los vectores directores unitarios del haz de rectas que pasa por un punto cualquiera $p \in r$ y por los diferentes puntos q de la recta s (dichas rectas también intersectarán a t por ser las tres rectas paralelas y pertenecer al mismo plano), que estarán contenidos en un mismo plano vectorial de \mathbb{R}^3 .

(1b) En el caso de tener tres rectas paralelas dos a dos, r, s y t que no pertenezcan al mismo plano, ningún punto de la recta r pertenecerá al plano $\Pi_{s,t}$. Dado que todas las rectas que intersectan a s y t están contenidas en el plano $\Pi_{s,t}$ no existirá ninguna recta que intersecte a r, s y t , como se puede apreciar en la Figura 2.14.

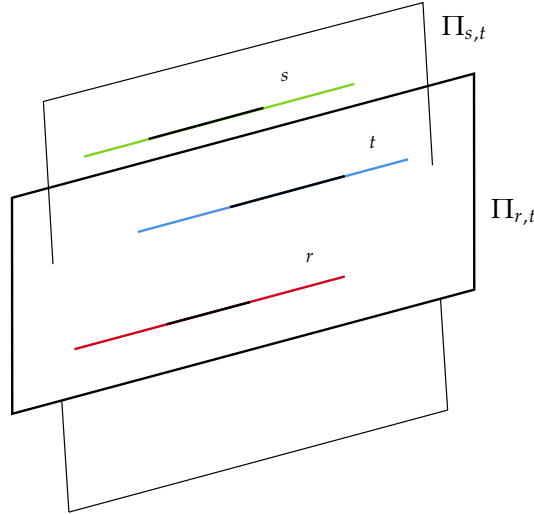


Figura 2.14.: Tres rectas paralelas no pertenecientes al mismo plano.

El resto de casos del escenario (1) se analizarían de forma similar a los dos casos ya estudiados, dando como resultado lo dicho anteriormente, que los vectores directores de las rectas que intersectan a dichas rectas r, t y s estarán en un mismo plano vectorial de \mathbb{R}^3 , por

2. Nudos y Tipos de nudos

lo que estarán contenidos en una recta proyectiva, que es un conjunto cerrado de interior vacío en \mathbb{P}^2 . Por ejemplo, el caso (1e) es muy similar al caso (1b), donde al no existir ningún punto de la recta r contenido en el plano $\Pi_{s,t}$, no existirá ninguna recta que intersecte a r, s y t , como podemos ver en la Figura 2.15.

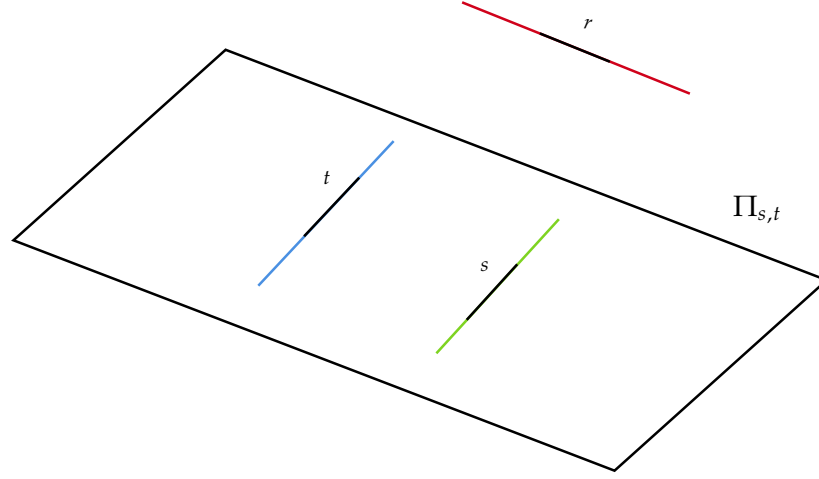


Figura 2.15.: Dos rectas paralelas y otra que se cruza, paralela al plano $\Pi_{s,t}$.

Vamos a pasar a analizar los casos del escenario (2).

Supongamos que tan solo dos de las tres rectas se intersectan, por ejemplo r y t , y la recta s se cruza con estas dos. Tomamos el plano que contiene a la recta r y t , que lo llamaremos $\Pi_{r,t}$, y tenemos que como la recta s no se intersecta con r ni con t no puede estar contenida en $\Pi_{r,t}$. Por tanto, tenemos dos posibilidades, que la recta s y el plano $\Pi_{r,t}$ sean paralelos o secantes.

(2a) En el caso de que sean paralelos tomamos el punto de corte de las rectas r y t , $\{p\} = r \cap t$. Para cada $q \in s$ tenemos que la recta que pasa por q y p tiene a $p - q$ como vector director y obviamente intersecta a las tres rectas, como podemos ver en la Figura 2.16.

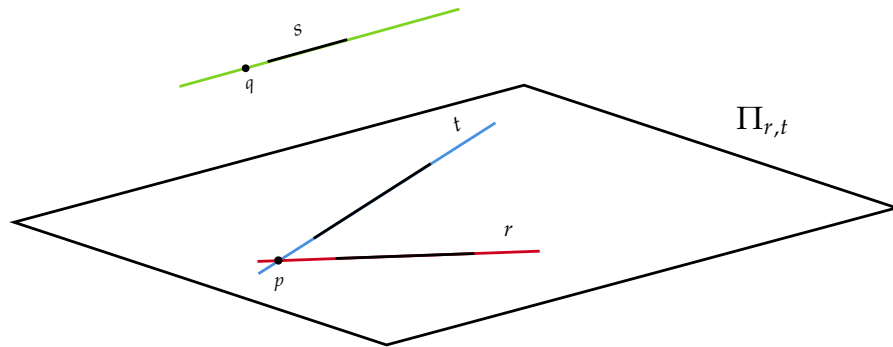


Figura 2.16.: Dos rectas secantes y la otra paralela al plano $\Pi_{r,t}$.

Es más, estas rectas serán las únicas que intersectan a r, s y t , y por tanto, todas estas rectas estarán contenidas en el plano que contiene a la recta s y al punto p . En consecuencia,

los vectores directores de esas rectas estarán en un mismo plano vectorial de \mathbb{R}^3 , por lo que estarán contenidos en una recta proyectiva que es un conjunto cerrado de interior vacío en \mathbb{P}^2 .

(2b) En el caso en el que la recta s y el plano $\Pi_{r,t}$ sean secantes, tomamos el punto de corte de las rectas r y t , $\{p\} = r \cap t$. Para cada $q \in s$ tenemos que la recta que pasa por q y p tiene a $p - q$ como vector director y obviamente intersecta a las tres rectas. Además, si tomamos el punto de corte de la recta s y el plano $\Pi_{r,t}$, $\{x\} = s \cap \Pi_{r,t}$, tenemos que toda recta que intersecte a r, s y t y que no pase por el punto p , tendrá que pasar por el punto x , como se puede apreciar en la [Figura 2.17](#).

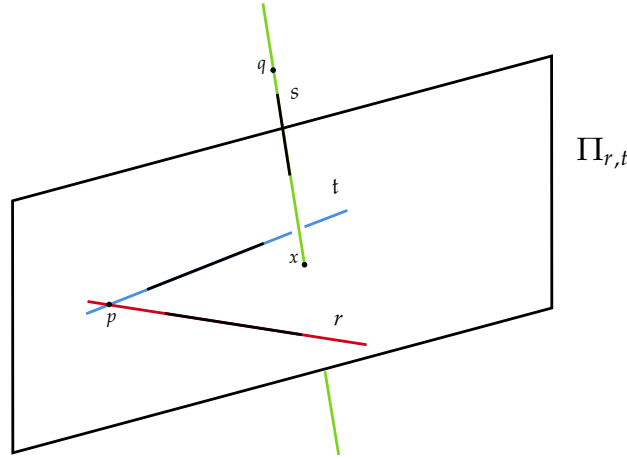


Figura 2.17.: Dos rectas secantes y la otra secante al plano $\Pi_{r,t}$.

Por consiguiente, las rectas que intersecten a r, s y t tendrán que estar contenidas en la unión del plano que contiene al punto q y a la recta s y del plano $\Pi_{r,t}$. En consecuencia, los vectores directores de esas rectas estarán en dos planos vectoriales de \mathbb{R}^3 , por lo que estarán contenidos en la unión de dos rectas proyectivas que es un conjunto cerrado de interior vacío en \mathbb{P}^2 , al ser una unión finita de conjuntos cerrados de interior vacío en \mathbb{P}^2 .

En el caso en que r y s sean secantes y r y t también lo sean, pueden darse dos situaciones, que las tres rectas pertenezcan al mismo plano, o que no.

(2c) En caso de pertenecer al mismo plano, las rectas que intersecten con r, s y t tendrán que estar contenidas en el plano que las contiene, que lo llamaremos $\Pi_{r,s}$, al igual que ocurría en el escenario (1). Esto se puede ver en la [Figura 2.18](#).

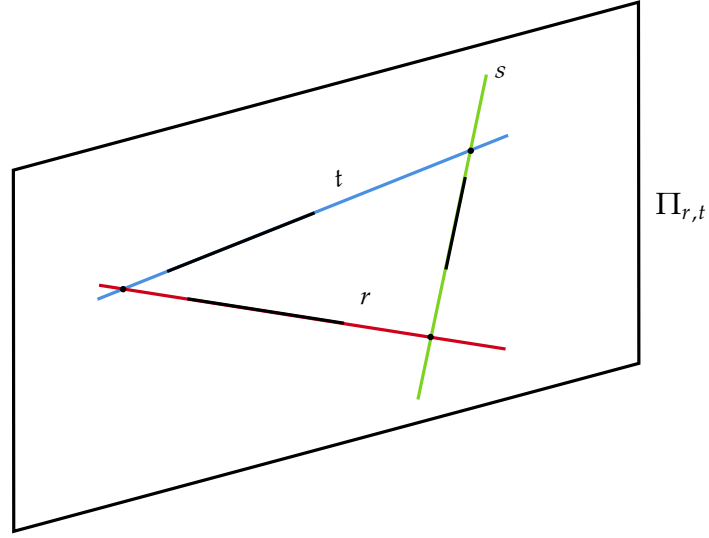


Figura 2.18.: Tres rectas secantes dos a dos.

Por tanto, los vectores directores de esas rectas estarán en un mismo plano vectorial de \mathbb{R}^3 , por lo que estarán contenidos en una recta proyectiva que es un conjunto cerrado de interior vacío en \mathbb{P}^2 .

(2d) También tenemos el caso de que r y s sean secantes, al igual que r y t , mientras que s y t sean rectas que se cruzan. Consideramos los puntos de intersección de las rectas r, s y t , que los llamaremos $\{p_1\} = r \cap s$ y $\{p_2\} = r \cap t$, y tenemos que para cada $q_1 \in t$ y $q_2 \in s$, la recta que pasa por p_i y q_i que tiene vector director $p_i - q_i$ interseca a las tres rectas para $i = 1, 2$. Además, dichos haces de rectas son los únicos que intersecan a r, s y t , como se puede ver en la Figura 2.19.

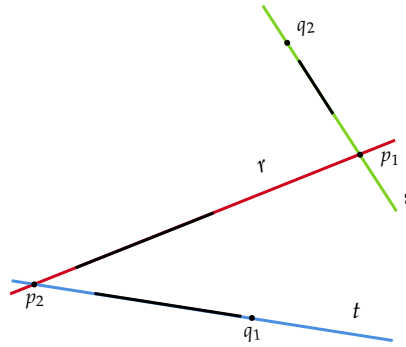


Figura 2.19.: r y t secantes, r y s secantes y s y t se cruzan.

Estas son las dos únicas posibilidades que existen, por lo que las rectas que intersecan a r, s y t van a estar contenidas en la unión del plano que contiene a p_2 y a la recta s y del plano que contiene a p_1 y a la recta t . En consecuencia, los vectores directores de esas rectas estarán en dos planos vectoriales de \mathbb{R}^3 , por lo que estarán contenidos en la unión de dos

rectas proyectivas, que es un conjunto cerrado de interior vacío en \mathbb{P}^2 , al ser una unión finita de conjuntos cerrados de interior vacío en \mathbb{P}^2 .

(3) El último caso que nos queda es el caso en el que las tres rectas r, s y t se cruzan dos a dos, es decir, el caso en el que ningún par de esas tres rectas pertenezcan a un mismo plano. De esta forma, tomamos un punto arbitrario $p \in r$ y consideramos el plano que contiene a p y s , que llamaremos Π_p . Nos interesa el caso en el que el plano Π_p y la recta t se intersecten, por lo tanto, dado que r, s y t son tres rectas que se cruzan, será único el punto $p_{1*} \in r$ para el que $\Pi_{p_{1*}}$ y t son paralelas. En consecuencia tomaremos $p \in r \setminus \{p_{1*}\}$.

Por ejemplo, tenemos las siguientes tres rectas que se cruzan,

$$\begin{aligned} r &\equiv \{\lambda(-1, 0, 1) : \lambda \in \mathbb{R}^3\}, \\ s &\equiv \{(0, 1, 0) + \lambda(1, 0, 0) : \lambda \in \mathbb{R}^3\}, \\ t &\equiv \{(0, 0, 1) + \lambda(0, 1, 0) : \lambda \in \mathbb{R}^3\}, \end{aligned}$$

y tomamos el punto $p = (0, 0, 0) \in r$, representados en **Figura 2.20**.

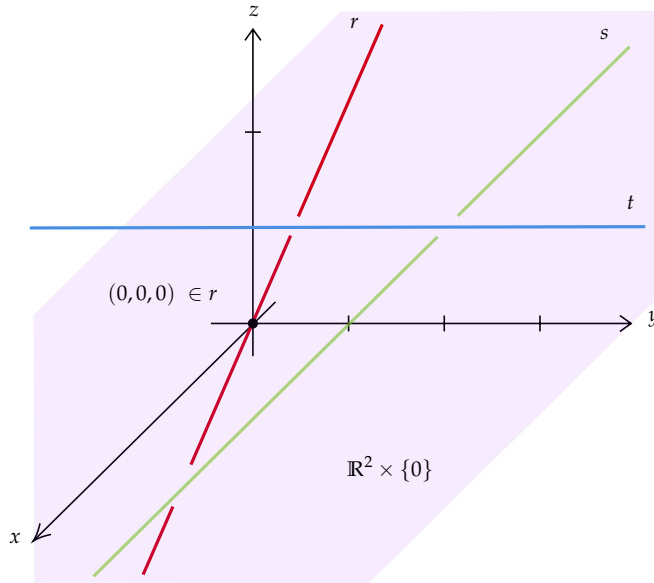


Figura 2.20.: Ejemplo de tres rectas que se cruzan y $p \in r$.

El plano que contiene al punto p y a la recta s será el plano $\mathbb{R}^2 \times \{0\}$, representado con color lila en la **Figura 2.20**, que sabemos que es paralelo a la recta t , ya que el vector normal unitario de dicho plano es $(0, 0, 1)$ cuyo producto escalar con el vector director unitario de t , $(0, 1, 0)$, es 0. Así pues, en este caso $p_{1*} = (0, 0, 0)$.

Volviendo al caso general, dadas las tres rectas que se cruzan r, s y t tomamos un punto $p \in r \setminus \{p_{1*}\}$. Sea σ la proyección ortogonal de \mathbb{R}^3 sobre la recta s y \vec{d} un vector director de s . Entonces $\{\vec{d}, p - \sigma(p)\}$ forman una base del plano vectorial asociado a Π_p . Tomamos el

2. Nudos y Tipos de nudos

vector normal del plano Π_p que denotaremos por \vec{n} , que se puede escoger como

$$\vec{n} = \vec{d} \wedge (p - \sigma(p)),$$

y la ecuación del plano Π_p es

$$x \in \Pi_p \Leftrightarrow \langle \vec{n}, x \rangle = \langle \vec{n}, p \rangle.$$

Se define ξ_p como el punto de corte de t y Π_p , que sabemos que existe ya que $p \in r \setminus \{p_{1*}\}$, así pues,

$$\xi_p = \Pi_p \cap t.$$

Todos estos elementos se pueden ver en la **Figura 2.21**.

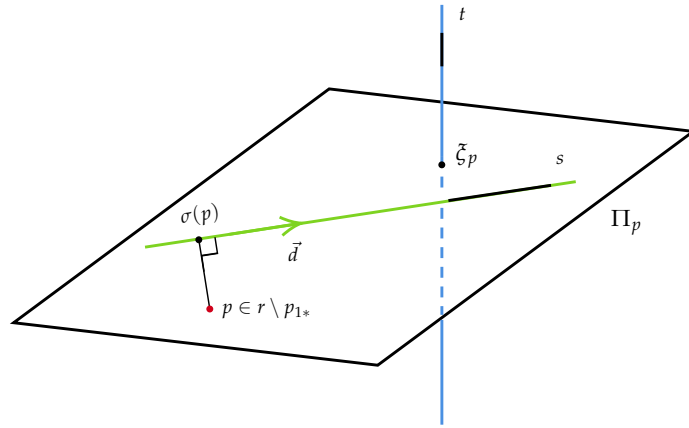


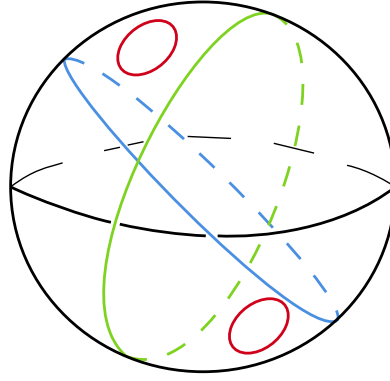
Figura 2.21.: Tres rectas que se cruzan y plano Π_p .

Dado que estamos buscando las rectas que intersecten a r, s y t , tenemos que para $p \in r \setminus \{p_{1*}\}$, la recta que pasa por p y ξ_p intersectará a la recta s para todo $p \in r \setminus \{p_{1*}\}$ excepto para un punto $p_{2*} \in r \setminus \{p_{1*}\}$, para el que serán paralelas. Obtenemos de esta forma la curva parametrizada de “puntos triples”

$$p \in r \setminus \{p_{1*}, p_{2*}\} \mapsto \left[\frac{\xi_p - p}{|\xi_p - p|} \right] \in \mathbb{P}^2.$$

Como es C^1 , su imagen será de medida cero en \mathbb{P}^2 , y por tanto, tendrá interior vacío en \mathbb{P}^2 .

Denotamos por Q_2 a la unión de todas estas imágenes de curvas (descritas en (3)) y rectas proyectivas (descritas en (1) y (2)) en \mathbb{P}^2 . Como hay un número finito de estas por haber un número finito de segmentos de recta en K , Q_2 no será denso en ninguna parte de \mathbb{P}^2 , por tener todas las imágenes de las curvas y las rectas vectoriales interior vacío en \mathbb{P}^2 . Si describimos el plano proyectivo como los puntos de la esfera S^2 , como habíamos hecho previamente para Q_1 , tenemos que los conjuntos de Q_2 , rectas proyectivas e imágenes de las curvas descritas en (3), podrían ser como los representados en la **Figura 2.22**.

Figura 2.22.: Rectas proyectivas e imágenes de curvas en \mathbb{P}^2 .

También podría haberse dado el caso de que las tres rectas r, s y t , conteniendo cada una de ellas a una arista del nudo K , se intersectaran en un mismo punto. Sin embargo, no podrá ocurrir en ningún caso que las tres aristas contenidas en esas tres rectas se intersecten en un mismo punto, ya que en dicho caso, el grado de dicho punto de intersección o vértice sería 3, mientras que en un nudo poligonal todos los vértices tienen que ser de grado 2, ya que un vértice es punto inicial de una arista y punto final de otra. Por ello, no hemos analizado este caso previamente.

Por tanto, la imagen de K bajo cualquier proyección que corresponda a cualquier punto de $\mathbb{P}^2 \setminus (Q_1 \cup Q_2)$ no tiene puntos múltiples de orden $n \geq 3$. Dado que para hallar Q_2 hemos considerado las rectas que intersectan a tres rectas que contienen a tres aristas del nudo en vez de considerar las rectas que intersectan a tres aristas del nudo K , en Q_2 habrá proyecciones para las cuales el nudo K estará en posición regular. Por lo tanto, tendremos que

$$Q \subset (Q_1 \cup Q_2),$$

y en consecuencia

$$\mathbb{P}^2 \setminus (Q_1 \cup Q_2) \subset \mathbb{P}^2 \setminus Q.$$

Por consiguiente, Q es un subconjunto de $Q_1 \cup Q_2$ que no es denso en ninguna parte de \mathbb{P}^2 al no serlo $Q_1 \cup Q_2$. Así, queda demostrado el teorema. \square

2.5. Invariantes de nudos

Los conceptos que se describen en esta sección se basan en el libro *Knot and links* [Rol76] y en el artículo *An overview of knot invariants* [Adk15].

No hay que olvidar que el problema central del estudio matemático de los nudos es realizar una distinción de nudos para su posterior clasificación sin importar cómo sean aparentemente, focalizando dicho análisis en aspectos geométricos y algebraicos. Surgen de esta forma los invariantes de nudos, que definiremos más adelante.

2. Nudos y Tipos de nudos

En la sección anterior hemos demostrado que bajo una rotación cercana a la identidad de \mathbb{R}^3 todo nudo admisible es equivalente a un nudo poligonal en posición regular respecto a una proyección ortogonal sobre un plano arbitraria. Este hecho es el punto de partida para calcular los invariantes básicos por los que se distinguen los diferentes tipos de nudos, que se definen como sigue.

Definición 2.7. Un **invariante de nudos** es una función f que asigna a cada nudo admisible K un objeto $f(K)$ de forma que a nudos equivalentes les son asignados objetos equivalentes.

La principal utilidad de los invariantes de nudos es poder determinar con certeza la no equivalencia de dos nudos admisibles arbitrarios K y K' , viendo la no equivalencia de dichos objetos $f(K)$ y $f(K')$.

Definición 2.8. Dado un invariante de nudos f , se dice que es un **invariante completo** si dados dos nudos admisibles arbitrarios K y K' se verifica que si $f(K)$ y $f(K')$ son equivalentes, entonces K y K' son equivalentes.

Lo que se busca de un invariante de nudos es un equilibrio entre la dificultad de su cálculo y la efectividad de los resultados en cuanto a la distinguibilidad de nudos se refiere, teniendo en cuenta el contexto del problema.

Por una parte, existen lo que se conoce como invariantes elementales, que son por ejemplo el número de cruces de un nudo o la tricolorabilidad de este (no nos centraremos en ellos debido a que no es el objetivo central del trabajo). Estos invariantes son simples de calcular, y aunque no son muy potentes, el número de cruces es útil y se emplea para clasificar y listar nudos, y el estudio de la tricolorabilidad puede ser un método muy simple para probar que un nudo no es trivial, sabiendo que el nudo trivial es claramente no tricoloreable.

Otro invariante mucho más potente es el llamado grupo de un nudo, que es el grupo fundamental del complemento de un nudo y será el invariante de cuyo cálculo nos encargaremos en lo que resta de esta parte del trabajo. Una de las principales desventajas de este invariante es que puede resultar bastante difícil de probar si los grupos de dos nudos admisibles arbitrarios son o no isomorfos.

Por último, tenemos los invariantes polinomiales que causaron una revolución dentro de la teoría de nudos, volviéndose a retomar su estudio tras casi 100 años de abandono. Estos invariantes tienen la ventaja de que son bastante fáciles de calcular, y a diferencia del grupo de un nudo, es fácil comprobar si los polinomios resultantes de dos nudos arbitrarios son equivalentes o no. Entre los invariantes polinomiales destacamos el polinomio de Alexander, el polinomio de Jones y el polinomio del corchete de Kauffman, aunque al igual que hemos dicho con los invariantes elementales, no indagaremos en su cálculo y análisis.

Los invariantes mencionados anteriormente no son completos, por ello muchas veces se opta por utilizar más de un invariante para ser más preciso en la clasificación de los nudos. Dicha clasificación de los nudos es un problema de gran relevancia, ya que hasta la fecha no se ha logrado clasificar los nudos de más de 14 cruces, por no ser los invariantes existentes lo suficientemente fuertes.

3. Grupo de un nudo

En este capítulo, el objetivo principal será calcular presentaciones del grupo de un nudo, ya que como el grupo de un nudo es un invariante de nudos, si comprobamos que los grupos de dos nudos no son isomorfos, dichos nudos no serán equivalentes. En primer lugar, utilizando los conceptos e ideas estudiadas en el capítulo 2, dado un nudo, obtendremos dos presentaciones de grupos, que supondremos que son presentaciones del grupo del nudo que estamos estudiando, a las que llamaremos presentación superior e inferior del grupo de ese nudo. En la siguiente sección comprobaremos que efectivamente, las presentaciones calculadas anteriormente son presentaciones del grupo de dicho nudo. Finalmente, estudiaremos unas presentaciones de grupos llamadas presentaciones Wirtinger, calcularemos unas presentaciones de los grupos de unos nudos concretos y por último, probaremos utilizando el grupo de un nudo como invariante, que el nudo trébol y el trivial no son equivalentes.

Lo expuesto en este capítulo se basa principalmente en el libro *Introduction to Knot Theory* [CF77], sostenido por conceptos topológicos obtenidos mayoritariamente de los apuntes *Topología II*, *Apuntes de clase* [LF19].

3.1. Presentación superior e inferior del grupo de un nudo

Como estudiamos en la Sección 2.4, todo nudo poligonal K es equivalente, bajo una rotación cercana a la identidad de \mathbb{R}^3 , a un nudo poligonal en posición regular respecto a una proyección ortogonal sobre un plano arbitraria. Por tanto, durante este capítulo consideraremos sin pérdida de generalidad que todo nudo poligonal está en posición regular respecto a la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ tal que $\mathcal{P}(x, y, z) = (x, y, 0)$. Debido a esto, siempre que consideremos la proyección de un nudo, consideraremos la imagen del nudo bajo dicha proyección $\mathcal{P}(x, y, z) = (x, y, 0)$.

3.1.1. Arcos superiores e inferiores

Para el estudio que realizaremos de los nudos trataremos de dividir los nudos en diferentes conjuntos denominados arcos, que son conjuntos de \mathbb{R}^3 homeomorfos al intervalo $[0, 1]$.

Dado un nudo poligonal arbitrario K , tenemos que $\exists n \in \mathbb{N}$ tal que $\forall m \geq n$ podemos seleccionar un subconjunto Q de K conteniendo exactamente $2m$ puntos ninguno de los cuales es superior ni inferior de forma que dividan al nudo K en dos clases de arcos segmentados conectados y cerrados, los arcos **superiores** y los **inferiores**, cuyos extremos sean los puntos de Q . La subdivisión debe ser elegida de forma que ningún arco superior contenga un punto inferior y ningún arco inferior contenga un punto superior.

3. Grupo de un nudo

Nos interesa el caso en el que el número de puntos de Q es n , siendo n el mínimo número natural tal que existen $2n$ puntos de K que dividen dicho nudo poligonal en arcos superiores e inferiores. Tomando este n , los arcos superiores e inferiores se alternarán alrededor del nudo de forma que cada punto de Q pertenecerá a un arco superior y a uno inferior.

Por ejemplo, para el nudo de la Figura 2.5 el valor mínimo sería $n = 1$, por tanto, podríamos hacer la división del nudo en arcos superiores e inferiores como se muestra en la Figura 3.1.

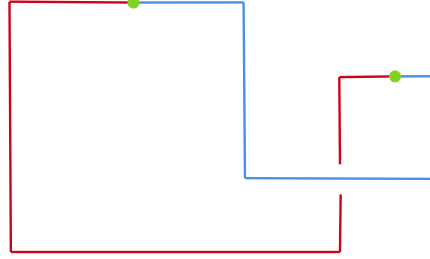


Figura 3.1.: Nudo dividido en arcos superiores e inferiores.

Los dos puntos señalados con color verde son los dos puntos de Q (son dos ya que $n = 1$), el arco señalado en azul es el arco superior, ya que, como vemos, contiene un punto superior y ninguno inferior, y el arco señalado en rojo es el arco inferior, ya que contiene un punto inferior y ninguno superior. Utilizaremos esta distinción de colores en los diversos gráficos que habrá a lo largo del trabajo.

Una vez que hemos dividido un nudo poligonal arbitrario K , en la menor cantidad de arcos superiores e inferiores posible, que por lo tanto se alternarán alrededor del nudo, vamos a proceder a etiquetar cada uno de esos arcos. Denotamos los arcos superiores por A_1, \dots, A_n , llamando A a su unión $A_1 \cup \dots \cup A_n$, y los arcos inferiores por B_1, \dots, B_n , llamando B a su unión $B_1 \cup \dots \cup B_n$, siendo n el número de arcos superiores e inferiores del nudo. Una vez dividido el nudo poligonal en arcos superiores e inferiores, tenemos que existe un homeomorfismo de \mathbb{R}^3 en sí mismo que desplaza los puntos verticalmente, es decir, paralelo al eje z , de tal forma que la imagen de $A \setminus Q$ se encuentra sobre el plano $\mathbb{R}^2 \times \{0\}$ y la imagen de $B \setminus Q$ se encuentra bajo el plano $\mathbb{R}^2 \times \{0\}$. Como K y su imagen bajo este homeomorfismo son nudos poligonales equivalentes, durante todo el trabajo consideraremos que $A \setminus Q$ está en $\{(x, y, z) \in \mathbb{R}^3 : z > 0\}$ y $B \setminus Q$ se encuentra en $\{(x, y, z) \in \mathbb{R}^3 : z < 0\}$, por lo que, $Q \subset \mathbb{R}^2 \times \{0\}$. En consecuencia, los puntos de Q van a ser los únicos puntos de K que se encuentran en el plano $\mathbb{R}^2 \times \{0\}$.

3.1.2. Grupo libre y presentación de un grupo

A continuación, vamos a definir ciertos conceptos que serán de crucial relevancia para el estudio del grupo de un nudo poligonal arbitrario K que realizaremos en la siguiente sección. Concretamente, estudiaremos los conceptos de grupo, grupo libre y presentación de un grupo.

Lo expuesto en esta subsección sobre grupos libres está basado en los libros *Non-commutative cryptography and complexity of group-theoretic problems* [MSU11] y *A First Course in Abstract Algebra* [FK03].

Definición 3.1. Un conjunto no vacío G con una operación binaria $*$ se dice que es un **grupo** si cumple las siguientes propiedades:

1. La operación binaria $*$ es asociativa, es decir,

$$a * (b * c) = (a * b) * c, \text{ para todo } a, b, c \in G.$$

2. Existe un elemento neutro $1 \in G$, tal que

$$a * 1 = a = 1 * a, \text{ para todo } a \in G.$$

3. Para todo elemento $a \in G$, existe un elemento $a' \in G$, llamado elemento simétrico, tal que

$$a * a' = 1 = a' * a.$$

4. Si también se cumple que la operación $*$ es conmutativa, es decir,

$$a * b = b * a, \text{ para todo } a, b \in G,$$

el grupo se dice que es un grupo abeliano.

Asumimos que tenemos un conjunto \mathcal{A} de cardinalidad l . Llamaremos *alfabeto* al conjunto \mathcal{A} , y a sus componentes, *letras*. Con *sílaba* nos referimos a un símbolo a^n , donde a es una letra del alfabeto \mathcal{A} , y n es un número entero y llamaremos *palabra* a una secuencia finita de sílabas. Por ejemplo, si a, b, c son letras del alfabeto \mathcal{A} , $c^{-2}b^0b^2a^1$ es una palabra de dicho alfabeto.

Hay dos tipos de contracciones de palabras:

- Si una palabra u es de la forma $r_1a^0r_2$, donde r_1 y r_2 son palabras, decimos que la palabra $v = r_1r_2$ es obtenida de u mediante una contracción elemental de tipo I.
- Si una palabra u es de la forma $r_1a^pa^qr_2$, donde r_1 y r_2 son palabras, decimos que la palabra $v = r_1a^{p+q}r_2$ es obtenida de u mediante una contracción elemental de tipo II.

Dos palabras u y v se consideran *equivalentes* (se escribe $u \sim v$) si una puede ser obtenida de la otra mediante un número finito de contracciones y expansiones (proceso opuesto a las contracciones) elementales. Denotaremos por $[u]$ a la clase de equivalencia de una palabra u , y por $F[\mathcal{A}]$ al conjunto de clases de equivalencia de palabras.

Es fácil de ver que si $u \sim u'$ y $v \sim v'$, entonces $uv \sim u'v'$. En otras palabras, obtenemos que, $[u][v] = [uv]$. También es claro que la multiplicación es asociativa, y que el elemento 1 es el elemento neutro. Además, en $F[\mathcal{A}]$ todo elemento tiene un inverso, el inverso $[u]^{-1}$ de la clase $[u]$ se representa mediante la palabra \bar{u} , que se obtiene de u invirtiendo el orden de las sílabas y cambiando el signo del exponente de cada sílaba. Por ejemplo, si $u = c^{-2}b^0b^2a^1$, entonces $\bar{u} = a^{-1}b^{-2}b^0c^2$. Esto muestra que $F[\mathcal{A}]$ es un grupo, y es llamado el **grupo libre**

3. Grupo de un nudo

sobre el alfabeto \mathcal{A} (o grupo libre generado por \mathcal{A}).

En el grupo $F[\mathcal{A}]$, cada elemento se puede escribir como un producto de potencias de $[a], [b], [c], \dots$ ($a, b, c \in \mathcal{A}$). Por ejemplo, $[a^3 b^2 c^{-2}] = [a]^3 [b]^2 [c]^{-2}$. Por ello, los elementos $[a], [b], [c], \dots$ constituyen un conjunto de elementos generadores del grupo $F[\mathcal{A}]$. Denotaremos a este conjunto de generadores por $[\mathcal{A}]$.

A continuación, definiremos los conceptos de homomorfismo e isomorfismo de grupos.

Definición 3.2. Dados dos grupos G y G' , un **homomorfismo** de grupos de G a G' es una aplicación $\varphi: G \rightarrow G'$ cumpliendo que

$$\varphi(ab) = \varphi(a)\varphi(b) \text{ para todo } a, b \in G.$$

Un isomorfismo de grupos es simplemente un homomorfismo de grupos biyectivo. Diremos que dos grupos son isomorfos si existe un isomorfismo entre ellos.

Definición 3.3. Dado un grupo G , un **conjunto generador** de un grupo G es un subconjunto S de G tal que todo elemento de G puede ser expresado como el producto de un número finito de elementos de S .

Ya estamos en disposición de definir el concepto de base libre.

Definición 3.4. Diremos que un conjunto generador E de un grupo G es una **base libre** si, dado un grupo cualquiera H , cualquier función $\phi: E \rightarrow H$ puede ser extendida a un homomorfismo $\phi^*: G \rightarrow H$, de forma que el diagrama siguiente sea conmutativo

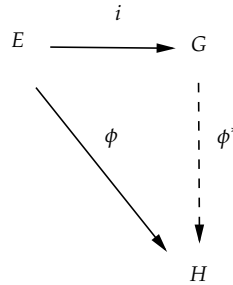


Figura 3.2.: Diagrama base libre E .

donde $i: E \rightarrow G$ es la aplicación inclusión.

De esta forma, un grupo que tenga una base libre se dirá que es un grupo libre.

Proposición 3.1. Un grupo es libre si y solo si es isomorfo a $F[\mathcal{A}]$ para algún alfabeto \mathcal{A} .

En consecuencia, se sigue que a cada grupo libre G le corresponde un número n , tal que cada base libre de G tiene exactamente cardinalidad n , que coincide con la cardinalidad del alfabeto \mathcal{A} . Por tanto, dos grupos libres G y G' son isomorfos si tienen respetivamente bases libres E y E' de la misma cardinalidad. A la cardinalidad de una base libre de un grupo libre G se le llama *rango* del grupo libre G .

Otro resultado clave que utilizaremos es el siguiente.

Proposición 3.2. *Todo grupo es imagen homomorfa de un grupo libre, es decir, dado un grupo arbitrario G , existe un grupo libre F y un epimorfismo $f: F \rightarrow G$.*

Demostración. Sea E un conjunto generador del grupo G y sea F un grupo libre sobre E respecto de la función $\varphi: E \rightarrow F$. Entonces existe un homomorfismo f que hace que el diagrama siguiente conmute

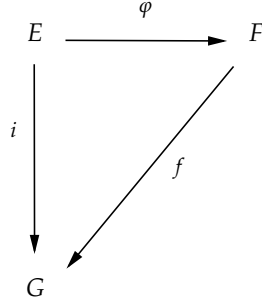


Figura 3.3.: Diagrama imagen homomorfa de un grupo libre F .

donde $i: E \rightarrow G$ es la aplicación inclusión. Debemos probar que f es una aplicación sobreyectiva.

Sea $a \in G$, entonces dado que E es un conjunto generador del grupo G existirán $x_1, x_2, \dots, x_k \in E$ y $n_1, n_2, \dots, n_k \in \mathbb{Z}$ tales que $a = x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$. Sea $g = \varphi(x_1)^{n_1} \varphi(x_2)^{n_2} \dots \varphi(x_k)^{n_k} \in F$ luego

$$\begin{aligned}
 f(g) &= f(\varphi(x_1)^{n_1} \varphi(x_2)^{n_2} \dots \varphi(x_k)^{n_k}) \\
 &= f(\varphi(x_1))^{n_1} f(\varphi(x_2))^{n_2} \dots f(\varphi(x_k))^{n_k} \\
 &= i(x_1)^{n_1} i(x_2)^{n_2} \dots i(x_k)^{n_k} \\
 &= x_1^{n_1} x_2^{n_2} \dots x_k^{n_k} \\
 &= a.
 \end{aligned}$$

Como el elemento $a \in G$ era arbitrario, hemos demostrado que $f: F \rightarrow G$ es un epimorfismo, como queríamos. □

Un grupo arbitrario G , está determinado si hay un conjunto dado de elementos g_1, g_2, \dots , llamados **generadores**, que generan el grupo, y un conjunto de ecuaciones $f_1(g_1, g_2, \dots) = 1$, $f_2(g_1, g_2, \dots) = 1, \dots$, llamadas ecuaciones de definición o **relaciones de definición**, que tienen la propiedad de que toda relación existente entre los elementos g_1, g_2, \dots , es una consecuencia algebraica de dichas ecuaciones.

Dado F un grupo libre con base libre x_1, x_2, \dots , en correspondencia uno a uno con los generadores g_1, g_2, \dots de G , sea ϕ el homomorfismo de F a G dado por $\phi x_j = g_j, j = 1, 2, \dots$. Para todas las ecuaciones de definición $f_i(g_1, g_2, \dots) = 1$, fijamos

$$r_i = f_i(x_1, x_2, \dots), \quad i = 1, 2, \dots$$

3. Grupo de un nudo

Es decir, r_i es el elemento de F que se obtiene de reemplazar cada ocurrencia de g_j , $j = 1, 2, \dots$, en la expresión $f_i(g_1, g_2, \dots)$ por x_j . Por ejemplo, si la i -ésima ecuación de definición es $g_1^{-1} g_2 g_1 g_2^{-1} = 1$, entonces $r_i = x_1^{-1} x_2 x_1 x_2^{-1}$. La afirmación que teníamos de que la ecuación $f_i(g_1, g_2, \dots) = 1$ se cumple en G es por tanto equivalente a decir que r_i está dentro del núcleo de ϕ , esto es,

$$\phi r_i = f_i(g_1, g_2, \dots) = 1.$$

A los elementos r_1, r_2, \dots se les llamará **relatores**.

Dado un elemento f de un grupo arbitrario Q , es llamado consecuencia de un conjunto de elementos f_1, f_2, \dots en Q si todo homomorfismo ψ de Q en un grupo H tal que la imagen de los elementos f_1, f_2, \dots es 1, cumple que la imagen por ψ de f es 1. Dado que el núcleo de todo homomorfismo ψ es un subgrupo normal de Q , podremos redefinir el concepto anterior de la siguiente forma.

Definición 3.5. Un elemento f de Q es **consecuencia** de los elementos f_1, f_2, \dots si está contenido en todo subgrupo normal de Q que contiene a todos los elementos f_1, f_2, \dots .

Una vez visto el concepto de relator ya podemos definir el concepto de presentación de un grupo. Sea F un grupo libre con base libre E . Una **presentación de grupo**, denotada por $\langle x : r \rangle$, es un objeto que consiste en un subconjunto x de los elementos de la base libre, E , y un subconjunto r del subgrupo $F[x]$, siendo $F[x]$ el grupo libre generado por x . El conjunto x es llamado el conjunto de *generadores* de la presentación y el conjunto r es llamado el conjunto de *relatores* de la presentación. El **grupo definido por la presentación** $\langle x : r \rangle$ es el grupo cociente $F[x]/R$, que lo denotaremos por $|x : r| = F[x]/R$, donde R es el conjunto de elementos consecuencia de los relatores r en $F[x]$.

Una **presentación de un grupo** G consiste en una presentación de grupo $\langle x : r \rangle$ y un isomorfismo ι del grupo $|x : r|$ al grupo G . Claramente, cualquier homomorfismo ϕ del grupo libre $F[x]$ al grupo G cuyo núcleo sea el conjunto de elementos consecuencia de r , determina una presentación de G . En sentido opuesto, cualquier presentación de G determina ese homomorfismo, es decir, si γ denota el homomorfismo de $F[x]$ a $F[x]/R$, descrito en el diagrama de la **Figura 3.4**, entonces las aplicaciones ϕ y ι determinan la una a la otra de forma única. Cuando deseamos precisar acerca de la presentación, como veremos que haremos a lo largo del trabajo, escribiremos $\langle x : r \rangle_\phi$ para indicar que $\langle x : r \rangle$ es la presentación del grupo G con respecto al homomorfismo ϕ .

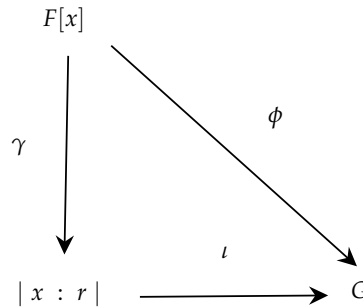


Figura 3.4.: Diagrama presentación de un grupo G .

3.1.3. Cálculo de la presentación superior e inferior

Vistos los conceptos previos, ya podemos centrarnos en el análisis del invariante de nudos que estamos estudiando, el grupo de un nudo. Para un nudo arbitrario K , cada presentación está hecha con respecto a una orientación de K y de \mathbb{R}^3 . De acuerdo a esto, una de las dos direcciones a través del nudo K es escogida como positiva y dibujaremos una flecha en K para representarla.

A la hora de etiquetar los arcos superiores e inferiores del nudo, con el fin de seguir un patrón determinado, si recorremos el nudo en la dirección del mismo, todo arco inferior etiquetado con B_i tendrá que ir seguido del arco superior etiquetado con A_i , para todo $i = 1, \dots, n$. Además el arco inferior recorrido a continuación del arco superior etiquetado con A_i , tiene que estar etiquetado con:

- B_{i+1} si $i = 1, \dots, n - 1$.
- B_1 si $i = n$.

A continuación, vamos a estudiar el concepto de camino simple. Antes de enunciar este concepto, recordemos que durante este capítulo considerábamos sin pérdida de generalidad que todo nudo poligonal está en posición regular respecto a la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ tal que $\mathcal{P}(x, y, z) = (x, y, 0)$.

Definición 3.6. Un **camino** en un espacio topológico X es una función continua $a: [0, 1] \rightarrow X$. A los puntos $a(0)$ y $a(1)$, los llamaremos respectivamente, origen y extremo del camino a . Dado un nudo poligonal K y considerando la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ tal que $\mathcal{P}(x, y, z) = (x, y, 0)$, diremos que un camino a en $\mathbb{R}^2 \times \{0\}$ es **simple** si satisface las siguientes condiciones:

- i. Es poligonal.
- ii. Ni el origen, $a(0)$, ni el extremo, $a(1)$, del camino pertenecen a $\mathcal{P}K$.
- iii. El camino a intersecta a $\mathcal{P}K$ en tan sólo un número finito de puntos, ninguno de los cuales es un vértice de a o un vértice de $\mathcal{P}K$.

En las figuras de la **Figura 3.5**, podemos ver la proyección de un nudo orientado K junto con un camino a en $\mathbb{R}^2 \times \{0\}$. El camino a de la **Figura 3.5a** no sería simple ya que no verifica la condición (iii) que tiene que verificar un camino para ser simple, mientras que el camino a de la **Figura 3.5b** sí que sería un camino simple.

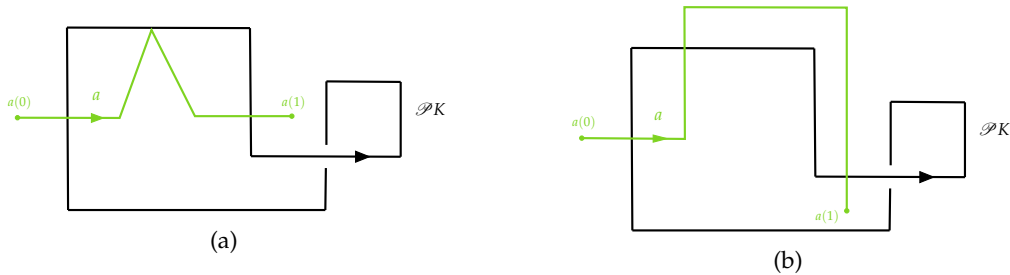


Figura 3.5.: (a) Camino a no simple. (b) Camino a simple.

3. Grupo de un nudo

Vamos a definir una operación para caminos consecutivos, es decir, caminos $a_1, a_2: [0, 1] \rightarrow X$ tales que $a_1(1) = a_2(0)$, que se utilizará con bastante frecuencia en este capítulo.

Definición 3.7. Dados dos caminos $a_1: [0, 1] \rightarrow X$ y $a_2: [0, 1] \rightarrow X$ que sean consecutivos, definimos

$$a_1 \cdot a_2: [0, 1] \rightarrow X, \quad a_1 \cdot a_2(t) = \begin{cases} a_1(2t) & \text{si } t \in [0, 1/2] \\ a_2(2t - 1) & \text{si } t \in [1/2, 1] \end{cases}$$

Dado un nudo poligonal K dividido en arcos superiores e inferiores, tomamos $x = \{x_1, \dots, x_n\}$, un conjunto formado por n símbolos que se encontrarán en biyección con los n arcos superiores del nudo poligonal, A_1, \dots, A_n . Dichos símbolos del conjunto determinarán la forma en la que un camino simple a , descrito anteriormente, intersectará a cada una de las proyecciones de los diferentes arcos superiores, $\mathcal{P}A_1, \dots, \mathcal{P}A_n$.

Definiremos el grupo libre $F[x]$, que tendrá como base libre las clases de equivalencia de x_1, \dots, x_n , siendo estos los símbolos del conjunto x antes descrito. A cada camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ le asignaremos un elemento $a^\#$ de $F[x]$ definido como sigue:

$$a^\# = x_{i_1}^{\epsilon_1} \cdots x_{i_l}^{\epsilon_l},$$

donde las proyecciones de los arcos superiores que se intersectan con el camino a son, en orden, $\mathcal{P}A_{i_1}, \dots, \mathcal{P}A_{i_l}$ (aquí se aprecia la biyección que existe entre los arcos superiores A_i y los símbolos x_i , $i = 1, \dots, n$) y donde el valor ϵ_k dependerá de la regla del tornillo que se introdujo para la orientación, por mantener la visión espacial. Concretamente

- $\epsilon_k = 1$ si A_{i_k} y el camino a forman un tornillo de la mano derecha (orientación horaria).



Figura 3.6.: Tornillo de la mano derecha para la proyección de un arco superior.

- $\epsilon_k = -1$ si A_{i_k} y el camino a forman un tornillo de la mano izquierda (orientación antihoraria).

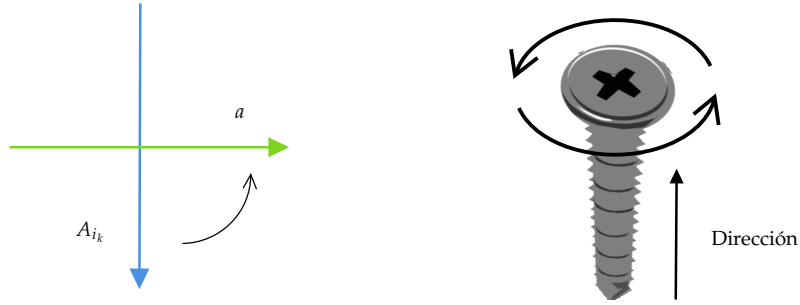


Figura 3.7.: Tornillo de la mano izquierda para la proyección de un arco superior.

La aplicación $a \rightarrow a^\#$ preserva el producto, es decir,

$$(a_1 \cdot a_2)^\# = a_1^\# a_2^\#,$$

para caminos simples consecutivos a_1, a_2 . Sin embargo, no tiene por qué ser una aplicación sobreyectiva.

Realizando una pequeña variación en el nudo de la Figura 2.5 (para que sea más representativo el ejemplo), tenemos el nudo de la Figura 3.8, en el que como habíamos indicado anteriormente, hemos representado los arcos superiores (A_1 y A_2) con color azul y los arcos inferiores con color rojo. También podemos ver un camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ (recordemos que B es la unión de todos los arcos inferiores del nudo poligonal en posición regular, K) representado con color verde, al que le vamos a asignar un elemento del grupo $F[x]$ según los criterios vistos anteriormente.

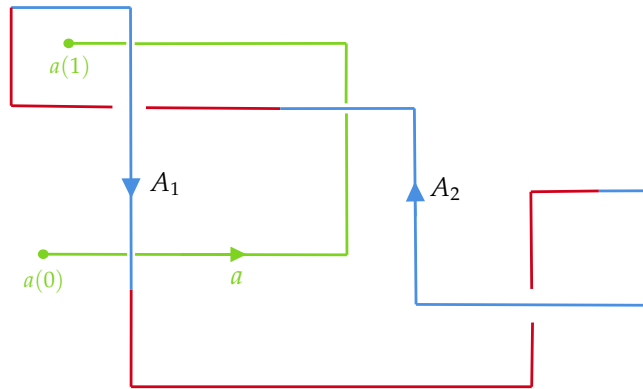


Figura 3.8.: Camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$.

En este caso $a^\# = x_1^{-1} x_2 x_1$.

3. Grupo de un nudo

Para continuar con el estudio de los nudos, dado un nudo poligonal K , seleccionaremos dos puntos base,

- p_0 , un punto cualquiera cuya coordenada z sea mayor que la de cualquier punto del nudo K ,
- p'_0 , un punto cualquiera cuya coordenada z sea menor que la de cualquier punto del nudo K .

Sin embargo, para mayor comodidad y sencillez, asumiremos que $p_0 = (0, 0, z_0)$ y $p'_0 = (0, 0, -z_0)$, siendo z_0 un número positivo que sea mayor que la coordenada z , en valor absoluto, de todos los puntos del nudo K . De esta forma, una rotación de 180° alrededor del eje x envía un punto base al otro (p_0 a p'_0 y viceversa). Por lo dicho anteriormente, dado un punto arbitrario del nudo $p = (p_x, p_y, p_z) \in K$, se tiene que cumplir que $-z_0 < p_z < z_0$. Finalmente, elegimos un punto $q_0 \in (\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}K$, que lo utilizaremos más adelante.

Por ejemplo, para el nudo representado con color verde en la [Figura 3.9](#), tenemos que el punto del nudo que tiene coordenada z mayor en valor absoluto es $r = (-1, -0.5, 3)$, por lo que podríamos tomar como punto p_0 el punto $p_0 = (0, 0, 4)$, ya que $4 > 3$, y en consecuencia, $p'_0 = (0, 0, -4)$, es decir, el punto obtenido de realizar una rotación de 180° alrededor del eje x , como dijimos previamente.

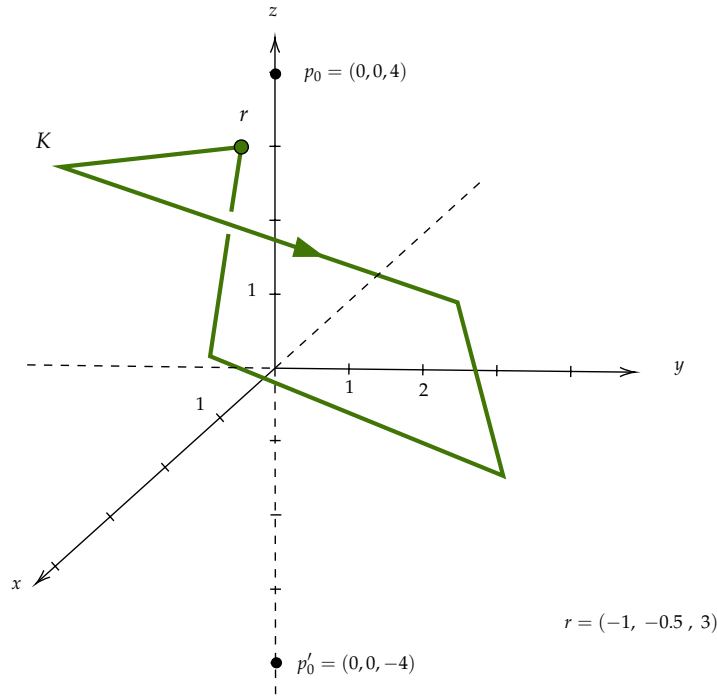


Figura 3.9.: Puntos p_0 y p'_0 para un nudo poligonal K .

Dado un punto $p \in \mathbb{R}^2 \times \{0\}$, llamamos \bar{p} al camino que recorre el segmento que une el punto p_0 con el punto justo encima de p , a la altura de p_0 , es decir, el punto cuya coordenada

3.1. Presentación superior e inferior del grupo de un nudo

z sea la de p_0 y las coordenadas x e y sean las del punto p , y después, el segmento que une dicho punto con el punto p . Por ejemplo, si $p_0 = (0, 0, 4)$ y $p = (-1, -1, 0)$, el camino \bar{p} sería el que recorre el segmento con origen el punto p_0 y extremo el punto $(-1, -1, 4)$ y el segmento con origen este último punto y extremo el punto p . Una vez definido el camino \bar{p} para cualquier punto $p \in \mathbb{R}^2 \times \{0\}$, dado un camino a en $\mathbb{R}^2 \times \{0\}$, definimos

$$*a = \overline{a(0)} \cdot a \cdot \overline{a(1)}^{-1}, \quad (3.1)$$

que está bien definido ya que son consecutivos los caminos implicados.

Este camino $*a$ en \mathbb{R}^3 recorre la curva que va desde p_0 hasta $a(0)$, pasando por el punto que hay encima de $a(0)$, a la altura de p_0 . Posteriormente recorre el camino a , y por último recorre la curva con origen $a(1)$ y extremo p_0 , pasando por el punto que hay encima de $a(1)$ a la altura de p_0 .

Dado el nudo poligonal K y el camino simple a de la [Figura 3.8](#), y tomando un punto p_0 verificando las condiciones mencionadas anteriormente, tenemos que el camino $*a$ sería el representado con color verde en la [Figura 3.10](#).

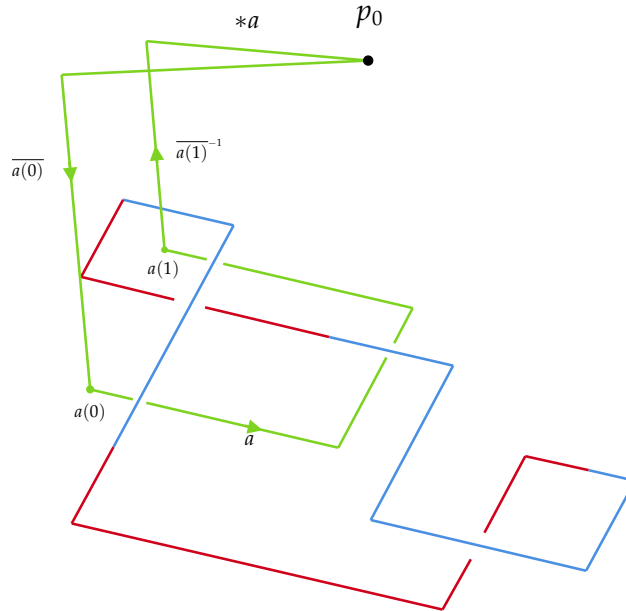


Figura 3.10.: Camino $*a$.

Antes de indagar en el estudio de la presentación superior e inferior del grupo de un nudo poligonal arbitrario K , vamos a definir el grupo fundamental de un espacio topológico en un punto. Para definirlo, debemos estudiar previamente el concepto de lazo y el de espacio de lazos.

Definición 3.8. Dado un espacio topológico X y un punto fijo $p \in X$, un **lazo** con base p es una aplicación continua $\gamma: [0, 1] \rightarrow X$ que verifica $\gamma(0) = \gamma(1) = p$.

3. Grupo de un nudo

Ahora veremos el criterio que utilizaremos para determinar cuando existe relación entre dos lazos con base un mismo punto fijo.

Definición 3.9. Sea X un espacio topológico, $p \in X$ y $\alpha, \beta: [0, 1] \rightarrow X$ dos lazos con base un punto común p . Se dice que α y β son **homotópicos** si existe una aplicación continua $H: [0, 1] \times [0, 1] \rightarrow X$ que verifique que

- $H(s, 0) = \alpha(s)$ para todo $s \in (0, 1)$,
- $H(s, 1) = \beta(s)$ para todo $s \in (0, 1)$,
- $H(0, t) = H(1, t) = p$ para todo $t \in (0, 1)$.

Definición 3.10. Dado $p \in X$, siendo X un espacio topológico, denotemos

$$\Omega_p(X) = \{\gamma: [0, 1] \rightarrow X \mid \gamma \text{ es un lazo con } \gamma(0) = \gamma(1) = p\}.$$

A este conjunto se le llamará **espacio de lazos** y dados dos lazos $\alpha, \beta \in \Omega_p(X)$ se escribirá $\alpha \simeq \beta$ si dichos lazos son homotópicos.

Proposición 3.3. La relación binaria \simeq es de equivalencia.

Denotaremos $\Pi(X, p)$ al espacio cociente $\Omega_p(X) / \simeq$. Si $\alpha \in \Omega_p(X)$ su clase en $\Pi(X, p)$ es denotada por $[\alpha]$, y a los elementos de $\Pi(X, p)$ les llamaremos clases de homotopía de lazos en X con base el punto p .

Como conjunto de elementos $\Pi(X, p)$ veremos que resultará ser el grupo fundamental de X en p . Sin embargo, para dotar al conjunto $\Pi(X, p)$ de la estructura algebraica de grupo hace falta definir una operación interna. Denotaremos por \cdot a esta operación, que será la previamente definida en *Def. 3.7*.

Para ver que la operación \cdot provee al conjunto $\Pi(X, p)$ de estructura de grupo algebraico, veremos primero que dicha operación está bien definida en clases de equivalencia.

Proposición 3.4. Sean X un espacio topológico y un punto $p \in X$. Sean $\alpha, \alpha', \beta, \beta' \in \Omega_p(X)$ lazos tales que $\alpha \simeq \alpha'$ y $\beta \simeq \beta'$. Entonces $\alpha \cdot \beta \simeq \alpha' \cdot \beta'$.

Con la siguiente proposición garantizaremos que $(\Pi(X, p), \cdot)$ es un grupo.

Proposición 3.5. Sean X espacio topológico, $p \in X$ y $\alpha, \beta, \gamma \in \Omega_p(X)$. Las siguientes propiedades son ciertas:

- i. $\alpha \cdot (\beta \cdot \gamma) \simeq (\alpha \cdot \beta) \cdot \gamma$,
- ii. $\epsilon_p \cdot \alpha \simeq \alpha \cdot \epsilon_p \simeq \alpha$,
- iii. $\alpha \cdot \bar{\alpha} \simeq \epsilon_p$.

siendo $\bar{\alpha}$ el lazo dado por $\bar{\alpha}: [0, 1] \rightarrow X$ tal que $\bar{\alpha}(t) := \alpha(1 - t)$, y siendo el lazo ϵ_p el lazo más simple que podemos construir, es decir $\epsilon_p: [0, 1] \rightarrow X$ tal que $\epsilon_p(t) = p$ para todo $t \in [0, 1]$, también conocido como lazo trivial.

Por tanto, llamaremos al grupo $(\Pi(X, p), \cdot)$ **grupo fundamental** del espacio X en el punto p .

Definición 3.11. Un espacio topológico X se dice que es **arcoconexo** si para cualesquiera $p, q \in X$, existe un camino $\alpha: [0, 1] \rightarrow X$ conectando p y q .

Definición 3.12. Dado un espacio topológico X , la **arcocomponente** de un punto $p \in X$, que denotaremos por \mathcal{C}_p , se define como el mayor subespacio arcoconexo de X conteniendo a p . Es decir,

$$\mathcal{C}_p = \{q \in X : \text{existe un camino en } X \text{ que une } p \text{ y } q\}$$

A continuación vamos a ver que en el caso de ser X un espacio arcoconexo (como lo serán los espacios con los que trabajaremos cuando nos centremos en los nudos), la clase de isomorfía de $\Pi(X, p)$ no dependerá del punto $p \in X$.

Proposición 3.6. Sean p y q puntos en la misma arcocomponente de un espacio topológico X , y tomamos un camino $f: [0, 1] \rightarrow X$ que conecte p con q , es decir, que $f(0) = p$ y $f(1) = q$. Consideramos

$$\mathcal{U}_f: \Pi(X, p) \rightarrow \Pi(X, q), \quad \mathcal{U}_f([\alpha]) = [\bar{f} \cdot \alpha \cdot f].$$

Entonces \mathcal{U}_f está bien definido y es un isomorfismo entre los grupos fundamentales $(\Pi(X, p), \cdot)$ y $(\Pi(X, q), \cdot)$.

Corolario 3.1. El grupo fundamental $\Pi(X, p)$ está determinado salvo isomorfismos por la arcocomponente del punto p . En particular, si X es arcoconexo, entonces la clase de isomorfía de $\Pi(X, p)$ es independiente del punto $p \in X$.

Como consecuencia del **Corolario 3.1**, en caso de que X sea arcoconexo podemos escribir simplemente $\Pi(X)$ sin especificar el punto base.

Centrándonos de nuevo en la teoría de nudos, tenemos que el grupo $F[x]$ definido en la página 36, será el grupo libre de la presentación superior del grupo de un nudo poligonal arbitrario K . Estamos ya en posición de definir el homomorfismo $\phi: F[x] \rightarrow \Pi(\mathbb{R}^3 \setminus K, p_0)$, que lo haremos como sigue. Para cada $j \in \{1, \dots, n\}$, tomamos a_j un camino simple cualquiera en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{PB}$ que verifique que $a_j^\# = x_j$ para todo $j = 1, \dots, n$. Definimos

$$\phi x_j = [*a_j] \quad \text{para todo } j = 1, \dots, n, \quad (3.2)$$

donde los corchetes indican la clase de equivalencia en $\mathbb{R}^3 \setminus K$ del lazo con base p_0 , $*a_j$. Por ende, es obvio que ϕx_j es independiente de la elección del camino representativo a_j , siempre que cumpla que $a_j^\# = x_j$. El homomorfismo ϕ se puede extender a todo el grupo $F[x]$, siendo

$$\phi a^\# = [*a],$$

para cualquier camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{PB}$. Será nuestro objetivo en la **Sección 3.2** probar que ϕ es sobreyectivo, o en otras palabras, que $\phi x_1, \dots, \phi x_n$ genera $\Pi(\mathbb{R}^3 \setminus K, p_0)$. Geométricamente, como se refleja en la **Figura 3.11**, es evidente que todo lazo con base p_0 en $\mathbb{R}^3 \setminus K$ es equivalente al producto de los lazos $*a_j, j = 1, \dots, n$, y sus inversos, pero en la

3. Grupo de un nudo

Sección 3.2 lo demostraremos matemáticamente.

Por ejemplo, para el nudo de la Figura 3.8, es geoméricamente obvio, como sugiere la Figura 3.11, que todo lazo con base p_0 en $\mathbb{R}^3 \setminus K$ es equivalente a una palabra sobre el alfabeto $\{*a_1, *a_2\}$, donde los lazos $*a_1$ y $*a_2$ están asociados a los dos arcos superiores del nudo. Dichos lazos están correspondidos con los caminos simples a_1 y a_2 respectivamente, que verifican que $a_1^\# = x_1$ y $a_2^\# = x_2$.

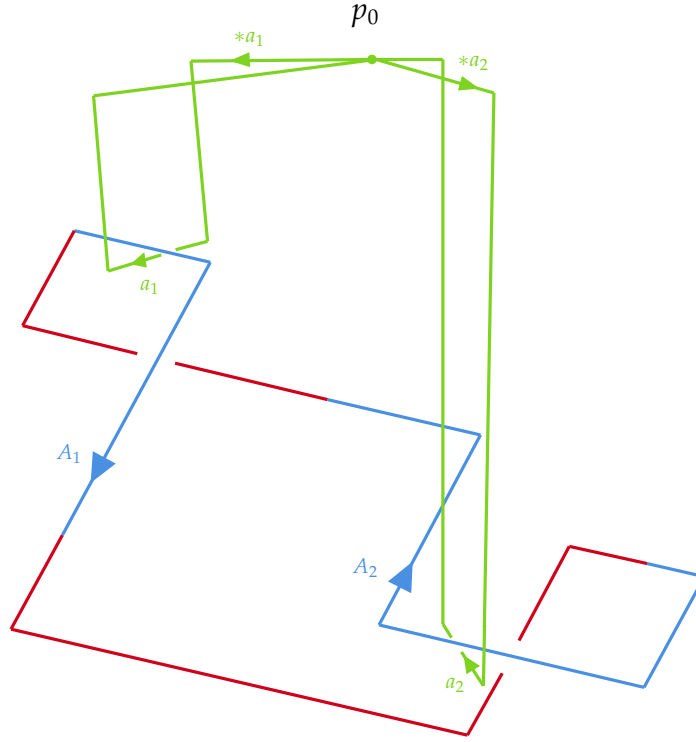


Figura 3.11.: Lazos con base en p_0 .

A continuación, vamos a efectuar una serie de pasos similares a los que hemos realizado para tratar de obtener la presentación superior del grupo de un nudo poligonal arbitrario K , pero en este caso serán para obtener la presentación inferior de este.

Los generadores de la presentación inferior son elegidos de manera análoga. Sea $y = \{y_1, \dots, y_n\}$ un conjunto formado por n símbolos que se encontrarán en biyección con los n arcos inferiores del nudo poligonal, B_1, \dots, B_n . Dichos símbolos determinarán la forma en la que un camino simple b intersectará a cada una de las proyecciones de los diferentes arcos inferiores $\mathcal{P}B_1, \dots, \mathcal{P}B_n$.

Definiremos el grupo libre $F[y]$, que tendrá como base libre las clases de equivalencia de y_1, \dots, y_n y a cada camino simple b en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$ le asignaremos un elemento b^\flat de $F[y]$ definido por

$$b^\flat = y_{j_1}^{\delta_1} \dots y_{j_m}^{\delta_m},$$

donde las proyecciones inferiores que se intersectan con el camino b son, en orden, $\mathcal{P}B_{j_1}, \dots, \mathcal{P}B_{j_m}$, y

- $\delta_k = 1$ si B_{j_k} y el camino b forman un tornillo de la mano derecha (orientación horaria).



Figura 3.12.: Tornillo de la mano derecha para la proyección de un arco inferior.

- $\delta_k = -1$ si B_{j_k} y el camino b forman un tornillo de la mano izquierda (orientación antihoraria).

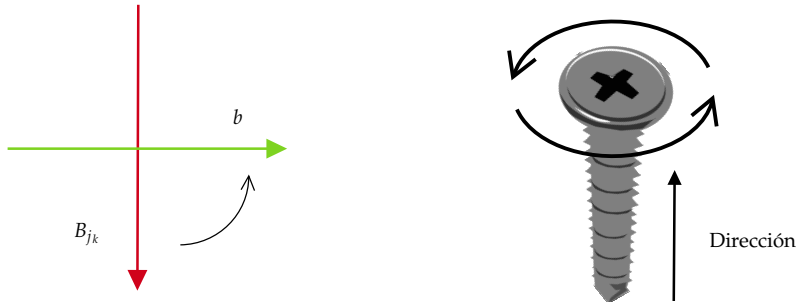


Figura 3.13.: Tornillo de la mano izquierda para la proyección de un arco inferior.

Estamos ya en posición de definir el homomorfismo $\phi': F[y] \rightarrow \Pi(\mathbb{R}^3 \setminus K, p'_0)$. Sea \mathcal{R} la reflexión respecto al plano $\mathbb{R}^2 \times \{0\}$, de forma que $\mathcal{R}(x, y, z) = (x, y, -z)$. Dado a un camino simple cualquiera en $\mathbb{R}^2 \times \{0\}$, definimos el camino cerrado ${}_a a$ como ${}_a a = \mathcal{R}(*a)$, siendo $*a$ el camino definido en (3.1). Para cada $i \in \{1, \dots, n\}$, tomamos b_i un camino simple cualquiera en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$ que verifique que $b_i^b = y_i$ para todo $i = 1, \dots, n$. De esta forma definimos

$$\phi' y_i = [{}_a b_i] \quad \text{para todo } i = 1, \dots, n, \quad (3.3)$$

donde los corchetes indican la clase de equivalencia en $\mathbb{R}^3 \setminus K$ del lazo con base p'_0 , ${}_a b_i$. Dicho homomorfismo ϕ' se puede extender a todo el grupo $F[y]$, siendo

$$\phi' b^\flat = [{}_a b],$$

para cualquier camino simple b en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$. De nuevo, supondremos que ϕ' es sobreyectivo, o en otras palabras, que $\phi' y_1, \dots, \phi' y_n$ genera $\Pi(\mathbb{R}^3 \setminus K, p'_0)$, suposición que

3. Grupo de un nudo

probaremos que es cierta en la siguiente sección.

Vamos a mostrar con un ejemplo que, dado un camino simple b en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$ es necesario realizar la reflexión al camino cerrado $*b$ para obtener lazos no triviales. Fijémosnos en la parte izquierda de la **Figura 3.14**, en la que se muestra un nudo el cual dividimos en un arco superior y otro inferior. Podemos observar también un camino simple b en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$ tal que $b^b = y_1^{-1}$, y el lazo $*b$ asociado a ese camino simple b .

Como podemos apreciar en la parte izquierda de la figura, el lazo $*b$ en el espacio $\mathbb{R}^3 \setminus K$ es homotópico al lazo trivial p_0 (es decir, al punto p_0), ya que tanto el punto p_0 como la curva recorrida por el camino $*b$ están incluidos en $\{(x, y, z) \in \mathbb{R}^3 : z \geq 0\}$, mientras que el arco inferior cuya proyección intersecta con el camino b , está incluido en $\{(x, y, z) \in \mathbb{R}^3 : z < 0\}$, por ser un arco inferior.

Por lo tanto, al realizar la reflexión del camino cerrado $*b$, como podemos ver en la parte derecha de la **Figura 3.14**, obtenemos el camino cerrado $*b = \mathcal{R}(*b)$, un lazo que pasa por p'_0 y que no es homotópico al lazo trivial p'_0 en $\mathbb{R}^3 \setminus K$, es decir, no pertenecerán a la misma clase de homotopía, que son las clases de equivalencia debidas a la relación de ser homotópico.

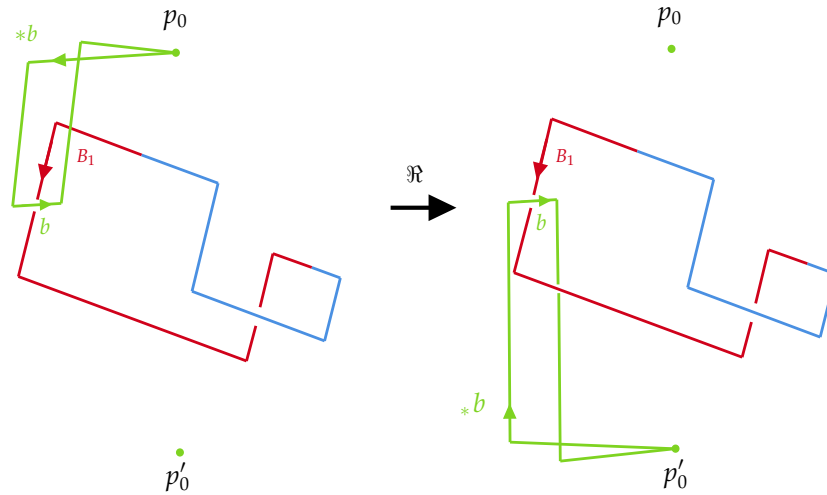


Figura 3.14.: Reflexión del camino cerrado $*b$.

Las imágenes $\mathcal{P}B_i$, $i = 1, \dots, n$ de los arcos inferiores son arcos disjuntos en $\mathbb{R}^2 \times \{0\}$, debido a que ningún arco inferior puede contener un punto superior. Por lo tanto, podemos seleccionar espacios disjuntos, abiertos y simplemente conexos V_1, \dots, V_n en $\mathbb{R}^2 \times \{0\}$ tal que $\mathcal{P}B_i \subset V_i$ para todo $i = 1, \dots, n$. Los bordes de dichos espacios V_1, \dots, V_n son curvas de Jordan lineales a trozos, que serán recorridas en sentido antihorario (visto desde arriba) por los caminos v_1, \dots, v_n respectivamente, que son simples en el sentido antes definido. De manera análoga, elegimos espacios disjuntos, abiertos y simplemente conexos U_1, \dots, U_n en $\mathbb{R}^2 \times \{0\}$ tal que $\mathcal{P}A_j \subset U_j$ para todo $j = 1, \dots, n$, cuyos bordes son curvas de Jordan lineales a trozos, que serán recorridas en sentido horario (visto desde arriba) por los caminos u_1, \dots, u_n , que son simples en el sentido antes definido. Además, dichas regiones V_i, U_j tal

que $i, j = 1, \dots, n$, deberán cumplir que el punto $q_0 \in (\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}K$ que habíamos definido previamente en la página 38, no se encuentre en el cierre de ninguno de esos espacios.

A continuación, vamos a escoger caminos simples c_1, \dots, c_n tales que para todo $i = 1, \dots, n$ cada camino c_i tenga como origen el punto q_0 y como extremo el punto $v_i(0)$, y además cumpla que, $c_i(t) \in (\mathbb{R}^2 \times \{0\}) \setminus \bigcup_{k=1}^n \overline{V_k}$ excepto para $t = 1$, ya que recordemos que un camino está definido entre 0 y 1. De manera análoga, elegimos un conjunto de caminos simples d_1, \dots, d_n tales que para todo $j = 1, \dots, n$ se cumpla que $d_j(0) = q_0$, $d_j(1) = u_j(0)$ y $d_j(t) \in (\mathbb{R}^2 \times \{0\}) \setminus \bigcup_{k=1}^n \overline{U_k}$, a menos que $t = 1$. Estos caminos, obviamente, no son únicos.

En la Figura 3.15 podemos ver unos posibles espacios V_1 y U_2 , señalados con color rojo y azul respectivamente, y unos posibles caminos v_1 , u_2 , c_1 y d_2 para el nudo de la Figura 3.8 y el punto q_0 representado en la siguiente figura.

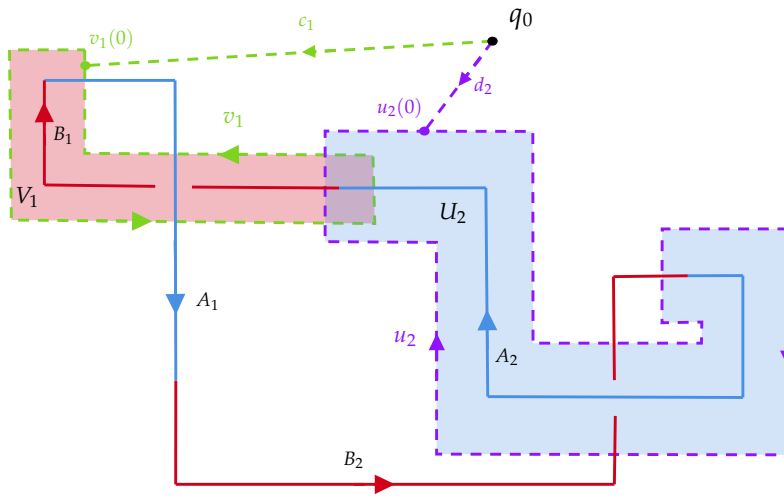


Figura 3.15.: Espacios V_1 y U_2 y caminos v_1, u_2, c_1 y d_2 .

Ya estamos en posición de describir las dos presentaciones del grupo de un nudo poligonal arbitrario, K . La **presentación superior** de $\Pi(\mathbb{R}^3 \setminus K, p_0)$ es

$$\langle x_1, \dots, x_n : r_1, \dots, r_n \rangle_{\phi}, \quad (3.4)$$

donde los n relatores son de la forma $r_i = (c_i \cdot v_i \cdot c_i^{-1})^{\#}$, $i = 1, \dots, n$.

La correspondiente **presentación inferior** de $\Pi(\mathbb{R}^3 \setminus K, p'_0)$ es

$$\langle y_1, \dots, y_n : s_1, \dots, s_n \rangle_{\phi'}, \quad (3.5)$$

donde los n relatores son de la forma $s_j = (d_j \cdot u_j \cdot d_j^{-1})^{\flat}$, $j = 1, \dots, n$.

La validez de las ecuaciones $\phi r_i = 1$ y $\phi' s_j = 1$ tal que $i, j = 1, \dots, n$, es geoméricamente fácil de ver. Previamente habíamos descrito que para cualquier camino simple a en $(\mathbb{R}^2 \times$

3. Grupo de un nudo

$\{0\}) \setminus \mathcal{P}B$, $\phi a^\# = [*a]$, por lo que tenemos que

$$\phi r_i = \phi((c_i \cdot v_i \cdot c_i^{-1})^\#) = [*(c_i \cdot v_i \cdot c_i^{-1})], \quad i = 1, \dots, n,$$

y dado que para cualquier camino simple b en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}A$, $\phi' b^b = [{}_*b]$, obtenemos que

$$\phi' s_j = \phi'((d_j \cdot u_j \cdot d_j^{-1})^b) = [{}_*(d_j \cdot u_j \cdot d_j^{-1})], \quad j = 1, \dots, n.$$

Queremos probar geoméricamente que $\phi r_i = 1$ para todo $i = 1, \dots, n$, que será lo mismo que probar que el lazo $*(c_i \cdot v_i \cdot c_i^{-1})$ es homotópico al lazo trivial p_0 , por lo que probaremos esto último. Dado que el lazo v_i no se intersecta con ninguna de las proyecciones de los arcos inferiores $\mathcal{P}B_1, \dots, \mathcal{P}B_n$, podemos “deslizar” dicho lazo v_i por debajo de B_i en el espacio $\mathbb{R}^3 \setminus K$ hasta contraerlo al punto $v_i(0)$, y trivialmente el lazo $*(c_i \cdot c_i^{-1})$ sería homotópico al lazo trivial p_0 , como queríamos demostrar.

Por ejemplo, tomando el nudo de la **Figura 3.1**, con el que ya hemos tratado anteriormente, y un punto $q_0 \in \mathbb{R}^2 \times \{0\}$ vamos a representar el camino $*(c_1 \cdot v_1 \cdot c_1^{-1})$ mediante líneas discontinuas de color verde. Como habíamos mencionado previamente, este camino no tiene por qué ser único y, puesto que verifica las condiciones descritas anteriormente, es un camino válido. Como podemos observar en la **Figura 3.16**, partiendo de (1), deslizando el camino v_1 por debajo de B_1 llegamos a (2), y de (2) contrayendo dicho lazo se llega trivialmente al punto $v_1(0)$. Por tanto, si llamamos $r_1 = (c_1 \cdot v_1 \cdot c_1^{-1})^\#$, obtenemos que $\phi r_1 = 1$ como queríamos. De manera análoga se probaría para nudos más complejos y diferentes caminos.

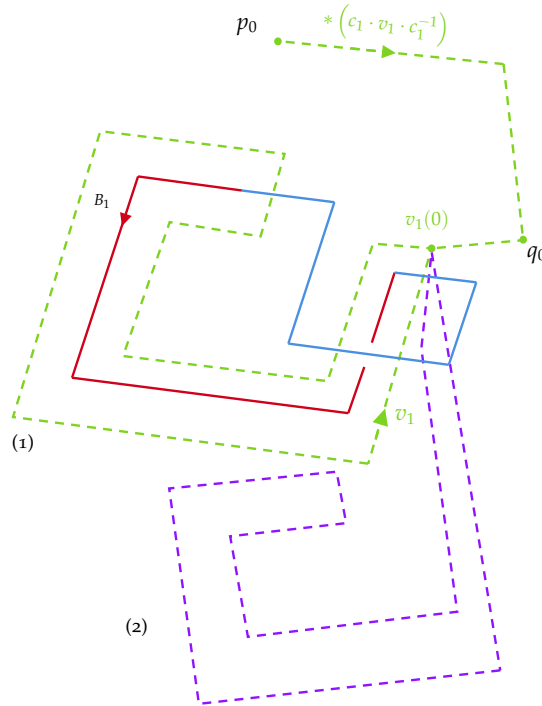


Figura 3.16.: Homotopía entre el lazo $*(c_1 \cdot v_1 \cdot c_1^{-1})$ y el lazo trivial p_0 en $\mathbb{R}^3 \setminus K$.

3.1. Presentación superior e inferior del grupo de un nudo

Un razonamiento parecido se utilizaría para probar que $\phi'(s_j) = 1$ para todo $j = 1, \dots, n$. Sin embargo, queda por probar todavía que las presentaciones superior e inferior calculadas anteriormente son lo que hemos afirmado que eran, presentaciones de los grupos $\Pi(\mathbb{R}^3 \setminus K, p_0)$ y $\Pi(\mathbb{R}^3 \setminus K, p'_0)$ respectivamente. Un teorema importante que obtendremos como consecuencia de dicha prueba será el siguiente:

Teorema 3.1. *En cualquier presentación superior [presentación inferior], cualquiera de los relatores r_1, \dots, r_n [s_1, \dots, s_n] se puede obtener de los $n - 1$ restantes.*

Por tanto, en cualquiera de las dos presentaciones, cualquiera de los relatores puede ser eliminado. Por ejemplo, una presentación del grupo $\Pi(\mathbb{R}^3 \setminus K, p_0)$ podría ser

$$\langle x_1, \dots, x_n : r_1, \dots, r_{n-1} \rangle_\phi,$$

donde se ha eliminado el último relator (podríamos haber eliminado cualquier otro), siendo x_1, \dots, x_n el conjunto de generadores de grupo, y r_1, \dots, r_{n-1}, r_n el conjunto de relatores de la presentación superior del grupo dados por (3.4).

Es consecuencia de este teorema que podamos afirmar que una presentación de $\Pi(\mathbb{R}^3 \setminus K, p_0)$, obtenida de la presentación superior del grupo de K , sería

$$\langle x_1, \dots, x_n : r_1, \dots, \hat{r}_k, \dots, r_n \rangle_\phi,$$

donde \hat{r}_k indica el borrado del k -ésimo relator r_k y ϕ es el homomorfismo (3.2). Equivalentemente, una presentación de $\Pi(\mathbb{R}^3 \setminus K, p_0)$, obtenida de la presentación inferior del grupo de K , sería

$$\langle y_1, \dots, y_n : s_1, \dots, \hat{s}_j, \dots, s_n \rangle_{\phi'},$$

donde \hat{s}_j indica el borrado del j -ésimo relator s_j y ϕ' es el homomorfismo (3.3).

Vamos a comprobar con un ejemplo que efectivamente podemos eliminar cualquier relator de la presentación superior e inferior del grupo de un nudo. Tomaremos el nudo de la Figura 3.8 y fijaremos un punto $q_0 \in (\mathbb{R}^2 \times \{0\}) \setminus \mathcal{PK}$.

Para hallar los relatores de la presentación superior del grupo del nudo representamos los caminos $c_1 \cdot v_1 \cdot c_1^{-1}$ y $c_2 \cdot v_2 \cdot c_2^{-1}$, como se muestra en la Figura 3.17.

3. Grupo de un nudo

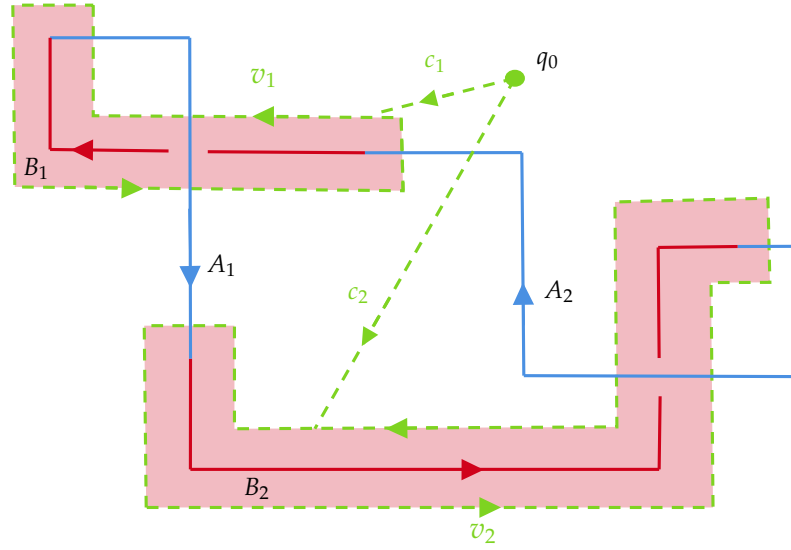


Figura 3.17.: Caminos $c_1 \cdot v_1 \cdot c_1^{-1}$ y $c_2 \cdot v_2 \cdot c_2^{-1}$.

En este caso, dado que hay dos arcos inferiores, la presentación superior del grupo del nudo va a tener dos relatores, $r_1 = (c_1 \cdot v_1 \cdot c_1^{-1})^\#$ y $r_2 = (c_2 \cdot v_2 \cdot c_2^{-1})^\#$. En consecuencia, vamos a tratar de verificar lo que nos afirma el Teorema 3.1, que r_1 se deduce de r_2 (o que r_2 se deduce de r_1). Calculemos dichos relatores:

$$r_1 = x_1 x_1^{-1} x_1^{-1} x_2 = x_1^{-1} x_2$$

$$r_2 = x_2^{-1} (x_1 x_2 x_2^{-1} x_2^{-1}) x_2 = x_2^{-1} x_1$$

Así pues, tenemos que las relaciones de la presentación superior son:

$$x_1^{-1} x_2 = 1 \quad \wedge \quad x_2^{-1} x_1 = 1$$

Sin embargo, si multiplicamos la primera expresión por $x_2^{-1} x_1$ por la derecha obtenemos la segunda, por lo que hemos comprobado que los relatores r_1 y r_2 se pueden obtener el uno del otro, como queríamos ver.

Para obtener los relatores de la presentación inferior del grupo del mismo nudo consideremos por ejemplo los caminos $d_1 \cdot u_1 \cdot d_1^{-1}$ y $d_2 \cdot u_2 \cdot d_2^{-1}$ representados en la Figura 3.18.

3.2. Presentación superior e inferior del grupo de un nudo, continuación

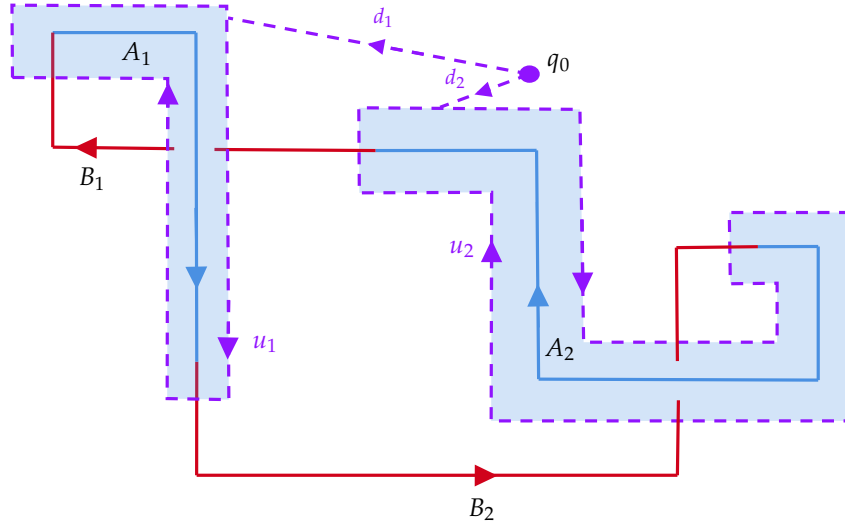


Figura 3.18.: Caminos $d_1 \cdot u_1 \cdot d_1^{-1}$ y $d_2 \cdot u_2 \cdot d_2^{-1}$.

Los relatores de dicha presentación inferior serán $s_1 = (d_1 \cdot u_1 \cdot d_1^{-1})^b$ y $s_2 = (d_2 \cdot u_2 \cdot d_2^{-1})^b$ que si los calculamos resultarán ser:

$$s_1 = y_1^{-1} y_2 y_1 y_1^{-1}$$

$$s_2 = y_2 y_2^{-1} y_2^{-1} y_1$$

Por tanto, las relaciones que se obtienen de dichos relatores son:

$$y_1^{-1} y_2 = 1 \quad \wedge \quad y_2^{-1} y_1 = 1$$

No obstante, si multiplicamos la primera expresión por la derecha por $y_2^{-1} y_1$ obtenemos la segunda, por lo que los relatores s_1 y s_2 se obtienen uno a partir del otro, como habíamos anticipado que sucedería.

3.2. Presentación superior e inferior del grupo de un nudo, continuación

En esta sección vamos a probar que la presentación superior e inferior dadas por (3.4) y (3.5) son presentaciones de $\Pi(\mathbb{R}^3 \setminus K, p_0)$ y $\Pi(\mathbb{R}^3 \setminus K, p'_0)$ respectivamente, como hemos afirmado anteriormente que eran.

3.2.1. Caracterización de la presentación inferior del grupo de un nudo

La primera observación que debemos tener en cuenta es que, no es necesario dar diferentes pruebas para ambas presentaciones. Una presentación inferior puede ser caracterizada en términos de una presentación superior, simplemente invirtiendo la orientación del nudo K e intercambiando los roles de arriba y abajo. Una vez probemos esto último nos podremos centrar en la prueba de que la presentación superior dada por (3.4) es una presentación de

3. Grupo de un nudo

$\Pi(\mathbb{R}^3 \setminus K, p_0)$, desatendiendo la misma prueba para la presentación inferior, al poder ser caracterizada en términos de una presentación superior.

Para demostrar que una presentación inferior puede ser caracterizada en términos de una presentación superior definámonos $h: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ como la rotación de 180° alrededor del eje x , definida por $h(x, y, z) = (x, -y, -z)$. Tomamos $K' = hK$ y tenemos que el homeomorfismo h induce el isomorfismo $h_*: \Pi(\mathbb{R}^3 \setminus K', p_0) \rightarrow \Pi(\mathbb{R}^3 \setminus K, p'_0)$. Además, consideramos como dirección positiva sobre K' la opuesta de la inducida por h de la orientación del nudo K . Hablando de forma simple, le damos la vuelta al nudo K y cambiamos el sentido de la flecha. Sea

$$\langle y_1, \dots, y_n : s_1, \dots, s_n \rangle_\zeta \quad (3.6)$$

una presentación superior de $\Pi(\mathbb{R}^3 \setminus K', p_0)$, construida de la misma forma que lo hicimos en (3.4), habiendo tenido en cuenta las siguientes consideraciones. Dados los arcos superiores A_1, \dots, A_n e inferiores B_1, \dots, B_n del nudo poligonal K , hemos asumido que, para $j = 1, \dots, n$, el j -ésimo arco superior de K' es hB_j , que es un arco superior de K' debido a que B_j es un arco inferior de K y la imagen por h de un arco superior es un arco inferior y viceversa. Por esta última razón, hemos asumido también para el cálculo de (3.6) que el i -ésimo relator de la presentación superior del grupo de K' es obtenido de rodear al arco $\mathcal{P}hA_i$ en $\mathbb{R}^2 \times \{0\}$, y finalmente, que hq_0 es el punto base común en $\mathbb{R}^2 \times \{0\}$ de los lazos que determinan los relatores de la presentación superior del grupo de K' . Entonces, tenemos que

$$|y_1, \dots, y_n : s_1, \dots, s_n|_{h_*\zeta} = \Pi(\mathbb{R}^3 \setminus K, p'_0),$$

por lo que coincide con (3.5) y damos por terminada la prueba.

Vamos a ver con un ejemplo que esto es obvio. En la Figura 3.19 se muestra el nudo obtenido de girar el nudo de la Figura 3.18 180° alrededor del eje x al aplicarle al nudo la función h definida previamente e invertir la orientación de este. En dicha figura se muestran también los caminos hu_1 y hu_2 , es decir, la rotación de los lazos u_1 y u_2 representados en la Figura 3.18, necesarios para calcular la presentación superior del grupo de K' .

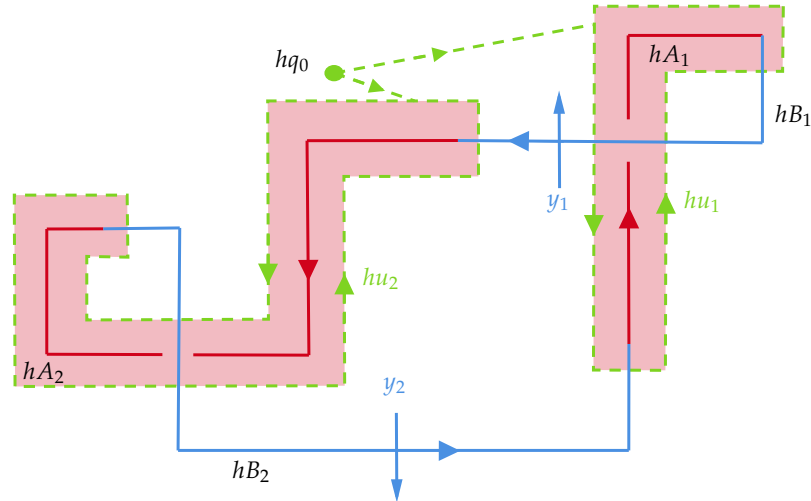


Figura 3.19.: Rotación de un nudo K mediante la aplicación h , y caminos hu_1 y hu_2 .

3.2. Presentación superior e inferior del grupo de un nudo, continuación

A continuación, vamos a calcular los relatores de dicha presentación superior. Dichos relatores serán

$$s_1 = y_1^{-1} y_2 y_1 y_1^{-1},$$

$$s_2 = y_2 y_2^{-1} y_2^{-1} y_1,$$

que como podemos observar, son los mismos relatores que los que habíamos obtenido en la presentación inferior del grupo del nudo de la **Figura 3.18**, calculados previamente a continuación de dicha figura.

3.2.2. Grupo fundamental de un espacio topológico

En esta subsección, vamos estudiar diferentes herramientas para obtener algunos resultados acerca del grupo fundamental de un espacio topológico. En primer lugar, vamos a enunciar el Teorema de Seifert-Van Kampen, que veremos que nos será de vital utilidad para calcular una presentación del grupo de un nudo poligonal arbitrario K . Antes de ello, necesitaremos definir algunos conceptos que serán claves para poder enunciar dicho teorema.

Definición 3.13. Dado un grupo G , y A un subconjunto suyo tenemos que

$$N(A) := \bigcap_{A \subseteq H \trianglelefteq G} H$$

es el menor subgrupo normal de G conteniendo a A , también llamado el subgrupo normal de G generado por A . $H \trianglelefteq G$ denota que H es un subgrupo normal de G .

Definición 3.14. Consideremos tres grupos A, G_1 y G_2 y los homomorfismos $\phi_1: A \rightarrow G_1$ y $\phi_2: A \rightarrow G_2$. Se define el producto libre amalgamado $G_1 \bullet_A G_2$ como el grupo cociente $(G_1 \bullet G_2)/N(R)$, siendo $R = \{\phi_1(a)\phi_2(a)^{-1} : a \in A\}$. El símbolo \bullet denotará el conjunto de palabras reducidas construidas sobre los dos grupos con los que opera.

Ya podemos enunciar el Teorema de Seifert-Van Kampen, para lo que será importante tener en cuenta las definiciones anteriores.

Teorema 3.2. (Seifert-Van Kampen) Sea X un espacio topológico y sean $U, V \subset X$ subconjuntos tales que:

- $U \cap V \neq \emptyset$ y $U \cup V = X$.
- U, V y $U \cap V$ son abiertos arcoconexos.

Definimos las aplicaciones inclusión $i: U \cap V \rightarrow U$ y $j: U \cap V \rightarrow V$, fijamos $x_0 \in U \cap V$ y consideramos los correspondientes homomorfismos inducidos $i_*: \Pi(U \cap V, x_0) \rightarrow \Pi(U, x_0)$ y $j_*: \Pi(U \cap V, x_0) \rightarrow \Pi(V, x_0)$.

Entonces $\Pi(X, x_0) \cong \Pi(U, x_0) \bullet_{(\Pi(U \cap V, x_0))} \Pi(V, x_0)$ donde el producto amalgamado es el relativo a los homomorfismos i_* y j_* , es decir,

$$\Pi(X, x_0) \cong \frac{\Pi(U, x_0) \bullet \Pi(V, x_0)}{N(R)},$$

3. Grupo de un nudo

donde $R = \{i_*([\gamma])j_*([\gamma])^{-1} : [\gamma] \in \Pi(U \cap V, x_0)\} = \{i_*([\gamma])j_*([\bar{\gamma}]) : [\gamma] \in \Pi(U \cap V, x_0)\}.$

Como bien sabemos, dos espacios homeomorfos tienen el mismo grupo fundamental, salvo isomorfismos. Pero existen condiciones topológicas más débiles que nos permiten llegar a la misma conclusión, que estudiaremos a continuación.

Definición 3.15. Dado un espacio topológico X y un subespacio topológico $A \subset X$, una **retracción** de X en A es una aplicación continua $r: X \rightarrow A$ cumpliendo que $r|_A = Id_A$. En este caso, se dice que A es un **retracto** de X .

Es interesante el caso en el que un espacio puede ser deformado en un retracto suyo de forma continua. Este concepto se trata en la siguiente definición.

Definición 3.16. Dado un espacio topológico X y un subespacio $A \subset X$, se dice que A es un **retracto de deformación** de X si existen una retracción $r: X \rightarrow A$ y una función continua $H: X \times [0, 1] \rightarrow X$ que cumplan:

- $H(x, 0) = x$ para todo $x \in X$.
- $H(x, 1) = r(x)$ para todo $x \in X$.

Si además

$$H(a, s) = a \quad \text{para todo } (a, s) \in A \times [0, 1],$$

entonces se dice que A es un **retracto fuerte de deformación** de X .

A las aplicaciones H y r se les llamará **deformación** y **retracción** asociadas al retracto de deformación A de X , respectivamente.

Por ejemplo, el centro de un círculo es un retracto fuerte de deformación de este. Si denotamos al círculo por X y a su centro por p_0 , existe una retracción $r: X \rightarrow A$, siendo $A = \{p_0\}$, dada por

$$r(x) = p_0 \quad \text{para todo } x \in X,$$

y una deformación $H: X \times [0, 1] \rightarrow X$ dada por

$$H(x, s) := (1 - s)x + sp_0 \quad \text{para todo } (x, s) \in X \times [0, 1],$$

que cumplen las condiciones mencionadas anteriormente para que A (el centro del círculo) sea un retracto fuerte de deformación de X (el círculo).

En la parte izquierda de la **Figura 3.20** se puede ver como actúa la retracción r sobre cada punto del círculo, y en la parte derecha vemos el proceso de deformación continuo del espacio total X ($s = 0$) en el retracto $\{p_0\} = A \subset X$ ($s = 1$), donde si interpretamos el parámetro s como medida del tiempo transcurrido en la deformación susodicha, $H_s(X) \subset X$ tal que $s \in [0, 1]$, reflejará el subespacio de X en el instante s .

3.2. Presentación superior e inferior del grupo de un nudo, continuación

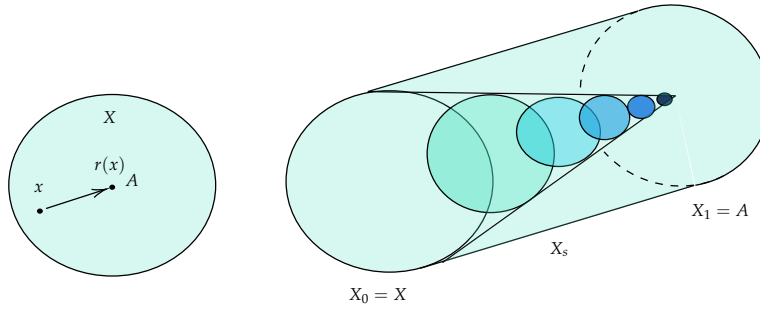


Figura 3.20.: Retracción r y deformación H .

Como habíamos mencionado anteriormente, estamos buscando condiciones topológicas que nos permitan asegurar cuando dos espacios van a tener el mismo grupo fundamental (salvo isomorfismos). El siguiente teorema nos mostrará como la condición de ser retracto fuerte de deformación, asegura el mismo grupo fundamental para los dos espacios topológicos implicados.

Teorema 3.3. Sea X un espacio topológico, y $A \subset X$ un retracto fuerte de deformación de X con retracción asociada $r: X \rightarrow A$. Entonces, dado un punto $a \in A$ tenemos que

$$r_*: \Pi(X, a) \rightarrow \Pi(A, a) \quad e \quad i_*: \Pi(A, a) \rightarrow \Pi(X, a)$$

son isomorfismos, siendo uno el inverso del otro.

En consecuencia, tanto un espacio topológico como un retracto fuerte de deformación suyo van a tener el mismo grupo fundamental asociado.

Definición 3.17. Un espacio topológico X se dice **simplemente conexo** si es arcoconexo y $\Pi(X, p) = \{[\epsilon_p]\}$ para algún $p \in X$ y por tanto para todo $p \in X$ (por el Corolario 3.1).

3.2.3. Cálculo de la presentación superior del grupo de un nudo

Vamos por tanto a comenzar con el proceso de prueba de que la presentación dada por (3.4) es una presentación de $\Pi(\mathbb{R}^3 \setminus K, p_0)$.

Primeramente, dado un nudo poligonal K , consideramos un **cuadrado cerrado** (frontera + interior) S paralelo al plano $\mathbb{R}^2 \times \{0\}$, situado “debajo de K ”, y tal que

$$\mathcal{P}K \cup \{q_0\} \cup \bigcup_{i=1}^n V_i \subset \mathcal{P}S. \quad (3.7)$$

Es decir, el cuadrado S tiene que tener coordenada z menor que todos los puntos del nudo K y, al proyectarlo en $\mathbb{R}^2 \times \{0\}$, tiene que contener tanto a la proyección del nudo, como al punto q_0 , como a los espacios abiertos V_i para $i = 1, \dots, n$.

Para cualquier subconjunto L del nudo K , vamos a definir el subconjunto L_* como la unión del nudo poligonal K , el cuadrado cerrado S y el conjunto de todos los puntos que se

3. Grupo de un nudo

encuentren entre S y L , es decir, la unión de K , S y el conjunto

$$\{(x, y, z) \in \mathbb{R}^3 : \exists z_1, z_2 \in \mathbb{R} \text{ tal que } (x, y, z_1) \in S \wedge (x, y, z_2) \in L \wedge z_1 \leq z \leq z_2\}. \quad (3.8)$$

Por ejemplo, Q_* es la unión de K , S y de los $2n$ segmentos verticales que unen los puntos de Q con sus correspondientes puntos de S descritos por (3.8). Recordemos que Q era el conjunto de puntos que separaba el nudo en arcos superiores e inferiores. Por ejemplo, el espacio Q_* del nudo de la Figura 3.8 es el espacio representado en la Figura 3.21, en la cual los puntos de Q están señalados con color verde, el cuadrado cerrado S está representado con color gris, y los puntos del conjunto (3.8) son los puntos pertenecientes a las aristas verticales representadas con color negro.

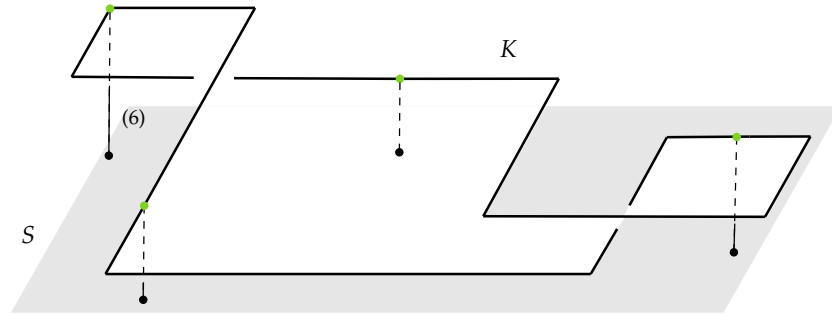


Figura 3.21.: Q_*

Una vez definidos los conjuntos anteriores y habiendo enunciado ya el Teorema de Seifert-Van Kampen, vamos a seguir una serie de pasos que nos llevarán al que es el objetivo principal de esta sección, dado un nudo poligonal arbitrario K dividido en n arcos superiores e inferiores, obtener una presentación de $\Pi(\mathbb{R}^3 \setminus K, p_0)$, a la que llamaremos presentación superior del grupo de K . Estos pasos serán los pasos que seguiremos:

1. Primeramente, veremos que el espacio $\mathbb{R}^3 \setminus K_*$ es simplemente conexo.
2. En segundo lugar, añadiremos el conjunto $K_* \setminus B_*$ a $\mathbb{R}^3 \setminus K_*$, y probaremos que el grupo $\Pi(\mathbb{R}^3 \setminus B_*, p_0)$ es un grupo libre de rango n .
3. Por último, añadiremos el conjunto $B_* \setminus K$ a $\mathbb{R}^3 \setminus B_*$ y conseguiremos dar una presentación de $\Pi(\mathbb{R}^3 \setminus K, p_0)$, concretamente, la presentación superior del grupo de K que habíamos descrito en (3.4).

Vamos a probar la primera (1) de las afirmaciones descritas previamente.

Proposición 3.7. *El espacio $\mathbb{R}^3 \setminus K_*$ es simplemente conexo.*

Demostración. El resultado es geoméricamente trivial, pero para demostrarlo formalmente vamos a aplicar el Teorema de Seifert-Van Kampen. Denotaremos por X al conjunto de todos los puntos de $\mathbb{R}^3 \setminus K_*$ que no se encuentran debajo del cuadrado S dado por (3.7). Es decir, si denotamos por S' al conjunto de todos los puntos (x, y, z') tales que $(x, y, z) \in S$ y $z' < z$, entonces

$$X = (\mathbb{R}^3 \setminus K_*) \setminus S'. \quad (3.9)$$

Por ejemplo, si S es el cuadrado relleno representado con color gris en la Figura 3.22, S' será el bloque relleno representado con color granate, que se extendería hasta $z = -\infty$.

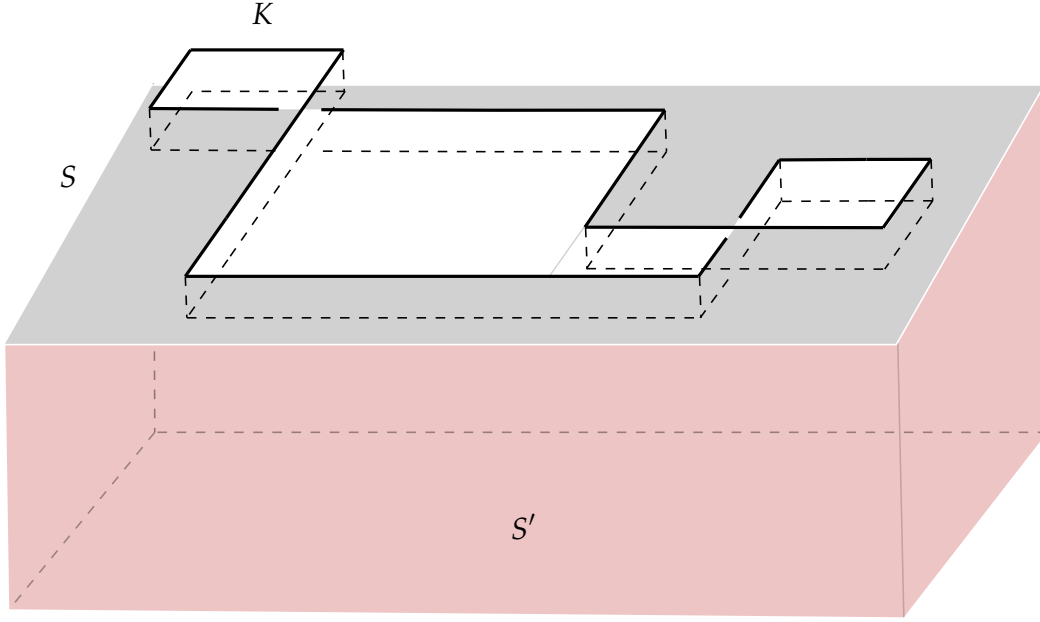


Figura 3.22.: Espacio S' .

Vamos a probar que X es simplemente conexo, concretamente vamos a ver que el punto base $p_0 = (0, 0, z_0)$ es un retracto de deformación de X . Este hecho se obtiene de las deformaciones

$$\begin{aligned} h_s(x, y, z) &= (x, y, (1-s)z + sz_0), \quad 0 \leq s \leq 1, \\ k_s(x, y, z_0) &= ((1-s)x, (1-s)y, z_0), \quad 0 \leq s \leq 1. \end{aligned}$$

Si denotamos por Z al plano horizontal que contiene al punto base p_0 , que obviamente está contenido en X , la primera aplicación, h_s , es la deformación asociada al retracto de deformación Z de X . Es decir, la aplicación h_s aplica verticalmente un punto de X en su punto asociado del plano horizontal Z . La segunda deformación, k_s , aplica linealmente todos los puntos de este plano Z en el punto base p_0 .

Por ejemplo, dado el nudo K de la Figura 3.8, tomando un espacio S cumpliendo (3.7) y el espacio X asociado dado por (3.9), en la Figura 3.23 podemos observar que dado cualquier punto de X , mediante la aplicación de las dos deformaciones anteriores obtenemos p_0 .

3. Grupo de un nudo

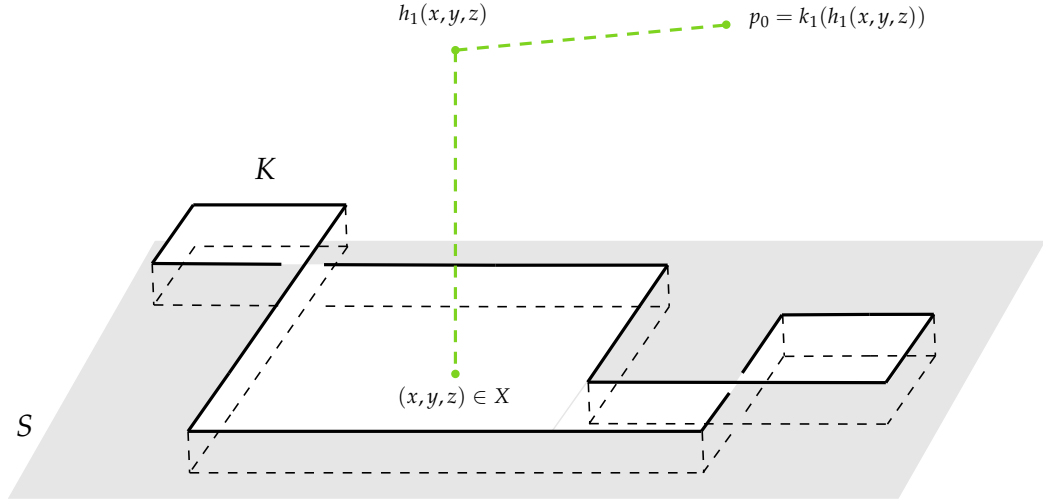


Figura 3.23. Aplicación de las deformaciones h_s y k_s a un punto de X .

Tomamos ahora el espacio consistente en los puntos que se encuentran bajo el plano horizontal que contiene al cuadrado S , que denotaremos por Y y que es trivialmente simplemente conexo. Por ejemplo, para el nudo K y el espacio S tomados en la [Figura 3.23](#), si suponemos que la coordenada z de los puntos del cuadrado S es $z_0 \in \mathbb{R}$, el espacio Y sería el espacio $\{(x, y, z) \in \mathbb{R}^3 : z < z_0\}$, representado con color azul en la [Figura 3.24](#).

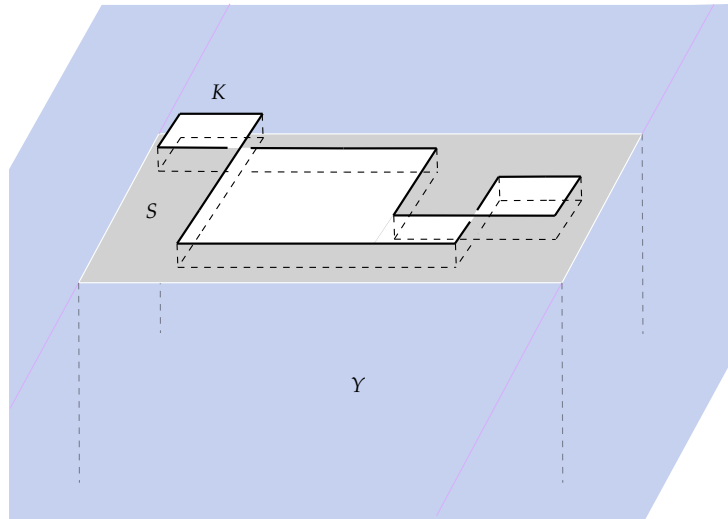


Figura 3.24. Espacio Y .

Dado que

$$X \cup Y = \mathbb{R}^3 \setminus K_*$$

y que los conjuntos X, Y , y $X \cap Y$ son no vacíos, abiertos y arcoconexos en $X \cup Y$, podemos aplicar el Teorema de Seifert-Van Kampen para hallar el grupo fundamental de $\mathbb{R}^3 \setminus K_*$ en un punto $x_0 \in X \cap Y$. Sin embargo, puesto que X e Y son simplemente conexos, $\mathbb{R}^3 \setminus K_* = X \cup Y$

es simplemente conexo como queríamos demostrar. \square

Una vez demostrado (1) vamos a pasar a definir una serie de espacios que nos ayudarán a demostrar la segunda (2) de las afirmaciones descritas previamente.

El conjunto $K_* \setminus B_*$ es la unión disjunta de n discos topológicos abiertos F_1, \dots, F_n , de forma que para $j = 1, \dots, n$, F_j consiste en aquellos puntos que se encuentran entre $A_j \setminus Q$ y S , menos cualquier punto que se encuentre debajo de un arco inferior, es decir, menos cualquier punto que se encuentre bajo un punto inferior. Para todo $j = 1, \dots, n$, sea a_j un camino simple cualquiera en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$, que tan solo se intersecte con la proyección del arco superior $\mathcal{P}A_j$, haciéndolo tan solo una vez y formando un tornillo de la mano derecha con la proyección de dicho arco superior. Por ello, dicho camino a_j intersectará a F_j una vez y el resto estará contenido en $\mathbb{R}^3 \setminus K_*$.

En la Figura 3.25 se puede observar el nudo con el que hemos estado tratando ultimamente, dividido en arcos superiores e inferiores. También se pueden ver representados con color lila los dos discos abiertos F_1 y F_2 , asociados a los arcos superiores A_1 y A_2 respectivamente y un camino simple a_2 en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ que, como su número asociado indica, se intersecta una vez con $\mathcal{P}A_2$ formando un tornillo de la mano derecha, de forma que el lazo $*a_2$ intersecta a F_2 una vez y el resto está contenido en $\mathbb{R}^3 \setminus K_*$.

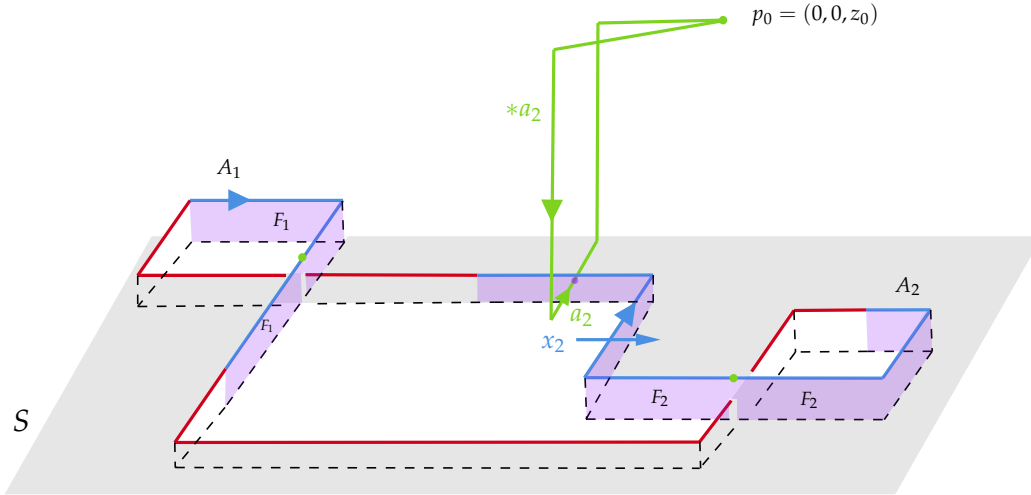


Figura 3.25.: Espacios F_1 y F_2 y camino $*a_2$.

Para $j = 1, \dots, n$ denotaremos por W_j a un entorno abierto de F_j y del conjunto de los puntos de $*a_j$, elegido de forma que:

- I. W_j es arcoconexo, y el grupo $\Pi(W_j, p_0)$ es un grupo cíclico infinito generado por la clase de equivalencia de $*a_j$ en W_j .
- II. $W_j \cap K_* = F_j$.
- III. $W_j \setminus K_*$ es simplemente conexo.

3. Grupo de un nudo

Que estos conjuntos pueden ser construídos es geoméricamente obvio. Cada W_j es simplemente la unión de F_j y la imagen de $*a_j$, ambos ligeramente engrosados a un abierto de \mathbb{R}^3 . Más explícitamente, sea $\epsilon > 0$ tan pequeño que el ϵ -entorno abierto W'_j de la imagen de $*a_j$ satisfaga que $W'_j \cap K_* \subset F_j$. Sea W''_j el conjunto de puntos cuya distancia a F_j es menor que ϵ y que están más cerca de F_j que de $K_* \setminus F_j$. La condición de que los puntos de W''_j estén más cerca de F_j que de $K_* \setminus F_j$ es necesaria para que ningún punto del nudo K ni del cuadrado cerrado S estén contenidos en W''_j , ya que en caso contrario se violaría la condición (II).

No es trivial probar que $W_j = W'_j \cup W''_j$ satisface las condiciones (I), (II) y (III), pero es factible dicha prueba.

Centrándonos en el espacio F_2 y el lazo $*a_2$ de la Figura 3.25, tomando un valor $\epsilon > 0$ suficientemente pequeño, tenemos que el espacio W'_2 será el espacio representado con color marrón en la Figura 3.26a, es decir, el “tubo” de radio ϵ con centro los puntos del lazo $*a_2$. El espacio W''_2 será el representado con color marrón en la Figura 3.26b, donde todos los puntos de W''_2 se encuentran a distancia menor que ϵ de F_2 y están más cerca de F_2 que de $K_* \setminus F_2$.

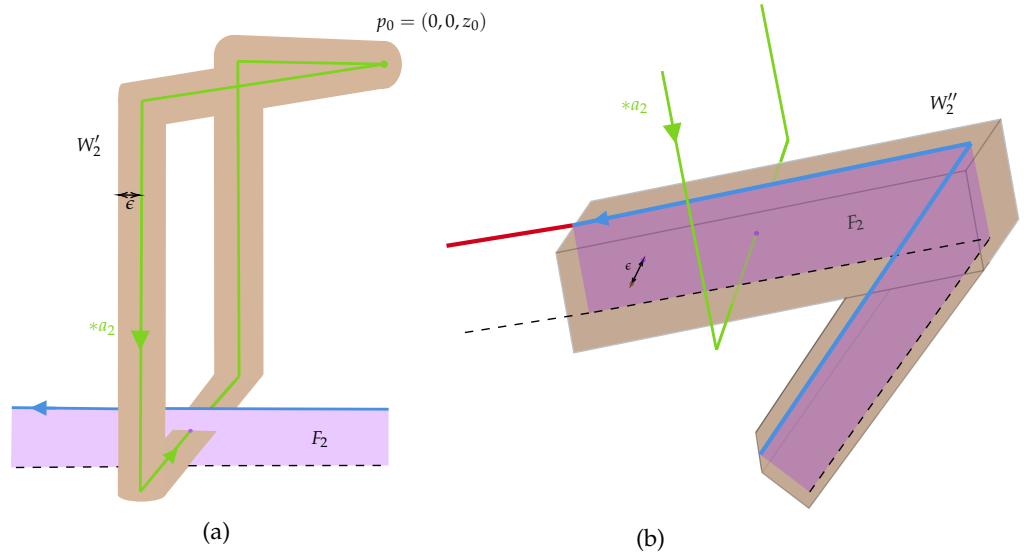


Figura 3.26.: (a) Espacio W'_2 . (b) Espacio W''_2 .

Una vez definidos los espacios anteriores ya estamos en disposición de demostrar (2).

Proposición 3.8. $\mathbb{R}^3 \setminus B_*$ es arcoconexo y $\Pi(\mathbb{R}^3 \setminus B_*, p_0)$ es un grupo libre. Además, el conjunto de clases de equivalencia x_1, \dots, x_n de los lazos $*a_1, \dots, *a_n$ es una base del grupo libre.

Demostración. Tomamos

$$X_0 = \mathbb{R}^3 \setminus K_*,$$

$$X_j = X_{j-1} \cup W_j, \quad j = 1, \dots, n.$$

3.2. Presentación superior e inferior del grupo de un nudo, continuación

Dado que $W_j \cap K_* = F_j$, entonces

$$X_0 \cup W_j = (\mathbb{R}^3 \setminus K_*) \cup W_j = \mathbb{R}^3 \setminus (K_* \setminus W_j) = \mathbb{R}^3 \setminus (K_* \setminus F_j) = (\mathbb{R}^3 \setminus K_*) \cup F_j,$$

y por lo tanto, puesto que $\{F_i : i = 1, \dots, n\}$ son disjuntos dos a dos, obtenemos que

$$X_j = (\mathbb{R}^3 \setminus K_*) \cup \bigcup_{k=1}^j F_k, \quad j = 1, \dots, n. \quad (3.10)$$

En particular, puesto que $\bigcup_{k=1}^n F_k = K_* \setminus B_*$ y por (3.10) tenemos que

$$X_n = (\mathbb{R}^3 \setminus K_*) \cup (K_* \setminus B_*) = \mathbb{R}^3 \setminus B_*,$$

que es el espacio para el cual queremos hallar su grupo fundamental. Además, utilizando la propiedad (ii) de W_j obtenemos que

$$F_k \cap W_j = K_* \cap W_k \cap W_j = K_* \cap W_k \cap W_j \cap K_* = F_k \cap F_j = \emptyset, \quad k \neq j,$$

y se sigue por (3.10) y por la igualdad anterior que

$$\begin{aligned} X_{j-1} \cap W_j &= ((\mathbb{R}^3 \setminus K_*) \cup \bigcup_{k=1}^{j-1} F_k) \cap W_j \\ &= ((\mathbb{R}^3 \setminus K_*) \cap W_j) \cup ((\bigcup_{k=1}^{j-1} F_k) \cap W_j) \\ &= (\mathbb{R}^3 \setminus K_*) \cap W_j \\ &= W_j \setminus K_*. \end{aligned}$$

En consecuencia, por la propiedad (iii) de W_j , tenemos que $X_{j-1} \cap W_j$ es simplemente conexo.

Gracias a la **Proposición 3.7** tenemos que X_0 es simplemente conexo, por lo que tomamos como hipótesis de inducción que X_{j-1} es arcoconexo, que $\Pi(X_{j-1}, p_0)$ es un grupo libre y que el conjunto de clases de equivalencia de los lazos $*a_1, \dots, *a_{j-1}$ en X_{j-1} es la base de dicho grupo libre. Dado que X_{j-1} , W_j y $X_{j-1} \cap W_j$ son arcoconexos, no vacíos, subespacios abiertos de $X_j = X_{j-1} \cup W_j$ tales que

- por (i), el grupo de W_j es un grupo libre con base la clase de equivalencia del lazo $*a_j$ en W_j ,
- el grupo de X_{j-1} un grupo libre con base las clases de equivalencia de $*a_0, \dots, *a_{j-1}$ en X_{j-1} ,
- el grupo de $X_{j-1} \cap W_j$ es trivial por ser dicho espacio simplemente conexo,

tenemos que por el Teorema de Seifert-Van Kampen, el grupo de $X_{j-1} \cup W_j$ es libre y una base de dicho grupo libre es $\{[*a_0a_j], [*a_1a_j], \dots, [*a_{j-1}a_j]\}$, o lo que es lo mismo,

3. Grupo de un nudo

$\{[*a_0], [*a_1], \dots, [*a_j]\}$, donde los corchetes indican la clase de equivalencia de un lazo en X_j . El proceso de inducción completa la prueba. \square

Los elementos x_1, \dots, x_n introducidos en la proposición anterior, asociados a los lazos $*a_1, \dots, *a_n$, son los elementos de la base libre del grupo $\Pi(\mathbb{R}^3 \setminus B_*, p_0)$, aunque también serán los generadores de la presentación superior del grupo del nudo que calcularemos en esta sección. Dicho con otras palabras, el grupo $\Pi(\mathbb{R}^3 \setminus B_*, p_0)$ es el grupo libre de dicha presentación. Fijémonos en que, dado

$$\phi : \Pi(\mathbb{R}^3 \setminus B_*, p_0) \rightarrow \Pi(\mathbb{R}^3 \setminus K, p_0) \quad (3.11)$$

el homomorfismo inducido por la inclusión, el elemento ϕx_j , $j = 1, \dots, n$, es la clase de equivalencia en $\mathbb{R}^3 \setminus K$ del lazo $*a_j$. Por tanto, dicho homomorfismo ϕ definido en (3.11) es idéntico al homomorfismo denotado por el mismo símbolo en (3.2). Para cualquier camino simple a en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$, consideraremos al elemento $a^\#$ como la clase de equivalencia en $\mathbb{R}^3 \setminus B_*$ del lazo $*a$.

Lo último que vamos a analizar es que el homomorfismo ϕ es sobreyectivo y que el núcleo de ϕ es consecuencia de cualquiera $n - 1$ relatores $r_1, \dots, \hat{r}_k, \dots, r_n$ de la presentación dada por (3.4), donde \hat{r}_k indica la no inclusión del relator r_k . Una vez que probemos esto último ya podremos afirmar que la presentación dada por (3.4) es lo que habíamos afirmado que era, una presentación del grupo de K .

Para la prueba, seguiremos un procedimiento similar al que hemos utilizado para calcular el grupo fundamental de $\Pi(\mathbb{R}^3 \setminus B_*, p_0)$ en la **Proposición 3.8**. Añadiremos a dicho espacio cuyo grupo fundamental ya conocemos un entorno abierto apropiado del conjunto $B_* \setminus K$, cuya unión resultará ser $\mathbb{R}^3 \setminus K$, el espacio para el que queremos calcular el grupo fundamental.

Para ello, consideramos un **prisma rectangular** (una caja rectancular hueca) que denotaremos por T de forma que contenga en su interior al cuadrado cerrado S y el nudo poligonal K se encuentre en el exterior de dicho prisma T . Topológicamente, sabemos que T es homeomorfo a una esfera, un homeomorfismo que utilizaremos más adelante para obtener de forma más sencilla el grupo fundamental de algunos espacios.

Por ejemplo, un posible espacio T para el nudo que estamos tratando podría ser el prisma rectangular hueco representado con color granate en la **Figura 3.27**, en la que el nudo K se encuentra en el exterior de dicho prisma.

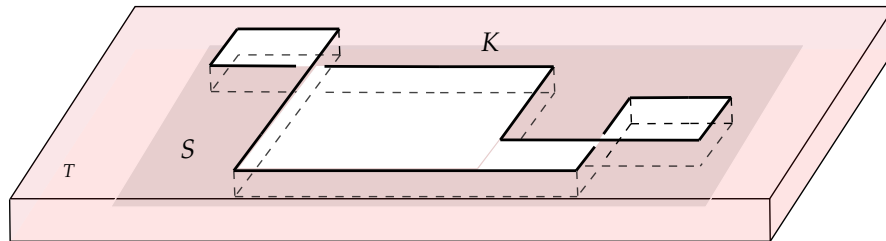


Figura 3.27.: Prisma rectangular T .

Sea W un entorno abierto de $B_* \setminus K$ elegido de forma que:

- I. W es simplemente conexo y contiene a p_0 .
- II. $W \cap K = \emptyset$.
- III. $T \setminus B_*$ es un retracto de deformación de $W \setminus B_*$.

Un conjunto W que cumpla las tres condiciones anteriores puede ser construido de diferentes formas. Por ejemplo, conectamos T a p_0 con un arco poligonal E que sea disjunto de K_* . Sea W' la unión de un ϵ -entorno de $T \cup E$ y los puntos que se encuentran en el interior de T , y sea W'' el conjunto de todos los puntos cuya distancia a B_* es menor que ϵ y que se encuentran más cerca de $B_* \setminus K$ que de K . La condición de que los puntos de W'' estén más cerca de $B_* \setminus K$ que de K , es para asegurar que ningún punto de K esté contenido en W'' . Para un ϵ suficientemente pequeño, podemos tomar W como $W' \cup W''$ y cumplirá las tres condiciones anteriores.

Para el nudo que estamos manejando, tenemos que el espacio W' podría ser el representado con color verde en la [Figura 3.28](#), es decir, la unión de un ϵ -entorno de $T \cup E$ para un ϵ suficientemente pequeño y los puntos del interior de T , siendo E un arco poligonal arbitrario que conecte un punto de T con p_0 y disjunto de K_* .

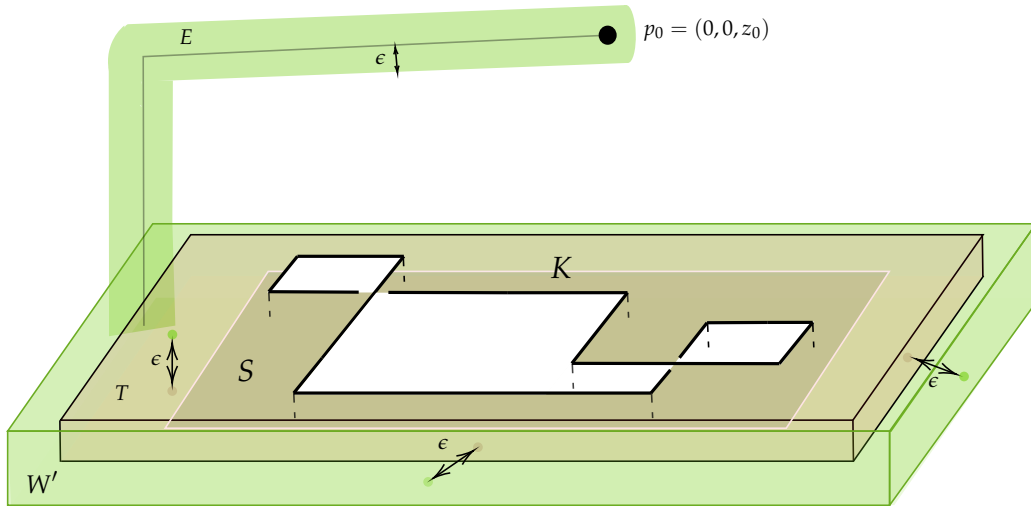


Figura 3.28.: Espacio W' .

El espacio W'' podría ser el representado con colores rosa y verde en la [Figura 3.29](#), el cual no se ha representado todo con el mismo color ya que iban a resultar indistinguibles los elementos. Por una parte, con color rosa está representado el ϵ -entorno de S (ya que recordemos que $S \subset B_*$), y con color verde representamos el espacio de puntos cuya distancia a $B_* \setminus (K \cup S)$ es menor que ϵ y que se encuentran más cerca de $B_* \setminus K$ que de K .

3. Grupo de un nudo

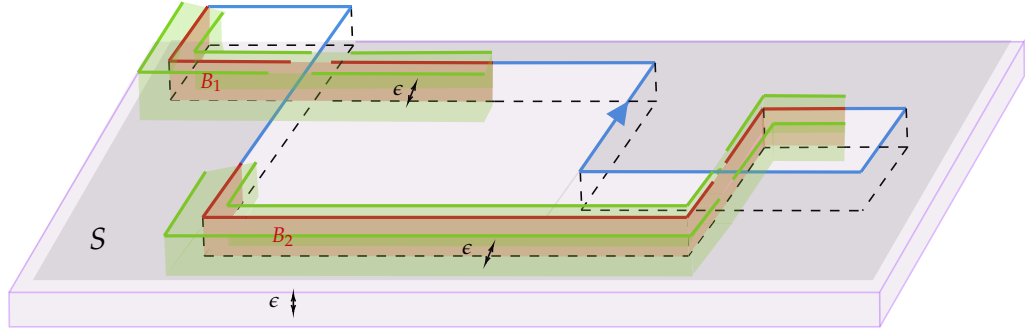


Figura 3.29.: Espacio W'' .

Mediante el ejemplo representado en las [Figura 3.28](#) y [Figura 3.29](#), podemos observar que las propiedades (i) y (ii) que debe cumplir el espacio W se verifican trivialmente, mientras que la propiedad (iii) aunque no sea tan obvia su comprobación, también se cumple.

Antes de enunciar y demostrar uno de los resultados más importantes de este trabajo, vamos a definir un nuevo concepto que se necesitará para dicha demostración. Para enunciar dicho concepto será necesario definir la homotopía de aplicaciones.

Definición 3.18. Dados dos espacios topológicos X e Y , dos aplicaciones continuas $\varphi_1, \varphi_2: X \rightarrow Y$ se dicen homotópicas, que lo denotaremos por $\varphi_1 \simeq \varphi_2$, si existe una aplicación continua $H: X \times [0, 1] \rightarrow Y$ cumpliendo

$$H(x, 0) = \varphi_1(x) \quad \forall x \in X \quad \text{y} \quad H(x, 1) = \varphi_2(x) \quad \forall x \in X.$$

Definición 3.19. Sean X e Y dos espacios topológicos. Una aplicación continua $f: X \rightarrow Y$ diremos que es una **equivalencia homotópica** si existe otra aplicación continua $g: Y \rightarrow X$ tal que

$$g \circ f \simeq \text{Id}_X \quad \text{y} \quad f \circ g \simeq \text{Id}_Y.$$

En este caso diremos que f y g son inversas homotópicas.

Dos espacios X e Y se dicen del mismo tipo de homotopía si existe una equivalencia homotópica entre ellos.

Obviamente, todo homeomorfismo es una equivalencia homotópica pero el recíproco no es cierto. Vamos a verlo con un ejemplo.

Consideremos la circunferencia unidad S^1 y el disco unidad cerrado de \mathbb{R}^2 menos su centro, es decir, $\overline{D} \setminus \{(0, 0)\}$. Es claro que dichos espacios no son homeomorfos pero vamos a ver que existe una equivalencia homotópica entre ellos. Consideramos la aplicación inclusión $i: S^1 \rightarrow \overline{D} \setminus \{(0, 0)\}$ y la aplicación $r: \overline{D} \setminus \{(0, 0)\} \rightarrow S^1$ dada por $r(x, y) = (\frac{x}{\sqrt{x^2+y^2}}, \frac{y}{\sqrt{x^2+y^2}})$, es decir $r(p) = (\frac{p}{\|p\|})$, $p \in \overline{D} \setminus \{(0, 0)\}$.

3.2. Presentación superior e inferior del grupo de un nudo, continuación

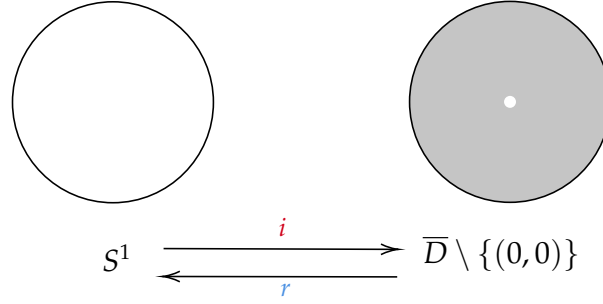


Figura 3.30.: Aplicaciones r e i .

Es claro que $r \circ i = \text{Id}_{S^1}$, por lo tanto $r \circ i \simeq \text{Id}_{S^1}$. Ahora queremos ver que $i \circ r \simeq \text{Id}_{\overline{D} \setminus \{(0,0)\}}$, por lo que para comprobarlo definiremos una deformación $H: \overline{D} \setminus \{(0,0)\} \times [0,1] \rightarrow \overline{D} \setminus \{(0,0)\}$ dada por

$$H(p,t) = (1-t)p + t \frac{p}{\|p\|} \quad \text{para todo } (p,t) \in \overline{D} \setminus \{(0,0)\} \times [0,1].$$

Puesto que la deformación H verifica que

$$H(p,0) = \text{Id}_{\overline{D} \setminus \{(0,0)\}}(p) \quad \forall p \in \overline{D} \setminus \{(0,0)\} \quad \text{y} \quad H(p,1) = i \circ r(p) \quad \forall p \in \overline{D} \setminus \{(0,0)\},$$

tenemos que $i \circ r \simeq \text{Id}_{\overline{D} \setminus \{(0,0)\}}$ como queríamos demostrar, por lo que las aplicaciones r e i definidas anteriormente son equivalencias homotópicas.

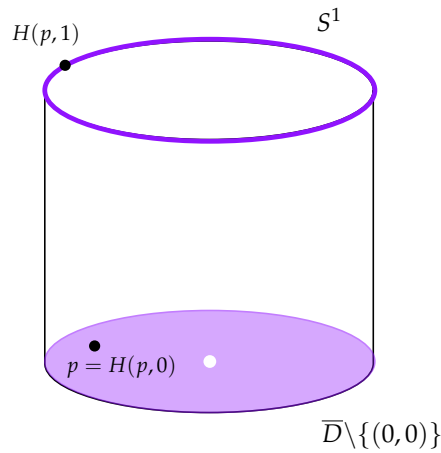


Figura 3.31.: Deformación H asociada al retracts de deformación S^1 de $\overline{D} \setminus \{(0,0)\}$.

Mediante el siguiente teorema veremos que la equivalencia homotópica es suficiente para asegurar el isomorfismo entre sus grupos fundamentales.

Teorema 3.4. *Dados X e Y espacios topológicos y $f: X \rightarrow Y$ una equivalencia homotópica con inversa homotópica $g: Y \rightarrow X$, fijamos $x_0 \in X$. Entonces*

$$f_*: \Pi(X, x_0) \rightarrow \Pi(Y, f(x_0)) \text{ es un isomorfismo de grupos.}$$

3. Grupo de un nudo

Y por fin llegamos al resultado que queríamos demostrar, es decir, vamos a demostrar (3).

Teorema 3.5. *El grupo de un nudo K tiene presentación superior*

$$\langle x_1, \dots, x_n : r_1, \dots, \hat{r}_k, \dots, r_n \rangle_\phi,$$

donde \hat{r}_k indica el borrado del k -ésimo relator r_k y ϕ es el homomorfismo (3.11). Por lo tanto,

$$\Pi(\mathbb{R}^3 \setminus K, p_0) = |x_1, \dots, x_n : r_1, \dots, \hat{r}_k, \dots, r_n|_\phi,$$

Demostración. La idea de la demostración consistirá en aplicar el Teorema de Seifert-Van Kampen considerando los espacios (utilizando la nomenclatura especificada en el **Teorema 3.2**) $U = (\mathbb{R}^3 \setminus B_*)$, $V = W$ y por tanto $U \cap V = ((\mathbb{R}^3 \setminus B_*) \cap W)$, siendo los tres espacios abiertos arcoconexos y $U \cap V$ no vacío.

Primeramente, observamos que

$$B_* \setminus W = (K \cup (B_* \setminus K)) \setminus W = (K \setminus W) \cup ((B_* \setminus K) \setminus W) = K, \quad (3.12)$$

por ser K y W disjuntos y por estar $B_* \setminus K$ contenido en W ($(B_* \setminus K) \subset W$).

Hemos escogido los espacios U y V descritos anteriormente, ya que su unión resultará ser

$$U \cup V = (\mathbb{R}^3 \setminus B_*) \cup W = \mathbb{R}^3 \setminus (B_* \setminus W) = \mathbb{R}^3 \setminus K,$$

que es justo el espacio de cuyo grupo fundamental queremos hallar una presentación, donde en la última igualdad hemos utilizado (3.12).

También tenemos que la intersección de U y V es

$$U \cap V = (\mathbb{R}^3 \setminus B_*) \cap W = W \setminus B_*,$$

que por (III) sabemos que $T \setminus B_*$ es un retracto fuerte de deformación suyo y por tanto, pertenecen a la misma clase de homotopía. El espacio $T \setminus B_*$ es un prisma rectangular con n cortes de cuchillo en la parte superior, siendo n el número de arcos inferiores del nudo K y teniendo dichos cortes la misma forma que las proyecciones de los arcos inferiores $\mathcal{P}B_1, \dots, \mathcal{P}B_n$.

Dicho espacio $T \setminus B_*$ es por tanto del mismo tipo de homotopía que una esfera con n agujeros, por lo que su grupo fundamental es un grupo libre de rango $n - 1$. Vamos a verlo paso a paso en la **Figura 3.32**, donde hemos utilizado el símbolo \sim para denotar que dos espacios son del mismo tipo de homotopía.

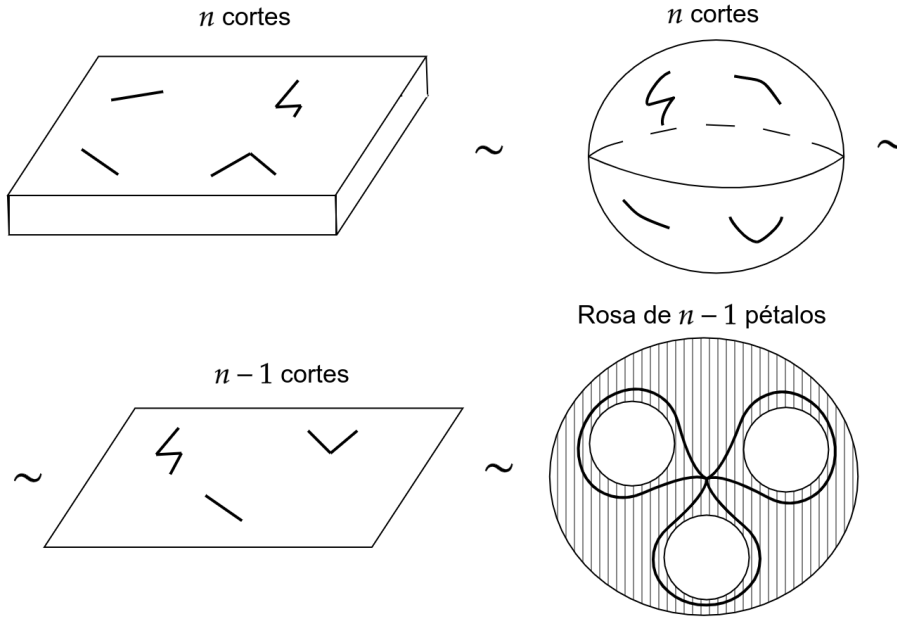


Figura 3.32.: Espacios del mismo tipo de homotopía.

Vamos a demostrar esto último matemáticamente. Sea h la proyección ortogonal del plano $\mathbb{R}^2 \times \{0\}$ sobre el plano que contiene la parte superior de T . Es consecuencia de (3.7) que

$$(\{hq_0\} \cup \bigcup_{i=1}^n hV_i) \subset T.$$

Dado que los cierres $\bar{V}_1, \dots, \bar{V}_n$ de las regiones V_1, \dots, V_n son disjuntos dos a dos, debemos seleccionar un conjunto de caminos poligonales e_1, \dots, e_n en $T \setminus \bigcup_{i=1}^n hV_i$ con un origen común hq_0 y sujetos a las siguientes restricciones:

- El extremo de cada camino e_i es $h(v_i(0))$, siendo v_i el lazo que rodea a V_i .
- Las imágenes de e_1, \dots, e_n son, exceptuando el punto hq_0 , dos a dos disjuntas y están contenidas, exceptuando el origen y el extremo, en $T \setminus \bigcup_{i=1}^n h\bar{V}_i$.

Ahora, utilizando dichos caminos poligonales e_i , definimos

$$w_i = e_i \cdot hv_i \cdot e_i^{-1}, \quad i = 1, \dots, n.$$

En la Figura 3.33 se puede ver el prisma rectangular T , el camino e_1 y el lazo hv_1 a través de los que se infiere el lazo w_1 . También podemos observar el arco hB_1 representado mediante un arco discontinuo y el espacio hV_1 representado con color rojo, que contiene al arco hB_1 .

3. Grupo de un nudo

Para $i = 2, 3, 4$, los arcos, lazos, etc., se representarían de manera equivalente a como hemos hecho para $i = 1$.

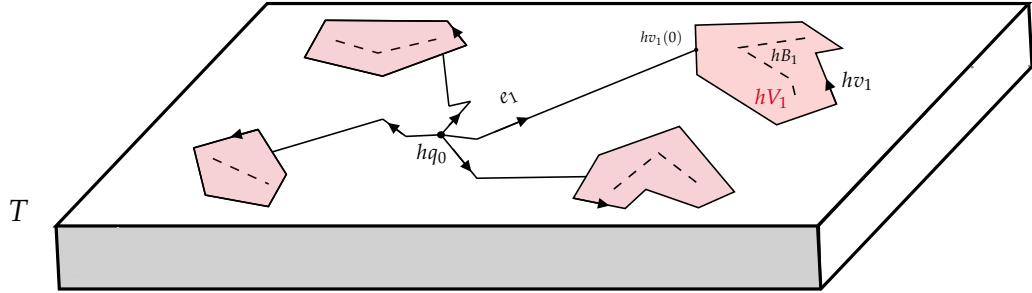


Figura 3.33.: Lazo w_1

Claramente, la unión de las imágenes de n lazos w_1, \dots, w_n es del mismo tipo de homotopía que una rosa de n pétalos, como se puede observar en la Figura 3.34.

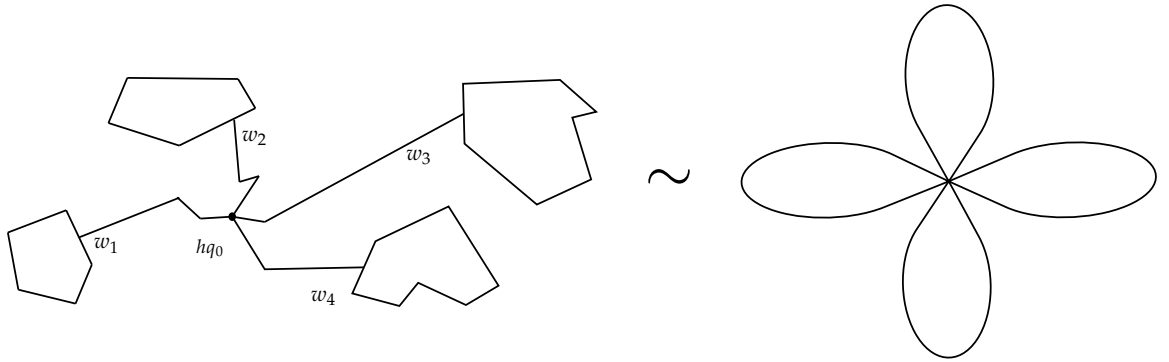


Figura 3.34.: Homotopía de n lazos y rosa de n pétalos.

Además, la unión de las imágenes de cualquiera $n - 1$ lazos de w_1, \dots, w_n es un retracto de deformación de $T \setminus B_*$. Vamos a demostrar esto último.

En primer lugar, ampliamos todos los cortes de cuchillo en $T \setminus B_*$ hasta los caminos cerrados hv_i , $i = 1, \dots, n$. Después, elegimos uno de los agujeros cuyo borde es uno de los caminos hv_i , y estirando ese agujero al máximo, obtenemos los $n - 1$ lazos restantes w_i en un plano, donde $i = 1, \dots, \hat{k}, \dots, n$. Independientemente del agujero con borde hv_i que hayamos decidido estirar, concluimos que $\Pi(T \setminus B_*, hq_0)$ es un grupo libre y las clases de equivalencia de cualquiera $n - 1$ lazos, $w_1, \dots, \hat{w}_k, \dots, w_n$, constituye una base libre de dicho grupo, donde \hat{w}_k indica que no se ha incluido el k -ésimo lazo w_k . Dado que $T \setminus B_*$ es un retracto de deformación de $W \setminus B_*$, lo dicho anteriormente también es cierto para el grupo $\Pi(W \setminus B_*, hq_0)$.

Finalmente, si consideramos un camino a en $W \setminus B_*$ con origen p_0 y extremo el punto hq_0 , que existe ya que $W \setminus B_*$ es arcoconexo y $p_0 \in (W \setminus B_*)$, el conjunto de clases de equivalencia de cualquiera $n - 1$ de los lazos $a \cdot w_i \cdot a^{-1}$, $i = 1, \dots, n$, será una base libre de $\Pi(W \setminus B_*, p_0)$.

Por ejemplo, un posible camino a para el nudo K y el punto q_0 asociado representados en

la Figura 3.17 podría ser el señalado con una línea gruesa negra en la Figura 3.35, en la cual se ha representado el espacio W con color verde.

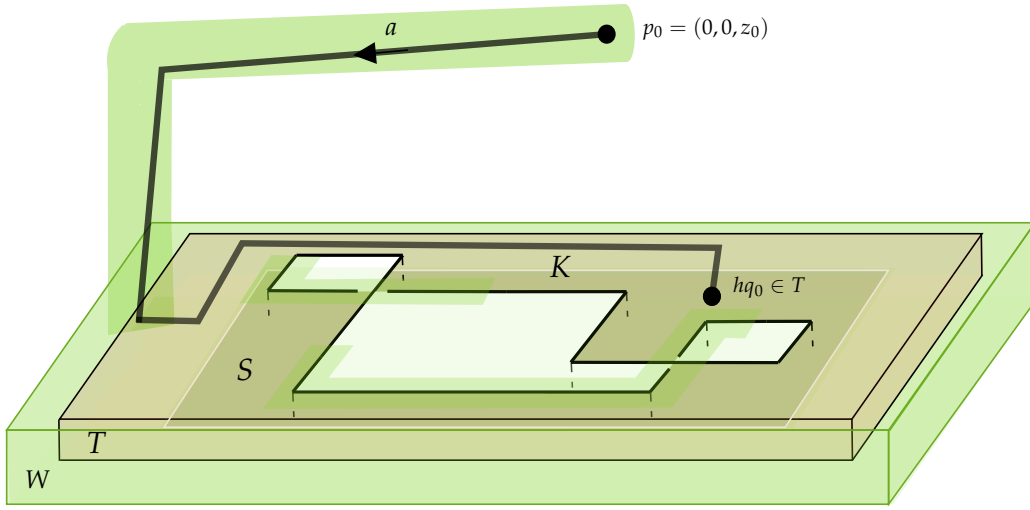


Figura 3.35.: Camino a en $W \setminus B_*$.

Dado que por (i) W es simplemente conexo, un corolario del Teorema de Seifert-Van Kampen nos asegurará que el homomorfismo $\phi : \Pi(\mathbb{R}^3 \setminus B_*, p_0) \rightarrow \Pi(\mathbb{R}^3 \setminus K, p_0)$ es sobreyectivo y su núcleo es consecuencia de cualquiera $n - 1$ clases de equivalencia en $\mathbb{R}^3 \setminus B_*$ de los lazos $a \cdot w_i \cdot a^{-1}, i = 1, \dots, n$. Por tanto, obtenemos que

$$\Pi(\mathbb{R}^3 \setminus K, p_0) = |x_1, \dots, x_n : [a \cdot w_i \cdot a^{-1}], \quad i = 1, \dots, \hat{k}, \dots, n|_{\phi},$$

donde \hat{k} indica la no inclusión del número natural $k \in \{1, \dots, n\}$.

Definición 3.20. En teoría de grupos, se denomina **conjugación** a un tipo de acción de un grupo sobre sí mismo. Dado un grupo $(G, *)$ y $g \in G$ uno de sus elementos, se denomina conjugado de $a \in G$ por g al elemento $b = g^{-1} * a * g$. Entonces se dice que los elementos a y b son conjugados.

Dado un grupo G , definiremos la siguiente relación:

$$\text{Para todo } a, b \in G: \quad a \sim b \Leftrightarrow \exists g \in G \text{ tal que } b = g^{-1} * a * g.$$

Esta relación definida así es una relación de equivalencia. Por ello, llamaremos a la clase de un elemento $a \in G$, clase de conjugación de a , siendo esta

$$[a] = \{x^{-1} * a * x : x \in G\}.$$

La demostración se completa con la observación de que, para todo $i = 1, \dots, n$, el elemento $[a \cdot w_i \cdot a^{-1}]$ es conjugado de $v_i^\#$, y por tanto, de r_i (recordemos que $r_i = (c_i \cdot v_i \cdot c_i^{-1})^\#$, $i = 1, \dots, n$). Para probar esto, consideramos un camino b_i (análogo al camino $\bar{v}_i(0)$) que recorre el segmento con origen p_0 y extremo el punto encima de $v_i(0)$ a la altura de p_0 , y después el segmento que une dicho punto con el punto $h v_i(0)$. Una equivalencia válida en $\mathbb{R}^3 \setminus B_*$

sería

En la **Figura 3.36** se muestran los elementos involucrados en la equivalencia anterior y se ve geoméricamente dicha equivalencia.

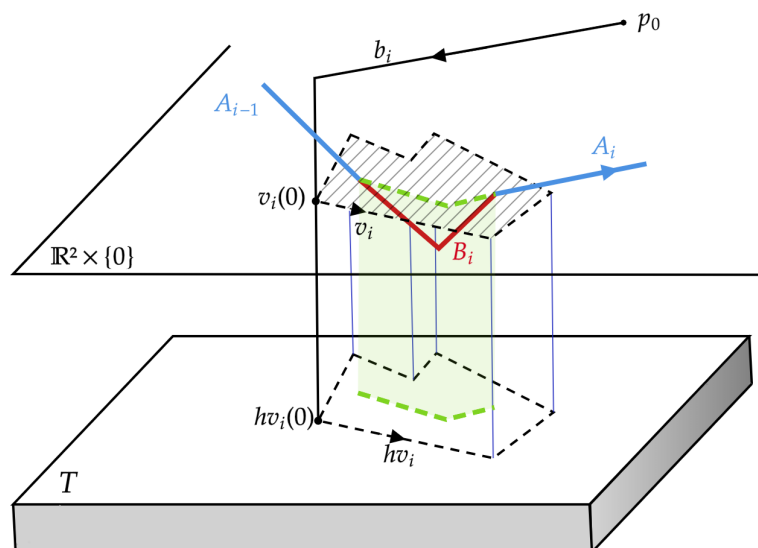


Figura 3.36.: Equivalencia entre los caminos $*v_i$ y $b_i \cdot hv_i \cdot b_i^{-1}$.

Consecuentemente,

$$a \cdot w_i \cdot a^{-1} = (a \cdot e_i) \cdot hv_i \cdot (a \cdot e_i)^{-1} \simeq (a \cdot e_i \cdot b_i^{-1}) \cdot b_i \cdot hv_i \cdot b_i^{-1} \cdot (a \cdot e_i \cdot b_i^{-1})^{-1} \simeq f_i \cdot *v_i \cdot f_i^{-1},$$

donde $f_i = (a \cdot e_i \cdot b_i^{-1})$. Por tanto,

$$[a \cdot w_i \cdot a^{-1}] = [f_i]v_i^\# [f_i]^{-1}, \quad i = 1, \dots, n.$$

Puesto que $v_i^\#$ es conjugado de r_i , el núcleo de ϕ es consecuencia de $r_1, \dots, \hat{r}_k, \dots, r_n$, y queda demostrado por tanto el **Teorema 3.5**. \square

Durante lo que llevamos visto en esta capítulo hemos tratado únicamente con nudos poligonales. Esto se ha debido fundamentalmente a que, aunque sea poco natural dibujar o encontrarnos con nudos con arcos segmentados, la existencia de nudos no admisibles hace que se utilice esta conveniencia.

3.3. La presentación Wirtinger

En esta sección se utilizan también conceptos basados en el artículo *Tietze Equivalences as Weak Equivalences* [HM21].

Antes de estudiar el concepto de presentación Wirtinger, vamos a estudiar ciertas operaciones existentes entre presentaciones equivalentes, entre las cuales vamos a estudiar las

llamadas operaciones I y I' de Tietze.

Sea $\langle x: r \rangle$ una presentación y sea l una relación que se deriva de las relaciones de r . Consideramos la presentación $\langle y: s \rangle$, dada por $y = x$ y $s = r \cup l$. En este caso, las relaciones derivadas de r son las mismas que las relaciones derivadas de s , por lo tanto, las presentaciones $\langle x: r \rangle$, $\langle y: s \rangle$ y el automorfismo identidad $1: F[x] \rightarrow F[y]$ definen la aplicación entre presentaciones $I: \langle x: r \rangle \rightarrow \langle y: s \rangle$. De manera similar, las presentaciones $\langle x: r \rangle$, $\langle y: s \rangle$ y el automorfismo identidad 1 definen la aplicación entre presentaciones $I': \langle y: s \rangle \rightarrow \langle x: r \rangle$. Es trivial que las operaciones I y I' van a aplicar una presentación en otra equivalente.

Un ejemplo para el que se podrían definir estas operaciones I y I' de Tietze serían las presentaciones $\langle x: x^3 = 1 \rangle$ y $\langle x: x^3 = 1, x^6 = 1 \rangle$ ya que la relación $x^6 = 1$ de la segunda presentación es consecuencia de la relación $x^3 = 1$, puesto que multiplicando la relación $x^3 = 1$ a ambos lados por x^3 obtenemos que $x^6 = x^3 = 1$.

Hay otras dos operaciones de Tietze más, que se denotan por II y II' , pero no hará falta definir las porque para la aplicación que haremos de estas operaciones en la teoría de nudos solo requeriremos de las operaciones I y I' .

Definición 3.21. Dadas dos presentaciones $\langle x: r \rangle$ y $\langle y: s \rangle$, una transformación de Tietze consiste en una secuencia finita de presentaciones

$$\langle x: r \rangle = \langle x_0: r_0 \rangle, \langle x_1: r_1 \rangle, \dots, \langle x_n: r_n \rangle = \langle y: s \rangle$$

tal que para todo $i = 0, \dots, n-1$ existe una operación de Tietze que aplica $\langle x_i: r_i \rangle$ en $\langle x_{i+1}: r_{i+1} \rangle$.

Las operaciones de Tietze son importantes debido a que nos dan la equivalencia entre los grupos definidos por las presentaciones que intervienen en dichas operaciones, como nos hace ver el teorema siguiente.

Teorema 3.6. Dos presentaciones definen grupos isomorfos si y solo si una puede ser transformada en la otra mediante una transformación de Tietze.

Por tanto, como la presentación

$$\langle x_1, \dots, x_n : v_1^\#, \dots, v_n^\# \rangle_\phi \quad (3.13)$$

de $\Pi(\mathbb{R}^3 \setminus K, p_0)$ se obtiene de la presentación superior (3.4) mediante n aplicaciones de operaciones I y I' de Tietze, los grupos que definen son isomorfos.

La principal ventaja de (3.13) sobre (3.4) es que requiere menos trabajo calcularla, ya que, no tenemos que calcular los elementos $c_1^\#, \dots, c_n^\#$. Por esta razón, de ahora en adelante utilizaremos la presentación (3.13). Dada la equivalencia entre las dos presentaciones mencionadas anteriormente, se deduce que cualquiera de los relatores $v_1^\#, \dots, v_n^\#$ de (3.13) es consecuencia de los $n-1$ restantes y por tanto puede ser omitido.

Una vez visto que las presentaciones (3.13) y (3.4) definen grupos isomorfos, podemos ya definir el concepto de presentación Wirtinger. Una presentación (3.13) del grupo de un nudo dado es llamada **presentación Wirtinger** si:

- Cada arco inferior B_i contiene exactamente un punto inferior.

3. Grupo de un nudo

- Cada camino v_i intersecta a los arcos superiores proyectados $\mathcal{P}A_1, \dots, \mathcal{P}A_n$ en tan solo 4 puntos.

Las dos condiciones anteriores pueden ser impuestas, salvo que el nudo no tenga puntos inferiores y por tanto, no se podría verificar la primera condición. Una característica llamativa y atractiva de las presentaciones Wirtinger es que los relatores son bastante simples, lo que hizo que fuera una de las primeras presentaciones del grupo de un nudo que fueron estudiadas. Escritos dichos relatores como relaciones, cada una de ellas es de la forma $x_k = x_i^\epsilon x_l x_i^{-\epsilon}$ con $\epsilon = \pm 1$, ya que el camino v_j , para $j = 1, \dots, n$ solo intersecta en 4 puntos con las proyecciones de los arcos superiores $\mathcal{P}A_1, \dots, \mathcal{P}A_n$. En consecuencia, dicho lazo v_j que rodea a la proyección del arco inferior $\mathcal{P}B_i$ de un nudo poligonal arbitrario K se asemejará al mostrado en la **Figura 3.37**.

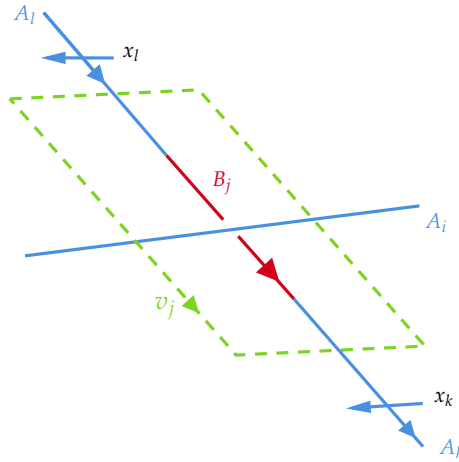


Figura 3.37.: Camino v_j .

Para dicho camino v_j tenemos que

$$v_j^\# = x_k^{-1} x_i^\epsilon x_l x_i^{-\epsilon}, \text{ donde } \epsilon = \pm 1.$$

Por lo tanto, se cumple lo dicho anteriormente, que

$$x_k = x_i^\epsilon x_l x_i^{-\epsilon}, \text{ donde } \epsilon = \pm 1.$$

Veremos que, pese a esta ventaja, la presentación Wirtinger no es la más eficiente en cuanto al número de generadores y relatores que obtenemos, a menos que el nudo sea un nudo alterno, es decir, un nudo que a medida que lo recorramos los puntos superiores e inferiores se alternen.

3.4. Ejemplos de presentaciones

En esta sección vamos a calcular presentaciones superiores de algunos de los nudos más conocidos. Dado que habíamos visto en la **Sección 3.3** que era más simple calcular las presentaciones dadas por (3.13) que las dadas por (3.4), para el cálculo de la presentación del grupo de los nudos de esta sección son las primeras las que vamos a calcular. Además,

debemos tomar ventaja del hecho de que un relator arbitrario puede ser eliminado (ya que se obtiene de los $n - 1$ restantes). Sin embargo, aunque eliminemos uno de los relatores, la presentación resultante a menudo es todavía innecesariamente compleja y puede ser enormemente simplificada usando las operaciones de Tietze vistas en la sección anterior. Vamos a calcular y simplificar las presentaciones de los grupos de algunos nudos para familiarizarnos con su cálculo y sus reducciones.

3.4.1. Nudo trivial

En la Figura 3.38 hemos representado el nudo trivial dividido en arcos superiores e inferiores. El único arco superior, A_1 , está representado con color azul, y utilizando una línea discontinua de color verde hemos representado el camino v_1 .

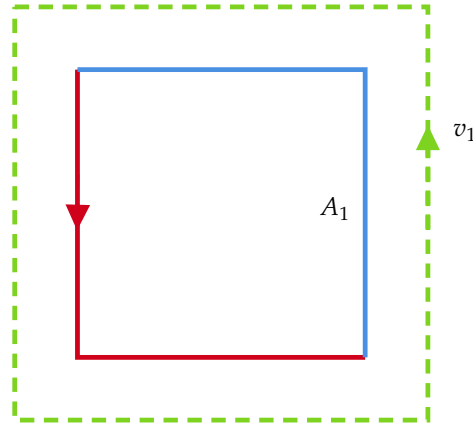


Figura 3.38.: Nudo trivial.

Por tanto, como el camino v_1 no se intersecta con la proyección del arco superior $\mathcal{P}A_1$ en ningún punto, la presentación del grupo del nudo K será

$$\langle x_1 : \cdot \rangle,$$

por lo que

$$\Pi(\mathbb{R}^3 \setminus K) = |x_1 : \cdot|.$$

Por tanto, el grupo del nudo trivial es un grupo cíclico infinito.

Para los siguientes nudos se representará con color azul los arcos superiores A_i , con color rojo los arcos inferiores B_i , y con líneas verdes discontinuas los caminos v_i , al igual que los hemos representado en este último ejemplo. Además, para no sobrecargar las figuras, los puntos $v_i(0)$, necesarios para hallar la presentación del grupo del nudo, serán los puntos representados con una flecha al lado del nombre de los lazos, señalados con el nombre " v_i'' " para $i = 1, \dots, n$.

3. Grupo de un nudo

3.4.2. Nudo trébol

Queremos hallar una presentación del grupo del nudo trébol. Para ello, elegimos generadores x_1, x_2, x_3 tales que

$$x_1 = a_1^\#, x_2 = a_2^\#, x_3 = a_3^\#, \quad (3.14)$$

donde a_1, a_2 y a_3 son los caminos simples en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{PB}$ representados con color negro en la Figura 3.39.

En dicha figura se pueden observar también los arcos superiores e inferiores del nudo trébol y los caminos v_i que rodean a las proyecciones de los arcos inferiores \mathcal{PB}_i en el plano $\mathbb{R}^2 \times \{0\}$ para $i = 1, 2, 3$.

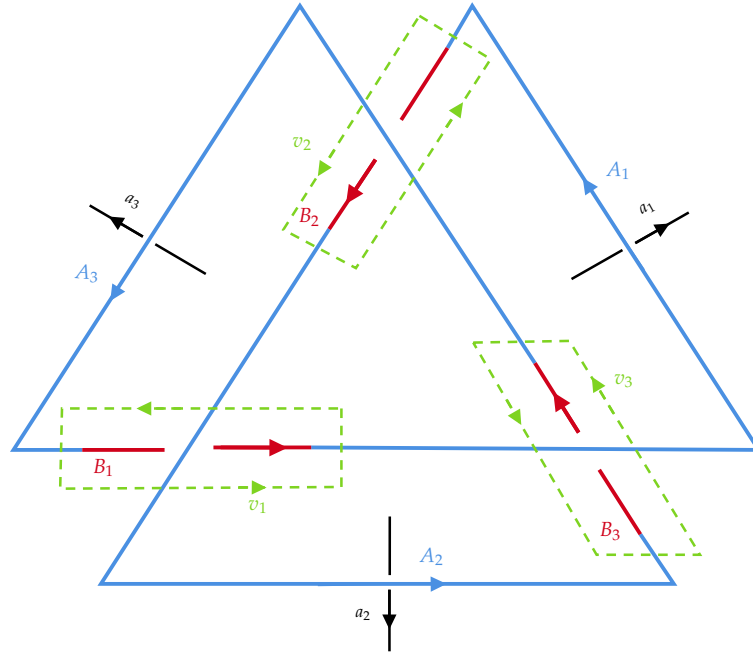


Figura 3.39.: Nudo trébol

En consecuencia, los relatores de la presentación Wirtinger serán

$$v_1^\# = x_1^{-1} x_2 x_3 x_2^{-1},$$

$$v_2^\# = x_2^{-1} x_3 x_1 x_3^{-1},$$

$$v_3^\# = x_3^{-1} x_1 x_2 x_1^{-1}.$$

Consecuentemente, obtenemos que una presentación del grupo del nudo trébol es

$$\langle x_1, x_2, x_3 : x_1^{-1} x_2 x_3 x_2^{-1}, x_2^{-1} x_3 x_1 x_3^{-1} \rangle,$$

donde hemos eliminado el relator $v_3^\#$, ya que, como habíamos demostrado previamente, es consecuencia de $v_1^\#$ y de $v_2^\#$.

Supongamos que no eliminamos el relator $v_3^\#$, es decir, tenemos la presentación con los tres relatores descritos anteriormente. Escribiendo las relaciones en vez de los relatores, obtenemos que el grupo del nudo K es

$$\Pi(\mathbb{R}^3 \setminus K) = |x_1, x_2, x_3 : x_1 = x_2 x_3 x_2^{-1}, x_2 = x_3 x_1 x_3^{-1}, x_3 = x_1 x_2 x_1^{-1}|.$$

Al sustituir $x_3 = x_1 x_2 x_1^{-1}$ (relación obtenida de $v_3^\#$) en las otras dos relaciones obtenemos que

$$\Pi(\mathbb{R}^3 \setminus K) = |x_1, x_2 : x_1 = x_2 x_1 x_2 x_1^{-1} x_2^{-1}, x_2 = x_1 x_2 x_1 x_2^{-1} x_1^{-1}|.$$

Si multiplicamos por la derecha la segunda relación por $x_1 x_2 x_1^{-1} x_2^{-1}$ obtenemos la primera, por lo que hemos comprobado que cualquiera de las relaciones obtenidas de rodear a un arco inferior en una presentación superior se deduce de las otras relaciones. Por lo tanto, la presentación del grupo del nudo sería

$$\langle x_1, x_2 : x_1 = x_2 x_1 x_2 x_1^{-1} x_2^{-1} \rangle,$$

y multiplicando por la derecha por $x_2 x_1$ obtenemos que la presentación del grupo $\Pi(\mathbb{R}^3 \setminus K)$, o la llamada presentación del grupo del nudo trébol, es

$$\langle x_1, x_2 : x_1 x_2 x_1 = x_2 x_1 x_2 \rangle.$$

3.4.3. Nudo de ocho

Queremos hallar una presentación del grupo del nudo de ocho. Como hicimos para hallar una presentación del grupo del nudo trébol, elegimos generadores x_1, x_2, x_3, x_4 de forma que

$$x_1 = a_1^\#, x_2 = a_2^\#, x_3 = a_3^\#, x_4 = a_4^\#,$$

donde a_1, a_2, a_3 y a_4 son los caminos simples en $(\mathbb{R}^2 \times \{0\}) \setminus \mathcal{P}B$ representados con color negro en la [Figura 3.40](#).

En la [Figura 3.40](#) se pueden observar también los arcos superiores e inferiores del nudo de ocho y los caminos v_i que rodean a las proyecciones de los arcos inferiores $\mathcal{P}B_i$ en el plano $\mathbb{R}^2 \times \{0\}$ para $i = 1, 2, 3, 4$.

3. Grupo de un nudo

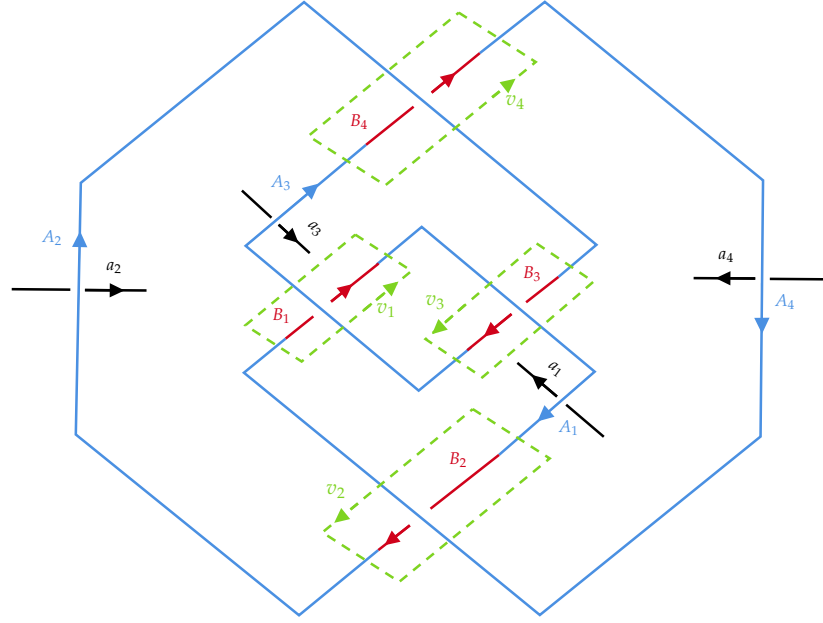


Figura 3.40.: Nudo de ocho.

En consecuencia, los relatores de la presentación Wirtinger serán

$$\begin{aligned} v_1^\# &= x_1^{-1} x_3^{-1} x_4 x_3, \\ v_2^\# &= x_2^{-1} x_4 x_1 x_4^{-1}, \\ v_3^\# &= x_3^{-1} x_1^{-1} x_2 x_1, \\ v_4^\# &= x_4^{-1} x_2 x_3 x_2^{-1}. \end{aligned}$$

Eliminando el relator $v_4^\#$ de la presentación Wirtinger del grupo del nudo de ocho, obtenemos la presentación

$$\langle x_1, x_2, x_3, x_4 : x_1 = x_3^{-1} x_4 x_3, x_2 = x_4 x_1 x_4^{-1}, x_3 = x_1^{-1} x_2 x_1 \rangle.$$

Vamos a tratar de simplificar más aún dicha presentación, para ello, sustituimos $x_3 = x_1^{-1} x_2 x_1$ en las otras dos igualdades, obteniendo

$$\Pi(\mathbb{R}^3 \setminus K) = |x_1, x_2, x_4 : x_2 = x_1^{-1} x_2^{-1} x_1 x_4 x_1^{-1} x_2 x_1, x_4 = x_4 x_1 x_4^{-1}|.$$

De la primera relación deducimos que $x_4 = x_1^{-1} x_2 x_1 x_2^{-1} x_1$. Sustituyendo en la segunda obtenemos

$$\Pi(\mathbb{R}^3 \setminus K) = |x_1, x_2 : x_2 = x_1^{-1} x_2 x_1 x_2^{-1} x_1 x_2 x_1^{-1} x_2^{-1} x_1|.$$

Finalmente, multiplicando la relación por la derecha por $x_1^{-1} x_2 x_1 x_2^{-1}$, obtenemos que una presentación del grupo $\Pi(\mathbb{R}^3 \setminus K)$, es decir, una presentación del grupo del nudo de ocho, es

$$\langle x_1, x_2 : x_2 x_1^{-1} x_2 x_1 x_2^{-1} = x_1^{-1} x_2 x_1 x_2^{-1} x_1 \rangle.$$

3.5. Existencia de tipos de nudos no triviales

Una vez que hemos llegado a este punto, podemos ya probar la existencia de diferentes tipos de nudos. Para ello, vamos a probar que el nudo trébol, nudo para cuyo grupo habíamos hallado una presentación en la [Subsección 3.4.2](#), no es equivalente al nudo trivial. Para esta prueba, además de utilizar como documentación el libro *Introduction to Knot Theory* [CF77], nos hemos basado también en ideas del libro *Knots and Links* [Rol76]

Recordemos que si dos nudos K y K' son del mismo tipo, entonces los espacios complementarios $\mathbb{R}^3 \setminus K$ y $\mathbb{R}^3 \setminus K'$ son homeomorfos, y por tanto, $\Pi(\mathbb{R}^3 \setminus K)$ y $\Pi(\mathbb{R}^3 \setminus K')$ son grupos fundamentales isomorfos. Debido a que el grupo de un nudo es un invariante de nudos, dos nudos del mismo tipo tendrán grupos isomorfos, por lo que probar que los grupos de dos nudos no son isomorfos, será el principal método que utilizaremos para distinguir tipos de nudos. Habíamos visto que el grupo del nudo trébol era $\Pi(\mathbb{R}^3 \setminus K) = \langle x_1, x_2 : x_1 x_2 x_1 = x_2 x_1 x_2 \rangle$, y demostraremos que no es del mismo tipo que el nudo trivial viendo que ese grupo no es cíclico infinito, a diferencia del grupo del nudo trivial, que ya vimos que sí lo era en la [Subsección 3.4.1](#). Para demostrarlo, vamos a considerar el grupo de las permutaciones de orden 3, S_3 , que está generado por las transposiciones (12) y (23). Es fácil de ver que el grupo S_3 no es abeliano, ya que por ejemplo,

$$(13)(23) = (123) \neq (132) = (23)(13).$$

Tenemos que una presentación del nudo de trébol está determinada por un homomorfismo ϕ del grupo libre $F[x]$, donde $x = \{x_1, x_2\}$ es una base libre de dicho grupo siendo x_1 y x_2 los elementos hallados en (3.14), al grupo del nudo trébol, siendo el núcleo de ϕ consecuencia de $x_1 x_2 x_1 (x_2 x_1 x_2)^{-1}$. Por otra parte el homomorfismo θ de $F[x]$ a S_3 dado por

$$\theta(x_1) = (13) \quad \theta(x_2) = (23),$$

induce un homomorfismo del grupo del nudo trébol a S_3 siempre que $\theta(x_1 x_2 x_1) = \theta(x_2 x_1 x_2)$. Dado que

$$\theta(x_1 x_2 x_1) = \theta(x_1) \theta(x_2) \theta(x_1) = (13)(23)(13) = (12),$$

$$\theta(x_2 x_1 x_2) = \theta(x_2) \theta(x_1) \theta(x_2) = (23)(13)(23) = (12),$$

tenemos que se cumple que $\theta(x_1 x_2 x_1) = \theta(x_2 x_1 x_2)$, y puesto que además tenemos que $\theta(x_1^2) = 1$, $\theta(x_1 x_2) = (13)(23) = (123)$ y $\theta(x_2 x_1) = (23)(13) = (132)$, el homomorfismo θ inducirá un epimorfismo del grupo del nudo trébol al grupo S_3 , que al ser no abeliano como hemos visto previamente, el grupo del nudo trébol también será no abeliano. Se sigue por tanto, que el grupo del nudo trébol es no abeliano, y en consecuencia, no es un grupo cíclico infinito. Por ello, el nudo trébol no es equivalente al nudo trivial.

De la misma forma, para probar que el nudo trébol es distinto del nudo de ocho, es suficiente con ver que sus grupos no son isomorfos. Sin embargo, no hay un procedimiento general para determinar si dos presentaciones determinan grupos isomorfos o no. Esta es una de las principales razones por las que este invariante de nudos, el grupo de un nudo, no es siempre el elegido para comprobar la no equivalencia de dos nudos. Es más, se podría haber demostrado que el nudo trivial y el nudo trébol no son equivalentes utilizando como invariante la tricolorabilidad del nudo, mencionada en la [Sección 2.5](#), ya que el nudo trivial

3. Grupo de un nudo

no es tricoloreable mientras que el nudo trébol sí, y es un proceso mucho menos tedioso que el que hemos seguido para demostrar la no equivalencia de estos utilizando el grupo del nudo como invariante.

Sin embargo, aunque existen invariantes más sencillos de calcular y más fáciles de manejar, aunque a veces alguno de ellos nos sea de utilidad como nos hubiera sido la tricolorabilidad en el ejemplo anterior, se perderá inevitablemente alguna información, es decir, habrá más probabilidad de que dados dos nudos no equivalentes, al calcular dicho invariante, estos invariantes si sean equivalentes. Por ello, aunque a veces sea complicado comprobar que los grupos de dos nudos no son isomorfos, puede que merezca la pena dicha prueba ya que puede darse el caso de que con invariantes más sencillos que el grupo del nudo no logremos demostrar la distinguibilidad de esos dos nudos.

Parte II.

Parte Informática

4. Codificación de nudos

En este capítulo estudiaremos diferentes formas de codificar la información que nos da un objeto asociado a una proyección de un nudo, llamado diagrama de un nudo, que definiremos en la [Sección 4.1](#). A través del diagrama de un nudo, utilizando cualquiera de las formas de codificación que estudiaremos en este capítulo, obtendremos una serie de datos, que dependiendo de la forma de codificación utilizada, permitirán identificar a un nudo en mayor o menor medida. Además de obtener de un determinado diagrama de un nudo los datos obtenidos de codificarlo, realizaremos también el proceso inverso, para cada forma de codificar los diagramas que estudiaremos, dado un conjunto de datos, construiremos el diagrama asociado a dicho conjunto de datos.

Lo expuesto en este capítulo se basa principalmente en el libro *The Knot Book* [[Ada94](#)] y en los artículos *The programmatic manipulation of planar diagram codes to find an upper bound on the bridge index of prime knots* [[Joh17](#)] y *What are Gauss and Dowker-Thistlethwaite codes?* [[Mil19](#)].

En esta sección vamos a estudiar diferentes notaciones para describir el diagrama de un nudo arbitrario K que, como veremos en la siguiente sección, es un objeto matemático asociado a una proyección ortogonal sobre un plano de dicho nudo.

4.1. Diagrama de un nudo

Una forma de estudiar un nudo arbitrario es realizar en un plano un diagrama de este. En la [Parte I](#) del trabajo consideramos sin pérdida de generalidad que todo nudo admisible estaba en posición regular respecto a la proyección $\mathcal{P}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ tal que $\mathcal{P}(x, y, z) = (x, y, 0)$, por lo que para la construcción de un diagrama consideraremos dicha proyección \mathcal{P} .

Por tanto, dado un nudo admisible, para construir el diagrama de dicho nudo comenzaremos por proyectar dicho nudo mediante la aplicación \mathcal{P} en el plano $\mathbb{R}^2 \times \{0\}$. A los puntos dobles de la proyección de dicho nudo los llamaremos cruces, y asociado a cada cruce habrá dos hebras, la superior, cuya preimagen contiene el punto superior asociado a ese punto doble, y la inferior, cuya preimagen contiene el punto inferior asociado a ese punto doble. La convención habitual es dividir la hebra inferior en dos, dejando un hueco en los puntos cercanos a dicho cruce, dando una especie de efecto de profundidad, para simbolizar que a través de dicha hebra se pasará el cruce por debajo. La proyección del nudo, junto con esa información que nos otorga la división de las hebras inferiores de los cruces en dos, es lo que conocemos como diagrama de un nudo.

Por ejemplo, en la [Figura 4.1](#) podemos observar que en la [Figura 4.1a](#) está representada la proyección del nudo de ocho, donde no podemos distinguir cuál de las dos hebras asociadas a cada cruce es la superior y cuál la inferior. En cambio, en la [Figura 4.1b](#) está representado el diagrama del nudo de ocho, donde sí podemos distinguir cuáles son la hebra superior e

4. Codificación de nudos

inferior asociadas a cada cruce, debido a la división de las hebras inferiores, para dar el ya mencionado efecto de profundidad.

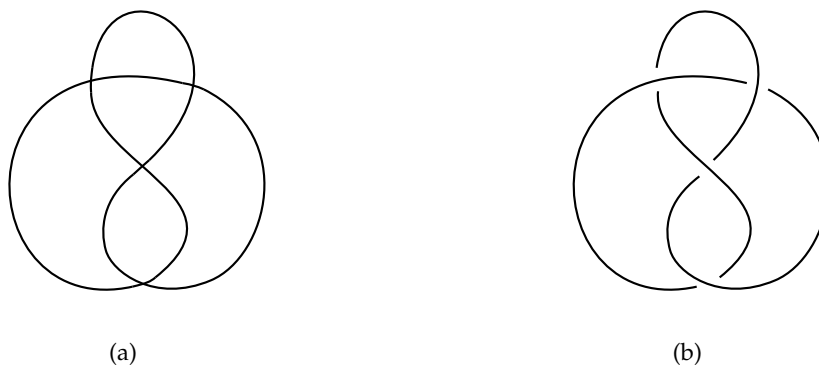


Figura 4.1.: (a) Proyección del nudo de ocho. (b) Diagrama del nudo de ocho.

Debemos tener en cuenta que un diagrama tiene

- cruces, que pueden considerarse como dos hebras que se superponen en un solo punto,
- aristas, que son las líneas que hay entre dos cruces.

Para una mayor interpretabilidad del diagrama, consideraremos que los cruces son vértices del diagrama, por tanto, una arista será la línea que une dos vértices del diagrama.

Los diagramas reducen a una cantidad finita de datos la enorme complejidad que suponen todas las formas en las que una circunferencia puede estar embebida en un espacio tridimensional. En esta sección obtendremos diferentes datos de los diagramas de nudos y codificaremos los datos de dichos diagramas, siendo el objetivo principal que el ordenador sea capaz de interpretarlos y así poder estudiar computacionalmente los nudos.

Supongamos el caso de que una persona encuentra un nudo interesante y se lo quiere contar a un amigo suyo, pero sólo dispone de un móvil con llamadas ¿Qué le debe decir para asegurarse de que su amigo lo ha entendido perfectamente? Este ejemplo muestra la importancia de encontrar dichos datos que identifiquen un determinado nudo, y que al introducirlos a un programa informático, este sea capaz de identificar el nudo del que queremos trasladar información.

Por ello, en esta sección estudiaremos diferentes notaciones que nos permitan, a través del conjunto de datos asociados al diagrama de un nudo, obtenidos después de haber utilizado una determinada notación, poder ser capaces de reconstruir dicho diagrama, y en consecuencia el nudo, y viceversa, dado el diagrama de un nudo, obtener los datos asociados a dicho diagrama utilizando la notación susodicha.

4.2. Formas de codificar el diagrama de un nudo

En esta sección, como el título de la misma indica, vamos a estudiar diferentes formas de codificar el diagrama de un nudo. Concretamente, vamos a ver cuatro formas de codificarlo, para las cuales veremos que las codificaciones que utilizan más información del diagrama para codificarlo, permitirán identificar a un nudo o a su imagen especular. Sin embargo, las que utilizan menos información del diagrama, veremos que no nos asegurarán que dos nudos tales que uno no sea la imagen especular del otro tengan diagramas asociados con distinta codificación.

4.2.1. Código de un diagrama etiquetado

Nos gustaría tener una forma de conocer el orden cíclico de las aristas incidentes alrededor de cada vértice y como están unidas. Si se conoce esto, se puede reconstruir el nudo. También nos gustaría poder tratar con nudos orientados, lo que implica que cada arista del grafo tenga una orientación. Para ello, definiremos el concepto de diagrama etiquetado de un nudo, que tendrá asociado un determinado diagrama, el cual hay que puntualizar que no puede tener ningún bucle propio, es decir no puede contener aristas que conecten un vértice con él mismo.

Definición 4.1. Un **diagrama etiquetado** es un diagrama de un nudo para el cual, recorriendo el nudo en la orientación de éste, las aristas de dicho diagrama están etiquetadas con números naturales consecutivos. Para crear un diagrama etiquetado, a partir de un diagrama seguiremos los siguientes pasos:

- I. Seleccionamos un punto de una arista del diagrama, que será el punto de partida.
- II. Etiquetamos la arista que contiene a dicho punto con el número 1.
- III. Seleccionamos una dirección para recorrer el diagrama, a menos de que dicha dirección fuera dada con anterioridad. Recordemos que a la dirección en la que se recorre un nudo se le denomina orientación del nudo.
- IV. Desde el punto de partida, recorreremos el diagrama del nudo siguiendo la orientación de éste y etiquetamos cada arista que vayamos recorriendo con el número natural siguiente al de la arista previamente etiquetada, hasta llegar a la arista que contiene el punto de partida.

Un mismo diagrama puede tener diferente etiquetado de las aristas de este en función del punto de partida y de la orientación del nudo elegidos. Por ejemplo, en la [Figura 4.2](#) podemos ver un mismo diagrama del nudo de ocho, con tres etiquetados de las aristas distintos. Con la flecha dibujada en dichos diagramas, además de indicar la orientación del nudo, indicamos que el punto sobre el que se sitúa la flecha es el punto de partida para etiquetar las aristas del nudo. Por lo tanto, podemos observar que aunque los diagramas etiquetados de la [Figura 4.2a](#) y de la [Figura 4.2b](#) tengan el punto de partida en la misma arista, al tener orientaciones distintas el etiquetado de las aristas es distinto. Por otra parte, pese a que los nudos de la [Figura 4.2b](#) y de la [Figura 4.2c](#) tienen la misma orientación, el etiquetado de las aristas de ambos diagramas es distinto debido a que sus puntos de partida se encuentran en aristas distintas.

4. Codificación de nudos

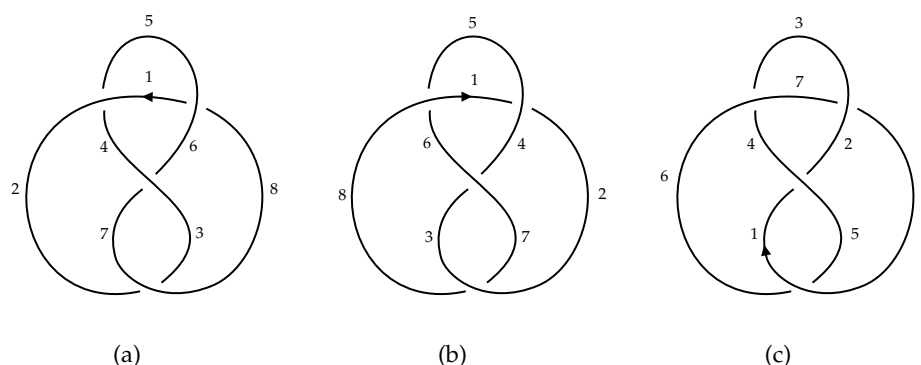


Figura 4.2.: Tres etiquetados para el diagrama del nudo de ocho.

Dado un diagrama etiquetado de un nudo, queremos ser capaces de asociar un determinado código o notación a dicho diagrama, de forma que lo caracterice completamente.

Definición 4.2. El **código de un diagrama etiquetado** de un nudo es una representación numérica del diagrama etiquetado de un nudo, compuesta de 4-tuplas. Cada elemento de una 4-tupla corresponde a una arista del diagrama del nudo, y cada 4-tupla corresponde a un cruce.

Dicho conjunto de 4-tuplas del código de un diagrama etiquetado tendrá que reflejar las aristas que participan en la parte superior de un cruce, y las que lo hacen en la parte inferior de este, por lo que lo generaremos como sigue. Para cada cruce, incluimos en una 4-tupla las aristas involucradas en dicho cruce, empezando por la arista que forma parte de la parte inferior del cruce y que se recorre antes de llegar a dicho cruce, siguiendo la orientación del nudo indicada en el diagrama etiquetado. A partir de esa arista incluimos las tres aristas restantes recorriendo en sentido antihorario dicho cruce. De esta forma, la primera y la tercera arista de una 4-tupla participarán en la parte inferior de un cruce y la segunda y la cuarta participarán en la parte superior de este.

Veamos a continuación un ejemplo de la obtención de un diagrama etiquetado de un nudo a partir del diagrama de este, y del posterior cálculo del código de dicho diagrama etiquetado. Dado el diagrama de un nudo con ningún bucle propio, y dado un punto de una arista de dicho diagrama y una orientación de dicho nudo, simbolizados ambos con una flecha, etiquetamos las aristas del diagrama siguiendo los pasos mencionados anteriormente, hasta llegar al diagrama etiquetado mostrado en la [Figura 4.3](#).

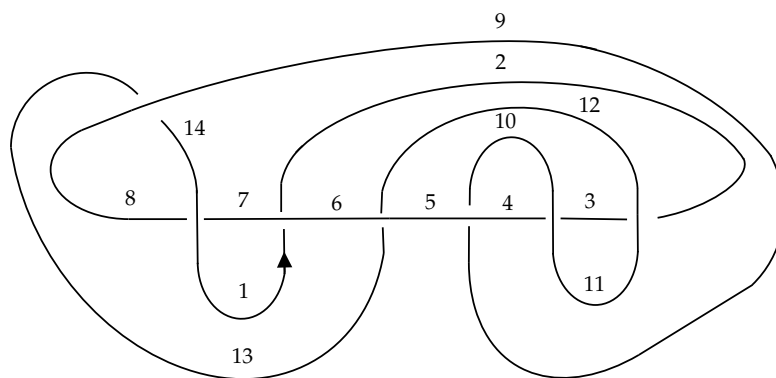


Figura 4.3.: Diagrama etiquetado de un nudo.

Para el diagrama etiquetado del nudo anterior, obtenemos que el código de dicho diagrama etiquetado será el representado en la [Tabla 4.1](#), donde cada columna está asociada a un cruce y contiene las etiquetas de las aristas asociadas a dicho cruce, ordenadas como se ha mencionado anteriormente.

1	2	3	7	9	12	13
6	12	10	14	4	6	8
2	3	4	8	10	13	14
7	11	11	1	5	5	9

Tabla 4.1.: Código del diagrama etiquetado de la [Figura 4.3](#)

Estas 4–tuplas del código de un diagrama etiquetado de un nudo son exactamente los datos que utiliza la biblioteca *SnapPy* de *Python*, cuya documentación está disponible en [\[CDGW\]](#), para identificar un nudo. Para crear un objeto asociado a dicho código, se haría de la siguiente forma:

```
L=Link( [ [1,6,2,7], [2,12,3,11], [3,10,4,11], [7,14,8,1], [9,4,10,5], [12,6,13,5], [13,8,14,9] ] ).
```

Gracias a esta biblioteca podemos visualizar el diagrama asociado al código introducido, como veremos a continuación, además de poder obtener otras propiedades del nudo asociado a dicho código, como por ejemplo, la notación Dowker de este (la veremos más adelante). Dado el objeto *L* instanciado previamente, llamando a la función *view* de la biblioteca *SnapPy*, se nos mostrará en una ventana emergente el diagrama del nudo asociado a dicho código, representado con color rojo en la [Figura 4.4](#).

4. Codificación de nudos

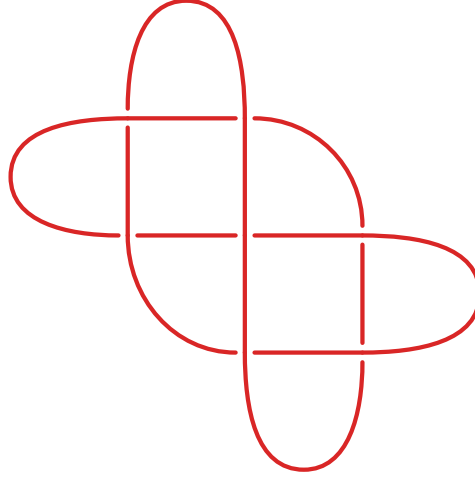


Figura 4.4.: Salida de la función *view* de la clase *Link* para el código de la Tabla 4.1.

Aunque dicho diagrama parezca a simple vista no guardar ninguna relación con el diagrama etiquetado de la Figura 4.3, si tomamos como punto de partida el punto bajo la flecha representada en la Figura 4.5, indicando esta flecha también la orientación del nudo, y etiquetamos las aristas de dicho diagrama siguiendo los pasos descritos previamente, obtenemos el diagrama etiquetado representado en dicha figura, que comprobamos que tiene el mismo código que el diagrama etiquetado de la Figura 4.3.

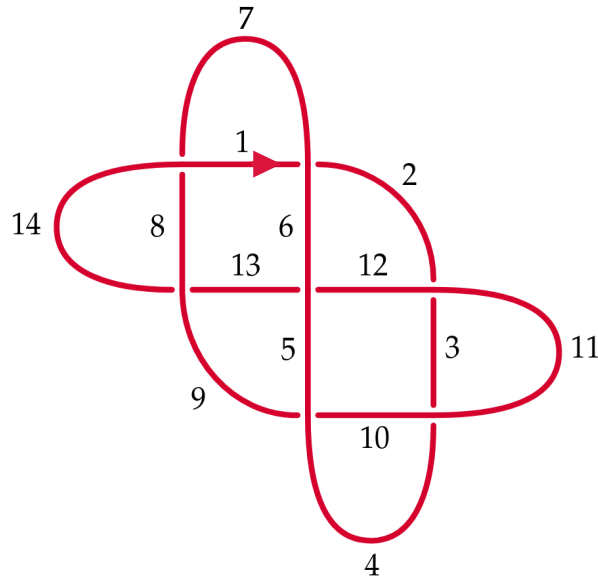


Figura 4.5.: Diagrama etiquetado del diagrama de la Figura 4.4.

También podemos utilizar el sistema algebraico computacional *SageMath*, más conocido como *Sage*, para visualizar el diagrama de un nudo asociado a un determinado código de un diagrama etiquetado. Sin embargo, la clase *Knot* de *Sage*, cuya documentación está

disponible en [MAMB21], interpreta que las aristas pertenecientes a las 4–tuplas asociadas a los cruces del diagrama, son obtenidas al recorrer el cruce en sentido horario, empezando por la arista que forma parte de la parte inferior del cruce y que se recorre antes de llegar a dicho cruce siguiendo la orientación indicada en el diagrama. En consecuencia, para crear una instancia de la clase *Knot* asociada al código de un diagrama etiquetado, habría que permutar las etiquetas en las posiciones 2 y 4 de todas las 4–tuplas de dicho código. Por lo tanto, un objeto de la clase *Knot* cuyo código sea el dado por las 4–tuplas de la [Tabla 4.1](#) se instanciaría así:

$L = \text{Knot}([[1, 7, 2, 6], [2, 11, 3, 12], [3, 11, 4, 10], [7, 1, 8, 14], [9, 5, 10, 4], [12, 5, 13, 6], [13, 9, 14, 8]])$.

Llamando a la función *plot* de la clase *Knot* se nos mostrará el diagrama asociado a dicho código en una ventana emergente, representado en la [Figura 4.6](#).

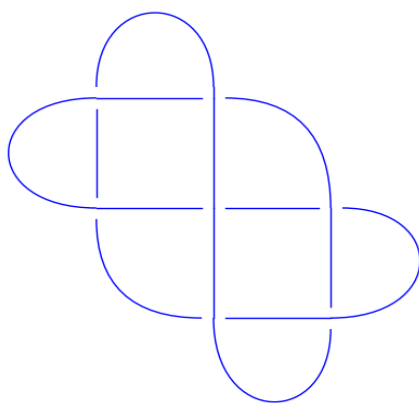


Figura 4.6.: Salida de la función *plot* de la clase *Knot* para el código de la [tabla 4.1](#)

En el caso del diagrama de la [Figura 4.6](#), si tomamos como punto de partida el punto bajo la flecha representada en la [Figura 4.5](#) con color azul, indicando esa flecha también la orientación del nudo, y etiquetamos las aristas de dicho diagrama obtenemos el diagrama etiquetado representado en la figura susodicha, comprobando que tiene el mismo código de diagrama etiquetado que el diagrama etiquetado de la [Figura 4.3](#).

4. Codificación de nudos

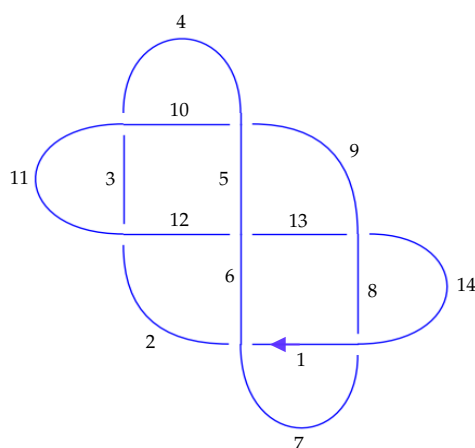


Figura 4.7.: Diagrama etiquetado del diagrama de la Figura 4.6

A continuación vamos a realizar el proceso contrario, dado el código de un diagrama etiquetado, vamos a construir el correspondiente diagrama etiquetado. Por ejemplo, vamos a tratar de reconstruir el diagrama etiquetado de la Figura 4.2a, cuyo código viene dado por las 4-tuplas $(2, 8, 3, 7)(4, 1, 5, 2)(6, 4, 7, 3)(8, 5, 1, 6)$. Comenzamos dibujando los cruces, etiquetando los cuatro extremos de cada cruce con las etiquetas de las aristas dadas por el código calculado anteriormente. Es indiferente la posición en la que situemos dichos cruces, aunque es conveniente situar cerca aquellos cruces con etiquetas comunes; ya veremos en el siguiente paso su justificación. Por ejemplo, podríamos situar los cruces como se muestra en la Figura 4.8.

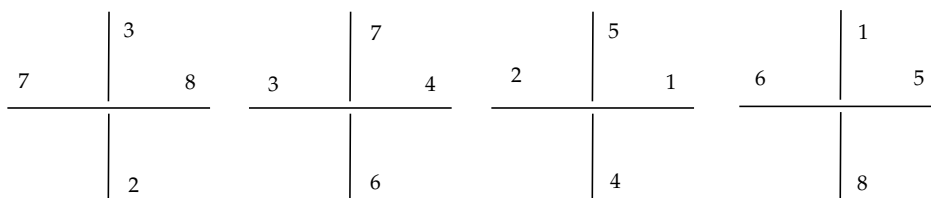


Figura 4.8.: Cruces del diagrama etiquetado de la Figura 4.2a

El siguiente paso es unir los extremos con la misma etiqueta, evitando que se intersecten las aristas recién construídas. Por ejemplo, para los cruces de la Figura 4.8, una forma de unir dichos extremos sería la que se muestra en la Figura 4.9. Además, como las aristas codificadas estaban ordenadas numéricamente en función de la orientación del nudo, podemos deducir también la orientación del nudo recién construído, que señalizaremos con una flecha.

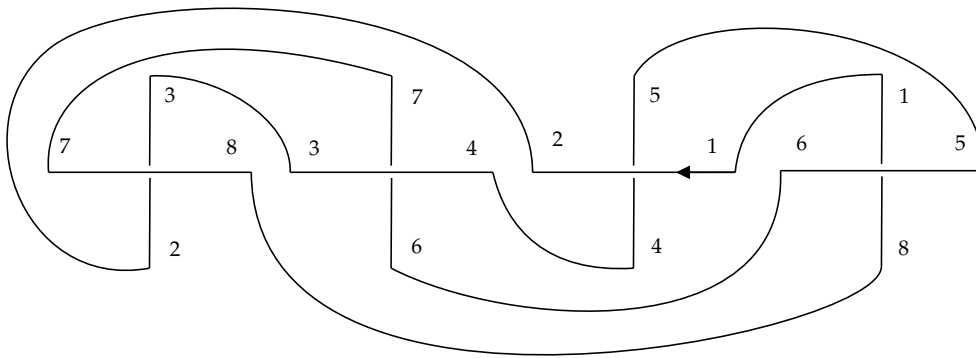


Figura 4.9.: Diagrama etiquetado obtenido de la unión de los extremos de los cruces de la [Figura 4.8](#)

Dado que habíamos afirmado anteriormente que la posición en la que situáramos los cruces era indiferente, si giramos el cruce de la izquierda 90° en sentido horario y el cruce de la derecha 90° en sentido antihorario y unimos los extremos con la misma etiqueta, obtendríamos el diagrama etiquetado de la [Figura 4.10](#), que es mucho más fácil de construir y sería equivalente al diagrama etiquetado de la [Figura 4.9](#).

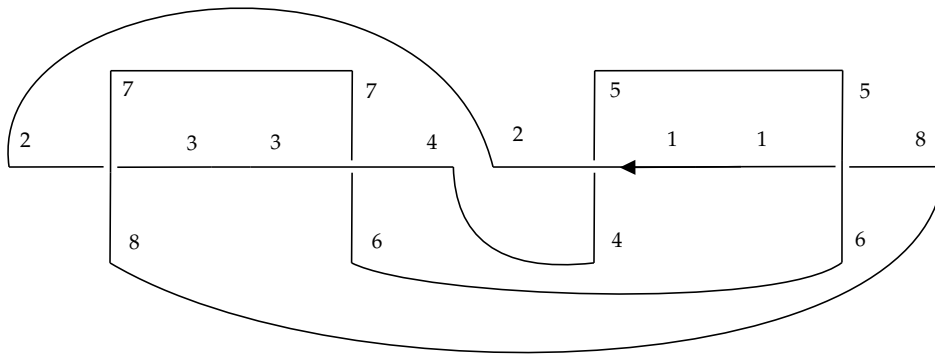


Figura 4.10.: Diagrama etiquetado equivalente al de la [Figura 4.9](#).

Ni que decir tiene que si colocamos los cruces en la misma posición que se encuentran en la [Figura 4.2a](#), uniendo los extremos con la misma etiqueta podríamos obtener ese mismo diagrama etiquetado.

4.2.2. Código de Gauss

Otra notación utilizada para codificar el diagrama de un nudo es el código de Gauss. El objetivo de codificar el diagrama de un nudo utilizando el código de Gauss es, al igual que cuando utilizamos otro tipo de notación, ser capaces de reconstruir un diagrama de un nudo equivalente utilizando dicha codificación. Antes de estudiar el código de Gauss, vamos a definir una serie de conceptos relacionados con los nudos que se mencionarán más adelante en esta subsección.

Definición 4.3. Una reflexión es una aplicación de un espacio euclídeo en sí mismo, que es una isometría con un hiperplano como conjunto de puntos fijos.

Por **imagen especular** de un nudo K nos referimos a la imagen de K bajo la reflexión \mathcal{R} definida por $\mathcal{R}: (x, y, z) \rightarrow (x, y, -z)$. Diremos que un nudo es **quiral** si es equivalente a su imagen especular. En caso contrario, diremos que el nudo es quiral. La circunferencia y el nudo de ocho son nudos aquirales mientras que el nudo trébol es quiral.

Dadas las proyecciones de dos nudos, podemos definir un nuevo nudo cuya proyección será obtenida eliminando un pequeño arco de cada proyección de los nudos iniciales y conectando los cuatro extremos resultantes dos a dos, mediante dos nuevos arcos sin que dichos arcos se intersecten. Si denotamos los dos nudos por los símbolos K_1 y K_2 , su composición se denotará por $K_1 \# K_2$. Para hacer la composición hemos asumido que las proyecciones de los dos nudos no se solapan y los dos arcos que hemos eliminado están en el exterior de la proyección y no contienen ningún cruce. Por ejemplo, el resultado de la composición del nudo de ocho y el nudo trébol es el que aparece en la [Figura 4.11](#).

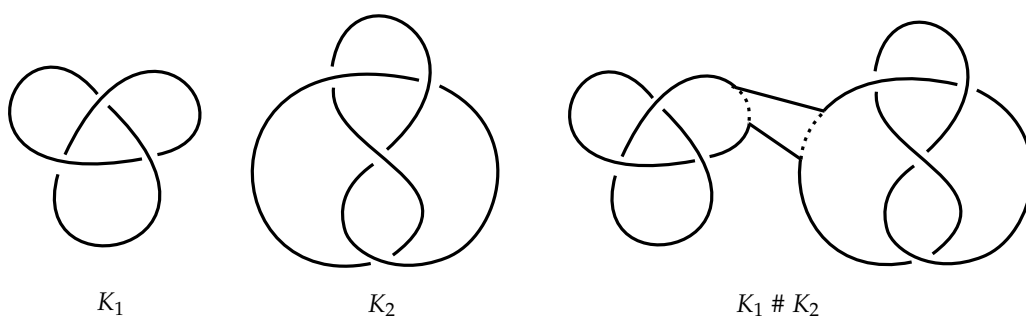


Figura 4.11.: Composición del nudo trébol y el nudo de ocho

Definición 4.4. Un nudo K se dice que es un **nudo compuesto** si puede ser expresado como composición de dos nudos, ninguno de ellos siendo el nudo trivial. Los nudos no triviales de los que se compone un nudo compuesto se llaman nudos factores.

Si un nudo no es composición de ningún par de nudos no triviales, lo llamaremos **nudo primo**. Puede ser un problema no trivial determinar si un determinado nudo es primo o no.

Centrándonos de nuevo en el código de Gauss, dado el diagrama de un nudo, dicho código es quizás el más fácil de describir. Supongamos que tenemos el diagrama de un nudo

arbitrario del cual también tenemos su orientación. Tomamos un punto arbitrario P que se encuentre en una arista del diagrama, y a partir de dicho punto P , comenzamos a recorrer el diagrama del nudo en el sentido que indique de la orientación de este, hasta que lleguemos al primer cruce, que etiquetaremos con el número 1. Seguimos recorriendo el diagrama hasta llegar al próximo cruce que, si no está etiquetado, lo marcamos con el número 2. En caso contrario, lo saltaríamos sin etiquetarlo de nuevo. Repetiremos este proceso hasta que todos los cruces estén etiquetados una vez.

Una vez estén etiquetados todos los cruces del diagrama del nudo, lo recorreremos otra vez comenzando en el punto P . A medida que vamos recorriendo el diagrama, partiendo del punto P y en el sentido que marca la orientación del nudo, para cada cruce que nos encontramos escribimos el número con el que está etiquetado dicho cruce, añadiéndole signo negativo si estamos pasando el cruce por la hebra inferior del cruce. Seguimos con este procedimiento hasta volver al punto P . Vamos a ver como se realizaría el etiquetado de los cruces de un diagrama con un ejemplo.

En la [Figura 4.12](#) se puede ver el diagrama de un nudo orientado para el que se ha escogido un punto arbitrario P en una arista de dicho diagrama, señalado con un punto azul, y se ha realizado la numeración de los cruces siguiendo el método descrito anteriormente.

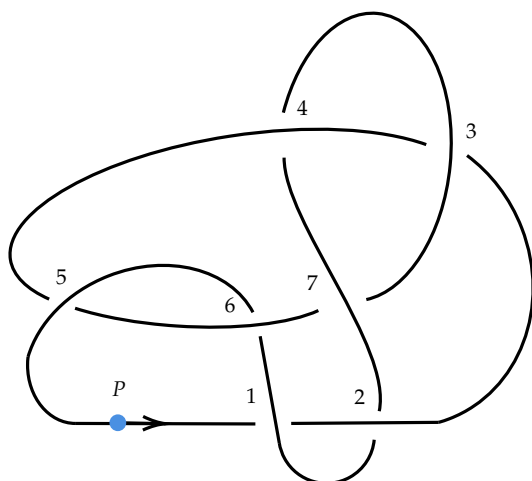


Figura 4.12.: Numeración de los cruces del diagrama de un nudo orientado. Código de Gauss.

En consecuencia, si recorremos el diagrama de la [Figura 4.12](#) partiendo del punto P y en el sentido que indica la orientación del nudo, el código de Gauss asociado a dicho diagrama será:

$$-1, 2, -3, 4, -5, 6, -7, 3, -4, 7, -2, 1, -6, 5.$$

Como era de prever, cada número aparece en el código dos veces, una con signo positivo y otra con signo negativo, ya que cada cruce es atravesado por una hebra superior (signo positivo) y por una hebra inferior (signo negativo). Por motivos de normalización, se puede multiplicar todo por -1 para que el primer valor sea positivo.

4. Codificación de nudos

$$1, -2, 3, -4, 5, -6, 7, -3, 4, -7, 2, -1, 6, -5$$

Se puede normalizar de esta forma dado que el mismo código de Gauss determina tanto a un nudo como a su imagen especular, como nos dice el siguiente teorema.

Teorema 4.1. *Si dos diagramas de nudos primos tienen el mismo código de Gauss, entonces o son isotópicos o uno es la imagen especular del otro.*

Por ejemplo, consideramos el diagrama del nudo trébol y el diagrama de la imagen especular del nudo de trébol. Para ambos diagramas seleccionamos el punto P , y la orientación de este señalados en la [Figura 4.13a](#) y en la [Figura 4.13b](#), y realizamos la numeración de los cruces de dichos diagramas, que también se ve reflejada en ambas figuras.

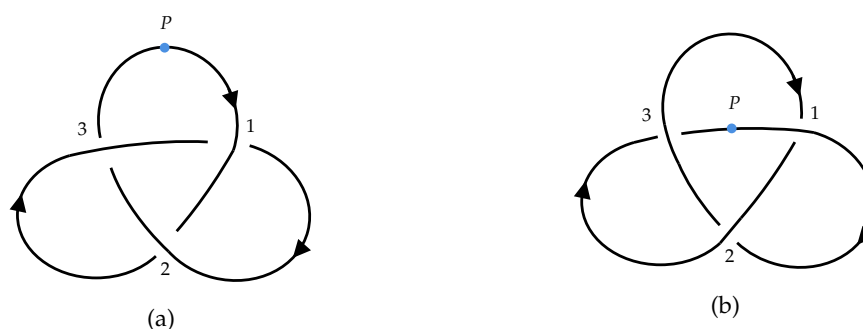


Figura 4.13.: (a) Diagrama del nudo trébol. (b) Diagrama de la imagen especular.

Si calculamos el código de Gauss para los diagramas de la [Figura 4.13a](#) y de la [Figura 4.13b](#), observamos que ambos tienen el mismo código de Gauss asociado, siendo este:

$$1, -2, 3, -1, 2, -3.$$

Sin embargo, habíamos estudiado que el nudo trébol es un nudo quiral y, por tanto, no es equivalente a su imagen especular. En consecuencia, hemos comprobado que los diagramas de dos nudos pueden tener el mismo código de Gauss, pero en cambio, no ser equivalentes los nudos asociados a dichos diagramas.

Debido a que hemos probado que los diagramas de dos nudos equivalentes pueden tener el mismo código de Gauss asociado, lo que no cumple con nuestro objetivo de identificar un tipo de nudo mediante un código o notación, vamos a ver un código muy relacionado con el código de Gauss que nos dará más información que este último.

4.2.3. Código de Gauss Extendido

En general, el código de Gauss para un diagrama de nudos no puede utilizarse para reconstruir el diagrama de un nudo equivalente, pero una pequeña revisión de la notación hará posible dicha reconstrucción. Esta revisión se denomina **código de Gauss extendido**.

Si nos fijamos, hay cierta redundancia en el código de Gauss, ya que sabemos que la segunda aparición de un número en el código de Gauss de un diagrama cualquiera va a tener

signo opuesto al primero. En consecuencia, con el código de Gauss extendido vamos a tratar de eliminar dicha información redundante del código de Gauss y añadir nueva información útil acerca del diagrama.

Los cruces los etiquetaremos como lo hicimos con el código anterior, pero a la hora de registrar las etiquetas modificaremos las reglas para asignarles signo a estas. Una vez estén etiquetados los cruces del diagrama, la primera vez que nos encontremos con un determinado cruce, lo haremos como antes, si recorremos la hebra superior registraremos el número de la etiqueta con signo positivo, y en caso contrario, con signo negativo. La segunda vez que lleguemos a un cruce, registraremos su etiqueta con un signo u otro dependiendo del signo del cruce. Si el cruce forma un tornillo de la mano derecha (desde la hebra inferior a la superior), le asignamos signo positivo, y en caso contrario le asignamos signo negativo, como se refleja en la **Figura 4.14**.

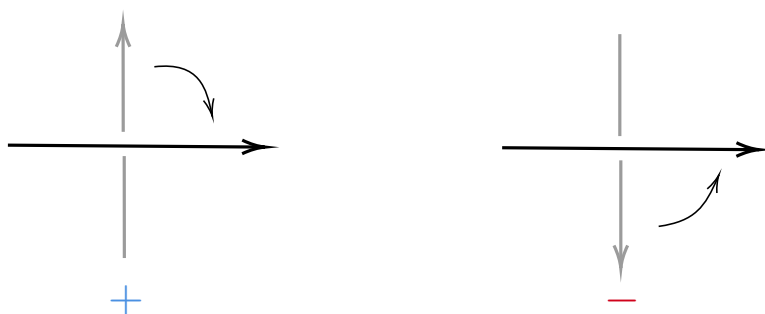


Figura 4.14.: Signo de los cruces en el código de Gauss extendido.

Por ejemplo, dados los diagramas de la **Figura 4.13**, el código de Gauss extendido asociado a estos sería

- 1, -2, 3, 1, 2, 3 para el diagrama de la **Figura 4.13a**,
- 1, -2, 3, -1, -2, -3 para el diagrama de la **Figura 4.13b**.

Por tanto, hemos comprobado que dichos diagramas, que tenían el mismo código de Gauss asociado, tienen distinto código de Gauss extendido, debido a la ya mencionada información extra de la orientación de los cruces que ofrece el código de Gauss extendido sobre el código de Gauss.

Para el diagrama de la **Figura 4.12**, en la **Figura 4.15** podemos observar el signo de todos los cruces de dicho diagrama.

4. Codificación de nudos

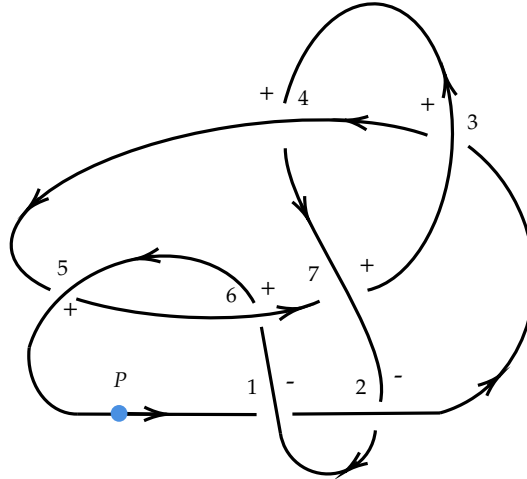


Figura 4.15.: Signo de los cruces del diagrama de la Figura 4.12.

En consecuencia, el código de Gauss extendido para el diagrama de la Figura 4.12 será

$$-1, 2, -3, 4, -5, 6, -7, \textcolor{red}{3}, \textcolor{red}{4}, \textcolor{red}{7}, -2, -\textcolor{red}{1}, \textcolor{red}{6}, \textcolor{red}{5},$$

donde hemos señalado en color rojo los números cuyo signo depende del signo del cruce.

Otra forma de representar los datos del código de Gauss extendido es mediante la utilización de dos listas, siendo la primera de estas el código de Gauss asociado a dicho código de Gauss extendido, y la segunda de ellas conteniendo información del signo de cada uno de los cruces del diagrama. La segunda lista estará compuesta por los valores 1 y -1 dependiendo de los signos de los cruces, 1 si el cruce es positivo y -1 si el cruce es negativo, siendo el primer valor de esa lista el signo del cruce etiquetado con el número 1, y así sucesivamente. Si nos fijamos, la única diferencia con respecto al código de Gauss extendido es que el signo de los cruces con esta nomenclatura se expone en una lista compuesta de 1 y -1 , mientras que con el código de Gauss extendido el signo del cruce acompaña a la segunda aparición del cruce en dicho código.

Para el diagrama de la Figura 4.12, la nomenclatura mencionada en el párrafo anterior asociada a dicho diagrama sería

$$\{-1, 2, -3, 4, -5, 6, -7, \textcolor{red}{3}, -\textcolor{red}{4}, \textcolor{red}{7}, -2, \textcolor{red}{1}, -\textcolor{red}{6}, \textcolor{red}{5}\} \{-1, -1, 1, 1, 1, 1\},$$

que como hemos descrito en el párrafo anterior, ofrece la misma información que el código de Gauss extendido.

Dado un determinado código de Gauss extendido asociado al diagrama de un nudo, podemos construir lo que llamaremos un **diagrama de Gauss** asociado a dicho código. Para la construcción de dicho diagrama me he basado en el artículo *On Gauss diagrams of Knots: A modern approach* [KNN⁺18]. Para la construcción del diagrama de Gauss asociado al diagrama de un nudo se seguirán los siguientes pasos:

- I. Construimos una circunferencia y escogemos un punto de esta, que denotaremos por P' .
- II. Recorremos la circunferencia y vamos etiquetando puntos de la circunferencia con el valor absoluto de los valores del código de Gauss extendido, siguiendo el orden en el que se encuentran situados en dicha lista hasta terminar de recorrer la circunferencia, tras lo cual debemos haber asignado puntos a todos los valores del código de Gauss extendido.
- III. A continuación, dibujamos caminos que unan los puntos de la circunferencia con la misma etiqueta.
- IV. Para cada cruce, el punto inicial de su camino asociado será el punto de la circunferencia que tenga asociada la etiqueta de dicho cruce, y que dicha etiqueta sea la asociada a pasar dicho cruce por encima. Su punto final será por tanto el punto de la circunferencia que tenga asociada la etiqueta de dicho cruce, y que dicha etiqueta sea la asociada a pasar dicho cruce por debajo.
- v. Etiquetamos cada camino con el signo de su cruce asociado.

Por ejemplo, dado el código de Gauss extendido obtenido del diagrama de la [Figura 4.15](#), el diagrama de Gauss asociado a dicho código de Gauss extendido sería el representado en la [Figura 4.16](#), que se ha construido siguiendo los pasos enumerados anteriormente.

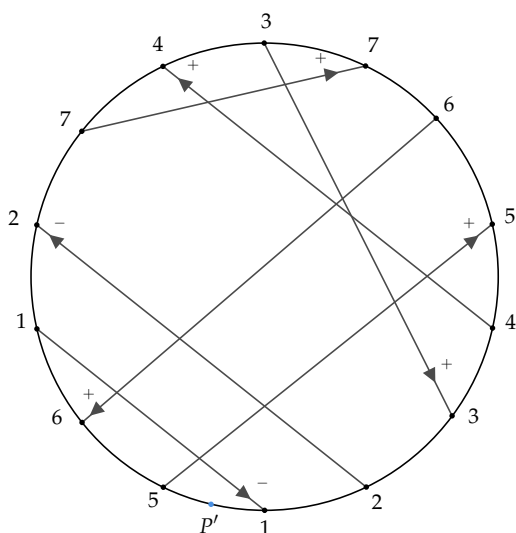


Figura 4.16.: Diagrama de Gauss para el código $-1, 2, -3, 4, -5, 6, -7, 3, 4, 7, -2, -1, 6, 5$.

La importancia del diagrama de Gauss se ve reflejada en la siguiente proposición.

Proposición 4.1. *Dos nudos no equivalentes K y K' , no pueden tener un mismo diagrama de Gauss, o un mismo código de Gauss extendido asociado. En consecuencia, un determinado diagrama de Gauss o un código de Gauss extendido tendrán asociados nudos equivalentes.*

Previamente, cuando estudiamos el código de Gauss, vimos que tanto el nudo trébol como la imagen especular del nudo trébol, que recordemos que no eran equivalentes al ser

4. Codificación de nudos

el nudo trébol un nudo quiral, podían tener el mismo código de Gauss asociado. Con la información extra que nos da el código de Gauss extendido, la **Proposición 4.1** nos indica que hemos solucionado este grave problema que tenía el código de Gauss.

El sistema *SageMath*, que ya utilizamos previamente, también permite visualizar el diagrama asociado a un código de Gauss extendido. Para ello, debemos crear un objeto de tipo *Knot*, que reciba como argumento el código de Gauss extendido en el formato mencionado en la página 92, es decir, recibiendo como parámetro la lista con el código de Gauss asociado al código de Gauss extendido y la lista compuesta de los valores 1 y -1 que indicarán el signo de los cruces.

Por ejemplo, un objeto de la clase *Knot* asociado al código de Gauss extendido del diagrama de la **Figura 4.15** se instanciaría de la siguiente forma:

```
L=Knot ( [ [ [-1,2,-3,4,-5,6,-7,3,-4,7,-2,1,-6,5] ], [-1,-1,1,1,1,1,1,1] ] )
```

Para visualizar el diagrama asociado, llamaremos al método *plot*, que mostrará en una ventana emergente el diagrama de la **Figura 4.17**, que es el diagrama asociado a dicho código de Gauss extendido.

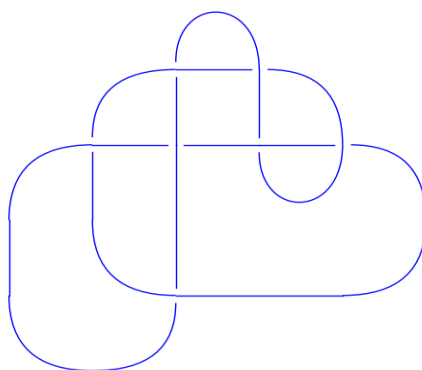


Figura 4.17.: Salida de la función *plot* para el código $-1,2,-3,4,-5,6,-7,3,4,7,-2,-1,6,5$

Aunque de nuevo pueda parecer que el diagrama anterior no guarde ninguna relación con el diagrama de la **Figura 4.15**, si etiquetamos los cruces y tomamos la orientación como se muestra en la **Figura 4.18**, vemos que dichos diagramas tienen el mismo código de Gauss extendido asociado, y por tanto, son diagramas de nudos equivalentes por la **Proposición 4.1**. En dicha figura, también se ha señalado el signo de cada cruce, para que se vea más fácilmente que el código de Gauss extendido de dicho diagrama es $-1,2,-3,4,-5,6,-7,3,4,7,-2,-1,6,5$.

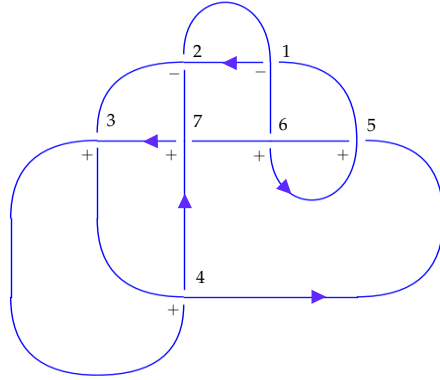


Figura 4.18.: Numeración de los cruces del diagrama de la Figura 4.17

La clase *Knot* de *Sage* tiene un método llamado *pd_code* que devuelve, para un objeto de la clase *Knot*, el conjunto de 4–tuplas de aristas, correspondientes al código de un diagrama etiquetado, pero al igual que habíamos mencionado previamente en la Subsección 4.2.1, nos devuelve las aristas que intervienen en cada cruce empezando por la arista que forma parte de la parte inferior del cruce y que se recorre antes de llegar a dicho cruce, e incluyendo las tres aristas restantes recorriendo en sentido horario el cruce. Ya que habíamos definido que para obtener el código de un diagrama etiquetado debíamos recorrer el cruce en sentido antihorario, debemos de permutar las posiciones 2 y 4 de las 4–tuplas que devuelva la función *pd_code* para obtener el código del diagrama etiquetado asociado a ese objeto de la clase *Knot*. Por ejemplo, si consideramos el objeto de la clase *Knot* definido en la página 94 y llamamos al método *pd_code*, la llamada a dicha función devolverá:

$[[1, 13, 2, 12], [11, 3, 12, 2], [3, 8, 4, 9], [9, 4, 10, 5], [5, 14, 6, 1], [13, 6, 14, 7], [7, 10, 8, 11]]$.

Por lo tanto, el código del diagrama etiquetado asociado a dicha salida será el mostrado en la Tabla 4.2, donde cada cruce tiene asociada una columna.

1	11	3	9	5	13	7
12	2	9	5	1	7	11
2	12	4	10	6	14	8
13	3	8	4	14	6	10

Tabla 4.2.: Código del diagrama etiquetado asociado al código de Gauss extendido $-1, 2, -3, 4, -5, 6, -7, 3, 4, 7, -2, -1, 6, 5$.

Vamos a tratar de reconstruir el diagrama etiquetado a partir de las 4–tuplas de la Tabla 4.2, siguiendo el método descrito en la Subsección 4.2.1. Para ello comenzaremos colocando los cruces, etiquetando los cuatro extremos de cada cruce con las etiquetas de las aristas asociadas a dicho cruce, como se indica en la Tabla 4.2. Colocaremos dichos cruces

4. Codificación de nudos

con cierto ingenio (recordemos que podíamos colocarlos a nuestro gusto) como se muestra en la [Figura 4.19](#).

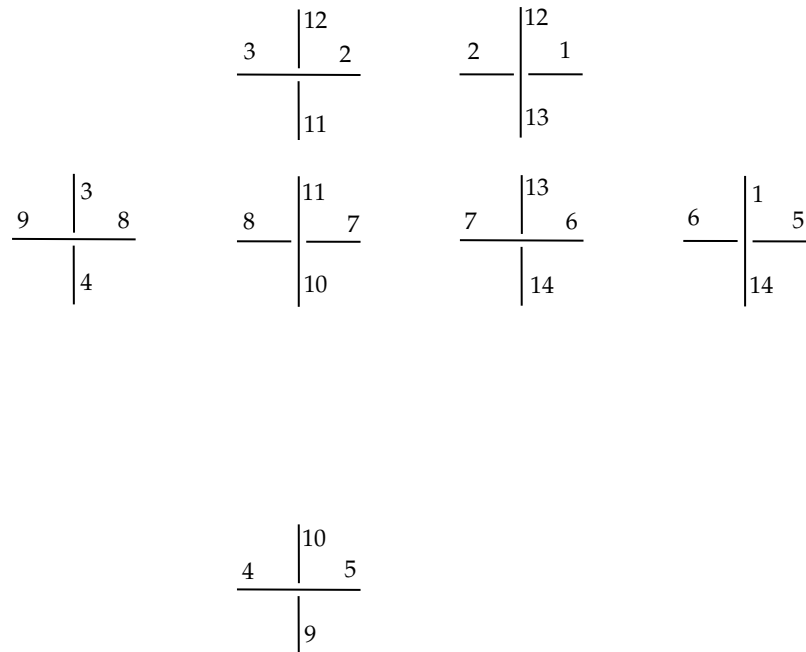


Figura 4.19.: Cruces obtenidos de la [Tabla 4.2](#)

A continuación, siguiendo con el método de reconstrucción del diagrama etiquetado, unimos los extremos con la misma etiqueta, evitando que se intersecten las aristas recién construídas. Una vez hayamos unido todos los extremos con la misma etiqueta tendremos como resultado el diagrama etiquetado de la [Figura 4.20](#), cuya orientación se infiere de la numeración de las aristas.

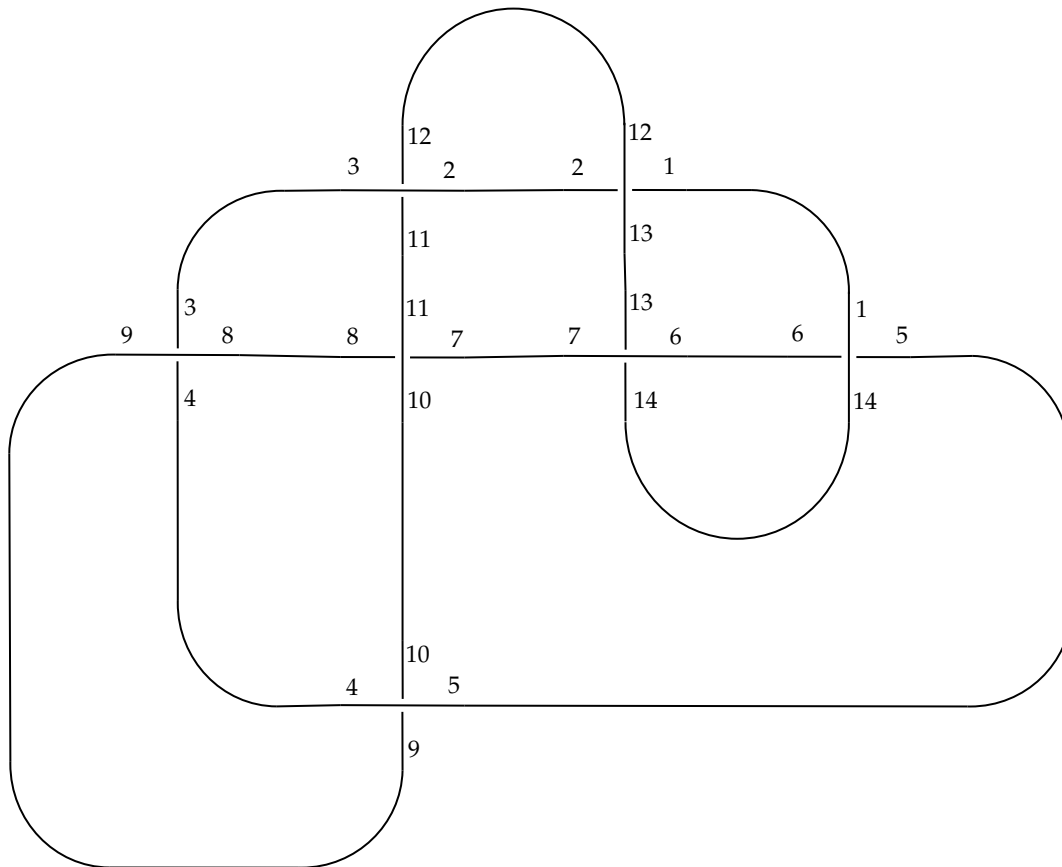


Figura 4.20.: Unión de los extremos de los cruces de la **Figura 4.19**

Si nos fijamos, hemos conseguido obtener el mismo diagrama que teníamos en la [Figura 4.18](#) a partir del código del diagrama etiquetado obtenido del código de Gauss extendido asociado al diagrama de la figura antes referenciada.

4.2.4. Notación Dowker de un nudo

La notación Dowker es una forma bastante simple de describir el diagrama de un nudo. Primero de todo, vamos a definir el concepto de nudo alterno.

Definición 4.5. Un nudo poligonal en posición regular respecto a una proyección ortogonal sobre un plano se dice que es alterno si los puntos superiores e inferiores de este, se alternan alrededor de nudo.

Vamos a etiquetar el diagrama de un nudo alterno para posteriormente poder hallar la notación Dowker de dicho diagrama. Dado un nudo alterno K , tomamos un cruce cualquiera del diagrama, que etiquetaremos con el valor 1, y elegimos una orientación de dicho diagrama. Seguimos recorriendo el diagrama en la orientación de este, y etiquetamos el siguiente cruce con el valor 2. Continuamos etiquetando los cruces del diagrama con números naturales consecutivos, y a diferencia de lo que vimos en el etiquetado de un diagrama para obtener el código de Gauss, etiquetaremos de nuevo los cruces por los que ya hayamos pasado una vez. De esta forma, cada cruce tendrá asociadas dos etiquetas, y puesto que el nudo es alterno, una de esas dos etiquetas será par y la otra impar.

Supongamos que queremos etiquetar el diagrama de un nudo alterno como el de la **Figura 4.21** para hallar posteriormente la notación Dowker de este.

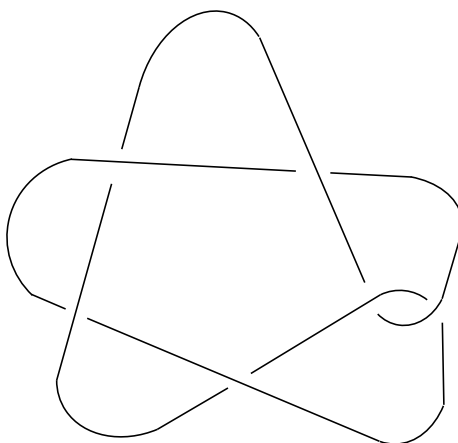


Figura 4.21.: Diagrama de un nudo alterno.

Seleccionamos un cruce y una orientación del diagrama del nudo, y realizamos el proceso de etiquetado de los cruces empezando por el cruce seleccionado y siguiendo la orientación tomada. El resultado del proceso de etiquetado del diagrama de la **Figura 4.21** es el mostrado en la **Figura 4.22**.

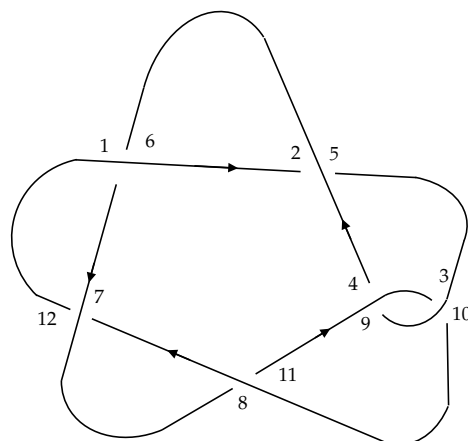


Figura 4.22.: Diagrama de la Figura 4.21 numerado.

Podemos almacenar la información de las etiquetas asociadas a cada cruce en una tabla de dos filas y tantas columnas como cruces tenga el diagrama. Cada cruce tendrá asociada una columna, y dado que cada cruce tendrá asociado un valor impar y otro par, la fila de arriba contendrá los valores impares ordenados de las etiquetas de los cruces, y la fila de abajo los valores pares asociados a los valores impares de la fila superior. De esta forma, el diagrama numerado de la Figura 4.22 tendrá la Tabla 4.3 asociada.

1	3	5	7	9	11
6	10	2	12	4	8

Tabla 4.3.: Tabla de cruces asociada al diagrama de la Figura 4.22.

Dado que hemos impuesto que los valores impares almacenados en la tabla asociada a un diagrama numerado estén ordenados de menor a mayor, como abreviatura de la información que hay en la Tabla 4.3, podemos simplemente escribir 6 10 2 12 4 8 y tener en cuenta que el 1 está emparejado con el 6, el 2 con el 5, etc. Por tanto, del diagrama numerado de un nudo obtenemos una secuencia de naturales pares, en la que el número de elementos de la secuencia nos da el número de cruces del nudo. A esta secuencia de naturales pares la llamaremos **notación Dowker** de un nudo.

Ahora vamos a tratar de realizar el proceso contrario, dada una secuencia de números pares, es decir, dada la notación Dowker de un nudo alterno, queremos representar el diagrama de dicho nudo. Vamos a ver con un ejemplo como se realizaría.

Trataremos de obtener el diagrama del nudo alterno asociado a la notación Dowker 8 10 12 2 14 6 4, que es la abreviatura de la información de la Tabla 4.4.

4. Codificación de nudos

1	3	5	7	9	11	13
8	10	12	2	14	6	4

Tabla 4.4.: Tabla de cruces asociada a la secuencia 8 10 12 2 14 6 4.

Vamos a empezar a reconstruir el nudo. Comenzaremos por el valor 1, y dibujaremos el cruce asociado a dicho valor, es decir, el etiquetado con los valores 1 y 8. Utilizaremos la convención de que empezaremos recorriendo dicho cruce por debajo. Posteriormente, dibujamos el siguiente cruce, el asociado al valor 2, etiquetado con un 2 y un 7, y dado que es un nudo alterno y por la convención antes mencionada, sabemos que debemos pasar por encima de ese cruce. Por lo tanto, unimos el primer cruce con este último mediante una arista. Continuamos con este proceso hasta que el próximo valor que analicemos ya esté presente en alguno de los cruces existentes. Entonces sabemos que debemos rodear lo que llevamos de nudo para pasar por ese cruce. Sin embargo, tenemos dos opciones para volver a pasar por dicho cruce, girar a la izquierda y pasar por encima de lo que llevamos de nudo, o girar a la derecha y pasar por debajo de este. De momento, ignoremos esta ambigüedad y elijamos cualquiera de las opciones. En el ejemplo que estamos estudiando, al llegar el valor número 7, verificamos que al estar el cruce etiquetado con el valor 7 ya dibujado, tendremos que unir con una arista el cruce etiquetado con los valores 6 y 11 al cruce etiquetado con los valores 2 y 7. Si giramos a la izquierda y rodeamos el nudo por arriba obtendríamos el diagrama simbolizado en la [Figura 4.23](#).

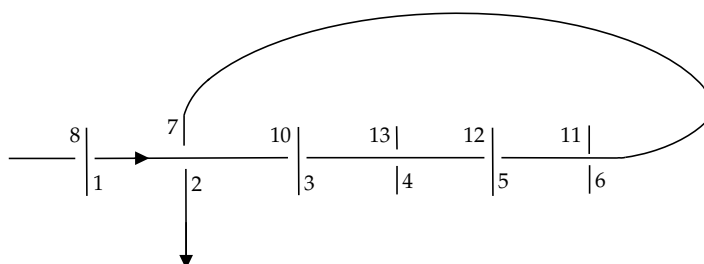


Figura 4.23.: Primeros pasos de la representación del diagrama del nudo.

Tras pasar por segunda vez por el cruce etiquetado con los valores 2 y 7, seguimos la misma metodología, si el siguiente valor analizado no está presente en un cruce ya representado, dibujamos un nuevo cruce, y en caso contrario volvemos a recorrer dicho cruce, teniendo en cuenta que los cruces se alternan a medida que vamos recorriendo el nudo. Siguiendo este algoritmo, llegaríamos a obtener el diagrama representado en la [Figura 4.24](#).

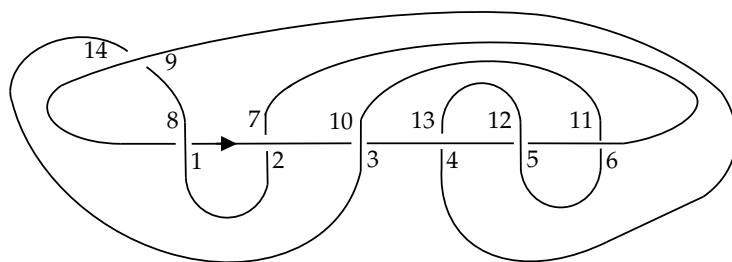


Figura 4.24.: Diagrama reconstruido para la notación Dowker 8 10 12 2 14 6 4.

A continuación, vamos a centrarnos en la ambigüedad en la elección del sentido de giro del nudo cuando tenemos que pasar por segunda vez por un cruce, como ocurría por ejemplo en la reconstrucción del diagrama realizada previamente, cuando teníamos que volver a pasar por el cruce etiquetado con los valores 2 y 7. Vamos a comprobar que dicha elección puede provocar que los nudos asociados a dos diagramas distintos con la misma notación Dowker no sean equivalentes, y que uno no sea la imagen especular del otro.

Por ejemplo, la notación Dowker 4 6 2 10 12 8, está asociada a dos nudos que resultarán ser no equivalentes y no siendo uno la imagen especular del otro, como se muestra en la Figura 4.25. Para la construcción del diagrama de la Figura 4.25a, tras recorrer por primera vez el cruce etiquetado con los valores 9 y 12, tenemos que volver a pasar por el cruce etiquetado con los valores 7 y 10, y decidimos girar a la izquierda para llegar a dicho cruce. En cambio, para la construcción del diagrama de la Figura 4.25b, en vez de girar a la izquierda en ese punto, decidimos girar a la derecha.

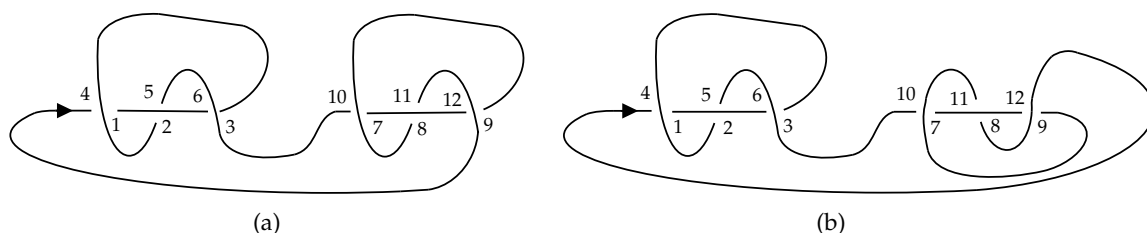


Figura 4.25.: Diagramas de dos nudos cuya notación Dowker es 4 6 2 10 12 8.

Si nos fijamos, la secuencia del ejemplo anterior, 4 6 2 10 12 8, es una permutación de los tres primeros números pares naturales, 2, 4 y 6, concatenada con una permutación de los tres números pares naturales siguientes, 8, 10 y 12. En el siguiente párrafo veremos que este es el motivo por el cual los nudos asociados a dicha notación, representados en la Figura 4.25, no son equivalentes ni uno es la imagen especular del otro.

Si vemos la notación Dowker como una permutación de números pares, cuando la notación Dowker puede ser separada en dos subpermutaciones, el nudo resultante será compuesto, y el nudo no estará completamente determinado por dicha notación Dowker. Sin embargo, si la notación Dowker es una permutación de números pares que no pueda ser separada en dos subpermutaciones, dicha notación Dowker determinará a un nudo o a su imagen especular. Por tanto, cuando el nudo determinado por dicha notación sea aquiral,

4. Codificación de nudos

dicha notación Dowker solo determinará un tipo de nudo.

El sistema que hemos explicado funciona muy bien para describir el diagrama de un nudo alterno, pero vamos a intentar extenderlo a nudos que no sean alternos. Para ello, a la hora de etiquetar los cruces del diagrama de un nudo, el número par asociado a cada cruce tendrá un signo asociado. Para el proceso de etiquetado de un diagrama de un nudo no alterno seguiremos los siguientes pasos:

- I. Recorremos el nudo usando el sistema de etiquetas que hemos usado anteriormente, de forma que cada cruce tenga un número par y un número impar asociado.
- II. Para cada cruce, si el número par es asignado a dicho cruce mientras estamos recorriendo la parte superior de este, dejamos el número par positivo. En cambio, si el número par es asignado a dicho cruce mientras recorremos la parte inferior de este, añadimos el signo $-$ a ese número par.

Por ejemplo, para el diagrama de la [Figura 4.26](#), los números pares 6 y 10 pasan a ser negativos, ya que son asignados cuando se recorre la parte inferior de sus cruces asociados, siendo por tanto su notación Dowker $8 - 10 12 2 14 - 6 4$.

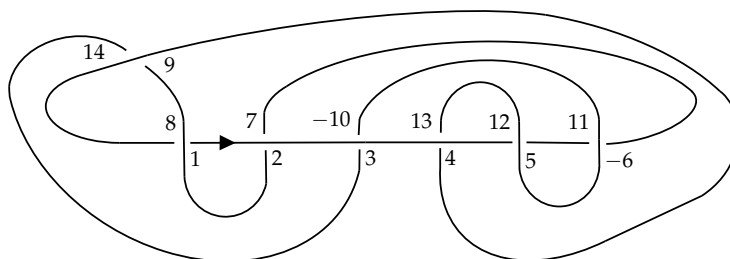
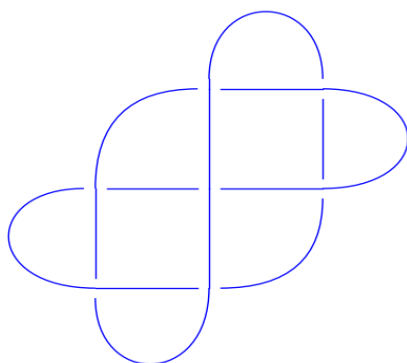


Figura 4.26.: Diagrama asociado al código Dowker $8 - 10 12 2 14 - 6 4$

El sistema *Sage* nos permite también visualizar el diagrama del nudo asociado a una determinada notación Dowker. Para ello, debemos definir un objeto de la clase *Knots*, y una vez tengamos instanciado un objeto de dicha clase, llamaremos al método *from_dowker_code* de dicha clase, pasándole como argumento la notación Dowker del nudo del cual queremos visualizar su diagrama, y dicho método devolverá un objeto de la clase *Knot*. Por ejemplo, para visualizar el diagrama asociado al código Dowker $8 - 10 12 2 14 - 6 4$, se haría de la siguiente forma:

```
W=Knots()
W1=W.from_dowker_code([8, -10, 12, 2, 14, -6, 4])
W1.plot()
```

La ejecución de esta última sentencia muestra en una ventana emergente el diagrama representado en la [Figura 4.27](#).

Figura 4.27.: Salida del método *plot* para el objeto $W1$.

Vamos a verificar que si seleccionamos un cruce inicial y una orientación concretos en el diagrama representado en la Figura 4.27, el diagrama resultante tendrá código Dowker $8, -10, 12, 2, 14, -6, 4$. Para ello, seleccionamos en dicho diagrama el cruce inicial y la orientación simbolizados en la Figura 4.28a, y etiquetamos el resto de cruces del diagrama a partir de dicho cruce inicial y orientación, dando como resultado el diagrama de la Figura 4.28b. A partir de este último diagrama comprobamos que la notación Dowker asociada a dicho diagrama es $8, -10, 12, 2, 14, -6, 4$, como queríamos.

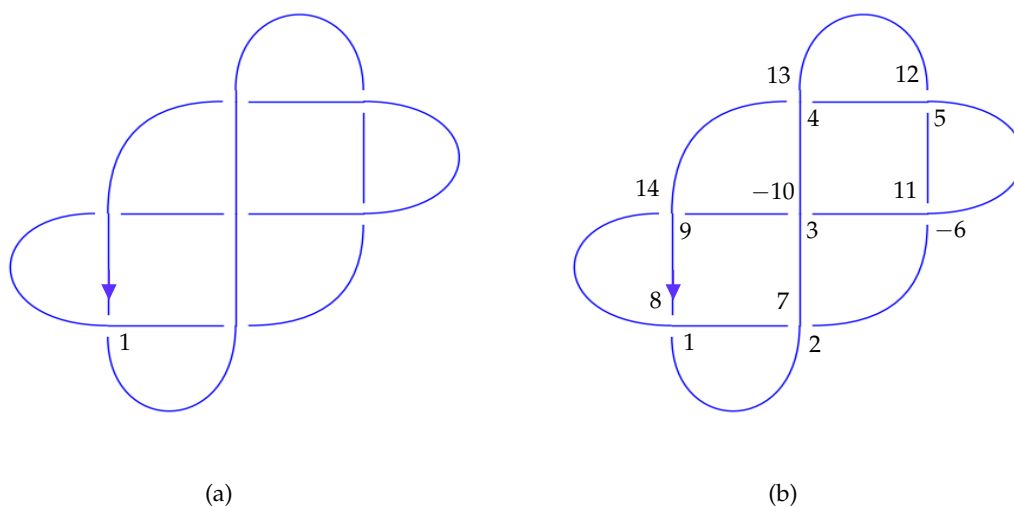


Figura 4.28.: (a) Selección de cruce inicial y orientación. (b) Numeración del diagrama.

Sage también nos permite obtener la notación Dowker de un objeto de la clase *Knot*. Dicha notación Dowker se obtendrá llamando al método *dowker_notation* de la clase *Knot*, aunque el formato en el que dicho método devuelve la notación Dowker del nudo no es el mismo que el estudiado en esta sección. Dicha función devuelve una lista conteniendo los pares de

4. Codificación de nudos

etiquetas asociadas a los cruces del nudo, siendo el primer elemento de cada par, la etiqueta que asignamos al cruce asociado a ese par cuando lo recorremos por debajo, y el segundo elemento del par, la etiqueta que asignamos al cruce asociado a ese par cuando lo recorremos por encima. Por ejemplo, considerando el objeto de la clase *Knot* definido en la página 85, la llamada a la función *dowker_notation* devuelve:

$$[(1, 6), (2, 11), (3, 10), (7, 14), (9, 4), (12, 5), (13, 8)].$$

Por consiguiente, el código Dowker asociado a dicha salida será el representado en la [Tabla 4.5](#).

1	3	5	7	9	11	13
6	10	-12	14	4	-2	8

Tabla 4.5.: Tabla de cruces asociada a la lista $[(1, 6), (2, 11), (3, 10), (7, 14), (9, 4), (12, 5), (13, 8)]$.

Dado que si reconstruimos el diagrama a partir de dicha notación Dowker debe resultar un diagrama equivalente al de la [Figura 4.6](#) (suponiendo que la función haya devuelto el código Dowker correcto), nos damos cuenta que etiquetando en dicho diagrama el cruce señalado en la [Figura 4.29a](#) con el valor 1, y considerando la orientación simbolizada con una flecha en dicha figura, si etiquetamos el resto de cruces siguiendo el método descrito en esta sección y calculamos la notación Dowker, resulta ser la que queríamos, la dada por la [Tabla 4.5](#). El etiquetado del diagrama de la [Figura 4.29a](#) está representado en la [Figura 4.29b](#).

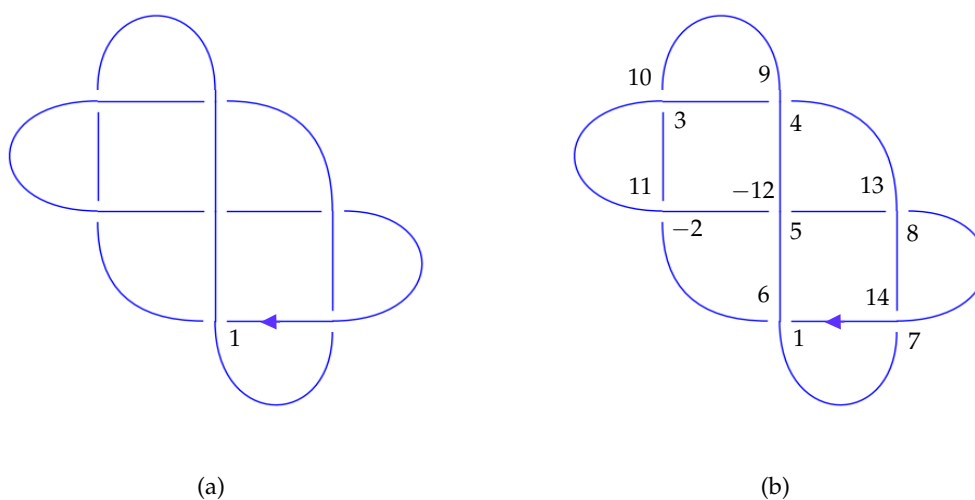


Figura 4.29.: (a) Selección de cruce inicial y orientación. (b) Numeración del diagrama.

La notación Dowker nos permite asociarle a un diagrama de un nudo una secuencia de números pares. Supongamos que queremos clasificar los nudos de 13 cruces con la notación Dowker. El número de permutaciones sin repetición de los 13 primeros números pares

2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26 es $13!$, que es aproximadamente 6 mil millones. Después, dado que dichos números pares pueden ser positivos o negativos, habría que multiplicar $13!$ por 2^{13} . Obviamente no hay esta cantidad de nudos diferentes con 13 cruces ya que muchas de esas secuencias representan al mismo nudo, es más, muchas de esas secuencias representan el mismo diagrama de un nudo.

Para que se vea la importancia de esta notación, en 1982 dos matemáticos ingleses, Hugh Dowker y Morwen Thistlethwaite, usaron la notación Dowker para implementar un algoritmo que generara nudos con 13 o menos cruces y de esta manera poder tabularlos. Dicho programa concluyó con una lista de 12965 nudos primos, que aunque se demostró que eran distintos, no había certeza de la completitud de esa lista, y existía la posibilidad de que algunos de los nudos listados no fueran primos. La clasificación de los nudos estuvo estancada durante el resto de la década de los 80, hasta que un matemático llamado Jim Hoste retomó su clasificación con la ayuda de una supercomputadora. Desde ese momento, Jim Hoste, Morwen Thistlethwaite y un matemático estadounidense llamado Jeff Weeks, comenzaron con la tabulación de los nudos primos de más de 13 cruces hasta que en el año 1998 publicaron en el artículo *The first 1, 701, 936 knots* [HTW98] que habían conseguido tabular los nudos primos de 16 cruces o menos. En la Tabla 4.6 podemos ver el número de nudos primos distintos con 16 cruces o menos.

Nº cruces	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nº nudos primos	1	1	2	3	7	21	49	165	552	2176	9988	46972	253293	1388705

Tabla 4.6.: Número de nudos primos distintos con 16 cruces o menos.

5. Estudio computacional de un nudo mediante su notación Dowker

En el capítulo anterior, estudiamos diferentes formas de codificar un nudo, para que estos pudieran ser identificados mediante un conjunto finito de datos. En este capítulo, a través de la notación Dowker de un nudo, trataremos de reconstruir un diagrama de ese nudo que además sea poligonal, para poder ser capaces calcular una presentación superior e inferior del grupo de ese nudo, siguiendo el método que describimos en el capítulo 3. Para conseguir nuestro objetivo, crearemos una clase en *Python* cuyo único atributo sea la notación Dowker del nudo a estudiar, e implementaremos diferentes funciones que realicen computacionalmente los pasos para el cálculo de dichas presentaciones, al igual que nos ayudaremos del módulo *Turtle* de *Python*, para visualizar el diagrama poligonal del nudo y los espacios necesarios para el cálculo de la presentación superior e inferior del grupo de dicho nudo.

5.1. Introducción

El problema central del estudio matemático de los nudos es poder distinguir los nudos no equivalentes para su posterior clasificación. En el capítulo 3 estudiamos el grupo de un nudo, que al ser un invariante de nudos, si comprobábamos que los grupos de dos nudos no eran isomorfos, podíamos asegurar que dichos nudos no eran equivalentes. En dicho capítulo, estudiamos el algoritmo que debíamos seguir para, a partir de un nudo poligonal obtener la presentación superior e inferior del grupo dicho nudo. Dado que en dicho capítulo vimos que para el cálculo de dichas presentaciones era necesario seguir una serie de pasos que hacían de dicho cálculo un proceso largo y propenso a cometer errores; en este capítulo, mediante la construcción de una clase en *Python*, trataremos de obtener el diagrama de un nudo poligonal asociado a una determinada notación Dowker, a partir de la cual, calcularemos computacionalmente la presentación superior e inferior del grupo de ese nudo poligonal.

Para llevar a cabo nuestro cometido, crearemos una clase en *Python* que llamaremos *Nudo*, cuyo constructor reciba como parámetro la notación Dowker de un nudo, e implementaremos métodos para obtener los vértices de las aristas del diagrama del nudo poligonal asociado a dicha notación, dibujar mediante el módulo *Turtle* el diagrama de dicho nudo, dividir el nudo en arcos superiores e inferiores, etc. En el **Apéndice A** documentaremos los métodos más importantes de la clase *Nudo*, que se pueden encontrar en el archivo *claseNudo.py*, disponible en github.com/imm98/TFG. En primer lugar, en la sección **Sección 5.2** estudiaremos mediante un pequeño tutorial las salidas y el funcionamiento interno de algunas funciones para unos objetos de la clase *Nudo* concretos, que nos ayudarán a entender mejor algunos de los métodos implementados en dicha clase. En segundo lugar describiremos las conclusiones obtenidas del estudio computacional de los nudos realizado y los posibles trabajos futuros.

Además, se ha probado el funcionamiento de todas las funciones de la clase *Nudo* para las notaciones Dowker de todos los nudos primos con 9 cruces o menos, notaciones extraídas de [Wik21], además de para otras notaciones Dowker de nudos con más cruces, dando en todos los casos resultados satisfactorios. Las notaciones Dowker de todos los nudos primos con 9 cruces o menos y de los otros nudos mencionados se encuentran comentadas en el archivo *claseNudo.py*, acompañadas de su notación AlexanderBriggs asociada, para probar para una notación Dowker concreta el funcionamiento de los métodos de la clase si se desea.

5.2. Tutorial del funcionamiento de la clase Nudo

En esta sección vamos a estudiar el funcionamiento de los métodos principales de la clase *Nudo*, métodos que tendrán como objetivo principal poder calcular la presentación superior e inferior del grupo de un nudo, a partir de la notación Dowker asociada a este, además de representar gráficamente diferentes diagramas de dicho nudo. Durante este tutorial, instanciaremos diferentes objetos de la clase *Nudo*, para los cuales estudiaremos las salidas de las llamadas a algunos métodos, que nos ayudarán a comprender mejor el funcionamiento de estos.

■ `__eliminar_lazos(self)`:

Este método elimina de la lista `self.__numeros` los pares asociados a lazos existentes en el diagrama de un nudo. Esto se puede realizar debido a que dos diagramas se corresponden con el mismo nudo si se puede pasar de uno a otro mediante movimientos de Reidemeister, siendo uno de los movimientos de Reidemeister el proceso de lo que hemos llamado “eliminar un lazo del diagrama de un nudo”, representado en la [Figura 5.1](#).

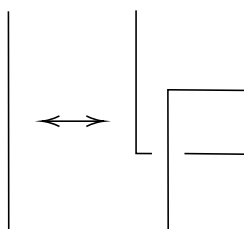


Figura 5.1.: Movimiento de Reidemeister Tipo I.

Vamos a ver lo que ocurre con la lista `self.__numeros` cuando instanciamos algunos objetos de la clase *Nudo*.

```
In [1]: x=Nudo(6, -12, 2, 8, -4, -10)
```

```
Out[1]: self.__numeros: [6, -8, 2, -4]
```

La notación Dowker 6, -12, 2, 8, -4, -10, viene dada por la [Tabla 5.1](#)

1	3	5	7	9	11
6	-12	2	8	-4	-10

Tabla 5.1.: Tabla de cruces asociada a la notación Dowker 6, -12, 2, 8, -4, -10.

En dicha tabla, podemos apreciar que tras representar el cruce etiquetado con los valores 7 y 8 en el diagrama, tendríamos que volver a recorrer dicho cruce, y por lo tanto se produciría un lazo en el diagrama. Igual ocurriría con el cruce etiquetado con las etiquetas -10 y 11. en consecuencia, la notación Dowker equivalente a haber eliminado esos dos lazos sería la resultante de restar (si son positivos) o sumar (si son negativos) 2 a cada número par restante de la notación Dowker anterior, si dichos números pares son mayores en valor absoluto que los valores absolutos de los números eliminados. En la [Figura 5.2](#) se puede ver que los diagramas asociados a las notaciones Dowker 6, -12, 2, 8, -4, -10 y 6, -8, 2, -4 son equivalentes.

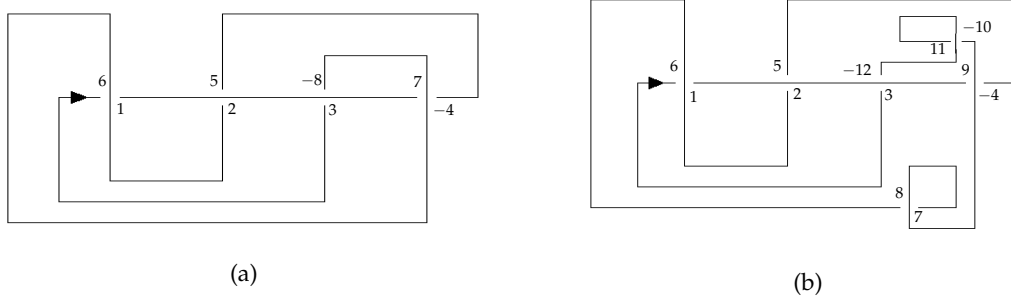


Figura 5.2.: (a) Notación Dowker: 6, -8, 2, -4. (b) Notación Dowker: 6, -12, 2, 8, -4, -10.

En cambio, para la notación Dowker [\[2\]](#), podemos ver que no existen lazos en el diagrama resultante, y en consecuencia, la lista `self.__numeros` no varía.

```
In [2]: x=Nudo (6, 10, 2, 12, 4, 8)
```

```
Out[2]: self.__numeros: [6, 10, 2, 12, 4, 8]
```

■ `__comprobar_notacion_correcta(self)`:

Este método devuelve True si la lista `self.__numeros` está asociada a una notación Dowker válida y False en caso contrario. Si dicho método fuera público y por tanto pudiéramos llamarlo desde fuera de la clase, vamos a ver algunos ejemplos de las salidas que se producirían:

```
In [3]: x=Nudo (6, 10, 2, -10, 4, 8)
        x.__comprobar_notacion_correcta()
```

```
Out[3]: False
```

5. Estudio computacional de un nudo mediante su notación Dowker

```
In [4]: x=Nudo (6, 10, 2, -10, 4, 8)
x.__comprobar_notacion_correcta()
```

```
Out[4]: True
```

La salida de [3] es False ya que dicha lista tiene 6 elementos, y por lo tanto, debería de contener, en valor absoluto, los números pares del 2 al 12, pero ni el 12 ni el -12 se encuentran en dicha lista. En cambio, la lista `self.__numeros` del objeto creado en [4], que también tiene 6 elementos, contiene en valor absoluto los números pares del 2 al 12, y en consecuencia, dicho método devuelve True.

■ numero_arcos_superiores(self):

Devuelve el número de arcos superiores del nudo. El algoritmo que se seguirá es el siguiente:

Mantendremos en una variable booleana que llamaremos *superior* información acerca de si el cruce anterior al que vamos a recorrer actualmente lo hemos cruzado por encima, en cuyo caso *superior* = True, o por debajo, *superior* = False. Cada vez que se pase por encima [debajo] de un cruce y la variable *superior* sea False [True], se sumará uno a un contador que denotaremos por *cont*, y se modificará el valor de la variable *superior* a True [False]. Una vez hayamos recorrido todos los cruces dos veces, el número de arcos superiores del nudo será *cont*/2.

Vamos a ver las salidas de dicho método para dos objetos de la clase Nudo distintos, explicando lo que realiza internamente el método para calcular dicho valor.

```
In [5]: x=Nudo(8, -10, 12, -2, 6, 4)
x.numero_arcos_superiores()
```

```
Out[5]: 3
```

```
In [6]: x=Nudo (8,10,2,12,4, 6)
x.numero_arcos_superiores()
```

```
Out[6]: 6
```

Comencemos por estudiar la salida de [5], para lo cual, representaremos en la **Tabla 5.2** la notación Dowker extendida asociada a dicho nudo, para explicar de forma más clara el funcionamiento del método.

1	3	5	7	9	11
8	-10	12	-2	6	4

Tabla 5.2.: Tabla de cruces asociada a la notación Dowker 8, -10 , 12, -2 , 6, 4.

- I. Dado que todo nudo tiene como mínimo un arco superior y otro inferior, inicializamos el contador a 2, $cont = 2$.
- II. El valor 1, que tiene asociado el cruce dado por 1 y 8, al ser 8 un número positivo significa que recorreremos primero dicho cruce por debajo, por lo que inicializamos $superior = False$.
- III. El valor 2 tiene signo negativo en la lista, por lo tanto su cruce asociado se recorrerá por debajo, y debido a que $superior = False$, no modificaremos dicho valor ni el contador.
- IV. El valor 3 tiene asociado el cruce dado por 3 y -10 , y al ser -10 negativo significa que recorreremos el cruce por arriba. Ya que $superior = False$, sumaremos 1 a la variable $cont$, por lo que $cont = 3$, y estableceremos $superior = True$.
- v. Habría que seguir este algoritmo hasta analizar todos los valores del 1 al 12, y tras realizarlo, obtendríamos que $cont = 6$, por lo que el número de arcos superiores sería $6/2 = 3$, que es la salida obtenida en [5].

Habría que realizar lo mismo en [6], aunque al ser todos los números pares positivos, es un nudo alterno, y por tanto, el número de arcos superiores será igual a la longitud de dicha lista, que es 6 y es la salida de [6] como era de prever.

Vamos a comprobar llamando al método `dibujar_nudo_arcos` que el número de arcos superiores obtenidos en [5] y [6] es correcto. La llamada a dicho método dibujará, utilizando el módulo `Turtle`, los diagramas de los nudos divididos en arcos superiores (color azul) y arcos inferiores (color rojo). Dichos diagramas se pueden ver en la Figura 5.3.

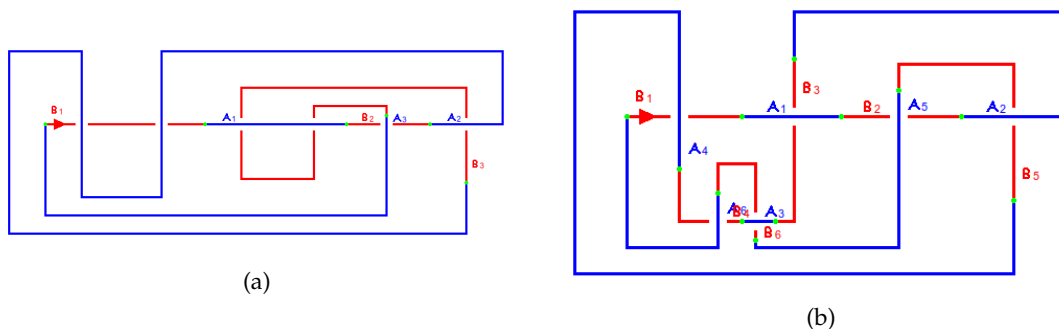


Figura 5.3.: (a) Notación Dowker: 8, -10 , 12, -2 , 6, 4. (b) Notación Dowker: 8, 10, 2, 12, 4, 6.

■ `obtener_puntos_nudo_dowker(self)`:

Este es el método más complejo de la clase `Nudo`. Devuelve una lista de vértices, siendo un vértice un par $[x', y']$, donde x' es la coordenada x , e y' es la coordenada y de dicho vértice. Dichos vértices denotarán los vértices de las aristas que formarán el diagrama del nudo poligonal, siendo los vértices de la arista i del diagrama del nudo, los dados por las posiciones $2i$ y $2i + 1$ de la lista de vértices que devuelve este método. Vamos a ver un ejemplo.

```
In [7]: x=Nudo ()
        x.obtener_puntos_nudo_dowker ()
```

5. Estudio computacional de un nudo mediante su notación Dowker

```
Out[7]: vertices: [[-0.5, 0],[0.5, 0], [0.5, 0], [0.5, -1], [0.5, -1],  
                 [-0.5, -1], [-0.5, -1], [-0.5, 0]]
```

Dado que en este caso la lista `self.__numeros` es vacía quiere decir que el diagrama del nudo no tendrá crucees, por lo tanto los vértices que devuelve el método son los vértices de las aristas del nudo trivial. Los vértices obtenidos de [7] son los representados en la Figura 5.4.

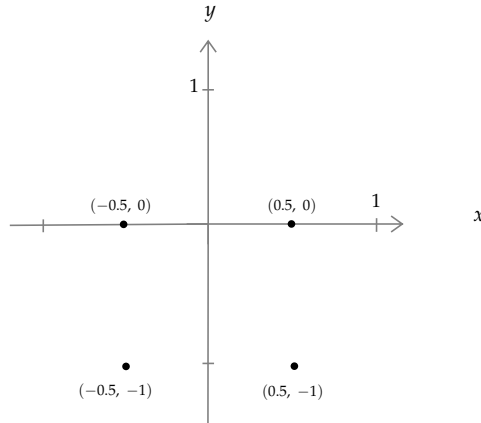


Figura 5.4.: Vértices devueltos por el método `obtener_puntos_nudo_dowker` en [7].

Como hemos justificado antes, la primera arista del diagrama del nudo será la que tiene como vértices, los vértices en la posiciones 0 y 1 de la lista de vértices devuelta, por lo que en el caso de la salida [7], la primera arista del diagrama del nudo será el conjunto $\{(-0.5, 0), (0.5, 0)\}$. La segunda arista tendrá como vértices, los vértices en las posiciones 2 y 3 de la lista de vértices devuelta, por lo que será el conjunto $\{(0.5, 0), (0.5, -1)\}$. Si realizamos este proceso con las aristas restantes y consideramos la unión de estas aristas, tendremos el conjunto representado en la Figura 5.5, que será el diagrama del nudo trivial.

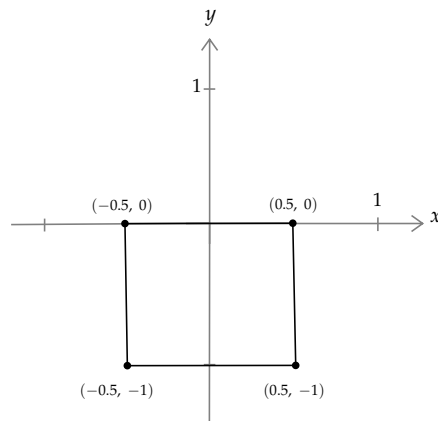


Figura 5.5.: Aristas del nudo trivial.

Una vez hemos comprendido el significado de los vértices que devuelve este método,

vamos a pasar a analizar como calcularlos. El primer vértice de todos los diagramas de nudos que construiremos será el vértice dada por $(-0.5, 0)$, vértice el cual almacenaremos en una lista que llamaremos *puntos*, donde almacenaremos todos los vértices de las aristas que compondrán el diagrama del nudo. Una vez hayamos introducido dicho vértice, comenzaremos a analizar la notación Dowker del nudo cuyo diagrama queremos construir, de la que su información se encuentra en el atributo `self.__numeros`. La notación Dowker escrita en forma de tabla del nudo que queremos construir será la representada en la [Tabla 5.3](#).

1	3	...	$2 * \text{len}(\text{self.__numeros}) - 1$
<code>self.__numeros[0]</code>	<code>self.__numeros[1]</code>	...	<code>self.__numeros[len(self.__numeros) - 1]</code>

Tabla 5.3.: Tabla asociada a la notación Dowker de un nudo arbitrario de la clase Nudo.

El primer cruce será el dado por los valores 1 y `self.__numeros[0]`. Si el número par asociado al valor 1 es positivo querrá decir que en primer lugar se recorre ese cruce por debajo y si es negativo significará que en primer lugar se recorre el cruce por arriba. Por lo tanto,

- si `self.__numeros[0] < 0` introduciremos en la lista *puntos* el vértice $(0, 0)$,
- y si `self.__numeros[0] > 0` introduciremos en la lista *puntos* el vértice $(-0.1, 0)$.

A continuación veremos a que se debe esta diferencia de criterio. Almacenaremos en unas variables locales, que llamaremos *cont_x* y *cont_y*, las coordenadas del vértice que tendremos que añadir en la lista *puntos*, justo cuando comencemos a estudiar un determinado cruce. Los valores que les asignemos a dichas variables también dependerán del signo de `self.__numeros[0] > 0`.

- Si `self.__numeros[0] < 0`, `cont_x = 0` y `cont_y = 0`, ya que queremos que el primer vértice de la siguiente arista coincida con el segundo vértice de la arista que estamos analizando, puesto que estamos recorriendo el cruce por arriba.
- Si `self.__numeros[0] > 0`, `cont_x = 0.1` y `cont_y = 0`, ya que, el primer vértice de la siguiente arista queremos que no sea el mismo que el segundo vértice de la arista que estamos analizando, es decir, queremos que haya un hueco en el diagrama, que simbolice que estamos recorriendo ese cruce por debajo.

Veamos gráficamente los dos escenarios que se pueden dar, representados en la [Figura 5.6a](#) y en la [Figura 5.6b](#), en las cuales se ha denotado por c_x y c_y a las variables *cont_x* y *cont_y* respectivamente.

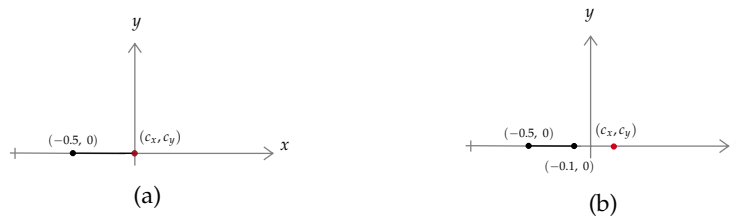


Figura 5.6.: (a) `self.__numeros[0] < 0`. (b) `self.__numeros[0] > 0`.

5. Estudio computacional de un nudo mediante su notación Dowker

Sigamos con el siguiente valor a analizar, el 2, que puede encontrarse en la lista `self.__numeros` con signo positivo o negativo. Recordemos que al haber eliminado los valores de la lista `self.__numeros` asociados a lazos en el diagrama, el valor `self.__numeros[0]` del cruce ya analizado no puede ser ni 2 ni -2 . Dado que tenemos que añadir un nuevo cruce, lo primero que haremos es añadir el vértice $(cont_x, cont_y)$ a la lista `puntos`, como habíamos comentado antes que había que hacer antes de comenzar a estudiar un cruce. Para que nos hagamos una idea, la lista `puntos` contendría hasta ahora los siguientes vértices:

- $[[-0.5, 0], [0, 0], [0, 0]]$ si `self.__numeros[0] < 0`.
- $[[-0.5, 0], [-0.1, 0], [0.1, 0]]$ si `self.__numeros[0] > 0`.

Dependiendo de si es el número 2 o el -2 el que está en la lista `self.__numeros` realizaremos lo siguiente:

- Si 2 in `self.__numeros`, incluimos en `puntos` el vértice $(cont_x + 1, cont_y)$ y `cont_x = cont_x + 1`.
- Si -2 in `self.__numeros`, incluimos en `puntos` el vértice $(cont_x + 0.9, cont_y)$ y `cont_x = cont_x + 1.1`.

Esto se hace, al igual que hicimos a la hora de añadir los vértices del primer cruce, para dejar un hueco si es que dicho cruce se recorre por debajo. Realizaremos este algoritmo hasta que tengamos que recorrer un cruce de nuevo. Vamos a ver con un ejemplo el los vértices resultantes de lo que llevamos hasta ahora.

Supongamos que tenemos la notación Dowker 6,8,10,2,4 dada por la [Tabla 5.4](#).

1	3	5	7	9
6	8	10	2	4

Tabla 5.4.: Tabla asociada a la notación Dowker 6,8,10,2,4.

Podemos observar en la [Tabla 5.4](#), que el primer cruce que se recorre dos veces es el etiquetado por 1 y 6, por lo que todos los cruces serán recorridos una vez antes de volver a recorrer dicho cruce de nuevo. Si llamamos al método `obtener_puntos_nudo_dowker`, los vértices de la lista devuelta que han sido insertados antes de analizar el valor 6 son los siguientes:

$[[-0.5, 0], [-0.1, 0], [0.1, 0], [1.1, 0], [1.1, 0], [2.0, 0], [2.2, 0], [3.2, 0], [3.2, 0], [4.1, 0], [4.3, 0]]$

Representaremos dichos vértices y las aristas asociadas a dichos vértices en la [Figura 5.7](#), para comprobar el correcto funcionamiento de la función.

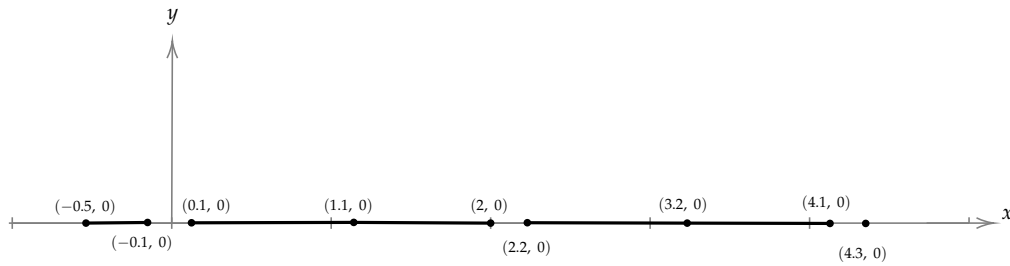


Figura 5.7.: Representación de las aristas y vértices del diagrama del nudo 6,8,10,2,4, hasta volver a recorrer el cruce etiquetado con 1 y 6.

Dado que los cruces asociados a las etiquetas 1,3 y 5 se recorren por debajo, hemos dejado huecos como se puede observar en la figura anterior. En cambio, puesto que los cruces asociados a las etiquetas 2 y 4 se recorren por arriba, no hemos dejado ese hueco, es decir hemos hecho que el vértice final una arista coincida con el inicial de la siguiente. Conclusiones y trabajos futuros

También, debemos de almacenar la coordenada x asociada a cada uno de los cruces. Para los cruces inferiores será la coordenada x del punto medio del hueco que dejamos, y para los cruces superiores será la coordenada x del vértice que introducimos repetido en la lista *puntos*, que es el segundo vértice de una arista y el primero de otra. Almacenaremos dichos valores en una lista llamada *coordenada_x_cruces*. Dicha lista será clave para poder volver a recorrer los cruces, ya que necesitamos saber la posición en la que se encuentran estos. El contenido de la lista *coordenada_x_cruces* para el ejemplo que estamos analizando será:

0, 1.1, 2.1, 3.2, 4.2.

Podemos visualizar en la [Figura 5.7](#) que dichos valores eran los que cabía esperar que fueran.

Ahora tenemos que volver a recorrer el cruce etiquetado por 1 y 6. Primero obtendremos la coordenada x asociada a dicho cruce de la lista *coordenada_x_cruces*, que en este caso será 0. A continuación, añadiremos el punto $(cont_x + 0.5, 0)$ a la lista *puntos*, y lo que haremos será mediante la inclusión de vértices construir un arco poligonal que recorra por arriba lo que llevamos de nudo y llegue al punto $(0, 0)$, ya que el cruce etiquetado por 1 y 6 se recorrerá por encima, como el que se muestra en la [Figura 5.8](#).

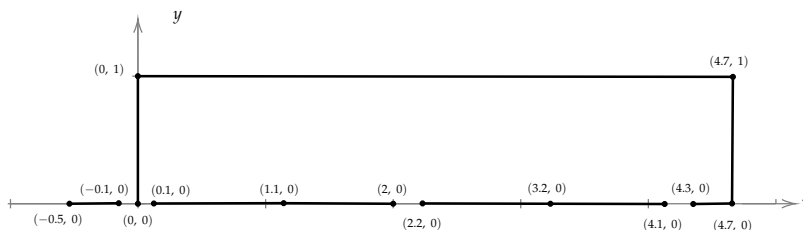


Figura 5.8.: Volver a recorrer el cruce etiquetado por 1 y 6

5. Estudio computacional de un nudo mediante su notación Dowker

Una vez que hayamos recorrido un cruce dos veces, almacenaremos en una variable booleana que llamaremos *arriba* si los siguientes vértices que añadamos estarán en la parte de arriba del diagrama del nudo (coordenada $y > 0$), en cuyo caso $arriba = True$, o en la parte de abajo del diagrama del nudo, escenario en el cual $arriba = False$. En el escenario de la [Figura 5.8](#), $arriba = False$, ya que los siguientes vértices que añadamos tendrán que tener coordenada $y \leq 0$. El siguiente valor a analizar es el 7, cuyo cruce asociado ya se encuentra dibujado, es el cruce etiquetado por 2 y 7 al que debemos llegar por debajo de lo que llevamos de nudo ya que $arriba == False$. Por tanto, tras obtener de la lista *coordenada_x_cruces* la coordenada x de dicho cruce, insertamos en la lista *puntos* los vértices necesarios para construir el arco representado en la [Figura 5.9](#).

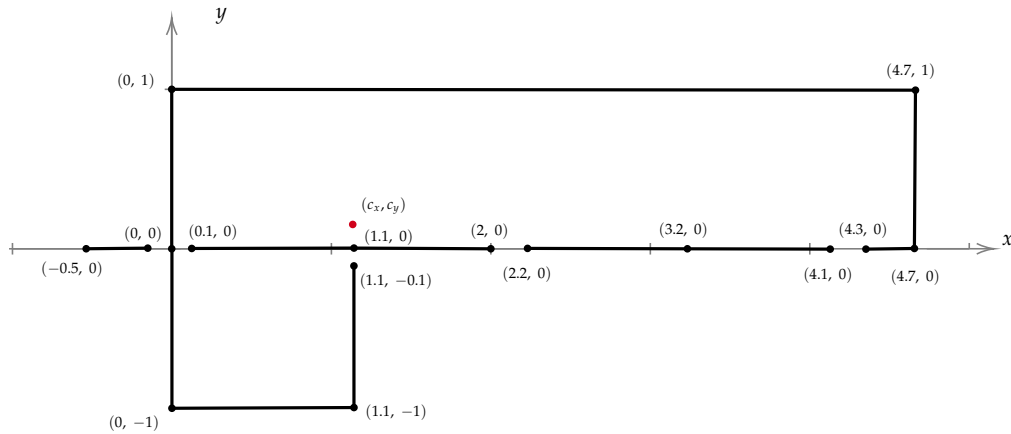


Figura 5.9.: Volver a recorrer el cruce etiquetado por 2 y 7.

Como podemos observar, dado que debemos recorrer dicho cruce por debajo, debemos dejar un hueco, por el que ya pasan las aristas que conforman la hebra superior del cruce. Tras terminar de analizar este cruce debemos cambiar el valor de la variable *arriba* a *True*.

El siguiente cruce que tenemos que analizar es el asociado al valor 8, que también está ya representado. Dado que $arriba == True$ deberemos de llegar a pasar por el cruce etiquetado con los valores 3 y 8 sin intersectar dicho arco con ninguna arista del diagrama. Por lo tanto, nos damos cuenta de que la coordenada y de dicho arco poligonal que construyamos tiene que ser menor que 1. Esto lo calcula el método ya que almacena información de todas las aristas horizontales del nudo, para evitar que se produzcan estas colisiones, siendo este uno de los principales motivos de la complejidad de este método. En consecuencia, el arco poligonal que construirá el programa será el representado en la [Figura 5.10](#).

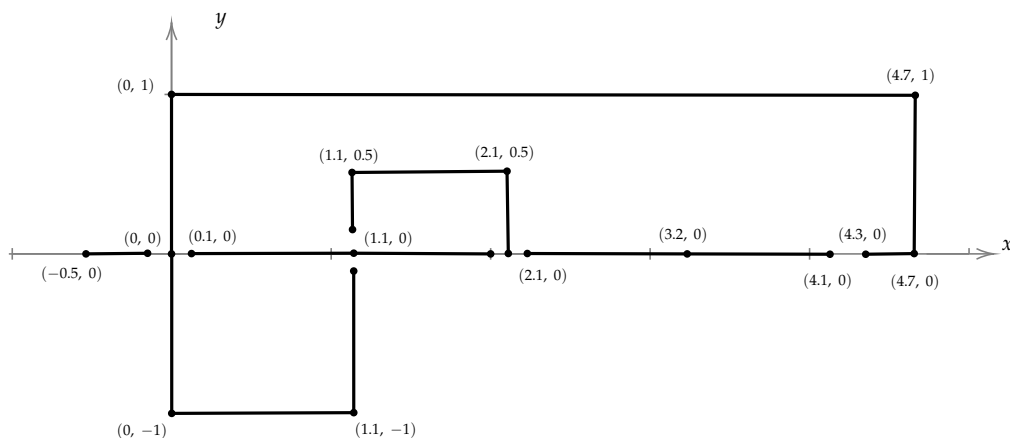


Figura 5.10.: Volver a recorrer el cruce etiquetado por 3 y 8.

Siguiendo el mismo mecanismo para los cruces restantes, una vez hayamos analizado el último cruce deberemos de “cerrar” el nudo, por lo que tendremos que conectar el último vértice que hayamos añadido al vértice inicial de la lista *puntos*, que recordemos que era el vértice $(-0.5, 0)$ y lo haremos mediante un arco. Por tanto el diagrama del nudo resultante será el representado en la [Figura 5.11](#).

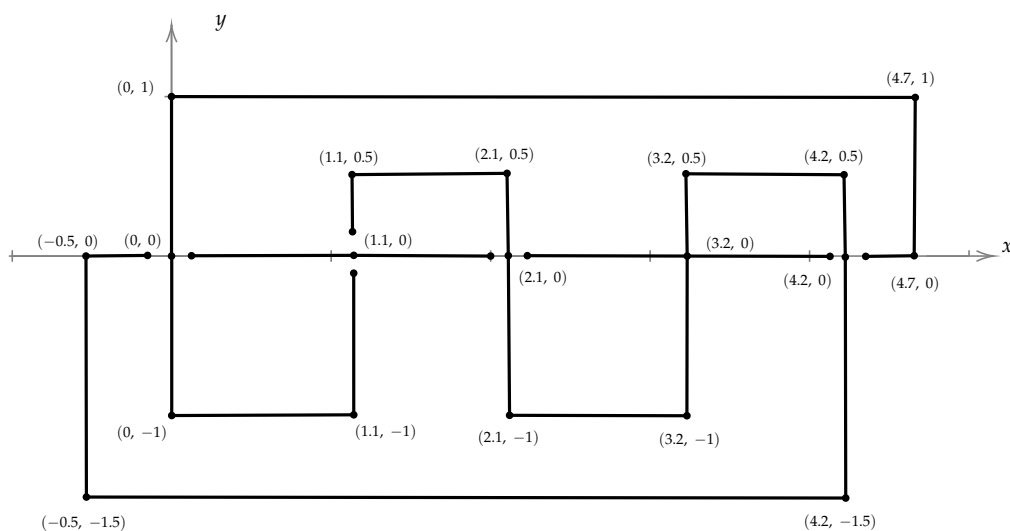


Figura 5.11.: Diagrama completo del nudo con notación Dowker 6,8,10,2,4.

Para volver desde el vértice $(4.2, 0)$ al vértice inicial, el programa ha tenido en cuenta la información de las aristas horizontales dadas por $\{(0, -1), (1.1, -1)\}$ y $\{(2.1, -1), (3.2, -1)\}$. Por esta razón la arista horizontal asociada a ese último arco tiene coordenada y igual a -1.5 .

Vamos a comprobar que la salida de la función *obtener_puntos_nudo_dowker* para el objeto de la clase Nudo con notación Dowker asociada 6,8,10,2,4, son los vértices del diagrama de la [Figura 5.11](#).

5. Estudio computacional de un nudo mediante su notación Dowker

```
In [8]: x=Nudo (6, 8, 10, 2, 4)
        x.obtener_puntos_nudo_dowker()
```

```
Out[8]: puntos: [[-0.5, 0], [-0.1, 0], [0.1, 0], [1.1, 0], [1.1, 0], [2.0, 0],
[2.2, 0], [3.2, 0], [3.2, 0], [4.1, 0], [4.3, 0], [4.7, 0], [4.7, 0],
[4.7, 1], [4.7, 1], [0, 1], [0, 1], [0, 0], [0, 0], [0, -1], [0, -1],
[1.1, -1], [1.1, -1], [1.1, -0.1], [1.1, 0.1], [1.1, 0.5], [1.1, 0.5],
[2.1, 0.5], [2.1, 0.5], [2.1, 0], [2.1, 0], [2.1, -1], [2.1, -1],
[3.2, -1], [3.2, -1], [3.2, -0.1], [3.2, 0.1], [3.2, 0.5], [3.2, 0.5],
[4.2, 0.5], [4.2, 0.5], [4.2, 0], [4.2, 0], [4.2, -1.5], [4.2, -1.5],
[-0.5, -1.5], [-0.5, -1.5], [-0.5, 0]]
```

Dado que coinciden dicho vértices, con los del diagrama de la [Figura 5.11](#), al dibujar dicho nudo utilizando el módulo *Turtle* en el método *dibujar_nudo* tendrá que dibujarse un diagrama equivalente al de la figura mencionada. Vamos a ejecutar dicho método y la salida de [9] será el diagrama de la [Figura 5.12](#).

```
In [9]: x=Nudo (6, 8, 10, 2, 4)
        puntos=x.obtener_puntos_nudo_dowker()
        dibujar_nudo(puntos)
```

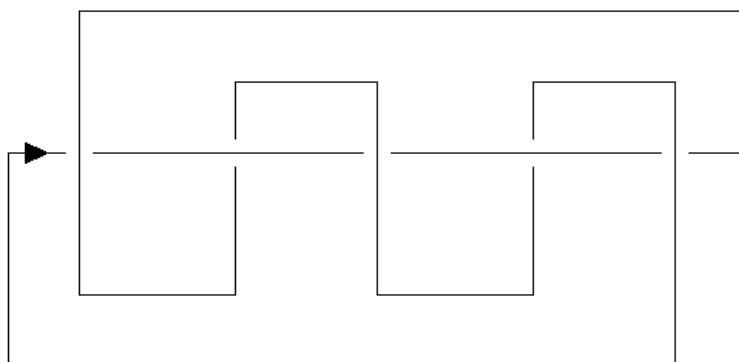


Figura 5.12.: Salida del método *dibujar_nudo* para el objeto con notación Dowker 6, 8, 10, 2, 4.

Hemos explicado la construcción de un nudo sencillo, pero vamos a ver por ejemplo la construcción del diagrama del nudo con notación Dowker 16, 18, 20, -22, 4, 2, 8, -6, 12, 10, -14. La salida del método `dibujar_nudo` para dicha notación Dowker sería la representada en la [Figura 5.13](#), donde se puede observar que hay que tener en cuenta muchísimas más variables de las que hemos mencionado en esta sección del trabajo.

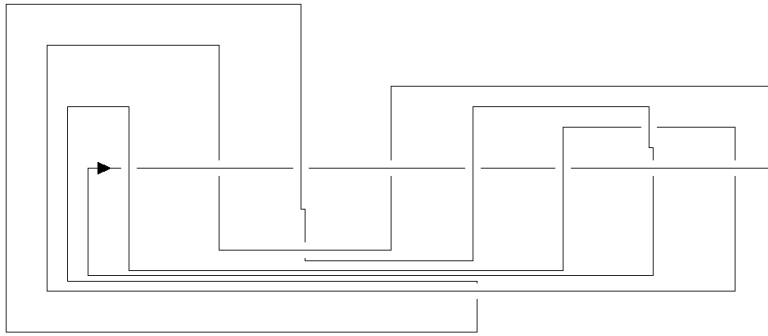


Figura 5.13.: Salida del método *dibujar_nudo* para el objeto con notación Dowker 16, 18, 20, -22, 4, 2, 8, -6, 12, 10, -14.

- `obtener_presentacion_superior(self, puntos_vi, puntos, vector_cambio):`

Este método se encuentra documentado en la ??, por lo que vamos a ver simplemente un ejemplo de su funcionamiento. Si consideramos el nudo con notación Dowker 4, 8, 10, 2, 6, la salida del método *dibujar_nudo_arcos* sería la representada en la Figura 5.14. En dicha figura podemos ver las etiquetas asignadas a cada uno de las proyecciones de los arcos superiores e inferiores.

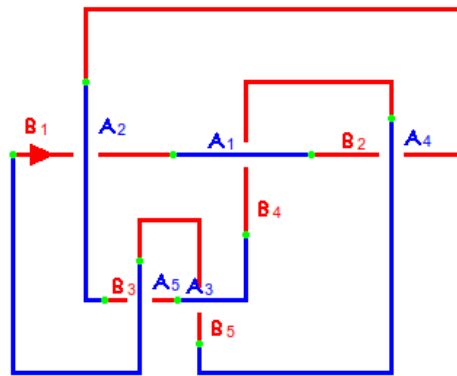


Figura 5.14.: Salida del método *dibujar_nudo_arcos* para el objeto con notación Dowker 4, 8, 10, 2, 6.

Vamos a representar también la salida del método *dibujar_caminos_vi*, que nos ayudará a comprender mejor la salida del método *obtener_presentacion_superior*. La salida de *dibujar_caminos_vi* está representada en la Figura 5.15.

5. Estudio computacional de un nudo mediante su notación Dowker

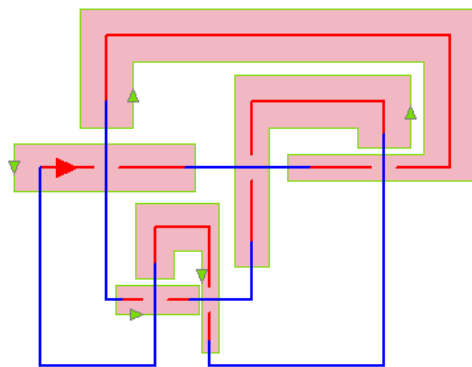


Figura 5.15.: Salida del método *dibujar_caminos_vi* para el objeto con notación Dowker 4, 8, 10, 2, 6.

En la figura anterior, se han representado con color rojo los espacios B_i , $i = 1, \dots, 5$, y con color verde las curvas de Jordan recorridas por los caminos v_i , $i = 1, \dots, 5$, indicando una flecha la orientación antihoraria con la que dichos caminos recorren su curva de Jordan asociada. Ahora que tenemos la Figura 5.14 y la Figura 5.15 representadas, podremos entender mejor la salida del método *obtener_presentacion_superior*. Vamos a verla:

```
In [10]: x=Nudo (4, 8, 10, 2, 6)
puntos=x.obtener_puntos_nudo_dowker()
vector_cambio=x.dividir_nudo_en_arcos(puntos)
aristas=x.obtener_aristas(puntos)
puntos_vi=x.obtener_caminos_vi (puntos, aristas, vector_cambio)
x.obtener_presentacion_superior(puntos_vi, puntos, vector_cambio)

Out[10]: Presentación superior del grupo del nudo:
[[5, -2, -1, 2], [4, 1, -4, -2], [2, -5, -3, 5],
[1, 3, -1, -4], [3, 4, -3, -5]]
```

Por ejemplo la primer lista de las devueltas por el método significa que el camino v_1 se intersecta en primer lugar con el arco A_5 con signo positivo, en segundo lugar con el arco A_2 con signo negativo, en tercer lugar con el arco A_1 con signo negativo y en último lugar con el arco A_2 con signo positivo. Podemos comprobarlo fijándonos en la Figura 5.14 y la Figura 5.15.

■ *obtener_presentacion_inferior*(self, puntos_ui, puntos, vector_cambio):

Este método se encuentra documentado en la ??, por lo que vamos a ver simplemente un ejemplo de su funcionamiento. Consideremos el nudo de la Figura 5.14, es decir, el nudo con notación Dowker 4, 8, 10, 2, 6. La salida de *dibujar_caminos_ui* está representada en la Figura 5.16.

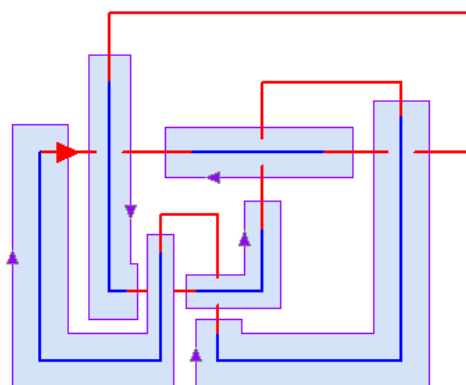


Figura 5.16.: Salida del método *dibujar_caminos_ui* para el objeto con notación Dowker 4, 8, 10, 2, 6.

Ahora que tenemos la **Figura 5.14** y la **Figura 5.16** representadas, podremos entender mejor la salida del método `obtener_presentacion_inferior`. Vamos a verla:

```
In [11]: x=Nudo (4, 8, 10, 2, 6)
puntos=x.obtener_puntos_nudo_dowker()
vector_cambio=x.dividir_nudo_en_arcos(puntos)
aristas=x.obtener_aristas(puntos)
puntos_ui=x.obtener_caminos_ui (puntos, aristas, vector_cambio)
x.obtener_presentacion_inferior(puntos_ui, puntos, vector_cambio)
```

```
Out[11]: Presentación inferior del grupo del nudo:
[[ -1,  4,  2, -4], [ 1,  3, -1, -2], [ 5,  4, -5, -3],
 [ 2,  5, -2, -4], [ 3,  1, -3, -5]]
```

Por ejemplo la primer lista de las devueltas por el método significa que el camino u_1 se intersecta en primer lugar con el arco B_1 con signo negativo, en segundo lugar con el arco B_4 con signo positivo, en tercer lugar con el arco B_2 con signo positivo y en último lugar con el arco B_4 con signo negativo. Podemos comprobarlo fijándonos en la [Figura 5.14](#) y la [Figura 5.16](#).

5.3. Conclusiones y trabajos futuros

Tras el estudio matemático e informático realizado acerca del grupo de un nudo admisible, hemos podido comprobar que, computacionalmente, es posible calcular la presentación superior e inferior del grupo de un nudo a través de su notación Dowker, además de poder visualizar el diagrama del nudo asociado a dicha notación Dowker. El poder obtener computacionalmente la presentación superior e inferior del grupo de un nudo es bastante útil, ya que como vimos en la parte matemática del trabajo, no es trivial el cálculo de dichas presentaciones.

En un futuro, sería interesante probar el funcionamiento del método `obtener_puntos_nudo_dowker` de la clase `Nudo` para nudos primos con 10 o más cruces, ya que aunque el método

5. Estudio computacional de un nudo mediante su notación Dowker

funcione perfectamente para todos los nudos primos de 9 o menos cruces, podría dar algún fallo para nudos con más cruces, debido a que para cada notación Dowker se pueden dar escenarios muy diversos, y puede darse el caso de que falte algún caso por contemplar. Es por este motivo que el método *obtener_puntos_nudo_dowker* esté compuesto por más de 3.000 líneas de código.

Otra interesante línea de investigación para próximos años sería comparar el potencial del grupo de un nudo con el de otros invariantes de nudos, para ver cuál de los invariantes existentes tiene un éxito mayor en la complicada tarea de distinguir nudos no equivalentes. Este estudio sería posible gracias a que la clase *Nudo*, construída en este trabajo, nos permite obtener la presentación superior e inferior del grupo de un nudo en un corto periodo de tiempo, por lo que podríamos hacer un estudio de su potencial tomando una gran cantidad de nudos como entrada.

A. Documentación de la clase Nudo

Tras el estudio matemático e informático realizado acerca del grupo de un nudo admisible, hemos podido comprobar que, computacionalmente, es posible calcular la presentación superior e inferior del grupo de un nudo a través de su notación Dowker, además de poder visualizar el diagrama del nudo asociado a dicha notación Dowker. El poder obtener computacionalmente la presentación superior e inferior del grupo de un nudo es bastante útil, ya que como vimos en la parte matemática del trabajo, no es trivial el cálculo de dichas presentaciones.

En un futuro, sería interesante probar el funcionamiento del método *obtener_puntos_nudo_dowker* de la clase Nudo para nudos primos con 10 o más cruces, ya que aunque el método funcione perfectamente para todos los nudos primos de 9 o menos cruces, podría dar algún fallo para nudos con más cruces, debido a que para cada notación Dowker se pueden dar escenarios muy diversos, y puede darse el caso de que falte algún caso por contemplar. Es por este motivo que el método *obtener_puntos_nudo_dowker* esté compuesto por más de 3.000 líneas de código.

Otra interesante línea de investigación para próximos años sería comparar el potencial del grupo de un nudo con el de otros invariantes de nudos, para ver cuál de los invariantes existentes tiene un éxito mayor en la complicada tarea de distinguir nudos no equivalentes. Este estudio sería posible gracias a que la clase Nudo, construida en este trabajo, nos permite obtener la presentación superior e inferior del grupo de un nudo en un corto periodo de tiempo, por lo que podríamos hacer un estudio de su potencial tomando una gran cantidad de nudos como entrada.

En esta sección documentaremos los métodos principales de la clase *Nudo* de *Python*, implementados en el archivo *claseNudo.py*, archivo en el cual también se encontrarán documentados dichos métodos. Para algunos de los métodos de la clase Nudo es difícil describir su funcionamiento debido a que son bastante geométricos o por su gran extensión. Por ello, se ha acompañado esta documentación de la clase con un tutorial del uso de esta, ubicado en la [Sección 5.2](#), que se recomienda ser consultado en caso de no comprender del todo alguno de los métodos que describiremos a continuación.

Constructor Constructor de la clase Nudo que recibe como parámetro la notación Dowker de un nudo. Dentro del constructor se comprueba que la notación introducida es una notación Dowker válida, y elimina los números pares de la notación Dowker recibida que estén asociados a lazos del nudo.

```
def __init__(self, *numeros):  
    """  
        Inicializa el atributo self.__numeros con la lista que recibe  
        por parámetro, en caso de que dicha lista tenga el formato de  
        una notación Dowker.
```

A. Documentación de la clase Nudo

Parámetros :

**numeros: list*
Notación Dowker de un nudo.

Métodos llamados :

--eliminar_lazos
Para eliminar de self.__numeros los valores pares asociados a lazos.

--comprobar_notacion_correcta
Para comprobar que la lista introducida por parámetro está asociada a una notación Dowker correcta. En caso de que no sea así se lanza un error.

"""

Eliminar lazos Método utilizado para eliminar los números pares de la lista self.__numeros que estén asociados a lazos del nudo (si es que los hay). En caso de que eliminemos alguno de los elementos de dicha lista, puede que tengamos que modificar otros de los elementos de esa lista, para que la lista resultante esté asociada a una notación Dowker válida.

```
def __eliminar_lazos(self):
```

"""

Recorre la lista self.__numeros y elimina, si es que existen, los números pares de dicha lista asociados a lazos. Dado que si eliminamos un número par de dicha lista puede que no resulten ser números pares consecutivos en valor absoluto, tenemos que restar (si son positivos) o sumar (si son negativos) 2 a dichos números pares que sean mayores en valor absoluto que el valor absoluto del número eliminado.

Es llamado por:

--init--
El método __eliminar_lazos solo se llama cuando instanciamos un objeto de la clase Nudo.

"""

Comprobar formato de notación Dowker Método utilizado para comprobar si la lista self.__numeros tiene el formato de la notación Dowker de un nudo.

```
def __comprobar_notacion_correcta(self)::
```

"""

Recorre la lista self.__numeros y comprueba si contiene en valor absoluto los n primeros números pares, siendo n la longitud de

dicha lista.

Es llamado por:

__init__

Este método únicamente se llama cuando instanciamos un objeto de la clase Nudo.

Devuelve:

True si tiene el formato de notación Dowker y False en caso contrario.

"""

Número de arcos superiores del nudo Método utilizado para calcular el número de arcos superiores del nudo asociado a la notación Dowker dada por la lista self.__numeros.

def numero_arcos_superiores(self):

"""

Calcula el número de arcos superiores del nudo asociado a la lista self.__numeros, recorriendo esta última.

Es llamado por:

dibujar_nudo_arcos

Se llamará a este método cuando dibujemos con Turtle el diagrama del nudo dividido en arcos superiores e inferiores.

__representar_generadores_presentacion_superiores

Ya que una presentación superior tendrá tantos generadores como número de arcos superiores tenga el nudo.

__representar_generadores_presentacion_inferior

Por el mismo motivo que el método anterior.

Devuelve:

Número de arcos superiores del nudo.

"""

Dividir notación Dowker en dos subpermutaciones Comprueba si la permutación dada por la lista self.__numeros, puede ser dividida en dos subpermutaciones de números pares, en cuyo caso el nudo asociado será compuesto.

__dividir_en_dos_subpermutaciones(self):

"""

A. Documentación de la clase Nudo

Recorre la lista `self.__numeros` para comprobar si, en valor absoluto, se puede escribir como concatenación de dos subpermutaciones de números pares.

Es llamado por:

obtener_puntos_nudo_dowker

Ya que el cálculo de los vértices de las aristas que conforman el diagrama de un nudo compuesto se realizará de distinta forma al de un nudo simple.

Devuelve:

Posición de la lista a partir de la cual se puede dividir en dos subpermutaciones, y -1 en caso contrario.

"""

Obtener vértices de las aristas del diagrama del nudo poligonal Método para calcular las coordenadas x e y de los vértices de las aristas horizontales y verticales que conformarán el diagrama del nudo poligonal asociado a la notación Dowker dada por la lista `self.__numeros`. Es el método más sofisticado y con más líneas de código de la clase, ya que hay que tener bastantes criterios en cuenta para poder obtener dichos vértices de las aristas sin que ningún par de las aristas resultantes se intersecten entre sí. En consecuencia, se tratará de mostrar el funcionamiento de este método en el tutorial que describiremos en la [Sección 5.2](#), ya que mediante los dibujos del diagrama del nudo que realizaremos con el módulo *Turtle* será más sencillo de comprender este método.

def obtener_puntos_nudo_dowker(self):

"""

Calcula la lista de vertices de las aristas que conformarán el diagrama del nudo asociado a la notación Dowker dada por `self.__numeros`. Los vértices en posiciones pares de la lista serán los puntos iniciales de una arista y los impares los puntos finales. En consecuencia, las aristas estarán formadas por los vértices en las posiciones $\{2i, 2i+1\}$.

Devuelve:

Lista de los vértices de las aristas del diagrama del nudo, siendo un vértice un par tal que su primer elemento es la coordenada x del vértice y el segundo es la coordenada y de este.

"""

Dividir nudo en arcos superiores e inferiores Una vez hayamos calculado la lista de vértices de las aristas del diagrama del nudo mediante la llamada al método *obtener_puntos_nudo_dowker*, dividiremos las aristas asociadas a dichos vértices en aristas superiores e inferiores. Para ello, recorreremos la lista de vértices y detectaremos cuando se pasa por un cruce superior y por uno inferior, para que cuando se produzcan dichos cambios de pasar de un cruce superior a uno inferior o viceversa, almacenar en una lista las posiciones de la lista de vértices en las que se producen dichos cambios. También existen casos en los que tenemos que añadir vértices a la lista que se recibe por parámetro; ya que es necesario crear nuevas aristas para que sea posible dividir el diagrama del nudo en aristas asociadas a arcos superiores del nudo y aristas asociadas a arcos inferiores de este. Para entender correctamente este método, también es más conveniente visualizar el diagrama del nudo gráficamente, como haremos en el tutorial de la siguiente sección.

```
def dividir_nudo_en_arcos(self , puntos):
    """
        Devuelve las posiciones de la lista 'puntos' que recibe por
        parámetro en los que se produce un cambio de un arco superior
        a uno inferior o viceversa .

        Parámetros
        

---


        puntos: list of lists
            Salida del método obtener_puntos_nudo_dowker

        Devuelve :
        

---


        Lista de las posiciones de la lista 'puntos' (que puede
        haber sido modificada), en los que se producen cambios
        de un arco superior a uno inferior o viceversa .

    """
```

Dibujar diagrama de un nudo Este método utiliza el módulo *Turtle* de *Python* para dibujar el diagrama del nudo asociado a la notación Dowker dada por la lista *self.__numeros*. Simplemente dibuja las aristas dadas por la lista de vértices que recibe dicho método por parámetro, y señala con una flecha la orientación de este.

```
def dibujar_nudo(self , puntos):
    """
        Dibuja el diagrama de un nudo poligonal representando las
        aristas dadas por los vértices de la lista que recibe
        por parámetro .

        Parámetros :
        

---


        puntos: list of lists
            Salida del método obtener_puntos_nudo_dowker ,
            modificada por el método dividir_nudo_en_arcos .

    """
```

"""

Dibujar diagrama de un nudo dividido en arcos superiores e inferiores Este método también utiliza el módulo *Turtle* de *Python* para dibujar el diagrama del nudo asociado a la notación Dowker dada por la lista `self.__numeros`, utilizando la lista de vértices que recibe por parámetro. Sin embargo, en este caso dibujará el diagrama del nudo poligonal dividido en arcos superiores, representados con color azul, e inferiores, representados con color rojo. También etiquetaremos cada uno de esos arcos superiores e inferiores, que como vimos en el capítulo 3, se etiquetará cada arco superior con A_1, \dots, A_n , y cada arco inferior con B_1, \dots, B_n , siendo n el número de arcos superiores del nudo. También se representarán los puntos del conjunto Q con color verde, e indicaremos con una flecha la orientación del nudo.

def dibujar_nudo_arcos(`self` , `puntos` , `numeros_cambio`):

"""

Dibuja el diagrama de un nudo poligonal dividido en arcos superiores e inferiores , utilizando los vértices de la aristas que recibe por parámetro .

Parámetros :

puntos: list of lists

*Salida del método obtener_puntos_nudo_dowker ,
modificada por el método dividir_nudo_en_arcos .*

numeros_cambio: list

Salida del método dividir_nudo_en_arcos .

"""

Obtener aristas del diagrama del nudo Este método utiliza la lista de vértices que recibe por parámetro; para almacenar la información relativa a las aristas verticales y horizontales del diagrama del nudo. El formato en el que se almacenarán las aristas horizontales y verticales será:

- Cada arista horizontal la representaremos una lista de 3 números de tipo float $[y, x_1, x_2]$ donde el primero de ellos será la coordenada y de la arista horizontal, el segundo elemento será la coordenada x del vértice de la arista con menor valor y el tercero será la coordenada x del vértice de la arista con mayor valor.
- Cada arista vertical la representaremos mediante una lista de 3 números de tipo float $[x, y_1, y_2]$ donde el primero de ellos será la coordenada x de la arista vertical, el segundo elemento será la coordenada y del vértice de la arista con menor valor, y el tercero será la coordenada y del vértice de la arista con mayor valor.

Todas las aristas horizontales se almacenarán en una lista y las aristas verticales en otra, y dicho método devolverá las aristas como una lista de dos elementos, siendo el primero de ellos la lista de las aristas horizontales, y el segundo de ellos la lista de las aristas verticales.

Dichas aristas serán utilizadas por los métodos *obtener_caminos_ui* y *obtener_caminos_vi*, para que al calcular dichos caminos u_i, v_j tal que $i, j = 1, \dots, n$, se cumpla que los espacios asociados U_1, \dots, U_n sean disjuntos, al igual que los espacios V_1, \dots, V_n , como afirmamos en la **Sección 3.1** que debían ser.

```
def obtener_aristas(self , puntos):
```

```
    """
```

```
        Recorre la lista de vértices que recibe por parámetro
        para obtener las aristas horizontales y verticales del
        diagrama del nudo asociado a dicha lista de vértices .
```

```
        Parámetros :
```

```
        puntos: list of lists
```

```
        Salida del método obtener_puntos_nudo_dowker ,
        modificada por el método dividir_nudo_en_arcos .
```

```
        Devuelve :
```

```
        Lista compuesta de dos listas , conteniendo la primera
        de ellas las aristas horizontales del diagrama del
        nudo , y la segunda de ellas las aristas verticales del
        diagrama del nudo .
```

```
    """
```

Obtener aristas inferiores del diagrama del nudo Este método utiliza el mismo formato que el método *obtener_aristas* para almacenar las aristas horizontales y verticales. Sin embargo, este método solo devuelve aquellas aristas asociadas a un arco inferior del nudo, información que se obtendrá de que este método recibe como parámetro la lista de índices que devuelve el método *dividir_nudo_en_arcos*.

Dichas aristas inferiores serán utilizadas por el método *obtener_caminos_vi*, para que al calcular los caminos v_i , $i = 1, \dots, n$, es decir, los caminos que rodean a las proyecciones de los arcos inferiores, se cumpla que los espacios asociados V_1, \dots, V_n sean disjuntos.

```
def __obtener_aristas_inferiores(self , puntos , vector_cambio):
```

```
    """
```

```
        Recorre la lista de vértices que recibe por parámetro ,
        y teniendo en cuenta la lista 'vector_cambio' , obtiene
        las aristas horizontales y verticales que pertenezcan
        a los arcos inferiores del diagrama del nudo asociado
        a dicha lista de vértices .
```

```
        Parámetros :
```

```
        puntos: list of lists
```

```
        Salida del método obtener_puntos_nudo_dowker ,
```

A. Documentación de la clase Nudo

modificada por el método dividir_nudo_en_arcos.

vector_cambio: list

Salida del método dividir_nudo_en_arcos.

Devuelve:

Lista compuesta de dos listas, conteniendo la primera de ellas las aristas horizontales asociadas a los arcos inferiores del diagrama del nudo, y la segunda de ellas las aristas verticales del diagrama del nudo asociadas a los arcos inferiores de este.

"""

Obtener aristas superiores del diagrama del nudo Este método utiliza el mismo formato que el método *obtener_aristas* para almacenar las aristas horizontales y verticales. Sin embargo, este método solo devuelve aquellas aristas asociadas a un arco superior del nudo, información que se obtendrá de que este método recibe como parámetro la lista de índices que devuelve el método *dividir_nudo_en_arcos*

Dichas aristas superiores serán utilizadas por el método *obtener_caminos_ui*, para que al calcular los caminos u_i tal que $i = 1, \dots, n$, es decir, los caminos que rodean a las proyecciones de los arcos superiores, se cumpla que los espacios asociados U_1, \dots, U_n sean disjuntos.

def __obtener_aristas_superiores(self, puntos, vector_cambio):
"""

Recorre la lista de vértices que recibe por parámetro, y teniendo en cuenta la lista 'vector_cambio', obtiene las aristas horizontales y verticales que pertenezcan a los arcos superiores del diagrama del nudo asociado a dicha lista de vértices.

Parámetros:

puntos: list of lists

Salida del método obtener_puntos_nudo_dowker, modificada por el método dividir_nudo_en_arcos.

vector_cambio: list

Salida del método dividir_nudo_en_arcos.

Devuelve:

Lista compuesta de dos listas, conteniendo la primera de ellas las aristas horizontales asociadas a los arcos superiores del diagrama del nudo, y la segunda de ellas las aristas verticales del diagrama

del nudo asociadas a los arcos superiores de este.

"""

Obtener aristas inferiores orientadas del diagrama del nudo Este método almacena las aristas horizontales y verticales en un formato diferente al que las habíamos almacenado en los métodos anteriores. Las aristas horizontales y verticales se almacenarán siguiendo el siguiente formato:

- Cada arista horizontal la representaremos mediante una lista de 3 números $[y, x_1, x_2]$, donde el primero de ellos será la coordenada y de la arista horizontal, el segundo elemento será la coordenada x del primer vértice de la arista que ha sido recorrido (es decir, el que tiene un índice menor en la lista de vértices), y el tercero será la coordenada x del segundo vértice de la arista que ha sido recorrido.
- Cada arista vertical la representaremos mediante una lista de 3 números $[x, y_1, y_2]$, donde el primero de ellos será la coordenada x de la arista vertical, el segundo elemento será la coordenada y del primer vértice de la arista que ha sido recorrido (es decir, el que tiene un índice menor en la lista de vértices), y el tercero será la coordenada y del segundo vértice de la arista que ha sido recorrido.

Sin embargo, este método solo devuelve aquellas aristas asociadas a los arcos inferiores del nudo, información que se obtendrá de que este método recibe como parámetro la lista de índices que devuelve el método *dividir_nudo_en_arcos*. Además, en vez de almacenarse todas las aristas horizontales en una lista, y todas las verticales en otra, almacenaremos las aristas horizontales [verticales] asociadas los arcos inferiores B_i , $i = 1, \dots, n$ en listas distintas, las cuales introduciremos posteriormente en la lista de aristas horizontales [verticales].

Realizaremos la división de las aristas horizontales y verticales en listas asociadas a los arcos inferiores B_i , $i = 1, \dots, n$, ya que dichas aristas serán utilizadas por el método *obtener_presentacion_inferior* para calcular la presentación inferior del grupo del nudo. Para calcular dicha presentación, tenemos que saber la orientación de las aristas que formarán parte de los arcos inferiores del nudo, y el arco inferior asociado a cada arista orientada, ya que recordemos que para calcular los relatores de dicha presentación, debíamos saber el signo de la intersección de los caminos u_i , $i = 1, \dots, n$ con las aristas asociadas a los arcos inferiores del nudo, como vimos en la [Sección 3.1](#).

```
def __obtener_aristas_inferiores_orientadas(self, puntos, vector_cambio):
```

```
    """
```

```
        Recorre la lista de vértices que recibe por parámetro,
        y teniendo en cuenta la lista 'vector_cambio', obtiene
        las aristas horizontales y verticales orientadas que
        pertenezcan a los arcos inferiores del nudo obtenido
        a partir de dicha lista de vértices.
```

```
    Parámetros:
```

```
    puntos: list of lists
```

```
    Salida del método obtener_puntos_nudo_dowker,
```

A. Documentación de la clase Nudo

modificada por el método dividir_nudo_en_arcos.

vector_cambio: list

Salida del método dividir_nudo_en_arcos.

Devuelve:

Lista compuesta de dos listas, conteniendo la primera de ellas las listas de aristas horizontales orientadas asociadas a los arcos inferiores B_i del diagrama del nudo, y la segunda de ellas las listas de aristas verticales orientadas asociadas a los arcos inferiores B_i del diagrama del nudo.

"""

Obtener aristas superiores orientadas del diagrama del nudo Este método almacena las aristas horizontales y verticales en el mismo formato que el método `__obtener_aristas_inferiores_orientadas`

Este método devolverá aquellas aristas asociadas a los arcos superiores del nudo, información que se obtendrá de que este método recibe como parámetro la lista de índices que devuelve el método `dividir_nudo_en_arcos`, y además, en vez de almacenarse todas las aristas horizontales en una lista, y todas las verticales en otra, almacenaremos las aristas horizontales [verticales] asociadas los arcos superiores A_i , $i = 1, \dots, n$ en listas distintas, las cuales introduciremos posteriormente en la lista de aristas horizontales [verticales].

Dichas aristas superiores orientadas, serán utilizadas por el método `obtener_presentacion_superior` cuando tengamos que calcular la presentación superior del grupo del nudo, ya que para calcularla, tenemos que saber la orientación de las aristas que formarán parte de los arcos superiores del nudo, puesto que recordemos que para calcular los relatores de dicha presentación, debíamos saber el signo de la intersección de los caminos v_i , $i = 1, \dots, n$ con las aristas asociadas a los arcos superiores del nudo, además de saber la proyección del arco superior que se intersecta con los caminos v_i .

```
def __obtener_aristas_superiores_orientadas(self, puntos, vector_cambio):
```

```
    """
```

Recorre la lista de vértices que recibe por parámetro, y teniendo en cuenta la lista vector_cambio, obtiene las aristas horizontales y verticales orientadas que pertenezcan a los arcos superiores del nudo obtenido a partir de dicha lista de vértices.

Parámetros:

puntos: list of lists

Salida del método obtener_puntos_nudo_dowker, modificada por el método dividir_nudo_en_arcos.

vector_cambio: list
Salida del método dividir_nudo_en_arcos.

Devuelve:

Lista compuesta de dos listas, conteniendo la primera de ellas las listas de aristas horizontales orientadas asociadas a los arcos superiores A_i del diagrama del nudo, y la segunda de ellas las listas de aristas verticales orientadas asociadas a los arcos superiores A_i del diagrama del nudo.

"""

Obtener bordes y giros de los caminos v_i y u_i Este método será uno de los métodos más importantes y extensos de la clase Nudo, que será clave para el cálculo de los caminos $u_i [v_i]$, $i = 1, \dots, n$, a través de los cuales, seremos capaces de calcular la presentación inferior [superior] del grupo del nudo. Vamos a explicar geométricamente el funcionamiento de este método.

Dada la proyección de un arco superior $\mathcal{P}A_i$ [inferior $\mathcal{P}B_i$], $i = 1, \dots, n$, cuya información recibe este método por parámetro, vamos a introducir a cada una de las aristas pertenecientes a la proyección de dicho arco superior [inferior] en un rectángulo. Para cada rectángulo, su arista asociada va a estar contenida en el interior de ese rectángulo, a una distancia d_1 del borde izquierdo del rectángulo, a una distancia d_2 del borde derecho del rectángulo, etc. A dichas distancias las llamaremos *borde_izquierda*, *borde_derecha*, *borde_arriba* y *borde_abajo*. Dicho método calculará, para cada uno de las aristas superiores [inferiores] de la proyección del arco superior [inferior], las distancias que deberá haber entre dichas aristas y sus rectángulos asociados, para que podamos considerar el espacio $U_i [V_i]$ como la unión de dichos rectángulos, y por lo tanto, se cumpla que los espacios $U_i [V_i]$, $i = 1, \dots, n$ sean espacios disjuntos.

Para conseguir nuestro cometido, empezaremos por la arista de la proyección $\mathcal{P}A_i [\mathcal{P}B_i]$ que sea recorrida en primer lugar, recorriendo el diagrama del nudo empezando por el punto de partida y siguiendo la orientación de este (señalados con una flecha), y dependiendo de si la arista es vertical u horizontal, y del sentido en el que la recorramos, podremos hallar tres de los cuatro bordes de su rectángulo asociado. Introduciremos dichos bordes en una lista de 4 elementos en el orden [*borde_izquierda*, *borde_derecha*, *borde_arriba*, *borde_abajo*], introduciendo un 0 en el borde que no hayamos podido calcular todavía. La siguiente arista recorrida, tendrá que tener verticalidad opuesta a la ya recorrida, por lo que si recorreremos dicha proyección del nudo siguiendo la orientación de este, habrá que realizar un giro. Por lo tanto, dada la proyección de un arco superior [inferior] consistente de n aristas, tendremos que realizar $n - 1$ giros. Queremos almacenar información acerca de los giros que realizamos al recorrer la proyección $\mathcal{P}A_i [\mathcal{P}B_i]$, por lo que consideraremos 4 tipo de giros, los pertenecientes al conjunto {*ab_i*, *ab_d*, *ar_i*, *ar_d*}, que se almacenarán en una lista que llamaremos **giros**, y que detallaremos posteriormente cuando se asigna un giro del conjunto u otro.

Pongamos un ejemplo. Supongamos que la primera arista de la proyección de un arco superior $\mathcal{P}A_i$, $i = 1, \dots, n$ es una arista horizontal y la recorremos de izquierda a derecha; por lo tanto, el vértice derecho de dicha arista tiene que ser el vértice de una arista vertical también de la proyección $\mathcal{P}A_i$, suponiendo que la proyección del arco superior $\mathcal{P}A_i$ esté compuesta de más de una arista. Para dicha arista vertical hay dos opciones, que se recorra en sentido vertical creciente o vertical decreciente. Al rectángulo asociado a la primera lista le podemos asignar `borde_izquierda`, `borde_arriba` y `borde_abajo`, ya que estos, no van a tener que coincidir con ningún borde asociado a rectángulos del resto de aristas de la proyección. Por lo tanto, introducimos en una lista llamada **bordes** la lista [`borde_izquierda`, 0, `borde_arriba`, `borde_abajo`]. Dependiendo de si la arista siguiente es una arista vertical que se recorre en sentido vertical creciente o vertical decreciente, tendremos dos opciones de giro:

- `ab_d`, que significa ‘abajo derecha’, e indicará que la siguiente arista será una arista vertical que se recorrerá en sentido vertical decreciente, y que ha sido añadida a la derecha de la arista horizontal previa.
- `ar_d`, que significa ‘arriba derecha’, e indicará que la siguiente arista será una arista vertical que se recorrerá en sentido vertical creciente, y que ha sido añadida a la derecha de la arista horizontal previa.

Si nos fijamos, en ambos casos podremos calcular el valor `borde_derecho` asociado a dicha arista vertical, que tendrá que coincidir con el valor `borde_derecho` de la primera arista horizontal, por lo que cuando lo calculemos, sustituiremos el valor 0 de la todavía única lista de la lista `bordes` por el valor `borde_derecho`, calculado ahora. Además, de dicha arista vertical obtendremos que:

- En caso de ser la arista vertical una arista que se recorra en sentido vertical decreciente, el valor `borde_arriba` asociado a dicha arista vertical, será el mismo que el valor `borde_arriba` de la arista horizontal previa, por lo que no hará falta calcularlo.
- En caso de que la arista vertical sea recorrida en sentido vertical creciente, será el valor `borde_abajo` de dicha arista vertical el que induciremos del valor `borde_abajo` de la arista horizontal previa.

Realizaremos este proceso hasta que hayamos recorrido todas las aristas asociadas a la proyección $\mathcal{P}A_i$.

Los valores `borde_izquierda`, `borde_derecha`, `borde_arriba` y `borde_abajo` asociados a cada rectángulo, serán calculados mediante una serie de funciones que se encuentran documentadas en el archivo *claseNudo.py*.

Los datos devueltos por este método serán utilizados por las funciones *obtener_caminos_ui* y *obtener_caminos_vi* para pasárselos como parámetro al método *__obtener_puntos_vi_ui*, que como el nombre de dicho método indica, calculará los vértices de los caminos simples u_i y v_i , para $i = 1, \dots, n$.

```
def __obtener_bordes_y_giros_vi_ui(self, puntos, aristas,
    valor_inicial, valor_final, aristas_inferiores_o_superiores):
```

"""

Recorre las aristas asociadas a la proyección de un arco superior [inferior] y calcula los bordes y giros asociados a dicha proyección.

Parámetros:

puntos: list of lists

Salida del método obtener_puntos_nudo_dowker, modificada por el método dividir_nudo_en_arcos.

aristas:

Salida del método obtener_aristas.

valor_inicial:

Indice de la lista 'puntos' del primer vértice de la primera arista de la proyección del arco superior [inferior].

valor_final:

Indice de la lista 'puntos' del segundo vértice de la última arista de la proyección del arco superior [inferior].

aristas_inferiores_o_inferiores:

Caso camino u_i : Salida de __obtener_aristas_superiores.

Caso camino v_i : Salida de __obtener_aristas_inferiores.

Devuelve:

Lista de dos elementos, siendo el primero de ellos la lista de bordes y el segundo la lista de giros, siendo la primera lista de tamaño n y la segunda de tamaño $n-1$.

"""

Obtener vértices de un camino v_i o u_i Este método toma como entrada la lista de bordes y giros asociados a la proyección de un arco superior $\mathcal{P}A_i$ [inferior $\mathcal{P}B_i$] y calcula los vértices de los caminos u_i [v_i] que recorrerán los bordes de los espacios U_i [V_i], que recordemos que son la unión de los rectángulos cerrados determinados por las aristas asociadas a $\mathcal{P}A_i$ [$\mathcal{P}B_i$] y su lista de bordes asociada.

Los datos devueltos por este método, serán utilizados por las funciones *obtener_caminos_ui* y *obtener_caminos_vi* para introducir todas las listas de vértices de los caminos u_i [v_i] en una lista, que posteriormente será utilizada para calcular los relatores de la presentación inferior [superior] del grupo del nudo.

```
def __obtener_puntos_vi_ui(self, bordes_y_giros, puntos,
valor_inicial, valor_final):
"""
    Recorre las aristas asociadas a la proyección de un arco
    superior [inferior] y mediante los datos acerca de los
    bordes y giros de las aristas de esa proyección, calcula
    los vértices del camino simple  $u_i$  [ $v_i$ ] asociado a esa
    proyección

    Parámetros:
    -----
    bordes_y_giros:
        Salida del método __obtener_bordes_y_giros_vi_ui
        para los mismos valores de los parámetros
        'valor_inicial' y 'valor_final'.

    puntos: list of lists
        Salida del método obtener_puntos_nudo_dowker,
        modificada por el método dividir_nudo_en_arcos.

    valor_inicial:
        Índice de la lista 'puntos' del primer vértice
        de la primera arista de la proyección del arco
        superior [inferior].

    valor_final:
        Índice de la lista 'puntos' del segundo vértice
        de la última arista de la proyección del arco
        superior [inferior].

    Devuelve:
    -----
    Lista de los vértices asociados al camino  $u_i$  [ $v_i$ ].

    """
```

Obtener los vértices de todos los caminos v_i Este método toma como entrada la lista de vértices del nudo y la salida del método `dividir_nudo_en_arcos`, que permitirá obtener la posición en la lista de vértices, del primer vértice de la primera arista y del segundo vértice de la última arista asociados a una determinada proyección de un arco inferior $\mathcal{P}B_i$. Una vez tengamos esas dos posiciones, para todos los arcos inferiores llamaremos a los métodos `__obtener_bordes_y_giros_vi_ui` y `__obtener_puntos_vi_ui` en ese orden, ya que el segundo utiliza como parámetro la salida del primero, para obtener la lista de vértices de los caminos v_i , $i = 1, \dots, n$. Conforme vayamos obteniendo dichas listas de vértices las iremos añadiendo a otra lista, que llamaremos `vertices_caminos_v_i`, y que será la salida de dicho método. Para la construcción de la lista `vertices_caminos_v_i`, debemos tener en cuenta que la lista asociada al camino v_i tiene que ocupar la posición $i - 1$ en la lista `vertices_caminos_v_i`.

Los datos devueltos por este método, serán utilizados por la función *obtener_presentacion_superior* para calcular los relatores de la presentación superior del grupo del nudo, y por el método *dibujar_caminos_vi*, que como veremos más adelante, gracias al módulo *Turtle* de *Python*, dibujará el diagrama del nudo poligonal dividido en arcos superiores e inferiores, junto con los caminos simples v_i , $i = 1, \dots, n$ y los espacios disjuntos V_i , $i = 1, \dots, n$.

```
def obtener_caminos_vi (self , puntos , aristas , vector_cambio):
    """
    Calcula la posición inicial y final de la lista
    'puntos' asociada a la proyección de cada arco inferior ,
    y llama a los métodos __obtener_puntos_vi_ui y
    __obtener_bordes_y_giros_vi_ui , para obtener los vértices
    asociados a todos los caminos v_i.

    Parámetros:
    puntos: list of lists
            Salida del método obtener_puntos_nudo_dowker ,
            modificada por el método dividir_nudo_en_arcos .

    aristas :
            Salida del método obtener_aristas .

    vector_cambio: list
            Salida del método dividir_nudo_en_arcos .

    Devuelve:
    Lista con las listas de vértices de los caminos simples
    v_i asociados al diagrama del nudo, lista que estará
    ordenada en función del natural 'i'.
    """
```

Obtener los vértices de todos los caminos u_i El método asociado es *obtener_caminos_ui* y su funcionamiento es casi equivalente al del método *obtener_caminos_vi*, visto previamente.

Dibujar los caminos v_i Este método tiene los mismos parámetros que el método *dibujar_nudo_arcos*, además de la lista de los vértices de los caminos v_i , $i = 1, \dots, n$. Este método utilizará el módulo *Turtle* para dibujar el diagrama del nudo dividido en arcos superiores e inferiores, tal y como hicimos en el método *dibujar_nudo_arcos*, y además, dibujará los caminos v_i , $i = 1, \dots, n$ con color verde, y para cada uno de esos caminos simples v_i , el interior de la curva de Jordan que es recorrida por dichos camino se rellenará de color rojo, y representará al espacio V_i , $i = 1, \dots, n$. Asimismo, se representará con una flecha el sentido de giro con el que cada camino v_i recorrerá su curva de Jordan asociada, que recordemos que la recorrián

en sentido antihorario. Puede ser que en algún caso concreto, dicha flecha no se vea representada, ya que este método realiza una comprobación para situar la flecha asociada a cada camino simple v_i , $i = 1, \dots, n$ en una arista cuya longitud sea la suficiente para contener dicha flecha y comprobando que si se sitúa la flecha en dicha arista, la flecha no se intersecta con ninguna de las aristas restantes del diagrama del nudo. Por tanto, puede ocurrir que dicha flecha no se pueda situar en ninguna arista del camino v_i , $i = 1, \dots, n$.

Como nota aclaratoria, cabe destacar que hubiera sido más intuitivo dibujar antes con *Turtle* el diagrama del nudo dividido en arcos superiores e inferiores, y posteriormente los caminos v_i , $i = 1, \dots, n$ y los espacios V_i , $i = 1, \dots, n$. Sin embargo, puesto que el módulo *Turtle* no permite modificar la opacidad del color utilizado para rellenar una determinada curva, si dibujáramos primero el diagrama del nudo y posteriormente los espacios V_i , $i = 1, \dots, n$, taparíamos partes de dicho nudo, entre ellas todas las proyecciones de los arcos inferiores de este.

```
def dibujar_caminos_vi (self , puntos , puntos_vi , numeros_cambio):
    """
        Dibuja el diagrama del nudo dividido en arcos superiores
        e inferiores , junto con las curvas recorridas por los
        caminos  $v_i$  , cuyo interior serán los espacios  $V_i$  , que
        colorearemos con color rojo .

        Parámetros :
        -----
        puntos : list of lists
            Salida del método obtener_puntos_nudo_dowker ,
            modificada por el método dividir_nudo_en_arcos .

        puntos_vi :
            Salida del método obtener_caminos_vi .

        vector_cambio : list
            Salida del método dividir_nudo_en_arcos .

    """
```

Dibujar los caminos u_i El método *dibujar_caminos_ui* tiene un funcionamiento casi equivalente al método *dibujar_caminos_vi* que acabamos de ver. Las únicas diferencias son que toma como parámetro la lista de los vértices de los caminos u_i , $i = 1, \dots, n$, en vez de la de los vértices de los caminos v_i , $i = 1, \dots, n$. Dichos vértices representan las curvas de Jordan recorridas por los caminos u_i , $i = 1, \dots, n$, que representaremos con color lila, y el interior de dichas curvas serán los espacios U_i , $i = 1, \dots, n$, que estarán coloreados con color azul. En este caso, como vimos en la [Sección 3.1](#), los caminos recorrerán dichas curvas de Jordan en sentido horario, por lo que dibujaremos las flechas simbolizando dicho sentido.

Intersección de aristas verticales superiores [inferiores] del nudo con arista horizontal de camino v_i [u_i] Este método es llamado por los métodos *obtener_presentacion_superior* y *obtener_presentacion*

_inferior para hallar las aristas verticales del nudo asociadas a arcos superiores [inferiores] de este, que se intersectan con una determinada arista horizontal del camino $v_i [u_i]$, $i = 1, \dots, n$. Necesitaremos saber tanto las proyecciones de los arcos superiores asociadas a las aristas verticales que se intersecten con la arista horizontal estudiada de $v_i [u_i]$, así como el signo del cruce de dichas aristas verticales la arista horizontal, que se necesitará para calcular el relator i de la presentación superior [inferior] del nudo.

En consecuencia, este método recibe como parámetro la salida de la función *__obtener_aristas_superiores_orientadas* [*__obtener_aristas_inferiores_orientadas*], y comprueba para cada una de las aristas verticales incluidas en esta lista, si se intersecta con la arista horizontal que estamos analizando del camino $v_i [u_i]$. En caso de intersectarse alguna de esas aristas, dado que estamos trabajando con las aristas orientadas en función de la orientación del nudo, calcularemos el signo de dicho cruce. Dado que además, la lista de aristas orientadas verticales recordemos que está compuesta de n listas de aristas, siendo la lista en la posición $i = 0, \dots, n - 1$, la lista de aristas orientadas verticales asociadas a la proyección superior $\mathcal{P}A_{i+1}$ [inferior $\mathcal{P}B_{i+1}$], tendremos también el índice del arco superior [inferior] que se intersecta con dicha arista horizontal del camino $v_i [u_i]$. Con estos dos datos, el índice del arco superior [inferior] cuya proyección se intersecta con la arista horizontal del camino $v_i [u_i]$, y el signo de dicho cruce, almacenaremos en una lista el valor entero asociado al índice de dicho arco superior [inferior], acompañado del signo “-” en caso de ser el cruce negativo.

También, habrá que ordenar la lista resultante en función del orden en el que se producen dichas intersecciones, teniendo en cuenta que el camino $v_i [u_i]$ recorre su curva de Jordan asociada en sentido antihorario [horario].

```
def __interseccion_aristas_verticales(self, segmento,
aristas_inferiores_o_superiores):
    """
```

Halla las aristas verticales superiores [inferiores] que se intersectan con la arista dada por la lista 'segmento', calculando el signo de dichos cruces, y el arco superior [inferior] al que pertenezcan esas aristas verticales.

Parámetros:

segmento:

Lista de dos puntos, asociados a una arista horizontal del camino $v_i [u_i]$.

aristas_inferiores_o_superiores:

*Caso camino v_i : Salida de *__obtener_aristas_superiores_orientadas*.*

*Caso camino u_i : Salida de *__obtener_aristas_inferiores_orientadas*.*

Devuelve:

Lista de los índices asociados a las proyecciones de los arcos superiores [inferiores] que se intersectan con la arista horizontal dada por 'segmento', acompañados del signo de dicho cruce.

"""

Intersección de aristas horizontales superiores [inferiores] del nudo con aristas verticales de camino v_i [u_i] El método asociado a esta funcionalidad es `__interseccion_aristas_horizontales`, cuyo objetivo es el mismo que el del método `__interseccion_aristas_verticales`, pero en este caso se recibe como parámetro una arista vertical de un camino v_i [u_i], $i = 1, \dots, n$, y hay que hallar los arcos superiores [inferiores] para los cuales alguna de sus aristas horizontales asociada se intersecta con dicha arista vertical, al igual que el signo de dichas intersecciones. El resto de pasos son equivalentes a los del método `__interseccion_aristas_verticales`.

Obtener presentación superior del nudo Este método toma como entrada los vértices de todos los caminos v_i , $i = 1, \dots, n$, y para cada uno de esos caminos, obtiene las aristas orientadas que lo componen, que se distinguirán en aristas horizontales y verticales. Para calcular los arcos superiores con los que se intersectan dichas aristas horizontales y verticales del camino v_i y el signo de dichos cruces, para cada una de esas aristas horizontales se llamará al método `__interseccion_aristas_verticales` y para cada una de las aristas verticales al método `__interseccion_aristas_horizontales`. A dichos métodos se les pasará como argumento la lista de aristas superiores orientadas, ya que al ser los espacios V_i , $i = 1, \dots, n$ disjuntos, dichos caminos v_i no podrán intersectarse con ninguna arista inferior del diagrama del nudo. Hay que tener en cuenta que tendremos que ir recorriendo las aristas horizontales y verticales de los caminos v_i , $i = 1, \dots, n$ y llamando a los métodos descritos anteriormente en sentido antihorario, puesto que recordemos que dichos caminos recorren su curva de Jordan asociada en sentido antihorario.

Una vez tengamos todos los datos de las intersecciones de las aristas de un camino v_i , $i = 1, \dots, n$ con los arcos superiores del nudo, habremos obtenido toda la información acerca del relator i -ésimo de la presentación superior del grupo del nudo. Por lo tanto, una vez hayamos realizado este mismo proceso para todos los caminos v_i , $i = 1, \dots, n$, habremos obtenido todos los relatores de la presentación superior del nudo, y como el número de generadores será el número de arcos superiores del nudo, habremos obtenido ya la presentación superior del grupo del nudo.

```
def obtener_presentacion_superior(self, puntos_vi, puntos,
vector_cambio):
    """
```

Para cada camino v_i , halla los arcos superiores que se intersectan con las aristas de v_i , recorridas en sentido antihorario, además de hallar el signo de dichas intersecciones. Estos datos determinarán los relatores de la presentación superior del grupo del nudo.

Parámetros:

puntos_vi:

Salida del método obtener_caminos_vi.

puntos: list of lists

*Salida del método obtener_puntos_nudo_dowker,
modificada por el método dividir_nudo_en_arcos.*

vector_cambio: list

Salida del método dividir_nudo_en_arcos.

Devuelve:

*Lista de las listas de números enteros, asociada cada una de
esas listas a un relator de la presentación superior del
grupo del nudo.*

"""

Obtener presentación inferior del nudo El método asociado a esta funcionalidad es *obtener_presentacion_inferior* que tiene un funcionamiento similar al del método *obtener_presentacion_superior*, diferenciándose en que en vez de recibir como entrada los vértices de los caminos v_i , $i = 1, \dots, n$, recibe como entrada los vértices de los caminos u_i , $i = 1, \dots, n$, ya que estos últimos van a ser los necesarios para obtener la presentación inferior del grupo del nudo. Además, debemos recorrer las aristas de cada camino u_i , $i = 1, \dots, n$ en sentido horario, ya que recordemos que dichos caminos recorren sus curvas de Jordan asociadas en sentido horario. En este caso, a los métodos *__interseccion_aristas_verticales* e *__interseccion_aristas_verticales* habrá que pasarles como argumento la lista de aristas inferiores orientadas, ya que dichos caminos u_i , $i = 1, \dots, n$ no pueden intersectarse con ninguna arista superior, al ser los espacios U_i , $i = 1, \dots, n$ disjuntos.

Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [Ada94] Colin C Adams. *The knot book*, pages 31–40. American Mathematical Soc., 1994. [Citado en págs. [xv](#) and [79](#)]
- [Adk15] Will Adkisson. An overview of knot invariants. *University of Chicago REU*, 2015. [Citado en [pág. 27](#)]
- [CDGW] Marc Culler, Nathan M. Dunfield, Matthias Goerner, and Jeffrey R. Weeks. SnapPy, a computer program for studying the geometry and topology of 3-manifolds. <http://snappy.computop.org>. [Citado en [pág. 83](#)]
- [CF77] Richard H. Crowell and Ralph H. Fox. *Introduction to Knot Theory*. Springer New York, 1977. [Citado en págs. [xiv](#), [xiv](#), [3](#), [11](#), [16](#), [29](#), and [75](#)]
- [FK03] J.B. Fraleigh and V.J. Katz. *A First Course in Abstract Algebra*, pages 341–346. Addison-Wesley world student series. Addison-Wesley, 2003. [Citado en [pág. 31](#)]
- [HM21] Simon Henry and Samuel Mimram. Tietze equivalences as weak equivalences. pages 4–6, 01 2021. [Citado en [pág. 68](#)]
- [HTW98] Jim Hoste, Morwen Thistlethwaite, and Jeff Weeks. The first 1, 701, 936 knots. *The Mathematical Intelligencer*, 20(4):33–48, September 1998. [Citado en [pág. 105](#)]
- [Joh17] G. R. Johnson. The programmatic manipulation of planar diagram codes to find an upper bound on the bridge index of prime knots. *University of Northern Iowa*, pages 1–6, 2017. [Citado en [pág. 79](#)]
- [KNN⁺18] Young Kwun, Abdul Nizami, Waqas Nazeer, Mobeem Munir, and Shin Kang. On gauss diagrams of knots: A modern approach. *Journal of Computational Analysis and Applications*, 01 2018. [Citado en [pág. 92](#)]
- [LF19] Francisco José López Fernández. Topología II, Apuntes de clase. *Universidad de Granada UGR*, 2019. [Citado en págs. [xiv](#) and [29](#)]
- [MAMB21] Amit Jamadagni Miguel Angel Marco Buzunariz. Sage 9.4 reference manual: Knot theory, Aug 2021. <https://doc.sagemath.org/html/en/reference/knots/sage/knots/knot.html>. [Citado en [pág. 85](#)]
- [Mil19] Kyle Miller. What are gauss and dowker-thistlethwaite codes?, Aug 2019. https://math.berkeley.edu/~kmill/2019_8_25/gauss_dt_codes.html. [Citado en págs. [xv](#) and [79](#)]
- [MSU11] Alexei G Myasnikov, Vladimir Shpilrain, and Alexander Ushakov. *Non-commutative cryptography and complexity of group-theoretic problems*, pages 13–15. Number 177. American Mathematical Soc., 2011. [Citado en [pág. 31](#)]
- [Rol76] D. Rolfsen. *Knots and Links*, pages 47–52. Mathematics lecture series. Publish or Perish, 1976. [Citado en págs. [27](#) and [75](#)]
- [Wik21] Wikipedia. List of prime knots — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=List%20of%20prime%20knots&oldid=1050750031>, 2021. [Online; accessed 20-November-2021]. [Citado en [pág. 108](#)]