UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL

Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto

TRABAJO FIN DE GRADO

Trenzas para enlaces pretzel

Autor: Ángel del Pozo Manglano

Tutor:

Pedro María González Manchón

Dpto. de Matemática Aplicada a la Ingeniería Industrial

Madrid, Septiembre 2020

Agradecimientos

Siento una gratitud inmensa hacia mis padres, que siempre han estado ahí cuando les he necesitado, junto con mi hermana.

También a mis abuelas. Sin ellas no habría sido posible llegar hasta aquí.

A mi tutor Pedro, por su dedicación y pasión por las matemáticas, y por ayudarme y apoyarme para lograr que este trabajo se hiciera realidad.

Tengo que agradecer a Javier Gómez y David Fernández el gran equipo que hemos hecho a lo largo de estos 4 años.

Y por supuesto, no olvido a todos los demás que han estado presentes durante este camino.

Finalmente, al equipo y comunidad de Inkscape, herramienta esencial a lo largo de todo el trabajo.

Resumen/Abstract

Este TFG se encuadra en la teoría de nudos. En el capítulo 1 se hace una introducción a su estudio, algunas propiedades y particularidades importantes.

En el capítulo 2 se introduce Inkscape como herramienta fundamental usada en todo el trabajo para representarlos.

En el capítulo 3 se habla de los enlaces pretzel, cómo se construyen y sus propiedades.

El capítulo 4 habla de las trenzas, su relación con los enlaces y sus principales propiedades.

En el capítulo 5 logro encontrar una estrategia general para convertir cualquier nudo pretzel en trenza. Además, muestro un algoritmo que especifica, para cada enlace pretzel de una, tres o un número par de entradas, una trenza concreta cuya clausura es el nudo pretzel.

En el capítulo 6 se implementa el algoritmo obtenido en un programa Python para después llevarlo a una página web y hacerlo accesible online desde cualquier plataforma.

Palabras clave: nudo, enlace pretzel, trenza, círculo de Seifert, Inkscape, Python.

This final degree project is framed in Knot theory. On chapter 1 I do an introduction to its study and some important properties.

On chapter 2 Inkscape is introduced as a fundamental tool used for representing Knots.

On chapter 3 pretzel links are introduced, how they are built and their properties.

On chapter 4 braids are introduced, their relationship with links and main properties.

On chapter 5 I find a strategy to convert any pretzel knot into a braid. Besides, I put together an algorithm that gives, given a pretzel link of one, three or an even number of entries, a braid whose closure is the pretzel link.

On chapter 6 the algorithm is written on a Python program, and a website is made to make it accessible online from any platform.

Keywords: knot, pretzel link, braid, Seifert circle, Inkscape, Python.

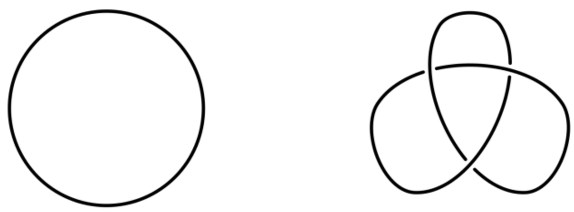
Contenidos

Introducción

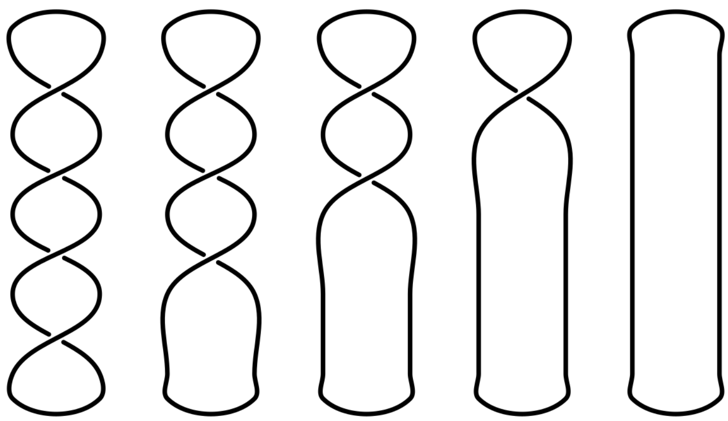
Este proyecto de fin de grado trata de buscar una relación entre trenzas y enlaces pretzel, dos conceptos de la teoría de nudos, una rama de la topología.

Quizá esa frase es demasiado enrevesada, hay muchos conceptos desconocidos y parece un trabalenguas. Trenzas, enlaces pretzel, teoría de nudos, topología, etc. Vamos a analizar estos ingredientes del proyecto para ver qué podemos cocinar con ellos.

¿Qué es un nudo, y qué relación tiene con las matemáticas y la ingeniería? La respuesta a la primera pregunta la tenemos si nos atamos los zapatos y conectamos los herretes de los extremos. Tendremos una cuerda la cual, si empezamos a recorrerla, acabaremos en el punto inicial. ¿Y cómo podemos deshacer el nudo? En la mayoría de los casos será imposible. Estos son el nudo trivial, que es lo que surge si podemos deshacer un nudo, y el nudo trébol, un nudo imposible de deshacer.

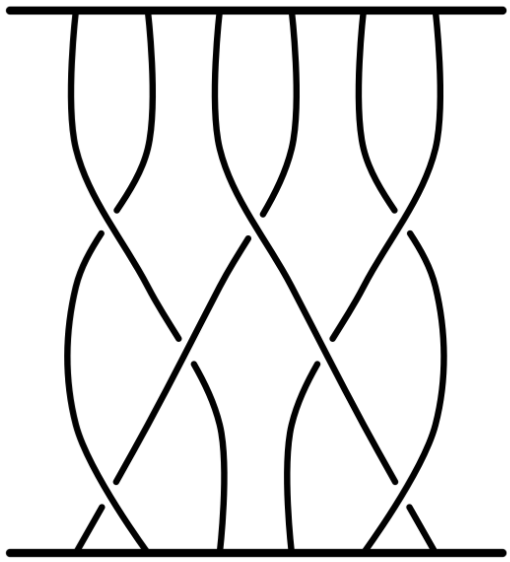


Tenemos que tener en cuenta que hay cosas que podemos hacer con un nudo que no podremos hacer con un cordón, ya que tenemos que pensar en la cuerda de nuestro nudo como si fuese infinitamente deformable y su sección fuese un punto. La intuición nos puede hacer pensar que cuantos más cruces tengamos, más complejo será el nudo, pero puede ser que la situación sea toda la contraria, y que tengamos un diagrama con más cruces que el nudo trébol, pero más sencillo que éste, como puede ocurrir si cogemos el nudo trivial y lo giramos sobre sí.

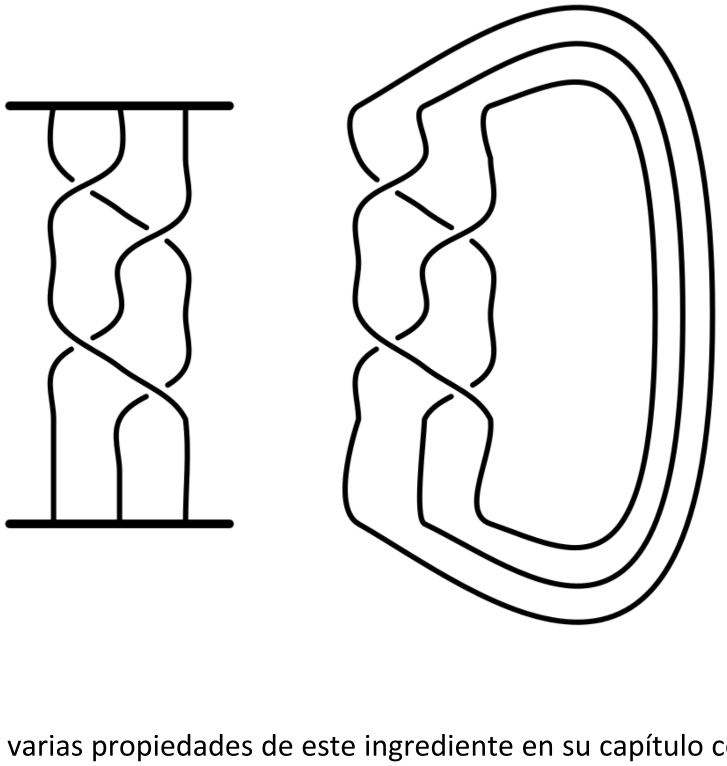


En el capítulo 1. Introducción a los nudos vamos a ver muchas de las propiedades de los nudos, y algunas de ellas nos van a servir de base para el trabajo posterior.

¿Y qué tienen que ver las trenzas en esto? Una trenza es un conjunto de cuerdas unidas a dos barras horizontales, una superior y una inferior. Caen monótonamente sin hacer bucles y se van cruzando entre ellas, como si fuese el cabello de una chica.

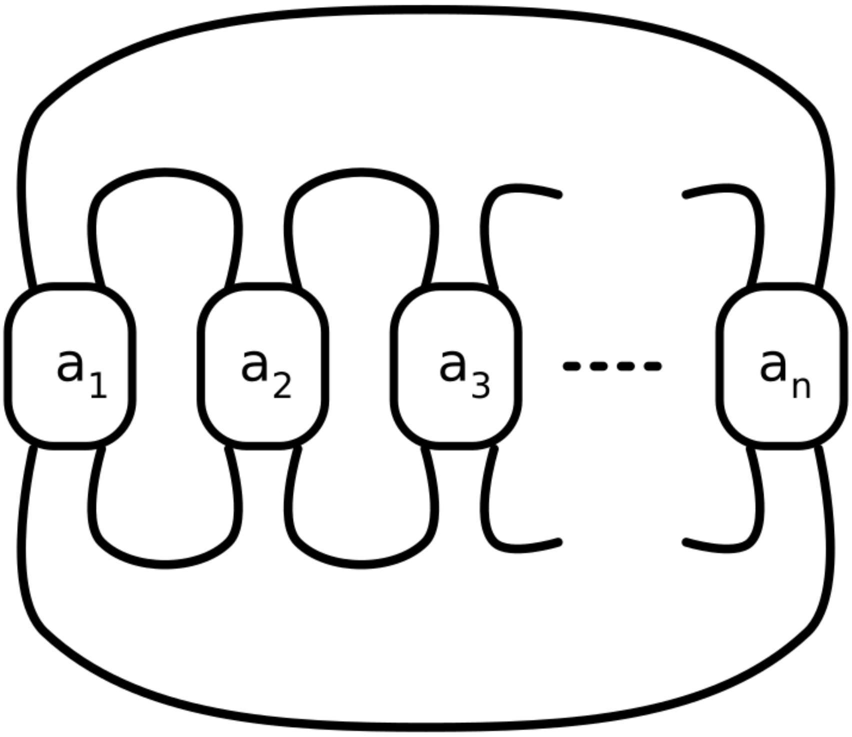


Esto es muy curioso, pero, ¿dónde está la relación con los nudos? Pues resulta que, si cogemos los extremos de las cuerdas y los conectamos uno a uno, clausurando la trenza, siempre obtendremos un nudo o un enlace (una colección de nudos disjunta).

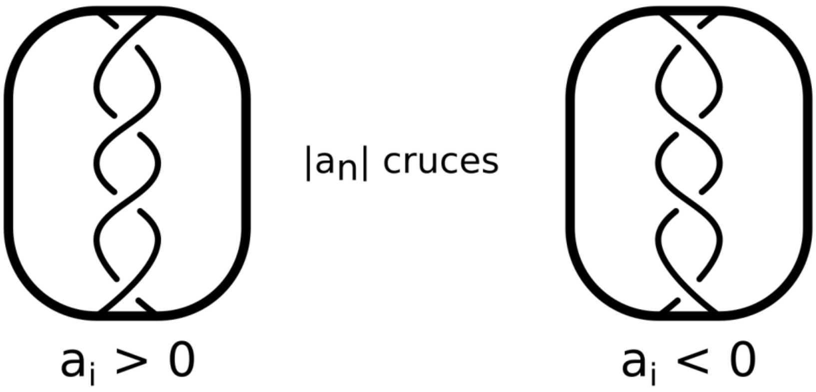


Vamos a ver varias propiedades de este ingrediente en su capítulo correspondiente, 4. Trenzas, junto con el 4.1 Teorema de Alexander, que constituye la sartén del proyecto.

No se nos pueden olvidar los enlaces pretzel, un ingrediente esencial de este proyecto. Un enlace pretzel P(a_1, a_2, …, a_n) tiene *n* entradas, y se construye siguiendo el siguiente esquema:



Dentro de cada cajita tenemos una columna de cruces, tantos como el valor absoluto del número que aparece en la cajita. Además, tenemos que tener en cuenta si el valor en la cajita es positivo o negativo, tomando los cruces de la manera que podemos ver en la siguiente ilustración:



Los nudos nos permiten atarnos los zapatos, las trenzas hacer peinados endiabladamente enrevesados, y los pretzel los podemos ver en algunas galletas saladas muy populares en el extranjero.

Ya tenemos todos los ingredientes que necesitamos para nuestro proyecto. Recordamos que buscamos cocinar un enlace pretzel y salir de la cocina con una trenza. Esto es lo que vamos a ver en el capítulo 5. Trenzas para enlaces pretzel, que es el más relevante y de contenido original del TFG.

Como el lector apreciará en las más de 100 páginas de dicho capítulo, hay muchas variables para cocinar estos enlaces, tenemos que tener cuidado si hay números pares, dónde están,… Hay que ser un cocinero experto para no equivocarnos. Pero nosotros lo que queremos es una Thermomix, que demos un enlace pretzel y ocurra una magia tras el telón que nos acabe devolviendo una trenza. Así que en el capítulo 6. Programación con Python implemento los algoritmos en un programa hecho en Python que hace perfectamente la función de nuestra Thermomix: le damos un enlace pretzel, nos lo clasifica debidamente y nos da la trenza que le corresponde. Como esta Thermomix es un poco cara y necesita que la descarguemos y la instalemos he hecho una página web que nos permite usar el programa sin tener que instalar nada. Así, desde cualquier lugar o dispositivo podremos pedir a domicilio que nos manden un enlace pretzel cocinado como una deliciosa trenza.

La página web está accesible en http://adelpozoman.es/tfg.

Como el lector podrá apreciar en el capítulo II, el mejor cuchillo y tabla para cortar nudos del mercado es Inkscape, un programa de dibujo vectorial que nos permite mover las cuerdas como si fuesen reales, teniendo en cuenta que hay cruces en los que una cuerda irá sobre otra cortando su proyección y moviéndolos al mover las cuerdas.

Todavía no hemos respondido a la pregunta inicial que nos hemos hecho sobre la relación entre los nudos y la ingeniería. Lo que en un principio puede parecer un juego de niños tiene estrechas relaciones con la física teórica, la topología algebraica, la combinatoria y muchos otros campos, pasando por la bioquímica, la criptografía o la robótica.

Una aplicación podría ser la optimización topológica de dispositivos mezcladores mediante diagramas de trenza. Sin entrar a fondo, la eficiencia de un proceso de homogenización de una mezcla de fluidos puede ser medida mediante la entropía topológica [1]. En bioquímica y farmacología, el polinomio de Jones permite distinguir enantiómeros quirales sobre el papel [2]. Las trenzas, y la teoría de nudos en general, pueden usarse también en el estudio de la recombinación del ADN ([3], [4]).

Por otra parte, los grupos de trenzas se proponen desde hace más de dos décadas como base de sistemas de encriptado de clave pública. ¿Qué significa esto en cristiano? Pues que podemos usarlo para asegurar los datos que se transfieren cuando usamos internet como nuestra tarjeta del banco, nuestra información médica, mensajes, etc. Los sistemas actuales de clave pública suelen usar el problema del algoritmo o el problema de la factorización. En el del algoritmo si queremos romper la seguridad tenemos que encontrar el número que han usado como exponente de un número conocido para obtener otro número conocido. En el de la factorización tenemos que buscar los factores primos de un número que tiene una cantidad monstruosa de dígitos. Así que se está probando la teoría de los grupos combinatorios para establecer un nuevo sistema que no sea vulnerable a los fallos en los existentes. [5]

1. Introducción a los nudos

Un nudo surge al coger una cuerda arbitraria y pegar sus extremos. Podemos tener una construcción más o menos complicada, pero siempre se cumplirá que, si nos colocamos en un trozo de la cuerda y empezamos a recorrerla, tarde o temprano llegaremos al mismo lugar. Tenemos que pensar en el nudo como una cuerda sin grosor, y cuya sección es un punto. Para tratar los nudos se dibujan diagramas, que son como proyecciones, pero con una diferencia esencial: en los diagramas los cruces están marcados de tal manera que podemos saber qué cuerda va por encima y qué cuerda va por debajo.

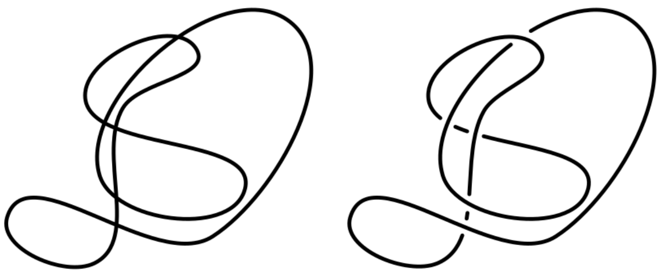


Ilustración 1. Una proyección y un diagrama de un nudo.

Sin tijeras jamás podremos deshacer el nudo. A menos que estemos hablando del nudo trivial, que es el único nudo con cero cruces. A parte del nudo trivial, el nudo más famoso es el nudo trébol.

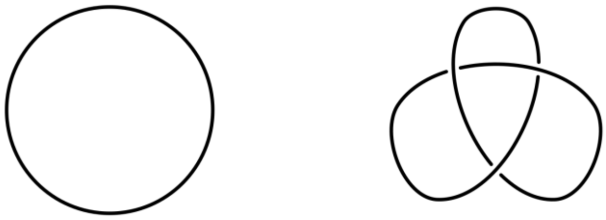


Ilustración 2. Nudo trivial y nudo trébol.

La cuerda de nuestro nudo es deformable, casi como si estuviese hecha de goma. Podemos moverla y doblarla, pero nunca puede atravesarse a sí misma.

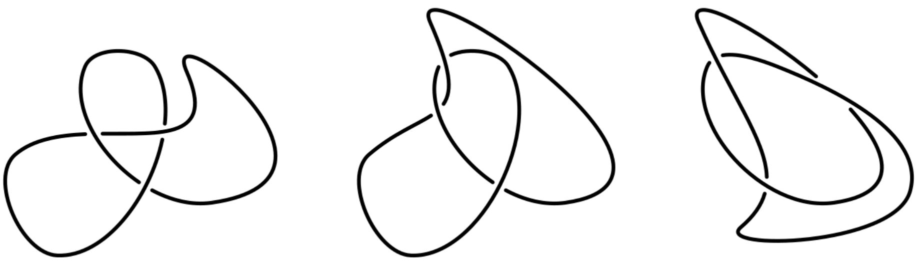


Ilustración 3. Nudo trébol deformado.

Para modificar el diagrama de un nudo usamos los movimientos de Reidemeister.

1.1 Movimientos de Reidemeister

Suponiendo que tenemos dos diagramas del mismo nudo, deberíamos poder hacer una serie de movimientos que nos lleven de la una a la otra. Pues bien, Reidemeister demostró que, teniendo dos diagramas del mismo nudo, a partir de una serie de movimientos (llamados así en su honor [6]) podemos ir de un diagrama al otro. Estos movimientos son locales, es decir, aislamos una parte del nudo y la modificamos sin afectar al resto del nudo.

Teorema de Reidemeister: dos diagramas definen el mismo nudo si y sólo si se puede pasar del uno al otro mediante una secuencia finita de movimientos de Reidemeister.

Los movimientos de Reidemeister son de tres tipos:

- Movimiento de tipo I: nos permite añadir o quitar un rizo (*loop*) sobre sí mismo.

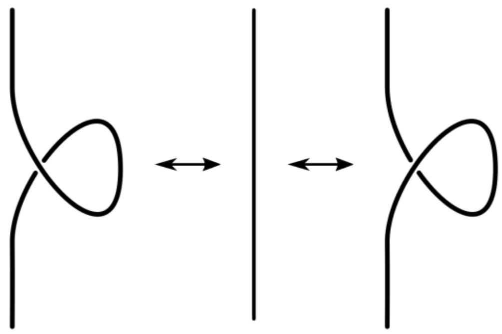


Ilustración 4. Movimiento de Reidemeister de tipo I.

- Movimiento de tipo II: nos permite añadir o eliminar dos cruces.

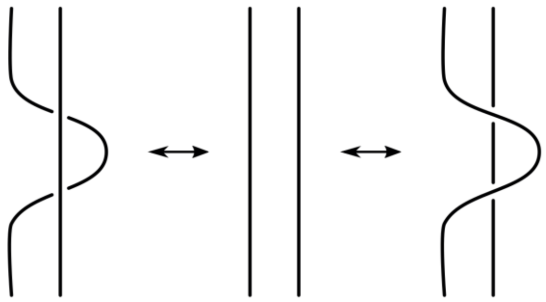


Ilustración 5. Movimiento de Reidemeister de tipo II.

- Movimiento de tipo III: nos permite deslizar una cuerda por encima de un cruce al otro lado.

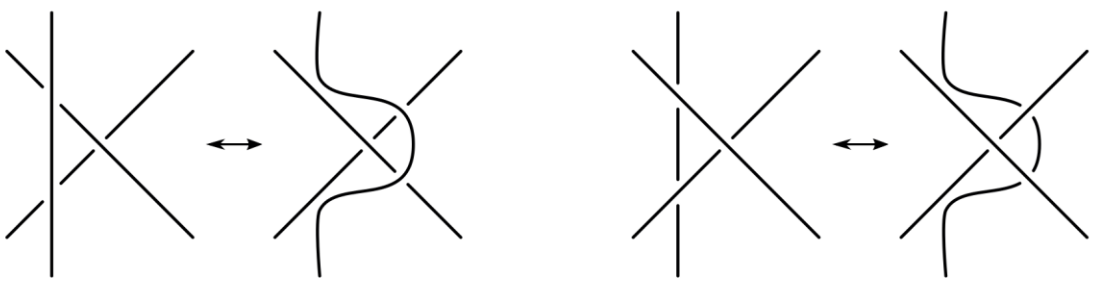


Ilustración 6. Movimiento de Reidemeister de tipo III.

Veamos un ejemplo de movimientos de Reidemeister.

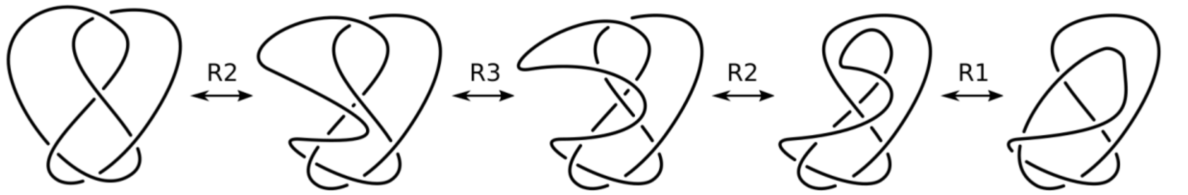


Ilustración 7. Movimientos de Reidemeister.

El nudo de la figura 8 es el único nudo que tiene un diagrama con 4 cruces, y no con menos. Si cambiamos todos los cruces del diagrama del nudo de la figura 8 podemos observar que el nudo sigue siendo el mismo. Esto ocurre con los nudos que se llaman aquirales, es decir, que coinciden con su imagen especular. Los nudos que no coinciden con su imagen especular se llaman quirales, como el trébol.

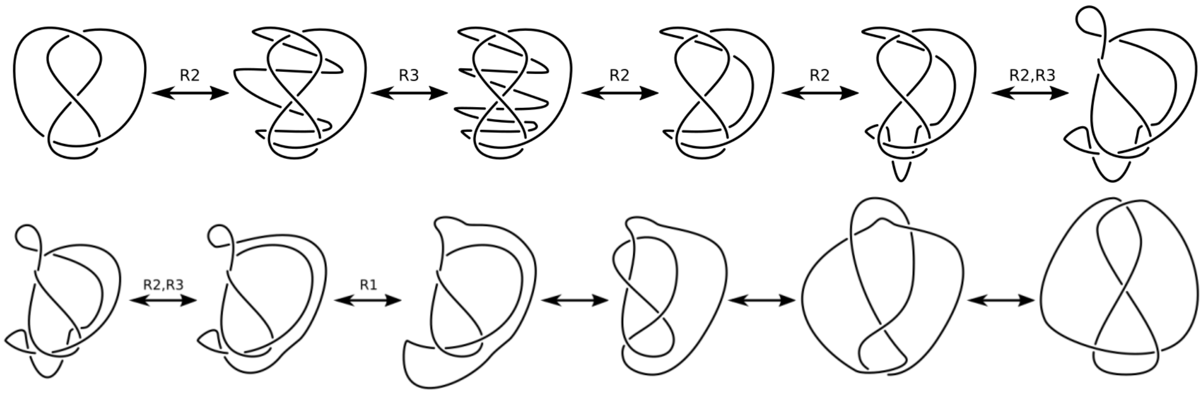


Ilustración 8. La imagen especular del nudo 8 a partir de movimientos de Reidemeister.

Un ejemplo de movimiento que no se puede hacer es el siguiente, ya que como se puede ver se deshacen cruces de forma artificial.

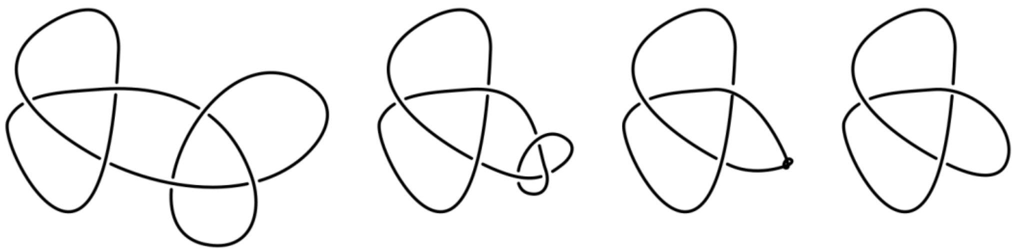


Ilustración 9. Movimiento inválido.

Uno de los ejercicios sobre nudos realizado ha sido transformar el siguiente enredo local en este otro. Se puede apreciar intuitivamente que claramente son el mismo, pero la secuencia de movimientos de Reidemeister que hay que hacer ni es inmediata ni es corta, como vamos a ver a continuación.

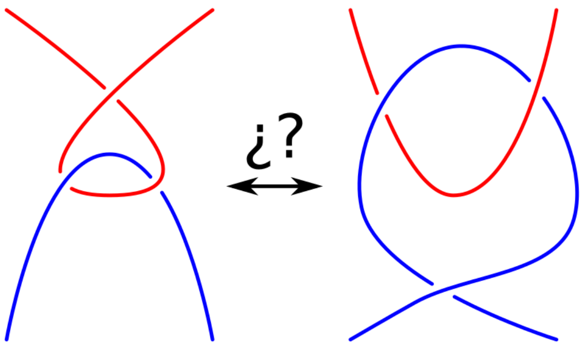


Ilustración 10. Dos proyecciones del mismo enredo.

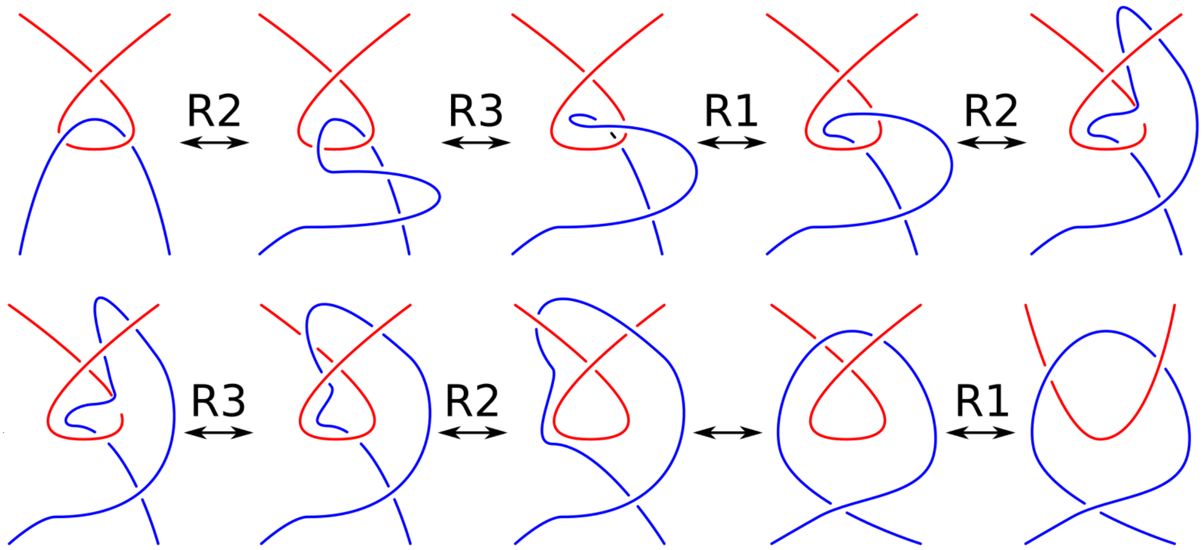


Ilustración 11. Movimientos de Reidemeister.

Podríamos pensar, a la vista de lo que dice el teorema de Reidemeister, que ya podemos saber si dos diagramas representan al mismo nudo viendo una serie de movimientos de Reidemeister que va de uno al otro. El problema es que no hay una cota superior en el número de movimientos que esto nos pueda requerir.

Vayamos a una pregunta aparentemente más simple, ¿estamos seguros de que el nudo trivial y el nudo trébol son realmente nudos diferentes? Es decir, aunque el trébol tenga más cruces, ¿cómo sabemos que no los podemos deshacer haciendo movimientos de Reidemeister?

Buena parte del estudio de la teoría de nudos trata sobre esto. La mejor prueba empírica que podemos hacer es coger los cordones de nuestros zapatos, construir el nudo trébol y pegar los extremos. Podríamos pasar una cuarentena intentando deshacerlo para llegar al nudo trivial, pero mucho antes estaremos convencidos de que es imposible. Una propiedad muy visual de los nudos nos dice si son distintos: la tricoloreabilidad. Esta propiedad es una invariante de los nudos. Una invariante tiene el mismo valor para cualquier diagrama de un nudo, y así, podríamos saber si dos números son diferentes.

1.2 Tricoloreabilidad

La tricoloreabilidad nos permite diferenciar unos nudos de otros, aunque no siempre. Para ello vamos a coger un nudo, y vamos a marcar la cuerda cada vez que pase por un cruce inferior, sin importar el número de cruces superiores que haya por el camino. A los espacios entre las marcas los llamamos hebras. Decimos que un nudo es tricoloreable si podemos pintar cada hebra de uno de 3 colores disponibles, de tal manera que en cada cruce o bien se juntan los tres colores o sólo se junta uno. Además, es obligatorio usar los tres colores.

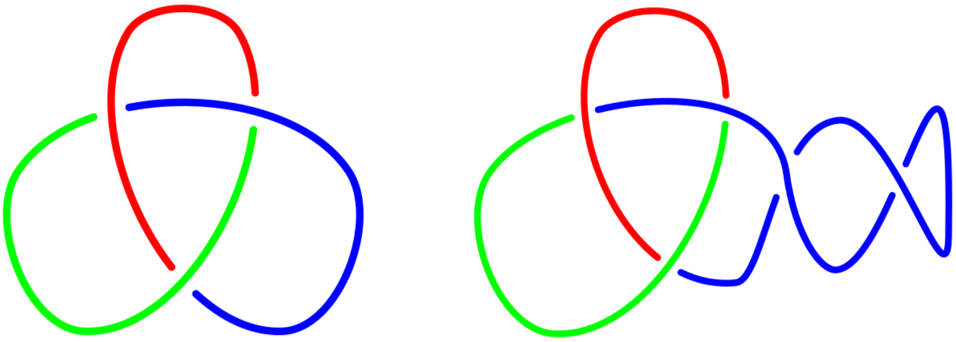


Ilustración 12. Tricoloreabilidad en el nudo trébol.

Esta ilustración que he hecho con Inkscape es un ejemplo perfecto, ya que en el primer diagrama coinciden los 3 colores en todos los cruces, mientras que en el segundo en algunos cruces sólo coincide un color. En lo que tenemos que fijarnos es que en ningún cruce tenemos sólo 2 colores, por lo que el trébol es tricoloreable.

Ahora vamos a explicar la tricoloreabilidad en los movimientos de Reidemeister. En un R1 no hay ningún problema, ya que como sólo hay una cuerda sólo hay un color. En R2 tenemos que colorear el trozo que nos sale del tercer color para cumplir la tricoloreabilidad.

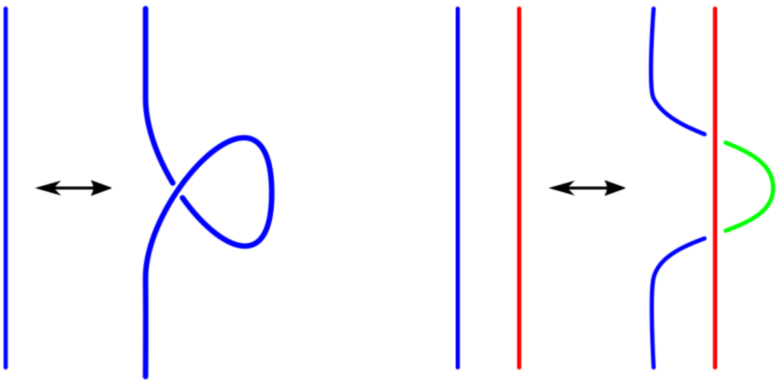


Ilustración 13. Tricoloreabilidad en R1 y R2.

Con los movimientos de R3 es menos inmediato, pero también se conserva la tricoloreabilidad.

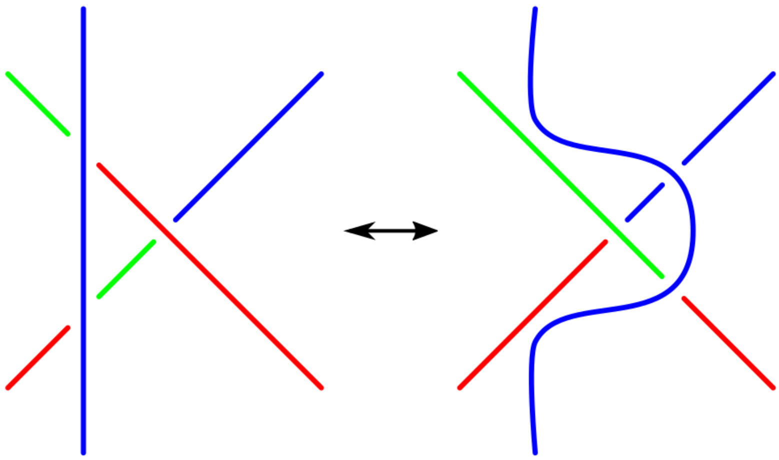


Ilustración 14. Tricoloreabilidad en R3.

Ya que, como muestran las ilustraciones anteriores, los movimientos de Reidemeister no afectan a la tricoloreabilidad, si un diagrama de un nudo es tricoloreable lo serán todos sus diagramas. Del mismo modo, si no es tricoloreable, ninguno de sus diagramas lo será. Pues bien, como el nudo trivial no es tricoloreable (recordar que tenemos que utilizar los 3 colores), estamos seguros de que no es un diagrama del nudo trébol. También podemos concluir que cualquier nudo tricoloreable no es el nudo trivial.

Como ejemplo final vamos a ver que el nudo trébol y el nudo de la figura de 8 son de hecho diferentes, ya que el segundo no es tricoloreable. De hecho, la tricoloreabilidad no nos puede decir que el nudo de la figura de 8 y el nudo trivial son diferentes, ya que ambos no son tricoloreable.

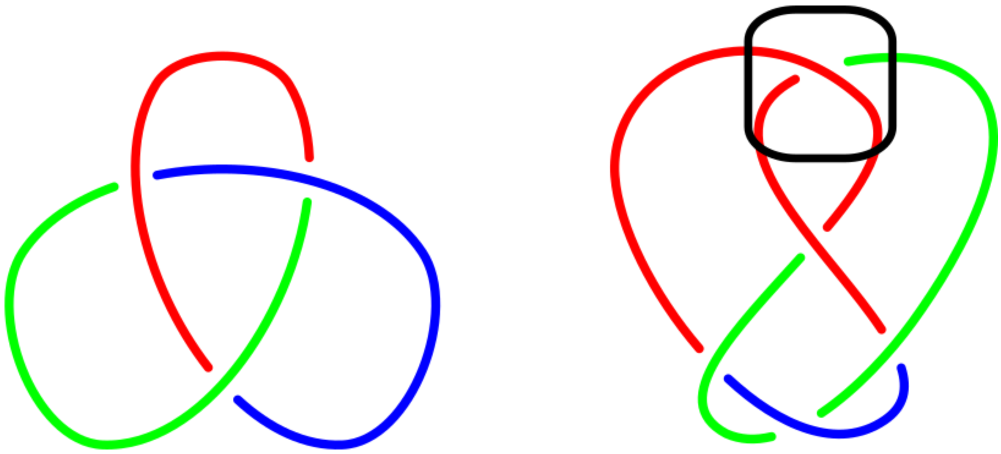


Ilustración 15. El trébol es tricoloreable, el nudo de la figura 8 no lo es.

1.3 Orientación de nudos

Una herramienta muy útil para el estudio de los nudos es orientarlos. Para orientar un nudo sólo tenemos que colocarnos en cualquier punto de su trazado y elegir una dirección que seguir. Mientras la seguimos vamos pintando flechas que recuerden la dirección adecuada. Tarde o temprano llegaremos al punto de origen, y entonces diremos que tenemos nuestro nudo orientado. Si tenemos un enlace tendremos que repetirlo tantas veces como componentes tengamos, teniendo en cuenta que un enlace tiene más de una componente.

Cuando se orienten nudos voy a procurar que siempre tengan la cuerda superior orientada en sentido anti horario, como se puede apreciar en la siguiente ilustración.

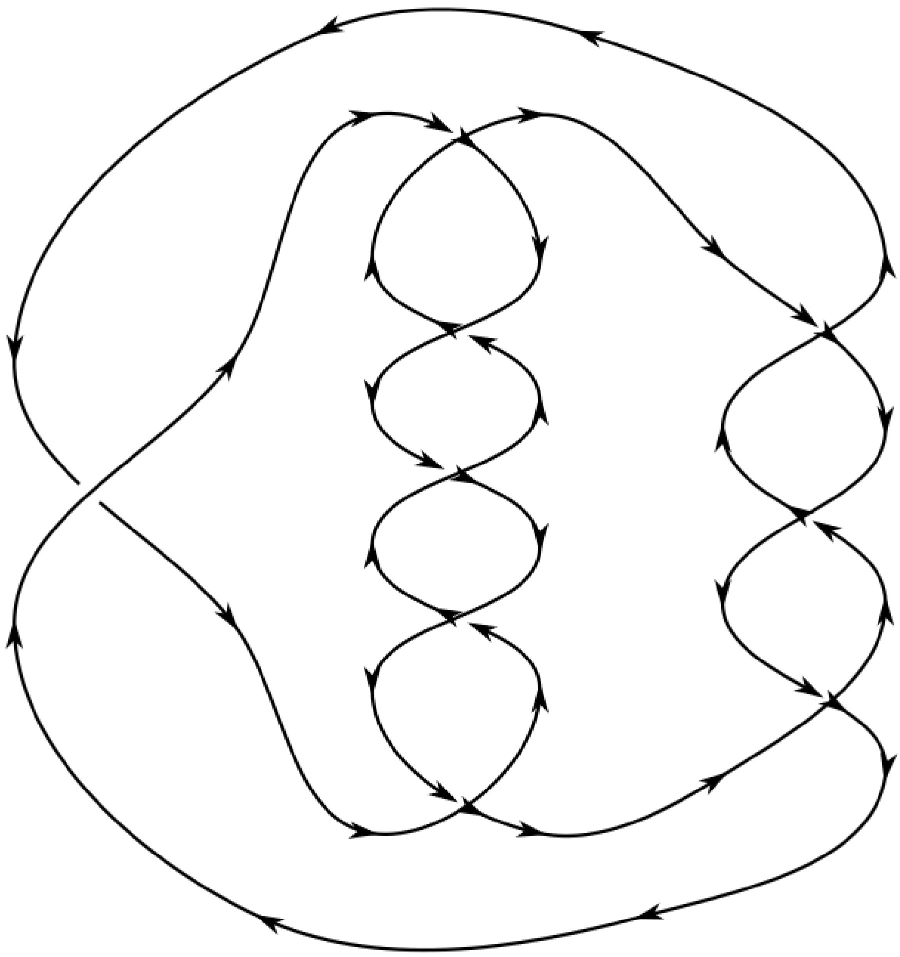


Ilustración 16. Nudo orientado.

1.4 Número de cruce

El número de cruce es el menor número de cruces posible que podemos tener en cualquier diagrama de un nudo.

Usando la tricoloreabilidad ya hemos visto que en el nudo trébol el número de cruce es 3, y podemos observar otros diagramas del mismo nudo con un número mayor de cruces.

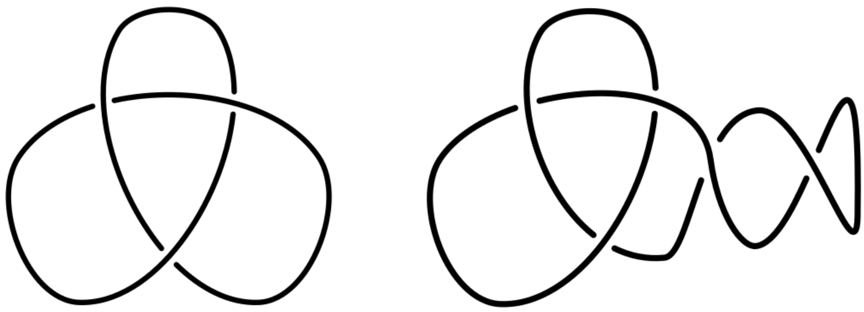


Ilustración 17. Dos diagramas del nudo trébol.

Ahora bien, ¿cómo podemos saber el número de cruce de un nudo? Para empezar, cogemos un diagrama de éste y contamos los cruces. Estamos seguros de que el número de cruce va a ser este número *n* u otro menor, pero nunca uno superior. Si ya conocemos todos los nudos que tienen menos de *n* cruces y nuestro nudo no está en la lista de nudos que tienen menos de *n* cruces, entonces *n* es el número de cruce.

De aquí sacamos una de las formas más populares de clasificar los nudos, según su número de cruce. Así sabremos, por ejemplo, que si tenemos un diagrama con 3 cruces que no representa al nudo trivial, es necesariamente un diagrama del nudo trébol.

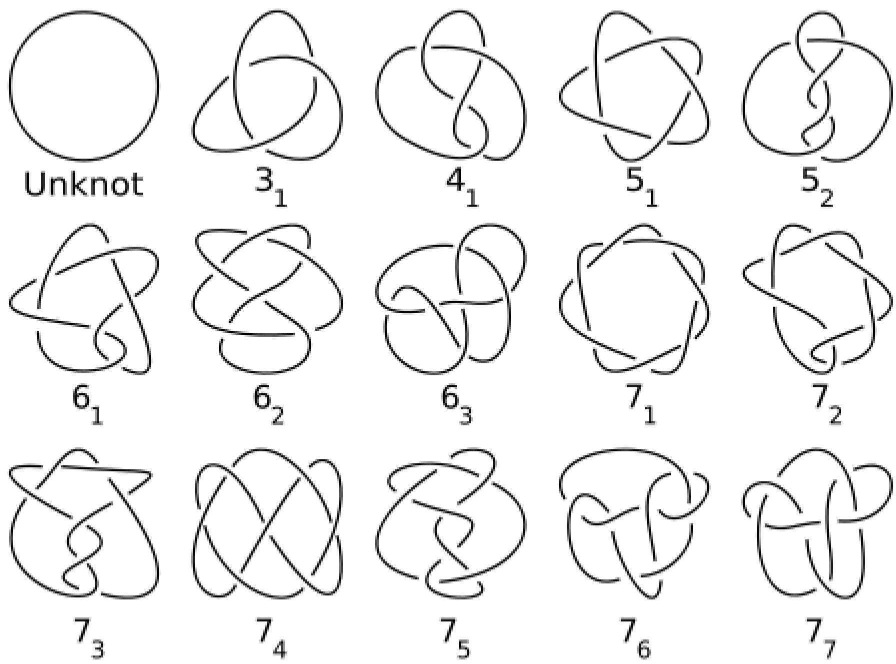


Ilustración 18. Tabla de nudos según el número de cruce [7].

1.5 Composición de nudos

Imaginemos que tenemos dos nudos separados en dos habitaciones. Ahora los unimos mediante un rectángulo de cartón que se apoya sus aristas extremas en cada nudo de manera que no se corten. Esto es la composición de nudos, y a este nudo lo denotamos *J#K*. A los nudos a partir de los cuales se ha formado la composición se los llama nudos factor.

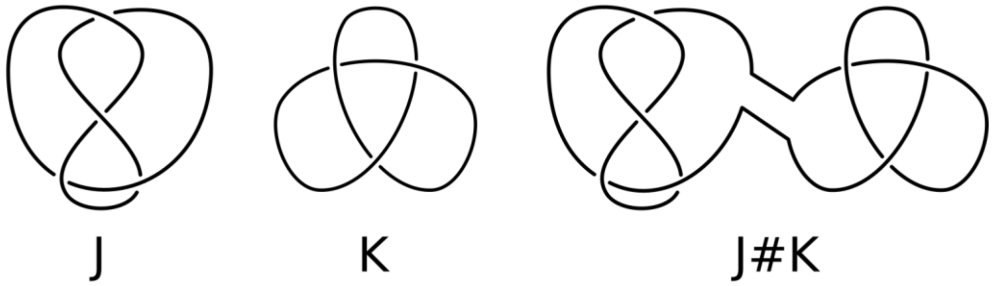


Ilustración 19. Composición de nudos.

El rectángulo nunca puede cortar a los nudos.

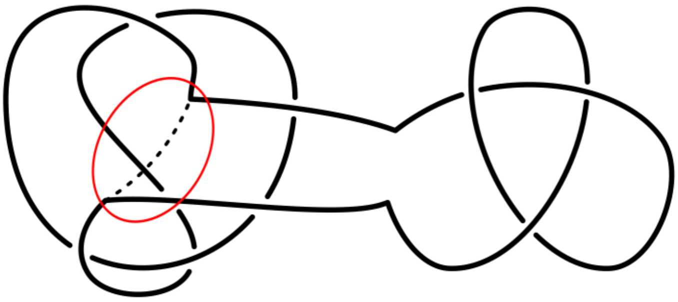


Ilustración 20. Composición de nudos incorrecta.

Es en cierta manera obvio que si componemos un nudo con el nudo trivial obtendremos el nudo inicial. Esto se puede ver como el producto de dos números: si el segundo es la unidad el resultado será el primero.

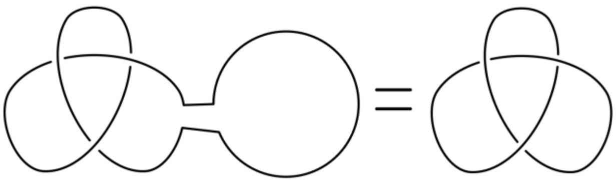


Ilustración 21. Composición con el nudo trivial.

De la misma manera un nudo primo es un nudo que no es composición de otros. Por ejemplo, el nudo trébol y el nudo de la figura ocho son primos.

Una pregunta que puede dar que pensar es si el nudo trivial es compuesto, es decir, si podemos deformarlo de tal manera que acabemos viendo dos nudos no triviales unidos por dos cuerdas. Por suerte, esto es imposible, por lo que el nudo trivial es primo.

En la tabla del apéndice del libro de Adams [8] nos encontramos con los diagramas de los nudos primos con número de cruce 9 o menos, como una tabla de números primos. Al igual que los números primos, están clasificados de menor a mayor, pero en este caso de menor a mayor número de cruce.

Una pregunta final que nos puede surgir es si cambiando el lugar de apoyo del rectángulo nos puede dar otro nudo, y la respuesta es que sí. No voy a entrar en esta explicación ([8], página 11), pero daré un caso en el que el lugar elegido no influye, como se ve en la siguiente figura. Lo único que hay que hacer para ir de un diagrama al otro es comprimir uno de nuestros nudos para deslizarlo a lo largo del otro hasta llegar al diagrama deseado.

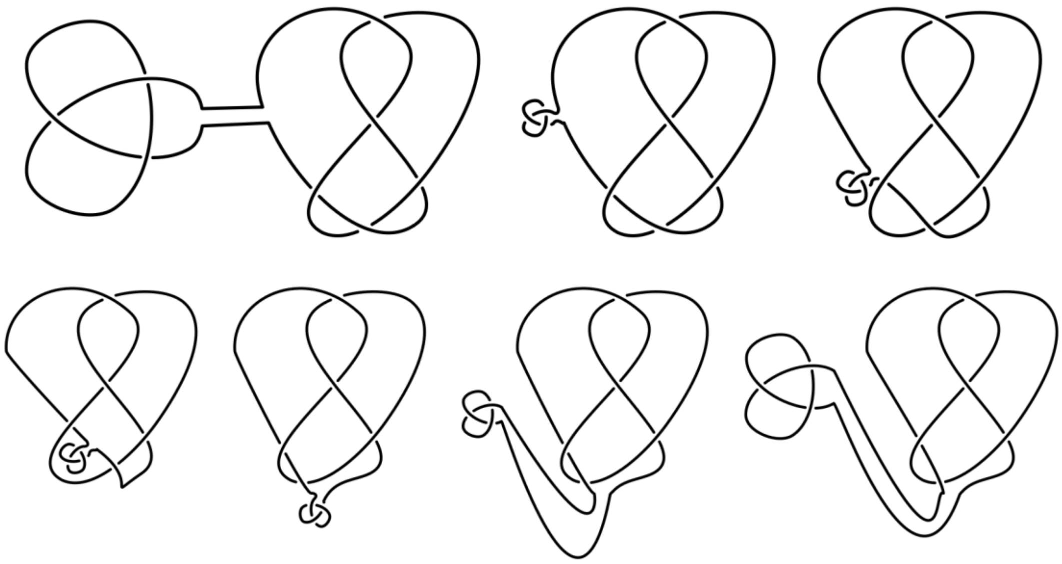


Ilustración 22. Composición.

1.6 Enlaces

Un enlace es una colección finita de nudos disjuntos. Cada nudo es llamado una componente del enlace. Cada componente es independiente: si estamos recorriendo una es imposible llegar a la otra. Por ello tendremos que orientar en más de un paso, ya que hay que ir componente a componente. Si en los nudos hablando de los más sencillos tenemos al nudo de trébol, entre los enlaces más sencillos tenemos el enlace de Hopf y el de Whitehead.

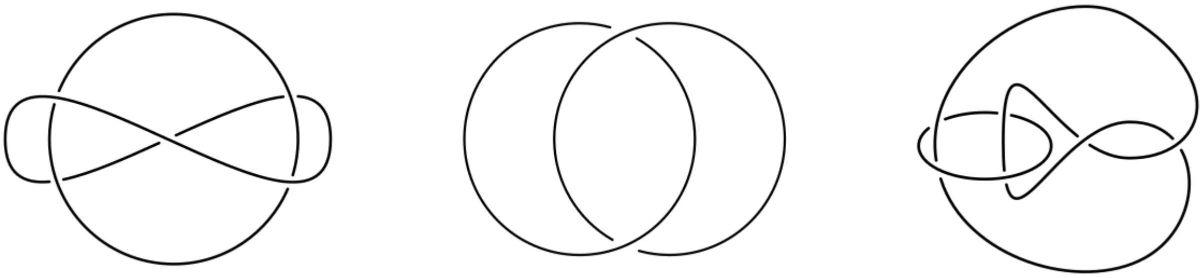


Ilustración 23. Enlace trivial de dos componentes, enlace de Hopf y enlace de Whitehead.

Como está hecho de 2 cuerdas decimos que es un enlace de 2 componentes. El siguiente enlace más conocido es el enlace Borromeo, que podemos encontrar en diferentes logos como el de Audi o los juegos olímpicos [9]. Lo he coloreado para constatar mejor que tiene más de una componente.

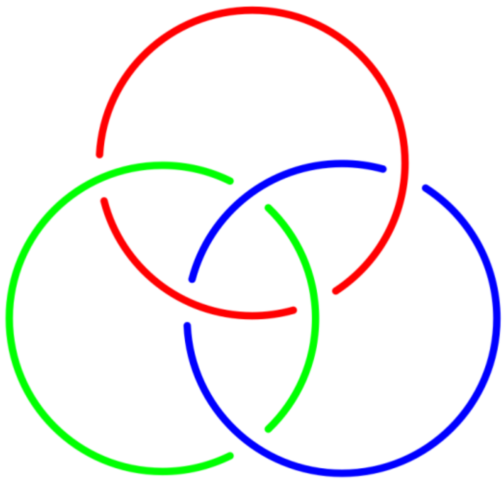


Ilustración 24. Enlace Borromeo.

Decimos que un enlace es separable si podemos separar sus componentes por un plano en el espacio tridimensional, sin cortar a ninguno. Por ejemplo, el enlace trivial de dos componentes que hemos visto en la Ilustración 23.

Con los enlaces nos surge el mismo dilema que con los nudos: ¿cómo sabemos si dos proyecciones representan al mismo nudo? Surgen los mismos problemas que con los nudos, así que es un buen momento para terminar este capítulo, que espero que haya introducido al lector en el fascinante mundo de la teoría de los nudos.

2. Inkscape

Este capítulo es un pequeño manual sobre la representación y manipulación digital de diagramas de enlaces, ya que no he encontrado ninguna referencia de calidad en este aspecto. Incluir este capítulo aquí nos parece necesario para dar a lector una herramienta con la que manejar los nudosy poder observarlos con detenimiento gracias a los archivos adjuntos.

Esto surge como solución al método que he usado al principio, dibujar a mano paso por paso y con colores. El problema es que es muy fácil equivocarse, y borrar sólo ensucia el dibujo.

Por eso hice una pequeña investigación sobre los métodos que se suelen usar para dibujar nudos digitalmente. Los primeros resultados fueron poco fructíferos, pues hablaban de hacer nudos simples de forma llamativa para logos y otro tipo de elementos gráficos.

Mi tutor me recomendó CorelDraw, IPE o LaTeX, que suelen usar los matemáticos para esta clase de labores. CorelDraw hace dibujos vectoriales y se parece bastante a Inkscape, pero como vamos a ver a continuación no es rival para dibujar nudos. IPE en cambio es bastante complejo, y pide entrada matemática. Con LaTeX ocurre básicamente lo mismo, hay que escribir un código que luego dibujará el nudo, pero el dibujo en sí no será editable. Lo que yo buscaba era un programa que permitiese mover las cuerdas de los nudos como si de cuerdas reales se tratase, y que representase los cruces debidamente.

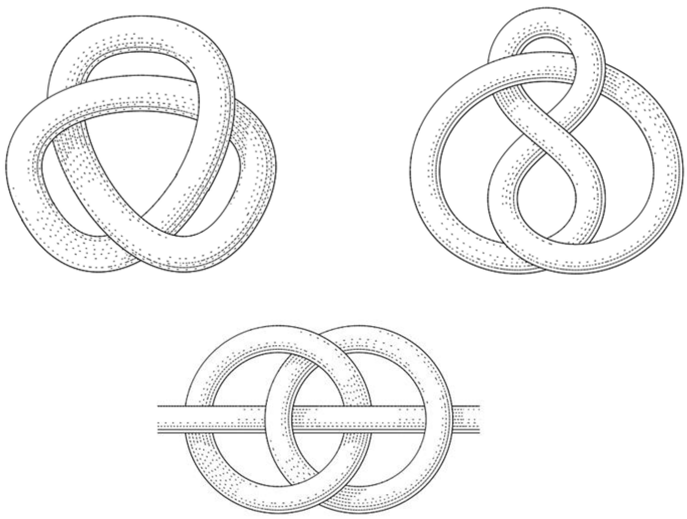


Ilustración 25. Representaciones de nudos usando fiziko, un complemento para LaTeX [10].

Teniendo claro que el candidato iba a ser un programa de dibujo vectorial surgieron varias opciones: Illustrator, que he usado en algunas asignaturas de la carrera y manejo con cierta soltura; CorelDraw, otro programa de pago de gran prestigio y nivel, y finalmente Inkscape, un programa gratuito de código abierto poco conocido, ya que muchos lo consideran inferior a sus contrapartidas ya mencionadas.

Hice una pequeña comparación para ver qué programa me iba a convenir más:

Tabla 1. Comparación programas de vectorizado.

Programa	Precio	Sistema operativo	Requisitos de hardware	Desarrollo
Inkscape	Gratis	Linux / Windows / Mac	Bajo	Abierto
Illustrator	211€/año	Windows / Mac	Alto	Privado
CorelDraw	355€	Windows	Medio	Privado

Ya daba por elegido a Illustrator, al ser el programa que ya conocía y sabía usar gracias a las asignaturas de Diseño, hasta que uno de los últimos resultados de búsqueda me llevó al vídeo Knot Effect in Inkscape. En este vídeo de casi hace 10 años enseñan en pocos minutos una función extra de Inkscape que no tiene ningún otro programa de vectorizado, y es que es capaz de cortar las líneas de sus cruces automáticamente. Aquí vemos un nudo de Lissajous, que surge al superponer movimientos armónicos simples en direcciones perpendiculares, como pasa en el osciloscopio.

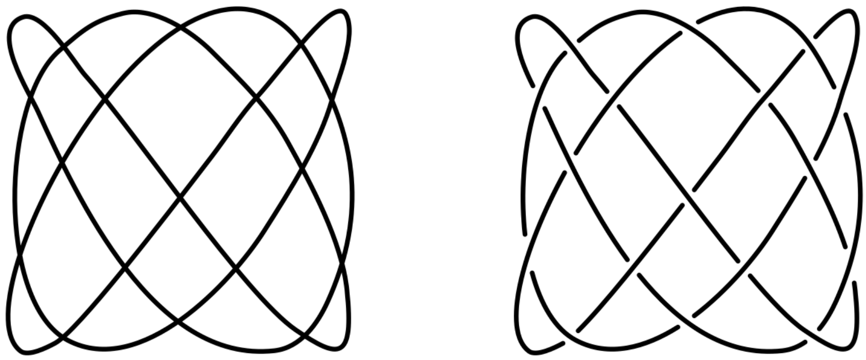


Ilustración 26. Nudo de Lissajous creado por mí con Inkscape y su herramienta Knot.

Una vez que ya sabía lo que buscar pude encontrar toda la información necesaria, y me decidí a instalar y probar este programa que, pese a tener una interfaz menos trabajada y llamativa que la de Illustrator, no tiene nada que envidiar en funcionalidad.

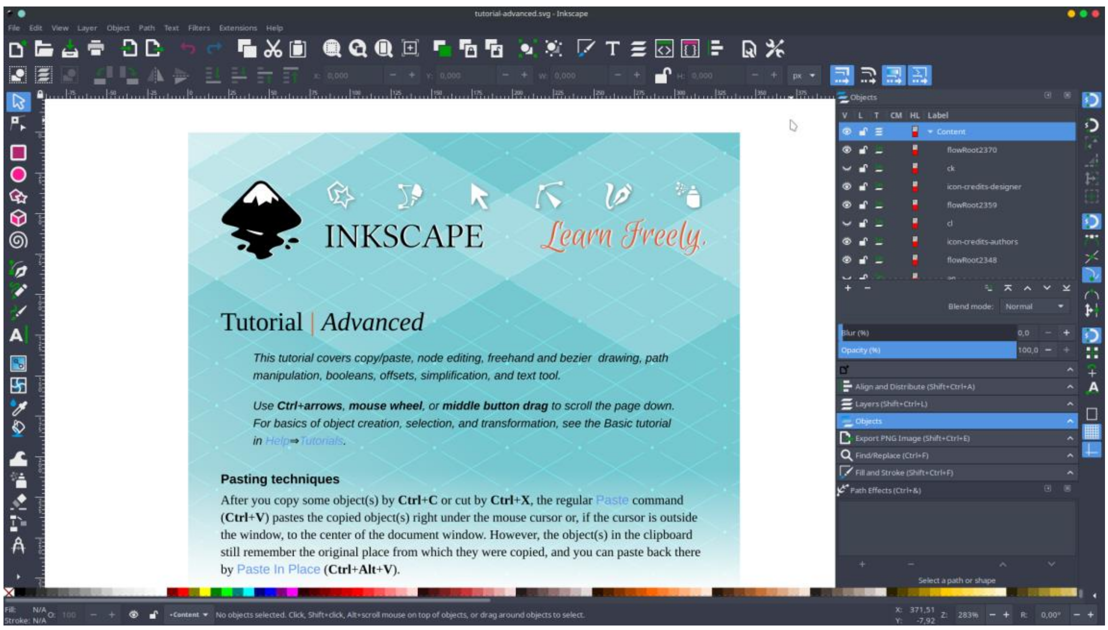


Ilustración 27. Interfaz de Inkscape (1.1-dev).

2.1 Manual de dibujo de nudos con Inkscape

Ahora podemos empezar a *dibujar libremente*, tal y como dice el lema del programa. Voy a dar los pasos para hacer un diagrama decente de un enlace pretzel cualquiera.

1. Primero se hacen las columnas. Para ello se usa la herramienta Bézier (letra B desde el teclado) y se hace un zigzag buscando el número de cruces de la columna mayor de nuestro nudo, en este caso, 8. Salimos de la herramienta Bézier pulsando Enter, seleccionamos el trazo realizado, lo duplicamos con CTRL + D y pulsamos H para que nuestro duplicado se invierta horizontalmente, obteniendo así una imagen espejo. Por último, seleccionamos los dos trazos manteniendo pulsado SHIFT y seguidamente pulsamos CTRL + K para combinarlos en una única entidad.

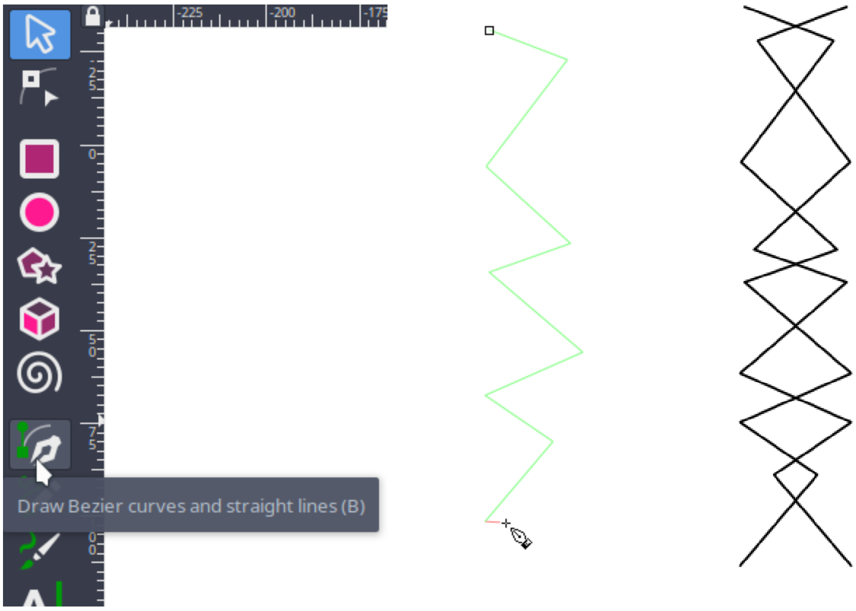


Ilustración 28. Creación de la primera columna de cruces.

2. Ahora vamos a alinear los nodos (las esquinas que hemos creado con la herramienta Bézier) para hacerlos más agradable visualmente. Para ello, teniendo seleccionado el trazo, pulsaremos N para entrar en modo de edición de nodos. Seguidamente pulsamos CTRL + SHIFT + A para abrir el panel de alineación.

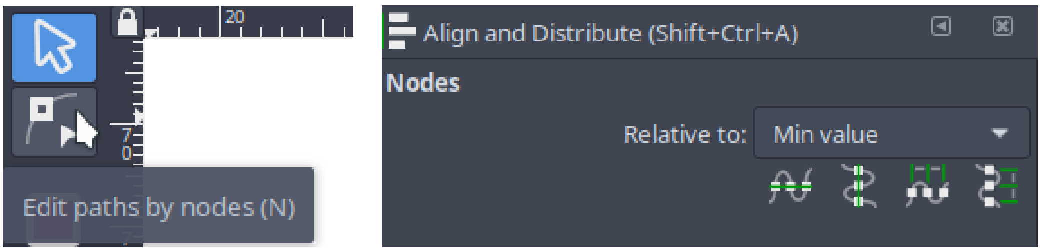


Ilustración 29. Entrada en el modo de edición de nodos y panel de alineación.

Una vez tengamos localizado el panel vamos a seleccionar todos los nodos de un lado con SHIFT y hacemos clic en alinear verticalmente y distribuir verticalmente. Cuando lo hagamos con las dos columnas veremos que ahora es más agradable visualmente.

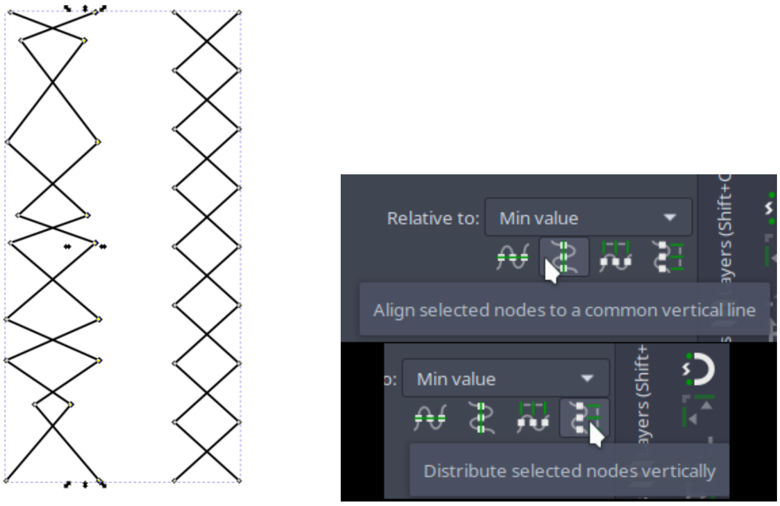


Ilustración 30. Columna con los nodos alineados y distribuidos.

3. El siguiente paso es, en el modo normal (F1), duplicar nuestra columna con CTRL + D tantas veces como columnas necesitemos y desplazarlas lateralmente manteniendo CTRL pulsado. En las columnas en las que sobren cruces los borraremos entrando en modo nodos (N), seleccionándolos y pulsando la tecla suprimir.

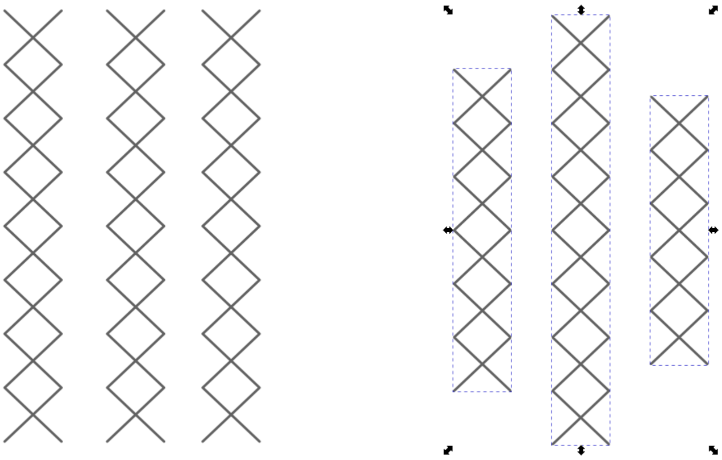


Ilustración 31. Distribución de las columnas.

4. Seguidamente combinaremos las columnas con CTRL + K, entraremos en modo edición de nodos con N y seleccionaremos los extremos que tenemos que unir. Luego pulsaremos el botón de crear uniones. Para que la unión sea más suave crearemos un punto intermedio pinchando dos veces en la línea creada y lo situaremos.

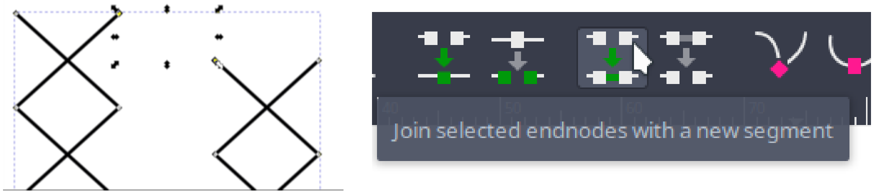


Ilustración 32. Unión de nodos.

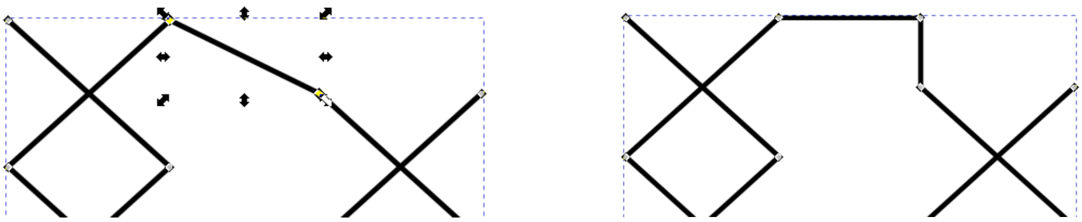


Ilustración 33. Unión de nodos.

5. Realizamos la misma operación con todos los extremos que hay que unir. A continuación, estando en el modo de edición de nodos seleccionamos todos los nodos con CTRL + A y los suavizamos con SHIFT + A.

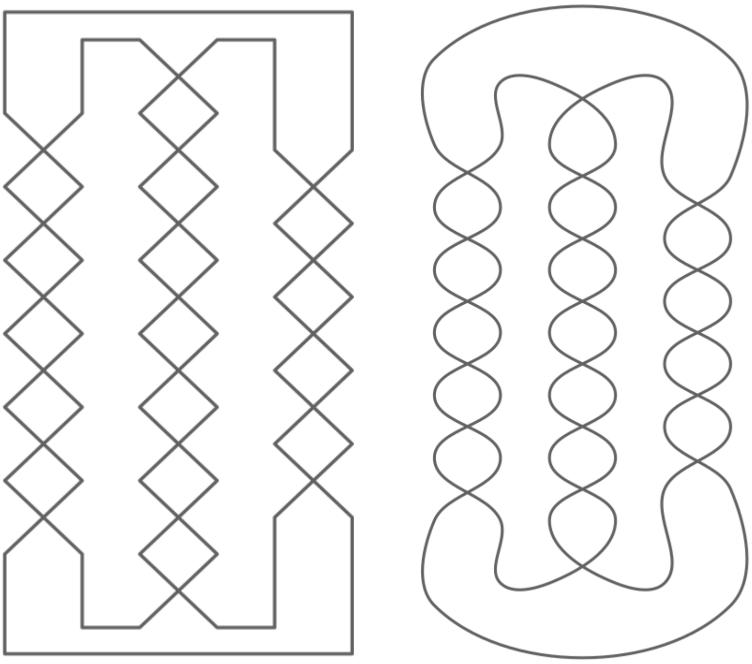


Ilustración 34. Suavizado de los nodos.

6. Ahora llega la hora del componente estrella de Inkscape, los Live Path Effects (LPE). En nuestro caso, buscamos el LPE Knot. Nos dará nuevas opciones en el menú de efectos, lo que nos permitirá cambiar, entre otras cosas, el espacio que se deja en los cruces.

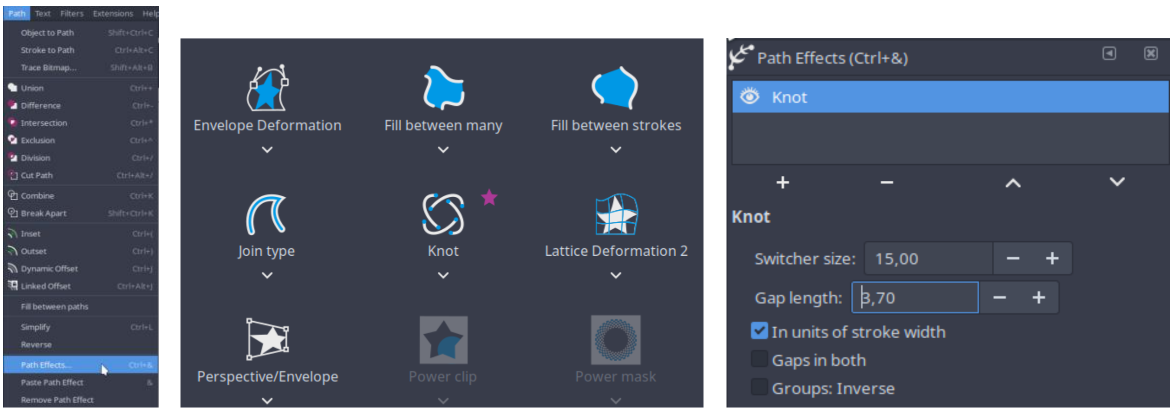


Ilustración 35. Menú de efecto Knot de Inkscape.

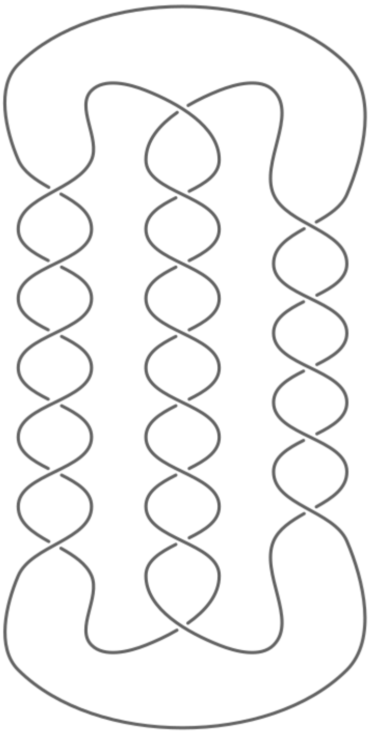


Ilustración 36. Enlace pretzel P(6,-8,-5).

Ya tenemos nuestro nudo, pero hay un pequeño problema. Puede que los cruces generados tengan por encima la cuerda que queremos que vaya debajo, ya que los ha asignado aleatoriamente. Esto se soluciona entrando en el modo edición de nodos. Si nos fijamos, en un cruce tendremos una manecilla que podemos transportar entre cruces. Cuando la tengamos en un cruce, haciendo clic cambiaremos el sentido de este cruce. Tendremos que hacer esto tantas veces como sea necesario.

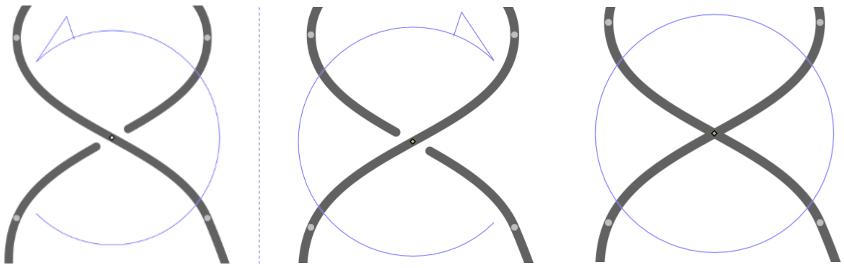


Ilustración 37. Manecilla de los cruces y su uso.

La inmensa mayoría de las ilustraciones que se encontrará el lector en este TFG son de elaboración propia excepto la demostración del teorema de Alexander de Adams, la tabla de nudos y las representaciones creadas con fiziko.

Como no es oro todo lo que reluce, ahora voy a hablar de dos errores graves que han dificultado en cierta medida (el primero poco y el segundo bastante) nuestro trabajo con los nudos cuando los representábamos con Inkscape.

2.2 Bugs

Hay dos bugs que nos hemos encontrado en esta funcionalidad de Inkscape. El primero hace que la manilla de cambiar la cuerda superior de los cruces no funcione.

Esto en un principio me chocó bastante, ya que ocurre con la versión 1.0, que se ha publicado el 1 de mayo de 2020 [11]. Hasta ahora estaba usando la versión 0.92, que salió en 2017 y funciona perfectamente. Tras hacer un reporte de problema en el GitLab público de Inkscape (https://gitlab.com/inkscape/inbox/-/issues/2752) pude saber que es una regresión en esta nueva versión, pero que está solucionado en la versión en desarrollo 1.1-dev. Por eso es esta la que se ha utilizado a partir de mayo.

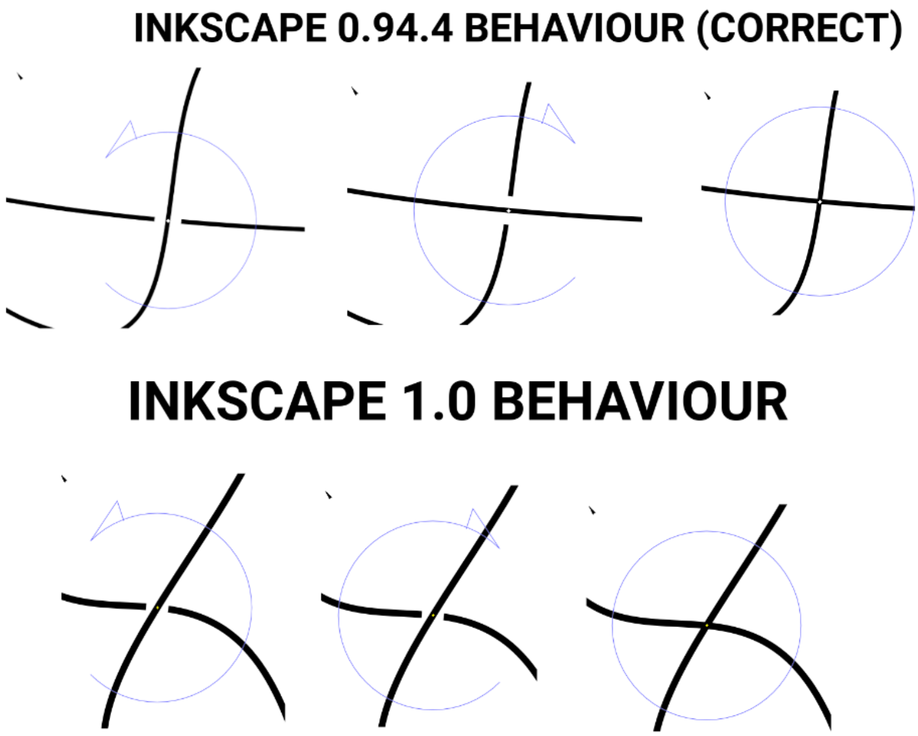


Ilustración 38. Ilustración del primer error adjuntada por mí en el informe online.

El segundo error es extremadamente peligroso. De forma un poco aleatoria (realmente bajo ciertas condiciones, como que haya un gran número de cruces) puede ser que moviendo una cuerda cualquiera se invierta un número arbitrario de cruces alrededor.

Esto ocurre cuando un objeto (en Inkscape los trazados se llaman objetos) por alguna razón ha perdido su memoria interna. Es decir, cuando un objeto se “rompe” no lo sabemos en un primer momento, ya que el síntoma (que algunos cruces se modifiquen) sólo aparecerá cuando tratemos al objeto. El problema mayor es que tratar al objeto no sólo es mover una de sus cuerdas. Puede que hayamos terminado de analizar un nudo y esté roto, pero lo hemos analizado bien. Si el objeto está roto, y lo guardamos para analizar los cruces de la trenza al día siguiente nos encontraremos con cruces cambiados y lo más probable es que no nos demos cuenta.

En un principio yo no supe darme cuenta de esto; sólo tras varias veces en las que el trabajo realizado un día no concordaba con el del día siguiente supe que estaba sucediendo algo que no debería pasar.

Otra forma de tratar a un objeto y hacer que se reproduzca el fallo es duplicarlo. Cuando duplicamos un nudo roto, su doble aparece encima, pero con algunos cruces cambiados, lo que hace que veamos cruces dobles. He tomado como costumbre después de realizar cualquier paso duplicar el objeto para comprobar su estado. Si estaba roto y ha cambiado algunos cruces, como ahora se ven dobles, sé cuáles son, y los puedo arreglar. Una vez arreglado cojo el doble y borro el original, ya que el doble es un objeto que funciona correctamente mientras que el original seguirá fallando.

Como siempre se dice que una imagen vale más que mil palabras, recomiendo encarecidamente ver mi reporte en (https://gitlab.com/inkscape/inkscape/-/issues/1546). En el primer vídeo se puede apreciar cómo ocurre el bug al mover una cuerda, y en el séptimo mensaje (https://gitlab.com/inkscape/inkscape/-/issues/1546#note_346755457) se puede apreciar el error al duplicar.

En la siguiente ilustración se muestra el error de la duplicación. Mientras que el original tiene la cuerda casi horizontal por arriba, el duplicado surge con la cuerda casi vertical por arriba.

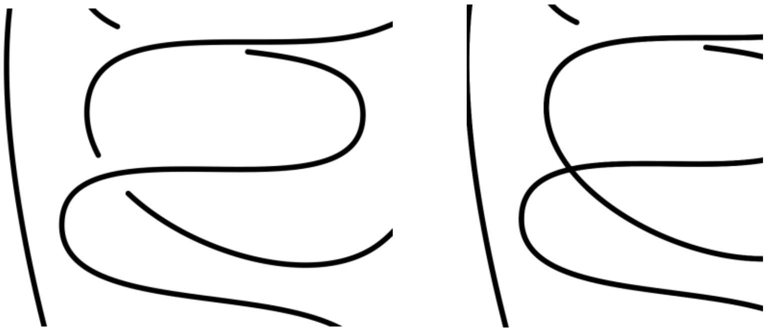


Ilustración 39. Bug 2 de Inkscape.

Lo que ocurre es que se superpone el original, en el que la cuerda casi horizontal pasa por encima, con el duplicado, en el que la cuerda casi vertical pasa por encima. Entonces se ve un cruce doble como en la ilustración.

Como última comprobación, hoy 15 de junio, este bug sigue sin estar arreglado en la versión compilada diaria, 1.1-dev (c9281a1, 2020-06-15).

3. Enlaces pretzel

Ahora tenemos que explicar un ingrediente muy importante de este trabajo, y es la familia de los enlaces pretzel. Un enlace pretzel se construye con una serie de columnas de cruces, tantas como entradas tenga el enlace. Cada columna tendrá tantos cruces como sea el valor de su entrada. Un enlace pretzel se presenta de la forma P (a_1, a_2, …, a_n), como se ve a continuación.

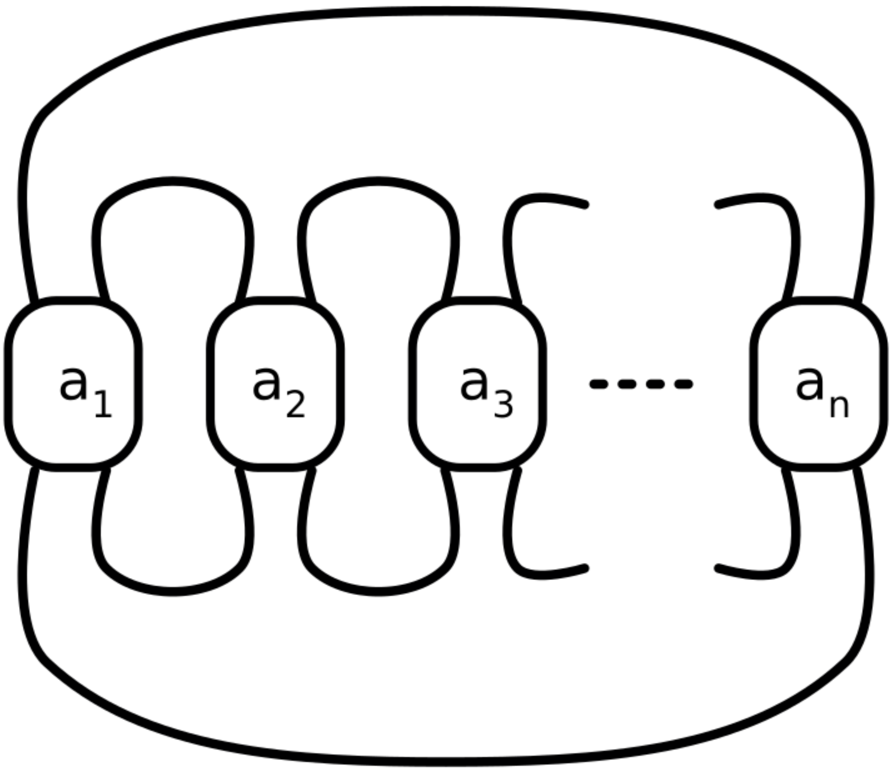


Ilustración 40. Dibujo esquemático de un enlace pretzel.

Cada caja tiene tantos cruces como indique el valor absoluto de dicha caja.

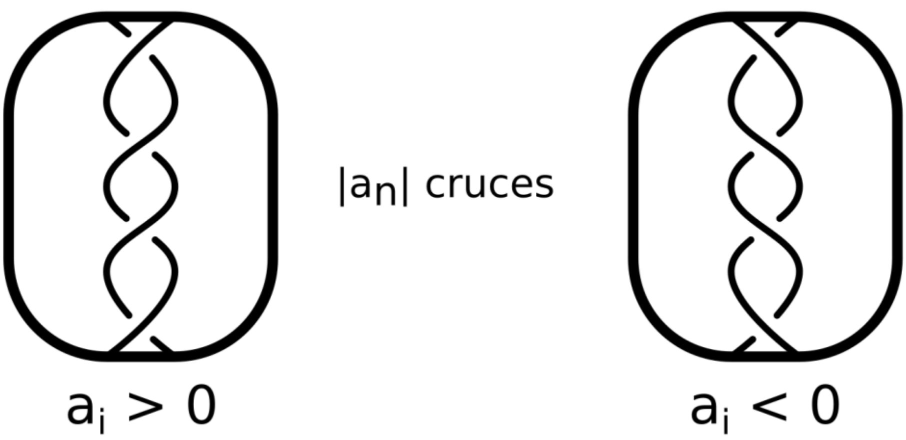


Ilustración 41. Contenido de una caja.

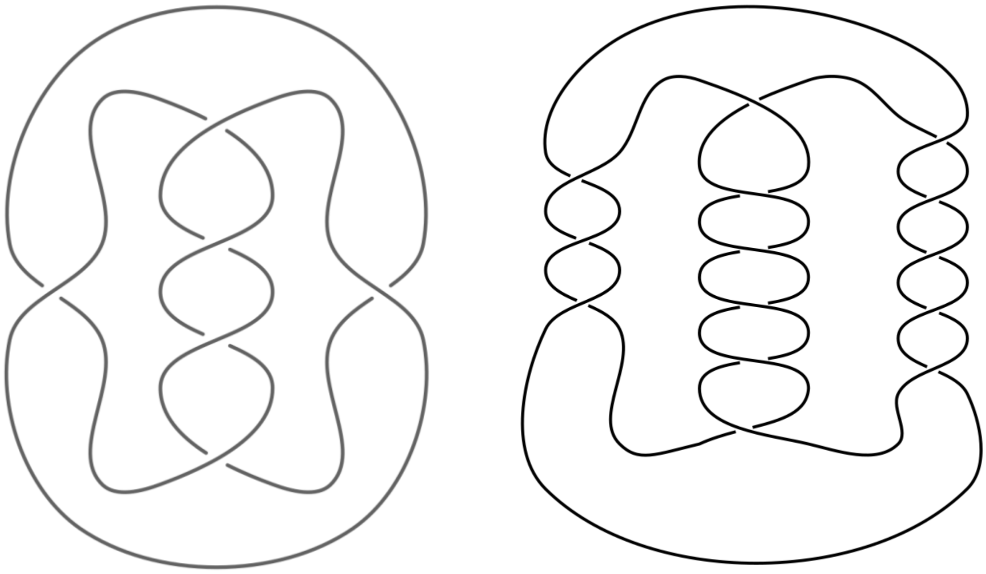


Ilustración 42. Ejemplos de enlaces pretzel: P(1, 4, -1) y P(3, -6, 5).

Un enlace pretzel es un nudo, es decir, tiene una única componente, si cumple cualquiera de las siguientes condiciones:

- Sólo tiene una única entrada par.
- Tiene un número impar de entradas y todas son impares.

Para un enlace pretzel de 3 columnas tenemos 8 posibilidades: todas pares, todas impares, una única entrada par en la primera, segunda o tercera columna o una única impar en la primera, segunda o tercera columna. Los nudos son 4 de estas posibilidades, se dan si hay una entrada par o si todas son impares:

- P(a, b, c) siendo una única entrada par.
- P(a, b, c) siendo a, b y c impares.

Si completamos los pretzel de 3 columnas con los enlaces, se dividen en:

- P(a, b, c) siendo a y c pares y b impar.
- P(a, b, c) siendo b y c pares y a impar.
- P(a, b, c) siendo a y b pares y c impar.
- P(a, b, c) siendo a, b y c pares.

La familia de los enlaces pretzel es importante como banco de pruebas para analizar invariantes de nudos. Hay razones múltiples: es una familia con una cantidad infinita de nudos diferentes, que pueden dibujarse fácilmente y definirse con una simple secuencia de números enteros. Además, ofrece una serie de comportamientos matemáticos muy interesante. Por ejemplo, mientras hay infinitos nudos pretzel con el mismo polinomio de Conway, no ocurre tal cosa si hablamos del polinomio de Jones [12].

3.1 Propiedades

Para tratar los enlaces pretzel, y con una importancia muy especial en la síntesis de los pretzel de 3 columnas que se hace en el capítulo V, vamos a introducir dos propiedades que nos dan mayor flexibilidad al tratar dichos enlaces.

Propiedad simétrica

P (a, b, c) = P (c, b, a). Lo mismo se puede decir para cualquier número de entradas, independientemente de su valor. Esto se puede observar fácilmente volteando esta hoja (o esta pantalla, probablemente) para observar que en la Ilustración 43 tenemos P (7, 6, 10) o P (10, 6, 7).

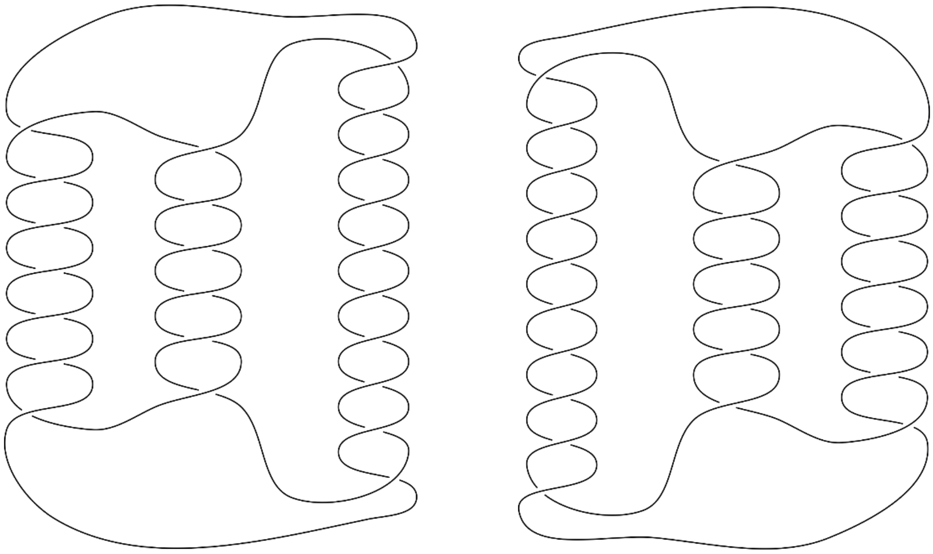


Ilustración 43. P(7, 6, 10) o, de forma simétrica, P(10, 6, 7).

Propiedad cíclica

Se pueden ver los enlaces pretzel de forma circular, es decir, desplazar todas las entradas a izquierda o derecha tantas veces como se quiera. Esto se debe a que podemos verlos de forma circular, y una vez de esta manera, girarlos y volver a verlos en su forma normal.

Es decir, P(a, b, c) = P(b, c, a) = P(c, a, b).

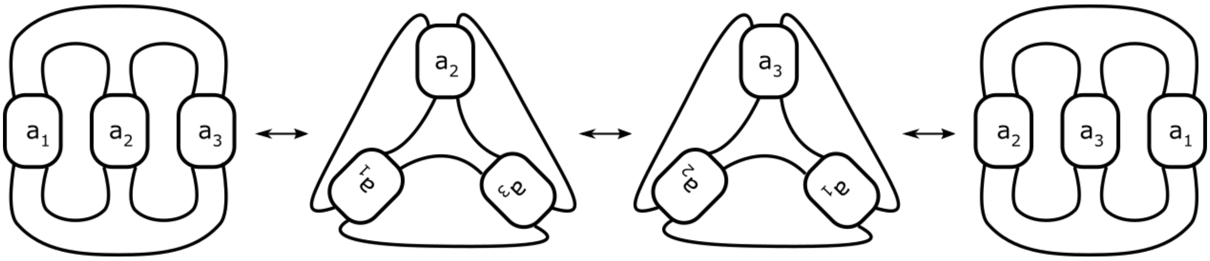


Ilustración 44. Propiedad cíclica de los enlaces pretzel.

4. Trenzas

Una trenza es un conjunto de *n* cuerdas unidas a dos barras, una superior y otra inferior. Estas cuerdas caen monótonamente hacia abajo, sin realizar bucles sobre sí mismas, y cada cuerda atraviesa cualquier plano horizontal una única vez. Aquí vemos un ejemplo de un diagrama de trenza válido y uno inválido.

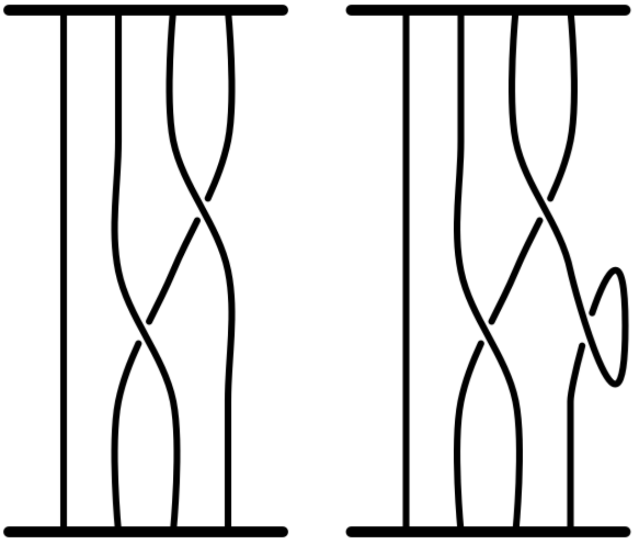


Ilustración 45. Un diagrama de trenza correcto y otro incorrecto.

Ahora bien, ¿qué relación tienen las trenzas y los nudos? Pues resulta que, si clausuramos (cerramos) los extremos de una trenza, tendremos un enlace.

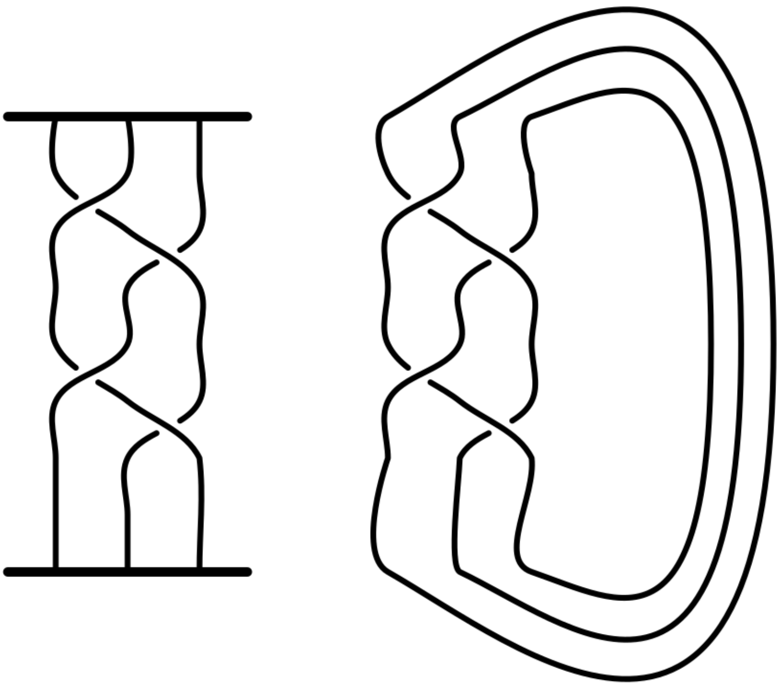


Ilustración 46. Una trenza y su clausura.

Para referirnos a una trenza usaremos la **palabra de la trenza**, que es la manera de expresarla sin tener que dibujarla. Para expresarla tenemos que tener en cuenta qué cuerdas se están cruzando, y cuál es la que va por encima. Cualquier trenza se puede obtener concatenando los generadores σ_i y sus inversos, donde *i* = 1, 2,…, n-1.

Ilustración 47. Generadores del grupo de las trenzas.

Si se cruza la primera cuerda con la segunda tenemos $\sigma_1^{\pm 1}$, si se cruza la segunda con la tercera tenemos $\sigma_2^{\pm 1}$, si es la tercera con la cuarta $\sigma_3^{\pm 1}$, y así sucesivamente. Por ejemplo, la trenza anterior se expresa con la palabra

$$\sigma_1\sigma_2^{-1}\sigma_1^{+1}\sigma_2^{-1}$$

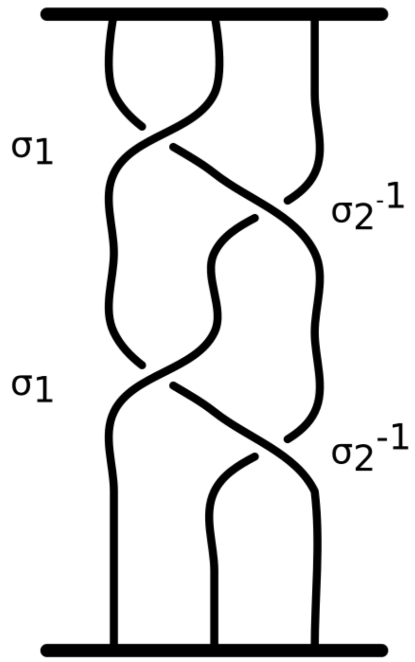


Ilustración 48. Diagrama de trenza con sus cruces.

Más adelante, en pos de la sencillez, se omitirá el uso de σ y se expresará la palabra de la trenza como

1, -2, 1, -2

Ahora voy a dar tres posibles movimientos (análogos a los movimientos de Reidemeister) que podemos hacer en una trenza, tanto en su diagrama como en su palabra.

1) Teniendo la palabra $\sigma_k\sigma_k^{-1}$podemos obviar este enredo local.

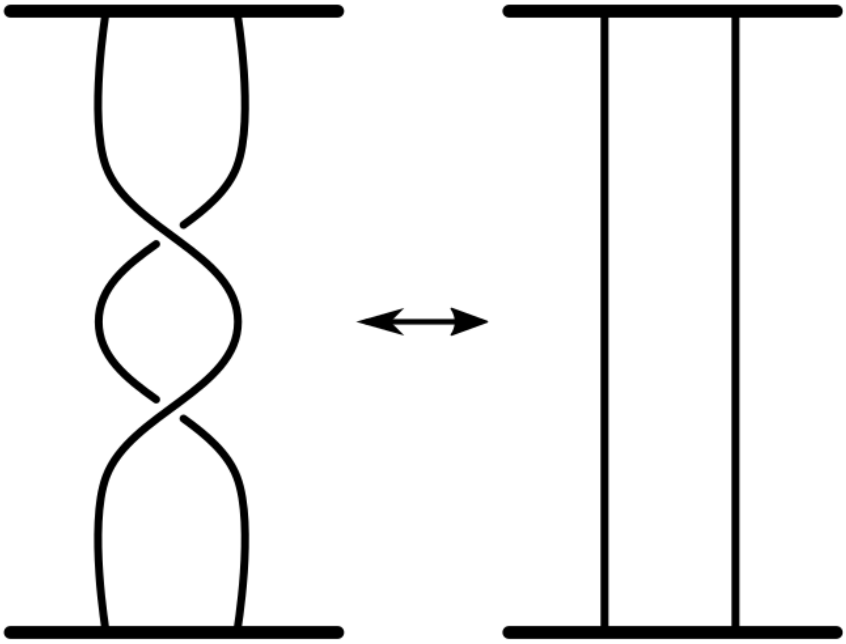


Ilustración 49. Primer movimiento en las trenzas.

Por ejemplo, $\sigma_3^{-1}\sigma_2^{-1}\sigma_1^{-1}\sigma_1\sigma_2\sigma_3$se nos quedaría en *nada*, la trenza trivial.

2) $\sigma_i^{\pm 1}\sigma_j^{\pm 1}\sigma_i^{\pm 1} = \sigma_j^{\pm 1}\sigma_i^{\pm 1}\sigma_j^{\pm 1}$.

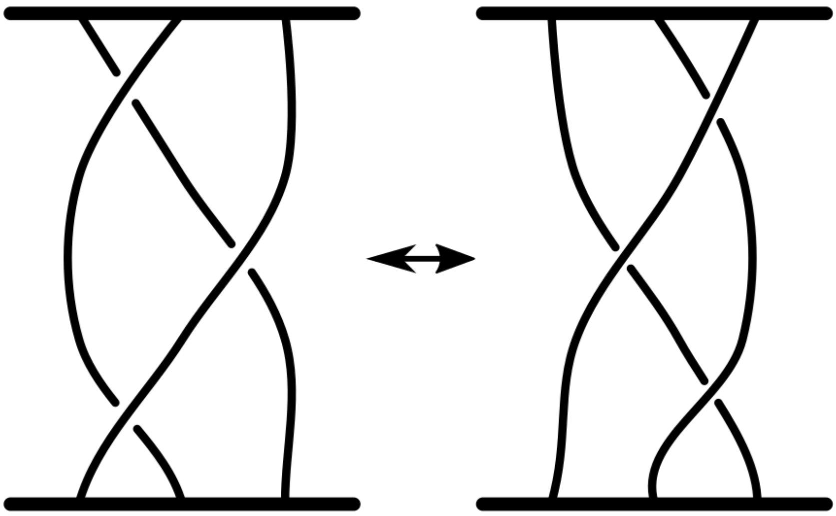


Ilustración 50. Segundo movimiento en las trenzas.

Si el primer movimiento claramente se basa en el movimiento de Reidemeister de tipo 2, este se basa en el movimiento de Reidemeister de tipo 3.

3) $\sigma_i^{\pm 1}\sigma_j^{\pm 1} = \sigma_j^{\pm 1}\sigma_i^{\pm 1}$ si |i-j|>1.

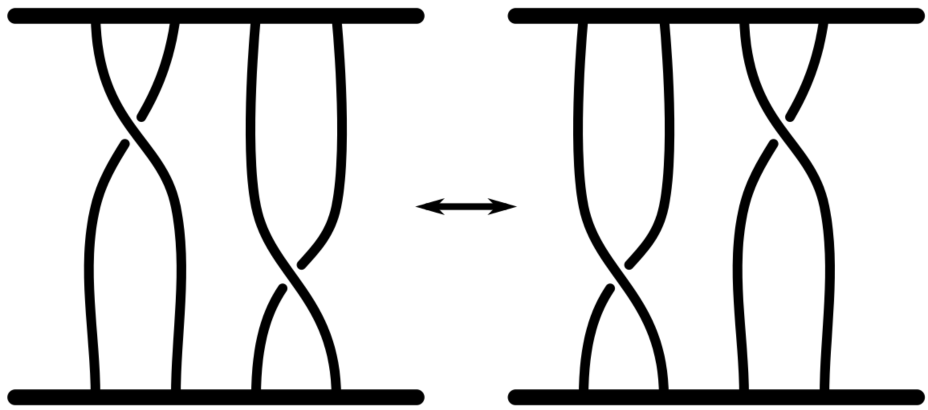


Ilustración 51. Tercer movimiento en las trenzas.

Dos trenzas, ambas con *n* cuerdas, b y b', pueden multiplicarse para dar una nueva trenza que se llamaría bb'.

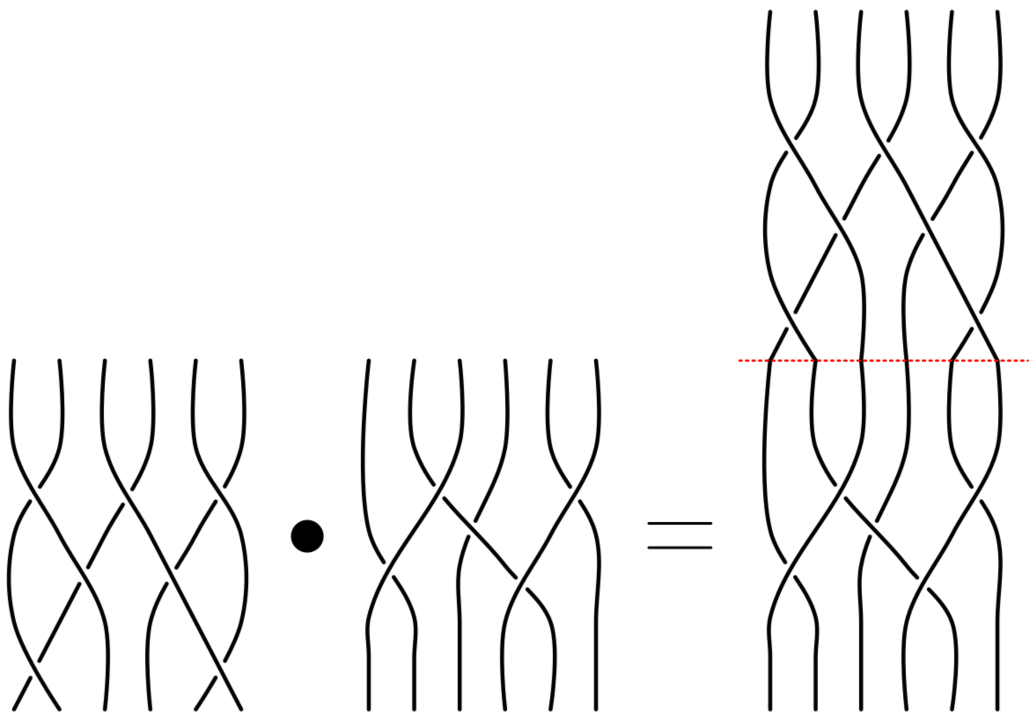


Ilustración 52. Producto de dos trenzas.

- Existe una trenza (la trenza trivial o trenza unidad) tal que si multiplicamos cualquier trenza por esta nos dará la trenza inicial.
- Cada trenza tiene su inversa, y al multiplicarse, dan la trenza unidad del apartado previo (esto no ocurre en la composición de nudos).
- Esta multiplicación es asociativa.

Por todo esto podemos hablar del grupo B_n de trenzas de *n* cuerdas.

4.1 Teorema de Alexander

Hemos visto que al clausurar una trenza aparece un enlace. Pero, ¿puede obtenerse cualquier enlace como clausura de una trenza? Este problema fue resuelto por James Waddell Alexander II en 1923 [13]:

Teorema de Alexander

"Todo enlace es la clausura de una trenza".

Colin Adams reproduce una demostración del teorema de Alexander bastante buena en The Knot Book, usando el número de puentes. Para ello vamos a recorrer el nudo y a marcar cada hebra entre dos cruces inferiores. Todas estas hebras pueden ser cortadas por un plano, así que nos quedan algunos "puentes" por la cara vista y otros por la cara oculta. Los puentes de una cara no se cortan entre ellos, tal y como vemos en la figura punteada.

Ahora vamos a distribuirlos de tal forma que por un lado sean paralelos (en nuestro ejemplo los puentes de la cara oculta, como se ve en la tercera ilustración) y por el otro adaptamos sus puentes de forma que sigan sin tocarse. Ahora podemos imaginar un eje que pertenece al plano y pasa entre de todos los puentes. Los de la cara oculta son todos arcos iguales paralelos, mientras que los de la cara vista tienen ciertas curvas, pero siguen yendo de una orilla a la otra.

Si damos grosor a este eje tendríamos los puentes anudados alrededor de un tubo, así que ya tenemos la trenza cerrada (véase [8] para más detalles).

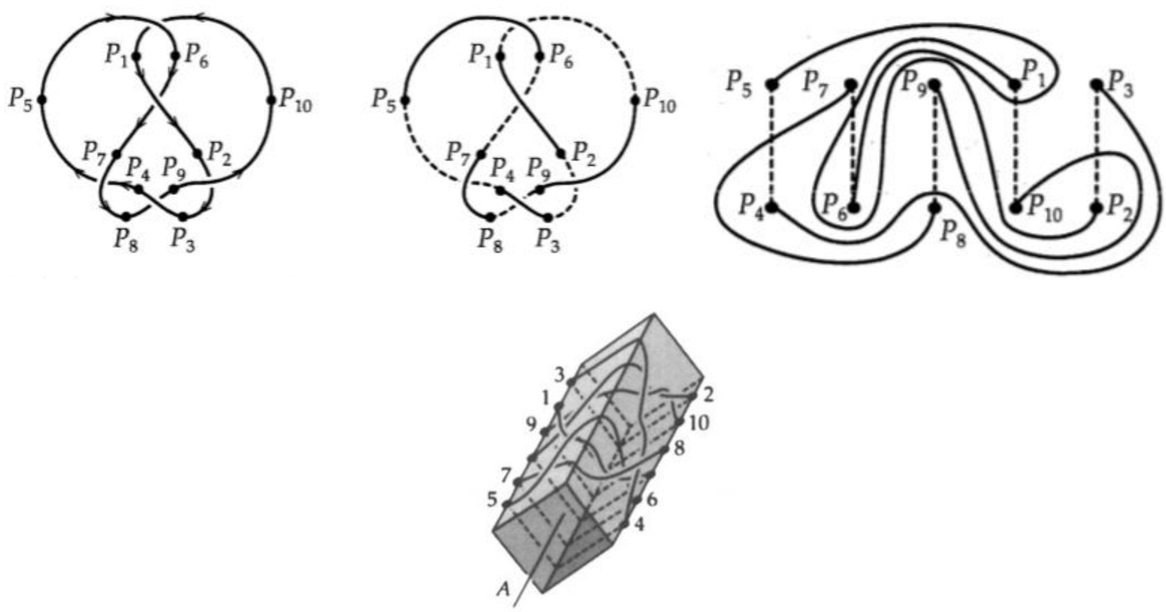


Ilustración 53. Demostración del teorema de Alexander [8].

4.2 Círculos de Seifert. Compatibilidad. Complejidad

El objetivo de este TFG es encontrar para cada enlace pretzel una trenza. La demostración anterior del teorema de Alexander no tiene control de la trenza obtenida. Por ello vamos a explicar otra demostración más constructiva que en principio pensamos que podría ser de gran ayuda. Está tomada del libro de Peter Cromwell [14].

Empezamos orientando el nudo. A continuación, dibujamos lo que llamaremos sus círculos de Seifert. Para ello, nos colocamos en punto del diagrama y seguimos su orientación hasta llegar a un cruce.

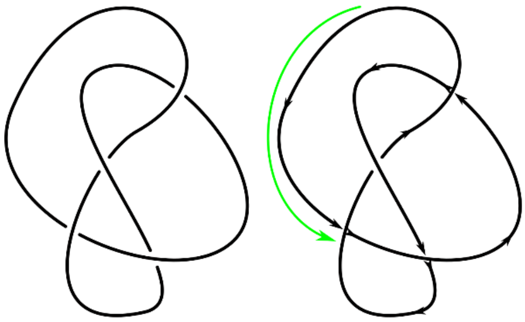


Ilustración 54. Inicio de un círculo de Seifert.

Cuando lleguemos a este cruce vamos a seguir la otra cuerda en el sentido de su orientación. En nuestro caso, hacia arriba. Repetiremos esto tantas veces como haga falta hasta llegar al punto inicial.

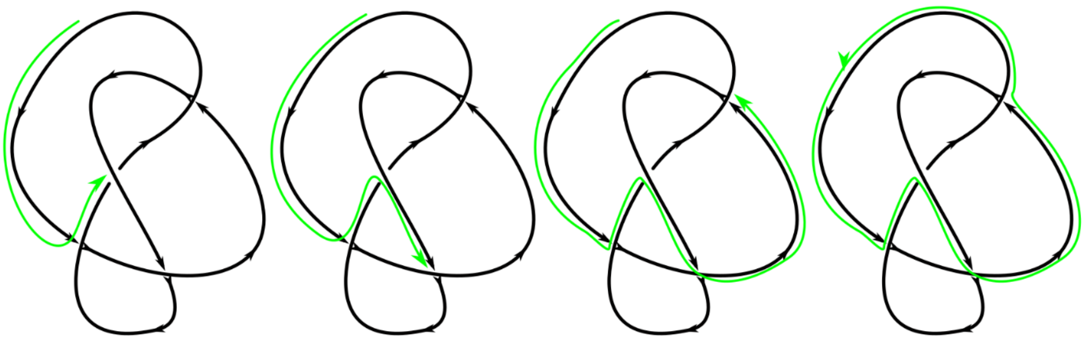


Ilustración 55. Círculo de Seifert completo.

En este ejemplo obtenemos 3 círculos de Seifert:

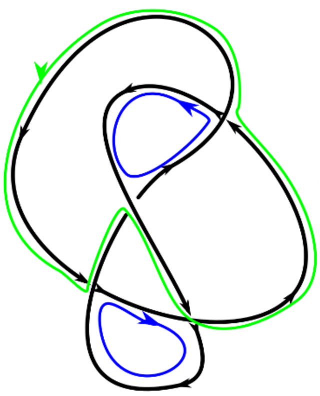


Ilustración 56. Nudo y sus círculos de Seifert.

Ahora vamos a introducir el concepto de compatibilidad y complejidad. Un par de círculos de círculos de Seifert se dice incompatible si cuando hay *contacto* entre ellos, las flechas tienen sentido contrario.

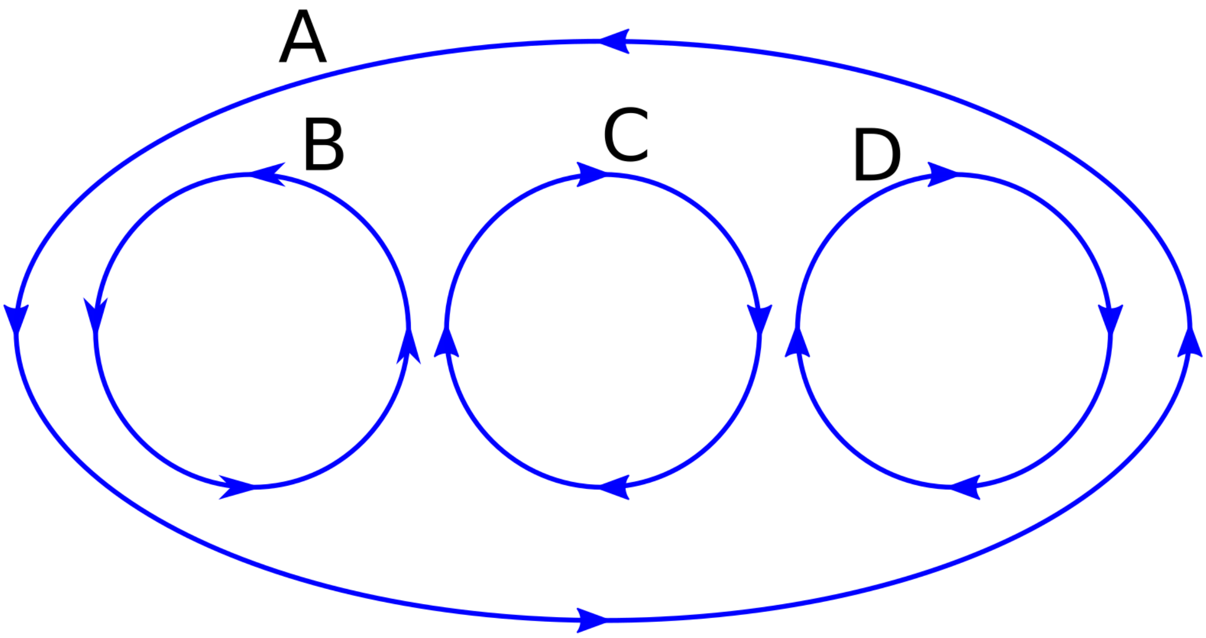


Ilustración 57 Los pares A-B, B-C y B-D son compatibles. Los pares A-C, A-D y C-D son incompatibles.

Podemos pensarlo como si fuesen engranajes. Dos engranajes exteriores tienen que girar en sentido contrario, si son interiores deben girar en el mismo sentido. En caso contrario se romperían; serían incompatibles. Ahora podemos calcular el grado de complejidad, que es el número de pares de círculos incompatibles. Si volvemos a la Ilustración 56:

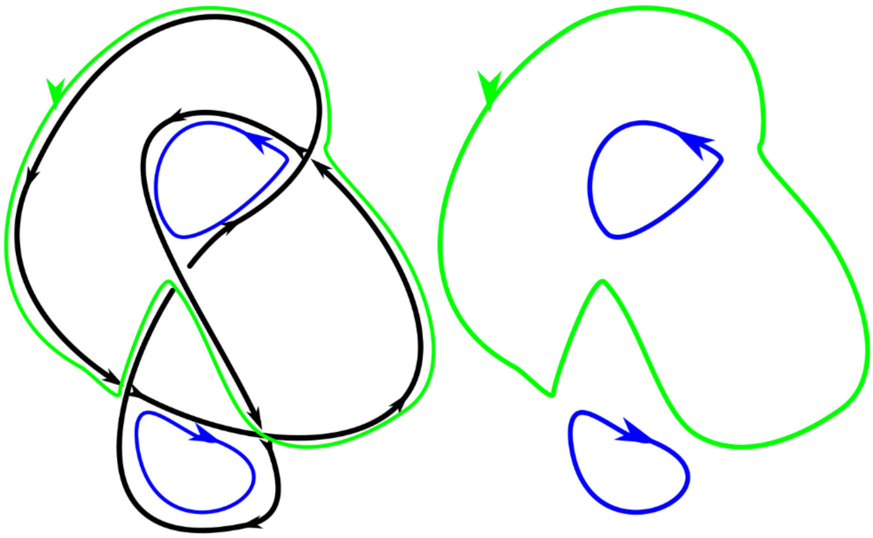


Ilustración 58. Este diagrama tiene complejidad 0.

En este caso cualquier pareja de círculos es compatible, por lo que su complejidad es 0. En cambio, en la Ilustración 57 tenemos complejidad 3.

Una de las observaciones claves es que la complejidad del diagrama de una trenza es 0, es decir, todos sus círculos son compatibles. Y si visualizamos la trenza en una esfera esto caracteriza los diagramas de trenza. Volvamos a ver el diagrama de la trenza $\sigma_1\sigma_2^{-1}\sigma_1\sigma_2^{-1}$. Una vez la clausuramos dibujamos sus círculos de Seifert.

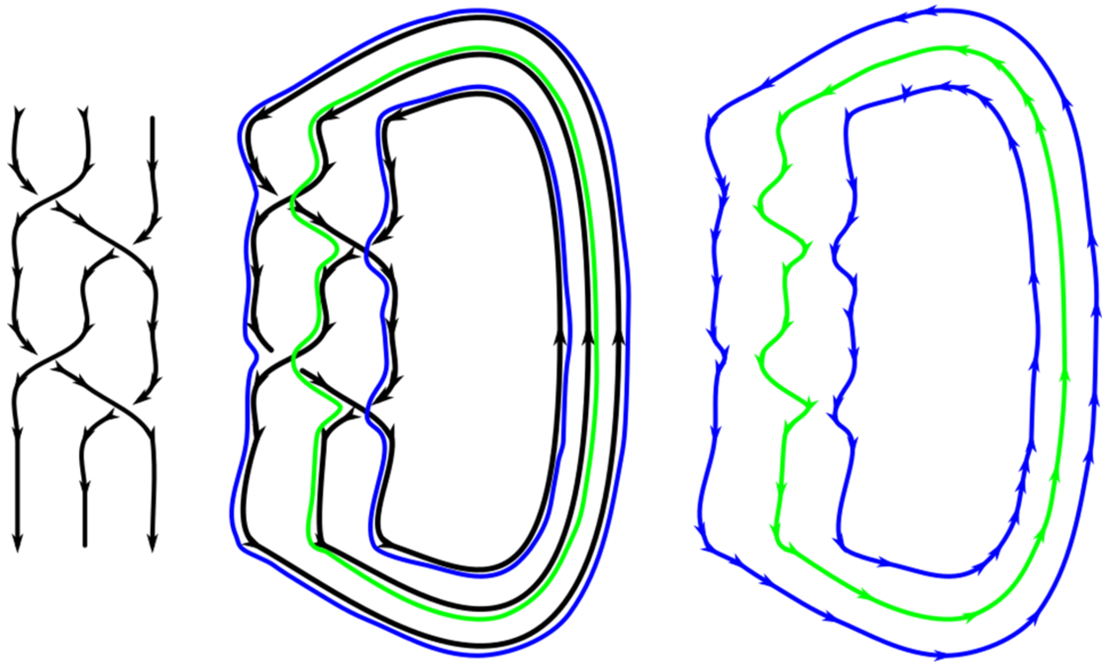


Ilustración 59. Trenza, su clausura y sus círculos de Seifert.

Es decir que, teniendo un diagrama cualquiera, si lo queremos pasar a trenza, tenemos que hacer una serie de manipulaciones que nos vayan bajando la complejidad hasta que llegue a 0.

Esto se logra mediante **movimientos de reducción**. Veamos lo que ocurre localmente con dos arcos de círculos incompatibles.

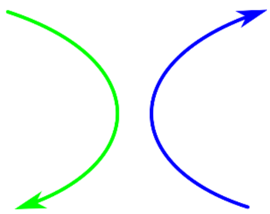


Ilustración 60. Arcos incompatibles.

La solución es superponer un arco encima de otro, tal y como se hacía en el movimiento de Reidemeister de tipo 2. De esta forma se nos formará un nuevo círculo de Seifert en el centro y otro exterior, siendo ambos compatibles. El arco que pasemos por encima o por debajo depende de nuestra preferencia, pero siempre que he podido he pasado el izquierdo por encima del derecho.

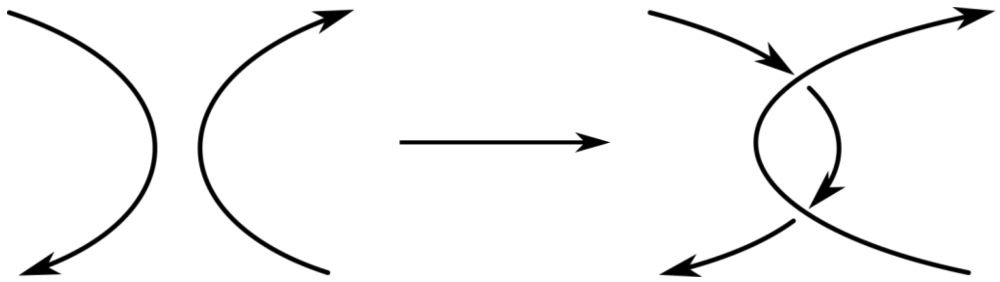


Ilustración 61. Movimiento de reducción.

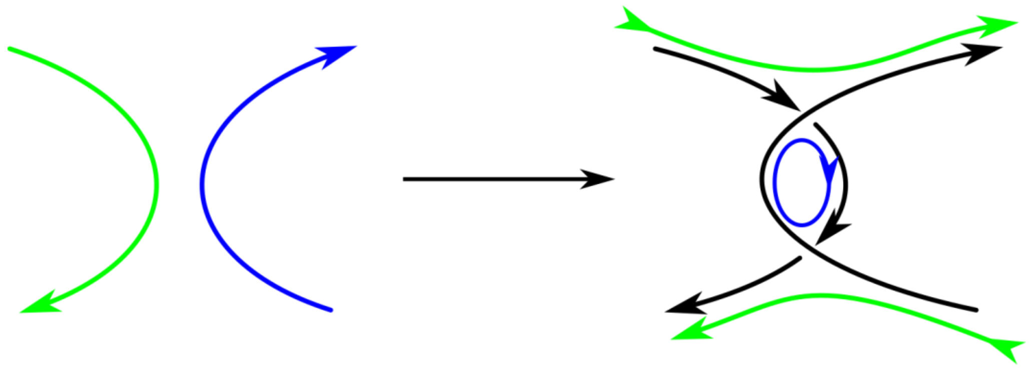


Ilustración 62. Influencia del movimiento de reducción en los círculos de Seifert.

Un movimiento de reducción no modifica el número de círculos de Seifert, pero reduce la complejidad en una unidad. Es decir, si tenemos un nudo de complejidad 3 tendremos que hacer 3 movimientos de reducción para obtener una trenza.

4.3 Teorema de Markov

Ya hemos visto que podemos representar cualquier enlace mediante una trenza. Desde luego la correspondencia no es uno a uno, un nudo puede tener varias representaciones de trenza, como veremos por ejemplo en la resolución de enlaces pretzel de cuatro entradas.

Esto nos genera una incógnita extra: ¿cómo sabemos si dos trenzas representan al mismo nudo?

Teorema de Markov

Las clausuras de dos trenzas definen el mismo enlace sí y sólo sí puede pasarse de una trenza a la otra por una secuencia finita de los llamados movimientos de Markov [15]*:*

- (Des)estabilización: $b \leftrightarrow b\sigma_n^{\pm 1}$ $b \in B_n$, $b\sigma_n \in B_{n+1}$.

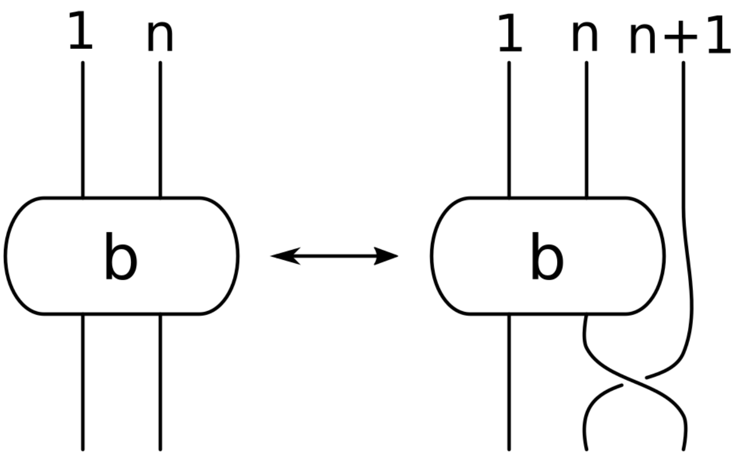


Ilustración 63. (Des)estabilización.

- Conjugación $b \leftrightarrow \sigma_i b \sigma_i^{-1}$ ó $\sigma_i^{-1} b \sigma_i$.

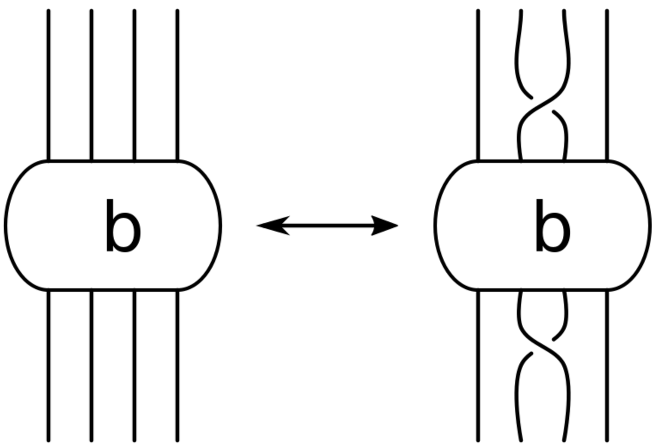


Ilustración 64. Conjugación.

Como el lector apreciará, este teorema es análogo al teorema de Reidemeister, pero aplicado a trenzas.

Por otra parte, sabiendo que diferentes trenzas pueden representar al mismo nudo, surge una pregunta: ¿tenemos la representación con el menor número de cuerdas? Así llegamos al concepto de *índice de trenza*, que es el menor número de cuerdas necesario para representar un nudo como clausura de una trenza.

El índice de trenza es un invariante de los nudos muy difícil de computar. Una vez obtenemos una trenza de un nudo sabemos una cota superior para el índice, pero puede resultar muy complicado saber si hay otra trenza con menos cuerdas.

Shuji Yamada ha demostrado que el índice de trenza coincide con el menor número de círculos de Seifert en la proyección de un nudo. Para el nudo trivial obviamente vale 1, para el nudo trébol vale 2 y para el nudo de la figura 8 vale 3 ([16], [8]).

4.4 Notación para trenzas

Para simplificar la notación más adelante, vamos a establecer ciertas abreviaturas.

$$\sigma_2\ \sigma_2\ \sigma_2 = \sigma_2^{3}$$
$$\sigma_1^{-1}\ \sigma_1^{-1}\ \sigma_1^{-1} = \sigma_1^{-3}$$

Más adelante, en pos de la sencillez, se omitirán las sigmas.

$$\sigma_1\ \sigma_2 = 1,\ 2$$
$$\sigma_1\ \sigma_1\ \sigma_2^{-1}\ \sigma_2^{-1}\ \sigma_2^{-1}\ \sigma_1 = 1^2,\ 2^{-3},\ 1$$

Las trenzas resultantes en los teoremas que veremos en el capítulo V están compuestas por pequeños bloques que de aquí en adelante llamaré *agrupaciones*. Una agrupación es un conjunto de números que van en serie, o bien progresando numéricamente (1, 3, 5, …, 13) o bien repitiendo un mismo valor una cantidad de veces determinadas ($5^{|3|}$). Cuando sea necesario para mejorar la visibilidad se separarán las agrupaciones por punto y coma (;).

Las progresiones crecientes y decrecientes se señalarán con una flecha. Por ejemplo

$$3{\uparrow}8 = 3,\ 4,\ 5,\ 6,\ 7,\ 8 = \sigma_3\ \sigma_4\ \sigma_5\ \sigma_6\ \sigma_7\ \sigma_8$$

Las agrupaciones más complejas son *progresiones de agrupaciones*. Estas agrupaciones se separarán del resto por llaves {}. Ahora sí que es mejor un ejemplo.

{1, 2, 1; 5, 6, 5; 9, 10, 9; …: (7 veces)}

Es

1, 2, 1, 5, 6, 5, 9, 10, 9; 13 ,14, 13, 17, 18, 17, 21, 22, 21

En este primer tipo de progresión las veces que se repite nos indica las veces que se repite la agrupación en sí, no todo lo que tenemos dentro de las llaves. Normalmente la agrupación inicial irá acompañada de uno o dos términos más de su serie para poder observar la progresión.

El segundo tipo de progresión de agrupaciones se da cuando el límite superior es la variable inicial, y no sabemos cuántas agrupaciones van a salir. Por ejemplo:

$$\{X, X+1, X, X; X-1, X, X-1; X-2, X-1, X-2; \ldots; 1,2,1\}$$

sería, para X=5,

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

El tercer tipo de progresión se da con la agrupación inicial y la modificación que va a tener dicha agrupación. Este caso es el más complicado de seguir. Por ejemplo

$\{(-8) \uparrow (-3)^{\downarrow 4 \downarrow}\}$ = -8,-7,-6,-5,-4,-3,-9,-8,-7,-6,-5,-4,-10,-9,-8,-7,-6,-5,-11,-10,-9,-8,-7,-6

Lo que ocurre es que tenemos una progresión normal del estilo $(-8) \uparrow (-3)$ en la que el argumento inicial y final varía según la flecha del exponente, hasta tener tantas progresiones como diga el exponente. Como ambas indican hacia abajo (siempre en el mismo sentido), disminuye el valor tanto del argumento inicial como del final.

5. Trenzas para enlaces pretzel

En este capítulo se va a desarrollar el objetivo del trabajo, que es la búsqueda de una relación entre enlaces pretzel y trenzas cerradas, basándonos en que el teorema de Alexander nos dice que podemos llegar a una trenza cerrada desde cualquier enlace**.** Ahora voy a sintetizar los teoremas del capítulo para luego demostrarlos, ya que la demostración es extensa y es fácil desubicarse. También hablaremos de las herramientas necesarias para tratar los nudos un poco más adelante.

5.1 Resultados

Teorema 1

El nudo pretzel $P(a_1)$ es el nudo trivial, clausura de la trenza trivial.

Para entenderlo lo único que tenemos que hacer es coger un extremo y girarlo tantas veces como el valor de la entrada en el sentido que deshaga los cruces. Cuando terminamos nos encontramos con el nudo trivial, o lo que es lo mismo, una trenza cerrada de una cuerda y cero cruces.

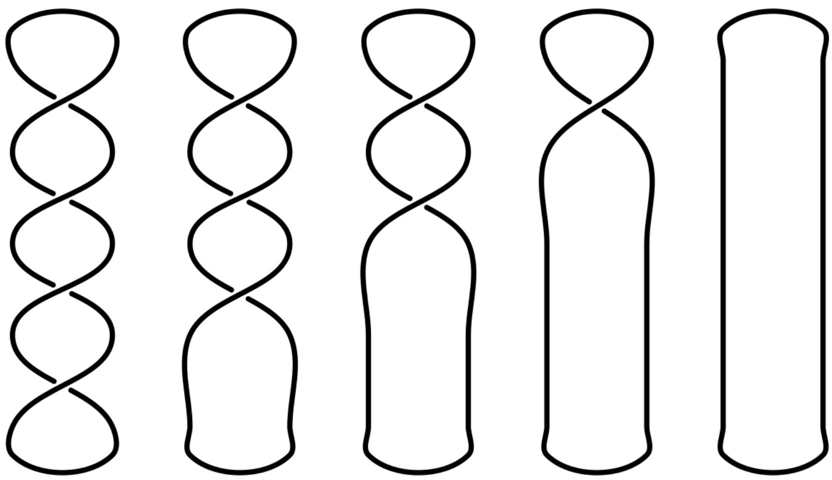


Ilustración 65. Equivalencia pretzel-trenza para $P(a_1)$.

Teorema 2

Sea P(a, b) un enlace pretzel de dos entradas. Entonces este enlace es la clausura de la trenza

$$1^{\mathrm{a+b}}.$$

Por ejemplo, al pretzel P(-2, 5) le corresponde la trenza 1^{-2+5}=1^3=$\sigma_1\sigma_1\sigma_1$.

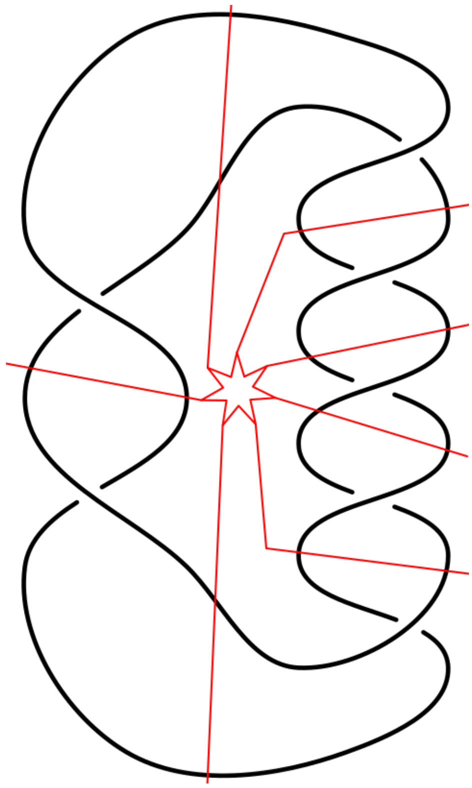


Ilustración 66. P(-2, 5) y su trenza cerrada.

Teorema 3

Sea P(a, b, c) un enlace Pretzel de tres entradas. Entonces este enlace es la clausura de la trenza:

A. Si todas las entradas son impares

$$
\begin{array}{c}
-1 \downarrow -(a'+1), -(a'+1) \uparrow -2 \\
\{(-2-a') \uparrow (-3)^{\downarrow b' \downarrow}\} \\
\{(2+b'+a') \downarrow (3+b')^{\uparrow c' \uparrow}\} \\
\{(2+b') \downarrow (3)^{\uparrow c' \uparrow}\} \\
2 \uparrow (1+c') \\
-1 \downarrow -(c'+1), -(c'+1) \uparrow -2 \\
\{(-2-c') \uparrow (-3)^{\downarrow b' \downarrow}\} \\
2 \uparrow (1+b') \\
-1 \downarrow\ -(b'+1), -(b'+1) \uparrow -2 \\
\{(-2-b'-c') \downarrow (-3-b')^{\uparrow a' \uparrow}\} \\
\{(2+b') \downarrow (3)^{\uparrow a' \uparrow}\} \\
2 \uparrow (1+a')
\end{array}
$$

donde

$$a' = \frac{|a|-1}{2},\ \ b' = \frac{|b|-1}{2},\ \ c' = \frac{|c|-1}{2}.$$

B. En cualquiera de los otros casos (la presentación circular permite asumir que b es par, y tomando orientaciones de forma que en dicha columna central sean opuestas):

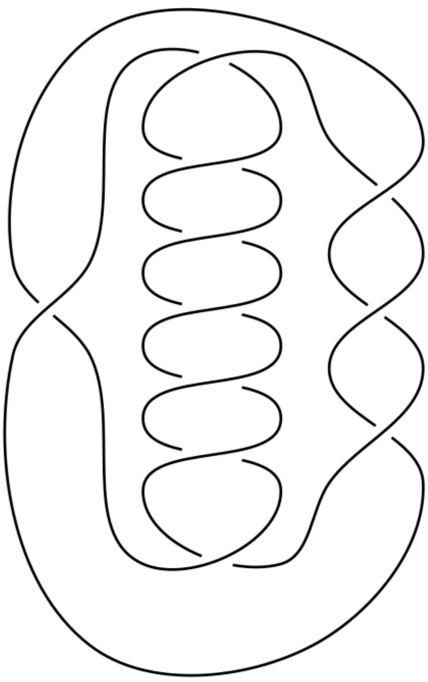
$$-2 \downarrow -\frac{|b|+2}{2}, 1^a, 2 \uparrow \frac{|b|}{2}, -\frac{|b|+2}{2} \uparrow -2, 1^c$$

La demostración del teorema 3 toma más de 60 páginas, cosa a tener en cuenta antes de enfrentarse a su lectura. Concretamente, la variante P(impar, impar, impar) requiere 32 de estas 60 páginas, por lo que es fácil perderse dentro de ella. Los apartados se han enumerado en orden de dificultad creciente. Se comienza por los nudos, ya que era el objetivo inicial de TFG, y después se analizan los enlaces.

Al terminar, se hace una síntesis y se llega a un nuevo método más rápido y directo que permite englobar todos los casos excepto aquel en el que todas las entradas son impares. Si el lector prefiere no leer toda la casuística previa, puede dirigirse directamente a la subsección “Enlaces pretzel con tres entradas. Síntesis”, en la página 129.

Como ejemplo, el nudo pretzel P(1, 7, 3) es la clausura de la trenza

-1, 5, 4, 3, 2, -1, -2, -2, -3, -4, -5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2



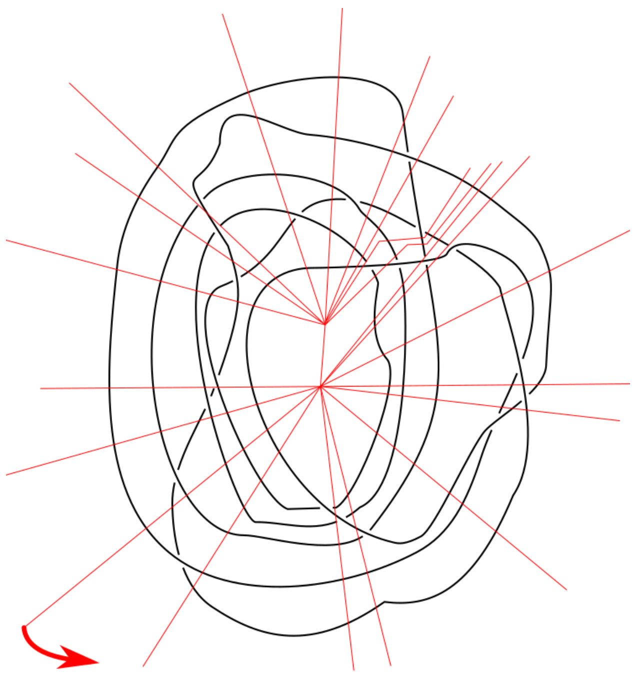


Ilustración 67. Nudo P(1, 7, 3) y la clausura de la trenza correspondiente.

Teorema 4

Sea P(a, b, c, d) un enlace pretzel de 4 entradas. Entonces este enlace es la clausura de la trenza

$$-3, 2^b, -1, 2^a, 3, 2^d, 1, 2^c$$

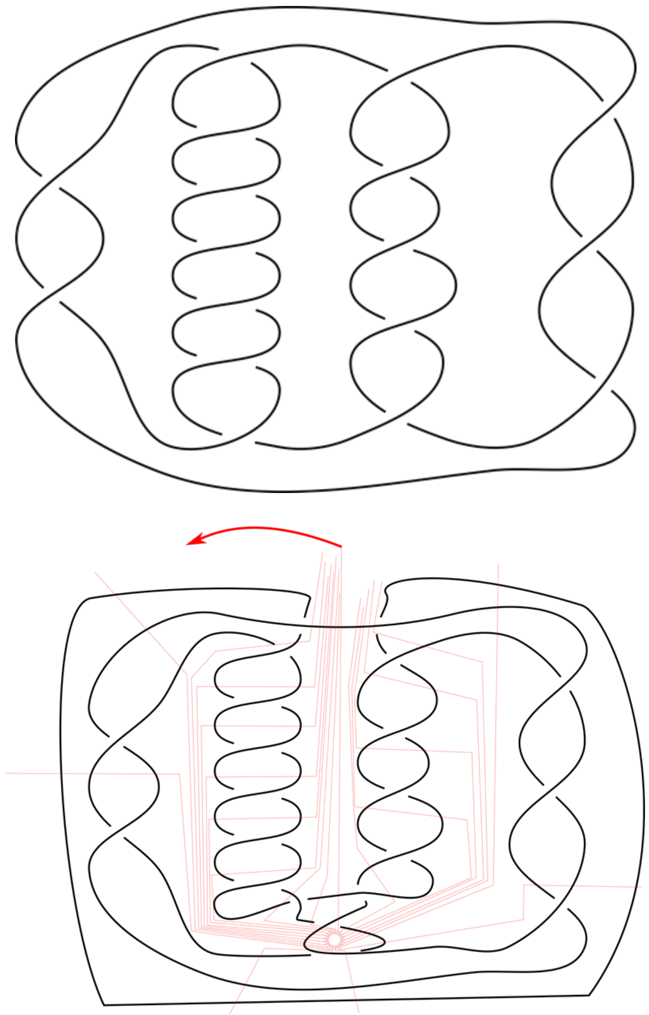


Ilustración 68. Nudo pretzel de 4 entradas y su trenza cerrada.

En este tipo de enlaces dejamos de tener el problema de que influya si las entradas son pares o no, y tenemos un algoritmo único para todos los casos. Esto se justificará y demostrará en la subsección correspondiente. El lector apreciará en dicha subsección que hay otro algoritmo posible, como se mostrará en el teorema 5.

Teorema 5

Sea P(a_1, a_2, a_3, …, a_N) un enlace pretzel con un número de entradas N par mayor que 2. Entonces este enlace es la clausura de la siguiente trenza:

$$\left\{-2,1,2;-4,3,4;-6,5,6;\ldots;\left(\frac{N-2}{2}\text{ veces}\right)\right\},$$
$$\left\{(N-1)^{a_1},(N-3)^{a_3},(N-5)^{a_5},\ldots,\left(\frac{N}{2}\text{ veces}\right)\right\},$$
$$\{-N+2,-N+3,N-2;\ldots;-6,-5,6;-4,-3,4;-2,-1,2\},$$
$$\{(N-1)^{a_2},(N-3)^{a_4},(N-5)^{a_6},\ldots,(\frac{N}{2}\text{ veces})\}$$

Como ejemplo aquí destacamos el enlace P(6, 4, 6, 8, 6, 4), clausura de la trenza

-2,1,2,-4,3,4,5,5,5,5,5,5,3,3,3,3,3,3,1,1,1,1,1,1,-4,-3,4,-2,-1,2,5,5,5,5,3,3,3,3,3,3,3,3,1,1,1,1

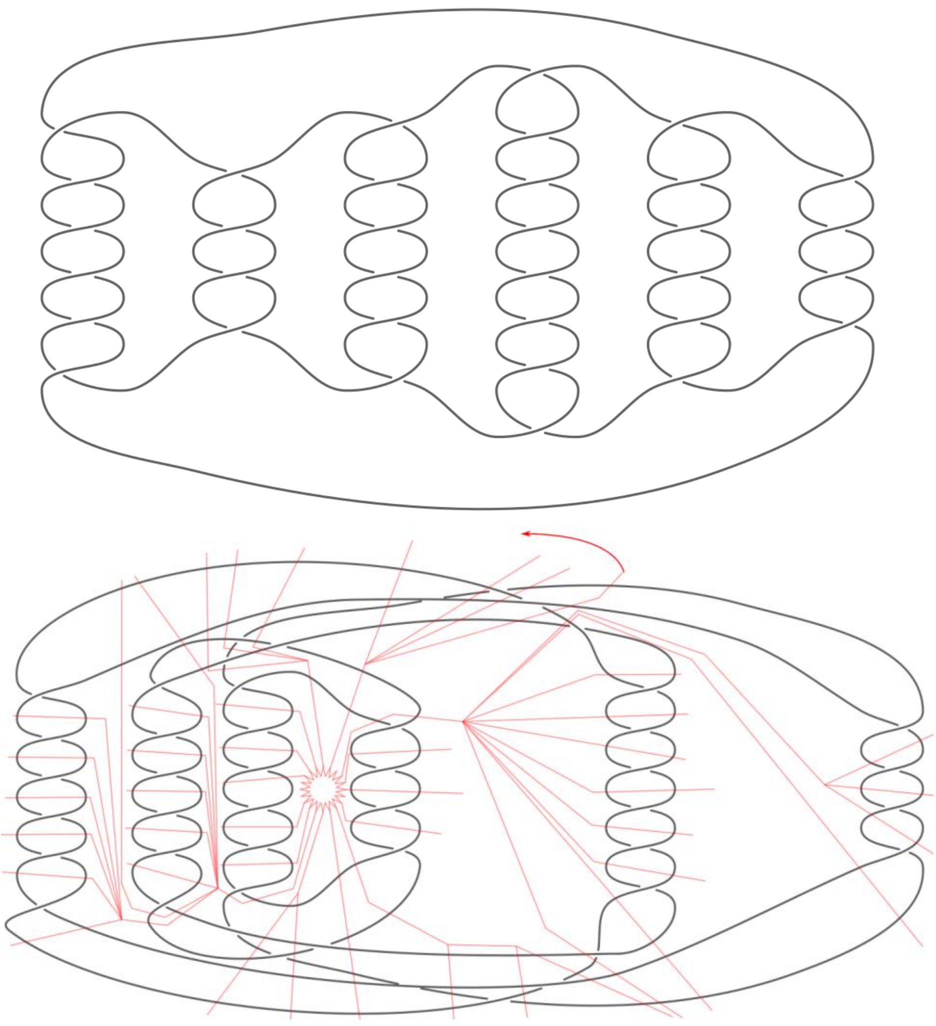


Ilustración 69. Pretzel con un número par de columnas y la trenza cerrada correspondiente.

Estrategia general para cualquier nudo pretzel

Con los teoremas anteriores hemos dado un algoritmo para cualquier enlace pretzel (no necesariamente nudo) de una, tres o cualquier número par de entradas.

Pero, más aún, he encontrado una estrategia general para abordar cualquier *nudo* pretzel con un número arbitrario de entradas. Para completar dicha estrategia, nos quedaría por considerar la situación en la que tenemos un número impar de entradas mayor que tres. Aunque aún no hayamos explicado los tipos de movimientos que hemos ideado para ello, podemos resumir la estrategia como sigue:

- Número de entradas impar y mayor que tres, *siendo todas impares*: en cada columna se hacen tantos movimientos básicos como sea necesario y finalmente se hacen reducciones múltiples.
- Número de entradas impar y mayor que tres, *siendo una de ellas par*: por la presentación circular podemos suponer que la entrada par se encuentra en el centro. Entonces se desplazan columnas de lado a lado de forma apropiada y manteniendo la columna par en el centro. Finalmente se harán movimientos básicos en la columna central.

Se pueden ver los detalles en la subsección Nudos pretzel con n entradas, n impar y mayor que tres, en la página 156, donde se procede separando los casos sin tener en cuenta la presentación circular.

5.2 Movimientos desarrollados

Para trabajar los enlaces pretzel en un principio contamos con movimientos de reducción y movimientos de polo sur (isotopías de diagrama)**.**

Estas herramientas por sí solas son insuficientes para nuestra labor, por lo que hemos desarrollado tres movimientos que resultarán esenciales: **movimiento básico, desplazamiento de columna y reducción múltiple.**

Movimiento de polo sur

Tenemos que imaginar que nuestro nudo está posado sobre la superficie de una esfera, pongamos de la Tierra. Podemos coger la cuerda exterior superior, pasarla por el polo norte, por el ecuador, luego por el polo sur, y acabará en el lado contrario al inicial. Como siempre se dice que una imagen vale más que mil palabras, pongamos este ejemplo:

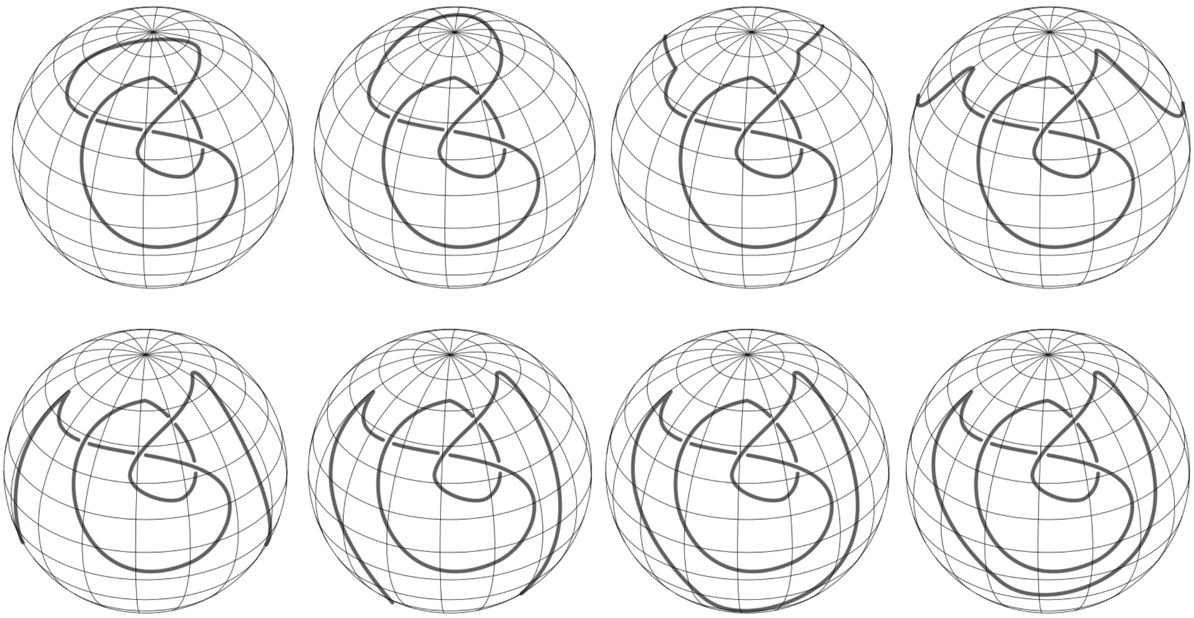


Ilustración 70. Movimiento de polo sur.

Es decir, hacemos esta transformación:

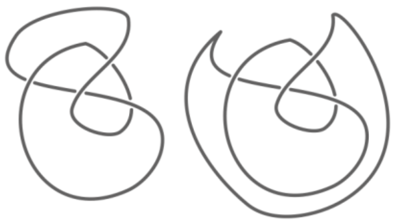


Ilustración 71. Movimiento de polo sur en el plano.

Claramente el nombre es una metáfora y se podrían llamar movimientos de polo norte, movimientos de meridiano, de ecuador, etc. La idea matemática es simple si pensamos que los diagramas de los nudos se hacen en la esfera y no en el plano.

Movimiento básico

Al tratar de hallar la trenza para el pretzel P(1, Par, 1) siguiendo la estrategia de reducir la complejidad con movimientos de reducción surgió un gran problema, ya que con tal cantidad de círculos de Seifert el resultado era casi incontrolable.

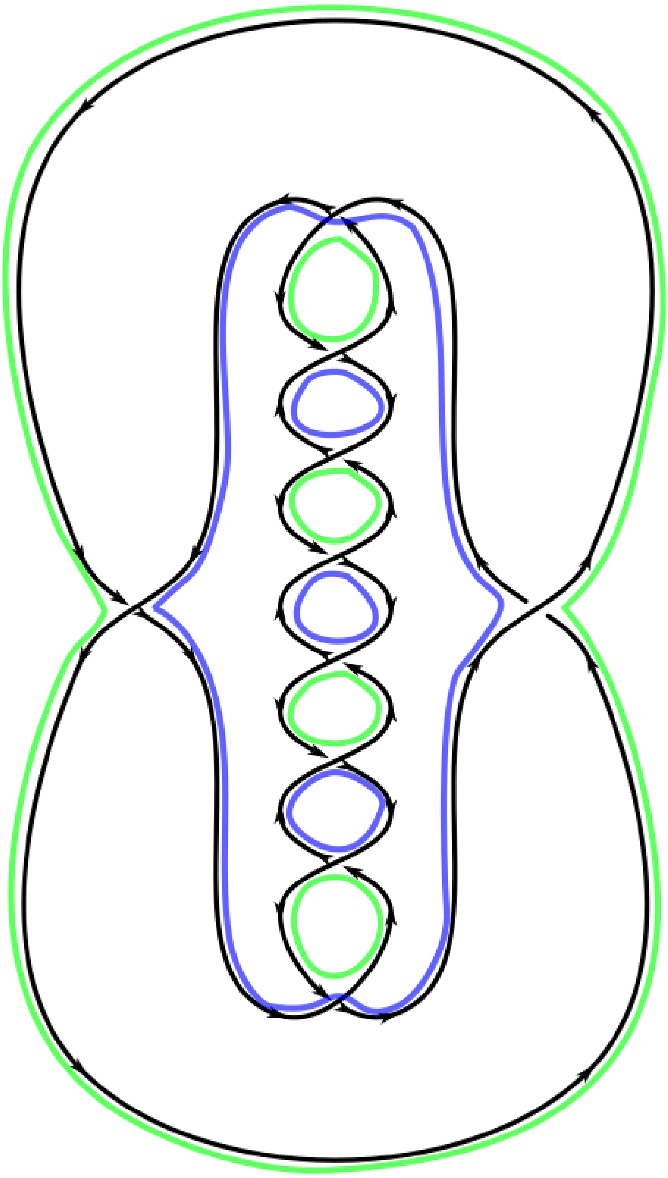


Ilustración 72. P(1, 8, 1).

En concreto, este nudo tiene complejidad 15, y por experiencia puedo decir que no es sencillo manejar 15 movimientos de reducción.

En una sesión online con mi tutor estábamos buscando otra forma de atacar este problema. Trabajando conjuntamente se nos ocurrió la idea de trasponer una de las cuerdas superiores sobre todas las demás en el caso de haber una columna de círculos de Seifert, de la manera que vamos a ver a continuación. Como el P(1, 8, 1) es muy basto, vamos a verlo primero con P(1, 4, 1), que es un ejemplo perfecto.

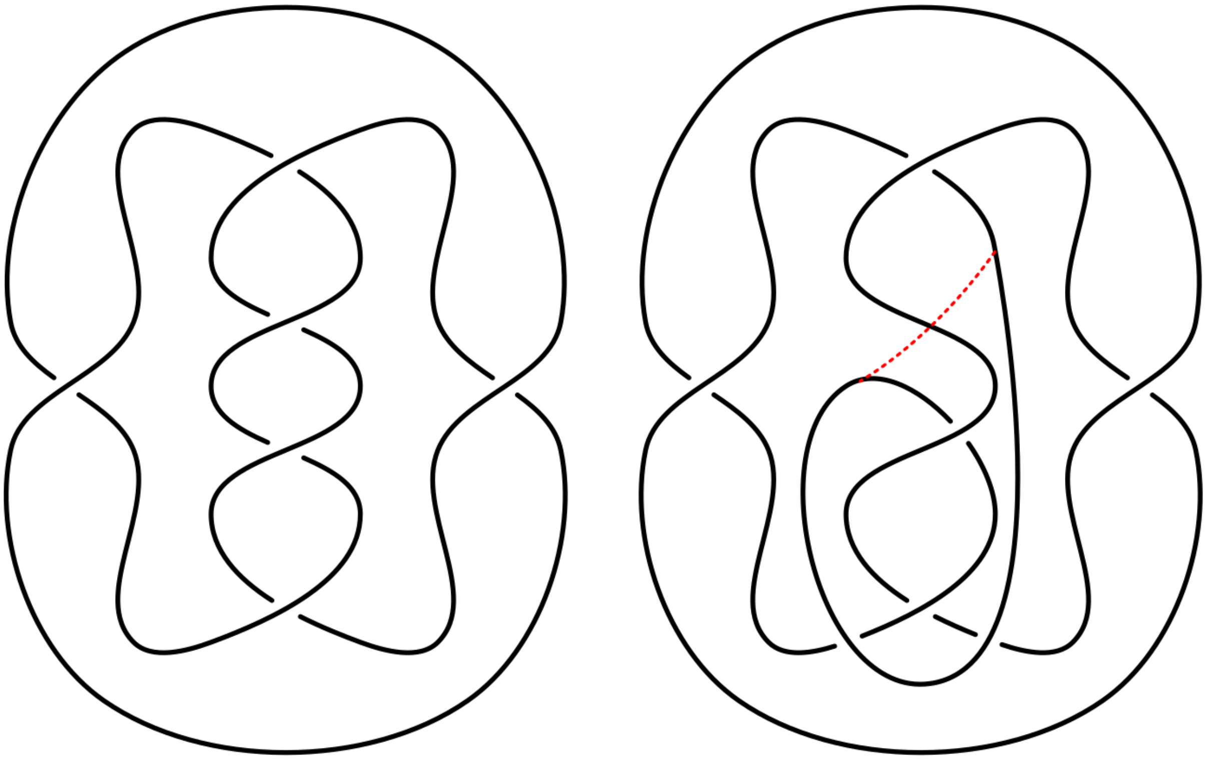


Ilustración 73. Un movimiento básico en la columna central.

Para saber si hemos ganado o perdido algo, tenemos que estudiar los círculos de Seifert antes y después de dicho movimiento.

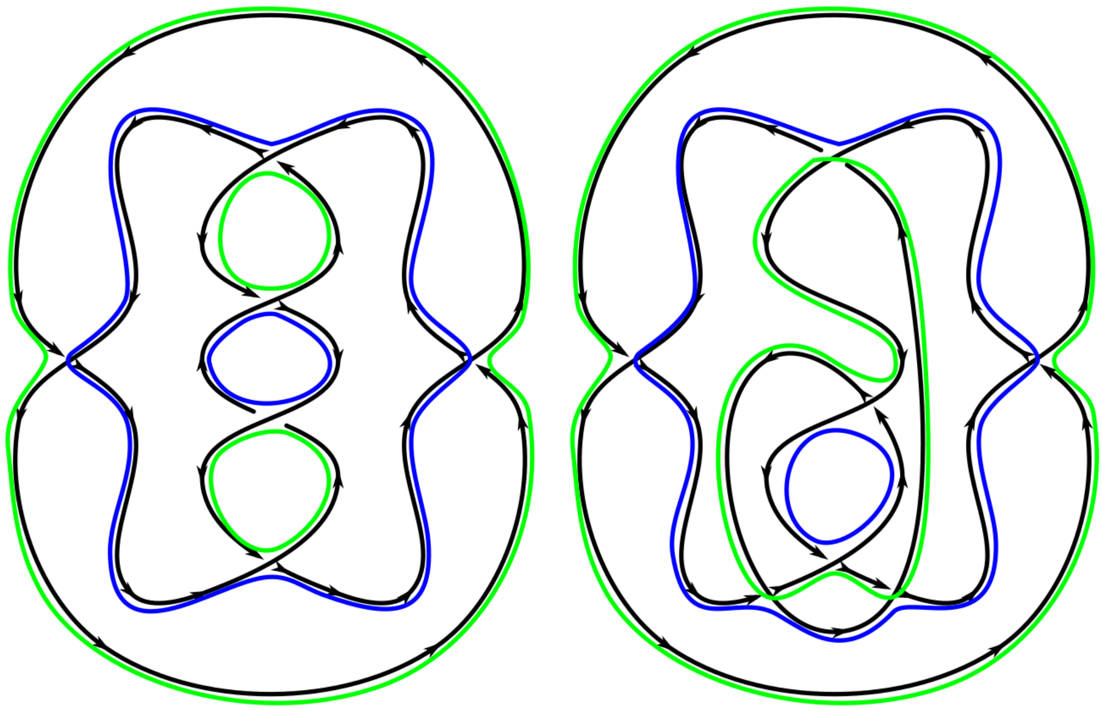


Ilustración 74. Círculos de Seifert antes y después de un movimiento básico.

Podemos observar que ahora tenemos un círculo de Seifert menos, y que todos los que han quedado son compatibles. Por inesperado que parezca, este movimiento no sólo nos soluciona el problema de los círculos incompatibles en una columna, además reduce el número total de círculos de Seifert en una unidad.

Para columnas con mayor cantidad de círculos sólo es necesario repetir este movimiento. En P(1, 8, 1), por ejemplo, son necesarios tres movimientos básicos.

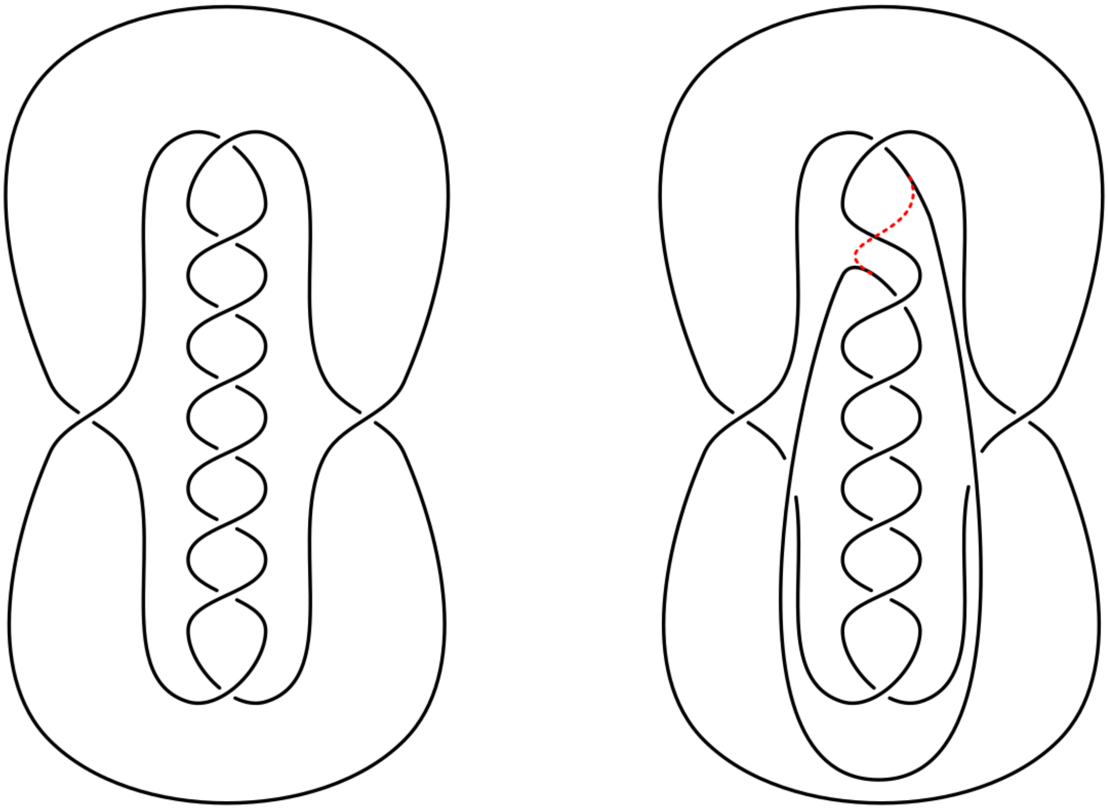


Ilustración 75. Primer movimiento básico.

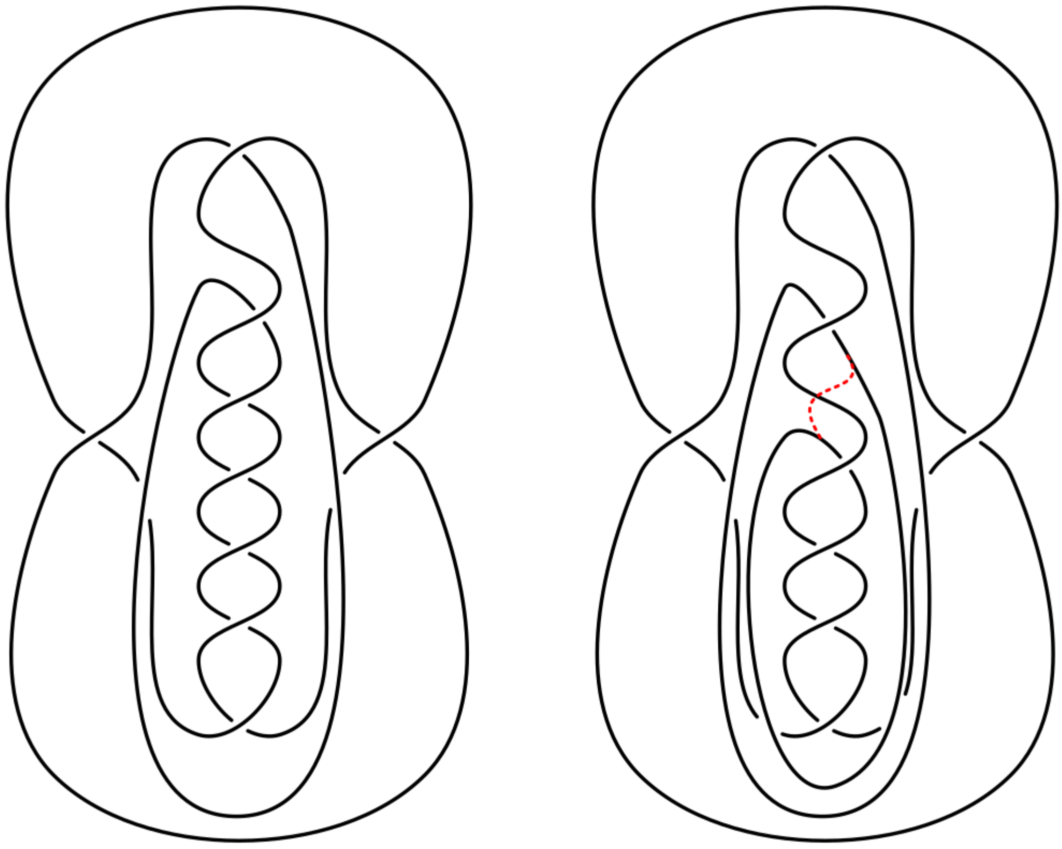


Ilustración 76. Segundo movimiento básico.

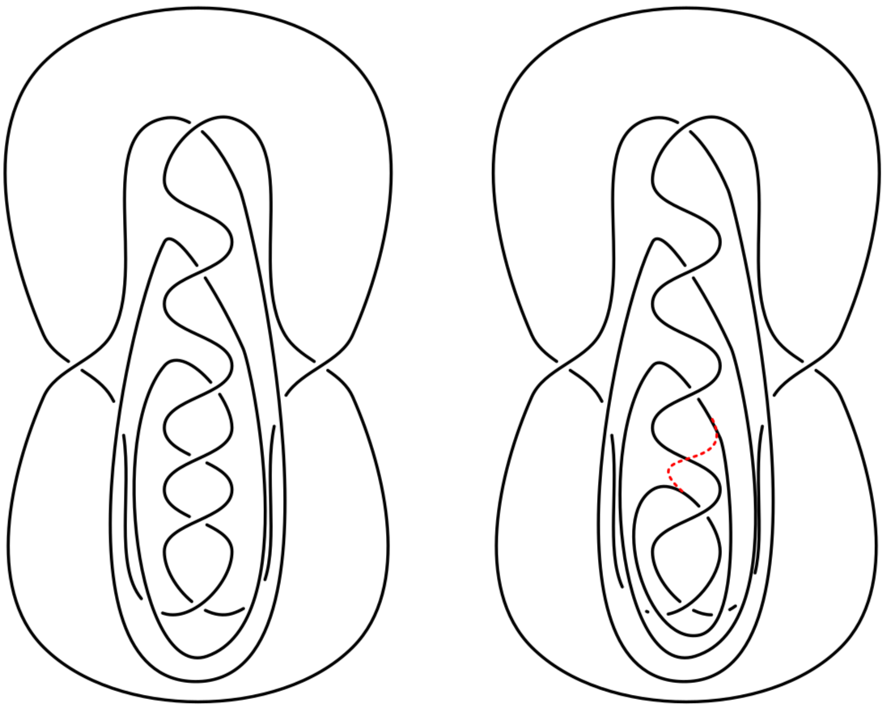


Ilustración 77. Tercer y último movimiento básico.

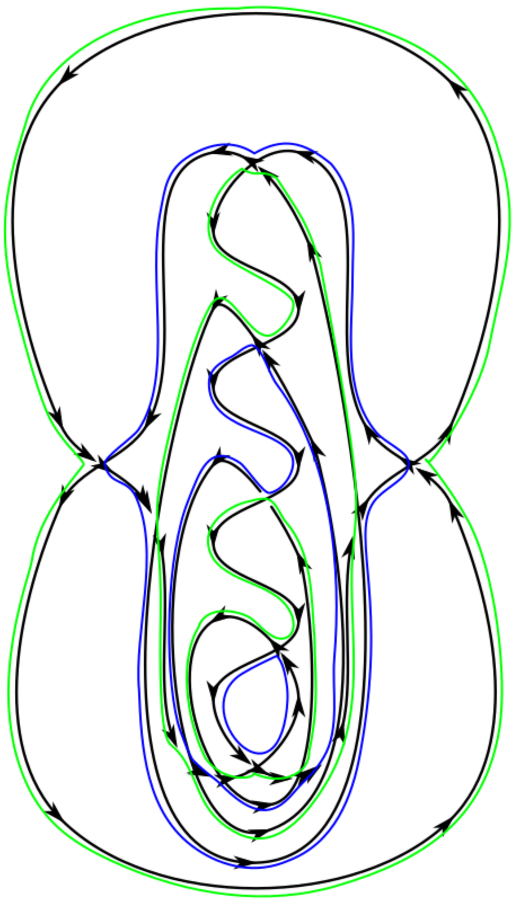


Ilustración 78. Círculos de Seifert tras los movimientos básicos.

Como el lector puede imaginar, este método es bastante más atractivo que realizar las 15 reducciones que serían necesarias por el método usual. Y es que hemos llegado a complejidad cero, pues todos los círculos de Seifert son compatibles (y concéntricos).

Movimiento de reducción múltiple

Cuando tratemos enlaces pretzel con un número impar de entradas y todas las entradas sean impares, nos encontraremos con grupos de círculos de cuerdas concéntricas. Como coincidirán con círculos de Seifert, tendremos que reducirlos juntos, de manera que se va a formar una alta cantidad de cruces en un espacio muy pequeño.

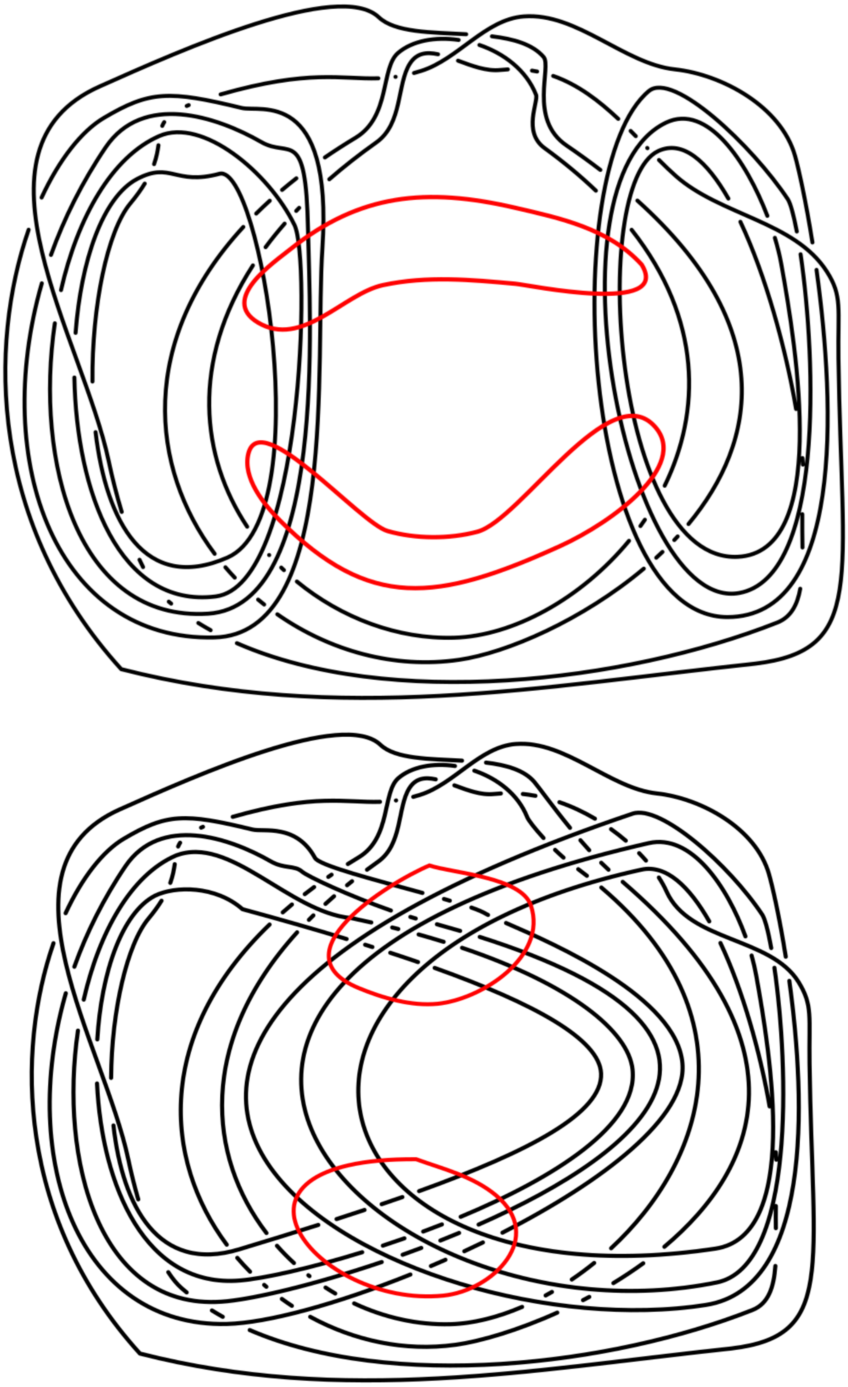


Ilustración 79. Movimiento de reducción múltiple.

Como el lector apreciará, obtendremos tantos cruces como el producto del número de cuerdas de un grupo por el del otro. Todo esto está explicado muy a fondo en la página 98, ya que un análisis adecuado de un ejemplo de este movimiento requiere una cantidad de páginas bastante excesiva para esta introducción de los movimientos.

Desplazamiento de columna

Nuestro último as en la manga no lo podremos usar hasta dentro de 60 páginas, pero cuando llegue la hora de llamar a filas a esta herramienta el lector se maravillará de su gran utilidad. Mientras que los pasados movimientos básicos nos resolvían el caso de tener círculos de Seifert dentro de las columnas (entre los cruces), este desplazamiento nos ayuda si nos encontramos en el aprieto de tener los círculos de Seifert en los huecos entre columnas.

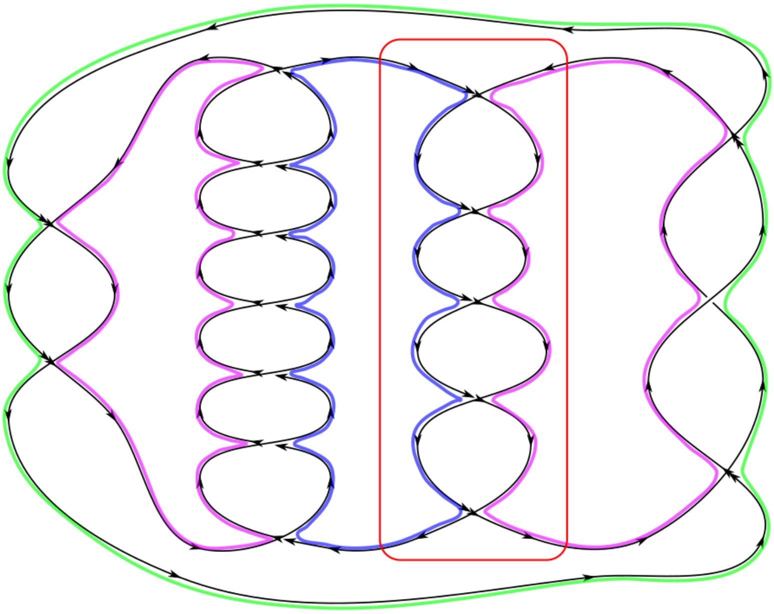


Ilustración 80. Columna que será desplazada a la izquierda.

¿Cómo lo usamos? Tal y como indica su nombre, tendremos que coger una columna de cruces (interior, no valen las extremas) y desplazarla fuera del nudo. En concreto, ahora vamos a desplazar la tercera columna a la izquierda.

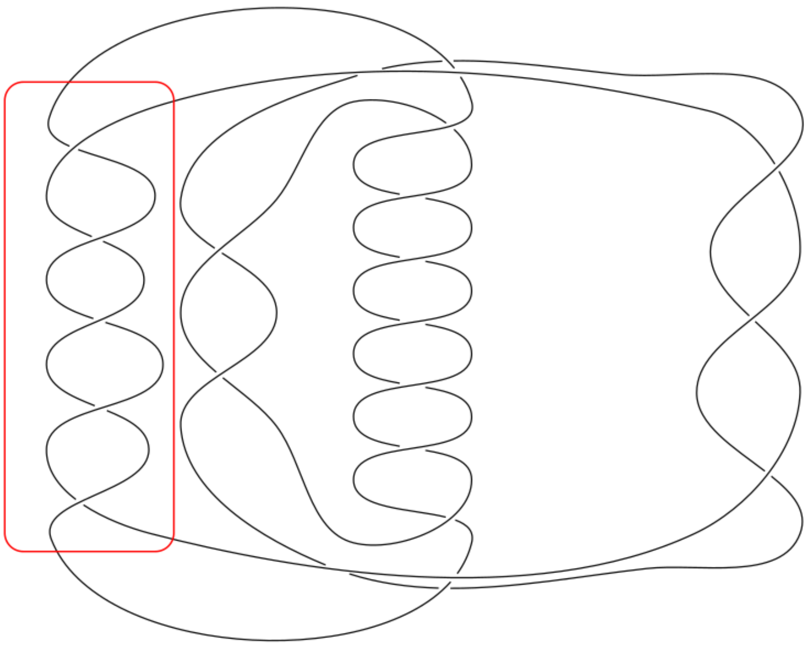


Ilustración 81. Columna desplazada a la izquierda.

Esto lo hemos hecho cogiendo la columna, elevándola, y llevándola a la izquierda para volver a posarla. De ahí que surjan 6 nuevos cruces, 3 arriba y 3 abajo. Observando la ilustración, de los 3 cruces, dos (marcados) los crean las cuerdas que conectan la columna con el resto del diagrama al pasar por encima de lo que antes era la cuerda exterior del susodicho. El tercero se crea ya que la cuerda del lado contrario hacia el que lo hemos desplazado tiene que pasar encima de la otra. En nuestro caso, la derecha tiene que pasar por encima de la izquierda. Esto requiere visión espacial, y se recomienda emular el movimiento en la realidad para poder observarlo.

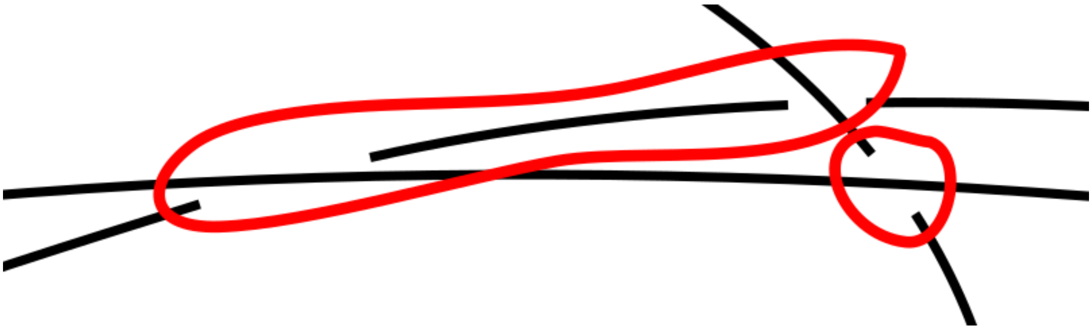


Ilustración 82. Detalle de los 3 cruces generados.

Como podemos observar, ahora todos los círculos son compatibles y concéntricos, situándose el centro entre las que antiguamente eran la primera y segunda columnas.

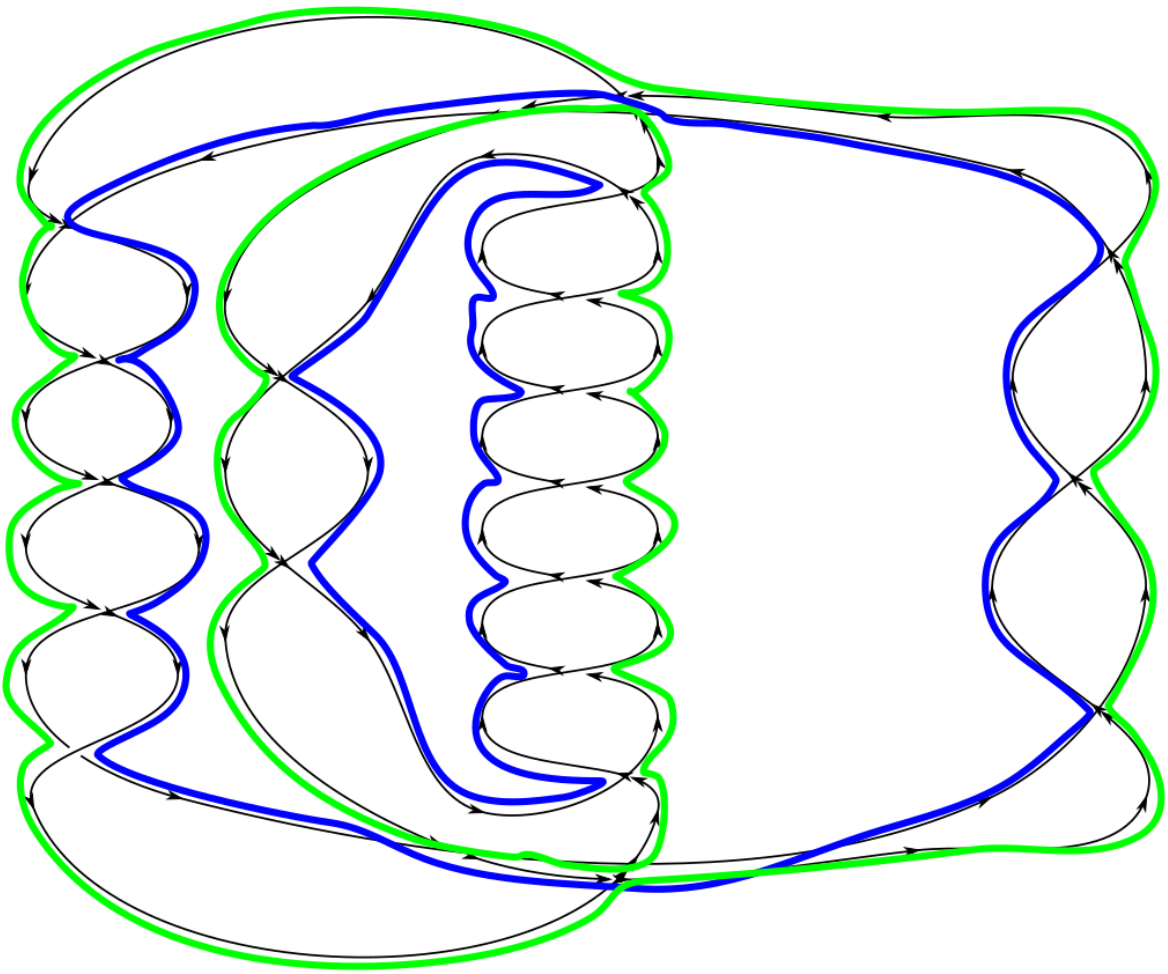


Ilustración 83. Círculos de Seifert tras un movimiento de columna.

5.3 Demostración para un ejemplo seleccionado

Antes de pasar a los detalles de todos los casos, que son largos y densos, vamos a hacer la demostración en un ejemplo concreto. Esto debe servir de guía para los casos futuros, y nos permitirá visualizar las ideas claves sin perdernos.

Trataremos un nudo pretzel en el que tenemos que usar tres de nuestras herramientas: el movimiento básico, el movimiento de polo sur y el desplazamiento de columna. En concreto es el nudo P(3, 3, 3, 3, 3, 3, 4).

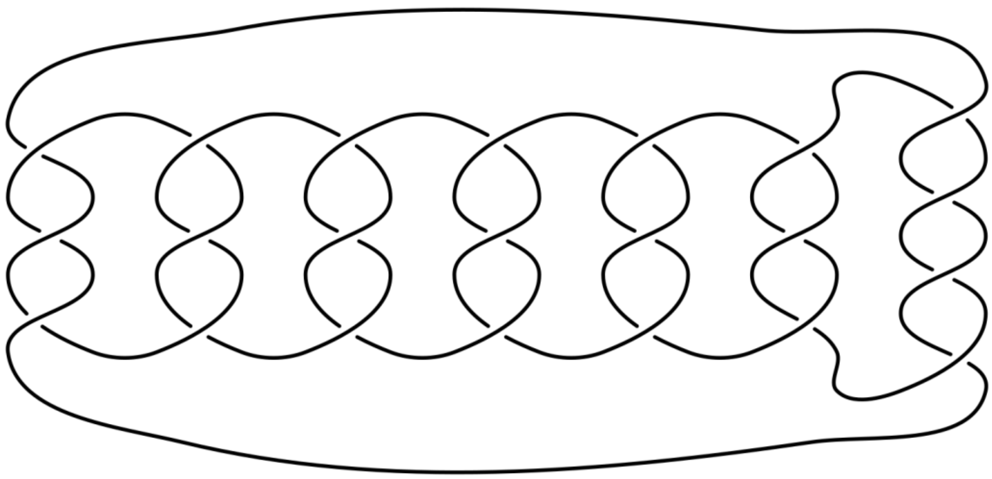


Ilustración 84. Nudo P(3, 3, 3, 3, 3, 3, 4).

En su correspondiente subsección veremos detalladamente la estrategia a seguir, pero la voy a ir adelantando. Este ejemplo nos permitirá destacar las ideas principales. Siempre estudiaremos los círculos de Seifert, ya que dependiendo de su situación tendremos que tomar un camino u otro.

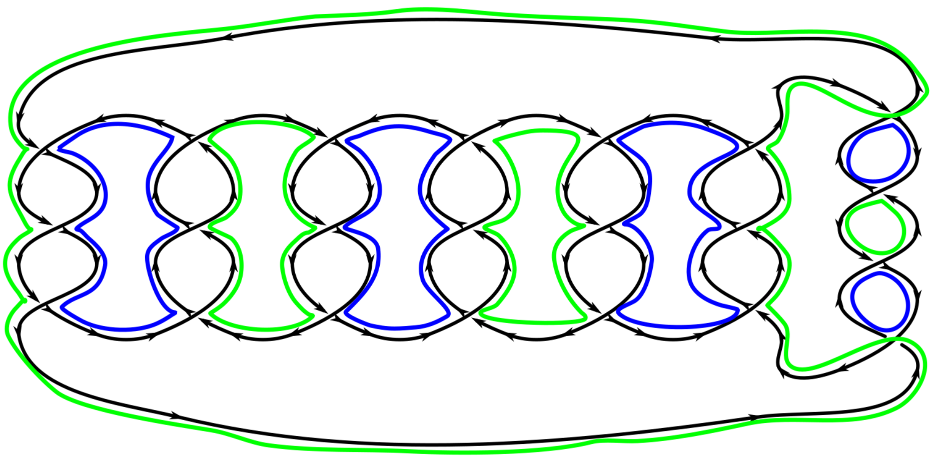


Ilustración 85. Nudo P(3, 3, 3, 3, 3, 3, 4) orientado y sus círculos de Seifert.

Por una parte, nos encontramos con círculos de Seifert en los huecos entre columnas. Para tratarlos haremos 2 movimientos de torres. Las moveremos a la izquierda, ya que si las moviésemos a la derecha la columna que tiene círculos de Seifert dentro dejaría de estar en un extremo, y necesitamos que este tipo de columnas esté siempre o en el medio o en un extremo.

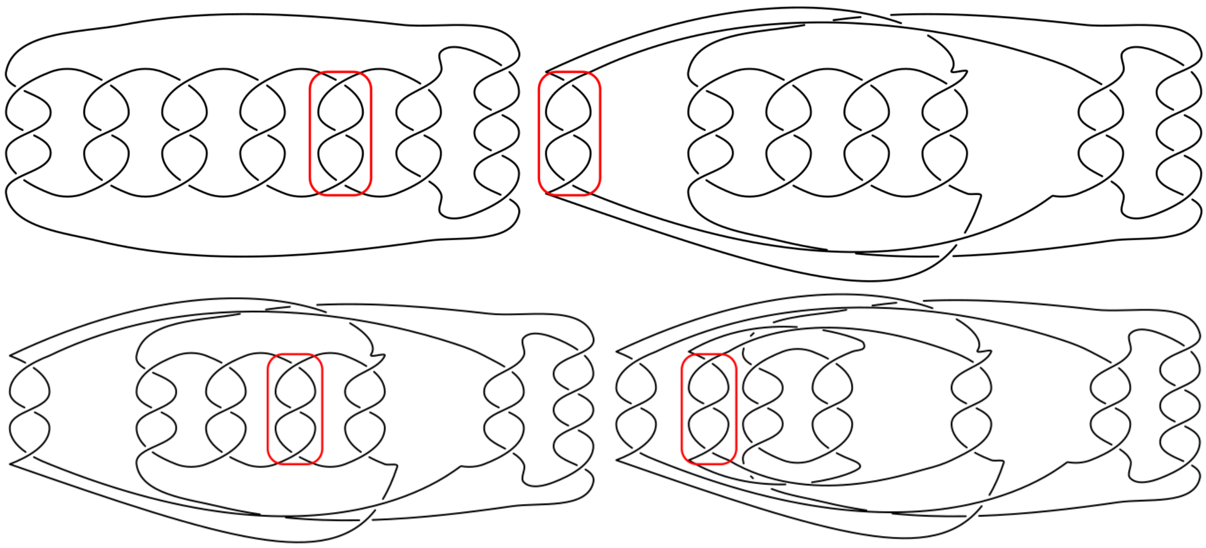


Ilustración 86. Dos desplazamientos de columnas.

Ahora nos queda solucionar la columna derecha, que está llena de círculos de Seifert. Esto lo hacemos con un movimiento básico para seguidamente hacer dos movimientos de polo sur.

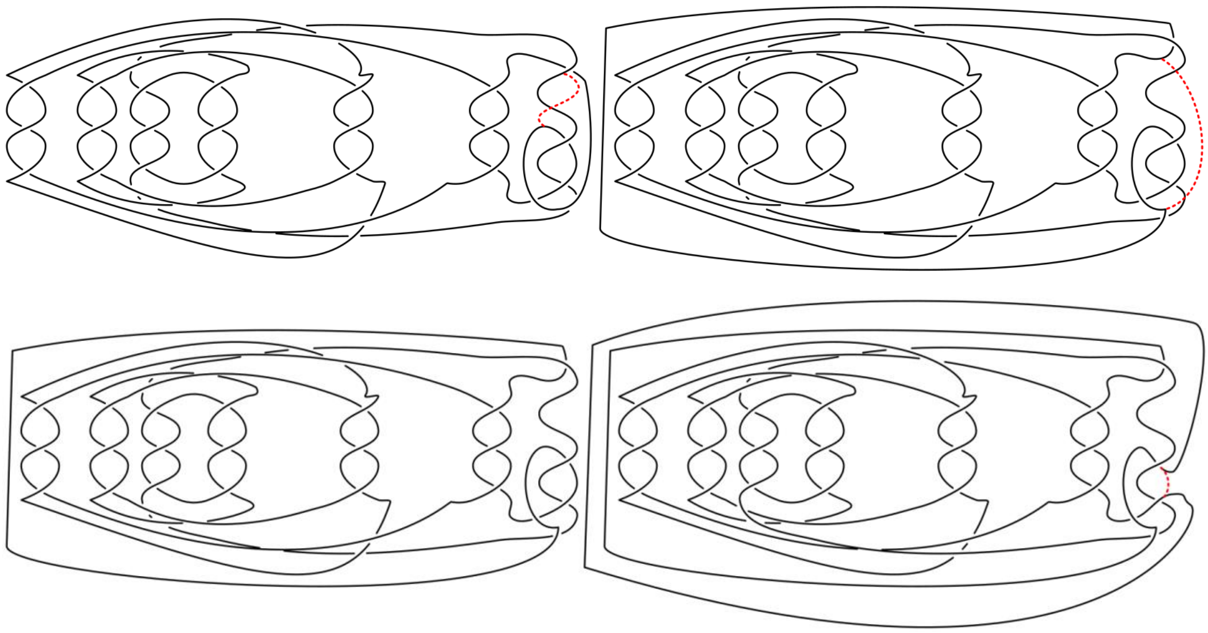


Ilustración 87. Movimiento básico en la columna derecha y dos movimientos de polo sur.

Si orientamos ahora el nudo observamos que es una trenza con sentido anti horario.

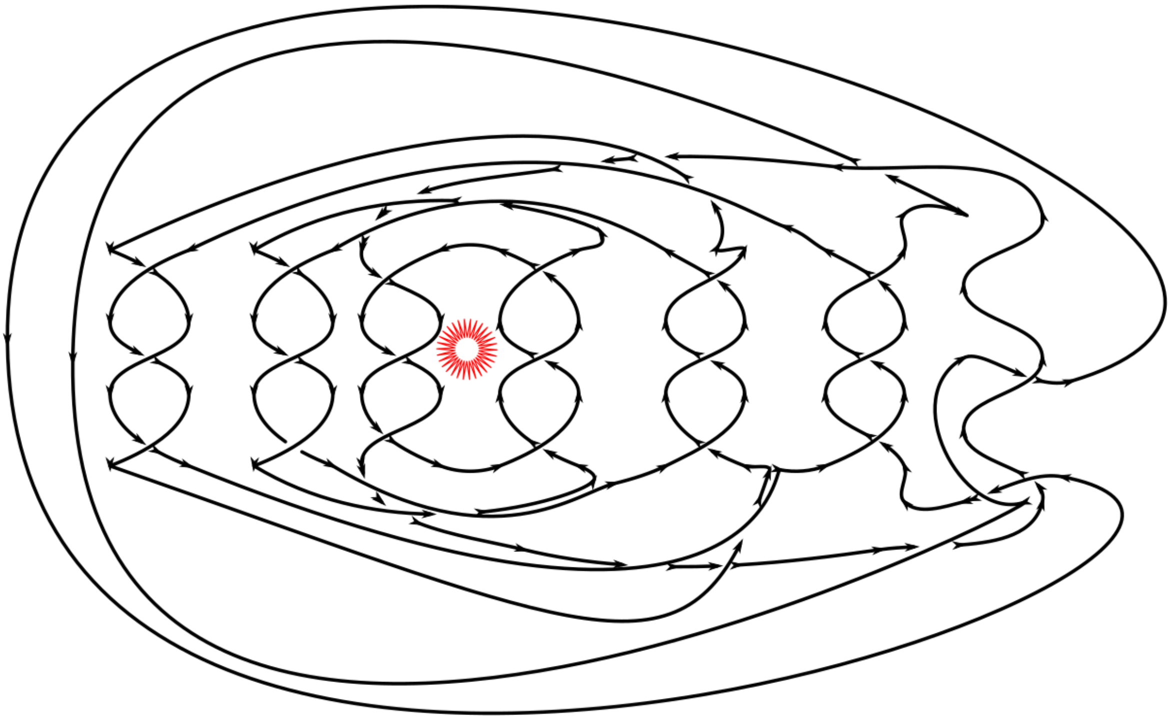


Ilustración 88. El nudo pretzel P(3, 3, 3, 3, 3, 3, 4) como clausura de una trenza.

5.4 Demostraciones

Pasamos ya a la demostración cuidadosa de los teoremas, empezando por los nudos pretzel de 3 entradas. Aquí empieza la demostración del Teorema 3.

Nudos pretzel con tres entradas

Como ya se ha mencionado, dependiendo si las entradas son pares o no, se resuelve el enlace de una forma u otra, así que se dividirá esta subsección. He empezado con los nudos pretzel, es decir o una entrada es par o todas son impares.

El orden ha sido: P(impar, par, impar), P(par, impar, impar), P(impar, impar, par) y por último P(impar, impar, impar)

Nudos P(impar, par, impar)

Para ir poco a poco, sobre todo al principio, se han ido haciendo casos particulares para luego deducir el método y pasar al caso general. Por ello empiezo usando sólo la columna central variable, y dejando 1 en la columna izquierda y derecha. Luego hago variable la columna derecha y finalmente la izquierda.

Nudos P(1, par, 1)

Los nudos pretzel con una columna par en el centro tienen una propiedad muy curiosa, y es que empiezan teniendo dos círculos de Seifert compatibles externos y en la columna par tantos círculos como sea el número par menos uno. Esto es extremadamente útil, ya que sólo nos faltan los círculos concéntricos interiores.

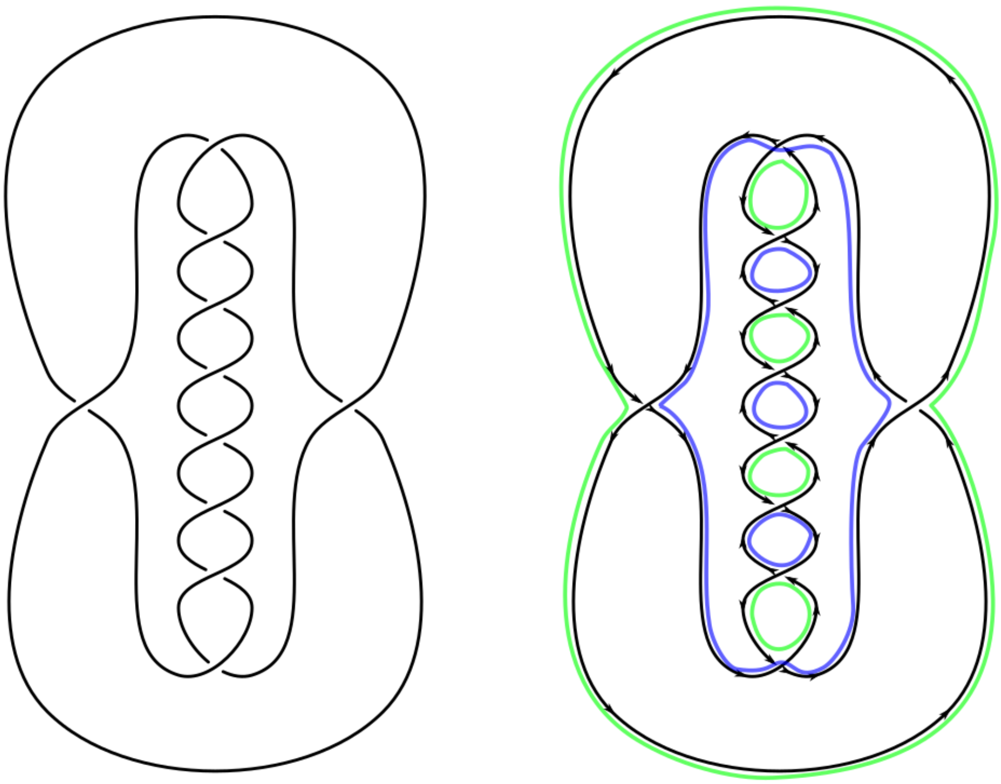


Ilustración 89. P(1, 8, 1) orientado y sus círculos de Seifert.

Como ya hemos visto con los movimientos básicos, reducir este nudo nos resultará muy laborioso. Una de las razones es que tiene complejidad 15.

Por ello, vamos a probar con movimientos básicos a ver si se nos simplifica. Empezamos cogiendo el segundo cruce de la columna central y lo trasladamos abajo. Luego lo mismo con el cuarto, y luego con el sexto.

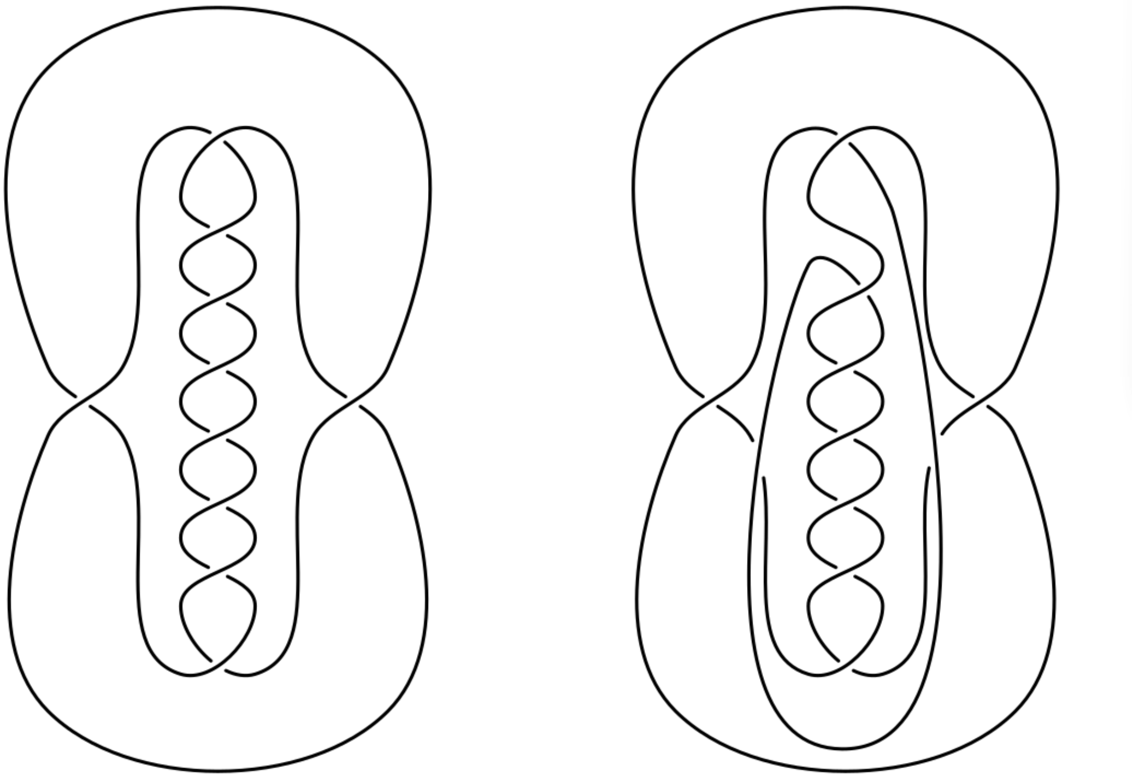


Ilustración 90. Primer movimiento básico.

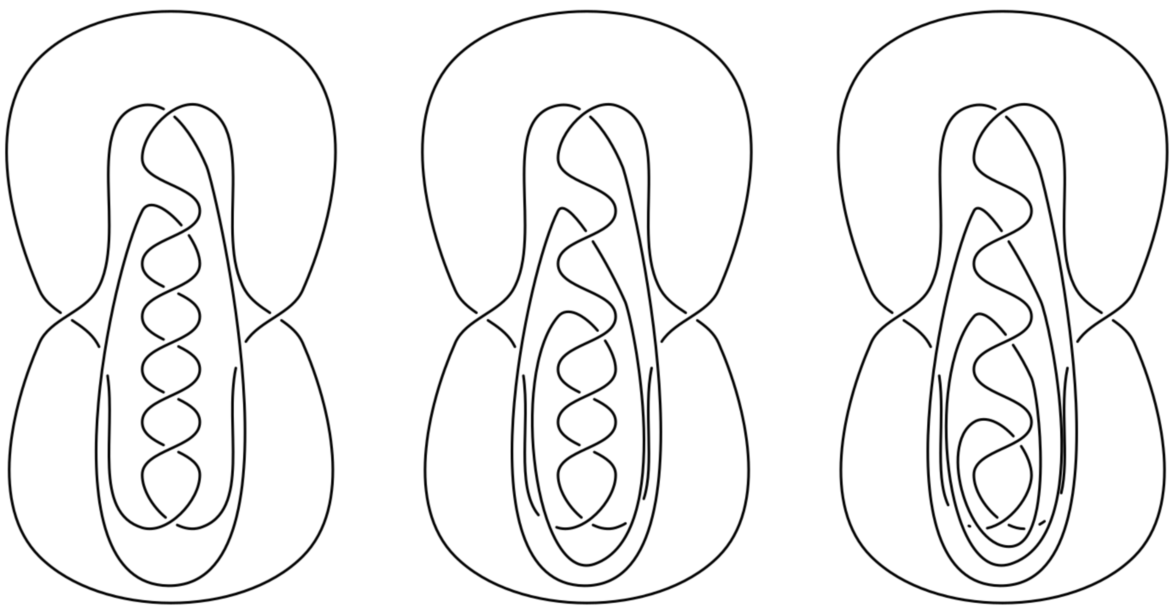


Ilustración 91. Segundo y tercer movimiento básico.

Si analizamos ahora los círculos de Seifert vemos algo maravilloso, y es que no sólo hemos reducido el número de éstos en 3 (ya que hemos hecho 3 movimientos básicos), si no que ahora todas las parejas de círculos son compatibles. Así que ya tenemos la trenza buscada.

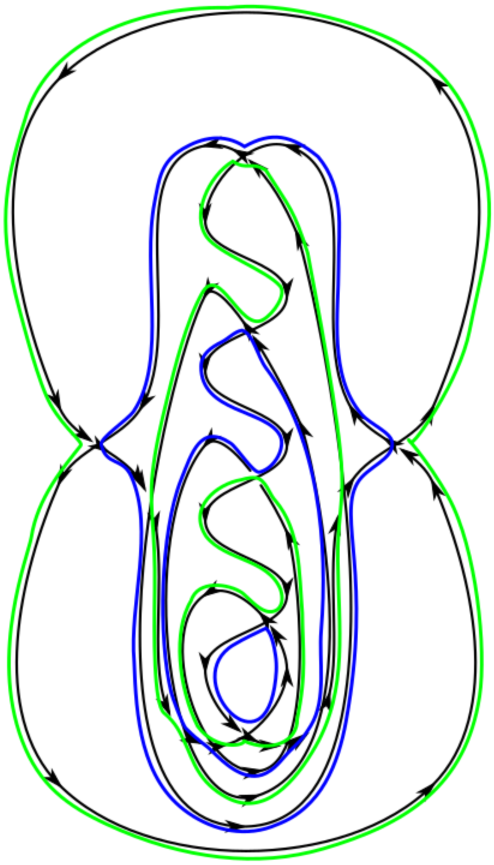


Ilustración 92. P(1, 8, 1) como trenza clausurada.

Puede que no se vea muy bien la trenza, pero si movemos un poco nuestras cuerdas con cuidado de no crear ni deshacer ningún cruce tenemos:

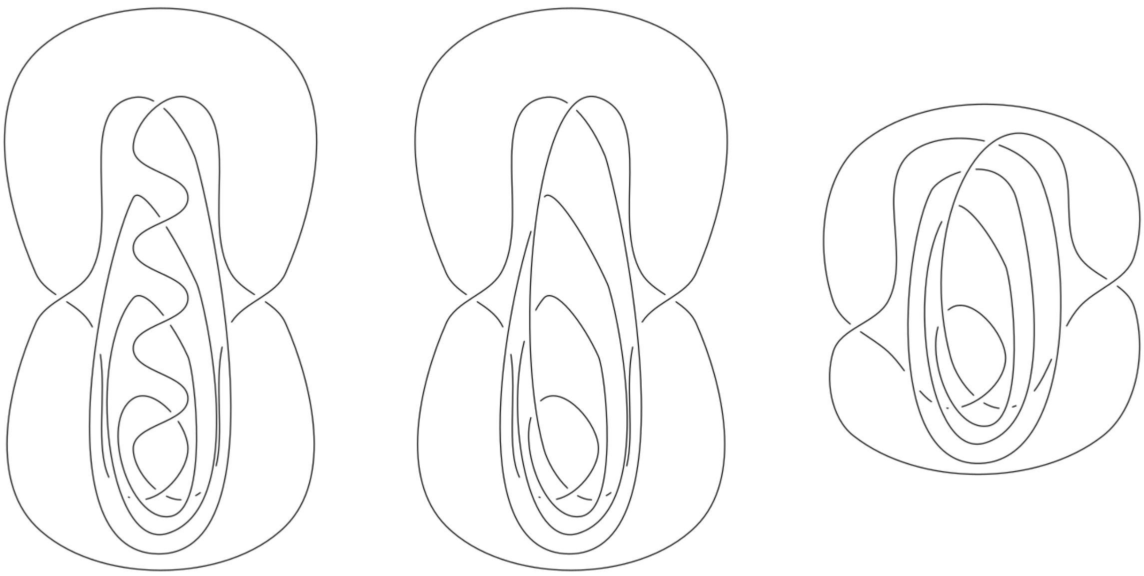


Ilustración 93. Isotopía para simplificar P(1, 8, 1).

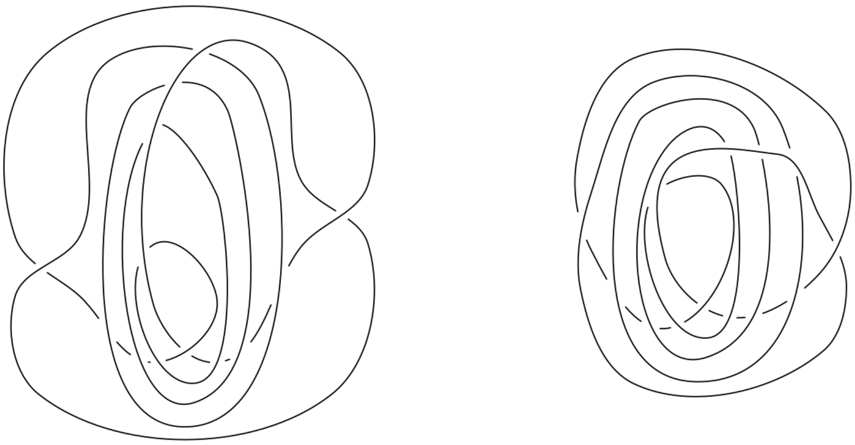


Ilustración 94. Isotopía para simplificar P(1, 8, 1), continuación.

Estos movimientos tienen que hacerse lentamente y comprobando siempre que se hayan hecho bien, ya que como ya se ha dicho en los errores de Inkscape, un pequeño despiste puede hacer que acabemos con una trenza diferente a la que buscamos.

Si contamos tenemos 9 cruces, así que situaremos una estrella de 9 puntas en el centro de la trenza y trazaremos rayos que separen los cruces. Una vez hecho tomaremos un cruce como nuestro primer cruce y empezaremos a escribirlos en sentido anti horario.

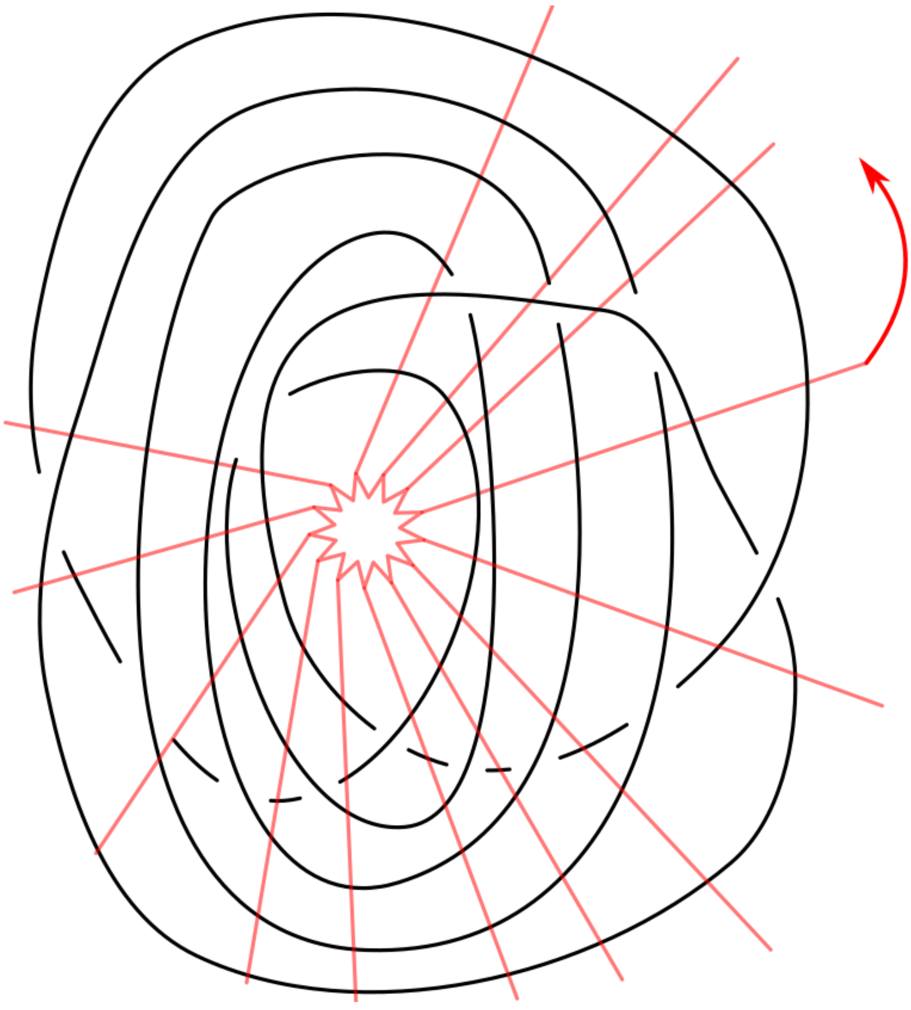


Ilustración 95. P(1, 8, 1) como trenza clausurada.

Si lo leemos con cuidado, la palabra de la trenza es

-2, -3, -4, -5, 1, 2, 3, 4, -5, -4, -3, -2, 1

He tratado otros como el P(1, 1, 1), el P(1, 2, 1), el P(1, 4, 1) y el P(1, 6, 1), pero éste es el más representativo del algoritmo. Es importante usar números grandes para que no desaparezcan componentes.

Si prestamos atención, independientemente del valor de la columna central la primera agrupación es (-2, -3, -4,-5); en general siempre empieza en -2, y llegará hasta –(1+b/2), siendo b el valor de la entrada central. Luego irá desde 1 hasta b/2 y luego hará la primera serie, pero a la inversa. Para terminar, acabará en 1.

$$\boxed{-2 \downarrow -\frac{|b|+2}{2}, 1 \uparrow \frac{|b|}{2}, -\frac{|b|+2}{2} \uparrow -2,1}$$

Ecuación 1. Algoritmo para P(1, Par, 1).

Se podría profundizar más pero vamos a entender bien todas las agrupaciones y la influencia de columnas con cruces negativos cuando estudiemos el algoritmo general en la subsección Nudos P(impar, par, impar).

Nudos P(1, par, impar)

Una vez que tenemos el algoritmo para Nudos P(1, par, 1), llega el momento de añadir otra variable y pasar a P(1, par, impar). Para ello se ha estudiado el P(1, 4, 3), el P(1, 4, 5) y el P(1, 4, 7). Todos son igual de representativos por lo que vamos a ver a continuación.

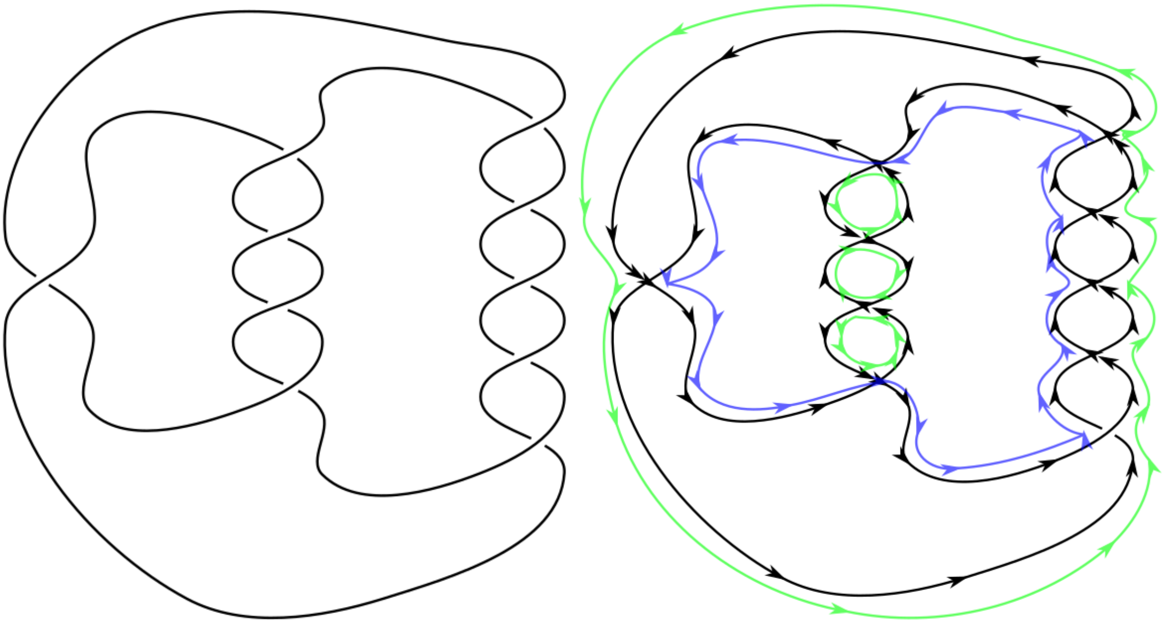


Ilustración 96. P(1, 4, 5).

Se da el mismo caso que en el problema anterior, 2 círculos exteriores y una columna central con círculos, así que realizaremos un movimiento básico en la columna central.

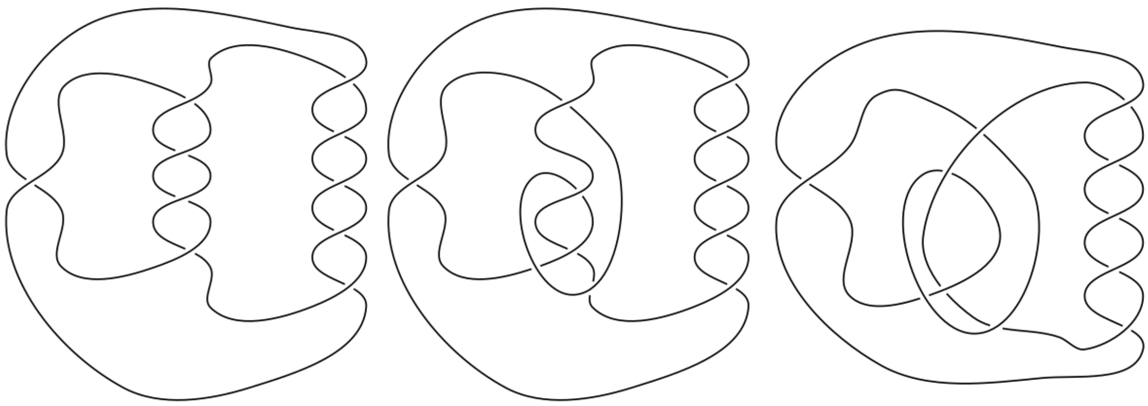


Ilustración 97. P(1, 4, 5). Movimiento básico y simplificación.

Tras orientarlo y hacer los círculos de Seifert tenemos exactamente la misma situación que con Nudos P(1, par, 1), solo que el 1 del final va a aparecer tantas veces como el número de la tercera columna. Vamos a verlo.

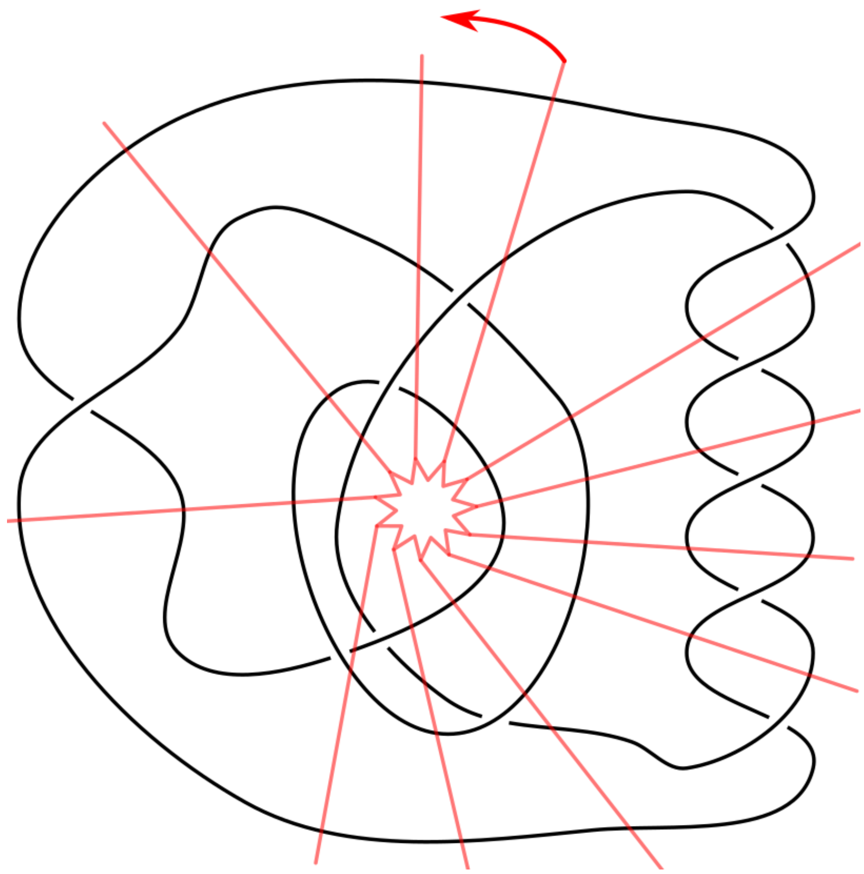


Ilustración 98. Pretzel P(1, 4, 5) como trenza cerrada.

Vamos a estudiar esta trenza ayudándonos del algoritmo que ya tenemos para Nudos P(1, par, 1). Recordemos la idea de ese caso:

"La primera agrupación (-2, -3, -4,-5) siempre empieza en -2, y llegará hasta –(1+b/2), siendo b el valor de la columna central. Luego irá desde 1 hasta b/2 y luego hará la primera serie, pero a la inversa. Para terminar, acabará en 1."

Empezamos en -2 y llegamos hasta -(1+4/2) = -3 → (-2,-3).

Luego empezamos en 1 hasta 4/2 =2 → (-2,-3,1,2).

A continuación, la primera serie a la inversa → (-2,-3,1,2,-3,-2).

Para finalizar tenemos la diferencia, en vez de un 1 tenemos tantos 1 como sea el valor de la última columna, en este caso cinco, → (-2,-3,1,2,-3,-2,1,1,1,1,1).

Es decir, el nudo pretzel (1, 4, 5) se corresponde con la trenza -2,-3,1,2,-3,-2,1,1,1,1,1.

$$\boxed{-2 \downarrow -\frac{|b|+2}{2}, 1 \uparrow \frac{|b|}{2}, -\frac{|b|+2}{2} \uparrow -2, 1^c}$$

Ecuación 2. Algoritmo para P(1, Par, Impar).

Vamos a estudiar la influencia de los signos en el caso general, ya que engloba a éste.

Nudos P(impar, par, impar)

Sabiendo que haciendo variable la columna de la derecha sólo hemos tenido que hacer que el 1 final se convierta en una agrupación que se repita tantas veces como la entrada de dicha columna, podemos prever que pasará algo parecido con la columna izquierda.

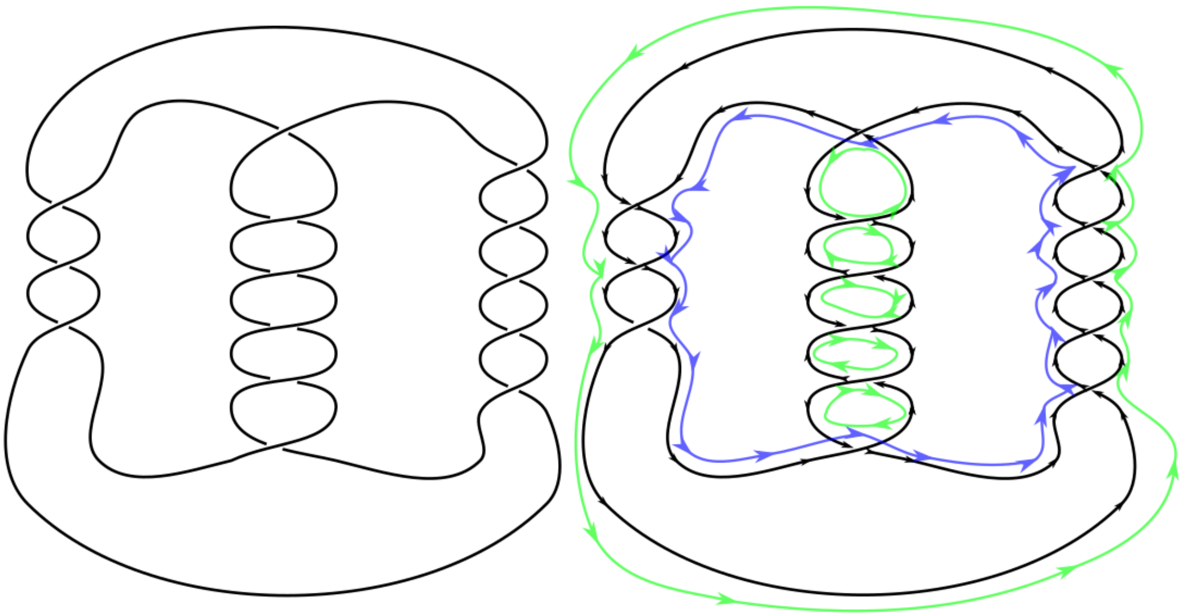


Ilustración 99. P(3, 6, 5).

Antes de hacer los movimientos básicos, paremos un momento para apreciar los movimientos de reducción que serían necesarios por el método que aparece desarrollado en el libro de Peter Cromwell [14] y la alta probabilidad de equivocarse. Mientras tanto, con 2 movimientos básicos hemos resuelto el problema como se verá en la siguiente ilustración:

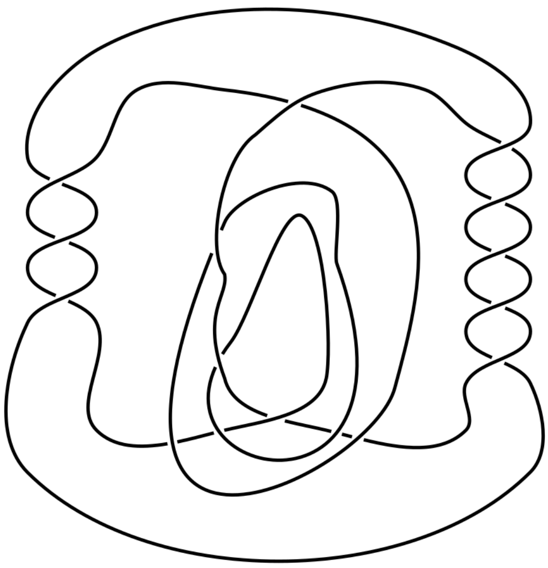


Ilustración 100. Dos movimientos básicos en la columna central de P(3, 6, 5).

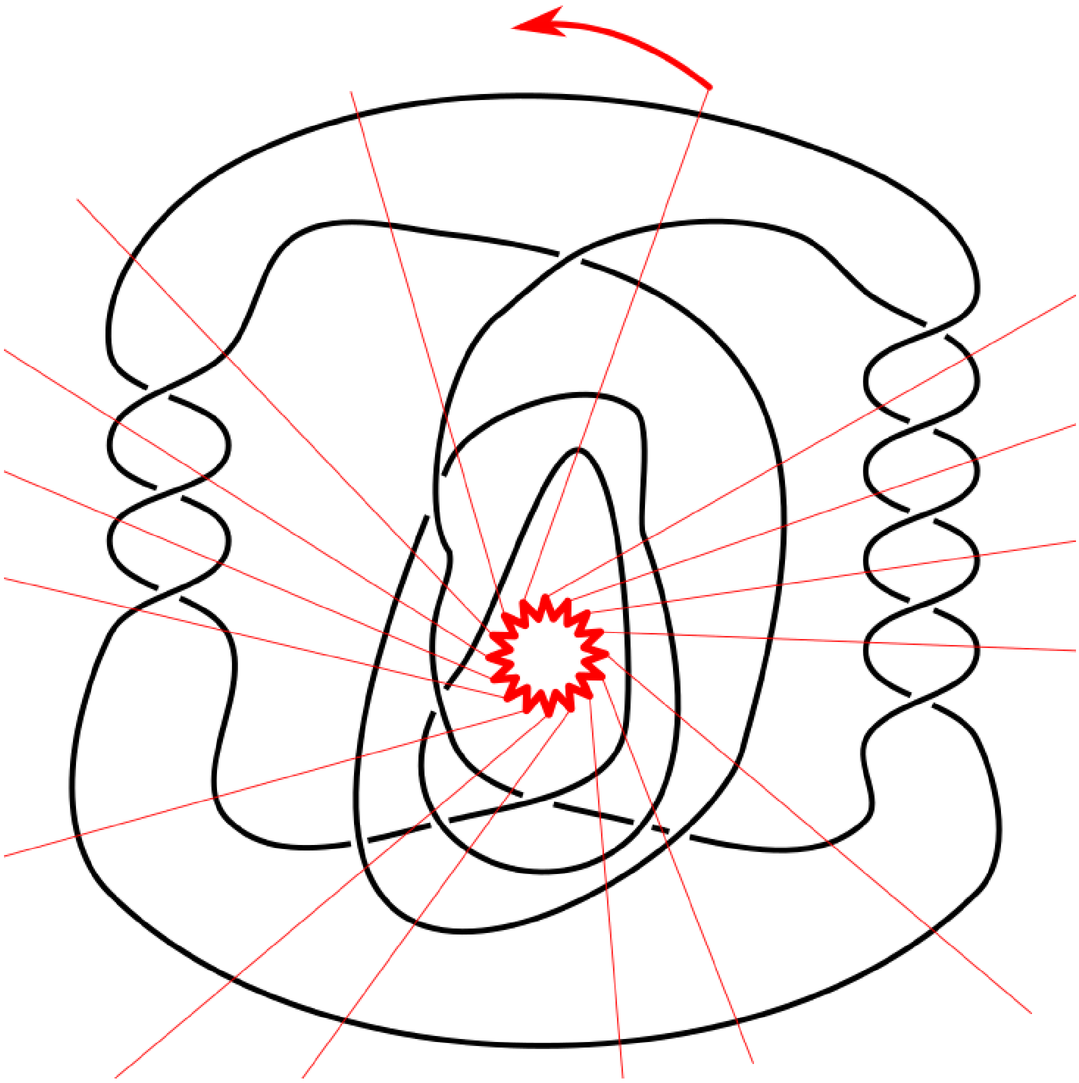


Ilustración 101. P(3, 6, 5) visto como una trenza.

Esta ilustración es de gran importancia, ya que también lleva implícito el algoritmo de Nudos P(1, par, 1), Nudos P(1, par, impar) y de Nudos P(impar, par, impar). Nos da

-2, -3, -4, 1, 1, 1, 2, 3, -4, -3, -2, 1, 1, 1, 1, 1

Se confirma la hipótesis de que modificar el valor de la columna izquierda lo único que hace es repetir el 1 de la segunda agrupación. Es decir, las columnas izquierda y derecha, al ser externas, nos modifican el número de unos, mientras que la central nos modifica los cruces que se producen entre dichas columnas.

$$\boxed{-2 \downarrow -\frac{|b|+2}{2}, 1^a, 2 \uparrow \frac{|b|}{2}, -\frac{|b|+2}{2} \uparrow -2, 1^c}$$

Ecuación 3. Algoritmo para P(Impar, Par, Impar).

Ahora bien, ¿qué ocurre si todas nuestras columnas tienen signo negativo? Pues que la trenza tendrá todos los signos invertidos. ¿Y si es una columna concreta la de signo negativo? Vamos a verlo.

Al estudiar la Ilustración 101, observamos que si nuestra primera columna vale -3 en vez de 3, el único cambio que se produce es que nuestros unos de la agrupación de esta columna se convierten en -1. Exactamente pasa lo mismo con la columna derecha.

El signo de la columna central invierte el signo de los grupos en los que se encuentra.

Es decir, si

P (3, 6, 5) da:	-2, -3, -4, 1, 1, 1, 2, 3, -4, -3, -2, 1, 1, 1, 1, 1

entonces

P (-3, 6, 5) da:	-2, -3, -4, -1, -1, -1, 2, 3, -4, -3, -2, 1, 1, 1, 1, 1
P (3, 6, -5) da:	-2, -3, -4, 1, 1, 1, 2, 3, -4, -3, -2, -1, -1, -1, -1, -1
P (-3, 6, -5) da:	-2, -3, -4, -1, -1, -1, 2, 3, -4, -3, -2, -1, -1, -1, -1, -1
P (3, -6, 5) da:	2, 3, 4, 1, 1, 1, -2, -3, 4, 3, 2, 1, 1, 1, 1, 1
P (-3, -6, -5) da:	2, 3, 4, -1, -1, -1, -2, -3, 4, 3, 2, -1, -1, -1, -1, -1

Nudos P(par, impar, impar)

Ahora en vez de tener la entrada par en la columna central vamos a tenerla en la izquierda. Ahora que ya tenemos algo de soltura, podemos intentar atacar directamente un nudo con las 3 columnas variables.

En su momento cometí el error de estudiar el nudo P(6, 3, 3). ¿Cuál es el inconveniente? Pues como ya hemos visto, que 2 de mis variables sean la misma es muy poco recomendable, ya que habrá agrupaciones de 3 números, y no sabré si dependen de mi columna central o izquierda. Por otra parte, el 3 es un número muy pequeño. Por ello, se va a ejemplificar con el nudo P(6, 5, 7).

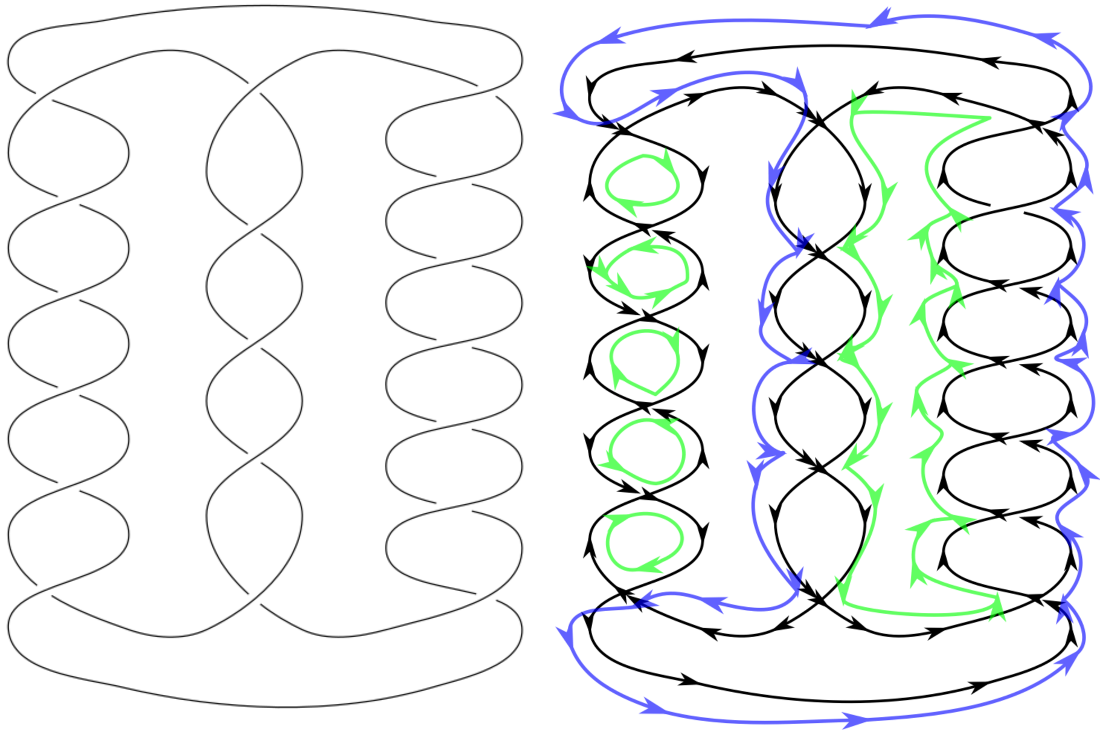


Ilustración 102. P(6, 5, 7).

Lo que a primera vista parece que va a ser un problema trivial que resolveremos de la misma manera que P(Impar, Par, Impar) no lo es en absoluto, cosa que descubrimos con los círculos de Seifert.

¿Qué hacer ahora? No podemos hacer movimientos básicos en la columna central, porque esta vez aumentaría el número de círculos y no conseguiríamos nada. Por otro lado, el método de las reducciones no es nada apetecible después de haber descubierto una opción mucho mejor.

Si nos fijamos entre la columna central y derecha ya hay un círculo verde dentro del azul. Siempre hay que tener en mente que este es nuestro objetivo, que *los círculos de Seifert sean concéntricos y compatibles* pues en ese momento tendremos la proyección de la trenza cerrada.

Así que nuestro objetivo es operar la columna izquierda para que englobe dichos círculos.

Después de varios ensayos de prueba y error llego al método siguiente:

Para empezar, hacer movimientos básicos (recordando que cuando se crucen con otra cuerda siempre irán por encima) en la columna de la izquierda, que es la que tiene círculos que son una gran molestia en su estado actual. Haremos *a/2*—1 movimientos básicos, es decir, *6/2-1=2*.

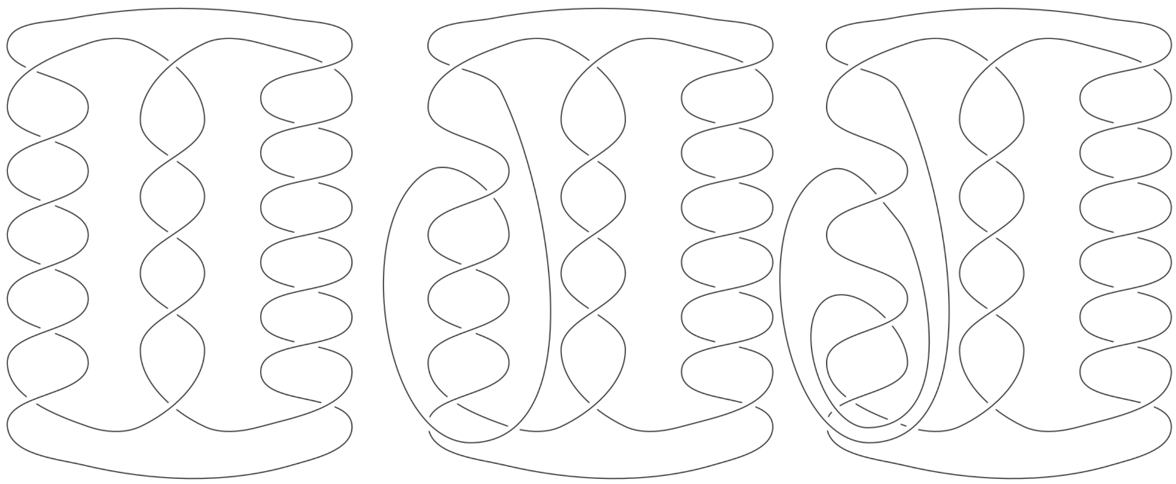


Ilustración 103. Dos movimientos básicos en P(6, 5, 7).

Ahora estamos en un momento crítico: hemos gastado nuestro movimiento estrella, pero, ¿hemos conseguido algo? Estudiemos los círculos de Seifert.

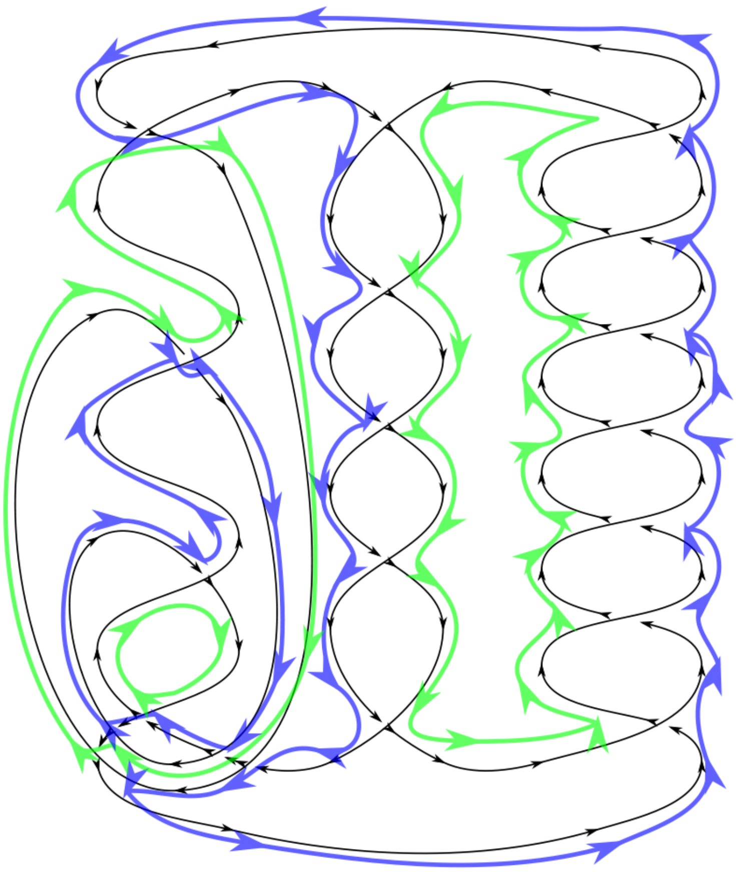


Ilustración 104. P(6, 5, 7).

De nuevo sorprende la utilidad de los movimientos básicos. No sólo nos han reducido los círculos de Seifert de la columna izquierda de 5 a 3 (tras 2 movimientos básicos), si no que ahora *todos los círculos de Seifert son compatibles entre sí*. Ahora la pregunta es, ¿dónde está la trenza? Sólo tenemos que recordar nuestra segunda herramienta, mencionada al principio de este capítulo, los movimientos de polo sur.

Vamos a hacer a/2 = 3 movimientos de polo sur que van a hacer que los círculos de Seifert que tenemos en la columna izquierda engloben a los otros. Vamos a verlo.

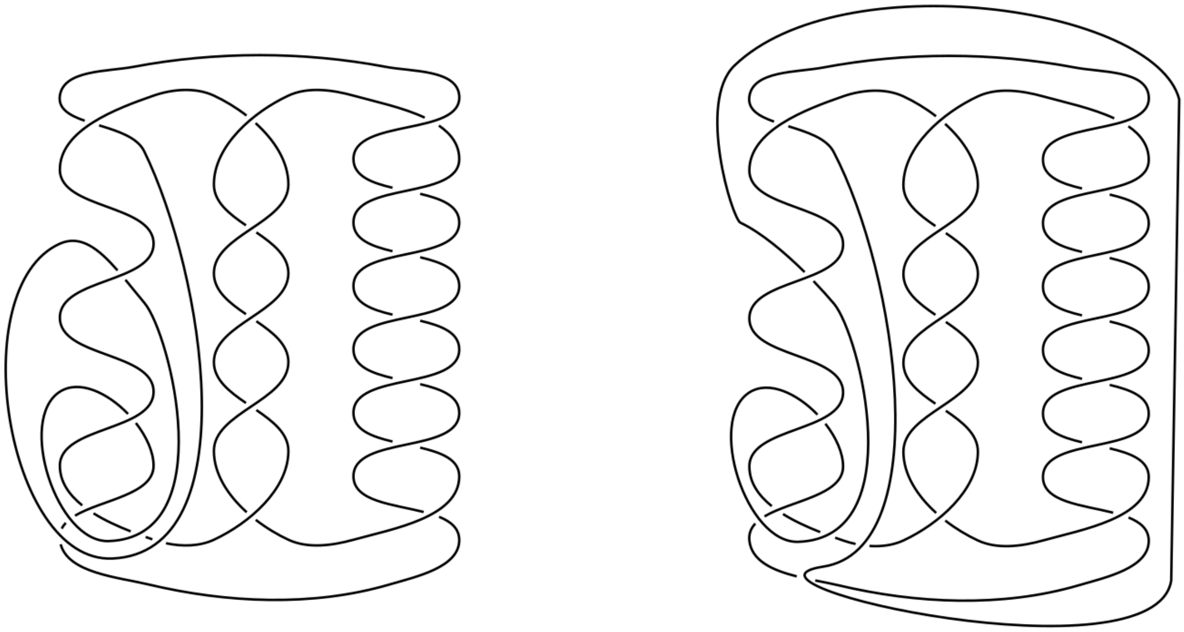


Ilustración 105. Primer movimiento de polo sur en P (6, 5, 7).

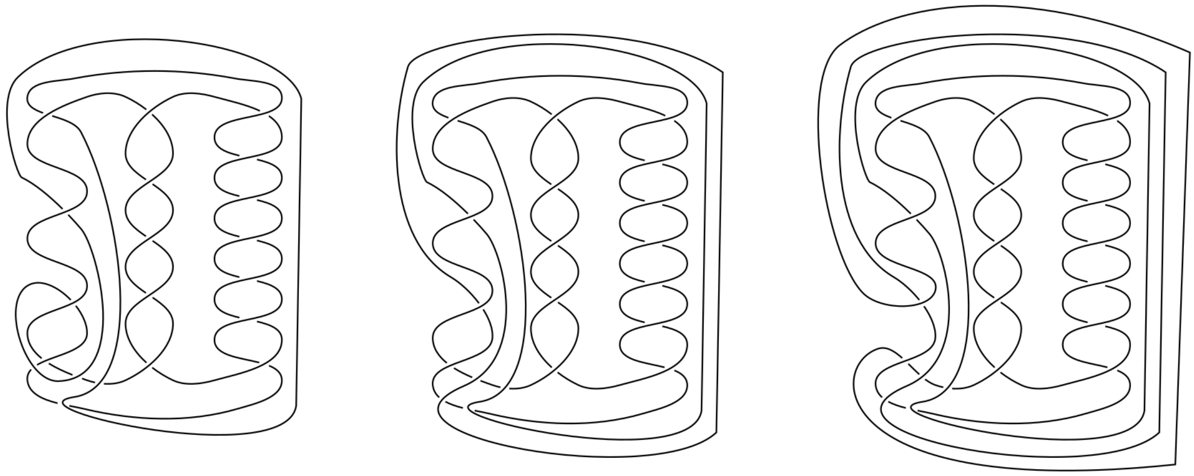


Ilustración 106. Segundo y tercer movimiento de polo sur.

Aunque no lo parezca, ya tenemos nuestra trenza, y su centro está entre la columna central y la derecha. Comprobémoslo.

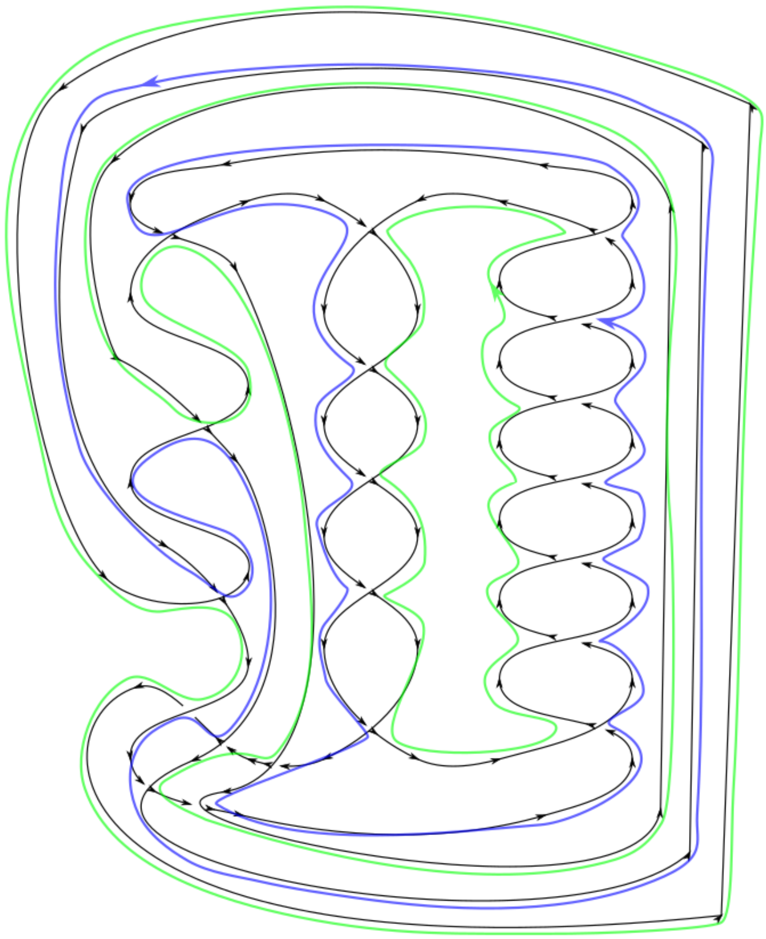


Ilustración 107. P(6, 5, 7) como trenza clausurada.

Tenemos círculos compatibles y concéntricos, así que podemos afirmar que tenemos la trenza. Vamos a moverla un poco para ver mejor los cruces.

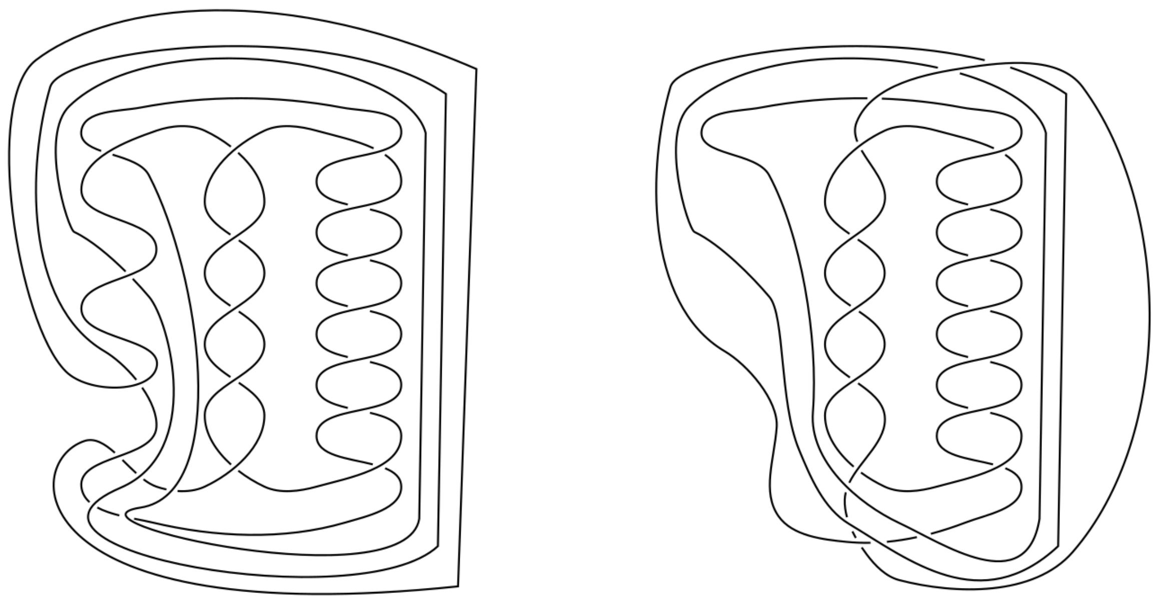


Ilustración 108. Enlace P(6, 5, 7) como clausura de una trenza.

Cuento 20 cruces así que usaré una estrella de 20 puntas, aunque esta vez no nos quedará tan visual como los casos previos.

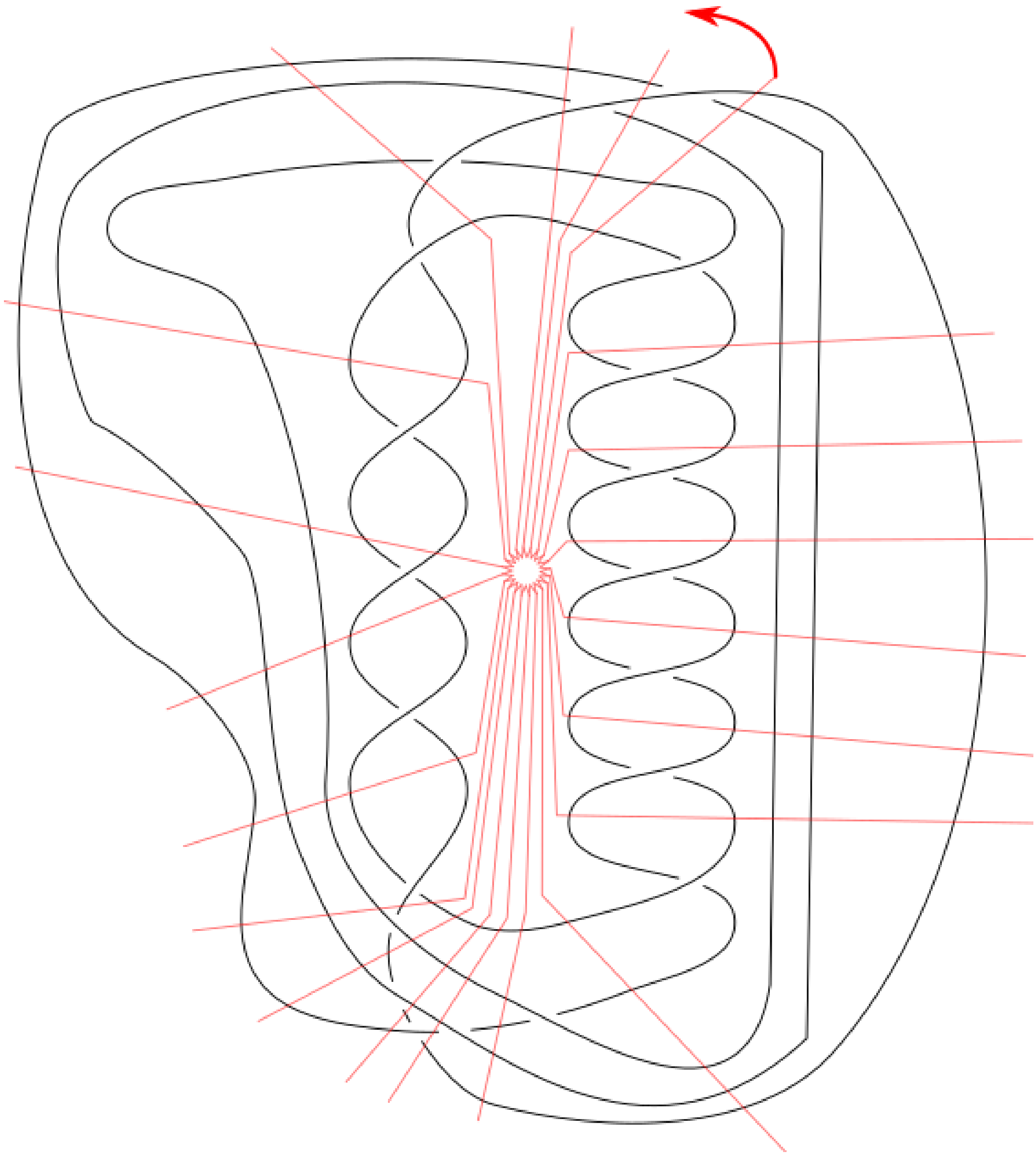


Ilustración 109. Cruces de P(6, 5, 7) en trenza.

Si anotamos los cruces tenemos

-1, -2, -3, 4, 4, 4, 4, 4, -3, -2, -1, 2, 3, 4, 4, 4, 4, 4, 4, 4

¿Cómo estudiamos el caso general?

Empecemos por las dos series de 4 que se repiten. Está claro que la primera serie se repite tantas veces como cruces tenga la columna central, mientras que la segunda tantas veces como cruces tenga la columna izquierda. Un cambio de signo cambiará el signo de los 4.

¿Y cuál es la influencia del 6, o, mejor dicho, de la columna izquierda? Si recordamos el proceso, hemos tenido que hacer 2 movimientos básicos en esta columna. Luego hemos usado los círculos que hemos obtenido para envolver al nudo con 3 movimientos de polo sur. Por lo tanto, llegamos a la conclusión de que las columnas central y derecha estarán separadas del exterior por tantas cuerdas como movimientos de polo sur hayamos hecho, es decir, a/2.

Y ya tenemos la clave, ya que en nuestro caso es 3, y por eso los siguientes cruces, los de las columnas central y derecha, valen 4.

Analicemos el algoritmo. Siempre tendremos la cuerda que empieza por el exterior y entra sobre las demás hasta llegar a la columna central. En nuestro caso, -1,-2,-3. A continuación ocurren los cruces de la columna central. Después de recorrerla otra cuerda sale realizando el camino inverso, -3,-2,-1. Finalmente, otra va a la columna izquierda con 2,3 y realiza los cruces de la columna izquierda.

Es decir, nuestro algoritmo será:

$$\boxed{-1 \downarrow -\frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{b}, -\frac{|a|}{2} \uparrow -1, 2 \uparrow \frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{c}}$$

Ecuación 4. Algoritmo para P(par, impar, impar).

Como ya hemos dicho, el signo de b y de c cambiará el valor de $\frac{a+2}{2}$ en su respectiva serie. El signo de a cambiará el signo de las otras 2 agrupaciones si es negativo.

Hagamos un test del algoritmo. Recordemos: (a, b, c) = (6, 5, 7).

$$-1 \downarrow -\frac{6}{2}, \left(\frac{6+2}{2}\right)^{5}, -\frac{6}{2} \uparrow -1, 2 \uparrow \frac{6}{2}, \left(\frac{6+2}{2}\right)^{7}$$

$$-1, -2, -3, 4, 4, 4, 4, 4, -3, -2, -1, 2, 3, 4, 4, 4, 4, 4, 4, 4$$

El algoritmo se verifica.

Nudos P(impar, impar, par)

Ahora voy a hacer una pequeña demostración de la propiedad simétrica de los Pretzel. Si estudiamos los círculos de Seifert de P(7, 5, 6) tenemos:

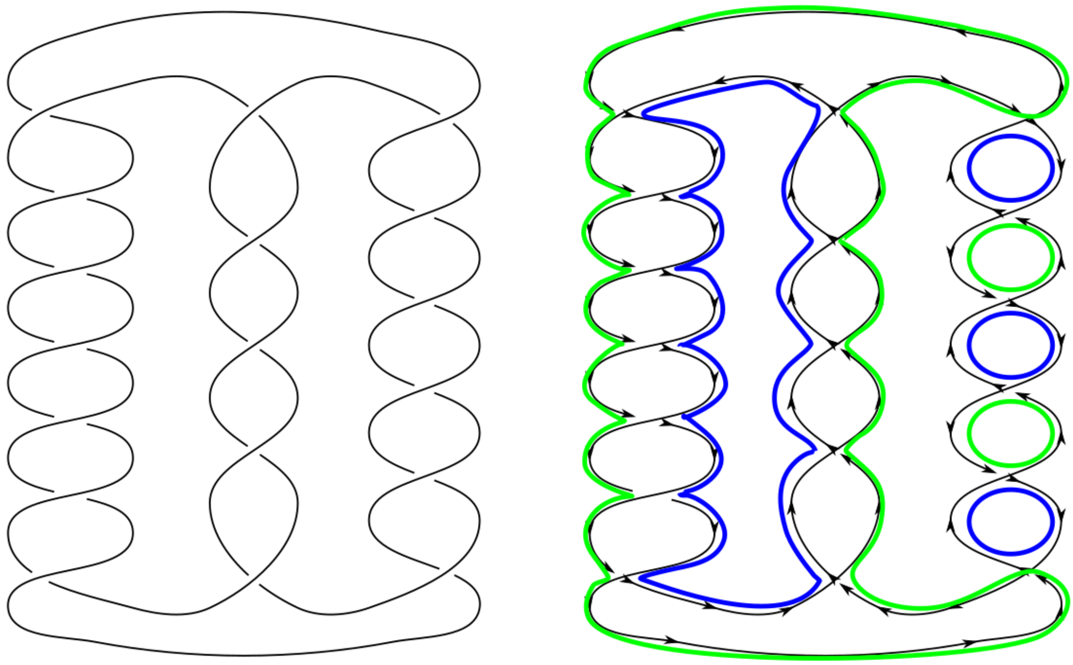


Ilustración 110. P(7, 5, 6).

Este nudo parece estar pidiendo que hagamos con él lo mismo que con Nudos P(par, impar, impar), es decir, 2 movimientos básicos en la columna derecha y luego 3 movimientos de polo sur. Pero vamos a ser un poco pícaros y usemos una idea feliz. ¿Qué diferencia hay entre P(6, 5, 7) y P(7, 5, 6)? Como podemos ver en la siguiente ilustración, si giramos una de las proyecciones 180º obtenemos la otra.

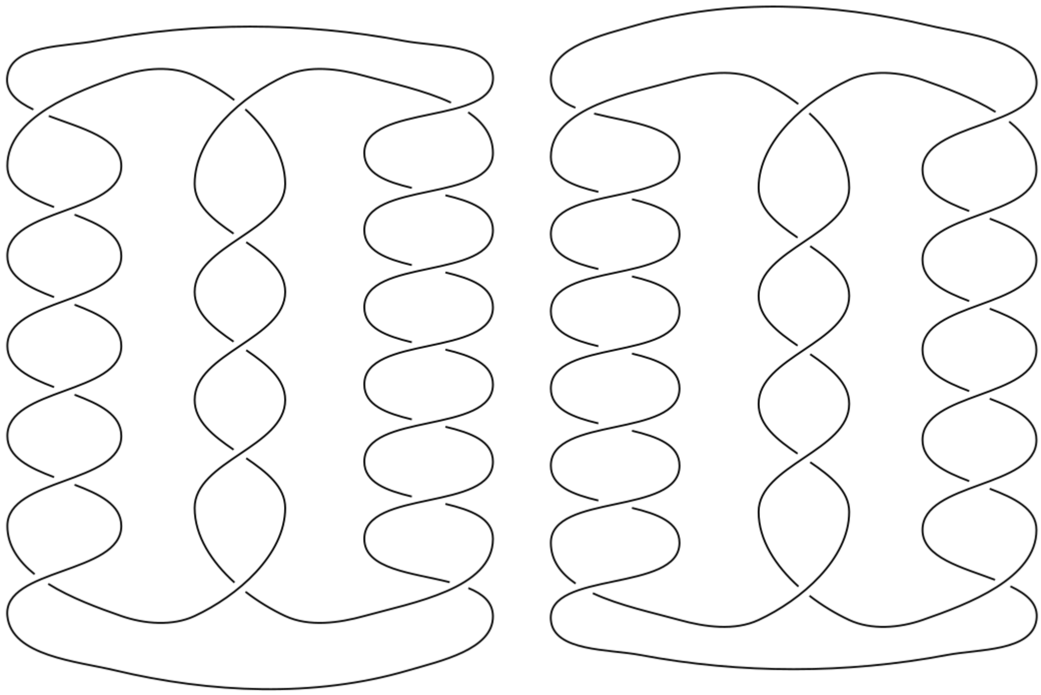


Ilustración 111. P(6, 5, 7) y P(7, 5, 6).

Pues bien, el algoritmo de P(Par, impar, impar) nos sirve para P(Impar, impar, par), invirtiendo las entradas. Sólo tenemos que tener en cuenta que si lo hiciésemos a mano nuestra trenza comenzará en otro punto, pero las dos trenzas son la misma. Esto se reflejará en el capítulo de programación con Python, en el que llamaremos a la misma función para tratar los dos tipos de nudo.

Este hecho es independiente de si las entradas son pares o impares o si son negativas o positivas, como se ve en la siguiente figura.

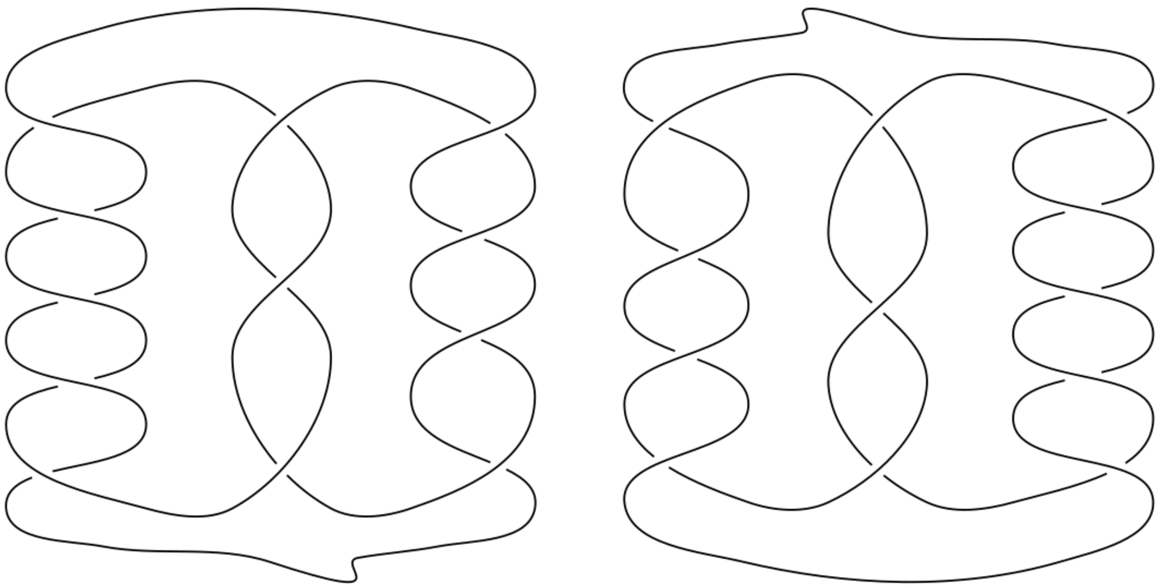


Ilustración 112. P(-5, 3, 4) y P(4, 3, -5) son el mismo nudo pretzel.

Nudos P(impar, impar, impar)

Estos tipos de nudos pretzel han sido, con diferencia, los más difíciles de todos. Empecé con P(1, 1, Impar), luego con P(1, 3, Impar), luego P(1, 5, Impar) y finalmente P(1, 7, Impar) sin ver ninguna progresión clara. No sólo no había ninguna progresión visible, si no que convertirlos en trenza ha sido muy complicado y no ha servido la estrategia usada hasta ahora.

Nudos P (1, impar, 1)

Primero se ha probado con el P (1, 3, 1), y para sorpresa de nadie, no ha servido de mucho para el caso general. Para resolverlo hay que hacer movimientos básicos en la columna central y un movimiento de polo sur. Se ha obtenido -1, -1, 2, -1, -2, -2.

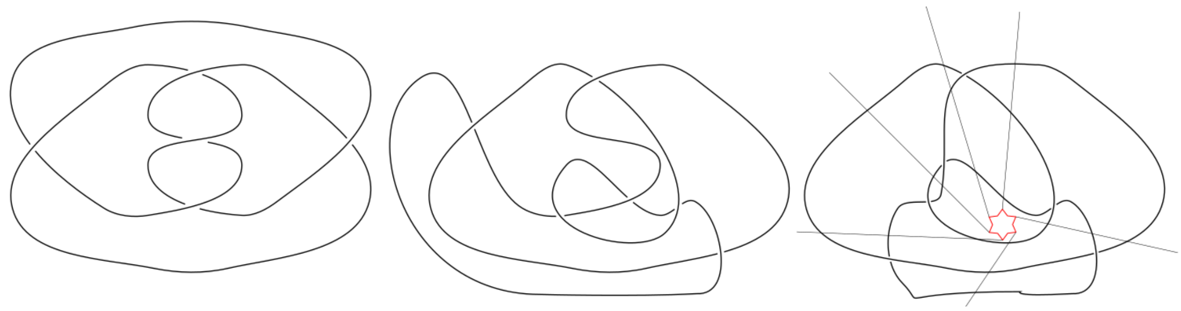


Ilustración 113. P (1, 3, 1).

P (1, 5, 1) requiere más trabajo, pero el resultado sigue sin mostrar ninguna progresión: -1, -1, 2, 3, -1, -2, -3, -3, -2

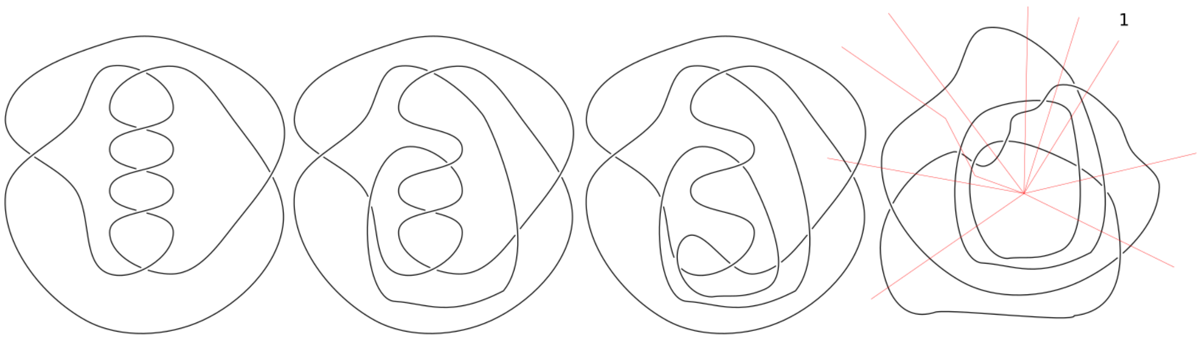


Ilustración 114. P (1,5,1).

Con P (1, 7, 1) ya vemos la progresión claramente: -1, -1, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2

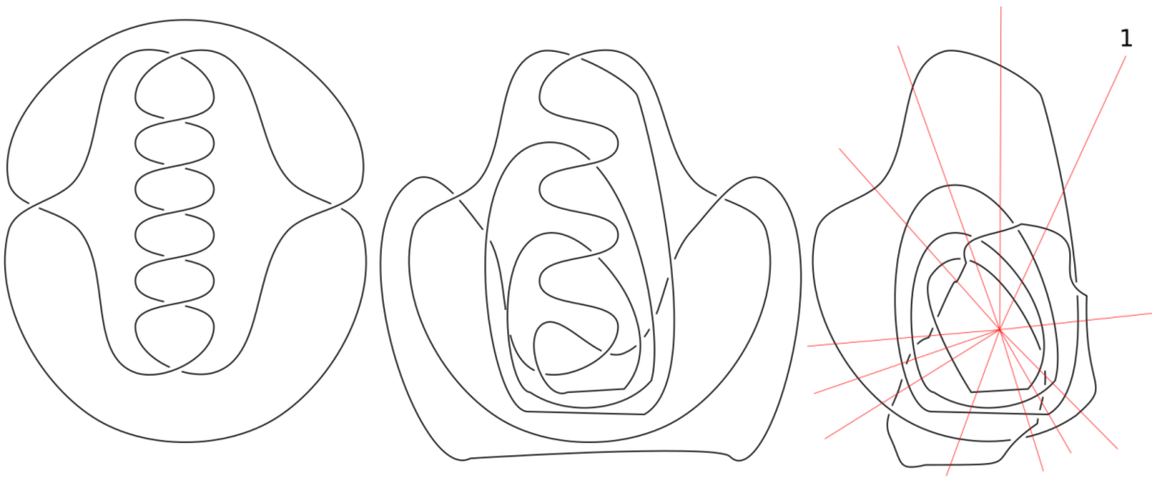


Ilustración 115. P (1, 7, 1).

$$-1, -1, 2 \uparrow \frac{|b|+1}{2}, -1 \downarrow -\frac{|b|+1}{2}, -\frac{|b|+1}{2} \uparrow -2$$

Ecuación 5. Algoritmo para P(1, impar, 1).

Hacemos la comprobación con P(1, 5, 1):

$$-1, -1, 2 \uparrow \frac{5+1}{2}, -1 \downarrow -\frac{5+1}{2}, -\frac{5+1}{2} \uparrow -2$$

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

Lo que coincide con el resultado experimental.

Estudiaremos la influencia de los signos en el caso general, que engloba a este.

Nudos P(1, 1, impar)

Curiosamente, vamos a ver que éstos se resuelven exactamente igual que los anteriores. Se realizan movimientos básicos en la columna que tenga los cruces y se realiza un movimiento de polo sur.

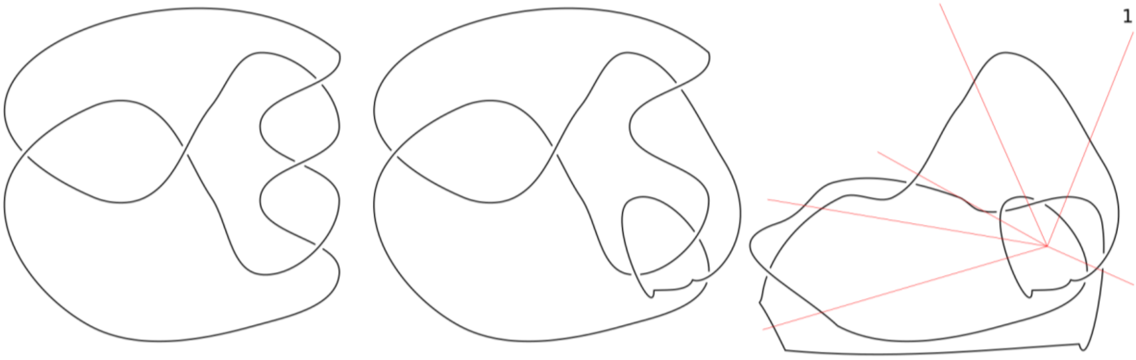


Ilustración 116. Un movimiento de polo sur y uno básico en la columna derecha de P(1, 1, 3).

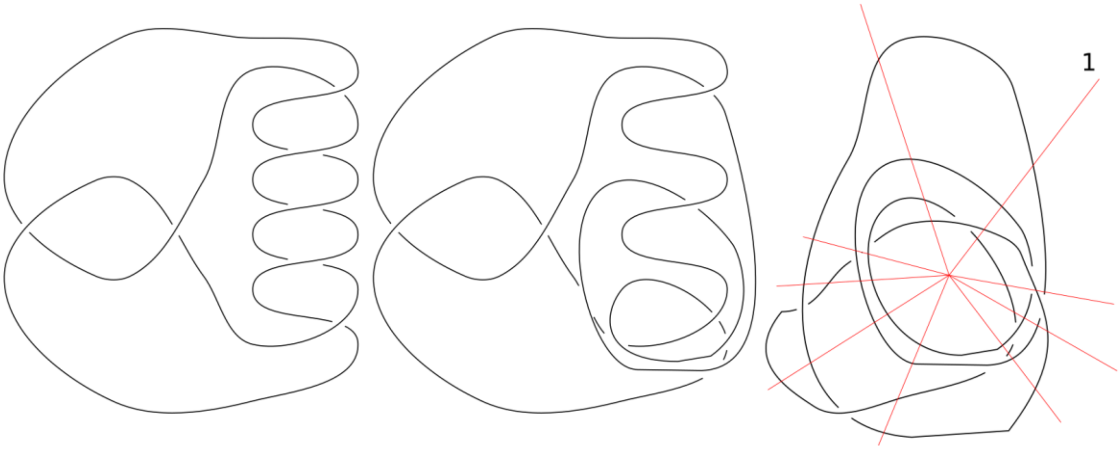


Ilustración 117. Dos movimientos básicos en la columna derecha de P(1, 1, 5).

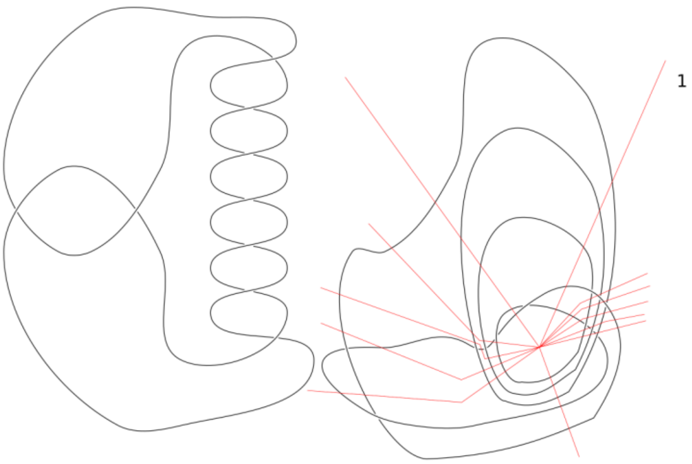


Ilustración 118. Tres movimientos básicos en la columna derecha de P(1 ,1, 7).

Si vemos juntos los valores de las trenzas es muy fácil intuir el algoritmo:

P (1,1,3): -1, 2, -1, -2, -2, -1

P (1,1,5): -1, 2, 3, -1, -2, -3, -3, -2, -1

P (1,1,7): -1, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -1

$$-1, 2 \uparrow \frac{|c|+1}{2}, -1 \downarrow -\frac{|c|+1}{2}, -\frac{|c|+1}{2} \uparrow -1$$

Ecuación 6. Algoritmo para P(1, 1, Impar).

Hacemos la comprobación con P(1, 1, 5):

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

Lo que coincide con el resultado experimental.

Estudiaremos la influencia de los signos en el caso general.

Nudos P(1, impar, 3)

Ahora ya se empieza a poner interesante la cosa, pues tenemos que hacer movimientos básicos en 2 columnas, en concreto en las que tienen cruces, ya que tienen círculos solitarios que queremos eliminar:

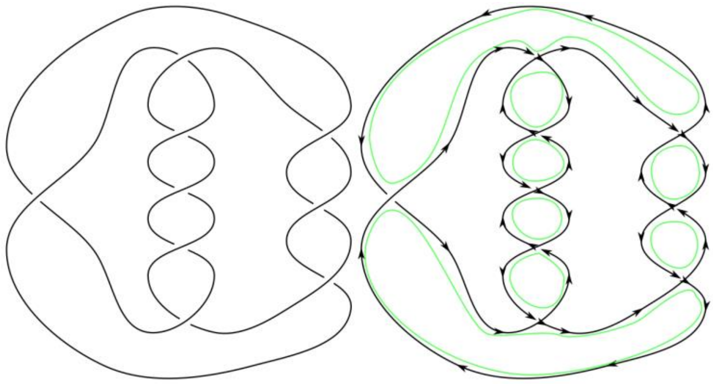


Ilustración 119. P (1, 5, 3).

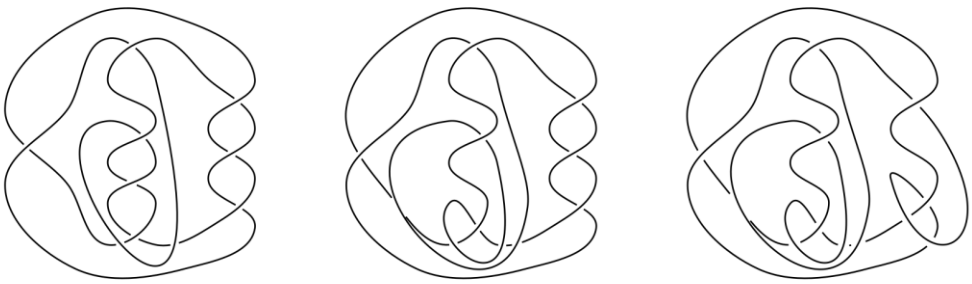


Ilustración 120. Dos movimientos básicos en la columna central y uno en la derecha en P (1, 5, 3).

Si estudiamos ahora los círculos de Seifert observaremos que tenemos uno superior, uno inferior, 2 círculos en la columna central y uno en la derecha. Podemos hacer un movimiento de polo sur con el superior para que englobe a todo y luego hacer reducciones clásicas entre los círculos internos para arreglar su incompatibilidad.

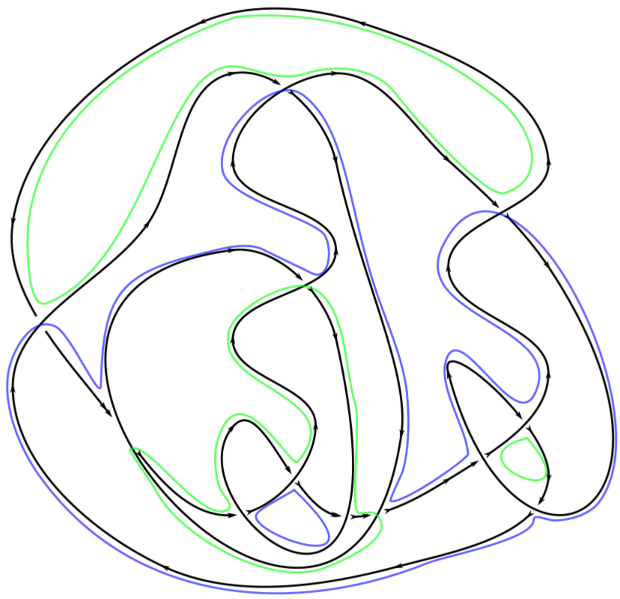


Ilustración 121. P(1, 5, 3).

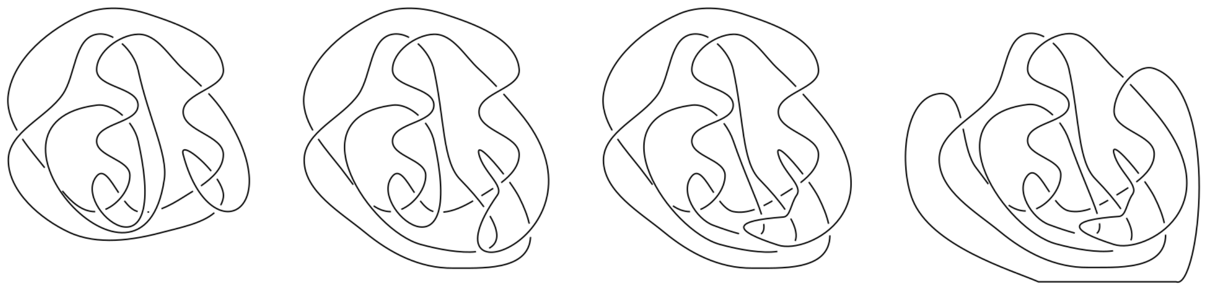


Ilustración 122. Dos movimientos de reducción en el interior y un movimiento de polo sur en el exterior.

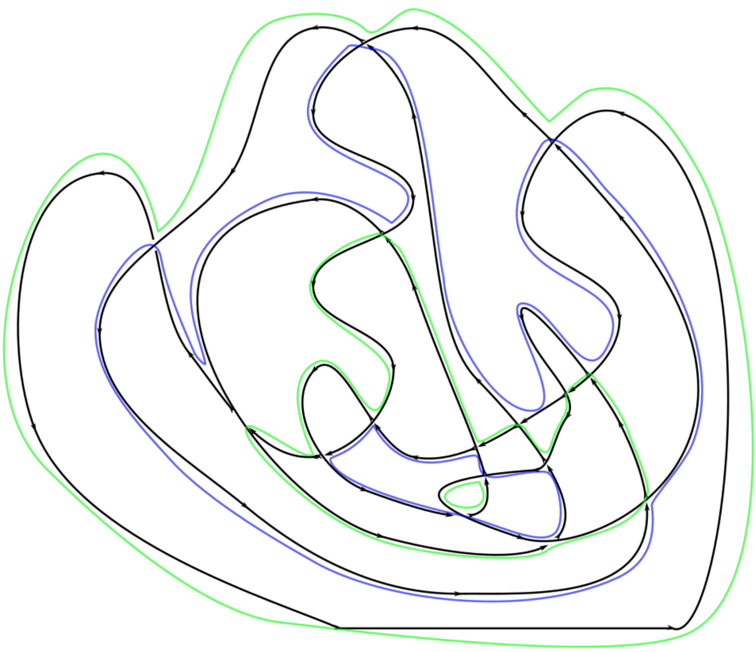


Ilustración 123. P (1, 5, 3) orientado con sus círculos de Seifert.

Como podemos observar ya tenemos nuestro diagrama de trenza. Ahora mismo es un poco difícil de ver, pero con esmero podemos trazar la división de sus cruces.

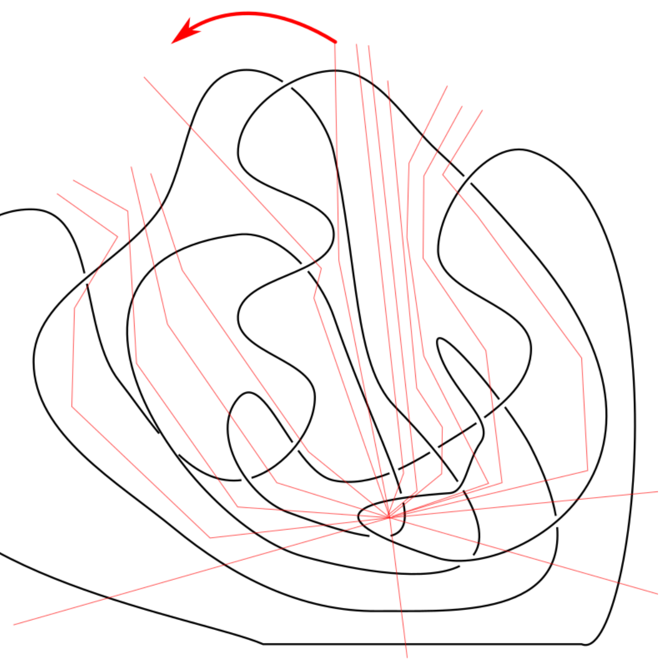


Ilustración 124. P (1, 5, 3) y sus cruces de trenza.

Anotando los cruces tenemos

-1, -2, -3, -3, -2,-1, 4, 3, 2, -1, -2, -2, -3, -4, 2, 3

Para P (1, 7, 3) seguiremos el mismo proceso: tantos movimientos básicos como sean necesarios en la columna central ((b-1)/2) y uno en la columna derecha; luego reducción de los círculos resultado.

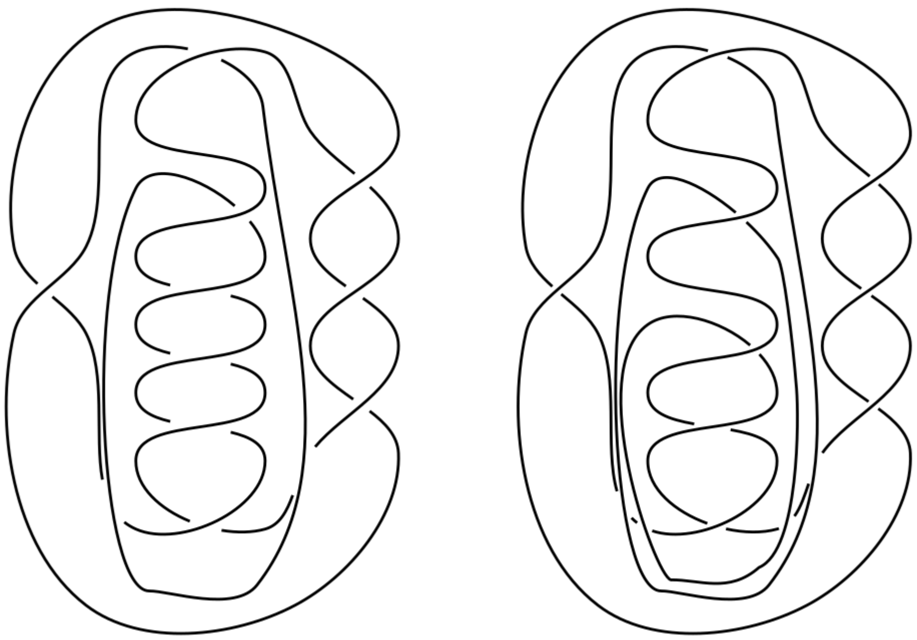


Ilustración 125. Dos movimientos básicos en la columna central de P (1, 7, 3).

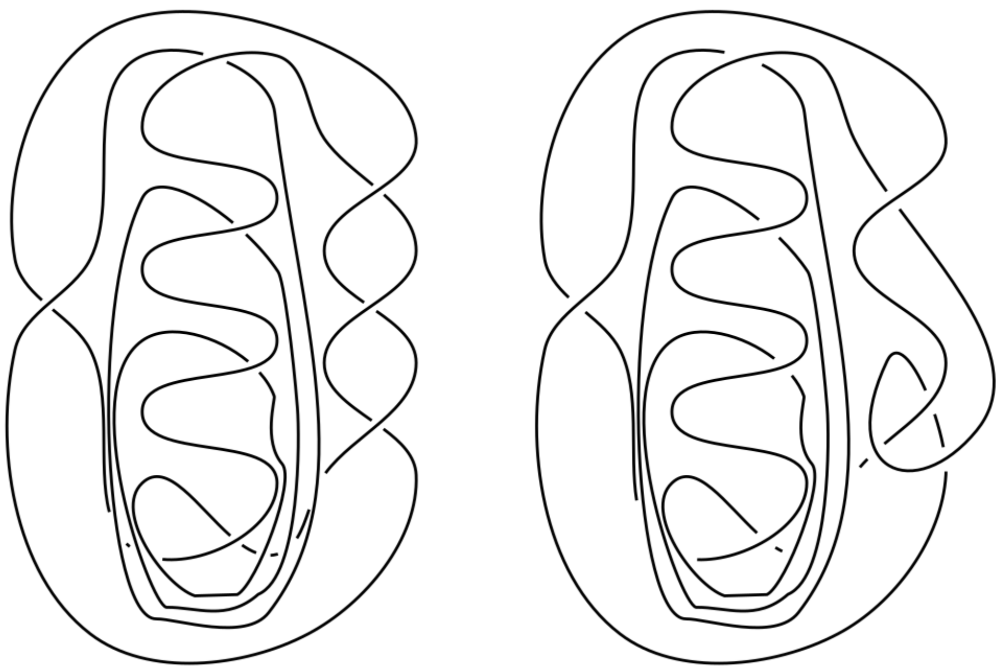


Ilustración 126. Tercer movimiento básico en la columna central y primero en la derecha.

Ahora sólo nos queda un movimiento de polo sur, y las reducciones:

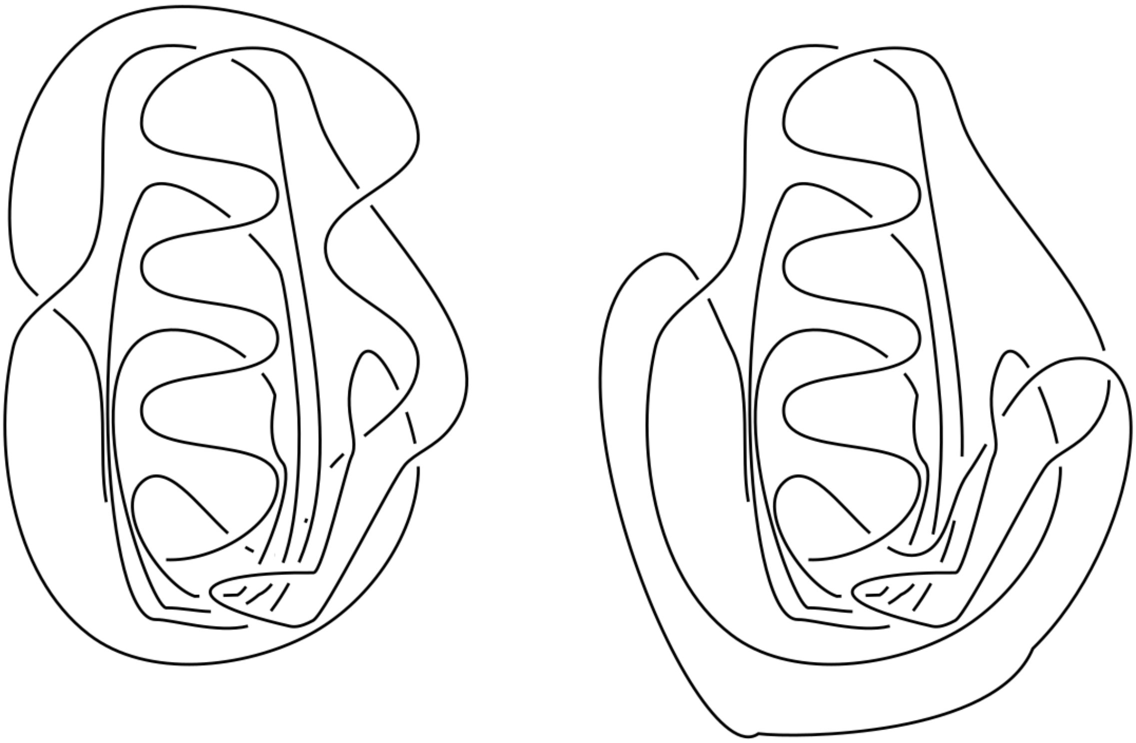


Ilustración 127. Movimiento de reducción y de polo sur.

Esta trenza es demasiado compleja y habrá que modificarla un poco antes de poder leerla.

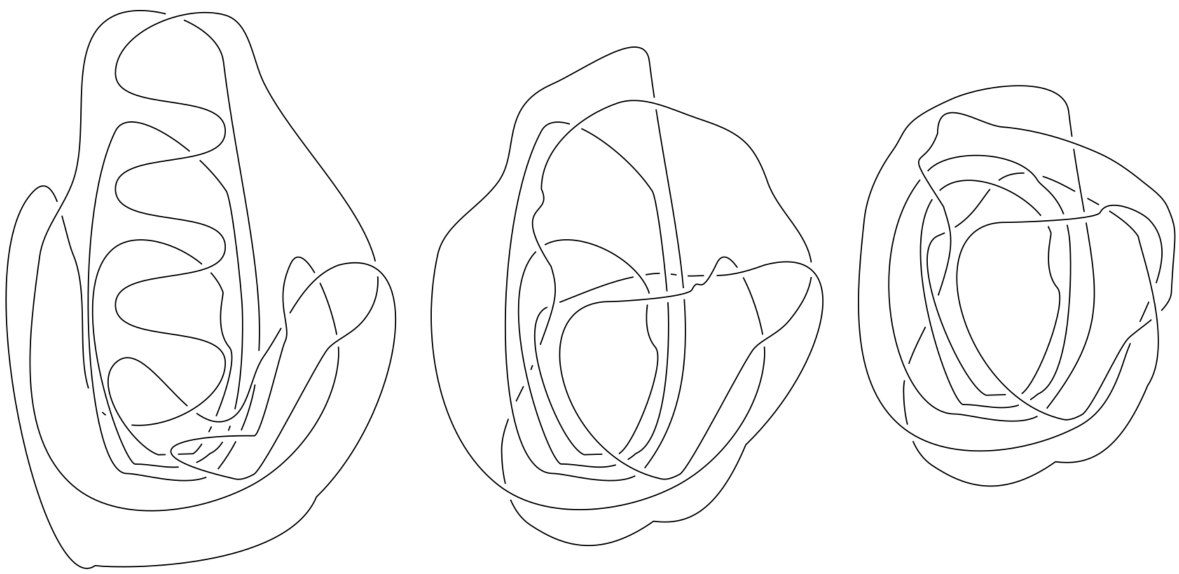


Ilustración 128. Movimientos para simplificar la trenza de P (1, 7, 3).

Ahora ya podemos estudiarla:

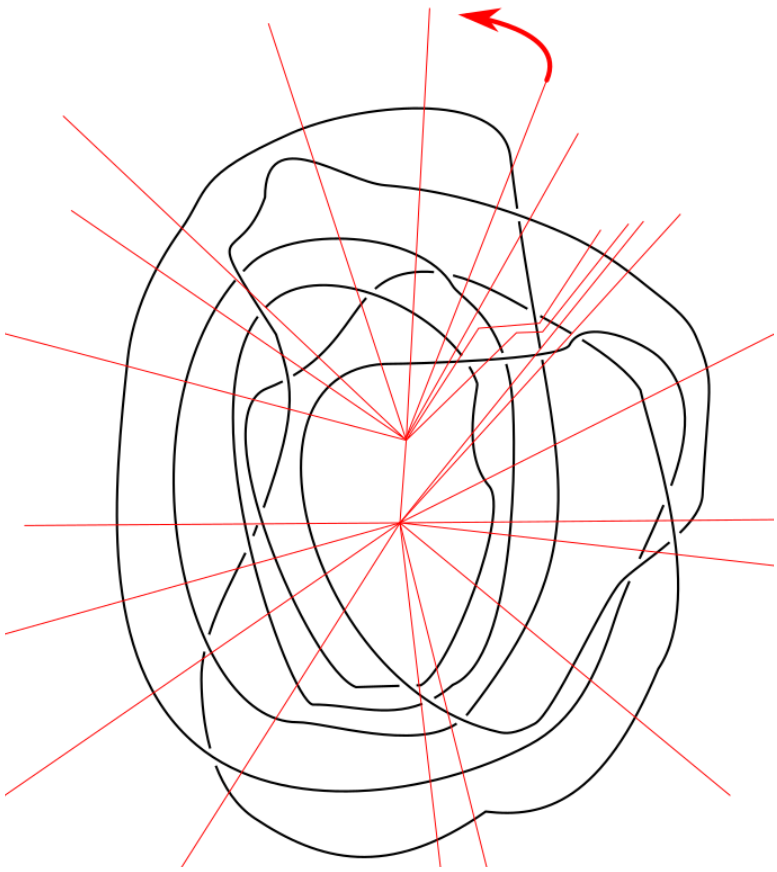


Ilustración 129. Trenza de P (1, 7, 3).

Si comparamos las trenzas que tenemos para buscar la progresión:

P (1,1,3): -1, 2, -1, -2, -2, -1

P (1,3,3): -1, 3, 2, -1, -2, -2, -3, 2, -1, -2, -2

P (1,5,3): -1, 4, 3, 2, -1, -2, -2, -3, -4, 2, 3, -1, -2, -3, -3, -2

P (1,7,3): -1, 5, 4, 3, 2, -1, -2, -2, -3, -4, -5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2

Recordando que b es el valor de la columna central, y comenzando un cruce después:

$$\boxed{\frac{|b|+3}{2} \downarrow 2, -1, -2, -2 \downarrow -\frac{|b|+3}{2}, 2 \uparrow \frac{|b|+1}{2}, -1 \downarrow -\frac{|b|+1}{2}, -\frac{|b|+1}{2} \uparrow -1}$$

Ecuación 7. Algoritmo para P(1, impar, 3).

Para P(1, 5, 3):

$$\frac{5+3}{2} \downarrow 2, -1, -2, -2 \downarrow -\frac{5+3}{2}, 2 \uparrow \frac{5+1}{2}, -1 \downarrow -\frac{5+1}{2}, -\frac{5+1}{2} \uparrow -1$$

$$-1,4,3,2,-1,-2,-2,-3,-4,2,3,-1,-2,-3,-3,-2$$

Coincide con el resultado experimental, por lo que el algoritmo se verifica.

Estudiaremos la influencia de los signos en el caso general, que engloba a éste.

Nudos P(impar, impar, impar)

Para poner ir concienciando al lector, mientras que el P (3, 6, 5) tiene menos de 20 cruces, el P (7, 3, 7) tiene nada más y nada menos que 54 cruces. Por si fuera poco, de poco nos sirve ese nudo, ya que tiene dos variables iguales y otra muy pequeña. Así que para afrontar esta búsqueda se ha usado el P(9, 5, 7), de números distintos y de tamaño considerable.

El método que viene a continuación es bastante difícil de asimilar y seguir, pero viene de varios intentos de prueba y error y es el método definitivo para atacar este tipo de nudos.

Como siempre, empezamos creando nuestro nudo, orientándolo y estudiando los círculos de Seifert:

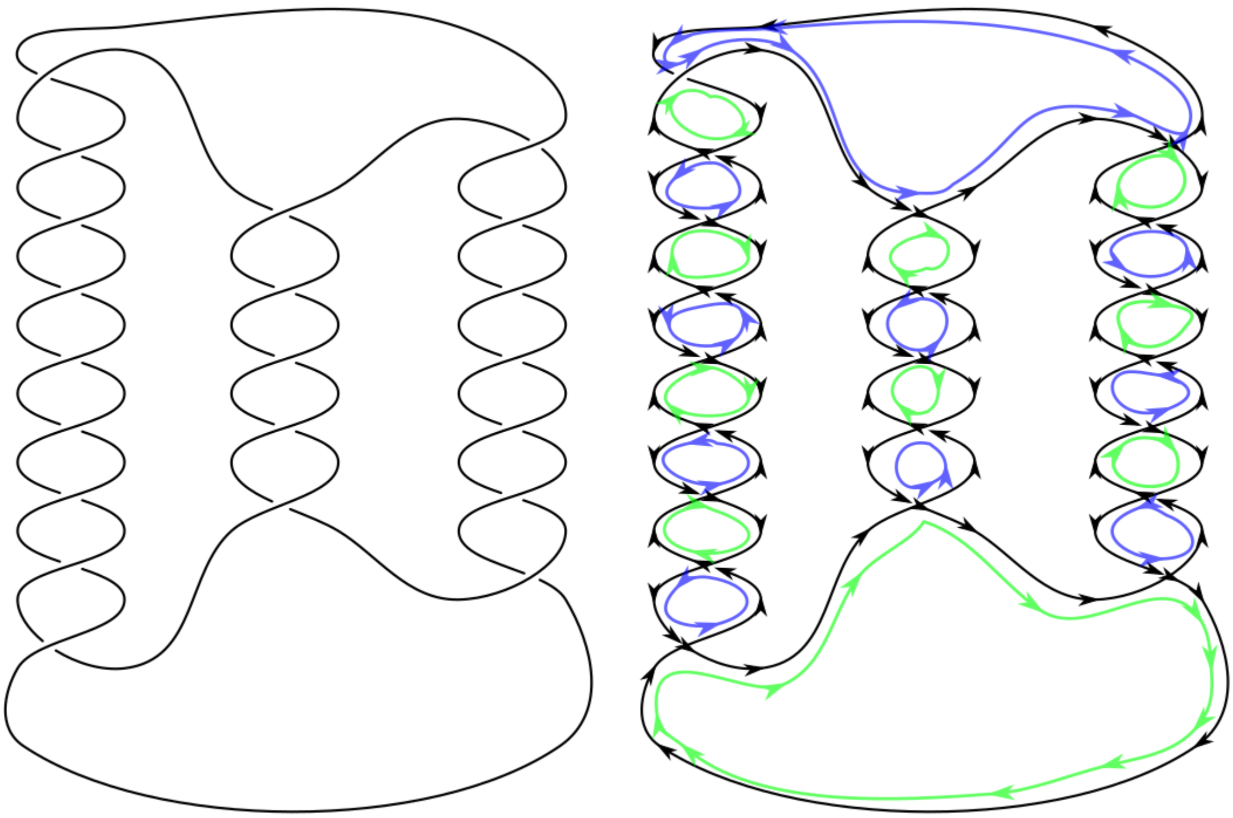


Ilustración 130. P(9, 5, 7).

Como podemos apreciar no parece halagüeño. Para empezar, no hay ni un solo círculo dentro de otro, pero tenemos torres de círculos en cada columna, así que podrían funcionar los movimientos básicos.

Comenzamos con un movimiento de polo sur sobre el círculo superior (azul) para que englobe a todos los demás. Será compatible con el verde inferior, pero quedan muchos círculos incompatibles por tratar.

A continuación, hacemos movimientos básicos en las 3 columnas. ¿Cuántos? (|N|-1)/2, siendo N el valor de la columna:

$$(9-1)/2 = 4,\ (5-1)/2 = 2,\ (7-1)/2 = 3.$$

Muy importante: hay que recordar en todo momento que, si nuestro nudo es a, b, c; a' serán los movimientos básicos de la primera columna, b' los de la segunda y c' los de la tercera. Esto será esencial más adelante en el desarrollo del algoritmo. En este caso a'=4, b'=2, c'=3.

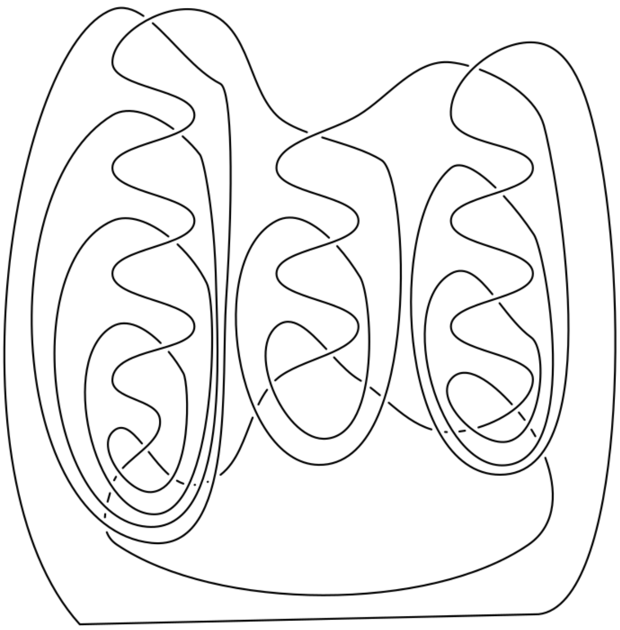


Ilustración 131. Cuatro movimientos básicos a la izquierda, dos en el centro y tres a la derecha en P(9, 5, 7).

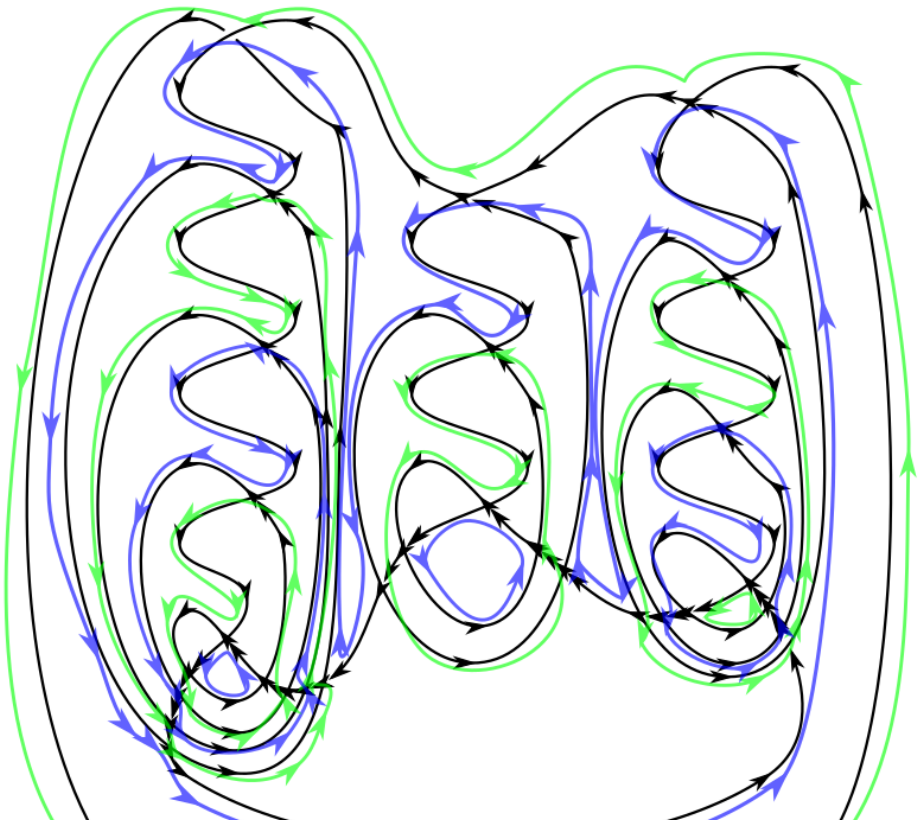


Ilustración 132. P(9, 5, 7) y sus círculos de Seifert.

Ahora ya tiene mejor forma. Tenemos un círculo exterior (azul) anti horario compatible con otro círculo (verde). Luego tenemos *3 grupos de círculos, uno por cada columna*, todos anti horarios. El primero de 4 círculos (a’), el del centro de 2 círculos (b’) y a la derecha 3 círculos (c’). Eso significa que son incompatibles los de la izquierda con los del centro, con los de la derecha, y los del centro con los de la izquierda. Es decir, todos con todos. Sin dejar de recordar que buscamos hacer que todos los círculos sean concéntricos, vamos a hacer reducciones múltiples para no ir una a una como en los casos anteriores.

Primero se adaptarán los círculos para que sean más manejables, luego se pasarán los círculos *centrales por debajo* de los de la columna izquierda y derecha mediante movimientos de reducción múltiple.

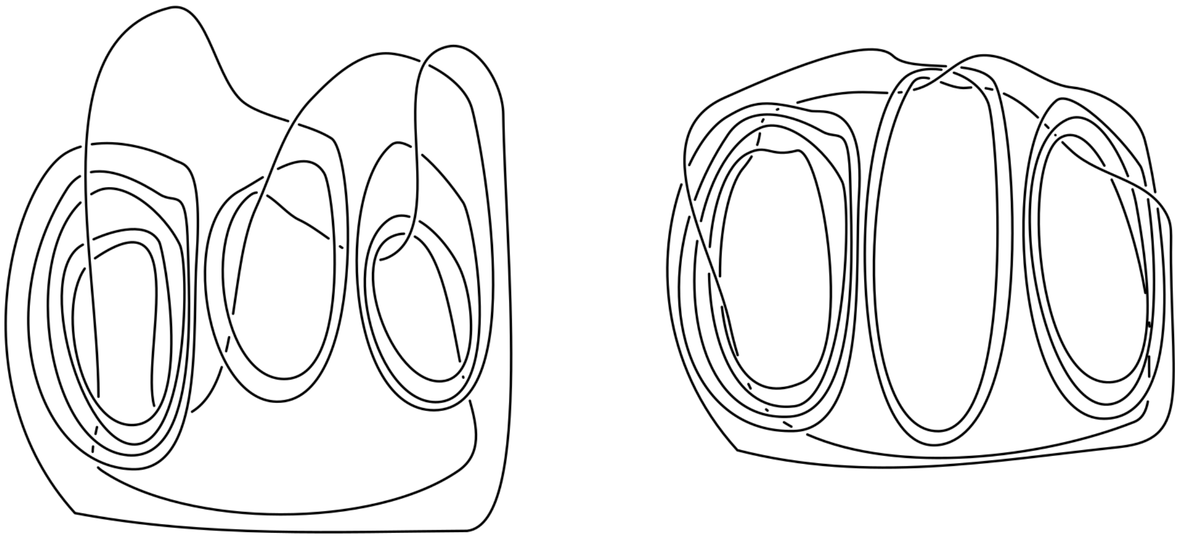


Ilustración 133. Movimientos para limpiar el dibujo y facilitar las reducciones futuras.

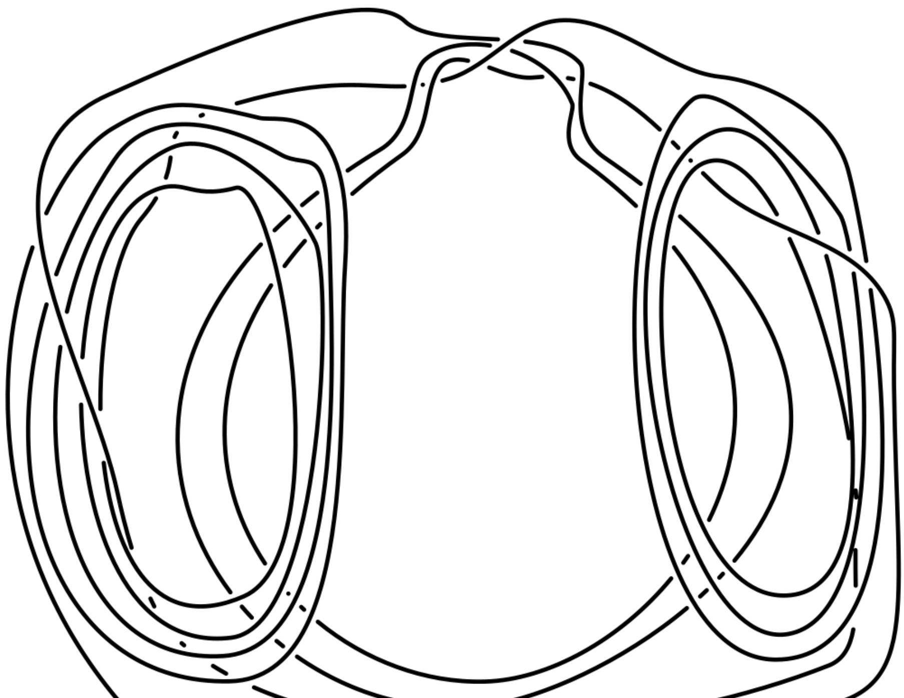


Ilustración 134. Reducción múltiple de la columna central bajo las columnas laterales.

Finalmente se pasarán los círculos de la *columna izquierda sobre la derecha* mediante más movimientos de reducción múltiple.

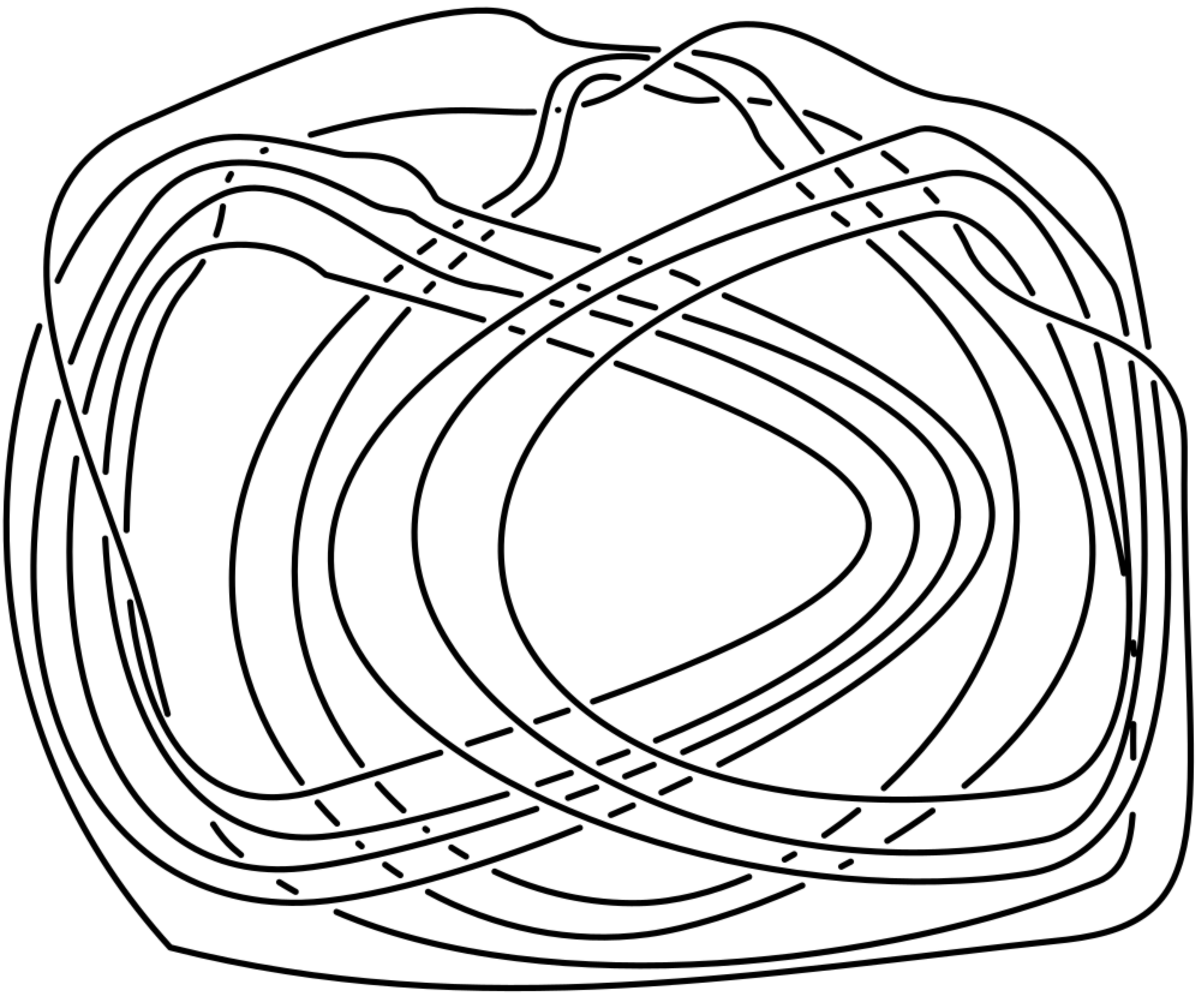


Ilustración 135. Reducción múltiple de los círculos columna izquierda bajo los círculos de la columna derecha.

Aunque no lo llegue a parecer ya hemos llegado a la representación de la trenza, aunque sea difícil de apreciar.

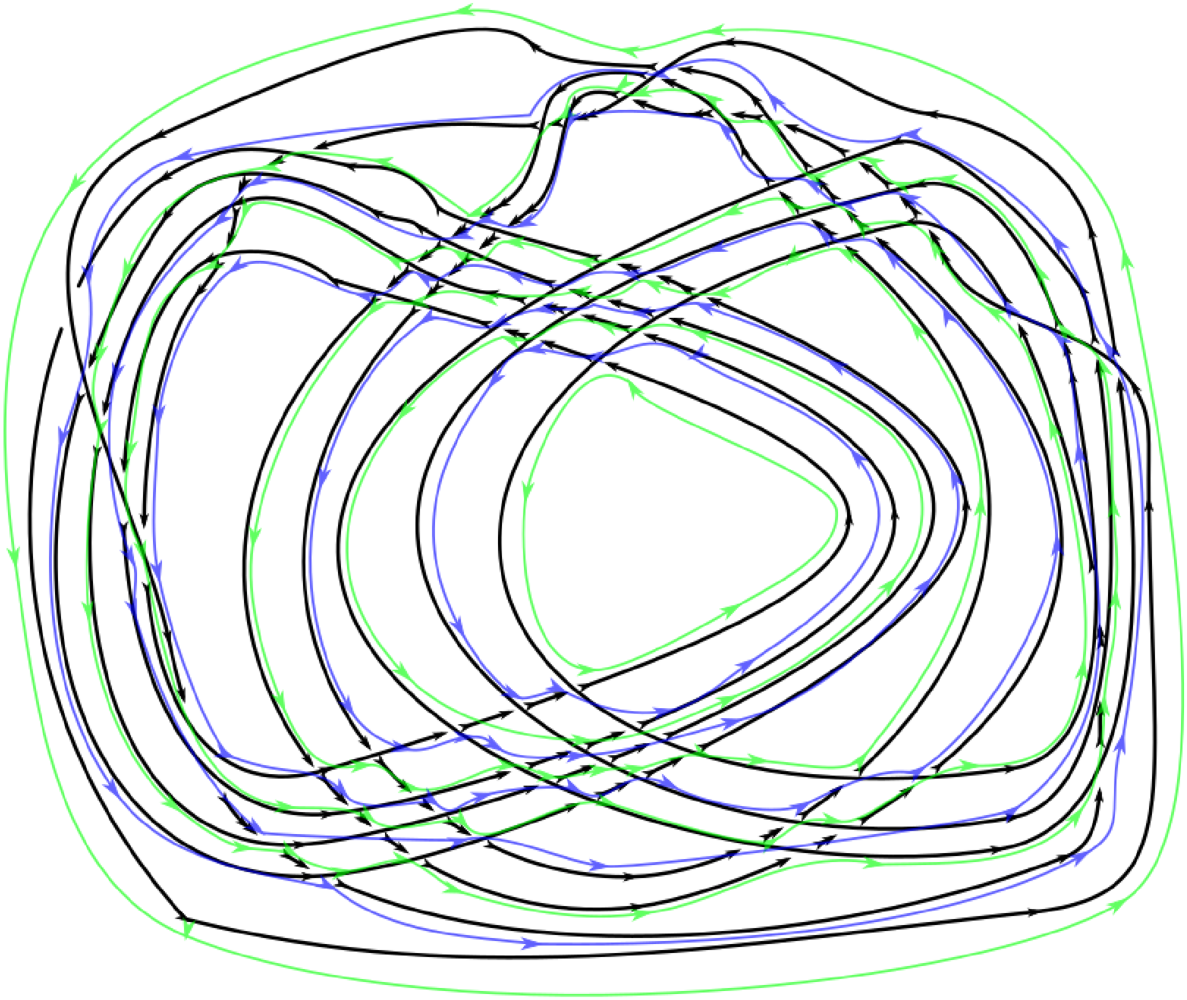


Ilustración 136. P(9,5,7) con sus círculos de Seifert.

Ahora sí que podemos estar totalmente convencidos de que esta es la representación correcta. No sólo todos los círculos son concéntricos, sino que son compatibles entre sí.

Antes de pasar a la descomposición en cruces de la trenza me gustaría aclarar una cosa. Si nos fijamos, en los cruces de grupos de cuerdas se generan muchos cruces, en concreto el producto de las cuerdas que se cruzan, generando una alta densidad de cruces.

Este detalle se va a ejemplificar en el cruce inferior de los círculos de la columna derecha bajo los círculos de la columna izquierda. En concreto, como la izquierda tenía 4 círculos y la derecha 3, tenemos 4x3=12 cruces.

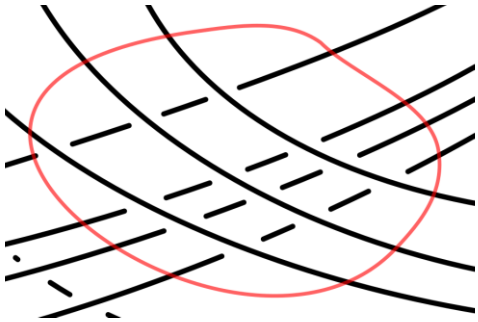


Ilustración 137. Cruce cuerdas columna derecha sobre cuerdas columna izquierda (agrupación 3).

Teniendo en cuenta que siempre recorremos la trenza en sentido anti horario, estos grupos de cruces se pueden escribir de dos maneras. La primera, que es la que se va a utilizar siempre por convenio en este trabajo, se basa en realizar primero los cruces que ocurren a las cuerdas que están más cerca del interior. Es decir, las cuerdas que antes de producirse estos cruces tienen un índice mayor. En este caso son las de la columna derecha, que empiezan por el interior, pasan sobre las de la izquierda, y acaban en el exterior.

Este grupo de cruces en concreto es la agrupación 3.

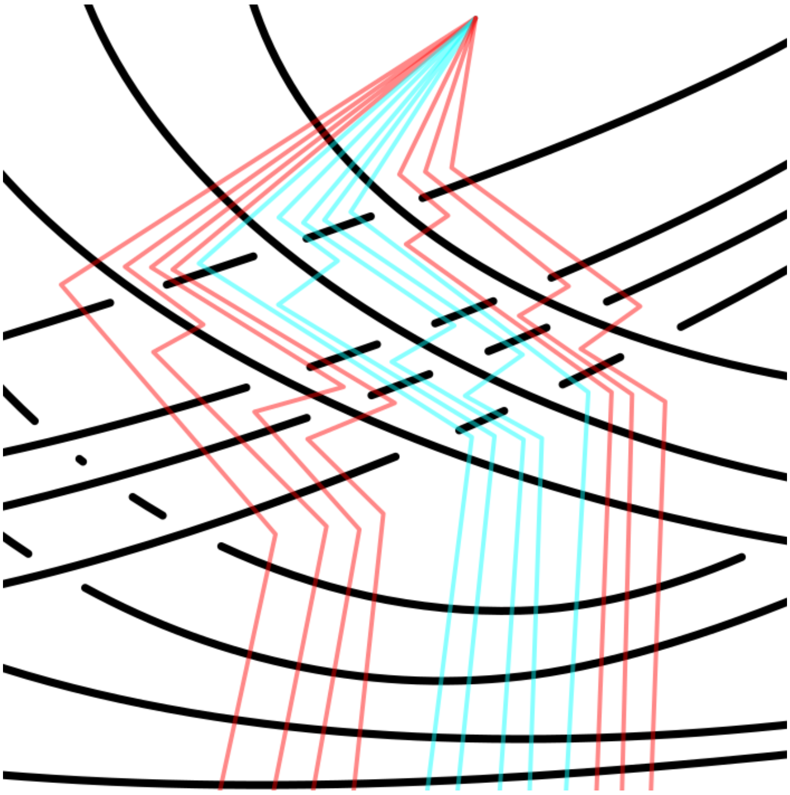


Ilustración 138. Primera forma de resolver los cruces en grupos de cuerdas.

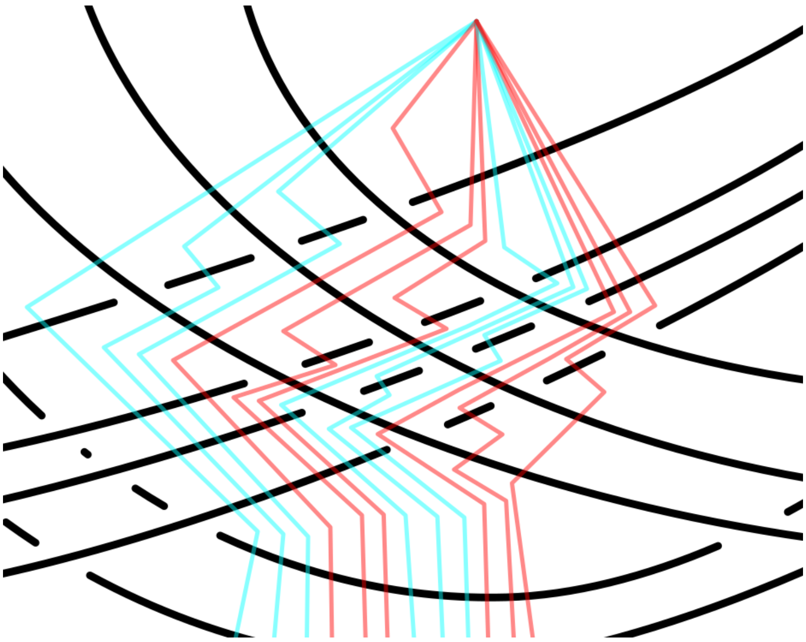


Ilustración 139. Segunda forma de resolver los cruces en grupos de cuerdas.

Como en la primera forma seguimos los cruces de las 3 cuerdas del interior, estas cruzan con las 4 de fuera, de forma que tenemos 3 progresiones de 4 números.

De la segunda forma seguimos los cruces de las 4 cuerdas externas bajo las 3 internas, de forma que tenemos 4 progresiones de 3 números.

Para darle más claridad a este factor de los cruces he dado diferente color a cada grupo, para su mejor visualización. Es de esperar la siguiente pregunta ¿cuál es la diferencia? ¿Es la misma trenza? En la siguiente tabla se han puesto los cruces de un modo y del otro, y posteriormente se demostrará que son los mismos.

Tabla 2. Comparación métodos para clasificar los cruces.

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

Para entender que son la misma trenza, tenemos que recordar la relación que vimos en el capítulo IV: Trenzas, que nos dice que:

$$\sigma_i\sigma_j = \sigma_j\sigma_i \ si \ |i-j| > 0.$$

Si cogemos el 9 que está en la posición 5 del primer modo, lo podemos arrastrar hasta la posición 2 perfectamente en el paso 1, ya que |9-5|>1, |9-6|>1 y |9-7|>1. Lo mismo ocurre con el 10 de la posición 9 en el paso 2.

Ya empiezan a parecerse. Continuamos, llevando el 8 de la posición 7 a la posición 5 en el paso 3, luego el 9 de la posición 10 a la posición 6 en el paso 4 y luego el 7 de la posición 9 a la posición 8 en el paso 5. Para el paso 6 pasamos el 8 de la posición 11 a la 9.

Ya podemos apreciar que son la misma trenza.

Tabla 3. Test de equivalencia entre trenzas.

Cruce	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	11º	12º
Modo 1	**8**	**7**	**6**	**5**	**9**	**8**	**7**	**6**	**10**	**9**	**8**	**7**
Paso 1	8	9	7	6	5	8	7	6	10	9	8	7
Paso 2	8	9	10	7	6	5	8	7	6	9	8	7
Paso 3	8	9	10	7	8	6	5	7	6	9	8	7
Paso 4	8	9	10	7	8	9	6	5	7	6	8	7
Paso 5	8	9	10	7	8	9	6	7	5	6	8	7
Paso 6	8	9	10	7	8	9	6	7	8	5	6	7
Modo 2	**8**	**9**	**10**	**7**	**8**	**9**	**6**	**7**	**8**	**5**	**6**	**7**

Recordar que estos cruces que estamos estudiando pertenecen a la agrupación 3 del algoritmo para P(Impar, Impar, Impar). Vamos a ver cómo se resuelven cruces que se encuentran en un espacio tan pequeño.

Ahora vamos a clasificar los cruces de esta agrupación, ya que es algo que en el dibujo general no se puede apreciar. Yo lo hice usando una tabla Excel, ya que el nudo P(9, 5, 7) tiene 82 cruces, pero para estos 12 nos basta la tabla que ya tenemos.

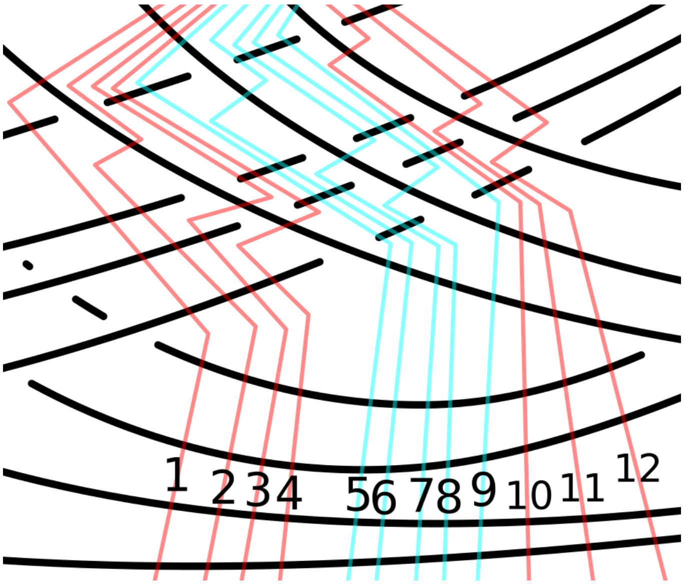


Ilustración 140. Cruces de los círculos de la columna izquierda y derecha de P(9, 5, 7).

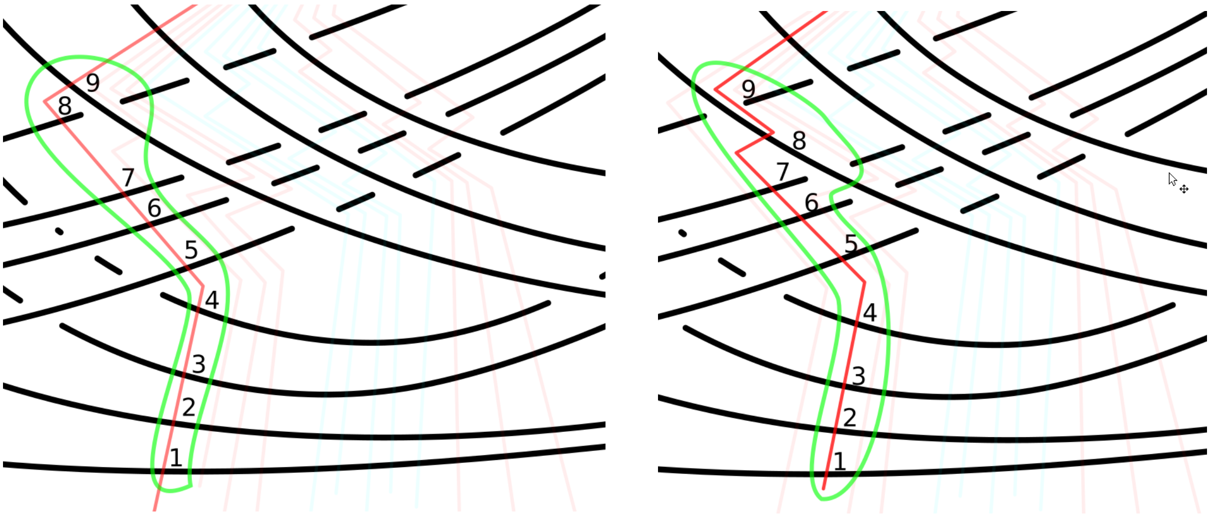


Ilustración 141. Cruces 1 y 2.

Como vemos en la Ilustración 141, denominamos las cuerdas *desde fuera hacia dentro*, y buscamos el siguiente cruce que encontramos entre la línea actual y la siguiente en sentido anti horario. En la ilustración izquierda observamos que la cuerda 8 pasa bajo la cuerda 9, por lo que el cruce vale 8. En la ilustración de la derecha el siguiente cruce consta del paso de la cuerda 7 bajo la que ahora es la cuerda 8 (¡antes era la 9!), por lo que este cruce vale 7. Ya tenemos 8,7.

En los cruces 3, 4 y 5 podemos ver como se cruzan, respectivamente, la cuerda 6 bajo la cuerda 7, la cuerda 5 bajo la cuerda 6 y la cuerda 9 bajo la cuerda 10. Tenemos 8, 7, 6, 5, 9.

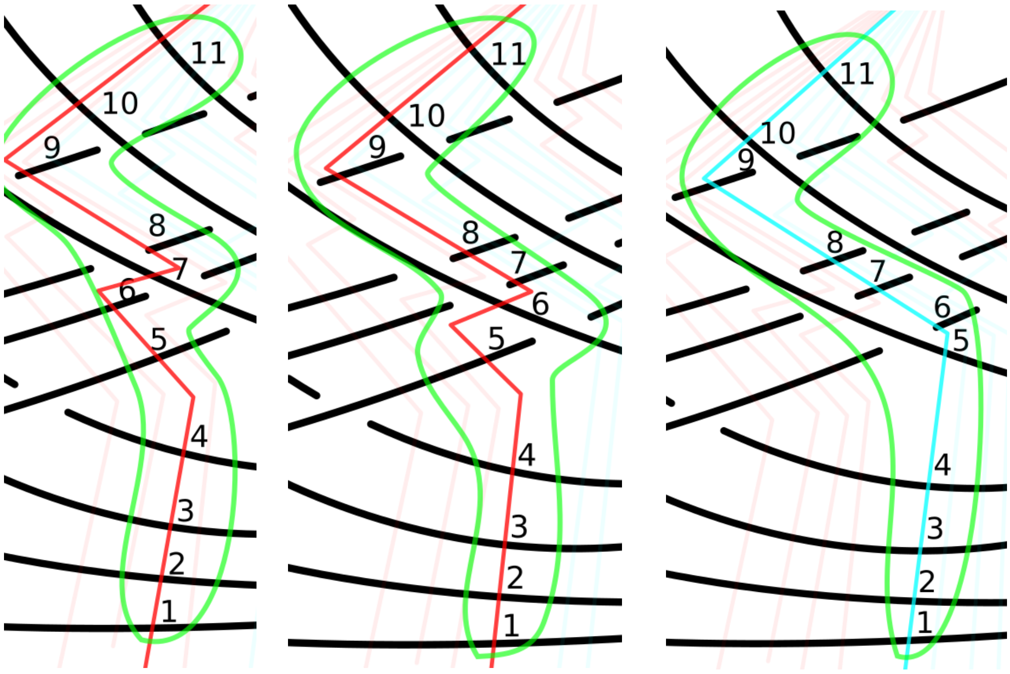


Ilustración 142. Cruces 3, 4 y 5 de la agrupación 3 de P(9,5,7).

En el cruce 6 podemos ver el paso de la cuerda 8 bajo la 9. En el cruce 7 la cuerda 7 bajo la 8. En el cruce 8 la cuerda 6 bajo la 7. En el cruce 9 la cuerda 10 bajo la 11. Tenemos 8,7,6,5,9,8,7,6,10.

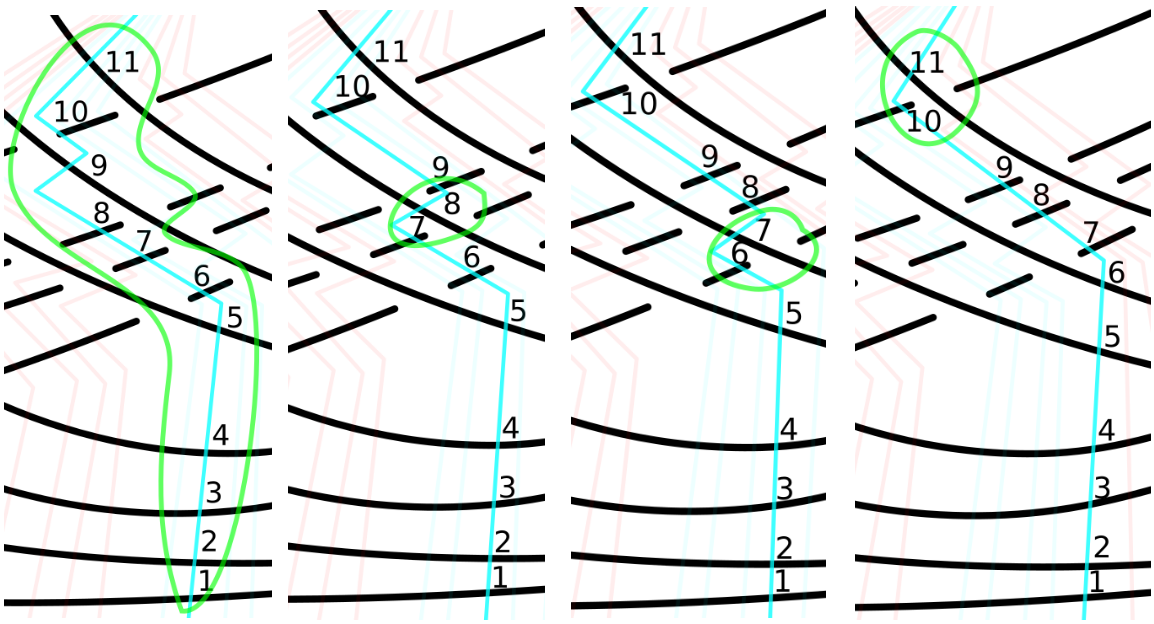


Ilustración 143. Cruces 6, 7, 8 y 9 de la agrupación 3 de P(9,5,7).

Para terminar, nos quedan los cruces 10, 11 y 12. Ya tenemos la agrupación 3:

8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7.

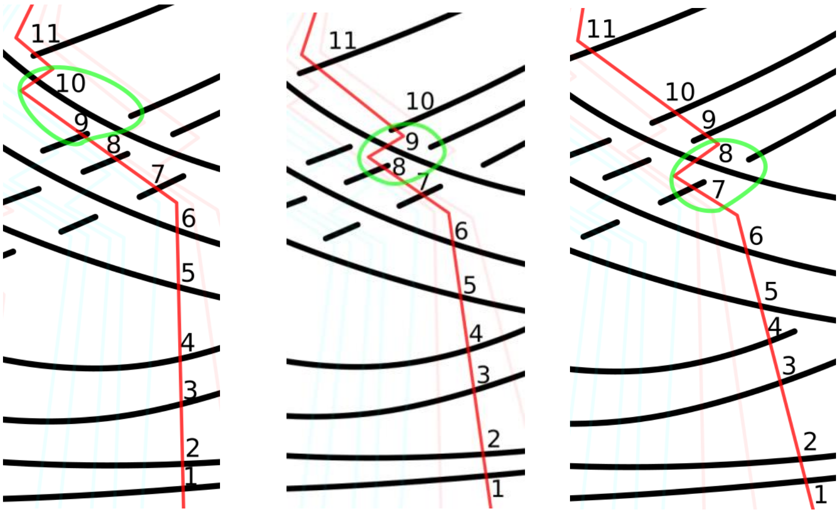


Ilustración 144. Cruces 10, 11 y 12 de la agrupación 3 de P(9, 5, 7).

Ahora llega la hora de la verdad, ya que apenas hemos empezado con la esencia de esta sección.

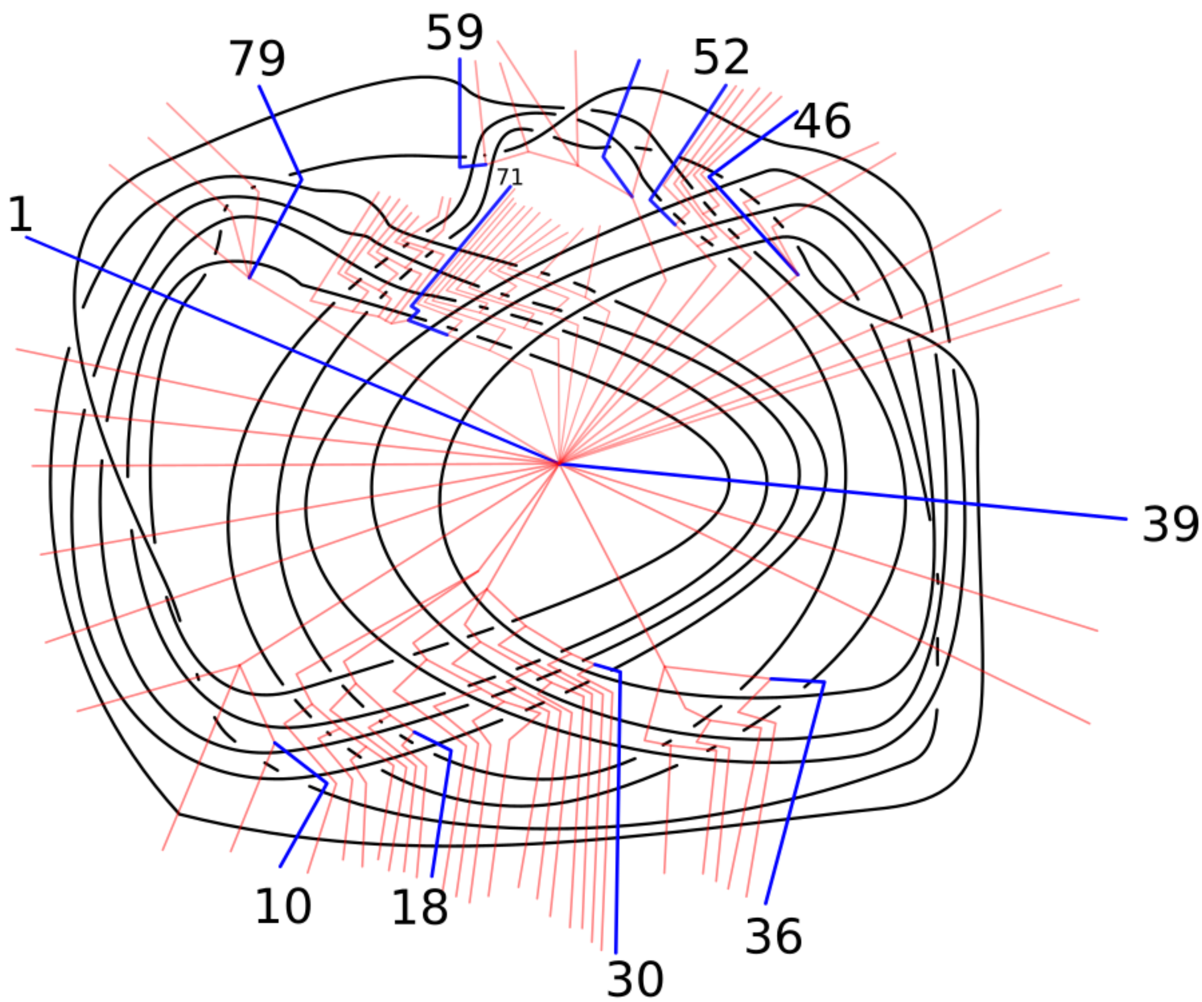


Ilustración 145. Nudo P(9,5,7) dividido en los cruces de su trenza.

Con sus 82 cruces de poco nos puede servir colocar una estrella en su centro como hemos estado haciendo hasta ahora. Incluso hay líneas que no salen hasta el exterior, ya que ensuciarían mucho el dibujo.

Las líneas azules separan las agrupaciones. Hasta ahora cada agrupación sólo dependía de 1 entrada, pero ahora algunas dependerán de 2. Cada comienzo de agrupación se ha marcado con una línea azul y con el número del cruce al que corresponde. Como se puede observar hay 12 agrupaciones. A continuación, se va a estudiar y justificar cada una, el algoritmo para generarlas y las variables de las que dependen. El algoritmo de P(Impar, Impar, Impar) es la concatenación de estos 12 algoritmos.

Agrupación 1

Para estudiar las agrupaciones tenemos que aislarlas y hacer seguimiento de sus movimientos, lo hacemos en la Ilustración 133, Ilustración 134 e Ilustración 135 para ver de dónde han salido y de qué dependen.

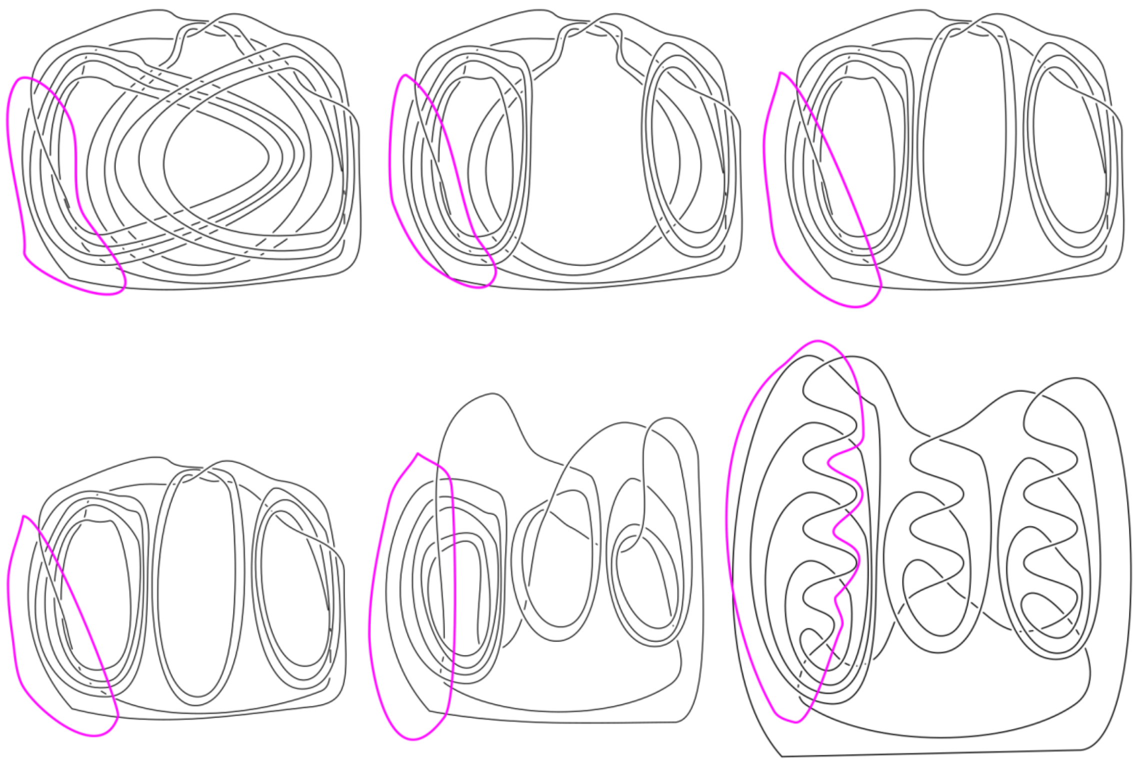


Ilustración 146. Seguimiento de la agrupación 1.

Como podemos ver, siempre empezará por -1 y llegará hasta el núcleo de los círculos que surgen al hacer los movimientos básicos en la columna izquierda, a'. Como son 4 movimientos, sumando la cuerda que cruza al principio, llega hasta -5. El algoritmo dará:

-1, -2, -3, -4, -5, -5, -4, -3, -2

$$\boxed{-1 \downarrow -(a' + 1), -(a' + 1) \uparrow -2}$$

Ecuación 8. Algoritmo para la agrupación 1 de P(impar, impar, impar).

Si la columna izquierda fuese negativa el algoritmo sería el mismo con signo cambiado.

Agrupación 2

Esta surge prácticamente al final del proceso de la Ilustración 134, pero eso no significa que sea más sencilla que la agrupación 1.

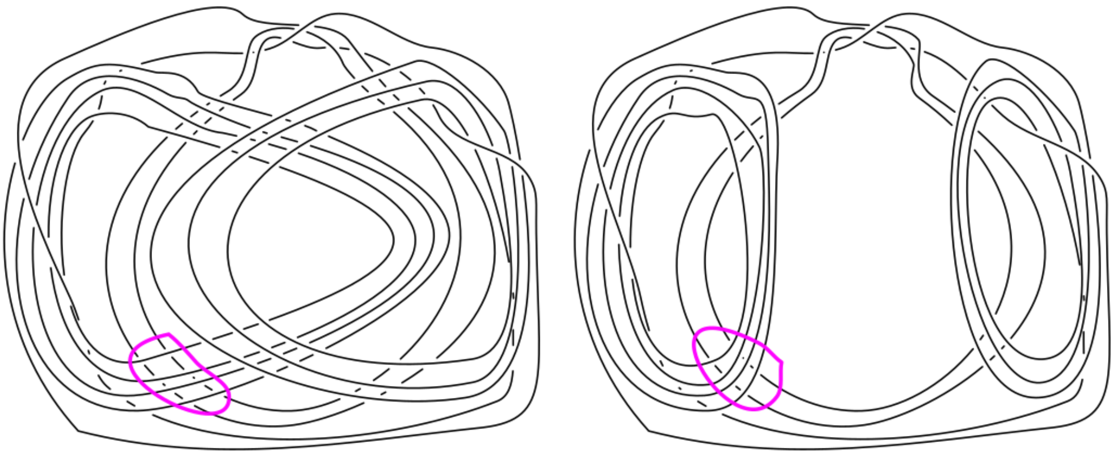


Ilustración 147. Seguimiento de la agrupación 2.

Podemos observar que esta agrupación surge del paso de los círculos de la columna central bajo la columna izquierda, por lo que tendremos 2 variables, el 4 de los 4 movimientos de la columna izquierda y el 2 de los 2 movimientos de la columna derecha. Esta agrupación es un cruce de grupos de cuerdas, que ya hemos visto cómo resolver en la agrupación 3. Ahora tenemos que estudiar el caso general para obtener el algoritmo.

Esta agrupación se puede dividir en 2 componentes:

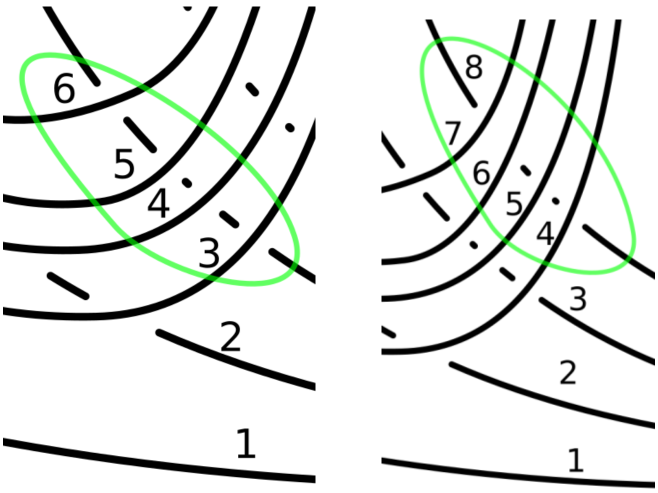


Ilustración 148. Componentes 1 y 2 de la agrupación 2.

Como podemos observar en la primera componente, la cuerda 1 y la 2 siempre estarán ahí. El primer cruce vale (-a’-2) =-6, y el último (-3). Cuanto mayor sea el valor de b, mayor será el número de cruces.

En la segunda componente tenemos que tener en cuenta que no sólo tenemos por debajo la cuerda 1 y 2, si no que ahora también la 3. Por ello, el primer cruce de esta componente vale (-a'-3) =-8 y el último (-4).

Serán necesarias tantas componentes como b', ya que b' (los círculos de la columna central) son los que aparecen por el interior. Recordemos lo dicho del número de cruces, estamos cruzando a' (4) cuerdas sobre b' (2) cuerdas. Como b' (2) son las cuerdas que van por el interior y a' (4) las que van por el exterior, tendremos 2 componentes de 4 unidades cada una.

Recapitulando, *tantas componentes como cuerdas entren por el interior* en el cruce. Expresando el algoritmo tendríamos:

General	Comienzo: (-2-a'-X).	Final: (-3-X), y X irá de 0 hasta -b'-1.	
Componente 1→X=0.	Comienzo: (-2-4-0) =-6.	Final: (-3-0) =-3.	-6,-5,-4,-3.
Componente 2→X=1.	Comienzo: (-2-4-1) =-7.	Final: (-3-1) =-4.	-7,-6,-5,-4.

$$\boxed{\{(-2-a') \uparrow (-3)^{\downarrow b' \downarrow}\}}$$

Ecuación 9. Algoritmo para la agrupación 2 de P(Impar, Impar, Impar).

El signo de b y de a no influye, ya que tal y como hemos dicho los círculos de la columna central siempre cruzarán por debajo de los de las columnas izquierda y derecha.

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

Agrupación 3

Esta agrupación surge, junto con la Agrupación 10, en el último paso.

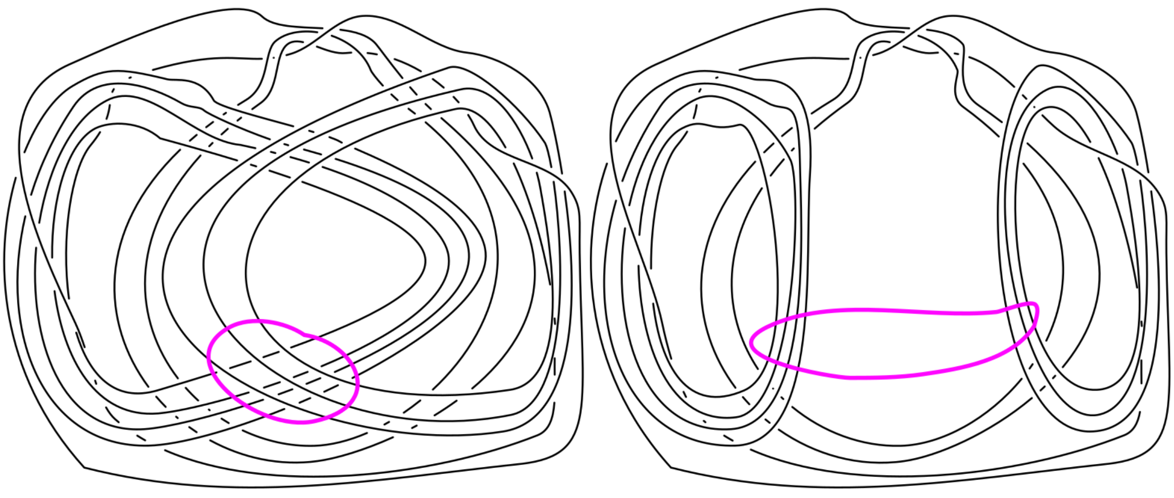


Ilustración 149. Seguimiento de la agrupación 3.

Si la agrupación 2 tenía 2 componentes porque tenía un grupo de 2 cuerdas desde el interior (b'), esta tiene 3 componentes ya que tiene 3 cuerdas desde el interior (c'). Estas 3 cuerdas se cruzan con 4 (a'), por lo que cada componente tendrá 4 unidades.

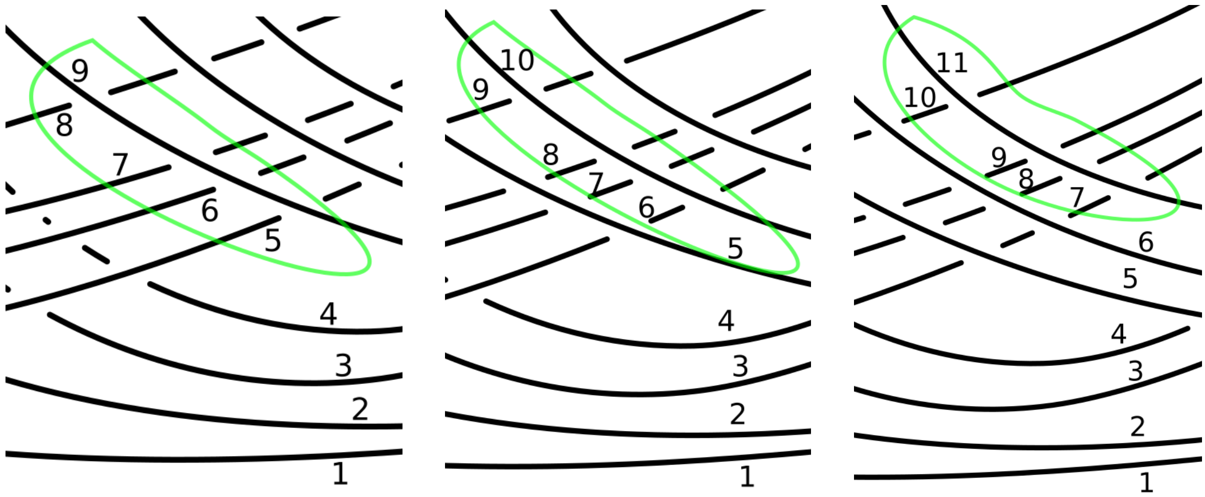


Ilustración 150. Componentes 1, 2 y 3 de la agrupación 3.

Las cuerdas 1 y 2 se siguen manteniendo ahí, pero nos han surgido las cuerdas 3 y 4. Estas dos cuerdas surgen de los círculos generados en la columna central, es decir, b' (2 en este caso).

En el componente 1 el primer cruce se produce cuando contamos las 2 cuerdas exteriores, las 2 de b' y las 4 de c'. Es decir, vale (2+b'+a') =8. Va a ir bajando hasta valer (2+b'+1) = 5.

Para el segundo componente tenemos abajo las 2 cuerdas exteriores, las 2 de b', una de c' que ya ha pasado y se queda abajo (la cuerda 5 en la segunda ilustración) y las 4 de a'. Es decir, vale (2+b'+1+a') =9 y va bajando hasta (2+b'+1) =6. Este es el algoritmo:

General:	Comienzo: (2+b'+X+a')	Final: (2+b'+1+X)	
Componente 1:	Comienzo: (2+2+0+4) =8	Final: (2+2+1+X) =5	(8,7,6,5)
Componente 2:	Comienzo: (2+2+1+4) =9	Final: (2+2+1+1) =6	(9,8,7,6)
Componente 3:	Comienzo: (2+2+2+4) =10	Final: (2+2+1+2) =7	(10,9,8,7)

Como ya sabemos, tendremos tantas componentes como C' (en este caso 3).

$$\{(2 + b' + a') \downarrow (3 + b')^{\uparrow c' \uparrow}\}$$

Ecuación 10. Algoritmo para la agrupación 3 de P(Impar, Impar, Impar).

El signo de las columnas no influye, ya que siempre pasamos los círculos de la columna izquierda sobre la columna derecha.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7

Agrupación 4

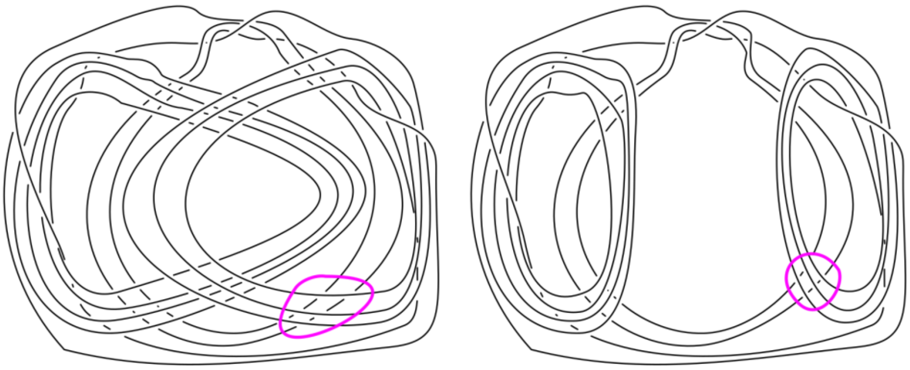


Ilustración 151. Seguimiento de la agrupación 4.

Esta agrupación es análoga a la Agrupación 2, pero en vez de surgir del cruce de los círculos de la izquierda sobre los círculos del centro, surge de los círculos de la izquierda sobre los círculos del centro. Es decir, en vez de cruzar a’(4) sobre b’(2), pasando c’ por el interior y dando 2 componentes, cruza c’(3) sobre b’(2), pasando c’ por el interior y dando 3 componentes (de 2 unidades cada una).

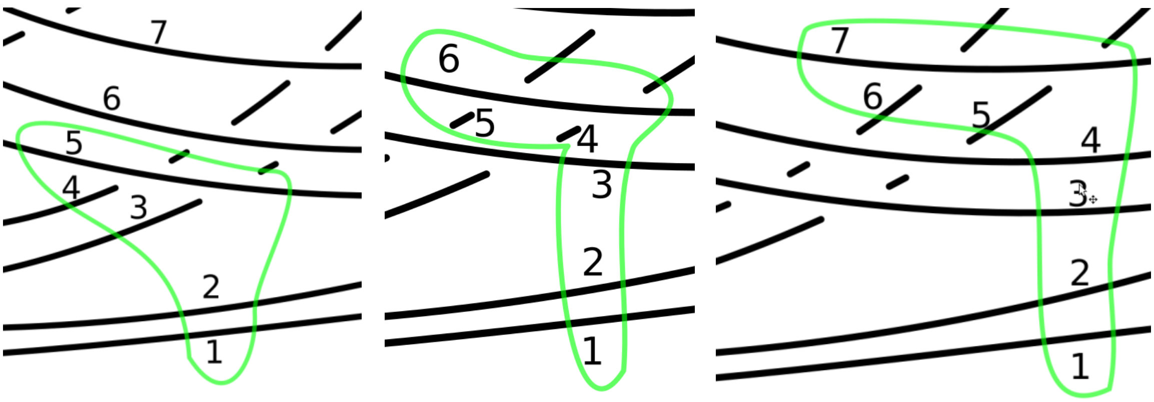


Ilustración 152. Componentes 1, 2 y 3 de la agrupación 4.

Sabiendo resolver la Agrupación 2 esta resulta trivial. Tenemos las mismas 2 cuerdas exteriores que se mantienen, sólo tenemos que cambiar las variables ya dichas.

General:	Comienzo: (2+b’+X)	Final: (3+X)	
Componente 1	Comienzo: (2+2+0) =4	Final: (3+0) =3.	4,3
Componente 2	Comienzo: 2+2+1) =5	Final: (3+1) =4.	5,4
Componente 3	Comienzo: (2+2+2) =6	Final: (3+2) =5	6,5

$$\{(2+b') \downarrow (3)^{\uparrow c' \uparrow}\}$$

Ecuación 11. Algoritmo para la agrupación 4 de P(Impar, Impar, Impar).

Los signos de las columnas no influyen, ya que siempre pasamos los círculos de la columna central bajo los de la columna derecha.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5

Agrupación 5

Esta no se parece a ninguna que hayamos hecho hasta ahora, así que habrá que tener cuidado.

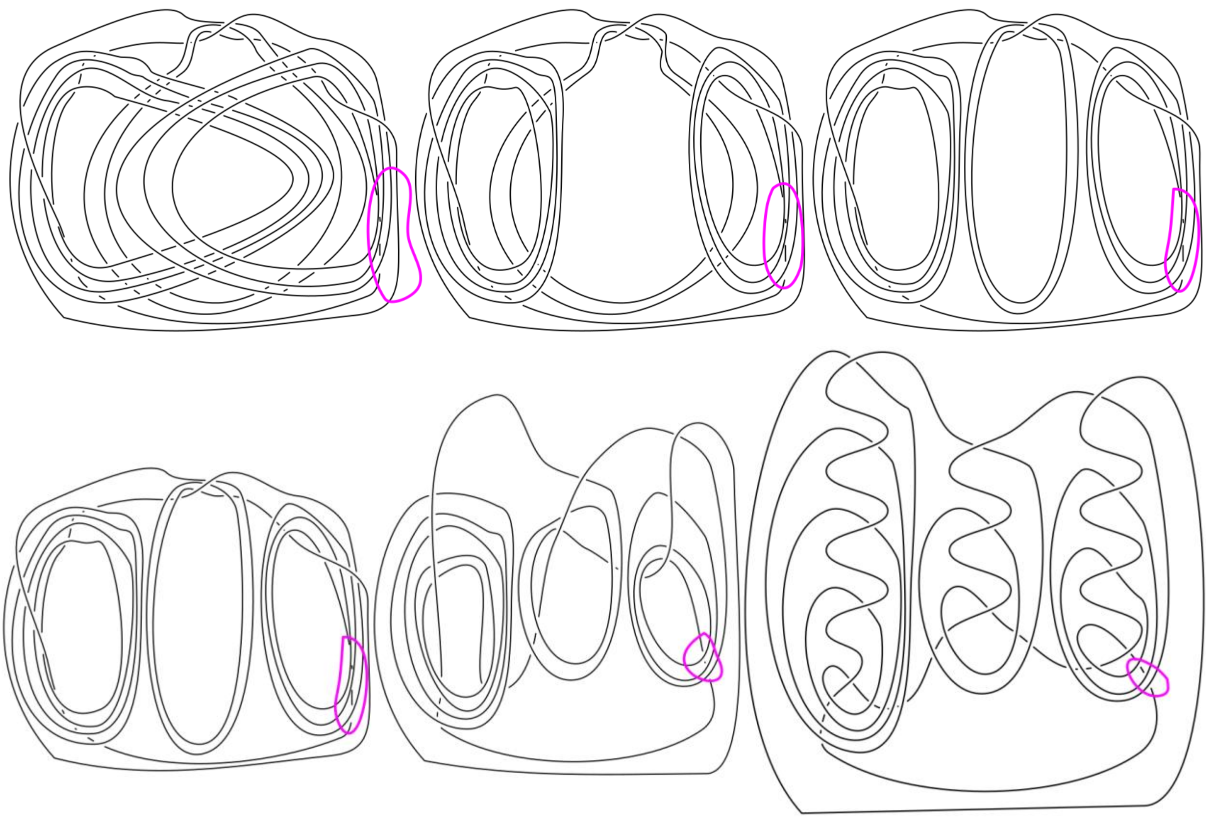


Ilustración 153. Seguimiento de la agrupación 5.

Como podemos observar la agrupación 5 surge al realizar los movimientos básicos en la columna derecha. Siempre comenzará en 2 y acabará en 1+c'.

General:	Comienzo: 2	Fin: 1+c'	
P(9,5,7):	Comienzo: 2	Fin: 1+3=4	2,3,4

$$\boxed{2 \uparrow (1 + c')}$$

Ecuación 12. Algoritmo para la agrupación 5 de P(Impar, Impar, Impar).

Si c es negativo se invierte el signo de la agrupación.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4

Agrupación 6

Es el equivalente a la Agrupación 1, pero esta vez para la columna derecha.

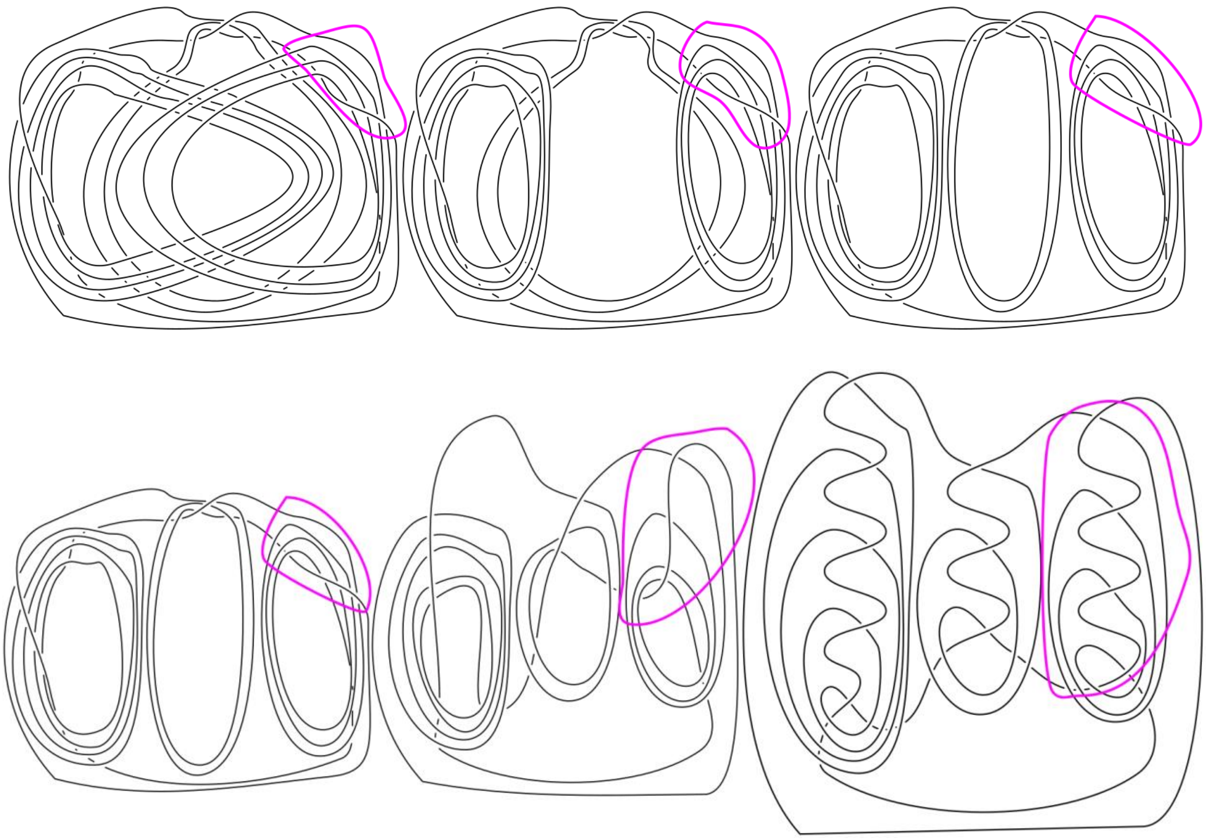


Ilustración 154. Seguimiento agrupación 6.

Es decir, es exactamente el mismo algoritmo que en Agrupación 1, pero cambiando a' por c'.

-1,-2,-3,-4,-4,-3,-2

$$\boxed{-1 \downarrow -(c' + 1), -(c' + 1) \uparrow -2}$$

Ecuación 13. Algoritmo para la agrupación 6 de P(Impar, Impar, Impar).

Si c' es negativo se invierte el signo de toda la serie.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2

Agrupación 7

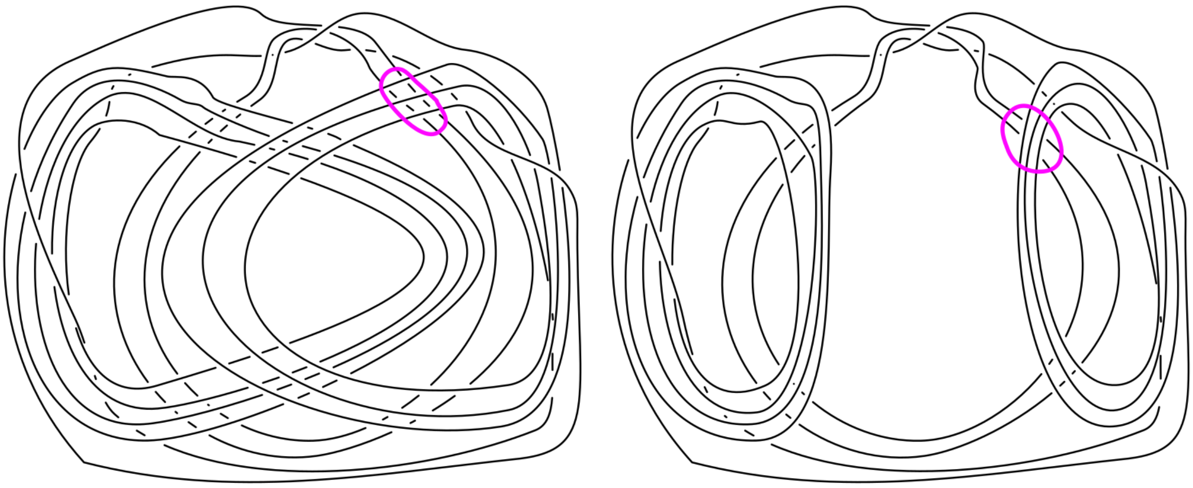


Ilustración 155. Seguimiento agrupación 7.

Es el equivalente superior a la Agrupación 4. Es decir, si en la 4 pasaba b’ bajo c’, pasando por el interior c’, esta vez vuelve a pasar b’ bajo c’, pero por el interior tenemos a c’. Como en la agrupación 4 c’ (3) pasaba por debajo, teníamos 3 componentes, como ahora pasa b’ (2) por debajo, tenemos 2 componentes.

General:	Comienzo: (-2-c’-X)	Final: (-3-X)	
Componente 1	Comienzo: (-2-3-0) =-5	Final: (-3-0) =-3.	-5,-4,-3
Componente 2	Comienzo: (-2-3-1) =-6	Final: (-3-1) =-4.	-6,-5,-4

$$\{(-2-c') \uparrow (-3)^{\downarrow b'\downarrow}\}$$

Ecuación 14. Algoritmo para la agrupación 7 de P(Impar, Impar, Impar).

El signo no influye en esta agrupación.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4

Agrupación 8

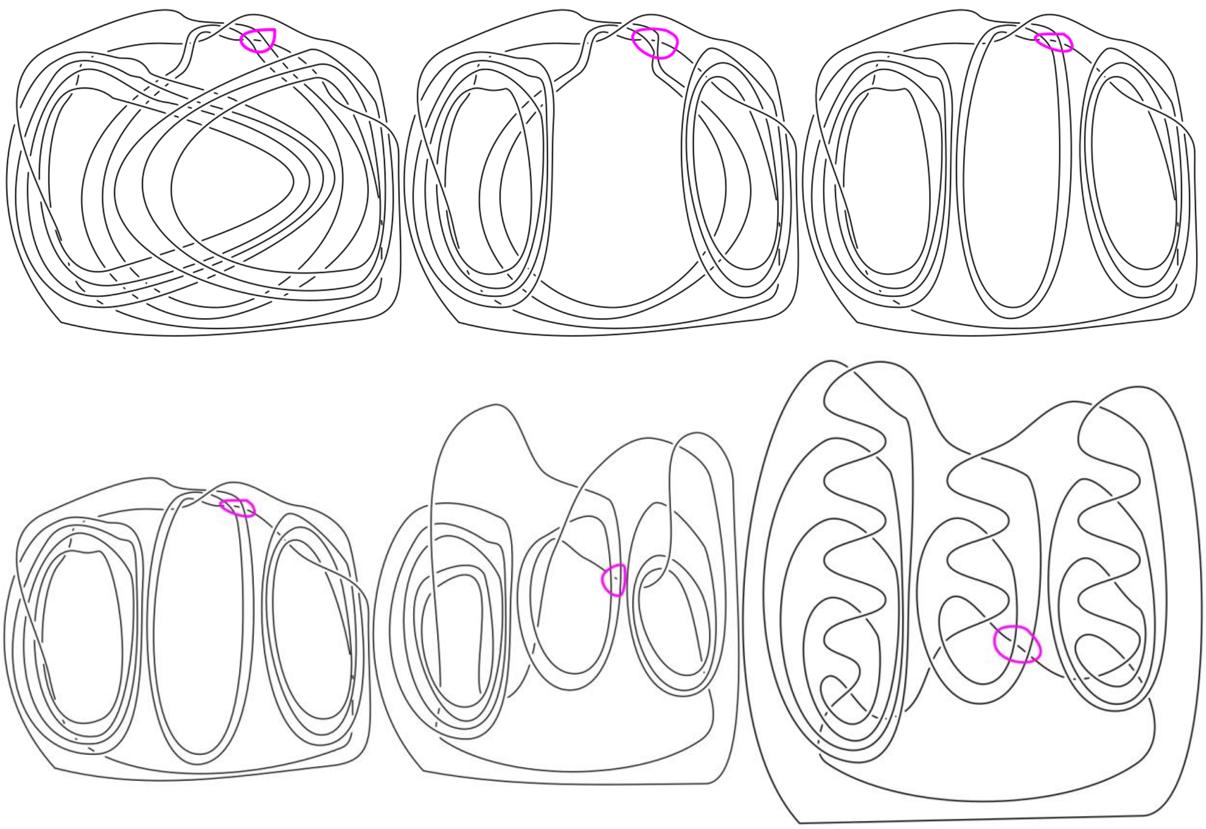


Ilustración 156. Seguimiento agrupación 8.

Aquí tenemos un ejemplo de manual de porqué hay que usar números grandes para estudiar los nudos pretzel. Un b = 5 en la columna central parece un número suficientemente grande, pero b’ vale 2, lo que implica que podemos tener 2 cruces que, aunque sean simples, son componentes de por sí, y no se pueden obviar. Si observamos con suficiente detenimiento, nos daremos cuenta de que esta agrupación es equivalente a la Agrupación 5, pero en esta en vez de entrar en los círculos de la columna derecha, entra en los círculos de la columna central.

Por eso la Agrupación 5 tenía 3 unidades, porque dependía de c’. Esta agrupación tiene 2 unidades porque depende de b’.

General:	Comienzo: 2	Fin: 1+b’	
P (9,5,7):	Comienzo: 2	Fin: 1+2=3	2,3

$$\boxed{2 \uparrow (1 + b')}$$

Ecuación 15. Algoritmo para la agrupación 8 de P(Impar, Impar, Impar).

Si b es negativo se invierte el signo de la agrupación.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4, 2, 3

Agrupación 9

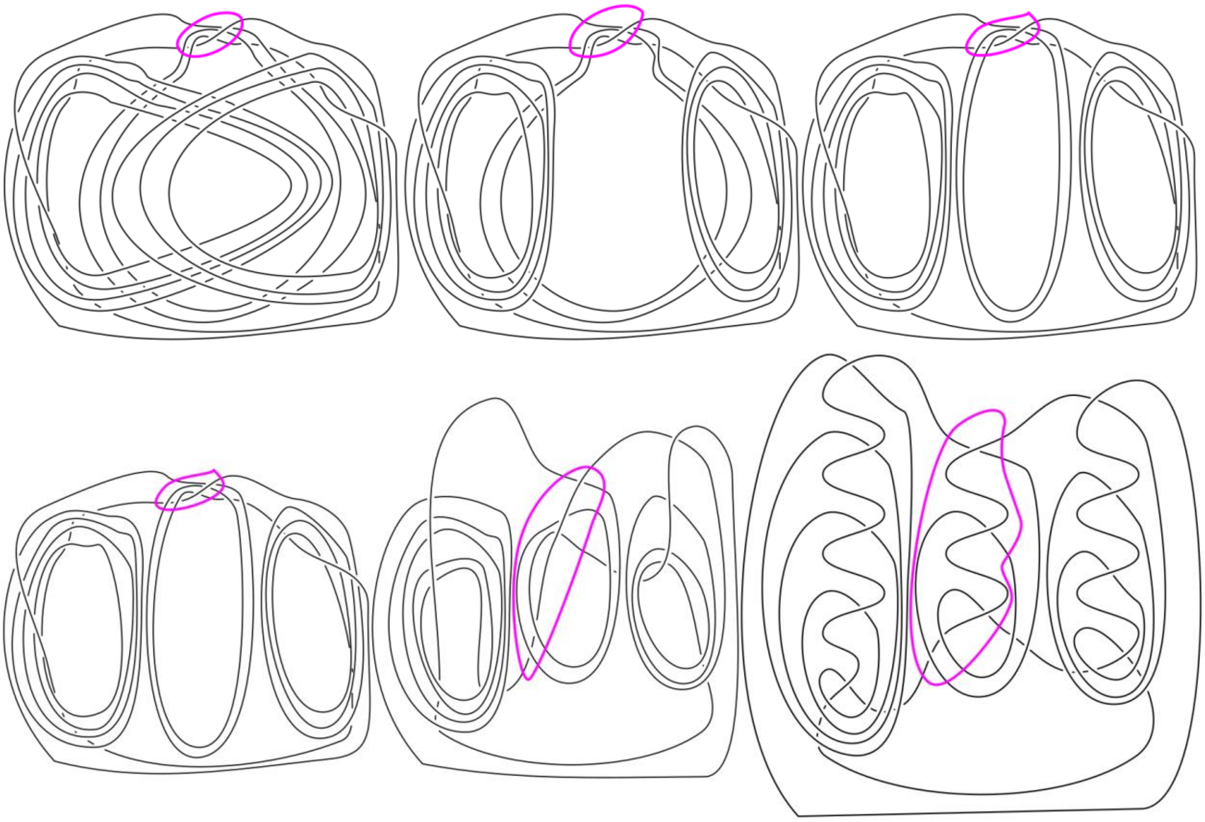


Ilustración 157. Seguimiento de la agrupación 9.

Ya podemos ver con facilidad que esta agrupación es equivalente a la Agrupación 1 y a la Agrupación 6, pero en vez de depender de a’ y c’ respectivamente, depende de b’, es decir, la columna central.

-1, -2, -3, -3, -2

$$\boxed{-1 \downarrow\ -(b' + 1), -(b' + 1) \uparrow -2}$$

Ecuación 16. Algoritmo para la agrupación 9 de P(Impar, Impar, Impar).

Si la columna central fuese negativa se invertiría el signo de la serie.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4, 2, 3, -1, -2, -3, -3, -2

Agrupación 10

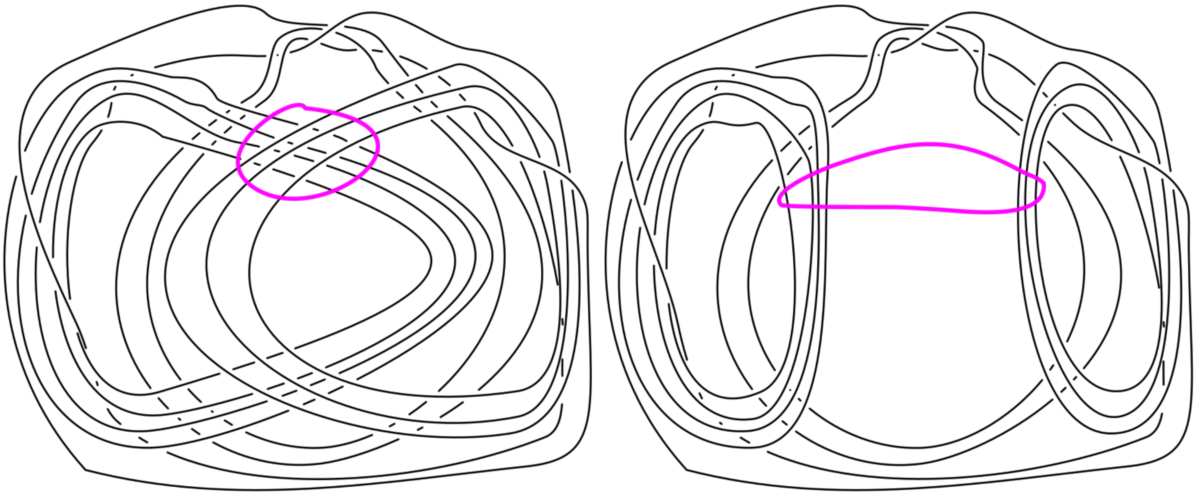


Ilustración 158. Seguimiento de la agrupación 10.

Como podemos ver, este es claramente el equivalente de la Agrupación 3 pero arriba. Aunque esta vez, como en vez de tener c' (3) por el interior tenemos a' (4) por el interior, tendremos 4 componentes en vez de 3. Es decir, en vez de 3 componentes de 4 unidades, tenemos 4 unidades de 3 componentes. El número de cruces es el mismo ya que 3x4=4x3=12.

General:	Comienzo: (-2-b'-X-c')	Final: (-2-b'-1-X)	
Componente 1:	Comienzo: (-2-2-0-3) =-7	Final: (-2-2-1-0) =-5	-7,-6,-5
Componente 2:	Comienzo: (-2-2-1-3) =-8	Final: (-2-2-1-1) =-6	-8,-7,-6
Componente 3:	Comienzo: (-2-2-2-3) =-9	Final: (-2-2-1-2) =-7	-9,-8,-7
Componente 4:	Comienzo: (-2-2-3-3) =-10	Final: (-2-2-1-3) =-8	-10,-9,-8

$$\{(-2-b'-c') \downarrow (-3-b')^{\uparrow a' \uparrow}\}$$

Ecuación 17. Algoritmo de la agrupación 10 de P(Impar, Impar, Impar).

De nuevo, el signo de las columnas no influye, ya que siempre que hagamos esta reducción pasaremos el grupo de cuerdas de la derecha sobre el de la izquierda.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4, 2, 3, -1, -2, -3, -3, -2, -7, -6, -5, -8, -7, -6, -9, -8, -7, -10, -9, -8

Agrupación 11

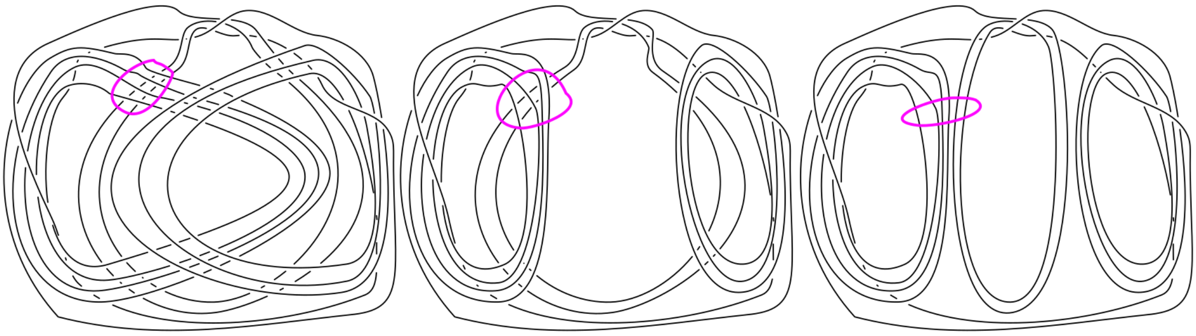


Ilustración 159. Seguimiento agrupación 11.

Ya tenemos experiencia para ver que este es el equivalente a la Agrupación 2, la Agrupación 4 y la Agrupación 7. Esta vez nuestras variables son a’ y b’, las mismas que en la Agrupación 2. Pero ahora tenemos por el interior a a’ (4), por lo que tendremos 4 componentes, mientras que antes teníamos por dentro a b’ (2) y teníamos 2 componentes.

General	Comienzo: (2+b’+X).	Final: (3+X)	
Componente 1	Comienzo: (2+2+0) =4.	Final: (3+0) =3.	4,3
Componente 2	Comienzo: (2+2+1) =5.	Final: (3+1) =4.	5,4
Componente 3	Comienzo: (2+2+2) =6	Final: (3+2) =5	6,5
Componente 4	Comienzo: (2+2+3) =7	Final: (3+3) =6	7,6

$$\{(2 + b') \downarrow (3)^{\uparrow a'\uparrow}\}$$

Ecuación 18. Algoritmo de la agrupación 11 de P(Impar, Impar, Impar).

Los signos de las columnas no influyen.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4, 2, 3, -1, -2, -3, -3, -2, -7, -6, -5, -8, -7, -6, -9, -8, -7, -10, -9, -8, 4, 3, 5, 4, 6, 5, 7, 6

Agrupación 12

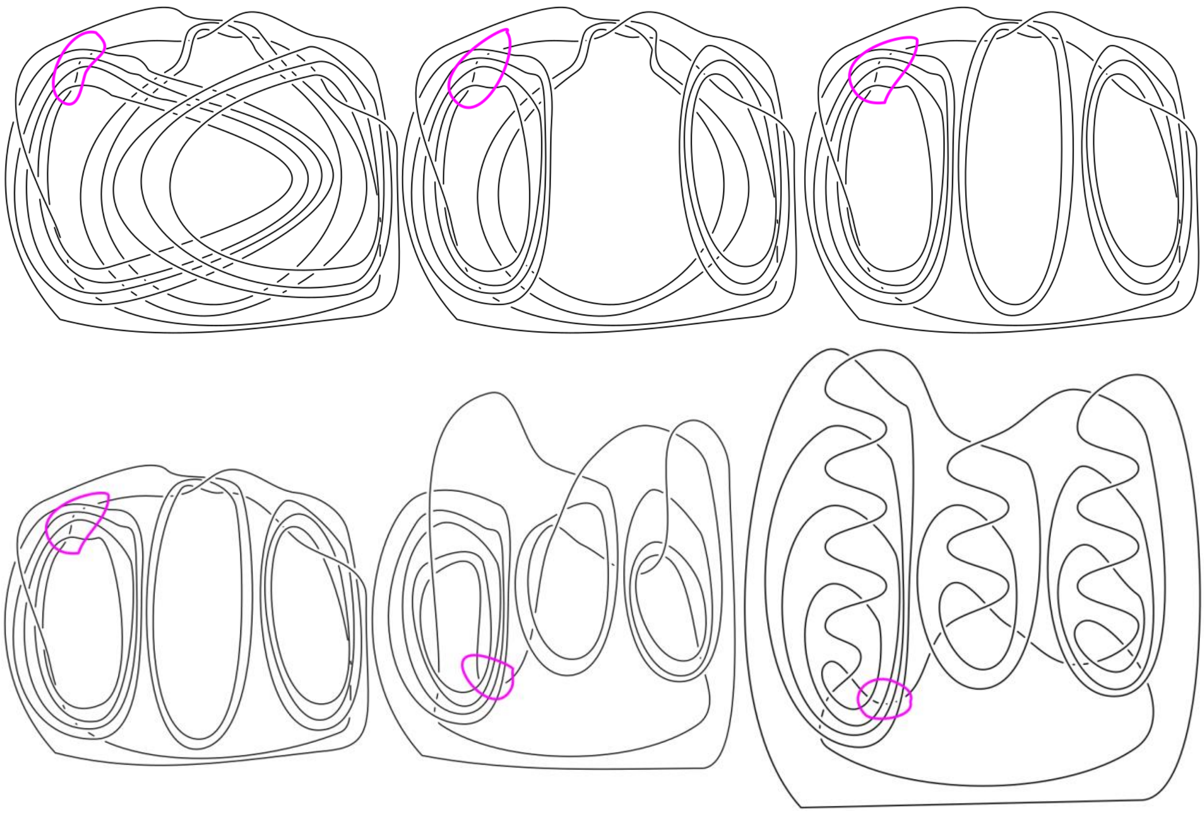


Ilustración 160. Seguimiento agrupación 12.

Finalmente nos encontramos lo que podemos apreciar que es el equivalente a las agrupaciones Agrupación 5 y Agrupación 8.

La diferencia esta vez es que dependemos de los círculos de la primera columna, es decir, a'.

General:	Comienzo: 2	Fin: 1+a'	
P(9,5,7):	Comienzo: 2	Fin: 1+4=5	2,3,4,5

$$2 \uparrow (1 + a')$$

Ecuación 19. Algoritmo de la agrupación 12 de P(impar, Impar, Impar).

Si el signo de esta primera columna es negativo toda la serie saldrá con signo negativo.

-1, -2, -3, -4, -5, -5, -4, -3, -2, -6, -5, -4, -3, -7, -6, -5, -4, 8, 7, 6, 5, 9, 8, 7, 6, 10, 9, 8, 7, 4, 3, 5, 4, 6, 5, 2, 3, 4, -1, -2, -3, -4, -4, -3, -2, -5, -4, -3, -6, -5, -4, 2, 3, -1, -2, -3, -3, -2, -7, -6, -5, -8, -7, -6, -9, -8, -7, -10, -9, -8, 4, 3, 5, 4, 6, 5, 7, 6, 2, 3, 4, 5

Agrupaciones equivalentes

Hemos visto que hay varias agrupaciones que son equivalentes a otras, y sólo se diferencian en las variables. Vamos a recapitular cuáles y de dónde surgen.

Para empezar, tenemos la 1, la 6 y la 9. Las 3 empiezan con una cuerda exterior que entra en el centro de los círculos de su respectiva columna pasando por encima, para seguidamente salir pasando por debajo de los círculos de su columna, pero dejando esta vez otra cuerda en el exterior.

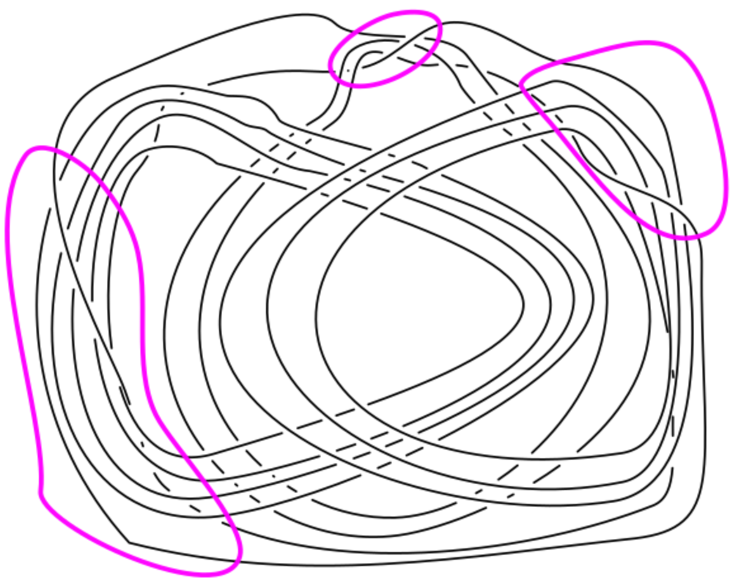


Ilustración 161. Agrupaciones 1, 6 y 9.

Luego tenemos las agrupaciones 2, 4, 7 y 11, que se basan en los cruces entre las cuerdas de los círculos de la columna central bajo las de la columna izquierda y derecha.

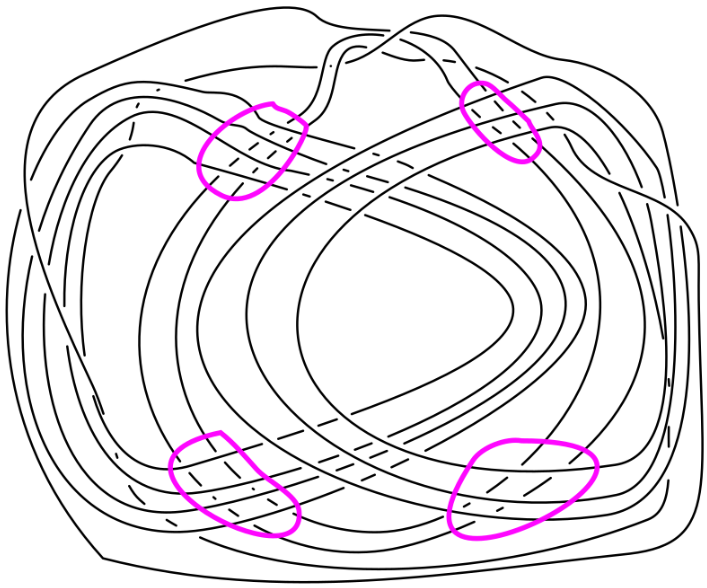


Ilustración 162. Agrupaciones 2, 4, 7 y 11.

La agrupación 3 y 10 surge de los cruces de los círculos de la columna izquierda sobre la columna derecha:

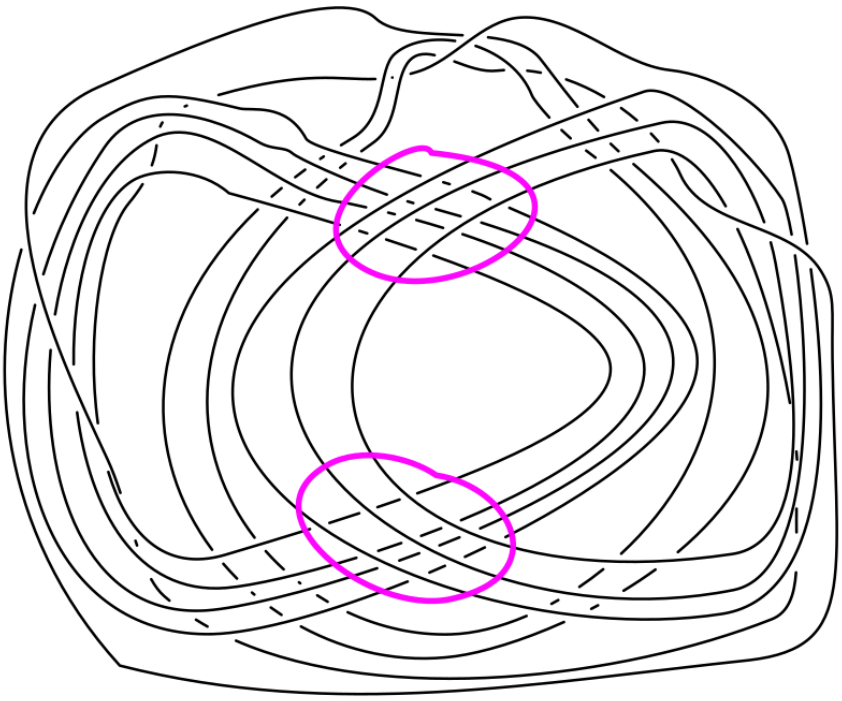


Ilustración 163. Agrupaciones 3 y 10.

Para finalizar tenemos las agrupaciones 5, 8 y 12, que atraviesan los círculos de sus respectivas columnas.

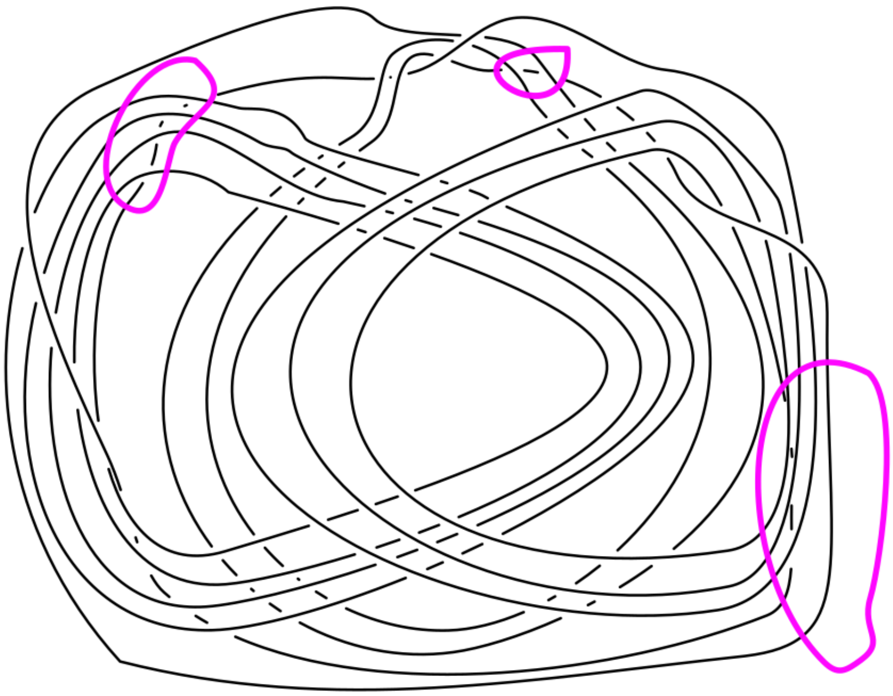


Ilustración 164. Agrupaciones 5, 8 y 12.

Conjunto

Ya tenemos la trenza dividida en sus 12 agrupaciones, junto con sus algoritmos. El algoritmo global es la concatenación de todos estos.

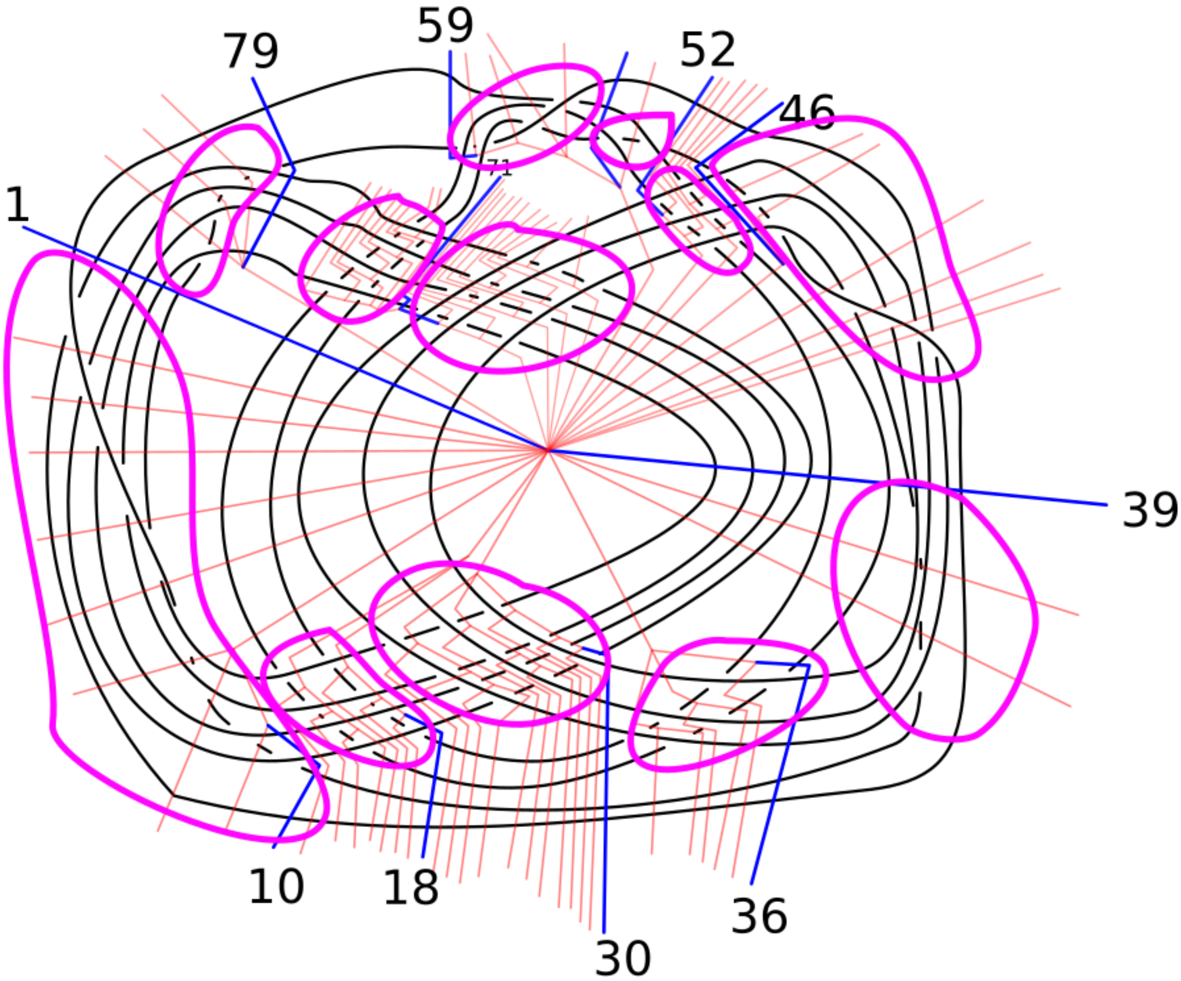


Ilustración 165. Agrupaciones de la trenza de P(9, 5, 7).

Enlaces pretzel con tres entradas

Hasta ahora he alcanzado el objetivo inicial del TFG, consistente en obtener una trenza para los nudos pretzel de 3 columnas.

Llegados a este punto estábamos más que satisfechos, e implementamos los algoritmos descubiertos en un programa Python, tal y como se puede ver en el capítulo VI. Alguna vez nos descuidábamos y le dábamos dos o tres números pares, cuando sólo puede ser uno o ninguno. El programa identificaba que le habíamos dado un enlace y anunciaba un error.

Una vez aquí, llegamos a la siguiente pregunta: ¿será difícil implementar los enlaces? Investiguemos qué ocurre cuando tenemos 2 o 3 columnas con entrada par.

Antes de estudiar los enlaces tenemos que tener en cuenta que la orientación va a afectar a los círculos de Seifert. Hasta ahora hemos establecido que la cuerda superior tenía que estar orientada en sentido antihorario. Pero ahora estamos tratando con enlaces, lo que significa que hay otra componente que orientar, y la orientación de esta segunda componente determina la localización de la columna que tiene los círculos de Seifert.

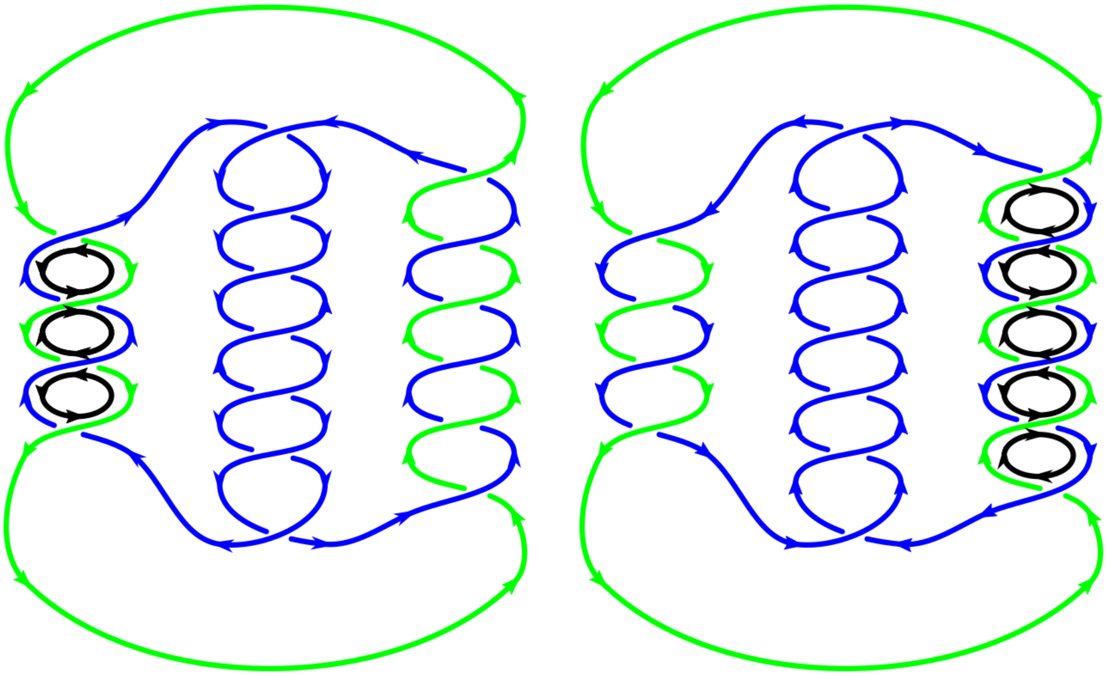


Ilustración 166. Influencia de la orientación en la localización de los círculos de Seifert.

Podemos observar que, independientemente de la orientación, la columna de círculos de Seifert siempre estará en una columna con entrada par.

Se va a mantener el convenio de orientar la cuerda superior en sentido antihorario, y en cada caso se establecerá la orientación de la otra cuerda.

Enlaces P(par, impar, par)

Usando dos colores se observa cómodamente que tenemos entre manos un enlace de dos componentes.

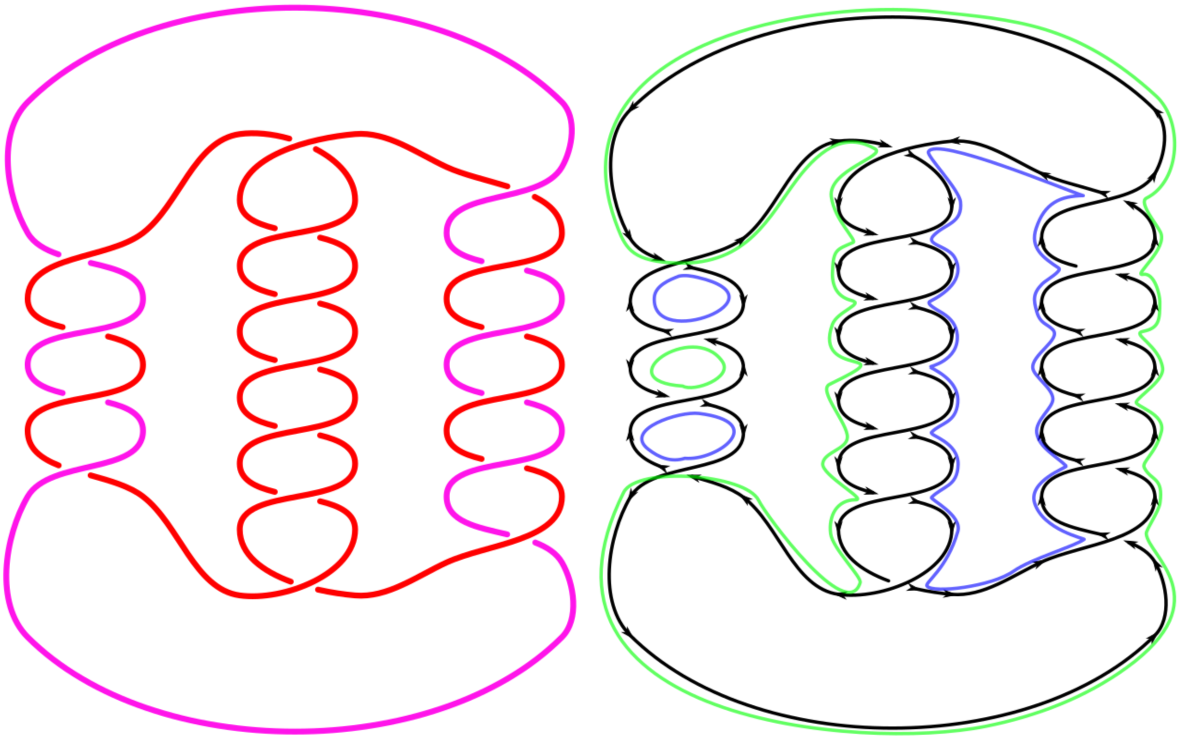


Ilustración 167. P(4, 7, 6) y su orientación con círculos de Seifert.

Orientamos la componente roja de forma que las orientaciones sean opuestas en la columna de entrada par a la izquierda de la columna de entrada impar. Si estudiamos los círculos de Seifert nos damos cuenta de que estamos ante la misma situación que en la sección Nudos P(par, impar, impar), por lo que vamos a probar el mismo método.

Hacemos un movimiento básico en la columna izquierda, luego su movimiento de polo sur y otro movimiento básico y su movimiento de polo sur.

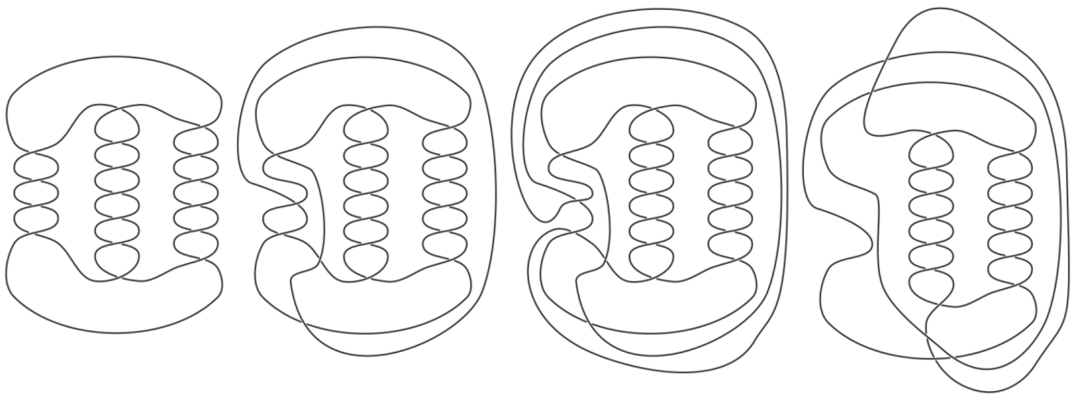


Ilustración 168. Dos movimientos básicos en la columna izquierda y sus dos movimientos de polo sur.

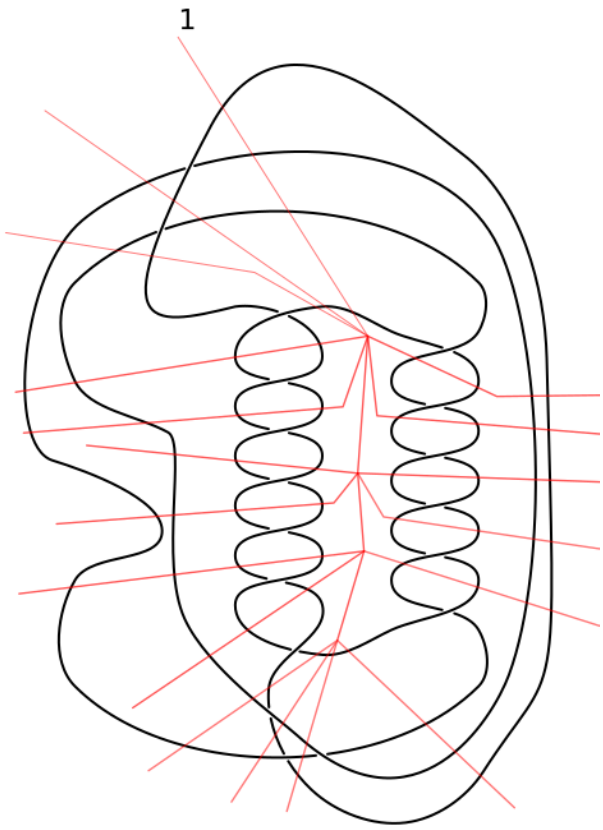


Ilustración 169. P(4, 7, 6) con los cruces de trenza marcados.

Vamos a hacer otro más, el P(8, 3, 2) para comprobar si la equivalencia con el caso Nudos P(par, impar, impar) es cierta. Si lo es, el valor de la columna central y derecha sólo nos dicta cuántas veces se repetirá el número (*movimientos básicos + 1*).

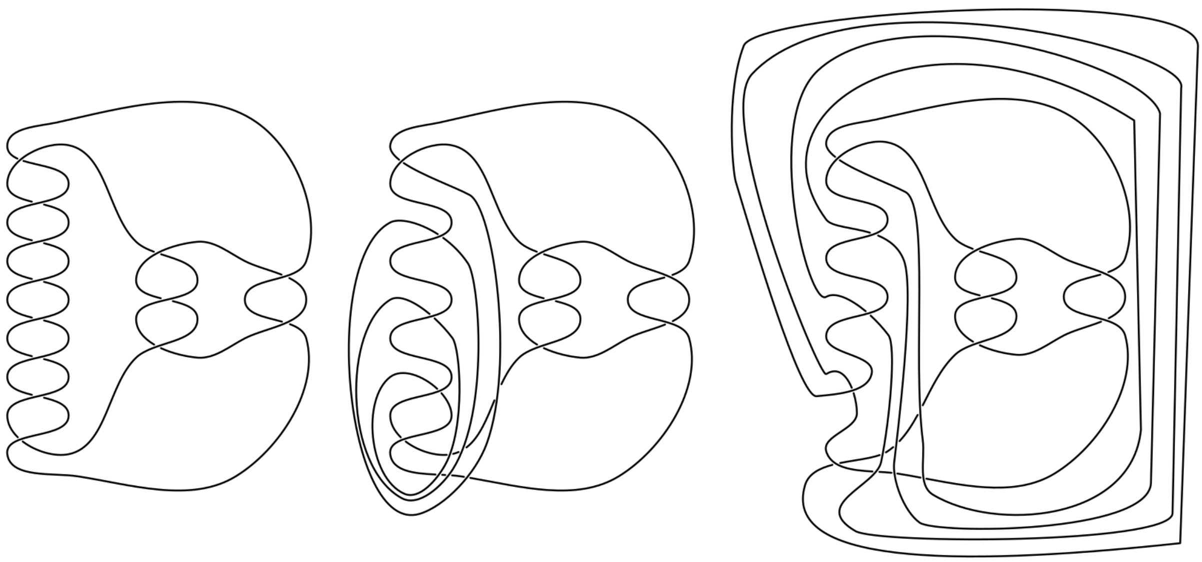


Ilustración 170. Tres movimientos básicos y 3 de polo sur en P(8, 3, 2).

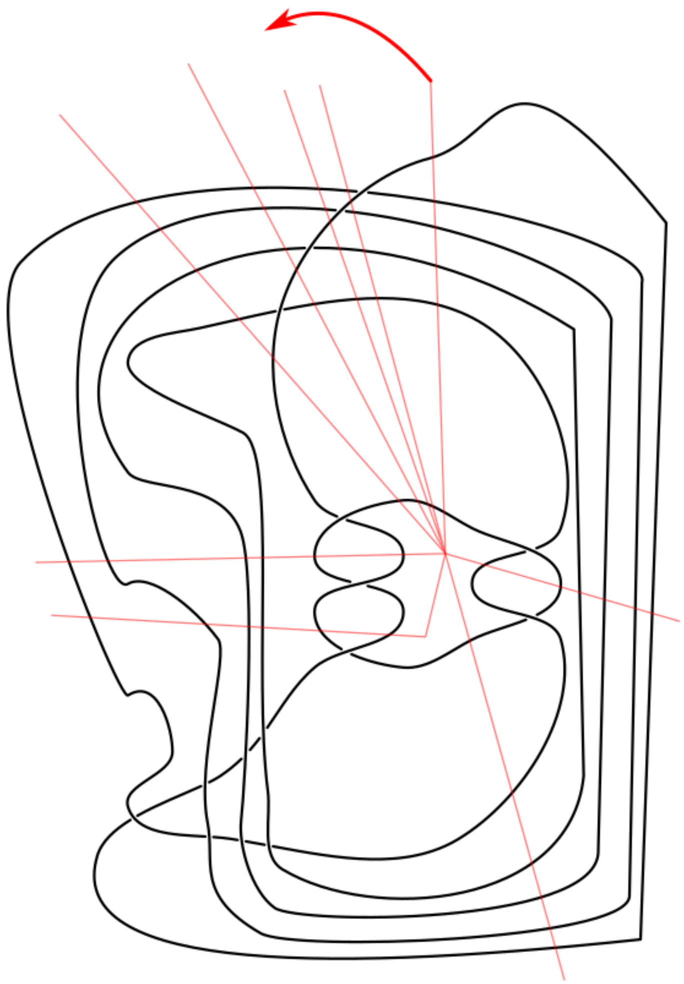


Ilustración 171. P(8, 3, 2) con sus cruces de trenza marcados.

Comparamos estos nudos con los del caso P(Par, impar, impar):

P(4, 7, 6): -1, -2, 3, 3, 3, 3, 3, 3, 3, -2, -1, 2, 3, 3, 3, 3, 3, 3

P(8, 3, 2): -1, -2, -3, -4, 5, 5, 5, -4, -3, -2, -1, 2, 3, 4, 5, 5

P(6, 5, 7): -1, -2, -3, 4, 4, 4, 4, 4, -3, -2, -1, 2, 3, 4, 4, 4, 4, 4, 4, 4

Como se puede observar claramente, comparten el algoritmo generador. Recordémoslo:

$$-1 \downarrow -\frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{b}, -\frac{|a|}{2} \uparrow -1, 2 \uparrow \frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{c}$$

Ecuación 20. Algoritmo para P(par, impar, impar) y para P(par, impar, par).

Y, como siempre, vamos a comprobar, recordando que (a, b, c) = (8, 3, 2):

$$-1 \downarrow -\frac{8}{2}, \left(\frac{8+2}{2}\right)^{3}, -\frac{8}{2} \uparrow -1, 2 \uparrow \frac{8}{2}, \left(\frac{8+2}{2}\right)^{2}$$

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

Al tener el mismo algoritmo que P(par, impar, impar) la influencia de los signos es la misma.

Enlaces P(impar, par, par)

Para ir al grano vamos a empezar por un buen nudo, el P(7, 6, 10), que tiene todos sus números diferentes y suficientemente grandes.

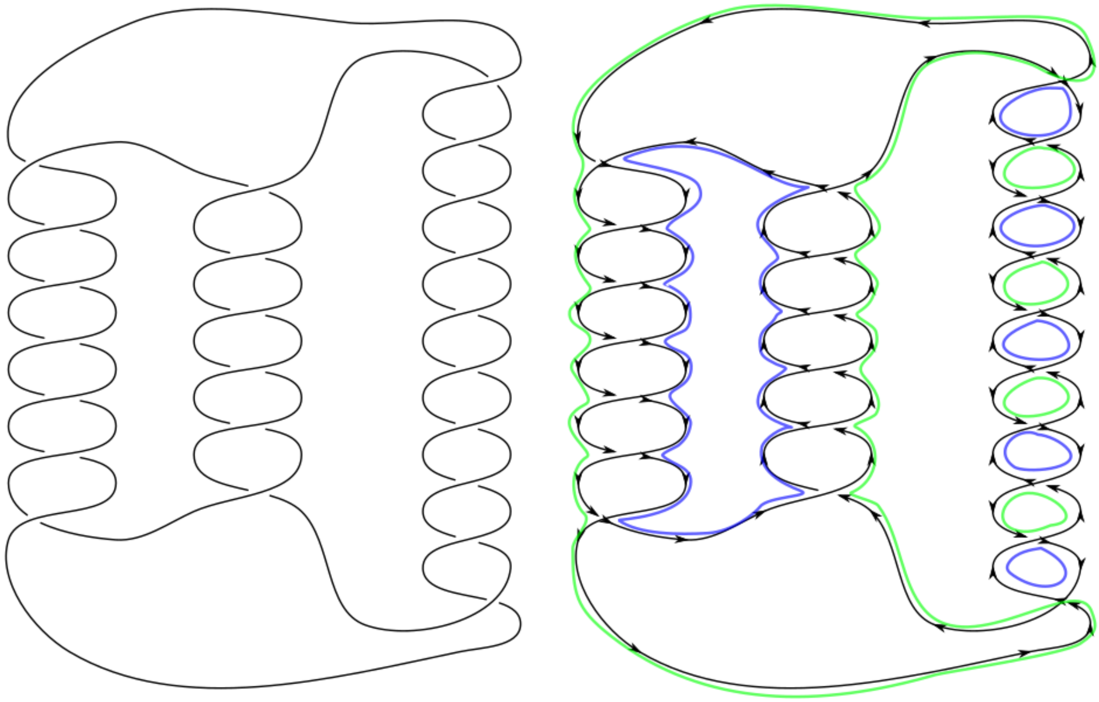


Ilustración 172. P(7, 6, 10), su orientación y sus círculos de Seifert.

Orientamos de la misma manera que en el caso anterior, de forma que la columna de círculos de Seifert esté en la columna de entrada par a la izquierda de la columna de entrada impar.

Sorprendentemente, parece ser muy parecido al caso Nudos P(par, impar, impar) y a Enlaces P(par, impar, par), sólo que con la columna que hay que desarrollar a la derecha. Hagamos los movimientos básicos y de polo sur y veamos qué ocurre.

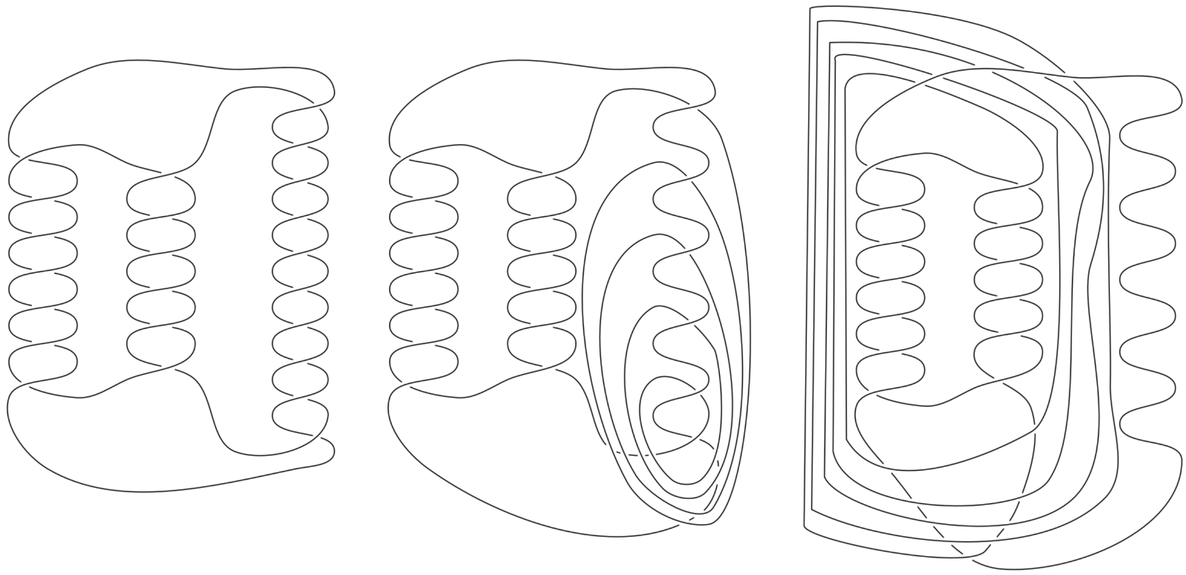


Ilustración 173. Cuatro movimientos básicos y de polo sur en P(7, 6, 10).

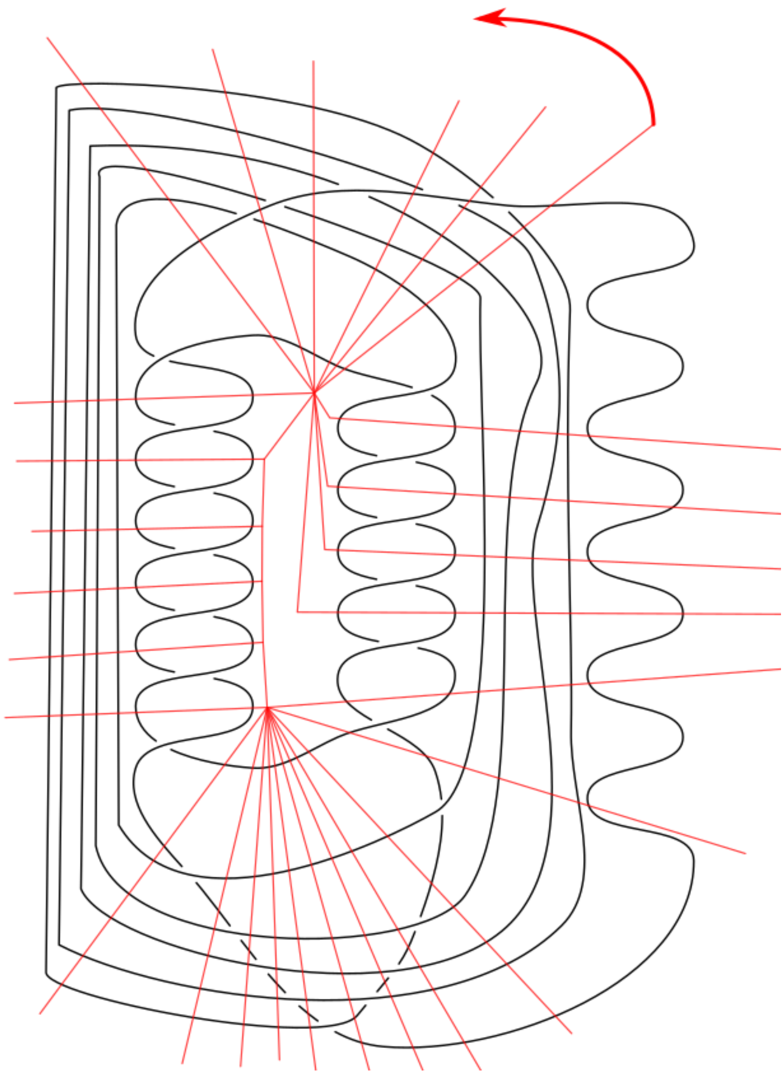


Ilustración 174. P(7, 6, 10) y los cruces de su trenza.

Para comprobar si nos sirve el mismo algoritmo vamos a comparar las trenzas:

P(7, 6, 10): -1, -2, -3, -4, -5, 6, 6, 6, 6, 6, 6, 6, -5, -4, -3, -2, -1, 2, 3, 4, 5, 6, 6, 6, 6, 6, 6

P(4, 7, 6): -1, -2, 3, 3, 3, 3, 3, 3, 3, -2, -1, 2, 3, 3, 3, 3, 3, 3

P(6, 5, 7): -1, -2, -3, 4, 4, 4, 4, 4, -3, -2, -1, 2, 3, 4, 4, 4, 4, 4, 4, 4

Mientras que en P(7, 6, 10) hemos hecho los movimientos básicos en la columna derecha, en P(4, 7, 6) y en P(6, 5, 7) los hemos hecho en la izquierda.

Si nos fijamos, en P(4, 7, 6) y en P(6, 5, 7) la entrada de las columnas que no realizan los movimientos básicos (central y derecha) nos indica las veces que se repetirá el número (movimientos básicos + 1), mientras que en P(7, 6, 10) son las entradas izquierda y central las que nos dictan cuántas veces se repetirá.

Es decir, nuestra variable principal es 4 en P(4, 7, 6), 6 en P(6, 5, 7) y 10 en P(7, 6, 10). Analicemos el algoritmo ya existente para el caso de Nudos P(par, impar, impar):

$$-1 \downarrow -\frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{b}, -\frac{|a|}{2} \uparrow -1, 2 \uparrow \frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^{c}$$

Lo que en el antiguo algoritmo era a ahora tendrá que ser c, lo que era b ahora será a y lo que era c ahora será b. El algoritmo quedaría así:

$$-1 \downarrow -\frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^{a}, -\frac{|c|}{2} \uparrow -1, 2 \uparrow \frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^{b}$$

Ecuación 21. Algoritmo para P(impar, par, par).

Para confirmarlo, vamos a probar la hipótesis con nuestro (a, b, c) = (7, 6, 10):
-1, -2, -3, -4, -5, 6, 6, 6, 6, 6, 6, 6, -5, -4, -3, -2, -1, 2, 3, 4, 5, 6, 6, 6, 6, 6, 6

$$-1 \downarrow -\frac{10}{2}, \left(\frac{10+2}{2}\right)^{7}, -\frac{10}{2} \uparrow -1, 2 \uparrow \frac{10}{2}, \left(\frac{10+2}{2}\right)^{6}$$

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

Como hemos visto tras actualizar la posición de las variables la ecuación 21 se verifica.

Enlaces P(par, par, impar)

Como hemos dicho en la subsección Propiedad simétrica y demostrado previamente en la página 83, el algoritmo para P(a,b,c) es el mismo que el de P(c,b,a). Basta entonces usar el algoritmo para el caso ya visto P(impar,par,par) teniendo en cuenta el invertir las entradas.

Enlaces P(par, par, par)

Para terminar, nos queda la última posibilidad en un pretzel de 3 columnas: par, par, par. Antes de empezar podríamos pensar que este caso va a ser otra odisea de 32 páginas como el caso P(impar, impar, impar), ya que ambos tienen las 3 entradas del mismo tipo.

Aunque en un principio parezca demasiado sencillo, me voy a basar en P(6, 8, 2) para analizar este caso, y en el caso de que se complique habrá que preparar números mayores.

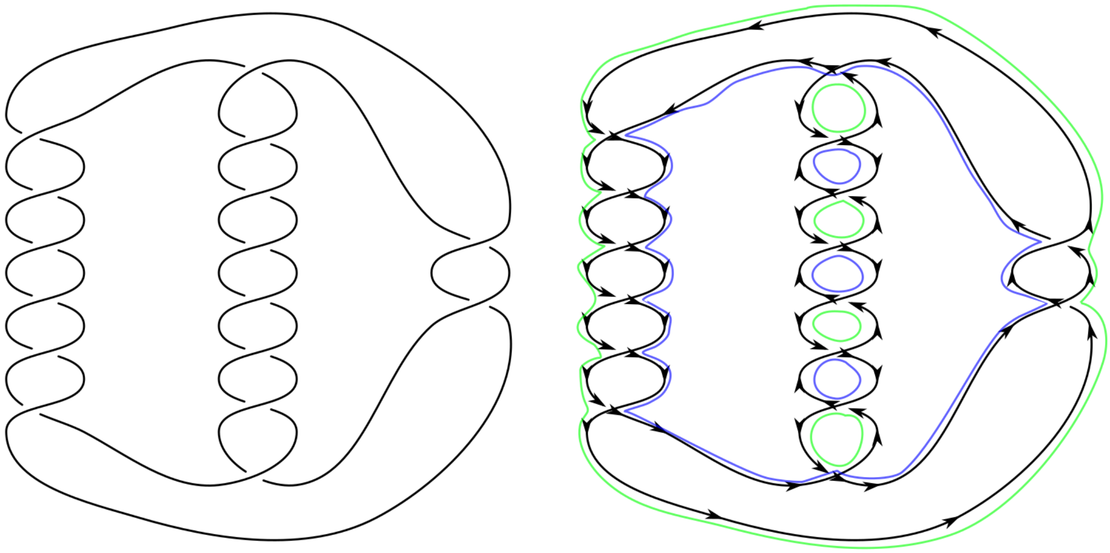


Ilustración 175. P(6, 8, 2) orientado y con sus círculos de Seifert.

Estudiar los círculos de Seifert nos revela algo extremadamente esperanzador, y es que tenemos dos círculos exteriores compatibles y una columna central de círculos, al estilo del caso Nudos P(impar, par, impar). Probemos a resolverlo con el mismo método, haciendo b/2—1 movimientos básicos.

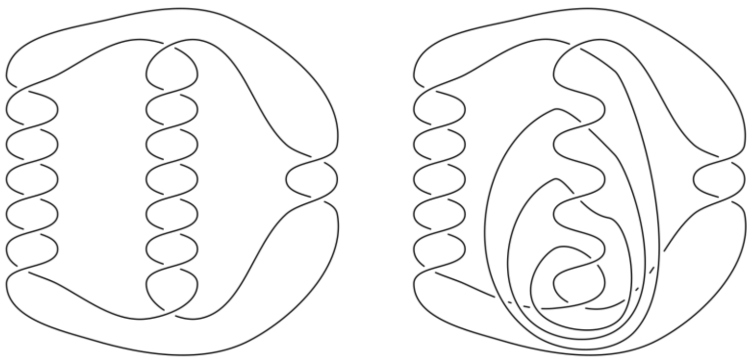


Ilustración 176. Tres movimientos básicos en la columna central.

Como era de esperar, ha funcionado y ya tenemos nuestra trenza:

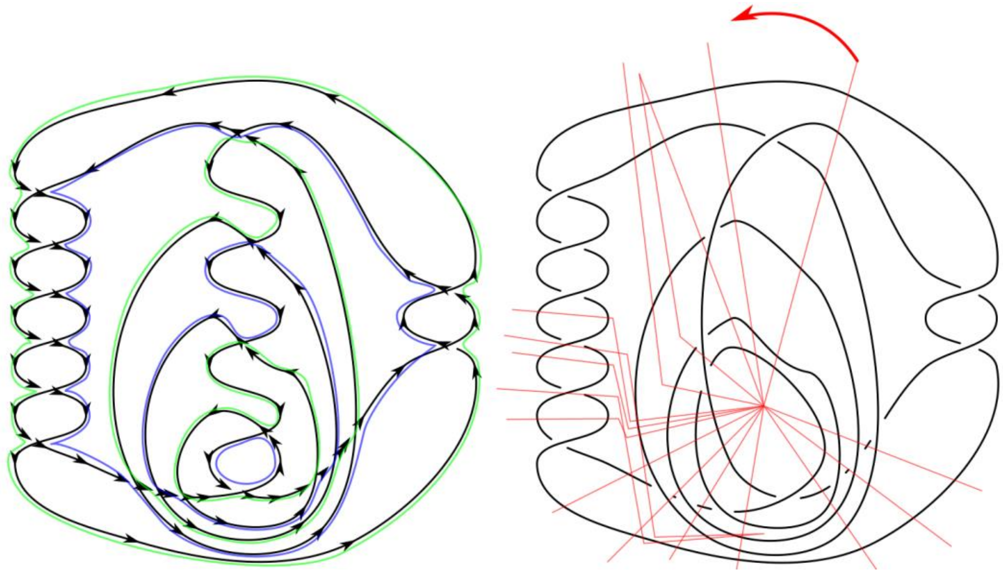


Ilustración 177. P(6, 8, 2) transformado en trenza, orientado y con los cruces marcados.

[-2, -3, -4, -5, 1, 1, 1, 1, 1, 1, 2, 3, 4, -5, -4, -3, -2, 1, 1]

Ahora vemos que el parecido con el caso Nudos P(impar, par, impar) es absoluto. Vamos a comprobar si nos sirve su algoritmo (usando a, b y c como entrada):

$$\boxed{-2 \downarrow -\frac{|b|+2}{2}, 1^{a}, 2 \uparrow \frac{|b|}{2}, -\frac{|b|+2}{2} \uparrow -2, 1^{c}}$$

Ecuación 22. Algoritmo para P(impar, par, impar) y para P(par, par, par).

$$-2 \downarrow -\frac{8+2}{2}, 1^{6}, 2 \uparrow \frac{8}{2}, -\frac{8+2}{2} \uparrow -2, 1^{2}$$

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

Ya podemos afirmar que es el mismo algoritmo.

Enlaces pretzel con tres entradas. Síntesis

Vamos a recopilar todos los algoritmos elaborados hasta ahora, siendo *P(a, b, c)* un pretzel de tres entradas. Entonces este enlace sería la clausura de la trenza

A. Si todas las entradas son pares o sólo lo es la entrada central b:

$$-2\downarrow -\frac{|b|+2}{2}, 1^a, 2\uparrow \frac{|b|}{2}, -\frac{|b|+2}{2}\uparrow -2, 1^c$$

B. Si a es par y b y c son impares, o bien a y c son pares y b es impar:

$$-1\downarrow -\frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^b, -\frac{|a|}{2}\uparrow -1, 2\uparrow \frac{|a|}{2}, \left(\frac{|a|+2}{2}\right)^c$$

C. Si a y b son impares y c es par:

$$-1\downarrow -\frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^b, -\frac{|c|}{2}\uparrow -1, 2\uparrow \frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^a$$

D. Si a es impar y b y c son pares:

$$-1\downarrow -\frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^a, -\frac{|c|}{2}\uparrow -1, 2\uparrow \frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^b$$

E. Si a y b son pares y c es impar:

$$-1\downarrow -\frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^b, -\frac{|c|}{2}\uparrow -1, 2\uparrow \frac{|c|}{2}, \left(\frac{|c|+2}{2}\right)^a$$

F. Si a, b y c son impares:

$$-1\downarrow -(a'+1), -(a'+1)\uparrow -2$$
$$\{(-2-a')\uparrow(-3)^{\downarrow b'\downarrow}\}$$
$$\{(2+b'+a')\downarrow(3+b')^{\uparrow c'\uparrow}\}$$
$$\{(2+b')\downarrow(3)^{\uparrow c'\uparrow}\}$$
$$2\uparrow(1+c')$$
$$-1\downarrow -(c'+1), -(c'+1)\uparrow -2$$
$$\{(-2-c')\uparrow(-3)^{\downarrow b'\downarrow}\}$$
$$2\uparrow(1+b')$$
$$-1\downarrow -(b'+1), -(b'+1)\uparrow -2$$
$$\{(-2-b'-c')\downarrow(-3-b')^{\uparrow a'\uparrow}\}$$
$$\{(2+b')\downarrow(3)^{\uparrow a'\uparrow}\}$$
$$2\uparrow(1+a')$$

donde

$$a' = \frac{|a|-1}{2},\ \ b' = \frac{|b|-1}{2},\ \ c' = \frac{|c|-1}{2}.$$

El problema es que estos enunciados están incompletos, porque no tienen en cuenta la influencia de la orientación cuando hay más de una componente, es decir, en los enlaces. Incluso es posible que usando otras orientaciones coincidan los algoritmos, quedándonos con menos casos. Por ello vamos a intentar abordarlo desde otra perspectiva.

Podemos clasificar los algoritmos en 2 tipos, los que se componen de 5 progresiones (A, B, C, D y E) y el algoritmo F, compuesto por un número mayor de agrupaciones. Es más, dentro de este primer tipo, los algoritmos B, C, D y E son extraordinariamente parecidos, lo que puede hacer que el lector se pregunte si los podemos expresar de una manera única. Recordamos que partimos de:

A: Si todas las entradas son pares o sólo lo es la entrada central b.
B: Si a es par y b y c son impares, o bien a y c son pares y b es impar.
C: Si a y b son impares y c es par.
D: Si a es impar y b y c son pares.
E: Si a y b son pares y C es impar.

Como vimos en la subsección Enlaces P(par, par, impar), E se obtiene del simétrico de D. Es más, si nos vamos a la demostración de D en la subsección Enlaces P(impar, par, par), veremos que D se obtiene basándonos en B. Por otra parte, C se obtiene del simétrico de B.

Podemos intuir que deberíamos ser capaces de quedarnos con 3 algoritmos, el A, uno que englobe B-C-D-E y el F.

Si volvemos a cada uno de los casos vistos, siempre comenzamos estudiando los círculos de Seifert para ver qué tenemos que hacer y no estrellarnos contra callejones sin salida como concursantes de Humor Amarillo. Hay un factor fundamental, la base de toda esta síntesis, que no ha sido descubierto hasta bien pasado el momento que supusimos iba a ser la finalización del TFG. Por eso está estructurado de esta manera poco convencional: dando primero el teorema final, luego distrayendo con 60 páginas que ahora sabemos podrían haberse estudiado de forma más directa, para finalmente llegar a esta estrategia definitiva.

La clave es estudiar la columna del pretzel formada por círculos de Seifert. Lo que tienen en común el teorema A, B, C, D y E es que tienen una (única) columna de círculos de Seifert.

A: Si todas las entradas son pares o sólo lo es la entrada central b, la columna central se compone de círculos de Seifert.

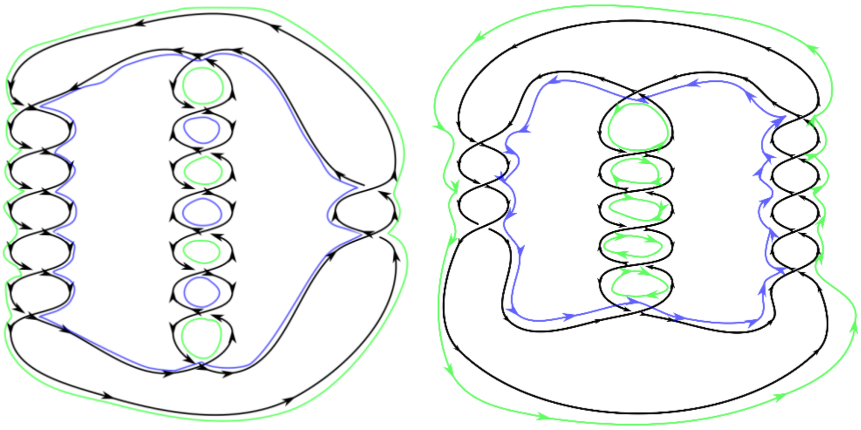


Ilustración 178. P(par, par, par) y P(impar, par, impar).

B: Si a es par y b y c son impares, o bien a y c son pares y b es impar. La columna izquierda se compone de círculos de Seifert.

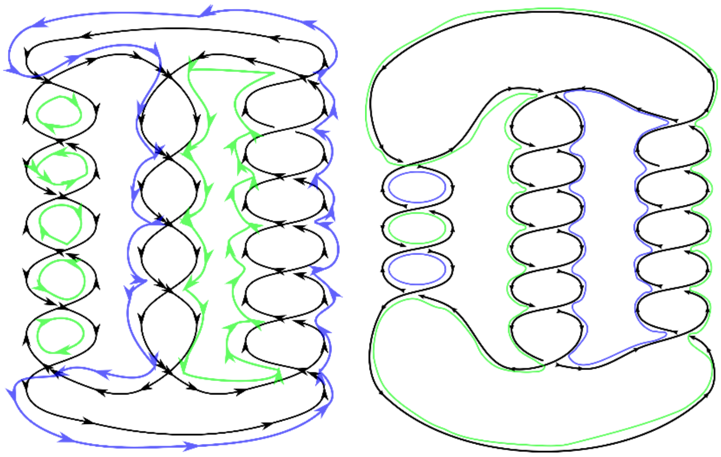


Ilustración 179. P(par, impar, par) y P(par, impar, par).

C: Si a y b son impares y c es par. La columna derecha se compone de círculos de Seifert.

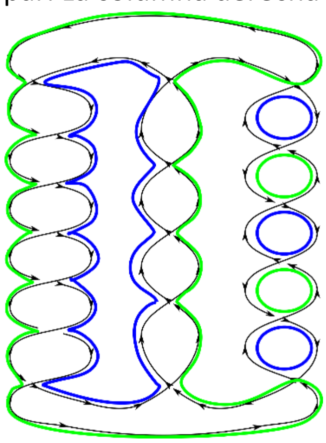


Ilustración 180. P(impar, impar, par).

D: Si a es impar y b y c son pares. La columna derecha se compone de círculos de Seifert.

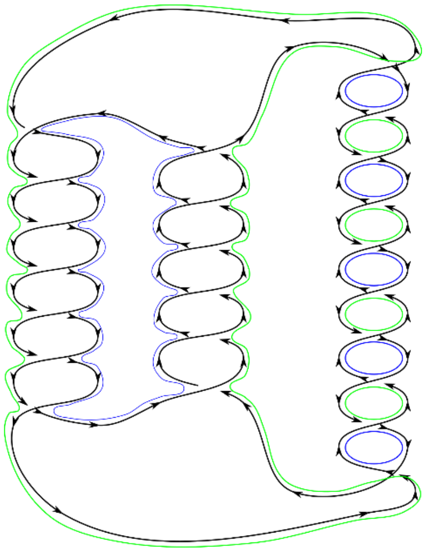


Ilustración 181. P(impar, par, par).

E: Si a y b son pares y c es impar. La columna izquierda se compone de círculos de Seifert.

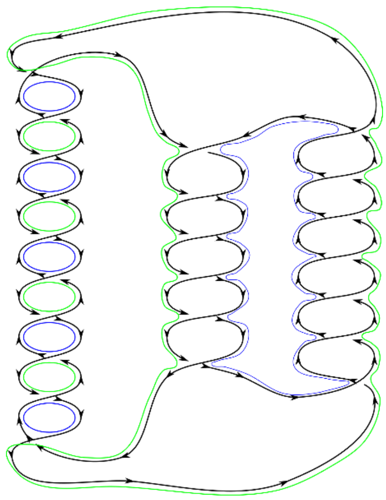


Ilustración 182. P(par, par, impar).

Pues bien, tal y como vimos al estudiar la propiedad circular, podemos desplazar todas las columnas del pretzel a izquierda o derecha. De esta manera podemos dejar que la (única) columna compuesta de círculos de Seifert acabe en el centro, lo que nos deja en el caso A. Este es el caso preferible, ya que su resolución sólo necesita movimientos básicos.

Entonces llegamos a una nueva estrategia: para pretzel de 3 entradas, siempre que no sean las 3 impares, localizamos la columna compuesta formada por círculos de Seifert (la columna en la que las orientaciones sean contrarias). Si no está en el centro, realizamos un movimiento cíclico para llevarla al centro, seguidamente realizamos los movimientos básicos necesarios y tendremos la trenza. Y así llegamos al enunciado final del teorema 3 (ver página 51).

Después de los métodos que hemos visto hasta ahora y el trabajo que requieren, parece maravilloso que podamos hacerlo así de simple, así que vamos a ver un ejemplo, P(7, 6, 10).

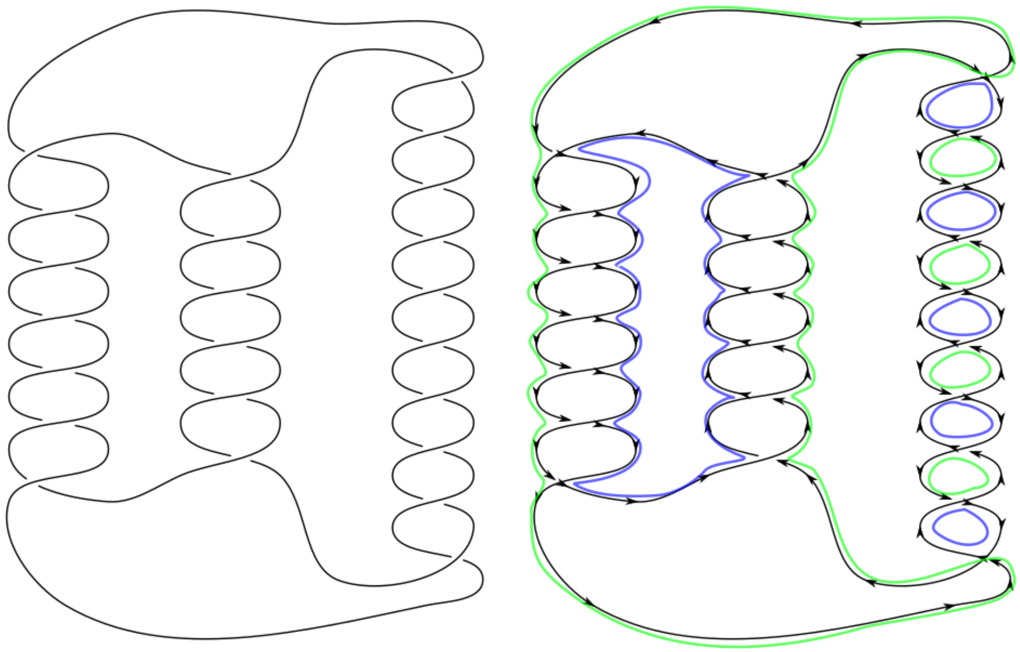


Ilustración 183. P(7, 6, 10) y sus círculos de Seifert, siendo la columna derecha círculos de Seifert.

Vemos que la columna de círculos de Seifert está a la derecha, así que la llevamos al centro haciendo un desplazamiento cíclico a la izquierda.

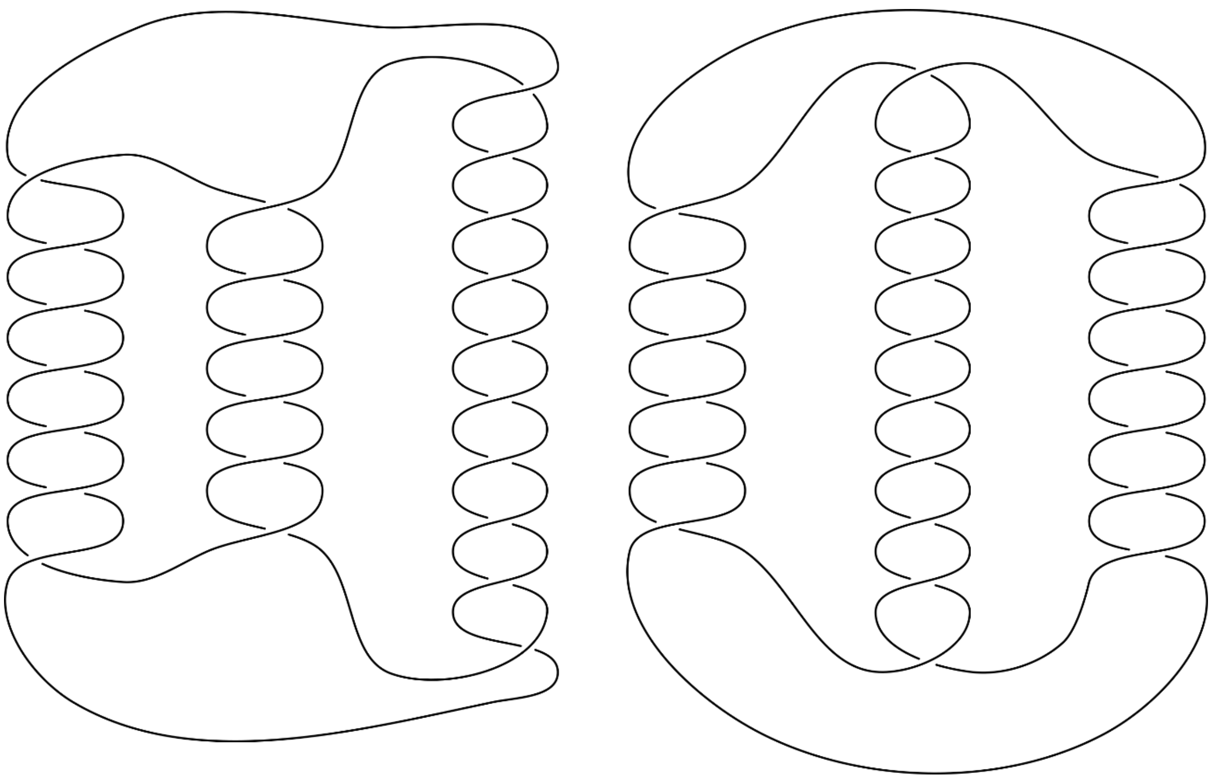


Ilustración 184. P(7, 6, 10) y P(6, 10, 7), que son el mismo enlace pretzel.

Si ahora estudiamos los círculos de Seifert veremos que tenemos la columna central compuesta de dichos círculos, lo que significa que tras hacer movimientos básicos llegaremos a la trenza.

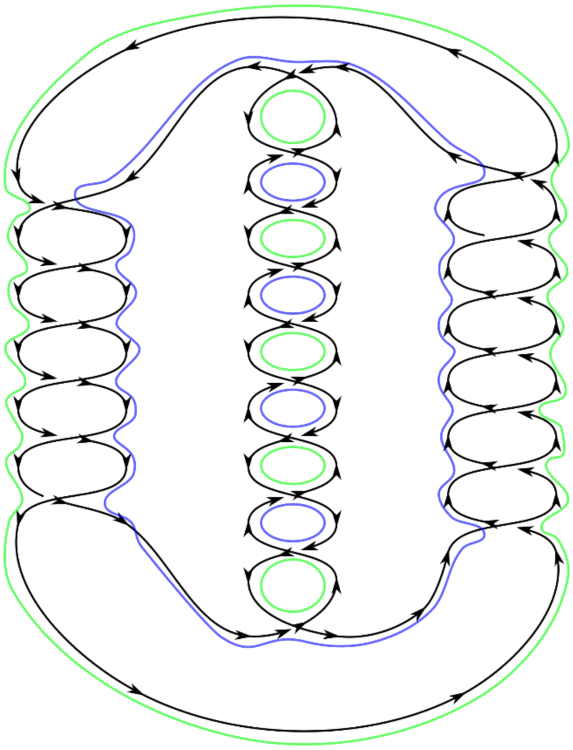


Ilustración 185. P(6, 10, 7) y sus círculos de Seifert.

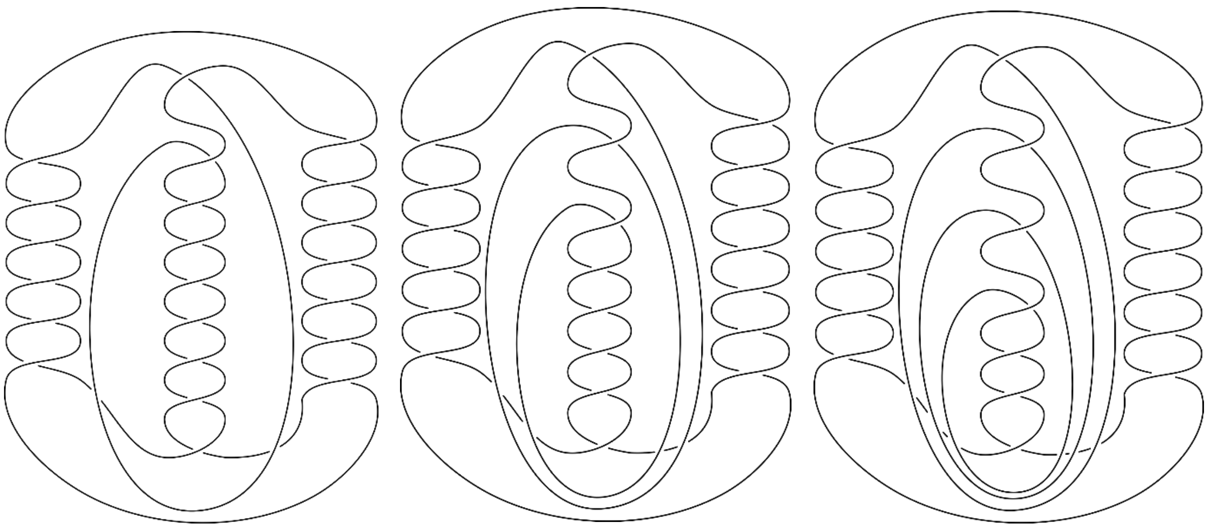


Ilustración 186. Movimientos básicos en P(6, 10, 7).

Nos queda un último movimiento básico, pero ya habremos obtenido la trenza, pues si orientamos el diagrama veremos que todas las cuerdas avanzan en sentido antihorario, con centro en la estrella.

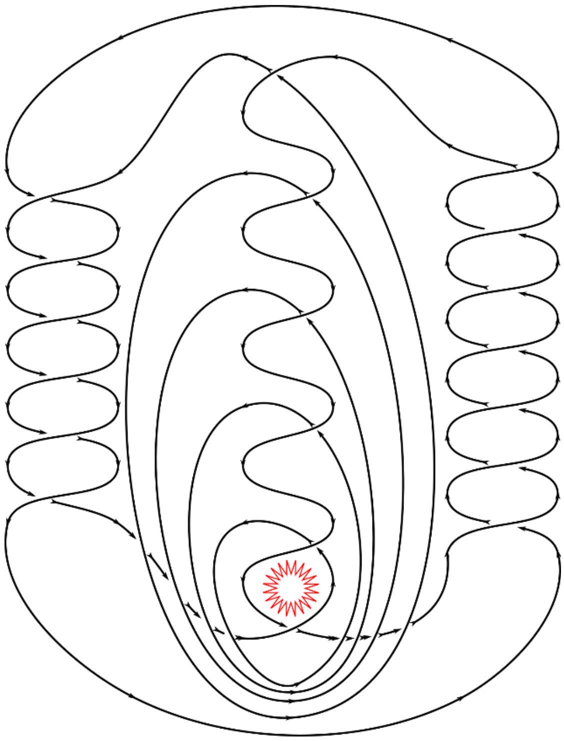


Ilustración 187. Trenza de P(6, 10, 7) y, a su vez, de P(7, 6, 10).

Ahora podemos decir con total confianza que podemos usar el algoritmo A para los casos A, B, C, D y E (teniendo en cuenta que la columna central ha de ser par y tiene que tener las orientaciones cruzadas).

Para terminar, tenemos que recordar que en un enlace podemos cambiar la localización de la columna que tiene orientaciones contrarias cambiando la orientación de una de sus cuerdas, como se ve a continuación con la cuerda roja.

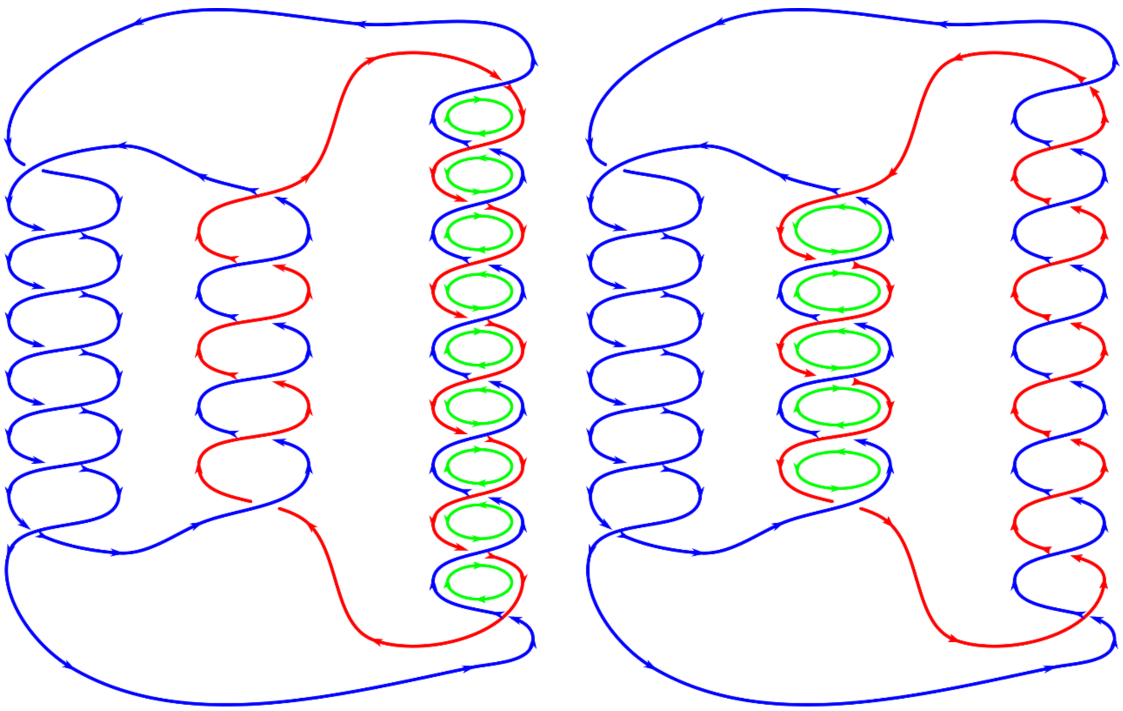


Ilustración 188. Influencia de la orientación de las cuerdas en los círculos de Seifert.

Enlaces pretzel con cuatro entradas

Hemos llegado muy lejos, y surge la duda de lo que ocurrirá con 4 columnas. ¿Servirán los métodos usados hasta ahora? ¿Podrán los movimientos básicos ser el as en la manga que nos permita atacarlos? La resolución es realmente sorprendente, sobre todo porque tiene cierta similitud con los movimientos básicos, pero absolutamente ninguna con los métodos usados hasta ahora. Pero no nos adelantemos; empecemos poco a poco como hasta ahora.

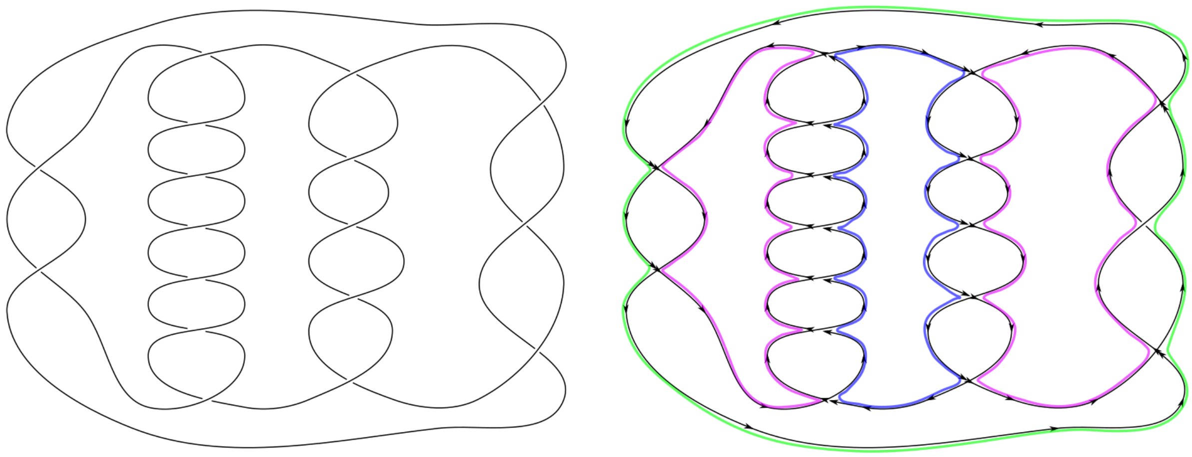


Ilustración 189. P(2, 7, 5, 3), su orientación y sus círculos de Seifert.

Ciertamente nos encontramos en una situación insólita. No tenemos ni un solo círculo de Seifert en las columnas; los tenemos en los huecos entre columnas. Si hacemos movimientos básicos va a empeorar la situación, así que no podemos seguir usando el mismo método. Esta situación es independiente de la cantidad de números pares o de su posición.

Si analizamos tenemos complejidad 2, una se debe al círculo azul y al verde, y otra a los dos rosas. De una cosa podemos estar seguros, y es que siguiendo paso a paso el método en Cromwell [14] llegaremos a la trenza buscada. Así que vamos a reducir por arriba el círculo azul con el verde y por abajo los rosas. El azul, por ser central, lo pasaremos siempre por debajo. Y el rosa derecho siempre sobre el izquierdo.

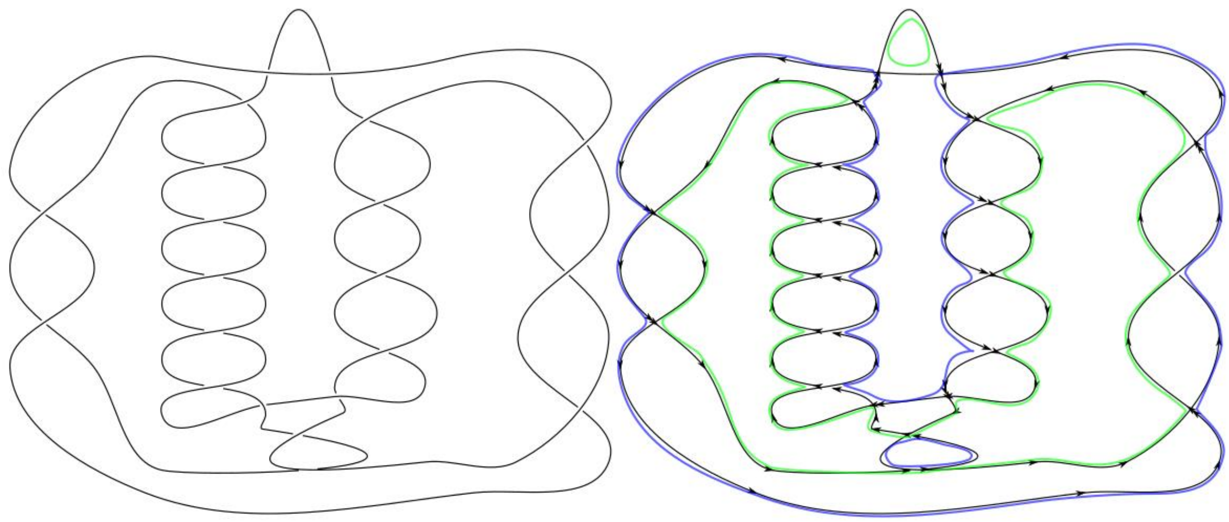


Ilustración 190. Dos movimientos de reducción y nuevos círculos de Seifert.

Por suerte parece que ya son todos compatibles y sólo falta un movimiento polo sur.

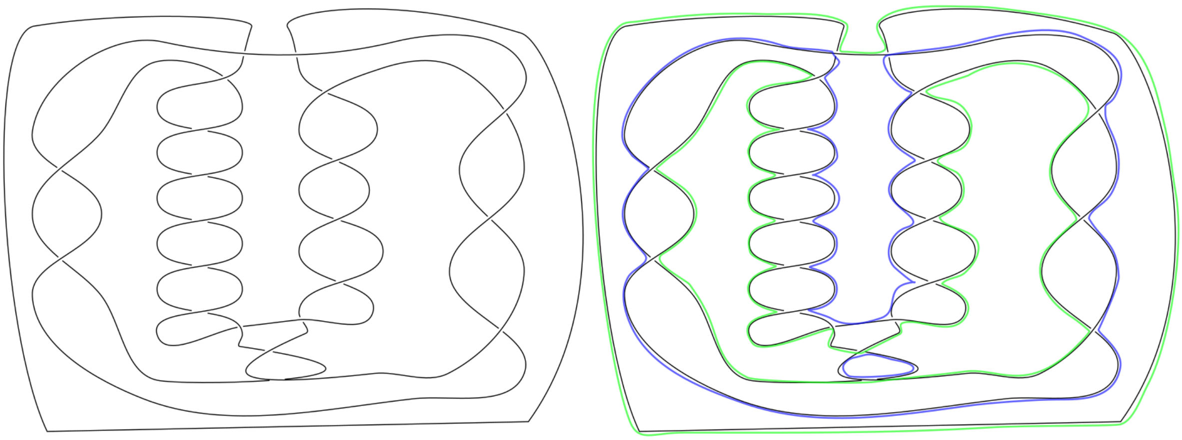


Ilustración 191. P(2, 7, 5, 3) y su trenza

Por curioso que parezca, ninguna de las columnas tiene influencia en el método que se usa, excepto su signo. Vamos a estudiar los cruces y a extraer el algoritmo. Aunque ya podemos predecir que los cruces van a ser 2+7+5+3+2x[número de reducciones]=21.

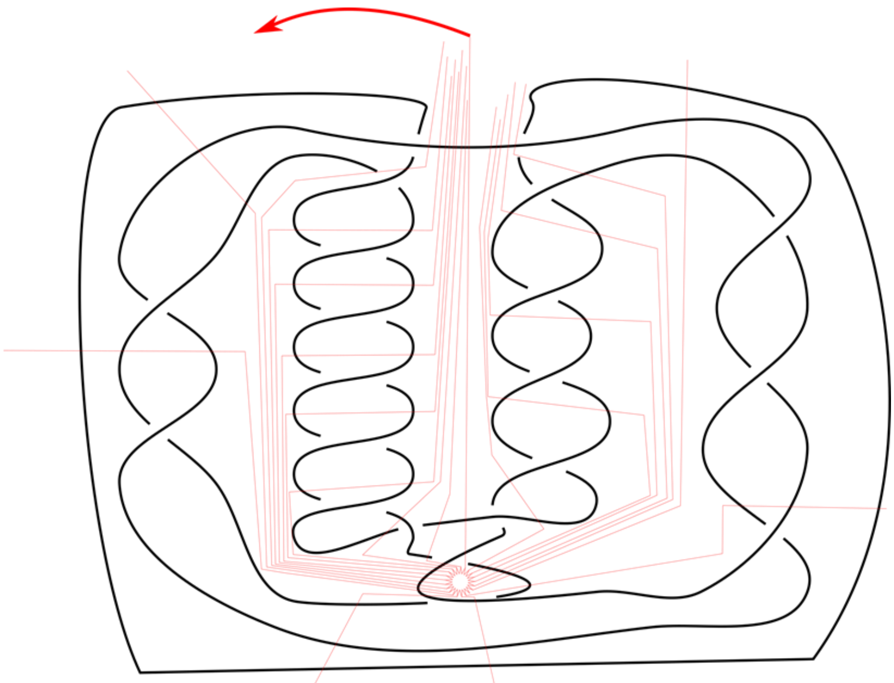


Ilustración 192. Cruces de la trenza de P(2, 7, 5, 3).

Nuestra predicción ha sido correcta y tenemos 21 cruces. En concreto:

-3,2,2,2,2,2,2,2,-1,2,2,3,2,2,2,1,2,2,2,2,2

Podríamos decir que nuestro algoritmo es 3, 2, 1, 2, 3, 2, 1, 2, sólo que los 2 se repiten b, a, d y c veces respectivamente (b=7, a=2, d=3, c=5).

$$\boxed{-3, 2^b, -1, 2^a, 3, 2^d, 1, 2^c}$$

Ecuación 23. Algoritmo para 4 columnas.

Si cambiamos el signo de alguna columna cambiará el signo del 2 que corresponde a esa columna.

En este momento hice el descubrimiento de los desplazamientos de columna. No es moco de pavo haber obtenido el algoritmo para 4 columnas en apenas 2 páginas, pero hay algo de la Ilustración 189 que me llamó especialmente la atención. Es algo que no es inmediato, pero a mí me salta a la vista tras haber dibujado tantos nudos. Para explicarme voy a compararla con P(1, 4, 1).

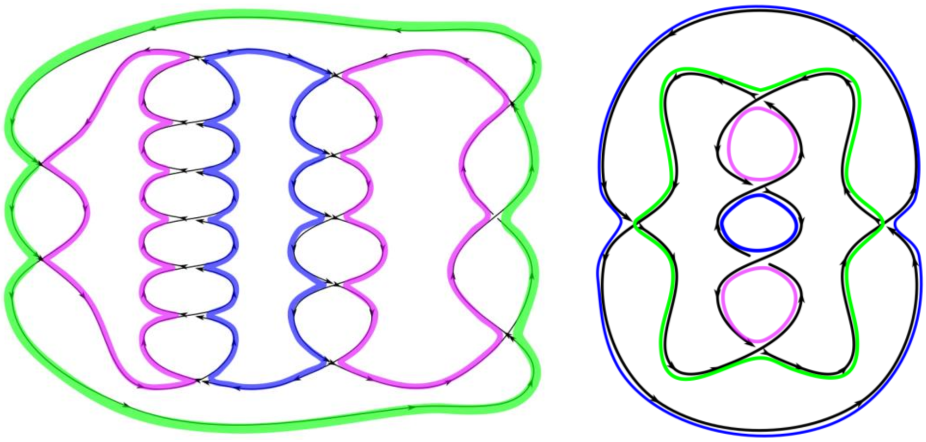


Ilustración 193. Comparación P(2, 7, 5, 3) y P(1, 4, 1).

Todavía no es evidente a lo que me refiero, comparemos con P(1, 4, 1) girado 90 grados.

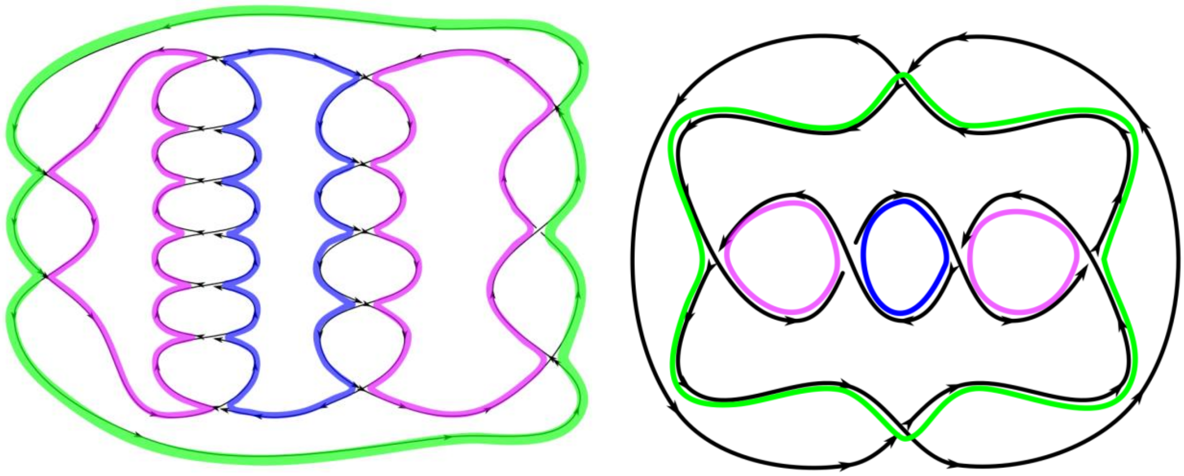


Ilustración 194. Comparación P(2, 7, 5, 3) y P(1, 4, 1) girado 90 grados.

Este ya es el ejemplo perfecto de lo que quiero decir. En ambos tenemos un círculo que tiene dentro tres círculos, siendo el círculo exterior incompatible con el del centro y los de los lados entre ellos.

Imaginemos que en P(2, 7, 5, 3) comprimimos las columnas para que sean de un cruce, como en P(1, 4, 1). Tendríamos exactamente la imagen de P(1, 4, 1) girada 90 grados. ¿Y cómo lo resolvíamos? Haciendo un único movimiento básico. Pues esto lo podemos resolver moviendo la columna entera, no sólo una cuerda, hasta afuera.

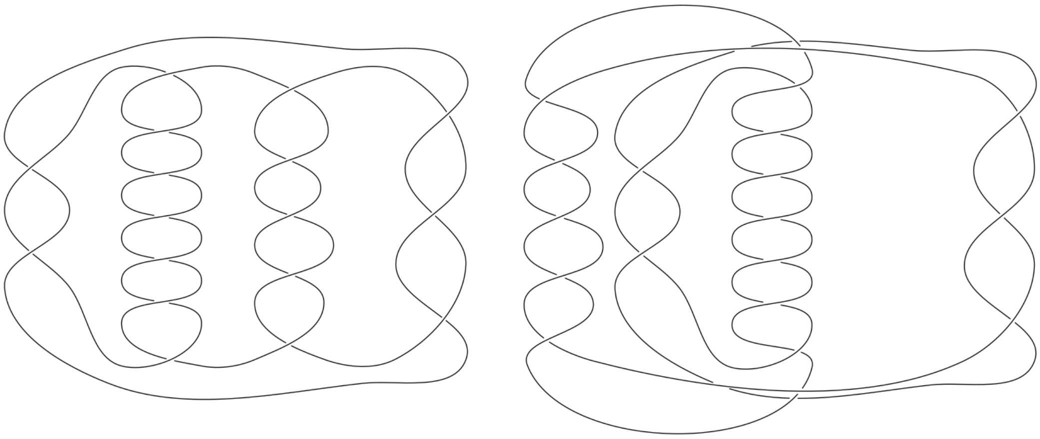


Ilustración 195. P(2, 7, 5, 3) tras un desplazamiento de la tercera columna hacia la izquierda.

Hay que recordar que como llevamos la columna por arriba, los cruces en las cuerdas existentes serán por arriba, y el que se genera entre las dos cuerdas de nuestra columna lleva por arriba la cuerda de la derecha (es físicamente imposible que vaya por debajo). Pues resulta que esta es la trenza.

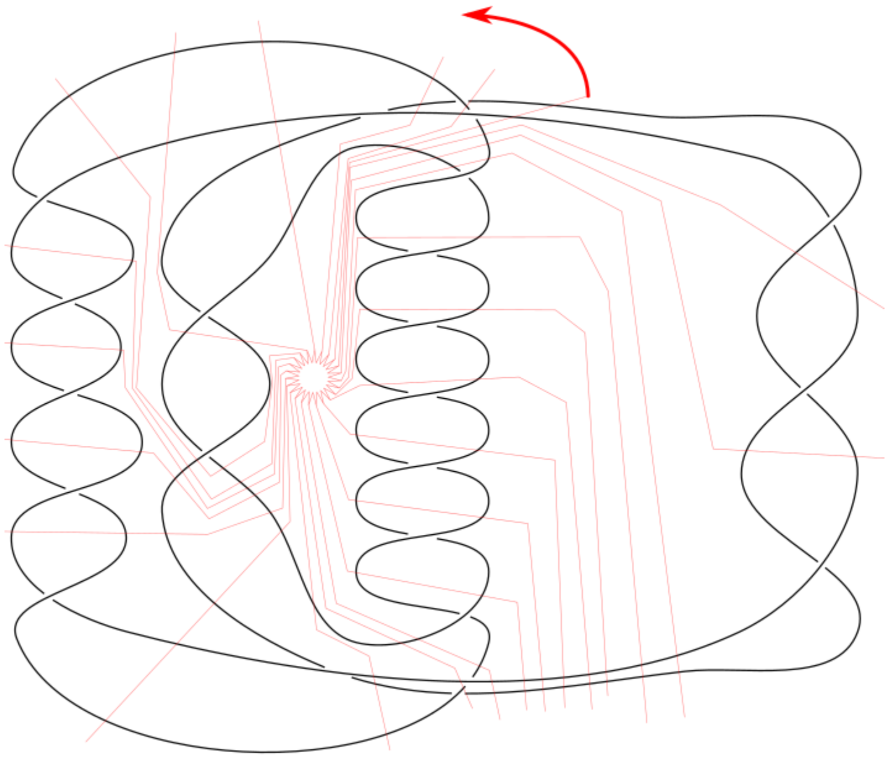


Ilustración 196. Cruces de la trenza de P(2 ,7, 5, 3) utilizando movimiento de columnas.

Para empezar con las diferencias, ahora tenemos 2+7+5+3+3x2x[movs.columnas] cruces = 23.

-2, 1, 2, 3, 3, 1, 1, 1, 1, 1, -2, -1, 2, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1

$$\boxed{-2,1,2,3^a,1^c,-2,-1,2,3^b,1^d}$$

Ecuación 24. Algoritmo para 4 columnas haciendo un movimiento básico.

Si cambiamos el signo de una columna cambiará el signo de la agrupación que depende de esa columna. Los números que no dependen de columnas se mantienen iguales.

Hay gran parecido entre los dos algoritmos ¿Cuál es el mejor? Ciertamente el primero es más llamativo y menos complejo, pese a tener una agrupación más. Vamos a reformular la pregunta: si fuésemos a intentar un nudo pretzel de 6 entradas, ¿qué método querríamos seguir?

Enlaces pretzel con n entradas, n par y mayor que dos

Ya sabemos lo que ocurre si tenemos 2 o 4 columnas, pero ¿qué ocurre si tenemos 6, 8 o 10? ¿Seguiremos pudiendo resolverlo? Voy a volver a mostrar antiguas ilustraciones:

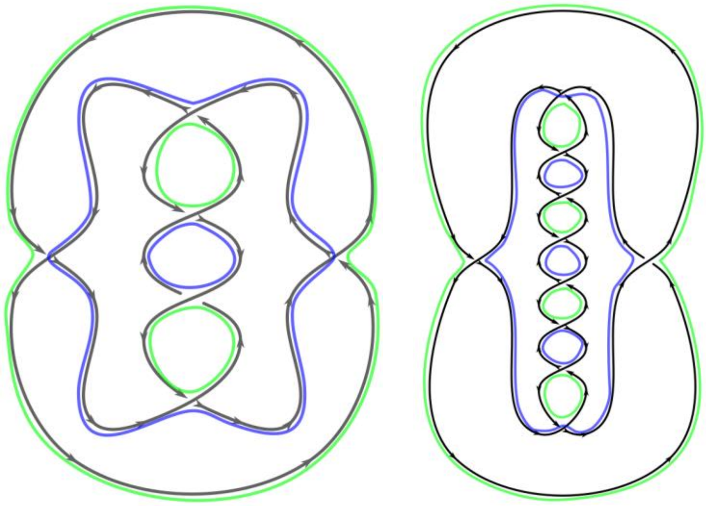


Ilustración 197. Comparación P(1, 4, 1) con P(1, 8, 1).

Cuando aumentamos el número de cruces en la columna central no surge ningún problema, simplemente hacemos un movimiento básico más. Lo que puede llevarnos a pensar que esto funcionará igual con las columnas; si hay más, hacemos más desplazamientos de columnas. Es hora de probarlo con el P(6, 4, 6, 8, 6, 4):

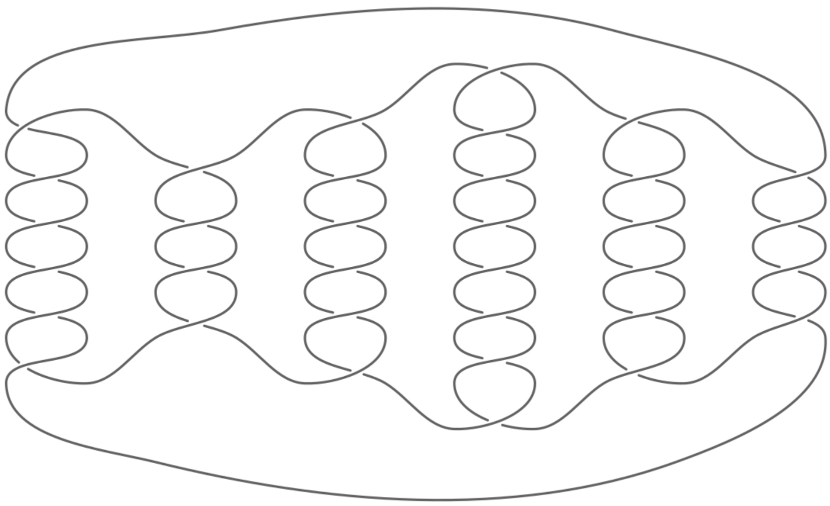


Ilustración 198. P(6, 4, 6, 8, 6, 4).

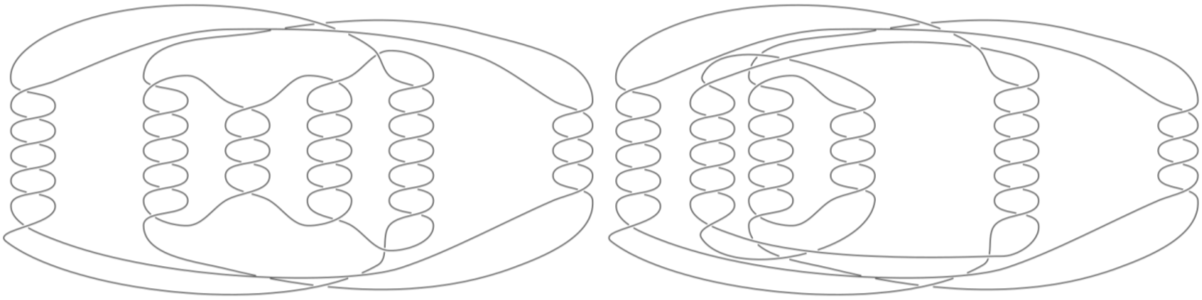


Ilustración 199. Dos desplazamientos de columnas. La 5ª pasa a 1ª posición y la 3ª a 2ª.

Parece que ha vuelto a funcionar. La estrella vuelve a estar entre las que eran la primera y la segunda columnas inicialmente, igual que pasaba con los movimientos básicos de cruces, que generaban círculos que tenían en el centro la estrella.

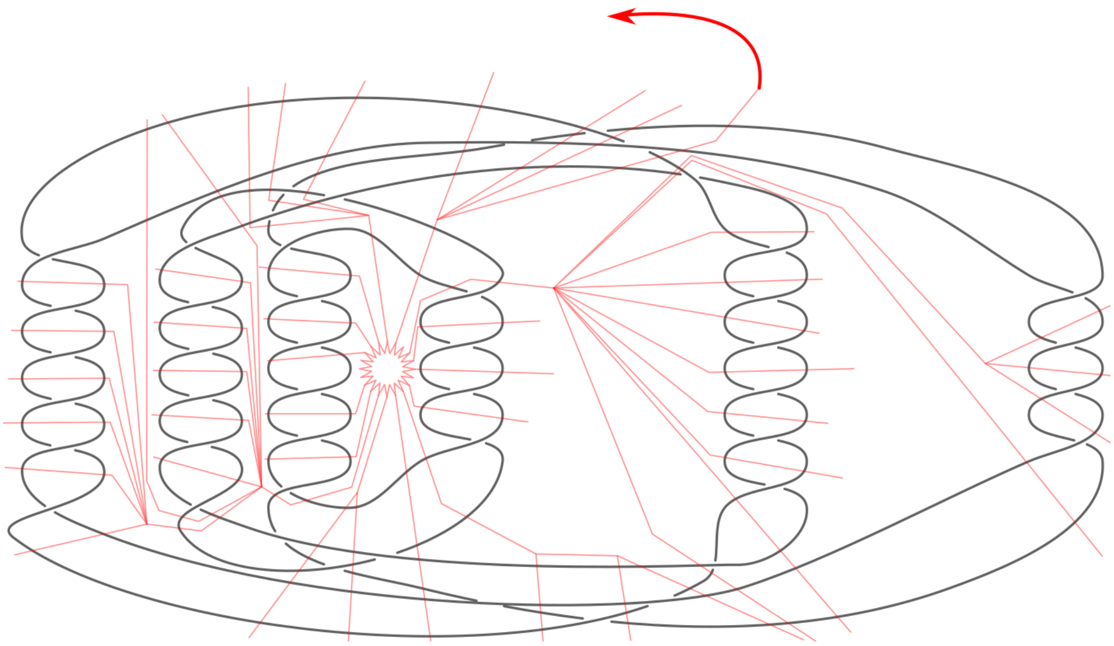


Ilustración 200. Cruces de la trenza de P(6, 4, 6, 8, 6, 4).

-2,1,2,-4,3,4,5,5,5,5,5,5,3,3,3,3,3,3,1,1,1,1,1,1,-4,-3,4,-2,-1,2,5,5,5,5,3,3,3,3,3,3,3,3,1,1,1,1

$$\boxed{-2,1,2;-4,3,4;5^{a},3^{c},1^{e};-4,3,4;-2,-1,2;5^{b},3^{d},1^{f}}$$

Ecuación 25. Algoritmo para seis columnas.

Si estudiamos el algoritmo para N columnas**,** tenemos que tener en cuenta que nos enfrentamos a progresiones de agrupaciones.

$$\left\{-2,1,2;-4,3,4;-6,5,6;\dots;\left(\frac{N-2}{2}\ veces\right)\right\},$$
$$\left\{(N-1)^{a_1},(N-3)^{a_3},(N-5)^{a_5},\dots,\left(\frac{N}{2}\ veces\right)\right\},$$
$$\{-N+2,-N+3,N-2;\dots;-6,-5,6;-4,-3,4;-2,-1,2\},$$
$$\{(N-1)^{a_2},(N-3)^{a_4},(N-5)^{a_6},\dots,(\frac{N}{2}\ veces)\}$$

Ecuación 26. Algoritmo para N columnas pares.

En caso de que una entrada sea negativa, en la agrupación en la que esté se cambiará en signo. Ejemplificando en un pretzel de 8 columnas, ($N-1^{|-5|}$) se resuelve como

-7, -7, -7, -7, -7

Dado lo complejo del algoritmo, vamos a ejemplificar con las dos trenzas trabajadas. La primera es P(2, 7, 5, 3), N=4.

$$\left\{-2{,}1{,}2;-4{,}3{,}4;-6{,}5{,}6;\ldots;\left(\frac{4-2}{2}\ veces\right)\right\},$$
$$\left\{(4-1)^2,(4-3)^5,(4-5)^0,\ldots,\left(\frac{4}{2}\ veces\right)\right\},$$
$$\{-4+2,-4+3{,}4-2;\ldots;-6,-5{,}6;-4,-3{,}4;-2,-1{,}2\},$$
$$\{(4-1)^7,(4-3)^3,(4-5)^0,\ldots,(\frac{4}{2}\ veces)\}$$

Si nos fijamos en la segunda progresión de agrupaciones, P(a_5) es 0 y su agrupación ya no aparece, algo que ya sabemos ya que tenemos que tener (4/2) agrupaciones en esa progresión. Lo mismo ocurre en la cuarta y última progresión de agrupaciones

$$\{-2{,}1{,}2,\}$$
$$\{3{,}3{,}1{,}1{,}1{,}1{,}1,\}$$
$$\{-2,-1{,}2,\}$$
$$\{3{,}3{,}3{,}3{,}3{,}3{,}3{,}1{,}1{,}1\}$$

El algoritmo se verifica con el método experimental:

1, 2, 3, 3, 1, 1, 1, 1, 1, -2, -1, 2, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, -2

Ahora verifiquemos P(6, 4, 6, 8, 6, 4) recordando que N=6:

$$\left\{-2{,}1{,}2;-4{,}3{,}4;-6{,}5{,}6;\ldots;\left(\frac{6-2}{2}\ veces\right)\right\},$$
$$\left\{(6-1)^6,(6-3)^6,(6-5)^6,\ldots,\left(\frac{6}{2}\ veces\right)\right\},$$
$$\{-6+2,-6+3{,}6-2;\ldots;-6,-5{,}6;-4,-3{,}4;-2,-1{,}2\},$$
$$\{(6-1)^4,(6-3)^8,(6-5)^4,\ldots,(\frac{6}{2}\ veces)\}$$

$$\{-2{,}1{,}2,-4{,}3{,}4,\}$$
$$\{5{,}5{,}5{,}5{,}5{,}5{,}3{,}3{,}3{,}3{,}3{,}3{,}1{,}1{,}1{,}1{,}1{,}1,\}$$
$$\{-4,-3{,}4,-2,-1{,}2,\}$$
$$\{5{,}5{,}5{,}5{,}3{,}3{,}3{,}3{,}3{,}3{,}3{,}3{,}1{,}1{,}1{,}1\}$$

Como podemos ver se ha vuelto a verificar, dando

-2,1,2,-4,3,4,5,5,5,5,5,5,3,3,3,3,3,3,1,1,1,1,1,1,-4,-3,4,-2,-1,2,5,5,5,5,3,3,3,3,3,3,3,3,1,1,1,1

Nudos pretzel con cinco entradas

Hemos conseguido un algoritmo para cualquier enlace pretzel con un número par de entradas. El lector ya se puede imaginar el siguiente paso. ¿Es posible un algoritmo para cualquier enlace pretzel con un número impar de entradas? Después de todo el trabajo necesario para 3 entradas, podemos prever que esto no va a ser trivial, así que esta vez me voy a limitar a los nudos, empezando por los de 5 entradas.

- P(impar, impar, impar, impar, par) y su espejo P(par, impar, impar, impar, impar).
- P(impar, impar, impar, par, impar) y su espejo P(impar, par, impar, impar, impar).
- P(impar, impar, par, impar, impar).
- P(impar, impar, impar, impar, impar).

Observamos que tenemos 4 posibilidades sin contar las espejo, mientras que si estudiásemos los enlaces de 5 entradas tendríamos 20 posibilidades sin contar las espejo.

Esta vez no voy a buscar los algoritmos, sino el método a desarrollar, ya que una vez lo tenga es posible que sirva para cualquier número impar de entradas.

P(impar, impar, impar, impar, par)

Empezamos con un nudo muy complejo, el P(9, 5, 7, 11, 12). El primer paso va a ser estudiar los círculos de Seifert, el ingrediente principal que nos dice qué camino deberemos tomar.

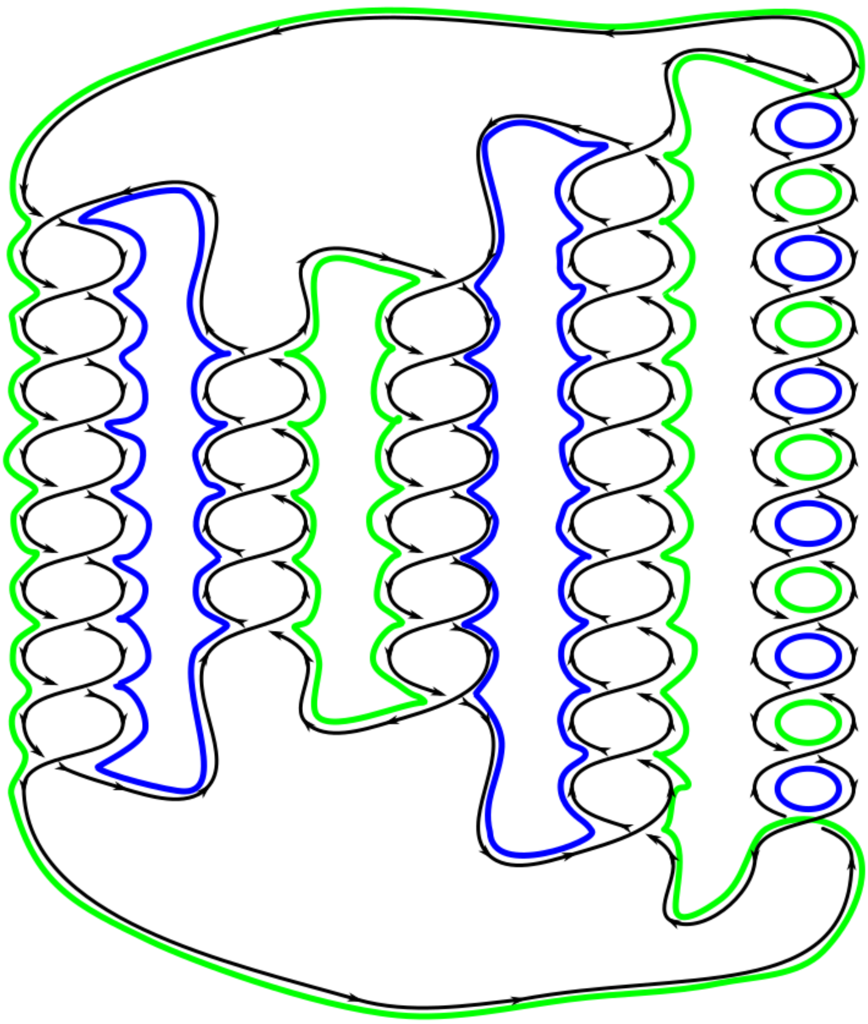


Ilustración 201. P(9, 5, 7, 11, 12) y sus círculos de Seifert.

Este es el momento importante. Tenemos que reflexionar qué paso vamos a hacer, qué herramientas usar, porque desde luego no es una buena idea usar sólo reducciones normales y corrientes.

Sabemos que la columna de círculos de la derecha la podemos arreglar con movimientos básicos, y como está en el exterior podríamos hacer movimientos de polo sur para que engloben al resto del nudo. ¿Pero qué hacemos con los círculos entre columnas? Cuando teníamos 3 columnas nos quedaba un círculo entre columnas dentro de otro, tal y como está el verde entre la columna segunda y tercera dentro del verde grande. Pero nos han surgido dos círculos azules a los lados que nos impiden que todos los círculos sean concéntricos. Pues bien, tras varios intentos y una profunda reflexión he llegado a la conclusión de que este problema se puede arreglar con desplazamientos de columnas.

Como el nudo P(9, 5, 7, 11, 12) es muy complejo y nos ocupa toda la pantalla, voy a ejemplificar con el P(3, 3, 3, 3, 4), ya que 3 es el menor número impar que no es 1, y 4 es el menor par que no es 2.

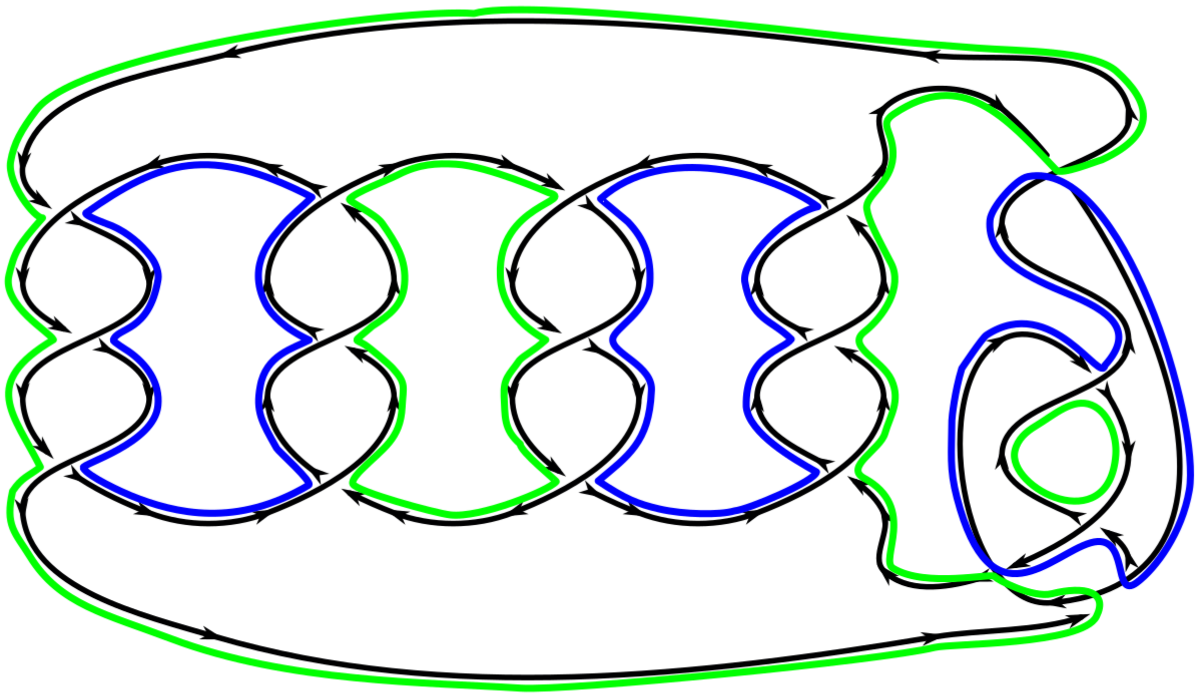


Ilustración 202. P(3, 3, 3, 3, 4) tras un movimiento básico en la columna derecha.

Ahora probemos el desplazar una columna a la izquierda.

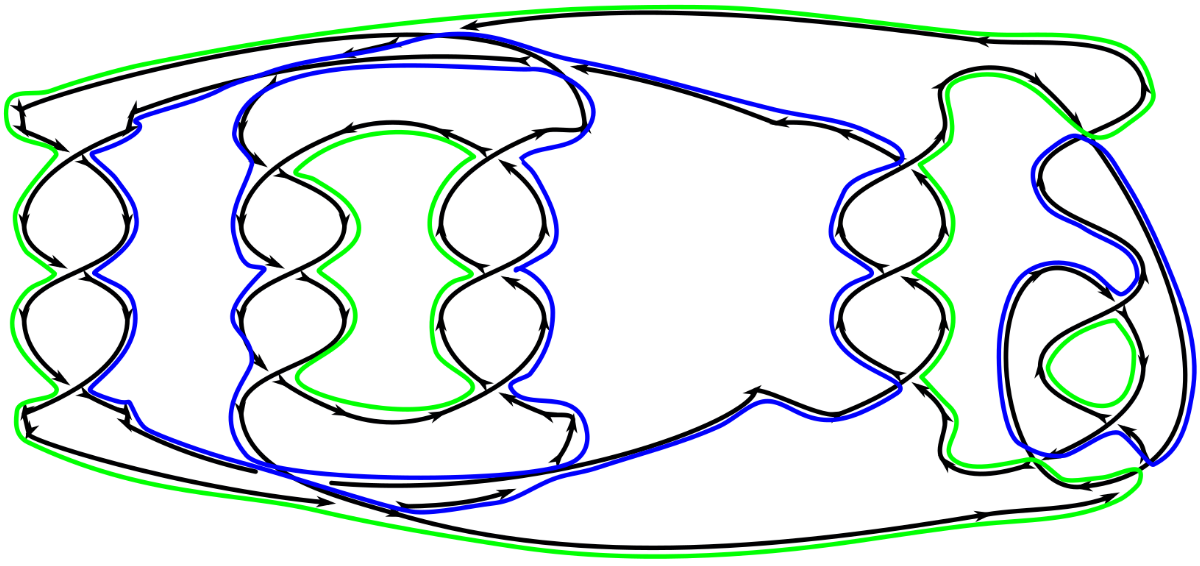


Ilustración 203. Un movimiento de columnas.

Como podemos observar, nuestra hipótesis ha estado muy acertada, y todos los círculos son compatibles. Sólo nos queda lo que ya sabemos que es una nimiedad, y es que el círculo azul y el círculo verde de la derecha engloben a los demás, mediante 2 movimientos de polo sur. En ese momento tendremos todos los círculos compatibles y concéntricos, es decir, el diagrama de nuestra trenza.

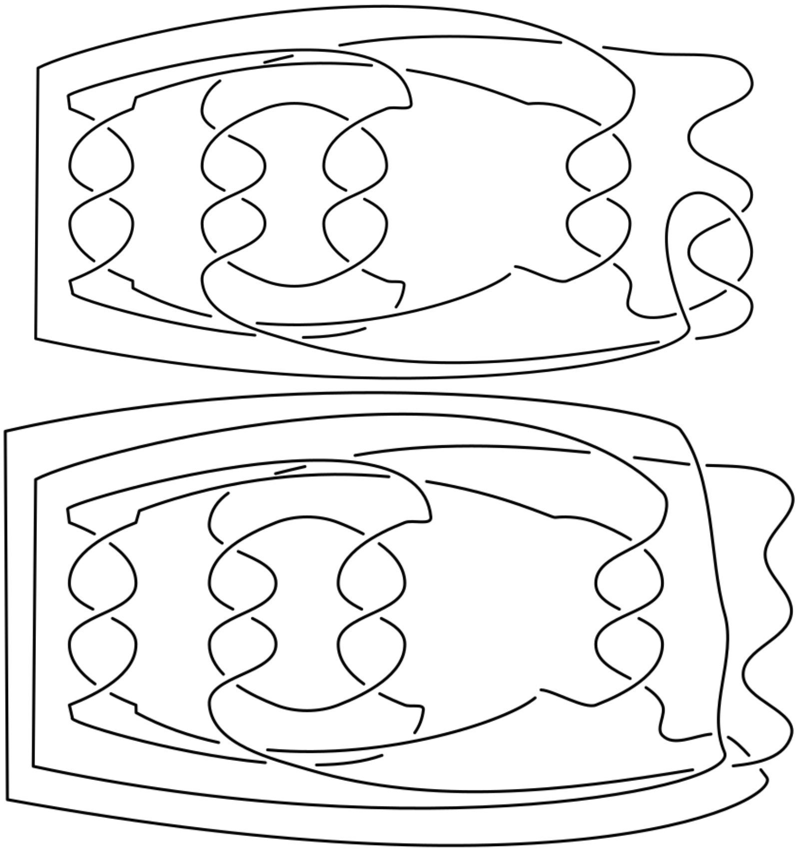


Ilustración 204. Movimientos de polo sur.

Si orientamos el nudo ya observamos la trenza, ya que las flechas apuntan en todo momento en sentido anti horario.

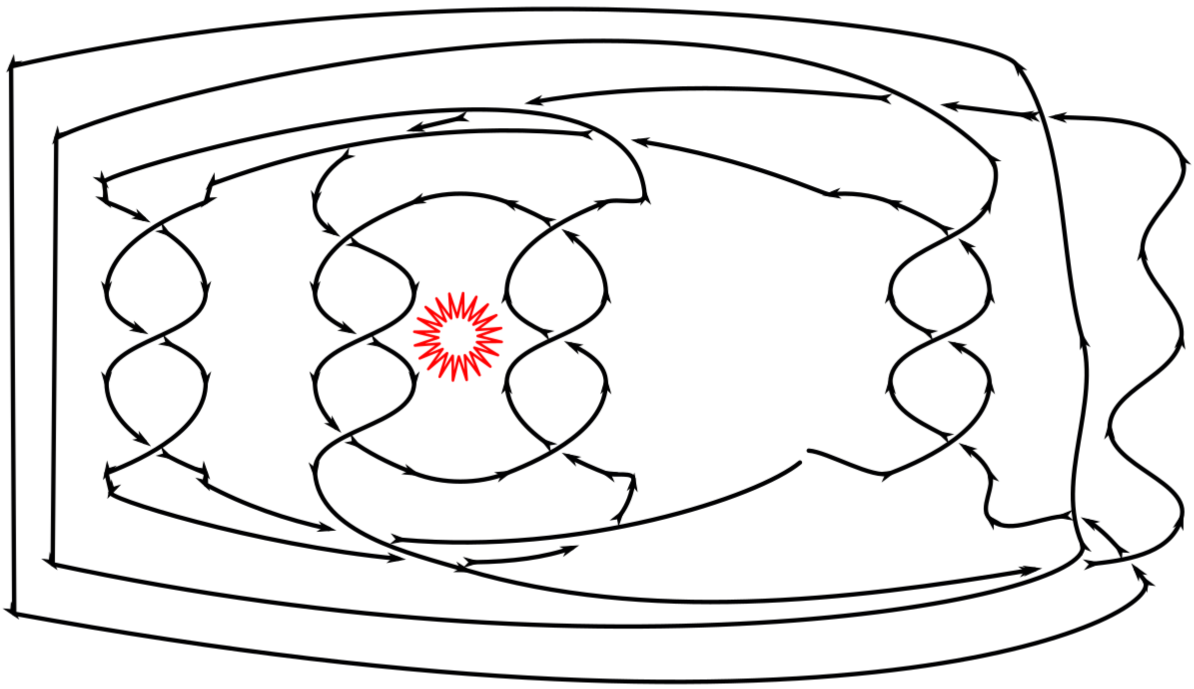


Ilustración 205. Trenza de P(3, 3, 3, 3, 4) orientada.

Como el lector puede apreciar, resulta que sólo tiene verdadera influencia en el desarrollo el tamaño del número par, ya que implica movimientos básicos y de polo sur.

P(impar, impar, impar, par, impar)

Le llega al turno a P(3, 3, 3, 4, 3). Como siempre, empiezo estudiando los círculos de Seifert.

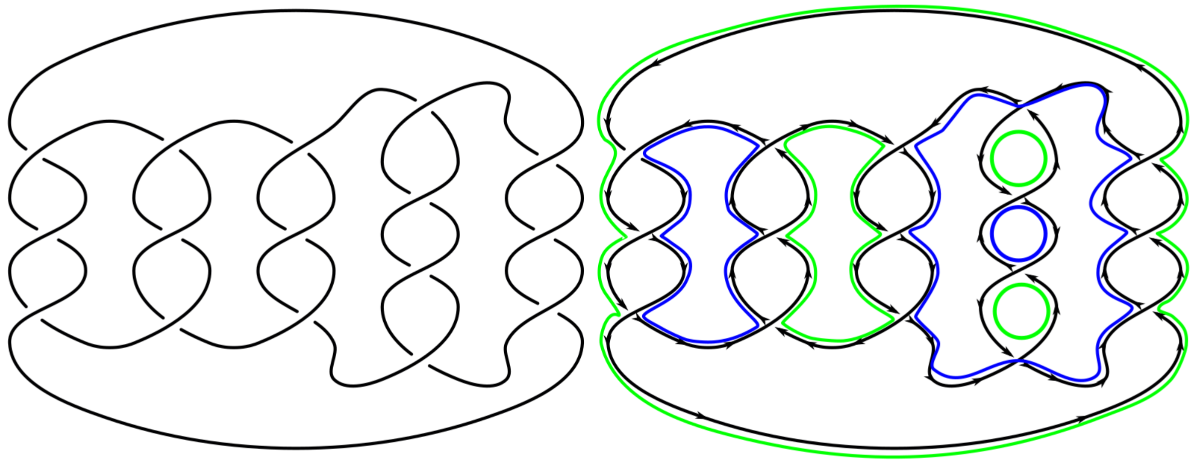


Ilustración 206. Nudo P(3, 3, 3, 4, 3) orientado y con sus círculos de Seifert.

Ahora nos encontramos en una situación algo difícil. Tenemos los ingredientes necesarios para hacer movimientos básicos o para desplazamientos de columnas, pero ¿cómo? Si hacemos básicos en la penúltima columna tendremos 3 círculos concéntricos, contando el azul que ya engloba a dicha columna, pero no podremos sacarlos con movimientos de polo sur, ya que estarían atrapados entre la 3ª y 5ª columna. Se me ocurre que quizá pueda hacer un movimiento de columnas llevándome una a la derecha, de tal manera que la 4ª columna se convierta en la 3ª y se nos quede en el centro. Probemos.

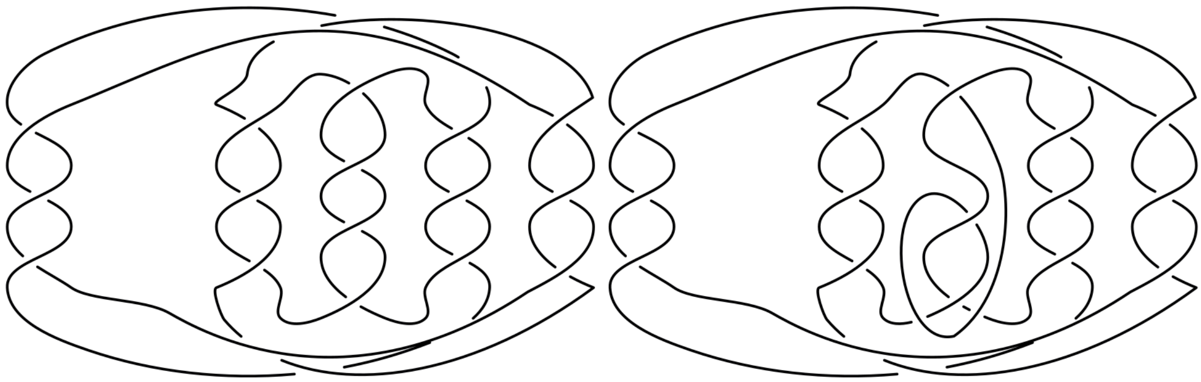


Ilustración 207. Movimiento de la segunda columna a la derecha, y movimiento básico en la columna central.

Parece que hemos vuelto a acertar, ya que una vez tenemos la columna de los círculos (la columna par) en el centro, podemos hacer movimientos básicos y se nos solucionan los círculos de Seifert. Parece un camino obvio, pero he tenido que hacer varios intentos probando con diferentes columnas y con reducciones normales hasta descubrir que este método es el adecuado.

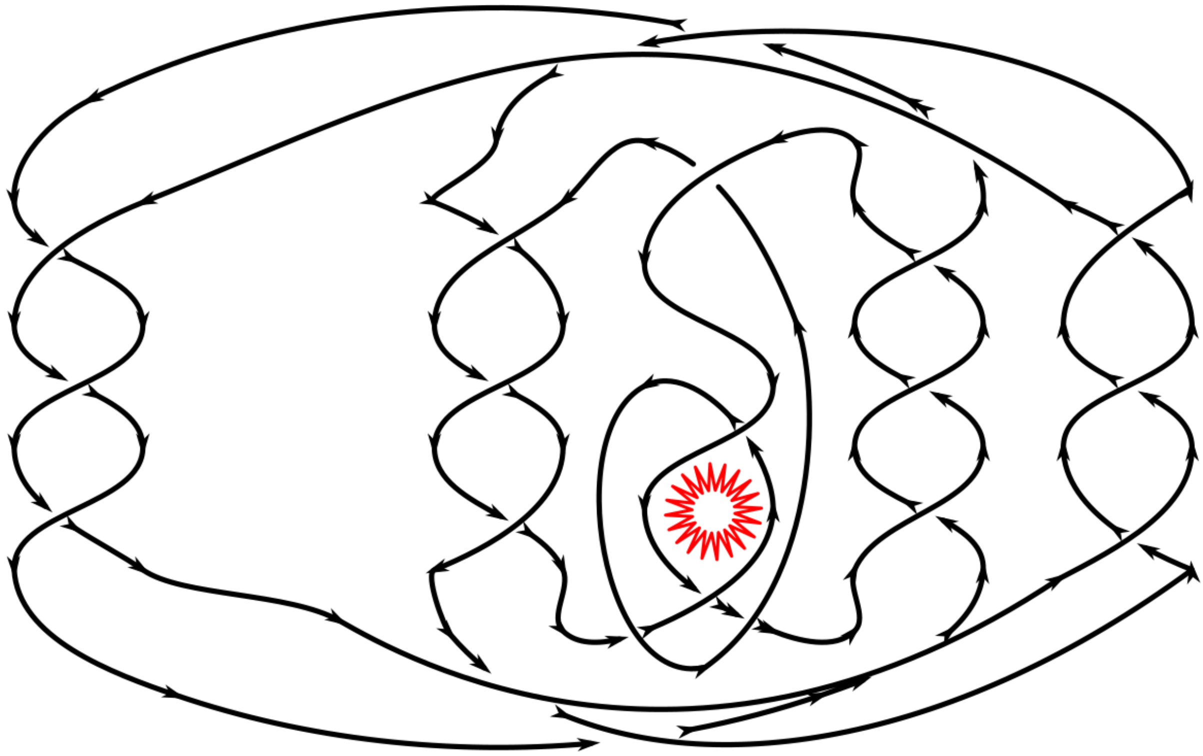


Ilustración 208. Trenza de P(3, 3, 3, 4, 3) orientada.

P(impar, impar, par, impar, impar)

Este en un principio puede parecer esperanzador, pues la columna de círculos estará en el centro. Pues resulta que no, y es el que más me ha costado; después del caso de todos impares, por supuesto.

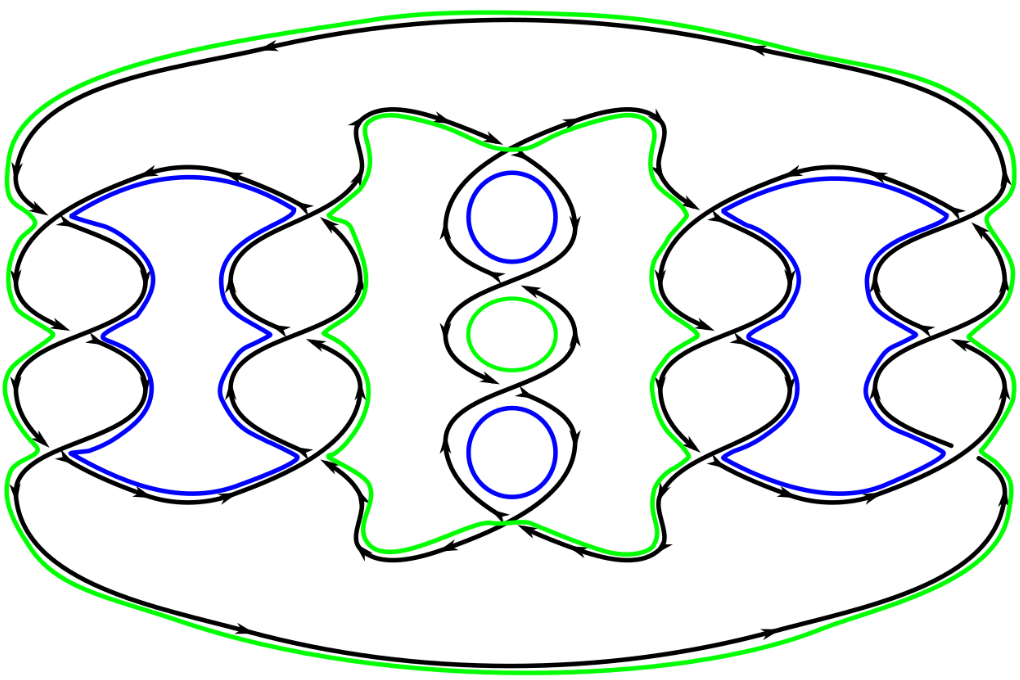


Ilustración 209. Nudo P(3, 3, 4, 3, 3) orientado y con sus círculos de Seifert.

He tenido varios intentos infructuosos. Primero haciendo movimientos básicos con columnas, pero siempre surgía el problema de que la columna de círculos deja de estar en el centro. Ahora llega el momento de reflexión sobre los métodos pasados. Siempre que hay círculos de Seifert en los huecos entre columnas hay que hacer movimientos de columnas para arreglarlos de forma sencilla. Y cuando hay columnas de círculos hay que hacer movimientos básicos. La clave está en que esta columna siempre ha estado o en el centro o en un extremo, y si ahora tenemos que hacer movimientos de columnas difícilmente seguirá en el centro. Así que una solución podría ser un movimiento de columnas que nos lleve esta columna fuera, para que cuando hagamos los movimientos básicos con sus círculos luego hagamos movimientos de polo sur que engloben al resto del nudo.

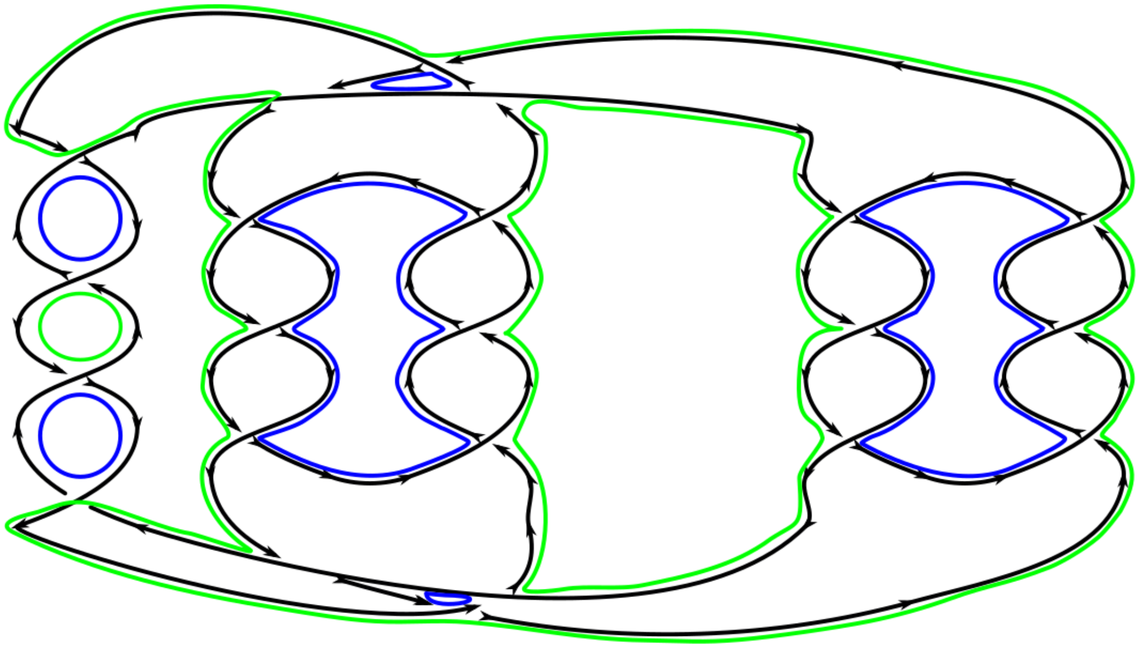


Ilustración 210. P(3, 3, 4, 3, 3) tras un movimiento de columnas llevando la central a la izquierda.

Analicemos bien lo que tenemos. Por una parte, una columna de círculos a la izquierda. Podemos hacerle movimientos básicos y luego de polo sur para que englobe a lo demás. Por otra parte, círculos de Seifert en huecos entre columnas, lo que podemos solucionar con otro movimiento de columnas.

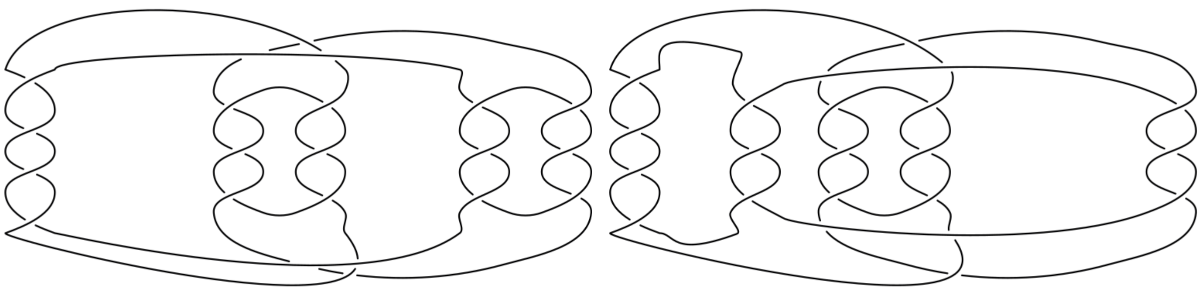


Ilustración 211. Movimiento de la penúltima columna hasta la segunda posición.

Ahora sólo tendríamos pendiente los movimientos básicos en la que ahora es la primera columna y sus respectivos movimientos de polo sur. Primero lo adaptaré comprimiéndolo para hacerlo más manejable

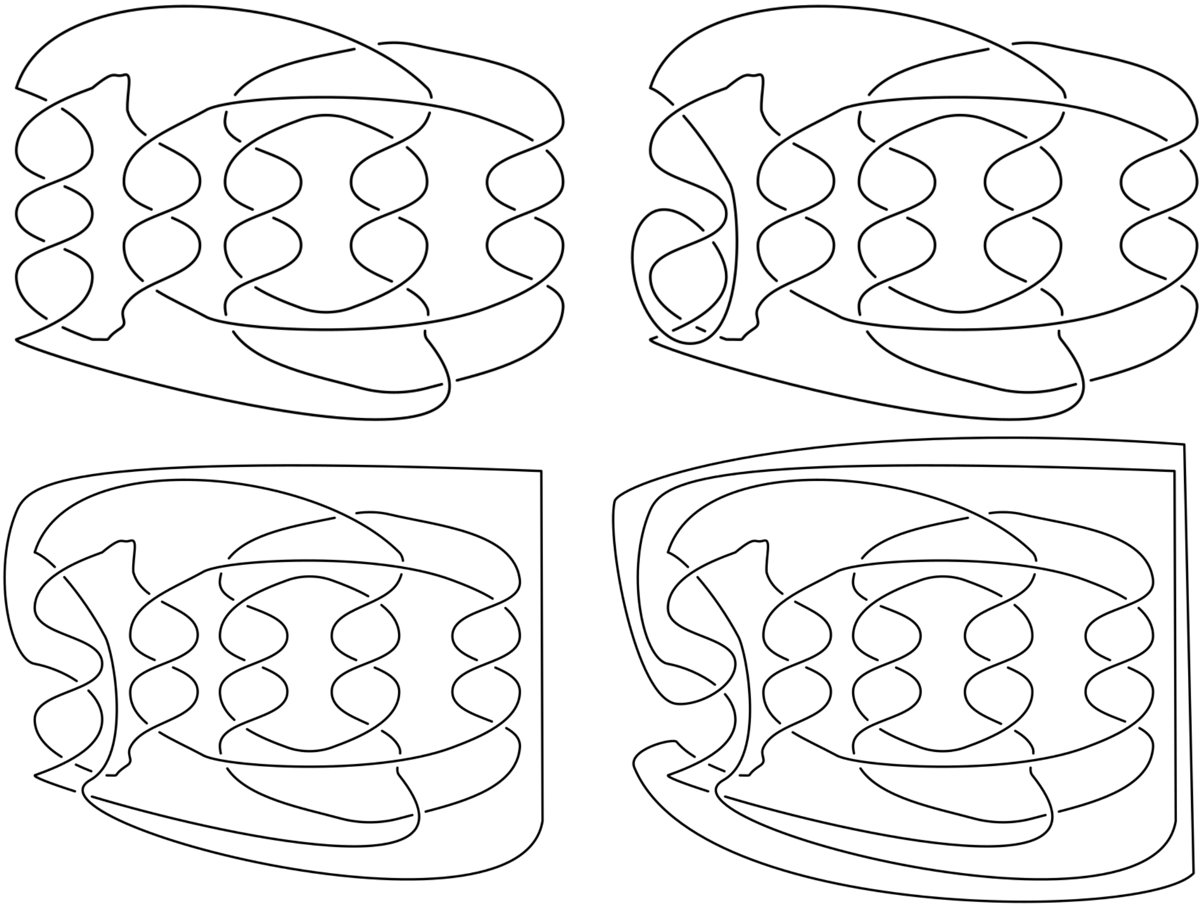


Ilustración 212. Movimiento básico en la columna izquierda y dos movimientos de polo sur.

Si orientamos nuestro nudo observamos que ya hemos logrado la trenza.

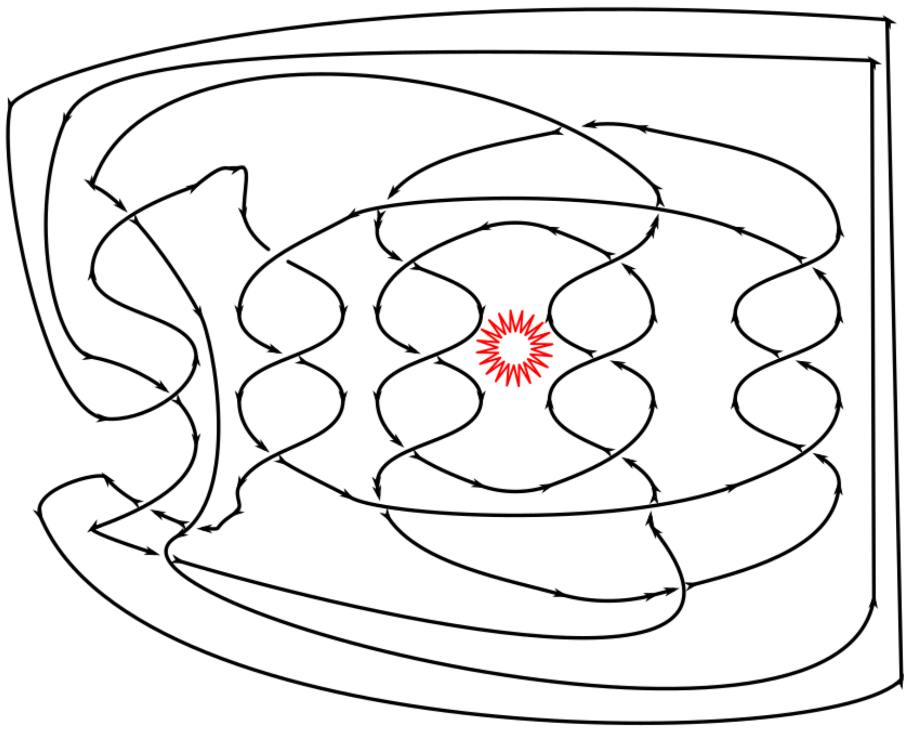


Ilustración 213. Trenza de P(3, 3, 4, 3, 3) orientada.

P(impar, impar, impar, impar, impar)

Parece intuitivo que el método para Nudos P(par, impar, impar) podría funcionar en este caso de 5 columnas. Vamos a ahorrarnos los casos con una sola columna variable y tomemos desde el principio un pretzel significativo. Presento al lector el nudo P(9, 5, 7, 11,13):

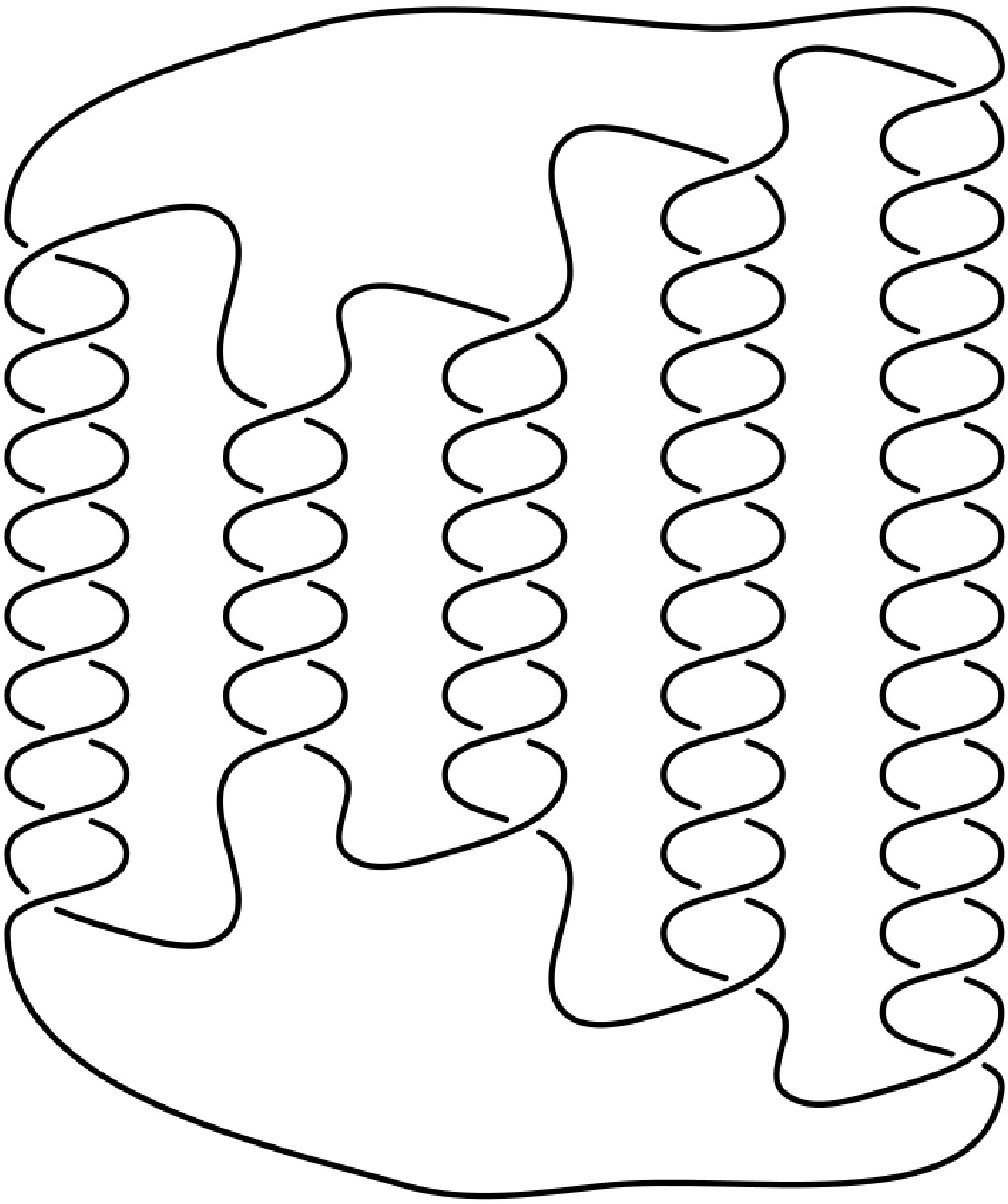


Ilustración 214. P(9, 5, 7, 11, 13).

Primero vamos a analizar los círculos de Seifert para ver en qué situación nos encontramos y saber si podemos usar nuestra hipótesis.

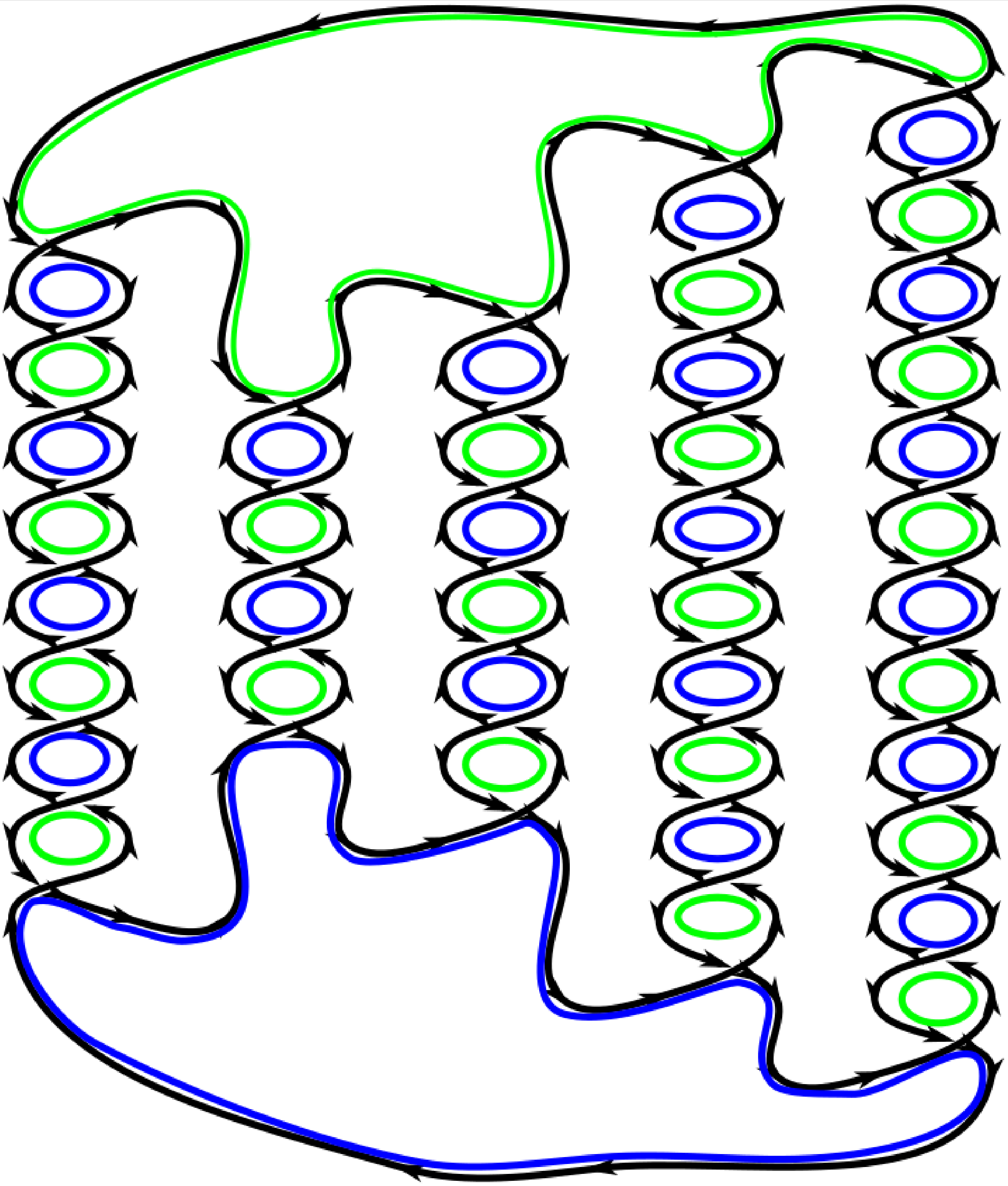


Ilustración 215. P(9, 5, 7, 11, 13) con sus círculos de Seifert.

Como podemos comprobar, la situación es la misma que teníamos con P(Impar, Impar, Impar), por lo que la resolución se haría haciendo todos los movimientos básicos de cada columna, el movimiento de polo sur y finalmente las reducciones múltiples.

Es decir, para cualquier pretzel con un número impar de entradas, si todas las entradas son impares, podremos seguir este método.

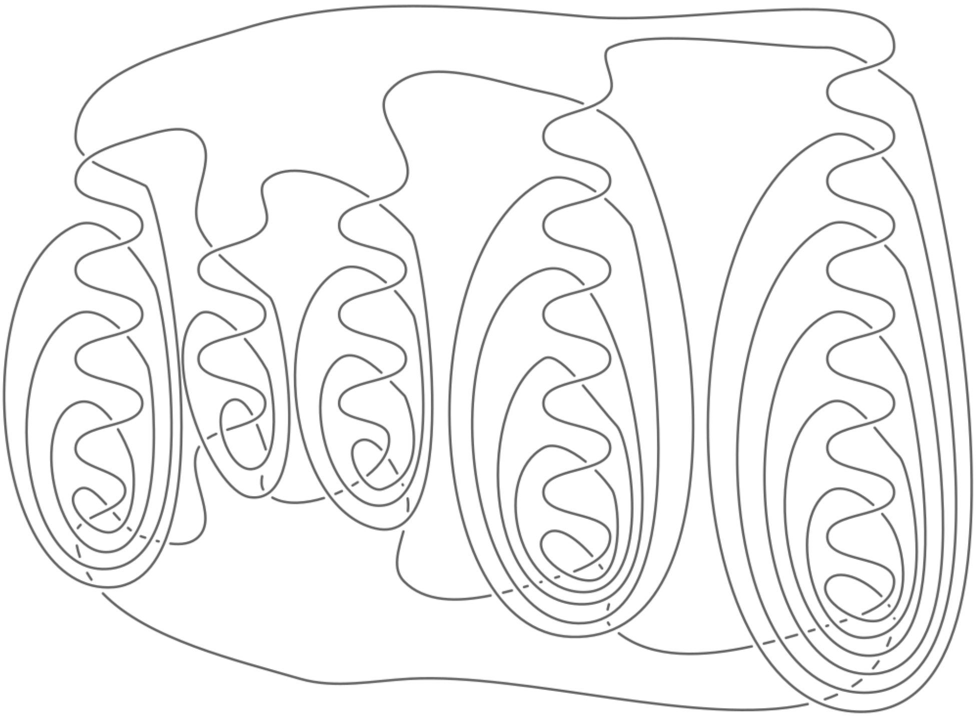


Ilustración 216. Movimientos básicos en cada columna.

Vemos que por ahora nuestra predicción se cumple, y se forma una especie de ovillo de lana por cada columna. Igual que entonces, vamos a mover un poco las cuerdas sin añadir o quitar cruces para facilitar las reducciones múltiples.

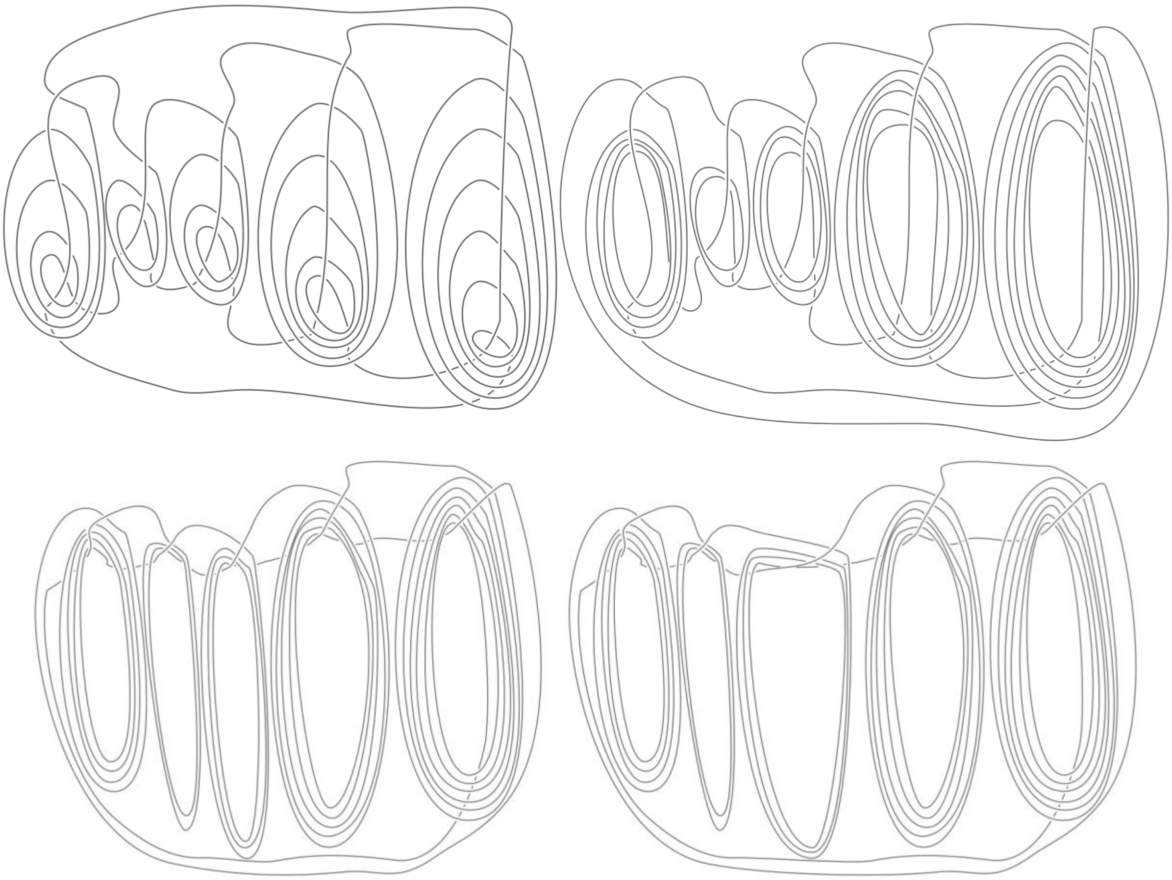


Ilustración 217. Isotopías para simplificar el nudo.

Ahora llega el momento complicado, los movimientos de reducción múltiple. Lo que buscamos es colocar nuestra estrella en el centro y que esté envuelta en los ovillos de lana, que son los grupos de cuerdas de cada columna.

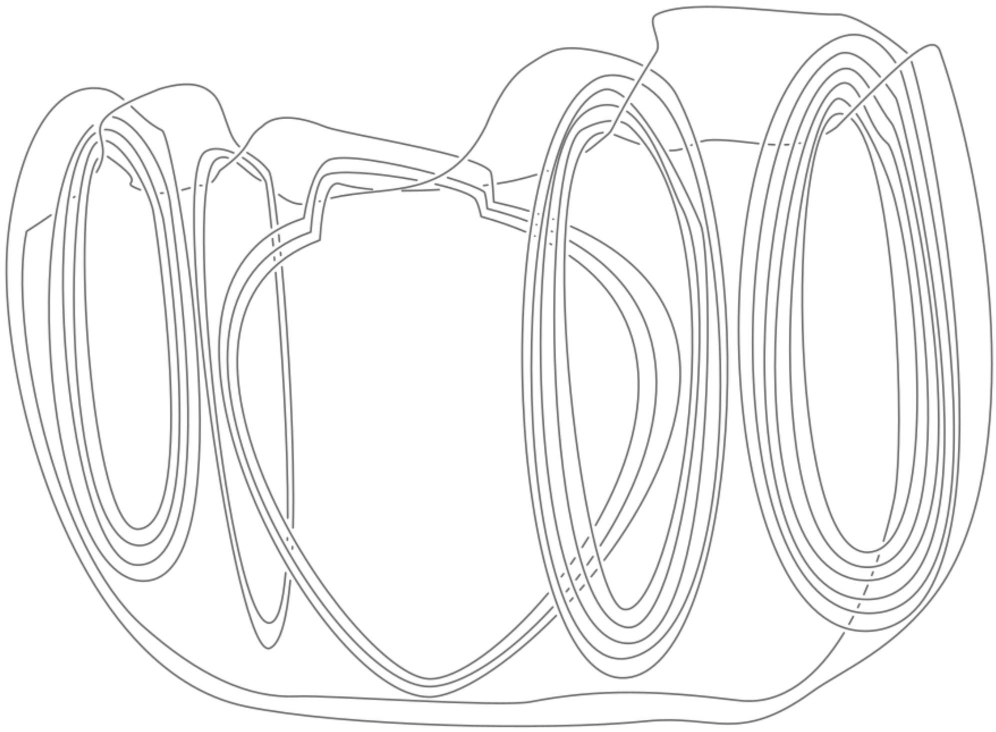


Ilustración 218. Reducción de las cuerdas centrales bajo las cuerdas a su derecha y sobre las cuerdas a su izquierda.

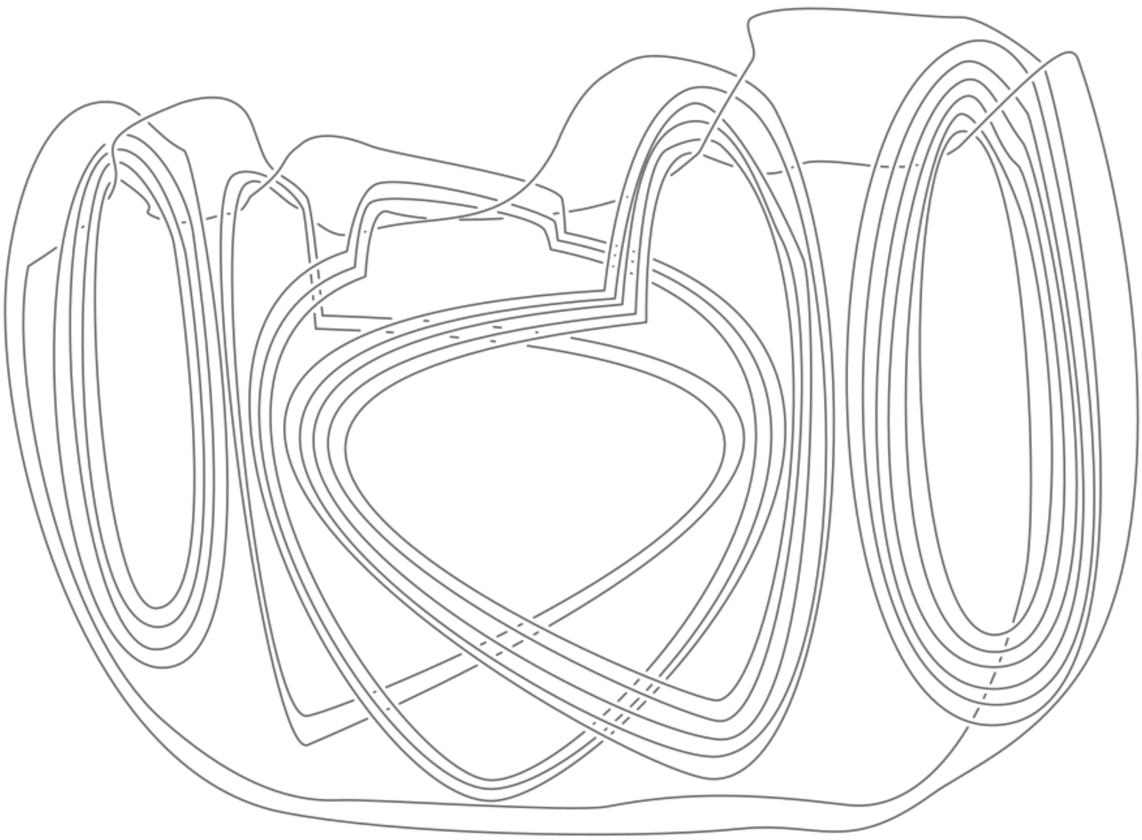


Ilustración 219. Reducciones cuerdas de la 4º columna sobre las de la 3º y la 2º, y de la 2º bajo la 3º y la 4º.

Y ahora llega el momento final, cuando reducimos las cuerdas de la columna derecha sobre todas las que sean necesarias hasta llegar al centro, y las de la columna izquierda bajo todas las que haga falta hasta llegar al centro. Le presento al lector la aterradora ilustración de la clausura de la trenza de P(9, 5, 7, 11, 13).

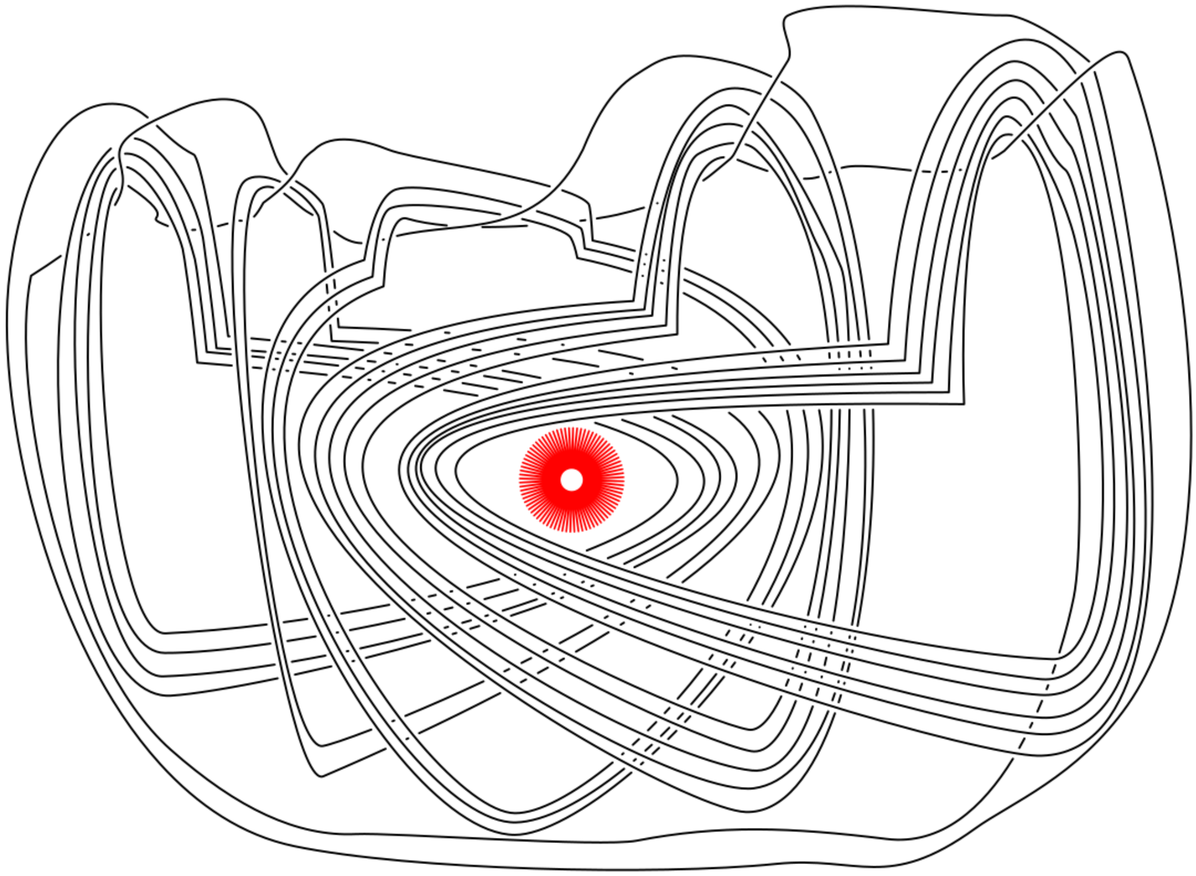
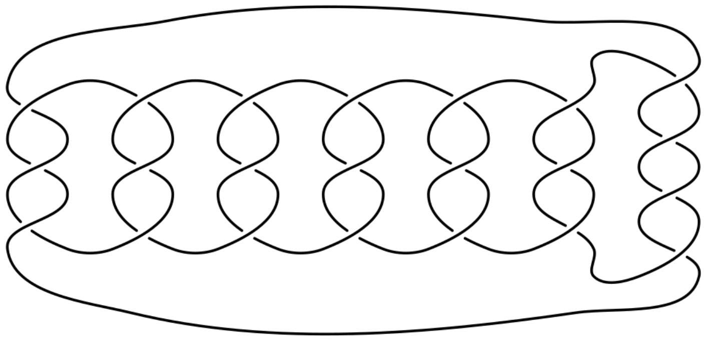


Ilustración 220. Clausura de la trenza de P(9, 5, 7, 11, 13). 21 cuerdas y casi 200 cruces.

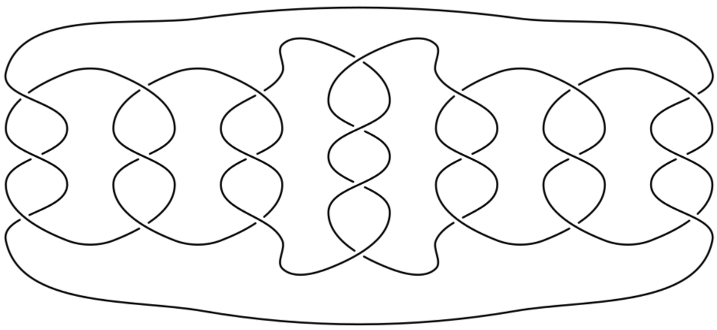
Nudos pretzel con n entradas, n impar y mayor que tres

En el caso de que tener un número impar de entradas y que todas sean impares ya hemos descubierto la estrategia a seguir en la subsección P(impar, impar, impar, impar, impar) de la página 151, así que nos quedan los nudos con una entrada par. Tenemos tres estrategias a seguir:

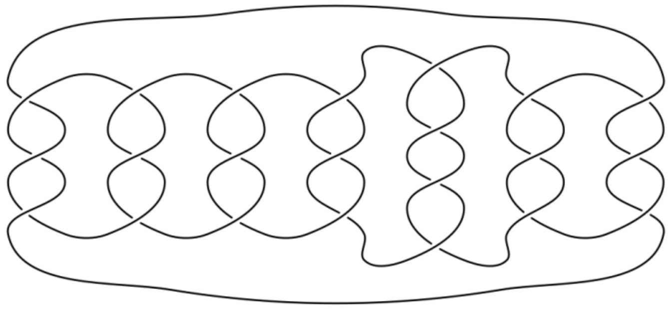
- Si tenemos la entrada par en un extremo: haremos tantos movimientos básicos como sean necesarios en la columna de la entrada par, y luego sus correspondientes movimientos de polo sur. Seguidamente solucionaremos la incompatibilidad de las otras columnas con movimientos de desplazamientos de columnas.



- Si tenemos la entrada par en el centro: haremos movimientos de columnas. La mitad de estos movimientos desplazarán columnas de la izquierda a la derecha y la otra mitad desplazarán sus columnas simétricas de la derecha a la izquierda. El objetivo es que la columna par siga en el centro. Terminamos con los movimientos básicos necesarios según el valor de la entrada par.



- Si la entrada par no está ni en el centro ni en un extremo: haremos movimientos de columnas de tal manera que la columna par acabe en el centro.



Dicho esto, estudiemos cada uno de estos 3 casos.

Entrada par en un extremo

Como podemos observar en los círculos de Seifert, tenemos una columna de ellos en la columna par, y 5 en los huecos entre columnas.

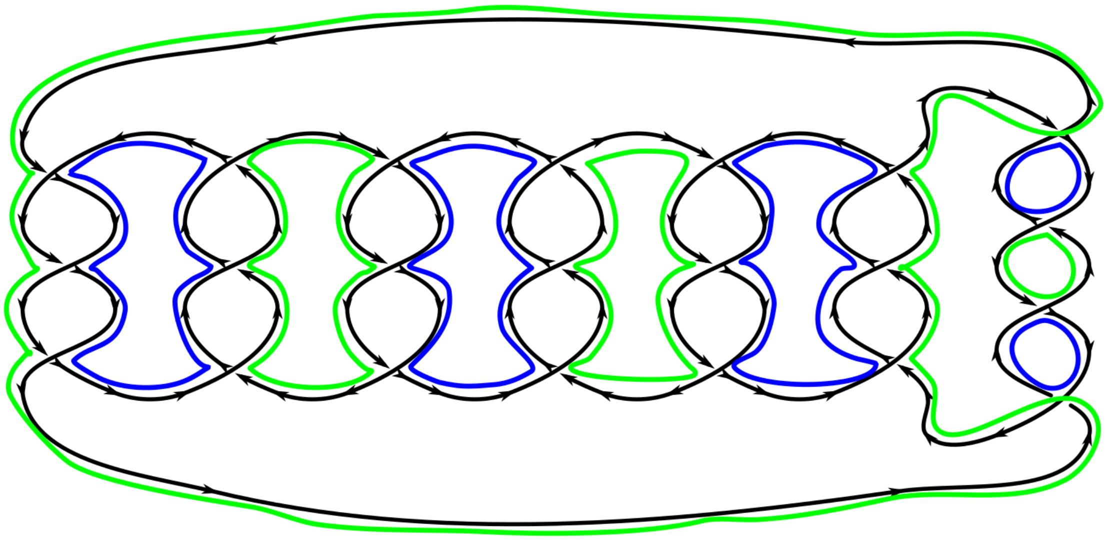


Ilustración 221. P(3, 3, 3, 3, 3, 3, 4) orientado con sus círculos de Seifert.

Recordamos la estrategia para este caso: primero vamos a hacer desplazamientos de columnas, en este caso 2.

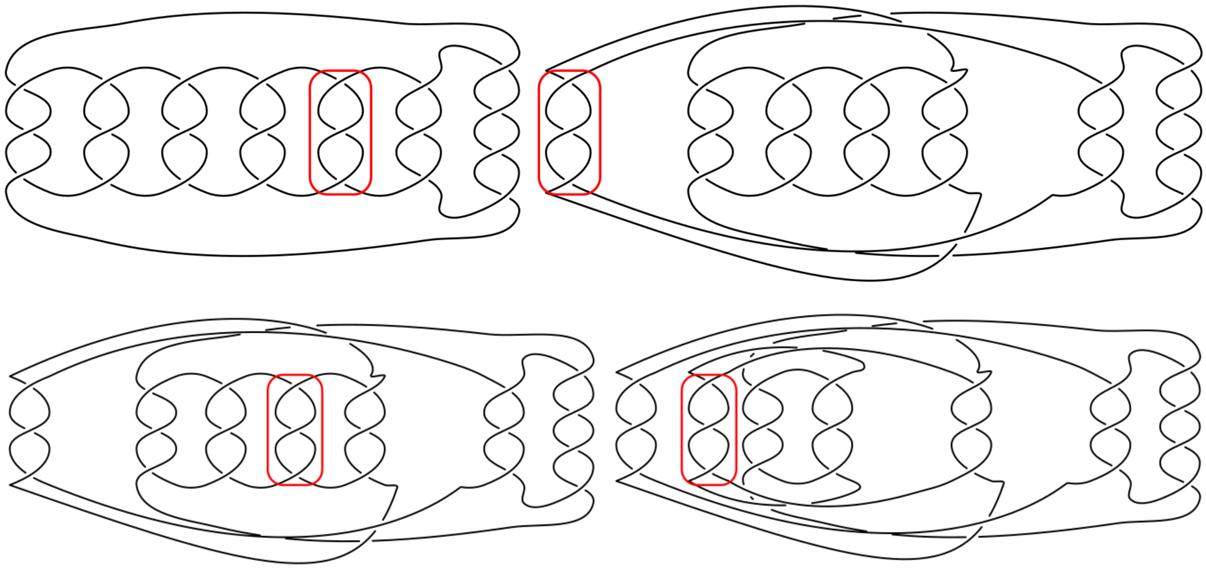


Ilustración 222. Dos desplazamientos de columna.

Si ahora comprimo el nudo para hacerlo más manejable y estudio sus círculos de Seifert observo que todos los círculos son compatibles y concéntricos menos los de la columna de entrada par.

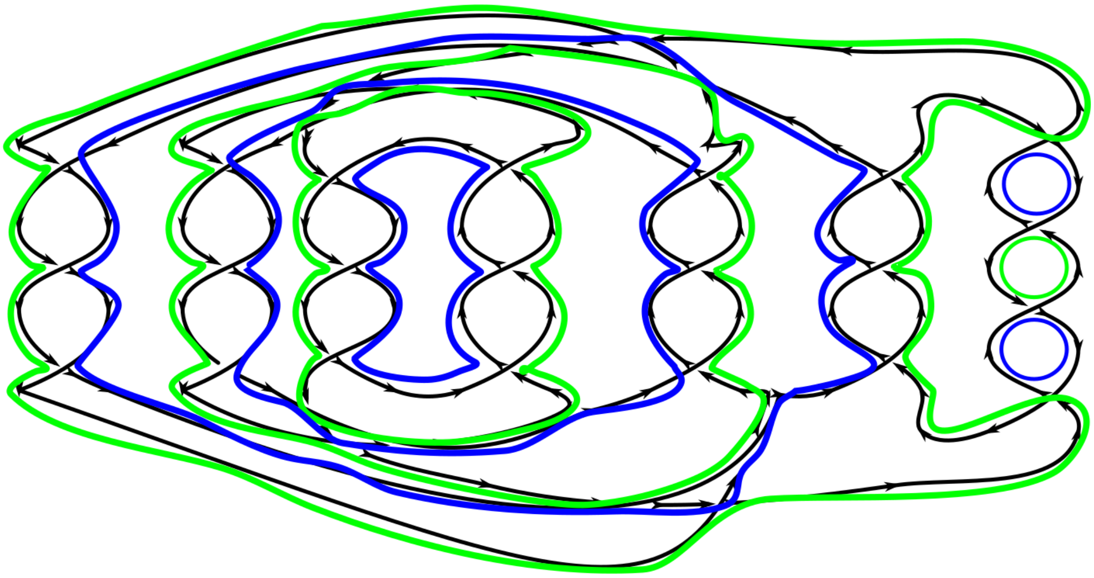


Ilustración 223. Nudo P(3, 3, 3, 3, 3, 3, 4) tras dos desplazamientos de columnas, orientado y con sus círculos de Seifert.

Para arreglar los círculos de la columna par se hace un movimiento básico y luego dos movimientos de polo sur.

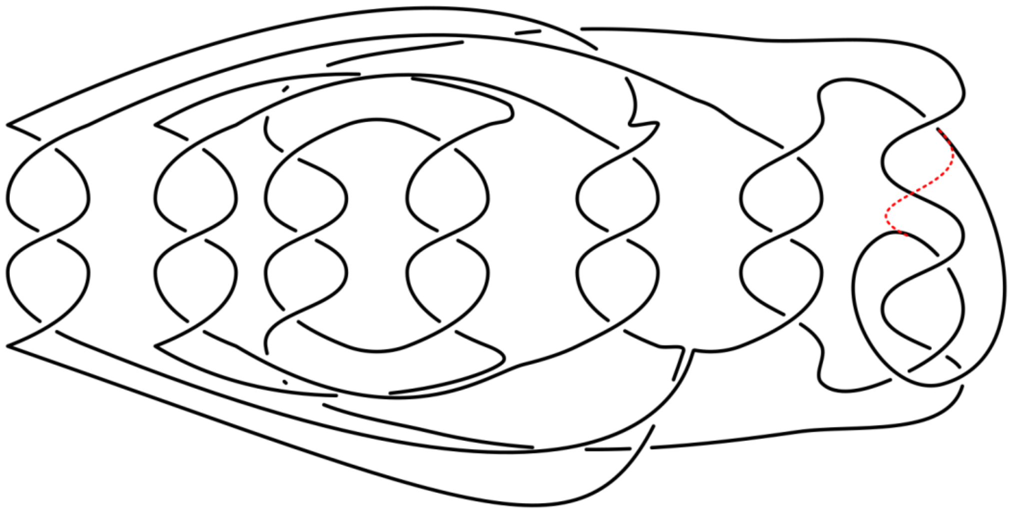


Ilustración 224. Movimiento básico en la columna derecha.

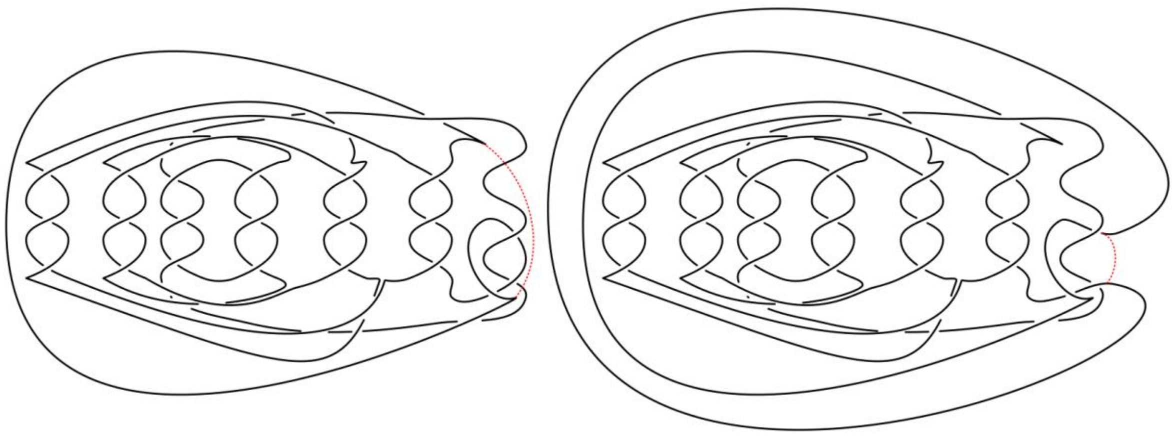


Ilustración 225. Dos movimientos de polo sur.

Si orientamos el nudo observamos que tenemos la trenza, pues todas las cuerdas están orientadas en sentido anti horario.

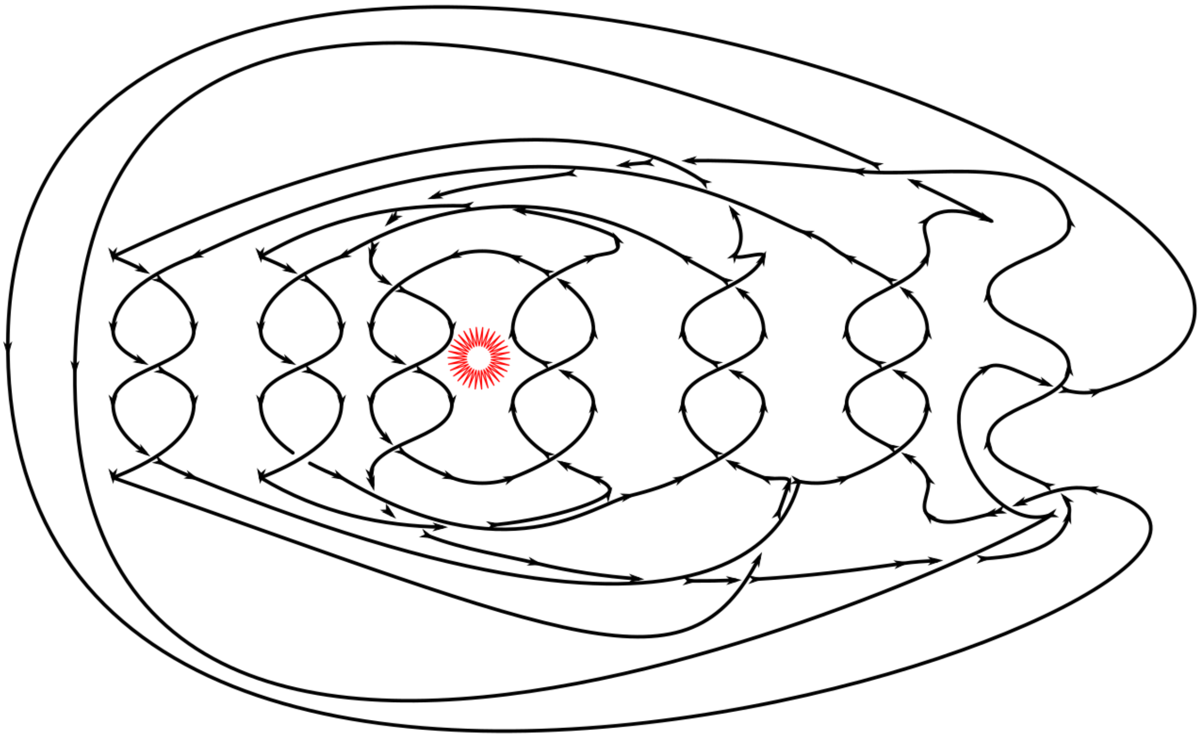


Ilustración 226. Nudo P(3, 3, 3, 3, 3, 3, 4) como clausura de una trenza orientada.

Siempre tendremos que hacer $N/2-1$ desplazamientos, redondeando al entero inferior. Por ejemplo, en este caso de 7 columnas hacemos $7/2-1=2{,}5\rightarrow 2$ desplazamientos. Con 9 tendríamos 3, con 11 tendríamos 4, etc.

Entrada par en el centro

Esta vez observamos que tenemos la columna par con los círculos en el centro, lo que implica que no podremos hacer movimientos de polo sur.

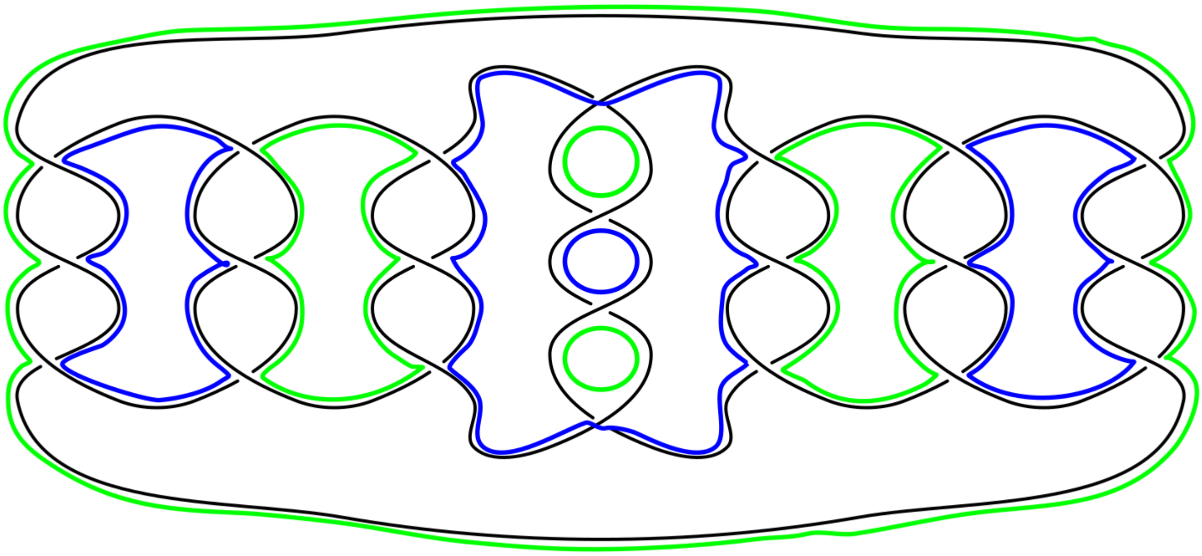


Ilustración 227. Nudo P(3, 3, 3, 4, 3, 3, 3) orientado y con sus círculos de Seifert.

La clave va a estar en dejarla en el centro, y así cuando hagamos el movimiento básico ya tendremos tres círculos concéntricos. Así que primero tenemos que buscar que los círculos en los huecos entre columnas sean concéntricos y en su interior esté la columna central. Para esto vamos a hacer desplazamientos de columnas, una izquierda a la derecha y una derecha a la izquierda, de forma que la entrada par se quede en el centro.

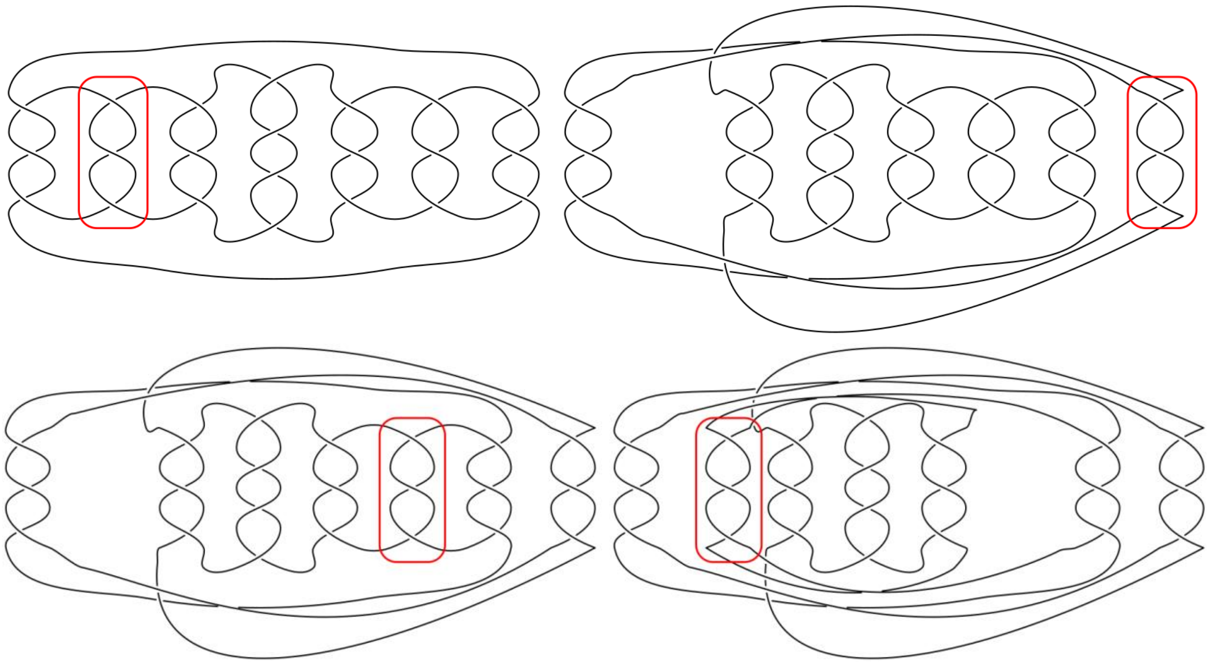


Ilustración 228. Desplazamiento de la segunda columna (desde la izquierda) a la derecha, desplazamiento de la segunda columna (desde la derecha) a la izquierda

Ahora sólo nos queda un simple movimiento básico en la columna central.

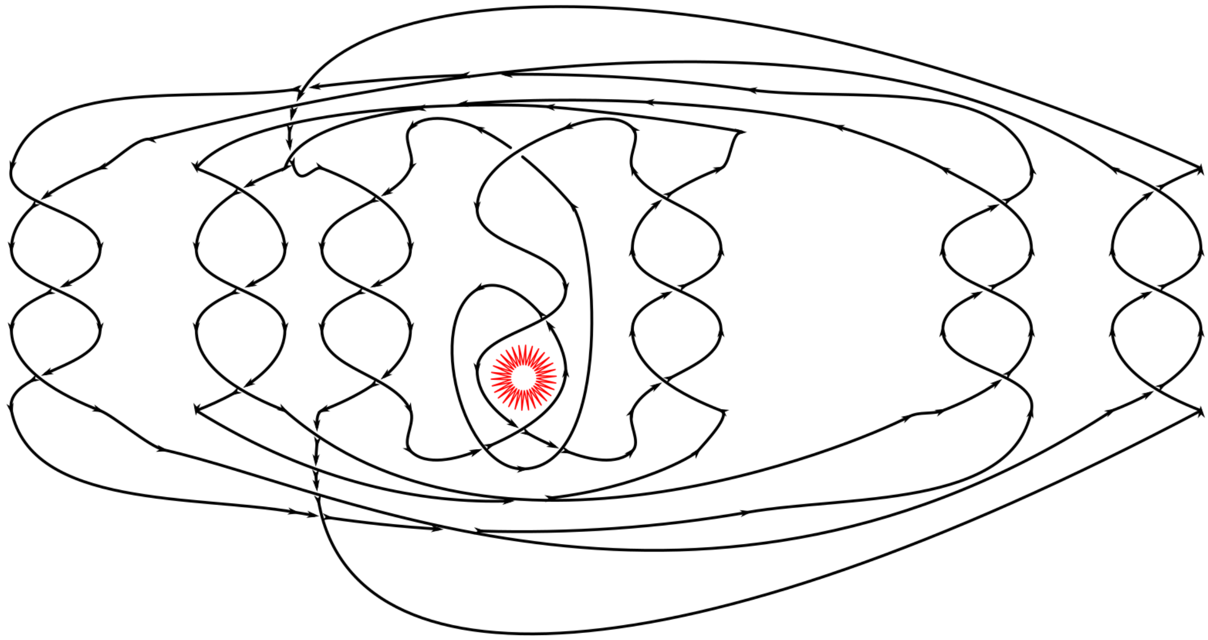


Ilustración 229. P(3, 3, 3, 4, 3, 3, 3) orientado.

Recapitulando la estrategia en el caso de tener un número impar de entradas y que la central sea la única entrada par: vamos a desplazar columnas de un lado al otro, y realizando desde el otro lado la misma cantidad de movimientos para que el nudo siempre tenga la columna par en el centro.

Habrá que hacer (N-1)/4 pares de desplazamientos de columnas. En este caso, (9-1)/2 = 4 pares. En las ocasiones en las que no de un entero, el entero inferior. Por ejemplo, con 9 columnas tendremos (7-1)/4 = 1,5 → 1 par de desplazamientos.

Hay que recordar que un par de desplazamientos significa desplazar uno de la izquierda a la derecha y uno de la derecha a la izquierda.

Entrada par en un lugar que no es ni el centro ni un extremo

Nuestra estrategia en este caso es realizar tantos desplazamientos de columnas como sean necesarios para que nuestra columna de entrada par acabe en el centro. Si tenemos la entrada par a la derecha del centro, moveremos columnas de su izquierda a la derecha, y si es al contrario viceversa.

Una vez que tengamos la entrada par en el centro sólo nos quedará por hacer movimientos básicos.

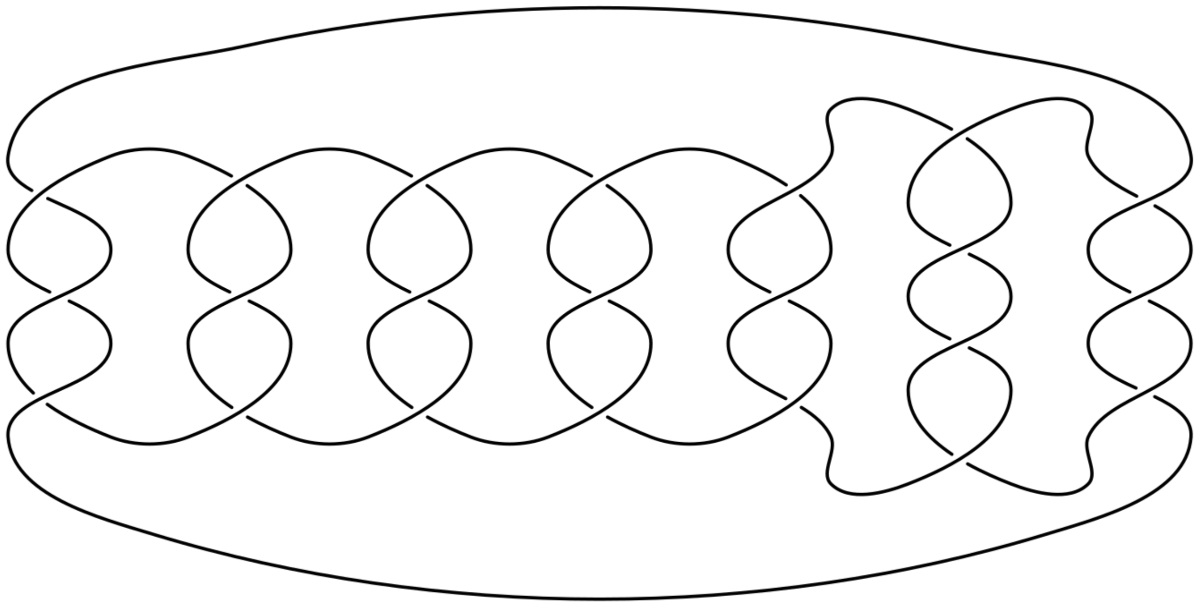


Ilustración 230. P(3, 3, 3, 3, 3, 4, 3).

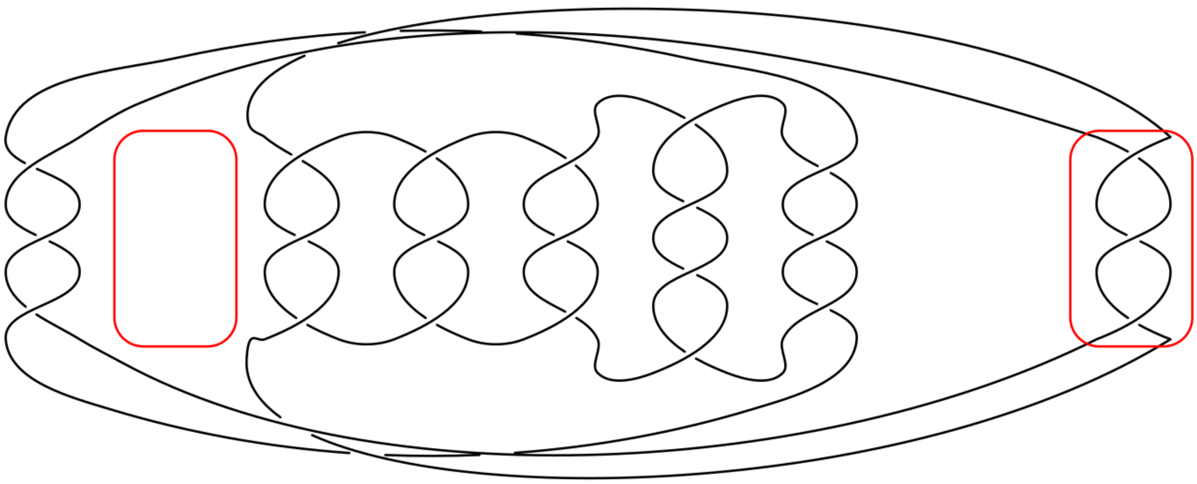


Ilustración 231. Primer desplazamiento de columna.

Como podemos ver, nuestra columna par está en la posición 5ª de 7, por lo que con otro desplazamiento más ya estaría.

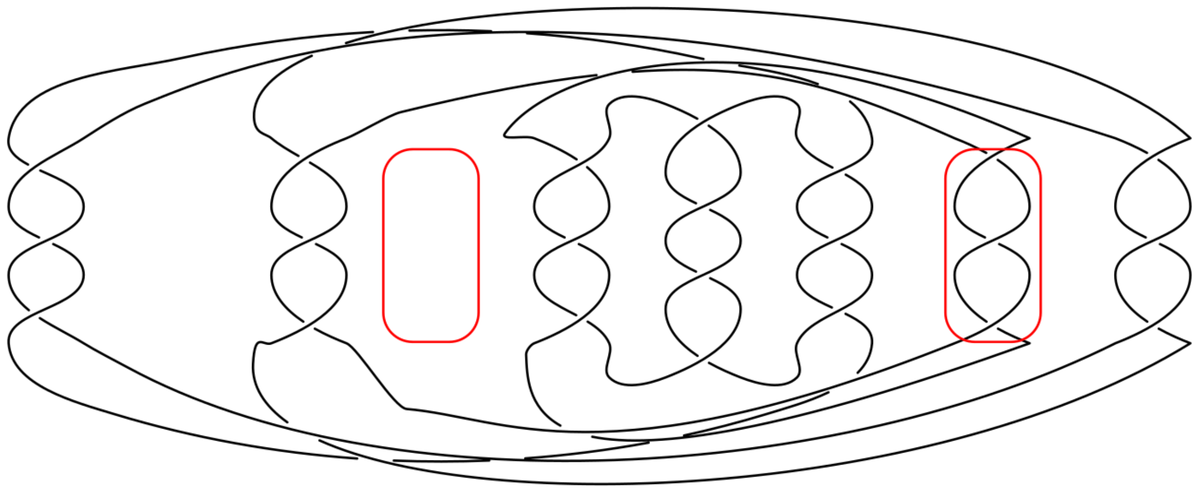


Ilustración 232. Segundo desplazamiento de columna, la que antes era la 4ª a la penúltima posición.

Ya tenemos la columna par en el centro, es la cuarta columna. Hacemos un movimiento básico y tenemos la clausura de la trenza.

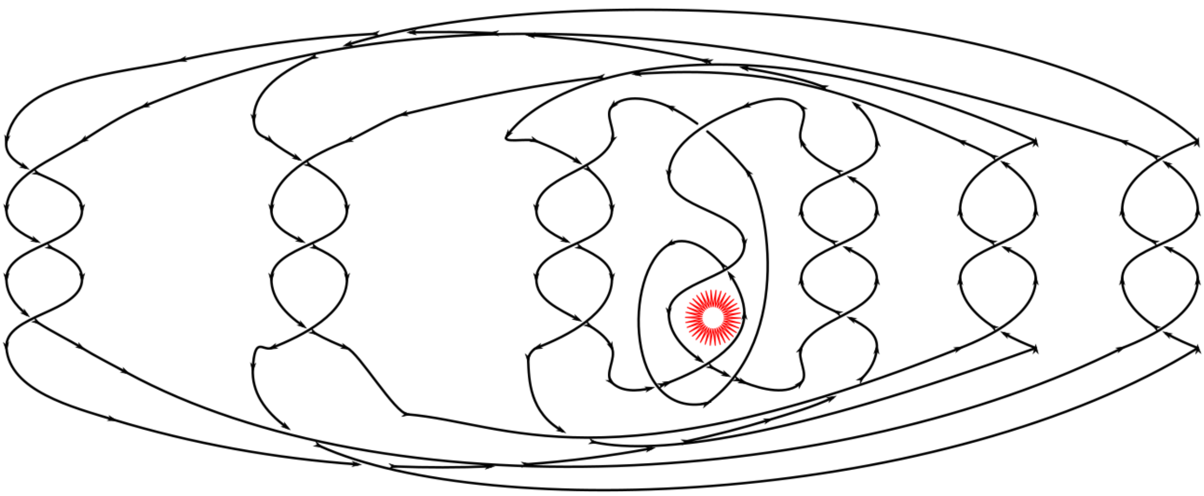


Ilustración 233. Clausura de la trenza correspondiente a P(3, 3, 3, 3, 3, 4, 3).

6. Programación con Python

6.1 Python

A la hora de elegir un lenguaje de programación para poder utilizar el algoritmo de forma dinámica y rápida, la primera idea que se le ocurre a un estudiante de Diseño Industrial de la ETSIDI es utilizar C, después de haberlo estudiado en primero y haber visto todas sus posibilidades y su potencia.

Pero tiene algunas contrapartidas, como la rigurosidad a la hora de declarar variables, usarlas y convertirlas. Pero como buscarle pegas a C es difícil, vamos a hablar de las bondades de Python, el que ha sido uno de los lenguajes con mayor crecimiento en los últimos años [17]. La cuestión es, ¿por qué? Según GeeksforGeeks, podemos dar 6 razones principales [17]:

1. Es fácil de usar. Es muy legible y no es enrevesado, además de ser muy eficiente, permitiendo hacer con menos líneas lo que en otro lenguaje necesitaría más. ¿Cómo escribes las 3 primeras letras de un string en Python?
   str2 = str[0:3].
   ¿Y cómo sería en C?
   char str2[3]; strncpy(str2, str, 3);.
2. Su comunidad: su naturaleza le ha dado una buena acogida y esto ha generado numerosos recursos online que consultar y lugares donde conseguir ayuda.
3. Librerías y frameworks muy versátiles, que hacen que, por ejemplo, para proyectos de WebScrapping sea una de las opciones más indicadas.
4. Soporte empresarial: hay numerosas empresas de gran tamaño financiando el desarrollo de este lenguaje, lo que implica nuevas versiones y funcionalidades. Entre estas empresas tenemos a Facebook, Google o Amazon [18].
5. Se usa en Big Data y Machine Learning. Su gran versatilidad hace que sea elegido para proyectos de las dos grandes tendencias en el mundo de la computación.
6. Se usa en desarrollo web con frameworks muy utilizados como Django, Flask o bottle.

Python es un lenguaje dinámico, mientras que C es estático. Por ejemplo, en C declaramos un entero con *int manzanas=7;*. Si luego intentásemos *manzanas="Casa"*; tendríamos un error. En Python, en cambio, es perfectamente válido manzanas=7 para luego decir *manzanas="Casa"*, ya que se encarga por sí mismo de hacer el casting, aunque se puede hacer manualmente n=str(n).

Otra de las particularidades es que se deben definir las funciones al principio del código, ya que si el intérprete llega a una línea que llama a una función que no está definida todavía porque está abajo no la podrá ejecutar. Otra particularidad muy útil es que una función puede devolver más de un valor, aunque no la llego a usar.

Por otra parte, Python viene de C, por lo que tiene muchas expresiones en común como break o continue.

Es un lenguaje de tipado débil, por lo que hace cosas que de otra manera parecerían una locura en C. Por ejemplo, teniendo 2 enteros 6 y 5, si los dividimos en C tendremos un error, mientras que en Python nos creará un float y hará la división.

Otra de sus maravillas es que los avisos de error son muy específicos sobre qué ocurre:

```
word[42]  # the word only has 6 characters
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
```

¿Qué ha ocurrido? Pues que hemos intentado extraer el 43º valor de una lista que tiene 6 valores.

Otro ejemplo en el que sale ventajoso Python es en la función para calcular los movimientos básicos. Mientras en C hubiésemos necesitado:

```
int Pri(num){
   If(num%2==0){
         Return num/2;
   else{
         return ((num-1)/2)
```

En Python nos vale con:

```
def pri(num): return(int(abs(num//2)))
```

En el programa usaremos bastante la función range(), y es importante tener en cuenta cuando la usamos que tiene 3 variables, el comienzo, el final, y el paso (range(comienzo,final,paso)). Comienzo y paso pueden omitirse, pero final no. La particularidad que quiero destacar es que se ejecuta hasta que llega a final, sin llegar a ejecutar este último valor. Por ejemplo:

```
for i in range(0,5):
  print(i)
```

Imprime:

1
2
3
4

Es decir, se ha ejecutado 5 veces, pero no ha llegado al 5. Por eso muchas veces en el código veremos un +1.

Una utilidad de la que se ha hecho mucho uso ha sido la función breakpoint(), para hacer depuración. Cuando el intérprete llega a esta instrucción pausa la ejecución, y nos permite escribir cualquier comando para imprimir el valor de las variables, por ejemplo.

Por otra parte, hay momentos en que he necesitado imprimir un número variable de veces el mismo valor, como en la función doble(). En estos casos, si uso el bucle:

```
for i in range(rango):
  print(1)
```

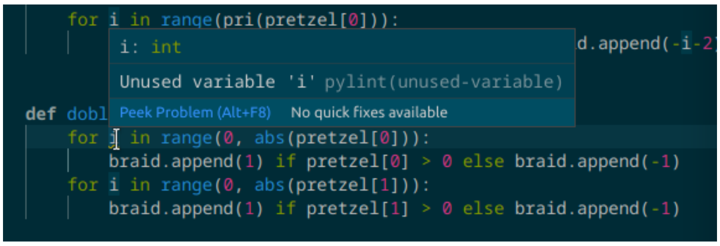


Ilustración 234. Aviso de pylint por variable sin usar.

Pylint avisa de que estoy declarando una variable i que no uso. Esto se arregla sustituyendo la variable por dummy, así le estamos diciendo a pylint que voy a hacer un bucle en el que la variable a iterar no será usada dentro del mismo.

¿Qué es pylint? Seguidamente lo veremos en la subsección 6.2 Visual Studio Code.

En vez de seguir hablando de Python, analizaremos el código del algoritmo en el organigrama.

6.2 Visual Studio Code

En un principio he encontrado recomendados Visual Studio Code (VSCode a partir de ahora), Atom y Sublime. He buscado un entorno que me ofreciese lo mismo que DevC++ pero para Python, es decir, un editor que pudiese ejecutar.

Pero VSCode va mucho más allá. Tiene una librería de extensiones que le permite compilar prácticamente cualquier lenguaje, y añadir cualquier funcionalidad extra que necesitemos.

Lo primero que ocurrirá cuando intentemos ejecutar código Python es que nos saltarán 2 alarmas, la primera porque no tenemos instalado el Linter por defecto, pylint. La segunda porque no tenemos instalado el formateador pep8. ¿Y esto qué significa?

Un Linter analiza el código y busca posibles problemas. Por ejemplo, si en vez de *print("Hello World")* escribimos *print "Hello World",* el Linter nos va a decir no sólo qué es exactamente lo que ocurre, sino cómo arreglarlo:

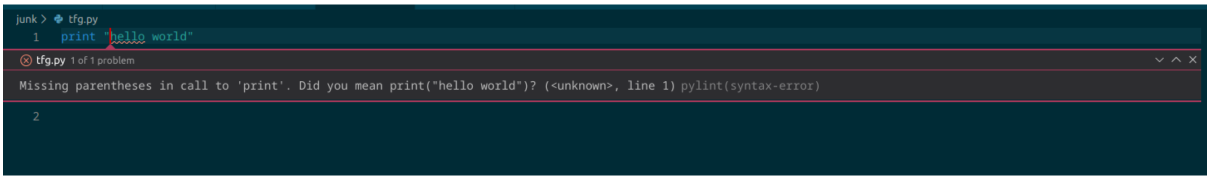


Ilustración 235. Aviso de error de pylint.

Para esto ni siquiera es necesario intentar compilar y ejecutar, simplemente al guardar ya analiza el código.

VSCode permite elegir diferentes Linter, como flake8, mypy, prospector, pydocstyle y pylama. Cada uno tiene pequeñas diferencias, pero el que se suele recomendar es pylint.

¿Y qué es el formatter? En Python existen los Python Enhacement Proposals [19], documentos en los que se recomiendan guías de estilo para homogeneizar el código y hacer que leer código de otra persona sea una tarea menos ardua. Esto significa que una vez lo tengamos activado, al ejecutarlo nos va a transformar x=1 en x = 1. Fuera de la discusión de si de una manera o de otra es más legible, lo importante es que esto hace que se use el mismo estilo en todos lados. Para que se ejecute cada vez que se guarde el archivo tenemos que activar *Format On Save en Ajustes.*

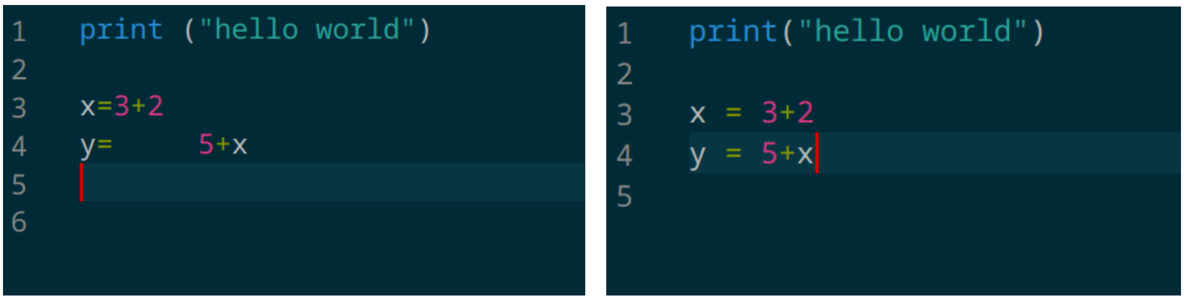


Ilustración 236. El mismo código antes y después de que el formatter haga su función.

6.3 Programa

El programa acepta pretzel de 1 columna (el nudo trivial), de 2, de 3 y de cualquier múltiplo de 2. Se debería encontrar disponible para usarse en todo momento desde http://adelpozoman.es/tfg, de forma que se puede usar con cualquier dispositivo que tenga conexión a internet sin tener que instalar ninguna dependencia como Python o VSCode.

Si se quiere probar o modificar de forma local el código se puede encontrar en el Anexo correspondiente Código main.py, Código funciones.py y Código dibuja.py. También debería estar disponible en todo momento en: https://gitlab.com/adelpozoman/tfg.

Name	Last commit	Last update
.gitignore	Obviar archivos	1 week ago
README.md	Fix description	1 week ago
dibuja.py	Arreglo dibujo Pretzel, mejora Braid	1 week ago
funciones.py	Mejora entrada, acepta 1, 2, 3 y 4 componentes.	1 week ago
main.py	Cambios sugeridos	6 days ago

Ilustración 237. Archivos de código disponibles en GitLab.

El programa se divide en 3 archivos:

- Main.py: es el programa principal, que pide al usuario un enlace pretzel, lo clasifica en función del algoritmo que necesita, llama a funciones.py para procesarlo, y en el caso adecuado, llama a dibuja.py para representar el enlace pretzel y la trenza.
- Funciones.py: incluye todos los cálculos y algoritmos que hay que realizar, para que main.py sea legible y versátil. Incluye código para calcular el número de movimientos básicos de cada columna, leer la entrada del usuario y convertirla en un enlace pretzel, preguntar al usuario sí o no y todos los algoritmos de cada tipo de enlace pretzel.
- Dibuja.py: para dar más esencia al programa se ha incluido una función para dibujar enlaces pretzel de 3 columnas o las trenzas de cualquier número de cuerdas y cruces. Las trenzas no se representan bien cuando el ancho es insuficiente, pero esto no tiene solución sencilla.

El entorno usado para programarlo consta de: Python 3.8.2 64 bits, Kubuntu 20.04, VSCode 1.45.2.

El código se ha probado con todas las variables y combinaciones posibles, y se invita al lector a contrastarlo con los nudos desarrollados a lo largo del capítulo V.

A continuación, se va a analizar el código del programa.

6.4 Organigrama del código

Se ha hecho un organigrama para observar visualmente el flujo del programa.

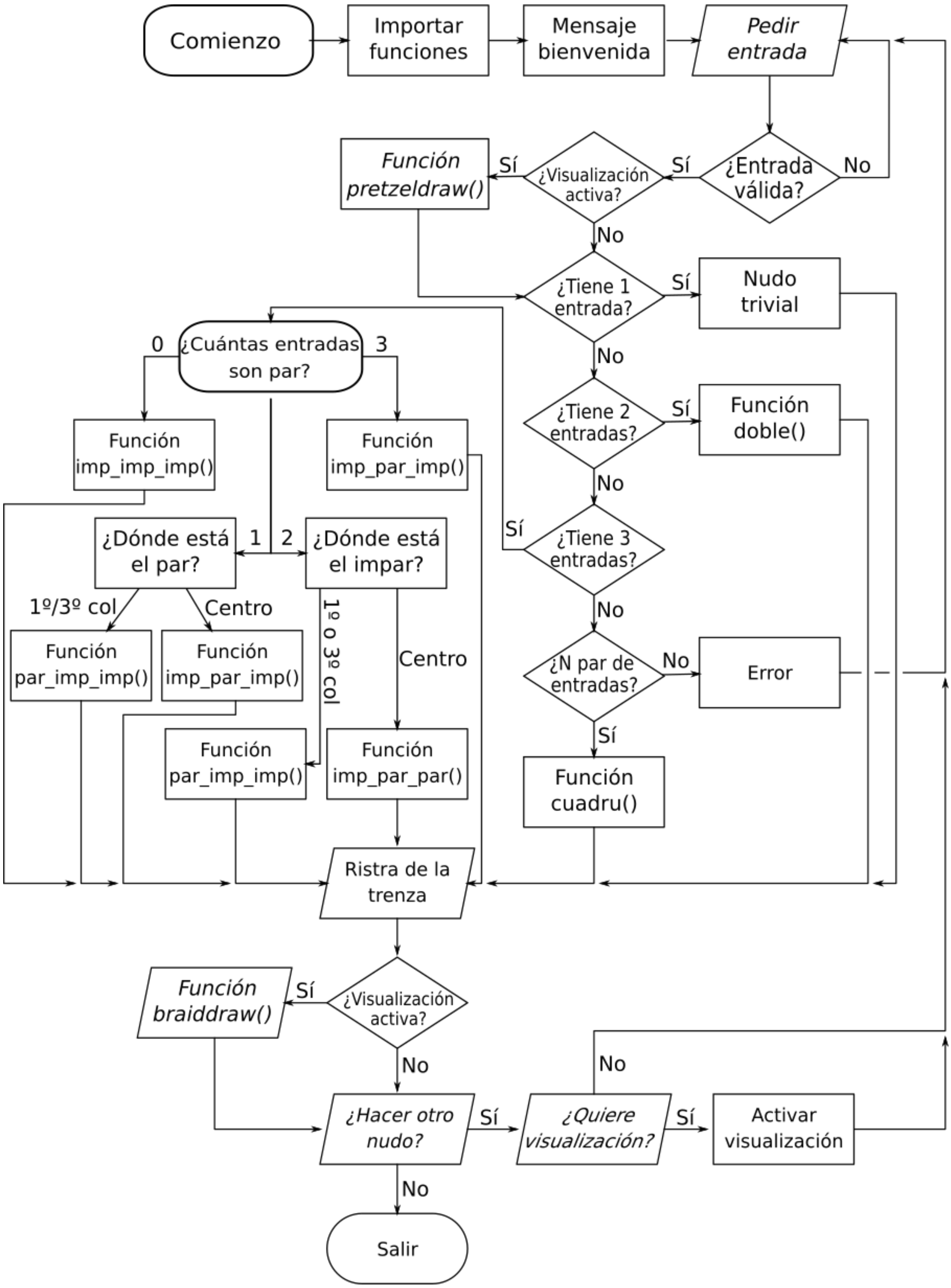


Ilustración 238. Organigrama del código.

El programa empieza importando las funciones necesarias de otros archivos, luego da un mensaje de bienvenida al usuario y le pide que introduzca un nudo pretzel. Esto lo hace llamando a una función que comprueba si ha recibido una secuencia separada por comas, espacios o puntos y en caso de ser así la guarda en una *lista* de números. Si ha habido alguna letra, dos puntos entre los que no hay un número o algún otro error así pide la entrada de nuevo.

A continuación, dibuja el pretzel (sólo si es de 3 columnas) si la visualización está activa, para lo que es necesario que se haya ejecutado al menos una vez un nudo y se haya activado, ya que por defecto está desactivada para no sobrecargar la pantalla.

Ahora llega la gracia del programa. ¿Tiene una entrada? Pues devuelve el nudo trivial. En caso contrario, ¿tiene dos entradas? Pues llama a la función correspondiente. ¿Tiene 3 entradas? Entonces tenemos que clasificar primero el enlace para saber a qué función llamar:

- Ninguna entrada par: función imp_imp_imp().
- Una entrada par: ¿dónde está?
    - Primera columna: función par_imp_imp().
    - Segunda columna: función imp_par_imp().
    - Tercera columna: función par_imp_imp().
- Dos entradas pares: ¿dónde está el impar?
    - Primera columna: función par_imp_imp().
    - Segunda columna: función par_imp_imp().
    - Tercera columna: función par_imp_imp().
- Tres entradas pares: función imp_par_imp().

Como el lector apreciará, esto es absolutamente confuso, hay funciones que sirven para más de un tipo de enlace, tal y como se justifica en el capítulo V. Recordemos también que teniendo una función para P(a, b, c) dicha función nos sirve para P(c, b, a) si lo invertimos.

Si se hubiese dado el caso de no tener 3 entradas, pasaría a comprobar si hay un número par de ellas. Si es un número par llama a la función cuadru(); el caso contrario no debería ocurrir nunca, ya que debería detectarse en la lectura.

Tras detectar una entrada válida y llamar a su función correspondiente, imprime la ristra de números de la palabra de la trenza. En el caso de estar la visualización activa la representa horizontalmente.

Para finalizar pregunta al usuario si quiere procesar otro nudo. En caso negativo se acaba el bucle, pero en caso positivo nos presenta la opción de activar la visualización del enlace pretzel y la trenza cuando corresponda. Independientemente de lo que el usuario elija, volverá a la petición de entrada inicial.

Creo necesario comentar otros aspectos del programa que no tienen cabida en el organigrama.

- La función pregunta() comprueba si la entrada del usuario es Sí o una palabra parecida o si es No o una palabra parecida, en cuyo caso devuelve un 1 o un 0 respectivamente.
- La función pretzeldraw() sólo dibuja pretzel de 3 torres ya que era el objetivo inicial del TFG y es difícil adaptar el código para más torres.
- La función braiddraw() falla si el ancho de la pantalla en la que se está visualizando es insuficiente, algo un poco difícil de solucionar.
- La función pri() calcula la cantidad de movimientos básicos que hay que hacer en una columna, se usa especialmente en la función imp_imp_imp(), donde se hacen movimientos básicos en las tres columnas y es un valor recurrente.
- La función pares(), como su nombre indica, calcula la cantidad de números pares entre las entradas de un pretzel.

Si para un pretzel de 3 columnas con una única entrada par al principio usamos la función par_imp_imp(), con uno que tenga la entrada par al final usamos la misma función pero dándole los valores al revés aprovechando que P(a, b, c)=P(c, b, a). Esto se hace llamando a par_imp_imp(reversed()), que invierte los argumentos antes de dárselos a la función.

Debo pedir disculpas al lector, pues si ha leído la sección "Enlaces pretzel con tres entradas. Síntesis", encontrará extraño que use un algoritmo para cada tipo de pretzel (de tres entradas), si estamos de acuerdo en que sólo hay 2 tipos. La razón es que el programa se hizo con bastante anterioridad al descubrimiento de la síntesis que facilitaba la propiedad cíclica. Por otra parte, me gusta tener de esta manera los algoritmos individuales, ya que cuando quiero probar otro enlace pretzel, el algoritmo me permite ver el resultado que obtendría por el método directo, sin usar la propiedad cíclica. Aunque siendo sinceros, siendo el método de la síntesis más rápido y atractivo, no quería desechar el trabajo ya realizado.

6.5 Página web

Una de las cosas que he creído más convenientes, y que le puede dar más visibilidad al proyecto es poder hacerlo accesible desde internet. Está muy bien poder ver el código y analizarlo, pero si sólo se quiere evaluarlo y probarlo, tener que instalar Python y un compilador puede ser pedir algo excesivo y muy poco conveniente.

Mismamente mientras se lee este proyecto puede surgir la duda de qué ocurriría si en el capítulo Nudos P(par, impar, impar) se escogiese el nudo P(6,5,9) en vez del P(6,5,7). Teniendo cualquier navegador web en apenas 10 segundos podemos corroborar nuestra hipótesis de la trenza que saldría según el algoritmo encontrado en dicho capítulo.

Por eso, una vez finalizado el programa, surge esta idea. Si Python es tan famoso y tan portable, veamos si funciona online. Pues realmente funciona.

Tengo en mi casa una Raspberry Pi 3B+ (como curiosidad, salió a la venta el día de Pi, 3/14) que no estaba teniendo mucho uso, sólo como servidor DNS, así que hice una pequeña investigación sobre las posibilidades de este hardware para correr un sitio web de forma medianamente decente.

Ilustración 239. Raspberry Pi 3 B+ [20].

Resulta que es una opción perfecta para lo que yo buscaba, un pequeño sitio web que sirva contenido estático. Para hacer la web hay muchas opciones, pero me decanté por la más recomendada para este tipo de situación, una web WordPress.

La instalación no fue especialmente complicada. Primero se instala NGINX como servidor, PHP para generar las páginas y MariaDB como gestor de la base de datos. Luego se colocan los archivos descargados de wordpress.org en la carpeta pública del servidor, se accede desde el navegador a la dirección local de la Raspberry y comienza la instalación.

Una vez que la página estaba corriendo y tenía suficiente fluidez, había que hacerla accesible desde el exterior. Para ello desde la configuración del router tuve que abrir los puertos 80 y 443 (para http y https respectivamente) y re direccionarlos a la IP local de la Raspberry, 192.168.1.114.

Pedí a diferentes amigos que probaran desde sus casas, desde Canarias, Sevilla o Galicia, y a todos les funcionaba bien. Pero tenían que usar mi IP dinámica. Por decirlo de forma simple, todas las casas tienen una dirección IP que es la que damos si queremos que alguien acceda a un recurso en nuestra red, como es en nuestro caso la web. Esta IP consiste en 4 números separados por puntos, como puede ser 37.152.39.235. Puedes consultar la tuya en vermiip.es.

El problema es que esta IP no es fija y va cambiando frecuentemente, por lo que la dirección que hemos dado hoy puede no funcionar mañana. Para esto existen las IP fijas como el servidor 8.8.8.8 de Google. El problema es que son caras, y la gente sigue usando números para conectarse a nuestra web.

La solución es comprar un dominio. Me decidí por adelpozoman porque es mi correo y contiene mi nombre y apellidos, y .es porque está en España. Hay múltiples proveedores de dominios, pero me decidí por GoDaddy, dado lo expandido que está por internet, llegando a ser en 2018 el servicio más grande de alojamiento web [21] [22].

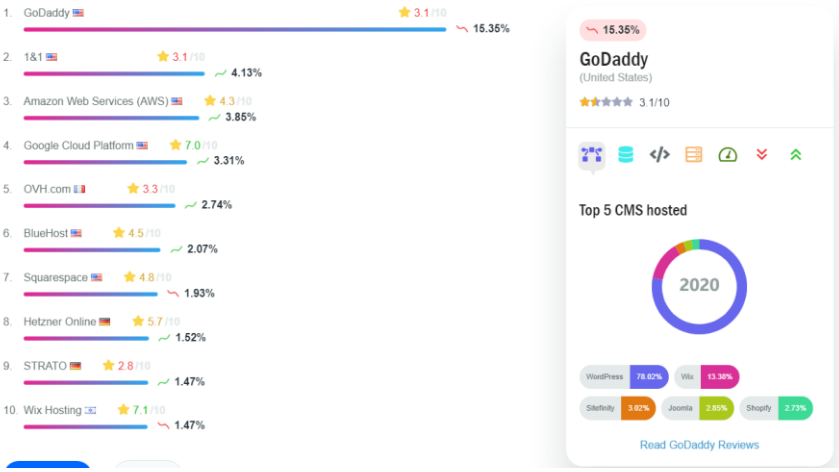


Ilustración 240.Ranking de alojamiento web [22].

Como podemos apreciar en la Ilustración 240.Ranking de alojamiento web un gran porcentaje de personas usa GoDaddy para alojar su web WordPress, en concreto un 78%. Pero mientras que una WordPress en GoDaddy puede costar desde 4 a 15€ al mes, únicamente el dominio me ha costado 1€ al año y 10€ al año los consecutivos, pero esto es un problema futuro.

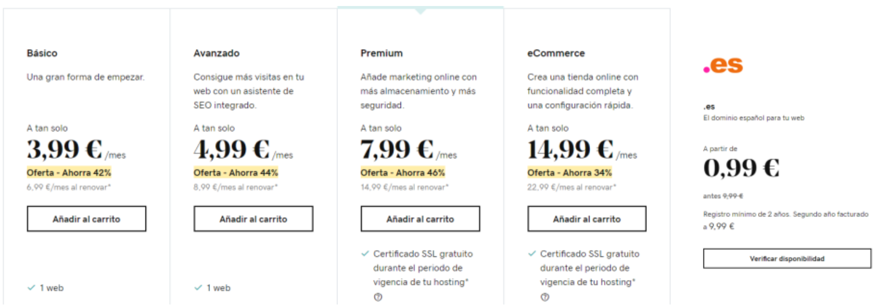


Ilustración 241. Precios para webs y para dominios en es.godaddy.com.

Ahora sólo nos queda decirle a GoDaddy que cuando alguien visite adelpozoman.es lo dirija a nuestra IP. Eso se hace desde su panel de control:

Registros

Última actualización 08/06/20 14:29

Tipo	Nombre	Valor	TTL
A	@	37.134.9.142	600 segundos
CNAME	www	@	1 hora
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com	1 hora

Ilustración 242. Panel de control de dominios de GoDaddy.

Pero todavía queda otro pequeño problema. ¿Qué ocurrirá cuando cambie nuestra IP pública? Pues que la página dejará de funcionar, ya que GoDaddy dirigirá a la IP antigua (37.134.9.142). Por esto existen los dominios dinámicos, que son capaces de cambiar la IP a la que apuntan de forma rápida cuando sea necesario.

Esto se implementa creando un script en nuestro servidor (recordemos, en nuestro caso la Raspberry) que compruebe la IP pública que tiene en ese momento, accede a GoDaddy mediante una API y una clave privada personal y actualiza la información en caso de ser necesario. Este script se ejecuta cada 10 minutos. Para hacerlo se han seguido las instrucciones encontradas en Instructables.com. [23]

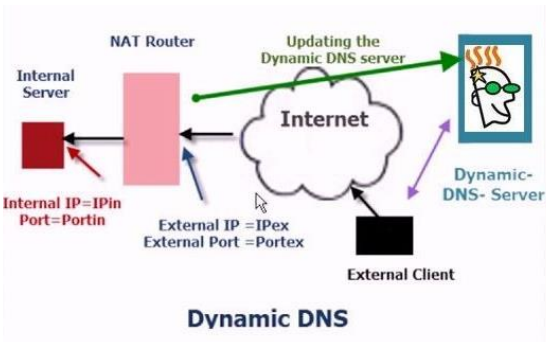


Ilustración 243. Diagrama de funcionamiento de un dominio dinámico [23].

En un principio cuando compremos el dominio puede tardar en funcionar hasta 24 o 48 horas, ya que tiene que llegar a los servidores de DNS de todo el mundo la existencia de este nuevo dominio, pero en mi caso ha tardado sólo 4 horas. Este fenómeno se llama DNS propagation [24].

6.6 Interfaz web

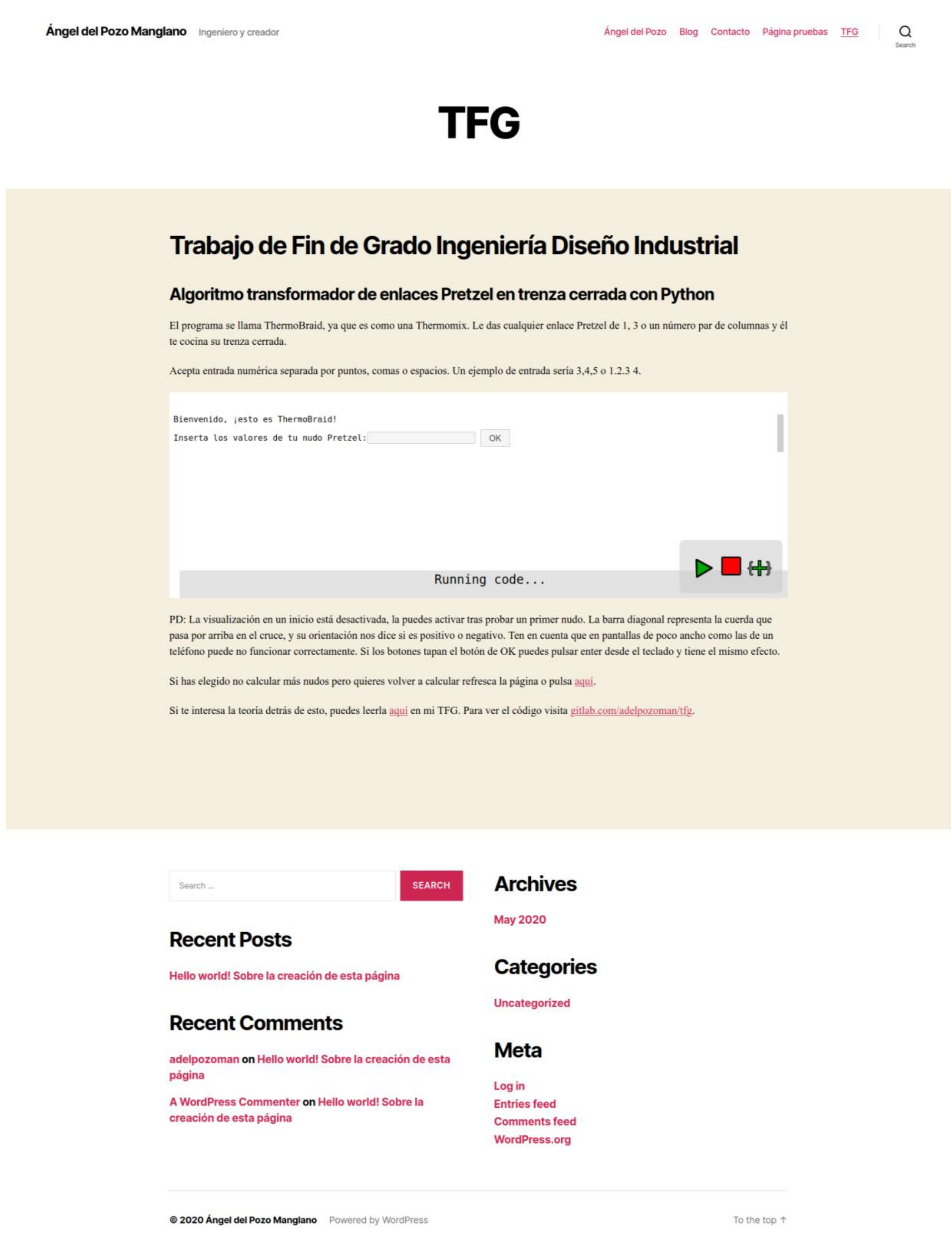


Ilustración 244. Interfaz de la página web http://adelpozoman.es/tfg.

Además del programa hay una breve introducción al TFG y unas instrucciones para usarlo. La página me puede venir bien en el futuro ya que puedo mantenerla como una web-currículum bastante llamativa.

Conclusión

Este proyecto ha acabado adquiriendo gran complejidad y nos da una lección muy importante sobre la gran diferencia entre un teorema y su aplicación. Mientras que en casos muy simples dos movimientos de reducción pueden ser suficientes para hallar la trenza cuya clausura es el enlace pretzel dado, en general, para un nudo pretzel de tres entradas y las tres impares, la solución adquiere gran complejidad, no siendo suficiente el movimiento de reducción.

El objetivo inicial del TFG, encontrar las trenzas para los nudos pretzel de tres entradas, ha sido ampliamente satisfecho, llegando a metas mayores. El descubrimiento de un método válido para cualquier enlace pretzel con un número par de entradas ha sido muy inesperado por su accesibilidad, sobre todo en comparación con el caso de tres entradas. Más aún, hemos logrado dar una estrategia general para cualquier *nudo* pretzel con un número arbitrario de entradas.

Desde un punto de vista matemático, han sido clave en el desarrollo de la estrategia y los algoritmos los dos movimientos descubiertos: movimiento básico y desplazamiento de columna.

Por otro lado, ha sido de vital importancia la búsqueda de nuevas formas de trabajar, y no centrarse en los métodos establecidos, como ha demostrado el uso de Inkscape, herramienta sin la cual no habría sido posible llevar a cabo este TFG.

Por último, la inclusión de una página web ha sido un elemento refrescante que le da un gran valor por su accesibilidad y versatilidad.

A lo largo de todo el trabajo han ido surgiendo trenzas con diferente número de cuerdas. Ahora nos surge una pregunta que se deja para que el lector consulte con la almohada: cuando obtenemos la trenza correspondiente a un pretzel, ¿es ésta la que tiene el menor número de cuerdas? ¿Puede haber otro método para obtener una trenza con menos cuerdas? Finalmente, y como posible trabajo futuro que nos parece al alcance de la mano, nos gustaría generalizar la estrategia dada para nudos pretzel al caso de enlaces pretzel.

Bibliografía

[1] L. Eguren, «Optimización topológica,» julio 2012. [En línea]. Available: http://138.100.100.254/index/departamentos/matematicas/manchon/pman_archivos/PFCLaura21Junio.pdf. [Último acceso: junio 2020].

[2] N. S. Calvo, «Independencia promedio de grafos y polinomio de Jones. Aplicaciones en Ingeniería.,» Febrero 2007. [En línea]. Available: http://138.100.100.254/index/departamentos/matematicas/manchon/pman_archivos/PFC%20Nadia%20Srour.doc. [Último acceso: junio 2020].

[3] E. Flapan, When topology meets chemistry, Cambridge University Press , 2000.

[4] M. Suffczynski, «Braids for Pretzel Knots,» *Acta Physica Polonica A,* vol. 98, nº 6, pp. 663-671, 2000.

[5] D. Garber, «Braid Group Cryptography,» 26 noviembre 2007. [En línea]. Available: https://arxiv.org/abs/0711.3941. [Último acceso: 10 junio 2020].

[6] K. Reidemeister, de *Elementare Begründung der Knotentheorie*, Hamburg, 1927, pp. 24 - 32.

[7] Jkasd, «File:Knot table.svg,» 25 Abril 2008. [En línea]. Available: https://commons.wikimedia.org/wiki/File:Knot_table.svg. [Último acceso: junio 2020].

[8] C. C. Adams, The Knot Book: an elementary introduction to the mathematical theory of knots, Williamstown, MA: American Mathematical Society , 2004.

[9] P. Cromwell y E. Beltrami, «The mathematical tourist,» de *The Mathematical Intelligencer 20*, 1998, pp. 53-62.

[10] П. @jemmybutton, «Various things in MetaPost,» Habr, 3 Junio 2019. [En línea]. Available: https://m.habr.com/en/post/454376/. [Último acceso: Mayo 2020].

[11] Inkscape Team, «Download Inkscape 1.0,» Inkscape, 1 Mayo 2020. [En línea]. Available: https://inkscape.org/release/inkscape-1.0/. [Último acceso: 5 Mayo 2020].

[12] R. Díaz y P. M. G. Manchón, «Pretzel knots up to nine crossings,» Preprint 2020.

[13] J. W. Alexander, «A Lemma on Systems of Knotted Curves,» Princeton, Department of Mathematics, Princeton University, 1923, pp. 93-95.

[14] P. R. Cromwell, «10 Closed Braids and Arc Presentations,» de *Peter R. Cromwell*, Cambridge University Press, 2004, pp. 238-239.

[15] A. M. Jr., *Über die freie Äquivalenz der geschlossenen Zöpfe,* 1936, pp. 73-78.

[16] S. Yamada, «The minimal number of Seifert circles equals,» *Inventiones mathematicae,* pp. 347-356, 1987.

[17] H. Kaur, «Python – Fastest Growing Programming Language,» GeeksforGeeks, [En línea]. Available: https://www.geeksforgeeks.org/python-fastest-growing-programming-language/. [Último acceso: 2 June 2020].

[18] «Python Software Foundation Sponsors,» Python Software Foundation, [En línea]. Available: https://www.python.org/psf/sponsorship/sponsors/. [Último acceso: Mayo 2020].

[19] p.-d. <.-d. a. python.org>, «PEP 0 -- Index of Python Enhancement Proposals (PEPs),» Python, 13 Julio 2000. [En línea]. Available: PEP 0 -- Index of Python Enhancement Proposals (PEPs). [Último acceso: Mayo 2000].

[20] «Raspberry Pi 3 Model B+,» Raspbery Pi, 14 Marzo 2018. [En línea]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. [Último acceso: Junio 2020].

[21] «Usage Statistics and Market Share of Web Hosting Providers for Websites,» w3techs.com, 04 Julio 2018. [En línea]. Available: https://w3techs.com/technologies/overview/web_hosting. [Último acceso: 04 Julio 2018].

[22] «Global Web Hosting Market Share June 2020,» HostAdvice, 2020. [En línea]. Available: https://hostadvice.com/marketshare/. [Último acceso: Junio 2020].

[23] Tod-SoS, «Quick and Dirty Dynamic DNS Using GoDaddy,» Instructables, [En línea]. Available: https://www.instructables.com/id/Quick-and-Dirty-Dynamic-DNS-Using-GoDaddy/. [Último acceso: Junio 2020].

[24] «WHAT IS DNS PROPAGATION?,» ssl247, [En línea]. Available: https://www.ssl247.es/kb/domain-name-management/about/what-is-dns-propagation. [Último acceso: Junio 2020].

Índice de ilustraciones

Índice de ecuaciones

Índice de tablas

Anexo

Código main.py

```
from funciones import pri, pares, pregunta, lectura, imp_par_imp,
par_imp_imp, imp_imp_imp, imp_par_par, doble, cuadru
from dibuja import pretzeldraw, braiddraw

print("Bienvenido, ¡esto es ThermoBraid!")

grafico = 0
while True:
    pretzel = []
    braid = []
    lectura(pretzel)
    print("Tu nudo Pretzel es:", pretzel)
    if grafico == 1 and len(pretzel) == 3:
        pretzeldraw(pretzel)

    if len(pretzel) == 1:
        print("¡El nudo trivial!")
        braid = [0]
    elif len(pretzel) == 2:
        doble(pretzel, braid)
    elif len(pretzel) == 3:
        if int(pares(pretzel)) == 2:
            if pretzel[1] % 2 != 0:
                par_imp_imp(pretzel, braid)
            elif pretzel[0] % 2 != 0:
                imp_par_par(pretzel, braid)
            elif pretzel[2] % 2 != 0:
                imp_par_par(list(reversed(pretzel)), braid)
        elif int(pares(pretzel)) == 3:
            imp_par_imp(pretzel, braid)
        elif int(pares(pretzel)) == 1:
            if pretzel[0] % 2 == 0:
                par_imp_imp(pretzel, braid)
            elif pretzel[1] % 2 == 0:
                imp_par_imp(pretzel, braid)
            elif pretzel[2] % 2 == 0:
                par_imp_imp(list(reversed(pretzel)), braid)
        elif int(pares(pretzel)) == 0:
            imp_imp_imp(pretzel, braid)
    elif len(pretzel) % 2 == 0:
        cuadru(pretzel, braid)
    else:
        print('error')
    print("\nSi lo transformamos en trenza, tiene {} cruce{}, {} cuerda{} y
vale:\n{}".format(
        len(braid), 's' if len(braid) > 1 else '', max(braid)+1 if
abs(min(braid)) < max(braid) else abs(min(braid))+1, 's' if len(braid) > 1
else '', braid))
    if grafico == 1 and (braid != None or braid != [0]):
        braiddraw(braid)
```

```
    print("")

    if pregunta("¿Quieres introducir otro nudo Pretzel?(si/no)") != 1:
        break
    if pregunta("¿Quieres visualizar el nudo?(si/no)") != grafico:
        if grafico == 0:
            grafico = 1
            print("Visualización activada")
        elif grafico == 1:
            grafico = 0
            print("Visualización desactivada")
```

Código funciones.py

```
def pri(n): return int(abs(n)//2)


def lectura(pretzel):
    while True:
        str = input("Inserta los valores de tu nudo Pretzel:")
        str = str.replace('.', ',').replace(
            '+', ',').replace(' ', ',').split(',')
        try:
            for i in str:
                pretzel.append(int(i))
            if len(pretzel) % 2 == 0 or len(pretzel) == 3 or len(pretzel)
== 1:
                return
            else:
                print("No puedo procesar un Pretzel de {} columnas, vuelve
a intentarlo".format(
                    len(pretzel)))
                pretzel.clear()
        except ValueError:
            print("No me has dado números")
            pretzel.clear()


def pregunta(quest):
    while True:
        ok = input(quest+' ')
        if ok in ('Sí', 'Si', 'si', 'sí', 'SI', 'sI', 'S', 'yes', 'Yes',
'sure', 'roger', '1', 'Continuar', 'continuar', 'claro', 'Activar',
'activar', 'activa'):
            return True
        if ok in ('No', 'n', 'no', 'nO', 'nop', 'nope', 'Negative', '0',
'Salir', 'salir'):
            return False
        print("No has escogido una opción válida")


def pares(pretzel):
    cont = 0
    for i in range(0, len(pretzel)):
        if pretzel[i] % 2 == 0:
            cont += 1
    return cont


def imp_par_imp(pretzel, braid):
    for i in range(1, pri(pretzel[1])+1):
        braid.append(-i-1) if pretzel[1] > 1 else braid.append(i+1)
    for i in range(0, abs(pretzel[0])):
        braid.append(1) if pretzel[0] > 0 else braid.append(-1)
    for i in range(1, pri(pretzel[1])):
        braid.append(i+1) if pretzel[1] > 0 else braid.append(-i-1)
    for i in reversed(range(1, pri(pretzel[1])+1)):
        braid.append(-i-1) if pretzel[1] > 0 else braid.append(i+1)
    for i in range(0, abs(pretzel[2])):
        braid.append(1) if pretzel[2] > 0 else braid.append(-1)
```

```
def par_imp_imp(pretzel, braid):
    for i in range(0, pri(pretzel[0])):
        braid.append(-i-1) if pretzel[0] > 0 else braid.append(i+1)
    for i in range(0, abs(pretzel[1])):
        braid.append(
            pri(pretzel[0])+1) if pretzel[1] > 0 else braid.append(-
pri(pretzel[0])-1)
    for i in reversed(range(0, pri(pretzel[0]))):
        braid.append(-i-1) if pretzel[0] > 0 else braid.append(i+1)
    for i in range(2, pri(pretzel[0])+1):
        braid.append(i) if pretzel[0] > 0 else braid.append(-i)
    for i in range(0, abs(pretzel[2])):
        braid.append(
            pri(pretzel[0])+1) if pretzel[2] > 0 else braid.append(-
pri(pretzel[0])-1)


def imp_par_par(pretzel, braid):
    for i in range(0, pri(pretzel[2])):
        braid.append(-i-1) if pretzel[2] > 0 else braid.append(i+1)
    for i in range(0, abs(pretzel[0])):
        braid.append(
            pri(pretzel[2])+1) if pretzel[0] > 0 else braid.append(-
pri(pretzel[2])-1)
    for i in reversed(range(0, pri(pretzel[2]))):
        braid.append(-i-1) if pretzel[2] > 0 else braid.append(i+1)
    for i in range(2, pri(pretzel[2])+1):
        braid.append(i) if pretzel[2] > 0 else braid.append(-i)
    for i in range(0, abs(pretzel[1])):
        braid.append(
            pri(pretzel[2])+1) if pretzel[1] > 0 else braid.append(-
pri(pretzel[2])-1)


def imp_imp_imp(pretzel, braid):
    for i in range(1, pri(pretzel[0])+2):
        braid.append(-i) if pretzel[0] > 0 else braid.append(i)
    for i in reversed(range(2, pri(pretzel[0])+2)):
        braid.append(-i) if pretzel[0] > 0 else braid.append(i)
    # A2
    for i in range(pri(pretzel[1])):
        for j in range(-(2+pri(pretzel[0])+i), -2-i):
            braid.append(j)
    # A3
    for i in range(pri(pretzel[2])):
        for j in range(2+pri(pretzel[1])+pri(pretzel[0])+i,
2+pri(pretzel[1])+1+i-1, -1):
            braid.append(j)
    # A4
    for i in range(pri(pretzel[2])):
        for j in range(2+pri(pretzel[1])+i, 2+i, -1):
            braid.append(j)
    # A5
    for i in range(pri(pretzel[2])):
        braid.append(i+2) if pretzel[2] > 0 else braid.append(-i-2)
    # A6, casi equivalente a A1
    for i in range(1, pri(pretzel[2])+2):
        braid.append(-i) if pretzel[2] > 0 else braid.append(i)
    for i in reversed(range(2, pri(pretzel[2])+2)):
        braid.append(-i) if pretzel[2] > 0 else braid.append(i)
    # A7
```

```
    for i in range(pri(pretzel[1])):
        for j in range(-2-pri(pretzel[2])-i, -2-i):
            braid.append(j)
    # A8
    for i in range(pri(pretzel[1])):
        braid.append(i+2) if pretzel[1] > 0 else braid.append(-i-2)
    # A9
    for i in range(1, pri(pretzel[1])+2):
        braid.append(-i) if pretzel[1] > 0 else braid.append(i)
    for i in reversed(range(2, pri(pretzel[1])+2)):
        braid.append(-i) if pretzel[1] > 0 else braid.append(i)
    # A10
    for i in range(pri(pretzel[0])):
        for j in range(pri(pretzel[2])):
            braid.append(-2-pri(pretzel[1])-pri(pretzel[2])-i+j)
    # A11
    for i in range(pri(pretzel[0])):
        for j in range(2+pri(pretzel[1])+i, 2+i, -1):
            braid.append(j)
    # A12
    for i in range(pri(pretzel[0])):
        braid.append(i+2) if pretzel[0] > 0 else braid.append(-i-2)


def doble(pretzel, braid):
    for dummy in range(0, abs(pretzel[0])):
        braid.append(1) if pretzel[0] > 0 else braid.append(-1)
    for dummy in range(0, abs(pretzel[1])):
        braid.append(1) if pretzel[1] > 0 else braid.append(-1)


def cuadru(pretzel, braid):
    for i in range(1, int((len(pretzel)/2))):
        braid.extend((-2*i, i*2-1, 2*i))
    for i in reversed(range(0+1, len(pretzel)+1, 2)):
        for dummy in range(abs(pretzel[len(pretzel)-i-1])):
            braid.append(i) if pretzel[len(pretzel) -
                                       i-1] > 0 else braid.append(-i)
    for i in reversed(range(1, int(len(pretzel)/2))):
        braid.extend((-2*i, -2*i+1, 2*i))
    for i in range(0, len(pretzel), 2):
        for dummy in range(abs(pretzel[1+i])):
            braid.append(
                len(pretzel)-1-i) if pretzel[i+1] > 0 else braid.append(-
(len(pretzel)-1-i))
```

Código dibuja.py

```
def pretzeldraw(pretzel):
    print("╭────────────╮")
    print("│  __   __   │")
    altura = int(max(pretzel))
    if abs(min(pretzel)) > altura:
        altura = int(abs(min(pretzel)))
    for i in range(0, int(altura)):
        print("\ /  \ /  \ /")
        if pretzel[0] < 0 and i < abs(pretzel[0]):
            print(" \\ ", end='')
        elif pretzel[0] > 0 and i < abs(pretzel[0]):
            print(" / ", end='')
        else:
            print("│ │", end='')
        print("  ", sep='', end='')
        if pretzel[1] < 0 and i < abs(pretzel[1]):
            print(" \\ ", end='')
        elif pretzel[1] > 0 and i < abs(pretzel[1]):
            print(" / ", end='')
        else:
            print("│ │", end='')
        print("  ", sep='', end='')
        if pretzel[2] < 0 and i < abs(pretzel[2]):
            print(" \\")
        elif pretzel[2] > 0 and i < abs(pretzel[2]):
            print(" /")
        else:
            print("│ │")
        print("/ \  / \  / \\")

    print("│  ‾ ‾    ‾ ‾    │")
    print("╰────────────╯")


def braiddraw(braid):
    ancho = int(max(braid))+1
    if abs(min(braid))+1 > ancho:
        ancho = int(abs(min(braid)))+1
    for i in range(0, ancho*2-1):
        if i % 2 == 0:
            print('─', end='')
            for j in range(0, len(braid)):
                if abs(braid[j]) == ancho-i/2-1 or int(abs(braid[j])) ==
ancho-i/2:
                    if braid[j] < 0:
                        print(" ─", end='')
                    elif braid[j] > 0:
                        print(" ─", end='')
                else:
                    print("──", end='')

        if i % 2 == 1:
            print(' ', end='')
            for j in range(0, len(braid)):
                if abs(braid[j]) == ancho-i//2-1:
                    if braid[j] < 0:
                        print("╱ ", end='')
                    elif braid[j] > 0:
                        print("╲ ", end='')
                else:
                    print("  ", end='')
        print("")
```