

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.

Tesis:
MONODROMÍAS GEOMÉTRICAS EN
FAMILIAS DE CURVAS DE GÉNERO 4

Isabel Berna Sepúlveda
Director: Jaume Amorós Torrent

Programa de Doctorat en Matemàtica Aplicada
Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya, Barcelona

Noviembre de 2011

Memoria presentada para aspirar al título de Doctora
por la Universitat Politècnica de Catalunya

Índice general

Dedicatoria y Agradecimientos	2
Resumen	4
0. Introducción	5
0.1. Introducción de la Tesis	5
0.2. Estructura y contenidos de la Tesis	8
0.3. Resultados originales de la Tesis	12
1. Fundamentos de la Tesis	13
1.1. Estructura Trigonal en Curvas de Género 4	13
1.2. Forma normal de Clebsch del recubrimiento trigonal	17
1.3. Familia de Curvas de Género 4	23
1.4. Monodromía en el Grupo Fundamental	25
1.5. El Grupo de Trenzas	27
1.6. Singular	30
2. Monodromía de la Familia Universal de Curvas Trigonaes	31
2.1. La familia universal de Clebsch-Hurwitz	31
2.2. Monodromía de la familia universal de Clebsch-Hurwitz de curvas trigonaes de género 4	38
3. Cálculo	153
3.1. Cálculos Algebraicos	153
3.2. Listado de los programas del cálculo algebraico	156
3.2.1. Listado de los programas de la librería genero_4.lib	156
3.2.2. Listado programas Pari-GP	177
3.3. Cálculos Analíticos	180
3.3.1. Fundamento Teórico del Cálculo Analítico	180
3.4. Listado de los programas de cálculo de monodromías de trenzas en Matlab	188
3.4.1. Listado programas Matlab	191
4. Ejemplos	215
4.1. Ejemplos Académicos	215
4.1.1. Familia Simple 1	215
4.1.2. Familia Simple 2	218
4.1.3. Familia de Seidel	220
4.1.4. Familia de Bring	223
4.2. Cálculo Analítico: Ejemplo Académico	225
Bibliografía	232

Dedicatoria

Dedico este trabajo de Tesis a mi gran familia; a mi padre, a mi madre, a mis tres hermanos y tres hermanas, a mis siete sobrinos y siete sobrinas, que a pesar de la distancia, siempre han estado presente en todo este tiempo.

A los amigos de siempre, a los que están lejos y a los que están cerca, por transmitirme su cariño, apoyo y ánimos.

Agradecimientos

Primero y como más importante, me gustaría agradecer sinceramente a mi director de Tesis, Dr. Jaume Amorós Torrent, a quien admiro y agradezco su esfuerzo y dedicación. Sus conocimientos, sus orientaciones, su manera de trabajar, su persistencia, su paciencia y su motivación que han sido fundamentales para mi formación. Estoy en deuda con él por todo lo recibido durante el período de tiempo que ha durado esta Tesis Doctoral.

A la Universidad Metropolitana de Ciencias de la Educación, a los profesores del Departamento de Matemáticas, en especial los profesores Juan Yañez González, Nelsón Aravena Castillo y Marcela Pinochet Bustos, por haber sido parte mi formación inicial, y posteriormente seguir apoyándome en esta etapa y tener la confianza en mi.

Resumen

Se describe el proceso de transformación de una familia de superficies de Riemann compactas, conexas de género 4 en una familia de recubrimientos trigonales de la esfera de Riemann.

Se halla una familia universal para tales recubrimientos trigonales de género 4.

Se calcula la monodromía en el grupo fundamental de las fibras para dicha familia.

Se desarrolla una librería informática basada en **Singular** para el cálculo algebraico efectivo en estas curvas y familias.

Se desarrolla una librería informática basada en MATLAB para el cálculo analítico efectivo de monodromías de trenzas.

Abstract

The transformation proces of a family of connected compact Riemann surfaces of genus 4 into a family of trigonal coverings of the Riemann sphere is cescribed.

A universal family for such trigonal covers of genus 4 is found.

The monodromy in the fundamental group of the fibres of this family is computed.

A software library based a **Singular** is developed for the effective algebraic computation of these curves and families.

A software library based a MATLAB is developed for the effective analytic computation of braid monodromies.

Capítulo 0

Introducción

0.1. Introducción de la Tesis

El propósito de esta tesis es el estudio de la monodromía geométrica y en el grupo fundamental de familias de curvas de género 4.

Las razones para el estudio de la monodromía de familias de curvas son:

- Proporcionar una vía de acceso para determinar la topología y clasificación del espacio total de la familia, que es una variedad de dimensión superior (así se han clasificado las superficies elípticas y Horikawa clasificó las familias de curvas de género 2, véase [BPV]).
- Paul Seidel ha formulado (véase en [Seidel]), una versión de la conjetura de Simetría Espejo Homológica en la que las subvariedades lagrangianas estudiadas son los ciclos evanescentes de un pincel de Lefschetz. Los pocos casos en que se ha podido comprobar incluyen curvas de género 2 (Véase [Seidel]) y alguna familia de curvas de género 3 (véase [Ku]).

Esta memoria estudia familias algebraicas de curvas de género 4, pero las técnicas analíticas desarrolladas también podrán aplicarse a pinceles de Lefschetz en variedades simplécticas, cuyas fibras tenga género 4.

La elección de familias algebraicas de curva de género 4 se debe a que ya hay estudios precedentes sobre las familias de curvas de género 0, 1, 2.

La teoría clásica de superficies regladas, con Lalonde y McDuff demostrando en [LM], que la clasificación en el caso simpléctico es la misma que en el caso algebrico.

1. La clasificación de familia de curvas de género 1, que es clásica en el caso de familias algebraicas (teoría de superficies elípticas, véase [BPV]) y que fue extendida al caso topológico, en particular simpléctico, por Ron Livné (véase [Mo1]).
2. Las familias de curvas de género 2 fueron estudiadas en el caso algebraico por Horikawa (véase [BPV]), y esta clasificación fue extendida al caso simpléctico por Siebert y Tian ([ST]), D. Auroux ([Au1]) y

recientemente por Auroux-Donalson-Katzarkov-Yotov (véase [ADKY]). El trabajo de todos estos anteriores se basa en que las curvas de género 2 son hiperelípticas, de manera que para controlar la variación de las fibras en una familia basta con controlar la evolución del divisor de ramificación, que tiene grado relativo 6, dentro la familia de $\mathbb{P}_{\mathbb{C}}^1$ recubiertos por las fibras.

Los trabajos de estudio de familias de curvas de género superior son mucho más escasos:

Irene Stankova-Frenkel estudia en [SF] las familias de curvas trigonales, pero desde un enfoque orientado más directamente a la geometría del espacio total, evitando el estudio de la monodromía.

Las técnicas de Siebert-Tian y Aurox pueden extenderse al estudio de curvas hiperelípticas de cualquier género (comunicación personal de los 3 anteriores).

David Moroni, anunció trabajos sobre monodromía geométrica y en el grupo fundamental para familias de curvas hiperelípticas de cualquier género. No ha llegado a publicar resultados. En la extensión de los trabajos de Siebert-Tian y Auroux, y en el trabajo inconcluso de Moroni, el punto clave para estudiar las familias de curvas hiperelípticas de género g es verlas como recubrimientos dobles de $\mathbb{P}_{\mathbb{C}}^1$ de la base (donde forman un divisor de ramificación, de grado relativo $2g + 2$, que describe una monodromía de trenzas).

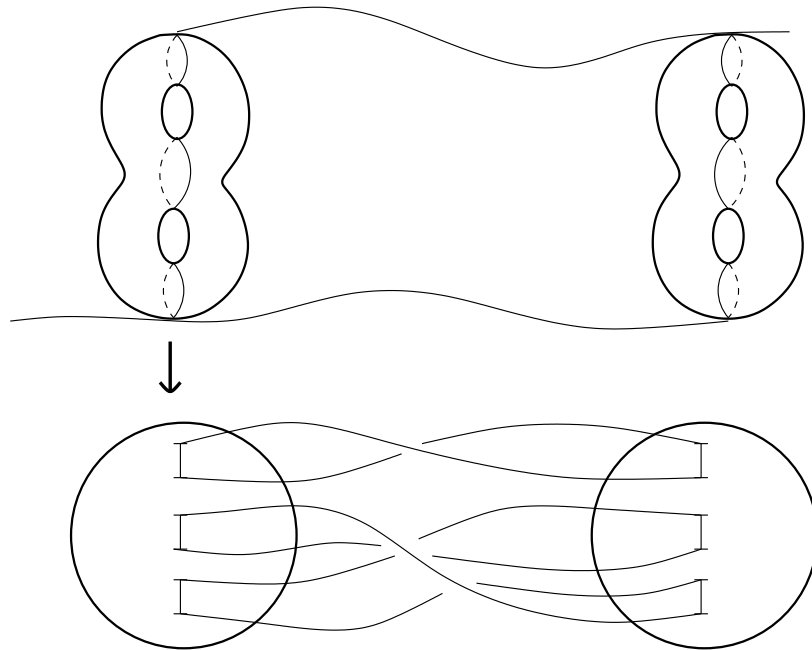


Figura 1: Monodromía del divisor de ramificación en una familia de trenzas.

Por qué en esta memoria continuamos con curvas de género 4 en vez de género 3: Todas las curvas no hiperelípticas de género 3, 4 son trigonales (recubrimientos triples de $\mathbb{P}_{\mathbb{C}}^1$, o sea, admiten serie lineal g_3^1). Pero hay 2 diferencias importantes:

- Las curvas de género 3 admiten una familia 1-paramétrica de estructuras trigonales. Esto hace que para pegar las estructuras trigonales en una familia de curvas haya que recurrir a procedimientos poco naturales, que complican el cálculo posterior (véase [ABV]), en cambio las curvas de género 4 no hiperelípticas admiten solo una o dos estructuras trigonales, de manera que haciendo como máximo un cambio de $2 : 1$, toda familia de curvas de género 4 se convierte en familia de recubrimientos trigonales

$$\begin{array}{ccccc}
 & & Y & \longleftarrow & \tilde{X} & \longrightarrow & X & & \\
 & & & & \searrow & & \downarrow & & \downarrow \\
 \text{familia después} & & & & & & \square & & \text{familia original} \\
 \text{del cambio de base} & & & & & & \tilde{B} & \xrightarrow{2:1} & B
 \end{array}$$

En la fórmula anterior $\tilde{X} \rightarrow \tilde{B}$ es familia de curvas de género 4, $Y \rightarrow \tilde{B}$ es familia de $\mathbb{P}_{\mathbb{C}}^1$ sobre \tilde{B} , y el morfismo $\tilde{X} \rightarrow Y$ es $3 : 1$, con un divisor de ramificación R .

- Las curvas hiperelípticas forman un divisor de codimensión 1 en el espacio de móduli de curvas de género 3. En cambio, forman una subvariedad de codimensión compleja 2 en el espacio de móduli de curvas de género 4. Esto significa que al estudiar una fibración de Lefschetz simpléctica con fibras de género 3 es inevitable que haya fibras hiperelípticas. En cambio, una fibración de Lefschetz simpléctica con base de dimensión real 2 podemos escoger una estructura cuasicompleja compatible genérica, y no habrá ninguna fibra hiperelíptica. Esto simplifica el estudio de tales fibraciones.

0.2. Estructura y contenidos de la Tesis

El capítulo 1, es un repaso de resultados clásicos de la Geometría Algebraica sobre la estructura trigonal de las curvas de género 4, su extensión de curvas a familias complejas, y la aparición en este contexto de la monodromía de trenzas en el cálculo de la monodromía geométrica.

El capítulo 2, contiene los resultados originales de investigación básica: se construye una familia trigonal universal para las curvas de género 4, y se procede a calcular la monodromía en el grupo fundamental de esta familia para una presentación concreta. Como

$$\begin{aligned} \text{Out}(\pi_1(C_4, *)) &= \text{Aut}(\pi_1(C_4, *)) / \sim \\ (\sim &= \text{relación por conjugación global}) \end{aligned}$$

de esta monodromía en el π_1 podemos deducir la monodromía geométrica de las familias trigonales de género 4, siempre que logremos calcular la monodromía de trenzas del divisor de ramificación.

La técnica empleada en este capítulo permite el cálculo de la monodromía en el grupo fundamental para familias de curvas hiperelípticas, y aún para otros recubrimientos de grado bajo. Las dos fases de este cálculo son:

- Calcular la monodromía en el grupo fundamental para la familia universal de este tipo de recubrimientos, a partir de ramificación en esta familia universal.
- Una vez calculada la monodromía de la familia universal, para cualquier familia de recubrimientos

$$\begin{array}{ccc} \tilde{X} & \longrightarrow & Y = \tilde{B} \times \mathbb{P}^1 \\ \downarrow & \swarrow & \\ \tilde{B} & & \end{array}$$

Hay que identificar el divisor de ramificación del recubrimiento $\tilde{X} \longrightarrow Y = \tilde{B} \times \mathbb{P}^1$, $R \subset \tilde{B} \times \mathbb{P}^1$, y calcular la monodromía de trenzas de este divisor R sobre la base \tilde{B} (véase figura 1).

El cálculo efectivo del divisor de ramificación R y su monodromía de trenzas para familias de curvas algebraicas de género 4 es el objetivo del resto de la tesis, que se dedica a aspectos computacionales.

En el capítulo 3, discutimos estas técnicas de cálculo y mostramos su implementación:

En primer lugar es necesario un **Cálculo Algebraico** para, dada una familia trigonal de curvas de género 4

$$\begin{array}{ccc} \tilde{E} & \longrightarrow & \tilde{B} \times \mathbb{P}^1 \supset R \\ \downarrow & \swarrow & \\ \tilde{B} & & \end{array}$$

identificar (o sea, hallar su ecuación), el divisor de ramificación $R \subset \tilde{B} \times \mathbb{P}^1_{\mathbb{C}}$.

Este cálculo se hace por medio del programa Singular. Se ha desarrollado una librería dentro de Singular para el estudio de curvas de género 4, presentadas en forma canónica debido a la complejidad del cálculo esta identificación de r es posible actualmente para familias trigonales en las que la base \tilde{B} de la familia sea una recta proyectiva o una cónica.

Una vez identificada la ecuación del divisor de ramificación $R \subset \tilde{B} \times \mathbb{P}^1_{\mathbb{C}}$, la siguiente fase es el **Cálculo Analítico** de la monodromía de trenzas de este divisor, si se obtiene este cálculo podemos usar la tabla de monodromías de la familia universal del capítulo 2, y obtener la monodromía geométrica para nuestra familia trigonal $\tilde{E} \rightarrow \tilde{B}$.

También podemos obtener la monodromía en el grupo fundamental de las fibras para cualquier familia con sección

$$\begin{array}{ccc} \tilde{E} & & \{\pi_1(\tilde{E}_s, \sigma(s))\}_{s \in \tilde{B}} \\ \uparrow \sigma & & \downarrow \\ \tilde{B} & & \tilde{B} \end{array}$$

si calculamos la monodromía de las “trenzas enriquecidas” formadas por el divisor de ramificación R y la sección punto base $\sigma : \tilde{B} \rightarrow \tilde{E}$.

Aunque hay modelos locales clásicos (véase [Za]), e incluso globales módulo conjugación analítica en el caso en que R es liso (véase [Mo1]), como el divisor R no suele ser liso y la teoría de Moishezon no es muy efectiva en el cálculo de la conjugación a su modelo teórico, optamos por el cálculo de la monodromía de trenzas por un procedimiento más elemental: si el divisor R tiene una ecuación local

$$R : f(s, z) = 0 \rightarrow f(s, z)$$

en coordenadas locales (s, z) en $\tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$, al parametrizar localmente una rama de R en función de la coordenada base, implica que

$$\frac{\partial}{\partial s} f(s, z(s)) = \frac{\partial f}{\partial s} + \frac{\partial f}{\partial z} \cdot \frac{\partial z}{\partial s} = 0$$

de donde deducimos que podemos hallar explícitamente la trayectoria de la rama $z(s)$ del divisor R cuando s describe un camino en \tilde{B} integrando la ecuación diferencial ordinaria

$$\frac{\partial z}{\partial s} = - \frac{\frac{\partial f}{\partial s}}{\frac{\partial f}{\partial z}}$$

Realizamos esta integración y cálculo posterior de la monodromía de trenzas a partir de la trayectoria de las ramas de R , por medio de una librería de funciones para MATLAB. Las razones para el uso de MATLAB son:

- Permite calcular con números complejos de manera tan simple como con números reales.
- Ofrece muchas funciones de alto nivel para los cálculos previos y posteriores a la integración de la edo:
 - determinación de descomposiciones celulares de Voronoi, y otras funciones de grafos que permiten hallar presentaciones de $\pi_1(\tilde{B}, s_0)$ óptimas para el cálculo de la monodromía de trenzas.
 - facilidades de post-proceso y representación gráfica.

Sin embargo, no se ha podido aprovechar ninguno de los integradores de edos existentes en MATLAB, porque el problema de integración de las ramas del divisor R por separado y con garantía de precisión se ha mostrado demasiado duro para ellos.

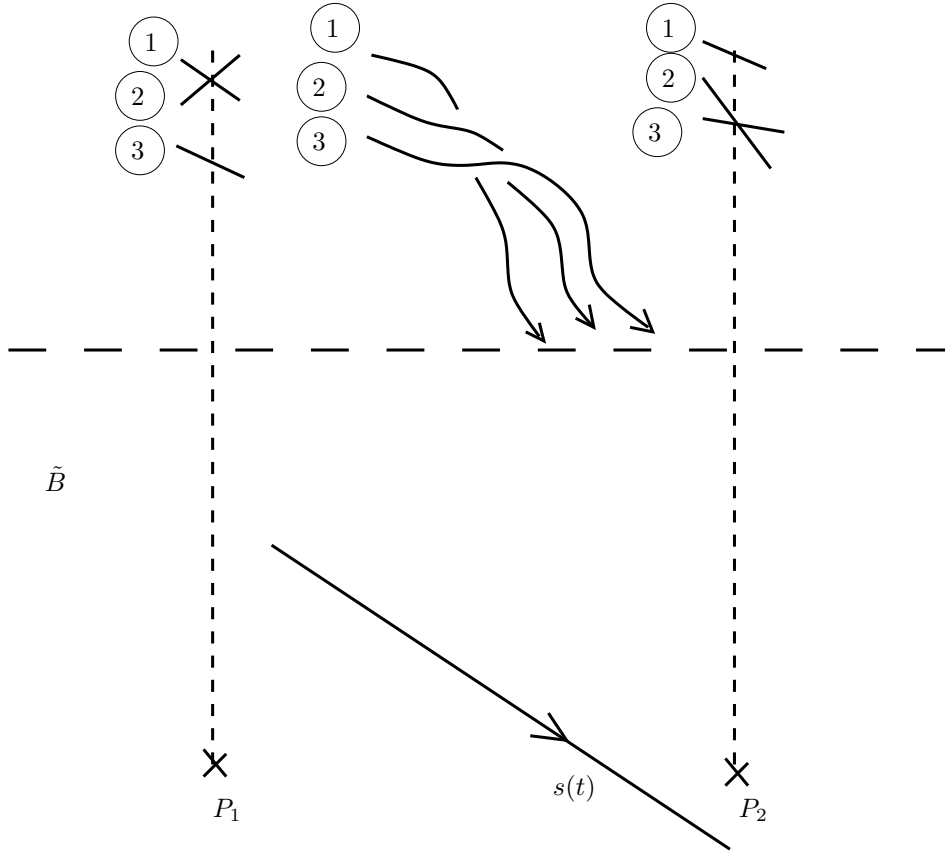


Figura 2: Cuando el camino de integración separa a 2 valores singulares muy próximos P_1, P_2 , las ramas de R se aproximan mucho y esto dificulta e imposibilita el trabajo del integrador numérico.

Debido a la dificultad creada por la gran proximidad de las ramas del divisor de ramificación R al pasar entre valores de ramificación próximos, que hace que los integradores *ode*{ xy } de MATLAB se confundan y salten de una rama a otra, ha sido necesario desarrollar una variante de integrador de Runge-Kutta, que usa un método RK4 de paso variable para avanzar, y controla la longitud de paso y la garantía del cálculo a través de la integral primera

$$f(s, z(s)) = 0$$

Este integrador **rk4int1avect** permite integrar la edo de todas las ramas del divisor R , y ofrecer garantía de corrección del cálculo, pero a un precio: el cálculo se ralentiza tanto que jamás acaba en familias con valores de ramificación próximos. Debido a esto, la versión actual de la librería de funciones de MATLAB puede calcular monodromía de trenzas para curvas de grado bajo $C \subset \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$ (funciona consistentemente hasta grado 6, esporádicamente hasta grado 9), pero es por el momento incapaz de calcular las monodromías de trenzas de los divisores $R \subset \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$ de grado relativo 12 que aparecen en las familias de curvas trigonales de género 4.

El capítulo 4, muestra algunos ejemplos para ilustrar la capacidad de cálculo de las librerías desarrolladas:

- La librería en Singular llamada **genero4.lib**, es capaz de hallar la ecuación del divisor de ramificación R para familias trigonales de género 4 de interés geométrico: la familia de Bring, la familia de una superficie K3 de género 4 necesaria para la conjetura de Simetría Espejo de P. Seidel.
- La librería de MATLAB por el momento sólo calcula monodromías de trenzas para curvas $C \subset \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$ de grado bajo. Se muestra con curvas $C \subset \mathbb{P}_{\mathbb{C}}^2$ de grado hasta 6, a modo de ejemplos académicos.

0.3. Resultados originales de la Tesis

En el capítulo 1 se define la familia universal de Clebsch-Hurwitz para recubrimientos de $\mathbb{P}_{\mathbb{C}}^1$ de grado 3 y género 4, que es una extensión no inmediata de la forma normal de Clebsch-Hurwitz del recubrimiento trigonal de una curva a una familia.

En el teorema 5 de este capítulo se establece la propiedad de universalidad de esta familia.

En el teorema 6 de este capítulo se calcula la monodromía en el grupo fundamental para esta familia universal. El cálculo es módulo una conjetura que ha quedado abierta sobre la acción de grupo de trenzas en estas familias. Pero aunque la conjetura resulte ser falsa el cálculo hecho en la demostración del teorema 6 permanece cierto, y bastaría con añadirles la monodromía de los nuevos elementos del estabilizador para completar el teorema.

En el capítulo 3, presentamos la librería para Singular **genero4.lib**, que permite estudiar la estructura geométrica de curvas de género 4 a partir de sus ecuaciones canónicas, e identifica el divisor de ramificación de la estructura trigonal en familias.

También en el capítulo 3, presentamos una librería para MATLAB que permite el cálculo de la monodromía de trenzas asociada a una curva $C \subset \mathbb{C}^2$. La librería permite escoger generadores óptimos del grupo fundamental sobre los que calcular la monodromía de trenzas usando una descomposición celular de Voronoi, después halla la trayectoria de las ramas del divisor de ramificación de la familia sobre estos caminos, y deduce las trenzas a partir de la evolución de estas posiciones.

Capítulo 1

Fundamentos de la Tesis

1.1. Estructura Trigonal en Curvas de Género 4

Sea E una curva de género 4, no hiperelíptica, entonces por el teorema de Riemann-Roch (ver [Gr]) hay un embedding canónico

$$E \hookrightarrow \mathbb{P}_{\mathbb{C}}^3$$

tal que $E = Q \cap C$, donde $Q \subset \mathbb{P}_{\mathbb{C}}^3$ y $C \subset \mathbb{P}^3$.

Para la cuádrica Q sólo hay dos posibilidades:

1. Que sea una cuádrica lisa, es decir existen coordenadas proyectivas tales que

$$Q = \{X^2 + Y^2 - Z^2 - T^2 = 0\}$$

o bien,

2. Un cono, con un punto singular. Es decir, existen coordenadas proyectivas tal que

$$Q = \{X^2 + Y^2 - Z^2 = 0\}$$

En una familia $Q_\alpha \cap C_\alpha$ que depende del parámetro $\alpha \in \mathbb{P}_{\mathbb{C}}^k$ (o $\alpha \in \mathbb{C}^k$), los conos vienen dados por la ecuación $\det(Q) = 0$, por lo que forman un cerrado de Zariski en el espacio de parámetros en general de codimensión 1.

Estudiamos en primer lugar las familias sin conos, donde p_s y q_s son polinomios homogéneos de grado 2 y 3 respectivamente. Como suponemos que las cuádricas

$$\{p_s(X : Y : Z : T) = 0\}$$

son todas lisas, existe un cambio de coordenadas proyectivas

$$\Phi_s(X : Y : Z : T) = [\tilde{X} : \tilde{Y} : \tilde{Z} : \tilde{T}]$$

que lleva la cuádrica Q_s a tener coordenadas

$$\tilde{X}^2 + \tilde{Y}^2 - \tilde{Z}^2 - \tilde{T}^2 = 0$$

independientes de s .

$$\begin{array}{ccc} Q \cap C & \hookrightarrow & \mathbb{P}_{\mathbb{C}}^3 \times \mathbb{C} \\ & \searrow \pi|_{Q \cap C} & \downarrow \pi \\ & & \mathbb{C} \ni s \end{array}$$

$$\pi|_{Q \cap C}^{-1}(s) = \{p_s(X : Y : Z : T) = 0\} \cap \{q_s(X : Y : Z : T) = 0\} \subset \mathbb{P}_{\mathbb{C}}^3 \times \{s\}$$

Aplicamos a $\mathbb{P}_{\mathbb{C}}^3 \times \mathbb{C}$ el cambio de coordenadas dado por $[\Phi_s, Id]_{s \in \mathbb{C}}$, y conseguimos nuevas coordenadas proyectivas en cada $\mathbb{P}_{\mathbb{C}}^3 \times \{s\}$, de manera que

$$Q_s = \{\tilde{X}^2 + \tilde{Y}^2 - \tilde{Z}^2 - \tilde{T}^2 = 0\}$$

Para escribir menos, llamaremos $[X : Y : Z : T]$ a las nuevas coordenadas, así hemos conseguido en el diagrama anterior, Q está definida por la ecuación homogénea de grado 2 en X, Y, Z, T de la forma

$$\{X^2 + Y^2 - Z^2 - T^2 = 0\}$$

en $\mathbb{P}_{\mathbb{C}}^3$ y C definida por la ecuación homogénea de grado 3 en X, Y, Z, T con

$$\{q_s(X : Y : Z : T) = 0\}$$

que corresponde a una superficie cúbica que varía con el parámetro s .

Ejemplo 1. Sea C_s una superficie cúbica definida en el espacio proyectivo $\mathbb{P}_{\mathbb{C}}^3 \times \{s\}$, dada por el polinomio $q_s(X : Y : Z : T) = 0$, es una superficie cúbica que depende de s , que puede ser uno de la familia de polinomios homogéneos definidos por:

$$\{q_s = 0\} : \{X^3 + Y^3 - Z^3 - T^3 - s(2-s)XYZ + 7s^2X^2Y + 13s^5Y^2Z + XZ^2 = 0\}$$

$$X^3 - sXYZ + 2XT^2 - s(1-s)YZT + s^7T^3 - s^2XZT = 0$$

Por el teorema de Riemann-Roch [Gr], cuando $Q_s \cap C_s$ es lisa (en todo $\mathbb{P}_{\mathbb{C}}^3$), es una curva E_s de género 4, la cuádrica Q_s tiene dos reglajes, que da lugar a dos estructuras trigonales en E_s ; cada reglaje es una familia de rectas que recubre a Q_s . Además cumplen las siguientes propiedades:

- Dos rectas del mismo reglaje se cruzan.
- Una recta de un reglaje corta a todas las rectas del otro reglaje.
- Por todo punto de la cuádrica Q_s , pasa una recta de cada reglaje.

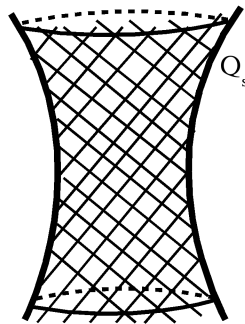


Figura 1.1: Cuádrica Q_s

Las estructuras trigonales en $E_s = Q_s \cap C_s$, provienen de proyectar la cuádrica Q_s sobre una recta ℓ , de uno de los reglajes siguiendo la dirección del otro reglaje.

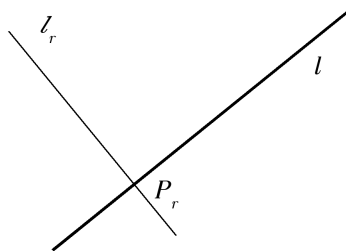


Figura 1.2: $P_r \in \ell_r \cap \ell$

Para cada punto $P_r \in \ell$, con coordenada r en ℓ , tenemos una recta ℓ_r del otro reglaje que pasa por el punto P_r . Parametrizando la recta ℓ_r con coordenada α ;

$$\ell_r : P_r + \alpha v_r$$

donde $\alpha \in \mathbb{C}$ (o bien $\mathbb{P}_{\mathbb{C}}^1$), las coordenadas (r, α) son coordenadas válidas en la cuádrica Q_s , ya que Q_s menos sus puntos del infinito, vive en el espacio afín $\mathbb{A}_{\mathbb{C}}^3$ con coordenadas x, y, z , pero podemos parametrizar Q_s por las coordenadas afines α y r .

$E_s = Q_s \cap C_s$ tiene un morfismo de grado 3 sobre $\mathbb{P}_{\mathbb{C}}^1$, definido por la proyección de Q_s sobre la recta ℓ en la dirección de las ℓ_r .

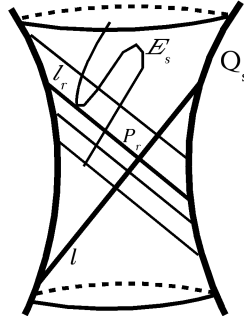


Figura 1.3: $E_s = Q_s \cap C_s$

Así obtenemos el morfismo:

$$E_s \xrightarrow{3:1} \ell \cong \mathbb{P}_{\mathbb{C}}^1$$

Las ramificaciones posibles son: P_r tal que la recta ℓ_r corta a C_s en un punto doble y uno simple, o bien, en un punto triple.

Los valores de s tal que $E_s \rightarrow \ell$ tiene algún punto triple forman un cerrado de Zariski en $\{s \in \mathbb{C}\}$.

La Fórmula de Hurwitz aplicada sobre un valor de $s \in \mathbb{C}$, donde no hay puntos triples implica que el morfismo $E_s \rightarrow \mathbb{P}_{\mathbb{C}}^1$ tiene 12 puntos dobles.

Lo que significa que estos 12 valores de r (12 puntos $P_r \in \ell$) definen rectas.

$$P_r + \alpha v_r = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Llamaremos curva de ramificación a R con $\alpha \in \mathbb{C}$ o $\mathbb{P}_{\mathbb{C}}^1$ tal que al restringir $Q_s(X : Y : Z : 1)$ a una recta aparece una ecuación de grado 3 en α con un cero doble. Al variar $s \in \mathbb{C}$ los 12 valores de r describen una curva en $R \subset \ell \times \mathbb{C}$.

1.2. Forma normal de Clebsch del recubrimiento trigonal

Denotamos por P_w al subconjunto abierto del producto simétrico w -ésimo de \mathbb{P}^1 que consiste en w -tuplas no ordenadas de puntos distintos en \mathbb{P}^1 . Para cualquier $A \in P_w$, tenemos

$$H(d, A)$$

el conjunto de clases de equivalencia de d -hojas de recubrimientos simplemente ramificados

$$f : C \longrightarrow \mathbb{P}^1 \quad (1.1)$$

tales que la ramificación de f es A . Recordemos que el recubrimiento (1.1) se dice que es equivalente a un recubrimiento $f' : C' \longrightarrow \mathbb{P}^1$ si existe un isomorfismo $\phi : C \longrightarrow C'$ tal que $f' \circ \phi = f$. Denotaremos por $[f]$ la equivalencia clase de f .

Recordemos el teorema fundamental de Riemann de existencia . Para ello, tenemos que introducir algunas definiciones y notación. Si G y H dos grupos,

$$Hom^{ext}(G, H) = Hom(G, H) / \sim .$$

donde $\lambda \sim \mu$ si existe $h \in H$ tal que $\mu = h\lambda h^{-1}$. Denotaremos por \mathfrak{S}_d el grupo simétrico en d letras. Ahora vamos a $A = \{a_1, \dots, a_w\}$ es un elemento de P_w y elegir un punto base $y \in \mathbb{P}^1 - A$. Se define un sistema estándar de generadores $\sigma_1, \dots, \sigma_w$ de $\pi(\mathbb{P}^1 \setminus A, y)$ como vemos en la siguiente figura: 1.4

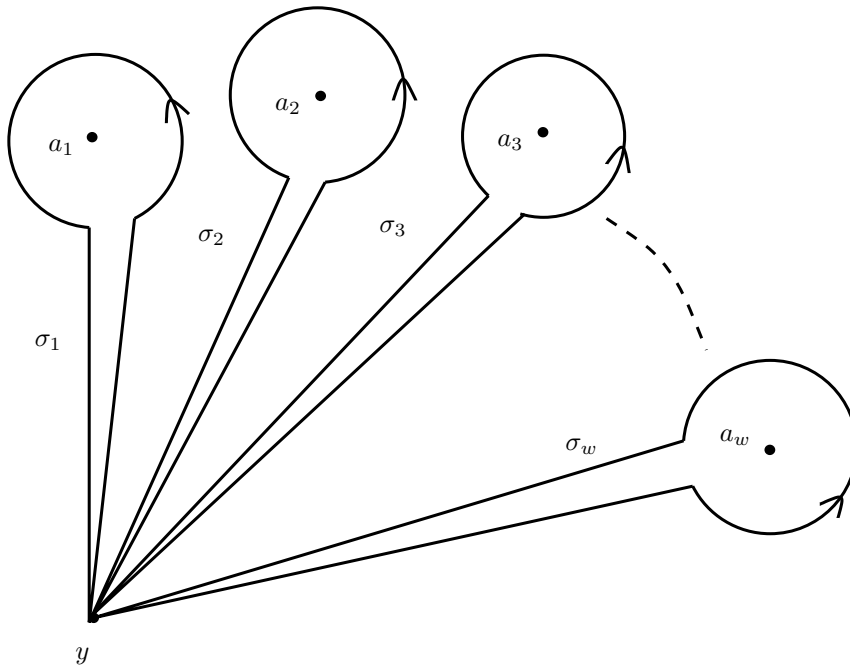


Figura 1.4: Sistema estándar de generadores

Se define la representación de monodromía

$$\phi(f) : \pi_1(\mathbb{P}^1 \setminus A, y) \longrightarrow [\text{grupo de permutaciones de } f^{-1}(y)]$$

mediante la asignación a cada lazo σ en $\mathbb{P}^1 \setminus A$ con un punto base en y para la permutación de $f^{-1}(y)$ que se obtiene asociando a cada punto x de $f^{-1}(y)$ el punto final del levantamiento único σ^{-1} que tiene un punto inicial en x . La numeración de los puntos de $f^{-1}(y)$ da una identificación entre el grupo de permutaciones de $f^{-1}(y)$ y \mathfrak{S}_d , canónica hasta automorfismos internos de \mathfrak{S}_d . Por lo tanto, la representación de monodromía induce una aplicación bien definida

$$\phi : H(d, A) \longrightarrow \text{Hom}^{\text{ext}}(\pi_1(\mathbb{P}^1 \setminus A, y), \mathfrak{S}_d), \quad (1.2)$$

obtenidas por la asociación a todos los elementos $[f]$ de $H(d, A)$ la clase de $\phi(f)$. Comprendido esto, podemos establecer la siguiente instancia especial de Teorema de Riemann de existencia.

Teorema 1. (*Teorema de Riemann de existencia*). *La aplicación (1.2) es inyectiva. Por otra parte, la imagen de la figura 1.4 se compone de las clases que son inducidos por representaciones irreducibles ξ tal que*

$$t_i = \xi(\sigma_i), \quad i = 1, \dots, w,$$

es una transposición, y $\prod t_i = 1$.

Que t_i es una transposición es un reflejo del hecho de que se trata de recubrimientos simples. En conclusión, a través de ϕ , podemos identificar $H(d, A)$ con el conjunto

$$\mathcal{G}_{d,w} = \left[\begin{array}{l} \text{conjunto de clases de conjugación } \{t_1, \dots, t_w\} \text{ de} \\ \text{w-tuplas de transposiciones tal que } \prod t_i = 1 \text{ y } t_1, \dots, t_w \\ \text{generan un subgrupo transitivo de } \mathfrak{S}_d \end{array} \right]$$

Consideremos el conjunto

$$H(d, w) = \prod_{A \in \mathcal{P}_w} H(d, A),$$

Vamos a llamar a $H(d, w)$ el espacio de Hurwitz de tipo (d, w) . Se desprende de la construcción que hay, sobre el espacio de Hurwitz, una familia d -hojas de recubrimiento simple, en el sentido siguiente.

Existe una variedad compleja X y transformaciones lisas

$$\begin{array}{ccc} X & \xrightarrow{F} & \mathbb{P}^1 \\ \psi \downarrow & & \\ H(d, w) & & \end{array}$$

de tal manera que, para cada $s \in H(d, w)$, la fibra $\psi^{-1}(s) = X_s$ es una curva lisa.

Por otra parte,

$$F_s = F|_{X_s} : X_s \longrightarrow \mathbb{P}^1$$

es un recubrimiento de d -hojas con ramificación simple que cubre de tal manera que $[F_s] = s$. Es evidente que, por la fórmula de Riemann-Hurwitz, X_s es una curva de género g , donde

$$w = 2g + 2d - 2.$$

En esta memoria vamos a discutir el caso de recubrimientos de grado $d = 3$ (trigonales) de curvas de género 4 a $\mathbb{P}_{\mathbb{C}}^1$. Por la fórmula de Riemann-Hurwitz tales recubrimientos tienen

$$w = 2 \cdot 4 + 2 \cdot 3 - 2 = 12$$

puntos de ramificación.

Sea $C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$ morfismo trigonal con 12 puntos de ramificación simple, se instala un sistema de caminos cualquiera. Los sistemas estándar de generadores y monodromía definida por cada uno de ellos que hemos discutido de manera general anteriormente. Para el caso de estos recubrimientos trigonales de género 4 definimos:

Definición 1. *Llamamos Sistema estándar de generadores, o simplemente, Sistema de Caminos al conjunto de lazos con punto base z_0 , disjuntos dos a dos, formados por caminos que van de z_0 a cada valor de ramificación z_1, \dots, z_{12} , un lazo que da la vuelta al punto de ramificación en sentido directo y regresan a z_0 , por donde han venido.*

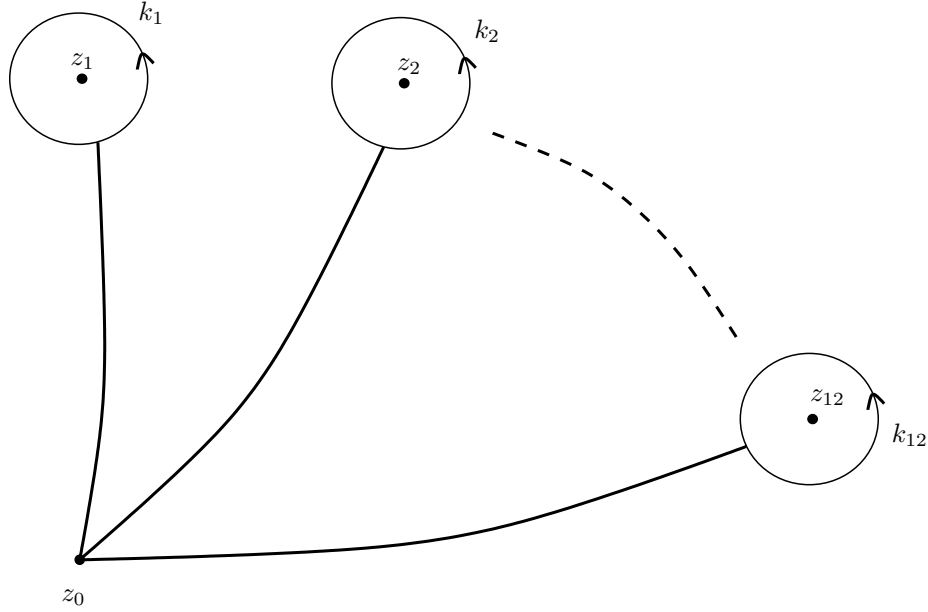


Figura 1.5: Sistema de Caminos

Definición 2. La monodromía del recubrimiento trigonal $C_4 \xrightarrow{p} \mathbb{P}_{\mathbb{C}}^1$, $\rho : \pi_1(\mathbb{P}_{\mathbb{C}}^1 \setminus \{z_1, \dots, z_{12}\}, z_0) \rightarrow \mathcal{S}_3 = \{\text{permutaciones de } \{1, 2, 3\}\}$. se define cada lazo k que empieza y acaba en z_0 , al intercambio de las 3 hojas de p que sucede al desplazarlas sobre k .

Las monodromías $\rho(k_1), \dots, \rho(k_{12})$ son transposiciones de dos elementos $\rho(k_1) = (a, b), \rho(k_2) = (a_2, b_2), \dots, \rho(k_{12}) = (a_{12}, b_{12})$, donde $a_i, b_i \in 1, 2, 3$ sólo han de cumplir que:

$$\rho(k_1) \cdot \rho(k_2) \cdot \dots \cdot \rho(k_{12}) = Id$$

(Rodear a todos los puntos críticos es homótopo al camino constante).

El sistema de caminos para un divisor de ramificación $\{z_1, \dots, z_{12}\}$ y un punto base z_0 dado no es único. Hurwitz demostró que se pueden obtener todos los sistemas de caminos y monodromías posibles mediante la composición de movimientos de Hurwitz:

Definición 3. *Movimiento de Hurwitz en un sistema de caminos; es el sistema de caminos que se obtiene al tomar dos lazos consecutivos en el sistema de caminos, hacer que los puntos de ramificación se intercambien siguiendo su generador del grupo de trenzas (véase sección 1.5) y moviendo de manera continua los lazos que los rodean, como se indica en la figura 1.6.*

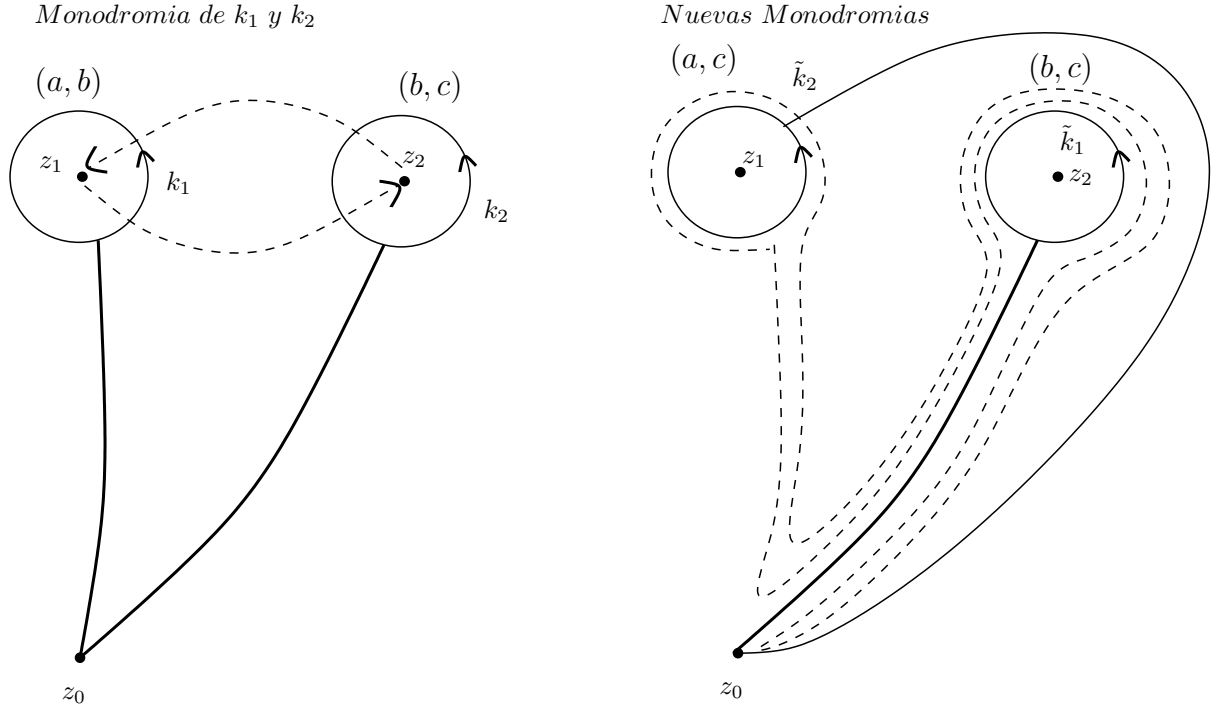


Figura 1.6: Movimiento de Hurwitz

La figura 1.6 demuestra el siguiente lema:

Lema 1. *Sea una parte del sistema de caminos en un recubrimiento trigonal, tal que:*

Monodromía de k_1 : $\rho(k_1) = (a, b)$

Monodromía de k_2 : $\rho(k_2) = (b, c)$

Al hacer el movimiento de Hurwitz en dos valores de ramificación indicados en la figura 1.6, los caminos \tilde{k}_1, \tilde{k}_2 son:

$$\tilde{k}_1 = k_2,$$

$$\tilde{k}_2 = k_2 \cdot k_1 \cdot k_2^{-1}$$

y las nuevas monodromías en el grupo simétrico son:

- $\rho(\tilde{k}_1) = \rho(k_2) = (b, c)$

- $\rho(\tilde{k}_2) = \rho(k_2) \cdot \rho(k_1) \cdot \rho(k_2^{-1}) = (a, c).$

Teorema 2. (Clebsch, Hurwitz) ([Hur], ver [ACG2], pag. 859).

Aplicando sucesivos movimientos de Hurwitz al sistema de caminos original escogido para el recubrimiento trigonal $C_4 \rightarrow \mathbb{P}_{\mathbb{C}}^1$, podemos acabar obteniendo un sistema de caminos $\{k_1^{\#}, k_2^{\#}, \dots, k_{12}^{\#}\}$ en que las monodromías sean:

$$\rho(k_1^{\#}) = (a, b), \rho(k_2^{\#}) = (a, b), \rho(k_3^{\#}) = (b, c), \dots, \rho(k_{12}^{\#}) = (b, c)$$

donde $\{a, b, c\} = \{1, 2, 3\}$ en algún orden.

Cambiando después la numeración asignada a las hojas, es decir:

- La hoja **a** pasa a ser la hoja 1
- La hoja **b** pasa a ser la hoja 2
- La hoja **c** pasa a ser la hoja 3

La monodromía pasa a ser:

$$\rho(k_1^{\#}) = (1, 2), \rho(k_2^{\#}) = (1, 2), \rho(k_3^{\#}) = (2, 3), \dots, \rho(k_{12}^{\#}) = (2, 3)$$

1.3. Familia de Curvas de Género 4

Sea $E_s \rightarrow B$ una familia de curvas de género 4, no hiperelípticas sobre una base B cuasiproyectiva.

$$\begin{array}{ccc} E_s & \subset & E \\ \downarrow & & \downarrow \\ s & \in & B \end{array}$$

Como un E_s genérico tiene dos estructuras trigonales, hay que hacer un cambio de base 2 : 1, para poder pegarlas.

$$\begin{array}{ccc} \tilde{E} & \longrightarrow & E \\ \downarrow & & \downarrow \\ \tilde{B} & \xrightarrow{2:1} & B \end{array}$$

Ramificado en $s \in B$ tal que $\det(Q_s) = 0$, y $\tilde{B} = \{(s, \Phi) \mid s \in B, \Phi = \text{estructura trigonal en } E_s\}$.

La nueva familia permite pegar los recubrimientos trigonales de las fibras y queda un morfismo.

$$\begin{array}{ccc} & & \tilde{E} \\ & \swarrow & \downarrow \\ Y & \longrightarrow & \tilde{B} \end{array}$$

Donde Y es la familia de $\mathbb{P}_{\mathbb{C}}^1$ sobre \tilde{B} , de manera que,

$$\Phi : \tilde{E}_s \rightarrow Y_s \cong \mathbb{P}_{\mathbb{C}}^1$$

es un recubrimiento trigonal de, E_s a $\mathbb{P}_{\mathbb{C}}^1$. Si $\dim B = 1$, \tilde{B} es una curva e Y es una superficie reglada.

La superficie reglada $Y \rightarrow \tilde{B}$ trivializa sobre la base \tilde{B} porque ésta es cuasiproyectiva. Por lo tanto $Y \cong \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$. De esta manera, para cada $s \in \tilde{B}$ obtenemos un divisor efectivo de grado 12, $R_s \subset \mathbb{P}_{\mathbb{C}}^1$, formado por los 12 puntos de ramificación del morfismo trigonal $E_s \rightarrow \mathbb{P}_{\mathbb{C}}^1$.

Estos puntos definen el **divisor de ramificación** $R \subset Y = \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$, y una **aplicación clasificante**

$$\Phi : \tilde{B} \rightarrow (\mathbb{P}^{12} \setminus \Delta)$$

donde el espacio de llegada son los divisores efectivos de grado 12 en $\mathbb{P}_{\mathbb{C}}^1$, y Δ es la diagonal generalizada, de manera que $\mathbb{P}^{12} \setminus \Delta$ son las 12-plas de puntos diferentes no ordenados en \mathbb{P}^1 .

Para facilitar el ulterior cálculo informático, restringimos la familia original $E \rightarrow B$ para eliminar las fibras que tengan punto de ramificación en el infinito. Entonces obtenemos aplicación clasificante

$$\Phi : \tilde{B} \rightarrow \mathbb{C}^{12} \setminus \Delta$$

Y si restringimos nuevamente la familia $E \rightarrow B$ para retirar las fibras con ramificación no simple, obtenemos a veces el vacío, pero en el caso genérico, una nueva familia que también denotaremos $E \rightarrow B$ y que define la versión que usaremos de la aplicación clasificante:

$$\Phi : \tilde{B} \rightarrow \mathbb{C}^{12} \setminus \Delta$$

Aquí $\Delta =$ diagonal generalizada de \mathbb{C}^{12} y $\mathbb{C}^{12} \setminus \Delta = \{12\text{-plas de puntos diferentes no ordenados en } \mathbb{C}\}$. Como estos espacios son asféricos, la clase de homotopía de la aplicación clasificante queda determinada por el morfismo de grupos fundamentales.

$$\pi_1(\Phi) : \pi_1(\tilde{B}, s_0) \rightarrow \pi_1(\mathbb{C}^{12} \setminus \Delta)$$

Como explicaremos en la sección 1.5 el grupo fundamental de llegada es el grupo de trenzas de Artin.

El grupo $\pi_1(\tilde{B}, s_0)$ es libre, finito generado, por ejemplo por un sistema de caminos como los usados en la sección 1.2.

La aplicación $\pi_1(\Phi)$ es una monodromía de trenzas.

1.4. Monodromía en el Grupo Fundamental

En la anterior familia

$$\begin{array}{ccc} & \tilde{E} & \\ \Phi \swarrow & \downarrow & \\ Y & \xrightarrow{p} & \tilde{B} \end{array}$$

Al mover el punto $s \in \tilde{B}$ podemos mover los recubrimientos trigonales

$$\tilde{E}_s \xrightarrow{\Phi_s} \mathbb{P}_{\mathbb{C}}^1$$

Si movemos Φ_s sobre s que describe un lazo $s = \gamma(t)$ en \tilde{B} , volvemos a un difeomorfismo de

$$\tilde{E}_{\gamma(0)} \xrightarrow{\Phi_{\gamma(0)}} \mathbb{P}_{\mathbb{C}}^1$$

en si mismo, este morfismo es *la monodromía geométrica* (ver [Mo1]) a lo largo del camino γ . Está definida módulo isotopía.

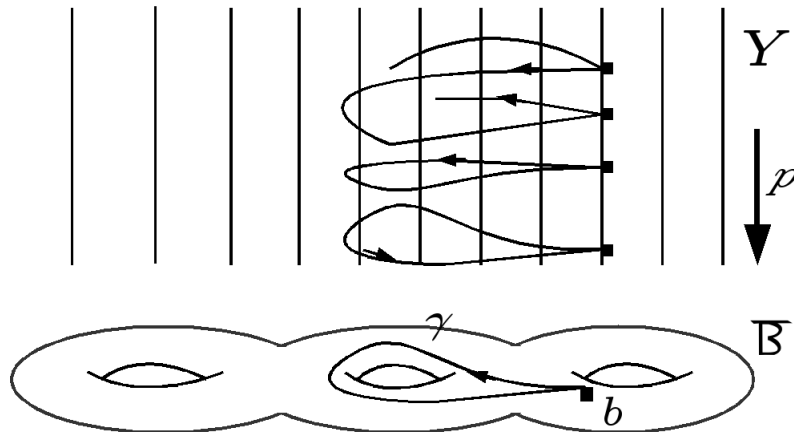


Figura 1.7: Monodromía a lo largo del camino γ

Los 12 puntos de ramificación del morfismo trigonal de $\tilde{E}_{\gamma(t)}$ en $\mathbb{P}_{\mathbb{C}}^1$, describen una trenza en $\mathbb{P}_{\mathbb{C}}^1$, al recorrer todo el camino $\gamma(t)$, esta trenza determina toda la monodromía de la familia.

Sea S_{12} el grupo de permutaciones de 12 elementos. En cada $Y_b = p^{-1}(b) \cong \mathbb{P}_{\mathbb{C}}^1$ hay 12 valores de ramificación distinguidos $\{r_1, \dots, r_{12}\}$ (los de $E_b \xrightarrow{3:1} Y_b$).

Al mover el punto b siguiendo un lazo en \tilde{B} , estos 12 valores regresan sobre ellos mismos, intercambiados. Obtenemos así una monodromía,

$$p : \pi(\tilde{B}, b) \longrightarrow S_{12}$$

Por otra parte al mover el punto $b \in \tilde{B}$ siguiendo un lazo $\gamma : [0, 1] \rightarrow \tilde{B}$ tal que $\gamma(0) = b = \gamma(1)$ los 12 puntos de ramificación del morfismo $E_{\gamma(t)} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ se mueven en $p^{-1}(\gamma([0, 1])) \cong \mathbb{P}_{\mathbb{C}}^1 \times \gamma([0, 1])$ y describen caminos $r_1(t), \dots, r_{12}(t)$ en $\mathbb{P}_{\mathbb{C}}^1$. Estos 12 caminos forman una trenza de 12 valores en la esfera de Riemann $\mathbb{P}_{\mathbb{C}}^1$. Entonces tenemos un movimiento.

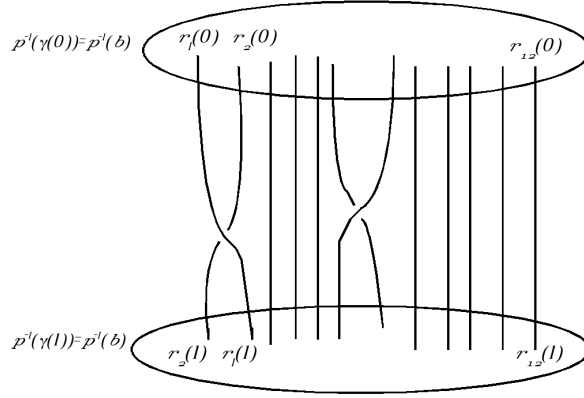


Figura 1.8: trenza de 12 valores

Si restringimos \tilde{B} a los valores de b en los que las trenzas se mantienen apartadas del infinito, obtendremos una representación de monodromía,

$$p : \pi(\tilde{B}, b) \rightarrow B_{12}$$

donde B_{12} es el grupo de trenzas en 12 hebras de \mathbb{C} , subgrupo del grupo fundamental de automorfismos exteriores del grupo libre en 12 generadores $F_{12} = \pi_1(\mathbb{C} \setminus \{12 \text{ puntos}\})$.

Y como $E_{\gamma(t)} \rightarrow p^{-1}(\gamma(t)) \cong \mathbb{P}_{\mathbb{C}}^1$ es un recubrimiento 3 : 1 ramificado en $r_1(t), \dots, r_{12}(t)$, a partir de la trenza que describe el movimiento de $r_1(t), \dots, r_{12}(t)$, podemos deducir cómo varía la curva $E_{\gamma(t)}$ y su grupo fundamental para $t \in [0, 1]$ (ver [Mo2], [Sti], [Za]).

1.5. El Grupo de Trenzadas

Calculamos la monodromía en el grupo fundamental de E_s a partir del recubrimiento $3 : 1$ de $E \rightarrow \mathbb{P}_{\mathbb{C}}^1$, con 12 puntos de ramificación y el grupo B_{12} de trenzadas de 12 hebras en \mathbb{C} .

El grupo de trenzadas de 12 hebras en \mathbb{C} es:

Definición 4. Sea el grupo presentado con generadores $\sigma_1, \sigma_2, \dots, \sigma_{n-1}$, para todo $n \geq 2$, sujetos a las siguientes relaciones:

$$\sigma_i \sigma_j = \sigma_j \sigma_i, \quad |i - j| > 1,$$

$$\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j, \quad |i - j| = 1,$$

Definición 5. Sea el espacio de configuración de n puntos en el plano complejo, para todo $n \geq 2$:

$$\mathbb{C}^n \setminus \Delta = \{(z_1, \dots, z_n) \in \mathbb{C} \times \dots \times \mathbb{C} \mid z_i \neq z_j, \text{ para } i \neq j\}$$

Un punto en $\mathbb{C}^n \setminus \Delta$, es denotado por un vector $\vec{z} = (z_1, \dots, z_n)$. El grupo simétrico actúa libremente en $\mathbb{C}^n \setminus \Delta$ permutando las coordenadas en cada $\vec{z} \in (\mathbb{C}^n \setminus \Delta)$. El espacio de órbitas de la acción es $(\mathbb{C}^n \setminus \Delta) / \Sigma_n$ y la proyección al espacio de órbitas es $\tau : (\mathbb{C}^n \setminus \Delta) \rightarrow (\mathbb{C}^n \setminus \Delta) / \Sigma_n$.

Elegimos un punto base $\vec{p} = (p_1, \dots, p_n)$, y tomamos el grupo fundamental $\pi_1((\mathbb{C}^n \setminus \Delta) / \Sigma_n, \tau(\vec{p}))$.

Definición 6. Sea $\mathbb{F}^n = \pi_1(\mathbb{C} \setminus \{1, 2, \dots, n\}, *)$ el grupo libre sobre un conjunto de n generadores x_1, \dots, x_n para un entero fijo $n \geq 2$, y sea $\text{Aut}(\mathbb{F}^n)$ el grupo de automorfismos de \mathbb{F}^n . Sea el subgrupo de $\text{Aut}(\mathbb{F}^n)$ compuesto por los automorfismos σ en \mathbb{F}^n que cumplen las siguientes condiciones:

$$\sigma(x_i) = A_j x_{\tau(i)} A_i^{-1} \quad \text{para } 1 \leq i \leq n$$

y

$$\sigma(x_1 \cdots x_n) = x_1 \cdots x_n$$

donde τ es una permutación de $1, \dots, n$ y los $A_i, A_i \in \mathbb{F}^n$, y sólo dependen de los índices i, j respectivamente.

Teorema 3. *Los tres grupos definidos en las definiciones 4,5,6 para el mismo $n \geq 2$, son isomorfos. Llamamos a este grupo, el grupo de trenzas B_n de Artin.*

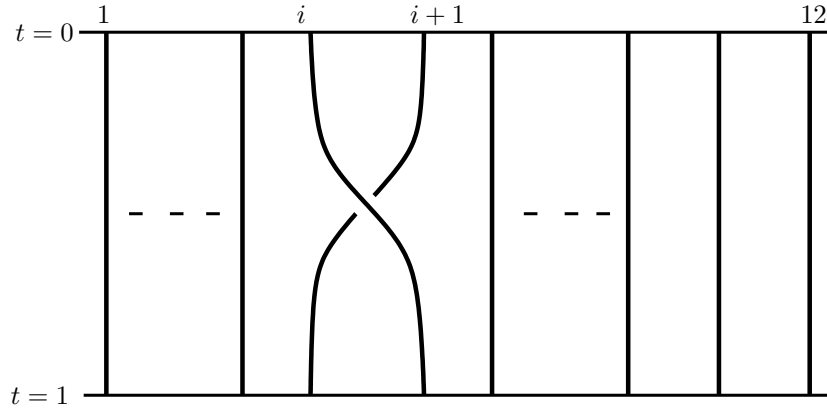


Figura 1.9: Generador σ_i (vista frontal)

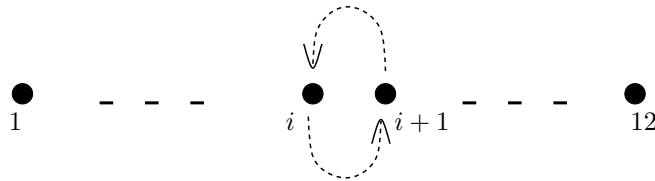


Figura 1.10: Generador σ_i (vista cenital)

Siguiendo esta definición establecida por Emil Artin, el grupo de trenzas con $n=12$ hebras tiene generadores: $\sigma_1, \sigma_2, \dots, \sigma_{11}$, y relaciones:

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \text{ para } i = 1, 2, \dots, 10$$

$$\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j, \quad |i - j| = 1.$$

Por teorema de Dehn-Nielsen-Baer (ver [FM]) para una superficie topológica compacta conexa C_4 de género g , $M(g, 0) \cong \text{Out}^+(\pi_1(C_g, *))$, es decir, que el grupo de difeomorfismos de C_4 módulo isotopía diferenciable es isomorfo al grupo de automorfismos exteriores del grupo fundamental $\pi_1 C_4$.

Esto significa que la monodromía en el grupo de trenzas de los recubrimientos $C_4 \rightarrow \mathbb{P}_{\mathbb{C}}^1$ determina la monodromía geométrica y en el grupo fundamental.

Los puntos de ramificación de los recubrimientos trigonales que estudiamos se mueven en esferas de Riemann. Por ello debemos introducir las trenzas esféricas.

Definición 7. *El espacio de configuración de n puntos en la esfera $S^2 \cong \mathbb{P}_{\mathbb{C}}^1$ para $n \geq 2$ es*

$$\mathbb{P}^n \setminus \Delta = \{(z_1, \dots, z_n) \in \mathbb{P}_{\mathbb{C}}^1 | z_i \neq z_j \text{ para todo } i \neq j\} / \mathfrak{S}_n$$

donde el cociente \mathfrak{S}_n indica que no nos importa en que orden aparecen los puntos en la n -pla. El grupo de trenzas esférico de n hebras ($n \geq 2$) es

$$\tilde{B}_n = \pi_1(\mathbb{P}^n \setminus \Delta, *)$$

El grupo de trenzas esférico admite los mismos generadores $\sigma_1, \dots, \sigma_{n-1}$ que el grupo de trenzas de Artin clásico. El único cambio es que la completación $\mathbb{P}_{\mathbb{C}}^1 = \mathbb{C} \cup \{\infty\}$ introduce una nueva relación en la presentación de \tilde{B}_n .

Teorema 4. *El grupo de trenzas \tilde{B}_n de la 2-esfera S^2 admite una presentación con generadores $\sigma_1, \dots, \sigma_{n-1}$ y las relaciones:*

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ si $|i - j| \geq 2$, $i \leq n-1$, $j \leq n-1$
- $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ $1 \leq i \leq n-2$
- $\sigma_1 \cdots \sigma_{n-2} \sigma_{n-1}^2 \sigma_{n-2} \cdots \sigma_1 = 1$ (la nueva relación introducida al cambiar \mathbb{C} por $\mathbb{P}_{\mathbb{C}}^1$).

1.6. Singular

Singular (ver [Sin]) es un sistema de álgebra computacional para el cálculo de polinomios, con especial énfasis en álgebra conmutativa, geometría algebraica, y la teoría de singularidades. Es gratuito y de código abierto bajo la licencia GNU General Public Licence. Su objeto principal de cómputo son: ideales, módulos y matrices de más de un gran número de anillos de base.

Estos incluyen:

- Anillos de polinomios sobre los cuerpos y algunos anillos (incluidos los números enteros),
- Localizaciones de los anteriores.
- Una clase general de álgebra no conmutativa (incluido el álgebra exterior y el álgebra de Weyl),
- Anillos cociente de los anteriores,
- Productos tensoriales de los anteriores.

Singular se maneja con algoritmos fundamentales:

- Bases de Gröbner. bases estándar y resoluciones libres,
- Factorización de polinomios,
- Resultantes, conjuntos característicos, y encontrar raíces numéricas.

El programa Singular, con sus algoritmos básicos, proporciona soporte a librerías para la resolución de problemas más especializados de la Geometría Algebraica.

Singular está desarrollado bajo la dirección de Wolfram Decker, Gert-Martin Greuel, Pfister Gerhard, y Hans Schönemann, quienes conforman la cabeza principal de Singular, equipo de desarrollo en el Departamento de Matemáticas de la Universidad de Kaiserslautern.

El lector puede hallar una introducción al uso de Singular para cálculos típicos de Álgebra conmutativa y Geometría Algebraica en [DL].

Capítulo 2

Monodromía de la Familia Universal de Curvas Trigonaes

2.1. La familia universal de Clebsch-Hurwitz

Sea $H_{3,4}$ el espacio de Hurwitz de los recubrimientos $\phi : C \rightarrow \mathbb{P}_{\mathbb{C}}^1$ de grado 3, con $g(C) = 4$ y la ramificación simple (presentado en la sección 1.2).

El envío de cada ϕ a sus 12 valores de ramificación define una aplicación holomorfa

$$r : H_{3,4} \rightarrow \mathbb{P}^{12} \setminus \Delta = \left\{ \begin{array}{l} \text{12-plas de puntos diferentes} \\ \text{ordenados en } \mathbb{P}_{\mathbb{C}}^1 \end{array} \right\}$$

Esta aplicación de ramificación tiene grado el número de Hurwitz $h_{4;2,1}$.

Si fijamos una fibra base genérica $\{1, 2, 3, \dots, 12\} \in \mathbb{C}^{12} \setminus \Delta \subset \mathbb{P}^{12} \setminus \Delta$, la fibra $r^{-1}(\{1, 2, 3, \dots, 12\})$ es un conjunto discreto, que corresponde a todas las posibles monodromías del recubrimiento $C \rightarrow \mathbb{P}_{\mathbb{C}}^1$ en el grupo simétrico \mathfrak{S}_3

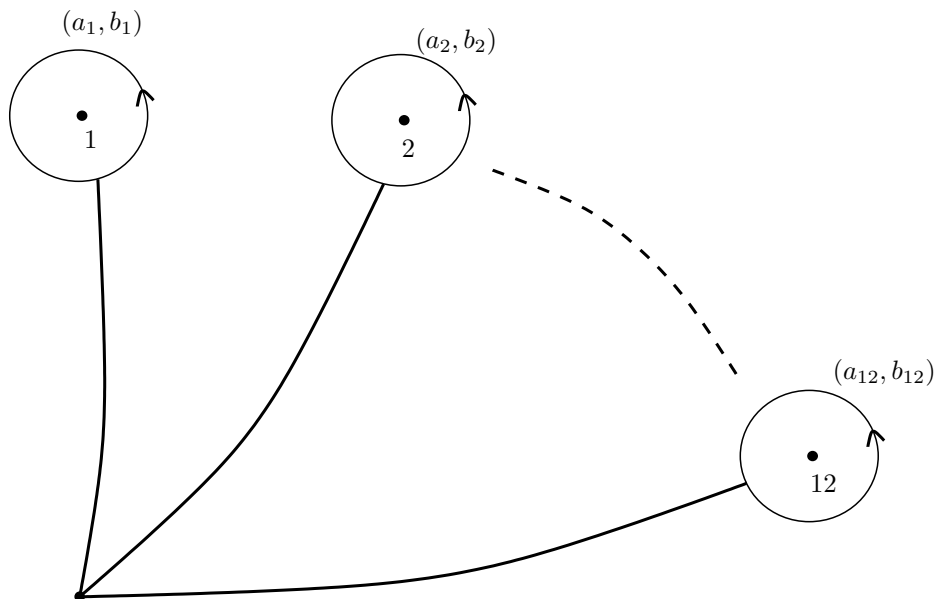


Figura 2.1:

tales que $(a_1, b_1), \dots, (a_{12}, b_{12}) = Id \in \mathfrak{S}_3$ módulo conjugación global en \mathfrak{S}_3 (véase sección 1.2 y [ACG2] pag. 859).

El grupo fundamental de la base del morfismo de ramificación r es

$$\pi_1(\mathbb{P}^{12} \setminus \Delta, \{1, 2, 3, \dots, 12\}) \cong \tilde{B}_{12}$$

(las **trenzas esféricas** de 12 hebras, véase la sección 1.5).

Escogamos el elemento $\{C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1\} \in r^{-1}\{1, 2, 3, \dots, 12\}$ dado por tomar como sistema de caminos el de la figura 2.1, y por monodromía la forma normal de Clebsch-Hurwitz con

$$p(\gamma_1) = p(\gamma_2) = (1, 2), p(\gamma_3) = \dots = p(\gamma_{12}) = (2, 3)$$

En las familias de recubrimientos trigonales

$$\begin{array}{ccc} & \tilde{E} & \\ & \swarrow & \downarrow \\ Y & \longrightarrow & \tilde{B} \end{array} \quad (2.1)$$

la sección 1.3 podemos hallar siempre una conjugación topológica que lleve al recubrimiento $E_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ a esta forma normal de Clebsch-Hurwitz. Esta conjugación puede hallarse de manera efectiva, calculando la monodromía en el grupo simétrico de una araña de Voronoi con las técnicas analíticas descritas en los capítulos 3 y 4, operando por sucesivos movimientos de Hurwitz (véase sección 1.2) para llevar el sistema de caminos a la forma normal.

Como la familia (2.1) es globalmente una familia de recubrimientos trigonales, todo camino cerrado en $\pi_1(\tilde{B}, s_0)$ nos devuelve a la curva $\tilde{E}_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ de partida.

Esto significa que, en la acción de $\pi_1(\mathbb{P}^{12} \setminus \Delta, \{1, 2, 3, \dots, 12\}) \cong \tilde{B}_{12}$ en $r^{-1}(\{1, 2, 3, \dots, 12\})$ nos va a interesar el estabilizador $Stab\{C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1\}$ del recubrimiento normal de Clebsch-Hurwitz $C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1$, osea, los cambios de $\pi_1(\mathbb{P}^{12} \setminus \Delta, \{1, 2, 3, \dots, 12\}) \cong \tilde{B}_{12}$ tales que el transporte paralelo del recubrimiento trigonal $C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1$ y no a un recubrimiento con otra monodromía en el grupo simétrico.

Proposición 1. Sean $\sigma_1, \dots, \sigma_{11}$ los generadores estándar del grupo de trenzas B_{12} y del grupo de trenzas esféricas \tilde{B}_{12} . Tenemos una inclusión:

$$\langle \sigma_1, \sigma_2^3, \sigma_3, \dots, \sigma_{11} \rangle \subseteq Stab\{C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1\} \subset \tilde{B}_{12}$$

Demostración:

Es un cálculo de movimientos de Hurwitz aplicados a la forma normal de Clebsch-Hurwitz

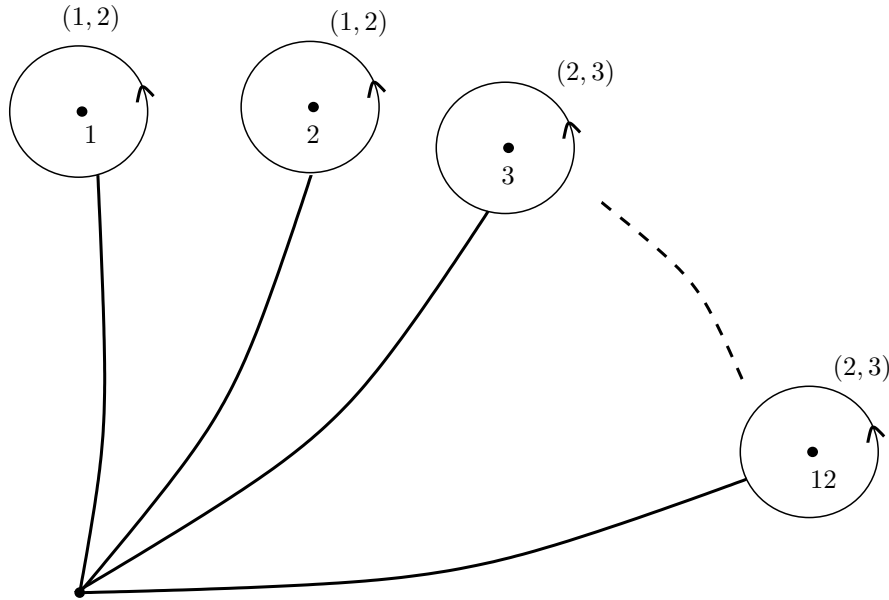


Figura 2.2:

Por el lema 1, sección 1.2, el efecto del movimiento de Hurwitz dado por un generador σ_i aplicado a un sistema de caminos es que la transposición que estaba en posición $(i + 1)$ pasa a estar en posición i , y la que estaba en posición i se conjuga por la que estaba en posición $(i + 1)$ y pasa a estar en posición $(i + 1)$.

La aplicación de este lema muestra inmediatamente que $\sigma_1, \sigma_3, \dots, \sigma_{11}$ dejan invariante la monodromía del sistema de caminos de la presentación de Clebsch-Hurwitz.

Cuando aplicamos σ_2 pasamos de tener una lista de monodromías $\{(1, 2), (1, 2), (2, 3), \dots, (2, 3)\}$ a tener $\{(1, 2), (2, 3), (1, 3), (2, 3), \dots, (2, 3)\}$. Pero aplicar 3 veces el movimiento de Hurwitz σ_2 restaura la forma de Clebsch-Hurwitz original. \square

El subgrupo detectado de $Stab\{C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1\}$ tiene interés notable.

Conjetura 1. $Stab C_1 = \langle \sigma_1, \sigma_2^3, \sigma_3, \dots, \sigma_{11} \rangle$ Pero la demostración de esta conjetura se ha revelado un problema difícil de la teoría de grupos.

Sea ahora el espacio recubridor

$$\tilde{M} \xrightarrow{c} \mathbb{P}^{12} \setminus \Delta$$

asociado al subgrupo $Stab \{C_1 \longrightarrow \mathbb{P}^1 \subset \tilde{B}_{12} = \pi_1(\mathbb{P}^{12} \setminus \Delta, \{1, 2, \dots, 12\})\}$.

Si tomamos el pullback del morfismo de ramificación de Hurwitz por este cambio de base c :

$$\begin{array}{ccc} \tilde{H} & \longrightarrow & H_{3,4} \\ \downarrow \square & & \downarrow r \\ \tilde{M} & \xrightarrow{c} & \mathbb{P}^{12} \setminus \Delta \end{array}$$

tenemos ahora que $\pi_1(\tilde{M}, \{1, 2, \dots, 12\}) = Stab C_1$, por lo tanto

Proposición 2. *Existe una sección*

$$s : \tilde{M} \longrightarrow \tilde{H}$$

que induce biholomorfismo entre \tilde{M} y su imagen $S(\tilde{M}) \subset \tilde{H}$.

Tomemos ahora la familia universal de curvas sobre el espacio de Hurwitz

$$\begin{array}{c} C \\ \downarrow \pi \\ H_{3,4} \end{array}$$

donde $\pi^{-1}(C \longrightarrow \mathbb{P}_{\mathbb{C}}^1) = C$ (cada punto $z \in H_{3,4}$ corresponde a un recubrimiento de $\mathbb{P}_{\mathbb{C}}^1$), $\pi^{-1}(z)$ es la curva que recubre a $\mathbb{P}_{\mathbb{C}}^1$.

Podemos hacer las operaciones de pullback de la familia C por el recubrimiento $c : \tilde{M} \longrightarrow \mathbb{P}^{12} \setminus \Delta$, más restricción a la sección $s(\tilde{M})$:

$$\begin{array}{ccccc} \nu \cdot c \cdot C & \hookrightarrow & c \cdot C & \longrightarrow & C \\ \downarrow & & \downarrow \square & & \downarrow \\ s(\tilde{M}) & \xrightarrow{\nu} & \tilde{H} & \longrightarrow & H_{3,4} \\ & \nearrow \cong & \downarrow \square & & \downarrow \\ & & \tilde{M} & \xrightarrow{c} & \mathbb{P}^{12} \setminus \Delta \end{array}$$

Definición 8. *La familia universal de Clebsch-Hurwitz de grado 3 y género 4 es la familia*

$$\tilde{C} = \nu \cdot (c \cdot C) \longrightarrow \tilde{M} \cong s(\tilde{M})$$

Esta familia tiene una propiedad heredada inmediatamente de la familia de Hurwitz:

Proposición 3. *Existe un morfismo $\phi, 3 : 1$*

$$\begin{array}{ccc} \tilde{C} & \xrightarrow{\tilde{\phi}} & \tilde{M} \times \mathbb{P}_{\mathbb{C}}^1 \\ & \searrow & \downarrow \\ & & \tilde{M} \end{array}$$

que hace conmutativo el diagrama anterior, y tal que para todo $z \in \tilde{M}$ $\tilde{C}_2 \longrightarrow \{z\} \times \mathbb{P}_{\mathbb{C}}^1 \cong \mathbb{P}_{\mathbb{C}}^1$ es el recubrimiento de Hurwitz asociado a \tilde{C}_2 .

Demostración:

Esta propiedad es consecuencia inmediata de la propiedad análoga para la familia de Hurwitz

$$\begin{array}{ccc} C & \longrightarrow & H_{3,4} \times \mathbb{P}_{\mathbb{C}}^1 \\ \downarrow & & \swarrow \\ H_{3,4} & & \end{array}$$

([ACG2]), que se conserva por pullbacks. \square

La segunda propiedad que esta familia heredada de la familia de Hurwitz $C \longrightarrow H_{3,4}$ es una propiedad de universalidad módulo conjugación topológica más sofisticada.

Teorema 5. *Sea*

$$\begin{array}{ccc} \tilde{E} & \longrightarrow & Y \cong \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1 \\ \downarrow & & \swarrow \\ \tilde{B} & & \end{array}$$

una familia holomorfa de recubrimientos trigonales, tal que para todo $s \in \tilde{B}$, \tilde{E}_s tiene género 4, y $\tilde{E}_s \longrightarrow \mathbb{C}^1$ es recubrimiento trigonal con ramificación simple en 12 puntos. Entonces:

- Si el recubrimiento $\tilde{E}_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ tiene por valores de ramificación $\{1, 2, \dots, 12\} \subset \mathbb{P}_{\mathbb{C}}^1$, existe una aplicación clasificante $Y : \tilde{B} \rightarrow \tilde{M}$ tal que la familia $\tilde{E} \rightarrow \tilde{B}$ es pullback de la familia universal de Clebsch-Hurwitz por la aplicación φ .
- Si fijamos un punto base $s_0 \in \tilde{B}$ cualquiera, y una forma normal de Clebsch-Hurwitz para el recubrimiento $\tilde{B}_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ ramificado en la 12-pla $R(s_0)$ existe un difeomorfismo $h : \mathbb{P}_{\mathbb{C}}^1 \rightarrow \mathbb{P}_{\mathbb{C}}^1$ que lleva la 12-pla $R(s_0)$ a $\{1, 2, \dots, 12\}$ preservando la forma normal de Clebsch-Hurwitz, y de manera que el pullback

$$\begin{array}{ccc}
E^\sharp = (Id \times h)^* \tilde{E} & \longrightarrow & \tilde{B} \\
\downarrow \square & & \downarrow \\
\tilde{B} \times \mathbb{P}_{\mathbb{C}}^1 & \xrightarrow{Id \times h} & \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1 \\
\downarrow & \swarrow & \\
\tilde{B} & &
\end{array}$$

define una aplicación clasificante holomorfa $\varphi : \tilde{B} \rightarrow \tilde{M}$ tal que $E^\sharp \cong \varphi^* \tilde{C}$ (E^* es pullback por biholomorfismo de la familia universal de Clebsch-Hurwitz).

Demostración:

- (i) Si el recubrimiento trigonal $\tilde{E}_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ tiene ramificación en $\{1, 2, \dots, 12\}$, existe un sistema de generadores para $\pi_1(\mathbb{P}_{\mathbb{C}}^1 \setminus \{1, 2, \dots, 12\}, *)$ tal que el recubrimiento trigonal tiene la monodromía de Clebsch-Hurwitz (véase sección 1.2).

Como la aplicación de ramificación

$$\tilde{B} \xrightarrow{r} \mathbb{P}^{12} \setminus \Delta$$

es holomorfa, esta aplicación levantará como aplicación holomorfa al recubrimiento $\tilde{M} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ si y sólo si, la imagen $r_*(\pi_1(\tilde{B}, s_0))$ está contenida en $\pi_1(\tilde{M}) = Stab C_1 \subset \pi_1 \mathbb{P}^{12} \setminus \Delta$.

Como la familia

$$\begin{array}{ccc}
\tilde{E} & \longrightarrow & \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1 \\
\downarrow & \swarrow & \\
\tilde{B} & &
\end{array}$$

es de recubrimientos trigonales, si tomamos un lazo $\gamma : (S^1, 1) \longrightarrow (\tilde{B}, s_0)$ la familia de curvas $\tilde{E}_{\gamma(t)} \longrightarrow \gamma(e^{it}) \in \gamma(S^1)$ es localmente trivial con $\tilde{E}_{\gamma(0)} \cong \tilde{E}_{\gamma(1)}$. De hecho, por el teorema de Ehresman (véase sección 1.2) en versión relativa, se tiene una familia topológicamente localmente trivial de recubrimientos trigonales

$$\begin{array}{ccc} \tilde{E}_{\gamma(t)} & \longrightarrow & \gamma(e^{it}) \\ \downarrow & \nearrow & \\ \mathbb{P}_{\mathbb{C}}^1 & & \end{array}$$

con el divisor de ramificación $r(\gamma(t))$ variando de manera localmente trivial.

Al hacer transporte paralelo del divisor $r(\gamma(t))$ para $e^{it} \in S^1$ acabamos regresando al divisor de partida: $r(\gamma(1)) = r(\gamma(0))$. (ya que $\gamma(1) = \gamma(0)$). Ahora podemos aplicar el sistema de generadores de $\pi_1(\mathbb{P}_{\mathbb{C}}^1) \setminus \{1, 2, \dots, 12\}, *$ que teníamos de partida a la curva $\tilde{E}_{\gamma(1)} \cong \tilde{E}_{\gamma(0)}$. Como es la misma curva, y usamos el mismo sistema de generadores, obtenemos de vuelta que $\tilde{E}_{s_0} \longrightarrow \mathbb{P}_{\mathbb{C}}^1$ es el recubrimiento normal de Clebsch-Hurwitz de partida.

Por lo tanto, para toda clase $\gamma \in \pi_1(\tilde{B}, s_0)$ la monodromía sobre γ estabiliza la curva $C \longrightarrow \mathbb{P}_{\mathbb{C}}^1$, y la aplicación clasificante $\tilde{B} \longrightarrow \mathbb{P}^{12} \setminus \Delta$ levanta a \tilde{M} .

Una vez tenemos la aplicación clasificante $\tilde{B} \longrightarrow \tilde{M} \cong s(\tilde{M}) \subset H_{3,4}$ obtenemos que la familia sobre \tilde{B} es pullback de la familia universal sobre $H_{3,4}$.

Este pullback factoriza como

$$\begin{array}{ccccc} \tilde{E} & \longrightarrow & \tilde{C} & \longrightarrow & C \\ \downarrow & & \downarrow \square & & \downarrow \\ \tilde{B} & \xrightarrow{\varphi} & \tilde{M} & \longrightarrow & H_{3,4} \end{array}$$

- (ii) En el caso en el que la 12-pla de valores de ramificación de $\tilde{E}_{s_0} \rightarrow \mathbb{P}_{\mathbb{C}}^1$ no sea la $\{1, 2, \dots, 12\}$ sino otra $r(s_0)$. Tenemos el sistema de caminos generadores de $\pi_1(\mathbb{P}^1 \setminus r(s_0), *)$ la monodromía de Clebsch-Hurwitz. Existe un difeomorfismo que lleva $r(s_0)$ a $\{1, 2, \dots, 12\}$ y este sistema de caminos al estándar adoptado para definir $C_1 \rightarrow \mathbb{P}_{\mathbb{C}}^1$.

Este difeomorfismo no tiene por qué conservar la monodromía en el grupo \mathfrak{S}_3 , pero define un difeomorfismo h entre los recubrimientos con sus respectivas ramificaciones.

$$\begin{array}{ccc} C_1 & \xrightarrow{h} & \tilde{E}_{s_0} \\ \downarrow & \swarrow & \\ \mathbb{P}_{\mathbb{C}}^1 & & \end{array}$$

El pullback por $Id \times h$ de $\tilde{E} \rightarrow \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1$ extiende este difeomorfismo a uno de las familias $E^{\sharp} \cong_{e\infty} \tilde{E}$, y la familia $E^{\sharp} \rightarrow \tilde{B}$ cumple las hipótesis del caso (i) en s_0 . \square

2.2. Monodromía de la familia universal de Clebsch-Hurwitz de curvas trigonales de género 4

Partimos de la familia universal de Clebsch-Hurwitz $\tilde{C} \rightarrow \tilde{M}$ hallada en la sección anterior. Como toda familia de recubrimientos trigonales de género 4

$$\begin{array}{ccc} \tilde{E} & \longrightarrow & \tilde{B} \times \mathbb{P}_{\mathbb{C}}^1 \\ \downarrow & \swarrow & \\ \tilde{B} & & \end{array}$$

es pullback de esta familia universal módulo una posible conjugación topológica del cálculo de la monodromía en el grupo fundamental de esta familia, para una selección particular de punto base obtenemos:

- la monodromía en el mapping class group $M(4, 0) = Out(\pi_1(C_4, *))$ para cualquier familia de recubrimientos trigonales de género 4, ya que el valor de cada monodromía en $Out(\pi_1(C_4, *)) = Aut \pi_1(C_4, *) / \pi_1(C_4, *)$ es independiente del punto base escogido.
- la monodromía en el grupo fundamental de una familia trigonal de curvas de género 4, tal que el punto base escogido sea pullback del seleccionado en la demostración del teorema, va el morfismo clasificante.

- la monodromía en el grupo fundamental de cualquier familia trigonal de curvas de género 4, si conseguimos determinar además el camino trazado por la sección punto base escogida una vez transportada a la familia universal por el difeomorfismo $Id \times h$ del teorema 5 anterior.

Asumiremos en esta sección la conjetura de 2.1 de que $Stab C_1 = \langle \sigma_1, \sigma_2^3, \sigma_3, \dots, \sigma_{11} \rangle \subseteq \tilde{B}_{12}$. Si esta conjetura es cierta, tendremos la monodromía completa de la familia universal de Clebsch-Hurwitz. Si la conjetura es falsa, como se ha demostrado en la proposición que $\langle \sigma_1, \sigma_2^3, \sigma_3, \dots, \sigma_{11} \rangle \subseteq Stab C_1$ las monodromías aquí calculadas son correctas, y para tener una descripción completa de la monodromía geométrica y en el grupo fundamental de la familia universal de Clebsch-Hurwitz de género 4 bastará con añadir a los cálculos del teorema 4, el cálculo de la monodromía de los nuevos caminos generadores que puedan aparecer de $Stab C_1$.

El resultado es:

Teorema 6. *Sea $\pi_1(C_4, p) = \langle a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4 | [a_1, b_1] \cdots [a_4, b_4] = 1 \rangle$, la presentación del grupo fundamental de la superficie de género 4 indicada en la figura 2.3, y $p : C_4 \rightarrow \mathbb{P}_{\mathbb{C}}^1 \cong S^2$ la proyección trigonal con 12 puntos de ramificación simples $\{1, 2, \dots, 12\}$.*

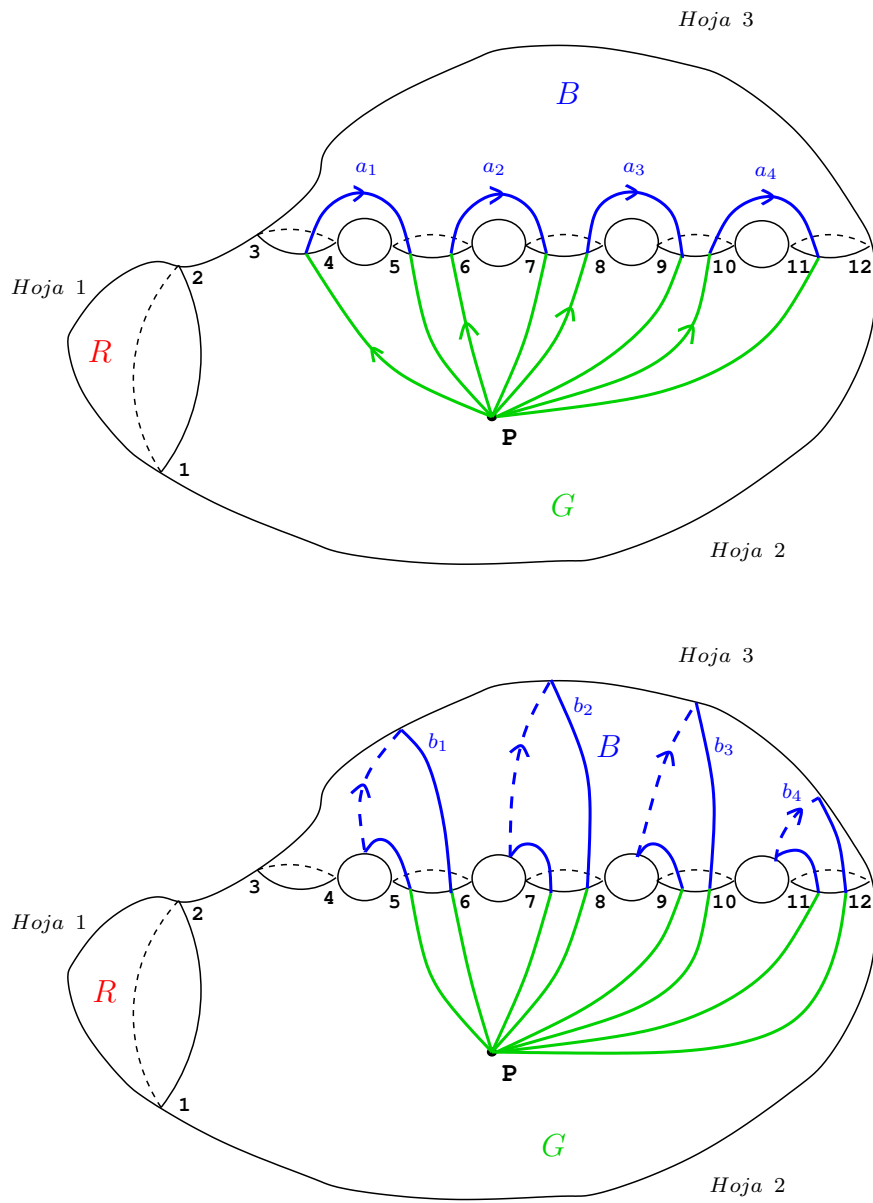


Figura 2.3: R, G, B describen las tres hojas del recubrimiento a $C_4 \rightarrow \mathbb{P}_{\mathbb{C}}^1$ ramificado en $\{1, 2, \dots, 12\}$.

La acción de monodromía del grupo de trenzas B_{12} en $\pi_1(C_4, p)$ definida por este recubrimiento es la indicada en las siguientes tablas:

	σ_1	σ_2^3	σ_3	σ_4	σ_5	σ_6
a_1			$a_1 b_1$		$a_1 b_1 a_2 b_2^{-1} a_2^{-1}$	
a_2					$a_2 b_2 a_2^{-1} b_1^{-1} a_2$	
a_3						
a_4						
b_1				$a_1^{-1} b_1$	$a_2 b_2 a_2^{-1} b_1 a_2 b_2^{-1} a_2^{-1}$	
b_2						$a_2^{-1} b_2$
b_3						
b_4						

	σ_7	σ_8	σ_9	σ_{10}	σ_{11}
a_1					
a_2	$a_2 b_2 a_3 b_3^{-1} a_3^{-1}$				
a_3	$a_3 b_3 a_3^{-1} b_2^{-1} a_3$		$a_3 b_3 a_4 b_4^{-1} a_4^{-1}$		
a_4			$a_4 b_4 a_4^{-1} b_3^{-1} a_4$		$a_4 b_4$
b_1					
b_2	$a_3 b_3 a_3^{-1} b_2 a_3 b_3^{-1} a_3^{-1}$				
b_3		$a_3^{-1} b_3$	$a_4 b_4 a_4^{-1} b_3 a_4 b_4^{-1} a_4^{-1}$		
b_4				$a_4^{-1} b_4$	b_4

Convenio:

Las casillas en blanco significan que el lazo ha quedado invariante por la monodromía de la correspondiente trenza.

Demostración:

Para calcular la acción de monodromía de B_{12} en $\pi_1(C_4, p)$ en la familia universal trigonal:

- Tomamos el recubrimiento trigonal $p : C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$ de la figura 2.1
- Representamos en $\mathbb{P}_{\mathbb{C}}^1 = \mathbb{C} \cup \{\infty\}$ las imágenes de los caminos $a_1, \dots, a_4, a_1, \dots, a_4$ que genera $\pi_1(C_4, P)$. Indicamos los cambios de hoja del camino imagen de la monodromía según su color en C_4 y el sistema de cortes entre puntos de ramificación generador a $\sigma_1, \sigma_1, \dots, \sigma_{11}$.
- Aplicamos la trenza σ a los puntos de ramificación y al sistema de cortes que definen el recubrimiento $p : C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$. Obtenemos el camino imagen \tilde{c} .
- Se describe la trenza σ , y se regresa a $C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$ de partida. Volvemos, si hace falta, al sistema de cortes original entre los puntos de ramificación, y representamos aquí \tilde{c} . Estos caminos resultantes se levantan al recubrimiento $3 : 1$. Esta operación está bien definida porque la monodromía de σ es la identidad entorno al punto base de $\pi_1(C_4, P)$, y la hoja del recubrimiento en P , más los cambios de hoja al cruzar los cortes determinan el camino \tilde{c} en C_4 .
- Para cada camino que halla variado con la trenza, indicados caminos $\tilde{a}_1, \dots, \tilde{b}_4$, imagen por monodromía en el sistema de cortes del recubrimiento original $p : C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$, los representamos en la superficie con la presentación en asas, por un sólo color.
- Pasamos de la representación trigonal $p : C_4 \longrightarrow \mathbb{P}_{\mathbb{C}}^1$ a la presentación estándar de C_4 como superficie de cuatro asas (figura 2.4), cortamos las asas donde pasa el camino imagen \tilde{c} y deformamos hasta un polígono, para luego hallar la palabra en los generadores originales a_1, \dots, b_4 de la imagen del camino por la trenza.

Convenios:

1. Para cada trenza $\sigma_1, \dots, \sigma_{11}$ vamos a calcular sólo la monodromía de los caminos entre a_1, \dots, b_4 que intersequen al soporte del transporte paralelo y la monodromía, dado que los caminos que no cortan a este soporte no son movidos por la monodromía.
2. Anotamos al principio de cada figura del cálculo:
 - El generador σ_i del que calculamos la monodromía.
 - La imagen $\sigma(a_j) = \tilde{a}_j$ o $\sigma(b_j) = \tilde{b}_j$ que estamos calculando.

En la figura 2.3 se ha asignado los colores rojo (R), verde (G) y azul (B) en las 3 hojas del recubrimiento a $\mathbb{P}_\mathbb{C}^1$. Los caminos estudiados se concentran en dos hojas, representados por los colores verde y azul. La hoja en la que el color rojo representaría algún camino, no tiene uso debido a la “forma normal de Hurwitz” escogida para el recubrimiento a $\mathbb{P}_\mathbb{C}^1$.

Para estudiar la homotopía que llevan los caminos $\tilde{a}_1, \dots, \tilde{b}_4$ imagen de cada monodromía hasta una palabra en los generadores $a_1, \dots, a_4, b_1, \dots, b_4$ originales, deformamos la presentación de la figura del enunciado (adaptada al recubrimiento a una presentación a la descomposición de la figura en asas), como muestra la figura 2.4:

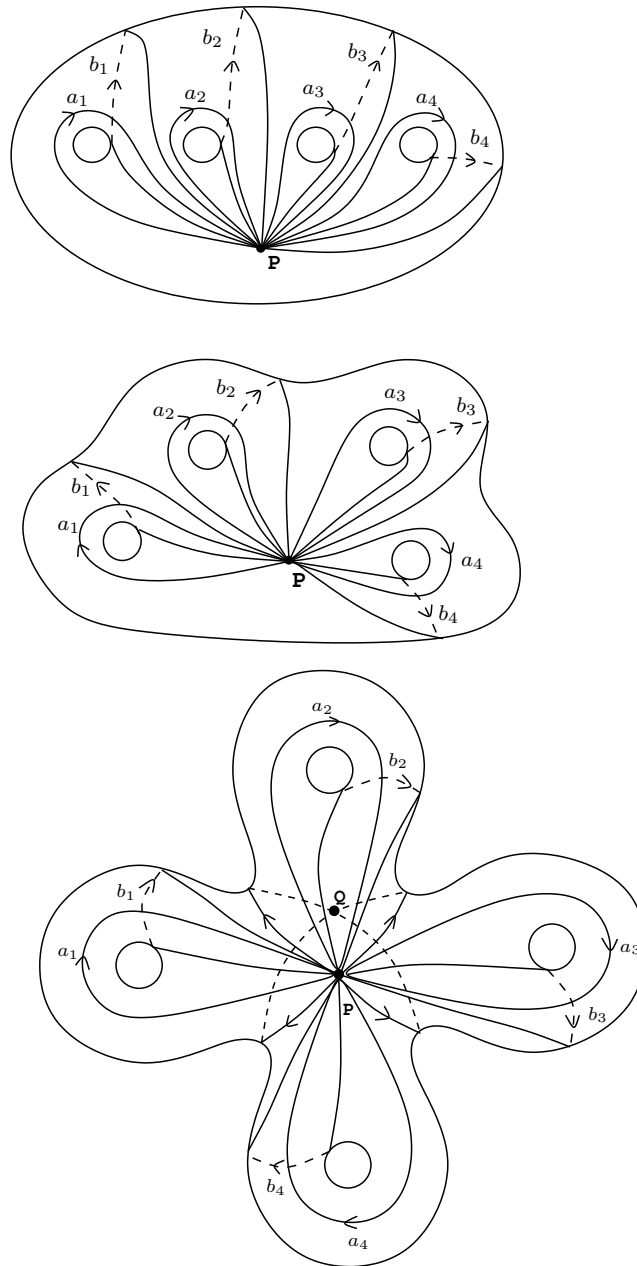


Figura 2.4: Deformación de C_4 hasta su representación topológica estandar como esfera con 4 asas.

Vamos a discutir primero la monodromía de los caminos σ_1 y de σ_2^3 , porque es trivial.

Monodromía de σ_1 : Es identidad fuera de un entorno tubular del camino de 1 a 2. Todos los generadores de la representación de $\pi_1(C_4, P)$ evita este entorno, como muestra la figura 2.6. Por lo tanto, la monodromía de σ_1 es Id en $\pi_1(C_4, P)$.

Monodromía de σ_2^3 : Es identidad fuera de un entorno tubular del camino de 2 a 3. Todos los generadores de la representación de $\pi_1(C_4, P)$ evita este entorno, como muestra la figura 2.6. Por lo tanto, la monodromía de σ_2^3 es Id en $\pi_1(C_4, P)$.

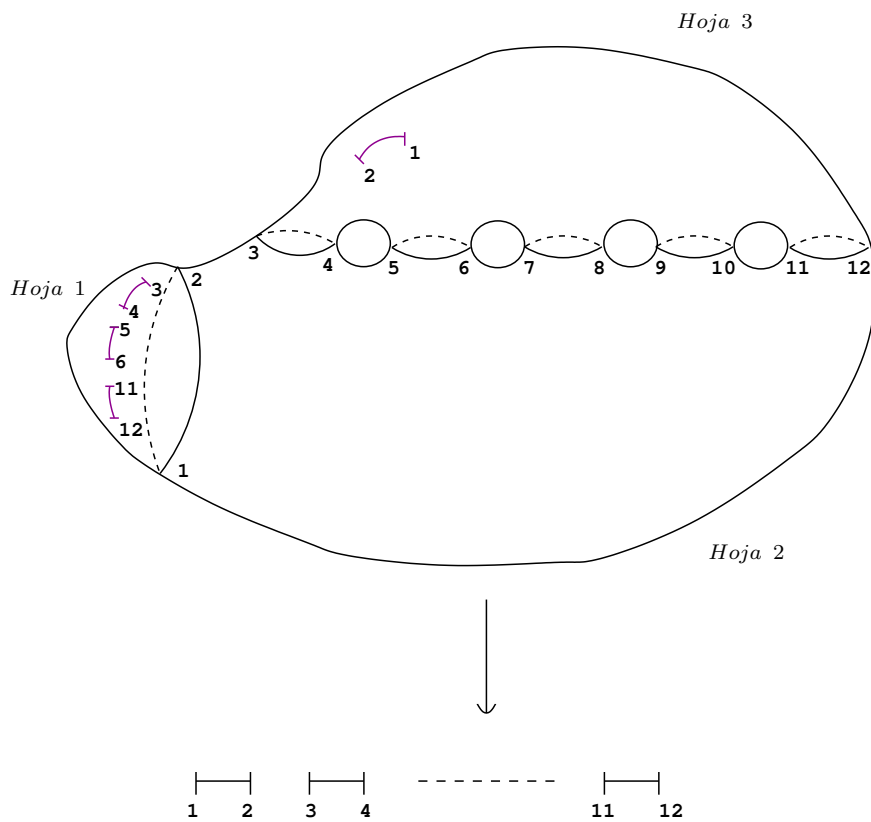


Figura 2.5: Anti-imágenes del sistema de cortes entre puntos de ramificación que definen las 3 hojas del recubrimiento. (hoja R = hoja 1, G = hoja 2, B = hoja 3).

Cada punto de ramificación tiene dos anti-imágenes en C_4 : una doble (hojas R, G para 1 y 2, G y B para los demás puntos de ramificación) y una anti-imagen simple.

Cada corte entre 2 puntos de ramificación tiene tres anti-imágenes: dos que separan las hojas en que se ramifican, una tercera que corresponde a la anti-imagen de no ramificación de los extremos, y que no juega ningún papel (representada en magenta).

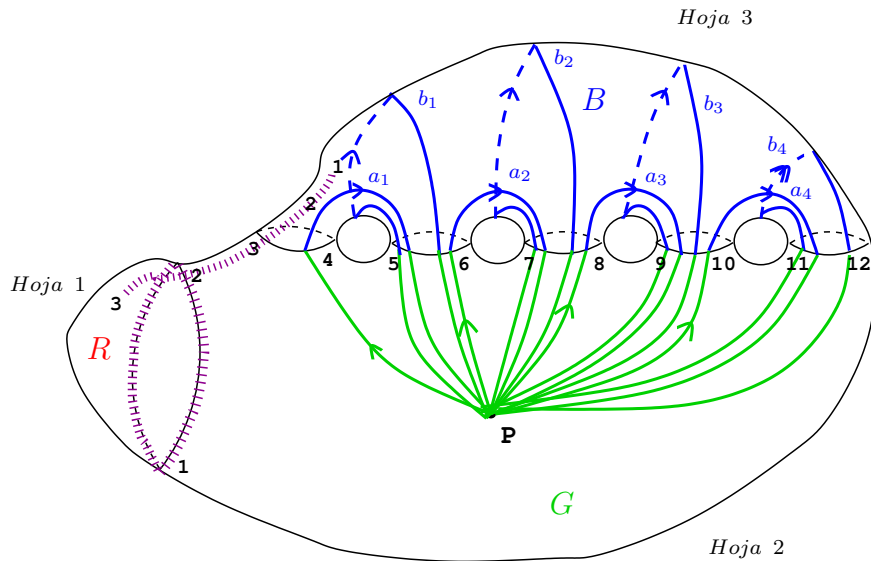


Figura 2.6: La presentación escogida de $\pi_1(C_4, P)$. Nótese como los generadores de los cortes $\overline{12}, \overline{23}$ (sombreados en magenta).

Ahora queda por discutir la monodromía de los generadores $\sigma_3, \dots, \sigma_{11}$ en la presentación escogida de $\pi_1(C_4, *)$. Este es el resto de la prueba, calculando la monodromía generador a generador.

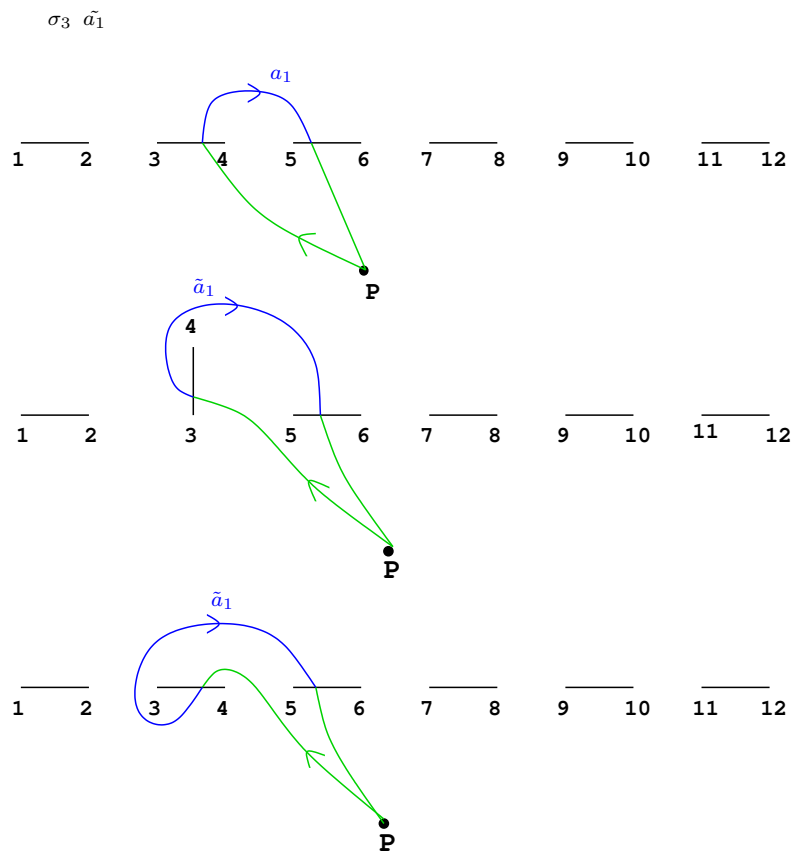


Figura 2.7: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

La trenza σ provoca el giro de una semivuelta en sentido directo de los 2 puntos de ramificación a los que afecta (véase figuras 1.9, 1.10).

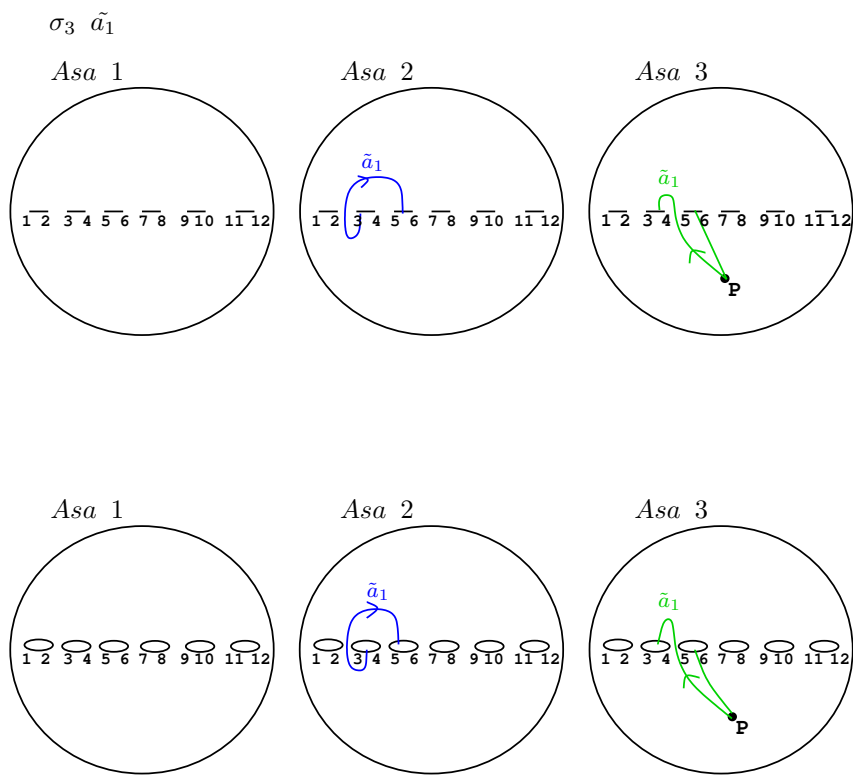


Figura 2.8: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

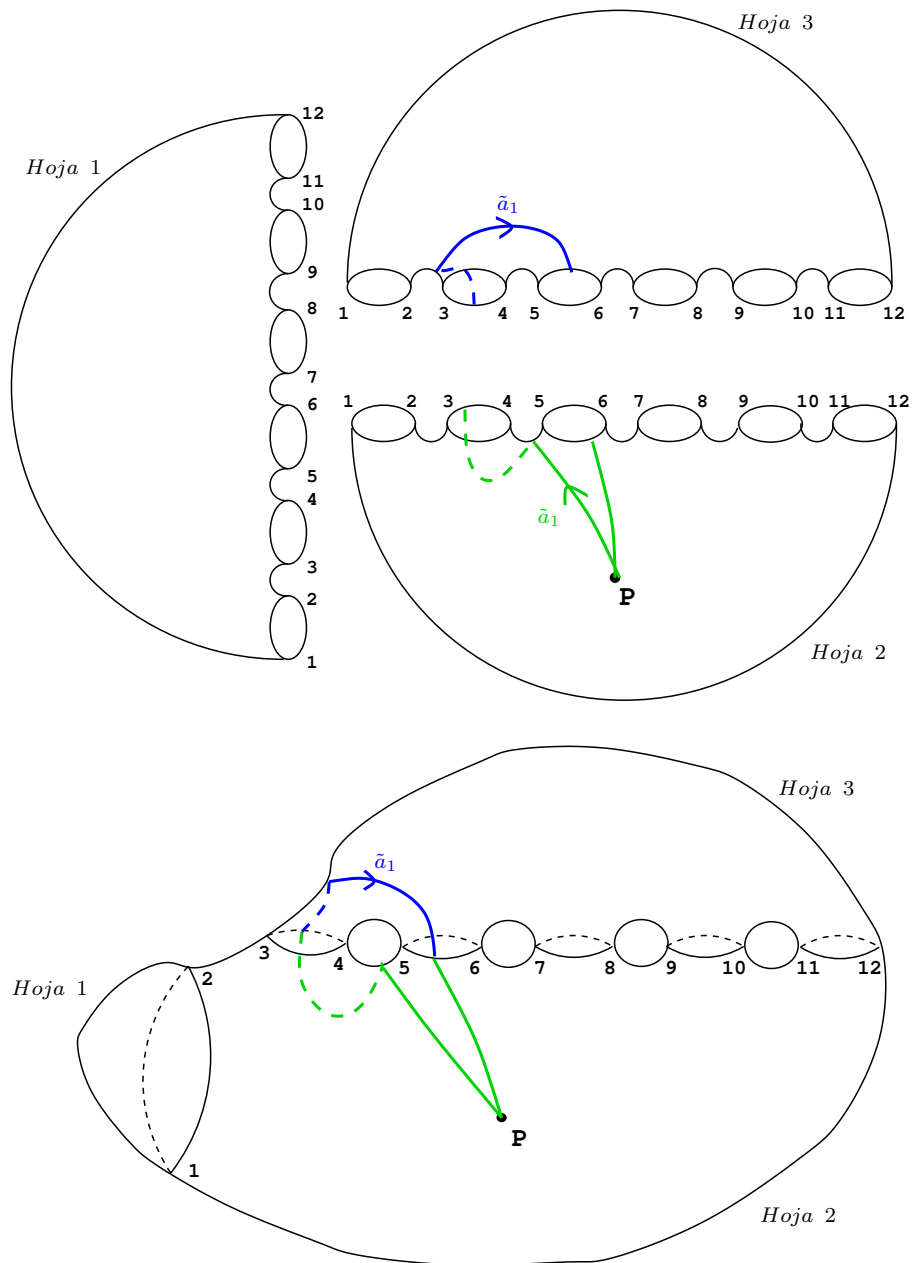


Figura 2.9: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

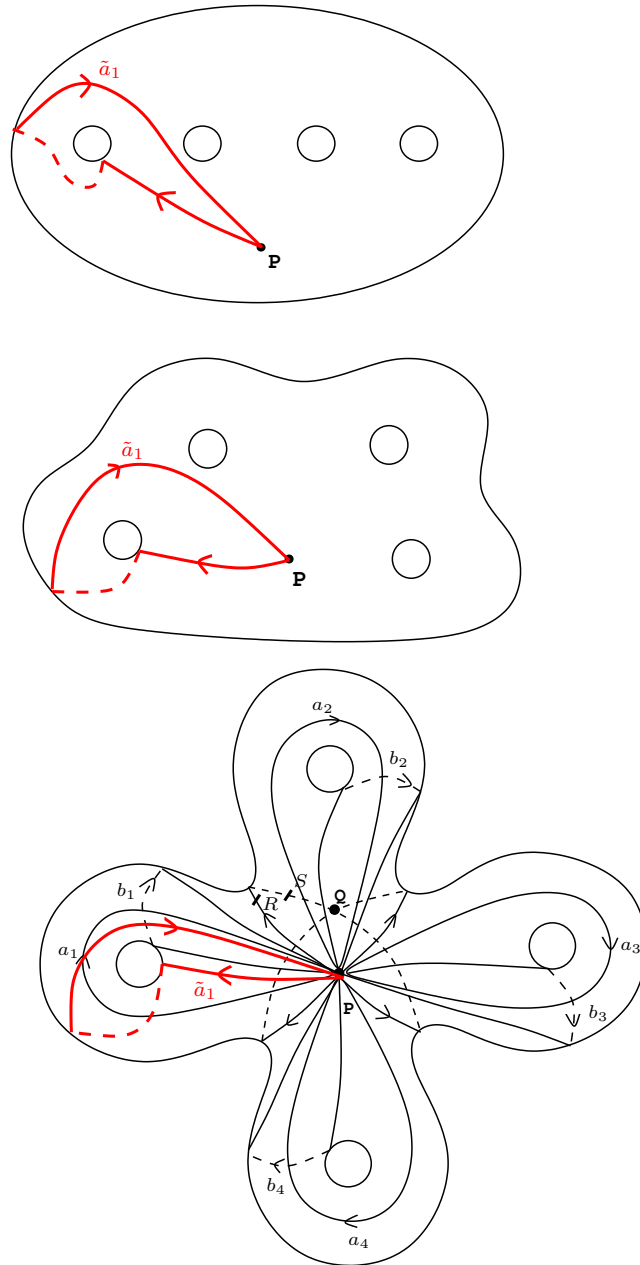


Figura 2.10: Camino imagen \tilde{c} en la esfera con 4 asas.

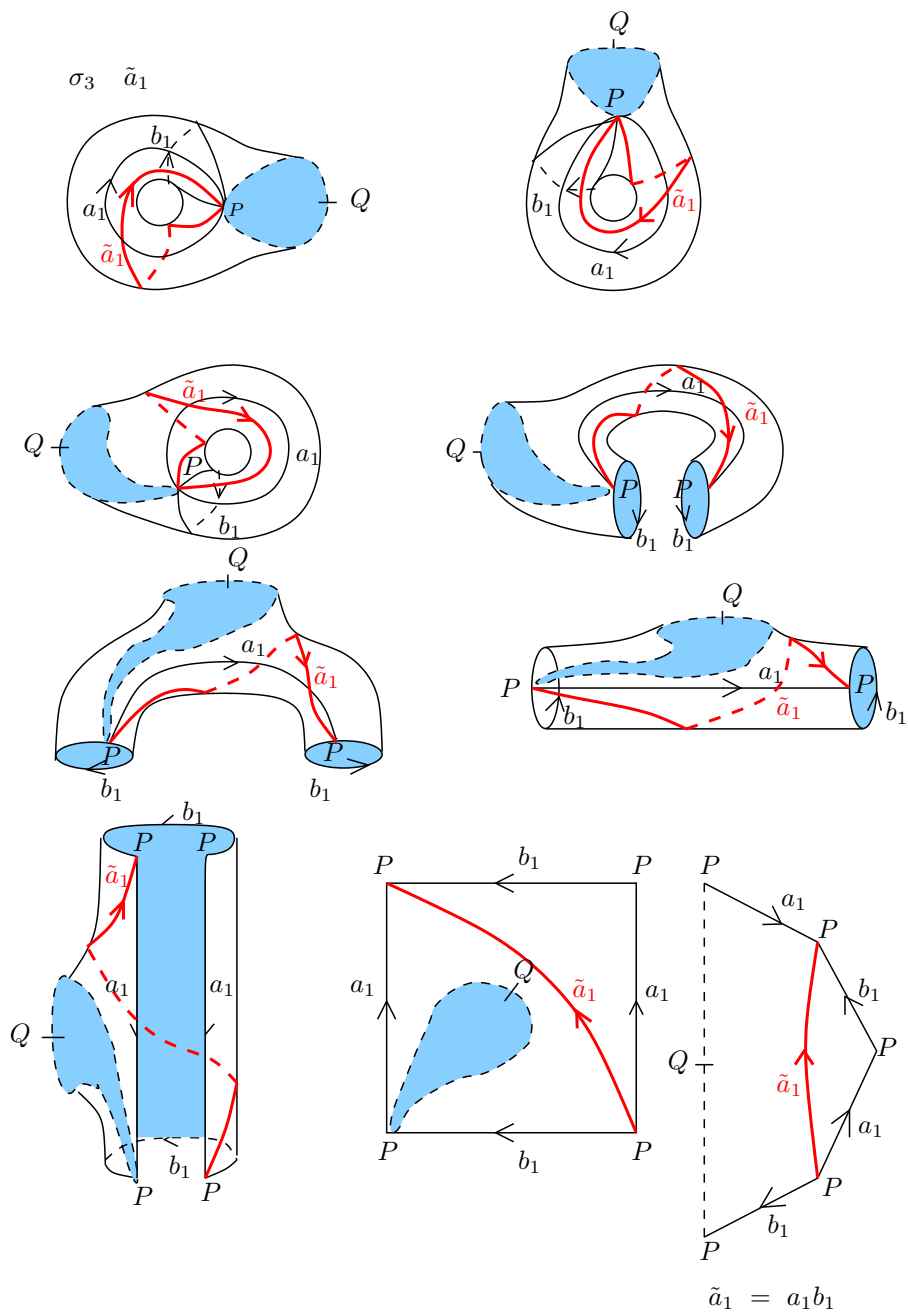


Figura 2.11: Deformación de una asa de la superficie, en la que se muestra paso a paso la homotopía del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

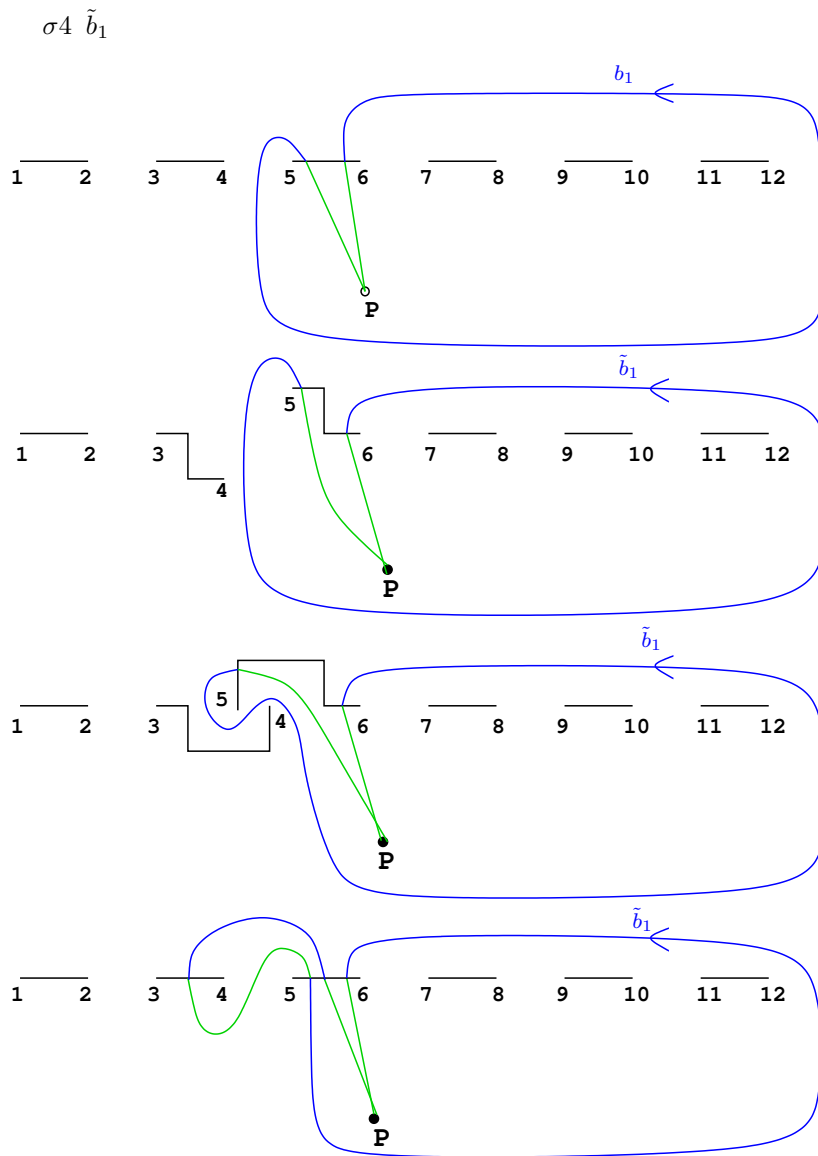


Figura 2.12: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

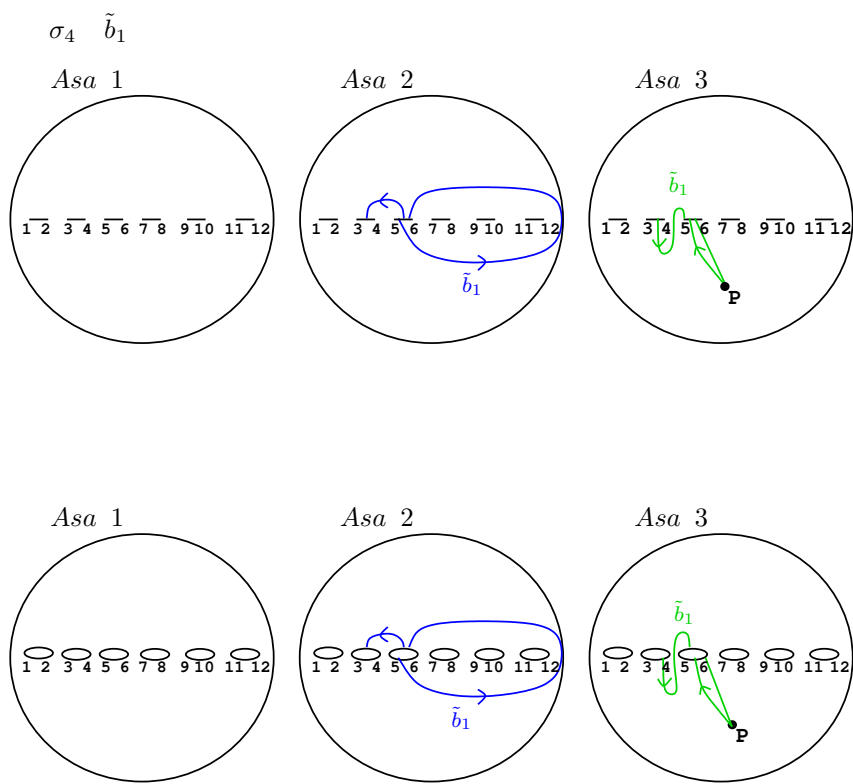


Figura 2.13: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

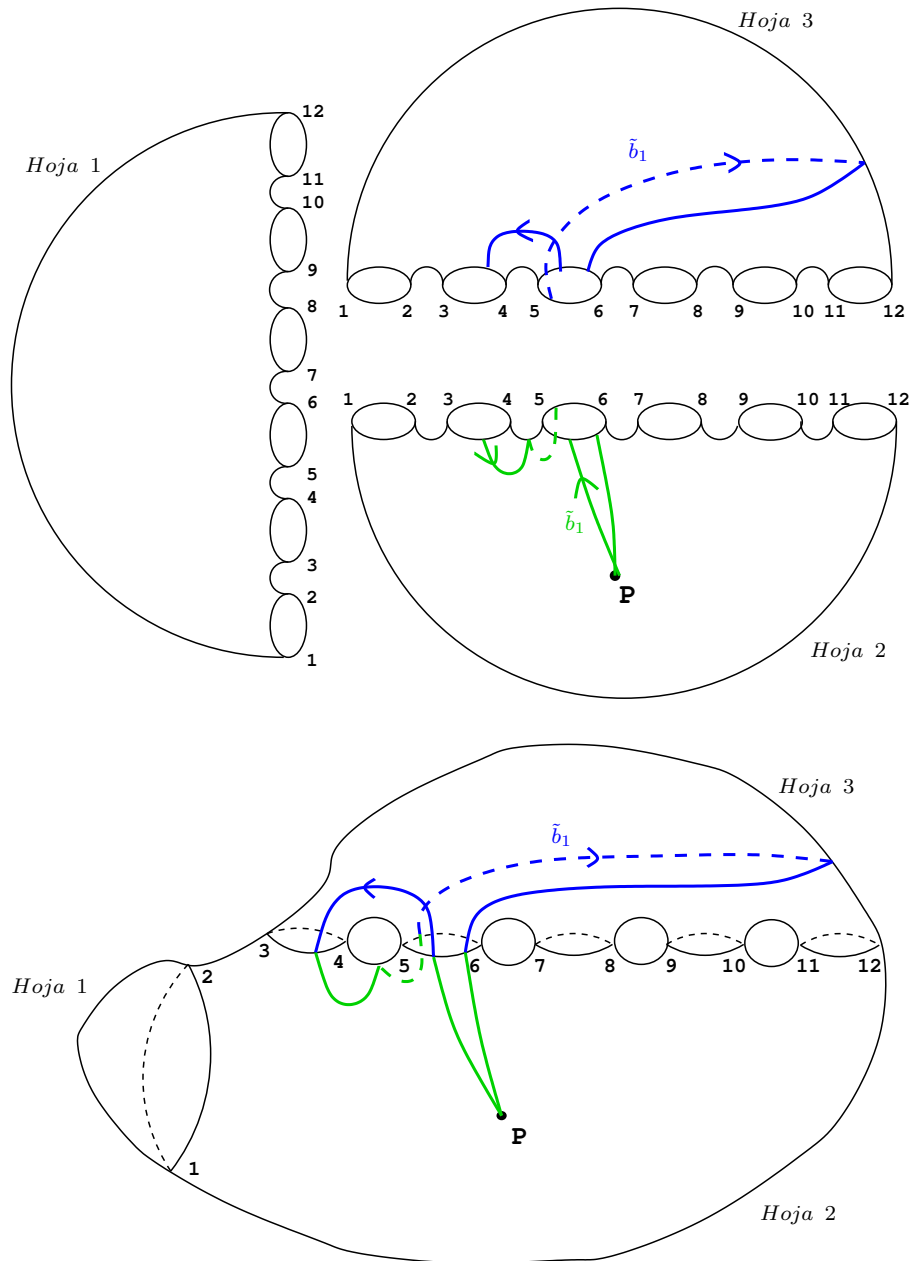


Figura 2.14: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

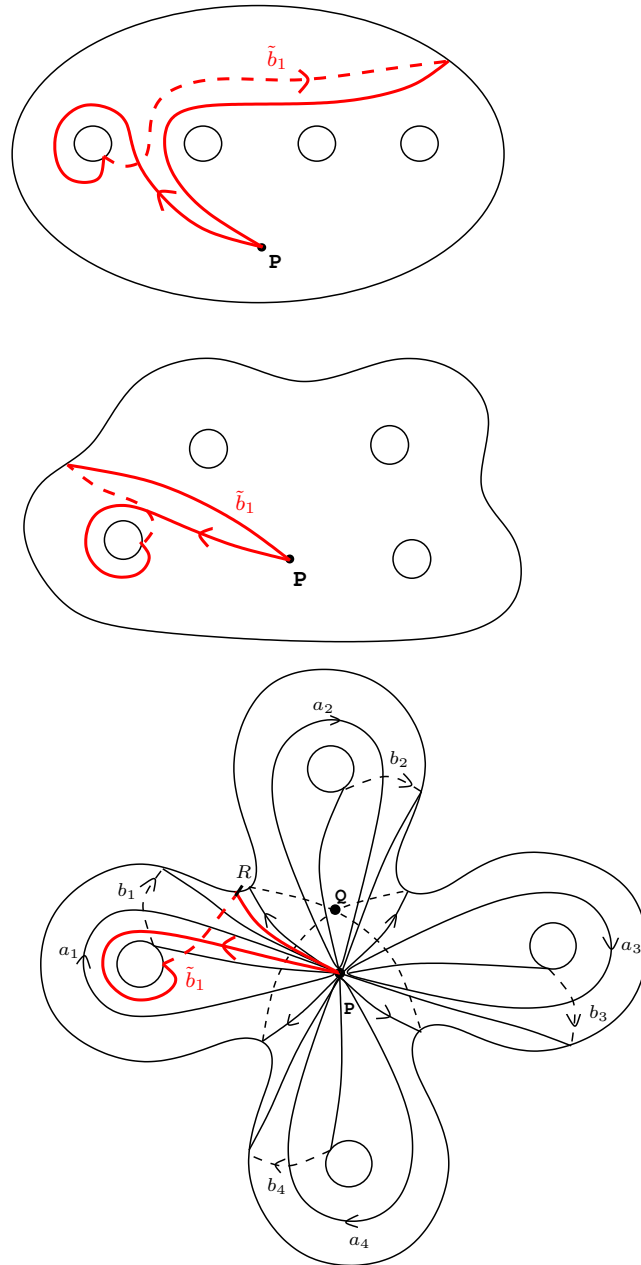


Figura 2.15: Camino imagen \tilde{c} en la esfera con 4 asas.

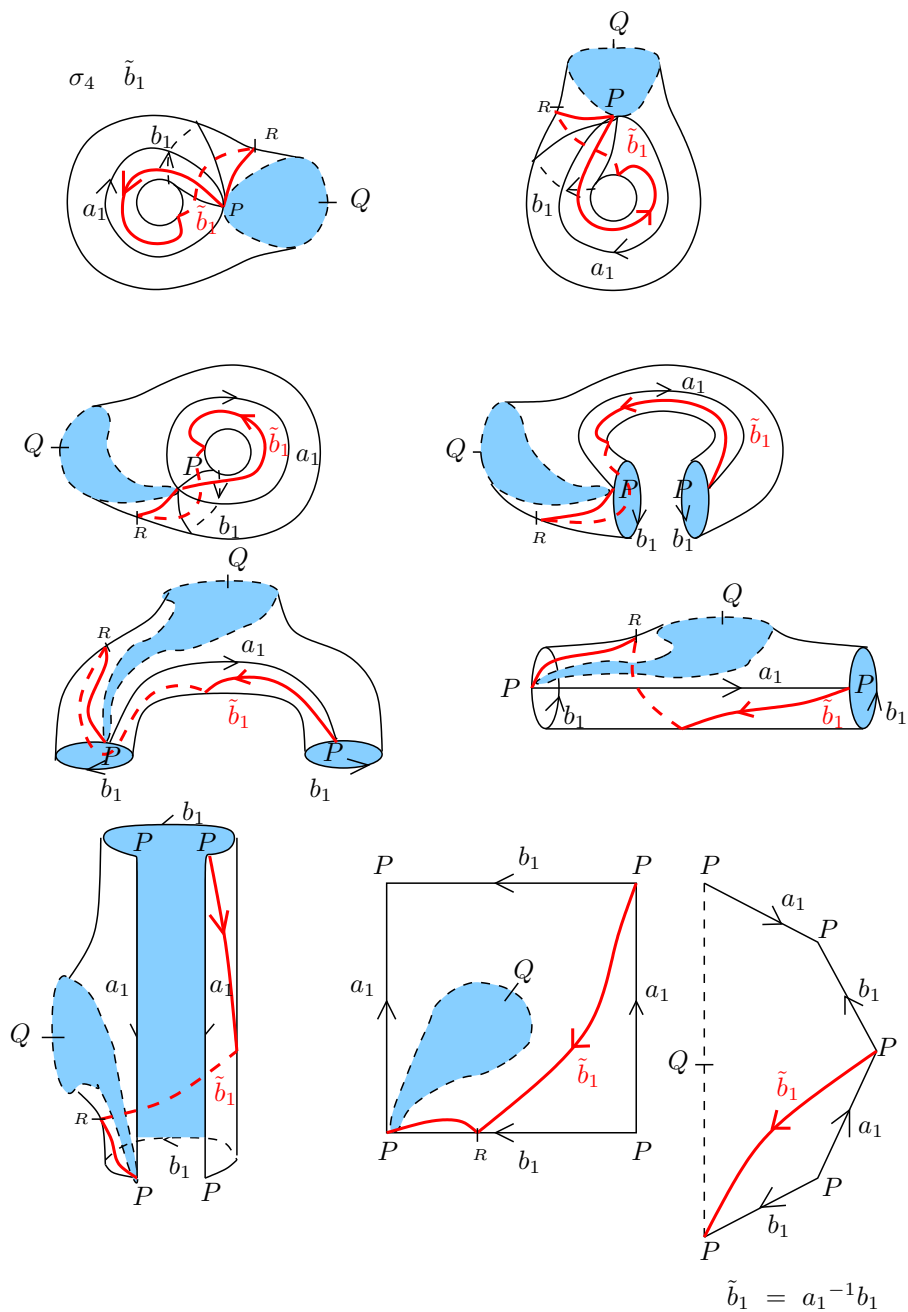


Figura 2.16: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

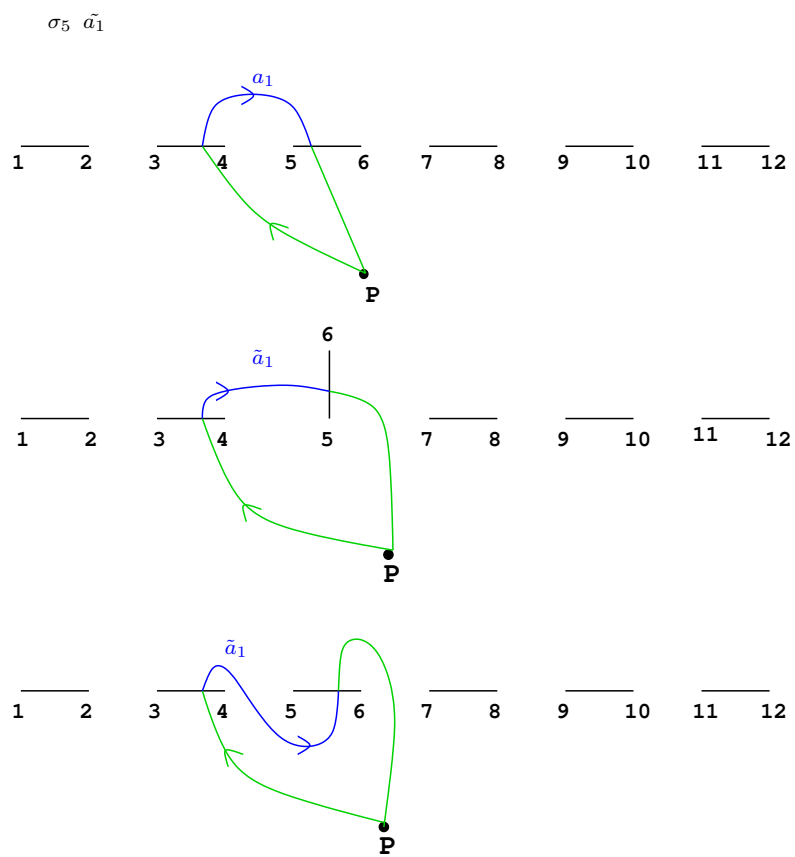


Figura 2.17: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

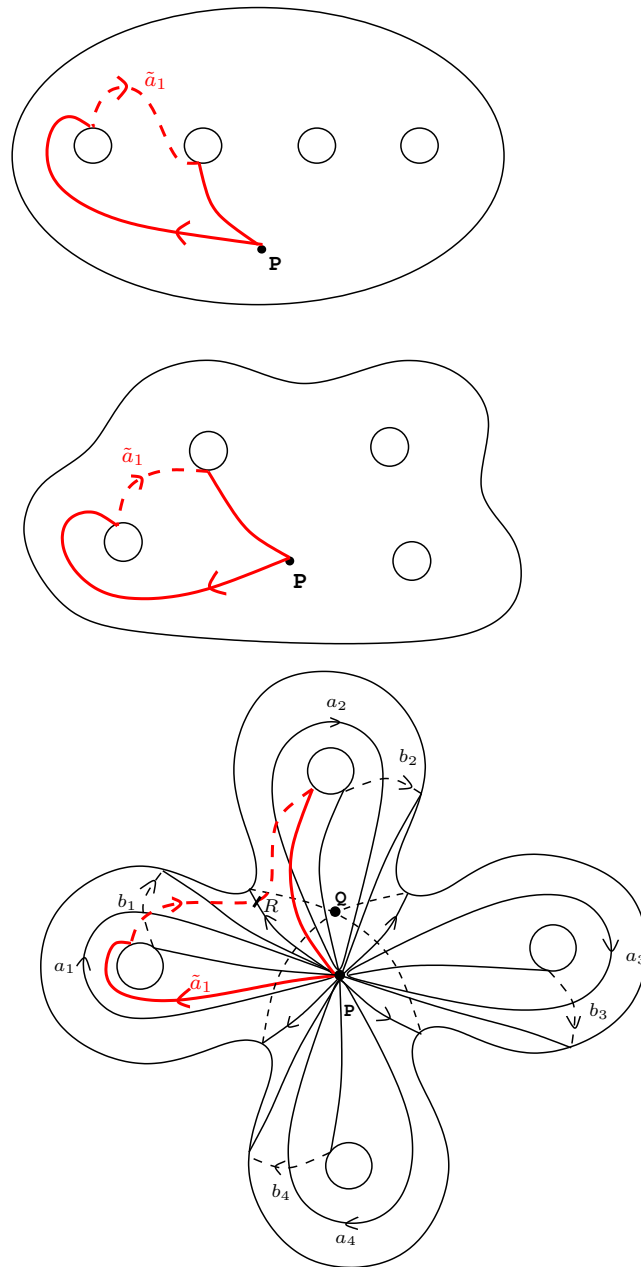


Figura 2.20: Camino imagen \tilde{c} en la esfera con 4 asas.

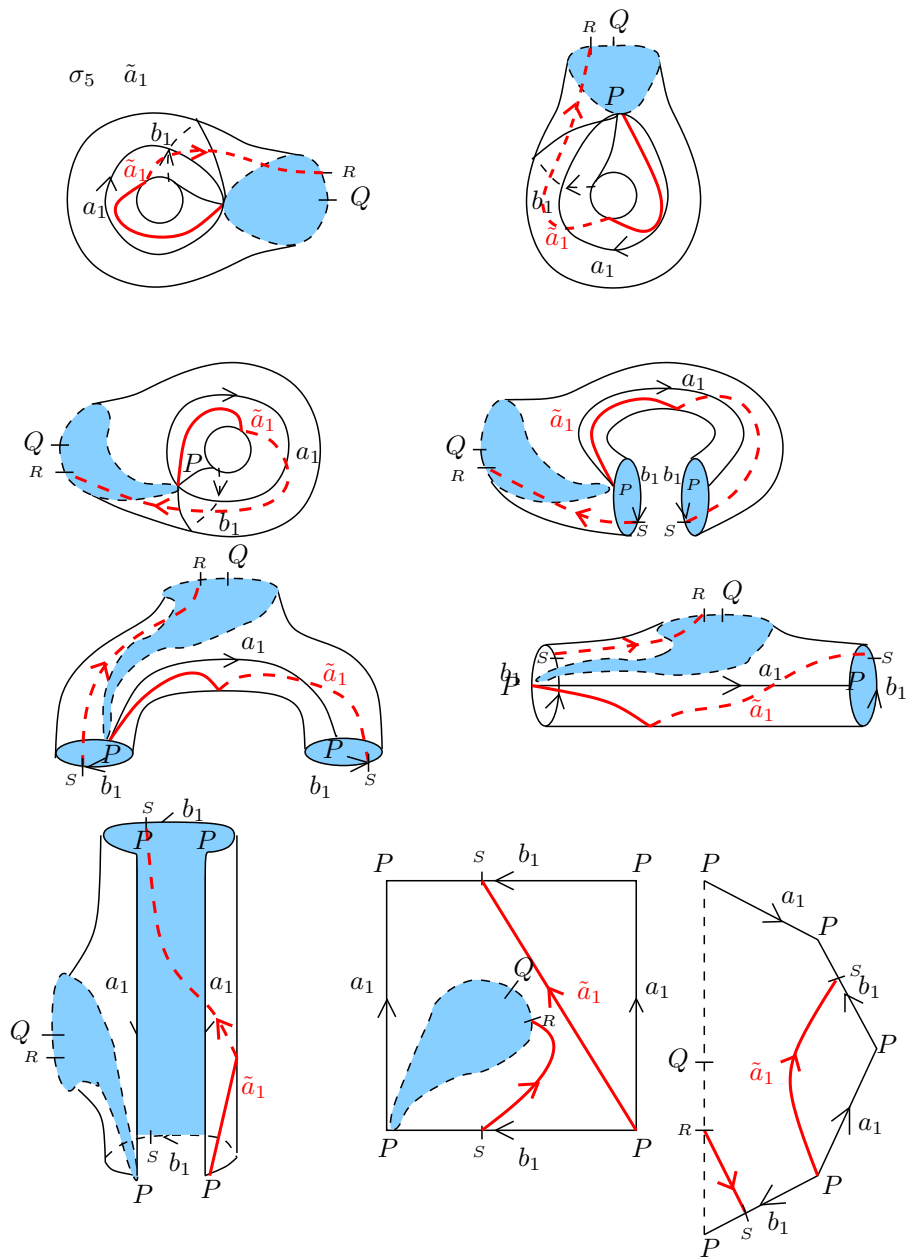


Figura 2.21: Deformación de la asa 1 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

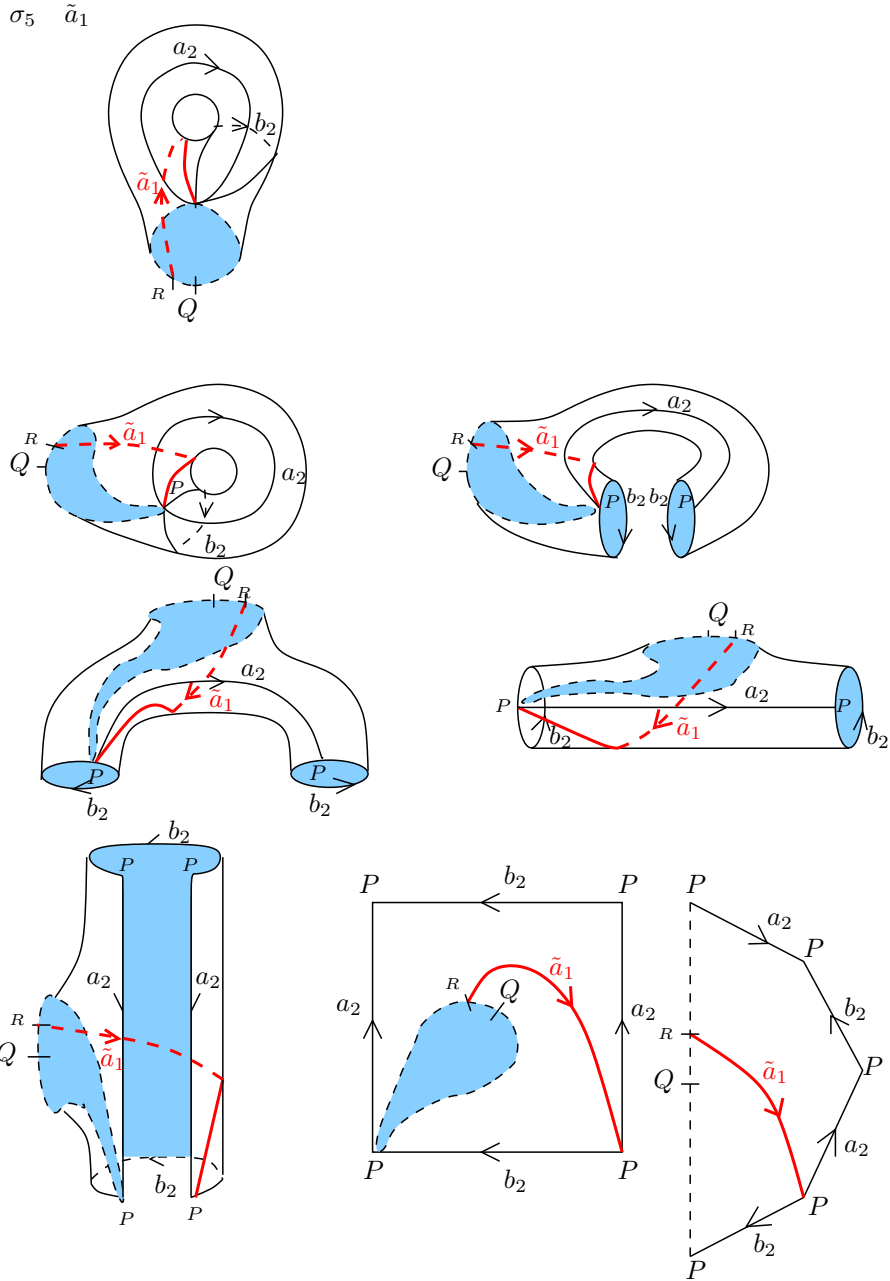
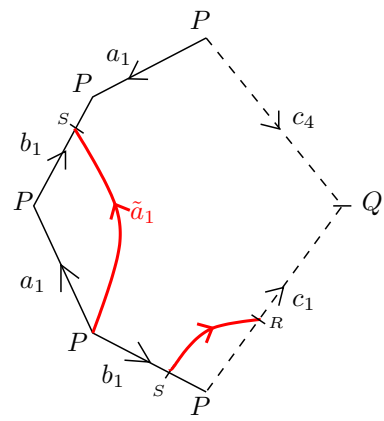


Figura 2.22: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 1



ASA 2

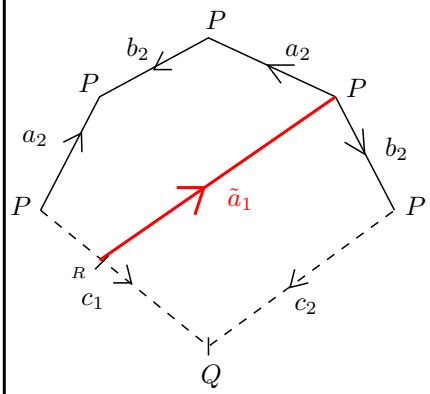
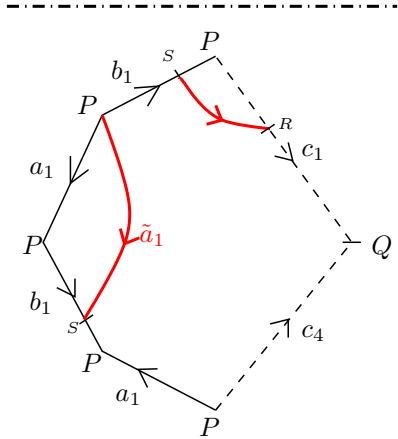
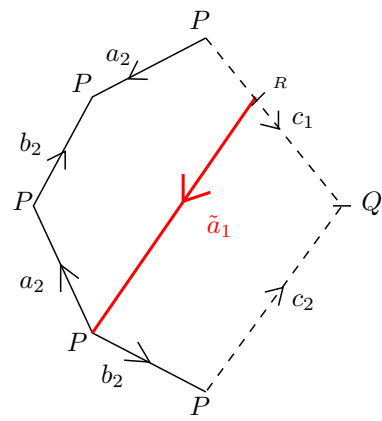
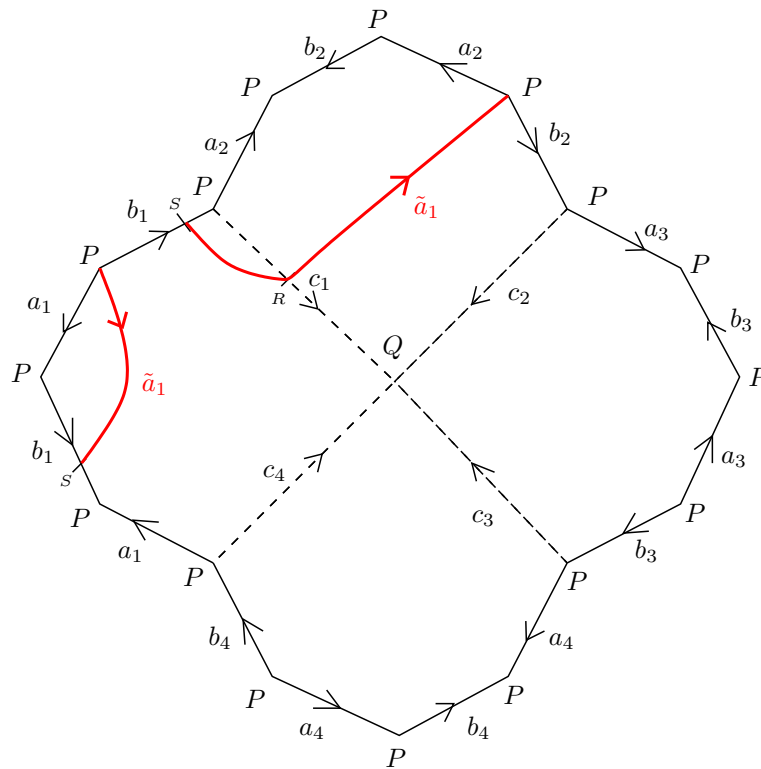


Figura 2.23: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 1, aplicamos simetría especular y en la Asa 2 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_5 \tilde{a}_1$



$$\tilde{a}_1 = a_1 b_1 a_2 b_2^{-1} a_2^{-1}$$

Figura 2.24: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

$\sigma_5 \tilde{a}_2$

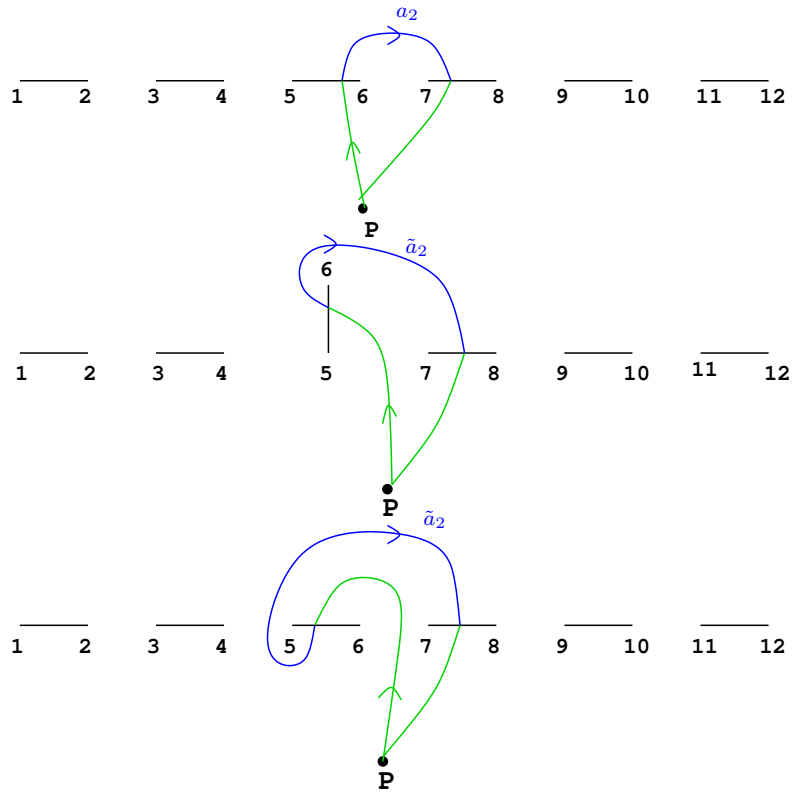


Figura 2.25: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

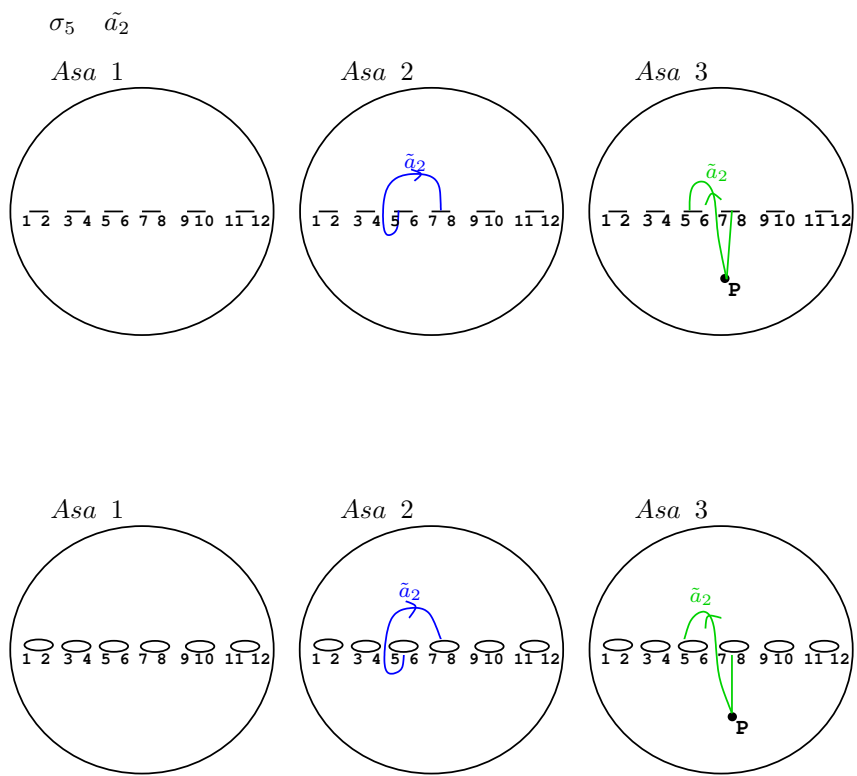


Figura 2.26: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

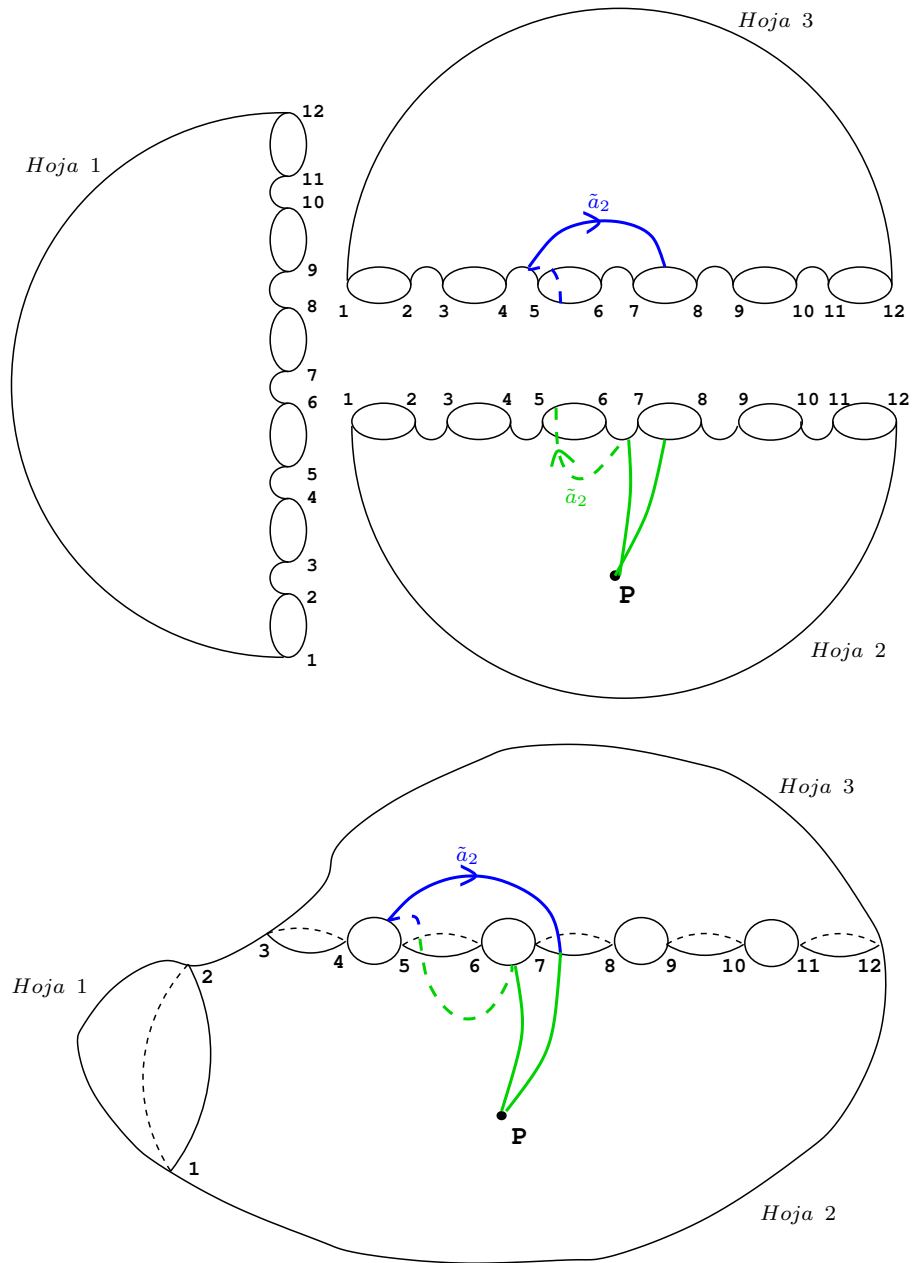


Figura 2.27: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

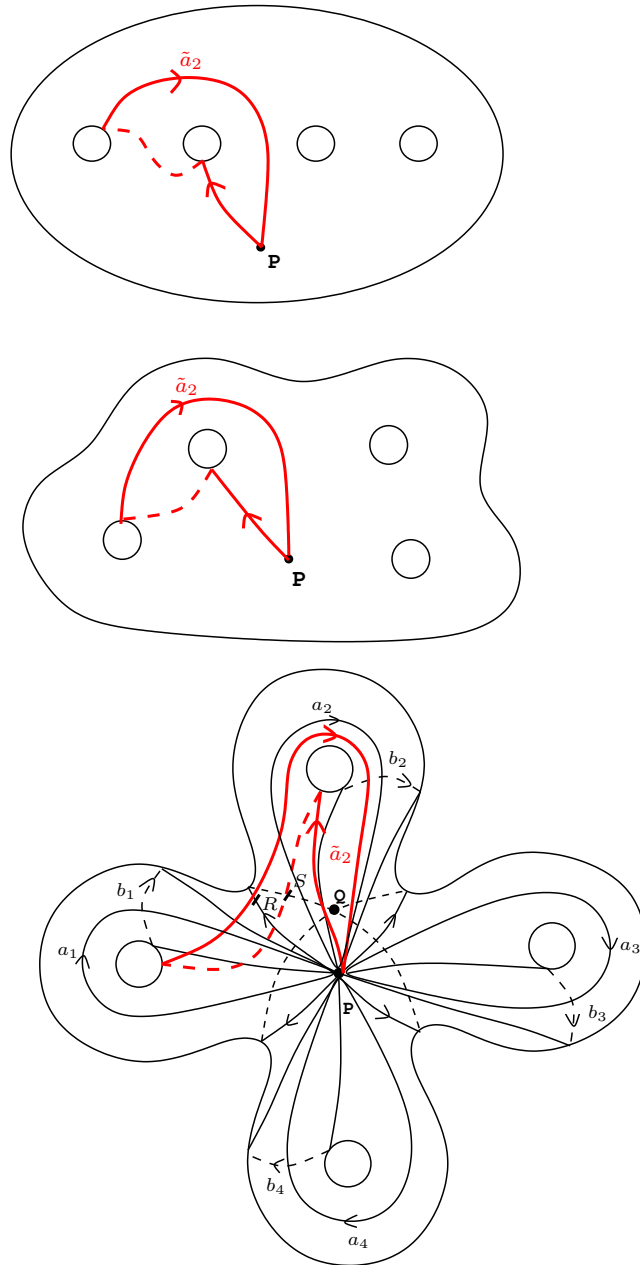


Figura 2.28: Camino imagen \tilde{c} en la esfera con 4 asas.

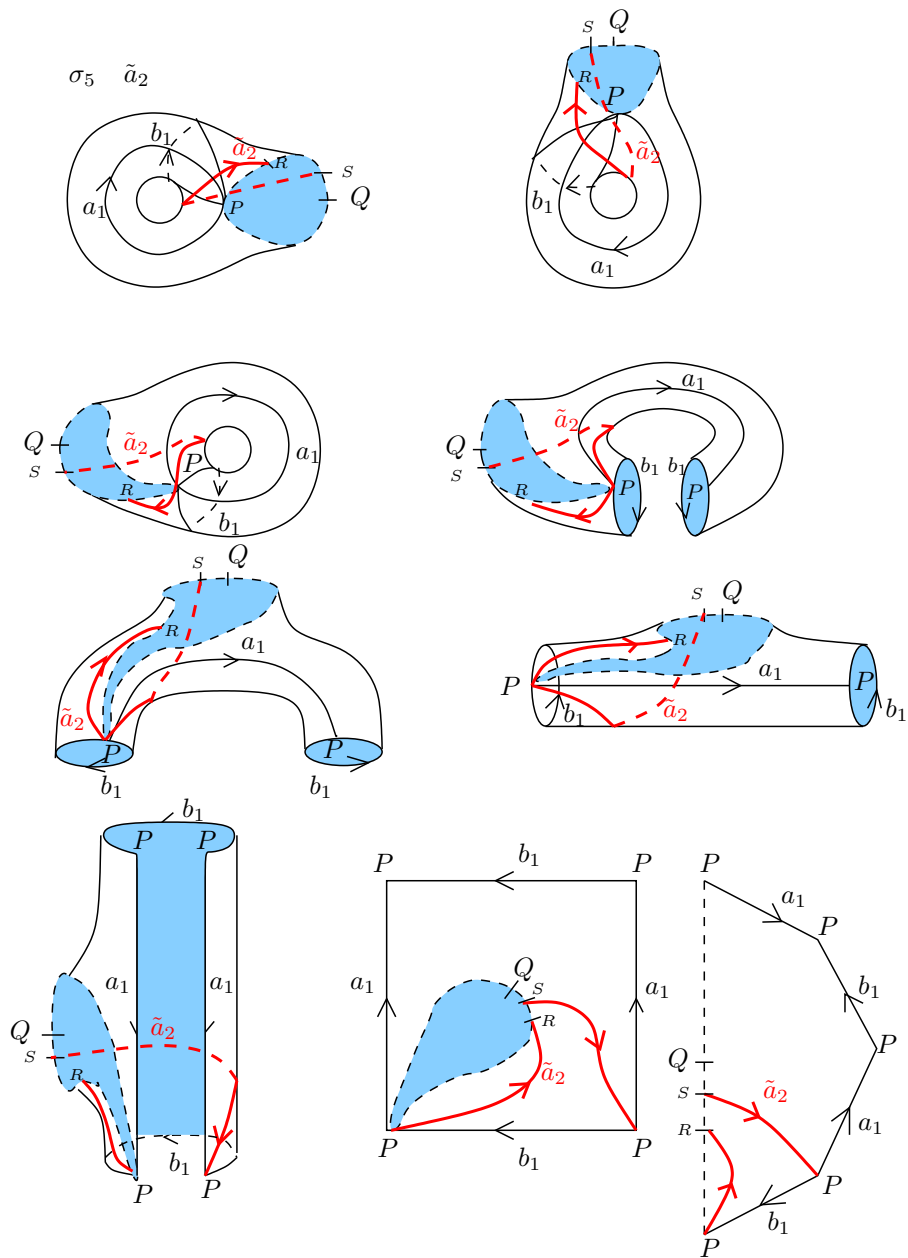


Figura 2.29: Deformación de la asa 1 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

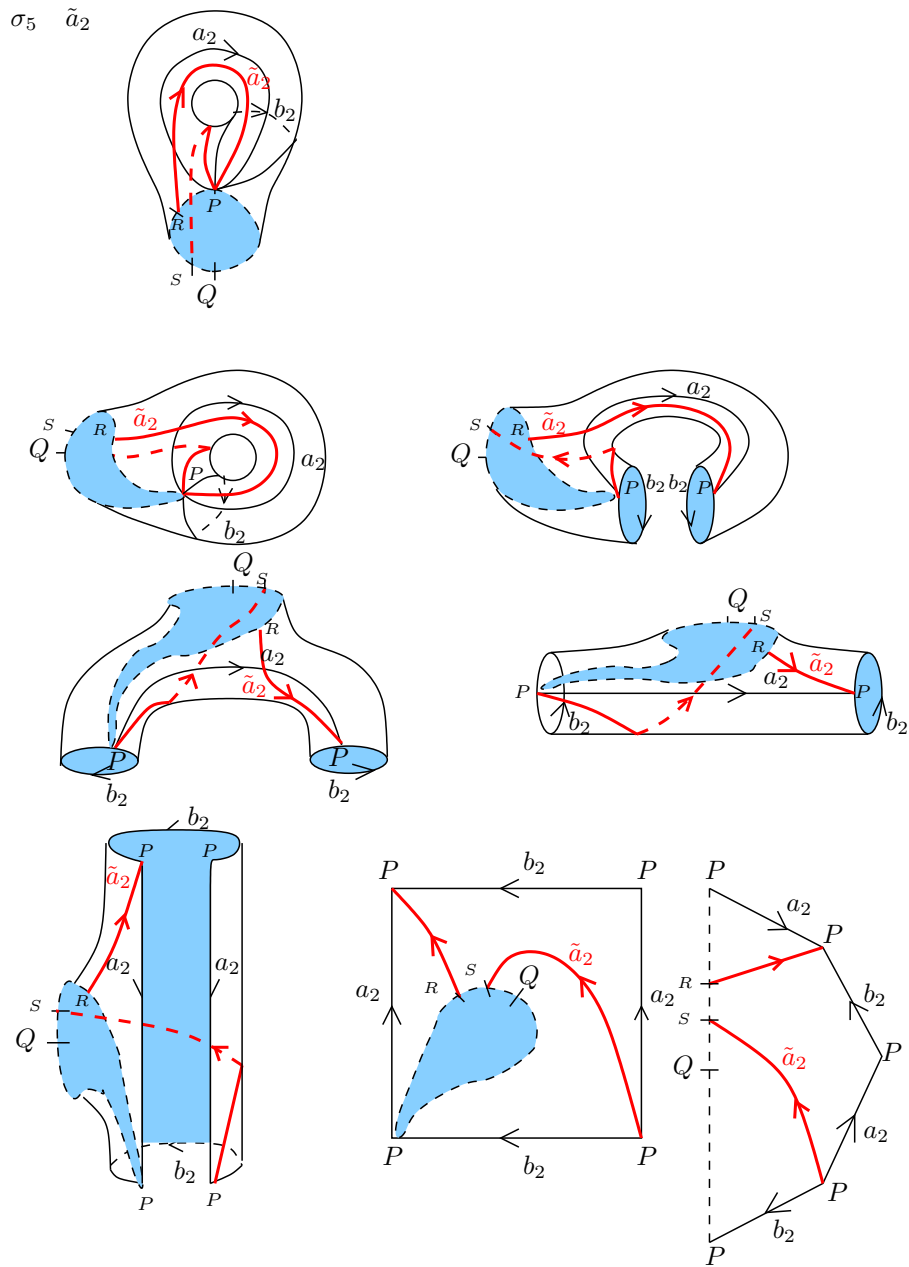
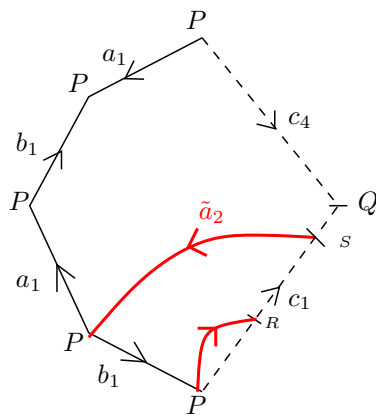


Figura 2.30: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 1



ASA 2

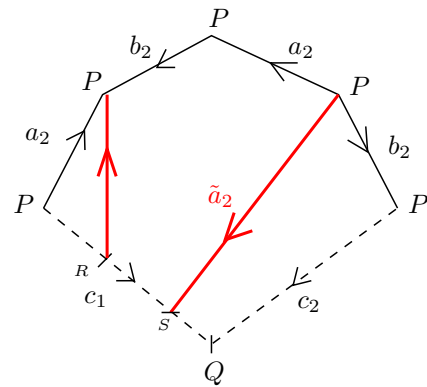
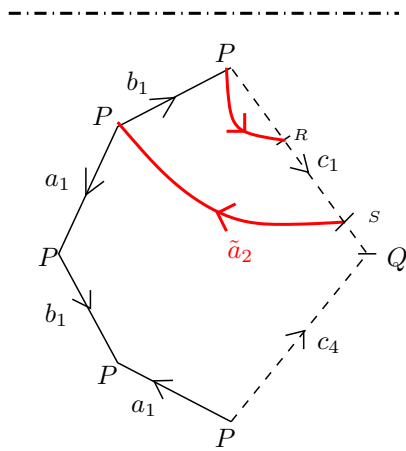
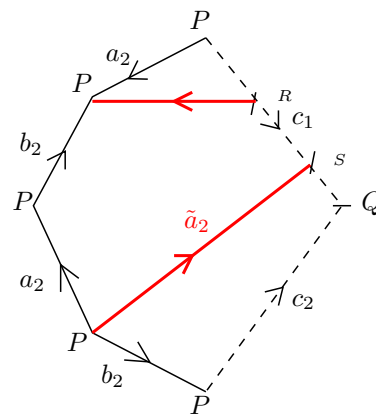
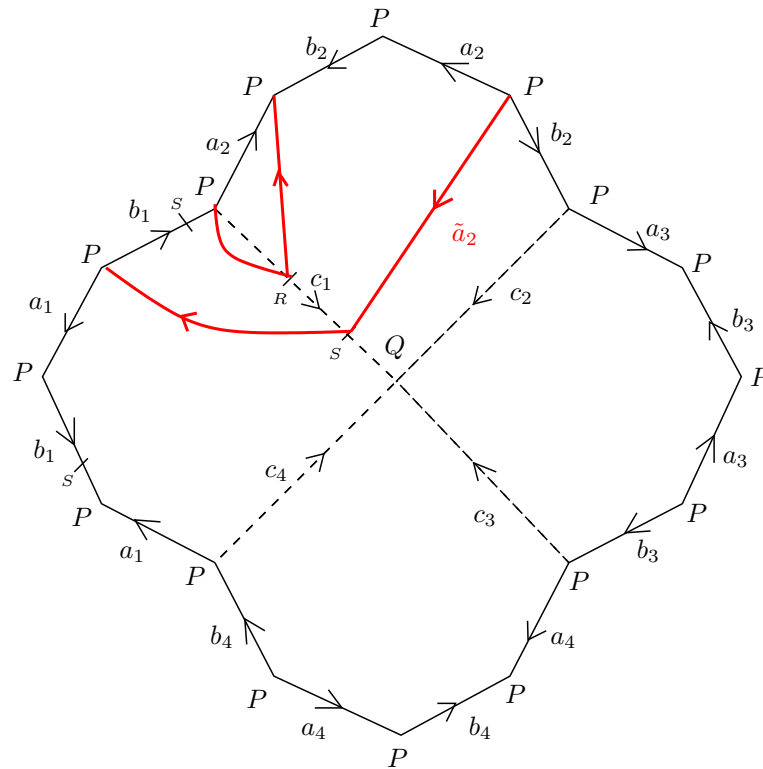


Figura 2.31: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 1, aplicamos simetría especular y en la Asa 2 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_5 \tilde{a}_2$



$$\tilde{a}_2 = a_2 b_2 a_2^{-1} b_1^{-1} a_2$$

Figura 2.32: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

$\sigma_5 \tilde{b}_1$

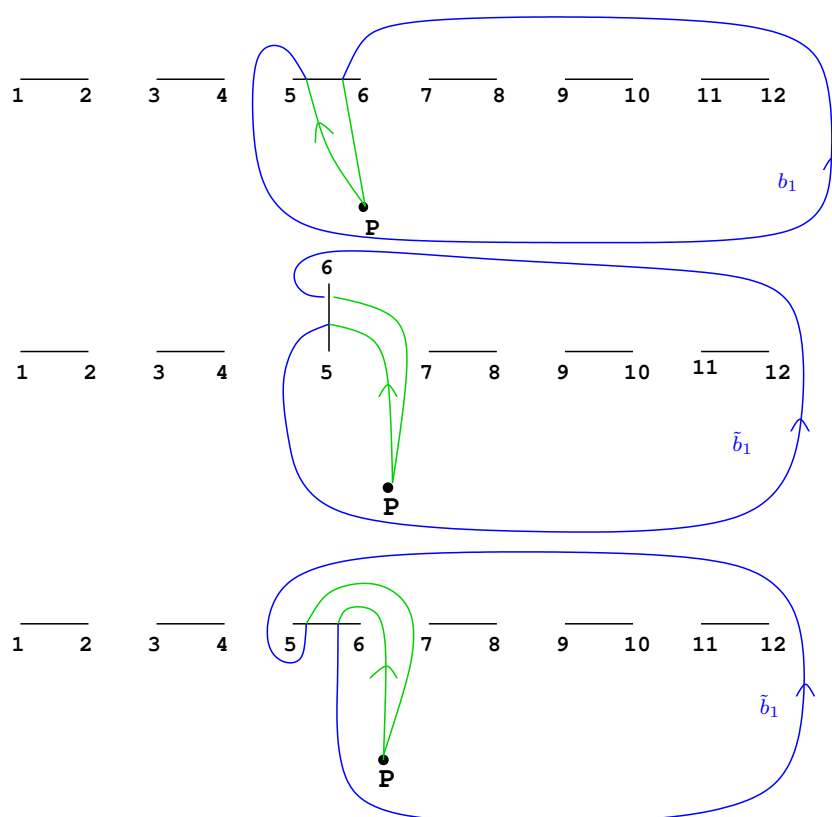


Figura 2.33: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

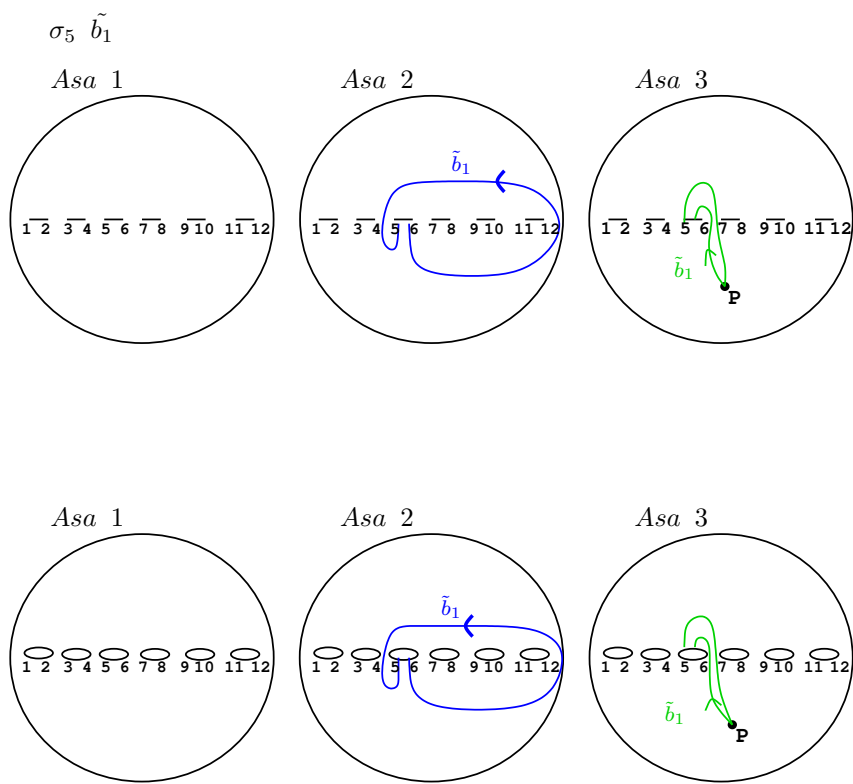


Figura 2.34: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

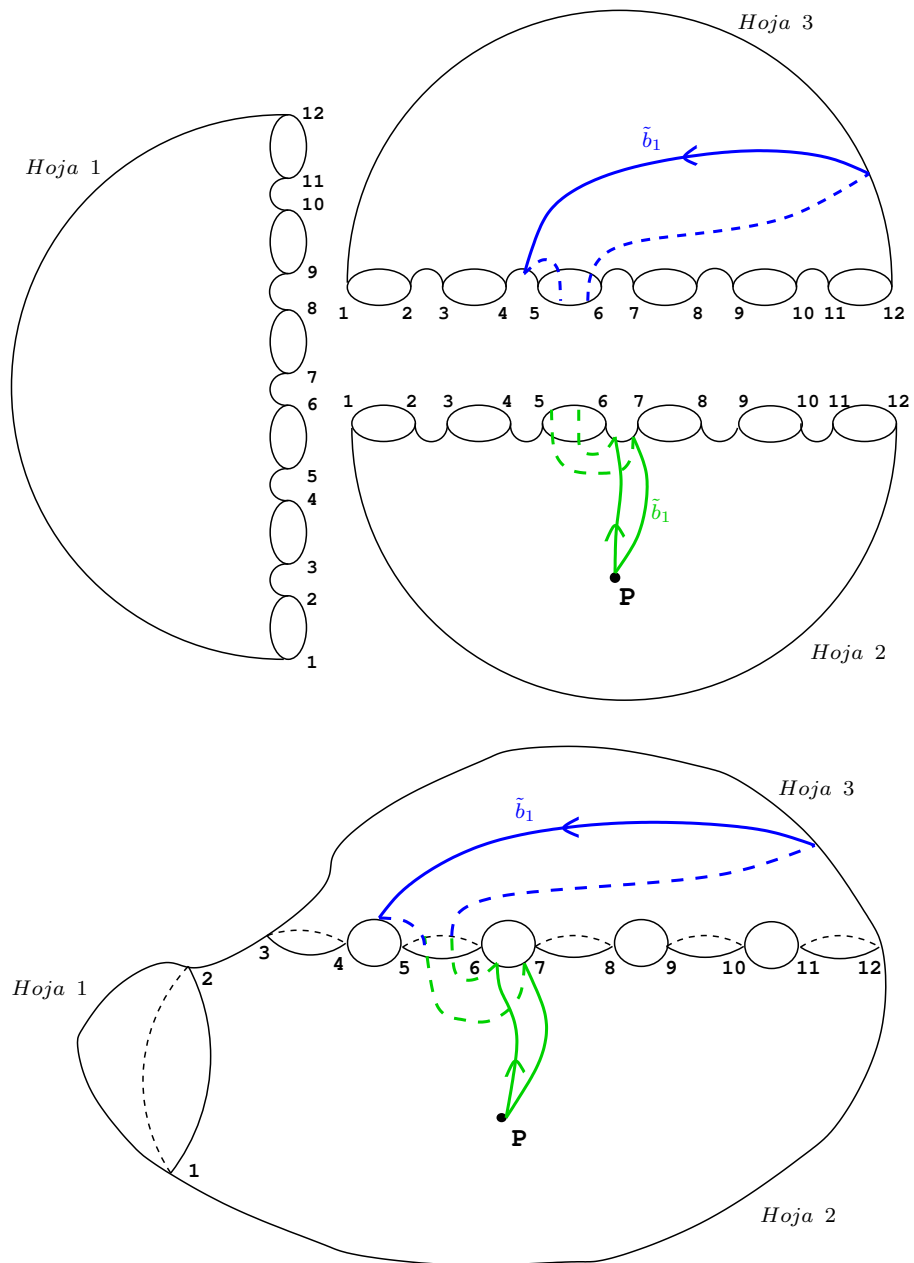


Figura 2.35: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

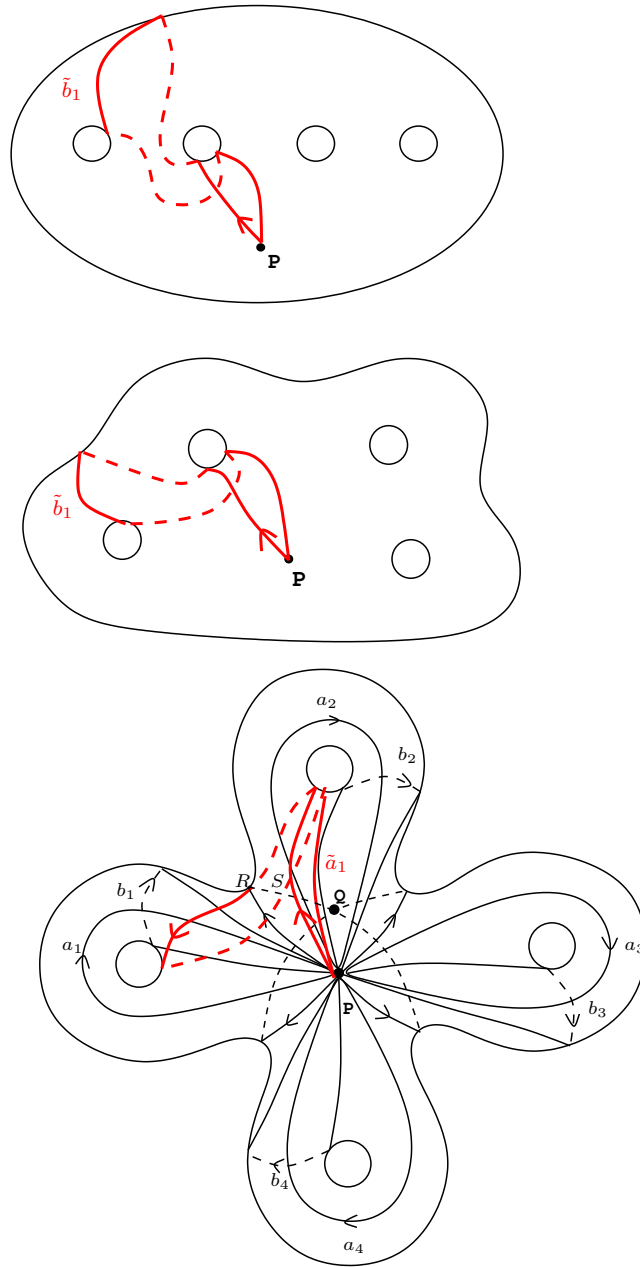


Figura 2.36: Camino imagen \tilde{c} en la esfera con 4 asas.

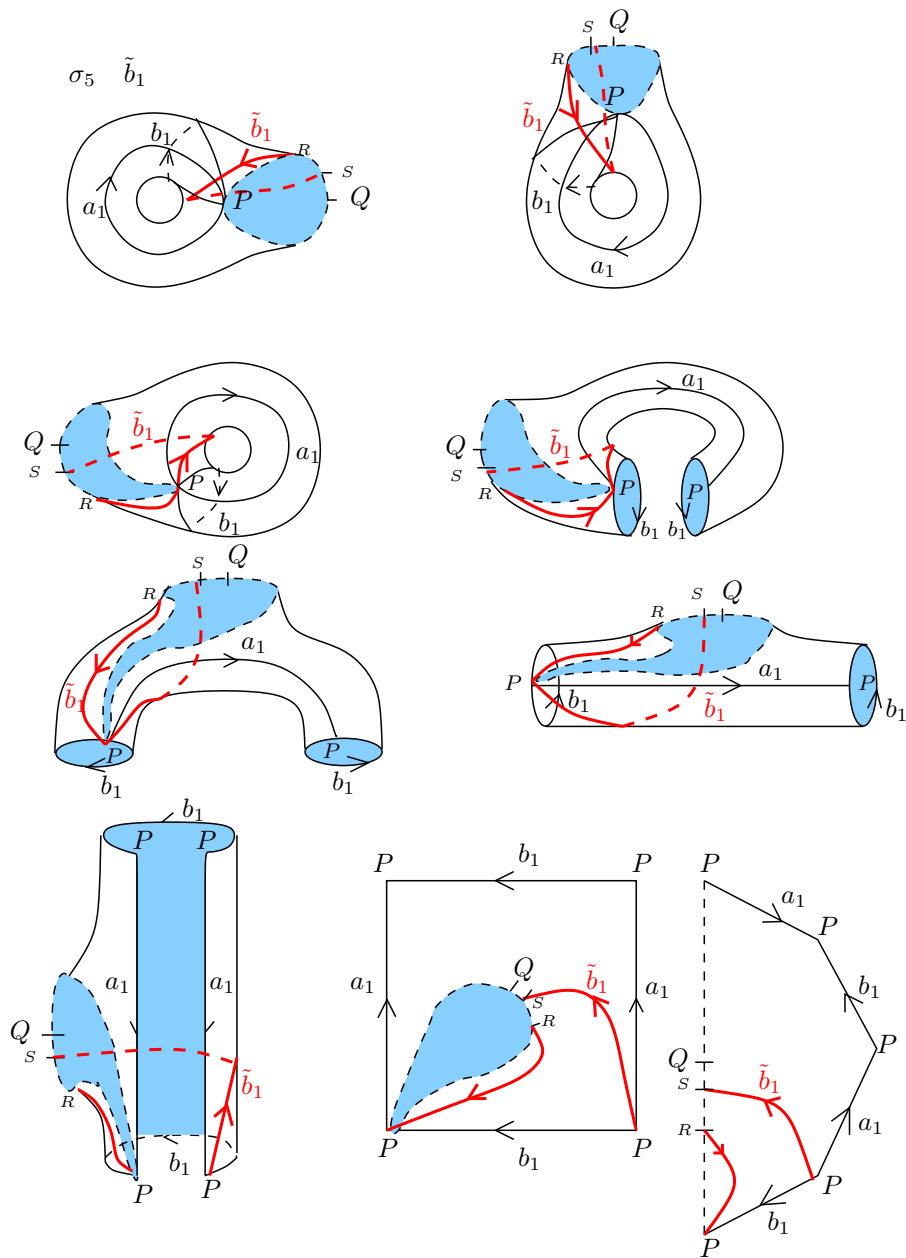


Figura 2.37: Deformación de la asa 1 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

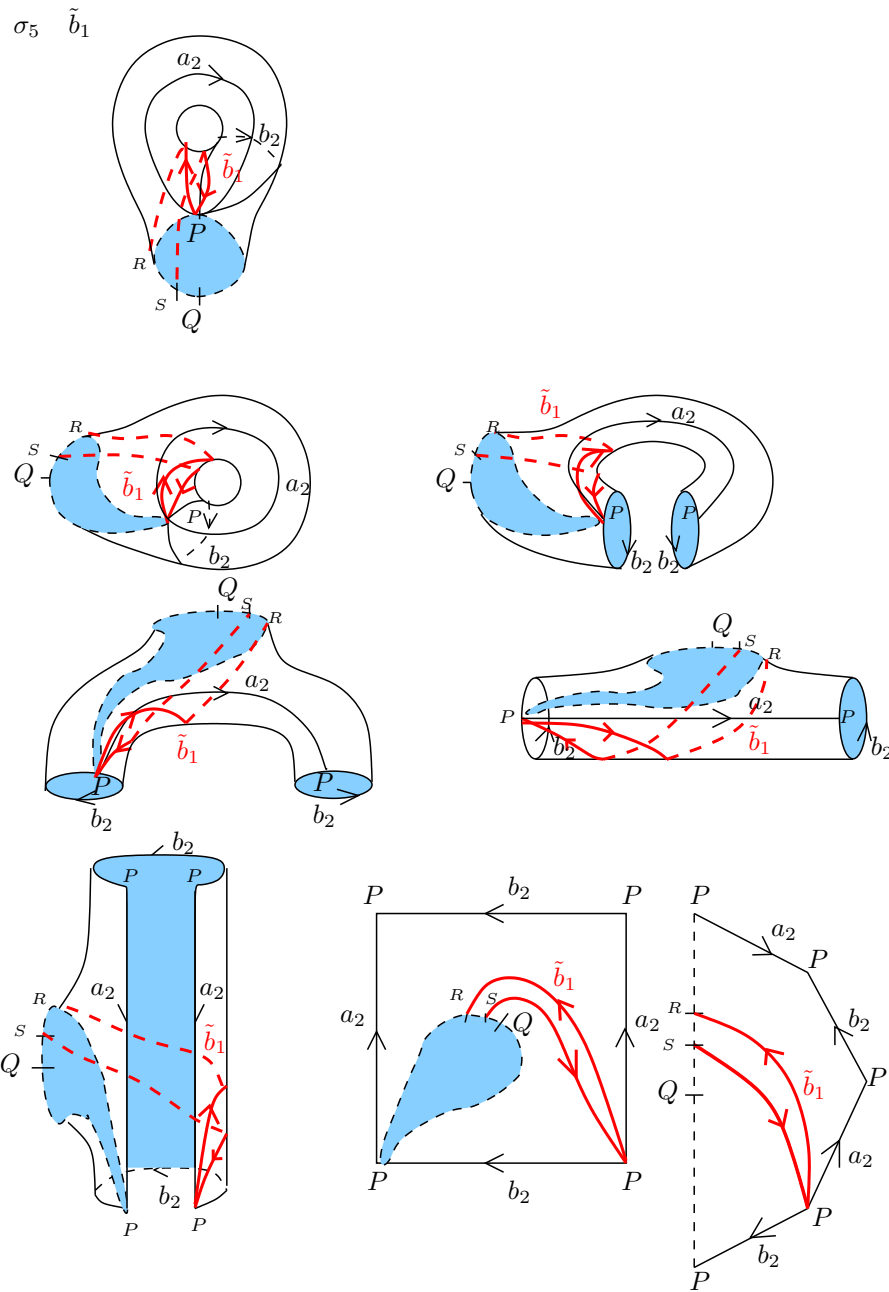
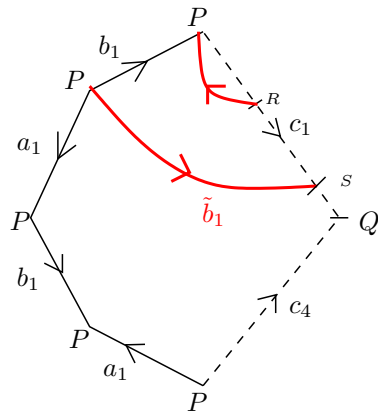
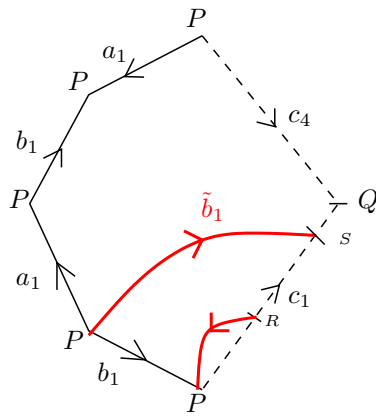


Figura 2.38: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 1



ASA 2

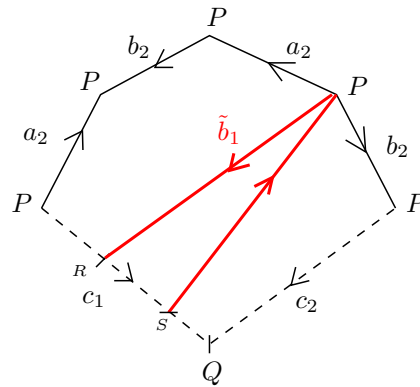
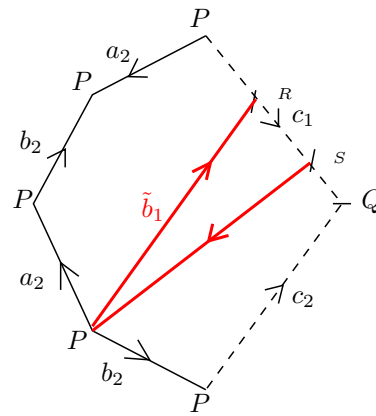
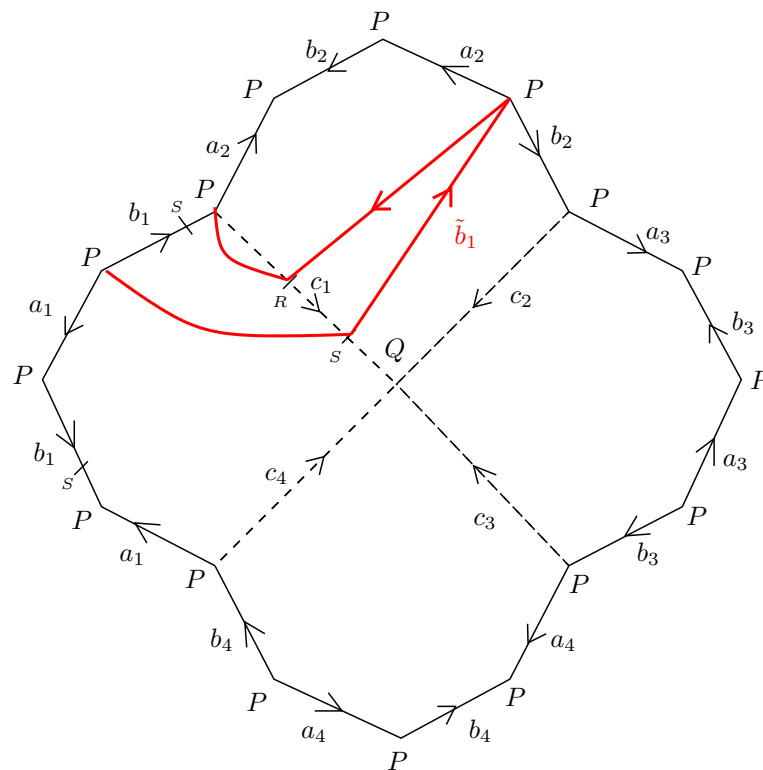


Figura 2.39: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 1, aplicamos simetría especular y en la Asa 2 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_5 \tilde{b}_1$



$$\tilde{b}_1 = a_2 b_2 a_2^{-1} b_1 a_2 b_2^{-1} a_2^{-1}$$

Figura 2.40: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

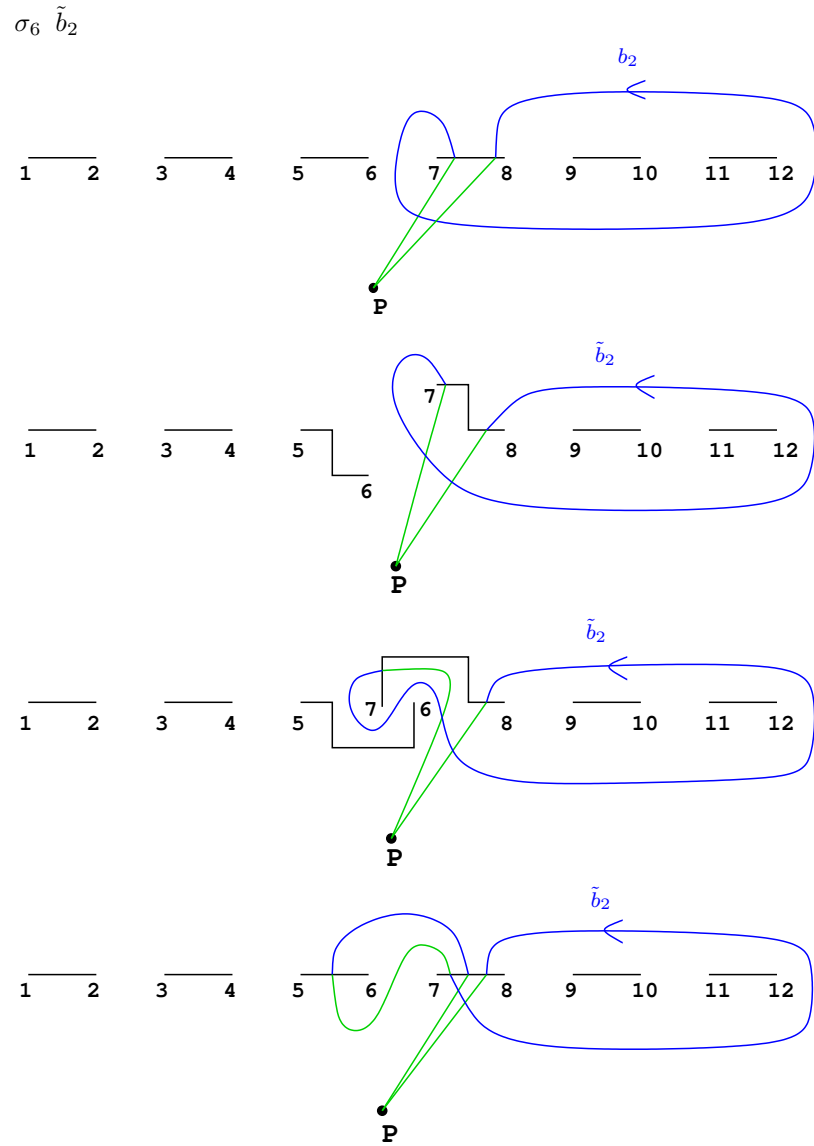


Figura 2.41: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

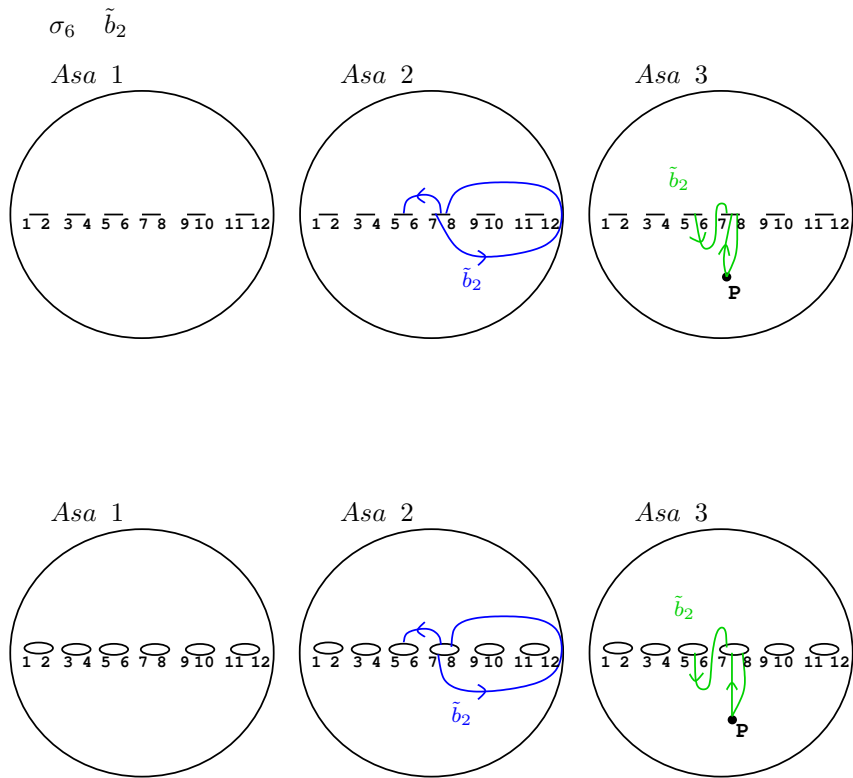


Figura 2.42: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

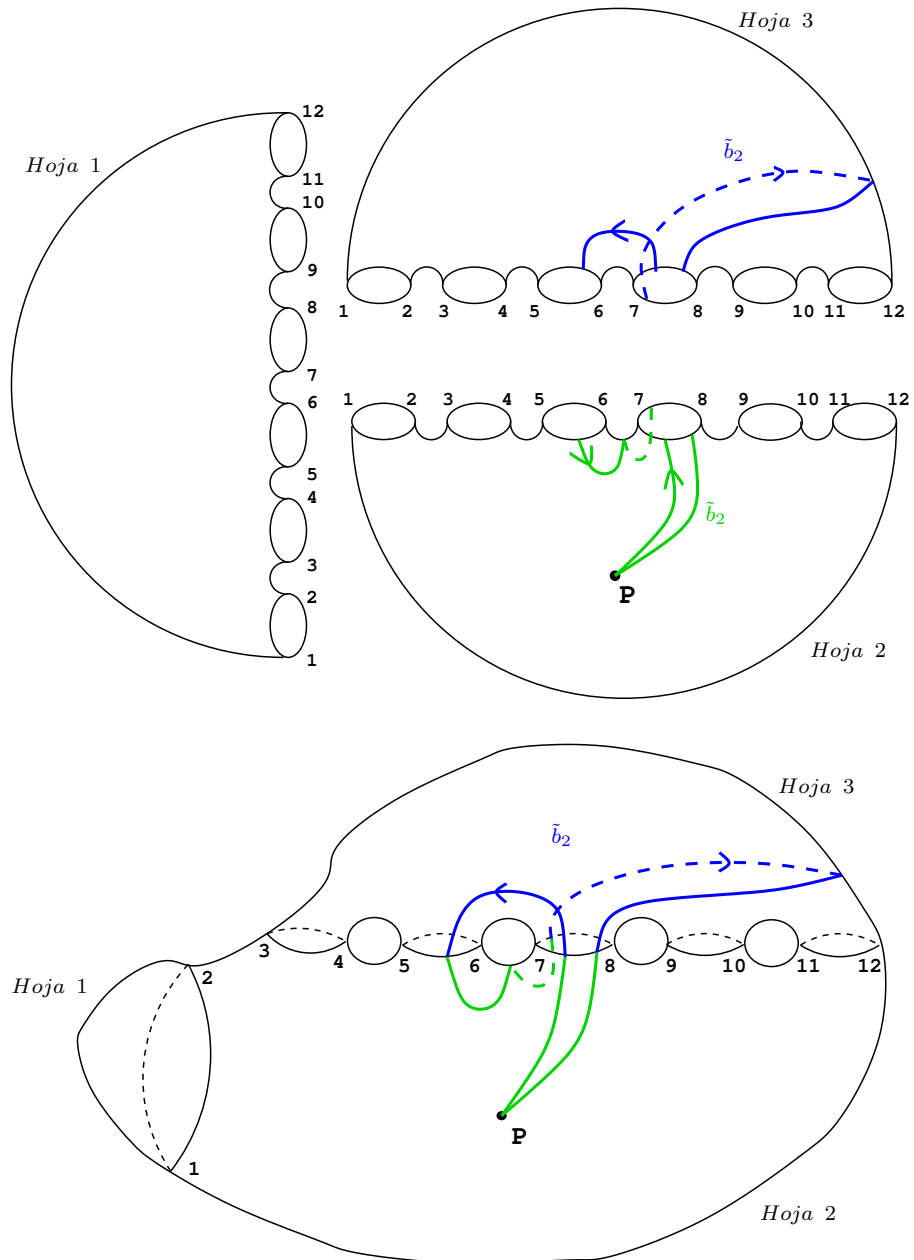


Figura 2.43: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

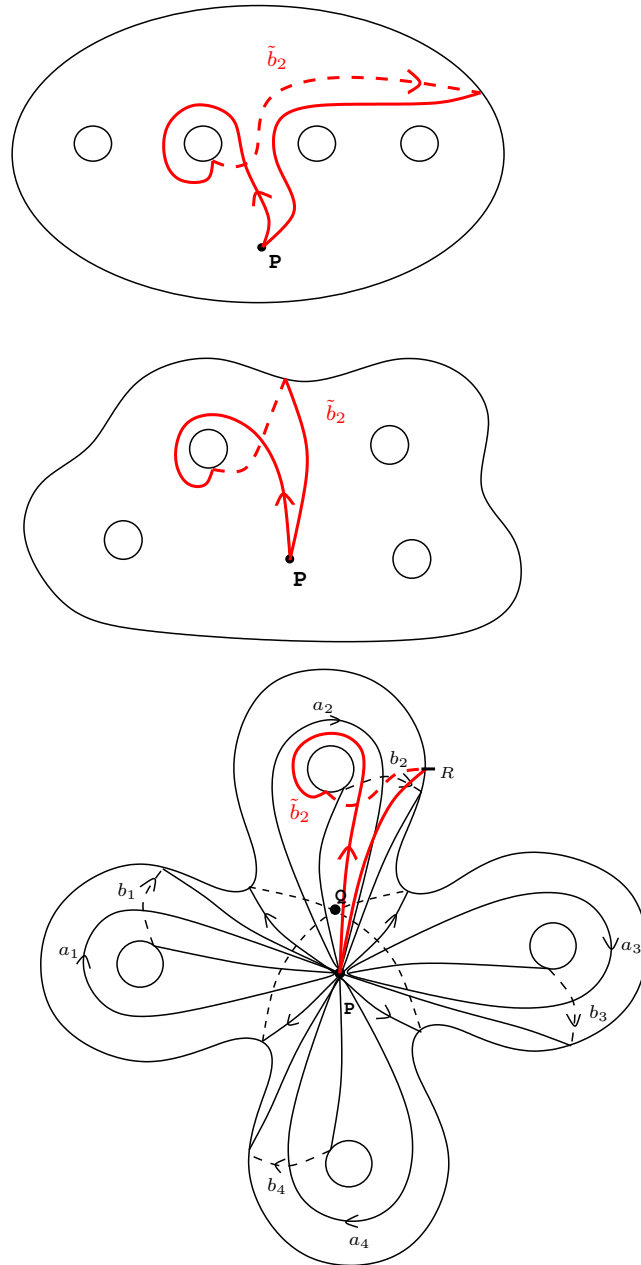


Figura 2.44: Camino imagen \tilde{c} en la esfera con 4 asas.

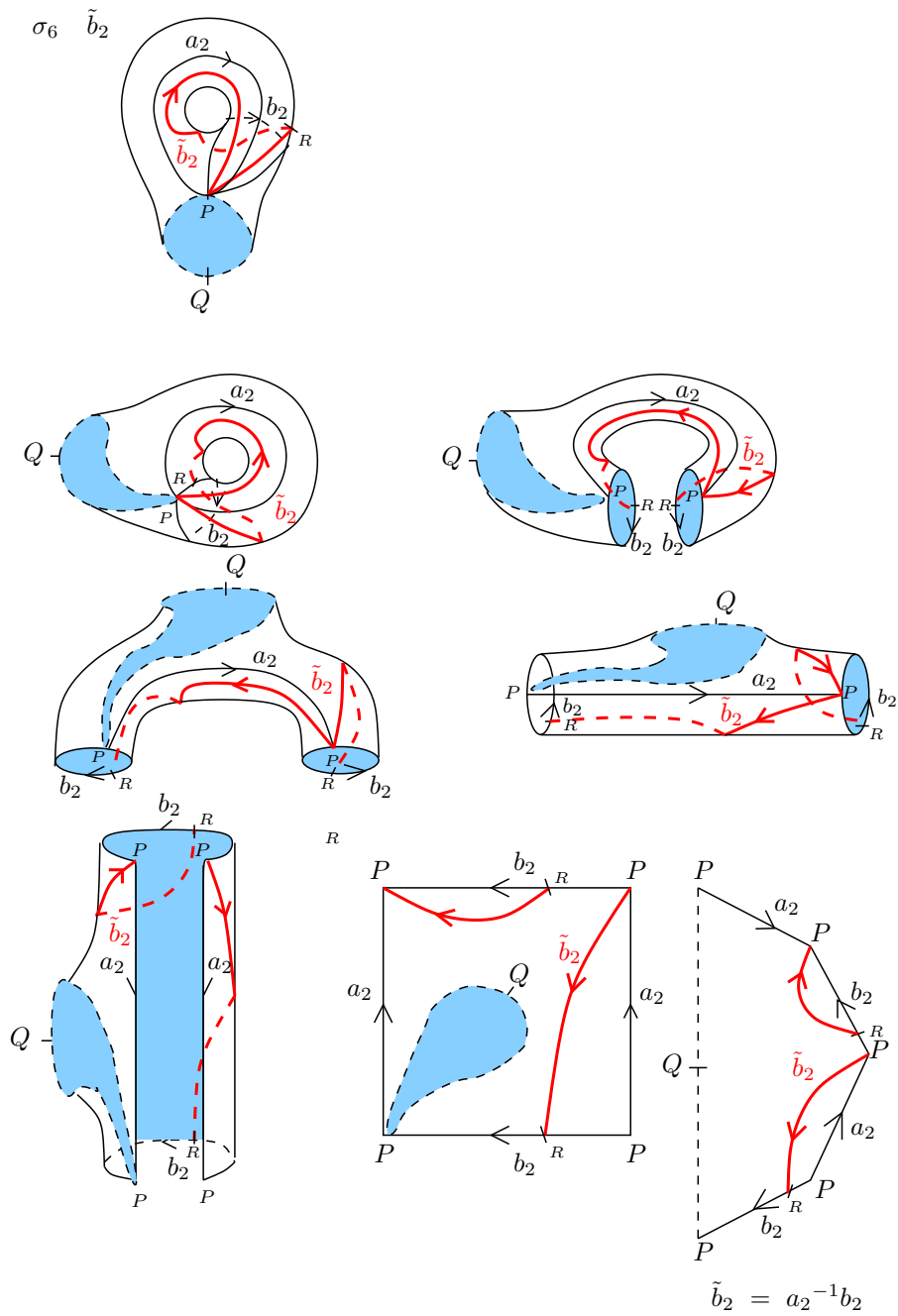


Figura 2.45: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

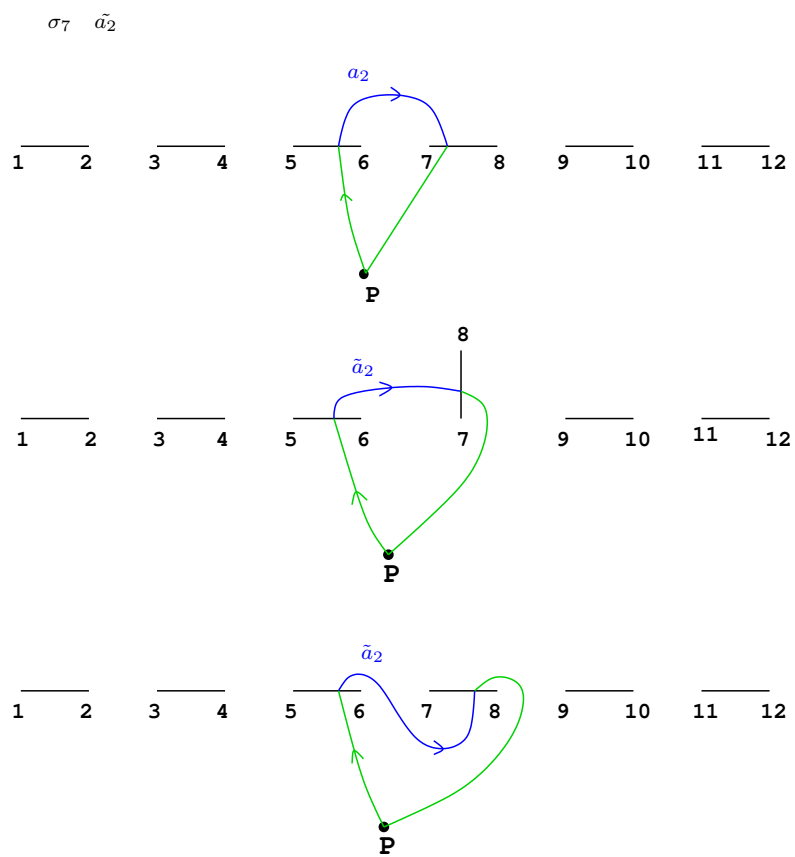


Figura 2.46: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

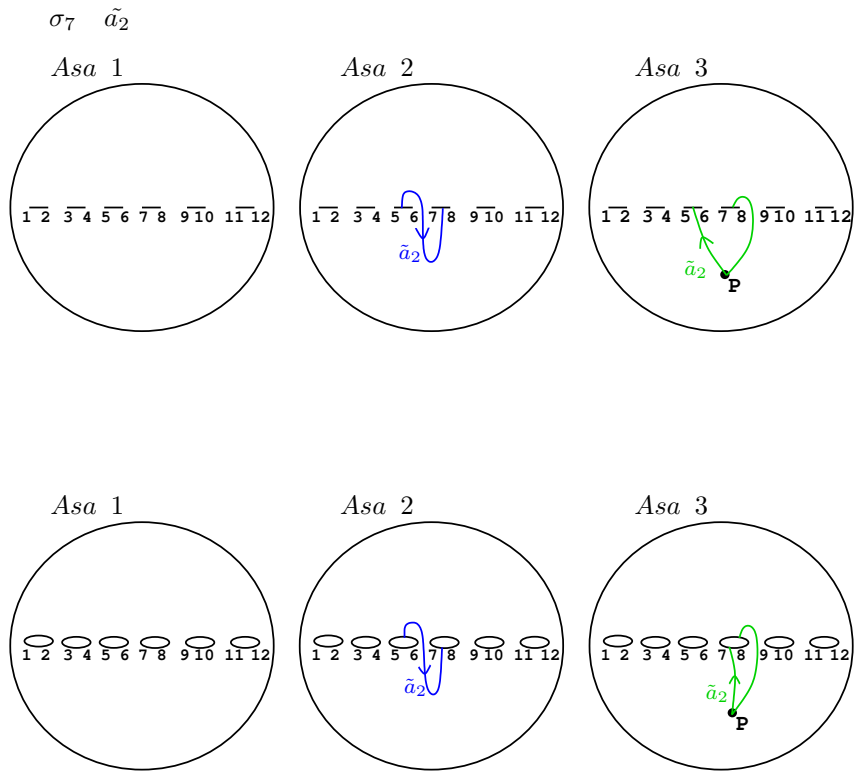


Figura 2.47: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

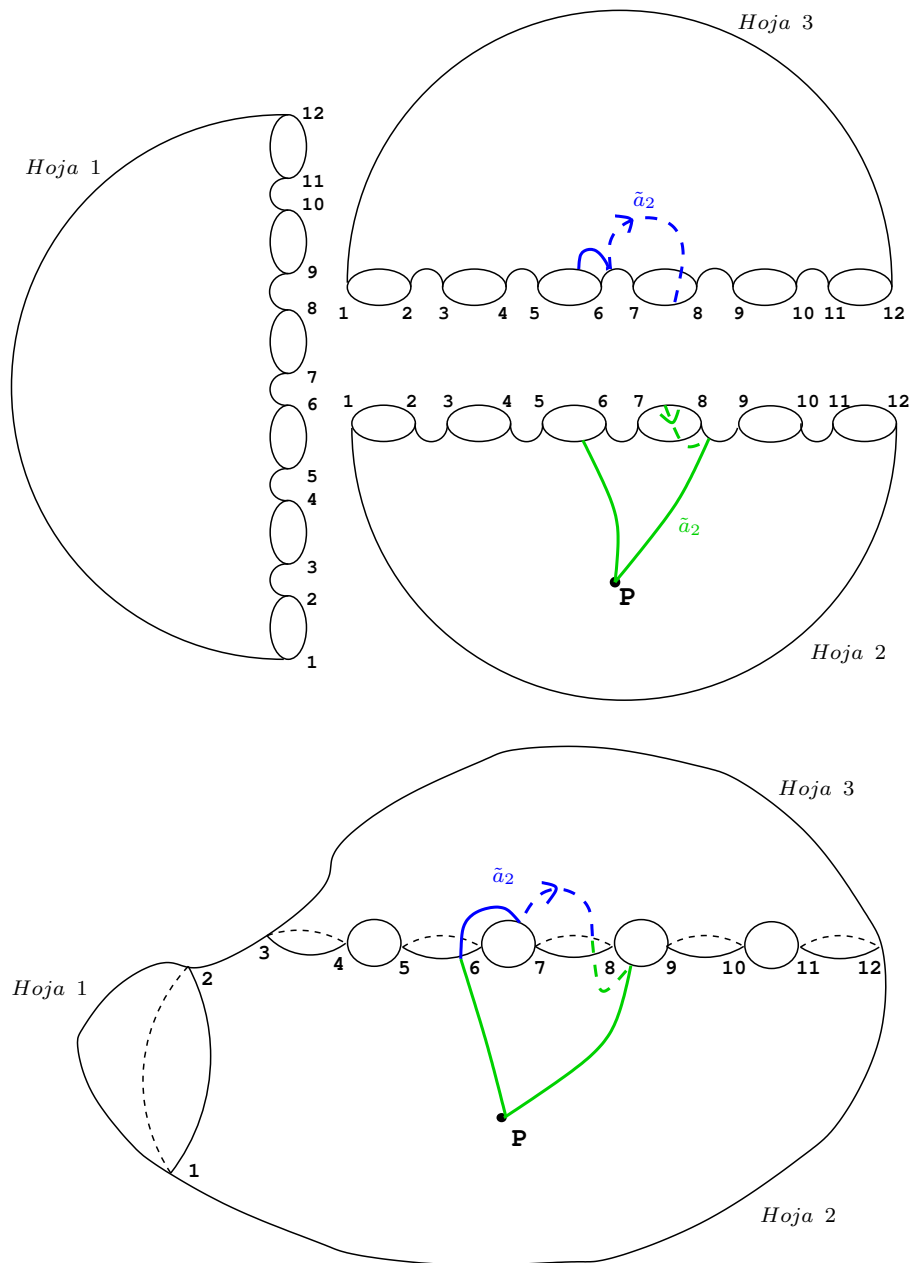


Figura 2.48: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

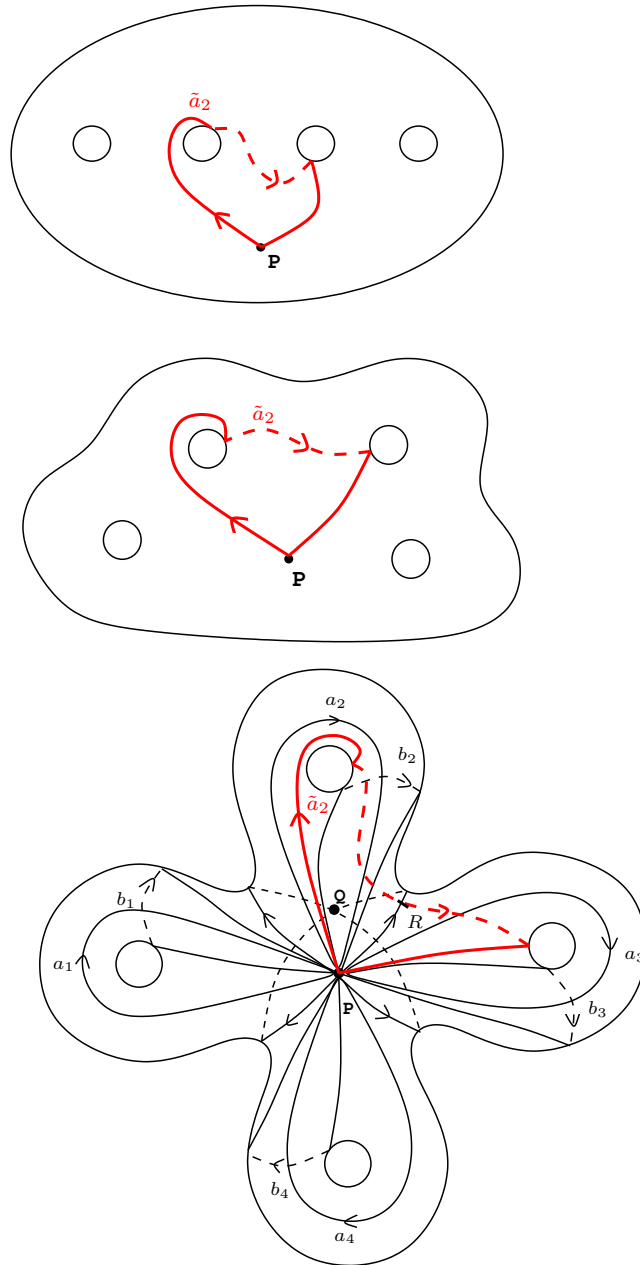


Figura 2.49: Camino imagen \tilde{c} en la esfera con 4 asas.

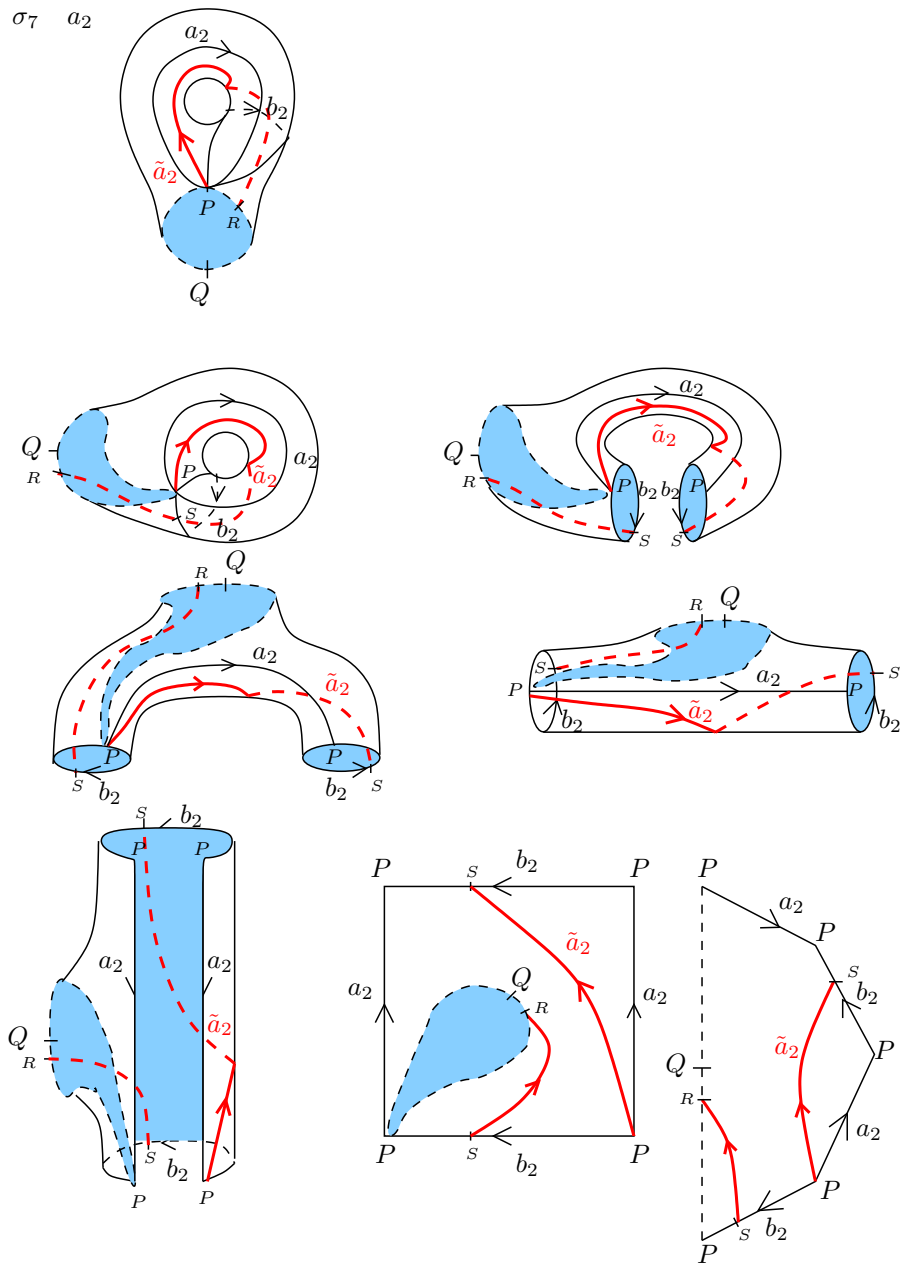


Figura 2.50: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

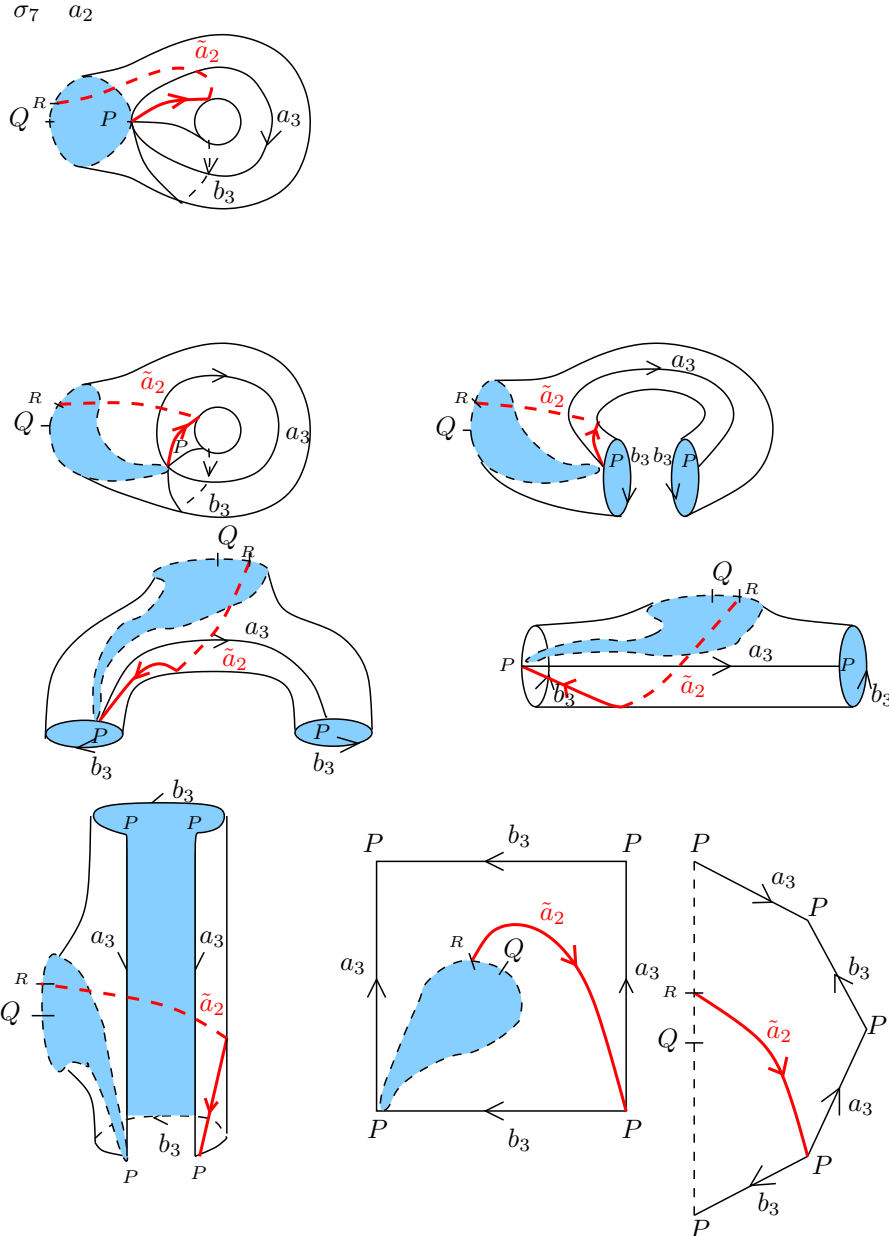
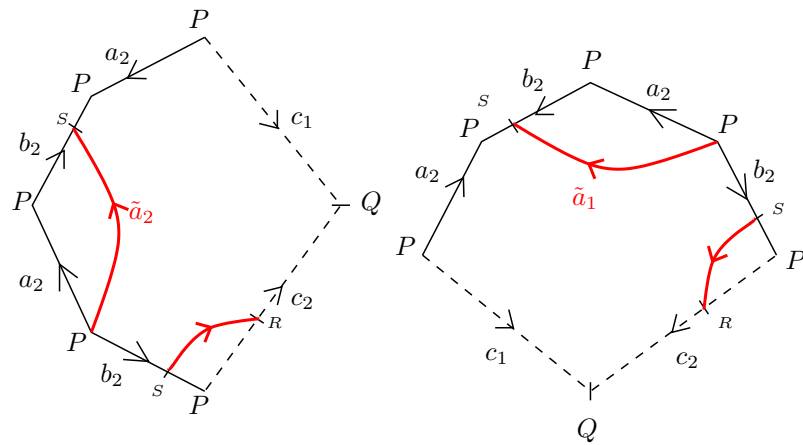


Figura 2.51: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 2



ASA 3

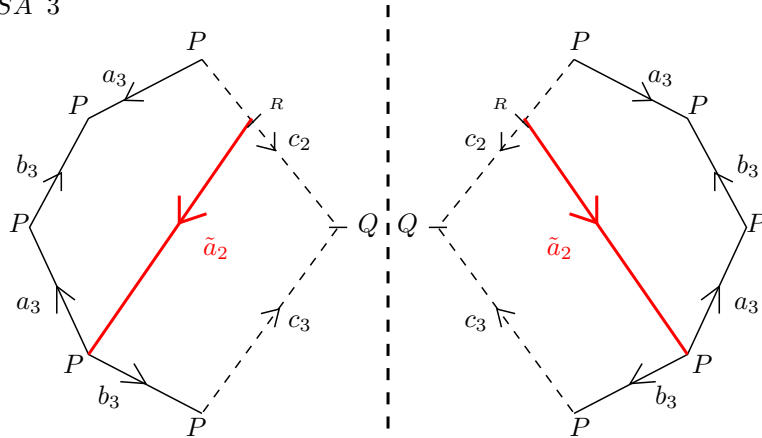
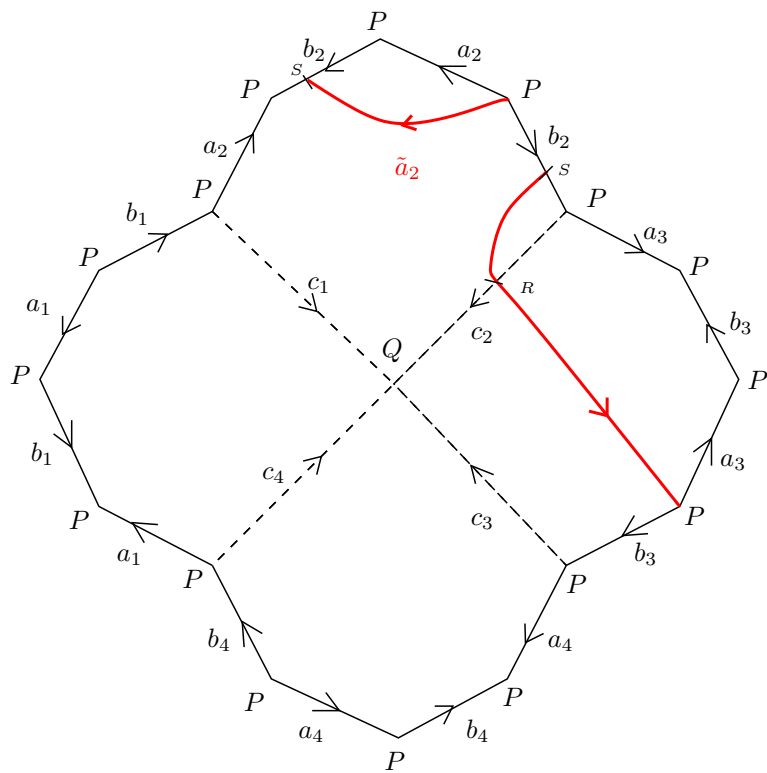


Figura 2.52: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 2, aplicamos simetría especular y luego rotación de 90° en sentido anti-horario, y en la Asa 3 aplicamos simetría especular.

$\sigma_7 \tilde{a}_2$



$$\tilde{a}_2 = a_2 b_2 a_3 b_3^{-1} a_3^{-1}$$

Figura 2.53: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

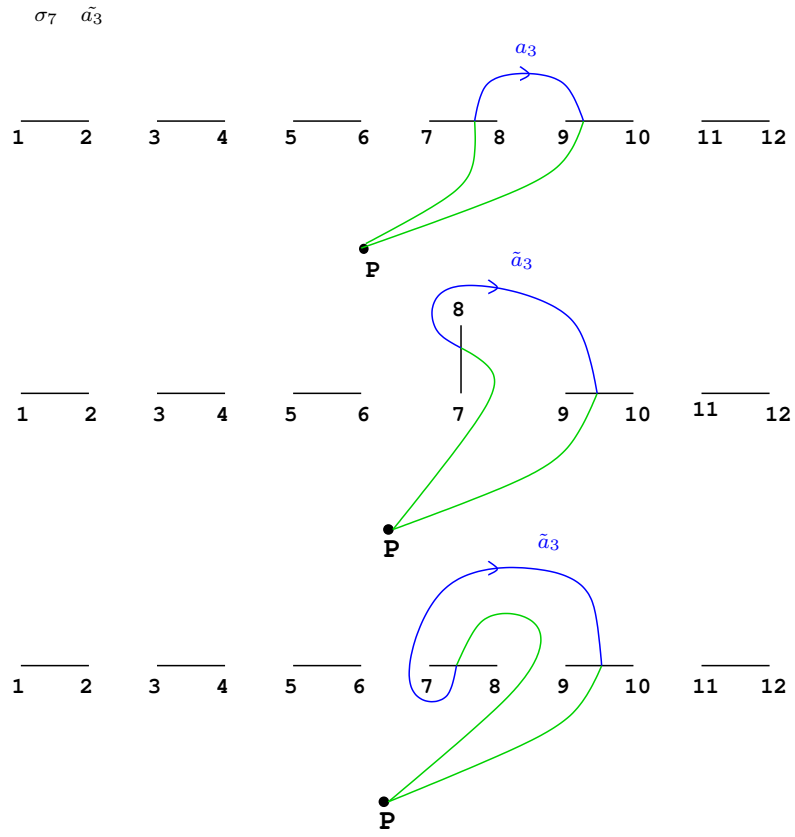


Figura 2.54: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

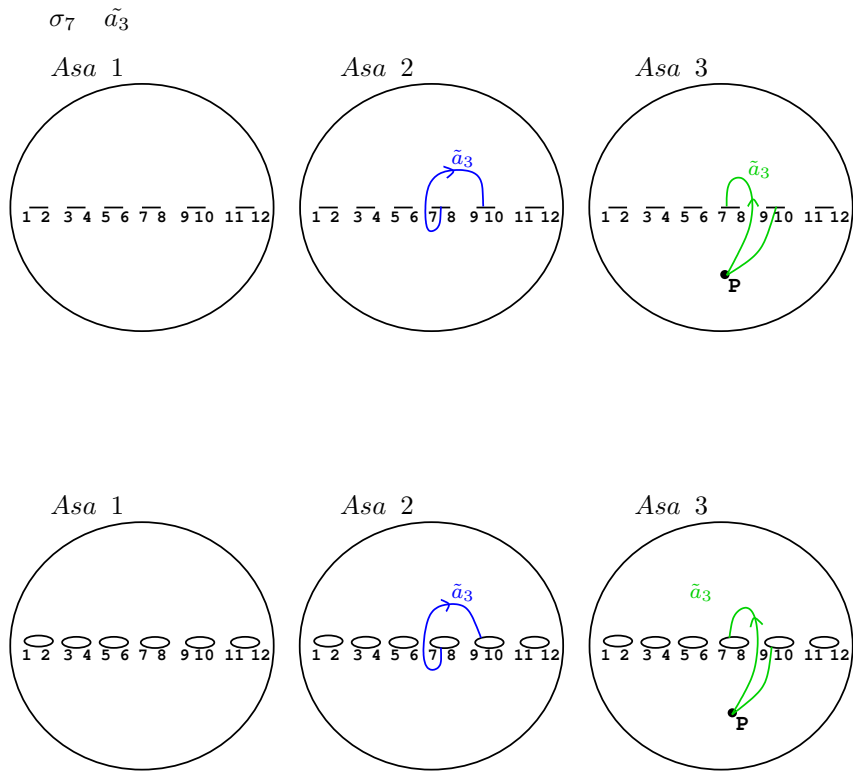


Figura 2.55: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

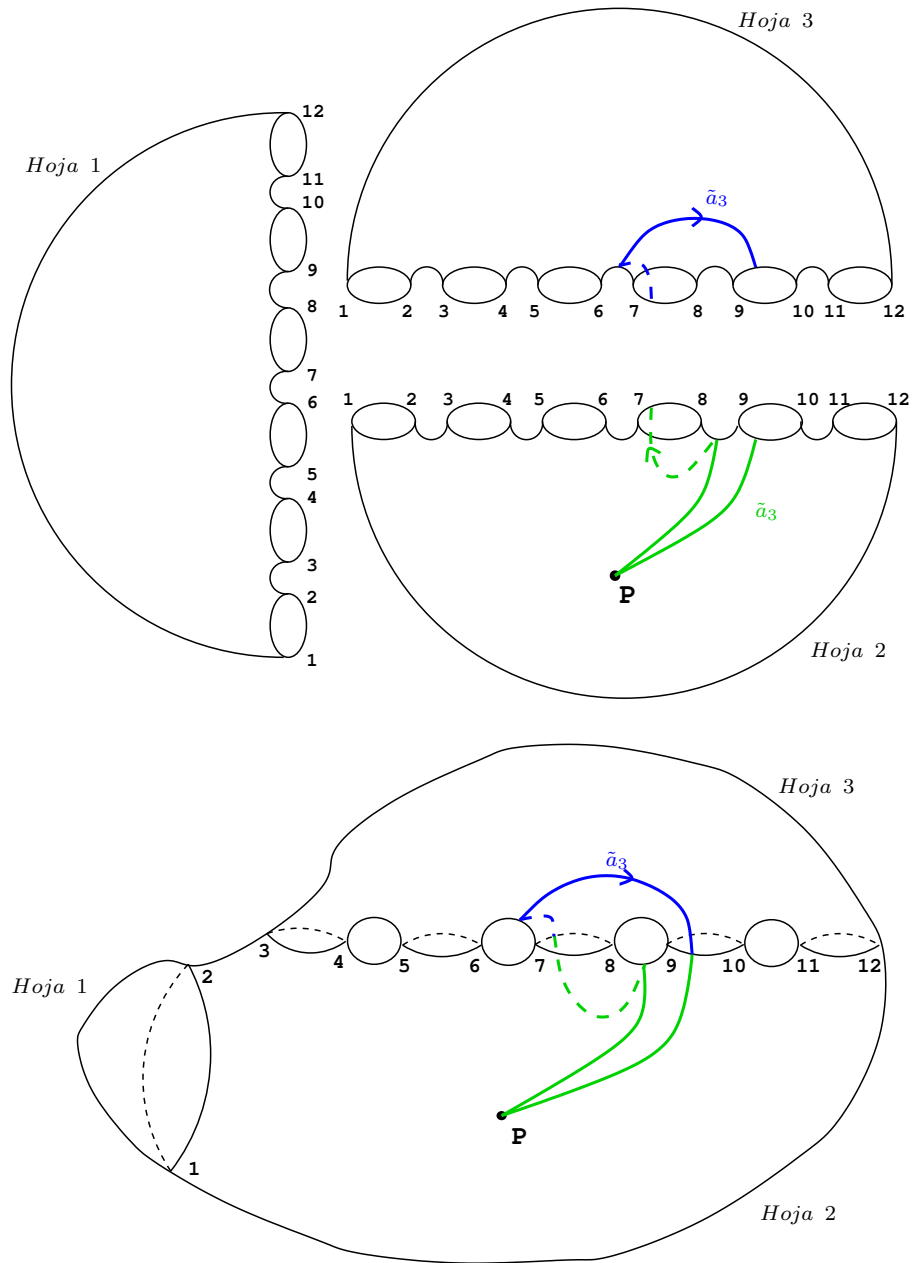


Figura 2.56: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

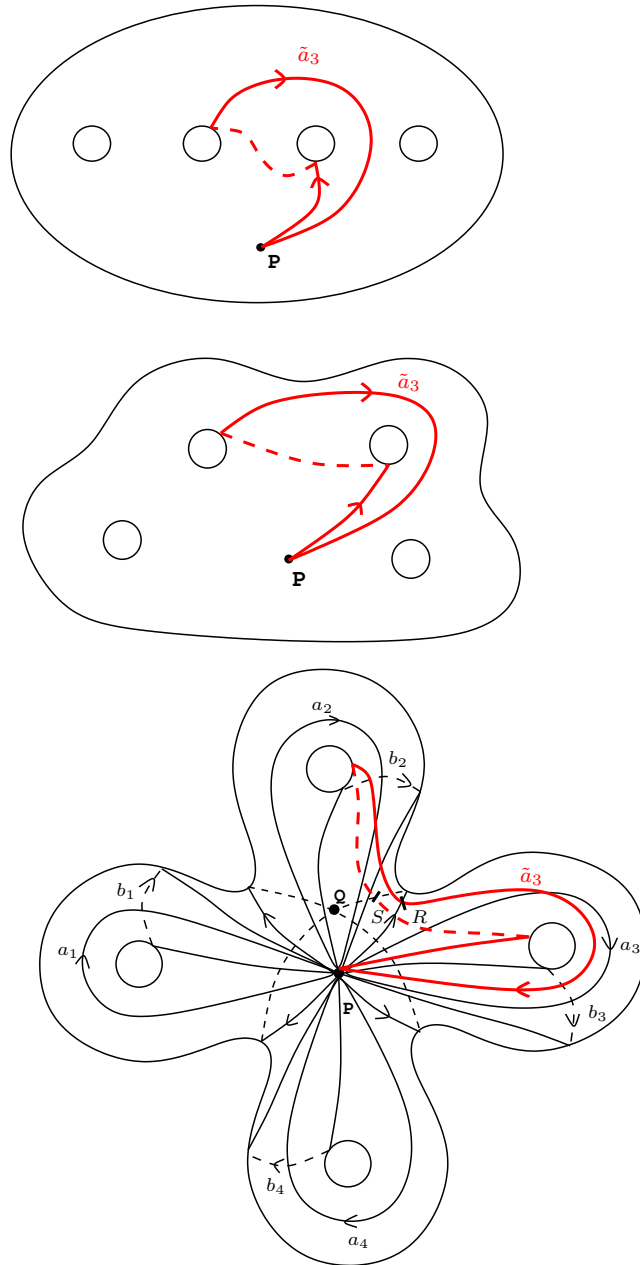


Figura 2.57: Camino imagen \tilde{c} en la esfera con 4 asas.

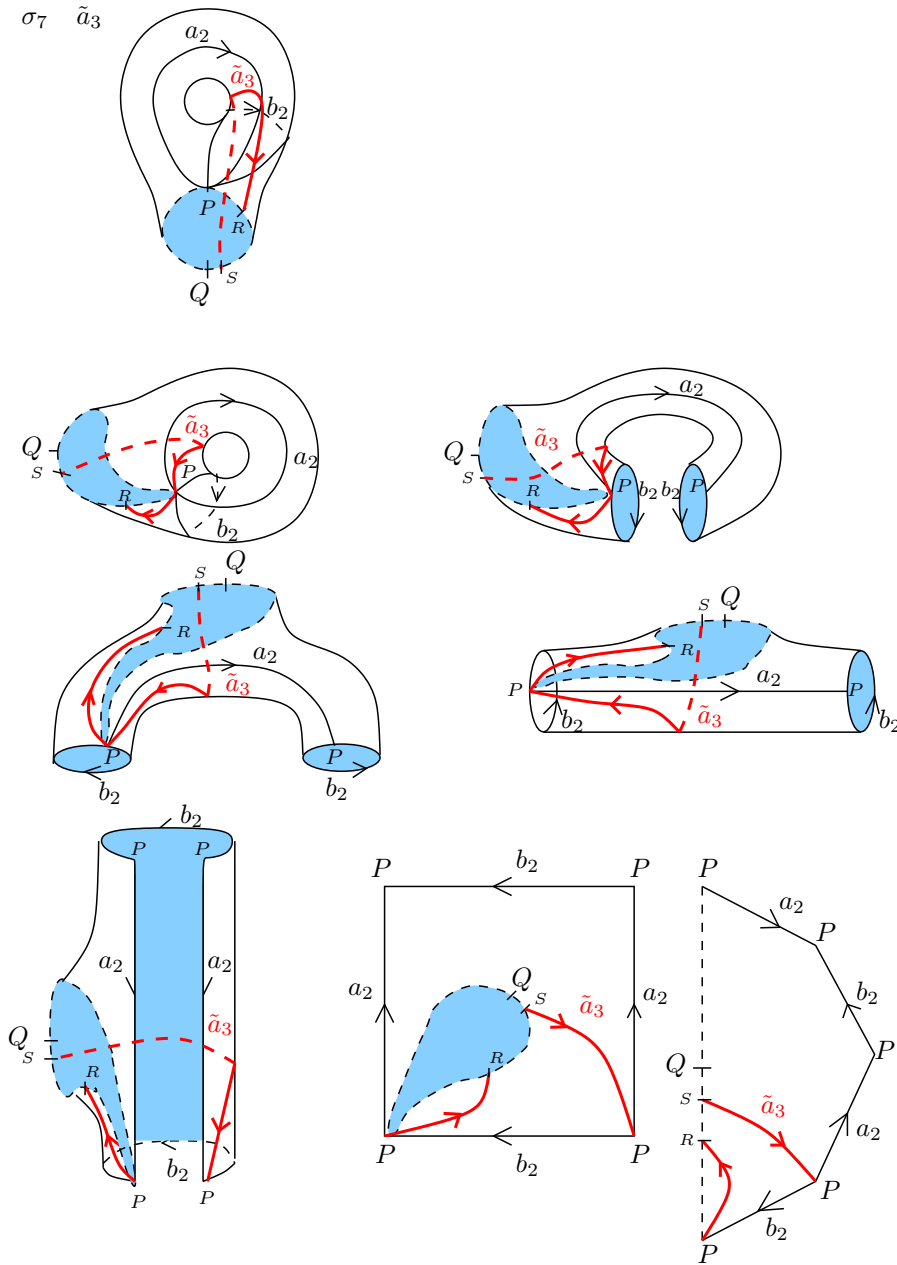


Figura 2.58: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

$\sigma_7 \tilde{a}_3$

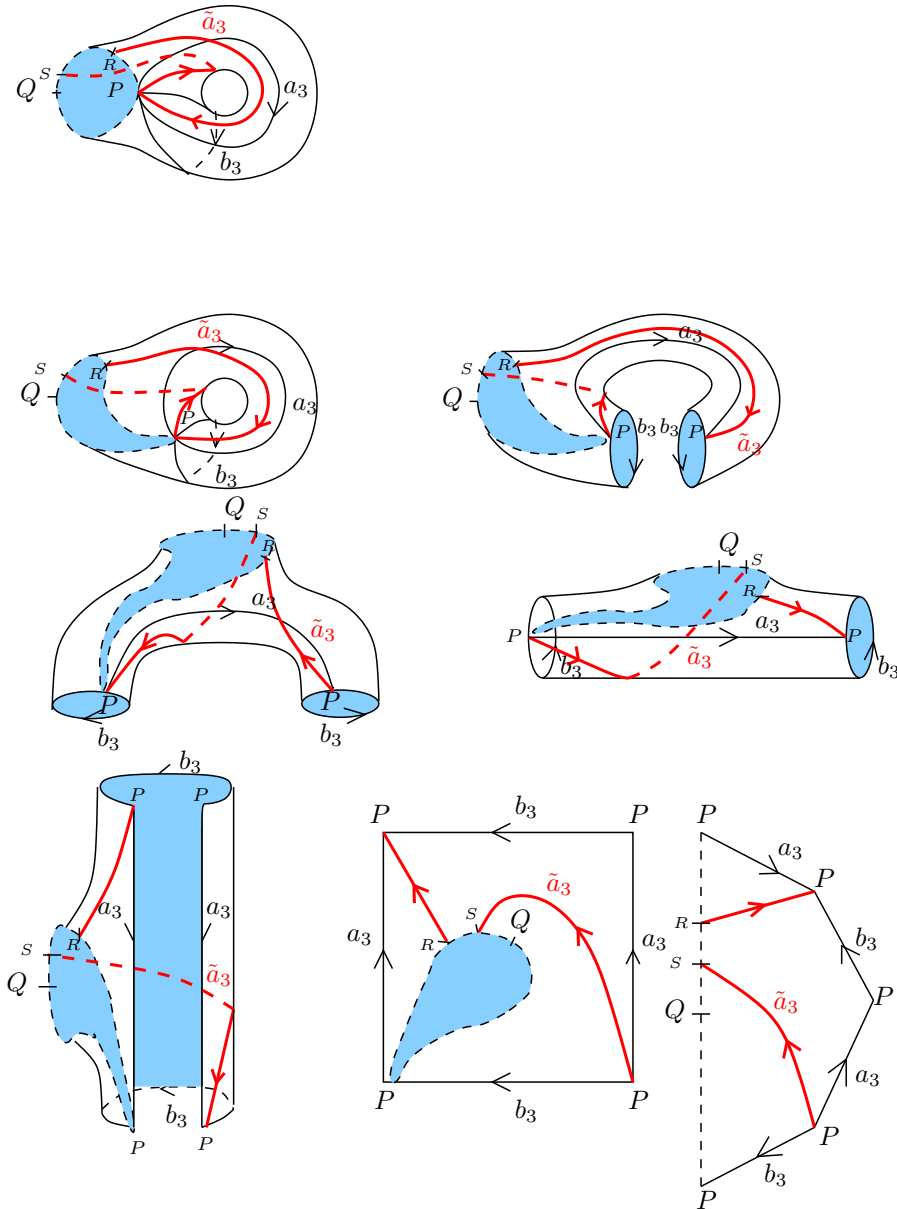
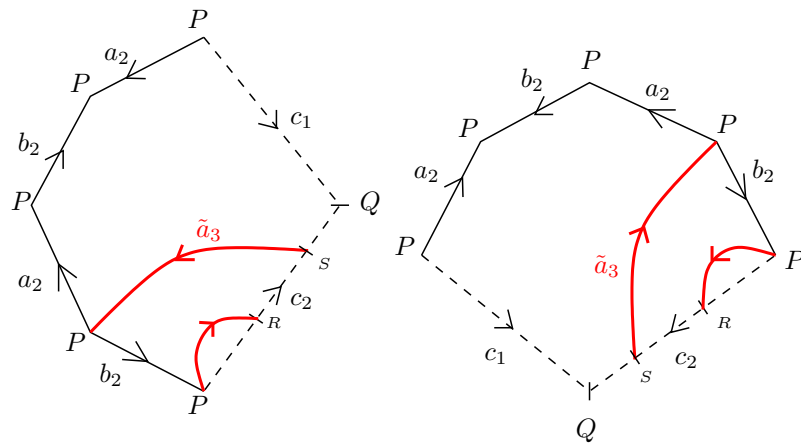


Figura 2.59: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 2



ASA 3

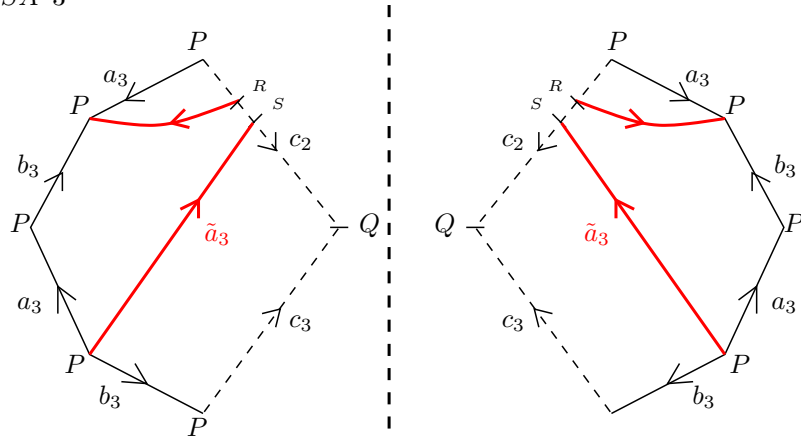
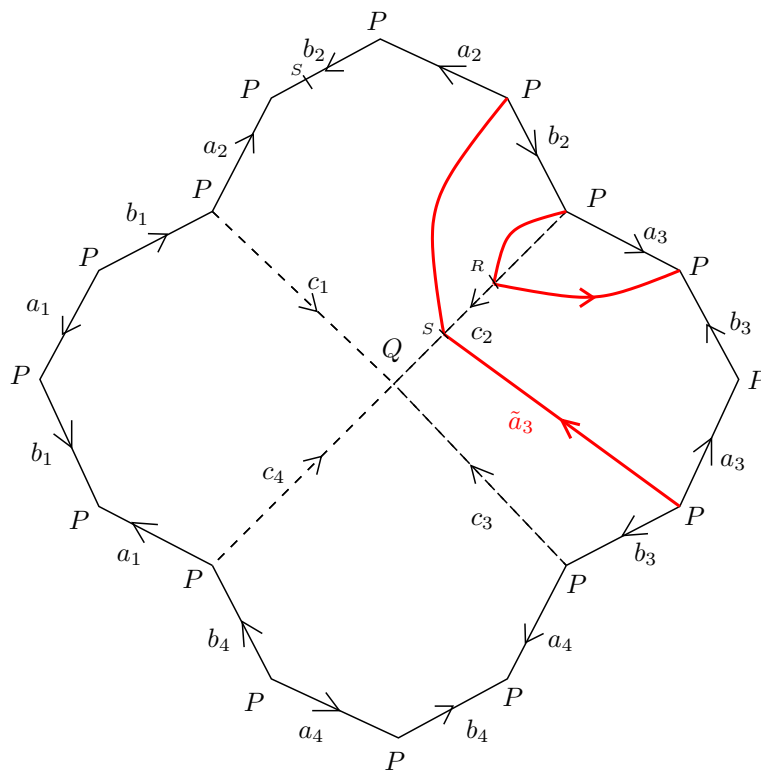


Figura 2.60: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 2, aplicamos simetría especular y luego rotación de 90° en sentido anti-horario, y en la Asa 3 aplicamos simetría especular.

$\sigma_7 \tilde{a}_3$



$$\tilde{a}_3 = a_3 b_3 a_3^{-1} b_3^{-1} a_3$$

Figura 2.61: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

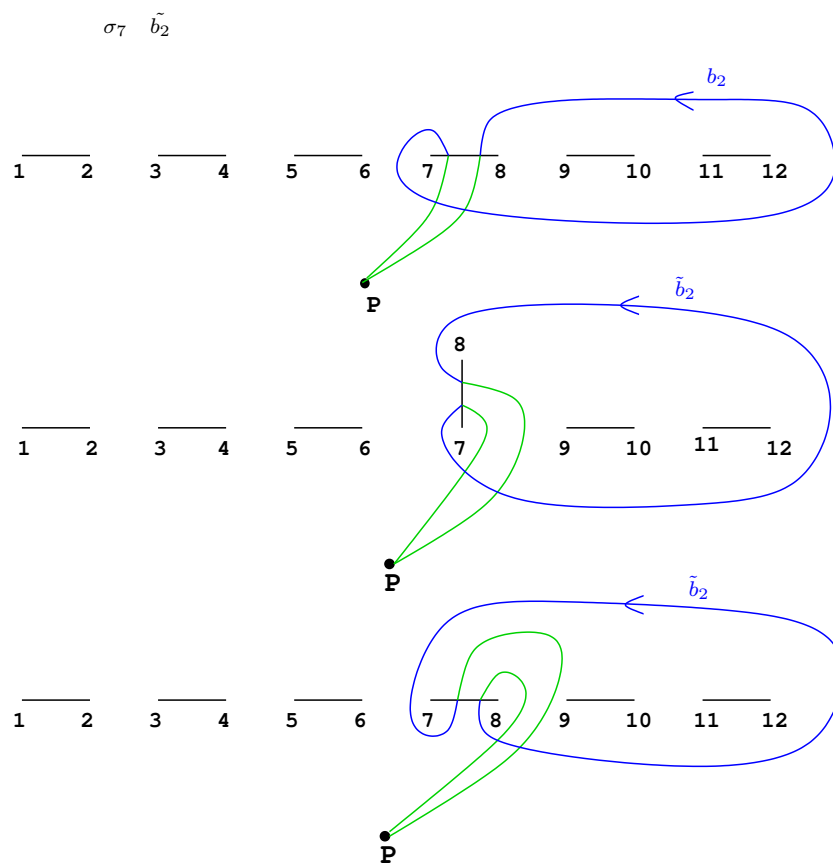


Figura 2.62: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

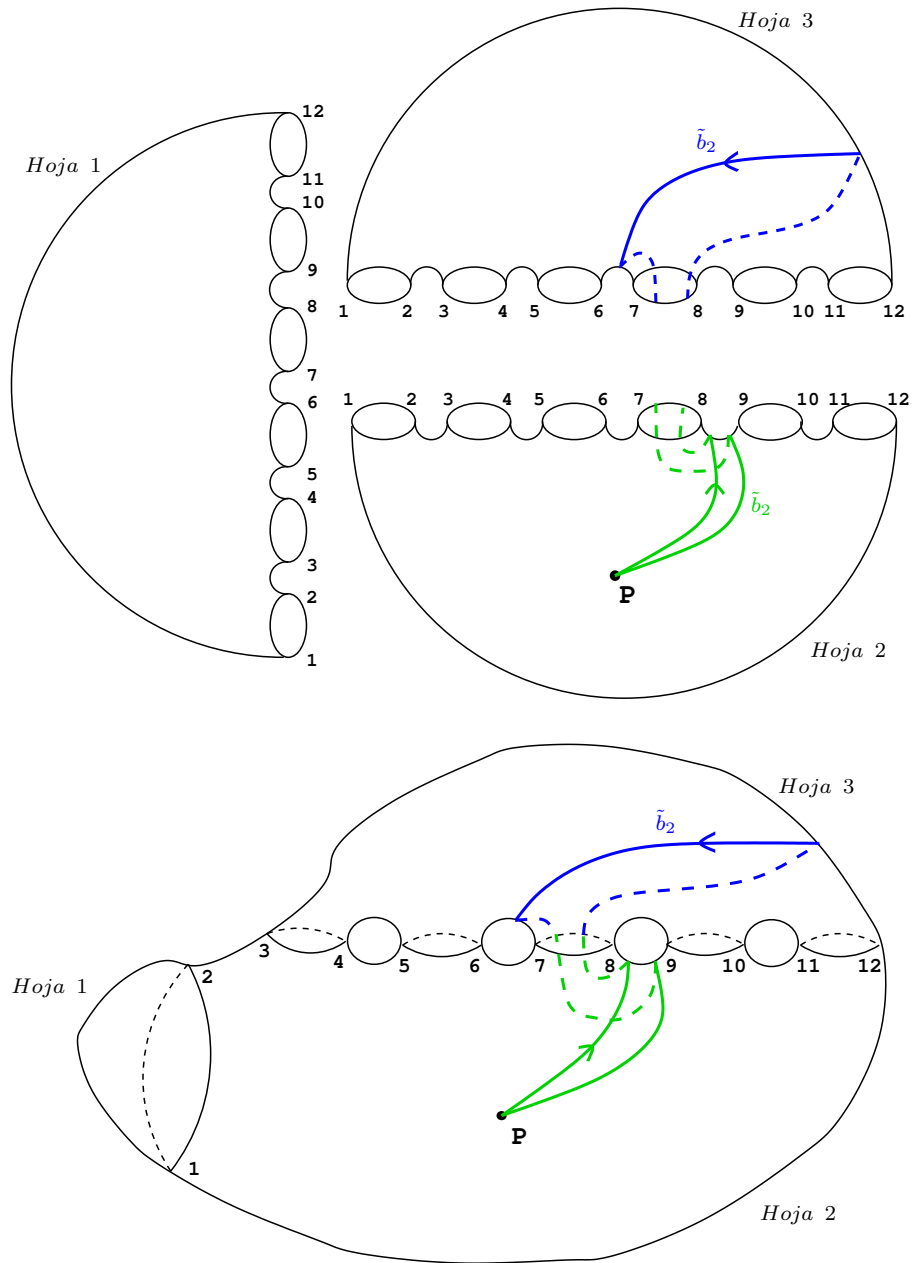


Figura 2.64: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

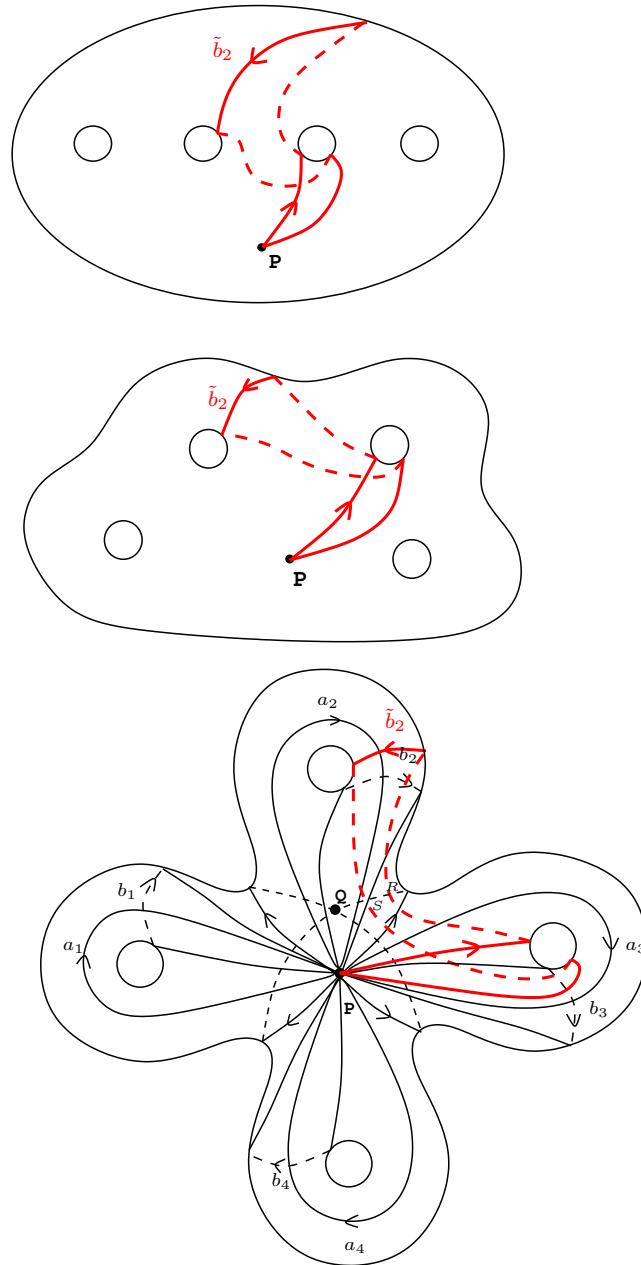


Figura 2.65: Camino imagen \tilde{c} en la esfera con 4 asas.

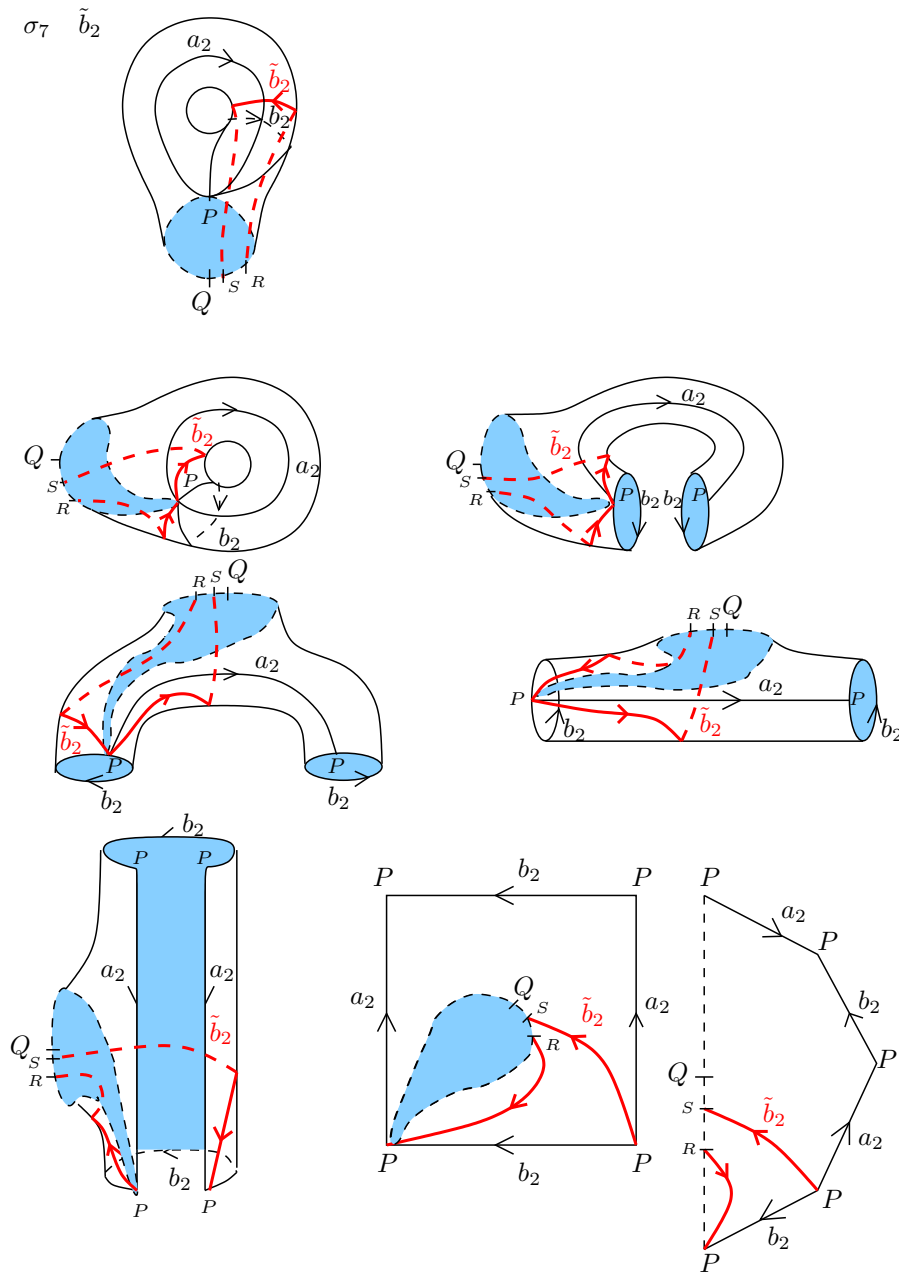


Figura 2.66: Deformación de la asa 2 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

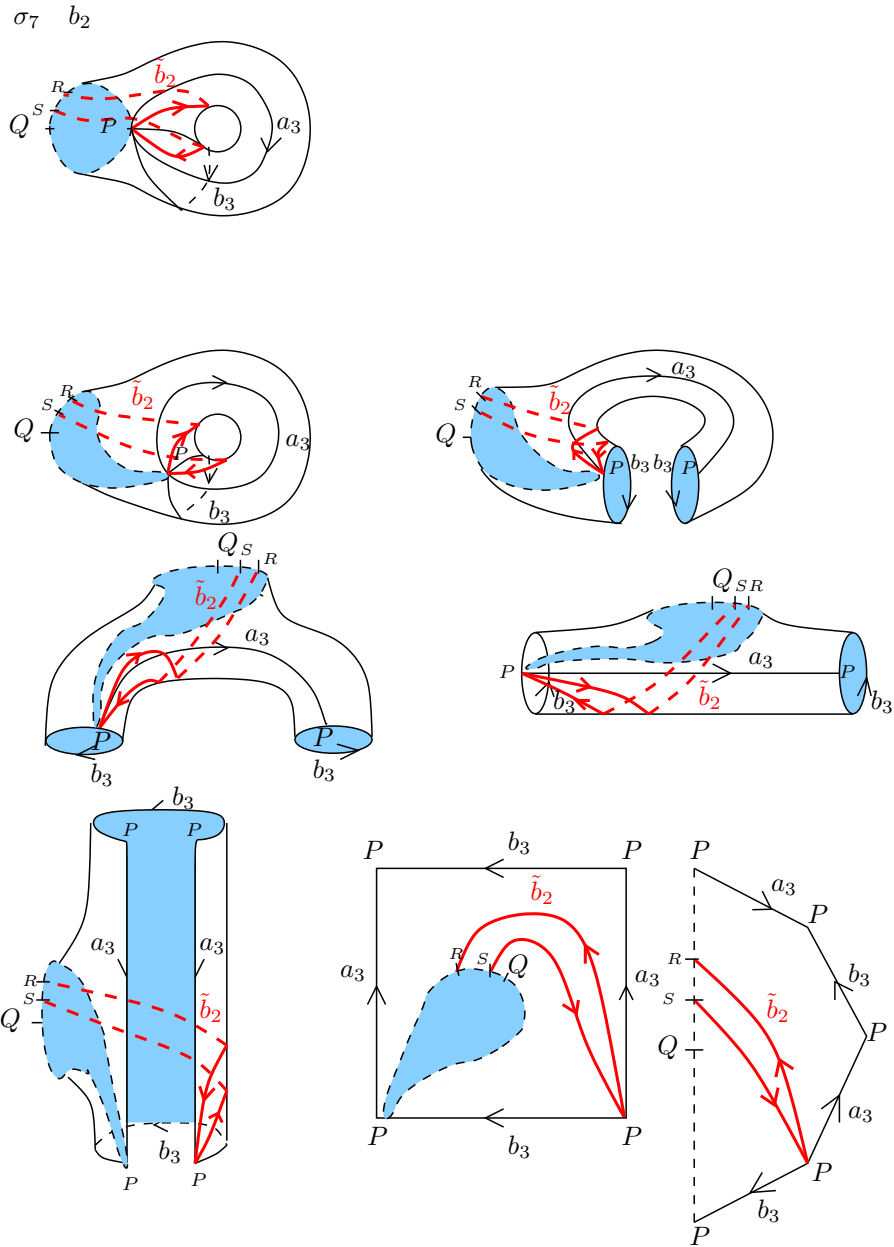
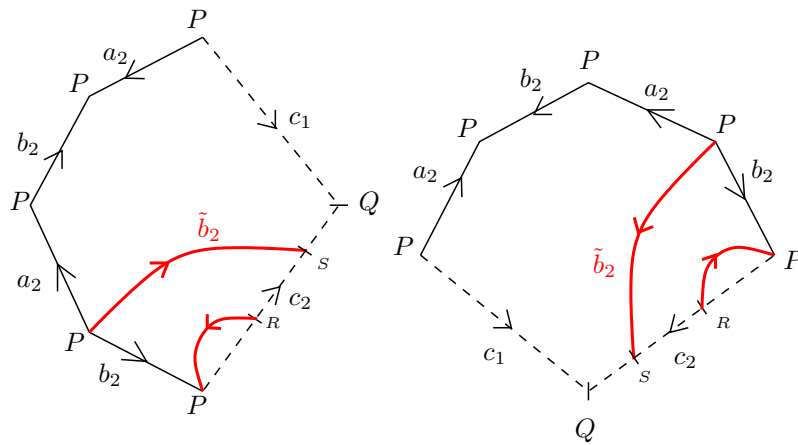


Figura 2.67: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 2



ASA 3

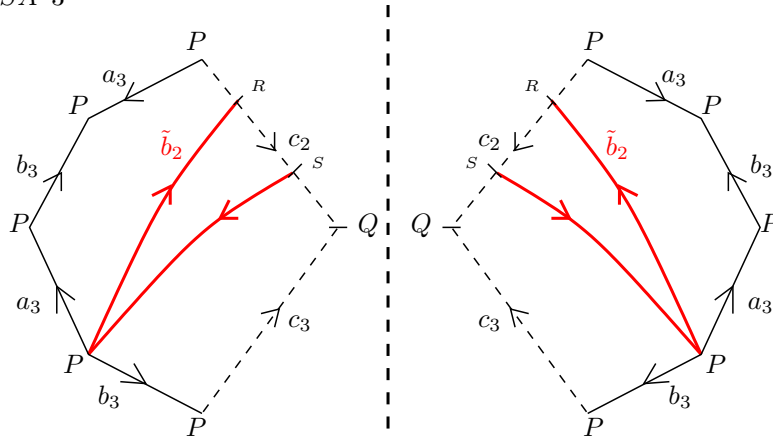
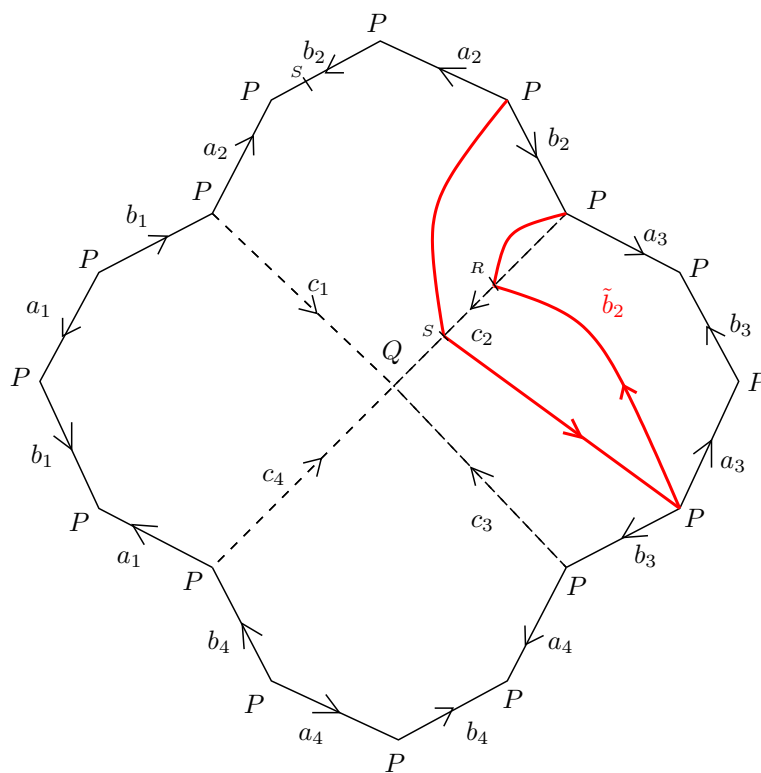


Figura 2.68: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 2, aplicamos simetría especular y luego rotación de 90° en sentido anti-horario, y en la Asa 3 aplicamos simetría especular.

$\sigma_7 \tilde{b}_2$



$$\tilde{b}_2 = a_3 b_3 a_3^{-1} b_2 a_3 b_3^{-1} a_3^{-1}$$

Figura 2.69: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

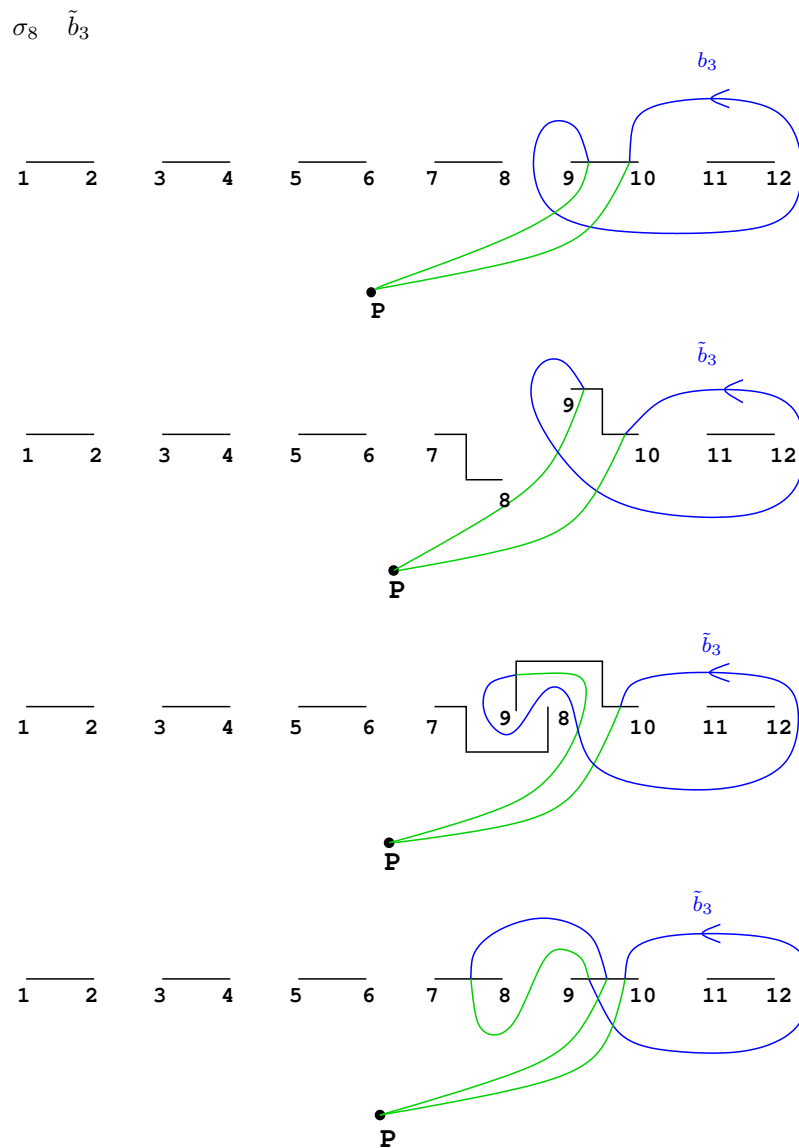


Figura 2.70: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

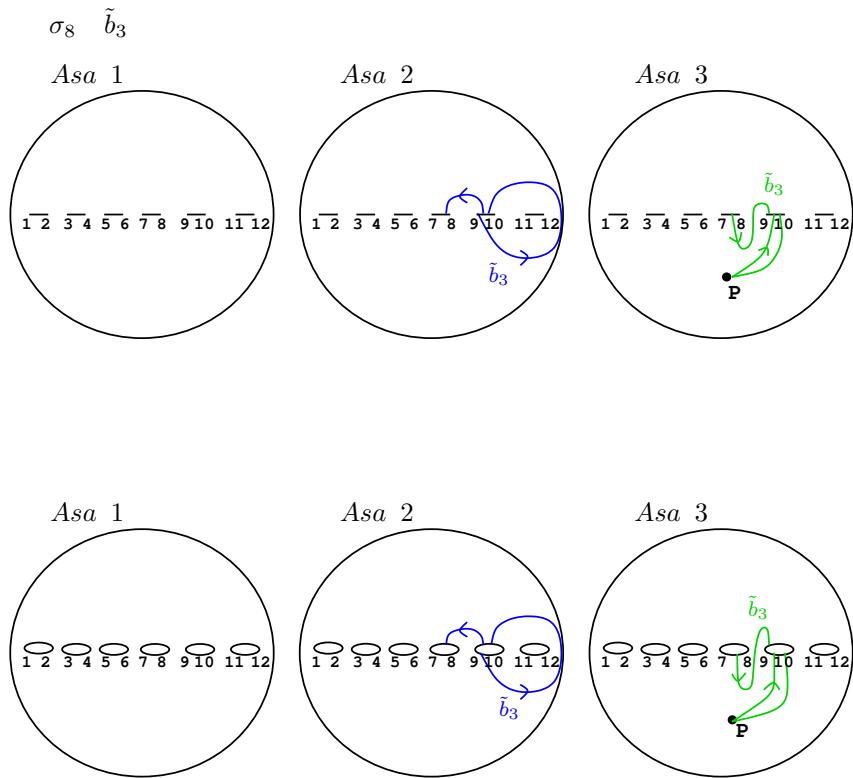


Figura 2.71: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

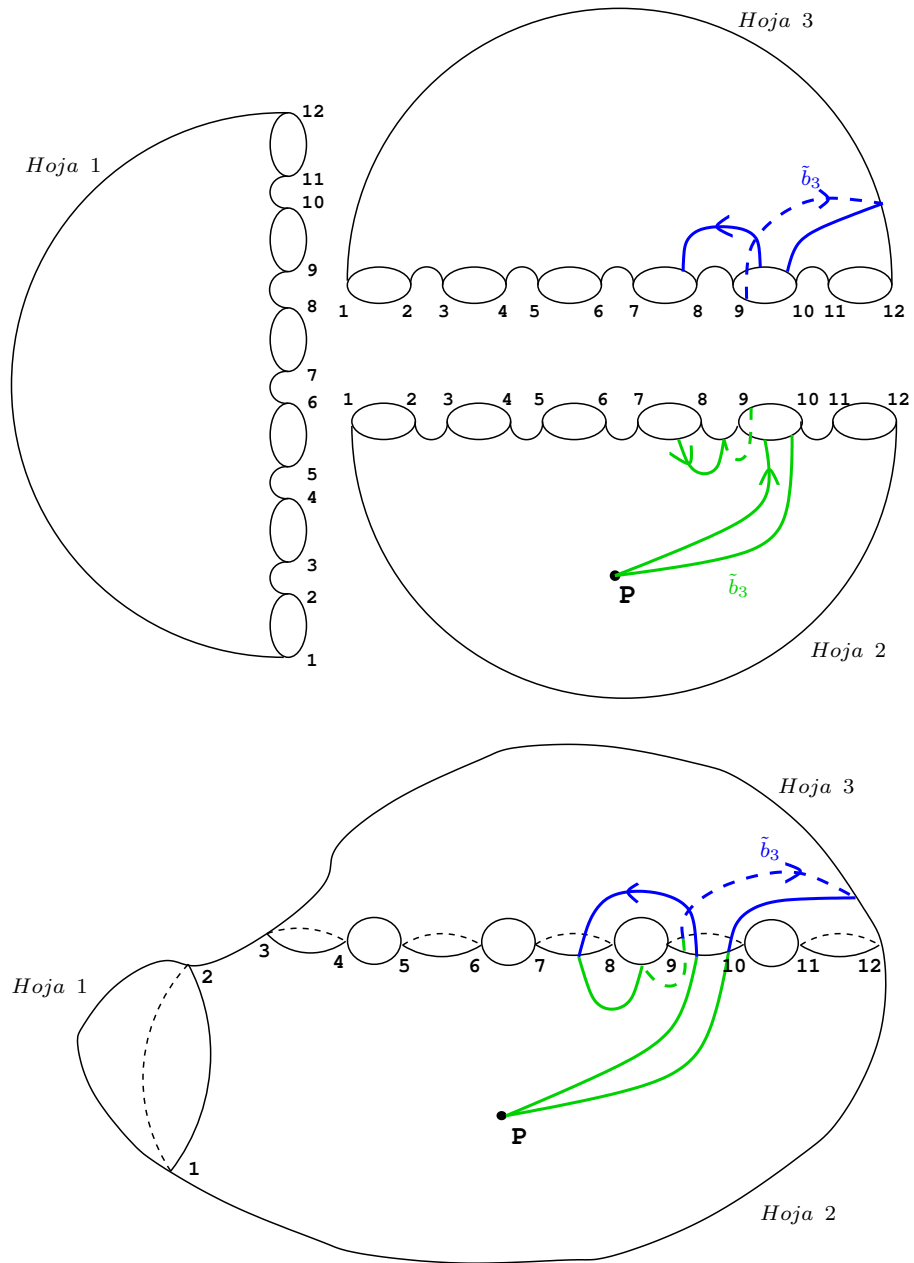


Figura 2.72: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

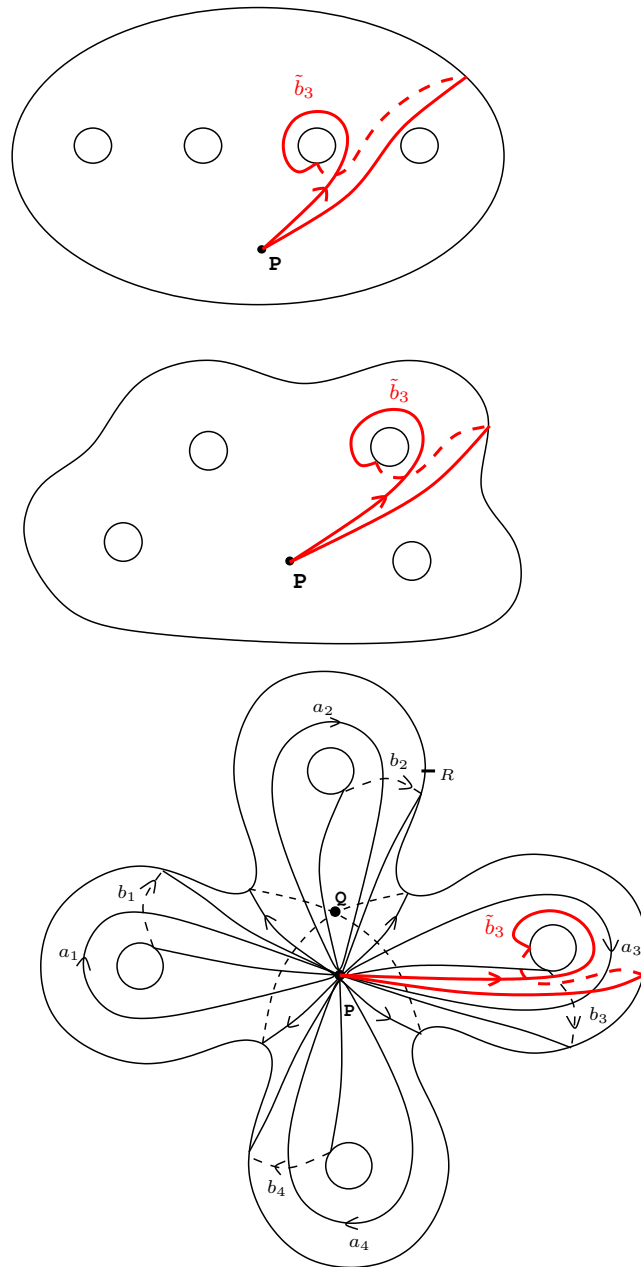


Figura 2.73: Camino imagen \tilde{c} en la esfera con 4 asas.

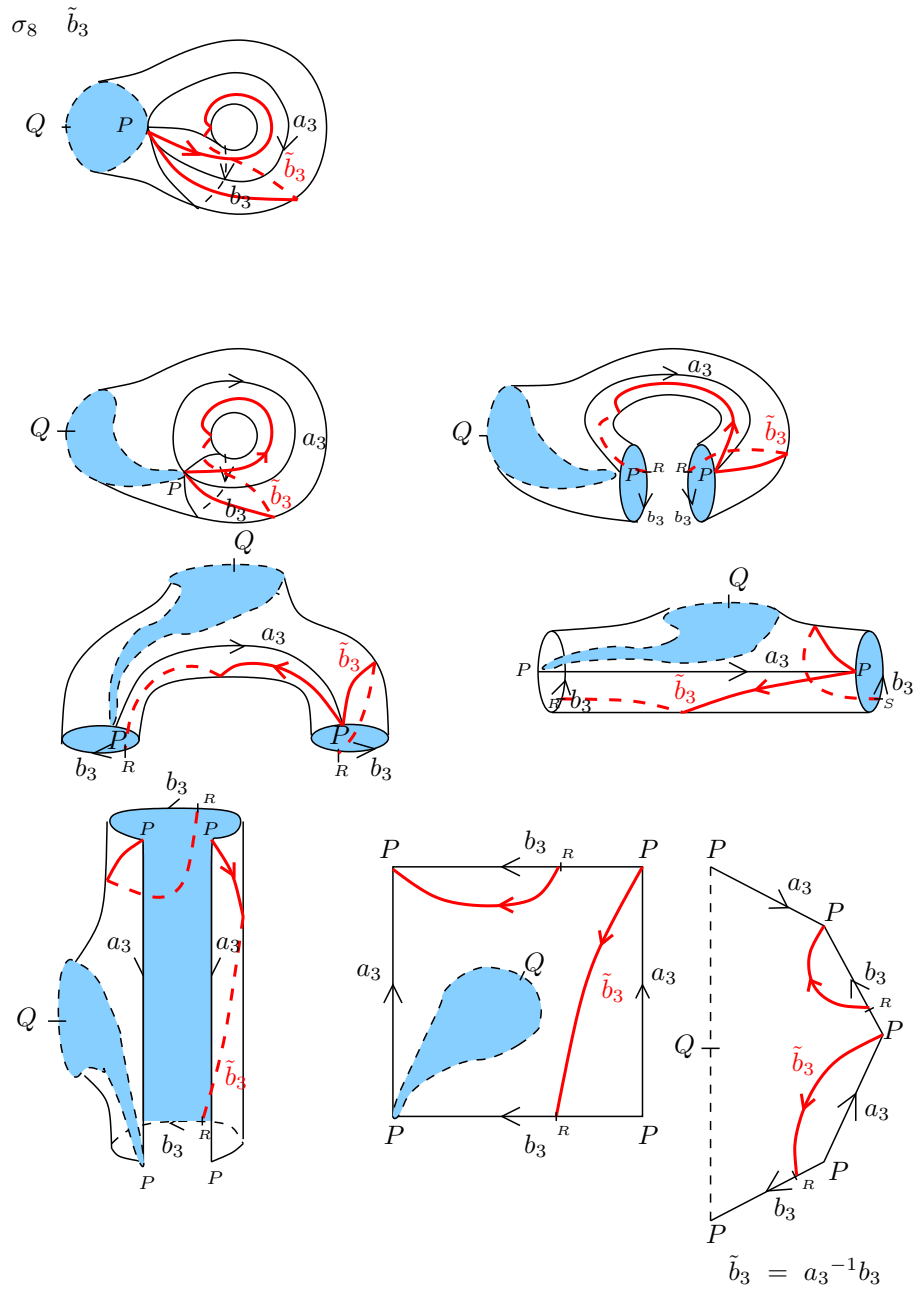


Figura 2.74: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

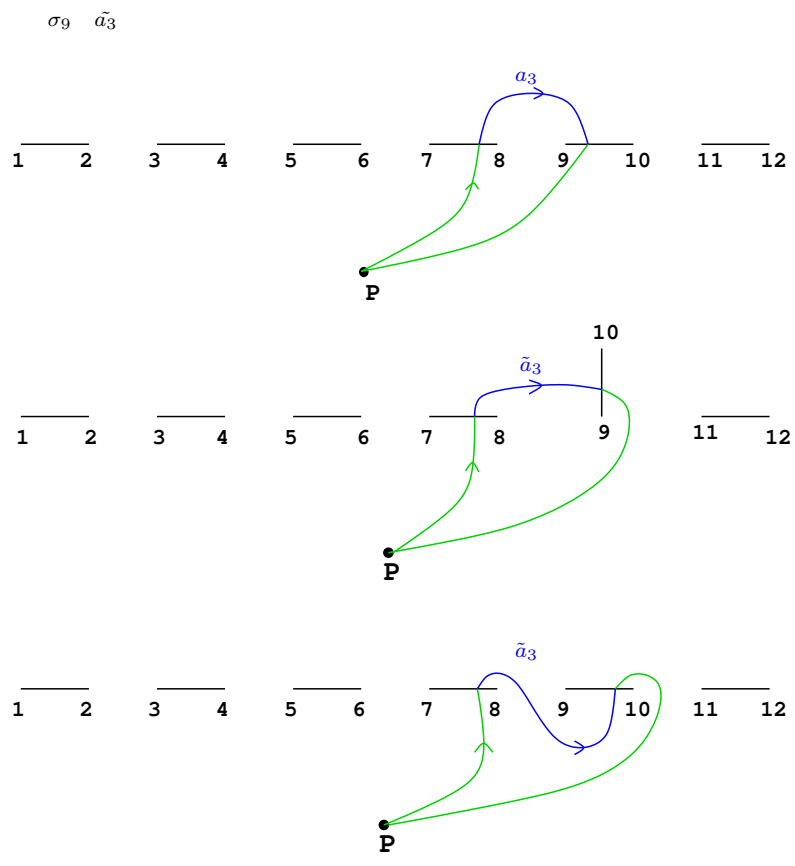


Figura 2.75: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

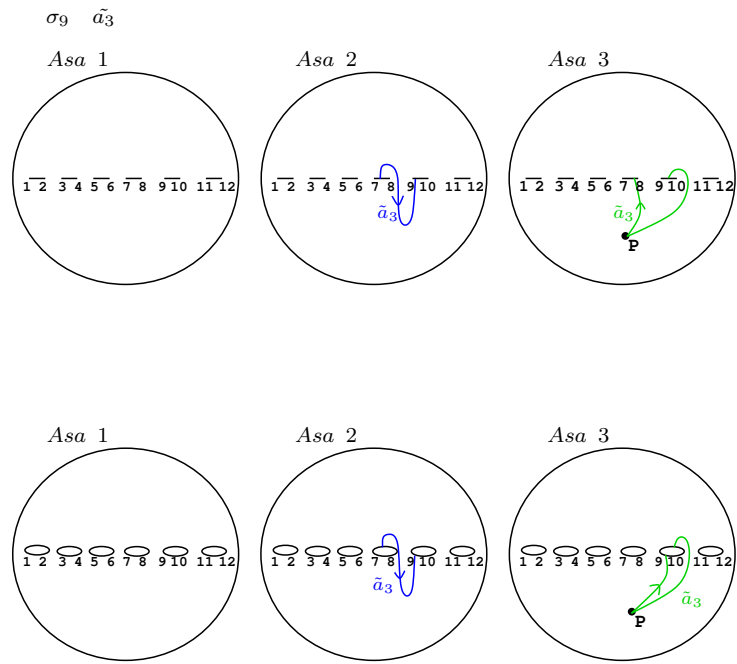


Figura 2.76: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

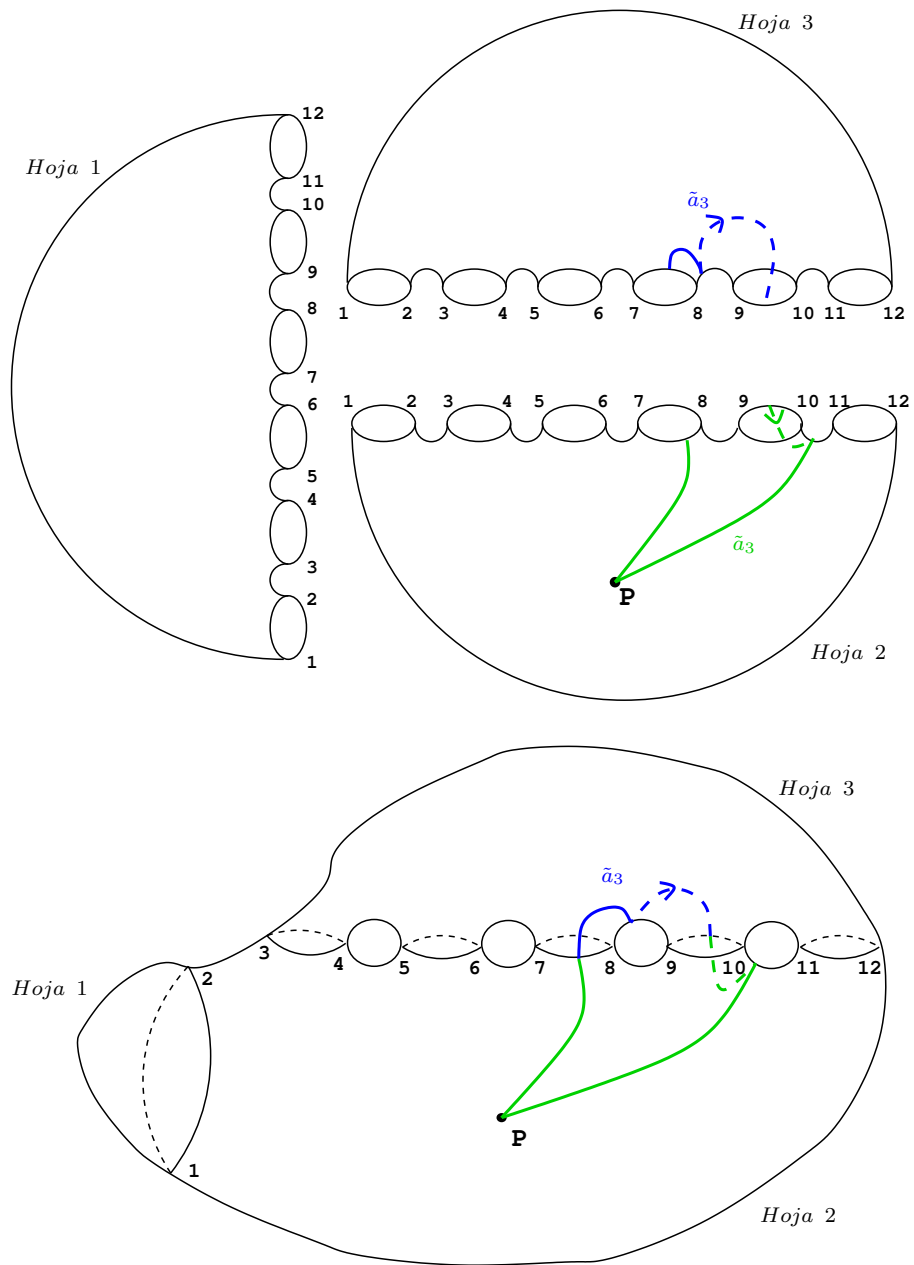


Figura 2.77: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

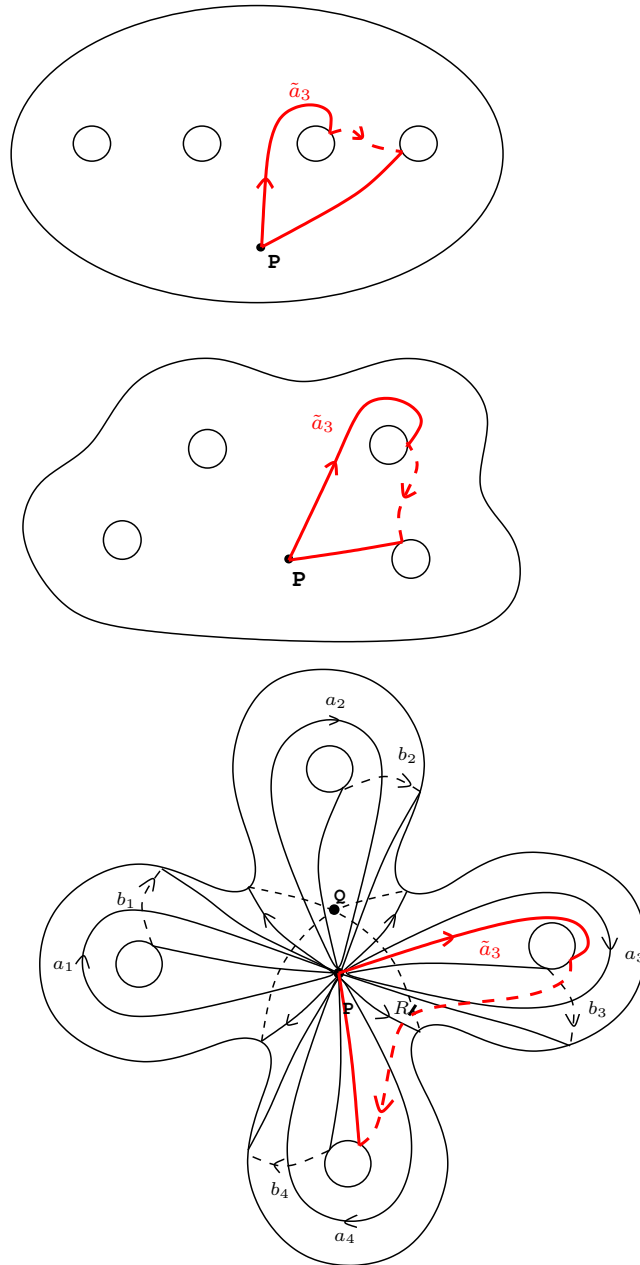


Figura 2.78: Camino imagen \tilde{c} en la esfera con 4 asas.

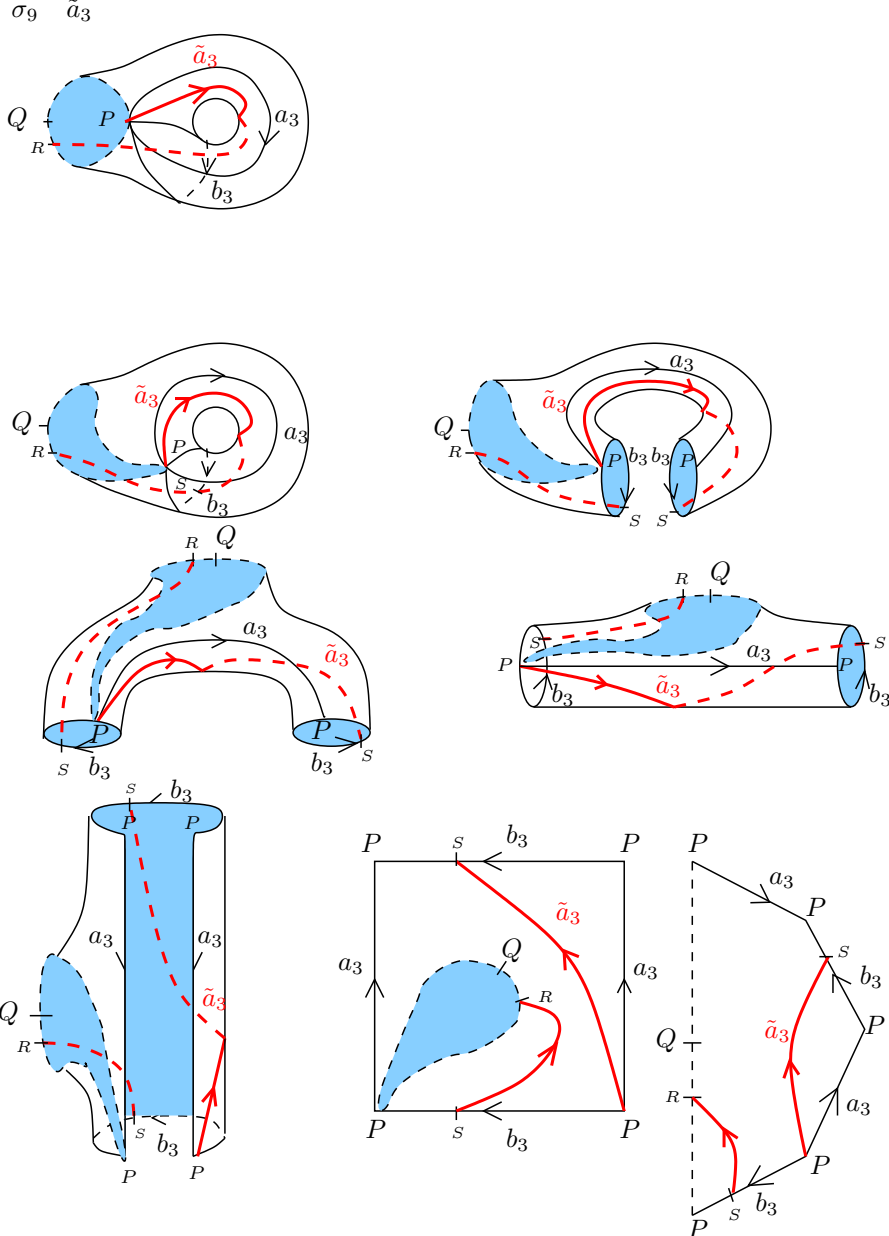


Figura 2.79: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

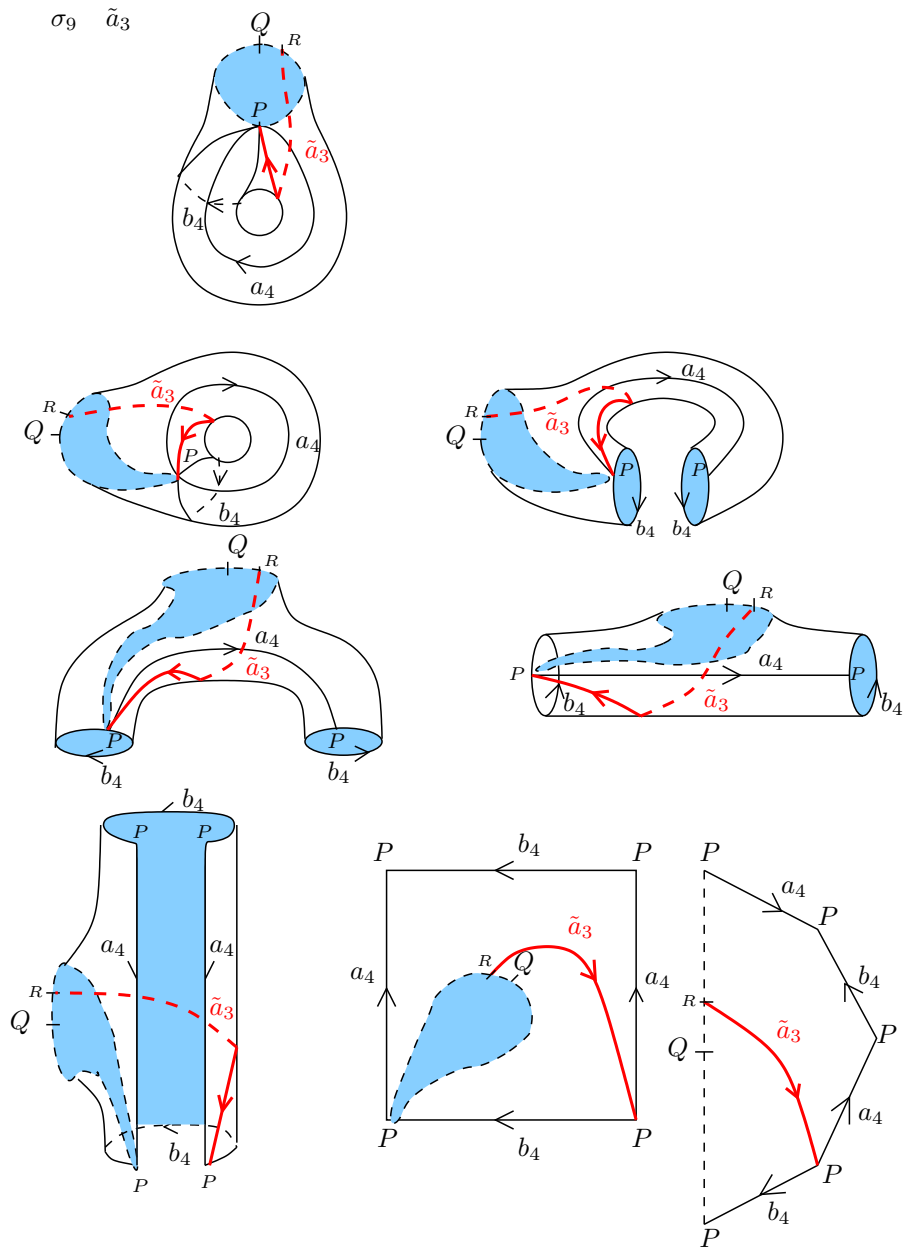
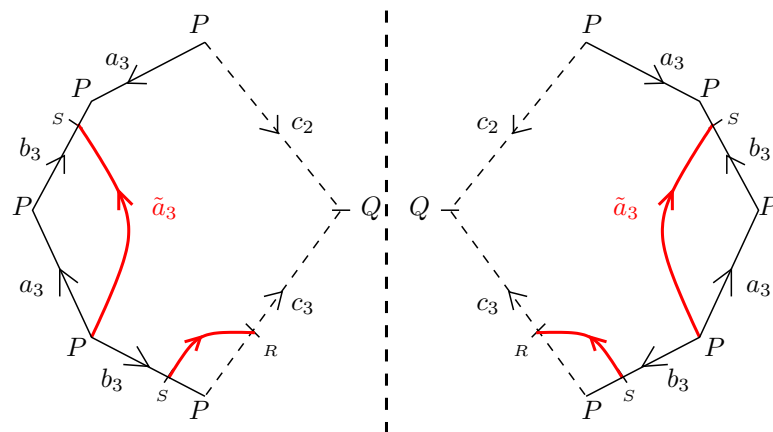


Figura 2.80: Deformación de la asa 4 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 3



ASA 4

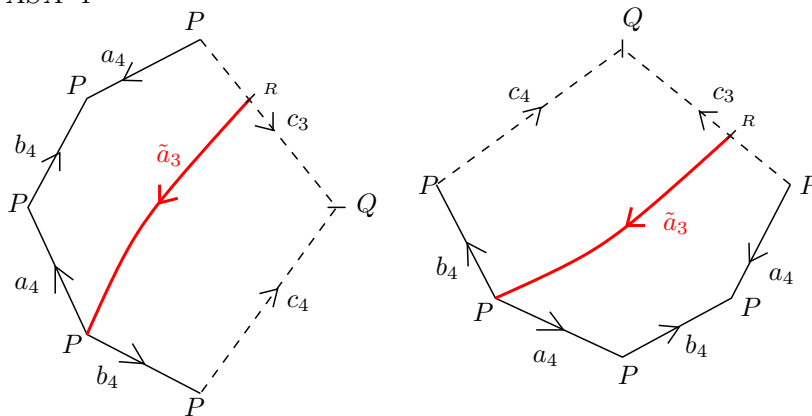
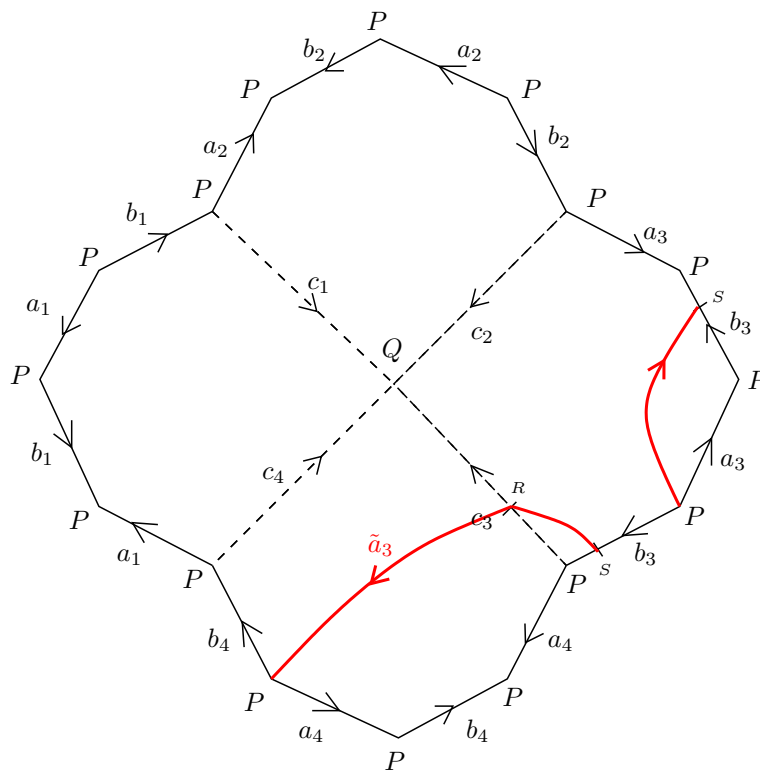


Figura 2.81: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 3, aplicamos simetría especular, y en la Asa 4 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_9 \tilde{a}_3$



$$\tilde{a}_3 = a_3 b_3 a_4 b_4^{-1} a_4^{-1}$$

Figura 2.82: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

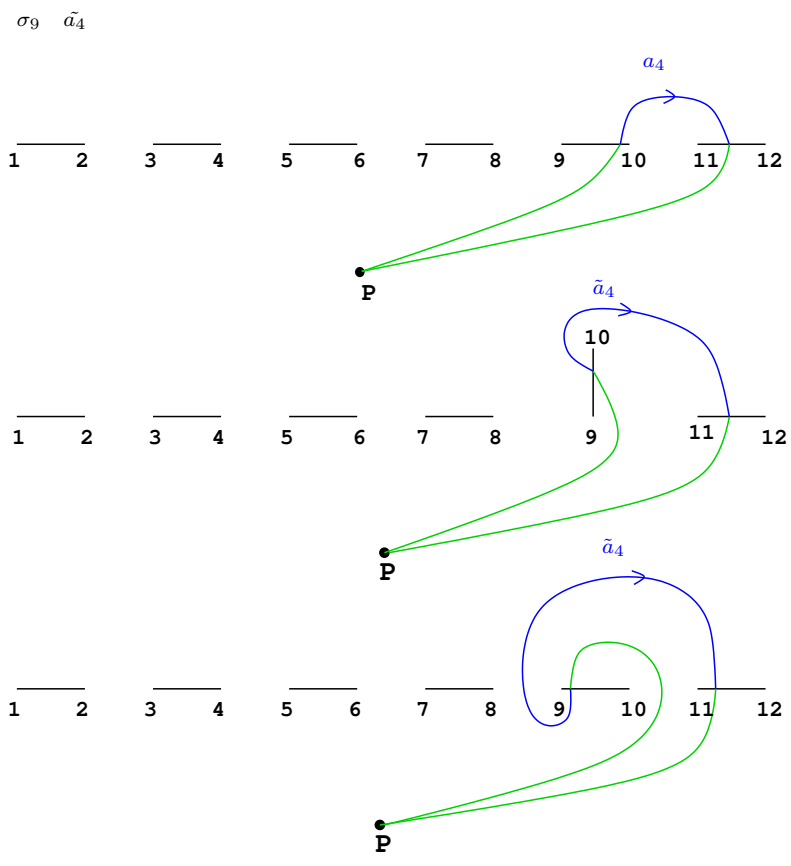


Figura 2.83: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

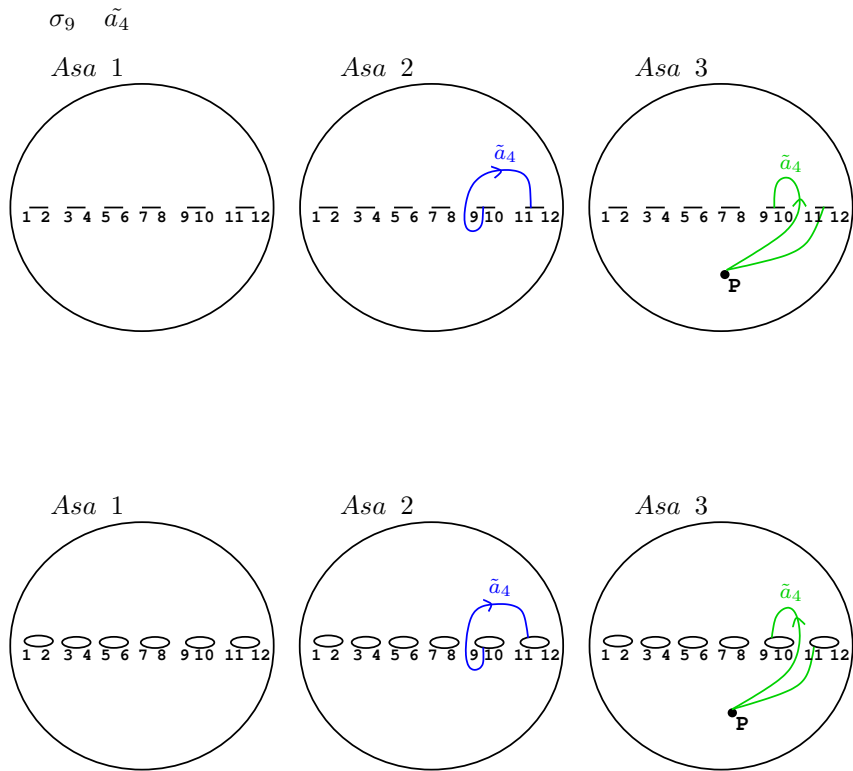


Figura 2.84: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

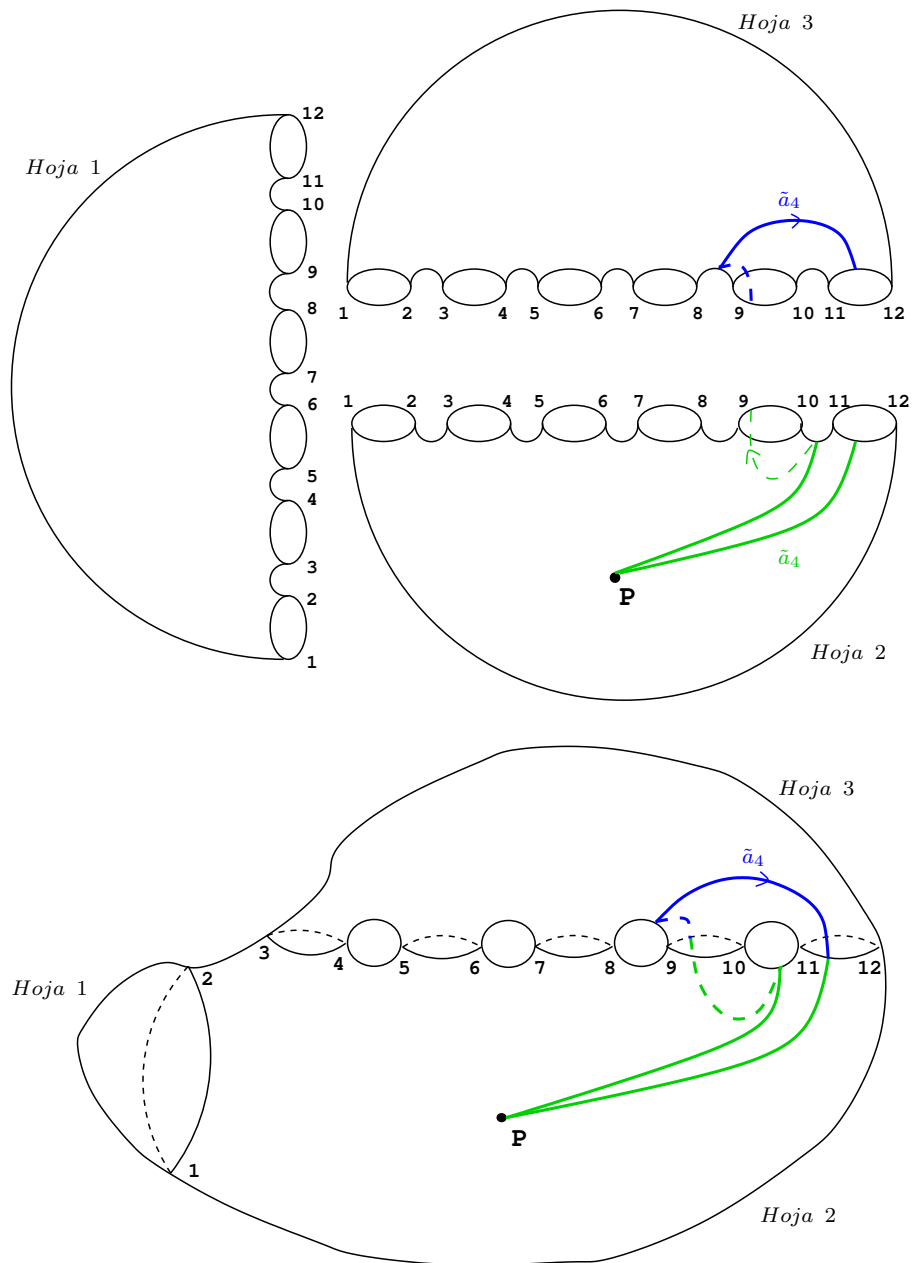


Figura 2.85: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

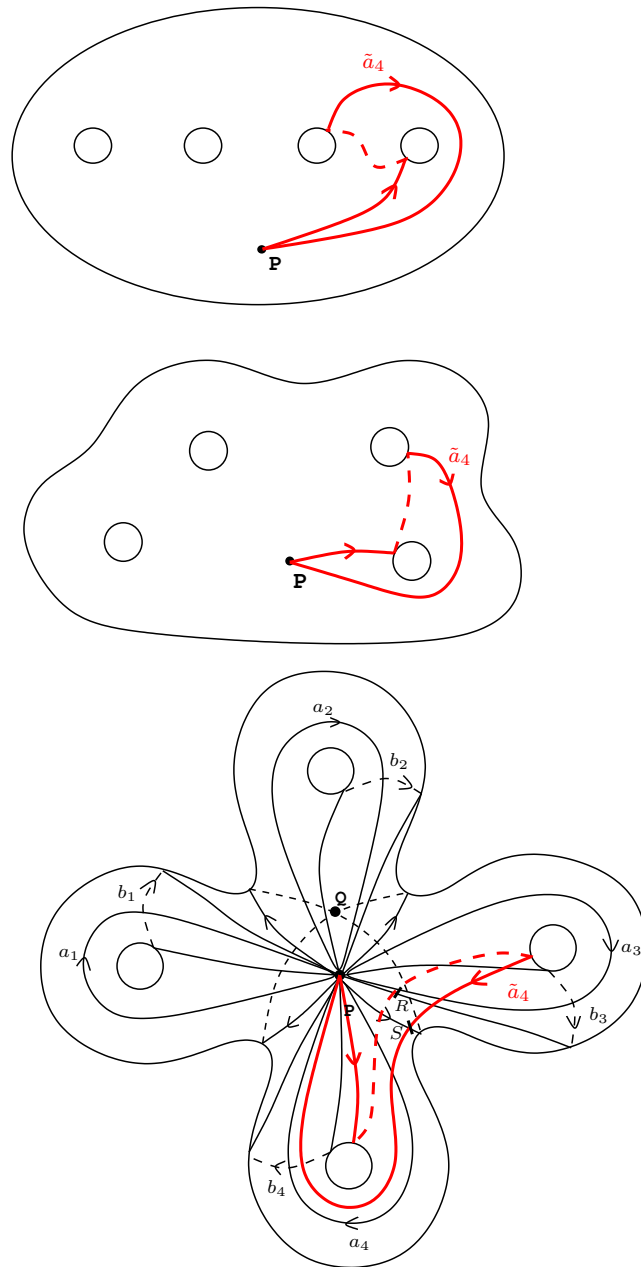


Figura 2.86: Camino imagen \tilde{c} en la esfera con 4 asas.

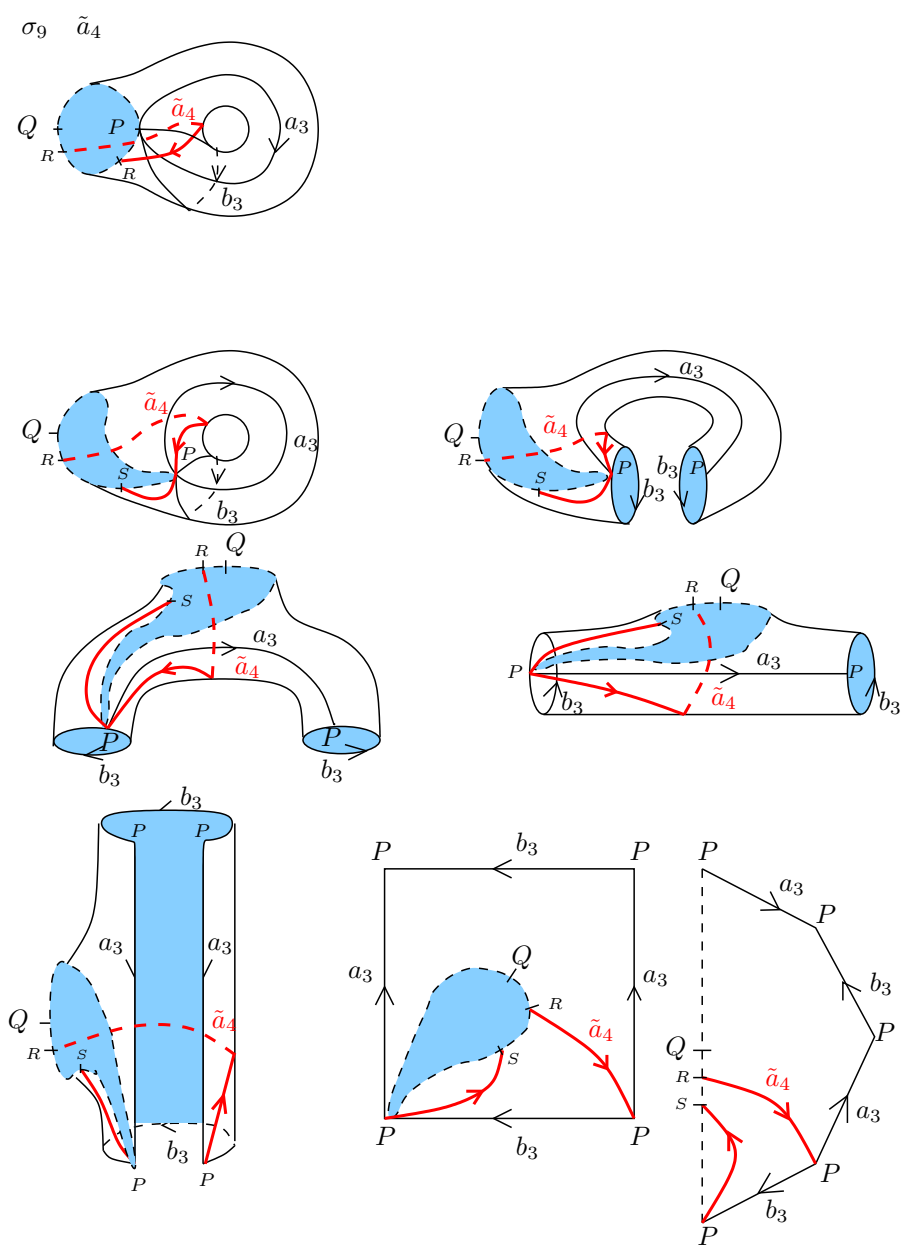


Figura 2.87: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

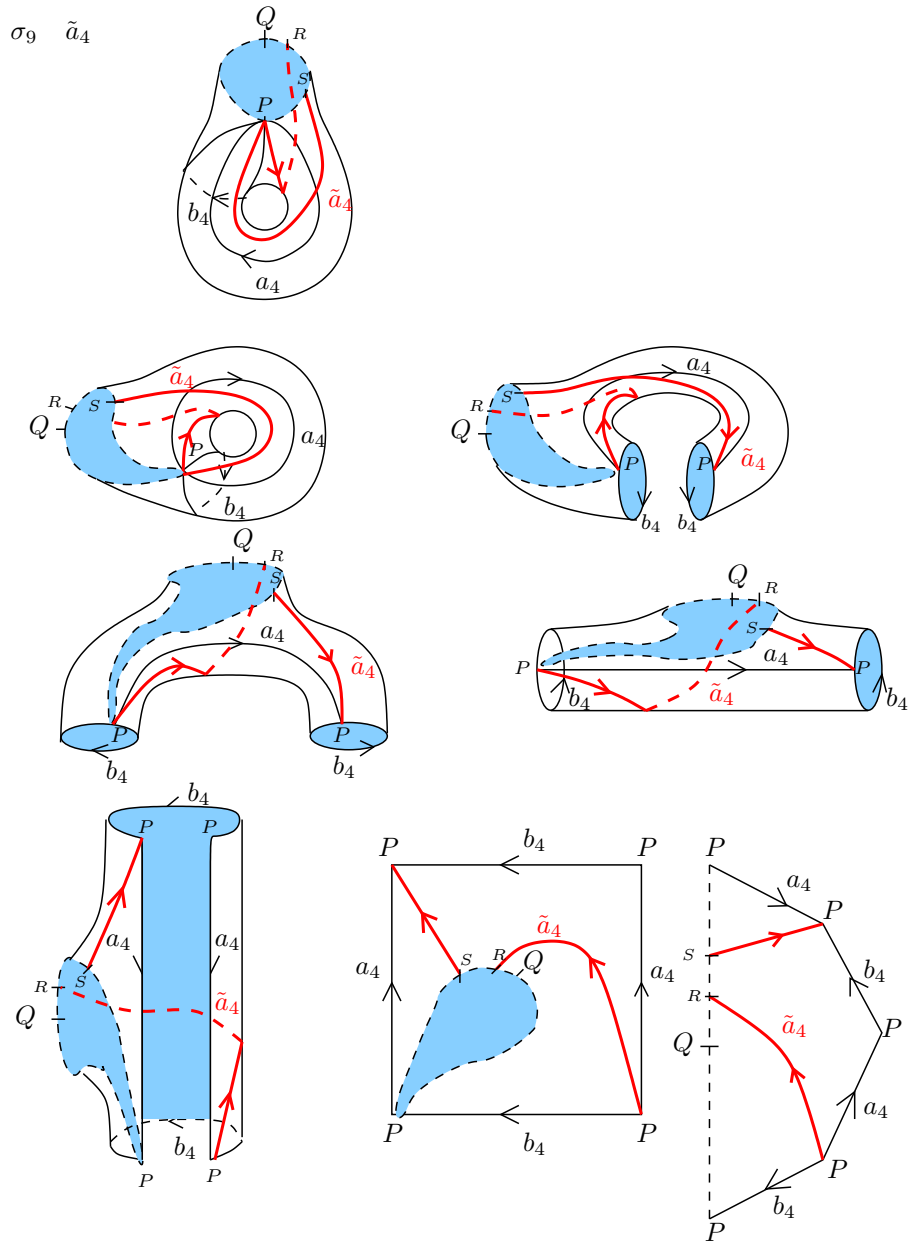
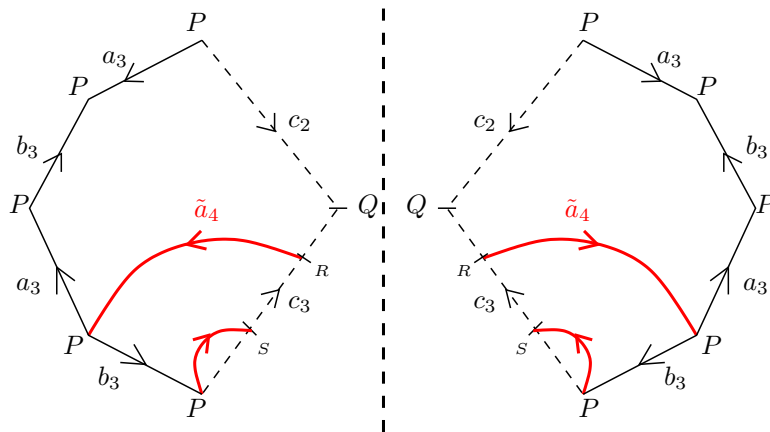


Figura 2.88: Deformación de la asa 4 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

ASA 3



ASA 4

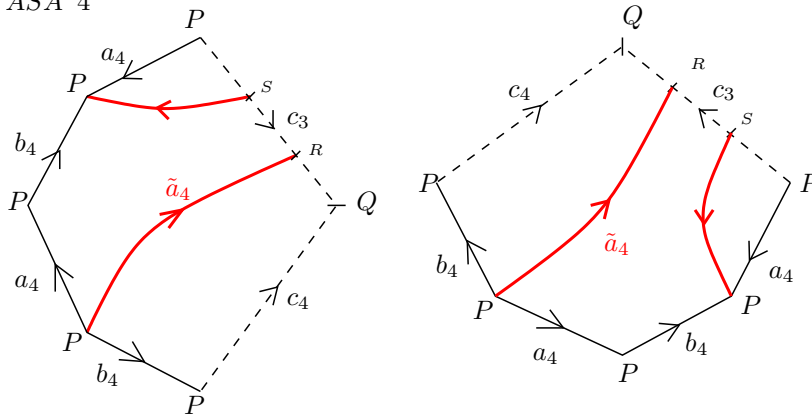
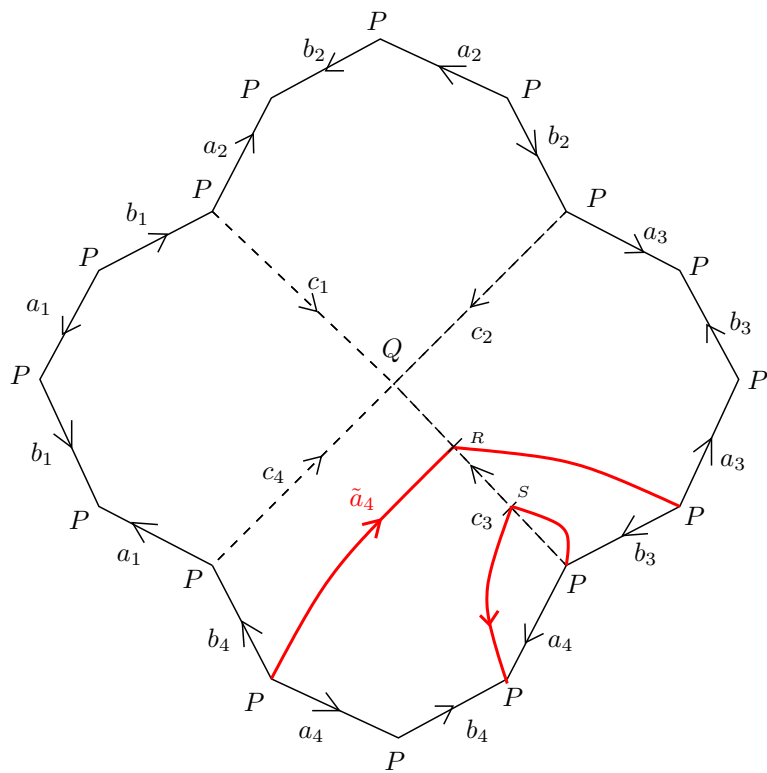


Figura 2.89: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 3, aplicamos simetría especular, y en la Asa 4 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_9 \tilde{a}_4$



$$\tilde{a}_4 = a_4 b_4 a_4^{-1} b_3^{-1} a_4$$

Figura 2.90: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

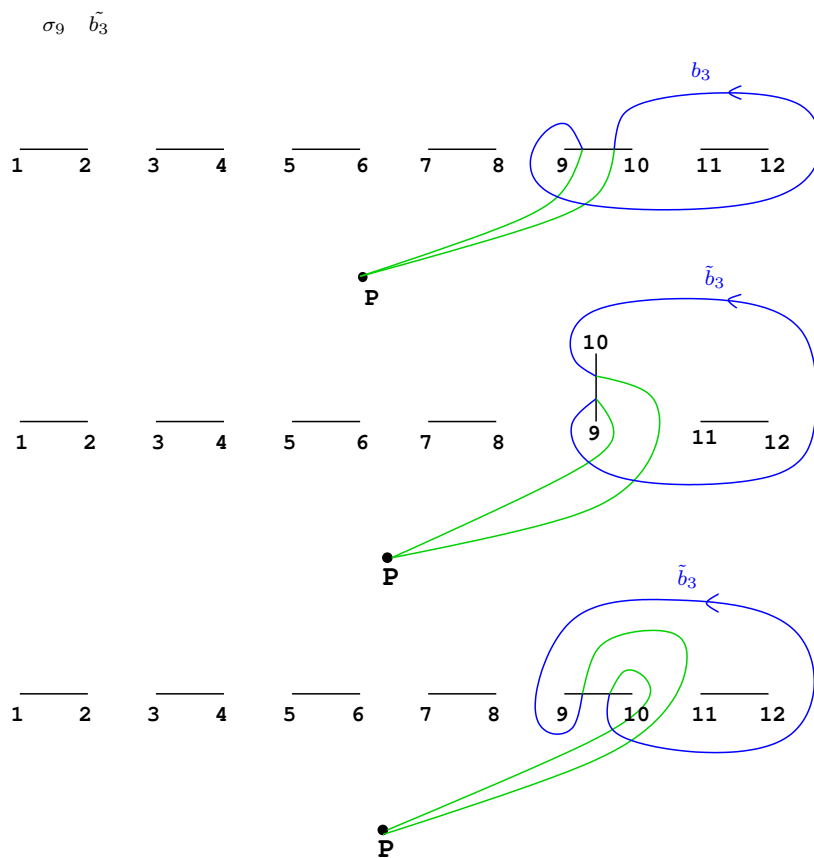


Figura 2.91: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

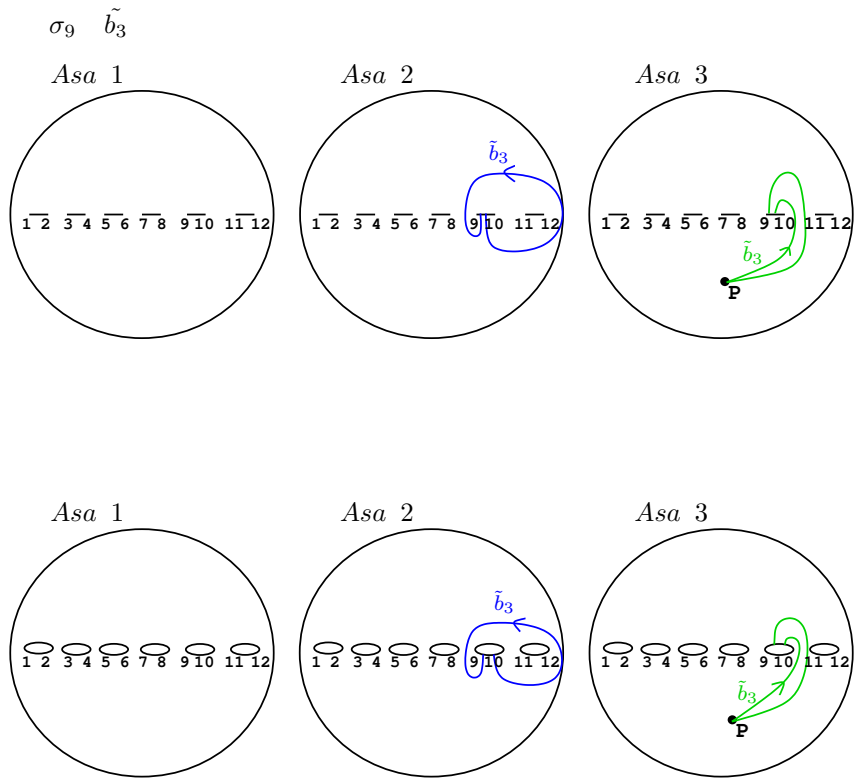


Figura 2.92: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

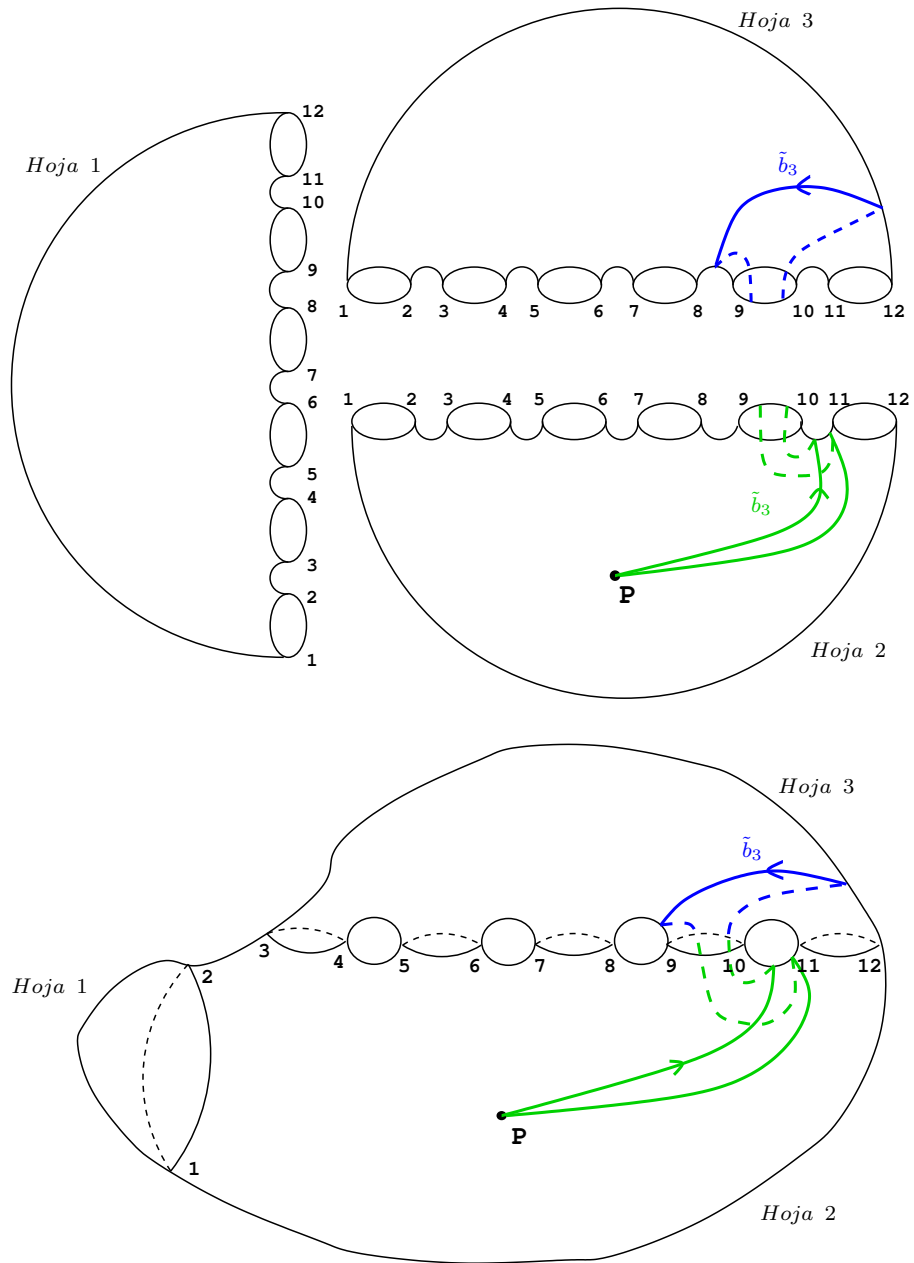


Figura 2.93: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

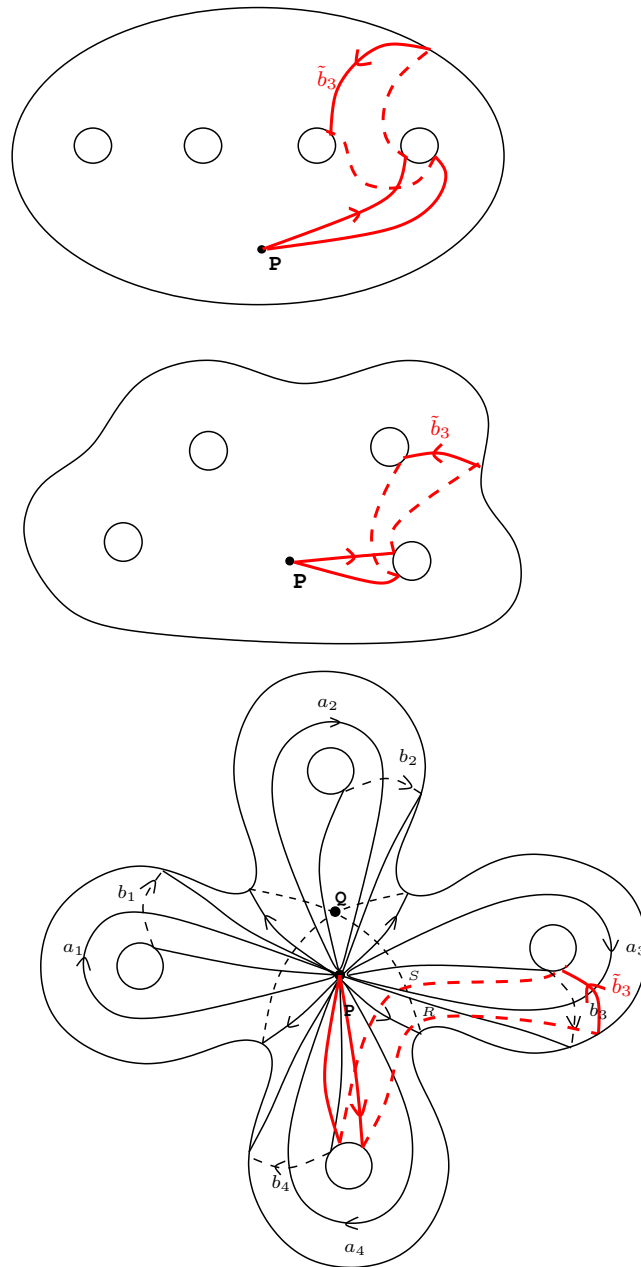


Figura 2.94: Camino imagen \tilde{c} en la esfera con 4 asas.

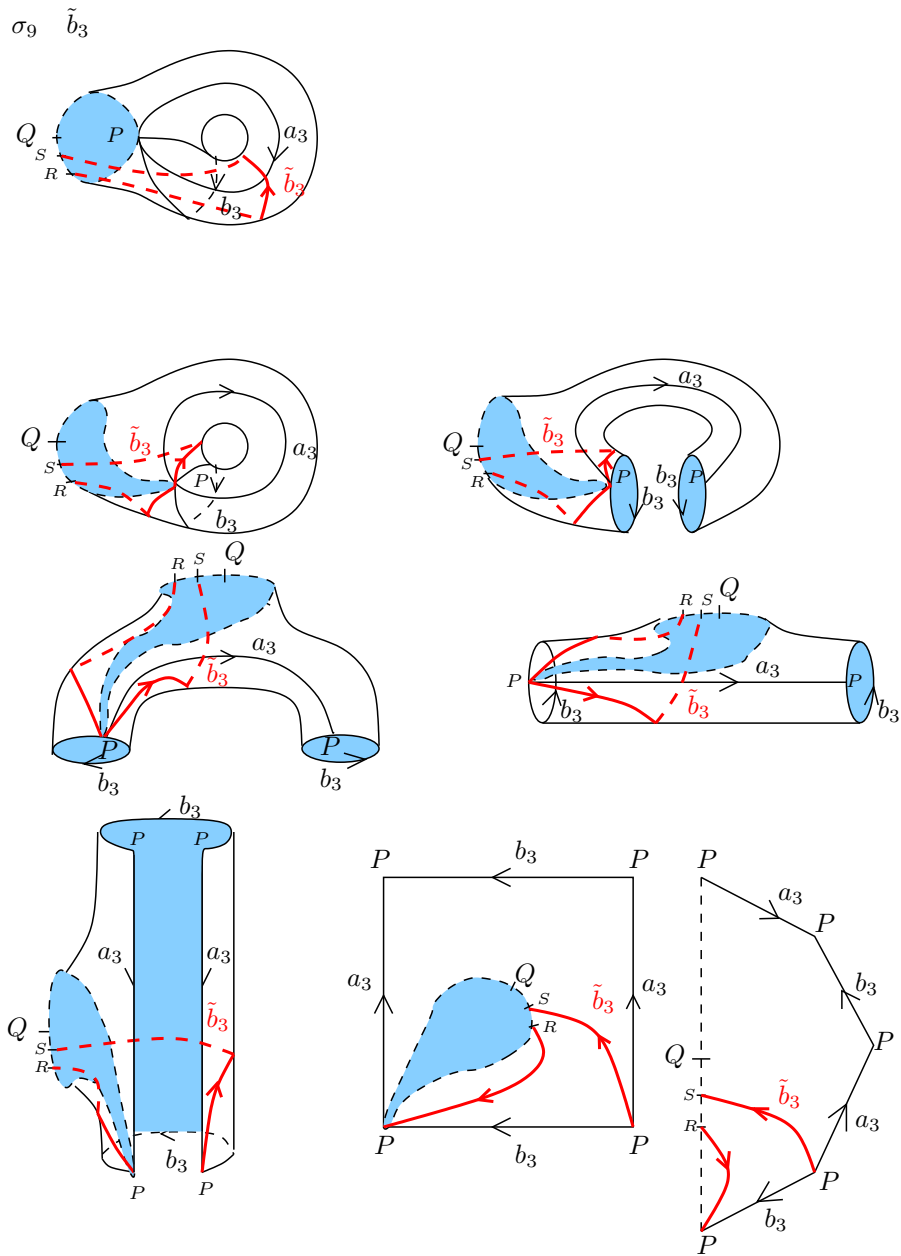


Figura 2.95: Deformación de la asa 3 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

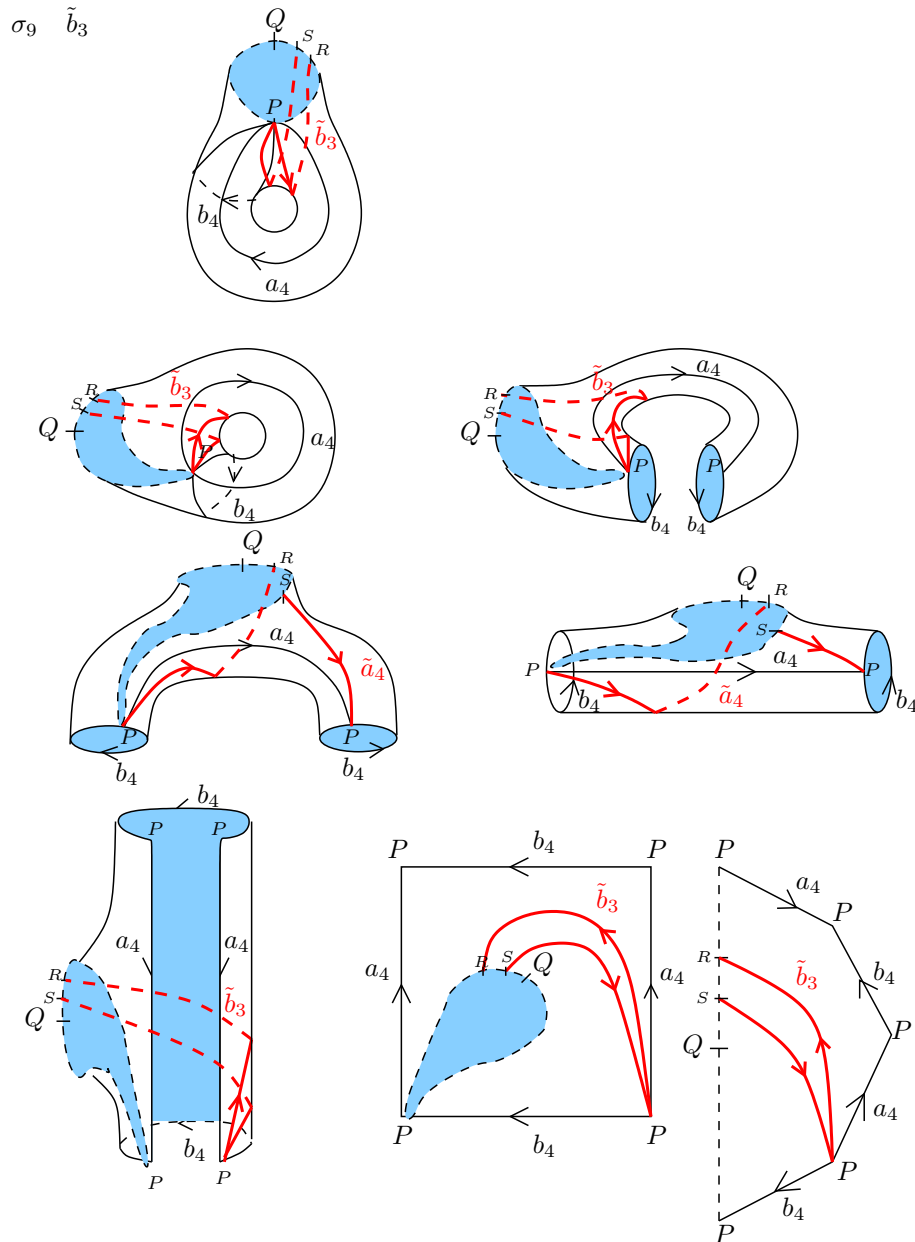


Figura 2.96: Deformación de la asa 4 de la superficie, en la que se muestra paso a paso la homotopía del tramo de \tilde{c} .

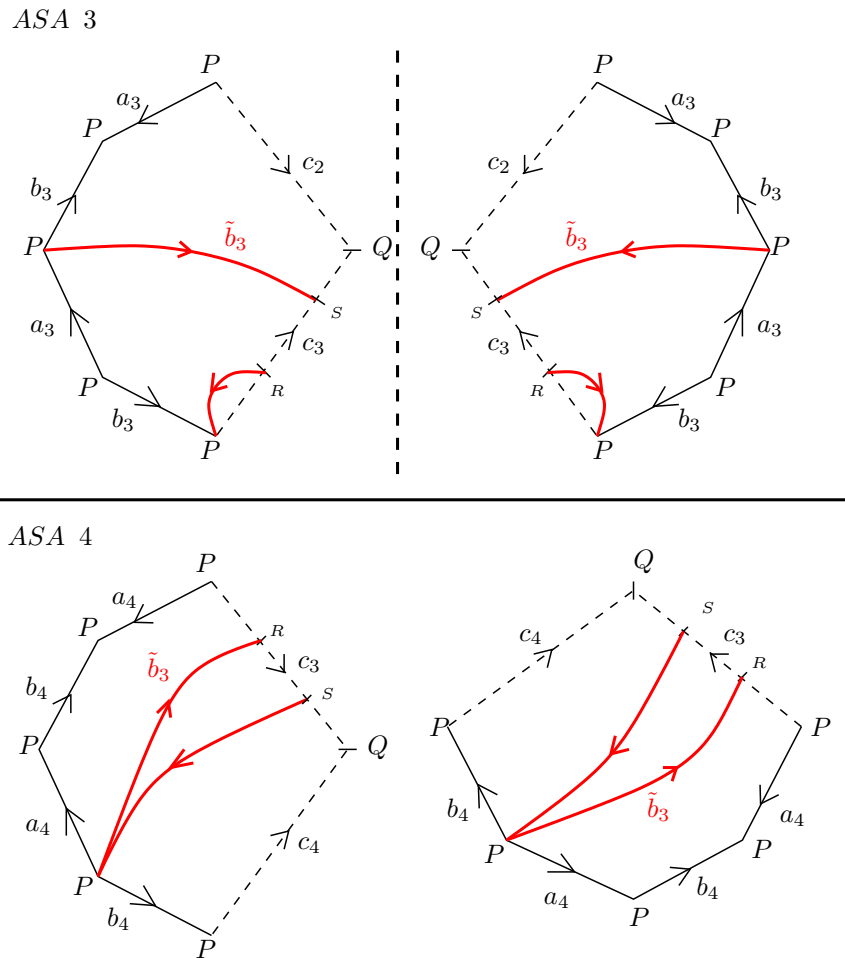
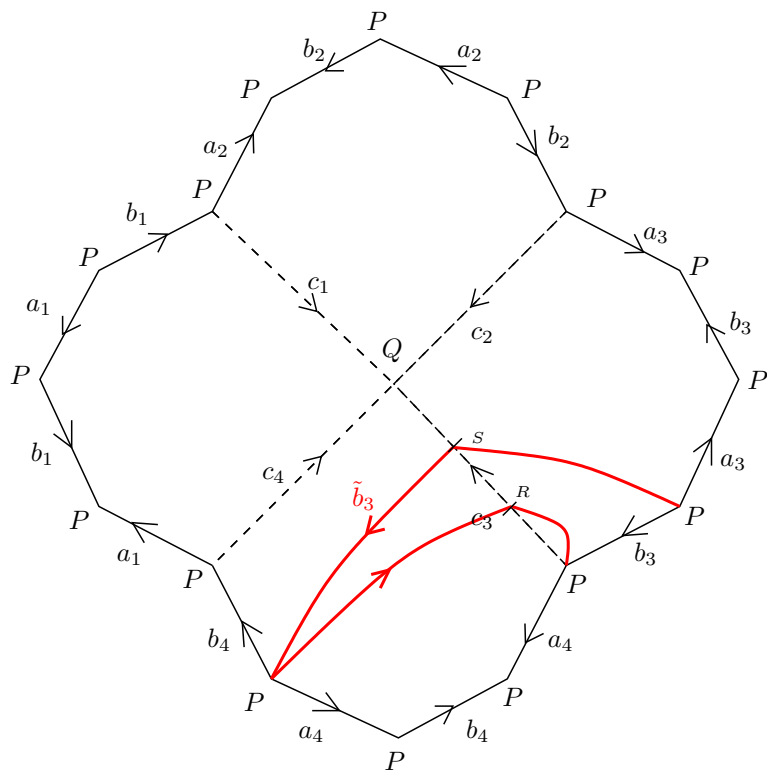


Figura 2.97: El camino imagen \tilde{c} representado en las asas del polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. En la Asa 3, aplicamos simetría especular, y en la Asa 4 rotación de 90° en sentido horario y luego simetría especular.

$\sigma_9 \tilde{b}_3$



$$\tilde{b}_3 = a_4 b_4 a_4^{-1} b_3 a_4 b_4^{-1} a_4^{-1}$$

Figura 2.98: Representación del camino imagen \tilde{c} en el polígono con los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$. La homotopía final es la deformación de \tilde{c} hasta el borde del polígono fundamental de C_4 .

$\sigma_{10} \tilde{b}_4$

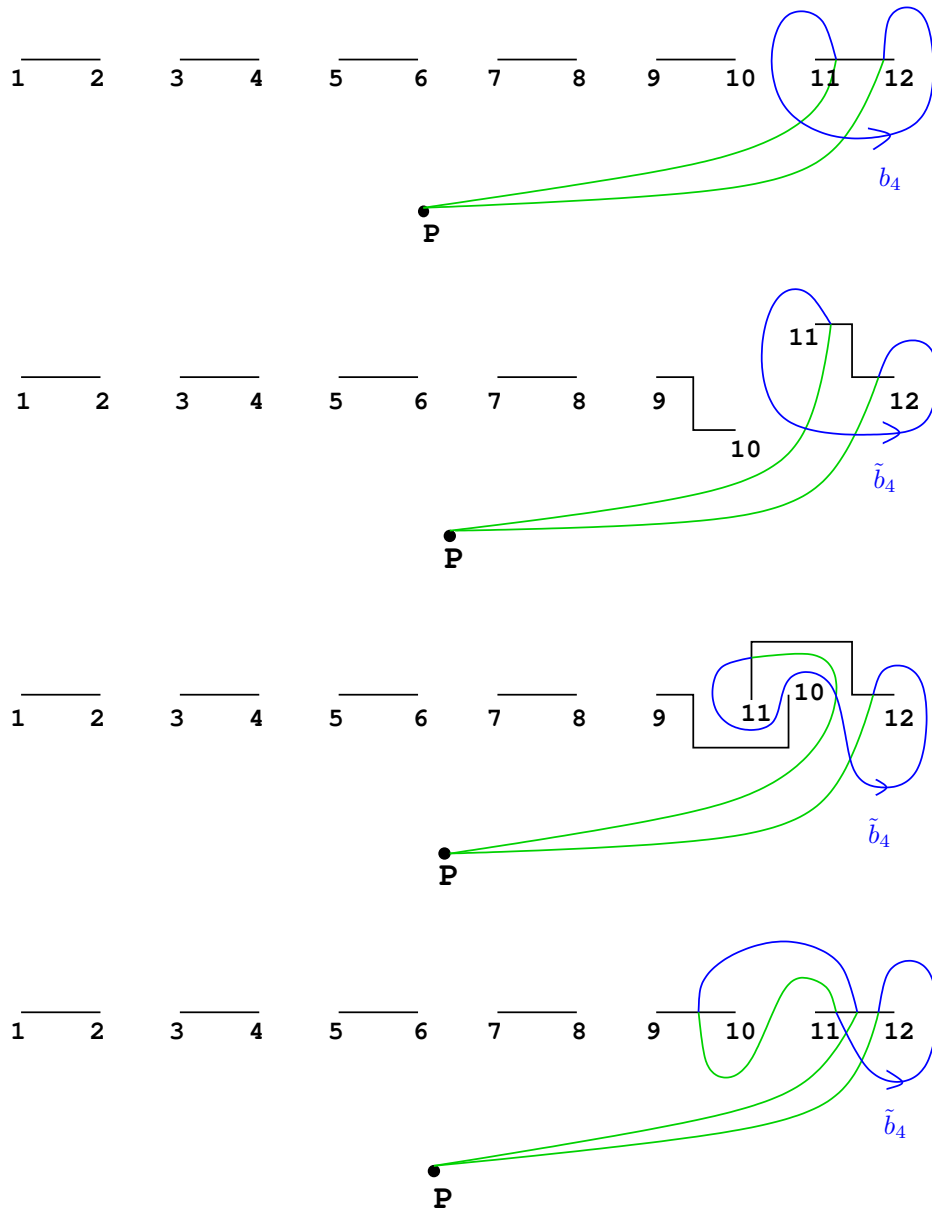


Figura 2.99: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

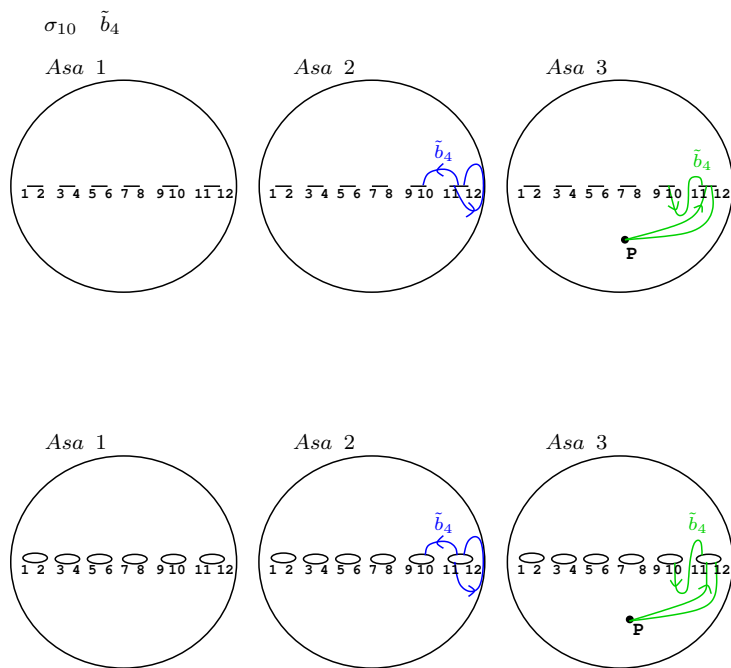


Figura 2.100: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

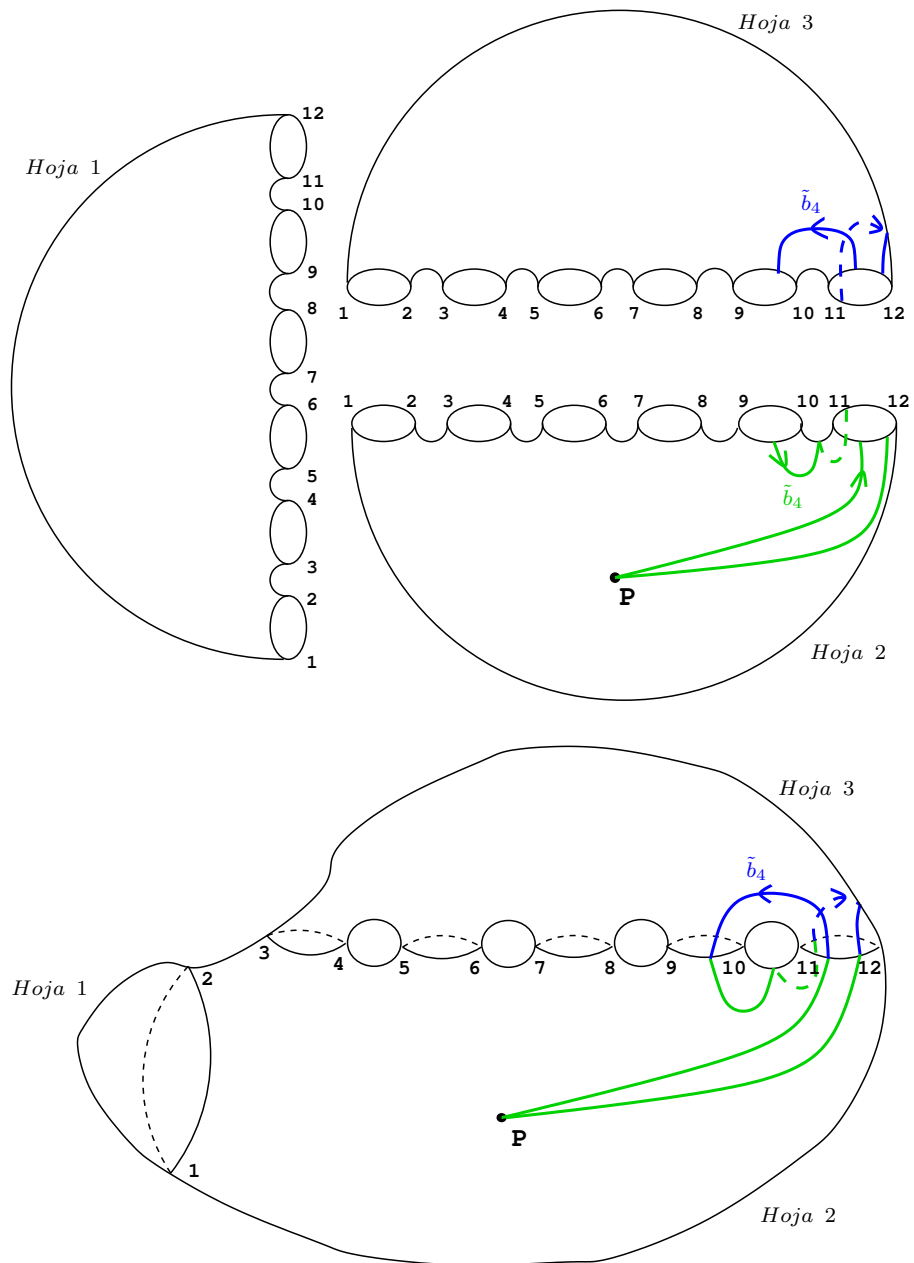


Figura 2.101: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

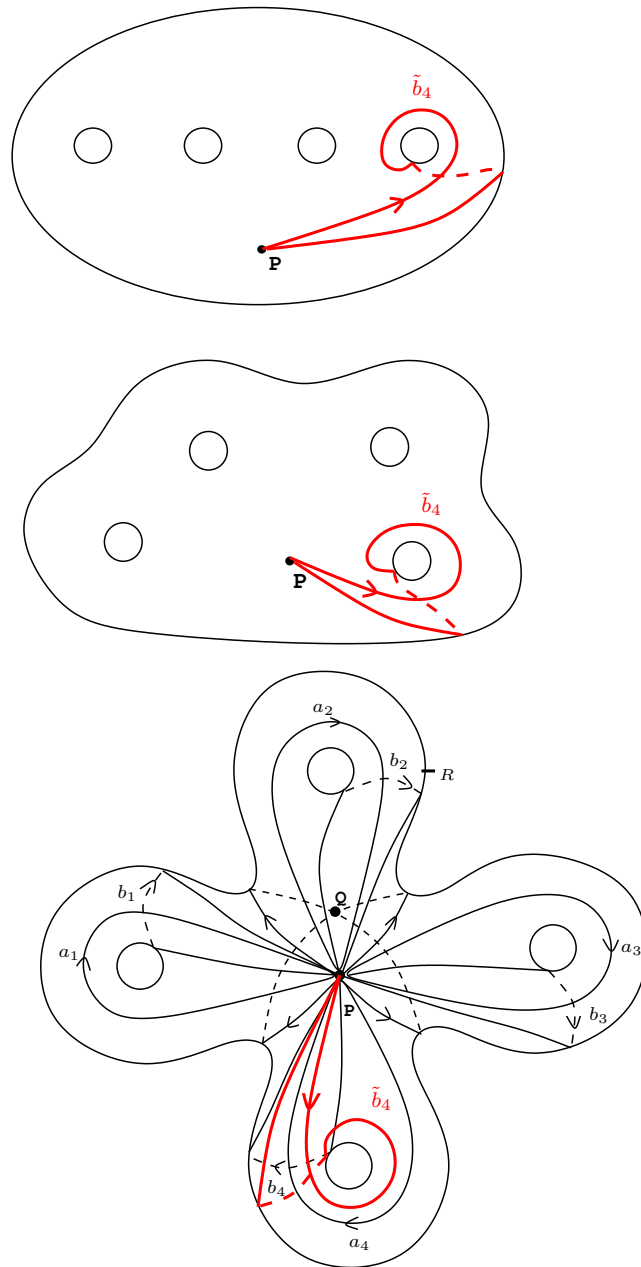


Figura 2.102: Camino imagen \tilde{c} en la esfera con 4 asas.

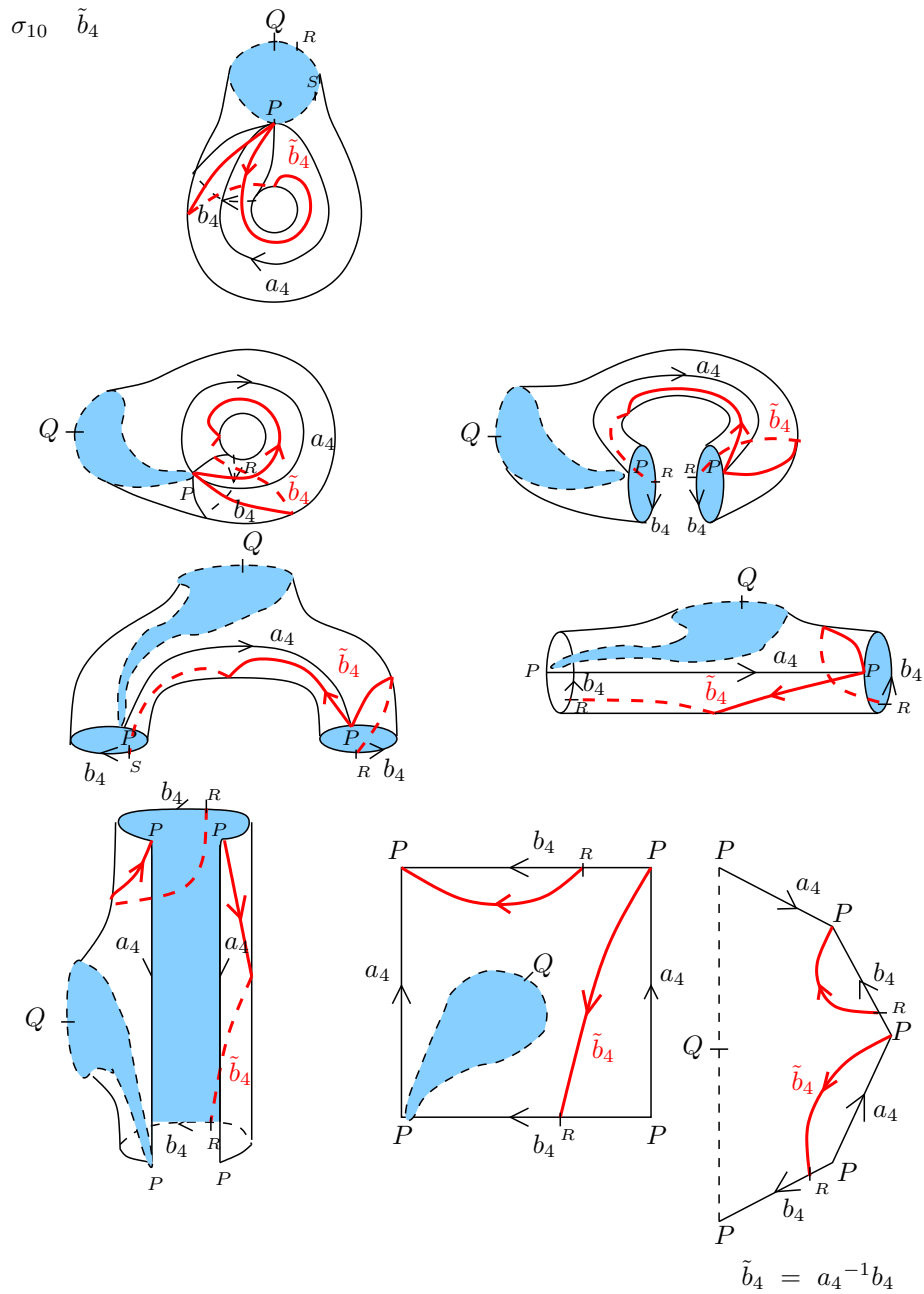


Figura 2.103: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

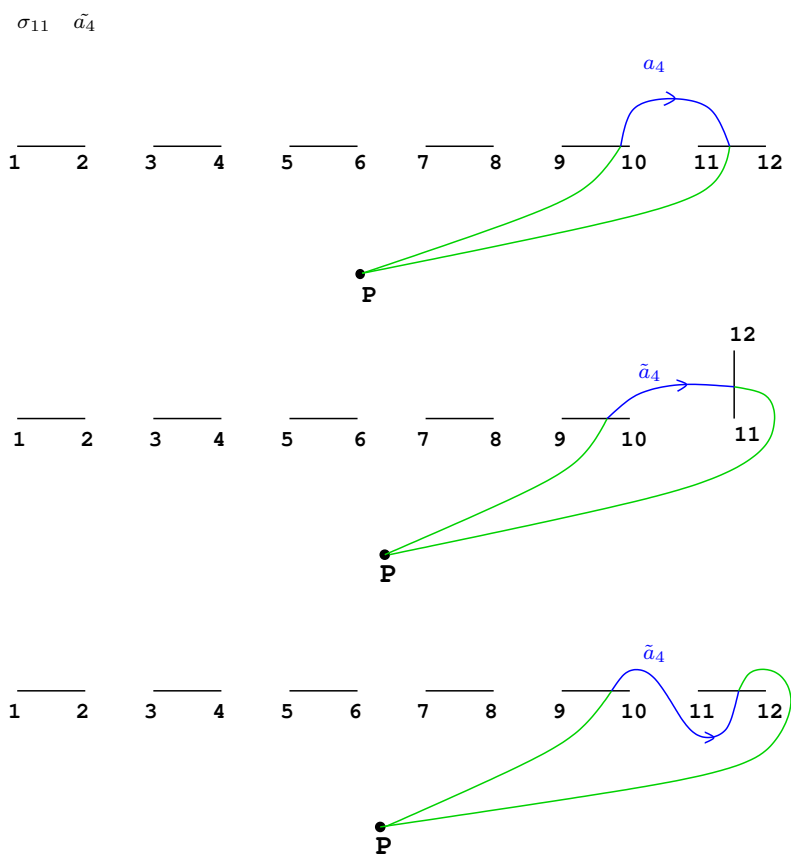


Figura 2.104: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

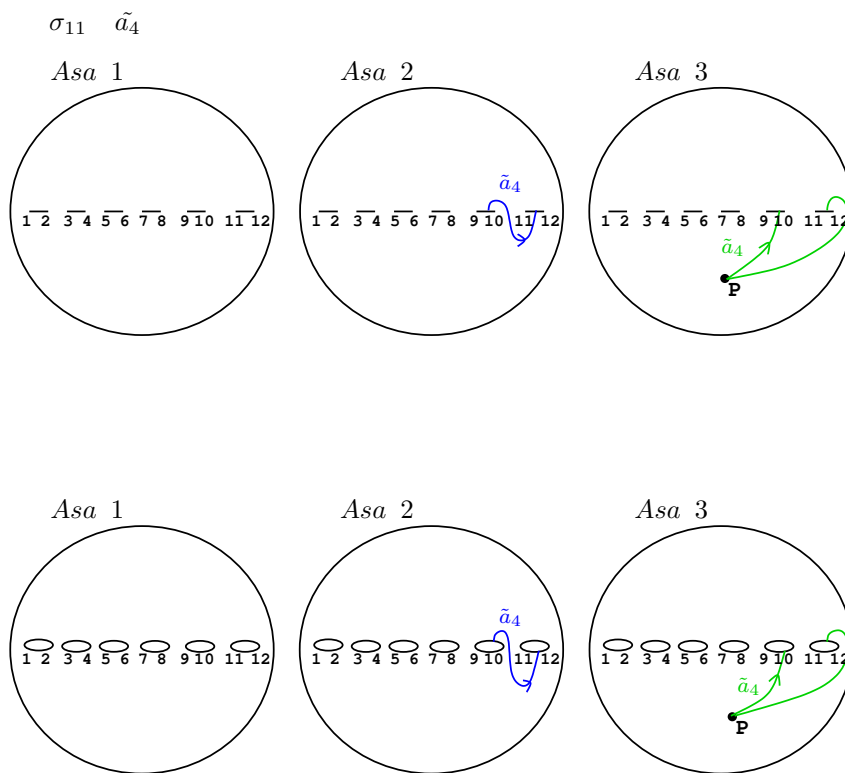


Figura 2.105: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

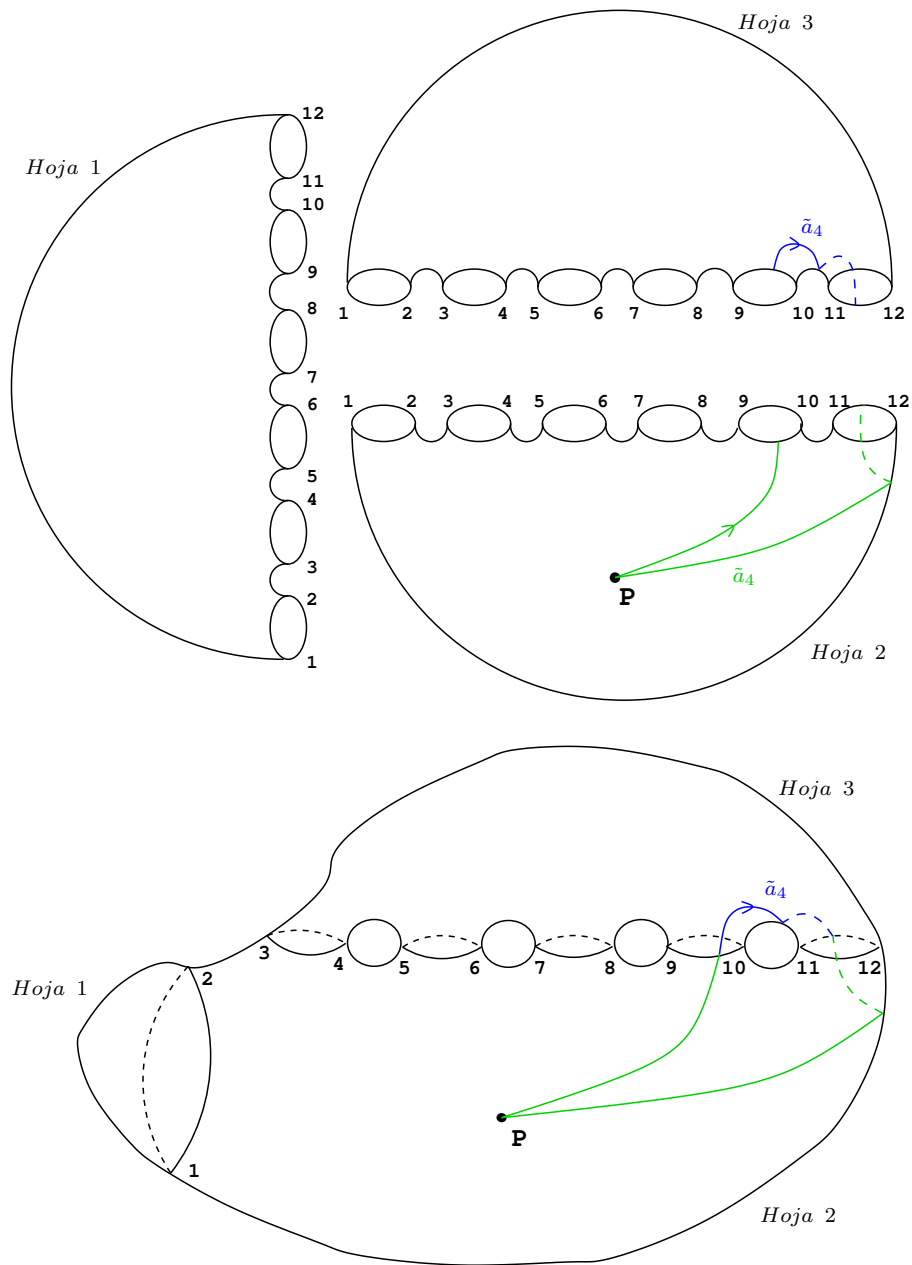


Figura 2.106: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

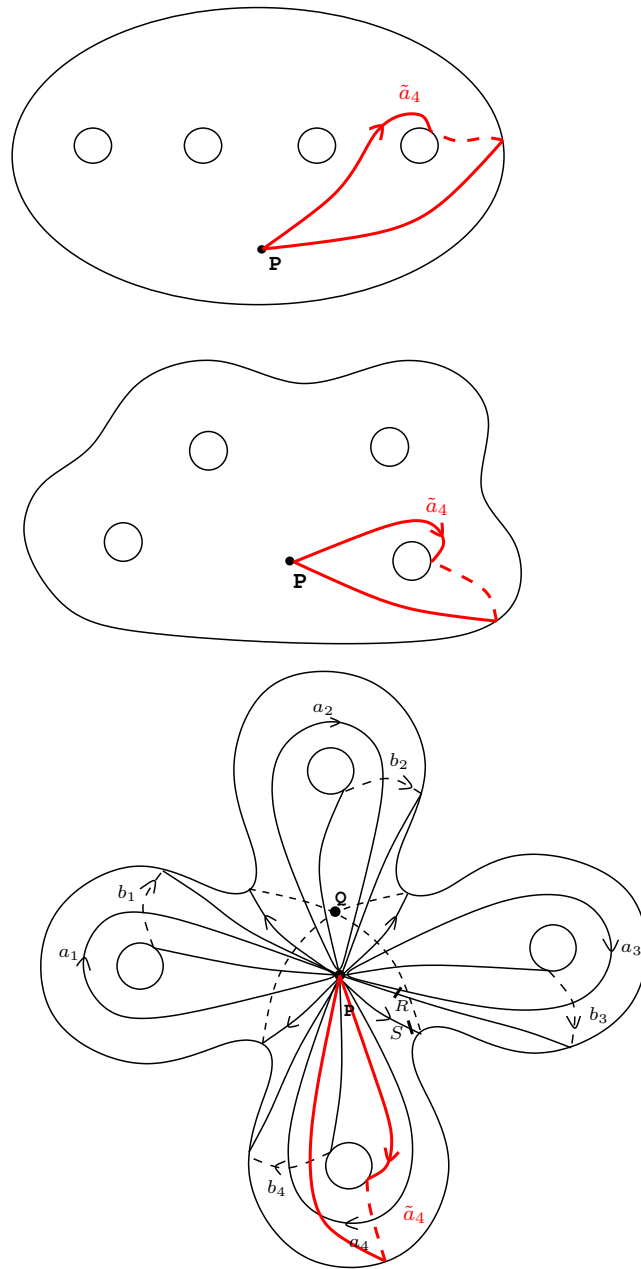


Figura 2.107: Camino imagen \tilde{c} en la esfera con 4 asas.

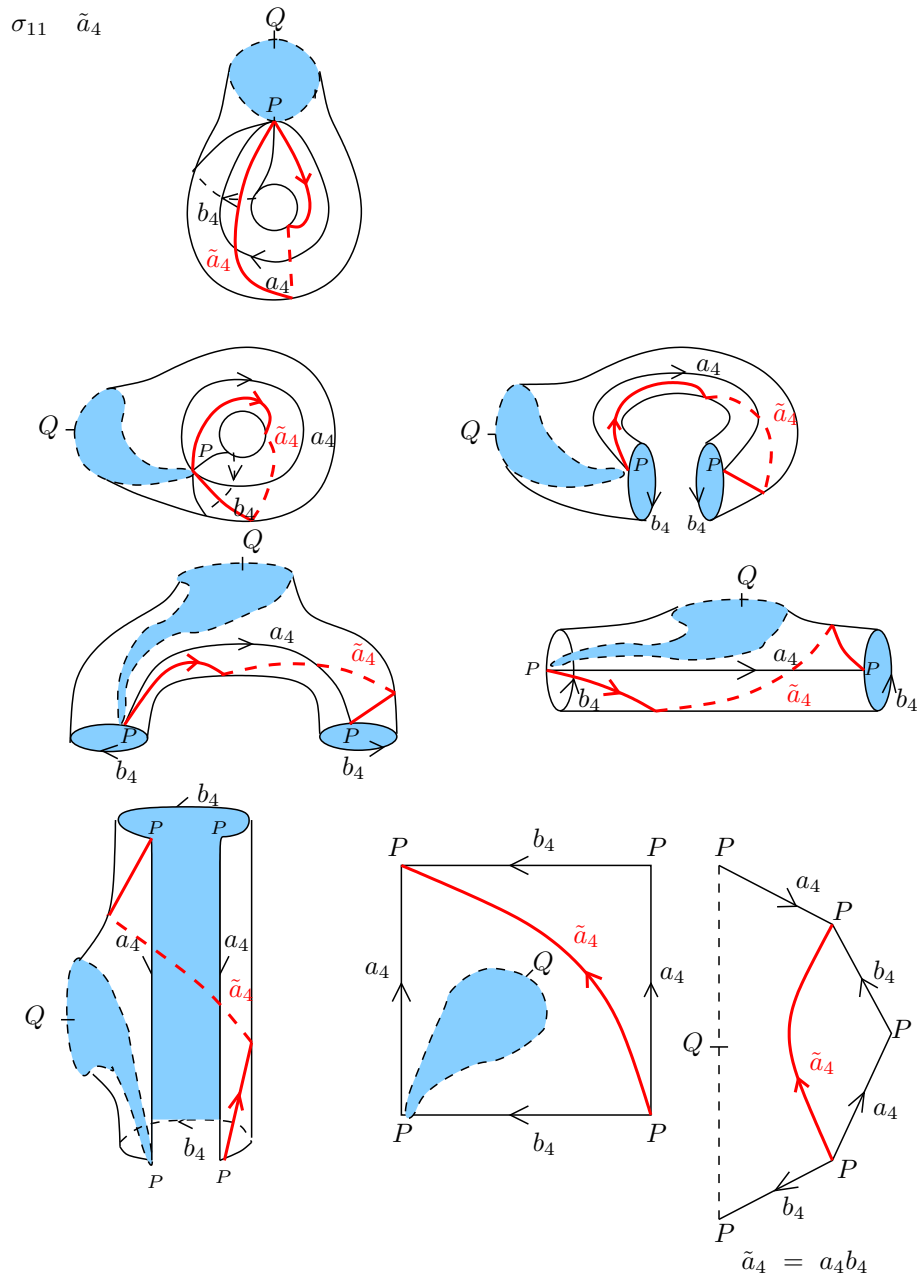


Figura 2.108: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

$\sigma_{11} \tilde{b}_4$

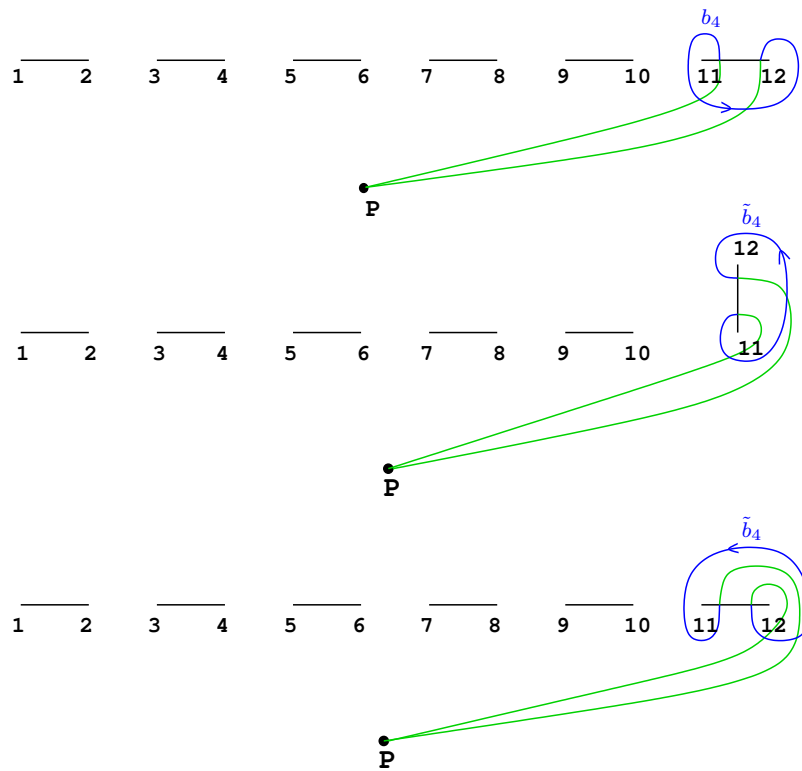


Figura 2.109: Movimiento de puntos de ramificación según la trenza σ y obtención del camino imagen \tilde{c} .

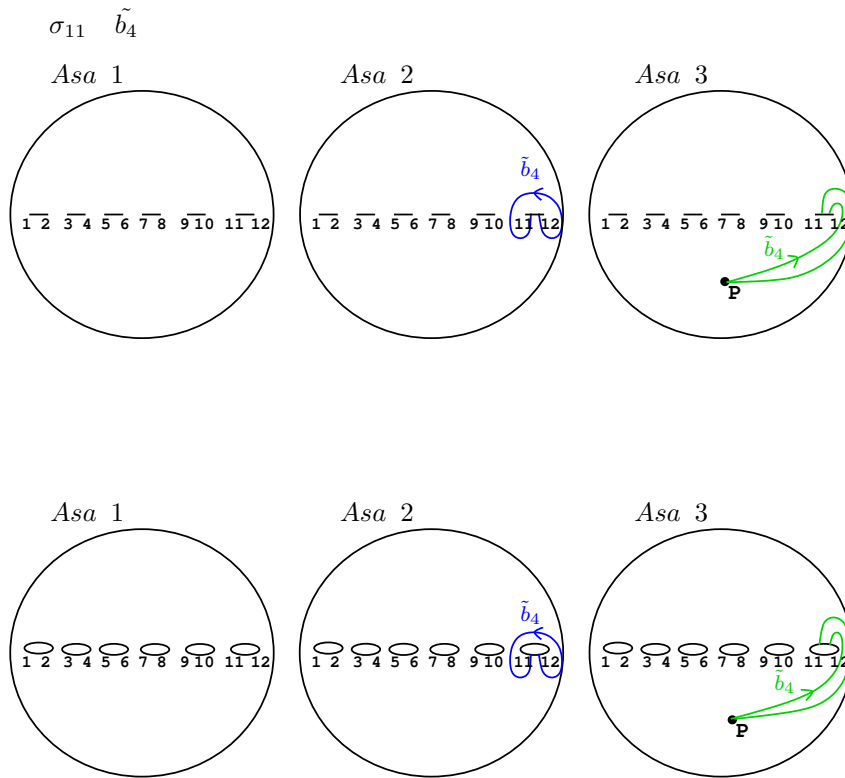


Figura 2.110: Representación del camino imagen \tilde{c} en $\mathbb{P}_{\mathbb{C}}^1$.

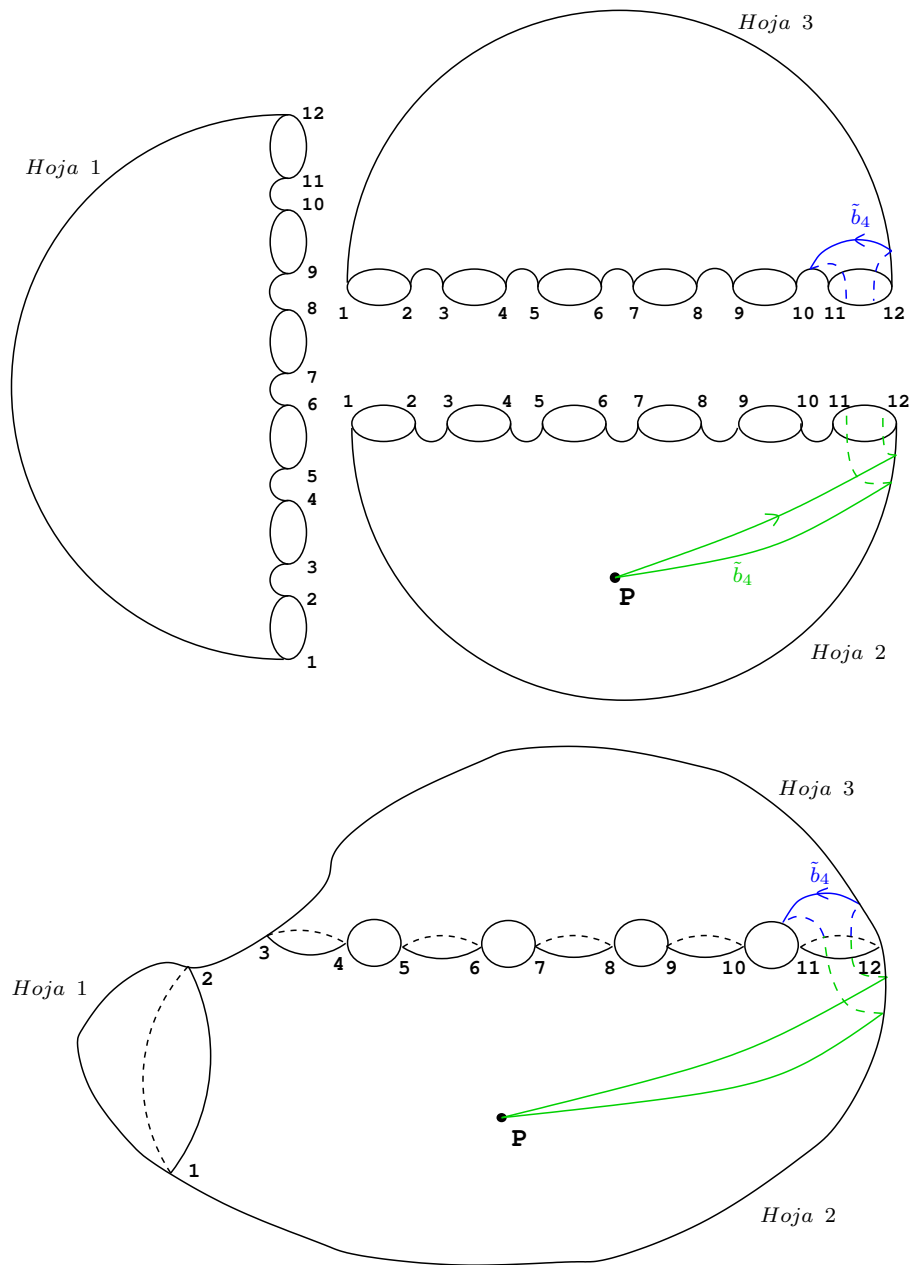


Figura 2.111: Levantamiento del camino imagen \tilde{c} al recubrimiento trigonal.

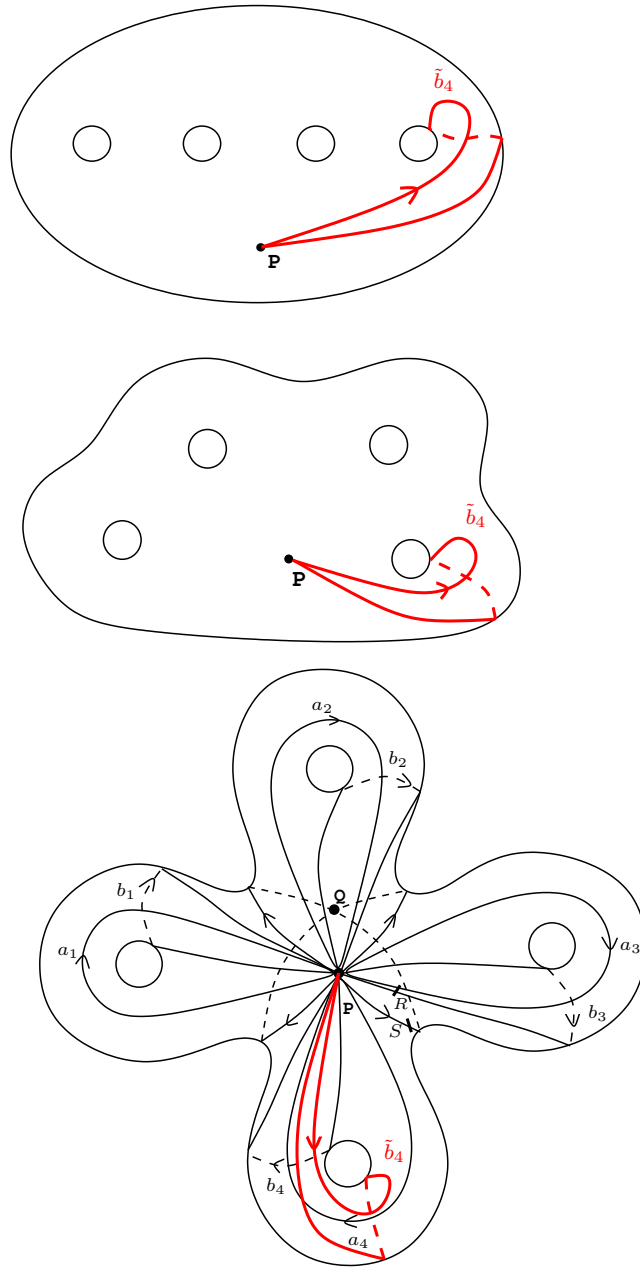


Figura 2.112: Camino imagen \tilde{c} en la esfera con 4 asas.

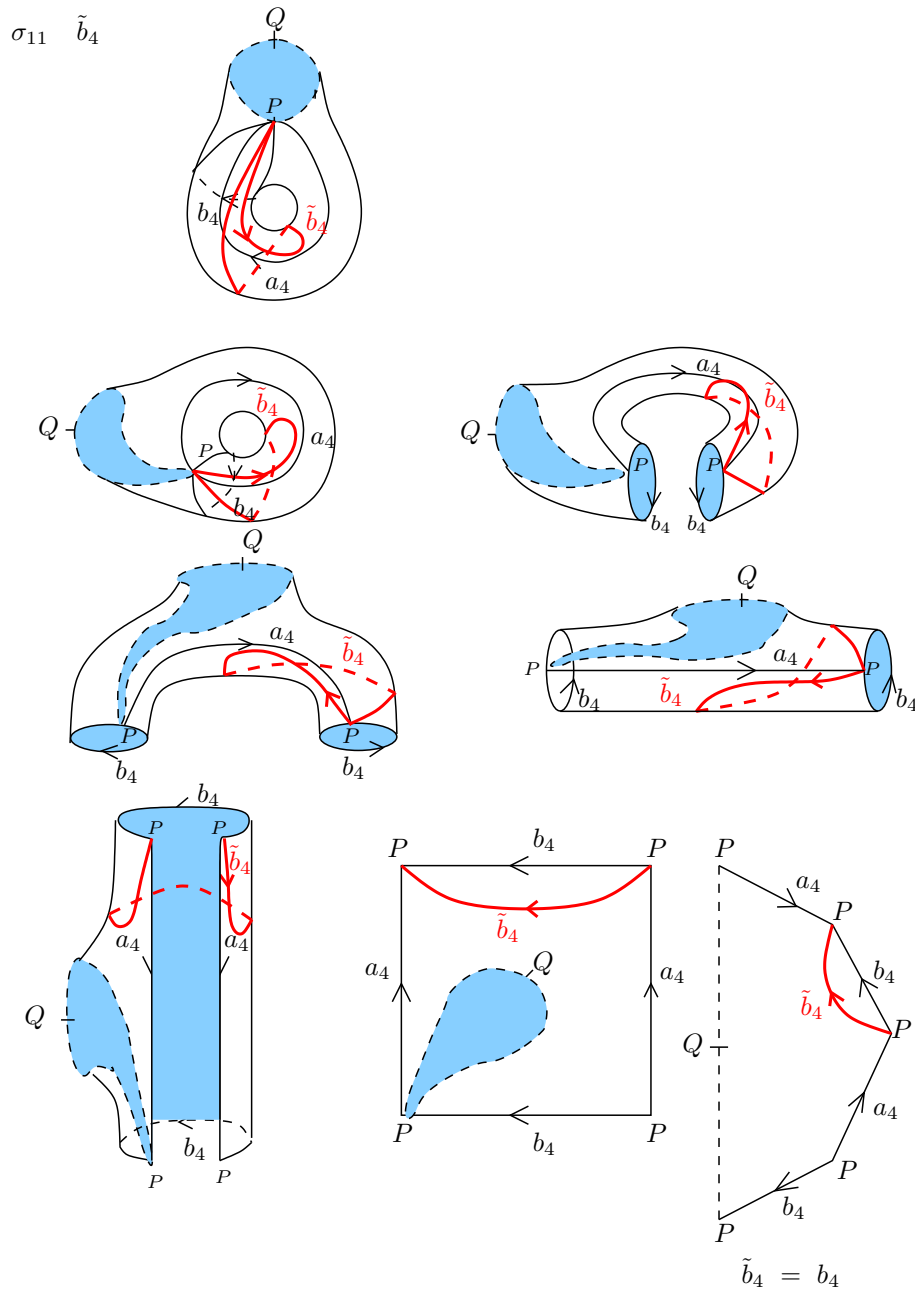


Figura 2.113: Deformación de una asa de la superficie, en la que se muestra paso a paso la conversión del camino imagen \tilde{c} en una palabra en los generadores originales $a_1, \dots, a_4, b_1, \dots, b_4$.

Capítulo 3

Cálculo

3.1. Cálculos Algebraicos

Para una curva $E = Q \cap C$:

- Llevamos la ecuación de la cuádrica Q a la ecuación reducida, para luego hallar sus reglajes, en caso de que Q es lisa, tendrá dos reglajes y si Q es un cono, tendrá un sólo reglaje. La reducción de la cuádrica se hace por completación de cuadrados y controla el rango de la cuádrica que aparece.
- Comprobamos si la intersección de la cuádrica y la cúbica es lisa, buscando singularidades en la intersección e identificando los puntos singulares. En caso liso, cortamos el reglaje con la cúbica para hallar la ramificación del morfismo trigonal.
- Para una familia 1-paramétrica de intersecciones $E_s = Q_s \cap C_s$, realizamos los cálculos anteriores correspondientes a familias 1-paramétricas. Si se detecta singularidad, crearemos una lista con los valores s para los que se registra.
- Hallamos el divisor de grado relativo 12 formado por los valores de ramificación.
- Hallamos la ramificación de la proyección del divisor de grado relativo 12, sobre el espacio de valores del parámetro.

Para realizar estos cálculos, hemos implementado una librería en Singular, con procedimientos específicos para las funciones anteriormente descritas, excepto para el cálculo de la ramificación del divisor de grado 12 en familias 1-paramétricas, que Singular no ha podido hacer debido a la complejidad doble exponencial del cálculo con bases de Gröebner.

Este último cálculo, lo hemos realizado en Pari-GP, al ser un programa que usa cálculo elemental. el resultado esperado se consiguió con éxito.

Lo descrito anteriormente se lleva a cabo utilizando la librería **genero4.lib** programada en Singular, cuyas **Funciones** son:

- QuadricReduction
- ParamQuadricReduction
- ProjCutSing
- ParamProjCutSing
- QuadricBranching
- ConeBranching
- ParamBranching
- cresultant

Descripción librería genero4.lib para Singular

- QuadricReduction: recibe el polinomio que define a la cuádrica Q , aquí se reduce por completación de cuadrados el polinomio de Q a la forma diagonal, entregando como resultado las matrices C (matriz del cambio de base canónica a reducida), D (matriz diagonal de la forma reducida), A (matriz de la cuádrica en base canónica).
- ParamQuadricReduction: recibe el polinomio que define a la cuádrica Q , analiza si la cuádrica Q es lisa o es un cono. Exporta el rango del polinomio de la cuádrica Q , matrices A , C , D (como en el caso anteriormente descrito).
- ProjCutSing: recibe los polinomios de la cuádrica Q y cúbica S , este procedimiento busca singularidades de la intersección en las distintas cartas afines, si existe, entrega una lista de valores.
- ParamProjCutSing: realiza el mismo procedimiento que ProjCutSing, pero en el caso 1-paramétrico.
- QuadricBranching: Trabaja en el anillo de polinomios complejo, recibe el polinomio de la cúbica S , halla los valores de ramificación del morfismo trigonal $S \cap Q \rightarrow \mathbb{P}^1$ (12 valores de ramificación y para cada uno; un punto doble y un punto simple). Devuelve los puntos de ramificación de dos maneras: en coordenadas (x, y, z) del espacio afín y en coordenadas (a, r) de una parametrización de la parte afín de Q por \mathbb{C}^2 .
 AVISO: Esta función puede no detectar los puntos dobles y devolver dos puntos simples con coordenadas casi idénticas en vez de uno doble.

- ConeBranching: como QuadricBranching pero devuelve puntos y valores de ramificación del morfismo trigonal de $S \cap Q$ en \mathbb{P}^1 dado por Q que es un cono estándar y sin único reglaje.
- ParamBranching: Corta la familia 1-paramétrica de cuádricas con la de cúbicas y halla el divisor de grado relativo 12 formado por los valores de ramificación de la familia de recubrimientos trigonales.
- cresultant: recibe dos polinomios f, g en un anillo complejo polinomial. Calcula la resultante respecto de la variable indicada por el tercer argumento h , y entrega el polinomio del determinante de la matriz resultante. Este procedimiento es llamado en ConeBranching y en QuadricBranching.

Programa de calculo con Pari-GP

Utilizamos el programa Pari-GP, para dos funciones de cálculo, estas son:

- calculo_ramif_r_verificacion.gp: Halla la ramificación de la proyección del divisor de grado relativo 12, obtenido del procedimiento ParamBranching,

Desde aquí llamamos al archivo que registra en formato texto el divisor, con variables r, s, u, v, w, b , donde:

s es el parámetro de la familia 1-paramétrica.

r es la coordenada afín en $\mathbb{P}_{\mathbb{C}}^1$ que contiene los valores de ramificación.

u, v, w, b son las raíces cuadradas necesarias para llevar la cuádrica a la forma estándar.

Sustituimos las variables u, v, w, b por 1, para dejar un polinomio al menos con dos variables. Calculamos el discriminante que da exactamente los valores de s para los que el divisor con variables r, s tiene raíces múltiples.

Aquí creamos la lista de los valores de s en los que hay ramificación y lo guardamos en un fichero texto leible por Matlab.

- pol2matriu.gp: Lee el polinomio divisor_r de la familia a estudiar, y lo guarda como una matriz de coeficientes, creando un script de Matlab en donde se realiza el cálculo de monodromías.

3.2. Listado de los programas del cálculo algebraico

3.2.1. Listado de los programas de la librería `genero_4.lib`

```
////////////////////////////////////  
version="$Id: genero4.lib,v 1.1 2011/05/18 $";  
category="Algebraic Geometry";  
info="  
LIBRARY:   genero4.lib  procedures to compute canonical curves of  
@*        genus 4 as trigonal coverings of the projective line  
AUTHOR:    JAUME AMOROS, email: jaume.amoros@upc.edu  
@*        ISABEL BERNA, email: isbernaib@hotmail.com  
KEYWORDS:  genus 4 curves, trigonal coverings, family of curves,  
@*        quadric surfaces  
PROCEDURES:  
cresultant(poly, poly, poly)  resultant polynomial in a complex  
@*                             ring  
ConeBranching(poly)          branching points of the trigonal  
@                             map from the intersection of a  
@*                             cone and a cubic surface to the  
@*                             projective line  
QuadricBranching(poly)      branching points of the trigonal  
@                             map from the intersection of a  
@*                             smooth quadric and a cubic surface  
@*                             to the projective line  
QuadricReduction(poly)      reduction of a quadric to diagonal  
@*                             form  
ProjCutSing(poly, poly)     detects singularities in the inter-  
@*                             section of a quadric and a cubic  
@*                             surface  
ParamProjCutSing(poly, poly) detects singularities in 1_paramete-  
@*                             ter intersection of cubics and  
@*                             quadrics  
ParamQuadricReduction(poly) reduction of a 1_parameter family  
@*                             of quadrics to diagonal form  
ParamBranching(poly, poly)  branching divisor of a 1_parameter  
@*                             family of trigonal coverings from  
@*                             canonical curves of genus 4";  
LIB "solve.lib";  
LIB "linalg.lib";
```



```

/////////////////////////////////////////////////////////////////
proc cresultant (poly f, poly g, poly h)
"USAGE:   resultant(f,g,h); f poly, g poly, h poly
RETURN:   poly: det(Mres)
NOTE:     Calculate the resultant of two polynomials f, g in a
@*       complex polynomial ring, with respect to the ring va-
@*       riable indicated by h
KEYWORDS: resultant, complex polynomials
EXAMPLE:  example resultant; shows an example"

{
//store the polynomial coefficients in ascending order
matrix Mg=coeffs(g,h);
Mg=transpose(Mg);
matrix Mf=coeffs(f,h);
Mf=transpose(Mf);

int degf=size(Mf)-1;
int degg=size(Mg)-1;
//rearrangement of the matrix coefficients in descending order
matrix Mfdec[1][degf+1];
matrix Mgdec[1][degg+1];
// completion of the matrix of the resultant
int j;
for (j=1;j<=degf+1;j++)
{
Mfdec[1,j]=Mf[1,degf+2-j];
}
for (j=1;j<=degg+1;j++)
{
Mgdec[1,j]=Mg[1,degg+2-j];
}

matrix Mres[degf+degg][degf+degg];

for (j=1;j<=degg;j++)
{
Mres[j,j..j+degf]=Mfdec;
}
for (j=1;j<=degf;j++)
{
Mres[j+degg,j..j+degg]=Mgdec;
}
// Determinant of the Resultant matrix
return(det(Mres));
}

```

```

example
{
ring R=(complex,12,I),(x,y),dp;
poly g=x3+y3-x2y+y2-1;
poly f=diff(g,x);
poly resc=cresultant(f,g,x);
resc;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
proc ConeBranching (poly S)
"USAGE: ConeBranching(S); S poly
RETURN: list: ramificacion_cono
NOTE: gives the list of branch points and values of the trigo-
@* nal map from the intersection of the standard cone
@* x2+y2=z2 and a cubic surface
KEYWORDS: cone, cubic surface, trigonal covering, branching points
EXAMPLE: example ConeBranching; shows an example"

{
ring R1=(complex,12,I), (a,r), dp;
// reparametrize the standard cone by a,r
map F=R, a*(r2-1),a*(r-2), a*(r2+1),1;
poly g=F(S);
matrix coef_a=coef(g,a);
poly a3_r=coef_a[2,1];
poly dg=diff(g,a);
poly valesp=cresultant(g,dg,a);
list qres=division(valesp,a3_r);
poly p_12=qres[1][1,1];
def ptos_p12_cono=laguerre_solve(p_12,10,30,1,0);

// Prepare list of 36 points for antiimages of 12 branch values,
// branchpoints appear multiplicity times; using same ring R1
list ptos_ar_cono;
list puntos;
list ar;
int j;
int k;
int ind=1;
number r_j;
poly g1;
map H;

```

```

for (j=1; j<=size(ptos_p12_cono); j++)
{
  r_j=ptos_p12_cono[j];
  H=R1, a,r_j;
  g1=H(g);
  puntos=laguerre_solve(g1,10,30,1,0);
  for (k=1; k<=size(puntos); k++)
  {
    ar=puntos[k],r_j;
    ptos_ar_cono[ind]=ar;
    ind++;
  }
}
number aj;
number rj;
number xj;
number yj;
number zj;
list xyz;
//prepare list of the values of x, y, z using a and r from para-
//metrization.
list ptos_xyz_cono;

for (j=1; j<=size(ptos_ar_cono);j++)
{
  aj=ptos_ar_cono[j][1];
  rj=ptos_ar_cono[j][2];
  xj=aj*(rj^2-1);
  yj=aj*(rj-2);
  zj=aj*(rj^2+1);
  xyz=xj,yj,zj;
  ptos_xyz_cono[j] = xyz;
}

list ptos_cono=ptos_p12_cono,ptos_ar_cono,ptos_xyz_cono;
setring R;
list ramificacion_cono=fetch(R1,ptos_cono);
return(ramificacion_cono);
}
example
{
ring R=(complex,12,I), (x,y,z,t), dp;
poly S=x^3+y^3-z^3-t^3-x*y*z+7*x^2*y+13*y^2*z+x*z^2-3*x*t^2+23*y*z*t-11*y^2*t;
ConeBranching(S);
ramificacion_cono;
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
proc QuadricBranching (poly S)
"USAGE:   QuadricBranching(S); S poly
RETURN:   list: ramificacion
NOTE:     gives the list of branch points of the trigonal map from
@*        the intersection of the standard smooth quadric
@*         $x^2+y^2-z^2-1=0$  and a cubic surface given

KEYWORDS: quadric, cubic surface, trigonal covering, branching points
EXAMPLE:  example QuadricBranching; shows an example"

{
  ring R1=(complex,12,I), (a,r), dp;
// parametrization of the affine standard quadric by a,r from
//  $A^1 \times A^1$ , we will use ruling with lines indexed by r, and
// coordinate a on each line.
  map F=R, -r+a*(1-r^2),1+2*a*r, r+ a*(r^2+1),1;
  poly g=F(S);
  matrix coef_a=coef(g,a);
  poly a3_r=coef_a[2,1];
  poly dg=diff(g,a);
  poly valesp=cresultant(g,dg,a);
  list qres=division(valesp,a3_r);
  poly p_12=qres[1][1,1];
  def ptos_p12=laguerre_solve(p_12,10,30,1,0);

//creating list of 36 points, with no discussion of multiplicity,
//using coordinates a and rj on the same ring R1
  list ptos_ar;
  list puntos;
  list ar;
  int j;
  int k;
  int ind=1;
  number r_j;
  poly g1;
  map H;

  for (j=1; j<=size(ptos_p12); j++)
  {
    r_j=ptos_p12[j];
    H=R1, a,r_j;
    g1=H(g);
    puntos=laguerre_solve(g1,10,30,1,0);
  }
}

```

```

for (k=1; k<=size(puntos); k++)
{
  ar=puntos[k],r_j;
  ptos_ar[ind]=ar;
  ind++;
}
}
number aj;
number rj;
number xj;
number yj;
number zj;
list xyz;
//list of the values of x, y, z using the values of a,r of the
//branchpoints
list ptos_xyz;

for (j=1; j<=size(ptos_ar);j++)
{
  aj=ptos_ar[j][1];
  rj=ptos_ar[j][2];
  xj=-rj+aj*(1-rj^2);
  yj=1+2*aj*rj;
  zj=rj+aj*(rj^2+1);
  xyz=xj,yj,zj;
  ptos_xyz[j] = xyz;
}

list ptos=ptos_p12,ptos_ar,ptos_xyz;
setring R;
list ramificacion=fetch(R1,ptos);
return(ramificacion);
}
example
{
ring R=(complex,12,i),(x,y,z,t),dp;//works in a complex ring
poly Q=x^2-3xy+2xz-2y^2+6yz+3z^2+2xt+7yt-zt+6t^2;//Smooth quadric
poly S=x^3+y^3-z^3-t^3-x*y*z+7*x^2*y+13*y^2*z+x*z^2;
QuadricBranching(S);
ramificacion;
}

```

```

/////////////////////////////////////////////////////////////////
proc QuadricReduction (poly Q)
"USAGE:   QuadricReduction(Q); Q poly
RETURN:  matrix: matrices
NOTE:    Calculate the reduction of a quadric complex polynomial
@*       to a diagonal form

KEYWORDS: complex polinomial, diagonal form
EXAMPLE:  example QuadricReduction; shows an example"

{
matrix A=jacob(jacob(Q))/2;
int i,j;
int n=nrows(A);

// reduction of a symmetric matrix to diagonal form
matrix C[n][n]=diag(1,n);
matrix P[n][n]=diag(1,n);
matrix D[n][n];
D=A;
for(i=1; i<4; i++)
{
for(j=i+1; j<=4; j++)
{
if(D[j,i]!=0)
{
if(D[i,i]==0)
{
P[i,j]=1;
D=P*D;
D=D*transpose(P);
C=P*C;
P[i,j]=0;
j=i+1;
}
P[j,i]=(-1)*(D[j,i]/D[i,i]);
D=P*D;
D=D*transpose(P);
C=P*C;
P[j,i]=0;
}
}
}
}
}

```

```

list matrices;
//Rank the matrix D diagonal and clasification in Cone or Smooth
//Quadric.
if (mat_rk(D)<=2)
{
"Quadric has rank below 3";
}
else
{
if(mat_rk(D)==3)
{
"The quadric is a cone";
list valores_raiz;
list puntos;
poly a;
//If the first element of the matrix A is zero, changing from
//place to fourth place
for (j=1; j<=4; j++)
{
if (D[j,j]==0)
{
i=j;
}
}
P[i,i]=0;
P[4,4]=0;
P[i,4]=1;
P[4,i]=1;
D=P*D;
D=D*transpose(P);
C=P*C;
P=diag(1,n);

poly sqrt;
ideal sqrt_1;
ideal sqrt_std;
def T;
//Calculates the square root function
for(i=1; i<4; i++)
{
a=D[i,i];
sqrt=x2-a;
sqrt_1=(sqrt,y,z,t);
sqrt_std=std(sqrt_1);
}
}

```

```

T=solve(sqrt_std,12,1,"nodisplay");
setring T;
list punto;
punto=SOL[1][1][1][1];
setring R;
puntos=fetch(T,punto);
valores_raiz=valores_raiz+puntos;
}

P[1,1]=1/valores_raiz[1];
P[2,2]=1/valores_raiz[2];
P[3,3]=-I/valores_raiz[3];
D=P*D;
D=D*transpose(P);
C=P*C;
P[1,1]=1;
P[2,2]=1;
P[3,3]=1;
}
else
{
"Smooth quadric";
list valores_raiz;
list puntos;
poly a;
poly sqrt;
ideal sqrt_1;
ideal sqrt_std;
def T;
for(i=1; i<=4; i++)
{
a=D[i,i];
sqrt=x2-a;
sqrt_1=(sqrt,y,z,t);
sqrt_std=std(sqrt_1);
T=solve(sqrt_std,12,1,"nodisplay");
setring T;
list punto;
punto=SOL[1][1][1][1];
setring R;
puntos=fetch(T,punto);
valores_raiz=valores_raiz+puntos;
}
}

```



```

P[1,1]=1/valores_raiz[1];
P[2,2]=1/valores_raiz[2];
P[3,3]=-I/valores_raiz[3];
P[4,4]=-I/valores_raiz[4];
D=P*D*transpose(P);
C=P*C;
P[1,1]=1;
P[2,2]=1;
P[3,3]=1;
P[4,4]=1;
}
matrices=C,D,A;
}
return(matrices);
}
example
{
ring R=(complex,12,I),(x,y,z,t),dp;//works in a complex ring
poly Q=x2-3xy+2xz-2y2+6yz+3z2+2xt+7yt-zt+6t2;//Smooth quadric
poly S=x3+y3-z3-xyz+7x2y-xy2+13y2z+xz2+5x2t-3yz+7xt-8zt-t3;
QuadricReduction(Q);
matrices;
}

////////////////////////////////////
proc ProjCutSing (poly S, poly Q)
"USAGE:    ProjCutSing(S, Q); S poly, Q poly
RETURN:   ring: Rfinal
NOTE:     detects singularities in the intersection of a quadric
@*        and a cubic surface to the affine space

KEYWORDS: quadric, cubic surface, trigonal covering, singularities
EXAMPLE:  example ProjCutSing; shows an example"

{
//Complex ring for saving solutions
ring Rfinal=(complex,12,i),(x,y,z,t),dp;
list puntos_singulares;
setring R;
//Enter a quadric with projective coordinates
ideal QS=(Q,S);
matrix dQdS=(jacob(QS));

```

```

ideal menores=((dQdS[1,1]*dQdS[2,2]-dQdS[1,2]*dQdS[2,1]),
(dQdS[1,1]*dQdS[2,3]-dQdS[1,3]*dQdS[2,1]),(dQdS[1,1]*dQdS[2,4]
-dQdS[1,4]*dQdS[2,1]),(dQdS[1,2]*dQdS[2,3]-dQdS[1,3]*dQdS[2,2]),
(dQdS[1,2]*dQdS[2,4]-dQdS[1,4]*dQdS[2,2],dQdS[1,3]*dQdS[2,4]
-dQdS[1,4]*dQdS[2,3]));
ideal menores_std=std(menores);
//Singularity looking at the various Charters Affine, beginning
//with {t/=0}
ideal singT=Q,S,menores_std,t-1;
ideal singT_std=std(singT);

if (dim(singT_std)==-1)
{
  "No singular points in the Charter Afin {T/=0}";
}
else
{
  if (dim(singT_std)==0)
  {
    "There are singular points in the Charter Afin {T/=0}";
    def psingT=solve(singT_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingT,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {T/=0}";
  }
}

// solve puntos con t=0,z/=0
ideal singZ=Q,S,menores_std,t,z-1;
ideal singZ_std=std(singZ);

if (dim(singZ_std)==-1)
{
  "No singular points in the Charter Afin {Z/=0} with T==0";
}
else
{
  if (dim(singZ_std)==0)
  {
    "There are singular points in the Charter Afin {Z/=0} with
    T==0";
  }
}

```

```

def psingZ=solve(singZ_std,12,"nodisplay");
setring Rfinal;
puntos_singulares=puntos_singulares+fetch(psingZ,SOL);
setring R;
}
else
{
  "Dimension is greater than zero in the Charter Afin {Z/=0}
  with T==0";
}
}

// solve puntos con t==0,z==0, y/=0
ideal singY=Q,S,menores_std,t,z,y-1;
ideal singY_std=std(singY);

if (dim(singY_std)==-1)
{
  "No singular points in the Charter Afin {Y/=0} with T==0, Z==0";
}
else
{
  if (dim(singY_std)==0)
  {
    "There are singular points in the Charter Afin {Y/=0} with
    T==0, Z==0";
    def psingY=solve(singY_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingY,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {Y/=0}
    with T==0, Z==0";
  }
}

// solve puntos con t==0,z==0, y==0, x/=0
ideal singX=Q,S,menores_std,t,z,y,x-1;
ideal singX_std=std(singX);

```

```

if (dim(singX_std)==-1)
{
  "No singular points in the Charter Afin {X/=0} with T==0, Z==0,
  Y==0";
}
else
{
  if (dim(singX_std)==0)
  {
    "There are singular points in the Charter Afin {X/=0} with
    T==0, Z==0, Y==0";
    def psingX=solve(singX_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingX,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {X/=0}
    with T==0, Z==0, Y==0";
  }
}
//list of points as global variable and return the ring in which
//they live.

setring Rfinal;
export(puntos_singulares);
return(Rfinal);
}
example
{
ring R=0,(x,y,z,t),dp;//ring of characteristic zero
poly Q=x2-3xy+2xz-2y2+6yz+3z2+2xt+7yt-zt+6t2;//Smooth quadric
poly S=x3+y3-z3-xyz+7x2y-xy2+13y2z+xz2+5x2t-3yz+7xt-8zt-t3;
def H=ProjCutSing(S);
setring H;
puntos_singulares;
}

////////////////////////////////////
proc ParamProjCutSing (poly S, poly Q)
"USAGE:   ParamProjCutSing(S, Q); S poly, Q poly
RETURN:   ring: Rfinal
NOTE:     detects singularities in 1_parameter intersection of cu-
@*        bics and quadrics to the affine space

```

KEYWORDS: quadric, cubic surface, trigonal covering, singularities
 EXAMPLE: example ParamProjCutSing; shows an example"

```
{
//Preparing the ring for saving solutions
ring Rfinal=(complex,12,i),(x,y,z,t,s),dp;
list puntos_singulares;
setring R;

//Enter a quadric anyone with projective coordinates
ideal QS=(Q,S);
matrix dQdS=(jacob(QS));
ideal menores=((dQdS[1,1]*dQdS[2,2]-dQdS[1,2]*dQdS[2,1]),
(dQdS[1,1]*dQdS[2,3]-dQdS[1,3]*dQdS[2,1]),(dQdS[1,1]*dQdS[2,4]
-dQdS[1,4]*dQdS[2,1]),(dQdS[1,2]*dQdS[2,3]-dQdS[1,3]*dQdS[2,2]),
(dQdS[1,2]*dQdS[2,4]-dQdS[1,4]*dQdS[2,2],dQdS[1,3]*dQdS[2,4]
-dQdS[1,4]*dQdS[2,3]));
ideal menores_std=std(menores);
//Singularity looking at the various Allied Letters, beginning
//with {t/=0}

ideal singT=Q,S,menores_std,t-1;
ideal singT_std=std(singT);

if (dim(singT_std)==-1)
{
  "No singular points in the Charter Afin {T/=0}";
}
else
{
  if (dim(singT_std)==0)
  {
    "There are singular points in the Charter Afin {T/=0}";
    def psingT=solve(singT_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingT,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {T/=0}";
  }
}
}
```

```

// solve puntos con t=0,z/=0
ideal singZ=Q,S,menores_std,t,z-1;
ideal singZ_std=std(singZ);

if (dim(singZ_std)==-1)
{
  "No singular points in the Charter Afin {Z/=0} con T==0";
}
else
{
  if (dim(singZ_std)==0)
  {
    "There are singular points in the Charter Afin {Z/=0} con
      T==0";
    def psingZ=solve(singZ_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingZ,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {Z/=0}
      con T==0";
  }
}

// solve puntos con t==0,z==0, y/=0
ideal singY=Q,S,menores_std,t,z,y-1;
ideal singY_std=std(singY);

if (dim(singY_std)==-1)
{
  "No singular points in the Charter Afin {Y/=0} con T==0, Z==0";
}
else
{
  if (dim(singY_std)==0)
  {
    "There are singular points in the Charter Afin {Y/=0} con
      T==0, Z==0";
    def psingY=solve(singY_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingY,SOL);
    setring R;
  }
}

```

```

else
{
  "Dimension is greater than zero in the Charter Afin {Y≠0}
  con T==0, Z==0";
}
}

// solve puntos con t==0,z==0, y==0, x≠0
ideal singX=Q,S,menores_std,t,z,y,x-1;
ideal singX_std=std(singX);

if (dim(singX_std)==-1)
{
  "No singular points in the Charter Afin {X≠0} con T==0, Z==0,
  Y==0";
}
else
{
  if (dim(singX_std)==0)
  {
    "There are singular points in the Charter Afin {X≠0} con
    T==0, Z==0, Y==0";
    def psingX=solve(singX_std,12,"nodisplay");
    setring Rfinal;
    puntos_singulares=puntos_singulares+fetch(psingX,SOL);
    setring R;
  }
  else
  {
    "Dimension is greater than zero in the Charter Afin {X≠0}
    con T==0, Z==0, Y==0";
  }
}

//list of points as global variables (puntos_singulares and
//s_singulares) and return the ring in which they live

setring Rfinal;

//list of values of "s" of the singular points detected in the
//related letters

```

```

int j;
list s_singulares;
list puntos;
for (j=1; j<=size(puntos_singulares); j++)
{
puntos=puntos_singulares[j][5];
s_singulares=s_singulares+puntos;
}
export(puntos_singulares);
export(s_singulares);
return(Rfinal);
}
example
{
ring R=0, (x,y,z,t,s), dp; //ring of characteristic zero
poly Qs=x2+y2-z2-(1+s2)*t2;//Smooth quadric
poly Ss=x3+y3-s*z3-s*t3-x*y*z+7*x2*y+13*y2*z+x*z2;
def H=ParamProjCutSing(Ss,Qs);
setring H;
puntos_singulares;
s_singulares;
}

////////////////////////////////////
proc ParamQuadricReduction (poly Q)
"USAGE:   ParamQuadricReduction(Q); Q poly
RETURN:   ring: Rqloc
NOTE:     Calculate the reduction of 1_parameter family of the
@*        complex polynomial to a diagonal form
@*        gives the polynomial range 3 and the polynomial range
@*        and gives the list of square of diagonal matrix

KEYWORDS: quadric, cubic surface, trigonal covering, singularities
EXAMPLE:  example ParamQuadricReduction; shows an example"

{
matrix A5=jacob(jacob(Qs))/2;

//Create matrix A with rows and columns 1 .. 4 1 .. 4, A5

intvec v=1,2,3,4;
matrix A=submat(A5,v,1..4);

```



```

int i,j;
int n=nrows(A);
list s_cono;
list s_degenerada;

//ideal values of s for which the range is less than 4, less
//than 3

ideal menor4=minor(A,4);
ideal menor3=minor(A,3);

ideal menor4std=std(menor4);
ideal menor3std=std(menor3);

poly det4=menor4std[1];
poly rangom3=menor3std[1];
//the s where this polynomial vanishes A range are less than 3
//"very degenerate quadric"

list rango3=division(det4,rangom3);
//the "s" where do rango3 vanishes rank A = 3 "Cone"

poly rango_3=rango3[1][1,1];
ring Rloc_inicial=(0,s), (x,y,z,t), dp;
//polynomials in x, y, z, t on the field Q (s)

matrix A=imap(R,A);

int n=nrows(A);
matrix C[n][n]=diag(1,n);
matrix P[n][n]=diag(1,n);
matrix D[n][n];
D=A;
for(i=1; i<4; i++)
{
  for(j=i+1; j<=4; j++)
  {
    if(D[j,i]!=0)
    {
      if(D[i,i]==0)
      {
        P[i,j]=1;
        D=P*D;
        D=D*transpose(P);
        C=P*C;
        P[i,j]=0;
        j=i+1;
      }
    }
  }
}

```

```

        P[j,i]=(-1)*(D[j,i]/D[i,i]);
        D=P*D;
        D=D*transpose(P);
        C=P*C;
        P[j,i]=0;
    }
}
}

ring Rloc_u=(0,s), (x,y,z,t,u(1..4)), dp;

matrix D=imap(Rloc_inicial,D);

ideal relacio=u(1)^2-D[1,1],u(2)^2-D[2,2],u(3)^2+D[3,3],u(4)^2+D[4,4];

qring Rqloc=std(relacio); //let u2=s2+1
matrix A=imap(Rloc_inicial,A);
matrix C=imap(Rloc_inicial,C);
matrix D=imap(Rloc_inicial,D);
matrix P=imap(Rloc_inicial,P);

list cuadrados=D[1,1],D[2,2],-D[3,3],-D[4,4];
P[1,1]=u(1)/D[1,1];
P[2,2]=u(2)/D[2,2];
P[3,3]=-u(3)/D[3,3];
P[4,4]=-u(4)/D[4,4];

C=P*C;
D=P*D*transpose(P);

poly poly_rango_3=imap(R,rango_3);
poly poly_rangom3=imap(R,rangom3);

export(poly_rango_3);
export(poly_rangom3);

export(A);
export(C);
export(D);

export(cuadrados);

return(Rqloc);
}

```

```

example
{
//polynomials in x, y, z, t, s on the field Q (s)
ring R=0, (x,y,z,t,s), dp;
poly Qs=x2+y2-z2-(1+s2)*t2;
poly S=x3+y3-s*z3-s*t3-x*y*z+7*x2*y+13*y2*z+x*z2;
def T=ParamQuadricReduction(Qs);
setring T;
poly_rango_3;
poly_rangom3;
print(A);
print(C);
print(D);
cuadrados;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
proc ParamBranching(poly Qs, poly S)
"USAGE:   ParamBranching(Qs, S); Qs poly, S poly
RETURN:  ring: R1q
NOTE:    branching divisor of a 1_parameter family of trigonal
@*       coverings from canonical curves of genus 4 and the list
@*       of square of diagonal matrix

KEYWORDS: quadric, cubic surface, trigonal covering, resultant
EXAMPLE:  example ParamBranching; shows an example"

{
def Rqloc=ParamQuadricReduction(Qs);
setring Rqloc;
matrix reforiginal[4][1]=x,y,z,t;
matrix refcanonica=transpose(C)*reforiginal;
map S_C=R, refcanonica[1,1],refcanonica[2,1],refcanonica[3,1],
refcanonica[4,1],s;
poly gS=S_C(S);
ring R1=(0,s), (a,r,u,v,w,b), dp;
int save=short;
short=0;

list cuadrados=imap(Rqloc,cuadrados);
ideal raices=u2-cuadrados[1],v2-cuadrados[2],w2-cuadrados[3],
b2-cuadrados[4];
qring R1q=std(raices);
short=0;
map F=Rqloc, -r+a*(1-r^2),1+2*a*r, r+ a*(r^2+1),1,u,v,w,b;
poly gS_r=F(gS);

```

```

list cuadrados=imap(R1,cuadrados);

matrix coef_a=coef(gS_r,a);
poly a3_r=coef_a[2,1];
poly dg=diff(gS_r,a);
poly valesp=resultant(gS_r,dg,a);
list qres=division(valesp,a3_r);
poly divisor_r0=qres[1][1,1];
poly divisor_r=cleardenom(qres[1][1,1]);

export(divisor_r);
export(cuadrados);
return(R1q);
}
example
{
//polynomials in x, y, z, t, u on the field Q (s)
ring R=0, (x,y,z,t,s), dp;
poly Qs=x2+y2-z2-t2;
poly S=x3+y3-s*z3-s*t3-x*y*z+7*x2*y+13*y2*z+x*z2;
def T=ParamBranching(Qs, S);
setring T;
divisor_r;
cuadrados;
}

```

3.2.2. Listado programas Pari-GP

Listado programa calculo_ramif_r_verificacion.gp

```
allocatemem(800000000);

\\ IMPORTANTE: COMO CAMBIAR DE FAMILIA PARA EL CALCULO
\\ Hay que cambiar comentando/descomentando lineas en
\\ 3 PUNTOS del programa. Estan indicados en el listado.

\\ leemos el divisor_r

\\ PUNTO 1 DE CAMBIO DE FAMILIA: el divisor de ramificacion
\\ Familia simple 1:
  \r divisor_r_Q_x2+y2-z2-t2_S_t3s+x3+y3+xyz+z3.txt
p=%;
\\ substituimos u,v,w,b por 1
p1=subst(p,u,1);
p11=subst(p1,v,1);
\\\pfinal=p11;
p111=subst(p11,w,1);
p1111=subst(p111,b,1); b=1 solo en la familia simple numero 1
pfinal=p1111;

\\ discriminante da exactamente valores de s
\\ para los que pfinal tiene raices multiples
\\ (cuando se calcula el discriminante no hace falta
\\ dividir por el coeficiente de grado maximo de pfinal)
discpfinal=poldisc(pfinal,r);
\\ PUNTO 2 DE CAMBIO DE FAMILIA: el vector de cuadrados de
\\ la diagonal de la cuadrica
  \r cuadrados_Q_x2+y2-z2-t2_S_t3s+x3+y3+xyz+z3.txt \\familia_1

cuadrados=%;

\\ Simplificacion: aplicar que b^2 es igual a cuadrados[4]
\\ (dejarla comentada en las familias simples 1 y 2)
simplificador=x^2-cuadrados[4];
discpfinalbx=subst(discpfinal,b,x);
discpsimplbx=discpfinalbx%simplificador;
discpsimpl=subst(discpsimplbx,x,b);

\\ en la familia simple numero 2, al dividir por b^2-s
\\ desaparecen todas las b, esto permite pasar directamente a hallar
\\ los ceros del polinomio en s ramif_r
\\ Si no desaparecen todas las b: multiplicar por el conjugado
\\ respecto de b, y substituir b^2 por su valor en s
```

```

conjdisc=subst(discpsimpl,b,-b);
producto=discpsimpl*conjdisc;
productobx=subst(producto,b,x);

\\ dividimos por el mcd (entero) de los coeficientes
productosimpl=productobx%simplificador;
\\ lista de los valores de s en los que hay ramificacion
productosimplsx=subst(productosimpl,s,x);
productosimplsx=productosimplsx/content(productosimplsx);
ramif_r=subst(productosimplsx,x,s);

\\ bajamos la precision para que pari-gp no escriba con tantos
\\ decimales
default(realprecision, 12);
arrels=polroots(ramif_r);

\\ escribe la lista de valores de s de ramificacion en un
\\ fichero para matlab
\\ PUNTO 3 DE CAMBIO DE FAMILIA: fichero en que dejar los valores
\\ de ramificacion
\\ Descomentar la linea que pone el nombre de la familia que toca
\\ Familia Simple 1
nomramif="ramificacio_Q_x2+y2-z2-t2_S_t3s+x3+y3+xyz+z3.txt";

\\ guarda la lista de valores de ramificacion en columna en
\\ fichero de texto
\\ (leible desde Matlab con el comando load)
for (j=1, length(arrels), write(nomramif,real(arrels[j]),
"+(",imag(arrels[j]),")*i" ) );

\\ restaura la configuracion por defecto de pari-gp
default(realprecision, 28);

```

Listado programa calculo_ramif_r_verificacion.gp

```

/* lee el divisor_r de la familia
(que es un polinomi de debo)
y guarda como una matriz de coeficientes
a punto para el calculo de monodromias con Matlab
(files=potencias de r, columnes=potencias de s,
de grado maximo al termino independiente */

\\ AQUI EL USUARIO ENTRA A MA EL NOM DE LA FAMILIA
\\ lectura del polinomi0 deL fichero de texto
\\ Familia de Seidel
\r divisor_r_Q_-t2s2+x2+y2-z2_S_-z3s-t3s+x3+7x2y+y3-xyz+13y2z+xz2.txt

```

```

p=%;
\\ AQUI EL USUARIO ENTRA A MA LOS VALORES DE u,v,w,b QUE DE
\\ MOMENTO HAN DE SER POLINOMIOS EN s
\\ ponemos las raices cuadradas a 1
p1=subst(p,u,1);
p11=subst(p1,v,1);
p111=subst(p11,w,1);
p1111=subst(p111,b,1);

\\ escribimos el polinomis en 2 variables como una matriz
m=poldegree(p1111,r)+1;
n=poldegree(p1111,s)+1;

f=matrix(m,n);

for(a=1,m,for(b=1,n,f[a,b]=polcoeff(polcoeff(p1111,n-b,s),m-a,r)));

\\ crea matriz con parte real y parte imaginaria de los coeficientes
freal=real(f);
fimag=imag(f);

\\ AQUI EL USARIO PUEDE ENTRAR A MA EL NOMBRE DEL FICHERO DE SALIDA
\\ (O MODIFICARLO DESDE EL SISTEMA OPERATIVO)
\\ guarda la matriz en un script de Matlab
write1("matriu_divisor.m","freal=");
write1("matriu_divisor.m",freal);
write("matriu_divisor.m",";");
write1("matriu_divisor.m","fimag=");
write1("matriu_divisor.m",fimag);
write("matriu_divisor.m",";");

```

3.3. Cálculos Analíticos

3.3.1. Fundamento Teórico del Cálculo Analítico

Situación

$$\begin{array}{ccc} (x, y) \in C = \{f(x, y) = 0\} \subset \mathbb{C}^2 & \text{curva afin} & \\ \downarrow \pi & & \downarrow \\ x & & \mathbb{C} \end{array}$$

$\pi|_C : C \rightarrow \mathbb{C}$ presenta C como recubrimiento ramificado de \mathbb{C} . Para cada $x \in \mathbb{C}$ tenemos su fibra $\pi^{-1}(x) = \{(x, y_1), (x, y_2), \dots, (x, y_n)\}$ donde $n = \text{grado en } y \text{ de } f(x, y)$.

Los valores de ramificación del recubrimiento son los $x \in \mathbb{C}$ tales que $\pi^{-1}(x)$ no contiene n puntos distintos, sino algún punto múltiple.

Un cálculo rápido y conocido muestra.

Proposición 4. *Sea $x \in \mathbb{C}$ no es valor de ramificación de $C \xrightarrow{\pi} \mathbb{C} \iff$ para todos los puntos de la fibra $\pi^{-1}(x) = \{(x, y_1), \dots, (x, y_n)\}$ existe una parametrización local de C de la forma $y = y(x)$, con $f(x, y(x)) = 0$.*

Este recubrimiento nos permite partir de un punto $(x, y(x)) \in C$ y seguir la curva C al mover $x \in \mathbb{C}$:

Como $f(x, y(x)) = 0$ en C , por derivación implícita

$$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial x} = 0 \quad \text{en } C$$

Es decir

$$\frac{\partial y}{\partial x} = -\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}} \quad \text{en } C$$

Como $x \in \mathbb{C}$, que tiene dimensión real 2, podemos tomar un camino $x(t), t \in [0, 1]$ en \mathbb{C} , y nuevamente podemos seguir los puntos $(x(t), y(x(t))) \in C$.

A partir de la derivación implícita.

$$\begin{aligned} \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial x} \cdot \frac{\partial x}{\partial t} &= 0 \\ \Rightarrow \frac{\partial y}{\partial t} &= -\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}} \cdot \frac{\partial x}{\partial t} \end{aligned} \quad (3.1)$$

En la práctica el camino $x(t)$ será recto: una arista con vértice inicial $x_0 \in \mathbb{C}$ y vértice final $x_f \in \mathbb{C}$. Su parametrización será:

$$x(t) = (1-t)x_0 + tx_f, t \in [0, 1]$$

$$\Rightarrow \frac{\partial x}{\partial t} = x_f - x_0 \quad (\text{el vector director})$$

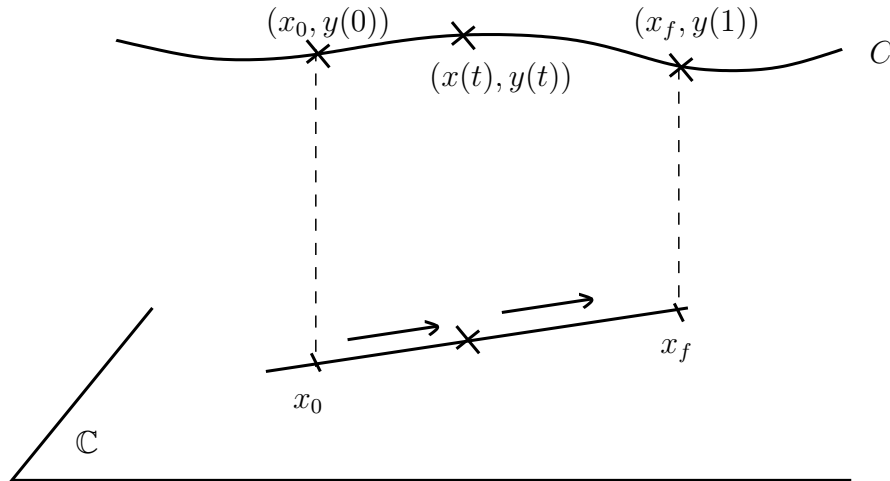


Figura 3.1: camino $x(t)$

Así $f(x, y) = 0$ es la ecuación de la curva (conocida), $\frac{\partial x}{\partial t} = x_f - x_0$, (determinado por la arista sobre la que integramos), y si tenemos un punto de la curva $(x_0, y_0) \in C$, sobre x_0 , podemos transportar este punto en la curva sobre el camino $\overline{x_0 x_f}$ integrando la ecuación diferencial.

$$\frac{\partial y}{\partial t} = -\frac{\frac{\partial f}{\partial x}(x, y)}{\frac{\partial f}{\partial y}(x, y)} \cdot (x_f - x_0) \quad (3.2)$$

para $t \in [0, t]$, con condición inicial $y(0) = y_0$.

Método de integración del problema de valores iniciales (3.2):

Usaremos el método de Runge-Kutta de 4 pasos (RK4) con ajuste y control de paso dado por la integral primera $f(x, f(x)) = 0$.

Es decir; dado un paso inicial h .

1. Calculamos $\bar{y}(h)$ por el método de Runge-Kutta de 4 pasos.
2. El punto $\bar{y}(h)$ debería cumplir que $f(x(h), \bar{y}(h)) = 0$. Como el cálculo ha sido aproximado, no cumple esta igualdad exactamente. Pero está mucho más cerca de una solución de la ecuación $f(x(h), y) = 0$ que de las otras, entonces escogemos esta solución más próxima como $y(h)$.

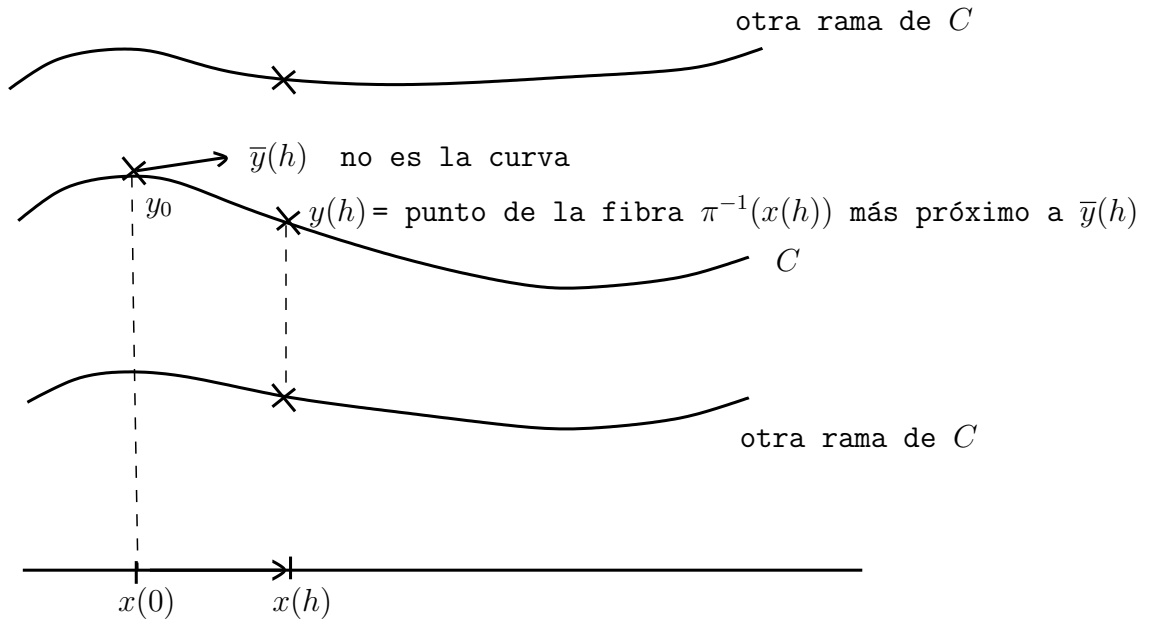


Figura 3.2:

3. Control de paso: pedimos siempre que el punto $y(h)$ este más próximo a $\bar{y}(h)$ que todos los demás puntos de la fibra $\pi^{-1}(h)$ por un factor prefijado. Usualmente pedimos $|\bar{y}(h) - y(h)| \leq 10^{-12} \cdot |\bar{y}(h) - y|$ para cualquier otro $y \in \pi^{-1}(x(h))$.

Si la distancia $|\bar{y}(h) - y(h)|$ se acerca a esta cota, acotamos el paso h .

Si la distancia $|\bar{y}(h) - y(h)|$ es menor que esta cota, alargamos el paso h .

La fórmula de control de paso es nuevo paso:

$$\tilde{h} = 0,75 \cdot h \cdot \left(\frac{\Delta_0}{\Delta}\right)^{1/5}$$

donde

$$\Delta_0 = 10^{-12} \cdot \min_{y \in \pi^{-1}(x(h))} |\bar{y}(h) - y|$$

$$\Delta = |\bar{y}(h) - y(h)|$$

Esta cota es eficiente porque no integramos la solución para una y aislada, sino para todas las y en $\pi^{-1}(x_0)$ a la vez.

La potencia $1/5$ en la cota se debe a que el método de RK4 tiene error $\sigma(h^5)$.

Integrando el PVI (3.2), podemos hacer dos cálculos:

1. Monodromía del recubrimiento $C \rightarrow \mathbb{C}$.

Cuando seguimos un camino cerrado en \mathbb{C} que evite los valores de ramificación, obtendremos la permutación que la monodromía introduce en la fibra $\pi^{-1}(x_0)$.

2. Monodromía de trenzas del morfismo

$$\begin{array}{ccc} \mathbb{C}^2 \setminus C & & (x, y) \\ \downarrow \tilde{\pi} & & \downarrow \\ \mathbb{C} & & x \end{array}$$

En $\mathbb{C} \setminus \text{Ramificación}$, $\tilde{\pi}$ es un morfismo topológicamente localmente trivial, con fibras $\mathbb{C} \setminus \{n \text{ puntos}\}$.

Por tanto este morfismo $\tilde{\pi}$ tiene monodromía determinada en

$$Diff^+(\mathbb{C} \setminus n \text{ puntos}) / Diff_0^+(\mathbb{C} \setminus n \text{ puntos})$$

Este es el grupo de Artin.

La monodromía se detecta estudiando los cruzamientos de los n agujeros, que son los puntos (x, y) en que la curva C corta a cada plano $x = constante$

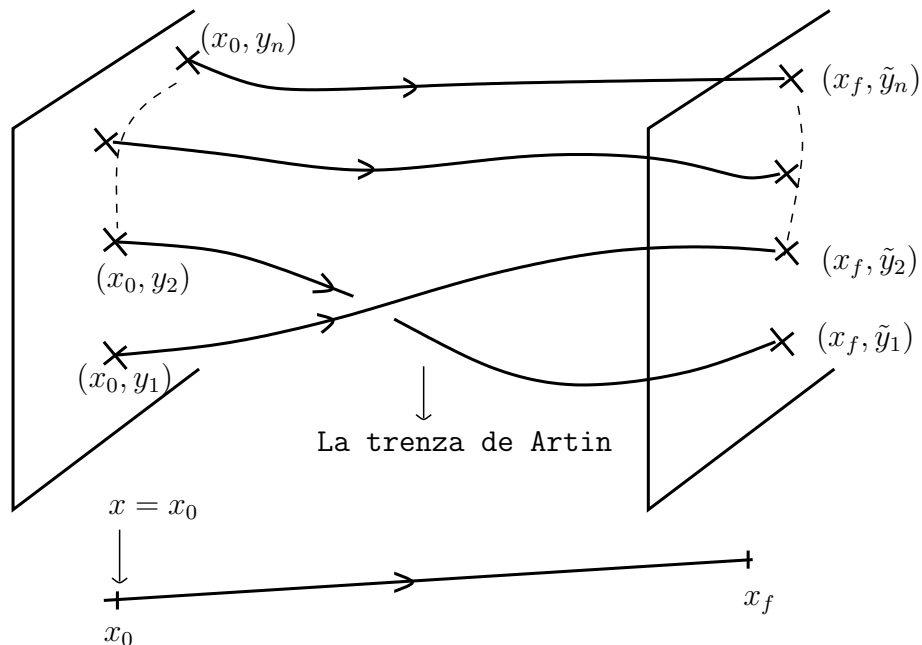


Figura 3.3: El transporte paralelo determina la trenza.

Caminos donde integrar el PVI (3.2):

Sea $R = \{\text{valores de ramificación de } \pi, \tilde{\pi}\}$ (calculados en el cálculo algebraico).

Las monodromías de la curva y de trenzas quedan determinadas por la monodromía sobre un sistema de generadores del grupo fundamental $\pi_1(\mathbb{C} \setminus R, x_0)$: una araña (veáse Capítulo 1, sección 1.2).

Clásicamente se usaban arañas de la forma:

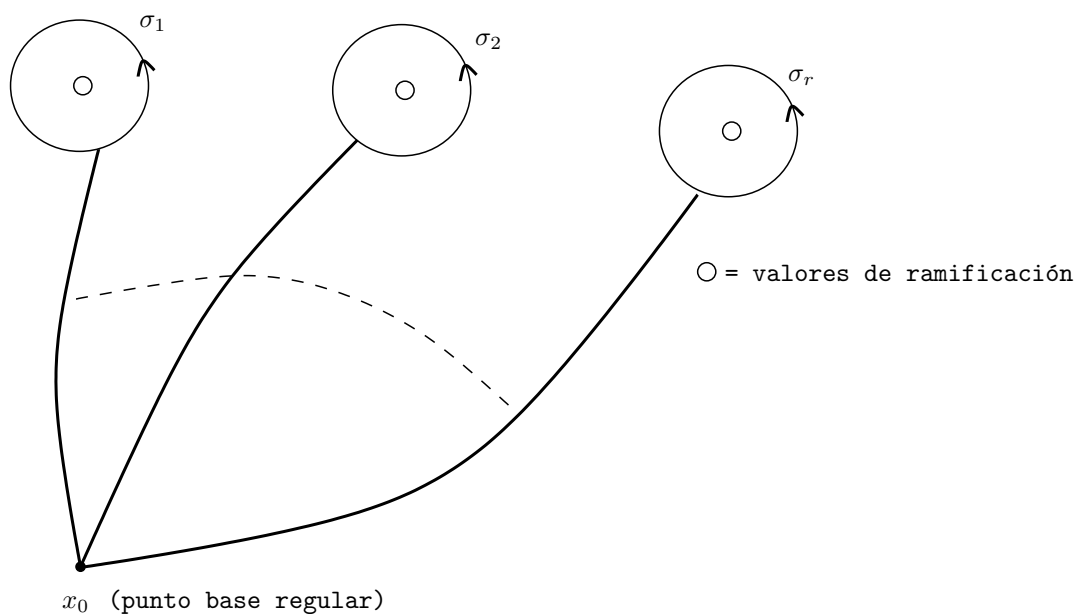


Figura 3.4: Araña.

Con los caminos $\gamma_1, \dots, \gamma_r$ ordenados para que el producto $\gamma_1, \dots, \gamma_r$ sea homotopo a rodear todos los valores de ramificación.

Problema:

Estos caminos $\gamma_1, \dots, \gamma_r$ se acercan mucho a los valores de ramificación, y esto es malo para la integración numérica: las soluciones de algunos de los puntos de C son más estables que las otras, y el integrador numérico salta de una solución a otra, que no es la que debería seguir.

Solución:

Tomar la descomposición celular de Voronoi para los valores de ramificación x_1, \dots, x_r .

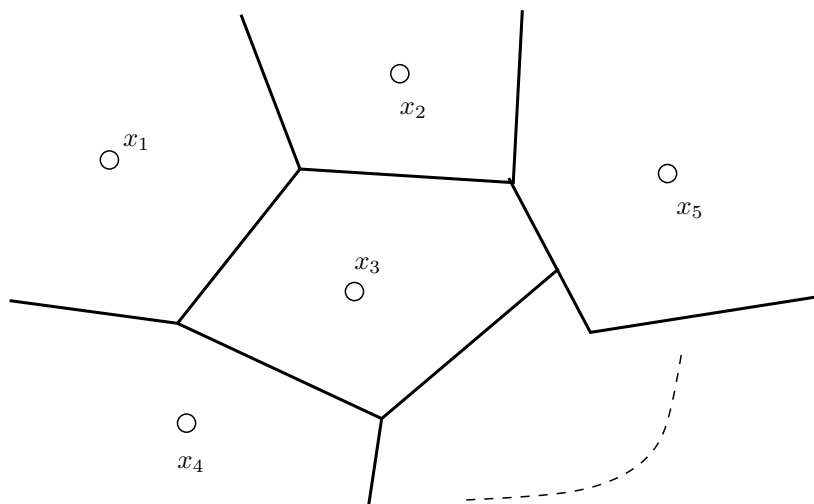


Figura 3.5: Celdas de Voronoi.

Cada celda de Voronoi es convexa, poligonal.

Podemos hacer que Matlab calcule estas celdas, que las cierre por fuera (pidiéndole diagrama de Voronoi para los valores de ramificación y más puntos ficticios).

Escogemos un vértice x_0 del grafo de Voronoi como puntos base y podemos hallar un sistema de generadores de $\pi_1(\mathbb{C} \setminus R, x_0)$ formado por caminos de aristas en el grafo de Voronoi.

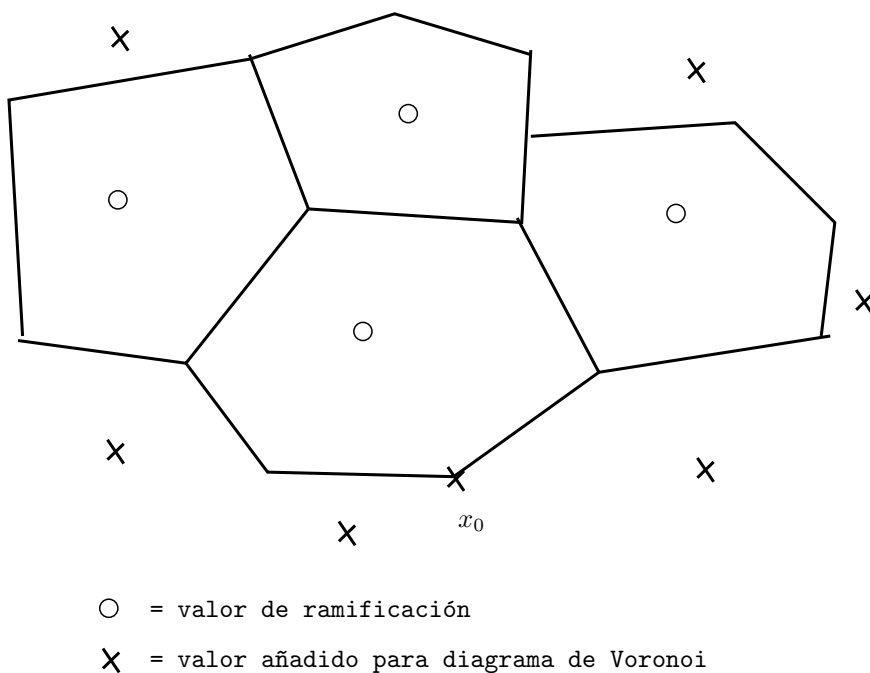


Figura 3.6: Las celdas de los valores de ramificación permiten definir la araña de Voronoi.

La araña de Voronoi se halla tomando un árbol maximal en el grafo de las celdas de Voronoi de los valores de ramificación, rodeándolo y usando como lazos el camino que va a cada celda por el árbol maximal, la rodea y regresa por donde ha venido.

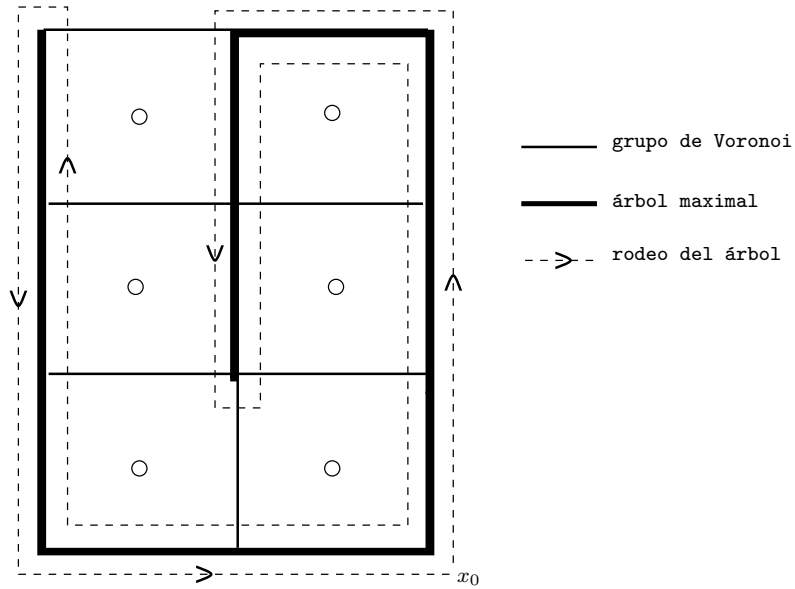


Figura 3.7: Grafo de las celdas de Voronoi.

En este ejemplo la araña de Voronoi está formada por los caminos.

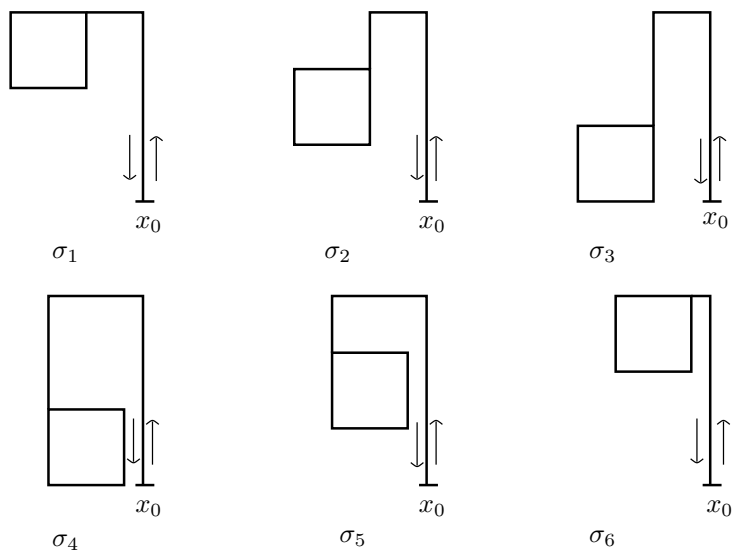


Figura 3.8: Caminos.

Estos caminos ordenados generan $\pi_1(\mathbb{C} \setminus R, x_0)$ y cumplen que $\gamma_1 \cdot \gamma_2 \cdot \gamma_3 \cdot \gamma_4 \cdot \gamma_5 \cdot \gamma_6 \cdot \dots \sim \gamma_\infty$, donde γ_∞ es:

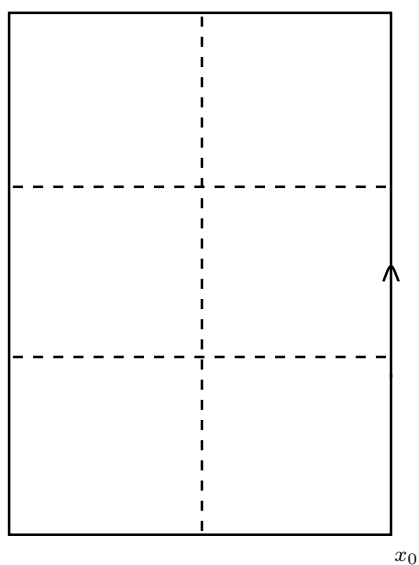


Figura 3.9: Camino que da la vuelta a todo.

3.4. Listado de los programas de cálculo de monodromías de trenzas en Matlab

En esta sección se muestran los programas utilizamos en Matlab. Esta librería para el cálculo de monodromías de trenzas con Matlab ha sido desarrollada hasta el presente por J. Amorós e I. Berna, y se halla inacabada por el rendimiento insuficiente del programa **transporte.m**.

Calculamos monodromías de trenzas para el divisor de ramificación. Hacemos este cálculo parametrizando diferenciablemente tramos del divisor. Usamos derivación implícita para convertir el problema en un sistema de ecuaciones diferenciales ordinarias a valores complejos, que integramos con un integrador numérico, implementado en Matlab.

Las monodromías de trenzas que queremos calcular, se calcularán usando la librería `_analítica`, en la cual se hallan las siguientes funciones:

- `cancela.m`: lleva a la forma reducida un camino de aristas en un poliedro cancelando todos los tramos donde el camino es constante y pares de aristas consecutivas que sean la misma recorrida en sentidos opuestos. Itera el procedimiento hasta que ya no quedan más pares, entonces el camino de aristas reduce.
- `diferencial.m`: calcula las derivadas de la diferencial f respecto de x, y como una matriz.
- `diferents.m`: recibe una lista de números complejos y devuelve otra que contiene todos los números de la original que eran diferentes dos a dos. Define diferente como estar a distancia más grande que todo.
- `edo_segment_vect.m`: Edo (ecuación diferencial ordinaria) a integrar para seguir la rama de curva algebraica sobre un segmento. Procedimiento documentado en `log_monodromia.txt`.
- `empalmar.m`: creación de un transporte paralelo engancho dos tramos de forma que la ordenación seguida al acabar el primero sea la usada al empezar el segundo. Transportes paralelos a empalmar y_1, y_2 dados como movimiento de un n -plano de puntos en el plano C , el movimiento de cada puntos descrito por una columna de la matriz. Devuelve `y_empalmado` en el mismo formato, resultado poner y_1 primero, y_2 después.
- `exemple_vectorial.m`: ejemplo con un cálculo vectorializado para integrar todos los pasos al mismo tiempo.
- `fibra.m`: calcula lista de valores de y tales que $f(x, y) = 0$ para el polinomio f definido en 2 variables. Después de encontrarlas con `roots` de Matlab hace mejorar iterativa para NR.

- `imatge_trena.m`: recibe la proyección de la trenza en una recta, parametrizada por `t_trenza`, y el fichero `estilfibres` que indica en los cruces qué tramo va por delante (oscuro, continuo) y qué tramo por última (claro, discontinuo). Dibuja la trenza en los colores especificados, siguiendo el convenio de que el tiempo va de arriba abajo. Guarda la trenza como un fichero `.png`.
- `ordenacio_arestes.m`: recibe los caminos de la base dados por los vértices por donde pasan, y la lista de aristas dadas por el vértice inicial, vértice final. Devuelve la lista de caminos expresados por número de aristas en lugar de vértice.
- `ordre_arestes.m`: para cada vértice de un grafo plano, ordena ciclicamente las aristas por él.
- `ordre_celles.m`: para cada vértice de un grafo plano, ordena ciclicamente las aristas por él.
- `param2paraula.m`: reconoce la trenza descrita por un camino de un n -plano de puntos parametrizado y (cada columna=posiciones sucesivas en C de un puntos del n -plano), a partir del sistema de generadores standard del grupo de mira ortogonalmente a esta recta desde y_{∞} .
- `poli2val.m`: evalúa un polinomio en 2 variables dado como matriz de coeficientes (potencias de x en columna por la derecha, potencias de y en fila por la izquierda, en los dos casos en orden de potencia decreciendo).
- `reduccio.m`: reducción cíclica de palabras en grupo libre no conmutativo. Palabras codificadas indicando los generadores en letras minúsculas y los suyos inversos con la mayúscula correspondiente. El elemento neutro se vuelve como una cadena vacía.
- `representa_trena.m`: script de Matlab para dibujar trenzas particulares calculadas por `ejemplo_familia`. Funciona ejecutándose a Matlab justo después de la ejecución de `ejemplo_familia`.
- `rk4intlavect.m`: integrador numérico para edos complejas de orden 1 proviniendo de seguir curvas $f(x, y) = 0$ sobre caminos de tiempos rectilíneos en el plan complejo. Usa RK4 y ajuste del paso Δt a la integral primera dada por la propia f .
- `transport.m`: Recibe lista de vértices y de aristas entre ellos. Calcula la fibra de cada vértice. Integra vectorialmente el edo sobre cada arista, de la fibra de partida a la de llegada. Devuelve mesas de valores del tiempo t y de los vectores y por cada arista.

- `trenador.m`: cálculo de las trenzas y permutaciones definidas por el transporte paralelo de un recubrimiento ramificado de C a lo largo de un sistema de caminos cerrados.
- `video.trena.m`: recibe la mesa de valores complejas de y a lo largo de la trenza y la convierte en un video de duración preestipulada (s) que enseña por pantalla y guarda como `nombre.abuelo`.

3.4.1. Listado programas Matlab

Listado programa cancela.m

```
function vf=cancela(v0)
% Programa per portar a forma reduida un cami d'arestes en
% poliedre, cancel.lant tots els trams on el cami es constant
% i parells d'arestes consecutives que siguin la mateixa
% recorreguda en sentits oposats. Itera el procediment fins
% que ja noqueden mes d'aquests parells, aleshores el cami
% d'arestes es reduit.
% INPUT: v0=camí en el poliedre (llista de vertexs que recorre,
% en fila). OUTPUT: vf=camí reduit (llista de vertexs que
% recorre, en fila).
% Jaume Amoros, UPC, Barcelona 2011/8/2
% Provat i funciona.

% Detecta trams on el cami es constant i el redueix a un vertex
diffs=[0,v0]-[v0,0];
reps=find(diffs==0);
while length(reps)>0
    v0=[v0(1:reps-1),v0(reps+1:end)];
    diffs=[0,v0]-[v0,0];
    reps=find(diffs==0);
end;

% Detecta parelles d'arestes consecutives oposades i les cancela
vf=v0;
diffs=[0,0,vf]-[vf,0,0];
cancs=find(diffs==0,1)-2;

while (length(cancs)>0)&&(length(vf)>1)
    vf=[vf(1:cancs(1)),vf(cancs(1)+3:end)];
    diffs=[0,0,vf]-[vf,0,0];
    cancs=find(diffs==0,1)-2;
end;
```

Listado programa diferencial.m

```
function [dfdx,dfdy]=diferencial(f)
% Calcula les derivades de f respecte de x,y com a matrius.
% Jaume Amoros, UPC, Barcelona
% 2009/5/28
% Provat i funciona.

global degx degy

for k=1:degy,
    dfdy(k,:)=(degy-k+1)*f(k,:);
end;
for k=1:degx,
    dfdx(:,k)=(degx-k+1)*f(:,k);
end;
```

Listado programa diferents.m

```
function v=diferents(u,tol)
% Rep una llista de nombres complexos
% i en retorna una altra que conte tots els nombres de la
% original que erendiferents dos a dos. Defineix diferents
% com estar a distancia mes gran que tol.
%
% Jaume Amoros, UPC, Barcelona
% 2010/7/26
% Provat i funciona.

if nargin==1
    tol=1e-12;
end;

n=length(u);
v=u(1);

for j=2:n,
    altres=v-u(j);
    if min(abs(altres))>tol
        v=[v,u(j)];
    end;
end;

if size(u,1)>1
    v=v';
end;
```

Listado programa edo_segment_vect.m

```
function z=edo_segment_vect(t,y)
% Edo a integrar per seguir branca de corba algebraica sobre
% un segment. Procediment documentat a log_monodromia.txt
% Nova versio vectorial (sistema: y es una fila amb diversos
% VI de l'edo)
% Jaume Amoros, UPC, Barcelona 2011/8/1
% En desenvolupament.

global dfdx dfdy degx degy
global x0 xf

x=(1-t).*x0+t.*xf;
prodx=1;
vecx=1;
for k=1:degx,
    prodx=prodx.*x;
    vecx=[prodx;vecx];
end;
prody=ones(degy,1);
vecy=prody;
for k=1:degy,
    prody=prody.*conj(y');
    vecy=[prody,vecy];
end;
z=-(vecy*dfdx*vecx(2:end))./(vecy(:,2:end)*dfdy*vecx).*(xf-x0);
z=conj(z');
```

Listado programa empalmar.m

```
function y_empalmada=empalmar(y1,y2)
% Creacio d'un transport paral.lel enganxant-ne dos trams
% de manera que l'ordenacio seguida en acabar el primer
% sigui la usada en comenc,ar el segon. Transports paral.lels
% a empalmar y1,y2 donats com a moviment d'unan-pla de punts
% en el pla C, moviment de cada punt descrit per una columna
% de la matriu. Retorna y_empalmada en el mateix format,
% resultat de comosar y1 primer, y2 despres.
%
% Jaume Amoros, UPC, Barcelona
% 2010/9/28
%
% Provat i funciona.

for k=1:size(y1,2),
    [err,index]=min(abs(y2(1,:)-y1(end,k)));
    empalme(k)=index;
end;

y_empalmada=[y1;y2(:,empalme)];
```

Listado programa exemple_vectorial.m

```
% exemple amb un calcul vectorialitzat, per a integrar
% tots els passos al mateix temps
%
% Jaume Amoros, UPC, Barcelona
% 2011/8/4
%
% en desenvolupament

global degx degy dfdx dfdy f x0 xf

% NOM DE LA FAMILIA AMB QUE CALCULAREM
nom='seidel_3';

% lectura del polinomi (fpreglobal=polinomi com a matriu)
eval(['matriu_divisor_' nom]);
fpreglobal=freal+i*fimag;

f=fpreglobal;

% lectura de la ramificacio
nomramif=['llista_ramificacio_' nom '.txt'];
fid=fopen(nomramif,'r');
h1=fscanf(fid,'%c');
fclose(fid);
ch=['singulars0=[' h1 '];'];
eval(ch);

% separa valors de ramificacio diferents
tol=1e-6;
singulars=diferents(singulars0,tol)';

n=length(singulars);

degx=size(f,2)-1;
degy=size(f,1)-1;
[dfdx,dfdy]=diferencial(f);

'Aranya'
% buscar l'aranya de voronoi
[base,v,arestes,vbase]=aranya_voronoi(nom,real(singulars),
imag(singulars));

base_arestes=ordenacio_arestes(base,arestes);
```



```

'Transport'
% calcula fibres dels vertexs i transports sobre les arestes
[yv,taula_t,taula_y]=transport(v,arestes);

% taula de trenes i de permutacions
% Fixem la recta de projeccio per determinar ordre de la
fibra i trenes
obase=-2-1.5*i;
ptbase=-1.2*i;

op=ptbase-obase;
oq=yv(:,4)-obase;
lambda=real(conj(op)*oq)/abs(op)^2;
projq=obase+lambda*op;

% i les distancies orientades a la recta
nu=i*op/abs(op);
mu=real((oq-lambda*op)/nu);

% reordenacio dels brins de la trena segons la projeccio
% escollida
[l1,ordre_base]=sort(lambda);

'Trenador'
% calcul de la trena de cada cami de la base
[taula_trenes,taula_permutacions,parametritzacions]=
trenador(base_arestes,taula_t,taula_y,obase,ptbase);

% COMPROVACIO: MONODROMIA A L'INFINIT
trena_infinit=[];
permutacio_infinit=linspace(1,degy,degy);

for k=1:length(taula_trenes),
    trena_infinit=[trena_infinit,taula_trenes{k}];
    permutacio_infinit=permutacio_infinit(taula_permutacions(k,:));
end;
trena_infinit=reduccio(trena_infinit);

```

Listado fibra.m

```
function llista_y=fibra(x)
% Calcula llista de valors de y tals que f(x,y)=0
% per f polinomi en 2 variables.
% Despres de trobar-les amb roots de matlab fa millora
% iterativa per NR
% Jaume Amoros, UPC, Barcelona
% 2011/8/4
% En desenvolupament.
global f degx

% precissio amb que volem la fibra
tol=1e-10;

% polinomi d'una variable amb x donat, i les seves arrels
% segons roots
potx=1;
vecx=1;
for k=1:degx,
    potx=potx*x;
    vecx=[potx;vecx];
end;
fx=f*vecx;
llista_0=roots(fx);
fxp=polyder(fx);

llista_y=llista_0;

% millora de la precissio de les arrels amb NR
% llista_0=llista_0-7*tol;
% while max(abs(llista_y-llista_0))>tol
% llista_0=llista_y;
% llista_y=llista_0-polyval(fx,llista_0)./polyval(fxp,llista_0);
% end;
```

Listado imatge_trena.m

```
function nom_fitxer=imatge_trena(nom,t_trena,lambda,creuaments)
% Rep la projeccio de la trena en una recta, parametrizada per
% t_trena, i el fitxer estilfibres que indica en els creuaments
% quin tram va per davant (fosc, continu) i quin per darrera
% (clar, discontinu). Dibuixa la trena en els colors especificats,
% seguint el conveni de que el temps va de dalt a baix.
% Guarda la trena com un fitxer png.
% Jaume Amoros, UPC, Barcelona
% 2010/10/14
%
% Provat i funciona.

estils='b-c: ';
fils=size(lambda,2);
nom_fitxer=['imatge_' nom '.png'];
nc=size(creuaments,1);
if nc==0
    [nom ' Trena trivial: no hi han creuaments. ']
    return;
end;
t_trena=t_trena/t_trena(end);

% semitemps que s'oculta la fibra de darrera en els creuaments
tcreu=t_trena(creuaments(:,1));
tcreu=[0;tcreu;1];
diff_t=diff(tcreu);
marge=0.45*min(diff_t);

% taula de colors (estils) del dibuix
% 1=color del front, 2=color del fons
estilfibres=ones(size(lambda));
ordrefibres=linspace(1,fils,fils);
for k=1:nc,
    ind0=creuaments(k,1);
    while ind0>0 && (tcreu(k)-t_trena(ind0)<marge)
        estilfibres(ind0,ordrefibres(creuaments(k,2)+1-creuaments(k,3)))=2;
        ind0=ind0-1;
    end;
    indf=creuaments(k,1);
    while indf<=size(estilfibres,1) && (t_trena(indf)-tcreu(k)<marge)
        estilfibres(indf,ordrefibres(creuaments(k,2)+1-creuaments(k,3)))=2;
        indf=indf+1;
    end;
end;
```

```
transit=ordrefibres(creuaments(k,2));
ordrefibres(creuaments(k,2))=ordrefibres(creuaments(k,2)+1);
ordrefibres(creuaments(k,2)+1)=transit;
end;

% dimensions del dibuix
lmin=min(min(lambda));
lmax=max(max(lambda));
lcentre=(lmin+lmax)/2;
```

Listado ordenacio_arestes.m

```
function base_arestes=ordenacio_arestes(base,arestes)
% Rep els camins de la base donats pels vertexs per on
% passen, i la llista d'arestes donades per vertex inicial,
% vertex final. Retorna la llista de camins expressats
% per nombre d'aresta enlloc de per vertex.
% Jaume Amoros, UPC, Barcelona
% 2011/8/4
%
% Probat i funciona.

na=size(arestes,1);
nc=length(base);

% Matriu de les arestes
G=sparse(1,1);
for k=1:na,
    G(arestes(k,1),arestes(k,2))=k;
    G(arestes(k,2),arestes(k,1))=0;
end;

% Traduccio dels camins
for k=1:nc,
    camiv=base{k};
    lc=length(camiv);
    camia=[];
    for l=1:lc-1,
        if G(camiv(l),camiv(l+1)) ~=0
            camia(l)=G(camiv(l),camiv(l+1));
        else
            camia(l)=-G(camiv(l+1),camiv(l));
        end;
    end;
    base_arestes{k}=camia;
end;
```

Listado ordre_arestes.m

```
function adjcicl=ordre_arestes(G,v)
% Per cada vertex d'un graf pla, ordena ciclicament les
% arestes per ell. INPUT: G=matriu d'adjacencies del graf,
% v=llista de vertexs del graf (coord. en fila)
% OUTPUT: adjcicl=estructura que te una cel.la per cada
% vertex, amb els seus vertexs adjacents ordenats segons
% argument de -pi a pi
% Jaume Amoros, UPC, Barcelona
% 2010/11/19
%
% Provat i funciona.

nv=size(G,1);

for k=1:nv,
    % vertexs adjacents
    adj=[find(G(k,:)),find(G(:,k))'];
    % vectors aresta adjacent, i els ordena segons argument
    aradj=v(adj,:)-ones(length(adj),1)*v(k,:);
    argadj=atan2(aradj(:,2),aradj(:,1));
    [argord,adjord]=sort(argadj);
    adjcicl{k}=adj(adjord);
end;
```

Listado ordre_celles.m

```
function cvord=ordre_celles(c,v)
% Ordena ciclicament les cel·les de Voronoi per cada vertex
% de la descomposicio, descartant el vertex de l'infini i
% les cel·les per ell.
% INPUT: c=estructura de cel·les de Voronoi (tal com queda
% despres de voronoin i eliminar el punt i les cel·les per
% l'infini de la llista), v=llista de vertexs.
% OUTPUT: cvord=estructura que per cada vertex te matriu
% amb la fila el vector de cel·les per ell ordenat
% ciclicament, i 2a fila els respectius arguments dels
% vectors bissecant la cel·la en el vertex (ordenades segons
% angle bissecant de -pi a pi)
% Jaume Amoros, UPC, Barcelona
% 2010/11/19
% Modificat 2011/8/3
%
% Provat i funciona.
n=length(c);
nv=size(v,1);
% Matriu que a fila k-essima te 1 en les posicions dels
% vertexs de cel·la k-essima
gd=zeros(n,nv);
for k=1:n,
    vertexs=c{k};
    gd(k,vertexs)=ones(1,length(vertexs));
end;
% llegim la matriu de vores per columnes, i fem la ordenacio
for k=1:nv,
    cels=find(gd(:,k))';
    argcels=[];
    for l=1:length(cels),
        % ubica el vertex k en la cel·la, i busca els veïns
        vcela=c{cels(l)};
        [zm,pv]=min(abs(vcela-k));
        vcela=[vcela(end),vcela,vcela(1)];
        vei1=vcela(pv);
        vei2=vcela(pv+2);
        vec1=v(vei1,:)-v(k,:);
        vec1=vec1/norm(vec1);
        vec2=v(vei2,:)-v(k,:);
        vec2=vec2/norm(vec2);
        veccela=vec1+vec2;
        argcels(l)=atan2(veccela(2),veccela(1));
    end;
    [argsord,celsord]=sort(argcels);
    cvord{k}=[cels(celsord);argsord];
end;
```

Listado param2paraula.m

```
function [nou_ordre,trena,creuaments,lambda,mu]=
param2paraula(y,obase,ptbase)
% funcio per a reconeixer la trena descrita per un cami
% d'una n-pla de punts parametrizats y (cada columna=posicions
% successives en C d'un punt de la n-pla), a partir del
% sistema de generadors standard del grup de trenes donat
% per la projeccio ortogonal a la recta orientada
% [obase,ptbase]. La trena es determina segons la veu un
% observador que mira ortogonalment a aquesta recta des de
% y=-infinit.
% Jaume Amoros, UPC, Barcelona 2010/9/28
% Provat i funciona.

alfabet='abcdefghijklmnopABCDEFGHIJKLMNPO';
Capshift=16; % longitud de l'alfabet emprat

% calcula matricialment les projeccions dels punts de y en
% la recta [obase, ptbase]
op=ptbase-obase;
oq=y-obase;
lambda=real(conj(op)*oq)/abs(op)^2;
projq=obase+lambda*op;

% i les distancies orientades a la recta
nu=i*op/abs(op);
mu=real((oq-lambda*op)/nu);

% reordenacio dels brins de la trena segons la
% projeccio escollida
[l1,nou_ordre]=sort(lambda(1,:));
lambda=lambda(:,nou_ordre);
mu=mu(:,nou_ordre);

% determinacio de la trena com a paraula en els generadors,
% i dels punts de creuament de fibres

trena=[];
creuaments=[];
[valor,indvell]=sort(lambda(1,:));
for k=2:size(y,1),
    [valor,ind]=sort(lambda(k,:));
    variacio=ind-indvell;
    if max(variacio)>0.1
        if sum(sign(abs(variacio)))>2
            ['Error: Creuament amb mes de dos variacions a ' num2str(k-1)]
        end;
    end;
end;
```



```

    [ax,bx]=max(variacio);
    [an,bn]=min(variacio);
if (abs(bx-bn) ~= 1)
['Error: canvi de posicio en elements no consecutius a ' num2str(k-1)]
end;
    grador=min([bx,bn]);
if mu(k,indvell(grador))<mu(k,indvell(grador+1))
    trena=[trena alfabet(grador)];
    davant=0;
else
    trena=[trena alfabet(grador+Capshift)];
    davant=1;
end;
    creuaments=[creuaments;k-1,grador,davant];
    indvell=ind;
end;
end;

```

Listado poli2val.m

```
function z=poli2val(pol2,x,y)
% avalua un polinomi en 2 variables donat com a matriu de
% coeficients (potencies de x en columna per la dreta,
% potencies de y en fila per l'esquerra, en tots dos casos
% en ordre de potencia decreixent).
% Jaume Amoros, UPC, Barcelona
% 2010/9/21
% Provat i funciona.
degpx=size(pol2,2)-1;
degpy=size(pol2,1)-1;
prodx=1;
vecx=1;
for k=1:degpx,
    prodx=prodx.*x;
    vecx=[prodx;vecx];
end;
prody=1;
vecy=1;
for k=1:degpy,
    prody=prody.*y;
    vecy=[prody,vecy];
end;

z=vecy*pol2*vecx;
```

Listado reduccio.m

```
function pred=reduccio(paraula)
% Reduccio ciclica de paraules en grup lliure no commutatiu.
% Paraules codificades indicant els generadors en lletres
% minuscules i els seus inversos amb la majuscula
% corresponent. L'element neutre es torna com la cadena
% buida ''.
% Jaume Amoros, UPC, Barcelona
% 2010/10/11
%
% Provat i funciona.

pred=paraula;
llargvell=length(paraula)+1;
llarg=length(pred);

while llarg<llargvell
    difs=abs(pred(2:end)-pred(1:end-1));

    ind=1;
    indred=[];

    while ind<llarg
        if difs(ind)==32
            ind=ind+2;
        else
            indred=[indred,ind];
            ind=ind+1;
        end;
    end;

    if ind==llarg
        indred=[indred,ind];
    end;

    pred=pred(indred);
    llargvell=llarg;
    llarg=length(pred);
end;
```

Listado representa_trena.m

```
% representa_trena.m
% Script de Matlab per dibuixar trenes particulars calculades
% per exemple_{familia}
% Funciona executant-se a Matlab just despres de l'execucio
% de exemple_{familia}

% provat i funciona

% tria d'una trena
ntrena=18;

% la resta funciona sola
nom_trena=[nom '_trena' num2str(ntrena)];
t_trena=parametritzacions{ntrena}{1};
y_trena=parametritzacions{ntrena}{2};
lambda_trena=parametritzacions{ntrena}{3};
creuaments_trena=parametritzacions{ntrena}{4};

% imatge de la trena
nom_fitxer_trena=imatge_trena(nom_trena,t_trena,lambda_trena,
creuaments_trena);

% video de la trena
nom_video_trena=video_trena(y_trena,t_trena,nom_trena,30);
```

Listado rk4int1avect.m

```
function [t,y]=rk4int1avect(g,temps,y0,marge)
% Integrador numeric per a edos complexes d'ordre 1 provenint
% de seguir corbes f(x,y)=0 sobre camins de temps rectilinis
% en el pla complex. Usa RK4 i ajust del pas via la integral
% primera donada per la propia f. Versio 'sistema d'edos':
% y0 es un vector fila de valors, i els integra tots alhora
% Jaume Amoros, UPC, Barcelona (Provat i funciona).
global degy x0 xf % NOMES PER INTEGRAR SEGUIMENT DE CORBA
f(x,y)=0
t=temps(1);
y=y0;
h=(temps(2)-temps(1))/100;
if nargin==3
    marge=0.01;
end;
while t(end)<temps(2)
% un pas de RK4
    k1=h*g(t(end),y(end,:));
    k2=h*g(t(end)+h/2,y(end,:)+k1/2);
    k3=h*g(t(end)+h/2,y(end,:)+k2/2);
    k4=h*g(t(end)+h,y(end,:)+k3);
    y1=y(end,:)+(k1+2*k2+2*k3+k4)/6;
% busca la fibra d'on hem caigut, calcula l'aparellament de
% punts i l'error
    yfibra=fibra((1-t(end)-h)*x0+(t(end)+h)*xf);
% NOMES PER INTEGRAR SEGUIMENT DE CORBA f(x,y)=0
    yfc=yfibra*ones(1,degy);
    yff=ones(degy,1)*conj(yfibra');
    yfdists=abs(yfc-yff);
    yfdists=yfdists+(yfdists(1,2)+1)*eye(degy);
    delta0=marge*min(yfdists);
% aparella elements del pas y1 amb fibra de desti, i comprova
% error
    y1f=ones(degy,1)*y1;
    yerr=abs(y1f-yfc);
    [delta,yex]=min(yerr);
% calcula l'error optim, si estem per sota accepta el pas i
% si estem
% per sobre, no
    if min(delta0-delta)>0
        t=[t;t(end)+h];
        y=[y;conj(yfibra(yex)')];
    end;
% reescala el pas per a buscar error optim en el seguent pas
    h=min([0.75*h*(min(delta0./delta))^0.2,temps(2)-t(end)]);
end;
```

Listado transport.m

```
function [yv,taula_t,taula_y]=transport(v,arestes)
% Rep llista de vertexs i d'arestes entre ells.
% Calcula la fibra de cada vertex.
% Integra vectorialment l'edo sobre cada aresta,
% de la fibra de partida a la d'arribada.
% Retorna taules de valors del temps t i dels vectors y
% per cada aresta.
% Jaume Amoros, UPC, Barcelona
% 2011/8/4
%
% Probat i funciona.

global degx degy dfdx dfdy f x0 xf

% fibres dels vertexs
for k=1:size(v,1),
    yv(:,k)=fibra(v(k,1)+i*v(k,2));
end;

% transport sobre les arestes
marge=1e-12;
% marge d'error tolerat en l'ajust a la fibra al final
% del pas

for k=1:size(arestes,1),
    x0=v(arestes(k,1),1)+i*v(arestes(k,1),2);
    xf=v(arestes(k,2),1)+i*v(arestes(k,2),2);
    y0=conj(yv(:,arestes(k,1)))';
    [t,y]=rk4int1avect(@edo_segment_vect,[0,1],y0,marge);
    taula_t{k}=t;
    taula_y{k}=y;
    [k,length(t)]
end;
```

Listado trenador.m

```
function [taula_trenes,taula_permutacions,parametritzacions]=
trenador(base_arestes,taula_t,taula_y,obase,ptbase)
% Calcul de les trenes i permutacions definides pel
% transport paral.lel d'un recobriment ramificat de C al
% llarg d'un sistema de camins tancats.
% INPUT: base_arestes=sistema de camins tancats, yv=fibra
% dels vertexs, taula_y=n-pla de valors de y en cada aresta,
% obase,ptbase=2 punts que defineixen la recta de projeccio
% per donar les trenes.
% OUTPUT: taula_trenes=llista de paraules de trena de cada
% camí tancat, taula_permutacions=llista de permutacions de
% monodromia sobre cada camí tancat(indicades en vector [1,2,,n])
% Jaume Amoros, UPC, Barcelona 2011/8/4 (Provat i funciona).
nc=length(base_arestes);
for k=1:nc,
    % taula de valors de y en tot el camí poligonal
    y=[];
    lc=length(base_arestes{k});
    if base_arestes{k}(1)>0
        y=taula_y{base_arestes{k}(1)};
        t_trena=taula_t{base_arestes{k}(1)};
    else
        y=flipud(taula_y{-base_arestes{k}(1)});
        t_trena=1-flipud(taula_t{-base_arestes{k}(1)});
    end;
    for l=2:lc,
        if base_arestes{k}(l)>0
            y=empalmar(y,taula_y{base_arestes{k}(l)});
            t_trena=[t_trena;l-1+taula_t{base_arestes{k}(l)}];
        else
            y=empalmar(y,flipud(taula_y{-base_arestes{k}(l)}));
            t_trena=[t_trena;l-1-flipud(taula_t{-base_arestes{k}(l)})];
        end;
    end;
    % s'envia a calcular la trena
    [nou_ordre,trena,creuaments,lambda,mu]=param2paraula(y,obase,ptbase);
    taula_trenes{k}=trena;
    % ara calculem la permutacio induida pel transport en la fibra base
    n=size(y,2);
    y0=conj(y(1,:))'*ones(1,n);
    yf=ones(n,1)*y(end,:);
    y0=y0(nou_ordre,:);
    yf=yf(:,nou_ordre);
    diffsy=abs(y0-yf);
    [yerr,yex]=min(diffsy);
    taula_permutacions(k,:)=yex;
    parametritzacions{k}={t_trena,y,lambda*abs(ptbase-obase),creuaments};
    k
end;
```

Listado video_trena.m

```
function nom_video=video_trena(y_trena,t_trena,nom,duracio)
% Rep la taula de valors complexes de y al llarg de la trena
% i la converteix en un video de duracio preestipulada (s)
% que ensenya per pantalla i guarda com a nom.avi.
% AVIS: en versions antigues de Matlab (fins la 7.5 en
% Linux) el video pot tenir poca compressio i un tamany
% descomunal.
% Jaume Amoros, UPC, Barcelona 2010/10/14 (Provat i funciona).
% duracio aproximada del video (s) per defecte
if nargin==3
    duracio=30;
end;
us=10; % usarem un de cada us fotogrames
% nom i velocitat del video, dimensions de la finestra
nom_video=['video_' nom '.avi'];
nt=size(y_trena,1);
fps=ceil(nt/(us*duracio));
t_trena=t_trena/t_trena(end);

remin=min(min(real(y_trena)));
remax=max(max(real(y_trena)));
immin=min(min(imag(y_trena)));
immax=max(max(imag(y_trena)));
centre=[(remin+remax)/2,(immin+immax)/2];
xmin=centre(1)-1.2*(centre(1)-remin);
xmax=centre(1)+1.4*(remax-centre(1)); % deixem espai pel rellotge
ymin=centre(2)-1.2*(centre(2)-immin);
ymax=centre(2)+1.2*(immax-centre(2));
% dibuixa en pantalla la figura i guarda els fotogrames
h=figure;
ind=1;
for k=1:us:length(t_trena),
    plot([xmin,xmax],[ymin,ymax],'ow');
    hold on
    plot(real(y_trena(1,:)),imag(y_trena(1:')), 'ro');
    text(remax,immax,num2str(t_trena(k)), 'FontSize',16);
    plot(real(y_trena(k,:)),imag(y_trena(k:')), 'k*');
    hold off
    Trena(ind)=getframe(figure(h));
    ind=ind+1;
end;

for l=ind:round(1.2*ind),
    Trena(l)=getframe(figure(h));
end;
```


log_monodromia.txt

CALCUL DE MONODROMIA DE CORBES ALGEBRAIQUES COMPLEXES PLANES

Corba plana afi \leftrightarrow polinomi en dos variables $f(x,y)$
(coeficients complexes)

Ojectius:

1. Calcular la monodromia de la corba $C = \{ f(x,y)=0 \}$ posada com a recobriment finit de la recta afi complexa via la projeccio $(x,y) \mapsto x$

Per a això es detectaran/importaran de Singular els valors singulars (on hi ha ramificació) i s'integrarà l'EDO

\$\$

$$\frac{dy}{dt} = - \frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}$$
$$\frac{dx}{dt}$$

\$\$

sobre camins $x(t)$ de valors regulars de dos tipus:

- 1) segments de x_0 a x_f , parametrizats per t de 0 a 1
- 2) cercles amb centre singular, radi r_{min} , punt inicial p_0 , parametrizats de forma $x(t) = \text{centre} + r_{min} \cdot \exp(2\pi i t)$, t de 0 a 1.

Tots els càlculs són al pla afi, assumim que la recta de l'infinit talla a la corba sense ramificació.

2. Calcular la monodromia en el grup de trenes del complementari de la corba $C = \{ f(x,y)=0 \}$, posat com a família de rectes afins punxades $\mathbb{C} \setminus C$ via la projecció $(x,y) \mapsto y$, sobre els valors regulars de f .

FORMAT DE LES DADES:

Els polinomis en 2 variables x,y es guarden com a matrius f tals que

$$[y^m, y^{(m-1)}, \dots, y, 1] * f * [x^n; x^{(n-1)}; \dots; x; 1] = f(x,y)$$

(per coherència amb el sistema que té Matlab per a operar amb polinomis d'una variable; també coherent amb conv2?)

Cada monodromia en el grup simètric de d elements es guarda com un vector fila de longitud d , que té en posició k -èsima la imatge de k per la permutació de monodromia.

Cada monodromia de trenes es guarda com a paraula en els generadors standard d'Artin i els seus inversos: els generadors standard $\sigma_1, \sigma_2, \dots$ es guarden com a, b, \dots i els inversos $\sigma_1^{-1}, \sigma_2^{-1}, \dots$ es guarden amb la lletra majúscula: A, B, \dots

Variables globals:

f el polinomi
degx degy graus de f en x i y
dfdx dfdy les derivades parcials (conveni: dfdx te
obligatoriament mida (degx+1) x degx, i dfdy te mida degy x
(degx+1). Omplir amb zeros si cal).
x0 xf extrems de segment sobre el que s'integra l'edo

Variables amb resultats del calcul:

singulars : llista de valors singulars (punts de ramificacio
de la projeccio) obase, ptbase : recta sobre la que projectem
per presentar les trenes de front v : matriu dels vertexs del
graf de Voronoi (coord reals x,y) arestes : llista d'arestes
del graf de Voronoi (donades per vertex inicial,vertex final
numerats segons v)
base : sistema de camins tancats que forma la base de Voronoi,
donats pels vertexs per on passa el cami (base{k}=llista de
vertexs que defineixen el cami poligonal k-essim)
base_arestes : el sistema de camins anterior, pero donat
numerant les arestes que el formen segons la llista a arestes
(signe menys indica que la recorrem en sentit contrari)
taula_trenes : estructura que te per cel.les les trenes de
les monodromies dels camins de base (taula_trenes{1} es la
paraula de la 1a trena,...) taula_permutacions : matriu que
te en cada fila k-essima la permutacio induida per la monodromia
sobre el cami base{k} parametritzacions : estructura amb dades
auxiliars. Per cada monodromia k-essima:
parametritzacions{k}{1}=taula de temps t_trena en que el calcul
recorre la trena parametritzacions{k}{2}=taula de posicions
y_trena dels punts de la trena en el pla complex C
(cada fila de y_trena es la posicio dels punts de la trena en
el seu temps de t_trena) parametritzacions{k}{3}=taula de
posicions lambda_trena de la projeccio ortogonal dels punts
de la trena en la recta base obase,ptbase (per dibuixar-la
frontalment) parametritzacions{k}{4}=taula de creuaments de
la trena, que te una fila per cada creuament de la trena
indicant: temps de creuament (posicio en t_trena), el fil a
l'esquerra dels 2 veins que es creuen, i 0 si el fil de
l'esquerra passa per davant, 1 si el fil de l'esquerra passa
per darrera. taula_permutacions : matriu que te per files
les permutacions dels elements de la fibra que cada monodromia
indueix (la monodromia en el grup simetric de la corba f=0)

Capítulo 4

Ejemplos

En este capítulo aplicamos los programas referidos en el capítulo 3 y el teorema descrito del capítulo 2, para hallar la monodromía geométrica y el grupo fundamental de las cuatro familias de curvas que hemos elegido.

Analizaremos dos ejemplos de tipo académico, otro que tiene interés en mirror symmetry como es la familia Seidel, y un último que presenta unas simetrías únicas, la familia de Bring.

4.1. Ejemplos Académicos

4.1.1. Familia Simple 1

Sea la cuádrica Q_s definida por la ecuación $x^2 + y^2 - z^2 - t^2 = 0$, y la cúbica S_s definida por la ecuación $x^3 + y^3 + z^3 + xyz + st^3 = 0$, la Familia 1 que vamos analizar.

Ecuación del divisor de ramificación de la Familia 1:

$$(54r^{10} - 324r^9 + 54r^8 + 1296r^7 + 756r^6 + 216r^5 + 756r^4 + 432r^3 + 54r^2 + 108r + 54)s^2 + (2r^{12} - 36r^{11} + 144r^{10} + 108r^9 - 1686r^8 - 1080r^7 - 1944r^5 - 2010r^4 - 612r^3 - 144r^2 + 108r + 110)s + (-2r^{12} + 36r^{11} - 218r^{10} + 464r^9 + 348r^8 + 72r^7 + 1300r^6 + 1392r^5 + 462r^4 - 108r^3 - 218r^2 + 56).$$

Lista de valores de ramificación para la curva E_s con $s =$ de la Familia Simple 1:

Lista de valores de ramificación Familia 1

r	a Punto Simple	a Punto Doble
-0.800504533199	-0.562090751958	1.948283029608
9.025290459343	-0.13851704727	-0.13851704727
(-0.74211110944+i*0.144597530168)	(0.536244128161+i*1.462194284217)	(0.024417255113-i*0.264678703274)
(-0.74211110944-i*0.144597530168)	(0.536244128161+i*1.462194284217)	(0.024417255113+i*0.264678703274)
(-0.269253168714+i*0.617463605067)	(0.485842330109+i*0.224787956313)	(-0.548220272402+i*0.945589213461)
(-0.269253168714-i*0.617463605067)	(0.485842330109-i*0.224787956313)	(-0.548220272402-i*0.945589213461)
(0.40182418322+i*0.186785864937)	(-0.309066630721+i*0.357323199221)	(-0.506623876737-i*1.125339983714)
(0.40182418322-i*0.186785864937)	(-0.309066630721-i*0.357323199221)	(-0.506623876737+i*1.125339983714)
(0.596820643159+i*1.349275273949)	(-0.301048197032+i*0.545114403225)	(0.103150228535+i*0.133579311939)
(0.596820643159-i*1.349275273949)	(-0.301048197032-i*0.545114403225)	(0.103150228535-i*0.133579311939)
(4.9003264887+i*3.30829542754)	(-0.0866449717825+i*0.125632099109)	(-0.139371031559+i*0.0671748461444)
(4.9003264887-i*3.30829542754)	(-0.0866449717825-i*0.125632099109)	(-0.139371031559-i*0.0671748461444)

Posición de las fibras de ramificación de la curva E_s en el espacio afín.

(x, y, z) coordenada afín con $t = 1$ Punto Simple
(1.500313170363,-2.119218794312,2.396252888852)
(2.119218794312,-1.500313170363,-2.396252888852)
((0.810395588107-i*0.263803737007),(1.040303141+i*0.399903361829),(-0.761561077881-i*0.265553669541))
((0.810395588107+i*0.263803737007),(1.040303141-i*0.399903361829),(-0.761561077881+i*0.265553669541))
((-0.762653846052+i*0.437801887455),(0.127486241683-i*1.18621791559),(-0.333786698752+i*1.453376539467))
((-0.762653846052-i*0.437801887455),(0.127486241683+i*1.18621791559),(-0.333786698752-i*1.453376539467))
((-1.013247753474-i*1.093637997629),(1.013247753474-i*1.093637997629),(-i*1.1570419698))
((-1.013247753474+i*1.093637997629),(1.013247753474+i*1.093637997629),(i*1.1570419698))
((-0.127486241683-i*1.18621791559),(0.762653846052+i*0.437801887455),(0.333786698752+i*1.453376539467))
((-0.127486241683+i*1.18621791559),(0.762653846052-i*0.437801887455),(-0.333786698752-i*1.453376539467))
((-1.040303141+i*0.399903361829),(-0.810395588107-i*0.263803737007),(0.761561077881-i*0.265553669541))
((-1.040303141-i*0.399903361829),(-0.810395588107+i*0.263803737007),(0.761561077881+i*0.265553669541))

(x, y, z) coordenada afín con $t = 1$ Punto Doble
(0.598605755089,1.899912390024,-1.722787259005)
(-1.899912390024,-0.598605755089,1.722787259005)
((0.680433498749+i*0.657982278259),(-0.218764814008-i*2.015142091958),(0.392054757573+i*2.266406290176))
((0.680433498749-i*0.657982278259),(-0.218764814008+i*2.015142091958),(0.392054757573-i*2.266406290176))
((0.830362328638-i*0.161722719589),(0.460774062484+i*0.478930174234),(0.14132233158+i*0.611298632215))
((0.830362328638+i*0.161722719589),(0.460774062484-i*0.478930174234),(0.14132233158-i*0.611298632215))
((-0.618133261443+i*0.17170364946),(0.618133261443+i*0.17170364946),(i*0.542942748982))
((-0.618133261443-i*0.17170364946),(0.618133261443-i*0.17170364946),(-i*0.542942748982))
((-0.460774062484+i*0.478930174234),(-0.830362328638-i*0.161722719589),(-0.14132233158+i*0.611298632215))
((-0.460774062484-i*0.478930174234),(-0.830362328638+i*0.161722719589),(-0.14132233158-i*0.611298632215))
((0.218764814008-i*2.015142091958),(-0.680433498749+i*0.657982278259),(-0.392054757573+i*2.266406290176))
((0.218764814008+i*2.015142091958),(-0.680433498749-i*0.657982278259),(-0.392054757573-i*2.266406290176))

Lista de valores de ramificación del divisor de ramificación de la familia
1:

<u>Listado de Valores de Ramificación Familia 1</u>	
<u>Valores 1 a 19</u>	
1	$-1.000000000000+(0.E-19)*i$
2	$-0.990264404429+(0.E-19)*i$
3	$0.990264404429+(0.E-19)*i$
4	$1.000000000000+(0.E-19)*i$
5	$0.E-19+(1.000000000000)*i$
6	$0.E-19+(-1.000000000000)*i$
7	$0.624840798157+(-0.0392063024644)*i$
8	$0.624840798157+(0.0392063024644)*i$
9	$-0.624840798157+(0.0392063024644)*i$
10	$-0.624840798157+(-0.0392063024644)*i$
11	$0.E-19+(1.90515868883)*i$
12	$0.E-19+(-1.90515868883)*i$
13	$0.E-19+(0.989153555773)*i$
14	$0.E-19+(-0.989153555773)*i$
15	$0.777658053547+(-0.246734513524)*i$
16	$0.777658053547+(0.246734513524)*i$
17	$-0.777658053547+(0.246734513524)*i$
18	$-0.777658053547+(-0.246734513524)*i$
19	$0.E-19+(0.E-19)*i$

4.1.2. Familia Simple 2

Sea la cuádrica Q_s definida por la ecuación $x^2 + y^2 - z^2 - st^2 = 0$, y la cúbica S_s definida por la ecuación $x^3 + y^3 + z^3 + xyz + t^3 = 0$, la Familia 2 que vamos a analizar.

Ecuación del divisor de ramificación de la Familia 2:

$$(-2r^{12} + 36r^{11} - 218r^{10} + 464r^9 + 348r^8 + 72r^7 + 1300r^6 + 1392r^5 + 462r^4 - 108r^3 - 218r^2 + 56)s^3 + (2br^{12} - 36br^{11} + 144br^{10} + 108br^9 - 1686br^8 - 1080br^7 - 1944br^5 - 2010br^4 - 612br^3 - 144br^2 + 108br + 110b)s + (54r^{10} - 324r^9 + 54r^8 + 1296r^7 + 756r^6 + 216r^5 + 756r^4 + 432r^3 + 54r^2 + 108r + 54)$$

Lista de valores de ramificación para la curva E_s con $s =$ de la Familia Simple 2:

Lista de valores de ramificación Familia		
r	a Punto Simple	a Punto Doble
-0.845577936582	(1.800645587937,-0.845577936582)	(-1.094296870815,-0.845577936582)
0.960856140422	(-0.511634988915,0.960856140422)	(-0.38104144704,0.960856140422)
2 (-0.698336538711+i*0.143502059032)	((-0.0293535835397-i*0.578234008328),(-0.698336538711+i*0.143502059032))	((0.578438804201+i*1.834907755241),(-0.698336538711+i*0.143502059032))
(-0.698336538711-i*0.143502059032)	((-0.0293535835397+i*0.578234008328),(-0.698336538711-i*0.143502059032))	((0.578438804201-i*1.834907755241),(-0.698336538711-i*0.143502059032))
(-0.0660796952098+i*0.668990299046)	((-0.642454064928+i*0.640519219071),(-0.0660796952098+i*0.668990299046))	((0.725396698531+i*0.403546915191),(-0.0660796952098+i*0.668990299046))
(-0.0660796952098-i*0.668990299046)	((-0.642454064928-i*0.640519219071),(-0.0660796952098-i*0.668990299046))	((0.725396698531-i*0.403546915191),(-0.0660796952098-i*0.668990299046))
(5.906777132+i*4.904389764088)	((-0.100478769503+i*0.0828569664736),(5.906777132+i*4.904389764088))	((-0.0661663438037+i*0.0822942003512),(5.906777132+i*4.904389764088))
(5.906777132-i*4.904389764088)	((-0.100478769503-i*0.0828569664736),(5.906777132-i*4.904389764088))	((-0.0661663438037-i*0.0822942003512),(5.906777132-i*4.904389764088))
1	(-0.25,1)	(-0.5,1)

Posición de las fibras de ramificación de la curva E_s en el espacio afín.

(x, y, z) coordenada afín con $t = 1$	
Punto Simple	
(1.35875824352,-2.045172361527,2.242532932353)	
(-1.000126928258,0.0167847584921,-0.0231430495728)	
((0.79858627718-i*0.457537107607),(1.20695290145+i*0.799179272526),(-0.85729344426-i*0.698930909048))	
((0.79858627718+i*0.457537107607),(1.20695290145-i*0.799179272526),(-0.85729344426+i*0.698930909048))	
((-0.917728572076+i*0.198593668354),(0.227904049772-i*0.944241703583),(-0.36717955778+i*1.082444769788))	
((-0.917728572076-i*0.198593668354),(0.227904049772+i*0.944241703583),(-0.36717955778-i*1.082444769788))	
((-0.117780097552+i*0.102114691683),(-0.999737112423-i*0.00673882773526),(-0.0831774414549+i*0.0635992412639))	
((-0.117780097552-i*0.102114691683),(-0.999737112423+i*0.00673882773526),(-0.0831774414549-i*0.0635992412639))	
(-1,0.5,0.5)(2 puntos simples)	
(x, y, z) coordenada afín con $t = 1$	
Punto Doble	
(0.533705568245,2.850626580064,-2.722299309875)	
(-0.990103158604,0.267747971712,0.228020264523)	
((0.638835286015+i*0.950288864349),(-0.334515986784-i*2.396751942445),(0.518042322387+i*2.719526646133))	
((0.638835286015-i*0.950288864349),(-0.334515986784+i*2.396751942445),(0.518042322387-i*2.719526646133))	
((1.077279851707-i*0.022464106759),(0.364194071564+i*0.917234194236),(0.373513545354+i*0.829557937142))	
((1.077279851707+i*0.022464106759),(0.364194071564-i*0.917234194236),(0.373513545354-i*0.829557937142))	
((-0.487916342542-i*1.880352133564),(-0.588865360669+i*0.323175922906),(0.355583654935+i*2.044940534267))	
((-0.487916342542+i*1.880352133564),(-0.588865360669-i*0.323175922906),(0.355583654935-i*2.044940534267))	
(-1,0,0)(2 punto doble)	

Lista de valores de ramificación del divisor de ramificación de la familia
2:

Listado de Valores de Ramificación Familia 2		
	Valores 1 a 44	Valores 45 a 88
1	-1.00729695782+(0.E-19)*i	-0.458264842046+(0.227148337827)*i
2	-1.00000000000+(0.E-19)*i	-0.458264842046+(-0.227148337827)*i
3	-0.650697636830+(0.E-19)*i	-0.211367837896+(0.449870220976)*i
4	0.E-19+(0.E-19)*i	-0.211367837896+(-0.449870220976)*i
5	1.00000000000+(0.E-19)*i	-0.0875894929788+(0.483930553037)*i
6	1.00654351340+(0.E-19)*i	-0.0875894929788+(-0.483930553037)*i
7	0.500000000000+(0.866025403784)*i	-0.0620960057558+(0.512335450197)*i
8	0.500000000000+(-0.866025403784)*i	-0.0620960057558+(-0.512335450197)*i
9	1.36523003936+(-0.0570669754788)*i	-0.435939541630+(0.0471321648338)*i
10	1.36523003936+(0.0570669754788)*i	-0.435939541630+(-0.0471321648338)*i
11	-0.633193569201+(1.21085738384)*i	0.672522845612+(1.44863797381)*i
12	-0.633193569201+(-1.21085738384)*i	0.672522845612+(-1.44863797381)*i
13	-0.732036470164+(-1.15379040836)*i	0.455434007556+(1.00853503910)*i
14	-0.732036470164+(1.15379040836)*i	0.455434007556+(-1.00853503910)*i
15	0.325348818415+(0.563520683678)*i	0.846205763056+(0.764816612854)*i
16	0.325348818415+(-0.563520683678)*i	0.846205763056+(-0.764816612854)*i
17	-0.500000000000+(0.866025403784)*i	1.11435774742+(0.737309458094)*i
18	-0.500000000000+(-0.866025403784)*i	1.11435774742+(-0.737309458094)*i
19	0.503648478908+(0.872344754624)*i	1.27321433064+(0.0669004906357)*i
20	0.503648478908+(-0.872344754624)*i	1.27321433064+(-0.0669004906357)*i
21	-0.503271756701+(0.871692252620)*i	1.32675373175+(0.0701967999248)*i
22	-0.503271756701+(-0.871692252620)*i	1.32675373175+(-0.0701967999248)*i
23	1.12136812534+(0.232947303967)*i	-1.73071911381+(0.512295872498)*i
24	1.12136812534+(-0.232947303967)*i	-1.73071911381+(-0.512295872498)*i
25	-0.762422345647+(0.854659631552)*i	-1.34992349598+(0.442114130268)*i
26	-0.762422345647+(-0.854659631552)*i	-1.34992349598+(-0.442114130268)*i
27	-0.358945779690+(-1.08760693552)*i	-1.47714936474+(-0.0605826814750)*i
28	-0.358945779690+(1.08760693552)*i	-1.47714936474+(0.0605826814750)*i
29	-1.90489632769+(0.E-19)*i	-1.05464682789+(1.24720819016)*i
30	-1.28549110539+(0.E-19)*i	-1.05464682789+(-1.24720819016)*i
31	-1.00000000000+(0.E-19)*i	-0.818884149178+(0.838633531492)*i
32	0.E-19+(0.E-19)*i	-0.818884149178+(-0.838633531492)*i
33	1.00000000000+(0.E-19)*i	0.158753676568+(1.38940665497)*i
34	1.00002184611+(0.E-19)*i	0.158753676568+(-1.38940665497)*i
35	1.03584278511+(-0.0395619792408)*i	0.366010697495+(1.64421310518)*i
36	1.03584278511+(0.0395619792408)*i	0.366010697495+(-1.64421310518)*i
37	0.593864444340+(0.0839284025446)*i	0.223181343788+(1.24504221386)*i
38	0.593864444340+(-0.0839284025446)*i	0.223181343788+(-1.24504221386)*i
39	0.527034430284+(0.380110129951)*i	-0.197563359995+(1.44388242637)*i
40	0.527034430284+(-0.380110129951)*i	-0.197563359995+(-1.44388242637)*i
41	0.E-58+(1.00000000000)*i	-0.218992469959+(1.70142869185)*i
42	0.E-58+(-1.00000000000)*i	-0.218992469959+(-1.70142869185)*i
43	0.0988711054534+(0.435933672897)*i	0.00627238628393+(1.28595373611)*i
44	0.0988711054534+(-0.435933672897)*i	0.00627238628393+(-1.28595373611)*i

4.1.3. Familia de Seidel

Sea la cuádrica Q_s definida por la ecuación $x^2 + y^2 - z^2 - (1 + s^2)t^2 = 0$, y la cúbica S_s definida por la ecuación $x^3 + y^3 - sz^3 - st^3 - xyz + 7x^2y + 13y^2z + xz^2 = 0$, la Familia de Seidel que vamos analizar.

El espacio total de esta familia es una superficie K_3 de género 4. La descripción de la monodromía geométrica de esta superficie $Q \cap S$ como familia de género 4 es un pincel de Lefschetz.

La monodromía geométrica de este pincel es un paso necesario para la comprobación de la conjetura de Simetría Especular versión de Seidel para esta superficie.

Ecuación del divisor de ramificación de la Familia de Seidel:

$$\begin{aligned} & (27r^{12} + 1350r^{10} + 432r^9 + 15525r^8 + 10800r^7 - 32076r^6 - 10800r^5 + 15525r^4 - \\ & 432r^3 + 1350r^2 + 27)s^8 + (-6808r^{12} + 58392r^{11} - 15348r^{10} - 254928r^9 - \\ & 160404r^8 + 44400r^7 + 301928r^6 + 146016r^5 - 123984r^4 - 125128r^3 - 40884r^2 + \\ & 48048r - 7492)s^7 + (-22411r^{12} + 151216r^{11} + 22034r^{10} + 289808r^9 + 62335r^8 + \\ & 349664r^7 + 74300r^6 + 285600r^5 + 285443r^4 + 77616r^3 + 46834r^2 - 58992r + \\ & 9737)s^6 + (-20370r^{12} + 175176r^{11} - 44532r^{10} - 764352r^9 - 478350r^8 + 134496r^7 + \\ & 905784r^6 + 439344r^5 - 374814r^4 - 374952r^3 - 124164r^2 + 144144r - 22530)s^5 + \\ & (-65280r^{12} + 462504r^{11} + 11832r^{10} + 843288r^9 + 80688r^8 + 870864r^7 + 401280r^6 + \\ & 963504r^5 + 797568r^4 + 230472r^3 + 173352r^2 - 183816r + 27888)s^4 + (-14254r^{12} + \\ & 208152r^{11} - 127980r^{10} - 571128r^9 - 73458r^8 + 751824r^7 + 981336r^6 + 303984r^5 - \\ & 794610r^4 - 412136r^3 - 161964r^2 + 155016r - 20910)s^3 + (-65253r^{12} + 460776r^{11} + \\ & 10050r^{10} + 891240r^9 + 105285r^8 + 832848r^7 + 631644r^6 + 967824r^5 + 826485r^4 + \\ & 261576r^3 + 179778r^2 - 182520r + 27915)s^2 + (-800r^{12} + 92232r^{11} - 96204r^{10} - \\ & 60192r^9 + 255612r^8 + 661728r^7 + 394328r^6 + 10224r^5 - 532440r^4 - 161448r^3 - \\ & 75660r^2 + 59568r - 5764)s + (-22492r^{12} + 151216r^{11} + 17984r^{10} + 288512r^9 + \\ & 15760r^8 + 317264r^7 + 170528r^6 + 318000r^5 + 238868r^4 + 78912r^3 + 42784r^2 - \\ & 58992r + 9656) \end{aligned}$$

Lista de valores de ramificación para la curva E_s con $s =$ de la Familia de Seidel:

Lista de valores de ramificación Familia Seidel

r	a Punto Simple	a Punto Doble
-0.760450397067	0.0992706575101	0.877821934239
7.098377791608	0.0763717131873	-0.131797524192
(-0.524138000709+i*0.830026305695)	(0.562228693754+i*0.564679395825)	(0.243633405478+i*0.341607876438)
(-0.524138000709-i*0.830026305695)	(0.562228693754-i*0.564679395825)	(0.243633405478-i*0.341607876438)
(-0.352770661726+i*0.896648507527)	(0.126807242585+i*0.444479172706)	(0.418960497162+i*0.475133446298)
(-0.352770661726-i*0.896648507527)	(0.126807242585-i*0.444479172706)	(0.418960497162-i*0.475133446298)
(0.000710074632701+i*1.0193915601)	(1.001922614881+i*0.765839422462)	(0.000573633787176+i*0.50932222053)
(0.000710074632701-i*1.0193915601)	(1.001922614881-i*0.765839422462)	(0.000573633787176-i*0.50932222053)
(0.250701626288+i*0.0216080321369)	(-0.767306062569+i*0.524584755428)	(-0.761882865063-i*0.225889719376)
(0.250701626288-i*0.0216080321369)	(-0.767306062569-i*0.524584755428)	(-0.761882865063+i*0.225889719376)
(0.818084037851+i*0.825674550205)	(-1.296719584212+i*0.0356332039566)	(-0.274296876388+i*0.343560851478)
(0.818084037851-i*0.825674550205)	(-1.296719584212-i*0.0356332039566)	(-0.274296876388-i*0.343560851478)

Posición de las fibras de ramificación de la curva E_s en el espacio afín.

(x, y, z) coordenada afín con $t = 1$ Punto Simple
(0.802314341618, 0.849019178159, -0.603773026598)
(-10.870145091328, 2.084230545592, 11.022888517702)
((0.827929109897+i*0.457749604056), (-0.526768352607+i*0.341389352127), (0.296528277611+i*0.671609187593))
((0.827929109897-i*0.457749604056), (-0.526768352607-i*0.341389352127), (0.296528277611-i*0.671609187593))
((0.284559849565-i*0.0699106532401), (0.113449076576-i*0.0861953741439), (-0.0309453643953+i*0.958868998653))
((0.284559849565+i*0.0699106532401), (0.113449076576+i*0.0861953741439), (-0.0309453643953-i*0.958868998653))
((2.0434777874+i*0.540826050128), (-0.559957607634+i*2.043790521258), (-0.0396325576403+i*0.990852794796))
((2.0434777874-i*0.540826050128), (-0.559957607634-i*2.043790521258), (-0.0396325576403-i*0.990852794796))
((-0.96445623344+i*0.47856407185), (0.592599755999+i*0.229868554506), (-0.570155891697+i*0.570605439006))
((-0.96445623344-i*0.47856407185), (0.592599755999-i*0.229868554506), (-0.570155891697-i*0.570605439006))
((-2.082844349309+i*0.962196637538), (-1.180494046124+i*2.083034808122), (-0.510594819114-i*0.890930229625))
((-2.082844349309-i*0.962196637538), (-1.180494046124-i*2.083034808122), (-0.510594819114+i*0.890930229625))

(x, y, z) coordenada afín con $t = 1$ Punto Doble
(1.130641244011, -0.335080076893, 0.625002624467)
(-0.589297777757, -0.871097237426, 0.325702729373)
((0.571458101011-i*0.134931969941), (0.177517900541+i*0.0463449322203), (-0.0841912900553+i*0.818147722818))
((0.571458101011+i*0.134931969941), (0.177517900541-i*0.0463449322203), (-0.0841912900553-i*0.818147722818))
((0.755847830589+i*0.166396906102), (-0.147649334639+i*0.416094328469), (0.082073163735+i*0.783869986493))
((0.755847830589-i*0.166396906102), (-0.147649334639-i*0.416094328469), (0.082073163735-i*0.783869986493))
((0.00119699542159+i*0.0191964203734), (-0.0383967313132+i*0.00189282845981), (-0.0000497278472384+i*0.999448020686))
((0.00119699542159-i*0.0191964203734), (-0.0383967313132-i*0.00189282845981), (-0.0000497278472384-i*0.999448020686))
((-0.967502252664-i*0.225151261103), (0.627751518007-i*0.146187418884), (-0.556263477461-i*0.226628177649))
((-0.967502252664+i*0.225151261103), (0.627751518007+i*0.146187418884), (-0.556263477461+i*0.226628177649))
((-0.631672407485-i*0.107267836302), (-0.0161346954352+i*0.109163397181), (0.0830786547089+i*0.794389539258))
((-0.631672407485+i*0.107267836302), (-0.0161346954352-i*0.109163397181), (0.0830786547089-i*0.794389539258))

Lista de valores de ramificación del divisor de ramificación de la familia de Seidel:

Listado de Valores de Ramificación Familia Seidel

	Valores 1 a 61	Valores 62 a 122
1	-6.95598242785+(0.E-19)*i	0.172701637602+(0.608076125604)*i
2	-6.89780351580+(0.E-19)*i	0.172701637602+(-0.608076125604)*i
3	-2.96811118520+(0.E-19)*i	18.9063419831+(-3.04253288890)*i
4	0.879164597260+(0.E-19)*i	18.9063419831+(3.04253288890)*i
5	1.26938106546+(0.E-19)*i	0.468833891319+(0.904405829703)*i
6	1.31742686986+(0.E-19)*i	0.468833891319+(-0.904405829703)*i
7	1.38111901148+(0.E-19)*i	0.720994843049+(0.995682203298)*i
8	1.50029457104+(0.E-19)*i	0.720994843049+(-0.995682203298)*i
9	9.32553395547+(0.E-19)*i	-0.0119819744395+(1.60164401825)*i
10	9.37257182464+(0.E-19)*i	-0.0119819744395+(-1.60164401825)*i
11	12.0541602135+(0.E-19)*i	0.187967430921+(1.10787883129)*i
12	12.5170241832+(0.E-19)*i	0.187967430921+(-1.10787883129)*i
13	12.5248205547+(0.E-19)*i	0.149565848162+(1.06321222083)*i
14	14.0000000000+(0.E-19)*i	0.149565848162+(-1.06321222083)*i
15	325.861073915+(0.E-19)*i	0.132485433870+(1.10796362431)*i
16	325.873097395+(0.E-19)*i	0.132485433870+(-1.10796362431)*i
17	-0.157886307025+(1.07693897462)*i	-0.00103562952055+(1.24506646489)*i
18	-0.157886307025+(-1.07693897462)*i	-0.00103562952055+(-1.24506646489)*i
19	-0.142335605071+(1.09891755024)*i	0.0286314701645+(1.16251177077)*i
20	-0.142335605071+(-1.09891755024)*i	0.0286314701645+(-1.16251177077)*i
21	-0.121143085414+(0.642526176397)*i	0.273318419704+(1.32230550617)*i
22	-0.121143085414+(-0.642526176397)*i	0.273318419704+(-1.32230550617)*i
23	0.00850017118079+(0.834196120381)*i	-0.0568058575879+(1.39767908887)*i
24	0.00850017118079+(-0.834196120381)*i	-0.0568058575879+(-1.39767908887)*i
25	-0.0149312322569+(0.836362754017)*i	-0.159123382986+(1.05077807489)*i
26	-0.0149312322569+(-0.836362754017)*i	-0.159123382986+(-1.05077807489)*i
27	-0.196545380822+(0.875476319887)*i	-0.133432095204+(1.09203192451)*i
28	-0.196545380822+(-0.875476319887)*i	-0.133432095204+(-1.09203192451)*i
29	-0.713270869854+(0.869625478296)*i	-0.194695115642+(1.11765686468)*i
30	-0.713270869854+(-0.869625478296)*i	-0.194695115642+(-1.11765686468)*i
31	-0.452458487401+(0.960827775156)*i	-0.349887618483+(1.06682513211)*i
32	-0.452458487401+(-0.960827775156)*i	-0.349887618483+(-1.06682513211)*i
33	-0.467941106127+(0.836835154191)*i	-0.260046095224+(0.136596031817)*i
34	-0.467941106127+(-0.836835154191)*i	-0.260046095224+(-0.136596031817)*i
35	-0.259980438602+(1.30687995447)*i	0.143155212000+(0.196607978010)*i
36	-0.259980438602+(-1.30687995447)*i	0.143155212000+(-0.196607978010)*i
37	-0.154207700697+(-0.108133254368)*i	0.116363880372+(0.901068269634)*i
38	-0.154207700697+(0.108133254368)*i	0.116363880372+(-0.901068269634)*i
39	0.138224544003+(0.139945607179)*i	0.198540698182+(0.874516685973)*i
40	0.138224544003+(-0.139945607179)*i	0.198540698182+(-0.874516685973)*i
41	0.711426911942+(1.00069138221)*i	-0.440230664934+(0.733021148725)*i
42	0.711426911942+(-1.00069138221)*i	-0.440230664934+(-0.733021148725)*i
43	0.410344786460+(1.07307848839)*i	-0.197497929228+(0.873578315578)*i
44	0.410344786460+(-1.07307848839)*i	-0.197497929228+(-0.873578315578)*i
45	0.466173856042+(0.837155460001)*i	-0.145232452298+(0.935250382713)*i
46	0.466173856042+(-0.837155460001)*i	-0.145232452298+(-0.935250382713)*i
47	0.216465206668+(1.43940017978)*i	0.00699609237050+(0.778549794427)*i
48	0.216465206668+(-1.43940017978)*i	0.00699609237050+(-0.778549794427)*i
49	-0.0742493791962+(1.59780707817)*i	0.0434785394926+(0.660242205013)*i
50	-0.0742493791962+(-1.59780707817)*i	0.0434785394926+(-0.660242205013)*i
51	-0.000120629009337+(1.39827700174)*i	-0.0156219169172+(0.846479395979)*i
52	-0.000120629009337+(-1.39827700174)*i	-0.0156219169172+(-0.846479395979)*i
53	0.157818669862+(1.06713916617)*i	0.0261583570735+(0.842831971592)*i
54	0.157818669862+(-1.06713916617)*i	0.0261583570735+(-0.842831971592)*i
55	0.148834438724+(1.09128333392)*i	-0.727848190929+(-1.01656775866)*i
56	0.148834438724+(-1.09128333392)*i	-0.727848190929+(1.01656775866)*i
57	0.000573944858338+(1.24297519913)*i	8.51169507597+(-2.39770820118)*i
58	0.000573944858338+(-1.24297519913)*i	8.51169507597+(2.39770820118)*i
59	0.195969731972+(0.874957180362)*i	8.57267835108+(-2.43612392382)*i
60	0.195969731972+(-0.874957180362)*i	8.57267835108+(2.43612392382)*i
61	0.E-19+(1.00000000000)*i	0.E-19+(-1.00000000000)*i

4.1.4. Familia de Bring

Sea la cuádrica Q_s definida por la ecuación $5x^2 + y^2 + z^2 + t^2 = 0$, y la cúbica S_s definida por la ecuación $5x^3 - sx(y^2 + z^2 + t^2) - 2yzt = 0$, la Familia de Bring que vamos analizar.

La fibra para $s = 0$ de esta familia es la curva de Bring, que es la única curva de género 4 con automorfismos de orden 5.

Ecuación del divisor de ramificación de la Familia de Bring:

$$(27r^{10} - 54r^6 + 27r^2)s^2 + (-2r^{12} + 1350r^{10} + 66r^8 - 2700r^6 + 66r^4 + 1350r^2 - 2)s + (-50r^{12} + 16855r^{10} + 1650r^8 - 33710r^6 + 1650r^4 + 16855r^2 - 50)$$

Lista de valores de ramificación para la curva E_s con $s =$ de la Familia de Bring:

Lista de valores de ramificación Familia Bring

r	a Punto Simple	a Punto Doble
-8.219438452884	0.135667124111	0.11816693231
-0.121662812579	-0.946119642219	0.236178764077
0.121662812579	0.946119642219	-0.236178764077
8.219438452884	-0.135667124111	-0.11816693231
(-0.970828117605+i*0.239776491898)	(0.374589772834+i*0.568603347586)	(1.5013142796-i*3.993377784538)
(-0.970828117605-i*0.239776491898)	(0.374589772834-i*0.568603347586)	(1.5013142796+i*3.993377784538)
(i*1.277029856692)	(i*0.735827596123)	(i*0.3425106735)
(-i*1.277029856692)	(-i*0.735827596123)	(-i*0.3425106735)
(i*0.783067047931)	(i*0.0770383462459)	(i*0.718461650494)
(-i*0.783067047931)	(-i*0.0770383462459)	(-i*0.718461650494)
(0.970828117605-i*0.239776491898)	(-0.374589772834-i*0.568603347586)	((-1.5013142796+i*3.993377784538)
(0.970828117605+i*0.239776491898)	(-0.374589772834+i*0.568603347586)	((-1.5013142796-i*3.993377784538)

Posición de las fibras de ramificación de la curva E_s en el espacio afín.

(x, y, z) coordenada afín con $t = 1$ Punto Simple
(-0.810452518108,-1.230215153418,1.08178676633)
(-0.810452518108,1.230215153418,-1.08178676633)
(0.810452518108,1.230215153418,1.08178676633)
(0.810452518108,-1.230215153418,-1.08178676633)
(0.749179545668,(-i*0.924396591939),(i*1.137206695171))
(0.749179545668,(i*0.924396591939),(-i*1.137206695171))
((i*0.658789249878),-0.879347619255,(i*0.812865942369))
((-i*0.658789249878),-0.879347619255,(-i*0.812865942369))
((-i*0.658789249878),0.879347619255,(i*0.812865942369))
((i*0.658789249878),0.879347619255,(-i*0.812865942369))
(-0.749179545668,(-i*0.924396591939),(-i*1.137206695171))
(-0.749179545668,(i*0.924396591939),(i*1.137206695171))

(x, y, z) coordenada a fin con $t = 1$
Punto Doble

(0.354345696388,-0.942531654582,-0.118011831767)
 (0.354345696388,0.942531654582,0.118011831767)
 (-0.354345696388,0.942531654582,-0.118011831767)
 (-0.354345696388,-0.942531654582,0.118011831767)
 (3.002628559205,(i*8.4737266173),(-i*7.986755569075))
 (3.002628559205,(-i*8.4737266173),(i*7.986755569075))
 ((-i*0.375950976994),0.125207287409,(i*1.060972323995))
 ((i*0.375950976994),0.125207287409,(-i*1.060972323995))
 ((i*0.375950976994),-0.125207287409,(i*1.060972323995))
 ((-i*0.375950976994),-0.125207287409,(-i*1.060972323995))
 (-3.002628559205,(i*8.4737266173),(i*7.986755569075))
 (-3.002628559205,(-i*8.4737266173),(-i*7.986755569075))

Lista de valores de ramificación del divisor de ramificación de la familia de Bring:

Listado de Valores de Ramificación Familia Bring	
Valores 1 a 5	
1	-26.3277092210+(0.E-19)*i
2	-25.5579088621+(0.E-19)*i
3	-25.0000000000+(0.E-19)*i
4	-24.4420911379+(0.E-19)*i
5	-23.6722907790+(0.E-19)*i

4.2. Cálculo Analítico: Ejemplo Académico

Debido al mal comportamiento de la integración numérica en divisores de ramificación de alto grado, no podemos calcular la monodromía de trenzas del divisor de ramificación de las familias estudiadas en los ejemplos algebraicos.

Para mostrar que el problema se reduce a la inestabilidad de la ecuación diferencial de la ramificación, aplicamos nuestro paquete de Matlab al cálculo de la monodromía de trenzas definida por una curva plana $C \subset \mathbb{P}_{\mathbb{C}}^2$ donde C tiene grado 6 y la proyección $\pi(x, y) = x$ restringida a C no ramifica en la recta del infinito.

Para curvas de grado 6, nuestro paquete de Matlab funciona y calcula la monodromía de trenzas.

Al crecer el grado de la curva C aparecen valores de ramificación de la aplicación π de $\pi(x, y) = x$ más y más próximos, y el integrador numérico `rk4int1avect.m` se encalla en las aristas de la araña de Voronoi que los separan.

La curva escogida para probar el paquete es una curva $C \subset \mathbb{C}^2$ de grado 6, con coeficientes complejos, cuyas partes real e imaginaria son racionales, fracciones cociente de enteros aleatorios. Tiene ecuación:

$$f(x, y) = (y^6 \cdot y^5 \cdot \dots \cdot y \cdot 1)A \begin{pmatrix} x^6 \\ x^5 \\ \vdots \\ x \\ 1 \end{pmatrix} = 0$$

donde A es la matriz:

$$\begin{pmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & -\frac{90477999}{2097152} \\
 0 & 0 & 0 & 0 & 0 & -\frac{334576667}{4194304} & \frac{200241597}{1048576} \\
 0 & 0 & 0 & 0 & \frac{658552831}{4194304} & \frac{287400699}{2097152} & -\frac{413829239}{4194304} \\
 0 & 0 & 0 & -\frac{87953595}{524288} & -\frac{403149565}{4194304} & -\frac{296334331}{2097152} & \frac{267075309}{4194304} \\
 0 & 0 & -\frac{179945465}{1048576} & \frac{656502727}{4194304} & \frac{205385583}{2097152} & \frac{654507537}{4194304} & -\frac{65692467}{262144} \\
 0 & -\frac{167902133}{4194304} & \frac{512390969}{2097152} & -\frac{11577099}{4194304} & \frac{160433425}{1048576} & \frac{296365711}{4194304} & -\frac{75076253}{2097152} \\
 \frac{858174361}{4194304} & -\frac{54814385}{524288} & \frac{379564691}{4194304} & \frac{291421165}{2097152} & \frac{107426301}{4194304} & -\frac{120198949}{1048576} & -\frac{637922217}{4194304}
 \end{pmatrix} +$$

$$i \cdot \begin{pmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & -\frac{238244649}{2097152} \\
 0 & 0 & 0 & 0 & 0 & -\frac{669337567}{4194304} & -\frac{10413927}{131072} \\
 0 & 0 & 0 & 0 & \frac{746231771}{4194304} & \frac{505408033}{2097152} & -\frac{941954555}{4194304} \\
 0 & 0 & 0 & \frac{83628901}{1048576} & \frac{189546783}{4194304} & \frac{497581515}{2097152} & -\frac{218813527}{4194304} \\
 0 & 0 & -\frac{18061095}{524288} & \frac{42456099}{4194304} & \frac{142660053}{2097152} & -\frac{313614067}{4194304} & \frac{107064495}{1048576} \\
 0 & -\frac{982002457}{4194304} & \frac{280473663}{2097152} & -\frac{636998095}{4194304} & -\frac{50331561}{262144} & -\frac{1063483541}{4194304} & -\frac{203799799}{2097152} \\
 -\frac{941459691}{4194304} & -\frac{58663751}{1048576} & \frac{695201711}{4194304} & \frac{517870643}{2097152} & \frac{895606713}{4194304} & \frac{130088419}{524288} & -\frac{806760013}{4194304}
 \end{pmatrix}$$

Con Pari-GP podemos hallar la lista de valores de ramificación de la proyección $\pi(x, y) = x$ sobre C .

Esta lista sale:

Listado de Valores de Ramificación
0.294079093720+(0.0619810833635)*i
0.295771502100+(-0.441080857621)*i
0.241006013194+(-0.537517667066)*i
0.405364356603+(-0.350470045778)*i
1.47999936841+(0.288567974643)*i
0.780828483303+(0.381848797808)*i
0.749116463602+(0.00833693376190)*i
0.871540496681+(0.0223597937043)*i
1.19438618828+(-0.844213092347)*i
1.01019494084+(-0.504949128341)*i
1.43557288542+(-0.369773749244)*i
0.856406059796+(-0.378438519391)*i
-0.188288936653+(-0.833756559001)*i
-0.0977092127327+(-0.782592184934)*i
0.0606240830678+(-0.966070547183)*i
-0.795232281197+(-0.501373426147)*i
-0.583358440297+(-1.27534822672)*i
-0.746277117230+(-0.666410203573)*i
-1.97249650595+(-0.415871383005)*i
-1.18211851520+(-0.188082353135)*i
-1.00346057480+(-0.342177130638)*i
-0.931140459843+(-0.149342701495)*i
0.195072660895+(1.49517785480)*i
0.155462580553+(1.07502766386)*i
0.00467910755291+(1.07204682004)*i
0.0356032688170+(1.19320905902)*i
-0.636387359742+(0.825964244774)*i
-0.617644806218+(0.660693278542)*i
-1.02776241306+(0.901338734114)*i
-1.16916097585+(0.729586826227)*i

Al aplicar el paquete de Matlab a esta curva, araña de Voronoi produce la araña poligonal indicada en la figura 4.1.

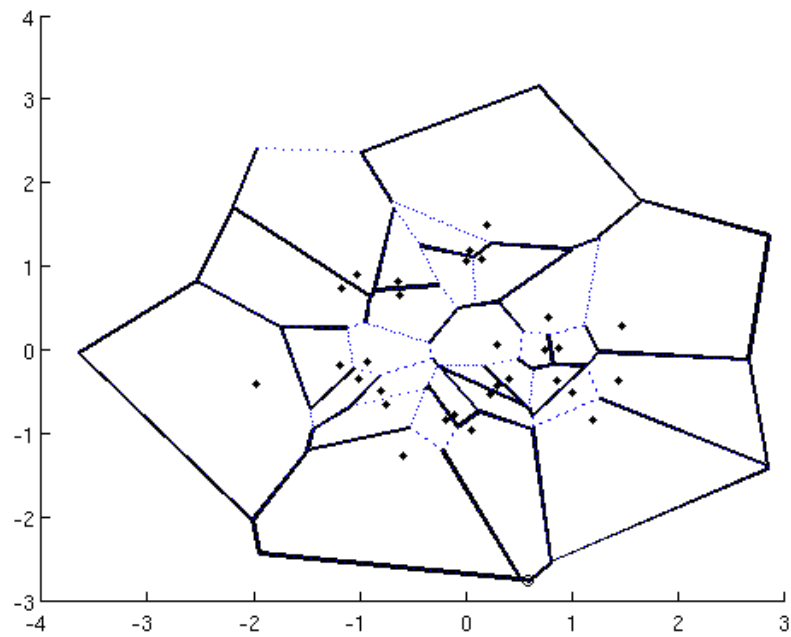


Figura 4.1: *Araña Voronoi ejemplo academico. Punto base en $x = 0,5959, y = -2,768$*

La integración numérica de la edo de la curva, seguida de la evaluación de cruzamientos, produce la siguiente tabla de valores de la monodromía:

<i>Generador</i>	<i>Valor Singular que rodea (redondeado al 4^o decimal)</i>	<i>Trenza</i>
1	-1.0278 + 0.9013i	<i>ecbadbcebdcebabdcedebEcBCBDCdDdDBABECDBECBDABCE</i>
2	0.0356 + 1.1932i	<i>ecbadbcebdcebabdDBABEdCcebabdDBABECDBECBDABCE</i>
3	0.1951 + 1.4952i	<i>ecbadbcebdcebabdDBABECbDBECBDABCE</i>
4	-0.6364 + 0.8260i	<i>ecbadbcebdBcDebabdDcdDCBAbEdCbDBECBDABCE</i>
5	0.0047 + 1.0720i	<i>ecbadbcebdBcDebabdABACbDBECBDABCE</i>
6	0.1555 + 1.0750i	<i>ecbadbcebdBcbDBECBDABCE</i>
7	-0.6176 + 0.6607i	<i>ecbadbcebdBabaDcdcDCBAbDBECBDABCE</i>
8	-0.9311 - 0.1493i	<i>ecbadbcebdBabcedCEcDCBAbDBECBDABCE</i>
9	0.2941 + 0.0620i	<i>ecbadbcebdBabBcAbDBECBDABCE</i>
10	0.7491 + 0.0083i	<i>ecbadbcebdBaDedAbDBECBDABCE</i>
11	0.7808 + 0.3818i	<i>ecbadbcebdBaDbdDBECBDABCE</i>
12	1.4800 + 0.2886i	<i>ecbadbcebdDBECbcCBDABCE</i>
13	0.8715 + 0.0224i	<i>ecbadbcCBceEdeECbcCBDABCE</i>
14	0.4054 - 0.3505i	<i>ecbadbcCBceECbcbabCBbABeECbcCBDABCE</i>
15	0.8564 - 0.3784i	<i>ecbadbcCBceEbaBebBECbcCBDABCE</i>
16	0.2958 - 0.4411i	<i>ecbadbcCBcebBEbcBABCcdabCaBebBECbcCBDABCE</i>
17	0.2410 - 0.5375i	<i>ecbadbcCBcebBEbAcBADcdbeEaebBECbcCBDABCE</i>
18	1.0102 - 0.5049i	<i>ecbadbcCBcebBEAeadEeECbcCBDABCE</i>
19	1.4356 - 0.3698i	<i>ecbadbcCBceEbBDABCE</i>
20	1.1944 - 0.8442i	<i>ecbadbBeDAEeBCE</i>
21	-0.7952 - 0.5014i	<i>ecbEBDCcdCBabcDdbeBCE</i>
22	-0.0977 - 0.7826i	<i>ecbEBDdCbcbBCE</i>
23	-0.7463 - 0.6664i	<i>ecbEBCBcDdCBbcdCbcbBCE</i>
24	-0.1883 - 0.8338i	<i>ecbEBCBcDceaDcdcACbcbBCE</i>
25	0.0606 - 0.9661i	<i>ecbEBCBcaCb</i>
26	-0.5834 - 1.2753i	<i>BDCdAEacedb</i>
27	-1.0035 - 0.3422i	<i>BDECABAbEabacedb</i>
28	-1.9725 - 0.4159i	<i>BDECABAcDdabacedb</i>
29	-1.1821 - 0.1881i	<i>BDECABADdEecDdabacedb</i>
30	-1.1692 + 0.7296i	<i>BDECABADEbCecdabacedb</i>

Para indicar las trenzas de 6 hebras en B_6 , hemos seguido el convenio de escritura de los generadores:

$$a = \sigma_1, b = \sigma_2, c = \sigma_3, d = \sigma_4, e = \sigma_5$$

$$A = \sigma_1^{-1}, B = \sigma_2^{-1}, C = \sigma_3^{-1}, D = \sigma_4^{-1}, E = \sigma_5^{-1}$$

Como ejemplo, el generador número 1 de la araña de Voronoi empleada:

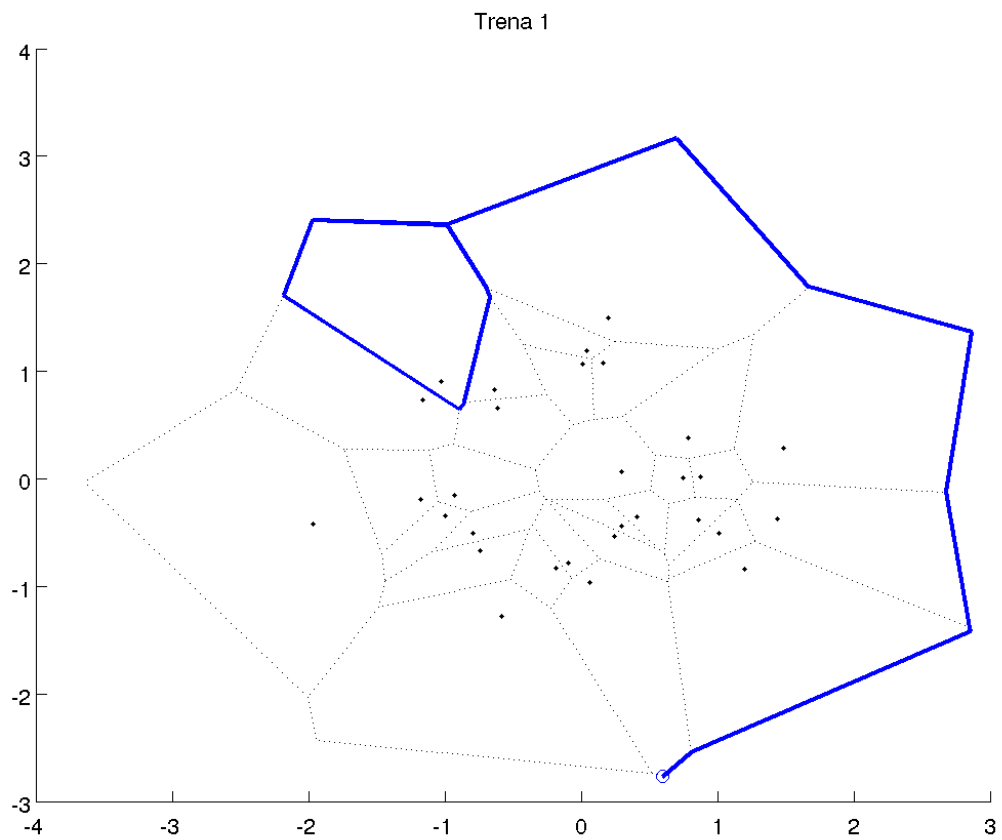


Figura 4.2: *imagen de la trenza 1 del ejemplo academico*

La trenza descrita por los seis puntos de la curva de ramificación a lo largo de este camino es la siguiente:

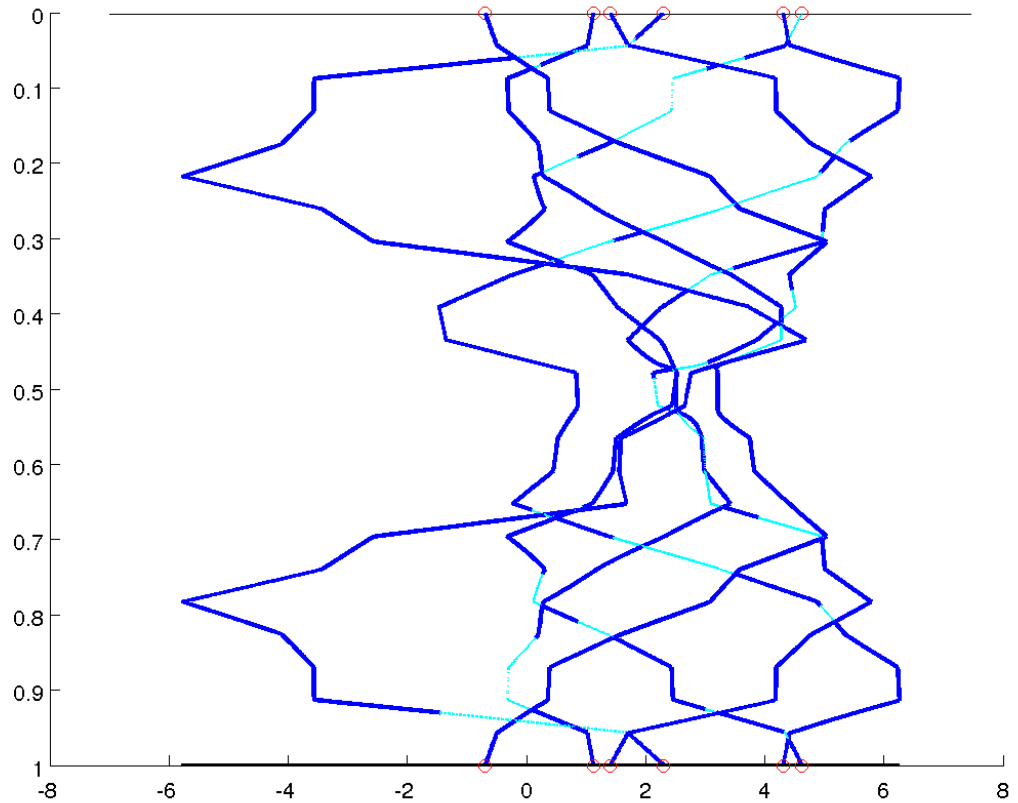


Figura 4.3: *imagen de la trenza 1 del ejemplo academico*

En la figura 4.3, mostramos a la trenza número 1, calculada por el paquete de Matlab. Aquí hemos establecido un convenio de representación, cuando se cruzan dos hebras sobre el papel, la línea oscura pasa por encima y la línea clara pasa por debajo.

El paquete de Matlab, también graba videos con visión central del movimiento de los puntos de la curva. Este video de la trenza será mostrado en la defensa de esta tesis.

Bibliografía

- [ABV] Amorós, Braungardt, Vindel. *Symplectic Lefschetz pencils of curves as gonial coverings*. Peperint, dept. Matemática Aplicada I, Universidad Politécnica de Cataluña.
- [ACG2] Arbarello E., Cornalba M., Griffiths Ph. *Geometry of algebraic curves*, Vol II, 21.11.5 pag. 859.
- [Am] Amorós, Jaume. *Mètodes Numèrics per a Enginyeria Química*, www.ma1.upc.edu/~numeric.
- [Ar] Arbarello, E. *Thesis, (Chapter II and III)*, Columbia University (1973).
- [Au1] Auroux, Denis. *Fiber Sums of Genus 2 Lefschetz Fibrations*. Turkish J. Math. 27 (2003), no. 1, 1–10.
- [Au2] Auroux, Denis. *The Canonical Pencils on Horikawa Surfaces*. *Geom. Topol.* 10 (2006), 2173–2217.
- [ADKY] Auroux, D. Donaldson, S. K.; Katzarkov, L.; Yotov, M. *Fundamental Groups of Complements of Plane Curves and Symplectic Invariants*. *Topology* 43 (2004), no. 6, 1285–1318.
- [BPV] Barth W. Peters C.; Van der Ven A. *Compact Complex Surfaces*. Springer-Verlag, 1984.
- [Bu] Burden; Faires *Numerical Analysis*. Third Edicition, 1985.
- [Cl] Clebsch A. *Zür Theorie der Riemannâschen Fläche*, *Math. Ann.*, 6 (1872), pp. 216-230. *Ann. of Math.*
- [DL] Decker, Losen. *Computing in Algebraic Geometry. A quick start using SINGULAR*.
- [Di] Dimca, A. *Singularities and Topology of Hypersurfaces*. Springer Verlag, 1992 (seccion 4.2).
- [Db] De Berg, Van Kreveld, Overmars, Schwarzkopf *Computational Geometry, Algorithms and Applications*. Second Edition, Springer Verlag, 2000.
- [FM] Farb, Margalit, *A primer on mapping class groups*. Princeton U.P., 2011.

- [Gr] Griffiths Ph. A. *Introduction to Algebraic Curves*. VOL 76, American Mathematic Society, 1989.
- [Gp] Guillemin, Pollack. *Differential Topology*. Prentice Hall, 1974.
- [Hur] Hurwitz A. *Ueber Riemann'sche Flächen mit gegebenen Verzweigungspunkten*, in Math Annalen, vol 39; pp. 33 (1891).
- [Ku] Yusuke Kuno. *On the Global Monodromy of a Lefschetz Fibration Arising from the Fermat*. Surface of degree 4. arXiv:0811.3274.
- [LM] Lalonde, F. McDuff, D. *The Classification of ruled symplectic 4-minifolds*. Math. Res. Letters 3 (1996) pag. 769-778.
- [Mch] Mochizuki, Shinichi. *The Geometry of the Compactification of the Hurwitz Scheme*. Publ. Res. Inst. Math. secc 31 (1995) 355-341.
- [Mo1] Moishezon B. *Complex Surfaces and Connected Sums of Complex Projective Planes*. LNM 603, Springer-Verlag, 1977.
- [Mo2] Moishezon B. Teicher M. *Braid Group Techniques in Complex Geometry IV. Braid Monodromy of the Branch Curve a 'S₃ of $\mathbb{V} \rightarrow \mathbb{CP}^3$ and Application to $\pi_1(\mathbb{CP}^3 \rightarrow S_3, *)$. Classification of Algebra Varieties (L'Alquila, 1992)*. 333–358, Contemp. Math., 162, American Mathematic Society, 1994.
- [R] Riemann, B. *Principes fondamentaux pour une théorie generale des fonctions d'une variable complexe*. contenido en la traducción francesa: Oeuvres Completes de B. Riemann, J. Gabay, Paris, 1990.
- [Seidel] Seidel, P. *Fukuya Categories and Picard-Lefschetz Theory*. European Math. Soc, 2008.
- [Sin] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR 3-1-2 — A computer algebra system for polynomial computations. <http://www.singular.uni-kl.de> (2011).
- [ST] Siebert, Bernd, Tian, Gang. *On the Holomorphicity of Genus two Lefschetz Fibrations*. Ann. of Math. (2) 161 (2005), no. 2, 959–1020.
- [SF] Stankova-Frenkel, Z.E. *Moduli of trigonal curves*. Algebraic Geometry 9 (2000), pag. 607-662.
- [Sti] Stilwell J. *Classical Topology and Combinatorial Group Theory*. Second Edition, Springer Verlag, 1996.
- [Wi] Wilson. *Introduction to Graph Theory*. Fourth Edition, Springer, 1993.
- [Za] Zariski O. *Algebraic Surfaces*. Second supplemented edition, Springer-Verlag, 1970.