Autobahn

Carlos Arturo Murcia Andrade

27 de febrero de 2023

Definiciones

Definición 1. *Una operación binaria se dice* ***asociativa*** *si* $\forall a, b, c \in G$ *tiene* $(a * b) * c = a * (b * c)$.

Definición 2. *Un* ***grupo*** *es un conjunto de elementos* G *cuya operación binaria "$*$" es cerrada* ($a, b \in G \to a * b \in G$) *con un elemento especial "e" (conocido como neutro) que:*

1. $\exists a^{-1} \forall a$, *tal que* $a * a^{-1} = e$
2. $b * e = e * b = b$

Además, la operación binaria es asociativa.

Definición 3. *Un* ***grafo*** *es un par* $\Gamma = (V, E)$ *donde:*

1. V *es un conjunto finito de vertices (nodos).*
2. E *es una colección de pares NO ordenados de vértices (estos son conocidos como edges o aristas).*

Definición 4. *Dos* ***grafos*** $\Gamma_1 = (V_1, E_1)$ *y* $\Gamma_2 = (V_2, E_2)$ *son* ***isomorfos*** *si existe una correspondencia* $1-1$ $\theta : \Gamma_1 \to \Gamma_2$. *Si* a *adj* b *entonces* $\theta(a)$ *adj* $\theta(b)$. *En otras palabras, existe una correspondencia entre los vértices que preservan la estructura del grafo.*

Definición 5. *Dos* ***grupos*** G *y* H *son* ***isomorfos*** *si existe una correspondencia* $1-1$ $\theta : G \to H$ *tal que* $\theta(g_1 * g_2) = \theta(g1) * \theta(g2)$ $\forall g_1, g_2 \in G$.

Definición 6. *Una **correspondencia de un conjunto** X a sí mismo es llamada una **permutación** en X (correspondencia $1-1$).*

Definición 7. *El conjunto de todas las permutaciones (correspondencias de un conjunto) es llamado el **grupo simétrico**.*
Se puede decir que las permutaciones ocultan simetrías.

Definición 8. *El **grupo de automorfismos** de un grafo $\Gamma = (V, E)$ es el conjunto de todas las permutaciones en γ que mantienen la estructura del grafo.*

Información adicional

Las capturas de pantalla que se mostrarán a continuación exhibirán la implementación de un programa (cuyos detalles, por simplicidad, están abstraídos) que permite determinar si una tabla de multiplicación representa un grupo o no (o si las operaciones definidas en la tabla de multiplicación son cerradas). Esto será de utilidad en las pruebas que se realizarán más adelante.

```
'''
Function to determine if the a given multiplication table qualifies as a group
(we call several methods already defined)
Input:
 * a 2D array -> multiplication_table
 * an array (the domain) -> table_set
Output:
 * a boolean (is the table a group?)
'''
def is_matrix_a_group(multiplication_table, table_set):
    if (is_square_matrix(multiplication_table)):
        if (is_the_table_closed_under_operation(multiplication_table, table_set) and
            is_there_the_identity_element_in_table(multiplication_table, table_set) and
            is_there_the_inverse_element_in_table(multiplication_table, table_set) and
            is_the_operation_asociative(multiplication_table, table_set)):
                return True

    return False
```

Figura 1: Función usada para determinar si una tabla de multiplicación es un grupo o no (los detalles de las funciones internas están abstraídos).

```python
'''
Function to print the basic info about this homework
'''
def print_program_info():
    print("--------------------------------------------------")
    print("Python homework:")
    print("Detect groups from multiplication tables")
    print("--------------------------------------------------")
    print("Author:")
    print("Carlos Arturo Murcia Andrade")
    print("--------------------------------------------------")
    print("\n")

    print("--------------------------------------------------")
    print("Homework description:")
    print("We will determine if sample matrices")
    print("(loaded to this program) are considered")
    print("groups or not.")
    print("We will also determine if those matrices")
    print("are latin squares or not")
    print("--------------------------------------------------")
    print("\n")
```

Figura 2: Función que explica el funcionamiento del programa y muestra su autor (el mismo autor de este documento).

1 ¿Qué es una Autobahn y para qué sirve?

Una Autobahn (grafo de red neuronal basado en automorfismos) es un tipo de grafo que usa el grupo de automorfismos de un grafo para definir redes neuronales "equivariantes" (tales que preservan las simetrías de un grafo).
Sus aplicaciones pueden ser usadas en (por ejemplo) análisis de conexiones de redes sociales (los nodos representan personas) y en descubrimiento de fármacos (los grafos representan moléculas) [TZK21].

2 ¿Por qué los autores proponen utilizar los automorfismos de grafos para reflejar las simetrías internas de un grafo?

Los automorfismos de grafos son propuestos para reflejar las simetrías internas de un grafo porque preservan las relaciones de adyacencia entre sus vértices (mediante el uso de permutaciones).
Si un subgrafo es simétrico a un automorfismo, entonces, estos son identificados como equivalentes para realizar el proceso de computación del algoritmo definido

en la Autobahn [TZK21].

3 Probar los isomorfismos sugeridos en la siguiente figura

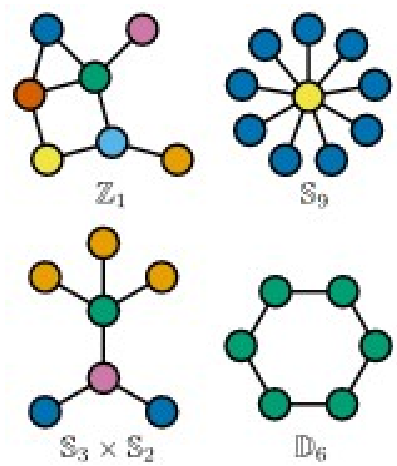

Figura 3: 4 grafos y sus grupos de automorfismos.

Los grupos de automorfismos son los siguientes:

- Z_1 : El grupo compuesto por el resultado de todos los números módulo 1. O en otras palabras, el número cero (es decir, el grupo de automorfismo tiene un único elemento "inmutable"/neutro).
- S_9: El grupo simétrico de orden 9.
- $S_3 \times S_2$: El producto cartesiano de los grupos simétrico S_3 (de orden 3) y S_2 (de orden 2).
- D_6: El grupo diedral de orden 6.

3.1 Probar que el automorfismo de grafo de la esquina superior izquierda en la figura 3 es isomorfo al grupo Z_1

Solución. Se conoce que el grupo Z_1 contiene un solo elemento, en este caso, el grafo sin alterar. Entonces, existe una sola permutación (el grafo inalterado), es decir, el grupo de automorfismos tiene un solo elemento. Ahora, este grafo será isomorfo a un grafo con las mismas características (misma cantidad de vértices, aristas, y misma estructura).

Sean $\Gamma_1 : \{[A,C],\ [A,B],\ [B,C],\ [B,D],\ [B,F],\ [C,E],\ [D,E],\ [D,G]\}$ y $\Gamma_2 :$ $\{[1,3],\ [1,2],\ [2,3],\ [2,4],\ [2,6],\ [3,5],\ [4,5],\ [4,7]\}$ grafos donde $\Gamma_2 \in Z_1$.

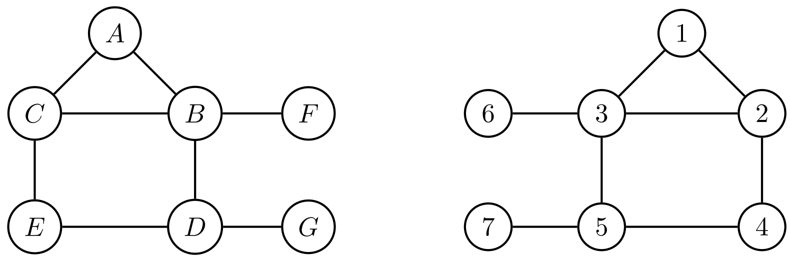

Figura 4: Representación de Γ_1 y Γ_2

Sabemos que Γ_1 y Γ_2 tienen la misma cantidad de vértices y aristas. Y si dibujamos las matrices de adyacencia, verificaremos que sus adyacencias coinciden. Por lo tanto, Γ_1 es isomorfo a Z_1.

	A	**B**	**C**	**D**	**E**	**F**	**G**	**GRADO**
A	0	1	1	0	0	0	0	2
B	1	0	1	1	0	1	0	4
C	1	1	0	0	1	0	0	3
D	0	1	0	0	1	0	1	3
E	0	0	1	1	0	0	1	3
F	0	1	0	0	0	0	0	1
G	0	0	0	1	0	0	0	1

Tabla 1: Matriz de adyacencia de Γ_1

	1	**2**	**3**	**4**	**5**	**6**	**7**	**GRADO**
1	0	1	1	0	0	0	0	2
2	1	0	0	1	0	0	0	3
3	1	1	0	0	1	1	0	4
4	0	1	0	0	1	0	0	3
5	0	0	1	1	0	0	1	3
6	0	0	1	0	0	0	0	1
7	0	0	0	0	1	0	0	1

Tabla 2: Matriz de adyacencia de Γ_2

□

3.2 Probar que el automorfismo de grafo de la esquina inferior izquierda en la figura 3 es isomorfo al grupo $S_2 \times S_3$

Solución. Es necesario definir las posibles permutaciones de $S_2 \times S_3$ (el conjunto de entrada) del siguiente modo (daremos por conveniencia valores $a - l$ a los valores del dominio):

a	[45] [123]
b	[54] [123]
c	[45] [132]
d	[54] [132]
e	[45] [213]
f	[54] [213]
g	[45] [231]
h	[54] [231]
i	[45] [312]
j	[54] [312]
k	[45] [321]
l	[54] [321]

Tabla 3: Elementos del dominio (posibles permutaciones) del conjunto $S_2 \times S_3$

Entonces, el conjunto de entrada queda definido como $S_2 \times S_3 : \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h,\ i,\ j,\ k,\ l\}$. Y su tabla de multiplicación (del grupo de automorfismos) así:

*	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
a	a	b	c	d	e	f	i	j	g	h	k	l
b	b	a	d	c	f	e	j	i	h	g	l	k
c	c	d	a	b	i	j	e	f	k	l	g	h
d	d	c	b	a	j	i	f	e	l	k	h	g
e	e	f	g	h	a	b	k	l	c	d	i	j
f	f	e	h	g	b	a	l	k	d	c	j	i
g	g	h	e	f	k	l	a	b	i	j	c	d
h	h	g	f	e	l	k	b	a	j	i	d	c
i	i	j	k	l	c	d	g	h	a	b	e	f
j	j	i	l	k	d	c	h	g	b	a	f	e
k	k	l	i	j	g	h	c	d	e	f	a	b
l	l	k	j	i	h	g	d	c	f	e	b	a

Tabla 4: Tabla de multiplicación de todas las operaciones entre $S_2 \times S_3$.

Con la tabla de multiplicación definida, se usará el programa hecho en Python (mencionado al inicio de este documento) para verificar las características de la tabla de multiplicación.

```python
# Representation of the operations between S2 x S3
sample_table_6 = [['a', 'b', 'c', 'd', 'e', 'f', 'i', 'j', 'g', 'h', 'k', 'l'],
                  ['b', 'a', 'd', 'c', 'f', 'e', 'j', 'i', 'h', 'g', 'l', 'k'],
                  ['c', 'd', 'a', 'b', 'i', 'j', 'e', 'f', 'k', 'l', 'g', 'h'],
                  ['d', 'c', 'b', 'a', 'j', 'i', 'f', 'e', 'l', 'k', 'h', 'g'],
                  ['e', 'f', 'g', 'h', 'a', 'b', 'k', 'l', 'c', 'd', 'i', 'j'],
                  ['f', 'e', 'h', 'g', 'b', 'a', 'l', 'k', 'd', 'c', 'j', 'i'],
                  ['g', 'h', 'e', 'f', 'k', 'l', 'a', 'b', 'i', 'j', 'c', 'd'],
                  ['h', 'g', 'f', 'e', 'l', 'k', 'b', 'a', 'j', 'i', 'd', 'c'],
                  ['i', 'j', 'k', 'l', 'c', 'd', 'g', 'h', 'a', 'b', 'e', 'f'],
                  ['j', 'i', 'l', 'k', 'd', 'c', 'h', 'g', 'b', 'a', 'f', 'e'],
                  ['k', 'l', 'i', 'j', 'g', 'h', 'c', 'd', 'e', 'f', 'a', 'b'],
                  ['l', 'k', 'j', 'i', 'h', 'g', 'd', 'c', 'f', 'e', 'b', 'a']]

# Representation of the set of all possible permutation of S2 x S3
table_set_6 = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l"]
```

Figura 5: Datos de entrada correspondientes a la representación de la tabla de multiplicación y el conjunto de entrada de $S_2 \times S_3$.

```
--------------------------------------------------
Sample 6 of 6:
--------------------------------------------------
Multiplication table:
['a', 'b', 'c', 'd', 'e', 'f', 'i', 'j', 'g', 'h', 'k', 'l']
['b', 'a', 'd', 'c', 'f', 'e', 'j', 'i', 'h', 'g', 'l', 'k']
['c', 'd', 'a', 'b', 'i', 'j', 'e', 'f', 'k', 'l', 'g', 'h']
['d', 'c', 'b', 'a', 'j', 'i', 'f', 'e', 'l', 'k', 'h', 'g']
['e', 'f', 'g', 'h', 'a', 'b', 'k', 'l', 'c', 'd', 'i', 'j']
['f', 'e', 'h', 'g', 'b', 'a', 'l', 'k', 'd', 'c', 'j', 'i']
['g', 'h', 'e', 'f', 'k', 'l', 'a', 'b', 'i', 'j', 'c', 'd']
['h', 'g', 'f', 'e', 'l', 'k', 'b', 'a', 'j', 'i', 'd', 'c']
['i', 'j', 'k', 'l', 'c', 'd', 'g', 'h', 'a', 'b', 'e', 'f']
['j', 'i', 'l', 'k', 'd', 'c', 'h', 'g', 'b', 'a', 'f', 'e']
['k', 'l', 'i', 'j', 'g', 'h', 'c', 'd', 'e', 'f', 'a', 'b']
['l', 'k', 'j', 'i', 'h', 'g', 'd', 'c', 'f', 'e', 'b', 'a']
--------------------------------------------------
Is this matrix a square matrix?: True
--------------------------------------------------
Is this matrix a latin square?: True
--------------------------------------------------
In this matrix closed under operation?: True
--------------------------------------------------
Is there the identity element in the matrix?: True
--------------------------------------------------
Is there the inverse element in the matrix?: True
--------------------------------------------------
Is the operation asociative?: False
--------------------------------------------------
Does this matrix represent a group?: False
--------------------------------------------------
```

Figura 6: Salida del programa que las diferentes características de la tabla de multiplicación de $S_2 \times S_3$.

Aunque la tabla de multiplicación representada por el producto cartesiano entre S_2 y S_3 no forma un grupo, su operación si es interna (todas pertenecen a $S_2 \times S_3$). Ahora bien, hay correspondencia de vértices, aristas y adyacencias (tal como se puede extrapolar de la tabla 3). Esto nos permite concluir que los automorfismos definidos en la tabla 4 son isomorfos a un grafo $S_2 \times S_3$. $\square$

3.3 Probar que el automorfismo de grafo de la esquina inferior derecha en la figura 3 es isomorfo al grupo D_6

Solución. Para comenzar, es preciso dibujar un hexágono regular (que podrá ser organizado como un grafo de 6 vértices y 6 nodos) así:

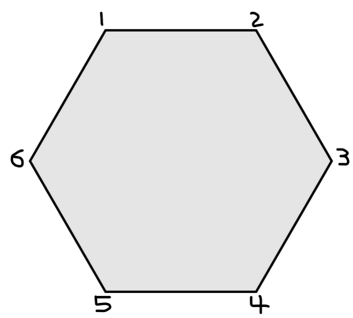

Figura 7: Representación de un hexágono regular, con sus vértices numerados de 1 a 6.

Ahora, procederemos a representar las 6 rotaciones (de 60° en 60°, desde 0° hasta 300°) y sus 6 reflexiones (el "espejo" de estas 6 rotaciones) en una tabla ("organizando" sus vértices) del siguiente modo:

Original	1	2	3	4	5	6
Original Rotated 60°	6	1	2	3	4	5
Original Rotated 120°	5	6	1	2	3	4
Original Rotated 180°	4	5	6	1	2	3
Original Rotated 240°	3	4	5	6	1	2
Original Rotated 300°	2	3	4	5	6	1
Flipped	4	5	3	2	1	6
Flipped Rotated 60°	6	4	5	3	2	1
Flipped Rotated 120°	1	6	4	5	3	2
Flipped Rotated 180°	2	1	6	4	5	3
Flipped Rotated 240°	3	2	1	6	4	5
Flipped Rotated 300°	5	3	2	1	6	4

Tabla 5: Representación numérica de todas los posibles "posicionamientos" de un hexágono regular.

Esta tabla nos permite hallar el dominio (elementos del conjunto) representativo de todas las posibles simetrías del hexágono:

$$G = \{Original,\ Original Rotated 60°,\ Original Rotated 120°,\\ Original Rotated 180°,\ Original Rotated 240°,\ Original Rotated 300°,\\ Flipped,\ Flipped Rotated 60°,\ Flipped Rotated 120°,\\ Flipped Rotated 180°,\ Flipped Rotated 240°,\ Flipped Rotated 300°\}$$

Habiendo definido el dominio, es necesario escribir (”generar”) la tabla de multiplicación:
Nota: Por conveniencia se asignan valores $a - l$ a los elementos del conjunto, tal que: $G = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h,\ i,\ j,\ k,\ l\}$

*	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
a	a	b	c	d	e	f	g	h	i	j	k	l
b	b	c	d	e	f	a	h	i	j	k	l	g
c	c	d	e	f	a	b	i	j	k	l	g	h
d	d	e	f	a	b	c	j	k	l	g	h	i
e	e	f	a	b	c	d	k	l	g	h	i	j
f	f	a	b	c	d	e	l	g	h	i	j	k
g	g	h	i	j	k	l	a	b	c	d	e	f
h	h	i	j	k	l	g	b	c	d	e	f	a
i	i	j	k	l	g	h	c	d	e	f	a	b
j	j	k	l	g	h	i	d	e	f	a	b	c
k	k	l	g	h	i	j	e	f	a	b	c	d
l	l	g	h	i	j	k	f	a	b	c	d	e

Tabla 6: Tabla de multiplicación de todas las operaciones entre simetrías del hexágono regular.

Con la tabla de multiplicación definida, se usará el programa hecho en Python (mencionado al inicio de este documento) para verificar las características de la tabla de multiplicación.
Al igual que en el caso anterior, se puede ver que todos los posibles automorfismos pertenecen al grupo D_6, adicionalmente a partir de la tabla 5 se puede inferir que hay coincidencia en el número de aristas, vértices y adyacencias. Esto nos permite concluir que los automorfismos definidos en la tabla 6 son isomorfos a un grafo D_6. $\square$

```
# Representation of symmetries of a regular hexagon
sample_table_5 = [["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l"],
                  ["b", "c", "d", "e", "f", "a", "h", "i", "j", "k", "l", "g"],
                  ["c", "d", "e", "f", "a", "b", "i", "j", "k", "l", "g", "h"],
                  ["d", "e", "f", "a", "b", "c", "j", "k", "l", "g", "h", "i"],
                  ["e", "f", "a", "b", "c", "d", "k", "l", "g", "h", "i", "j"],
                  ["f", "a", "b", "c", "d", "e", "l", "g", "h", "i", "j", "k"],
                  ["g", "h", "i", "j", "k", "l", "a", "b", "c", "d", "e", "f"],
                  ["h", "i", "j", "k", "l", "g", "b", "c", "d", "e", "f", "a"],
                  ["i", "j", "k", "l", "g", "h", "c", "d", "e", "f", "a", "b"],
                  ["j", "k", "l", "g", "h", "i", "d", "e", "f", "a", "b", "c"],
                  ["k", "l", "g", "h", "i", "j", "e", "f", "a", "b", "c", "d"],
                  ["l", "g", "h", "i", "j", "k", "f", "a", "b", "c", "d", "e"]]

# Representation of the set of all possible rotations and reflections of a regular hexagon
table_set_5 = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l"]
```

Figura 8: Datos de entrada que corresponden a la tabla de multiplicación y conjunto de entrada D_6.

3.4 Probar que el automorfismo de grafo de la esquina superior derecha en la figura 3 es isomorfo al grupo S_9

Solución. Sabemos que todas las posibles simetrías de un grafo S_9 son de grado 9! (es decir, existen 9! simetrías). Ahora, también es cierto decir que cualquier simetría tendrá un nodo central y 9 nodos "orbitantes" conectados a dicho nodo central (de ahí la forma de "estrella" de todos los automorfismos). Entonces, todo automorfismo tendrá la misma configuración de nodos y vértices (10 y 9 respectivamente), y, también, la estructura del grafo se mantendrá (es decir, tendrá la misma forma de "estrella" con un nodo central y 9 "orbitantes"). Por ello, se puede concluir que los grafos son isomorfos al grupo S_9. $\square$

```
--------------------------------------------------
Sample 5 of 6:
--------------------------------------------------
Multiplication table:
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']
['b', 'c', 'd', 'e', 'f', 'a', 'h', 'i', 'j', 'k', 'l', 'g']
['c', 'd', 'e', 'f', 'a', 'b', 'i', 'j', 'k', 'l', 'g', 'h']
['d', 'e', 'f', 'a', 'b', 'c', 'j', 'k', 'l', 'g', 'h', 'i']
['e', 'f', 'a', 'b', 'c', 'd', 'k', 'l', 'g', 'h', 'i', 'j']
['f', 'a', 'b', 'c', 'd', 'e', 'l', 'g', 'h', 'i', 'j', 'k']
['g', 'h', 'i', 'j', 'k', 'l', 'a', 'b', 'c', 'd', 'e', 'f']
['h', 'i', 'j', 'k', 'l', 'g', 'b', 'c', 'd', 'e', 'f', 'a']
['i', 'j', 'k', 'l', 'g', 'h', 'c', 'd', 'e', 'f', 'a', 'b']
['j', 'k', 'l', 'g', 'h', 'i', 'd', 'e', 'f', 'a', 'b', 'c']
['k', 'l', 'g', 'h', 'i', 'j', 'e', 'f', 'a', 'b', 'c', 'd']
['l', 'g', 'h', 'i', 'j', 'k', 'f', 'a', 'b', 'c', 'd', 'e']
--------------------------------------------------
Is this matrix a square matrix?: True
--------------------------------------------------
Is this matrix a latin square?: True
--------------------------------------------------
In this matrix closed under operation?: True
--------------------------------------------------
Is there the identity element in the matrix?: True
--------------------------------------------------
Is there the inverse element in the matrix?: True
--------------------------------------------------
Is the operation asociative?: True
--------------------------------------------------
Does this matrix represent a group?: True
--------------------------------------------------
```

Figura 9: Salida del programa que muestra que la matriz de multiplicación generada por las operaciones entre las posibles simetrías de un hexágono regular, forman un grupo.

4 Explique en que consiste la siguiente figura ¿Cuál es su relación con el grupo de automorfismos de D_6?

La figura anterior explica el algoritmo de la Autobahn aplicado a un grafo cíclico. Se relaciona con el grupo de automorfismos D_6 en el hecho de que este algoritmo solo puede ser aplicado en grupos de automorfismos, y en el hecho de que el grafo resultante es parte del grupo de automorfismos D_6.

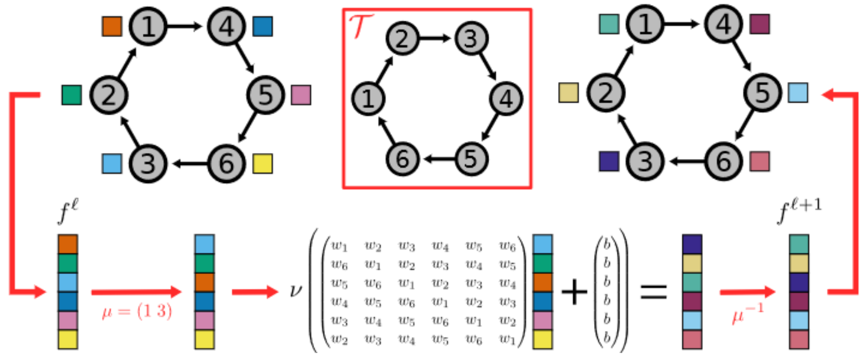

Figura 10: Algoritmo de la Autobahn aplicado a un grafo ciclico.

Referencias

[TZK21] Erik H Thiede, Wenda Zhou y Risi Kondor. *Autobahn: Automorphism-based graph neural nets*. 2021. URL: `https://ehthiede.github.io/pdfs/autobahn.pdf`.