

1. Introducción

La función `definirManiobra` tiene como objetivo reordenar un tren inicial t_1 en un tren objetivo t_2 , utilizando una estación de maniobras con tres vías: una vía principal (que contiene inicialmente el tren t_1), una vía auxiliar uno, y una vía auxiliar dos.

Para ello, se genera una secuencia de movimientos (Maniobra) que representa las operaciones necesarias para convertir t_1 en t_2 . Cada movimiento se modela como una instancia de `Movimiento`, del tipo `Uno(n)` o `Dos(n)`.

2. Componentes del Sistema

- **Tren:** Representado como una lista (`List`) de elementos tipo `Vagon`.
- **Movimiento:** `Uno(n)` o `Dos(n)`, moviendo elementos entre vías.
- **Estado:** Tupla (`actual, objetivo, uno, dos`), representando las vías en cada paso.

3. Descripción del Algoritmo `definirManiobra`

La función principal `definirManiobra` contiene una subfunción recursiva llamada `construir`, la cual implementa la lógica central del algoritmo mediante correspondencias de patrones.

El proceso se divide en:

1. Si los trenes son iguales \rightarrow no se requieren movimientos.
2. Si los trenes son reversos \rightarrow se retorna una maniobra fija.
3. Caso general \rightarrow se usa la función `construir` recursivamente.

4. Funcionamiento de `construir`

Se consideran tres casos:

- **Caso base:** ambos trenes vacíos \rightarrow devolver movimientos acumulados.
- **Coincidencia directa:** vagones coinciden en orden \rightarrow se avanza.
- **Caso general:** se prueba mover con `Uno(1)`, `Uno(-1)`, `Dos(-1)`, o `Dos(n)`, actualizando el estado y recursando.

5. Modelo Matemático

Sea un tren una secuencia $T = \langle v_1, v_2, \dots, v_n \rangle$. Sean T1 el tren inicial y T2 el objetivo. Queremos encontrar una secuencia de movimientos:

$$M = \langle m_1, m_2, \dots, m_k \rangle$$

tal que:

$$\text{apply}(M, T_1) = T_2$$

Estado del sistema:

$S = (A, O, U, D)$

- A: vía principal (actual)
- O: parte restante del objetivo
- U, D : vías auxiliares uno y dos

Transición de estados:

$$\delta(S, m) \rightarrow S' \quad \text{aplicando un movimiento } m$$

Algoritmo recursivo:

$$\text{construir}(S, M) = \begin{cases} M^{-1} & \text{si } A = [] \wedge O = [] \\ \text{construir}(S', m \cdot M) & \text{si existe transición válida} \\ \text{vacío} & \text{si no hay transición posible} \end{cases}$$

6. Ejemplo Ilustrativo

- **Inicial:** [1, 3, 2]
- **Objetivo:** [1, 2, 3]

Movimientos esperados:

```
Uno(1)    // mueve 2 a vía uno
Uno(-1)   // devuelve 2 a principal
Uno(1)    // mueve 3 a vía uno
Uno(-1)   // devuelve 3 a principal
```

7. Conclusión

El algoritmo `definirManiobra` encuentra una serie de movimientos que reordenan un tren utilizando vías auxiliares. Su estructura recursiva basada en coincidencias y reglas de prioridad garantiza que, si una solución es posible, será encontrada. Además, su modelado matemático permite un análisis riguroso y formal del proceso.