

1. Instrucciones del algoritmo de Deconstrucción Conceptual Programada

Fase 1: Entrada de Datos

1. Definir el concepto inicial $C_{i,j,k}$.

Ejemplo: "Antigua Grecia".

- i : Nivel de profundidad del concepto.
- j : Numeración del concepto dentro del nivel i .
- k : Valor j del concepto del que proviene. Si es el concepto inicial, entonces $k = 0$.

2. Establecer un nivel de profundidad máximo N :

Determina hasta qué nivel se subdividirá el concepto inicial.

Ejemplo: $N = 3$

3. Definir el número máximo de subconceptos por concepto M :

Limita la cantidad de subconceptos generados en cada deconstrucción

Ejemplo: $M = 5$

4. Inicializar las listas y estructuras de datos necesarias

Lista de conceptos pendientes: C_p .

Lista de conceptos analizados: C_a .

Waitlist: W .

Lista de no retorno: N_r .

Lista de Quarks Conceptuales: Q_C .

Lista de Bucles Conceptuales: B_c .

Ejemplo de inicialización: $C_p = [\text{Antigua Grecia}_{0.1.0}]$, $C_a = []$, $W = []$, $N_r = []$, $Q_C = []$, $B_c = []$.

Fase 2: Procesamiento

1. Iteración de deconstrucción:

Para cada concepto $C_{i,j,k}$ en la lista de conceptos pendientes:

1. Descomponer $C_{i,j,k}$ en subconceptos $[C_{i.1.k}, C_{i.2.k}, \dots, C_{i.j.k}]$ sin que $j > M$.

- Donde ' k ' es el valor ' j ' del concepto del que proviene y ' j ' es la numeración de cada concepto en este nivel i de profundidad.
- Insertar todos los nuevos conceptos en la waitlist.

Ejemplo: $W = [\text{Historia}_{1.1.1}, \text{Filosofía}_{1.2.1}, \text{Política}_{1.3.1}, \text{Arte}_{1.4.1}, \dots, \text{Mitología}_{1.j.1}]$

- Si no se puede deconstruir en subconceptos relevantes, se debe de insertar $C_{i,j,k}$ en la lista de no retorno y en la lista de Q_C .

2. Añadir C_{ijk} a la lista de conceptos analizados

Ejemplo: $C_p = []$, $C_a = [\text{Antigua Grecia}_{0.1.0}]$, $W = [\text{Historia}_{1.1.1}, \text{Filosofía}_{1.2.1}, \text{Política}_{1.3.1}, \text{Arte}_{1.4.1}, \dots, \text{Mitología}_{1.j.1}]$, $N_r = []$, $Q_C = []$

3. Verificar profundidad:

- Si el valor de 'i' es menor que N:

Trasladar los conceptos que se encuentren dentro de la waitlist que no estén en la lista de no return a la lista de conceptos pendientes.

Ejemplo: Cp = [Historia_1.1.1, Filosofía_1.2.1, Política_1.3.1, Arte_1.4.1, ..., Mitología_1.j.1], Ca = [Antigua Grecia_0.1.0], W = [], Nr = [], QC = []

- Si el nivel de profundidad es igual a N:

Eliminar los subconceptos dentro de la waitlist

Ejemplo: Cp = [], Ca = [Antigua Grecia_0.1.0], W = [], Nr = [], QC = []

4. Verificar bucles:

- Si dentro de Cp un concepto C_{i,j,k} representa el mismo concepto C que uno de los que se encuentren dentro de Ca:

Ejemplo: Cp = [Antigua Grecia_4.1.1], Ca = [Antigua Grecia_0.1.0, Historia_1.1.1, Filosofía_2.1.1, Sócrates_3.1.1]

Comprobar si ambos conceptos están ligados empleando el algoritmo de Cierre Conceptual:

1. Crea una lista de bucle temporal: Bt = []
2. Selecciona el concepto que se repite C_{i,j,k} dentro de la lista Cp y añádelo a Bt.
Ejemplo: Bt = [Antigua Grecia_4.1.1]
3. Si el concepto añadido es C_{i,j,k}, añade a la lista el concepto con: i = i-1; j = k;
Ejemplo: Bt = {Antigua Grecia_4.1.1, Sócrates_3.1.1}
4. Repite hasta i = 0 o hasta que C_{i,j,k} = C_{i₂,j₂,k₂}:
 - Si llegas al nivel de profundidad 0 y no has encontrado un C_{i₂,j₂,k₂} con el mismo valor C, no existe un bucle conceptual.
 - Si llegas a un concepto C_{i₂,j₂,k₂} con el mismo valor C, has encontrado un bucle conceptual.

Ejemplo de bucle conceptual: Bt = [{Antigua Grecia_0.1.0, Historia_1.1.1, Filosofía_2.1.1, Sócrates_3.1.1, Antigua Grecia_4.1.1}]

- Si existe un bucle conceptual, se debe insertar cada elemento de dicho bucle dentro de una lista dentro de la lista QC, y se debe de añadir también a la lista de no return el elemento dentro de Bt con 'i' de mayor valor.

Ejemplo: Nr = [Antigua Grecia_4.1.1], QC = [{Antigua Grecia_0.1.0, Historia_1.1.1, Filosofía_2.1.1, Sócrates_3.1.1, Antigua Grecia_4.1.1}]

Fase 3: Output

Se debe generar un informe con los siguientes elementos:

Lista de conceptos analizados: Ca.

Lista de Quarks Conceptuales: QC