Curso 2025-26 Práctica 1. Búsqueda Inteligente. Satisfacción de restricciones. Sesión 3

Forward Checking (FC)

- En cada etapa de la búsqueda, FC comprueba hacia delante la asignación actual con todos los valores de las futuras variables que están restringidas con la variable actual.
- Los valores de las variables futuras que son inconsistentes con la asignación actual son temporalmente eliminados de sus dominios.
- Si el dominio de una variable futura se queda vacío, la instanciación de la variable actual se deshace y se prueba con un nuevo valor. Si ningún valor es consistente, entonces se lleva a cabo el backtracking cronologico.

FC

Pseudocódigo intuitivo:

- 1. Selectionar x_i .
- 2. Instanciar $x_i \leftarrow a_i : a_i \in D_i$.
- 3. Razonar hacia adelante (forward-check):
 - Eliminar de los dominios de las variables (x_{i+1}, \ldots, x_n) aún no instanciadas, aquellos valores inconsistentes con respecto a la instanciación (x_i, a_i) , de acuerdo al conjunto de restricciones.
- 4. Si quedan valores posibles en los dominios de todas las variables por instanciar, entonces:
 - Si i < n, incrementar i, e ir al paso (1).
 - Si i = n, salir con la solución.
- 5. Si existe una variable por instanciar, sin valores posibles en su dominio, entonces retractar los efectos de la asignación $x_i \leftarrow a_i$. Hacer:
 - Si quedan valores por intentar en D_i , ir al paso (2).
 - Si no quedan valores:
 - Si i > 1, decrementar i y volver al paso (2).
 - Si i = 1, salir sin solución.

Universidad de Alicante Universitat d'Alacant

Ejemplo de FC

- Configuración inicial:
 - Variables:
 - Dominios: $D_x = D_v = \{1, 2, 3, 4, 5\}$
 - Restricción: x < y - 1
- Inicialmente:
 - $CD_x = CD_v = \{1, 2, 3, 4, 5\}$
- Proceso:
 - **Caso 1**: Si asignamos x = 2
 - Los únicos valores que puede tomar y son 4, 5
 - $CD_{v} = \{4,5\}$ • Por tanto:
 - Caso 2: Asignamos x = 4
 - No hay asignación posible compatible con la restricción
 - Por tanto:
 - Por tanto: $CD_y = \{\}$ Acción: Deshacer x = 4 y realizar backtracking

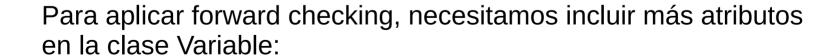
FC

Algoritmo

```
Algoritmo Forward Checking
 1: function FC(i : variable)
                                                             ⊳ Retorna booleano
       for all a \in dominio[i] do
           X_i \leftarrow a
 3:
           if i = N then return VERDADERO
                                                     \triangleright; \leftarrow todas variables ?
           else
              if FORWARD(i, a) then
                  if FC(i+1) = VERDADERO then return VERDADERO
 7:
              RESTAURAR(i)
 8:
       return Falso
 9:
10: function FORWARD(i: variable, a: valor)
                                                             ▷ Retorna booleano
       dominio \ vacio \leftarrow Falso
11:
       for i \leftarrow i + 1 to N do
12:
           for all b \in dominio[i] do
13:
              if not ES CONSISTENTE(i, a, j, b) then
14:
                                                               22: procedure RESTAURAR(i: variable)
                  Eliminar b de dominio[j]
15:
                                                                      for i \leftarrow i + 1 to N do
                                                               23:
                  Añadir b a podado[j]
16:
                                                                         for all b \in podado[j] do
                                                               24:
           if dominio[j] está vacío then
17:
                                                                            if X_i es responsable del filtrado de b then
                                                               25:
                                                                                Eliminar b de podado[j]
              dominio \ vacio \leftarrow Verdadero
                                                               26:
18:
                                                                                Añadir b a dominio[j]
                                                               27:
              break
19:
                                                               28:
       if dominio vacio then return FALSO
20:
                                                               29: function ES_CONSISTENTE(i: variable, a: valor, j: variable, b: valor) \triangleright
       return Verdadero
                                                                  Retorna booleano
                                                                                          \triangleright; 'a' en X_i es consistente con 'b' en X_i?
                                                               30:
                                                                                                                  ▷ Depende del problema
                                                               31:
```

✓ Clase Variable

- Representa cada casilla del sudoku.
- Atributos clave:
 - fila, columna: posición en el tablero.
 - valor: número asignado (o None si está vacía).
 - dominio: lista de valores posibles (1-9, reducida por restricciones).
 - fijada: indica si la casilla es parte de la plantilla inicial.
- Permite:
 - Asignar y desasignar valores.
 - Actualizar y restaurar dominios durante el backtracking.
- **Función:** Modelar cada celda como una variable del CSP, con su estado y opciones válidas.



En cada variable debemos poder controlar:

Podados: Una lista de valores que han sido eliminados temporalmente del dominio de esta variable durante el proceso de Forward Checking. Sirve para poder restaurarlos más tarde si se hace backtracking.

Causas: Una lista de referencias a las variables (o sus posiciones) que causaron la eliminación (poda) de valores en esta variable. Sirve para saber por qué se eliminó cada valor. Esto es útil si quieres restaurar solo lo que eliminó una variable específica

Universitat d'Alacant Universidad de Alicante

Tareas de la sesión 3 de la Práctica 1

- Modificar la clase Variable añadiendo los atributos necesarios para poder implementar Forward Checking en la resolución de sudokus.
- Empezar la implementación de Forward Checking