Alejandro García Gil 77042008N

PRÁCTICA 1: RESOLUCIÓN DE CRUCIGRAMAS MEDIANTE FC Y AC3

Explicación de los algoritmos usados

-Clases creadas para los algoritmos:

-Variable:

Atributos->
- tam= tamaño de la variable
- posIni= posición inicial de la variable
- posFin= posición final de la variable
- dom= dominio inicial de la variable
- podas= lista donde guardamos las podas que realizamos
- orientacion= orientación de la variable (h/v)
- valorActual= valor de la variable
- nombre= nombre de la variable
- restriccion= lista donde guardamos las restricciones que se hacen

Métodos->
- Getters: getDominio(), getRestrricciones(), getValorActual(), getPodas(), getTam().
- Setters: setNewDominio(), setValorActual(), setDom().
- primeraPos: (getter) devuelve el valor de la posición inicial de la variable.
- ultimaPos: (getter) devuelve el valor de la posición final de la variable.
- impVariableAux: imprime información formateada sobre la instancia de la clase, incluyendo el nombre, la posición actual, el tamaño, las posiciones inicial y final, la lista obtenida a través del método getLista() del atributo dom, y la orientación.
- impVariable: imprime información formateada sobre la instancia de la clase, especificando si la variable es de tipo horizontal o vertical, y mostrando el nombre, la posición inicial (coordenadas x e y) y el dominio.
- addDominio: añade el dominio a la instancia de la clase.
- borrarDom: borra un valor específico del dominio (val) y registra la eliminación junto con una causa asociada en una lista llamada podas.
- addRestriccion: añade una restricción a la instancia de la clase.
- printRestriccion: imprime la información sobre la restricción en la lista de restricciones nombrada anteriormente.
- deletePoda: elimina las entradas en la lista de podas donde el primer elemento de la tupla sea igual a la variable proporcionada. Se emplea posteriormente en el AC3.

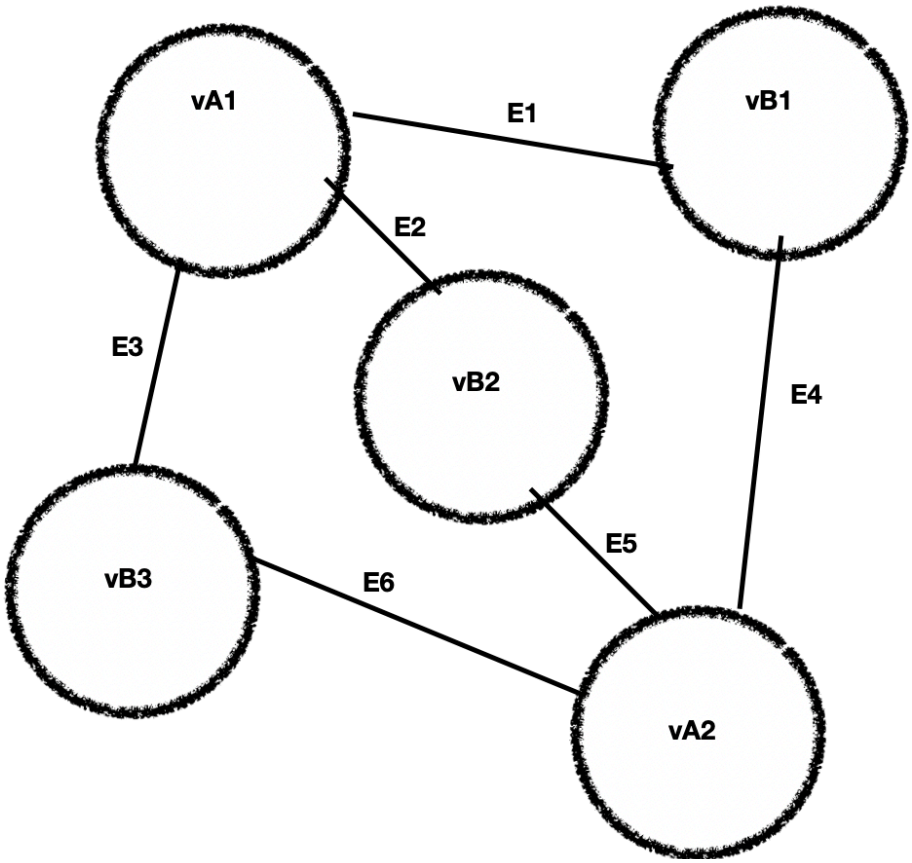
-Restricción:

Atributos->
- fil: fila de la restricción de la variable.
- col: columna de la restricción de la variable.
- v1: primera variable que tiene restricción
- v2: segunda variable que tiene restricción

Métodos->
- Getters: getV1(), getV2(), getPosRestV1() (devuelve la posición de la primera variable con restricción), getPosRestV2() (devuelve la posición de la segunda variable con restricción), getFil(), getCol(), getPos() (tupla de col y fil).
- printRest: imprime información formateada sobre una restricción, incluyendo los nombres de dos variables (v1 y v2), y dos atributos numéricos (fil y col).

FORWARD CHECKING

1. Define una función fc con los parámetros ‘variables’ y ‘guia’



2. Se comprueba que ‘guia’ es igual a la longitud de la lista de las variables. En caso de serlo, habrá alcanzado el final de la lista y devolverá True e indicará que se ha encontrado una asignación correcta para todas las variables.
3. En el primer bucle for que encontramos, iteramos sobre una copia del dominio de la variable en la posición ‘guia’ de la lista de variables y, poder así evitar algún problema si el dominio cambia a lo largo de la ejecución del programa.
4. Posteriormente, asignamos el valor ‘i’ al atributo del valor actual de la variable en la posición ‘guia’.
5. Llamamos a la función del **forward** (método) para realizar toda la propagación de las restricciones. Además, hacemos una llamada recursiva a la función fc, llamándose a sí misma con la siguiente variable de la lista. Si dicha llamada devuelve True, se habrá encontrado una asignación válida para todas las variables y la función actual también devolverá True.
6. Lo siguiente es llamar a la función change(variables, guia) que revierte algunos cambios realizados en las variables durante la propagación en curso de las restricciones.
7. Por último, restablecemos el valor actual de la variable en la posición ‘guia’ a None. La función llega a este punto si no se encuentra una asignación válida para la variable actual, por lo tanto vuelve a False.

AC3

Este método inicia un bucle mientras haya restricciones en la lista. Toma la primera restricción de la lista y la asigna a la variable ‘rest’, eliminándola de la lista. Además, inicializa ‘cambio’ como False y, poder así confirmar si se hizo algún cambio en el dominio de las variables.
En el primer for, de la primera variable (v1), iteramos sobre el dominio de la variable V1 de la restricción en la que estamos. En la comprobación if not fixed, realiza una verificación si el valor ‘i’ en el dominio v1 es consistente con el dominio de la segunda variable, v2, según la restricción. Cuando ‘cambio’ se establece como True, indica que algún cambio se ha hecho en el dominio.

A continuación, verifica si el dominio de v1 está vacío una vez se hayan hecho las eliminaciones y, si está vacío, devolverá False, indicando así que no hay una asignación válida para la variable. Ahora realiza de nuevo una comprobación por si hay algún cambio en el dominio y, para cada restricción de v1, añade las restricciones a la lista de restricciones, indicando True si la propagación de dichas restricciones se ha realizado correctamente.
Finalmente, en los métodos auxiliares que complementan el AC3 (AC3Aux y AC3Print), el primero es una función auxiliar que se utiliza en el proceso de la propagación de las restricciones y, el segundo, imprime la información sobre las variables antes y después de aplicar el algoritmo de AC3.

Grafo de restricciones
El tamaño del mapa para las restricciones será de 2x3 con el siguiente diccionario inventado:
Vocabulario inventado: {pan lal tim uno te po le tu in mo }

Variables:
V={vA1, vA2, vB1, vB2, vB3}
Como dominios tenemos:
Dj={te, po, le, tu, in, mo}$\forall j$, $3 \leq j \leq 5$
Di= {pan, lal, tim, uno}, $\forall i$, $1 \leq i \leq 2$
Aristas:
E= {e1, e2, e3, e4, e5, e6}
Y las conexiones son:
c(e1) = <vA1, vB1>, c(e2) = <vA1, vB2>, c(e3) = <vA1, vB3>, c(e4) = <vA2, vB1>, c(e5) = <vA2, vB2>, c(e6) = <vA2, vB3>
Restricciones:
I(e1) = {<pan, po>, <lal, le>, <tim, te>, <tim, tu>}, I(e2) = {<tim, in>}, I(e3) = {<tim, mo>}, I(e4) = {<uno, tu>}, I(e5) = {<uno, in>} I(e6) = {<pan, in}, <uno, mo>}

Traza del Forward Checking para el ejemplo recientemente hecho
El tablero tendrá el tamaño 2x3 y una vez generadas las 5 variables queda de la siguiente forma.
vA1:posicion=[(0,0), horizontal], tamanyo=3, dominio=[pan, lal, tim, uno]
vA2:posicion=[(1,0), horizontal], tamanyo=3, dominio=[pan, lal, tim, uno]
vB1:posicion=[(0,0), vertical], tamanyo=2, dominio=[te, po, le, tu, in, mo]
vB2:posicion=[(0,1), vertical], tamanyo=2, dominio=[te, po, le, tu, in, mo]
vB3:posicion=[(0,2), vertical], tamanyo=2, dominio=[te, po, le, tu, in, mo]

A continuación, mediante las siguientes iteraciones generarán 6 restricciones.

- Restriccion1:(vA1, vB1)

- Restriccion2 = (vA1, vB2)

- Restriccion3 = (vA1, vB3)

- Restriccion4 = (vA2, vB1)

- Restriccion5 = (vA2, vB2)

- Restriccion6 = (vA2, vB3)

En la primera iteración elegimos la variable vA1 y le asignamos la palabra “pan”.

Ahora, se recorrerá todas las restricciones de vB1 para borrar dicha palabra.

- Restriccion1: se elimina [te, le, tu, in ,mo] del dominio porque su primera letra no es P

- Restriccion2: eliminamos [[te, le, tu, in ,mo] del dominio de vB2 porque su primera letra no es A

- Restriccion3: eliminamos [[te, le, tu, in ,mo] del dominio de vB3 porque su primera letra no es N

- Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vH2: [lal, tim, uno]
  vV1: [po]
  vV2: []
  vV3: []
  Algún dominio se ha quedado vacío, así que:

- Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA1: [lal, tim ,uno]
  vA2: [pan, lal, tim, uno]

vB1: [te, le, tu in, mo, po]

vB2: [te, le, tu, in, mo, po]

vB3: [te, le, tu, in, mo ,po]

Como el dominio de la variable actual no está vacío, seguimos con esa variable para probar el siguiente valor de su dominio.

En la segunda iteración haremos lo siguiente:

Variable seleccionada: vA1

Dominios de las variables no asignadas al inicio:

vA1: [lal, tim , uno]

vA2: [pan, lal, tim, uno]

vB1: [te, le, tu, in, mo, po]

vB2: [te, le, tu, in, mo, po]

vB3: [te, le, tu, in, mo, po]

Asignamos “lal” a vA1:

Recorremos todas las variables sin asignar y eliminamos lal para no permitir palabras repetidas: Ningún dominio queda vacío.

Recorremos restricciones de vA1:

- Restriccion1: eliminamos [le] del dominio de vB1 porque su primera letra no es L

- Restriccion2: eliminamos [[te, le, tu, in ,mo] del dominio de vB2 porque su primera letra no es A

- - Restriccion3: eliminamos [LE] del dominio de vB3 porque su primera letra no es L
  Algún dominio se ha quedado vacío, así que:
    - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
    - Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vH1: [TIM, UNO]
  vA2: [pan, lal, tim, uno]
  vB1: [[te, le, tu, in ,mo, po]
  vB2: [[te, le, tu, in ,mo, po] vB3: [[te, le, tu, in ,mo, po]
  Como el dominio de la variable actual no está vacío, seguimos con esa variable para probar el siguiente valor de su dominio.
  Tercera iteración:
  Variable seleccionada: vA1
  Dominios de las variables no asignadas al inicio: vH1: [TIM, UNO]
  vH2: [PAN, LAL, TIM, UNO]
  vV1: [TE, LE, TU, IN, MO, PO]

vV2: [TE, LE, TU, IN, MO, PO]

vV3: [TE, LE, TU, IN, MO, PO]

Asignamos TIM a la variable seleccionada:

Recorremos todas las variables sin asignar y eliminamos TIM para no permitir palabras repetidas: Ningún dominio queda vacío.

Recorremos restricciones de vH1:

- Restriccion1: eliminamos [LE, IN, MO, PO] del dominio de vB1 porque su primera letra no es T
  - Restriccion2: eliminamos [TE, LE, TU, MO, PO] del dominio de vB2 porque su primera letra no es I
  - Restriccion3: eliminamos [TE, LE, TU, IN, PO] del dominio de vB3 porque su primera letra no es M
  Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vA2: [PAN, LAL, UNO]
  vB1: [TE, TU]
  vB2: [IN]
  vB3: [MO]
  Ningún dominio ha quedado vacío, así que pasamos a la siguiente variable.
  Cuarta iteración
  Variable seleccionada: vA2
  Asignamos PAN a la variable seleccionada
  Recorremos todas las variables sin asignar y eliminamos PAN para no permitir palabras repetidas: Ningún dominio queda vacío.
  Recorremos restricciones de vA2:
  - Restriccion4: eliminamos [TE, TU] del dominio de vB1 porque su segunda letra no es P
  - Restriccion5: eliminamos [IN] del dominio de vB2 porque su segunda letra no es A
  - Restriccion6: eliminamos [MO] del dominio de vB3 porque su segunda letra no es N
  Algún dominio se ha quedado vacío, así que:

- Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra

- Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA2: [LAL, UNO]
  vB1: [TE, TU] vB2: [IN] vB3: [MO]

Como el dominio de la variable actual no está vacío, seguimos con esa variable para probar el siguiente valor de su dominio.

Quinta iteración:

Variable seleccionada: vA2

Asignamos LAL a la variable seleccionada

Recorremos todas las variables sin asignar y eliminamos LAL para no permitir palabras repetidas: Ningún dominio queda vacío.

Recorremos restricciones de va2:

- Restriccion4: eliminamos [TE, TU] del dominio de vB1 porque su segunda letra no es L

- Restriccion5: eliminamos [IN] del dominio de vB2 porque su segunda letra no es A

- Restriccion6: eliminamos [MO] del dominio de vB3 porque su segunda letra no es L
  Algún dominio se ha quedado vacío, así que:
  - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  - Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA2: [UNO]
  vB1: [TE, TU]
  vB2: [IN]
  vB3: [MO]
  Como el dominio de la variable actual no está vacío, seguimos con esa variable para probar el siguiente valor de su dominio.
  Sexta iteración:
  Variable seleccionada: vA2
  Asignamos UNO a la variable seleccionada
  Recorremos todas las variables sin asignar y eliminamos UNO para no permitir palabras repetidas: Ningún dominio queda vacío.
  Recorremos restricciones de vA2:
  - Restriccion4: eliminamos [TE] del dominio de vB1 porque su segunda letra no es U
  - Restriccion5: eliminamos [] del dominio de vB2 porque su segunda letra no es N
  - Restriccion6: eliminamos [] del dominio de vB3 porque su segunda letra no es O
  Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vB1: [TU]
  vB2: [IN]
  vB3: [MO]
  Ningún dominio ha quedado vacío, así que pasamos a la siguiente variable.

Séptima iteración:

- Variable seleccionada: vB1
  Asignamos TU a la variable seleccionada
  La variable seleccionada carece de restricciones.
  Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vB2: [IN]
  vB3: [MO]
  Octava iteración:
  Variable seleccionada: vB2
  Asignamos IN a la variable seleccionada
  La variable seleccionada carece de restricciones.
  Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vB3: [MO]
  ITERACIÓN 9:
  Variable seleccionada: vB3
  Asignamos MO a la variable seleccionada
  La variable seleccionada carece de restricciones.
  Como todas las variables ya tienen valor asignado podemos dar por terminado el algoritmo

Traza del algoritmo de AC3 para el ejemplo anterior

Dimensiones del tablero: 2x3
Casillas llenas en coordenadas: []
Casillas con letras en coordenadas: []
Generadas 5 variables:
vA1: pos = [(0, 0), horizontal], tam = 3, dom = [PAN, LAL TIM, UNO]
vA2: pos = [(1, 0), horizontal], tam = 3, dom = [PAN, LAL, TIM, UNO]
vB1: pos = [(0, 0), vertical], tam = 2, dom = [TE, PO, LE, TU, IN, MO]
vB2: pos = [(0, 1), vertical], tam = 2, dom = [TE, PO, LE, TU, IN, MO]
vB3: pos = [(0, 2), vertical], tam = 2, dom = [TE, PO, LE, TU, IN, MO]
Generadas 6 restricciones:
Restriccion1 = (vA1, vB1)

Restriccion2 = (vA1, vB2)

Restriccion3 = (vA1, vB3)

Restriccion4 = (vA2, vB1)

Restriccion5 = (vA2, vB2)

Restriccion6 = (vH2, vV3)

Primera iteración:
Variable seleccionada: vA1
Asignamos PAN a la variable seleccionada.
Recorremos restricciones de vA1:

- Restriccion1: eliminamos [TE, LE, TU, IN, MO] del dominio de vB1 porque su primera letra no es P
  - Restriccion2: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB2 porque su primera letra no es A
  - Restriccion3: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB3 porque su primera letra no es N
  Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
  vA2: [LAL, TIM, UNO]
  vB1: [PO]
  vB2: []
  vB3: []
  Algún dominio se ha quedado vacío, así que:
  - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA1: [LAL, TIM, UNO]
  vA2: [PAN, LAL, TIM, UNO]
  vB1: [TE, LE, TU, IN, MO, PO]
  vB2: [TE, LE, TU, IN, MO, PO]

ITERACIÓN 2:
Variable seleccionada: vH1
Dominios de las variables no asignadas al inicio: vH1: [LAL, TIM, UNO]
vH2: [PAN, LAL, TIM, UNO]
vV1: [TE, LE, TU, IN, MO, PO]
vV2: [TE, LE, TU, IN, MO, PO]
vV3: [TE, LE, TU, IN, MO, PO]
Asignamos LAL a la variable seleccionada. Recorremos restricciones de vH1:

- Restriccion1: eliminamos [LE] del dominio de vB1 porque su primera letra no es L
  - Restriccion2: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB2 porque su primera letra no es A
  - Restriccion3: eliminamos [LE] del dominio de vB3 porque su primera letra no es L

Algún dominio se ha quedado vacío, así que:
- Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
- Restauramos los dominios afectados.

Dominios de las variables no asignadas después de restaurar: vA1: [TIM, UNO]
vA2: [PAN, LAL, TIM, UNO]
vB1: [TE, LE, TU, IN, MO, PO]
vB2: [TE, LE, TU, IN, MO, PO]
vB3: [TE, LE, TU, IN, MO, PO]
Tercera iteración:
Variable seleccionada: vA1
Dominios de las variables no asignadas al inicio: vA1: [TIM, UNO]
vA2: [PAN, LAL, TIM, UNO]
vB1: [TE, LE, TU, IN, MO, PO]
vB2: [TE, LE, TU, IN, MO, PO]
vB3: [TE, LE, TU, IN, MO, PO]
Asignamos TIM a la variable seleccionada. Recorremos restricciones de vA1:
- Restriccion1: eliminamos [LE, IN, MO, PO] del dominio de vB1 porque su primera letra no es T
- Restriccion2: eliminamos [TE, LE, TU, MO, PO] del dominio de vB2 porque su primera letra no es I
- Restriccion3: eliminamos [TE, LE, TU, IN, PO] del dominio de vB3 porque su primera letra no es M

Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
vA2: [PAN, LAL, TIM, UNO]
vB1: [TE, TU]
vB2: [IN]
vB3: [MO]
Como ninguna variable se ha quedado con el dominio vacío recuperamos las palabras y procedemos a probar con la siguiente palabra.
Cuarta iteración:
Variable seleccionada: vA1
Dominios de las variables no asignadas al inicio: vA1: [TIM, UNO]
vA2: [PAN, LAL, TIM, UNO]
vB1: [TE, LE, TU, IN, MO, PO]
vB2: [TE, LE, TU, IN, MO, PO]
vB3: [TE, LE, TU, IN, MO, PO]
Asignamos UNO a la variable seleccionada. Recorremos restricciones de vA1:
- Restriccion1: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB1 porque su primera letra no es U
- r2: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB2 porque su primera letra no es N
- r3: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB3 porque su primera letra no es O

Dominios de las variables no asignadas después de la asignación de la variable seleccionada:
vA2: [PAN, LAL, TIM, UNO]
vB1: []
vB2: []
vB3: []
Algún dominio se ha quedado vacío, así que:
- Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
- Restauramos los dominios afectados.

Dominios de las variables no asignadas después de restaurar: vA1: [TIM]
vA2: [PAN, LAL, TIM, UNO]
vB1: [TE, LE, TU, IN, MO, PO]
vB2: [TE, LE, TU, IN, MO, PO]
vB3: [TE, LE, TU, IN, MO, PO]
Hemos recorrido todo el dominio de vA1 así que avanzamos a la siguiente variable horizontal: vA2.

- Quinta iteracción:

Variable seleccionada: vA2
Asignamos PAN a la variable seleccionada Recorremos restricciones de vA2:

- Restriccion4: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB1 porque su segunda letra no es P
  - Restriccion5: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB2 porque su segunda letra no es A
  - Restriccion6: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB3 porque su segunda letra no es N
  Algún dominio se ha quedado vacío, así que:
  - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  - Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA1: [TIM]
  vA2: [LAL, TIM, UNO]
  vB1: [TE, LE, TU, IN, MO, PO]
  vB2: [TE, LE, TU, IN, MO, PO]
  vB3: [TE, LE, TU, IN, MO, PO]
  Como el dominio de la variable actual no está vacío, seguimos con esa variable para probar el siguiente valor de su dominio.
  Sexta iteración:
  Variable seleccionada: vA2
  Asignamos LAL a la variable seleccionada
  Recorremos todas las variables sin asignar y eliminamos LAL para no permitir palabras repetidas: Ningún dominio queda vacío.
  Recorremos restricciones de vA2:
  - Restriccion4: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB1 porque su segunda letra no es T
  - Restriccion5: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB2 porque su segunda letra no es I
  - Restriccion6: eliminamos [TE, LE, TU, IN, MO, PO] del dominio de vB3 porque su segunda letra no es M
  Algún dominio se ha quedado vacío, así que:
  - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  - Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: VA1: [TIM]
  vA2: [TIM, UNO]
  vB1: [TE, LE, TU, IN, MO, PO]
  vB2: [TE, LE, TU, IN, MO, PO]
  vB3: [TE, LE, TU, IN, MO, PO]
  Séptima iteración:
  Variable seleccionada: vA2
  Asignamos TIM a la variable seleccionada
  Recorremos todas las variables sin asignar y eliminamos TIM para no permitir palabras repetidas: El dominio de vA1 queda vacío.
  Algún dominio se ha quedado vacío, así que:
  - Deshacemos la asignación de la variable y eliminamos de su dominio de esa palabra
  - Restauramos los dominios afectados.
  Dominios de las variables no asignadas después de restaurar: vA1: [TIM]
  vA2: [UNO]
  vB1: [TE, LE, TU, IN, MO, PO]
  vB2: [TE, LE, TU, IN, MO, PO]
  vB3: [TE, LE, TU, IN, MO, PO]
  Octava itreración:
  Variable seleccionada vA2.
  Asignamos UNO a la variable seleccionada.
  Recorremos todas las variables sin asignar y eliminamos UNO para no permitir palabras repetidas: Ningún dominio queda vacío.
  Recorremos restricciones de vA2:
  - Restriccion4: eliminamos [TE, LE, IN, MO, PO] del dominio de vB1 porque su segunda letra no es U

- Restriccion5: eliminamos [TE, LE, TU, MO, PO] del dominio de vB2 porque su segunda letra no es N
- Restriccion6: eliminamos [TE, LE, TU, IN] del dominio de vB3 porque su segunda letra no es O

Ningún dominio se ha quedado vacío, así que:

- Deshacemos la asignación de la variable.

- Restauramos los dominios afectados.

Dominios de las variables no asignadas después de restaurar: VA1: [TIM]
vA2: [UNO]
vB1: [TE, LE, TU, IN, MO, PO]
vB2: [TE, LE, TU, IN, MO, PO]
vB3: [TE, LE, TU, IN, MO, PO]

Como ya hemos recorrido todo el dominio de vA2 y no hay más variables horizontales hemos acabado la traza de AC3.

Sección de experimentación

1. Tamaño de crucigrama: el objetivo es evaluar la eficacia de los algoritmos con crucigramas de cierto tamaño.

2. Pruebas realizadas: en tableros de 2x3. Comprobamos el comportamiento de los algoritmos en diversas situaciones. Para ello, probamos un crucigrama sin solucion y ya determinado por AC3, donde deja valores en el dominio pero no tiene solución y, crucigramas con solución. Creamos el tablero 2x3 sin solución y aplicamos el AC3 y verificamos que no se asignan valores.

3. Nos aseguramos que registramos el tiempo de ejecución y cualquier cambio en los dominios de las variables durante las pruebas. Además, podemos imprimir información detallada sobre el estado de las variables antes y después de aplicar los algoritmos para facilitar el análisis.

Estudio de tiempos del algoritmo

Analizando los tiempos generados a partir del ejemplo proporcionado en el enunciado de la práctica:

FC	0,002056360244750970
AC3	0,002255678176879880
FC CON AC3	0,000441551208496093
ACE+FC	0,00269722938537597

Podemos observar que el tiempo invertido en AC3 es mayor que el invertido en FC, pero el tiempo de FC con o

sin AC3 cambia bastante, llegando a obtener en este ejemplo una ganancia de 4,7, es decir, aplicando AC3 conseguimos que FC sea 4,7 veces más rápido.