Alumno: Tartaglia, Juan Ignacio
Legajo: 12566
UNCUYO - Facultad de Ingeniería.
Licenciatura en Ciencias de la Computación.
Inteligencia Artificial 1:
Ejercitacion: CSP

---

1. Describir en detalle una formulación CSP para el Sudoku.

2. Utilizar el algoritmo AC-3 para demostrar que la arco consistencia puede detectar la inconsistencia de la asignación parcial *{WA=red, V=blue}* para el problema del colorar el mapa de Australia (Figura 5.1 AIMA 2da edición).

3. Cual es la complejidad en el peor caso cuando se ejecuta AC-3 en un árbol estructurado CSP. (i.e. Cuando el grafo de restricciones forma un árbol: cualquiera dos variables están relacionadas por a lo sumo un camino).

4. (opcional) AC-3 coloca de nuevo en la cola todo arco **(X_i, X_k)** cuando cualquier valor es removido del dominio de X_i incluso si cada valor de X_k es consistente con los valores restantes de X_i. Si por cada arco **(X_i,X_k)** se <u>lleva cuenta del número de valores que quedan de X_j que sean consistentes con X</u>. Explicar como actualizar ese número de manera eficiente y demostrar que la arco consistencia puede lograrse en un tiempo total $O(n^2d^2)$ (AC-4)
5. Demostrar la correctitud del algoritmo CSP para árboles estructurados (sección 5.4, p. 172 AIMA 2da edicion). Para ello, demostrar:
    a. Que para un CSP cuyo grafo de restricciones es un árbol, 2-consistencia (consistencia de arco) implica n-consistencia (siendo n número total de variables)
    b. Argumentar por qué lo demostrado en a es suficiente.

Alumno: Tartaglia, Juan Ignacio **Legajo:** 12566

UNCUYO - Facultad de Ingeniería.
Licenciatura en Ciencias de la Computación.

Inteligencia Artificial 1:
Ejercitacion: CSP

1- Descripcion de una formulacion CSP para el Sudoku:

- Creo una matriz de 9x9, la cual va a contener en cada una de sus posiciones valores numericos enteros que van desde el 1 hasta el 9 (1,2,3,4,5,6,7,8,9).
- Tomo en cuenta las restricciones del juego que son:
  - No pueden haber numeros repetidos en una misma columna o fila
  - No puede haber numeros repetidos en cada sub-matriz de 3x3 contenida en mi matriz de 9x9.

2-

- Vamos a comenzar en un estado inicial sin asignaciones, es decir:
  - WA → R / V / A
  - NT→ R / V / A
  - Q→ R / V / A
  - NSW→ R / V / A
  - V→ R / V / A
  - SA→ R / V / A
  - T→ R / V / A
- Luego, elegimos el color rojo para WA, es decir:
  - WA → R
  - NT→ R / V / A
  - Q→ R / V / A
  - NSW→ R / V / A
  - V→ R / V / A
  - SA→ R / V / A
  - T→ R / V / A
- Verificamos la consistencia de arco y llegamos al estado:
  - WA → R
  - NT→ V / A
  - Q→ R / V / A
  - NSW→ R / V / A
  - V→ R /V / A
  - SA→ V / A
  - T→ R / V / A

- Luego, elegimos el color azul para V, es decir:
  - WA → R
  - NT→ V / A
  - Q→ R / V / A
  - NSW→ R / V / A
  - V→ A
  - SA→ V / A
  - T→ R / V / A
- Se verifica la consistencia de arco y llegamos al estado:
  - WA → R
  - NT → A
  - Q →
  - NSW→ R
  - V → A
  - SA→ V
  - T→ R / V A
- Finalmente el conjunto Q, quedo vacio por lo que concluimos que es incosistente.

3- La complejidad en el peor caso cuando se ejecuta AC-3 en un árbol estructurado CSP es $O(nd^2)$.

5- Asignamos a cualquier variable como la raíz del árbol, y ordenamos las variables desde la raíz hasta las las hojas de tal modo que el padre de cada nodo en el árbol lo precede en el ordenamiento y luego etiquetamos las variables X1…, Xn.

Aplicamos la comprobacion de consistencia de arco para j=n hasta j=2 al arco (Xi, Xj), donde Xi es el padre de Xj, quitando los valores del DOMINIO[Xi] que sea necesario.

Para j=1 hasta j=n, asignamos cualquier valor para Xj consistente con el valor asignado para Xi, donde Xi es el padre de Xj.

Luego de este paso, el CSP sera arco consistente, por ello al asignarle mas valores a el mismo no deberemos realizar ninguna vuelta hacia atras, lo cual nos asegura que el arbol sera n-consistente.