Algoritmia - Diseño recursivo.

Octubre 2015

$Q \equiv \{ n \geq 1 \}$

Funcion Evaluacion (A [1..n][1..n] : matriz de enteros ; v : entero) retorna (b : booleano)

$R \equiv \{ b = (\exists i)\ (A[i-1][i-1] \times A[i][i] = v\ ;\ 2 \leq i \leq n \}$

Inducción Noetheriana: A cada matriz le asociamos un natural n que corresponde al orden de la matriz. Seguidamente aplicaremos inducción simple sobre dicho natural n.

Sustituimos n por j en R:

$Q' \equiv \{ 1 \leq j \leq n \}$

Funcion iEvaluacion (A [1..n][1..n] : matriz de enteros ; v : entero ; j = entero) retorna (b : booleano)

$R' \equiv (b = (\exists i)(A[i-1][i-1] \times A[i][i] = v\ :\ 2 \leq i \leq j \}$

Dado el problema iEvaluacion (A, v, j), para pasar al siguiente problema eliminamos una fila y una columna. Así, el problema sucesor será iEvaluacion (A, v, j-1).

$B_T \equiv j = 1$, la sección de la matriz tiene un único elemento, esto es, una única fila y columna.

$B_{nT} \equiv j > 1$, la sección de la matriz tiene varias filas y columnas, el mismo nº de ambas.

En caso de que la matriz tratada tuviera un único elemento, j = 1, solución del problema sería falso ya que no existen dos elementos de la matriz principal cuyo producto sea v.

La solucion del problema iEvaluacion (A, v, j) consiste en comprobar si hay dos elementos de la diagonal principal consecutivos, A[K][K] con K = 1, 2, ... j cuyo producto sea v, esto es: $(\exists i)(A[i-1][i-1] \times A[i][i] = v\ :\ 2 \leq i \leq j)$, llamamos b al resultado

Hipótesis de recurrencia: suponemos conocida la solución del subproblema iEvaluación (A, v, j-1) y llamamos b' a tal valor. Esto quiere decir que suponemos conocida la solución de si existen dos elementos de la diagonal principal, A[K][K] con K = 1, 2.. j-1, sea v:

$$(\exists i)(A[i-1][i-1] * A[i][i] = v\ :\ 2 \leq i \leq j-1)$$

Para obtener la solución al problema iEvaluacion (A, v, j) a partir de la solución del subproblema iEvaluacion (A, v, j-1) bastaría con comprobar si $A[j][j] \times A[j-1][j-1] = v$:

$$(\exists i)(A[i-1][i-1] * A[i][i] = v\ :\ 2 \leq i \leq j-1)$$

o lo que nos devuelve:

$$b = b' \text{ OR } (A[i][i] * A[i-1][i-1] = v)$$

```
Funcion iEvaluacion (A[1..n][1..n] : matriz de enteros ; v, j : entero) retorna
(b : booleano)
caso
    j = 1 → FALSO
    j > 1 → iEvaluacion (A, v, j-1) OR (A[j-1][j-1] * A[j][j] = v)
fcaso
ffuncion
```

donde $A[j-1][j-1] * A[j][j] = v$, proporciona el valor verdadero si $A[j-1][j-1] * A[j][$
es igual a v, y falso en caso contrario. Llamada inicial con: iEvaluacion(A, v, n)

$Q(\bar{x}) \wedge B_T(\bar{x}) \Rightarrow R(\bar{x}, triv(\bar{x}))$

$1 \leq j \leq n \wedge j = 1 \Rightarrow b = (\exists i)(A[j-1][i-1] * A[i][i] = v : 2 \leq i \leq j)$

¿b cuando $j = 1$?

Cuando $j = 1$, el predicado: $(\exists i)(A[j-1][i-1] * A[i][i] = v : 2 \leq i \leq 1)$

Donde $\exists$ actuá sobre un rango vacío, por definición FALSO, que es el caso base luego se cumple h. recurrencia.

$Q(\bar{x}) \wedge B_{NT}(\bar{x}) \wedge R(s(\bar{x}), \bar{y}') \Rightarrow R(\bar{x}, c(\bar{y}', \bar{x}))$

$1 \leq j \leq n \wedge j = 1 \wedge b' = (\exists i)(A[i-1][i-1] * A[i][i] = v : 2 \leq i \leq j-1)$

entonces $b = b'$ OR $(A[j-1][j-1] * A[j][j] = v)$, con lo que sustituimos b' por su valor (hip. de inducción)

$b = b'$ OR $(A[j-1][j-1] * A[j][j] = v) =$

$= (\exists i)(A[i-1][i-1] * A[i][i] = v : 2 \leq i \leq j-1)$ OR $(A[j-1][j-1] * A[j][j] = v) =$

$= (\exists i)(A[i-1][i-1] * A[i][i] = v : 2 \leq i \leq j)$, que corresponde con la postcondición del problema, luego se cumple.

Definimos la funcion t del siguiente modo: $t(A, v, j) \overset{def}{=} j$ por tanto $1 \leq j \leq n$; $j \geq 0$