CC41A Lenguajes de Programación

Tarea 2 - Otoño 2006

Profesor: Luis Mateu

En esta tarea se pide programar en Scheme *estrictamente funcional* la función resolver-sistema. Esta función resuelve un sistema de ecuaciones lineales por medio de la diagonalización de un matriz. Recibe como argumento la matriz asociada al sistema de ecuaciones (representada como una lista de listas) y entrega como resultado los valores de la incógnitas del sistema (representados como una lista).

El cuadro de abajo a la izquierda muestra un ejemplo de un sistema de ecuaciones lineales y su solución. El cuadro de abajo a la derecha muestra la invocación de matriz-inversa para resolver el sistema y el resultado de su evaluación:

```
  x + 5y + 7z = 25
-3x + 5y + 4z = 15
11x - 3y + 3z = 17
Sol.: x= 4  y=5  z=-2
```

```
(resolver-sistema '(
  ( 1  5  7  25)
  (-3  5  4  15)
  (11 -3  3  17)))  => (4 7 -2)
```

Metodología

En un sistema de ecuaciones el resultado no cambia al reemplazar una ecuación por otra que resulta ser la suma de esa ecuación con otra ponderada por algún valor. Por ejemplo si F_i y F_k son filas de la matriz que representan las i-ésima y k-ésima ecuaciones del sistema, entonces se puede hacer el siguiente reemplazo:

$$F_k \quad \text{--->} \quad F_i + F_k*a$$

El sistema se resuelve en dos etapas: diagonalización y resolución. Para diagonalizar, se construye una sucesión de matrices M_i en donde M_i es de la forma como se indica en el cuadro de más abajo. M_{i+1} se construye a partir de M_i reemplazando las filas F_k, con k>i, como se indica más arriba con:

$$a = - 1 / m_{k\,i}$$

Luego se divide la fila i+1 por $m_{i+1\ i+1}$ para que quede un 1 en la diagonal. Al final, la matriz M_n está diagonalizada porque contiene solo unos en su diagonal.

$$\begin{array}{llllllll}
1 & m_{12} & \ldots & m_{1i-1} & m_{1i} & m_{1\ i+1} & \ldots & m_{1\ n+1} \\
0 & 1 & \ldots & m_{2i-1} & m_{2i} & m_{2\ i+1} & \ldots & m_{2\ n+1} \\
0 & 0 & \ldots & & & & \ldots & \\
0 & 0 & \ldots & 0 & 1 & m_{i\ i+1} & \ldots & m_{i\ n+1} \\
0 & 0 & \ldots & 0 & m_{i+1\ i} & m_{i+1\ i+1} & \ldots & m_{i+1\ n+1} \\
 & & & & & \ldots & & \\
0 & 0 & \ldots & 0 & m_{n\ i} & m_{n\ i+1} & \ldots & m_{n\ n+1}
\end{array}$$

Para la segunda etapa, la resolución, se procede de manera similar, pero en orden inverso: desde la n-ésima columna hacia la primera columna, para llevar la matriz a la forma indicada en el cuadro derecho. De ahí se desprenden trivialmente los valores para las incógnitas.

$$\begin{array}{lllll}
1 & 0 & \ldots & 0 & m_{1\ n+1} \\
0 & 1 & \ldots & 0 & m_{2\ n+1} \\
 & & \ldots & & \\
0 & 0 & \ldots & 1 & m_{n\ n+1}
\end{array}$$

Indicaciones

Para resolver esta tarea:

- Construya funciones pequeñas y *pruébelas en forma independiente*. Si una función *f* invoca otras funciones *g* y *h*, no intente probar *f* si no está seguro que *g* y *h* funcionan correctamente.
- Los ciclos se logran con funciones recursivas.
- Utilice `(length L)` para calcular el largo de la lista L y (list-ref L k) para obtener el (k+1)-ésimo elemento de la lista L.
- Le serán de utilidad una función que calcule la suma de 2 fila (listas) y otra que obtenga la ponderación de una fila por un valor escalar. No olvide que siempre debe construir nuevas filas para estos efectos.
- Están prohibidos los procedimientos que en su nombre incluyen el caracter '!', pues se salen del paradigma funcional.
- No necesita ser eficiente.

Entrega

La tarea se entrega mediante U-cursos. El plazo vence el Martes 30 de Mayo. Se descontará medio punto por día de atraso. Haga su tarea antes del control 2. *No* se aceptarán tareas que funcionen a medias.