Estructuras de Datos y Algoritmos

Año 2022

Introducción a las Estructuras de Datos

¿Qué son las Estructuras de Datos?

Debemos discutir por separado qué son las estructuras y qué son los datos.

El término estructura aparece en muchas disciplinas. Existen estructuras económicas, estructuras sociales, estructuras matemáticas, etc. ¿Qué hay de común en cada una de estas denominaciones?

En primer lugar, se habla de una *multiplicidad de elementos*. No se habla de la estructura de un elemento, salvo que se quisiera referir a que éste tiene muchas partes y que fuera motivo de estudio la estructura de las partes.

En segundo lugar, para que haya estructura los múltiples elementos deben *estar relacionados*. Un conjunto de elementos carentes de relaciones entre ellos no ofrece una estructura, o si admitimos esto como caso extremo, tiene por estructura la inconexión total. Esta parece ser la menos interesante de las posibles estructuras. Una estructura la vemos más rica cuanto mayor sea el número de relaciones que vincula a sus elementos.

Entonces, en general, *una estructura es un conjunto provisto de relaciones*. Puede corroborarse este análisis repasando diversas estructuras. Una estructura social es un conjunto de personas y las relaciones entre ellas. Una estructura económica es un conjunto de elementos económicos y las relaciones entre ellos y así sucesivamente. Por lo tanto una estructura de datos es un conjunto de datos y las relaciones entre ellos.

Falta precisar *qué es un dato*. Esta palabra tiene dos acepciones, en la primera de ellas hay que considerarlo como opuesto a resultado. Así, decimos que un proceso recibe datos y entrega resultados. No es ésta la acepción que nos interesa.

Para abordar la segunda acepción, y luego poder discernir entre dos tipos de estructuras de datos, debemos precisar algunas nociones básicas de la informática.

Todo conocimiento es expresado extramentalmente en forma de enunciados. La informática no es excepción respecto de esto. Los enunciados forzosamente debieron ser codificados, o sea, representados por hileras de algún alfabeto para el cual se dispone de una tecnología de representación.

La construcción de la representación se encara de dos modos.

En la primera, los enunciados son representados en una lengua natural cuyo alfabeto es usualmente recodificado con otro más primitivo. Así, hablar del alfabeto ASCII es mencionar una de las más habituales recodificaciones de nuestros símbolos ordinarios de escritura a hileras binarias.

Esta representación es la habitual para tratamiento de textos. Las ayudas computacionales facilitan acciones sobre el texto en respuesta a las indicaciones de un operador de la máquina. Estos procesos no requieren ninguna interpretación de los símbolos.

El sistema es más amigable si le fueron incorporadas las reglas sintácticas elementales, como ser el reconocer fronteras sintácticas usuales en las lenguas naturales: el espacio como frontera de palabras, los puntos como frontera de enunciados, etc. De este modo el operador puede referirse en sus comandos a estas componentes lingüísticas.

Cuando se opta por esta representación, sólo se espera que la máquina transforme el texto de acuerdo a las órdenes del operador y apoye a éste en organizar la distribución sobre el papel (formateo) y en

las tareas subalternas de memorización (almacenamiento) y comunicación, desligándolo de muchos detalles.

Sin embargo existen aplicaciones en las cuales se desea delegar en la máquina, además de la memorización y comunicación, tareas de lógica deductiva.

La programación de la máquina es el medio de incorporar a ésta los esquemas deductivos que la máquina deberá aplicar.

Si los enunciados estuvieran todos conformados con dos términos y el verbo ser como cópula, tal como los supone la lógica tradicional, y fueran todos universales afirmativos, bastaría con rastrear posibles términos medios para aplicar con los enunciados así apareados silogismos del tipo "Bárbara":

Sí	Los animales son mortales.	[premisa mayor]
y	Los hombres son animales.	[premisa menor]
Entonces	Los hombres son mortales.	[conclusión]

Para que esto sea posible, cada término debiera estar codificado por la misma hilera en todas sus apariciones. No nos interesa ahora recorrer las letras de los términos. Es más, nos conviene cambiar la codificación para no depender de inflexiones gramaticales, artículos, etc., y lo que puede ser aún más importante lograr una codificación de largo constante.

Bajo condiciones que aseguren la posterior descomposición unívoca de la hilera resultante en sus partes, se puede utilizar como codificación de un enunciado la concatenación, en un orden convenido, de las codificaciones de sus partes. Justamente, una codificación de largo constante es uno de los modos más sencillos de asegurar la descomposición unívoca.

Por supuesto que la codificación de los términos debiera incluir la suposición, para evitar los silogismos defectuosos que surgirían de aparear términos idénticos de distinta suposición.

Y aquí tenemos la segunda acepción de dato: *cada instancia del código de un término*.

Sin embargo, este esquema tan sencillo tiene diversos inconvenientes. En primer lugar, habría una explosión combinatoria de conclusiones. La utilidad de cada una de éstas no puede ser juzgada por la máquina. En segundo lugar, las premisas de las deducciones computacionales no suelen tener la estructura presupuesta en la lógica tradicional preocupada por la actividad científica: descubrir esencias y propiedades. En ella, los enunciados relacionan términos usando el verbo ser como cópula. El trabajo informático, por lo contrario, está dirigido sobre todo a lo contingente. La descripción de lo accidental, sobre todo del obrar es mucho más cómodo con el uso de otros verbos. Para enunciados de este tenor no hay una regla general de deducción. Los razonamientos se plantean con modalidades como la que sigue:

Premisas:

mayor: *Si*

el sujeto S realiza la acción A_1 sobre el objeto O

entonces

debe realizar la acción A_2 sobre el objeto O.

menor: NN *realizó la acción* A_1 *sobre el objeto* O.

Conclusión:

NN *debió realizar la acción* A_2 *sobre el objeto* O.

o bien

NN *deberá realizar la acción* A_2 *sobre el objeto* O.

La premisa mayor surge generalmente de una actividad ordenadora del hombre; es una norma. El

análisis de la situación a informatizar lleva a descubrir la vigencia de tales normas. La conclusión en su primera forma, corresponde a alguien que quiere controlar al individuo NN y, en la segunda, a alguien que quiere ordenar la actividad de NN. La conveniencia de una u otra forma dependerá de matices temporales.

Mirando el ejemplo, se ve que la norma reflejada en la premisa mayor puede ser tomada como un encadenamiento de acciones. Completar la segunda tras la otra insume un tiempo. Falta una explicitación del plazo que se otorga para realizar la segunda tras la primera. Una redacción más precisa lo hubiera incluido.

En la primera redacción, de la conclusión suponemos que el plazo ya expiró y en la segunda aún no. Esta subordinación de la validez de las afirmaciones al tiempo, es una consecuencia de la contingencia de las mismas.

En lenguajes de programación de tipo lógico, se hablará de reglas para representar conocimientos normativos del tenor de lo ilustrado con la premisa mayor y hechos para hablar de afirmaciones singulares como la ilustrada en la premisa menor. El lenguaje permitirá describir unas y otras entremezcladamente.

En una programación más antigua, las premisas individuales estarán en archivos y la premisa normativa estará presente, aunque veladamente, en el programa.

Estos archivos contendrán hileras obtenidas siguiendo la segunda técnica de codificación descripta. Así para referirse a:

"NN *realizó la acción* A_1 *sobre el objeto* O."

se usará la hilera que surge concatenando los códigos de los términos NN, A_1 y O. Como sabemos, de la lógica, que con suposición formal no pueden estar NN, A_1 y O sino sólo sus códigos. Omitiremos, en bien de la agilidad de la lectura, indicar explícitamente que se trata de códigos. En lugar de decir código(NN) diremos NN, y así sucesivamente. Entonces, representaremos a la hilera resultante de codificar la premisa como (NN; A_1; O) en lugar de código(NN) - código(A_1) - código(O).

La presencia de la premisa mayor se pone de manifiesto si, por ejemplo, dicho programa busca hileras cuya parte central sea A_1 y lista las partes correspondientes a NN y a O en una hoja de título: "Notificar que deben hacer A_2".

Las yuxtaposiciones que se realizan para representar enunciados constituyen un tipo de relaciones entre ellas. Estas relaciones están impuestas por los enunciados que se quieren representar y sería un engaño colocar NN al lado de A_1 si efectivamente NN no realizó la acción A_1. Estos juicios que versan sobre materia contingente constituyen información. Estudiar las propiedades de las relaciones entre datos que surgen de representar la información es un tema de estudio que podría llamarse **estructura de la información**.

Además, hay otras relaciones que afectan a los datos. Cuando éstos se almacenan en una memoria se establece una relación entre celdas y sus contenidos que son datos. Estas relaciones están influenciadas por las propiedades tecnológicas de la memoria en uso y no afectan a la información que los datos representan. Del modo en que se dispongan las hileras que representan información en la memoria surgirá una mayor o menor eficacia en la evocación. Esto constituye entonces un arte que construye **estructuras de almacenamiento**. El título hace hincapié en el hecho de aparecer estas relaciones sólo al ser almacenada la información en una memoria.

Esta materia se ocupará primordialmente de esta segunda visión. La primera, o sea las estructuras de la información, le interesará sólo en la medida que su conocimiento influya en un mejor diseño de las estructuras de almacenamiento. Pensamos que el estudio de la estructura de la información es más propio del Análisis de Sistemas, ya que ella estudia las realidades dadas, extra-computacionales y por lo tanto, está en una actitud más científica. Las estructuras de almacenamiento son realizaciones humanas

y como tales, corresponden a una actitud técnica.

La función de las Estructuras de Almacenamiento

Tener siempre presente la completa incomprensión que tiene la máquina de lo que procesa, ayuda a ver algunas técnicas con un campo de aplicación más amplio del que se consigue de otro modo.

La máquina es una procesadora de símbolos. El usuario le coloca hileras y le instala las reglas de transformación con lo cual consigue disponer en lugar y tiempo oportunos la misma u otras hileras derivadas que para él significan algo.

Poco importa para lo que sigue, si las hileras representan información (planteo inicial y origen de la informática), enunciados esenciales, imágenes, sonidos, etc.

Lo que interesa es que tales hileras a veces deben ser conservadas fuera del foco de la atención para un posterior retorno. A esta actividad la denominamos *memorización* o refiriéndonos a sus dos extremos temporales *memorización* (en sentido más estricto) y *evocación*.

Otra actividad neutra respecto de los significados es la *comunicación*, pero ella es estudiada por otra disciplina.

Para realizar la memorización se construyen dispositivos que pueden retener materializaciones de hileras para devolverlas posteriormente. Los dispositivos que se construyen no suelen satisfacer los requerimientos conceptuales de las evocaciones lo que lleva a desarrollar estrategias de almacenamiento que ayuden luego a localizar las hileras que responden a un pedido de evocación. Por lo tanto, se deberán estudiar formas de organización de los dispositivos que memorizan y sus limitaciones, así como analizar el problema conceptual de la evocación.

En general, una computadora cuenta con un dispositivo de almacenamiento que podría satisfacer los requerimientos conceptuales de las evocaciones: las *memorias asociativas*. Las memorias asociativas se diseñan para propósitos específicos, son costosas y por lo tanto de tamaño acotado, y entonces no son adaptables para propósitos generales. En nuestro caso que deseamos mantener información, sin conocer previamente la clase de información que se deseará almacenar y además que para distintas aplicaciones se almacenarán posiblemente distintos tipos de información, las memorias asociativas no resultan adecuadas. Por otra parte están las *memorias direccionables*. Ellas son menos costosas y por consiguiente una computadora en general tiene mayor capacidad de almacenamiento de este tipo, están compuestas por celdas de tamaño fijo, que en la actualidad suelen ser de 32 bytes o 64 bytes y cada una de ellas tiene asociada una *dirección*. Además son capaces de almacenar cualquier tipo de información, sólo por yuxtaposición de celdas (la cantidad de celdas necesarias para almacenar los datos requeridos). Por lo tanto son más versátiles en cuanto a lo que pueden almacenar, tienen mayor capacidad que las memorias asociativas, aunque no se encuentran intrínsecamente preparadas para responder a evocaciones. Por lo tanto, nos dedicaremos a analizar como estructurar los datos en una memoria direccionable, de manera tal de poder responder a evocaciones, tratando además de resolverlas eficientemente.

Otro dispositivo de almacenamiento más masivo y más barati, son las *memorias secundarias* o *disco*. En general ellos tienen también tamaño de celda fijo, pero poseen la desventaja de que su acceso es mecánico “se debe mover la cabeza lectora hasta el sector donde se encuentra la información a recuperar”, lo que implica un tiempo adicional en la respuesta, cosa que no ocurre ni en la memoria asociativa ni en la direccionable (se accede a una celda de la memoria asociativa por circuito y a una celda de la memoria direccionable a través de su dirección en un ciclo de memoria).