Recursión

La recursión consiste en la definición de una propiedad, una operación, un procedimiento o una función en términos de sí misma. Un algoritmo es recursivo cuando tiene la capacidad de invocarse a sí mismo (de forma directa o indirecta). En cualquier caso, se garantiza que el número de llamadas recursivas garantice la finalización de un bloque de instrucciones.

Cada invocación recursiva resuelve un problema de menor tamaño, hasta llegar a un problema cuya resolución sea directa. Esta técnica se conoce como divide y vencerás, donde se trata de resolver un problema mediante su descomposición en varios subproblemas similares al original (o del mismo tipo) pero con datos más pequeños. Si un problema puede subdividirse en subproblemas, entonces este proceso puede ser aplicado a cada uno de los subproblemas hasta que se pueda encontrar una solución directa. Finalmente, se combinan las soluciones de los subproblemas para producir la solución final del problema original.

Un algoritmo recursivo está formado por:

Caso Base : Consiste en los casos del problema que se resuelven con un segmento de código sin recursividad. Generalmente corresponden a instancias del problema simples y fáciles de implementar

Caso Recursivo : Consiste en los casos que se resuelven mediante invocaciones a sí mismo, tratan de reducir el problema de dimensión para llegar al Caso Base

Parámetros : Una función recursiva tiene al menos un parámetro (asumiendo que no existen parámetros globales) y generalmente es pasado por valor. Así, el algoritmo debe proveer un mecanismo que modifique el valor de dicho parámetro en las invocaciones recursivas hasta llegar al caso base. El resultado puede ser retornado, o dejado en un parámetro por referencia

```
function Factorial (Integer iN) : Integer
  if iN <= 0 then
    return 1;
  else
    return iN * Factorial (iN - 1)
  end
end
```

Como ya es sabido, al invocar una función se crea un ambiente de ejecución. En una función recursiva sucede de la misma forma, donde cada ambiente se puede definir como una pila para almacenar cada ambientes local de ejecución (i.e. parámetros y demás variables locales). Se debe tomar en cuenta las reglas de alcance en cada ambiente de ejecución.

Las funciones recursivas se pueden clasificar de varias maneras:

De acuerdo al lugar de invocación

- Directa: Una función A se invoca a sí misma
- Indirecta: Una función A invoca a una función B que invoca a la función A

De acuerdo a la cantidad de invocaciones

- Simple: Se realiza solo una invocación
- Múltiple: Se realizan dos ó una invocaciones

Considerando el punto de su invocación

- Final: La invocación recursiva es la última que se produce en el bloque de instrucciones
- No-final: Se realiza un bloque de instrucciones al regresar del ambiente de ejecución de la invocación recursiva

Observaciones

Así como existen algoritmos recursivos, también existen estructuras de datos recursivas (e.g. listas enlazadas, árboles, etc.).

Algoritmo Recursivo vs. Iterativo

Algunos algoritmos recursivos: Fibonacci, Torres de Hanoi, búsqueda binaria, búsqueda en profundidad en un grafo,función de Ackermann, etc.

esmitt ramírez/2014