Aula 03 Análise de Complexidade Baseada na Estrutura

Estudo que possibilita a obtenção da complexidade de um algoritmo a partir das complexidades de suas componentes.

§ Soma Pontual de Funções

Sendo f e g : D -> R+ , a soma pontual é definida por:

(f + g)(d) = f(d)+g(d), Para cada d $\in$ D

§ Máximo e Mínimo Pontual

Sendo f e g : D -> R+ , o máximo e mínimo pontuais são definidos por:

Max (f,g)(d) = Máx {f(d), g(d)}
Min (f,g)(d) = Mín {f(d), g(d)}

§ Classificação de Componentes

Uma componente de um algoritmo será dita ***conjuntiva*** quanto ela é sempre executada em qualquer execução do algoritmo. Por outro lado, chamaremos de ***disjuntivas*** as componentes que são executadas em algumas execuções do algoritmo, dependendo dos valores de entrada.

♠ Exemplos: Sejam j:=i+1 e j:=i-1 duas componentes. Na composição
j:=i+1 ; j:=i-1

elas são conjuntivas. Na composição

se i<>j então j:=i+1 senão j:=i-1

elas são disjuntivas.

§ Componentes Conjuntivas

Sendo a_0 um algoritmo com componentes a_1 e a_2 organizados de forma conjuntiva, para entradas de tamanho até *n*, a complexidade pessimista, é:

Cp_{a0} é O (Cp_{a1} + Cp_{a2})

§ Componentes Disjuntivas

Sendo a_0 um algoritmo com componentes a_1 e a_2 organizados de forma disjuntiva, para entradas de tamanho até *n*, a complexidade pessimista, é:

Cp_{a0} é O (MxAO (Cp_{a1}, Cp_{a2}))

onde, MxAO é a denotação para máximo assintótico.

§ Princípio da Absorção

Sendo f e g : D -> R+ , diz-se que f é absorvida por g se f é O(g).

Se f é absorvida por g , a soma pontual f+g é Θ (g).

Exemplos:

♠ a) Sendo a0 um algoritmo composto pelos componentes conjuntivos a1 e a2, de complexidade n.log n e n^2, respectivamente,

Cp_{a0} é O(n.log n+ n^2)

Como n.log n é O (n^2), pelo princípio da absorção temos que

n.log n + n^2 é O (n^2) , logo

Cp_{a0} é O(n^2)

♠ b) Atribuição (componentes conjuntivos):

Cp [u <- Ordene(v) é O (Cp[Ordene] + Cp[<-])

Supondo Cp[Ordene] é O (n^2) e Cp[<-] é O(n), temos que

Cp [u <- Ordene(v)] é O(n^2 + n)

Como n é O(n2), pelo princípio da absorção

Cp [u <- Ordene(v)] é O(n^2)

♠ c) Seqüência Simples (componentes conjuntivos)

Cp[S; T] é O (Cp [S] + Cp [T]),

Ordene(v); Reversa(v)

♠ d) Condicional Simples (componentes disjuntivos)

Cp[se b então S senão V] = O (Cp[b] + MxAO(Cp[S],Cp[V]))

Se Max(v)=0 então Ordene(v) senão Reversa(v)

♣ Exercício: Dado o algoritmo abaixo (Prog1) e a complexidade de seus componentes, calcule a sua complexidade através da análise da complexidade da sua estrutura.

```
Modelo de Máquina

        Comparação é O(1)
        Cp[<- vet] é O (n)
        Cp [ Max (vet) ] é O(n)
        Cp [ Reverso (vet) ] é O (15.n)
        Cp [ Ordene (vet) ] é O (n.(log n)^2)

Prog1 (vet v) {

                se Max(v) > 10 entao
                        v <- Reverso(v);
                senao
                        v <- Ordene(v);
        }
```

♠ Exemplos:

e) Seqüência - Em Geral

$c_p[S;\, T](n) = O\,(c_p[S](n) + c_p\,[\,T](s(n))\,)$

Exemplo 3.2.6, pg. 33

f) Condicional - Caso Geral

$c_p[\text{ se b então S senão T}] = O\,(c_p[b] + \text{Max}(c_p\,[\,S],\, c_p\,[\,T])\,)$

Exemplo 3.2.10, pg. 36

g) Iteração Definida - Preservação Assintótica de Tamanho

$c_p[\text{ para i de j até m faça S}]\,(n) = O\,(\,N(n).c_p[S](n)\,)\,)$

Exemplo 3.2.13, pg. 38

h) Iteração Definida - Em Geral

$$cp[\text{para i de j até m faça S}](n) = O\,(\sum_{i=j(n)}^{m(n)} cp[S]\;\;(s^{(i-j(n))}\,(n))\,)$$

Exemplo 3.2.15, pg. 40

i) Iteração Indefinida - Com testes absorvidos

$$cp[\text{enqto b faça S}](n) = O\,(\sum_{i=0}^{h(n)-1} cp[S]\;\; cp[S]\,(s^{i}(n))\,)$$

Exemplo 3.2.19, pg. 44

j) Iteração Indefinida - Em Geral

$$c_p[\text{enqto b faça S}](n) = O\,(\,c_p\,[b]\,(\,s^{\,h(n)}\,(n)\;\; + \sum_{i=0}^{h(n)-1}\;[\,c_p[b]\;\;(s^{i}(n) + c_p[S]\,(\,s^{i}(n)\,))\,]\,)$$

Exemplo 3.2.20, pg. 45