Geração de Código

Algoritmo de Programação Dinâmica para Geração Dinâmica

Geração de Código

- Um algoritmo baseado em programação dinâmica pode ser usado para estender a classe de máquinas (processadores) para as quais pode ser gerado código ótimo a partir de árvores de expressões em tempo linear
- O algoritmo de programação dinâmica particiona o problema de gerar código ótimo para uma expressão nos subproblemas de geração de código ótimo para as subexpressões

Geração de Código

- Em uma expressão E=E1+E2, o código ótimo para E é formado pela combinação de programas ótimos para E1 e E2. A geração de código ótimo para E1 e E2 é resolvida similarmente
- Um programa ótimo produzido pelo algoritmo de programação dinâmica possui uma propriedade importante: deve avaliar E=E1+E2 de maneira contígua.

Geração de Código

- Dizemos que um programa P avalia uma árvore T, com sub-árvores T1 e T2, contiguamente se o programa avalia, primeiramente, a sub-árvore T1 (T2) completamente e depois T2 (T1), para depois avaliar a raíz de T.
  - Obs: Um passo inicial geralmente é realizado visando avaliar as sub-árvores de T que necessitam ser computadas na memória

Geração de Código

- Pode-se provar que para qualquer dada linguagem de máquina e um programa P que avalia uma árvore de expressão T, pode-se determinar um programa equivalente P' tal que:
  1. P' não possui custo maior que P;
  2. P' não usa mais registradores que P;
  3. P' avalia a árvore T contiguamente
- A existência dessa prova implica que toda árvore de expressão T pode ser avaliada (gerar código) de maneira ótima por um programa que avalia T contiguamente

Algoritmo de Programação Dinâmica

- O algoritmo de Prog. Dinâmica é executado em três fases
- Considere que a máquina alvo possui r registradores
- Cada uma das fases pode ser implementada em tempo linear no tamanho da árvore de expressão

Algoritmo de Programação Dinâmica

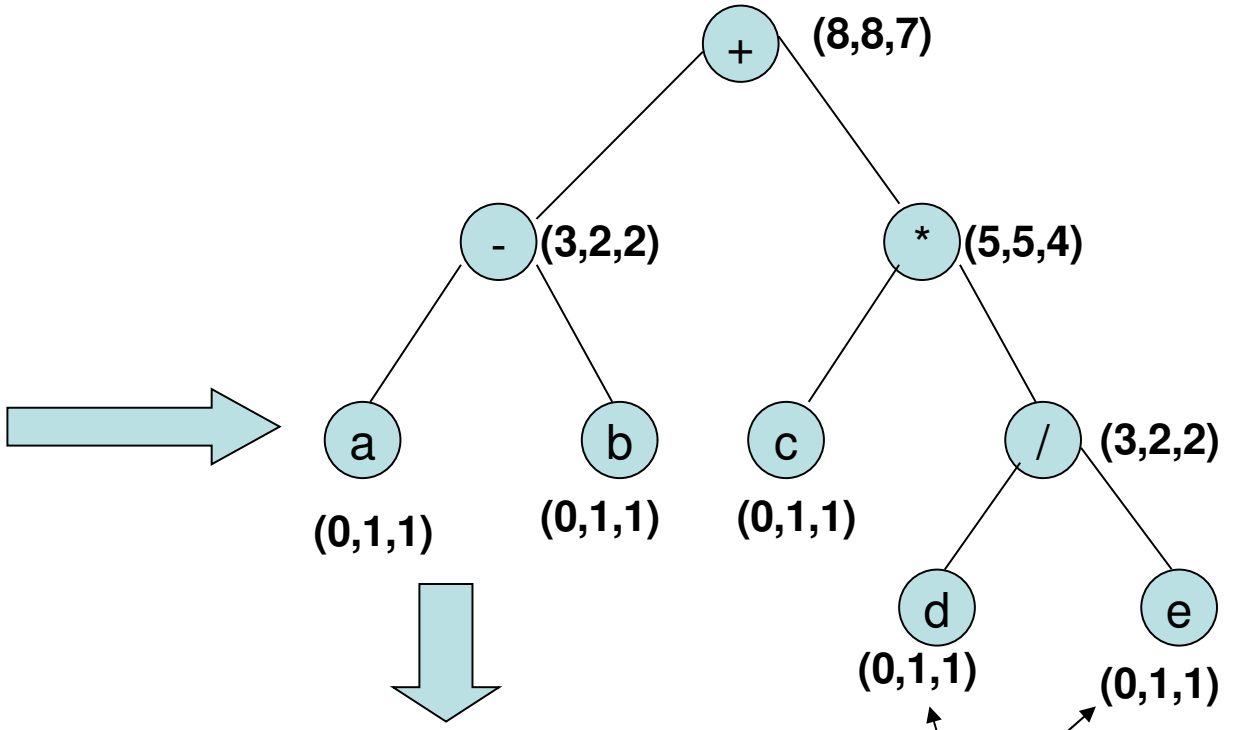
1. Computar, das folhas para a raíz, para cada vértice n da árvore de expressão T, um vetor de custo C em que o i-ésimo elemento C[i] é o custo ótimo para computar a sub-árvore S, com raíz n, em um registrador
   - Considerando que i registradores são disponíveis para a computação, para $1 \leq i \leq r$
   - O elemento na posição 0 do vetor de custos é o custo ótimo para computar o vértice n na memória
2. Atravessar T usando o vetor de custos para determinar quais sub-árvores de T devem ser computadas em memória
3. Atravessar T usando o vetor de custos e instruções associadas para gerar o código alvo final
   - O código para as sub-árvores computadas na memória é gerado primeiro

Algoritmo de Programação Dinâmica

- Exemplo:
  - Considerar uma máquina com registradores R0 e R1 e os seguintes padrões de instruções de custo unitário:
    - lw Ri, Mj
    - op Ri, Ri, Rj
    - op Ri, Ri, Mj
    - lw Ri, Rj
    - sw Rj, Mi

Algoritmo de Programação Dinâmica

- Exemplo:
  - Código:
    - t1=a-b
    - t2=d/e
    - t3=c*t2
    - t4=t1+t3



lw R0, c
lw R1, d
div R1, R1, e
mul R0, R0, R1
lw R1, a
sub R1, R1, b
add R1, R1, R0

Vetores de custo para as operações **d** e **e**. Observe que a 1a. Posição é reservada considerando que o valor da operação está na memória

Algoritmo de Programação Dinâmica

- Exercícios: executar o algoritmo de prog. dinâmica sobre as expressões a seguir:

1. a/(b+c)-d*(e+f)
2. a+b*(c*(d+e))
3. (a+p)*((b-q)/(c+r))