
Alocação de Registradores

Alocação de Registradores: Introdução

- A IR e a seleção de instruções assumiram que o número de registradores era infinito
- Objetivo da Alocação de Registradores:
 - Atribuir registradores físicos (da máquina) para os temporários usados nas instruções
 - Se possível, atribuir a fonte e o destino de MOVES para o mesmo registrador e assim eliminar MOVES inúteis

Alocação de Registradores: Introdução

O método usual para lidar com a alocação de registradores é por coloração de grafos.

- Um programa P pode ser representado com um grafo de interferência $G = (V, E)$.
 - ▶ Um vértice $v \in V$ representa um *Live Range*.
 - ▶ Uma aresta $e \in E$ conectando dois vértices $v_i, v_j \in V$ indica que v_i e v_j não podem ser atribuídos ao mesmo registrador.
- Um grafo é K -colorável se pode ser colorido com K cores, de modo que nenhum vértice adjacente tenha a mesma cor
 - ▶ K é o número de registradores disponíveis na arquitetura alvo.

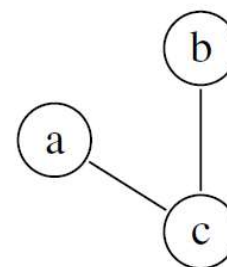
Alocação de Registradores: Grafo de Interferência

- Grafo de Interferência (IG):
 - Temos arestas entre duas variáveis $v1$ e $v2$ se elas não podem ocupar o mesmo registrador
 - Live ranges têm intersecção
 - Restrições da arquitetura
 - $a = a + b$ não pode ser atribuído ao $r12$
- O problema se transforma em um problema de coloração de grafos

Representação do Grafo de Interferência:

	a	b	c
a			x
b			x
c	x	x	

(a) Matrix



(b) Graph

Alocação de Registradores: Grafo de Interferência

Construção do Grafo de Interferência:

1. Definição de uma variável a que não seja move:

Live-out = b_1, \dots, b_j

- Adicione as arestas $(a, b_1), \dots, (a, b_j)$.

2. MOVE: $a \leftarrow c$:

Live-out = b_1, \dots, b_j

- Adicione as arestas $(a, b_1), \dots, (a, b_j)$ para os b_i 's

- Se c não interfere com a , então indica-se que entre essas variáveis existe uma instrução de cópia.

Alocação de Registradores: Coloração de Grafos

Coloração de vértices:

- Deseja-se colorir o IG com o mínimo de cores possíveis, de maneira que nenhum par de nós conectados por uma aresta tenham a mesma cor.
- As cores representam os registradores
- Se a máquina alvo possui K registradores e for possível encontrar uma K -coloração para o IG, essa coloração é uma alocação válida dos registradores.

Alocação de Registradores: Coloração de Grafos

E se não existir uma K-coloração?

- Então deve-se colocar alguns dos temporários ou variáveis na memória, operação essa chamada de *spilling*.

Coloração de vértices é um problema NP-Completo, logo a alocação de registradores por coloração também o é.

Existe uma aproximação linear que traz bons resultados...

Coloração de Grafos: Simplificação

Dividida em 4 fases:

1. Build:
2. Simplify
3. Spill
4. Select

Coloração de Grafos: Simplificação

1. Build:

- Construir o IG
- Utiliza Liveness Analysis (análise de longevidade)

Coloração de Grafos: Simplificação

2. Simplify:

Heurística:

- Suponha que o grafo G tenha um nó m com menos de K vizinhos
- K é o número de registradores
- Faça $G' = G - \{m\}$
- Se G' pode ser colorido com K cores, G também pode
- Leva a um algoritmo recursivo (pilha)
 - Repetidamente:
 - Remova nós de grau menor que K
 - Coloque na pilha
 - Cada remoção diminui o grau dos nós em G , dando oportunidades para novas remoções

Coloração de Grafos: Simplificação

3. Spill:

- Em algum momento se não houver um nó com grau $< K$, a heurística falha
- Neste caso deve-se marcar algum nó para spill
- A escolha desse nó é também uma heurística
 - Nó que reduza o grau do maior número de outros nós
 - Nó com menor custo relacionado as operações de memória

Coloração de Grafos: Simplificação

4. Select:

- Atribui as cores
- Reconstroi o grafo G adicionando os nós na ordem determinada pela pilha
- Quando adicionamos um nó, devemos ter uma cor para ele dado o critério de seleção usado para remover
- Isso não vale para os nós empilhados marcados como spill
 - Se todos os vizinhos já usarem K cores, não o adicionamos no grafo

Coloração de Grafos: Simplificação

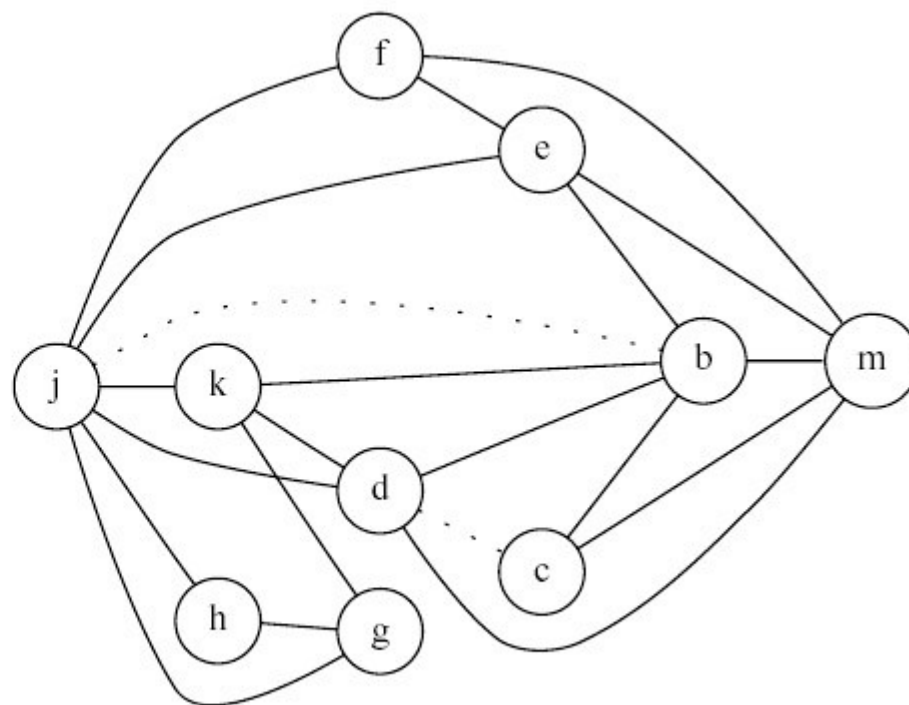
5. Start Over:

- Pode ser que o Select não consiga atribuir uma cor a algum nó
- Reescreve o programa para pegar esse valor da memória antes de cada uso e armazená-lo de volta após o uso
- Isso gera novos temporários
 - Com live ranges mais curtas
- O algoritmo é repetido desde a construção do IG
- O processo acaba quando Select tiver sucesso para todos os vértices

Coloração de Grafos: Exemplo

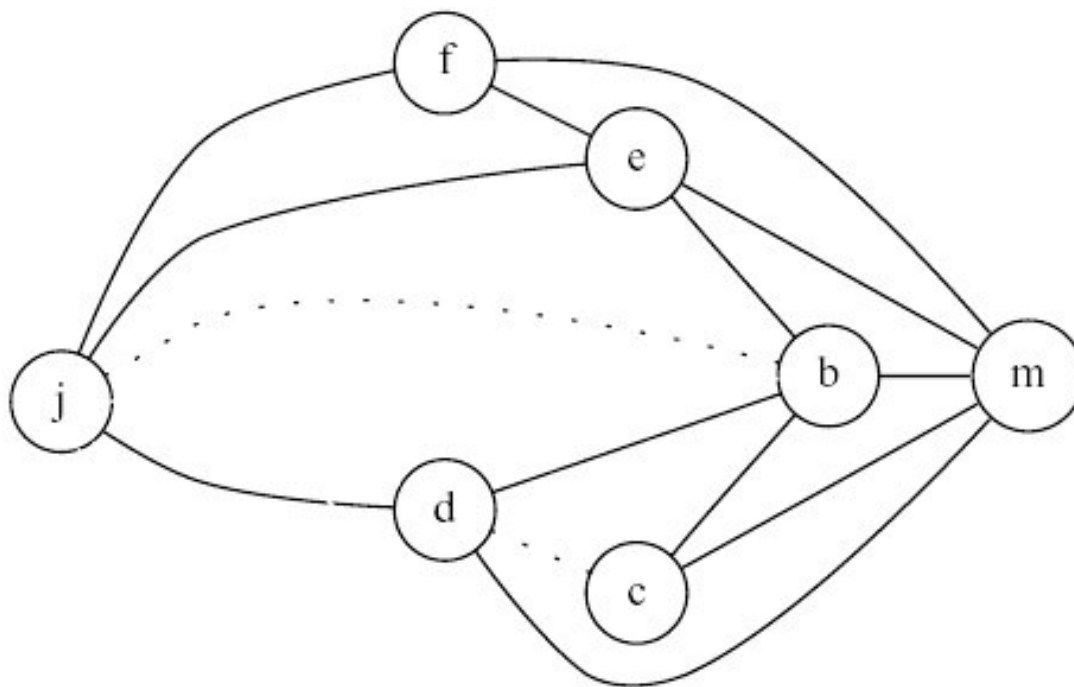
- Supor que estão disponíveis 4 registradores (4 cores)

```
live-in: k j
  g := mem[j+12]
  h := k - 1
  f := g * h
  e := mem[j+8]
  m := mem[j+16]
  b := mem[f]
  c := e + 8
  d := c
  k := m + 4
  j := b
live-out: d k j
```



Coloração de Grafos: Exemplo

- Removendo h, g , k



Coloração de Grafos: Exemplo

- Final

m	1
c	3
b	2
f	2
e	4
j	3
d	4
k	1
h	2
g	4

(a) stack (b) assignment

