

Introdução à Otimização de Código

I - Independente de Máquina

1. Peephole

Objetiva otimizar seqüências locais de código como load-stores redundantes

```
sto A, x  
lod A, x
```

2. Seqüências de desvios

```
Goto A:  
A: Goto B:
```

3. Sub-expressões comuns

Objetiva aproveitar sub-expressões já calculadas anteriormente

```
A=b+c+d; E=b+f+d  
X=b+d; A=x+c; E=x+f
```

4. Propagação de cópia

```
x=y; z=x  
z=y
```

5. Movimentação de código

For (i=0; i<N; i++) x[i]=2*PI*i;
Aux=2*PI; For (i=0; i<N; i++) x[i]=aux*i;

6. Desdobramento de loops

For (i=0; i<3; i++) x[i]=i;
x[0]=0; x[1]=1; x[2]=2;

7. Funções inline

Substituir a chamada a uma função pelo seu corpo.

II - Dependente de Máquina

1. Instruções especiais e identidades algébricas

Objetiva substituir operações mais caras por outras equivalentes mais baratas.

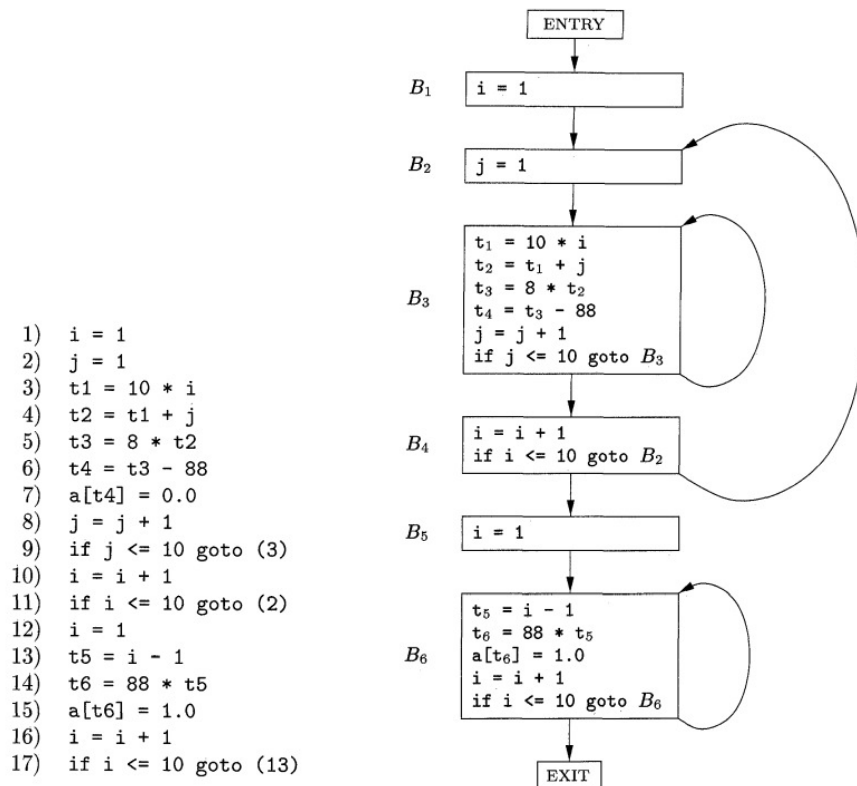
EXPENSIVE		CHEAPER
x^2	=	$x \times x$
$2 \times x$	=	$x + x$
$x/2$	=	$x \times 0.5$

$x = x + 1$ (LOD + STI + ADD + STO -> INC)

2. Alocação de registradores

Objetiva evitar acessos à memória, deixando os dados temporários em registradores

a) Construir o grafo de fluxo do código



b) Determinar o tempo de vida dos temporários

Tempo de vida é o trecho do programa entre a escrita de um temporário e a última leitura desse valor antes que ele seja reescrito. Dois temporários que estão vivos simultaneamente não podem ser alocados ao mesmo registrador

c) Construir um grafo de interferência

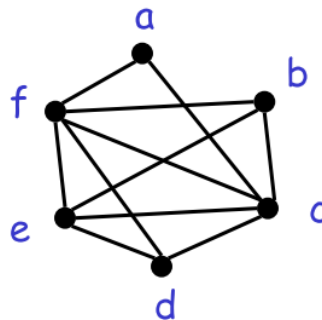
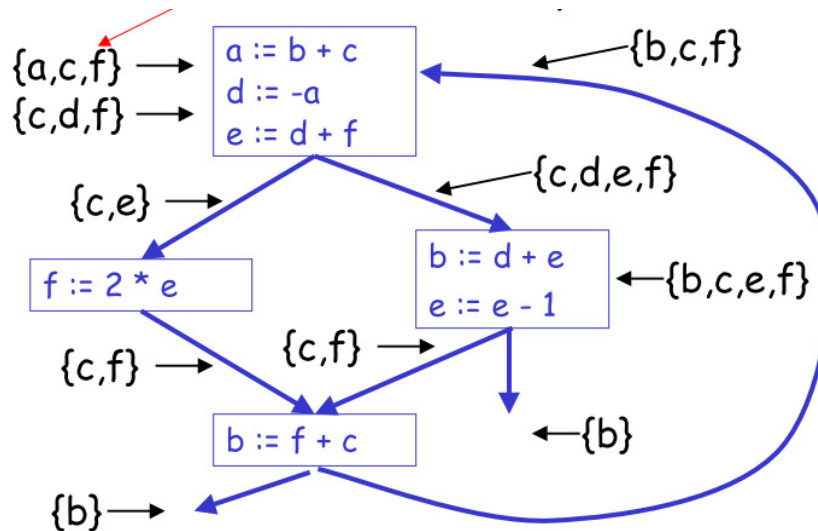
- Um nó para cada temporário

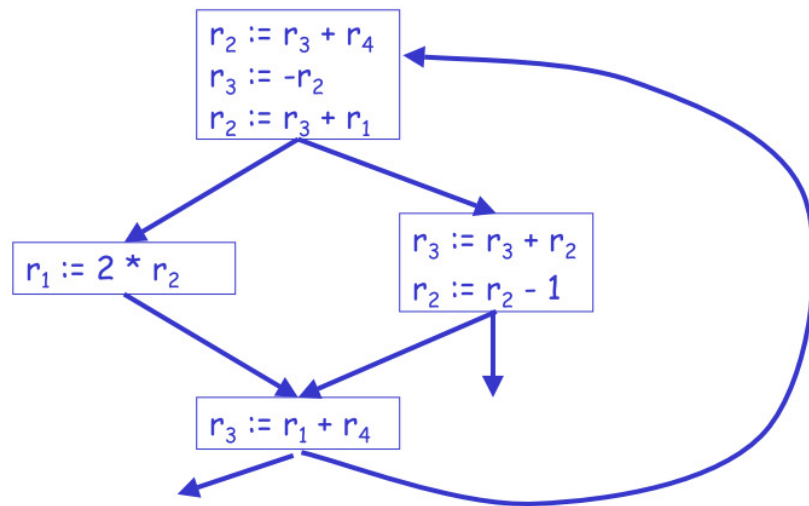
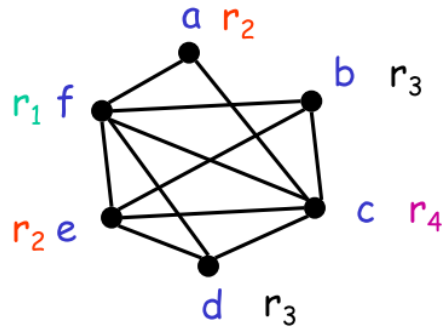
- Uma aresta entre dois temporários se estiverem vivos simultaneamente, em algum ponto no programa
- Dois temporários podem ser alocados para o mesmo registrador se não houver uma aresta conectando-os

d) Resolver o problema de coloração de grafo (NP-completo, logo usar heurística)

e) Se o número de registradores for insuficiente, utilizar a memória para alguma variável e retornar a (c)

Ex:





3. Otimização de acesso a cache

For (j=0; j<N; j++) For (i=0; i<N; i++) x[i,j]=0;

4. Otimização de pipeline

Objetiva explorar o paralelismo de instruções da arquitetura.