

# **POC II - Apresentação Oral**



**Implementação de um Algoritmo  
para Alocação de Registradores  
com Informações sobre  
Escalonamento de instruções  
Embutidas**

*Nizam de Abreu Pfeilsticker*

# MOTIVAÇÃO



**Novas arquiteturas se baseiam na simplicidade de seu conjunto de instruções e cabe ao compilador a complexa tarefa de otimização de código.**

# OBJETIVO



**Implementar um algoritmo de alocação de registradores que não restrinja a ação do escalonador e que utilize bem os recursos das máquinas superescalares**

# ALGORITMO DE PINTER



Construções dos seguintes

Grafos:

- **Interferência**
- **Escalonamento**
- **Complemento**
- **Interferência Paralelizável**

# GRAFO DE COMPLEMENTO

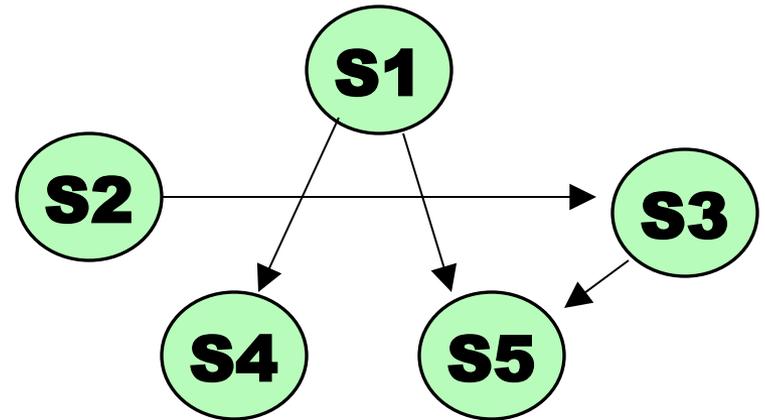
$S1 := load\ Z$

$S2 := i$

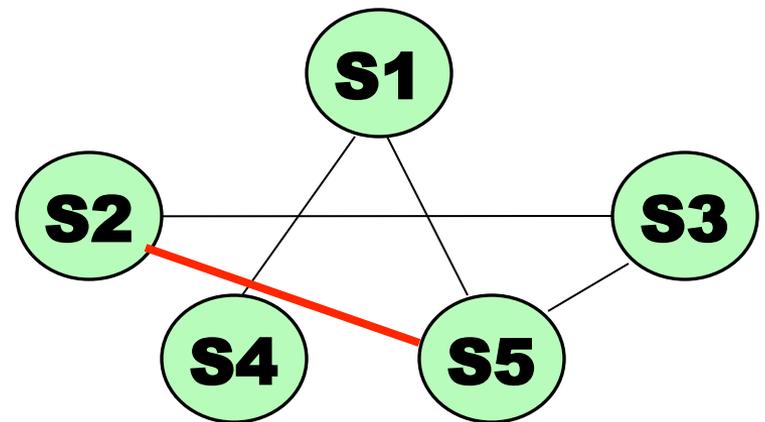
$S3 := a[S2]$

$S4 := S1 + S1$

$S5 := S3 * 5 + S1$

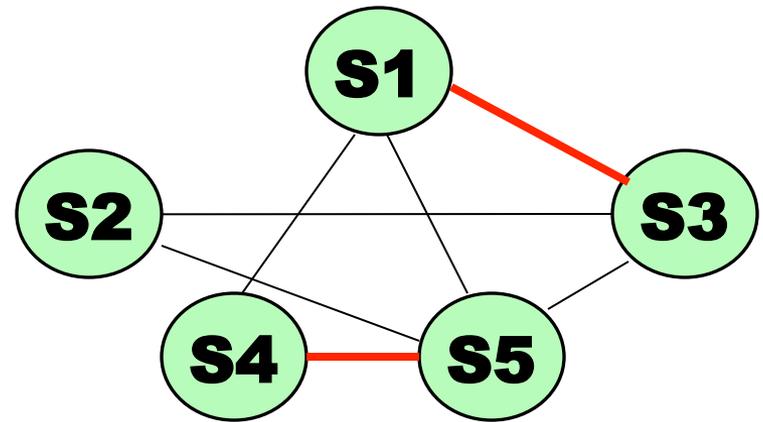


Passo 2: Faz-se o fecho transitivo do grafo de escalonamento e removem-se as direções das arestas.

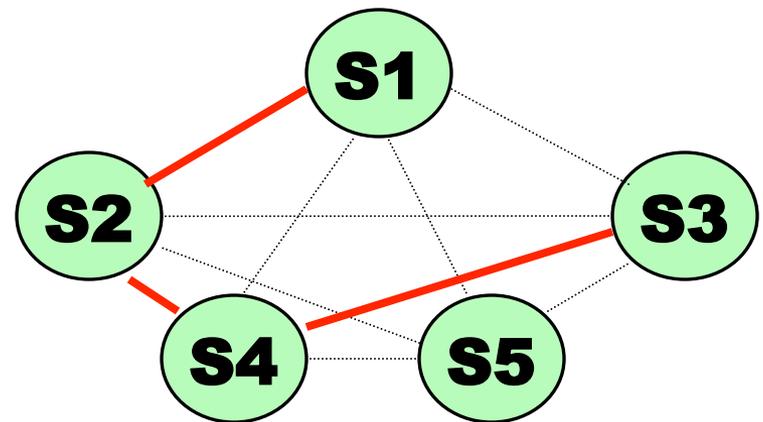


# GRAFO DE COMPLEMENTO

Passo 3: Insere-se arestas entre instruções que são dependentes da máquina, que não de precedência.



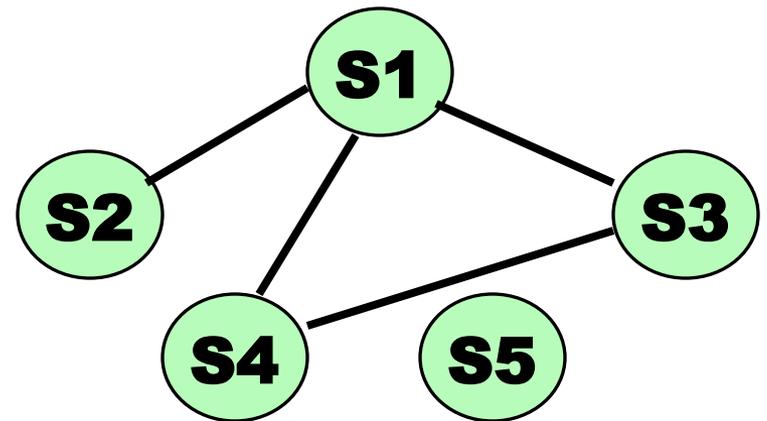
Passo 4: Preenche-se as arestas que ainda não estão preenchidas e retira-se as demais



# GRAFO DE INTERFERÊNCIA

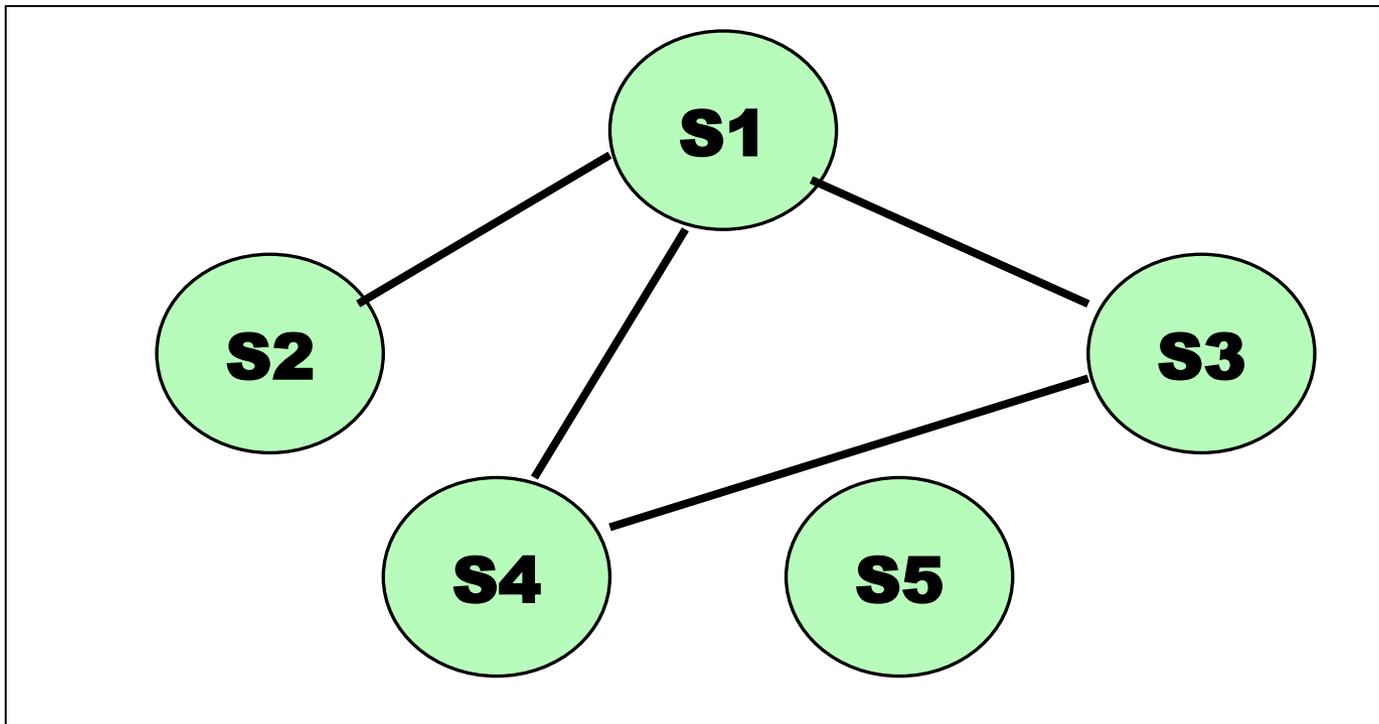
Devemos adicionar a S1, como seu último uso foi em S5, arestas a S2, S3 e S4; a S2 não devemos adicionar novas arestas; a S3 devemos adicionar a aresta (S3,S4); a S4 e S5 não devemos adicionar arestas novas.

```
S1 := load Z
S2 := i
S3 := a[S2]
S4 := S1 + S1
S5 := S3 * 5 + S1
```



*Grafo de Interferência*

# GRAFO DE INTERFERÊNCIA PARALELIZÁVEL



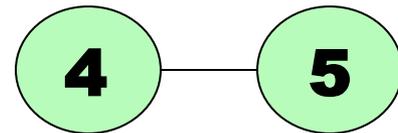
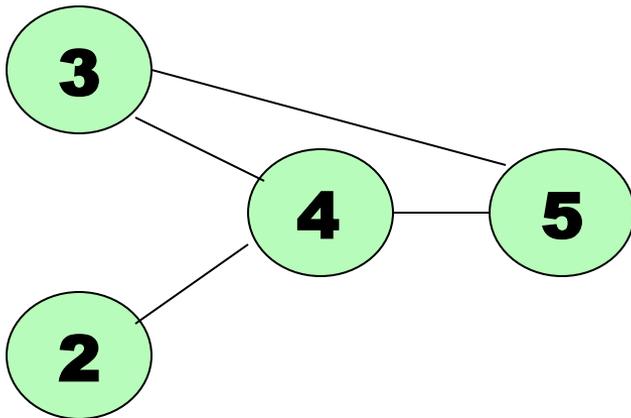
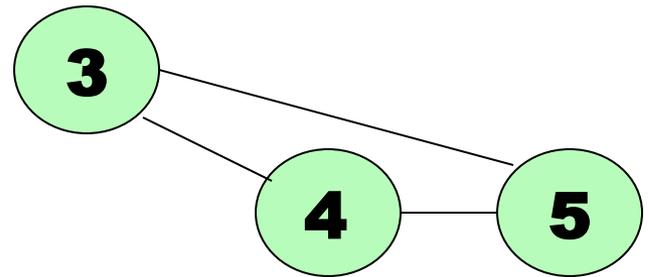
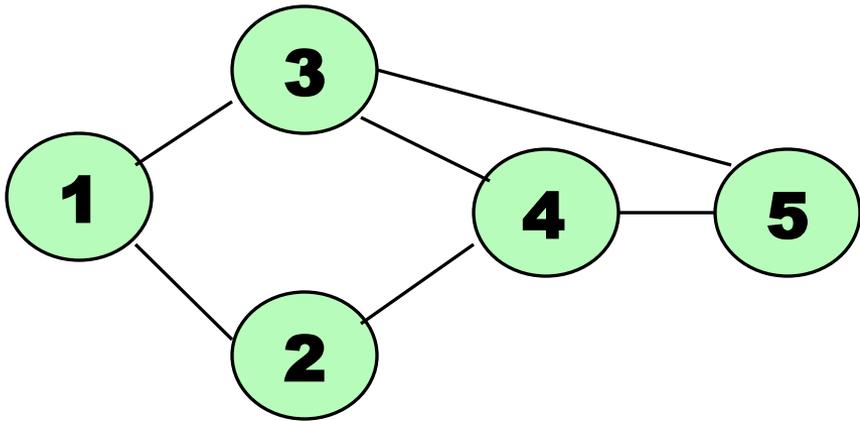
*Grafo de Interferência Paralelizável*

# COLORAÇÃO DE GRAFOS



- **Empilha-se os nós dos grafos, retirando-os deste repetidamente;**
- **Se o nó tiver  $K$  ou mais vizinhos, onde  $K$  é o número de registradores físicos do processador marca-se que aquele nó terá um derramamento para a memória;**

# COLORAÇÃO DE GRAFOS

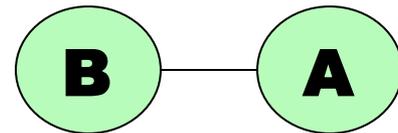
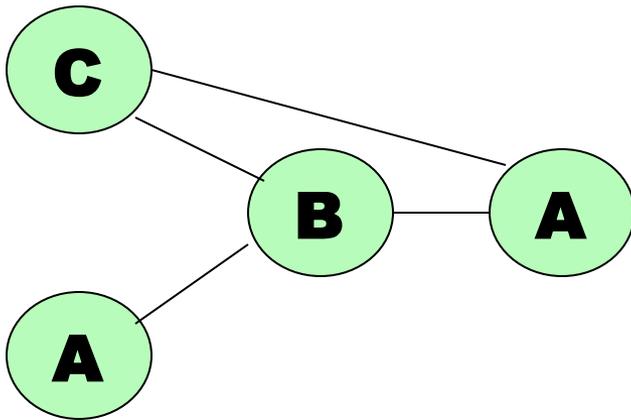
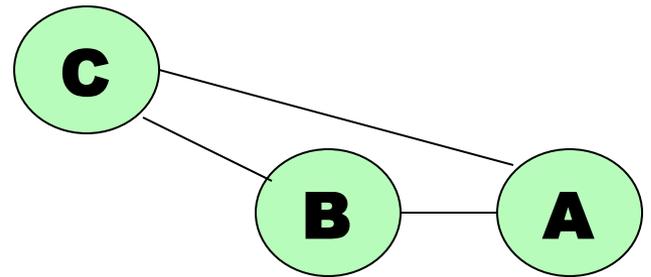
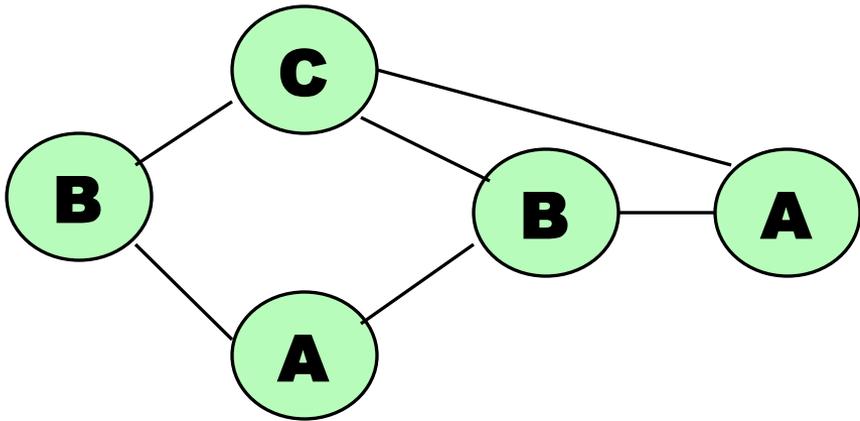


# COLORAÇÃO DE GRAFOS



- **Desempilha-se os nós, devolvendo-os para o grafo de maneira que deve se colorir cada nó com a primeira cor livre, que não está sendo utilizada por nenhum vizinho;**
- **Se o nó estiver com marca de *spill*, deve-se gerar código para o derramamento imediatamente;**

# COLORAÇÃO DE GRAFOS



# CONCLUSÃO



**Com este novo grafo contendo as informações sobre escalonamento de instruções embutidas pode se usar o algoritmo de coloração de grafo para a alocação de registradores e o algoritmo de lista para o escalonamento de instruções.**

**E mais importante, a alocação de registradores pode ser feita antes do escalonamento de instruções, sem prejudicar o paralelismo existente.**