

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA



PUC Minas

Compiladores

Integrantes

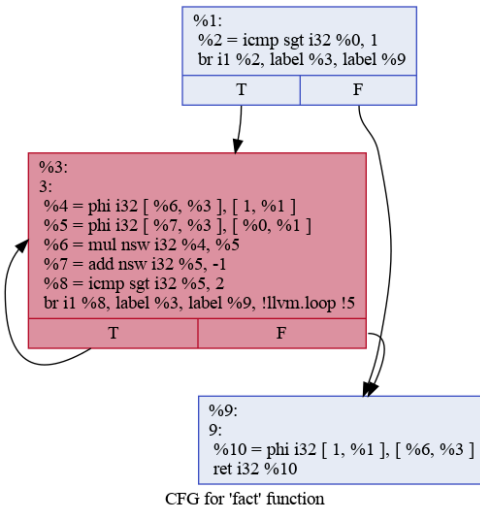
Sarah Luiza de Souza Magalhães

Análise dos grafos

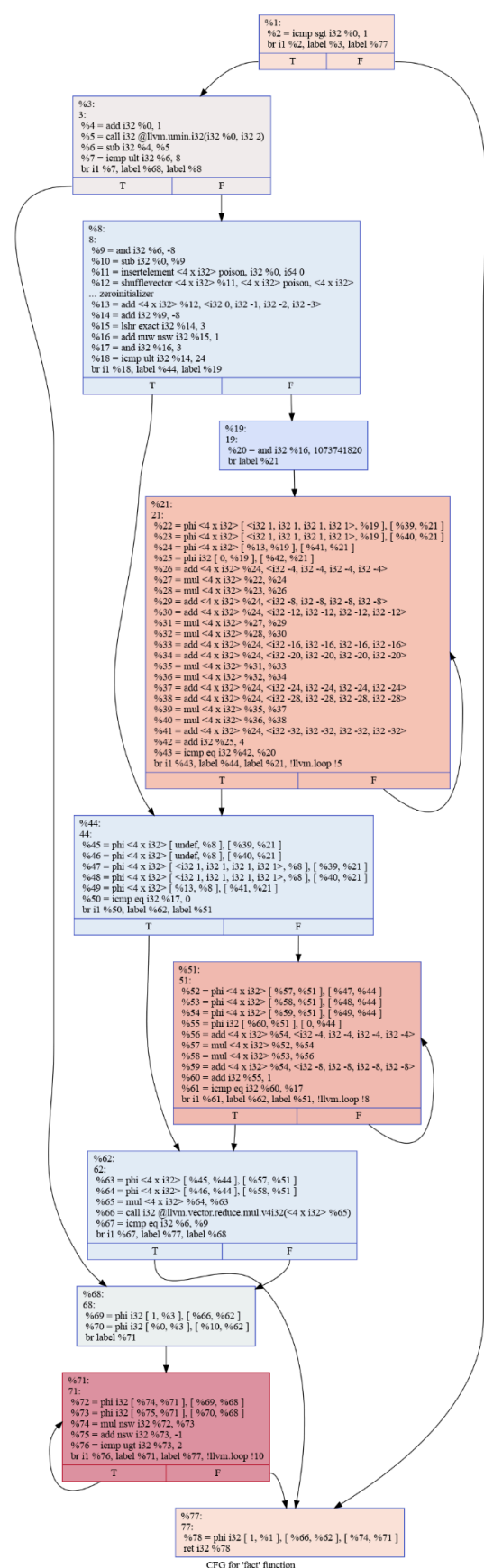
1- Programa de fatorial, referente a função “fact” (src/fact/fact.c)

Obs**: As imagens deste documento podem ser adquiridas no programa através do seguinte comando: `python3 gen_cfg.py src/fact/fact.c`

Grafo fact otimização -01



Grafo fact otimização -02



Comparação:

Complexidade:

Fica claro a diferença de complexidade entre os dois grafos, sendo o primeiro bem mais simples que o segundo.

Estruturas de Laços

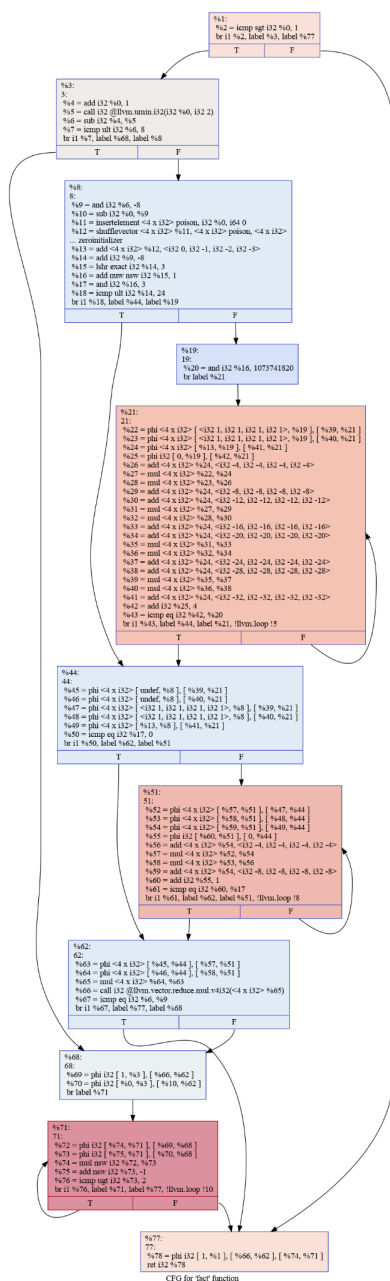
No CFG de **-O1**, o laço principal é direto, sem muitas transformações adicionais. Já no CFG de **-O2**, há **desenrolamento de laços** (loop unrolling) parcial, adicionando múltiplos nós para representar várias iterações de uma só vez. Isso melhora o desempenho em tempo de execução ao reduzir sobrecargas de controle de laços.

Operações Vetoriais

No CFG de **-O2**, operações vetorizadas são introduzidas (exemplo:

llvm.vector.reduce.mul.v4i32). Isso indica que o compilador tentou paralelizar operações utilizando registradores vetoriais, algo inexistente no CFG de **-O1**.

Grafo fact otimização -O3



Eficiência

O nó de saída (retorno) no CFG de -O3 é ajustado para processar a maior quantidade de dados previamente calculados, reduzindo operações finais.

Isso contrasta com os CFGs de -O1 e -O2, que mantêm estruturas mais básicas no nó final.