

Concurso/Encontro Nacional de Programação Lógica CeNPL'02

Universidade de Coimbra / Instituto Politécnico de Coimbra

11–13 de Abril de 2002

Problema nº5 ALOCADOR DE REGISTOS

Introdução

Na construção de compiladores, um sub-problema comum e bem identificado é o da geração de código para expressões aritméticas. As arquitecturas reais terem um número limitado de registos, facto que condiciona o código que pode ser gerado. Um compilador deverá saber qual o número de registos de que dispõe para calcular as expressões intermédias.

Tarefa

Considere que estamos a gerar código para um PDP-11 glorificado (ie. com número arbitrário de registos), que dispõe das seguintes instruções:

Instrução	Acção	Observações
mov SRC, DST	$DST \leftarrow SRC$	Copia SRC para DST.
add SRC, DST	$DST \leftarrow DST + SRC$	Efectua uma operação aritmética.
sub SRC, DST	$DST \leftarrow DST - SRC$	
mul SRC, DST	$DST \leftarrow DST * SRC$	
div SRC, DST	$DST \leftarrow DST / SRC$	

Os operandos poderão ser **registos**, da forma rN (p/ex. $r0$, $r7$), **constant**es da forma $\#NNN$ (p/ex. $\#12$) ou **nomes**, que obedecem à sintaxe habitual para os identificadores.

Pretende-se determinar o número mínimo de registos com os quais se consegue avaliar uma determinada expressão sem “spilling,”¹ produzindo neste processo uma sequência de instruções que avalie a expressão, colocando o resultado no registo $r0$.

O algoritmo a aplicar pode ser resumido da seguinte forma: considere que se pretende gerar código para uma expressão $E = E_L \circ E_R$, em que E_R e E_L são expressões e \circ é uma operação aritmética. Seja N_L e N_R o número de registos necessários à avaliação de E_L e E_R respectivamente, o número de registos para avaliar E será dado por $N' = \max(N_L, N_R + 1)$ se

¹ie. sem colocar resultados intermédios em localizações temporárias, na memória principal.

avaliarmos E_L primeiro e guardarmos o seu valor num registo enquanto avaliarmos E_R . Se optarmos por avaliar E_R primeiro, precisaremos de $N'' = \max(N_L + 1, N_R)$ registos. O objectivo será escolher a ordem de avaliação que resulte no menor valor de N (de entre N' e N''), preservando a correcção do resultado (a operação deverá ser comutativa).

Notas:

- Para avaliar uma expressão trivial (constante ou nome) precisamos de 0 registos pois estas podem ser utilizadas directamente em operações.
- Depois de avaliar uma subexpressão, todos os registos usados nesse processo (excepto aquele em que ficou o resultado) estão disponíveis para avaliar outra subexpressão.
- **Atenção:** aplicam-se as regras de comutatividade habituais.

Os Dados

Pretende-se obter N , tal que sejam necessários N registos distintos (de r_0 a r_{N-1}) para avaliar uma expressão aritmética E . E será representada por um termo Prolog “ground”, encarado como uma árvore cujas folhas poderão ser átomos (designam os *nomes*) ou números (designam as *constantes*) e cujos nós interiores corresponderão a uma operação aritmética (+/2, -/2, */2 e '/'/2). Deverá ser implementado o predicado `sethi(+EXPR, ?NREGS)` que produz como output uma sequência de instruções do PDP-11 correspondente à avaliação de `EXPR` com `NREGS` registos, colocando o resultado final no registo `r0`. `NREGS` será instanciado, caso esteja livre à entrada

Os Resultados

Exemplos de execução:

<pre> ?- sethi(a+7*b, NR). mov #7,r0 mul b,r0 add a,r0 NR = 1 yes ?-</pre>	<pre> ?- sethi((a+3*b-4/c)*(7-8*d), NR). mov #3,r0 mul b,r0 add a,r0 mov #4,r1 div c,r1 sub r1,r0 mov #7,r1 mov #8,r2 mul d,r2 sub r2,r1 mul r1,r0 NR = 3 yes ?-</pre>
--	--
