

Compilerbouw

Peephole optimizer

Alexandra Moraga Pizarro (6129544)
Tamara Ockhuijsen (6060374)
Fredo Tan (6132421)

31 december 2011

1 Inleiding

Dit verslag beschrijft de uitwerking van de praktische opdracht voor het vak Compilerbouw, gegeven aan de Universiteit van Amsterdam in het jaar 2011/2012. Voor deze opdracht is een peephole optimizer voor SimpleScalar DLX assembly code ontwikkeld. Er zijn verschillende benchmark programma's in de taal C meegegeven. Een gcc cross compiler kan deze C-code compileren in assembly code die dient als invoer voor de peephole optimizer. Deze leest de assembly code en genereert nieuwe assembly code waarin overbodige instructies zijn verwijderd. De functionaliteit blijft hetzelfde als die van de originele assembly code, maar het programma werkt wel sneller.

2 Implementatie

De PLY package is gebruikt voor het parsen van assembly code. Dit is een ingebouwde implementatie van lex en yacc voor Python.

De implementatie is onderverdeeld in het parsen met lex en yacc en de optimalisatie.

2.1 Lex

2.2 Yacc

2.3 Optimalisatie

De volgende optimalisaties zijn toegepast op de assembly code.

Original sequence	Replacement
mov \$regA,\$regB	---
mov \$regA,\$regB	
instr \$regA, \$regA,...	instr \$regA, \$regB,...
instr \$regA,...	instr \$4,...
mov \$4, \$regA	jal XXX
jal XXX	

sw \$regA,XXX	sw \$regA, XXX
ld \$regA,XXX	
shift \$regA,\$regA,0	---
add \$regA,\$regA,X	lw ...,X(\$regA)
lw ...,0(\$regA)	
beq ..., \$Lx	bne ..., \$Ly
j \$Ly	\$Lx:
\$Lx: ...	

3 Resultaten

4 Conclusie

5 Referenties

http://www.science.uva.nl/~andy/compiler/users_guide_v2.pdf
<http://www.dabeaz.com/ply/>
<http://staff.science.uva.nl/~andy/compiler/prac.html>