

Discussions on Peephole Optimizers in C and in Perl

Cecilia Chao Chen
CSc 71010 Presentation

Peephole Optimizer

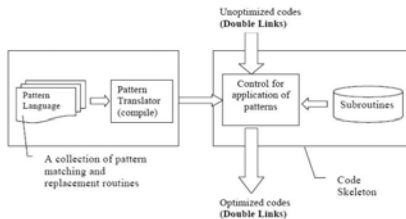
- Compiled machine code needs to optimize


```
mov%0 $0, %1    =>  clr%0 %1

mov%0 %1, r%2,  mov%0 r%2, %3  =>  mov%0 %1, %3
```
- Implementation
 - Quick Compilers Using Peephole Optimization (C)
 - Declarative Peephole Optimization Using String Pattern Matching (Perl)

2

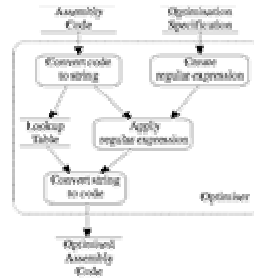
Peephole Optimizer in C



- Rule-derived; one C program per rule

3

Peephole Optimizer in Perl



- Fast – work on tokenized code
- Declarative – easy to maintain rules
- Perl's powerful pattern matching – short program

4

Problems

- Rules interacting


```
mov%0 $0, %1    =>  clr%0 %1
mov%0 %1, r%2,  mov%0 r%2, %3  =>  mov%0 %1, %3
Movl $0, r3
Movl r3, i.(fp)  =>  ?
```
- Solution
 - Add another rule
 - Delay optimization – match till more contexts
- Too many rules → too much time on sifting
→ speed affected by infrequent rules

5

Conclusion

- Conclusion
 - Simple task
 - Different implementations
 - Improved compilers
 - Tradeoff: optimality vs. speed
- Other approaches
 - Global optimizer + improving optimizer

6