Semester

VI

Subject Code

CS603 (C)

Subject Name

Compiler Design

Unit-5

Topic: Peephole Optimisation

As Per
RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL
(www.rgpv.ac.in)
New Scheme Based on AICTE Flexible Curricula
Computer Science and Engineering

SIRT
Sagar Institute of Research & Technology
9001-2008 ISO Certified Institute of MP

Mr Sandeep Wadekar
Assistant Professor
Department of Computer Science and Engineering
SAGAR INSTITUTE OF RESEARCH AND TECHNOLOGY BHOPAL

PEEPHOLE OPTIMIZATION

Dec-2002, Dec-2003, June-2004 Dec-2004, 05, 06 June-2007 Dec-2007 Dec-2008 Dec2010, June-Dec-2011, Dec-2013, Dec-2015, 16, 17, 18, 19

If the code is generated statement wise then it consist several redundancies and suboptional construct. In order to remove such redundancies in a code for increasing efficiency we need optimization over a target code. Peephole optimization is a technique by which we can locally optimise a code.

In Peephole Optimization we replace shorter and slower sequence with faster sequences according to the available possibilities.

Characteristics of Peephole Optimization -

The Peephole Optimization can be applied on the target code using following characteristics -

1. Elimination of redundant load and store instruction
2. Elimination of multiple jumps. / How to control optimization
3. Elimination of unreachable code.
4. Algebraic simplifications.
5. Reduction for strength improvement.
6. Use of machine idioms.

1. Elimination of Redundant load and store instruction -

Example of redundant loads and store like the instruction sequence.

(a) MOV R_0, a

(b) MOV a, R_0

In the above, we can eliminate second one (b).

Since 'a' is already in 'R_0'

Mr Sandeep Wadekar
Assistant Professor
Department of Computer Science and Engineering
SAGAR INSTITUTE OF RESEARCH AND TECHNOLOGY BHOPAL

2. Elimination of Multiple Jumps -

Multiple unnecessary jumps are avoided to increase efficiency and speed of execution.

Example-

```
if (a<b) go to L1
- - - - - -
- - - - - -
L1 : go to L2
- - - - - -
- - - - - -
L2 : a = a+3;
```

⟹

```
if (a<b) go to L2
- - - - - -
- - - - - -
L2 : a = a+3;
```

3. Elimination of Unreachable Code-

Unreachable means the code which does not lies in the instructions boundaries. Any unlabeled instructions which immediately appears after the unconditional jumps can be removed. For debugging purpose unreachability always avoided.

Example-1)

```
Sum = 0
if (sum)  ⟶ Unreachable code
printf ("%d", sum);
```

In above example, "if statement" will never get executed, hence can be eliminated.

Example-2)

```
int fun (int a, int b)
{
   c = a+b;
   return c;
   printf ("%d", c);  ⟶ Unreachable code
}
```


Mr Sandeep Wadekar
Assistant Professor
Department of Computer Science and Engineering
SAGAR INSTITUTE OF RESEARCH AND TECHNOLOGY BHOPAL

4. Algebraic Simplification -

Useless algebraic calculations always waste memory in the program. Such instructions always be simplified to increase execution speed in a code.

Example -

$A = A+0$

or

$A = A * 1$

These are generated by intermediate code generation - algorithm and can be eliminated through peephole optimization.

5. Reduction of Strength Improvement -

To increase efficiency of the code we can replace costlier instruction with the cheaper instruction.

Example -

1. $x = y ** 2$ can be replaced with $x = y * y$
2. x^2 is cheaper than $x * x$
3. Addition and Subtraction is cheaper than multiplication and division.

6. Use of Machine Idioms -

Some machines having specific hardware instruction to implement certain operations efficiently. Some machine contain also to increment and decrements modes with the help of counters in addressing modes. These machine idioms are very helpful to save the memory.

Example -

- $a = a+1$
- $a = a-1$

Mr Sandeep Wadekar
Assistant Professor
Department of Computer Science and Engineering
SAGAR INSTITUTE OF RESEARCH AND TECHNOLOGY BHOPAL