Register Allocation Based on Boolean Satisfiability

Yang Mengmeng and Liu Jie

Software College, Shenyang Normal University, 110034 Shenyang, China
mengmengmelody.yang@gmail.com, nan127@sohu.com

Abstract. Graph Coloring is an effective method which is used to solve the register allocation problem, it is also an NP-complete problem, heuristic algorithms and various evolutionary algorithms have been proposed in order to improve the performance of register allocation, in this paper, we propose to solve this problem by converting the graph coloring problem into Boolean Satisfiability problem (SAT), the experiments show that our algorithm can use fewer number of registers, which can improve the execution efficiency of the generated codes.

Keywords: Register Allocation, Graph Coloring, SAT Problem.

1 Introduction

Register allocation is an important stage before the compiler outputs code. It is the process of assigning the intermediate variable to the available registers. It decides the register that each variable should be put into, it is also related to the performance of the generated code, so many compilers using a variety of register allocation algorithms, hoping to improve the efficiency of the compiler, the traditional register algorithms are mostly based on graph coloring algorithm.

Given an undirected graph G (V, E), V represents the set of vertices, E represents the set of all edges. Color all the vertices of a graph G, and any two connected vertices cannot be colored by the same color, which is known as graph coloring. If the number of colors in a coloring scheme does not exceed k, it is called k coloring.

The idea of using graph coloring to solve register allocation problem was first proposed by Chaitin [1] et al. Graph G represents limitations and conflicts in the process of allocating, named register interference graph, all vertices of the graph represents the liveness range, and the edge between vertices represents the conflict among liveness range. Two conflicting lifetime cannot be active at the same time, that is, two vertices of each edge cannot use the same color. The problem that we need to solve is how to find a smallest possible value k to color graph G.

Various solutions have been proposed for graph coloring problem, Gend Lal Prajapati [2] proposed to divide the neighbor nodes of each vertex into two sets, set A contain vertices which are connected to the vertex w, and set B includes vertices which are not connected to the vertex w. Find the color which is different from all the colors that have been used in the connected vertices from the first element of the color

J.-S. Pan et al. (eds.), *Intelligent Data Analysis and Its Applications*, *Volume 2*,
Advances in Intelligent Systems and Computing 298,
DOI: 10.1007/978-3-319-07773-4_27,

set to reduce the number of colors; WDChen [3] proposed to select the color value of the maximum potential node v first, and to use the smallest number of available color. Many heuristic algorithms [7-11] also have been used to solve the graph coloring problem , Carla Negri Lintzmaye [4] and others get a good solution with Ant Colony Optimization, the method ensures the quality of solution, but it costs a high running time. Later on they used evolutionary algorithms [5] to solve the same problem, on the base of maintaining the original performance the proposed algorithm reduced the running time and improve the efficiency of the algorithm. Raja Marappan [6] and others proposed to solve this problem using genetic algorithm, which applies fitness-proportionate selection method, Single Parent Conflict Gene crossover (SPCGX) and conflict-edge mutation operators to reduce the search space, but it requires several initial populations which affects the efficiency and quality. There are also some other methods to solve this problem [12-14].

In this paper, we propose to solve this problem by converting graph coloring problem into SAT problem and compare this method with the Kempe's algorithm, the experiments demonstrate that we can get a smaller k value through the proposed algorithm.

The second part describes the Kempe's algorithm; our algorithm is proposed in the third part; In part 4, we report our experimental results; the last part gives the conclusion.

2 Kempe's Algorithm

Kempe's algorithm is a classic algorithm which is used to solve the problem of k graph coloring, the specific steps are:

Step1: find a vertex whose degree is less than k, push this vertex onto the stack s, at the same time, move this vertex and its all vertices from the graph G.
Step2: Repeat step1 until all the nodes are piled into stack s and graph G becomes empty FIG.
Step3: Pop the vertex w from the stack s and give w a color that is different from the color of its neighbors.
Step4: Repeat step3 until all vertices in the stack are colored.
Simple example:

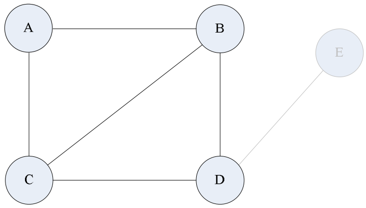


Fig. 1.

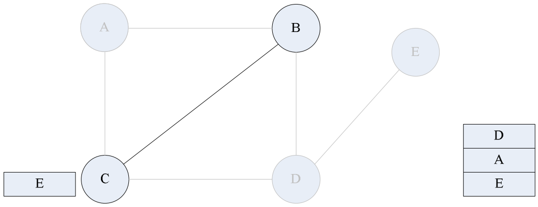


Fig. 2.

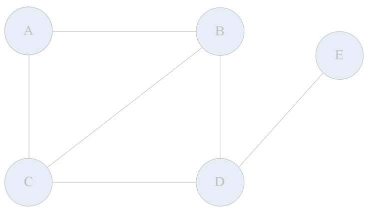


Fig. 3.

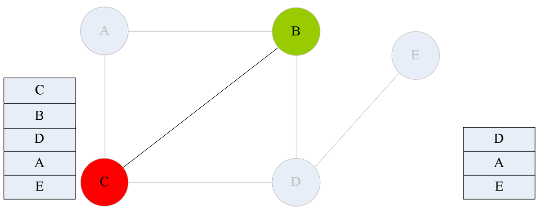


Fig. 4.

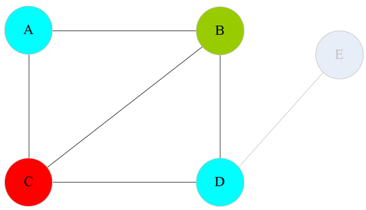


Fig. 5.

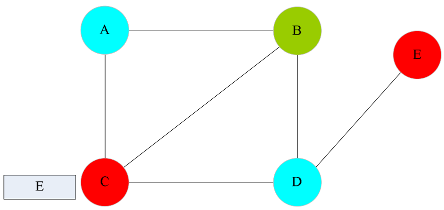


Fig. 6.

Theorem. If we can color a graph G=(V, E) using k colors based on Kempe's algorithm, then G is colorable using k colors.

Proof. We use induction on |V|.
(Base Case): Suppose |V| = 0. This theorem is trivially true.
(Induction Step): Suppose this theorem is true for all |V| = n-1, we want to show that this it is also true for |V| = n. According to our induction hypothesis, the first n-1 vertices can be colored successfully, assume that the vertex at the bottom of the stack is v_i. According to Kempe's algorithm, the degree of v_i is less than k. Let us also assume that (in the worst case) each of v_i's adjacent vertices is assigned a different color, we can still color vi using any of the remaining colors. Hence all of the vertices in V can be colored using k colors.

3 Our Algorithm

3.1 SAT Program

SAT problem is the satisfiability problem in propositional logic , given a CNF:

$$A = B_1 \wedge B_2 \wedge \ldots \wedge B_n \tag{1}$$

The form of clause :

$$Bi = b_1 \vee b_2 \vee \ldots \vee b_m \tag{2}$$

Where bi is a Boolean variable or the negation of the Boolean variable.

SAT problem is to find some assignment to variables so that the conjunctive normal form A is true. That is, if each clause of CNF A has at least one Boolean variable being true, A is true, otherwise A is false.

3.2 Specific Operation

Here are the specific steps of the proposed algorithm :
Step1: Convert register allocation problem into a register interference graph G (V, E);
Step2: According to the graph coloring constraints convert the graph into logical formulas and output in the form of a text file;

Where the constraints are:

1. Each vertex must select a color from the set {1,2 ... k};
2. Any two colors cannot be assigned to the same vertex, that is, each vertex can choose a unique color;
3.Two adjacent vertices cannot be assigned the same color.

Converted into a logical language:

$$1.\begin{cases} V(1,1) \vee V(1,2) \vee ... \vee V(1, K) \\ ... \\ V(n,1) \vee V(n,2) \vee ... \vee V(n, K) \end{cases} \tag{3}$$

$$2.\begin{cases} \neg V(1,1) \vee \neg V(1,2) \\ \neg V(1,1) \vee \neg V(1,3) \\ ... \\ \neg V(\mathrm{n}, \mathrm{k}-1) \vee \neg V(n,k) \end{cases} \tag{4}$$

If a and b are two connected vertices, then:

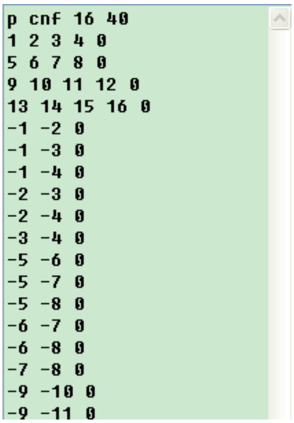
$$3.\begin{cases} \neg V(\mathrm{a},1) \vee \neg V(\mathrm{b},1) \\ ... \\ \neg V(\mathrm{a},k) \vee \neg V(\mathrm{b},k) \\ ... \end{cases} \tag{5}$$

Where V(i,j) represents the i-th vertex having the j-th color and n is the number of vertices.
Step3: Use minisat to solve the satisfiability problem.
The pseudo code of the algorithm:

```
Given an undirected graph G = (V, E), k-coloring
if (There is an edge between V1 and V2)
    S[V1,V2]=1;
```

```
else
    S[Vi,Vj]=0;
FILE *fpt;
fpt = fopen("SAT.txt","w");
fprintf(fpt,"p cnf %d %d\n",N,M);
for i=0 to n-1 do                //i represents vetex
  for c=1 to k do          //c represents color
    fprintf(fpt,"%d ",i*k+c);    fprintf(fpt,"0\n");
// Constraint 1
for i=1 to n-1 do
  for j=1 to k-1 do
      for c=j+1 to k do
        fprintf(fpt,"-%d -%d 0\n",i*k+j,i*k+c);
//Constraint 2
for i=0 to n-2 do      //n represents the number of vetices
  for j=i+1 to n-1 do
    if S[i][j]==1  then
      for c=1to k do          //c represents color
        fprintf(fpt,"-%d -%d 0\n",i*k+c,j*k+c);
//Constraint 3
fclose(fpt);
```

Text document shown in figure 7:



```
p cnf 16 40
1 2 3 4 0
5 6 7 8 0
9 10 11 12 0
13 14 15 16 0
-1 -2 0
-1 -3 0
-1 -4 0
-2 -3 0
-2 -4 0
-3 -4 0
-5 -6 0
-5 -7 0
-5 -8 0
-6 -7 0
-6 -8 0
-7 -8 0
-9 -10 0
-9 -11 0
```

Fig. 7. Text document

4 Experimental Results and Analysis

This paper compared the proposed algorithm with Kempe's algorithm, which is a classic graph coloring algorithm. We use C to implement these two algorithms and test the performance of the algorithms, the experimental results are shown in Table 1:

Table 1. Experimental results

vertex	edge	Maximum degree	Kempe's algorithm	Proposed algorithm
10	25	8	5	5
20	93	14	6	5
30	218	20	9	7
40	400	26	11	8
50	605	32	13	9
100	2478	64	19	16
150	5616	91	28	23

As can be seen from the table, compared to Kempe's algorithm the proposed algorithm can obtain a smaller k value, that is to say our algorithm can use fewer colors to color graph. With the increase of vertices, edges, and the degree, the proposed algorithm show more obvious advantages, that is, using our algorithm, the same number of registers can be used to store more variables to save storage space.

In addition to getting a smaller k value and using fewer colors to do graph coloring, our method is very simple, we just use minisat to solve the problem.

5 Conclusion

Register allocation is an important step in the process of compiling codes, the quality of the register allocation algorithm determines the quality of the compiled code and the efficiency of the compiler, graph coloring is widely used to solve the problem of register allocation, the traditional algorithms view graph coloring algorithm as a combinatorial optimization problems and use heuristic to solve it, while this paper converts constraints in graph coloring to Boolean formulas and solve the problem by existing sat solver. This method uses a smaller number of registers, which can effectively improve the performance of register allocation, improve the quality of the compiler output code, experimental results demonstrate the effectiveness of our algorithm.

References

1. Chaitin, G.J., Auslander, M.A., Chandra, A.K., Cocke, J., Hopkins, M.E., Markstein, P.W.: Register Allocation via Coloring. Computer Languages 6(1), 47–57 (1981)
2. Gend, L.P., Amit, M., Neha, B.: An Efficient Colouring of Graphs Using Less Number of Colours. In: World Congress on Information and Communication Technologies, pp. 666–669 (2012)
3. Chen, W.D.: New Algorithms for Graph Coloring Problem. Microcomputer Applications 25(4), 391–395 (2004)
4. Carla, N.L., Mauro, H.M., Anderson, F.: Register Allocation with Graph Coloring by Ant Colony Optimization. In: International Conference of the Chilean Computer Science Society, pp. 247–255 (2011)
5. Carla, N.L., Mauro, H.M., Anderson, F.: Register Allocation by Evolutionary Algorithm. In: International Conference of the Chilean Computer Science Society, pp. 207–215 (2012)
6. Raja, M., Gopalakrishnan, S.: A New Genetic Algorithm for Graph Coloring. In: International Conference on Computational Intelligence, Modelling and Simulation, pp. 49–54 (2013)
7. Dorrigiv, M., Markib, H.Y.: Algorithms for the graph coloring problem based on swarm intelligence. In: CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), pp. 473–478 (2012)
8. Ge, F.Z., Wei, Z., Tian, Y.M., Huang, Z.J.: Chaotic ant swarm for graph coloring. In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), pp. 512–516 (2010)
9. Yesil, C., Yilmaz, B., Korkmaz, E.E.: Hybrid local search algorithms on Graph Coloring Problem. In: International Conference on Hybrid Intelligent Systems (HIS), pp. 468–473 (2011)
10. Consoli, P., Collera, A., Pavone, M.: Swarm Intelligence heuristics for Graph Coloring Problem. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1909–1916 (2013)
11. Fister, I., Brest, J.: Using differential evolution for the graph coloring. In: IEEE Symposium on Differential Evolution (SDE), pp. 1–7 (2011)
12. Falk, H.: WCET-aware register allocation based on graph coloring. In: ACM/IEEE Conference on Design Automation, pp. 726–731 (2009)
13. Mittal, A., Jain, P., Mathur, S., Bhatt, P.: Graph Coloring with Minimum Colors: An Easy Approach. In: International Conference on Communication Systems and Network Technologies (CSNT), pp. 638–641 (2011)
14. Wang, X.H., Qiao, Q.G.: Solving Graph Coloring Problems Based on a Chaos Nueral Network with Non-monotonous Activation Function. In: Fifth International Conference on Natural Computation, ICNC, vol. 1, pp. 414–417 (2009)