ITP 20002-0X Discrete Mathematics, Fall 2020

Homework 2

Wonbin Lee, 21900543, 2190543@handong.edu

1. Introduction

The tasks are two. One is constructing each of set data structure (Array-based, Bitvector-based), and the other is solving Minesweeper puzzle using one of two set data structures.

First, array-based data structure consists of n_elems and elems. n_elems means the number of elements in the set, and the set is elems. With this data structure, I should implement various operation including set operation. I implemented all of given operations, and distinguished the given input set(that can have duplicates) and the unique set (that only have each one member, kind).

Second, bitvector-based data structure consists of four elements. The two things are about universal, and the other two things are about bitvector representing whether the set has certain elements in universal. To say specifically, they are univ, n_univ, bitvector, and n_emlems. I implemented almost operations using bitwise operation, except the two functions about subsets.

Last, Minesweeper puzzle consists of n x n grid, and one or multiple cells may have invisible mines. Among the given input, there can be a assigned number (0~8) in a certain cell, and it means there are mines corresponding to the number. I should implement this puzzle solving program with using one of two set data structures that I implemented. However, there was a serious error on my code, and I couldn't do debugging after all.

2. Approach

First is implementing array-based data structure and its operations. First of all, I thought there could be duplicates in given set (from test case) so I get a set of non-redundant elements (unique set – that I named). By doing that, implementing operations become very easy. The first one is getting size. I just counted the total elements of unique set. On add, remove operations, I just checked whether the given integer as argument already exist in the given set. Contain operation is same. On equal operation, I just checked whether the all elements in first set exist in the second set and the opposite. On union, intersection, difference operation, I used contains operations and add operations. For example, on intersection operation, if the one element of the first set exist in the second set, I added it to the intersection set. Finally, getting subsets operations, subsets and powerset, I used recursion for implementing them. The initial case is when the size of given set (or given size) is 0. On that case, I returned the subset array that contains only a empty set. The general case is when the size of given set is bigger than 0. For this, I picked one of given set and make it as one of a subset, and call the same function but the parameter are the set of difference, given set and picked one set, and the size is smaller by one. I used for loop for implementing all of elements of given set.

Second is implementing bitvector-based data structure and its operations. These is really similar to array-based data structure's operation. The only difference is, for using bitvect, I should use bitwise operations such as and, or, etc. It is enough to implement all operations.

The last is minesweeper puzzle. I approached this like following: while getting the input one by one, check whether the given input is in 0 ~ 8. If so, I make a intset consisting of the arounds cell, and make subsets with the number(input). I delivered it to z3. But it doesn't work.

3. Evaluation

For array-based data structure, the given test are accomplished. However the bitvector-based data structure, the subset and powerset operation doesn't work. Minesweeper is also doesn't work well, but the formula was created.

4. Discussion

I'm not sure whether data structures is more efficient.

5. Conclusion