Academic Misconduct in Computer Science

David W. Juedes

September 7, 2009

1 Introduction

Plagiarism is the *appropriation or imitation of the language, ideas, and thoughts of another author and representation of them as one's original work.*[1] Plagiarism can take many forms, and in the classroom context, most are considered **academic misconduct.** The penalty for such academic misconduct may consist of (i) failure in the class, (ii) referral to University Judiciaries, (iii) loss of credit for the assignment in question, or (iv) some other penalty. Acts of plagiarism are considered serious acts of academic misconduct because they, in essence, constitute academic fraud. However, like any issue surrounding academic misconduct, it is a difficult matter for professors to address.

This document is written to provide beginning students with an understanding of what constitutes plagiarism in an academic computer science setting, and it provides some guidance of how to avoid being accused of plagiarism. To begin, we will look at probably the most common form of plagiarism, i.e., plagiarism in writing.

2 Plagiarism in Writing

When plagiarism is mentioned in the news, it most frequently refers to plagiarism in writing. To understand this concept, consider an English assignment for a class. Perhaps the professor asked you to write a narrative the described the activities of children on a winter day. Perhaps you wrote the following:

[1]The Random House College Dictionary, *Random House*, 1988.

> On a cold winter day, John and his friend Larry went to the park. They brought their sleds and began to sled down the big hill in the park. John and and Larry had fun racing down the hill with their sleds. They had a lot of fun that day until John crashed into a tree. John broke his arm and had to be taken to the hospital. From that day forward, John vowed never to race again.

If someone else in the class turned in the same paragraph, one or both of you might be accused of plagiarism. Please note that if you provided another student with your solution to the assignment, that act itself may be considered academic misconduct.

Similarly, if another student in the class turned in a similar paragraph, such as

> On a cold winter day, Jim and his friend Harry went to the park. They brought their sleds and began to sled down the big hill in the park. Jim and and Harry had fun racing down the hill with their sleds. They had a lot of fun that day until Harry crashed into a tree. Harry broke his arm and had to be taken to the hospital. From that day forward, Harry vowed never to race again.

In this case, again, one or both of the students might be accused of plagiarism. The key in both cases is that the second student spent little if any effort writing their assignment. Simply copying someone else's work, or changing or rearranging the work of someone else is considered plagiarism. Please note that having someone else write a solution for you, or copying a solution from a book or from the Internet is also considered plagiarism.[2]

3 Plagiarism in Computer Science

Plagiarism in computer science most commonly occurs when students submit written programs as assignments. Consider the following program that was by the author for an assignment for CS 240A.

```
//**************************************************
// Student #1's code
```

[2]Please note that in certain cases, it is OK to provide verbatim copies of another person's work if the work is properly cited.

```
//
//
//**************************************************
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>

using namespace std;


class TicTacToe {
public:
  void print_board(ostream &out, const string &board) {
    out << board.substr(0,3) << endl;
    out << board.substr(3,3) << endl;
    out << board.substr(6,3) << endl;
  }
  bool valid_board(const string &board) {
    if (board.length()!=9) return false;
    for (int i=0;i<board.length();i++) {
      switch (board[i]) {
      case '_':
      case 'X':
      case 'O': break;
      default: return false;
      }
    }
    return true;
  }
  bool winning_board(const string &board, const char c) {
    assert(valid_board(board));
    bool diag_win = true;
    for (int i=0;i<3;i++) {
      if (board[i*3+i]!=c) {
        diag_win= false;
      }
    }

    bool diag_win1 = true;
    for (int i=0;i<3;i++) {
      if (board[(2-i)*3+i]!=c) {
        diag_win1= false;
      }
    }

    bool across_win = false;
    for (int i=0;i<3;i++) {
      bool t = true;
      for (int j=0;j<3;j++) {
        if (board[i*3+j]!=c) t=false;
      }
      if (t) across_win =true;
    }

    bool down_win = false;
    for (int i=0;i<3;i++) {
      bool t = true;
      for (int j=0;j<3;j++) {
        if (board[j*3+i]!=c) t=false;
      }
      if (t) down_win =true;
    }
    return diag_win || diag_win1 || across_win || down_win;
  }


  bool full(const string &board) {

    bool fullv = true;
    for (int i=0;i<board.length();i++) {
      if (board[i]=='_') fullv = false;
    }
    return fullv;
  }

  void mod_i_j(string &new_board, int i, int j, const char c) {
    if ((i<0) || (i>2)) return;
    if ((j<0) || (j>2)) return;
    new_board[i*3+j]=c;
  }

  bool valid_move(const string &old_board, const string &new_board, const char c) {
    assert(valid_board(old_board));
    if (!valid_board(new_board)) return false;

    int count = 0;
    for (int i=0;i<new_board.length();i++) {
      if (old_board[i]!='_') {
        if (old_board[i]!=new_board[i]) return false;
      } else {
        if ((new_board[i]!='_')) {
          if ((new_board[i]==c)) {
          count++;
        } else {
            return false;
          }
        }
      }
    }
    if (count!=1) return false;

    return true;
  }
};

int main() {
  ofstream out;

  out.open("inputs");

  TicTacToe t;


  bool x_turn=true;

  string start="_________";
  string current,old;

  current = start;
```

```
int i,j;

while ((!t.winning_board(current,'X')) && (!t.winning_board(current,'O')) && (!t.full(
  cout << "Current Board:" << endl;
  t.print_board(cout,current);
  if (x_turn) { cout << "X's turn -- input your move" << endl;
    old=current;
    //getline(cin,current);
    cin >> i >> j;
    out << i << " " << j << endl;
    i=i-1;
    j=j-1;
    t.mod_i_j(current,i,j,'X');

    while (!t.valid_move(old,current,'X')) {
      current = old;
      cout << "Invalid Move --- try again" << endl;
      cin >> i >> j;
      out << i << " " << j << endl;
      i=i-1;
      j=j-1;
      t.mod_i_j(current,i,j,'X');
      //getline(cin,current);
    }
    x_turn  = false;
  } else {
    cout << "O's turn -- input your move" << endl;
    old=current;
    cin >> i >> j;
    out << i << " " <<j << endl;
    i=i-1;
    j=j-1;
    t.mod_i_j(current,i,j,'O');
    while (!t.valid_move(old,current,'O')) {
      current = old;
      cout << "Invalid Move --- try again" << endl;
      cin >> i >> j;
    out << i << " " << j << endl;
      i = i-1;
```

```
        j=j-1;
        t.mod_i_j(current,i,j,'O');
        //getline(cin,current);
      }
      x_turn = true;
    }
  }
  t.print_board(cout,current);
  if (t.winning_board(current,'X')) cout << "X wins!!! " << endl;
  else if (t.winning_board(current,'O')) cout << "O wins!!! " << endl;
  else cout << "It's a tie!" << endl;

  out.close();
}
```

If another student simply changed the comments and turned in the same assignment, this is clearly an act of plagiarism. If someone turned a similar assignment where the comments were changed and the variable names were changed, as below, this is also considered plagiarism.

```
//*****************************************************
// Student #2's code
// This is essentially the same as student #1's code
// with some simple rearrangements and
// variable/class name changes
//
//*****************************************************
#include <iostream>
#include <cassert>
#include <fstream>
#include <string>

using namespace std;


class Utils {
public:
  bool valid(const string &board) {
    if (board.length()!=9) return false;
    for (int i=0;i<board.length();i++) {
      switch (board[i]) {
```

```
    case '_':
    case 'X':
    case 'O': break;
    default: return false;
    }
  }
  return true;
}

void print(ostream &out, const string &board) {
  out << board.substr(0,3) << endl;
  out << board.substr(3,3) << endl;
  out << board.substr(6,3) << endl;
}
bool winning(const string &board, const char c) {
  assert(valid(board));
  bool diag_win = true;
  for (int i=0;i<3;i++) {
    if (board[i*3+i]!=c) {
      diag_win= false;
    }
  }

  bool diag_win1 = true;
  for (int i=0;i<3;i++) {
    if (board[(2-i)*3+i]!=c) {
      diag_win1= false;
    }
  }

  bool across_win = false;
  for (int i=0;i<3;i++) {
    bool t = true;
    for (int j=0;j<3;j++) {
      if (board[i*3+j]!=c) t=false;
    }
    if (t) across_win =true;
  }

  bool down_win = false;
  for (int i=0;i<3;i++) {
    bool t = true;
    for (int j=0;j<3;j++) {
      if (board[j*3+i]!=c) t=false;
    }
    if (t) down_win =true;
  }
  return diag_win || diag_win1 || across_win || down_win;
}


bool full(const string &board) {

  bool fullv = true;
  for (int i=0;i<board.length();i++) {
    if (board[i]=='_') fullv = false;
  }
  return fullv;
}

void mod_i_j(string &new_board, int i, int j, const char c) {
  if ((i<0) || (i>2)) return;
  if ((j<0) || (j>2)) return;
  new_board[i*3+j]=c;
}

bool valid_move(const string &old_board, const string &new_board, const char c) {
  assert(valid(old_board));
  if (!valid(new_board)) return false;

  int count = 0;
  for (int i=0;i<new_board.length();i++) {
    if (old_board[i]!='_') {
      if (old_board[i]!=new_board[i]) return false;
    } else {
      if ((new_board[i]!='_')) {
        if ((new_board[i]==c)) {
        count++;
      } else {
          return false;
          }
        }
      }
    }
    if (count!=1) return false;

    return true;
  }
};

int main() {
  ofstream out;

  out.open("inputs");

  Utils my_board;


  bool x_turn=true;

  string start="_________";
  string current,old;

  current = start;

  int i,j;

  while ((!my_board.winning(current,'X')) && (!my_board.winning(current,'O')) && (!my_bo
    cout << "Current Board:" << endl;
    my_board.print(cout,current);
    if (x_turn) { cout << "X's turn -- input your move" << endl;
      old=current;
      //getline(cin,current);
      cin >> i >> j;
      out << i << " " << j << endl;
      i=i-1;
      j=j-1;
      my_board.mod_i_j(current,i,j,'X');

      while (!my_board.valid_move(old,current,'X')) {
        current = old;
        cout << "Invalid Move --- try again" << endl;
        cin >> i >> j;
        out << i << " " << j << endl;
        i=i-1;
        j=j-1;
        my_board.mod_i_j(current,i,j,'X');
        //getline(cin,current);
      }
      x_turn  = false;
    } else {
      cout << "O's turn -- input your move" << endl;
      old=current;
      cin >> i >> j;
      out << i << " " <<j << endl;
      i=i-1;
      j=j-1;
      my_board.mod_i_j(current,i,j,'O');
      while (!my_board.valid_move(old,current,'O')) {
        current = old;
        cout << "Invalid Move --- try again" << endl;
        cin >> i >> j;
      out << i << " " << j << endl;
        i = i-1;
        j=j-1;
        my_board.mod_i_j(current,i,j,'O');
        //getline(cin,current);
      }
      x_turn = true;
    }
  }
  my_board.print(cout,current);
  if (my_board.winning(current,'X')) cout << "X wins!!! " << endl;
  else if (my_board.winning(current,'O')) cout << "O wins!!! " << endl;
  else cout << "It's a tie!" << endl;

  out.close();
}
```

As this example illustrates, making straightforward changes to someone's code and turning that code in as your own is an act of academic misconduct. This is academic misconduct even if you typed in the code yourself. DON'T do it! Similarly, **if you work in a group on an individual effort assignment**, the result will be same, and you will be guilty of plagiarism.

4 Avoiding Plagiarism

Here are some general tips about how to avoid being accused of plagiarism or contributing to plagiarism.

1. Do not work in groups on individual effort assignments. Speaking to each other in *general terms* about an assignment is usually OK, but writing code on a board or sharing source code could lead to problems.

2. Do not share your solutions to assignment with others in the same class during the same or following quarters.

3. Do not copy code from the Internet.