Homework 7

Collaborators: Jay Hou, Thomas Weng, David Marcano

Will Childs-Klein

wdc22

April 21, 2015

Problem 1

We will show that the Generalized Domino Problem (DOM) is NP-Complete. In order to do this, we must show that both:

1. DOM $\in$ NP-Complete
2. DOM $\in$ NP-Hard

First, we will prove (1) directly by providing a description of a polynomial complexity certifier, then we will prove (2) by reducing a known NP-Complete problem, the 3-Color Graph Problem (3CG), to DOM. Specifically, we will provide a reduction as follows:

$$3\text{CG} \leq_P \text{DOM}$$

Lemma 7.1.1

$$\text{DOM} \in \text{NP}$$

Proof

In order to show that DOM $\in$ NP, let us consider a certifier for DOM, which we will call B. In order to check the correctness of a proposed certificate t to the DOM problem, all B has to do is to iterate over the set of dominoes mapped to edges (as enumerated fully by t), ensuring each of the following constraints

1. every edge $e \in E$ is covered by a domino
2. no vertex has different values of dominoes assigned to it

Lemma 7.1.2

$$3\text{CG} \leq_P \text{DOM}$$

Proof

Recall the 3-Color Graph Problem (3CG). Given a graph $G = (V, E)$, the goal is to assign each vertex $v \in V$ a color such that no two nodes $u, v \in V$ of the same color $c_i (1 \leq i \leq 3)$ are connected by an edge $(u, v) \in E$. We will now describe a polynomial complexity Karp reduction of $3CG$ to DOM.

Reduction 7.1: 3CG to DOM

```
1   input: graph G = (V, E), colors C = {c_1, c_2, c_3}, k = |C| = 3
2   init: D, E' = ∅
3   begin
4     for i ∈ [1, k]:
5       for j ∈ [1, k]:
6         if i ≠ j, D = D ∪ (i, j)
7       end for
8     end for
9
10    for each (u, v) ∈ E:
11      E' = E' ∪ {(u, v), (v, u)}
12    end for
13
14  if DOM returns "yes" given (G = (V, E'), D) as input,
15    return "yes"
16  else
17    return "no"
18  end
```

In this reduction, D is the set of allowable dominoes, and E' is a set of *directed* edges. First, we map each of the 3 colors in C to two "dominoes", one for each directionality, and add those two dominoes to D. Next, for each undirected edge in $(u, v) \in E$, we add two edges $\{(u, v), (v, u)\}$ to E'. We then make our call to DOM.

Claim 7.1.3

Reduction 7.1 is of polynomial complexity.

Proof This reduction is clearly polynomial, as lines $[4 - 8]$ have a constant upper bound of 6, and line 11 is iterated only $|E|$ times.

Claim 7.1.4

Reduction 7.1 is correct.

Proof We convert the inputted undirected graph to a directed graph by adding to E' directed edges, one in each direction, for every edge in original E. Next, perform a similar mapping from colors to dominoes, consructing the set of dominoes such that there are no dominoes with equal endpoints. This ensures the color-difference constraint of the 3CG problem.

Theorem 7.1.5

$$\text{DOM} \in \text{NP-Complete}$$

Proof

Problem 2

We will show that the Short Lattice Vector Problem (SLVP) is NP-Complete. In order to do this, we must show that both:

1. SLVP $\in$ NP-Complete
2. SLVP $\in$ NP-Hard

First, we will prove (1) directly by providing a description of a polynomial complexity certifier, then we will prove (2) by reducing a known NP-Complete problem, the 3-Dimentional Matching Problem (3DM), to SLVP. Specifically, we will provide a reduction as follows:

$$\text{3DM} \leq_P \text{SLVP}$$

Lemma 7.2.1

$$\text{SLVP} \in \text{NP}$$

Proof

In order to show that SLVP $\in$ NP, let us consider a certifier for SLVP, which we will call B. In order to check the correctness of a proposed certificate t to the SLVP problem, all B has to do is to iterate over $A = \{a_1, ..., a_n\}$ and ensure that

1. $\sum\limits_{j=1}^{n} a_j = k$
2. $||\sum\limits_{j=1}^{n} a_j v_j||^2 \leq T$

Lemma 7.2.2

$$\text{3DM} \leq_P \text{SLVP}$$

Proof

Reduction 7.2: 3DM to SLVP

```
input:
init:
begin

end
```

Claim 7.2.3

Reduction 7.2 is of polynomial complexity

Proof

Claim 7.2.4

Reduction 7.2 is correct

Proof

Theorem 7.2.5

SLVP $\in$ NP-Complete

Proof

Problem 3

Below in `Reduction 7.3`, we define a Karp reduction of the k-Subset Sum Problem (kSS) to the General Subset Sum Problem (gSS)

Reduction 7.3: kSS to gSS

```
input: X = {x_1, ..., x_n}, k, T
init: X' = ∅, let T' be an int, n = |X|
begin
  for i = 1...n:
    x_i' = x_i * n + 1
    X' = X' ∪ {x_i'}
  end for

  T' = T * n + k

if gSS returns "yes" given (X', T') as input,
  return "yes"
else
  return "no"
end
```

Analysis

In the problem set, we are only asked to give the direct reduction kSS $\leq_P$ gSS. I have done so above in `Reduction 7.3`. It is of polynomial complexity, specifically $O(n)$.

Problem 4

We will show that the Hard Stable Matching Problem (HSMP) is NP-Complete. In order to do this, we must show that both:

1. HSMP $\in$ NP-Complete
2. HSMP $\in$ NP-Hard

First, we will prove (1) directly by providing a description of a polynomial complexity certifier, then we will prove (2) by reducing a known NP-Complete problem, the 3-Satisfiability Problem (3SAT), to HSMP. Specifically, we will provide a reduction as follows:

$$3\text{SAT} \leq_P \text{HSMP}$$

Lemma 7.4.1

$$\text{HSMP} \in \text{NP}$$

Proof

In order to show that HSMP $\in$ NP, let us consider a certifier for HSMP, which we will call B. In order to check the correctness of a proposed certificate t to the HSMP problem, all B has to do is ensure that each of the following is satisfied:

1. that every person is matched to exactly one company
2. every company is matched to exactly one person
3. the matching is stable
4. none of the forbidden matching constraints are violated

(1),(2), and (3) are linear in $|E|$, and (4) is polynomial in $|F|$ where $|F|$ is the forbidden matching.

Lemma 7.4.2

$$3\text{SAT} \leq_P \text{HSMP}$$

Proof

We will now construct this reduction. The idea here is to assign each $x \in X$ and its negation $\bar{x}$ to a value of true or false. We will do so by

Reduction 7.4: 3SAT to HSMP

```
input:  X = {x_1, ..., x_n} ,  C = {C_1, ..., C_k}
init:  set  V = {x_1, x̄_1, ..., x_n, x̄_n} ,
       set  A = {T_1, F_1, ..., T_n, F_n}
       func  pref(a) : a ∈ {V ∪ A} → a's preference profile
       set  F = ∅
begin
  for  i = 1...n:
    pref(x_i) = T_i > F_i > ...
    pref(x̄_i) = F_i > T_i > ...
    pref(T_i) = x̄_i > x_i > ...
    pref(F_i) = x_i > x̄_i > ...
  end  for

  for  j = 1...k:
    F_j = ∅
    for  each  of  C_j's  3  terms  y_i ∈ {x_i, x̄_i ,
     where  i  is  the  term's  respective  index:
        if  y_i  is  x_i
          F_j = F_j ∪ {(x_i, F_i)}
        else  if  y_i  is  x_i's  negation
          F_j = F_j ∪ {(x̄_i, T_i)}
    end  for
    F = F ∪ F_j
  end  for

end
```

Claim 7.4.3

Reduction 7.4 is of polynomial complexity

Proof

Claim 7.4.4

Reduction 7.4 is correct

Proof

Theorem 7.4.5

HSMP $\in$ NP-Complete

Proof