36111-cwk2-S-exerciseB

Name: Bowen Cai stu ID:10406140

Name: Haorui Chen stu ID:10407315

1. To prove that 3D-MATCH is in NPTIME, we need to show that checking the validity of a “guess” of 3D-MATCH is in polynomial time.

To check the “guess”, we first check whether the size of the “guess” is the same with |U|, |X| or |Y|, then we will check whether each element in U, X, Y only appears once in the “guess”. A “guess” is valid if and only if it satisfies the two conditions. And the check is in polynomial time since we will only need to go through the finite “guess” once. Therefore, 3D-MATCH is in NPTIME.

1. For every 1≤i≤n, Ti = { (ui,j, ai,j, bi,j), 1≤j≤m } ∪ { (ûi,j, ai,j+1, bi,j), 1≤j≤m-1 }∪{ (ûi,m, ai,1, bi,m) }.
2. In SAT problem, given a set of clauses Γ, we must find a group of truth assignment to make Γ satisfiable, or to say, we must make every single clause γ in Γ satisfiable. Following this way of thinking, we are going to select appropriate triples construct Z and explain meanings of every group of triples to make the proof.

For every clause γj where 1≤j≤m, exact one triple in Sj is selected. This is to show our selected literal is going to be made truth to further make the clause γj be truth. The triple we choose here form a subset of Sj.

Now say we’ve selected ui,j, then triple (ui,j, cj, dj) it means pi occurs in γj, according to definition of Sj. Now we focus on all ûi,js, ai,js and bi,js where index i is remains same with ui,j that we just selected.

Since none of the elements ui,j or ûi,j lie in more than one of the triples of Z, we choose all triples (ûi,j, ai,j, bi,j), 1≤j≤m, making selection of this group of triples mean that assigning TRUE to letter pi. ( for another propositional letter ph, if we are choosing all (uh,j, ah,j, bh,j), then this means we are assigning FALSE to ph. ). All triples we choose here forms a subset of Ti.

After above two steps of triple choosing, what we have done is assigning TRUE to propositional letter pi, and this assignment will make clause γj to be TRUE.

For every other clauses, we choose tuples like what is described above. If for all the clauses we have chosen tuples (ui,j, ai,j, bi,j) or (ûi,j, ai,j, bi,j) for truth assignment and a tuple (ui,j,cj,dj) or (ûi,j,cj,dj) for making occurrence of corresponding literal in a clause γ to be TRUE ( and making the clause γ be TRUE ), then all the clauses are TRUE, meaning that given set of clauses Γ is satisfiable.

Finally we check again if constraints of Z is met.

1. Either { (ui,j, cj, dj) } or { (ûi,j, cj, dj) } is chosen, it is a subset of Sj; either { (ui,j, ai,j, bi,j), 1≤i≤n } or { (ûi,j, ai,j, bi,j), 1≤i≤n } is chosen, it is a subset of Ti, either being Z+ or being Z--, so Z⊆(T1∪…∪Tn∪S1∪…∪Sm).
2. For every clauses γj, 1≤j≤m, a corresponding (ui,j, cj, dj) or (ûi,j, cj, dj) is chosen, so each cj, dj lies in exactly one triple.
3. For every propositional letter pi, 1≤i≤n, a corresponding group of (ui,j, ai,j, bi,j)s or (ûi,j, ai,j, bi,j)s, 1≤j≤m, is chosen, so each ai,j, bi,j lies in exactly one triple.
4. If a (ui,j, cj, dj) is selected to be in Z, then for the same index I, only all (ui,j, ai,j, bi,j), 1≤j≤m, can be selected to avoid multiple occurrences of ui,j/ûi,j. so none of the elements ui,j or ûi,j lies in more than one of the triples.

Thus, if Z⊆(T1∪…∪Tn∪S1∪…∪Sm) is a set of triples such that, for all i (1≤i≤n) and all j (≤ j≤m) each of the elements ai,j, bi,j, cj, dj lies in exactly one triple in Z, and none of the elements ui,j or ûi,j lies in more than one of the triples of Z, then the clause set Γ is satisfiable.

1. First we show a group of truth assignment for a set of clauses Γ that can satisfy it corresponds to a set Z’ defined in the previous question.

For every 1≤i≤n, if pi = FALSE, then we select all triples (ui,j, ai,j, bi,j), 1≤j≤m; else select (ûi,j, ai,j+1, bi,j), 1≤j≤m-1 and (ûi,m, ai,1, bi,m). All triples we choose here forms a subset of Ti.

Since given group of assignments can make Γ satisfiable, then in each clause γj there must be at least one literal, either pi or~pi, is TRUE making the clause TRUE. We select exactly one of such kind of literals, select (ûi,j, cj, dj) if literal is in form ~pi, else select (ui,j, cj, dj). The triple we choose here form a subset of Sj.

We check if all triples we select above form a set Z’ that meets requirements described in Q3.

* 1. Either { (ui,j, cj, dj) } or { (ûi,j, cj, dj) } is chosen, it is a subset of Sj; either { (ui,j, ai,j, bi,j), 1≤i≤n } or { (ûi,j, ai,j, bi,j), 1≤i≤n } is chosen, it is a subset of Ti, either being Z+ or being Z--, so Z⊆(T1∪…∪Tn∪S1∪…∪Sm).
  2. For every clauses γj, 1≤j≤m, a corresponding (ui,j, cj, dj) or (ûi,j, cj, dj) is chosen, so each cj, dj lies in exactly one triple.
  3. For every propositional letter pi, 1≤i≤n, a corresponding group of (ui,j, ai,j, bi,j)s or (ûi,j, ai,j, bi,j)s, 1≤j≤m, is chosen, so each ai,j, bi,j lies in exactly one triple.
  4. If a (ui,j, cj, dj) is selected to be in Z’, then for the same index I, only all (ui,j, ai,j, bi,j), 1≤j≤m, can be selected to avoid multiple occurrences of ui,j/ûi,j. so none of the elements ui,j or ûi,j lies in more than one of the triples.

So a group of truth assignment for a set of clauses Γ that can satisfy it corresponds to a set Z’ defined in the previous question.

Finally we can give definition of R and Z.

Let K be the collection of the m\*(n-1) objects ui,j or ûi,j that are not contained in any tuple of Z’. Define R ⊆ K×G×H to be the set of triples such that every element in K, G or H lies in exactly one of the triples of R. Then, Z could be defined as Z’ ∪ R.

1. We have learned that SAT is NPTIME-hard and proved 3D-MATCH is NPTIME. To prove that 3D-MATCH is NPTIME-complete, we now prove that 3D-MATCH is NPTIME-hard by showing that SAT is many-one logspace reducible to 3D-MATCH problem.

Let disjoint sets W=U= U1∪…∪ Un, X=A∪C∪G=A1∪…∪ An∪C∪G and Y=B∪D∪H=B1∪…∪ Bn∪D∪H. As demonstrated in problem 3 and 4, if and only if Γ is satisfiable, there is a collection of triples Z such that every element ui,j, ûi,j, ai,j, bi,j, cj, dj, gk, hk lies in exactly one of the triple of Z. Clearly, Z is a matching for W, X and Y.

A deterministic TM that maps a group of truth assignment to a set Z of triples runs in LogSpace. Because in such a TM, we always know which tuple we will choose in every step. So the run is deterministic. We only need to record current i and j, and it takes logarithmic space. All recording of produced triples are written in output tape.

It is easy to see from the description above that we could compute Z in LogSpace since all operations are working locally and not acquiring external space.

Therefore, 3D-MATCH is NP-TIME-complete.