Xiangyu Pen CMPSC464 Collaborator: None

1) Assume there are 2 language $L_1 \in NP$, $L_2 \in NP$
there are polynomial-time nondeterministic-Turing Machine M_1 decides L_1, M_2 decides L_2
thus to construct a poly-time NTM M to decide $L_1 \circ L_2$.
M = on input string w with length n
1) For each i between 1 and n
2) partition w into $w_1 = a_1 a_2 \dots a_i$, $w_2 = a_{i+1} \dots a_n$
3) run M_1 on w_1, run M_2 on w_2
4) if both accept, accept.
5) if at for loop end and not accept, reject
The for loop stage run in poly-time non-deterministically and are repeated at most $O(n)$ times
Thus M is poly-time NTM of $L_1 \circ L_2$
Therefore, NP is closed under concatenation

2.

1) Let UNARY_SSUM be the subset sum problem in which all numbers are in unary

Considering the complexity of size of the input and number of step required

if input in unary, size of the input equals to the value. that is $N \Rightarrow N$ size of same input in unary

if input in binary, $\log x$ squares to represent value x. that is $N \Rightarrow 2^N$

size of input in current base

from that. we can say the the size to represent in unary increases exponentially.

thus the reduction proof fails since the reduction 3SAT to UNARY_SSUM requires more than poly-time $\Rightarrow$ we can't say $3SAT \leq_p$ UNARY_SSUM

Since the input size increases exponentially, the number of step to represent corresponding input for UNARY_SSUM is exponential

Therefore the NP-Completeness proof for SUBSET-SUM fails to show UNARY_SSUM is NP-Complete.

2) To show a language L is in P

we need to show there is a poly-time algorithm decides L.

Simulate a nondeterministic TM in exponential time using deterministic TM

$\Rightarrow$ if SUBSET-SUM $\in$ NP, then UNARY_SSUM $\in$ P

N = on input x with size N

1) convert input x to binary X'. thus size will be $\log N$

2) Simulate the dynamic programming algorithm of SUBSET-SUM on the converted input X'

Since we can simulate the nondeterministic poly-time TM, which is polynomial in $\log N$

Therefore, UNARY_SSUM $\in$ P

reference: cs.virginia.edu/~evans/cs3102-s10/ps/ps6/ps6-comment.pdf

3.

1) in a 3-cnf formula, an $\neq$-assignment means each clause has at least 1 literal assign to 1, at least 1 literal assign to 0.

thus, since $\neq$-assignment satisfies ϕ without assigning 3 true literals in any clause, the negation of $\neq$-assignment will also preserve the property above.

So, we can say negation of any $\neq$-assignment to ϕ is also an $\neq$-assignment.

2) to show if ϕ is mapped into ϕ', then ϕ is satisfiable IFF ϕ' has an $\neq$-assignment

① if ϕ is satisfiable, we can obtain an $\neq$-assignment for ϕ' by extending the assignment to ϕ

if y_1 and y_2 in C_i is assigned to 0, assign z_i to 1, else assign z_i to 0.

assign b to 0

② if ϕ' has an $\neq$-assignment

by part ①, we may assume the $\neq$-assignment assign b to 0. otherwise negate the assignment

(This $\neq$-assignment can't assign y_1, y_2, y_3 to 0, since it would make one of 2 clause $(y_1 \vee y_2 \vee z_i)$ and $(\bar{z}_i \vee y_3 \vee b)$ to have all 0_s.

thus, restricting this assignment to variables of ϕ would sastify the assignment.

Therefore, we can obtain a poly-time reduction from 3SAT to $\neq$SAT.

3) from part 2)

3SAT $\leq_p$ $\neq$SAT

Since 3SAT $\in$ NP.

Therefore, $\neq$SAT $\in$ NP

reference: web.cse.ohio-state.edu/~rademacher.10/Sp16_6321/ps9sol.pdf

4.

① Union

assume there are 2 languages $L_1 \in PSPACE$ $L_2 \in PSPACE$

So there are NTM M_1 decides L_1 in $O(n^x)$ M_2 decides L_2 in $O(n^y)$

construct a NTM M to decides $L_1 \cup L_2$

M = on input w with length n

1) Run M_1 on w.

2) Run M_2 on w

3) if both accept, accept, else, reject

M is a NTM decides $L_1 \cup L_2$ with poly-time $O(n^{max(x,y)})$

Therefore, PSPACE is closed under Union

② Intersection

assume there are 2 languages $L_1 \in PSPACE$ $L_2 \in PSPACE$

So there are NTM M_1 decides L_1 in $O(n^x)$ M_2 decides L_2 in $O(n^y)$

construct a NTM M to decides $L_1 \cap L_2$

M = on input w with length n

1) Run M_1 on w, if reject, reject

2) Run M_2 on w, if reject, reject

3) else, accept

M is a NTM decides $L_1 \cap L_2$ with poly-time $O(n^{max(x,y)})$

Therefore, PSPACE is closed under intersection

③ Star

Assume there is a language $L \in PSPACE$

so there is a NTM M decides L in poly-time $O(n^k)$

Construct a NTM M' to decide L^*

M' = on input w with length n

1) if $w = \varepsilon$, accept

2) nondeterministically select a number m where $1 \leq m \leq n$

3) nondeterministically split w into m pieces

4) for all pieces of partition, w_i, where $1 \leq i \leq m$. run M on w_i, if reject, reject

5) else, accept

with this operation, 1) & 2) & 3) are all in poly-time,

4) takes at most $O(n^{k+1})$ where the for loop loops at most n times, and each loop takes at most $O(n^k)$

thus M' is a poly-time NTM decides L^*

Therefore, PSPACE is closed under star.

5.

Since $NP \subset PSPACE$, thus anything outside PSPACE is outside NP

Since any SAT problem is reducible to TQBF, and TQBF problem can be reduced to any PSPACE-hard problem by def of PSPACE-hard, and TQBF is PSPACE-complete by Theorem

thus any PSPACE-hard problem is NP-hard.

reference: aduni.org/courses/theory/courseware/psets/Problem_set_05_Solutions.html