hw12

Graded

Student

Giancarlos Marte

Total Points

22.5 / 40 pts

Question 1

A Way to prove P = NP?

2.5 / 5 pts

- ✔ **− 1.5 pts** Did not reduce all languages A in NP to crossword puzzle problem
- ✔ **− 1 pt** Using example for reduction instead of all languages A

Question 2

Subset-sum

9 / 12 pts

- ✔ **− 3 pts** Not descriptive enough
- All the right steps, but you skipped over creation of the function, and showing that the runtime is polynomial, you only stated it.

Question 3

Knapsack

3 / 12 pts

- ✔ **− 1 pt** verifier input incorrect, should be $\langle\langle I, w, v\rangle, c\rangle$ where c is cert that is a set of pairs and $c \subseteq I$
- ✔ **− 1 pt** verifier incorrect behavior: should check cert, not try to solve the general problem
- ✔ **− 1 pt** verifier runtime computation missing or incorrect
- ✔ **− 1 pt** computable fn: behavior incorrect or undefined variables
- ✔ **− 1 pt** computable fn: runtime missing, unclear, or incorrect
- ✔ **− 1 pt** iff => stated incorrectly or missing, should be something like $\langle S, t\rangle \in$ SUBSET-SUM then $f(\langle S, t\rangle) \in$ Knapsack
- ✔ **− 1 pt** iff => proof missing, unclear, or incorrect
- ✔ **− 1 pt** iff <= stated incorrectly or missing, should be something like if $\langle S, t\rangle \notin$ SUBSET-SUM then $f(\langle S, t\rangle) \notin$ Knapsack
- ✔ **− 1 pt** iff <= proof missing, unclear, or incorrect

Question 4

NP closed operations **7** / 10 pts

4.1 **op3** **4** / 5 pts

- ✔ **– 1 pt** verifier input incorrect, should be something like $\langle w, \langle c_1, c_2 \rangle \rangle$ where the cert is a pair of certs and c1 is for B's verifier and c2 is for C's verifier

4.2 **op2** **3** / 5 pts

- ✔ **– 1 pt** non-deterministic TM won't work, bc the branches cannot return; must use verifiers
- ✔ **– 1 pt** part2 explanation unclear, should be something like "when a verifier rejects (like L's verifier), it's not enough to say that the string is not in the language"

Question 5

readme **1** / 1 pt

- ✔ **– 0 pts** Correct

Question assigned to the following page: 1

1) A way to prove p = np?

step	statement	justification
1	prove that if the crossword puzzle algorithm problem is NP-complete, then P=NP	statement to prove
2	A language B is NP-complete if it satisfies two conditions: 1. B is in NP, and 2. every A in NP is polynomial time reducible to B	definition of NP-completeness
3	NP is the class of languages that have polynomial time verifiers or A language is in NP iff it is decided by some nondeterministic polynomial time turing machine	definition and theorem of NP
4	P is the class of languages that are decidable in polynomial time on a deterministic single-tape turing machine	definition of P
5	Let L = language of the crossword puzzle algorithm	renaming
6	Let V be a time verifier for L: V = "On input <S, c>, where S is a puzzle and c is a set of strings: 1. Test whether S contains all words in c row by row 2. if the test passes, then accept; otherwise, reject."	creating a verifier
7	Since V is a verifier for L, that means L is in NP	(2)
8	If B is NP-complete and B is in P, then P = NP	theorem
9	The crossword puzzle algorithm problem is in P	proven in hw11
10	$P \in NP$	P is a subset of NP
11	Since L is in P and P is in NP, that means every A in NP is polynomial time reducible to L	(10)
12	Therefore, the crossword puzzle algorithm is NP-complete and P=NP	(11) and (8)

⬛

Question assigned to the following page: 2

2) Subset-sum problem

step	statement	justification
1	prove that the subset-sum problem is NP-complete	statement to prove
2	if B is NP-complete and $B \leq_p C$ for C in NP, then C is NP-complete 3 steps to prove a language C is NP-complete: 1. show C is in NP 2. choose B, the NP-complete problem to reduce from 3. show a poly time mapping reduction from B to C	theorem
3	3 steps to prove subset-sum problem is NP-complete: 1. subset-sum is in NP based on theorem from book and slides 2. 3SAT 3. $3SAT \leq_p$subset-sum	theorem applied to subset-sum
4	To show poly time mapping reducibility: 1. create computable fn, 2. show that it run in poly time, 3. then show forward direction 4. and show reverse/contrapositive direction.	how to show polynomial time mapping reducibility
5	4 steps to show $3SAT \leq_p$subset-sum 1. function converts 3SAT to subset-sum a. <3 cnf-formula> → <S,t> 2. big O runtime of the function is polynomial 3. if 3SAT accepts 3 cnf-formula then TM that computes function also accepts 4. if 3SAT does not accept 3 cnf-formula then TM that computes function does not accept	(4) applied to $3SAT \leq_p$subset-sum
6	Therefore subset-sum problem is NP-complete	(3) and (5)

■

Question assigned to the following page: 3

3) Knapsack problem

step	statement	justification
1	prove that the knapsack problem is NP-complete	statement to prove
2	if B is NP-complete and $B \leq_p C$ for C in NP, then C is NP-complete 3 steps to prove a language C is NP-complete: 1. show C is in NP 2. choose B, the NP-complete problem to reduce from 3. show a poly time mapping reduction from B to C	theorem
3	3 steps to prove subset-sum problem is NP-complete: 1. knapsack problem is in NP: 2. 3SAT 3. $3SAT \leq_p knapsack$	theorem applied to knapsack
4	To show poly time mapping reducibility: 1. create computable fn, 2. show that it run in poly time, 3. then show forward direction 4. and show reverse/contrapositive direction.	how to show polynomial time mapping reducibility
5	4 steps to show $3SAT \leq_p knapsack$ 1. converts 3SAT to knapsack a. <3 cnf formula> → <I, wgt_{max}, val_{min}> 2. big O of function is polynomial 3. if 3SAT accepts 3 cnf-formula then TM that computes function also accepts 4. if 3SAT does not accept 3 cnf-formula then TM that computes function does not accept	(4) applied to $3SAT \leq_p knapsack$
6	Therefore knapsack problem is NP-complete	(3) and (5)

⬛

Question assigned to the following page: 4.1

4) Is NP closed under…?

4.1) if applying OP3 to a 3 languages in NP results in another language in NP then OP3 is closed for NP languages.

4.2)

step	statement	justification
1	Let A, B and C be languages in NP	given
2	The following is a nondeterministic polynomial time decider N for OP3: N = “On input <A, B, C, w>, where A, B and C are languages in NP and w is a string in B and in C: 1. Nondeterministically check if w is in B and in C 2. if yes, accept; otherwise, reject.”	creating a nondeterministic polynomial time decider
3	Since there is a nondeterministic polynomial time decider N for OP3 languages based on 3 given NP languages, that means the resulting language is in NP	(2)
4	if applying OP3 to a 3 languages in NP results in another language in NP then OP3 is closed for NP languages.	(3)

⬛

Question assigned to the following page: 4.2

4.2.1) if applying OP2 to a language in NP results in another language in NP then OP2 is closed for NP languages.
4.2.2)

step	statement	justification
1	Let L be languages in NP	given
2	The following is a nondeterministic polynomial time decider N for OP2: N = "On input <L, w> where L is a language in NP and w is a string that is not in L: 1. Nondeterministically check if w is not in L a. This might not be true because it is very difficult to prove that something does not exist. In this case it would be trying to prove that w does not exist in L. 2. if yes, accept; otherwise, reject."	creating a nondeterministic polynomial time decider
3	Since there is a nondeterministic polynomial time decider for OP2 languages based on a given NP language, that means the resulting language is in NP	(2)
4	if applying OP2 to a language in NP results in another language in NP then OP2 is closed for NP languages.	(3)

⬛

Question assigned to the following page: 5

README
other students: none
books/websites used: class lecture slides and class textbook Sipser
time spent: 4 hours