CS 271: Automata and Computation Theory **Spring 2015**

Problem Set #4

Due: Thursday, April 30, 2015

Problem 1 If ϕ is a boolean formula in 3-CNF, an sfst-assignment (sfst for "some false; some true") to the variables in ϕ is one where each clause has some literals which are false and some which are true. That is, each clause has at least one true literal and one false literal. For example, if $\phi = (a \vee b \vee a) \wedge (\bar{a} \vee b \vee c)$, then $a = T$, $b = F$, and $c = T$ is an sfst-assignment for ϕ but $a = b = c = T$ is *not* an sfst-assignment for ϕ because the first clause has no false literals. We say that ϕ is sfst-satisfiable if ϕ has an sfst-assignment.

a. Show that the negation of any sfst-assignment for ϕ is also an sfst-assignment for ϕ. For example, using the example formula above, $a = F$, $b = T$, and $c = F$ is an sfst-assignment for it.

b. Define SFST-SAT $= \{\langle\phi\rangle \mid \phi$ has an sfst-assignment$\}$. Prove that 3-SAT $\leqslant_{\mathrm{P}}$ SFST-SAT. [*Hint: Map each clause* $C_i = (a \vee b \vee c)$ *to*

$$(a \vee b \vee x_i) \wedge (\overline{x_i} \vee c \vee y_i) \wedge (x_i \vee y_i \vee z)$$

where x_i *and* y_i *are new variables for each* C_i *and* z *is a single new variable that is used multiple times. You have to show that if clause* C_i *is satisfied by some assignment of variables in* ϕ*, then the three new clauses have some sfst-assignment (be careful that different clauses don't require* z *to be set in contradictory manners). You also have to show that if the three clauses are sfst-satisfiable, then* C_i *is. Use the fact proved in part* ***a*** *to fix* z *to a particular truth value.*]

c. Conclude that SFST-SAT is NP-complete.

Problem 2 If ϕ is a boolean formula in 3-CNF, a jot-assignment ("jot" for just one true) of variables in ϕ is a satisfying assignment such that in each clause, exactly one literal is true. Prove that

$$\text{JOT-SAT} = \{\langle\phi\rangle \mid \phi \text{ is a boolean formula in 3-CNF that has a jot-assignment}\}$$

is NP-complete. [*Hint: Reduce from 3-SAT. For each clause* $(a \vee b \vee c)$*, produce the three clauses*

$$(\overline{a} \vee w \vee x) \wedge (b \vee x \vee y) \wedge (\overline{c} \vee y \vee z)$$

where w*,* x*,* y*, and* z *are four new variables per clause. Show that the original clause is satisfied if and only if the three new clauses are jot-satisfied.*]

Problem 3 For an undirected graph $G = (V, E)$, a set $S \subseteq V$ is called an independent set if for all $u, v \in S$, $(u, v) \notin E$. Prove that

$$\textsc{IndSet} = \{\langle G, k\rangle \mid G \text{ has an independent set of size } k\}$$

is NP-complete. [*Hint: Reduce from* VERTEXCOVER.]

Problem 4 In class, we defined the language

$$\textsc{Partition} = \Big\{\langle S\rangle \mid S \subseteq \mathbb{Z}^+ \text{ is a multiset and there exists a sub-multiset } A \subseteq S \text{ such that } \textstyle\sum_{x\in A} x = \sum_{x\in S\setminus A} x\Big\}$$

and showed that PARTITION $\in$ NP. Prove that PARTITION is NP-complete by giving a reduction from SUBSETSUM. [*Hint: Given an instance $\langle S, t\rangle$ of* SUBSETSUM, *let $s = \sum_{x\in S} x$ and form an instance of* PARTITION *by adding the elements $s + t$ and $2s - t$. You need to prove two parts: (1) if S has a sub-multiset that sums to t, then the instance of partition can be split into two sets which have equal sums; and (2) if the instance of partition can be split into two sets P_1 and P_2 which have equal sums, then S has a sub-multiset that sums to t. For (2), it's helpful to prove that $2s - t$ and $s + t$ cannot both be in the same P_i.*]

Problem 5 The "Bag of Holding" problem is the following. Given a finite multiset U, a function $\mathit{size} : U \to \mathbb{Z}^+$, a function $\mathit{value} : U \to \mathbb{Z}^+$, a size constraint s and a value goal v, is there a subset $S \subseteq U$ such that

$$\sum_{x\in S} \mathit{size}(x) \leqslant s \quad \text{and} \quad \sum_{x\in S} \mathit{value}(x) \geqslant v?$$

Put another way, imagine you have a collection of items U and a bag that can hold a fixed amount of stuff s. You want to put enough items in the bag such that the value of all of the items is at least v.

- **a.** Formulate the Bag of Holding problem as a language BAGOFHOLDING $= \{\ldots\}$.
- **b.** Prove that BAGOFHOLDING $\in$ NP.
- **c.** Prove that PARTITION $\leqslant_{\mathrm{P}}$ BAGOFHOLDING and thus BAGOFHOLDING is NP-complete.