CECS 329, Learning Outcome Assessment 7, October 19th, Fall 2023, Dr. Ebert

Problems

LO3. Recall the mapping reduction from `SAT` to `3SAT` described in lecture.

(a) Given the `SAT` instance $F(x_1, x_2, x_3) = x_1 \wedge (\overline{x}_2 \vee x_3)$, draw its parse tree and provide the associated Boolean formula G that is satisfiability equivalent to F and serves as the beginning step of the reduction. Hint: formula G introduces y-variables.

(b) Rewrite formula G by making use of the logical identity

$$(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$$

(c) Rewrite the formula from part b by making use of the logical identity

$$(P \rightarrow Q) \Leftrightarrow (\overline{P} \vee Q).$$

(d) Rewrite the formula from part c by performing one or more applications of De Morgan's rule.

(e) Rewrite the formula from part d by performing one or more applications of the distributive rule in order to obtain an AND of OR's. Then convert the AND of OR's to an AND of ternary (i.e. three) OR's and use `3SAT` notation to complete the reduction.

LO4. Answer the following. Note: scoring 14 or more points counts for passing.

(a) Write either P, NP, or co-NP next to each of the following decision problems in terms of which one best characterizes the complexity of the problem. Note: there is only one best answer for each. Two points each.

i. An instance of the `Vertex Cover` decision problem is a pair (G, k), where $G = (V, E)$ is a simple graph, $k \geq 0$ is an integer, and the problem is to decide if G has a **vertex cover** of size k, i.e. a set $C \subseteq V$ for which every edge $e \in E$ is incident with at least one vertex in C.

ii. An instance of `Substring` is a pair (s_1, s_2) of binary strings and the problem is to decide if s_1 occurs as a substring of s_2. For example $(10101, 001010111)$ is a positive instance of `Substring` since 10101 is a substring of 001010111.

iii. An instance of `2SAT` is a set $\mathcal{C}$ of Boolean disjunctive clauses, each having two literals. The problem is to decide if there is a variable assignment α over the variables of $\mathcal{C}$ for which each clause has at least one of its literals assigned 1.

iv. An instance of `Bounded Independent Set` is a graph $G = (V, E)$ and an integer $k \geq 0$ and the problem is to decide if the maximum independent set in G is of a size that does not exceed k.

(b) Based on mapping reductions from the Mapping Reducibility and Computational Complexity lectures, the three polynomial-time mapping reductions $A \leq_m^p B$, $B \leq_m^p C$, and $C \leq_m^p$ `Independent Set` establish that `Independent Set` is an NP-complete problem. Provide the specific names of decision problems A, B, and C. (3 points each)

(c) Which of the following decision problems is *not* NP-complete? (9 points)

i. `UNSAT`
ii. `Set Partition`
iii. `Hamilton Cycle`
iv. `Vertex Cover`

LO5. Solve the following.

(a) Provide the instructions of a URM program that computes the function $f(x) = \lfloor x/3 \rfloor$.

(b) In one or more paragraphs describe how your program goes about computing $f(x)$.

LO6. Solve the following problems.

(a) Compute the Gödel number for program $P = J(2,3,1), S(5), Z(6), T(5,3)$. Write your answer as a sum of powers of two minus 1 (see part b for an example). Show all work.

(b) Provide the URM program P whose Gödel number equals

$$2^{30} + 2^{54} + 2^{104} + 2^{117} - 1.$$

Show all work.

LO7. Answer/solve the following.

(a) When simulating the computation $P_x(y)$, why must a universal URM program compute the encoding of each of the configurations that comprise the computation, rather than work directly with the configurations themselves?

(b) A universal program P_U is simulating a program that has 205 instructions and whose Gödel number is

$$x = 2^3 + 2^{45} + 2^{67} + 2^{78} + 2^{117} + 2^{141} + \cdots + 2^{c_{205}} - 1.$$

If the current configuration of the computation of P_x on some input has encoding

$$\sigma = 2^2 + 2^6 + 2^8 + 2^{14} - 1,$$

then provide the next configuration of the computation *and* its encoding.

Solutions

LO3. Recall the mapping reduction from `SAT` to `3SAT` described in lecture.

(a) Given the `SAT` instance $F(x_1, x_2, x_3) = x_1 \wedge (\overline{x}_2 \vee x_3)$, draw its parse tree and provide the associated Boolean formula G that is satisfiability equivalent to F and serves as the beginning step of the reduction. Hint: formula G introduces y-variables.

Solution. For all parts, see the solution to Example 6.3 of the Complexity lecture, while replacing $\overline{x}_1$ in the solution with x_1, x_2 with $\overline{x}_2$, and $\overline{x}_3$ with x_3.

(b) Rewrite formula G by making use of the logical identity

$$(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$$

(c) Rewrite the formula from part b by making use of the logical identity

$$(P \rightarrow Q) \Leftrightarrow (\overline{P} \vee Q).$$

(d) Rewrite the formula from part c by performing one or more applications of De Morgan's rule.

(e) Rewrite the formula from part d by performing one or more applications of the distributive rule in order to obtain an AND of OR's. Then convert the AND of OR's to an AND of ternary (i.e. three) OR's and use `3SAT` notation to complete the reduction.

LO4. Answer the following. Note: scoring 14 or more points counts for passing.

(a) Write either P, NP, or co-NP next to each of the following decision problems in terms of which one best characterizes the complexity of the problem. Note: there is only one best answer for each. Two points each.

i. An instance of the `Vertex Cover` decision problem is a pair (G, k), where $G = (V, E)$ is a simple graph, $k \geq 0$ is an integer, and the problem is to decide if G has a **vertex cover** of size k, i.e. a set $C \subseteq V$ for which every edge $e \in E$ is incident with at least one vertex in C.

ii. An instance of `Substring` is a pair (s_1, s_2) of binary strings and the problem is to decide if s_1 occurs as a substring of s_2. For example $(10101, 001010111)$ is a positive instance of `Substring` since 10101 is a substring of 001010111.

iii. An instance of `2SAT` is a set $\mathcal{C}$ of Boolean disjunctive clauses, each having two literals. The problem is to decide if there is a variable assignment α over the variables of $\mathcal{C}$ for which each clause has at least one of its literals assigned 1.

iv. An instance of `Bounded Independent Set` is a graph $G = (V, E)$ and an integer $k \geq 0$ and the problem is to decide if the maximum independent set in G is of a size that does not exceed k.

Solution. i) `NP` ii) `P` iii) `P` iv) `co-NP`

(b) Based on mapping reductions from the Mapping Reducibility and Computational Complexity lectures, the three polynomial-time mapping reductions $A \leq_m^p B$, $B \leq_m^p C$, and $C \leq_m^p$ `Independent Set` establish that `Independent Set` is an NP-complete problem. Provide the specific names of decision problems A, B, and C. (3 points each)

Solution. $A =$ `SAT`, $B =$ `3SAT`, $C =$ `Clique`

(c) Which of the following decision problems is *not* `NP`-complete? (9 points)

i. `UNSAT`
ii. `Set Partition`
iii. `Hamilton Cycle`
iv. `Vertex Cover`

Solution. `UNSAT` is the complement of `SAT` and so is in `co-NP`. The general consensus is that it is not a member of `NP`.

LO5. Solve the following.

(a) Provide the instructions of a URM program that computes the function $f(x) = \lfloor x/3 \rfloor$.

Solution.

i. $J(1, 2, 9)$
ii. $S(2)$
iii. $J(1, 2, 9)$
iv. $S(2)$
v. $J(1, 2, 9)$
vi. $S(2)$
vii. $S(3)$
viii. $J(1, 1, 1)$
ix. $T(3, 1)$

(b) In one or more paragraphs describe how your program goes about computing $f(x)$.

Solution. R_2 counts up to x, and for every three 1's added to R_2, a 1 is added to R_3. Thus, once $R_1 = R_2$, R_3 will be storing the quotient $\lfloor x/3 \rfloor$. Finally, since $R_1 = R_2$ can happen after any application of $S(2)$ to R_2, we must check this condition after each $S(2)$ application.

LO6. Solve the following problems.

(a) Compute the Gödel number for program $P = J(2, 3, 1), S(5), Z(6), T(5, 3)$. Write your answer as a sum of powers of two minus 1 (see part b for an example). Show all work.

Solution. We have

$$\beta(J(2,3,1)) = 4\xi(1,2,0) + 3 = 4\pi(\pi(1,2),0) + 3 = 4\pi(9,0) + 3 = 4(511) + 3 = 2047,$$

$$\beta(S(5)) = 4(4) + 1 = 17,$$

$$\beta(Z(6)) = 4(5) = 20,$$

and

$$\beta(T(5,3)) = 4\pi(4,2) + 2 = 4(79) + 2 = 318.$$

$$\gamma(P) = \tau(2047, 17, 20, 318) = 2^{2047} + 2^{2065} + 2^{2086} + 2^{2405} - 1.$$

(b) Provide the URM program P whose Gödel number equals

$$2^{30} + 2^{54} + 2^{104} + 2^{117} - 1.$$

Show all work.

Solution. $P = T(4,1), J(2,1,2), S(13), Z(4)$.
For example, to get $J(2,1,2)$, we have $54 - 30 - 1 = 23$ and $23 \bmod 4 = 3$ which implies a Jump instruction. Also, $(23-3)/4 = 5$, and $\pi^{-1}(5) = (1,1)$ since $(5+1) = 2^1(2(1)+1)$. Finally, $\pi^{-1}(1) = (1,0)$ since $1+1 = 2^1(2(0)+1)$. Therefore, we have $J(1+1, 0+1, 1+1) = J(2,1,2)$.

LO7. Answer/solve the following.

(a) When simulating the computation $P_x(y)$, why must a universal URM program compute the encoding of each of the configurations that comprise the computation, rather than work directly with the configurations themselves?

Answer. The length of a computation configuration can be arbitrarily long, depending on how many registers is used by P_x. On the other hand, a universal program P_U is itself a URM and hence only has a finite number of registers. Therefore, instead of storing the components of a configuration, P_U encodes it as a single number which can be stored in one of its registers. Then P_U moves from one configuration encoding to the next with help of a finite number of URM programmed primitive recursive functions. All of this requires only a finite number of registers which is all that P_U possesses.

(b) A universal program P_U is simulating a program that has 205 instructions and whose Gödel number is

$$x = 2^3 + 2^{45} + 2^{67} + 2^{78} + 2^{117} + 2^{141} + \cdots + 2^{c_{205}} - 1.$$

If the current configuration of the computation of P_x on some input has encoding

$$\sigma = 2^2 + 2^6 + 2^8 + 2^{14} - 1,$$

then provide the next configuration of the computation *and* its encoding.

Solution.

$$c = \tau^{-1}(\sigma) = (2, 3, 1, 5).$$

Also, $\beta(I_5) = 38$ and $38 \bmod 4 \ = 2$ implies that I_5 is a transfer instruction $T(i, j)$, where $\pi(i-1, j-1) = 9 = (38-2)/4$. Finally, to get $\pi^{-1}(9)$ we see that

$$9 + 1 = 10 = 2^1(2(2) + 1),$$

and so $\pi^{-1}(9) = (1, 2)$ and $\beta^{-1}(38) = T(2, 3)$. Therefore,

$$c_{\text{next}} = (2, 3, 3, 6)$$

and

$$\tau(c_{\text{next}}) = 2^2 + 2^6 + 2^{10} + 2^{17} - 1.$$