HW 09 Solutions

Exposition by William Gasarch—U of MD

We do the Problems in a Funny Order

We do Problem 3 then 1 then 2.
This is in order of how interesting they are.

Problem 3a

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know if *at least 17 of them are in HALTON0*.

Problem 3a

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know if *at least 17 of them are in HALTON0*.

Come up with a Turing Machine M (by that I mean just write psuedocode that uses $M_1, \ldots, M_{100}$) so that

$M \in \mathrm{HALTON0}$ iff at least 17 of $M_1, \ldots, M_{100}$ are in $\mathrm{HALTON0}$.

Problem 3a

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know if *at least 17 of them are in HALTON0*.

Come up with a Turing Machine M (by that I mean just write psuedocode that uses $M_1, \ldots, M_{100}$) so that

$M \in \mathrm{HALTON0}$ iff at least 17 of $M_1, \ldots, M_{100}$ are in $\mathrm{HALTON0}$.

Here is M:

1. Input(x) (this will be ignored).
2. Run $M_1(0), \ldots, M_{100}(0)$ at the same time.
3. If you see that 17 of them halted then STOP

$M(0)$ halts IFF 17 of the $M_1(0), \ldots, M_{100}(0)$ halt.
Note Can replace 17 with anything and get similar result.

Problem 3b

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know HOW MANY halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,...,
$M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

Problem 3b

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know HOW MANY halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,…, $M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

What if you are allowed to ask HALTON0 less than 100 questions? IS there a number $q < 100$ such that you can determine WHICH of $M_1, \ldots, M_{100}$ are in HALTON0 with q questions to HALTON0? Prove your result.

Problem 3b

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know HOW MANY halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,…, $M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

What if you are allowed to ask HALTON0 less than 100 questions? IS there a number $q < 100$ such that you can determine WHICH of $M_1, \ldots, M_{100}$ are in HALTON0 with q questions to HALTON0? Prove your result.

Do binary search: First ask if there exists ≥ 50 of $M_1, \ldots, M_{100}$ that halt on 0. If yes then ask ≥ 75. If No then ask ≥ 25. Etc. This takes $\lceil \log_2(100) \rceil = 7$.

Problem 3c

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know WHICH ONES halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,. . .,
$M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

Problem 3c

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know WHICH ONES halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,. . .,
$M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

What if you are allowed to ask HALTON0 less than 100 questions? IS there a number $q < 100$ such that you can determine WHICH of $M_1, \ldots, M_{100}$ are in HALTON0 with q questions to HALTON0? Prove your result.

Problem 3c

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know WHICH ONES halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,…, $M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

What if you are allowed to ask HALTON0 less than 100 questions? IS there a number $q < 100$ such that you can determine WHICH of $M_1, \ldots, M_{100}$ are in HALTON0 with q questions to HALTON0? Prove your result.

First find HOW MANY halt on 0 by 3b. That is only 8 questions. You now KNOW x: EXACTLY x of $M_1, \ldots, M_{100}$ halt on 0. What to do now? Discuss?

Problem 3c

Bill gives you 100 Turing Machines $M_1, \ldots, M_{100}$. He wants to know WHICH ONES halt on 0.
If you could ASK HALTON0 100 questions then you could do this—just ask $M_1 \in \mathrm{HALTON0}$?, $M_2 \in \mathrm{HALTON0}$?,. . .,
$M_{100} \in \mathrm{HALTON0}$? and see see which ones return YES.

What if you are allowed to ask HALTON0 less than 100 questions?
IS there a number $q < 100$ such that you can determine WHICH of $M_1, \ldots, M_{100}$ are in HALTON0 with q questions to HALTON0?
Prove your result.

First find HOW MANY halt on 0 by 3b. That is only 8 questions.
You now KNOW x: EXACTLY x of $M_1, \ldots, M_{100}$ halt on 0.
What to do now? Discuss?

RUN $M_1(0), \ldots, M_{100}(0)$ simul UNTIL x of them halt.
Those x halt. Great
Key The rest DO NOT HALT on 0 since exactly x halt on 0.

A Thought Question

Known The problem of, given 100 TM's, determine WHICH ONES halt on 0, can be done with 7 queries to HALT0.

A Thought Question

Known The problem of, given 100 TM's, determine WHICH ONES halt on 0, can be done with 7 queries to HALT0.

Question Can it be done with 6 queries?

A Thought Question

Known The problem of, given 100 TM's, determine WHICH ONES halt on 0, can be done with 7 queries to HALT0.

Question Can it be done with 6 queries?

No.

A Thought Question

Known The problem of, given 100 TM's, determine WHICH ONES halt on 0, can be done with 7 queries to HALT0.

Question Can it be done with 6 queries?

No.

There is an entire field called Bounded Queries in Computability Theory

`https://www.amazon.com/Bounded-Queries-Recursion-Progress-Computer/dp/1461268486`

Problem 1: 3COL $\le$ CNF-SAT

SOLUTION

We are given a graph $G = (V, E)$. We assume $V = \{1, \ldots, n\}$.
For every vertex i we have 3 Boolean variables. We list them and what they mean

x_{iR}: T if $COL(i) = R$.
x_{iB}: T if $COL(i) = B$.
x_{iG}: T if $COL(i) = G$.
Formula has two parts

3COL $\leq$ CNF-SAT, PART ONE

Making sure that a satisfying assignment really is a (not necessarily proper) coloring

Every vertex has at least one color:

$$\bigwedge_{i=1}^{n} (x_{iR} \vee x_{iB} \vee x_{iG})$$

Every vertex has at most one color:

$$\bigwedge_{i=1}^{n} \neg(x_{iR} \wedge x_{iB}) \wedge \neg(x_{iR} \wedge x_{iG}) \wedge \neg(x_{iB} \wedge x_{iG})$$

3COL $\leq$ CNF-SAT, PART TWO

Make sure it's a proper coloring

$$\bigwedge_{(i,j)\in E} \neg(x_{iR} \wedge x_{jR}) \wedge \neg(x_{iB} \wedge x_{jB}) \wedge \neg(x_{iG} \wedge x_{jG})$$

Prob 2a: Coding TMs into Numbers

All TMs: $\Sigma = \{1, 2, 3\}$, Q is $\{1, \ldots, n\}$.
Describe a procedure to code Turing Machines into $\mathbb{N}$ such that the following holds:

- Two diff Turing Machines map to diff numbers. (Some numbers do not get mapped to.)
- The following should be computable:
  Input: $x, y \in \mathbb{N}$
  Output:
  If x does not code a TM than output THATSBSMAN.

HINT Do not over think this. Any way you code a TM into numbers should work.

Prob 2a: Coding TMs into Numbers, SOLUTION

THE CODING: Let $M = (Q, \{a, b, \#\}, \delta, s, h)$
The number will be the product of the following numbers

1. $2^{|Q|}$.
2. 3^{s} (Recall that s, the start state, is a number)
3. 5^{h} (Recall that h, the halt state, is a number)
4. For coding the transitions, next slide

Prob 2a: Coding TMs into Numbers, SOLUTION

There will be $n = (Q-1) \times \Sigma$ rules. Let $p_1, \ldots, p_n$ be the n primes after 5 (so $p_1 = 7$). (It's $Q-1$ since there are no transitions out of h.) Order the rules lexicographically by $Q \times \Sigma$, so

$\delta(1,1)$

$\delta(1,2)$

$\delta(1,3)$

$\delta(2,1)$

$\vdots$

$\delta(|Q|-1,3)$.

For $1 \le i \le n$ take rule i and form the following number.

1. $\delta(p,\sigma) = (q,\sigma')$ maps to $2^p \times 3^\sigma \times 5^q \times 7^{\sigma'}$. $(\sigma' \in \{1,2,3\})$.
2. $\delta(p,\sigma) = (q,L)$ maps to $2^p \times 3^\sigma \times 5^q \times 7^4$. $(4 \notin \{1,2,3\})$.
3. $\delta(p,\sigma) = (q,R)$ maps to $2^p \times 3^\sigma \times 5^q \times 7^5$. $5 \notin \{1,2,3\})$.

Problem 2c

Let M be the TM: $Q = \{1, 2, 3\}$, $\Sigma = \{1, 2, 3\}$, $s = 1$, $h = 3$,
$\delta(1, 1) = (1, L)$.
$\delta(1, 2) = (1, 2)$.
$\delta(1, 3) = (2, R)$.
$\delta(2, 1) = (1, 1)$.
$\delta(2, 2) = (3, 3)$.
$\delta(2, 3) = (3, L)$.
Code this TM into a number using your procedure.

Problem 2c Solution

The number will be the product of many numbers:
$Q = \{1, 2, 3\}$ (so 2^3),
$\Sigma = \{1, 2, 3\}$,
$s = 1$ (so 3^1),
$h = 3$ (so 5^3).
$\delta(1, 1) = (1, L)$. This is coded by $7^{2^1 3^1 5^1 7^4}$
$\delta(1, 2) = (1, 2)$. This is coded by $11^{2^1 3^2 5^1 7^2}$
$\delta(1, 3) = (2, R)$. This is coded by $13^{2^1 3^3 5^2 7^5}$
$\delta(2, 1) = (4, 1)$. This is coded by $17^{2^2 3^1 5^4 7^1}$
$\delta(2, 2) = (3, 3)$. This is coded by $23^{2^2 3^2 5^3 7^3}$
$\delta(2, 3) = (3, L)$. This is coded by $29^{2^2 3^3 5^3 7^4}$
Take the product of all of the above numbers.