COMPUTATION MIDTERM 2022 SPRING - 206631848

GUR TELEM (ID: 206631848)

1. Question

1.1. Section. I don't have enough time to formalize fully. Sorry.

As seen in the tutorial, the execution tree of M (non deterministic) is a binary tree. We can do a BFS order of execution to simulate taking every possible branch, and if we halted and accepted in one of the branches, we accept in the M' deterministic.

We know that BFS is computable, thus M' is as following

(1) For each $i \in \{1, 2, \dots\}$
   (a) Take the c_i the ith config that we get from the BFS search on the execution tree of M.
   (b) If in this config, we halt M, return the same as M.

1.2. Section. NEED TO PROVE WITH NON DETERMINISTIC? (I don't have time to re-do the proof, :-(

Note: NDTM is Non deterministic Turing machine.

We'll build M' TM (Turing Machine) s.t. (such that) $L(M') = L$.

M' for $\langle M \rangle$ will be defined as such:

(1) Iterate with $i \in \{1, 2, 3, 4, \dots\}$
   (a) Take $w_1, w_2, \dots, w_i \in \Sigma^*$ (w_i is the ith word in Σ^* in lexicographic order. It's possible because $|\Sigma^*| = \aleph_0$)
   (b) Run M on each of these for i steps (i steps each)
   (c) If M halts for one of them (say w_t), check if the output of M is w_t. If it is, accept and halt.
   (d) Continue to next iteration.

M' is a valid TM because it's made of basic compilation steps.

RTP: $L(M') = L$.

Assume $\langle M \rangle \in L$. Thus there's an $x \in \Sigma^*$ s.t. $f_M(x) = x$. We know that x has a lexicographic index in Σ^*. Assume the index is i.

Since $f_M(x)$ is defined, it means that M halts on x. Mark the number of steps k. Mark $t = \max\{i, k\}$.

Thus in the iteration t, M' will run M on $w_1, \dots, w_t$ (which includes $x = w_i$ because $i \leq t$). And also it will run M for $t \geq k$ steps. Meaning that M will halt on x returning x itself. And from the definition of M', it will accept and halt in step $1.c$ meaning that $\langle M \rangle \in L(M)$

Here I realize that I did it in a deterministic way and left it. Sorry, no time.

2. Question

2.1. RTP $RE \subseteq A$. Let $L \in RE$.

Thus there's an M TM s.t. $L(M) = L$.

Let's build M' as such:

(1) Run M for one step
(2) If it did an S step writing x and moving to q_i, do instead:
  (a) an R, writing x and moving to q_i and immediately after L, writing the same value we read, and staying in q_i
(3) If M halted, return as it did.
(4) Go to step 1

We see that M' is equivalent to M because M' accepts a word, it's only because M' accepted a word.

And since before and after step $2.a$ the configuration of M is the same, then it will not change it's operation (it also won't get stuck there).

Thus $L(M') = L(M)$.

Now, since M' accepts $L(M) = L$, we got that $Flex - L(M') = L$.

Meaning that $L \in A$.

Thus $RE \subseteq A$.

Q.E.D

2.2. RTP: $\overline{HP} \in A$. Let's define M' for input $(\langle M \rangle, x)$ as follows:

(1) Run M on x for 1 step
(2) If M halted, reject and halt.
(3) Do an S step
(4) Go back to step 1 (to continue running for 1 more step, not to start over).

Another way to describe it is (but it's equivalent, we will refer to the previous definition):

(1) For $i \in \{1, 2, \dots\}$
  (a) Run M on x for i steps (from the beginning)
  (b) If M halted, reject and halt.
  (c) Do an S step

Let $(\langle M \rangle, x) \in Flex - L(M')$. Since the only place where M' halts is step 2, then if M' halts it's because M halted and then M' rejects the input.

Meaning that for input $(\langle M \rangle, x)$ M' did an infinite amount of S steps.

BWoC $(\langle M \rangle, x) \in HP$.

Thus M' will halt after k iterations (k is the number of steps until M halts on x).

Since each iteration is a finite number of steps for M', we got a contradiction to M' doing an infinite amount of S steps.

Thus $(\langle M \rangle, x) \notin HP \implies (\langle M \rangle, x) \in \overline{HP}$.

Now assume $(\langle M \rangle, x) \in \overline{HP}$.

$\implies M$ never halts for x

$\implies M'$ (on $(\langle M \rangle, x)$) will do an infinite amount of iterations

$\implies M'$ will do an infinite amount of S steps

$\implies (\langle M \rangle, x) \in Flex - L(M')$

together we got that $Flex - L(M') = \overline{HP} \implies \overline{HP} \in A$.

Q.E.D

2.3. Section. Proof:

Let's define M' a TM which defines for input $\langle M \rangle$ as such:

(1) Iterate with $i \in \{1, 2, 3, 4, \dots\}$
  (a) Set variable $count = 0$
  (b) Take $w_1, w_2, \dots, w_i \in \Sigma^*$ (lexicographical order)
  (c) Run M on each of them for i steps (replacing each S step with R and then L which is equivalent).
  (d) For each w_t $(1 \leq t \leq i)$ that accepts (and halts), increase $count$ by 1.
  (e) If $count \geq 4$, halt and reject.
  (f) If $count = 3$ do an S step.

Let $\langle M\rangle \in Flex - L(M')$.

Thus M' doesn't halt on $\langle M\rangle$ (if M' halts, it only rejects). Meaning that M' did an infinite amount of S steps.

BWoC: $\langle M\rangle \notin L_{=3}$.

Case 1. $|L(M)| < 3$: Thus $count$ will only be incremented twice at most in each iteration (we never found 3 different w_t that M accepts). Since we reset $count$ before each, then $count$ never reaches 3. Meaning we will never do an S step (we removed the S steps from everywhere else). Thus $\langle M\rangle \notin Flex - L(M')$ (let alone we don't have an infinite amount of S steps in its execution).

Case 2. $|L(M)| > 3$: Assume w_i is the 4th word lexical index which is accepted by M. Assume k is the max between the steps needed to accept all of these 4 words. Thus after max $\{i,k\}$ iterations, M' will hat in step $1.e$ rejecting. Meaning that $\langle M\rangle \notin Flex - L(M')$.

Thus $\langle M\rangle \in L_{=3}$. Meaning $Flex - L(M') \subseteq L_{=3}$

Now assume $\langle M\rangle \in L_{=3}$.

Similarly to before Assume w_i is the 3th word lexical index which is accepted by M. Assume k is the max between the steps needed to accept all of these 3 words.

From the max $\{i,k\}$th iteration and forward, we will always find exactly 3 words that are accepted by M (the definition of $L_{=3}$).

Thus from this iteration forward we will never halt, and for each iteration we will do an S step.

Meaning M' will do an infinite amount of S steps $\Longrightarrow L_{=3} \subseteq Flex - L(M')$.

thus $Flex - L(M') = L_{=3} \Longrightarrow L_{=3} \in A$.

Q.E.D

3. Question

Mark $L_1 \in S_1$ and $L_2 \in S_2$.

Because $S_1, S_2 \subseteq RE$ we get that there are $M_1$and M_2 s.t. $L(M_1) = S_1$ and $L(M_2) = S_2$.

Using a reduction $HP \leq L_{S_1,S_2}$ with $f((\langle M\rangle, x)) = (\langle M_{x,1}\rangle, \langle M_{x,2}\rangle)$ with $M_{x,i}$ defined for input w as such:

(1) Run M on x.
(2) Run M_i on w and return as it.

This is a full computable function.

I don't have time to show it's a valid reduction but from the reduction's theorem we get that since $HP \notin R \Longrightarrow L_{S_1,S_2} \notin R$.

Kinda Q.E.D

4. Question

4.1. **Section.** RTP: $L_1 \notin RE$.

Using a reduction $\overline{HP} \leq L_1$ with $f((\langle M\rangle, x)) = \langle M_x\rangle$ with M_x defined for any input (ignores input) as such:

(1) Run M on x.
(2) Accept (and halt)

This is a full and computable function.

Additionally $(\langle M\rangle, x) \in \overline{HP} \iff M$ doesn't halt for $x \iff M_x$ is stuck at step 1 for all inputs $\iff L(M_x) = \emptyset = R(M_x) \Longrightarrow \langle M_x\rangle \in L_1$.

But if $f((\langle M\rangle, x)) = \langle M_x\rangle \notin L_1$ it means that there's $w \in L(M_x) \wedge w \notin R(M_x)$ (the opposite isn't possible because M_x will never reject).

$w \in L(M_x)$ meaning that M_x accepted $w \Longrightarrow M$ halted for $x \Longrightarrow (\langle M\rangle, x) \notin \overline{HP}$.

We got that $f((\langle M\rangle, x)) \in L_1 \iff (\langle M\rangle, x) \in \overline{HP}$.

This is a valid reduction and from the reduction's theorem we get that since $\overline{HP} \notin RE \Longrightarrow L_1 \notin RE$.

Q.E.D

4.2. **Section.** RTP: $L_2 = \Sigma^*$.

It's obvious that $L_2 \subseteq \Sigma^*$.

Assume $\langle M \rangle \in \Sigma^*$.

Let's build M' for input x as such:

(1) Run M on x
(2) Return the opposite of M (if M accepted, reject, and if M rejected, accept).

Case 1. M rejects $x \Longrightarrow M'$ accepts $x \Longrightarrow x \in L(M')$
Case 2. M accepts $x \Longrightarrow M'$ rejects $x \Longrightarrow x \notin L(M')$
Case 3. M doesn't stop on $x \Longrightarrow M'$ doesn't stop on $x \Longrightarrow x \notin L(M')$

$x \in L_2 \iff x \in R(M) \iff x$ is rejected by $M \iff x$ is accepted by $M' \iff x \in L(M')$.

Meaning that $\langle M \rangle \in L_2$.

Thus $L_2 = \Sigma^*$.

Meaning that $L_2 \in R$.

Q.E.D

4.3. **Section.** Let's show a reduction $HP \leq L_3$ defined with $f((\langle M \rangle, x)) = \langle M_x \rangle$ while M_x is defined for any input as such:

(1) Run M on x.
(2) Reject and halt immediately.

$(\langle M \rangle, x) \in HP \iff M$ halts for $x \iff M_x$ halts for ε and rejects $\iff \varepsilon \in R(M) \neq \emptyset \iff \langle M_x \rangle \in L_3$

This is because the only way M_x halts if after M finished running for x.

Thus this is a valid reduction (it's a full, computable function with $(\langle M \rangle, x) \in HP \iff f((\langle M \rangle, x)) \in L_3$.

And from the reduction's theorem, because $HP \notin R \Longrightarrow L_3 \notin R$.

Now we'll prove that $L_3 \in RE$.

Let's build M' TM which accepts L_3. M' for input $\langle M \rangle$ is defined as such:

(1) Iterate with $i \in \{1, 2, 3, 4, \dots\}$
    (a) Take $w_1, w_2, \dots, w_i \in \Sigma^*$ (lexicographical order)
    (b) Run M on $w_1, w_2, \dots, w_i$ for i steps. If (at least) one of them halted **and rejected**, accept and halt.

$\langle M \rangle \in L(M') \iff M'$ will halt in step $1.b \iff \exists i, t \in \mathbb{N}$ s.t. $t \leq i$ and M rejected w_t within i steps at most $\iff R(M) \neq \emptyset \iff \langle M \rangle \in L_3$

Thus $L(M') = L_3$ which means $L_3 \in RE \backslash R$.

Q.E.D