In the Name of God

Theory of Computer Science - HW2

Ali Fathi - 99204943 Instructor: Dr. Ebrahimi

---

1 Every Decidable $\leq_m 0^*1^*$

1.1 $A \leq_m 0^*1^* \Rightarrow A$ is decidable

Recalling reduction theorem:

Reduction Theorem. *For two L_1 and L_2 languages for which $L_1 \leq_m L_2$:*

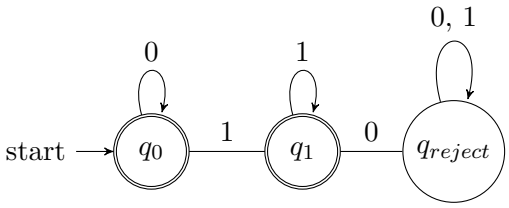
a) if L_1 is unrecognizable, then L_2 would be unrecognizable too.

b) if L_1 is undecidable, then L_2 would be undecidable too.

c) if L_2 is decidable, then L_1 would be decidable too.

d) if L_2 is recognizable, then L_1 would be recognizable too.

According to (c), it is enough to show 0^*1^* is decidable and it is so simple, because 0^*1^* is a regular language. Following DFA would decide 0^*1^*:



1.2 A is decidable $\Rightarrow A \leq_m 0^*1^*$

We need a Turing-computable function $f : \Sigma^* \to \Sigma^*$ for which $\forall w \in \Sigma^*$, $w \in A$ iff $f(w) \in 0^*1^*$.
A was a decidable language, so there exists some decider D for it which halts on every input on accept or reject state. We describe f as follow:

for *each* $w \in \Sigma^*$ **do**
  run D on w ;
  if D *accepted* w **then**
    write "01" on Tape (after clearing it) and halt ;
  else
    write "10" on Tape (after clearing it) and halt ;
  end
end

First of all, f would be Turing-computable because D halts on every word. Second, if $w \in \Sigma^*$, then $f(w) = 01 \in 0^*1^*$ and if $w \notin \Sigma^*$, then $f(w) = 10 \notin 0^*1^*$ and the reduction is complete. ■

---

2 Every Recognizable $\leq_m A_{TM}$

2.1 $A \leq_m A_{TM} \Rightarrow A$ is Turing-Recognizable

According to (d) in Reduction Theorem, A_{TM} is recognizable therefore A is a recognizable language too. To say more about it, assume f is a computable function which maps every member of A to a member of A_{TM}. We make the recognizer R for A by having U_{TM} (universal Turing machine) as follow: " R on input w:

- Compute $f(w)$
- Run U_{TM} on $f(w)$ and accept the input w, whenever U_{TM} accepted $f(w)$. "

As the definition of reducibility, for every $w \in A$, $f(w) \in A_{TM}$ so it would be accepted by U_{TM}. Also nothings out of A would be accepted by R so R is a recognizer for A then that's a recognizable language.

2.2 A is Turing-Recognizable $\Rightarrow A \leq_m A_{TM}$

Just like previous question, we need a Turing-computable function $f : \Sigma^* \to \Sigma^*$ for which $\forall w \in \Sigma^*$, $w \in A$ iff $f(w) \in A_{TM}$.
A is a recognizable language, so there exists some Turing-machine R such that $L(R) = A$. We describe f as follow:

$$f(w) \coloneqq \langle R, w \rangle$$

So if $w \in A$, then R accepts w in finite time therefore $\langle R, w\rangle \in A_{TM}$. Also if $w \notin A$, R does not halt on w (or rejects it) therefore $\langle R, w\rangle \notin A_{TM}$. As R and w are definite, encoding them by f is a trivial task and f would be clearly Turing-computable; then reduction is complete. ■

---

3 Undecidable Language $L \subseteq \{1\}^*$

We use the fact that the tape alphabet (Γ) could be different from the input alphabet (Σ). We take Γ to be $\{0, 1, \sqcup\}$. Consider an invertible, Turing-computable function f as below:

$$\begin{cases} f : \{1\}^* \to \{0,1\}^* \\ f(1^n) \coloneqq binary(n) \quad 0 \leq n \end{cases}$$

Which $binary(n)$ is the representation of integer n in binary form (e.g. $f(1111) = 100$ and $f(\epsilon = 1^0) = 0$). Every input $x = 1^n$, has a definite $f(x)$, and for every $y \in \{0,1\}^*$ starting with 1, there exists a unique $x = 1^n$ for which $f(x) = y$ that n is $decimal(y)$ ($y = 0$ is an exception which starts with 0 and is $f(1^0)$).
Besides, f is a Turing-computable function, because a Turing machine F can move along input $x = 1^n$ written on the tape, and at each step, adds to a binary counter (on another tape) until reaches to $\sqcup$. Also f^{-1} is Turing-computable on words starting with 1, and a machine G can decrease the binary input at each step and add a 1 to a second tape (G is map from $1.\{0,1\}^*$ to $\{1\}^*$).
The descriptions of Turing machines ($\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$) are definite on $\{0,1\}^*$. With this encoding, $A_{TM} = \{\langle M, w\rangle \,|\, M \in TMs\,,\, w \in L(M)\}$ is also a definite language on $\{0,1\}^*$. We add a "1" at the first of each member of A_{TM} and apply f^{-1} on this set to get a set of descriptions on $\{1\}^*$:

$$\mathscr{A}_{TM} \coloneqq \{f^{-1}(1\langle M, w\rangle) \,|\, \langle M, w\rangle \in A_{TM}\}$$

$\mathscr{A}_{TM} \subseteq \{1\}^*$, is an encoding language on $\Sigma = \{1\}$ and must be undecidable; Because if there exists any decider $\mathscr{D}$ for $\mathscr{A}_{TM}$, we can make a decider D for A_{TM} as follow:
" D on input x parsed as $\langle M, w\rangle$:

- Run G on $\langle M, w\rangle$ to compute $x' = f^{-1}(1\langle M, w\rangle)$.
- Run $\mathscr{D}$ on x' and go to accept or reject state according to it. "

Therefore as it must be impossible, $\mathscr{A}_{TM}$ should be an undecidable language on $\{1\}$. ■

---

4 Decidability Check

a) $L = \{\langle M, w, k\rangle \mid M \text{ uses at most first } k \text{ cells of it's tape when runs on } w\}$

L is decidable and a proof is like what is done for A_{LBA}. First of all, we should compare $|w|$ with k and reject the input if $|w| > k$. So we only need to decide about $\langle M, w, k\rangle$ with $|w| \leq k$. There is $|\Sigma|^k$ different values possible for tape content, and $|Q|$ states in M, and k positions for header if we ignore it's appearance out of $[1, k]$. So there would be just a finite different configurations for M using only k cells of tape in this case:

$$C \coloneqq |Q| \times |\Sigma|^k \times k \tag{1}$$

Consider U_{TM} as the *universal Turing machine*. We use U_{TM} to simulate M on w. If it accepted or rejected w using only these C configurations, we accept $\langle M, w, k\rangle$. If it visited one of these configurations again, we accept the input too! (because it loops on $\langle M, w, k\rangle$ which means nothings new would happen, and header wouldn't see any new position out of $[1, k]$).
If it went out of these C configurations (i.e. header enters $k + 1$), we simply reject the input and the description of our decider is done. ■

b) $L = \{\langle M, w\rangle \mid M \text{ uses finite memory when runs on } w\}$

L is an undecidable language and we show it by contradiction.
Consider D as a decider for L. We make the decider T for $HALT_{TM}$ as follows:
" T on input x parsed as $\langle M, w\rangle$:

- Run D on $\langle M, w\rangle$. If rejected, you reject the input x too.
- If accepted, simulate M on w and write all of your configurations during sumilation. Between every step of simulation, check whether the new configuration is repetitious or not. If it was repetitious, reject the input (because this means a loop). Otherwise, as the number of configurations in a finite tape is finite,

you indeed enter one of the accept or reject states in M and in this case, accept the input x. "

If M halts on w, obviously a finite number of tape cells would be used and D accepts $\langle M, w\rangle$ too. If M does not halt on w, it makes a finite or infinite loop. In the case of infinite loop, D rejects this input too but in the case of finite loop, D accepts this input but a further check about repetitious configurations (like what we did in part (a)) would lead us to know that it is a loop or not (as we know that the number of configurations for a bounded tape usage is finite, this checking process would certainly end). This T is a decider for $HALT_{TM}$ so the assumption would be wrong by contradiction. ■

c) -

d) $\forall w \in \Sigma^* : f(w) = |\{x \mid M_{1000} \text{ does not halt on } x\}|$ (-1 if infinite)

M_{1000} is a definite Turing machine, so the number of word on which M_{1000} does not halt (-1 if infinite), is a certain number in $\{-1, 0, 1, 2, \ldots\}$. If we consider functions $f_{-1}, f_0, f_1, f_2, \ldots$ which f_i prints i for all inputs, our answer would be one of these functions which are clearly computable. ■

e) $L = \{w \in \Sigma^* \mid M_{1000} \text{ halts on } \langle M_{1000}\rangle\}$

It is a decidable language for the same reason. M_{1000} is a definite Turing machine, so the answer of it't halting or looping on the definite word $\langle M_{1000}\rangle$ is true or false. Consider T_{accept} as a Turing machine which accepts all inputs, and T_{reject} as a Turing machine which rejects all inputs. Therefore one of these TMs are a decider for L so it is a decidable language. ■

f) $L = \{\langle M\rangle \mid M \text{ moves it's header to the left during running on } \epsilon\}$

It is a decidable language. We make decider D for L as follow:
" D on input x parsed as $\langle M\rangle$:

- Simulate U_{TM} on $\langle M, \epsilon\rangle$ (M on a blank tape) and write the states on another tape during simulation.
- Whenever you visited a state q which $\delta(q, \sqcap)$ says to go left, accept the input x.

– In each step, check whether you have visited a state again or not. If yes, reject the input x.

As the number of the states in M is finite, the machine would certainly visit one of the states again in $|Q|$ steps, so one of the cases would surly happen and the machine D is a decider. ■

---

5 Recognizability Check

Recall Rice theorem for recognizability:

Rice Theorem for Recognizability. *Consider P as a non-trivial semantic property and $L_P = \{\langle M\rangle \,|\, M \in TMs\,,\ M$ possesses the property $P\}$ and $\mathscr{L}_P = \{L(M) \,|\, \langle M\rangle \in L_P\}$. Then L_P is a recognizable language if and only if all following properties are held:*

1. $\forall L_1 \in \mathscr{L}_P \quad \forall$*recognizable* $L_2 \supseteq L_1 : \quad L_2 \in \mathscr{L}_P$
2. $\forall L_1 \in \mathscr{L}_P \quad \exists L_3 \subseteq L_1$ *with* $|L_3| < \infty : \quad L_3 \in \mathscr{L}_P$
3. $\exists$*enumerator E which prints all finite languages of* $\mathscr{L}_P$

We use Rice theorem to discuss about languages recognizability.

a) $P : |L| \geq 2$
$L_P = \{\langle M\rangle \,|\, M \in TMs\,,\ L(M) \geq 2\}$
$\mathscr{L}_P = \{L(M) \,|\, M \in TMs\,,\ L(M) \geq 2\}$

L_P is a recognizable language. All three conditions of Rice theorem are held:

1) $\forall M_2$ for which $L_1 \subseteq L_2 = L(M_2)$, $|L_2| \geq |L_1| \geq 2$ so $L_2 \in \mathscr{L}_P$.
2) $\forall L_1 \in \mathscr{L}_P$, L_1 contains at least two members. We take two of these members for L_3 and it is clearly a member of $\mathscr{L}_P$
3) If we assign numbers to all finite subsets of Σ^* (it is possible as we know they are countable), we would know that these subsets are easily enumerable. We make enumerator E to just check whether a subset is null or single membered, and print it if not.

We could describe a recognizer R for L_P as follow (consider E_0 as a enumerator on Σ^*):
" R on input x parsed as $\langle M \rangle$:

- In round k, run E_0 to print first of it's k members.
- Simulate M for k step on these k members.
- Add a counter (for example in a different tape) whenever one of those members was accepted by M, and go to q_{accept} whenever the counter reached 2 then halt. "

By this R, if a machine M has at least two words in it's language, R would accept it. ■

b) $P : |L(M)| < \infty$
$L_P = \{\langle M \rangle \,|\, M \in TMs\,,\, L(M) < \infty\}$
$\mathscr{L}_P = \{L(M) \,|\, M \in TMs\,,\, L(M) < \infty\}$

L_P is unrecognizable because the first condition of Rice theorem is easily violated. Conceder M_1 as a Turing machine which accepts only $w = 1$ and rejects all other inputs (i.e. $L_1 = L(M_1) = \{1\}$). Obviously $L_1 \in \mathscr{L}_P$. Assume M_2 is a machine which accepts inputs containing only 1 and rejects the other (i.e. $L_2 = L(M_2) = \{1, 11, 111, 1111, \ldots\}$). Clearly $L_1 \subseteq L_2$ and $L_2 \notin \mathscr{L}_P$ (as $|L_2| = \infty$) so the first condition of Rice theorem is violated.

c) $P : L(M)$ is CFL
$L_P = \{\langle M \rangle \,|\, M \in TMs\,,\, L(M)$ is $CFL\}$
$\mathscr{L}_P = \{L(M) \,|\, M \in TMs\,,\, L(M)$ is $CFL\}$

L_P is unrecognizable because again the first condition of Rice theorem is violated. Assume $L_1 = \{a^n b^n : n > 0\}$ and $L_2 = \{a^n b^n c^n : n > 0\}$ (on $\Sigma = \{a, b, c\}$). Both of them are Turing-recognizable and $L_1 \subseteq L_2$ and $L_1 \in \mathscr{L}_P$ but $L_2 \notin \mathscr{L}_P$, so the first condition of theorem is not held then L_P is an unrecognizable language.
