Wen Liang(Wen_Liang@student.uml.edu) 01724877

Note:The extra question I did was 4.9, 4.13, 4.16, 5.9, 5.22, 5.23

Extra Question:

4.9

Let $\sim$ be the binary relation "is the same size". In other words If A $\sim$ B, A and B are the same size. We show that $\sim$ is an equivalence relation. First, $\sim$ is reflexive because the identity function f(x) = x, $\forall$ x $\in$ A is a correspondence f: A $\rightarrow$ A. Second, $\sim$ is symmetric because any correspondence has an inverse, which itself is a correspondence. Third, $\sim$ is transitive because if A $\sim$ B via correspondence f, and B $\sim$ C via correspondence g, then A $\sim$ C via correspondence f $\circ$ g (the composition of f and g). Because $\sim$ is reflexive, symmetric, and transitive, $\sim$ is an equivalence relation.

4.13

We observe that L(R) $\subseteq$ L(S) if and only if $\overline{L(S)} \cap L(R) = \phi$. The following TM X decides A.
X = On input (R,S) where R and S are regular expressions:
1. Construct DFA E such that L(E) = $\overline{L(S)} \cap L(R)$.
2. Run TM T from Theorem 4.4 on input $< E >$, where T decides E_{DFA}.
3. If T accepts, accept. If T rejects, reject.

4.16

The folowing TM X decides A.
X = On input $< R >$ where R is a regular expression:
1. Construct DFA E that accepts $\sum^* 111 \sum^*$.
2. Construct DFA B such that L(B) = L(R)$\cap$L(E).
3. Run TM T from Theorem 4.4 on input $< B >$,where T decides E_{DFA}.
4. If T accepts, reject. If T rejects, accept.

5.9

Assume T is decidable and let decider R decide T. Reduce from A_{TM} by constructing a TM S as follows:
S: on input $< M, w >$
. 1. create a TM Q as follows:
. On input x:
. 1. if x does not have the form 01 or 10,reject

. 2. if x has the form 01, then accept.
. 3. else(x has the form 10), Run M on w and accept if M accepts w.
. 2. Run R on $< Q >$
. 3. Accept if R accepts, reject if R rejects.

Because S decides A_{TM}, which is known to be undecidable, we then know that T is not decidable.

5.22

($\Rightarrow$) If A $\leq A_{TM}$, then A is Turing-recognizable because A_{TM} is Turing recognizable.
($\Leftarrow$) If A is Turing-recognizable then there exists some TM R that recognizes A. That is, R would receive an input w and accept if w is in A (otherwise R does not accept). To show that A $\leq {}_m A_{TM}$, we design a TM that does the following: On input w, writes $< R, w >$ on the tape and halts. It is easy to check that $< R, w >$ is in A_{TM} if and only if w is in A. Thus, we get a mapping reduction of A to A_{TM}.

5.23

($\Rightarrow$) If A $\leq_m$ 0*1*, then A is decidable because 0*1* is a decidable language.
($\Leftarrow$) If A is decidable, then there exists some TM R that decides A. That is, R would receive an input w and accept if w is in A, reject if w is not in A. To show A $\leq$, reject if w is not in A. To show A $\leq {}_m 0^*1^*$, we design a TM Q that does the following : On input w, runs R on w. If R accepts, outputs 01; otherwise, outputs 10. It is easy to check that:

w $\in$ A $\Leftrightarrow$ output of Q $\in$ 0*1*.

Thus, we obtain a mapping reduction of A to 0*1*.

Required Question:

4.2

Define the language as C = $\{< M, R >|$ M is a DFA and R is a regular expression with L(M) = L(R) $\}$
Recall that the proof of Theorem 4.5 defines a Turing machine F that decides the language $EQ_{DFA} = \{ < A, B > |$ A and B are DFAs and L(A) = L(B) $\}$. Then the following Turing machine T decides C:

T = On input $< M, R >$, where M is a DFA and R is a regular expression:
1. Convert R into a DFA D_R using the algorithm in the textbook.
2. Run TM decider F from Theorem 4.5 on input $< M, D_R >$.
3. If F accepts, accept. If F rejects, reject.

4.3

Let $ALL_{DFA} = \{< A >|$ A is a DFA that recognizes $\sum^*\}$. The following TM L decides ALL_{DFA}.

L= On input $< A >$ where A is a DFA:
1. Construct DFA B that recognizes $\overline{L(A)}$.
2. Run TM T from Theorem 4.4 on input $< B >$, where T decides E_{DFA}.
3. If T accepts, accept. If T rejects, reject.

4.4

Let $A\epsilon_{CFG} = \{< G >| G$ is a CFG that generates $\epsilon\}$. The following TM V decides Aϵ_{CFG}.
V = " On input $< G >$ where G is a CFG:
1. Run TM S from Theorem 4.7 on input $< G, \epsilon >$, where S is a decider for A_{CFG}
2. If S accepts, accept. If S rejects, reject."

4.6

(a)

No, f is not one-to-one because f(1) = f(3).

(b)

No, f is not onto because there does not exist x $\in$ X such that f(x) = 10.

(c)

No, f is not correspondence because f is not one-to-one and onto.

(d)

Yes, g is one-to-one.

(e)

Yes, g is onto.

(f)

Yes, g is a correspondence.

4.7

Suppose B is countable and a correspondence f: N $\rightarrow$ B exists. We construct x in B that is not paired with anything in N. Let x = $x_1 x_2 ...$. Let $x_i = 0$ if $f(i)_i = 1$, and $x_i = 1$ if $f(i)_i = 0$ where $f(i)_i$ is the ith bit of f(i). Therefore, we ensure that x is not f(i) for any i because it differs from f(i) in the ith symbol, and a contradiction occurs.

4.8

We demonstrate a one-to-one f: T $\rightarrow$ N. Let $f(i, j, k) = 2^i 3^j 5^k$. Function f is one-to-one because if a $\neq$ b, f(a) $\neq$ f(b). Therefore, T is countable.

5.1

Suppose for a contradiction that EQ_{CFG} were decidable. We construct a decider M for $ALL_{CFG} = \{< G >|$ G is a CFG and L(G) =$\sum^*\}$ as follows:

M = "On input $< G >$:
1. Construct a CFG H such that L(H) = $\sum^*$
2. Run the decider for EQ_{CFG} on $< G, H >$.
3. If it accepts, accept. If it rejects, reject."

M decides ALL_{CFG} assuming a decider for EQ_{CFG} exists. Since we know ALL_{CFG} is undecidable, we have a contradiction.

5.2

Here is a Turing Machine M which recognizes the complement of EQ_{CFG}:
M = "On input $< G, H >$:
1. Lexicographically generate the strings x $\in \sum^*$.
2. For each such string x:
3. Test whether x $\in$ L(G) and whether x $\in$ L(H), using the algorithm for A_{CFG}.
4. If one of the tests accepts and the other rejects, accept; otherwise, continue."

5.3

$[\frac{ab}{abab}]$ $[\frac{ab}{abab}]$ $[\frac{aba}{b}]$ $[\frac{b}{a}]$ $[\frac{b}{a}]$ $[\frac{aa}{a}]$ $[\frac{aa}{a}]$

5.4

No, For example, define the languages $A = \{0^n 1^n \mid n \geq 0\}$ and B = { 1 }, both over the alphabet $\sum = \{0, 1\}$. Define the function f : $\sum^* \to \sum^*$ as

$$f(w) = \begin{cases} 1 & \text{if w} \in \text{A} \\ 0 & \text{if w} \notin \text{A} \end{cases} \tag{1}$$

Observe that A is a context-free language, so it is also Turing-decidable. Thus, f is a computable function. Also, w $\in$ A if and only if f(w) = 1, which is true if and only if f(w) $\in$ B. Hence, A $\leq {}_m B$. Language A is nonregular, but B is regular since it is finite.