Homework 1

Yutong Huang (yxh589)

Problem 1

Proof. (a) Assume L_1 and L_2 are decidable,
then $\exists m_1$ *that decides* L_1, $\exists m_2$ *that decides* L_2.
Because $L = L_1 \times L_2$, then $\forall (a, b) \in L, a \in L_1 \ \wedge \ b \in L_2$.
Consider the following algorithm M_a:

Algorithm 1: M_a

Input : (a, b)
Run m_1 with input a
if m_1 *accepts* **then**
  Run m_2 with input b
  if m_2 *accepts* **then**
    return accept
return reject

Case 1: $(a, b) \in L$:
then $a \in L_1$, $b \in L_2$.
Therefore both m_1 and m_2 accept $\implies \forall (a, b) \in L \ M_a$ accepts (a, b)
Case 2: $(a, b) \notin L$
Case 2.1 $a \in L_1 \wedge b \notin L_2$
$m1$ will accept but m_2 will reject
Therefore M_a will reject.
Case 2.2 $a \notin L_1 \wedge b \in L_2$
$m1$ will reject,
therefore M_a will reject.
Case 3.3 $a \notin L_1 \wedge b \notin L_2$
$m1$ will reject,
therefore M_a will reject.
Therefore $\forall (a, b) \notin L$, M_a will reject.
In conclusion, M_a decides L.

(b) Assume L_1 and L_2 are recognizable, then $\exists m_1$ *that regocnizes* L_1, $\exists m_2$ *that regocnizes* L_2.
Because $L = L_1 \times L_2$, then $\forall (a, b) \in L, a \in L_1 \ \wedge \ b \in L_2$.
Consider the following algorithm M_b:

Algorithm 2: M_b

Input : (a, b)
Run m_1 with input a
if m_1 *accepts* **then**
  Run m_2 with input b
  if m_2 *accepts* **then**
    return accept
return reject

Case 1: $(a, b) \in L$:
then $a \in L_1$, $b \in L_2$.
Therefore both m_1 and m_2 accept $\implies \forall (a, b) \in L \ M_b$ accepts (a, b)
Case 2: $(a, b) \notin L$

Case 2.1 $a \in L_1 \land b \notin L_2$
$m1$ will accept but m_2 will reject or loop infinitely
Therefore M_b will reject or loop infinitely.
Case 2.2 $a \notin L_1 \land b \in L_2$
$m1$ will reject or loop infinitely,
therefore M_b will reject or loop infinitely.
Case 3.3 $a \notin L_1 \land b \notin L_2$
$m1$ will reject or loop infinitely,
therefore M_b will reject or loop infinitely.
Therefore $\forall (a, b) \notin L$, M_b will reject or loop infinitely.
In conclusion, M_b recognizes L. □

Problem 2

Turing Machine:
$L = \{a^n b^{2n} | n \in \mathbb{N}\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, b, \phi, x, y\}$
$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_{accept}, q_{reject}\}$

δ :
$(q_1, a) \to (q_2, x, R)$
$(q_1, b) \to (q_{reject} \dots)$
$(q_1, x) \to (q_{reject} \dots)$
$(q_1, y) \to (q_{reject} \dots)$
$(q_1, \phi) \to (q_{accept} \dots)$
$(q_2, a) \to (q_2, a, R)$
$(q_2, b) \to (q_3, y, R)$
$(q_2, x) \to (q_2, x, R)$
$(q_2, y) \to (q_2, y, R)$
$(q_2, \phi) \to (q_{reject} \dots)$
$(q_3, a) \to (q_{reject} \dots)$
$(q_3, b) \to (q_4, y, L)$
$(q_3, x) \to (q_{reject} \dots)$
$(q_3, y) \to (q_3, y, R)$
$(q_3, \phi) \to (q_{reject} \dots)$
$(q_4, a) \to (q_4, a, L)$
$(q_4, b) \to (q_4, b, L)$
$(q_4, x) \to (q_5, x, R)$
$(q_4, y) \to (q_4, y, L)$
$(q_4, \phi) \to (q_{reject} \dots)$
$(q_5, a) \to (q_2, x, R)$
$(q_5, b) \to (q_{reject} \dots)$
$(q_5, x) \to (q_{reject} \dots)$
$(q_5, y) \to (q_6, y, R)$
$(q_5, \phi) \to (q_{reject} \dots)$
$(q_6, a) \to (q_{reject} \dots)$
$(q_6, b) \to (q_{reject} \dots)$
$(q_6, x) \to (q_{reject} \dots)$
$(q_6, y) \to (q_6, y, R \dots)$
$(q_6, \phi) \to (q_{accepts} \dots)$

Proof. Case 1: input $\in L$
Proof by induction:
Base case: n = 0 $\implies$ input is ϕ
The machine accepts immediately.
Base case: n = 1 $\implies$ input is abb
Steping through the machine:

Step	State	Tape	Head
0	q_1	abb	a
1	q_2	xbb	b
2	q_3	xyb	b
3	q_4	xyy	y
4	q_4	xyy	x
5	q_5	xyy	y
6	q_6	xyy	y
7	q_6	xyy	ϕ
8	q_{accept}	xyy	ϕ

Induction: assume the machine accepts input $a^k b^{2k}, k \in \mathbb{N}$, $k \geq 1$.
Consider input $a^k a b^{2k} bb$, the machine will run through the first k groups of a's and b's, at state q_3 the tape would be $x^k a y^{2k} bb$. Then the machine would enter q_4 to go back and find the last a, thus producing $x^{k+1} y^{2k} bb$. Next the machine would enter q_2 and find the second to last b, replace it with y, and enter q_3, and then replace the last b with y and enter q_4. This time, the machine will not find any a at q_5 and would enter q_6 to check if there's any leftover b's. When it finds that there's no b left, it will accept the input.

Therefore, the machine accepts input $\in L$.
Case 2: input $\notin L$
Case 2.1: too many a's
The machine will find a ϕ in either q_3 *or* q_4, and reject the input.
Case 2.2: too many b's
The machine will find a b in q_6 and reject.
Case 2.3: a after b
The machine will find a b in q_6 and reject.

Therefore, the machine will reject all input $\notin L$.
In conclusion, the machine is correct. □

Problem 3

Turing Machine:
Assume this machine can stay at a cell (S).
$L = \{a^n b^{2n} | n \in \mathbb{N}\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, b, \_, x, y, s\}$
$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{accept}, q_{reject}\}$

δ :
$(q_1, a) \to (q_9, s, R)$
$(q_1, b) \to (q_4, s, R)$
$(q_1, x) \to (q_{reject} \dots)$
$(q_1, y) \to (q_{reject} \dots)$
$(q_1, \_) \to (q_{accept} \dots)$
$(q_2, a) \to (q_2, a, R)$
$(q_2, b) \to (q_3, y, S)$
$(q_2, x) \to (q_2, x, R)$
$(q_2, y) \to (q_2, y, R)$
$(q_2, \_) \to (q_{reject} \dots)$
$(q_3, s) \to (q_4, \_, R)$
$(q_3, \_) \to (q_4, \_, R)$
$(q_3, *) \to (q_3, *, L)$
$(q_4, a) \to (q_5, x, R)$
$(q_4, b) \to (q_4, b, R)$
$(q_4, x) \to (q_4, x, R)$
$(q_4, y) \to (q_4, y, R)$
$(q_4, \_) \to (q_{reject} \dots)$
$(q_5, a) \to (q_6, x, S)$
$(q_5, b) \to (q_5, b, R)$
$(q_5, x) \to (q_5, x, R)$
$(q_5, y) \to (q_5, y, R)$
$(q_5, \_) \to (q_{reject} \dots)$
$(q_6, s) \to (q_7, s, R)$
$(q_6, \_) \to (q_7, \_, R)$
$(q_6, *) \to (q_6, *, L)$
$(q_7, b) \to (q_3, y, S)$
$(q_7, \_) \to (q_8, \_, L)$
$(q_7, *) \to (q_7, *, R)$
$(q_8, \_) \to (q_{accept})$
$(q_8, a) \to (q_{reject})$
$(q_8, s) \to (q_{accept})$
$(q_8, *) \to (q_8, *, L)$
$(q_9, \_) \to (q_6, a, L)$
$(q_9, *) \to (q_9, *, L)$

Proof. Case 1: input $\in L$
Proof by induction
Base case: Empty input:
The machine accepts immediately.
Base case: 2 'a' and 1 'b':
The machine will find 1 'b', then go back to the start (initial position of header), and start looking for 'a's. When it finds 2 'a's, the machine would go back to the start again, and check for additional 'b's. In this case, there is no more 'b', the machine then checks if there's any redundant 'a', which there is not in this case. Therefore, the machine accepts languages consists of 2 'a's and 1 'b'.

Induction Hypothesis: The machine accepts 2k 'a's and k 'b's, $k \in \mathbb{N} \wedge k \geq 2$.
Consider an input consists of 2k+2 'a's and k+1 'b's, the machine will find and mark the first k groups of 3 charaters (2 'a's and 1 'b'). Then there's only 2 'a's and 1 'b' left. At this point, the machine should be in state q_7 at the start of the input, looking for additional 'b's. It will find the 'b' and 2 of its corresponding 'a's. After that, with no more 'b's, the machine would be in state q_8 looking for redundant 'a's. It will not find any in this case, therefore it will accept the input.

Therefore, the machine accepts all inputs $\in L$.

Case 2: input $\notin L$
Case 2.1: too many a's
The machine will find redundant 'a's in q_8 and reject.
Case 2.2: too many b's
The machine will find a _ in either q_4 or q_5 and reject
Therefore, the machine will reject all inputs $\notin L$

In conclusion, the machine is correct. □