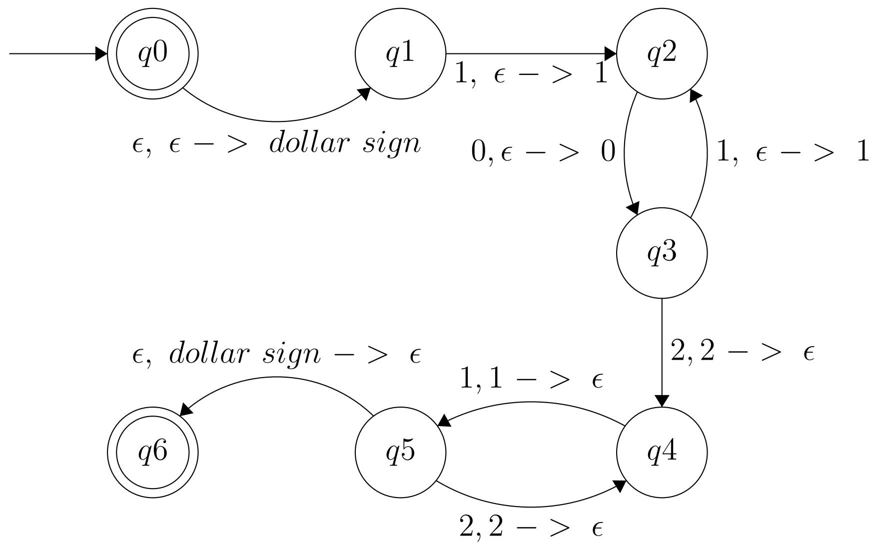
Question 1:

1.1 Language of two NFA

$L_{MA} = \{\omega | \omega \in \{(10)\}^*\}$
$L_{MB} = \{\omega | \omega \in \{(21)\}^*\}$

1.2 Combined PDA



1.3 Table

state transition	stack	string
q0 to q1	$\$\varepsilon$	empty
q1 to q2	$1\$\varepsilon$	1
q2 to q3	$01\$\varepsilon$	10
q3 to q2	$101\$\varepsilon$	101
q2 to q3	$0101\$\varepsilon$	1010
q3 to q4	$101\$\varepsilon$	10102
q4 to q5	$01\$\varepsilon$	101021
q5 to q4	$1\$\varepsilon$	1010212
q4 to q5	$\$\varepsilon$	10102121
q5 to q6	ε	accept

Question 2:

2.1

$S \to 1S00|\varepsilon$

2.2

- We already know: For L_1, there is a pumping length p such that any string $\omega \in L_1$ of length $\geq p$ can be written as $\omega = uvxyz$, where $vy \neq \varepsilon$, $|vxy| \leq p$, and for all $i \geq 0$, $uv^ixy^iz \in L_1$.

- so for part $uvxyz$, a suitable v will consist only 1 and y with only 00 so that in $uv^ixy^iz \in L_1$, the number of 0 is two times of 1 can still hold.

2.3

- Consider $v = 1$ and $y = 00$: the string is unpumpable because the number of 1's in the end of the string will not match with the 1's ahead.

- Consider $v = 1$ and $y = 01$: the string is unpumpable because it will have 1's betwwen 0's and the string order is broken.

- Consider $v = 10$ and $y = 0$: the string is unpumpable because it will have 0's between 1's and the string order is broken.

- Consider $v = 10$ and $y = 1$: the string is unpumpable because the number of 0's in the middle of the string will not be two times of 1's ahead.

- Consider $v = 10$ and $y = 01$: the string is unpumpable because it will have 1's betwwen 0's and the string order is broken.

- So for every possible combination, the string is unpumpable.

Question 3:

3.1

$01\#01$
$q_1 01\#01$
$xq_2 1\#01$
$x1q_2\#01$
$x1\#q_4 01$
$x1q_6\#x1$
$xq_7 1\#x1$
$q_7 x1\#x1$
$xq_1 1\#x1$
$xxq_3\#x1$
$xx\#q_5 x1$
$xx\#xq_5 1$
$xx\#q_6 xx$
$xxq_6\#xx$
$xq_7 x\#xx$
$xxq_1\#xx$
$xx\#q_8 xx$
$xx\#xq_8 x$
$xx\#xxq_8$
$xx\#xxq_{accept}$

3.2

$$
\begin{aligned}
&00\#0\\
&q_1 00\#0\\
&x q_2 0\#0\\
&x0 q_2 \#0\\
&x0\# q_4 0\\
&x0 q_6 \# x\\
&x q_7 0\# x\\
&q_7 x0\# x\\
&x q_1 0\# x\\
&xx q_2 \# x\\
&xx\# q_4 x\\
&xx\# x q_4\\
&xx\# x q_{reject}
\end{aligned}
$$

Question 4:

- Prove that a Turing decidable language is closed under complementation.
- Proof:
- If both L and complement of L are Turing decidable, we let Machine $M1$ be the Turing decider for L and $M2$ be the Turing decider for complement of L.
- we construct a Turing machine M' as follows:
- on input string w, M':
- runs both $M1$ and $M2$ in parallel for the input w;
- if $M1$ accepts, then M' accepts, elif $M2$ accepts then M' accepts
- otherwise M' rejects.
- Thus this Turing machine M' halts on every input w so it is a decider, therefore L and complement of L are Turing decidable.

Question 5:

- This Turing machine is no more powerful than a regular Turing machine because of the fact that every possible move it makes can be done by the regular Turing machine, and all these extra moves it owns can be achieved with atomic move that the regular Turing machine possesses.
- Three extra moves: S (stay), LL (two steps left) and RR (two steps right).
- A regular Turing machine can achieve S by doing a L then a R or a R then a L. Similarly, LL can be done by doing two Ls, RR can be done by doing two Rs.
- By splitting one move into two moves, we need two transition states. For example: $\delta(q_i, a) \to (q_{i+2}, c, RR)$ can be done by adding an intermediate state with $\delta(q_i, a) \to (q_{i+1}, b, R)$ and $\delta(q_{i+1}, b) \to (q_{i+2}, c, R)$.
- All other moves can be done in similar way, so this machine can be replaced by a normal Turing machine.