Homework 6 Solutions

1. Give pushdown automata that recognize the following languages. Give both a drawing and 6-tuple specification for each PDA.

(a) \( A = \{ w \in \{0, 1\}^* \mid w \text{ contains at least three } 1s \} \)

Answer:

We formally express the PDA as a 6-tuple \((Q, \Sigma, \Gamma, \delta, q_1, F)\), where

- \( Q = \{ q_1, q_2, q_3, q_4 \} \)
- \( \Sigma = \{0, 1\} \)
- \( \Gamma = \{0, 1\} \)
- transition function \( \delta : Q \times \Sigma \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma) \) is defined by

<table>
<thead>
<tr>
<th>Stack:</th>
<th>0</th>
<th>ε</th>
<th>1</th>
<th>ε</th>
<th>ε</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, ε → ε</td>
<td>0, ε → ε</td>
<td>0, ε → ε</td>
<td>0, ε → ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1, ε → ε</td>
<td>1, ε → ε</td>
<td>1, ε → ε</td>
<td>1, ε → ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q_1</td>
<td>{(q_1, ε)}</td>
<td>{(q_2, ε)}</td>
<td>{(q_3, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
</tr>
<tr>
<td>q_2</td>
<td>{(q_2, ε)}</td>
<td>{(q_3, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td></td>
</tr>
<tr>
<td>q_3</td>
<td>{(q_3, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td></td>
</tr>
<tr>
<td>q_4</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td>{(q_4, ε)}</td>
<td></td>
</tr>
</tbody>
</table>

Blank entries are \( \emptyset \).

- \( q_1 \) is the start state
- \( F = \{q_4\} \)

Note that \( A \) is a regular language, so the language has a DFA. We can easily convert the DFA into a PDA by using the same states and transitions and never push nor pop anything to/from the stack.
(b) \( B = \{ w \in \{0, 1\}^* \mid w = w^R \text{ and the length of } w \text{ is odd}\} \)

Answer:

\[
\begin{array}{cccc}
q_1 & \varepsilon, \varepsilon \rightarrow$ & q_2 & 0, \varepsilon \rightarrow \varepsilon \\
& & q_3 & 1, \varepsilon \rightarrow \varepsilon \\
& & & 0, 0 \rightarrow \varepsilon \\
& & & 1, 1 \rightarrow \varepsilon \\
& & & 0, \varepsilon \rightarrow 0 \\
& & & 1, \varepsilon \rightarrow 1 \\
\end{array}
\]

Since the length of any string \( w \in B \) is odd, \( w \) must have a symbol exactly in the middle position; i.e., \( |w| = 2n + 1 \) for some \( n \geq 0 \), and the \((n + 1)\)th symbol in \( w \) is the middle one. If a string \( w \) of length \( 2n + 1 \) satisfies \( w = w^R \), the first \( n \) symbols must match (in reverse order) the last \( n \) symbols, and the middle symbol doesn’t have to match anything. Thus, in the above PDA, the transition from \( q_2 \) to itself reads the first \( n \) symbols and pushes these on the stack. The transition from \( q_2 \) to \( q_3 \) nondeterministically identifies the middle symbol of \( w \), which doesn’t need to match any symbol, so the stack is unaltered. The transition from \( q_3 \) to itself then reads the last \( n \) symbols of \( w \), popping the stack at each step to make sure the symbols after the middle match (in reverse order) the symbols before the middle.

We formally express the PDA as a 6-tuple \((Q, \Sigma, \Gamma, \delta, q_1, F)\), where

- \( Q = \{q_1, q_2, q_3, q_4\} \)
- \( \Sigma = \{0, 1\} \)
- \( \Gamma = \{0, 1, \$\} \) (use \$ to mark bottom of stack)
- transition function \( \delta : Q \times \Sigma \times \varepsilon \times \Gamma \varepsilon \rightarrow P(Q \times \Gamma \varepsilon) \) is defined by

<table>
<thead>
<tr>
<th>Input:</th>
<th>0</th>
<th>1</th>
<th>\varepsilon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack:</td>
<td>0</td>
<td>1</td>
<td>$</td>
</tr>
<tr>
<td>(q_1)</td>
<td>{(q_2, 0), (q_3, \varepsilon)}</td>
<td>{(q_2, 1), (q_3, \varepsilon)}</td>
<td>{(q_2, $)}</td>
</tr>
<tr>
<td>(q_2)</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_4, \varepsilon)}</td>
</tr>
<tr>
<td>(q_3)</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_4, \varepsilon)}</td>
</tr>
<tr>
<td>(q_4)</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_3, \varepsilon)}</td>
<td>{(q_4, \varepsilon)}</td>
</tr>
</tbody>
</table>

Blank entries are \(\emptyset\).

- \( q_1 \) is the start state
- \( F = \{q_4\} \)

(c) \( C = \{ w \in \{0, 1\}^* \mid w = w^R \} \)

Answer:
The length of a string $w \in C$ can be either even or odd. If it's even, then there is no middle symbol in $w$, so the first half of $w$ is pushed on the stack, we move from $q_2$ to $q_3$ without reading, pushing, or popping anything, and then match the second half of $w$ to the first half in reverse order by popping the stack. If the length of $w$ is odd, then there is a middle symbol in $w$, and the description of the PDA in part (b) applies.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, \$\}$ (use $\$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma \epsilon \times \Gamma \epsilon \rightarrow P(Q \times \Gamma \epsilon)$ is defined by

<table>
<thead>
<tr>
<th>Input:</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack:</td>
<td>0</td>
<td>1</td>
<td>$$</td>
<td>$\epsilon$</td>
<td>0</td>
<td>1</td>
<td>$$</td>
<td>$\epsilon$</td>
<td>0</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$(q_2, 0), (q_3, \epsilon)$</td>
<td>$(q_3, \epsilon)$</td>
<td>$(q_4)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_2$</td>
<td>$(q_2, 1), (q_3, \epsilon)$</td>
<td>$(q_3, \epsilon)$</td>
<td>$(q_4)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_3$</td>
<td>$(q_3, \epsilon)$</td>
<td>$(q_3, \epsilon)$</td>
<td>$(q_4, \epsilon)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_4$</td>
<td>$(q_2, 0), (q_3, \epsilon)$</td>
<td>$(q_3, \epsilon)$</td>
<td>$(q_4, \epsilon)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \{q_1, q_4\}$

(d) $D = \{ a^i b^j c^k | i, j, k \geq 0, \text{ and } i = j \text{ or } j = k \}$

**Answer:**
The PDA has a nondeterministic branch at \( q_1 \). If the string is \( a^i b^j c^k \) with \( i = j \), then the PDA takes the branch from \( q_1 \) to \( q_2 \). If the string is \( a^i b^j c^k \) with \( j = k \), then the PDA takes the branch from \( q_1 \) to \( q_5 \).

We formally express the PDA as a 6-tuple \((Q, \Sigma, \Gamma, \delta, q_1, F)\), where

- \( Q = \{q_1, q_2, \ldots, q_8\} \)
- \( \Sigma = \{a, b, c\} \)
- \( \Gamma = \{a, b, \$\} \) (use \$ to mark bottom of stack)
- transition function \( \delta : Q \times \Sigma \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma) \) is defined by

Blank entries are \( \emptyset \).

- \( q_1 \) is the start state
- \( F = \{q_4, q_8\} \)

\( E = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i + j = k\} \)

**Answer:**

For every \( a \) and \( b \) read in the first part of the string, the PDA pushes an \( x \) onto the stack. Then it must read a \( c \) for each \( x \) popped off the stack.

We formally express the PDA as a 6-tuple \((Q, \Sigma, \Gamma, \delta, q_1, F)\), where

- \( Q = \{q_1, q_2, \ldots, q_5\} \)
- \( \Sigma = \{a, b, c\} \)
- \( \Gamma = \{x, \$\} \) (use \$ to mark bottom of stack)
- transition function \( \delta : Q \times \Sigma \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma) \) is defined by

| Stack | a | b | c | $ | ε | a | b | c | $ | ε | a | b | c | $ | ε | a | b | c | $ | ε | a | b | c | $ | ε |
| q1    |   |   |   | $ | ε |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| q2    |   |   |   |   |   | (q2, x) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| q3    |   |   |   |   |   | (q3, x) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| q4    |   |   |   |   |   |   | (q4, ε) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| q5    |   |   |   |   |   |   |   | (q5, ε) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \{ q_5 \}$

(f) $F = \{ a^{2n}b^{3n} \mid n \geq 0 \}$

Answer:

The PDA pushes a single $x$ onto the stack for every 2 $a$’s read at the beginning of the string. Then it pops a single $x$ for every 3 $b$’s read at the end of the string. We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{ q_1, q_2, \ldots, q_7 \}$
- $\Sigma = \{ a, b \}$
- $\Gamma = \{ x, $ \} (use $ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma \epsilon \times \Gamma \epsilon \rightarrow \mathcal{P}(Q \times \Gamma \epsilon)$ is defined by

<table>
<thead>
<tr>
<th>Stack:</th>
<th>$a$</th>
<th>$b$</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_1$</td>
<td>x</td>
<td>$$$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>{ (q3, $\epsilon$) }</td>
<td>{ (q4, $\epsilon$) }</td>
<td></td>
</tr>
<tr>
<td>$q_3$</td>
<td>{ (q2, x) }</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_4$</td>
<td>{ (q5, $\epsilon$) }</td>
<td>{ (q7, $\epsilon$) }</td>
<td></td>
</tr>
<tr>
<td>$q_5$</td>
<td></td>
<td>{ (q6, $\epsilon$) }</td>
<td></td>
</tr>
<tr>
<td>$q_6$</td>
<td></td>
<td>{ (q4, $\epsilon$) }</td>
<td></td>
</tr>
<tr>
<td>$q_7$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \{ q_7 \}$

(g) $L = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i + k = j \}$

Answer: A PDA $M$ for $L$ is as follows:
We formally express the PDA for $L$ as a 6-tuple $M = (Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_6\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{x, \$\}$ (use $\$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma \times \varepsilon \times \Gamma \rightarrow P(Q \times \Gamma \times \varepsilon)$ is defined by

<table>
<thead>
<tr>
<th>Input:</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack:</td>
<td>$x$</td>
<td>$$</td>
<td>$\varepsilon$</td>
<td>$x$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>$q_2$</td>
<td>{(q_2, x)}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>$q_3$</td>
<td>{}</td>
<td>{}</td>
<td>$(q_3, \varepsilon)$</td>
<td>{}</td>
</tr>
<tr>
<td>$q_4$</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>$(q_5, \varepsilon)$</td>
</tr>
<tr>
<td>$q_5$</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>$q_6$</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
</tbody>
</table>

Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \{q_6\}$

To explain the PDA $M$ for $L = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i + k = j\}$, note that $L = L_1 \circ L_2$ for

$L_1 = \{a^i b^j \mid i \geq 0\},$
$L_2 = \{b^k c^k \mid k \geq 0\},$

because concatenating any string $a^i b^j \in L_1$ with any string $b^k c^k \in L_2$ results in a string $a^i b^j b^k c^k = a^{i+j} c^k \in L$. Thus, for a string $a^i b^j c^k \in L$, the number $i$ of $a$’s at the beginning has to be no more than the number $j$ of $b$’s in the middle (because $i + k = j$ implies $i \leq j$), and the remaining number $j - i$ of $b$’s in the middle must match the number $k$ of $c$’s at the end. Hence, if we have PDAs $M_1$ and $M_2$ for $L_1$ and $L_2$, respectively, then we can then build a PDA for $L$ by connecting $M_1$ and $M_2$ so that $M_1$ processes the first part of the string $a^i b^j$, and $M_2$ processes the second part of the string $b^k c^k$. A PDA $M_1$ for $L_1$ is
(Another PDA for $L_1$ is given slide 2-38 of the notes.) Similarly, a PDA $M_2$ for $L_2$ is

![Diagram of a PDA](image)

But in connecting the two PDAs $M_1$ and $M_2$ to get a PDA $M$ for $L$, we need to make sure the stack is empty after $M_1$ finishes processing the first part of the string and before $M_2$ starts processing the second part of the string. This is accomplished in the PDA $M$ for $L$ by the transition from $q_3$ to $q_4$ with label “$\varepsilon, \$ \rightarrow \$”.

(h) $\emptyset$, with $\Sigma = \{0, 1\}$

**Answer:**

Because the PDA has no accept states, the PDA accepts no strings; i.e., the PDA recognizes the language $\emptyset$.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{x\}$
- transition function $\delta : Q \times \Sigma \times \Gamma \varepsilon \rightarrow \mathcal{P}(Q \times \Gamma \varepsilon)$ is defined by

<table>
<thead>
<tr>
<th>Input:</th>
<th>0</th>
<th>1</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack:</td>
<td>$x$</td>
<td>$\varepsilon$</td>
<td>$x$</td>
</tr>
<tr>
<td>$q_1$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \emptyset$

(i) The language $H$ of strings of properly balanced left and right brackets: every left bracket can be paired with a unique subsequent right bracket, and every right bracket can be paired with a unique preceding left bracket. Moreover, the string between any such pair has the same property. For example, $[[[[[[]]]]]] \in A$.

**Answer:**
We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{[, ]\}$
- $\Gamma = \{[, $\}$ (use $\$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma \epsilon \times \Gamma \epsilon \rightarrow P(Q \times \Gamma \epsilon)$ is defined by

| Input: | \[ | \] | \[ | \] | \$ | $ | \epsilon |
|---|---|---|---|---|---|
| Stack: | \[ | $ | \epsilon | | \[ | $ | \epsilon | | \[ | $ | $ | \epsilon |
| $q_1$ | | $\{(q_2, [)\}$ | | $\{(q_2, \epsilon)\}$ | | $\{(q_3, \epsilon)\}$ |
| $q_2$ | | $\{(q_2, [)\}$ | | $\{(q_2, \epsilon)\}$ | | $\{(q_3, \epsilon)\}$ |
| $q_3$ | | $\{(q_2, [)\}$ | | $\{(q_2, \epsilon)\}$ | | $\{(q_3, \epsilon)\}$ |

Blank entries are $\emptyset$.

- $q_1$ is the start state
- $F = \{q_3\}$

2. (a) Use the languages

$$A = \{a^m b^n c^n \mid m, n \geq 0\}$$
$$B = \{a^n b^n c^m \mid m, n \geq 0\}$$

together with Example 2.36 of the textbook to show that the class of context-free languages is not closed under intersection.

**Answer:** The language $A$ is context free since it has CFG $G_1$ with rules

$$S \rightarrow XY$$
$$X \rightarrow aX \mid \epsilon$$
$$Y \rightarrow bY c \mid \epsilon$$

The language $B$ is context free since it has CFG $G_2$ with rules

$$S \rightarrow XY$$
$$X \rightarrow aX b \mid \epsilon$$
$$Y \rightarrow cY \mid \epsilon$$

But $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$, which we know is not context free from Example 2.36 of the textbook. Thus, the class of context-free languages is not closed under intersection.
(b) Use part (a) and DeMorgan’s law (Theorem 0.20 of the textbook) to show that the class of context-free languages is not closed under complementation.

**Answer:** We will use a proof by contradiction, so we first assume the opposite of what we want to show; i.e., suppose the following is true:

**R1.** The class of context-free languages is closed under complementation.

Define the context-free languages $A$ and $B$ as in the previous part. Then R1 implies $\overline{A}$ and $\overline{B}$ are context-free. We know the class of context-free languages is closed under union, as shown on slide 2-101, so we then must have that $\overline{A} \cup \overline{B}$ is context-free. Then again apply R1 to conclude that $\overline{A} \cup \overline{B}$ is context-free. Now DeMorgan’s law states that $A \cap B = \overline{A \cup B}$, but we showed in the previous part that $A \cap B$ is not context-free, which is a contradiction. Therefore, R1 must not be true.

3. Consider the following CFG $G = (V, \Sigma, R, S)$, where $V = \{S, T, X\}$, $\Sigma = \{a, b\}$, the start variable is $S$, and the rules $R$ are

\[
\begin{align*}
S & \to aTXb \\
T & \to XTS \mid \varepsilon \\
X & \to a \mid b
\end{align*}
\]

Convert $G$ to an equivalent PDA using the procedure given in Lemma 2.21.

**Answer:** First we create a PDA for $G$ that allows for pushing strings onto the stack:

Then we need to fix the non-compliant transitions, i.e., the ones for which a string of length more than 1 is pushed onto the stack. The only non-compliant transitions are the first two from $q_2$ back to itself, and the transition from $q_1$ to $q_2$. Fixing these gives the following PDA:
4. Use the pumping lemma to prove that the language \( A = \{ 0^{2n} 1^3 n 0^n \mid n \geq 0 \} \) is not context free.

**Answer:** Assume that \( A \) is a CFL. Let \( p \) be the pumping length of the pumping lemma for CFLs, and consider string \( s = 0^{2p} 1^3 p 0^p \in A \). Note that \( |s| = 6p > p \), so the pumping lemma will hold. Thus, there exist strings \( u, v, x, y, z \) such that \( s = uvxyz = 0^{2p} 1^3 p 0^p, uv^i y^i z \in A \) for all \( i \geq 0 \), and \( |vy| \geq 1 \). We now consider all of the possible choices for \( v \) and \( y \):

- Suppose strings \( v \) and \( y \) are uniform (e.g., \( v = 0^j \) for some \( j \geq 0 \), and \( y = 1^k \) for some \( k \geq 0 \)). Then \( |vy| \geq 1 \) implies that \( j \geq 1 \) or \( k \geq 1 \) (or both), so \( uv^2 xy^2 z \) won’t have the correct number of 0’s at the beginning, 1’s in the middle, and 0’s at the end. Hence, \( uv^2 xy^2 z \not\in A \).

- Now suppose strings \( v \) and \( y \) are not both uniform. Then \( uv^2 xy^2 z \) will not have the form \( 0 \cdots 01 \cdots 10 \cdots 0 \). Hence, \( uv^2 xy^2 z \not\in A \).

Thus, there are no options for \( v \) and \( y \) such that \( uv^i y^i z \in A \) for all \( i \geq 0 \). This is a contradiction, so \( A \) is not a CFL.

5. The Turing machine \( M \) below recognizes the language \( A = \{ 0^{2n} | n \geq 0 \} \).
In each of the parts below, give the sequence of configurations that $M$ enters when started on the indicated input string.

(a) 00

Answer: $q_100 \xrightarrow{q_20} xq_30 \xrightarrow{q_5x} q_5x \xrightarrow{q_2x} xq_20 \xrightarrow{\omega q_{accept}}$

(b) 00000

Answer: $q_1000000 \xrightarrow{q_2000000} xq_3000000 \xrightarrow{x0q_40000}$
$x0q_50000 \xrightarrow{q_5x000} xq_50000 \xrightarrow{q_5x000} q_5x000 \xrightarrow{q_2x000}$
$xq_20000 \xrightarrow{xq_30000} xxq_30000 \xrightarrow{xx0q_40000} xx0q_40000 \xrightarrow{xx0q_40000}$

$q_1000000 \xrightarrow{q_2000000} xq_3000000 \xrightarrow{x0q_40000}$
$x0q_50000 \xrightarrow{q_5x000} xq_50000 \xrightarrow{q_5x000} q_5x000 \xrightarrow{q_2x000}$
$xq_20000 \xrightarrow{xq_30000} xxq_30000 \xrightarrow{xx0q_40000} xx0q_40000 \xrightarrow{xx0q_40000}$