Albert Nguyen  
CS 91.503  
December 14, 2006  

Homework 10

1. (20 points) Draw a state-transition diagram for a string-matching automaton for the pattern babbaaabba over the alphabet $\Sigma = \{a, b\}$.

<table>
<thead>
<tr>
<th>state</th>
<th>input</th>
<th>a</th>
<th>b</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>3</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>8</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>9</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>10</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

State 11 is the accepting state.

2. (20 points) Working modulo $q = 7$, how many spurious hits does the Rabin-Karp matcher encounter in the text $T = 4126019021256$ when looking for the pattern $P = 601$?

$601 \equiv 6 \pmod{7}$

$\begin{align*}
412 & \equiv 6 \pmod{7} \\
126 & \equiv 0 \pmod{7} \\
260 & \equiv 1 \pmod{7} \\
601 & \equiv 6 \pmod{7} \\
019 & \equiv 5 \pmod{7} \\
190 & \equiv 1 \pmod{7}
\end{align*}$

$\begin{align*}
902 & \equiv 6 \pmod{7} \\
021 & \equiv 0 \pmod{7} \\
212 & \equiv 2 \pmod{7} \\
125 & \equiv 6 \pmod{7} \\
256 & \equiv 4 \pmod{7}
\end{align*}$

There are 3 spurious hits.
3. (20 points) Compute the prefix function \( \pi \) for the pattern ccbaaacabacab when the alphabet is \( \Sigma = \{a, b, c\} \).

Since \( P \) starts with cc, and there are no other occurrences of cc anywhere in \( P \), we can see that \( \pi[q] < 2 \) for \( 2 \leq q \leq 13 \). In fact, the only values of \( q \) for which \( \pi[q] = 1 \) is when \( P_q \) ends in c and \( |P_q| > 1 \). This occurs at \( q = 2, 7, \) and \( 11 \). For the rest \( \pi[q] = 0 \). So we have:

<table>
<thead>
<tr>
<th>( q )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P[q] )</td>
<td>c</td>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>c</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>( \pi[q] )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>


Explain how to determine the occurrences of pattern \( P \) in the text \( T \) by examining the \( \pi \) function for the string \( PT \) (the string of length \( m + n \) that is the concatenation of \( P \) and \( T \)).

\( PT_m = P \). Everything after \( PT[m] \) belongs to \( T \). The first possible occurrence of \( P \) that occurs exclusively in \( T \) will also occur in \( PT[m+1 \ldots 2m] \). If \( P \) occurs at this point, then it must be true that \( \pi_{PT}[2m] \geq m \). Likewise for each \( q \) such that \( \pi_{PT}[q] \geq m \) and \( q \geq 2m \), \( P \) occurs in \( T \) at \( q - m \).
5 (15 points) This problem concerns the paper “Winnowing: Local Algorithms for Document Fingerprinting.” Consider the Robust Winnowing scheme described in Definition 3. Modify it to read:

In each window select the minimum hash value. If possible, break ties by selecting the same hash as the window one position to the left. If not, select the leftmost minimal hash. Save all selected hashes as the fingerprints of the document.

Does this change in the definition change the list of fingerprints that are saved? Explain your answer.

Yes. Take the case where there is a long string of 0s with only one $k$-gram, and say the windows of hashes have length 4. The following sequence of hashes:

00 00 00 00 00 00 00 00

yields the following windows:

(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)
(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)

Under the definition of Robust Winnowing selecting the rightmost minimal hash, the following would be selected:

(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)
(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)

The **bold underline** indicates when a hash has been chosen for the first time. The underline indicates that the hash has already been chosen.

Under the old definition, this example has 2 fingerprints.

Under the modified definition, the following would be selected:

(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)
(00, 00, 00, 00) (00, 00, 00, 00) (00, 00, 00, 00)

As can be seen, once a hash is chosen as a fingerprint, it no longer appears in the next window. The modified definition yields 6 fingerprints.