String matching
Announcements

Programming assignment due Sunday
Prefix vs suffix

w is a prefix of x = \[ x, \] means exists y s.t. wy = x (also implies |w| \leq |x|)
(w ] y = w is a suffix of x)

An easy way to remember prefix vs suffix is: prefix = [, which looks like beginning of an array (similar suffix)
Finite Automata

A finite automata has 5 parts:
(1) A set of states \( Q \)
(2) An initial state \( q_0 \)
(3) Some accepting states, \( A \) subset \( Q \)
(4) An alphabet, \( \Sigma \)
(5) A transition function \( \delta \), from \( Q \times \Sigma \) to \( Q \), namely \( \delta(q,a) = \sigma(P_q a) \)
Finite Automata

Let $\sigma(x) = \max \{k : P_k ] x\}$

So $\sigma$ is the longest prefix of $P$ that is also a suffix of $x$:

$P = \{a, b, a, a, b, c, a\}$

$\sigma(b a a c b a b) = 2$ (all longer bad)
Finite Automata

Compute-Transition-Function(P, ∑)
for q = 0 to |P|
    for each a in ∑
        k = min( |P|, q+1 ) // end P or q
        while: not P_k ] P_q a
            k = k – 1
        δ(q,a) = k // runtime?
Finite Automata

$O(|P|^3|\Sigma|)$, but can get to $O(|P| |\Sigma|)$ if smart

$|P|$ - outside loop
$|\Sigma|$ - outside loop
$|P|$ - repeat runs at most $|P|$ times
$|P| - P_k \ P_q a$ checks $O(|P|)$ chars
Finite Automata

Finite-Automaton-Matcher(T,δ,|P|)
q=0 // q is state
for i = 1 to |T|
    q = δ(q,T[i])
    if q == |P|
        print "Pattern occurs at shift" i-|P|

Runtime = O(|T|)
Finite Automata

1, 2, 3, 4, 5, 6, 7

$P = \{a, b, a, a, b, c, a\}$, then $\delta$ is:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(see FAsigma.py)
Finite Automata

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$S = \{a, b, c, a, b, a, a, b, c, a, c, a\}$

Start 0, see $S_1 = 'a'$, goto 1...

At 1, see $S_2 = 'b'$, goto 2...

At 2, see $S_3 = 'c'$, goto 0...

At 0, see $S_4 = 'a'$, goto 1...
Finite Automata

\[
\begin{array}{cccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
a & 1 & 1 & 3 & 4 & 1 & 3 & 7 & 1 \\
b & 0 & 2 & 0 & 2 & 5 & 0 & 0 & 2 \\
c & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\
\end{array}
\]

\[S = \{a, b, c, a, b, a, a, b, c, a, c, a\}\]

At 1, see \(S_5 = 'b'\), goto 2...

At 2, see \(S_6 = 'a'\), goto 3...

At 3, see \(S_7 = 'a'\), goto 4...

At 4, see \(S_8 = 'b'\), goto 5...
Finite Automata

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

S = \{a, b, c, a, b, a, a, b, c, a, c, a\}

At 5, see $S_9$ = 'c', goto 6...

At 6, see $S_{10}$ = 'a', goto 7... MATCH!

At 7, see $S_{11}$ = 'a', goto 1...

At 1, see $S_{12}$ = 'c', goto 0...
Finite Automata

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\( S = \{a, b, c, a, b, a, a, b, c, a, c, a\} \)

At 0, see \( S_{13} = 'a' \), goto a...

Done, one match found ending at \( S_{10} \) (so match starts \( S_4 \) )
You try it!
P = {a, b, a, a}
S = {a, a, b, a, c, a, a, b, a, a, b, a, a, a, a}

What is automata?
Where are matches?
Lemma 32.2: $\sigma(xa) \leq \sigma(x) + 1$

Obvious...
If $x \parallel z$ and $y \parallel z$, then:

(a) If $|x| \leq |y|$, $x \parallel y$
(b) If $|y| \leq |x|$, $y \parallel x$
(c) If $|x| = |y|$, $x = y$
Lemma 32.3: if $q = \sigma(x)$, then $\sigma(xa) = \sigma(P^q a)$

Proof:

Let $r = \sigma(xa)$ then $P^r$ ] xa and $r \leq q + 1$

So $|P^r| < |P^q a|$ means $P^r$ ] $P^q a$

$\sigma(xa) \leq \sigma(P^q a)$,

$P^q a$ ] xa, so also $\sigma(P^q a) \leq \sigma(xa)$
Theorem 32.4: if \( \Phi \) is the final-state function, then \( \Phi(T_i) = \sigma(T_i) \)

Base: \( T_0 = \varepsilon \), so \( \Phi(T_0) = 0 = \sigma(T_0) \)

\[
\Phi(T_{i+1}) = \Phi(T_ia) = \delta(\Phi(T_i),a) \\
= \sigma(P_{\Phi(T_i)}a) = \sigma(T_ia) = \sigma(T_{i+1})
\]
Knuth-Morris-Pratt

Faster computation by using pattern symmetries within itself (vs transitions for each char/state)

The function $\pi$ does this, namely $\pi(q) = \max(k : k < q \text{ and } P_k \mid P_q)$

Namely, $\pi$ finds shifts of $P$ on itself
Knuth-Morris-Pratt

\begin{table}[h]
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline
  & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
\( P[i] \) & a & b & a & b & a & b & a & b & c & a \\
\hline
\( \pi[i] \) & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 \\
\hline
\end{tabular}
\end{table}

\begin{figure}[h]
\centering
(a)
\end{figure}

\begin{figure}[h]
\centering
(b)
\end{figure}
Knuth-Morris-Pratt

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>P[i]</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>π(i)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

(See: FAsigma.py ... again)
Knuth-Morris-Pratt

KMP-Matcher(T,P,π) // runtime?
q = 0
for i = 1 to |T|
    while q > 0 and P[q+1] ≠ T[i]
        q = π[q]
    if P[q+1] == T[i], then q = q+1
if q == |P|
    match found, and set q = π[q]
The while loop decreases $q$, so it can only run as many times as $q$ increases. $q$ increases only if there is a match in $T$, so at most $|T|$ times.

$O(|T| + |T|) = O(|T|)$

(why not $|T| \times |T|$?)
Knuth-Morris-Pratt

Compute-Prefix-Function(P)

\[ k = 0, \pi[1] = 0 \]

\textbf{for} q = 2 \textbf{to} |P|

\hspace{1em} \textbf{while} k > 0 \textbf{and} P[k+1] \neq P[q]

\hspace{2em} k = \pi[k]

\hspace{1em} \textbf{if} P[k+1] == P[q]

\hspace{2em} k = k+1

\hspace{1em} \pi[q] = k \quad // \text{Runtime} = O(|P|)
KMP correctness

Let $\pi^*[q] = \{\pi[q], \pi[\pi[q]], \ldots, 0\}$

Lemma 32.5: $\pi^*[q] = \{k : k < q \text{ and } P_k \bot P_q\}$

Remember:

$\pi(q) = \max(k : k < q \text{ and } P_k \bot P_q)$,

so fairly obvious (see next slide)

(Tip: prove 2 sets equal by showing $A \subseteq B$ and $B \subseteq A$)
KMP correctness

\[ \pi^*[8] = \{6,4,2,0\} \]
KMP correctness

Lemma 32.6: if $\pi[q] > 0$, then $\pi[q]-1$ in $\pi^*[q-1]$

Proof: $\pi[q] < q$ and $P_{\pi[q]} P_q$, so $\pi[q] - 1 < q - 1$ and $P_{\pi[q]-1} P_{q-1}$ (we know $\pi[q] > 0$, so we can drop a char)

Previous lemma says: $\pi^*[q] = \{k : k < q$ and $P_k P_q \}$, above let $q=q-1$, $k=\pi[q]-1$, then done
Let $E_{q-1} = \{k \text{ in } \pi^*[q-1] : P[k+1] = P[q]\}$

Corollary 32.7: $\pi[q] = \{0 \text{ or } 1+\max\{k \text{ in } E_{q-1}\} \text{ if } E_{q-1} \text{ not empty}\}$

Proof:
Case 1: $E_{q-1}$ empty, no match, so 0
Case 2: By def of $E_{q-1}$, $k+1 < q$ and $P_{k+1}P_q$ implies $\pi[q] \geq 1+\max\{k \text{ in } E_{q-1}\}$
KMP correctness

\( (E_{q-1} = \{k \text{ in } \pi *[q-1] : P[k+1]=P[q] \}) \)

Case 2 (cont): \( \pi[q] \geq 1+\max \{k \text{ in } E_{q-1} \} \)

Let \( r = \pi[q] - 1 \), then \( P_{r+1} P_q \) so \( P[r+1] = P[q] \). Lemma 32.6 says \( r \text{ in } \pi *[q-1] \), so \( r \text{ in } E_{q-1} \).

Thus \( \pi[q] \leq 1+\max \{k \text{ in } E_{q-1} \} \)

Thus \( \pi[q] = 1+\max \{k \text{ in } E_{q-1} \} \)
KMP correctness

$k = \pi[q-1]$ at the start of the for loop in Compute-Prefix-Function alg
The while loop finds $\max\{k \in E_{q-1}\}$ and adds one for Corollary 32.7

If there $k = 0$, then either the max was 0 and it will be incremented to 1 or no match and will stay 0
KMP correctness

KMP alg correctness (map to FA alg):
Base: both start with q=0
Step (q'=σ(T_{i-1})): 
Case σ(T_i)=0: q=0 and same 
Case σ(T_i)=q'+1: while does not run, then increases q, so q=q'+1=σ(T_i) (continued)
Step: $q' = \sigma(T_{i-1})$, Case $0 < \sigma(T_i) < q'$:

while loop terminates when $P[q+1] = T[i]$, so $q + 1 = \sigma(P_{q'}T[i])$

$= \sigma(T_{i-1}T[i])$

$= \sigma(T_i)$, then $q$ is incremented so...

$q = \sigma(T_i)$