The Boyer-Moore algorithm (continued)

Recall there are two rules for Boyer-Moore that we covered last week: the bad character rule and the good suffix rule. There are several issues remaining.

1. For the good suffix rule, what if there is no copy of the matched suffix to the left of the matched suffix? Here we need to be careful: we cannot just shift the entire pattern passing the matched suffix. This is because even there is no complete copy of the matched suffix, there can still be partial copy. This happens if there is a prefix-suffix match in $P$. If there are multiple prefix-suffix matches, pick the longest such match. If such prefix-suffix match does exist, shift $P$ to make the prefix of $P$ match up with the suffix of originally matched $P$. If no such prefix-suffix match exists, then we can shift $P$ by $n = |P|$ positions. Draw a picture here to ensure you understand this point.

2. What if we find a match? That is very similar to the above case: you also need to consider the prefix-suffix match in $P$.

3. If the two rules report different shifts, which one we should take? Answer is taking the maximum shift of the two: both rules work.

4. How do we determine the position of the good suffix? Again, use Z preprocessing. But Z concerns prefix of the string, not the suffix. But this is easy: we reverse $P$. Then it is almost immediate how to find where the suffix part will be: start from $j = 1$ to $n − 1$ (this ensures we will consider the rightmost copy of matched part). Draw a picture here to ensure to understand how this is done.

This is pretty much it. Boyer-Moore is essentially the modification of the naive algorithm with two rules to move at faster pace. This is helped with preprocessing of pattern $P$. The interesting thing is, after these two simple observations, the worst case time is $O(n + m)$. We now move to prove this claimed time bound. I have to say this is not obvious at all.

Analysis of Boyer-Moore algorithm

Before proving the linear running time of Boyer-Moore, a useful property of periodicity of string is useful. First a notation, for a string that can be written as $B = pp \ldots p$ where $p$ is a string and there are $k$ copies of $p$, we write $B = p^k$.

Lemma 2.1. Suppose there are two nonempty strings $A$ and $B$ where $AB = BA$. Then $A = p^i$ and $B = p^j$ for some string $p$.

Proof. To see why this Lemma holds, suppose $|A| > |B|$ (the $=$ case is trivial). Then $A$ is semi-periodic of $B$. We say $A$ is semi-periodic of $B$ if $A = B^kB_1$ where $B_1$ is a prefix of $B$. This fact can be seen through induction on the total length of $A$ and $B$. But drawing a picture will make this clear. Noting that if $B_1 = B$, then we already have a period which is $B$. Otherwise suppose
Then we have $BB_1 = B_1B$. So we can repeat the above process by treating $B$ as $A$ and $B_1$ as $B$. And this will continue until eventually fully periodicity is obtained.

Here is another useful lemma.

**Lemma 2.2.** If pattern $P$ occurs in $T$ with positions $p$ and $p'$ where $p > p'$ and $p - p' \leq \frac{n}{2}$, then $P$ is semiperiodic with period length $p' - p$.

Again, drawing a picture here and it is almost immediate.

Now we start the proof of linear time. For simplicity, we assume only the “Good Suffix Rule” is used. Of course if “bad character rule” is also used, Boyer-Moore will be even faster. But using the good suffix rule alone will allow easier analysis. We also assume pattern $P$ does not appear in $T$. The general case can also be shown with some additional trick (see Gusfield’s book). We divide comparisons of characters into phases, where in each phase there are several matches and end with a mismatch; then shift $P$. Let $s_i$ be the shift of phase $i$. Let $g_i$ be the number of comparisons regarding to $T$ at phase $i$ that are already compared in prior phases (i.e. redundant), and $g'_i$ be the number of comparisons for novel positions of $T$. Clearly $\sum_{i=1}^{K} g'_i \leq m$ since a position in $T$ can be new only once. Also, $\sum_{i=1}^{K} s_i \leq m$ since we can only shift right by at most $m$. Here $K$ is the number of phases. Now I claim $s_i \geq g_i/3$. That is, if there are many redundant comparisons, we can shift at a fast pace. If this holds, $\sum_i g_i \leq 3\sum_i s_i \leq 3m$. The the total number of comparison of characters is at most $4m$, and we are done.

Now, we prove the claim $s_i \geq g_i/3$. Let $t_i$ be the portion of $T$ matching suffix of $P$ and then $P$ shift $s_i$ to the right. First, if shift size $s_i \geq \frac{|t_i|+1}{3}$, then clearly $s_i \geq g_i/3$. This is because $g_i \leq |t_i| + 1$: this is the worst case when even all comparisons count toward $g_i$. Now we deal with the other case: the shift is small. In this case, there will be periodicity in $P$ if the shift is small. Let $\alpha$ be the suffix length $s_i$ of $P$. It is almost immediate that $t_i$ is semiperiodic with $\alpha$. Why? You should draw a picture here. Maybe $\alpha$ is periodic with some smaller period say $\beta$. And so we know $t_i$ is semiperiodic with atomic period of $\beta$. Now another observation is that no previous phase will align $P$ to compare some contained $\beta$ copy. This is detailed in the next Lemma. It is a good idea to think first why this lemma helps. High-level idea: if $P$ has been aligned before, we will violate the assumption that $\beta$ is the smallest period.

**Lemma 2.3.** If $|t_i| + 1 > 3s_i$, then in any phase $h < i$, the right end of $P$ could not have been aligned opposite the right end of full copy of $\beta$ in substring $t_i$ of $T$.

**Proof.** Suppose at phase $h$, $P$ align properly with some internal copy $\beta$. Note $t_i$ is also semi-periodic of $\beta$. Thus, at phase $h$, we will have matches until the left end of $t_i$ called $k'$ (they are all $\beta$ anyway from the right end). Let $k$ be the mismatch position in $P$. But then we have trouble in shifting $P$: no matter what you shift, you run into contradiction. First note you can not shift over the region of $t_i$ (noting $h < i$). So whatever the shift, the right end of $P$ stays within $t_i$. Then we consider two cases. CASE 1: $P$ shift s.t. the right end of $P$ again aligns with a full copy of $\beta$. But this is not possible. Why? Note the good suffix rule says the char prior to the pattern must be different from that of the suffix. If you ever align $P$ to an internal $\beta$, then prior to that shift, the symbol must be the same as the suffix one (which violates the good suffix rule). CASE 2: $P$ shifts s.t. $P$ aligns with an internal char of some $\beta$. In this case, we observe $P$ will semi-periodic with a smaller phase since we have a rotation of string making $\beta = AB = BA$, and this will lead to a smaller period than $\beta$. Contradiction. Now we have the following observation: no previous phase will align $P$ with internal $\beta$ (and thus only the two ending $\beta$ may have more comparisons).

By a similar reasoning (proof omitted), we also have this lemma.
Lemma 2.4. If $|t_i| + 1 > 3s_i$, then in any phase $h < i$, $P$ can match $t_i$ in $T$ for at most $|\beta| - 1$ chars.

That is, any previous comparison can not be too long: no more than $\beta$. Otherwise contradict the selection of $\beta$.

One last lemma (will not prove; similar reasoning) is:

Lemma 2.5. If $|t_i| + 1 > 3s_i$, then in phase $h < i$ if ever $P$ aligns to some character in $t_i$, it only can align to the left $|\beta| - 1$ or the rightmost $\beta$.

That is, $P$ can only be aligned to some small number of positions.

Now put everything together.

Theorem 2.6. Assume $P$ does not occur in $T$, then $s_i \geq g_i/3$.

Proof. We only consider the case $|t_i| + 1 > 3s_i$ (i.e. small shift). By Lemma 2.5, at any previous phase $h < i$, the right end of $P$ is opposite to at most the left end or right end with $\beta$ each. Then by Lemma 2.4, the number of comparisons cannot be longer than $|\beta|$. And so the comparison can not be more than another $\beta$ to the left of the rightmost $\beta$. So the number of characters of $t_i$ compared before is bounded by $3|\beta| \leq 3s_i$ (note: $s_i = |\alpha| \geq |\beta|$).