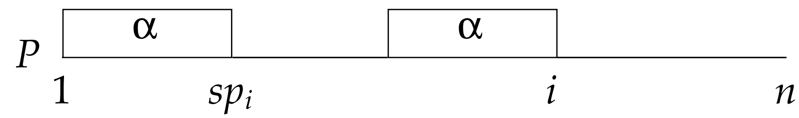
The Knuth-Morris-Pratt Algorithm

- $O(n+m)$ in worst case.
- Slower than Boyer-Moore in practice.
- Can be adapted to search for k patterns of total length n in $O(k + n + m)$ time (Aho-Corasick).

Key idea:

Preprocess P to get larger shifts at mismatches (analogous to Boyer-Moore, but adapted to left-to-right scan).

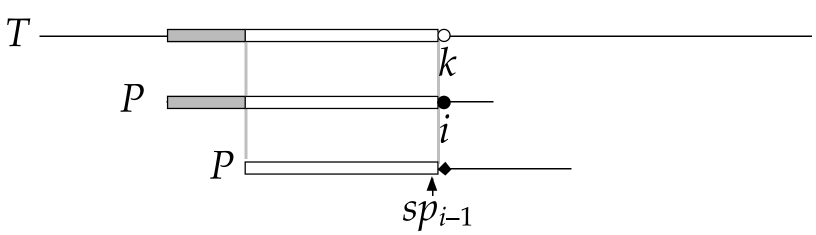
Definition. For each position i in P, $sp_i(P)$ is the length of the longest proper *suffix* of $P[1\,..\,i]$ that matches a *prefix* of P.



i	1	2	3	4	5	6	7	8	9	0	1
$P[i]$	a	b	a	b	b	a	b	a	b	a	b
sp_i	0	0	1	2	0	1	2	3	4	3	4

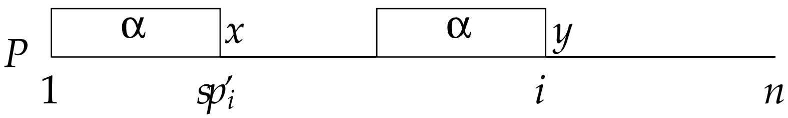
The Weak KMP Shift Rule

For any alignment of P and T, if the first mismatch is between $P[i]$ and $T[k]$, then shift P right by $i - (sp_{i-1} + 1) = i - sp_{i-1} - 1$ places, so that $P[sp_{i-1} + 1]$ is aligned with $T[k]$. If an occurrence of P is found, then shift P right by $n - sp_n$ places.




```
abababbababbaababbababababbababab
ababbababab
12345678901
abababbababbaababbababababbababab
  ababbababab
  12345678901
abababbababbaababbababababbababab
       ababbababab
       12345678901
abababbababbaababbababababbababab
            ababbababab
            12345678901
abababbababbaababbababababbababab
             ababbababab
             12345678901
                    ababbababab
```

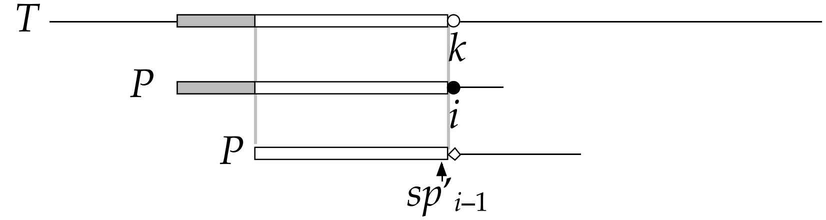
Definition. For each position i in P, $sp'_i(P)$ is the length of the longest proper *suffix* of $P[1..i]$ that matches a *prefix* of P such that $P[i+1] \neq P[sp'_i+1]$.



i	1	2	3	4	5	6	7	8	9	0	1
$P[i]$	a	b	a	b	b	a	b	a	b	a	b
sp'_i	0	0	0	2	0	0	0	0	4	0	4

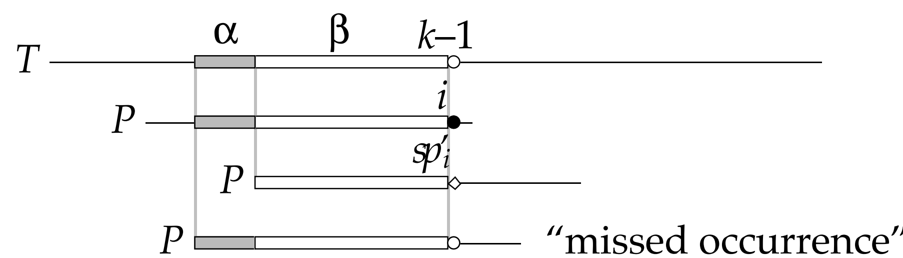
The Strong KMP Shift Rule

For any alignment of P and T, if the first mismatch is between $P[i]$ and $T[k]$, then shift P right by $i-(sp'_{i-1}+1) = i - sp'_{i-1} - 1$ places, so that $P[sp'_{i-1}+1]$ is aligned with $T[k]$. If an occurrence of P is found, then shift P right by $n - sp'_n$ places.




```
ababababbababbaababbababababbababab
ababbababab
12345678901
ababababbababbaababbababababbababab
  ababbababab
  12345678901
ababababbababbaababbababababbababab
        ababbababab
        12345678901
ababababbababbaababbababababbababab
                ababbababab
                12345678901
                        ababbababab
```


The strong KMP shift rule does not miss any occurrences of P . . .



The same is true for the weak rule.

The Knuth-Morris-Pratt algorithm does at most $2m$ character comparisons:

- A *compare/shift phase* consists of all comparisons between successive shifts.
- After a shift, comparisons start with either the last character from T that was examined in the previous phase or with the character to its right.

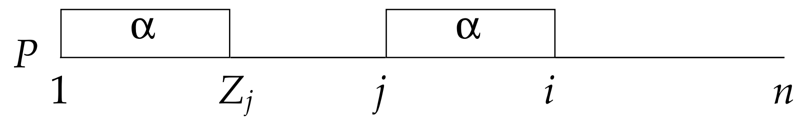
$\Rightarrow$ Total number of comparisons is at most $m + s$, where $s = \#(\text{shifts})$

Since $s < m$, $\#(\text{comparisons}) < 2m$.

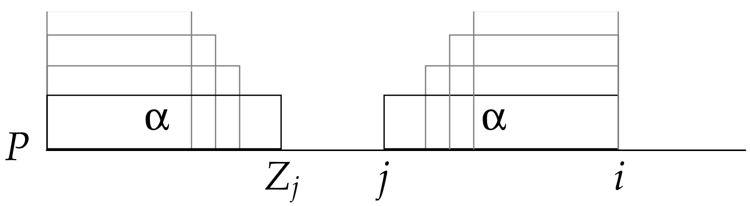
Definition. Position $j > 1$ *maps to* i if

$$i = j + Z_j(P) - 1$$

(i.e., i is the right end of a Z-box starting at j).



Theorem. For any $i > 1$, $sp'_i(P) = Z_j = i - j + 1$, where j is the smallest position that maps to i. If there is no such j then $sp'_i(P) = 0$.



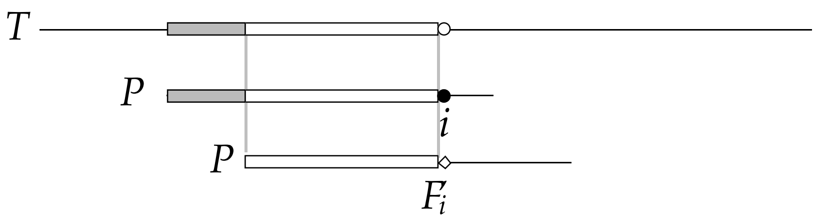
Example. For $P =$ `ababbababab`, two Z-boxes end at $P[11]$: `ab` (Z_{10}) and `abab` (Z_8). Thus, $sp'_{11}(P) = 4$.

Preprocessing to find the sp'-values:

compute $Z(P)$
for $i \leftarrow 1$ **to** n **do**
    $sp'_i(P) \leftarrow 0$
for $j \leftarrow n$ **downto** 2 **do**
    $i \leftarrow j + Z_j(P) - 1$
    $sp'_i(P) \leftarrow Z_j(P)$

All sp'-values can be computed in $O(n)$ time.

Definition. For $i = 1, 2, \ldots, n+1$, the (strong) *failure function* is $F'(i) = sp'_{i-1} + 1$, where $sp'_0 = 0$.



KMP(P,T)

compute the failure function F'

$c \leftarrow 1$; $p \leftarrow 1$

while $c + (n - p) \leq m$ **do**

while $P[p] = T[c]$ **and** $p \leq n$ **do**

p++; c++

if $p = n + 1$ **then**

report occurrence of P starting at $T[c - n]$

if $p = 1$ **then** c++

$p \leftarrow F'(p)$