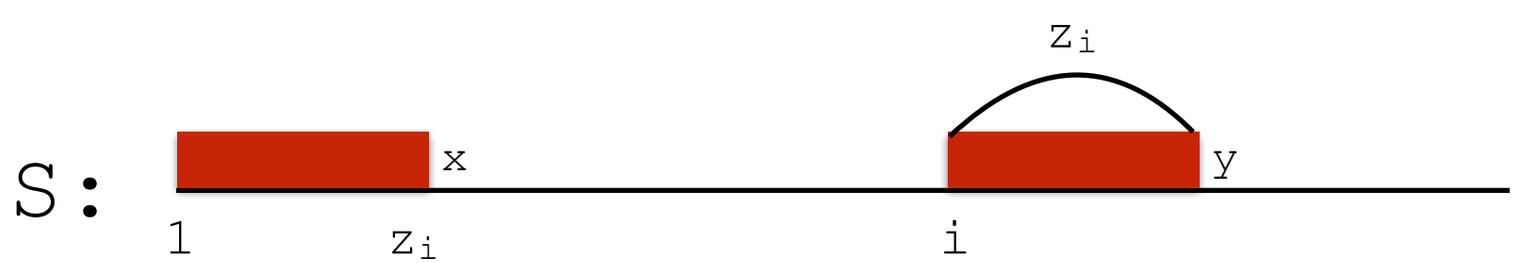
Def: Given string *S* and position *i>1*, $z_i(S)$ is the length of the longest substring of *S* starting at position *i* that matches a prefix of *S*.



S: ABCDABD

$z_5 = 2$

Solving pattern matching problem with z-scores:

```
T:  ABCABCDABABCDABDABDE
P:  ABCDABD
```

```
S:  ABCDABD$ABCABCDABABCDABDABDE
                ↑
z18 = |P|
```

Suppose you have all *z*-scores, then finding occurrences of P is O(|T|)

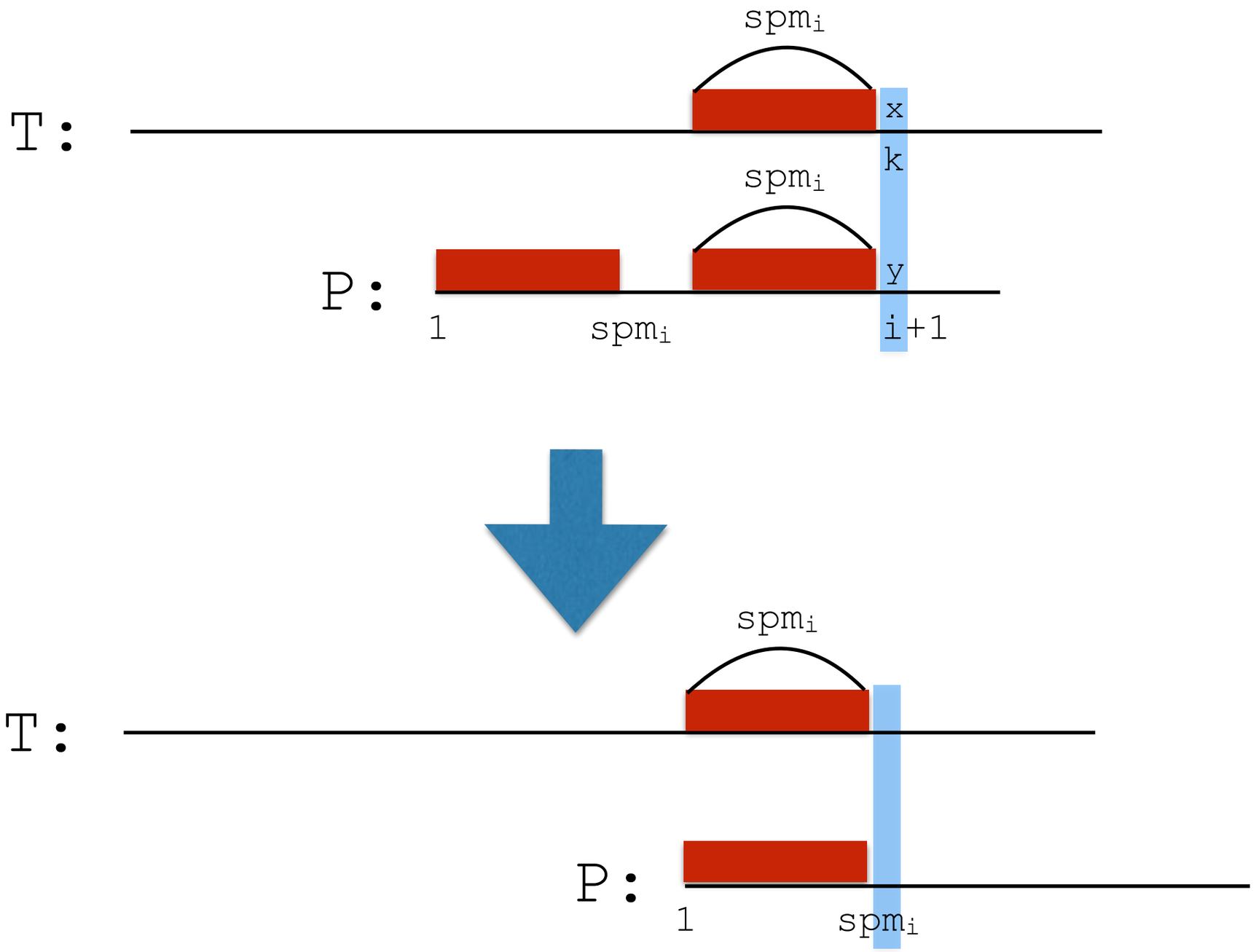
KMP algorithm

```
T:  ABCABCDABCA…
P:  ABCABD
```

```
T:  ABCABCDABCA…
P:     ABCABD
```

S: ABCDABD

$spm_6 = 2$


T:
spm_i
x
k
P:
spm_i
y
1
spm_i
i+1
T:
spm_i
P:
1
spm_i

P: ABCDEFGHABCDC

$spm_{12} = 4$

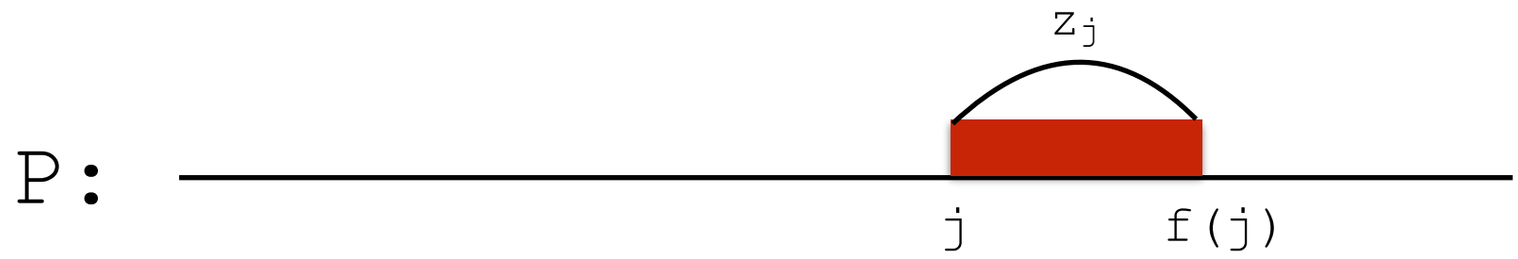
T: …ABCDEFGHABCDE…

P: ABCDEFGHABCDC

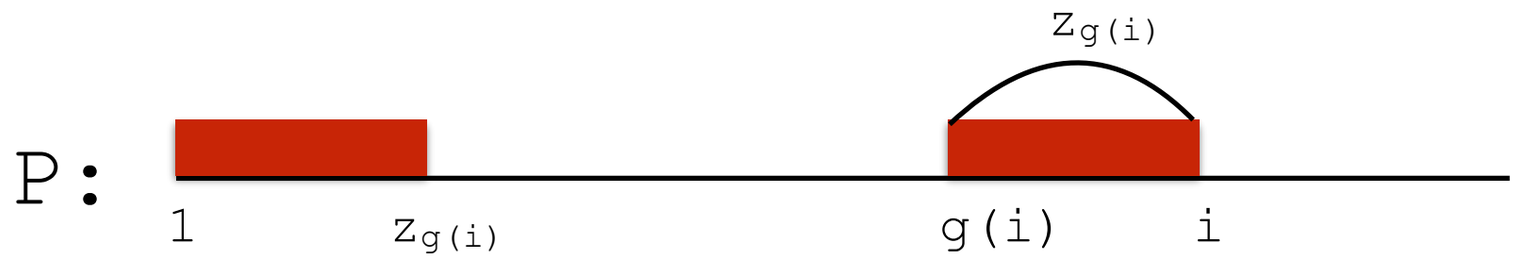
T: …ABCDEFGHABCDE…

P: ABCDEFGHABCDC

Def. *f(j)*: right end of z-box starting at position *j* of *P*



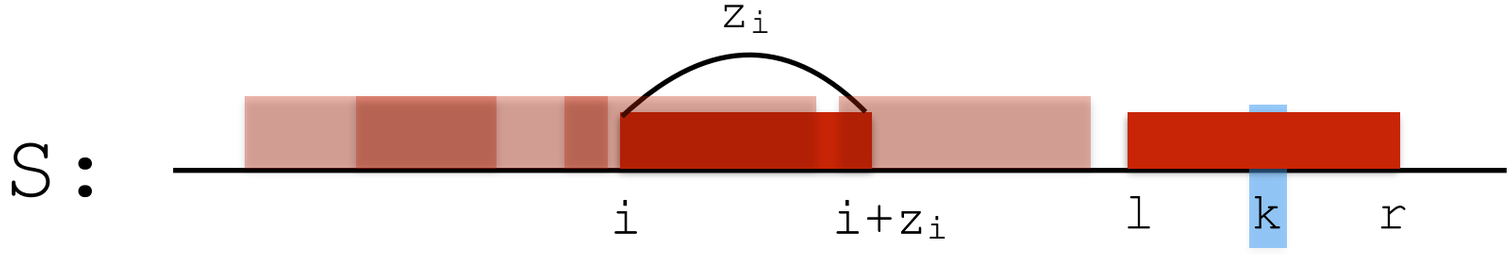
Def. $g(i) = \min\{j \mid f(j) = i\}$, i.e., the left-most starting point of z-boxes ending at position *i*



Thm. $spm_i = z_{g(i)}$

Linear time algorithm to calculate z-scores of a string S

Def. *l:* start of right-most *z*-box found through *k=2,…,k-1*
r: end of right-most *z*-box found through *k=2,…,k-1*



Linear time algorithm to calculate z-scores of a string S

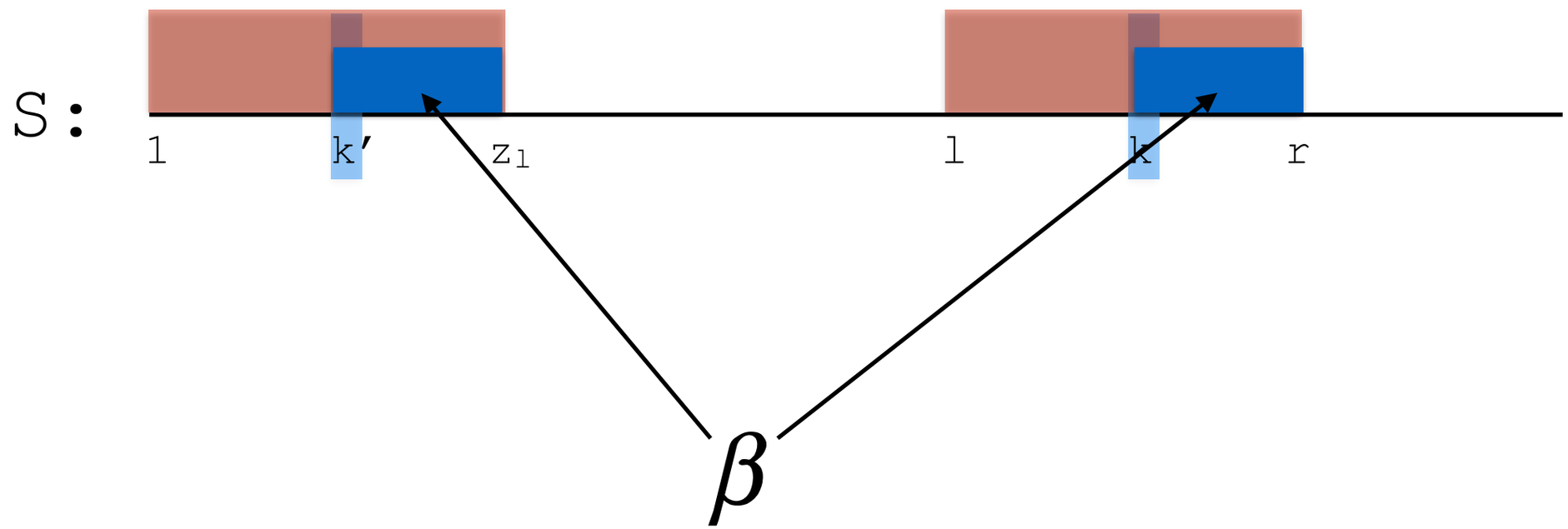
By induction on position *k* of string *S*

k=2: Compare $S[2,...] and S[1,...]$ until first mismatch.
Suppose found $q \geq 0$ matches, then set

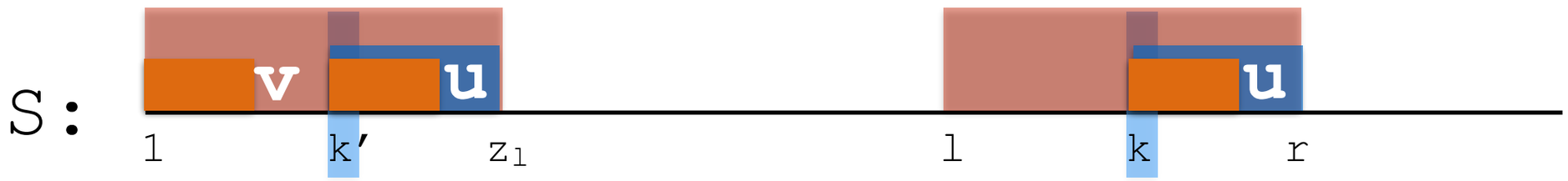
$$z_2 = q$$
$$r = 2 + q$$
$$l = 2$$

k>2: Suppose already computed $z_2, z_3, \ldots, z_{k-1}$

Case 2: $k \leq r$

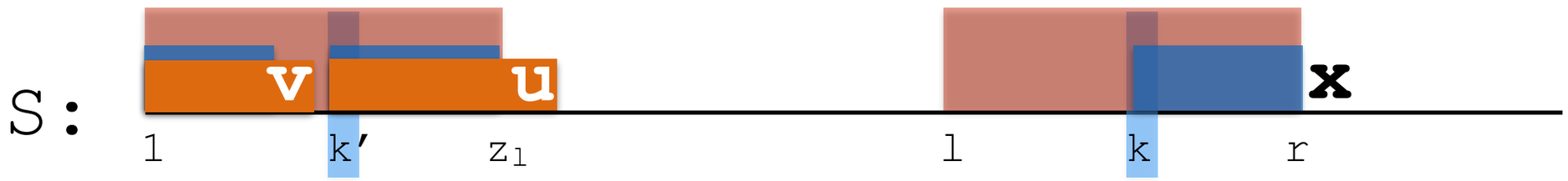


Case 2a: $k \leq r, z_{k'} < |\beta|$



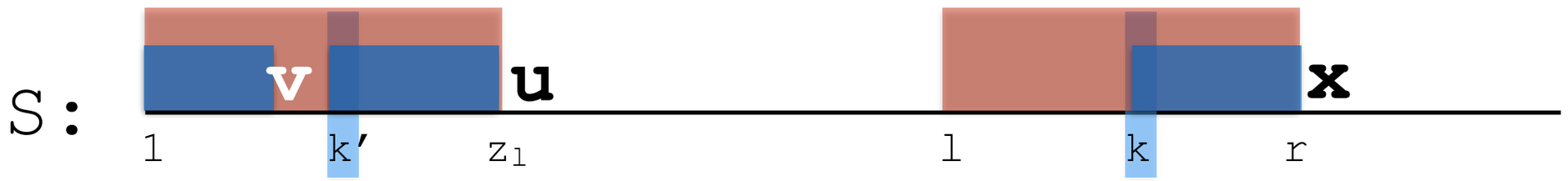
$$u \neq v \Longrightarrow z_k = z_{k'}$$

Case 2b: $k \leq r, z_{k'} > |\beta|$



$$u = v \wedge x \neq u \Longrightarrow x \neq v \Longrightarrow z_k = |\beta|$$

Case 2c: $k \leq r, z_{k'} = |\beta|$



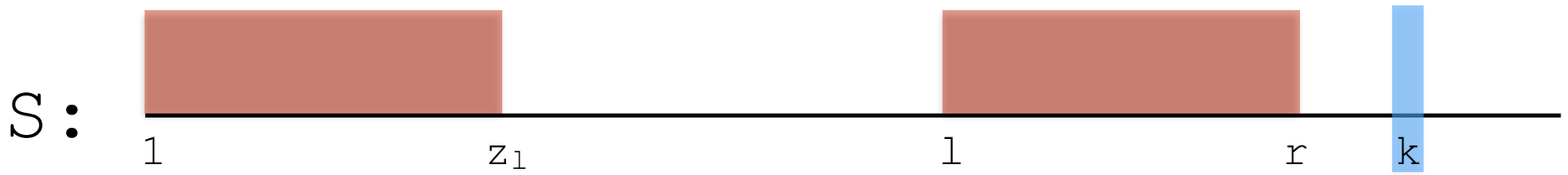
$$u \neq v \wedge x \neq u \Longrightarrow ??$$

Compare $S[r+1,...] and S[|\beta|+1,...]$ until first mismatch.

$$r_k = r_{k-1} + q$$

$$l = k$$

Case 1: $k > r$



Compare $S[k,...] and S[1,...]$ until first mismatch.
Suppose found $q \geq 0$ matches, then set

$$r_k = k + q$$

$$l = k$$

Run-time analysis:

- Note that $r_k >= r_{k-1}$
- Therefore, what we need to worry about is how many comparisons we make when moving this pointer in each iteration
    - # matches + 1 (at most) mismatch
    - if a character is matched, it is not compared again, so # matches is O(|S|)
    - # mismatches is O(|S|) since there are O(|S|) iterations