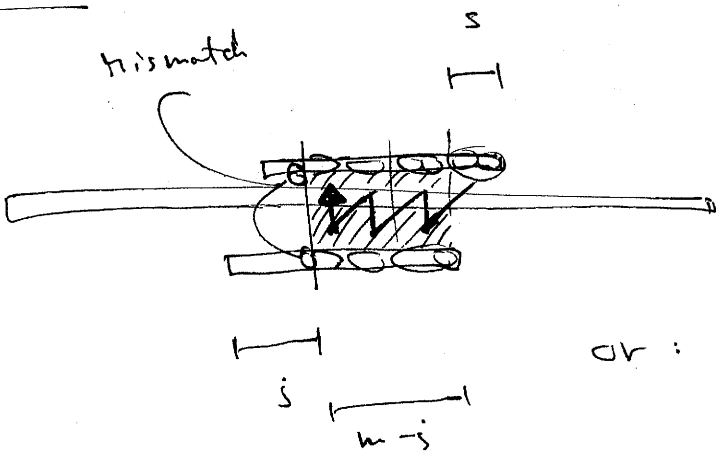
0

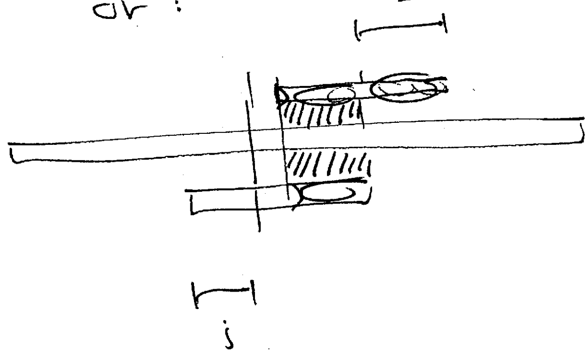
Basic Fact ← From BM proof   By def. of shift

Case 1



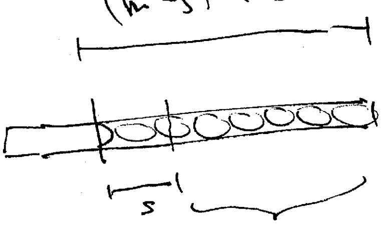
or :

Case 2



So we know that for v = ○ being the smallest "root" of ⬭ we have

(OR m)

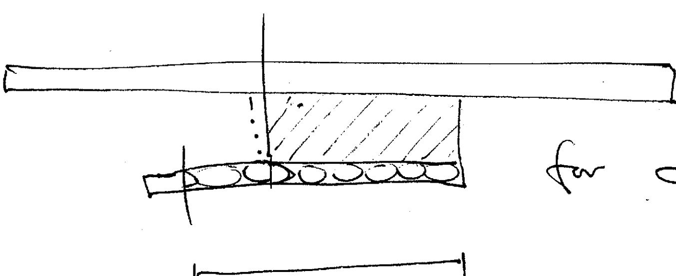


pat

current match

(v = smallest string such that ⬭ $= \underbrace{v.v \ldots v}_{k \geq 1}$)

I.e.: current match :



for ○ primitive.

(OR entire m)

Case 1   Case 2

BM Proof

0) Define match (area) of [unsuccessful] iteration of BM.

1) Repeat basic fact (see other note page)

Note: based on current match exit of (i.e.; shift made.)

v primitive, $|v| \le s$.

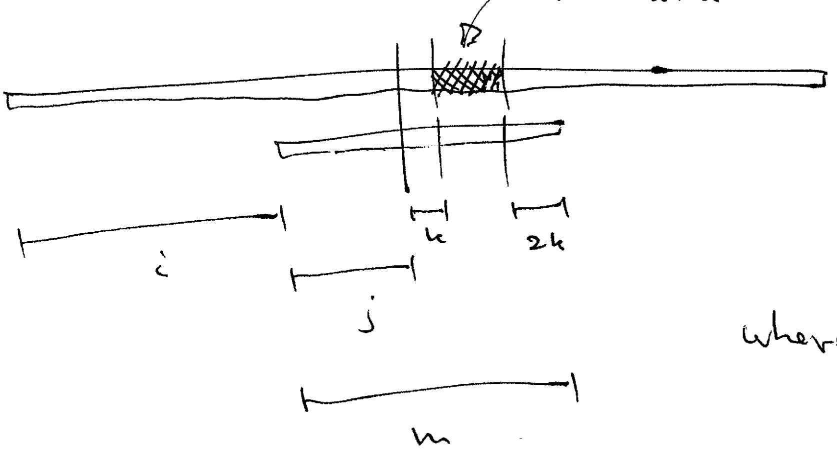
s = BM_Shift (current)

2) Prop.: A primitive string has all its cyclic shifts unique (they are all different). Proof: Not here.

3) Goal: Show that no previous match (iteration) can have overlapped

text[$i + j + k$ .. $i + m - 2k$]

new area touched.



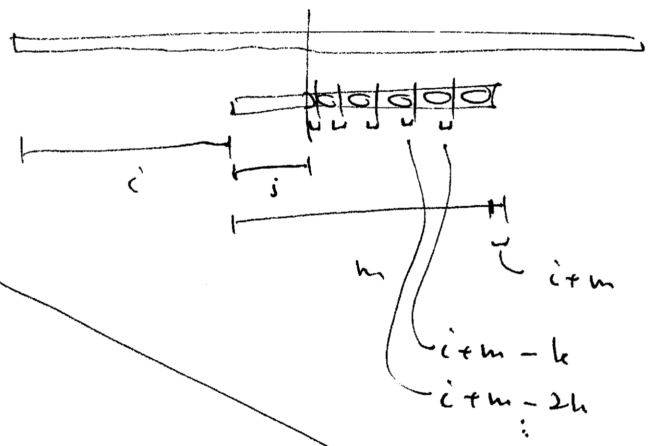
where $k = |v|$
$\le s$
= shift of current iteration.

Since then work this it. $\le$ $3k$ + new area $\le$ $3s$ + new area

So: $\sum_{\text{all it}}$ work $\le 3(n - \cancel{m}) + n \le 4 \cdot n$

<u>Def.</u>: Critical pos: $i+m-t\cdot k$ for t integer ≥ 1.

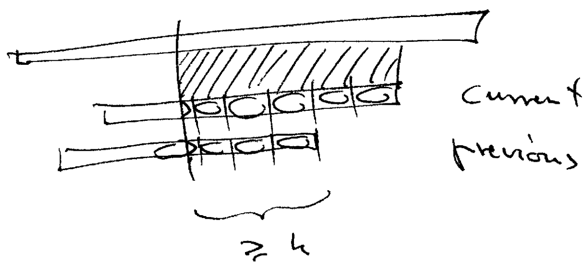
Claim 1:



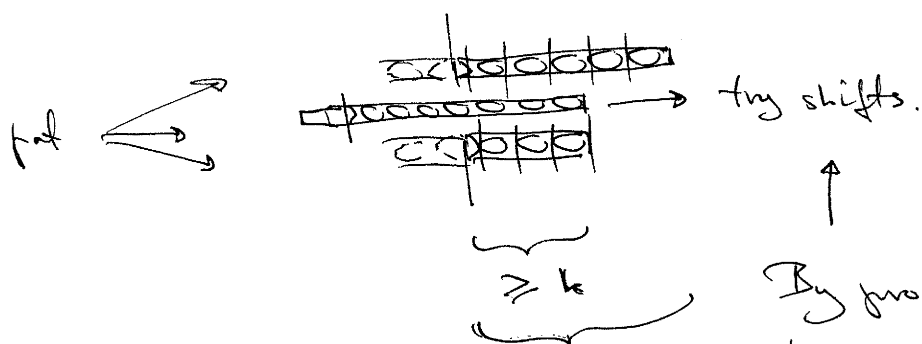
No prev. it. can have match ending at critical pos. at $\geq i+j+k$

<u>Proof</u>: Assume one has.

By basic fact, that previous ~~matches~~ extends back ~~[illegible]~~ to $i+j+1$ exactly.

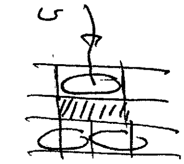


Look at finding BM-Shift for current and for previous:



By proposition, can only match here when shifted multiples of k.

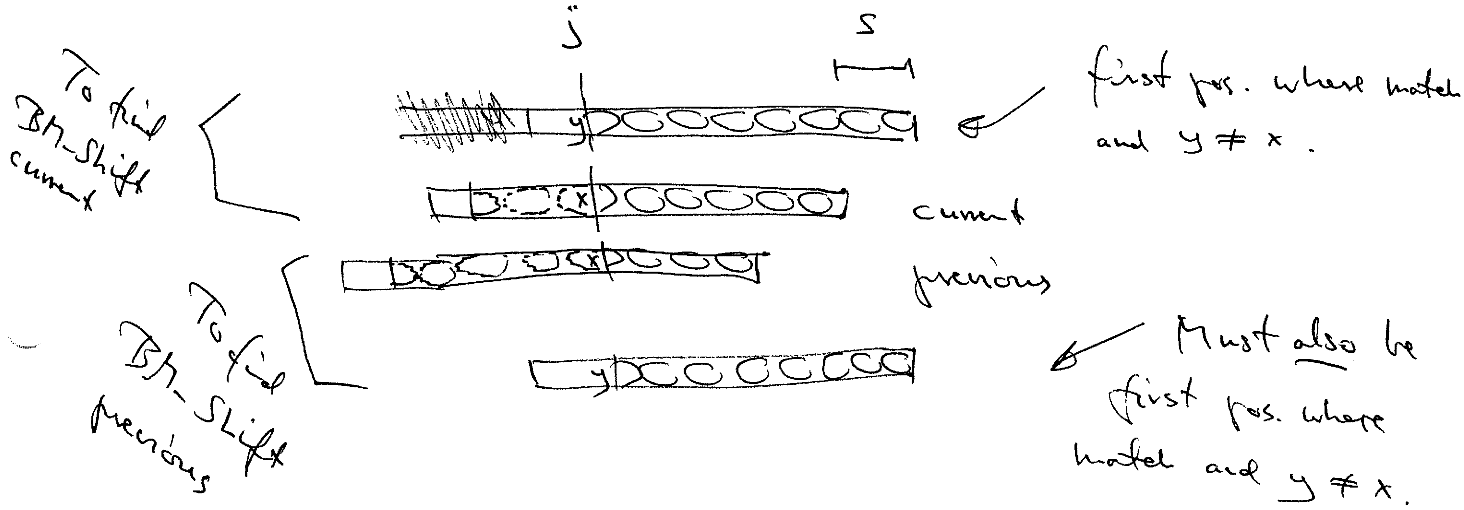
Else we have:



which is impossible by prop.

[NB: here we use previous match $\geq k$! (Not included in Lecroq's Claim 1!)].

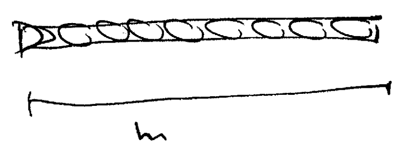
By the mismatch property/part of BM_Shift, both current and previous must shift to <u>same</u> new position



<u>BUT</u>: This means current never existed!
(previous would have led to next after current).
Contradiction.
□.

<u>Note</u>: <u>This</u> is where the mismatch part of BM-Shift is used in proof. <u>And</u> that current match is unsuccessful (else previous could easily only shift k) (if close to successfull – as we now don't know it stops matching at j)

(This is for case 1 in current. Case 2 is similar (period extends over entire pattern

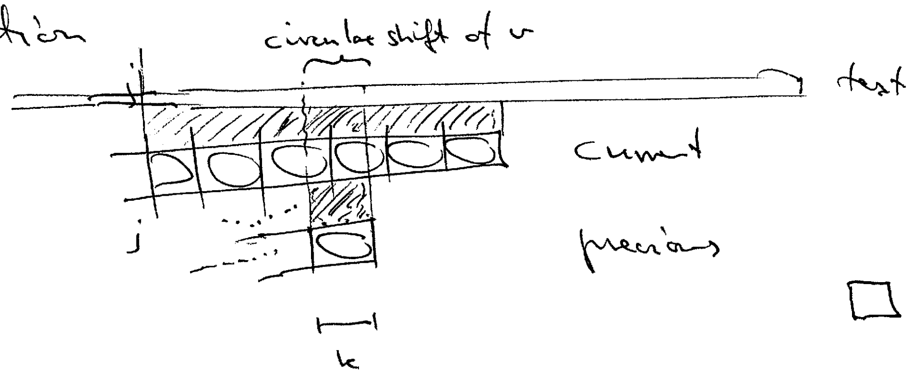


, so no mismatch will occur against pattern (as only synchronized position [synchronized at critical pos.] possible for matches) so will stop first time a synchronized position completely past j is reached – <u>same</u> in current and prev.)

Claim 2 No previous match can have overlap $\geq k$ with current match.

Proof: Previous cannot end at crit pos., by Claim 1 (or at end of current, as shift ≥ 1 between iterations).

But then we have a conflict with primitivity-proposition

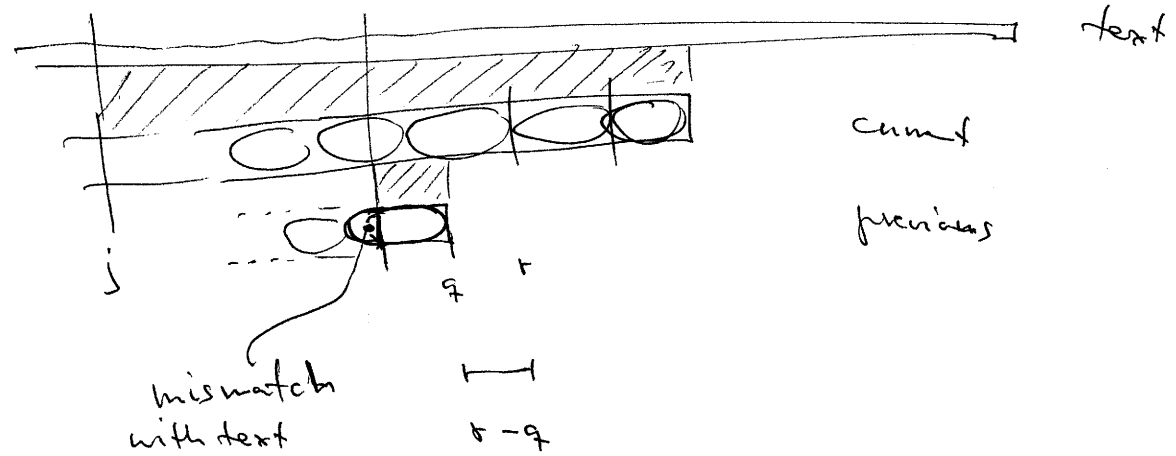


□

So if previous match ends at text[i+j+k] or later, it must have size $< k$. Hence, be contained in current match, in particular.

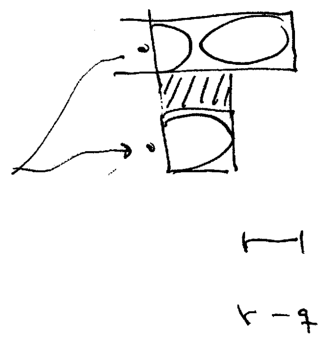
Claim 3 If previous match is ~~included~~ contained completely in current match, it must end past last (leftmost) critical pos.

<u>Proof</u>: Assume a previous match ends left of some critical position:



mismatch
with text
⇒ mismatch
with current (as
contained in
current match
area)

Part of Picture again:



shows that shift of $r-q$ is fullfilling criteria for BM_Shift (previous).
Is maybe not <u>smallest</u> such shift.

So $1 \le$ BM_Shift $\le r-q$.

If BM_Shift $< r-q$, repeat argument with next iteration after previous. This next match

(but still $\leq r$)

must ~~…~~ strictly closer to r. Hence, we

end (repeating)

get seq. of matches with increasing end points $\leq r$

strictly

$\Rightarrow$ must reach r (a critical point).

Contradiction with Claim 1.

$\square$

By Claim 2 and Claim 3, no previous match can overlap. text[$i+j+k..i+m-2k$]

So goal is reached.