Are Three Squares Impossible?
III — The "Easy" Case & The Future

Bill Smyth

Algorithms Research Group, Department of Computing & Software
McMaster University, Hamilton, Canada
email: smyth@mcmaster.ca

10–12 May 2012

Abstract: Series of Talks

These talks describe work done over the last 30 years or so both to understand and to compute repetitions in string — especially since 1999. We will discover that, although much has been learned, much combinatorial insight gained, there remains much more that is unknown about the occurrence of repetitions in strings and the restrictions they are subject to. I present combinatorial results discovered only recently, and I suggest that possibly extensions of these results can be used to compute repetitions in an entirely new way. I hope that members of the audience will be motivated to work on some of the many open problems that remain, to extend combinatorial knowledge even further.

Abstract of Talk III

We describe the “easy case” when $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ occur at the same location with $u < v \leq 3u/2$ — even two squares can scarcely occur! Then we show how all this work has barely scratched the surface — we take a timorous look ahead to the General Case and its consequences that await us sometime in the future!

Outline

The "Easy" Case

Two squares $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ occur at the same position, but now we suppose $u < v \leq 3u/2$. We get a breakdown into substrings $\boldsymbol{u_1}$ & $\boldsymbol{u_2}$, but a DIFFERENT breakdown, a DIFFERENT $\boldsymbol{u_1}$ & $\boldsymbol{u_2}$:

Lemma

If $\boldsymbol{x} = \boldsymbol{v}^2$ with prefix $\boldsymbol{u}^2$, $u < v \leq 3u/2$, then

$$\boldsymbol{x} = \boldsymbol{u_1}^m \boldsymbol{u_2} \boldsymbol{u_1}^{m+1} \boldsymbol{u_2} \boldsymbol{u_1}, \tag{1}$$

where $u_1 = v - u \leq u/2$, $u_2 = u \bmod u_1 \geq 0$, $m = \lfloor u/u_1 \rfloor \geq 2$, and $\boldsymbol{u_2}$ is a proper prefix of $\boldsymbol{u_1}$.

The “Easy” Case — Examples

Consider the example

$$\boldsymbol{x} = \underline{(aabaabaaba)(aab}\textcolor{red}{||}\underline{aabaaba)abaaab}, \tag{2}$$

with $\boldsymbol{u} = aabaabaaba$, $\boldsymbol{v} = aabaabaabaaab$, $\boldsymbol{u_1} = aab$, $\boldsymbol{u_2} = a$, $m = 3$. Corresponding to the square $\boldsymbol{u}^2 = (\boldsymbol{u_1}{}^3\boldsymbol{u_2})^2$, there is a run $\boldsymbol{u}^2a$.

For

$$\boldsymbol{x} = \underline{(abbabbabba)(abb}\textcolor{red}{||}\underline{abbabba)bbaabb}, \tag{3}$$

we again have $\boldsymbol{u_2} = a$, $m = 3$, but now $\boldsymbol{u_1} = abb$ and the run $\boldsymbol{u}^2$ cannot be extended.

These examples can be generalized, as we now discover:

The "Easy" Case — Runs

(R1) $\boldsymbol{v}^{\mathbf{2}}$ and $\boldsymbol{u}^2\boldsymbol{u}^*$ for some possibly empty proper prefix $\boldsymbol{u}^*$ of $\boldsymbol{u_1}$ such that both $\boldsymbol{u}^*$ and $\boldsymbol{u_2}\boldsymbol{u}^*$ are prefixes of $\boldsymbol{u_1}$; for example, $\boldsymbol{u}^* = a$ in (2), ε in (3).

(R2) $\boldsymbol{u}\boldsymbol{u}^* = \boldsymbol{u_1}^m\boldsymbol{u_2}\boldsymbol{u}^*$ and $\boldsymbol{u_1}\boldsymbol{u}\boldsymbol{u}^* = \boldsymbol{u_1}^{m+1}\boldsymbol{u_2}\boldsymbol{u}^*$, runs that may be adjacent as in (3) or overlap as in (2), and that together cover all of $\boldsymbol{x}$ except for a suffix of the final copy of $\boldsymbol{u_1}$.

(R3) $m+1$ runs

$$\boldsymbol{u_2}^2\boldsymbol{u}^*, (\boldsymbol{u_1}\boldsymbol{u_2})^2\boldsymbol{u}^*, \ldots, (\boldsymbol{u_1}^m\boldsymbol{u_2})^2\boldsymbol{u}^* = \boldsymbol{u}^2\boldsymbol{u}^*, \tag{4}$$

all centred at position $u+1$ of $\boldsymbol{x}$, with the first one $\boldsymbol{u_2}^2\boldsymbol{u}^*$ repeated at position $(2m+1)u_1+u_2+1$. The centred runs (4) arise in the analyses of [R06] and [CI08].

(R4) Miscellaneous runs of period strictly less than u_1. For example, the runs aa that occur as a substring of occurrences of $\boldsymbol{u_1}$ in (2). Another example: in the case $\boldsymbol{u_1} = abaab$, $\boldsymbol{u_2} = a$, $m = 2$,

$$\boldsymbol{x} = abaababaabaabaab\,abaababaabaabaab,$$

we identify, in addition to $2m+4$ runs aa, a sequence of four overlapping runs $(aba)^2$, $(aba)^4$, $(aba)^2$, $(aba)^3ab$, that cover $\boldsymbol{x}$.

The "Easy" Case — Main Result

Lemma

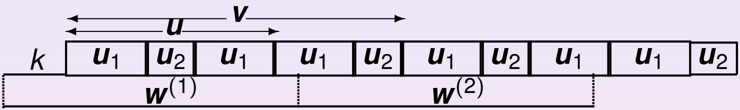
The string (1), $u_2 > 0$, contains no repetitions (runs) of period $z \geq u_1$ other than those characterized in (R1)-(R3).

Thus for $\boldsymbol{x} = \boldsymbol{u_1}^m \boldsymbol{u_2} \boldsymbol{u_1}^{m+1} \boldsymbol{u_2} \boldsymbol{u_1}$ of length

$$x = (2m+2)u_1 + 2u_2 = (2u_1)m+2(u_1+u_2),\ u_2 > 0,$$

there exist exactly $m+5$ runs (R1)-(R3), together with runs (R4) of period strictly less than u_1, and no others: over a wide range of lengths w, no square $\boldsymbol{w}^2$ can occur!

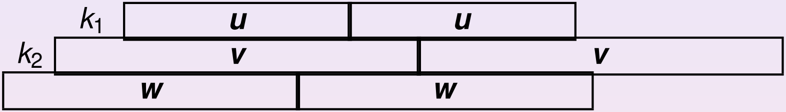
Negative k



This simple extension of the problem has not been studied at all:

* What values of k will be interesting?
* What values of w?
* Note that $\boldsymbol{w}$ will be forced to have a prefix composed of some substring of a rotation of $\boldsymbol{u}$; in some cases, this will result in an illegal leftward extension of either the run $\boldsymbol{u}^2$ (when the stub $\boldsymbol{k}$ ends with a suffix of $\boldsymbol{u_1}$) or the run $\boldsymbol{v}^2$ (when $\boldsymbol{k}$ ends with a suffix of $\boldsymbol{u_2}$).
* In particular, the letter to the left of $\boldsymbol{u}^2$ is constrained NOT to be either the last letter of $\boldsymbol{u}$ or the last letter of $\boldsymbol{v}$!

Three Neighbouring Squares



Nothing is known!

But it is certain that somewhere in the area of overlap of the three squares, analysis will reveal a similar breakdown into repetitions of small period.

Three squares are ... impossible!

A New Kind of Repetitions Algorithm?

If three squares are impossible, then shouldn’t we be able to process the string left-to-right, just keeping track of a “small” square ($\boldsymbol{u}^2$) and a “big” square ($\boldsymbol{v}^2$), while watching out for multiple repetitions of small period?

???

Exercise

Write

(a) pseudocode

(b) a C program

that inputs a value x_{max}, and then

* for all values $x \in 3..x_{max}$,
* for all $u \in 2..x-1$ such that $x \bmod u > 0$,
* for all $v \in 1..x-1$,

computes $\boldsymbol{x}$ subject to the constraint that $\boldsymbol{x}$ and $R_v(\boldsymbol{x})$ both have period u (perhaps using a methodology similar to that of Jenya's program). Then compare your result to the result predicted by the lemma “Two Rotations, Same Period”; if the result is not the same, make a suitable output.

(Maybe you'll find an error in the lemma — it's only been “proved” by a human brain, not checked by a computer!)

Send your solution to `smyth@mcmaster.ca` with subject line “Two Rotations, Same Period”.

- Maxime Crochemore & Lucian Ilie, **Maximal repetitions in strings**, *J. Comput. Sys. Sci.* (2008) 796–807.
- Evguenia Kopylova & W. F. Smyth, **The three squares lemma revisited**, *J. Discrete Algorithms 11* (2012) 3–14.
- Wojciech Rytter, **The number of runs in a string: improved analysis of the linear upper bound**, *Proc. 23rd Symp. Theoretical Aspects of Computer Science*, B. Durand & W. Thomas (eds.), LNCS 2884, Springer-Verlag (2006) 184–195.