Completing the New Periodicity Lemma

Widmer Bland

A thesis submitted to the School of Graduate Studies in partial fulfilment of the requirements for the degree of Master of Science

Master of Science
Computer Science
2014

Department of Computing & Software
McMaster University
Hamilton, Ontario, Canada

Title:	Completing the New Periodicity Lemma
Author:	Widmer Bland
Supervisor:	Dr W.F. Smyth
Number of pages:	iv, 45

Abstract

The "Three Squares Lemma" (Crochemore and Rytter 1995) famously explored the consequences of supposing that three squares occur at the same position in a string. Essentially, it showed that this phenomenon could not occur unless the longest of the three squares was at least the sum of the lengths of the other two. More recently, several papers (Fan et al. 2006; Franek, Fuller, et al. 2012; Kopylova and Smyth 2012; Simpson 2007) have greatly extended this result to a "New Periodicity Lemma" (NPL) by supposing that only two of the squares occur at the same position, with a third occurring in a neighbourhood to the right. The proof of the NPL involves fourteen subcases, twelve of which have been proven over the last seven years. In this thesis, we prove the final two remaining.

Acknowledgements

Bill Smyth imparted his enthusiasm for strings, and provided direction and encouragement that enabled my work. I feel greatly fortunate to have had his mentorship.

Michael Soltys offered helpful comments on this thesis and was kindly supportive on many occasions over the last few years.

Thanks.

Contents

List of Figures

Chapter 1

Introduction

There has for several years been considerable interest in the limitations that may exist on periodicity in strings. The "Three Squares Lemma" (Crochemore and Rytter 1995) showed that three squares could exist at the same position in a string only if the longest of the three was at least the sum of the lengths of the other two. A sequence of papers (Fan et al. 2006; Franek, Fuller, et al. 2012; Kopylova and Smyth 2012; Simpson 2007) greatly generalized this result by considering two squares $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ at the same position, with however the third square $\boldsymbol{w}^2$ offset a distance $k \geqslant 0$ to the right. First stated and proved as the "New Periodicity Lemma" (NPL) in Fan et al. (2006), the main theorem has since been made more specific: the existence of three neighbouring squares in certain well-defined configurations has been shown to cause a breakdown into repetitions of small period. The statement of the NPL includes 14 subcases, with 12 previously proven. This thesis contributes proofs of the two that remain.

Interest has been added to this research by a parallel development over the last dozen years or so: the attempt to specify sharp bounds on the number of maximal periodicities ("runs") that can occur in any string of given length n. Kolpakov and Kucherov (2000) showed that the maximum number of runs—usually denoted $\rho(n)$—is linear in n, and moreover they described a linear-time algorithm to compute all the runs in any given string. But their proof of linearity was nonconstructive: the maximum number of runs was shown to be $\mathcal{O}(n)$ but no constant of proportionality was specified. Subsequent research has shown that $\rho(n)$ is at least $0.9445757n$ (Kusano et al. 2013; Simpson 2010) and asymptotically at most $1.029n$ (Crochemore et al. 2011), or, in other words, more or less the string length n.

What links these two streams of research is a simple observation:

> If the maximum number of runs over all strings of length n is itself approximately n, then on average there will be about one run starting at each position. Thus, if two runs start at some position, there must be some other position, probably nearby, at which no run can start — "probably nearby" because the interference of overlapping squares typically precludes periodic behaviour at one or more positions within the range of the double periodicity. More generally, determining combinatorial constraints on the occurrence of overlapping squares (runs) may lead to a better characterization of $\rho(n)$.

There is a third avenue of research that relates closely to overlapping squares: the computation of all the runs/repetitions in a given string. At present the only way that this can be done involves global data structures (suffix array, longest common prefix array, Lempel-Ziv factorization) that need to be computed in an extended preprocessing phase. This seems uneconomical considering that runs are generally a local phenomenon, and it has been shown (Puglisi and Simpson 2008) that the expected number of runs in a string is much less than the string's length (i.e. runs tend to occur sparsely). Another "local" problem, string searching, has local solutions: there exist many string searching algorithms that only preprocess the query, not the full text to be searched. Thus the current global approach to computing runs might be unnecessary but for the absence of a detailed understanding of the combinatorics of overlapping runs. A local approach would be desirable for space efficiency, particularly as string data, such as biological sequences, gets larger.

In Chapter 2, we review terminology, notation and the relevant background. Then, in Chapter 3, we prove Subcases 3 and 7 of the NPL. We conclude in Chapter 4 with a discussion of future research directions, namely, the general case of three overlapping squares (no two constrained to begin at the same position).

Chapter 2

Background

2.1 Strings

We begin with some basic terminology[1]. A *string* is a finite sequence of symbols (*letters*) drawn from some (possibly infinite) set Σ called the *alphabet*. The alphabet *size* is $\sigma = |\Sigma|$. To reduce notational clutter, we write a string $\boldsymbol{x}$ in mathbold and its length $x = |\boldsymbol{x}|$ in plain math font. We represent a string $\boldsymbol{x}$ as an array $\boldsymbol{x}[1\,..\,x]$ for $x \geqslant 0$. For $x = 0$, $\boldsymbol{x} = \boldsymbol{\varepsilon}$, the *empty string*. If $\boldsymbol{x} = \boldsymbol{uvw}$, then $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ are *substrings* of $\boldsymbol{x}$, and furthermore, $\boldsymbol{u}$ is a *prefix* and $\boldsymbol{w}$ is a *suffix* of $\boldsymbol{x}$. If $\boldsymbol{vw} \neq \boldsymbol{\varepsilon}$, then $\boldsymbol{u}$ is a *proper* prefix. Similarly, if $\boldsymbol{uv} \neq \boldsymbol{\varepsilon}$, $\boldsymbol{w}$ is a proper suffix. If $\boldsymbol{x} = \boldsymbol{uv} = \boldsymbol{wu}$ for $u < x$, then $\boldsymbol{u}$ is a *border* of $\boldsymbol{x}$, that is, a proper prefix that equals a proper suffix. (Note that every nonempty string has an empty border.) If $\boldsymbol{x} = \boldsymbol{uv}$, $0 \leqslant u < x$, then $\boldsymbol{vu}$ is said to be the u^{th} *rotation* of $\boldsymbol{x}$, written $\mathrm{R}_u(\boldsymbol{x})$.

2.2 String data structures

In this section, we briefly describe a few commonly-used string data structures that, as explained in Section 2.4.1, help motiviate our work:

- suffix tree
- suffix array
- longest common prefix array

[1]Our usage generally follows Smyth (2003).

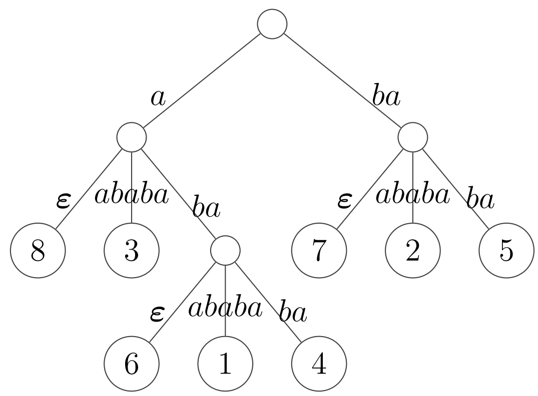

Figure 2.1: $\mathrm{ST}_{\boldsymbol{x}}$ for $\boldsymbol{x} = abaababa$

- LZ factorization

The *suffix tree* is an instance of a more general data structure, the trie. Used to index a set of strings, a trie (also known as a radix tree or prefix tree) is a tree in which the root is associated with the empty string, leaves are associated with strings in the set, and each internal node is associated with the longest common prefix of all the strings associated with its descendents (Fredkin 1960). The suffix tree $\mathrm{ST}_{\boldsymbol{x}}$ of a string $\boldsymbol{x}$ is a trie containing all the suffixes of $\boldsymbol{x}$. The leaves of a suffix tree are labeled with the starting position of the associated suffix, and a node's children are ordered lexicographically. Figure 2.1 shows the suffix tree of $\boldsymbol{x} = abaababa$.

Weiner (1973) introduced the suffix tree, along with a $\mathcal{O}(n \log \sigma)$ time construction algorithm. A few years later, McCreight (1976) presented another $\mathcal{O}(n \log \sigma)$ construction algorithm that is significantly more time- and space-efficient in practice (Giegerich and Kurtz 1997). Nearly twenty years passed before a third algorithm appeared (Ukkonen 1995). Ukkonen's algorithm is on-line and simpler than its predecessors, but in practice marginally slower than McCreight's (Giegerich and Kurtz 1997). Though not practical for large strings, a $\mathcal{O}(n)$ time (independent of σ) algorithm has also been conceived (Farach 1997).

Suffix trees enable many linear-time string algorithms (Gusfield 1997, Chapter 7) and have been memorably lauded for their "myriad virtues" (Apostolico 1985). Nevertheless, large (though linear) space requirements limit their usefulness. A space-efficient implementation might require $10n$ bytes on average, and twice as much in the worst case (Kurtz 1999). Because

$$
\begin{array}{rcccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\boldsymbol{x} = & a & b & a & a & b & a & b & a \\
\mathrm{SA}_{\boldsymbol{x}} = & 8 & 3 & 6 & 1 & 4 & 7 & 2 & 5 \\
\mathrm{LCP}_{\boldsymbol{x}} = & 0 & 1 & 1 & 3 & 3 & 0 & 2 & 2
\end{array}
$$

Figure 2.2: $\mathrm{SA}_{\boldsymbol{x}}$ and $\mathrm{LCP}_{\boldsymbol{x}}$ for $\boldsymbol{x} = abaababa$

of this significant drawback, the suffix tree has been mostly superceded by the more economical suffix array (Abouelhoda et al. 2004).

The *suffix array* $\mathrm{SA}_{\boldsymbol{x}}$ of a string $\boldsymbol{x}$ is a sorted list of the suffixes of $\boldsymbol{x}$, succintly encoded: for $i, j \in [1 \,..\, x]$, $\mathrm{SA}_{\boldsymbol{x}}[j] = i$, where i is the starting position of the j^{th} suffix of $\boldsymbol{x}$ in lexicographic order (Manber and Myers 1993). Figure 2.2 shows the suffix array of $\boldsymbol{x} = abaababa$. (Comparing Figures 2.2 and 2.1, note that performing a depth-first traversal of a suffix tree while printing leaf node labels produces the corresponding suffix array.) Whereas tree traversal is used to query suffix trees, suffix arrays can be queried by binary search. The original suffix array construction algorithms were $\mathcal{O}\big(n \log n\big)$-time, but linear algorithms were later developed (Kärkkäinen and Sanders 2003; Kim et al. 2003; Ko and Aluru 2003; Nong et al. 2009). However, according to a survey of suffix array construction algorithms (Puglisi, Smyth, et al. 2007), the fastest algorithm in practice (Maniscalco and Puglisi 2006) actually has worst-case $\mathcal{O}\big(n^2 \log n\big)$ time complexity. The most space-efficient algorithms in practice (Maniscalco and Puglisi 2008, 2006; Manzini and Ferragina 2004) use as little as $5n$ bytes on average for a string of length n.

The suffix array is enhanced by other data structures, most notably the *longest common prefix* array, in which $\mathrm{LCP}_{\boldsymbol{x}}[1] = 0$ and for $j \in [2 \,..\, x]$, $\mathrm{LCP}_{\boldsymbol{x}}[j]$ is the length of the longest prefix between the j^{th} and $(j-1)^{\text{th}}$ suffixes in $\mathrm{SA}_{\boldsymbol{x}}$. Figure 2.2 shows an example LCP array.

The Lempel-Ziv (LZ) factorization (Lempel and Ziv 1976) partitions a string into substrings: $\boldsymbol{x} = \boldsymbol{w_1 w_2} \cdots \boldsymbol{w_k}$, where for all $j \in [1 \,..\, k]$, $\boldsymbol{w}_j$ is

1. a single character that does not occur in $\boldsymbol{w_1 w_2} \cdots \boldsymbol{w_{j-1}}$; or

2. the longest substring that occurs twice in $\boldsymbol{w_1 w_2} \cdots \boldsymbol{w_j}$.

In case 2, the first of the two occurrences may overlap with $\boldsymbol{w_j}$. Note that $\boldsymbol{w_1} = \boldsymbol{x}[1]$. As an example, the LZ factorization of $\boldsymbol{x} = abaababa$ is $\boldsymbol{x} = (a)(b)(a)(aba)(ba)$. Al-Hafeedh et al. (2012) compares the many LZ factorization algorithms, some of which are linear-time, and which together offer various tradeoffs between time and space. All LZ factorization algorithms use an ST, SA, LCP, or other global data structures.

2.3 Periodicity

If $\boldsymbol{x}[i] = \boldsymbol{x}[i+p]$ for all $i \in [1..x-p]$, then $\boldsymbol{x}$ has *period* p. Every period of a string corresponds to a border:

Lemma 1 (Lothaire 2002, Section 8.1.1)**.** *If $\boldsymbol{v}$ is a border of $\boldsymbol{w}$, then $\boldsymbol{w}$ has period $w-v$. Conversely, if w has period p, then it has border $\boldsymbol{w}[1..w-p]$.*

For example, the string

$$\begin{array}{cccccccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \boldsymbol{x} & = & a & b & a & a & b & a & b & a & a & b \end{array} \tag{2.1}$$

has borders *abaab* and *ab*, hence corresponding periods 5 and 8, respectively.

The analysis of periodicity often involves strings of more than one period, or periodic strings that overlap. The next few lemmas express some possible consequences of coincident periodicities. The first follows readily from Lemma 1.

Lemma 2 (Lothaire 2002, Lemma 8.1.1)**.** *If $\boldsymbol{x}$ has periods p and q such that $q < p \leqslant x$, then the border of $\boldsymbol{x}$ of length $x-q$ has period $p-q$.*

Another basic lemma applies to strings of one period with a substring of another period:

Lemma 3 (Lothaire 2002, Lemma 8.1.3)**.** *If $\boldsymbol{x}$ has period p and there exists a substring $\boldsymbol{u}$ of $\boldsymbol{x}$ with $p \leqslant u$ that has period q, where q divides p, then $\boldsymbol{x}$ has period q.*

The next lemma, known as the Periodicty Lemma, is one of the most important in combinatorics on words, featuring in many correctness proofs of string algorithms. For strings of two periods p and q, the Periodicity Lemma provides the minimum length for which all strings of at least that length also have period $\gcd(p,q)$.

Lemma 4 (Periodicity Lemma (Fine and Wilf 1965; Lothaire 2005))**.** *If $\boldsymbol{x}$ has periods p and q, and $p+q \leqslant x+\gcd(p,q)$, then $\boldsymbol{x}$ also has period $\gcd(p,q)$.*

For example, the string

$$\begin{array}{cccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ \boldsymbol{x} = & a & b & a & a & b & a & a & b & a & a & b & a & a \end{array}$$

has length $n = 13$, and periods $p = 6$ and $q = 9$. Since $d = \gcd(p, q) = 3$ and $p + q = 15 < n + d = 16$, the Periodicity Lemma allows us to infer that the string also has period $d = 3$.

In practice, the Periodicity Lemma is often applied via one of the following two corollaries. The first applies to overlapping strings of the same period, the second to overlapping strings of different periods.

Lemma 5 (Lothaire 2002, Lemma 8.1.2)**.** *If $\boldsymbol{x} = \boldsymbol{uvw}$, and $\boldsymbol{uv}$ and $\boldsymbol{vw}$ have period $p \leqslant v$, then $\boldsymbol{x}$ has period p.*

Lemma 6 (Simpson 2007, Section 1)**.** *If $\boldsymbol{x} = \boldsymbol{uvw}$, where $\boldsymbol{uv}$ has period p, $\boldsymbol{vw}$ has period q, and $p + q \leqslant v + \gcd(p, q)$, then $\boldsymbol{x}$ has period $\gcd(p, q)$.*

The Periodicity Lemma has been generalized to three periods (Castelli et al. 1999), an arbitrary number of periods (Constantinescu and Ilie 2005; Holub 2006; Justin 2000; Tijdeman and Zamboni 2009), multiple dimensions (Simpson and Tijdeman 2003), and the case in which the length of a string with two periods does not satisfy the bound given by the Periodicity Lemma (Fraenkel and Simpson 2005). In particular, the generalization of the Periodicity Lemma to three periods uses the function

$$f(p_1, p_2, p_3) = \frac{1}{2}\left[p_1 + p_2 + p_3 - 2\gcd(p_1, p_2, p_3) + h(p_1, p_2, p_3)\right] \qquad (2.2)$$

where h is a function derived from Euclid's algorithm for computing the greatest common divisor of three integers.

Lemma 7 (Periodicity Lemma for Three Periods (Castelli et al. 1999))**.** *If a string $\boldsymbol{x}$ has periods p_1, p_2 and p_3, with $p_1 \leqslant p_2 \leqslant p_3$ and $f(p_1, p_2, p_3) \leqslant x$ (f as defined by 2.2), then $\boldsymbol{x}$ also has period $\gcd(p_1, p_2, p_3)$.*

As we will see, the New Periodicity Lemma also deals with three periods imposed by three "overlapping squares". In a sense, Lemma 7 is more general than the New Periodicity Lemma because it only requires three periods rather than squares. On the other hand, the New Periodicity Lemma provides more information and covers a wider range of cases.

2.4 Repetitions and runs

If $\boldsymbol{x} = \boldsymbol{v}\boldsymbol{u}^e\boldsymbol{w}$, where $e \geqslant 2$ is an integer, and $\boldsymbol{u}$ is neither a suffix of $\boldsymbol{v}$ nor a prefix of $\boldsymbol{w}$ (e is maximum), then $\boldsymbol{u}^e$ is said to be a *repetition* in $\boldsymbol{x}$. The integers u and e are the *period* and *exponent*[2], respectively, of the repetition. The string (2.1) has repetitions $(aba)^2, (abaab)^2, a^2, (ab)^2, (ba)^2$, each of which is a *square*. In general, every repetition has a square prefix. We say that a square $\boldsymbol{u}^2$ is *irreducible*[3] if $\boldsymbol{u}$ is not itself a repetition, *regular* if $\boldsymbol{u}$ has no square prefix, and *minimal* if no proper prefix of $\boldsymbol{u}^2$ is a square. Note that minimality implies regularity, which in turn implies irreducibility.

If $\boldsymbol{v} = \boldsymbol{x}[i\,..\,j]$ has period u, where $v/u \geqslant 2$, and if neither $\boldsymbol{x}[i-1\,..\,j]$ nor $\boldsymbol{x}[i\,..\,j+1]$ (whenever these are defined) has period u, then $\boldsymbol{v}$ is said to be a *maximal periodicity* or *run* in $\boldsymbol{x}$ (Main 1989) with a (now fractional) *exponent* $e = v/u$. All of the repetitions in (2.1) are runs except for $(ab)^2$ and $(ba)^2$: these are substrings of the run $\boldsymbol{v} = ababa = (ab)^{5/2}$. In general, every repetition is a substring of some run; thus computing all the runs implicitly computes all the repetitions.

From Lemmas 1 and 4, it follows that if a string $\boldsymbol{x}$ equals its u^{th} rotation, then it is a repetition of period $\gcd(u, x)$.

Lemma 8 (Smyth 2003, Theorem 1.4.2)**.** *For any string $\boldsymbol{x}$, $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{v} = \boldsymbol{v}\boldsymbol{u}$ if and only if $\boldsymbol{x}$ is a repetition of period* $\gcd(u, v) = \gcd(u, x-u) = \gcd(u, x)$.

The following lemma, used in Chapter 3, also connects repetitions and rotations.

Lemma 9 (Kopylova and Smyth 2012, Lemma 8)**.** *Suppose both $\boldsymbol{x}$ and $\mathrm{R}_v(\boldsymbol{x})$, $0 < v < x$, have period u. Let $\ell = x \mod u > 0$, $r = \left\lfloor \frac{x}{u} \right\rfloor$, and $d = \gcd(u, \ell)$. Then:*

(a) *if $r = 1$ and $v \geqslant \ell$, $\mathrm{R}_{v-\ell}(\boldsymbol{x})[1\,..\,2\ell]$ is a square of period ℓ;*

(b) *if $r = 1$ and $v \leqslant \ell$, $\boldsymbol{x}[1\,..\,v+\ell]$ has period ℓ;*

(c) *if $r > 1$ and $v < u$, $\boldsymbol{x}[1\,..\,v+\ell]$ has period ℓ; if moreover $v + d \geqslant u$, then $\boldsymbol{x}$ is a repetition of period d;*

(d) *if $r > 1$ and $u \leqslant v \leqslant x-u$, $\boldsymbol{x}[1\,..\,u+\ell]$, hence $\boldsymbol{x}$, is a repetition of period d;*

[2]We use $\boldsymbol{u}^{(h)}$ (with the exponent in parentheses) to denote the h^{th} occurrence of $\boldsymbol{u}$.
[3]Others use the term *primitively rooted*.

(e) *if* $r > 1$ *and* $x - u < v$, *where* $v' = v - (x - u)$, $\boldsymbol{x}[v' + 1\,..\,u + \ell]$ *has period* ℓ*; if moreover* $v' \leqslant d$, *then* $\boldsymbol{x}$ *is a repetition of period* d.

2.4.1 Algorithms for finding repetitions and runs

Three classical algorithms for computing all the repetitions in a string were proposed in the early 1980s:

1. Crochemore (1981)

2. Apostolico and Preparata (1983)

3. Main and Lorentz (1984)

Smyth (2003, page 340) points out that Crochemore's algorithm basically constructs a suffix tree. The Apostolico algorithm uses a suffix tree explicitly. Main's algorithm uses LZ factorization. All of these algorithms execute in $\mathcal{O}(n \log n)$ time, asymptotically optimal since the *Fibonacci string* $\boldsymbol{f_k}$, defined by

$$\boldsymbol{f_k} = \begin{cases} b, & k = 0 \\ a, & k = 1 \\ \boldsymbol{f_{k-1}} \boldsymbol{f_{k-2}}, & k \geqslant 2 \end{cases}$$

contains $\mathcal{O}(f_k \log f_k)$ squares (Crochemore 1981, Lemma 10; Fraenkel and Simpson 1999; Iliopoulos et al. 1997). To obtain a lower time complexity, Main (1989) introduced a new encoding of repetitions, the "maximal periodicity" or run, and described an LZ-factorization-based algorithm to compute all the "leftmost" runs in $\mathcal{O}(n)$ time. This was extended in Kolpakov and Kucherov (2000) to compute all runs $\mathcal{O}(n)$ time.

All of these algorithms for computing repetitions and runs use a suffix tree or LZ factorization, both global data structures with significant memory requirements. More efficient algorithms for computing runs have been proposed — for example, Chen et al. (2007) — but still with extensive preprocessing and the same general approach. (For a survey of algorithms for computing repetitions and runs, see Kopylov (2010, Chapter 3).)

While asymptotically optimal, the heavy-handed global approach seems inappropriate because runs are local and often sparsely occurring. Puglisi and Simpson (2008) derives a formula for the expected number of runs in a string as a function of its length n and alphabet size σ. This value is largest for binary alphabets, for which the expected number of runs per unit length is 0.41. This value decreases with increasing σ: 0.24 for DNA strings ($\sigma = 4$), 0.04 for protein ($\sigma = 24$), and 0.01 for English-language text. Experiments confirm that these values are good predictions of the average number of runs in real-world data.

A future runs algorithm might dispense with global data structures, instead taking advantage of some combinatorial properties of runs to detect them during a single left-to-right scan.

2.4.2 The combinatorics of runs

The linearity of the Kolpakov and Kucherov (2000) algorithm was established by a complex proof that the maximum number $\rho(n)$ of runs (irreducible and not extendible to the left or right) in any string of length n satisfies

$$\rho(n) \leqslant K_1 n - K_2 \sqrt{n} \log_2 n \tag{2.3}$$

for some universal positive constants K_1 and K_2. The method of proof allowed no bounds to be placed on K_1 and K_2, but based on computational evidence up to $n = 60$, it was conjectured that $\rho(n) < n$ (for large n). This is known as the "Runs Conjecture". Over the last decade, the bounding of $\rho(n)/n$ has become a growth industry. Rytter (2006) provided the first upper bound, showing that $\rho(n)/n < 5$. The bound improved successively, first to 3.9 (Rytter 2007), then to 3.48 (Puglisi, Simpson, and Smyth 2008), 1.6 (Crochemore and Ilie 2008), 1.52 (Giraud 2009, 2008), 1.048 (Crochemore et al. 2008), and most recently to 1.029 (Crochemore et al. 2011), the last result achieved using three years of CPU time on a supercomputer. The lower bound has progressed from 0.927 (Franek, Simpson, et al. 2003), to 0.944542 (Matsubara et al. 2008), to 0.944575 (Kusano et al. 2013; Simpson 2010). So, for sufficiently large n, we know that

$$0.944575 < \rho(n)/n \leqslant 1.029.$$

This work has confined the possible values of $\rho(n)/n$ to a narrow range, but it seems not to have yielded the combinatorial knowledge needed for a new runs algorithm. Another approach, the subject of this thesis, has sought to find a combinatorial basis for estimating the maximum number of runs in a string by considering the consequences of three overlapping squares.

2.5 Three overlapping squares

The Runs Conjecture provides a simple motivation for studying overlapping squares (Fan et al. 2006). If $\rho(n) \leqslant n$, then the average number of runs starting at each of the n positions is at most one. When two runs occur at the same position, there must be another position at which no run occurs. Perhaps a proof of the Runs Conjecture is to be found in a detailed understanding of how two overlapping runs combine to make a third run impossible. Runs always have a square prefix, so analyzing overlapping squares seems a promising way to proceed[4]. It seems plausible that a position at which a third run is precluded lies within the range of the overlapping square prefixes. If this is true, a thorough understanding of the restrictions imposed by overlapping squares could lead directly to a proof of the Runs Conjecture, as well as a new paradigm for the computation of runs.

Recent research on overlapping squares extends the earlier "Three Squares Lemma" (Crochemore and Rytter 1995) that shows if three squares occur at the same position in a string, one of them must be long:

Lemma 10 (Three Squares Lemma (Crochemore and Rytter 1995))**.** *Suppose $\boldsymbol{u}$ is irreducible, and suppose $\boldsymbol{v} \neq \boldsymbol{u}^j$ for any $j \geqslant 1$. If $\boldsymbol{u}^2$ is a prefix of $\boldsymbol{v}^2$, in turn a proper prefix of $\boldsymbol{w}^2$, then $w \geqslant u + v$.*

The Fibonacci string has three square prefixes of lengths 6, 10, and $10 + 6 = 16$, respectively, showing the Three Squares Lemma is best possible:

$$\begin{array}{rcccccccccccccccc} & {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle 3} & {\scriptstyle 4} & {\scriptstyle 5} & {\scriptstyle 6} & {\scriptstyle 7} & {\scriptstyle 8} & {\scriptstyle 9} & {\scriptstyle 10} & {\scriptstyle 11} & {\scriptstyle 12} & {\scriptstyle 13} & {\scriptstyle 14} & {\scriptstyle 15} & {\scriptstyle 16} \\ \boldsymbol{f} = & a & b & a & a & b & \underline{a} & b & a & a & \underline{b} & a & a & b & a & b & \underline{a} \end{array}$$

The Three Squares Lemma has been generalized to the case with two of the squares occurring at the same position in a string and the third nearby, somewhat to the right, producing a result called the "New Periodicity Lemma".

2.5.1 The New Periodicity Lemma

Here is the original NPL, as stated and proved in Fan et al. (2006):

Lemma 11. *If $\boldsymbol{x}$ has regular prefix $\boldsymbol{u}^2$ and irreducible prefix $\boldsymbol{v}^2$, $u < v < 2u$, then for every $k \in 0..v-u-1$ and every $w \in v-u+1..v-1$, $w \neq u$, $\boldsymbol{x}[k+1..k+2w]$ is not a square.*

[4] Though not because the maximum number of irreducible squares in a string of length n might be n; as mentioned in Section 2.4.1, the optimal bound is known to be $\mathcal{O}(n \log n)$.

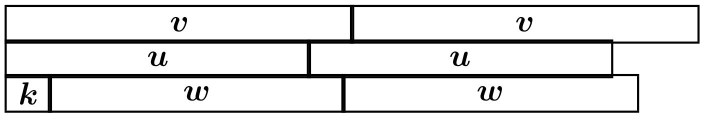

Figure 2.3: Three overlapping squares, as postulated in Lemma 13.

The proof required consideration of 14 subcases based on the magnitudes of k and w (see Figure 2.4), each of which led to a proof by contradiction of the regularity of $\boldsymbol{u}$. Subsequent work has split the range $u < v < 2u$ into two sections $(u, 3u/2]$ and $(3u/2, 2u)$, while eliminating the regularity condition altogether, as we now describe.

In Kopylova and Smyth (2012) it was shown that for $u < v \leqslant 3u/2$, the requirement that $\boldsymbol{x} = \boldsymbol{v}^2$ with prefix $\boldsymbol{u}^2$ necessitates

$$\boldsymbol{x} = \boldsymbol{t_1}^m \boldsymbol{t_2} \boldsymbol{t_1}^{m+1} \boldsymbol{t_2} \boldsymbol{t_1}, \tag{2.4}$$

where $t_1 = v - u,\ t_2 = u \bmod t_1,\ m = \lfloor u/t_1 \rfloor \geqslant 2$ and $\boldsymbol{t_2}$ is a proper prefix of $\boldsymbol{t_1}$. It was shown further that, except for $m + 5$ precisely identified runs that always occur in $\boldsymbol{x}$, there could be no other runs of period greater than t_1. Thus for $u < v \leqslant 3u/2$, the structure of $\boldsymbol{x}$ is well defined, even without reference to $\boldsymbol{w}$.

On the other hand, for $3u/2 < v < 2u$, there is a different breakdown:

Lemma 12 (Fan et al. 2006, Lemma 9)**.** *If $\boldsymbol{x} = \boldsymbol{v}^2$ has prefix $\boldsymbol{u}^2$, then*

$$\boldsymbol{x} = (\boldsymbol{u_1} \boldsymbol{u_2} \boldsymbol{u_1} \boldsymbol{u_1} \boldsymbol{u_2})^2, \tag{2.5}$$

where $u_1 = 2u - v > 0$ and $u_2 = 2v - 3u > 0$, if and only if $\frac{3u}{2} < v < 2u$. Observe that $\boldsymbol{u} = \boldsymbol{u_1}\boldsymbol{u_2}\boldsymbol{u_1}$ and $\boldsymbol{v} = \boldsymbol{u_1}\boldsymbol{u_2}\boldsymbol{u_1}\boldsymbol{u_1}\boldsymbol{u_2}$.

Note that setting $\boldsymbol{t_1} = \boldsymbol{u_1}\boldsymbol{u_2},\ \boldsymbol{t_2} = \boldsymbol{u_1}$ converts the form (2.5) into (2.4), but with $m < 2$. For this case, Kopylova and Smyth (2012) provided, with the assistance of a computer program, conjectures for the breakdown of $\boldsymbol{x}$ in each of the 14 subcases. In half of the subcases (1, 2, 5, 6, and 8–10), $\boldsymbol{x}$ was conjectured (and, in the same paper, proved) to always be a repetition of period d. In the other cases, $\boldsymbol{x}$ was conjectured to have a different but still highly repetitive structure. An earlier paper (Simpson 2007) had already provided proofs for subcases 5, 6, and 10, as well as results for subcases 11–14 that were later refined in Franek, Fuller, et al. (2012). The latter also proved subcase 4, leaving only two of the 14 subcases unconfirmed. In this thesis, we prove the remaining two, subcases 3 and 7. Thus after much experimental and theoretical work, the revised NPL can be stated as follows:

Subcase S	k	$k+w$	$k+2w$	Special Conditions
1	$0 \leqslant k \leqslant u_1$	$k+w \leqslant u$	$k+2w \leqslant u+u_1$	$k \geqslant u_2$
2	$0 \leqslant k \leqslant u_1$	$k+w \leqslant u$	$k+2w \leqslant u+u_1$	$k < u_2$
3	$0 \leqslant k \leqslant u_1$	$k+w \leqslant u$	$k+2w > u+u_1$	—
4	$0 \leqslant k \leqslant u_1$	$u < k+w \leqslant u+u_1$	—	—
5	$0 \leqslant k \leqslant u_1$	$u+u_1 < k+w \leqslant v$	—	—
6	$0 \leqslant k \leqslant u_1$	$v < k+w < 2u$	—	—
7	$u_1 < k < u_1+u_2$	$k+w \leqslant u+u_1$	$k+2w \leqslant 2u$	—
8	$u_1 < k < u_1+u_2$	$k+w \leqslant u+u_1$	$k+2w > 2u$	—
9	$u_1 < k < u_1+u_2$	$u+u_1 < k+w \leqslant v$	—	$w < u$
10	$u_1 < k < u_1+u_2$	$k+w \leqslant v$	$k+2w \leqslant u+v$	$w > u$
11	$u_1 < k < u_1+u_2$	$k+w \leqslant v$	$u+v < k+2w \leqslant 2v-u_2$	—
12	$u_1 < k < u_1+u_2$	$k+w \leqslant v$	$2v-u_2 < k+2w$	—
13	$u_1 < k < u_1+u_2$	$v < k+w \leqslant 2u$	—	—
14	$u_1 < k < u_1+u_2$	$2u < k+w < 2u+u_2-1$	—	—

Figure 2.4: The 14 subcases identified in Fan et al. (2006), slightly modified, for three neighbouring squares $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ (with $v-u < w < v, w \neq u, 0 \leqslant k < v-u$).

Subcases S	Conditions	Breakdown of $\boldsymbol{x}$
$1, 2, 5, 6, 8$–10	$(\forall \boldsymbol{x}, \sigma = d)$	$\boldsymbol{x} = \boldsymbol{d}^{x/d}$
$3, 4, 7$	$(\forall \boldsymbol{x})$ specified cases	$\boldsymbol{x} = \boldsymbol{d_1}^{u/d_1}\boldsymbol{d_1}^{v/d_1}\boldsymbol{d_1}^{(v-u)/d_1}$ $\boldsymbol{x} = \boldsymbol{d}^{x/d}$
11–14	$\sigma = d$ or $d_2 \leqslant 2u - v$ otherwise	$\boldsymbol{x} = \boldsymbol{d}^{x/d}$ $\boldsymbol{x} = \big((\boldsymbol{d_3}^{d_2/d_3})^{v/d_2}\big)^2$

Figure 2.5: Structure of $\boldsymbol{x}$ for subcases $S \in 1..14$: σ is the largest alphabet size consistent with u, v, k, w (Franek, Fuller, et al. 2012); $\boldsymbol{d}$, $\boldsymbol{d_1}$ and $\boldsymbol{d_3}$ are prefixes of $\boldsymbol{x}$ with $d = \gcd(u, v, w)$, $d_1 = \gcd(u-w, v-u)$, $d_2 = \gcd(u, v-w)$, $d_3 = v \bmod d_2$.

Lemma 13. *Suppose that a string $\boldsymbol{x}$ has prefixes $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$, $3u/2 < v < 2u$, and suppose further that a third square $\boldsymbol{w}^2$ occurs at position $k + 1$ of $\boldsymbol{x}$, where $v - u < w < v$, $w \neq u$, and $0 \leqslant k < v - u$. Then for each of the 14 subcases S identified in Figure 2.4, the corresponding structure of $\boldsymbol{x}$ is given in Figure 2.5.*

In other words, $\boldsymbol{x}$ breaks down into repetitions of small period — essentially, the postulate of three such squares cannot be satisfied.

2.5.2 The general case characterized

The proof of the New Periodicity Lemma now complete, we believe that further generalization is of interest: what happens when the three squares $\boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{w}^2$ are merely constrained to be "neighbouring", without the requirement that $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ occur at the same position? What is an appropriate formulation of such a problem? What relative values of k, u, v, w are of combinatorial interest?

The following lemma, to appear in a forthcoming paper (Bland and Smyth 2014), begins to address these questions. It states the consequences of a square $\boldsymbol{u}^2$ beginning at some position i in a string and overlapping with a second square $\boldsymbol{v}^2$ at position $i + k$, $k \geqslant 0$, to its right.

Lemma 14. *Suppose $\boldsymbol{x}$ has prefixes $\boldsymbol{u}^2$ and $\boldsymbol{k}\boldsymbol{v}^2$, $k \geqslant 0$, where $x = \max(2u, k+2v)$, $k \leqslant u < 2v$.*

(a) $k+v<u<2v\ (k<\min(v-1,u-v))$:

$$\boldsymbol{x}=(\boldsymbol{p}^e\boldsymbol{z})^2=\boldsymbol{p}^e\boldsymbol{q}^f\boldsymbol{q}^{f-e}=\boldsymbol{p}^e\boldsymbol{q}^f\boldsymbol{p}[k+1\,..\,u-v],$$

where $\boldsymbol{p}=\boldsymbol{u}[1\,..\,u-v]$, $e=\frac{k+v}{u-v}>1$, $\boldsymbol{z}=\boldsymbol{v}[1\,..\,u-(k+v)]$, $\boldsymbol{q}=R_k(\boldsymbol{p})$, $f=\frac{u}{u-v}>2$, $f-e\leqslant 1$.

(b) $\frac{k}{2}+v\leqslant u\leqslant k+v\ (1\leqslant u-v\leqslant k\leqslant 2(u-v))$:

$$\boldsymbol{x}=(\boldsymbol{z}\boldsymbol{p}^e)^2=(\boldsymbol{q}[1\,..\,k+v-u]\boldsymbol{p}^e)^2=(\boldsymbol{k}\boldsymbol{p}^{e-1})^2,$$

where $\boldsymbol{z}=\boldsymbol{u}[1\,..\,k+v-u]$, $\boldsymbol{p}=\boldsymbol{v}[1\,..\,u-v]$, $e=1+\frac{u-k}{u-v}\geqslant 1$, $\boldsymbol{q}=R_c(\boldsymbol{p})$, $c=(u-k)\bmod(u-v)$.

(c) $v<u<\frac{k}{2}+v\ (k>2(u-v))$:

$$\boldsymbol{x}=(\boldsymbol{q}\boldsymbol{y}\boldsymbol{p}^e)^2\boldsymbol{y},$$

where $\boldsymbol{p}=\boldsymbol{v}[1\,..\,u-v]$, $e=1+\frac{u-k}{u-v}>1$, $\boldsymbol{q}=R_c(\boldsymbol{p})$, $c=(u-k)\bmod(u-v)$, $\boldsymbol{y}=\boldsymbol{v}[2u-(k+v)+1\,..\,v]$. *Moreover, both* $\boldsymbol{x}$ *and* $\boldsymbol{kv}$ *have border* $\boldsymbol{qy}$*.*

(d) $\frac{2(k+v)}{3}\leqslant u<v\ (k\leqslant\frac{3u}{2}-v<\frac{v}{2})$:

$$\boldsymbol{x}=(\boldsymbol{k}\boldsymbol{p}^e)^2\boldsymbol{q}\boldsymbol{k}\boldsymbol{p},$$

where $\boldsymbol{p}=\boldsymbol{v}[1\,..\,v-u]$, $e=\frac{u-k}{v-u}>1$, $\boldsymbol{q}=R_c(\boldsymbol{p})$, $c=(u-k)\bmod(v-u)$. *Both* $\boldsymbol{x}$ *and* $\boldsymbol{kv}$ *have border* $\boldsymbol{kp}$*.*

(e) $\frac{k+v}{2}<u<\frac{2(k+v)}{3}<v\ (\frac{3u-2v}{2}<k<2u-v<u)$:

$$\boldsymbol{x}=\boldsymbol{k}(\boldsymbol{p}^e\boldsymbol{k}\boldsymbol{p})^2,$$

where $\boldsymbol{p}=\boldsymbol{v}[1\,..\,v-u]$, $e=\frac{u-k}{v-u}>1$.

(f) $k\leqslant u\leqslant\frac{k+v}{2}$ ($\boldsymbol{u}^2$ *a prefix of* $\boldsymbol{kv}$) :

$$\boldsymbol{x}=\boldsymbol{k}(\boldsymbol{p}^e\boldsymbol{z})^2,$$

where $\boldsymbol{p}=\boldsymbol{u}[k+1\,..\,u]\boldsymbol{u}[1\,..\,k]$, $e=\frac{2u-k}{u}\geqslant 1$, $\boldsymbol{z}=\boldsymbol{v}[2u-k+1\,..\,v]$.

Because every instance of three overlapping squares can be seen as two pairs of two overlapping squares, this lemma can characterize three overlapping squares in any configuration. We first use this lemma in Chapter 3 to prove the two remaining subcases of the NPL, then in Chapter 4 we discuss its application to the general case of three overlapping squares.

Chapter 3

Completing the New Periodicity Lemma

In this chapter[1], we prove the two remaining subcases of Lemma 13.

3.1 Subcase 3

We first deal with the general case valid for all occurrences of Subcase 3, then go on to identify circumstances in which $\boldsymbol{x}$ is constrained to be a repetition of small period $d = \gcd(u, v, w)$.

Lemma 15 (Subcase 3)**.** *Suppose that a string $\boldsymbol{x}$ has prefixes $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$, $3u/2 < v < 2u$, and suppose further that a third square $\boldsymbol{w}^2$, $w \neq u$, occurs at position $k+1$ of $\boldsymbol{x}$, where*

$$0 \leqslant k \leqslant u_1 < u_1 + u_2 < w < v \tag{3.1}$$

$$k + w \leqslant u \tag{3.2}$$

$$k + 2w > u + u_1 \tag{3.3}$$

and $u_1 = 2u - v$ and $u_2 = 2v - 3u$. Then $\boldsymbol{x} = \boldsymbol{d_1}^{u/d_1}\boldsymbol{d_1}^{v/d_1}\boldsymbol{d_1}^{(v-u)/d_1}$, where $d_1 = \gcd(u - w, v - u)$.

[1]This chapter is to appear in a forthcoming paper (Bland and Smyth 2014).

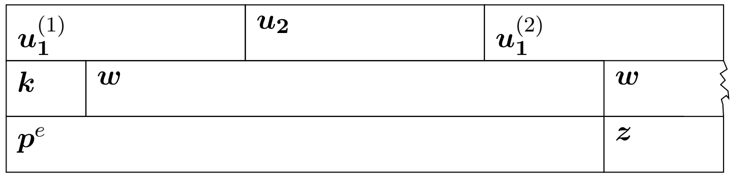

Figure 3.1: String $\boldsymbol{u}$ in Subcase 3

Proof. By Lemma 12, the overlap of $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ forces $\boldsymbol{x} = (\boldsymbol{u_1 u_2 u_1 u_1 u_2})^2$, with $\boldsymbol{u} = \boldsymbol{u_1 u_2 u_1}$. By Lemma 14(a), $\boldsymbol{u} = \boldsymbol{p}^e \boldsymbol{z}$, where $\boldsymbol{z} = \boldsymbol{w}[1\,..\,u-(k+w)]$, $\boldsymbol{p} = \boldsymbol{u}[1\,..\,u-w]$ and $e = \frac{k+w}{u-w} > 1$.

We first show that if $\boldsymbol{u}$ has period $p = u-w$, the lemma holds. Note that $\boldsymbol{u}$ has period $u_1 + u_2$ and

$$u_1 + u_2 + p = u + u_1 + u_2 - w < u$$

since $u_1 + u_2 < w$ from (3.1). Therefore, assuming $\boldsymbol{u}$ has period p, $\boldsymbol{u} = \boldsymbol{x}[1\,..\,u]$ has period $d_1 = \gcd(p, u_1+u_2)$ by Lemma 4. It follows that $\boldsymbol{u_1 u_2} = \boldsymbol{x}[u+v+1\,..\,x]$, a prefix of $\boldsymbol{u} = \boldsymbol{u_1 u_2 u_1}$, has period d_1 as well. Finally, $\boldsymbol{x}[u+1\,..\,u+v] = \boldsymbol{u_1 u_2 u_1 u_2 u_1}$ has period $u_1 + u_2$ and prefix $\boldsymbol{u}$ of length $u > u_1 + u_2$ with period d_1. Since $d_1 = \gcd(u-w, u_1+u_2)$ divides $u_1 + u_2$, $\boldsymbol{x}[u+1\,..\,u+v]$ has period d_1 by Lemma 3. Thus the lemma holds assuming $\boldsymbol{u}$ has period p.

Note first from (3.1) and (3.2) that $u_1 u_2 < w < u$, hence that $u_1 - p = u_1 - (u-w) > 0$. Then to see that $\boldsymbol{u}$ in fact has period p, consider two cases:

$$u_1 \geqslant k + w - (u_1 + u_2) \geqslant p \tag{3.4}$$

and

$$k + w - (u_1 + u_2) < p < u_1. \tag{3.5}$$

In the first case, the prefix $\boldsymbol{kw} = \boldsymbol{p}^e$ of $\boldsymbol{u}$ extends at least p positions into the suffix $\boldsymbol{u_1}^{(2)}$. Since $\boldsymbol{u_1}$ is a prefix of $\boldsymbol{kw}$, $\boldsymbol{u_1}$ has period p, and therefore $\boldsymbol{u}$ has a prefix and suffix of period p which overlap by at least p. Consequently, by Lemma 6, $\boldsymbol{u}$ has period p.

The second case (3.5) is more complicated (see Figure 3.2). Both $\boldsymbol{p}$ and $\boldsymbol{u_1}$ are prefixes of $\boldsymbol{u}$, so $\boldsymbol{p}$ is a proper prefix of $\boldsymbol{u_1}$. Both $\boldsymbol{u_1}$ and $\boldsymbol{z}$ are suffixes of $\boldsymbol{u}$, and

$$z = u - k - w \leqslant p < u_1,$$

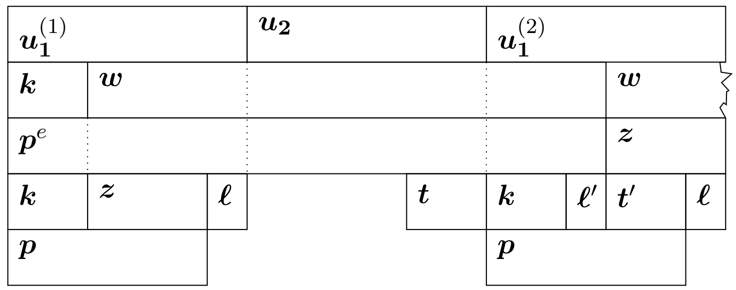

Figure 3.2: String $\boldsymbol{u}$ in Subcase 3 when (3.5) holds

so $\boldsymbol{z}$ is a proper suffix of $\boldsymbol{u_1}$. The prefix $\boldsymbol{p}$ and the suffix $\boldsymbol{z}$ of $\boldsymbol{u_1}$ must overlap because (3.5) is equivalent to $u_1 < p + z$. Noting that $p = u - w = k + z$ so that $\boldsymbol{p} = \boldsymbol{kz}$, we have (Figure 3.2)

$$\boldsymbol{z} = \boldsymbol{\ell' t'} = \boldsymbol{t' \ell} \tag{3.6}$$

and

$$\boldsymbol{u_1} = \boldsymbol{p\ell} = \boldsymbol{kz\ell} = \boldsymbol{k\ell' z} = \boldsymbol{k\ell' t' \ell}, \tag{3.7}$$

where $\boldsymbol{\ell}$ and $\boldsymbol{\ell'}$ are respectively the proper suffix and proper prefix of $\boldsymbol{z}$ of length $\ell = \ell' = u_1 - p$, and $\boldsymbol{t'}$ is the border of $\boldsymbol{z}$ of length $t' = z - \ell$. Since by (3.6) $\boldsymbol{z}$ has border $\boldsymbol{t'}$, it therefore has period ℓ, as does $\boldsymbol{\ell' z \ell} = \boldsymbol{\ell' \ell' t' \ell} = \boldsymbol{\ell' t' \ell \ell}$. Since $\boldsymbol{u_1}$ has period p, $\boldsymbol{p}$ has prefix $\boldsymbol{\ell}$; $\boldsymbol{p} = \boldsymbol{kz}$ also has suffix $\boldsymbol{\ell}$, so that $\boldsymbol{p}$ has border $\boldsymbol{\ell}$. Observe also that because $k + w - (u_1 + u_2) = k + \ell$, $\boldsymbol{k\ell'}$ is the prefix of $\boldsymbol{u_1}$ that overlaps $\boldsymbol{w}$.

Let $\boldsymbol{t}$ be the suffix of $\boldsymbol{u_1 u_2}$ of length $t = t'$. Then $\boldsymbol{w}$ has suffix $\boldsymbol{tk\ell'}$ in which $\boldsymbol{k\ell'}$ is a prefix of $\boldsymbol{u_1}$. Recall $\boldsymbol{u_1} = \boldsymbol{k\ell' t' \ell}$ has period $p = k + t + \ell$. If $\boldsymbol{t} = \boldsymbol{t'}$, then $\boldsymbol{tu_1}$ has period p; moreover, $\boldsymbol{kw} = \boldsymbol{p}^e$ and $\boldsymbol{tu_1}$ share substring $\boldsymbol{tk\ell'}$ of length p, so $\boldsymbol{u}$ has period p, as desired. Hence, it will suffice to show $\boldsymbol{t} = \boldsymbol{t'}$.

From $\boldsymbol{kw} = \boldsymbol{p}^e$, where $e > 1$, a complete copy of $\boldsymbol{p}$ occurs $h = \lfloor e \rfloor$ times in $\boldsymbol{kw}$. Three cases arise based on where in $\boldsymbol{u}$ the h^{th} occurrence of $\boldsymbol{p}$ ends:

(C1) $\boldsymbol{p}^{(h)}$ ends inside the suffix $\boldsymbol{t}$ of $\boldsymbol{u_2}$.

(C2) $\boldsymbol{p}^{(h)}$ ends inside the prefix $\boldsymbol{k}$ of $\boldsymbol{u_1^{(2)}}$.

(C3) $\boldsymbol{p}^{(h)}$ ends inside the suffix $\boldsymbol{\ell'}$ of $\boldsymbol{w}$.

We will see that $\boldsymbol{t} = \boldsymbol{t'}$ in each of these cases.

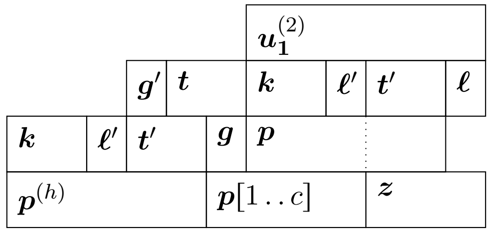

Figure 3.3: Subcase 3 when (C1) holds

(C1)

Suppose (C1) holds; that is, $\boldsymbol{p}^{(h)}$ ends inside the suffix $\boldsymbol{t}$ of $\boldsymbol{u_2}$. We introduce the "gap" $g = u_1+u_2-ph$, a measure of the overlap between $\boldsymbol{t}$ and the suffix $\boldsymbol{t}'$ of $\boldsymbol{p}^{(h)}$. Note that if $g = 0$, then $\boldsymbol{t} = \boldsymbol{t}'$ immediately. Let $\boldsymbol{g} = \boldsymbol{t}[t-g+1\,..\,t] = \boldsymbol{p}[1\,..\,g]$ be the suffix of $\boldsymbol{t}$ that follows $\boldsymbol{p}^{(h)}$, and let $\boldsymbol{g}' = \boldsymbol{t}'[1\,..\,g]$ be the prefix of $\boldsymbol{t}'$ that precedes $\boldsymbol{t}$. Also, let

$$c = (k + w) \bmod p = g + k + \ell$$

and observe that $\boldsymbol{kw}$ has suffix $\boldsymbol{p}[1\,..\,c] = \boldsymbol{gk\ell}' = \boldsymbol{k\ell}'\boldsymbol{g}'$.

Thus $\boldsymbol{p}[1\,..\,c]$ has border $\boldsymbol{k\ell}'$ and therefore period g. String $\boldsymbol{\ell}'\boldsymbol{g}'$ has period ℓ as a prefix of $\boldsymbol{z} = \boldsymbol{\ell}'\boldsymbol{t}'$, and period g as a suffix of $\boldsymbol{p}[1\,..\,c]$, so by Lemma 4 it has period $\gcd(g, \ell)$. Then $\boldsymbol{p}[1\,..\,c]$ has period g and suffix $\boldsymbol{\ell}'\boldsymbol{g}'$ of period $\gcd(g, \ell) \mid g$ and length $\ell + g \geqslant g$, so that by Lemma 3 $\boldsymbol{p}[1\,..\,c]$ itself has period $\gcd(g, \ell)$. Both $\boldsymbol{p}[1\,..\,c]$ and $\boldsymbol{\ell}'\boldsymbol{z}$ have period ℓ and share substring $\boldsymbol{\ell}'$, so $\boldsymbol{p}[1\,..\,c]\boldsymbol{z}$ has period ℓ by Lemma 6. It also has substring $\boldsymbol{p}$, so $\boldsymbol{p}$ has period ℓ. Because $\boldsymbol{p}$ has border $\boldsymbol{\ell}$ as well as period ℓ, any power of $\boldsymbol{p}$ has period ℓ. It follows that $\boldsymbol{kw} = \boldsymbol{p}^e$ has period ℓ and, since $\boldsymbol{p}[1\,..\,c]\boldsymbol{z}$ has period ℓ and shares with $\boldsymbol{kw}$ a substring of length $c > \ell$, $\boldsymbol{u}$ has period ℓ by Lemma 6. Recall that $\boldsymbol{u}$ has substring $\boldsymbol{p}[1\,..\,c]$ of period $\gcd(g, \ell) \mid \ell$, so $\boldsymbol{u}$ itself has period $\gcd(g, \ell)$ by Lemma 3. Recalling that $\boldsymbol{t}$ is a suffix of $\boldsymbol{t}'\boldsymbol{g}$ and that both are substrings of $\boldsymbol{u}$, we find that $\boldsymbol{t}$ and $\boldsymbol{t}'$ have period $\gcd(g, \ell)$ and suffix $\boldsymbol{g}$, so $\boldsymbol{t} = \boldsymbol{t}'$.

(C2)

Suppose (C2) holds; that is, $\boldsymbol{p}^{(h)}$ ends inside the prefix $\boldsymbol{k}$ of $\boldsymbol{u_1}^{(2)}$. Let

$$g = ph - u_1 - u_2$$

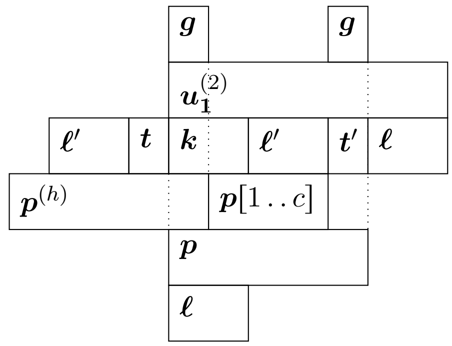

Figure 3.4: Subcase 3 when (C2) holds

be the overlap of $\boldsymbol{p}^{(h)}$ with $\boldsymbol{k}$, and let $\boldsymbol{g} = k[1\,..\,g]$. Note that if $g = 0$, then $\boldsymbol{t} = \boldsymbol{t}'$ immediately. Otherwise, $\boldsymbol{g}$ is a border of $\boldsymbol{p}$. Let

$$c = (k + w) \bmod p = k + \ell - g$$

and observe that, from the overlap of the suffix $\boldsymbol{p}[1\,..\,c]$ of $\boldsymbol{kw}$ and the prefix $\boldsymbol{p}$ of $\boldsymbol{u_1}$, $\boldsymbol{p}[1\,..\,k+\ell] = \boldsymbol{k\ell}'$ has period g. Also note that since by (3.7) $\boldsymbol{p} = \boldsymbol{k\ell}'\boldsymbol{t}'$ and since $\boldsymbol{kw} = \boldsymbol{p}^e$ has period p, therefore $\boldsymbol{p}^{(h)}$ has suffix $\boldsymbol{\ell}'\boldsymbol{tg}$. Consider four cases: $0 < g < \ell$, $g = \ell$, $\ell < g \leqslant z$, and $z < g$.

If $0 < g < \ell$, then since $\boldsymbol{\ell}$ is a prefix of $\boldsymbol{u_1}$, $\boldsymbol{\ell}$ has period g as a prefix of $\boldsymbol{k\ell}'$. Recall that $\boldsymbol{\ell}$ is also a suffix of $\boldsymbol{p}$, so $\boldsymbol{\ell}$ has border $\boldsymbol{g}$ and period $\ell - g$. Hence, by Lemma 4, $\boldsymbol{\ell}$ has period $\gcd(g, \ell - g) = \gcd(g, \ell)$. Recall also that $\boldsymbol{\ell}'\boldsymbol{z}$ has period ℓ and substring $\boldsymbol{\ell}$ of period $\gcd(g, \ell) \mid \ell$, so by Lemma 3, $\boldsymbol{\ell}'\boldsymbol{z}$ has period $\gcd(g, \ell)$. Prefix $\boldsymbol{\ell}'$ of $\boldsymbol{\ell}'\boldsymbol{z}$ then has period $\gcd(g, \ell) \mid g$, and since $\boldsymbol{\ell}'$ is also a suffix of the string $\boldsymbol{k\ell}'$ of period g, $\boldsymbol{k\ell}'$ has period $\gcd(g, \ell)$ by Lemma 3. Since $\boldsymbol{\ell}'\boldsymbol{z}$ and $\boldsymbol{k\ell}'$ have period $\gcd(g, \ell)$, $\boldsymbol{u_1} = \boldsymbol{k\ell}'\boldsymbol{z}$ has period $\gcd(g, \ell)$ by Lemma 6. Both $\boldsymbol{tg}$ and $\boldsymbol{t}'\boldsymbol{g}$ are substrings of $\boldsymbol{u_1}$, so $\boldsymbol{t} = \boldsymbol{t}'$.

If $g = \ell$, then $\boldsymbol{p}$ has suffixes $\boldsymbol{t\ell}$ and $\boldsymbol{z} = \boldsymbol{\ell}'\boldsymbol{t} = \boldsymbol{t}'\boldsymbol{\ell}$, so immediately $\boldsymbol{t} = \boldsymbol{t}'$. Note that $\boldsymbol{k\ell}'$ and $\boldsymbol{\ell}'\boldsymbol{z}$ have period $g = \ell$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{k\ell}'\boldsymbol{t}'$ has period ℓ.

If $\ell < g \leqslant z$, then since $\boldsymbol{\ell}$ is a border of $\boldsymbol{g}$, $\boldsymbol{g}$ has period $g - \ell$; it also has period ℓ as a substring of the suffix $\boldsymbol{z}$ of $\boldsymbol{p}$, and thus by Lemma 4 period $\gcd(g, g - \ell) = \gcd(g, \ell)$. String $\boldsymbol{g}$ is a substring of $\boldsymbol{k\ell}'$, which as we have seen has period g, so that by Lemma 3, $\boldsymbol{k\ell}'$ has period $\gcd(g, \ell)$. Since $\boldsymbol{\ell}'\boldsymbol{z}$ and $\boldsymbol{k\ell}'$ have period $\gcd(g, \ell)$, $\boldsymbol{u_1} = \boldsymbol{k\ell}'\boldsymbol{z}$ has period $\gcd(g, \ell)$ by Lemma 6. Both $\boldsymbol{tg}$ and $\boldsymbol{t}'\boldsymbol{g}$ are substrings of $\boldsymbol{u_1}$, so $\boldsymbol{t} = \boldsymbol{t}'$.

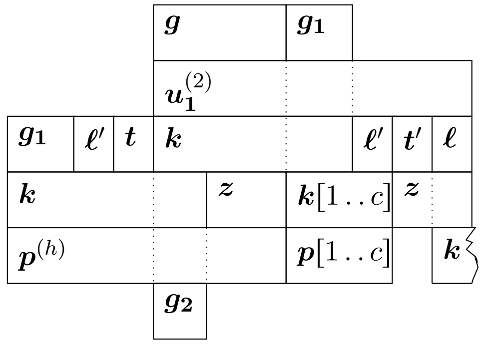

Figure 3.5: Subcase 3 when (C2) holds and $z < g$

If $z < g$, then, as shown in Figure 3.5, the suffix $\boldsymbol{z}$ of $\boldsymbol{p}^{(h)}$ is a substring of the prefix $\boldsymbol{k}$ of $\boldsymbol{u_1}$, and $\boldsymbol{\ell' t}$ is a substring of the prefix $\boldsymbol{k}$ of $\boldsymbol{p}^{(h)}$. $\boldsymbol{k}$ also has two borders $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$: $\boldsymbol{g_1}$ is the border of $\boldsymbol{k}$ of length $g_1 = k - g$ resulting from the overlap of the prefix $\boldsymbol{k}$ of $\boldsymbol{u_1}$ with $\boldsymbol{p}[1\,.\,.\,c] = \boldsymbol{k}[1\,.\,.\,c]$, while $\boldsymbol{g_2}$ is the border of $\boldsymbol{k}$ of length $g_2 = g - z$ resulting from the overlap of the prefix $\boldsymbol{k}$ of $\boldsymbol{p}^{(h)}$ with the prefix $\boldsymbol{k}$ of $\boldsymbol{u_1}$. We then have $\boldsymbol{k} = \boldsymbol{g_1 \ell' t g_2} = \boldsymbol{g_2 \ell' t' g_1}$. Also recall that $\boldsymbol{\ell}$ is a prefix of $\boldsymbol{p} = \boldsymbol{kz}$, so that either $\boldsymbol{\ell}$ is a prefix of $\boldsymbol{g_1}$ or $\boldsymbol{g_1}$ is a prefix of $\boldsymbol{\ell}$.

If $g_1 = g_2$, then $\boldsymbol{t} = \boldsymbol{t'}$ immediately. If $g_1 \neq g_2$, then several cases arise:

1. $g_1 < g_2$

   Let $g' = g'' = g_2 - g_1$, let $\boldsymbol{g'}$ be the prefix of $\boldsymbol{g_2}$ such that $\boldsymbol{g_2} = \boldsymbol{g' g_1}$, and let $\boldsymbol{g''}$ be the suffix of $\boldsymbol{g_2}$ such that $\boldsymbol{g_2} = \boldsymbol{g_1 g''}$. Observe that $\boldsymbol{g_1}$ is a border of $\boldsymbol{g_2}$, so $\boldsymbol{g_2}$ has period g'.

   (a) $g' \leqslant z$ (Figure 3.6)

   The demonstration requires several steps:

   - $\boldsymbol{g'\ell}$ has period ℓ as a suffix of $\boldsymbol{z\ell}$; it also has border $\boldsymbol{\ell}$ and period g', so by Lemma 4, $\boldsymbol{g'\ell}$ has period $\gcd(g', \ell)$.
   - $\boldsymbol{z\ell}$ has period ℓ and suffix $\boldsymbol{g'\ell}$ of period $\gcd(g', \ell) \mid \ell$, so by Lemma 3, $\boldsymbol{z\ell}$ has period $\gcd(g', \ell)$.

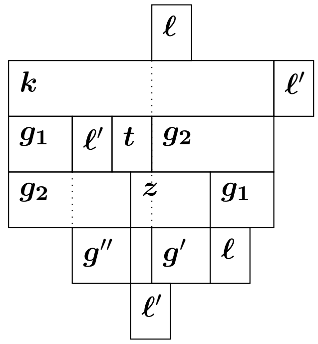

Figure 3.6: Subcase 3 when (C2) holds, $z < g$, $g_1 < g_2$, and $g' \leqslant z$

- $\boldsymbol{g_2}$ has period g' and prefix $\boldsymbol{g'}$ of period $\gcd(g', \ell) \mid g'$, so by Lemma 3, $\boldsymbol{g_2}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{z\ell}$ and $\boldsymbol{g_2}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{g'}$, so by Lemma 6, $\boldsymbol{zg_1}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{g''\ell'}$ has border $\boldsymbol{\ell'}$ and period $g' = g''$; it also has prefix $\boldsymbol{g''}$ which, as a suffix of $\boldsymbol{g_2}$, has period $\gcd(g', \ell) \mid g'$, so by Lemma 3, $\boldsymbol{g''\ell'}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{g_2}$ and $\boldsymbol{g''\ell'}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{g''}$, so by Lemma 6, $\boldsymbol{g_2\ell'}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{g_2\ell'}$ and $\boldsymbol{zg_1}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{\ell'}$, so by Lemma 6, $\boldsymbol{k} = \boldsymbol{g_2zg_1}$ has period $\gcd(g', \ell)$.

Since $\boldsymbol{\ell't}$ and $\boldsymbol{\ell't'}$ are substrings of $\boldsymbol{k}$, therefore $\boldsymbol{t} = \boldsymbol{t'}$.

Note that $\boldsymbol{g''z\ell}$ has period $\gcd(g', \ell)$, so that $\boldsymbol{k} = \boldsymbol{g_1\ell'tg_1g''}$ and $\boldsymbol{g''z\ell}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{g''}$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{kz\ell}$ has period $\gcd(g', \ell)$.

(b) $g' > z$

$\boldsymbol{k}$ has period $k - g_2 = g_1 + z$, so $\boldsymbol{k}$ has a prefix $\boldsymbol{g_1\ell'tg_2'} = \boldsymbol{g_2'zg_1}$, where $\boldsymbol{g_2'}$ is a prefix of $\boldsymbol{g_2}$ and $|g_2' - g_1| \leqslant z$, so one of cases 1(a) and 2(a) applies.

2. $g_1 > g_2$

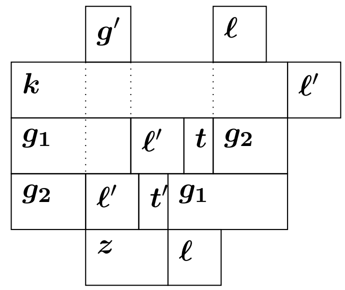

Figure 3.7: Subcase 3 when (C2) holds, $z < g$, $g_1 > g_2$, and $g' \leqslant \ell$

Let $g' = g_1 - g_2$, and let $\boldsymbol{g'}$ be the suffix of $\boldsymbol{g_1}$ such that $\boldsymbol{g_1} = \boldsymbol{g_2 g'}$. Observe that $\boldsymbol{g_2}$ is a border of $\boldsymbol{g_1}$, so $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ have period g'.

(a) $g' \leqslant z$ (Figures 3.7 & 3.8)

Again several steps are required:

- $\boldsymbol{g'\ell'}$ has period ℓ as a prefix of $\boldsymbol{z\ell}$ and also shares prefix $\boldsymbol{\ell'}$ with $\boldsymbol{z}$, so it has border $\boldsymbol{\ell'}$ and period g'.
- $\boldsymbol{g'\ell'}$ has periods ℓ and g', so by Lemma 4, $\boldsymbol{g'\ell'}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{g_1}$ has period g' and suffix $\boldsymbol{g'}$ of period $\gcd(g', \ell) \mid g'$, so by Lemma 3, $\boldsymbol{g_1}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{z\ell}$ has period ℓ and prefix $\boldsymbol{g'}$ of period $\gcd(g', \ell) \mid \ell$, so again by Lemma 3, $\boldsymbol{z\ell}$ has period $\gcd(g', \ell)$.
- $\boldsymbol{z\ell}$ and $\boldsymbol{g_1}$ have period $\gcd(g', \ell)$ and share a substring of length at least $\min(g', \ell)$, so by Lemma 6, $\boldsymbol{zg_1}$ has period $\gcd(g', \ell)$.

Since $\boldsymbol{\ell' t}$ and $\boldsymbol{\ell' t'}$ are substrings of $\boldsymbol{zg_1}$, therefore $\boldsymbol{t} = \boldsymbol{t'}$.

Note that $\boldsymbol{g_1}$ and $\boldsymbol{zg_1}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{g'}$, so by Lemma 6, $\boldsymbol{k} = \boldsymbol{g_2 z g_1}$ has period $\gcd(g', \ell)$. $\boldsymbol{g' z \ell}$ then has period $\gcd(g', \ell)$, so that $\boldsymbol{k} = \boldsymbol{g_2 z g_2 g'}$ and $\boldsymbol{g' z \ell}$ have period $\gcd(g', \ell)$ and share substring $\boldsymbol{g'}$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{kz\ell}$ has period $\gcd(g', \ell)$.

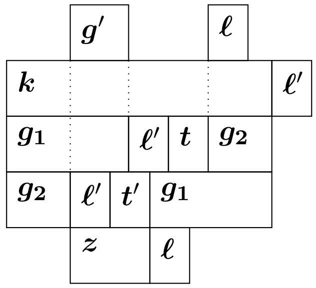

Figure 3.8: Subcase 3 when (C2) holds, $z < g$, $g_1 > g_2$, and $\ell < g' \leqslant z$

(b) $g' > z$

$\boldsymbol{k}$ has period $k - g_1 = g_2 + z$, so $\boldsymbol{k}$ has a prefix $\boldsymbol{g_2 z g_1'} = \boldsymbol{g_1' \ell' t g_2}$, where $\boldsymbol{g_1'}$ is a prefix of $\boldsymbol{g_1}$ and $|g_1' - g_2| \leqslant z$, so one of cases 1(a) and 2(a) applies.

(C3)

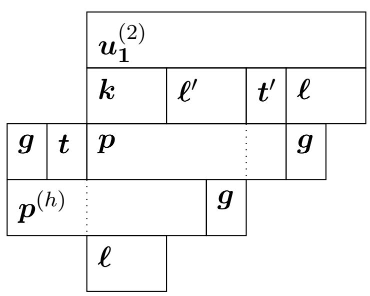

Figure 3.9: Subcase 3 when (C3) holds

Suppose (C3) holds; that is, $\boldsymbol{p}^{(h)}$ ends inside the suffix $\boldsymbol{\ell'}$ of $\boldsymbol{w}$. Let $g = k + w - ph$ and let $\boldsymbol{g} = \boldsymbol{\ell'}[\ell - g + 1 \,..\, \ell]$ be the suffix of $\boldsymbol{\ell'}$ that follows $\boldsymbol{p}^{(h)}$. Because $\boldsymbol{kw}$ (of which $\boldsymbol{\ell'}$ is a suffix) has period p, $\boldsymbol{g}$ is a prefix of $\boldsymbol{p}$. Recall that $\boldsymbol{\ell}$ is a prefix of $\boldsymbol{p}$, so $\boldsymbol{g}$ is also a prefix of $\boldsymbol{\ell}$. From $\boldsymbol{p}^{(h)}$ and the occurrence of $\boldsymbol{p}$ that prefixes $\boldsymbol{u_1}$, we have

$$\boldsymbol{p} = \boldsymbol{gtk}(\boldsymbol{\ell'}[1 \,..\, \ell - g]) = \boldsymbol{k}(\boldsymbol{\ell'}[1 \,..\, \ell - g])\boldsymbol{gt'}$$

and $\boldsymbol{p}$ has period $t+g$.

Consider the string $\boldsymbol{\ell' t' g}$, which occurs near the end of $\boldsymbol{u_1}$ as a prefix of $\boldsymbol{\ell' t' \ell} = \boldsymbol{\ell' z}$. As a substring of $\boldsymbol{\ell' z} = \boldsymbol{\ell' \ell' t'}$, it has period ℓ. Since $\boldsymbol{p}$ has period $t+g$ and suffixes $\boldsymbol{\ell' t'}$ and $\boldsymbol{g t'}$, $\boldsymbol{\ell' t' g}$ also has period $t+g$. $\boldsymbol{\ell' t' g}$ has periods $t+g$ and ℓ, so by Lemma 4, it has period $\gcd(t+g, \ell)$. Now $\boldsymbol{p}$ has period $t+g$ and suffix $\boldsymbol{g t'}$ of period $\gcd(t+g,\ell) \mid t+g$, so $\boldsymbol{p}$ itself has period $\gcd(t+g,\ell)$. Because $\boldsymbol{p}$ has border $\boldsymbol{\ell}$ as well as period $\gcd(t+g,\ell)$, any power of $\boldsymbol{p}$ has period $\gcd(t+g,\ell)$. Thus $\boldsymbol{kw} = \boldsymbol{p}^e$ has period $\gcd(t+g,\ell)$ and, since $\boldsymbol{\ell' z}$ has period $\gcd(t+g,\ell)$ and shares with $\boldsymbol{kw}$ a substring of length ℓ, $\boldsymbol{u}$ has period $\gcd(t+g,\ell)$ by Lemma 6. Since $\boldsymbol{t\ell}$ and $\boldsymbol{t'\ell}$ are substrings of $\boldsymbol{u}$, therefore $\boldsymbol{t} = \boldsymbol{t'}$.

This completes the proof of Subcase 3. □

Lemma 16. *Suppose the conditions of Subcase 3 hold (Lemma 15). Then* $\boldsymbol{x} = \boldsymbol{d}^{x/d}$*, except possibly for*

1. $k + 2w \geqslant v$ ***or***

2. $k + 2w < v$ ***and***

   (a) $v - u = h(u - w)$ ***or***

   (b) $v - u = (h - \frac{1}{2})(u - w)$,

where $d = \gcd(u, v, w)$ *and* $h = \left\lfloor \frac{k+w}{u-w} \right\rfloor$.

Proof. Note first that conditions 1 and 2 are simply reformulations of inequalities (3.4) and (3.5), respectively, found at the beginning of the proof of Lemma 15, where $v-u$ and $u-w$ replace u_1+u_2 and p, respectively. These inequalities constitute the two main cases considered in the proof, and so the result holds for condition 1.

Condition 2 however breaks down into cases (C1)-(C3). For both (C1) and (C3), it is shown that $\boldsymbol{u_1}$ has period ℓ (all symbols used as defined in the proof of Lemma 15). The same is true also for the various subcases of (C2), except when

(a′) the gap $g = 0$ **or**

(b′) $z < g$ and $g_1 = g_2$.

In all other cases, it is shown that $\boldsymbol{u_1} = \boldsymbol{kz\ell} = \boldsymbol{p\ell}$ has period ℓ. Then, by Lemma 15, $\boldsymbol{u} = \boldsymbol{u_1 u_2 u_1}$ has period $d_1 = \gcd(p, u_1 + u_2)$. Hence $\boldsymbol{p}$ is a suffix of $\boldsymbol{u_1}$, thus a border of $\boldsymbol{u}$. Therefore $\boldsymbol{u}^2$ has period d_1, as well as period u, so that by Lemma 4, $\boldsymbol{u}^2$ has period $\gcd(u, d_1) = \gcd(u, \gcd(u-w, v-u)) = \gcd(u, v, w) = d$. This periodicity clearly extends to all of $\boldsymbol{x}$.

Next observe that in the proof of Lemma 15, $g = u_1 + u_2 - ph$, so that the condition $g = 0$ given in (a′) converts to the condition of (a) using the indicated substitutions for $u_1 + u_2$ and p. Again from the proof of Lemma 15, we find that when $z < g$, $g_1 = k - g, g_2 = g - z$, from which we conclude that $p = k + z = 2g$ in (b′). This in turn implies that h copies of $\boldsymbol{p}$ ($2h$ copies of $\boldsymbol{g}$) cover $\boldsymbol{u_1 u_2 g}$ of length $v - u + g$, from which (b) follows. □

3.2 Subcase 7

Here we give results for Subcase 7 corresponding to those for Subcase 3:

Lemma 17 (Subcase 7)**.** *Suppose that a string $\boldsymbol{x}$ has prefixes $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$, $3u/2 < v < 2u$, and suppose further that a third square $\boldsymbol{w}^2$, $w \neq u$, occurs at position $k+1$ of $\boldsymbol{x}$, where*

$$u_1 < k < u_1 + u_2 < w < v \tag{3.8}$$

$$k + w \leqslant u + u_1 \tag{3.9}$$

$$k + 2w \leqslant 2u \tag{3.10}$$

and $u_1 = 2u - v$ and $u_2 = 2v - 3u$. Then $\boldsymbol{x} = \boldsymbol{d_1}^{u/d_1} \boldsymbol{d_1}^{v/d_1} \boldsymbol{d_1}^{(v-u)/d_1}$, where $d_1 = \gcd(u - w, v - u)$.

Proof. By Lemma 12, the overlap of $\boldsymbol{u}^2$ and $\boldsymbol{v}^2$ forces $\boldsymbol{x} = (\boldsymbol{u_1 u_2 u_1 u_1 u_2})^2$, with $\boldsymbol{u} = \boldsymbol{u_1 u_2 u_1}$. By Lemma 14(b), $\boldsymbol{u} = \boldsymbol{z p}^e$, where $\boldsymbol{z} = \boldsymbol{u}[1 \,..\, k + w - u]$, $\boldsymbol{p} = \boldsymbol{w}[1 \,..\, u - w]$, and $e = 1 + \frac{u-k}{u-w}$. See Figure 3.10.

We first show that if $\boldsymbol{u}$ has period $p = u - w$, the lemma holds. Note that $\boldsymbol{u}$ has period $u_1 + u_2$ and

$$u_1 + u_2 + p = u + u_1 + u_2 - w < u$$

since $u_1 + u_2 < w$ from (3.8). Assuming $\boldsymbol{u}$ has period p, $\boldsymbol{u} = \boldsymbol{x}[1 \,..\, u]$ has period $d_1 = \gcd(p, u_1 + u_2)$ by Lemma 4. It follows that $\boldsymbol{u_1 u_2} = \boldsymbol{x}[u + v +$

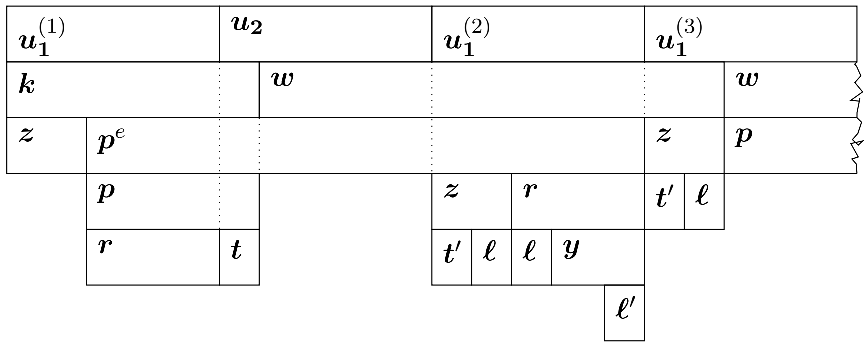

Figure 3.10: String $\boldsymbol{uu_1}$ in Subcase 7

$1\,..\,x]$, a prefix of $\boldsymbol{u} = \boldsymbol{u_1u_2u_1}$, has period d_1 as well. Finally, $\boldsymbol{x}[u+1\,..\,u+v] = \boldsymbol{u_1u_2u_1u_2u_1}$ has period u_1+u_2 and prefix $\boldsymbol{u}$ of length $u > u_1+u_2$ with period d_1. Since $d_1 = \gcd(u-w, u_1+u_2)$ divides u_1+u_2, $\boldsymbol{x}[u+1\,..\,u+v]$ has period d_1 by Lemma 3. Thus the lemma holds assuming $\boldsymbol{u}$ has period p.

We now embark on a demonstration that $\boldsymbol{u}$ has period p. Notice (Figure 3.10) that $\boldsymbol{u_1u_2}$, $\boldsymbol{k}$, and $\boldsymbol{zp}$ are prefixes of $\boldsymbol{u}$. Given that $z = k-p$, that $k \in (u_1, u_1+u_2)$ by (3.8), and that $z \leqslant u_1$ by (3.9), we have

$$\boldsymbol{k} = \boldsymbol{zp} = \boldsymbol{zrt} = \boldsymbol{u_1t}, \tag{3.11}$$

where $\boldsymbol{r} = \boldsymbol{u_1}[z+1, u_1]$, and $\boldsymbol{t} = \boldsymbol{u_2}[1\,..\,k-u_1]$.

Observe that

$$z - p = (k+w-u) - (u-w) = k - (2u-2w) \leqslant 0$$

by (3.10), so that $p \geqslant z$. Also, by (3.8)

$$z = k + w - u > k + u_1 + u_2 - u = k - u_1 > 0,$$

so that in fact $z \geqslant 2$. Note further from (3.8) that

$$p = u - w < u - (u_1+u_2) = u_1,$$

while from (3.10) and (3.8),

$$p = u - w \geqslant k/2 > u_1/2. \tag{3.12}$$

Thus $u_1/2 < k/2 \leqslant p < u_1$. Putting these inequalities together, we find

$$2 \leqslant z \leqslant p < u_1 = z + r, \tag{3.13}$$

from which we conclude that $r > 0$.

Also, since

$$z \leqslant p = r + t < u_1 = r + z,$$

and recalling from (3.8) that $t = k - u_1 > 0$, we see that $0 < t < z$, where since $k = z + p \geqslant 2z$, $z \leqslant k/2$. Hence

$$0 < t < z \leqslant k/2 \leqslant p < u_1.$$

Let $\boldsymbol{t'}$ be the prefix of $\boldsymbol{z}$ of length t. Since $t' = t < z$ and $\boldsymbol{u_1 z}$ is a suffix of $\boldsymbol{w}^{(1)}$, there exists within $\boldsymbol{w}$ a complete occurrence of $\boldsymbol{u_1 t'}$.

Since $\boldsymbol{w}^{(1)}$ has prefix $\boldsymbol{p}$, so also does $\boldsymbol{w}^{(2)}$, with $\boldsymbol{p} = \boldsymbol{rt}$. Furthermore $\boldsymbol{k} = \boldsymbol{zrt} = \boldsymbol{zp}$, so that $\boldsymbol{p}$ is a suffix of $\boldsymbol{k}$ with nonempty prefix $\boldsymbol{r}$ that is a suffix of $\boldsymbol{u_1}$. Since $\boldsymbol{u_1}$ is a proper substring of $\boldsymbol{w}^{(1)}$ and $p < u_1$, it follows that $\boldsymbol{u_1}$ has period p. In fact, the string

$$\boldsymbol{u'} = R_z(\boldsymbol{u}) = \boldsymbol{p}^e \boldsymbol{z} = \boldsymbol{pw} = \boldsymbol{r u_2 u_1 z}$$

has period p. Then $\boldsymbol{u_1} = \boldsymbol{zr}$, $\boldsymbol{rz}$ and $\boldsymbol{u_1 z} = \boldsymbol{zrz}$ all have period p.

Again by the periodicity of $\boldsymbol{u'}$, there exists a possibly empty $\boldsymbol{y'}$ such that $\boldsymbol{p_1} = \boldsymbol{zy'}$ is a prefix of $\boldsymbol{u_1}$, and a $\boldsymbol{y}$, with $y = y' = p - z$, such that $\boldsymbol{p_2} = \boldsymbol{zy}$ is a suffix of $\boldsymbol{u_1}$, where $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ are both rotations of $\boldsymbol{p}$.

Now consider $\boldsymbol{u_1 z}$, a suffix of $\boldsymbol{u'}$ with period p: this string has prefix $\boldsymbol{zy'z}$ and suffix $\boldsymbol{zyz}$ which overlap each other by

$$\hat{p} = u_1 + z - 2u_1 + 2p = p + (p + z) - u_1 > p$$

positions. We may therefore conclude that all substrings of length p in $\boldsymbol{zy'z}$ and $\boldsymbol{zyz}$ are rotations of each other. Then $\boldsymbol{u_1}$ and $R_z(\boldsymbol{u_1})$ both have period p, and so, since $\ell = u_1 - p = z - t < z$, we can apply Lemma 9(a) (with $(x, v, u) \sim (u_1, z, p)$) to conclude that $R_t(\boldsymbol{u_1})[1..2(z-t)]$ is a square of period ℓ. Thus we may write $\boldsymbol{u_1} = \boldsymbol{t'}\boldsymbol{\ell}^2 \cdots$, where $\boldsymbol{t'\ell} = \boldsymbol{z}$. In fact, since $p = z + y = u_1 - \ell$, so that $u_1 = z + \ell + y$, we find that $\boldsymbol{u_1} = \boldsymbol{t'}\boldsymbol{\ell}^2\boldsymbol{y}$ with $\boldsymbol{z} = \boldsymbol{t'\ell}$, $\boldsymbol{r} = \boldsymbol{\ell y}$ and $\boldsymbol{p} = \boldsymbol{\ell y t}$.

Since $\boldsymbol{u'} = \boldsymbol{pw}$ has period p, $\boldsymbol{w}$ is a prefix of $\boldsymbol{u'}$. As we see from Figure 3.10, this prefix $\boldsymbol{w}$ ends distance $r + t = \ell + y + t' = y + t' + \ell$ before the end of $\boldsymbol{w}^{(1)}$, from which we conclude that $\boldsymbol{w}$ has suffix $\boldsymbol{t'\ell\ell}$ as well as suffix $\boldsymbol{z} = \boldsymbol{t'\ell}$. Thus $\boldsymbol{t'}$ is a border of $\boldsymbol{z}$. Now let $\boldsymbol{\ell'}$ be the prefix of $\boldsymbol{z}$ of length ℓ, so that $\boldsymbol{z} = \boldsymbol{t'\ell} = \boldsymbol{\ell' t'}$ has period ℓ. Note further that since $\boldsymbol{w}$ has suffix $\boldsymbol{yz}$, which in turn has suffix $\boldsymbol{t'\ell\ell} = \boldsymbol{\ell' t' \ell}$, therefore $\boldsymbol{\ell'}$ is a suffix of $\boldsymbol{y}$.

Assume $\boldsymbol{t} = \boldsymbol{t}'$. Then $\boldsymbol{k} = \boldsymbol{zrt} = \boldsymbol{zrt}'$ occurs in $\boldsymbol{w}$, and as $\boldsymbol{w}$ has period p, so does $\boldsymbol{k}$. $\boldsymbol{u}$ then has prefix $\boldsymbol{k} = \boldsymbol{zp}$ and suffix $\boldsymbol{p}^e$ both of period p and both including $\boldsymbol{p}$, so by Lemma 6, $\boldsymbol{u}$ has period p, as desired. Hence it will suffice to show $\boldsymbol{t} = \boldsymbol{t}'$.

From (3.13) it follows that a complete copy of $\boldsymbol{p}$ occurs $h \geqslant 2$ times in $\boldsymbol{u}'$. Several cases arise, based on the position of the suffix $\boldsymbol{t}$ of the h^{th} occurrence of $\boldsymbol{p}$:

(C1) $\boldsymbol{t}$ ends inside the prefix $\boldsymbol{z}$ of $\boldsymbol{u_1}^{(3)}$

(C2) $\boldsymbol{t}$ is a substring of the suffix $\boldsymbol{y}$ of $\boldsymbol{u_1}^{(2)}$, but $\boldsymbol{t\ell}$ is not.

(C3) $\boldsymbol{t\ell}$ is a substring of the suffix $\boldsymbol{y}$ of $\boldsymbol{u_1}^{(2)}$.

(C4) $\boldsymbol{t}$ begins to the left of the suffix $\boldsymbol{y}$ of $\boldsymbol{u_1}^{(2)}$ and ends inside $\boldsymbol{y}$.

We will see $\boldsymbol{t} = \boldsymbol{t}'$ in all of these cases.

(C1)

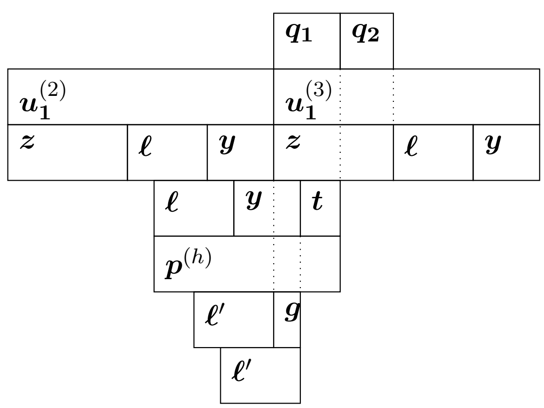

Figure 3.11: Subcase 7 when (C1) holds and $g > 0$

Suppose first that (C1) holds, and write $\boldsymbol{z} = \boldsymbol{q_1 q_2}$, where $\boldsymbol{q_1}$ is a nonempty suffix of $\boldsymbol{p}$ and, by the periodicity of $\boldsymbol{u}'$, $\boldsymbol{q_2}$ is a prefix of $\boldsymbol{p}$.

We have shown that $\boldsymbol{u}' = \boldsymbol{p}^h\boldsymbol{q_2}$, where $\boldsymbol{p} = \boldsymbol{\ell yt}$, and so $\boldsymbol{u}' = \boldsymbol{p}^{h-1}(\boldsymbol{\ell y})(\boldsymbol{t})\boldsymbol{q_2}$. As in the proof of Lemma 15, we introduce the "gap" $g = q_1 - t$, a measure of the overlap between the prefix $\boldsymbol{q_1}$ of $\boldsymbol{u_1}^{(3)}$ and the suffix $\boldsymbol{t}$ of $\boldsymbol{p}^{(h)}$. If $g \geqslant 0$, then $\boldsymbol{t}$ is a substring of $\boldsymbol{z}$; otherwise, $\boldsymbol{t}$ ends inside $\boldsymbol{z}$ but begins before it. Note that if $g = 0$, then $\boldsymbol{q_1} = \boldsymbol{t} = \boldsymbol{t}'$ and the remainder of the proof follows.

Suppose then that $g > 0$ (Figure 3.11), so that $\boldsymbol{q_1} = \boldsymbol{gt}$ for some string $\boldsymbol{g}$ of length g. In this case, note that $\boldsymbol{\ell' t}$ and $\boldsymbol{\ell' t'}$ are substrings of $\boldsymbol{\ell' z} = \boldsymbol{\ell' \ell' t'}$, as we have seen of period ℓ, and so both these strings also have period ℓ, implying that $\boldsymbol{t} = \boldsymbol{t}'$, as required.

We now show further that for $g > 0$, $\boldsymbol{u_1}$ has period $\gcd(g, \ell)$. Since $\boldsymbol{t}$ is a substring of $\boldsymbol{z} = \boldsymbol{t'\ell}$, $g \leqslant \ell$. Therefore, since $\boldsymbol{\ell y}$ is a suffix of $\boldsymbol{u_1}$ and $\boldsymbol{p} = \boldsymbol{\ell y t}$, $\boldsymbol{\ell y}$ and $\boldsymbol{\ell}$ both have period g, as does $\boldsymbol{\ell}'$, since it is a suffix of $\boldsymbol{\ell y}$. Observe that $\boldsymbol{\ell' g}$ has period ℓ as a prefix of $\boldsymbol{\ell' z}$, as well as period g as a suffix of $\boldsymbol{\ell y}$, so that by Lemma 4, $\boldsymbol{\ell' g}$ has period $\gcd(g, \ell)$. $\boldsymbol{z\ell}$ then has period ℓ and a substring $\boldsymbol{\ell}'$ of period $\gcd(g, \ell) \mid \ell$, so by Lemma 3, $\boldsymbol{z\ell}$ has period $\gcd(g, \ell)$. $\boldsymbol{\ell y}$ has period g and suffix $\boldsymbol{g}$ of period $\gcd(g, \ell)$, so by Lemma 3, it has period $\gcd(g, \ell)$. $\boldsymbol{z\ell}$ and $\boldsymbol{\ell y}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{z\ell y}$ has period $\gcd(g, \ell)$.

Suppose next that $g < 0$, so that $\boldsymbol{t} = \boldsymbol{g q_1}$ for some string $\boldsymbol{g}$ of length $|g|$, as shown in Figure 3.12. Again $\boldsymbol{\ell y}$ and $\boldsymbol{\ell}$ both have period $|g|$. If $|g| \leqslant \ell$, then $\boldsymbol{t\ell}$ is a substring of $\boldsymbol{\ell' z}$, so $\boldsymbol{t} = \boldsymbol{t}'$. However, when $g < 0$, it is possible that $|g| > \ell$. In general, let $\boldsymbol{g}'$ be the suffix of $\boldsymbol{z}$ of length $|g|$. The suffix $\boldsymbol{q_2} = \boldsymbol{\ell g'}$ of $\boldsymbol{z}$ has border $\boldsymbol{\ell}$ and thus period $q_2 - \ell = |g|$. It also has period ℓ as a suffix of $\boldsymbol{z}$, so by Lemma 4, it has period $\gcd(g, \ell)$. $\boldsymbol{\ell' z}$ then has period ℓ and suffix $\boldsymbol{\ell g'}$ of period $\gcd(g, \ell) \mid \ell$, so that by Lemma 3, $\boldsymbol{\ell' z}$ has period $\gcd(g, \ell)$. Also by Lemma 3, $\boldsymbol{\ell y}$ has period $\gcd(g, \ell)$ since it has period $|g|$ and, by the periodicity of $\boldsymbol{u}'$, substring $\boldsymbol{g}'$ of period $\gcd(g, \ell) \mid |g|$. Both $\boldsymbol{\ell y}$ and $\boldsymbol{\ell' z}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}'$, so that by Lemma 6, $\boldsymbol{\ell y z}$ has period $\gcd(g, \ell)$. Since $\boldsymbol{t\ell}$ and $\boldsymbol{t'\ell}$ are both substrings of $\boldsymbol{\ell y z}$, therefore again $\boldsymbol{t} = \boldsymbol{t}'$.

Note finally that since $\boldsymbol{z\ell}$ and $\boldsymbol{\ell y}$ both have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}$, therefore by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{z\ell y}$ again has period $\gcd(g, \ell)$, as it did also for $g > 0$.

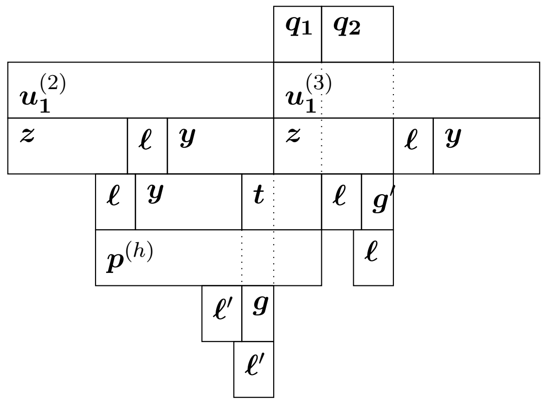

Figure 3.12: Subcase 7 when (C1) holds and $g < 0$

(C2)

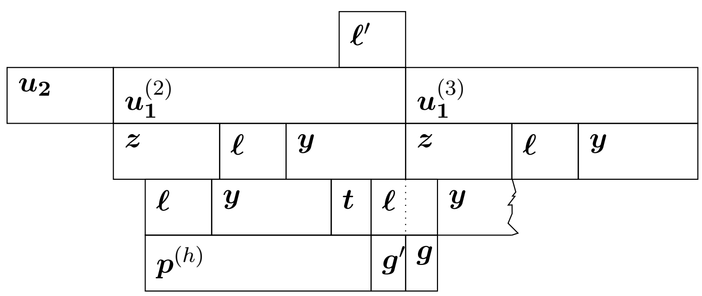

Figure 3.13: Subcase 7 when (C2) holds

Suppose (C2) holds; that is, $\boldsymbol{t}$ is a substring of $\boldsymbol{y}$ and ends within distance $\boldsymbol{\ell}$ of the end of $\boldsymbol{y}$. By the periodicity of $\boldsymbol{u}'$, a prefix of $\boldsymbol{p} = \boldsymbol{\ell y t}$ follows $\boldsymbol{p}^{(h)}$. Let $\boldsymbol{g}$ be the suffix of $\boldsymbol{\ell}$ that overlaps $\boldsymbol{u_1}^{(3)}$, and let $\boldsymbol{g}'$ be the possibly empty prefix of $\boldsymbol{\ell}$ such that $\boldsymbol{\ell} = \boldsymbol{g}'\boldsymbol{g}$. $\boldsymbol{\ell y}$ has period $t + g'$ because the suffix $\boldsymbol{\ell y}$ of $\boldsymbol{u_1}^{(2)}$ and the prefix $\boldsymbol{\ell y}$ of $\boldsymbol{p}^{(h)}$ are offset by length $t + g'$. $\boldsymbol{g}'\boldsymbol{z}$ is a prefix of $\boldsymbol{\ell y}$ since, by assumption, $y \geqslant t + g' = z - g$.

$\boldsymbol{g}'\boldsymbol{z}$ has period $t + g'$ as a prefix of $\boldsymbol{\ell y}$ and period ℓ as a suffix of $\boldsymbol{\ell}'\boldsymbol{z}$, so by lemma 4 it has period $\gcd(t + g', \ell)$. Thus $\boldsymbol{\ell y}$ has period $t + g'$, and a prefix $\boldsymbol{g}'\boldsymbol{z}$ of period $\gcd(t + g', \ell) \mid t + g'$, so $\boldsymbol{\ell y}$ has period $\gcd(t + g', \ell)$ by Lemma 3. Since $\boldsymbol{z\ell}$ has period ℓ and prefix $\boldsymbol{z}$ of period $\gcd(t + g', \ell) \mid \ell$, therefore by Lemma 3, $\boldsymbol{z\ell}$ has period $\gcd(t + g', \ell)$. $\boldsymbol{z\ell}$ and $\boldsymbol{\ell y}$ have period $\gcd(t + g', \ell)$ and share substring $\boldsymbol{\ell}$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{z\ell y}$ has period $\gcd(t + g', \ell)$.

Remarking that $\boldsymbol{\ell}'\boldsymbol{t}$ and $\boldsymbol{\ell}'\boldsymbol{t}'$ are both substrings of $\boldsymbol{u_1}$, we again conclude that $\boldsymbol{t} = \boldsymbol{t}'$.

(C3)

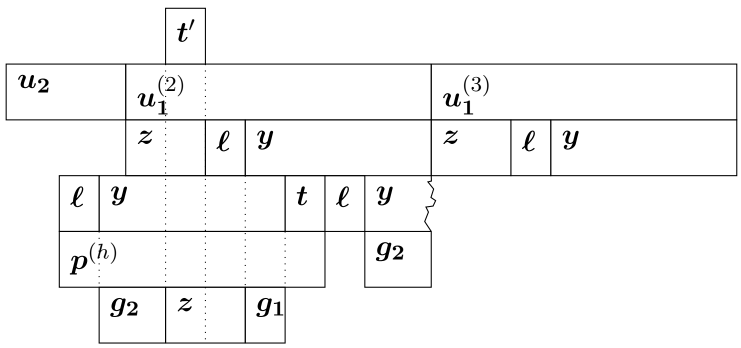

Figure 3.14: Subcase 7 when (C3) holds

Suppose (C3) holds; that is, $\boldsymbol{t\ell}$ is a substring of $\boldsymbol{y}$. In this case, $\boldsymbol{z} = \boldsymbol{t}'\boldsymbol{\ell}$ is also a substring of $\boldsymbol{y}$ because $y \geqslant t + \ell$ and the prefix $\boldsymbol{\ell y}$ of $\boldsymbol{p}^{(h)}$ ends at least z and at most y positions from the end of $\boldsymbol{u_1^{(2)}} = \boldsymbol{\ell}'\boldsymbol{t}'\boldsymbol{\ell y}$.

Let $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ be the (possibly empty) substrings of $\boldsymbol{y}$ immediately before and after $\boldsymbol{t\ell}$ such that $\boldsymbol{y} = \boldsymbol{g_1 t\ell g_2}$. Since $\boldsymbol{u_1 z}$ has period p, $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ are borders of $\boldsymbol{y}$ such that $\boldsymbol{y} = \boldsymbol{g_1 t\ell g_2} = \boldsymbol{g_2 t}'\boldsymbol{\ell g_1}$.

Recall that $\boldsymbol{\ell}'$ is a suffix of $\boldsymbol{y}$, so $\boldsymbol{\ell}'$ is a suffix of $\boldsymbol{\ell g_2}$. Since the prefix $\boldsymbol{\ell g_2}$ of $\boldsymbol{p}$ occurs before the substring $\boldsymbol{z}$ of $\boldsymbol{y}$, $\boldsymbol{\ell}'$ also occurs before $\boldsymbol{z}$.

If $g_1 = g_2$, then $\boldsymbol{t}$ and $\boldsymbol{t}'$ occur at the same positions in two copies of $\boldsymbol{y}$, so that $\boldsymbol{t} = \boldsymbol{t}'$. If $g_1 \neq g_2$, several cases arise:

1. $g_1 < g_2$ $(g = g_2 - g_1)$

   Let $g = g_2 - g_1$, and let $\boldsymbol{g}$ be the nonempty prefix of $\boldsymbol{g_2}$ such that $\boldsymbol{g_2} = \boldsymbol{g g_1}$. $\boldsymbol{g_1}$ is a border of $\boldsymbol{g_2}$, so that $\boldsymbol{g_2}$ has period g.

   (a) $g \leqslant z$ (Figure 3.15)

      The proof requires several steps:

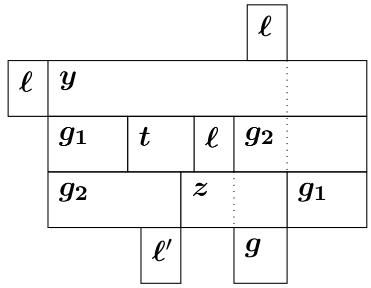

Figure 3.15: Subcase 7 when (C3) holds, $g_1 < g_2$, and $g \leqslant z$

- As a suffix of $\boldsymbol{\ell' z}$, $\boldsymbol{\ell g}$ has period ℓ and suffix $\boldsymbol{\ell}$, hence border $\boldsymbol{\ell}$ and period g, therefore by Lemma 4 period $\gcd(g, \ell)$.
- Then $\boldsymbol{\ell' z}$ has period ℓ and a suffix $\boldsymbol{\ell g}$ of period $\gcd(g, \ell) \mid \ell$, so by Lemma 3, $\boldsymbol{\ell' z}$ has period $\gcd(g, \ell)$.
- Since $\boldsymbol{g_2}$ has period g and prefix $\boldsymbol{g}$ of period $\gcd(g, \ell) \mid g$, therefore by Lemma 3, $\boldsymbol{g_2}$ has period $\gcd(g, \ell)$.
- Since prefix $\boldsymbol{g_2}$ of $\boldsymbol{y}$ and $\boldsymbol{\ell' z}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell'}$, therefore $\boldsymbol{g_2 z}$ has period $\gcd(g, \ell)$ by Lemma 6.
- Thus $\boldsymbol{t\ell}$ and $\boldsymbol{t'\ell}$ both have period $\gcd(g, \ell)$ as substrings of $\boldsymbol{g_2 z}$, implying that $\boldsymbol{t} = \boldsymbol{t'}$.

Note that $\boldsymbol{g_2 z}$ and $\boldsymbol{g_2}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{g}$, so that by Lemma 6, $\boldsymbol{y} = \boldsymbol{g_2 z g_1}$ has period $\gcd(g, \ell)$. Since $\boldsymbol{\ell g_2}$ has period $\gcd(g, \ell)$ as a substring of $\boldsymbol{y}$, and since $\boldsymbol{g_2}$ is a prefix of $\boldsymbol{y}$, therefore by Lemma 6, $\boldsymbol{\ell y}$ has period $\gcd(g, \ell)$. $\boldsymbol{z\ell} = \boldsymbol{\ell' z}$ and $\boldsymbol{\ell y}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}$, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{z \ell y}$ has period $\gcd(g, \ell)$.

(b) $g > z$

$\boldsymbol{y}$ has period $y - g_2 = z + g_1$, so $\boldsymbol{y}$ has a prefix $\boldsymbol{g_1 t \ell g_2'} = \boldsymbol{g_2' z g_1}$, where $\boldsymbol{g_2'}$ is a prefix of $\boldsymbol{g_2}$ and $|g_2' - g_1| \leqslant z$, so one of cases 1(a) and 2(a) applies.

2. $g_1 > g_2$ ($g = g_1 - g_2$)

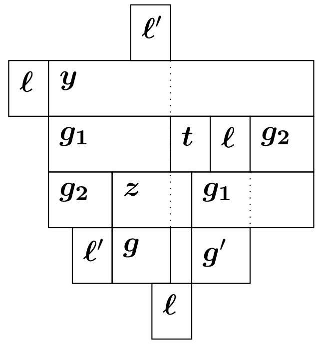

Figure 3.16: Subcase 7 when (C3) holds, $g_1 > g_2$ and $g \leqslant z$

Let $g = g' = g_1 - g_2$, let $\boldsymbol{g}$ be the nonempty suffix of $\boldsymbol{g_1}$ such that $\boldsymbol{g_1} = \boldsymbol{g_2 g}$, and let $\boldsymbol{g'}$ be the nonempty prefix of $\boldsymbol{g_1}$ such that $\boldsymbol{g_1} = \boldsymbol{g' g_2}$. Since $\boldsymbol{g_2}$ is a border of $\boldsymbol{g_1}$, therefore $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ have period $g = g'$.

(a) $g \leqslant z$ (Figure 3.16)

Again there are several steps:

- $\boldsymbol{\ell' g}$ has period ℓ as a prefix of $\boldsymbol{\ell' z}$ and shares suffix $\boldsymbol{\ell'}$ with $\boldsymbol{\ell g_1}$; accordingly, $\boldsymbol{\ell' g}$ has border $\boldsymbol{\ell'}$ and period g, hence periods g and ℓ, thus by Lemma 4 period $\gcd(g, \ell)$.
- $\boldsymbol{\ell' z}$ has period ℓ and prefix $\boldsymbol{\ell' g}$ of period $\gcd(g, \ell) \mid \ell$, hence by Lemma 3, it also has period $\gcd(g, \ell)$.
- $\boldsymbol{g_1}$ has period g and suffix $\boldsymbol{g}$ of period $\gcd(g, \ell) \mid g$, so again by Lemma 3, it also has period $\gcd(g, \ell)$.
- $\boldsymbol{g_1}$ and $\boldsymbol{\ell' z}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{g}$, so that by Lemma 6, $\boldsymbol{g_2 z}$ has period $\gcd(g, \ell)$.
- Prefix $\boldsymbol{\ell g'}$ of $\boldsymbol{\ell g_1}$ shares suffix $\boldsymbol{\ell}$ with $\boldsymbol{t\ell}$, so $\boldsymbol{\ell g'}$ has border $\boldsymbol{\ell}$ and period g. Moreover $\boldsymbol{\ell g'}$ has suffix $\boldsymbol{g'}$ which, as a prefix of $\boldsymbol{g_1}$, has period $\gcd(g, \ell) \mid g$, implying that $\boldsymbol{\ell g'}$ also has period $\gcd(g, \ell)$ by Lemma 3.
- $\boldsymbol{g_2 z}$ and $\boldsymbol{\ell g'}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}$, so by Lemma 6, $\boldsymbol{g_2 z g'}$ has period $\gcd(g, \ell)$.

- Then $\boldsymbol{g_2 z g'}$ and $\boldsymbol{g_1}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{g'}$, so that by Lemma 6 the entire string $\boldsymbol{y} = \boldsymbol{g_2 z g_1}$ has period $\gcd(g, \ell)$.
- Therefore $\boldsymbol{t\ell}$ and $\boldsymbol{t'\ell}$ both have period $\gcd(g, \ell)$ as substrings of $\boldsymbol{y}$, so that $\boldsymbol{t} = \boldsymbol{t'}$, as required.

Since, as a substring of $\boldsymbol{y}$, $\boldsymbol{\ell g_1}$ has period $\gcd(g, \ell)$, and since $\boldsymbol{g_1}$ is also a prefix of $\boldsymbol{y}$, it follows from Lemma 6 that $\boldsymbol{\ell y}$ has period $\gcd(g, \ell)$. Note then that $\boldsymbol{z\ell} = \boldsymbol{\ell' z}$ and $\boldsymbol{\ell y}$ have period $\gcd(g, \ell)$ and share substring $\boldsymbol{\ell}$, so that by Lemma 6 $\boldsymbol{u_1} = \boldsymbol{z\ell y}$ has period $\gcd(g, \ell)$.

(b) $g > z$

$\boldsymbol{y}$ has period $y - g_1 = g_2 + z$, so $\boldsymbol{y}$ has a prefix $\boldsymbol{g_2 z g_1'} = \boldsymbol{g_1' t \ell g_2}$, where $\boldsymbol{g_1'}$ is a prefix of $\boldsymbol{g_1}$ and $|g_1' - g_2| \leqslant z$, so one cases 1(a) and 2(a) applies.

(C4)

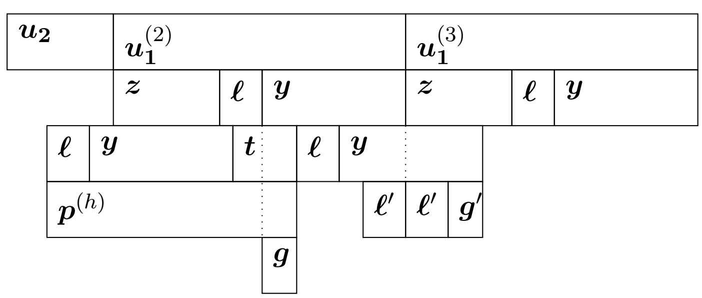

Figure 3.17: Subcase 7 when (C4) holds

Suppose (C4) holds; that is, $\boldsymbol{t}$ begins to the left of $\boldsymbol{y}$ and ends inside it. Let $\boldsymbol{g}$ be the suffix of $\boldsymbol{t}$ that is also a prefix of $\boldsymbol{y}$. Let $\boldsymbol{g'}$ be the suffix of $\boldsymbol{y}$ of length $g' = g$. By the periodicity of $\boldsymbol{u'}$, a copy of $\boldsymbol{ly}$ follows $\boldsymbol{t}$, extending $\ell + g$ positions into $\boldsymbol{u_1^{(3)}}$. $\boldsymbol{\ell'}$ is a suffix of $\boldsymbol{\ell y}$, so $\boldsymbol{\ell'\ell'g'}$ is a suffix of $\boldsymbol{\ell y}$.

The suffix $\boldsymbol{\ell y}$ of $\boldsymbol{u_1}^{(2)}$ and the occurrence of $\boldsymbol{\ell y}$ that follows $\boldsymbol{t}$ are offset by length $\ell+g$, so $\boldsymbol{\ell y}$ has period $\ell+g$. Since $\boldsymbol{\ell'\ell'g'}$ has period $\ell+g$ as a suffix of $\boldsymbol{\ell y}$ and period ℓ as a prefix of $\boldsymbol{\ell' z}$, it therefore has period $\gcd(\ell+g,\ell) = \gcd(g,\ell)$ by Lemma 4. $\boldsymbol{\ell y}$ has period $\ell+g$ and suffix $\ell'\ell'g'$ of period $\gcd(g,\ell) \mid \ell+g$, implying that it has period $\gcd(g,\ell)$ by Lemma 3. $\boldsymbol{z\ell}$ has period ℓ and substring ℓ of period $\gcd(g,\ell) \mid \ell$, so by Lemma 3, it has period $\gcd(g,\ell)$. $\boldsymbol{z\ell}$ and $\boldsymbol{\ell y}$ have period $\gcd(g,\ell)$ and share substring ℓ, so by Lemma 6, $\boldsymbol{u_1} = \boldsymbol{z\ell y}$ has period $\gcd(g,\ell)$. Since $\boldsymbol{t\ell}$ and $\boldsymbol{t'\ell}$ are substrings of $\boldsymbol{u_1}$, we conclude finally that $\boldsymbol{t} = \boldsymbol{t'}$.

This completes the proof of Subcase 7. □

Lemma 18. *Suppose the conditions of Subcase 7 hold (Lemma 17). Then* $\boldsymbol{x} = \boldsymbol{d}^{x/d}$*, except possibly for*

(a) $v - u = (h-1)(u-w)$ ***or***

(b) $v - u = (h - \frac{1}{2})(u-w)$,

where $d = \gcd(u,v,w)$ *and* $h = \left\lfloor \frac{u}{p} \right\rfloor$.

Proof. (All symbols used as defined in the proof of Lemma 17; refer to Figure 3.10.) Notice that in the proof of the lemma, $\boldsymbol{u_1}$ has period ℓ in all cases except when

(a′) (C1) holds and $g = 0$ **and**

(b′) (C3) holds and $g_1 = g_2$.

Suppose then that $\boldsymbol{u_1} = \boldsymbol{zr} = \boldsymbol{z\ell y}$ does indeed have period ℓ. Since $\boldsymbol{z} = \boldsymbol{\ell' t'}$ is a prefix of $\boldsymbol{u_1}$, it follows that $\boldsymbol{t'r} = \boldsymbol{t'\ell y}$ is a suffix of $\boldsymbol{u_1}$. Since $\boldsymbol{u_1}$ has period ℓ, $\boldsymbol{r} = \boldsymbol{\ell y}$ has prefix $\boldsymbol{y}$, and so $\boldsymbol{u_1}$ has border $\boldsymbol{t'\ell y}$ of length $p = t + \ell + y$, as therefore $\boldsymbol{u}$ does also. By Lemma 17, $\boldsymbol{u}$ has period p, so that by Lemma 15, $\boldsymbol{u} = \boldsymbol{u_1 u_2 u_1}$ has period $d_1 = \gcd(p, u_1+u_2) = \gcd(u-w, v-u)$. Since $\boldsymbol{u}$ has a border of length p, it follows that $\boldsymbol{u}^2$ also has period d_1, as well as period u, so that by Lemma 4, $\boldsymbol{u}^2$ has period $\gcd(u, d_1) = \gcd(u, \gcd(u-w, v-u)) = \gcd(u,v,w) = d$. This periodicity clearly extends to all of $\boldsymbol{x}$.

Now consider the exceptional cases. For (a′), recall that in (C1) the gap g is the difference between the two prefixes of $\boldsymbol{x}$, $\boldsymbol{zp}^h$ and $\boldsymbol{u}$, where $p = u - w$, so that $g = 0$ implies $hp + z = u + t$. Substituting $z = k + w - u$, $t = k - u_1$ yields

$$h(u-w) = u - k + (u-w) + k - u_1,$$

from which, with a little manipulation, (a) follows. For (b′), from Figure 3.14 $g_1 = g_2$ in (C3) implies

$$z + ph - t - (u_1 + u_2 + z + \ell) = u - (z + ph + \ell),$$

which since $z = \ell + t$ and $\ell = u_1 - p$ becomes

$$2ph = u + u_1 + u_2 - u_1 + p.$$

A bit more manipulation yields (b), completing the proof. □

Chapter 4

On to the General Case

In this thesis, we have proven the last two remaining subcases of the New Periodicity Lemma, which describes the regularity that must result from three overlapping squares of which two begin at the same position and the third begins to the right. Future work should generalize this result to cases in which three squares occur close to each other, but with no two of them necessarily at the same position. As explained in Section 2.5.2, Lemma 14 provides six cases (a–f) covering all possible configurations of two overlapping squares. This allows the characterization of any instance of the general case of three overlapping squares $\boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{w}^2$ as a pair $[ij]$, $i, j \in [a \mathinner{..} f]$, in which case i corresponds to the overlap of $\boldsymbol{u}^2$ with $\boldsymbol{v}^2$, j the overlap of $\boldsymbol{v}^2$ with $\boldsymbol{w}^2$.

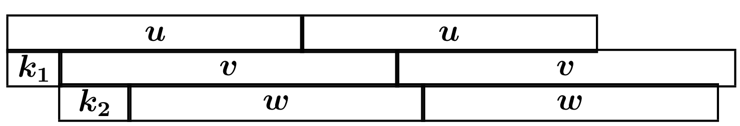

Figure 4.1: $\boldsymbol{u^2}$ overlapping $\boldsymbol{v^2}$ (case (d)) that in turn overlaps $\boldsymbol{w^2}$ (case (b)): what is the combined effect?

As an example, we present a lemma that combines cases d and b (illustrated in Figure 4.1). According to our computational experiments, this lemma applies to about three-quarters of cases in which the maximum alphabet size $\sigma = \gcd(u, v, w)$.

Lemma 19. *In case $[db]$, if $k_2 \leqslant 2u - v - k_1 + d_1$, then $\boldsymbol{x}$ has period d, where $d = \gcd(u, v, w)$ and $d_1 = \gcd(v - u, v - w) = \gcd(u - w, v - w)$.*

Proof. We use subscripts $_1$ to identify variables for $\boldsymbol{u}$ and $\boldsymbol{v}$, subscripts $_2$ for those of $\boldsymbol{v}$ and $\boldsymbol{w}$. Observe then that for $e_1 > 1,\ e_2 \geqslant 1$,

$$\boldsymbol{v} = \boldsymbol{p_1}^{e_1}\boldsymbol{k_1}\boldsymbol{p_1} = \boldsymbol{z_2}\boldsymbol{p_2}^{e_2},$$

where the variables subscripted $_1$ relate to case (d) of Lemma 14, those subscripted $_2$ to case (b). The substring $\boldsymbol{v'} = \boldsymbol{v}[z_2 + 1, v - k_1 - p_1]$ has two periods, $p_1 = v - u$ and $p_2 = v - w$. To apply Lemma 4, we must have

$$\begin{aligned} p_1 + p_2 - \gcd(p_1, p_2) &\leqslant v - k_1 - p_1 - z_2 \\ k_2 &\leqslant 2u - v - k_1 + d_1 \end{aligned}$$

Thus if $k_2 \leqslant 2u - v - k_1 + d_1$, then $\boldsymbol{v'}$ has period d_1. Moreover, $\boldsymbol{v}$ has a prefix of period p_1 that includes $\boldsymbol{v'}$, as well as a suffix of period p_2 that includes $\boldsymbol{v'}$, so $\boldsymbol{v}$ itself has period d_1. Since $\boldsymbol{p_1}$ is a border of $\boldsymbol{v}$, $\boldsymbol{v}$ is a repetition of period d_1. Because $\boldsymbol{u}$ is a substring of $\boldsymbol{v}^2$, $\boldsymbol{u}$ has period d_1. Therefore $\boldsymbol{x}$ has prefix $\boldsymbol{u}$ and suffix $\boldsymbol{v}^2$ both of period d_1 that include $\boldsymbol{v}[1..u - k_1]$. In case (d), $\frac{2(k_1+v)}{3} \leqslant u$ which implies $d_1 \leqslant u - k_1$, so $\boldsymbol{x}$ has period d_1. The substrings $\boldsymbol{u}^2$ and $\boldsymbol{w}^2$ then have periods $\gcd(d_1, u)$ and $\gcd(d_1, w)$, respectively, and so $\boldsymbol{x}$ itself has those periods. Finally, $\boldsymbol{x}$ has period $\gcd(d_1, u, w) = d$. □

The preceding lemma is a sample of the combinatorial information that may be obtained from considering all cases $[ij]$ as specified above. Much work remains to be done to state and prove similar results for all 36 $[ij]$ pairs. To date, all the results given for the New Periodicity Lemma (Crochemore and Rytter 1995; Fan et al. 2006; Franek, Fuller, et al. 2012; Kopylova and Smyth 2012; Simpson 2007) deal only with the special cases $[ij]$, $i \in \{a, d\}$, that arise for $k_1 = 0$. Kopylova and Smyth (2012) used computer simulations for small values of k, u, v, w to help generate conjectures, and it seems that similar techniques can profitably be used for the general case.

Once the combinatorics of overlapping squares is well understood, we may find new and more combinatorially-sophisticated approaches to the bounding of the number of runs in any string of given length. Moreover, it may be possible to design an algorithmic approach to the computation of runs in a manner consistent with their sparseness of occurrence.

Bibliography

[1] Mohamed Ibrahim Abouelhoda, Stefan Kurtz, and Enno Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2(1):53–86, 2004 (cited on page 5).

[2] Alberto Apostolico. The myriad virtues of subword trees. In, *Combinatorial Algorithms on Words*, pages 85–96. Springer, 1985 (cited on page 4).

[3] Alberto Apostolico and Franco P. Preparata. Optimal off-line detection of repetitions in a string. *Theoretical Computer Science*, 22(3):297–315, 1983 (cited on page 9).

[4] Widmer Bland and W.F. Smyth. Three overlapping squares: The general case characterized & applications, 2014. Submitted (cited on pages 14, 16).

[5] M. Gabriella Castelli, Filippo Mignosi, and Antonio Restivo. Fine and Wilf's theorem for three periods and a generalization of Sturmian words. *Theoretical Computer Science*, 218(1):83–94, 1999 (cited on page 7).

[6] Gang Chen, Simon J. Puglisi, and William F. Smyth. Fast and practical algorithms for computing all the runs in a string. In *Combinatorial Pattern Matching*, 2007, pages 307–315 (cited on page 9).

[7] Sorin Constantinescu and Lucian Ilie. Generalised Fine and Wilf's theorem for arbitrary number of periods. *Theoretical Computer Science*, 339(1):49–60, 2005 (cited on page 7).

[8] Max Crochemore. An optimal algorithm for computing the repetitions in a word. *Information Processing Letters*, 12(5):244–250, 1981 (cited on page 9).

[9] Maxime Crochemore and Lucian Ilie. Maximal repetitions in strings. *Journal of Computer and System Sciences*, 74(5):796–807, 2008 (cited on page 10).

[10] Maxime Crochemore, Lucian Ilie, and Liviu Tinta. The "runs" conjecture. *Theoretical Computer Science*, 412(27):2931–2941, 2011 (cited on pages 1, 10).

[11] Maxime Crochemore, Lucian Ilie, and Liviu Tinta. Towards a solution to the "runs" conjecture. In *Combinatorial Pattern Matching*, 2008, pages 290–302 (cited on page 10).

[12] Maxime Crochemore and Wojciech Rytter. Squares, cubes, and time-space efficient string searching. *Algorithmica*, 13(5):405–425, 1995 (cited on pages i, 1, 11, 39).

[13] Kangmin Fan, Simon J. Puglisi, William F. Smyth, and Andrew Turpin. A new periodicity lemma. *SIAM Journal on Discrete Mathematics*, 20(3):656–668, 2006 (cited on pages i, 1, 11–13, 39).

[14] Martin Farach. Optimal suffix tree construction with large alphabets. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*. IEEE, 1997, pages 137–143 (cited on page 4).

[15] Nathan J. Fine and Herbert S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16(1):109–114, 1965 (cited on page 6).

[16] Aviezri S. Fraenkel and Jamie Simpson. An extension of the periodicity lemma to longer periods. *Discrete Applied Mathematics*, 146(2):146–155, 2005 (cited on page 7).

[17] Aviezri S. Fraenkel and Jamie Simpson. The exact number of squares in Fibonacci words. *Theoretical Computer Science*, 218(1):95–106, 1999 (cited on page 9).

[18] Frantisek Franek, Robert C.G. Fuller, Jamie Simpson, and William F. Smyth. More results on overlapping squares. *Journal of Discrete Algorithms*, 17:2–8, 2012 (cited on pages i, 1, 12, 14, 39).

[19] Frantisek Franek, R.J. Simpson, and William F. Smyth. The maximum number of runs in a string. In *Proceedings 14th Australasian Workshop on Combinatorial Algorithms*, 2003, pages 26–35 (cited on page 10).

[20] Edward Fredkin. Trie memory. *Communications of the ACM*, 3(9):490–499, 1960 (cited on page 4).

[21] Robert Giegerich and Stefan Kurtz. From ukkonen to mccreight and weiner: a unifying view of linear-time suffix tree construction. *Algorithmica*, 19(3):331–353, 1997 (cited on page 4).

[22] Mathieu Giraud. Asymptotic behavior of the numbers of runs and microruns. *Information and Computation*, 207(11):1221–1228, 2009 (cited on page 10).

[23] Mathieu Giraud. Not so many runs in strings. In, *Language and Automata Theory and Applications*, pages 232–239, 2008 (cited on page 10).

[24] Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer science and computational biology*. Cambridge University Press, 1997 (cited on page 4).

[25] Anisa Al-Hafeedh, Maxime Crochemore, Lucian Ilie, Evguenia Kopylova, William F. Smyth, German Tischler, and Munina Yusufu. A comparison of index-based Lempel-Ziv LZ77 factorization algorithms. *ACM Computing Surveys*, 45(1):5, 2012 (cited on page 5).

[26] Štěpán Holub. On multiperiodic words. *RAIRO — Theoretical Informatics and Applications*, 40(04):583–591, 2006 (cited on page 7).

[27] Costas S. Iliopoulos, Dennis Moore, and William F. Smyth. A characterization of the squares in a Fibonacci string. *Theoretical Computer Science*, 172(1):281–291, 1997 (cited on page 9).

[28] Jacques Justin. On a paper by Castelli, Mignosi, Restivo. *RAIRO — Theoretical Informatics and Applications*, 34(5):373–377, 2000 (cited on page 7).

[29] Juha Kärkkäinen and Peter Sanders. Simple linear work suffix array construction. In, *Automata, Languages and Programming*, pages 943–955. Springer, 2003 (cited on page 5).

[30] Dong Kyue Kim, Jeong Seop Sim, Heejin Park, and Kunsoo Park. Linear-time construction of suffix arrays. In *Combinatorial Pattern Matching*. Springer, 2003, pages 186–199 (cited on page 5).

[31] Pang Ko and Srinivas Aluru. Space efficient linear time construction of suffix arrays. In *Combinatorial Pattern Matching*. Springer, 2003, pages 200–210 (cited on page 5).

[32] Roman Kolpakov and Gregory Kucherov. On maximal repetitions in words. *Journal on Discrete Algorithms*, 1(1):159–186, 2000 (cited on pages 1, 9, 10).

[33] Evguenia Kopylov. Computing repetitions in strings: Current algorithms & the combinatorics of future ones. McMaster University, 2010 (cited on page 9).

[34] Evguenia Kopylova and William F. Smyth. The three squares lemma revisited. *Journal of Discrete Algorithms*, 11:3–14, 2012 (cited on pages i, 1, 8, 12, 39).

[35] Stefan Kurtz. Reducing the space requirement of suffix trees. *Software—Practice and Experience*, 29(13):1149–71, 1999 (cited on page 4).

[36] Kazuhiko Kusano, Kazuyuki Narisawa, and Ayumi Shinohara. On morphisms generating run-rich strings. In *Prague Stringology Conference 2013*, 2013, page 35 (cited on pages 1, 10).

[37] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22(1):75–81, 1976 (cited on page 5).

[38] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002 (cited on pages 6, 7).

[39] M. Lothaire. *Applied Combinatorics on Words*. Cambridge University Press, 2005 (cited on page 6).

[40] Michael G. Main. Detecting leftmost maximal periodicities. *Discrete Applied Mathematics*, 25(1):145–153, 1989 (cited on pages 8, 9).

[41] Michael G. Main and Richard J. Lorentz. An $\mathcal{O}(n \log n)$ algorithm for finding all repetitions in a string. *Journal of Algorithms*, 5(3):422–432, 1984 (cited on page 9).

[42] Udi Manber and Gene Myers. Suffix arrays: A new method for online string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993 (cited on page 5).

[43] Michael A. Maniscalco and Simon J. Puglisi. An efficient, versatile approach to suffix sorting. *Journal of Experimental Algorithmics*, 12:1–2, 2008 (cited on page 5).

[44] Michael A. Maniscalco and Simon J. Puglisi. Faster lightweight suffix array construction. In *Proceedings of the 17th International Workshop on Combinatorial Algorithms*, 2006, pages 16–29 (cited on page 5).

[45] Giovanni Manzini and Paolo Ferragina. Engineering a lightweight suffix array construction algorithm. *Algorithmica*, 40(1):33–50, 2004 (cited on page 5).

[46] Wataru Matsubara, Kazuhiko Kusano, Akira Ishino, Hideo Bannai, and Ayumi Shinohara. New lower bounds for the maximum number of runs in a string. In *Prague Stringology Conference 2008*, 2008, page 140 (cited on page 10).

[47] Edward M. McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM*, 23(2):262–272, 1976 (cited on page 4).

[48] Ge Nong, Sen Zhang, and Wai Hong Chan. Linear time suffix array construction using d-critical substrings. In *Combinatorial Pattern Matching*. Springer, 2009, pages 54–67 (cited on page 5).

[49] Simon J. Puglisi and Jamie Simpson. The expected number of runs in a word. *Australasian Journal of Combinatorics*, 42:45–54, 2008 (cited on pages 2, 9).

[50] Simon J. Puglisi, Jamie Simpson, and William F. Smyth. How many runs can a string contain? *Theoretical Computer Science*, 401(1):165–171, 2008 (cited on page 10).

[51] Simon J. Puglisi, William F. Smyth, and Andrew H. Turpin. A taxonomy of suffix array construction algorithms. *ACM Computing Surveys*, 39(2):4, 2007 (cited on page 5).

[52] Wojciech Rytter. The number of runs in a string. *Information and Computation*, 205(9):1459–1469, 2007 (cited on page 10).

[53] Wojciech Rytter. The number of runs in a string: Improved analysis of the linear upper bound. In *Proceedings of the 23rd Symposium on Theoretical Aspects of Computer Science*. Volume 3884, 2006, page 184 (cited on page 10).

[54] Jamie Simpson. Intersecting periodic words. *Theoretical Computer Science*, 374(1):58–65, 2007 (cited on pages i, 1, 7, 12, 39).

[55] Jamie Simpson. Modified Padovan words and the maximum number of runs in a word. *Australasian Journal of Combinatorics*, 46:129–145, 2010 (cited on pages 1, 10).

[56] R.J. Simpson and Robert Tijdeman. Multi-dimensional versions of a theorem of Fine and Wilf and a formula of Sylvester. *Proceedings of the American Mathematical Society*, 131(6):1661–1671, 2003 (cited on page 7).

[57] Bill Smyth. *Computing Patterns in Strings*. Pearson Education, 2003 (cited on pages 3, 8, 9).

[58] Robert Tijdeman and Luca Q. Zamboni. Fine and Wilf words for any periods II. *Theoretical Computer Science*, 410(30):3027–3034, 2009 (cited on page 7).

[59] Esko Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995 (cited on page 4).

[60] Peter Weiner. Linear pattern matching algorithms. In *Proceedings of the 14^{th} Annual Symposium on Switching and Automata Theory*. IEEE, 1973, pages 1–11 (cited on page 4).