MATHEMATICS OF COMPUTATION
Volume 75, Number 253, Pages 369–384
S 0025-5718(05)01769-2
Article electronically published on July 12, 2005

MINIMALITY AND OTHER PROPERTIES OF THE WIDTH-w NONADJACENT FORM

JAMES A. MUIR AND DOUGLAS R. STINSON

ABSTRACT. Let $w \geq 2$ be an integer and let D_w be the set of integers that includes zero and the odd integers with absolute value less than 2^{w-1}. Every integer n can be represented as a finite sum of the form $n = \sum a_i 2^i$, with $a_i \in D_w$, such that of any w consecutive a_i's at most one is nonzero. Such representations are called *width-w nonadjacent forms* (w-NAFs). When $w = 2$ these representations use the digits $\{0, \pm 1\}$ and coincide with the well-known *nonadjacent forms*. Width-w nonadjacent forms are useful in efficiently implementing elliptic curve arithmetic for cryptographic applications. We provide some new results on the w-NAF. We show that w-NAFs have a minimal number of nonzero digits and we also give a new characterization of the w-NAF in terms of a (right-to-left) lexicographical ordering. We also generalize a result on w-NAFs and show that any base 2 representation of an integer, with digits in D_w, that has a minimal number of nonzero digits is at most one digit longer than its binary representation.

1. INTRODUCTION

In many radix 2 positional number systems, integers are represented using *finite* sums of the form $\sum_{i \geq 0} a_i 2^i$. If n is an integer and $n = \sum_{i \geq 0} a_i 2^i$, we call $\sum_{i \geq 0} a_i 2^i$ a *radix 2 representation* of n. To denote radix 2 representations, the following notation is commonly used:

$$(\cdots a_3 a_2 a_1 a_0)_2 = \cdots + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0.$$

Each of the a_i's is called a *digit*. In the usual radix 2 positional number system each digit is equal to 0 or 1.

Let $w \geq 2$ be an integer. A radix 2 representation is called a *width-w nonadjacent form* (w-NAF, for short) if it satisfies the following conditions:

(1) each nonzero digit is an odd integer with absolute value less than 2^{w-1};
(2) of any w consecutive digits, at most one is nonzero.

It is convenient to define D_w to be the set of w-NAF digits; that is, D_w is the set of integers that includes zero and the odd integers with absolute value less than 2^{w-1}. For example, if $w = 3$, then $D_w = \{0, \pm 1, \pm 3\}$. The number 42 has a 3-NAF since the representation $(300\bar{3}0)_2$ (note that $\bar{1}$ denotes -1, $\bar{3}$ denotes -3, etc.) satisfies conditions (1) and (2), and

$$(300\bar{3}0)_2 = 3 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 - 3 \cdot 2^1 + 0 \cdot 2^0 = 42.$$

Received by the editor March 29, 2004 and, in revised form, August 21, 2004.
2000 *Mathematics Subject Classification.* Primary 94A60, 11T71, 14G50; Secondary 94B40.
Key words and phrases. Efficient representations, minimal weight, elliptic curve arithmetic.
The second author is supported by NSERC grant RGPIN 203114-02.

©2005 American Mathematical Society

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

When $w = 2$, $D_w = \{0, \pm 1\}$ and the w-NAF coincides with the well-known *non-adjacent form* [6]. Because of this, the w-NAF may be regarded as a generalization of the ordinary NAF.

Cryptographers became interested in the w-NAF primarily through efforts to efficiently implement elliptic curve scalar multiplication (i.e., computing nP for an integer, n, and an elliptic curve point, P). The basic technique for scalar multiplication is the *binary method* (also known as the *double-and-add method*). The number of elliptic curve group operations required to compute nP using the binary method is related to how the integer n is represented. In particular, if $n = (a_{\ell-1} \cdots a_1 a_0)_2$, then the number of elliptic curve addition operations required is equal to one less than the number of nonzero a_i's.[1]

Suppose, for example, that we wish to compute $3885P$. Consider the following radix 2 representations of 3885:

$$(111100101101)_2,\ (1000\bar{1}010\bar{1}0\bar{1}01)_2,\ (1000\bar{1}00030003\bar{3})_2.$$

The ordinary binary method can compute nP by processing any $\{0, 1\}$-radix 2 representation of n from left to right. For any n, there is exactly one such representation, and for $n = 3885$, this representation is listed above. This representation has eight nonzero digits which results in 7 elliptic curve addition operations.

The *signed binary method* can compute nP by processing any $\{0, \pm 1\}$-radix 2 representation of n from left to right. Using the digit -1 takes advantage of the fact that, in an elliptic curve group, inverses can be computed essentially for free (so it is not necessary to precompute and store $-P$ since it can be computed from P as needed). There are an infinite number of $\{0, \pm 1\}$-radix 2 representations of 3885; however 3885 has a 2-NAF, which is listed above, and it is, in one sense, an optimal choice because it has a *minimal number of nonzero digits* (a result initially due to Reitwiesner [15]). Using this 2-NAF results in 5 elliptic curve addition operations.

The *signed binary sliding window method* [9], with window width $w \geq 2$, can compute nP by processing any D_w-radix 2 representation of n from left to right. Unlike the previous two methods, this method requires that dP be precomputed and stored for each positive digit d in D_w. A 3-NAF of 3885 is listed above, and using it results in 3 elliptic curve addition operations. However, without performing a lengthy computation, it is not obvious if some other D_3-radix 2 representation of 3885 could result in fewer addition operations. In general, it was not known if the w-NAF of an integer has a minimal number of nonzero digits, except in the case when $w = 2$.

We provide an answer to this question in Section 3 of this paper: we prove that no other D_w-radix 2 representation of an integer has fewer nonzero digits than its w-NAF. This result complements the average case analysis carried out in [4] and provides further evidence that the w-NAF is a good representation to use with the signed binary sliding window method. As well, this result may also have applications to the theory of arithmetic codes [10].

In Section 4, we generalize a known result about the length of w-NAFs. It is stated without proof in [12] that the length of the w-NAF of an integer is *at most one digit longer* than its binary representation. We show that this is in fact a property of representations with a minimal number of nonzero digits; that is, any

[1]This does not account for any addition operations that might be performed during a precomputation step.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

D_w-radix 2 representation of an integer with a minimal number of nonzero digits is at most one digit longer than its binary representation.

In Section 5, we provide a new characterization of the w-NAF in terms of a (right-to-left) *lexicographical ordering*. For an integer n, we consider the set of all D_w-radix 2 representations of n. The positions of the zero and nonzero digits in these representations define binary strings. Dictionary, or lexicographical, order is the usual way to compare strings, and we show that under this order the smallest representation in this set is the w-NAF.

Before we present our results, we first establish some of the basic theory on w-NAFs in Section 2. Aside from being of value to readers new to the w-NAF, this material provides proofs for some results which are stated without proof in the literature.

1.1. **Notation.** All of the radix 2 representations we are concerned with in this paper are finite sums of the form $\sum_{i\geq 0} a_i 2^i$, which we denote by $(\cdots a_2a_1a_0)_2$. Since $(\cdots a_2a_1a_0)_2$ stands for a finite sum, all but a finite number of the a_i's are zero. Because of this property, we can consider the *length* of a representation:

Definition 1.1. The *length* of a representation $(\cdots a_2a_1a_0)_2$ is the integer

$$\min\{\ell \in \mathbb{Z} : \ell \geq 0, \text{ and for any } i \geq \ell,\ a_i = 0\}.$$

For the representation $(a_{\ell-1}\cdots a_1a_0)_2$, it is implicit that $a_i = 0$ for all $i \geq \ell$; note that if $a_{\ell-1} \neq 0$, then this representation has length ℓ.

The set, D_w, of w-NAF digits, was defined earlier. The set of all *strings* of digits from D_w is denoted by ${D_w}^*$. The empty string is in ${D_w}^*$ and is denoted by ϵ. Now, given a representation $(a_{\ell-1}\cdots a_1a_0)_2$, where each a_i is in D_w, then $a_{\ell-1}\cdots a_1a_0$ is a string in ${D_w}^*$. Conversely, any string $\alpha \in {D_w}^*$ corresponds to a radix 2 representation with digits in D_w, namely $(\alpha)_2$. If $\alpha, \beta \in {D_w}^*$, then we denote their *concatenation* by $\alpha\|\beta$. Also, we denote the string formed by deleting the leading zeros from α by $\widehat{\alpha}$.

It is important to note that representations and strings have different properties. For example, the strings $300\bar{3}0$ and $00300\bar{3}0$ are different; however $(300\bar{3}0)_2$ and $(00300\bar{3}0)_2$ denote the same representation.

If α is a string of digits, then we write $\mathsf{wt}(\alpha)$ to denote the number of nonzero digits in α. The value $\mathsf{wt}(\alpha)$ is often called the Hamming weight of α.

2. Known results

The w-NAF seems to have been first described by Cohen, Miyaji and Ono [5]. However, the w-NAF is closely related to the binary window method and this may explain why it was proposed independently by Blake, Seroussi and Smart [2] and by Solinas [16].

Results on the w-NAF are scattered among different papers, and often proofs are not given. For completeness, we give proofs of the following basic facts about the w-NAF:

(1) every integer has at most one w-NAF;
(2) every integer has a w-NAF;
(3) an integer's w-NAF is at most one digit longer than its binary representation.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

2.1. **Uniqueness.**

Proposition 2.1. *Every integer has at most one w-NAF.*

Proof. We suppose the result is false and show that this leads to a contradiction. Suppose there are two different w-NAFs, say $(a_{\ell-1} \cdots a_2 a_1 a_0)_2$ and $(b_{\ell'-1} \cdots b_2 b_1 b_0)_2$, such that

$$(a_{\ell-1} \cdots a_2 a_1 a_0)_2 = (b_{\ell'-1} \cdots b_2 b_1 b_0)_2,$$

where ℓ and ℓ' are the respective lengths of these representations. We can assume that ℓ is as small as possible. These representations stand for the same integer, call it n.

If $a_0 = b_0$, then

$$(a_{\ell-1} \cdots a_2 a_1)_2 = (b_{\ell'-1} \cdots b_2 b_1)_2,$$

and so we have two different, and shorter, w-NAFs which stand for the same integer, contrary to the minimality of ℓ. So, it must be that $a_0 \neq b_0$.

If n is even, then $a_0 = b_0 = 0$. However, $a_0 \neq b_0$, so it must be that n is odd; hence, both a_0 and b_0 are nonzero. Because the representations are both w-NAFs, we have

$$(a_{\ell-1} \cdots a_w 00 \cdots 0 a_0)_2 = (b_{\ell'-1} \cdots b_w 00 \cdots 0 b_0)_2$$
$$\implies a_0 \equiv b_0 \pmod{2^w}.$$

However, $-(2^{w-1} - 1) \leq a_0, b_0 \leq 2^{w-1} - 1$, and thus

$$-2(2^{w-1} - 1) \leq a_0 - b_0 \leq 2(2^{w-1} - 1).$$

The only multiple of 2^w in this range is 0, and since $2^w | (a_0 - b_0)$ it must be that $a_0 - b_0 = 0$. However, this contradicts the fact that $a_0 \neq b_0$. Thus, the representations cannot exist and the result follows. □

2.2. **Existence.** We present an algorithm which, on input n, computes a string α such that $(\alpha)_2$ is a w-NAF of n. Unlike the algorithms in [2] and [16], our algorithm handles negative integers as well as positive ones. Proving that the algorithm is correct establishes that every integer has a w-NAF.

The quotient-remainder theorem tells us that, for any integer n, there exist unique integers q' and r' such that

$$n = q' \cdot 2^w + r' \quad \text{where } 0 \leq r' < 2^w.$$

It is common to denote this value of r' by "$n \bmod 2^w$". It follows that there also exist unique integers q and r such that

$$n = q \cdot 2^w + r \quad \text{where } -2^{w-1} < r \leq 2^{w-1}.$$

We will denote this value of r by "n mods 2^w". For example, if $w = 3$, then $13 \text{ mods } 2^3 = -3$. Note that if n is odd, then so is n mods 2^w. As well, when $n > 0$ it must be that $q \geq 0$, and similarly, when $n < 0$, $q \leq 0$. So, for $n \neq 0$, we have $q/n \geq 0$.

Our algorithm makes use of the following two functions:

$$f_w(n) := \begin{cases} n/2 & \text{if } n \text{ is even}, \\ (n - r)/2^w & \text{where } r = n \text{ mods } 2^w, \text{ otherwise}; \end{cases} \tag{2.1}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

$$g_w(n) := \begin{cases} 0 & \text{if } n \text{ is even,} \\ 0^{w-1}r & \text{where } r = n \text{ mods } 2^w, \text{ otherwise.} \end{cases} \tag{2.2}$$

Note that f_w returns an integer and g_w returns a string. For example, if $w = 3$, then $f_3(13) = 2$ and $g_3(13) = 00\overline{3}$.

Now we can describe our algorithm:

Algorithm2.1: NAF$_w(n)$

```
$\alpha \leftarrow \epsilon$
while $n \neq 0$
  do $\begin{cases} \alpha \leftarrow g_w(n) \parallel \alpha \\ n \leftarrow f_w(n) \end{cases}$
return $\widehat{\alpha}$
```

As NAF$_w(n)$ executes, it builds a string, α, in ${D_w}^*$. Assuming NAF$_w(n)$ terminates, which we will prove in a moment, it returns this string minus its leading zeros (i.e., $\widehat{\alpha}$).

We justify the title "Algorithm" by showing that NAF$_w(n)$ terminates for all $n \in \mathbb{Z}$. If $n = 0$, then NAF$_w(n)$ clearly terminates, so we need only consider $n \neq 0$. We will argue that $|f_w(n)| < |n|$ whenever $n \neq 0$.

If n is even, then $f_w(n) = n/2$ and thus $|f_w(n)| < |n|$. If n is odd, then we consider two cases. First, suppose $|n| < 2^{w-1}$. Then we see that n mods $2^w = n$ and thus

$$|f_w(n)| = 0 < |n| .$$

Second, suppose $|n| \geq 2^{w-1}$; then we have

$$|f_w(n)| = \left|\frac{n-r}{2^w}\right| \leq \left|\frac{n}{2^w}\right| + \left|\frac{r}{2^w}\right| < \left|\frac{n}{2^w}\right| + \frac{1}{2}.$$

Now,

$$\frac{1}{2} = \frac{2^{w-1}}{2^w} \leq \frac{|n|}{2^w},$$

and, since $w \geq 2$, it follows that

$$|f_w(n)| < 2\left|\frac{n}{2^w}\right| = \left|\frac{n}{2^{w-1}}\right| < |n| .$$

So, the sequence formed by taking the absolute value of the variable n during the execution of NAF$_w(n)$ is strictly decreasing. Thus, the variable n must reach 0, and so NAF$_w(n)$ terminates for all $n \in \mathbb{Z}$.

Now we argue that the algorithm is correct.

Proposition 2.2. *Let α be the string returned by NAF$_w(n)$ where $n \in \mathbb{Z}$. Then $(\alpha)_2$ is a w-NAF and $(\alpha)_2 = n$.*

Proof. By the definition of g_w, it is clear that $(\alpha)_2$ is a w-NAF, so we just have to show that $(\alpha)_2 = n$. If i is a nonnegative integer, we write ${f_w}^i$ to denote i applications of the map f_w; that is,

$${f_w}^i = \underbrace{f_w \circ f_w \circ \cdots \circ f_w}_{i} .$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Since $\mathrm{NAF}_w(n)$ terminates, there is some integer $i \geq 0$ such that ${f_w}^i(n) = 0$. We will argue by induction on i.

When $i = 0$, ${f_w}^i(n) = 0$ implies $n = 0$. For $n = 0$, we have $\alpha = \epsilon$ and then $n = (\alpha)_2$ as required. Suppose $i > 0$. Let $n' = f_w(n)$ and let α' be the string returned by $\mathrm{NAF}_w(n')$. Note that ${f_w}^{i-1}(n') = {f_w}^i(n) = 0$ and so by induction we have that $n' = (\alpha')_2$. By the definition of Algorithm 2.1 we see that

$$\begin{aligned} & \alpha = \alpha' \| g_w(n) \\ \implies & (\alpha)_2 = (\alpha' \| g_w(n))_2 \\ \implies & (\alpha)_2 = 2^{|g_w(n)|}(\alpha')_2 + (g_w(n))_2 \\ \implies & (\alpha)_2 = 2^{|g_w(n)|} n' + (g_w(n))_2 \\ \implies & (\alpha)_2 = 2^{|g_w(n)|} f_w(n) + (g_w(n))_2. \end{aligned} \tag{2.3}$$

From (2.1), we see that the function f_w can be defined in terms of g_w as follows:

$$f_w(n) = \frac{n - (g_w(n))_2}{2^{|g_w(n)|}}.$$

Thus, the right-hand side of (2.3) equals n, and so $(\alpha)_2 = n$ as required. □

Because of Propositions 2.1 and 2.2, we now know that each integer n has a unique w-NAF. Henceforth, we will refer to this representation as *the* w-NAF of n.

2.3. **Length.** We show that the length of the w-NAF of n is at most one digit longer than the $\{0, 1\}$-radix 2 representation of $|n|$. This fact seems to have been first stated, without proof, by Möller [12]. A more general result is proved in Section 4; however, we feel it is of interest to provide a direct proof here.

We start with a lemma.

Lemma 2.3. *Let $(a_{\ell-1} \cdots a_1 a_0)_2$ be a w-NAF of length ℓ where $\ell \geq 1$. If $n = (a_{\ell-1} \cdots a_1 a_0)_2$, then $n > 0$ if and only if $a_{\ell-1} > 0$.*

Proof. Note that since the length of $(a_{\ell-1} \cdots a_1 a_0)_2$ is ℓ we have $a_{\ell-1} \neq 0$. We argue by induction on ℓ. The result is clearly true when $\ell = 1$. Suppose $\ell > 1$.

If $a_0 = 0$ we let $n' = (a_{\ell-1} \cdots a_1)_2$. Then we see that

$$n > 0 \iff 2n' > 0 \iff n' > 0 \iff a_{\ell-1} > 0,$$

where the last equivalence follows by induction.

If $a_0 \neq 0$, then

$$a_{\ell-1} \cdots a_w \cdots a_1 a_0 = a_{\ell-1} \cdots a_w 0 \cdots 0 a_0,$$

since $(a_{\ell-1} \cdots a_1 a_0)_2$ is a w-NAF. Thus,

$$n = 2^w (a_{\ell-1} \cdots a_w)_2 + a_0 \quad \text{where } -2^{w-1} < a_0 \leq 2^{w-1}.$$

Since $a_{\ell-1} \neq 0$, we have $(a_{\ell-1} \cdots a_w)_2 \neq 0$; thus

$$n > 0 \iff (a_{\ell-1} \cdots a_w)_2 > 0 \iff a_{\ell-1} > 0,$$

where the last equivalence follows by induction. This proves the result. □

Proposition 2.4. *For any integers n, w, where $w \geq 2$, the length of the w-NAF of n is at most one digit longer than the binary representation of $|n|$.*

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Proof. Let ℓ be the length of the w-NAF of $|n|$ and let m be the length of the binary representation of $|n|$. If $n = 0$, then $\ell = m = 0$, and so the result is true. Suppose $n \neq 0$. Let $(a_{\ell-1} \cdots a_1 a_0)_2$ be the w-NAF of $|n|$. Since $|n|$ is positive, by Lemma 2.3 we have that $a_{\ell-1} \geq 1$. Let $a = -(2^{w-1} - 1)$. Note that

$$\begin{aligned}
&(a_{\ell-1}a_{\ell-2} \cdots a_1 a_0)_2 = |n| \\
\Longrightarrow\ &(1a_{\ell-2} \cdots a_1 a_0)_2 \leq |n| \\
\Longrightarrow\ &(1\underbrace{00\cdots 0a}_{w}\underbrace{00\cdots 0a}_{w}\cdots)_2 \leq |n| \\
\Longrightarrow\ &(\underbrace{11\cdots 1}_{w}\underbrace{aa\cdots a}_{w}\underbrace{aa\cdots a}_{w}\cdots)_2 \leq (2^{w-1} + \cdots + 2^2 + 2^1 + 1)\,|n|\,.
\end{aligned}$$

Consider the representation on the left-hand side of this last inequality. Reading from left to right, its digits consist of a run of ones, followed by a run of a's, ended by a (possibly empty) run of zeros. If we replace this run of zeros with a run of a's, then we have

$$\begin{aligned}
&(\underbrace{11\cdots 1}_{w}\underbrace{aa\cdots a}_{\ell-1})_2 \leq (2^{w-1} + \cdots + 2^2 + 2^1 + 1)\,|n| \\
\Longrightarrow\ &2^{\ell-1}(2^w - 1) + a(2^{\ell-1} - 1) \leq (2^w - 1)\,|n| \\
\Longrightarrow\ &2^{\ell-1}(2^w - 1) - (2^{w-1} - 1)(2^{\ell-1} - 1) \leq (2^w - 1)\,|n| \\
\Longrightarrow\ &2^{\ell-1} - \frac{2^{w-1} - 1}{2^w - 1}(2^{\ell-1} - 1) \leq |n| \\
\Longrightarrow\ &2^{\ell-1} - \frac{1}{2}(2^{\ell-1} - 1) < |n| \\
\Longrightarrow\ &2^{\ell-2} + \frac{1}{2} < |n| \\
\Longrightarrow\ &2^{\ell-2} < |n|\,.
\end{aligned}$$

Now, from the binary representation of $|n|$, we have $|n| < 2^m$; thus

$$\begin{aligned}
&2^{\ell-2} < 2^m \\
\Longrightarrow\ &\ell - 2 < m \\
\Longrightarrow\ &\ell - m \leq 1.
\end{aligned}$$

This gives us the required result. □

3. Minimality

The main topic of this section is to prove that the w-NAF has a minimal number of nonzero digits; that is, we want to show that no other representation of an integer, with digits in D_w, has fewer nonzero digits than its w-NAF. We begin with a discussion of addition of representations. We will see that the properties of addition provide a key step in our proof of minimality.

For any $\alpha \in {D_w}^*$ and $c_0 \in \mathbb{Z}$ with $|c_0| < 2^{w-1}$, we show that there exists some $\beta \in {D_w}^*$ such that $(\beta)_2 = (\alpha)_2 + c_0$ and

$$\mathsf{wt}(\beta) \leq \mathsf{wt}(\alpha) + 1. \tag{3.1}$$

We do so by developing a certain algorithm for addition.

Given α and c_0, we want to compute a representation $\beta \in {D_w}^*$ with $(\beta)_2 = (\alpha)_2 + c_0$. Let $\alpha = \cdots a_2 a_1 a_0$ and $\beta = \cdots b_2 b_1 b_0$. To compute the sum we define a

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

sequence of integers $c_1, c_2, \dots$. Writing our variables in the following array suggests how these values are related:

$$\begin{array}{r} \cdots \overset{c_3}{a_3}\overset{c_2}{a_2}\overset{c_1}{a_1}a_0 \\ +\underline{\qquad\qquad c_0} \\ \cdots b_3b_2b_1b_0 \end{array}$$

Starting with $i = 0$, we examine a_i and c_i and then assign values to b_i and c_{i+1}. The following rules are used to define b_i:

$a_i \bmod 2$	$c_i \bmod 2$	b_i
0	0	a_i
0	1	c_i
1	0	a_i
1	1	0

Furthermore, c_{i+1} is always set to the value $(a_i + c_i - b_i)/2$.

We claim the representation β is in ${D_w}^*$. To justify this claim, we first show that each c_{i+1} satisfies $|c_{i+1}| < 2^{w-1}$. Note that $b_i \in \{0, a_i, c_i\}$. Since $a_i \in D_w$, we have $|a_i| < 2^{w-1}$, and by induction $|c_i| < 2^{w-1}$. Thus,

$$\begin{aligned} b_i = 0 &\implies c_{i+1} = \frac{a_i + c_i}{2} \implies |c_{i+1}| < 2^{w-1}, \\ b_i = a_i &\implies c_{i+1} = \frac{c_i}{2} \implies |c_{i+1}| < 2^{w-2}, \\ b_i = c_i &\implies c_{i+1} = \frac{a_i}{2} \implies |c_{i+1}| < 2^{w-2}. \end{aligned}$$

Now, it is easy to see that $\beta \in {D_w}^*$. If b_i equals 0 or a_i, then clearly $b_i \in D_w$. If $b_i = c_i$, then, according to our rules, it must be that c_i is odd. Since c_i is odd and $|c_i| < 2^{w-1}$, $c_i \in D_w$. Hence, $b_i \in D_w$ for all i.

Based on the grade-school method of addition, it may seem natural for the rules that define b_i to be implemented inside an appropriate "for" loop. However, for our purposes, it is more convenient if we take a different approach. We first initialize the string $\beta = \cdots b_2b_1b_0$ to equal $\alpha = \cdots a_2a_1a_0$ and then correct the digits of β as necessary. Here is a description of this process in pseudocode:

```
**Algorithm3.1:** ADD-DIGIT($\alpha, c_0$)

comment: $\alpha = \cdots a_2a_1a_0,\ a_i \in D_w,\ |c_0| < 2^{w-1}$
$\cdots b_2b_1b_0 \leftarrow \cdots a_2a_1a_0$
$i \leftarrow 0$
while $c_i \neq 0$
  do
    $(a, c) \leftarrow (a_i, c_i) \bmod 2$
    if $(a, c) = (0, 1)$
      then $b_i \leftarrow c_i$
    else if $(a, c) = (1, 1)$
      then $b_i \leftarrow 0$
    $c_{i+1} \leftarrow (a_i + c_i - b_i)/2$
    $i \leftarrow i + 1$
return $\beta = \cdots b_2b_1b_0$
```

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

For the input $\alpha = \cdots a_2a_1a_0$, let $\ell - 1$ be the largest value of i such that $a_i \neq 0$ (thus, the representation $(\alpha)_2$ has length ℓ). By convention, we let $a_i = 0$ for all $i \geq \ell$. The algorithm terminates if and only if the sequence $c_0, c_1, c_2, \ldots$ reaches zero. If none of $c_0, c_1, \ldots, c_\ell$ are equal to zero, then certainly one of $c_{\ell+1}, c_{\ell+2}, \ldots$ will be; this is because for $i \geq \ell$, c_{i+1} is equal to either 0 or $c_i/2$. Thus, we see that ADD-DIGIT(α, c_0) always terminates.

A short example helps illustrate how the algorithm works. Let $w = 4$. Then $D_4 = \{0, \pm1, \pm3, \pm5, \pm7\}$. Suppose $\alpha = 13570001357$ and $c_0 = 6$. Then the algorithm adds $(13570001357)_2$ and 6 as follows:

$$\begin{array}{r} {\scriptstyle 0\,1\,2\,4\,3} \\ 13570001357 \\ +\underline{6} \\ 13570011307 \end{array}$$

It is interesting to note that ADD-DIGIT computes a sum in ${D_w}^*$ without using the operator "mods 2^w".

The following lemma verifies that the algorithm is correct.

Lemma 3.1. *Let β be the string returned by* ADD-DIGIT(α, c_0) *where $\alpha \in {D_w}^*$ and $|c_0| < 2^{w-1}$. Then, $(\beta)_2 = (\alpha)_2 + c_0$.*

Proof. Let i^* be the value of i when ADD-DIGIT(α, c_0) returns. We argue by induction on i^*. If $i^* = 0$, then

$$\begin{aligned} i^* = 0 &\implies c_0 = 0 \text{ and } \beta = \alpha \\ &\implies (\beta)_2 = (\alpha)_2 + c_0. \end{aligned}$$

So, the result holds for $i^* = 0$.

Suppose $i^* > 0$. From the strings $\alpha = \cdots a_2a_1a_0$ and $\beta = \cdots b_2b_1b_0$, we define $\alpha' = \cdots a_2a_1$ and $\beta' = \cdots b_2b_1$. Let $c_1, c_2, \ldots$ be the sequence of carries which occurs during the computation of ADD-DIGIT(α, c_0). From the description of Algorithm 3.1, we see that ADD-DIGIT(α', c_1) must return the string β'. Moreover, the value of i when ADD-DIGIT(α', c_1) returns must be $i^* - 1$. Now,

$$\begin{aligned} & (\beta')_2 = (\alpha')_2 + c_1 \qquad \text{(by induction)} \\ \implies & (\beta' \| 0)_2 = (\alpha' \| 0)_2 + 2c_1 \\ \implies & (\beta' \| 0)_2 + b_0 = (\alpha' \| 0)_2 + 2c_1 + b_0 \\ \implies & (\beta)_2 = (\alpha' \| 0)_2 + a_0 + c_0 \qquad \text{(since } c_1 = (a_0 + c_0 - b_0)/2\text{)} \\ \implies & (\beta)_2 = (\alpha)_2 + c_0. \end{aligned}$$

□

Returning to (3.1), if we are given $\alpha \in {D_w}^*$ and c_0, with $|c_0| < 2^{w-1}$, we can use ADD-DIGIT(α, c_0) to compute a string $\beta \in {D_w}^*$ such that $(\beta)_2 = (\alpha)_2 + c_0$. We will show that $\mathsf{wt}(\beta) \leq \mathsf{wt}(\alpha) + 1$.

Lemma 3.2. *Let β be the string returned by* ADD-DIGIT(α, c_0) *where $\alpha \in {D_w}^*$ and $|c_0| < 2^{w-1}$. Then, $\mathsf{wt}(\beta) \leq \mathsf{wt}(\alpha) + 1$.*

Proof. Let i^* be the value of i when ADD-DIGIT(α, c_0) returns. Consider the sequence $t_0, t_1, \ldots, t_{i^*}$ where

$$t_i = \mathsf{wt}(\cdots a_{i+1}a_i) + \mathsf{wt}(c_i) + \mathsf{wt}(b_{i-1} \cdots b_1b_0)\,.$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

We argue that this sequence is monotonically decreasing. Once we establish this fact, we show that the lemma follows.

First, note that from the description of Algorithm 3.1, we have

$$c_{i+1} = \frac{a_i + c_i - b_i}{2}, \quad \text{for} \quad 0 \le i \le i^* - 1.$$

Now, $b_i \in \{0, a_i, c_i\}$, and after applying a simple case analysis to this equality we can conclude that

$$\mathsf{wt}(c_{i+1}) \le \mathsf{wt}(a_i) + \mathsf{wt}(c_i) - \mathsf{wt}(b_i)\,. \tag{3.2}$$

For example, suppose $b_i = 0$. Then we must show that $\mathsf{wt}(c_{i+1}) \le \mathsf{wt}(a_i) + \mathsf{wt}(c_i)$. Since $i < i^*$, we have $c_i \ne 0$ and thus $\mathsf{wt}(c_i) = 1$. Now,

$$\mathsf{wt}(c_{i+1}) \le 1 = \mathsf{wt}(c_i) \le \mathsf{wt}(a_i) + \mathsf{wt}(c_i)\,.$$

The other cases are argued in a similar manner.

For $0 \le i \le i^* - 1$ we need to show that $t_i \ge t_{i+1}$. If we compare

$$t_i = \mathsf{wt}(\cdots a_{i+1}a_i) + \mathsf{wt}(c_i) + \mathsf{wt}(b_{i-1} \cdots b_1 b_0)$$

to

$$t_{i+1} = \mathsf{wt}(\cdots a_{i+2}a_{i+1}) + \mathsf{wt}(c_{i+1}) + \mathsf{wt}(b_i \cdots b_1 b_0)$$

and eliminate equal digits, we see that

$$t_i \ge t_{i+1} \iff \mathsf{wt}(a_i) + \mathsf{wt}(c_i) \ge \mathsf{wt}(c_{i+1}) + \mathsf{wt}(b_i)\,.$$

However, this last inequality holds by (3.2). Thus, the sequence of t_i's is monotonically decreasing.

Since $t_0 \ge t_1 \ge \cdots \ge t_{i^*}$ we have $t_0 \ge t_{i^*}$. Note that

$$t_0 = \mathsf{wt}(\cdots a_1 a_0) + \mathsf{wt}(c_0) = \mathsf{wt}(\alpha) + \mathsf{wt}(c_0)\,.$$

Since $b_i = a_i$, for $i \ge i^*$, and $c_{i^*} = 0$ we have

$$t_{i^*} = \mathsf{wt}(\cdots a_{i^*+1}a_{i^*}) + \mathsf{wt}(c_{i^*}) + \mathsf{wt}(b_{i^*-1} \cdots b_1 b_0) = \mathsf{wt}(\beta)\,.$$

Thus, from $t_{i^*} \le t_0$, we can conclude that

$$\mathsf{wt}(\beta) \le \mathsf{wt}(\alpha) + \mathsf{wt}(c_0) \le \mathsf{wt}(\alpha) + 1.$$

□

Now we have all the tools we need to proceed with our main result.

Theorem 3.3. *If $(\alpha)_2$ is a w-NAF, then for any $\beta \in {D_w}^*$ with $(\beta)_2 = (\alpha)_2$, we have $\mathsf{wt}(\alpha) \le \mathsf{wt}(\beta)$.*

Proof. Suppose the result is false. Then for some w-NAF, $(\alpha)_2$, there exists $\beta \in {D_w}^*$ with $(\beta)_2 = (\alpha)_2$ and $\mathsf{wt}(\alpha) > \mathsf{wt}(\beta)$. Choose $(\alpha)_2$ so that its length is minimal. Any w-NAF with length less than that of $(\alpha)_2$ must have a minimal number of nonzero digits.

Let $\alpha = \cdots a_2 a_1 a_0$ and $\beta = \cdots b_2 b_1 b_0$. If $a_0 = b_0$, then $(\cdots a_2 a_1)_2 = (\cdots b_2 b_1)_2$ and so $(\cdots a_2 a_1)_2$ is a shorter counterexample. However, this contradicts our choice of $(\alpha)_2$, so it must be that $a_0 \ne b_0$. A consequence of this is that $(\alpha)_2$ must be odd, since otherwise $a_0 = b_0 = 0$. Hence, both a_0 and b_0 are nonzero.

Since $(\alpha)_2$ is a w-NAF we have

$$\alpha = \cdots a_w 0 0 \cdots 0 a_0.$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Write

$$\alpha = \alpha_1 \| \overbrace{00\cdots 0a_0}^{w} \qquad \text{and} \qquad \beta = \beta_1 \| \overbrace{b_{w-1}\cdots b_1 b_0}^{w}$$

where $\alpha_1, \beta_1 \in {D_w}^*$. Note that since $(\alpha)_2$ is a w-NAF, so is $(\alpha_1)_2$, and furthermore, $\mathsf{wt}(\alpha) = \mathsf{wt}(\alpha_1) + 1$.

We show that at least two of the digits in the string $b_{w-1}\cdots b_1 b_0$ must be nonzero. Suppose not; then all of the digits $b_{w-1}\cdots b_1 b_0$ are zero except for b_0, and so

$$\begin{aligned} (\alpha)_2 &= (\beta)_2 \\ \implies (\alpha_1\|00\cdots 0a_0)_2 &= (\beta_1\|00\cdots 0b_0)_2 \\ \implies a_0 &\equiv b_0 \pmod{2^w} \\ \implies a_0 &= b_0 \qquad (\text{since } a_0, b_0 \in D_w). \end{aligned}$$

But this is a contradiction since a_0 and b_0 cannot be equal. So $\mathsf{wt}(b_{w-1}\cdots b_1 b_0) \geq 2$, and hence $\mathsf{wt}(\beta) \geq \mathsf{wt}(\beta_1) + 2$.

Now,

$$\begin{aligned} (\alpha)_2 &= (\beta)_2 \\ \implies (\alpha_1\|00\cdots 0a_0)_2 &= (\beta_1\|b_{w-1}\cdots b_1 b_0)_2 \\ \implies (\alpha_1)_2\cdot 2^w + (00\cdots 0a_0)_2 &= (\beta_1)_2 \cdot 2^w + (b_{w-1}\cdots b_1 b_0)_2 \\ \implies (\alpha_1)_2 &= (\beta_1)_2 + \frac{(b_{w-1}\cdots b_1 b_0)_2 - (00\cdots 0a_0)_2}{2^w}. \end{aligned}$$

Let $c_0 = \left((b_{w-1}\cdots b_1 b_0)_2 - (00\cdots 0a_0)_2\right)/2^w$. Note that c_0 must be an integer. We can derive a bound on $|c_0|$. Every digit in D_w has absolute value at most $2^{w-1}-1$; thus

$$\begin{aligned} |(b_{w-1}\cdots b_1 b_0)_2| &\leq (2^{w-1}-1)(2^w - 1), \text{ and} \\ |(00\cdots 0a_0)_2| &\leq 2^{w-1} - 1. \end{aligned}$$

Combining these two inequalities gives

$$|(b_{w-1}\cdots b_1 b_0)_2 - (00\cdots 0a_0)_2| \leq (2^{w-1}-1)2^w$$

and thus $|c_0| \leq 2^{w-1} - 1$, or equivalently, $|c_0| < 2^{w-1}$.

So, we have $(\alpha_1)_2 = (\beta_1)_2 + c_0$. Let ${\beta_1}'$ denote the string returned by ADD-DIGIT(β_1, c_0). Then $({\beta_1}')_2 = (\beta_1)_2 + c_0$ and, by Lemma 3.2, $\mathsf{wt}({\beta_1}') \leq \mathsf{wt}(\beta_1) + 1$.

Now, we come to the end of the proof. We have

$$\begin{aligned} &\mathsf{wt}(\alpha) > \mathsf{wt}(\beta) \\ \implies &\mathsf{wt}(\alpha_1) + 1 > \mathsf{wt}(\beta) \qquad (\text{since } \mathsf{wt}(\alpha) = \mathsf{wt}(\alpha_1) + 1) \\ \implies &\mathsf{wt}(\alpha_1) + 1 > \mathsf{wt}(\beta_1) + 2 \qquad (\text{since } \mathsf{wt}(\beta) \geq \mathsf{wt}(\beta_1) + 2) \\ \implies &\mathsf{wt}(\alpha_1) > \mathsf{wt}(\beta_1) + 1 \\ \implies &\mathsf{wt}(\alpha_1) > \mathsf{wt}({\beta_1}') \qquad (\text{since } \mathsf{wt}(\beta_1) + 1 \geq \mathsf{wt}({\beta_1}')). \end{aligned}$$

But, $(\alpha_1)_2 = ({\beta_1}')_2$ and $(\alpha_1)_2$ is a w-NAF. Thus $(\alpha_1)_2$ is a shorter counterexample, contrary to our choice of $(\alpha)_2$. This proves the result. □

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

4. Length of minimal weight representations

We have already seen that the length of the w-NAF of an integer is at most one digit longer than its binary representation. In this section, we see that this property is actually a consequence of a more general result. We will show that the length of *any* representation in ${D_w}^*$ with a *minimal number of nonzero digits* is at most one digit longer than its binary representation.

Theorem 4.1. *Let $\alpha = a_{\ell-1} \cdots a_1 a_0$ be a string in ${D_w}^*$ such that $a_{\ell-1} \neq 0$ and $(\alpha)_2 = n$. If $\mathsf{wt}(\alpha) \leq \mathsf{wt}(\beta)$ for any $\beta \in {D_w}^*$ with $(\beta)_2 = n$, then $\ell \leq \lfloor \lg |n| \rfloor + 2$.*

Proof. We argue by induction on ℓ, the length of $(\alpha)_2$. Note that since $a_{\ell-1}$ is nonzero the length of $(\alpha)_2$ cannot be zero (i.e., $\ell \geq 1$). Also, $a_{\ell-1} \neq 0$ tells us that $n = (\alpha)_2 \neq 0$ and so $\lg |n|$ is defined.

If $\ell = 1$, then

$$\ell = 1 \leq \lfloor \lg |n| \rfloor + 1 < \lfloor \lg |n| \rfloor + 2,$$

and so the result is true.

Suppose now that $\ell > 1$. Let $\alpha_1 = a_{\ell-1} \cdots a_2 a_1$, so that $\alpha = \alpha_1 \| a_0$. Note that

$$\frac{n - a_0}{2} = (\alpha_1)_2.$$

Since $(\alpha)_2$ is a minimal Hamming weight representation of n, $(\alpha_1)_2$ must be a minimal Hamming weight representation of $(n - a_0)/2$. The length of $(\alpha_1)_2$ is $\ell - 1$, so by induction we have

$$\begin{aligned} & \ell - 1 \leq \lfloor \lg |(n - a_0)/2| \rfloor + 2 \\ \Longrightarrow\ & \ell - 1 \leq \lfloor \lg |n - a_0| \rfloor - 1 + 2 \\ \Longrightarrow\ & \ell \leq \lfloor \lg |n - a_0| \rfloor + 2. \end{aligned}$$

If $\lfloor \lg |n - a_0| \rfloor \leq \lfloor \lg |n| \rfloor$, then from the previous step we can conclude

$$\ell \leq \lfloor \lg |n| \rfloor + 2,$$

which is the result we want. Thus, we can restrict to the case $\lfloor \lg |n - a_0| \rfloor > \lfloor \lg |n| \rfloor$. In this case, it is clear that $a_0 \neq 0$.

The string α contains at least two nonzero digits, namely $a_{\ell-1}$ and a_0. This tells us that $\mathsf{wt}(\alpha) \geq 2$. Because a_0 is nonzero and $n = (\alpha)_2$, we have also that n is odd. The integer n cannot be equal to any of the digits in D_w. To see this, suppose $n \in D_w$. Then the string $\beta = n$ is in ${D_w}^*$ and $(\beta)_2 = n$. However, $\mathsf{wt}(\beta) = 1$ which is less than $\mathsf{wt}(\alpha) \geq 2$, contrary to our hypothesis. Thus, $|n| > 2^{w-1}$. A consequence of this is that n and $n - a_0$ are either both positive or both negative.

We will suppose n is positive (the case where n is negative is argued in the same manner). Now,

$$\begin{aligned} \lfloor \lg n \rfloor < \lfloor \lg (n - a_0) \rfloor & \Longrightarrow \lfloor \lg n \rfloor + 1 \leq \lfloor \lg (n - a_0) \rfloor \\ & \Longrightarrow \lfloor \lg n \rfloor + 1 \leq \lg (n - a_0) \\ & \Longrightarrow 2^{\lfloor \lg n \rfloor + 1} \leq n - a_0, \end{aligned}$$

so we see that the closed interval $[n, 2^{\lfloor \lg n \rfloor + 1}]$ sits inside the closed interval $[n, n - a_0]$. Let $d = 2^{\lfloor \lg n \rfloor + 1} - n$. Since n is odd, d is odd. The intervals $[n, 2^{\lfloor \lg n \rfloor + 1}]$ and $[n, n - a_0]$ have lengths d and $|a_0|$, respectively. By comparing these lengths, we see that $d \leq |a_0| < 2^{w-1}$. Thus, $d \in D_w$.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Consider the string $\beta \in {D_w}^*$ where

$$\beta = \underbrace{100 \cdots 0d}_{\lfloor \lg n \rfloor + 2}.$$

Then, $(\beta)_2 = 1 \cdot 2^{\lfloor \lg n \rfloor + 1} + d \cdot 2^0 = n$. So, n can be represented using only 2 nonzero digits. Thus, $\mathsf{wt}(\alpha) \leq 2$. Now, both $\mathsf{wt}(\alpha) \geq 2$ and $\mathsf{wt}(\alpha) \leq 2$, and so $\mathsf{wt}(\alpha) = 2$. Thus, the string α has the form

$$\alpha = a_{\ell-1} 00 \cdots 0 a_0.$$

Now, $(\alpha)_2 = n$ and so $a_{\ell-1} \cdot 2^{\ell-1} + a_0 = n$. Since n is positive and $\lfloor \lg n \rfloor < \lfloor \lg(n - a_0) \rfloor$ it must be that a_0 is negative. However, if a_0 is negative, then $a_{\ell-1}$ must be positive. Thus,

$$\begin{aligned} & a_{\ell-1} \cdot 2^{\ell-1} = n - a_0 \\ \Longrightarrow\ & 2^{\ell-1} \leq n - a_0 \\ \Longrightarrow\ & \ell - 1 \leq \lfloor \lg(n - a_0) \rfloor \\ \Longrightarrow\ & \ell \leq \lfloor \lg(n - a_0) \rfloor + 1. \end{aligned}$$

If $\lfloor \lg(n - a_0) \rfloor \leq \lfloor \lg n \rfloor + 1$, then this gives us $\ell \leq \lfloor \lg n \rfloor + 2$, which is the desired result. To finish the proof, we show that $\lfloor \lg(n - a_0) \rfloor > \lfloor \lg n \rfloor + 1$ would lead to a contradiction. Observe

$$\begin{aligned} & \lfloor \lg n \rfloor + 1 < \lfloor \lg(n - a_0) \rfloor \\ \Longrightarrow\ & \lg n < \lfloor \lg(n - a_0) \rfloor \quad (\text{since } x < \lfloor x \rfloor + 1) \\ \Longrightarrow\ & \lg n + 1 \leq \lg(n - a_0) \\ \Longrightarrow\ & 2n \leq n - a_0 \\ \Longrightarrow\ & n \leq -a_0 \\ \Longrightarrow\ & n \in D_w \quad (\text{since } n \text{ is odd}), \end{aligned}$$

contradicting the fact that $\mathsf{wt}(\alpha) = 2$. This concludes our proof. □

Note that Proposition 2.4, which we proved directly in Section 2, can now be obtained as a consequence of Theorem 3.3 and Theorem 4.1.

5. Lexicographic characterization

For an integer n, consider the set of all representations of n with digits in D_w. We can compare representations in this set in a number of ways. For example, we can order representations according to how many nonzero digits they have. By Theorem 3.3, we know that the w-NAF is a minimal representation under this order, but it is not necessarily unique in this respect. For example, when $w = 3$, the 3-NAF of 5 is $(100\overline{3})_2$ which has two nonzero digits, and so too do the representations $(101)_2, (13)_2$ and $(3\overline{1})_2$. However, there is another comparison we can make between representations which does, in fact, uniquely identify the w-NAF. This comparison is based on the *position* of nonzero digits and we introduce it now.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

From any string $\alpha \in {D_w}^*$, we can derive a string $\alpha' \in \{0,1\}^*$ defined as follows: if $\alpha = \cdots a_2 a_1 a_0$, then $\alpha' = \cdots {a_2}' {a_1}' {a_0}'$ where

$$ {a_i}' := \begin{cases} 0 & \text{if } a_i = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{5.1} $$

For example, if $\alpha = 300\bar{3}0$, then $\alpha' = 10010$. For $\alpha, \beta \in {D_w}^*$ we write $\alpha \preceq \beta$ if α' is less than or equal to β' under a *right-to-left* lexicographic ordering. If $\beta = 1003\bar{1}0$, then $\beta' = 100110$ and, after comparing α' to β', we see that $\alpha \preceq \beta$.

The relation "$\preceq$" induces an order on the set of representations of n with digits in D_w. Each representation of n identifies a unique string in ${D_w}^*$ that does not have any leading zeros. Two representations of n are compared by applying the relation "$\preceq$" to their corresponding strings. For example, suppose $w = 3$ and $n = 42$. Below, we list a number of strings in ${D_3}^*$ which correspond to representations of 42. For each string, we give the associated string in $\{0,1\}^*$. The list is sorted under the relation "$\preceq$".

	3	0	0	-3	0		1	0	0	1	0
1	1	0	0	-3	0	1	1	0	0	1	0
3	-3	0	0	-3	0	1	1	0	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0
	1	3	0	1	0		1	1	0	1	0
	3	-1	0	1	0		1	1	0	1	0
1	0	0	3	-1	0	1	0	0	1	1	0
	3	0	-3	3	0		1	0	1	1	0
	3	0	-1	-1	0		1	0	1	1	0
		3	3	3	0			1	1	1	0

Notice that the 3-NAF of 42 is the unique smallest representation of the ones considered. Even if we considered *all* the representations of 42 with digits in D_3, the 3-NAF of 42 would still be the unique smallest representation. This result is true in general and is proven in Theorem 5.1.

Theorem 5.1. *Let n be an integer. Of all the representations of n with digits in D_w, the w-NAF of n is the unique smallest representation under the order $\preceq$.*

Proof. When $n = 0$, the only representation of n with digits in D_w is the all-zero representation. The all-zero representation is the w-NAF of 0, so the result is true for $n = 0$. Suppose the result is false for some $n \neq 0$. Choose n so that the length of the w-NAF of n is minimal. Let $(\alpha)_2$ be the w-NAF of n. There is some string $\beta \in {D_w}^*, \beta \neq \alpha$, such that $n = (\beta)_2$ and $\beta \preceq \alpha$.

Recall the definition of α' and β' from (5.1). If n is even, then ${a_0}' = {b_0}' = 0$ and so the result is also false for $n/2$, contrary to our choice of n. Thus, n is odd and so ${a_0}' = {b_0}' = 1$. Since α is a w-NAF, ${a_{w-1}}' = {a_{w-2}}' = \cdots = {a_1}' = 0$, and since $\beta \preceq \alpha$, we have ${b_{w-1}}' = {b_{w-2}}' = \cdots = {b_1}' = 0$. Thus

$$ \beta = \cdots b_w 00 \cdots 0 b_0, $$
$$ \alpha = \cdots a_w 00 \cdots 0 a_0. $$

Since $(\alpha)_2 = (\beta)_2$, we have $a_0 \equiv b_0 \pmod{2^w}$. However, $a_0, b_0 \in D_w$, so it must be that $a_0 = b_0$. But this contradicts our choice of n since we see the result is also false for $(n - a_0)/2^w$. Hence, we have the desired result. $\square$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

6. Comments and further work

A detailed discussion of the costs and benefits of using the w-NAF window method for elliptic curve scalar multiplication, including several examples applied to the NIST recommended elliptic curves, is given in [7, Ch. 3]. Much of this analysis is based on the fact that the average density of nonzero digits among all w-NAFs of length ℓ is approximately $1/(w+1)$ (a proof of a similar result is given in [4]). Because of Theorem 3.3, we now know that no other family of D_w-radix 2 representations can have density lower than that of the w-NAF.

As with the result on the length of the w-NAF, the result presented in Lemma 2.3 can be generalized to any minimal weight D_w-radix 2 representation.

The w-NAF window method for scalar multiplication is described in [2] and [16] as a *left-to-right* method.[2] However, Algorithm 2.1 computes the w-NAF of an integer from right to left. This means that the w-NAF of n must first be computed and stored in its entirety before computations to determine nP can begin. There are other representations which use the same digits as the w-NAF but can be deduced from left to right [1, 13]. Like the w-NAF, these new representations have a minimal number of nonzero digits. These representations result in a left-to-right window method which uses less memory.

The results of this paper further strengthen the analogy between the ordinary NAF and the w-NAF. After the submission of this manuscript, another property of the ordinary NAF was shown to carry over to the w-NAF. In [10], a simple algorithm is described (due to Chang and Tsao-Wu [3]) which constructs the NAF of n by subtracting the binary representation of n from the binary representation of $3n$. A similar construction has been discovered for the w-NAF [8, 14]. Lemma 2.3 and Proposition 2.4 appear to be very natural consequences of this construction.

After the submission of this manuscript, the authors became aware of a related paper by Avanzi [1], which presents an alternate proof of Theorem 3.3. Avanzi's proof considers radix 2 representations where the nonzero digits have absolute value at most 2^{w-1} and can be either odd or even. Even when these additional digits are included, Avanzi's argument shows that the w-NAF still has minimal weight.

Acknowledgments

This paper benefited greatly from the detailed comments of the referee. In particular, we would like to thank the referee for suggesting a more direct way to argue that Algorithm 2.1 terminates. Also, we thank the referee for proposing a shorter proof of Lemma 3.2.

References

1. R. M. Avanzi. A Note on the Sliding Window Integer Recoding and its Left-to-Right Analogue, to appear in "Workshop on Selected Areas in Cryptography – SAC 2004".
2. I. F. Blake, G. Seroussi and N. P. Smart. *Elliptic Curves in Cryptography*, Cambridge University Press, 1999. MR1771549 (2001i:94048)
3. S. H. Chang and N. Tsao-Wu. Distance and Structure of Cyclic Arithmetic Codes, in "Proceedings of the Hawaii International Conference on System Sciences" (1968), 463–466.
4. H. Cohen. Analysis of the Flexible Window Powering Algorithm, Preprint. Available from `http://www.math.u-bordeaux.fr/~cohen/window.dvi`.

[2]The w-NAF can also be employed in a right-to-left method for scalar multiplication. More details on this can be found in [12]. However, the left-to-right method is usually preferred over the right-to-left method since it saves a few operations.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

5. H. Cohen, A. Miyaji and T. Ono. Efficient Elliptic Curve Exponentiation Using Mixed Coordinates, in "Advances in Cryptology – ASIACRYPT '98", *Lecture Notes in Computer Science* **1514** (1998), 51–65. MR1726152
6. D. M. Gordon. A Survey of Fast Exponentiation Methods, *Journal of Algorithms* **27** (1998), 129–146. MR1613189 (99g:94014)
7. D. Hankerson, A. Menezes and S. Vanstone. *Guide to Elliptic Curve Cryptography*, Springer, 2004. MR2054891 (2005c:94049)
8. C. Heuberger, R. Katti, H. Prodinger and X. Ruan. The Alternating Greedy Expansion and Applications to Left-to-Right Algorithms in Cryptography, Preprint. Available from `http://www.wits.ac.za/helmut/paperlst.htm`.
9. K. Koyama and Y. Tsuruoka. Speeding up Elliptic Cryptosystems by Using a Signed Binary Window Method, in "Advances in Cryptology – CRYPTO '92", *Lecture Notes in Computer Science* **740** (1993), 345–357. MR1287864
10. J. H. van Lint, *Introduction to Coding Theory*, 3rd edition, Springer, 1999. MR1664228 (2000a:94001)
11. A. Miyaji, T. Ono and H. Cohen. Efficient Elliptic Curve Exponentiation, in "Information and Communication Security – ICICS '97", *Lecture Notes in Computer Science* **1334** (1997), 282–290.
12. B. Möller. Improved Techniques for Fast Exponentiation, in "Information Security and Cryptology – ICISC 2002", *Lecture Notes in Computer Science* **2587** (2002), 298–312. MR2080830
13. J. A. Muir and D. R. Stinson. New Minimal Weight Representations for Left-to-Right Window Methods, Technical Report CORR 2004-19, Centre for Applied Cryptographic Research. Available from `http://www.cacr.math.uwaterloo.ca/techreports/2004`.
14. K. Okeya, K. Schmidt-Samoa, C. Spahn and T. Takagi. Signed Binary Representations Revisited, in "Advances in Cryptology – CRYPTO 2004", *Lecture Notes in Computer Science* **3152** (2004), 123–139.
15. G. W. Reitwiesner. Binary Arithmetic, in *Advances in Computers, Vol. 1*, Academic Press, 1960, pp. 231–308. MR0122018 (22:12745)
16. J. A. Solinas. Efficient arithmetic on Koblitz curves. *Designs, Codes and Cryptography* **19** (2000), 195–249. MR1759617 (2002k:14039)

Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

E-mail address: `jamuir@uwaterloo.ca`

URL: `http://www.uwaterloo.ca/~jamuir`

School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

E-mail address: `dstinson@uwaterloo.ca`

URL: `http://www.cacr.math.uwaterloo.ca/~dstinson`

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use