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 which 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 lexicographical ordering. We also generalize a result on w-NAF 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 a base 2 (or *radix* 2) positional number system, representations of integers are converted into integers via the rule

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

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 base 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 be to the set of w-NAF digits; that is, D_w is the set of integers which 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 $300\bar{3}0$ (note that $\bar{1}$ denotes -1, $\bar{3}$ denotes -3, etc.) satisfies conditions (1) and (2), and

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

Date: 28 March 2004.

1991 *Mathematics Subject Classification.* Primary 94A60, 11T71, 14G50; Secondary 94B40.

Key words and phrases. efficient representations, minimal Hamming weight, elliptic curve arithmetic.

The second author is supported by NSERC grant RGPIN 203114-02.

When $w = 2$, $D_w = \{0, \pm 1\}$ and the w-NAF coincides with the well known *nonadjacent form* [5]. 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 $n = (a_{\ell-1} \dots a_1 a_0)_2$ 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 [12]). Using this 2-NAF results in 5 elliptic curve addition operations.

The *signed binary sliding window method* [7], 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 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 [3] 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 [8].

In Section 4, we generalize a known result about the length of w-NAFs. It is stated without proof in [10] 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.

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 *lexicographical ordering*. For an integer n, we consider the set of all D_w-radix 2 representations of n. The position 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.** If n is an integer and we write $n = (\dots a_2 a_1 a_0)_2$ then we are expressing n as the sum of an *infinite* series. If there is some ℓ such that $a_i = 0$ for all $i \geq \ell$ then n is the sum of a *finite* series and we indicate this by writing $n = (a_{\ell-1} \dots a_2 a_1 a_0)_2$. If, in addition, $a_{\ell-1} \neq 0$ we say this representation has *length* ℓ.

Definition 1.1. The *length* of a representation $(\dots a_2 a_1 a_0)_2$ is the largest integer ℓ such that $a_{\ell-1} \neq 0$ but $a_i = 0$ for all $i \geq \ell$. The length of the all zero representation is defined to be zero.

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 the representation, $(a_{\ell-1} \dots a_1 a_0)_2$ where each a_i is in D_w, then $a_{\ell-1} \dots a_1 a_0$ 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 $\alpha \| \beta$.

We apply some of our terminology for representations to strings. If $\alpha \in {D_w}^*$ satisfies property (2) then we call α a w-NAF. If, in addition, $(\alpha)_2 = n$, we say α is a w-NAF for n. Notice that if α is a w-NAF for n then α with any leading zeros removed is also a w-NAF for n. We denote the string formed by deleting the leading zeros from α by $\widehat{\alpha}$.

If α is a string of digits then $w(\alpha)$ denotes the number of nonzero digits in α. When $\alpha \in \{0,1\}^*$, $w(\alpha)$ is equal to the Hamming weight of α.

2. Known Results

The w-NAF seems to have been first described by Cohen, Miyaji and Ono [4]. 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 [1] and by Solinas [13].

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.

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 $(a_{\ell-1}\dots a_2a_1a_0)_2$ and $(b_{\ell'-1}\dots b_2b_1b_0)_2$, such that

$$(a_{\ell-1}\dots a_2a_1a_0)_2 = (b_{\ell'-1}\dots b_2b_1b_0)_2,$$

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

$$(a_{\ell-1}\dots a_2a_1)_2 = (b_{\ell'-1}\dots b_2b_1)_2,$$

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

$$\begin{aligned}(a_{\ell-1}\dots a_w00\dots0a_0)_2 &= (b_{\ell'-1}\dots b_w00\dots0b_0)_2\\ &\implies a_0 \equiv b_0 \pmod{2^w}.\end{aligned}$$

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 representation of n which is a w-NAF. Unlike the algorithms in [1] and [13], 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

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

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

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

We will denote this value of r by n mods 2^w. For example, if $w = 3$ then 13 mods $2^w = \bar{3}$. 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:

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

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

In both functions, we define $r = n \text{ mods } 2^w$. Note that f_w returns an integer and g_w returns a string. For example, if $w = 3$, then $f_w(13) = 2$ and $g_w(13) = 00\overline{3}$.

Now we can describe our algorithm:

```
Algorithm 2.1: NAF_w(n)

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

As $\text{NAF}_w(n)$ executes, it builds a string, α, in ${D_w}^*$. And assuming $\text{NAF}_w(n)$ 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 $\text{NAF}_w(n)$ terminates for all $n \in \mathbb{Z}$. Suppose $\text{NAF}_w(n_0)$ fails to terminate for some $n_0 \in \mathbb{Z}$. Choose n_0 so that $|n_0|$ is as small as possible. Note that n_0 must be nonzero since $\text{NAF}_w(0)$ clearly terminates. If n_0 is even then $f_w(n_0) = n_0/2$ and $\text{NAF}_w(n_0/2)$ also fails to terminate, but $|n_0/2| < |n_0|$, contrary to our choice of n_0. Thus, n_0 is odd.

Now, there are integers q and r such that

$$n_0 = q \cdot 2^w + r \quad \text{where } -2^{w-1} < r \leq 2^{w-1}.$$

However, n_0 is odd, so the bound on r can be tightened. This gives:

$$\begin{aligned} n_0 &= q \cdot 2^w + r \quad \text{where } -2^{w-1} < r < 2^{w-1} \\ \implies 1 &= \frac{q}{n_0} \cdot 2^w + \frac{r}{n_0} \quad \text{where } -2^{w-1} < \frac{r}{n_0} < 2^{w-1} \end{aligned}$$

We claim that $0 \leq q/n_0 < 1$. We noted earlier that the first part of this equality holds, so it remains to prove that $q/n_0 < 1$. Suppose to the contrary that $q/n_0 \geq 1$; then we have

$$\begin{aligned} &\frac{q}{n_0} \geq 1 \\ \implies &\frac{q}{n_0} 2^w \geq 2^w \\ \implies &\frac{q}{n_0} 2^w + \frac{r}{n_0} > 2^w - 2^{w-1} \\ \implies &1 > 2^{w-1}. \end{aligned}$$

The last implication tells us that w is less than 1, however this is a contradiction because $w \geq 2$. Hence, $0 \leq q/n_0 < 1$, and thus

$$0 \leq \frac{f_w(n_0)}{n_0} < 1.$$

So, if $n_0 > 0$,

$$0 \leq f_w(n_0) < n_0,$$

and if $n_0 < 0$,

$$0 \geq f_w(n_0) > n_0.$$

In either case, $|f_w(n_0)| < |n_0|$, and the algorithm fails to terminate on input $f_w(n_0)$. Thus, n_0 cannot exist, and so the algorithm terminates for all $n \in \mathbb{Z}$.

Proposition 2.2. *For any $n \in \mathbb{Z}$, let α be the string returned by $NAF_w(n)$. Then α is a w-NAF and $(\alpha)_2 = n$.*

Proof. By the definition of g_w, it is clear that α is a w-NAF, so we just have to show that α is a representation of n. For any $n \in \mathbb{Z}$ there exists a smallest 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 $n = (\alpha)_2$ for all $n \in \mathbb{Z}$ with $i = k$ where $k \geq 0$. Now consider $n \in \mathbb{Z}$ with $i = k + 1$. Let $n' = f_w(n)$ and let α' be the string returned by $\mathrm{NAF}_w(n')$. Note that by the induction hypothesis, $n' = (\alpha')_2$. By the definition of Algorithm 2.1 we have

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

From (2.1), we see 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 bit longer than the $\{0, 1\}$-radix 2 representation of $|n|$. This fact is often mentioned with regards to the 2-NAF but a proof apparently has not appeared in the literature. We will first give a proof for this special case and then generalize the argument to establish the result for all w-NAFs.

We start with a Lemma.

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

Proof. Note that since the length of $(a_{\ell-1} \ldots a_1 a_0)_2$ is ℓ we have $a_{\ell-1} \neq 0$. We argue by induction on ℓ. The result is clearly true when $\ell = 1$, so suppose it is true for all representations with $\ell \leq k$ where $k \geq 1$. Consider a representation with $\ell = k + 1$.

If $a_0 = 0$, let $n' = (a_{\ell-1} \ldots a_1)_2$ and then

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

where the last equivalence follows from the induction assumption.

If $a_0 \neq 0$ then

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

since this representation is a w-NAF. Thus,

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

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

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

This gives us the required result. □

Consider 2-NAFs. Let n be a nonzero integer and let ℓ be the length of the 2-NAF of $|n|$. By Lemma 2.3, the most significant digit of the 2-NAF of $|n|$ is a 1. Thus, we have

$$\begin{aligned}
&(\underbrace{10\bar{1}0\bar{1}0\bar{1}\dots}_{\ell})_2 \leq |n| \\
\implies &(\underbrace{11\overline{111111}\dots}_{\ell+1})_2 \leq (2^1+1)|n| \\
\implies &2^{\ell-1}(2^2-1) - 1 \cdot (2^{\ell-1}-1) \leq 3|n| \\
\implies &2^{\ell}+1 \leq 3|n| \\
\implies &2^{\ell} < 3|n|. \qquad (2.4)
\end{aligned}$$

Let m be the length of the usual binary representation of $|n|$. Then

$$\begin{aligned}
&2^m > |n| \\
\implies &1/2^m < 1/|n|. \qquad (2.5)
\end{aligned}$$

Multiplying (2.4) and (2.5) gives us

$$\begin{aligned}
&2^{\ell-m} < 3 \\
\implies &\ell - m \leq 1.
\end{aligned}$$

So, ℓ is at most one more than m. This is also true for general w-NAFs, which we now prove. This result seems to have been first stated, without proof, by Möller [10].

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 $|n|$.*

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 $n \neq 0$. Since $|n|$ is positive, by Lemma 2.3 we know the most significant digit of the w-NAF of $|n|$ is positive. Let $a = -(2^{w-1}-1)$, then we have

$$\begin{aligned}
&(1\underbrace{00\ldots0a}_{w}\underbrace{00\ldots0a}_{w}\ldots)_2 \le |n| \\
\Longrightarrow\; &(\underbrace{11\ldots1}_{w}\underbrace{aa\ldots a}_{w}\underbrace{aa\ldots a}_{w}\ldots)_2 \le (2^{w-1}+\cdots+2^2+2^1+1)|n| \\
\Longrightarrow\; &2^{\ell-1}(2^w-1)+a(2^{\ell-1}-1) \le (2^w-1)|n| \\
\Longrightarrow\; &2^{\ell-1}(2^w-1)-(2^{w-1}-1)(2^{\ell-1}-1) \le (2^w-1)|n| \\
\Longrightarrow\; &2^{\ell-1}-\frac{2^{w-1}-1}{2^w-1}(2^{\ell-1}-1) \le |n| \\
\Longrightarrow\; &2^{\ell-1}-\frac{1}{2}(2^{\ell-1}-1) < |n| \\
\Longrightarrow\; &2^{\ell-2}+\frac{1}{2} < |n| \\
\Longrightarrow\; &2^{\ell-2} < |n|. \qquad (2.6)
\end{aligned}$$

Now, from the binary representation of $|n|$, we have

$$\begin{aligned}
&2^m > |n| \\
\Longrightarrow\; &1/2^m < 1/|n|. \qquad (2.7)
\end{aligned}$$

Multiplying (2.6) and (2.7), we find

$$\begin{aligned}
&2^{\ell-m-2} < 1 \\
\Longrightarrow\; &\ell-m-2 < 0 \\
\Longrightarrow\; &\ell-m \le 1.
\end{aligned}$$

This gives us the required result. □

3. Minimality of Hamming Weight

The main topic of this section is to prove that the w-NAF has minimal Hamming weight; 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. With this goal in mind, it might seem like a diversion to start with a discussion of addition of representations in ${D_w}^*$. Nevertheless, this is how we begin since 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

$$w(\beta) \le w(\alpha)+1. \qquad (3.1)$$

The integer $(\alpha)_2 + c_0$ has an infinite number of representations in ${D_w}^*$. Most of these representations (an infinite number, in fact) do not satisfy (3.1) but we find one that does 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 $\alpha = \ldots a_2a_1a_0$ and $\beta = \ldots b_2b_1b_0$. To compute the sum we define a sequence of integers $c_1, c_2, \ldots$. Writing our variables in the following array suggests how our algorithm will proceed:

$$\begin{array}{rr}
 & \ldots \overset{c_3}{a_3}\overset{c_2}{a_2}\overset{c_1}{a_1}a_0 \\
+ & c_0 \\
\hline
 & \ldots b_3b_2b_1b_0
\end{array}$$

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

Further, 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.

Here is a description of the algorithm in pseudocode:

```
**Algorithm 3.1:** $D_w$-ADD-DIGIT$(\alpha, c_0)$

comment: $\alpha = \ldots a_2 a_1 a_0,\ a_i \in D_w,\ |c_0| < 2^{w-1}$
$\beta \leftarrow \alpha$
$i \leftarrow 0$
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 $\beta = \ldots b_2 b_1 b_0$
```

For the input $\alpha = \ldots a_2 a_1 a_0$, 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 either 0 or $c_i/2$. Thus, we see that D_w-ADD-DIGIT(α, c_0) always terminates.

A short example helps illustrate how the algorithm works. Let $w = 4$, then $D_w = \{0, \pm 1, \pm 3, \pm 5, \pm 7\}$. Suppose $\alpha = 1357$ and $c_0 = 6$. Then the algorithm adds $(1357)_2$ and 6 as follows:

$$\begin{array}{r} {\scriptstyle 0\,1\,2\,4\,3} \\ 001357 \\ +\underline{6} \\ 11307 \end{array}$$

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

An induction argument can be used to verify that the algorithm is correct.

Lemma 3.1. *Let β be the string returned by D_w-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 D_w-ADD-DIGIT(α, c_0) returns. 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 the result holds for $i^* = k$ where $k \geq 0$. Now, consider the case where $i^* = k + 1$.

Let $\alpha = \ldots a_2 a_1 a_0$ and $\beta = \ldots b_2 b_1 b_0$. From these strings, define $\alpha' = \ldots a_2 a_1$ and $\beta' = \ldots b_2 b_1$. Let $c_1, c_2, \ldots$ be the sequence of carries which occurs during the computation of D_w-ADD-DIGIT(α, c_0). From the description of Algorithm 3.1, we see that D_w-ADD-DIGIT(α', c_1) must return the string β' with $i^* = k$. Now,

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

So, no matter what the value of i^* is, D_w-ADD-DIGIT(α, c_0) correctly computes the sum of $(\alpha)_2$ and c_0. □

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

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

Proof. We proceed as in the proof of Lemma 3.1. Let i^* be the value of i when D_w-ADD-DIGIT(α, c_0) returns. If $i^* = 0$ then

$$\begin{aligned} i^* = 0 &\implies \beta = \alpha \\ &\implies w(\beta) = w(\alpha) \\ &\implies w(\beta) \leq w(\alpha) + 1. \end{aligned}$$

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

Suppose now that the result is false. Then for some α and c_0, we have $w(\beta) > w(\alpha) + 1$. For such α and c_0, it must be that $i^* > 0$. Choose α and c_0 so that i^* is minimal, and let k denote this minimal value.

Let $\alpha = \ldots a_2 a_1 a_0$ and $\beta = \ldots b_2 b_1 b_0$. From these strings, define $\alpha' = \ldots a_2 a_1$ and $\beta' = \ldots b_2 b_1$. Let $c_1, c_2, \ldots$ be the sequence of carries which occurs during the computation of D_w-ADD-DIGIT(α, c_0). Note that D_w-ADD-DIGIT(α', c_1) must return the string β' with $i^* = k-1$.

The digit a_0 must be odd. To see this, suppose a_0 is even. Since $a_0 \in D_w$ and the only even digit in D_w is 0, it must be that $a_0 = 0$. Now consider c_0.

If c_0 is odd, our rules for addition tell us that $b_0 = c_0$. Now,

$$c_1 = \frac{a_0 + c_0 - b_0}{2} = \frac{0 + c_0 - c_0}{2} = 0.$$

Thus, from the description of Algorithm 3.1, we see that $b_i = a_i$ for $i \geq 1$. So,

$$\begin{aligned} \alpha &= \ldots a_2 a_1 0 \\ \beta &= \ldots a_2 a_1 c_0. \end{aligned}$$

But these strings satisfy $w(\beta) \leq w(\alpha) + 1$. This is a contradiction since we chose α and c_0 so that $w(\beta) > w(\alpha) + 1$.

If c_0 is even, then by our rules for addition, $b_0 = a_0$ and we have

$$\begin{aligned} & w(\beta) > w(\alpha) + 1 \\ \Longrightarrow\; & w(\beta' \| b_0) > w(\alpha' \| a_0) + 1 \\ \Longrightarrow\; & w(\beta' \| a_0) > w(\alpha' \| a_0) + 1 \\ \Longrightarrow\; & w(\beta') > w(\alpha') + 1. \end{aligned}$$

Thus α' and c_1 provide a counter-example with $i^* = k-1$; but this contradicts our choice of α and c_0.

So, when a_0 is even, c_0 can be neither even nor odd. Thus, it must be that a_0 is odd and this tells us that $w(\alpha) = w(\alpha') + 1$. Now,

$$\begin{aligned} & w(\beta) > w(\alpha) + 1 \\ \Longrightarrow\; & w(\beta) > w(\alpha') + 1 + 1 \\ \Longrightarrow\; & w(\beta') + 1 > w(\alpha') + 1 + 1 \quad (\text{since } w(\beta') + 1 \geq w(\beta)) \\ \Longrightarrow\; & w(\beta') > w(\alpha') + 1. \end{aligned}$$

But, this contradicts our choice of α and c_0.

Thus, there can be no α and c_0 for which $w(\beta) > w(\alpha) + 1$. □

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

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

Proof. Suppose the result is false. Then for some w-NAF, α, there exists $\beta \in {D_w}^*$ with $(\beta)_2 = (\alpha)_2$ and $w(\alpha) > w(\beta)$. Choose α so that $|\alpha|$ (i.e., the length of α) is minimal. Any w-NAF with length less than $|\alpha|$ must have minimal Hamming weight.

Let $\alpha = \ldots a_2 a_1 a_0$ and $\beta = \ldots b_2 b_1 b_0$. If $a_0 = b_0$ then $(\ldots a_2 a_1)_2 = (\ldots b_2 b_1)_2$ and so $\ldots a_2 a_1$ is a shorter counter-example. However, this contradicts our choice of α, so it must be that $a_0 \neq b_0$. 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 α is a w-NAF we have

$$\alpha = \ldots a_w 00 \ldots 0 a_0.$$

Write

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

where $\alpha_1, \beta_1 \in {D_w}^*$. Note that since α is a w-NAF, so is α_1, and further, $w(\alpha) = w(\alpha_1) + 1$.

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

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

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

Now,

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

Let $c_0 = ((b_{w-1}\ldots b_1 b_0)_2 - (00\ldots 0a_0)_2)/2^w$. We can derive a bound on $|c_0|$. Every digit in D_w has absolute value at most $2^{w-1} - 1$, thus

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

Combining these two inequalities gives

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

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 D_w-ADD-DIGIT(β_1, c_0). Then $({\beta_1}')_2 = (\beta_1)_2 + c_0$ and, by Lemma 3.2, $w({\beta_1}') \leq w(\beta_1) + 1$.

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

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

But, $(\alpha_1)_2 = ({\beta_1}')_2$ and α_1 is a w-NAF. Thus α_1 is a shorter counter-example, contrary to our choice of α. This proves the result. □

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 $\alpha = a_{\ell-1} \dots a_1 a_0$ be a representation in ${D_w}^*$ of an integer n with $a_{\ell-1} \neq 0$. If $w(\alpha) \leq w(\beta)$ for any $\beta \in {D_w}^*$ with $(\beta)_2 = n$ then $\ell \leq \lfloor \lg |n| \rfloor + 2$.*

Proof. We proceed by induction on ℓ, the length of α. Note that since $a_{\ell-1}$ is nonzero the length of α 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 $\alpha_1 = a_{\ell-1} \dots a_2 a_1$, so that $\alpha = \alpha_1 \| a_0$. Note that

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

Since α is a minimal Hamming weight representation of n, α_1 must be a minimal Hamming weight representation of $(n - a_0)/2$. The length of α_1 is $\ell - 1$, so by induction we have

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

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 assume $\lfloor \lg |n - a_0| \rfloor > \lfloor \lg |n| \rfloor$.

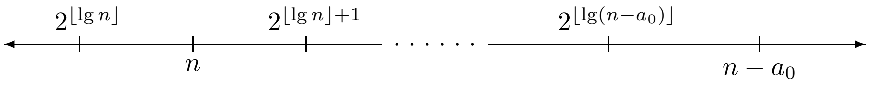
Now,

$$\begin{aligned} & \lfloor \lg |n - a_0| \rfloor > \lfloor \lg |n| \rfloor \\ \implies & |n - a_0| > |n| \\ \implies & a_0 \neq 0. \end{aligned}$$

Thus, α contains at least two nonzero digits, namely $a_{\ell-1}$ and a_0. This tells us that $w(\alpha) \geq 2$. 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, $w(\beta) = 1$ which is less than $w(\alpha) \geq 2$, 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). The following diagram displays part of the positive number line and helps illustrate some of our arguments:


$2^{\lfloor \lg n \rfloor}$
n
$2^{\lfloor \lg n \rfloor + 1}$
$\cdots\cdots$
$2^{\lfloor \lg(n-a_0) \rfloor}$
$n - a_0$

Let $d = 2^{\lfloor \lg n \rfloor + 1} - n$. Since n is odd, d is odd and from our number line we see that $d \leq |a_0| < 2^{w-1}$. Thus, $d \in D_w$. Consider the string $\beta \in {D_w}^*$ where

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

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 in D_w. Thus, $w(\alpha) \leq 2$. Now, both $w(\alpha) \geq 2$ and $w(\alpha) \leq 2$, and so $w(\alpha) = 2$. Thus, the string α has the following form

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

Now,

$$(\alpha)_2 = n \implies a_{\ell-1} \cdot 2^{\ell-1} + a_0 = n$$

Since n is positive and $n < n - a_0$ it must be that a_0 is negative. However, if a_0 is negative, then $a_{\ell-1}$ must be positive, and so we have

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

If $\lfloor \lg(n-a_0) \rfloor = \lfloor \lg n \rfloor + 1$ then this gives us the desired result. To finish the proof, we show this equality is valid.

We have $\lfloor \lg n \rfloor < \lfloor \lg(n - a_0) \rfloor$ and

$$\lfloor \lg n \rfloor < \lfloor \lg(n - a_0) \rfloor \implies \lfloor \lg n \rfloor + 1 \leq \lfloor \lg(n - a_0) \rfloor,$$

so if $\lfloor \lg(n - a_0) \rfloor \neq \lfloor \lg n \rfloor + 1$ then it must be that $\lfloor \lg n \rfloor + 1 < \lfloor \lg(n - a_0) \rfloor$. But,

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

However, as we saw earlier, $n \in D_w$ contradicts the fact that $w(\alpha) = 2$. Thus, $\lfloor \lg(n - a_0) \rfloor = \lfloor \lg n \rfloor + 1$ and this concludes our proof. □

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 w-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.

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

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

For example, if $\alpha = 0300\overline{3}0$ then $\alpha' = 010010$. 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 $\beta = 0300\overline{1}$, then $\beta' = 01001$ 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. For example, suppose $w = 3$ and $n = 42$. Below, we list several representations of 42 and for each one we give its associated string in $\{0,1\}^*$. The list is sorted under the relation "$\preceq$".

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

Notice that the 3-NAF of 42 is the smallest representation in this list. Even if the list contained *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 α 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

$$\begin{aligned} \beta &= \ldots b_w 00 \ldots 0 b_0 \\ \alpha &= \ldots a_w 00 \ldots 0 a_0. \end{aligned}$$

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 $(n - a_0)/2^w$. So, n can be neither even nor odd, which is a contradiction. Hence, we have the desired result. □

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 [6, 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 [3]). Because of Theorem 3.3, we now know that no other family of D_w-radix 2 representations can have average density lower than that of the w-NAF.

The w-NAF window method for scalar multiplication is a left-to-right method, 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 representations with digits in D_w that have minimal Hamming weight which can be computed from left to right [11]. Using such representations eliminates the bottleneck associated the w-NAF and results in a more efficient window method.

The results of this paper further strengthen the analogy between the ordinary NAF and the w-NAF. However, there is one property of the ordinary NAF which is not known to carry over to the w-NAF. In [8], a simple algorithm is described (due to Chang and Tsao-Wu [2]) which constructs the NAF of n by subtracting the binary representation of n from the binary representation of $3n$. It is not known if there is an analogous procedure which can be used to construct the w-NAF.

References

1. I. F. Blake, G. Seroussi and N. P. Smart. *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.
2. 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.
3. H. Cohen. Analysis of the Flexible Window Powering Algorithm, Preprint. Available from `http://www.math.u-bordeaux.fr/~cohen/window.dvi`.
4. 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.
5. D. M. Gordon. A Survey of Fast Exponentiation Methods, *Journal of Algorithms* **27** (1998), 129–146.
6. D. Hankerson, A. Menezes and S. Vanstone. *Guide to Elliptic Curve Cryptography*, Springer, 2004.
7. 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.
8. J. H. van Lint, *Introduction to Coding Theory*, 3rd edition, Springer, 1999.

9. 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.
10. B. Möller. Improved Techniques for Fast Exponentiation, in “Information Security and Cryptology – ICISC 2002”, *Lecture Notes in Computer Science* **2587** (2002), 298–312.
11. J. A. Muir and D. R. Stinson. New Minimal Weight Representations for Left-to-Right Window Methods, Preprint.
12. G. W. Reitwiesner. Binary Arithmetic, in *Advances in Computers, Vol. 1*, Academic Press, 1960, pp. 231–308.
13. J. A. Solinas. Efficient arithmetic on Koblitz curves. *Designs, Codes and Cryptography* **19** (2000), 195–249.

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://cacr.uwaterloo.ca/~dstinson