Combinatorics, Probability and Computing
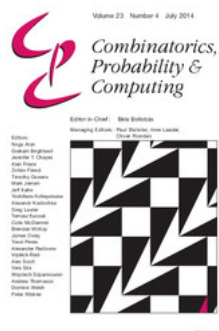
Additional services for **Combinatorics, Probability and Computing:**

Email alerts: Click here
Subscriptions: Click here
Commercial reprints: Click here
Terms of use : Click here

Analysis of the Binary Asymmetric Joint Sparse Form

CLEMENS HEUBERGER and SARA KROPF

Link to this article: http://journals.cambridge.org/abstract_S0963548314000352

How to cite this article:
CLEMENS HEUBERGER and SARA KROPF Analysis of the Binary Asymmetric Joint Sparse
Form. Combinatorics, Probability and Computing, Available on CJO 2014 doi:10.1017/
S0963548314000352

Request Permissions : Click here

Analysis of the Binary Asymmetric
Joint Sparse Form

CLEMENS HEUBERGER[1] and SARA KROPF[2†]

[1]Institut für Mathematik, Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65–67,
9020 Klagenfurt am Wörthersee, Austria
and
Institut für Optimierung und Diskrete Mathematik (Math B), TU Graz, Steyrergasse 30, 8010 Graz, Austria
(e-mail: clemens.heuberger@aau.at)

[2]Institut für Mathematik, Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65–67,
9020 Klagenfurt am Wörthersee, Austria
(e-mail: sara.kropf@aau.at)

We consider redundant binary joint digital expansions of integer vectors. The redundancy is used to minimize the Hamming weight, *i.e.*, the number of non-zero digit vectors. This leads to efficient linear combination algorithms in abelian groups, which are used in elliptic curve cryptography, for instance.

If the digit set is a set of contiguous integers containing zero, a special syntactical condition is known to minimize the weight. We analyse the optimal weight of all non-negative integer vectors with maximum entry less than N. The expectation and the variance are given with a main term and a periodic fluctuation in the second-order term. Finally, we prove asymptotic normality.

1. Introduction

We deal with integer representations of vectors of integers called joint representations.

Definition 1.1. For base 2, dimension d and a digit set $D \subseteq \mathbb{Z}$, the dimension-d *joint representation* of a vector $n \in \mathbb{Z}^d$ is a word $(\varepsilon_L \cdots \varepsilon_0)$ with $\varepsilon_i \in D^d$ and $n = \mathsf{value}(\varepsilon_L \cdots \varepsilon_0)$ with $\mathsf{value}(\varepsilon_L \cdots \varepsilon_0) = \sum_{i=0}^{L} \varepsilon_i 2^i$.

Such representations can be used for computing a linear combination $m_1 P_1 + \cdots + m_d P_d$ of points P_i of an elliptic curve, or more generally an abelian group (see [9]). For every non-zero digit ε_i, an elliptic curve addition is performed. Since these are expensive, we want to minimize the number of non-zero digits. On the other hand, every non-zero column vector $\varepsilon \in D^d$ corresponds to a precomputed point. The number of doublings corresponds to the length of the expansion. Each ε_i in the expansion $(\varepsilon_L \cdots \varepsilon_0)$ is called a column vector of the expansion.

Example 1.2. A dimension-3 digit expansion with digit set $\{0, 1, 2\}$ is

$$\begin{pmatrix} 1011 \\ 0020 \\ 2001 \end{pmatrix}.$$

It is a representation of $(11, 4, 17)^T$, because

$$\begin{pmatrix} 11 \\ 4 \\ 17 \end{pmatrix} = \mathsf{value} \begin{pmatrix} 1011 \\ 0020 \\ 2001 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} 2^3 + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} 2^2 + \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} 2^1 + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} 2^0.$$

Definition 1.3. The *Hamming weight* $h(\varepsilon_L \cdots \varepsilon_0)$ of a digit expansion $(\varepsilon_L \cdots \varepsilon_0)$ is the number of non-zero columns $\varepsilon_i \neq 0$.

Example 1.4. Continuing with Example 1.2, we have the Hamming weight

$$h \begin{pmatrix} 1011 \\ 0020 \\ 2001 \end{pmatrix} = 3.$$

The Hamming weight of an integer depends on the representation we use. For example, we have two representations of $4 = \mathsf{value}(12) = \mathsf{value}(100)$ with Hamming weight $h(12) = 2$ and $h(100) = 1$. But since we always use a specific digit expansion in this paper, we just write $h(n)$ for the Hamming weight of this digit expansion.

This specific digit expansion is the asymmetric joint sparse form (or AJSF) as presented by Heuberger and Muir in [7]. The AJSF is the unique dimension-d joint integer representation in base 2 with digit set $D_{l,u} = \{a \in \mathbb{Z} \mid l \leqslant a \leqslant u\}$ described in Theorem 2.3 (see [7, Theorem 6.1]). There, Heuberger and Muir proved that the AJSF is colexicographically minimal and has minimal Hamming weight among all representations with this digit set $D_{l,u}$.

The width-w non-adjacent form [8, 1] and the simple joint sparse form [5] are special cases of the asymmetric joint sparse form. For the width-w non-adjacent form, we use $l = -2^{w-1} + 1$, $u = 2^{w-1} - 1$ and dimension 1. The simple joint sparse form has digit set $D_{-1,1}$ and dimension 2. The special case of Theorem 1.5 for the simple joint sparse form has been proved in [5]. For further results on syntactically defined optimal digit expansions, we refer to [7] and the references therein.

We compute the expected value, the variance and the asymptotic distribution of the Hamming weight of the AJSF. We obtain a main term plus a periodic fluctuation and an error term, similar to the asymptotic estimates of digital sums in [3]. The definitions and algorithms of the AJSF are recalled in Section 2. In Section 3, we construct a transducer from this algorithm. In Theorem 3.6, we explicitly describe this transducer to compute the Hamming weight. In Section 4, we prove the following Theorem 1.5 about the asymptotic normal distribution of the Hamming weight. We use the discrete probability space $\{n \in \mathbb{Z} \mid 0 \leqslant n < N\}^d$ with uniform distribution as a probabilistic model, in contrast to [7]. There, only residue classes modulo powers of 2 have been considered in the 'full block-length' analysis.

Theorem 1.5. *The Hamming weight $h(m_1, \ldots, m_d)$ of the AJSF of an integer vector $(m_1, \ldots, m_d)^T$ over the digit set $D_{l,u}$ in dimension d, with equidistribution of all vectors $(m_1, \ldots, m_d)^T$ with $0 \leqslant m_i < N$ for an integer N, is asymptotically normally distributed. There exist constants $e_{l,u,d}$, $v_{l,u,d} \in \mathbb{R}$ and $\delta > 0$, depending on u, l and d, such that the expected value is*

$$e_{l,u,d} \log_2 N + \Psi_1(\log_2 N) + O(N^{-\delta} \log N)$$

and the variance is

$$v_{l,u,d} \log_2 N - \Psi_1^2(\log_2 N) + \Psi_2(\log_2 N) + O(N^{-\delta} \log^2 N),$$

where Ψ_1 and Ψ_2 are continuous, 1-periodic functions on $\mathbb{R}$. In particular, we have

$$\mathbb{P}\left(\frac{h(m_1, \ldots, m_d) - e_{l,u,d} \log_2 N}{\sqrt{v_{l,u,d} \log_2 N}} < x \right) = \int_{-\infty}^{x} e^{-y^2/2} dy + O\left(\frac{1}{\sqrt[4]{\log N}} \right)$$

for all $x \in \mathbb{R}$. For $d = 1$, we have

$$e_{l,u,1} = \frac{1}{w - 1 + \lambda} \quad \text{and} \quad v_{l,u,1} = \frac{(3 - \lambda)\lambda}{(w - 1 + \lambda)^3},$$

where

$$\lambda = \frac{2(u - l + 1) - (-1)^l - (-1)^u}{2^w}$$

and w is the unique integer such that $2^{w-1} \leqslant u - l + 1 < 2^w$. Furthermore, for $d = 1$, the function $\Psi_1(x)$ is nowhere differentiable. General formulas for $e_{l,u,d}$ for $d = 2$ are given in [7, Table 3]. For $d \in \{1, 2, 3, 4\}$, general formulas for $e_{l,u,d}$ and $v_{l,u,d}$ are given in [6].

For higher dimension or the variance, the question of non-differentiability of the periodic fluctuations remains open.

In the last Section 5, we further investigate the error term of the expected value and the variance of the Hamming weight in the case of the width-w non-adjacent form. In this case, we have $\delta = \log_2\left(1 + 3\pi^2/w^3\right)$ for sufficiently large w: see Theorem 5.1.

2. Preliminaries

First, we define some properties of the digit set.

Definition 2.1. Let $D_{l,u} = \{a \in \mathbb{Z} \mid l \leqslant a \leqslant u\}$ for $l \leqslant 0$ and $u \geqslant 1$ be the digit set. It contains $u - l + 1$ digits. We define w to be the unique integer s.t. $2^{w-1} \leqslant u - l + 1 < 2^w$.

Because $0, 1 \in D_{l,u}$, we have $w \geqslant 2$. The digit set contains at least a complete set of residues modulo 2^{w-1}. However, some residues modulo 2^w are not contained. Thus we define the following sets:

Definition 2.2. Let

$$\begin{aligned}
\mathsf{unique}(D_{l,u}) &= \{a \in D_{l,u} \mid u - 2^{w-1} < a < l + 2^{w-1}\}, \\
\mathsf{nonunique}(D_{l,u}) &= D_{l,u} \setminus \mathsf{unique}(D_{l,u}), \\
\mathsf{upper}(D_{l,u}) &= \{a \in D_{l,u} \mid u - 2^{w-1} < a \leqslant u\}.
\end{aligned}$$

The sets $\mathsf{unique}(D_{l,u})$ and $\mathsf{nonunique}(D_{l,u})$, respectively, contain the unique and nonunique residues modulo 2^{w-1}. The set $\mathsf{upper}(D_{l,u})$ is a complete set of representatives modulo 2^{w-1}.

Without loss of generality, we can restrict l to be greater than -2^{w-1}. Otherwise, we would take the digit set $D_{-u,-l}$ where we have $-2^{w-1} < -u \leqslant -1$. Then every representation of a vector n of integers with digit set $D_{l,u}$ would correspond to a representation of $-n$ with digit set $D_{-u,-l}$ by changing the sign of each digit. By this transformation, the weight of the representation does not change.

Theorem 2.3 ([7]). *Let $D_{l,u}$ be a digit set and $n \in \mathbb{Z}^d$ (with $n \geqslant 0$ if $l = 0$). Then there exists exactly one representation $(\varepsilon_L \cdots \varepsilon_0)$ (up to leading zeros) of n, such that the following conditions are satisfied.*

(1) *Each column ε_j is 0 or contains an odd digit.*
(2) *If $\varepsilon_j \neq 0$ for some j, then $\varepsilon_{j+w-2} = \cdots = \varepsilon_{j+1} = 0$.*
(3) *If $\varepsilon_j \neq 0$ and $\varepsilon_{j+w-1} \neq 0$ for some j, then:*

    (a) *there is an $i \in \{1, \ldots, d\}$ such that $\varepsilon_{j+w-1,i}$ is odd and $\varepsilon_{j,i} \in \mathsf{unique}(D_{l,u})$,*

    (b) *if $\varepsilon_{j,i} \in \mathsf{nonunique}(D_{l,u})$, then $\varepsilon_{j+w-1,i} \not\equiv u + 1 \bmod 2^{w-1}$,*

    (c) *if $\varepsilon_{j,i} \in \mathsf{upper}(D_{l,u}) \cap \mathsf{nonunique}(D_{l,u})$, then $\varepsilon_{j+w-1,i} \equiv u \bmod 2^{w-1}$.*

Definition 2.4. The digit expansion described in Theorem 2.3 is called the *asymmetric joint sparse form* (or AJSF) of n with digit set $D_{l,u}$.

Example 2.5. The AJSF of $(7, 11)^T$ with digit set $D_{-2,3}$ is

$$\begin{pmatrix} 100\bar{1} \\ 1003 \end{pmatrix},$$

where $\bar{1}$ is the digit -1. Thus its Hamming weight is $h(7, 11) = 2$.

We also consider the width-w non-adjacent form (see [8, 1]).

Definition 2.6. The *width-w non-adjacent form* (or w-NAF) of an integer n is a radix-2 representation $(\varepsilon_L \cdots \varepsilon_0)$ of n with the digit set $D_w := \{0, \pm 1, \pm 3, \ldots, \pm(2^{w-1} - 3), \pm(2^{w-1} - 1)\}$ and the following property:

$$\text{If } \varepsilon_i \neq 0, \text{ then } \varepsilon_{i+1} = \cdots = \varepsilon_{i+w-1} = 0.$$

The AJSF is a generalization of the w-NAF. In the one-dimensional case, only odd digits and 0 are used in the AJSF due to Theorem 2.3. After a non-zero digit, there are $w - 1$ zeros. Thus, for $l = -2^{w-1} + 1$ and $u = 2^{w-1} - 1$, we obtain the w-NAF.

It is known that the w-NAF representation exists and is unique for every integer (see [8]).

In [7] Heuberger and Muir introduce the AJSF, provide an algorithm to compute it, and prove its minimality with respect to the Hamming weight.

Theorem 2.7 ([7]). *The AJSF has minimal Hamming weight among all digit expansions of an integer vector n with digit set $D_{l,u}$. Algorithm 3 in [7] computes the AJSF in dimension d for an integer vector n.*

We present a slightly modified version of Algorithm 3 in [7] as Algorithm 1. The modification takes into account that we are only interested in the weight. Furthermore, those iterations of the **while** loop where the output is already predetermined are skipped.

For simplicity, we write $n + a$, for a vector n and an integer a, to denote that we add a to every coordinate of the vector n.

The **if** branch in line 3 of Algorithm 1 makes the digit at the current position a zero column if possible. If this is not possible, the **else** branch in line 6 chooses the smallest digit in each component which is congruent to the input. In the inner **if** and **else** branches, the algorithm checks whether we should change any non-unique digits. In the **if** statement in line 12, we check whether we can make the $(w - 1)$st digit after the current digit 0. Otherwise, in the **else** statement in line 17, we check whether we can increase the redundancy at the $(w - 1)$st digit after the current digit by changing any non-unique digits at the current position.

In the one-dimensional case, we can simplify Algorithm 1 further:

$$\text{If } I_{\text{unique}} \neq \emptyset, \text{ then } I_{\text{nonunique}} = \emptyset.$$

Thus the **else** branch in line 17 will not be processed. Algorithm 2 is the simplified version for the one-dimensional case.

3. Construction of the transducers

In this section we describe the construction of the transducers for the computation of the Hamming weight. We start with the easiest case, the w-NAF. We will then modify

---

Algorithm 1 Algorithm to compute the weight of the AJSF with digit set $D_{l,u}$

Input: A vector of integers $n \in \mathbb{Z}^d$, integers $l \leqslant 0$, $u > 0$, $n \geqslant 0$ if $l = 0$
Output: Weight $h(n)$
1: $h = 0$
2: **while** $n \neq 0$ **do**
3:     **if** $n \equiv 0 \bmod 2$ **then**
4:         $a = 0$, $h = h + 0$
5:         $m = n/2$
6:     **else**
7:         $a = l + ((n - l) \bmod 2^{w-1})$
8:         $h = h + 1$
9:         $m = (n - a)/2^{w-1}$
10:        $I_{\text{unique}} = \{j \in \{1, 2, \ldots, d\} \mid a_j \in \text{unique}(D_{l,u})\}$
11:       $I_{\text{nonunique}} = \{j \in \{1, 2, \ldots, d\} \mid a_j \in \text{nonunique}(D_{l,u})\}$
12:       **if** $m_j \equiv 0 \bmod 2$ for all $j \in I_{\text{unique}}$ **then**
13:           **for** $j \in I_{\text{nonunique}}$ such that m_j is odd **do**
14:              $a_j = a_j + 2^{w-1}$
15:              $m_j = m_j - 1$
16:           **end for**
17:         **else**
18:           **for** $j \in I_{\text{nonunique}}$ such that $m_j \equiv u + 1 \bmod 2^{w-1}$ **do**
19:              $a_j = a_j + 2^{w-1}$
20:              $m_j = m_j - 1$
21:           **end for**
22:         **end if**
23:     **end if**
24:     $n = m$
25: **end while**
26: **return** h

---

the ideas to deal with the asymmetric case of $D_{l,u}$-expansions in dimension 1. We finally generalize the approach to the d-dimensional $D_{l,u}$-expansions.

All transducers and automata take a (joint) binary expansion as input and read from right to left. The output of the transducers is a sequence of zeros and ones. Then the computed Hamming weight is the number of ones in this output.

Lemma 3.1. *Let $w \geqslant 2$. The transducer in Figure 1 calculates the weight $h(n)$ of the w-NAF of an integer n.*

Proof. Let $n = \text{value}(n_L \cdots n_0)$ with $n_j \in \{0, 1\}$ be the standard binary expansion of n and $(\varepsilon_K \cdots \varepsilon_0)$ be the w-NAF representation of n. If $n \equiv 0 \bmod 2$, then $\varepsilon_0 = 0$ and we stay in the initial state. Otherwise, we have $\varepsilon_0 \neq 0$ and the weight is $h(\varepsilon_0) = 1$. Since we have a w-NAF representation, the next $w - 1$ digits fulfil $\varepsilon_1 = \varepsilon_2 = \cdots = \varepsilon_{w-1} = 0$, no

---

Algorithm 2 Algorithm to compute the weight of the one-dimensional AJSF with digit set $D_{l,u}$

Input: Integers n, $l \leqslant 0$, $u > 0$, $n \geqslant 0$ if $l = 0$
Output: $h(n)$

  1: $h = 0$
  2: **while** $n \neq 0$ **do**
  3:      **if** $n \equiv 0 \bmod 2$ **then**
  4:         $a = 0$
  5:         $h = h + 0$
  6:         $m = n/2$
  7:      **else**
  8:         $a = l + ((n - l) \bmod 2^{w-1})$
  9:         $h = h + 1$
10:         $m = (n - a)/2^{w-1}$
11:         **if** $m \equiv 1 \bmod 2$ **and** $(n - l) \bmod 2^{w-1} \leqslant u - l - 2^{w-1}$ **then**
12:            $a = a + 2^{w-1}$
13:            $m = m - 1$
14:         **end if**
15:      **end if**
16:      $n = m$
17: **end while**
18: **return** h

---

matter what the corresponding n_j, $j = 1, \ldots, w - 1$ are. The sign of the digit ε_0 depends on $n_{w-1} \bmod 2$. If $n_{w-1} \equiv 0 \bmod 2$, then $\varepsilon_0 > 0$ and we go to state w with the next input $(n - \varepsilon_0)/2^w$ with carry 0. If $n_{w-1} \equiv 1 \bmod 2$, then $\varepsilon_0 < 0$ and we therefore have a carry of 1 and go to state $w + 1$. There, reading an input of 1 and having a carry of 1 results in the same outcome as reading 0, but the carry remains 1. Reading an input 0 with carry 1 is equivalent to reading an input 1 with a carry 0, so we are in state 1 again.   □

In the next step, we construct a transducer for the Hamming weight of the one-dimensional AJSF. Therefore, we need the following automaton to compare integers.

Lemma 3.2. *Automaton 2 in Figure 2 accepts the input of three integers a, b, c if and only if $a + b \leqslant c$. The binary expansions of a, b and c must have the same length, where leading zeros are allowed.*

Remark. Automaton 2 could be simplified by combining the input $a + b$ and merging the states $(1, 0)$ and $(0, 1)$. But since we use this automaton in Theorems 3.3 and 3.6 with the input of the transducer as a and some fixed parameter as b, this would complicate the constructions in Theorems 3.3 and 3.6.
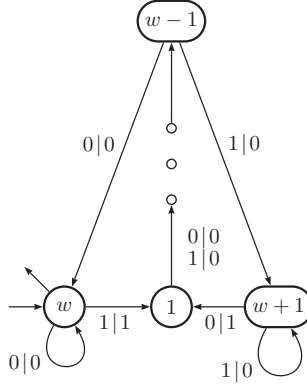
Figure 1. Transducer to compute the Hamming weight of a *w*-NAF representation.
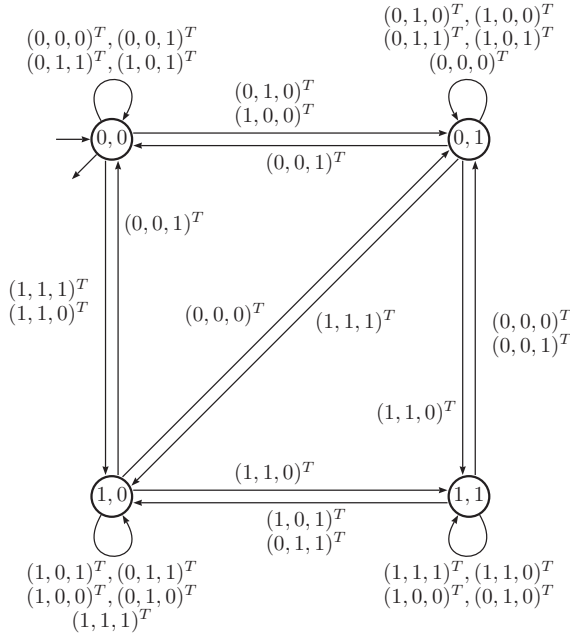

Figure 2. Automaton 2 to compare three integers *a*, *b* and *c*, accepts if $a + b \leqslant c$.

Proof. The states are (s, t) with $s, t \in \{0, 1\}$. The label *s* signifies the carry of the addition $a + b$ which still has to be processed. The label *t* corresponds to the truth value of the expression $(a + b) \bmod 2^i > c \bmod 2^i$, where *i* is the number of read digits up to now. So the automaton accepts the input if it stops in state $(0, 0)$ where there is no carry and $a + b > c$ is false. The initial state is $(0, 0)$.

Therefore, there is a path from $(0, 0)$ to (s, t) in Automaton 2 with input label

$$\begin{pmatrix} \alpha_{i-1} \cdots \alpha_0 \\ \beta_{i-1} \cdots \beta_0 \\ \gamma_{i-1} \cdots \gamma_0 \end{pmatrix}$$

if and only if

$$s = \left\lfloor \frac{\mathsf{value}(\alpha_{i-1} \cdots \alpha_0) + \mathsf{value}(\beta_{i-1} \cdots \beta_0)}{2^i} \right\rfloor$$

and

$$t = \left[\left(\mathsf{value}(\alpha_{i-1} \cdots \alpha_0) + \mathsf{value}(\beta_{i-1} \cdots \beta_0) \right) \bmod 2^i > \mathsf{value}(\gamma_{i-1} \cdots \gamma_0) \right].$$

Here, we use Iverson's notation, that is, $[expression]$ is 1 if *expression* is true and 0 otherwise. From this, the rules for the transitions follow. There is a transition

$$(s, t) \xrightarrow{(\alpha, \beta, \gamma)^T} (s', t')$$

if and only if $s' = \lfloor (\alpha + \beta + s)/2 \rfloor$ and $t' = \left[(\alpha + \beta + s) \bmod 2 > \gamma - t \right]$. □

Theorem 3.3. *There exists a transducer with input and output alphabet $\{0, 1\}$, having less than $4w - 2$ states, where one state is initial and final, that computes the Hamming weight of the AJSF from the binary expansion of an integer.*

Proof. We construct a transducer performing the same calculation as Algorithm 2. It will look similar to the transducer in Figure 1. We start at state 0. Then there is a vertical block of states with $w - 1$ rows having states $(0, 0)_i$, $(1, 0)_i$, $(0, 1)_i$ and $(1, 1)_i$ in each row $i = 1, \ldots, w - 1$. After this block, we either go back to state 0, or to a similar state 1, or again to the block of states (see Figure 3). We call the states 0 and 1 the looping states. Their labels signify the carry which is to be processed. The state 0 is also the final state.

The block of states corresponds to the **if** statement in line 11 in Algorithm 2. In this line, we have to check the inequality $(n - l) \bmod 2^{w-1} \leqslant u - l - 2^{w-1}$. A first step to this aim is to compare $n + \tilde{l} \leqslant \tilde{u}$ with $\tilde{l} := -l$ and $\tilde{u} := u - l - 2^{w-1}$. Therefore, we use Automaton 2.

Next, we examine the binary expansions of $\tilde{u}$ and $\tilde{l}$. Since we have assumed that $l > -2^{w-1}$, we know that the length of the binary expansion of $\tilde{l}$ is at most $w - 1$. Furthermore, $-1 \leqslant \tilde{u} < 2^{w-1}$. In the case $\tilde{u} = -1$, the set nonunique$(D_{l,u})$ is empty and we have no choices for the digits. We will return to this case later. Then the length of the binary expansion of $\tilde{u}$ is at most $w - 1$. Let $(l_{w-2} \cdots l_0)$ and $(u_{w-2} \cdots u_0)$ be the binary expansions of $\tilde{l}$ and $\tilde{u}$, respectively.

Now we can verify $n + \tilde{l} \bmod 2^{w-1} \leqslant \tilde{u}$ by checking the label t of the state (s, t) after reading $w - 1$ digits from the binary expansion of $(n, \tilde{l}, \tilde{u})^T$ in Automaton 2. If $t = 0$, then the inequality is true, otherwise it is false. Since the length of $\tilde{u}$ is less than or equal to $w - 1$, there are no digits of $\tilde{u}$ left. Only a possible carry of the addition $n + \tilde{l}$ is left. This carry is the label s of the current state (s, t). Therefore, we have checked $n + \tilde{l} \bmod 2^{w-1} \leqslant \tilde{u}$. To ensure that we read exactly $w - 1$ digits, the transducer in Figure 3 has $w - 1$ copies of the four states of Automaton 2. The transitions start in a state of the ith copy and go to an appropriate state of the $(i + 1)$th copy while reading the ith digit of the expansion.

In the **if** statement in line 11 in Algorithm 2, we must also check the other condition $m \equiv 1 \bmod 2$. Let $(s, t)_{w-1}$ be the current state at the end of the block of states. We know
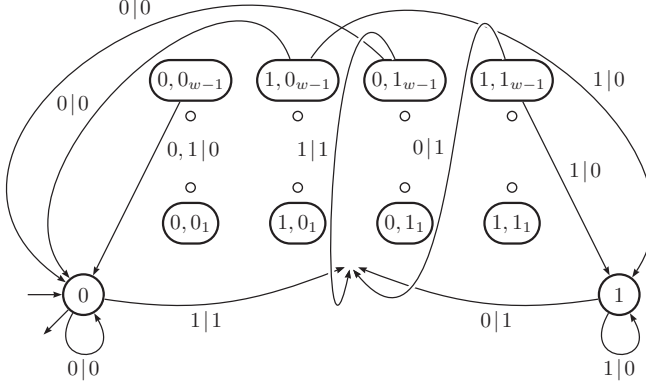
Figure 3.   Transducer to compute the weight of the AJSF with digits in $D_{l,u}$.
The transitions into and inside the block of states depend on l and u.

that

$$m = \frac{(n + \tilde{l}) - (n + \tilde{l}) \bmod 2^{w-1}}{2^{w-1}}.$$

Therefore, the least significant digit of m is simply the next digit of the addition $n + \tilde{l}$. Since there are no digits of the expansion of $\tilde{l}$ left, we only have to look at the next digit ε of n and consider the carry s. Thus we have $m \equiv s + \varepsilon \bmod 2$.

If the inequality of the **if** statement is satisfied, that is if $t = 0$, then whatever digit ε we read next, the transducer starts from a looping state again. If m is even, then the next written digit is 0 in any case. If m is odd, we can change the digit in the representation (because it is non-unique) and m becomes even too. We only have to remember the carry. If $s = 0$ or $s = 1$ and we read $\varepsilon = 0$, then there will be no carry propagation and we continue with state 0. If $s = 1$ and we read $\varepsilon = 1$, then there is a carry propagation and we start at state 1.

If the inequality is not satisfied, that is if $t = 1$, and $m \equiv s + \varepsilon \bmod 2$ is odd, then we have to start with the $w - 1$ transitions of Automaton 2 immediately. If m is even, however, then the transducer starts from a looping state again. In both cases, we have to consider the carry propagation as well.

At state $s \in \{0, 1\}$, we stay in state s as long as we read s. If we read $1 - s$ we start with the $w - 1$ transitions of Automaton 2.

In the case $\tilde{u} = -1$, the set $\mathsf{nonunique}(D_{l,u})$ is empty. Therefore, we have $t = 1$ in each state, and the initial state of Automaton 2 has to be $(0, 1)$. Let $(u_{w-2} \cdots u_0) = (0^{w-1})$. Then we have a transition from s to $(s', t')_1$ with input label $1 - s$ if and only if there is a transition from $(s, 1)$ to (s', t') with input label $(1 - s, l_0, u_0)^T$ in Automaton 2.

To summarize, we have the following transitions in the transducer in Figure 3 for s, s', t, t', $\varepsilon \in \{0, 1\}$ and $i \in \{1, \ldots, w - 2\}$:

- $s \xrightarrow{\varepsilon | 0} s$ if $s = \varepsilon$,
- $s \xrightarrow{\varepsilon | 1} (s', t')_1$ if $s \neq \varepsilon$ and $(s, [\tilde{u} = -1]) \xrightarrow{(\varepsilon, l_0, u_0)^T} (s', t')$ is a transition in Automaton 2,

- $(s,t)_i \xrightarrow{\varepsilon|0} (s',t')_{i+1}$ if $(s,t) \xrightarrow{(\varepsilon,l_i,u_i)^T} (s',t')$ is a transition in Automaton 2,
- $(s,t)_{w-1} \xrightarrow{\varepsilon|0} s'$ if $t=0$ or $\varepsilon + s \equiv 0 \mod 2$, and $s' = \lfloor(\varepsilon+s)/2\rfloor$,
- $(s,t)_{w-1} \xrightarrow{\varepsilon|1} (s',t')_1$ if $t=1$, $\varepsilon + s \equiv 1 \mod 2$ and $(s,[\tilde{u}=-1]) \xrightarrow{(\varepsilon,l_0,u_0)^T} (s',t')$ is a transition in Automaton 2.

We note that there is only one accessible state in the first row because the transitions $0 \xrightarrow{1|1} (s,t)_1$ and $1 \xrightarrow{0|1} (s,t)_1$ both have the same target state. This target state depends on l and u.

Finally, we restrict the transducer to the states which are actually accessible from the initial state. $\qquad\square$

Now we can describe the last state of the path with input label $(\varepsilon_L \cdots \varepsilon_0)$, a binary expansion. The following lemma can easily be proved by induction.

Lemma 3.4. *Let k_i, s_i, t_i for $i \geq 0$ and a_i, f_i for $i \geq 1$ be sequences with $s_0 = 0$, $t_0 = 1$. The states $(s_i, t_i)_{w-1}$ are the states in the last row of the path. The integers k_i count how often we circle in a looping state after the state $(s_i, t_i)_{w-1}$. The integers f_i are the positions of the non-zeros in the AJSF and a_i is the digit at position f_i. For $\tilde{u} \geq 0$, these sequences satisfy the following recursions for $i \geq 1$:*

$$
k_i = \begin{cases}
\max\{k \in \mathbb{N} \mid (\varepsilon_{f_i+k+w-2}\cdots\varepsilon_{f_i+w-1}) = (0^{k-1}1) \text{ or } (0^k)\} & \text{if } s_i = t_i = 0, \\
\max\{k \in \mathbb{N} \mid (\varepsilon_{f_i+k+w-2}\cdots\varepsilon_{f_i+w-1}) = (1^k) \text{ or } (0^k)\} & \text{if } s_i = 1, t_i = 0, \\
\max\{k \in \mathbb{N} \mid (\varepsilon_{f_i+k+w-2}\cdots\varepsilon_{f_i+w-1}) = (0^k)\} & \text{if } s_i = 0, t_i = 1, \\
\max\{k \in \mathbb{N} \mid (\varepsilon_{f_i+k+w-2}\cdots\varepsilon_{f_i+w-1}) = (1^k)\} & \text{if } s_i = t_i = 1,
\end{cases}
$$

$$
f_i = k_0 + \cdots + k_{i-1} + (i-1)(w-1),
$$

$$
s_i = \left\lfloor \frac{\mathsf{value}(\varepsilon_{f_i+w-2}\cdots\varepsilon_{f_i+1}1) + \tilde{l}}{2^{w-1}} \right\rfloor,
$$

$$
t_i = \left[(\mathsf{value}(\varepsilon_{f_i+w-2}\cdots\varepsilon_{f_i+1}1) + \tilde{l}) \mod 2^{w-1} > \tilde{u}\right],
$$

$$
a_i = -\tilde{l} + (\mathsf{value}(\varepsilon_{f_{i+1}-1}\cdots\varepsilon_{f_i+1}1) + \tilde{l} \mod 2^{k_i+w-1}).
$$

Then we have

$$
\mathsf{value}(\varepsilon_{f_i+w-2}\cdots\varepsilon_{f_i+1}1) = a_i + 2^{w-1}s_i(1 - [s_i = 1 \wedge t_i = 0 \wedge \varepsilon_{f_i+w-1} = 0]).
$$

There is a path from 0 to $(s,t)_j$ with input label $(\varepsilon_L \cdots \varepsilon_0)$ if and only if $j = L - f_i + 1$, $f_i \leq L \leq f_i + w - 2$ and

$$
s = \left\lfloor \frac{\mathsf{value}(\varepsilon_L \cdots \varepsilon_{f_i+1}1) + \mathsf{value}(l_{j-1}\cdots l_0)}{2^j} \right\rfloor,
$$

$$
t = \left[(\mathsf{value}(\varepsilon_L \cdots \varepsilon_{f_i+1}1) + \mathsf{value}(l_{j-1}\cdots l_0)) \mod 2^j > \mathsf{value}(u_{j-1}\cdots u_0)\right].
$$

There is a path from 0 to s with input label $(\varepsilon_L \cdots \varepsilon_0)$ if and only if $f_i + w - 1 \leq L \leq f_{i+1} - 1$ and $s = s_i$ or $s = 0$, $s_i = 1$ and $(\varepsilon_L \cdots \varepsilon_{f_i+w-1}) = (0^{L-f_i-w+2})$.
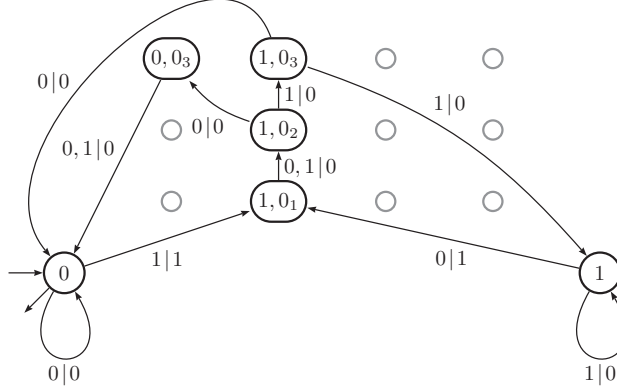*For $\tilde{u} = -1$ the only difference is $t_i = 1$ and $t = 1$.*

Figure 4.   Transducer to compute the weight of the AJSF with digits in $D_{-3,11}$. The non-accessible states are
grey.

Example 3.5. For $l = -3$ and $u = 11$, we have $w = 4$, $\tilde{l} = (011)_2$ and $\tilde{u} = u - l - 2^{w-1} = (110)_2$. The transducer can be seen in Figure 4, where all non-accessible states are grey.

We recall that a *reset sequence* of a transducer is a sequence $(n_L \cdots n_0)$ such that there exists a state s with the following property. For all states t, if the transducer is in state t and the next input is $(n_L \cdots n_0)$, then the transducer is in state s.

Now we generalize this transducer to arbitrary dimension d.

Theorem 3.6. *There exists a transducer to compute the Hamming weight of the AJSF for the joint binary expansion of a d-dimensional vector of integers as input. It has one state which is initial and final, input alphabet $\{0,1\}^d$, output alphabet $\{0,1\}$ and less than $8^d w$ states.*

The word 0^{4w} is a reset sequence of this transducer. It leads to the initial and final state of the transducer.

Proof.   We construct a transducer calculating the weight of AJSF. In order to explain the structure of this transducer, we first consider a provisional transducer implementing a simpler version of Algorithm 1, which omits the **else** branch in line 17; see also the algorithm on page 306 of [7]. The resulting provisional transducer is similar to the transducer in Figure 3.

For every vector $s \in \{0,1\}^d$, there is a state. These states are called looping states. The vector s signifies the carry at each coordinate. The state $(0,\ldots,0)^T$ is the initial state. Furthermore, there is a block of states. The states inside the block have the labels $(s,t)_i$ where $s, t \in \{0,1\}^d$, and i is the row in the block. The coordinates of s and t have the same meaning as in the proof of Theorem 3.3, that is, s is the carry of the addition $n + \tilde{l}$ and t signifies whether the digit is in $\mathsf{nonunique}(D_{l,u})$ or not.

If $s \in \{0,1\}^d$ is a looping state, then there is a loop with label $s \mid 0$ at this state. Because if we read $\varepsilon = s$, then we have $\varepsilon + s \equiv 0 \bmod 2$, the output is 0 and the carry propagates to the next step. If we read $\varepsilon \neq s$, then we start with the $w - 1$ transitions of Automaton 2 in Figure 2 in each coordinate. These $w - 1$ transitions are processed independently for

every coordinate. Therefore, we need 4^d states in each row and $w-1$ rows to process exactly $w-1$ transitions of Automaton 2.

After the last row of the block of states, we either go back to a looping state or again start with the block of states immediately. Let $(s,t)_{w-1}$ be the current state in the last row and ε the next input digit. As in the one-dimensional case we have $m \equiv \varepsilon + s \bmod 2$. If for every coordinate j, $t_j = 1$ implies that m_j is even, then we have to process the **if** branch in line 12. We write this condition as the scalar product $t \cdot (s+\varepsilon \bmod 2) = 0$. In this case, the next output digit will be 0 and we go on to a looping state s' where the new carry is $s' = \lfloor (s+\varepsilon)/2 \rfloor$.

If $t \cdot (s+\varepsilon \bmod 2) = 0$ does not hold, then we would have to process the **else** branch in line 17. But since we skip this part for now, we simply have to restart the transducer with the input m in the case $t \cdot (s+\varepsilon \bmod 2) > 0$. We know that m is the original next input ε plus the carry s. In this case, $s \neq \varepsilon$, otherwise $t \cdot (s+\varepsilon \bmod 2) > 0$ would be false. Therefore, there is a transition $s \xrightarrow{\varepsilon|1} (s',t')_1$ in this transducer. This ensures that, when restarting the transducer with input m, we immediately go on to the state $(s',t')_1$. Hence, we have a transition $(s,t)_{w-1} \xrightarrow{\varepsilon|1} (s',t')_1$ in the provisional transducer.

Altogether, for $s, s', t, t' \in \{0,1\}^d$, $i \in \{1,\dots w-2\}$, $j \in \{1\dots d\}$ and $\varepsilon \in \{0,1\}^d$, we have the following transitions in this provisional transducer:

- $s \xrightarrow{\varepsilon|0} s$ if $\varepsilon = s$,
- $s \xrightarrow{\varepsilon|1} (s',t')_1$ if $\varepsilon \neq s$ and $\forall j : (s_j,[\tilde{u}=-1]) \xrightarrow{(\varepsilon_j,l_0,u_0)^T} (s'_j,t'_j)$ is a transition in Automaton 2,
- $(s,t)_i \xrightarrow{\varepsilon|0} (s',t')_{i+1}$ if $\forall j : (s_j,t_j) \xrightarrow{(\varepsilon_j,l_i,u_i)^T} (s'_j,t'_j)$ is a transition in Automaton 2,
- $(s,t)_{w-1} \xrightarrow{\varepsilon|0} s'$ if $t \cdot (s+\varepsilon \bmod 2) = 0$ and $s' = \lfloor (s+\varepsilon)/2 \rfloor$,
- $(s,t)_{w-1} \xrightarrow{\varepsilon|1} (s',t')_1$ if $t \cdot (s+\varepsilon \bmod 2) > 0$ and $s \xrightarrow{\varepsilon|1} (s',t')_1$ is a transition in this transducer.

This transducer does the same as Algorithm 1 without the **else** branch in line 17. In the case $\tilde{u} = -1$ we are finished because in the **else** branch nothing is done. Otherwise we must consider the **else** statement.

Let $(s,t)_{w-1}$ be the current state in the last row and let ε be the next input digit. To process the **else** branch, $t \cdot (s+\varepsilon \bmod 2) > 0$ must hold in the state $(s,t)_{w-1}$. Otherwise, we would process the **if** branch. First let us examine one coordinate j. If $t_j = 1$, nothing is done in the **else** branch because the digit at this coordinate is unique. If $t_j = 0$, we have to decide whether $m_j \equiv u+1 \bmod 2^{w-1}$. Here, $m_j \bmod 2^{w-1}$ corresponds to the next $w-1$ input digits plus the carry s_j from the current state $(s,t)_{w-1}$. So we just have to compare the input letters plus the carry with the binary expansion of $u+1 \bmod 2^{w-1}$ or, equivalently, we compare $m_j - l \bmod 2^{w-1}$ with $\tilde{v} = u - l + 1 \bmod 2^{w-1}$. If they are not the same at some point, then we just go on as we did in the provisional transducer.

If they are the same, we have to process the **else** branch. There we would have taken $m_j - 1$ as the next input of the algorithm instead of m_j. Therefore we have to decide where we would be in the provisional transducer, when starting in $(s,t)_{w-1}$ and the input is the original input minus 1. This case only happens if initially the next non-zero

digit is unique, but changing the current digit ensures that the next non-zero digit is non-unique. Nevertheless, the next digit will not be 0, since this is the case when the **if** branch is processed. Therefore we would start in $(s, t)_{w-1}$ with original input minus 1 and immediately go to the block of states again. Otherwise, the next digit would be 0. Thus after $w - 1$ transitions, we are again in a state $(s', t')_{w-1}$ in the last row. Since the next digit is non-unique, we have $t'_j = 0$.

To determine the value of s'_j, we have to decide whether there is a carry at position $w - 1$ in the addition of $m_j - 1$ and $\tilde{l}$. We have $m_j - 1 \bmod 2^{w-1} = u + k2^{w-1}$ for $k \in \mathbb{Z}$. Since $0 \leqslant u \leqslant 2^w - 1$, we have $k \in \{0, -1\}$. Then the carry is

$$
\begin{aligned}
s'_j &= \left\lfloor \frac{(m_j - 1) \bmod 2^{w-1} + \tilde{l} \bmod 2^{w-1}}{2^{w-1}} \right\rfloor \\
&= \left\lfloor \frac{u + \tilde{l} + k2^{w-1}}{2^{w-1}} \right\rfloor \\
&= 1 + k,
\end{aligned}
$$

because $2^{w-1} \leqslant u + \tilde{l} < 2^w$. Therefore, we have

$$
s'_j = \begin{cases} 0 & \text{if } u \geqslant 2^{w-1}, \\ 1 & \text{if } u < 2^{w-1}. \end{cases}
$$

As a result, the state $(s', t')_{w-1}$ where we would be in the provisional transducer has

$$
([u < 2^{w-1}], 0)
$$

in the jth coordinate.

To remember that we can change the jth coordinate at the end of the block, we have to use a second identical block $\{j\}$. Let $\emptyset$ be the first block, which already exists in the provisional transducer. Let $(s, t)_i^C$ be a state in block C. At the end of block $\emptyset$, we go to block $\{j\}$ if $t \cdot (s + \varepsilon \bmod 2) > 0$ and $t_j = 0$. Otherwise, we go to a looping state or to the block $\emptyset$. If we find out that $m_j \not\equiv u + 1 \bmod 2^{w-1}$ in block $\{j\}$, then we go back to the appropriate state in block $\emptyset$. At the end of block $\{j\}$ in the state $(s, t)_{w-1}^{\{j\}}$, we go to the same states as we would go from the state with $([u < 2^{w-1}], 0)_{w-1}^{\emptyset}$ in the jth coordinate.

So far we have only considered one coordinate. Now we combine this approach for all coordinates. Since for each coordinate we have to remember whether we are allowed to change it or not, we need one block for every subset of coordinates. Let block $C \subseteq \{1, \ldots, d\}$ be the block where we can change the coordinates in C. The states in block C are denoted by $(s, t)_i^C$. The block $\emptyset$ is the block which already exists in the provisional transducer. The block $\{1, \ldots, d\}$ is not accessible since we need at least one unique coordinate and only non-unique coordinates can be changed.

If we are in block $C \neq \emptyset$ and we find out that not every coordinate $j \in C$ satisfies $m_j \equiv u + 1 \bmod 2^{w-1}$, we go to the appropriate state in block

$$
C' = C \setminus \{j \in \{1, \ldots, d\} \mid m_j \not\equiv u + 1 \bmod 2^{w-1}\}.
$$

At the end of block C in state $(s, t)_{w-1}^C$, we can change the coordinates in C and all other coordinates remain the same. Therefore, we go to the same states as we would go

from $(\tilde{s}, \tilde{t})^{\emptyset}_{w-1}$ where $\tilde{s}_j = [u < 2^{w-1}]$, $\tilde{t}_j = 0$ for $j \in C$ and all other coordinates stay the same, that is $\tilde{s}_j = s_j$ and $\tilde{t}_j = t_j$ for $j \notin C$.

Let $(v_{w-2} \cdots v_0)$ be the binary expansion of $\tilde{v}$. Further, let s, s', t, $t' \in \{0,1\}^d$, C, $C' \subsetneq \{1, \ldots, d\}$, $j \in \{1 \ldots d\}$, $i \in \{1, \ldots, w-2\}$ and $\varepsilon \in \{0,1\}^d$. Then altogether there are the following transitions in the final transducer:

- $s \xrightarrow{\varepsilon|0} s$ if $s = \varepsilon$,
- $s \xrightarrow{\varepsilon|1} (s', t')^{\emptyset}_1$ if $s \xrightarrow{\varepsilon|1} (s', t')_1$ is a transition in the provisional transducer,
- $(s,t)^C_i \xrightarrow{\varepsilon|0} (s', t')^{C'}_{i+1}$ if $(s,t)_i \xrightarrow{\varepsilon|0} (s', t')_{i+1}$ is a transition in the provisional transducer and $C' = C \setminus \{j : s_j + \varepsilon_j + l_i \bmod 2 \neq v_i\}$,
- $(s,t)^{\emptyset}_{w-1} \xrightarrow{\varepsilon|0} s'$ if $t \cdot (s + \varepsilon \bmod 2) = 0$ and $s' = \lfloor (s + \varepsilon)/2 \rfloor$,
- $(s,t)^{\emptyset}_{w-1} \xrightarrow{\varepsilon|1} (s', t')^{C'}_1$ if $t \cdot (s + \varepsilon \bmod 2) > 0$, $(s,t)_{w-1} \xrightarrow{\varepsilon|1} (s', t')_1$ is a transition in the provisional transducer and $C' = \{j : s_j + \varepsilon_j + l_0 \bmod 2 = v_0 \text{ and } t_j = 0\}$,
- $(s,t)^C_{w-1} \xrightarrow{\varepsilon|1} (s', t')^{C'}_1$ if $C \neq \emptyset$ and $(\tilde{s}, \tilde{t})^{\emptyset}_{w-1} \xrightarrow{\varepsilon|1} (s', t')^{C'}_1$ is a transition in this transducer with $\tilde{s}_j = [u < 2^{w-1}]$, $\tilde{t}_j = 0$ for $j \in C$ and $\tilde{s}_j = s_j$, $\tilde{t}_j = t_j$ for $j \notin C$,
- $(s,t)^C_{w-1} \xrightarrow{\varepsilon|0} s'$ if $C \neq \emptyset$ and $(\tilde{s}, \tilde{t})^{\emptyset}_{w-1} \xrightarrow{\varepsilon|0} s'$ is a transition in this transducer with $\tilde{s}_j = [u < 2^{w-1}]$, $\tilde{t}_j = 0$ for $j \in C$ and $\tilde{s}_j = s_j$, $\tilde{t}_j = t_j$ for $j \notin C$.

Finally, we restrict the transducer to the states which are actually accessible from the initial state.

Due to the construction of the transducer, the sequence 0^{4w} leads to the initial and final state from any state.

It is possible to define similar sequences to those in Lemma 3.4, but since this requires more than one page, we omit this here. $\square$

Example 3.7. In Figure 5 there is a sketch of the transducer computing the weight of the AJSF over $D_{-2,3}$ in dimension 2. The labels of transitions are omitted in the figure and the transitions going back at the end of a block or inside a block are grey. We have $w = 3$, $\tilde{u} = (01)$, $\tilde{l} = (10)$ and $\tilde{v} = (10)$.

For example, the state $\binom{01}{11}^{\{2\}}_2$ has transitions to the same states as the state $\binom{01}{10}^{\emptyset}_2$ since $u < 2^{w-1}$.

4. Proof of Theorem 1.5

This section contains the proof of Theorem 1.5, which is a generalization of Theorem 6 in [5]. With the transducer in Theorem 3.6, we can compute the asymptotic Hamming weight. Therefore we use the following lemma, which can be proved by induction on L.

Lemma 4.1. *Let A_0, A_1 be matrices in $\mathbb{C}^{n \times n}$, $H : \mathbb{N} \to \mathbb{C}^{n \times n}$ be any function and $G : \mathbb{N} \to \mathbb{C}^{n \times n}$ be a function which satisfies the recurrence relation*

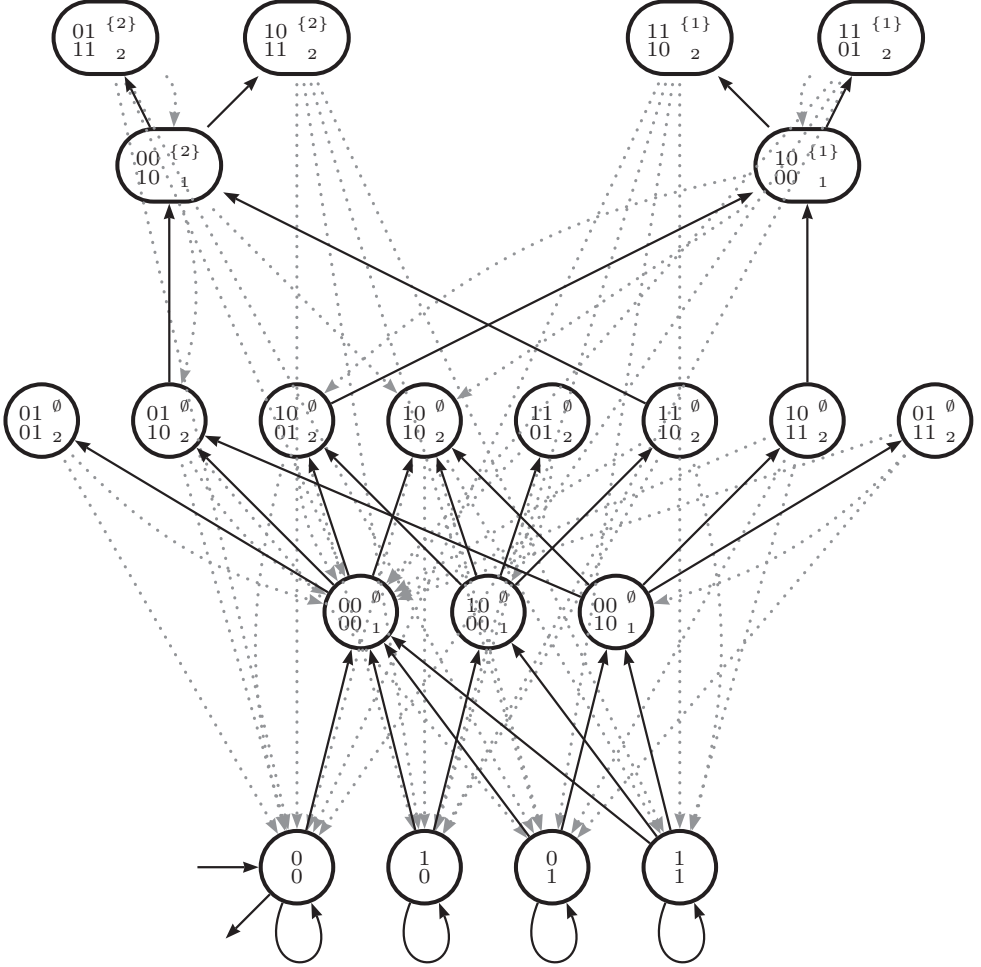$$G(2N + \varepsilon) = A_\varepsilon G(N) + \varepsilon H(N)$$

Figure 5. Transducer to compute the Hamming weight of the AJSF in dimension 2 over the digit set $D_{-2,3}$.

for $N \geqslant 1$ and $\varepsilon \in \{0,1\}$. Then

$$G\left(\sum_{p=0}^{L} \varepsilon_p 2^p\right) = \sum_{p=0}^{L} \varepsilon_p \left(\prod_{j=0}^{p-1} A_{\varepsilon_j}\right) H\left(\sum_{j=p+1}^{L} \varepsilon_j 2^{j-p-1}\right),$$

where we additionally set $H(0) = G(1)$.

We define $f(m_1,\ldots,m_d) := e^{ith(m_1,\ldots,m_d)}$. The matrices $M_{\varepsilon_1,\ldots,\varepsilon_d}$ for $\varepsilon_i \in \{0,1\}$ are defined as follows. The (j,k)th entry of the matrix $M_{\varepsilon_1,\ldots,\varepsilon_d}$ is e^{ith} if there is a transition from state j to k with input label $(\varepsilon_1,\ldots,\varepsilon_d)^T$ and output label h. The entry is 0 if there is no transition from state j to k with this input label. The ordering of states is considered to be fixed in

such a way that the initial state is the last state. Then we have

$$f(m_1,\ldots,m_d) = v^T \prod_{p=0}^{L} M_{m_{1,p},\ldots,m_{d,p}} M_{0,\ldots,0}^{4w} v \qquad (4.1)$$

for $v^T = (0,\ldots,0,1)$ and $m_i = \sum_{p=0}^{L} m_{i,p} 2^p$. The product describes all possible paths between states, using edges with input labels corresponding to the input $(m_1,\ldots,m_d)$. The exponent of the entries of the matrix product is the sum of output labels on these paths. Since we are interested in paths starting and ending in state $(0,\ldots,0)^T$, we multiply by v^T from the left and v from the right. The factor $M_{0,\ldots,0}^{4w}$ is due to the reset sequence from Theorem 3.6 and ensures that we stop at the final state.

We further define the following summatory functions

$$E(N) = \sum_{0 \leqslant m_1,\ldots,m_d < N} f(m_1,\ldots,m_d),$$

$$F(N) = \sum_{0 \leqslant m_1,\ldots,m_d < N} M(m_1,\ldots,m_d),$$

with

$$M(m_1,\ldots,m_d) = \prod_{p=0}^{L} M_{m_{1,p},\ldots,m_{d,p}}.$$

In other words, this last equation says that the function $M(m_1,\ldots,m_d)$ is 2-multiplicative (see [2]). By (4.1), we have

$$E(N) = v^T F(N) M_{0,\ldots,0}^{4w} v.$$

To write down a recursion formula for $F(N)$, we need the matrices

$$B_{C,D} := \sum_{\substack{\varepsilon_i = 0,1 \\ i \notin C \cup D}} \sum_{\substack{\varepsilon_i = 0 \\ i \in C}} \sum_{\substack{\varepsilon_i = 1 \\ i \in D}} M_{\varepsilon_1,\ldots,\varepsilon_d}$$

for disjoint $C, D \subseteq \{1,\ldots,d\}$. The first index C of $B_{C,D}$ is the set of coordinates where the digit is 0. The second index D is the set of coordinates where the digit is 1. All other coordinates in $(C \cup D)^c$ can be any digit. By construction, we have $\|M_{\varepsilon_1,\ldots,\varepsilon_d}\|_1 = 1$, where $\|\cdots\|_1$ denotes the row sum norm of a matrix. We conclude that $\|B_{C,D}\|_1 \leqslant 2^{d-|C|-|D|}$. As a special matrix we define $A = B_{\emptyset,\emptyset}$.

Furthermore we define the functions

$$G_C(N) := \sum_{\substack{0 \leqslant m_i < N \\ i \notin C}} \sum_{\substack{m_i = N \\ i \in C}} M(m_1,\ldots,m_d)$$

for every set $C \subseteq \{1,\ldots,d\}$.

Then we have $F(N) = G_\emptyset(N)$, and the functions satisfy the following recursion formulas due to 2-multiplicativity:

$$G_C(2N) = \sum_{\substack{\varepsilon_i=0,1 \\ i\notin C}} \sum_{\substack{\varepsilon_i=0 \\ i\in C}} \sum_{\substack{2m_i+\varepsilon_i<2N \\ i\notin C}} \sum_{\substack{2m_i+\varepsilon_i=2N \\ i\in C}} M(2m_1+\varepsilon_1,\ldots,2m_d+\varepsilon_d)$$

$$= B_{C,\emptyset} G_C(N),$$

$$G_C(2N+1) = \sum_{\substack{\varepsilon_i=0,1 \\ i\notin C}} \sum_{\substack{\varepsilon_i=1 \\ i\in C}} \sum_{\substack{2m_i+\varepsilon_i<2N+1 \\ i\notin C}} \sum_{\substack{2m_i+\varepsilon_i=2N+1 \\ i\in C}} M(2m_1+\varepsilon_1,\ldots,2m_d+\varepsilon_d)$$

$$= \sum_{D\subseteq C^c} \sum_{\substack{\varepsilon_i=0,1 \\ i\notin C\cup D}} \sum_{\substack{\varepsilon_i=0 \\ i\in D}} \sum_{\substack{\varepsilon_i=1 \\ i\in C}} M_{\varepsilon_1,\ldots,\varepsilon_d} \sum_{\substack{m_i<N \\ i\notin C\cup D}} \sum_{\substack{m_i=N \\ i\in C\cup D}} M(m_1,\ldots,m_d)$$

$$= \sum_{D\subseteq C^c} B_{D,C} G_{C\cup D}(N).$$

From this recursion, we can determine $G_C(N)$ inductively because all functions $G_{C'}$ required for computing G_C have $C' \supsetneq C$. Therefore, we have the following recursion formula for $F(N) = G_\emptyset(N)$:

$$F(2N+\varepsilon) = AF(N) + \varepsilon H(N)$$

for $N \geqslant 1$, $\varepsilon \in \{0,1\}$ and

$$H(N) = \sum_{\emptyset\neq D\subseteq\{1,\ldots,d\}} B_{D,\emptyset} G_D(N).$$

If we define $H(0) = G_\emptyset(1)$, we can use Lemma 4.1 and get

$$F\left(\sum_{p=0}^{L} \varepsilon_p 2^p\right) = \sum_{p=0}^{L} \varepsilon_p A^p H\left(\sum_{j=p+1}^{L} \varepsilon_j 2^{j-p-1}\right). \tag{4.2}$$

Here, $H(N)$ is considered to be a known function because it is a sum of functions $G_C(N)$, which are recursively known by Lemma 4.1.

From the definition of $G_C(N)$, we can derive the growth rates of the functions $G_C(N)$ and $H(N)$. We have $\|G_C(N)\|_1 = O(N^{d-|C|})$ and $\|H(N)\|_1 = O(N^{d-1})$.

Next, we investigate the eigenvalues of the matrix A. We first consider the case $t = 0$. In this case, A is the adjacency matrix of the underlying graph of the transducer in Theorem 3.6. Therefore, it has an eigenvalue 2^d with eigenvector $(1,\ldots,1)^T$. By the Perron–Frobenius theorem, there is a unique dominant eigenvalue $\mu(0)$ of A which is easily seen to be primitive as every state is reachable from any other state in exactly $4w$ steps. As $\|A\|_1 \leqslant 2^d$ and the largest eigenvalue is always at most $\|A\|_1$, $\mu(0) = 2^d$ is the largest eigenvalue. We denote the modulus of the second largest eigenvalue by $\beta(0)$. Since eigenvalues are continuous, for t in a suitable neighbourhood of 0, A has a unique dominant eigenvalue $\mu(t)$ and the modulus $\beta(t)$ of the second largest eigenvalue fulfils $\beta(t) < |\mu(t)|$.

Now we want to split up (4.2) into two parts, one for the dominating eigenvalue and one for the remaining eigenvalues. Therefore, let $J = T^{-1}AT$ be a Jordan decomposition

of A where J has been sorted such that it has $\mu(t)$ in the upper left corner. We define

$$\Lambda := T\operatorname{diag}(\mu(t)^{-1}, 0, \ldots, 0)T^{-1} \quad \text{and} \quad R := T(J - \operatorname{diag}(\mu(t), 0, \ldots, 0))T^{-1}.$$

Then $A^p = \mu^L \Lambda^{L-p} + R^p$ holds for $p \leqslant L$. Further, we define

$$\Lambda(x_0, x_1, \ldots) = \sum_{p=0}^{\infty} x_p \Lambda^p H\left(\sum_{j=0}^{p-1} x_j 2^{p-1-j}\right), \tag{4.3}$$

$$R\left(\sum_{p=0}^{L} \varepsilon_p 2^p\right) = \sum_{p=0}^{L} \varepsilon_p R^p H\left(\sum_{j=p+1}^{L} \varepsilon_j 2^{j-p-1}\right).$$

The function Λ is well defined on the infinite product space $\{0,1\}^{\mathbb{N}}$ because it is dominated by a geometric series. We extend Λ to a function on $[1,2)$ by setting

$$\Lambda\left(\sum_{p=0}^{\infty} x_p 2^{-p}\right) := \Lambda(x_0, x_1, \ldots)$$

with the standard binary expansion and choosing the representation ending on 0^{ω} in the case of ambiguity.

Then we have

$$F\left(\sum_{p=0}^{L} \varepsilon_p 2^p\right) = \mu^L \Lambda(\varepsilon_L, \varepsilon_{L-1}, \ldots, \varepsilon_0, 0^{\omega}) + R(\varepsilon_L, \ldots, \varepsilon_0)$$

and

$$E(N) = \mu(t)^{\log_2 N} \Psi(\log_2 N, t) + \tilde{R}(N, t)$$

with

$$\Psi(x, t) = \mu(t)^{-\{x\}} v^T \Lambda(2^{\{x\}}) M_{0,\ldots,0}^{4w} v \quad \text{and} \quad \tilde{R}(N, t) = v^T R(N) M_{0,\ldots,0}^{4w} v.$$

Furthermore, there is a $\delta \in (0,1]$ such that $\log_2 \beta(t) < d - \delta$ in a suitable neighbourhood of 0. Then we have

$$|\tilde{R}(N, t)| = O(N^{d-\delta}).$$

So we have

$$E(N) = N^{d+a_1 t + a_2 t^2 + O(t^3)} \Psi(\log_2 N, t) + O(N^{d-\delta}) \tag{4.4}$$

with a_1 and a_2 depending on the Taylor expansion of $\log_2 \mu(t)$ at $t = 0$. If we insert $t = 0$ in (4.4), we obtain $\psi_0 = \Psi(\log_2 N, 0) = 1 + O(N^{-\delta})$.

The function $\Psi(x, t)$ is periodic in x with period 1 and is well defined for all $x \in \mathbb{R}^+$. To prove continuity in x, we first note that continuity for $x \in [0, 1)$ with $x = \log_2 y$, where y is not a dyadic rational follows from (4.3). To prove it for $x = \log_2 y$ with $y = \sum_{p=0}^{L} \varepsilon_p 2^{-p}$ a dyadic rational with $\varepsilon_L = 1$, we observe that the two one-sided limits exist due to (4.3).

Next, we prove that they are the same. Therefore, we look at the two sequences $N_k = y2^{L+k}$ and $\tilde{N}_k = y2^{L+k} - 1$. Then

$$\lim_{k \to \infty} 2^{\{\log_2 N_k\}} = (\varepsilon_0 \cdot \varepsilon_1 \cdots \varepsilon_{L-1} 10^\omega) \quad \text{and} \quad \lim_{k \to \infty} 2^{\{\log_2 \tilde{N}_k\}} = (\varepsilon_0 \cdot \varepsilon_1 \cdots \varepsilon_{L-1} 01^\omega).$$

If we insert these two sequences in (4.4), we get

$$O(N_k^{d-1}) = E(N_k) - E(\tilde{N}_k) = N_k^d \Psi(\log_2 N_k, t) - \tilde{N}_k^d \Psi(\log_2 \tilde{N}_k, t) + O(N_k^{d-\delta}),$$

and hence $\lim_{k \to \infty} \Psi(\{\log_2 N_k\}, t) = \lim_{k \to \infty} \Psi(\{\log_2 \tilde{N}_k\}, t)$. Therefore, $\Psi(x, t)$ is continuous in x.

In t, $\Psi(x, t)$ is also continuous because the eigenvalues of a matrix are continuous. Furthermore, the function $\Psi(x, t)$ is arbitrarily often differentiable in t because it is dominated by a geometric series. By the same argument as above, these derivatives are continuous in x.

The first and second derivative of $E(N)$ with respect to t at $t = 0$ imply that the expected value of the Hamming weight is

$$\frac{1}{N^d} \sum_{m_i < N} h(m_1, \ldots, m_d) = e_{l,u,d} \log_2 N + \Psi_1(\log_2 N) + O(N^{-\delta} \log N) \tag{4.5}$$

with $e_{l,u,d} = -i a_1 \log 2$ and

$$\Psi_1(\log_2 N) = -i \frac{\partial}{\partial t} \Psi(\log_2 N, t)|_{t=0},$$

and

$$\frac{1}{N^d} \sum_{m_i < N} h^2(m_1, \ldots, m_d) = v_{l,u,d} \log_2 N + e_{l,u,d}^2 \log_2^2 N + 2 e_{l,u,d} \log_2 N \Psi_1(\log_2 N)$$

$$+ \Psi_2(\log_2 N) + O(N^{-\delta} \log^2 N)$$

with $v_{l,u,d} = -2 a_2 \log 2$ and

$$\Psi_2(\log_2 N) = -\frac{\partial^2}{\partial t^2} \Psi(\log_2 N, t)|_{t=0}.$$

From that, we calculate the variance, which is

$$\frac{1}{N^d} \sum_{m_i < N} h^2(m_1, \ldots, m_d) - \left(\frac{1}{N^d} \sum_{m_i < N} h(m_1, \ldots, m_d) \right)^2 =$$

$$v_{l,u,d} \log_2 N - \Psi_1^2(\log_2 N) + \Psi_2(\log_2 N) + O(N^{-\delta} \log^2 N).$$

We first compute the characteristic function $\hat{g}_N(t)$ of the random variable

$$Z = \frac{h(m_1, \ldots, m_d) - e_{l,u,d} \log_2 N}{\sqrt{v_{l,u,d} \log_2 N}},$$

which is

$$\hat{g}_N(t) = \frac{1}{N^d} \sum_{m_i < N} \exp\left(it \frac{h(m_1, \ldots, m_d) - e_{l,u,d} \log_2 N}{\sqrt{v_{l,u,d} \log_2 N}}\right)$$

$$= e^{-t^2/2}\left(1 + O\left(\frac{t^3}{\log^{3/2} N}\right)\right) \psi\left(\log_2 N, \frac{t}{\sqrt{v_{l,u,d} \log_2 N}}\right)$$

$$+ \frac{1}{N^d} \tilde{R}\left(N, \frac{t}{\sqrt{v_{l,u,d} \log_2 N}}\right) \exp\left(-it \frac{e_{l,u,d}}{\sqrt{v_{l,u,d}}} \sqrt{\log_2 N}\right).$$

Since $\hat{g}_N(t)$ is a characteristic function, we have $1 = \psi_0 + r_0$ for $r_0 = (1/N^d)\tilde{R}(N, 0)$ and $\psi_0 = \Psi(\log_2 N, 0)$. We know that $(1/N^d)\tilde{R}(N, t) = O(N^{-\delta})$. Next, we can estimate the difference from $\hat{g}_N(t)$ to the characteristic function $\hat{f}(t) = e^{-t^2/2}$ of the normal distribution with mean 0 and variance 1, which is

$$|\hat{g}_N(t) - \hat{f}(t)|$$

$$= \left| e^{-t^2/2}\left(1 + O\left(\frac{t^3}{\log^{3/2} N}\right)\right)\left(\psi_0 + O\left(\frac{t}{\sqrt{\log N}}\right)\right)\right.$$

$$\left. + \frac{1}{N^d} \tilde{R}\left(N, \frac{t}{\sqrt{v_{l,u,d} \log_2 N}}\right) \exp\left(-it \frac{e_{l,u,d}}{\sqrt{v_{l,u,d}}} \sqrt{\log_2 N}\right) - (\psi_0 + r_0)e^{-t^2/2}\right|$$

$$= O\left(\frac{t}{\sqrt{\log N}}\right),$$

for $t = o(\sqrt{\log N})$.

Therefore, the Berry–Esseen inequality (see [11]) implies

$$\mathbb{P}\left(\frac{h(m_1, \ldots, m_d) - e_{l,u,d} \log_2 N}{\sqrt{v_{l,u,d} \log_2 N}} < x\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy + O\left(\frac{1}{\sqrt[4]{\log N}}\right).$$

For a specific digit set and dimension, we can compute the constants $e_{l,u,d}$ and $v_{l,u,d}$ explicitly.

Example 4.2. We consider the digit set $D_{-2,3}$ in dimension 2. See Example 3.7 and Figure 5 for the transducer. The adjacency matrix A of the underlying graph of this transducer is given in Table 1, where $z = e^{it}$.

The characteristic polynomial of A is

$$-(x - 1)x^7(x^2 - 2z)(x^3 - x^2 - xz - 2z)^2(x^5 - x^4 - 7x^3z - 20x^2z + 6xz^2 - 24z^2).$$

At $t = 0$, the dominating eigenvalue $\mu(0) = 4$ is a root of the fourth factor. Therefore the Taylor expansion of $\mu(t)$ around $t = 0$ is

$$\mu(t) = 4 + \frac{128i}{89}t - \frac{673216}{2114907}t^2 + O(t^3).$$

Hence the expected value of the Hamming weight is

$$\frac{32}{89} \log_2 N + O(1)$$

Table 1. Adjacency matrix of the transducer in Example 4.2

$$A = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
3z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
z & 0 & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
z & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
2z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
z & 0 & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
z & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
2z & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
z & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
z & 0 & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
z & z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
2z & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
2z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
3z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

and the variance is

$$\frac{63200}{2114907} \log_2 N + O(1).$$

In order to determine the constants $e_{l,u,d}$ and $v_{l,u,d}$ giving mean and variance in general, we rephrase the results of the 'full block-length' analysis in [7] in a probability model which is easily compared with our main results.

Lemma 4.3. *Let $k \geqslant w$ be a positive integer. Let $\widetilde{W}_k$ be the Hamming weight of the AJSF of a random vector $m = (m_1, \ldots, m_d)^T$ with equidistribution of all vectors $m = (m_1, \ldots, m_d)^T$ with $0 \leqslant m_i < 2^k$.*
    Then

$$\mathbb{E}\widetilde{W}_k = e_{l,u,d}k + O(1) \quad and \quad \mathbb{V}\widetilde{W}_k = v_{l,u,d}k + O(\sqrt{k})$$

for the constants given in Theorem 1.5.

Proof. For $j < k$, we denote the jth digit of the AJSF of a random vector $m = (m_1, \ldots, m_d)^T$ by $\widetilde{X}_j$, where we assume equidistribution of all vectors $m = (m_1, \ldots, m_d)^T$ with $0 \leqslant m_i < 2^k$.
    In [7, §6.2], the random variables X_j denoting the jth digit of a random AJSF has been considered, where the probability measure was defined to be the image of the Haar measure on the space of d-tuples of 2-adic integers under the AJSF, i.e., equidistribution on all residue classes modulo 2^l for all l has been assumed. Furthermore, W_j was defined to be the weight of the first j digits.

From Algorithm 1, it is clear that $\widetilde{X}_j$ only depends on m modulo 2^{j+w}. This implies that $\widetilde{X}_j$ and X_j are identically distributed for $j < k - w$. Therefore W_{k-w} and $\widetilde{W}_{k-w}$ are identically distributed, too. Furthermore, we always have $|\widetilde{W}_k - \widetilde{W}_{k-w}| \leqslant w$.

By [7, Theorem 6.7], we have

$$\mathbb{E}\widetilde{W}_{k-w} = \mathbb{E}W_{k-w} = e_{l,u,d}(k-w) + O(1), \quad \mathbb{V}\widetilde{W}_{k-w} = \mathbb{V}W_{k-w} = v_{l,u,d}(k-w) + O(1).$$

We conclude that

$$\mathbb{E}\widetilde{W}_k = \mathbb{E}\widetilde{W}_{k-w} + O(1) = \mathbb{E}W_{k-w} + O(1) = e_{l,u,d}k + O(1),$$
$$\mathbb{V}\widetilde{W}_k = \mathbb{V}\widetilde{W}_{k-w} + \mathbb{V}(\widetilde{W}_k - \widetilde{W}_{k-w}) + 2\operatorname{Cov}(\widetilde{W}_{k-w}, \widetilde{W}_k - \widetilde{W}_{k-w}) = \mathbb{V}W_{k-w} + O(\sqrt{k}),$$

where the Cauchy–Schwarz inequality has been used in the form

$$\operatorname{Cov}(\widetilde{W}_{k-w}, \widetilde{W}_k - \widetilde{W}_{k-w}) \leqslant \sqrt{\mathbb{V}\widetilde{W}_{k-w}\mathbb{V}(\widetilde{W}_k - \widetilde{W}_{k-w})}. \qquad \square$$

In the next lemma, we prove that the function $\Psi_1(x)$ is non-differentiable at any real number in the one-dimensional case. The proof uses the method presented by Tenenbaum [10]; see also Grabner and Thuswaldner [4].

Lemma 4.4. *Let $d = 1$. Then the function $\Psi_1(x)$ in Theorem 1.5 is nowhere differentiable.*

Proof. Let $g(N) = \lfloor 2^{-1-4w}N^{c/(c+1)} \rfloor$ be a positive integer-valued function with $c \in \mathbb{Z}$ and $c > 1/\delta - 1$. We have $g(N) = o(N)$ and $N^{1-\delta} \log N = o(g(N))$.

Assume Ψ_1 is differentiable at $x \in [0, 1)$. Let $2^x = \sum_{p=0}^{\infty} \varepsilon_p 2^{-p}$ be the standard binary digit expansion choosing the representation ending on 0^ω in the case of ambiguity. Further, let x_k, y_k and N_k be such that $2^{x_k} = \sum_{p=0}^{k} \varepsilon_p 2^{-p}$, $N_k = 2^{k(c+1)+x_k} \in \mathbb{Z}$ and $2^{k(c+1)+y_k} = N_k + g(N_k)$. Then we have

$$x - x_k = O(2^{-k}),$$
$$y_k - x_k = \log_2\left(1 + \frac{g(N_k)}{N_k}\right) = \frac{1}{\log 2}\frac{g(N_k)}{N_k} + O\left(\frac{g(N_k)^2}{N_k^2}\right),$$
$$\lim_{k\to\infty} y_k = x.$$

Because of the choice of $g(N)$, we have

$$g(N_k) < 2^{ck-4w},$$
$$\frac{g(N_k)}{N_k} = \Theta(2^{-k}).$$

We have $h(2^{p+4w}n + m) = h(n) + h(m)$ for $p \geqslant 0$ and $m < 2^p$ because 0^{4w} is a reset sequence leading to the initial state (see Theorem 3.6). Due to (4.5) and the periodicity

and continuity of Ψ_1, we have

$$\sum_{N_k \leqslant n < N_k + g(N_k)} h(n) \tag{4.6}$$

$$= g(N_k)h(N_k) + \sum_{n < g(N_k)} h(n)$$

$$= g(N_k)h(N_k) + e_{l,u,1}g(N_k)\log_2 g(N_k) + g(N_k)\Psi_1\left(\frac{c}{c+1}x\right) + o(g(N_k)).$$

On the other hand, we have

$$\sum_{N_k \leqslant n < N + g(N_k)} h(n) \tag{4.7}$$

$$= e_{l,u,1}(N_k + g(N_k))\log_2(N_k + g(N_k)) + (N_k + g(N_k))\Psi_1(\log_2(N_k + g(N_k)))$$

$$- e_{l,u,1}N_k \log_2 N_k - N_k\Psi_1(\log_2 N_k) + O(N_k^{1-\delta}\log N_k)$$

$$= e_{l,u,1}N_k(y_k - x_k) + N_k(\Psi_1(y_k) - \Psi_1(x_k)) + e_{l,u,1}g(N_k)(k(c+1) + y_k)$$

$$+ g(N_k)\Psi_1(y_k) + o(g(N_k)).$$

If we divide by $g(N_k)$ in (4.6) and (4.7), then we obtain

$$h(N_k) + e_{l,u,1}\log_2 g(N_k) + \Psi_1\left(\frac{c}{c+1}x\right)$$

$$= e_{l,u,1}(y_k - x_k)\frac{N_k}{g(N_k)} + \frac{N_k}{g(N_k)}(\Psi_1(y_k) - \Psi_1(x_k))$$

$$+ e_{l,u,1}(k(c+1) + y_k) + \Psi_1(y_k) + o(1).$$

Now we can write the difference of the Ψ_1 on the right-hand side in terms of the derivative

$$\Psi_1(y_k) - \Psi_1(x_k) = \Psi_1'(x)(y_k - x_k) + o(x - x_k) + o(|y_k - x|),$$

and we get

$$h(N_k) = -\Psi_1\left(\frac{c}{c+1}x\right) + \frac{e_{l,u,1}}{\log 2} + \frac{\Psi_1'(x)}{\log 2} + e_{l,u,1}\left(k + \frac{x}{c+1}\right) + \Psi_1(x) + o(1).$$

Next, we take the difference of two subsequent terms

$$h(N_{k+1}) - h(N_k) = e_{l,u,1} + o(1), \tag{4.8}$$

where the left-hand side is an integer. We have $e_{l,u,1} \notin \mathbb{Z}$ since $0 < e_{l,u,1} = 1/(w - 1 + \lambda) < 1$.

Therefore the right-hand side of (4.8) is not an integer for k large enough. This contradicts our assumption that Ψ_1 is differentiable in x. $\qquad\square$

5. Asymptotic distribution of the w-NAF

In this section we specialize the result of Theorem 1.5 to the w-NAF.

Theorem 5.1. *The weight $h(n)$ of the w-NAF of the integer n with equidistribution on $\{n \in \mathbb{Z} \mid 0 \leqslant n < N\}$ is asymptotically normally distributed. There exists a $\delta > 0$ such that the*

mean is

$$\frac{1}{w+1} \log_2 N + \Psi_1(\log_2 N) + O(N^{-\delta} \log N)$$

and the variance is

$$\frac{2}{(w+1)^3} \log_2 N - \Psi_1^2(\log_2 N) + \Psi_2(\log_2 N) + O(N^{-\delta} \log^2 N),$$

where Ψ_1 and Ψ_2 are continuous, 1-periodic functions on $\mathbb{R}$. If w is large enough, then

$$\delta = \log_2\left(1 + \frac{3\pi^2}{w^3}\right).$$

In particular, we have

$$\mathbb{P}\left(\frac{h(n) - \log_2 N/(w+1)}{\sqrt{2/(w+1)^3 \log_2 N}} < x\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy + O\left(\frac{1}{\sqrt[4]{\log N}}\right)$$

for all $x \in \mathbb{R}$.

This follows from the next lemma and Theorem 1.5.

Lemma 5.2. *The characteristic polynomial of the matrix A of the transducer in Figure 1 is $(x-1)(x^w - x^{w-1} - 2^{w-1}e^{it})$. The largest eigenvalue $\mu(t)$ is unique around $t = 0$ and $\mu(0) = 2$. Furthermore, for large enough w and $t = 0$, there is exactly one simple eigenvalue in*

$$T_k = \left\{x \in \mathbb{C} : |x| \geqslant \frac{2}{1 + \frac{3}{w}}, \left|\arg x - \frac{2k\pi}{w}\right| \leqslant \frac{\pi}{2w}\right\}$$

for each $k = 0, \ldots, w - 1$. Additionally, there is the obvious eigenvalue $x = 1$. The eigenvalues with the second largest absolute value are in T_1 and T_{w-1}. For each eigenvalue at $t = 0$, an expansion in $1/w$ can be computed with arbitrarily small error term.

Proof. The characteristic polynomial of A is obtained by Laplace expansion. With $z = 2/x$, the interesting factor of the characteristic polynomial is transformed into $z^w + z - 2 = 0$. The smallest root in absolute value of this polynomial is 1 because for $|z| < 1$ we have

$$|z^w + z| = |z| \cdot |z^{w-1} + 1| < 2|z| < 2.$$

We use the fixed-point equation $f_k(z) = z$ with

$$f_k(z) = (2 - z)^{1/w} e^{2\pi ik/w}$$

for $k = 0, \ldots, w - 1$. Here, we take the main branch of the wth root. After the substitution, we have

$$\widetilde{T}_k = \left\{z \in \mathbb{C} : |z| \leqslant 1 + \frac{3}{w}, \left|\arg z - \frac{2k\pi}{w}\right| \leqslant \frac{\pi}{2w}\right\},$$

which corresponds to $T_{-k \bmod w}$. For w large enough and $|z| \leqslant 1 + 3/w$, we have

$$|f'_k(z)| = \frac{1}{w}|2 - z|^{1/w - 1}$$

$$\leqslant \frac{1}{w}\left(1 - \frac{3}{w}\right)^{1/w - 1} \leqslant \frac{1}{w - 3}$$

and

$$|f_k(z)| \leqslant |f_k(z) - f_k(1)| + |f_k(1)|$$

$$\leqslant \frac{1}{w - 3}|z - 1| + 1 \leqslant \frac{2 + 3/w}{w - 3} + 1 \leqslant 1 + \frac{3}{w}.$$

Furthermore, we have

$$\left|\arg f_k(z) - \frac{2k\pi}{w}\right| \leqslant \frac{\pi}{2w}.$$

Thus, $f_k(\widetilde{T}_k) \subseteq \widetilde{T}_k$, and f_k is a contraction on $\widetilde{T}_k$ with Lipschitz constant $(w - 3)^{-1} < 1$. Therefore, there exists a unique fixed point of f_k in $\widetilde{T}_k$ for each k. Because $\widetilde{T}_k$ for $k = 0, \ldots, w - 1$ only intersect in 0, which is certainly no root of the polynomial, we found w distinct roots of the polynomial $z^w + z - 2$. Thus, we have found all roots of this polynomial. We only have to investigate $0 \leqslant k \leqslant w/2$ because the coefficients of the polynomial are real. Let $z \in \widetilde{T}_k$ be the fixed point of f_k. For $\rho = \exp(2\pi i/w)$, we have $|z - \rho^k| = O(1/w)$. Therefore, $z = \rho^k + O(1/w)$.

For $k \leqslant w^\alpha$ with a fixed $\alpha \in (1/2, 1)$, we have

$$z = f(z) = 1 + \frac{2\pi ik}{w} + O\left(\frac{k^2}{w^2}\right).$$

Iterating, we successively get

$$z = 1 + \frac{2\pi ik}{w} - \frac{2\pi^2 k^2}{w^2} - \frac{2\pi ik}{w^2} + O\left(\frac{k^3}{w^3}\right) \quad \text{and}$$

$$|z| = 1 + \frac{4\pi^2 k^2}{w^3} + O\left(\frac{k^3}{w^4}\right).$$

Therefore, for large w, only the fixed points for $k = w - 1, 0, 1$ are in the disk

$$\left\{z \in \mathbb{C} : |z| \leqslant 1 + \frac{10\pi^2}{w^3}\right\}.$$

For $k \geqslant w^\alpha$, we have

$$|2 - \rho^k| = \left(5 - 4\cos\left(\frac{2k\pi}{w}\right)\right)^{1/2} \geqslant \left(5 - 4\cos(2w^{\alpha - 1}\pi)\right)^{1/2}$$

$$= 1 + 4\pi^2 w^{2\alpha - 2} + O(w^{4\alpha - 4}),$$

$$|2 - z| \geqslant |2 - \rho^k| - |z - \rho^k|$$

$$= 1 + 4\pi^2 w^{2\alpha - 2} + O(w^{4\alpha - 4} + w^{-1}),$$

$$|f_k(z)| = \exp\left(\frac{1}{w}\log|2 - z|\right) \geqslant 1 + 4\pi^2 w^{2\alpha - 3} + O(w^{4\alpha - 5} + w^{-2}).$$

Thus for $k \geqslant w^{\alpha}$, the fixed point of f_k is not in the disk

$$\left\{ z \in \mathbb{C} : |z| \leqslant 1 + \frac{10\pi^2}{w^3} \right\}$$

for large w. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Acknowledgement

References

[1] Avanzi, R. M. (2005) A note on the signed sliding window integer recoding and a left-to-right analogue. In *Selected Areas in Cryptography* (H. Handschuh and A. Hasan, eds), Vol. 3357 of *Lecture Notes in Computer Science*, Springer, pp. 130–143.

[2] Barat, G. and Grabner, P. J. (2001) Distribution of binomial coefficients and digital functions. *J. London Math. Soc.* (2) **64** 523–547.

[3] Flajolet, P., Grabner, P., Kirschenhofer, P., Prodinger, H. and Tichy, R. F. (1994) Mellin transforms and asymptotics: Digital sums. *Theoret. Comput. Sci.* **123** 291–314.

[4] Grabner, P. and Thuswaldner, J. (2000) On the sum of digits function for number systems with negative bases. *Ramanujan J.* **4** 201–220.

[5] Grabner, P. J., Heuberger, C. and Prodinger, H. (2004) Distribution results for low-weight binary representations for pairs of integers. *Theoret. Comput. Sci.* **319** 307–331.

[6] Heuberger, C. and Muir, J. A. (2006) Minimal weight and colexicographically minimal integer representations: Online resources. http://www.math.tugraz.at/~cheub/publications/colexi/.

[7] Heuberger, C. and Muir, J. A. (2007) Minimal weight and colexicographically minimal integer representations. *J. Math. Cryptol.* **1** 297–328.

[8] Muir, J. A. and Stinson, D. R. (2006) Minimality and other properties of the width-*w* nonadjacent form. *Math. Comp.* **75** 369–384.

[9] Straus, E. (1964) Addition chains of vectors (problem 5125). *Amer. Math. Monthly* **71** 806–808.

[10] Tenenbaum, G. (1997) Sur la non-dérivabilité de fonctions périodiques associées à certaines formules sommatoires. In *The Mathematics of Paul Erdős* I, Vol. 13 of of *Algorithms Combin.*, Springer, pp. 117–128.

[11] Vaaler, J. D. (1985) Some extremal functions in Fourier analysis. *Bull. Amer. Math. Soc.* (NS) **12** 183–216.